Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και
Μηχανικων Υπολογιστων

Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Κατανοώντας τα Νευρωνικά Δίκτυα με Κάψουλες

Προς έναν Κλιμακώσιμο
Αλγόριθμο Δρομολόγησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΛΕΞΑΝΔΡΟΣ Κ. ΜΠΑΡΜΠΕΡΗΣ

Επιβλέπων: Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2022

Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και
Μηχανικων Υπολογιστων

Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Κατανοώντας τα Νευρωνικά Δίκτυα με Κάψουλες

Προς έναν Κλιμακώσιμο
Αλγόριθμο Δρομολόγησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΛΕΞΑΝΔΡΟΣ Κ. ΜΠΑΡΜΠΕΡΗΣ

Επιβλέπων: Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 21η Οκτωβρίου 2022.

.....	.....	.....
Στέφανος Κόλλιας	Αθανάσιος Βουλόδημος	Γιώργος Στάμου
Καθηγητής Ε.Μ.Π.	Επίκουρος Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2022

Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και
Μηχανικων Υπολογιστων

Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δε μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

(υπογραφή)

.....

Αλέξανδρος Μπαρμπέρης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Αθήνα, Οκτώβριος 2022

Περίληψη

Τελευταία, στον κλάδο της Τεχνητής Νοημοσύνης, παρατηρείται ραγδαία αύξηση του μεγέθους των Βαθιών Νευρωνικών Δικτύων. Με τα νέα συστήματα να έχουν κολοσσιαίο ενεργειακό κόστος για την ανάπτυξή τους, προκύπτει το ερώτημα του αν η απόδοσή τους επιδέχεται βελτίωση.

Μια ελπιδοφόρα λύση είναι τα Νευρωνικά Δίκτυα με Κάψουλες που, αντιμετωπίζοντας ανεπάρκειες στον σχεδιασμό της αρχιτεκτονικής των δικτύων, οδηγούν σε συστήματα Όρασης Υπολογιστών με υψηλή απόδοση. Σε αυτά, οι αποκρίσεις τεχνητών νευρώνων του δικτύου οργανώνονται σε ομάδες, τις κάψουλες. Η κάθε κάψουλα μαθαίνει να αναγνωρίζει ένα συγκεκριμένο αντικείμενο (ή τμήμα του). Μέσω μιας διαδικασίας που θυμίζει ανάστροφα γραφικά, αποδομεί το αντικείμενο σε χαρακτηριστικές ιδιότητες όπως η πόζα, τις οποίες ενθυλακώνει. Επειδή σε ένα Νευρωνικό Δίκτυο με Κάψουλες, περιλαμβάνονται πολλά επίπεδα από αυτές, σχηματίζεται μια ιεραρχική διάταξη όπου κάψουλες χαμηλότερων επιπέδων (αναπαριστούν τμήματα αντικειμένου) δρομολογούνται σε ανώτερες κάψουλες που αναγνωρίζουν μεγαλύτερα αντικείμενα και σχηματίζονται από τη σύνθεση μερών τους. Η ιεραρχική αποδόμηση των αντικειμένων μαζί με την εξαγωγή των παραμέτρων πόζας αυτών επιτρέπει την εύρωστη μοντελοποίηση των αντικειμένων από το δίκτυο οδηγώντας σε αποδοτικότερη αναγνώρισή τους υπό διαφορετικές οπτικές γωνίες.

Δυστυχώς, τα Νευρωνικά Δίκτυα με Κάψουλες δεν έχουν λάβει τη δέουσα προσοχή, γεγονός που αποδίδεται στη δυσνοητότητά τους και στην αδυναμία κλιμάκωσής τους σε μεγαλύτερα συστήματα. Η αντιμετώπιση αυτών των προβλημάτων αποτελεί τον στόχο της παρούσας εργασίας.

Το πρώτο πρόβλημα, το προσεγγίζουμε διατελώντας μια διεξοδική μελέτη στα βασικά έργα που θεμελιώνουν την εν λόγω τεχνολογία. Η μελέτη περιλαμβάνει, μεταξύ άλλων, πρωτότυπα πειράματα που φανερώνουν την εσωτερική λειτουργία της και γραφικά σχήματα που διευκολύνουν την κατανόηση της. Αναφορικά με το δεύτερο πρόβλημα, δανειζόμενοι ιδέες από τον δημοφιλή Μηχανισμό Προσοχής και από τους χάρτες αυτο–οργάνωσης αντίστοιχα, δημιουργούμε δύο νέα, γρήγορα συστήματα Νευρωνικών Δικτύων με Κάψουλες.

Μέσα από τα πειράματα, πιστοποιούμε έμπρακτα όλους τους θεμελιακούς ισχυρισμούς της τεχνολογίας που μελετάμε ενώ παράλληλα αποδεικνύουμε ότι το ένα εκ των γρήγορων παραλλαγών που προτείνουμε εμφανίζει την τρίτη καλύτερη επίδοση σε πρόβλημα ορόσημο (smallNORB) ανοίγοντας τον δρόμο για αποδοτικά, κλιμακώσιμα συστήματα.

Λέξεις κλειδιά

Τεχνητή Νοημοσύνη, Μηχανική Μάθηση, Βαθιά Νευρωνικά Δίκτυα, Όραση Υπολογιστών, Συνελικτικά Νευρωνικά Δίκτυα, Διανυσματικές Κάψουλες, Νευρωνικά Δίκτυα με Κάψουλες, Δυναμική Δρομολόγηση με Συμφωνία, Δρομολόγηση Μεγιστοποίησης Προσδοκιών, Χάρτης Αυτο-οργάνωσης, Μηχανισμός Προσοχής, Κάψουλες με Μηχανισμό Προσοχής Πολλών Κεφαλών, Αλγόριθμος Δρομολόγησης με Προσοχή, Μετασχηματιστές

Abstract

Lately, in the field of Artificial Intelligence there is a growing trend for Deeper Neural Networks. Given that these costly Machine Learning systems require extreme amounts of energy resources to be developed, it makes sense to ask whether their efficiency can be improved.

Capsule Networks, a modern approach which tackles inefficiencies in core Neural Network's architectural design principles, constitutes a promising solution for building more efficient Computer Vision systems. More specifically, in the proposed systems neural activations are grouped together into «capsules». Every capsule is trained to recognize a specific entity (object or object part) and encapsulates it's equivariant properties (e.g. it's pose) which it computes through a procedure that resembles inverse graphics. When feeding a Capsule Network comprised of multiple capsule layers with an image depicting an object, a parse tree is dynamically created as lower level capsules (representing object parts) selectively activate higher level capsules (representing bigger objects) through a routing algorithm. The resulting hierarchical decomposition of the entities along with the extraction of their equivariant properties helps the Network form robust, intrinsic object models thus leading to efficient, viewpoint invariant object recognition.

Unfortunately, Capsule Networks have not received much attention due to their abstruse nature as well as their inability to scale into larger systems. Our goal in the thesis at hand is to address those two problems, pushing towards a more sustainable future.

The first problem is confronted by our thorough investigation of the fundamental academic articles which define the technology. It also includes extensive, novel experimentation and graphical representations of the results, all aiming to elucidate the new systems. Regarding the second problem, we propose two alternative, fast Capsule Network systems inspired by the popular Attention Mechanism and the algorithm used in Self-Organizing Maps respectively.

Through our experiments, we testify the foundational assumptions of Capsule Networks and make significant observations. In addition, we show that one of our proposed systems archives state-of-the-art results on the smallNORB benchmark while greatly reducing computational load and thus, paving the way for efficient, scalable Capsule Networks.

Keywords

Artificial Intelligence, AI, Machine Learning, Deep Neural Networks, Computer Vision, Convolutional Networks, Vector Capsules, Matrix Capsules, Capsule Neural Networks, CapsNet, Capsule Network, Dynamic Routing by Agreement, Expectation Maximization Routing, Efficient CapsNet, Self Organizing Map, SOM, Kohonen, Competitive Learning, Transformer, Attention, Self-Attention Capsules, RoMAV, Routing by Merged Agreeing Votes, RoWSS, Routing by Winner of Similarity Score, RoWLS, Routing by Winner of Length Scores

Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής αυτής εργασίας, το ταξίδι στη Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσοβίου Πολυτεχνείου έρχεται στο τέλος του. Συνεπώς, θα ήθελα να ευχαριστήσω όλους τους συνοδοιπόρους που με υποστήριξαν τόσο στην εκπόνηση της διπλωματικής μου εργασίας όσο και κατά τη διάρκεια των σπουδών μου.

Καταρχάς, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή αυτής της διπλωματικής, κ. Στέφανο Κόλλια, για την ευκαιρία που μου έδωσε να ασχοληθώ με το συγκεκριμένο θέμα καθώς και για το ενδιαφέρον που μου καλλιέργησε κατά τη διάρκεια των σπουδών μου.

Ιδιαίτερες ευχαριστίες θα ήθελα να δώσω στον υποψήφιο διδάκτωρ κ. Εμμανουήλ Σεφέρη για την καθοδήγησή του και τη διαρκή και άμεση στήριξή του. Η συμβολή του είναι εμφανής σε όλη την έκταση της παρούσας εργασίας.

Ευχαριστώ, ακόμα, τα μέλη της εξεταστικής επιτροπής, τον κ. Γιώργο Στάμου και τον κ. Αθανάσιο Βουλόδημο, ως εξεταστές αλλά και για όσα μου προσέφεραν ως καθηγητές μου.

Έπειτα, θέλω να ευχαριστήσω τους στενούς φίλους και κοντινούς μου ανθρώπους οι οποίοι ομόρφυναν τα αξέχαστα φοιτητικά μου χρόνια. Ευελπιστώ, η παρέα τους να συνοδεύει και τα επόμενα εγχειρήματα της ζωής μου.

Τέλος, το μεγαλύτερο ευχαριστώ οφείλω στην οικογένεια μου, και κυρίως τους γονείς μου, για την ανιδιοτελή στήριξη και την αγάπη τους όλα αυτά τα χρόνια.

Αλέξανδρος Μπαρμπέρης,
Αθήνα, 21 Οκτωβρίου 2022

Περιεχόμενα

Περίληψη **i**

Abstract **iii**

Ευχαριστίες **v**

Περιεχόμενα **ix**

1 Εισαγωγή **1**
1.1 Επισκόπηση του Κόσμου της Τεχνητής Νοημοσύνης 1
1.2 Ιστορική Αναδρομή Τεχνιτής Νοημοσύνης 3
1.3 Κίνητρο 11
1.4 Συνεισφορά Εργασίας 11
1.5 Οργάνωση του Τόμου 12

2 Θεωρητικό Υπόβαθρο **13**
2.1 Τεχνητά Νευρωνικά Δίκτυα 13
2.1.1 Μηχανική Μάθηση 14
2.1.2 Εκμάθηση Χαρακτηριστικών 15
2.1.3 Πολυεπίπεδα Νευρωνικά Δίκτυα 17
2.2 Νευρωνικά Δίκτυα με Κάψουλες 34
2.2.1 Στοιχεία Έμπνευσης των Νευρωνικών Δικτύων με Κάψουλες 35
2.2.2 Θετικά Γνωρίσματα Συνελικτικών Νευρωνικών Δικτύων 36
2.2.3 Βασικές Ανεπάρκειες των Συνελικτικών Νευρωνικών Δικτύων 37
2.2.4 Αρχές Λειτουργείας Νευρωνικών Δικτύων με Κάψουλες 40
2.3 Μετασχηματιστές 51
2.3.1 Επαναλαμβανόμενα Νευρωνικά Δίκτυα 51
2.3.2 Μηχανισμός Προσοχής 57
2.3.3 Μετασχηματιστές 60
2.4 Χάρτες Αυτο-οργάνωσης 67
2.4.1 Αρχιτεκτονική Χάρτη Αυτο–οργάνωσης 68
2.4.2 Ο Ανθρώπινος Εγκέφαλος ως Πηγή Έμπνευσης 69
2.4.3 Αλγόριθμος Σχηματισμού Χάρτη Αυτο–οργάνωσης 70

3 Βιβλιογραφική Επισκόπηση **73**
3.1 Θεμελίωση Θεωρίας Νευρωνικών Δικτύων με Κάψουλες 74
3.2 Παραλλαγές Νευρωνικών Δικτύων με Κάψουλες 78
3.3 Νευρωνικά Δίκτυα με Κάψουλες με Μη–Επιβλεπόμενη Μάθηση 88
3.4 Λοιπές Δημοσιεύσεις που Ενέμπνευσαν τις Μεθόδους της Παρούσας Διπλωματικής 89

4 Μέθοδος **92**
4.1 Dynamic Routing Between Capsules 92
4.1.1 Αρχιτεκτονική Νευρωνικού Δικτύου 92
4.1.2 Συνάρτηση Σύνθλιψης 94
4.1.3 Δυναμικός Αλγόριθμος Δρομολόγησης μέσω Συμφωνίας 95
4.1.4 Συνάρτηση Σφάλματος 96
4.1.5 Παραλλαγές Δυναμικού Αλγορίθμου Δρομολόγησης 97
4.2 Matrix Capsules with EM Routing 99
4.2.1 Αρχιτεκτονική Νευρωνικού Δικτύου 100
4.2.2 Υπολογισμός Τιμής Ενεργοποίησης 103
4.2.3 Αλγόριθμος Δρομολόγησης EM 105
4.2.4 Συνάρτηση Σφάλματος και Λοιπά Στοιχεία Υλοποίησης 107
4.3 Multi-Head Self-Attention Capsules 107
4.3.1 Αρχιτεκτονική Νευρωνικού Δικτύου 108
4.3.2 Αλγόριθμοι Δρομολόγησης 110
4.3.3 Αρχιτεκτονική Αποκωδικοποιητή 120
4.3.4 Συνάρτηση Σφάλματος και Λοιπά Στοιχεία Υλοποίησης 121
4.3.5 Αλγόριθμοι με Μηχανισμούς Αυτοπροσοχής Πολλών Κεφαλών 121
4.4 SOM-Caps 126
4.4.1 Αρχιτεκτονική Νευρωνικού Δικτύου 126
4.4.2 Αλγόριθμος Δρομολόγησης 128
4.4.3 Λοιπά Στοιχεία Υλοποίησης 135

5 Πειραματική Μελέτη **136**
5.1 Πλατφόρμα Διεξαγωγής Πειραμάτων, Μετρικές και Σύνολα Δεδομένων 137
5.1.1 Πειραματική Πλατφόρμα 137
5.1.2 Μετρικές Επίδοσης 138
5.1.3 Σύνολα Δεδομένων 139
5.2 Πειραματική Μελέτη Μεθόδου 1 140
5.2.1 Εύρεση Βέλτιστων Υπερπαραμέτρων 141
5.2.2 Επιλεκτική Εμβάθυνση Πειραμάτων και Σύγκριση 151
5.3 Ειδικά Πειράματα Μεθόδου 1 153
5.3.1 Τι Μαθαίνει να Αναπαριστά η Κάθε Διάσταση του Διανύσματος DigitCap 154
5.3.2 Κριτήριο Επιλογής Ψήφων Αλγορίθμου Δυναμικής Δρομολόγησης με Συμφωνία 159
5.3.3 Συμφωνία Ψήφων 160
5.3.4 Κατανομή των Ψήφων 162
5.3.5 Κατανομή των Βαρών Δρομολόγησης 163
5.3.6 Ιστογράμματα των Σταθμισμένων Ψήφων ανα Αριθμό Επαναλήψεων . . . 166
5.4 Πειραματική Μελέτη Μεθόδου 2 166
5.4.1 SmallNORB 167
5.4.2 MNIST 168
5.4.3 Δοκιμή στο SmallNORB με Pretrained Model 168

5.5 Πειραματική Μελέτη Μεθόδου 3 169
5.5.1 Αναζήτηση στον Χώρο των Υπερπαραμέτρων 171
5.5.2 Πειράματα Παραμέτρου Ανακατασκευής 178
5.5.3 Επιλεκτική Εμβάθυνση στα Σύνολα Δεδομένων 178
5.5.4 Ειδικά Πειράματα 182
5.6 Πειραματική Μελέτη Μεθόδου 4 184
5.6.1 Αναζήτηση στον Χώρο των Υπερπαραμέτρων 184
5.6.2 Επιλεκτική Εμβάθυνση με Δοκιμή σε Όλα τα Προβλήματα 189
5.7 Σύγκριση Πειραματικών Αποτελεσμάτων Μεθόδων 191
5.7.1 Σύνολο Δεδομένων SmallNORB 192
5.7.2 Σύνολο Δεδομένων MNIST 193
5.7.3 Σύνολο Δεδομένων CIFAR10 194
5.7.4 Σύνολο Δεδομένων MultiMNIST 195
5.7.5 Σύνολο Δεδομένων FashionMNIST 196

6 Επίλογος **198**
6.1 Σύνοψη και Συμπεράσματα 198
6.2 Μελλοντικές Κατευθύνσεις 200

Βιβλιογραφία **202**

Α΄ Ορισμοί Εννοιών **212**

Β΄ Απόδοση Ξενόγλωσσων Όρων **217**

Γ΄ Αναλυτική Περιγραφή Αλγορίθμου Δρομολόγησης με SOM **218**

Λίστα Αλγορίθμων

1 Δυναμικός Αλγόριθμος Δρομολόγησης μέσω Συμφωνίας 95
2 Αλγόριθμος Argmax Scaled Routing 98
3 Αλγόριθμος Argmax Routing 99
4 Αλγόριθμος Max Routing 99
5 Αλγόριθμος Δρομολόγησης Βασισμένος στον EM 106
6 Απλοϊκός Αλγόριθμος Δρομολόγησης με Αυτο–προσοχή 113
7 Αλγόριθμος Δρομολόγησης με Αυτο–προσοχή 1 (Αλγόριθμος RoMAV) 116
8 Αλγόριθμος Δρομολόγησης με Αυτο–προσοχή 2 (Αλγόριθμος RoWSS) 119
9 Αλγόριθμος Δρομολόγησης με Αυτο–προσοχή 3 (Αλγόριθμος RoWLS) 120
10 Διαδικασία Αυτο–Προσοχής Πολλών Κεφαλών (Multi–Head Procedure) 122
11 Συμπληρωματικές, Βοηθητικές Διαδικασίες (Multi–Head Helper Procedures) . 123
12 Αλγόριθμος 1 με Αυτο–Προσοχή Πολλών Κεφαλών (Multihead RoMAV) . . . 124
13 Αλγόριθμος 2 με Αυτο–Προσοχή Πολλών Κεφαλών (Multihead RoWSS) . . . 124
14 Αλγόριθμος 3 με Αυτο–Προσοχή Πολλών Κεφαλών (Multihead RoWLS) . . . 125
15 Αλγόριθμος Δρομολόγησης Βασισμένος στον SOM (SOM-Based Routing) . . 132

Κεφάλαιο 1

Εισαγωγή

«Η επιτυχημένη δημιουργία [γενικευμένης] Τεχνητής Νοημοσύνης θα είναι το μεγαλύτερο γεγονός στην ανθρώπινη ιστορία. Δυστυχώς, ίσως είναι και το τελευταίο εάν δε μάθουμε πώς να αποφεύγουμε τα ρίσκα.» — Stephen Hawking

1.1 Επισκόπηση του Κόσμου της Τεχνητής Νοημοσύνης

Είναι πλέον γεγονός, η Τεχνητή Νοημοσύνη (Artificial Intelligence - AI) εντοπίζεται σε πολλές εκφάνσεις της καθημερινότητας των περισσότερων ανθρώπων [1]. Δεν αποτελεί απλά έναν ακόμα μοδάτο όρο που καταχράται στον χώρο της αγοραστικής (marketing). Εν αντιθέσει, διαδραματίζει καθοριστικό ρόλο στην εργασία μας, στη μετακίνησή μας και στην ψυχαγωγία μας. Για παράδειγμα, εντοπίζεται στις μηχανές αναζήτησης όπως η Google, στους ηλεκτρονικούς χάρτες πλοήγησης αλλά και στα συστήματα συστάσεων (Recomender systems) όπως αυτό του YouTube, του Twitter και του Netflix [2] που εξατομικεύουν το προβαλλόμενο περιεχόμενο στα ενδιαφέροντα του χρήστη.

Η επιρροή που έχει η Τεχνητή Νοημοσύνη είναι ακόμα πιο ευδιάκριτη αν τη μελετήσει κανείς υπό μια συλλογική σκοπιά. Για παράδειγμα, στον χώρο του επιχειρείν, η αξιοποίηση τεχνολογιών Τεχνητής Νοημοσύνης έχει αποδειχθεί ότι αυξάνει την επιχειρηματική αξία (business value) μέσα από τη βελτίωση της επίδοσης τόσο στο οικονομικό (financial), αγοραστικό (marketing) και διοικητικό (administrative) επίπεδο όσο και στο επίπεδο επιχειρηματικών διαδικασιών (business process) [3]. Χαρακτηριστικό παράδειγμα αποτελεί η χρήση των συστημάτων συστάσεων αφού με το να φιλτράρουν το περιεχόμενο και να παρουσιάζουν στον χρήστη μόνο αυτό που του είναι οικείο και θεμιτό, αυξάνουν τον βαθμό ενασχόλησής του με κίνδυνο την παγίδευσή του σε μια «φούσκα προκατειλημμένου φιλτραρίσματος» («biased filter bubble») [4]. Άλλωστε, όπως δηλώνει η ομάδα ανάπτυξης του εν λόγω συστήματος για λογαριασμό της συνδρομητικής υπηρεσίας streaming, Netflix, «Πιστεύουμε ότι αθροιστικά, η επίδραση της εξατομίκευσης και των συστάσεων μας εξοικονομεί ένα δισεκατομμύριο δολάρια τον χρόνο» [5].

Στον εργασιακό χώρο, πολλές δουλειές που περιλαμβάνουν επαναλαμβανόμενες, προβλέψιμες εργασίες κυρίως στον τομέα της βιομηχανίας και της γεωργίας αντικαθίστανται από αυτοματισμούς Τεχνητής Νοημοσύνης (AI automations) εκτοπίζοντας έτσι τον άνθρωπο. Η μείωση των διαθέσιμων θέσεων εργασίας στους τομείς αυτούς δοκιμάζει τα όρια του κοινωνικού οικοδομήματος: τα «εκτοπισμένα» άτομα καλούνται να αποκτήσουν νέες δεξιότητες προκειμένου να βρουν απασχόληση στις πιο δημιουργικές (και συνάμα λιγότερο τυποποιήσιμες) θέσεις του εξελισσόμε-

νου τομέα των υπηρεσιών [6]. Βέβαια, η μείωση των θέσεων εργασίας επαναλαμβανόμενης φύσης είναι μόνο η μια πλευρά του νομίσματος. Σύμφωνα με αναλύσεις, μέχρι την επόμενη δεκαετία οι εφαρμογές της Τεχνητής Νοημοσύνης εν δυνάμει θα αυξήσουν το παγκόσμιο Ακαθάριστο Εθνικό Προϊόν (Gross Domestic Product - GDP) κατά 26% (δεκαπέντε τρισεκατομμύρια δολάρια) [7]. Αυτό, με τη σειρά του, θα οδηγήσει στη δημιουργία πολλών νέων θέσεων εργασίας έτσι ώστε να μην παρατηρηθεί αύξηση στους δείκτες ανεργίας [7, 8].

Τέλος, δε θα μπορούσαμε να παραλείψουμε την επιρροή που έχει η Τεχνητή Νοημοσύνη στον χώρο της υγείας. Οι εφαρμογές είναι ατελείωτες: από συστήματα για πρόωρη διάγνωση ασθενειών μέχρι ρομποτικά συστήματα υποβοήθησης χειρουργείου [9]. Αξιοσημείωτη είναι επίσης η επιτυχημένη εφαρμογή της για την πρόβλεψη της τρισδιάστατης δομής των πρωτεϊνών [10] - ένα θέμα με σημαντικές προεκτάσεις που απασχολούσε την επιστημονική κοινότητα για 50 χρόνια. Παρόλα αυτά, μαζί με την προσπάθεια για αξιοποίηση των νέων τεχνολογιών στην κλινική πράξη προκύπτουν νέες προκλήσεις. Μια πρώτη δυσκολία είναι η ανάπτυξη συστημάτων Τεχνητής Νοημοσύνης που απαιτούν μεγάλο όγκο δεδομένων σε χώρους προβλημάτων όπου αυτά σπανίζουν (όπως για παράδειγμα, στην περίπτωση μιας ασυνήθιστης ασθένειας όπου ο αριθμός των ιατρικών υποθέσεων είναι ελάχιστος). Αυτή η δυσκολία εντείνεται αφενός λόγω της έλλειψης ασφαλών υποδομών για τη συλλογή ιατρικών δεδομένων [11] και αφετέρου λόγω της απόρρητης φύσης αυτών, κάτι που δυσχεραίνει τον ελεύθερο διαμοιρασμό τους. Μια τελευταία δυσκολία αποτελεί το γεγονός ότι πολλά συστήματα Τεχνητής Νοημοσύνης που έχουν αναπτυχθεί σε περιβάλλον εργαστηρίου (lab setting) δεν παρέχουν αρκετά κίνητρα για μεταστροφή της καθιερωμένης κλινικής πράξης [11]. Για να επιτευχθεί κάτι τέτοιο, μεταξύ άλλων θα πρέπει τα συστήματα να αποδίδουν αποδεδειγμένα τόσο καλά όσο και το καταρτισμένο προσωπικό στο συγκεκριμένο πεδίο εφαρμογής τους και να παρέχουν πληροφορίες που θα τα καθιστούν περισσότερο έμπιστα π.χ. αιτιολογώντας την απόφασή τους (explainability), παρέχοντας μια μετρική αβεβαιότητας (uncertainty) ή δίνοντας τη δυνατότητα αλληλεπίδρασης [12].

Αντιλαμβανόμενοι το εύρος των εφαρμογών της Τεχνητής Νοημοσύνης, η εκτίμηση από την International Data Corporation - IDC πως οι Ευρωπαϊκές δαπάνες σε τέτοιες εφαρμογές θα έχουν σχεδόν τριπλασιαστεί μέσα στα επόμενα τρία χρόνια δε θα πρέπει να μας εκπλήσσει. Ωστόσο, με τη μεγάλη ισχύ έρχεται και μεγάλη ευθύνη. Είναι αλήθεια, η Τεχνητή Νοημοσύνη είναι ήδη εδώ και θα συνεχίζει να επιδρά όλο και εντονότερα στην καθημερινότητα μας και στην κοινωνία. Βέβαια, η σημερινή Τεχνητή Νοημοσύνη είναι εντελώς διαφορετική από αυτό που φαντάζονταν η κοινή γνώμη τις προηγούμενες δεκαετίες (σαφώς επηρεασμένη από ταινίες επιστημονικής φαντασίας όπως το Terminator). Θα λέγαμε, αντίθετα, πως εμφανίζεται περισσότερο με μια περιορισμένη μορφή στην εκάστοτε συγκεκριμένη εφαρμογή (narrow AI). Έτσι, ένα «εφυές» σύστημα για μια εργασία δεν μπορεί να «γενικεύσει» και να εφαρμοστεί σε άλλο χώρο προβλημάτων. Ούτε λόγος δε για αισθήματα και υπαρξιακή συνείδηση· αυτά (ακόμα) ανήκουν στην επιστημονική φαντασία. Αυτό όμως δε σημαίνει ότι η επιπόλαιη χρήση της Τεχνητής Νοημοσύνης δεν ελοχεύει κινδύνους. Σύμφωνα με το περιοδικό Spectrum της IEEE [13] προτού επιτευχθεί Τεχνητή Νοημοσύνη επιπέδου ανθρώπου (human-like Artificial Inteligence) - αυτή στην οποία αναφέρεται ο Stephen Hawking - υπάρχουν ήδη πολλά σενάρια όπου εφαρμογές της μπορούν να αποβούν μοιραίες. Ενδεικτικά, ένα από αυτά είναι τα deepfakes - ψεύτικα πολυμέσα βίντεο και εικόνας κατασκευασμένα από εφαρμογές Τεχνητής Νοημοσύνης - έχουν υπονομεύσει

την εμπιστοσύνη στα συστήματα πληροφόρησης. Επιπρόσθετα, ένα ακόμα καταστροφικό σενάριο σχετίζεται με την ιδιωτικότητα (privacy) και την ελεύθερη βούληση (free will). Με την παραχώρηση ευαίσθητων δεδομένων σε επιχειρήσεις και κυβερνήσεις τους παρέχουμε τη δυνατότητα να μας εποπτεύουν ακόμα και να μας χειραγωγούν. Ένα τελευταίο σενάριο για το οποίο διαδραματίζουν άμεσο ρόλο τα κοινωνικά δίκτυα είναι αυτό του μειωμένου διαστήματος προσοχής (short attention span) ως απόρροια της εκμετάλλευσης του μηχανισμού επιβράβευσης του εγκεφάλου ώστε οι χρήστες να εθίζονται σε αυτά. Το περιοδικό καλεί τον αναγνώστη να αναλογιστεί τις συνέπειες της συνεχόμενης βελτίωσης των μηχανισμών που μας καθηλώνουν από τη νέα τεχνολογία. Συμπερασματικά, η Τεχνητή Νοημοσύνη αν και δεν προσομοιάζει την ανθρώπινη νοημοσύνη δεν παύει να αποτελεί μια πολύ ισχυρή τεχνολογία που μπορεί να αποβεί είτε σωτήρια είτε μοιραία ανάλογα με τον τρόπο αξιοποίησής της.

Είναι λοιπόν απαραίτητη η εξασφάλιση της συνετής χρήσης αυτών των τεχνολογιών μέσω μιας σειράς κανονισμών. Στην Ευρωπαϊκή Ένωση, μια σειρά από διατάξεις επιχειρούν να θέσουν ένα νομοθετικό πλαίσιο ώστε να ωθήσουν στην αξιοποίηση της Τεχνητής Νοημοσύνης διασφαλίζοντας παράλληλα την ασφάλεια των θεμελιωδών δικαιωμάτων [14]. Άλλωστε, σύμφωνα με την von der Lein [15], «Η Τεχνητή Νοημοσύνη πρέπει να εξυπηρετεί τους ανθρώπους και συνεπώς, πρέπει πάντα να συμμορφώνεται με τα δικαιώματά τους. Αυτός είναι ο λόγος που ένα άτομο πρέπει πάντα να έχει τον έλεγχο στην περίπτωση κρίσιμων αποφάσεων[...] Εφαρμογές της Τεχνητής Νοημοσύνης που μπορεί να παρέμβουν στα ανθρώπινα δικαιώματα θα πρέπει να ελέγχονται και να πιστοποιούνται πριν φτάσουν στην Ευρωπαϊκή αγορά». Αν και οι αυστηροί διακανονισμοί καθυστερούν τη μετάβαση εφαρμογών Τεχνητής Νοημοσύνης από το εργαστήριο στην αγορά, εξασφαλίζουν την ασφάλειά τους συμβάλλοντας στην αξιοπιστία τους.

Με την παραπάνω σύντομη εισαγωγή καλύψαμε εμπεριστατωμένα πολλά από τα μη τεχνικά θέματα που σχετίζονται με την Τεχνητή Νοημοσύνη (Artificial Inteligence). Αναλυτικότερα, αρχίσαμε από παραδείγματα εντοπισμού της στην καθημερινή ζωή σε ατομικό και σε συλλογικό επίπεδο με τα οποία αντιληφθήκαμε τη σημασία της. Συνεχίσαμε με τους κινδύνους που ελοχεύει η απερίσκεπτη εφαρμογή της σε συγκεκριμένες εργασίες επισημαίνοντας ταυτόχρονα ότι η τεχνητή νοημοσύνη απέχει από την ανθρώπινη. Κλείσαμε, με μερικές από τις προσπάθειες που γίνονται σε Ευρωπαϊκό επίπεδο για την αποφυγή αυτών των κινδύνων. Πολλά από τα προαναφερθέντα στοιχεία πιθανότατα να είναι ήδη γνωστά σε έναν έμπειρο αναγνώστη. Εντούτοις, εξυπηρετούν σε μια ομαλή εισαγωγή για τον αρχάριο και σε μια υπενθύμιση για τον έμπειρο αναγνώστη του κόσμου της Τεχνητής Νοημοσύνης. Στην επόμενη υπο-ενότητα αυτού του κεφαλαίου θα παρουσιάσουμε μια ιστορική αναδρομή της τεχνητής νοημοσύνης. Έτσι, ο αναγνώστης θα κατανοήσει σε βάθος την έννοια γύρω από την οποία εκτυλίσσεται η παρούσα διπλωματική προτού εισαχθεί στο συγκεκριμένο τεχνικό της θέμα. Έπειτα, περιγράφεται το κίνητρο που με ωθούσε καθ'όλη τη διάρκεια συγγραφής του έργου. Τέλος, αναφερόμαστε στην τεχνική συνεισφορά της παρούσας εργασίας και στην οργάνωση του τόμου.

1.2 Ιστορική Αναδρομή Τεχνητής Νοημοσύνης

Οι εφευρέτες οραματίζονται εδώ και χιλιετίες τη δημιουργία μηχανών που σκέφτονται. Ήδη, γύρω στο 700 π.Χ. αναφέρεται από τον Ησίοδο ο Τάλος: ο μυθικός χάλκινος γίγαντας φτιαγμένος

από τον Ήφαιστο με αποστολή να προστατεύσει το νησί της Κρήτης από τους επιδρομείς [16]. Παρόμοια παραδείγματα αποτελούν αυτά της Πανδώρας και της Γαλάτειας. Η μακρόβια επιθυμία για απομίμηση της νοημοσύνης μαρτυρά την αξία που της δίνει ο άνθρωπος. Γεγονός απόλυτα δικαιολογημένο αφού η νοημοσύνη - η νοητική ικανότητα που μας επιτρέπει να σκεπτόμαστε λογικά, να επιλύουμε προβλήματα και να μαθαίνουμε - έχει συμβάλει σημαντικά στην επιβίωση του είδους από τον διαειδικό ανταγωνισμό (interspecific competition)[1]. Σε τελική ανάλυση, ο όρος homo sapiens - άνθρωπος ο σοφός - οφείλεται στη σημασία που έχει η νοημοσύνη στη ζωή μας.

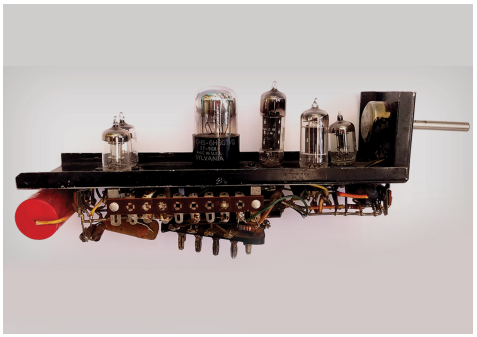
Το πρώτο ευρέως αναγνωρισμένο έργο προς την επίτευξη Τεχνητής Νοημοσύνης είναι αυτό της μαθηματικής μοντελοποίησης της λειτουργίας ενός νευρώνα από τους Warren McCulloch και Walter Pitts (1943) [20]. Αναλυτικότερα, βασιζόμενοι στην υπόθεση ότι η κατάσταση λειτουργίας ενός νευρώνα είναι δυαδική («all-or-none») αναπαρέστησαν κάθε νευρώνα ενός δικτύου ως μια πρόταση (proposition) της προτασιακής λογικής (propositional logic). Όπως περιγράφουν, η διέγερση (excitation) του μοντέλου ενός νευρώνα είναι ταυτόσημη με το να είναι η πρόταση του νευρώνα θετική, κάτι που εξαρτάται από την κατάσταση των γειτονικών νευρώνων. Όσο περισσότεροι, διεγερτικά διασυνδεδεμένοι (excitatory connected) προσυναπτικοί νευρώνες (presynaptic neurons) είναι ενεργοποιημένοι τόσο πιο πιθανή είναι η ενεργοποίηση του μετασυναπτικού νευρώνα (postsynaptic neuron). Μεταξύ άλλων, απέδειξαν ότι κάθε συνάρτηση που μπορεί να υπολογιστεί από μια μηχανή Turing μπορεί να υπολογιστεί και από ένα δίκτυο από διασυνδεδεμένους τεχνητούς νευρώνες ύστερα από την κατάλληλη παραμετροποίησή του. Το έργο των Warren McCulloch και Walter Pitts ήταν πρωτοπόρο για την εποχή του αφού έθεσε τις βάσεις όχι μόνο για την σημερινή Τεχνητή Νοημοσύνη αλλά και για την Υπολογιστική Νευρωεπιστήμη (Computational Neuroscience). Ωστόσο, οι συγγραφείς δεν παρουσίασαν κανέναν αλγόριθμο για την αλλαγή της τοπολογίας και των παραμέτρων του τεχνητού νευρωνικού δικτύου αν και φαίνεται πως αναγνώριζαν τη σημασία του για τη μάθηση.

Στο βιβλίο του [21] ο D. Hebb το 1949 επιδίωξε να ενώσει τις αποκλίνουσες θεωρίες της ψυχολογίας και της νευροεπιστήμης θέτοντας κοινές βάσεις για την ερμηνεία της ανθρώπινης συμπεριφοράς. Προηγούμενες θεωρίες απέφευγαν να δώσουν εξήγηση στις διεργασίες του εγκεφάλου ως μεσάζοντα μεταξύ του αισθητηριακού ερεθίσματος (sensory stimuli) και της πιθανής, καθυστερημένης απόκρισης. Αντίθετα, κατέφευγαν στη φιλοσοφία για την ανάλυση των χαρακτηριστικών της ανθρώπινης συμπεριφοράς όπως η προσοχή (attention), το ενδιαφέρον (interest) και η «προσδοκία» (expectancy). Ο Hebb όμως, εργάστηκε στο να αποδείξει ότι η ανθρώπινη συμπεριφορά μπορεί να γίνει κατανοητή υπό το πρίσμα της φυσιολογίας ("[expectancy] can be a physiologically intelligible process"). Φαινομενικά, ενώ το έργο του απασχολεί μόνο την επιστήμη της Νευροφυσιολογίας (Neurophysiology) συνεισέφερε σημαντικά στο κίνημα του διασυνδετισμού (Connectionism) - το κίνημα μελέτης των διανοητικών διεργασιών με τη χρήση

[1]Για την ακρίβεια, ενώ σύμφωνα με τη Δαρβινική θεώρηση η αφηρημένη νοημοσύνη μπορεί να προκύψει άμεσα από τη θεωρία της εξέλιξης των ειδών, νεότερες έρευνες το διαψεύδουν αφού το χαρακτηριστικό της αφηρημένης σκέψης ήταν αχρείαστο στην πραγματιστική παλαιολιθική εποχή. Στην προσπάθειά τους να ερμηνεύσουν την εμφάνιση νοημοσύνης στους ανθρώπους ορίζουν τον όρο «διανοητική βιοθέση» (cognitive niche) για να περιγράψουν όλα τα ζωολογικά ασυνήθιστα χαρακτηριστικά (zoologically unusual traits) που εμφανίζει ο άνθρωπος με τα κυριότερα να είναι η κοινωνικότητα και η λογική αιτίου - αποτελέσματος (cause-and-effect reasoning) [17,18]. Υποστηρίζουν λοιπόν, ότι η εμφάνιση της διανοητικής βιοθέσης αποτέλεσε τον καταλύτη για την εξέλιξη της ανθρώπινης νοημοσύνης [19]

τεχνητών νευρωνικών δικτύων. Για παράδειγμα, η θεώρηση της διαδικασίας μάθησης ως της επαναλαμβανόμενης, ταυτόχρονης πυροδότησης γειτονικών νευρώνων με αποτέλεσμα [την ενδυνάμωση των δεσμών και] τη διαμόρφωση νευρικών συστάδων (cell assemblies) είναι η λογική πίσω από πολλούς αλγορίθμους μάθησης τεχνητών νευρωνικών δικτύων. Επίσης, ένα στοιχείο που θα μας φανεί χρήσιμο στη συνέχεια είναι η παρατήρησή του ότι η κίνηση των ματιών δεν είναι τυχαία αλλά σχετίζεται με τη διαδικασία αντίληψής των θωρούμενων αντικειμένων. Τα παραπάνω δύο έργα έθεσαν το απαραίτητο θεωρητικό υπόβαθρο εμπνέοντας τους ερευνητές στην πειραματική υλοποίηση της Τεχνητής Νοημοσύνης.

Αρκετές ήταν οι απόπειρες δημιουργίας Τεχνητής Νοημοσύνης. Ο πρώτος υπολογιστής τεχνητών νευρωνικών δικτύων ονομάζονταν SNARC (Stochastic Neural Analog Reinforcement Calculator) και κατασκευάστηκε από τους Marvin Minsky και Dean Edmonds το 1950 [22]. Χρησιμοποιώντας 3000 λυχνίες κενού και έναν μηχανισμό αυτόματου πιλότου εξομοίωσαν τη λειτουργία 40 διασυνδεδεμένων νευρώνων. Χρησιμοποιώντας έναν απλό μηχανισμό επιβράβευσης τα «βάρη» του δικτύου - υπο τη μορφή ποτενσιόμετρων - προσαρμόζονταν στο πρόβλημα του λαβυρίνθου στο οποίο δοκιμάστηκε. Ακόμα ένα παράδειγμα τεχνητής νοημοσύνης μπορεί να θεωρηθεί το πρόγραμμα του Christopher Strachey στον υπολογιστή Manchester Mark 1 [23] που αργότερα θεωρήθηκε το πρώτο βιντεοπαιχνίδι. Ήταν ένα παιχνίδι ντάμας που πιθανώς χρησιμοποιούσε κάποιον μη πλήρη (incomplete) αλγόριθμο αναζήτησης στον χώρο των επιτρεπτών ακολουθιών κινήσεων (action sequences). Παρόλα αυτά, η δυνατότητά του να ανταγωνίζεται αποτελεσματικά τον άνθρωπο οδήγησε στη θεώρησή του ως Τεχνητή Νοημοσύνη. Άλλωστε, η σύγχυση για το θέμα ήταν ακόμα μεγαλύτερη την εποχή εκείνη.



Σχήμα 1.1: Ένας από τους 40 νευρώνες του SNARC. Χορηγία της κα. Margaret Minsky [22]

Ο αρχικός ενθουσιασμός για το επιστημονικό πεδίο τράβηξε τα βλέμματα πολλών επιφανών ερευνητών της εποχής. Ο πατέρας της Επιστήμης των Υπολογιστών (Computer Science) και της Τεχνητής Νοημοσύνης, Alan Turing, στην προσπάθειά του να διασαφηνίσει το ερώτημα του «αν οι μηχανές σκέφτονται» επινόησε το επονομαζόμενο Turing Test. Σύμφωνα με τη δημοσίευσή του [24] το 1950, πρόκειται για μια δοκιμασία που εμπλέκει έναν «ανακριτή» ο οποίος διατυπώνει γραπτές ερωτήσεις σε δύο «μάρτυρες»: έναν άνθρωπο και μια μηχανή. Η δοκιμασία θεωρείται επιτυχής όταν ο ανακριτής - χωρίς να έχει οπτική επαφή με τους «μάρτυρες» - δεν μπορεί να ξεχωρίσει τον άνθρωπο από τη μηχανή. Στην ίδια δημοσίευση τόνισε τη σημασία της μάθησης για την ανάπτυξη της Τεχνητής Νοημοσύνης. Υποστήριζε ότι αντί να επιχειρείται η εξονυχιστική συγγραφή ενός προγράμματος που θα μοιάζει με τη σκέψη ενός ώριμου ενήλικα (με αμέτρητες προγραμματισμένες εντολές) είναι προτιμότερη και ταχύτερη η προσομοίωση της νοημοσύνης ενός παιδιού που μέσα από μηχανισμούς εκπαίδευσης, έμμεσα, αποκτά ώριμη σκέψη. Επίσης, εναπόθεσε τους σπόρους για τους γενετικούς αλγορίθμους, ενώ σε επόμενη δημοσίευσή του [25] μελέτησε τους τρόπους με τους οποίους μια μηχανή με νοημοσύνη θα μπορούσε να λειτουργεί. Ένα ακόμα δημοφιλές όνομα, ο John von Neumann συνεισέφερε στον χώρο ανα-

πτύσσοντας τα «τεχνητά αυτόματα» (artificial automata) [26] ενώ η συμβολή του πιθανώς θα ήταν ακόμα μεγαλύτερη αν προλάβαινε να ολοκληρώσει το βιβλίο του «Ο Υπολογιστής και το Μυαλό» (The Computer and the Brain). Μια τελευταία απόδειξη της προσοχής που έλαβε η Τεχνητή Νοημοσύνη διαφαίνεται στα δέκα μέλη σχετικού σεμιναρίου (workshop) που έλαβε χώρα το καλοκαίρι του 1955 στο Dartmouth College [27]. Ίσως το ποιο σημαντικό πόρισμα αυτής της συνάθροισης ήταν η ανάπτυξη του Logic Theorist από τους Allen Newell και Herbert Simon, ενός συστήματος για την απόδειξη θεωρημάτων στα μαθηματικά.

Η δεκαετία που ακολούθησε χαρακτηρίζεται από έντονη αισιοδοξία για τις δυνατότητες της Τεχνητής Νοημοσύνης: γενναιόδωρες επενδύσεις σε ερευνητικά προγράμματα ενθάρρυναν τη δημιουργία ποικίλων προγραμμάτων που υποδείκνυαν κάποια μορφή νοημοσύνης. Πιο συγκεκριμένα, αν εξαιρέσουμε την εξέχουσα δουλειά του Arthur Samuel όπου ανέπτυξε ένα παιχνίδι ντάμας χρησιμοποιώντας ενισχυτική μάθηση (Reinforcement Learning), οι περισσότερες προσπάθειες εστίασαν στον χώρο της μίμησης της ανθρώπινης συλλογιστικής (reasoning). Η ιδέα είναι ότι με μια τυπική γλώσσα (formal language) για την αναπαράσταση της γνώσης (knowledge representation) στον υπολογιστή μαζί με την εφαρμογή απλών κανόνων λογικής συμπερασματολογίας (logical inference) σε αυτή καθιστούν δυνατή την εξαγωγή πορισμάτων. Καθοριστικό ρόλο σε αυτή τη «σχολή» είχε η - αναχρονιστική - υπόθεση ότι η νοημοσύνη είναι άρρηκτα συνδεδεμένη με τη δυνατότητα χειρισμού συμβόλων οργανωμένων σε δομές δεδομένων (physical symbol system hypothesis).

Σε αυτήν την κατεύθυνση, δηλαδή της συμβολικής Τεχνητής Νοημοσύνης (symbolic Artificial Inteligence), εργάστηκαν αρκετοί επιστήμονες της εποχής. Για παράδειγμα, οι δύο ερευνητές πίσω από το Logic Theorist επινόησαν το General Problem Solver. Πρόκειται ουσιαστικά για έναν αλγόριθμο ο οποίος δέχεται σαν είσοδο μια τυποποιημένη περιγραφή του προβλήματος και το επιλύει ακολουθώντας μια στρατηγική ευρετικής αναζήτησης (heuristic search) της λύσης [27]. Στο ίδιο μήκος κύματος εργάστηκε και ο John McCarthy. Εκτός από το ότι ανέπτυξε τη γλώσσα προγραμματισμού Lisp, ειδικά φτιαγμένη για εφαρμογές Τεχνητής Νοημοσύνης, εξέλιξε το πεδίο με το δημοσίευμά του «Programs with Common Sense» (1958) στο οποίο περιέγραφε το Advice Taker. Αυτό ήταν ένα πρόγραμμα για την επίλυση προβλημάτων μέσω της εφαρμογής «κοινής λογικής» σε προτάσεις διατυπωμένες σε τυπική γλώσσα. Για παράδειγμα, δοθέντος μιας σειράς υποθέσεων σχετικά με το περιβάλλον του προβλήματος διατυπωμένων σε τυπική γλώσσα (π.χ. «Εγώ είμαι στο γραφείο.», «Θέλω να πάω αεροδρόμιο.» κτλ.) ο αλγόριθμος εξήγαγε ένα πλάνο με τα βήματα που πρέπει να ακολουθηθούν για τη μετάβαση στο αεροδρόμιο [28]. Το έργο του συνέχισε ο J. A. Robinson όπου και επινόησε μια πλήρη μέθοδο επίλυσης (complete resolution method) για προβλήματα εκφρασμένα σε λογική πρώτης τάξης. Οι εφαρμογές του ήταν πολλές: από συστήματα μαθηματικού λογισμού (James Slagle's SAINT program [29] και Daniel Bobrow's STUDENT program) μέχρι εφαρμογές ερωταπαντήσεων (Cordell Green's question-answering and planning systems) και ρομποτικής (Shakey Robotics Project). Τέλος, πολλές εφαρμογές της Συμβολικής Τεχνητής Νοημοσύνης αναπτύχθηκαν για το «παιχνίδι» blocks world: ένα περιβάλλον αποτελούμενο από τουβλάκια που αποσκοπούσε στον πειραματισμό συστημάτων αναπαράστασης γνώσης και συλλογιστικής [30].

Την ίδια εποχή, ειδικά για τον χώρο των νευρωνικών δικτύων υπήρξε σημαντική πρόοδος με τα έργα Perceptron και ADALINE. Το πρώτο συγγράφηκε από τον F. Rosenblatt το 1958

και αποτέλεσε το πρώτο μοντέλο νευρωνικού δικτύου με δυνατότητα επιβλεπόμενης μάθησης (supervised learning). Πιο συγκεκριμένα, το Perceptron είναι ένας ταξινομητής γραμμικά διαχωρίσιμων προτύπων με ένα μεμονωμένο τεχνητό νευρώνα του οποίου οι ελεύθεροι παράμετροι - τα προσυναπτικά βάρη (presynaptic weights) και η πόλωση (bias) - προσαρμόζονται στα δεδομένα εισόδου σύμφωνα με έναν αλγόριθμο μάθησης (perceptron rule) [31]. Σε μια εκτενή παρουσίαση του έργου [32], ο Rosenblatt βασίστηκε στη θεωρία του D. Hebb και την επέκτεινε προτείνοντας ένα μοντέλο (το Perceptron) με το οποίο η συμπεριφορά (καμπύλη εκμάθησης) μπορεί να προβλεφθεί από τη νευροφυσιολογία του συστήματος (τα συναπτικά βάρη). Παρόμοιο ήταν και το έργο ADALINE (Adaptive Linear Neuron) του B. Widrow στο οποίο περιγράφεται και πάλι ένας αλγόριθμος μάθησης για την προσαρμογή των βαρών. Αυτή τη φορά όμως, είναι ο (γνωστός) αλγόριθμος στοχαστικής καθόδου κλίσης (stochastic gradient descent) που χρησιμοποιείται ακόμα και σήμερα στον αλγόριθμο γραμμικής παλινδρόμησης (linear regression). Μια ακόμα αξιοσημείωτη διαφορά έγκειται στη συνάρτηση ενεργοποίησης όπου ενώ στο πρώτο έργο είναι η βηματική συνάρτηση (step function), στο δεύτερο έργο είναι η γραμμική συνάρτηση (linear activation function - identity function) που καθιστά τον αλγόριθμο κατάλληλο για την πρόβλεψη πραγματικών τιμών [33, 34]. Συνεπώς, ενώ το πρώτο έργο αποτελεί όπως προαναφέρθηκε έναν αλγόριθμο ταξινόμησης, το δεύτερο έργο ανήκει στην κατηγορία αλγορίθμων γραμμικής παλινδρόμησης. Βέβαια, μολονότι και τα δύο έργα είχαν καθοριστική σημασία στην εξέλιξη της Τεχνητής Νοημοσύνης με τη μορφή που τη συναντάμε σήμερα, όπως θα δούμε στη συνέχεια, η έντονη κριτική που ακολούθησε τα επισκίασε για μια ολόκληρη δεκαετία.

Γύρω στο 1970, το επιστημονικό πεδίο της Τεχνητής Νοημοσύνης διήλθε μια εποχή «χειμώνα» (AI winter). Η χρηματοδότηση ερευνητικών προγραμμάτων πάγωσε και έτσι το ενδιαφέρον στράφηκε αλλού. Κοιτώντας πίσω, είναι εύκολο να εντοπίσει κανείς τα αίτια αυτού του «χειμώνα». Καταρχάς, ένας λόγος ήταν (και είναι) το ελλιπές επιστημονικό υπόβαθρο σε ό,τι αφορά την ανθρώπινη νοημοσύνη [35]. Αναμενόμενο, αφού για μια επιτυχημένη μίμηση της ανθρώπινης νοημοσύνης απαιτείται πρώτα η κατανόησή της. Βέβαια, ίσως ο σημαντικότερος λόγος ήταν η απογοήτευση που προκλήθηκε όταν φιλόδοξες υποσχέσεις για τις δυνατότητες της Τεχνητής Νοημοσύνης στο εγγύς μέλλον δεν μπόρεσαν να ικανοποιηθούν. Όπως αποδείχθηκε, η μετάβαση της Τεχνητής Νοημοσύνης από εφαρμογές παιδικών κόσμων όπως το blocks world σε πραγματικά προβλήματα δεν ήταν απλώς ζήτημα γραμμικής αύξησης της υπολογιστικής δύναμης. Για παράδειγμα, στην περίπτωση της συμβολικής Τεχνητής Νοημοσύνης, με την ωρίμανσή της θεωρίας πολυπλοκότητας (computational complexity) αναδείχθηκε το θέμα της συνδυαστικής έκρηξης (combinatorial explosion) αποκαλύπτοντας έτσι τη δυσεπίλυτη (intractable) φύση πολλών προβλημάτων του αληθινού κόσμου. Αντίστοιχα εμπόδια έκαναν την εμφάνισή τους στον χώρο των Νευρωνικών Δικτύων. Το σημαντικότερο ήταν αυτό της αδυναμίας ενός μεμονωμένου Perceptron με δύο εισόδους να αναπαραστήσει πολλές συναρτήσεις όπως τη συνάρτηση XOR [27] - περιγράφηκε από το βιβλίο Perceptrons των M. Minsky και S. Papert το 1969. Συνολικά, αν και ορισμένα από τα ανωτέρω αίτια είναι ακόμα σε ισχύ, το ενδιαφέρον σύντομα αναζωπυρώθηκε.

Παρά τις ανωτέρω αδυναμίες της Τεχνητής Νοημοσύνης την εποχή εκείνη, αυτό δεν εμπόδισε την αξιοποίησή της σε εξειδικευμένες εφαρμογές. Πιο συγκεκριμένα, τις δεκαετίες του 1970 και 1980 αναπτύχθηκαν τα «έμπειρα συστήματα» (expert systems). Πρόκειται για προγράμματα που εφαρμόζουν κανόνες συλλογιστικής σε εξειδικευμένη βάση γνώσης (domain-specific knowledge

base) μιμούμενοι τη διαδικασία λήψης αποφάσεων ενός εμπειρογνώμονα. Η εξειδικευμένη βάση γνώσης περιόριζε σημαντικά τον χώρο αναζήτησης λύσεων (search space) έτσι ώστε η συνδυαστική έκρηξη να μην αποτελεί πρόβλημα. Αυτό, επέτρεψε να αναπτυχθούν πολλά έμπειρα συστήματα για εμπορική χρήση - κυρίως στον χώρο της υγείας - αποδεικνύοντας για πρώτη φορά έμπρακτα τα οφέλη της Τεχνητής Νοημοσύνης. Ενδεικτικά, δύο δημοφιλή παραδείγματα είναι το MYCIN και το INTERNIST. Το MYCIN αποσκοπούσε στη διάγνωση βακτηριακών μολύνσεων μέσω ενός αλγορίθμου συλλογιστικής που μοντελοποιούσε την αβεβαιότητα των λογικών υποθέσεων και συμπερασμάτων. Το INTERNIST από την άλλη βοηθούσε στη διάγνωση ασθενειών μετά από την περιγραφή των εκδηλούμενων συμπτωμάτων [36]. Αν και τα έμπειρα συστήματα ανανέωσαν το ενδιαφέρον για την Τεχνητή Νοημοσύνη, αυτό δε διήρκησε πολύ λόγω προβλημάτων που εμφάνιζαν με το κυριότερο να είναι η έλλειψη «κοινής λογικής» [37].

Η έρευνα στον χώρο της Τεχνητής Νοημοσύνης αποκαταστάθηκε στα συνήθη υψηλά επίπεδα σε σύντομο χρονικό διάστημα. Αυτό μπορεί σε μεγάλο βαθμό να αποδοθεί σε ένα μεμονομένο έργο· το Parallel Distributed Processing που συγγράφηκε από τους David E. Rumelhart et al. και δημοσιεύτηκε το 1986 [38]. Οι συγγραφείς, μεταξύ των οποίων και ο Geoffrey Hinton εμπνεόμενοι από τα παλαιότερα έργα πάνω στη γνωστική νευροεπιστήμη (cognitive neuroscience) έστρεψαν την έρευνα του χώρου από πειραματικές «αλχημείες» (π.χ. αυτή της συμβολικής λογικής) σε μια πιο επίσημη, φορμαλιστική διαδικασία βασιζόμενη λιγότερο στη φιλοσοφία και περισσότερο στις θετικές επιστήμες[2]. Με αυτόν τον τρόπο, η σχολαστική συγγραφή αναρίθμητων κανόνων προτασιακής λογικής για τη δημιουργία βάσεων γνώσης εγκαταλείφθηκε, μαζί της και η θεωρία της συμβολικής Τεχνητής Νοημοσύνης. Άλλωστε, η τεχνολογία των έμπειρων συστημάτων είχε παρακμάσει αφού όπως φάνηκε από την απουσία «κοινής λογικής» σε αυτά, ήταν εξαιρετικά περιοριστική η χρήση προτασιακής λογικής για την περιγραφή του πραγματικού, αβέβαιου κόσμου [27, 39].

Τη δεκαετία του 1980 τη θέση της συμβολικής Τεχνητής Νοημοσύνης πήρε το κίνημα του κονεκτιβισμού (connectionist movement). Όπως θα δούμε, αυτό συνέβαλλε καθοριστικά στη διαμόρφωση του σημερινού κλάδου των νευρωνικών δικτύων. Τυπικά, ο κονεκτιβισμός είναι το κίνημα της γνωστικής επιστήμης (cognitive science) που επιχειρεί να εξηγήσει τις διανοητικές διεργασίες με τη χρήση ενός δικτύου με βάρη (weighted network) που διασυνδέει απλές μονάδες επεξεργασίες [40]. Σε αυτήν τη θεωρία καταπιάστηκαν και οι συγγραφείς του έργου Parallel Distributed Processing [38] οι οποίοι εδραίωσαν τις ιδέες που σήμερα θεμελιώνουν τη θεωρία των νευρωνικών δικτύων. Η πρώτη σημαντική ιδέα που περιγράφεται λεπτομερώς στο έργο είναι η κατανεμημένη αναπαράσταση (distributed representation) σύμφωνα με την οποία κάθε είσοδος στο σύστημα αναπαρίσταται από πολλά χαρακτηριστικά κατανεμημένα στο δίκτυο και ανάποδα, δηλαδή κάθε μεμονωμένο χαρακτηριστικό μπορεί να αποτελεί μέρος της περιγραφής πολλών, ετερογενών εισόδων. Μια ακόμα σημαντική ιδέα είναι αυτή της μηχανικής μάθησης (machine learning) με την οποία η επίδοση ενός συστήματος βελτιώνεται (μαθαίνει) από την εμπειρία. Για τον σκοπό αυτό, παρουσιάζουν έναν επαναστατικό αλγόριθμο ο οποίος αυτοματοποιεί τη διαδικασία μηχανικής μάθησης στα νευρωνικά δίκτυα. Πρόκειται για τον αλγόριθμο ανάστροφης διάδοσης σφάλματος (back propagation) ο οποίος, παραδόξως, ενώ είχε αναπτυχθεί περίπου το 1960 γνώρισε ευρεία χρήση από το 1980 και μετά. Συνεπώς, η σημασία του κονεκτιβισμού είναι

[2]Στη βιβλιογραφία αυτό το γεγονός αναφέρεται σαν «η νίκη των καθαρών» (victory of the neats) [27].

καθοριστική αφού αναβίωσε τις ιδέες της γνωστικής επιστήμης επανατοποθετώντας κατά αυτόν τον τρόπο τον κλάδο της Τεχνητής Νοημοσύνης σε μια πιο επιστημονική τροχιά.

Η στροφή σε μια πιο επιστημονική προσέγγιση του κλάδου της Τεχνητής Νοημοσύνης το 1980 δε συνέβαλλε μόνο στην ανάπτυξη του κλάδου των νευρωνικών δικτύων. Για παράδειγμα, ο κλάδος της επεξεργασίας φυσικής γλώσσας επωφελήθηκε σημαντικά από την επιτυχημένη μοντελοποίηση ακολουθιών με τη χρήση κρυφών Μαρκοβιανών μοντέλων (hidden Markov models) και αργότερα, το 1997 με μονάδες μακράς-βραχέας μνήμης (Long-Short Term Memory block - LSTM). Επίσης, ο χώρος της όρασης υπολογιστών επωφελήθηκε από τη σύγκλιση της Τεχνητής Νοημοσύνης με τις θετικές επιστήμες. Την πρόοδο μαρτυρά η εμφάνιση των πρώτων εφαρμογών οπτικής αναγνώρισης χαρακτήρων (optical character recognition) τη δεκαετία του 1980 και ύστερα, των τυποποιημένων βάσεων δεδομένων για ανάπτυξη και μέτρηση απόδοσης οπτικών συστημάτων αναγνώρισης μοτίβων (π.χ. MNIST [41]). Επίσης, σημαντικά αναπτύχθηκε ο χώρος της ταξινόμησης προτύπων με τη δημιουργία ή βελτίωση αρκετών μοντέλων όπως οι μηχανές διανυσματικής υποστήριξης με μέθοδο πυρήνα (Support Vector Machines with kernel trick) και τα δίκτυα ακτινικής βάσης (Radial Basis Networks). Ακόμα και ο χώρος της συλλογιστικής για τον οποίο κάναμε λόγο σε προηγούμενες παραγράφους εμπλουτίστηκε με μια σχολαστική και αποδοτική - αυτή τη φορά - μοντελοποίηση αβεβαιότητας της γνώσης μέσω της ανάπτυξης Μπεϋζιανών δικτύων (Bayesian networks). Τέλος, ο χώρος της στατιστικής γνώρισε πρόοδο αφού στις παραδοσιακές τεχνικές συμπερασματολογίας προστέθηκαν αυτές βασιζόμενες σε μηχανική μάθηση.

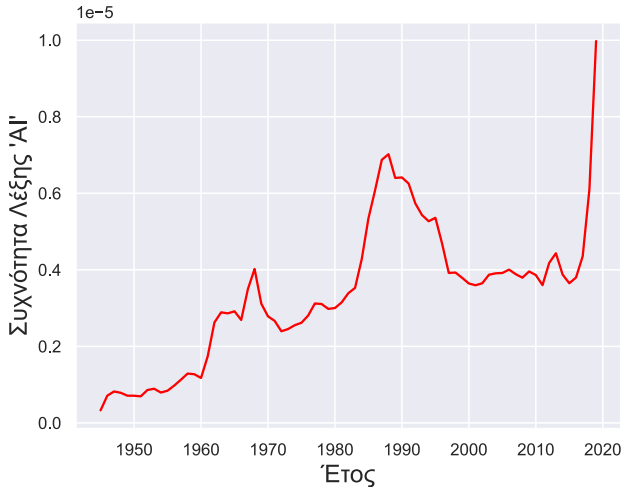
Σχήμα 1.2: Η βάση δεδομένων ImageNet.

Προς τα τέλη της δεκαετίας του 1990 και τις αρχές του 2000 η έρευνα είχε εστιάσει σε κλάδους της Τεχνητής Νοημοσύνης που δε σχετίζονταν με τα νευρωνικά δίκτυα. Η κατάσταση αυτή όμως σύντομα αναστράφηκε. Αρχικά, η μεγάλη υπολογιστική ισχύ που απαιτούσαν αλγόριθμοι εκπαίδευσης βαθιών νευρωνικών δικτύων δε διευκόλυναν τον πειραματισμό [33]. Παρόλα αυτά, χάρη σε τρεις ερευνητές (Geoffrey Hinton, Yann LeCun και Yoshua Bengio) που χρηματοδοτούνταν από το Καναδικό Ινστιτούτο για προηγμένη έρευνα (CIFAR) η ενασχόληση με τα νευρωνικά δίκτυα κρατήθηκε ζωντανή οδηγώντας τελικά σε αξιοσημείωτη πρόοδο. Το πρώτο ορόσημο των βαθιών νευρωνικών δικτύων ήταν το 2006 όπου οι Hinton et al. [42] απέδειξαν ότι ένα είδος βαθέος νευρωνικού δικτύου - τα βαθιά δίκτυα πίστης (deep belief networks) - μπορούν να εκπαιδευτούν αποδοτικά και γρήγορα μέσω ενός άπληστου (greedy) αλγορίθμου. Το δεύτερο ορόσημο που απέδειξε τις προοπτικές των βαθιών νευρωνικών δικτύων ήταν η επιτυχία αυτής της τεχνολογίας σε διαγωνισμούς ταξινόμησης εικόνων της βάσης ImageNet το 2010 και το 2012. Στη δημοσίευσή τους ImageNet Classification with Deep Convolutional Neural Networks, οι A. Krizhevsky et al. [43] περιγράφουν μια νέα αρχιτεκτονική νευρωνικών δικτύων (βαθιά συνελικτικά δίκτυα) αλλά και πρωτοπόρες μεθόδους για αποδοτική εκπαίδευση (dropout, ReLU activation function κλπ.). Έκτοτε, το ενδιαφέρον

απογειώθηκε και σε συνδυασμό με την αυξημένη διαθεσιμότητα δεδομένων, εξειδικευμένων συσκευών για παράλληλη επεξεργασία και αποδοτικών αλγορίθμων εκπαίδευσης δημιούργησαν το πιο πρόσφορο έδαφος για την άνθιση της Τεχνητής Νοημοσύνης.

Σήμερα, η εξέλιξη της Τεχνητής Νοημοσύνης και δη των νευρωνικών δικτύων είναι ραγδαία. Για παράδειγμα, με τη χρήση μοντέλων νευρωνικών δικτύων βασισμένων στον μηχανισμό προσοχής (attention-based neural networks) όπως τα λεγόμενα transformers, σημειώθηκε αξιοσημείωτη πρόοδος τόσο στον κλάδο της επεξεργασίας φυσικής γλώσσας (natural language processing) (βλέπε GPT -3) όσο και στον χώρο της όρασης υπολογιστών (βλέπε Vision Transformer και CoAtNet). Αναλυτικότερα για το Generative Pre-trained Transformer 3 - GPT 3, πρόκειται για το γλωσσικό μοντέλο που μπορεί να δημιουργήσει κείμενο σε φυσική γλώσσα ακόμα και να διατηρήσει για εύλογο χρόνο συζήτηση με έναν άνθρωπο. Ακόμα, σημαντική πρόοδος παρατηρείται στη μέθοδο μάθησης με αυτο-επίβλεψη (self-supervision) επιτρέποντας έτσι την εκπαίδευση δικτύων χωρίς να απαιτείται η σχολαστική και χρονοβόρα ανάθεση ετικετών στα δεδομένα εκπαίδευσης. Τέλος, μεταξύ άλλων, σπουδαία εξέλιξη υπήρξε πρόσφατα στις εφαρμογές όπου δοθέντων μερικών εικόνων από ένα αντικείμενο συνθέτουν εικόνες αυτού από νέες γωνίες θέασης (Novel View Synthesis). Ενδεικτικά, πρωτοπόρα έργα στον χώρο αυτό είναι το NeRF και το GIRAFFE [44].

Κλείνοντας το ιστορικό αυτό σημείωμα, δεν μπορώ παρά να αντικρίσω με δέος το μέλλον που επιφέρει η τεχνολογία της Τεχνητής Νοημοσύνης. Ανύπαρκτη πριν έναν αιώνα, σήμερα είναι μέρος της καθημερινότητά μας με τεράστια ορμή που δε φαίνεται να κατευνάζει. Αν μέσα σε μερικές δεκαετίες έχει τόσες δυνατότητες, στο εγγύς μέλλον η δύναμή της θα είναι τεράστια. Δύναμη που θα δημιουργήσει μια επίγεια ουτοπία ή θα αποτελέσει ένα ακόμα τουβλάκι στην οικοδόμηση της κοινωνίας του ρίσκου[3]...ο χρόνος θα δείξει.



Σχήμα 1.3: Γραφική παράσταση της συχνότητας του όρου AI σε βιβλία γραμμένα στην αγγλική γλώσσα ανά έτος (από 1945 μέχρι και 2019). Είναι εμφανείς οι τρεις περίοδοι ακμής του κλάδου. *Παράχθηκε από το Google Ngram Viewer.*

[3]Ο όρος «κοινωνία του ρίσκου» (risk society) είναι δανεισμένος από το ομώνυμο βιβλίο του Ulrich Beck [45] όπου περιγράφεται το χαρακτηριστικό των μοντέρνων κοινωνιών να οργανώνονται γύρω από νέες μορφές, απαρατήρητου ρίσκου (όπως αυτό της κλιματικής αλλαγής).

1.3 Κίνητρο

Σύμφωνα με τον Andrew Ng, έναν πολύ επιφανή επιστήμονα στον χώρο της Τεχνητής Νοημοσύνης, έχουμε φτάσει στο σημείο να υπάρχουν αξιόπιστα, «θεμελιώδη» (foundational) συστήματα μηχανικής μάθησης γενικού σκοπού στον χώρο της επεξεργασίας φυσικής γλώσσας (natural language processing). Κάτι τέτοιο όμως δεν ισχύει ακόμα για τον χώρο της όρασης υπολογιστών όπου τα μοντέλα είναι πολύ λιγότερο ευέλικτα και περιορίζονται σε συγκεκριμένες εφαρμογές [46]. Στην προσπάθεια κατασκευής θεμελιωδών μοντέλων, δημιουργούνται όλο και πιο βαθιές αρχιτεκτονικές νευρωνικών δικτύων με εκατοντάδες δισεκατομμύρια παραμέτρους. Για την εκπαίδευση αυτών των αρχιτεκτονικών δαπανιούνται τεράστια ποσά ενεργειακών και χρηματικών πόρων γεγονός που καθιστά την παρούσα τάση μη βιώσιμη (unsustainable).

Στην προσπάθεια εύρεσης λύσης στο πρόβλημα αυτό, καλούμαστε να εντοπίσουμε παθογένειες των αρχιτεκτονικών νευρωνικών δικτύων που χρησιμοποιούνται σε εφαρμογές όρασης υπολογιστών και να προτείνουμε βελτιώσεις. Μια από αυτές τις βελτιώσεις συναντάται στη βιβλιογραφία υπό το όνομα «Νευρωνικά Δίκτυα με Κάψουλες» και σκοπό έχει την αποδοτικότερη αναγνώριση αντικειμένων όταν αυτά παρουσιάζονται υπό διαφορετικές οπτικές γωνίες. Αφορμώμενοι από τα σχετικά έργα στην τεχνολογία αυτή και πάντα λαμβάνοντας υπόψη το υπολογιστικό κόστος, εμβαθύνουμε σε αυτή μέσα από εκτενή πειράματα και προτείνουμε κλιμακώσιμες παραλλαγές επιδιώκοντας τη βελτίωσή της.

1.4 Συνεισφορά Εργασίας

Μέσα από την ολιστική μελέτη της τεχνολογίας των νευρωνικών δικτύων και δη αυτών που χρησιμοποιούν κάψουλες στο περιβάλλον επιβλεπόμενης μάθησης, μπορούμε να ισχυριστούμε ότι το παρόν έργο συνεισφέρει στον εν λόγω χώρο. Αφενός, εμβαθύνει στις προϋπάρχουσες τεχνολογίες με μοναδικά - στη βιβλιογραφία - πειράματα και αφετέρου προτείνει νέες τεχνολογίες που - όπως αποδεικνύεται - υπερβαίνουν ορισμένες από τις αδυναμίες που εντοπίζονται σε παλαιότερες βιβλιογραφικές αναφορές.

Πιο συγκεκριμένα, η συνεισφορά της παρούσας εργασίας μπορεί να συνοψιστεί στα εξής (παρατίθενται με τη σειρά που αναπτύσσονται στον τόμο):

- Αναλυτική επεξήγηση του χώρου των νευρωνικών δικτύων με πολλές αναφορές σε σύγχρονες τεχνικές που χρησιμοποιούνται από τα συστήματα τεχνητής νοημοσύνης τελευταίας τεχνολογίας.
- Εκτενής βιβλιογραφική μελέτη μιας πολλά υποσχόμενης υποκατηγορίας των νευρωνικών δικτύων, αυτής των νευρωνικών δικτύων με κάψουλες (capsule networks) και εντοπισμός των αδυναμιών της.
- Ενδελεχής πειραματισμός με τις βασικές υλοποιήσεις των νευρωνικών δικτύων με κάψουλες αλλά και παραλλαγές αυτών (μέθοδοι 1,2 και 4) που διαφωτίζουν την εσωτερική λειτουργία τους και αποδεικνύουν (ή καταρρίπτουν) θεμελιακές υποθέσεις αυτών.
- Παρουσίαση ενός πρωτότυπου αλγορίθμου δρομολόγησης νευρωνικών δικτύων με κάψουλες που δανείζεται στοιχεία από μια δημοφιλή τεχνοτροπία (αυτή του μηχανισμού προσοχής).

Πρόκειται για μια γρήγορη και κλιμακώσιμη τεχνολογία που επιτυγχάνει αξιοσημείωτες επιδόσεις χρησιμοποιώντας πολύ λιγότερους πόρους.

1.5 Οργάνωση του Τόμου

Η παρούσα διπλωματική εργασία έχει οργανωθεί με αυξανόμενο βαθμό εξειδίκευσής από κεφάλαιο σε κεφάλαιο. Στο πρώτο, εισαγωγικό κεφάλαιο, γίνεται μια σύντομη επισκόπηση του χώρου της τεχνητής νοημοσύνης αλλά και της ιστορίας της. Απευθύνεται κυρίως στον αναγνώστη που δεν έχει μελετήσει σφαιρικά τον επιστημονικό κλάδο με τον οποίο καταπιάνεται το παρόν έργο.

Στο δεύτερο κεφάλαιο γίνεται μια εκτενής επεξήγηση των εννοιών που χρησιμοποιούνται σε όλη την έκταση του τόμου. Μάλιστα, έχει δοθεί τόση έμφαση στην ανάπτυξη της θεωρίας που μια τόσο εμπεριστατωμένη μελέτη των βασικών εννοιών είναι δυσεύρετη, ακόμα και αν η αναζήτηση περιλαμβάνει και ξενόγλωσσα κείμενα. Η περιγραφή συνοδεύεται από σχήματα τα οποία έχουν σχεδιαστεί ειδικά για τον σκοπό αυτό σε επαγγελματικά προγράμματα διανυσματικών γραφικών (vector graphics). Διατίθενται ελεύθερα, τόσο στην Ελληνική όσο και στην Αγγλική γλώσσα στην ιστοσελίδα της διπλωματικής.

Στο τρίτο κεφάλαιο πραγματοποιείται η βιβλιογραφική επισκόπηση. Αυτή δεν περιορίζεται μόνο στον χώρο των νευρωνικών δικτύων με κάψουλες αλλά άπτεται και γενικότερων θεμάτων σχετικά με τη δυνατότητα των νευρωνικών δικτύων για αποδοτική αναγνώριση αντικειμένων υπό διαφορετικές οπτικές γωνίες. Μέσω αυτής της μελέτης εντοπίζονται οι κατευθύνσεις που είναι ωφέλιμο να στραφεί το πρακτικό μέρος της εργασίας.

Στο τέταρτο κεφάλαιο παρουσιάζονται οι τέσσερεις μέθοδοι με τις οποίες πειραματιζόμαστε. Η πρώτη μέθοδος περιλαμβάνει ένα δημοφιλή αλγόριθμο στον χώρο των νευρωνικών δικτύων με κάψουλες και τρεις δικές μας παραλλαγές που σκοπό έχουν να εξετάσουν ορισμένες ιδιότητες της τεχνολογίας. Η δεύτερη μέθοδος αποτελεί μια διασκευή του έργου [47] της οποίας η υλοποίηση βασίζεται σε μεγάλο βαθμό στο [48]. Χρησιμοποιείται για λόγους σύγκρισης με τις υπόλοιπες μεθόδους. Η τρίτη μέθοδος, εμπνεόμενη από το [49], προτείνει ένα καινοτόμο και γρήγορο μοντέλο νευρωνικού δικτύου με κάψουλες. Τέλος, η τέταρτη μας μέθοδος συνδυάζει την τεχνολογία με έναν αλγόριθμο εύρεσης συστάδων με σκοπό να δοκιμάσει την επίδραση διαφόρων υποθέσεων στην επίδοση του συστήματος μηχανικής μάθησης.

Στο πέμπτο κεφάλαιο καταγράφονται τα αποτελέσματα των μεθόδων μας αλλά και οι σχετικές παρατηρήσεις. Επίσης, συγκρίνονται οι μέθοδοι τόσο μεταξύ τους όσο και με άλλες σχετικές μεθόδους που απαντώνται στη βιβλιογραφία. Τέλος, αναφέρονται χρήσιμα συμπεράσματα από πειράματα που διευκολύνουν τη βαθύτερη κατανόηση του τρόπου λειτουργίας της τεχνολογίας με την οποία καταπιανόμαστε.

Στο τελευταίο κεφάλαιο γίνεται μια ανακεφαλαίωση και αναγράφονται τα συμπεράσματα. Επίσης, γίνεται αναφορά σε πιθανές μελλοντικές κατευθύνσεις στις οποίες θα μπορούσε να στραφεί η έρευνα.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

Στο παρόν κεφάλαιο θα οικοδομήσουμε την απαραίτητη γνώση στην οποία βασίζεται η έρευνα των επόμενων ενοτήτων. Αρχικά, θα παρουσιαστούν συνοπτικά τα τεχνητά νευρωνικά δίκτυα [1] υπό μια μαθηματική σκοπιά. Έπειτα, θα αναλυθούν τα νευρωνικά δίκτυα με κάψουλες (capsule networks) τα οποία και αποτελούν το κύριο θέμα της εργασίας. Τέλος, θα γίνει αναφορά σε νέες τεχνικές και αλγορίθμους που χρησιμοποιήθηκαν στο παρόν έργο ώστε η μετέπειτα εισαγωγή των μεθόδων μας για την εξέλιξη των νευρωνικών δικτύων με κάψουλες να είναι περισσότερο ομαλή και κατανοητή.

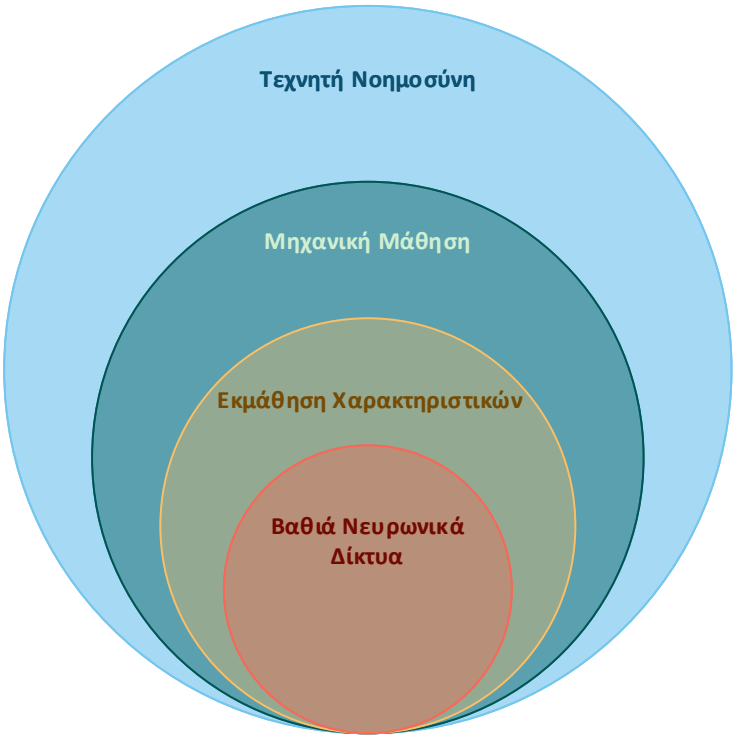
2.1 Τεχνητά Νευρωνικά Δίκτυα

Τα σημερινά τεχνητά νευρωνικά δίκτυα, όπως είναι αναμενόμενο, απέχουν σημαντικά από το πρώτο μοντέλο των Warren McCulloch και Walter Pitts [20] που συζητήσαμε στην ενότητα 1.2. Με την ωρίμανση της τεχνολογίας, αυτή ανεξαρτητοποιήθηκε από την (υπολογιστική) νευροεπιστήμη και εντάχθηκε στην Τεχνητή Νοημοσύνη υπό μια ιεραρχική δομή. Κρίνεται λοιπόν σκόπιμο να παρουσιάσουμε αυτήν την ιεραρχική δομή οργάνωσης της Τεχνητής Νοημοσύνης και μετέπειτα να αναφερθούμε στα επιμέρους στοιχεία της.

Όπως βλέπουμε στο σχήμα 2.1 τα νευρωνικά δίκτυα πολλών επιπέδων (βαθιά νευρωνικά δίκτυα) είναι ένα μέρος του κλάδου της εκμάθησης χαρακτηριστικών (feature learning ή representation learning) που είναι ένα μέρος της μηχανικής μάθησης η οποία με τη σειρά της ανήκει στο ευρύτερο επιστημονικό πεδίο της τεχνητής νοημοσύνης. Φυσικά, η τεχνητή νοημοσύνη περιλαμβάνει αρκετούς άλλους κλάδους εκτός από αυτόν της μηχανικής μάθησης[2]. Μια χρήσιμη παρατήρηση είναι ότι οι σχέσεις υποσυνόλου συμπίπτουν με τη χρονική αλληλουχία ανάπτυξης του κάθε κλάδου. Δηλαδή, κάθε υποσύνολο αναπτύχθηκε ταυτόχρονα ή αργότερα από το οποιοδήποτε υπερσύνολό του.

Στη συνέχεια, θα γίνει λόγος για τα στοιχεία εκείνα που περιλαμβάνουν την τεχνολογία των βαθιών νευρωνικών δικτύων προκειμένου να αποκτηθεί μια εποπτικότερη εικόνα.

[1] Από εδώ και στο εξής, με τον όρο «νευρωνικά δίκτυα» θα αναφερόμαστε στα «τεχνητά νευρωνικά δίκτυα».
[2] Βέβαια, ο κλάδος της μηχανικής μάθησης είναι σήμερα ο γρηγορότερα αναπτυσσόμενος.



Σχήμα 2.1: Διάγραμμα Venn όπου απεικονίζει τη θέση των νευρωνικών δικτύων στην οργάνωση της τεχνητής νοημοσύνης. *Παράχθηκε από το Microsoft Visio™.*

2.1.1 Μηχανική Μάθηση

Όπως προδίδει ο όρος, σε αδρές γραμμές τα συστήματα μηχανικής μάθησης έχουν τη δυνατότητα να μαθαίνουν μια εργασία χωρίς να έχουν προγραμματιστεί με ρητές εντολές για τη συγκεκριμένη εργασία αυτή[3]. Ίσως, ο πιο πλήρης ορισμός δίνεται από τον Tom M. Mitchell [50] σύμφωνα με τον οποίο, ένα υπολογιστικό πρόγραμμα λέγεται ότι μαθαίνει από μια εμπειρία E, ως προς ένα σύνολο εργασιών T και ένα μέτρο απόδοσης P, εάν η απόδοσή του σε εργασίες του T, όπως αυτή μετριέται από το P, βελτιώνεται με την E. [4]

Σύμφωνα με τον ανωτέρω ορισμό διακρίνουμε τρία βασικά συστατικά ενός συστήματος μηχανικής μάθησης. Αυτά είναι τα παρακάτω:

Εργασία - T Είναι το πρόβλημα το οποίο επιθυμούμε να λύσουμε.

Μέτρο Απόδοσης - P Αποτελεί μια μετρική του στόχου ως ένδειξη ποιότητας της λύσης μας. Από μαθηματική σκοπιά, είναι αυτό που ο αλγόριθμος μάθησης βελτιστοποιεί.

Εμπειρία - E Πρόκειται για τα δεδομένα εισόδου που λαμβάνει το σύστημα υπό τη μορφή παραδειγμάτων ή ως ερεθίσματα ανάδρασης από το περιβάλλον. Όπως θα δούμε στη συνέχεια,

[3] Η δυνατότητα αυτή είναι πολύ σημαντική αφού, όπως διαπιστώσαμε στην ενότητα 1.2 όταν έγινε λόγος για τα έμπειρα συστήματα, για πολλές εργασίες είναι πρακτικός αδύνατο να περιγραφούν ρητά και ντετερμινιστικά οι λύσεις τους.

[4] Ο ορισμός αυτός εξηγεί γιατί για παράδειγμα η λήψη μιας ιστοσελίδας της Wikipedia και η αποθήκευσή της τοπικά στον υπολογιστή δεν αποτελεί μηχανική μάθηση. Όπως προκύπτει, η «γνώση» αυτή δεν καθιστά καλύτερο τον υπολογιστή σε κάποια εργασία [51].

ο τρόπος απόκτησης αυτών των δεδομένων αλλά και η φύση τους καθορίζει το είδος της μάθησης.

Βασικά Είδη Συστημάτων Μηχανικής Μάθησης

Τα είδη των συστημάτων μηχανικής μάθησης μπορούν να ταξινομηθούν ανάλογα με το:

- *Αν εκπαιδεύονται με ανθρώπινη επίβλεψη.*
  Ανάλογα με αυτό το κριτήριο έχουμε τις εξής βασικές κατηγορίες: επιβλεπόμενη (supervised), μη-επιβλεπόμενη (un-supervised) και ενισχυτική μάθηση (reinforcement learning).
- *Αν μαθαίνουν σταδιακά (incrementally) και «στον αέρα» (on the fly).*
  Σε αυτήν την περίπτωση χωρίζουμε τα συστήματα μηχανικής μάθησης σε αυτά που πραγματοποιούν μάθηση σε ζωντανό χρόνο (online learning) και σε αυτά που μαθαίνουν κατά δέσμες (batch learning).
- *Αν κατασκευάζουν μοντέλα προσαρμοσμένα στα δεδομένα.*
  Με αυτό το κριτήριο χωρίζονται σε συστήματα βασισμένα σε μοντέλο (model-based) ή σε συστήματα βασισμένα σε παραδείγματα (instance-based). [51]

Προφανώς, κάθε δυνατός συνδυασμός των παραπάνω κριτηρίων είναι αποδεκτός, οδηγώντας έτσι στην ταξινόμηση των συστημάτων μηχανικής μάθησης σε μια πληθώρα από διαφορετικές κατηγορίες. Κρίνεται χρήσιμο, να αναφέρουμε σε όλη την έκταση του έργου τις κατηγορίες στις οποίες ανήκει το κάθε σύστημα που παρουσιάζουμε. Για αυτόν τον λόγο, παροτρύνουμε τον αναγνώστη που δεν είναι εξοικειωμένος με τους ανωτέρω όρους να διαβάσει τους αντίστοιχους ορισμούς στο παράρτημα Α΄.

2.1.2 Εκμάθηση Χαρακτηριστικών

Η ανάπτυξη των πρώτων συστημάτων μηχανικής μάθησης απεμπόλησε την ανάγκη των ευφυών εφαρμογών για σχολαστική και ρητή (hard-coded) αναπαράσταση του χώρου του προβλήματος (π.χ. με τη χρήση προτασιακής λογικής). Με τα νέα συστήματα, η γνώση για το πρόβλημα μαθαίνονταν αυτοματοποιημένα μέσω αλγορίθμων μάθησης από το σύνολο δεδομένων εκπαίδευσης. Με άλλα λόγια, τα αλγοριθμικά κατασκευάσματα μάθαιναν να αντιστοιχίζουν με αυτοματοποιημένο τρόπο τα δεδομένα εισόδου (κωδικοποιημένα σε μια μορφή αναπαράστασης) σε τιμές εξόδου.

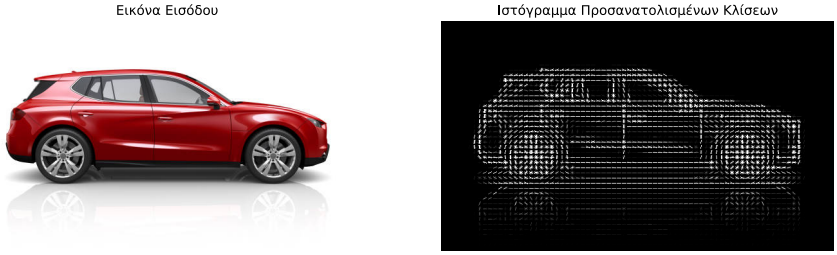
Παρόλα αυτά, τα πρώτα, απλά συστήματα μηχανικής μάθηση δεν έλυσαν όλα τα προβλήματα. Όπως είναι εμφανές από την ανωτέρω περιγραφή, αν και δεν απαιτούνταν η λεπτομερής συγγραφή βάσεων γνώσης, παρέμενε η ανάγκη για αναπαράσταση των δεδομένων εισόδου με μια αποδοτική μορφή. Είναι γεγονός, άλλωστε, ότι η αναπαράσταση σε πολλά συστήματα επηρεάζει καθοριστικά την απόδοση του συστήματος[5]. Για αυτόν τον λόγο, εξελίχθηκαν διαδικασίες «μηχανικής χαρακτηριστικών» (feature engineering) όπου αξιοποιώντας την τεχνική γνώση του χώρου του προβλήματος (domain knowledge) στόχος είναι η αναπαράσταση των ακατέργαστων δεδομένων εκπαίδευσης ως σύνολο (συνήθως διάνυσμα) από κατάλληλα χαρακτηριστικά. Η καταλληλότητα έγκειται στο πόσο χρήσιμη πληροφορία παρέχουν τα χαρακτηριστικά υπό τον περιορισμό να

[5]Η σημασία της αναπαράστασης δεδομένων στην απόδοση των αλγοριθμικών κατασκευασμάτων δε θα πρέπει να μας εκπλήσσει αφού κάτι αντίστοιχο ισχύει και στους ανθρώπους. Για παράδειγμα, οι περισσότεροι είναι πολύ πιο αποδοτικοί στην αριθμητική χρησιμοποιώντας την αραβική αναπαράσταση αριθμών απ΄ ότι τη λατινική [33].

είναι όσο το δυνατόν περισσότερο ανεξάρτητα μεταξύ τους ώστε να αποπλέκουν (disentagle) τους παράγοντες διακύμανσης (factors of variation) των δεδομένων που επηρεάζουν την τιμή εξόδου [33].

Οι ανωτέρω έννοιες μπορούν να καταστούν περισσότερο κατανοητές με ένα παράδειγμα συστήματος εκτίμησης τιμών κατοικιών [51] (πρόβλημα παλινδρόμησης, επίλυση με επιβλεπόμενη μάθηση κατά δέσμες). Πιο συγκεκριμένα, δοθέντος ενός συνόλου ακατέργαστων δεδομένων που αφορούν την αγορά σπιτιών σε μια περιοχή, το σύστημα, μέσω μηχανικής μάθησης, θα είναι ικανό να εκτιμήσει την τιμή με την οποία μια κατοικία θα πρέπει να κοστολογηθεί για να βγει στην αγορά. Όπως εξηγήσαμε, προτού τροφοδοτήσουμε το σύστημα με το σύνολο δεδομένων, είναι σκόπιμο να εφαρμόσουμε διαδικασίες μηχανικής χαρακτηριστικών σε αυτά και να δημιουργήσουμε μια νέα αναπαράσταση. Τα ακατέργαστα δεδομένα εκπαίδευσης αποτελούνται από μια λίστα όπου κάθε γραμμή αντιστοιχεί σε μια οικία με όλες τις προδιαγραφές της και την τιμή πώλησής της. Στο πρόβλημα του παραδείγματος:

- Ένας παράγοντας διακύμανσης θα μπορούσε να είναι η ακρίβεια της συγκεκριμένης περιοχής. Εντούτοις, σαν προδιαγραφές ας υποθέσουμε ότι αναφέρονται μόνο το γεωγραφικό πλάτος και γεωγραφικό μήκος με αποτέλεσμα η ακρίβεια της περιοχής να μην είναι άμεσα παρατηρήσιμη (συνηθισμένο φαινόμενο στους παράγοντες διακύμανσης). Θα μπορούσαμε λοιπόν να μετατρέψουμε τις συντεταγμένες σε ένα νέο χαρακτηριστικό: την «κλάση» της περιοχής. Ένας ακόμα παράγοντας διακύμανσης που είναι όμως άμεσα παρατηρήσιμος είναι το εμβαδόν επιφάνειας της κατοικίας.
- Μη χρήσιμη πληροφορία θα μπορούσε να είναι ο προσανατολισμός της οικίας. Σε αυτή την περίπτωσή, η δημιουργία μιας νέας αναπαράστασης δεδομένων χωρίς το παρόν χαρακτηριστικό θα βοηθούσε την επίδοση του συστήματος.
- Δύο αλληλοεξαρτώμενα χαρακτηριστικά (με υψηλή συν–διακύμανση) θα μπορούσαν να είναι ο αριθμός των υπνοδωματίων και ο αριθμός των μπάνιων όπου τότε η επιλογή της συγχώνευσής τους πιθανότατα θα βελτίωνε την απόδοση.



Σχήμα 2.2: Παράδειγμα εξαγωγής χαρακτηριστικών σε εικόνα ενός αυτοκινήτου με τη μέθοδο του Ιστογράμματος Προσανατολισμένων Κλίσεων (Histogram of Oriented Gradients). *Ο κώδικάς μας για την παραγωγή του σχήματος βρίσκεται, μαζί με όλες τις εικόνες της διπλωματικής σε αυτήν την ιστοσελίδα.*

Αν και στο παραπάνω πρόβλημα ήταν σχετικά εύκολη η «χειρονακτική» εξαγωγή χαρακτηρι-

στικών, υπάρχουν πολλοί χώροι προβλημάτων όπου κάτι τέτοιο είναι από πολύ απαιτητικό έως απίθανο. Ενδεικτικά, σε ένα πρόβλημα οπτικής αναγνώρισης ζώων και αντικειμένων (όπως αυτό του CIFAR-10 [52]) είναι εξαιρετικά δύσκολη η περιγραφή χαρακτηριστικών που θα λαμβάνουν μια αναπαράσταση σε εικονοστοιχεία (pixel) και θα παράγουν μια χρήσιμη αναπαράσταση. Μολονότι υπάρχουν γενικευμένες μέθοδοι για την εξαγωγή χαρακτηριστικών σε εικόνες όπως αυτή του σχήματος 2.2, αυτές δεν είναι βέλτιστα προσαρμοσμένες στον χώρο του εκάστοτε προβλήματος. Απόρροια αυτού μεταξύ άλλων είναι η απόρριψη στοιχείων των ακατέργαστων δεδομένων που μπορεί να είναι χρήσιμα (π.χ. σε ένα πρόβλημα αναγνώρισης μάρκας και χρώματος αυτοκινήτου, η μέθοδος Ιστογράμματος Προσανατολισμένων Κλίσεων θα απέρριπτε το χρώμα, στοιχείο χρήσιμο για τον χώρο του προβλήματος).

Η λύση για την αντιμετώπιση των προβλημάτων της χειροκίνητης εξαγωγής χαρακτηριστικών είναι η χρήση των αλγορίθμων μηχανικής μάθησης για την εκμάθηση όχι μόνο της αντιστοίχισης των δεδομένων εκπαίδευσης στην επιθυμητή έξοδο αλλά και των ίδιων των αναπαραστάσεων των δεδομένων. Αν και συνήθως, οι προκύπτουσες αναπαραστάσεις μετά τον μετασχηματισμό των ακατέργαστων δεδομένων δεν είναι κατανοητές από τον άνθρωπο, εφόσον η εκπαίδευση γίνει επιτυχημένα, αποτελούνται από χρήσιμα χαρακτηριστικά (που κωδικοποιούν τους παράγοντες διακύμανσης). Χαρακτηριστικό παράδειγμα συστήματος για την εκμάθηση χαρακτηριστικών είναι ο Αυτοκωδικοποιητής (Autoencoder).

2.1.3 Πολυεπίπεδα Νευρωνικά Δίκτυα

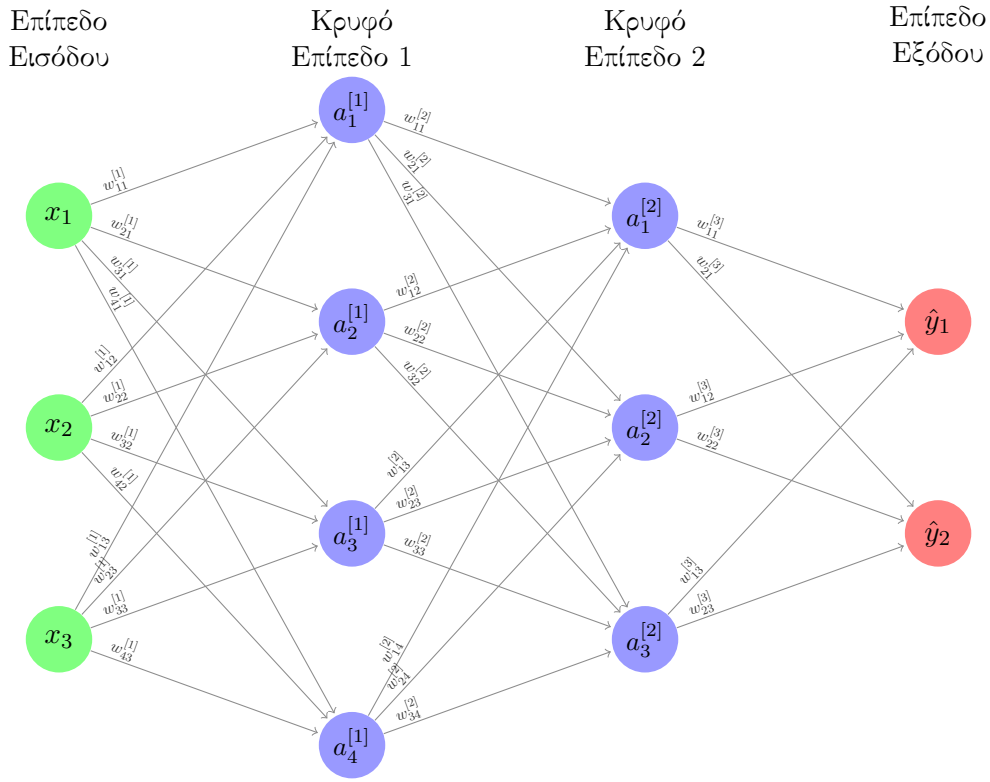
Η εκμάθηση χαρακτηριστικών σε συνδυασμό με τις ιδέες του κονεκτιβισμού περί κατανεμημένης αναπαράστασης (βλ. ενότητα 1.2) μας οδηγεί αναπόφευκτα στη βαθιά μάθηση (deep learning). Υπό μια αφαιρετική σκοπιά, πρόκειται για τα λεγόμενα «πολυεπίπεδα νευρωνικά δίκτυα» τα οποία συνδυάζουν τόσο τον μετασχηματισμό της αναπαράστασης των δεδομένων εισόδου όσο και την αντιστοίχιση αυτών των νέων αναπαραστάσεων στην τιμή εξόδου. Τα συστήματα αυτά, όπως θα δούμε στη συνέχεια, είναι δομημένα από απλές υπολογιστικές μονάδες που τους επιτρέπουν να δημιουργούν σύνθετες αναπαραστάσεις μέσω μιας σειράς από εμφωλευμένες, απλούστερες αναπαραστάσεις. Σημειώνουμε ότι πραγματοποιούν μάθηση με κατασκευή μοντέλου (model–based learning systems) και χρησιμοποιούνται τόσο σε προβλήματα ταξινόμησης όσο και παλινδρόμησης. Στις επόμενες παραγράφους, θα περιγράψουμε με μεγαλύτερη λεπτομέρεια τον χώρο των νευρωνικών δικτύων[6].

Δομή Απλών Νευρωνικών Δικτύων

Τα Νευρωνικά Δίκτυα στην πιο βασική τους μορφή (Feedforward Neural Networks ή Multilayer Perceptron) αποτελούνται από απλούς τεχνητούς νευρώνες διασυνδεδεμένους μεταξύ τους με συνάψεις σχηματίζοντας μια πολυεπίπεδη διάταξη. Το πρώτο επίπεδο ονομάζεται επίπεδο εισόδου (input layer) ενώ το τελευταίο ονομάζεται επίπεδο εξόδου (output layer). Όλα τα ενδιάμεσα επίπεδα λέγονται κρυφά επίπεδα (hidden layers) διότι οι τιμές τους δε δίνονται από τα δεδομένα [33]. Στην απλή περίπτωση που εξετάζουμε, κάθε νευρώνας δέχεται ως είσοδο τιμές από όλους τους νευρώνες του αμέσως προηγούμενου επιπέδου (fully connected layer) και αφού κάνει

[6]Για έναν τυπικό ορισμό, παραπέμπουμε τον αναγνώστη στο παράρτημα Α΄

υπολογισμούς με αυτές στέλνει το αποτέλεσμα σε όλους τους νευρώνες του αμέσως επόμενου επιπέδου.



Σχήμα 2.3: Διάγραμμα Τεχνητού Νευρωνικού Δικτύου με δύο κρυφά επίπεδα. Οι αγκύλες στους εκθέτες προσδιορίζουν τον αριθμό του επιπέδου. *Παράχθηκε από το LaTeX πακέτο neuralnetwork. Το πακέτο τροποποιήθηκε και επεκτάθηκε τοπικά.*

Η αναπαράσταση των νευρωνικών δικτύων γίνεται με έναν γράφο από ακμές (συνάψεις) και κόμβους (νευρώνες ή τιμές εισόδου). Κοιτώντας κανείς το σχήμα 2.3 μπορεί να παρατηρήσει πως οι κόμβοι των επιπέδων εισόδου και εξόδου ξεχωρίζουν από τους κόμβους των κρυφών επιπέδων. Αυτό έχει γίνει για να τονιστεί η ξεχωριστή λειτουργία τους. Πιο συγκεκριμένα, στην περίπτωση του επιπέδου εισόδου, αυτό περιέχει τόσους κόμβους όσος είναι και ο αριθμός των χαρακτηριστικών που περιγράφουν το κάθε παράδειγμα (δηλαδή όσο και το μήκος του διανύσματος εισόδου). Ουσιαστικά, οι κόμβοι εισόδου απλά λαμβάνουν τις τιμές των χαρακτηριστικών και, χωρίς να τις μεταβάλλουν, τις δρομολογούν στους κόμβους του επόμενου επιπέδου (για αυτό και αποφεύγεται η επονομασία αυτών των κόμβων ως νευρώνες). Στην περίπτωση του επιπέδου εξόδου, ο αριθμός των κόμβων είναι τόσος όσος και ο αριθμός των χαρακτηριστικών για την περιγραφή της πρόβλεψης (ίσος με το μήκος του διανύσματος εξόδου). Οι κόμβοι εξόδου συνήθως επιβάλλουν περιορισμούς στις τιμές των χαρακτηριστικών εξόδου ώστε αυτές να ανήκουν σε ένα φραγμένο σύνολο αριθμών (π.χ. το [0,1]).

Ένα νευρωνικό δίκτυο χωρίς κρυφά επίπεδα δε διαφέρει από έναν γραμμικό ταξινομητή. Είναι γεγονός ότι οι εκπληκτικές δυνατότητες των νευρωνικών δικτύων αποδίδονται στα κρυφά επίπεδα. Χάρη σε αυτά είναι δυνατή η σταδιακή σύνθεση αφηρημένων αναπαραστάσεων από επίπεδο σε επίπεδο που κωδικοποιούν τους παράγοντες διακύμανσης. Τα κρυφά επίπεδα τα απαρτίζουν

οι κόμβοι κρυφού επιπέδου[7]. Ο κάθε ένας από αυτούς υπολογίζει την έξοδο μιας μη γραμμικής συνάρτησης με είσοδο ένα γραμμικό συνδυασμό των τιμών των κόμβων του προηγούμενου επιπέδου. Αξίζει να αναφερθεί στο σημείο αυτό πως δεν υπάρχει κάποιος συγκεκριμένος περιορισμός για τον αριθμό των κόμβων των κρυφών επιπέδων.

Η φορμαλιστική περιγραφή των παραμέτρων[8] και των υπολογισμών που λαμβάνουν χώρα κατά τη διαδικασία πρόβλεψης ενός νευρωνικού δικτύου περιγράφονται παρακάτω.

Έστω ένα παράδειγμα εισόδου το οποίο περιγράφεται από n_x χαρακτηριστικά με το διάνυσμα $X = [x_1, x_2, x_3, \ldots, x_{n_x}]^T$. Όλα τα δεδομένα εκπαίδευσης, έστω M, μπορούν να ομαδοποιηθούν σε έναν πίνακα $\boldsymbol{X}$ ως εξής:

$$\boldsymbol{X} = \underset{(n_x \times M)}{\begin{bmatrix} | & | & & | \\ X^{(1)} & X^{(2)} & \ldots & X^{(M)} \\ | & | & & | \end{bmatrix}}. \tag{2.1}$$

Όπου οι παρενθέσεις στους εκθέτες δηλώνουν τον αριθμό του παραδείγματος.

Αφού προσδιορίσαμε μια μαθηματική αναπαράσταση για τα δεδομένα εισόδου, πάμε να προσδιορίσουμε με φορμαλιστικό τρόπο τις παραμέτρους του νευρωνικού δικτύου. Με L θα συμβολίζουμε τον αριθμό των επιπέδων του νευρωνικού δικτύου (χωρίς να μετράμε το επίπεδο εισόδου). Για παράδειγμα, στο δίκτυο του σχήματος 2.3 ισχύει $L = 3$. Επίσης, ο αριθμός των κόμβων ενός επιπέδου, έστω l, θα συμβολίζεται με $n^{[l]}$. Προφανώς, θα ισχύει ότι $l \in [0, L]$. Στο παράδειγμα του σχήματος 2.3 ισχύει $n^{[0]} = n_x = 3, n^{[1]} = 4, n^{[2]} = 3, n^{[3]} = n_y = 2$. Έχοντας δώσει έναν συμβολισμό ορισμένων βασικών υπερπαραμέτρων[9], μπορούμε να αποδώσουμε φορμαλιστικά τις παραμέτρους του δικτύου οι οποίες είναι:

- Τα βάρη των ακμών (weights) που συνδέουν δύο διαδοχικά επίπεδα.
  Τα βάρη μεταξύ διαδοχικών επιπέδων $l-1$ και l μπορούμε να τα οργανώσουμε σε μια μορφή πίνακα ως εξής:

$$\boldsymbol{W}^{[l]} = \underset{(n^{[l]} \times n^{[l-1]})}{\begin{bmatrix} w_{11}^{[l]} & w_{12}^{[l]} & \ldots & w_{1n^{[l-1]}}^{[l]} \\ w_{21}^{[l]} & w_{22}^{[l]} & \ldots & w_{2n^{[l-1]}}^{[l]} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n^{[l]}1}^{[l]} & w_{n^{[l]}2}^{[l]} & \ldots & w_{n^{[l]}n^{[l-1]}}^{[l]} \end{bmatrix}}. \tag{2.2}$$

- Τα δυναμικά πόλωσης (biases) του κάθε νευρώνα.
  Όπως είναι λογικό, οι κόμβοι του επιπέδου 0 (επίπεδο εισόδου) δε διαθέτουν δυναμικά πόλωσης. Για όλους τους άλλους κόμβους σε κάθε επίπεδο (έστω l) έχουμε το εξής

[7] Εφεξής θα αποκαλούνται ως κόμβοι.

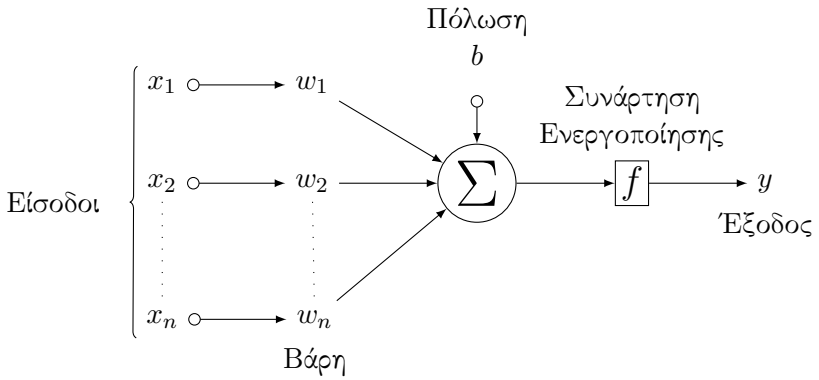
[8] Πρόκειται για μεταβλητές των οποίων οι τιμές μαθαίνονται κατά τη διάρκεια της εκπαίδευσης. Έτσι το νευρωνικό δίκτυο λέμε ότι προσαρμόζεται στα δεδομένα.

[9] Σε αντίθεση με τις παραμέτρους, οι υπερπαράμετροι είναι μεταβλητές που ορίζει ο χρήστης και δε μεταβάλλονται κατά την εκπαίδευση. Ονομάζονται έτσι διότι ελέγχουν έμμεσα τις τιμές των παραμέτρων.

διάνυσμα στήλη:

$$\boldsymbol{b}^{[l]} = \underset{(n^{[l]}\times 1)}{\begin{bmatrix} b_1^{[l]} \\ b_2^{[l]} \\ \vdots \\ b_{n^{[l]}}^{[l]} \end{bmatrix}}. \tag{2.3}$$

Τώρα, είμαστε σε θέση να περιγράψουμε τους υπολογισμούς που πραγματοποιεί κάθε νευρώνας. Για τον σκοπό αυτό, παρουσιάζουμε μια αναπαράστασή του στο σχήμα 2.4.



Σχήμα 2.4: Διάγραμμα ενός τεχνητού νευρώνα. *Παράχθηκε από το πακέτο tikz.*

Εσωτερικά ο κάθε νευρώνας δέχεται ως είσοδο τις τιμές των νευρώνων του προηγούμενου επιπέδου ή (στην περίπτωση του δεύτερου επιπέδου) τις τιμές του παραδείγματος εκπαίδευσης και πραγματοποιεί υπολογισμούς με αυτές για να παράξει μια τιμή εξόδου. Την κάθε τιμή εισόδου τη συμβολίζουμε με x ενώ την τιμή εξόδου του μεμονωμένου νευρώνα τη συμβολίζουμε με y. Η πράξη που επιτελείται εσωτερικά είναι:

$$y = f\Big(\sum_{i=1}^{n} w_i * x_i + b\Big) \tag{2.4}$$

Όπου $f(x)$ είναι η συνάρτηση ενεργοποίησης (activation function). Αυτές οι συναρτήσεις, μεταξύ άλλων, επιτρέπουν στο σύστημα να μοντελοποιήσει μη γραμμικές σχέσεις εισόδου–εξόδου. Πρωταρχικά, χρησιμοποιούνταν για να μοντελοποιήσουν τη μη–γραμμική, δίτιμη έξοδο των βιολογικών νευρώνων. Στον χώρο των υπολογιστών όμως, έγιναν γρήγορα αντιληπτά τα πρακτικά πλεονεκτήματα της κατασκευής τεχνητών νευρώνων με έξοδο συνεχείς τιμές που ανήκουν στον χώρο των πραγματικών αριθμών[10]. Έτσι, από τη βηματική συνάρτηση (όπως αυτή στο Perceptron του H. Rosenblat [32]) που έχει ως σύνολο τιμών το $\{0, 1\}$, άρχισε να γίνεται χρήση άλλων με συνεχές πεδίο τιμών όπως για παράδειγμα η σιγμοειδής ή η υπερβολική συνάρτηση εφαπτομένης (tanh) κ.τ.λ.

Στο σημείο αυτό να αναφέρουμε πως το όρισμα της συνάρτησης ενεργοποίησης, δηλαδή την

[10]Αυτός είναι και ο λόγος που ο όρος «ενεργοποίηση» θέλει προσοχή όταν αναφερόμαστε σε τεχνητούς νευρώνες. Στην περίπτωση χρήσης του όρου σε ένα τέτοιο πλαίσιο, θα εννοούμε ότι ο νευρώνας αυτός έχει σχετικά μεγαλύτερη τιμή εξόδου και συνάμα έχει μεγαλύτερη «ευθύνη» στη διαμόρφωση της τελικής εξόδου.

ποσότητα $\sum_{i=1}^{n} w_i * x_i + b$ τη συμβολίζουμε με z ενώ την τιμή του y την ονομάζουμε και τιμή ενεργοποίησης (συμβολίζοντάς τη με a). Στην περίπτωση δε που ο νευρώνας ανήκει στο επίπεδο εξόδου, την τιμή y την ονομάζουμε τιμή πρόβλεψης και την αναπαριστούμε με το γράμμα $\hat{y}$.

Κοιτώντας τώρα εποπτικά τη διαδικασία παραγωγής νέων προβλέψεων ενός τυπικού νευρωνικού δικτύου, μπορούμε να την περιγράψουμε χρησιμοποιώντας αναπαράσταση με διανύσματα (vector notation). Αναλυτικότερα, αν συγκεντρώσουμε όλες τις τιμές ενεργοποίησης a ενός επιπέδου l σε ένα διάνυσμα $A^{[l]} = \left[a_1^{[l]}, a_2^{[l]}, a_3^{[l]}, \ldots, a_{n^{[l]}}^{[l]}\right]^T$ και κάνουμε το ίδιο για τα ορίσματα z των συναρτήσεων ενεργοποίησης δηλαδή $Z^{[l]} = \left[z_1^{[l]}, z_2^{[l]}, z_3^{[l]}, \ldots, z_{n^{[l]}}^{[l]}\right]^T$ τότε συνολικά, γράφουμε:

- Τα στοιχεία a προκύπτουν από τα στοιχεία z του ίδιου επιπέδου ως εξής:

$$A^{[l]} = F^{[l]}\left(Z^{[l]}\right) \tag{2.5}$$

  Όπου η συνάρτηση F εφαρμόζει την f σε κάθε στοιχείο του ορίσματός της ξεχωριστά (elementwise).

- Τα στοιχεία z υπολογίζονται από τα στοιχεία a του προηγούμενου επιπέδου μέσω της σχέσης:

$$Z^{[l]} = W^{[l]} \times A^{[l-1]} + \boldsymbol{b}^{[l]} \tag{2.6}$$

  Όπου $A^0 = X$, το διάνυσμα χαρακτηριστικών ενός παραδείγματος ενώ $A^{[L]} = \hat{Y}$ το διάνυσμα εξόδου.

Η ανωτέρω ανάλυση πραγματοποιήθηκε για ένα μεμονωμένο παράδειγμα. Θα βοηθήσει για την επεξήγηση του τρόπου εκπαίδευσης των νευρωνικών δικτύων αν παρουσιάσουμε τώρα μια μορφή μαθηματικών τύπων που συγκεντρώνουν τις τιμές όλων των παραδειγμάτων του συνόλου εκπαίδευσης. Σημειώστε, ότι κάθε στήλη του πίνακα δεδομένων εισόδου $\boldsymbol{X}$ περιέχει ένα διάνυσμα παραδείγματος με αποτέλεσμα ο αριθμός των στηλών να ισούται με τον αριθμό των παραδειγμάτων, M. Κατά αυτόν τον τρόπο έχουμε:

- Για τα z ενός επιπέδου l για το κάθε παράδειγμα εισόδου:

$$\boldsymbol{Z^{[l]}} = \underset{(n^{[l]} \times M)}{\begin{bmatrix} | & | & & | \\ Z^{[l](1)} & Z^{[l](2)} & \ldots & Z^{[l](M)} \\ | & | & & | \end{bmatrix}}. \tag{2.7}$$

- Αντίστοιχα, για τα a ενός επιπέδου l για το κάθε παράδειγμα εισόδου:

$$\boldsymbol{A^{[l]}} = \underset{(n^{[l]} \times M)}{\begin{bmatrix} | & | & & | \\ A^{[l](1)} & A^{[l](2)} & \ldots & A^{[l](M)} \\ | & | & & | \end{bmatrix}}. \tag{2.8}$$

Όπως προκύπτουν από τις σχέσεις

$$\underset{(n^{[l]} \times M)}{\boldsymbol{Z}^{[l]}} = \underset{(n^{[l]} \times n^{[l-1]})}{W^{[l]}} \times \underset{(n^{[l-1]} \times M)}{\boldsymbol{A}^{[l-1]}} + \underset{(n^{[l]} \times 1)}{\boldsymbol{b}^{[l]}} \tag{2.9}$$

και

$$\underset{(n^{[l]}\times M)}{\boldsymbol{A}^{[l]}} = F^{[l]}\big(\underset{(n^{[l]}\times M)}{\boldsymbol{Z}^{[l]}}\big) \tag{2.10}$$

αντίστοιχα, όπου η μόνη διαφορά με τις 2.6, 2.5 είναι ότι αντικαταστήσαμε τα διανύσματα στήλες Z και A με τους πίνακες $\boldsymbol{Z}$ και $\boldsymbol{A}$. Και πάλι, $\boldsymbol{A}^0 = \boldsymbol{X}$, ο πίνακας όλων των δεδομένων εισόδου ενώ $\boldsymbol{A}^{[L]} = \hat{\boldsymbol{Y}}$ το σύνολο διανυσμάτων εξόδου.

