



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

**Ανάπτυξη Αρχιτεκτονικών Βαθιάς Μάθησης με Χρήση
Δικτύων Μακράς Βραχυπρόθεσμης Μνήμης Για Ανάλυση
Συναισθήματος**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Θεόδωρος Ι. Φλωράκης

Επιβλέπων: Θεοδώρα Βαρβαρίγου

Καθηγήτρια Ε.Μ.Π.

Αθήνα, Νοέμβριος 2018



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

**Ανάπτυξη Αρχιτεκτονικών Βαθιάς Μάθησης με Χρήση
Δικτύων Μακράς Βραχυπρόθεσμης Μνήμης Για Ανάλυση
Συναισθήματος**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Θεόδωρος Ι. Φλωράκης

Επιβλέπων: Θεοδώρα Βαρβαρίγου

Καθηγήτρια Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 9η Νοεμβρίου 2018

.....
Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

.....
Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π.

.....
Συμεών Παπαβασιλείου
Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2018

.....
Θεόδωρος Ι. Φλωράκης

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

Copyright © Θεόδωρος Φλωράκης 2018.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Ευχαριστίες

Ολοκληρώνοντας την διπλωματική μου εργασία και τον κύκλο των προπτυχιακών σπουδών μου θα ήθελα να ευχαριστήσω όσους συνέβαλαν στην προσπάθειά μου.

Αρχικά θα ήθελα να ευχαριστήσω την επιβλέπουσα της εργασίας μου, καθηγήτρια της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου, κα. Θεοδώρα Βαρβαρίγου, για την ανάθεση ενός τόσο ενδιαφέροντος θέματος, αλλά και για την πολύτιμη βοήθεια και καθοδήγηση που μου προσέφερε κατά την εκπόνησή του.

Ιδιαίτερες ευχαριστίες θέλω να εκφράσω προς τον υποψήφιο διδάκτωρ Βρεττό Μουλό, για την άριστη συνεργασία μας, αλλά και για τις γνώσεις και συμβουλές που μου προσέφερε σε κάθε στάδιο της εργασίας, πάντα με προθυμία και ενθουσιασμό.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου για την αμέριστη συμπαράσταση και υποστήριξή τους καθ' όλη τη διάρκεια των σπουδών μου.



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη αρχιτεκτονικών βαθιάς μάθησης με χρήση δικτύων μακράς βραχυπρόθεσμης μνήμης για ανάλυση συναισθήματος

Θεόδωρος Ι. Φλωράκης

Επιβλέπουσα: Καθηγήτρια Θεοδώρα Βαρβαρίγου
Συνεπιβλέπων: Υποψήφιος Διδάκτωρ Βρεττός Μουλός
Νοέμβριος 2018

ΠΕΡΙΛΗΨΗ

Οι απόψεις των ανθρώπων συνιστούν μια από τις πιο ισχυρές μορφές πληροφορίας που μπορεί κάποιος να αναζητήσει. Διαδραματίζουν ρόλο εξέχουσας σημασίας σε ενέργειες τόσο απλές όσο η καθημερινή λήψη αποφάσεων, όσο και τόσο σύνθετες όπως επίσημες έρευνες παγκόσμιας κλίμακας. Μαζί με την ραγδαία εξέλιξη του Διαδικτύου, δημιουργήθηκαν και όλες οι απαραίτητες προϋποθέσεις προκειμένου οι απόψεις των ανθρώπων να εκφραστούν μαζικά. Πλέον, στην εποχή των Μεγάλων Δεδομένων (Big Data) έχει δημιουργηθεί ένας τεράστιος όγκος δεδομένων από κείμενα, αναρτήσεις, σχολιασμούς και κριτικές που καλύπτουν οποιοδήποτε θέμα στο οποίο μπορεί να εκφραστεί μια άποψη. Ωστόσο, στην πλειοψηφία τους, τα δεδομένα αυτά παραμένουν αποθηκευμένα και ανεπεξέργαστα και ως εκ τούτου δεν είναι δυνατόν να εξορυχθεί η γνώση η οποία εμφωλεύεται σε αυτά.

Η Ανάλυση Συναισθήματος (Sentiment Analysis), γνωστή και ως Εξόρυξη Άποψης, είναι η διαδικασία επεξεργασίας της φυσικής γλώσσας η οποία αποσκοπεί στην υπολογιστική αναγνώριση και κατηγοριοποίηση των απόψεων που εκφράζονται σε ένα κομμάτι κειμένου, προκειμένου να καθοριστεί εάν η στάση του συγγραφέα έναντι ενός συγκεκριμένου θέματος, προϊόντος κ.λπ. είναι θετική, αρνητική ή ουδέτερη. Ως ερευνητικός τομέας, η Ανάλυση Συναισθήματος έχει γνωρίσει τεράστια άνθιση η οποία συμπίπτει με την ανάπτυξη της επιστήμης των υπολογιστών αλλά και με τις ολοένα και αυξανόμενες ανάγκες για την μαζική επεξεργασία δεδομένων απόψεων. Πλέον οι τεχνικές της επιστρατεύονται από επιχειρήσεις και οργανισμούς, όπου βρίσκουν τεράστιο πλήθος εφαρμογών, προσφέροντας σημαντικές λύσεις στην οργάνωση της πληροφορίας και την εξαγωγή γνώσεων ζωτικής σημασίας

Στόχος της παρούσας Διπλωματικής εργασίας είναι η ανάπτυξη συστημάτων μηχανικής μάθησης, τα οποία να είναι ικανά να διεξάγουν ανάλυση συναισθήματος σε μια συλλογή από κριτικές ταινιών, με τελικό σκοπό την συναισθηματικής τους ταξινόμηση ως θετικό ή αρνητικό. Συγκεκριμένα τα συστήματα που θα αναπτυχθούν και θα υλοποιηθούν, θα αποτελούν μοντέλα βαθιάς μάθησης. Θα αναπτυχθούν τρεις διαφορετικές αρχιτεκτονικές βαθιάς μάθησης με αναδρομικά νευρωνικά δίκτυα (LSTM) και στο τέλος θα διεξαχθεί συγκριτική μελέτη για την εξέταση της απόδοσης τους στην ταξινόμηση των εν λόγω κειμένων.