Εκπαίδευση Νευρωνικών Δικτύων

Στην προηγούμενη παράγραφο διατυπώσαμε τους μαθηματικούς τύπους σύμφωνα με τους οποίους ένα νευρωνικό δίκτυο, δοθέντος ενός συνόλου δεδομένων $\boldsymbol{X}$ παράγει ένα σύνολο από προβλέψεις $\hat{\boldsymbol{Y}}$. Η διαδικασία αυτή ονομάζεται και πρόσθια διάδοση (forward propagation). Παρόλα αυτά, δεν αναφερθήκαμε καθόλου στη διαδικασία μάθησης του δικτύου. Υποθέσαμε σιωπηρά ότι αυτό ήταν ήδη εκπαιδευμένο και οι παράμετροί του (τα βάρη και τα δυναμικά πόλωσης) ήταν σταθερά. Με τη μέχρι τώρα παρουσίαση, το νευρωνικό δίκτυο δεν είναι τίποτα άλλο παρά μια μη γραμμική συνάρτηση. Σε αυτήν την παράγραφο όμως, θα κάνουμε τη σύνδεση των νευρωνικών δικτύων με τη μηχανική μάθηση αναλύοντας τον μηχανισμό εκπαίδευσής τους.

Όπως έχουμε αναφέρει, τα νευρωνικά δίκτυα είναι πολυδύναμα συστήματα μηχανικής μάθησης βασισμένα σε μοντέλο ενώ καμία επιπλέον υπόθεση δεν μπορεί να γίνει εκ των προτέρων. Με άλλα λόγια, υπάρχουν νευρωνικά δίκτυα που ανήκουν σε όλες τις υπόλοιπες κατηγορίες που παρατέθηκαν στην ενότητα 2.1.1. Παρόλα αυτά, για τον σκοπό της παρούσας παραγράφου, θα περιορίσουμε αυτά τα πολυδύναμα συστήματα σε αυτά που πραγματοποιούν επιβλεπόμενη μάθηση. Ευτυχώς, αυτή είναι η πιο βασική κατηγορία και οι ιδέες που θα παρουσιαστούν υπό αυτή εύκολα μεταφέρονται και στις υπόλοιπες.

Στο πλαίσιο της επιβλεπόμενης μάθησης, εκτός από το σύνολο δεδομένων εισόδου $\boldsymbol{X}$ παρέχεται, και ένα σύνολο δεδομένων εξόδου $\boldsymbol{Y}$. Το τελευταίο αποτελείται από ένα επιθυμητό διάνυσμα (ή τιμή) στόχο για κάθε παράδειγμα του συνόλου δεδομένων εισόδου. Έτσι, (όπως αναφέρουμε και στο παράρτημα Α′) σχηματίζονται ζεύγη διανυσμάτων εισόδου–επιθυμητής εξόδου. Στόχος του νευρωνικού δικτύου σε αυτήν την περίπτωση είναι να δημιουργήσει μια συνάρτηση που θα κάνει την αντιστοίχιση από τα παραδείγματα X στις προβλέψεις του στόχου, $\hat{Y}$ να είναι όσο πιο πιστή γίνεται στην αντιστοίχιση X σε Y. Με μαθηματικούς όρους, έστω η συνάρτηση του νευρωνικού δικτύου: $\mathcal{F}(X;\overline{W},\overline{b})$, όπου με $\overline{W}$ και $\overline{b}$ συμβολίζεται αντίστοιχα το σύνολο των βαρών και δυναμικών πόλωσης σε όλα τα επίπεδα[11]. Θέλουμε για τη συνάρτηση $\mathcal{F}(X;\overline{W},\overline{b}) : X \to \hat{Y}$ να ισχύει $\mathcal{F}(X;\overline{W^*},\overline{b^*}) \approx \mathcal{G}(X)$ όπου $\mathcal{G}$ η (άγνωστη) συνάρτηση από την οποία (θεωρητικά) παράχθηκαν τα ζεύγη εισόδου–εξόδου και οι δείκτες αστερίσκοι συμβολίζουν τις ιδανικές τιμές των παραμέτρων[12].

[11] Το σύμβολο «;» διαβάζεται ως «παραμετροποιημένο από».

[12] Ουσιαστικά, το κύριο πρόβλημα που καλούνται να λύσουν τα νευρωνικά δίκτυα (και τα συστήματα μηχανικής μάθησης γενικότερα) πηγάζει από το γεγονός ότι η συνάρτηση $\mathcal{G}$ είναι άγνωστη. Αντ΄ αυτής, διατίθεται ένα σύνολο ζευγών $X - Y$ που αποτελεί υποσύνολο του πληθυσμού όλων των δυνατών εισόδων και το σύστημα επιδιώκει από αυτό το υποσύνολο να μάθει να γενικεύει σε παραδείγματα που δεν έχει δει. Αυτός είναι και ο λόγος που, υπό την αυστηρά μαθηματική έννοια, η διαδικασία της εκπαίδευσης αποκαλείται και συμπερασματολογία (inference). Σε τελική ανάλυση, τα περισσότερα νευρωνικά δίκτυα εκπαιδεύονται χρησιμοποιώντας συμπερασματολογία βασισμένη στη μέγιστη πιθανοφάνεια (maximum likelihood inference) [33]. Δηλαδή, προσπαθούν να μεγιστοποιήσουν την ποσότητα $p(y|x;\overline{W},\overline{b})$ και γιαυτό ονομάζονται μοντέλα διάκρισης (descriminative models).

Για να προσαρμόσουμε με βέλτιστο τρόπο τις παραμέτρους του μοντέλου στην εμπειρία απαιτείται (σύμφωνα με τον ορισμό της μηχανικής μάθησης) μια μετρική η οποία θα μας δείχνει πόσο κατάλληλη είναι η προσέγγισή μας (fittness). Αυτή η μετρική ονομάζεται συνάρτηση απώλειας (loss function) και στη γενική της μορφή είναι $\mathsf{L}(\hat{\boldsymbol{Y}}, \boldsymbol{Y}) = \mathsf{L}(\mathcal{F}(\boldsymbol{X}; \overline{W}, \overline{b}), \boldsymbol{Y})$. Ανάλογα με το είδος του προβλήματος, τυπικές συναρτήσεις απώλειας είναι:

- Η (binary cross entropy loss): $\mathsf{L}(\hat{\boldsymbol{Y}}, \boldsymbol{Y}) = -\frac{1}{M}\sum_{i=1}^{M}\left(y^{(i)}\log(\hat{y}^{(i)}) + (1-y^{(i)})\log(1-\hat{y}^{(i)})\right)$ στην περίπτωση προβλήματος δυαδικής ταξινόμησης (όπου η έξοδος $\hat{y}$ είναι ίση με την πιθανότητα το παράδειγμα x να ανήκει στην κλάση 1)
- Η (mean square error loss): $\mathsf{L}(\hat{\boldsymbol{Y}}, \boldsymbol{Y}) = \frac{1}{M}\sum_{i=1}^{M}\left\|y^{(i)} - \hat{y}^{(i)}\right\|^2$
στην περίπτωση προβλήματος παλινδρόμησης (όπου $\|x\|$ συνήθως είναι η $L2$ νόρμα).

Να σημειώσουμε ότι ορίσαμε τη συνάρτηση απώλειας ώστε να δέχεται ένα σύνολο από προβλέψεις και επιθυμητές τιμές στόχους. Αντίστοιχα, θα μπορούσαμε να ορίσουμε συνάρτηση απώλειας που συγκρίνει την πρόβλεψη με την έξοδο στόχο ενός συγκεκριμένου παραδείγματος (και να μην αθροίζει όλα τα παραδείγματα). Τότε, θα είχαμε $L(\hat{y}, y) = L(\mathcal{F}(X; \overline{W}, \overline{b}), y)$.

Έχοντας στη διάθεσή μας μια μετρική απόδοσης, το πρόβλημα της εκπαίδευσης του νευρωνικού δικτύου μπορεί να διατυπωθεί ως πρόβλημα βελτιστοποίησης. Με μαθηματικούς όρους έχουμε:

$$\overline{W^*}, \overline{b^*} = \underset{\overline{W}, \overline{b}}{\operatorname{argmin}}(\mathsf{L}(\mathcal{F}(\boldsymbol{X}; \overline{W}, \overline{b}), \boldsymbol{Y})) \tag{2.11}$$

Εξετάζοντας τη μετρική L σαν συνάρτηση των $\overline{W}$ και $\overline{b}$, για την επίλυση του ανωτέρω προβλήματος αρκεί να βρούμε το σημείο στον χώρο των παραμέτρων που την ελαχιστοποιεί. Λόγω των μη γραμμικών στοιχείων όμως, δεν υπάρχει κλειστός τύπος (closed form) για την εύρεση του σημείου αυτού. Όπως φαίνεται και από το σχήμα 2.5, η συνάρτηση απώλειας είναι μη κυρτή. Αυτό μας οδηγεί στη χρήση επαναληπτικών μεθόδων για την εύρεση κάποιου (τοπικού) ελάχιστου.

Ο πιο δημοφιλής αλγόριθμος για αυτόν τον σκοπό είναι ο αλγόριθμος καθόδου κλίσης (gradient descent). Σύμφωνα με τον αλγόριθμο αυτό, πραγματοποιούνται επαναλαμβανόμενα βήματα «καθόδου» προς την κατεύθυνση με τη μεγαλύτερη κλίση. Διαισθητικά, φαίνεται λογικό σε κάθε βήμα να υπολογίζουμε σημειακά την κλίση της συνάρτησης που θέλουμε να ελαχιστοποιήσουμε και να κινούμαστε προς την κατεύθυνση με τη μικρότερη κλίση. Πιο συγκεκριμένα, πρώτα αρχικοποιούνται ανεξάρτητα όλες οι παράμετροι σε τυχαίες τιμές $\overline{W}_0$, $\overline{b}_0$ και έπειτα ξεκινά μια επαναληπτική διαδικασία όπου σε κάθε βήμα αυτής (έστω i):

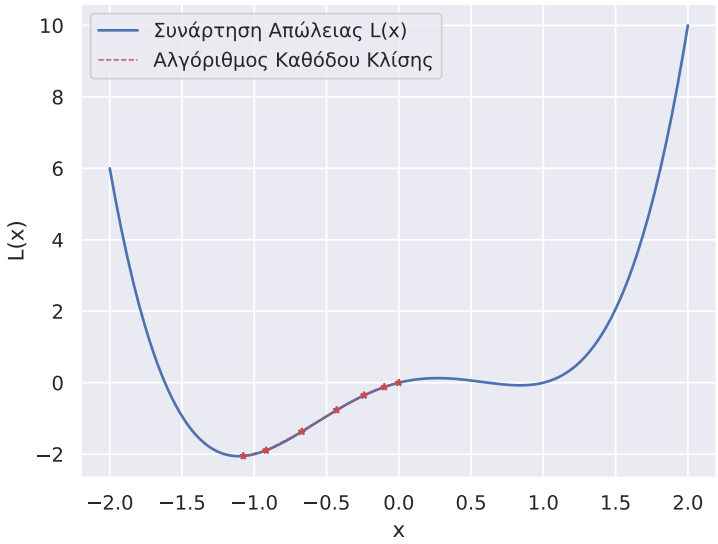
1. Υπολογίζονται οι μερικοί παράγωγοι (η κλήση) της συνάρτησης απώλειας ως προς όλες τις παραμέτρους σημειακά:

$$dw_i = \left.\frac{\partial \mathsf{L}(\overline{\mathsf{W}}, \overline{\mathsf{b}})}{\partial w}\right|_{(\overline{W}, \overline{b}) = (\overline{W}_{i-1}, \overline{b}_i - 1)}, \forall w \tag{2.12}$$

και

$$db_i = \left.\frac{\partial \mathsf{L}(\overline{\mathsf{W}}, \overline{\mathsf{b}})}{\partial b}\right|_{(\overline{W}, \overline{b}) = (\overline{W}_{i-1}, \overline{b}_i - 1)}, \forall b \tag{2.13}$$

Όπου $\mathsf{L}(\overline{\mathsf{W}}, \overline{\mathsf{b}})$ η συνάρτηση απώλειας υπολογισμένη για ένα σύνολο δεδομένων $\boldsymbol{X}$ υπό τις



Σχήμα 2.5: Γραφική παράσταση στην οποία εφαρμόζεται ο αλγόριθμος καθόδου κλίσης σε μια μη κυρτή συνάρτηση με μια παράμετρο (την x). Ο κώδικάς μας για την παραγωγή της γραφικής παράστασης βρίσκεται στην ιστοσελίδα του γραπτού μέρους της διπλωματικής.

παραμέτρους $\overline{W}, \overline{b}$. Συνοπτικά, αν συγκεντρώσουμε όλες τις παραμέτρους $\overline{W}$ και $\overline{b}$ στο διάνυσμα στήλη $\overline{W_{all}}$ τότε γράφουμε:

$$d\overline{W_{all}}_i = \nabla \mathsf{L}(\overline{\mathsf{W}_{\mathsf{all}}})\big|_{\overline{W_{all}}=\overline{W_{all}}_{i-1}} \tag{2.14}$$

2. Μετακινείται το σημείο στον χώρο παραμέτρων προς την κατεύθυνση της μεγαλύτερης κλίσης σύμφωνα με τον κανόνα ενημέρωσης (update rule) των παραμέτρων[13]. Ο κανόνας είναι ο εξής:

$$w_i = w_{i-1} - \alpha * dw_i, \forall w \tag{2.15}$$

και

$$b_i = b_{i-1} - \alpha * db_i, \forall b \tag{2.16}$$

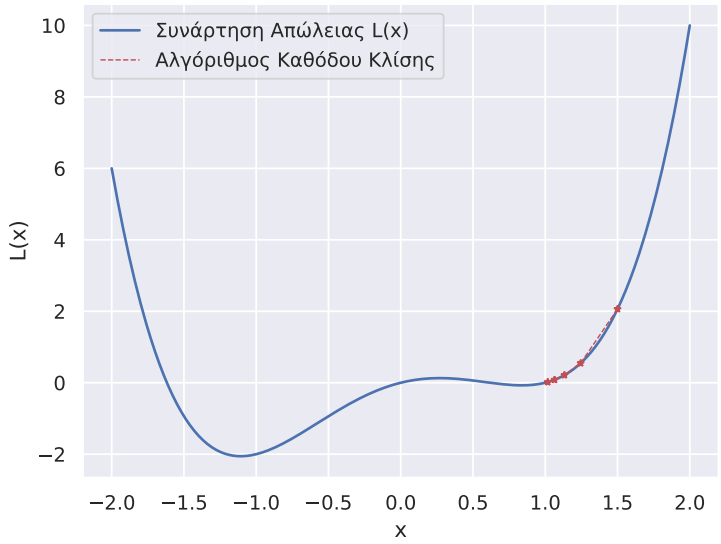
Όπου α ο ρυθμός μάθησης (learning rate): μια υπερπαράμετρος που καθορίζει το μέγεθος του βήματος κατά την ενημέρωση των παραμέτρων. Αντίστοιχα με πριν, συνοπτικά, έχουμε:

$$\overline{W_{all}}_i = \overline{W_{all}}_{i-1} - \alpha * d\overline{W_{all}}_i \tag{2.17}$$

Ο αλγόριθμος τελειώνει είτε όταν οι ενημερώσεις είναι πλέον αμελητέες και η τιμή της συνάρτησης απώλειας δε μειώνεται άλλο από επανάληψη σε επανάληψη (ο αλγόριθμος έχει βρει ένα τοπικό ελάχιστο) είτε όταν ξεπεραστεί ο μέγιστος αριθμός επαναλήψεων.

Να σημειώσουμε ότι ο αλγόριθμος καθόδου κλίσης δε βρίσκει πάντα το ολικό ελάχιστο της συνάρτησης. Ανάλογα με την αρχικοποίηση των παραμέτρων και τις τιμές των υπερπαραμέτρων

[13] Ας φανταστούμε τις παραγώγους ως «ευθύνες» της κάθε παραμέτρου για τις σωστές ή λάθος προβλέψεις. Όσο πιο μεγάλη η παράγωγος, τόσο πιο καθοριστικό ρόλο παίζει η μεταβλητή στη διαμόρφωση της τιμής της συνάρτησης απώλειας.



Σχήμα 2.6: Γραφική παράσταση στην οποία εφαρμόζεται ο αλγόριθμος καθόδου κλίσης σε μια μη κυρτή συνάρτηση με μια παράμετρο (την x) αρχικοποιημένη όμως στην τιμή 1.5 με αποτέλεσμα να βρίσκεται ένα τοπικό ελάχιστο.

(ρυθμός μάθησης, αριθμός επαναλήψεων) οδηγούμαστε κάθε φορά σε διαφορετικά αποτελέσματα. Για παράδειγμα, στο σχήμα 2.6 αρχικοποιήσαμε την τιμή της παραμέτρου x με την τιμή $+1.5$ με αποτέλεσμα ο αλγόριθμος να τερματίσει στο τοπικό ελάχιστο της συνάρτησης. Ανεξάρτητα από τον αριθμό επαναλήψεων, δε θα εντόπιζε ποτέ το ολικό ελάχιστο υπό αυτήν την αρχικοποίηση. Η αδυναμία εγγύησης για την εύρεση του ολικού ελαχίστου αποτελεί το λόγο για τον οποίο σε αρκετά προβλήματα μηχανικής μάθησης οι αλγόριθμοι εκπαιδεύονται πολλές φορές με διαφορετική όμως αρχικοποίηση των παραμέτρων τους. Ευτυχώς, στα πολυεπίπεδα νευρωνικά δίκτυα δεν ενδιαφέρει η εύρεση του ολικού ελαχίστου[14] αλλά ενός τοπικού ελαχίστου [53].

Ο αλγόριθμος της καθόδου κλίσης δε θα χρησίμευε στην εκπαίδευση των νευρωνικών δικτύων αν δεν υπήρχε η δυνατότητα αποδοτικού υπολογισμού των μερικών παραγώγων. Τη λειτουργία αυτή την επιτελεί η μέθοδος οπισθοδιάδοσης σφάλματος (back propagation). Με λίγα λόγια, πρόκειται για μια μέθοδο η οποία χρησιμοποιώντας τον κανόνα της αλυσίδας υπολογίζει την παράγωγο της συνάρτησης απώλειας ως προς όλες τις παραμέτρους του δικτύου (σημειακά), ξεκινώντας από αυτές του τελευταίου επιπέδου και τερματίζοντας σε αυτές του πρώτου.

Παρακάτω παρατίθενται οι υπολογισμοί που λαμβάνουν χώρα κατά τη διάρκεια εύρεσης των μερικών παραγώγων ως προς τις παραμέτρους ενός επιπέδου $L-1$ μέσω της οπισθοδιάδοσης σφάλματος για την i–οστή επανάληψη του αλγορίθμου καθόδου κλίσης (με δεδομένα εισόδου ένα σύνολο από M παραδείγματα). Αν και οι παράγωγοι υπολογίζονται σημειακά για τις τιμές των παραμέτρων $\overline{W}_i$ και $\overline{b}_i$, για λόγους ευκολότερης ανάγνωσης αυτό δε θα απεικονίζεται κατά τη διατύπωση των παρακάτω μερικών παραγώγων. Ξεκινώντας από το επίπεδο L έχουμε:

- Η παράγωγος της συνάρτησης απώλειας ως προς τα βάρη από τον κόμβο k του επιπέδου

[14] Καθώς οδηγεί σε overfitting, δηλαδή, υπερπροσαρμογή του νευρωνικού δικτύου στα δεδομένα εκπαίδευσης.

$L-1$ στον κόμβο j του επιπέδου L είναι:

$$\frac{\partial \mathsf{L}(\overline{\mathsf{W}},\overline{\mathsf{b}})}{\partial w_{jk}^{[L]}} = \frac{\partial \mathsf{z}_{\mathsf{j}}^{[\mathsf{L}]}}{\partial w_{jk}^{[L]}} * \frac{\partial \mathsf{a}_{\mathsf{j}}^{[\mathsf{L}]}}{\partial z_j^{[L]}} * \frac{\partial \mathsf{L}(\overline{\mathsf{W}},\overline{\mathsf{b}})}{\partial a_j^{[L]}} \tag{2.18}$$

Όπου ο όρος $\frac{\partial \mathsf{L}(\overline{\mathsf{W}},\overline{\mathsf{b}})}{\partial a_j^{[L]}}$ υπολογίζεται άμεσα από την επιλεγμένη συνάρτηση απώλειας.

Ο όρος $\frac{\partial \mathsf{a}_{\mathsf{j}}^{[\mathsf{L}]}}{\partial z_j^{[L]}}$ υπολογίζεται άμεσα από την επιλεγμένη συνάρτηση ενεργοποίησης.

Τέλος, η μερική παράγωγος $\frac{\partial \mathsf{z}_{\mathsf{j}}^{[\mathsf{L}]}}{\partial w_{jk}^{[L]}}$ υπολογίζεται λαμβάνοντας την παράγωγο του γραμμικού συνδυασμού

- Η παράγωγος της συνάρτησης απώλειας ως προς τα δυναμικά πόλωσης είναι:

$$\frac{\partial \mathsf{L}(\overline{\mathsf{W}},\overline{\mathsf{b}})}{\partial b_j^{[L]}} = \frac{\partial \mathsf{z}_{\mathsf{j}}^{[\mathsf{L}]}}{\partial b_j^{[L]}} * \frac{\partial \mathsf{a}_{\mathsf{j}}^{[\mathsf{L}]}}{\partial z_j^{[L]}} * \frac{\partial \mathsf{L}(\overline{\mathsf{W}},\overline{\mathsf{b}})}{\partial a_j^{[L]}} \tag{2.19}$$

Στην περίπτωση αυτή, η μερική παράγωγος $\frac{\partial \mathsf{z}_{\mathsf{j}}^{[\mathsf{L}]}}{\partial b_j^{[L]}} = 1$.

- Τέλος, η παράγωγος της συνάρτησης απώλειας ως προς τις τιμές ενεργοποίησης του προηγούμενου επιπέδου $a_k^{[L-1]}$ είναι:

$$\frac{\partial \mathsf{L}(\overline{\mathsf{W}},\overline{\mathsf{b}})}{\partial a_k^{[L-1]}} = \sum_{j=1}^{n^{[L]}} \frac{\partial \mathsf{z}_{\mathsf{j}}^{[\mathsf{L}]}}{\partial a_k^{[L-1]}} * \frac{\partial \mathsf{a}_{\mathsf{j}}^{[\mathsf{L}]}}{\partial z_j^{[L]}} * \frac{\partial \mathsf{L}(\overline{\mathsf{W}},\overline{\mathsf{b}})}{\partial a_j^{[L]}} \tag{2.20}$$

Παρατηρούμε ότι οι μερικοί παράγωγοι των μεταβλητών ενός επιπέδου $l-1$ εξαρτώνται από το επίπεδο l. Για αυτό και όπως προαναφέραμε, οι υπολογισμοί ξεκινούν από το τελευταίο επίπεδο. Επαγωγικά, με τη χρήση της 2.20 στις 2.18 και 2.19 μπορούμε να βρούμε τις μερικές παραγώγους ως προς τις παραμέτρους όλων των επιπέδων.

Συγκεντρωτικά, χρησιμοποιώντας την αναπαράσταση με χρήση πίνακα που παρουσιάσαμε στην προηγούμενη παράγραφο, οι σχέσεις 2.18, 2.19 και 2.20 γράφονται αντίστοιχα [54]:

$$\frac{\partial \mathsf{L}(\overline{\mathsf{W}},\overline{\mathsf{b}})}{\partial \boldsymbol{W}^{[l]}} = \frac{1}{M} * \left(\frac{\partial \mathsf{L}(\overline{\mathsf{W}},\overline{\mathsf{b}})}{\partial \boldsymbol{A}^{[l]}} \odot \frac{\partial \boldsymbol{A}^{[\mathsf{l}]}}{\partial \boldsymbol{Z}^{[l]}}\right) \times {\boldsymbol{A}^{[l-1]}}^T \tag{2.21}$$

$$\frac{\partial \mathsf{L}(\overline{\mathsf{W}},\overline{\mathsf{b}})}{\partial \boldsymbol{b}^{[l]}} = \frac{1}{M} * \left(\frac{\partial \mathsf{L}(\overline{\mathsf{W}},\overline{\mathsf{b}})}{\partial \boldsymbol{A}^{[l]}} \odot \frac{\partial \boldsymbol{A}^{[\mathsf{l}]}}{\partial \boldsymbol{Z}^{[l]}}\right) \times \mathbf{1}_M^T \tag{2.22}$$

$$\frac{\partial \mathsf{L}(\overline{\mathsf{W}},\overline{\mathsf{b}})}{\partial \boldsymbol{A}^{[l-1]}} = {\boldsymbol{W}^{[l]}}^T \times \left(\frac{\partial \mathsf{L}(\overline{\mathsf{W}},\overline{\mathsf{b}})}{\partial \boldsymbol{A}^{[l]}} \odot \frac{\partial \boldsymbol{A}^{[\mathsf{l}]}}{\partial \boldsymbol{Z}^{[l]}}\right) \tag{2.23}$$

Όπου ο τελεστής $\odot$ συμβολίζει το γινόμενο στοιχείο προς στοιχείο (elementwise product), το σύμβολο $*$ το γινόμενο μονοδιάστατου αριθμού (scalar) με διάνυσμα ή πίνακα (ή τον πολλαπλασιασμό scalars μεταξύ τους) ενώ ισχύει ότι $\mathbf{1}_n = \left[1,1,1,\ldots,1\right] \in \Re^n$. Για τον όρο στην παρένθεση που συναντάται συχνά στους ανωτέρω τύπους ισχύει $\left(\frac{\partial \mathsf{L}(\overline{\mathsf{W}},\overline{\mathsf{b}})}{\partial \boldsymbol{A}^{[l]}} \odot \frac{\partial \boldsymbol{A}^{[\mathsf{l}]}}{\partial \boldsymbol{Z}^{[l]}}\right) = \frac{\partial \mathsf{L}(\overline{\mathsf{W}},\overline{\mathsf{b}})}{\partial \boldsymbol{Z}^{[l]}}$.

Σαν τελικά σχόλια σχετικά με την εκπαίδευση των νευρωνικών δικτύων είναι ωφέλιμο να κάνουμε δύο παρατηρήσεις:

- Κατά την εφαρμογή του αλγορίθμου καθόδου κλίσης σε ένα νευρωνικό δίκτυο, σε κάθε βήμα αυτού γίνονται δύο περάσματα: μια πρόσθια διάδοση που περιγράφεται από τις εξισώσεις 2.9 και 2.10 για τον υπολογισμό της συνάρτησης απώλειας και μια οπισθοδιάδοση που περιγράφεται από τις εξισώσεις 2.21, 2.22 και 2.23 για τον υπολογισμό των παραγώγων που χρησιμοποιούνται στον κανόνα ενημέρωσης.

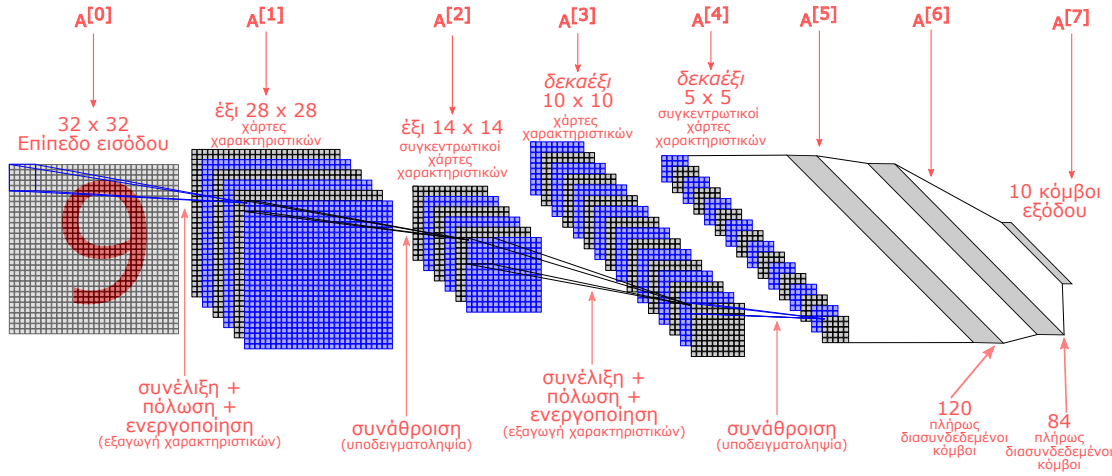
- Επειδή το σύνολο δεδομένων εισόδου μπορεί να είναι πολύ μεγάλο, αντί να λαμβάνονται όλα τα παραδείγματα M για τον υπολογισμό του $d\overline{W_{all}}$ με βάση τη συνάρτηση απώλειας, συνηθίζεται να χωρίζεται σε μικρά πακέτα (mini batches) από m παραδείγματα το καθένα. Έτσι, πραγματοποιείται ένα βήμα ενημέρωσης για κάθε μικρό πακέτο δεδομένων. Όταν εφαρμόζεται αυτή η τακτική, σιωπηρά γίνεται η υπόθεση ότι το κάθε δείγμα των m παραδειγμάτων είναι επαρκώς αντιπροσωπευτικό ώστε η συνάρτηση απώλειας υπολογισμένη στα m παραδείγματα να είναι καλή προσέγγιση της συνάρτησης υπολογισμένης στα M παραδείγματα. Ακραία μορφή αυτού είναι ο στοχαστικός αλγόριθμος καθόδου κλίσης (stochastic gradient descent) στον οποίο $m = 1$. Οι μαθηματικοί τύποι που παραθέσαμε σε αυτό το κεφάλαιο ισχύουν σε κάθε περίπτωση μετά την κατάλληλη ανάθεση της υπερπαραμέτρου M.

Συνελικτικά Νευρωνικά Δίκτυα

Έχοντας περιγράψει τη δομή και την εκπαίδευση των απλών νευρωνικών δικτύων πρόσθιας διάδοσης, εύκολα μπορούμε να κατανοήσουμε μερικές από τις παραλλαγές του. Μια από τις σημαντικότερες, είναι αυτή των Συνελικτικών Νευρωνικών Δικτύων (Convolutional Neural Networks) που χρησιμοποιείται συστηματικά στον χώρο της όρασης υπολογιστών. Πρόκειται για την υποκατηγορία των νευρωνικών δικτύων πρόσθιας διάδοσης στην οποία οδηγήθηκε η επιστημονική κοινότητα αφενός επιδιώκοντας να λύσει ορισμένα από τα πρακτικά προβλήματα της εφαρμογής νευρωνικών δικτύων στον χώρο της όρασης υπολογιστών και αφετέρου μελετώντας τη νεύρο–φυσιολογία του οπτικού φλοιού (visual cortex).

Από τη σκοπιά της νευροεπιστήμης, οι David H. Hubel και Torsten Wiesel μετά από μια σειρά πειραμάτων σε γάτες [55, 56] γύρω στο 1960 και αργότερα, σε πιθήκους [57] έριξαν φως στη δομή του οπτικού φλοιού, εμπνέοντας έτσι το κίνημα του διασυνδετισμού (connectionism). Σύμφωνα με το έργο τους, (για το οποίο τιμήθηκαν με το βραβείο nobel το 1981) πολλοί νευρώνες του οπτικού φλοιού έχουν μικρά, τοπικά πεδία υποδοχής (receptive fields) που μπορεί να επικαλύπτονται μεταξύ τους. Πιο συγκεκριμένα, ο κάθε νευρώνας αφορά ένα περιορισμένο τμήμα του οπτικού πεδίου αλλά όλοι μαζί, καλύπτουν το σύνολό του. Επιπλέον, μετά από πειράματα οπτικής αναγνώρισης σχημάτων (ορθογώνιο παραλληλόγραμμο σε μορφή μπάρας) σε διάφορες γεωμετρίες παρατηρήθηκε ότι διαφορετικοί νευρώνες με το ίδιο πεδίο υποδοχής ενεργοποιούνται ανάλογα με τη γεωμετρία του σχήματος (κάποιοι νευρώνες ενεργοποιούνται κατά τον κάθετο προσανατολισμό της μπάρας ενώ άλλοι με τον οριζόντιο προσανατολισμό της). Τέλος, επισήμαναν ότι ορισμένοι νευρώνες ενεργοποιούνται με την αναγνώριση πιο περίπλοκων μοτίβων όπως προκύπτουν από τη σύνθεση απλών γεωμετριών χαμηλότερου επιπέδου [51].

Από πρακτικής σκοπιάς, η υπολογιστική πολυπλοκότητα που προκύπτει από την τροφοδότηση ενός πλήρως διασυνδεδεμένου νευρωνικού δικτύου με εικόνες είναι απαγορευτική. Για παράδειγμα, έστω ότι διατίθεται ένα σύνολο από ασπρόμαυρες εικόνες μεγέθους 100×100 εικονοστοιχεία. Αυτό συνεπάγεται ότι το επίπεδο εισόδου θα διαθέτει τόσους κόμβους όσα είναι και τα χαρακτηριστικά (τα εικονοστοιχεία), δηλαδή 10,000[15]. Στην κλασική περίπτωση, ο αριθμός των κόμβων του πρώτου κρυφού επιπέδου θα είναι περίπου ίσος με τον αριθμό των χαρακτηριστικών εισόδου, δηλαδή πάλι 10,000. Αυτό σημαίνει ότι μόνο στο πρώτο επίπεδο του νευρωνικού δικτύου θα υπήρχαν $10,000 \times 10,000$ βάρη και $10,000$ δυναμικά πόλωσης, ένας αριθμός, πολύ μεγάλος. Θα μπορούσαμε, φυσικά, αντί να τροφοδοτήσουμε το νευρωνικό δίκτυο με ολόκληρη την εικόνα σε ακατέργαστη μορφή, να εξάγαμε με ντετερμινιστικό τρόπο ορισμένα χαρακτηριστικά ώστε να καταλήξουμε με ένα μικρότερο διάνυσμα χαρακτηριστικών που θα εσωκλείει όλη τη χρήσιμη πληροφορία. Μια τέτοια διαδικασία χειροναχτικής εξαγωγής χαρακτηριστικών απεικονίζεται στο σχήμα 2.2. Παρόλα αυτά, για τους λόγους που αναφέραμε στην ενότητα 2.1.2 θα επιθυμούσαμε την αυτοματοποιημένη εκμάθησή χαρακτηριστικών.



Σχήμα 2.7: Αρχιτεκτονική Συνελικτικού Νευρωνικού Δικτύου (LeNet-5) [58]. Η αναπαράστασή τους διαφέρει από αυτήν των απλών νευρωνικών δικτύων αφού εδώ δίνεται έμφαση στους πίνακες τιμών ενεργοποίησης $A^{[l]}$. Οι χάρτες χαρακτηριστικών αναπαριστώνται με τετράγωνα ενώ τα βάρη με ακμές. Τα δύο τελευταία επίπεδα είναι πλήρως διασυνδεδεμένα. *Παράχθηκε από το Inkscape τροποποιώντας αυτήν την εικόνα.*

Η λύση στο πρόβλημα δόθηκε μέσω της αξιοποίησης της τοπικής χωρικής συνεκτικότητας (local spacial coherence) και της ιεραρχικής δομής (hierarchical structure) των δεδομένων εικόνων. Εμπνευσμένη από τις ανωτέρω επιστημονικές παρατηρήσεις, ενσωματώθηκε η γνώση του χώρου προβλημάτων με εικόνες στη δομή των νευρωνικών δικτύων οδηγώντας έτσι στη δημιουργία των συνελικτικών νευρωνικών δικτύων (βλ. σχήμα 2.7). Οι δομικές διαφορές των συνελικτικών νευρωνικών δικτύων που τους διακρίνουν από τα νευρωνικά δίκτυα που παρουσιάστηκαν στην προηγούμενη ενότητα μπορούν να συνοψιστούν ως εξής:

- Μια πρώτη διαφορά που επιλύει το πρόβλημα της απαγορευτικής υπολογιστικής πολυπλοκότητας έγκειται στο τρόπο διασύνδεσης των κόμβων ενός επιπέδου με τους κόμβους του

[15]Χρησιμοποιούμε τελεία για τη διάκριση των δεκαδικών (decimal point separator) και υποδιαστολή για τη διάκριση των χιλιάδων.

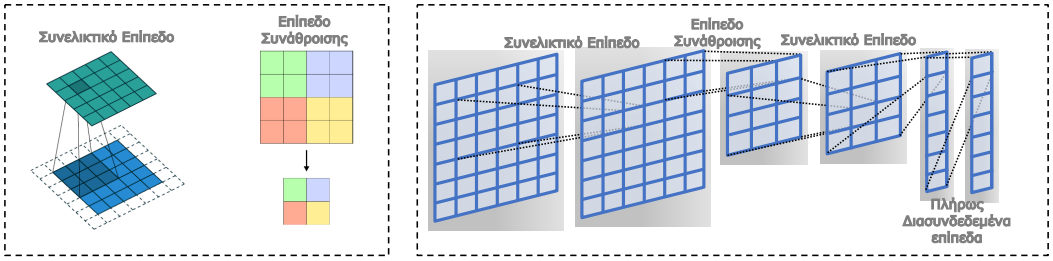
αμέσως προηγούμενου. Αντί να είναι πλήρως διασυνδεδεμένοι με αυτούς του προηγούμενου επιπέδου όπως στην περίπτωση των απλών νευρωνικών δικτύων, ενώνονται με βάρη μόνο με αυτούς που ανήκουν στο λεγόμενο πεδίο υποδοχής. Με άλλα λόγια, κάθε νευρώνας επιπέδου l δέχεται σαν είσοδο ένα διαφορετικό και περιορισμένο τμήμα του πίνακα $A^{[l-1]}$.

- Στα συνελικτικά νευρωνικά δίκτυα, οι κόμβοι του κάθε επιπέδου είναι οργανωμένοι σε όγκους τριών διαστάσεων με πλάτος, ύψος και βάθος. Με αυτόν τον τρόπο, διατηρείται η τοπική χωρική συνεκτικότητα. Αναλυτικότερα, οι κόμβοι εισόδου, για παράδειγμα, οργανώνονται όπως τα εικονοστοιχεία σε μια εικόνα: το βάθος του επιπέδου αντιστοιχεί στον αριθμό των καναλιών της εικόνας (π.χ. RGB) ενώ το ύψος και το πλάτος του επιπέδου στο ύψος και πλάτος της εικόνας. Έτσι, το νευρωνικό δίκτυο έχει τη δυνατότητα να αντλήσει εύκολα πληροφορία από μια χωρική γειτονιά της εικόνας (το πεδίο υποδοχής κάποιου νευρώνα) αφού οι αποστάσεις μεταξύ των εικονοστοιχείων διατηρούνται αναλλοίωτες. Αν όμως λαμβάναμε την εικόνα και την αναπτύσσαμε σε μια διάσταση (flatten) δημιουργώντας ένα μεγάλο διάνυσμα, τότε οι σχετικές αποστάσεις των στοιχείων εισόδου δε θα διατηρούνταν. Ανάλογες παρατηρήσεις ισχύουν και για τα κρυφά επίπεδα. Δηλαδή, και στα επόμενα επίπεδα οι κόμβοι οργανώνονται σε τρισδιάστατες δομές οι οποίες διατηρούν τοπικό χαρακτήρα. Η διαφορά έγκειται στο ότι η τιμή των κόμβων των κρυφών επιπέδων δεν είναι η τιμή του εκάστοτε εικονοστοιχείου στη θέση αυτή. Αντίθετα, είναι η τιμή ενός (σύνθετου) χαρακτηριστικού της περιοχής που έχει υπολογιστεί από την επεξεργασία απλούστερων χαρακτηριστικών προηγούμενων επιπέδων. Σχετικά με την ιδιότητα της ιεραρχικής δομής των εικόνων του πραγματικού κόσμου, αυτή αξιοποιείται μέσω διαδοχικών κρυφών επιπέδων που σταδιακά διευρύνουν το οπτικό πεδίο και συνθέτουν ολοένα και πιο σύνθετα χαρακτηριστικά. Έχει δειχθεί, ότι το σύστημα μαθαίνει να εξάγει μέσω των πρώτων επιπέδων απλά χαρακτηριστικά (π.χ. οριζόντιες και κάθετες ακμές) τα οποία σε επόμενα επίπεδα συνδυάζει για να εξάγει πιο περίπλοκα χαρακτηριστικά [59]. Αφού οι κόμβοι είναι οργανωμένοι σε όγκους, προκύπτει φυσικά ότι το διάνυσμα ενεργοποίησης $A^{[l]}$ του κάθε επιπέδου l που κατασκευάζεται από την έξοδο κάθε κόμβου έχει τη μορφή πίνακα τριών διαστάσεων. Διαισθητικά για τα κρυφά επίπεδα, το ύψος και το πλάτος του διανύσματος ενεργοποίησης κωδικοποιούν αμυδρά τη θέση του χαρακτηριστικού στην εικόνα ενώ το βάθος κωδικοποιεί τα διάφορα χαρακτηριστικά (π.χ. βάθος 1: οριζόντιες ακμές, βάθος δύο: κατακόρυφες ακμές[16]). Στην περίπτωση των δικτύων που εξετάζουμε, το $A^{[l]}$ λέμε ότι αποτελείται από επίπεδα φύλλα τα οποία στοιβάζονται στη διάσταση z και ονομάζονται χάρτες χαρακτηριστικών (feature maps).

- Μια ακόμα δομική διαφορά είναι ότι στα συνελικτικά νευρωνικά δίκτυα η εκμάθηση και εξαγωγή των χαρακτηριστικών δε γίνεται ανεξάρτητα σε κάθε περιοχή της εικόνας. Νευρώνες των οποίων τα βάρη προσαρμόζονται ώστε να αναγνωρίζουν και να εξάγουν γενικά χαρακτηριστικά της εικόνας όπως τα χαρακτηριστικά ακμών θα ήταν ασύμφορο να είχαν εφαρμογή μόνο στο πεδίο υποδοχής τους και όχι σε όλη την εικόνα. Έτσι, οδηγούμαστε στην έννοια του διαμοιρασμού παραμέτρων (weight sharing). Σύμφωνα με αυτήν την έννοια, αντί οι κόμβοι ενός επιπέδου να είναι διασυνδεδεμένοι με τους κόμβους του προηγούμενου

[16]Πρακτικά στη διαδικασία εκμάθησης χαρακτηριστικών είναι δύσκολο να εκφράσουμε με σαφήνεια τι αναπαριστά το καθένα.

επιπέδου με ξεχωριστά βάρη και δυναμικά πόλωσης, οι παράμετροι αυτές μοιράζονται μεταξύ των κόμβων. Έτσι, οι κόμβοι ενός επιπέδου επιτελούν τον ίδιο γραμμικό συνδυασμό $y = f\left(\sum_{i=1}^{n} w_i * x_i + b\right)$ αλλά με διαφορετικό διάνυσμα εισόδου X που εξαρτάται από το οπτικό πεδίο. Σημειώνουμε δε ότι ο διαμοιρασμός βαρών θα ήταν δύσκολο να εφαρμοστεί στην περίπτωση που η είσοδος αποτελούνταν από δομημένα δεδομένα καθώς αυτά μπορεί να είχαν πλήρως ετερογενή χαρακτηριστικά.

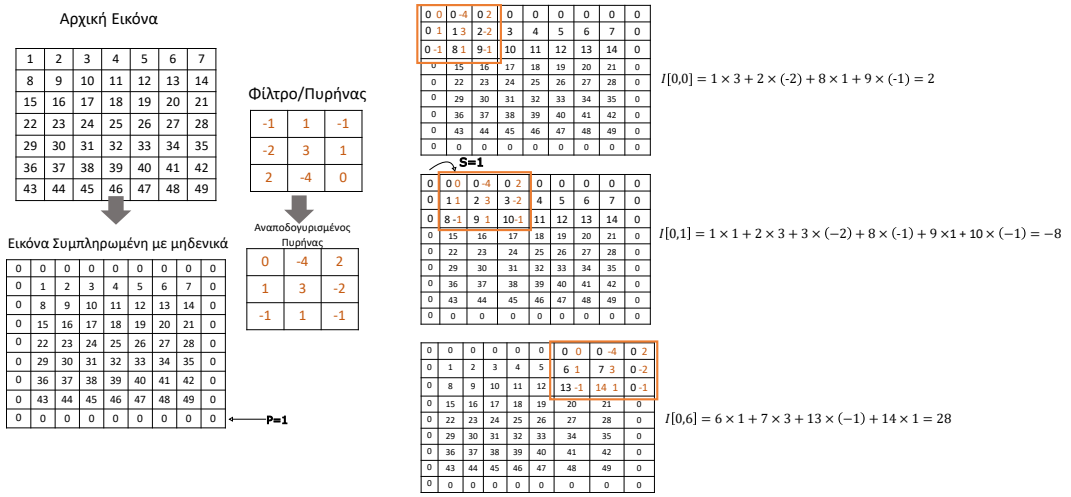


Σχήμα 2.8: Μεμονωμένο συνελικτικό επίπεδο και επίπεδο υποδειγματοληψίας (αριστερά). Συνδυασμός των επιπέδων για την κατασκευή ενός συνελικτικού νευρωνικού δικτύου (δεξιά). *Παράχθηκε από το Inkscape τροποποιώντας αυτήν την εικόνα.*

Πρακτικά, αν εξαιρέσουμε τα τελευταία, πλήρως διασυνδεδεμένα επίπεδα ενός συνελικτικού νευρωνικού δικτύου, οι ανωτέρω δομικές διαφορές υλοποιούνται με τη χρήση αφενός των συνελικτικών επιπέδων και αφετέρου των επιπέδων υποδειγματοληψίας. Αναφορικά με τα πρώτα, η εσωτερική τους λειτουργία απεικονίζεται στο αριστερό τμήμα του σχήματος 2.8. Το οπτικό πεδίο αναπαρίσταται με ένα σκούρο παραλληλόγραμμο επάνω στον χάρτη χαρακτηριστικών του προηγούμενου επιπέδου. Τα βάρη, είναι ευκολότερο να τα φανταστεί κανείς σαν ένα παραλληλόγραμμο (ή ένα ορθογωνικό κυβοειδές σε τρεις διαστάσεις) το οποίο έχει τις ίδιες διαστάσεις με το οπτικό πεδίο πάνω στο προηγούμενο επίπεδο. Η τιμή ενεργοποίησης κάθε στοιχείου του τρισδιάστατου πίνακα $A^{[l]}$ υπολογίζεται ως το αποτέλεσμα της εφαρμογής της συνάρτησης ενεργοποίησης στον γραμμικό συνδυασμό των στοιχείων του πίνακα $A^{[l-1]}$ που βρίσκονται εντός του οπτικού πεδίου με βάρη τα στοιχεία του $W^{[l]}$ και το δυναμικό πόλωσης $b^{[l]}$. Στην ουσία, τα βάρη υπερτίθενται στο οπτικό πεδίο και επιτελείται γινόμενο μεταξύ των πινάκων στοιχείο προς στοιχείο (elementwise product). Αν υπήρχε ένας πίνακας από βάρη για κάθε στοιχείο του $A^{[l]}$, τότε δε θα είχαμε διαμοιρασμό βαρών. Αντιθέτως, ο διαμοιρασμός βαρών έγκειται στην ολίσθηση αυτού του παραλληλόγραμμου (ή ορθογωνικού κυβοειδούς στις τρεις διαστάσεις) στο ύψος και πλάτος της εικόνας, όπως φαίνεται και στο σχήμα 2.9 για την περίπτωση των δύο διαστάσεων. Για την περίπτωση που έχουμε πολλούς χάρτες χαρακτηριστικών, παραπέμπουμε τον αναγνώστη στο σχήμα 2.10. Αυτή η διαδικασία της ολίσθησης του πίνακα βαρών ονομάζεται δισδιάστατη συνέλιξη[17], ενώ το κυλιόμενο παράθυρο ονομάζεται και φίλτρο (filter) ή πυρήνας (kernel).

Με κάθε συνελικτικό επίπεδο εισάγεται μια σειρά από παραμέτρους πέρα από αυτές που υπήρχαν σε κάθε απλό νευρωνικό δίκτυο. Για κάθε επίπεδο l, μεταξύ δύο στοιβαγμένων συνόλων από χάρτες χαρακτηριστικών $A^{[l-1]}$ και $A^{[l]}$ με διαστάσεις $Width^{[l-1]} \times Height^{[l-1]} \times Depth^{[l-1]}$ και $Width^{[l]} \times Height^{[l]} \times Depth^{[l]}$ αντίστοιχα πρέπει να ορίσουμε:

[17] Τυπικά, η πράξη ονομάζεται διασταυρούμενη συσχέτιση (cross correlation) και είναι ίδια με τη συνέλιξη αν στην πρώτη περίπτωση αναποδογυρίσουμε τον πυρήνα (γυρνώντας τον ως προς την κύρια και δευτερεύουσα διαγώνιο).



Σχήμα 2.9: Παράδειγμα συνέλιξης σε δύο διαστάσεις *Παράχθηκε από το Inkscape τροποποιώντας και διορθώνοντας αυτήν την εικόνα.*

- Το μέγεθος του κυλιόμενου πυρήνα (ή φίλτρου). Αυτό εισαγάγει τις παραμέτρους $F_x{}^{[l]}$ και $F_y{}^{[l]}$. Το βάθος του φίλτρου δεν αποτελεί υπερπαράμετρο και είναι ίσο με $Depth^{[l-1]}$.

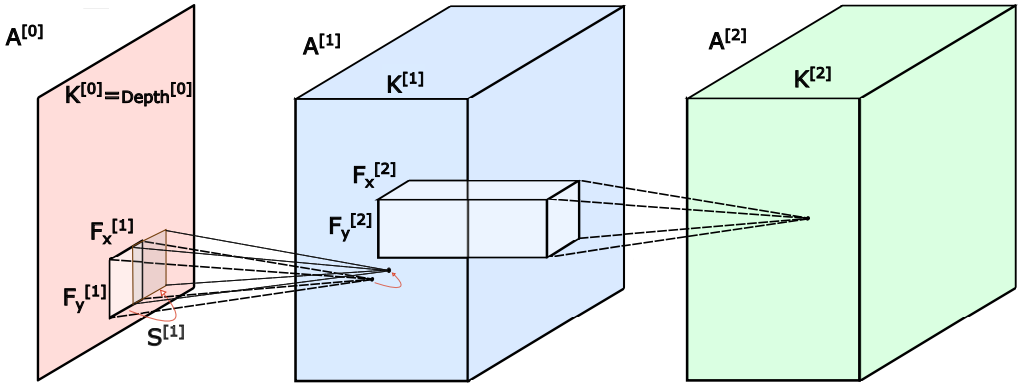
- Ο αριθμός των φίλτρων, $K^{[l]}$. Όπως έχει γίνει σαφές από τα ανωτέρω σχήματα, το κάθε φίλτρο $k \in [1, K^{[l]}]$ λαμβάνει σαν είσοδο έναν όγκο από χαρακτηριστικά και έχει ως έξοδο έναν αριθμό $a \in \Re$. Για να έχει ο παραγόμενος χάρτης χαρακτηριστικών βάθος, θα πρέπει να γίνει χρήση πολλαπλών φίλτρων που θα εξάγουν διαφορετικά χαρακτηριστικά. Έτσι, μπορούμε να καθορίσουμε το $Depth^{[l]}$ θέτοντας το $K^{[l]}$ αφού ισχύει ότι $K^{[l]} = Depth^{[l]}$. Για κάθε ένα φίλτρο παράγεται ένας χάρτης χαρακτηριστικών.

- Το βήμα της συνέλιξης, $s_x^{[l]}$ κατά τον x άξονα και $s_y^{[l]}$ κατά τον y άξονα (συνήθως, οι δύο ποσότητες είναι ίσες και συμβολίζονται ως $s^{[l]}$). Αυτή η ποσότητα καθορίζει το βήμα της ολίσθησης του φίλτρου πάνω στο χάρτη χαρακτηριστικών του επιπέδου $l-1$. Για παράδειγμα, μετά την εφαρμογή του στο άνω αριστερά άκρο της εισόδου στο σχήμα 2.9 θα μετακινηθεί με βήμα 1 μια θέση δεξιά (ή κάτω, αργότερα) για να υπολογίσει την επόμενη έξοδο. Στην περίπτωση που η τιμή του βήματος είναι ίση με το μέγεθος του φίλτρου (στον x ή y άξονα), τότε δεν υπάρχει επικάλυψη μεταξύ των οπτικών πεδίων. Μεγαλύτερη τιμή από αυτή δεν είναι επιθυμητή καθώς οδηγεί σε συστηματική αγνόηση στοιχείων του $A^{[l-1]}$.

- Το ποσό της περιμετρικής συμπλήρωσης της εισόδου $A^{[l-1]}$ με μηδενικά[18], $P_x^{[l]}$ και $P_y^{[l]}$. Αν και κάτι τέτοιο δεν είναι απαραίτητο, συνήθως γίνεται για τον έλεγχο των διαστάσεων ύψους και πλάτους του χάρτη χαρακτηριστικών της εξόδου (ή των χαρτών χαρακτηριστικών της εξόδου αν $K^{[l]} > 1$). Μετά από αυτήν τη διαδικασία, καταλήγουμε με ένα σύνολο χαρτών χαρακτηριστικών $\grave{A}^{[l-1]}$ του οποίου οι νέες διαστάσεις είναι $\grave{Width}^{[l-1]} = Width^{[l-1]} + 2 \times P_x^{[l]}$ και $\grave{Height}^{[l-1]} = Height^{[l-1]} + 2 \times P_y^{[l]}$

Συγκεντρωτικά, το συνελικτικό επίπεδο, για την παραγωγή ενός συνόλου χαρτών χαρακτηριστικών εφαρμόζει τις εξής διαδικασίες:

[18] Υπάρχουν και άλλες επιλογές στον τρόπο περιμετρικής συμπλήρωσης, όπως αυτός που περιλαμβάνει ανάκλαση αλλά η χρήση μηδενικών είναι η πιο συνηθισμένη μέθοδος και για αυτό εστιάζουμε σε αυτή.



Σχήμα 2.10: Συνελικτικά επίπεδα στη σειρά με έμφαση στην περίπτωση όπου οι πίνακες A είναι τριών διαστάσεων. Στο σχήμα φαίνονται οι υπερπαράμετροι των συνελικτικών επιπέδων. *Παράχθηκε από το Inkscape τροποποιώντας αυτήν την εικόνα.*

1. Συνέλιξη πάνω στους χάρτες χαρακτηριστικών $\grave{A}^{[l-1]}$ (αφού έχουν ίσως συμπληρωθεί με μηδενικά) με τον (τρισδιάστατο) πίνακα βαρών $W_k^{[l]}$.

2. Σημειακή πρόσθεση της τιμής πόλωσης ($b_k^{[l]}$) σε κάθε στοιχείο του προηγούμενου πίνακα με αποτέλεσμα την παραγωγή ενός πίνακα $Z_k^{[l]}$ με τις ίδιες διαστάσεις.

3. Εφαρμογή συνάρτησης ενεργοποίησης $F^{[l]}$ σημειακά ώστε τελικά να παραχθεί ο χάρτης χαρακτηριστικών $A_k^{[l]}$ με διαστάσεις ίδιες με τον $Z_k^{[l]}$, δηλαδή:

$$Width^{[l]} = \left\lfloor \frac{Width^{[l-1]} - {F_x}^{[l]} + 2 \times P_x^{[l]}}{S^{[l]}} \right\rfloor + 1, \tag{2.24}$$

και

$$Height^{[l]} = \left\lfloor \frac{Height^{[l-1]} - {F_y}^{[l]} + 2 \times P_y^{[l]}}{S^{[l]}} \right\rfloor + 1 \tag{2.25}$$

4. Επανάληψη από το βήμα 1 $K^{[l]}$ φορές, όσο και το βάθος του $A^{[l]}$.

5. Στοίβαξη των παραχθέντων χαρτών χαρακτηριστικών ως προς τον άξονα z ώστε να κατασκευαστεί ο τρισδιάστατος πίνακας $A^{[l]}$. Τελικά, το σύνολο των χαρτών χαρακτηριστικών $A^{[l]}$ έχει διαστάσεις μήκους και πλάτους ίδιες με αυτές του $A_k^{[l]}$ αλλά το βάθος τώρα, αντί για μονάδα είναι:

$$Depth^{[l]} = K^{[l]} \tag{2.26}$$

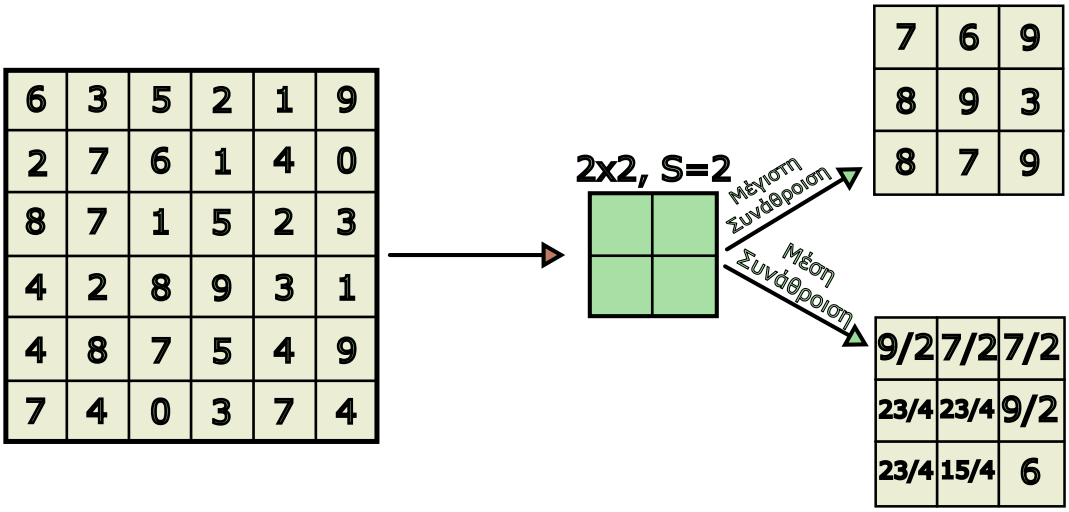
Με μαθηματικούς όρους, οι υπολογισμοί που εκτελούνται σε ένα συνελικτικό επίπεδο είναι οι εξής:

$$Z_k^{[l]} = {W_k^{[l]}}^{T^\tau} \underset{step=S^{[l]}}{\star} A^{[l-1]} + b_k^{[l]} \tag{2.27}$$

και

$$A_k^{[l]} = F^{[l]}(Z_k^{[l]}). \tag{2.28}$$

Όπου το σύμβολο τ στον εκθέτη ενός πίνακα δηλώνει την αναστροφή του πίνακα υπό τη δευτερεύουσα διαγώνιο ενώ το σύμβολο $\underset{step=S^{[l]}}{\star}$ δηλώνει τη συνέλιξη με βήμα ίσο με $S^{[l]}$.



Σχήμα 2.11: Σχήμα επιπέδου συνάθροισης. *Παράχθηκε από το Inkscape.*

Ένα δεύτερο είδος επιπέδου αποτελεί αυτό της υποδειγματοληψίας (ή συνάθροισης) όπως φαίνεται στο σχήμα 2.11. Το συγκεκριμένο είδος δε διαθέτει καμία παράμετρο αφού η μόνη λειτουργία του είναι να πραγματοποιεί υποδειγματοληψία στο χάρτη χαρακτηριστικών. Ο τρόπος εφαρμογής του είναι παρόμοιος με αυτόν του συνελικτικού επιπέδου. Δηλαδή, και πάλι υπάρχει ένα κυλιόμενο παράθυρο πάνω στον χάρτη χαρακτηριστικών το οποίο συναθροίζει τα στοιχεία στα οποία υπερτίθεται σε ένα στοιχείο υπό μια προκαθορισμένη στρατηγική. Πιο συγκεκριμένα, ανάλογα με το αν επιλέγεται σαν έξοδος το μεγαλύτερο στοιχείο στη γειτονιά συνάθροισης (το οπτικό πεδίο) ή μια μέση τιμή αυτών, έχουμε τη μέγιστη συνάθροιση (max pooling) ή τη μέση συνάθροιση (average pooling) αντίστοιχα. Σε κάθε περίπτωση, μια διαφορά με τα συνελικτικά επίπεδα είναι ότι το κυλιόμενο παράθυρο υπερτίθεται σε κάθε «φύλλο» της εισόδου $A^{[l-1]}$ ξεχωριστά (δηλαδή, σε κάθε $A_k^{[l-1]}$). Με άλλα λόγια, παρόλο που ο πίνακας $A^{[l-1]}$ μπορεί να είναι τρισδιάστατος και να αποτελείται από πολλούς χάρτες χαρακτηριστικών στοιβαγμένους στον z άξονα, η γειτονιά συνάθροισης θα είναι πάντα ένα δισδιάστατο παράθυρο. Τέλος, να σημειώσουμε ότι αυτό το επίπεδο συμβάλλει στην ευρωστία του συστήματος καθιστώντας τις τιμές ενεργοποίησης των επόμενων επιπέδων αμετάβλητες σε μικρές διακυμάνσεις της θέσης των αντικειμένων, μια ιδιότητα που θα αναλύσουμε περαιτέρω στην επόμενη ενότητα.

Σε κάθε επίπεδο συνάθροισης l έχουμε τις εξής υπερπαραμέτρους:

- Το μέγεθος του πυρήνα υποδειγματοληψίας ${Pk_x}^{[l]}$ και ${Pk_y}^{[l]}$. Καθορίζει τη γειτονιά συνάθροισης αλλά και το μέγεθος του συναθροισμένου χάρτη χαρακτηριστικών.

- Τη στρατηγική του επιπέδου συνάθροισης. Όπως αναφέραμε, εδώ οι στρατηγικές είναι δύο: μέγιστη συνάθροιση και μέση συνάθροιση.

- Το βηματισμό του κυλιόμενου παραθύρου $s_x^{[l]}$ κατά τον x άξονα και $s_y^{[l]}$ κατά τον y άξονα (όπως και στα συνελικτικά επίπεδα, συνήθως, οι δύο ποσότητες είναι ίσες και συμβολίζονται ως $s^{[l]}$). Είθισται, το βήμα να ισούται με το μέγεθος του πυρήνα.

Αναφορικά με το μέγεθος της εξόδου ενός επιπέδου συνάθροισης l, με είσοδο έναν χάρτη χαρακτηριστικών $A^{[l-1]}$ με διαστάσεις $Width^{[l-1]} \times Height^{[l-1]} \times Depth^{[l-1]}$ ισχύει:

$$Width^{[l]} = \frac{Width^{[l-1]} - Pk_x{}^{[l]}}{S^{[l]}} + 1, \tag{2.29}$$

$$Height^{[l]} = \frac{Height^{[l-1]} - Pk_y{}^{[l]}}{S^{[l]}} + 1 \tag{2.30}$$

και

$$Depth^{[l]} = Depth^{[l-1]} \tag{2.31}$$

Σύγχρονες Αρχιτεκτονικές Βαθιών Νευρωνικών Δικτύων

Αρχιτεκτονικές βαθιών νευρωνικών δικτύων έχουν υλοποιηθεί και χρησιμοποιηθεί σε διάφορες εφαρμογές από μέλη του Εργαστηρίου Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης (Artificial Intelligence and Learning Systems Laboratory) του ΕΜΠ. Ειδικότερα, έχουν αναπτυχθεί τεχνικές CNN και CNN-RNN [60–62], ενώ έμφαση έχει δοθεί στη διαφάνεια (transparency) και στην προσαρμογή των μοντέλων [63–65]. Επιπλέον, έμφαση έχει δοθεί και στην ανάπτυξη πλέον σύνθετων αρχιτεκτονικών, Μπαϋεσιανών με αβεβαιότητα και βαθιών τρισδιάστατων νευρωνικών δικτύων για αναγνώριση και σύνθεση συναισθήματος [66–70], αλλά και σε άλλες αρχιτεκτονικές οι οποίες εφαρμόζονται σε προβλήματα ανάλυσης σημάτων, εικόνων και αλληλεπίδρασης ανθρώπου-υπολογιστή [71–74].

Έχοντας καλύψει πλήρως τα νευρωνικά δίκτυα και την υποκατηγορία τους η οποία χρησιμοποιείται στην όραση υπολογιστών, είμαστε σε θέση να περιγράψουμε ένα νεότερο είδος νευρωνικών δικτύων για τον ίδιο σκοπό, τα λεγόμενα νευρωνικά δίκτυα με κάψουλες.

2.2 Νευρωνικά Δίκτυα με Κάψουλες

Τα τελευταία χρόνια, διερευνάται μια ακόμα παραλλαγή των νευρωνικών δικτύων για εφαρμογές όρασης υπολογιστών: αυτή των νευρωνικών δικτύων με κάψουλες (capsule networks). Η ιδέα πίσω από τη νέα αρχιτεκτονική παρουσιάστηκε από τον Geoffrey Hinton, το ίδιο άτομο που είχε συμβάλλει καθοριστικά στην ανάπτυξη και εδραίωση των συνελικτικών δικτύων [43]. Αυτή τη φορά όμως, στα σχετικά έργα του [47, 75, 76] τονίζει ορισμένες αδυναμίες της εδραιωμένης, πλέον, τεχνολογίας ενώ προτείνει μια νέα αρχιτεκτονική που θα τις αντιμετωπίζει. Ένας έμπειρος αναγνώστης μπορεί να επισημάνει ότι η σύλληψη της ιδέας των νευρωνικών δικτύων με κάψουλες δεν είναι νέα (2011). Παρόλα αυτά, όπως θα διαπιστώσουμε στο κεφάλαιο 3, μόλις πρόσφατα άρχισε να λαμβάνει πρακτική υπόσταση με την ανάπτυξή σύνθετων αρχιτεκτονικών που την πραγματώνουν.

Στην ενότητα αυτή θα ξεκινήσουμε κάνοντας αναφορά σε ορισμένα στοιχεία του ανθρώπινου μηχανισμού αναγνώρισης προτύπων εικόνων που αποτέλεσαν πηγή έμπνευσης για τα νευρωνικά δίκτυα με κάψουλες. Έπειτα, θα διατυπώσουμε τα ισχυρά και αδύναμα σημεία που παρουσιάζουν τα συνελικτικά νευρωνικά δίκτυα της προηγούμενης ενότητας. Τέλος, βασιζόμενοι στα κύρια έργα του G. Hinton σχετικά με τα νευρωνικά δίκτυα με κάψουλες στο πλαίσιο επιβλεπόμενης

μάθησης [47, 75, 76], θα εμβαθύνουμε στις αρχές λειτουργίας τους.

2.2.1 Στοιχεία Έμπνευσης των Νευρωνικών Δικτύων με Κάψουλες

Για άλλη μια φορά, πηγή έμπνευσης για αυτήν την υποκατηγορία των νευρωνικών δικτύων με την οποία θα ασχοληθούμε σε μεγάλο βαθμό στην υπόλοιπη έκταση της εργασίας αποτέλεσε η νευροφυσιολογία. Πιο αναλυτικά, όπως έχουμε αναφέρει και στην ενότητα 1.2, οι νευρώνες στον εγκέφαλο οργανώνονται σε ολοένα και μεγαλύτερες δομές ανάλογα με τη λειτουργία τους. Σε γενικές γραμμές, γειτονικοί νευρώνες που επιτελούν παρόμοιες λειτουργίες ενισχύουν τις μεταξύ τους συνδέσεις σχηματίζοντας συστάδες[19]. Προκύπτει λοιπόν η διάθεση πειραματισμού για τη σχεδίαση μιας αρχιτεκτονικής νευρωνικών δικτύων που θα εμπεριέχει ρητά συστάδες από νευρώνες[20]. Αυτές τις συστάδες θα τις ονομάζουμε και κάψουλες (capsules).

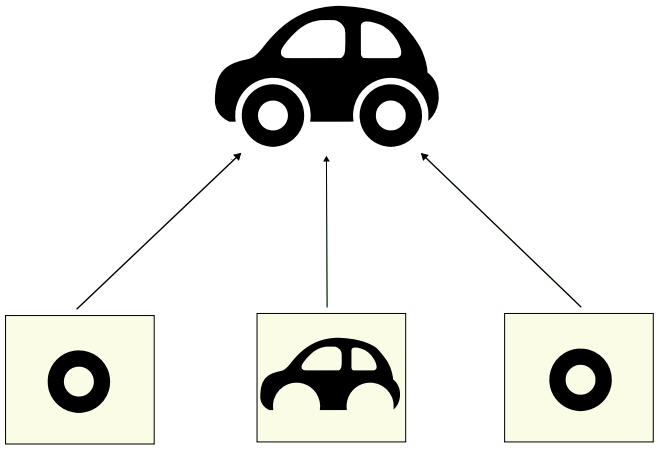
Επιπρόσθετα, έχει παρατηρηθεί ότι ο άνθρωπος αναγνωρίζει ένα αντικείμενο δημιουργώντας δυναμικά ένα ιεραρχικό δέντρο του οποίου η ρίζα εμπεριέχει το αντικείμενο προς αναγνώριση —υπό μια κωδικοποιημένη αναπαράσταση—ενώ τα κλαδιά τα επιμέρους τμήματα (ή χαρακτηριστικά) από τα οποία απαρτίζεται. Εκτός αυτού, θα μπορούσαμε να ισχυριστούμε ότι η ιεραρχική δομή είναι «εμπλουτισμένη», υπό την έννοια ότι τα κλαδιά του δέντρου κωδικοποιούν τη σχετική θέση των επιμέρους τμημάτων [77]. Αυτό προκύπτει από το γεγονός ότι ο άνθρωπος, με την αναγνώριση ενός αντικειμένου είναι πάντα σε θέση να προσδιορίσει τη σχετική θέση των μερών του (βλ. σχήμα 2.12). Η ιεραρχική δομή μεταξύ των αντικειμένων και των αποτελούμενων μερών του φαίνεται να υπάρχει παντού στη φύση. Είναι λογική λοιπόν η επιδίωξη ρητής ενσωμάτωσης μηχανισμών στα νευρωνικά δίκτυα που θα αξιοποιούν την πρότερη γνώση σχετικά με την εμπλουτισμένη ιεραρχική δομή των αντικειμένων του φυσικού κόσμου[21].

Το τρίτο και ίσως πιο σημαντικό στοιχείο από το οποίο εμπνεύστηκαν τα νευρωνικά δίκτυα με κάψουλες προκύπτει από την παρατήρηση ότι οι άνθρωποι πάντα εφαρμόζουν ένα σύστημα συντεταμένων στα αντικείμενα που αναγνωρίζουν. Με άλλα λόγια, η αναγνώριση ενός αντικειμένου είναι άρρηκτα διασυνδεδεμένη με την αναγνώριση της γεωμετρίας του αντικειμένου. Για παράδειγμα, με τη θωριά ενός αυτοκινήτου αντιλαμβανόμαστε άμεσα και τον προσανατολισμό του. Μάλιστα, όπως φαίνεται στο σχήμα 2.13 ο τρόπος με τον οποίο εφαρμόζεται το σύστημα συντεταγμένων σε μια εικόνα διαδραματίζει πρωτεύοντα ρόλο στην κατανόησή της. Αυτή η λειτουργία του ανθρώπινου οπτικού φλοιού μας προδιαθέτει να δοκιμάσουμε στα τεχνητά νευρωνικά δίκτυα τη ρητή εκμάθηση ενός συστήματος αναφοράς για κάθε αντικείμενο που καλούνται να αναγνωρίσουν και τη σύγκριση κάθε νέας εικόνας εισόδου με αυτό. Όπως θα δούμε στη συνέχεια, μια τέτοια μέθοδος θα οδηγήσει το νευρωνικό δίκτυο σε αποδοτικότερη γενίκευση σχετικά με την

[19] Παράδειγμα συστάδων με νευρώνες στον άνθρωπο που διαθέτουν κοινή είσοδο και κοινή έξοδο είναι η φλοιική μικρή στήλη (cortical minicolumn)

[20] Στις μέχρι τώρα αρχιτεκτονικές πλήρως διασυνδεδεμένων νευρωνικών δικτύων που έχουμε παρουσιάσει, δεν υπάρχουν τέτοιες δομές. Θα μπορούσαμε να υποθέσουμε ότι κάθε επίπεδο από νευρώνες αποτελεί μια τέτοια οργανωτική δομή. Η υπόθεση αυτή όμως δεν είναι πλήρως ευσταθής καθότι εσωτερικά αυτής οι νευρώνες δεν αλληλεπιδρούν άμεσα μεταξύ τους.

[21] Μπορεί να ισχυριστεί κανείς ότι κάτι τέτοιο ισχύει σε αδρές γραμμές στα κλασικά είδη βαθιών νευρωνικών δικτύων όπου εξάγοντας απλούστερα χαρακτηριστικά στα πρώτα επίπεδα καθίστανται ικανά να συνθέσουν πιο σύνθετα χαρακτηριστικά στα επόμενα. Μολονότι τα κλασικά είδη αναλύουν τα δεδομένα μέσω διαδοχικών επιπέδων αξιοποιώντας έτσι την ιεραρχική τους φύση, αδυνατούν εκ κατασκευής να ενσωματώσουν με ρητό τρόπο τη γνώση των σχέσεων σύνδεσης μεταξύ μερών του όλου (τμημάτων ενός αντικειμένου) και του όλου (ολόκληρου του αντικειμένου).



Σχήμα 2.12: Σχήμα ιεραρχικού δέντρου μιας εικόνας αυτοκινήτου. *Παράχθηκε από το Inkscape.*

εργασία αναγνώρισης αντικειμένων σε νέες γεωμετρίες.

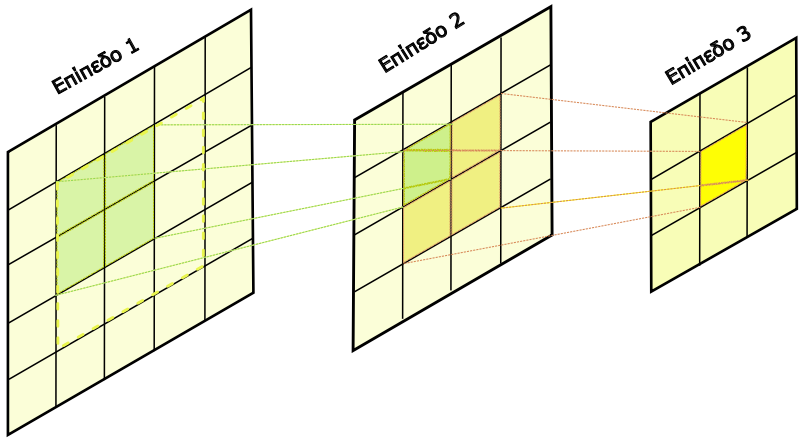
Σχήμα 2.13: Σχήμα όπου απεικονίζεται μια ηλικιωμένη κυρία και μια νεαρή γυναίκα ταυτόχρονα. Ανάλογα με το πιο σύστημα αναφοράς θεωρούμε (προσανατολισμός του κεφαλιού), ο εγκέφαλός μας κάτω από το ίδιο οπτικό ερέθισμα αναγνωρίζει δύο πρόσωπα. *Έγινε λήψη από αυτή την ιστοσελίδα.*

2.2.2 Θετικά Γνωρίσματα Συνελικτικών Νευρωνικών Δικτύων

Προτού αναφερθούμε στα μειονεκτήματα των συνελικτικών νευρωνικών δικτύων που θέλουμε να βελτιώσουμε με τα νευρωνικά δίκτυα από κάψουλες, κρίνεται σκόπιμο να αναγνωρίσουμε ορισμένα θετικά στοιχεία τους τα οποία είναι χρήσιμο να κρατήσουμε. Τα βασικά θετικά στοιχεία που έχουμε προαναφέρει συνοπτικά είναι:

- Η αξιοποίηση της χωρικής συνεκτικότητας της εισόδου με τη διατήρηση των σχέσεων απόστασης μεταξύ των χαρακτηριστικών και τη χρήση φίλτρων εξαγωγής χαρακτηριστικών που δρουν τοπικά.
- Η αξιοποίηση της ιεραρχικής δομής των δεδομένων εικόνων με την ενσωμάτωση διαδοχικών συνελικτικών επιπέδων. Ως εκ τούτου, τα φίλτρα των βαθύτερων επιπέδων έχουν μεγαλύτερο οπτικό πεδίο και δύνανται να συνθέσουν πιο σύνθετα χαρακτηριστικά κωδι-

κοποιώντας έτσι πληροφορία ευρύτερου τμήματος της εικόνας εισόδου (βλ. σχήμα 2.14).



Σχήμα 2.14: Σχήμα τριών διαδοχικών συνελικτικών επιπέδων με μέγεθος φίλτρου 2×2 και βήμα 1. Όσο μεγαλύτερη είναι η απόσταση βάθους μεταξύ πρώτου και τελευταίου επιπέδου, τόσο μεγαλύτερο είναι το οπτικό πεδίο από το οποίο εξάγονται τα χαρακτηριστικά του τελικού επιπέδου. Στην εικόνα, το οπτικό πεδίο ενός στοιχείου στο χάρτη χαρακτηριστικών «επιπέδου 3» σχηματίζει στο πρώτο επίπεδο ένα παραλληλόγραμμο 3×3 (απεικονίζεται με διακεκομμένες, κίτρινες γραμμές). *Παράχθηκε από το Inkscape.*

- Η ελαχιστοποίηση του υπολογιστικού κόστους (και των απαιτήσεων μνήμης) με την εφαρμογή των τοπικών φίλτρων (δηλαδή όχι πλήρως διασυνδεδεμένων) ως κυλιόμενων παραθύρων στον x και y άξονα πάνω στην εικόνα.

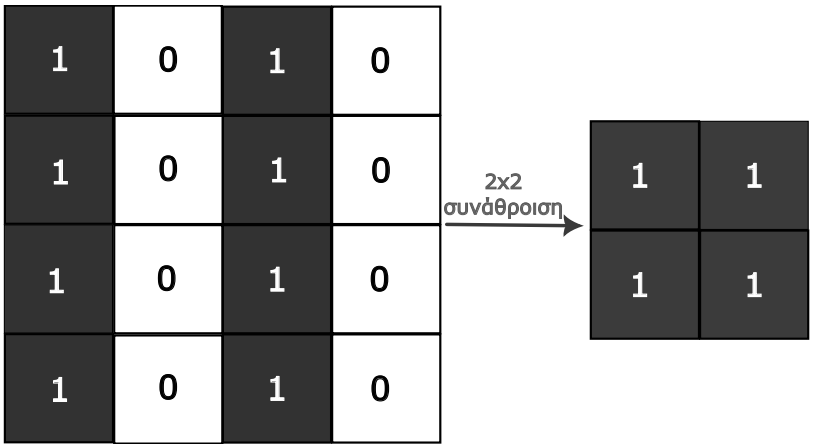
Από τα θετικά αυτά δομικά στοιχεία εμμέσως προκύπτει μια πολύ σημαντική ιδιότητα των συνελικτικών δικτύων: αυτή της μεταφοράς των διακυμάνσεων θέσης των αντικειμένων σε μια εικόνα εισόδου σε κατάλληλες εσωτερικές διακυμάνσεις των χαρτών χαρακτηριστικών (translation equivariance). Αναλυτικότερα, με τη μετακίνηση ενός αντικειμένου στην εικόνα κατά τον x ή y άξονα, λόγω της δισδιάστατης συνέλιξης, αυτή μεταφράζεται σε αντίστοιχη μετακίνηση των εξαχθέντων χαρακτηριστικών. Συνεπώς, θα μπορούσαμε να πούμε ότι ένα συνελικτικό νευρωνικό δίκτυο διαθέτει μηχανισμούς που να μοντελοποιούν τις οριζόντιες και κάθετες μετατοπίσεις της εισόδου ώστε αυτές να γίνονται αντιληπτές από το σύστημα.

2.2.3 Βασικές Ανεπάρκειες των Συνελικτικών Νευρωνικών Δικτύων

Το βασικό πρόβλημα που αντιμετωπίζουν οι αρχιτεκτονικές συνελικτικών νευρωνικών δικτύων που παρουσιάσαμε είναι η αδυναμία γενίκευσης σε νέες οπτικές γωνίες (novel viewpoints). Με άλλα λόγια, είναι σε θέση να αναγνωρίζουν αντικείμενα μόνο όταν βρίσκονται στον ίδιο προσανατολισμό, κλίμακα, διάτμηση (orientation, scale, shear) κ.τ.λ. με τα στιγμιότυπα αντικειμένων που απεικονίζονται στις εικόνες του συνόλου εκπαίδευσης. Έτσι λοιπόν, οι μόνοι αφινικοί μετασχηματισμοί (affine transformations) τους οποίους ένα συνελικτικό νευρωνικό δίκτυο μπορεί να

χειριστεί αποδοτικά είναι οι μεταφορές (μεταθέσεις των αντικειμένων της εικόνας) [76].

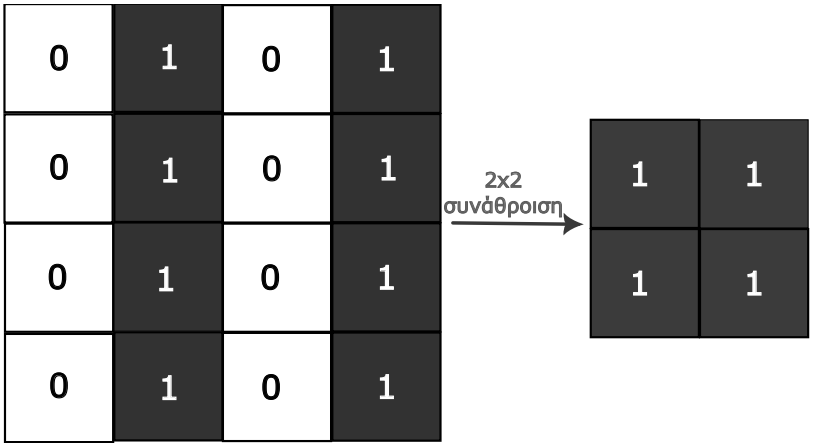
Για την αναγνώριση αντικειμένων υπό νέες οπτικές γωνίες από τα συνελικτικά νευρωνικά δίκτυα χρησιμοποιούνται μη–αποδοτικές μέθοδοι. Για παράδειγμα, μια μέθοδος είναι ο πολλαπλασιασμός των δεδομένων εισόδου μετά από τυχαία εφαρμογή μετασχηματισμών γνωστή ως «επαύξηση δεδομένων» (data augmentation). Μια άλλη μέθοδος που μπορεί να χρησιμοποιηθεί παράλληλα με την προηγούμενη είναι αυτή της ενσωμάτωσης επιπέδων μέγιστης συνάθροισης (max pooling). Όπως έχουμε αναφέρει, τα επίπεδα αυτά αυξάνουν την ευρωστία του συστήματος. Το επιτυγχάνουν, μέσω της υποδειγματοληψίας των χαρτών χαρακτηριστικών έτσι ώστε μικρές μεταβολές στη θέση (ή ακόμα και στον προσανατολισμό [51]) των αντικειμένων να μην αλλάζει τις αποκρίσεις (εξόδους) των φίλτρων των επακόλουθων επιπέδων. Η ιδιότητα αυτή ονομάζεται ανεξαρτησία υπό μεταφορά (translation invariance) και σε αντίθεση με την ιδιότητα των συνελικτικών επιπέδων που περιγράψαμε στην παράγραφο 2.2.2, οι μικρές διακυμάνσεις στην είσοδο του επιπέδου συνάθροισης απορρίπτονται και δε μοντελοποιούνται εσωτερικά του συστήματος. Με απλά λόγια, το σύστημα επιδιώκει να πετύχει γενίκευση στους αφινικούς μετασχηματισμούς κάτω από τους οποίους αναγνωρίζει τα αντικείμενα με το να αχρηστεύει την πληροφορία σχετικά με το συγκεκριμένο στιγμιότυπο εισόδου και να δημιουργεί μια ανεξάρτητη αναπαράσταση (εξαρτώμενη μόνο από το είδος του αντικειμένου) την οποία τα επόμενα επίπεδα θα επεξεργαστούν.



Σχήμα 2.15: Σχήμα όπου εφαρμόζεται μέγιστη συνάθροιση με πυρήνα 2×2 και βήμα 2 σε μια δυαδική εικόνα δύο κάθετων ακμών. *Παράχθηκε από το Inkscape.*

Σύμφωνα με τον G. Hinton [78], τα νευρωνικά δίκτυα θα πρέπει να χειρίζονται όλους τους αφινικούς μετασχηματισμούς με την ίδια λογική που διαχειρίζονται τα συνελικτικά επίπεδα τις κάθετες και οριζόντιες μετατοπίσεις. Δηλαδή, αντί να απορρίπτουν χρήσιμη πληροφορία μέσω των επιπέδων συνάθροισης να διαθέτουν μηχανισμούς που θα μοντελοποιούν εσωτερικά τις διακυμάνσεις στην οπτική γωνία των αντικείμενων. Η πρόταση αυτή βασίζεται στη σημαντική παρατήρηση ότι αλλαγές στη σκοπιά ενός αντικειμένου μεταβάλλουν με σύνθετο, μη γραμμικό τρόπο τα εικονοστοιχεία της εικόνας ενώ τροποποιούν με απλό, γραμμικό τρόπο τη μήτρα πόζας (pose matrix) του αντικειμένου[22]. Συνεπώς, φαίνεται ασύμφορη η προσπάθεια των συνελικτικών δικτύων να δημιουργούν ανεξάρτητες (υπό την οπτική γωνία) αναπαραστάσεις αντικειμένων απευθείας από

[22]Οι μήτρες πόζας είναι πίνακες που περιγράφουν τη θέση και τον προσανατολισμό ενός αντικειμένου, δύο χαρακτηριστικά τα οποία μεταβάλλονται γραμμικά με την αλλαγή της οπτικής γωνίας θέασης ενός αντικειμένου. Χρησιμοποιούνται κατά κόρον στον χώρο των γραφικών με υπολογιστή για την περιγραφή του τρόπου τοποθέτησης των αντικειμένων σε έναν εικονικό κόσμο.



Σχήμα 2.16: Σχήμα όπου εφαρμόζεται μέγιστη συνάθροιση με πυρήνα 2×2 και βήμα 2 σε μια δυαδική εικόνα δύο κάθετων ακμών, αφού η θέση λήψης μετατοπιστεί. Χάρη στο επίπεδο συνάθροισης, η απόκριση είναι ανεξάρτητη από μικρές μετατοπίσεις της εικόνας εισόδου. *Παράχθηκε από το Inkscape.*

τον χώρο των εικονοστοιχείων, χωρίς δηλαδή να λαμβάνουν υπόψη τη γραμμική σχέση μεταξύ των διακυμάνσεων της οπτικής γωνίας και των παραμέτρων του στιγμιότυπου (instantiation parameters) του αντικειμένου[23]. Αντίθετα, θα ήταν πιο αποδοτική η μοντελοποίηση αυτής της γραμμικής σχέσης με έναν μηχανισμό ο οποίος θα πραγματοποιούσε ανάστροφα γραφικά (inverse graphics): θα αντιστοίχιζε τον χώρο των εικονοστοιχείων της εικόνας εισόδου σε έναν ιεραρχικό χώρο από μήτρες πόζας για το κάθε απεικονιζόμενο αντικείμενο. Σε αυτήν τη νέα αναπαράσταση, οι αφινικοί μετασχηματισμοί θα άλλαζαν με προβλέψιμο τρόπο τις -απεπλεγμένες από το είδος του αντικειμένου- παραμέτρους των επιμέρους στιγμιοτύπων οδηγώντας στην επιθυμητή γενίκευση σε νέες οπτικές γωνίες.

Επιπρόσθετα, ο παρόν τρόπος διαχείρισης αφινικών μετασχηματισμών από τα συνελικτικά νευρωνικά δίκτυα τα καθιστά επιρρεπή σε αντιπαραθετική επίθεση (adversarial attacks). Αυτή τους η αδυναμία, θα μπορούσε να καταπολεμηθεί με την ενσωμάτωση ενός μηχανισμού που θα μοντελοποιούσε τις σχέσεις μεταξύ των τμημάτων ενός αντικειμένου ούτως ώστε, για την αναγνώρισή του, να λαμβάνονταν υπόψη η γεωμετρία των επιμέρους μερών του. Με άλλα λόγια, αν υπήρχε «αποθηκευμένη» στο νευρωνικό δίκτυο η πληροφορία για τον τρόπο σύνδεσης των στοιχείων που απαρτίζουν ένα αντικείμενο τότε θα ήταν περισσότερο εύρωστο σε αυτού του είδους τις επιθέσεις. Στο παράδειγμα του σχήματος 2.17 το συνελικτικό νευρωνικό δίκτυο αφενός δε διαθέτει κάποιο μηχανισμό να αναγνωρίζει την ακριβή θέση του ματιού στην εικόνα (αφού αυτή η πληροφορία απορρίπτεται σε ένα βαθμό μέσω των επιπέδων συνάθροισης) και αφετέρου, ακόμα και αν ήταν διαθέσιμη αυτή η πληροφορία, θα έμενε αναξιοποίητη διότι δεν αποθηκεύεται η γνώση για το ποια θα πρέπει να είναι η θέση του ματιού σε σχέση με τα υπόλοιπα μέρη.

Ένα τελευταίο σημείο αδυναμίας των κλασικών νευρωνικών δικτύων είναι το λεγόμενο πρόβλημα της αποκλειστικής διάζευξης (XOR problem) [79]. Αυτό, προκύπτει από την παρατήρηση ότι η συνάρτηση της αποκλειστικής διάζευξης δεν μπορεί να υλοποιηθεί από ένα μεμονωμένο τεχνητό

[23]Με τον όρο «παράμετροι στιγμιοτύπου» θα αναφερόμαστε κυρίως στην μήτρα πόζας του στιγμιοτύπου. Παρόλα αυτά, παράμετροι στιγμιοτύπου είναι και άλλοι παράγοντες που δεν εξαρτόνται από την κλάση του αντικειμένου προς αναγνώριση όπως ο φωτισμός, το μέγεθος ή χρώμα του αντικειμένου κ.τ.λ.

Σχήμα 2.17: Σχήμα όπου απεικονίζεται το λεγόμενο πρόβλημα του Picasso στο οποίο η εικόνα έχει όλα τα σωστά μέρη αλλά οι σχέσεις μεταξύ τους είναι λάθος. Ένα υποθετικό συνελικτικό νευρωνικό δίκτυο θα δυσκολεύονταν να αντιληφθεί ότι το σχήμα της εικόνας δεν είναι κανονικό πρόσωπο. (Το παράδειγμα είναι ενδεικτικό αφού δεν έχει αποδειχθεί ότι ένα συνελικτικό δίκτυο θα αναγνώριζε το συγκεκριμένο παράδειγμα ως πρόσωπο.) *Παράχθηκε από το Inkscape τροποποιώντας αυτή την εικόνα.*

νευρώνα. Σύμφωνα με την περιγραφή του τεχνητού νευρώνα του σχήματος 2.4, στον πυρήνα του προβλήματος βρίσκεται το γεγονός ότι δεν υπάρχει δυνατότητα σύγκρισης των εισόδων μεταξύ τους. Αντ΄ αυτού, πραγματοποιείται σύγκριση μεταξύ ενός διανύσματος εισόδων με ένα διάνυσμα (αποθηκευμένων) βαρών. Αυτή η αδυναμία όμως οδηγεί σε μη–αποδοτικές λύσεις του προβλήματος (με την προσθήκη κρυφών επιπέδων). Επακόλουθη, λοιπόν, είναι η διάθεση πειραματισμού με ένα νέο είδος τεχνητού νευρώνα που θα μπορεί να συγκρίνει τις εισόδους (ή τα διανύσματα εισόδων) επιλύοντας το πρόβλημα της αποκλειστικής διάζευξης και κυρίως επιτρέποντας τον εντοπισμό συνδιακυμάνσεων στα χαρακτηριστικά εισόδου.

2.2.4 Αρχές Λειτουργείας Νευρωνικών Δικτύων με Κάψουλες

Στην προσπάθεια αντιμετώπισης των ανωτέρω μειονεκτημάτων των συνελικτικών νευρωνικών δικτύων σε εργασίες αυτόματης αναγνώρισης αντικειμένων, αναπτύχθηκαν τα νευρωνικά δίκτυα με κάψουλες. Αυτά διαμορφώθηκαν:

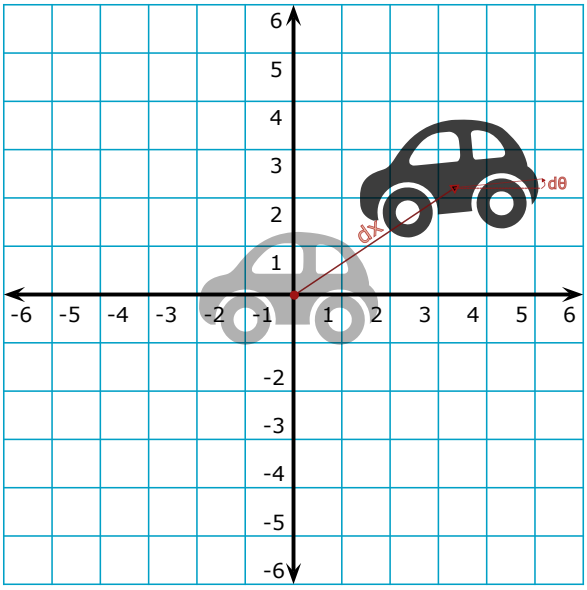
- αντλώντας στοιχεία από την επιστήμη της νευροφυσιολογίας
- διατηρώντας αρκετά θετικά στοιχεία των συνελικτικών νευρωνικών δικτύων (π.χ. συνέλιξη)
- εισάγοντας τις ιεραρχικές δομές νευρώνων (κάψουλες) και έναν μηχανισμό για τη διασύνδεσή τους.

Ορισμός Κάψουλας και Αρχές Λειτουργίας της

Η κάψουλα δεν είναι τίποτα άλλο παρά μια ομάδα από νευρώνες. Κάθε κάψουλα έχει μια και μοναδική λειτουργία μέσα σε ένα νευρωνικό δίκτυο. Πιο συγκεκριμένα, κάθε μια εκπαιδεύεται να αναπαριστά την πιθανότητα παρουσίας και τις παραμέτρους στιγμιοτύπου μιας συγκεκριμένης οντότητας στο πεδίο υποδοχής της. Για παράδειγμα, σε μια εφαρμογή οπτικής αναγνώρισης

του τύπου τροχοφόρων οχημάτων, μια κάψουλα θα μπορούσε να είναι επιφορτισμένη με την αναπαράσταση της οντότητας «ρόδα». Σε περίπτωση που η οντότητα ήταν παρούσα στο πεδίο υποδοχής της αντίστοιχης κάψουλας τότε θα είχε ως έξοδο μεγάλη τιμή πιθανότητας παρουσίας και φυσικά τις τιμές που θα προσδιόριζαν τη θέση, τον προσανατολισμό, το μέγεθος κ.τ.λ. της αναγνωρισμένης ρόδας. Στο σημείο αυτό, γίνεται αντιληπτό το πλεονέκτημα της χρήσης συστάδων από νευρώνες έναντι μεμονωμένων νευρώνων (όπως αυτός του σχήματος 2.4 που παράγει μια μονοδιάστατη τιμή εξόδου ή ενεργοποίησης), αφού έτσι γίνεται εφικτή η έξοδος πιο εκφραστικών, πολυδιάστατων αναπαραστάσεων.

Όπως είναι φυσικό, για την παραγωγή των παραμέτρων στιγμιοτύπου μιας οντότητας είναι απαραίτητη η διατήρηση ενός συστήματος αναφοράς για τη συγκεκριμένη οντότητα[24] (βλ. σχήμα 2.18). Αυτό είναι σύμφωνο με τις παρατηρήσεις λειτουργίας του ανθρώπινου οπτικού φλοιού που, όπως διατυπώσαμε, εφαρμόζει συστήματα αναφοράς σε κάθε οπτικό ερέθισμα. Το είδος και η γεωμετρία της οντότητας αναφοράς σχηματίζεται κατά τη διαδικασία εκμάθησης του νευρωνικού δικτύου.



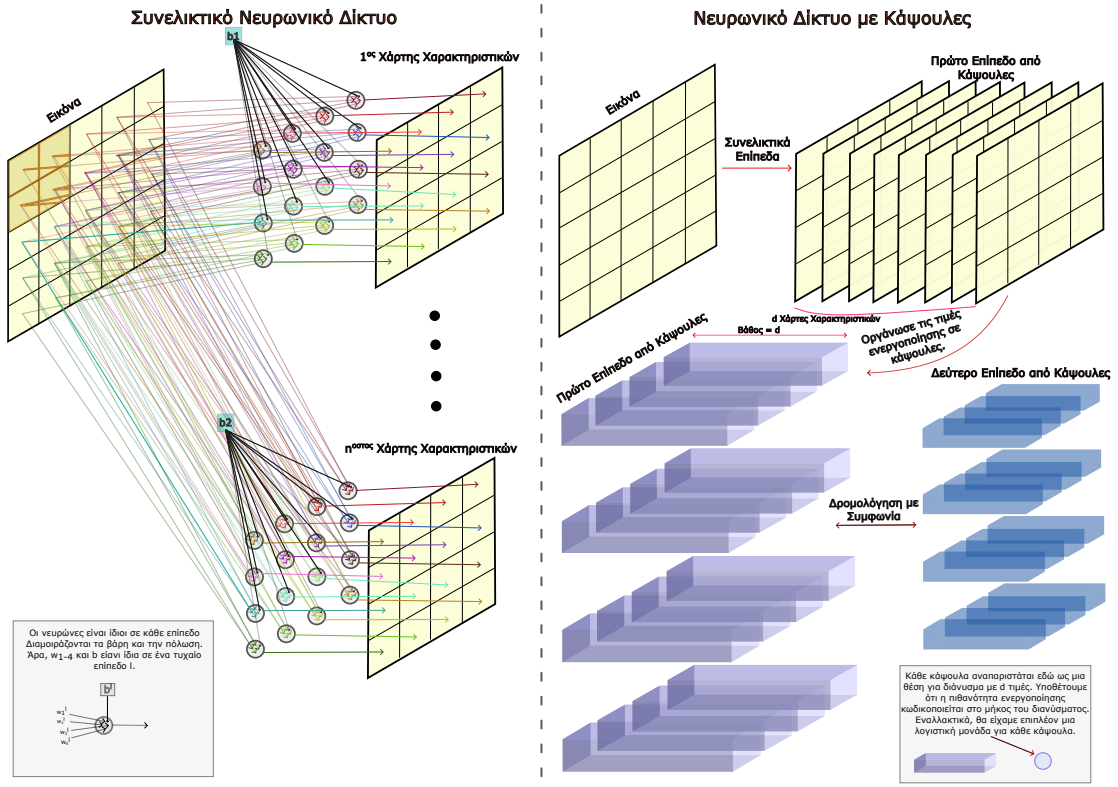
Σχήμα 2.18: Σχήμα όπου με τη βοήθεια μιας οντότητας αναφοράς (απεικονίζεται ως αχνό αυτοκίνητο) και του καρτεσιανού συστήματος συντεταγμένων υπολογίζονται οι παράμετροι στιγμιοτύπου (απόσταση από αρχή αξόνων και γωνία περιστροφής). *Παράχθηκε από το Inkscape.*

Με μαθηματικούς όρους, κάθε κάψουλα c_i αποτελείται από ένα διάνυσμα $m_i \in \Re^d$ ή πίνακα $M_i \in \Re^{\sqrt{d}\times\sqrt{d}}$ από παραμέτρους στιγμιοτύπου και μια τιμή πιθανότητας παρουσίας $a_i \in [0, 1]$. Η εξασφάλιση ότι η τιμή πιθανότητας ανήκει στο διάστημα $[0, 1]$ γίνεται με την εφαρμογή μιας μη γραμμικής συνάρτησης ενεργοποίησης με σύνολο εξόδου το διάστημα αυτό. Σε ορισμένες υλοποιήσεις, η τιμή πιθανότητας παρουσίας της οντότητας που αναπαριστά η κάψουλα c_i κωδικοποιείται στο μήκος του διανύσματος m_i.

[24] Η αναγκαιότητα ύπαρξης συστήματος αναφοράς της οντότητας γίνεται ακόμα πιο προφανής αν αναλογιστεί κανείς ότι οι παράμετροι στιγμιοτύπου ουσιαστικά περιγράφουν τη σχέση μεταξύ μιας οντότητας υπό την οπτική γωνία λήψης της εικόνας και της αντίστοιχης οντότητας αναφοράς.

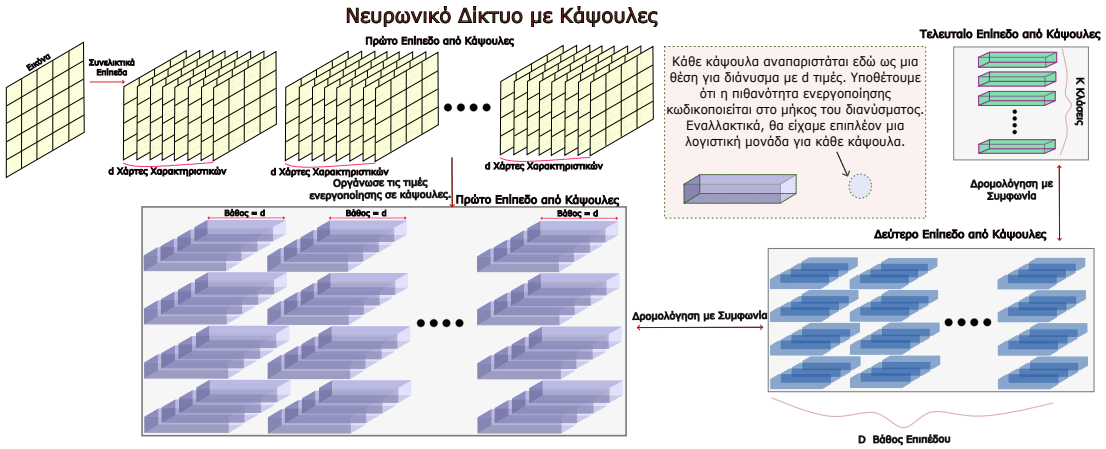
Οργάνωση των Καψουλών στο Νευρωνικό Δίκτυο

Τα νευρωνικά δίκτυα από κάψουλες αξιοποιούν την τοπική χωρική συνεκτικότητα των εικόνων αφού οι κάψουλες οργανώνονται σε τρισδιάστατες δομές. Επίσης, αξιοποιούν την ιεραρχική δομή των φυσικών αντικειμένων με το να διατηρούν την πολυεπίπεδη διάταξη που χαρακτηρίζει τα συνελικτικά νευρωνικά δίκτυα (βλ. σχήμα 2.20). Η διαφορά έγκειται στο γεγονός ότι αντί για επίπεδα από χάρτες χαρακτηριστικών υπάρχουν επίπεδα από κάψουλες (συμβολίζονται ως $C^{[l]}$)[25]. Τα επίπεδα αυτά μπορεί να είναι πλήρως διασυνδεδεμένα ή να είναι συνελικτικά. Για παράδειγμα, στην περίπτωση των συνελικτικών επιπέδων από κάψουλες, αντί για παραγωγή χαρτών χαρακτηριστικών με μοναδιαίο βάθος ο καθένας, θα λέγαμε ότι παράγονται πλούσιοι σε πληροφορία χάρτες από διανύσματα ή πίνακες (πολυδιάστατες αναπαραστάσεις των αναγνωριζόμενων μοτίβων–χαρακτηριστικών) που αποτελούν και το επόμενο επίπεδο από κάψουλες. Η παραγωγή του επόμενου επιπέδου από κάψουλες με βάση το προηγούμενο δε γίνεται με τη χρήση επιπέδων από νευρώνες αλλά πραγματοποιείται μέσω ενός αλγορίθμου «δρομολόγησης μέσω συμφωνίας» ο οποίος θα αναλυθεί στη συνέχεια.



Σχήμα 2.19: Δομικές διαφορές μεταξύ συνελικτικών νευρωνικών δικτύων και δικτύων με κάψουλες. Παρατηρούμε ότι οι κάψουλες δεν έχουν ομοιότητες με τους κλασικούς τεχνητούς νευρώνες. Κυρίως μπορούν να συσχετιστούν με σύνολα από χάρτες χαρακτηριστικών. *Παράχθηκε από το Inkscape.*

[25]Ιδιαίτερη προσοχή απαιτείται καθώς οι κάψουλες, αποτελούμενες από ένα σύνολο χαρακτηριστικών της οντότητας που αναγνωρίζουν, μπορούν να αντιπαραβληθούν με τους χάρτες χαρακτηριστικών (δηλαδή τον πίνακα τιμών ενεργοποίησης A). Οι νευρώνες από τους οποίους αποτελούνται οι κάψουλες διαφέρουν από αυτούς της ενότητας 2.1 (βλ. σχήμα 2.19).



Σχήμα 2.20: Σχήμα όπου φαίνεται μια τυπική οργάνωση ενός νευρωνικού δικτύου με κάψουλες. *Παράχθηκε από το Inkscape.*

Αλγόριθμος Δρομολόγησης με Συμφωνία

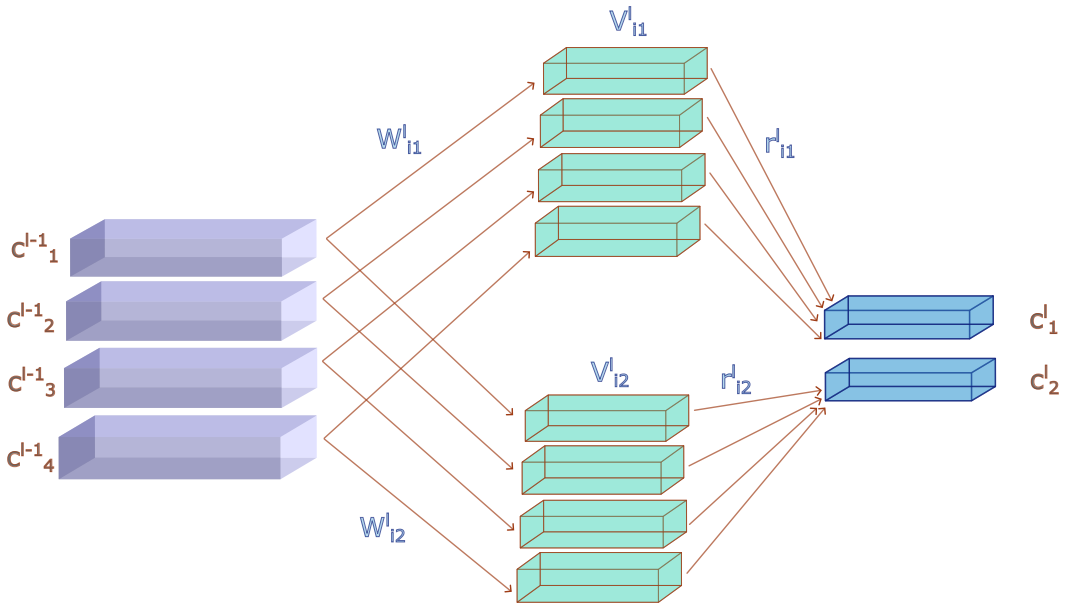
Θα θέλαμε να εστιάσουμε στους υπολογισμούς που λαμβάνουν χώρα μεταξύ δύο διαδοχικών επιπέδων από κάψουλες. Αν και κάθε υλοποίηση είναι διαφορετική, θα παρουσιάσουμε υπό μια αφαιρετική σκοπιά τις βασικές αρχές που διέπουν τον κάθε αλγόριθμο δρομολόγησης. Επειδή σε ένα νευρωνικό δίκτυο από κάψουλες τα πρώτα επίπεδά του επιτελούν ιδιαίτερες λειτουργίες, ας υποθέσουμε χωρίς βλάβη της γενικότητας ότι το έχουμε τροφοδοτήσει με μια εικόνα και διαδοχικά έχουν υπολογιστεί ήδη οι τιμές των επιπέδων του μέχρι και το $l-1$. Θα επιθυμούσαμε στο σημείο αυτό να υπολογίσουμε τις τιμές για το επίπεδο l.

Με τους μέχρι τώρα υπολογισμούς, έχουμε στη διάθεσή μας μια τρισδιάστατη δομή από κάψουλες όπως φαίνεται στο σχήμα 2.20 σε μώβ χρώμα. Κάθε κάψουλα εμπεριέχει ένα σύνολο από τιμές που περιγράφουν την πόζα του τμήματος του αντικειμένου με το οποίο έχουν ταυτιστεί και αναγνωρίζουν. Επίσης, διαθέτουν και μια τιμή ενεργοποίησης που περιγράφει την πιθανότητα αυτό το τμήμα να είναι παρόν στο οπτικό πεδίο της κάψουλας. Κάψουλες με χαμηλή τιμή ενεργοποίησης θα λέμε ότι είναι ανενεργές και δε θα έχουν μεγάλη βαρύτητα στον υπολογισμό των τιμών του επόμενου επιπέδου.

Όπως είναι γνωστό, στο επίπεδο l οι κάψουλες αναπαριστούν ανώτερες ιεραρχικά οντότητες σε σχέση με αυτές του επιπέδου $l-1$. Συνεπώς, η διαμόρφωση των τιμών του επιπέδου l ανάγεται στο πρόβλημα της αντιστοίχισης (δρομολόγησης) επιμέρους τμημάτων των αντικειμένων σε μια εικόνα (αναπαριστώνται με κάψουλες $C[l-1]$) στα γενικότερα αντικείμενα που τα περιέχουν (αναπαριστώνται με τις κάψουλες $C^{[l]}$). Αυτή η δρομολόγηση προϋποθέτει την ύπαρξη ενός μηχανισμού που θα προβλέπει τις παραμέτρους που περιγράφουν τις γενικότερες οντότητες που αναγνωρίζουν οι κάψουλες $C^{[l]}$ με βάση τις παραμέτρους στιγμιοτύπων των επιμέρους οντοτήτων που αναγνωρίζουν οι κάψουλες $C^{[l-1]}$[26]. Ο μηχανισμός υλοποιείται με πίνακες από βάρη $\boldsymbol{W}^{[l]}$ που αναλαμβάνουν να αποθηκεύσουν τις σχέσεις μέρους–όλου (part–whole relationships), δηλαδή τις σχέσεις όλων των δυνατών ζευγών μεταξύ των καψουλών επιπέδου $l-1$ και l. Έτσι αν στο επίπεδο $l-1$ έχουμε $n^{[l-1]}$ κάψουλες και στο επίπεδο l, $n^{[l]}$ τότε θα υπάρχουν $n^{[l-1]} \times n^{[l]}$ πίνακες

[26] π.χ. με βάση τη γεωμετρία της επιμέρους οντότητας «χέρι» να προβλέπεται η γεωμετρία της οντότητας «άνδρας».

βαρών μεταξύ των δύο επιπέδων. Μαθηματικά, η πρόβλεψη (ή ψήφο) για μια κάψουλα $c_j^{[l]}$ με βάση την κάψουλα $c_i^{[l-1]}$ παράγεται πολλαπλασιάζοντας τον πίνακα βαρών $W_{ij}^{[l]}$ με το διάνυσμα ή πίνακα παραμέτρων στιγμιότυπου $M_i^{[l-1]}$, δηλαδή $V_{ij}^{[l]} = M_i^{[l-1]} \times W_{ij}^{[l]}$. Να σημειώσουμε ότι τα βάρη $\boldsymbol{W}$ μαθαίνονται κατά την εκπαίδευση με τον αλγόριθμο της οπισθοδιάδοσης.



Σχήμα 2.21: Τρόπος παραγωγής των ψήφων για την απλή περίπτωση όπου έχουμε τέσσερεις κάψουλες στο επίπεδο $l-1$ και δύο στο επίπεδο l. Στο σχήμα, $i \in [1, 4]$. Επίσης, η τιμή ενεργοποίησης θεωρήθηκε ότι κωδικοποιείται στο μήκος τω διανυσμάτων από παραμέτρους. *Παράχθηκε από το Inkscape.*

Μέχρι τώρα, έχουμε αναφέρει τον τρόπο με τον οποίο η κάθε μια κάψουλα $c^{[l]}$ διαθέτει $n^{[l-1]}$ προβλέψεις (μια από κάθε κάψουλα $c^{[l-1]}$) για το ποιες εκτιμά ότι είναι οι παράμετροι στιγμιοτύπου της $M^{[l]}$. Δεν έχουμε περιγράψει όμως τον τρόπο με τον οποίο αυτές οι προβλέψεις συγκροτούνται για την τελική διαμόρφωση των $\boldsymbol{M}^{[l]}$ και $\boldsymbol{a}^{[l]}$. Αυτό μας οδηγεί στην έννοια του φιλτραρίσματος μέσω της πολυδιάστατης σύμπτωσης (high dimensional coincidence filtering). Σύμφωνα με την έννοια αυτή, όταν μερικές από τις ψήφους των $C^{[l-1]}$ συμπέσουν σε μια γειτονιά του πολυδιάστατου χώρου των αναπαραστάσεων $\Re^d$ τότε αυτή η σύμπτωση δεν μπορεί να είναι τυχαία. Αντιθέτως, το γεγονός ότι μεγάλος αριθμός από κάψουλες $C^{[l-1]}$ συμφωνούν στο ποιες θα είναι οι τιμές παραμέτρων της εκάστοτε $c^{[l]}$ σημαίνει ότι πιθανότατα, αυτές είναι οι κατάλληλες τιμές της (και φυσικά ότι η οντότητα που εκπροσωπεί υπάρχει στην εικόνα). Έτσι η διαδικασία δρομολόγησης τελικά ανάγεται σε πρόβλημα εύρεσης συστάδων από ψήφους (clusters of votes) στον χώρο $\Re^d$.

Με βάση τα ανωτέρω, θα ήταν δυνατή η δρομολόγηση ενός τμήματος μιας οντότητας που αναπαριστάται από μια κάψουλα $c^{[l-1]}$ σε όλες τις κάψουλες $C^{[l]}$. Διαισθητικά, κάτι τέτοιο δεν είναι ορθό αφού ένα τμήμα ενός αντικειμένου δεν είναι δυνατό να αποτελεί μέρος όλων των

αντικειμένων που αναπαριστώνται από τις κάψουλες του επόμενου επιπέδου. Επιπλέον, θα δημιουργούσε σύγχυση στον χώρο αναπαραστάσεων κατακλύζοντάς τον με περιττή πληροφορία. Συνεπώς, εισάγουμε τον περιορισμό ότι κάθε κάψουλα μπορεί να δρομολογεί τελικά την ψήφο της μόνο σε μια κάψουλα του επόμενου επιπέδου (single parent assumption). Με αυτόν τον τρόπο προκαλούμε ανταγωνισμό μεταξύ των $C^{[l]}$ να «εξηγήσουν» όσο το δυνατόν περισσότερες ψήφους των $C^{[l-1]}$[27].

Η κάθε κάψουλα $c^{[l-1]}$ είναι αδύνατο να γνωρίζει εκ των προτέρων (a priori) ποια κάψουλα του επόμενου επιπέδου (κάψουλα πατέρας) θα την εκφράζει καλύτερα αφού αυτό εξαρτάται όπως είπαμε από το αν η ψήφος της συμπίπτει μαζί με άλλες ψήφους στον χώρο αναπαραστάσεων (φιλτράρισμα πολυδιάστατης σύμπτωσης). Μονόδρομος λοιπόν φαίνεται να είναι η επαναληπτική φύση του αλγορίθμου με συμφωνία κατά την οποία αρχικά οι ψήφοι της κάθε κάψουλας $c^{[l-1]}$, βεβαρημένες υπό μια ομοιόμορφη κατανομή διακριτής πιθανότητας, δρομολογούνται σε όλες τις $C^{[l]}$. Στη συνέχεια, μέσω του φιλτραρίσματος πολυδιάστατης σύμπτωσης, κάθε κάψουλα $c^{[l]}$ ανταγωνίζεται για να προσαρτήσει κάψουλες $c^{[l-1]}$ των οποίον οι ψήφοι σχηματίζουν συστάδες και άρα μπορεί εύκολα να «εξηγήσει»[28]. Χωρίς βλάβη της γενικότητας, κάθε κάψουλα $c^{[l]}$ «εξηγεί» τις ψήφους προσαρμόζοντας τον πίνακα (ή το διάνυσμα) $M^{[l]}$ στο κέντρο βάρους των ψήφων. Επιπρόσθετα, προσαρμόζει την τιμή πιθανότητας ενεργοποίησής της ανάλογα με το πόσο καλά εξηγεί τις ψήφους[29]. Αυτές οι δύο προσαρμογές ανατροφοδοτούνται πίσω στις κάψουλες $c^{[l-1]}$ οι οποίες αλλάζουν, η κάθε μία, την κατανομή των βαρών υπό τα οποία δρομολογούν τις ψήφους τους (coupling coefficients) έτσι ώστε να προτιμούν κάψουλες γονείς που είναι ενεργές και εκφράζουν καλύτερα την ψήφο τους (το διάνυσμα της ψήφου τους είναι πιο κοντά στο διάνυσμα $M^{[l]}$). Όσο εξελίσσονται οι επαναλήψεις, τόσο οι κάψουλες $c^{[l-1]}$ είναι πιο σίγουρες για το που θα αποστείλουν τις ψήφους τους (η ομοιόμορφη κατανομή εκφυλίζεται σε ένα σημείο–κάψουλα) και οι πίνακες $\boldsymbol{M}^{[l]}$ συγκλίνουν στο κέντρο των συστάδων από ψήφους.

Συγκεντρωτικά, μπορούμε να παρουσιάσουμε έναν αφαιρετικό αλγόριθμο δρομολόγησης με συμφωνία μεταξύ δύο διαδοχικών επιπέδων από κάψουλες $l-1$ και l ως εξής:

1. Για κάθε κάψουλα $c_i^{[l-1]} \in C^{[l-1]}$ υπολόγισε τις ψήφους–προβλέψεις ως $V_{ij} = M_i \times W_{ij}$
2. Αρχικοποίησε τις τιμές βαρών δρομολόγησης $r_{ij} = 1/n^{[l]}, \forall i \in [1, n^{[l-1]}], \forall j \in [1, n^{[l]}]$ έτσι ώστε $\sum_{j=1}^{n^{[l]}} r_{ij} = 1$.
3. Επανέλαβε για προκαθορισμένο αριθμό επαναλήψεων:
   (α΄) Για καθεμία κάψουλα $c_j^{[l]}$ εντόπισε τις συστάδες από βεβαρημένες ψήφους στον χώρο αναπαράστασης $\Re^d$ και με βάση αυτές υπολόγισε τις παραμέτρους στιγμιοτύπου $M_j^{[l]}$ που θα την εκφράζουν, $\forall j \in [1, n[l]]$.

[27] Χρησιμοποιούμε τον όρο «εξηγήσουν» διότι το σύνολο παραμέτρων $M^{[l]}$ που δημιουργείται για μια κάψουλα $c^{[l]}$ κατά μια έννοια εκφράζει τις απόψεις των $C^{[l-1]}$ που την ψήφισαν αναφορικά με το ποιο είναι το στιγμιότυπο του αντικειμένου που αναπαριστά.

[28] Στον σχηματισμό συστάδων βοηθάει το γεγονός ότι οι κάψουλες του συνόλου $C^{[l]}$ βλέπουν διαφορετικές ψήφους της κάθε κάψουλας του συνόλου $C^{[l-1]}$ αφού παράγονται μετά από πολλαπλασιασμό με διαφορετικό πίνακα βαρών. Έτσι, κάψουλες οι οποίες σχηματίζουν συστάδες από ψήφους στον χώρο αναπαράστασης μιας κάψουλας $c^{[l]}$ μπορεί στον χώρο αναπαράστασης μιας διαφορετικής κάψουλας να σχηματίζουν απόμακρες προβλέψεις.

[29] Ο μηχανισμός υπολογισμού πιθανότητας ενεργοποίησης $a^{[l]}$ διαφέρει σημαντικά από υλοποίηση σε υλοποίηση αλλά γενικά είναι ανάλογος του αριθμού από κάψουλες $C^{[l-1]}$ που προτιμούν την κάψουλα $c^{[l]}$ και της πυκνότητας των ψήφων τους.

(β′) Υπολόγισε για καθεμία κάψουλα $c_j^{[l]}$ την τιμή ενεργοποίησής της με βάση το πόσο καλά εξηγεί τα δεδομένα.

(γ′) Για κάθε κάψουλα $c_i^{[l-1]}$ ενημέρωσε τα βάρη δρομολόγησης $r_{ij}, \forall j \in [1, n^{[l]}]$ ώστε να δίνεται μεγαλύτερο βάρος σε κάψουλες γονείς των οποίων οι παράμετροι στιγμιοτύπου εξηγούν καλύτερα τις ψήφους.

4. Τερμάτισε με έξοδο $\boldsymbol{M}^{[l]}$ και $\boldsymbol{a}^{[l]}$.

Ας φανταστούμε τώρα ότι έχουμε πολλά διαδοχικά επίπεδα από κάψουλες. Με τον αλγόριθμο δρομολόγησης (και λόγω της υπόθεσης μοναδικού πατέρα), δημιουργείται δυναμικά κατά την πρόσθια τροφοδότηση του δικτύου ένα ιεραρχικό δέντρο από ενεργές κάψουλες όπου η κάθε μια αναπαριστά τις οντότητες που βρίσκονται στην εικόνα. Ο σχηματισμός του ιεραρχικού δέντρου θα μπορούσε να παρομοιαστεί με το σκάλισμα ενός γλυπτού από ένα κομμάτι μαρμάρου [76]. Το μάρμαρο είναι όλες οι κάψουλες σε κάθε επίπεδο ενώ η διαδικασία σκαλίσματος πραγματοποιείται από τον αλγόριθμο δρομολόγησης με συμφωνία που επιλεκτικά συνδέει και ενεργοποιεί ορισμένες κάψουλες. Επειδή η κάθε κάψουλα αναπαριστά όχι μόνο την πιθανότητα ύπαρξης της οντότητας αλλά και τις παραμέτρους στιγμιοτύπου της, μπορούμε να υποθέσουμε (χρησιμοποιώντας την ορολογία της προηγούμενης υποενότητας) ότι η ιεραρχική δομή είναι εμπλουτισμένη. Με άλλα λόγια, η δενδροειδής δομή από τις ενεργές κάψουλες που σχηματίζεται δυναμικά κατά την πρόσθια τροφοδότηση δε μοντελοποιεί μόνο την ιεραρχία μεταξύ ενός αντικειμένου και των μερών του (με σχέσεις τύπου κόμβοι γονέων $\supseteq$ κόμβοι παιδιών) αλλά και τις γεωμετρικές σχέσεις μεταξύ αυτών (π.χ. σε ποια θέση τοποθετούνται τα επιμέρους τμήματα για να συνθέσουν το όλον).

Πρώτα Επίπεδα ενός Νευρωνικού Δικτύου με Κάψουλες

Τα πρώτα επίπεδα ενός νευρωνικού δικτύου, όπως προκύπτει από τη μέχρι τώρα ανάλυση, είναι επιφορτισμένα με τον μετασχηματισμό από τον χώρο των εικονοστοιχείων στον χώρο αναπαράστασης των παραμέτρων στιγμιοτύπων[30] ώστε να μπορούν μετά σε αυτό να δράσουν οι κάψουλες. Πρακτικά, πρόκειται για συνελικτικά επίπεδα από κλασικούς νευρώνες που παράγουν χάρτες χαρακτηριστικών που αποτελούνται από βαθμωτά μεγέθη. Στη συνέχεια αυτά τα βαθμωτά μεγέθη ομαδοποιούνται σε διανύσματα ή πίνακες ούτος ώστε κάθε ομάδα να ενθυλακώνει τις παραμέτρους μιας κάψουλας. Μέσω εκπαίδευσης, τα πρώτα επίπεδα μαθαίνουν να πραγματοποιούν ανάστροφα γραφικά (derendering)[31] και να παράγουν αυτό που ονομάζουμε «αρχείο σκηνής» (βλ. παράρτημα Α′). Αυτή τη λειτουργία δε διαθέτουν τα κλασικά συνελικτικά νευρωνικά δίκτυα διότι οφείλεται στους δομικούς και λειτουργικούς περιορισμούς που επιβάλει το νευρωνικό δίκτυο από κάψουλες.

Τελευταίο Επίπεδο ενός Νευρωνικού Δικτύου με Κάψουλες

Συνήθως, το τελευταίο επίπεδο ενός τέτοιου δικτύου είναι ένα επίπεδο από κάψουλες (όπως φαίνεται στο σχήμα 2.20). Στις περισσότερες περιπτώσεις για εφαρμογές ταξινόμησης, υπάρχει μία κάψουλα ανά κατηγορία. Η πρόβλεψη $\hat{y}$ του νευρωνικού δικτύου λαμβάνεται ως η οντότητα

[30] Υπενθυμίζουμε ότι στον χώρο αυτό, οι αλλαγές στην οπτική γωνία προκαλούν γραμμικές μεταβολές στις παραμέτρους (στα χαρακτηριστικά).

[31] Ουσιασιαστικά πραγγματοποιούν μη παραμετρικό μετασχηματισμό Hough.

που εκπροσωπείται από την κάψουλα του τελευταίου επιπέδου που έχει την μεγαλύτερη τιμή ενεργοποίησης (για την συγκεκριμένη είσοδο).

Πως τα Νευρωνικά Δίκτυα με Κάψουλες Γενικεύουν σε Νέες Οπτικές Γωνίες

Ο κύριος λόγος της αποδοτικής γενίκευσης των νευρωνικών δικτύων με κάψουλες αποδίδεται στο ότι εργάζονται στον χώρο παραμέτρων των στιγμιοτύπων μιας εικόνας (ονομάζεται και χώρος αναπαράστασης γραφικών) όπου οι αλλαγές στην οπτική γωνία προκαλούν γραμμικές μεταβολές στις παραμέτρους. Έχοντας διαχωρίσει τους παράγοντες διακύμανσης της κάθε οντότητας (την πιθανότητα ύπαρξης από τις παραμέτρους στιγμιοτύπου της), και έχοντας μεταβεί σε έναν χώρο όπου αλλαγές στην οπτική γωνία αλλάζουν με γραμμικό τρόπο τα χαρακτηριστικά στιγμιοτύπου, η γενίκευση σε νέες οπτικές γωνίες έγκειται απλά στη γραμμική παρεμβολή των χαρακτηριστικών αυτών. Έτσι, στο απλό παράδειγμα που το δίκτυο έχει εκπαιδευτεί να αναγνωρίζει ένα ψηφίο στραμμένο με τυχαίο τρόπο κατά $\theta^\circ \in [-5, +5]$ τότε θα διαθέτει κάποια κάψουλα $c_i^{[L]}$ που αναγνωρίζει την ύπαρξη αυτού του ψηφίου με πιθανότητα $a_i^{[L]}$ και κωδικοποιεί τον προσανατολισμό του (μεταξύ άλλων χαρακτηριστικών) στον πίνακα $M_i^{[L]}$. Έτσι, μπορεί εύκολα να προβλέψει μέσω παρεμβολής τι επίδραση θα έχει η στρέψη του ψηφίου κατά $+10^\circ$.[32]

Για να επιτευχθεί γενίκευση σε νέες οπτικές γωνίες, όλες οι σχεδιαστικές αποφάσεις των νευρωνικών δικτύων με κάψουλες εξυπηρετούν έμμεσα ή άμεσα την εσωτερική μοντελοποίηση των αλλαγών στις διακυμάνσεις της. Για παράδειγμα, δε χρησιμοποιούνται επίπεδα συνάθροισης καθώς αυτά όπως έχουμε αναφέρει οδηγούν σε αναπαραστάσεις, ανεξάρτητες της οπτικής γωνίας. Επιπλέον, χρησιμοποιούν κάψουλες και όχι χάρτες από βαθμωτά χαρακτηριστικά καθώς μέσω των πρώτων μπορούν να αναπαριστούν σε ένα διάνυσμα πλούσια πληροφορία σχετικά με τη γεωμετρία του αντικειμένου που αναγνωρίζουν (η πληροφορία αυτή θα ήταν αδύνατο να κωδικοποιηθεί σε μια μεμονωμένη τιμή). Οι παράμετροι στιγμιοτύπου μιας κάψουλας που αφορούν το αντικείμενο που αναπαριστά αλλάζουν με προβλέψιμο τρόπο καθώς το αντικείμενο μετακινείται στην πολλαπλότητα των δυνατών απεικονίσεων (manifold of possible appearances). Συνεπώς, οι αλλαγές στην οπτική γωνία μεταφέρονται με αποδοτικό τρόπο μέσα από το σύστημα. Αντίθετα, η τιμή πιθανότητας ύπαρξης του αντικειμένου που αναγνωρίζει η κάθε κάψουλα στο πεδίο υποδοχής της επιθυμούμε να είναι όσο το δυνατό ανεξάρτητη από τον τρόπο απεικόνισης του αντικειμένου[33]. Επίσης, ανεξάρτητοι επιβάλλεται να είναι και οι πίνακες $\boldsymbol{W}$ που αποθηκεύουν τις σχέσεις μεταξύ μερών και του όλου[34]. Ένα τελευταίο παράδειγμα που συμβάλλει έμμεσα στην επίτευξη γενίκευσης σε μεταβολές οπτικής γωνίας είναι η ενσωμάτωση του αλγορίθμου δρομολόγησης με συμφωνία. Εκτός από τον ρόλο που περιγράψαμε στο να διαμορφώνει τις κάψουλες του επόμενου επιπέδου, είναι πολύ σημαντικό ότι εντοπίζει συνδιακυμάνσεις μεταξύ των αναπαραστάσεων εισόδου[35] συγκρίνοντας τα διανύσματα ψήφων μεταξύ τους μέσω του φιλτραρίσματος πολυδιάστατης σύμπτωσης.

[32] Η στρέψη του ψηφίου κατά +10 μοίρες είναι ισοδύναμη με τη στρέψη της κάμερας κατά -10 μοίρες. Συνεπώς, αποτελεί μια νέα οπτική γωνία για την οποία το δίκτυο δεν έχει εκπαιδευτεί.

[33] Στις μέχρι τώρα κύριες υλοποιήσεις, δεν υπάρχει πλήρης ανεξαρτησία σε όλο το φάσμα των πιθανών απεικονίσεων.

[34] Προφανώς, όπως και στα γραφικά υπολογιστή, ο πίνακας μετασχηματισμού που εκφράζει τις σχέσεις μέρους–όλου είναι ανεξάρτητος από την εκάστοτε οπτική γωνία του στιγμιοτύπου.

[35] Ισοδύναμα, δεν εμφανίζει το πρόβλημα της αποκλειστικής διάζευξης.

Απλό Παράδειγμα Εφαρμογής Αλγορίθμου Δρομολόγησης με Συμφωνία

Στο απλό παράδειγμα που εξετάζουμε [51], ας υποθέσουμε ότι κάθε κάψουλα έχει ως παραμέτρους στιγμιοτύπου τις τιμές που προσδιορίζουν τη θέση του αντικειμένου (x, y) και την τιμή θ του προσανατολισμού. Όπως αναφέραμε, στόχος είναι να γίνουν ανάστροφα γραφικά όπως φαίνεται και στο σχήμα 2.22. Η αναπαράσταση των τιμών μιας κάψουλας στο παράδειγμά μας θα γίνεται με ένα διάνυσμα του οποίου ο προσανατολισμός θα κωδικοποιεί τις παραμέτρους στιγμιοτύπου και το μήκος του την τιμή ενεργοποίησης. Όσο μεγαλύτερο το μήκος, τόσο πιο σίγουρη είναι η κάψουλα για την ύπαρξη της οντότητας που αναγνωρίζει. Ας υποθέσουμε ότι έχουμε δύο είδη από κάψουλες στο πρώτο επίπεδο (σχηματίζονται συνήθως από συνελικτικά επίπεδα)· αυτές που αναγνωρίζουν ορθογώνιο και αυτές που αναγνωρίζουν τρίγωνο (στο σχήμα 2.23 συμβολίζονται με πράσινα και μπλε βέλη αντίστοιχα). Τα δύο είδη από κάψουλες διαμοιράζονται στον χώρο όπως τα φίλτρα στα συνελικτικά επίπεδα. Στο σχήμα 2.23 παρατηρούμε ότι όλες οι κάψουλες έχουν μικρά διανύσματα εκτός από τις δύο που έχουν πεδίο υποδοχής το μέρος της εικόνας όπου τοποθετείται το τρίγωνο και το τετράγωνο.

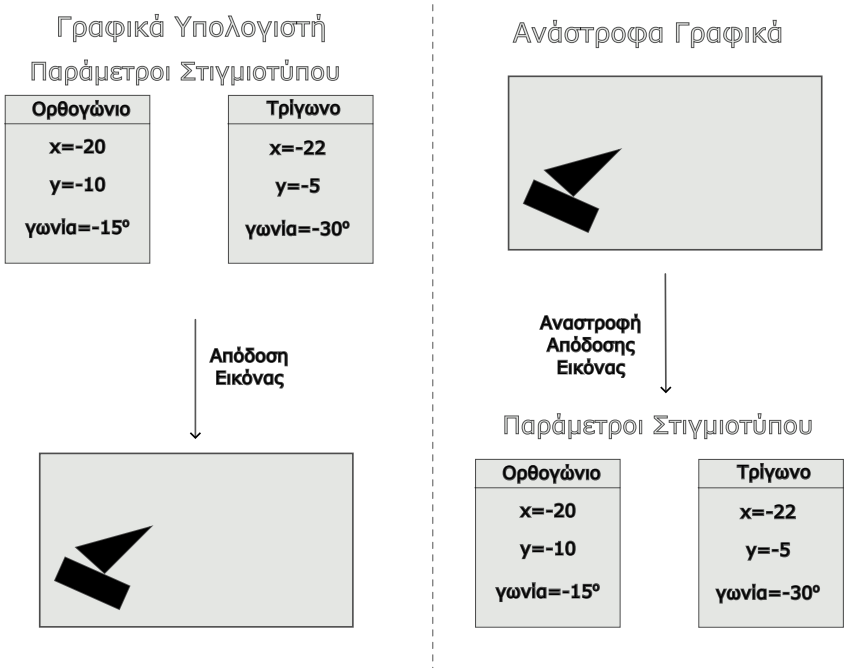
Τώρα, καλούμαστε να υπολογίσουμε τις κάψουλες του επόμενου επιπέδου γνωρίζοντας τις τιμές του προηγούμενου επιπέδου, μια διαδικασία που έχουμε ονομάσει δρομολόγηση μέσω συμφωνίας. Με αυτόν τον τρόπο, θα υπολογίσουμε τις παραμέτρους στιγμιοτύπου πιο σύνθετων αντικειμένων (βλ. σχήμα 2.24). Για τον σκοπό αυτό, κάθε κάψουλα του πρώτου επιπέδου, με βάση τις τιμές της, παράγει τόσες προβλέψεις όσες είναι οι κάψουλες του επόμενου επιπέδου που βλέπει. Ας υποθέσουμε ότι υπάρχουν δύο κάψουλες στο επόμενο επίπεδο: μια που αναπαριστά την οντότητα σπίτι και μια που αναπαριστά την οντότητα βάρκα. Όπως γίνεται κατανοητό στο σχήμα 2.25 η κάθε κάψουλα του πρώτου επιπέδου προβλέπει το διάνυσμα της κάψουλας που αναπαριστά το σπίτι και τη βάρκα με το να πολλαπλασιάζει τις τιμές της με τον αντίστοιχο πίνακα μετασχηματισμού W_{ij}.

Στην πρώτη επανάληψη του αλγορίθμου συμφωνίας, κάθε κάψουλα δρομολογεί στις κάψουλες ανώτερου επιπέδου τις προβλέψεις της ισάξια. Όμως, σύντομα αναγνωρίζεται ότι υπάρχει μεγάλη συμφωνία μεταξύ των προβλέψεων για βάρκα (βλ. σχήμα 2.25). Λόγω της συμφωνίας, το διάνυσμα της κάψουλας για τη βάρκα που σχηματίζεται από τις σύμφωνες συστάδες προβλέψεων αποκτά μεγάλο μήκος. Έτσι, επαναληπτικά, οι κάψουλες προσαρμόζουν τα βάρη δρομολόγησης ώστε τελικά να δρομολογούν όλη την ψήφο τους στην οντότητα που τους εκφράζει καλύτερα (στην περίπτωσή μας, τη βάρκα).

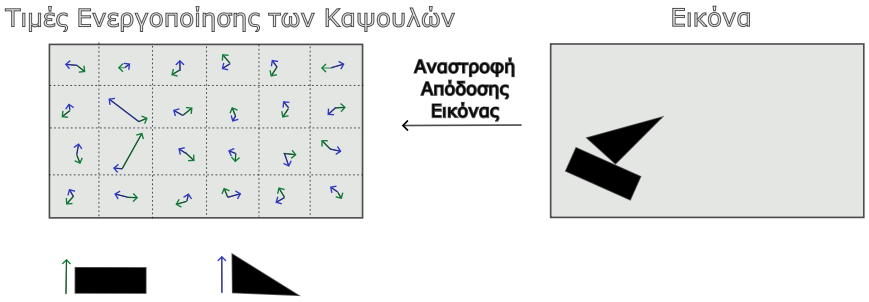
Υποθέσεις Νευρωνικών Δικτύων με Κάψουλες

Οι υποθέσεις στις οποίες βασίζονται τα νευρωνικά δίκτυα με κάψουλες είναι οι εξής:

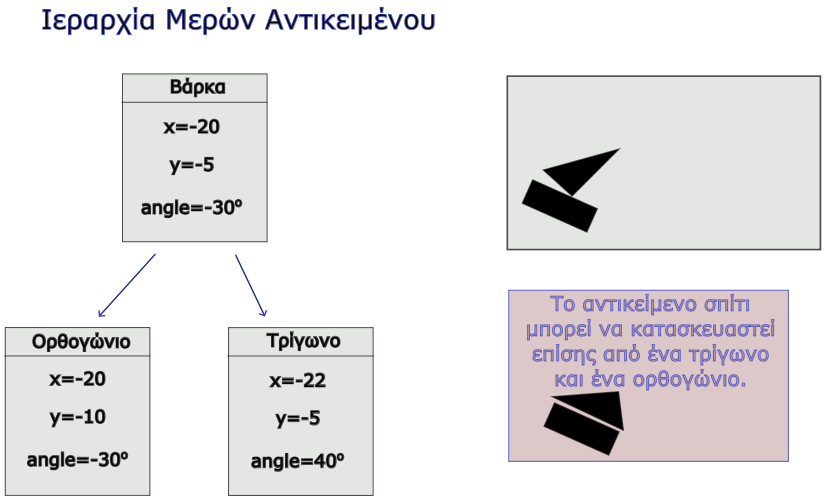
- Οι τιμές των καψουλών (M, a) εξηγούν πιστά τις όποιες μεταβολές της εικόνας εισόδου και των αντικειμένων που αυτή περιέχει (capturing equivariance). Αντίθετα, οι πίνακες βαρών W κωδικοποιούν την ανεξάρτητη (από την είσοδο) γνώση (capturing invariance).
- Οι πολυδιάστατες συμπτώσεις (high–dimensional coincidences) αποτελούν ένα κατάλληλο φίλτρο για εξαγωγή χαρακτηριστικών.
- Αλλαγές στην οπτική γωνία προκαλούν μη γραμμικές μεταβολές στα εικονοστοιχεία και



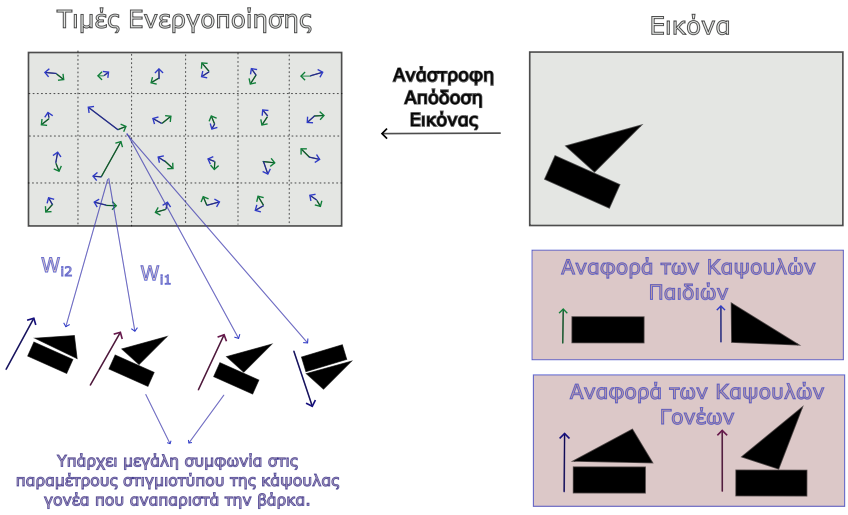
Σχήμα 2.22: Η διαδικασία ανάστροφων γραφικών που επιχειρείται από τα νευρωνικά δίκτυα από κάψουλες. Στο σχήμα, αντιπαραβάλλεται με τη διεργασία της απόδοσης εικόνας (rendering). *Παράχθηκε από το Inkscape.*



Σχήμα 2.23: Τιμές ενεργοποίησης όπως προκύπτουν από ένα συνελικτικό επίπεδο καψουλών που αναγνωρίζουν δύο οντότητες: ορθογώνιο και τρίγωνο. Οι κάψουλες αυτού του επιπέδου θα μπορούσε να είναι το αποτέλεσμα συνελικτικών επιπέδων από νευρώνες (στην περίπτωση αρχικών επιπέδων) ή το αποτέλεσμα δρομολόγησης μέσω συμφωνίας από προηγούμενο επίπεδο καψουλών. *Παράχθηκε από το Inkscape.*



Σχήμα 2.24: Παρατηρούμε ότι με τα ίδια τμήματα μπορούμε να φτιάξουμε δύο διαφορετικά αντικείμενα (βάρκα και σπίτι). Παίζει μεγάλο ρόλο λοιπόν η σωστή αναγνώριση της γεωμετρίας των μερών και το πως συνδέονται αυτά μεταξύ τους. *Παράχθηκε από το Inkscape.*



Σχήμα 2.25: Στο ανωτέρω σχήμα απεικονίζονται ενδεικτικά οι δύο προβλέψεις για τις δύο πιο ενεργές κάψουλες παιδιά (κάψουλες του προηγούμενου επιπέδου). Κάθε μια κάψουλα προσπαθεί να προβλέψει τη γεωμετρία των αντικειμένων του ανώτερου επιπέδου με βάση τη γεωμετρία της οντότητας που αναγνωρίζει. Βλέπουμε λοιπόν ότι το φιλτράρισμα συμπτώσεων υψηλής διάστασης είναι αποτελεσματικό αφού και οι δύο κάψουλες συμφωνούν στην οντότητα βάρκα. *Παράχθηκε από το Inkscape.*

γραμμικές στις σχέσεις μεταξύ αντικειμένων (ή μερών του) και της κάμερας.

- Κάθε τμήμα ενός αντικειμένου ανήκει σε ένα γενικότερο αντικείμενο (single parent assumption) και κάθε περιοχή περιέχει το πολύ ένα στιγμιότυπο του ίδιου αντικειμένου (crowding)[36].

Αδυναμίες των Νευρωνικών Δικτύων με Κάψουλες

Αυτό το είδος των νευρωνικών δικτύων, ακόμα και στην πιο πρόσφατη υλοποίησή τους από τον G. Hinton, παρουσιάζαουν ορισμένα προβλήματα. Αυτά, είναι τα εξής:

- Δεν κλιμακώνονται εύκολα σε πιο μεγάλα και σύνθετα σύνολα δεδομένων λόγω υψηλών απαιτήσεων μνήμης και της έλλειψης αποδοτικών αλγορίθμων βελτιστοποιημένων ως προς τους υπολογισμούς ενός δικτύου με κάψουλες.

- Κάψουλες που αναπαριστούν αντικείμενα τα οποία λόγω της γεωμετρίας τους έχουν ακαθόριστη πόζα, δεν μπορούν να προβλέψουν τις παραμέτρους στιγμιοτύπου των επόμενων καψουλών. Για παράδειγμα, μια κάψουλα που αναπαριστά μια ρόδα δεν μπορεί να προβλέψει τη γεωμετρία του αυτοκινήτου.

- Είναι δύσκολη η παραμετροποίηση του αλγορίθμου εύρεσης συστάδων ώστε να επιτυγχάνεται υψηλή επίδοση. Η ρύθμιση του αλγορίθμου πρέπει να είναι τέτοια ώστε να πετυχαίνει μια ισορροπία μεταξύ της πυκνότητας των συστάδων και τον αριθμό των ψήφων που περιέχουν.

2.3 Μετασχηματιστές

Σε αυτή την ενότητα θα αναφερθούμε σε μια αναδυόμενη αρχιτεκτονική νευρωνικών δικτύων, αυτή των Μετασχηματιστών (Transformers). Αν και αρχικά αναπτύχθηκε για εφαρμογές ακολουθιακών δεδομένων [80], η μεγάλη της επιτυχία οδήγησε σύντομα στον πειραματισμό της με μη–ακολουθιακά δεδομένα όπως αυτά των (στατικών) εικόνων [81, 82]. Σε αυτήν την ενότητα θα κάνουμε μια σύντομη εισαγωγή στην τεχνολογία των επαναλαμβανόμενων νευρωνικών δικτύων (Recurrent Neural Networks - RNNs) [79] και σε ορισμένα προβλήματά της [80, 83, 84]. Έπειτα, θα αναφερθούμε στις διαδοχικές βελτιώσεις - με κυριότερη αυτή της προσοχής (attention) [84] - οι οποίες τελικά διαμόρφωσαν την αρχιτεκτονική των μετασχηματιστών.

2.3.1 Επαναλαμβανόμενα Νευρωνικά Δίκτυα

Σε όλες τις αρχιτεκτονικές νευρωνικών δικτύων που έχουμε παρουσιάσει μέχρι τώρα, δε μας έχει απασχολήσει η σειρά με την οποία τροφοδοτούμε ένα σύστημα με τα παραδείγματα ενός συνόλου δεδομένων. Αυτό διότι έχουμε θεωρήσει ότι τα παραδείγματα εντός ενός συνόλου είναι διατεταγμένα με τυχαίο τρόπο, ανεξάρτητα μεταξύ τους[37]. Στην εφαρμογή αναγνώρισης τροχοφόρων οχημάτων, για παράδειγμα, η ταξινόμηση μιας εικόνας σε ένα από τα είδη τροχοφόρων οχημάτων

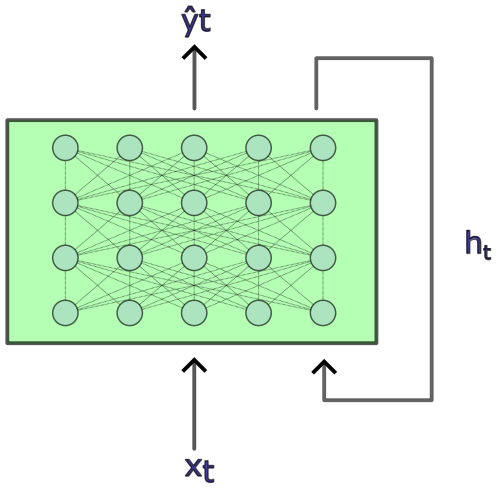
[36] Η τελευταία υπόθεση είναι αναγκαία διότι στο ίδιο οπτικό πεδίο υπάρχει μια κάψουλα για κάθε οντότητα. Αυτό είναι και το τίμημα της χρήσης της θέσης των καψουλών μέσα στο δίκτυο για να προσδιοριστεί η ακριβής θέση των οντοτήτων που αναπαριστούν (όπως στα συνελικτικά νευρωνικά δίκτυα χωρίς επίπεδα συνάθροισης).

[37] Η ιδιότητα της ανεξαρτησίας είναι η μία από τις δύο θεμελιώδεις υποθέσεις των συνόλων δεδομένων που χρησιμοποιούνται σε εφαρμογές μηχανικής μάθησης. Η δεύτερη υπόθεση είναι ότι όλα τα δείγματα ακολουθούν την ίδια κατανομή πιθανότητας.

δε θα προσέδιδε καμία πληροφορία για την επόμενη προς ταξινόμηση εικόνα. Παρόλα αυτά, στα ακολουθιακά δεδομένα η ανεξαρτησία μεταξύ των δειγμάτων δεν ισχύει. Με άλλα λόγια, τα επιμέρους δείγματα συνδέονται μεταξύ τους έτσι ώστε η γνώση του ενός να προσδίδει πληροφορία για το άλλο. Μάλιστα, οι σχέσεις αυτές είναι συνήθως εντονότερες όταν η απόσταση μεταξύ των δειγμάτων στην ακολουθία είναι μικρή. Λόγου χάρη, στην ακολουθία τιμών θερμοκρασίας ενός δωματίου, όπως προκύπτει από την περιοδική μέτρηση ενός αισθητήρα, μπορεί κανείς να προβλέψει τη μελλοντική τιμή βασιζόμενος στις αμέσως προηγούμενες μετρήσεις. Σε γενικότερες γραμμές, όλες οι χρονοσειρές μπορούν να ενταχθούν στην κατηγορία των ακολουθιακών δεδομένων.

Η ύπαρξη τέτοιων ακολουθιακών δεδομένων προδιέθεσε την ανάπτυξη αρχιτεκτονικών νευρωνικών δικτύων που αναγνωρίζουν μοτίβα σε αυτά. Έτσι, γίνεται αξιοποίηση της κρυφής (hidden) πληροφορίας που αποκαλύπτουν οι μη–ανεξάρτητες σχέσεις μεταξύ των δειγμάτων. Αν και ήδη από το 1943 οι Warren McCulloch και Walter Pitts [20] περιέγραφαν την ιδέα ύπαρξης κυκλικών (επαναλαμβανόμενων) νευρωνικών δικτύων, αυτή άρχισε να λαμβάνει πρακτική υπόσταση υπό το όνομα «επαναληπτικά νευρωνικά δίκτυα» αργότερα, με τα έργα των David E. Rumelhart et al. [79] και του Michael I. Jordan [85].

Σε μια πιο τυπική περιγραφή των επαναληπτικών νευρωνικών δικτύων, πρόκειται για το είδος αυτό που μπορεί να χειριστεί αποτελεσματικά ακολουθίες μεταβλητού μήκους [51]. Προκειμένου να το επιτύχει αυτό, απαιτείται ένας μηχανισμός ο οποίος θα αξιοποιεί τις σχέσεις αλληλεξάρτησης μεταξύ των επιμέρους δειγμάτων. Αυτό επιτυγχάνεται διαδίδοντας την πληροφορία που έχει εξαχθεί κατά την επεξεργασία των προηγούμενων δειγμάτων μιας ακολουθίας, στους κόμβους επεξεργασίας των επόμενων [86]. Για αυτό τον λόγο, καταλήγουμε σε μια αρχιτεκτονική όπως αυτή ενός νευρωνικού δικτύου πρόσθιας τροφοδότησης αλλά με επιπλέον, ανάστροφες ακμές [51].

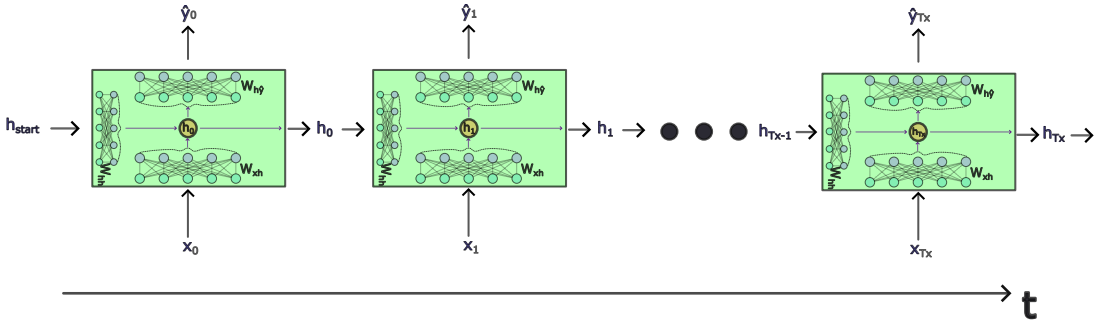


Σχήμα 2.26: Στο ανωτέρω σχήμα απεικονίζονται η αρχιτεκτονική ενός επαναληπτικού νευρωνικού δικτύου. Το δίκτυο δέχεται ένα δείγμα ακολουθίας τη χρονική στιγμή t και παράγει δύο εξόδους: την $\hat{y}_t$ και το διάνυσμα κρυφής κατάστασης h_t (το οποίο και χρησιμοποιεί στην επόμενη χρονική στιγμή). *Παράχθηκε από το Inkscape.*

Η αφαιρετική αρχιτεκτονική ενός επαναληπτικού νευρωνικού δικτύου φαίνεται στο σχήμα 2.26. Ας υποθέσουμε ότι στην είσοδο δίνεται ως παράδειγμα[38] μια ακολουθία $\underline{X} = [X_1, X_2, \ldots, X_{T_x}]$ α-

[38]Στο πλαίσιο των επαναληπτικών νευρωνικών δικτύων, με τον όρο παράδειγμα ενός συνόλου δεδομένων θα

ποτελούμενη από $X_1, X_2, \ldots, X_{T_x}$ επιμέρους δείγματα (όπου το καθένα αποτελείται από $d_{features}$ χαρακτηριστικά, δηλαδή: $X_i \in \Re^{d_{features}}$). Τότε, σειριακά, σε κάθε χρονικό βήμα (ή καρέ) t, το νευρωνικό δίκτυο θα δέχεται σαν είσοδο ένα δείγμα X_t και θα παράγει μια έξοδο $\hat{y}_t$. Η πρακτική διαφοροποίηση με τα νευρωνικά δίκτυα πρόσθιας τροφοδότησης έγκειται στο ότι εκτός από αυτά τα διανύσματα, ένα επαναληπτικό νευρωνικό δίκτυο παράγει σε κάθε βήμα και μια δεύτερη έξοδο, το διάνυσμα κατάστασης h_t το οποίο αποθηκεύει (σαν κυψέλη μνήμης) την πληροφορία των προηγούμενων δειγμάτων και τροφοδοτείται σαν είσοδο στο επόμενο καρέ (μέσω της ανάστροφης ακμής). Συνεπώς, στους υπολογισμούς του βήματος $t+1$, θα ληφθεί υπόψη όχι μόνο το παρόν δείγμα X_{t+1} αλλά και η πρότερη χρήσιμη πληροφορία, κωδικοποιημένη στο διάνυσμα κατάστασης h_t[39]. Τέλος, για λόγους κατανόησης θα εξυπηρετούσε να επισημάνουμε πως οι σειριακοί υπολογισμοί που προαναφέραμε μπορούν να «ξετυλιχτούν» στον χρόνο σχηματίζοντας το διάγραμμα 2.27.



Σχήμα 2.27: Στο ανωτέρω σχήμα απεικονίζεται η αρχιτεκτονική ενός επαναληπτικού νευρωνικού δικτύου «ξετυλιγμένου» στον χρόνο. Στην αρχή τροφοδοτούμε το δίκτυο με μια αρχική κρυφή κατάσταση (μπορεί να είναι τυχαία αρχικοποιημένη). Τροφοδοτούμε το δίκτυο με τα δείγματα της ακολουθίας εισόδου, ένα ανά χρονική στιγμή t (στο σχήμα έχουμε υποθέσει ότι $d_{features} = 5$). Σε κάθε βήμα, μετακινούμαστε μια θέση δεξιά (στον άξονα του χρόνου). Να σημειώσουμε ότι με πράσινο χρώμα απεικονίζονται οι κόμβοι εισόδου ενώ με μοβ οι κόμβοι εξόδου. *Παράχθηκε από το Inkscape.*

Με μαθηματικούς όρους, το διάνυσμα κατάστασης τη χρονική στιγμή t προκύπτει ως συνάρτηση της προηγούμενης (κρυφής) κατάστασης και του δείγματος εισόδου, δηλαδή:

$$h_t = f_{W_h}(x_t, h_{t-1}) = tanh(\boldsymbol{W_{hh}^T} \times h_{t-1} + \boldsymbol{W_{xh}^T} \times x_t). \tag{2.32}$$

Επιπλέον, για την έξοδο σε κάθε βήμα ισχύει:

$$\hat{y}_t = f_{W_{\hat{y}}}(x_t, h_{t-1}) = \boldsymbol{W_{h\hat{y}}^T} \times h_t, \tag{2.33}$$

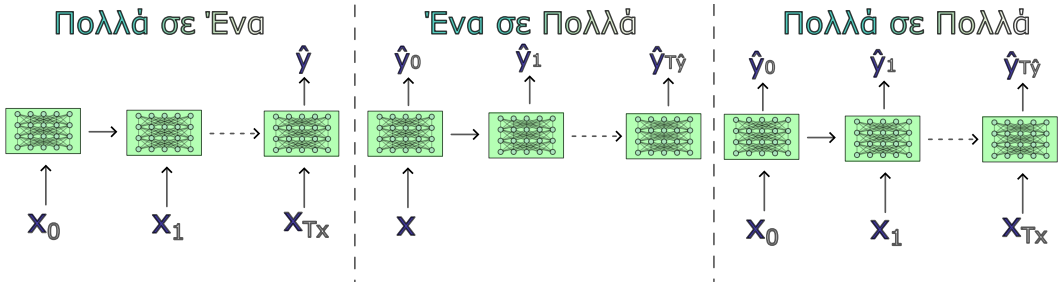
όπου το h_t ενσωματώνει τα x_t, h_{t-1}.

Να σημειωθεί ότι οι παράμετροι των δύο συναρτήσεων, $f_{W_{\hat{y}}}$ και f_{W_h}, δε μεταβάλλονται ανάλογα με την ακολουθία εισόδου αλλά εκπαιδεύονται μέσω ενός τροποποιημένου αλγορίθμου οπι-

αναφερόμαστε σε μια ακολουθία από δείγματα.

[39] Το διάνυσμα h_t θα μπορούσε στην απλούστερη περίπτωση να είναι ίδιο με το $\hat{y}_t$. Στις περισσότερες σύνθετες εφαρμογές όμως, παρατηρούνται οφέλη όταν τα διανύσματα αυτά είναι διαφορετικά.

σθοδιάδοσης (οπισθοδιάδοση στο χρόνο - Back Propagation Through Time) έτσι ώστε τα βάρη και τα δυναμικά πόλωσής τους να λάβουν τιμές οι οποίες θα μοντελοποιούν καλύτερα τη σχέση εισόδου εξόδου. Η ενημέρωση των παραμέτρων πραγματοποιείται τουλάχιστον ανά T_x χρονικές στιγμές. Η έννοια των συναρτήσεων $f_{W_{\hat{y}}}$ και f_{W_h} είναι ίδια με την κλασική περίπτωση νευρωνικού δικτύου όπου συμβολίζαμε τη συνάρτηση που αυτό μοντελοποιεί ως $\mathcal{F}(X; \overline{W}, \overline{b}) : X \to \hat{Y}$. Η διαφορά έγκειται ότι τώρα έχουμε δύο τέτοιες συναρτήσεις: μια που έχει ως έξοδο την επόμενη κρυφή κατάσταση h και μια που έχει έξοδο την παρατηρήσιμη τιμή $\hat{y}$.



Σχήμα 2.28: Οι τρείς παραλλαγές επαναληπτικών νευρωνικών δικτύων ανάλογα με την είσοδο και την έξοδό τους. Σημειώνουμε ότι στην δεξιότερη παραλλαγή (Πολλά σε Πολλά) υποχρεωτικά με την απεικονιζόμενη αρχιτεκτονική ισχύει $T_x = T_{\hat{y}}$. *Παράχθηκε από το Inkscape.*

Υπάρχουν διάφορες παραλλαγές της αρχιτεκτονικής των επαναληπτικών νευρωνικών δικτύων ανάλογα με την εφαρμογή που διαχειρίζονται. Οι πιο βασικές κατηγορίες είναι οι εξής (βλ. σχήμα 2.28):

- Πολλά σε Ένα: Πρόκειται για το σύστημα που δέχεται σαν είσοδο μια ακολουθία και παράγει σαν έξοδο μια τιμή ή ένα διάνυσμα μη–ακολουθιακού χαρακτήρα. Παράδειγμα εφαρμογής που απαιτεί αυτή την κατηγορία επαναληπτικών νευρωνικών δικτύων είναι η ταξινόμηση συναισθήματος (sentiment classification) όπου δοθέντος ενός κειμένου, καλείται παραδείγματος χάρη να το χαρακτηρίσει σαν θετικό, αρνητικό ή ουδέτερο.

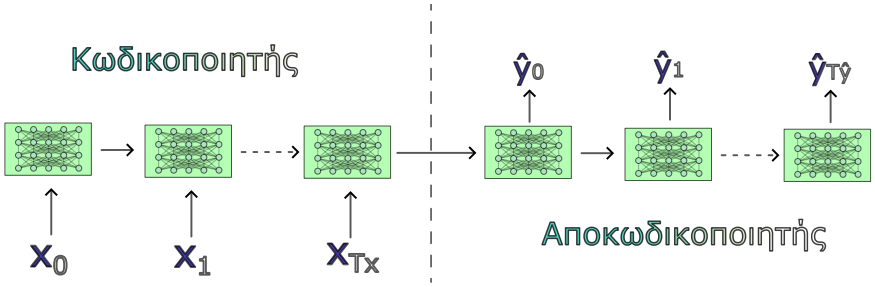
- Ένα σε Πολλά: Σε αυτή την κατηγορία, η είσοδος δεν έχει ακολουθιακή οργάνωση αλλά η έξοδος έχει. Παράδειγμα εφαρμογής αποτελεί η αυτόματη παραγωγή λεζάντων σε εικόνες (μη–ακολουθιακά δεδομένα).

- Πολλά σε Πολλά: Είναι ο μετασχηματισμός μιας ακολουθίας σε μια άλλη από μοντέλα τα οποία ονομάζονται γενικά «ακολουθία–σε–ακολουθία» (seq2seq models). Χαρακτηριστική εργασία που ανήκει σε αυτήν την κατηγορία είναι αυτή της μετάφρασης από μια γλώσσα σε μια άλλη.

Όλες οι εφαρμογές μοντελοποίησης ακολουθιών, ανεξάρτητα από τη λειτουργία τους, πρέπει να σχεδιάζονται έτσι ώστε να:

1. Διαχειρίζονται ακολουθίες μεταβλητού μήκους. Δηλαδή οι ακολουθίες εισόδου $\underline{X} = [X_1, X_2, \ldots, X_{T_x}]$ ή εξόδου $\underline{\hat{Y}} = [\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_{T_{\hat{y}}}]$ που διαχειρίζεται το σύστημα να μην έχουν υποχρεωτικά όλες το ίδιο T[40].

[40] Υποχρεωτικά, κάθε δείγμα της ακολουθίας εισόδου (και εξόδου, αντίστοιχα) κωδικοποιείται με τον ίδιο αριθμό χαρακτηριστικών ($d_{features}$), δηλαδή: $X_i \in \Re^{d_{features_X}}, \forall i \in [1, T_x]$ και $\hat{Y}_i \in \Re^{d_{features_{\hat{Y}}}}, \forall i \in [1, T_{\hat{y}}]$



Σχήμα 2.29: Μια διαφορετική αρχιτεκτονική για την παραλλαγή «Πολλά σε Πολλά». Ονομάζεται αρχιτεκτονική Κωδικοποιητή - Αποκωδικοποιητή. Παρατηρούμε ότι όλη η γνώση της ακολουθίας εισόδου μεταφέρεται στον αποκωδικοποιητή μέσα από μια ακμή (ένα διάνυσμα κρυφής κατάστασης προκαθορισμένου μεγέθους). Σημειώνουμε ότι μπορεί να ισχύει $T_x \neq T_{\hat{y}}$. *Παράχθηκε από το Inkscape.*

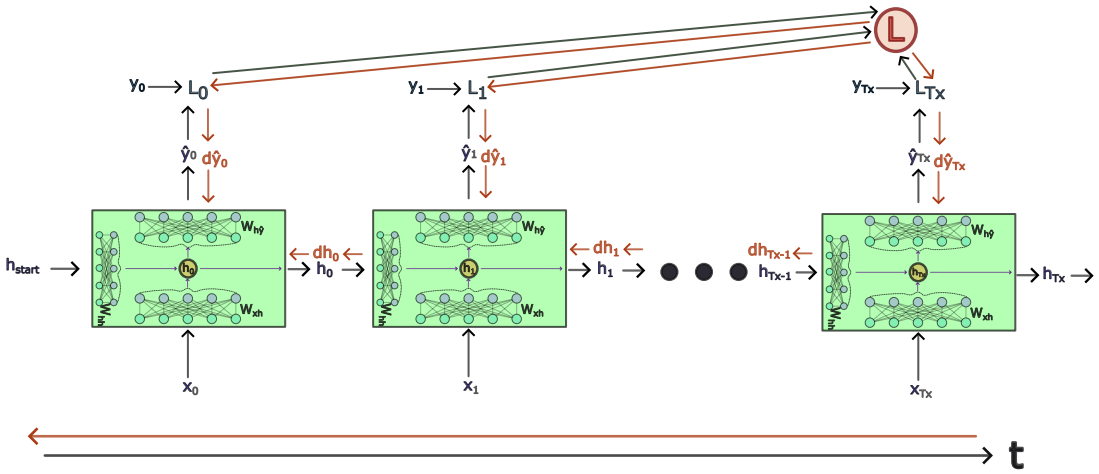
2. Ανιχνεύουν εξαρτήσεις μεταξύ δειγμάτων στην ακολουθία που μπορεί να έχουν μεγάλη απόσταση μεταξύ τους.

3. Διατηρούν την πληροφορία σχετικά με τη σειρά των στοιχείων στην ακολουθία. Αυτό έχει πρωτεύουσα σημασία σε τέτοιες εφαρμογές αφού για παράδειγμα, η αλλαγή της σειράς των λέξεων σε μια πρόταση μπορεί να αλλάξει ριζικά την ερμηνεία της.

4. Διαμοιράζονται παραμέτρους μεταξύ των χρονικών στιγμών (π.χ. μέσω διανυσμάτων καταστάσεων h) ώστε να μπορούν να εντοπίζουν μακρινές εξαρτήσεις. [86]

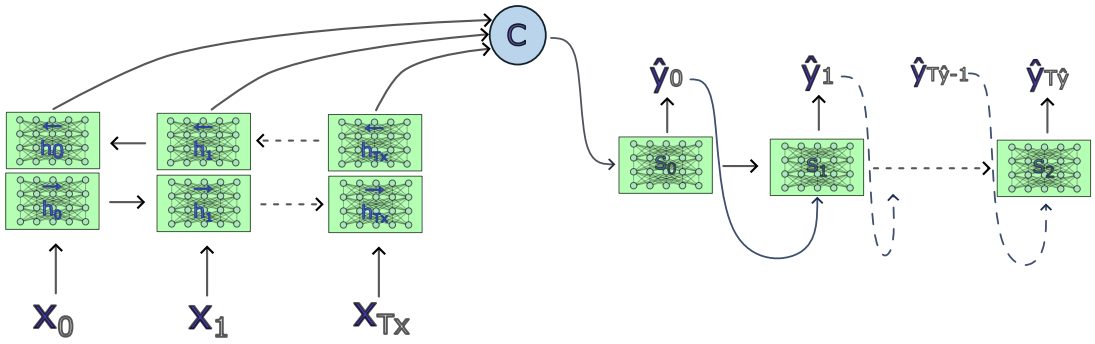
Προβλήματα Επαναλαμβανόμενων Νευρωνικών Δικτύων

Τα επαναληπτικά νευρωνικά δίκτυα, στην απλή τους μορφή που παρουσιάσαμε, μπορούν και πληρούν επαρκώς τα ανωτέρω κριτήρια σε απλές εφαρμογές. Παρόλα αυτά, σε πιο σύνθετες εφαρμογές δημιουργούνται ορισμένα προβλήματα. Η κύρια πηγή αυτών των προβλημάτων είναι το μεγάλο μήκος των ακολουθιών που καλούνται τα επαναληπτικά νευρωνικά δίκτυα να μοντελοποιήσουν.



Σχήμα 2.30: Σχήμα στο οποίο απεικονίζεται η οπισθοδιάδοση του σφάλματος στον χρόνο (Back Propagation Through Time). Με γκρι βέλη απεικονίζεται η πρόσθια διάδοση για τον υπολογισμό της συνάρτησης απώλειας ενώ με κόκκινα βέλη η οπισθοδιάδοση (η οποία συμβαίνει αφού έχουν υπολογιστεί όλα τα $\hat{y}$ για την συγκεκριμένη ακολουθία εξόδου). *Παράχθηκε από το Inkscape.*

Το πρώτο πρόβλημα είναι αυτό των εξαφανιζόμενων ή εκρηγνυόμενων κλίσεων (vanishing/exploding gradients). Σε αδρές γραμμές, όπως φαίνεται και στο σχήμα 2.30, οι μερικές παράγωγοι του σφάλματος ως προς τις παραμέτρους διαδίδονται μέσω του αλγορίθμου οπισθοδιάδοσης κλίσης όχι μόνο στον εγκάρσιο άξονα εισόδου–εξόδου αλλά και στον διαμήκη άξονα του χρόνου από το τελευταίο καρέ στο πρώτο. Έτσι, αν οι ιδιοτιμές λ του πίνακα $\boldsymbol{W_{hh}}$ είναι λίγο μεγαλύτερες της μονάδος, οι διαδοχικοί πολλαπλασιασμοί του σφάλματος με την ποσότητα $\boldsymbol{W_{hh}}$ για τον υπολογισμό της κλίσης ως προς τις παραμέτρους στα αρχικά καρέ θα διογκώσει το σφάλμα κατά λ^T οδηγώντας στην υπερχείλιση (overflow). Αντίστοιχα, στην περίπτωση που οι ιδιοτιμές είναι μικρότερες της μονάδος, έχουμε υποχείλιση (underflow). Επειδή λοιπόν το σφάλμα αδυνατεί να διαδοθεί στα πρώτα καρέ μακρινών ακολουθιών, το δίκτυο αδυνατεί να αξιοποιήσει εξαρτήσεις που έχουν μεγάλη απόσταση μεταξύ τους. Για παράδειγμα, ένα σύστημα πρόβλεψης επόμενης λέξης υλοποιημένο με ένα απλό επαναληπτικό νευρωνικό δίκτυο (vanilla RNN) θα αδυνατούσε να μεταφράσει την πρόταση «Μεγάλωσα στην Ελλάδα, ... άρα μιλάω άπταιστα 'θέση-λέξης-για-πρόβλεψη'.» αν μεταξύ της λέξης «Ελλάδα» και της πρόβλεψης παρεμβάλονταν πολλές λέξεις.



Σχήμα 2.31: Σχήμα στο οποίο απεικονίζεται μια πιο σύνθετη αρχιτεκτονική ενός επαναληπτικού νευρωνικού δικτύου τύπου κωδικοποιητή–αποκωδικοποιητή. Σε αυτό, το διάνυσμα (context) που κωδικοποιεί την πληροφορία της ακολουθίας εισόδου σχηματίζεται από τις κρυφές καταστάσεις όλων των προηγούμενων δειγμάτων. Ο κωδικοποιητής αποτελείται από δύο επίπεδα επαναληπτικών δικτύων (ένα που δέχεται την ακολουθία με την χρονική σειρά και ένα με την ανάποδη). Η αρχιτεκτονική αυτή του κωδικοποιητή ονομάζεται «αμφίδρομο επαναληπτικό νευρωνικό δίκτυο» (Bidirectional RNN). Ο αποκωδικοποιητής αποτελεί ένα μοντέλο αυτοπαλινδρόμησης αφού η (κύρια) έξοδός του τροφοδοτείται αυτούσια σε μια από τις εισόδους για να χρησιμοποιηθεί την επόμενη χρονική στιγμή. *Παράχθηκε από το Inkscape.*

Το δεύτερο πρόβλημα αφορά κυρίως τα επαναληπτικά νευρωνικά δίκτυα τύπου «ακολουθία–σε–ακολουθία» και δη αυτά σε μορφή κωδικοποιητή–αποκωδικοποιητή (βλ. σχήμα 2.29 και σχήμα 2.31). Σε αυτά τα δίκτυα, ο κωδικοποιητής διαβάζει την ακολουθία εισόδου και σχηματίζει ένα διάνυσμα κατάστασης προκαθορισμένου μήκους το οποίο ενσωματώνει την πληροφορία αυτή. Στην συνέχεια, ο αποκωδικοποιητής λαμβάνει αυτό το διάνυσμα ως μοναδική είσοδο και παράγει την ακολουθία εξόδου. Έχει όμως δειχθεί ότι η επίδοση ενός τέτοιου συστήματος μειώνεται σημαντικά με την αύξηση του μήκους της ακολουθίας εισόδου [87]. Θα μπορούσαμε να ισχυριστούμε ότι η μείωση της επίδοσης οφείλεται στην συμφόρηση (bottleneck) που προκαλείται με την απαίτηση ότι όλη η πληροφορία της ακολουθίας εισόδου να κωδικοποιείται σε ένα

σταθερού–μήκους διάνυσμα κατάστασης [84].

Το τρίτο πρόβλημα που γίνεται όλο και πιο έντονο σε σύνθετες εφαρμογές με μεγάλες ακολουθίες είναι οι αργοί χρόνοι εκπαίδευσης και πρόβλεψης. Τα επαναληπτικά νευρωνικά δίκτυα είναι σειριακής φύσεως με αποτέλεσμα να μην μπορεί να τροφοδοτηθεί κάθε δείγμα της ακολουθίας παράλληλα. Αντίθετα, για να τροφοδοτηθεί το επόμενο στην σειρά δείγμα θα πρέπει να έχει ολοκληρωθεί η επεξεργασία του προηγούμενου. Αυτό, αν και δεν είναι σοβαρό πρόβλημα σε απλές εφαρμογές με μικρό υπολογιστικό κόστος και μικρού μήκους ακολουθίες αποτελεί ένα ανυπέρβλητο εμπόδιο στην εκπαίδευση σύνθετων εφαρμογών που μοντελοποιούν ακολουθίες με εκτατοντάδες δείγματα η κάθε μια.

Λύσεις στα Προβλήματα Επαναλαμβανώμενων Νευρωνικών Δικτύων

Έχουν αναπτυχθεί διάφορες τεχνικές για την μετρίαση των ανωτέρω προβλημάτων. Ενδεικτικά, για το πρόβλημα της υπερχείλισης ενδείκνεται η περικοπή της μεγάλης τιμής (σε ένα μικρότερο νούμερο) ώστε να μπορεί να αποθηκευθεί στον χώρο μνήμης που έχει διατεθεί για αυτή [86]. Ανάλογα, για το πρόβλημα της υποχείλισης, αυτό βελτιώνεται με καλύτερη επιλογή συνάρτησης ενεργοποίησης και με προσεκτικότερη αρχικοποίηση των πινάκων βαρών [86]. Τέλος, και για τις δύο υποπεριπτώσεις, μια νέα αρχιτεκτονική επαναληπτικών νευρωνικών δικτύων με όνομα Μακροπρόθεσμη–Βραχυπρόθεσμη Μνήμη (Long–Short Term Memory - LSTM) [83] μπορεί να αυξήσει σημαντικά το μήκος των ακολουθιών που μπορεί ένα τέτοιο είδος δικτύου να διαχειριστεί[41].

Σε ό,τι αφορά το πρόβλημα της συμφόρησης της κωδικοποιημένης πληροφορίας, έχει προταθεί η ενίσχυση των επαναληπτικών νευρωνικών δικτύων με έναν μηχανισμό προσοχής (attention mechanism) [79]. Όπως θα αναλύσουμε στην συνέχεια, αυτός ο μηχανισμός επιτρέπει στον αποκωδικοποιητή επιλεκτικά, σε κάθε βήμα, να βλέπει τις προηγούμενες κρυφές καταστάσεις που θεωρεί πιο χρήσιμες για τον υπολογισμό της εξόδου στο βήμα αυτό[42].

Αναζητώντας λύση για το τρίτο πρόβλημα, η επιστημονική κοινότητα, αντλώντας στοιχεία από τον μηχανισμό προσοχής, οδηγήθηκε σε μια εξ ολοκλήρου νέα αρχιτεκτονική που διαφέρει από τα επαναληπτικά νευρωνικά δίκτυα. Αυτή ονομάζεται μετασχηματιστές (transformers) [80] και όπως θα περιγράψουμε στην συνέχεια, τα δείγματα σε κάθε ακολουθία δεν τροφοδοτούνται στο μοντέλο σειριακά αλλά παράλληλα. Επιπλέον, μπορεί να κλιμακώνει εύκολα με αποτέλεσμα να δύναται να μοντελοποιεί μεγάλες σε μήκος ακολουθίες με μακρυνές εξαρτήσεις.

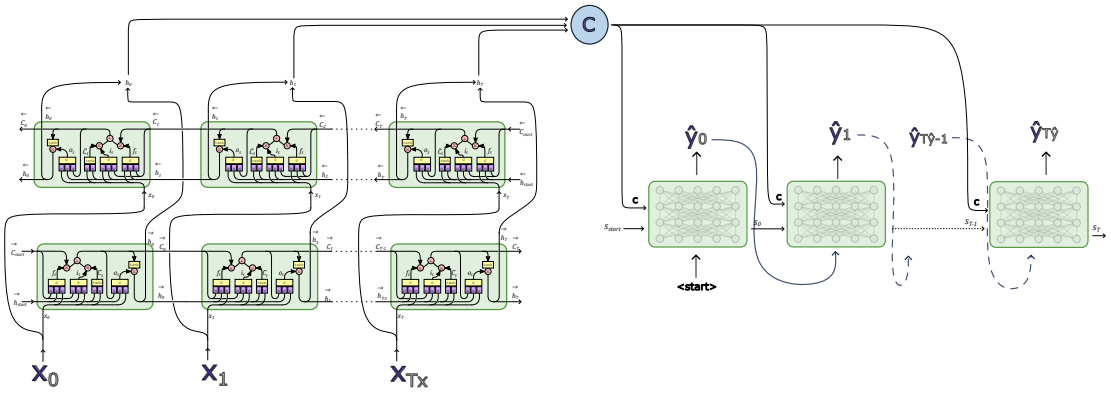
2.3.2 Μηχανισμός Προσοχής

Αν και ο μηχανισμός προσοχής έχει αποβεί χρήσιμος σε μια πληθώρα από εφαρμογές, αρχικά αναπτύχθηκε στο πλαίσιο του επαναληπτικού νευρωνικού δικτύου τύπου κωδικοποιητή αποκωδικοποιητή για να λύσει το πρόβλημα της συμφόρησης σε εφαρμογές μεταφράσεων. Η γενική ιδέα είναι ότι για τον υπολογισμό της εξόδου $\hat{y}_t$, κρίνεται σκόπιμο το δίκτυο να μπορεί να εστιάσει την προσοχή του σε συγκεκριμένα δείγματα εισόδου που θεωρεί πιο σχετικά. Τα δείγματα αυτά

[41] Η λύση αυτή δεν αποτελεί πανάκεια καθώς δεν εξαλείφει το πρόβλημα και επίσης με αυτή δεν μπορεί να γίνει μεταφορά γνώσης (transfer learning) [88].

[42] Η λύση αυτή μπορεί να συνδυαστεί με την αρχιτεκτονική επαναληπτικών νευρωνικών δικτύων «Μακροπρόθεσμης –Βραχυπρόθεσμης Μνήμης».

θα διαφέρουν από καρέ σε καρέ αφού διαφορετικές λέξεις εξόδου συσχετίζονται με διαφορετικές λέξεις εισόδου.



Σχήμα 2.32: Σχήμα στο οποίο απεικονίζεται μια ακόμα σύνθετη αρχιτεκτονική ενός επαναληπτικού νευρωνικού δικτύου τύπου κωδικοποιητή–αποκωδικοποιητή (peaky encoder-decoder). Σε αυτό, το διάνυσμα (context) που κωδικοποιεί την πληροφορία της ακολουθίας εισόδου σχηματίζεται από τις κρυφές καταστάσεις όλων των προηγούμενων δειγμάτων και είναι ορατό σε κάθε βήμα του αποκωδικοποιητή. Η αρχιτεκτονική του κωδικοποιητή ονομάζεται «αμφίδρομο επαναληπτικό νευρωνικό δίκτυο από μονάδες Μακροπρόθεσμης–Βραχυπρόθεσμης Μνήμης» (Bidirectional LSTM). Ο αποκωδικοποιητής μπορεί και αυτός να κατασκευάζεται από τροποποιημένες μονάδες Μακροπρόθεσμης–Βραχυπρόθεσμης Μνήμης που θα ενσωματώνουν πλήρως διασυνδεδεμένα επίπεδα για τον υπολογισμό της εξόδου από τα διανύσματα κατάστασης. *Παράχθηκε από το Inkscape.*

Στο σχήμα 2.32 φαίνεται αναλυτικά ένα τέτοιο σύστημα χωρίς τον μηχανισμό προσοχής. Παρατηρούμε ότι ο κωδικοποιητής αποτελείται από δύο επαναληπτικά νευρωνικά δίκτυα: ένα με δεξιά κατεύθυνση και ένα με την αντίθετη κατεύθυνση, στοιβαγμένα το ένα πάνω στο άλλο[43] [89]. Το κάθε ένα παράγει ένα διάνυσμα κατάστασης ($\overrightarrow{h}_t$ και $\overleftarrow{h}_t$ αντίστοιχα) σε κάθε καρέ τα οποία μετά τα ενώνουμε (concatenate) δηλαδή

$$h_t = \begin{bmatrix} \overrightarrow{h}_t \\ \overleftarrow{h}_t \end{bmatrix}.$$

Αυτό γίνεται επειδή επιθυμούμε το διάνυσμα κατάστασης σε κάθε σημείο να ενσωματώνει τόσο την πληροφορία για τα προηγούμενα δείγματα ($\overrightarrow{h}_t$) όσο και την πληροφορία για τα επόμενα ($\overleftarrow{h}_t$) [84]. Μετά την δημιουργία των διανυσμάτων κατάστασης, αυτά χρησιμεύουν για την κατασκευή ενός διανύσματος συμφραζομένων (context vector) σταθερού μήκους το οποίο συμπυκνώνει[44] την πληροφορία από όλα τα διανύσματα κατάστασης Δηλαδή:

$$c = q(h_1, h_2, \ldots, h_{T_x})$$

όπου q μια μη–γραμμική συνάρτηση. Τέλος, ο αποκωδικοποιητής (όπως περιγράφεται από τους Bahdanau D. et al. [84]) λαμβάνει σε κάθε βήμα την προηγούμενη πρόβλεψη $\hat{y}_{t-1}$, την προηγούμενη κατάσταση που στον αποκωδικοποιητή συμβολίζεται με s_{t-1} και το διάνυσμα συμφραζομένων

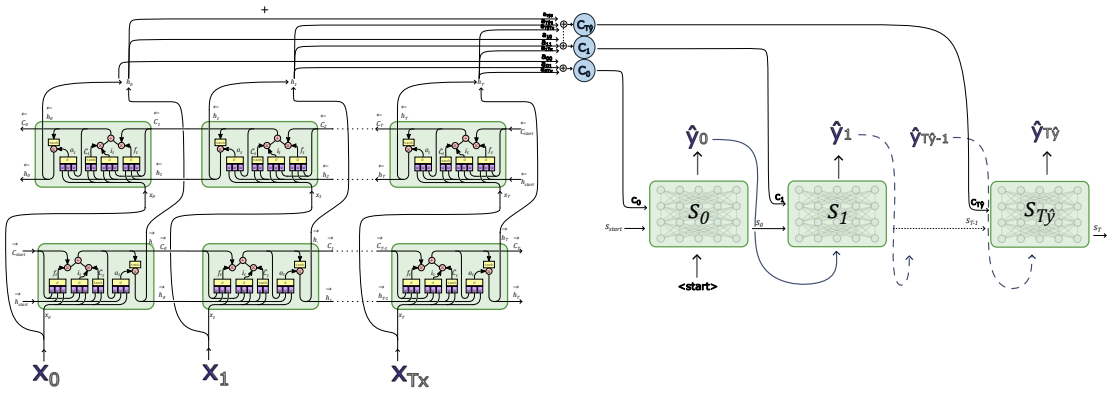
[43] Η αρχιτεκτονική αυτή ονομάζεται αμφίδρομο επαναληπτικό νευρωνικό δίκτυο (bidirectional recurrent neural network).

[44] Η συμπύκνωση προκαλεί απώλεια πληροφορίας που δημιουργεί το πρόβλημα της συμφόρησης.

c για να εξάγει την κατάσταση s_t. Δηλαδή, είναι:

$$s_t = f(s_{t-1}, \hat{y}_{t-1}, c)$$

Έπειτα, με αντίστοιχο τρόπο όπως παρουσιάσαμε με τη σχέση 2.33, υπολογίζουμε την πρόβλεψη $\hat{y}_t$ από το διάνυσμα s_t.



Σχήμα 2.33: Σχήμα στο οποίο απεικονίζεται η αρχιτεκτονική ενός επαναληπτικού νευρωνικού δικτύου τύπου κωδικοποιητή–αποκωδικοποιητή χρησιμοποιώντας μηχανισμό προσοχής. Παρατηρούμε ότι σε κάθε καρέ του αποκωδικοποιητή υπάρχει διαθέσιμο ένα ξεχωριστό διάνυσμα συμφραζομένων (context vector). Το διάνυσμα σε κάθε καρέ σχηματίζεται ως σταθμισμένο άθροισμα των κρυφών καταστάσεων h_t ανάλογα με το σε ποια σημεία της ακολουθίας εισόδου πρέπει να εστιάσει ο αποκωδικοποιητής για το συγκεκριμένο καρέ εξόδου. *Παράχθηκε από το Inkscape.*

Στο σχήμα 2.33 φαίνεται το ίδιο σύστημα αλλά με μηχανισμό προσοχής. Πλέον, δεν υπάρχει συμφόρηση καθώς δεν χρησιμοποιείται μόνο ένα διάνυσμα συμφραζομένων για να κωδικοποιήσει την πληροφορία όλων των δειγμάτων της ακολουθίας εισόδου. Συνεπώς, έχουμε:

$$s_t = f(s_{t-1}, y_{t-1}, c_t)$$

Όπου τα διανύσματα συμφραζομένων υπολογίζονται ως σταθμισμένα αθροίσματα των διανυσμάτων κατάστασης, δηλαδή:

$$c_t = \sum_{j=1}^{j=T_x} \alpha_{tj} * h_j.$$

Τα βάρη α_{tj} αναπαριστούν την σημασία του εκάστοτε διανύσματος κατάστασης h_j στους υπολογισμούς της εξόδου την χρονική στιγμή t. Με άλλα λόγια, στην εφαρμογή της μετάφρασης δείχνουν σε ποιες λέξεις εισόδου πρέπει να εστιάσει σε κάθε βήμα ο αποκωδικοποιητής.

Σε τελική ανάλυση, για τα βάρη α_{tj} επιβάλλουμε να ισχύει $\sum_{j=1}^{j=T_x} \alpha_{tj} = 1$ (κανονικοποίηση) ώστε να δημιουργούν μια κατανομή πιθανότητας. Αυτό το επιτυγχάνουμε ορίζοντας τις ενέργειες ϵ_{tj} ως τιμές σημασίας που έχει το δείγμα εισόδου X_j για την πρόβλεψη $\hat{y}_t$ και υπολογίζοντας τα βάρη ως εξής:

$$a_{tj} = \frac{\exp \epsilon_{tj}}{\sum_{k=1}^{T_x} \exp \epsilon_{tk}}.$$

Μένει τώρα να περιγράψουμε το κριτήριο με το οποίο υπολογίζουμε τις τιμές ενέργειας ϵ_{tj}

και συνεπώς τα βάρη α_{tj}. Ουσιαστικά, οι τιμές ενέργειας υπολογίζονται σύμφωνα με την συμφωνία που παρατηρείται μεταξύ του διανύσματος κατάστασης του αποκωδικοποιητή τη χρονική στιγμή t (ονομάζεται και ερώτημα - query) και των διανυσμάτων κατάστασης του κωδικοποιητή (ονομάζονται και κλειδιά - keys). Επομένως έχουμε:

$$e_{tj} = \alpha(s_{t-1}, h_j)$$

Όπου εδώ α είναι η συνάρτηση που μετρά την συμφωνία ή ευθυγράμμιση. Η πιο απλή υλοποίηση μιας τέτοιας συνάρτησης είναι με την συνάρτηση συνημιτόνου (στην περίπτωση που τα διανύσματα έχουν ίδιο μήκος). Στο έργο των Bahdanau D. et al. [84] υλοποιείται η συνάρτηση ευθυγράμμισης με ένα πλήρως διασυνδεδεμένο νευρωνικό δίκτυο του οποίου οι παράμετροι εκπαιδεύονται μαζί με τις υπόλοιπες παραμέτρους του μοντέλου.

2.3.3 Μετασχηματιστές

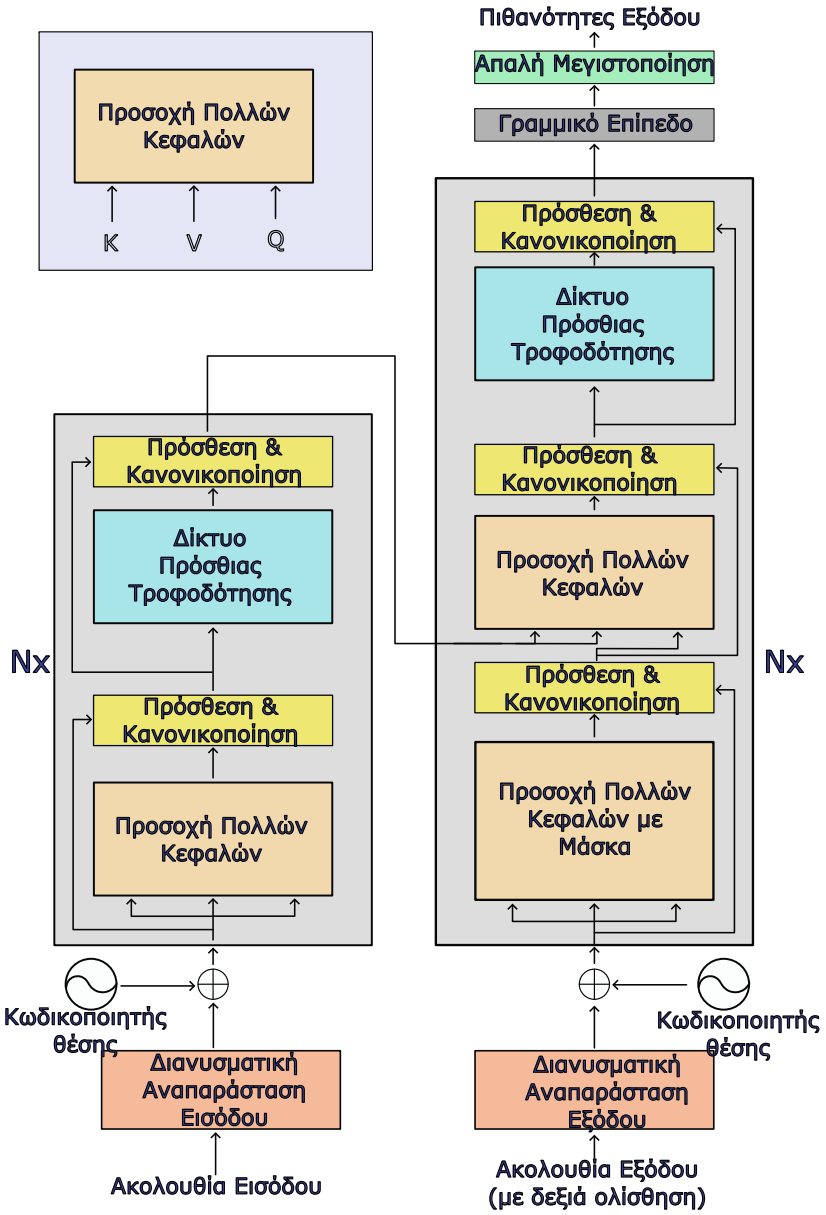
Ενώ τα αμφίδρομα νευρωνικά δίκτυα με μακροπρόθεσμη–βραχυπρόθεσμη μνήμη και μηχανισμό προσοχής μετριάζουν σημαντικά τα δύο πρώτα από τα τρία προβλήματα που αναφέραμε, το τρίτο πρόβλημα παραμένει ανεπίλυτο. Για την ακρίβεια, η φύση των επαναληπτικών νευρωνικών δικτύων είναι τέτοια ώστε να απαιτείται σειριακή επεξεργασία των δειγμάτων μιας ακολουθίας, γεγονός που αυξάνει σημαντικά τους χρόνους επεξεργασίας εμποδίζοντας την κλιμάκωσή τους σε πιο σύνθετες εφαρμογές. Συνεπώς, απαιτείται μια νέα αρχιτεκτονική που θα είναι απαλλαγμένη από την σειριακή επεξεργασία ενώ θα ικανοποιεί συγχρόνως τις τέσσερεις σχεδιαστικές αρχές για την μοντελοποίηση ακολουθιών που προαναφέραμε.

Λύση στο πρόβλημα αυτό έδωσαν οι Vaswani A. et al. με την δημοφιλή δημοσίευση υπό τον τίτλο «Attention Is All You Need». Στο έργο τους, περιγράφουν την αρχιτεκτονική του μετασχηματιστή, όπως αυτή φαίνεται στο σχήμα 2.34. Πρόκειται για μια αρχιτεκτονική νευρωνικού δικτύου υπό μορφή κωδικοποιητή—αποκωδικοποιητή που εμπεριέχει μόνο μηχανισμούς αυτο–προσοχής και πλήρως διασυνδεδεμένων επιπέδων για εφαρμογές μετάφρασης ακολουθιών από λέξεις. Αυτή η νέα αρχιτεκτονική μπορεί και ικανοποιεί πλήρως όλες τις σχεδιαστικές αρχές των συστημάτων μοντελοποίησης ακολουθιών επιτρέποντας την παράλληλη επεξεργασία δειγμάτων εντός της ίδιας ακολουθίας[45].

Αποκωδικοποιητής και Κωδικοποιητής

Το δίκτυο του σχήματος 2.34 μπορεί να χωριστεί σε δύο μέρη: τον κωδικοποιητή στα αριστερά και τον αποκωδικοποιητή στα δεξιά. Ας ξεκινήσουμε από το κατώτερο τμήμα του κωδικοποιητή, δηλαδή το επίπεδο ενσωμάτωσης (embedding layer). Επειδή ως γνωστόν ένα νευρωνικό δίκτυο δεν μπορεί να διαχειριστεί συμβολοσειρές χαρακτήρων παρά μόνο διανύσματα από αριθμούς, το επίπεδο αυτό αναλαμβάνει την αντιστοίχιση κάθε λέξης X_i σε μια συγκεκριμένη αναπαράσταση από $d_{features}$ χαρακτηριστικά. Η αντιστοίχιση δεν πραγματοποιείται τυχαία αλλά με τρόπο ώστε λέξεις σημασιολογικά κοντινές να έχουν μικρή απόσταση στον χώρο αναπαράστασης $\Re^{d_{features}}$. Φυσικά, το επίπεδο ενσωμάτωσης δέχεται ολόκληρη την ακολουθία μήκους T_x και παράγει ένα διάνυσμα

[45] Αν και η νέα αρχιτεκκτονική φαίνεται αρκετά εξειδικευμένη, στην πραγματικότητα είναι γενικότερη από τα νευρωνικά δίκτυα πρόσθιας τροφοδότησης με την έννοια των λιγότερων επαγωγικών προκαταλήψεων (inductive biases).



Σχήμα 2.34: Η αρχιτεκτονική ενός μετασχηματιστή [80]. *Παράχθηκε από το Inkscape.*

αναπαράστασης για κάθε λέξη—δείγμα παράλληλα. Έπειτα, σε κάθε διανυσματική αναπαράσταση λέξης υπερτίθεται το διάνυσμα αναπαράστασης θέσης (position embedding) (μοναδικό για κάθε θέση στην ακολουθία) έτσι ώστε να αναγνωρίζει το μοντέλο την σειρά των δειγμάτων στην ακολουθία[46]. Τέλος, οι προκύπτουσες αναπαραστάσεις μπορούν να συνδυαστούν σε έναν πίνακα $\boldsymbol{X}$ με μέγεθος $T_x \times d_{features}$.

Συνεχίζοντας την περιγραφή του σχήματος 2.34 σύμφωνα με την ροή της πληροφορίας εισόδου δηλαδή από κάτω προς τα πάνω και από τα αριστερά προς τα δεξιά, συναντάμε το μπλόκ του κωδικοποιητή το οποίο δέχεται τρία αντίγραφα του πίνακα $\boldsymbol{X}$. Ο κωδικοποιητής μπορεί να αποτελείται από N επίπεδα (βλ. σχήμα 2.35). Κάθε επίπεδο απαρτίζεται από δύο υπο–επίπεδα: το πρώτο σχηματίζεται από τον μηχανισμό αυτο–προσοχής πολλαπλών κεφαλών (multi–head attention) και

[46]Επειδή τώρα τα δείγματα μιας ακολουθίας δεν επεξεργάζονται σειριακά, απαιτείται κάποια άλλη μέθοδος προκειμένου να μην παραβιάζεται η τρίτη σχεδιαστική αρχή περί διατήρησης πληροφορίας σειράς δειγμάτων.

το δεύτερο από ένα νευρωνικό δίκτυο πρόσθιας τροφοδότησης με δύο πλήρως διασυνδεδεμένα επίπεδα που δρα με τον ίδιο τρόπο σε κάθε διάνυσμα λέξης (position–wise). Γύρω από το καθένα υποεπίπεδο υπάρχει μια υπολειμματική σύνδεση (residual connection), ακολουθούμενη από κανονικοποίηση επιπέδου (layer normalization)[47]. Με άλλα λόγια, αν συμβολίσουμε την έξοδο κάθε υπο–επιπέδου ως $Sublayer(\boldsymbol{X})$ τότε η έξοδος μετά από κάθε υποεπίπεδο μαζί με την υπολειμματική σύνδεση (residual connection) και την κανονικοποίηση επιπέδου (layer normalization) είναι $LayerNorm(\boldsymbol{X} + Sublayer(\boldsymbol{X}))$. Τελικά, ο κωδικοποιητής παράγει (παράλληλα) σαν έξοδο μια ακολουθία $\boldsymbol{Z} = [Z_1, Z_2, \ldots, Z_{T_x}]$ στην οποία κάθε δείγμα περιέχει πλούσια πληροφορία για τα συμφραζόμενά του.

Ο αποκωδικοποιητής σε γενικές γραμμές είναι ένα μοντέλο αυτοπαλινδρόμησης (auto–regressive model) που δέχεται την ακολουθία $\boldsymbol{Z} = [Z_1, Z_2, \ldots, Z_{T_x}]$ από τον κωδικοποιητή και τελικά παράγει σειριακά την ακολουθία $\hat{\boldsymbol{Y}} = [\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_{T_y}]$. Η αυτοπαλινδρόμηση έγκειται στο γεγονός ότι σε κάθε βήμα t, για να εξάγει το μοντέλο το διάνυσμα $\hat{Y}_t$ λαμβάνει υπόψη τις εξόδους που έχει παράξει τις προηγούμενες χρόνικές στιγμές, ολισθημένες κατά 1 θέση δεξιά, δηλαδή τα $[<SOS>, \hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_{t-1}]$[48]. Σε αντίθεση με τα μοντέλα επαναληπτικών νευρωνικών δικτύων με αυτο–παλινδρόμηση όπου η εκπαίδευση καθυστερεί, στους μετασχηματιστές δεν απαιτείται η παραγωγή ολόκληρης της ακολουθίας εξόδου για την εφαρμογή του αλογορίθμου οπισθοδιάδοσης σφάλματος. Δηλαδή, ο αλγόριθμος μάθησης εφαρμόζεται με κάθε δείγμα εξόδου.

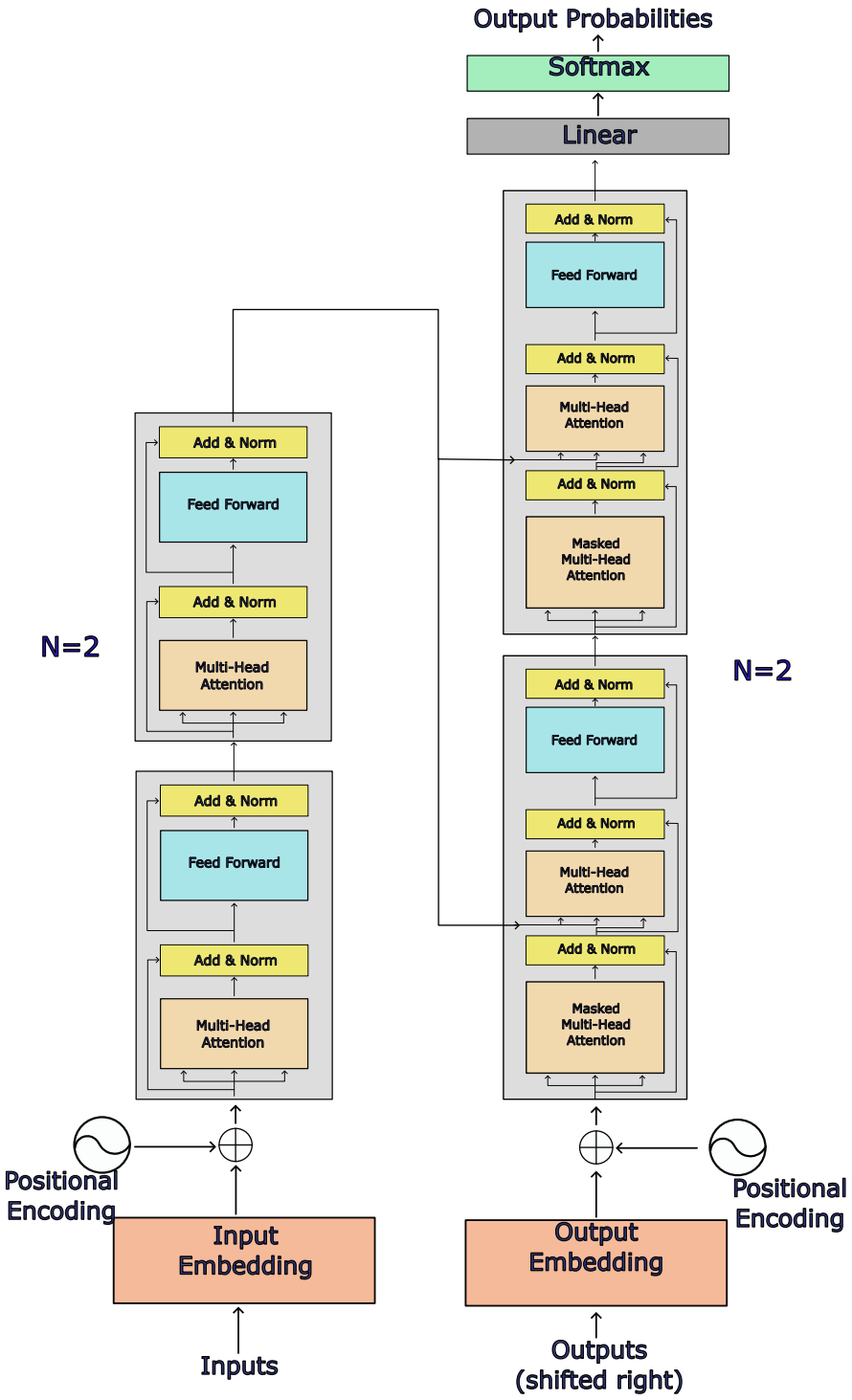
Κοιτώντας το σχήμα 2.34 και αναλύοντάς το από κάτω προς τα πάνω παρατηρούμε ότι και ο αποκωδικοποιητής τροφοδοτείται με λέξεις σε διανυσματική αναπαράσταση μέσω του επιπέδου ενσωμάτωσης (embedding layer) και του κωδικοποιητή θέσης (positional embedding). Βέβαια, σε αντίθεση με τον κωδικοποιητή, η ακολουθία που δέχεται σαν είσοδο ο αποκωδικοποιητής απαρτίζεται από τις προηγούμενες λέξεις—στόχους (δηλαδή τις$[<SOS>, \hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_{t-1}]$).

Συνεχίζοντας την ανάλυση του σχήματος 2.34, ο αποκωδικοποιητής αποτελείται από N πανομοιότηπα επίπεδα (βλ. σχήμα 2.35) και μπορεί να διαιρεθεί σε τρία υποεπίπεδα. Το πρώτο υποεπίπεδο είναι αυτό του μηχανισμού αυτο–προσοχής πολλών κεφαλών με μάσκα (masked multi–head attention). Γύρω από αυτό, όπως και από όλα τα υπο–επίπεδα, υπάρχει μια υπολειμματική σύνδεση που καταλήγει σε ένα επίπεδο κανονικοποίησης. Τα επόμενα δύο υπο–επιπεδα είναι τα ίδια με αυτά που περιγράψαμε στην περίπτωση του κωδικοποιητή. Να σημειώσουμε ότι στο μεσαίο υποεπίπεδο, η είσοδος σχηματίζεται από την ακολουθία διανυσματικών αναπαραστάσεων των προηγηθέντων λέξεων εξόδου που παράγεται από το πρώτο υποεπίπεδο του αποκωδικοποιητή (τον πίνακα αυτό τον ονομάζουμε Ερώτημα - Query και ισχύει $Q \in \Re^{d_{features} \times t}$[49]) και την έξοδο του κωδικοποιητή, αντεγραμμένη δύο φορές (σχηματίζοντας δύο πίνακες, τους Κλειδί - Key και Τιμή - Value). Τέλος, το κάθε επίπεδο αποκωδικοποιητή παράγει μια έξοδο μεγέθους $t \times d_{features}$ η οποία είτε δίνεται στο επόμενο επίπεδο αποκωδικοποιητή είτε, αν δεν υπάρχει κάποιο επόμενο επίπεδο (δηλαδή $N = 1$), μετασχηματίζεται μέσω ένος γραμμικού, πλήρως διασυνδεδεμένου επιπέδου το οποίο έχει σαν έξοδο ένα διάνυσμα $Z_{out} \in \Re^{vocabularysize}$. Το τελευταίο, αφού περάσει από την συνάρτηση απαλής μεγιστοποίησης (softmax) αποτελεί την έξοδο που είναι πρακτικά μια

[47] Βλέπε παράρτημα Α′

[48] Η λεξικογραφική μονάδα «<SOS>» χρησιμοποιείται για να σηματοδοτήσει στον αποκωδικοποιητή την αρχή της φράσης εξόδου.

[49] Θεωρούμε $t = 1$ τη στιγμή παραγωγής της πρώτης εξόδου $\hat{Y}_1$.



Σχήμα 2.35: Η αρχιτεκτονική ενός μετασχηματιστή [80] όταν αυτός αποτελείται από πολλά μπλόκ κωδικοποιητών και αποκωδικοποιητών (στο σχήμα, δύο από το κάθε είδος μπλόκ). *Παράχθηκε από το Inkscape.*

κατανομή διακριτής πιθανότητας πάνω σε όλο το λεξιλόγιο. Η λέξη με την μέγιστη πιθανότητα είναι και η πρόβλεψη $\hat{Y}_t$[50].

Υποεπίπεδο Νευρωνικού Δικτύου Πρόσθιας Τροφοδότησης

Πρόκειται για το τελευταίο υποεπίπεδο τόσο του μπλόκ αποκωδικοποιητή όσο και του μπλόκ κωδικοποιητή και σχηματίζεται από δύο πλήρως διασυνδεδεμένα επίπεδα από τεχνητούς νευρώνες (δεν προσμετράμε το επίπεδο εισόδου). Οι νευρώνες του πρώτου επιπέδου χρησιμοποιούν την συνάρτηση ενεργοποίησης «ReLU» ενώ οι νευρώνες του δεύτερου την ταυτοτική συνάρτηση. Έτσι, έχουμε:

$$FFN(X) = max(0, X \times W^{[1]} + b^{[1]}) \times W^{[2]} + b^{[2]}$$

όπου $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}$ παράμετροι που μαθαίνονται κατά την εκπαίδευση.

Όπως είπαμε, η συνάρτηση του νευρωνικού δικτύου εφαρμόζεται με τα ίδια βάρη σε κάθε δείγμα της ακολουθίας ξεχωριστά. Με άλλα λόγια, αν σαν είσοδο δίνεται ο πίνακας $\boldsymbol{X}$ με μέγεθος $T_x \times d_{features}$ τότε η ίδια συνάρτηση θα εφαρμοστεί T_x φορές[51].

Μηχανισμός Προσοχής Πολλών Κεφαλών

Κατά αναλογία με τον μηχανισμό πρόσοχής ο οποίος δέχεται δύο διαφορετικές ακολουθίες, ο μηχανισμός αυτο–προσοχής συσχετίζει τα δείγματα μιας μεμονωμένης ακολουθίας μεταξύ τους προκειμένου να υπολογίσει μια άλλη αναπαράσταση αυτής της ακολουθίας. Από το σχήμα 2.34 έχουμε παρατηρήσει ότι το υποεπίπεδο που υλοποιεί τον μηχανισμό αυτοπροσοχής δέχεται τρείς εισόδους. Αυτές συμβολίζονται - όπως έχουμε αναφέρει - με τα γράμματα Q (Ερώτημα - Query), K (Κλειδί - Key) και V (Τιμή - Value). Οι ονομασίες αυτές δεν είναι τυχαίες. Διαισθητικά, το ερώτημα χρησιμοποιείται για την αναζήτηση των κατάλληλων κλειδιών στα οποία θα δοθεί προσοχή με παρόμοιο τρόπο με τον οποίο μια μηχανή αναζήτησης διπλωματικών εργασιών χρησιμοποιεί τους όρους αναζήτησης και εξετάζει την ομοιότητά τους με τις λέξεις κλειδιά της κάθε εργασίας. Η τιμή, στο παράδειγμά μας, θα μπορούσε να είναι το περιεχόμενο της κάθε διατριβής.

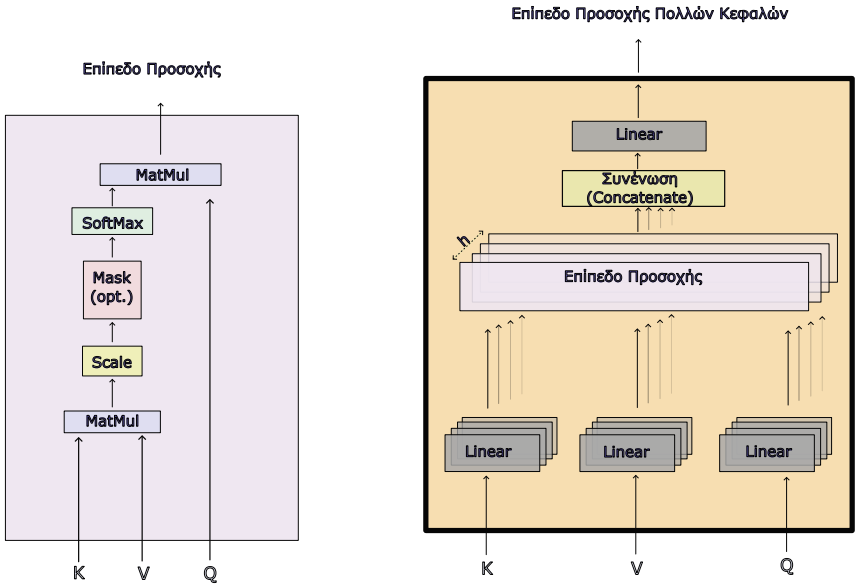
Για την λεπτομερή εξέταση της υλοποίησης του μηχανισμού προσοχής πολλών κεφαλών παρουσιάζεται το σχήμα 2.36. Σε αυτό διακρίνεται η πολυεπίπεδη φύση του υπο–επιπέδου. Πιο αναλυτικά, σε κάθε επίπεδο—κεφαλή $h_i, i \in [1, n_h]$ τα διανύσματα Q, K, V προβάλλονται μέσω γραμμικών επιπέδων (πινάκων παραμέτρων W_i^Q, W_i^K, W_i^V αντίστοιχα) στα Q_i, K_i, V_i. Έπειτα, πραγματοποιείται η «προσοχή με κλιμακωτό εσωτερικό γινόμενο» (Scaled Dot-Product Attention). Το αποτέλεσμα των επιπέδων ενώνεται σε ένα πίνακα ο οποίος τελικά διέρχεται απο ένα γραμμικό επίπεδο (με βάρη που συμβολίζονται ως W^O).

Συνολικά, χρησιμοποιώντας μαθηματική περιγραφή, το υποεπίπεδο προσοχής δέχεται τρείς πίνακες:

$$Q \in \Re^{T \times d_{features}}, K \in \Re^{T \times d_{features}}, V \in \Re^{T \times d_{features}} \tag{2.34}$$

[50] Εκτός αν χρησιμοποιείται ακτινική αναζήτηση beam search.

[51] Το ίδιο δίκτυο θα μπορούσαμε να διατυπώσουμε διαφορετικά ότι αποτελείται από δύο συνελικτικά επίπεδα με μοναδιαίο πυρήνα (point–wise convolutional layers).



Σχήμα 2.36: Το μπλοκ προσοχής πολλών κεφαλών (δεξιά) και το μπλοκ προσοχής με κλιμακωτό εσωτερικό γινόμενο (αριστερά). *Παράχθηκε από το Inkscape.*

$$\text{όπου } T = \begin{cases} T_x & \text{αν } Q|K|V \text{προέρχονται από κωδικοποιητή} \\ t & \text{αν } Q|K|V \text{προέρχονται από αποκωδικοποιητή} \end{cases} \tag{2.35}$$

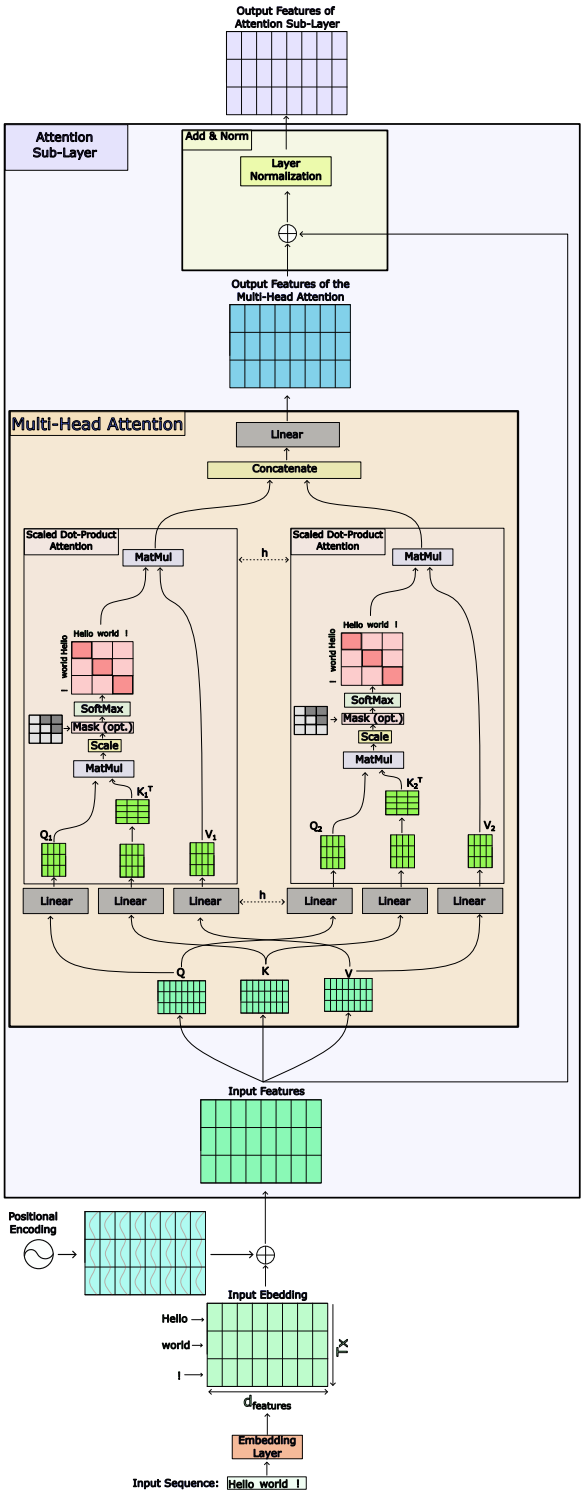
Έπειτα, για τα τρία διανύσματα αυτά, υπολογίζονται τόσες προβολές όσος και ο αριθμός κεφαλών n_h. Δηλαδή για την i κεφαλή έχουμε:

$$Q_i = Q \times W_i^Q, K_i = K \times W_i^K, V_i = V \times W_i^K, \tag{2.36}$$

$$\text{όπου } W_i^Q \in \Re^{d_{features} \times d_k}, W_i^K \in \Re^{d_{features} \times d_k}, W_i^V \in \Re^{d_{features} \times d_v} \tag{2.37}$$

Παρατηρούμε ότι με την προβολή, το μήκος του αριθμού χαρακτηριστικών για κάθε δείγμα της ακολουθίας μετατρέπεται από $d_{features}$ σε d_k ή d_v. Συνήθως, επιλέγεται $d_k = d_v = d_{features}/h$ προκειμένου το υπολογιστικό κόστος να μην πολλαπλασιάζεται καθώς αυξάνεται ο αριθμός των κεφαλών. Εν συνεχεία, για την κάθε κεφαλή πραγματοποιούμε τις ενέργειες του αριστερού σχήματος της εικόνας 2.36. Με μαθηματικούς όρους, είναι:

$$Attention(Q_i, K_i, V_i) = Softmax(\frac{Q_i \times K_i^T}{\sqrt[2]{d_k}} \odot M) \times V_i$$



Σχήμα 2.37: Παράδειγμα της ροής πληροφορίας σε ένα υποεπίπεδο προσοχής. Στο συγκεκριμένο παράδειγμα χρησιμοποιούνται δύο κεφαλές με $d_{features} = 8$ και $d_k = d_v = 4$. Αν και το παράδειγμα εστιάζει περισσότερο σε ενα μπλοκ κωδικοποιητή, οι διαφορές για την περίπτωση του αποκωδικοποιητή είναι ελάχιστες (η ακολουθία εισόδου θα αποτελούνταν από τις λέξεις που είχαν παραχθεί μέχρι την εκάστοτε χρονική στιγμή, ολισθημένες κατά μία θέση. Πριν παραχθεί η πρώτη λέξη, τροφοδοτείται στον αποκωδικοποιητή το σύμβολο «SOS» που σηματοδοτεί την αρχή της ακολουθίας). *Παράχθηκε από το Inkscape.*

όπου για ένα πίνακα

$$\boldsymbol{X} = \underset{(\alpha\times\beta)}{\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_\alpha \end{bmatrix}}$$

ορίζουμε

$$Softmax(\boldsymbol{X}) = \begin{bmatrix} softmax(X_1) \\ softmax(X_2) \\ \vdots \\ softmax(X_\alpha) \end{bmatrix}$$

και όπου $M \in \Re^{T\times T}$.

Ορίζουμε τον προαιρετικό πίνακα μάσκας M με ίδιες διαστάσεις με τον $Q_i \times K_i^T$. Οι δύο πίνακες πολλαπλασιάζονται σημειακά ώστε στην περίπτωση της διαδικασίας αποκωδικοποίησης, κατά την εκπαίδευση όπου είναι από πριν γνωστή η επιθυμητή ακολουθία εξόδου $\boldsymbol{Y}$, το μοντέλο να μην βλέπει τα μελλοντικά δείγματα στόχους. Ο πίνακας M είναι κάτω τριγωνικός με τα μη–μηδενικά στοιχεία ίσα με πλήν άπειρο.

Το αποτέλεσμα της συνάρτησης Softmax είναι ένας πίνακας με διαστάσεις $T \times T$ ο οποίος δείχνει τη συσχέτιση (βαθμός ομοιότητας) μεταξύ των δειγμάτων στις ακολουθίες K_i και Q_i. Προφανώς, κάθε γραμμή είναι κανονικοποιημένη ώστε να αποτελεί μια κατανομή πιθανότητας. Θα τον ονομάζουμε και χάρτη προσοχής (attention map). Δηλαδή:

$$AttentionMap(Q_i, K_i) = softmax(\frac{Q_i \times K_i^T}{\sqrt[2]{d_k}} \odot M).$$

Αφού γίνει και ο πολλαπλασιασμός με τον πίνακα των τιμών, ενώνουμε τα αποτελέσματα κάθε κεφαλής σε ένα διάνυσμα και τα περνάμε από ένα γραμμικό επίπεδο ώστε να λάβουμε ένα πίνακα με δείγματα μήκους $d_{features}$, δηλαδή:

$$MultiHead(Q, K, V) = [head_1 \frown head_2^{frown} \ldots head_{n_h}] \times W^O \tag{2.38}$$

όπου

$$head_i = Attention(Q_i, K_i, V_i) \text{ και } W^O \in \Re^{n_h d_v \times d_{features}}.$$

2.4 Χάρτες Αυτο-οργάνωσης

Η τελευταία έννοια που αναπτύσουμε στο κεφάλαιο αυτό είναι αυτή του «χάρτη αυτο–οργάνωσης» (self–organizing map - SOM) [90, 91]. Αφορά την τεχνική μη–επιβλεπόμενης μάθησης η οποία παράγει μια χαμηλής διαστατικότητας απεικόνιση (συνήθως δισδιάστατη) ενός συνόλου δεδομένων υψηλής διαστατικότητας, διατηρώντας την τοπολογική δομή τους. Για να εξηγήσουμε περαιτέρω την τεχνική αυτή που θα χρησιμοποιήσουμε στην συνέχεια, θα περιγράψουμε πρώτα τι είναι η ανταγωνιστική μάθηση. Έπειτα, θα αναφερθούμε στην αρχιτεκτονική ενός χάρτη που α-

κολουθεί το μοντέλο Kohonen. Στη συνέχεια, θα κάνουμε μια νύξη στην τοπογραφική οργάνωση του εγκεφαλικού φλοιού και πως αυτό το χαρακτηριστικό ενέμπνευσε την παρούσα τεχνολογία. Τέλος, θα παρουσιάσουμε τον αλγόριθμο σχηματισμού ενός χάρτη αυτο–οργάνωσης.

Ανταγωνιστική Μάθηση

Οι χάρτες αυτο–οργάνωσης αποτελούνται από μια ειδική κατηγορία τεχνητών νευρωνικών δικτύων που βασίζονται σε ένα είδος μη–επιβλεπόμενης μάθησης, την ανταγωνιστική μάθηση (competitive learning) [31]. Πιο αναλυτικά, στο τεχνητό νευρωνικό δίκτυο, οι νευρώνες ανταγωνίζονται μεταξύ τους για το δικαίωμα ενεργοποίησης (excitation), με αποτέλεσμα μόνο ένας νευρώνας εξόδου (ή ένας νευρώνας ανά ομάδα) να είναι ενεργός κάθε στιγμή. Το κριτήριο ενεργοποίησης ενός νευρώνα είναι ο βαθμός με τον οποίο ο νευρώνας μπορεί να εξηγήσει το εκάστοτε διάνυσμα εισόδου x_i (τραβηγμένο τυχαία από ένα σύνολο δεδομένων S_n). Ο νευρώνας που ενεργοποιείται στην είσοδο $x_i, i \in [1, n]$ αποκαλείται νευρώνας νικητής και απολαμβάνει την μεγαλύτερη τροποποίηση ώστε να αναπαριστά πιστότερα την είσοδο x_i. Στην ειδική περίπτωση που μόνο ο νευρώνας νικητής προσαρμόζεται στο διάνυσμα εισόδου τότε λέμε ότι αυτός απολαμβάνει το καθεστώς του «ο νικητής τα παίρνει όλα» (winner takes it all)[52].

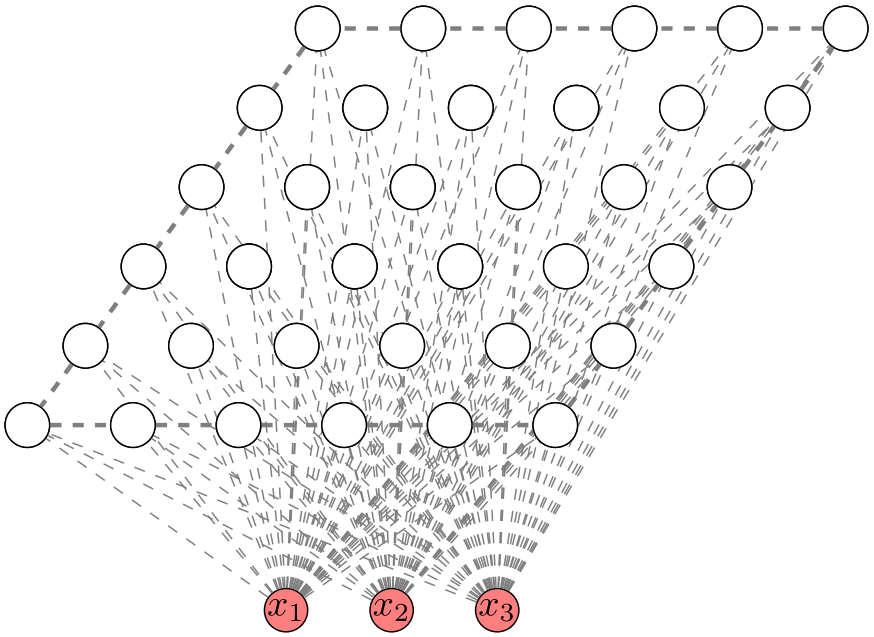
2.4.1 Αρχιτεκτονική Χάρτη Αυτο–οργάνωσης

Στο σχήμα 2.38 παρουσιάζεται η αρχιτεκτονική ενός χάρτη αυτο–οργάνωσης που ακολουθεί το μοντέλο του Kohonen[53]. Παρατηρούμε ότι οι τεχνητοί νευρώνες οργανώνονται σε ένα δισδιάστατο πλέγμα (lattice) από κόμβους, μεγέθους $d_x \times d_y$. Ο κάθε νευρώνας $w_j, j \in [1, d_x \times d_y]$, αποτελείται από τόσα βάρη όση και η διάσταση των διανυσμάτων εισόδου (m), δηλαδή $w_j = [w_{j1}, w_{j2}, \ldots, w_{jm}]$. Θα συμβολίζουμε το σύνολο όλων των νευρώνων του πλέγματος με το σύμβολο $\mathcal{A}$. Χάρη στην ανταγωνιστική μάθηση που εξηγήσαμε παραπάνω, οι νευρώνες συντονίζονται (tuned) επιλεκτικά σε διάφορα πρότυπα εισόδου τροποποιώντας κατάλληλα τα βάρη τους ώστε να τα αναπαριστούν καλύτερα. Ο συντονισμός γίνεται με τέτοιο τρόπο ώστε οι νευρώνες σταδιακά να διατάσσονται πάνω στο πλέγμα σε θέσεις ανάλογες με το πρότυπο που αναπαριστούν, σχηματίζοντας έτσι έναν λογικό χάρτη από συστάδες (clusters). Με άλλα λόγια, «ένας αυτο–οργανούμενος χάρτης χαρακτηρίζεται από το σχηματισμό ενός τοπογραφικού χάρτη αποτελούμενου από τα πρότυπα εισόδου, στον οποίο οι χωρικές θέσεις (οι συντεταγμένες) των νευρώνων στο πλέγμα είναι ενδεικτικές των εσωτερικών στατιστικών χαρακτηριστικών που περιέχονται στα πρότυπα εισόδου» [31].

Το μοντέλο Kohonen που εξετάζουμε ανήκει στην κατηγορία των αλγορίθμων διανυσματικής κωδικοποίησης. Αυτό διότι μέσω του χάρτη Kohonen παρέχεται μια τοπολογική αντιστοίχιση που τοποθετεί με βέλτιστο τρόπο ένα σταθερό αριθμό διανυσμάτων $w_j, j \in [1, d_x \times d_y]$ σε έναν υψηλότερης διαστατικότητας χώρο δεδομένων εισόδου $S_n, n >> d_x \times d_y$ πραγματοποιώντας κατά αυτόν τον τρόπο συμπίεση δεδομένων με απώλειες [31]. Με απλά λόγια, για κάθε ομάδα A από διανύσματα εισόδου που παρουσιαζει τα ίδια μοτίβα ή πρότυπα $(A \subseteq S_n)$ αντιστοιχίζεται ένα διάνυσμα w_j που τα περιγράφει (ονομάζεται και κεντροειδές - centroid). Έτσι, αντί να

[52] Στο μοντέλο που θα εξετάσουμε, δεν ισχύει αυτό το καθεστώς αφού ο νικητής νευρώνας δεν είναι ο μόνος που τροποποιείται ανάλογα με την είσοδο.

[53] Ένα άλλο μοντέλο είναι το λιγότερο δημοφιλές μοντέλο του Willshaw - von der Malsburg το οποίο όμως ξεφεύγει από τα πλαίσια της παρούσας διπλωματικής εργασίας.



Σχήμα 2.38: Η αρχιτεκτονική ενός δισδιάστατου χάρτη αυτο–οργάνωσης που ακολουθεί το μοντέλο του Kohonen. *Παράχθηκε από το Inkscape τροποποιώντας αυτήν την εικόνα.*

είναι αποθηκευμένη μια ομάδα από παραπλήσια αλλά διαφορετικά διανύσματα, μπορεί μόνο να αποθηκεύεται το διάνυσμα w_j που τα περιγράφει.

2.4.2 Ο Ανθρώπινος Εγκέφαλος ως Πηγή Έμπνευσης

Έχει παρατηρηθεί ότι ο εγκέφαλος και κυρίως ο εγκεφαλικός φλοιός είναι, σε πολλές περιοχές, οργανωμένος με τρόπο ώστε διαφορετικές αισθητηριακές είσοδοι να αναπαρίσταται από τοπολογικά διατεταγμένους υπολογιστικούς χάρτες [31]. Οι πρώτες ενδείξεις που οδήγησαν σε αυτό το συμπέρασμα ήδη από τον 18ο αιώνα [92] οφείλονται στην παρατήρηση ότι τοπικά περιορισμένες εγκεφαλικές κακώσεις προκαλούν συγκεκριμένες παθήσεις [91, 93]. Αργότερα, η ιδέα ότι διαφορετικές περιοχές του εγκεφάλου φαίνεται να αφορούν συγκεκριμένες εργασίες αποδείχθηκε με τις σύγχρονες απεικονιστικές μεθόδους [91]. Για παράδειγμα, έχει αποδειχθεί ότι οι απτικές, οι οπτικές και οι ακουστικές αισθητηριακές είσοδοι χαρτογραφούνται σε διαφορετικές περιοχές του εγκεφαλικού φλοιού με τοπολογικά διατεταγμένο τρόπο [31].

Η διαπίστωση της τοπογραφικής οργάνωσης των λειτουργιών του εγκεφάλου πυροδότησε, όπως θα ήταν αναμενόμενο σε μια εποχή όπου η νευροεπιστήμη ήταν σε στενή επαφή με την τεχνητή νοημοσύνη, μια σειρά από έρευνες για κατασκευή τεχνητών υπολογιστικών χαρτών. Οι απόπειρες επιδίωκαν να μιμηθούν τους βιολογικούς μηχανισμούς της αυτο–οργάνωσης διατηρώντας τις παρακάτω βασικές αρχές:

- Οι νευρώνες πάνω στον υπολογιστικό χάρτη λειτουργούν παράλληλα και επεξεργάζονται ο καθένας πληροφορία εισόδου η οποία προέρχεται από διαφορετικές πηγές (δηλαδή, τα χαρακτηριστικά των διανυσμάτων εισόδου ακολουθούν διαφορετικά πρότυπα).
- Σε κάθε χρονική στιγμή, κάθε εισερχόμενη πληροφορία αντιστοιχίζεται στο κατάλληλο νοητικό πλαίσιο (χωρική θέση πάνω στον χάρτη). Με άλλα λόγια, η χωρική θέση ενός

νευρώνα εξόδου σε έναν τοπογραφικό χάρτη αντιστοιχεί σε ένα συγκεκριμένο πεδίο (ή πρότυπο) των δεδομένων εισόδου [91].

- Οι νευρώνες που ασχολούνται με στενά σχετιζόμενα πρότυπα εισόδου τοποθετούνται στο πλέγμα κοντά μεταξύ τους (γεγονός που διευκολύνει τη συνεργασία).

- Ο χάρτης πραγματοποιεί μείωση της διαστατικότητας από τον χώρο των παραμέτρων στον (δισδιάστατο) χώρο αναπαράστασης. [31, 94, 95]

Έχοντας περιγράψει τις βασικές αρχές με τις οποίες οικοδομήθηκαν τα μοντέλα των τεχνητών χαρτών αυτο–οργάνωσης όπως αυτό του Kohonen, είμαστε σε θέση να περιγράψουμε αναλυτικά τον αλγόριθμο σχηματισμού τους.