Λέξεις Κλειδιά: Ανάλυση Συναισθήματος, Μηχανική Μάθηση, Βαθιά Μάθηση, Big Data, Αναδρομικά Νευρωνικά Δίκτυα, LSTM, Επεξεργασία Φυσικής Γλώσσας



NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
DIVISION OF COMMUNICATIONS, ELECTRONICS AND INFORMATION SYSTEMS

DIPLOMA THESIS

Deep Learning Architectures with LSTM Networks for Sentiment Classification

Theodoros Florakis

Supervisor: Professor Theodora Varvarigou,
Co-Supervisor: PhD Candidate Vrettos Moulos,
November 2018

ABSTRACT

People's opinion is one of the most powerful forms of information that one can look for. They play a paramount role in actions as simple as day-to-day decision-making, and as complex as formal world-wide research surveys. Together with the rapid evolution of the Internet, all the necessary prerequisites were created for the people's opinion to be massively expressed. Now, in the Big Data era, a huge volume of texts, posts, annotations and criticisms have been created that cover any subject in which an opinion can be expressed. However, for the most part, these data remain stored and untreated and therefore it is not possible to extract the knowledge that is contained therein.

Sentiment Analysis, also known as Opinion Mining, is the natural language process that aims to computationally identify and categorize the opinions expressed in a piece of text in order to determine whether the author's attitude towards a particular subject, product, etc. is positive, negative or neutral. As a research subject, Sentiment Analysis has experienced a huge bloom that coincides with the development of computer science and with the ever-increasing need for mass data processing. Nowadays, its techniques are employed by businesses and organizations, where they are exploited in a great number of different applications, offering essential solutions in organizing the information and extracting vital knowledge

The aim of this diploma thesis is to develop machine learning systems that are capable of conducting Sentiment Analysis in a collection of movie reviews, with the ultimate goal of emotionally categorizing them as positive or negative. In particular, the systems that will be developed and implemented will be models based on deep learning. In this thesis, will be developed three different deep learning recurrent neural networks (LSTM) and will be conducted a comparative study in order to examine their performance regarding the classification of these texts.

Keywords: Sentiment Analysis, Machine Learning, Deep Learning, Big Data, Recurrent Neural Networks, LSTM, Natural Language Processing

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. ΕΙΣΑΓΩΓΗ	1
1.1 ΑΝΑΛΥΣΗ ΣΥΝΑΙΣΘΗΜΑΤΟΣ.....	1
1.1.1 Ορισμός.....	1
1.1.2 Εφαρμογές.....	2
1.1.3 Τεχνικές Ταξινόμησης Συναισθήματος.....	3
1.2 ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ.....	6
1.3 ΔΟΜΗΣ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ.....	6
2. ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ.....	9
2.1 ΓΕΝΙΚΑ	9
2.2 ΘΕΜΕΛΙΩΣΗ	10
2.3 ΕΙΔΗ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ.....	11
2.3.1 Επιβλεπόμενη Μάθηση.....	11
2.3.2 Μη Επιβλεπόμενη Μάθηση.....	12
2.3.3 Ενισχυτική Μάθηση.....	12
2.3.4 Βαθιά Μάθηση.....	12
3. ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	15
3.1 ΒΙΟΛΟΓΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ.....	15
3.2 ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	16
3.2.1 Μοντέλο Τεχνητού Νευρώνα.....	17
3.2.2 Δομές Τεχνητών Νευρωνικών Δικτύων.....	18
3.2.3 Ανάκληση και Εκπαίδευση.....	19
3.3 ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΠΡΟΣΘΙΑΣ ΤΡΟΦΟΔΟΤΗΣΗΣ.....	20
3.3.1 Αισθητήρας Perceptron και Εκπαίδευση.....	20
3.3.2 Δίκτυο perceptron ως γραμμικός διαχωριστής.....	21
3.3.3 Πολυεπίπεδα Δίκτυα Perceptron (Multi-Layer Perceptron).....	22
3.3.4 Ανάκληση ΤΝΔ Πρόσθιας Τροφοδότησης.....	23
3.3.5 Εκπαίδευση Πολυεπίπεδου Δικτύου Πρόσθιας Τροφοδότησης με τον αλγόριθμο Ανάστροφης Μετάδοσης Σφάλματος (Back-Propagation).....	25
3.3.6 ΤΝΔ Πρόσθιας Τροφοδότησης Πολλών Επιπέδων ως καθολικοί προσεγγιστές.....	29
3.4 ΑΝΑΔΡΟΜΙΚΑ ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ.....	31
3.4.1 Αρχιτεκτονικές Αναδρομικών Δικτύων.....	31
3.4.2 Ανάκληση Αναδρομικού ΤΝΔ.....	35
3.4.3 Εκπαίδευση Αναδρομικού Δικτύου με τον αλγόριθμο οπισθοδιάδοσης σφάλματος μέσα στον χρόνο (Back-Propagation Through Time).....	37
3.4.4 Αμφίδρομα αναδρομικά δίκτυα.....	39
3.5 ΜΕΘΟΔΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ.....	41
3.5.1 Παραλλαγές της μεθόδου κατάβασης δυναμικού.....	41
3.5.2 Ορμή (Momentum).....	43