2.4.3 Αλγόριθμος Σχηματισμού Χάρτη Αυτο–οργάνωσης

Κάθε αλγόριθμος σχηματισμού χάρτη αυτο–οργάνωσης, για να σέβεται τις τέσσερεις αρχές που παρουσιάστηκαν παραπάνω, πρέπει να πραγματώνει τις εξής τρείς βασικές διαδικασίες:

1. Ανταγωνισμού. Σε αυτή τη διαδικασία οι νευρώνες του δικτύου θα πρέπει να ανταγωνίζονται μεταξύ τους για το ποιός είναι ο ποιο κατάλληλος να εξηγήσει το εκάστοτε δείγμα εισόδου. Ο νικητής του ανταγωνισμού για το συγκεκριμένο δείγμα (έστω x_i) είναι αυτός με την μεγαλύτερη επίδοση, όπως αυτή υπολογίζεται από μια συνάρτηση διάκρισης.

2. Συνεργασίας. Σε αυτή, ο νικητής νευρώνας καθορίζει το εύρος της χωρικής γειτονιάς οι νευρώνες της οποίας θα διεγερθούν ταυτόχρονα με τον νευρώνα νικητή, ενισχύοντας έτσι μια σχέση συνεργασίας[54].

3. Προσαρμογής Συναπτικών Βαρών. Σε αυτή τη διαδικασία λαμβάνει χώρα η μεταβολή των βαρών w των διεγερμένων νευρώνων σε κατεύθυνση ώστε να αυξάνεται η επίδοσή τους σε σχέση με το πρότυπο που ακολουθεί το δείγμα εισόδου x_i (όπως μετράται από την συνάρτηση διάκρισης). [31]

Στον επαναληπτικό αλγόριθμο για το σχηματισμό χάρτη αυτο–οργάνωσης του Kohonen, τα βήματα με τα οποία υλοποιούνται οι ανωτέρω διαδικασίες είναι πέντε:

1. Αρχικοποίηση. Στο βήμα αυτό αρχικοποιούνται τα βάρη όλων των νευρώνων του πλέγματος με τυχαίες, μικρές τιμές[55]. Θα μπορούσε για παράδειγμα να είναι:

$$w_0(0) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \boldsymbol{\mu} = \mathcal{O}_{m\times 1}, \boldsymbol{\Sigma} = \mathcal{I}_m, \forall j \in [1, d_x \times d_y]. \tag{2.39}$$

2. Δειγματοληψία. Σε αυτό το στάδιο λαμβάνεται τυχαία ένα δείγμα εισόδου $x_i \in \Re^m$ από το σύνολο δεδομένων S_n. Το δείγμα αντιπροσωπεύει το πρότυπο ενεργοποίησης (ερέθισμα) που εφαρμόζεται στο πλέγμα. Εναλλακτικά, τα δείγματα x_i μπορούν να λαμβάνονται σειριακά από το σύνολο δεδομένων εισόδου.

[54] Η συνεργασία η οποία διαμορφώνεται από την ταυτόχρονη πυροδότηση γειτονικών νευρώνων βασίζεται στην Χεμπιανή μάθηση, όπως την αναφέραμε στην ενότητα 1.2

[55] Εναλλακτικά, αρχικοποιούνται λαμβάνοντας τυχαία δείγματα από το σύνολο δεδομένων εισόδου S_n.

3. Ταίριασμα Ομοιότητας. Εδώ λαμβάνει χώρα ο ανταγωνισμός μεταξύ των νευρώνων. Σε αυτό το βήμα, δοσμένου του προτύπου ενεργοποίησης x_i επιλέγεται ο νευρώνας νικητής $v(x_i), v(x_i) \in \mathcal{A}$. Η συνάρτηση διάκρισης που χρησιμοποιείται είναι αυτή της ελάχιστης ευκλείδειας απόστασης[56]. Έτσι, με μαθηματικούς όρους, έχουμε:

$$v(x_i) = \min_j \|x_i - w_j\|, j \in [1, d_x \times d_y]. \tag{2.40}$$

4. Ενημέρωση. Σε αυτό το σημείο πραγματοποιούνται οι σημαντικές διαδικασίες της συνεργασίας και της προσαρμογής συναπτικών βαρών. Αναλυτικότερα, τα βάρη των νευρώνων του πλέγματος ανανεώνονται από την παλιά τιμή τους που είχαν την χρονική στιγμή t στην νέα τους τιμή μέσω του τύπου:

$$w_j(t+1) = w_j(t) + \eta(t) h_{j,v(x_i)}(t)(x_i - w_j(t)). \tag{2.41}$$

Όπου $\eta(t)$ είναι μια συνάρτηση–υπερπαράμετρος που καθορίζει τον ρυθμό μάθησης ανάλογα με τον αριθμό των επαναλήψεων που έχουν παρέλθει (t). Μια κατάλληλη τέτοια συνάρτηση είναι η

$$\eta(t) = \eta_0 \exp -\frac{t}{\tau_2} \tag{2.42}$$

όπου τ_2 μια σταθερά (υπερπαράμετρος) χρόνου και η_0 μια τιμή ρυθμού μάθησης κατά την εκκίνηση (όταν $t = 0$).

$h_{j,v(x_i)}(t)$ είναι η συνάρτηση–υπερπαράμετρος που ορίζει μια γειτονιά ενεργοποιημένων νευρώνων γύρω από τον νευρώνα νικητή. Είναι σημαντικό να τονίσουμε ότι η γειτονιά εξαρτάται από την πλευρική απόσταση (lateral distance) μεταξύ του νευρώνα νικητή και των υπόλοιπων νευρώνων στο πλέγμα, όπως αυτή υπολογίζεται στον (συνήθως δισδιάστατο) χώρο εξόδου. Δηλαδή, για τον υπολογισμό της γειτνίασης μεταξύ δύο νευρώνων, δεν λαμβάνονται υπόψη τα βάρη τους w παρά μόνο η σχετική τους απόσταση $d_{j,v}$ πάνω στο πλέγμα. Μια κατάλληλη επιλογή της συνάρτησης γειτονιάς είναι:

$$h_{j,v(x_i)}(t) = exp - \frac{d_{j,v}^2}{2\sigma^2(t)} \tag{2.43}$$

όπου το $\sigma(t)$ καθορίζει το εύρος της τοπολογικής γειτονιάς και είναι:

$$\sigma(t) = \sigma_0 exp - \frac{t}{\tau_1} \tag{2.44}$$

όπου τ_1 μια σταθερά χρόνου και σ_0 η τιμή αρχικού εύρους.

5. Συνέχιση. Το τελευταίο αυτό βήμα εξετάζει αν οι αλλαγές στο χάρτη χαρακτηριστικών είναι ευδιάκριτες. Αν κάτι τέτοιο είναι αληθές, τότε επαναφέρει την ροή προγράμματος στο βήμα 2. Αλλιώς, ο αλγόριθμος έχει συγκλίνει και συνεπώς τερματίζει.

[56] Εναλλακτικά, θα μπορούσε να χρησιμοποιηθεί το συνημίτονο ομοιότητας ως κριτήριο επιλογής του νικητή. Άλλωστε, το κριτήριο βέλτιστης ταύτισης βάση της μεγιστοποίησης του εσωτερικού γινομένου μεταξύ των διανυσμάτων $w_j, \forall j \in [1, d_x \times d_y]$ και x_i (cosine similarity) είναι μαθηματικώς ισοδύναμο με το κριτήριο της ελαχιστοποίησης της Ευκλείδειας απόστασης με την προϋπόθεση ότι τα διανύσματα βαρών των νευρώνων έχουν μοναδιαίο μήκος [31].

Σαν τελικά σχόλια σχετικά με τον αλγόριθμο σχηματισμού αυτο–οργανούμενου χάρτη να αναφέρουμε ότι βοηθάει αν χωρίσουμε την εκπαίδευση σε δύο φάσεις: την φάση αυτο–οργάνωσης και την φάση σύγκλισης. Στην πρώτη, λαμβάνει χώρα η τοπολογική διάταξη των διανυσμάτων των βαρών και συνήθως τόσο το εύρος της γειτονιάς όσο και ο ρυθμός μάθησης έχουν μεγάλες τιμές (οι οποίες βαθμιαία μειώνονται). Στην δεύτερη φάση, πραγματοποιούνται λεπτές προσαρμογές στον χάρτη έτσι ώστε να παρέχει μια επακριβή στατιστική ποσοτικοποίηση του χώρου εισόδου [31]. Για τον σκοπό αυτό, εξυπηρετεί η ανάθεση των υπερπαραμέτρων που καθορίζουν το μέγεθος της γειτονιάς και τον ρυθμό μάθησης σε μικρές τιμές.

Κεφάλαιο 3

Βιβλιογραφική Επισκόπηση

Πριν την έναρξη της εκπόνησης του πρακτικού τμήματος της παρούσας διπλωματικής πραγματοποιήθηκε βιβλιογραφική επισκόπηση προκειμένου να αναζητηθούν εργασίες σχετικές με το θέμα των νευρωνικών δικτύων με κάψουλες. Στο κεφάλαιο αυτό, θα γίνει αναφορά στις σημαντικότερες από αυτές οι οποίες λήφθηκαν υπόψιν και ενέπνευσαν τις μεθόδους που θα αναλύσουμε στο επόμενο κεφάλαιο.

Αρχικά, θα παρουσιάσουμε τις τρεις βασικές δημοσιεύσεις των Hinton G. et al. [47,75,76] που θεμελίωσαν τη θεωρία πίσω από τα νευρωνικά δίκτυα με κάψουλες σε ένα πλαίσιο επιβλεπομένης μάθησης. Έπειτα, θα αναφερθούμε στις ποικίλες παραλλαγές αυτών, όπως προκύπτουν από την τροποποίηση της αρχιτεκτονικής ή του αλγορίθμου δρομολόγησης. Στη συνέχεια, θα γίνει λόγος για τα νευρωνικά δίκτυα με κάψουλες σε περιβάλλον μη–επιβλεπομένης μάθησης. Τέλος, θα αναλυθούν συνοπτικά εργασίες οι οποίες δεν εντάσσονται άμεσα στις ανωτέρω κατηγορίες αλλά βοηθούν είτε άμεσα είτε έμμεσα στην αποτελεσματική επίλυση του γενικότερου προβλήματος της γενίκευσης σε νέες οπτικές γωνίες.

Συγκεντρωτικά, μέσα από την παρακάτω βιβλιογραφική μελέτη αντλούμε τα εξής χρήσιμα συμπεράσματα:

- Οι βασικές υλοποιήσεις των Hinton G. et al. εμφανίζουν ορισμένες ιδιότητες (π.χ. ευρωστία σε μετασχηματισμούς περιστροφής και σε επικαλυπτόμενα αντικείμενα) οι οποίες και ταυτοποιούν την αρχιτεκτονική των νευρωνικών δικτύων με κάψουλες. Συνεπώς, κάθε νέα αρχιτεκτονική που ισχυρίζεται ότι αποτελείται από κάψουλες οφείλει να υποβάλλεται σε δοκιμές που αποδεικνύουν την παρουσία των χαρακτηριστικών ιδιοτήτων (σύνολα δεδομένων affNIST ή SmallNORB και MultiMNIST).

- Πολλές από τις υλοποιήσεις των νευρωνικών δικτύων με κάψουλες (συμπεριλαμβανομένων των [47, 75, 76]) βασίζονται στις υποθέσεις που παρουσιάστηκαν στην ενότητα 2.2.4. Παρόλα αυτά, στη βιβλιογραφία δεν εντοπίστηκαν εκτενή πειράματα που ελέγχουν την εγκυρότητα των υποθέσεων.

- Αρκετές υλοποιήσεις επιδιώκουν να βελτιώσουν τον αλγόριθμο δρομολόγησης με συμφωνία με σκοπό να είναι ταχύτερος ή να οδηγεί σε καλύτερα πειραματικά αποτελέσματα. Αρκετές από αυτές τις προτάσεις δε διενεργούν εκτενή πειράματα ώστε να πιστοποιηθεί ότι η νέα, προτεινόμενη αρχιτεκτονική σέβεται τις θεμελιακές αρχές των νευρωνικών δικτύων με κάψουλες.

- Η εν λόγω τεχνολογία δεν κλιμακώνει εύκολα λόγο μεγάλου υπολογιστικού κόστους. Κατά συνέπεια, οι εφαρμογές της περιορίζονται σε απλά σύνολα δεδομένων.

- Η προσθήκη επιπλέον συνελικτικών επιπέδων στα πρώτα επίπεδα του δικτύου βελτιώνει την επίδοσή τους.

3.1 Θεμελίωση Θεωρίας Νευρωνικών Δικτύων με Κάψουλες

Όπως έχουμε αναφέρει, η ιδέα των νευρωνικών δικτύων με κάψουλες δεν είναι καινούρια αφού παρουσιάστηκε για πρώτη φορά από τους Hinton G. et al. το 2011. Παρόλα αυτά, σχετικά πρόσφατα, μετά από διαδοχικές δημοσιεύσεις, ωρίμασε και πέτυχε αξιοσημείωτα αποτελέσματα σε σύνολα δεδομένων όπως το MultiMNIST [76]. Στην ενότητα αυτή θα κάνουμε λόγο για τα πρώτα τρία βασικά έργα πάνω στην εν λόγω αρχιτεκτονική τεχνητών νευρωνικών δικτύων. Πιο αναλυτικά, θα ξεκινήσουμε από τη δημοσίευση στην οποία πρωτοπαρουσιάστικε η ιδέα και θα καταλήξουμε στην πιο σύνθετη έκδοση των νευρωνικών δικτύων με κάψουλες για επιβλεπόμενη μάθηση που με τις επιδόσεις της στο σύνολο δεδομένων smallNORB [96] κέντρισε το ενδιαφέρον των ερευνητών.

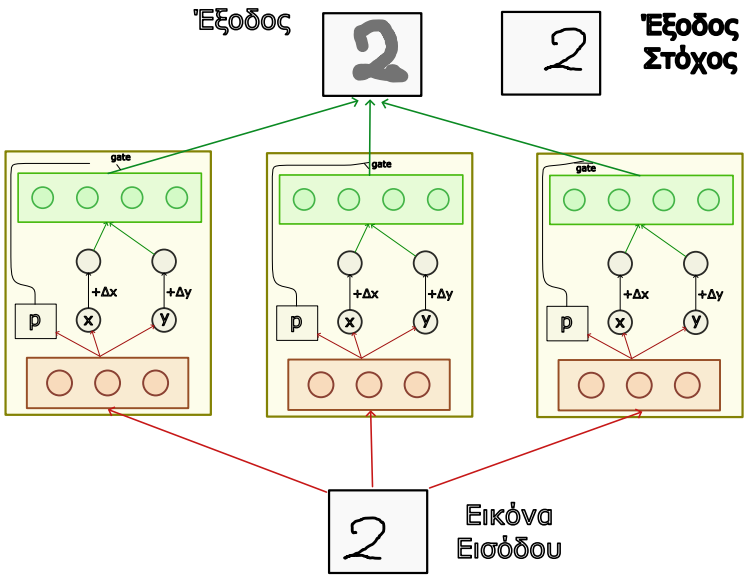
Transforming Autoencoders

Στο έργο των Hinton G. et al.[1] [75] παρουσιάζεται για πρώτη φορά η ιδέα των νευρωνικών δικτύων με κάψουλες. Η ιδέα απορρέει από την παρατήρηση ότι οι παρούσες μέθοδοι αναγνώρισης αντικειμένων σε εικόνες είναι ανεπαρκείς (για τους λόγους που αναφέραμε στην ενότητα 2.2). Έτσι, προκειμένου να γίνεται αποδοτικότερη αναγνώριση αντικειμένων (σε νέες οπτικές γωνίες), προτείνεται η αρχιτεκτονική του «αυτο–κωδικοποιητή μετατροπέα» (transforming auto–encoder). Η αρχιτεκτονική αυτή αποτελείται από ένα επίπεδο από «κάψουλες», όπως φαίνεται στο σχήμα 3.1[2].

Κάθε κάψουλα αποτελείται από τις «μονάδες αναγνώρισης» (recognition units) οι οποίες παράγουν τις παραμέτρους στιγμιοτύπου καθώς και μια τιμή που συμβολίζει την πιθανότητα η οντότητα που αναγνωρίζει η κάψουλα να είναι παρούσα στο οπτικό της πεδίο (στο τμήμα της εικόνας εισόδου με το οποίο συνδέονται οι μονάδες αναγνώρισής της —στην περίπτωσή μας, σε ολόκληρη την εικόνα). Οι μονάδες παραγωγής είναι υπεύθυνες και για τον μετασχηματισμό από τον χώρο εικονοστοιχείων της εικόνας εισόδου σε έναν χώρο όπου οι μετασχηματισμοί οπτικής γωνίας (μετατόπισή, περιστροφή κ.α.) είναι γραμμικοί. Στη συνέχεια, κάθε κάψουλα τροφοδοτείται με τους καθολικούς (global) μετασχηματισμούς που συνδέουν την εικόνα εισόδου με την εικόνα εξόδου οι οποίοι εφαρμόζονται στις υπολογισμένες παραμέτρους. Έτσι, οι παράμετροι στιγμιοτύπου της κάθε κάψουλας πλέον εκφράζουν τις παραμέτρους στιγμιοτύπου του αντικειμένου εξόδου. Τέλος, η εικόνα ανακατασκευάζεται από τις μονάδες παραγωγής generation units τις οποίες κάθε κάψουλα διαθέτει. Αυτές ουσιαστικά διαβάζουν τις (μετασχηματισμένες) παραμέτρους στιγμιοτύπου και συνεισφέρουν στην ανακατασκευή της εικόνας εξόδου. Θεωρητικά, κάθε κάψουλα αναγνωρίζει ένα συγκεκριμένο τμήμα του αντικειμένου της εικόνας και όλες μαζί οι κάψουλες, συνθέτουν τα τμήματα που αναπαριστούν σιωπηρά (implicitly). Φυσικά, αν η τιμή

[1] Σε ελεύθερη μετάφραση: «Αυτο–κωδικοποιητές Μετασχηματισμού».

[2] Στην πρώιμη μορφή τους, οι κάψουλες διέφεραν από τη γενική μορφή που περιγράψαμε στην προηγούμενη ενότητα.



Σχήμα 3.1: Αρχιτεκτονική αυτο–κωδικοποιητών μετασχηματισμού *Παράχθηκε από το Inkscape.*

πιθανότητας για μια κάψουλα είναι κοντά στο μηδέν, η συνεισφορά της θα είναι αμελητέα.

Ουσιαστικά, η αρχιτεκτονική του αυτο–κωδικοποιητή που παρουσιάζεται πραγματοποιεί έναν μετασχηματισμό από τον χώρο των εικονοστοιχείων σε έναν χώρο αναπαράστασης όπου οι γεωμετρικοί μετασχηματισμοί περιγράφονται με γραμμικές σχέσεις. Στον χώρο αυτό εφαρμόζεται ένας γραμμικός μετασχηματισμός στις παραμέτρους της κάθε κάψουλας και έπειτα, οι παράμετροι αποκωδικοποιούνται πίσω στον χώρο των εικονοστοιχείων όπου και λαμβάνεται η μετασχηματισμένη εικόνα.

Στη δημοσίευση γίνονται πειράματα κυρίως στο σύνολο δεδομένων MNIST [41] για μικρές μετατοπίσεις των ψηφίων κατά τον x και y άξονα. Από αυτά τα πειράματα φαίνεται ότι οι παράμετροι σωστά εντοπίζουν τη θέση των αντικειμένων αλλά το οπτικό πεδίο της κάθε κάψουλας, μετά την εκπαίδευση, δεν είναι τοπικά προσδιορισμένο. Με άλλα λόγια, χρησιμοποιείται το σύνολο της εικόνας από τις μονάδες αναγνώρισης της κάθε κάψουλας για την εξαγωγή των παραμέτρων της. Επιπρόσθετα πειράματα έγιναν χρησιμοποιώντας το σύνολο δεδομένων smallNORB [96] προκειμένου να διερευνηθεί η επίδοση της αρχιτεκτονικής σε σύνθετες μεταβολές της οπτικής γωνίας (3-D Orientation) που αναπαριστώνται με πίνακες 3×3. Για αυτό το σύνολο αυξήθηκε ο αριθμός των καψουλών του δικτύου και αυτές είχαν πεδίο υποδοχής που δεν κάλυπτε όλη την εικόνα. Όπως φαίνεται και στη δημοσίευση [75], οι παραχθείσες εικόνες φαίνονται θολές.

Αν και το έργο που περιγράφουμε έχει ιδιαίτερη αξία αφού θεμελίωσε τις αρχές των νευρωνικών δικτύων με κάψουλες και διατύπωσε ορισμένα προβλήματά τους (π.χ. crowding), δεν μπορούσε να έχει πρακτική εφαρμογή λόγω της επίδοσής του στα σύνολα δεδομένων που δοκιμάστηκε. Επιπλέον, το γεγονός ότι για την εκπαίδευσή του, εκτός από τις εικόνες εισόδου και εξόδου έπρεπε να παρέχεται και η σχέση μεταξύ αυτών ήταν ένας ακόμη ανασταλτικός παράγοντας. Επιπρόσθετα, δεν παρείχε κάποιον ρητό τρόπο για την ανάθεση μερών του αντικειμένου σε αυτό. Όλα αυτά οδήγησαν σε απόπειρες βελτίωσης της αρχιτεκτονικής από το επόμενο έργο που θα

παρουσιάσουμε.

Dynamic Routing Between Capsules

Το έργο των Sabour S. et al.[3] [76] εξελίσσει την προηγούμενη μελέτη των νευρωνικών δικτύων με κάψουλες προτείνοντας έναν αλγόριθμο δρομολόγησης μέσω συμφωνίας. Με αυτόν, καθίσταται δυνατή η σύνθεση αντικειμένων από τα επιμέρους τμήματά του. Επιπλέον, αναθεωρεί τη δομή της κάψουλας η οποία πλέον είναι απόλυτα σύμφωνη με τον ορισμό που δώσαμε στην ενότητα 2.2. Δηλαδή, οι κάψουλες πλέον δεν αποτελούνται από δύο διαφορετικές ομάδες από τεχνητούς νευρώνες αλλά είναι ομάδες νευρώνων και η κάθε μια αναπαριστά ιδιότητες της συγκεκριμένης οντότητας που αναγνωρίζει. Οι ιδιότητες της αναγνωρισμένης οντότητας αναπαριστώνται με ένα διάνυσμα ενώ η βεβαιότητα αναγνώρισής της στην εικόνα εισόδου (η τιμή πιθανότητας) κωδικοποιείται στο μήκος του διανύσματος αυτού. Οι ανωτέρω βελτιώσεις, σε συνδυασμό με μια καινούρια αρχιτεκτονική είχαν σαν αποτέλεσμα βελτιωμένες (για την εποχή) επιδόσεις στο MNIST [41] και στο MultiMNIST [76].

Αναλυτικότερα για τη δομή του δικτύου από κάψουλες, αυτή αποτελείται από τρία επίπεδα τεχνητών νευρώνων. Το πρώτο είναι ένα κλασσικό συνελικτικό επίπεδο, όπως περιγράφηκε στην ενότητα 2.1.3. Το δεύτερο επίπεδο είναι και αυτό συνελικτικό και μαζί με το πρώτο αναλαμβάνουν τον ρόλο του μετασχηματισμού του χώρου των εικονοστοιχείων (χώρος εισόδου) σε έναν χώρο όπου οι νευρικές αποκρίσεις μεταβάλλονται γραμμικά καθώς αλλάζει η γωνία θέασης της εικόνας εισόδου. Στη συνέχεια, οι χάρτες χαρακτηριστικών (οι νευρικές αποκρίσεις) του δεύτερου συνελικτικού επιπέδου ομαδοποιούνται σε διανύσματα τα οποία και αποτελούν τις παραμέτρους στιγμιοτύπου του πρώτου επιπέδου από κάψουλες (PrimaryCaps). Μέσω της εκπαίδευσης, οι νευρώνες που προηγούνται των διανυσμάτων της κάθε κάψουλας δυνητικά μαθαίνουν να συσχετίζουν τις τιμές μεταξύ τους με τέτοιο τρόπο ώστε να αναπαριστούν ιδιότητες του ίδιου τμήματος αντικειμένου. Το τελευταίο επίπεδο είναι ένα επίπεδο από κάψουλες το οποίο αποτελεί και το επίπεδο εξόδου (ονομάζεται ως DigitCaps). Ο αριθμός των καψουλών στο επίπεδο εξόδου είναι τόσος όσος και ο αριθμός των κλάσεων ταξινόμησης[4]. Τέλος, προαιρετικά προτείνεται η χρήση ενός αποκωδικοποιητή για την ανακατασκευή της αρχικής εικόνας με είσοδο το διάνυσμα της κάψουλας που εμπεριέχει το διάνυσμα ιδιοτήτων του αντικειμένου με το μεγαλύτερο μήκος (ονομάζεται διάνυσμα πρόβλεψης).

Για τη διαμόρφωση των διανυσμάτων του τελευταίου επιπέδου από κάψουλες (DigitCaps) χρησιμοποιείται ένας αλγόριθμος δρομολόγησης με συμφωνία του οποίου η βασική λειτουργία περιγράφηκε στην ενότητα 2.2. Συγκεκριμένα, με τον προτεινόμενο αλγόριθμο «Δυναμικής Δρομολόγησης μέσω Συμφωνίας» (Dynamic Routing by Agreement), οι κάψουλες του προηγούμενου επιπέδου παράγουν μια πρόβλεψη για τις παραμέτρους στιγμιοτύπου της κάθε κάψουλας του επόμενου επιπέδου. Τις προβλέψεις αυτές τις δρομολογούν στο επόμενο επίπεδο βεβαρημένες από τις «παραμέτρους σύζευξης» που προσαρμόζονται από τον εν λόγω αλγόριθμο. Όταν πολλές προβλέψεις συμφωνούν για τις παραμέτρους στιγμιοτύπου μιας κάψουλας, τότε με αυτόν τον τρόπο συνδιαμορφώνουν τις παραμέτρους της και αυτή αποκτά μεγάλη τιμή πιθανότητας (το

[3] Σε ελεύθερη μετάφραση: «Δυναμική Δρομολόγηση μεταξύ Καψουλών».

[4] Σε μερικές εξαιρέσεις, χρησιμοποιείται μια παραπάνω κάψουλα εξόδου για την περίπτωση όπου το αντικείμενο εξόδου δεν ανήκει σε καμία από τις κλάσεις για τις οποίες το δίκτυο έχει εκπαιδευτεί να αναγνωρίζει.

διάνυσμά της έχει μεγάλο μέτρο). Μια ακόμα βελτίωση που οφείλεται στη χρήση αλγορίθμου δρομολόγησης είναι ότι σε αντίθεση με την προηγούμενη μέθοδο, πλέον δεν απαιτείται να παρέχεται κατά την εκπαίδευση κάποιος πίνακας μετασχηματισμού. Αντίθετα, το δίκτυο αποθηκεύει εσωτερικά πίνακες μετασχηματισμού οι οποίοι μαθαίνουν (μέσω εκπαίδευσης) να αναπαριστούν τις (ανεξάρτητες–στιγμιοτύπου) σχέσεις τμημάτων - όλου.

Η δημοσίευση Dynamic Routing Between Capsules παρείχε υποσχόμενα πειραματικά αποτελέσματα. Πιο συγκεκριμένα, δοκιμάστηκε στο σύνολο δεδομένων MNIST [41] όπου και είχε 0.25% σφάλμα ελέγχου (test error) με μόλις 8.2M παραμέτρους. Για σύγκριση, ένα τυπικό baseline συνελικτικό δίκτυο πετυχαίνει 0.39% σφάλμα ελέγχου (test error) με πολύ περισσότερες παραμέτρους (35.4M). Αξιοσημείωτες επιδόσεις παρατηρήθηκαν και στο MultiMNIST [76] σύνολο δεδομένων το οποίο αποτελείται από αριθμούς με υψηλή επικάλυψη μεταξύ τους. Σε αυτό το σφάλμα ελέγχου ήταν 5.2%, πολύ μικρότερο από αυτό του συνελικτικού μοντέλου (8.1%). Επίσης, βέλτιστα (για την εποχή) αποτελέσματα παρατηρήθηκαν στα σύνολα δεδομένων affNIST και smallNORB. Ειδικά οι υψηλές επιδόσεις στο πρώτο σύνολο, όπου περιέχει ψηφία μετασχηματισμένα από διάφορους αφινικούς μετασχηματισμούς, αποδεικνύει την ευρωστία των δικτύων με κάψουλες σε μεταβολές της οπτικής γωνίας. Τέλος, το δίκτυο δοκιμάστηκε στο (σύνθετο) σύνολο δεδομένων CIFAR10 αλλά η επίδοσή του σε αυτό δεν ήταν εντυπωσιακή (10.6% σφάλμα ελέγχου).

Matrix Capsules with EM Routing

Η επιστημονική μελέτη των Hinton G. et al.[5] [47] βελτιώνει την προηγούμενη υλοποίηση τροποποιώντας την αρχιτεκτονική του νευρωνικού δικτύου από κάψουλες (αυξάνοντας τον συνολικό αριθμό παραμέτρων) και προτείνοντας έναν νέο αλγόριθμο δρομολόγησης μέσω συμφωνίας βασιζόμενο στον αλγόριθμο Μεγιστοποίησης Προσδοκιών (Expectation Maximization).

Πιο αναλυτικά, οι βασικότερες τροποποιήσεις της προηγούμενης μελέτης είναι οι εξής:

1. Η κάθε κάψουλα διαθέτει ξεχωριστή λογιστική μονάδα (logistic unit) για την αναπαράσταση της πιθανότητας ύπαρξης της οντότητας που αναγνωρίζει. Αυτός ο τρόπος, σύμφωνα με τους Hinton G. et al. [47] είναι καλύτερος από την κωδικοποίηση της τιμής πιθανότητας στο μήκος του διανύσματος παραμέτρων στιγμιοτύπου.

2. Σαν μετρική ομοιότητάς μεταξύ των ψήφων χρησιμοποιείται ο αρνητικός λογάριθμος της διακύμανσης (variance) των Γκαουσιανών συστάδων. Αυτή η μετρική ομοιότητας είναι καλύτερη από την ομοιότητα συνημιτόνου (cosine similarity) καθώς είναι πιο ευαίσθητη στην περιοχή υψηλής ομοιότητας[6].

3. Στην νέα μελέτη προτείνεται μια ελαφρώς τροποποιημένη δομή κάψουλας η οποία ενθυλακώνει τις παραμέτρους στιγμιοτύπου υπό τη μορφή πίνακα πόζας με n στοιχεία. Αυτή η αλλαγή επιτρέπει στους πίνακες μετασχηματισμού να έχουν μέγεθος n^2 και όχι μόνο n.

4. Εισάγεται μια νέα πολυεπίπεδη αρχιτεκτονική η οποία περιλαμβάνει συνελικτικά επίπεδα από κάψουλες προκειμένου να διαμοιράζεται η γνώση (που αποθηκεύεται στη μορφή πινάκων

[5] Σε ελεύθερη μετάφραση: «Πινακοειδής Κάψουλες με Αλγόριθμο Δρομολόγησης Μεγιστοποίησης Προσδοκιών».

[6] Με άλλα λόγια, μπορεί καλύτερα να διακρίνει μια σχετικά καλή ομοιότητα από μια άριστη ομοιότητα.

μετασχηματισμού) στον χώρο.

Με τα πειράματα που έγιναν στο προτεινόμενο μοντέλο μηχανικής μάθησης αποδεικνύεται η αποδοτικότερη αναγνώριση αντικειμένων όταν αυτά αναπαρίστανται σε εικόνες με διαφορετικές γωνίες λήψης. Για παράδειγμα, για το σύνολο δεδομένων smallNORB επιτυγχάνεται σφάλμα ελέγχου ίσο με 1.4% (πολύ μικρότερο σε σχέση με το σφάλμα 5.2% του βασικού μοντέλου - αποτελούμενου από συνελικτικά επίπεδα). Επιπλέον, υψηλές επιδόσεις παρατηρήθηκαν όταν δοκιμάστηκε η προτεινόμενη αρχιτεκτονική νευρωνικού δικτύου με κάψουλες στο ίδιο σύνολο δεδομένων αλλά σε οπτικές γωνίες απεικονιζομένων αντικειμένων που δεν είχε εκπαιδευτεί (novel viewpoints). Τέλος, το μοντέλο φάνηκε να είναι εύρωστο σε επιθέσεις τύπου λευκού–κουτιού (white–box adversarial attacks) [97][7].

3.2 Παραλλαγές Νευρωνικών Δικτύων με Κάψουλες

Στην ενότητα αυτή θα γίνει σύντομη αναφορά στις βασικότερες έρευνες που σχετίζονται άμεσα με τα νευρωνικά δίκτυα από κάψουλες σε περιβάλλον επιβλεπομένης μάθησης. Οι έρευνες αυτές κυρίως εστιάζουν σε τροποποιήσεις του αλγορίθμου δρομολόγησης και της αρχιτεκτονικής του δικτύου. Ακόμα, περιλαμβάνονται ορισμένες εργασίες που πειραματίζονται εκτενώς με τις βασικές υλοποιήσεις, όπως τις περιγράψαμε παραπάνω.

Κατά τη διάρκεια της βιβλιογραφικής μελέτης των νευρωνικών δικτύων με κάψουλες απαιτείται να έχουμε υπόψη τα εξής κριτήρια:

- Αν οι βασικές ιδιότητες που σχετίζονται με την αποδοτική διαχείριση των αντικειμένων υπό διαφορετικές οπτικές γωνίες διατηρούνται (π.χ. εύρωστες εσωτερικές αναπαραστάσεις που μεταβάλλονται ανάλογα με τις αλλαγές στην οπτική γωνία, δυνατότητα αποθήκευσης γνώσης ανεξάρτητη από τη γωνία θέασης κ.α.).
- Αν υπάρχουν αλλαγές στις υποθέσεις που αφορούν τις σχέσεις μέρους–όλου.
- Αν οι κάψουλες ενεργοποιούνται μέσω πολυδιάστατης σύμπτωσης high–dimensional coincidences
- Πώς διαχειρίζεται το προτεινόμενο σύστημα την εγγενή αβεβαιότητα της σύνθεσης ενός αντικειμένου από τα τμήματά του. [98]

Σημειώνουμε ότι στις βιβλιογραφικές μελέτες στις οποίες αναφερόμαστε παρακάτω αποφύγαμε να συμπεριλάβουμε τα έργα που παρουσιάζουν μεγάλες αποκλίσεις από τα βασικά κριτήρια των νευρωνικών δικτύων με κάψουλες.

Capsule Routing via Variational Bayes

Η εν λόγω μελέτη[8] [99] βασίζεται στην [47] και τη βελτιώνει προτείνοντας έναν διαφορετικό αλγόριθμο δρομολόγησης μέσω συμφωνίας. Πιο συγκεκριμένα, με τον αλγόριθμο δρομολόγησης βασισμένο στη συμπερασματολογία διακύμανσης (Variational Inference) - ονομάζεται δρομολόγηση μπεϋζιανής διακύμανσης (Variational Bayed Routing) είναι εφικτή η μοντελοποίηση

[7] Έχει δειχθεί ότι δεν ισχύει το ίδιο για επιθέσεις τύπου μαύρου–κουτιού (black–box adversarial attacks)
[8] Σε ελεύθερη μετάφραση: «Δρομολόγηση Καψουλών με Μπεϋζιανή Διακύμανση».

αβεβαιότητας στις παραμέτρους της κάψουλας (εκτός από τους συντελεστές δρομολόγησης). Με αυτήν την πιθανοκρατική προσέγγιση, είναι εφικτή η τροποποίηση των πρότερων πιθανοτήτων της κάθε κάψουλας για καλύτερο έλεγχο της πολυπλοκότητάς τους και για αποφυγή του προβλήματος της κατάρρευσης διασποράς (variance collapse). Επιπλέον, δείχνουν τον τρόπο με τον οποίον ένα νευρωνικό δίκτυο από κάψουλες μπορεί να μετατραπεί σε αυτο–κωδικοποιητή διακύμανσης (variational auto–encoder). Τέλος, παρέχουν μερικές οδηγίες για την εκπαίδευση του προτεινόμενου μοντέλου (αρχικοποίηση βαρών και σχέδια κανονικοποίησης).

Οι πειραματισμοί του προτεινόμενου μοντέλου στα σύνολα δεδομένων smallNORB, SVHN, MNIST και affNIST αποδεικνύουν την ισχυριζόμενη βελτίωση της βασικής υλοποίησης των νευρωνικών δικτύων με κάψουλες. Πιο αναλυτικά, στο σύνολο δεδομένων smallNORB [96] επιτυγχάνεται μείωση του σφάλματος ελέγχου στην τιμή 1.55% (σε αντίθεση με 1.8% όπως προκύπτει από το [47]) χρησιμοποιώντας μόλις τον μισό αριθμό από κάψουλες. Βελτιωμένα αποτελέσματα παρατηρήθηκαν και στα υπόλοιπα σύνολα δεδομένων αλλά και σε πειράματα που δοκιμάζουν την ικανότητα γενίκευσης του δικτύου και την ευρωστία του σε αφινικούς μετασχηματισμούς. Τέλος, αποδεικνύεται ότι ένα δίκτυο που χρησιμοποιεί τον προτεινόμενο αλγόριθμο συγκλίνει κατά 20% γρηγορότερα, με αυξημένη αριθμητική ευστάθεια.

Introducing Routing Uncertainty in Capsule Networks

Η επόμενη δημοσίευση που εξετάζουμε [9] [98] τροποποιεί την προηγούμενη υλοποίηση ώστε να είναι πιο αποδοτική με βελτιωμένα πειραματικά αποτελέσματα. Αρχικά, εντοπίζει ορισμένα μειονεκτήματα των τοπικών, επαναληπτικών αλγορίθμων δρομολόγησης τα οποία είναι:

- Το υψηλό υπολογιστικό κόστος ενός επαναληπτικού, αλγορίθμου δρομολόγησης που λαμβάνει χώρα μεταξύ δύο διαδοχικών επιπέδων από κάψουλες.
- Κατά τη δρομολόγηση της πληροφορίας από το ένα επίπεδο καψουλών στο επόμενο λαμβάνονται υπόψη μόνο τα τοπικά συμφραζόμενα (local context), δηλαδή η πληροφορία μεταξύ των δύο επιπέδων.
- Η τάση για υπερπροσαρμογή ή υποπροσαρμογή (overfitting/underfitting) ανάλογα με την επιλογή των αριθμών επανάληψης του αλγορίθμου δρομολόγησης (routing iterations).

Για τον σκοπό αυτό, προτείνεται η αντικατάσταση των τοπικών επαναλήψεων (local iterations) με μια «σφαιρική εικόνα» (global view) βασισμένη στην προσέγγιση της εκ των υστέρων πιθανότητας διακύμανσης (variational posterior) στις συνδέσεις μέρους - όλου σε ένα πιθανοκρατικό μοντέλο. Η χρήση καθολικών κρυφών μεταβλητών (global latent variables) που επηρεάζουν άμεσα την αντικειμενική συνάρτηση (obective function) προσδίδει στο δίκτυο την ικανότητα για εποπτικότερη δρομολόγηση της πληροφορίας. Οι μεταβλητές αυτές ενημερώνονται μεροληπτικά (discriminatively) σύμφωνα με την αρχή του ελάχιστου μήκους περιγραφής (minimum description length) της θεωρίας πληροφορίας (information theory).

Τα εκτενή πειράματα στο σύνολο δεδομένων smallNORB αποδεικνύουν ότι ακόμα και με μικρότερο αριθμό παραμέτρων σε σχέση με τις προηγούμενες υλοποιήσεις των νευρωνικών δικτύων

[9] Σε ελεύθερη μετάφραση: «Εισάγοντας Αβεβαιότητα Δρομολόγησης στα Νευρωνικά Δίκτυα από Κάψουλες».

με κάψουλες, η επίδοση του δικτύου είναι ελαφρώς βελτιωμένη. Πολλά πειράματα επίσης διενεργήθηκαν με σκοπό να διασφαλιστεί ότι διατηρούνται οι βασικές ιδιότητες του εν λόγω είδους νευρωνικών δικτύων. Ενδεικτικά, εκτός από τα πειράματα στο σύνολο δεδομένων smallNORB και MultiMNIST, έγιναν πειράματα σχετικά με τη δυνατότητα γενίκευσης σε νέες οπτικές γωνίες, την ευρωστία σε αφινικούς μετασχηματισμούς των εικόνων εισόδου για τους οποίους το δίκτυο δεν έχει εκπαιδευτεί αλλά και την ικανότητά του να εκπαιδεύεται αποδοτικά με λίγα παραδείγματα (Few–Shot Learning). Σε όλα τα πειράματα, οι επιδόσεις ήταν πλήρως ικανοποιητικές, αποδεικνύοντας έτσι ότι τηρούνται οι βασικές υποθέσεις των νευρωνικών δικτύων με κάψουλες.

Group Equivariant Capsule Networks

Το έργο των Lenssen et al. [10] [100] προτείνει ένα τροποποιημένο είδος από κάψουλες και έναν αλγόριθμο δρομολόγησης βασιζόμενα στη θεωρεία ομάδων. Προκύπτει, από τον συνδυασμό μελετών τόσο στο αντικείμενο των νευρωνικών δικτύων με κάψουλες όσο και στη μελέτη που εισήγαγαν τα δίκτυα συνέλιξης ομάδας [101]. Με αυτόν τον τρόπο, το προτεινόμενο μοντέλο αποδεικνύεται ότι εγγυάται τις ιδιότητες της ανεξαρτησίας των παραμέτρων ενεργοποίησης των καψουλών και της ισοδύναμης διακύμανσης των παραμέτρων πόζας (ανάλογα με τις μεταβολές της οπτικής γωνίας του αντικειμένου εισόδου). Ιδιαίτερα ενδιαφέρον είναι ο τρόπος με τον οποίο δημιουργείται το πρώτο επίπεδο από κάψουλες (primary capsules). Αναλυτικότερα, δε χρησιμοποιούνται κάποια προσαρμοζόμενα φίλτρα από τον αλγόριθμο εκπαίδευσης αλλά γίνεται χρήση των (στατικών) φίλτρων Sobel. Τα πειράματα περιορίστηκαν στο σύνολο δεδομένων MNIST όπου επιτεύχθηκε εκπληκτική ακρίβεια ταξινόμησης των ψηφίων (98.42%) όταν αυτά είχαν περιστραφεί τυχαία με πολλαπλάσια των 90° και ενώ το δίκτυο είχε εκπαιδευτεί με ψηφία χωρίς κανένα μετασχηματισμό.

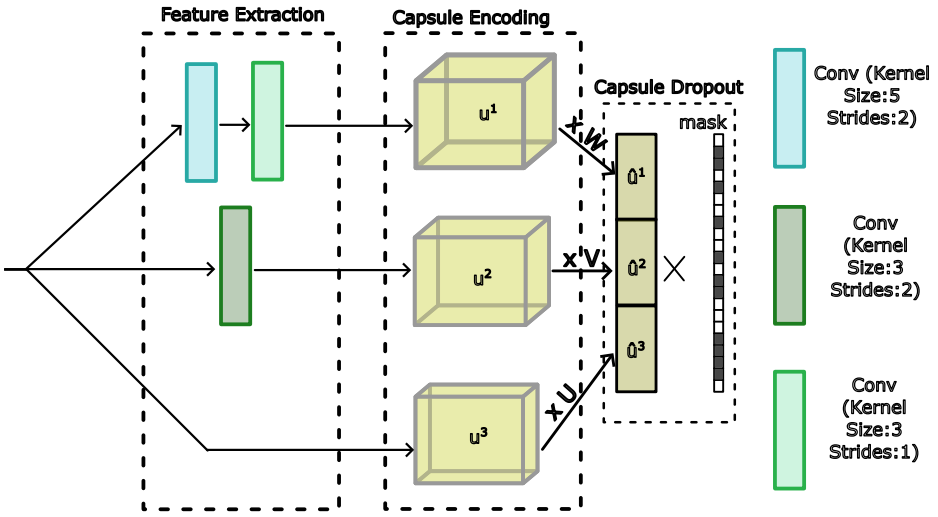
CapsuleGAN: Generative Adversarial Capsule Network

Στη δημοσίευση των Jaiswal et al. [11] [102] εφαρμόζεται η αρχιτεκτονική του νευρωνικού δικτύου με κάψουλες, όπως παρουσιάζεται από τους Sabour et al. [76] στο πλαίσιο των παραγωγικών, αντιπαραθετικών δικτύων. Συγκεκριμένα, πρόκειται περισσότερο για μια εφαρμογή που αντικαθίσταται το συνελικτικό δίκτυο διάκρισης (convolutional descriminative network) ενός παραγωγικού αντιπαραθετικού δικτύου (Generative Adversarial Network) με ένα δίκτυο διάκρισης από κάψουλες. Για την εκπαίδευση του δικτύου, χρησιμοποιούν μια συνάρτηση κόστους που προκαλεί το παιχνίδι αντιπαράθεσης (adversarial game) μεταξύ της γεννήτριας και του δικτύου διάκρισης, τροποποιημένη κατάλληλα για ένα δίκτυο διάκρισης από κάψουλες. Μέσα από τα πειράματα στα MNIST και CIFAR10 σύνολα δεδομένων, προκύπτει ότι ένα Παραγωγικό Αντιπαραθετικό Δίκτυο με Κάψουλες έχει καλύτερη επίδοση από ένα αντίστοιχο δίκτυο με αμιγώς συνελικτικά επίπεδα. Οι βελτιωμένες επιδόσεις εντοπίστηκαν στη μοντελοποίησης πιθανοτικής κατανομής των δεδομένων εικόνων (όπως προκύπτουν από τη μετρική παραγωγικής αντιπαράθεσης (GAM) [103] και από τα πειράματα ταξινόμησης εικόνων ημι–επιβλεπομένης μάθησης).

[10] Σε ελεύθερη μετάφραση: «Νευρωνικά Δίκτυα με Κάψουλες Ομάδας Ισοδύναμης Διακύμανσης».
[11] Σε ελεύθερη μετάφραση: «Παραγωγικό Αντιπαραθετικό Δίκτυο με Κάψουλες».

MS–CapsNet: A Novel Multi–Scale Capsule Network

Στη μελέτη των Xiang et al. [12] [104] παρουσιάζεται μια νέα αρχιτεκτονική νευρωνικών δικτύων με κάψουλες που βελτιώνει την ικανότητα αναπαράστασης ιεραρχικής πληροφορίας από τις κάψουλες, μειώνοντας παράλληλα την υπολογιστική πολυπλοκότητα. Η ιδέα πίσω από αυτή την τροποποίηση είναι ότι εξάγοντας πλουσιότερες αναπαραστάσεις, είναι δυνατή η βελτίωση της επίδοσης σε σύνθετα δεδομένα εισόδου. Επιπλέον, τροποποιείται η «τεχνική εγκατάλειψης» (dropout) προκειμένου να μπορεί να εφαρμοστεί σε ένα επίπεδο από κάψουλες. Τέλος, η αρχιτεκτονική δοκιμάζεται σε εργασίες ταξινόμησης στα σύνολα FashionMNIST και CIFAR10 όπου παρατηρούνται βελτιωμένα αποτελέσματα (ακρίβεια 0.927 και 0.757 αντίστοιχα με λιγότερες από τις μισές παραμέτρους) σε αντιπαραβολή με την υλοποίηση [76].



Σχήμα 3.2: Αρχιτεκτονική πολυ–κλιμακωτού νευρωνικού δικτύου με κάψουλες *Παράχθηκε από το Inkscape.*

Η αρχιτεκτονική του πρώτου και δεύτερου επιπέδου του δικτύου φαίνεται στο σχήμα 3.2. Έπειτα ακολουθεί το τελευταίο επίπεδο από κάψουλες (παρόμοια με το έργο [76]). Χωρίς να εμβαθύνουμε ιδιαίτερα, η εξαγωγή χαρακτηριστικών γίνεται πολυκλιμακωτά, με το πρώτο παρακλάδι να παράγει κωδικοποιήσεις σημασιολογικής πληροφορίας (πληροφορίας ανώτερης τάξης), το δεύτερο να παράγει διανύσματα που κωδικοποιούν πληροφορία μέσης τάξης και το τελευταίο, να έχει ως έξοδο τα ακατέργαστα χαρακτηριστικά. Τα εξαγόμενα χαρακτηριστικά οργανώνονται σε κάψουλες που εμπεριέχουν διανύσματα διαστατικότητας 12, 8 και 4 αντίστοιχα (ανάλογα με το παρακλάδι από το οποίο προκύπτουν). Τέλος, μέσω των πινάκων μετασχηματισμών W, V, U, παράγονται ψήφοι ίσου μεγέθους $\hat{u^1_{j|i}}, \hat{u^2_{j|i}}, \hat{u^3_{j|i}}$ για κάθε ζεύγος $i \to j$ που συνδέονται σειριακά (concatenate) σε ένα διάνυσμα $\hat{u_{j|i}} = concat(\hat{u^1_{j|i}}, \hat{u^2_{j|i}}, \hat{u^3_{j|i}})$.

Δυστυχώς, πέρα από τα προαναφερθέντα πειράματα, το δίκτυο δεν εξετάζεται κατά πόσον τηρεί τις θεμελιώδεις υποθέσεις των νευρωνικών δικτύων με κάψουλες. Επιπλέον, οι πίνακες W, V, U δεν έχουν τετραγωνική μορφή με αποτέλεσμα να μην είναι αναστρέψιμοι (και κατά συνέπεια, ο μετασχηματισμός δεν είναι γεωμετρικός).

[12] Σε ελεύθερη μετάφραση: «Ένα Νέο Πολυ–Κλιμακωτό Νευρωνικό Δίκτυο με Κάψουλες».

DDRM-CapsNet: Capsule Network Based on Deep Dynamic Routing Mechanism for Complex Data

Στο έργο των Liu et al. [13] [105] δοκιμάζεται μια πιο σύνθετη αρχιτεκτονική με περισσότερες παραμέτρους προκειμένου το δίκτυο να ανταποκρίνεται καλύτερα σε πιο σύνθετα σύνολα δεδομένων. Συγκεκριμένα, πειραματίζονται με διάφορες αρχιτεκτονικές που εισάγουν ένα, δύο, τρία ή τέσσερα συνελικτικά επίπεδα πριν το πρώτο επίπεδο από κάψουλες (PrimaryCaps). Επιπλέον, εισάγουν ένα ακόμα επίπεδο από κάψουλες τύπου DigitCaps με αποτέλεσμα ο (υπολογιστικά κοστοβόρος) αλγόριθμος δρομολόγησης να πρέπει να εφαρμοστεί δύο φορές κατά ένα πρόσθιο πέρασμα. Ακόμη, αυξάνουν την εκφραστικότητα της κάθε κάψουλας με την αύξηση της διάστασης του διανύσματος χαρακτηριστικών που ενθυλακώνουν.

Μετά από πειράματα, κατέληξαν στη βέλτιστη αρχιτεκτονική η οποία περιλαμβάνει τρία συνελικτικά επίπεδα και τρία επίπεδα από κάψουλες με το τελευταίο επίπεδο να έχει κάψουλες με μεγαλύτερα σε μήκος διανύσματα ($D = 24$). Χρησιμοποιώντας αυτή την παραμετροποίηση επιτεύχθηκε μεταξύ άλλων ακρίβεια 77.5% στο σύνολο δεδομένων CIFAR10 και 29.93% ακρίβεια στο σύνολο CIFAR100 (επιδόσεις βελτιωμένες κατά 11% και 8% αντίστοιχα σε σχέση με το [76]). Βέβαια, όλα αυτά γίνονται με αυξημένο υπολογιστικό κόστος και χωρίς να δοκιμάζεται αν συνεχίζουν να τηρούνται οι βασικές υποθέσεις των νευρωνικών δικτύων με κάψουλες.

DeepCapsnet: Going Deeper with Capsule Networks

Στο έργο των Rajasegaran et al. [14] [106] εισάγεται μια νέα αρχιτεκτονική προκειμένου να ανταποκρίνεται καλύτερα σε πιο σύνθετα δεδομένα. Η αρχιτεκτονική αυτή χρησιμοποιεί υπολειμματικές συνδέσεις (residual connections), ένα δίκτυο αποκωδικοποιητή ως μέθοδο ομαλοποίησης (regularization) και αποφυγής υπερπροσαρμογής (overfitting) που δέχεται μόνο το διάνυσμα της προβλεφθείσας κλάσης και ένα δυναμικό αλγόριθμο δρομολόγησης εμπνευσμένο από τρισδιάστατες συνελίξεις (3D convolutions).

Τα πειραματικά αποτελέσματα δείχνουν ότι το μοντέλο, συγκρινόμενο -μεταξύ άλλων- με το έργο [76] επιτυγχάνει ελαφρώς καλύτερα αποτελέσματα στα δεδομένα (CIFAR10, SVHN και FashionMNIST) με λιγότερες παραμέτρους (λόγω του διαμοιρασμού παραμέτρων που προσφέρει η συνέλιξη). Επίσης, μειώνουν το υπολογιστικό κόστος του δυναμικού αλγορίθμου δρομολόγησης που αναπτύσσουν μειώνοντας τον αριθμό των επαναλήψεων δρομολόγησης. Τέλος, παρατηρείται μια συνοχή στο τι αναπαριστά το κάθε στοιχείο του διανύσματος χαρακτηριστικών (των καψουλών του τελευταίου επιπέδου), ανεξαρτήτως κλάσης. Για παράδειγμα, το 28° στοιχείο του διανύσματος φαίνεται να επηρεάζει την κάθετη επιμήκυνση του ψηφίου, ανεξάρτητα από την κάψουλα εξόδου (και την κλάση του ψηφίου).

Αν και ο αλγόριθμος δρομολόγησης και η αρχιτεκτονική του προτεινόμενου δικτύου είναι αρκετά διαφορετική από τη βασική υλοποίηση [76], δεν έγιναν πειράματα που να αποδεικνύουν ότι οι βασικές ιδιότητες των νευρωνικών δικτύων με κάψουλες διατηρούνται. Μάλιστα, στην απλή περίπτωση του συνόλου δεδομένων MNIST, η επίδοση είναι πιο περιορισμένη.

[13] Σε ελεύθερη μετάφραση: «Νευρωνικό Δίκτυο από Κάψουλες Βασισμένο πάνω σε Βαθύ Δυναμικό Μηχανισμό Δρομολόγησης για Σύνθετα Δεδομένα».

[14] Σε ελεύθερη μετάφραση: «Πηγαίνοντας Βαθύτερα με τα Νευρωνικά Δίκτυα από Κάψουλες».

FSC–CapsNet: Fractionally–Strided Convolutional Capsule Network for complex data

Στο έργο των Liu et al. [15] [107] προτείνεται μια νέα αρχιτεκτονική η οποία βελτιώνει τόσο το κύριο μέρος του νευρωνικού δικτύου όσο και τον αποκωδικοποιητή. Πιο συγκεκριμένα, αυξάνεται ο αριθμός των συνελικτικών επιπέδων πριν από το πρώτο επίπεδο από κάψουλες προκειμένου να εξάγονται πιο πλούσια χαρακτηριστικά. Επιπρόσθετα, αναφορικά με τον αποκωδικοποιητή, χρησιμοποιούνται συνελικτικά επίπεδα κλιμακωτού βηματισμού (fractionally–strided convolutional layers) που αποσκοπούν να βελτιώσουν την ποιότητα των ανακατασκευασμένων εικόνων.

Μέσα από πειράματα σε πέντε σύνολα δεδομένων επιλέχθηκε η βέλτιστη παραλλαγή της προτεινόμενης αρχιτεκτονικής η οποία αποτελείται από τρία συνελικτικά επίπεδα (που προηγούνται του πρώτου επιπέδου από κάψουλες) και δύο συνελικτικά επίπεδα κλιμακωτού βηματισμού (fractionally–strided convolutional layers). Ενδεικτικά για τα πειραματικά αποτελέσματα, στο CIFAR10 επιτεύχθηκε ποσοστό ακρίβειας 77.53% ενώ στο CIFAR100 το αντίστοιχο ποσοστό ήταν 25.83%. Ακόμα, παρατηρήθηκε (οπτικά) καλύτερη ανακατασκευή των εικόνων λόγω του βελτιωμένου αποκωδικοποιητή. Δυστυχώς, δεν έγιναν τα απαραίτητα πειράματα για να διασφαλιστεί ότι οι ιδιότητες των νευρωνικών δικτύων με κάψουλες διατηρούνται (invariance on capsule activations, equivariance on capsule poses).

Self–Attention Capsule Networks for Object Classification

Στο έργο των Hoogi et al. [16] [108] παρουσιάζεται μια πρωτότυπη αρχιτεκτονική νευρωνικών δικτύων με κάψουλες η οποία ενσωματώνει μηχανισμό αυτο–προσοχής. Αναλυτικότερα, μεταξύ του συνελικτικού επιπέδου και του πρώτου επιπέδου από κάψουλες παρευρίσκεται ένα επίπεδο που υλοποιεί τον μηχανισμό αυτο–προσοχής, όπως τον περιγράψαμε στην ενότητα 2.3.3. Με αυτόν τον τρόπο, βελτιώνεται η αρχιτεκτονική του νευρωνικού δικτύου χωρίς να αυξάνεται σημαντικά η υπολογιστική πολυπλοκότητα του δικτύου.

Η προτεινόμενη αρχιτεκτονική δοκιμάστηκε σε έξι σύνολα δεδομένων, τα τρία εκ των οποίων αποτελούν σύνολα από ιατρικές εικόνες. Αναλυτικότερα για το διαδεδομένο σύνολο δεδομένων CIFAR10, παρατηρήθηκε 3.55% βελτίωση σε σχέση με την υλοποίηση στο [76] ενώ στο σύνολο δεδομένων MNIST δεν παρατηρήθηκε κάποια αξιοσημείωτη βελτίωση. Είναι σημαντικό να επισημάνουμε ότι το μοντέλο δε δοκιμάστηκε σε σύνολα δεδομένων όπως το affNIST και συνεπώς δεν μπορούμε να εγγυηθούμε τη διατήρηση των θεμελιωδών ιδιοτήτων σχετικά με την ικανότητά τους να γενικεύουν σε νέες οπτικές γωνίες.

DA–CapsNet: Dual Attention Mechanism Capsule Network

Στη μελέτη των Huang W. και Zhou F. [17] [109] δοκιμάζεται μια αρχιτεκτονική νευρωνικών δικτύων με κάψουλες που περιέχει διπλό μηχανισμό προσοχής. Αναλυτικότερα, εφαρμόζεται μηχανισμός προσοχής τόσο μεταξύ των συνελικτικών επιπέδων και του πρώτου επιπέδου από

[15] Σε ελεύθερη μετάφραση: «Νευρωνικά Δίκτυα με Κάψουλες Κλασματικού Βηματισμού για Σύνθετα Δεδομένα».

[16] Σε ελεύθερη μετάφραση: «Νευρωνικά Δίκτυα με Κάψουλες και Μηχανισμό Αυτο–προσοχής για την Ταξινόμηση Αντικειμένων».

[17] Σε ελεύθερη μετάφραση: «Νευρωνικά Δίκτυα με Κάψουλες και Διπλό Μηχανισμό Προσοχής».

κάψουλες (ονομάζεται Conv–Attention) όσο και μεταξύ του πρώτου επιπέδου από κάψουλες (PrimaryCaps) και του τελευταίου επιπέδου από κάψουλες (ο συγκεκριμένος μηχανισμός διαφέρει από το ν προηγούμενο και ονομάζεται Caps–Attention). Αναφορικά με τον πρώτο, μοιάζει περισσότερο με ένα επίπεδο προσοχής το οποίο γίνεται στους διαφορετικούς χάρτες χαρακτηριστικών (στα κανάλια) αφού η εξαγωγή των βαρών προσοχής περιλαμβάνει τη μέση συνάθροιση (average pooling) με πυρήνα το πλάτος και ύψος των χαρτών χαρακτηριστικών. Αναφορικά με το δεύτερο μηχανισμό προσοχής, περιλαμβάνει εκπαιδευόμενα βάρη προσοχής που εφαρμόζονται ανά 10 κάψουλες. Μετά από την εφαρμογή του Caps–Attention, σχηματίζονται άλλες κάψουλες που συμμετέχουν στον μηχανισμό δυναμικής δρομολόγησης, όπως περιγράφεται στο [76].

Ο διπλός μηχανισμός προσοχής που ενσωματώνεται στο προτεινόμενο μοντέλο αποσκοπεί στη βελτίωση της αξίας της πληροφορίας που περιγράφεται από τις κάψουλες, στην ελάττωση της περιττής πληροφορίας και στη βελτίωση της ιεραρχίας των καψουλών. Αποδεικνύεται (με μια διαδικασία που προσομοιάζει ablation study) ότι ο διπλός μηχανισμός προσοχής οδηγεί σε καλύτερα πειραματικά αποτελέσματα σε σχέση με άλλες παραλλαγές νευρωνικών δικτύων με κάψουλες που χρησιμοποιούν μονό μηχανισμό προσοχής. Το δίκτυο δοκιμάζεται σε έξι σύνολα δεδομένων και παρουσιάζει σε όλα βελτιωμένη απόδοση σε σχέση με τη βασική υλοποίηση του [76]. Ενδεικτικά, στα σύνολα δεδομένων MNIST, CIFAR10 και smallNORB παρατηρούνται ποσοστά ακρίβειας ταξινόμησης 99.53, 85.47 και 98.26 αντίστοιχα. Κλείνοντας, αξίζει να σημειώσουμε πως αν και το μοντέλο δοκιμάστηκε στο σύνολο δεδομένων smallNORB, δεν εξετάστηκε η ικανότητα γενίκευσης σε νέες οπτικές γωνίες, για τις οποίες δεν έχει εκπαιδευτεί.

Quick–CapsNet (QCN): A Fast Alternative to Capsule Networks

Στη μελέτη των Shiri et al. [18] [110] αναγνωρίζεται η αδυναμία των νευρωνικών δικτύων με κάψουλες αναφορικά με την ταχύτητα εκπαίδευσης και πρόβλεψης. Για τον σκοπό αυτό γίνονται μια σειρά από πειράματα για τη βελτίωση του υπολογιστικού κόστους με όσο το δυνατόν μικρότερη επίπτωση στις επιδόσεις ταξινόμησης. Αναλυτικότερα, αν εξαιρέσουμε τον (προαιρετικό) αποκωδικοποιητή ανεξαρτήτου–κλάσης (class–independent) που περιλαμβάνει συνελικτικά επίπεδα κλιμακωτού βηματισμού (fractionally–strided convolutional layers) και την αντικατάσταση του δευτέρου συνελικτικού επιπέδου με ένα πλήρως διασυνδεδεμένο επίπεδο, δεν προτείνεται κάποια άλλη τροποποίηση της αρχιτεκτονικής ή του αλγορίθμου δρομολόγησης επί της βασική υλοποίησης. Με άλλα λόγια, η ελάττωση του υπολογιστικού κόστους βασίζεται στη μεταβολή των υπερπαραμέτρων της βασικής υλοποίησης όπως περιγράφεται στο [76].

Μετά από την παρατήρηση ότι ο αριθμός των καψουλών επηρεάζει καθοριστικά την πολυπλοκότητα του δικτύου, έγιναν πειράματα σε σύνολα δεδομένων (MNIST, FashionMNIST, SVHN, CIFAR10, affNIST) ύστερα από τον περιορισμό του αριθμού των καψουλών του πρώτου επιπέδου (PrimaryCaps) από 1152 που είναι στη βασική υλοποίηση σε 8 είτε 6 είτε 4. Παρατηρήθηκε ότι αν και υπήρξε ελάττωση της ακρίβειας, οι χρόνοι εκπαίδευσης και πρόβλεψης αυξάνονταν σημαντικά. Ενδεικτικά, για το σύνολο δεδομένων MNIST και για την παραλλαγή που χρησιμοποιεί μόλις έξι πρωταρχικές κάψουλες (PrimaryCaps) και βελτιωμένο αποκωδικοποιητή, η ακρίβεια έπεσε από 99.47% σε 99.19% ενώ η ταχύτητα εκπαίδευσης και πρόβλεψης αυξήθηκε κατά 5.5 φορές. Τέλος, έγιναν πειράματα για να διασφαλιστεί η ευρωστία του προτεινόμενου μοντέλου στους αφινικούς

[18] Σε ελεύθερη μετάφραση: «Μια Ταχεία Εναλλακτική των Νευρωνικών Δικτύων με Κάψουλες».

μετασχηματισμούς - βασική ιδιότητα των νευρωνικών δικτύων με κάψουλες. Στα πειράματα αυτά, παρατηρήθηκε μια πτώση ακρίβειας ταξινόμησης της τάξης του 10%.

CapsNet vs CNN: Analysis of the Effects of Varying Feature Arrangement

Το έργο των Manogaran et al. [19] [111] εξετάζει την εγκυρότητα της βασικής υπόθεσης ότι τα νευρωνικά δίκτυα με κάψουλες μπορούν να διαχειρίζονται καλύτερα τις σχέσεις μερών–όλου και δεν υποφέρουν από το «πρόβλημα του Picasso». Για την εξέταση της υπόθεσης αυτής, κατασκευάζονται δύο απλά σύνολα δεδομένων. Το πρώτο αποτελείται από παραλληλόγραμμα και ισόπλευρα τρίγωνα. Το δεύτερο, αποτελείται από συνδυασμούς των δύο απλών αντικειμένων (ή μερών) που προαναφέραμε για τη σύνθεση αντικειμένων που είναι βέλη και μη–βέλη (τα μη βέλη προκύπτουν από τυχαίες διατάξεις ενός ορθογώνιου παραλληλογράμμου και ενός τριγώνου). Στη συνέχεια, ένα συνελικτικό δίκτυο και ένα νευρωνικό δίκτυο με κάψουλες (όπως περιγράφεται στο έργο [76] αφού τροποποιήθηκε κατάλληλα) εκπαιδεύτηκαν στο πρώτο σύνολο δεδομένων. Έτσι, τα συνελικτικά επίπεδα των δύο δικτύων έμαθαν να αναγνωρίζουν τα επιμέρους αντικείμενα (τρίγωνα και ορθογώνια παραλληλόγραμμα). Έπειτα, παγώνοντας (freeze) τα συνελικτικά επίπεδα, τα δύο δίκτυα μετεκπαιδεύτηκαν στο δεύτερο σύνολο δεδομένων στο οποίο και τελικά εξετάστηκαν. Από τα πειράματα προέκυψε ότι τα συνελικτικά δίκτυα μπορούσαν καλύτερα να διακρίνουν τα βέλη από τα μη–βέλη σε σχέση με τα νευρωνικά δίκτυα από κάψουλες. Δηλαδή, μπορούσαν να διαχειρίζονται καλύτερα τις σχέσεις μερών (ορθογώνιων παραλληλογράμμων και τριγώνων) και αντικειμένων (βελών). Αυτό είναι μια ένδειξη που κλονίζει μια από τις συνηθισμένες υποθέσεις των νευρωνικών δικτύων από κάψουλες. Παρόλα αυτά, σύμφωνα με τους συγγραφείς, είναι πιθανό το προτέρημα των συνελικτικών νευρωνικών δικτύων να πηγάζει από την απλότητα του συνόλου δεδομένων.

R–CapsNet: An Improvement of Capsule Network for More Complex Data

Το έργο των Luo et al. [20] [112] προτείνει μια αποδοτική αρχιτεκτονική νευρωνικού δικτύου με 45% λιγότερες παραμέτρους από αυτήν που περιγράφεται στο [76]. Το επιτυγχάνει αυτό εισάγοντας ένα επιπλέον συνελικτικό επίπεδο και μειώνοντας το μέγεθος των πυρήνων συνέλιξης (convolution kernels). Αναφορικά με τα πειράματα, δοκιμάζεται στα σύνολα δεδομένων CIFAR10 και FashionMNIST. Στο τελευταίο, η επίδοση στο σύνολο ελέγχου ανέρχεται στο 93.89% (έναντι 92.57% της βασικής υλοποίησης [76]). Δυστυχώς, δε γίνεται λόγος για πειράματα που εξετάζουν τη διατήρηση των βασικών υποθέσεων της εν λόγω τεχνολογίας νευρωνικών δικτύων.

CapsNet Based on Encoder and Decoder for Object Detection

Στο έργο των Luo et al. [21] [113] εξετάζεται η δυνατότητα μοντελοποίησης της μεταφοράς (translation equivariance) από νευρωνικά δίκτυα με κάψουλες. Για τον σκοπό αυτό, δίνεται ιδιαίτερη έμφαση στην κατασκευή ενός βελτιωμένου αποκωδικοποιητή (με τη χρήση συνελικτικών επιπέδων

[19] Σε ελεύθερη μετάφραση: «CapsNet vs CNN: Ανάλυση της Επίδρασης της Διακύμανσης της Χωρικής Διαρρύθμισης των Χαρακτηριστικών».

[20] Σε ελεύθερη μετάφραση: «Μια Βελτίωση των Νευρωνικών Δικτύων με Κάψουλες για Σύνθετα Δεδομένα».

[21] Σε ελεύθερη μετάφραση: «Νευρωνικό Δίκτυο με Κάψουλες Βασισμένο σε Αρχιτεκτονική Κωδικοποιητή–Αποκωδικοποιητή για την Αναγνώριση Αντικειμένων».

κλιμακωτού βηματισμού - fractionally–strided convolutional layers[22]). Αποδεικνύεται μέσα από πειράματα σε ένα ελαφρώς τροποποιημένο σύνολο δεδομένων MNIST - στο οποίο τα ψηφία τοποθετούνται σε τυχαίες θέσεις πάνω σε ένα μεγάλο «καμβά» - ότι η πληροφορία που κωδικοποιείται στις παραμέτρους στιγμιοτύπου της κάθε κάψουλας του τελευταίου επιπέδου (DiggitCaps) εμπεριέχει και τη θέση του ψηφίου στην εικόνα εισόδου. Οι παρατηρήσεις αυτές, ισχύουν και όταν στην εικόνα εισόδου υπάρχουν πολλαπλά ψηφία. Παρόλα αυτά, οι μεταφορές είναι το μόνο είδος μετασχηματισμού στο οποίο το προτεινόμενο δίκτυο δοκιμάζεται (είδος μετασχηματισμού που και τα συνελικτικά νευρωνικά δίκτυα διαχειρίζονται αποδοτικά).

Efficient–CapsNet: Capsule Network with Self–Attention Routing

Η μελέτη των Mazzia et al. [23] [49] αντικαθιστά τον αργό, επαναληπτικό αλγόριθμο δρομολόγησης με έναν μηχανισμό αυτο–προσοχής. Μέσα από μια αποδοτική αρχιτεκτονική με πολύ λιγότερες παραμέτρους (160K) είναι δυνατή η εξαγωγή ισχυρών αναπαραστάσεων (κωδικοποιημένων στα διανύσματα των καψουλών) και η επίτευξη αξιοσημείωτων αποτελεσμάτων σε μια πληθώρα από σύνολα δεδομένων.

Αναλυτικότερα για την αρχιτεκτονική του προτεινόμενου δικτύου, τα πρώτα επίπεδά του συγκροτούνται από τρία συνελικτικά επίπεδα τα οποία ακολουθούνται από ένα επίπεδο διαχωρίσιμης συνέλιξης κατά βάθος (depthwise separable convolution). Το επίπεδο αυτό μειώνει σημαντικά τον αριθμό των παραμέτρων και συνθέτει τις πρωταρχικές κάψουλες (PrimaryCaps). Έπειτα ακολουθεί το τελευταίο επίπεδο από κάψουλες «ψηφίου» (DigitCaps). Τα διανύσματα των καψουλών αυτών σχηματίζονται μέσω ενός μηχανισμού αυτο–προσοχής (λεπτομέρειες του μηχανισμού αυτού θα περιγραφούν στο επόμενο κεφάλαιο).

Σε ό,τι αφορά το πειραματικό μέρος της μελέτης, χρησιμοποιήθηκαν τα σύνολα δεδομένων MNIST, smallNORB και MultiMNIST. Σε αυτά, το ποσοστιαίο σφάλμα ελέγχου ήταν αντίστοιχα 0.26%, 2.54% και 5.1% (χωρίς ensemble). Τα προαναφερθέντα αποτελέσματα είναι στο ίδιο εύρος με αυτά διαφορετικών υλοποιήσεων των νευρωνικών δικτύων από κάψουλες αλλά χρησιμοποιώντας πολύ λιγότερες παραμέτρους. Αν και τα σύνολα δεδομένων που χρησιμοποιήθηκαν πιστοποιούν ότι ορισμένες ιδιότητες της τεχνολογίας διατηρούνται (π.χ. ευρωστία σε επικαλυπτόμενα αντικείμενα), δεν εξερευνάται η ικανότητα γενίκευσης σε νέες οπτικές γωνίες (για τις οποίες το δίκτυο δεν έχει εκπαιδευτεί). Επιπλέον, όπως θα δούμε στη συνέχεια, ο νέος αλγόριθμος δρομολόγησης που προτείνεται δεν παρέχει ανατροφοδότηση από τις κάψουλες γονείς στις κάψουλες παιδιά (top–down feedback).

Capsule Network Performance on Complex Data

Στη μελέτη των Xi et al. [24] [114] παρουσιάζονται τα αποτελέσματα πειραμάτων στα σύνολα δεδομένων CIFAR10 και MNIST όπως διενεργήθηκαν σε μια σειρά από παραλλαγές της βασική υλοποίησης του νευρωνικού δικτύου με κάψουλες [76]. Ενδεικτικά, οι αλλαγές αφορούσαν την προσθήκη επιπλέον συνελικτικών επιπέδων, την προσθήκη επιπλέον επιπέδων από κάψουλες, την αύξηση του αριθμού των καψουλών, την αλλαγή των παραμέτρων της συνάρτησης σφάλματος

[22] Ονομάζονται στον χώρο των νευρωνικών δικτύων και ως deconvolutions ή και transposed convolutions
[23] Σε ελεύθερη μετάφραση: «Νευρωνικό Δίκτυο από Κάψουλες με Μηχανισμό Αυτο–Προσοχής».
[24] Σε ελεύθερη μετάφραση: «Επίδοση Νευρωνικών Δικτύων με Κάψουλες σε Σύνθετα Δεδομένα».

κ.α. Από τους πειραματισμούς προκύπτει ότι η αύξηση των συνελικτικών επιπέδων οδηγεί σε μια μικρή βελτίωση της επίδοσης ενώ η αύξηση των επιπέδων από κάψουλες έχει το αντίστροφο αποτέλεσμα. Συνολικά, καμία από τις παραλλαγές δεν παρήγαγε αξιοσημείωτα αποτελέσματα.

Reasearch on Image Classification Based on Capsnet

Στο επιστημονικό έργο των Dong Z. και Lin S. [25] [115] κατασκευάζονται δύο μοντέλα ταξινόμησης εικόνων με το πρώτο να είναι βασισμένο στην τεχνολογία των «παραδοσιακών» συνελικτικών νευρωνικών δικτύων (που ενσωματώνουν επίπεδα συνάθροισης) και το δεύτερο να είναι ένα δίκτυο από κάψουλες (παρόμοιο με το [76]). Μέσα από τα πειράματα, υποστηρίζει τα προτερήματα του δεύτερου μοντέλου ενώ επισημαίνει τα μειονεκτήματά του, όταν το σύνολο δεδομένων γίνεται πιο σύνθετο.

An Optimization View on Dynamic Routing Between Capsules

Στη μελέτη των Wang D. και Liu Q. [26] [116] επιδιώκεται η χρήση μιας τυπικής, μαθηματικής διατύπωσης για την περιγραφή της στρατηγικής του αλγορίθμου δυναμικής δρομολόγησης μέσω συμφωνίας [76]. Πιο αναλυτικά, αποδεικνύει ότι αυτή είναι παρόμοια με την ελαχιστοποίηση του σφάλματος ομαδοποίησης με τη χρήση της μεθόδου κανονικοποίησης KL επί των συντελεστών σύζευξης (minimizing a standard clustering loss with KL regularization on the coupling probabilities). Επιπλέον, προτείνει έναν νέο αλγόριθμο δρομολόγησης που, σε ένα περιορισμένο πλαίσιο εφαρμογής, εμφανίζει ορισμένα προτερήματα έναντι του βασικού αλγορίθμου δυναμικής δρομολόγησης.

Analysis of Capsule Network (Capsnet) Architectures and Applications

Το έργο των Pande et al. [27] [117] επιχειρεί να συνοψίσει ορισμένες από τις πρόσφατες εξελίξεις των νευρωνικών δικτύων με κάψουλες αλλά και μερικές εφαρμογές τους. Επιπρόσθετα, κάνει αναφορά σε νέες επιτυχίες των συνελικτικών νευρωνικών δικτύων προκειμένου να εμπνεύσει την επιστημονική μελέτη. Από την περιγραφή των βελτιώσεων της βασικής υλοποίησης της εν λόγω τεχνολογίας γίνεται αντιληπτό το μεγάλο εύρος αυτών. Με άλλα λόγια, διαπιστώνεται ότι οι βελτιώσεις εστιάζουν τόσο στην κατασκευή αποδοτικότερου αλγορίθμου δρομολόγησης και μηχανισμού ανακατασκευής εικόνας όσο και στην ποιοτικότερη εξαγωγή χαρακτηριστικών.

A Convolutional Neural Network Based on a Capsule Network with Strong Generalization for Bearing Fault Diagnosis

Η μελέτη των Zhu et al. [28] [118] περιγράφει την εφαρμογή των νευρωνικών δικτύων με κάψουλες σε ένα σύστημα εποπτείας της υγείας περιστρεφόμενου εξοπλισμού (health condition monitoring system of rotating machinery). Η καινοτομία της παρούσας εφαρμογής σε ότι αφορά την

[25] Σε ελεύθερη μετάφραση: «Έρευνα στην Ταξινόμηση Εικόνων Βασισμένη στα Νευρωνικά Δίκτυα από Κάψουλες».

[26] Σε ελεύθερη μετάφραση: «Μια Βελτίωση του Αλγορίθμου Δυναμικής Δρομολόγησης Μεταξύ Καψουλών».

[27] Σε ελεύθερη μετάφραση: «Ανάλυση των Αρχιτεκτονικών και Εφαρμογών των Νευρωνικών Δικτύων με Κάψουλες».

[28] Σε ελεύθερη μετάφραση: «Ένα Νευρωνικό Δίκτυο με Κάψουλες με Ισχυρή Γενίκευση για τη Διάγνωση Σφαλμάτων σε Έδρανα Μηχανών».

τεχνολογία των νευρωνικών δικτύων από κάψουλες έγκειται στην ελαφρώς τροποποιημένη αρχιτεκτονική του χρησιμοποιούμενου δικτύου. Αναλυτικότερα, προτείνουν την χρήση ενός μπλόκ έναρξης (inception block) και επιπλέον χρησιμοποιούν το μήκος του μεγαλύτερου διανύσματος για εργασία παλινδρόμησης (regression task). Τα πειράματα που διενεργήθηκαν στο πλαίσιο της εφαρμογής αποδεικνύουν ότι οι προτεινόμενες αλλαγές οδηγούν σε καλύτερη γενίκευση.

Identifying Aggression and Toxicity in Comments using Capsule Network

Το έργο των Srivastava et al. [29] [119] είναι το μόνο (σύμφωνα με την παρούσα βιβλιογραφική έρευνα) που χρησιμοποιεί νευρωνικά δίκτυα με κάψουλες σε περιβάλλον επεξεργασίας φυσικής γλώσσας. Η αρχιτεκτονική που χρησιμοποιείται έχει την δομή του [76] αλλά για την αρχικοποίηση των πρωτεύοντων καψουλών (PrimaryCaps) αντί για συνελικτικά δίκτυα χρησιμοποιείται LSTM. Το προτεινόμενο δίκτυο εμφάνισε πολύ ενθαρρυντικά αποτελέσματα στο σύνολο δεδομένων Kaggle–toxic comment (98.46 ROC AUC).

3.3 Νευρωνικά Δίκτυα με Κάψουλες με Μη–Επιβλεπόμενη Μάθηση

Στην παρούσα ενότητα θα γίνει συνοπτική αναφορά σε ορισμένες παραλλαγές των νευρωνικών δικτύων με κάψουλες οι οποίες ανήκουν στο γενικότερο πλαίσιο της μη–επιβλεπόμενης μάθησης (unsupervised learning) αλλά και της αυτο–επιβλεπόμενης μάθησης (self–supervised learning). Οι παρακάτω μελέτες έχουν πραγματοποιηθεί από τον Hinton G. και την ομάδα του προκειμένου να στρέψουν την τεχνολογία που ανέπτυξαν σε μια νέα κατεύθυνση. Τέλος, να σημειωθεί ότι όλα τα έργα που παρουσιάζονται στην παρούσα ενότητα έπονται χρονικά των τριών βασικών έργων των νευρωνικών δικτύων με κάψουλες σε περιβάλλον επιβλεπόμενης μάθησης.

Stacked Capsule Atoencoders

Η μελέτη των Kosiorek et al. [30] [120] ορίζει μια καινούρια μέθοδο για την εκμάθηση χαρακτηριστικών (feature learingn). Αναλυτικότερα, η εκμάθηση εύρωστων αναπαραστάσεων (viewpoint-equivariant representations) γίνεται με την χρήση ειδικών κωδικοποιητών και δύο αποκωδικοποιητών. Ο πρώτος αποκωδικοποιητής μαθαίνει να αποσυνθέτει τα αντικείμενα εισόδου σε τμήματα (εξάγοντας πληροφορίες για την πιθανότητα ύπαρξης και την πόζα τους) ενώ ο δεύτερος μαθαίνει να συνθέτει τα τμήματα για τον σχηματισμό αντικειμένων.

Τα πειράματα που διενεργούνται σε απλά σύνολα δεδομένων είναι αρκετά ενθαρρυντικά. Για παράδειγμα, σε εργασίες μη επιβλεπόμενης ταξινόμησης επιτυγχάνεται ακρίβεια ταξινόμησης 55% και 98.7% για τα σύνολα δεδομένων SVHN και MNIST αντίστοιχα. Παρόλα αυτά, σε πιο σύνθετα σύνολα δεδομένων (π.χ. CIFAR10) τα αποτελέσματα δεν είναι το ίδιο ικανοποιητικά. Αυτό οφείλεται στο γεγονός ότι ο πρώτος στη σειρά αποκωδικοποιητής που προαναφέραμε χρησιμοποιεί

[29] Σε ελεύθερη μετάφραση: «Εντοπίζοντας Επιθετικότητα και Τοξικότητα σε Σχόλια Χρησιμοποιώντας Νευρωνικά Δίκτυα με Κάψουλες».

[30] Ο τίτλος του έργου θα μπορούσε να αποδοθεί στα Ελληνικά ως: «Στοιβαγμένοι Αυτο–κωδικοποιητές με Κάψουλες».

στάνταρ μοτίβα απλών αντικειμένων (fixed templates), γεγονός που περιορίζει την δυνατότητα μοντελοποίησης εικόνων πραγματικού κόσμου.

Umsupervised Part Representation by Flow Capsules

Στο έργο των Sabour et al. [31] [121] προτείνεται ένα σύστημα βασισμένο σε αυτο–επιβλεπόμενη μάθηση που αποσκοπεί να βελτιώσει την ποιότητα των περιγραφητών (θέση, σχήμα) των επιμέρους τμημάτων (part descriptors) που αναπαριστώνται από τις κάψουλες. Το επιτυγχάνει αυτό μέσα από μια σύνθετη αρχιτεκτονική κωδικοποιητή - αποκωδικοποιητή που αξιοποιεί την κίνηση σε διαδοχικά καρέ βίντεο. Πιο συγκεκριμένα, συγκρίνοντας τα διαδοχικά καρέ και παράγοντας ένα οπτικό πεδίο ροής με βάση αυτά επιτυγχάνουν να διακρίνουν τα (κινούμενα) αντικείμενα - μέρη.

Το προτεινόμενο μοντέλο δοκιμάζεται τόσο στην ικανότητά του για μη–επιβλεπόμενη κατάτμηση όσο και στην ικανότητά του για μη–επιβλεπόμενη ταξινόμηση (τροποποιώντας κατάλληλα το δίκτυο σε ένα FlowSCAE, όπως προκύπτει από τον συνδυασμό του μοντέλου SCAE με την προτεινόμενη αρχιτεκτονική). Αναφορικά με την πρώτη εργασία, χρησιμοποιήθηκαν τα σύνολα δεδομένων Geo και Exercise όπου το σκορ όπως μετρήθηκε χρησιμοποιώντας την μετρική IoU - Intersection over Union ήταν 0.96 και 0.58 αντίστοιχα. Τέλος, για την μη–επιβλεπόμενη ταξινόμηση, ενδεικτικά έχουμε σκορ 0.79 και 0.99 της K –Μέσης ακρίβειας συστάδων για 4 και 100 συστάδες αντίστοιχα (K–Means clustering accuracy where K equals 4 and 100 respectively).

Canonical Capsules: Unsupervised Capsles in Canonical Pose

Στο έργο των Sun et al. [32] [122] παρουσιάζεται μια μη–επιβλεπόμενη αρχιτεκτονική νευρωνικού δικτύου με κάψουλες για την διαχείριση τρισδιάστατων δεδομένων (3D point–cloud data). [33] Μέσω μιας αρχιτεκτονικής κωδικοποιητή - αποκωδικοποιητή γίνεται εφικτή η αποσύνθεση τρισδιάστατων εικόνων σε τμήματα διατηρώντας επιθυμητές ιδιότητες της αναγνώρισης αντικειμένων σε νέες οπτικές γωνίες (invariance - equivariance). Επίσης, γίνεται εφικτή η έμμεση εκμάθηση ενός μη ρητού (implicit) συστήματος αναφοράς (frame of referance) για το κάθε τρισδιάστατο αντικείμενο. Κλείνοντας, σε ότι αφορά το πειραματικό μέρος, το προτεινόμενο μοντέλο επιτυγχάνει βέλτιστη (state-of-the-art) επίδοση σε εργασίες μη–επιβλεπόμενης ανακατασκευής, ανάθεσης (registration) και ταξινόμησης.

3.4 Λοιπές Δημοσιεύσεις που Ενέμπνευσαν τις Μεθόδους της Παρούσας Διπλωματικής

Την τελευταία ενότητα του παρόντος κεφαλαίου την αφιερώνουμε στην παρουσίαση ορισμένων έργων που δεν σχετίζονται άμεσα με το θέμα της διπλωματικής. Τα έργα αυτά, είτε αφορούν το

[31] Ο τίτλος του έργου θα μπορούσε να αποδοθεί στα Ελληνικά ως: «Μη–επιβλεπόμενη Αναπαράσταση Τμημάτων από Κάψουλες Ροής».

[32] Ο τίτλος του έργου θα μπορούσε να αποδοθεί στα Ελληνικά ως: «Μη–επιβλεπόμενες Κάψουλες σε Κανονικοποιημένες Πόζες».

[33] Σημειώνουμε ότι εδώ (όπως και στις υπόλοιπες αναφορές της παρούσας ενότητας), οι κάψουλες δεν έχουν την ίδια δομή με αυτές που παρουσιάστηκαν στην ενότητα 2.2. Συνήθως, περιέχουν πληροφορία για τόσο για την πόζα του αντικειμένου (θέση ή και προσανατολισμό) και το σχήμα του.

γενικότερο πρόβλημα της διαχείρισης αντικειμένων υπο νέες οπτικές γωνίες είτε ενέμπνευσαν τις μεθόδους που παρουσιάζουμε στα επόμενα κεφάλαια.

How to Represent Part–Whole Hierarchies in a Neural Network

Η παρούσα μελέτη[34] [77] αφορά τις ιδέες του Hinotn G. σχετικά με ένα (θεωρητικό) μοντέλο υπολογιστικής όρασης υπό το όνομα GLOM. Πιο συγκεκριμένα, το μοντέλο συνδυάζει μετασχηματιστές (transformers), νευρωνικά πεδία (neural fields), αντιθετική μάθηση (contrastive learning), νευρωνικά δίκτυα με κάψουλες (capsule networks), αυτοκωδικοποιητές αποθορυβοποίησης (denoising autoencoders) και αναδρομικά νευρωνικά δίκτυα (RNNs). Αυτό το φανταστικό (imaginary) σύστημα αποσυνθέτει την εικόνα εισόδου σε ένα ιεραρχικό δέντρο (parse tree) από απεικονιζόμενα αντικείμενα και τα τμήματά τους. Επίσης, να σημειώσουμε ότι το ιεραρχικό δέντρο κατασκευάζεται δυναμικά (ανάλογα με την εκάστοτε εικόνα εισόδου) χωρίς να αλλάζει η αρχιτεκτονική του νευρωνικού δικτύου. Αυτό επιτυγχάνεται με την χρήση ενός πολυβηματικού αλγορίθμου συμφωνίας (multi–step consensus algorithm) που αναλύει τα σημεία της εικόνας παράλληλα και πολυκλιμακωτά (δηλαδή με διαφορετικά επίπεδα αφαιρετικότητας). Συμπερασματικά, το μοντέλο GLOM αν και δεν αποτελεί κάποιο υλοποιημένο μοντέλο, αφορά μια ιδέα που αλλάζει ριζικά τον τρόπο κατανόησης (και αναπαράστασης) οπτικών πολυμέσων από τα μοντέρνα συστήματα υπολογιστικής όρασης.

GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields

Το έργο των Niemeyer M. και Geiger A.[35] [123] παρουσιάζει ένα παραγωγικό μοντέλο (generative model) που παρέχει ευελιξία στην σύνθεση (φανταστικών) εικόνων. Πιο συγκεκριμένα, με την εσωτερική αναπαράσταση των σκηνών ως συνθετικά, παραγωγικά, νευρωνικά πεδία χαρακτηριστικών (compositional generative neural feature fields) είναι δυνατή η απόπλεξη των παραγόντων διακύμανσης (disentangle factors of variation) όπως για παράδειγμα την ελαχιστοποίηση της αλληλοσυσχέτησης μεταξύ των αντικειμένων σε πρώτο πλάνο και του υπόβαθρου της εικόνας. Έτσι, ο δημιουργός μπορεί να παράξει τεχνικά άρτιες, φανταστικές εικόνες έχοντας έλεγχο στο υπόβαθρο, στα επιμέρους αντικείμενα (και στην γεωμετρία τους) αλλά και στην γωνία θέασης.

Representing Scenes as Neural Radiance Fields for View Synthesis

Στο έργο των Mildenhall et al. [36] [124] παρουσιάζεται ένα σύστημα το οποίο είναι ικανό, δοσμένων ορισμένων δισδιάστατων εικόνων που αποτελούν προβολές μιας τρισδιάστατης σκηνής, να συνθέτει νέες προβολές υπό γωνίες θέασης που δεν ανήκουν στα δεδομένα εισόδου. Το επιτυγχάνει αυτό ενσωματώνοντας την γνώση για μια σκηνή στα βάρη ενός νευρωνικού δικτύου πρόσθιας τροφοδότησης το οποίο εκπαιδεύεται μέσω της οπισθοδιάδοσης σφάλματος σε

[34] Ο τίτλος του έργου θα μπορούσε να αποδοθεί στα Ελληνικά ως: «Πώς να Αναπαριστούμε ιεραρχίες Μέρους–Όλου σε ένα Νευρωνικό Δίκτυο».

[35] Ο τίτλος του έργου θα μπορούσε να αποδοθεί στα Ελληνικά ως: «Αναπαριστώντας Σκηνές ως Συνθετικά Παραγωγικά Νευρωνικά Πεδία Χαρακτηριστικών».

[36] Ο τίτλος του έργου θα μπορούσε να αποδοθεί στα Ελληνικά ως: «Αναπαριστώντας Σκηνές ως Νευρωνικά Ακτινωτά Πεδία για τη Σύνθεση σε Νέες Οπτικές Γωνίες».

μια διαδικασία διαφορικής απόδοσης όγκου (differential volume rendering procedure). Μετά την εκπαίδευση, το δίκτυο τροφοδοτείται με μια πεντάδα στοιχείων που προσδιορίζουν την θέση και την γωνία λήψης και συνθέτει μια ρεαλιστική προβολή η οποία σέβεται τις δομικές λεπτομέρειες των αντικειμένων της σκηνής καθώς επίσης και τον φωτισμό και τις ανακλάσεις.

Κεφάλαιο 4

Μέθοδος

Όπως αναφέραμε στο προηγούμενο κεφάλαιο, οι θεμελιακές υλοποιήσεις των νευρωνικών δικτύων από κάψουλες ([47,75,76]) βασίζονται σε υποθέσεις των οποίων η εγκυρότητα δεν έχει δοκιμαστεί εκτενώς. Για τον λόγο αυτό (αλλά και για λόγους σύγκρισης με άλλες μεθόδους), δύο από τις τέσσερεις μεθόδους που χρησιμοποιούμε στο πειραματικό μέρος της παρούσας διπλωματικής αναπτύχθηκαν (ή τροποποιήθηκαν) σε πηγαίο κώδικα σύμφωνα με την αρχιτεκτονική και τον αλγόριθμο δρομολόγησης που παρουσιάζονται στα έργα [76] και [47] αντίστοιχα.

Οι δύο τελευταίες μέθοδοι του παρόντος κεφαλαίου, πατώντας στις βασικές δομικές αρχές των νευρωνικών δικτύων με κάψουλες, επιδιώκουν να βελτιώσουν ορισμένες από τις ανεπάρκειες της τεχνολογίας. Έτσι, προτείνονται καινοτόμοι αλγόριθμοι και αρχιτεκτονικές νευρωνικών δικτύων που εστιάζουν είτε σε προβλήματα ταχύτητας και κλιμακωσιμότητας (τρίτη μέθοδος) είτε στα προβλήματα εγκυρότητας των αρχικών υποθέσεων (τέταρτη μέθοδος).

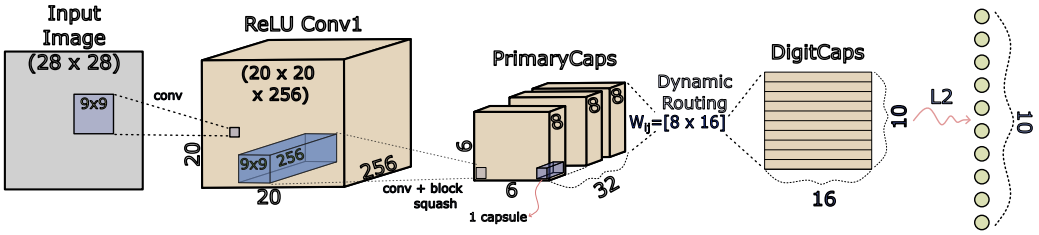
4.1 Dynamic Routing Between Capsules

Η μέθοδος της «Δυναμικής Δρομολόγησης με Κάψουλες» υλοποιήθηκε σε κώδικα ακολουθώντας πιστά την ομώνυμη δημοσίευση των Sabour S. et al. [76]. Αν και υπήρχαν έτοιμες υλοποιήσεις του έργου στο διαδίκτυο, δυστυχώς ο κώδικάς τους ήταν αναχρονισμένος και δε λειτουργούσε σε σύγχρονα συστήματα με τις νέες εκδόσεις των πακέτων λογισμικού. Για αυτό και υλοποιήθηκε, αντλώντας στοιχεία από αυτόν τον κώδικα, εκ νέου στη γλώσσα python 3 χρησιμοποιώντας τη βιβλιοθήκη tensorflow 2.

Σε αυτήν την ενότητα θα ξεκινήσουμε παρουσιάζοντας τη γενική αρχιτεκτονική του νευρωνικού δικτύου με κάψουλες που προτείνει το εν λόγω έργο. Στη συνέχεια και σε ξεχωριστή υπο–ενότητα θα παρουσιάσουμε τον αλγόριθμο δρομολόγησης που λαμβάνει χώρα μεταξύ των δύο διαδοχικών επιπέδων από κάψουλες ενώ παράλληλα θα κάνουμε ορισμένες σχετικές σημειώσεις. Έπειτα, θα αναφερθούμε στη συνάρτηση σφάλματος που χρησιμοποιείται για την εκπαίδευση του νευρωνικού δικτύου κάτω από τις διάφορες συνθήκες παραμετροποίησης. Τέλος, για λοιπές λεπτομέρειες υλοποίησης όπως τιμές αρχικοποίησης, τύπος βελτιστοποιητή (optimizer) κτλ. παραπέμπουμε τον αναγνώστη στην ιστοσελίδα όπου είναι αναρτημένος ο κώδικάς μας.

4.1.1 Αρχιτεκτονική Νευρωνικού Δικτύου

Το βασικό μοντέλο που χρησιμοποιείται στην πλειοψηφία των πειραμάτων μας παρουσιάζεται στο σχήμα 4.1. Αποτελείται από 3 επίπεδα εκ των οποίων τα πρώτα δύο είναι συνελικτικά (ονόματι Conv1 και PrimaryCaps) και το τρίτο πλήρως διασυνδεδεμένο (επίπεδο DigitCaps).



Σχήμα 4.1: Η αρχιτεκτονική του νευρωνικού δικτύου με κάψουλες της πρώτης μεθόδου. *Παράχθηκε από το Inkscape.*

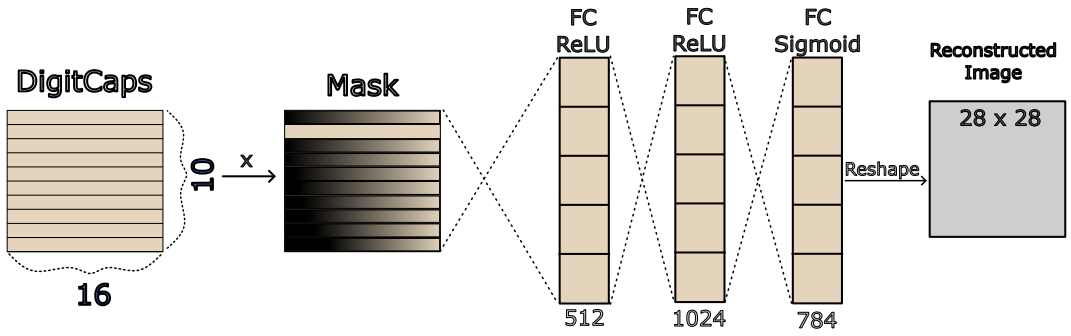
Πιο αναλυτικά, το επίπεδο Conv1 έχει 256 πυρήνες συνέλιξης μεγέθους 9×9, βήμα (stride) ίσο με 1 και συνάρτηση ενεργοποίησης ReLU.

Αναφορικά με το επίπεδο PrimaryCaps, ο ρόλος του είναι αυτός των ανάστροφων γραφικών, όπως εξηγήσαμε στην ενότητα 2.2. Πρακτικά, αποτελεί ένα συνελικτικό επίπεδο με 32 κανάλια από 8D κάψουλες–διανύσματα. Δηλαδή, κάθε κάψουλα περιέχει 8 μονάδες συνέλιξης με πυρήνες μεγέθους 9×9 και βήμα 2. Κατά αυτόν τον τρόπο, κάθε κάψουλα «βλέπει» $256 \times 9 \times 9$ στοιχεία (μονάδες) από τους χάρτες χαρακτηριστικών του προηγούμενου επιπέδου.

Όπως φαίνεται και από το σχετικό σχήμα, με αυτό το επίπεδο σχηματίζεται ένα πλέγμα από $32 \times 6 \times 6$ κάψουλες. Να σημειώσουμε ότι επειδή το επίπεδο PrimaryCaps αποτελεί ένα συνελικτικό επίπεδο από κάψουλες, τα βάρη των πυρήνων (κυλιόμενων παραθύρων) σε κάθε πλαίσιο στους άξονες $x-y$ μεγέθους 6×6 διαμοιράζονται. Η ειδοποιός διαφορά με ένα συνελικτικό επίπεδο χωρίς κάψουλες είναι στη συνάρτηση ενεργοποίησης. Αυτή αφενός είναι μια συνάρτηση σύνθλιψης (squashing function) που θα ορίσουμε στη συνέχεια και αφετέρου εφαρμόζεται σε ομάδες 8 στοιχείων τη φορά (δηλαδή σε κάθε 8D κάψουλα). Συνεπώς, το επίπεδο PrimaryCaps μπορεί να θεωρηθεί ως απλό συνελικτικό επίπεδο με 32×8 κανάλια στο οποίο, ανά ομάδες των 8 στοιχείων στη διάσταση βάθους, εφαρμόζεται μη γραμμικότητα τύπου «μπλοκ» (block non–linearity).

Το τελευταίο επίπεδο είναι το DigitCaps το οποίο έχει σαν έξοδο (συνήθως 10 από) 16D κάψουλες–διανύσματα των οποίων το μήκος (L_2 νόρμα), όπως έχουμε προαναφέρει, χρησιμοποιείται για τον εντοπισμό της κλάσης πρόβλεψης. Ανάμεσα στα πλήρως διασυνδεδεμένα επίπεδα PrimaryCaps και DigitCaps λαμβάνει χώρα ο αλγόριθμος δρομολόγησης μέσω συμφωνίας που αναλύουμε παρακάτω. Όπως φαίνεται και από το σχήμα, μεταξύ των δύο επιπέδων παρεμβάλλονται και μια σειρά από πίνακες βαρών ($6 \times 6 \times 32 \times 10$ τέτοιοι διαφορετικοί πίνακες διάστασης 8×16) που μετασχηματίζουν τις τιμές της κάθε 8D κάψουλας του επιπέδου PrimaryCaps σε 16D ψήφους, μια για κάθε κάψουλα γονέα (άρα 10 ψήφοι αντιστοιχούν σε κάθε κάψουλα επιπέδου PrimaryCaps). Επισημαίνουμε ότι οι πίνακες βαρών τροποποιούνται κατά την εκπαίδευση και δεν εξαρτώνται από το εκάστοτε μεμονωμένο παράδειγμα (αποθηκεύουν πληροφορία μεταξύ μερών –όλου που είναι ανεξάρτητη από την οπτική γωνία).

Ανακατασκευή



Σχήμα 4.2: Η αρχιτεκτονική του αποκωδικοποιητή που διευκολύνει την εκπαίδευση του νευρωνικού δικτύου με κάψουλες. *Παράχθηκε από το Inkscape.*

Προκειμένου να ενθαρρυνθεί η σύλληψη των παραμέτρων στιγμιότυπου του κάθε ψηφίου από το αντίστοιχο διάνυσμα DigitCaps χρησιμοποιείται ένας αποκωδικοποιητής του σχήματος 4.2. Αυτός δομείται από 3 πλήρως διασυνδεδεμένα επίπεδα με αριθμό νευρώνων $512, 1024$ και 28×28 αντίστοιχα. Να σημειώσουμε ότι μεταξύ του επιπέδου DigitCaps και του πρώτου πλήρως διασυνδεδεμένου επιπέδου του αποκωδικοποιητή παρεμβάλλεται μια μάσκα που σκοπό έχει να μηδενίσει όλες τις κάψουλες εξόδου παρά αυτήν που σχετίζεται με το ψηφίο–στόχο (target) κατά τη διάρκεια εκπαίδευσης ή αυτή που έχει το μεγαλύτερο μήκος κατά τη διάρκεια ελέγχου. [1] Έτσι, ο αποκωδικοποιητής δέχεται σαν είσοδο μόνο ένα διάνυσμα (DigitCap) και έχει ως έξοδο μια εικόνα 28×28 που μέσω εκπαίδευσης επιδιώκεται να είναι όσο πιο πιστό αντίγραφο γίνεται της εικόνας εισόδου.

4.1.2 Συνάρτηση Σύνθλιψης

Όπως έχουμε αναφέρει, τόσο για τον υπολογισμό των PrimaryCaps όσο και για τον υπολογισμό των DigitCaps μέσω του αλγορίθμου δρομολόγησης, χρησιμοποιείται η συνάρτηση σύνθλιψης (squashing function). Πρόκειται για μια μη γραμμική συνάρτηση που δέχεται ένα διάνυσμα τυχαίου μήκους και εγγυάται ότι στην έξοδό της, το μήκος του διανύσματος θα ανήκει στο διάστημα $(0, 1)$. Με αυτόν τον τρόπο, το μήκος του διανύσματος μοντελοποιεί την πιθανότητα ενεργοποίησης της εκάστοτε κάψουλας.

Με μαθηματικούς όρους, έχουμε:

$$squash(x) = \frac{\|x\|^2}{1 + \|x\|^2} \frac{x}{\|x\|} \tag{4.1}$$

[1] Όπως έχει γίνει σαφές, ο αύξοντας αριθμός της κάψουλας με τη μεγαλύτερη $\mathcal{L}_2$ νόρμα (μήκος) είναι και η κλάση πρόβλεψης $\hat{y}$.

4.1.3 Δυναμικός Αλγόριθμος Δρομολόγησης μέσω Συμφωνίας

Ο αλγόριθμος δρομολόγησης μέσω συμφωνίας λαμβάνει χώρα μεταξύ δύο επιπέδων από κάψουλες και όπως έχουμε εξηγήσει αναλαμβάνει τον ρόλο της ανάθεσης μερών των αντικειμένων σε ένα από τα αντικείμενα στόχους. Πιο αναλυτικά, η διαδικασία που λαμβάνει χώρα για τον σχηματισμό των καψουλών γονέων από τις κάψουλες παιδιά είναι η εξής:

1. Από την κάθε μια κάψουλα παιδί (έστω u_i) παράγονται τόσοι ψήφοι ($\hat{u}_{j|i}$) όσες και οι κάψουλες γονείς. Οι ψήφοι αποτελούν προβλέψεις της πόζας της εκάστοτε κάψουλας γονέα και υπολογίζονται πολλαπλασιάζοντας την κάψουλα παιδί με τον πίνακα βαρών W_{ij} που συνδέει το τμήμα του αντικειμένου i με την οντότητα που αναπαριστά η κάψουλα j. Έτσι έχουμε:
$$\hat{u}_{j|i} = W_{ij} u_i \tag{4.2}$$

2. Μέσω του αλγορίθμου δρομολόγησης που θα εξετάσουμε στη συνέχεια, υπολογίζονται τα διανύσματα των καψουλών γονέων (s_j) ως σταθμισμένα αθροίσματα των ψήφων παιδιών, δηλαδή:
$$s_j = \sum_i c_{ij} \hat{u}_{j|i} \tag{4.3}$$
όπου τα c_{ij} είναι οι παράμετροι σύζευξης (coupling coefficients). Αυτές υπολογίζονται σύμφωνα με τη συνάρτηση softmax, δηλαδή:
$$c_{ij} = \frac{exp(b_{ij})}{\sum_k exp(b_{ik})} \tag{4.4}$$
ανάλογα με τα b_{ij} (log priors) που τροποποιούνται σε κάθε επανάληψη του αλγορίθμου.

Πλέον, είμαστε σε θέση να παρουσιάσουμε το δυναμικό αλγόριθμο δρομολόγησης μέσω συμφωνίας αναλυτικά. Αυτός, όπως έχει γίνει σαφές, λαμβάνει μέρος αμέσως μετά τον υπολογισμό των ψήφων.

Algorithm 1 Δυναμικός Αλγόριθμος Δρομολόγησης μέσω Συμφωνίας

function ROUTING($\hat{u}_{j|i}, r, l$)
  Αρχικοποίηση:
    $\forall$ κάψουλα $i \in$ επίπεδο l και κάψουλα $j \in$ επίπεδο $(l+1)$: $b_{ij} \leftarrow 0$
  for r επαναλήψεις **do**
    $\forall$ κάψουλα $i \in$ επίπεδο l: $c_i \leftarrow softmax(b_i)$
    $\forall$ κάψουλα $j \in$ επίπεδο $(l+1)$: $s_j \leftarrow \sum_i c_{ij} \hat{u}_{j|i}$
    $\forall$ κάψουλα $j \in$ επίπεδο $(l+1)$: $v_j \leftarrow squash(s_j)$
    $\forall$ κάψουλα $i \in$ επίπεδο l και κάψουλα $j \in$ επίπεδο $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} \times v_j$
  end for
  return v_j
end function

Μερικές Σημειώσεις Σχετικά με τον Αλγόριθμο Δρομολόγησης

Προκειμένου να διευκολύνουμε τον αναγνώστη στη διαισθητική κατανόηση αλλά και πρακτική υλοποίηση του αλγορίθμου 1 κάνουμε τις εξής παρατηρήσεις:

- Σε κάθε κάψουλα γονέα (DigitCaps) αντιστοιχούν τόσες ψήφοι όσες είναι και οι κάψουλες παιδιά (PrimaryCaps). Επειδή κάθε ψήφος παράγεται από έναν μοναδικό πίνακα βαρών W_{ij}, γίνεται αντιληπτό ότι κάθε γονέας «βλέπει» διαφορετικές ψήφους.

- Αρχικοποιώντας τα b_{ij} με την τιμή 0, στην πρώτη επανάληψη του αλγορίθμου δρομολόγησης διαμορφώνονται τα διανύσματα των καψουλών γονέων (s_j) ως οι μέσοι όροι των ψήφων που τους αντιστοιχούν.

- Σε κάθε επανάληψη, κάθε κάψουλα παιδί i τροποποιεί τα βάρη δρομολόγησής της (coupling coefficients) c_{ij} ανάλογα με το πόσο συμφωνεί η πρόβλεψή του ($\hat{u}_{j|i}$) με το διάνυσμα s_j της κάθε κάψουλας γονέα. Η συμφωνία αυτή εκτιμάται με τη χρήση του εσωτερικού γινομένου $\hat{u}_{j|i} \times v_j$.

- Όταν το διάνυσμα πρόβλεψης $\hat{u}_{j|i}$ παρουσιάζει μεγάλη συμφωνία με το διάνυσμα s_j, η ποσότητα c_{ij} θα αυξηθεί, προκαλώντας (πιθανότατα) ακόμα μεγαλύτερη συμφωνία στην επόμενη επανάληψη. Αναπόφευκτα, λόγω του περιορισμού $\sum_j c_{ij} = 1$, αν μια κάψουλα γονέας (έστω k) παρουσιάζει μικρή συμφωνία με την αντίστοιχη πρόβλεψη $\hat{u}_{k|i}$ της κάψουλας i, η ποσότητα c_{ik} θα μειωθεί. Συνεπώς, η συγκεκριμένη κάψουλα παιδί i θα έχει μικρό ποσοστό στο μερίδιο διαμόρφωσης του διανύσματος s_k.

- Κάψουλες για τις οποίες πολλές ψήφοι συμφωνούν καταλήγουν μετά από λίγες επαναλήψεις του αλγορίθμου δρομολόγησης (συνήθως 3) να έχουν διανύσματα με μεγάλο μήκος (και άρα μεγάλη πιθανότητα). Αντίθετα, κάψουλες γονείς που έχουν μικρή συμφωνία με τις προβλέψεις των παιδιών, λαμβάνουν μικρά σε μήκος διανύσματα (λόγω των μικρών c_{ij}) που αντικρούουν το ένα το άλλο (cancel each other out) κατά την πράξη του σταθμισμένου αθροίσματος.

- Πρακτικά, επειδή η κάθε κάψουλα γονέα j διαμορφώνεται από τις αντίστοιχες ψήφους των καψουλών παιδιών, μεγάλη συμφωνία μεταξύ αυτών προκύπτει από μεγάλη συμφωνία των παιδιών στον χώρο των ψήφων (για τον γονέα j). Με άλλα λόγια, η ύπαρξη μιας πυκνής συστάδας (cluster) από ψήφους για την πόζα ενός γονέα j συνεπάγεται ότι το διάνυσμα του γονέα s_j θα έχει μεγάλο μέτρο[2]. Με αυτήν την παρατήρηση, ο όρος «φιλτραρίσματος μέσω της πολυδιάστατης σύμπτωσης» (high dimensional coincidence filtering) γίνεται πλήρως κατανοητός.

4.1.4 Συνάρτηση Σφάλματος

Για την εκπαίδευση του νευρωνικού δικτύου με κάψουλες εισάγεται η συνάρτηση «Απώλειας Περιθωρίου» (Margin Loss). Μέσω αυτής της συνάρτησης, κατά την εκπαίδευση ενθαρρύνεται η προσαρμογή των βαρών του δικτύου ώστε το διάνυσμα DigitCap που αναπαριστά την εκάστοτε

[2]Με εξαίρεση αν, λόγω ανταγωνισμού, υπάρχουν πολλές, πυκνότερες συστάδες που αντιστοιχούν σε άλλες κάψουλες γονείς.

κλάση πρόβλεψης $\hat{y}$ να έχει μεγάλο μήκος ενώ τα άλλα διανύσματα που αντιστοιχούν στις υπόλοιπες κλάσεις που δεν εντοπίζονται στην εικόνα εισόδου να έχουν (σχεδόν) μηδενικό μήκος. Επιπλέον, επιτρέπει την εκπαίδευση σε περιπτώσεις όπου στην εικόνα εισόδου απαντώνται περισσότερα του ενός αντικείμενα που ανήκουν σε διαφορετικές κλάσεις. Αναλυτικότερα, η συνάρτηση σφάλματος ορίζεται ξεχωριστά για κάθε κάψουλα DigitCap k ως εξής:

$$L_k = T_k max(0, m^+ - \|v_k\|)^2 + \lambda(1 - T_k)max(0, \|v_k\| - m^-)^2 \tag{4.5}$$

όπου $T_k = 1$ αν και μόνο αν το αντικείμενο κλάσης k βρίσκεται στην εικόνα εισόδου. Επίσης, συνήθως είναι $m^+ = 0.9$ και $m^- = 0.1$ ενώ η τιμή της παραμέτρου λ είναι $\lambda = 0.5$. Η συνάρτηση σφάλματος προκύπτει ως:

$$\mathcal{L} = \sum_k L_k. \tag{4.6}$$

Σε περίπτωση που χρησιμοποιείται και το πλήρως διασυνδεδεμένο δίκτυο αποκωδικοποιητή που έχει ως έξοδο προβλέψεις (σε μορφή διανύσματος) $\hat{x}$ τότε έχουμε:

$$\mathcal{L} = \sum_k L_k + \lambda_{rec} \|x - \hat{x}\|^2 \tag{4.7}$$

όπου $\lambda_{rec} = 0.0005$ ο όρος που μειώνει τη βαρύτητα του σφάλματος ανακατασκευής.

4.1.5 Παραλλαγές Δυναμικού Αλγορίθμου Δρομολόγησης

Στο πειραματικό μέρος που ακολουθεί το παρόν κεφάλαιο, πειραματιστήκαμε με ορισμένες παραλλαγές του αλγορίθμου αυτού προκειμένου να εξετάσουμε τη σημασία του αλγορίθμου δρομολόγησης. Πιο αναλυτικά, κατασκευάσαμε σε αδρές γραμμές δύο νέους αλγορίθμους που δε χρησιμοποιούν τον επαναληπτικό αλγόριθμο δρομολόγησης αλλά χρησιμοποιούν έναν παραπλήσιο, ταχύ αλγόριθμο δρομολόγησης. Τους αλγορίθμους αυτούς τους ονομάζουμε Argmax Routing και Max Routing. Τους παρουσιάζουμε παρακάτω με τη σειρά που τους αναφέραμε.

Αλγόριθμος Argmax Scaled Routing και Argmax Routing

Ο παρόν αλγόριθμος σκοπό έχει να εξετάσει αν υψηλές επιδόσεις επιτυγχάνονται με την επιλογή μόνο μιας κάψουλας παιδί (κάψουλας επιπέδου l) για δρομολόγηση στην εκάστοτε κάψουλα του επόμενου επιπέδου. Όπως θα φανεί από τα πειραματικά αποτελέσματα, δεν απαιτείται ο γραμμικός συνδυασμός των ψήφων για τη συγκρότηση των διανυσμάτων του επόμενου επιπέδου. Η επιλογή ενός «εκπροσώπου» για κάθε κάψουλα επιπέδου $(l+1)$ με κριτήριο το ποια ψήφος για την κάψουλα έχει το μεγαλύτερο βάρος δρομολόγησης θα δούμε στη συνέχεια πως είναι μια αποδοτική μέθοδος δρομολόγησης.

Οι δύο αλγόριθμοι που παρουσιάζονται παρακάτω εμφανίζουν, με μια πρώτη ματιά, ελάχιστες διαφορές. Η διαφορά τους έγκειται στο ότι ενώ ο πρώτος αλγόριθμος, κλιμακώνει τις επιλεχθέντες ψήφους (ψήφοι «εκπρόσωποι» της εκάστοτε κάψουλας επιπέδου $l + 1$) με τα αντίστοιχα βάρη δρομολόγησης και ύστερα εφαρμόζει τη συνάρτηση squash, ο δεύτερος απλά έχει σαν έξοδο τις μη–κλιμακωμένες (unscaled), επιλεχθέντες ψήφους.