3.5.3 Ο αλγόριθμος βελτιστοποίησης Adagrad	44
3.5.4 Ο αλγόριθμος βελτιστοποίησης Adadelta.....	45
3.5.5 Ο αλγόριθμος βελτιστοποίησης RMSprop.....	47
4. ΔΙΚΤΥΑ ΜΑΚΡΑΣ ΒΡΑΧΥΠΡΟΘΕΣΜΗΣ ΜΝΗΜΗΣ	49
4.1 ΤΟ ΠΡΟΒΛΗΜΑ ΤΩΝ ΜΑΚΡΙΝΩΝ ΕΞΑΡΤΗΣΕΩΝ ΣΤΑ ΑΠΛΑ ΑΝΑΔΡΟΜΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	49
4.2 ΤΑ ΔΙΚΤΥΑ ΜΑΚΡΑΣ ΒΡΑΧΥΠΡΟΘΕΣΜΗΣ ΜΝΗΜΗΣ	51
4.2.1 Γενικά	51
4.2.2 Το κύτταρο μακράς βραχυπρόθεσμης μνήμης.....	52
4.2.3 Λειτουργία κυττάρου μακράς βραχυπρόθεσμης μνήμης.....	53
4.2.4 Αρχιτεκτονικές και Εκπαίδευση Δικτύων Μακράς Βραχυπρόθεσμης Μνήμης	54
5. ΤΟ ΕΡΓΑΛΕΙΟ TENSORFLOW	57
5.1 ΕΙΣΑΓΩΓΗ.....	57
5.2 ΤΟ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΜΟΝΤΕΛΟ ΤΟΥ TENSORFLOW	58
5.2.1 Γράφοι Ροής Δεδομένων (Data Flow Graphs)	58
5.2.2 Τα πλεονεκτήματα των Γράφων Ροής Δεδομένων	60
5.3 ΤΑΝΥΣΤΕΣ, ΓΡΑΦΟΙ, ΣΥΝΕΔΡΙΕΣ.....	61
5.3.1 Τανυστές (Tensors).....	62
5.3.2 Σταθερές, Μεταβλητές και Placeholder	62
5.3.3 Δημιουργία Γράφου.....	63
5.3.4 Δημιουργία και εκτέλεση Συνεδρίας.....	64
6. ΠΡΟΣΕΓΓΙΣΗ ΠΡΟΒΛΗΜΑΤΟΣ.....	67
6.1 ΑΝΑΔΡΟΜΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΚΑΙ ΑΝΑΛΥΣΗ ΣΥΝΑΙΣΘΗΜΑΤΟΣ.....	67
6.2 ΑΝΑΠΑΡΑΣΤΑΣΗ ΛΕΞΕΩΝ ΣΤΟΝ ΔΙΑΝΥΣΜΑΤΙΚΟ ΧΩΡΟ.....	68
6.2.1 GloVe (Global Vectors for Word Representation)	69
6.3 ΕΠΙΛΕΓΜΕΝΕΣ ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΒΑΘΙΑΣ ΜΑΘΗΣΗΣ ΜΕ ΑΝΑΔΡΟΜΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	72
6.3.1 Επίπεδα Αρχιτεκτονικής	72
6.3.2 Αρχιτεκτονικές.....	75
6.4 ΕΚΠΑΙΔΕΥΣΗ	79
6.4.1 Επιλεγμένες Μέθοδοι Γενίκευσης	79
6.4.2 Βελτιστοποίηση.....	80
7. ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ	83
7.1 ΤΟ ΣΥΝΟΛΟ ΔΕΔΟΜΕΝΩΝ	83
7.2 ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΚΑΙ ΠΕΙΡΑΜΑΤΑ.....	83
7.2.1 Διανύσματα λέξεων.....	83
7.2.3 Πειράματα	85
7.3 ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΟΝΤΕΛΩΝ.....	86
7.4 ΑΞΙΟΛΟΓΗΣΗ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ.....	87