Αναφορικά με τον αλγόριθμο 2, είναι σημαντικό να τονίσουμε ότι δεν αναιρεί τις βασικές υποθέσεις των νευρωνικών δικτύων από κάψουλες. Και πάλι χρησιμοποιούνται ορισμένες επαναλήψεις του δυναμικού αλγορίθμου προκειμένου να πραγματοποιηθεί η πολυδιάστατη συμφωνία (high–dimensional coincidence filtering). Αυτή τη φορά όμως, ο βαθμός συμφωνίας ενσωματώνεται στα βάρη δρομολόγησης (τα οποία και τελικά κλιμακώνουν τις κάψουλες επιπέδου $(l+1)$). Σε τελική ανάλυση, οι ψήφοι που συμφωνούν για την πόζα μιας κάψουλας επόμενου επιπέδου θα έχουν (εν γένει) μεγαλύτερα βάρη δρομολόγησης λόγω του ανταγωνισμού που υπάρχει μεταξύ των καψουλών γονέων. Έτσι, κλιμακώνοντας τις τελικές κάψουλες με τα βάρη δρομολόγησης, αυξάνουμε το μήκος των διανυσμάτων που εκπροσωπούν κάψουλες επιπέδου $(l+1)$ στις οποίες υπήρξε μεγάλη συμφωνία ψήφων ενώ προκαλούμε το αντίθετο αποτέλεσμα σε κάψουλες με χαμηλή συμφωνία.

Algorithm 2 Αλγόριθμος Argmax Scaled Routing

function ARGMAX SCALED ROUTING($\hat{u}_{j|i}, r, l$)
  Αρχικοποίηση:
    $\forall$ κάψουλα $i \in$ επίπεδο l και κάψουλα $j \in$ επίπεδο $(l+1)$: $b_{ij} \leftarrow 0$
  for r επαναλήψεις **do**
    $\forall$ κάψουλα $i \in$ επίπεδο l: $c_i \leftarrow softmax(b_i)$
    $\forall$ κάψουλα $j \in$ επίπεδο $(l+1)$: $s_j \leftarrow \sum_i c_{ij}\hat{u}_{j|i}$
    $\forall$ κάψουλα $j \in$ επίπεδο $(l+1)$: $v_j \leftarrow squash(s_j)$
    $\forall$ κάψουλα $i \in$ επίπεδο l και κάψουλα $j \in$ επίπεδο $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} \times v_j$
  end for
  $\forall$ κάψουλα $j \in$ επίπεδο $(l+1)$: $i_{indices} \leftarrow \underset{i}{\text{argmax}}(c_{ij})$
  $\forall$ κάψουλα $j \in$ επίπεδο $(l+1)$: $s_j \leftarrow c_{i=i_{indices}j}\hat{u}_{j|i=i_{indices}}$
  $\forall$ κάψουλα $j \in$ επίπεδο $(l+1)$: $v_j \leftarrow squash(s_j)$
  return v_j
end function

Αναφορικά με τον αλγόριθμο 3 πρόκειται για την απλούστερη μορφή του αλγορίθμου argmax που δεν περιλαμβάνει την κλιμάκωση των καψουλών γονέων. Αυτό έχει σαν αποτέλεσμα, το μήκος των καψουλών επιπέδου $(l+1)$ να διαμορφώνεται αποκλειστικά από τα εκπαιδευόμενα βάρη του δικτύου. Κατά αυτόν τον τρόπο και σε αντίθεση με τον προηγούμενο αλγόριθμο, το φιλτράρισμα μέσω υψηλής διαστατικότητας συμπτώσεις (high–dimensional coincidence filtering) δε διαδραματίζει κάποιο ρόλο στη διαμόρφωση του μήκους των καψουλών εξόδου. Με άλλα λόγια, αν και ο αλγόριθμος αυτός περιλαμβάνει σημαντικό μέρος του δυναμικού αλγορίθμου, τη διαμόρφωση των καψουλών εξόδου την αναλαμβάνουν ως επί το πλείστον οι πίνακες μετασχηματισμού W. Έτσι, γίνεται πιο εμφανής ο βαθμός συμβολής του αρχικού αλγορίθμου δρομολόγησης στις επιδόσεις του δικτύου.

Algorithm 3 Αλγόριθμος Argmax Routing

1: **function** ARGMAX ROUTING($\hat{u}_{j|i}, r, l$)
2: **Αρχικοποίηση:**
3: $\forall$ κάψουλα $i \in$ επίπεδο l και κάψουλα $j \in$ επίπεδο $(l+1)$: $b_{ij} \leftarrow 0$
4: **for** r επαναλήψεις **do**
5: $\forall$ κάψουλα $i \in$ επίπεδο l: $c_i \leftarrow softmax(b_i)$
6: $\forall$ κάψουλα $j \in$ επίπεδο $(l+1)$: $s_j \leftarrow \sum_i c_{ij}\hat{u}_{j|i}$
7: $\forall$ κάψουλα $j \in$ επίπεδο $(l+1)$: $v_j \leftarrow squash(s_j)$
8: $\forall$ κάψουλα $i \in$ επίπεδο l και κάψουλα $j \in$ επίπεδο $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} \times v_j$
9: **end for**
10: $\forall$ κάψουλα $j \in$ επίπεδο $(l+1)$: $i_{indices} \leftarrow \underset{i}{\text{argmax}}(c_{ij})$
11: $\forall$ κάψουλα $j \in$ επίπεδο $(l+1)$: $s_j \leftarrow \hat{u}_{j|i=i_{indices}}$
12: **return** v_j
13: **end function**

Αλγόριθμος Max Routing

Ο αλγόριθμος αυτός αποτελεί μια ακραία εκδοχή του αλγορίθμου δρομολόγησης με μηδενικό αριθμό επαναλήψεων. Μέσω αυτού, εξετάζουμε την περίπτωση όπου ο δυναμικός, επαναληπτικός αλγόριθμος δρομολόγησης (Αλγόριθμος 1) δεν έχει κανένα ρόλο στη διαμόρφωση των καψουλών επιπέδου $(l+1)$. Συγκρίνοντας τις επιδόσεις, μπορούμε να κρίνουμε την απόδοση του αργού, δυναμικού αλγορίθμου 1 σε εργασίες ταξινόμησης. Ο αλγόριθμος που εξετάζουμε, απλώς επιλέγει από τις ψήφους της κάθε κάψουλας επιπέδου $(l+1)$ την κάψουλα με το μεγαλύτερο μήκος και δρομολογεί αυτήν στο επόμενο επίπεδο.

Algorithm 4 Αλγόριθμος Max Routing

1: **function** MAX ROUTING($\hat{u}_{j|i}, l$)
2: $\forall$ κάψουλα $j \in$ επίπεδο $(l+1)$: $v_j \leftarrow max_i(\hat{u}_{j|i})$
3: **return** v_j
4: **end function**

Σημειώνουμε ότι στη γραμμή 2 του αλγορίθμου 3 το κριτήριο για την επιλογή του μεγίστου (ένα μέγιστο για κάθε j) είναι η L_2 νόρμα των διανυσμάτων $\hat{u}_{j|i}$.

4.2 Matrix Capsules with EM Routing

Η μέθοδος αυτή χρησιμοποιείται σε ορισμένα πειράματα της διπλωματικής προκειμένου να δοκιμαστεί αν ορισμένες βασικές υποθέσεις των νευρωνικών δικτύων με κάψουλες διατηρούνται στη νεότερη υλοποίηση της τεχνολογίας από την ομάδα του G. Hinton. Αν και οι δύο αρχικές μέθοδοι που περιγράφουμε δε διαφέρουν στη θεωρία που έχουμε περιγράψει, η πιο καινούρια υλοποίηση

παρουσιάζει τεχνικές βελτιώσεις με τις οποίες επιτυγχάνεται αυξημένη ακρίβεια στα πειραματικά αποτελέσματα. Οι πιο βασικές από αυτές τις βελτιώσεις είναι οι εξής:

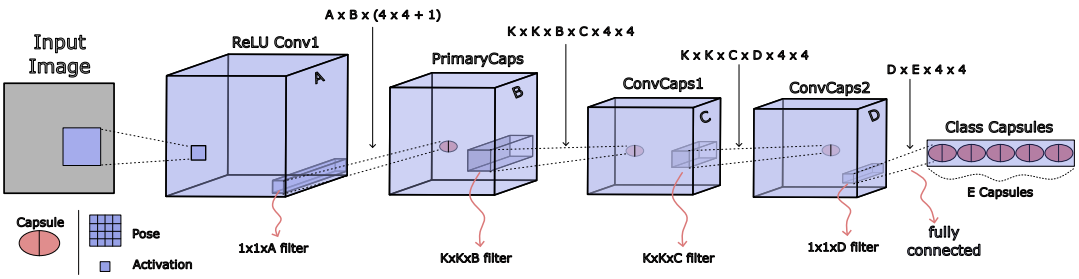
- Η αντικατάσταση του δυναμικού αλγορίθμου δρομολόγησης με τον αλγόριθμο δρομολόγησης βασισμένο στη μεγιστοποίηση προσδοκιών (EM). Κατά τη δρομολόγηση των ψήφων του ενός επιπέδου (l) στις κάψουλες του επόμενου ($l+1$), κάθε κάψουλα γονέας μοντελοποιείται σαν μια Γκαουσιανή κατανομή (με $\mu \in \Re^{16}$ και $\sigma \in \Re^{16}$) που προσαρμόζεται επαναληπτικά για να «εξηγήσει» τις ψήφους–σημεία (datapoints).

- Η αλλαγή του τρόπου αναπαράστασης της κάψουλας η οποία τώρα επιτελείται χρησιμοποιώντας έναν πίνακα 4×4 για την πόζα M_i και μια ξεχωριστή λογιστική μονάδα (logistic unit) α_i για την πιθανότητα ύπαρξης της, συνυφασμένης με την κάψουλα, οντότητας. Όπως είναι λογικό, οι ψήφοι V_{ij} και πάλι προκύπτουν ως εξής: $V_{ij} = M_i W_{ij}, W_{ij} \in \Re^{4\times4}$.

Δυστυχώς, η δημοσίευση του έργου [47] δε συνοδεύτηκε από κώδικα. Πολλοί ερευνητές επιδίωξαν να δημοσιεύσουν τη δική τους υλοποίηση αλλά τα πειραματικά αποτελέσματα ήταν πολύ πιο χαμηλά από αυτά που αναφέρονται στο σχετικό έργο. Η καλύτερη, ανοιχτού κώδικα υλοποίηση που εντοπίσαμε ακούει στο όνομα Avoiding Implementation Pitfalls of "Matrix Capsules with EM Routing by Hinton et al." [48]. Στην παρούσα διπλωματική εργασία, πειραματιζόμαστε με αυτήν την υλοποίηση ανοιχτού κώδικα (με μικρές αλλαγές) καθώς φαίνεται να είναι η πιο πιστή και επιτυχημένη υλοποίηση του δικτύου που περιγράφεται στο βασικό έργο. Μάλιστα, επιτυγχάνει ακρίβεια αποτελεσμάτων που είναι πολύ κοντά σε αυτήν που αναγράφεται στο [47]. Κυρίως όμως, πειραματιζόμαστε με την αυθεντική υλοποίηση που δημοσιεύθηκε αργότερα.

Σε αυτή την ενότητα, πρώτα θα παρουσιάσουμε την αρχιτεκτονική του νευρωνικού δικτύου με κάψουλες. Έπειτα θα παραθέσουμε τον νέο αλγόριθμο δρομολόγησης με συμφωνία μαζί με ορισμένες παρατηρήσεις. Τέλος, θα αναφερθούμε σε μερικές λεπτομέρειες υλοποίησης που αφορούν κυρίως τη διαδικασία δρομολόγησης μεταξύ διαδοχικών συνελικτικών επιπέδων από κάψουλες. Για περισσότερες λεπτομέρειες υλοποίησης, παραπέμπουμε τον αναγνώστη για άλλη μια φορά στην ιστοσελίδα όπου είναι αναρτημένος ο κώδικάς μας.

4.2.1 Αρχιτεκτονική Νευρωνικού Δικτύου



Σχήμα 4.3: Η αρχιτεκτονική του νευρωνικού δικτύου με κάψουλες της δεύτερης μεθόδου. *Παράχθηκε από το Inkscape.*

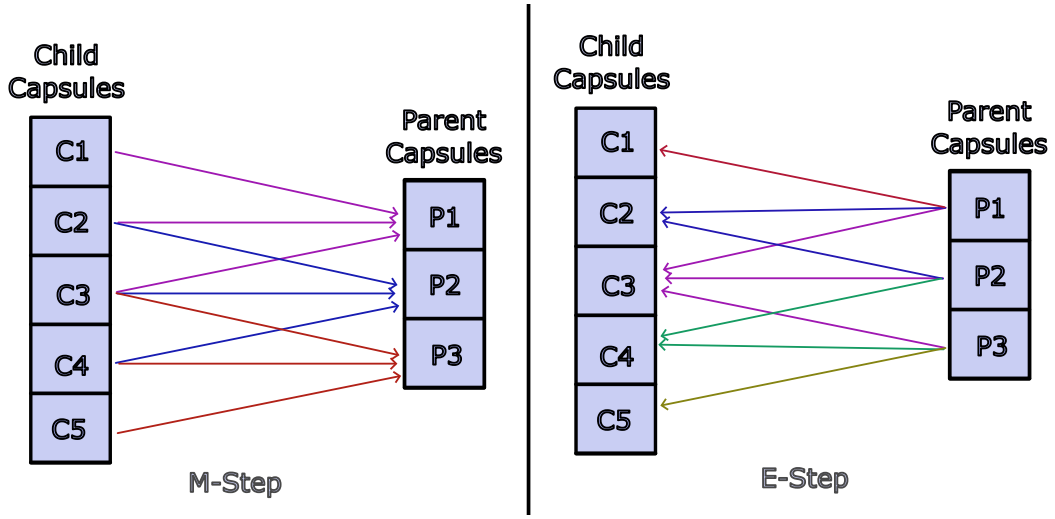
Στο σχήμα 4.3 παρατηρούμε τη γενική αρχιτεκτονική που χρησιμοποιείται σε αυτή τη μέθοδο. Κατά την περιγραφή της χρησιμοποιούμε τις μεταβλητές A, B, C, D για να αναφερθούμε σε παραμετροποιήσιμα χαρακτηριστικά του δικτύου. Όπως παρατηρούμε, όλα τα επίπεδα είναι συνελικτικά με εξαίρεση το τελευταίο, πλήρως διασυνδεδεμένο επίπεδο ενώ επίσης όλα αποτελούν επίπεδα από κάψουλες με εξαίρεση το πρώτο.

Ξεκινώντας από το πρώτο, πρόκειται για ένα απλό συνελικτικό επίπεδο με κυλιόμενο παράθυρο 5×5, βήμα 2 και μη γραμμική συνάρτηση ενεργοποίησης ReLU. Το επίπεδο αυτό απαρτίζεται από A τέτοια κυλιόμενα παράθυρα με αποτέλεσμα να παράγονται A χάρτες χαρακτηριστικών. Η διάσταση των χαρτών χαρακτηριστικών εξαρτάται από το μέγεθος της εικόνας εισόδου και μπορεί να υπολογιστεί από τις σχέσεις 2.24 και 2.25.

Συνεχίζοντας την ανάλυση του σχήματος από αριστερά προς τα δεξιά έχουμε το πρώτο επίπεδο από κάψουλες (PrimaryCaps). Το βάθος του επιπέδου αυτού είναι B. Το χαρακτηριστικό αυτό το ονομάζουμε και αριθμό των τύπων καψουλών (number of capsule types) και κάθε τύπος κάψουλας αναπαριστά (άρρητα) μια συγκεκριμένη οντότητα[3]. Το επίπεδο PrimaryCaps προκύπτει από τους προαναφερθέντες χάρτες χαρακτηριστικών έπειτα από συνέλιξη με πυρήνα 1×1 και βήμα 1. Πιο συγκεκριμένα, ο πίνακας πόζας μιας κάψουλας (PrimaryCap) προκύπτει από ένα σύνολο $A \times 4 \times 4$ εκπαιδευόμενων (learned) βαρών που επενεργούν πάνω στο σημείο του επιπέδου Conv1 που αντιστοιχεί στη συγκεκριμένη κάψουλα. Ουσιαστικά, κάθε πεδίο του 4×4 πίνακα προκύπτει ως γραμμικός συνδυασμός των A στοιχείων του επιπέδου Conv1 (ένα από κάθε κανάλι) βεβαρημένων από A εκπαιδευόμενες παραμέτρους. Η τιμή ενεργοποίησης της κάψουλας προκύπτει με παρόμοιο τρόπο (δηλαδή αποτελεί το γραμμικό συνδυασμό των ίδιων A στοιχείων του Conv1 με τα A βάρη) αλλά αυτή τη φορά, στο αποτέλεσμα εφαρμόζεται η σιγμοειδής συνάρτηση (ώστε το αποτέλεσμα να ανήκει στο διάστημα [0,1]). Σημειώνουμε ότι για κάθε τύπο κάψουλας (δηλαδή για καθένα από τα B κανάλια από κάψουλες) τα βάρη $A \times (4 \times 4 + 1)$ διαμοιράζονται.

Στη συνέχεια ακολουθούν τα επίπεδα ConvCaps1 και ConvCaps2. Πρόκειται για δύο συνελικτικά επίπεδα από κάψουλες τα οποία έχουν C και D τύπους από κάψουλες, πυρήνες 3×3 και βήμα 2 και 1 αντίστοιχα. Ο σχηματισμός όμως των ConvCaps1 και ConvCaps2 δεν είναι προφανής. Πιο αναλυτικά, για τον σχηματισμό του καθενός τέτοιου επιπέδου λαμβάνει χώρα ο αλγόριθμος δρομολόγησης με συμφωνία βασιζόμενος στον EM που θα αναλύσουμε παρακάτω. Στο σημείο αυτό αξίζει να σημειωθεί ότι αντίθετα με δύο πλήρως διασυνδεδεμένα επίπεδα από κάψουλες, σε δύο συνελικτικά επίπεδα από κάψουλες οι γονείς (κάψουλες του επιπέδου $L + 1$) ανταγωνίζονται να διεκδικήσουν τις ψήφους των παιδιών (κάψουλες επιπέδου L) που ανήκουν σε μια γειτονιά ($K \times K = 3 \times 3$) του επιπέδου L (κεντραρισμένη στην κάψουλα γονέα). Κατά αυτόν τον τρόπο, η εκάστοτε κάψουλα γονέας του επιπέδου $L + 1$ στέλνει ανατροφοδότηση (σύμφωνα με τον αλγόριθμο δρομολόγησης) μόνο στα παιδιά που ανήκουν στο οπτικό του πεδίο (βλ. σχήμα 4.4). Αντίστοιχα, τα παιδιά δεν παράγουν ψήφους για όλες τις κάψουλες του επόμενου επιπέδου αλλά μόνο για αυτές στων οποίων το οπτικό πεδίο ανήκουν. Επειδή μάλιστα οι πίνακες μετασχηματισμού W_{ij} διαμοιράζονται στις χωρικές διαστάσεις $x - y$, για τον σχηματισμό του επιπέδου ConvCaps1 απαιτούνται $K \times K \times B \times C \times 4 \times 4$ εκπαιδευόμενα (learned) βάρη (ανάλογα για το ConvCaps2 απαιτούνται $K \times K \times C \times D \times 4 \times 4$). Αυτό το νούμερο προκύπτει λογικά αν

[3]Κάψουλες ίδιου τύπου μαθαίνουν να αναγνωρίζουν την ίδια οντότητα αλλά σε διαφορετικές περιοχές της εικόνας.



Σχήμα 4.4: Στο σχήμα παρατηρούμε τη διασύνδεση μεταξύ των καψουλών δύο διαδοχικών επιπέδων κατά τις φάσεις M και E του αλγορίθμου δρομολόγησης. Για λόγους απλότητας, απεικονίζεται η περίπτωση όπου έχουμε μονοδιάστατη συνέλιξη (1D convolution). *Παράχθηκε από το Inkscape.*

αναλογιστεί κανείς ότι:

- Ο κάθε πίνακας μετασχηματισμού έχει διάσταση 4×4.
- Σε κάθε σημείο i του πυρήνα (kernel) αντιστοιχεί ξεχωριστός πίνακας βαρών Wij, άρα έχουμε $K \times K$ τέτοιους πίνακες μεγέθους 4×4.
- Το κυλιόμενο παράθυρο $K \times K$ έχει βάθος B, όσος και ο αριθμός των τύπων καψουλών εισόδου. Άρα είναι $B \times K \times K \times 4 \times 4$ βάρη.
- Θέλουμε C κανάλια εξόδου (ή αλλιώς τύπους από κάψουλες). Συνεπώς, έχουμε C κυλιόμενα παράθυρα.

Το τελευταίο επίπεδο είναι το Class Capsules όπου κάθε κάψουλά του αντιστοιχεί σε μια κλάση. Η κάψουλα με τη μεγαλύτερη τιμή ενεργοποίησης (logistic unit) σηματοδοτεί την κλάση πρόβλεψης του νευρωνικού δικτύου όταν αυτό τροφοδοτείται με μια εικόνα ενός αντικειμένου. Τα περιεχόμενα των καψουλών στο τελευταίο επίπεδο σχηματίζονται με τον αλγόριθμο δρομολόγησης που παρουσιάζουμε παρακάτω. Επειδή τα δύο τελευταία επίπεδα είναι πλήρως διασυνδεδεμένα μεταξύ τους, κάθε κάψουλα του επιπέδου ConvCaps2 παράγει μια ψήφο για κάθε κάψουλα του επιπέδου Class Capsules. Έτσι απαιτούνται $D \times E$ πίνακες μετασχηματισμού μεγέθους 4×4. Για να μη χαθεί η πληροφορία της θέσης (στον $x-y$ άξονα) στην οποία εντοπίζεται το κάθε αντικείμενο–μέρος, μετά τον πολλαπλασιασμό της πόζας της κάθε κάψουλας με το κατάλληλο πίνακα μετασχηματισμού, στα πρώτα δύο κελιά της δεξιότερης στήλης του πίνακα ψήφου υπερτίθενται η γραμμή και η στήλη της κάψουλας από την οποία προήλθε (η ψήφος).

Όπως εξηγήσαμε, ακολουθούμε μια ελαφρώς τροποποιημένη έκδοση της αρχιτεκτονικής που παρουσιάζεται στο [47] (βλ. πίνακα 4.1). Η διαφορά έγκειται στο ότι η μη αυθεντική υλοποίηση που χρησιμοποιούμε (ακολουθώντας τις παρατηρήσεις και τον κώδικα του [48]) έχει συνήθως μικρότερο βάθος επιπέδων, χωρίς αυτό να είναι ο καθοριστικός παράγοντας της ελάχιστα χαμη-

λότερης απόδοσης που παρατηρείται όταν αντιπαραβάλλεται με τα αποτελέσματα της αυθεντικής υλοποίησης.

Παράμετρος	Αυθεντική Υλοποίηση	Απλοποιημένη
A	32	64
B	32	8
C	32	16
D	32	16

Πίνακας 4.1: Πίνακας στον οποίο παρουσιάζεται η παραμετροποίηση της αρχιτεκτονικής όπως παρουσιάζεται στο έργο [47] (αυθεντική υλοποίηση) και όπως παρουσιάζεται στο [48] (απλοποιημένη). Η παράμετρος E είναι πάντα ίση με τον αριθμό των κλάσεων εξόδου.

Κλείνοντας την παράγραφο αυτή, να επισημάνουμε ότι για λόγους σύγκρισης με το έργο [76] υπάρχει η επιλογή να χρησιμοποιηθεί η αρχιτεκτονική που παρουσιάσαμε στην εικόνα 4.1 μέχρι και το επίπεδο των Primary Capsules.

4.2.2 Υπολογισμός Τιμής Ενεργοποίησης

Κατά αντιστοιχία με την προηγούμενη μέθοδο, οι κάψουλες παιδιά ψηφίζουν για το ποια εκτιμούν ότι είναι η πόζα του κάθε γονέα που «βλέπουν». Σε αυτούς τους γονείς δρομολογούν τις ψήφους με βαρύτητα που υποδεικνύεται από τις πιθανότητες ανάθεσης (assignment probabilities) - συμβολίζονται με R_{ij} και διαμορφώνονται δυναμικά από τον νέο αλγόριθμο δρομολόγησης. Προτού περιγράψουμε το νέο αλγόριθμο δρομολόγησης, κρίνεται σκόπιμο να εξηγήσουμε τον μηχανισμό με τον οποίο εκτιμάται αν θα ενεργοποιηθεί μια κάψουλα γονέας ή όχι (τιμή α_j). Η λογική που ακολουθείται για τον σκοπό αυτό είναι εμπνευσμένη από την αρχή του ελαχίστου μήκους περιγραφής (minimum description length principle).

Πιο αναλυτικά, μέσα από τον αλγόριθμο δρομολόγησης που θα παρουσιάσουμε παρακάτω επιλέγεται κατά πόσον θα ενεργοποιηθεί ή όχι μια κάψουλα γονέας ανάλογα με το ποια κατάσταση ελαχιστοποιεί το κόστος περιγραφής[4]. Για τον σχηματισμό του, λαμβάνονται υπόψη τα εξής:

- Στην περίπτωση που δεν ενεργοποιηθεί μια κάψουλα γονέας, το κόστος προκύπτει από την τιμή $-\beta_u$ που πρέπει να «πληρώσουμε» για κάθε κάψουλα–παιδί που αναθέτεται στον συγκεκριμένο γονέα (έστω j). Στη συνήθη περίπτωση που υπάρχει μερική ανάθεση μεταξύ παιδιού–πατέρα ($R_{ij} \neq 1$), στο κόστος προστίθεται η ποσότητα $-\beta_u$ βεβαρημένη με το κλάσμα ανάθεσης R_{ij}. Συνεπώς, το κόστος για μια πλήρως απενεργοποιημένη κάψουλα j θα είναι ίσο με:
$$-\beta_u \sum_i R_{ij}, \text{ όπου } i \in FoV(j) \tag{4.8}$$

- Στην περίπτωση ενεργοποίησης της κάψουλας γονέα j τότε «πληρώνουμε» πρώτα ένα κόστος $-\beta_\alpha$ που οφείλεται στην κωδικοποίηση των παραμέτρων που σχετίζονται με την κάψουλα γονέα. Επίσης, στο κόστος αυτό προστίθενται τα κόστη περιγραφής των διαφορών μεταξύ των ψήφων και της πόζας της κάψουλας j στην οποία αναθέτονται (ζυγισμένα

[4]Φυσικά, η τιμή ενεργοποίησης a_j λαμβάνει τιμές στο διάστημα [0,1] οπότε τυπικά υπάρχουν άπειρες ενδιάμεσες καταστάσεις.

υπό τις πιθανότητες ανάθεσης). Η τελευταία ποσότητα μπορεί να προσεγγιστεί ως εξής:

$$cost_j = \sum_h cost_j^h = \sum_h \sum_i -R_{ij} ln(P_{i|j}^h), \tag{4.9}$$

όπου ο δείκτης h αναφέρεται στην h-οστή διάσταση του διανύσματος βάσης. Προς επεξήγηση της ανωτέρω σχέσης, να αναφέρουμε ότι ουσιαστικά, η ποσότητα $-R_{ij} ln(P_{i|j}^h)$ περιγράφει το κόστος της διαφοράς της ψήφου i και της πόζας της κάψουλας j για τη διάσταση $h \in [1, 4 \times 4]$. Υπενθυμίζουμε, στο σημείο αυτό ότι οι ψήφοι $V_{ij} \in \Re^{4\times 4}$ μοντελοποιούνται σαν σημεία στον χώρο $\Re^1 6$ και οι κάψουλες M_j σαν Γκαουσιανές κατανομές με μέση τιμή $\mu_j \in \Re^{16}$ και διαγώνιο πίνακα συνδιακύμανσης Σ με στοιχεία διαγωνίου που συγκροτούν το διάνυσμα $\sigma \in \Re^{16}$. Έτσι, μπορούμε να προσεγγίσουμε το κόστος για την περιγραφή ενός σημείου (ψήφου) V_{ij} από την κατανομή (κάψουλα γονέας) j ως την αρνητική, λογαριθμική, πολυμεταβλητή, γκαουσιανή πυκνότητα πιθανότητας (negative log multivariate gaussian probability density) παραμετροποιημένη από τα μ_j και σ_j της κάψουλας j υπολογισμένη στο σημείο της ψήφου V_{ij} από την κάψουλα i (φυσικά το κόστος θα είναι ζυγισμένο από την πιθανότητα ανάθεσης R_{ij}). Έτσι με μαθηματικούς όρους, η γκαουσιανή Σ.Π.Π. (PDF) υπολογισμένη στο σημείο V_{ij} για τη διάσταση h είναι:

$$P_{i|j}^h = \frac{1}{\sqrt{2\pi(\sigma_j^h)^2}} exp(-\frac{(V_{ij}^h - \mu_j^h)^2}{2(\sigma_j^h)^2}). \tag{4.10}$$

Έτσι, ο αρνητικός λογάριθμος της ανωτέρω ποσότητας είναι:

$$-ln(P_{i|j}^h) = \frac{(V_{ij}^h - \mu_j^h)^2}{2(\sigma_j^h)^2} + ln(\sigma_j^h) + \frac{ln(2\pi)}{2}. \tag{4.11}$$

Επανερχόμενοι στο συνολικό κόστος για τη διάσταση h, έχουμε:

$$\begin{aligned} cost_j^h &= \sum_i -R_{ij} ln(P_{i|j}^h) && (4.12)\\ &= \frac{\sum_i R_{ij}(V_{ij}^h - \mu_j^h)^2}{2(\sigma_j^h)^2} + (ln(\sigma_j^h) + \frac{ln(2\pi)}{2}) \sum_i R_{ij} && (4.13)\\ &= \frac{\sum_i R_{ij}(\sigma_j^h)^2}{2(\sigma_j^h)^2} + (ln(\sigma_j^h) + \frac{ln(2\pi)}{2}) \sum_i R_{ij} && (4.14)\\ &= (ln(\sigma_j^h) + \frac{1}{2} + \frac{ln(2\pi)}{2}) \sum_i R_{ij} && (4.15) \end{aligned}$$

Συνολικά λοιπόν, η συνάρτηση ενεργοποίησης της κάψουλας j προκύπτει από την αντιπαραβολή του κόστους ενεργοποίησης και του κόστους απενεργοποίησής της. Τον ρόλο αυτό τον αναλαμβάνει η σιγμοειδής συνάρτηση (sigmoid function). Συνεπώς, έχουμε:

$$\alpha_j = sigmoid(\lambda(\beta_\alpha - \beta_u \sum_i R_{ij} - \sum_h cost_j^h)). \tag{4.16}$$

Το λ είναι μια παράμετρος ανάστροφης θερμοκρασίας. Στην αρχή η παράμετρος είναι ίση με τη μονάδα και αυξάνεται σε κάθε επανάληψη του αλγορίθμου δρομολόγησης κάνοντας την κλίση της λογιστικής συνάρτησης μεγαλύτερη και ωθώντας τις τιμές ενεργοποίησης να λάβουν πιο ακραίες τιμές (0 ή 1). Αν ενσωματώσουμε στο $cost_j^h$ τον όρο $\beta_u \sum_i R_{ij}$ τότε έχουμε:

$$\alpha_j = sigmoid(\lambda(\beta_\alpha - \sum_h \acute{cost}_j^h)) \tag{4.17}$$

όπου τώρα το κόστος είναι:

$$\acute{cost}_j^h = (\beta_u + ln(\sigma_j^h) + const) \sum_i R_{ij}. \tag{4.18}$$

Κλείνοντας την υποενότητα αυτή, να σημειώσουμε ότι οι ποσότητες β_u και β_α μαθαίνονται κατά τη διάρκεια εκπαίδευσης του αλγορίθμου. Αυτός είναι και ένας λόγος για τον οποίο ο σταθερός όρος στη σχέση 4.18 μπορεί να παραληφθεί.

4.2.3 Αλγόριθμος Δρομολόγησης EM

Όπως έχουμε προαναφέρει, μεταξύ δύο διαδοχικών επιπέδων από κάψουλες λαμβάνει χώρα ο αλγόριθμος δρομολόγησης βασισμένος στη Μεγιστοποίηση Προσδοκιών (Expectation Maximization) μέσω του οποίου οι τιμές ενεργοποίησης και οι πόζες των καψουλών γονέων υπολογίζονται επαναληπτικά. Κατά αντιστοιχία με τον αυθεντικό αλγόριθμο EM, η επαναληπτική διαδικασία αποτελείται από δύο βήματα τα οποία εναλλάσσονται διαδεχόμενα το ένα το άλλο. Στο βήμα Ε υπολογίζονται οι πιθανότητες ανάθεσης R_{ij} για κάθε ζευγάρι από κάψουλες μεταξύ των δύο επιπέδων. Ο υπολογισμός γίνεται χρησιμοποιώντας τις μέσες τιμές, τις διακυμάνσεις και τις τιμές ενεργοποίησης των καψουλών γονέων. Στο βήμα Μ υπολογίζονται οι παράμετροι των Γκαουσιανών κατανομών (που μοντελοποιούν τις κάψουλες γονείς) με βάση τα ανανεωμένα R_{ij}. Έτσι, μετά από ορισμένο αριθμό επαναλήψεων, ο αλγόριθμος συγκλίνει με αποτέλεσμα οι ενεργές κάψουλες γονείς να λαμβάνουν και να περιγράφουν συστάδες από όμοιες ψήφους παιδιών.

Μια περισσότερο τυπική περιγραφή του αλγορίθμου δρομολόγησης παρατίθεται παρακάτω. Σημειώνουμε ότι με Ω_L συμβολίζουμε το σύνολο όλων των καψουλών επιπέδου L. Επίσης, για τη διάσταση H του πίνακα πόζας όταν αναδιατάσσεται σε διάνυσμα ισχύει ότι $H = 16$.

Algorithm 5 Αλγόριθμος Δρομολόγησης Βασισμένος στον EM

procedure EM ROUTING(α, V)
  Αρχικοποίηση:
    $\forall\, i \in \Omega_L,\, j \in \Omega_{L+1} : R_{ij} \leftarrow 1/\left|\Omega_{L+1}\right|$
  for t επαναλήψεις **do**
    $\forall j \in \Omega_{L+1} : M - Step(\alpha, R, V, j)$
    $\forall i \in \Omega_L : E - Step(\mu, \sigma, \alpha, V, i)$
  end for
end procedure
procedure M-STEP(α, R, V, j) ▷ Για μια κάψουλα υψηλότερου επιπέδου, j
  $\forall i \in \Omega_L : R_{ij} \leftarrow R_{ij} * \alpha_i$
  $\forall h : \mu_j^h \leftarrow \frac{\sum_i R_{ij} V_{ij}^h}{\sum_i R_{ij}}$
  $\forall h : (\sigma_j^h)^2 \leftarrow \frac{\sum_i R_{ij} (V_{ij}^h - \mu_j^h)^2}{\sum_i R_{ij}}$
  $\forall h : \acute{cost}^h \leftarrow (\beta_u + log(\sigma_j^h)) \sum_i R_{ij}$
  $\alpha_j \leftarrow logistic(\lambda(\beta_\alpha - \sum_h \acute{cost}^h))$
end procedure
procedure E-STEP($\mu, \sigma, \alpha, V, i$) ▷ Για μια κάψουλα χαμηλότερου επιπέδου, i
  $\forall j \in \Omega_{L+1} : p_j \leftarrow \frac{1}{\sqrt{\prod_h^H 2\pi(\sigma_j^h)^2}} \exp(-\sum_h^H \frac{(V_{ij}^h - \mu_j^h)^2}{2(\sigma_j^h)^2})$
  $\forall j \in \Omega_{L+1} : R_{ij} \leftarrow \frac{\alpha_j p_j}{\sum_{k \in \Omega_{L+1}} \alpha_k p_k}$
end procedure

Για τον αλγόριθμο δρομολόγησης βασισμένο στον EM μπορούμε να κάνουμε τις εξής σημειώσεις:

- Αρχικοποιούμε με ομοιόμορφο τρόπο τις πιθανότητες ανάθεσης R_{ij} ώστε κάθε παιδί να συνδέεται ισοδύναμα με τους γονείς που ψηφίζει. Έπειτα, καλούμε (επαναληπτικά) το βήμα M και το βήμα E.

- Στο βήμα M υπολογίζουμε τα μ_j τα σ_j και τα α_j βασιζόμενοι στα R_{ij}, V_{ij} και εμμέσως, στις πιθανότητες ενεργοποίησης των παιδιών α_i. Ουσιαστικά, στο βήμα αυτό υπολογίζεται ένα βελτιωμένο Γκαουσιανό μοντέλο για την κάθε κάψουλα, μαζί με την πιθανότητα ανάθεσής της.

- Στο βήμα E υπολογίζονται οι πιθανότητες μέλους (membership probabilities) p_j που δείχνουν την πιθανότητα το «δείγμα» i να ανήκει στην γκαουσιανή j. Επίσης, επανεκτιμούνται οι πιθανότητες ανάθεσης R_{ij}. Οι εν λόγω υπολογισμοί γίνονται χρησιμοποιώντας τις νέες κατανομές που προέκυψαν από το βήμα M.

- Στο βήμα 13 του αλγορίθμου (σχέση 4.18) υπολογίζουμε και κλιμακώνουμε το κόστος ανάλογα με τη μέση ποσότητα δεδομένων που δέχονται οι κάψουλες γονείς στο εκάστοτε

επίπεδο. Η ποσότητα αυτή υπολογίζεται από τον τύπο:

$$mean\_data = \frac{child_W \times child_H \times child_{CH}}{parent_W \times parent_H \times parent_{CH}}, \tag{4.19}$$

όπου W, H και CH δηλώνουν το πλάτος, το ύψος και το βάθος (αριθμός από τύπους) του επιπέδου από κάψουλες. [5]

- Μετά από δύο ή τρεις επαναλήψεις συνήθως, ο αλγόριθμος τερματίζει όπου και αναδιατάσσουμε τα διανύσματα μ_j σε μορφή πίνακα 4×4 ώστε να λάβουμε τις πόζες M_j των καψουλών γονέων.

4.2.4 Συνάρτηση Σφάλματος και Λοιπά Στοιχεία Υλοποίησης

Για τους σκοπούς της εκπαίδευσης του νευρωνικού δικτύου με κάψουλες χρησιμοποιείται η απώλεια διασποράς (spread loss). Η συνάρτηση αυτή δίνεται από τη σχέση:

$$L = \sum_{i \neq t} L_i, \text{όπου} L_i = (max(0, m - (\alpha_t - \alpha_i)))^2. \tag{4.20}$$

Στην ανωτέρω σχέση το α_i είναι η πιθανότητα ενεργοποίησης της κάψουλας του τελευταίου επιπέδου με αύξοντα αριθμό i. α_t είναι η τιμή της πιθανότητας ενεργοποίησης της κάψουλας που έχει αύξοντα αριθμό ίδιο με αυτόν της κλάσης στόχου. Στα αρχικά παραδείγματα της εκπαίδευσης, το «περιθώριο» m [6] είναι μικρό (0.2) έτσι ώστε να αποφεύγεται ο σχηματισμός μόνιμα ανενεργών καψουλών. Καθώς η εκπαίδευση συνεχίζεται, το περιθώριο αυξάνεται σταδιακά στην τιμή 0.9.

Τέλος, σημειώνουμε ότι ακολουθώντας την περιγραφή της υλοποίησης στο [48], επιβάλουμε ένα κάτω φράγμα στη διασπορά $(\sigma_j^h)^2$ προσθέτοντας την τιμή $\epsilon = 10^{-4}$ έτσι ώστε να μην παρατηρείται έντονα το πρόβλημα της «σύνθλιψης της διακύμανσης» (variance collapse).

4.3 Multi-Head Self-Attention Capsules

Σε αυτήν την ενότητα, θα αναλύσουμε μια ομάδα μεθόδων που χρησιμοποιούν μηχανισμό αυτο–προσοχής για τη δρομολόγηση των καψουλών μεταξύ δυο διαδοχικών επιπέδων, οδηγώντας σε μια γρήγορη και κλιμακώσιμη υλοποίηση των νευρωνικών δικτύων με κάψουλες. Οι μέθοδοι αυτές, εμπνευσμένες από το έργο [49] και χρησιμοποιώντας ελάχιστες παραμέτρους σε σχέση με τη βασική υλοποίηση [76], επιτυγχάνουν υψηλά ποσοστά ακρίβειας.

Πιο αναλυτικά, αρχικά θα αναφερθούμε στη βασική αρχιτεκτονική του νευρωνικού δικτύου που αναπτύξαμε, περιγράφοντάς τη με μεταβλητές που αφορούν τα παραμετροποιήσημα μεγέθη. Σε αυτό το πλαίσιο, θα αναφερθούμε και σε μια εναλλακτική αρχιτεκτονική που αντικαθιστά τα πρώτα συνελικιτκά επίπεδα του νευρωνικού δικτύου με αυτό που χρησιμοποιείται στο έργο [76]. Έπειτα, θα παρουσιάσουμε τον αλγόριθμο που χρησιμοποιείται στο έργο [49] αλλά και όλες τις

[5] Η κλιμάκωση μπορεί να γίνει με διαίρεση του κόστους με τη μέση ποσότητα δεδομένων. Για περισσότερες πληροφορίες, παραπέμπουμε τον αναγνώστη στο [48].

[6] Το περιθώριο συμβολίζει τη μέγιστη διαφορά που μπορούν να έχουν η πιθανότητα ενεργοποίησης της κάψουλας που αναφέρεται στη σωστή κλάση και της αντίστοιχης τιμής της κάψουλας $i \neq t$ και να μην προσμετρηθεί στο σφάλμα.

παραλλαγές του που αναπτύξαμε εμείς όπως αυτή της πολυκέφαλης προσοχής. Στη συνέχεια, θα γίνει λόγος για το τμήμα του αποκωδικοποιητή (ή τμήμα ανακατασκευής) και δη τις δύο εναλλακτικές αρχιτεκτονικές που χρησιμοποιήσαμε στα πειράματα. Τέλος, θα διατυπώσουμε ορισμένες λεπτομέρειες υλοποίησης (συνάρτηση σύνθλιψης, συνάρτηση σφάλματος κτλ.) που θα διευκολύνουν τον αναγνώστη που επιθυμεί να μελετήσει τον σχετικό μας κώδικα.

Οι προσφορές της παρούσας μεθόδου μπορούν να συνοψιστούν ως εξής:

- Ο πειραματισμός με μια απλή αρχιτεκτονική που μειώνει σημαντικά τον αριθμό των παραμέτρων χωρίς να θυσιάζεται η επίδοση του δικτύου.
- Ο πειραματισμός με πιο σύνθετους αποκωδικοποιητές χρησιμοποιώντας επίπεδα αποσυνέλιξης (deconvolution) που βελτιώνουν τη δυνατότητα γενίκευσης του δικτύου με κάψουλες.
- Η παρουσίαση ενός πρωτοπόρου, μη–επαναληπτικού αλγορίθμου δρομολόγησης καψουλών με μηχανισμό αυτο–προσοχής πολλών κεφαλών που βελτιώνει την απόδοση.
- Η περιγραφή ενός αλγορίθμου που μπορεί να χρησιμοποιηθεί συμπληρωματικά με κάθε αλγόριθμο δρομολόγησης για την αποδοτική αρχικοποίηση των καψουλών γονέων (ελαττώνοντας έτσι τον αριθμό των αργών επαναλήψεων ενός δυναμικού, επαναληπτικού αλγορίθμου).
- Ο έλεγχος της ποιότητας γενίκευσης νευρωνικών δικτύων με κάψουλες που χρησιμοποιούν μη επαναληπτικούς αλγορίθμους δρομολόγησης με δοκιμές (της μεθόδου μας) σε σύνολα δεδομένων όπως το MultiMNIST και το SmallNorb αλλά και με ελέγχους διαταραχής (perturbation testing).

4.3.1 Αρχιτεκτονική Νευρωνικού Δικτύου

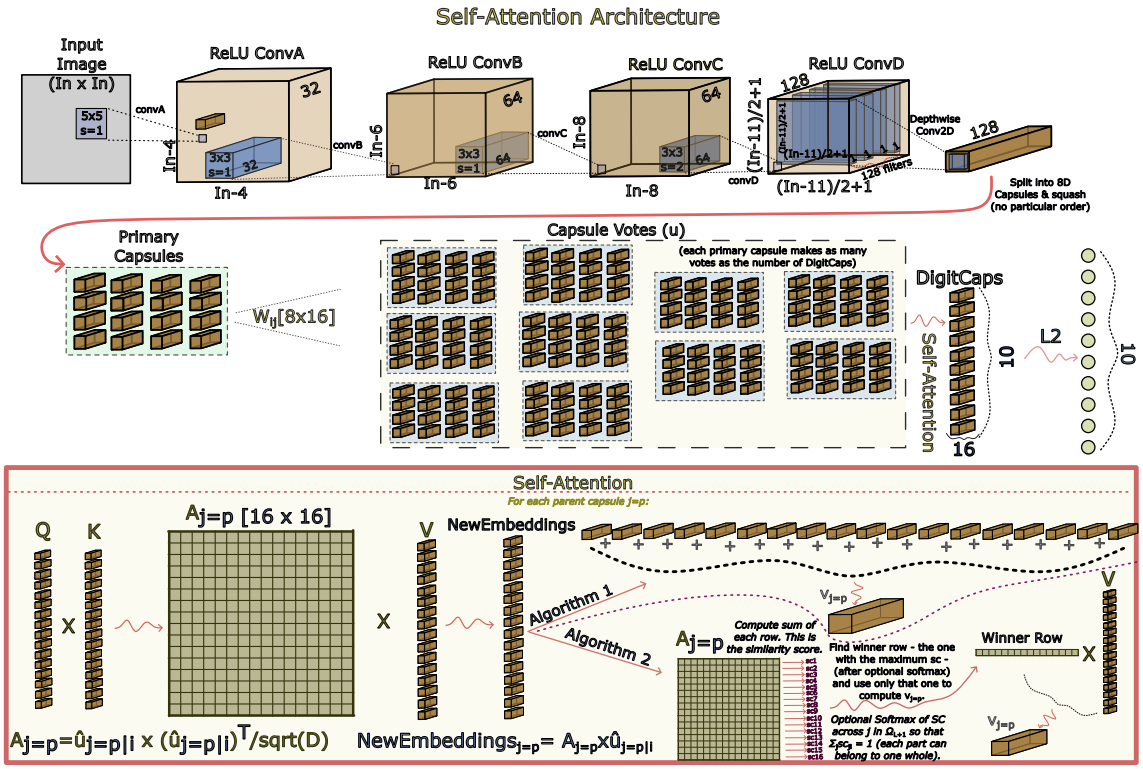
Η βασική αρχιτεκτονική του δικτύου φαίνεται στο σχήμα 4.5 (την ονομάζουμε αρχιτεκτονική DepthConv). Για τη μεταφορά από τον χώρο των εικονοστοιχείων στον χώρο των καψουλών χρησιμοποιούνται τέσσερα συνελικτικά επίπεδα και ένα επίπεδο συνέλιξης κατά βάθος (Depthwise 2D Convolution). Οι συνήθεις παράμετροι των πρώτων συνελικτικών επιπέδων παρουσιάζονται στον πίνακα 4.2. Επισημαίνεται ότι μετά από κάθε συνελικτικό επίπεδο έχουμε ένα επίπεδο κανονικοποίησης κατά δέσμες (Batch Normalization)[7].

Επίπεδο	Πυρήνας	Αριθμός Φίλτρων
A	5×5	32
B	3×3	64
C	3×3	64
D	3×3	128

Πίνακας 4.2: Πίνακας στον οποίο παρουσιάζεται η παραμετροποίηση της αρχιτεκτονικής των πρώτων επιπέδων.

Όπως προαναφέραμε, τα πρώτα τέσσερα επίπεδα διαδέχεται ένα επίπεδο συνέλιξης κατά βάθος. Ουσιαστικά, πρόκειται για ένα επίπεδο συνέλιξης με τόσα κυλιόμενα παράθυρα (δισδιάστατα φίλ-

[7]Εκτός από την αρχιτεκτονική για το σύνολο δεδομένων SmallNORB όπου τα πρώτα τρία είναι επίπεδα κανονικοποίησης κατά κανάλι παραδείγματος (Instance Normalization)



Σχήμα 4.5: Η αρχιτεκτονική του νευρωνικού δικτύου με κάψουλες της τρίτης μεθόδου (αρχιτεκτονική DepthConv). Στο σχήμα περιγράφονται σχηματικά οι δύο εναλλακτικοί αλγόριθμοι δρομολόγησης που αναπτύξαμε. Για λόγους απλότητας, δεν απεικονίζουμε την περίπτωση αυτο–προσοχής πολλών κεφαλών. Επίσης, για λόγους κατανόησης, δεν αναπαριστώνται ορισμένες αλγοριθμικές λεπτομέρειες όπως αυτή της κανονικοποίησης. Σημειώνεται ότι τα V, Q, K είναι ταυτόσημα με τη μήτρα V. *Παράχθηκε από το Inkscape.*

τρα) όση είναι και η διάσταση βάθους του προηγούμενου επιπέδου. Το καθένα από αυτά τα φίλτρα παράγει ένα χάρτη χαρακτηριστικών. Αξίζει να σημειωθεί ότι στην περίπτωσή μας, ακολουθώντας την υλοποίηση [49], το μέγεθος του κάθε δισδιάστατου πυρήνα το θέτουμε να είναι ίσο με τις διαστάσεις πλάτους και ύψους των χαρτών χαρακτηριστικών του προηγούμενου επιπέδου. Συνεπώς, από κάθε φίλτρο προκύπτει μια μονάδα (scalar). Συνολικά, δηλαδή, έχουμε σαν έξοδο έναν όγκο από αποκρίσεις (activations) μεγέθους $[1 \times 1 \times D_F]$, όπου D_F ο αριθμός των φίλτρων του επιπέδου D. Το διάνυσμα αυτό το διαχωρίζουμε σε υποδιανύσματα με οκτώ στοιχεία το καθένα τα οποία και τροφοδοτούμε στη συνάρτηση σύνθλιψης (squash) σχηματίζοντας έτσι το πρώτο επίπεδο από κάψουλες (PrimaryCapsules).

Για παράδειγμα, στις περισσότερες παραμετροποιήσεις των πειραμάτων το συνελικτικό επίπεδο D έχει σαν αποτέλεσμα τη δημιουργία 128 χαρτών χαρακτηριστικών μεγέθους 9×9. Τροφοδοτώντας αυτές τις αποκρίσεις σε ένα συνελικτικό επίπεδο κατά βάθος με 128 φίλτρα μεγέθους 9×9 λαμβάνουμε ένα διάνυσμα 128 στοιχείων. Αυτό το διάνυσμα το χωρίζουμε σε κάψουλες διάστασης 8 με αποτέλεσμα να λάβουμε 16 PrimaryCaps (αφού πρώτα εφαρμόσουμε σε αυτές την τροποποιημένη συνάρτηση σύνθλιψης που εξηγούμε παρακάτω).

Η εν λόγω αρχιτεκτονική ενσωματώνει πολύ περισσότερα συνελικτικά επίπεδα από την αρχιτεκτονική που προτείνεται στο έργο [76]. Παρόλα αυτά, μειώνοντας τον αριθμό των καψουλών που βρίσκονται στο πρώτο επίπεδο από κάψουλες, επιτυγχάνει να ελαττώσει σημαντικά τον α-

ριθμό των εκπαιδευόμενων παραμέτρων. Για λόγους πειραματισμού, δοκιμάσαμε να μιμηθούμε τον τρόπο δημιουργίας του επιπέδου PrimaryCaps που χρησιμοποιείται στο έργο [76], όπως τον περιγράψαμε στην πρώτη μέθοδο του παρόντος κεφαλαίου. Έτσι, στη θέση των τεσσάρων συνελικτικών επιπέδων έχουμε ένα επίπεδο από 256 φίλτρα με πυρήνες 9×9 (βλέπε σχήμα 4.1). Ειδικά για το σύνολο δεδομένων SmallNORB, η εναλλακτική αρχιτεκτονική για τη δημιουργία του επιπέδου PrimaryCaps είναι ίδια με αυτή του έργου [47] (βλέπε σχήμα 4.3). Θα αναφερόμαστε σε αυτήν την εναλλακτική αρχιτεκτονική ως «αρχιτεκτονική DynamLike» σε αντιπαραβολή με την «αρχιτεκτονική DepthConv» που απεικονίζεται στο σχήμα 4.5.

Συνεχίζοντας την περιγραφή της αρχιτεκτονικής από τα αριστερά προς τα δεξιά, μέσω του αλγορίθμου δρομολόγησης με μηχανισμό αυτο–προσοχής διαμορφώνονται οι κάψουλες του τελευταίου επιπέδου (ονομάζονται και DigitCaps). Αναπόσπαστο στοιχείο της διαδικασίας σχηματισμού των καψουλών του ανώτερου επιπέδου είναι ο υπολογισμός των ψήφων για την πόζα των DigitCaps. Κατά αναλογία με την υλοποίηση [76], η κάθε κάψουλα επιπέδου PrimaryCaps χρησιμοποιώντας πίνακες μετασχηματισμού με εκπαιδευόμενα βάρη παράγει μια πρόβλεψη για κάθε κάψουλα επιπέδου DigitCaps. Αυτές οι ψήφοι στη συνέχεια φιλτράρονται μέσω ενός γρήγορου αλγορίθμου δρομολόγησης με αυτο–προσοχή (αναλύεται παρακάτω) και σχηματίζονται οι κάψουλες του τελικού επιπέδου.

4.3.2 Αλγόριθμοι Δρομολόγησης

Μια σημαντική αδυναμία των νευρωνικών δικτύων με κάψουλες που διαπιστώθηκε κατά τη μελέτη σχετικής βιβλιογραφίας είναι αυτή της κλιμάκωσης σε μεγαλύτερες διατάξεις. Τροχοπέδη για την κλιμάκωση αποτελεί το αυξημένο υπολογιστικό κόστος και η αστάθεια των επαναληπτικών αλγορίθμων δρομολόγησης. Επιδιώκοντας να δοθεί λύση στο πρόβλημα, παρατηρήσαμε την πρόσφατη άνθιση των μοντέλων νευρωνικών δικτύων που χρησιμοποιούν μηχανισμό αυτο–προσοχής (υπό τη μορφή μετασχηματιστών) τόσο για εφαρμογές επεξεργασίας φυσικής γλώσσας (natural language processing) όσο και για εφαρμογές όρασης υπολογιστών (computer vision). Συγκεκριμένα, σημαντική πηγή έμπνευσης αποτέλεσαν τα έργα [81,125]. Δανειζόμενοι στοιχεία από τη χρήση μηχανισμού αυτο–προσοχής σε εικόνες και διατηρώντας παράλληλα τις βασικές υποθέσεις των νευρωνικών δικτύων με κάψουλες επινοήσαμε μια οικογένεια από γρήγορους, κλιμακώσιμους (scalable) και μη–επαναληπτικούς αλγορίθμους δρομολόγησης.

Όπως γίνεται αντιληπτό από το κεφάλαιο 3, δεν είναι η πρώτη φορά που χρησιμοποιείται μηχανισμός αυτο–προσοχής στο πλαίσιο των νευρωνικών δικτύων με κάψουλες. Στα έργα [108, 109] χρησιμοποιείται ο εν λόγω μηχανισμός ως επίπεδο που προηγείται του σχηματισμού των PrimaryCaps, το οποίο διαφέρει σημαντικά από την υλοποίησή μας που ενσωματώνει τον μηχανισμό αυτο–προσοχής στον αλγόριθμο δρομολόγησης. Η χρήση του μηχανισμού προσοχής μεταξύ δύο επιπέδων από κάψουλες δημιουργεί ένα σύστημα «παράλληλης προσοχής» αφού αυτός εφαρμόζεται παράλληλα, για κάθε κάψουλα επόμενου επιπέδου.

Αν και η υλοποίησή μας άντλησε στοιχεία από το έργο [49] (το μόνο που εφαρμόζει τον μηχανισμό στον αλγόριθμο δρομολόγησης) είναι πολύ πιο πιστή στις βασικές υποθέσεις των δικτύων με κάψουλες. Εν αντιθέσει, για τον αλγόριθμο δρομολόγησης των Mazzia et al. (τον ονομάζουμε «απλοϊκό αλγόριθμο δρομολόγησης με αυτο–προσοχή») δεν καταφέραμε να δώσουμε μια λογική

εξήγηση που να συμφωνεί με τις αρχές των νευρωνικών δικτύων με κάψουλες. Επίσης, όπως θα φανεί στο επόμενο κεφάλαιο, η υλοποίησή μας με παρόμοιο αριθμό παραμέτρων επιτυγχάνει καλύτερες επιδόσεις (χωρίς ιδιαίτερο πειραματισμό των παραμέτρων).

Στην παρούσα υποενότητα θα παρουσιάσουμε τους δύο εναλλακτικούς αλγορίθμους που αναπτύξαμε καθώς και τη λογική πίσω από τον καθένα. Επίσης, για λόγους πληρότητας, θα παρουσιάσουμε τον «απλοϊκό αλγόριθμο δρομολόγησης με αυτο–προσοχή» που χρησιμοποιείται στην υλοποίηση [49]. Προτού όμως ξεκινήσουμε την παρουσίαση, κρίνεται σκόπιμο να ορίσουμε τον συμβολισμό που θα χρησιμοποιηθεί σε όλη την έκταση των αλγορίθμων μας. Χρησιμοποιούμε το δικό μας συμβολισμό ο οποίος διαφέρει από αυτόν των προηγούμενων αλγορίθμων καθώς δεν ήταν επαρκής ώστε να περιγράψει τις νέες έννοιες που εισάγουμε. Έτσι έχουμε:

- Σαν σύμβαση, χρησιμοποιούμε το δείκτη i για να αναφερθούμε σε κάψουλες επιπέδου L και το δείκτη j για τις κάψουλες επιπέδου $L+1$.

- Το σύνολο των καψουλών ενός επιπέδου L το συμβολίζουμε ως Ω_L και τον πληθάριθμο του συνόλου (cardinality) ως $n^L = |\Omega_L|$.

- Τις επιμέρους κάψουλες ενός επιπέδου L τις συμβολίζουμε με C_i^L ενώ με C^L συμβολίζουμε τη μήτρα στην οποία οργανώνονται όλες οι κάψουλες επιπέδου L. Το μέγεθος της κάθε κάψουλας επιπέδου L (μήκος του διανύσματος με το οποίο αναπαρίσταται η κάψουλα) συμβολίζεται με d^L.

- Μεταξύ δύο επιπέδων από κάψουλες L και $L+1$, κάθε κάψουλα $C_i^L \in \Omega_L$ προβλέπει ποια θα είναι η πόζα της κάθε κάψουλας $C_j^{L+1} \in \Omega_{L+1}$. Έτσι προκύπτουν $n^L \times n^{L+1}$ προβλέψεις (ή ψήφοι) όπου την κάθε μια τη συμβολίζουμε με V_{ji}^L. Αυτές οι ψήφοι οργανώνονται στη μήτρα V^L η οποία όπως είναι λογικό, περιέχει $n^{L+1} \times n^L$ κάψουλες - ψήφους: n^L ψήφους για κάθε κάψουλα C_i^{L+1}.

- Σε περίπτωση που θα θέλαμε να αναφερθούμε σε όλες τις ψήφους που αφορούν μια κάψουλα επιπέδου $L+1$ αρκεί να «δεσμεύσουμε» την ελεύθερη μεταβλητή j και να γράψουμε V_j^L (ή, καλύτερα, $V_{j:}^L$ ώστε να είναι εμφανές ότι η μεταβλητή i που αφορά τις κάψουλες επιπέδου L παραμένει ελεύθερη). Κατά αντιστοιχία, αν θα θέλαμε να αναφερθούμε στις ψήφους που προκύπτουν από την κάψουλα C_i^L θα γράφαμε V_i^L (ή, σαφέστερα, $V_{:i}^L$). Κατά αυτόν τον τρόπο, αν δεσμεύσουμε όλες τις διαστάσεις μιας μήτρας, τότε έχουμε ένα βαθμωτό μέγεθος. Για παράδειγμα, έστω κάψουλες $C^L \in \Re^{n^L \times d^L}$. Τότε αν δεσμεύσουμε την ελεύθερη μεταβλητή $i \in [1, n^L]$ τότε ισχύει $C_i^L \in \Re^{d^L}$. Αν δεσμεύσουμε και το επιμέρους τμήμα του διανύσματος της κάψουλας λαμβάνουμε έναν πραγματικό αριθμό, δηλαδή: $C_{ik}^L \in \Re$[8].

- Όπως γίνεται κατανοητό από το ανωτέρω παράδειγμα, ο δείκτης k_L θα χρησιμοποιείται για να αναφερόμαστε στα επιμέρους στοιχεία ενός διανύσματος μεγέθους d^L.

- Οι πίνακες από βάρη ενός επιπέδου L που αποθηκεύουν τις σχέσεις μέρους - όλου οργανώνονται στη μήτρα W^L. Η μήτρα περιέχει όλα τα επιμέρους βάρη W_{ji}^L που χρησιμοποιούν

[8] Μάλιστα είναι $C_{ik}^L \in [0, 1]$ αφού η συνάρτηση squash δεν επιτρέπει τα διανύσματα των καψουλών να έχουν μήκος μεγαλύτερο της μονάδας.

οι κάψουλες C^L για να παράξουν τις προβλέψεις V^L. Για μια κάψουλα $C_i^L \in \Omega_L$, η ψήφος της υπολογίζεται με τη φόρμουλα: $V_{ji}^L = C_i^L \times W_{ji}^L$.

- Συνήθως, μπορεί να αναφερόμαστε στις κάψουλες επιπέδου L ως κάψουλες παιδιά και στις αντίστοιχες του επιπέδου $L+1$ ως κάψουλες γονείς.

- Ο επιμέρους πίνακας που προκύπτει από τον μηχανισμό αυτο–προσοχής των ψήφων V_j^L για μια κάψουλα C_j^{L+1} συμβολίζεται ως A_j^L. Η μήτρα που περιέχει όλους τους πίνακες προσοχής ($\forall C_j^{L+1} \in \Omega_{L+1}$) συμβολίζεται με A^L.

- Στην περίπτωση που χρησιμοποιείται αυτο–προσοχή πολλών κεφαλών (multi head self attention) η αναφορά σε επιμέρους κεφαλές γίνεται με το δείκτη h και ο συνολικός αριθμός των κεφαλών συμβολίζεται ως nh^L. Για παράδειγμα, η πρώτη κεφαλή (head) ενός πίνακα προσοχής για μια κάψουλα C_j^{L+1} συμβολίζεται ως $A_{jh=1}^L$. Η αναφορά στο σύνολο των κεφαλών προσοχής ενός επιπέδου L για μια κάψουλα C_j^{L+1} γίνεται με τον συμβολισμό H_j^L. Δηλαδή, ισχύει ότι $A_{jh}^L \in H_j^L$.

- Η μήτρα βαρών δρομολόγησης μεταξύ επιπέδων L και $L+1$ συμβολίζεται με $\mathbf{R}^L$. Αυτή, μεταξύ δύο πλήρως διασυνδεδεμένων επιπέδων από κάψουλες, περιέχει όλα τα βάρη ακμών τα οποία συμβολίζονται με $\mathbf{R}_{ij}^L$.

Απλοϊκός αλγόριθμος δρομολόγησης με αυτο–προσοχή

Στην παράγραφο αυτή γίνεται αναφορά στον αλγόριθμο που χρησιμοποιεί το έργο [49]. Σε γενικές γραμμές, πρόκειται για τον μη–επαναληπτικό αλγόριθμο ο οποίος χρησιμοποιεί τον μηχανισμό αυτο–προσοχής προκειμένου να υπολογίσει τα βάρη δρομολόγησης (coupling coefficients). Αν και το έργο δεν προσφέρει μια πειστική, λογική εξήγηση για όλα τα βήματα του αλγορίθμου, οι υψηλές επιδόσεις του παρακίνησαν την κατασκευή των εξελιγμένων αλγορίθμων που παρουσιάζουμε σε επόμενες παραγράφους.

Αν και τα ανωτέρω βήματα έχουν παρουσιαστεί πολύ αναλυτικά με ισοδύναμες ("equivalent"), σημειακές εκφράσεις, μπορούμε να κάνουμε τα παρακάτω σχόλια:

- Το σύμβολο $*$ συμβολίζει τον πολλαπλασιασμό μεταξύ δύο βαθμωτών μεγεθών (scalars).

- Η συνάρτηση $softmax()$ στο βήμα 19 σκοπό έχει να επιβάλει τον περιορισμό $\sum_j^{n^{L+1}} R_{ji} = 1$ και να ενισχύσει το βάρος δρομολόγησης με το μεγαλύτερο μέτρο έτσι ώστε, τελικά, μια κάψουλα παιδί να μην ανήκει ολοκληρωτικά σε πολλούς γονείς (single parent assumption). Σε τεχνικό επίπεδο, η συνάρτηση δέχεται σαν είσοδο ένα διάνυσμα στήλη και μπορεί να οριστεί ως εξής:
$$softmax(X \in \Re^{n\times 1}) = \frac{\exp(X_k)}{\sum_k^n \exp(X_k)}. \tag{4.21}$$

- Στη γραμμή 18 του αλγορίθμου, το αριστερό άθροισμα διενεργείται σημειακά σε διανύσματα γραμμές. Ουσιαστικά, για μια κάψουλα γονέα j, οι ομοιότητες της εκάστοτε ψήφου από την κάψουλα C_i^L με τις άλλες ψήφους από τις κάψουλες $C_{\dot{i}}^L \in \Omega_L \setminus C_i^L$ αθροίζονται στην τιμή R_{ji}^L. Η ποιοτική ερμηνεία αυτής της πράξης δεν αναλύεται στο έργο [49]. Σε μια προσπάθεια ερμηνείας, θα μπορούσαμε να αναφέρουμε ότι κάθε θέση i του διανύσματος γραμμής R_j^L

Algorithm 6 Απλοϊκός Αλγόριθμος Δρομολόγησης με Αυτο–προσοχή

Input PrimaryCaps $C^L \in \Re^{n^L \times d^L}$

Output DigitCaps $C^{L+1} \in \Re^{n^{L+1} \times d^{L+1}}$

Trainable Parameters $W^L \in \Re^{n^{L+1} \times n^L \times d^L \times d^{L+1}}, b^L \in \Re^{n^{L+1} \times n^L}$

procedure MAIN(C^L) ▷ Input: $C^L \in \Re^{n^L \times d^L}$
  $V^L \leftarrow$ COMPUTEVOTES(C^L)
  $A^L \leftarrow$ SELF-ATTENTION(V^L)
  $\mathbf{R}^L \leftarrow$ COMPUTEROUTINGCOEFFICIENTS(A^L)
  $\forall j \in \Omega_{L+1} : S_j^{L+1} \leftarrow (\mathbf{R}_j^L + b_j^L) \times V_j^L$ ▷ Equiv.: $S_{jk}^{L+1} \leftarrow \sum_i^{n^L} (\mathbf{R}_{ji}^L + b_{ji}^L) * V_{jik}^L$
  $\forall j \in \Omega_{L+1} : C_j^{L+1} \leftarrow squash(S_j^{L+1})$
  return C^{L+1} ▷ Output: $C^{L+1} \in \Re^{n^{L+1} \times d^{L+1}}$
end procedure
procedure COMPUTEVOTES(C^L) ▷ Input: $C^L \in \Re^{n^L \times d^L}$
  $\forall j \in \Omega_{L+1}, \forall i \in \Omega_L : V_{ji:}^L \leftarrow C_{i:}^L \times W_{ji::}^L$ ▷ Equiv.: $V_{jik_{L+1}}^L \leftarrow \sum_{k_L}^{d^L} C_{ik_L}^L * W_{jik_Lk_{L+1}}$
  return V^L ▷ Output: $V^L \in \Re^{n^{L+1} \times n^L \times d^{L+1}}$
end procedure
procedure SELF-ATTENTION(V^L) ▷ Input: $V^L \in \Re^{n^{L+1} \times n^L \times d^{L+1}}$
  $\forall j \in \Omega_{L+1} : A_j^L \leftarrow \frac{V_j^L \times V_j^{L\,T}}{\sqrt{d^{L+1}}}$ ▷ Equiv.: $A_{ji_1i_2}^L \leftarrow \sum_k^{d^{L+1}} V_{ji_1k}^L * V_{ji_2k}^L$
  return A^L ▷ Output: $A^L \in \Re^{n^{L+1} \times n^L \times n^L}$
end procedure
procedure COMPUTEROUTINGCOEFFICIENTS(A^L) ▷ Input: $A^L \in \Re^{n^{L+1} \times n^L \times n^L}$
  $\forall j \in \Omega_{L+1} : R_j^L \leftarrow \sum_{i_1}^{n^L} A_{ji_1:}^L$ ▷ Equiv.: $R_{ji_2}^L \leftarrow \sum_{i_1}^{n^L} A_{ji_1i_2}^L$
  $\forall i \in \Omega_L : \mathbf{R}_{:i}^L \leftarrow softmax(R_{:i}^L)$ ▷ Equiv.: $\mathbf{R}_{ji}^L \leftarrow \frac{\exp(R_{ji}^L)}{\sum_j^{n^{L+1}} \exp(R_{ji}^L)}$
  return $\mathbf{R}^L$ ▷ Output: $\mathbf{R}^L \in \Re^{n^{L+1} \times n^L}$
end procedure

που προκύπτει, φανερώνει ποιες ψήφοι καψουλών C_i^L εμφανίζουν μεγάλη ομοιότητα με τις υπόλοιπες ψήφους για το συγκεκριμένο j. Έτσι, στη συνέχεια, η κάθε κάψουλα C_j^{L+1} θα συντίθεται μόνο από τις ψήφους των C_i^L που εμφάνιζαν μεγάλη ομοιότητα με τις υπόλοιπες ψήφους από τις κάψουλες $C_{\acute{i}}^L \in \Omega_L \setminus C_i^L$ (πάντα για συγκεκριμένο j).

Πλέον, είμαστε σε θέση να παρουσιάσουμε τους δικούς μας αλγορίθμους οι οποίοι έχουν περισσότερο προφανή ποιοτική ερμηνεία και μάλιστα, επιτυγχάνουν καλύτερα πειραματικά αποτελέσματα.

Αλγόριθμος Δρομολόγησης με Αυτο–προσοχή 1 (Αλγόριθμος RoMAV)

Ο πρώτος αλγόριθμός μας που εξετάζουμε στην παρούσα ενότητα είναι αυτός της δρομολόγησης με αυτο–προσοχή όπου προσθέτουμε τις ψήφους που εμφανίζουν υψηλή συμφωνία (Routing by Merged Agreeing Votes - RoMAV). Ο αλγόριθμος αυτός, μοιάζει πολύ με τον απλοϊκό αλγόριθμο

6 αλλά εδώ, αντί να αθροίζουμε τις γραμμές του πίνακα προσοχής, αθροίζουμε τις προκύπτουσες αναπαραστάσεις (embeddings).

Η γενική ιδέα πίσω από τον αλγόριθμο 7 φαίνεται στο σχήμα 4.5. Αν και έχει δοθεί μεγάλη προσοχή στο να παρουσιαστεί ο αλγόριθμος με τον πιο σαφή τρόπο, πολλές φορές η κατανόηση του αλγορίθμου δε συνεπάγεται την αντίληψη της ποιοτικής του ερμηνείας. Για τον λόγο αυτό, κρίνεται σκόπιμη η διαισθητική παρουσίαση τόσο του αλγορίθμου 7 (RoMAV) όσο και των αλγορίθμων 8 (RoWSS) και 9 (RoWLS) που τον διαδέχονται.

Ο αλγόριθμος 7 αποσκοπεί στο να φιλτράρει τις ψήφους–διανύσματα V^L με κριτήριο το εσωτερικό τους γινόμενο (μετρική συμφωνίας) και να κατασκευάσει νέες αναπαραστάσεις (embeddings) χρησιμοποιώντας τα διανύσματα που συμφωνούν μεταξύ τους για την πόζα της εκάστοτε κάψουλας C_j^{L+1}. Για την αποδοτική υλοποίηση του φιλτραρίσματος δανειζόμαστε στοιχεία από τον μηχανισμό προσοχής, όπως τον παρουσιάσαμε στην ενότητα 2.3.3. Πιο αναλυτικά, κατασκευάζουμε ένα χάρτη προσοχής (attention map) A_j^L για κάθε κάψουλα C_j^L. Για όλους μαζί τους χάρτες αυτούς, ισχύει ότι $A^L \in [-1,1]^{n^{L+1} \times n^L \times n^L}$. Με άλλα λόγια, σε κάθε θέση $[i_1, i_2]$ ενός εξ αυτών (για μια κάψουλα C_j^L) αποθηκεύεται η ομοιότητα που έχει η ψήφος $V_{ji_1}^L$ με την ψήφο $V_{ji_2}^L$.

Πλέον, σε αυτήν τη φάση του αλγορίθμου έχουμε στη διάθεσή μας για κάθε κάψουλα C_j^{L+1} το βαθμό συμφωνίας μεταξύ όλων των ψήφων για την πόζα της. Από εκεί και πέρα, επιθυμούμε να φιλτράρουμε τις ψήφους με σκοπό να κρατήσουμε μόνον αυτές που εμφανίζουν μεγάλη ομοιότητα μεταξύ τους. Άλλωστε, όπως εξηγήσαμε στην ενότητα 2.2, όταν πολλές ψήφοι $V_{j:}^L$ συμφωνούν για το ποια είναι η πόζα της κάψουλας C_j^{L+1} τότε υπάρχει μεγάλη πιθανότητα, το αντικείμενο το οποίο εκπροσωπεί η κάψουλα C_j^{L+1} να είναι παρόν στην εικόνα. Αντίθετα, θα υπάρχουν πάντα ψήφοι που προέρχονται από κάψουλες παιδιά C_i^L που δε θα συμφωνούν μεταξύ τους (διότι μπορεί να αντιστοιχούν σε τμήματα αντικειμένων που δεν είναι παρόντα στην εικόνα εισόδου). Αυτές τις ψήφους επιθυμούμε να τις εκμηδενίσουμε καθότι, διαφορετικά, θα εισάγουν «θόρυβο» στις προβλέψεις μας. Για τον λόγο αυτό, εφαρμόζουμε μια μη γραμμική συνάρτηση όπως η ReLU σημειακά στις υπολογισμένες ομοιότητες.

Ύστερα από τον υπολογισμό αυτό, έχουμε στη διάθεσή μας μια μήτρα με τις ίδιες διαστάσεις με τον A^L αλλά με μη–αρνητικά στοιχεία. Τον νέο τρισδιάστατο πίνακα αυτόν τον συμβολίζουμε ως ${}^+A^L$ και περιέχει τους χάρτες προσοχής ${}^+A_j^L$ με στοιχεία μη μηδενικά μόνο στα σημεία που αντιστοιχούν σε δύο κάψουλες που συμφωνούν μεταξύ τους. Με άλλα λόγια, μπορεί κανείς να φανταστεί το ${}^+A_j^L$ σαν έναν δισδιάστατο πίνακα $[n^L \times n^L]$ (βλέπε σχήμα 4.5). Ποιοτικά, κάθε γραμμή i του πίνακα[9] A_j^L περιέχει τις ομοιότητες που εμφανίζει η ψήφος V_{ji}^L με όλες τις ψήφους $(V_{j:}^L)$ για την κάψουλα C_j^{L+1}. Φιλτράροντας με τη μη–γραμμική συνάρτηση, κρατάμε μόνο τις θετικές ομοιότητες. Σε τελική ανάλυση, η θέση $[1,1]$ του πίνακα A_j^L είναι το εσωτερικό γινόμενο της ψήφου $V_{ji=1}^L$ με τον εαυτό της, η θέση $[1,2]$ και $[2,1]$ είναι το εσωτερικό γινόμενο του διανύσματος $V_{ji=1}^L$ με το $V_{ji=2}^L$ κ.ο.κ.

Το τελευταίο κοινό βήμα των αλγορίθμων RoMAV και RoWSS (παρουσιάζεται αργότερα) είναι αυτό του υπολογισμού των νέων αναπαραστάσεων, όπως προκύπτουν από τη συνένωση των ψήφων που συμφωνούν μεταξύ τους. Ακολουθώντας και πάλι τη θεωρία των μετασχηματιστών, οι

[9] Το ίδιο ισχύει και για τις στήλες αφού ο πίνακας είναι συμμετρικός.

νέες αναπαραστάσεις (συμβολίζονται με $E^L \in \Re^{n^{L+1} \times n^L \times d^{L+1}}$) υπολογίζονται από το βεβαρημένο ανάλογα με την ομοιότητα άθροισμα των ψήφων. Ας πάρουμε για παράδειγμα τον υπολογισμό του E^L_{ji}. Πρόκειται για τη νέα αναπαράσταση της ψήφου V^L_{ji} η οποία λαμβάνει υπόψη τα «συμφραζόμενα» (context), δηλαδή, τις άλλες ψήφους καψουλών που αναπαριστούν διαφορετικά μέρη του αντικειμένου–όλου στο οποίο συμφωνούν. Ο υπολογισμός λοιπόν πραγματοποιείται προσθέτοντάς τα διανύσματα ψήφων $V^L_{j:}$ με βάρη από το διάνυσμα γραμμή ${}^+A^L_{ji:}$ σύμφωνα με την πράξη $E^L_{ji} \leftarrow {}^+A^L_{ji:} \times V^L_{j:}$. Όπως είναι εμφανές, εάν η ψήφος V^L_{ji} δε συμφωνεί με μια ψήφο $V^L_{ji_2}, i_2 \neq i$ για το προβλεφθέν αντικείμενο, τότε η τελευταία, δε θα ληφθεί υπόψη για τον υπολογισμό της νέας αναπαράστασης από τα συμφραζόμενα (η αντίστοιχη θέση του διανύσματος γραμμής θα είναι μηδενική).

Το τελευταίο σημείο (και αυτό που διαφοροποιεί τους αλγορίθμους RoMAV και RoWSS) είναι αυτό του υπολογισμού των C^{L+1}_j. Στον αλγόριθμο RoMAV, απλά έχουμε ότι $C^{L+1}_j \leftarrow \sum_i^{n^L} E^L_{ji}$. Δηλαδή, η κάψουλα C^{L+1}_j προκύπτει από το άθροισμα όλων των νέων αναπαραστάσεων των ψήφων που την αφορούν. Αυτό μπορεί να φανεί σαν ένα ακόμα βήμα «φιλτραρίσματος μέσω της πολυδιάστατης σύμπτωσης» (high dimensional coincidence filtering) αφού οι ψήφοι που εμφάνιζαν μεγάλη συμφωνία μεταξύ τους αφενός έχουν νέες αναπαραστάσεις με μεγαλύτερο μήκος και αφετέρου, κατά την πρόσθεση του τελευταίου βήματος θα ενισχυθούν μεταξύ τους και θα αποσιωπήσουν τις αναπαραστάσεις με τις οποίες δε συμφωνούν. Φυσικά, κοιτώντας την ευρύτερη εικόνα, αν καμία αναπαράσταση E^L_{ji} δε συμφωνεί σημαντικά με τις υπόλοιπες για μια κάψουλα C^{L+1}_j τότε η κάψουλα αυτή θα έχει διάνυσμα με μικρό μέτρο.

Σαν τελικό σχόλιο να αναφέρουμε ότι ο αλγόριθμος αυτός, όντας μη–επαναληπτικός, δεν περιλαμβάνει ρητά την ανατροφοδότηση από πάνω προς τα κάτω (top down feedback). Επεξηγηματικά, να αναφέρουμε ότι η συμφωνία που μπορεί να υπάρξει στις ψήφους για μια κάψουλα C^{L+1}_j δε θα επηρεάσει τη διαμόρφωση των άλλων καψουλών $C^{L+1}_{\hat{j}}$. Σημαντικός λόγος για αυτό το χαρακτηριστικό είναι ότι δεν επιβάλουμε κάποιον περιορισμό (τύπου «υπόθεσης μοναδικού πατέρα» - single parent assumption) ώστε να προκαλέσουμε ανταγωνισμό μεταξύ των καψουλών γονέων για το ποιες ψήφους θα «εξηγήσουν». Σύμφωνα με την παρούσα υλοποίηση, το κάθε τμήμα αντικειμένου μπορεί να ανήκει σε περισσότερα του ενός αντικείμενα και να συμμετέχει στη διαμόρφωση της πόζας τους.

Σε αυτό το πλαίσιο, αναπτύξαμε τον αλγόριθμο RoWSS ο οποίος διαφέρει στο τελευταίο βήμα και διατηρεί όλες τις βασικές υποθέσεις των νευρωνικών δικτύων με κάψουλες ενώ παράλληλα, είναι γρήγορος και μη–επαναληπτικός. Περισσότερα για αυτόν και την παραλλαγή του, RoWLS στην επόμενη υπο–ενότητα.

Algorithm 7 Αλγόριθμος Δρομολόγησης με Αυτο–προσοχή 1 (Αλγόριθμος RoMAV)

Input PrimaryCaps $C^L \in \Re^{n^L \times d^L}$

Output DigitCaps $C^{L+1} \in \Re^{n^{L+1} \times d^{L+1}}$

Trainable Parameters $W^L \in \Re^{n^{L+1} \times n^L \times d^L \times d^{L+1}}$

procedure MAIN-ROMAV(C^L) ▷ Input: $C^L \in \Re^{n^L \times d^L}$

  $V^L \leftarrow$ COMPUTEVOTES(C^L)

  $A^L \leftarrow$ SELF-ATTENTION(V^L)

  ${}^+A^L \leftarrow$ COMPUTENONNEGATIVEATTENTIONMAP(A^L)

  $E^L \leftarrow$ NEWEMB(${}^+A^L, V^L$) ▷ Computes new, context-aware, votes.

  $\forall j \in \Omega_{L+1} : S_j^{L+1} \leftarrow \sum_i^{n^L} E_{ji}^L$ ▷ Equiv.: $S_{jk}^{L+1} \leftarrow \sum_i^{n^L} E_{jik}^L$

  $\forall j \in \Omega_{L+1} : C_j^{L+1} \leftarrow squash(S_j^{L+1})$

  return C^{L+1} ▷ Output: $C^{L+1} \in \Re^{n^{L+1} \times d^{L+1}}$

end procedure

procedure COMPUTEVOTES(C^L) ▷ Input: $C^L \in \Re^{n^L \times d^L}$

  $\forall j \in \Omega_{L+1}, \forall i \in \Omega_L : V_{ji:}^L \leftarrow C_{i:}^L \times W_{ji::}^L$ ▷ Equiv.: $V_{jik_{L+1}}^L \leftarrow \sum_{k_L}^{d^L} C_{ik_L}^L * W_{jik_Lk_{L+1}}$

  return V^L ▷ Output: $V^L \in \Re^{n^{L+1} \times n^L \times d^{L+1}}$

end procedure

procedure SELF-ATTENTION(V^L) ▷ Input: $V^L \in \Re^{n^{L+1} \times n^L \times d^{L+1}}$

  $\forall j \in \Omega_{L+1} : A_j^L \leftarrow \frac{V_j^L \times V_j^{L^T}}{\sqrt{d^{L+1}}}$ ▷ Equiv.: $A_{ji_1i_2}^L \leftarrow \sum_k^{d^{L+1}} V_{ji_1k}^L * V_{ji_2k}^L$

  return A^L ▷ Output: $A^L \in \Re^{n^{L+1} \times n^L \times n^L}$

end procedure

procedure COMPUTENONNEGATIVEATTENTIONMAP(A^L) ▷ Input: $A^L \in \Re^{n^{L+1} \times n^L \times n^L}$

  $\forall j \in \Omega_{L+1} : {}^+A_j^L \leftarrow \mathbf{ReLU}(A_j^L)$ ▷ Equiv.: ${}^+A_{ji\tilde{i}}^L \leftarrow ReLU(A_{ji\tilde{i}}^L)$

  return ${}^+A^L$ ▷ Output: ${}^+A^L \in \Re^{n^{L+1} \times n^L \times n^L}$

end procedure

procedure NEWEMB(${}^+A^L, V^L$) ▷ Input: ${}^+A^L \in \Re^{n^{L+1} \times n^L \times n^L}, V^L \in \Re^{n^{L+1} \times n^L \times d^{L+1}}$

  $\forall j \in \Omega_{L+1} : E_j^L \leftarrow {}^+A_j^L \times V_j^L$ ▷ Equiv.: $E_{jik}^L \leftarrow \sum_{\tilde{i}}^{n^L} {}^+A_{ji\tilde{i}}^L * V_{j\tilde{i}k}^L$

  return E^L ▷ Output: $E^L \in \Re^{n^{L+1} \times n^L \times d^{L+1}}$

end procedure

Αλγόριθμος Δρομολόγησης με Αυτο–προσοχή 2 (Αλγόριθμος RoWSS)

Ο αλγόριθμος Routing by Winner of Similarity Scores - RoWSS (αλγόριθμος 8) μοιράζεται πολλά στοιχεία με τον αλγόριθμο RoMAV για αυτό και δε θα επαναλάβουμε την επεξήγηση των πρώτων βημάτων, παρά μόνο θα συνεχίσουμε από το σημείο στο οποίο οι δύο αλγόριθμοι διαφέρουν. Ειδικότερα, παρόλο που και οι δύο αλγόριθμοι υπολογίζουν τις νέες αναπαραστάσεις (embeddings) E^L, ο αλγόριθμος RoWSS χρησιμοποιεί έναν πιο σύνθετο μηχανισμό για τη συγκρότηση των C^{L+1}.

Ο πιο σύνθετος μηχανισμός περιλαμβάνει τη διεργασία της «εύρεσης των δεικτών νικητών» (find winner indices). Πρόκειται ουσιαστικά για έναν μηχανισμό που στην πρώτη φάση του υπολογίζει τα σκορ ομοιότητας (similarity scores) της κάθε γραμμής, για κάθε δισδιάστατο πίνακα A_j^L. Το σκορ αυτό προκύπτει από την πράξη $SC_j^L \leftarrow (\sum_{i_2}^{n^L} A_{j:i_2}^L)^T$ η οποία πραγματοποιείται στη γραμμή 28 του αλγορίθμου. Κατά αυτόν τον τρόπο, συνολικά, για κάθε j δημιουργείται ο πίνακας $SC^L \in [-1,1]^{n^{L+1} \times n^L}$.

Ποιοτικά, το διάνυσμα SC_j^L (για μια τυχαία κάψουλα γονέα C_j^{L+1}) δείχνει τη συνολική ομοιότητα που εμφανίζει κάθε ψήφος V_{ji}^L με όλες τις ψήφους $V_{j:}^L$ (συμπεριλαμβανομένου και του εαυτού της). Συνεπώς, έχοντας μια τέτοια μετρική είναι εύκολο έπειτα να επιλεγεί μια ψήφος–εκπρόσωπος ως αυτή που εμφανίζει τη μεγαλύτερη ομοιότητα με τις υπόλοιπες (γραμμή 31 όπου η πράξη $argmax$ δέχεται διανύσματα στήλες). Προτού γίνει αυτό όμως, στη γραμμή 30 του αλγορίθμου λαμβάνει χώρα η πράξη ομαλής μεγιστοποίησης (softmax) των σκορ ανά j και λαμβάνεται ο πίνακας $SoftSC^L$ (ο ορισμός της πράξης ομαλούς μεγιστοποίησης είναι ίδιος με αυτόν της σχέσης 4.21).

Η εφαρμογή της συνάρτησης ομαλής μεγιστοποίησης (softmax) έχει σαν στόχο να επιβάλει τον περιορισμό $\sum_i^{n^L} SC_{ji}^L = 1$. Εκφρασμένο διαφορετικά, έχει σκοπό να επιβάλει την υπόθεση μοναδικού πατέρα (single parent assumption)[10]. Με τη (προαιρετική) πράξη αυτή, προκαλούμε ανταγωνισμό μεταξύ των καψουλών του επόμενου επιπέδου (C^{L+1}) στο να προσελκύσουν όσο το δυνατόν περισσότερες ψήφους. Επίσης, μοναδικό χαρακτηριστικό του αλγορίθμου είναι ότι με μη–επαναληπτικό τρόπο επιτυγχάνεται αυτό που έχουμε ονομάσει στα προηγούμενα κεφάλαια ως ανατροφοδότηση από πάνω προς τα κάτω (top down feedback). Για παράδειγμα, έστω ότι μια κάψουλα C_i^L έχει ψήφους $V_{j_1 i}^L$ και $V_{j_2 i}^L$ που εμφανίζουν μεγάλη ομοιότητα με τις υπόλοιπες ψήφους για τα j_1 και j_2 αντίστοιχα. Αντί να διαδραματίσει σημαντικό ρόλο στη διαμόρφωση και των δύο διανυσμάτων $C_{j_1}^{L+1}$ και $C_{j_2}^{L+1}$, λόγω της υπόθεσης μοναδικού πατέρα, θα λάβει ανατροφοδότηση από τους βαθμούς συμφωνίας με τις υπόλοιπες ψήφους και τελικά θα συμβάλλει σημαντικά στη διαμόρφωση μόνο της μιας εκ των καψουλών $C_{j_1}^{L+1}$ και $C_{j_2}^{L+1}$ (αυτή όπου η ψήφος της $V_{j_1 i}^L$ είτε $V_{j_2 i}^L$ συμφωνούσε λίγο περισσότερο με τις υπόλοιπες ψήφους $V_{j_1:}^L$ και $V_{j_2:}^L$ αντίστοιχα).

Συνεχίζοντας την περιγραφή του αλγορίθμου, χρησιμοποιώντας τον νέο πίνακα $SoftSC^L$ βρίσκονται οι δείκτες των γραμμών με το μεγαλύτερο σκορ (δείκτες των νικητών, των νέων αναπαραστάσεων δηλαδή που θα εκπροσωπήσουν την κλάση). Οι δείκτες αυτοί οργανώνονται σε έναν πίνακα $Winners^L \in \mathbb{Z}^{n^{L+1} \times 1}$, ένα δείκτη δηλαδή για κάθε κάψουλα C_j^{L+1}. Έτσι, πολύ εύκολα θέτουμε στα διανύσματα των καψουλών του επόμενου επιπέδου τις νέες αναπαραστάσεις που αντιστοιχούν στους νικητές (αφού πρώτα εφαρμόσουμε τη συνάρτηση σύνθλιψης squash).

Στο σημείο αυτό κρίνεται ωφέλιμο να αναφέρουμε τα εξής:

- Στην υλοποίησή μας, υπάρχει (προαιρετικά) η δυνατότητα για κλιμάκωση των αναπαραστάσεων E^L με τα αντίστοιχά τους $SoftSC^L$ προτού εκχωρηθούν στις κάψουλες C^{L+1}.

[10] Φυσικά, όπως και σε όλους τους αλγορίθμους δρομολόγησης, μπορεί μια κάψουλα C_i^L να μοιράσει την ψήφο της σε δύο κάψουλες γονείς αλλά ποτέ να δώσει ολοκληρωτικά την ψήφο της (συντελεστής δρομολόγησης μονάδα) και στις δύο.

Δηλαδή, η πράξη θα ήταν:

$$\forall j \in \Omega_{L+1}, \forall i \in \Omega_L : ScaledE_{ji}^L \leftarrow E_{ji}^L * SoftSC_{ji}^L \tag{4.22}$$

Αυτή η εκχώρηση θα λάμβανε χώρα μετά το βήμα 6 όπου θα έπρεπε να τροποποιήσουμε τη διαδικασία $NewEmb$ να επιστρέφει και τον πίνακα $SoftSC^L$.

- Υπάρχει μια παραλλαγή του αλγορίθμου RoWSS που ακούει στο όνομα RoWLS ο οποίος αντί για τη διαδικασία FindWinnerIndexes χρησιμοποιεί τη διαδικασία FindWinnerIndexesLength. Ουσιαστικά, πρόκειται για τη διαδικασία που πραγματοποιεί την επιλογή των εκπροσώπων όχι με κριτήριο την αθροιστική ομοιότητα γραμμής (SC^L) αλλά με το μήκος των διανυσμάτων αναπαράστασης (L_2 νόρμα). Η μόνη διαφορά στην κύρια διεργασία του αλγορίθμου είναι ότι αντί για την κλήση της διαδικασίας $FindWinnerIndexes$ με όρισμα ${}^+A^L$ γίνεται κλήση στη διαδικασία $FindWinnerIndexesLength$ με όρισμα τη μήτρα V^L. Φυσικά, και εδώ υπάρχει το προαιρετικό βήμα της κλιμάκωσης των νέων αναπαραστάσεων με τα στοιχεία του πίνακα $SoftSC^L$.

- Όλοι οι αλγόριθμοί μας που χρησιμοποιούν προσοχή (RoMAV, RoWSS και RoWLS) υποστηρίζουν μηχανισμό προσοχής πολλών κεφαλών (multi–head attention). Επειδή όμως η προσθήκη του πιο σύνθετου μηχανισμού δεν αλλάζει τη διαισθητική ερμηνεία των αντίστοιχων αλγορίθμων, η διαδικασία που υλοποιεί τον σύνθετο αυτό μηχανισμό αλλά και οι υπόλοιπες διαδικασίες που πρέπει να τροποποιηθούν ελαφρώς για να τον υποστηρίξουν παρουσιάζονται στο τέλος της ενότητας.

Algorithm 8 Αλγόριθμος Δρομολόγησης με Αυτο–προσοχή 2 (Αλγόριθμος RoWSS)

Input PrimaryCaps $C^L \in \Re^{n^L \times d^L}$

Output DigitCaps $C^{L+1} \in \Re^{n^{L+1} \times d^{L+1}}$

Trainable Parameters $W^L \in \Re^{n^{L+1} \times n^L \times d^L \times d^{L+1}}$

procedure MAIN-ROWSS(C^L) ▷ Input: $C^L \in \Re^{n^L \times d^L}$

  $V^L \leftarrow$ COMPUTEVOTES(C^L)

  $A^L \leftarrow$ SELF-ATTENTION(V^L)

  ${}^+A^L \leftarrow$ COMPUTENONNEGATIVEATTENTIONMAP(A^L)

  $E^L \leftarrow$ NEWEMB(${}^+A^L, V^L$) ▷ Computes new, context-aware, votes.

  $Winners^L \leftarrow$ FINDWINNERINDEXES(${}^+A^L$)

  $\forall j \in \Omega_{L+1} : S_j^{L+1} \leftarrow E^L_{ji=Winners_j^L}$ ▷ Equiv.: $S^{L+1}_{jk} \leftarrow E^L_{ji=Winners_j^L k}$

  $\forall j \in \Omega_{L+1} : C_j^{L+1} \leftarrow squash(S_j^{L+1})$

  return C^{L+1} ▷ Output: $C^{L+1} \in \Re^{n^{L+1} \times d^{L+1}}$

end procedure

procedure COMPUTEVOTES(C^L) ▷ Input: $C^L \in \Re^{n^L \times d^L}$

  $\forall j \in \Omega_{L+1}, \forall i \in \Omega_L : V^L_{ji:} \leftarrow C^L_{i:} \times W^L_{ji::}$ ▷ Equiv.: $V^L_{jik_{L+1}} \leftarrow \sum_{k_L}^{d^L} C^L_{ik_L} * W_{jik_Lk_{L+1}}$

  return V^L ▷ Output: $V^L \in \Re^{n^{L+1} \times n^L \times d^{L+1}}$

end procedure

procedure SELF-ATTENTION(V^L) ▷ Input: $V^L \in \Re^{n^{L+1} \times n^L \times d^{L+1}}$

  $\forall j \in \Omega_{L+1} : A_j^L \leftarrow \frac{V_j^L \times {V_j^L}^T}{\sqrt{d^{L+1}}}$ ▷ Equiv.: $A^L_{ji_1i_2} \leftarrow \sum_k^{d^{L+1}} V^L_{ji_1k} * V^L_{ji_2k}$

  return A^L ▷ Output: $A^L \in \Re^{n^{L+1} \times n^L \times n^L}$

end procedure

procedure COMPUTENONNEGATIVEATTENTIONMAP(A^L) ▷ Input: $A^L \in \Re^{n^{L+1} \times n^L \times n^L}$

  $\forall j \in \Omega_{L+1} : {}^+A_j^L \leftarrow \mathbf{ReLU}(A_j^L)$ ▷ Equiv.: ${}^+A^L_{ji\tilde{i}} \leftarrow ReLU(A^L_{ji\tilde{i}})$

  return ${}^+A^L$ ▷ Output: ${}^+A^L \in \Re^{n^{L+1} \times n^L \times n^L}$

end procedure

procedure NEWEMB(${}^+A^L, V^L$) ▷ Input: ${}^+A^L \in \Re^{n^{L+1} \times n^L \times n^L}, V^L \in \Re^{n^{L+1} \times n^L \times d^{L+1}}$

  $\forall j \in \Omega_{L+1} : E_j^L \leftarrow {}^+A_j^L \times V_j^L$ ▷ Equiv.: $E^L_{jik} \leftarrow \sum_{\tilde{i}}^{n^L} {}^+A^L_{ji\tilde{i}} * V^L_{j\tilde{i}k}$

  return E^L ▷ Output: $E^L \in \Re^{n^{L+1} \times n^L \times d^{L+1}}$

end procedure

procedure FINDWINNERINDICES(A^L) ▷ Input: $A^L \in \Re^{n^{L+1} \times n^L \times n^L}$

  $\forall j \in \Omega_{L+1} : SC_j^L \leftarrow (\sum_{i_2}^{n^L} A^L_{j:i_2})^T$ ▷ Equiv.: $SC^L_{ji} \leftarrow \sum_{i_2}^{n^L} A^L_{jii_2}$

  ▷ $SC^L \in \Re^{n^{L+1} \times n^L}$

  $\forall i \in \Omega_L : SoftSC^L_{:i} \leftarrow softmax(SC^L_{:i})$ ▷ Equiv.: $SoftSC^L_{ji} \leftarrow \frac{\exp(SC^L_{ji})}{\sum_j^{n^{L+1}} \exp(SC^L_{ji})}$

  $\forall j \in \Omega_{L+1} : Winners_j^L \leftarrow \underset{i \in [1, n^L]}{\text{argmax}}(SoftSC_j^L)$

  return $Winners^L$ ▷ Output: $Winners^L \in \mathbb{Z}^{n^{L+1} \times 1}$

end procedure

Αλγόριθμος Δρομολόγησης με Αυτο–προσοχή 3 (Αλγόριθμος RoWLS)

Πρόκειται ουσιαστικά για την παραλλαγή του αλγορίθμου RoWSS που χρησιμοποιεί τη διεργασία $FindWinnerIndexesLength$. Με άλλα λόγια, όπως προαναφέρθηκε, ο αλγόριθμος Routing by Winner of Length Scores αλλάζει το κριτήριο επιλογής των αναπαραστάσεων - εκπροσώπων από αυτό των SC^L σε αυτό του μήκους των διανυσμάτων αναπαράστασης. Ο αλγόριθμος αυτός προέκυψε φυσικά από την παρατήρηση ότι ψήφοι (V^L_{ji}) που έχουν μεγάλο βαθμό ομοιότητας γραμμής (SC^L_{ji}) έχουν και μεγάλο μήκος διανύσματος αναπαράστασης (embedding E^L_{ji}). Συνεπώς, επαρκές κριτήριο επιλογής εκπροσώπων είναι το μήκος τους (L_2 νόρμα). Όπως και στον αλγόριθμο RoWSS, η γραμμή 14 είναι προαιρετική και σκοπό έχει να προκαλέσει ανταγωνισμό μεταξύ των καψουλών γονέων. Τέλος, να αναφέρουμε ότι και εδώ υπάρχει η δυνατότητα υπολογισμού της μήτρας $SoftSC^L$ και της κλιμάκωσης των νέων αναπαραστάσεων με αυτή.

Algorithm 9 Αλγόριθμος Δρομολόγησης με Αυτο–προσοχή 3 (Αλγόριθμος RoWLS)

Input PrimaryCaps $C^L \in \Re^{n^L \times d^L}$

Output DigitCaps $C^{L+1} \in \Re^{n^{L+1} \times d^{L+1}}$

Trainable Parameters $W^L \in \Re^{n^{L+1} \times n^L \times d^L \times d^{L+1}}$

1: **procedure** MAIN-ROWLS(C^L) ▷ Input: $C^L \in \Re^{n^L \times d^L}$
2: $V^L \leftarrow \text{COMPUTEVOTES}(C^L)$
3: $A^L \leftarrow \text{SELF-ATTENTION}(V^L)$
4: ${}^{+}A^L \leftarrow \text{COMPUTENONNEGATIVEATTENTIONMAP}(A^L)$
5: $E^L \leftarrow \text{NEWEMB}({}^{+}A^L, V^L)$ ▷ Computes new, context-aware, votes.
6: $Winners^L \leftarrow \text{FINDWINNERINDEXESLENGTH}(V^L)$
7: $\forall j \in \Omega_{L+1} : S^{L+1}_j \leftarrow E^L_{ji=Winners^L_j}$ ▷ Equiv.: $S^{L+1}_{jk} \leftarrow E^L_{ji=Winners^L_j k}$
8: $\forall j \in \Omega_{L+1} : C^{L+1}_j \leftarrow squash(S^{L+1}_j)$
9: **return** C^{L+1} ▷ Output: $C^{L+1} \in \Re^{n^{L+1} \times d^{L+1}}$
10: **end procedure**
11: **procedure** FINDWINNERINDEXESLENGTH(V^L) ▷ Input: $V^L \in \Re^{n^{L+1} \times n^L \times d^{L+1}}$
12: $\forall j \in \Omega_{L+1}, \forall i \in \Omega_L : Length^L_{ji} \leftarrow \left\| V^L_{ji:} \right\|_2$ ▷ Equiv.: $Length^L_{ji} \leftarrow \sqrt{\sum_k^{d^{L+1}} (V^L_{jik})^2}$
13: ▷ $Length^L \in \Re^{n^{L+1} \times n^L}$
14: $\forall i \in \Omega_L : SoftLength^L_{:i} \leftarrow softmax(Length^L_{:i})$ ▷ Equiv.: $SoftLength^L_{ji} \leftarrow \frac{\exp(Length^L_{ji})}{\sum_j^{n^{L+1}} \exp(Length^L_{ji})}$
15: $\forall j \in \Omega_{L+1} : Winners^L_j \leftarrow \underset{i \in [1, n^L]}{\text{argmax}}(SoftLength^L_j)$
16: **return** $Winners^L$ ▷ Output: $Winners^L \in \Re^{n^{L+1} \times 1}$
17: **end procedure**

4.3.3 Αρχιτεκτονική Αποκωδικοποιητή

Οι αλγόριθμοί μας διαθέτουν δύο εναλλακτικές εκδοχές αποκωδικοποιητών. Ο πρώτος αποκωδικοποιητής που μπορεί να χρησιμοποιηθεί είναι όμοιος με αυτόν που χρησιμοποιείται στο έργο [76]

και φαίνεται στην εικόνα 4.2. Πρόκειται για έναν απλό αποκωδικοποιητή με τρία πλήρως διασυνδεδεμένα επίπεδα. Στην υλοποίησή μας, κατά τη διάρκεια της εκπαίδευσης, εφαρμόζουμε μια μάσκα και μηδενίζουμε τις κάψουλες που δεν αντιστοιχούν στη σωστή κλάση. Κατά τον έλεγχο (validation) εφαρμόζουμε μια μάσκα που εκμηδενίζει όλες τις κάψουλες εκτός από αυτή που έχει το μεγαλύτερο μήκος. Ειδικά για το σύνολο δεδομένων MultiMNIST όπου έχουμε δύο προβλέψεις, εφαρμόζουμε δυο ξεχωριστές μάσκες εξόδου και τα δύο αποτελέσματα που προκύπτουν τα τροφοδοτούμε, ξεχωριστά, στον αποκωδικοποιητή.

Ο αλγόριθμός μας έχει τη δυνατότητα να χρησιμοποιήσει έναν ξεχωριστού είδους αποκωδικοποιητή βασισμένο σε συνελικτικά επίπεδα κλιμακωτού βηματισμού (fractionally–strided convolutional layers ή απλά deconvolution layers). Πατώντας στις παρατηρήσεις των έργων [107, 110, 113] ότι ένας τέτοιος, εξελιγμένος μηχανισμός ανακατασκευής εικόνας παράγει καλύτερα αποτελέσματα, τον υιοθετήσαμε με παρόμοιες παραμέτρους. Συγκεκριμένα, ο δεύτερος αποκωδικοποιητής μας περιλαμβάνει ένα πλήρως διασυνδεδεμένο επίπεδο και τρία επίπεδα κλιμακωτού βηματισμού με βήματισμους (strides) 2,1 και 1 αντίστοιχα (από την είσοδο του αποκωδικοποιητή προς την έξοδο). Οι άλλες διαστάσεις είναι τέτοιες ώστε να έχουμε ως έξοδο εικόνα στο μέγεθος της εικόνας εισόδου.

4.3.4 Συνάρτηση Σφάλματος και Λοιπά Στοιχεία Υλοποίησης

Η πιο σημαντική διαφορά στα στοιχεία υλοποίησης των μεθόδων αυτής της ενότητας με τη μέθοδο [76] είναι στη συνάρτηση σύνθλιψης (squashing function). Αναλυτικότερα, αυτή η συνάρτηση στις μεθόδους της ενότητας ορίζεται για ένα διάνυσμα εισόδου x ως εξής [49]:

$$squash(x) = (1 - \frac{1}{\exp \|x\|^2}) \frac{x}{\|x\|} \tag{4.23}$$

Πλην αυτής της εξαίρεσης, τα περισσότερα στοιχεία υλοποίησης είναι τέτοια ώστε να επιτρέπουν την άμεση σύγκριση με τον αλγόριθμο στο έργο [76]. Για παράδειγμα, χρησιμοποιούμε ίδιο μηχανισμό για την εξασθένηση του ρυθμού μάθησης (learning rate decay). Σημειώνουμε ότι για την εκπαίδευση του αλγορίθμου χρησιμοποιείται ο βελτιστοποιητής (optimizer) nAdam. Τέλος, αξίζει να αναφέρουμε ότι έχουμε αυτοματοποιήσει όλες τις διαδικασίες εκπαίδευσης και ελέγχου (validation) και κατά τη διάρκεια εκτέλεσης, παράγεται αναλυτικό αρχείο καταγραφής των παραμέτρων και των επιδόσεων.

4.3.5 Αλγόριθμοι με Μηχανισμούς Αυτοπροσοχής Πολλών Κεφαλών

Στην ενότητα αυτή παρατίθενται οι εκδόσεις των τριών αλγορίθμων μας αλλά με την πιο σύνθετη περίπτωση της προσοχής πολλών κεφαλών (multi–head attention). Επειδή η διαισθητική ερμηνεία των αλγορίθμων δε μεταβάλλεται, παρατίθενται εδώ, στο τέλος της ενότητας χωρίς επεξήγηση.

Procedure 10 Διαδικασία Αυτο–Προσοχής Πολλών Κεφαλών (Multi–Head Procedure)

Trainable Parameters $Wv^L \in \Re^{d^{L+1},nh^L,d_v^L}, Wk \in \Re^{d^{L+1},nh^L,d_k^L}$,
$Wq \in \Re^{d^{L+1},nh^L,d_k^L}, Wo \in \Re^{d_v^L * nh^L,d^{L+1}}$

Optional Trainable Parameters $bv^L \in \Re^{nh^L \times d_v^L}, bk^L \in \Re^{nh^L \times d_k^L}$,
$bq^L \in \Re^{nh^L \times d_q^L}, bo^L \in \Re^{d^{L+1}}$

procedure MULTI-HEAD SELF-ATTENTION(V^L) ▷ Input: $V^L \in \Re^{n^{L+1} \times n^L \times d^{L+1}}$

$Q \leftarrow V^L$

$K \leftarrow V^L$

$V \leftarrow V^L$

$\forall j \in \Omega_{L+1}, \forall h \in H^L : Vp^L_{j::h} \leftarrow V^L_j \times Wv^L_{:h:}$ ▷ Equiv.: $Vp^L_{jikh} \leftarrow \sum_{k_1}^{d^{L+1}} V^L_{jik_1} * Wv^L_{k_1ik}$

▷ $Vp^L \in \Re^{n^{L+1} \times n^L \times d_v^L \times nh^L}$

$\forall j \in \Omega_{L+1}, \forall h \in H^L : Qp^L_{j::h} \leftarrow V^L_j \times Wq^L_{:h:}$ ▷ Equiv.: $Qp^L_{jikh} \leftarrow \sum_{k_1}^{d^{L+1}} V^L_{jik_1} * Wq^L_{k_1ik}$

▷ $Qp^L \in \Re^{n^{L+1} \times n^L \times d_k^L \times nh^L}$

$\forall j \in \Omega_{L+1}, \forall h \in H^L : Kp^L_{j::h} \leftarrow V^L_j \times Wk^L_{:h:}$ ▷ Equiv.: $Kp^L_{jikh} \leftarrow \sum_{k_1}^{d^{L+1}} V^L_{jik_1} * Wk^L_{k_1ik}$

▷ $Kp^L \in \Re^{n^{L+1} \times n^L \times d_k^L \times nh^L}$

$\forall j \in \Omega_{L+1}, \forall i \in \Omega_L : Vp^L_{ji::} \leftarrow Vp^L_{ji::} + bv^{L^T}$

$\forall j \in \Omega_{L+1}, \forall i \in \Omega_L : Kp^L_{ji::} \leftarrow Kp^L_{ji::} + bk^{L^T}$

$\forall j \in \Omega_{L+1}, \forall i \in \Omega_L : Qp^L_{ji::} \leftarrow Qp^L_{ji::} + bq^{L^T}$

$\forall j \in \Omega_{L+1}, \forall h \in H^L : Amh^L_{j::h} \leftarrow \frac{Qp^L_{j::h} \times {Kp^L_{j::h}}^T}{\sqrt{d_k^L}}$

▷ Equiv.: $Amh^L_{ji_1i_2h} \leftarrow \sum_k^{d^{L+1}} Qp^L_{ji_1kh} * Kp^L_{ji_2kh}$

return A^L, Vp^L ▷ Output: $Amh^L \in \Re^{n^{L+1} \times n^L \times n^L \times nh^L}, Vp^L \in \Re^{n^{L+1} \times n^L \times d_v^L \times nh^L}$

end procedure

Procedure 11 Συμπληρωματικές, Βοηθητικές Διαδικασίες (Multi–Head Helper Procedures)

procedure ComputeNonNegativeAttentionMap2(Amh^L)
▷ Input: $Amh^L \in \Re^{n^{L+1} \times n^L \times n^L \times nh^L}$
$\forall j \in \Omega_{L+1}, \forall h \in H^L : {}^+Amh^L_{j::h} \leftarrow \mathbf{ReLU}(Amh^L_{j::h})$
▷ Equiv.: ${}^+Amh^L_{ji\tilde{i}h} \leftarrow ReLU(Amh^L_{ji\tilde{i}h})$
return ${}^+Amh^L$ ▷ Output: ${}^+Amh^L \in \Re^{n^{L+1} \times n^L \times n^L \times nh^L}$
end procedure
procedure NewEmb2($^+Amh^L, Vp^L$)
▷ Input: ${}^+Amh^L \in \Re^{n^{L+1} \times n^L \times n^L \times nh^L}, Vp^L \in \Re^{n^{L+1} \times n^L \times dv^L \times nh^L}$
$\forall j \in \Omega_{L+1}, \forall h \in H^L : Ep^L_{j::h} \leftarrow {}^+Amh^L_{j::h} \times Vp^L_{j::h}$
▷ Equiv.: $Ep^L_{jikh} \leftarrow \sum_{\tilde{i}}^{n^L} {}^+Amh^L_{ji\tilde{i}h} * V^L_{\tilde{i}ikh}$
▷ $Ep^L \in \Re^{n^{L+1} \times n^L \times d_v^L \times nh^L}$
$\forall j \in \Omega_{L+1}, \forall i \in \Omega_L : Econcat^L_{ji} \leftarrow Concatenate(E^L_{ji:h=1}, E^L_{ji:h=2}, \ldots, E^L_{ji:h=nh^L})$
▷ Equiv.: $Econcat^L_{jik} \leftarrow E^L_{ji\tilde{k}=(k-1)modnh^L+1h=kdivnh^L}$
▷ $Econcat^L \in \Re^{n^{L+1} \times n^L \times d_v^L * nh^L}$
$\forall j \in \Omega_{L+1} : E^L_j \leftarrow Econcat^L_j \times Wo^L$ ▷ Equiv.: $E^L_{jik} \leftarrow \sum_{k_2}^{nh^L * dv^L} Econcat^L_{jik_2} * Wo^L_{k_2k}$
$\forall j \in \Omega_{L+1}, \forall i \in \Omega_L : E^L_{ji} \leftarrow E^L_{ji} + bo^L$
return E^L ▷ Output: $E^L \in \Re^{n^{L+1} \times n^L \times d^{L+1}}$
end procedure
procedure AggregateAttentionHeads($^+Amh^L$)
▷ Input: ${}^+Amh^L \in \Re^{n^{L+1} \times n^L \times n^L \times nh^L}$
$\forall j \in \Omega_{L+1}, \forall i \in \Omega_L : {}^+A^L_{ji:} \leftarrow \sum_h^{nh^L} {}^+Amh^L_{ji:h}$ ▷ Equiv.: ${}^+A^L_{ji_1i_2} \leftarrow \sum_h^{nh^L} {}^+Amh^L_{ji_1i_2h}$
return ${}^+A^L$ ▷ Output: ${}^+A^L \in \Re^{n^{L+1} \times n^L \times n^L}$
end procedure

Algorithm 12 Αλγόριθμος 1 με Αυτο–Προσοχή Πολλών Κεφαλών (Multihead RoMAV)

Input PrimaryCaps $C^L \in \Re^{n^L \times d^L}$

Output DigitCaps $C^{L+1} \in \Re^{n^{L+1} \times d^{L+1}}$

Trainable Parameters $W^L \in \Re^{n^{L+1} \times n^L \times d^L \times d^{L+1}}, Wv^L \in \Re^{d^{L+1}, nh^L, d_v^L}$,
$Wk \in \Re^{d^{L+1}, nh^L, d_k^L}, Wq \in \Re^{d^{L+1}, nh^L, d_k^L}, Wo \in \Re^{d_v^L * nh^L, d^{L+1}}$

Optional Trainable Parameters $bv^L \in \Re^{nh^L \times d_v^L}, bk^L \in \Re^{nh^L \times d_k^L}$,
$bq^L \in \Re^{nh^L \times d_q^L}, bo^L \in \Re^{d^{L+1}}$

procedure Main-RoMAV-Multihead(C^L) ▷ Input: $C^L \in \Re^{n^L \times d^L}$
  $V^L \leftarrow$ ComputeVotes(C^L)
  $Amh^L, Vp^L \leftarrow$ Multi-Head Self-Attention(V^L)
  ${}^+Amh^L \leftarrow$ ComputeNonNegativeAttentionMap2(Amh^L)
  $E^L \leftarrow$ NewEmb2(${}^+Amh^L, Vp^L$) ▷ Computes new, context-aware, votes.
  ${}^+A^L \leftarrow$ AggregateAttentionHeads(${}^+Amh^L$)
  $\forall j \in \Omega_{L+1} : S_j^{L+1} \leftarrow \sum_i^{n^L} E_{ji}^L$ ▷ Equiv.: $S_{jk}^{L+1} \leftarrow \sum_i^{n^L} E_{jik}^L$
  $\forall j \in \Omega_{L+1} : C_j^{L+1} \leftarrow squash(S_j^{L+1})$
  return C^{L+1} ▷ Output: $C^{L+1} \in \Re^{n^{L+1} \times d^{L+1}}$
end procedure

Algorithm 13 Αλγόριθμος 2 με Αυτο–Προσοχή Πολλών Κεφαλών (Multihead RoWSS)

Input PrimaryCaps $C^L \in \Re^{n^L \times d^L}$

Output DigitCaps $C^{L+1} \in \Re^{n^{L+1} \times d^{L+1}}$

Trainable Parameters $W^L \in \Re^{n^{L+1} \times n^L \times d^L \times d^{L+1}}, Wv^L \in \Re^{d^{L+1}, nh^L, d_v^L}$,
$Wk \in \Re^{d^{L+1}, nh^L, d_k^L}, Wq \in \Re^{d^{L+1}, nh^L, d_k^L}, Wo \in \Re^{d_v^L * nh^L, d^{L+1}}$

Optional Trainable Parameters $bv^L \in \Re^{nh^L \times d_v^L}, bk^L \in \Re^{nh^L \times d_k^L}$,
$bq^L \in \Re^{nh^L \times d_q^L}, bo^L \in \Re^{d^{L+1}}$

procedure Main-RoWSS-Multihead(C^L) ▷ Input: $C^L \in \Re^{n^L \times d^L}$
  $V^L \leftarrow$ ComputeVotes(C^L)
  $Amh^L, Vp^L \leftarrow$ Multi-Head Self-Attention(V^L)
  ${}^+Amh^L \leftarrow$ ComputeNonNegativeAttentionMap2(Amh^L)
  $E^L \leftarrow$ NewEmb2(${}^+Amh^L, Vp^L$) ▷ Computes new, context-aware, votes.
  ${}^+A^L \leftarrow$ AggregateAttentionHeads(${}^+Amh^L$)
  $Winners^L \leftarrow$ FindWinnerIndexes(${}^+A^L$)
  $\forall j \in \Omega_{L+1} : S_j^{L+1} \leftarrow E_{ji=Winners_j^L}^L$ ▷ Equiv.: $S_{jk}^{L+1} \leftarrow E_{ji=Winners_j^L k}^L$
  $\forall j \in \Omega_{L+1} : C_j^{L+1} \leftarrow squash(S_j^{L+1})$
  return C^{L+1} ▷ Output: $C^{L+1} \in \Re^{n^{L+1} \times d^{L+1}}$
end procedure

Algorithm 14 Αλγόριθμος 3 με Αυτο–Προσοχή Πολλών Κεφαλών (Multihead RoWLS)

Input PrimaryCaps $C^L \in \Re^{n^L \times d^L}$

Output DigitCaps $C^{L+1} \in \Re^{n^{L+1} \times d^{L+1}}$

Trainable Parameters $W^L \in \Re^{n^{L+1} \times n^L \times d^L \times d^{L+1}}, Wv^L \in \Re^{d^{L+1}, nh^L, d_v^L}$,
$Wk \in \Re^{d^{L+1}, nh^L, d_k^L}, Wq \in \Re^{d^{L+1}, nh^L, d_k^L}, Wo \in \Re^{d_v^L * nh^L, d^{L+1}}$

Optional Trainable Parameters $bv^L \in \Re^{nh^L \times d_v^L}, bk^L \in \Re^{nh^L \times d_k^L}$,
$bq^L \in \Re^{nh^L \times d_q^L}, bo^L \in \Re^{d^{L+1}}$

1: **procedure** MAIN-ROWLS-MULTIHEAD(C^L) ▷ Input: $C^L \in \Re^{n^L \times d^L}$
2: $V^L \leftarrow$ COMPUTEVOTES(C^L)
3: $Amh^L, Vp^L \leftarrow$ MULTI-HEAD SELF-ATTENTION(V^L)
4: ${}^+Amh^L \leftarrow$ COMPUTENONNEGATIVEATTENTIONMAP2(Amh^L)
5: $E^L \leftarrow$ NEWEMB2(${}^+Amh^L, Vp^L$) ▷ Computes new, context-aware, votes.
6: $Winners^L \leftarrow$ FINDWINNERINDEXESLENGTH(V^L)
7: $\forall j \in \Omega_{L+1} : S_j^{L+1} \leftarrow E^L_{ji=Winners_j^L}$ ▷ Equiv.: $S_{jk}^{L+1} \leftarrow E^L_{ji=Winners_j^L k}$
8: $\forall j \in \Omega_{L+1} : C_j^{L+1} \leftarrow squash(S_j^{L+1})$
9: **return** C^{L+1} ▷ Output: $C^{L+1} \in \Re^{n^{L+1} \times d^{L+1}}$
10: **end procedure**

4.4 SOM-Caps

Σε αυτήν την ενότητα παρουσιάζουμε την τέταρτη μέθοδο, την οποία αναπτύξαμε στην προσπάθειά μας να εξετάσουμε ορισμένες εναλλακτικές προσεγγίσεις των βασικών αρχών των νευρωνικών δικτύων με κάψουλες. Όπως θα δούμε στη συνέχεια, υπό ορισμένες παραμετροποιήσεις του αλγορίθμου μας, αυτός συμπεριφέρεται αρκετά διαφορετικά από τους βασικούς αλγορίθμους δρομολόγησης που χρησιμοποιούνται στα έργα [47, 76].