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Εικόνα 1 Διάγραμμα λειτουργίας μοντέλου επιβλεπόμενης μηχανικής μάθησης για ταξινόμηση συναισθήματος.....	5
Εικόνα 2 Διάγραμμα ροής μοντέλου επιβλεπόμενης μάθησης.....	11
Εικόνα 3 Αναπαράσταση βιολογικού νευρώνα	15
Εικόνα 4 Μαθηματικό μοντέλο τεχνητού νευρώνα	17
Εικόνα 5 Οι συναρτήσεις ενεργοποίησης: Βηματική, Σιγμοειδής, ReLU, Υπερβολική Εφαπτομένη.....	19
Εικόνα 6 Ταξινόμηση γραμμικά διαχωρίσιμων κλάσεων	21
Εικόνα 7 Παράδειγμα μη γραμμικά διαχωρίσιμων κλάσεων.....	22
Εικόνα 8 Παράδειγμα πολυεπίπεδου τεχνητού νευρωνικού δικτύου.....	23
Εικόνα 9 Ενεργοποίηση του νευρώνα i στο επίπεδο K του δικτύου	24
Εικόνα 10 Αναπαράσταση εύρεσης ελαχίστου με την μέθοδο κατάβασης δυναμικού.....	25
Εικόνα 11 Απεικόνιση διαχωριστικών επιφανειών που δημιουργούν πολυεπίπεδα δίκτυα perceptron με διαφορετικό πλήθος νευρώνων και κρυφών επιπέδων	30
Εικόνα 12 Πλήρης αναπαράσταση συνδέσεων αναδρομικού δικτύου.....	32
Εικόνα 13 Απλοποιημένη αναπαράσταση αναδρομικού δικτύου και unfolding του δικτύου.....	33
Εικόνα 14 Πολυεπίπεδο αναδρομικό δίκτυο	34
Εικόνα 15 Αναδρομικό δίκτυο με μία έξοδο στο τελευταίο επίπεδο.....	35
Εικόνα 16 Ανάστροφη μετάδοση σφάλματος σε αναδρομικό δίκτυο	37
Εικόνα 17 Αμφίδρομο αναδρομικό δίκτυο.....	40
Εικόνα 18 Σύγκριση σύγκλισης μεθόδων (α) στοχαστικής κατάβασης δυναμικού και (β) mini-batch κατάβασης δυναμικού.....	43
Εικόνα 19 Σύγκριση σύγκλισης μεθόδων (αριστερά) απλής στοχαστικής κατάβασης δυναμικού και (δεξιά) στοχαστικής κατάβασης δυναμικού με ορμή	44
Εικόνα 20 Σύγκριση σύγκλισης αλγορίθμων, SGD, Ορμής, Adagrad, Adadelta	46
Εικόνα 21 Το πρόβλημα μείωσης της κλίσης για τα απλά αναδρομικά δίκτυα	50
Εικόνα 22 Κρυφό επίπεδο απλού αναδρομικού δικτύου	51
Εικόνα 23 Δομή κρυφού επιπέδου κυττάρου LSTM	52
Εικόνα 24 Απεικόνιση της διατήρησης των μακρινών εξαρτήσεων σε δίκτυο LSTM.....	55
Εικόνα 25 Λογότυπο TensorFlow	57
Εικόνα 26 Αναπαράσταση νευρωνικού επιπέδου ως γράφος στο Tensorflow	59
Εικόνα 27 Αφηρημένη απεικόνιση του γράφου, πριν και μετά την εκτέλεση της συνεδρίας.	61
Εικόνα 28 Γραφική απεικόνιση τανυστών με μία έως πέντε διαστάσεις.....	62
Εικόνα 29 Παράδειγμα γράφου	64
Εικόνα 30 Παράδειγμα αναπαράστασης λέξεων ως διανύσματα	68
Εικόνα 31 Αποτελέσματα αναπαράστασης της μεθόδου GloVe.....	71
Εικόνα 32 Max-pooling φίλτρου 2x2 με 2 strides	74
Εικόνα 33 Mean-pooling φίλτρου 2x2 με 2 strides	75
Εικόνα 34 Απεικόνιση Αρχιτεκτονικής I.....	76
Εικόνα 35 Απεικόνιση Αρχιτεκτονικής II.....	77

Εικόνα 36 Απεικόνιση Αρχιτεκτονικής ΙΙΙ	78
Εικόνα 37 Αποτελέσματα στην ταξινόμηση συναισθήματος μοντέλων Αρχιτεκτονικής Ι	86
Εικόνα 38 Αποτελέσματα στην ταξινόμηση συναισθήματος μοντέλων Αρχιτεκτονικής ΙΙ.....	86
Εικόνα 39 Αποτελέσματα στην ταξινόμηση συναισθήματος μοντέλων Αρχιτεκτονικής ΙΙΙ	87

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Ανάλυση Συναισθήματος

Τα τελευταία χρόνια με την έκρηξη του διαδικτύου και κυρίως των μέσων κοινωνικής δικτύωσης, έχει παραχθεί και έχει συγκεντρωθεί ένας τεράστιος όγκος δεδομένων ο οποίος στο μεγαλύτερο μέρος του περιλαμβάνει δεδομένα κειμένου, όπως μηνύματα ηλεκτρονικού ταχυδρομείου, συζητήσεις, αναρτήσεις σε κοινωνικά μέσα, έρευνες, άρθρα και έγγραφα. Αυτή η πληθώρα δεδομένων, παρά την χρήσιμη πληροφορία και την γνώση που μπορεί να περιέχουν, δεν οργανώνονται με κάποιο ουσιαστικό τρόπο και επομένως παραμένουν αδόμητα.

Η ανάλυση συναισθήματος, ως ερευνητικός τομέας, έχει προσφέρει συστήματα μέσω των οποίων, αυτήν η τεράστια ποσότητα αδόμητης πληροφορίας έχει δομηθεί και έχει μεταφραστεί μαζικά σε δημόσιες απόψεις σχετικά με προϊόντα, υπηρεσίες, πολιτική ή οποιοδήποτε άλλο θέμα για το οποίο μπορεί να εκφραστεί μια γνώμη, παράγοντας έτσι χρήσιμη γνώση τόσο για τον χώρο της επιστήμης όσο και τον χώρο των επιχειρήσεων και της βιομηχανίας.

1.1.1 Ορισμός

Η Ανάλυση Συναισθήματος (Sentiment Analysis), η οποία συχνά αναφέρεται και ως Εξόρυξη Άποψης (Opinion Mining), αποτελεί ένα από τα σημαντικότερα και πιο δημοφιλή γνωστικά πεδία της Επεξεργασίας Φυσικής Γλώσσας (Natural Language Processing). Κύριος σκοπός της είναι η ανάπτυξη και δημιουργία τεχνικών και συστημάτων, τα οποία να είναι ικανά να ανιχνεύουν, να εξάγουν και να ποσοτικοποιούν υποκειμενικές πληροφορίες όπως απόψεις και συναισθήματα μέσα από την φυσική γλώσσα είτε αυτά βρίσκονται σε γραπτή είτε σε ακουστική μορφή.

Η γραπτή πληροφορία, με βάση την αντικειμενικότητα της, κατά κύριο λόγο μπορεί να κατηγοριοποιηθεί σε δύο τύπους: στις *απόψεις* και στα *γεγονότα*. Οι απόψεις αποτελούν

υποκειμενικές εκφράσεις που περιγράφουν τις σκέψεις, τις εκτιμήσεις και τα συναισθήματα ενός ανθρώπου. Αντίθετα, τα γεγονότα αποτελούν αντικειμενικές και μόνο, εκφράσεις και περιγραφές της πραγματικότητας.

Με βάση αυτόν τον διαχωρισμό, η ανάλυση συναισθήματος, όπως και πολλά άλλα προβλήματα επεξεργασίας φυσικής γλώσσας, μπορεί να διαμορφωθεί ως πρόβλημα ταξινόμησης όπου πρέπει να επιλυθούν δύο υπό-προβλήματα:

- Ταξινόμηση μιας φράσης ως υποκειμενική ή αντικειμενική, γνωστή ως ταξινόμηση υποκειμενικότητας.
- Ταξινόμηση μιας φράσης ως έκφραση μιας θετικής, αρνητικής ή ουδέτερης άποψης, γνωστή και ως ταξινόμηση πολικότητας ή ταξινόμηση συναισθήματος. Το κομμάτι της ταξινόμησης συναισθήματος θα αποτελέσει και το βασικό ερευνητικό πεδίο με το οποίο θα ασχοληθεί η παρούσα διπλωματική.

1.1.2 Εφαρμογές

Η ανάλυση συναισθήματος, τα τελευταία χρόνια επιστρατεύεται όλο και περισσότερο από επιχειρήσεις. Χάριν στις δυνατότητές της, η επεξεργασία δεδομένων αυτοματοποιείται αποδοτικά και αποτελεσματικά σε μεγάλη κλίμακα, επιταχύνοντας τις επιχειρηματικές διαδικασίες καθώς εξοικονομούνται πολλές ώρες χειροκίνητης επεξεργασίας δεδομένων και παρέχοντας ταυτόχρονα χρήσιμες και εξεζητημένες πληροφορίες. Πιο συγκεκριμένα, μερικές από τις εφαρμογές της ανάλυσης συναισθήματος είναι:

- Η ανάπτυξη συστημάτων συστάσεων (recommendation systems) όπου με βάση τα σχόλια που έχει λάβει ένα προϊόν (αρνητικά ή θετικά), να συμπεραίνεται η γενική αξιολόγηση του και κατ' επέκταση αυτό το προϊόν να προτείνεται ή όχι αντίστοιχα σε άλλους πελάτες.
- Σε product analytics όπου μέσω της ανάλυσης συναισθήματος μπορεί να επεξεργαστεί και να αναλυθεί μια μεγάλη ποσότητα από συμπληρωμένα ερωτηματολόγια, σχόλια σε social media κτλ. προκειμένου να βγει ένα συμπέρασμα για τις προτιμήσεις των πελατών και κατ' επέκταση για την γενική τάση της κάθε αγοράς. Έτσι μια επιχείρηση μπορεί να αντλήσει χρήσιμες πληροφορίες για την δημιουργία του επόμενου προϊόντος της αλλά και σε ποιο κομμάτι της αγοράς να το διαθέσει.
- Στην υποστήριξη πελατών, όπου εργαλεία ανάλυσης συναισθήματος μπορούν να εντοπίσουν τους πιο δυσαρεστημένους πελάτες προκειμένου να τους δοθεί μεγαλύτερη προτεραιότητα εξυπηρέτησης.

- Στα χρηματοοικονομικά, όπου η ανάλυση συναισθήματος μπορεί να προσφέρει συμπεράσματα για τις γενικές προτιμήσεις και την διάθεση της αγοράς απέναντι σε μια εταιρία.
- Στην πολιτική, για την κατανόηση των προτιμήσεων πολιτικών προσώπων και απόψεων καθώς και για την συγκέντρωση τάσεων και την εξαγωγή προβλέψεων σχετικά με προεκλογικά αποτελέσματα.

1.1.3 Τεχνικές Ταξινόμησης Συναισθήματος

Οι ταξινομήσεις συναισθημάτων μπορούν να κατηγοριοποιηθούν σε τρία διαφορετικά επίπεδα: σε επίπεδο εγγράφου, σε επίπεδο πρότασης και σε επίπεδο χαρακτηριστικών.

- Στην ταξινόμηση συναισθήματος σε επίπεδο εγγράφου (Document level) καθορίζεται η πολικότητα ενός εγγράφου στην ολότητα του, κατατάσσοντας την σε θετική, αρνητική ή ουδέτερη. Σε αυτήν την ταξινόμηση γίνεται προσπάθεια να κατηγοριοποιηθεί ο γενικός συναισθηματικό προσανατολισμός του κειμένου και κατ' επέκταση η άποψη και η προδιάθεση του συγγραφέα απέναντι στην υπό σχολιασμό οντότητα.
- Αντίθετα, σε επίπεδο πρότασης (Sentence level), η ταξινόμηση συναισθήματος πραγματοποιείται μεμονωμένα για κάθε πρόταση του εξεταζόμενου κειμένου.
- Τόσο η ταξινόμηση σε επίπεδο εγγράφου όσο και σε επίπεδο προτάσεων δεν προσπαθούν να ανακαλύψουν τι πραγματικά άρεσε ή όχι στους συγγραφείς των κειμένων. Με την ταξινόμηση συναισθήματος σε επίπεδο χαρακτηριστικών (Feature level), αντί να κοιτάζουμε τις δομές της γλώσσας (έγγραφα, παραγράφους, προτάσεις, ρήτρες ή φράσεις), εξετάζουμε άμεσα την ίδια τη γνώμη. Η ταξινόμηση αυτή βασίζεται στην ιδέα ότι μια γνώμη συνίσταται τόσο σε ένα συναισθημα (θετικό ή αρνητικό) όσο και σε μια οντότητα-στόχο (της γνώμης).