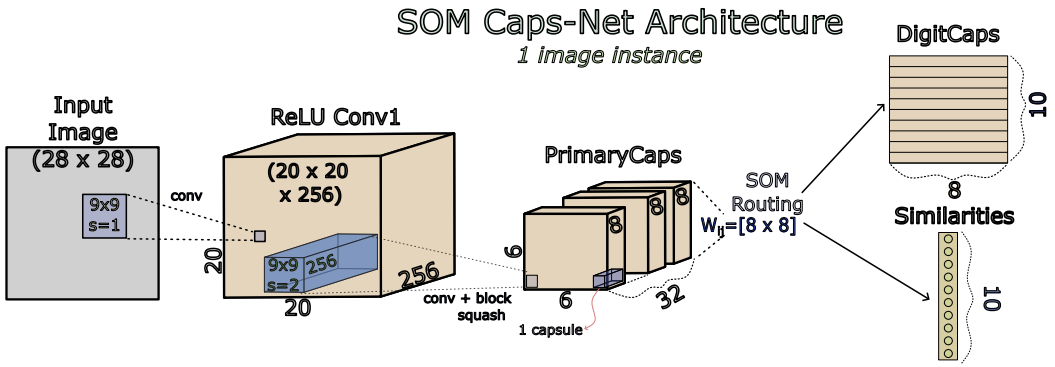
Πηγή έμπνευσης για την ανάπτυξη της μεθόδου αυτής ήταν ο αλγόριθμος για τον σχηματισμό του χάρτη αυτο–οργάνωσης (SOM algorithm), όπως παρουσιάστηκε στην ενότητα 2.4. Η βασική ιδέα είναι ότι η ομαδοποίηση (clustering) που πραγματοποιεί ο αλγόριθμος SOM, με τις κατάλληλες τροποποιήσεις, αποτελεί μια κατάλληλη μέθοδο δρομολόγησης μεταξύ των επιπέδων PrimaryCaps και DigitCaps. Κατά αντιστοιχία, στη θέση των datapoints εμείς έχουμε ψήφους (votes) και θέση των κεντροειδών (centroids) έχουμε τις κάψουλες γονείς. Κατά αυτόν τον τρόπο, η δρομολόγηση ανάγεται στην εύρεση συστάδων στο χώρο των ψήφων και κατόπιν, στην ενημέρωση των DigitCaps με βάση τα κέντρα βάρους των συστάδων αυτών.

Φυσικά, έπρεπε να γίνουν αρκετές τροποποιήσεις στον σειριακό αλγόριθμο SOM ώστε αυτός να είναι ένας ικανοποιητικός αλγόριθμος δρομολόγησης. Μεταξύ αυτών των αλλαγών, όπως θα δούμε στη συνέχεια, είναι η παραλληλοποίηση της διαδικασίας ενημέρωσης των «κόμβων» (DigitCaps στην περίπτωσή μας). Μια ακόμα διαφορά είναι ότι η τοπολογική διάταξη (και για αυτό το λόγο, και η πλευρική απόσταση) δεν είναι τόσο προφανής αφού η διάταξη των κόμβων εξόδου δεν είναι ούτε αυτή προφανής. Με αυτές τις διαφοροποιήσεις, οδηγούμαστε στον «αλγόριθμο δρομολόγησης βασισμένο στον SOM». Αξίζει στο σημείο αυτό να σημειώσουμε ότι, λόγω της πληθώρας των υπερπαραμέτρων που αλλάζουν σημαντικά τη συμπεριφορά του αλγορίθμου, στην ουσία πρόκειται για μια οικογένεια αλγορίθμων.

Στο παρόν κεφάλαιο ξεκινάμε περιγράφοντας τις δύο παραλλαγές της αρχιτεκτονικής του νευρωνικού δικτύου. Έπειτα, συνεχίζουμε με την ανάλυση του καινοτόμου αλγορίθμου δρομολόγησης. Αντιλαμβανόμενοι την πολυπλοκότητα του αλγορίθμου, παροτρύνουμε τον αναγνώστη στο παράρτημα Γ′ για μια αναλυτικότερη και συνάμα λιγότερο τυπική παρουσίαση του αλγορίθμου. Στην επόμενη ενότητα, αναφερόμαστε σε ορισμένες ποιοτικές διαφορές με το δυναμικό αλγόριθμο δρομολόγησης μιας και πρωταρχικός σκοπός της παρούσας μεθόδου είναι η επιλεκτική χαλάρωση ορισμένων περιορισμών και η εξέταση των αποτελεσμάτων. Τέλος, γίνεται λόγος σε λοιπά στοιχεία υλοποίησης όπως ο βελτιστοποιητής και η συνάρτηση σφάλματος.

4.4.1 Αρχιτεκτονική Νευρωνικού Δικτύου

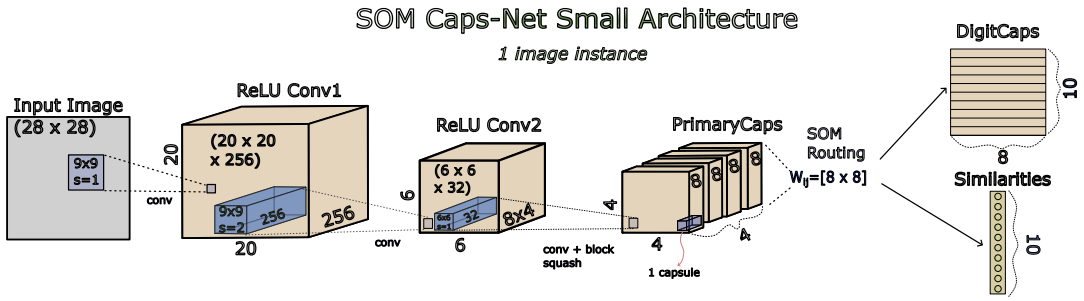
Για την τέταρτη μέθοδο αναπτύξαμε δύο αρχιτεκτονικές νευρωνικού δικτύου. Μια βασική αρχιτεκτονική και μια με ένα επιπλέον συνελικτικό επίπεδο (αρχιτεκτονική small) που σκοπό έχει να μειώσει το υπολογιστικό κόστος. Η πρώτη από αυτές τις αρχιτεκτονικές απεικονίζεται στο σχήμα 4.6 και είναι τέτοια ώστε να διευκολύνεται η σύγκριση με το έργο [76]. Επειδή αριστερό μέρος του δικτύου είναι ίδιο με αυτό προηγούμενων μεθόδων, δεν θα το περιγράψουμε αναλυτικά. Αρκεί να αναφέρουμε ότι για λόγους περιορισμού της πολυπλοκότητας, το μέγεθος των καψουλών εξόδου (DigitCaps) είναι ίσο με 8.



Σχήμα 4.6: Στο σχήμα παρουσιάζεται η βασική αρχιτεκτονική της τέταρτης μεθόδου. Παρατηρούμε ότι μοιράζεται πολλά κοινά στοιχεία με το δίκτυο του έργου [76]. Τα μεγέθη ύψους και πλάτους των χαρτών χαρακτηριστικών είναι υπολογισμένα για το σύνολο δεδομένων MNIST.*Παράχθηκε από το Inkscape.*

Το σημείο στο οποίο παρατηρούμε την μεγαλύτερη διαφορά εντοπίζεται στο επίπεδο εξόδου. Πιο συγκεκριμένα, στο νευρωνικό δίκτυο από κάψουλες με τον αλγόριθμο δρομολόγησης βασισμένο στο SOM (SOM-Caps) διακρύνουμε δύο εξόδους: μια για τις κάψουλες τελευταίου επιπέδου (DigitCaps) και μια για τις «ομοιότητες» (similarities). Ενώ η πρώτη έξοδος είναι παρούσα σε όλα τα νευρωνκά δίκτυα με κάψουλες, η δεύτερη έξοδος κάνει την αρχιτεκτονική της τέταρτης μεθόδου να διαφέρει αρκετά από τις υπόλοιπες.

Αναφορικά με την δεύτερη έξοδο, αυτή προκύπτει από την σύγκριση των αντίστοιχων προβλέψεων πόζας (ψήφων) της κάθε κάψουλας εξόδου με την κάψουλα αυτή. Με απλά λόγια, κάθε στοιχείο της εξόδου ομοιότητας (similarity) είναι μια μετρική του πόσο εύστοχες ήταν οι προβλέψεις των PrimaryCaps στην πρόβλεψη της πόζας της εκάστοτε κάψουλας εξόδου[11]. Αυτό είναι και το διάνυσμα από το οποίο παράγεται η κλάση πρόβλεψης (με την πράξη $argmax$).



Σχήμα 4.7: Στο σχήμα παρουσιάζεται η μικρή αρχιτεκτονική της τέταρτης μεθόδου. Αποσκοπεί στη μείωση του αριθμού των PrimaryCapsules με την προσθήκη ενός ακόμα συνελικτικού επιπέδου στη βασική αρχιτεκτονική. Τα μεγέθη ύψους και πλάτους των χαρτών χαρακτηριστικών είναι υπολογισμένα για το σύνολο δεδομένων MNIST.*Παράχθηκε από το Inkscape.*

Κλείνοντας την ενότητα αυτή, στο σχήμα 4.7 αναπαριστάνεται η πιο ελαφριά (light) εκδοχή του SOM-Caps. Ενώ φαινομενικά η εισαγωγή ενός επιπλέον συνελικτικού επιπέδου θα επιβάρυνε περισσότερο το υπολογιστικό κόστος, η ελάτωση των PrimaryCapsules που συμμετέχουν στον

[11] Βέβαια, οι προβλέψεις, εφόσον συγκροτούν πυκνές συστάδες, συν–διαμορφώνουν τα διανύσματα των καψουλών εξόδου (σε μικρότερο ή μεγαλύτερο βαθμό, ανάλογα με την παραμετροποίηση). Συνεπώς, οι ομοιότητες είναι και ένας βαθμός συμφωνίας των ψήφων μεταξύ τους. Περισσότερα σχετικά στην επόμενη ενότητα.

αλγόριθμο δρομολόγησης βασισμένο στο SOM συνολικά, μειώνει σημαντικά τον χρόνο εκπαίδευσης και πρόβλεψης του δικτύου.

4.4.2 Αλγόριθμος Δρομολόγησης

Σε αυτήν την ενότητα παρουσιάζουμε την οικογένεια αλγορίθμων που αναπτύξαμε και βασίζονται στον SOM. Όπως προαναφέραμε, η κεντρική ιδέα είναι η χρήση του αλγορίθμου για την εύρεση των συστάδων στον πολυδιάστατο χώρο των ψήφων. Φυσικά, ανάλογα με την παραμετροποίηση, η συμπεριφορά του αλγορίθμου μπορεί να αλλάξει σημαντικά. Παρόλα αυτά, με μια τέτοια υλοποίηση μας δίνεται η δυνατότητα της επιλεκτικής χαλάρωσης ορισμένων υποθέσεων των νευρωνικών δικτύων από κάψουλες και η παρατήρηση των αποτελεσμάτων αυτής.

Όπως είδαμε στην ενότητα της αρχιτεκτονικής του δικτύου, το κυρίαρχο χαρακτηριστικό που την διαφοροποιεί από τις άλλες μεθόδους εντοπίζεται στην έξοδο του αλγορίθμου. Αναλυτικότερα, οι DigitCaps, στις περισσότερες παραμετροποιήσεις, αποτελούν παράμετροι του δικτύου που ενημερώνονται σε επίπεδο δέσμης (batch). Συνεπώς, δεν έχουμε ένα ξεχωριστό σύνολο από κάψουλες εξόδου σε κάθε παράδειγμα εισόδου (instance). Ένα τέτοιο γνώρισμα συνεπάγεται ότι τα διανύσματα DigitCaps δεν ενσωματώνουν τα μεταβαλλόμενα χαρακτηριστικά του εκάστοτε στιγμιοτύπου (equi–variant).

Αντίθετα με την έξοδο DigitCaps, η έξοδος η οποία χαρακτηρίζει την κάθε εικόνα εισόδου είναι το διάνυσμα με τις ομοιότητες (similarities). Ποιοτικά, αυτό περιέχει τον μέσο βαθμό συμφωνίας των ψήφων με τα διανύσματα αναπαράστασης των DigitCaps. Αν θεωρήσουμε ότι τα διανύσματα των καψουλών εξόδου παραμένουν σταθερά, μεγάλος βαθμός συμφωνίας με μια κάψουλα εξόδου συνεπάγεται ότι οι ψήφοι συμφωνούν στο διάνυσμα που χρησιμοποιείται για την αναπαράσταση της συγκεκριμένης κλάσης που η κάψουλα–γονέας αναπαριστά.

Φυσικά, υπό διαφορετικές παραμετροποιήσεις (π.χ. μέγεθος δέσμης $= 1$) ο αλγόριθμος μπορεί να εξάγει διαφορετικά DigitCaps για κάθε εικόνα εισόδου. Επίσης, με μικρή τροποποίηση, έχει την ίδια δυνατότητα και για μέγεθος δέσμης μεγαλύτερο της μονάδας. Παρόλα αυτά, κάτι τέτοιο είχε δοκιμαστεί και οδηγούσε σε απαγορευτικούς χρόνους εκπαίδευσης αφού για κάθε δείγμα έτρεχε ένας (τροποποιημένος) αλγόριθμος SOM για την εκ νέου προσαρμογή των DigitCaps.

Σε κάθε περίπτωση, μια υπόθεση που παραμένει ακλόνητη είναι ο ανταγωνισμός μεταξύ των DigitCaps για το ποιά θα εξηγήσει την κάθε κάψουλα παιδί. Αυτή που εμφανίζει την μεγαλύτερη συμφωνία, κερδίζει την συγκεκριμένη κάψουλα και το διάνυσμά της τελευταίας θα ληφθεί υπ' όψιν στην ενημέρωση της DigitCap. Στις παραγράφους που ακολουθούν θα γίνουν ακόμα πιο σαφείς οι ποικίλες πτυχές του αλγορίθμου με την περιγραφή των υπερπαραμέτρων που τον ελέγχουν.

Υπερπαράμετροι Αλγορίθμου

Ο αλγόριθμος, λόγω της φύσης του, ενσωματώνει μια πληθώρα υπερπαραμέτρων για τον πειραματισμό με τις διάφορες υποθέσεις των νευρωνικών δικτύων από κάψουλες. Για αυτό, πριν γίνει η φορμαλιστική περιγραφή του αλγορίθμου κρίνεται σκόπιμη η περιγραφή της κάθε παραμέτρου που ο χρήστης μπορεί να τροποποιήσει (εξωτερικά, χωρίς την γνώση του κώδικα) καθώς και την επίδραση που αυτές έχουν στη διαμόρφωση του αλγορίθμου. Σημειώνουμε ότι αυτή η πληροφορία, μαζί με τον αναλυτικό αλγόριθμο καταγράφονται και στο παράρτημα Γ'.

Οι υπερπαράμετροι που μπορεί να ρυθμίσει κανείς κατά την εκτέλεση του αρχείου[12] είναι οι εξής:

non reduced votes

Αυτή η παράμετρος, όταν τίθεται ενεργή, αίρει τον περιορισμό του να αντιστοιχούν ξεχωριστοί πίνακες μετασχηματισμού (W_{ij}^{L}) για κάθε κάψουλα επιπέδου $(L+1)$. Με άλλα λόγια, όταν εισάγεται ως όρισμα στην κλήση του εκτελέσιμου αρχείου, παράγονται τόσοι μετασχηματισμοί της εκάστοτε κάψουλας C_i^L όσοι ορίζονται από μια άλλη υπερπαράμετρο, την m (και όχι όσες είναι οι κάψουλες του επόμενου επιπέδου). Έτσι, κάθε κάψουλα C_j^{L+1} έχει την ίδια όψη (view) των ψήφων. Σε τελική ανάλυση, αφήνεται στον αλγόριθμο ανταγωνιστικής μάθησης (τον αλγόριθμο SOM) να διαχωρίσει τις ψήφους και να τις συνδέσει με κάποια κλάση. Προφανώς, όταν η παράμετρος *nonreducedvotes* είναι ενεργή ισχύει ότι η μεταβλητή του αλγορίθμου *reducedvotes* δεν είναι ενεργή (False).

m

Η παράμετρος αυτή ρυθμίζει τον αριθμό των πινάκων μετασχηματισμού, για κάθε κάψουλα C_i^L (εφόσον η προηγούμενη παράμετρος είναι ενεργή). Έτσι, προκύπτουν m ψήφοι για κάθε C_i^L. Συνολικά λοιπόν, κάθε DigitCap «βλέπει» $n^L * m$ ψήφους.

radical

Η παράμετρος αυτή επηρεάζει άμεσα τον μηχανισμό ενημέρωσης των C^{L+1}. Όταν είναι ενεργή, τότε οι κάψουλες γονείς προκύπτουν σαν ένας μέσος όρος των ψήφων που μπορούν και εξηγούν καλύτερα (των ψήφων που κέρδισαν). Αυτό, διαφέρει από τον κλασικό αλγόριθμο SOM και την δική μας υλοποίηση χωρίς την συγκεκριμένη παράμετρο ενεργή όπου η ενημέρωση των κόμβων γίνεται με την πρόσθεση της διαφοράς τους με το datapoint που εξηγούν καλύτερα.[13]

SOM learning rate

Ο ρυθμός μάθησης του αλγορίθμου SOM-Based Routing. Καθορίζει το βάρος των ενημερώσεων.

routing iterations

Καθορίζει τον αριθμό των ενημερώσεων που θα γίνουν στα διανύσματα των DigitCaps όταν ο αλγόριθμος εκπαίδευσης έχει τροφοδοτηθεί με μια δέσμη παραδειγμάτων (batch). Όσο πιο πολλές ενημερώσεις εκτελούνται και όσο πιο μικρό είναι το μέγεθος της δέσμης τόσο πιο πολύ τα DigitCaps θα περιέχουν ιδιότητες που αφορούν τις συγκεκριμένες αναπαραστάσεις των αντικειμένων εισόδου (π.χ. προσανατολισμός, χρώμα κτλ.). Στην αντίθετη περίπτωση όπου οι ενημερώσεις είναι λίγες, ο ρυθμός μάθησης SOM μικρός και το μέγεθος της δέσμης μεγάλο, τα διανύσματα DigitCaps διαμορφώνονται έτσι ώστε να αναπαριστούν γενικά χαρακτηριστικά των κλάσεων που παραμένουν αμετάβλητα μέσα στα παραδείγματα μιας κλάσης (instance–invariant characteristics). Είναι επιτρεπτή και η τιμή 0. Σε αυτή τη περίπτωση, τα βάρη εκπαιδεύονται προκειμένου να παράγουν ψήφους που να εμφανίζουν μεγάλη ομοιότητα με το (αμετάβλητο πλέον) διάνυσμα της κάψουλας που αφορά την σωστή

[12] Ο κώδικας για όλες τις μεθόδους είναι αναρτημένος σε αυτή την ιστοσελίδα.

[13] Φυσικά, σε κάθε περίπτωση, η μετρική που χρησιμοποιείται για την επιλογή του νικητή διαφέρει από τον αυθεντικό αλγόριθμο που παρουσιάσαμε στο 2.4. Εκεί χρησιμοποιείται η Ευκλείδεια απόσταση ενώ εμείς χρησιμοποιούμε το εσωτερικό γινόμενο.

κλάση. Κατά τον έλεγχο, επειδή δεν θέλουμε να μεταβάλλονται τα διανύσματα βαρών από κάψουλες, η τιμή αυτή τίθεται στο μηδέν.

thetas

Πρόκειται για μια λίστα με βάρη που καθορίζουν το μέγεθος της «γειτονιάς» του νικητή[14] αλλά και το βάρος ενημέρωσης των γειτόνων (συμπεριλαμβανομένου του βάρους ενημέρωσης του νικητή). Με άλλα λόγια, είναι μια διακριτή συνάρτηση γειτνίασης με όρισμα την πλευρική απόσταση (lateral distance) και έξοδο το βάρος ενημέρωσης. Σημαντική διαφορά με τον αυθεντικό αλγόριθμο SOM, ωστόσο, είναι στο πως ορίζουμε αυτήν την πλευρική απόσταση (lateral distance). Αν θεωρήσουμε ότι η κάψουλα C_j^{L+1} είναι το BMU για την κάψουλα C_i^L τότε η C_j^{L+1} που είχε την δεύτερη μεγαλύτερη συμφωνία (εσωτερικό γινόμενο) με τη συγκεκριμένη κάψουλα παιδί, θα έχει με το BMU πλευρική απόσταση ίση με 1. Προφανώς, μια τέτοια τοπολογική διάταξη δεν είναι σταθερή αλλά μεταβάλλεται από δέσμη σε δέσμη. Για αυτό λέμε ότι ο αλγόριθμός μας δεν διαθέτει κάποια προφανή τοπολογική διάταξη, ούτε και είναι ποτέ σκοπός η δημιουργία μιας τέτοιας. Σημειώνουμε ότι αν η λιστα μας περιέχει μόνο μια τιμή τότε ακολουθούμε την πολιτική «ο νικητής τα παίρνει όλα» (winner takes it all).

softmax

Αν θέσουμε αυτή τη παράμετρο ενεργή τότε η επιλογή των νικητών δεν είναι μια διαδικασία με μόνο δύο καταστάσεις. Αντιθέτως έχουμε μαλακούς νικητές (soft winners) που καθορίζονται από το βαθμό ομοιότητας που εμφανίζουν με την κάθε ψήφο. Συνεπώς, στη διεκδίκηση της κάψουλας C_i^L από τις C^{L+1}, όλες «κερδίζουν» και η συμβολή του διανύσματος C_i^L στην κάθε κάψουλα C_j^{L+1} θα είναι ανάλογη του βαθμού ομοιότητας μεταξύ τους. Φυσικά, σε μια τέτοια περίπτωση, δεν έχει νόημα η ύπαρξη της γειτονιάς και ο αλγόριθμος διαφοροποιείται αρκετά από τον αυθεντικό SOM.

tanh like

Η παράμετρος αυτή έχει παρόμοια επίδραση με την παράμετρο $softmax$ και είναι αμοιβαία αποκλεινόμενες (δεν επιτρέπεται και οι δύο να είναι ενεργές). Η διαφορά έγκειται ότι εφαρμόζεται επιπλέον μια οίσθηση και κλιμάκωση προκειμένου η σιγμοειδής συνάρτηση με είσοδο το βαθμό ομοιότητας και έξοδο το βάρος ενημέρωσης να γίνεται πιο απότομη και ο αλγόριθμος να τείνει (σχεδόν) στην επιλογή σκληρών νικητών (hard winners).

take into account win ratio

Σε περίπτωση που είναι ενεργή, πραγματοποιείται κλιμάκωση των ενημερώσεων που λαμβάνει μια κάψουλα C_j^{L+1} με βάση το πόσοστό των C_i^L που καλύτερα εξήγησε σε μια δέσμη δεδομένων. Στο σημείο αυτό υπενθυμίζεται ότι οι ενημερώσεις, για λόγους απόδοσης, γίνονται παράλληλα και ανά δέσμη (όχι ανά εικόνα εισόδου).

take into account similarity

Παρόμοια λειτουργία με την προηγούμενη παράμετρο, μόνο που ο συντελεστής κλιμάκωσης της ενημέρωσης για μια κάψουλα C_j^{L+1} σε αυτή την περίπτωση καθορίζεται από τον μέσο βαθμό ομοιότητας μεταξύ αυτής και των ψήφων που κέρδισε.

[14] Στο πλαίσιο του αλγορίθμου SOM ο νικητής ονομάζεται και BMU (Best Matching Unit).

norm type

Το είδος της συνάρτησης σύνθλιψης. Σε περίπτωση που είναι μηδέν εφαρμόζεται η κλασική συνάρτηση σύνθλιψης (squash). Σε περίπτωση που είναι μονάδα, χρησιμοποιείται $tanh$ normalization για την κανονικοποίηση των ψήφων και unit normalization για τις C^{L+1}.

normalize votes

Χρησιμοποιείται σε περίπτωση που επιθυμούμε να κανονικοποιήσυμε τις ψήφους V^L.

normalize d in loop

Αν η ρύθμιση αυτή είναι ενεργή, τότε οι κάψουλες C^{L+1} κανονικοποιούνται αμέσως μετά από κάθε ενημέρωση. Συνήθως, βοηθάει στην ευστάθεια του αλγορίθμου όταν έχουμε μεγάλο αριθμό επαναλήψεων (παράμετρος r).

Φορμαλιστική Παρουσίαση Αλγορίθμου

Στην παράγραφο αυτή παρουσιάζουμε σε μια τυπική μορφή τον αλγόριθμο που αναπτύξαμε στην τέταρτη μέθοδο. Τον αλγόριθμο δεν τον συνοδεύουμε με πολλά σχόλια, αφενώς διότι στο παράρτημα Γ′ έχουμε αναπτύξει εννιά σελίδες περιγράφοντας τον εν λόγω αλγόριθμο και αφετέρου διότι οι μέχρι τώρα περιγραφές επαρκούν για την ποιοτική κατανόησή του.

Algorithm 15 Αλγόριθμος Δρομολόγησης Βασισμένος στον SOM (SOM-Based Routing)

Input PrimaryCaps $C^L \in \Re^{B \times n^L \times d^L}$

Output SimsOut $C^{L+1} \in \Re^{B \times n^{L+1}}$

Trainable Parameters If $reduced\_votes == True$: $W^L \in \Re^{n^L \times d^L \times d^{L+1} \times n^{L+1}}$

Trainable Parameters If $reduced\_votes == False$: $W^L \in \Re^{n^L \times d^L \times d^{L+1} \times m^L}$

Non-Trainable Parameters $C^{L+1} \in \Re^{n^{L+1} \times d^{L+1}}$

Hyperparameters $reduced\_votes, m, normalize\_votes, norm\_type, r, radical, \Theta,$ $take\_into\_account\_win\_ratio, take\_into\_account\_similarity, softmax, tanh\_like,$ $normalize\_d\_in\_loop, lr\_SOM$

procedure SOM-Based Routing(C^L) ▷ Input: $C^L \in \Re^{B \times n^L \times d^L}$

  Initialize:

    $C^{L+1} \leftarrow UniformRandom([-1, +1])$

    ▷ $C^{L+1} \in \Re^{n^{L+1} \times d^{L+1}}$

  if $reduced\_votes$ **then**

    $\forall b \in [1, B], \forall i \in \Omega_L, \forall j \in \Omega_{L+1} : V^L_{bij:} \leftarrow C^L_{bi:} \times W^L_{i::j}$

  else

    $\forall b \in [1, B], \forall i \in \Omega_L, \forall j \in \Omega_{L+1} : preV^L_{b(i+n^L*(j_m-1))d^{L+1}} \leftarrow C^L_{bi::} \times W^L_{i::j_m}$

    $\forall j \in \Omega_{L+1} : V^L_{::j:} \leftarrow preV^L$ ▷ Copy $n^{L+1} times.$

  end if

  ▷ $V^L \in \Re^{B \times \underline{n^L}, n^{L+1}, d^{L+1}}$

  ▷ where $\underline{n}^L = n^L$ OR $n^L * m$, depending on the value of $reduced\_votes$

  if $normalize\_votes$ **then**

    if $norm\_type == 0$ **then**

      $\forall b \in [1, B], \forall i \in \underline{\Omega}_L, \forall j \in \Omega_{L+1} : V^L_{bij} \leftarrow Squash(V^L_{bij})$

    else

      $\forall b \in [1, B], \forall i \in \underline{\Omega}_L, \forall j \in \Omega_{L+1} : V^L_{bij} \leftarrow tanh\left\|V^L_{bij}\right\| \frac{V^L_{bij}}{\|V^L_{bij}\|}$

    end if

  end if

  if $take\_into\_account\_win\_ratio$ **then**

    Initialize:

      $WinCount \leftarrow zeros^{n^{L+1}}$

  end if

for r iterations **do**
  Initialize:
    $SU^L \leftarrow zeros^{B \times \underline{n}^L \times n^{L+1} \times d^{L+1}}$
  if *radical* **then**
    $D^L \leftarrow V^L$ ▷ D^L stores differences
  else
    $\forall b \in [1, B], \forall i \in \underline{\Omega}_L : D^L_{bi::} \leftarrow V^L_{bi} - C^L$
  end if
  ▷ $D^L \in \Re^{B \times \underline{n}^L \times n^{L+1} \times d^{L+1}}$
  Initialize:
    $M^L \leftarrow ones^{B \times \underline{n}^L \times n^{L+1} \times d^{L+1}}$
  IterateOver$\Theta(\Theta)$ ▷ Inner loop, check pseudocode below.
  ▷ Make update dense.
  $U^L \leftarrow \frac{\sum_b^B \sum_i^{\underline{n}^L} SU^L_{bi}}{B * \underline{n}^L}$ ▷ $U^L \in \Re^{n^{L+1} \times d^{L+1}}$
  if not *take_into_account_win_ratio* **then**
    if not *radical* **then**
      $C^{L+1} \leftarrow C^{L+1} + U^L$
    else if *radical* **then**
      $C^{L+1} \leftarrow \frac{C^{L+1} * (r-1) + U^L}{r}$ ▷ Moving Average
    end if
  else
    $WinRatio^L \leftarrow \frac{WinCount^L}{\underline{n}^L * B}$
    $SoftWinRatio^L \leftarrow softmax(WinRatio^L)$
    $\forall j \in \Omega_{L+1} : U^L_j \leftarrow SoftWinRatio^L_j * U^L_j$
    if not *radical* **then**
      $C^{L+1} \leftarrow C^{L+1} + U^L$
    else if *radical* **then**
      $C^{L+1} \leftarrow \frac{C^{L+1} * (r-1) + U^L}{r}$
    end if
  end if
  if *normalize_in_d_loop* **then**
    if *norm_type* $== 0$ **then**
      $\forall j \in \Omega_{L+1} : C^{L+1}_j \leftarrow Squash(C^{L+1}_j)$
    else
      $\forall j \in \Omega_{L+1} : C^{L+1}_j \leftarrow tanh(\left\| C^{L+1}_j \right\|) \frac{C^{L+1}_j}{\|C^{L+1}_j\|}$
    end if
  end if
end for
$\forall b \in [1, B], \forall i \in \underline{\Omega}_L : FinalSims^{L+1}_{bi} \leftarrow \sum_k^{d^{L+1}} [V^L_{bi} * C^{L+1}]_{bik}$
▷ $FinalSims \in \Re^{B \times \underline{n}^L \times n^{L+1}}$
$\forall b \in [1, B] : SimsOut^{L+1}_b \leftarrow \frac{\sum_i^{\underline{n}^L} FinalSims^{L+1}_{bi}}{\underline{n}^L}$
return $SimsOut^{L+1}$ ▷ $SimsOut^{L+1} \in \Re^{B, n^{L+1}}$, optional output: C^{L+1}
end procedure

procedure ITERATEOVERΘ(Θ)
  for all $\theta \in \Theta$ **do**
    $\forall b \in [1, B], \forall i \in \underline{\Omega}_L : Sims^L_{bi} \leftarrow \sum_k^{d^L+1}[(V^L_{bi} * M^L_{bi}) * C^{L+1}]_{bi:k}$
    ▷ $Sims^L \in \Re^{B \times \underline{n}^L, n^{L+1}}$
    if not *softmax* nor *tanh_like* **then**
      $\forall b \in [1, B], \forall i \in \underline{\Omega}_L : Jwinners^L_{bi} \leftarrow \underset{j}{\text{argmax}}(Sims^L_{bi})$
      ▷ $Jwinners^L \in \Re^{B \times \underline{n}^L}$
      $\forall b \in [1, B] : Mwinners^L_b \leftarrow OneHot(Jwinners^L_b; n^{L+1})$
    else if *softmax* **then**
      $\forall b \in [1, B] : Mwinners^L_b \leftarrow \underset{across j}{SoftMax}(Sims^L_b)$
    else if *tanh_like* **then**
      $\forall b \in [1, B] : Mwinners^L_b \leftarrow (\underset{across j}{SoftMax}(Sims^L_b) - 0.5) * 2$
    end if
    ▷ $Mwinners^L \in \Re^{B \times \underline{n}^L \times n^{L+1}}$
    if not *take_into_account_similarity* nor *take_into_account_win_ratio* **then**
      $\forall b \in [1, B], \forall i \in \underline{\Omega}_L : Usparse^L_{bi} \leftarrow (Mwinners^L_{bi} \times D^L_{bi}) * lr_S OM * \theta$
    else if *take_into_account_similarity* **then**
      $SimsSparse^L \leftarrow Sims^L * Mwinners^L$
      $\forall b \in [1, B], \forall i \in \underline{\Omega}_L : Usparse^L_{bi} \leftarrow (SimsSparse^L_{bi} \times D^L_{bi}) * lr_S OM$
    end if
    ▷ $Usparse \in \Re^{B \times \underline{n}^L \times n^{L+1} \times d^{L+1}}$
    if *take_into_account_win_ratio* **then**
      $WinCountPartial^L \leftarrow \sum_i^{\underline{n}^L} \sum_b^B Mwinners^L_{bi}$
      $WinCount \leftarrow WinCount + WinCountPartial$ ▷ $WinCount \in \Re^{n^{L+1}}$
    end if
    if not *softmax* nor *tanh_like* **then**
      $\forall b \in [1, B], \forall i \in \underline{\Omega}_L, \forall j \in \Omega_{L+1} : M^L_{bij:} \leftarrow M^L_{bij:} - Mwinners^L_{bij}$
      ▷ $M^L \in \Re^{B \times \underline{n}^L \times n^{L+1} \times d^{L+1}}$
    end if
    ▷ Aggregate updates across θ.
    $SU^L \leftarrow SU^L + Usparse^L$ ▷ $SU^L \in \Re^{B \times \underline{n}^L \times n^{L+1} \times d^{L+1}}$
  end for
end procedure

4.4.3 Λοιπά Στοιχεία Υλοποίησης

Στον αλγόριθμο που περιγράψαμε χρησιμοποιείται ο βελτιστοποιητής nAdam ενώ οι περισσότερες παράμετροι εκπαίδευσης (π.χ. ρυθμός μάθησης) είναι παραμετροποιήσημες (ακόμα και η αρχιτεκτονική του πρώτου τμήματος του νευρωνικού δικτύου). Να αναφέρουμε ακόμα ότι δεν χρησιμοποιείται κάποιος προγραματισμός για την σταδιακή ελάττωση του ρυθμού εκπαίδευσης.

Και σε αυτήν την αρχιτεκτονική έχουμε την δυνατότητα ανακατασκευής της εικόνας εισόδου χρησιμοποιώντας δύο ξεχωριστά είδη αποκωδικοποιητών (ακριβώς ίδια με αυτά της προηγούμενης μεθόδου). Συγκεκριμένα, χρησιμοποιούμε έναν αποκωδικοποιητή όπως αυτόν της πρώτης μεθόδου αλλά και έναν αποκωδικοποιητή βασιζόμενο σε επίπεδα αποσυνέλιξης. Όπως είναι φυσικό, οι αποκωδικοποιητές μεταβάλλονται ελαφρώς ανάλογα με τα δεδομένα εικόνων που καλούνται να ανακατασκευάσουν[15].

Τέλος, αναφορικά με την συνάρτηση σφάλματος ειδικά για το σύνολο δεδομένων υποχρεωτικά πρέπει να χρησιμοποιηθεί MarginLoss το (όπως το ορίσαμε σε προηγούμενες μεθόδους) ενώ για τα υπόλοιπα (όπου έχουμε μια κλάση πρόβλεψης) υπάρχει η δυνατότητα επιλογής ανάμεσα σε αυτό και στο CategoricalCrossEntropyLoss.

[15]Η συγκεκριμένη μέθοδος μπορεί με την αλλαγή μιας παραμέτρου να δοκιμαστεί στα σύνολα δεδομένων MNIST, Cifar10, FashionMNIST, MultiMNIST και smallNORB

Κεφάλαιο 5

Πειραματική Μελέτη

Στην παρούσα ενότητα παρουσιάζουμε τα αποτελέσματα των πειραμάτων που διενεργήθηκαν στην κάθε μια οικογένεια αλγορίθμων. Παρόλα αυτά, δεν είναι σκοπός η βελτιστοποίηση της απόδοσης (όπως καταγράφεται από τις επιλεγμένες μετρικές) για κάθε αλγόριθμο. Όπως έχουμε αναφέρει, ο σκοπός της παρούσας διπλωματικής είναι διττός: αφενός επιθυμούμε να εξερευνήσουμε την επίδραση της χαλάρωσης ορισμένων υποθέσεων των νευρωνικών δικτύων με κάψουλες στην απόδοσή τους (μέθοδος 1) και αφετέρου να επιλύσουμε το πρόβλημα της κλιμακωσιμότητας προτείνοντας έναν αποδοτικό αλγόριθμο δρομολόγησης (μέθοδος 3). Η τέταρτη, πολυδύναμη, μέθοδος, βρίσκεται στο μεταίχμιο αυτών με τη δυνατότητα τόσο για επιλεκτική χαλάρωση των περιορισμών της εν λόγω τεχνολογίας όσο για μερική βελτίωση του χρόνου εκπαίδευσης. Τέλος, η δεύτερη μέθοδος, λόγω της μεγάλης υπολογιστικής πολυπλοκότητάς της που δεν επέτρεπε τον εκτενή πειραματισμό, περιορίζεται σε δύο σύνολα δεδομένων και αναλαμβάνει τον σκοπό της σύγκρισης με τις υπόλοιπες μεθόδους μας.

Όπως γίνεται αντιληπτό, κυρίως για τους αλγορίθμους της μεθόδου 1 αλλά και για αυτούς της μεθόδου 4, μας ενδιαφέρει περισσότερο η σχετική επίδοση μεταξύ αυτών αφού αυτή φανερώνει αν οι περιορισμοί που επιβάλλονται τελικά συμβάλουν στην καλύτερη γενίκευση του δικτύου ή όχι. Για τον λόγο αυτό, δίνουμε έμφαση στη σύγκριση των επιδόσεων μεταξύ των αλγορίθμων που ανήκουν στην ίδια οικογένεια (ίδια μέθοδο). Βέβαια, για λόγους πληρότητας, επιλέγουμε τους καλύτερους αλγορίθμους από την κάθε μέθοδο και τους συγκρίνουμε στο τέλος του παρόντος κεφαλαίου.

Κρίνεται σκόπιμο να αναφερθεί πως σε κάθε περίπτωση, η πειραματική μας μελέτη δεν είναι πλήρης. Ορισμένοι αλγόριθμοι που αναπτύξαμε (και ειδικά αυτοί που απαντώνται στην πολυδύναμη τέταρτη μέθοδο) διαμορφώνονται από μια πληθώρα υπερπαραμέτρων όπου η κάθε μια επιδρά καθοριστικά στην απόδοσή τους. Επιπρόσθετα, οι μειωμένοι υπολογιστικοί πόροι που διαθέτουμε καθιστούν τη διαδικασία πειραματισμού ιδιαιτέρως χρονοβόρα. Συνεπώς, είναι χρήσιμο να έχουμε υπ΄ όψιν ότι οι επιδώσεις που καταγράφουμε πιθανότατα να επιδέχονται βελτίωση.

Το παρόν κεφάλαιο ακολουθεί την εξής διάρθρωση:

1. Αρχικά γίνεται μια σύντομη παρουσίαση των συνόλων δεδομένων που χρησιμοποιούμε, των μετρικών αλλά και της πλατφόρμας πειραματισμού.

2. Έπειτα ακολουθούν οι πειραματικές μελέτες της κάθε μεθόδου ξεχωριστά. Τα περιεχόμενα της κάθε τέτοιας υποενότητας διαφέρουν σημαντικά ανάλογα με τον σκοπό της εκάστοτε μεθόδου. Σε γενικές γραμμές όμως, περιλαμβάνουν τα αποτελέσματα που προκύπτουν από την αναζήτηση ικανοποιητικών υπερπαραμέτρων στα διάφορα σύνολα δεδομένων και τη

σύγκριση των αλγορίθμων μεταξύ τους.

3. Τέλος, επιλέγουμε τους καλύτερους αλγορίθμους από διαφορετικές μεθόδους και τους συγκρίνουμε μεταξύ τους αλλά και με άλλες υλοποιήσεις νευρωνικών δικτύων με κάψουλες που συναντώνται στη βιβλιογραφία.

5.1 Πλατφόρμα Διεξαγωγής Πειραμάτων, Μετρικές και Σύνολα Δεδομένων

Στην ενότητα αυτή κάνουμε λόγο για τα αμετάβλητα στοιχεία που συνθέτουν το περιβάλλον της πειραματικής μας μελέτης. Αυτά περιλαμβάνουν το υπολογιστικό σύστημα στο οποίο διενεργήθηκαν όλα τα πειράματα, τις μετρικές που χρησιμοποιήθηκαν για την εκτίμηση της επίδοσης και τα σύνολα δεδομένων με τα οποία οι αλγόριθμοι τροφοδοτήθηκαν.

5.1.1 Πειραματική Πλατφόρμα

Όλα τα πειράματα εκτελέστηκαν τοπικά, στον προσωπικό υπολογιστή (PC). Επειδή το σύστημα εκτέλεσης των πειραμάτων επηρεάζει τους χρόνους εκπαίδευσης, κρίνεται σκόπιμη η περιγραφή των δυνατοτήτων του υπολογιστικού μας συστήματος. Από πλευράς υλικού (hardware) λοιπόν, τα χαρακτηριστικά του είναι τα εξής:

- 16GB DDR4 RAM
- AMD Ryzen 9 3900X, 12-core CPU
- Nvidia RTX 2070 super GPU

Ένα πολύ καλό εργαλείο για την αντιστοίχηση της υπολογιστικής δυνατότητας μιας συσκευής σε μια μετρική για την αντιπαραβολή με τις δυνατότητες άλλων συστημάτων είναι το *ai-benchmark*. Τρέχοντας το σχετικό πρόγραμμα εκτίμησης δυνατοτήτων, λάβαμε, μεταξύ άλλων τα παρακάτω αποτελέσματα:

- *Device Inference Score: 12150*
- *Device Training Score: 12115*
- *Device AI Score: 24265*

Βέβαια, το περιβάλλον πειραματισμού απαρτίζεται και από τις εκδόσεις των πακέτων λογισμικού που είναι εγκατεστημένες στο σύστημα. Για αυτό τον σκοπό, στην ιστοσελίδα όπου είναι αναρτημένος ο κώδικας, έχουμε καταγράψει όλα τα απαιτούμενα πακέτα λογισμικού. Ενδεικτικά, τα βασικότερα στοιχεία λογισμικού είναι τα εξής:

- Platform: *Linux, Release: 5.15.0-48-generic, Version: 20.04.1-Ubuntu*
- CUDA version: *11.0*
- cudnn version: *8*
- Tensorflow version: *2.4.1*

- Pytorch version: *1.7.1+cu110*

Για να διευκολύνουμε την αναπαραγωγή πειραμάτων, ενσωματώνουμε και ένα εικονικό περιβάλλον (Docker). Το σχετικό αρχείο (DockerFileGenericGPU) δημιουργεί ένα εικονικό περιβάλλον με τα απαραίτητα πακέτα λογισμικού (dependences) που χρειάζεται να είναι εγκατεστημένα για την εκτέλεση των προγραμμάτων.

5.1.2 Μετρικές Επίδοσης

Λόγω του ερευνητικού χαρακτήρα της παρούσας διπλωματικής, μας ενδιαφέρει κυρίως η σύγκριση των μεθόδων μας με τις υπόλοιπες σχετικές μεθόδους που απαντώνται στη βιβλιογραφία. Για τον σκοπό αυτό και με δεδομένο ότι όλες οι εργασίες είναι εργασίες ταξινόμησης, η μετρική της ακρίβειας (accuracy) είναι η πλέον κατάλληλη μετρική. Η μετρική αυτή ορίζεται από τον λόγο των σωστά ταξινομημένων προβλέψεων προς το σύνολο των προβλέψεων. Με μαθηματικούς όρους δηλαδή, έχουμε:

$$Accuracy = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{5.1}$$

Ειδικά για το σύνολο δεδομένων MultiMNIST όπου έχουμε δύο προβλέψεις για κάθε δείγμα εισόδου, η μετρική μας ονομάζεται πολλαπλή–Ακρίβεια (multi-Accuracy). Παρόλα αυτά, σύμφωνα με τον παραπάνω ορισμό (που εστιάζει στον αριθμό των προβλέψεων και όχι των δειγμάτων εισόδου) οι δύο μετρικές είναι ταυτόσημες.

Συχνά, αντί για την ακρίβεια, χρησιμοποιείται σαν μετρική το ποσοστιαίο σφάλμα ελέγχου (test error rate%). Το σφάλμα αυτό είναι αντιστρόφως ανάλογο της ακρίβειας. Δηλαδή, χαμηλότερη τιμή ποσοστιαίου σφάλματος σηματοδοτεί καλύτερη επίδοση. Στην πραγματικότητα, δεν αποτελεί μια ξεχωριστή μετρική αφού ισχύει ότι

$$test\_error\_rate = 100 - Accuracy * 100\%.$$

Όπως έχει γίνει αντιληπτό από το πρώτο κεφάλαιο της εργασίας, μας ενδιαφέρει να εντοπίσουμε μια αρχιτεκτονική με μικρό υπολογιστικό κόστος. Δύο μετρικές που φανερώνουν την ποσότητα αυτή είναι ο (σχετικός) μέσος χρόνος εκπαίδευσης ενός συνόλου δέσμης και ο αριθμός των εκπαιδευόμενων παραμέτρων ενός μοντέλου (των παραμέτρων δηλαδή που ρυθμίζονται με τον αλγόριθμο της οπισθοδιάδοσης σφάλματος). Συνεπώς, εκτός από την ακρίβεια, οι δύο αυτές μετρικές προστίθενται στα κριτήρια επιλογής των καλύτερων αλγορίθμων.

Μια ακόμα μετρική υπολογιστικού κόστους είναι ο αριθμός των υπολογισμών το δευτερόλεπτο με στοιχεία κινητής υποδιαστολής (FLOating Point operations per second - FLOPs). Μάλιστα, πρόκειται για την πιο αντικειμενική μετρική υπολογιστικού κόστους. Ιδιαίτερη προσπάθεια καταβλήθηκε για την απόκτηση των τιμών της μετρικής αυτής για όλες τις μεθόδους μας. Μάλιστα, για τις υλοποιήσεις των έργων [76] και [47] οι τιμές ακριβείας που υπολογίσαμε (για το σύνολο δεδομένων smallNORB) δεν είναι δυνατόν να βρεθούν με μια βιβλιογραφική αναζήτηση.

5.1.3 Σύνολα Δεδομένων

Στις περισσότερες μεθόδους μας χρησιμοποιούμε όλα τα σχετικά σύνολα δεδομένων με τα οποία δοκιμάζονται συνήθως οι αρχιτεκτονικές νευρωνικών δικτύων με κάψουλες. Γεγονός, που μας επιτρέπει να εξετάσουμε αν τηρούνται οι χαρακτηριστικές ιδιότητες της εν λόγω τεχνολογίας από τα νέα μοντέλα που αναπτύξαμε. Τα σύνολα δεδομένων με τα οποία καταπιανόμαστε είναι τα MNIST [126], FashionMNIST [127], CIFAR10 [128], MultiMNIST [76] και smallNORB [96]. Για λόγους πληρότητας, στον πίνακα 5.1 φαίνονται τα μοντέλα επιβλεπόμενης μάθησης που επιτυγχάνουν τη μέγιστη ακρίβεια για το κάθε σύνολο δεδομένων (όπως καταγράφονται στην ιστοσελίδα τον Οκτώβριο του 2022).

Dataset	Method	Test Error (%)
MNIST	Heterogeneous ensemble with simple CNN [129]	0.09
FashionMNIST	Fine-Tuning DARTS [130]	3.09
CIFAR-10	ViT-H/14 [81]	0.5
MultiMNIST	RUCapsNet [98]	1.8
smallNORB	Heinsen Routing [131]	0.90

Πίνακας 5.1: Πίνακας που συγκεντρώνει το καλύτερο μοντέλο και την απόδοσή του, για κάθε σύνολο δεδομένων.

Περιγραφή Συνόλων Δεδομένων smallNORB και multiMNIST

Τα δύο σύνολα δεδομένων είναι λιγότερο δημοφιλή στην ακαδημαϊκή κοινότητα για αυτό αφιερώνουμε αυτή την παράγραφο για την περιγραφή τους. Τα δύο σύνολα εξετάζουν την ιδιότητα των νευρωνικών δικτύων από κάψουλες να γενικεύουν σε νέες οπτικές γωνίες και να εξηγούν εικόνες με σημαντική επικάλυψη αντίστοιχα.

Αναφορικά με το σύνολο δεδομένων smallNORB, αυτό περιέχει στερεο–οπτικές εικόνες που απεικονίζουν 50 αντικείμενα (παιχνίδια) τα οποία ανήκουν σε 5 κλάσεις: τετράποδα ζώα, ανθρώπινες φιγούρες, αεροπλάνα, φορτηγά και αυτοκίνητα. Η κάθε κλάση εκπροσωπείται από δέκα φυσικά αντικείμενα, τα μισά εξ΄ αυτών βρίσκονται στο σύνολο εκπαίδευσης. Το σύνολο δεδομένων δημιουργήθηκε από τη στερεοσκοπική λήψη αυτών των αντικειμένων από δύο κάμερες υπό 6 διαφορετικές συνθήκες φωτισμού, 9 διαφορετικά υψόμετρα προβολής (γωνίες 30∘έως 70∘ με βήμα 5∘) και 18 διαφορετικά αζιμούθια (γωνίες 0∘ έως 340∘ με βήμα 20∘). Έτσι, τόσο το σύνολο εκπαίδευσης όσο και το σύνολο ελέγχου αποτελούνται από 24,300 ζευγάρια στερεο–οπτικών εικόνων το καθένα. Οι αλγόριθμοί μας, δέχονται κάθε ζεύγος εικόνων σαν ένα δείγμα και στόχος τους είναι να προβλέψουν το απεικονιζόμενο αντικείμενο.

Το σύνολο δεδομένων multiMNIST δημιουργήθηκε από τους Hinton et al. κατα τη συγγραφή του έργου [76]. Στην πραγματικότητα, ο αριθμός των δειγμάτων και τα περιεχόμενα του συνόλου δεν είναι προκαθορισμένα αφού η κάθε υλοποίηση κατασκευάζει δυναμικά το σύνολο αυτό λαμβάνοντας εικόνες από το σύνολο δεδομένων MNIST (αυτός είναι και ένας λόγος του περιορισμένου πειραματισμού με αυτό το σύνολο δεδομένων στη βιβλιογραφία). Ουσιαστικά, αποτελείται από εικόνες που απεικονίζουν στο ίδιο πλαίσιο, δύο επικαλυπτόμενα ψηφία (με ποσοστό επικάλυψης περίπου 80%). Όπως είναι λογικό, κάθε ένα τέτοιο δείγμα συνοδεύεται από δύο ετικέτες: μια για κάθε απεικονιζόμενο ψηφίο.

5.2 Πειραματική Μελέτη Μεθόδου 1

Στην παρούσα ενότητα πραγματοποιούνται πειράματα στους τέσσερεις αλγορίθμους της μεθόδου 1 για να εκτιμηθεί η επίδοσή τους στα σύνολα δεδομένων MNIST [126], FashionMNIST [127], CIFAR10 [128] και smallNORB [96]. Υπενθυμίζεται ότι όλοι οι αλγόριθμοι της μεθόδου 1 έχουν ίδια αρχιτεκτονική με αυτήν που χρησιμοποιείται στο έργο [76] αλλά διαφέρουν στον αλγόριθμο δρομολόγησης ο οποίος βαθμιαία, από τον πρώτο αλγόριθμο στον τέταρτο γίνεται απλούστερος. Συνεπώς, ο ρόλος των πειραμάτων της πρώτης μεθόδου δεν είναι απαραίτητα να προτείνει μια αντικατάσταση του δυναμικού αλγορίθμου δρομολόγησης. Βασικότερος σκοπός είναι να εξεταστεί η επίδραση που έχουν ορισμένες απλουστεύσεις του αλγορίθμου δυναμικής δρομολόγησης (αλγόριθμος 1) στη μάθηση (αλγόριθμοι 2 και 3) και την επίδοση του αλγορίθμου μετά τη χαλάρωση της υπόθεσης περί φιλτραρίσματος υψηλής, πολυδιάστατης συμφωνίας (αλγόριθμος 4).

Μεταξύ των τεσσάρων αλγορίθμων που εξετάζονται, συμπεριλαμβάνεται και ο αυθεντικός αλγόριθμος της δυναμικής δρομολόγησης με συμφωνία (αλγόριθμος 1). Αν και αυτός δεν αποτελεί μια δική μας μέθοδο, συμπεριλαμβάνεται στα πειράματα για να διευκολύνει την ισότιμη σύγκριση με τους άλλους αλγορίθμους της μεθόδου. Επίσης, οι υψηλές επιδόσεις που εντοπίζονται στον αλγόριθμο 1 πιστοποιούν την ορθή υλοποίηση της αρχιτεκτονικής του δικτύου που μοιράζονται και οι άλλες τρεις παραλλαγές.

Η διάρθρωση των πειραμάτων του κεφαλαίου έχει ως εξής: Για τα σύνολα δεδομένων MNIST και CIFAR-10 κάνουμε πειράματα για τον εντοπισμό των υπερπαραμέτρων των μεθόδων (ρυθμός μάθησης κ.ο.κ.) με τα καλύτερα αποτελέσματα (χρησιμοποιώντας λίγες εποχές). Χρησιμοποιούμε αυτά τα δύο σύνολα καθώς έχουν μεγάλη ετερογένεια μεταξύ τους. Αφού βρεθούν αυτές οι υπερπαράμετροι, εμβαθύνουμε στο κάθε σύνολο δεδομένων ξεχωριστά εκπαιδεύοντας τους τέσσερεις αλγορίθμους με περισσότερες εποχές και με τις αντίστοιχες παραμετροποιήσεις τους ενώ έπειτα, τους συγκρίνουμε. Στην προ–τελευταία υπο-ενότητα παρουσιάζουμε συγκεντρωτικά τα αποτελέσματα και αναφέρουμε τις παρατηρήσεις μας σχετικά με την επίδραση των υποθέσεών μας στις επιδόσεις. Στην τελευταία υπο-ενότητα παρουσιάζουμε τα συμπεράσματά μας από ορισμένα ειδικά πειράματα που αποσκοπούν να διερευνήσουν την εσωτερική λειτουργία των αλγορίθμων μας.

Σχετικά με τις λεπτομέρειες υλοποίησης να αναφέρουμε ότι η συνάρτηση σφάλματος είναι αυτή που έχει οριστεί στην ενότητα 4.1.4. Δηλαδή, είναι η συνάρτηση σφάλματος περιθωρίου (Margin Loss) με την προσθήκη του μέσου τετραγωνικού σφάλματος (κλιμακωμένου κατά 0.0005) στην περίπτωση που χρησιμοποιείται αποκωδικοποιητής. Ο βελτιστοποιητής (optimizer) που χρησιμοποιούμε είναι ο Adam (adaptive moment estimation) ενώ χρησιμοποιούμε και σύστημα ελάττωσης του ρυθμού μάθησης όταν δεν παρατηρείται μείωση του σφάλματος στη διάρκεια ενός προκαθορισμένου αριθμού εποχών (learning rate scheduler with reduce on plateau and patience). Σε αρκετές περιπτώσεις, είναι απαραίτητη η χρήση ενός συνόλου επικύρωσης (validation set) το οποίο πάντα αποτελεί το 10% του συνόλου δεδομένων εκπαίδευσης. Αξίζει να σημειωθεί επίσης ότι για την αποφυγή της υπερπροσαρμογής (overfitting) αποθηκεύουμε το μοντέλο με το μικρότερο σφάλμα κατά τη διάρκεια της εκπαίδευσης και χρησιμοποιούμε αυτό στο σύνολο ελέγχου.

Τέλος, σχετικά με την προεπεξεργασία των συνόλων δεδομένων, αυτή είναι όσο πιο πιστή γίνεται στο έργο [76]. Πιο συγκεκριμένα, για το κάθε σύνολο δεδομένων έχουμε:

MNIST Κανονικοποίηση και ολίσθηση στον κατακόρυφο και οριζόντιο άξονα κατά 2 εικονοστοιχεία το μέγιστο (με zero padding). Μέγεθος εισόδου: $[1 \times 28 \times 28]$[1].

FashionMNIST Κανονικοποίηση μόνο. Μέγεθος εισόδου: $[1 \times 28 \times 28]$

CIFAR10 Κανικοποίηση για το καθένα από τα τρία κανάλια και περικοπή παραθύρων μεγέθους 28×28 (το αρχικό ύψος και πλάτος είναι 32×32). Μέγεθος εισόδου $[3 \times 28 \times 28]$.

smallNORB Κλιμάκωση σε μέγεθος 48×48 (από το αρχικό μέγεθος που είναι 96×96), κάνουμε περικοπή σε ένα παράθυρο μεγέθους 32×32 και τέλος, προσθέτουμε τυχαία φωτεινότητα και αντίθεση στο σύνολο εκπαίδευσης (παρόμοια με το έργο [47]). Επειδή το σύνολο αποτελείται από στερεοπτικές εικόνες, τις στοιβάζουμε δημιουργώντας μια εικόνα με δύο κανάλια. Μέγεθος εισόδου μετά από προεπεξεργασία: $[2 \times 32 \times 32]$.

Σημειώνουμε ότι κατά τη δημιουργία του συνόλου ελέγχου, το παράθυρο περικοπής είναι κεντραρισμένο στην εικόνα ενώ κατά τη δημιουργία του συνόλου εκπαίδευσης το παράθυρο αυτό είναι τυχαίο για το κάθε δείγμα.

5.2.1 Εύρεση Βέλτιστων Υπερπαραμέτρων

Στα πρώτα πειράματα της μεθόδου 1 διερευνούμε τις υπερπαραμέτρους των αλγορίθμων που εμφανίζουν τα καλύτερα αποτελέσματα. Οι υπερπαράμετροι αυτοί αφορούν τον ρυθμό μάθησης (Learning Rate - Lr), το σύνολο δέσμης (Batch Size - Bs), τη χρήση αποκωδικοποιητή (reconstruction) ή όχι και τον αριθμό επαναλήψεων (Routing Iterations - r) - με εξαίρεση τον αλγόριθμο Max Routing ο οποίος δεν είναι επαναληπτικός. Ο αριθμός των εποχών είναι μόλις 30 καθώς τα πειράματα είναι πολλά και οι υπολογιστικοί πόροι περιορισμένοι. Παρόλα αυτά, ο αριθμός των εποχών είναι αρκετός για να βρεθεί μια καλή παραμετροποίηση για τον κάθε αλγόριθμο. Στην επόμενη υποενότητα, θα εκπαιδεύσουμε επιλεκτικά τα μοντέλα για περισσότερες επαναλήψεις.

Σύνολο Δεδομένων MNIST

Τα πρώτα πειράματα που έγιναν στο σύνολο δεδομένων MNIST αφορούν τον κλασικό δυναμικό αλγόριθμο δρομολόγησης με συμφωνία (αλγόριθμος 1). Τα αποτελέσματα των πειραμάτων παρατίθενται στον πίνακα 5.2.

Από τα ανωτέρω πειράματα μπορούμε να εξάγουμε τα εξής συμπεράσματα:

1. Αναφορικά με το μέγεθος της δέσμης, κατά την αύξησή του, παρατηρείται σημαντική βελτίωση στην περίπτωση που δε χρησιμοποιείται αποκωδικοποιητής. Στην περίπτωση που χρησιμοποιείται δίκτυο αποκωδικοποιητή, έχει ένα μικρό προβάδισμα η ρύθμιση με το μικρότερο μέγεθος δέσμης. Αυτό πιθανότατα οφείλεται στον μικρότερο αριθμό των βημάτων οπισθοδιάδοσης σφάλματος που συμβαίνει κατά τον διπλασιασμό του μεγέθους δέσμης σε

[1]Οι αλγόριθμοι της μεθόδου 1 είναι ανεπτυγμένοι στη γλώσσα Pytorch και συνεπώς, χρησιμοποιείται μια αναπαράσταση δεδομένων τύπου channels-first

Experiment	Batch Size	Routing Iter.	Learning Rate	Recon.	Test Error (%)
Batch Size	32	3	0.001	no	0.51
	64	3	0.001	no	0.37[2]
	32	3	0.001	yes	0.37
	64	3	0.001	yes	0.41
Routing Iter.	64	1	0.001	no	0.44
	64	1	0.001	yes	0.45
	64	2	0.001	no	0.48
	64	2	0.001	yes	0.35
	64	3	0.001	no	0.37
	64	3	0.001	yes	0.41
Learning Rate	64	3	0.0005	yes	**0.33**
	64	3	0.001	yes	0.37
	64	3	0.002	yes	0.51
	64	3	0.01	yes	90.0*

Πίνακας 5.2: Πειράματα στο MNIST για την αναζήτηση υπερπαραμέτρων στον αλγόριθμο δυναμικής δρομολόγησης με συμφωνία (αλγόριθμος 1) για 30 εποχές. Ο αστερίσκος (*) συμβολίζει αστάθεια.

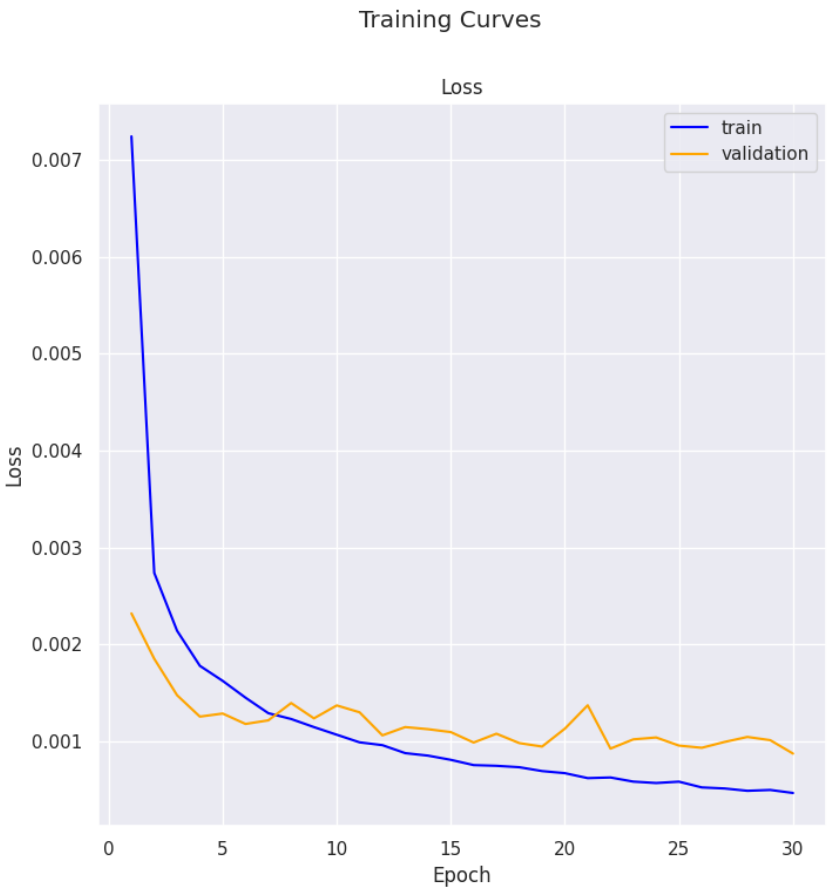
συνδυασμό τόσο με τη σημαντική αύξηση των παραμέτρων λόγω της προσθήκης αποκωδικοποιητή όσο και με την κλιμάκωση του σφάλματος ανακατασκευής. Συνεπώς, μπορούμε με ασφάλεια να υποθέσουμε ότι εν γένει, ένα σύνολο δέσμης μεγέθους 64 οδηγεί σε καλύτερη εκπαίδευση από ένα σύνολο δέσμης 32 (αρκεί να συνοδεύεται από μεγάλο αριθμό εποχών).

2. Αναφορικά με τον αριθμό των επαναλήψεων, κατά μέσο όρο παρατηρείται βελτίωση με την αύξηση των επαναλήψεων του αλγορίθμου. Όταν ο αριθμός επαναλήψεων είναι ίσος με τη μονάδα, ουσιαστικά δεν έχουμε κάποιον δυναμικό αλγόριθμο επανάληψης αφού η κάθε κάψουλα DigitCap προκύπτει από τον μέσο όρο των ψήφων της. Στηριζόμενοι και στα πειράματά του έργου [76], οι τρεις επαναλήψεις είναι η ιδανική παράμετρος για τη μέγιστη επίδοση. Η αύξηση του σφάλματος που παρατηρείται στην περίπτωση της χρήσης ανακατασκευής ίσως να οφείλεται στον μικρό αριθμό εποχών που έχει σαν αποτέλεσμα να μην προλαβαίνει να εκπαιδευτεί πλήρως το δίκτυο αποκωδικοποιητή[3].

3. Στα πειράματα που αφορούν τον ρυθμό μάθησης φαίνεται ότι ένας μικρότερος ρυθμός μάθησης οδηγεί σε καλύτερη εκπαίδευση. Βέβαια, κατά τη διάρκεια της εκπαίδευσης πολλές φορές ο ρυθμός μάθησης μειώθηκε κατά μία ή και δύο κλίμακες μεγέθους από τον learning rate scheduler. Συνεπώς, σε περισσότερες επαναλήψεις (epochs) ένας ρυθμός μάθησης με τιμή 0.001 δε θα αποτελεί πρόβλημα. Το ασφαλές συμπέρασμα λοιπόν είναι ότι ο ρυθμός μάθησης δεν ωφελεί να είναι μεγαλύτερος του 0.001.

Στην εικόνα 5.1 παρατίθεται το σφάλμα εκπαίδευσης (training loss) και επαλήθευσης (validation loss) για το μοντέλο με την καλύτερη επίδοση στον ανωτέρω πίνακα ($Bs = 64, lr = 0.0005, r = 3, Reconstruction = yes$). Παρατηρούμε ότι ακόμα και στην 30κοστή εποχή, το

[3]Άλλωστε, στα γραφήματα του σφάλματος επικύρωσης που παράγουμε κατά την εκπαίδευση, είναι προφανές ότι το σφάλμα μειώνεται ακόμα στις 30 εποχές.



Σχήμα 5.1: Στο σχήμα παρατηρούμε τις γραφικές παραστάσεις του σφάλματος εκπαίδευσης και ελέγχου (validation ή test) κατά τη διάρκεια των 30 εποχών για το μοντέλο με την καλύτερη επίδοση στον πίνακα 5.2.

σφάλμα επαλήθευσης μειώνεται.

Ο επόμενος σε σειρά αλγόριθμος είναι ο Argmax Scaled Routing. Ο αλγόριθμος αυτός κάνει την υπόθεση ότι μια ψήφος είναι αρκετή για τη διαμόρφωση μιας κάψουλας γονέα. Συνεπώς, για τη διαμόρφωση ολόκληρου του επιπέδου DigitCaps αρκεί να επιλεγούν (με κριτήριο τα βάρη δρομολόγησης) τόσες ψήφοι εκπρόσωποι όσος είναι και ο αριθμός των κλάσεων. Στον πίνακα 5.3 παρατίθενται τα αποτελέσματα των πειραμάτων. Προφανώς, η δοκιμασία για 1 επανάληψη αλγορίθμου δρομολόγησης δεν υπάρχει καθώς σε αυτή την περίπτωση, όλα τα βάρη δρομολόγησης είναι ίσα (λόγω αρχικοποίησης).

Από τα ανωτέρω πειράματα μπορούμε να εξάγουμε τα εξής συμπεράσματα:

1. Σχετικά με το πρώτο πείραμα, αν και οι διαφορές δεν είναι μεγάλες, φαίνεται να βοηθάει η αύξηση του μεγέθους δέσμης και η προσθήκη αποκωδικοποιητή κατά την εκπαίδευση.

2. Σε αντίθεση με τον κλασικό αλγόριθμο δυναμικής δρομολόγησης, ο βέλτιστος αριθμός επαναλήψεων είναι 2. Όπως αποδεικνύεται στα ειδικά πειράματα (ενότητα 5.3), όσο αυξάνονται οι επαναλήψεις τόσο τα βάρη δρομολόγησης λαμβάνουν ακραίες τιμές (είτε 0 αν δρομολογούν ψήφους σε κάψουλες γονείς που δεν ανήκουν στη σωστή κλάση είτε 1 για κάψουλες

Experiment	Batch Size	Routing Iter.	Learning Rate	Recon.	Test Error (%)
Batch Size	32	3	0.001	no	0.54
	64	3	0.001	no	0.53
	32	3	0.001	yes	0.52
	64	3	0.001	yes	0.51
Routing Iter.	64	2	0.001	no	0.53
	64	2	0.001	yes	**0.39**
	64	3	0.001	no	0.53
	64	3	0.001	yes	0.51
Learning Rate	64	3	0.0005	yes	0.45
	64	3	0.001	yes	0.51
	64	3	0.002	yes	0.57
	64	3	0.01	yes	0.59

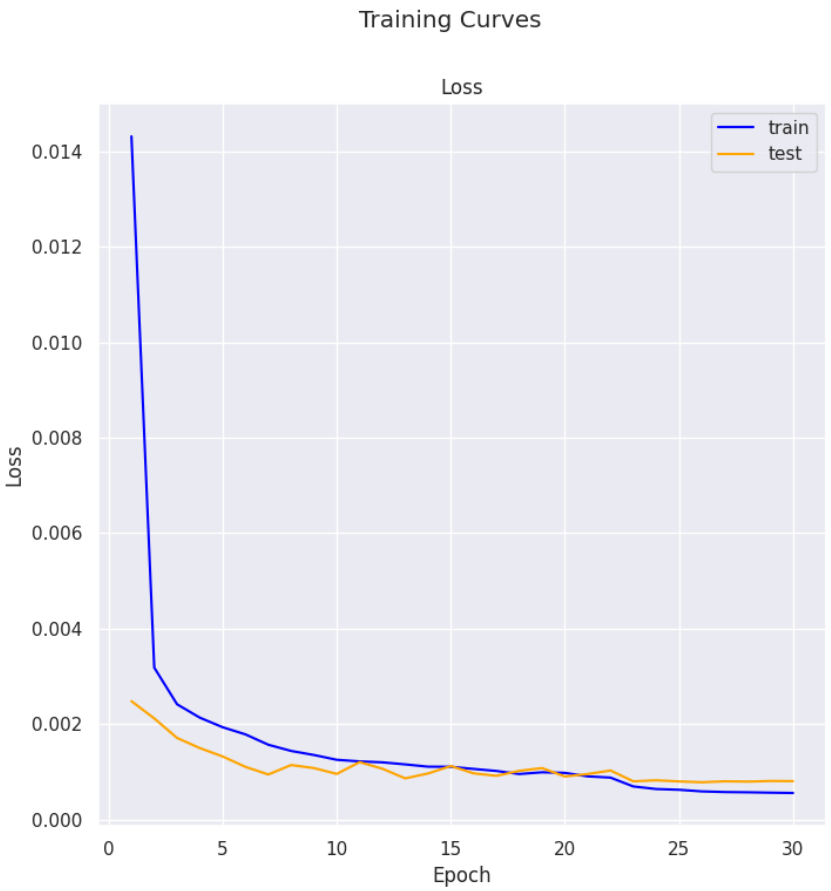
Πίνακας 5.3: Πειράματα στο MNIST για την αναζήτηση υπερπαραμέτρων στον αλγόριθμο Argmax Scaled Routing (αλγόριθμος 2) για 30 εποχές.

παιδιά που ανήκουν στη σωστή κλάση στόχο). Συνεπώς, στις πρώτες επαναλήψεις, μπορεί τα βάρη δρομολόγησης να μην έχουν κορεστεί σε ακραίες τιμές αλλά από νωρίς τα βάρη που αντιστοιχούν στη σωστή κλάση έχουν τις μεγαλύτερες τιμές (αποδεικνύεται στην ενότητα 5.3). Το γεγονός όμως ότι η αύξηση των επαναλήψεων προκαλεί μείωση της επίδοσης στον συγκεκριμένο αλγόριθμο μας προδιαθέτει ότι υπάρχουν πολλές περιπτώσεις σύγχυσης όπου ενώ έτεινε προς τη σωστή κλάση, σε αργότερη επανάληψη άλλαξαν ριζικά τα βάρη ευνοώντας κάποια λανθασμένη κλάση. Η συγκεκριμένη περίπτωση περιγράφεται έμπρακτα στο δεύτερο σχήμα του έργου [47]. Επίσης, είναι πιθανό να οφείλεται στο ότι υπάρχουν κάψουλες που έχουν παράξει ψήφους με μικρή συμφωνία για την κάψουλα πρόβλεψης. Κατά τον κορεσμό τους, αυτές θα αποκτήσουν βάρη δρομολόγησης ίσα με τη μονάδα προς άλλες κάψουλες.

3. Σε αντιστοιχία με τις παραμέτρους της προηγούμενης μεθόδου, βλέπουμε ότι ένας χαμηλότερος ρυθμός μάθησης βοηθάει τη διαδικασία εκπαίδευσης.

Στην εικόνα 5.2 παρατίθεται το σφάλμα εκπαίδευσης (training loss) και επαλήθευσης (validation loss) για το μοντέλο με την καλύτερη επίδοση στον ανωτέρω πίνακα ($Bs = 64, lr = 0.001, r = 2, Reconstruction = yes$). Παρατηρούμε ότι ακόμα και στη 30κοστή εποχή, το σφάλμα επαλήθευσης μειώνεται.

Ο επόμενος σε σειρά αλγόριθμος είναι ο Argmax Routing. Και αυτός ο αλγόριθμος κάνει την υπόθεση ότι μια ψήφος είναι αρκετή για τη διαμόρφωση μιας κάψουλας γονέα. Συνεπώς, και πάλι, για τη διαμόρφωση ολόκληρου του επιπέδου DigitCaps αρκεί να επιλεγούν (με κριτήριο τα βάρη δρομολόγησης) τόσες ψήφοι εκπρόσωποι όσος είναι και ο αριθμός των κλάσεων. Υπενθυμίζεται ότι η διαφορά με τον Argmax Scaled Routing είναι ότι δεν πραγματοποιείται κλιμάκωση των εκπροσώπων με βάση το αντίστοιχο βάρος δρομολόγησης. Αυτό πάει ένα βήμα παραπέρα την υπόθεσή μας αφού πλέον, το μήκος των DigitCaps που καθορίζει την κλάση πρόβλεψης δεν εξαρτάται από το δυναμικό αλγόριθμο αλλά από τις επιλεγμένες ψήφους. Με απλά λόγια, ερχόμαστε ακόμα πιο κοντά στον ισχυρισμό ότι τα βάρη δρομολόγησης διαμορφώνονται αποκλειστικά από



Σχήμα 5.2: Στο σχήμα παρατηρούμε τις γραφικές παραστάσεις του σφάλματος εκπαίδευσης και ελέγχου (validation ή test) κατά τη διάρκεια των 30 εποχών για το μοντέλο με την καλύτερη επίδοση στον πίνακα 5.3.

τις ψήφους με μεγάλο μέτρο (χωρίς να δίνεται ιδιαίτερη έμφαση στην πολυδιάστατη σύμπτωση).

Στον πίνακα 5.4 παρατίθενται τα αποτελέσματα των σχετικών πειραμάτων. Προφανώς, όπως και στον προηγούμενο αλγόριθμο, η δοκιμασία για 1 επανάληψη αλγορίθμου δρομολόγησης δεν υπάρχει καθώς σε αυτή την περίπτωση, όλα τα βάρη δρομολόγησης είναι ίσα (λόγω αρχικοποίησης).

Από τα πειράματα του πίνακα μπορούμε να εξάγουμε τα εξής συμπεράσματα:

1. Παρατηρούμε ότι η αύξηση του μεγέθους δέσμης συμβάλλει στην εκπαίδευση.

2. Η αύξηση του ρυθμού επαναλήψεων βοηθάει στον συγκεκριμένο αλγόριθμο την επίδοση. Γεγονός που ενισχύει την τελευταία υπόθεση της αντίστοιχης παρατήρησης του προηγούμενου αλγορίθμου. Δηλαδή, το γεγονός ότι ο αλγόριθμος της μη κλιμάκωσης των ψήφων εκπροσώπων παρουσιάζει βελτίωση στις 3 επαναλήψεις συνεπάγεται ότι για κάθε κάψουλα γονέα επιλέγονται οι σωστές ψήφοι. Παρόλα αυτά, όταν τα βάρη δρομολόγησης φτάνουν στον κορεσμό, ίσως υπάρχουν και άλλες (λίγες) κάψουλες των οποίων οι ψήφοι δε συμφωνούν με τις άλλες ψήφους της σωστής κλάσης πρόβλεψης (π.χ. επειδή αναπαριστούν μέρη αντικειμένων που δεν εντοπίζονται στη συγκεκριμένη εικόνα εισόδου) αλλά τυχαίνει

Experiment	Batch Size	Routing Iter.	Learning Rate	Recon.	Test Error (%)
Batch Size	32	3	0.001	no	0.87
	64	3	0.001	no	**0.68**
	32	3	0.001	yes	0.90
	64	3	0.001	yes	0.83
Routing Iter.	64	2	0.001	no	0.88
	64	2	0.001	yes	0.81
	64	3	0.001	no	0.68
	64	3	0.001	yes	0.83
Learning Rate	64	3	0.0005	yes	0.99
	64	3	0.001	yes	0.83
	64	3	0.002	yes	0.75
	64	3	0.01	yes	0.80

Πίνακας 5.4: Πειράματα στο MNIST για την αναζήτηση υπερπαραμέτρων στον αλγόριθμο Argmax Routing (αλγόριθμος 3) για 30 εποχές.

να συμφωνούν με λίγες ψήφους μιας άλλης κλάσης. Υπό αυτές τις συνθήκες, το αντίστοιχο βάρος δρομολόγησης της κάψουλας θα κορεστεί στη μονάδα πολύ γρήγορα[4].

3. Σε αυτόν τον αλγόριθμο απαιτείται λίγο μεγαλύτερος ρυθμός μάθησης (ή περισσότερες εποχές).

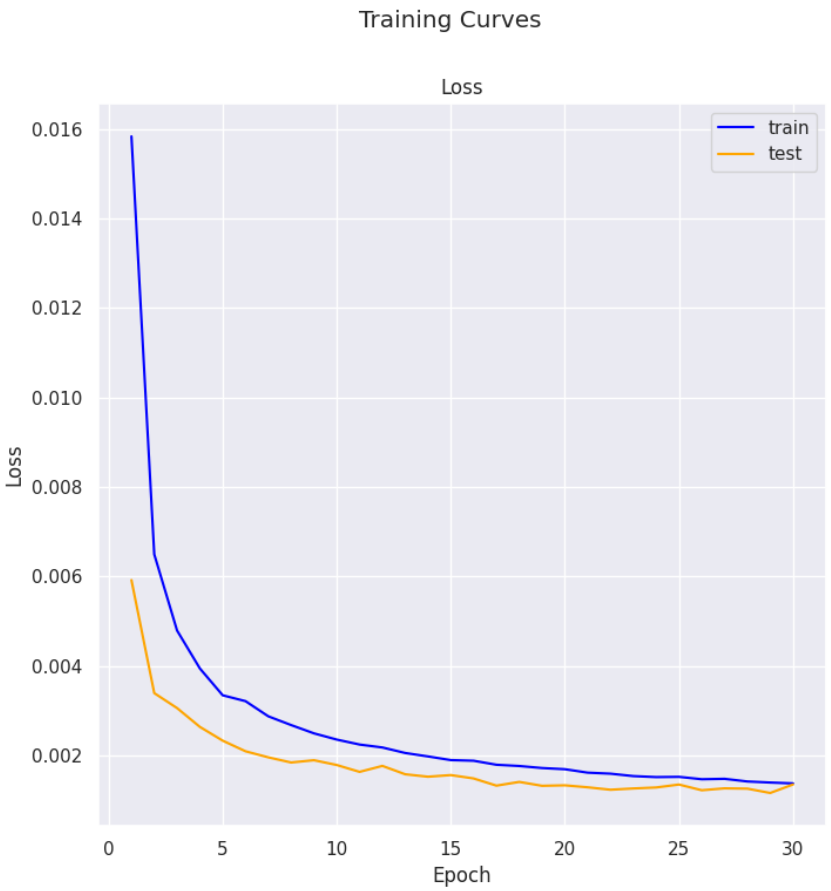
Στην εικόνα 5.3 παρατίθεται το σφάλμα εκπαίδευσης (training loss) και επαλήθευσης (validation loss) για το μοντέλο με την καλύτερη επίδοση στον ανωτέρω πίνακα ($Bs = 64, lr = 0.001, r = 3, Reconstruction = no$). Παρατηρούμε ότι ακόμα και στη 30κοστή εποχή, το σφάλμα επαλήθευσης μειώνεται. Πιθανότατα, το γεγονός αυτό είναι η εξήγηση της αύξησης του σφάλματος με τη χρήση του αποκωδικοποιητή.

Σε κάθε διάταξη, η μέθοδος αυτή παρουσιάζει χειρότερα αποτελέσματα. Συνεπώς, απορρίπτονται οι υποθέσεις σχετικά με τη μη συνεισφορά των βαρών δρομολόγησης. Τέλος, το πόρισμα αυτό είναι ήδη αναμφίβολο (και λόγω παλαιότερων πειραμάτων σε 100 εποχές) και άρα, δεν έχει νόημα η δοκιμή του αλγορίθμου αυτού σε όλα τα σύνολα δεδομένων.

Ο τελευταίος αλγόριθμος που εξετάζουμε σε αυτή τη μέθοδο είναι ο Max Routing. Υπενθυμίζεται ότι ο αλγόριθμος αυτός απορρίπτει τελείως τον δυναμικό αλγόριθμο δρομολόγησης και μαζί του, την υπόθεση των νευρωνικών δικτύων από κάψουλες περί φιλτραρίσματος με πολυδιάστατη σύμπτωση. Υποστηρίζει ότι στον δυναμικό αλγόριθμο δρομολόγησης, η ψήφος (για κάθε DigitCap) με το μεγαλύτερο μέτρο διαδραματίζει τον μεγαλύτερο ρόλο στη διαμόρφωση του μέσου διανύσματος s_j και με αυτόν τον τρόπο, απλά προσελκύει όσες ψήφους τυχαίνει να συμφωνούν μαζί της. Συνεπώς, υποστηρίζουμε με αυτή τη μέθοδο ότι δεν πρόκειται για ένα φιλτράρισμα συμφωνίας αλλά για μια αδικαιολόγητα περίπλοκη επιλογή μεγίστου διανύσματος.

Στον πίνακα 5.5 παρατίθενται τα αποτελέσματα των σχετικών πειραμάτων. Προφανώς, η δοκιμασία για τον αριθμό των επαναλήψεων δεν έχει νόημα καθώς ο αλγόριθμος δεν είναι επαναληπτικός.

[4]Στην ενότητα 5.3 παρουσιάζεται μια τέτοια περίπτωση.



Σχήμα 5.3: Στο σχήμα παρατηρούμε τις γραφικές παραστάσεις του σφάλματος εκπαίδευσης και ελέγχου (validation ή test) κατά τη διάρκεια των 30 εποχών για το μοντέλο που ακολουθεί τον αλγόριθμο Argmax Routing που έχει τις υπερπαραμέτρους με την καλύτερη επίδοση, όπως φαίνεται στον πίνακα 5.4.

Από τα πειράματα του πίνακα μπορούμε να εξάγουμε τα εξής συμπεράσματα:

1. Από τα αποτελέσματα, δεν μπορούμε να κρίνουμε ποια είναι η καλύτερη επιλογή μεγέθους δέσμης. Εκτιμούμε ότι σε αυτό τον αριθμό εποχών για τον συγκεκριμένο αλγόριθμο, μικρότερος αριθμός δέσμης είναι προτιμητέος.

2. Σε αυτόν τον αλγόριθμο η τιμή 0.001 για τον ρυθμό μάθησης φαίνεται να είναι ιδανική.

Για άλλη μια φορά, στην εικόνα 5.4 παρατίθεται το σφάλμα εκπαίδευσης (training loss) και επαλήθευσης (validation loss) για το μοντέλο με την καλύτερη επίδοση στον ανωτέρω πίνακα ($Bs = 32, lr = 0.001, Reconstruction = yes$). Παρατηρούμε ότι ακόμα και στη 30κοστή εποχή, το σφάλμα επαλήθευσης μειώνεται ακόμα.

Σε κάθε διάταξη, η μέθοδος αυτή παρουσιάζει χειρότερα αποτελέσματα σε σχέση με τις πρώτες δύο. Συνεπώς, αφενός δε θα γίνει εκτενή μελέτη αυτού σε όλα τα σύνολα δεδομένων και αφετέρου φαίνεται ότι ο αλγόριθμος δρομολόγησης πραγματικά συμβάλλει στην εκπαίδευση. Περισσότερα σχετικά πειράματα βρίσκονται στην ενότητα 5.3.

Experiment	Batch Size	Learning Rate	Recon.	Test Error (%)
Batch Size	32	0.001	no	0.92
	64	0.001	no	0.81
	32	0.001	yes	**0.72**
	64	0.001	yes	0.89
Learning Rate	64	0.0005	yes	1.00
	64	0.001	yes	0.89
	64	0.002	yes	0.90
	64	0.01	yes	0.95

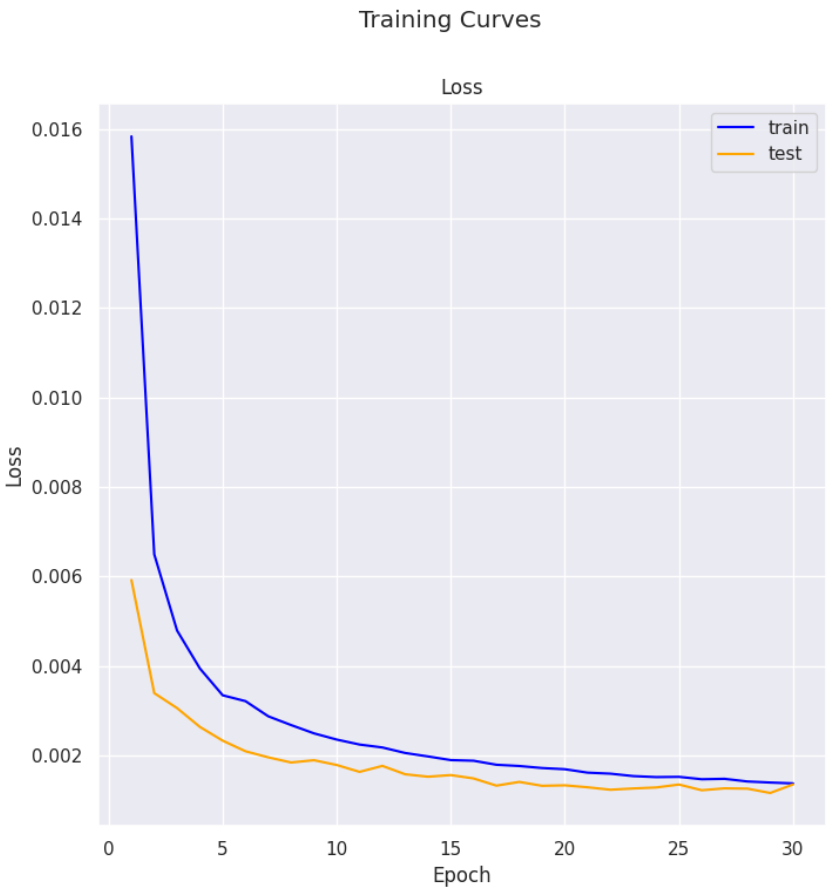
Πίνακας 5.5: Πειράματα στο MNIST για την αναζήτηση υπερπαραμέτρων στον αλγόριθμο Max Routing (αλγόριθμος 4) για 30 εποχές.

Σύνολο Δεδομένων CIFAR10

Αντίστοιχα με τα πειράματα για το σύνολο MNIST, παρατίθενται παρακάτω τα πειράματα για το πιο σύνθετο σύνολο δεδομένων CIFAR10. Οι λίγες εποχές που χρησιμοποιούνται για το συγκεκριμένο σύνολο σε συνδυασμό με την τάση των νευρωνικών δικτύων από κάψουλες να εξηγούν το οτιδήποτε βρίσκεται στην εικόνα (ακόμα και το παρασκήνιο) οδηγούν σε χαμηλές επιδόσεις. Σημειώνουμε ότι κατά αντιστοιχία με το έργο [76], χρησιμοποιήθηκε μια επιπλέον κλάση none-of-the-above για να δρομολογούν εκεί την ψήφο τους κάψουλες που αναπαριστούν αντικείμενα που δεν εμφανίζονται στην εικόνα. Επίσης, ο αριθμός των τύπων από κάψουλες αυξήθηκε από 32 σε 64 (όπως και στο έργο [76]).

Ο αριθμός των πειραμάτων είναι μικρότερος από αυτόν για το σύνολο MNIST. Αυτό διότι, αν και το σύνολο CIFAR10 έχει μεγάλη ετερογένεια με το προηγούμενο, ορισμένα πειράματα ποτέ δεν οδηγούν σε καλύτερα αποτελέσματα (ανεξαρτήτως συνόλου δεδομένων). Τα πειράματα αυτά έχουν αφαιρεθεί (όπως αυτό της δοκιμής του συνόλου δέσμης).

Τα πρώτα πειράματα για το σύνολο δεδομένων CIFAR10 αφορούν τον κλασικό δυναμικό αλγόριθμο δρομολόγησης με συμφωνία (αλγόριθμος 1). Τα αποτελέσματα των πειραμάτων παρατίθενται στον πίνακα 5.6.



Σχήμα 5.4: Στο σχήμα απεικονίζονται οι γραφικές παραστάσεις του σφάλματος εκπαίδευσης και ελέγχου (validation ή test) κατά τη διάρκεια των 30 εποχών για το καλύτερο μοντέλο που ακολουθεί τον αλγόριθμο Max Routing, όπως προκύπτει από τον πίνακα 5.5.

Experiment	Batch Size	Routing Iter.	Learning Rate	Recon.	Test Error (%)
Batch Size	32	3	0.0005	yes	**20.60**
	64	3	0.0005	yes	23.96
Routing Iter.	64	1	0.001	no	24.64
	64	1	0.001	yes	23.19
	64	2	0.001	no	29.25*
	64	2	0.001	yes	26.23*
	64	3	0.001	no	31.24*
	64	3	0.001	yes	29.15*
Learning Rate	64	3	0.0005	yes	23.96
	64	3	0.001	yes	29.15*

Πίνακας 5.6: Πειράματα στο CIFAR10 για την αναζήτηση υπερπαραμέτρων στον αλγόριθμο δυναμικής δρομολόγησης με συμφωνία (αλγόριθμος 1) για 30 εποχές. Οι αριθμοί με αστερίσκο αναφέρονται σε περιπτώσεις αστάθειας του αλγορίθμου εκπαίδευσης.

Από τα ανωτέρω πειράματα μπορούμε να εξάγουμε τα εξής συμπεράσματα:

- Αναφορικά με το μέγεθος της δέσμης, παρατηρήθηκε ότι χαμηλότερο μέγεθος (32) εξαλείφει την όποια αστάθεια στον αλγόριθμο εκπαίδευσης. Για αυτό και η καλύτερη επίδοση επιτυγχάνεται με αυτό.

- Αναφορικά με τον αριθμό των επαναλήψεων, παρατηρούμε ότι η αύξησή τους επιδεινώνει την αστάθεια. Το γεγονός ότι οι καλύτερες παραμετροποιήσεις είναι για 3 επαναλήψεις σε συνδυασμό με τα πειράματα στο προηγούμενο σύνολο δεδομένων υποδεικνύουν ότι αυτή είναι η βέλτιστη υπερπαράμετρος.

- Για άλλη μια φορά, χαμηλότερη τιμή ρυθμού μάθησης βελτιώνει την επίδοση και μειώνει την πιθανότητα αστάθειας.

Ο επόμενος αλγόριθμος είναι ο Argmax Scaled Routing. Στον πίνακα 5.7 παρατίθενται τα αποτελέσματα των πειραμάτων. Να επαναλάβουμε ότι και για αυτό το σύνολο δεδομένων η δοκιμασία για 1 επανάληψη αλγορίθμου δρομολόγησης δεν υπάρχει καθώς σε αυτή την περίπτωση, όλα τα βάρη δρομολόγησης είναι ίσα (λόγω αρχικοποίησης).

Experiment	Batch Size	Routing Iter.	Learning Rate	Recon.	Test Error (%)
Routing Iter.	64	2	0.001	no	28.10
	64	2	0.001	yes	**27.46**
	64	3	0.001	no	29.65
	64	3	0.001	yes	31.02
Learning Rate	64	3	0.0005	yes	29.06
	64	3	0.001	yes	31.02
	64	3	0.002	yes	31.60

Πίνακας 5.7: Πειράματα στο CIFAR10 για την αναζήτηση υπερπαραμέτρων στον αλγόριθμο Argmax Scaled Routing (αλγόριθμος 2) για 30 εποχές.

Τα ανωτέρω πειράματα, επιβεβαιώνουν τα συμπεράσματα των πειραμάτων στο σύνολο δεδομένων MNIST. Βλέπουμε ότι οι 2 επαναλήψεις είναι η βέλτιστη ρύθμιση για τον συγκεκριμένο αλγόριθμο ενώ επίσης, συνήθως βοηθάει η ελάττωση του ρυθμού μάθησης στην τιμή 0.0005. Σημειώνουμε ότι ο αλγόριθμος αυτός (όπως και όλοι οι αλγόριθμοι εκτός του προηγούμενου), ουδέποτε εμφάνισε την οποιαδήποτε αστάθεια κατά την εκπαίδευση.

Ο επόμενος σε σειρά αλγόριθμος είναι όπως και πριν ο Argmax Routing. Στον πίνακα 5.8 παρατίθενται τα αποτελέσματα των σχετικών πειραμάτων (με τη δοκιμή για 1 επανάληψη να μην έχει ουσία).

Experiment	Batch Size	Routing Iter.	Learning Rate	Recon.	Test Error (%)
Routing Iter.	64	2	0.001	no	38.96
	64	2	0.001	yes	37.96
	64	3	0.001	no	38.06
	64	3	0.001	yes	38.49
Learning Rate	64	3	0.0005	yes	38.97
	64	3	0.001	yes	38.49
	64	3	0.002	yes	**37.83**

Πίνακας 5.8: Πίνακας που περιέχει τα πειράματα που έγιναν στο σύνολο CIFAR10 για την αναζήτηση υπερπαραμέτρων στον αλγόριθμο Argmax Routing (αλγόριθμος 3) για 30 εποχές.

Για άλλη μια φορά επιβεβαιώνονται οι παρατηρήσεις που έγιναν στο σύνολο δεδομένων MNIST για τον αντίστοιχο αλγόριθμο. Φαίνεται ότι οι 3 επαναλήψεις οδηγούν σε καλύτερη επίδοση αφού σε αυτή τη ρύθμιση επιτυγχάνεται η καλύτερη επίδοση.

Κλείνοντας τα πειράματα για την εύρεση των βέλτιστων υπερπαραμέτρων, αναφέρουμε τα πειράματα που έγιναν στον αλγόριθμο Max Routing με το σύνολο δεδομένων CIFAR10. Στον πίνακα 5.9 παρατίθενται τα αποτελέσματα των σχετικών πειραμάτων. Προφανώς, όπως και στον προηγούμενο αλγόριθμο, η δοκιμασία για τον αριθμό των επαναλήψεων δεν έχει νόημα καθώς ο αλγόριθμος δεν είναι επαναληπτικός.

Experiment	Batch Size	Learning Rate	Recon.	Test Error (%)
	64	0.0005	yes	**37.19**
Learning Rate	64	0.002	yes	37.60

Πίνακας 5.9: Πειράματα στο CIFAR10 για την αναζήτηση υπερπαραμέτρων στον αλγόριθμο Max Routing (αλγόριθμος 4) για 30 εποχές.

Από τα δύο πειράματα αποδεικνύεται ότι ο μικρότερος ρυθμός μάθησης βοηθάει γενικότερα στην εκπαίδευση των αλγορίθμων μας.

5.2.2 Επιλεκτική Εμβάθυνση Πειραμάτων και Σύγκριση

Για τους δύο αλγορίθμους της μεθόδου 1 με την καλύτερη επίδοση (αλγόριθμοι 1 και 2) πραγματοποιήθηκαν εκτενέστερα πειράματα με περισσότερες εποχές, για όλα τα σύνολα δεδομένων και σε μεγαλύτερο μέγεθος δέσμης (εφόσον ήταν εφικτό). Τα νέα πειράματα αυτά, μαζί με τα παλαιότερα πειράματα από τις άλλες δύο μεθόδους (για 100 εποχές), τα παρουσιάζουμε συγκεντρωτικά στον πίνακα 5.10.

Το πιο βέβαιο συμπέρασμα που μπορούμε να εξάγουμε από τα ανωτέρω αποτελέσματα είναι ότι ο αλγόριθμός μας Argmax Scaled Routing δεν παρουσιάζει έντονες διαφορές στην επίδοση σε σχέση με τον κλασικό αλγόριθμο δρομολόγησης με συμφωνία. Αν εξαιρέσουμε τα σύνολα δεδομένων CIFAR10 και SmallNORB όπου ο αριθμός των εποχών δεν επαρκεί για τη σύγκλισή τους, στα υπόλοιπα σύνολα δεδομένων οι διαφορές στις επιδόσεις είναι μικρές.

Η παρόμοια ακρίβεια μεταξύ των δύο αλγορίθμων σημαίνει ότι η εξαγωγή ενός διανύσματος

Dataset	Algorithm	Bs	r	Lr	Recon.	Test Error (%)
MNIST	Classic	64	3	0.0005	yes	0.41
	Argmax Scaled	64	2	0.0005	yes	0.43
	Classic	128	3	0.0005	yes	**0.31***
	Argmax Scaled	128	2	0.001	yes	0.37*
	Classic	32	3	0.0005	yes	0.35
	Argmax Scaled	32	2	0.001	yes	0.42
	Argmax	64	3	0.001	yes	0.60
	Max	32	0	0.001	yes	0.75
FashionMNIST	Classic	64	3	0.0005	yes	**6.94**
	Argmax Scaled	64	2	0.0005	yes	7.00
CIFAR10	Classic	64	3	0.0005	yes	**21.64**
	Argmax Scaled	64	2	0.0005	yes	22.16
SmallNORB	Classic	64	3	0.0005	yes	17.01
	Argmax Scaled	64	2	0.0005	yes	**16.49**

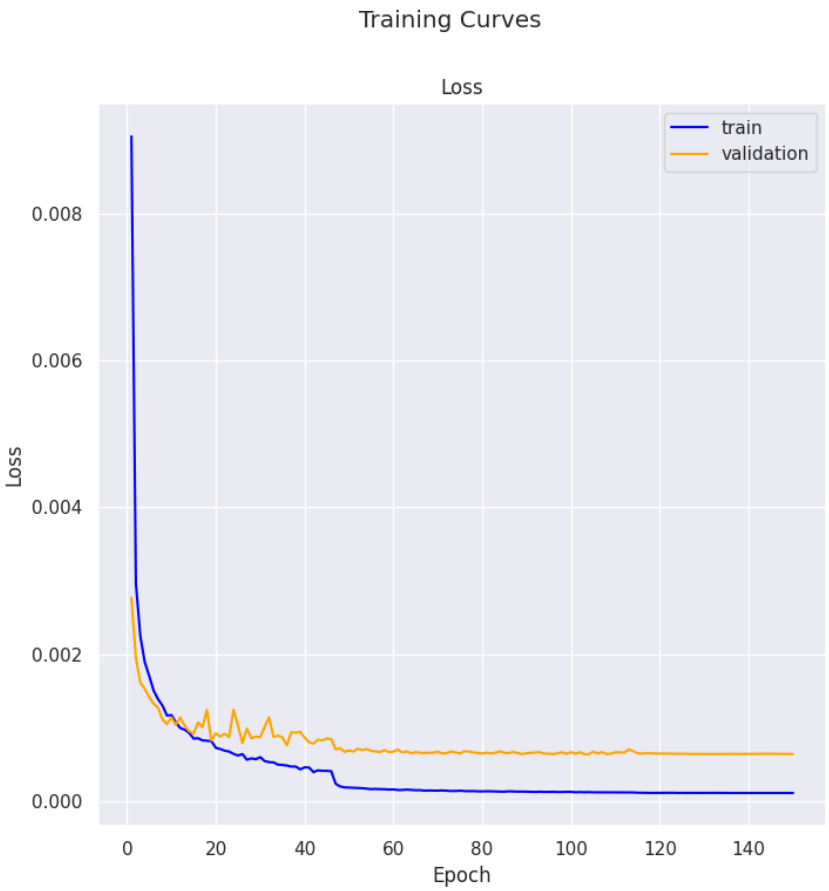
Πίνακας 5.10: Πίνακας που περιέχει τα πειράματα που έγιναν σε όλα τα σύνολα δεδομένων στους δύο αλγορίθμους με την καλύτερη επίδοση. Επίσης, περιέχονται για λόγους σύγκρισης ορισμένα πειράματα των δύο αλγορίθμων με τη λιγότερο καλή επίδοση (όπως μετρήθηκε στην προηγούμενη ενότητα). Σημειώνουμε ότι με τον όρο Classic αναφερόμαστε στον αλγόριθμο δυναμικής δρομολόγησης με συμφωνία. Επίσης, τα αποτελέσματα με αστερίσκο (*) προέκυψαν μετά από εκπαίδευση σε 150 εποχές.

για κάθε κάψουλα DigitCap ως το σταθμισμένο άθροισμα των ψήφων δεν ωφελεί πρακτικά την επίδοση του δικτύου[5]. Συνεπώς, τον πρωτεύοντα ρόλο τον έχουν τα βάρη δρομολόγησης που κλιμακώνουν τις ψήφους. Αυτά είναι ένα επαρκές κριτήριο για την επιλογή της ψήφου που θα δρομολογηθεί, κλιμακωμένη στο επόμενο επίπεδο. Το γεγονός αυτό μπορούμε να το εκμεταλλευτούμε μειώνοντάς τις επαναλήψεις και γλιτώνοντας υπολογιστικό κόστος. Αντί για τη μια περιττή επανάληψη θα μπορούσαμε να εισάγουμε ένα παραπάνω συνελικτικό επίπεδο που, όπως έχει αποδειχθεί στο κεφάλαιο 3, βελτιώνει την ακρίβεια.

Ένα ακόμα ερώτημα που προκύπτει φυσικά είναι αν το δίκτυο του αλγορίθμου Argmax Scaled Routing μαθαίνει τα βάρη του ώστε να ανταποκρίνεται με βέλτιστο τρόπο στον νέο μας αλγόριθμο ή αν εξαρχής ο κλασικός αλγόριθμος με συμφωνία έχει την ιδιότητα που περιγράφουμε (δηλαδή ότι το μέγιστο βάρος δρομολόγησης και η αντίστοιχη ψήφος - ψήφος «εκπρόσωπος» - αρκούν για τη δρομολόγηση). Το ερώτημα αυτό το απαντάμε στο 5.3 ύστερα από τη διενέργεια κατάλληλων ειδικών πειραμάτων.

Ένα τελευταίο σημείο της σύγκρισης των δύο αλγορίθμων είναι ο χρόνος επίδοσης. Αναλυτικότερα, ο αλγόριθμος δυναμικής δρομολόγησης με συμφωνία (στον πίνακα αναγράφεται ως Classic), για μέγεθος δέσμης ίσο με 64 είναι κατά 3 δευτερόλεπτα πιο αργός από τον αλγόριθμο Argmax Scaled Routing (χρόνοι 33 και 30 δευτερόλεπτα αντίστοιχα). Σημειώνουμε ότι όλοι οι αλγόριθμοι έχουν ίσο αριθμό παραμέτρων, δηλαδή είναι $8,227k$ (οκτώ εκατομμύρια 227 χιλιάδες παράμετροι). Εξ΄ αυτών, οι $6,816k$ αφορούν τον κωδικοποιητή ενώ οι $1,411k$ τον αποκωδικοποι-

[5]Παρόλα αυτά, το σταθμισμένο άθροισμα μπορεί να ωφελεί τη σύλληψη των παραμέτρων στιγμιοτύπου της. Αυτό το διερευνούμε στην ενότητα 5.3



Σχήμα 5.5: Στο σχήμα παρατηρούμε τις γραφικές παραστάσεις του σφάλματος εκπαίδευσης και ελέγχου (validation) κατά τη διάρκεια των 150 εποχών για το μοντέλο που ακολουθεί τον αλγόριθμο Dynamic Routing Between Capsules. Είναι βέβαιο ότι παρατηρείται (overfitting) αλλά όχι σε βαθμό που να επηρεάζει την απόδοση στο σύνολο ελέγχου.

ητή.

Σε σύγκριση με τον απόλυτο αλγόριθμο Max Rooting της πρώτης μεθόδου, φαίνεται πως το φιλτράρισμα με πολυδιάστατη συμφωνία πραγματικά βοηθάει τις επιδόσεις του μοντέλου χωρίς να εισάγει επιπλέον παραμέτρους. Προφανώς λοιπόν, η υπόθεση ότι ο κλασικός αλγόριθμος δρομολόγησης μπορεί να μην προσφέρει κάτι παραπάνω από την απλή δρομολόγηση της μέγιστης σε μήκος ψήφου (για κάθε κάψουλα γονέα) είναι ορθή. Τέλος, σε σύγκριση με τον αλγόριθμο Argmax Routing, παρατηρούμε ότι υπάρχουν οφέλη από τη χρήση των βαρών δρομολόγησης όχι μόνο ως κριτήριο για την επιλογή των ψήφων εκπροσώπων αλλά ως μέγεθος για την επιλογή της κλάσης εξόδου (αφού διαμορφώνει άμεσα το μήκος των DigitCaps).

5.3 Ειδικά Πειράματα Μεθόδου 1

Στην ενότητα αυτή πραγματοποιούνται ειδικά πειράματα τα οποία στόχο έχουν να διαφωτίσουν την εσωτερική λειτουργία των νευρωνικών δικτύων με κάψουλες. Επίσης, στόχο έχουν να πιστοποιήσουν ότι οι προτεινόμενοι αλγόριθμοι πληρούν τις χαρακτηριστικές ιδιότητες της τεχνολογίας αλλά και να αποδείξουν ή να καταρρίψουν ορισμένες υποθέσεις αυτών. Τα πειράματα που παρου-

σιάζουμε αφορούν κυρίως το σύνολο δεδομένων MNIST αλλά παρόμοια συμπεράσματα (αν και λιγότερο προφανή) εξάγονται και για το σύνολο δεδομένων Fashion-MNIST που δοκιμάσαμε.

5.3.1 Τι Μαθαίνει να Αναπαριστά η Κάθε Διάσταση του Διανύσματος DigitCap

Ένα συνηθισμένο πείραμα της τεχνολογίας νευρωνικών δικτύων με κάψουλες είναι αυτό στο οποίο χρησιμοποιείται ο ανακατασκευαστής για να διαπιστωθεί τι αναπαριστά η κάθε διάσταση του διανύσματος DigitCap. Υπενθυμίζουμε ότι ένα από τα χαρακτηριστικά των νευρωνικών δικτύων είναι η δυνατότητα αποδόμησης του απεικονιζόμενου αντικειμένου και η ενθυλάκωση των παραμέτρων στιγμιοτύπου του στην αντίστοιχη ενεργή κάψουλα του τελευταίου επιπέδου. Παραδείγματα παραμέτρων στιγμιοτύπου είναι η πόζα, η φωτεινότητα αλλά και άλλα χαρακτηριστικά που αφορούν το συγκεκριμένο στιγμιότυπο. Στο πείραμα αυτό, χρησιμοποιώντας τον ανακατασκευαστή μπορούμε να έχουμε μια εικόνα του τι αναπαριστά το κάθε στοιχείο μιας κάψουλας τελευταίου επιπέδου.

Το συγκεκριμένο πείραμα ονομάζεται τυπικά πείραμα διαταραχής (perturbation testing). Πρακτικά, τροφοδοτούμε σε ένα εκπαιδευμένο - με τη χρήση αποκωδικοποιητή - μοντέλο νευρωνικού δικτύου από κάψουλες μια εικόνα και λαμβάνουμε το διάνυσμα της κάψουλας DigitCap με τη μεγαλύτερη τιμή ενεργοποίησης (με το μεγαλύτερο μήκος). Στη συνέχεια, μεταβάλουμε τις τιμές του διανύσματος (την κάθε μια ξεχωριστά) και τροφοδοτούμε το πειραγμένο διάνυσμα στον αποκωδικοποιητή. Παρατηρώντας την επίδραση των μεταβολών της κάψουλας στην ανακατασκευασμένη εικόνα, είμαστε σε θέση να κατανοήσουμε το χαρακτηριστικό που η συγκεκριμένη θέση του διανύσματος DigitCap κωδικοποιεί. Προτού παρουσιάσουμε τα αποτελέσματα του πειράματος, σημειώνουμε ότι οι μεταβολές (perturbations) ανήκουν στο διάστημα $[-0.25, 0.25]$ και έχουν βήμα 0.05. Με αυτόν τον τρόπο, προκύπτει ότι για ένα διάνυσμα (κάψουλα) που έχει διάσταση 16, έχουμε 16 πειράματα για 11 μεταβολές.

Στο σχήμα 5.6 παρατίθενται τα αποτελέσματα του πειράματος για το σύνολο MNIST και για τον αλγόριθμο δυναμικής δρομολόγησης με συμφωνία. Αν και οι λίγες εποχές σε συνδυασμό με τον μικρό παράγοντα βαρύτητας του σφάλματος ανακατασκευής δεν έχουν επιτρέψει τον πλήρη σχηματισμό του αποκωδικοποιητή, είναι εμφανές ότι οι κάψουλες του τελευταίου επιπέδου (και συγκεκριμένα η κάψουλα 4 στο σχήμα) καταφέρνουν να συλλάβουν τις συγκεκριμένες παραμέτρους στιγμιοτύπου των απεικονιζόμενων αντικειμένων. Για παράδειγμα, για το ψηφίο 4, η 13^{η} διάσταση κωδικοποιεί τον προσανατολισμό του ψηφίου. Η 14^{η} την γωνία μεταξύ δύο ευθύγραμμων τμημάτων που συνθέτουν το ψηφίο 4 κ.ο.κ.

Αξίζει να σημειώσουμε ότι για κάθε κάψουλα, η κάθε διάσταση του διανύσματός της κωδικοποιεί διαφορετικές ιδιότητες του αντικειμένου που αναπαριστά. Παρόλα αυτά, ορισμένες διαστάσεις (π.χ. η δέκατη τρίτη) φαίνεται να έχουν καθολική ερμηνεία.

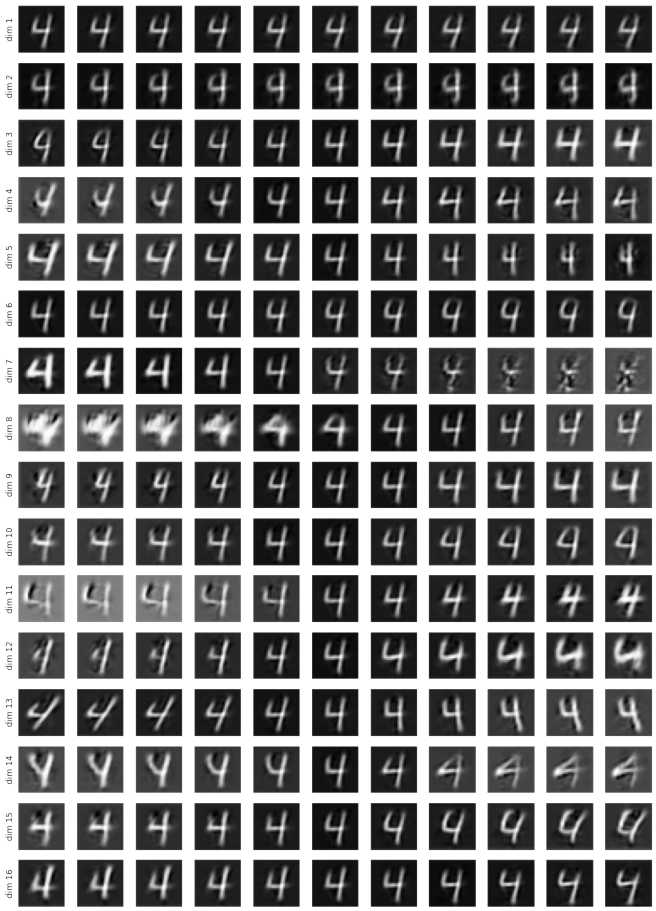
Με μεγάλη περιέργεια πραγματοποιήσαμε τη συγκεκριμένη δοκιμασία για τους υπόλοιπους αλγορίθμους της μεθόδου 1. Αρχικά, για τον αλγόριθμο Argmax Scaled Routing, φαίνεται πως μετά από την εκπαίδευσή του στον ίδιο, περιορισμένο αριθμό εποχών με τον κλασικό αλγόριθμο, αδυνατεί σε μεγάλο βαθμό να συλλάβει τις ιδιότητες που αφορούν την αναπαράσταση των στιγμιοτύπων των αντικειμένων εισόδου (equi–variant parameters). Για αυτό, στο σχήμα 5.8

δε διακρίνεται έντονα κάποια γραμμή του οποίου τα στοιχεία να μεταβάλλονται κατά προβλέψιμο τρόπο.