Ανεξαρτήτως σε τι επίπεδο εφαρμόζεται η ταξινόμηση συναισθήματος, σε γενικά πλαίσια, οι τεχνικές για την υλοποίησή της εμπίπτουν σε τρεις κατηγορίες: 1) στις τεχνικές βασισμένες σε μηχανική μάθηση (machine learning) 2) στις lexicon-based τεχνικές 3) και στις υβριδικές. Στις πρώτες, επιστρατεύονται όλοι οι γνωστοί αλγόριθμοι μηχανικής μάθησης για την ταξινόμηση του συναισθήματος, με κύριους και δημοφιλέστερους αυτούς που βασίζονται στην επιβλεπόμενη μάθηση. Στις lexicon-based τεχνικές γίνεται χρήση λεξικών συναισθήματος με συναισθηματικά πολωμένες λέξεις που χρησιμοποιούνται για το έργο της ταξινόμησης. Τα λεξικά αυτά περιέχουν την πολικότητα της κάθε λέξης είτε αυτές έχουν θετικό, είτε αρνητικό

συναισθηματικό αντίκτυπο. Τέλος, οι υβριδικές τεχνικές αποτελούν έναν συνδυασμό των προηγούμενων δύο κατηγοριών.

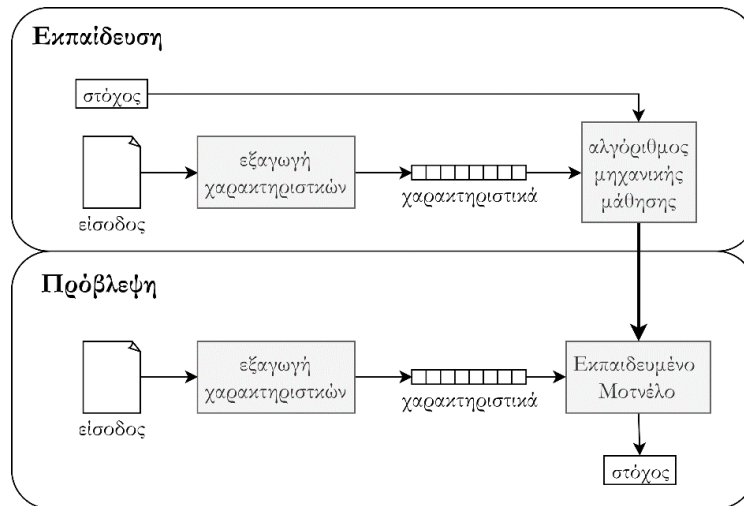
1.1.3.1 *Μηχανική Μάθηση*

Η ταξινόμηση συναισθήματος με επιβλεπόμενη μηχανική μάθηση χρησιμοποιεί δύο βασικά σύνολα δεδομένων: ένα σύνολο εκπαίδευσης (training dataset) και ένα σύνολο ελέγχου (testing dataset). Ένας αυτόματος ταξινομητής μαθαίνει τους κανόνες ταξινόμησης του εγγράφου από το σύνολο δεδομένων εκπαίδευσης και η ακρίβεια στην ταξινόμηση αυτή αξιολογείται χρησιμοποιώντας το σύνολο ελέγχου.

Το πρώτο βήμα στην ταξινόμηση με επιβλεπόμενη μηχανική μάθηση είναι αρχικά να δημιουργηθεί ή να επιλεγθεί το σύνολο εκπαίδευσης και στη συνέχεια ο κατάλληλος ταξινομητής.

Το επόμενο και ένα από τα βασικότερα βήματα, είναι η μετατροπή του σχετικού κειμένου σε μια αριθμητική αναπαράσταση, συνήθως ένα διάνυσμα, προκειμένου αυτή να δοθεί ως είσοδος στο σύστημα μηχανικής μάθησης. Συνήθως, κάθε στοιχείο του διανύσματος αντιπροσωπεύει τη συχνότητα μιας λέξης ή έκφρασης σε ένα προκαθορισμένο λεξικό (π.χ. ένα λεξικό πολωμένων λέξεων). Αυτή η διαδικασία είναι γνωστή ως εξαγωγή χαρακτηριστικών ή διανυσματοποίηση του κειμένου και η κλασική προσέγγιση είναι οι μέθοδοι bag-of-words, bag-of-ngrams κλπ. Πιο πρόσφατα, έχουν γίνει αρκετά δημοφιλείς και εφαρμόζονται νέες τεχνικές εξαγωγής χαρακτηριστικών που βασίζονται στην αναπαράσταση των λέξεων στον διανυσματικό χώρο (word embeddings). Αυτό το είδος αναπαραστάσεων καθιστά δυνατό, λέξεις με παρόμοιο σημασιολογική συνάφεια να έχουν και παρόμοια αναπαράσταση, κάτι το οποίο μπορεί να βελτιώσει την απόδοση των ταξινομητών.

Για την παρούσα προσέγγιση, έχει χρησιμοποιηθεί ένα πλήθος από δημοφιλής αλγορίθμους μηχανικής μάθησης που υλοποιούν τον ρόλο του ταξινομητή. Τέτοιοι αλγόριθμοι περιλαμβάνουν τα Μπεϋζιανά Δίκτυα (Bayesian Networks), την απλή Μπεϋζιανή Ταξινόμηση (Naive Bayesian Classification), την Μεγιστοποίηση Εντροπίας (Maximum Entropy), της Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines) και τα Νευρωνικά Δίκτυα (Neural Networks). Στα Νευρωνικά Δίκτυα, χάριν στην ανάπτυξη που έχει γνωρίσει το υλικό (hardware) τα τελευταία χρόνια και στην μεγάλη διαθεσιμότητα των υπολογιστικών πόρων, έχουν δοκιμαστεί και μοντέλα με βαθιές δομές, υλοποιώντας τεχνικές που emπίπτουν στον πεδίο της βαθιάς μάθησης (Deep Learning).