Από αυτό το πείραμα μπορούμε να συμπεράνουμε ότι στην πραγματικότητα, το σταθμισμένο άθροισμα των ψήφων μπορεί να μην επηρεάζει σημαντικά την επίδοση αλλά διαδραματίζει καθοριστικό ρόλο στην ορθή λειτουργία των καψουλών. Φαίνεται ότι όλες μαζί οι συμφωνούντες ψήφοι συνθέτουν τις ιδιότητες του στιγμιοτύπου.

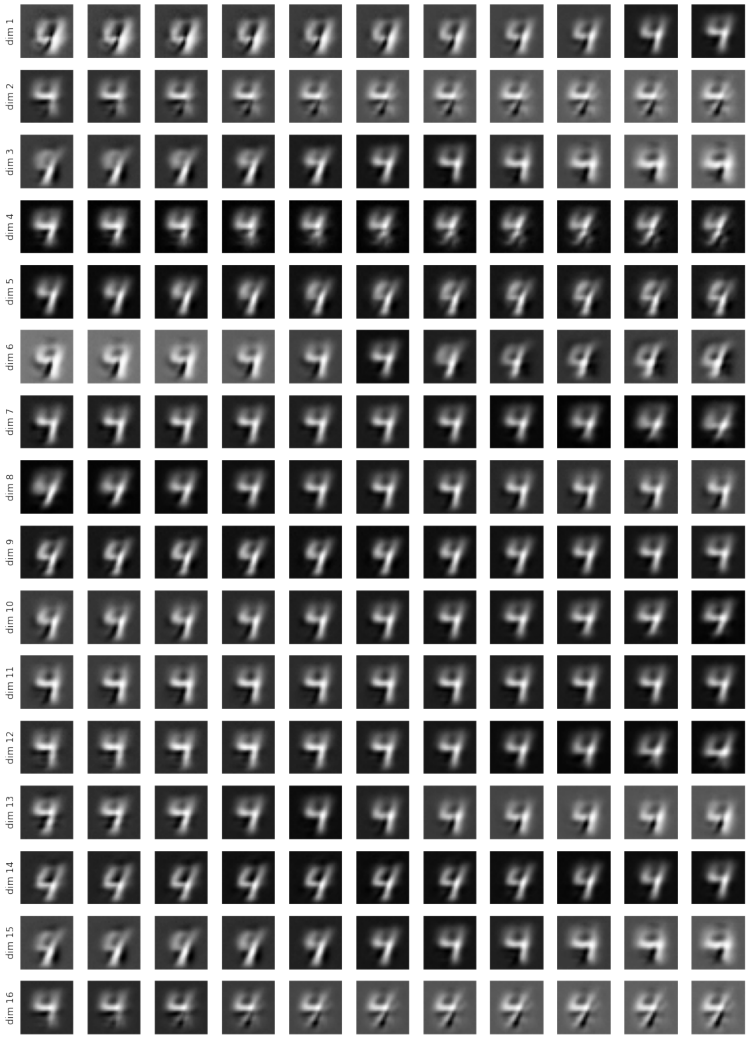
Η επίδοση στην παρούσα δοκιμασία γίνεται ακόμα πιο κακή καθώς απομακρυνόμαστε από τον κλασικό αλγόριθμο δρομολόγησης με συμφωνία. Στο σχήμα 5.9 παρατίθεται το αποτέλεσμα της δοκιμασίας για τον αλγόριθμο Max Routing.

Τέλος οδηγηθήκαμε σε μια μοναδική για την βιβλιογραφία δοκιμή εφαρμόζοντας το πείραμα στον κλασικό αλγόριθμο αλλά εκπαιδευμένο με το σύνολο δεδομένων Fashion-MNIST. Τα αποτελέσματα για τις περισσότερες διαστάσεις του διανύσματος της κάψουλας που αναπαριστά την κλάση υπόδημα δεν μπορούν να περιγραφούν με σαφή τρόπο. Παρόλα αυτά για ορισμένες διαστάσεις, οι μεταβολές είναι εμφανείς. Για παράδειγμα, στο σχήμα 5.10 η διάσταση 6 κωδικοποιεί το ύψος του τακουνιού.

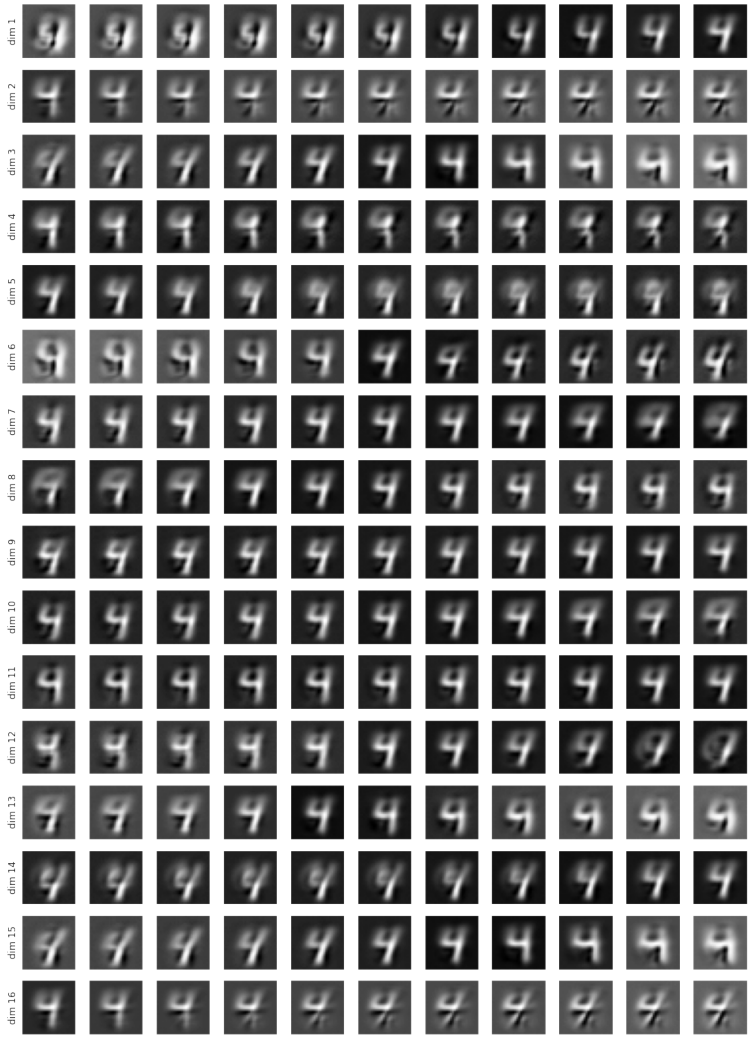


Σχήμα 5.6: Perturbation tests στο σύνολο δεδομένων MNIST στην κάψουλα DigitCap που αναπαριστά την κλάση 4 σε δίκτυο που εκπαιδεύτηκε με την χρήση του δυναμικού αλγορίθμου δρομολόγησης.

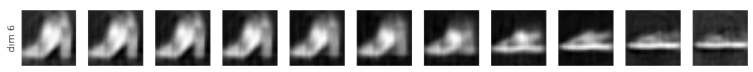
Σχήμα 5.7: Αντιπαραβολή της εικόνας εισόδου (πρώτη γραμμή) με την ανακατασκευασμένη εικόνα (δεύτερη γραμμή). Βλέπουμε ότι το δίκτυο μπορεί να λειτουργήσει ικανοποιητικά και ως αυτοκωδικοποιητής (autoencoder). Η θολούρα των εικόνων ανακατασκευής βελτιώνεται με την αύξηση των εποχών.



Σχήμα 5.8: Perturbation tests στο σύνολο δεδομένων MNIST στην κάψουλα DigitCap που αναπαριστά την κλάση 4 σε δίκτυο που εκπαιδεύτηκε με τη χρήση του αλγορίθμου Argmax Scaled Routing. Ενώ χρησιμοποιείται ο ίδιος αποκωδικοποιητής, το δίκτυο του κωδικοποιητή με τον συγκεκριμένο αλγόριθμο δρομολόγησης αδυνατεί να συλλάβει (capture) τις παραμέτρους στιγμιοτύπου των απεικονίσεων εισόδου.



Σχήμα 5.9: Perturbation tests στο σύνολο δεδομένων MNIST στην κάψουλα DigitCap που αναπαριστά την κλάση 4 σε δίκτυο που εκπαιδεύτηκε με την χρήση του αλγορίθμου Max Routing. Η ανακατασκευή δεν είναι ακριβής λόγω της φύσης του αλγορίθμου δρομολόγησης που δεν παράγει «ισομεταβλητές» (equi-variant) αναπαραστάσεις.



Σχήμα 5.10: Perturbation tests στο σύνολο δεδομένων MNIST στην έκτη διάσταση της κάψουλας DigitCap που αναπαριστά την κλάση «υπόδημα» σε δίκτυο που εκπαιδεύτηκε με την χρήση του αλγορίθμου Classic Routing. Στην εικόνα είναι εμφανές ότι η διάσταση αυτή κωδικοποιεί το ύψος του τακουνιού.

5.3.2 Κριτήριο Επιλογής Ψήφων Αλγορίθμου Δυναμικής Δρομολόγησης με Συμφωνία

Οι αλγόριθμοι που παρουσιάσαμε στην μέθοδο 1 έχουν όλοι τους τον ίδιο αριθμό βαρών. Το μόνο που μεταβάλλεται είναι ο αλγόριθμος δρομολόγησης των ψήφων από το επίπεδο PrimaryCaps επίπεδο DigitCaps. Το γεγονός αυτό μας επιτρέπει την εφαρμογή των αλγορίθμων δρομολόγησης 2, 3 και 4 στο μοντέλο που έχει εκπαιδευτεί με την χρήση του αλγορίθμου 1. Κάθε μια από τις τρείς παραλλαγές του αλγορίθμου δρομολόγησης εξετάζει και ένα κριτήριο επιλογής εκπροσώπων. Θέλουμε να διαπιστώσουμε αν το κριτήριο αυτό είναι επαρκές για την δρομολόγηση των ψήφων ή όχι.

Πρακτικά, στο πείραμα αυτό χρησιμοποιούμε τα βάρη του μοντέλου που εκπαιδεύτηκε στο δυναμικό αλγόριθμο δρομολόγησης με συμφωνία και είχε την καλύτερη επίδοση (0.31% Test Error). Τα βάρη αυτά τα φορτώνουμε στα μοντέλα των τριών υπολοίπων αλγορίθμων και έπειτα εξετάζουμε τις επιδόσεις τους. Τα αποτελέσματα των πειραμάτων παρουσιάζονται στον πίνακα 5.11.

Trained on Algorithm	Tested on Algorithm	r	Total Loss	Test Error (%)
Dynamic Routing (Classic)	Classic	3	0.0007	0.31
	Argmax Scaled	2	0.0064	**0.30**
	Argmax	3	0.0113	0.51
	Max	-1	2.6008	30.78

Πίνακας 5.11: Πειράματα στο MNIST με την εκπαίδευση του αλγορίθμου στον κλασικό, δυναμικό αλγόριθμο δρομολόγησης με συμφωνία και τον έλεγχο του μοντέλου αυτού με τη χρήση των τεσσάρων αλγορίθμων δρομολόγησης.

Τα αποτελέσματα είναι τα αναμενόμενα για τους αλγορίθμους Argmax και Max Routing. Επίσης αναμενόμενες είναι όλες οι απώλειες (Losses) που προκύπτουν από το άθροισμα του σφάλματος περιθωρίου (margin loss) και του σφάλματος ανακατασκευής (mean square error). Άλλωστε, όπως διαπιστώσαμε στα πειράματα που προηγήθηκαν, ο κλασικός αλγόριθμος κάνει την καλύτερη ανακατασκευή αφού μόνο αυτός καταφέρνει να αποδομεί την εικόνα στις παραμέτρους στιγμιοτύπου (το επιτυγχάνει μέσα από την πράξη του σταθμισμένου αθροίσματος που δεν παρατηρείται στους άλλους αλγορίθμους).

Το αξιοσημείωτο από τον πίνακα των αποτελεσμάτων είναι η επίδοση του αλγορίθμου Argmax Scaled Routing με βάρη που προέκυψαν από τον αλγόριθμο Dynamic Routing κατά την εκπαίδευσή του. Αν και η διαφορά είναι μικρή, το γεγονός αυτό συνεπάγεται ότι κατά την πρόβλεψη (inference) είναι προτιμότερο ένα κριτήριο επιλογής κάψουλας που θα δρομολογεί αυτή που αντιστοιχεί στο μέγιστο βάρος δρομολόγησης. Αυτό το γρήγορο κριτήριο οδηγεί σε καλύτερη επίδοση. Είναι λοιπόν προφανής ο ρόλος των βαρών δρομολόγησης στην επιλογή των κλάσεων. Αυτά εντοπίζουν την συμφωνία μεταξύ των ψήφων (που προκύπτουν από τα εκπαιδευμένα βάρη) και πριμοδοτούν την κλάση όπου υπάρχει μεγάλη συμφωνία με το να λαμβάνουν πολύ μεγάλη τιμή. Σε τελική ανάλυση, ο σημαντικός ρόλος των βαρών δρομολόγησης στην επιλογή της κλάσης προδίδεται από την χαμηλή επίδοση του αλγορίθμου Argmax Routing στον οποίο οι εκπρόσωποι δεν κλιμακώνονται από το αντίστοιχο βάρος δρομολόγησης.

5.3.3 Συμφωνία Ψήφων

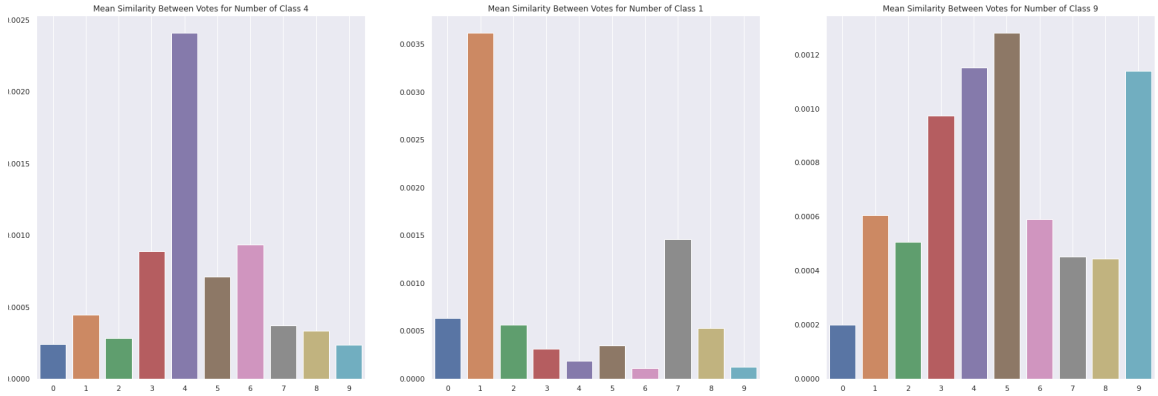
Στην ενότητα αυτή θέλουμε να εξετάσουμε την υπόθεση των νευρωνικών δικτύων με κάψουλες περί φιλτραρίσματος πολυδιάστατης σύμπτωσης. Αναλυτικότερα, χρησιμοποιούμε το καλύτερο μοντέλο μας που εκπαιδεύτηκε στον αλγόριθμο δυναμικής δρομολόγησης με συμφωνία και εξετάζουμε αν η κλάση πρόβλεψης είναι αυτή που εμφανίζει την μεγαλύτερη συμφωνία μεταξύ των ψήφων. Αν κάτι τέτοιο ισχύει, αποτελεί ακόμα μια επιβεβαίωση των θεμελιωδών υποθέσεων των νευρωνικών δικτύων με κάψουλες.

Αναλυτικότερα, για το πείραμά μας τροφοδοτούμε το μοντέλο με ένα παράδειγμα εισόδου της επιλογής μας και έπειτα συγκρίνουμε (με εσωτερικό γινόμενο) τις ψήφους που αντιστοιχούν στην κάθε κάψουλα γονέα μεταξύ τους. Έτσι, παράγεται ένας πίνακας προσοχής (attention matrix) για κάθε DigitCap. Έπειτα, λαμβάνουμε την μέση τιμή των στοιχείων του κάθε πίνακα προσοχής ξεχωριστά και έχουμε 10 ομοιότητες ψήφων: μια για κάθε κάψουλα γονέα. Σημειώνουμε ότι όλες τις ψήφους τις κλιμακώνουμε πριν υπολογίσουμε τα εσωτερικά γινόμενα ώστε να έχουν όλες μοναδιαίο μήκος.

Με μαθηματικούς όρους και χρησιμοποιώντας τον ενισχυμένο συμβολισμό (notation) του προηγούμενου κεφαλαίου έχουμε:

$$\forall j \in \Omega_{L+1} : Similarity_j^L \leftarrow \hat{V}_{j:}^L \times \hat{V}_{j:}^{LT} \tag{5.2}$$

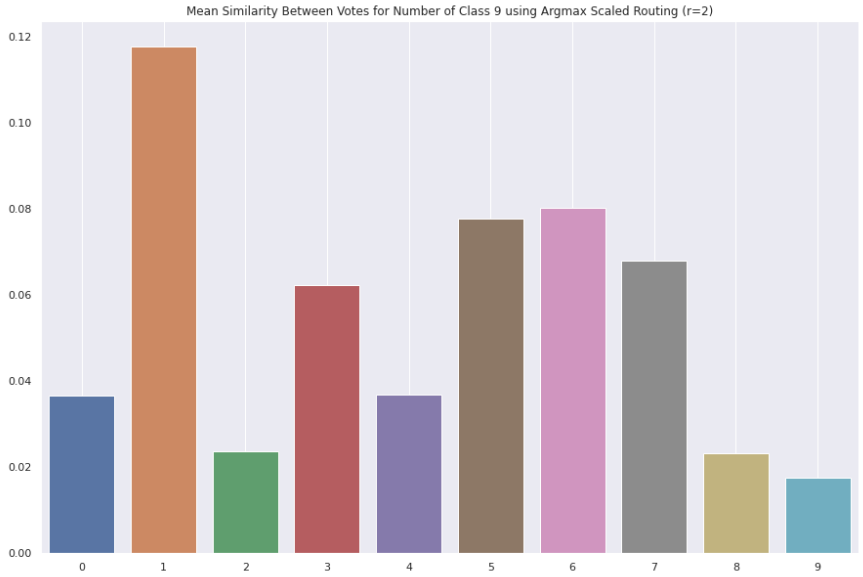
όπου το $\hat{V}^L$ συμβολίζει τις ψήφους αφότου (η κάθε μια) διαιρεθεί με την νόρμα της ώστε να έχει μοναδιαίο μήκος. Ισχύει $\hat{V}^L \in \Re^{n^{L+1} \times n^L \times d^{L+1}}$.



Σχήμα 5.11: Γραφικές παραστάσεις που απεικονίζουν την μέση συμφωνία ψήφων που έχουν προκύψει από τον αλγόριθμο δρομολόγησης με συμφωνία όταν αυτός τροφοδοτείται με εικόνες που περιέχουν τα νούμερα 4, 1 και 9 (από αριστερά προς τα δεξιά).

Από τα πειράματα αυτά, λαμβάνουμε τα αποτελέσματα που φαίνονται στην εικόνα 5.11 παρατηρούμε ότι η συμφωνία παίζει καθοριστικό ρόλο στην δρομολόγηση των καψουλών αφού μέσω αυτής διαμορφώνονται τα βάρη δρομολόγησης. Στην εικόνα 5.11, στα δεξιά, συμπεριλαμβάνουμε και μια σπάνια περίπτωση όπου η μέση συμφωνία είναι μεγαλύτερη σε μη ορθή κλάση. Αυτό, όταν η διαφορά είναι μικρή, δεν οδηγεί απαραίτητα σε λανθασμένη πρόβλεψη (διότι έχουμε λάβει

την μέση συμφωνία). Άλλωστε, για το σύνολο δεδομένω Fashion-MNIST στο οποίο δοκιμάσαμε και εκεί τα πειράματα, φάνηκε ότι στις περισσότερες περιπτώσεις, η μέση συμφωνία μεταξύ των ψήφων της κάψουλας που αντιστοιχεί στη σωστή πρόβλεψη δεν είναι η μεγαλύτερη[6].



Σχήμα 5.12: Γραφική παράσταση που απεικονίζει την μέση συμφωνία ψήφων που έχουν προκύψει από τον αλγόριθμο δρομολόγησης Argmax Scaled Routing όταν αυτός τροφοδοτείται με εικόνες που περιέχουν το νούμερο 9. Παρατηρούμε ότι η μέση συμφωνία της σωστής κλάσης είναι η πιο χαμηλή.

Δοκιμάσαμε το ίδιο πείραμα και σε μοντέλο εκπαιδευμένο στον αλγόριθμο Argmax Routing (βλέπε σχήμα 5.12). Όπως είναι αναμενόμενο, ο αλγόριθμος αδυνατώντας να αποδομήσει αποδοτικά την εικόνα, παράγει ψήφους που δεν έχουν μεγάλη μέση συμφωνία μεταξύ τους. Μάλιστα, φαίνεται στις περισσότερες περιπτώσεις να ισχύει το ανάποδο: δηλαδή η κλάση με την μικρότερη μέση συμφωνία φένεται να είναι η σωστή[7]. Από άλλα πειράματα που δεν περιλαμβάνονται προέκυψε ότι ο συγκεκριμένος αλγόριθμος προτιμά να παράγει ψήφους για την σωστή κλάση όπου οι περισσότερες να μην εμφανίζουν υψηλή συμφωνία (εξού και η χαμηλή τιμή μέσης συμφωνίας). Αντίθετα, προτιμά, μεταξύ των ψήφων που προορίζονται για την σωστή κλάση να υπάρχουν ελάχιστες με την μέγιστη δυνατή συμφωνία (ώστε να «κερδίσουν» το ένα και μεγαλύτερο βάρος δρομολόγησης). Φυσικά, κάτι τέτοιο είναι απολύτως λογικό αφού μια είναι η ψήφος που θα προωθηθεί στο επόμενο επίπεδο. Από την στιγμή που δεν λαμβάνεται σαν αποτέλεσμα το σταθμισμένο άθροισμα και δεν υπάρχει φόβος τα διανύσματα να ακυρώσουν το ένα το άλλο (cancel each other out) δεν αποτελεί πρόβλημα η μικρή μέση συμφωνία.

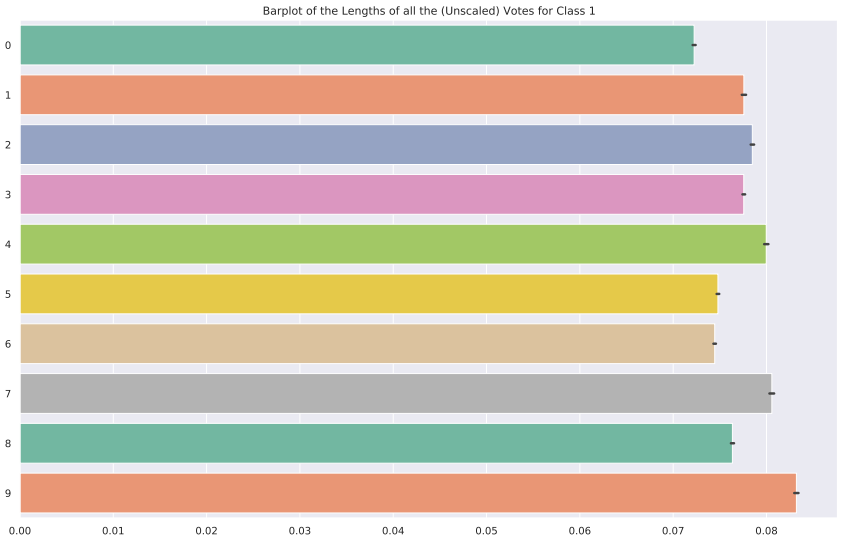
[6] Αυτό είναι το πείραμα στο οποίο εμφανίστηκε η μεγαλύτερη διαφορά μεταξύ των δύο συνόλων δεδομένων. Στα υπόλοιπα πειράματα, τα συμπεράσματα είναι παρόμοια.

[7] Παραπέμπουμε τον αναγνώστη στην ιστοσελίδα του κώδικα για περισσότερα αποτελέσματα.

5.3.4 Κατανομή των Ψήφων

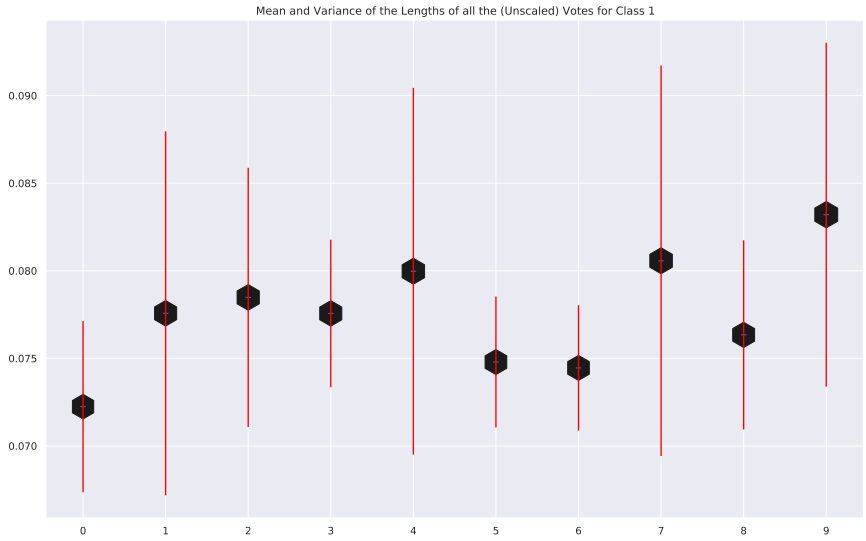
Συνεχίζοντας τα ειδικά πειράματα, θα θέλαμε να διερευνύσουμε την υπόθεση σύμφωνα με την οποία, ο δυναμικός αλγόριθμος δρομολόγησης μπορεί να εκφυλιστεί στην επιλογή της μέγιστης σε μήκος ψήφου. Ισοδύναμα, θέλουμε να ελέγξουμε αν το μοντέλο διαμορφώνει τέτοια εκπαιδευόμενα βάρη ώστε να μη διαδραματίζει τόσο σημαντικό ρόλο η συμφωνία μεταξύ των ψήφων όσο το μήκος των ψήφων.

Από την μειωμένη επίδοση του αλγορίθμου Max Routing, ακόμα και όταν για την εκπαίδευση έχει χρησιμοποιηθεί ο αλγόριθμος Classic Routing, είμαστε προδιατεθειμένοι να πιστέψουμε ότι ο ανωτέρω ισχυρισμός δεν ευσταθεί. Πράγματι, τα πειράματα που διενεργούνται σε αυτή τη παράγραφο επιβεβαιώνουν την προδιάθεσή μας.



Σχήμα 5.13: Γραφική παράσταση του μέσου μήκους των ψήφων ανά κλάση (όταν η ετικέτα είναι το ψηφίο 1).

Οπως φαίνεται από τις εικόνες 5.13 και 5.14 το μέτρο των αρχικών ψήφων (προτού κλιμακωθούν από τα βάρη δρομολόγησης) δεν φαίνεται να συσχετίζεται με κανέναν τρόπο με την κλάση πρόβλεψης. Να σημειώσουμε ότι τα εξής αποτελέσματα δεν αφορούν ένα μεμονωμένο παράδειγμα αλλά υπολογίζονται από τις ψήφους που προκύπτουν από (σχεδόν) όλα τα παραδείγματα της επιλεγμένης κλάσης στο σύνολο δεδομενων ελέγχου. Για αυτό άλλωστε και η διασπορά είναι τόσο μεγάλη.

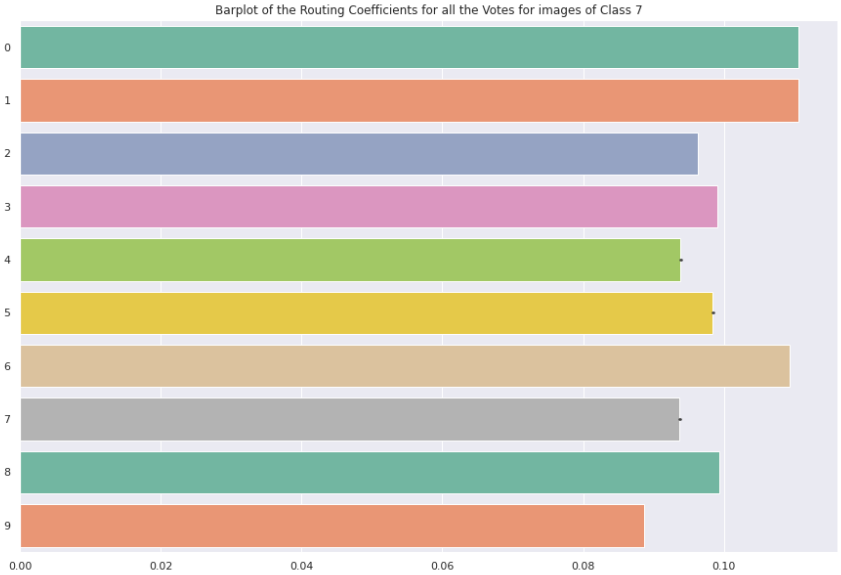


Σχήμα 5.14: Γραφική παράσταση του μέσου μήκους των ψήφων ανά κλάση (όταν η ετικέτα είναι το ψηφίο 1). Στο διάγραμμα εμφανίζεται και η διασπορά με κόκκινες, κάθετες γραμμές.

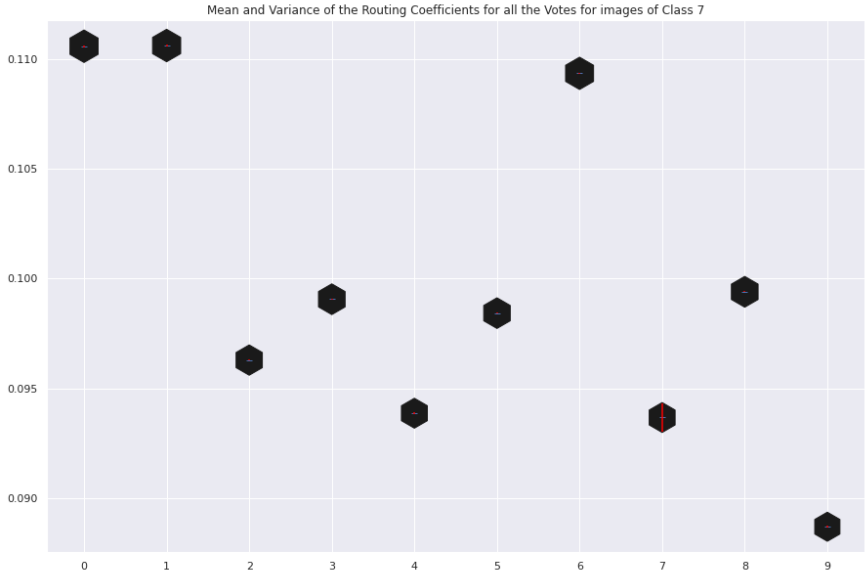
5.3.5 Κατανομή των Βαρών Δρομολόγησης

Φαίνεται ότι τον καθοριστικό ρόλο της επίδοσης τον έχουν τα βάρη δρομολόγησης. Μάλιστα, από τα μέχρι τώρα πειράματα έχει φανεί πως η επιλογή της κλάσης στην οποία αντιστοιχεί η μέγιστη τιμή βάρους δρομολόγησης μπορεί να έχει καλύτερη επίδοση από τον κλασικό αλγόριθμο δρομολόγησης. Θέλωντας να εμβαθύνουμε στην κατανομή των βαρών δρομολόγησης μετά από τρείς επαναλήψεις, δημιουργούμε τρείς γραφικές παραστάσεις όπου έχουμε συγκεντρώσει τα βάρη δρομολόγησης από πάρα πολλά παραδείγματα του συνόλου ελέγχου που απεικονίζουν ένα συγκεκριμένο ψηφίο. Τα βάρη αυτά τα χωρίζουμε ανάλογα με το σε ποιά κάψουλα DigitCap απευθύνονται.

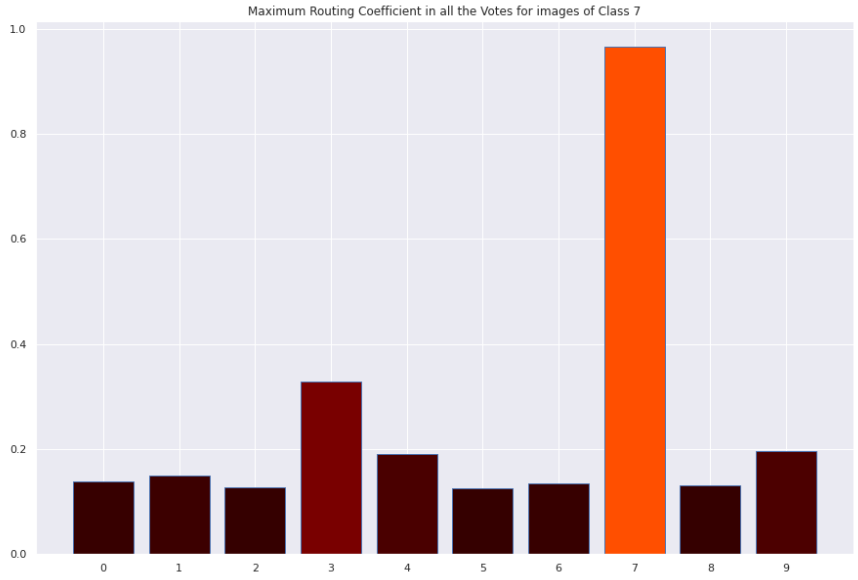
Από τα σχήματα 5.15 και 5.16 φαίνεται ότι η διασπορά είναι μεγαλύτερη για τα βάρη που αντιστοιχούν στις σωστές κάψουλες. Δεν μπορούμε όμως να διακρίνουμε την σωστή κλάση κοιτώντας τις μέσες τιμές. Αν όμως λάβουμε την μέγιστη τιμή των βαρών δρομολόγησης για την κάθε κάψουλα του τελευταίου επιπέδου, διαπιστώνουμε ότι πάντα η σωστή κλάση έχει βάρη δρομολόγησης με την μεγαλύτερη τιμή (βλέπε σχήμα 5.17).



Σχήμα 5.15: Γραφική παράσταση της μέσης τιμής των βαρών δρομολόγησης ανά κλάση.



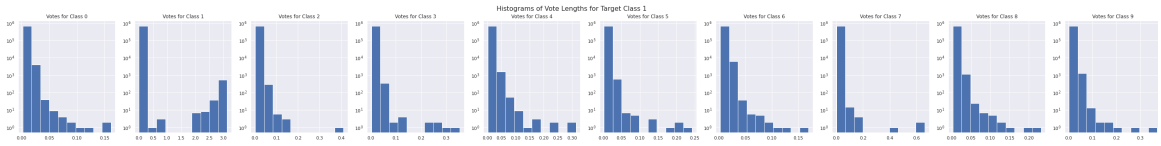
Σχήμα 5.16: Γραφική παράσταση της μέσης τιμής των βαρών δρομολόγησης ανά κλάση. Στο διάγραμμα εμφανίζεται και η διασπορά με κόκκινες, κάθετες γραμμές.



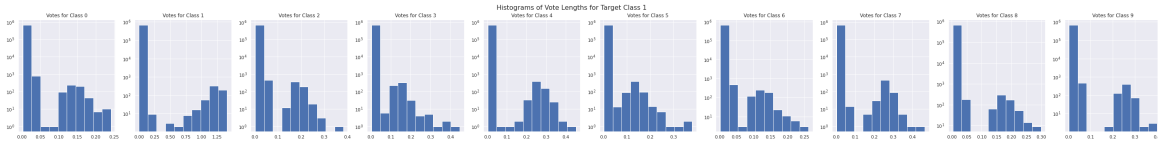
Σχήμα 5.17: Γραφική παράσταση της τιμής του μεγίστου βάρους δρομολόγησης ανά κλάση (για εικόνες εισόδου που απεικονίζουν το νούμερο 7).

5.3.6 Ιστογράμματα των Σταθμισμένων Ψήφων ανα Αριθμό Επαναλήψεων

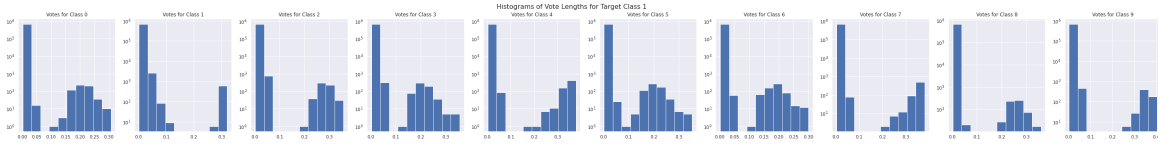
Το τελευταίο πείραμα που πραγματοποιείται εξετάζει την κατανομή των μηκών των σταθμισμένων (από το αντίστοιχο βάρος δρομολόγησης) ψήφων, καθώς αυξάνονται οι επαναλήψεις του αλγορίθμου δυναμικής δρομολόγησης. Χρησιμοποιώντας τον συμβολισμό που εισαγάγαμε στο προηγούμενο κεφάλαιο, υπολογίζουμε τα ιστογράμματα των διανυσμάτων $V_{ij}^L * R_{ij}^L$ ομαδοποιημένων κατά κλάση j. Έτσι προκύπτουν τα σχήματα 5.18 , 5.19 και 5.20 μετά από 3, 2 και 1 επαναναλήψεις αντίστοιχα. Προφανώς, στην πρώτη επανάληψη τα μήκη εξαρτώνται από τους πίνακες μετασχηματισμού αφού τα βάρη δρομολόγησης είναι όλα ίσα. Σε επόμενες επαναλήψεις όμως, τα βάρη δρομολόγησης επηρεάζουν το μήκος των ψήφων με το να αποσιωπούν τις ψήφους σε κλάσεις με μικρή συμφωνία (που δεν αντιστοιχούν στην σωστή κλάση πρόβλεψης) και με το να ενισχύουν ορισμένες ψήφους που δρομολογούνται στην σωστή κλάση. Σημειώνουμε ότι τα ιστογράμματα βρίσκονται σε λογαριθμική κλίμακα και για την παραγωγή τους λήφθηκαν υπόψη τα περισσότερα δείγματα του συνόλου ελέγχου που αφορούν την συγκεκριμένη κλάση στόχο (ο λόγος που δεν λήφθηκαν όλα υπόψη είναι καθαρά υπολογιστικός).



Σχήμα 5.18: Ιστογράμματα του μήκους των ψήφων σταθμισμένων από τα αντίστοιχά τους βάρη δρομολόγησης μετά την τρίτη επανάληψη του δυναμικού αλγορίθμου δρομολόγησης.



Σχήμα 5.19: Ιστογράμματα του μήκους των ψήφων σταθμισμένων από τα αντίστοιχά τους βάρη δρομολόγησης μετά την δεύτερη επανάληψη του δυναμικού αλγορίθμου δρομολόγησης.



Σχήμα 5.20: Ιστογράμματα του μήκους των ψήφων σταθμισμένων από τα αντίστοιχά τους βάρη δρομολόγησης μετά την πρώτη επανάληψη του δυναμικού αλγορίθμου δρομολόγησης.

5.4 Πειραματική Μελέτη Μεθόδου 2

Σε αυτή την ενότητα παρουσιάζουμε ορισμένα από τα πειράματα που πραγματοποιήθηκαν στον αλγόριθμο δρομολόγησης βασισμένο στον EM. Συγκεκριμένα, πειραματιστήκαμε με δύο υλοποιήσεις στην γλώσσα tensorflow. Η πρώτη υλοποίηση είναι αυτή της IBM και περιγράφεται αναλυτικά στο έργο [48]. Η δεύτερη αποτελεί την αυθεντική υλοποίηση του έργου [47] και εντοπίζεται σε αυτή την ιστοσελίδα. Αν και η δεύτερη, πολύπλοκη υλοποίηση από την ομάδα της

Google Brain επέτρεπε την εκπαίδευση και δοκιμή μόνο στο σύνολο δεδομένων smallNORB με τις κατάλληλες επεκτάσεις επιδιώξαμε να εκπαιδεύσουμε το δίκτυο και στο σύνολο δεδομένων MNIST. Προφανώς, όλη η προεπεξεργασία των συνόλων δεδομένων είναι ίδια με αυτή του έργου [47]. Επίσης, κατασκευάσαμε εκ νέου ένα αρχείο Docker για την κατασκευή εικονικού περιβάλλοντος με όλες τις κατάλληλες εκδόσεις λογισμικού. Να σημειώσουμε ότι καμία από τις δύο υλοποιήσεις δεν ήταν σε πλήρη μορφή καθώς δεν προορίζονταν για εκπαίδευση. Χρειάστηκε να γίνουν αρκετές αλλαγές προκειμένου να προσαρμοστούν στις δικές μας απαιτήσεις.

Προτού παρουσιάσουμε τα πειράματα που έγιναν στην μέθοδο δύο, να αναφέρουμε ότι πειραματιστήκαμε για λίγες εποχές και με ορισμένες άλλες επιλογές που παρέχουν οι υλοποιήσεις. Παρόλα αυτά, τα συμπεράσματα ήταν ασαφή για τις λίγες εποχές εκπαίδευσης. Μερικές από τις παραμετροποιήσεις που μπορούν να δοκιμαστούν είναι ο αριθμός των καψουλών του επιπέδου Primary Capsules, το μέγεθος των πυρήνων των πρώτων επιπέδων και ο αριθμός των φίλτρων τους, την προσθήκη επιπλέον συνελικτικών επιπέδων με ή χωρίς κάψουλες, την χρήση του μηχανισμού dropout, την χρήση ενός συνόλου ίδιων μοντέλων εκπαιδευμένων στο ίδιο σύνολο για την δοκιμή στο σύνολο ελέγχου (ensemble) κτλ.

Παρακάτω θα παρουσιάσουμε τα πειράματα που έγιναν σε περιορισμένο αριθμό επαναλήψεων για δύο διαφορετκούς αριθμούς επαναλήψεων (2 και 3) του αλγορίθμου δρομολόγησης. Αν και το μοντέλο δεν έχει πολλές εκπαιδευόμενες παραμέτρους ($310k$) η υπολογιστική του πολυπλοκότητα είναι περίπου πέντε φορές μεγαλύτερη (0.401 operations per batch versus 0.086 operations per batch). Αυτό δεν μας επέτρεψε να διενεργήσουμε εκτενή πειράματα καθώς και επίσης να χρησιμοποιήσουμε μεγάλο μέγεθος δέσμης.

5.4.1 SmallNORB

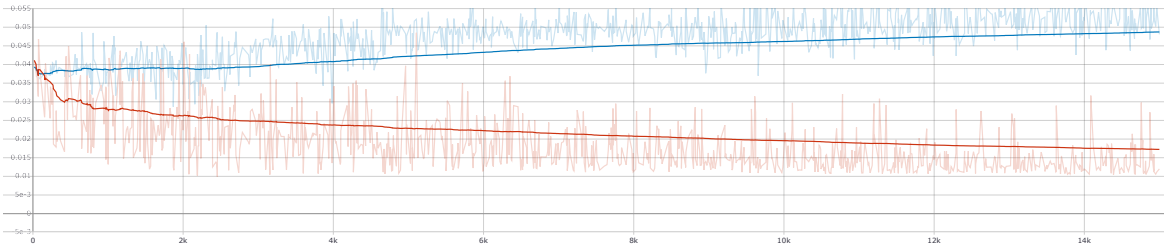
Για να διερευνήσουμε την επίδραση του αριθμού των επαναλήψεων στην ακρίβεια του αλγορίθμου στο σύνολο SmallNORB, εκπαιδεύσαμε το μοντέλο με τον αλγόριθμο δρομολόγησης βασισμένο στον EM (αλγόριθμος 5) για 15000 βήματα με σύνολο δέσμης ίσο με 8 (το μέγιστο δυνατό) το οποίο ισοδυναμεί με 5 επόχές. Τα αποτελέσματα που λάβαμε παρουσιάζονται στον πίνακα 5.12.

Dataset	Routing Iterations	Test Error (%)
SmallNORB	2	**14.04**
SmallNORB	3	63.49

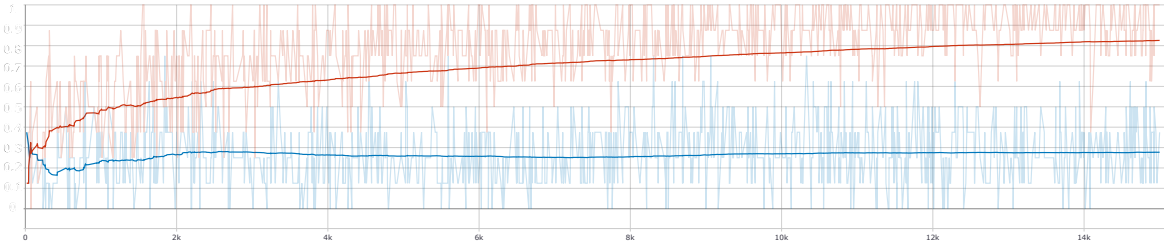
Πίνακας 5.12: Πίνακας στον οποίο φαίνεται η επίδραση του αριθμού των επαναλήψεων στην ακρίβεια όπως μετράται από το σύνολο ελέγχου SmallNORB, όταν χρησιμοποιούνται πολύ λίγες εποχές για την εκπαίδευση του μοντέλου.

Όπως προδίδουν τα αποτελέσματα, ειδικά όταν χρησιμοποιείται μικρό σύνολο δέσμης, ο αλγόριθμος είναι πιθανό να βρεθεί σε αστάθεια. Φυσικά, γνωστά προβλήματα του αλγορίθμου EM όπως το variance collapse δεν διευκολύνουν την διαδικασία εκπαίδευσης με αποτέλεσμα αυτή να γίνεται όλο και πιο ασταθής καθώς αυξάνονται οι επαναλήψεις. Στις εικόνες 5.21 και 5.22 παρουσιάζονται το συνολικό σφάλμα και η ακρίβεια (μέγιστη ακρίβεια είναι 1) για τις δύο περιπτώσεις του πίνακα 5.12 (με κόκκινο απεικονίζεται η εκπαίδευση για 2 επαναλήψεις ενώ με μπλέ

για 3). Παρατηρούμε ότι στην περίπτωση των 3 επαναλήψεων ο αλγόριθμος μπαίνει σε κατάσταση αστάθειας και δεν ανακάμπτει μέχρι το τέλος της εκπαίδευσης.



Σχήμα 5.21: Συνολικό σφάλμα μετρούμενο κατά την εκπαίδευση για τον αλγόριθμο EM της $2^{ης}$ μεθόδου με αριθμό επαναλήψεων 2 (κόκκινο) ή 3 (μπλέ).



Σχήμα 5.22: Ακρίβεια μετρούμενη κατά την εκπαίδευση για τον αλγόριθμο EM της $2^{ης}$ μεθόδου με αριθμό επαναλήψεων 2 (κόκκινο) ή 3 (μπλέ).

5.4.2 MNIST

Συνεχίζουμε με τη διερεύνηση της επιρροής του αριθμού των επαναλήψεων στην ακρίβεια του αλγορίθμου στο σύνολο δεδομένων MNIST. Και πάλι εκπαιδεύσαμε το μοντέλο με τον αλγόριθμο δρομολόγησης βασισμένο στον EM (αλγόριθμος 5) για 15000 βήματα με σύνολο δέσμης ίσο με 8 το οποίο ισοδυναμεί σε αυτή τη περίπτωση με μόλις 4 εποχές. Τα αποτελέσματα που λάβαμε εμφάνιζαν (λόγω και του μικρού συνόλου δέσμης) μεγάλο βαθμό αστάθειας και συνεπώς δεν ήταν αξιοσημείωτα.

5.4.3 Δοκιμή στο SmallNORB με Pretrained Model

Τέλος, δοκιμάσαμε την επίδοση του αλγορίθμου στο σύνολο δεδομένων SmallNORB με την χρήση ενός προ–εκπαιδευμένου μοντέλου (από το διαδίκτυο) με βάρη που σχηματίστικαν κατά την διάρκεια πολύ περισσότερων εποχών. Τα αποτελέσματα, παρουσιάζονται στον πίνακα 5.13 και είναι ίδια με αυτά που παρουσιάζονται στο σχετικό έργο [47] επιβεβαιώνοντας έτσι ότι οι αλλαγές που προκαλέσαμε στον κώδικα δεν επιβάρυναν την επίδοση. Στον σχετικό πίνακα συγκρίνονται οι δύο χρησιμοποιούμενες υλοποιήσεις του αλγορίθμου. Όπως φαίνεται, η υλοποίηση που χρησιμοποιούμε (μαζί με τις αλλαγές μας που δεν επηρεάζουν τον βασικό αλγόριθμο) εμφανίζει την καλύτερη επίδοση (μετρημένη με την χρήση του pre–trained μοντέλου).

Implementation	Framework	r	Test Error (%)
Hinton et al. [47]	tensorflow	3	1.8 (1.4*)
Hinton et al. + Our modifications	tensorflow	3	**1.8 (1.3*)**
Matrix Capsules IBM [48]	tensorflow	2	4.6
Matrix Capsules IBM [48]	tensorflow	3	6.3

Πίνακας 5.13: Πίνακας στον οποίο συγκρίνονται οι επιδόσεις των δύο χρησιμοποιούμενων υλοποιήσεων της μεθόδου 2 (αλγόριθμος 5) στο σύνολο δεδομένων SmallNORB. Για την δεύτερη υλοποίηση, δεν υπήρχε διαθέσιμο κάποιο προ–εκπαιδευμένο μοντέλο για να δοκιμάσουμε τα επικαλούμενα αποτελέσματα. Σημειώνουμε ότι το αστεράκι σημαίνει ότι η πρόβλεψη προκύπτει από την μέση τιμή των προβλέψεων τυχαίων παραθύρων (random crops) της ίδιας εικόνας. Η βελτιωμένη επίδοση μετά τις δικές μας αλλαγές δεν οφείλεται σε αυτές αλλά στον καλύτερο, pre-trained μοντέλο που χρησιμοποιούμε.

5.5 Πειραματική Μελέτη Μεθόδου 3

Στην ενότητα αυτή θα πειραματιστούμε με τους γρήγορους αλγορίθμους δρομολόγησης με μηχανισμό αυτοπροσοχής που παρουσιάσαμε στην ενότητα 4.3. Υπενθυμίζεται ότι στην ενότητα αυτή παρουσιάστηκαν τέσσερεις αλγόριθμοι καθώς και οι παραλλαγές τριών εξ΄ αυτών για την περίπτωση που χρησιμοποιείται μηχανισμός πολυκέφαλης προσοχής (multiheadd attention). Οι τέσσερεις αλγόριθμοι και οι τρείς πολυκέφαλες παραλλαγές τους που παρουσιάστηκαν στην σχετική μέθοδο είναι με την σειρά οι εξής:

- «Απλοικός Αλγόριθμος Δρομολόγησης με Αυτο–Προσοχή» (αλγόριθμος 6)
- «Αλγόριθμος RoMAV» (αλγόριθμος 7)
- «Αλγόριθμος RoWSS» (αλγόριθμος 8)
- «Αλγόριθμος RoWLS» (αλγόριθμος 9)
- «Αλγόριθμος Multihead RoMAV» (αλγόριθμος 12)
- «Αλγόριθμος Multihead RoWSS» (αλγόριθμος 13)
- «Αλγόριθμος Multihead RoWLS» (αλγόριθμος 14)

Ο πρώτος στην λίστα, απλοικός αλγόριθμος μιας και αποτελεί έργο των Mazzia V. et al. [49] στο οποίο παρουσιάζονται εκτενή πειράματά του, δεν συμμετέχει στο πειραματικό μέρος αυτής της ενότητας. Αντίθετα, οι άλλοι (δικοί μας) αλγόριθμοι δοκιμάζονται με πειράματα των οποίων στόχος είναι η πιστοποίηση μιας γρήγορης, κλιμακώσιμης αρχιτεκτονικής νευρωνικών δικτύων με κάψουλες. Για τα πειράματα αυτά, χρησιμοποιούνται όλα τα απαραίτητα σύνολα δεδομένων που σε ένα βαθμό ελέγχουν την τήρηση των ιδιοτήτων των νευρωνικών δικτύων με κάψουλες. Τα σύνολα δεδομενων είναι τα MNIST, CIFAR10, SmallNORB και multiMNIST με τα τελευταία δύο να εξετάζουν την ικανότητα του νευρωνικού δικτύου από κάψουλες να γενικεύει σε νέες οπτικές γωνίες (SmallNORB) και να χειρίζεται αποδοτικά μερική επικάλυψη των εικόνων (multiMNIST) αντίστοιχα.

Εστιάζοντας ακόμα περισσότερο στα σύνολα δεδομένων που χρησιμοποιούνται στα πειράματα της τρίτης μεθόδου, η προεπεξεργασία τους διαφέρει από αυτή που παρουσιάσαμε στα περιάματα

της μεθόδου 1. Αναλυτικότερα, για κάθε σύνολο δεδομένων κάνουμε τα εξής:

MNIST Ακολουθούμε την προεπεξεργασία που προτείνεται στο έργο [132]. Μέγεθος εισόδου: $[28 \times 28 \times 1]$

FashionMNIST Ίδια προεπεξεργασία με το σύνολο δεδομένων MNIST. Μέγεθος εισόδου: $[28 \times 28 \times 1]$

CIFAR10 Διαιρούμε όλα τα στοιχεία με την τιμή 255 προκειμένου να ανήκουν στο σύνολο $[0, 1]$. Έπειτα λαμβάνουμε κεντρικά παράθυρα 28×28 των εικόνων αρχικού μεγέθους 32×32 Μέγεθος εισόδου $[28 \times 28 \times 3]$.

smallNORB Ακολουθούμε Κλιμάκωση σε μέγεθος 64×64 (από το αρχικό μέγεθος που είναι 96×96), κάνουμε περικοπή σε ένα παράθυρο μεγέθους 48×48 και τέλος, προσθέτουμε τυχαία φωτεινότητα και αντίθεση στο σύνολο εκπαίδευσης (παρόμοια με το έργο [47]). Επειδή το σύνολο αποτελείται από στερεοπτικές εικόνες, τις στοιβάζουμε δημιουργώντας μια εικόνα με δύο κανάλια. Μέγεθος εισόδου μετά από προεπεξεργασία: $[48 \times 48 \times 2]$.

MultiMNIST Αρχικά, γεμίζουμε το περιθώριο της κάθε εικόνας του συνόλου δεδομένων MNIST περιμετρικά με μηδενικά προκειμένου το νέο μέγεθος της εικόνας να είναι 36×36. Μια εικόνα MultiMNIST προκύπτει από την εναπόθεση των δύο εικόνων μεταξύ τους με μια πιθανή ολίσθηση μέχρι 4 εικονοστοιχεία στον $x - y$ άξονα. Κατά την εκπαίδευση, παράγουμε 2 MultiMNIST εικόνες για κάθε εικόνα στο αρχικό σύνολο δεδομένων MNIST. Το σύνολο ελέγχου είναι δεκαπλάσιο από το αντίστοιχο σύνολο του MNIST και προκύπτει από την παραγωγή, για κάθε (βασική) εικόνα του αρχικού συνόλου, 10 εικόνων που αποτελούν το συσσωμάτωμα της βασικής εικόνας με 10 τυχαίες (διαφορετικού ψηφίου). Μέγεθος εισόδου $[36 \times 36 \times 2]$.

Αναφορικά με τις λεπτομέρειες υλοποίησης, χρησιμοποιείται ο βελτιστοποιητής nAdam ενώ σε λίγες περιπτώσεις δοκιμάζεται ο βελτιστοποιητής Adam (όπου γίνεται η χρήση του θα αναγράφεται με ρητό τρόπο). Στις περιπτώσεις που χρησιμοποιείται ανακατασκευαστής, εκτός από το σφάλμα Margin Loss έχουμε και το σφάλμα που προκύπτει από την ανακατασκευή (MSE loss). Τα δύο σφάλματα αθροίζονται και διαμορφώνουν το συνολικό σφάλμα εκπαίδευσης, όπως στην περίπτωση της μεθόδου 4.1. Η διαφορά έγκειται στο ότι το σφάλμα ανακατασκευής κλιμακώνεται προτού αθροιστεί με το σφάλμα Margin loss με τον παράγοντα 0.2. Επιπλέον, αξίζει να αναφέρουμε ότι σε ορισμένες περιπτώσεις δοκιμάζουμε εκθετικό πρόγραμμα μείωσης του ρυθμού μάθησης (exponential decay) τον οποίο και θέτουμε στην τιμή 0.001.

Στις επόμενες υπο–ενότητες παρουσιάζονται τα πειράματα που έγιναν με τους διαθέσιμους αλγορίθμους. Δυστυχώς, οι περιορισμένοι πόροι δεν μας επέτρεψαν να κάνουμε μια εξονυχιστική αναζήτηση στον χώρο των υπερπαραμέτρων για την κάθε μέθοδο ξεχωριστά και για το κάθε σύνολο δεδομένων. Άλλωστε, σκοπός μας σε αυτή τη μέθοδο είναι να παρακινήσουμε την έρευνα προς μια υποσχόμενη αρχιτεκτονική που αναπτύξαμε στην τρίτη μέθοδο καταδεικνύοντας αρκετά καλά αποτελέσματα που μπορούν να βελτιωθούν σημαντικά με περαιτέρω αναζήτηση των βέλτιστων υπερπαραμέτρων.

5.5.1 Αναζήτηση στον Χώρο των Υπερπαραμέτρων

Στην υποενότητα αυτή παρουσιάζουμε ορισμένα από τα πειράματα που έγιναν στο πλαίσιο αναζήτησης των βέλτιστων υπερπαραμέτρων για τους αλγορίθμους μας. Οι υπερπαράμετροι για τις οποίες αναζητήσαμε την βέλτιστη τιμή συμπεριλαμβάνουν την χρήση ή μη πολυκέφαλης προσοχής, τη χρήση ή μη ανακατασκευαστή (αλλά και το είδος του ανακατασκευαστή) αλλά και ειδικές - για τον κάθε αλγόριθμο - υπερπαραμέτρους όπως η κλιμάκωση των εκπροσώπων με το αντίστοιχο σκόρ ομοιότητας (Similarity Score) και η ομαλή μεγιστοποίηση ή όχι των similarity scores.

Το σύνολο στο οποίο γίνεται η αναζήτηση των υπερπαραμέτρων είναι το SmallNORB. Επιλέξαμε το συγκεκριμένο σύνολο καθώς τα περισσότερα νευρωνικά δίκτυα με κάψουλες εξετάζουν την επίδοσή τους σε αυτό γεγονός που επιτρέπει τη σύγκριση των επιδόσεών τους. Επίσης, είναι ένα πιό σύνθετο σύνολο δεδομένων από το MNIST γεγονός που διακρύνει άμεσα τις παραμετροποιήσεις που οδηγούν σε μοντέλα με μέτριες επιδόσεις.

Δυστυχώς, οι υπολογιστικοί πόροι που έχουμε στην διαθεσιμότητά μας δεν μας επιτρέπουν να πειραματιστούμε με την εναλλακτική αρχιτεκτονική των πρώτων επιπέδων (αυτή δηλαδή που ακολουθεί και το έργο [47] και όχι αυτή που παρουσιάσαμε στο σχήμα 4.1). Αυτό συμβαίνει διότι η διαδικασία εξαγωγής καψουλών που ακολουθείται στο έργο Dynamic Routing Between Capsules [76] παράγει πολύ μεγάλο αριθμό από Primary Capsules οι οποίες λόγω του αλγορίθμου δρομολόγησής μας, θα πρέπει να συγκριθούν μεταξύ τους (μέσω των παραγόμενων ψήφων τους). Συνεπώς, τα παρακάτω αποτελέσματα αφορούν την αρχιτεκτονική DepthConv. Σε κάθε περίπτωση, μια τέτοια υλοποίηση θα ξέφευγε από τους σκοπούς της μεθόδου για την ανάπτυξη μιας κλιμακώσιμης (scalable) και γρήγορης αρχιτεκτονικής νευρωνικών δικτύων με κάψουλες.

Σαν μια προσπάθεια ελάττωσης της υπερπροσαρμογής του δικτύου στα δεδομένα εκπαίδευσης, εφαρμόσαμε Dropout μετά τα πρώτα δύο συνελικτικά επίπεδα με παράμετρο $p_{value}0.2$, όπως είθισται για την περίπτωση των συνελικτικών δικτύων.

Εκτός από τα πειράματα στις παραμέτρους που εμφανίζονται στους πίνακες που ακολουθούν, έγιναν επιπλέον μελέτες για την μη γραμμική συνάρτηση που εφαρμόζεται σημειακά στα στοιχεία των πινάκων προσοχής. Συγκεκριμένα, περίπου το 40% των πειραμάτων που παρατίθενται σε αυτή την ενότητα επαναλήφθηκαν για την μη γραμμική συνάρτηση Leaky ReLU (αντί για τη συνάρτηση ReLU). Τα αποτελέσματα των πειραμάτων αυτών, δεν ήταν καλύτερα.

Τέλος, σημειώνουμε ότι έγινε εκτενής πειραματική μελέτη (15 περίπου πειράματα) για το είδος και τον αριθμό των επιπέδων κανονικοποίησης που χρησιμοποιούνται στα πρώτα επίπεδα του δικτύου. Ειδικά για το σύνολο SmallNORB, λόγω της ιδιαίτερης φύσης του, είναι ωφέλιμο να γίνεται αξιοποίηση της τεχονολίας κανονικοποίησης που εφαρμόζεται ξεχωριστά σε κάθε παράδειγμα και σε κάθε κανάλι αυτού (Instance Normalization). Τελικά βρέθηκε ότι η χρήση Instance Normalization μετά τα πρώτα τρία συνελικτικά επίπεδα ενισχύει την επίδοση ενώ μετά το τέταρτο συνελικτικό επίπεδο παρατηρείται βελτίωση με τη χρήση επιπέδου κανονικοποίησης κατά δέσμες Batch Normalization.

Πειραματικά Αποτελέσματα Αλγορίθμων RoMAV και Multihead RoMAV

Στην παράγραφο αυτή παρουσιάζουμε τα αποτελέσματα από τα πειράματα που πραγματοποιήθηκαν στους αλγορίθμους RoMAV και Multihead RoMAV (αλγόριθμοι 7 και 12) σε διάφορες παραμετροποιήσεις αυτών. Σημειώνουμε ότι οι δοκιμές αυτές δεν περιλαμβάνουν παραμέτρους όπως ο ρυθμός μάθησης και η τιμή κλιμάκωσης του σφάλματος ανακατασκευής. Επίσης, σημειώνουμε ότι χρησιμοποιούμε σχετικά μικρό μέγεθος δέσμης (με τιμή 8) και εκπαιδεύουμε για 30 εποχές.

Φυσικά, καλύτερα αποτελέσματα μπορούν να προκύψουν με την διερεύνηση επιπλέον υπερπαραμέτρων (π.χ. σχετικών με τον περιορισμό της υπερπροσαρμογής) και την εκπαίδευση σε περισσότερες εποχές. Άλλωστε, σκοπός των πειραμάτων αυτών δεν είναι η εύρεση της καλύτερης παραμετροποίησης αλλά η απόδειξη ότι οι αλγόριθμοι που έχουμε αναπτύξει είναι ικανοί, με λιγότερη πολυπλοκότητα, να ανταγωνιστούν τους αλγορίθμους που ανέπτυξε η ομάδα του Hinton G. αλλά και άλλους αλγορίθμους δρομολόγησης νευρωνικών δικτύων με κάψουλες της βιβλιογραφίας.

Στον πίνακα 5.14 παρατίθενται τα αποτελέσματα των σχετικών πειραμάτων. Για το κάθε πείραμα έχουμε καταγράψει τον αριθμό παραμέτρων του κωδικοποιητή αλλά και τον χρόνο για την εκτέλεση μιας πλήρους εποχής και ενός βήματος. Επιπρόσθετα, έχουμε καταγράψει μεταξύ άλλων, τον τύπο του αποκωδικοποιητή όταν χρησιμοποιούμε έναν. Τα δύο είδη είναι ο πλήρως διασυνδεδεμένος (FC) με $5,289.9k$ και ο αποκωδικοποιητής με επίπεδα αποσυνέλιξης (Decon.) που έχει $229.5k$ παραμέτρους. Σημειώνουμε ότι όταν ο αριθμός των κεφαλών είναι μονάδα ουσιαστικά προστίθενται δύο πλήρως διασυνδεδεμένα επίπεδα (βλέπε ενότητα 2.3.3).

Epoch T. (s)	Step T. (ms)	Param. Count	Heads	Recon. (type)	Test Error (%)
34	11	151.8k	0	yes (FC)	2.2222
45	15	151.8k	0	yes (Decon.)	2.0123
28	09	151.8k	0	no	2.3086
36	12	152.8k	1	yes (FC)	1.9095
47	16	152.8k	1	yes (Decon.)	1.7901
31	10	152.8k	1	no	2.2428
36	12	152.8k	2	yes (FC)	1.7984
47	16	152.8k	2	yes (Decon.)	**1.4403**
31	10	152.8k	2	no	**1.5761**

Πίνακας 5.14: Αποτελέσματα πειραμάτων για την εύρεση των βέλτιστων υπερπαραμέτρων του αλγορίθμου RoMAV αρχιτεκτονικής DepthConv στο σύνολο δεδομένων SmallNORB. Τα πειράματα αυτά πραγματοποιήθηκαν για 30 εποχές με μέγεθος δέσμης ίσο με 8.

Τα αποτελέσματα είναι αρκετά ενθαρρυντικά, ειδικά αν αναλογιστεί κανείς ότι η καλύτερη υλοποίηση της ομάδας του Hinton et al. στο συγκεκριμένο dataset έχει ποσοστό σφάλματος (test error) ίσο με 1.8. Συγκριτικά, ο απλοικός αλγόριθμος δρομολόγησης με προσοχή (αλγόριθμος 6) από τον οποίο εμπνεύστηκε η τρίτη μέθοδος έχει ποσοστό σφάλματος 2.54. Επίσης, πρέπει να λάβουμε υπόψη ότι τα αποτελέσματα αυτά έχουν προκύψει από μόλις 30 εποχές χωρίς πολλά αντίμετρα υπερπροσαρμογής (overfitting).

Από τα αποτελέσματα των πειραμάτων για τον αλγόριθμο RoMAV είναι εμφανές ότι η προσθήκη κεφαλών προσοχής (attention heads) συμβάλλει στην επίδοση του δικτύου χωρίς να αυξάνεται σημαντικά ο αριθμός των παραμέτρων. Μάλιστα, από τις δύο στις τρείς κεφαλές προσοχής ο αριθμός των παραμέτρων δεν αυξάνεται αφού διαιρούμε το μέγεθος αναπαράστασης των διανυσμάτων της κάθε κεφαλής (Q, K, V) δια δύο. Δηλαδή, με τους όρους της ενότητας 4.3, είναι $d_k^L = d_v^L = d^{L+1}//2$.

Επίσης, μπορούμε να παρατηρήσουμε ότι η χρήση αποκωδικοποιητή με επίπεδα αποσυνέλιξης (fractionally strided convolutional layers) αφενώς αυξάνει σε πιο μεγάλο βαθμό την επίδοση από άλλους κωδικοποιητές και αφετέρου δεν επιβαρύνει σημαντικά το υπολογιστικό κόστος αφού προστίθενται $229.5k$ παράμετροι (σε αντίθεση με τον αποκωδικοποιητή FC που προσθέτει $5,289.9k$ παραμέτρους).

Τέλος, στο σχήμα 5.23 απεικονίζονται οι γραφικές παραστάσεις του σφάλματος στα σύνολα εκπαίδευσης και επαλήθευσης για το μοντέλο με την καλύτερη επίδοση (αυτό με δύο κεφαλές και ανακατασκευαστή Decon.). Παρατηρούμε ότι η έλλειψη μεθόδων αποτροπής υπερπροσαρμογής (πέρα από τον ανακατασκευαστή και τα επίπεδα Dropout) αλλά και η χρήση μικρού συνόλου δέσμης οδηγούν σε βαθμιαία υπερπροσαρμογή των βαρών στο σύνολο εκπαίδευσης (το πρώτο) και σε αστάθεια (το δεύτρο). Το φαινομενικά παράδοξο μικρότερο σφάλμα του validatio set οφείλεται στο ότι κατά την εκπαίδευση, στο σύνολο validtaiton εφαρμόζουμε λιγότερους μετασχηματισμούς στις εικόνες εισόδου που το δίκτυο αποκωδικοποιητή πιο εύκολα ανακατασκευάζει.

Σχήμα 5.23: Συνολικό σφάλμα μετρούμενο κατά την εκπαίδευση για τον αλγόριθμο Multihead RoMAV της $3^{ης}$ μεθόδου με αριθμό κεφαλών 2.

Πειραματικά Αποτελέσματα Αλγορίθμου RoWSS και Multihead RoWSS

Συνεχίζοντας την πειραματική μελέτη της μεθόδου 3, επόμενος αλγόριθμος στην σειρά είναι ο RoWSS (αλγόριθμος 8) και η πολυκέφαλη παραλλαγή του (αλγόριθμος 13). Σε αυτούς τους αλγορίθμους, τα πειράματα είναι περισσότερα καθώς θέλουμε να εξετάσουμε, εκτός από τις υπερπαραμέτρους που πειράζαμε στην προηγούμενη παράγραφο, επιπέον υπερπαραμέτρους που δεν ήταν διαθέσιμοι στον προηγούμενο αλγόριθμο (εκ φύσεως). Αναλυτικότερα, αυτές οι παραλλαγές είναι η κλιμάκωση των εκπροσώπων με τις τιμές των αντίστοιχων σκορ ομοιότητας (ScaleEMB) αλλά και η εφαρμογή της συνάρτησης Softmax στα σκορ ομοιότητας (SoftSC) [8].

Τα σχετικά πειράματα παρατίθενται στον πίνακα 5.15. Σημειώνουμε ότι ορισμένα αποτελέσματα έχουν προκύψει από την χρήση εκθετικής μείωσης του ρυθμού μάθησης, καθώς μετά από πειράματα βρέθηκε πως συνέβαλε στην επίδοση[9].

Από τα σχετικά αποτελέσματα είναι ασφαλές να υποθέσουμε ότι η κλιμάκωση των εκπροσώπων δεν βοηθάει την επίδοση του δικτύου. Εν αντιθέση, προκαλεί ακόμα μεγαλύτερη αστάθεια στο δίκτυο κατά την εκπαίδευση. Σε ό,τι αφορά τη χρήση της συνάρτησης Softmax, είναι εμφανές ότι δεν συνεπάγεται πάντα καλύτερη επίδοση. Με άλλα λόγια, η υπόθεση της ύπαρξης μοναδικού πατέρα (single parent assumption), όταν παραβαίνεται, μπορεί να οδηγήσει σε καλύτερα αποτελέσματα.

Αναφορικά με τον αριθμό των κεφαλών και το είδος του αποκωδικοποιητή, οι παρατηρήσεις είναι παρόμοιες με αυτές του αλγορίθμου RoMAV. Δηλαδή, και εδώ η αύξηση των κεφαλών γενικά οδηγεί σε καλύτερη επίδοση χωρίς ουσιαστική επιβάρυνση της πολυπλοκότητας. Βέβαια, αξίζει να σημειώσουμε ότι στην περίπτωση που ο αριθμός κεφαλών είναι ίσος με τη μονάδα, δεν παρατηρείται βελτίωση. Πιθανώς, αυτό οφείλεται στην επιπλέον πολυπλοκότητα που εισάγουν οι προβολές του διανύσματος και την τάση για υπερπροσαρμογή.

[8] Βλέπε αλγόριθμο 8 γραμμή 30.

[9] Θα ήταν αδύνατο να ενσωματώσουμε όλα τα πειράματα στο παρόν έργο. Σε περίπτωση που ορισμένα δε βρίσκονται στην ιστοσελίδα του κώδικα, μπορούν να παρασχεθούν μετά από αίτημα μαζί με τα αρχεία βαρών των μοντέλων.

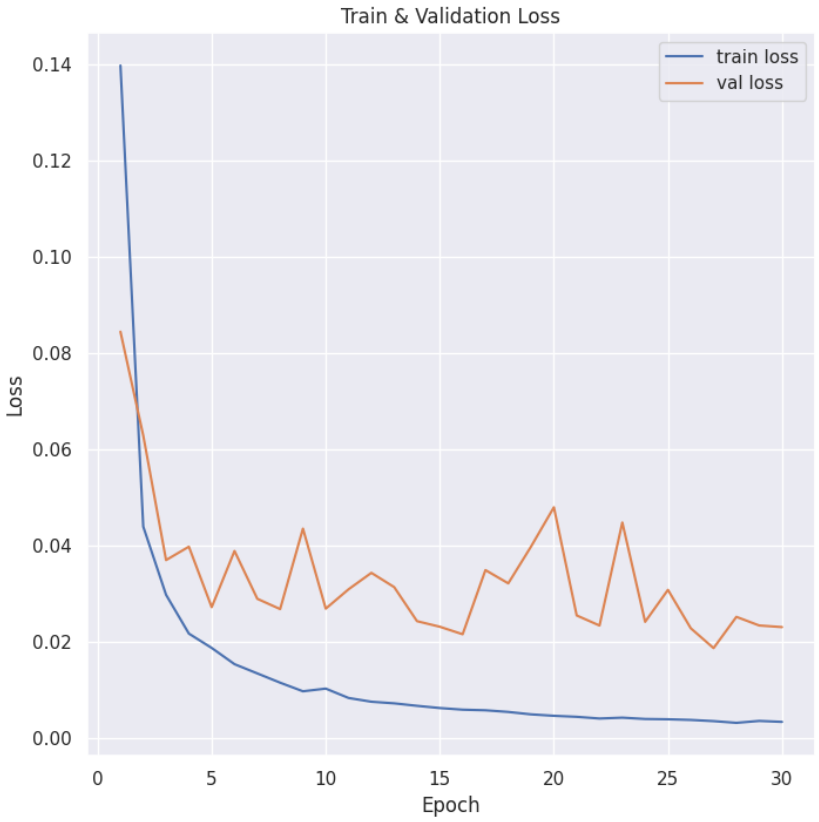
Epoch (s)	Step (ms)	Param.	Heads	SoftSC	ScaleEmb	Recon.	Test Error (%)
35	11	151.8k	0	yes	no	FC	1.9959
34	11	151.8k	0	no	no	FC	1.9095
35	12	151.8k	0	yes	yes	FC	3.0206
35	11	151.8k	0	no	yes	FC	2.1564
45	15	151.8k	0	yes	no	Decon.	1.7984
45	15	151.8k	0	no	no	Decon.	1.7407
46	15	151.8k	0	yes	yes	Decon.	2.5885
45	15	151.8k	0	no	yes	Decon.	2.8642
29	09	151.8k	0	yes	no	no	2.1193
28	09	151.8k	0	no	no	no	**1.5309**
29	10	151.8k	0	yes	yes	no	2.3498
29	10	151.8k	0	no	yes	no	2.2346
37	12	152.8k	1	yes	no	FC	2.0741
37	12	152.8k	1	no	no	FC	2.1481
37	12	152.8k	1	yes	yes	FC	2.8848
37	12	152.8k	1	no	yes	FC	1.9218
48	16	152.8k	1	yes	no	Decon.	2.1399
48	16	152.8k	1	no	no	Decon.	2.8930
48	16	152.8k	1	yes	yes	Decon.	2.2757
48	16	152.8k	1	no	yes	Decon.	2.1152
31	10	152.8k	1	yes	no	no	1.6626
31	10	152.8k	1	no	no	no	1.5679
31	10	152.8k	1	yes	yes	no	2.0370
31	10	152.8k	1	no	yes	no	2.9877
37	12	152.8k	2	yes	no	FC	1.6626
37	12	152.8k	2	no	no	FC	1.5844
38	12	152.8k	2	yes	yes	FC	2.8272
37	13	152.8k	2	no	yes	FC	2.8313
48	16	152.8k	2	yes	no	Decon.	**1.5597**
48	16	152.8k	2	no	no	Decon.	1.8807
49	16	152.8k	2	yes	yes	Decon.	3.4774
49	16	152.8k	2	no	yes	Decon.	2.1276
31	10	152.8k	2	yes	no	no	1.8272
31	10	152.8k	2	no	no	no	1.7695
31	10	152.8k	2	yes	yes	no	2.5679
31	10	152.8k	2	no	yes	no	2.3292

Πίνακας 5.15: Αποτελέσματα εκτενών πειραμάτων για την εύρεση των βέλτιστων υπερπαραμέτρων του αλγορίθμου RoWSS αρχιτεκτονικής DepthConv στο σύνολο δεδομένων SmallNORB. Πραγματοποιήθηκαν για 30 εποχές με μέγεθος δέσμης ίσο με 8.

Παρατηρούμε επιπλέον ότι η χρησιμότητα του δικτύου αποκωδικοποιητή είναι λιγότερο προφανής. Ένας πιθανός λόγος είναι ότι σε πολλές διατάξεις του αλγορίθμου η τιμή κλιμάκωσης του σφάλματος ανακατασκευής δεν είναι η κατάλληλη. Αυτό διότι το σφάλμα ανακατασκευής μετά την εκπαίδευση κυμαίνεται στις 1 με 2 μονάδες, ενώ το σφάλμα εκπαίδευσης είναι τρεις τάξεις μεγέθους χαμηλότερο (φυσικά και λόγο υπερπροσαρμογής). Συνεπώς, ακόμα και μετά την κλιμάκωση με τον όρο 0.2, στις τελευταίες εποχές του αλγορίθμου, το σφάλμα αποκωδικοποιητή

κυριαρχεί του σφάλματος κωδικοποιητή, γεγονός που δεν επιτρέπει την κατάλληλη προσαρμογή των βαρών. Επίσης, ο μικρός ανακατασκευαστής Decon. δεν επαρκεί για να ανακατασκευάσει την εικόνα εισόδου.

Στο σχήμα 5.24 απεικονίζονται οι γραφικές παραστάσεις του σφάλματος στα σύνολα εκπαίδευσης και επαλήθευσης για το μοντέλο με την καλύτερη επίδοση (αυτό με μηδέν κεφαλές και χωρίς ανακατασκευαστή).



Σχήμα 5.24: Συνολικό σφάλμα μετρούμενο κατά την εκπαίδευση για τον αλγόριθμο RoWSS της $3^{ης}$ μεθόδου. Εδώ δεν έχουμε ανακατασκευαστή. Για αυτό και το σφάλμα ελέγχου είναι μεγαλύτερο από το σφάλμα εκπαίδευσης.

Πειραματικά Αποτελέσματα Αλγορίθμων RoWLS και Multihead RoWLS

Τελευταίος αλγόριθμος για τον οποίο θα κάνουμε αναζήτηση υπερπαραμέτρων για το σύνολο δεδομένων SmallNORB είναι ο RoWLS (αλγόριθμος 9). Φυσικά, όπως και στα προειγούμενα πειράματα, όταν ο αριθμός κεφαλών είναι μεγαλύτερος από το μηδέν, υπονοείται ότι χρησιμοποιούμε τον αλγόριθμο Multihead RoWLS (αλγόριθμος 14).

Οι παράμετροι που εξετάζουμε είναι οι ίδιες με αυτές του προηγούμενου πειράματος. Η διαφορά είναι ότι παρατηρώντας ότι η κλιμάκωση των embeddings με τα σκορ δεν συμβάλει στην επίδοση του αλγορίθμου RoWLS, μειώσαμε τα πειράματα και για τον αλγόριθμο Multihead RoWLS εξετάσαμε μόνο την περίπτωση όπου δεν πραγματοποιείται η σχετική κλιμάκωση. Τα αποτελέσματα

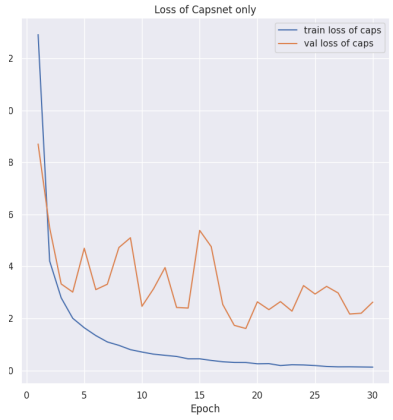
των πειραμάτων παρατίθενται στον πίνακα 5.16.

Epoch (s)	Step (ms)	Param.	Heads	SoftSC	ScaleEmb	Recon.	Test Error (%)
34	11	151.8k	0	yes	no	FC	2.3992
34	11	151.8k	0	no	no	FC	2.2757
35	12	151.8k	0	yes	yes	FC	2.2881
35	11	151.8k	0	no	yes	FC	2.2593
45	15	151.8k	0	yes	no	Decon.	2.5103
45	15	151.8k	0	no	no	Decon.	1.7613
46	15	151.8k	0	yes	yes	Decon.	2.3868
45	15	151.8k	0	no	yes	Decon.	2.2263
29	09	151.8k	0	yes	no	no	1.7942
28	09	151.8k	0	no	no	no	1.6831
29	10	151.8k	0	yes	yes	no	2.2675
29	10	151.8k	0	no	yes	no	2.5597
37	12	152.8k	1	yes	no	FC	2.4280
37	12	152.8k	1	no	no	FC	2.0658
48	16	152.8k	1	yes	no	Decon.	2.8272
48	16	152.8k	1	no	no	Decon.	1.8477
31	10	152.8k	1	yes	no	no	2.1235
31	10	152.8k	1	no	no	no	2.1523
37	12	152.8k	2	yes	no	FC	2.6996
37	12	152.8k	2	no	no	FC	2.1317
48	16	152.8k	2	yes	no	Decon.	2.2058
48	16	152.8k	2	no	no	Decon.	**1.5967**
31	10	152.8k	2	yes	no	no	**1.5021**
31	10	152.8k	2	no	no	no	2.1276

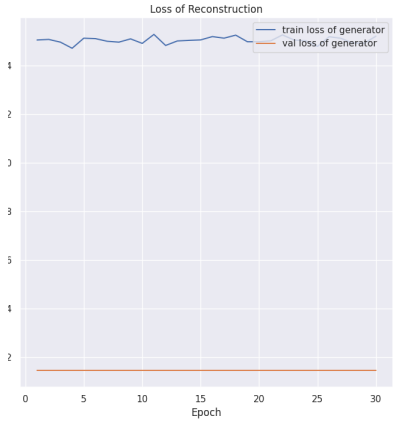
Πίνακας 5.16: Αποτελέσματα πειραμάτων για την εύρεση των βέλτιστων υπερπαραμέτρων του αλγορίθμου RoWLS αρχιτεκτονικής DepthConv στο σύνολο δεδομένων SmallNORB. Όλα τα πειράματα της ενότητας πραγματοποιήθηκαν για 30 εποχές με μέγεθος δέσμης ίσο με 8.

Παρατηρούμε ότι τα αποτελέσματα αν και είναι πολύ ικανοποιητικά, σε γενικές γραμμές φαίνεται ότι η μέθοδος RoWSS υπερτερεί της μεθόδου RoWLS (πλην λίγων εξαιρέσεων). Το γεγονός αυτό συνεπάγεται ότι το κριτήριο επιλογής εκπροσώπων με βάση τη συμφωνία είναι συνήθως καλύτερο από το κριτήριο του μήκους των αναπαραστάσεων (embeddings).

Σχετικά με τον ανακατασκευαστή, είναι αξιοσημείωτο πως και πάλι, το μοντέλο με τις καλύτερες παραμέτρους είναι ένα χωρίς ανακατασκευαστή. Είναι λοιπόν πιθανό ότι το σφάλμα ανακατασκευής είναι μεγαλύτερο από αυτό που θα έπρεπε στα συγκεκριμένα πειράματα (ή ότι ο ανακατασκευαστής δεν είναι κατάλληλος). Στις εικόνες 5.25 και 5.26 αντιπαραβάλλονται τα σφάλματα από τον κωδικοποιητή (αριστερά) και τον αποκωδικοποιητή (δεξιά). Βλέπουμε ότι το σφάλμα της ανακατασκευής είναι δύο κλάσεις μεγέθους μεγαλύτερο του σφάλματος margin loss. Επίσης, βλέπουμε ότι το σφάλμα ελέγχου για την ανακατασκευή είναι μικρότερο από το σφάλμα στο σύνολο εκπαίδευσης. Όπως έχουμε εξηγήσει, αυτό οφείλεται στην πιο έντονη επαύξηση δεδομένων που χρησιμοποιούμε για το σύνολο εκπαίδευσης.



Σχήμα 5.25: Σφάλμα κωδικοποιητή (αλγόριθμος RoWLS).



Σχήμα 5.26: Σφάλμα αποκωδικοποιητή (αλγόριθμος RoWLS).

5.5.2 Πειράματα Παραμέτρου Ανακατασκευής

Προκειμένου να εξεταστεί αν χαμηλότερη τιμή της παραμέτρου κλιμάκωσης του σφάλματος ανακατασκευής οδηγεί σε καλύτερα αποτελέσματα, διενεργούμε πείραμα στο μοντέλο του αλγορίθμου (RoWSS). Χρησιμοποιούμε την παραμετροποίηση που του εξασφαλίζει test error **1.5597** αλλά με πιο έντονη κλιμάκωση του σφάλματος ανακατασκευής (τιμές 0.02 και 0.0005). Τα πειράματα πραγματοποιούνται και πάλι για 30 εποχές με μέγεθος δέσμης ίσο με 8. Παρατίθενται στον πίνακα 5.17.

Scaling Factor	Recon. (type)	Test Error (%)
0.2	yes (Decon.)	**1.5597**
0.02	yes (Decon.)	2.3004
0.005	yes (Decon.)	1.8889

Πίνακας 5.17: Αποτελέσματα πειραμάτων για την εύρεση της βέλτιστης τιμής του παράγοντα κλιμάκωσης του σφάλματος ανακατασκευής στο σύνολο δεδομένων SmallNORB. Τα πειράματα πραγματοποιήθηκαν για 30 εποχές με μέγεθος δέσμης ίσο με 8.

Από τα σχετικά πειράματα, δεν φαίνεται να υπάρχει κάποια βελτίωση με την μείωση της τιμής της παραμέτρου. Για τον λόγο αυτό, στα πειράματα εμβάθυνσης, αφήνουμε την υπερπαράμετρο στην προκαθορισμένη της τιμή η οποία έχει προσφέρει αξιοσημείωτα αποτελέσματα σε όλους τους αλγορίθμους της τρίτης μεθόδου.

5.5.3 Επιλεκτική Εμβάθυνση στα Σύνολα Δεδομένων

Σε αυτήν την ενότητα, επιλέγουμε από τον κάθε τύπο αλγορίθμου (RoMAV, RoWSS, RoWLS) τα μοντέλα με τις δύο παραμετροποιήσεις που οδήγησαν στα καλύτερα αποτελέσματα στο σύνολο δεδομένων SmallNORB και τα εκπαιδεύουμε σε μια πληθώρα από σύνολα δεδομένων[10]. Ουσιαστικά, για τους αλγορίθμους (RoMAV, RoWSS, RoWLS) έχουμε δύο μοντέλα, ένα με ανακατα-

[10]Είναι γεγονός ότι οι παράμετροι που δουλεύουν για ένα σύνολο δεδομένων μπορεί να μην είναι οι καλύτεροι για κάποιο άλλο σύνολο.

σκευή και ένα χωρίς. Σημειώνουμε ότι όλα τα πειράματα γίνονται για 30 εποχές.

Σύνολο Δεδομένων MNIST

Τα αποτελέσματα των πειραμάτων για το πρόβλημα MNIST παρουσιάζονται στον πίνακα 5.18. Χρησιμοποιούμε μέγεθος δέσμης 32 και παράγοντα κλιμάκωσης σφάλματος ίσο με 0.05 διότι από επιπλέον πειράματα φάνηκε ότι αυτή η τιμή είναι καταλληλότερη για το εν λόγο dataset. Στον πίνακα αποτελεσμάτων αναγράφεται και η εφαρμογή ή μη της συνάρτησης ομαλής μεγιστοποίησης που συνεπάγεται την τήρηση ή μη της υπόθεσης μοναδικού πατέρα (παράμετρος SoftSC). Συμπληρωματικά αναφέρεται ότι ο αριθμός των παραμέτρων του αποκωδικοποιητή από επίπεδα αποσυνέλιξης (όπου χρησιμοποιείται) είναι $250.1k$.

Algorithm	SoftSC	Heads	Param. Count	Reconstruction	Test Error (%)
RoMAV	-	2	162.7	yes (Decon.)	0.34
RoMAV	-	2	162.7	yes (FC)	**0.30***
RoMAV	-	2	162.7	no	0.36
RoWSS	yes	2	162.7	yes (Decon.)	0.36
RoWSS	no	2	162.7	no	0.37
RoWLS	no	2	162.7	yes (Decon.)	0.38
RoWLS	yes	2	162.7	no	0.34

Πίνακας 5.18: Επίδοση των αλγορίθμων της μεθόδου 3 στο σύνολο δεδομένων MNIST, όταν χρησιμοποιούνται 30 εποχές για την εκπαίδευση του μοντέλου με μέγεθος δέσμης 32. Το αποτέλεσμα με αστερίσκο προέκυψε μετά από 60 εποχές.

Παρατηρούμε ότι τα αποτελέσμτατα είναι πολύ ικανοποιητικά (όπως θα δούμε και στο τέλος του κεφαλαίου όπου θα γίνει σύγκριση με άλλα συστήματα που συναντώνται στη βιβλιογραφία). Συγκρίνοντας τους αλγορίθμους της μεθόδου 3 μεταξύ τους, μπορούμε με ασφάλεια να ισχυριστούμε ότι η επίδοση των αλγορίθμων μας στο συγκεκριμένο πρόβλημα είναι παραπλήσια.

Στο συγκεκριμένο dataset χρησιμοποιούμε αρκετά μεγαλύτερο σύνολο δέσμης επειδή το μέγεθος της μνήμης τυχαίας προσπέλασης του επιταχυντή μας το επιτρέπει. Παρατηρήσαμε πολύ πιο σταθερή σύγκλιση και σχεδόν μηδενικές περιπτώσεις αστάθειας[11].

Σύνολο Δεδομένων Fashion-MNIST

Σχετικά με τα πειράματα για το Fashion-MNIST, τα πορίσματά τους αναγράφονται στον πίνακα 5.19. Χρησιμοποιούμε μέγεθος δέσμης 32 και παράγοντα κλιμάκωσης σφάλματος ίσο με 0.05. Ο αριθμός των παραμέτρων του αποκωδικοποιητή από επίπεδα αποσυνέλιξης (όπου χρησιμοποιείται) είναι $250.1k$.

Παρατηρούυμε ότι τα αποτελέσματα είναι και εδώ πολύ ικανοποιητικά μιας και όπως θα δούμε στην συνέχεια, χωρίς καμία αναζήτηση βέλτιστων παραμέτρων για το συγκεκριμένο dataset, ο αλγόριθμός μας είναι ανάμεσα στους καλύτερους σύμφωνα με αυτή τη λίστα.

[11] Σε ορισμένες περιπτώσεις μεγάλης αστάθειας, το σφάλμα γίνεται not-a-number και το πείραμα πρέπει να επαναληφθεί.

Algorithm	SoftSC	Heads	Param. Count	Reconstruction	Test Error (%)
RoMAV	-	2	162.7	yes (Decon.)	8.22
RoMAV	-	2	162.7	no	**7.84**
RoWSS	yes	2	162.7	yes (Decon.)	8.43
RoWSS	no	2	162.7	no	8.38
RoWLS	no	2	162.7	yes (Decon.)	8.32
RoWLS	yes	2	162.7	no	7.87

Πίνακας 5.19: Επίδοση των αλγορίθμων της μεθόδου 3 στο σύνολο δεδομένων Fashion-MNIST, όταν χρησιμοποιούνται 30 εποχές για την εκπαίδευση του μοντέλου με μέγεθος δέσμης 32.

Σημειώνουμε ότι η επίδοση μπορεί να βελτιωθεί με την αύξηση των εποχών αφού από το σύνολο validation φαίνονταν να συνεχίζει να αυξάνεται η επίδοση στις τελευταίες εποχές.

Σύνολο Δεδομένων MultiMNIST

Τα πειραματικά αποτελέσματα για το σύνολο δεδομένων MultiMNIST εμφανίζονται στον πίνακα 5.20. Έχουμε αυτή τη φορά μέγεθος δέσμης 64 και παράγοντα κλιμάκωσης σφάλματος ανακατασκευής ίσο με 0.05. Συμπληρωματικά αναφέρεται και πάλι ότι ο αριθμός των παραμέτρων του αποκωδικοποιητή από επίπεδα αποσυνέλιξης (όπου χρησιμοποιείται) είναι $353.1k$.

Algorithm	SoftSC	Heads	Param. Count	Reconstruction	Test Error (%)
RoMAV	-	2	156.9	yes (Decon.)	**6.2115**
RoMAV	-	2	156.9	no	6.2615
RoWSS	yes	2	156.9	yes (Decon.)	6.6670
RoWSS	no	2	156.9	no	7.2555
RoWLS	no	2	156.9	yes (Decon.)	6.7500
RoWLS	yes	2	156.9	no	6.4050

Πίνακας 5.20: Επίδοση των αλγορίθμων της μεθόδου 3 στο σύνολο δεδομένων MultiMNIST, όταν χρησιμοποιούνται 30 εποχές για την εκπαίδευση του μοντέλου με μέγεθος δέσμης 64.

Αν και η επίδοση δεν είναι καλύτερη από αυτή που εντοπίζεται στη βιβλιογραφία, το γεγονός ότι επιτυγχάνει ένα τόσο χαμηλό ποσοστό σφάλματος συνεπάγεται ότι το δίκτυό μας είναι εύρωστο στην αναγνώριση επικαλυπτόμενων ψηφίων, μια χαρακτηριστική ιδιότητα των νευρωνικών δικτύων με κάψουλες. Με άλλα λόγια, αποδεικνύεται ότι κάθε κάψουλα του τελευταίου επιπέδου μπορεί να λειτουργήσει ανεξάρτητα και να συλάβει το στυλ και την θέση του ψηφίου που αναπαριστά από τις ψήφους που εκλαμβάνει.

Σύνολο Δεδομένων CIFAR10

Τα αποτελέσματα των πειραμάτων για το πιο σύνθετο πρόβλημα CIFAR10 παρουσιάζονται στον πίνακα 5.21. Χρησιμοποιούμε μέγεθος δέσμης 32 και παράγοντα κλιμάκωσης σφάλματος ίσο με 0.05. Επίσης, ακολουθώντας το έργο [76] προσθέσαμε την κατηγορία «κανένα από τα παραπάνω». Συμπληρωματικά αναφέρεται ότι ο αριθμός των παραμέτρων του αποκωδικοποιητή από επίπεδα αποσυνέλιξης (όπου χρησιμοποιείται) είναι $141.7k$.

Το συγκεκριμένο σύνολο δεδομένων είναι το πιο σύνθετο που δοκιμάζουμε. Είναι γεγονός

Algorithm	SoftSC	Heads	Param. Count	Reconstruction	Test Error (%)
RoMAV	-	2	265.2	yes (Decon.)	**28.85**
RoMAV	-	2	265.2	no	29.74
RoWSS	yes	2	265.2	yes (Decon.)	30.01
RoWSS	no	2	265.2	no	30.72
RoWLS	no	2	265.2	yes (Decon.)	28.96
RoWLS	yes	2	265.2	no	30.66

Πίνακας 5.21: Επίδοση των αλγορίθμων της μεθόδου 3 στο σύνολο δεδομένων CIFAR10, όταν χρησιμοποιούνται 30 εποχές για την εκπαίδευση του μοντέλου με μέγεθος δέσμης 32.

ότι τα νευρωνικά δίκτυα από κάψουλες δεν ανταποκρίνονται καλά στο συγκεκριμένο σύνολο. Σε σχέση με άλλους αλγορίθμους της ίδιας τεχνολογίας, η επίδοση είναι χαμηλή (όπως θα δούμε στη συνέχεια). Αυτή η χαμηλή επίδοση μπορεί να προκύπτει από τρείς λόγους (με σειρά προτεραιότητας):

- Μικρός αριθμός εποχών. Στις 30 εποχές παρατηρούνταν ακόμα βελτίωση επίδοσης.
- Απουσία μεθόδων για την μετρίαση του προβλήματος της υπερπροσαρμογής (regulariation methods).
- Χαμηλός αριθμός παραμέτρων δικτύου.

Σύνολο Δεδομένων SmallNORB

Για λόγους πληρότητας, παραθέτουμε τα αποτελέσματα που αφορούν το πρόβλημα SmallNORB, όπως προέκυψαν στην προηγούμενη υπο-ενότητα στον πίνακα 5.22. Υπενθυμίζεται ότι το μέγεθος δέσμης είναι 8 ενώ ο παράγοντας κλιμάκωσης σφάλματος ανακατασκευής ίσος με 0.2. Υπενθυμίζεται επίσης ότι ο αριθμός των παραμέτρων του αποκωδικοποιητή από επίπεδα αποσυνέλιξης (όπου χρησιμοποιείται) είναι $229.5k$.

Algorithm	SoftSC	Heads	Param. Count	Reconstruction	Test Error (%)
RoMAV	-	2	152.8	yes (Decon.)	**1.4403**
RoMAV	-	2	152.8	no	1.5761
RoWSS	yes	2	152.8	yes (Decon.)	1.5309
RoWSS	no	2	152.8	no	1.5597
RoWLS	no	2	152.8	yes (Decon.)	1.5967
RoWLS	yes	2	152.8	no	1.5021

Πίνακας 5.22: Επίδοση των αλγορίθμων της μεθόδου 3 στο σύνολο δεδομένων SmallNORB, όταν χρησιμοποιούνται 30 εποχές για την εκπαίδευση του μοντέλου με μέγεθος δέσμης 8.

Συγκεντρωτικά Αποτελέσματα για Όλα τα Σύνολα Δεδομένων

Για λόγους πληρότητας, παρουσιάζουμε στον πίνακα 5.23 για κάθε σύνολο δεδομένων τον αλγόριθμο της τρίτης μεθόδου που εμφάνισε τα καλύτερα αποτελέσματα (με και χωρίς ανακατασκευαστή).

Dataset	Algorithm	SoftSC	Heads	Parameters (k)	Recon.	Test Error (%)
SmallNORB	RoMAV	-	2	152.8	Decon.	**1.4403**
SmallNORB	RoMAV	-	2	152.8	no	1.5761
MNIST	RoMAV	-	2	162.7	FC	**0.30**
MNIST	RoWLS	yes	2	162.7	no	0.34
Multi-MNIST	RoMAV	-	2	156.9	Decon.	**6.2115**
Multi-MNIST	RoMAV	-	2	156.9	no	6.2615
Fashion-MNIST	RoMAV	-	2	162.7	Decon.	8.22
Fashion-MNIST	RoMAV	-	2	162.7	no	**7.84**
CIFAR10	RoMAV	-	2	265.2	Decon.	**28.85**
CIFAR10	RoMAV	-	2	265.2	no	29.74

Πίνακας 5.23: Οι επιδόσεις των καλύτερων αλγορίθμων της μεθόδου 3 για το κάθε σύνολο δεδομένων.

5.5.4 Ειδικά Πειράματα

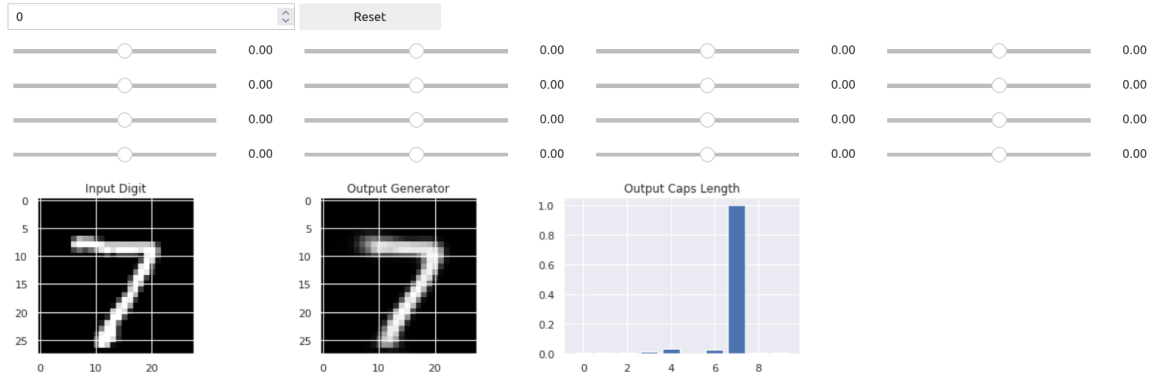
Οι αλγόριθμοι της μεθόδου 3 εμφανίζουν εξαιρετικά αποτελέσματα σε όλα τα σχετικά με τις κάψουλες σύνολα δεδομένων. Το γεγονός αυτό μας προδιαθέτει να πιστέψουμε πως πράγματι, πραγματοποιούν κατά μια έννοια «ανάστροφα γραφικά» και επιτυγχάνουν να αποσυνθέσουν την εικόνα σε εύρωστους παράγοντες συνδιακύμανσης (factors of variation).

Παρακινούμενοι από την επιθυμία μας να αποδείξουμε έμπρακτα την δυνατότητα των αλγορίθμων μας να αποκωδικοποιούν την εικόνα[12] εισόδου σε παραμέτρους στιγμιοτύπου όπως η πόζα και το στύλ, εφαρμόσαμε πειράματα διαταραχής (perturbation) στους αλγορίθμους Multihead RoMAV και Multihead RoWSS αφού εκπαιδεύτηκαν για 60 εποχές με την χρήση πλήρως διασυνεδεμένου αποκωδικοποιητή.

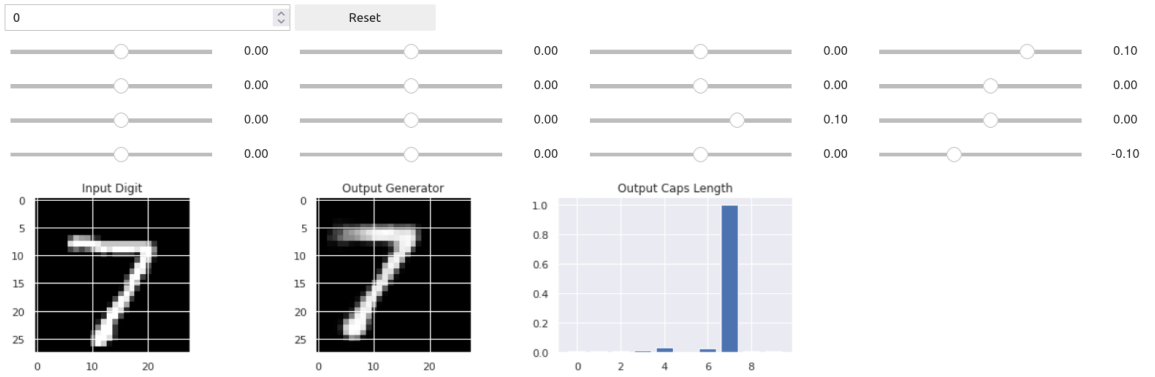
Τα αποτελέσματα και στις δύο περιπτώσεις ήταν πολύ ενθαρρυντικά. Για παράδειγμα, στην εικόνα 5.27 φαίνεται τόσο η αρχική εικόνα όσο και η ανακατασκευασμένη εικόνα από το δίκτυο RoMAV. Στην επόμενη εικόνα, την 5.28 φαίνεται πως μεταβάλλεται το ψηφίο εφτά αν μεταβάλουμε ελαφρώς τις παραμέτρους της αντίστοιχης κάψουλας.

Αντίστοιχα για τον αλγόριθμο RoWSS, στην εικόνα 5.29 παρουσιάζεται το πείραμα διαταραχής για το ψηφίο δύο. Επίσης πραγματοποιήσαμε πειράματα με αποκωδικοποιητή με επίπεδα αποσυνέλιξης που διαθέτει πολύ λιγότερες παραμέτρους (περίπου 250k) αλλά φαίνεται να αδυνατεί να ανακατασκευάσει καλά την εικόνα και να μην βοηθάει στην αποδόμηση της εικόνας του κωδικοποιητή.

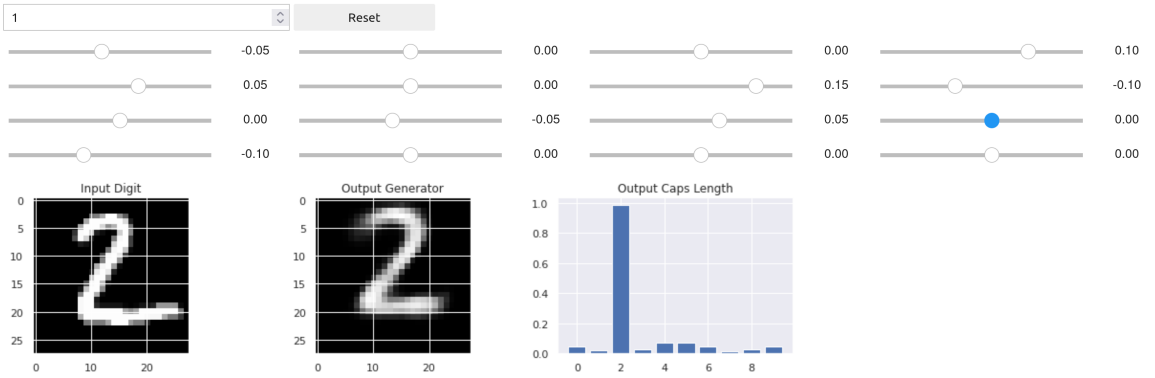
[12]Εικόνα από το σύνολο δεδομένων MNIST



Σχήμα 5.27: Ανακατασκευασμένη εικόνα αλγορίθμου Multihead RoMAV με FC ανακατασκευαστή.



Σχήμα 5.28: Πείραμα διαταραχής αλγορίθμου Multihead RoMAV μμε FC ανακατασκευαστή.



Σχήμα 5.29: Πείραμα διαταραχής αλγορίθμου Multihead RoWSS με FC ανακατασκευαστή.

5.6 Πειραματική Μελέτη Μεθόδου 4

Η τελευταία μέθοδος για την οποία πραγματοποιούμε πειράματα είναι η μέθοδος 4 που αφορά την οικογένεια αλγορίθμων υπό το όνομα «Αλγόριθμος Δρομολόγησης Βασισμένος στον SOM». Υπενθυμίζουμε ότι πρόκειται για τον αλγόριθμο που διαφέρει αρκετά από τα υπόλοιπα νευρωνικά δίκτυα με κάψουλες μιας και οι τελικές κάψουλες στην γενική περίπτωση είναι κοινές για όλα τα δείγματα ενός συνόλου δέσμης. Το χαρακτηριστικό αυτό είναι απαραίτητο λόγω της αργής φύσης του αλγορίθμου SOM που δεν επιτρέπει την εκτέλεσή του για κάθε παράδειγμα εισόδου ξεχωριστά.

Λόγω της ιδιαίτερης φύσης του αλγορίθμου, ο τρόπος προσέγγισης του πειραματικού αυτού μέρους θα είναι σε ένα βαθμό διαφορετικός. Αναλυτικότερα, ξεκινώντας από μια βασική αρχιτεκτονική, επιλεκτικά θα τροποποιούμε ορισμένες παραμέτρους και θα παρατηρούμε την επίδραση που έχει η αλλαγή στην επίδοση του αλγορίθμου. Άλλωστε πρωταρχικός σκοπός της μεθόδου αυτής είναι η διερεύνηση της επιλεκτικής άρσης ορισμένων περιορισμών. Η διερεύνηση αυτή θα γίνει σε ένα σύνολο δεδομένων και για λίγες εποχές αφού αυτά επαρκούν για την εξαγωγή των συμπερασμάτων μας.

Αναφορικά με το περιβάλλον των πειραμάτων, θα χρησιμοποιούμε την μικρή αρχιτεκτονική εκτός και αν αναφέρεται κάτι διαφορετικό. Σε όλα μας τα πειράματα χρησιμοποιούμε βελτιστοποιητή nAdam με ρυθμό μάθησης 0.001. Παρόμοια με την προηγούμενη μέθοδο, αποθηκεύουμε το καλύτερο μοντέλο κατά την διάρκεια της εκπαίδευσης και χρησιμοποιούμε τα βάρη του κατά τον έλεγχο. Μόνο στην περίπτωση των συνόλων MultiMNIST και SmallNORB χρησιμοποιούμε το Margin Loss. Σε όλα τα υπόλοιπα προβλήματα, χρησιμοποιούμε το Categorical Cross Entropy.

Σχετικά με την προεπεξεργασία των συνόλων δεδομένων, αυτή είναι ακριβώς η ίδια με αυτή που εφαρμόσαμε για τα πειράματα της προηγούμενης μεθόδου (ενότητα 5.5). Άρα και πάλι, τα σύνολα δεδομένων για τα οποία η υλοποίησή μας έχει προσαρμοστεί για να υποστηρίζει είναι πέντε: MNIST, MultiMNIST, SmallNORB, CIFAR10 και Fashion-MNIST.

Τέλος, να αναφέρουμε ότι υπάρχει και η δυνατότητα χρήσης ανακατασκευαστή (είτε από πλήρως διασυνεδεμένα επίπεδα είτε από επίπεδα αποσυνέλιξης). Παρόλα αυτά, δεδομένου ότι οι DigitCaps έχουν την ίδια τιμή για κάθε παράδειγμα μέσα σε μια δέσμη (batch ή mini-batch) η ανακατασκευή δεν παρουσιάζει κάποιο όφελος ως προς την επίδοση.

5.6.1 Αναζήτηση στον Χώρο των Υπερπαραμέτρων

Στην ενότητα αυτή κάνουμε μια διερεύνηση του χώρου των υπερπαραμέτρων προκειμένου όχι μόνο να επιλέξουμε μια καλή παραμετροποίηση για τα πειράματα της επόμενης υποενότητας αλλά και για να εξετάσουμε κατά πόσον ορισμένοι περιορισμοί που επιβάλλουν τα νευρωνικά δίκτυα με κάψουλες βοηθούν την επίδοση του αλγορίθμου μας. Για αυτό, θα εξετάσουμε μια μια τις περισότερες υπερπαραμέτρους του αλγορίθμου (και συνδυασμούς αυτών) για 10 εποχές με μέγεθος δέσμης 8 στο σύνολο δεδομένων SmallNORB.

Πιο αναλυτικά, ξεκινώντας από το βασικό default μοντέλο μας, επιλεκτικά θα μεταβάλουμε τις υπερπαραμέτρους και θα εξετάζουμε αν βελτιώνουν την επίδοση. Παράλληλα, θα επιδιώξουμε

να δώσουμε μια εξήγηση για τα αποτελέσματα στις περιπτώσεις που αυτή μπορεί να δοθεί με βεβαιότητα. Συνεπώς, σε κάθε πείραμα δεν θα αναφέρουμε εκ νέου όλες τις παραμέτρους αλλά μόνο αυτές που διαφοροποιούνται από τις προκαθορισμένες τιμές (του βασικού μοντέλου).

Το βασικό μοντέλο που χρησιμοποιούμε, έχει τις εξής τιμές υπερπαραμέτρων:

Theta: 1.0

routing iterations (r): 1

softmax: όχι

reconstructor: όχι (χρήση ή μη ανακατασκευαστή)

deconvolution: όχι (ανακατασκευαστής με αποσυνέλιξη)

small: ναι (χρήση μικρής, εναλλακτικής αρχιτεκτονικής)

non reduced votes: όχι

SOM learning rate: 1.0

radical: όχι

norm type: 0 (squash)

normalize d in loop: όχι

normalize digit caps: όχι

normalize votes: όχι

take into account similarity: όχι

take into accoynt winner ratios: όχι

tanh like: όχι

Το ποσοστιαίο σφάλμα ελέγχου του βασικού μοντέλου (default) εκπαιδευμένου στις 10 εποχές είναι 9.04%.

Υπερπαράμετροι «radical», «small» και «routing iterations»

Τα πειράματα για τις βασικότερες παραμέτρους παρουσιάζονται στον πίνακα 5.24. Η επίδραση της παραμέτρου small είναι προφανής. Η υπερπαράμετρος raical όπως έχουμε εξηγήσει και όπως αναφέρουμε και στο παράρτημα Γ′ αλλάζει τον τρόπο με τον οποίο ενημερώνονται οι ψήφοι ο οποίος διαφέρει από τον κανόνα ενημέρωσης του τυπικού αλγορίθμου SOM.

Η λειτουργία της παραμέτρου routing iterations είναι προφανής. Αυτό που δεν είναι προφανές είναι ο αλγόριθμος που προκύπτει όταν τεθεί η παράμετρος αυτή στο 0. Για τον λόγο αυτό επαναλαμβάνουμε ότι σε αυτή τη περίπτωση, τα διανύσματα των καψουλών αρχικοποιούνται με τυχαίες τιμές. Αυτά τα διανύσματα λέμε ότι ενσωματώνουν τα ανεξάρτητα (invariant) χαρακτηριστικά του εκάστοτε ψηφίου. Ο αλγόριθμος λοιπόν προσπαθεί να παράξει διανύσματα τα οποία δοθέντος μιας εικόνας, να εμφανίζουν μεγάλη ομοιότητα με την διανυσματική αναπαράσταση του σωστού ψηφίου (DigitCap).

Αξίζει, τέλος να σημειώσουμε ότι στην περίπτωση των 0 επαναλήψεων, επειδή δεν κάνουμε ενημέρωση των DigitCaps, η παράμετρος radical δεν αλλάζει την ροή των εντολών.

radical	Routing Iter.	small	Param. Count	Epoch (s)	Step (ms)	Test Error (%)
-	0	yes	670.5k	28	9	**7.11**
-	0	no	670.5k	102	34	8.65
no	1	no	670.5k	29	9	7.28
yes	1	no	670.5k	28	9	29.38
no	1	yes	670.5k	28	9	9.04
yes	1	yes	670.5k	28	9	15.18
no	2	yes	670.5k	29	9	16.11
yes	2	yes	670.5k	28	9	35.75

Πίνακας 5.24: Πειράματα στις υπερπαραμέτρους «radical», «small» και «routing iterations» για σύνολο δεδομένων SmallNORB. Τα πειράματα αυτά πραγματοποιήθηκαν για 10 εποχές με μέγεθος δέσμης ίσο με 8.

Από τα αποτελέσματα, για την κάθε παράμετρο μπορούμε να κάνουμε τις εξής παρατηρήσεις:

- Αναφορικά με την παράμετρο *small*, παρατηρούμε ότι επιβαρύνει σημαντικά το υπολογιστικό κόστος ενώ η επίδοση χειροτερεύει.
- Η παράμετρος *radical* εν γένη δεν βοηθάει στην επίδοση. Συνεπώς, η ενημέρωση των καψουλών του τελευταίου επιπέδου δεν οφελεί να γίνεται απευθείας με βάση τις ψήφους αλλά με βάση την απόσταση των ψήφων από τις εκάστοτε κάψουλες DigitCaps. Η παράμετρος αυτή έχει πιο άμεση ερμηνεία όταν παράγουμε ξεχωριστά διανύσματα DigitCaps για κάθε παράδειγμα εισόδου[13].
- Ο αριθμός των επαναλήψεων δεν προσφέρει κάπια βελτίωση όταν γίνεται μεγάλος. Φαίνεται ότι όσο αυξάνουμε τις επαναλήψεις ο αλγόριθμος δυσκολεύεται όλο και περισσότερο να συγκλίνει. Αυτό διότι τα βάρη που παράγουν τις ψήφους (που στη συνέχεια συγκρίνονται με τα DigitCaps για να προκύψει η πρόβλεψη) δεν μπορούν να εκπαιδευτούν γρήγορα σε στόχους που συνεχώς αλλάζουν. Εντυπωσιακό είναι ότι ο χρόνος εκτέλεσης δεν αυξάνεται. Αυτό διότι η μια επανάληψη παραπάνω γίνεται σε επίπεδο δέσμης και όχι σε κάθε παράδειγμα ξεχωριστά. Συνεπώς, η επιβάρυνση είναι αμελητέα.

Υπερπαράμετρος «non reduced votes»

Όταν έχουμε *reduced votes* (μειωμένες ψήφους) υπενθυμίζεται ότι οι κάψουλες γονείς βλέπουν τις ίδιες ψήφους και όχι διαφορετικές όψεις. Δηλαδή, απομακρυνόμαστε περισσότερο από την τεχνολογία των νευρωνικών δικτύων με κάψουλες που επιβάλλει κάθε κάψουλα γονέας να βλέπει διαφορετικές ψήφους, ψήφους που έχουν προκύψει από την εφαρμογή διαφορετικών μετασχηματισμών (που σχετίζονται με το εκπροσωπούμενο από την κάψουλα αντικείμενο). Στον πίνακα 5.26 καταγράφονται τα αποτελέσματα.

Βλέπουμε ότι ακόμα και στον αλγόριθμο SOM Based Routing (αλγόριθμος 15) που διαφέρει

[13]Τα πειράματα ήταν αρκετά χρονοβόρα και δεν περιλαμβάνονται στη μελέτη μας.

non reduced votes	Param. Count	Epoch (s)	Step (ms)	Test Error (%)
no	670.5k	28	9	15.03
yes (default)	670.5k	28	9	9.04

Πίνακας 5.25: Επίδραση της παραμέτρου *non reduced votes* της μεθόδου 4 στην επίδοση στο σύνολο δεδομένων ελέγχου SmallNORB. Τα πειράματα αυτά πραγματοποιήθηκαν για 10 εποχές με μέγεθος δέσμης ίσο με 8.

αρκετά από τους υπόλοιπους αλγορίθμους που συναντήσαμε στο παρόν έργο, η συγκεκριμένη υπόθεση δεν οφελεί την επίδοση του δικτύου.

Υπερπαράμετρος «softmax»

Όταν χρησιμοποιείται η συγκεκριμενη υπερπαράμετρος, ενημερώνονται όλες οι κάψουλες ανάλογα με τον βαθμό ομοιότητας που παρουσιάζουν με τις παραγόμενες ψήφους. Υπο μια διαφορετική διατύπωση, ο αλγόριθμος SOM Routing έχει soft winners. Τα σχετικά πειράματα εμφανίζονται στον πίνακα 5.26.

softmax	Param. Count	Epoch (s)	Step (ms)	Test Error (%)
no	670.5k	28	9	9.04
yes	670.5k	28	9	**7.19**

Πίνακας 5.26: Πειράματα για την παράμετρο *softmax* της μεθόδου 4 στο σύνολο δεδομένων ελέγχου SmallNORB. Τα πειράματα πραγματοποιήθηκαν για 10 εποχές με μέγεθος δέσμης ίσο με 8.

Παρατηρούμε ότι η παράμετρος αυτή συμβάλλει στην ελάττωση του ποσοστού σφάλματος. Αυτό είναι μια ένδειξη του ότι όσο ερχόμαστε πιο κοντά στην τεχνολογία των νευρωνικών δικτύων με κάψουλες τόσο αυξάνεται η επίδοση. Στην προκειμένη περίπτωση, βλέπουμε ότι η συμμετοχή όλων των ψήφων στην διαμόρφωση της κάθε κάψουλας (και όχι μόνο αυτών που η εκάστοτε κάψουλα εξηγεί καλύτερα) βοηθάει στην καλύτερη σύγκλιση του μοντέλου.

Υπερπαράμετρος «take into account similarity»

Έχοντας παρατηρήσει ότι η ομοιότητα μεταξύ των ψήφων διαδραματίζει καθοριστικό ρόλο στα νευρωνικά δίκτυα με κάψουλες, επιλέξαμε να πειραματιστούμε με το στοιχείο της μέσης ομοιότητας των ψήφων που η κάθε κάψουλα γονέας εξηγεί («κερδίζει») χρησιμοποιώντας τη για να κλιμακώσουμε τις ενημερώσεις των DigitCaps. Τα αποτελέσματα φαίνονται στον πίνακα 5.27.

similarity	Param. Count	Epoch (s)	Step (ms)	Test Error (%)
no (default)	670.5k	28	9	9.04
yes	670.5k	28	9	**7.95**

Πίνακας 5.27: Επίδραση της παραμέτρου *take into account similarity* της μεθόδου 4 στην επίδοση (όπως μετράται από το ποσοστό σφάλματος) στο σύνολο δεδομένων ελέγχου SmallNORB. Τα πειράματα αυτά πραγματοποιήθηκαν για 10 εποχές με μέγεθος δέσμης ίσο με 8.

Μπορούμε με βεβαιότητα να ισχυριστούμε ότι η παράμετρος αυτή έχει ευεργετικές ιδιότητες στην εκπαίδευση του αλγορίθμου. Βέβαια, αξίζει να αναφερθεί ότι το στοιχείο της μέσης ομοιότητας χρησιμοποιείται για την κλιμάκωση των καψουλών εξόδου αλλά η πρόβλεψη δεν υπολογίζεται άμεσα από τις κάψουλες εξόδου (όπως θα ίσχυε στην περίπτωση νευρωνικών δικτύων με κάψουλες ή αν είχαμε ξεχωριστό σετ καψουλών εξόδου για κάθε παράδειγμα εντός της δέσμης - batch). Σε κάθε περίπτωση, το συμπέρασμα που είναι δυνατό να εξάγουμε είναι ότι η εν λόγω κλιμάκωση διευκολύνει την σύγκριση των DigitCaps με της ψήφους.

Υπερπαράμετροι «thetas»

Μέχρι τώρα, με εξαίρεση την περίπτωση που χρησιμοποιούσαμε την επιλογή *softmax*, ακολουθούσαμε την πολιτική «ο νικητής τα παίρνει όλα» (winner take it all). Δηλαδή, η κάψουλα γονέας που εμφάνιζε την μεγαλύτερη ομοιότητα - σε σχέση με τις άλλες κάψουλες γονείς - με την δρομολογούμενη σε αυτόν ψήφο της κάψουλας παιδί «κέρδιζε» την κάψουλα και αυτή τελικά «δρομολογούνταν»[14] εξ ολοκλήρου μόνο στον νικητή. Θέτωντας την παράμετρο ίση με μια λίστα από δύο τιμές μικρότερες ή ίσες της μονάδας, είναι δυνατό η κάθε κάψουλα παιδί να μην «δρομολογείται» μόνο στον γονέα νικητή αλλά και στην κάψουλα που εμφάνισε την δεύτερη μεγαλύτερη ομοιότητα με την κάψουλα παιδί. Φυσικά, το βάρος δρομολόγησης στην δεύτερη κάψουλα θα είναι μικρότερο από αυτό του πρώτου νικητή. Αντίστοιχα, για την περίπτωση που θέτουμε την παράμετρο «thetas» ίση με μια λίστα από τρείς παραμέτρους κ.ο.κ.

Στον πίνακα 5.28 καταγράφονται τα πειράματα για την περίπτωση όπου η παράμετρος είναι μια λίστα με τις τιμές 1.0 και 0.2.

thetas	Param. Count	Epoch (s)	Step (ms)	Test Error (%)
1.0 (default)	670.5k	28	9	9.04
1.0, 0.2	670.5k	28	9	**8.89**

Πίνακας 5.28: Επίδραση της παραμέτρου *thetas* της μεθόδου 4 στην επίδοση στο σύνολο δεδομένων ελέγχου SmallNORB. Τα πειράματα αυτά πραγματοποιήθηκαν για 10 εποχές με μέγεθος δέσμης ίσο με 8.

Από τα πειράματα αυτά, δεν μπορούμε δυστυχώς να εξάγουμε κάποιο ασφαλές συμπέρασμα. Με βάση και άλλα πειράματα που έγιναν φάνηκε μια μικρή βελτίωση όταν δεν χρησιμοποιείται η πολιτική «ο νικητής τα παίρνει όλα».

Υπερπαράμετροι «reconstruction»

Τέλος, δοκιμάσαμε τη χρήση ανακατασκευαστή από πλήρως διασυνδεδεμένα επίπεδα. Αν και όπως είπαμε, τα διανύσματα από κάψουλες δεν ενθυλακώνουν τα συγκεκριμένα (ισομεταβλητά) χαρακτηριστικά της εικόνας εισόδου (πόζα και οπτική γωνία, στυλ κτλ.), ο ανακατασκευαστής μπορεί να βοηθήσει στην κατασκευή πιο εύρωστων διανυσματικών αναπαραστάσεων. Βέβαια, με τη χρήση του, το υπολογιστικό κόστος εκπαίδευσης αυξάνεται σημαντικά. Τα σχετικά πειράματα παρατίθενται στον πίνακα 5.29.

[14] Ουσιαστικά, η δρομολόγηση έγκειται στη συμμετοχή της στη διαμόρφωση της ενημέρωσης για την κάψουλα γονέα που την κέρδισε.

reconstruction	Param. Count	Epoch (s)	Step (ms)	Test Error (%)
no (default)	670.5k	39	13	**9.04**
yes	670.5k	39	13	11.99

Πίνακας 5.29: Επίδραση της παραμέτρου *reconstruction* της μεθόδου 4 στην επίδοση στο σύνολο δεδομένων ελέγχου SmallNORB. Τα πειράματα αυτά πραγματοποιήθηκαν για 10 εποχές με μέγεθος δέσμης ίσο με 8.

Στα αποτελέσματα δε διακρίνουμε βελτίωση γεγονός αναμενόμενο αν αναλογιστούμε ότι τα DigitCaps δεν μεταβάλονται εντός μιας δέσμης.

5.6.2 Επιλεκτική Εμβάθυνση με Δοκιμή σε Όλα τα Προβλήματα

Στην παράγραφο αυτή θα επιλέξουμε τρία από τα μοντέλα που εμφάνισαν καλές επιδόσεις στην προηγούμενη υποενότητα και θα τα εκπαιδεύσουμε για πολλές εποχές σε όλα τα σύνολα δεδομένων. Αυτά περιλαμβάνουν τα MNIST, Fashion-MNIST, MultiMNIST, SmallNORB και το απαιτητικό σύνολο δεδομένων CIFAR10. Τα μοντέλα που θα χρησιμοποιήσουμε είναι τα εξής[15]:

SOM-Based Variant 1: *routing iterations* = 0

SOM-Based Variant 2: *softmax* = yes

SOM-Based Variant 3: *thetas* = 1.0, 0.2

Ο πρώτος αλγόριθμος θέλει να εξετάσει την εκδοχή όπου έχουμε σταθερά, διανύσματα καψουλών αρχικοποιημένα με τη χρήση ομοιόμορφης κατανομής πιθανότητας (random uniform). Οι ψήφοι σε αυτή τη περίπτωση μαθαίνουν να εξάγουν εύρωστες αναπαραστάσεις που παρουσιάζουν μεγάλη ομοιότητα με τις κάψουλες εξόδου. Ο δεύτερος αλγόριθμος χρησιμοποιεί τη παράμετρο softmax προκειμένου να συμμετέχουν όλες οι κάψουλες παιδιά στη διαμόρφωση της κάθε κάψουλας γονέα (δηλαδή, όχι μόνο αυτές οι κάψουλες παιδιά που τις «κέρδισαν»). Η τρίτη έκδοση της μεθόδου είναι όσο πιο πιστή γίνεται στον αλγόριθμο SOM αφού χρησιμοποιούμε και μια (διακριτή) συνάρτηση γειτνίασης.

Πρωτού προχωρήσουμε στην παρουσίαση των αποτελεσμάτων να αναφέρουμε ότι καμία άλλη παράμετρος δεν μεταβλήθηκε. Όλα τα πειράματα έγιναν με τον ίδιο ρυθμό μάθησης (0.001) και τον ίδιο βελτιστοποιητή (nAdam).

Στον πίνακα 5.30 καταγράφονται τα αποτελέσματα της δοκιμής των τριών επιλεγμένων μοντέλων σε όλα τα σύνολα δεδομένων.

Τα συνολικά αποτελέσματα δεν είναι ενθαρρυντικά. Όπως θα δούμε στη συνέχεια, οι επιδόσεις είναι αρκετά χαμηλότερες από αυτές των καλύτερων συστημάτων σήμερα. Κατά την παρατήρηση των καμπυλών εκπαίδευσης, μπορούμε να καταλήξουμε στο ότι οι αλγόριθμοί μας αργούν να συγκλίνουν (βλέπε εικόνα 5.30). Σε ότι αφορά την σύγκριση των τριών παραλλαγών μεταξύ τους, δεν υπάρχει κάποια σαφή διαφορα. Η κάθε παραλλαγή φαίνεται να πηγαίνει καλύτερα σε διαφορετικό σύνολο δεδομένων.

[15] Όποιες παράμετροι δεν αναφέρονται, θεωρείται ότι έχουν την προκαθορισμένη (default) τιμή.

Dataset	SOM Variant	batch size	Test Error (%)
MNIST	1	64	0.53
	2	64	0.68
	3	64	**0.52**
Fashion-MNIST	1	64	9.51
	2	64	9.45
	3	64	**9.51**
MultiMNIST	1	64	**22.954***
	2	64	28.024*
	3	64	27.058*
SmallNORB	1	8	**5.30**
	2	8	5.317
	3	8	6.494
CIFAR10	1	32	34.93
	2	32	**34.555**
	3	32	36.255

Πίνακας 5.30: Συνολικά αποτελέσματα δοκιμής των τριών μοντέλων σε όλα τα σύνολα δεδομένων που υποστηρίζει η τέταρτη μέθοδος για 50 εποχές. Στα πειράματα με αστερίσκο (*) παρατηρήσαμε αστάθεια.

Η μη ικανοποιητική επίδοση του παρόντος αλγορίθμου, σε ένα βαθμό υποδεικνύει τη χρησιμότητα των βασικών υποθέσεων των νευρωνικών δικτύων με κάψουλες. Για παράδειγμα, η πολύ κακή επίδοση που εμφανίζεται στο σύνολο δεδομένων Multi-MNIST μας προδιαθέτει να πιστεύουμε ότι οι περισσότεροι περιορισμοί που εισάγει η τεχνολογία με την οποία καταπιάνεται ο παρόν τόμος οδηγούν σε καλύτερη ταξινόμηση των επικαλυπτώμενων ψηφίων.

Τέλος, να αναφέρουμε ότι η πιο τολμηρή υπόθεση που έγινε από την τέταρτη μέθοδο είναι αυτή της ενθυλάκωσης - ανεξάρτητων από το στιγμιότυπο της κλάσης - χαρακτηριστικών. Μέσα από τα πειράματα αποδείχθηκε ότι κάτι τέτοιο είναι απαραίτητο για την ορθή λειτουργία του νευρωνικού δικτύου. Επιδιώξαμε να αναιρέσουμε αυτή την υπόθεση στην τέταρτη μέθοδο και να εφαρμόζουμε τον αλγόριθμο SOM σε κάθε εικόνα εισόδου ξεχωριστά. Παρόλα αυτά, το μεγάλο υπολογιστικό κόστος του αλγορίθμου SOM κατέστησε την μέθοδο μη αποδοτική, γεγονός το οποίο ξέφευγε από τους σκοπούς της παρούσας διπλωματικής εργασίας.

Σχήμα 5.30: Σφάλμα εκπαίδευσης και ελέγχου μεθόδου 4 στο σύνολο MNIST.

5.7 Σύγκριση Πειραματικών Αποτελεσμάτων Μεθόδων

Έχωντας ολοκληρώσει όλα τα πειράματα για την κάθε μέθοδο ξεχωριστά και έχοντας συγκρίνει τους αλγορίθμους της κάθε μεθόδου μεταξύ τους, είμαστε πλέον σε θέση να συγκρίνουμε τους καλύτερους δικούς μας αλγορίθμους με αυτούς που βρίσκονται στη διεθνή βιβλιογραφία και εμφανίζουν την καλύτερη επίδοση την στιγμή της συγγραφής του έργου. Δεδομένου ότι ασχολούμαστε με μια ειδική κατηγορία τεχνητών νευρωνικών δικτύων με συγκεκριμένες ιδιότητες, θα γίνει σύγκριση και με τις πιο αξιόλογες υλοποιήσεις της εν λόγω τεχνολογίας.

Σε κάθε υποενότητα, πραγματοποιούμε τη σύγκριση σε καθένα πρόβλημα ξεχωριστά. Ο τρόπος με τον οποίο παρουσιάζουμε τα αποτελέσματα είναι ο εξής:

1. Στην ανώτερη βαθμίδα παρουσιάζονται τα αποτελέσματα των δικών μας αλγορίθμων.
2. Στην δεύτερη βαθμίδα παρουσιάζονται τα αποτελέσματα των «βασικών» αλγορίθμων από την ομάδα του Hinton αλλά και αυτά που αφορούν το έργο [49] από το οποίο εμπνεύστηκε η μέθοδος 3.
3. Στην τρίτη βαθμίδα καταγράφονται οι επιδόσεις άλλων υλοποιήσεων των νευρωνικών δικτύων από κάψουλες που βρίσκονται στην βιβλιογραφία.
4. Τέλος, στην κατώτερη βαθμίδα παρουσιάζεται το μοντέλο με την καλύτερη επίδοση για το

συγκεκριμένο dataset.

Προτού ξεκινήσουμε την ανάλυση για κάθε σύνολο δεδομένων ξεχωριστά, είναι σημαντικό να επαναλάβουμε ότι αν εξαιρέσουμε την τρίτη μέθοδο (και ως ένα βαθμό την τέταρτη) οι υπόλοιπες από τις μεθόδους μας δεν έχουν σκοπό την επίτευξη καλύτερης επίδοσης αλλά την διευκόλυνση στην κατανόηση του τρόπου λειτουργίας των νευρωνικών δικτύων με κάψουλες. Επίσης, διευκολύνουν στην σύγκριση με τις μεθόδους μας αφού μέσα από τον πειραματισμό τους αποκτήσαμε χρήσιμες μετρικές που ήταν αδύνατο να αποκομιστούν διαφορετικά. Ακόμα, για τις μεθόδους που προτείνουμε εμείς, τόσο η αναζήτηση στον χώρο παραμέτρων όσο και η εκπαίδευση (από πλευράς χρόνου) δεν είναι πλήρεις και καλύτερες επιδόσεις μπορούν να προκύψουν με μια πιο εκτενή πειραματική μελέτη.

5.7.1 Σύνολο Δεδομένων SmallNORB

Το σύνολο δεδομένων SmallNORB αποτελεί μια χαρακτηριστική δοκιμασία για τα νευρωνικά δίκτυα με κάψουλες αφού μέσω αυτού δοκιμάζεται η ικανότητά τους για αναγνώριση αντικειμένων από διαφορετικές οπτικές γωνίες. Ας μην ξεχνάμε ότι η αποδοτική αναγνώριση τέτοιων εικόνων είναι πρωταρχικός στόχος της συγκεκριμένης τεχνολογίας.

Ο αριθμός των παραμέτρων δεν μαρτυρά πάντα την υπολογιστική πολυπλοκότητα του δικτύου, ειδικά στην περίπτωσή μας όπου χρησιμοποιούνται επαναληπτικοί αλγόριθμοι δρομολόγησης. Για τον σκοπό αυτό, ειδικά για το συγκεκριμένο πρόβλημα κάναμε επιπλέον μετρήσεις του αριθμού των πράξεων κινητής υποδιαστολής που πραγματοποιούνται το δευτερόλεπτο, για ένα batch μεγέθους 1. Για να μετριάσουμε την υπολογιστική επιβάρυνση που προστίθεται κατά την αρχικοποίηση των μητρώων (tensors), όλες οι μετρήσεις έγιναν για μέγεθος δέσμης ίσο με 8 και το αποτέλεσμα το διαιρούμε με το 8 για να λάβουμε την τιμή FLOPs per 1 batch. Προφανώς, επειδή οι μετρήσεις μας αφορούν το κόστος πρόβλεψης (inference), στις μετρήσεις δεν περιλαμβάνεται το υπολογιστικό κόστος του αποκωδικοποιητή (όπου χρησιμοποιείται).

Στον πίνακα 5.31 παρουσιάζονται οι επιδόσεις τόσο των δικών μας μεθόδων όσο και μεθόδων που συναντώνται στην βιβλιογραφία. Στη τελευταία βαθμίδα παρουσιάζεται το έργο με την καλύτερη επίδοση. Αν τυχαίνει να είναι και νευρωνικό δίκτυο με κάψουλες (όπως στην περίπτωσή μας), θα εμφανίζεται δύο φορές στην λίστα (μια επιπλέον στην τρίτη βαθμίδα όπου παρατίθενται οι επιδόσεις των Capsule Networks της βιβλιογραφίας).

Τα αποτελέσματα των μεθόδων μας είναι πλήρως ικανοποιητικά, αν εξαιρέσουμε την τέταρτη μέθοδο. Ο αλγόριθμος Argmax Scaled της πρώτης μεθόδου μπορεί να μην εμφανίζει καλά αποτελέσματα αλλά πρέπει να λάβουμε υπόψη ότι δεν έγινε καμία ειδική αναζήτηση υπερπαραμέτρων για το συγκεκριμένο σύνολο δεδομένων στην πρώτη μέθοδο. Συνεπώς, το γεγονός ότι έχει καλύτερη επίδοση από την δική μας υλοποίηση του αλγορίθμου του έργου [76] όταν οι δύο εκπαιδεύτηκαν υπό τις ίδιες συνθήκες ενδέχεται να σημαίνει ότι τελικά, μπορεί να εμφανίζει καλύτερη επίδοση.

Την μεγαλύτερη επιτυχία την σημειώνει η τρίτη μέθοδος η οποία μετά από εκπαίδευση σε μόλις 30 εποχές και χωρίς ιδιαίτερα αντίμετρα υπερπροσαρμογής, ανταγωνίζεται τις καλύτερες μεθόδους της βιβλιογραφίας (τρίτη σε σειρά επίδοσης). Η επίδοση 1.44 (20.0% μείωση ποσοστιαίου σφάλματος σε σχέση με τον EM) επιτυγχάνεται με λιγότερο από το ένα δέκατο της υπολογιστικής

Algorithm (Method)	Recon.	Parameters (K)	$FLOPs\|_{1batch}$ (G)	SmallNORB (%)
Argmax Scaled (1)	yes	6816	0.705	16.49
Multi-Head RoMAV (3)	yes	152.8	0.0473	**1.44**
Multi-Head RoMAV (3)	no	152.8	0.0473	1.58
SOM-Based Var.1 (4)	no	670.5	0.889	5.30
Our Dynamic (1)	yes	6800	0.7	17.01
Our EM-Routing (2)	no	310	0.675	14.04
Dynamic [76]	yes	6800	0.7	2.7
EM-Routing [47]	no	310	0.675	**1.8**
Efficient-CapsNet [49]	yes	151	0.06	2.54
VB-Routing [99]	no	169	-	1.6
RU-Routing [98]	no	140	-	1.4
DC-Net [133]	yes	11800	-	5.57
DC-Net++ [133]	yes	11800	-	4.66
FREM [134]	no	1200	-	2.2
FRMS [134]	no	1200	-	2.6
STAR-Caps [135]	no	318	-	1.8
Heinsen Routing [131]	no	272	-	**0.9**
Heinsen Routing [131]	no	272	-	**0.9**

Πίνακας 5.31: Σύγκριση της επίδοσης των μεθόδων μας με αυτές στην βιβλιογραφία για το σύνολο δεδομένων SmallNORB.

πολυπλοκότητας (93% βελτίωση) και τις μισές παραμέτρους. Ακόμα και χωρίς ανακατασκευαστή, οι επιδόσεις είναι πολύ υψηλές, γνωρίζοντας μάλιστα ότι η μέθοδος επιδέχεται περαιτέρω βελτίωση. Αν και στοιχεία υπολογιστικής πολυπλοκότητας απουσιάζουν για τις περισσότερες μεθόδους, η μη επαναληπτική φύση του αλγορίθμου μας οδηγεί να πιστεύουμε ότι ο αλγόριθμος Multi-Head RoMAV της τρίτης μεθόδου είναι αυτός με την καλύτερη σχέση επίδοσης–ταχύτητας στην βιβλιογραφία στο συγκεκριμένο πρόβλημα. Τέλος, εφιστούμε την προσοχή στο αποτέλεσμα της μεθόδου Efficient-CapsNet [49] που χρησιμοποιεί τον «απλοικό αλγόριθμο προσοχής» της μεθόδου 3 (αλγόριθμος 6). Βλέπουμε ότι με σχεδόν τις ίδιες παραμέτρους και πράξεις κινητής υποδιαστολής επιτυγχάνουμε πολύ υψηλότερα αποτελέσματα χρησιμοποιώντας αλγορίθμους με πολύ καλύτερη ποιοτική ερμηνεία.

5.7.2 Σύνολο Δεδομένων MNIST

Το συγκεκριμένο σύνολο αποτελεί ένα από τα ευκολότερα σύνολα δεδομένων για τα σημερινά νευρωνικά δίκτυα και το πρόβλημα είναι σε μεγάλο βαθμό «λυμένο». Ο ρόλος του στα νευρωνικά δίκτυα με κάψουλες είναι κυρίως στο να αποδείξουν ότι μπορούν να αποδομήσουν τα ψηφία στις παραμέτρους στιγμιοτύπου τους. Επίσης, χρησιμοποιείται ως βάση σε άλλα σύνολα δεδομένων όπως το affNIST και το MultiMNIST. Τα σχετικά πειράματα παρουσιάζονται στον πίνακα 5.32.

Στο πρόβλημα MNIST ο αλγόριθμος με την καλύτερη επίδοση είναι πάλι ο Multi-Head RoMAV της τρίτης μεθόδου. Το υψηλό αποτέλεσμα είναι ιδιαίτερα αξιόλογο αν αναλογιστεί κανείς ότι δεν έγινε καμία αναζήτηση υπερπαραμέτρων για το συγκεκριμένο πρόβλημα και ότι η εκπαίδευση έγινε μόλις για 30 εποχές. Συνεπώς, παρατηρούμε ότι υπό την ίδια παραμετροποίηση ο

Algorithm (Method)	Recon.	Parameters (K)	MNIST (%)
Argmax Scaled (1)	yes	6816	0.37
Multi-Head RoMAV (3)	yes	162.7	**0.3**
Multi-Head RoWLS (3)	no	162.7	0.34
SOM-Based Var.3 (4)	no	670	0.52
Our Dynamic (1)	yes	6800	0.31
Dynamic [76]	yes	6800	0.25
Dynamic [76]	no	6800	0.35
EM-Routing [47]	no	310	0.44
Efficient-CapsNet [49]	yes	161	**0.26**
VB-Routing [99]	no	169	0.3
RU-Routing [98]	no	> 440	**0.28**
DC-Net [133]	yes	11800	0.28
DC-Net++ [133]	yes	11800	0.29
FREM [134]	no	1200	0.38
STAR-Caps [135]	no	281	0.43
DA-CapsNet [109]	yes	7000	0.47
HitNet [136]	yes	$>= 8000$	0.32
Nair [137]	no	8200	0.5
DeepCaps [106]	yes	7220	0.28
CNN Ensemble [129]	no	∞	**0.09**

Πίνακας 5.32: Σύγκριση της επίδοσης των μεθόδων μας με αυτές στην βιβλιογραφία για το σύνολο δεδομένων MNIST.

ίδιος αλγόριθμος που πετυχαίνει ανταγωνιστικά για τη βιβλιογραφία αποτελέσματα στο σύνολο SmallNORB έχει πολύ καλές επιδόσεις και στο σύνολο MNIST[16]. Είναι σημαντικό να αναφέρουμε ότι σχεδόν όλοι οι αλγόριθμοι της βιβλιογραφίας με καλύτερη επίδοση δεν αποδεικνύουν την αποδόμηση των εικόνων του συνόλου δεδομένων σε ισομεταβλητές παραμέτρους (equivariant parameters) όπως εμείς κάναμε. Επιπλέον, προσθέτουμε ότι από στοιχεία που αποκτήσαμε κατά την διάρκεια της εκπαίδευσης, πιστεύουμε ότι η επίδοση μπορεί να βελτιωθεί περαιτέρω. Τέλος, σημειώνουμε ότι σε σχέση με την καλύτερη υλοποίηση από τα βασικά έργα ([47, 76, 120]) χωρίς ανακατασκευαστή επιτυγχάνονται καλύτερα αποτελέσματα.

5.7.3 Σύνολο Δεδομένων CIFAR10

Πρόκειται για το πιο σύνθετο σύνολο δεδομένων. Είναι σύνηθες τα νευρωνικά δίκτυα με κάψουλες να μην έχουν υψηλή επίδοση σε αυτό. Τα σχετικά αποτελέσματα παρουσιάζονται στον πίνακα 5.33.

Στο συγκεκριμένο σύνολο, οι επιδόσεις μας είναι χειρότερες από τις περισσότερες της βιβλιογραφίας. Αυτό πιθανότατα οφείλεται αφενός στον μικρό αριθμό εποχών και αφετέρου στον χαμηλό αριθμό παραμέτρων που έχουμε (για τις μεθόδους 3 και 4). Φυσικά, επειδή για τους αλγορίθμους της μεθόδου 3, δεν έγινε καμία αναζήτηση υπερπαραμέτρων, η επίδοση πιθανότατα επιδέχεται βελτίωση.

[16]Με μικρές διαφορές στην αρχιτεκτονική των πρώτων επιπέδων η οποία τώρα δεν περιέχει επίπεδα Instance Normalization.

Algorithm (Method)	Recon.	Parameters (K)	CIFAR10 (%)
Argmax Scaled (1)	yes	7000	22.16
Multi-Head RoMAV (3)	yes	265.2	28.85
Multi-Head RoMAV (3)	no	265.2	29.74
SOM-Based Var.2 (4)	no	670	34.555
Our Dynamic (1)	yes	7000	**21.64**
Dynamic [76]	yes	6800	**10.6**
EM-Routing [47]	no	460	11.9
VB-Routing [99]	no	323	11.2
DC-Net [133]	yes	11800	17.37
DC-Net++ [133]	yes	11800	10.29
FREM [134]	no	1200	14.3
FRMS [134]	no	1200	15.6
STAR-Caps [135]	no	281	**8.77**
DA-CapsNet [109]	yes	7000	14.53
HitNet [136]	yes	>= 8000	26.70
Nair [137]	no	8200	32.47
DeepCaps [106]	yes	7220	8.99
ViT-H/14 [81]	no	∞	**0.5**

Πίνακας 5.33: Σύγκριση της επίδοσης των μεθόδων μας με αυτές στην βιβλιογραφία για το σύνολο δεδομένων CIFAR10.

5.7.4 Σύνολο Δεδομένων MultiMNIST

Για το σύνολο αυτό, δεν υπάρχουν αρκετές μέθοδοι με τις οποίες μπορεί να συγκριθεί. Η δυσκολεία υλοποίησης του συγκεκριμένου προβλήματος έχει αποτρέψει πολλούς ερευνητές στον να δοκιμάσουν το έργο τους σε αυτό. Τα πειράματα για το σύνολο αυτό και η σύγκρισή τους εμφανίζοντια στον πίνακα 5.34. Όπως και για το σύνολο SmallNORB, η μέθοδος με την καλύτερη επίδοση είναι μια μέθοδος νευρωνικών δικτύων με κάψουλες.

Δοκιμάσαμε τις νέες μεθόδους που προτείνουμε (3 και 4) στο MultiMNIST και τα αποτελέσματα είναι πολύ διαφωτιστικά. Βλέπουμε ότι η μέθοδος 4 που απέχει αρκετά από τις θεμελιώδεις υποθέσεις των νευρωνικών δικτύων με κάψουλες αποτυγχάνει στο συγκεκριμένο σύνολο. Στον αντίποδα βρίσκεται η μέθοδος 3 η οποία έχει εξαιρετικά αποτελέσματα δεδομένου ότι δεν έγινε καμία αναζήτηση υπερπαραμέτρων και καμία προσαρμογή της αρχιτεκτονικής του δικτύου σε αυτό το ιδιαίτερο σύνολο δεδομένων. Είμαστε βέβαιοι ότι με την χρήση καλύτερου αποκωδικοποιητή και περισσότερων εποχών τα αποτελέσματα της τρίτης μεθόδου θα βελτιωθούν[17]

[17]Παρατηρήσαμε πως κατά την εκπαίδευση σε 30 εποχές το σφάλμα ελέγχου είχε συνεχώς καθοδική τάση. Δυστυχώς, περιορισμοί χρονικοί και υλικού δεν μας επέτρεψαν τον εκτενή πειραματισμό των 6 αλγορίθμων της μεθόδου 3 στο συγκεκριμένο σύνολο.

Algorithm (Method)	Recon.	MultiMNIST (%)
Argmax Scaled (1)	yes	-
Multi-Head RoMAV (3)	yes	**6.2115**
Multi-Head RoMAV (3)	no	6.2615
SOM-Based Var.3 (4)	no	22.954
Dynamic [76]	yes	**5.0**
Efficient-CapsNet [49]	yes	5.1
RU-Routing [98]	no	**1.8**
Aff-Caps [138]	no	4.51
Aff-Caps [138]	yes	5
RU-Routing [98]	no	**1.8**

Πίνακας 5.34: Σύγκριση της επίδοσης των μεθόδων μας με αυτές στην βιβλιογραφία για το σύνολο δεδομένων MultiMNIST.

5.7.5 Σύνολο Δεδομένων FashionMNIST

Το σύνολο FashionMNIST αποτελεί μια πιο σύνθετη εκδοχή του MNIST και στα πειράματα σκοπό έχει να δοκιμάσει κυρίως την ικανότητα των νευρωνικών δικτύων με κάψουλες για αναγνώριση αντικειμένων υπό διαφορετικά στυλ. Αυτό διότι οι εικόνες που περιέχει είναι εικόνες «πρόσοψης» και δεν έχουν ληφθεί από διαφορετικές οπτικές γωνίες. Στον πίνακα 5.35 συγκεντώνονται τα αποτελέσματα των πειραμάτων μας αλλά και αυτά της βιβλιογραφίας. Όπως πάντα, στην τελευταία βαθμίδα έχουμε την καλύτερη μέθοδο της βιβλιογραφίας που στην περίπτωσή μας, δεν είναι νευρωνικό δίκτυο από κάψουλες.

Algorithm (Method)	Recon.	FashionMNIST (%)
Argmax Scaled (1)	yes	7.00
Multi-Head RoMAV (3)	yes	8.22
Multi-Head RoMAV (3)	no	7.84
SOM-Based Var.3 (4)	no	9.51
Our Dynamic (1)	yes	**6.94**
VB-Routing [99]	no	**5.2**
DC-Net [133]	yes	5.36
DC-Net++ [133]	yes	5.35
FREM [134]	no	6.2
FRMS [134]	no	6.0
DA-CapsNet [109]	yes	6.02
HitNet [136]	yes	7.7
Nair [137]	no	10.2
DeepCaps [106]	yes	5.54
Fine-Tuning DARTS [130]	no	**3.09**

Πίνακας 5.35: Σύγκριση της επίδοσης των μεθόδων μας με αυτές στην βιβλιογραφία για το σύνολο δεδομένων FashionMNIST.

Και πάλι, οι λίγες εποχές που χρησιμοποιούνται σε συνδυασμό με τον μειωμένο αριθμό παρα-

μέτρων οδηγούν σε χαμηλότερη επίδοση (υψηλότερο ποσοστό σφάλματος) σε σχέση με πολλές μεθόδους της βιβλιογραφίας. Επίσης, η έντονη επαύξηση των δεδομένων (με περιστροφή) που χρησιμοποιούμε στο συγκεκριμένο σύνολο δεδομένων (για τις μεθόδους 3 και 4) δεν διευκολύνει την επίδοση. Ειδικά για την τρίτη μέθοδο, το γεγονός ότι η προσθήκη ανακατασκευαστή επιδεινώνει το ποσοστιαίο σφάλμα μας προδιαθέτει στο να πιστέψουμε ότι καλύτερη αναζήτηση υπερπαραμέτρων θα προσφέρει υψηλότερη ακρίβεια.

Κεφάλαιο 6

Επίλογος

6.1 Σύνοψη και Συμπεράσματα

Στην παρούσα διπλωματική εργασία, μελετήσαμε σε βάθος τα νευρωνικά δίκτυα με κάψουλες με σκοπό την κατανόηση των εσωτερικών, κρυφών λειτουργιών τους αλλά και την περαιτέρω βελτίωση της απόδοσής τους προτείνοντας μια κλιμακώσιμη παραλλαγή.

Αρχικά, περιγράψαμε με προσιτό τρόπο τον γενικότερο χώρο της μηχανικής μάθησης, ξεκινώντας από την ιδέα της εκμάθησης χαρακτηριστικών και καταλήγοντας στις πιο σύγχρονες αρχιτεκτονικές βαθιών νευρωνικών δικτύων. Φυσικά, αφιερώσαμε μεγάλη έκταση του έργου στην παρουσίαση της κυρίαρχης τεχνολογίας με την οποία καταπιάνεται η παρούσα εργασία, δηλαδή αυτής των νευρωνικών δικτύων με κάψουλες.

Μέσα από τη θεωρητική μελέτη, μάθαμε μεταξύ άλλων ότι τα - βιολογικά εμπνευσμένα - νευρωνικά δίκτυα με κάψουλες[1] αποδομούν τα απεικονιζόμενα αντικείμενα (στιγμιότυπα) μιας εικόνας σε τμήματά τους με τρόπο ώστε να τα αναγνωρίζουν αποδοτικά υπό διαφορετικές οπτικές γωνίες. Επισημάναμε ότι σε ένα τέτοιο δίκτυο, τα χαμηλότερα επίπεδά του περιέχουν κάψουλες που αναπαριστούν απλά τμήματα αντικειμένων. Η μετάβαση σε ανώτερα επίπεδα περιλαμβάνει την ψηφοφορία των καψουλών χαμηλότερου επιπέδου για τις ανώτερες κάψουλες με βάση ποια από αυτές εκτιμούν ότι περιέχει το τμήμα του αντικειμένου που έχουν μάθει να αναπαριστούν. Αυτές οι ψήφοι, μέσω ενός αργού, επαναληπτικού αλγορίθμου δρομολόγησης που εντοπίζει την «κοινή γνώμη» (δηλαδή τη μεταξύ τους συμφωνία) ενεργοποιούν επιλεκτικά τις κάψουλες επόμενου επιπέδου, συνθέτοντας έτσι μια ιεραρχία μεταξύ μερών και αντικειμένων που ενθαρρύνει την εύρωστη μοντελοποίησή τους.

Συνεχίζουμε με τη βιβλιογραφική επισκόπηση στην οποία αναφερθήκαμε σε 30 συνολικά σχετικές με το θέμα εργασίες. Από τη μελέτη τους, εντοπίσαμε τα δύο προβλήματα που αποτελούν και στόχους της εργασίας, δηλαδή, αυτό της αδυναμίας κλιμάκωσης και αυτό της απουσίας πρακτικών αποδείξεων των ισχυρισμών των νευρωνικών δικτύων με κάψουλες. Συμπληρωματικά σημειώνουμε ότι μέσα από το εν λόγω κεφάλαιο διαπιστώσαμε ότι η εισαγωγή λίγων επιπλέον συνελικτικών επιπέδων στην αρχή του νευρωνικού δικτύου βοηθάει στην επίδοση.

Μέσα από την πειραματική μελέτη των δημοφιλών συστημάτων που θεμελιώνουν την τεχνολογία, παρατηρήσαμε ότι οι ισχυρισμοί ισχύουν όταν το σύνολο δεδομένων είναι απλό (π.χ. MNIST). Συγκεκριμένα, μέσω πειραμάτων διαταραχής (perturbation tests) αποδείξαμε ότι τα νευρωνικά δίκτυα με κάψουλες που περιλαμβάνουν αποκωδικοποιητές με αρκετές παραμέτρους (της τάξης

[1] Οι κάψουλες είναι συστάδες νευρικών αποκρίσεων που αναπαριστούν αντικείμενα ή μέρη τους και ενθυλακώνουν ισομεταβλητά χαρακτηριστικά τους όπως η θέση ή ο προσανατολισμός.

των εκατοντάδων χιλιάδων) πράγματι δημιουργούν εύρωστες εσωτερικές αναπαραστάσεις που μεταβάλλονται ανάλογα με τις αλλαγές στην οπτική γωνία και το στυλ του αντικειμένου. Με πειράματα και σε άλλα σύνολα δεδομένων, φάνηκε ότι η συγκεκριμένη υπόθεση ισχύει ακλόνητα όταν για την εκπαίδευση χρησιμοποιούνται σύνολα δεδομένων που περιέχουν αντικείμενα υπό πολλές βαθμιαίες μεταβολές στις παραμέτρους στιγμιότυπου των απεικονιζόμενων αντικειμένων. Αντίθετα, δεν μπορούμε να ισχυριστούμε το ίδιο για σύνολα δεδομένων με σύνθετο παρασκήνιο (όπως το CIFAR10).

Μια ακόμα ενδιαφέρουσα υπόθεση που εξετάσαμε είναι αυτή σύμφωνα με την οποία ο αλγόριθμος δρομολόγησης «φιλτράρει» τις ψήφους ανάλογα με τη συμφωνία μεταξύ τους και επιλέγει την κάψουλα ανώτερου επιπέδου που εκφράζει την «κοινή γνώμη». Μέσα από καινοτόμα πειράματα και εργαλεία που φανερώνουν όλες τις εσωτερικές λειτουργίες του δικτύου, επιβεβαιώσαμε ότι πράγματι, ο αλγόριθμος δρομολόγησης εφαρμόζει φιλτράρισμα βάση (πολυδιάστατης) συμφωνίας. Παρόλα αυτά, η ψηλότερη μέση συμφωνία των ψήφων για μια κάψουλα τελευταίου επιπέδου είδαμε ότι δεν αποτελεί την καλύτερη ένδειξη για το ποια τελικά κάψουλα θα επιλεγεί (ειδικά για πιο σύνθετα σύνολα δεδομένων, όπως το Fashion-MNIST που δοκιμάσαμε).

Σε ό,τι αφορά το πρόβλημα της κλιμακωσιμότητας, προτείναμε δύο μεθόδους (μέθοδοι 3 και 4) που μέσα από την τροποποίηση του αλγορίθμου δρομολόγησης, αποδεδειγμένα μειώνουν σημαντικά το υπολογιστικό κόστος. Η 4^η μέθοδος, αν και επιστημονικά θεμελιωμένη σε μια αποδοτική εκδοχή του αλγορίθμου χαρτών αυτο–οργάνωσης Kohonen, δεν εμφανίζει ανταγωνιστικά αποτελέσματα υπό τις παραμετροποιήσεις που εξετάσαμε.

Αντιθέτως, η 3^η μέθοδος που ενσωματώνει έναν μη–επαναληπτικό αλγόριθμο δρομολόγησης εμπνευσμένο από τον δημοφιλή μηχανισμό προσοχής πολλών κεφαλών, εμφανίζει πολύ υποσχόμενα αποτελέσματα σχεδόν σε όλα τα προβλήματα–ορόσημο που δοκιμάστηκε. Για παράδειγμα, με ελάχιστη αναζήτηση στον χώρο των υπερπαραμέτρων και μετά από εκπαίδευση σε μόλις 30 εποχές, επιτυγχάνει μείωση σφάλματος ταξινόμησης κατά 20% από το αντίστοιχο ποσοστό του δημοφιλούς αλγορίθμου δρομολόγησης «EM Routing» στο benchmark smallNORB με μόλις το $\frac{1}{10}$ του υπολογιστικού κόστους ανά δευτερόλεπτο (μετρούμενο σε FLOPs) και με τις μισές παραμέτρους. Επιπρόσθετα, μέσα από πειράματα διαταραχής (perturbation tests) παρατηρήσαμε ότι πράγματι το δίκτυο που προτείνουμε, όταν εκπαιδεύεται με τη χρήση πλήρως διασυνδεδεμένου ανακατασκευαστή, πραγματοποιεί μια μορφή ανάστροφων γραφικών αφού εξάγει ισομεταβλητά (equivariant) χαρακτηριστικά των αντικειμένων εισόδου. Σαν παράπλευρα αποτελέσματα, τονίσαμε τη σημασία της χρήσης πλήρως διασυνδεδεμένων ανακατασκευαστών για τα νευρωνικά δίκτυα με κάψουλες ενώ παράλληλα, αναφέραμε ότι η υπόθεση μοναδικού πατέρα (σύμφωνα με την οποία κάθε κάψουλα χαμηλότερου επιπέδου ανήκει μόνο σε μια κάψουλα αμέσως ψηλότερου επιπέδου) δεν οδηγεί πάντα σε καλύτερα αποτελέσματα.

Τέλος, πιστοποιούμε ότι η τρίτη μέθοδος πληροί όλες τις χαρακτηριστικές ιδιότητες των νευρωνικών δικτύων με κάψουλες, υποδεικνύοντας έτσι έμπρακτα ότι ο προτεινόμενος αλγόριθμος δρομολόγησης πιθανώς μπορεί να αποτελέσει το δομικό στοιχείο για αποδοτικότερα μεγάλα συστήματα μηχανικής μάθησης.

6.2 Μελλοντικές Κατευθύνσεις

Ευελπιστούμε ότι τα ελκυστικά αποτελέσματα της $3^{ης}$ μας μεθόδου σε συνδυασμό με την ευνόητη επεξήγηση των νευρωνικών δικτύων με κάψουλες, θα παρακινήσει την ακαδημαϊκή κοινότητα να δώσει προσοχή σε αυτήν την πολλά υποσχόμενη τεχνολογία.

Μια άμεση προέκταση που θα μπορούσε να έχει το παρόν έργο είναι η περαιτέρω πειραματική αναζήτηση καλύτερων επιδόσεων της $3^{ης}$ μεθόδου στον χώρο των υπερπαραμέτρων για όλα τα υποστηριζόμενα σύνολα δεδομένων, ξεχωριστά. Αν και οι περιορισμένοι πόροι που διαθέταμε δε μας επέτρεψαν να διενεργήσουμε επιπλέον πειράματα, είμαστε πεπεισμένοι ότι με ελάχιστες ή καθόλου αλλαγές, η μέθοδος αυτή θα εμφανίσει ακόμα καλύτερα αποτελέσματα.

Μια ακόμα σημαντική επέκταση της διπλωματικής αυτής είναι η σύνθεση ενός βαθύτερου συστήματος, χρησιμοποιώντας την τρίτη μέθοδο σαν δομικό στοιχείο και αντλώντας έμπνευση από τις ιδέες του Hinton G. που παρατίθενται στο [77]. Δυστυχώς, αν και υπήρχαν αρκετές ιδέες κλιμάκωσης του έργου, η προσθήκη $5^{ης}$ μεθόδου ξέφευγε από τα πλαίσια της παρούσας εργασίας.

Τέλος, θα μπορούσε να τροποποιηθεί ελαφρώς η τέταρτη μέθοδος ώστε να εφαρμόζεται ο τροποποιημένος αλγόριθμος χαρτών αυτο–οργάνωσης ξεχωριστά για κάθε παράδειγμα εισόδου (και όχι ανά δέσμη εικόνων). Ο πειραματισμός αυτής της μεθόδου σε ένα γρήγορο υπολογιστικό σύστημα με πολλούς επιταχυντές υλικού εκτιμούμε ότι μπορεί να επιφέρει καλύτερα αποτελέσματα.

Βιβλιογραφία

[1] C. S. Smith, "A.I. Here, There, Everywhere," *The New York Times*, Feb. 2021.

[2] B. Marr, "The 10 Best Examples Of How AI Is Already Used In Our Everyday Life," Dec. 2019.

[3] S.-L. Wamba-Taguimdje, S. F. Wamba, J. R. K. Kamdjoug, and C. E. T. Wanko, "Influence of artificial intelligence (ai) on firm performance: the business value of ai-based transformation projects," *Business Process Management Journal*, 2020.

[4] M. Fernández, A. Bellogín, and I. Cantador, "Analysing the effect of recommendation algorithms on the amplification of misinformation," *arXiv preprint arXiv:2103.14748*, 2021.

[5] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Transactions on Management Information Systems (TMIS)*, vol. 6, no. 4, pp. 1–19, 2015.

[6] S. Makridakis, "The forthcoming artificial intelligence (ai) revolution: Its impact on society and firms," *Futures*, vol. 90, pp. 46–60, 2017.

[7] J. Hawksworth, R. Berriman, and S. Goel, "Will robots really steal our jobs? an international analysis of the potential long term impact of automation," *PricewaterhouseCoopers, http://pwc. co. uk/economics, access*, vol. 13, 2018.

[8] W. E. Forum, "The future of jobs report 2020," 2020.

[9] M. C.-T. Tai, "The impact of artificial intelligence on human society and bioethics," *Tzu-Chi Medical Journal*, vol. 32, no. 4, p. 339, 2020.

[10] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. Nelson, A. Bridgland, *et al.*, "Improved protein structure prediction using potentials from deep learning," *Nature*, vol. 577, no. 7792, pp. 706–710, 2020.

[11] T. Panch, H. Mattie, and L. A. Celi, "The "inconvenient truth" about ai in healthcare," *NPJ digital medicine*, vol. 2, no. 1, pp. 1–3, 2019.

[12] A. Rajkomar, J. Dean, and I. Kohane, "Machine learning in medicine," *New England Journal of Medicine*, vol. 380, no. 14, pp. 1347–1358, 2019.

[13] N. Bajema, "AI's 6 Worst-Case Scenarios," Jan. 2022.

[14] European Comission, "A European approach to artificial intelligence | Shaping Europe's digital future," 2021.

[15] U. von der Leyen, "President von der Leyen on the Commission's new strategy: Shaping Europe's Digital Future," 2019.

[16] A. Mayor, *Gods and robots: myths, machines, and ancient dreams of technology.* Princeton University Press, 2020.

[17] S. Pinker, "The cognitive niche: Coevolution of intelligence, sociality, and language," *Proceedings of the National Academy of Sciences*, vol. 107, no. supplement_2, pp. 8993–8999, 2010.

[18] J. Tooby and I. DeVore, "The reconstruction of hominid behavioral evolution through strategic modeling," *The evolution of human behavior: Primate models*, pp. 183–237, 1987.

[19] J. C. Avise and F. J. Ayala, *In the Light of Evolution: Volume IV: The Human Condition.* National Academies Press, 2010.

[20] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[21] D. Hebb, *The organization of behavior; a neuropsychological theory.* Wiley, 1949.

[22] J. Akst, "Machine, learning, 1951," May 2019.

[23] J. A. Lee, *Computer pioneers.* IEEE Computer Society Press, 1995.

[24] A. M. TURING, "I.—COMPUTING MACHINERY AND INTELLIGENCE," *Mind*, vol. LIX, pp. 433–460, 10 1950.

[25] A. M. Turing, "Intelligent machinery," 1948.

[26] H. Muehlenbein, "Artificial intelligence and neural networks the legacy of alan turing and john von neumann," *Int J Comput*, vol. 5, no. 3, pp. 10–20, 2014.

[27] S. Russell and P. Norvig, *Artificial intelligence: a modern approach.* Pearson, 2020.

[28] J. McCarthy, "Programs with common sense," in *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, RLE and MIT computation center Cambridge, MA, USA, 1960.

[29] J. R. Slagle, "A heuristic program that solves symbolic integration problems in freshman calculus," *J. ACM*, vol. 10, p. 507–520, oct 1963.

[30] J. Slaney and S. Thiébaux, "Blocks world revisited," *Artificial Intelligence*, vol. 125, no. 1-2, pp. 119–153, 2001.

[31] S. S. Haykin, *Neural networks and learning machines.* Upper Saddle River, NJ: Pearson Education, third ed., 2009.

[32] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* MIT press, 2016.

[34] A. Vinhas, "Adaline neural networks: The origin of gradient descent," Mar 2021.

[35] M. Mitchell, "Why ai is harder than we think," *arXiv preprint arXiv:2104.12871*, 2021.

[36] B. S. Todd, *An introduction to expert systems*. Oxford University Computing Laboratory, Programming Research Group, 1992.

[37] M. Z. Bell, "Why expert systems fail," *Journal of the Operational Research Society*, vol. 36, no. 7, pp. 613–619, 1985.

[38] D. E. Rumelhart, J. L. McClelland, P. R. Group, *et al.*, *Parallel distributed processing*, vol. 1. IEEE New York, 1988.

[39] R. Singh, "Rise and fall of symbolic ai," Sep 2019.

[40] E. N. Zalta, "The stanford encyclopedia of philosophy," 2019.

[41] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[42] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[44] N. Benaich and I. Hogarth, "State of ai report 2021," Oct 2021.

[45] U. Beck, *Risk society*. Sage Publications Ltd, 1992.

[46] G. Anadiotis, "Andrew ng predicts the next 10 years in ai," Mar 2022.

[47] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with em routing," in *International conference on learning representations*, 2018.

[48] A. D. Gritzman, "Avoiding implementation pitfalls of "matrix capsules with em routing" by hinton et al.," in *International Workshop on Human Brain and Artificial Intelligence*, pp. 224–234, Springer, 2019.

[49] V. Mazzia, F. Salvetti, and M. Chiaberge, "Efficient-capsnet: Capsule network with self-attention routing," *Scientific reports*, vol. 11, no. 1, pp. 1–13, 2021.

[50] T. M. Mitchell *et al.*, "Machine learning," 1997.

[51] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2019.

[52] A. Krizhevsky, "Learning multiple layers of features from tiny images," pp. 32–33, 2009.

[53] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, "The loss surfaces of multilayer networks," in *Artificial intelligence and statistics*, pp. 192–204, PMLR, 2015.

[54] A. Ng, "Forward and backward propagation (c1w4l06)."

[55] D. H. Hubel, "Single unit activity in striate cortex of unrestrained cats," *The Journal of physiology*, vol. 147, no. 2, p. 226, 1959.

[56] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of physiology*, vol. 148, no. 3, p. 574, 1959.

[57] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of physiology*, vol. 195, no. 1, pp. 215–243, 1968.

[58] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[59] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, pp. 818–833, Springer, 2014.

[60] D. Kollias, A. Tagaris, A. Stafylopatis, S. Kollias, and G. Tagaris, "Deep neural architectures for prediction in healthcare," *Complex & Intelligent Systems*, vol. 4, no. 2, pp. 119–131, 2018.

[61] I. Kollia, A.-G. Stafylopatis, and S. Kollias, "Predicting parkinson's disease using latent information extracted from deep neural networks," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2019.

[62] A. Arsenos, D. Kollias, and S. Kollias, "A large imaging database and novel deep neural architecture for covid-19 diagnosis," in *2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, pp. 1–5, IEEE, 2022.

[63] D. Kollias, M. Yu, A. Tagaris, G. Leontidis, A. Stafylopatis, and S. Kollias, "Adaptation and contextualization of deep neural network models," in *2017 IEEE symposium series on computational intelligence (SSCI)*, pp. 1–8, IEEE.

[64] D. Kollias, N. Bouas, Y. Vlaxos, V. Brillakis, M. Seferis, I. Kollia, L. Sukissian, J. Wingate, and S. Kollias, "Deep transparent prediction through latent representation analysis," *arXiv preprint arXiv:2009.07044*, 2020.

[65] D. Kollias, Y. Vlaxos, M. Seferis, I. Kollia, L. Sukissian, J. Wingate, and S. Kollias, "Transparent adaptation in deep medical image diagnosis," in *International Workshop on the Foundations of Trustworthy AI Integrating Learning, Optimization and Reasoning*, pp. 251–267, Springer, 2020.

[66] F. De Sousa Ribeiro, F. Calivá, M. Swainson, K. Gudmundsson, G. Leontidis, and S. Kollias, "Deep bayesian self-training," *Neural Computing and Applications*, vol. 32, no. 9, pp. 4275–4291, 2020.

[67] A. Psaroudakis and D. Kollias, "Mixaugment & mixup: Augmentation methods for facial expression recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2367–2375, 2022.

[68] D. Kollias and S. Zafeiriou, "Training deep neural networks with different datasets in-the-wild: The emotion recognition paradigm," in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2018.

[69] D. Kollias and S. Zafeiriou, "Va-stargan: Continuous affect generation," in *International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 227–238, Springer, 2020.

[70] G. Caridakis, A. Raouzaiou, K. Karpouzis, and S. Kollias, "Synthesizing gesture expressivity based on real sequences," in *Workshop Programme*, vol. 10, p. 19.

[71] B. Alhnaity, S. Kollias, G. Leontidis, S. Jiang, B. Schamp, and S. Pearson, "An autoencoder wavelet based deep neural network with attention mechanism for multi-step prediction of plant growth," *Information Sciences*, vol. 560, pp. 35–50, 2021.

[72] P. Mylonas, E. Spyrou, Y. Avrithis, and S. Kollias, "Using visual context and region semantics for high-level concept detection," *IEEE Transactions on Multimedia*, vol. 11, no. 2, pp. 229–243, 2009.

[73] S. Kollias and D. Anastassiou, "A unified neutral network approach to digital image halftoning," *IEEE Transactions on signal processing*, vol. 39, no. 4, pp. 980–984, 1991.

[74] P. Tzouveli, A. Schmidt, M. Schneider, A. Symvonis, and S. Kollias, "Adaptive reading assistance for the inclusion of students with dyslexia: The agent-dysl approach," in *2008 Eighth IEEE International Conference on Advanced Learning Technologies*, pp. 167–171, IEEE, 2008.

[75] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *International conference on artificial neural networks*, pp. 44–51, Springer, 2011.

[76] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *Advances in neural information processing systems*, vol. 30, 2017.

[77] G. Hinton, "How to represent part-whole hierarchies in a neural network," *arXiv preprint arXiv:2102.12627*, 2021.

[78] G. E. Hinton, "Capsule networks."

[79] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[80] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[81] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[82] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, pp. 213–229, Springer, 2020.

[83] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[84] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[85] M. I. Jordan, "Serial order: A parallel distributed processing approach," in *Advances in psychology*, vol. 121, pp. 471–495, Elsevier, 1997.

[86] A. Soleimany, "Mit 6.s191: Recurrent neural networks and transformers."

[87] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[88] L. Dirac, "Lstm is dead. long live transformers!."

[89] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[90] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[91] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[92] S. Finger, *Minds behind the brain: A history of the pioneers and their discoveries*. Oxford University Press, 2000.

[93] S. B. Eickhoff, R. T. Constable, and B. T. Yeo, "Topographic organization of the cerebral cortex and brain cartography," *Neuroimage*, vol. 170, pp. 332–347, 2018.

[94] E. I. Knudsen, S. du Lac, and S. D. Esterly, "Computational maps in the brain.," *Annual review of neuroscience*, vol. 10, pp. 41–65, 1987.

[95] R. Durbin and G. Mitchison, "A dimension reduction framework for understanding cortical maps," *Nature*, vol. 343, no. 6259, pp. 644–647, 1990.

[96] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2, pp. II–104, IEEE, 2004.

[97] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[98] F. De Sousa Ribeiro, G. Leontidis, and S. Kollias, "Introducing routing uncertainty in capsule networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6490–6502, 2020.

[99] F. De Sousa Ribeiro, G. Leontidis, and S. Kollias, "Capsule routing via variational bayes," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 3749–3756, Apr. 2020.

[100] J. E. Lenssen, M. Fey, and P. Libuschewski, "Group equivariant capsule networks," *Advances in neural information processing systems*, vol. 31, 2018.

[101] T. Cohen and M. Welling, "Group equivariant convolutional networks," in *International conference on machine learning*, pp. 2990–2999, PMLR, 2016.

[102] A. Jaiswal, W. AbdAlmageed, Y. Wu, and P. Natarajan, "Capsulegan: Generative adversarial capsule network," in *Proceedings of the European conference on computer vision (ECCV) workshops*, pp. 0–0, 2018.

[103] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic, "Generative adversarial metric," 2016.

[104] C. Xiang, L. Zhang, Y. Tang, W. Zou, and C. Xu, "Ms-capsnet: A novel multi-scale capsule network," *IEEE Signal Processing Letters*, vol. 25, no. 12, pp. 1850–1854, 2018.

[105] J.-w. Liu, F. Gao, R.-k. Lu, Y.-f. Lian, D.-z. Wang, X.-l. Luo, and C.-r. Wang, "Ddrm-capsnet: capsule network based on deep dynamic routing mechanism for complex data," in *International Conference on Artificial Neural Networks*, pp. 178–189, Springer, 2019.

[106] J. Rajasegaran, V. Jayasundara, S. Jayasekara, H. Jayasekara, S. Seneviratne, and R. Rodrigo, "Deepcaps: Going deeper with capsule networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10725–10733, 2019.

[107] J.-w. Liu, F. Gao, R.-k. Lu, Y.-f. Lian, D.-z. Wang, X.-l. Luo, and C.-r. Wang, "Fsc-capsnet: Fractionally-strided convolutional capsule network for complex data," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, IEEE, 2019.

[108] A. Hoogi, B. Wilcox, Y. Gupta, and D. L. Rubin, "Self-attention capsule networks for object classification," *arXiv preprint arXiv:1904.12483*, 2019.

[109] W. Huang and F. Zhou, "Da-capsnet: dual attention mechanism capsule network," *Scientific Reports*, vol. 10, no. 1, pp. 1–13, 2020.

[110] P. Shiri, R. Sharifi, and A. Baniasadi, "Quick-capsnet (qcn): a fast alternative to capsule networks," in *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1–7, IEEE, 2020.

[111] U. Manogaran, Y. P. Wong, and B. Y. Ng, "Capsnet vs cnn: analysis of the effects of varying feature spatial arrangement," in *Proceedings of SAI Intelligent Systems Conference*, pp. 1–9, Springer, 2020.

[112] L. Luo, S. Duan, and L. Wang, "R-capsnet: An improvement of capsule network for more complex data," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 2124–2129, IEEE, 2019.

[113] M. Luo, X. Wang, and H. Ma, "Capsnet based on encoder and decoder for object detection," in *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1112–1117, IEEE, 2020.

[114] E. Xi, S. Bing, and Y. Jin, "Capsule network performance on complex data," *arXiv preprint arXiv:1712.03480*, 2017.

[115] Z. Dong and S. Lin, "Research on image classification based on capsnet," in *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 1, pp. 1023–1026, IEEE, 2019.

[116] D. Wang and Q. Liu, "An optimization view on dynamic routing between capsules," 2018.

[117] S. D. Pande and M. S. R. Chetty, "Analysis of capsule network (capsnet) architectures and applications," *J Adv Res Dynam Control Syst*, vol. 10, no. 10, pp. 2765–2771, 2018.

[118] Z. Zhu, G. Peng, Y. Chen, and H. Gao, "A convolutional neural network based on a capsule network with strong generalization for bearing fault diagnosis," *Neurocomputing*, vol. 323, pp. 62–75, 2019.

[119] S. Srivastava, P. Khurana, and V. Tewari, "Identifying aggression and toxicity in comments using capsule network," in *Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018)*, pp. 98–105, 2018.

[120] A. Kosiorek, S. Sabour, Y. W. Teh, and G. E. Hinton, "Stacked capsule autoencoders," *Advances in neural information processing systems*, vol. 32, 2019.

[121] S. Sabour, A. Tagliasacchi, S. Yazdani, G. Hinton, and D. J. Fleet, "Unsupervised part representation by flow capsules," in *Proceedings of the 38th International Conference on Machine Learning* (M. Meila and T. Zhang, eds.), vol. 139 of *Proceedings of Machine Learning Research*, pp. 9213–9223, PMLR, 18–24 Jul 2021.

[122] W. Sun, A. Tagliasacchi, B. Deng, S. Sabour, S. Yazdani, G. E. Hinton, and K. M. Yi, "Canonical capsules: Self-supervised capsules in canonical pose," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24993–25005, 2021.

[123] M. Niemeyer and A. Geiger, "Giraffe: Representing scenes as compositional generative neural feature fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11453–11464, 2021.

[124] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European conference on computer vision*, pp. 405–421, Springer, 2020.

[125] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, pp. 213–229, Springer, 2020.

[126] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[127] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *ArXiv*, vol. abs/1708.07747, 2017.

[128] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research),"

[129] S. An, M. Lee, S. Park, H. Yang, and J. So, "An ensemble of simple convolutional neural network models for mnist digit recognition," *arXiv preprint arXiv:2008.10400*, 2020.

[130] M. S. Tanveer, M. U. K. Khan, and C.-M. Kyung, "Fine-tuning darts for image classification," in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 4789–4796, IEEE, 2021.

[131] F. A. Heinsen, "An algorithm for routing capsules in all domains," *arXiv preprint arXiv:1911.00792*, 2019.

[132] A. Byerly, T. Kalganova, and I. Dear, "A branching and merging convolutional network with homogeneous filter capsules," *arXiv preprint arXiv:2001.09136*, 2020.

[133] S. S. R. Phaye, A. Sikka, A. Dhall, and D. Bathula, "Dense and diverse capsule networks: Making the capsules learn better," *arXiv preprint arXiv:1805.04001*, 2018.

[134] S. Zhang, Q. Zhou, and X. Wu, "Fast dynamic routing based on weighted kernel density estimation," in *International symposium on artificial intelligence and robotics*, pp. 301–309, Springer, 2018.

[135] K. Ahmed and L. Torresani, "Star-caps: Capsule networks with straight-through attentive routing," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[136] A. Deliege, A. Cioppa, and M. Van Droogenbroeck, "Hitnet: a neural network with capsules embedded in a hit-or-miss layer, extended with hybrid data augmentation and ghost capsules," *arXiv preprint arXiv:1806.06519*, 2018.

[137] P. Nair, R. Doshi, and S. Keselj, "Pushing the limits of capsule networks," *arXiv preprint arXiv:2103.08074*, 2021.

[138] J. Gu and V. Tresp, "Improving the robustness of capsule networks to image affine transformations," 2019.

[139] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4. Springer, 2006.

[140] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[141] C. Sammut and G. I. Webb, *Encyclopedia of machine learning*. Springer Science & Business Media, 2011.

Παράρτημα Α΄

Ορισμοί Εννοιών

Το παρόν παράρτημα περιέχει ορισμούς εννοιών που εισάγονται κατά τη διάρκεια της παρούσας εργασίας. Κατά αυτόν τον τρόπο, δε διακόπτεται η ροή του κυρίως κειμένου. [27, 33, 51, 139]

Τεχνητή Νοημοσύνη

Έχουν υπάρξει πολλοί διαφορετικοί ορισμοί της Τεχνητής Νοημοσύνης: Μερικοί την περιγράφουν σαν εσωτερική διαδικασία της σκέψης που προσομοιάζει αυτής του ανθρώπου ενώ άλλοι ως εξωτερική διαδικασία μαθηματικά βέλτιστης συμπεριφοράς. Σύμφωνα με το κυρίαρχο μοντέλο, η Τεχνητή Νοημοσύνη ασχολείται κυρίως με τη λογική δράση. Ένας ιδανικός ευφυής πράκτορας δρα βέλτιστα σε κάθε περίσταση. Έτσι λοιπόν, η μελέτη της δημιουργίας ευφυών πρακτόρων μπορεί να τεθεί ως ορισμός της Τεχνητής Νοημοσύνης.

Μηχανική Μάθηση

Με λίγα λόγια, πρόκειται για τον κλάδο της τεχνητής νοημοσύνης ο οποίος ασχολείται με την ανάπτυξη υπολογιστικών συστημάτων ικανών να μαθαίνουν από παραδείγματα. Αναλυτικότερα, μπορούν και προσαρμόζονται χωρίς να ακολουθούν ρητές εντολές αλλά μέσω αλγορίθμων και στατιστικών μοντέλων που τους επιτρέπουν να αναλύουν και να εξάγουν συμπεράσματα από μοτίβα σε δεδομένα. Χαρακτηριστικό γνώρισμα των συστημάτων μηχανικής μάθησης είναι η ικανότητά τους να βελτιώνουν την απόδοσή τους σε μια εργασία (όπως αυτή μετράται με κάποια κατάλληλη μετρική) όσο η «εμπειρία» τους σε αυτήν αυξάνεται [50].

Τεχνητά Νευρωνικά Δίκτυα

Τα τεχνητά νευρωνικά δίκτυα αποτελούν ένα αλγοριθμικό κατασκεύασμα από απλούς υπολογιστικούς κόμβους διασυνδεδεμένους μεταξύ τους μέσω ακμών κάτω από μια συγκεκριμένη τοπολογία (συνήθως οργανώνονται σε επίπεδα, βλ. 2.1.3). Εμπνευσμένα από τα βιολογικά νευρωνικά δίκτυα, οι κόμβοι μπορούν να παρομοιαστούν με κύτταρα νευρώνων ενώ οι ακμές με νευρικές συνάψεις.

Τα τεχνητά νευρωνικά δίκτυα είναι παράδειγμα συστήματος μηχανικής μάθησης αφού μετά την κατάλληλη εκπαίδευσή τους, γενικεύουν από τα δεδομένα (inference). Τελικά, μετά την ανάπτυξή τους, υπό μια αφαιρετική σκοπιά αποτελεί το καθένα μια συνάρτηση που αντιστοιχίζει δεδομένα από τον χώρο εισόδου σε «προβλέψεις» του χώρου εξόδου.

Βαθιά Μάθηση

Αποτελεί μια υποκατηγορία μηχανικής μάθησης όπου χρησιμοποιούνται πολυεπίπεδα νευρωνικά δίκτυα. Τα πολλαπλά επίπεδα που διαθέτουν τους επιτρέπουν να μαθαίνουν και να αναγνωρίζουν εσωτερικά, γενικευμένα χαρακτηριστικά των δεδομένων εισόδου.

Υπολογιστική Νευροεπιστήμη (Computational Neuroscience)

Πρόκειται για τον κλάδο της Νευρωεπιστήμης που χρησιμοποιεί μαθηματικά μοντέλα, μαθηματική ανάλυση και προσεγγιστικά προς τον εγκέφαλο συστήματα για να κατανοήσει τις αρχές ανάπτυξης, δομής, φυσιολογίας καθώς και των γνωστικών (cognitive) ικανοτήτων του νευρικού συστήματος.

Επιβλεπόμενη Μάθηση

Στους αλγορίθμους επιβλεπόμενης μάθησης, ως είσοδος παρέχεται ένα σύνολο δεδομένων μαζί με τους επιθυμητούς στόχους. Δηλαδή, τα δεδομένα δίνονται σε ζεύγη (παράδειγμα εισόδου—επιθυμητή τιμή εξόδου). Με βάση αυτά, το σύστημα καλείται να εξάγει μια συνάρτηση η οποία θα έχει μάθει να μοντελοποιεί τη σχέση εισόδου–εξόδου μέσα από τα παραδείγματα και τελικά θα είναι ικανή να προβλέψει την τιμή εξόδου σε νέα παραδείγματα για τα οποία η τιμή στόχος είναι άγνωστη. Συνήθως, εκτός από τα δεδομένα για την εκπαίδευση υπάρχουν και άλλα σύνολα δεδομένων για τον έλεγχο της απόδοσης του συστήματος πρόβλεψης.

Ανάλογα με το αν η τιμή στόχος είναι διακριτή ή συνεχής, έχουμε αντίστοιχα το πρόβλημα ταξινόμησης (classification) ή της παλινδρόμησης (regression). Παράδειγμα συστήματος ταξινόμησης επιβλεπόμενης μάθησης είναι το φίλτρο ανεπιθύμητης αλληλογραφίας το οποίο αφού εκπαιδεύτηκε με ένα σύνολο επισημασμένων αλληλογραφιών ως ανεπιθύμητων ή επιθυμητών έμαθε να εντοπίζει νέα εισερχόμενη ανεπιθύμητη αλληλογραφία. Ένα παράδειγμα συστήματος παλινδρόμησης επιβλεπόμενης μάθησης είναι αυτό της πρόβλεψης τιμών μετοχών καθώς ο στόχος (κόστος μετοχής) είναι συνεχής αριθμός.

Μη-επιβλεπόμενη Μάθηση

Οι αλγόριθμοι μη–επιβλεπόμενης μάθησης, σε αντιδιαστολή με τους αλγορίθμους επιβλεπόμενης μάθησης, δέχονται ως είσοδο ένα σύνολο δεδομένων που περιλαμβάνει παραδείγματα, χωρίς όμως να συνοδεύονται από αντίστοιχες τιμές–στόχους. Στην περίπτωση αυτή, το υπό εκπαίδευση σύστημα επιχειρεί να μάθει πρότυπα στα δεδομένα εισόδου χωρίς κάποιο μηχανισμό ανατροφοδότησης. Συνήθεις εφαρμογές μη–επιβλεπόμενης μάθησης είναι αυτές της ομαδοποίησης των δεδομένων σε συστάδες ή της αναπαράστασής τους με ένα γράφημα.

Ενισχυτική Μάθηση

Στην ενισχυτική μάθηση, στο σύστημα (το οποίο καλείται «ευφυής πράκτορας» στο πλαίσιο αυτό) δεν παρέχεται κάποιο σύνολο δεδομένων αλλά η όποια εμπειρία αποκτάται μέσω της αλληλεπίδρασής του με το περιβάλλον. Ο πράκτορας έχει τη δυνατότητα να παρατηρήσει το περιβάλλον του και τη (πιθανή) κατάστασή του και ανάλογα με μια στρατηγική (policy) να δράσει σε αυτό. Το περιβάλλον του, με κάθε δράση (και ανάλογα την κατάσταση) παρέχει την απαραίτητη εμπειρία υπό τη μορφή επιβράβευσης (reward) ή ποινής (punishment). Έτσι, ο πράκτορας μαθαίνει από την εμπειρία προσαρμόζοντας τη στρατηγική του ώστε να μεγιστοποιεί την επιβράβευση την οποία λαμβάνει και τελικά να πετυχαίνει τον στόχο του. Παράδειγμα ενός τέτοιου πράκτορα είναι ένα σύστημα το οποίο παίζει σκάκι.

Μάθηση Κατά Δέσμες

Αφορά το είδος συστημάτων μηχανικής μάθησης που δεν έχουν τη δυνατότητα να μαθαίνουν σταδιακά αλλά εκπαιδεύονται μονομιάς χρησιμοποιώντας όλο το σύνολο δεδομένων στην

είσοδό τους. Σε περίπτωση που προστεθούν νέα δεδομένα στα οποία θα επιθυμούσαμε το σύστημα να προσαρμοστεί, απαιτείται εκ νέου εκπαίδευση στο καινούριο σύνολο δεδομένων το οποίο θα περιέχει τόσο τα παλαιά όσο και τα επιπρόσθετα δεδομένα (διαδικασία χρονοβόρα και υπολογιστικά κοστοβόρα). Συνήθως, σε τέτοιες περιπτώσεις το σύστημα πρέπει να σταματήσει να λειτουργεί και να μεταβεί στη φάση σχεδιασμού. Παραδείγματα αυτών των μεθόδων αποτελούν ο αλγόριθμος Expectation Maximization και ο Self-organizing map όπως περιγράφεται στην ενότητα 2.4.

Μάθηση σε Ζωντανό Χρόνο

Πρόκειται για τα συστήματα μηχανικής μάθησης που, σε αντίθεση με αυτά που μαθαίνουν κατά δέσμες, είναι ικανά να εκπαιδεύονται σταδιακά, είτε με ένα παράδειγμα τη φορά είτε με μικρές δέσμες παραδειγμάτων στην είσοδό τους. Το θετικό σε αυτά τα συστήματα είναι η δυνατότητα προσαρμογής τους σε νέα δεδομένα με πολύ μικρό χρονικό και υπολογιστικό κόστος. Αποτέλεσμα αυτού είναι ότι υπάρχει (συνήθως) η δυνατότητα η εκπαίδευσή τους να γίνει ζωντανά (online) χωρίς να σταματήσει η λειτουργία του συστήματος. Παράδειγμα αποτελούν οι εφαρμογές πρόβλεψης τιμών μετοχών όπου απαιτείται συνεχής προσαρμογή του συστήματος στα νέα δεδομένα της αγοράς.

Μάθηση Βασισμένη σε Παραδείγματα

Είναι μια οικογένεια απλών συστημάτων μηχανικής μάθησης που αφορά τον τρόπο με τον οποίο ένα σύστημα γενικεύει από τα παραδείγματα του συνόλου εισόδου. Στα συγκεκριμένα, όταν τα τροφοδοτούμε με κάποιο νέο παράδειγμα, το συγκρίνουν με τα δεδομένα εισόδου (ή ένα υποσύνολο αυτών) τα οποία έχουν αποθηκευθεί στη μνήμη τους κατά την εκπαίδευση. Ένα χαρακτηριστικό μειονέκτημα αυτών των συστημάτων είναι ότι ο χώρος που απαιτείται για την αποθήκευση του μοντέλου (του συστήματος μάθησης μετά την εκπαίδευσή του) αυξάνεται με το μέγεθος του συνόλου εισόδου (συνήθως με γραμμικό τρόπο). Ενδεικτικά, ένα σύστημα που γενικεύει κατά αυτόν τον τρόπο είναι το K-nearest neighbors.

Μάθηση Βασισμένη σε μοντέλο

Είναι μια άλλη οικογένεια συστημάτων όπου η μηχανική μάθηση γίνεται μέσω της προσαρμογής (fitting) ενός μοντέλου στα δεδομένα εισόδου. Έχοντας εκφράσει το σύνολο των δεδομένων εκπαίδευσης (ή τη σχέση αυτών με την επιθυμητή έξοδο) χρησιμοποιώντας ένα κατάλληλα εκφραστικό (expressive) μοντέλο, λέμε ότι το σύστημα μαθαίνει να «γενικεύει» από τα παραδείγματα. Έτσι, για να παράξει προβλέψεις σε νέα δεδομένα, δεν απαιτείται η αποθήκευση όλων των δεδομένων εκπαίδευσης αλλά μόνο των παράμετρων του μοντέλου που εκφράζει.

Γνωστική Νευροεπιστήμη

Η Γνωστική νευρωεπιστήμη είναι το πεδίο μελέτης που ασχολείται με τα νευρωνικά υποστρώματα των διανοητικών διεργασιών. Είναι η τομή της ψυχολογίας με τη νευροεπιστήμη. Συνδυάζει τις θεωρίες της γνωστικής ψυχολογίας και της υπολογιστικής μοντελοποίησης με πειραματικά δεδομένα του εγκεφάλου.

Αναγνώριση Προτύπων

Είναι ένα επιστημονικό πεδίο με στόχο την ανάπτυξη αλγορίθμων για την αυτοματοποιημένη απόδοση κάποιας τιμής (παλινδρόμησης) ή διακριτικού στοιχείου (ταξινόμηση) με

βάση μοτίβα/χαρακτηριστικά που παρατηρούνται στα εισαγόμενα δεδομένα, συνήθως κωδικοποιημένα ως αλληλουχίες αριθμών.

Γραμμικά Διαχωρίσιμες Κλάσεις

Λέμε ότι ένα σύνολο δεδομένων για ταξινόμηση που περιέχει δύο κλάσεις είναι γραμμικά διαχωρίσιμο αν και μόνο αν μπορούμε να διαχωρίσουμε τις δύο κλάσεις στον πολυδιάστατο χώρο χαρακτηριστικών εισόδου χρησιμοποιώντας ένα υπερεπίπεδο. Στην περίπτωση όπου ο χώρος χαρακτηριστικών είναι δισδιάστατος, αρκεί να μπορούμε να χαράξουμε μια ευθεία γραμμή στο καρτεσιανό επίπεδο που να διαχωρίζει τις δύο κλάσεις.

Γραμμικά Μοντέλα

Τα γραμμικά μοντέλα περιγράφουν τη σχέση μεταξύ ενός ή περισσοτέρων μεταβλητών εισόδου (μεταβλητές πρόβλεψης) και μιας συνεχούς τιμής εξόδου (απόκρισης). Η χρήση των μοντέλων αυτών ενδείκνυται όταν οι σχέσεις μεταξύ εισόδου–εξόδου είναι (σχεδόν) γραμμικές στο διάστημα μελέτης. Μια στατιστική μέθοδος για την παραγωγή γραμμικών μοντέλων που μοντελοποιούν αυτές τις σχέσεις από σύνολα δεδομένων εισόδου–εξόδου είναι η γραμμική παλινδρόμηση.

Γενετικοί Αλγόριθμοι

Οι Γενετικοί αλγόριθμοι ανήκουν στο κλάδο της επιστήμης υπολογιστών και αποτελούν μια μέθοδο αναζήτησης βέλτιστων λύσεων σε προβλήματα βελτιστοποίησης. Είναι χρήσιμοι σε περιπτώσεις όπου ο χώρος αναζήτησης λύσης είναι πολύ μεγάλος και δεν υπάρχει αναλυτική μέθοδος που να μπορεί να βρει το βέλτιστο συνδυασμό τιμών των μεταβλητών του προβλήματος ώστε το υπό εξέταση σύστημα να αντιδρά με βέλτιστο τρόπο. Ο τρόπος λειτουργίας των Γενετικών Αλγορίθμων είναι εμπνευσμένος από τη βιολογία. Χρησιμοποιεί δηλαδή την ιδέα της εξέλιξης μέσω γενετικής μετάλλαξης, φυσικής επιλογής και διασταύρωσης. Για να αξιοποιήσουμε αυτές τις ιδέες, κωδικοποιούμε κάθε πιθανή λύση του προβλήματος σαν ένα συγκεκριμένο γονιδίωμα και ξεκινάμε από έναν τυχαίο πληθυσμό τέτοιων λύσεων/γονιδιωμάτων. Έπειτα, ορίζοντας μια συνάρτηση ικανότητας (fittness function) που περιγράφει την ποιότητα της λύσης είμαστε σε θέση να αφήσουμε τον μηχανισμό εξέλιξης να δράσει για ορισμένες γενιές ώστε τελικά να έχουν απομείνει και πολλαπλασιαστεί γονιδιώματα που περιγράφουν (σχεδόν) βέλτιστες λύσεις. Οι γενετικοί αλγόριθμοι δεν εγγυώνται την εύρεση της βέλτιστης λύσης.

Νευρωνικά Δίκτυα με Κάψουλες (Capsule Networks)

Πρόκειται για βαθιά νευρωνικά δίκτυα που επιδιώκουν να πραγματοποιήσουν ανάστροφα γραφικά για να λύσουν κυρίως προβλήματα αναγνώρισης αντικειμένων σε εικόνες. Αποτελούνται από επίπεδα από κάψουλες. Κάθε κάψουλα είναι σαν μια συνάρτηση η οποία προσπαθεί να προβλέψει τις παραμέτρους στιγμιοτύπου (π.χ. προσανατολισμός, θέση κ.τ.λ.) ενός συγκεκριμένου αντικειμένου και την πιθανότητα ύπαρξής του σε μια περιοχή της εικόνας (δηλαδή στο πεδίο υποδοχής της κάψουλας).

Γραφικά Υπολογιστή

Αφορά τον κλάδο της επιστήμης υπολογιστών που μελετά μεθόδους για ψηφιακή σύνθεση και χειρισμό οπτικού περιεχομένου. Εμπεριέχει μια δόση τέχνης αφού σχετίζεται με τον σχεδιασμό του περιεχομένου αυτού.

Απόδοση Εικόνας (Rendering) Είναι η διεργασία δημιουργίας εικόνας από ένα μοντέλο δύο ή τριών διαστάσεων με τη χρήση ενός προγράμματος υπολογιστή. Πολλά μοντέλα ορίζονται σε ένα αρχείο σκηνής (scene file) το οποίο περιγράφει την πληροφορία της οπτικής σκηνής που θα παραχθεί με την απόδοση εικόνας. Συνήθως, το αρχείο περιέχει πληροφορία για τη γεωμετρία, την οπτική γωνία, την υφή, τον φωτισμό και τη σκίαση των αντικειμένων.

Ανάστροφα Γραφικά

Πρόκειται για την ανάστροφη διαδικασία της απόδοσης εικόνας. Δηλαδή, δοθείσης μιας οπτικής εικόνας, να προσδιοριστεί το αρχείο σκηνής από το οποίο δημιουργήθηκε.

Ακολουθιακά Δεδομένα (Sequential Data)

Ο όρος αφορά δεδομένα των οποίων τα επιμέρους στοιχεία διατάσσονται σε μια συγκεκριμένη σειρά. Για παράδειγμα, οι λέξεις στον φυσικό λόγο αποτελούν ακολουθιακά δεδομένα. Άλλα παραδείγματα είναι οι ακολουθίες ΔΝΑ και η τιμή μιας μετοχής στο χρηματιστήριο, όπως αυτή μεταβλαλλεται στον χρόνο.

Κανονικοποίηση Επιπέδου (Layer Normalization)

Πρόκειται για μια τεχνική που χρησιμοποιείται για την κανονικοποίηση της κατανομής των τιμών ενεργοποίησης σε κάθε παράδειγμα εισόδου ξεχωριστά [140]. Η τεχνική αυτή μειώνει σημαντικά τον χρόνο εκπαίδευσης (οι συναρτήσεις ενεργοποίησης λειτουργούν στη γραμμική περιοχή τους, γύρω από το μηδέν).

Ανταγωνιστική Μάθηση (Competitive Learning)

Αφορά τη διαδικασία μη–επιβλεπόμενης μάθησης κατά την οποία διαφορετικοί νευρώνες (ή γενικότερα, υπολογιστικές μονάδες) ανταγωνίζονται για το ποιος θα αναλάβει να «εξηγήσει» και να μάθει να αναπαριστά την εκάστοτε είσοδο x_i (ενός συνόλου εδομένων). Από την στιγμή που όλοι οι νευρώνες, καθώς το δίκτυο τροφοδοτείται με παραδείγματα, μαθαίνουν να αναπαριστούν καλύτερα τις εισόδους που είναι ήδη καλοί στο να αναπαριστούν, εξειδικεύονται στο να εξηγούν συγκεκριμένα μοτίβα εισόδων ο καθένας. Μια από τις πιο απλές μορφές της ανταγωνιστικής μάθησης είναι η λεγόμενη «ο νικητής τα παίρνει όλα» (winner takes it all), όπως παρουσιάζεται στην ενότητα 2.4 [141].

Αυτοκωδικοποιητής (Autoencoder) Πρόκειται για μια αρχιτεκτονική τεχνητών νευρωνικών δικτύων που χρησιμοποιείται για την εκμάθηση αποδοτικών (διανυσματικών) αναπαραστάσεων από μη–σεσημασμένα δεδομένα. Σε αυτό, δίνεται σαν είσοδος ένα παράδειγμα (διάνυσμα χαρακτηριστικών) και απαιτούμε να λάβουμε στην έξοδο το ίδιο διάνυσμα (μέσω μιας συνάρτησης σφάλματος η οποία συγκρίνει την έξοδο με την είσοδο). Με λίγα λόγια, εκπαιδεύουμε το δίκτυο ώστε να έχει συμπεριφορά ταυτοτικής συνάρτησης. Ο περιορισμός όμως είναι ότι ανάμεσα στο επίπεδο εισόδου και το επίπεδο εξόδου επιβάλουμε (μέσω αρχιτεκτονικής) μια στένωση (βοττλενεςκ) με το να χρησιμοποιούμε λιγότερους κόμβους κρυφών επιπέδων από τους κόμβους εισόδου και εξόδου. Έτσι, το δίκτυο μαθαίνει συμπικνωμένες αναπαραστάσεις των δεδομένων. Λέμε ότι χωρίζεται σε δύο τμήματα: τον κωδικοποιητή και τον αποκωδικοποιητη. Το πρώτο τμήμα βρίσκεται πριν τη στένωση και ρόλο έχει να συμπηκνώσει την πληροφορία εισόδου σε ένα μικρότερο διάνυσμα χαρακτηριστικών. Το τμήμα του αποκωδικοποιητή έχει τον αντίστροφο ρόλο, της αποσυμπίεσης σε ένα πιστό αντίγραφο του διανύσματος εισόδου.

Παράρτημα Β΄

Απόδοση Ξενόγλωσσων Όρων

Ξενόγλωσσος όρος	Ελληνική απόδοση
batch learning	μάθηση κατά δέσμες
online learning	μάθηση σε ζωντανό χρόνο
supervised learning	επιβλεπόμενη μάθηση
unpervised learning	μη-επιβλεπόμενη μάθηση
reinforcement learning	ενισχυτική μάθηση
capsule networks	νευρωνικά δίκτυα με κάψουλες
instance based	βασισμένο σε παραδείγματα
perturbation test	πείραμα διαταραχής
learning rate	ρυθμός μάθησης
optimizer	βελτιστοποιητής
multihead attention	πολυκέφαλη προσοχή
equivariant	ισομεταβλητό
invariant	ανεξάρτητο
recurrent neural network	επαναλαμβανόμενο νευρωνικό δίκτυο
reconstructor	ανακατασκευαστής
inverse graphics	ανάστροφα γραφικά
primary capsules	κάψουλες πρώτου επιπέδου
digit capsules	κάψουλες τελευταίου επιπέδου
softmax	συνάρτηση ομαλής μεγιστοποίησης
squashing function	συνάρτηση σύνθλιψης
single parent assumption	υπόθεση μοναδικού πατέρα
similarity score	σκορ ομοιότητας
coupling coefficients	βάρη δρομολόγησης

Παράρτημα Γ΄

Αναλυτική Περιγραφή Αλγορίθμου Δρομολόγησης με SOM

Στο παρόν παράρτημα παρατίθεται η αναλυτική, σχηματική περιγραφή του σύνθετου αλγορίθμου SOM-Based Routing. Ο αλγόριθμος συνοδεύεται από σχηματικές απεικονίσεις των πινάκων αλλά και περιγραφές των αλγοριθμικών εντολών.

Τον αλγόριθμο επίσης συνοδεύουν σημειώσεις σχετικά με τα σημεία στα οποία ο αλγόριθμός μας διαφέρει από τον αυθεντικό αλγόριθμο SOM.

Τα χρώματα του μελανιού που χρησιμοποιούνται για την κάθε σημείωση έχουν ξεχωριστή σημασία, ανάλογα με το είδος της σημείωσης. Χρησιμοποιούμε:

- Μαύρο χρώμα για τις αλγοριθμικές εντολές.
- Γκρι χρώμα για τις σημειώσεις πάνω στις αλγοριθμικές εντολές όπως για την περιγραφή του σχήματος των τανυστών (tensors).
- Με μοβ χρώμα καταγράφονται οι διαφορές μεταξύ του αλγορίθμου SOM και του αλγορίθμου που αναπτύξαμε για τη δρομολόγηση των καψουλών που βασίζεται στον SOM.
- Το γαλάζιο χρώμα το χρησιμοποιούμε για τη σχηματική αναπαράσταση των τανυστών.
- Με πράσινο χρώμα δηλώνονται οι αρχικοποιήσεις.

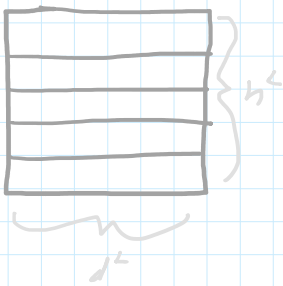
Έχοντάς περιγράψει τον χρωματικό κώδικα, είμαστε πλέον σε θέση να παραθέσουμε την αναλυτική περιγραφή του αλγορίθμου.

Let's begin from Primary Capsules

Primary Caps Shape: $[B, n^L, d^L]$, where $n^L = w^L \times h^L \times F^L$

For the schematic representations, let's assume that $n^L = 5$, $h^{L+1} = 2$, $m = 3$ (optional)

Under those assumptions, C_b^L:



$S_{simsOut_b}^L$:

Input: PrimaryCaps C^L

Output: Similarities $S_{simsOut}^L$

$S_{simsOut}^L.shape: [B, h^{L+1}]$

Trainable Parameters: W (transformation Matrices)

Non-Trainable Parameters: C^{L+1}

$C^{L+1}.shape: [h^{L+1}, d^{L+1}]$

Hyperparameters: reduced_votes, m, normalize_votes, norm_type,

- **reduced_votes**: If True, then each capsule C_i^L produces n^{L+1} votes. If False, each capsule produces m votes, where m is not necessarily equal to n^{L+1}.
- **m**: Transformation Matrices for each capsule. This is used only if reduced_votes ==False. So, it specifies the number of projections that Each capsule C_j^{L+1} has.
- **normalize_votes**: If we set normalize_votes hyperparameter to True then we transform the vote vectors so as every one of them has length no more than one.
- **norm_type**: If we set it to 0 then we apply squashing function as a normalization technique. If we set it to 1 we apply tanh normalization to votes and unit normalization to output capsules.

r, radical, Θ, take_into_account_win_ratio,

- **r**: Number of routing iterations. In each iteration, Capsules C_j^{L+1} are updated once.
- **radical**: If radical is set to True then instead of computing the differences as the votes minus the digit caps, we set the differences to be same as the votes. Radical approach was inspired by the perceptron algorithm (update rule) and differs significantly from the SOM's update rule.
- **Θ**: Specifies the neighborhood function. It is a list of float numbers that determine the gravity of the update for the second, third place. Always, the winner should get the full update so theta at least should contain 1.0 . It is worth mentioning that unlike SOM, our lateral distance is the relative to the winner parent capsule distance.
- **take_into_account_win_ratio**: If True then the updates for each digit capsule are scaled according to the number of votes they "won" in a batch.

take_into_account_similarity, softmax, tanh_like, normalize_in_loop,

- **take_into_account_similarity**: If it is set to True then the updates are scaled according to the similarity. Also, if set, the individual thetas in the theta list are not taken into consideration. Just the length of the list to define how many inner loop iterations.
- **softmax**: If set to True then instead of hard winners we get soft winners. So even capsules that do not have any winner may get updated with a small coefficient. Of course then there is no point in using a neighborhood.
- **tanh_like**: Similar to softmax. Just the operation is scaled so that dissimilar parent capsules move further away when updated.
- **normalize_in_loop**: Used to normalize digit capsules in every iteration of the outer loop. This, if many iterations are used improves stability.

lr_SOM

- **lr_SOM**: Learning rate used in SOM-based updates.

Start of SOM-Based Routing

At first we compute the votes using the transformation matrices. Depending on the value of the reduced_votes hyperparameter, we use different transformation matrices and get different votes.

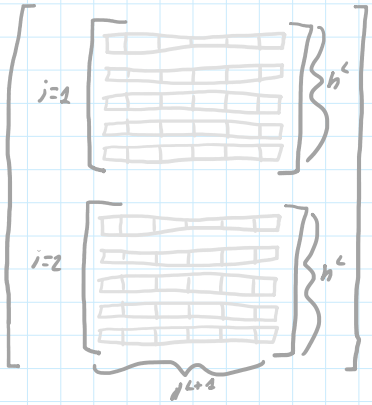
If reduced_votes == True:

W^L.shape: $[n^L, d^L, d^{L+1}, n^{L+1}]$

- $\forall b \in [1, B], \forall i \in \Omega_L, \forall j \in \Omega_{L+1}: V^L_{bij:} \leftarrow C^L_{bi:} \times W^L_{i:::j}$ $\quad [1 \times d^L] \quad [d^L \times d^{L+1}]$

V^L.shape: $[B, n^L, n^{L+1}, d^{L+1}]$

V_b^L:



else:

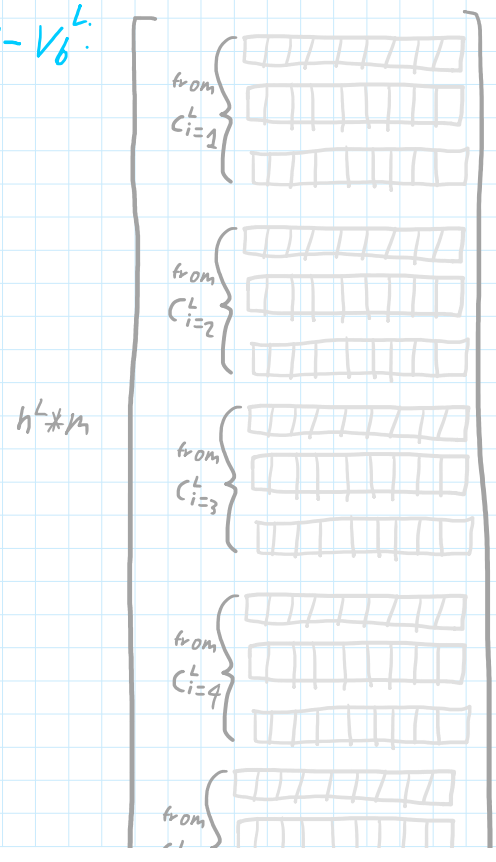
W^L.shape $[n^L, d^L, d^{L+1}, m^L]$

- $\forall b \in [1, B], \forall i \in \Omega_L, \forall j_m \in [1, m]:$ $pre\text{-}V^L_{b(i+n^L*(j_m-1))d^{L+1}} \leftarrow C^L_{bi:} \times W^L_{i:::j_m}$ $\quad [1 \times d^L] \quad [d^L \times d^{L+1}]$

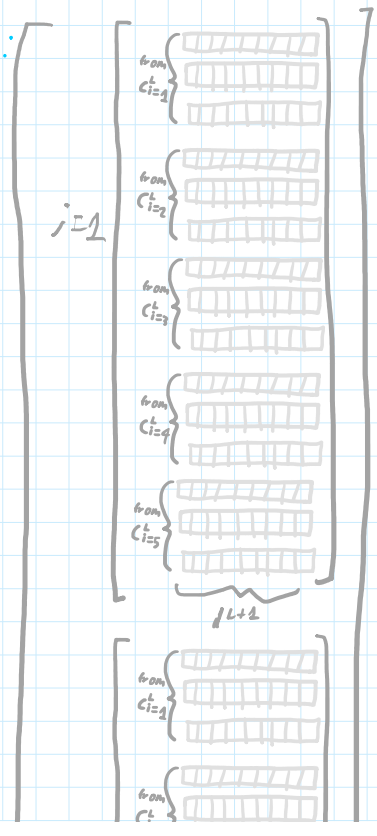
Actually, we compute m projections for each one of the n^L primary capsules and then we concatenate (we get them all together so we lose track of which vote comes from which primary capsule). We will still use the same index "i" but this, no longer means that this particular vote V_{ij}^L comes from capsule C_i^L.

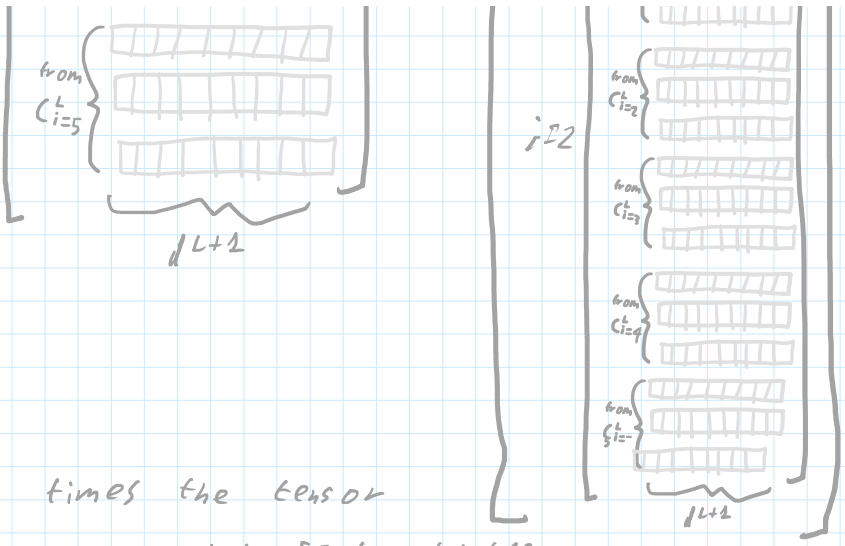
Now $i \in [1, n^L * m]$ so $pre\text{-}V^L_{bi:}$ comes from $C^L_{bi/m}$.

$pre\text{-}V_b^L$:



V_b^L:





Copy n^{L+1} times the tensor pre-V^L so as to get V.shape: $[B, n^L * m, n^{L+1}, d^{L+1}]$

- $\forall j \in \Omega_{L+1}: V^L_{:,:,j,:} \leftarrow$ pre-V^L

In other words, stack n^{L+1} copies of pre-V^L vertically.

So, V.shape: $[B, \underbrace{n^L * m}_{\underline{n}^L}, n^{L+1}, d^{L+1}]$

$$\underline{n}^L = \begin{cases} n^L & \text{iff reduced\_votes == True} \\ n^L * m & \text{iff reduced\_votes == False} \end{cases}$$

$$\underline{\Omega}_L = \begin{cases} \Omega_L & \text{iff reduced\_votes == True} \\ \text{set of all votes in tensor pre-}V^L_b, & \text{iff reduced\_votes == False} \end{cases}$$

So... in either case:

V.shape: $[B, \underline{n}^L, n^{L+1}, d^{L+1}]$

if normalize_votes:

Make vote vectors to not exceed length of 1 (L2 norm).

if norm_type == 0:

- $\forall b \in [1, B], \forall i \in \underline{\Omega}_L, \forall j \in \Omega_{L+1}: V^L_{b,i,j} \leftarrow Squash(V^L_{b,i,j})$

$$\frac{\|V^L_{b,i,j}\|^2}{2+\|V^L_{b,i,j}\|^2} \cdot \frac{V^L_{b,i,j}}{\|V^L_{b,i,j}\|}$$

else:

- $\forall b \in [1, B], \forall i \in \underline{\Omega}_L, \forall j \in \Omega_{L+1}: V^L_{b,i,j} \leftarrow \underbrace{\tanh(\|V^L_{b,i,j}\|)}_{\text{scaling term}} \left(\frac{V^L_{b,i,j}}{\|V^L_{b,i,j}\|}\right)$ ← unit norm

V^L has same shape as before.

If take_into_account_win_ratio:

Initialize WinCount with zeros, WinCount.Shape: $[n^{L+1}]$

for r iterations do:

Initialize SV with zeros, SV.shape: $[B, n^L, n^{L+1}, d^{L+1}]$
↳ sparce updates

Time to compute the differences between the V^L and C^{L+1}.
These differences "D" will be used in the update rule to update the parent capsules C^{L+1}.

If radical:

- $D^L \leftarrow V^L$

Radical updates is inspired by Perceptron rule.
This is significantly different to the updates computed in the original SOM.

original SOM update rule:
$C_j^{L+1}(s+1) \leftarrow C_j^{L+1}(s) + \alpha(s) * \Theta(c_s)_{j,BMU} * (V_{ij}^L - C_j^{L+1}(s))$
(D^L)
↓ learning rate, ↓ neighborhood function

Else:

- $\forall b \in [1, B],\ \forall i \in \Omega_L:\ D^L_{bi::} \leftarrow V^L_{bi} - C^L$
  $[n^{L+1}, d^{L+1}]$ $[n^{L+1}, d^{L+1}]$

Even if reduced_votes == False, if radical is not True, D^L tensor has different entries for different j's (unlike V^L which contains n^{L+1} copies of pre-V^L iff reduced_votes == False). That is because pre-V^L votes (in V^L) are subtracted by different parent capsules.

It is clear that the above update rule can not be parallelizable. Finding the "winner" (Best Matching Unit - BMU) parent capsule for each datapoint - vote V^L_{bij} and then updating C^{L+1} in a serial manner would have been very slow.

We need a new update rule that can be parallel, across parent capsules and across batch items.

D^L.Shape: $[B, n^L, n^{L+1}, d^{L+1}]$

Initialize mask M^L with ones, M^L.Shape: $[B, n^L, n^{L+1}, d^{L+2}]$

for each θ in Θ:

Our metric for similarity is the inner product. By this criterion we pick the BMUs (winners).

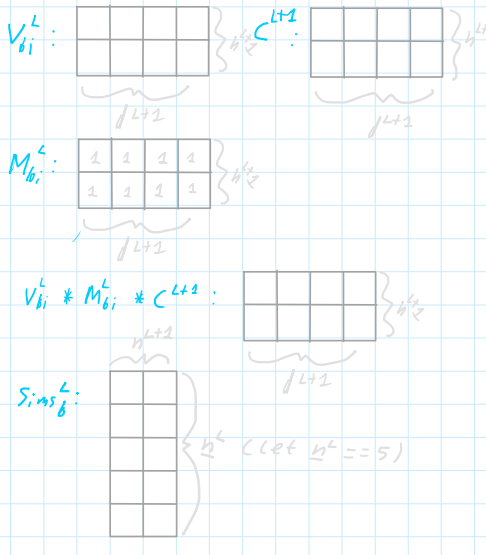
This is another difference to the original SOM, the criterion used there is the Euclidian Distance.

- $\forall b \in [1,B], \forall i \in \Omega_L:\ Sims^L_{bi} \leftarrow \sum_k^{d^{L+1}} \left[\left(V^L_{bi} * M^L_{bi} \right) * C^{L+1} \right]_{bik}$
  $[n^{L+1}, d^{L+1}]$ $[n^{L+1}, d^{L+2}]$

$Sims^L$.Shape: $[B, n^L, n^{L+1}]$

$Sims^L$ contains all the similarities between the parent capsules C^{L+1} and their corresponding pose-predictions (votes V^L), for each batch.

Take note that iff reduced_votes == False, then all the votes (pre-V^L) are compared to all the parent capsules. So no vote is tied to a particular parent capsule. We leave it to the SOM-based routing to make the descrimination. In other words, iff reduced_votes == False, C^{L+1} have same "view" of data.



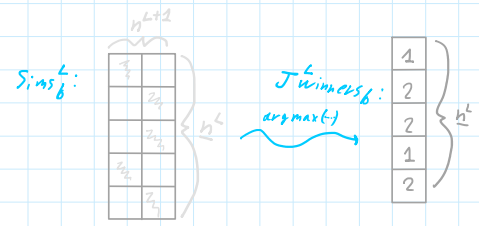
If not softmax nor tanh_like

We proceed by finding the Best Matching Units (winners). Competition is, as it should be, between the C^{L+1} as they are trying to explain the datapoints

In our SOM-analogy, the "datapoints" are in our case the votes while the "nodes" are the parent capsules C^{L+1}.

Matching Units (winners). Competition is, as it should be, between the C^{L+1} as they are trying to explain the datapoints (votes). The parent capsule that explains best a child capsule C^L_{bi} (i.e. has the largest similarity with the corresponding vote) wins.

} the votes while the "nodes" are the parent capsules C^{L+1}.



- $\forall b \in [1, B], \forall i \in \Omega_L : J^L_{winners_{bi}} \leftarrow \underset{j}{argmax}(Sims^L_{bi:})$

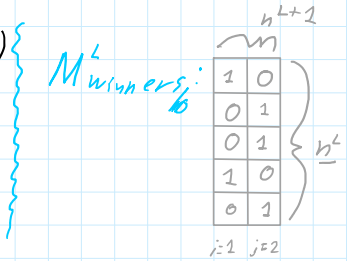
  $J^L_{winners}.shape : [B, h^L]$

  $J^L_{winners_b}$ contains one winner index per child capsule.

- $\forall b \in [1, B] : M^L_{winners_b} \leftarrow OneHot(J^L_{winners_b}, h^{L+1})$

  $M^L_{winners}$ is a mask that has 1's on the positions of sims where the maximum value (across parent capsules) is located.

  $M^L_{winners}.Shape : [B, h^L, h^{L+1}]$



else if softmax:

we produce "soft" winners.

- $\forall b \in [1, B] : M^L_{winners_b} \leftarrow \underset{across\ j}{Softmax}(Sims^L_b)$

  Where $M^L_{winners}.Shape : [B, h^L, h^{L+1}]$

  and $\underset{across\ j}{Softmax}(Sims^L_b) = \begin{bmatrix} softmax(Sims^L_{bi=1}) \\ softmax(Sims^L_{bi=2}) \\ \vdots \\ softmax(Sims^L_{bi=h^L}) \end{bmatrix}$

else if tanh_like

With tanh_like dissimilar parent capsules will point further away.

- $\forall b \in [1, B] : M^L_{winners_b} \leftarrow \left(\underset{across\ j}{Softmax}(Sims^L_b) - 0.5 \right) * 2$

  Where "−" and "*" are pointwise operations.

  Again, $M^L_{winners}.Shape : [B, h^L, h^{L+1}]$

If not take_into_account_similarity and not take_into_account_win_ratio:

Original, simple case: Compute the updates of the winning nodes (capsules). For the first iteration of thetas, we will update the winners. On the next iterations we will update the winners' 1st degree neighbours. On the third iteration of thetas (if, of course the hyperparameter Θ contains 3 items) we will update the 2nd degree neighbours of the winning nodes.

- $\forall b \in [1, B], \forall i \in \Omega_L : Usparse^L_{bi} \leftarrow \left(M^L_{winners_{bi}} \times P^L_{bi} \right) * lr\_SOM * \theta$

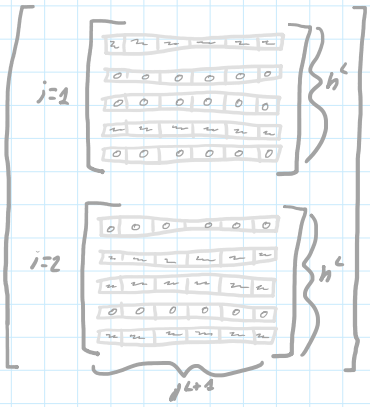
* $\forall\ b \in [1,B],\ \forall\ i \in \Omega_L$: $U^L_{sparse_{bi}} \leftarrow \left(M^L_{winners_{bi}} \times P^L_{bi} \right) * lr\_SOM * \theta$
  $[h^{L+1}]$ $[h^{L+1}, d^{L+1}]$ — masked differences; * pointwise

U^L_{sparse} contains the updates of selected capsules. If on the first iteration, then U^L_{sparse} contains the updates of the winner capsules.

$U^L_{sparse}.shape : [B, n^L, h^{L+1}, d^{L+1}]$

U^L_{sparse} :

similar shapes for P^L_{bi} and $M^L_{winners bi}$.



As you can see (I suppose), each parent capsule C^{L+1}_j will be updated according to the capsules C^L_{bi} that it attracted.

Child capsules that a parent capsule C^{L+1}_j did not won, will not contribute to it's update (thus, their row will be zero).

That is the case for the 1st iteration. On the next iteration, the capsule C^{L+1}_j that won children capsule C^L_{bi} can not "claim" that capsule again. On the second iteration, the winner capsule for datapoint C^L_{bi} will now be the parent capsule $C^{L+1}_{j'}$ that has the second best similarity.

One significant difference between our SOM and the original is this: Instead of iteratively finding the one parent capsule that best explains a datapoint (a vote V^L_{bi}) and updating that parent capsule with that vote, we find many winners (BMU's) and update them in parallel.

The above note is another major difference to the original SOM algorithm. Lateral distance, in our case, is defined as the rank distance on the similarity "leaderboard".

else if take_into_account_similarity:

We will perform similar operations as before. The difference now is that we will scale the Mask of the winners according to the similarities. In that way, the gravity of the update will be proportional to the agreement.

* $SimsSparse^L \leftarrow Sims^L * M^L_{winners}$
  $[B, n^L, n^{L+1}]$ * pointwise $[B, n^L, n^{L+1}]$
  $SimsSparse^L.shape : [B, n^L, n^{L+1}]$

* $b \in [1,B],\ \forall\ i \in \Omega_L$: $U^L_{sparse_{bi}} \leftarrow \left(SimsSparse^L_{bi} \times P^L_{bi} \right) * lr\_SOM$
  $[h^{L+1}]$ $[h^{L+1}, d^{L+1}]$ — scaled, masked differences; * pointwise

$U^L_{sparse}.shape : [B, n^L, h^{L+1}, d^{L+1}]$

No need to scale more using θ. Similarity-scaling dose this for us. (as we go from iteration to iteration, similarities will decrease)

If take_into_account_win_ratio :

This is another update-scaling term that will be used later.

* $WinCountPartial^L \leftarrow \sum_i^{n^L} \sum_b^B M^L_{winners_{bi}}$
  $[n^{L+1}]$

Contains the number of wins that each

* $WinCountPartial^L \leftarrow \sum_i^{h^L} \sum_b^B M^L_{winners\,bi}$ $[h^{L+1}]$

$WinCountPartial.Shape: [h^{L+1}]$

Contains the number of wins that each parent capsule made in one iteration of θ.

* $WinCount \leftarrow WinCount + WinCountPartial$

↳ contains the number of C_i^L's that each C_j^{L+1} "won" (in all iterations).

If not softmax and not tanh_like:

Update mask M^L so that in the next iteration of θ (neighboors with distance 1), we will not have the same winners. In other words, we want to find the neighboors of the winner capsules, not the winners themselves.

* $\forall b \in [1, B], \forall i \in \Omega_L, \forall j \in \Omega_{L+1}: M^L_{bij:} \leftarrow M^L_{bij:} - M^L_{winners\,bij}$ pointwise $[1]$

$M.Shape: [B, h^L, h^{L+1}, d^{L+1}]$ $[B, h^L, h^{L+1}, d^{L+1}]$

M has now zeros on the places where $M^L_{winners}$ had ones.

Aggregate the updates across thetas.

* $SU^L \leftarrow SU^L + U^L_{sparse}$

↳ total updates ↳ partial updates

$SU^L.shape: [B, h^L, h^{L+1}, d^{L+1}]$

End of neighbour loop.

Make updates dense by finding, for each parent capsule C_j^{L+1} the mean of all the votes from which capsules he won (across batch size). This is done to make less updates but more "educated".

* $U^L \leftarrow \frac{\sum_b^B \sum_i^{h^L} SU^L_{bi}}{B * h^L}$ $U^L.shape: [h^{L+1}, d^{L+1}]$

Time to change the DigitCaps (perform update Step).

If not take_into_account_win_ratio:

If not radical:

Basic, "simple" case:

* $C^{L+1} \leftarrow C^{L+1} + U^L$

$C^{L+1}.Shape: [h^{L+1}, d^{L+1}]$

As you can see, C^{L+1} is not dependent on a single input. Rather, it changes given a batch at training. So, it is not a prediction. The capsule vectors C^{L+1} are learned to represent the invariant properties of the image features. With these insp[illegible]

else if radical:

C . Shape: $[h^{L+1}, d^{L+1}]$

The capsule vectors C^{L+1} are learned to represent the invariant properties of the image features. With these instantiation invariant vectors the votes are compared and the similarities are produced. As we will see in a moment, these aggregated similarities are also the output of our model. Please note that during testing, preferably $r == 0$ so as to not update C^{L+1} anymore.

else if radical:

momentum

Average over r iterations.

* $C^{L+1} \leftarrow \frac{C^{L+1} * (r-1) + U^L}{r}$

else:

scale updates relative to the number of wins.

* $WinRatio^L \leftarrow \frac{WinCount^L}{n^L * B}$ $\quad WinRatio.Shape: [h^{L+1}]$

* $SoftWinRatio^L \leftarrow Softmax(WinRatio^L)$ $\quad$ SoftWinRatio is like assignment probabilities.

Shape $[h^{L+1}]$

* $\forall j \in \Omega_{L+1}$: $U_j^L \leftarrow SoftWinRatio_j^L * U_j^L$

Shape $[h^{L+1}, d^{L+1}]$

If not radical:

Basic, "simple" case:

* $C^{L+1} \leftarrow C^{L+1} + U^L$

C^{L+1}. Shape: $[h^{L+1}, d^{L+1}]$

else if radical:

momentum

Average over r iterations.

* $C^{L+1} \leftarrow \frac{C^{L+1} * (r-1) + U^L}{r}$

If normalize_l_in_loop:

Normalize digit capsules before next iteration.

if norm_type == 0:

* $\forall j \in \Omega_{L+1}$: $C_j^{L+1} \leftarrow Squash(C_j^{L+1})$

else:

* $\forall j \in \Omega_{L+1}$: $C_j^{L+1} \leftarrow \underbrace{\tanh(\|C_j^{L+1}\|)}_{\text{scaling term}} \underbrace{\frac{C_j^{L+1}}{\|C_j^{L+1}\|}}_{\text{unit norm}}$

Repeat if $r > 1$

End of r iterations

Compute the similarities now, using the updated C^{L+1}.

* $\forall b \in [1, B]$, $\forall i \in \Omega_L$: $FinalSims_{b,i}^{L+1} \leftarrow \sum_{k}^{I^{L+1}} \left(V_{b,i}^{L} * C^{L+1} \right)_{b,i,k}$

  shape: $[B, n^L, n^{L+1}]$; $[B, n^L, n^{L+1}, pl^{L+1}]$

* $\forall b \in [1, B]$: $SimsOut_b^{L+1} \leftarrow \frac{\sum_{i}^{n^L} FinalSims_{b,i}^{L+1}}{n^L}$

  shape: $[B, n^{L+1}]$

Find mean similarity across child capsules.

You can then argmax them and get the prediction for each instance in batch.

In test mode, these two lines below may be the only steps of the algorithm after computing the votes.

* Return $SimsOut^{L+1}$, C^{L+1}

For a behaviour that closely resembles the one of capsules, we can set batch size = 1 and use a bigger lr_som. In this way, capsules C^{L+1} will capture the equivariant properties of the data.