Εικόνα 1 Διάγραμμα λειτουργίας μοντέλου επιβλεπόμενης μηχανικής μάθησης για ταξινόμηση συναισθήματος

1.1.3.2 *Lexicon-based*

Η Lexicon-based προσέγγιση για την ταξινόμηση συναισθήματος, χρησιμοποιεί ένα λεξικό συναισθημάτων στο οποίο περιλαμβάνονται λέξεις που σχετίζονται με μια άποψη ή συναισθηματική πόλωση και οι οποίες αντιστοιχίζονται με το περιεχόμενο του σχετικού κειμένου προκειμένου να προσδιοριστεί η πόλωση του. Υπάρχουν τρεις μέθοδοι για την κατασκευή ενός λεξικού συναισθημάτων: η χειροκίνητη κατασκευή, η κατασκευή με μεθόδους που βασίζονται σε ένα σύνολο κειμένων και οι μέθοδοι που βασίζονται σε λεξικό. Η χειροκίνητη κατασκευή συνήθως είναι δύσκολη και χρονοβόρα εργασία. Οι μέθοδοι που βασίζονται σε σύνολα κειμένων, κάνουν χρήση κειμένων με γνωστή συναισθηματική φόρτιση και μπορούν να παράγουν λέξεις συναισθηματικής πόλωσης και άποψης με σχετικά υψηλή ακρίβεια. Τέλος, στις τεχνικές που βασίζονται σε κάποιο λεξικό, η ιδέα είναι στο πρώτο βήμα να συγκεντρωθεί χειροκίνητα ένα μικρό σύνολο λέξεων με γνωστό συναισθηματικό προσανατολισμό. Στη συνέχεια χρησιμοποιώντας το WordNet λεξικό, στόχος είναι να βρεθούν συνώνυμα και αντώνυμα αυτών των λέξεων αντιστοιχίζοντας ανάλογα το συναίσθημά τους, προκειμένου στο τέλος να δημιουργηθεί το εμπλουτισμένο επιθυμητό λεξικό.

1.2 Αντικείμενο της διπλωματικής

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η επίλυση του προβλήματος της ταξινόμησης συναισθήματος σε επίπεδο εγγράφου (Document Level Analysis) μέσω τεχνικών μηχανικής μάθησης. Πιο συγκεκριμένα, η υλοποίηση του ταξινομητή θα γίνει με χρήση νευρωνικών δικτύων και ειδικότερα με χρήση αναδρομικών νευρωνικών δικτύων μακράς βραχυπρόθεσμης μνήμης (Long Short Term Memory ή LSTM).

Η διπλωματική εργασία στοχεύει στην ανάπτυξη τριών μοντέλων βαθιάς μάθησης (Deep Learning), η αρχιτεκτονική των οποίων βασίζεται στην προαναφερθείσα κατηγορία νευρωνικών δικτύων. Σκοπός είναι η συγκριτική τους μελέτη και η εξέταση της τελικής τους απόδοσης στην ταξινόμηση συναισθήματος. Η ανάπτυξη των εν λόγω μοντέλων θα πραγματοποιηθεί με το εργαλείο μηχανικής μάθησης TensorFlow σε γλώσσα Python.

Η ταξινόμηση θα εφαρμοστεί σε ένα σύνολο κριτικών ταινιών που έχει συγκεντρωθεί από την βάση δεδομένων του IMDb. Κάθε κριτική θα αναγνωρίζεται από τον εκάστοτε αλγόριθμο βαθιάς μάθησης ως έγγραφο. Για την εξαγωγή των χαρακτηριστικών αυτών των εγγράφων και την αναπαράστασή τους, θα χρησιμοποιηθεί η τεχνική της αναπαράστασης των λέξεων στον διανυσματικό χώρο (word embeddings) ακολουθώντας την προσέγγιση που προτείνει η μέθοδος GloVe. Τα διανύσματα αυτά θα τροφοδοτούνται στο αλγόριθμο με σκοπό αυτός να αποφανθεί για την συνολική συναισθηματική τους ταξινόμηση.

1.3 Δομής της Διπλωματικής εργασίας

Η διπλωματική εργασία στην γενική της εικόνα, χωρίζεται λογικά σε δύο ενότητες: στο θεωρητικό υπόβαθρο που είναι απαραίτητο να θεμελιωθεί για την μετέπειτα παρουσίαση των μοντέλων και στην προτεινόμενη προσέγγιση του προβλήματος και την τελική υλοποίηση της. Τα κεφάλαια της διπλωματικής ταξινομούνται στις λογικές αυτές ενότητες ως εξής:

- **Θεωρητικό υπόβαθρο:** Κεφάλαια 2,3,4,5

- **Προσέγγιση προβλήματος και υλοποίηση:** Κεφάλαια 6,7,8

Συγκεκριμένα, το περιεχόμενο αυτών των κεφαλαίων είναι το εξής:

- **Κεφάλαιο 2**

Το Κεφάλαιο 2 αποτελεί μια εισαγωγή στην Μηχανική Μάθηση (Machine Learning) και στο αντικείμενο που πραγματεύεται. Εδώ θεμελιώνονται οι βασικές έννοιές της, οι οποίες

θα χρησιμοποιηθούν και θα βοηθήσουν στην κατανόηση των νευρωνικών δικτύων που αναλύονται παρακάτω.

- **Κεφάλαιο 3**

Στο κεφάλαιο 3 τίθεται όλο το απαραίτητο θεωρητικό υπόβαθρο για την περιγραφή και ανάλυση των νευρωνικών δικτύων. Παρουσιάζονται οι κατηγορίες των νευρωνικών δικτύων (πρόσθιας τροφοδότησης και αναδρομικά), εξηγείται η λειτουργία τους και ο τρόπος με τον οποίο επεξεργάζονται την πληροφορία καθώς και η διαδικασία της εκπαίδευσής τους μαζί με τις μεθόδους βελτιστοποίησης που την συνοδεύουν.

- **Κεφάλαιο 4**

Στο κεφάλαιο 4 γίνεται η εισαγωγή στα αναδρομικά δίτυα μακράς βραχυπρόθεσμης μνήμης (LSTM) τα οποία αποτελούν το βασικό μοντέλο νευρωνικών δικτύων που θα χρησιμοποιηθεί στις αρχιτεκτονικές βαθιάς μάθησης που εξετάζει η διπλωματική αυτή. Συγκεκριμένα αναφέρεται τι πρόβλημα επιλύουν και εξηγείται η λειτουργία τους.

- **Κεφάλαιο 5**

Στο κεφάλαιο 5 παρουσιάζεται το εργαλείο TensorFlow στο οποίο έχει βασιστεί η υλοποίηση των μοντέλων βαθιάς μάθησης της αρχιτεκτονικής. Γίνεται σύντομη αναφορά στο προγραμματιστικό του μοντέλο και στα βασικά του δομικά στοιχεία καθώς παρέχεται και παράδειγμα της λειτουργίας του

- **Κεφάλαιο 6**

Στο κεφάλαιο 6 περιγράφεται η προσέγγιση που ακολουθεί η παρούσα διπλωματική για το πρόβλημα της ταξινόμησης συναισθήματος. Αρχικά αναλύεται η επιλεγμένη μέθοδος για την αναπαράσταση των λέξεων στον διανυσματικό χώρο (word embeddings), γνωστή ως μέθοδος GloVe (Global Vectors for Word Representation). Στην συνέχεια παρουσιάζονται οι αρχιτεκτονικές των προτεινόμενων μοντέλων βαθιάς μάθησης καθώς και λεπτομέρειες σχετικά με την εκπαίδευσή τους.

- **Κεφάλαιο 7**

Στο κεφάλαιο παρουσιάζονται οι λεπτομέρειες την υλοποίησης των προτεινόμενων μοντέλων βαθιάς μάθησης μαζί με τα τελικά αποτελέσματα των προσομοιώσεων σχετικά

με την απόδοση ταξινόμησής τους με σκοπό την διεξαγωγή της συγκριτικής μελέτης. Τέλος προτείνονται πιθανοί τρόποι επέκτασης των σχετικών μοντέλων.

ΚΕΦΑΛΑΙΟ 2

ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

2.1 Γενικά

Η Μάθηση (Learning) είναι μία από τις θεμελιώδεις ιδιότητες της νοήμονος συμπεριφοράς του ανθρώπου. Αν και η έννοια της μάθησης, παρά τις μελέτες από πολλά επιστημονικά πεδία, δεν έχει γίνει πλήρως κατανοητή, όπως γίνεται αντιληπτή στην καθημερινή ζωή μπορεί να συνδεθεί με δύο βασικές ιδιότητες:

- την ικανότητα απόκτησης γνώσης μέσω της αλληλεπίδρασης με το περιβάλλον
- την ικανότητα βελτίωσης του τρόπου με τον οποίο εκτελείται μια ενέργεια μέσω της επανάληψης.

Ο άνθρωπος μέσω της παρατήρησης, μπορεί και κατανοεί το περιβάλλον του κατασκευάζοντας απλοποιημένες και αφαιρετικές εκδοχές αυτού, τα λεγόμενα *μοντέλα*, ενώ ταυτόχρονα έχει την δυνατότητα να οργανώνει και να συσχετίζει εμπειρίες, παράγοντας νέες δομές που ονομάζονται *πρότυπα*. Η δημιουργία μοντέλων και προτύπων μέσω ενός συνόλου δεδομένων από ένα υπολογιστικό σύστημα, ονομάζεται Μηχανική Μάθηση.

Ως κλάδος της Τεχνητής Νοημοσύνης, η Μηχανική Μάθηση έχει σαν σκοπό τη δημιουργία συστημάτων ικανών να μαθαίνουν, να βελτιώνουν, δηλαδή, την απόδοσή τους σε κάποιους τομείς μέσω της αξιοποίησης προηγούμενης γνώσης και εμπειρίας. Η Μηχανική Μάθηση, μας δίνει τη δυνατότητα να αντιμετωπίσουμε εργασίες που είναι πολύ δύσκολο να επιλυθούν με προγράμματα ρητά γραμμένα και σχεδιασμένα από ανθρώπους. Από επιστημονική και φιλοσοφική άποψη, η μελέτη της είναι ενδιαφέρουσα, διότι η ανάπτυξη της κατανόησής μας για αυτήν, συνεπάγεται την ανάπτυξη της κατανόησης μας για τις αρχές που διέπουν γενικά τη νοημοσύνη.

2.2 Θεμελίωση

Σε πρακτικό επίπεδο, όταν μιλάμε για ένα σύστημα Μηχανικής Μάθησης, ουσιαστικά μιλάμε για έναν αλγόριθμο Μηχανικής Μάθησης ο οποίος εφαρμόζεται σε ένα *σύνολο δεδομένων* (dataset) και έχει την δυνατότητα να μαθαίνει από αυτό. Σύμφωνα με τον Mitchell (1997):

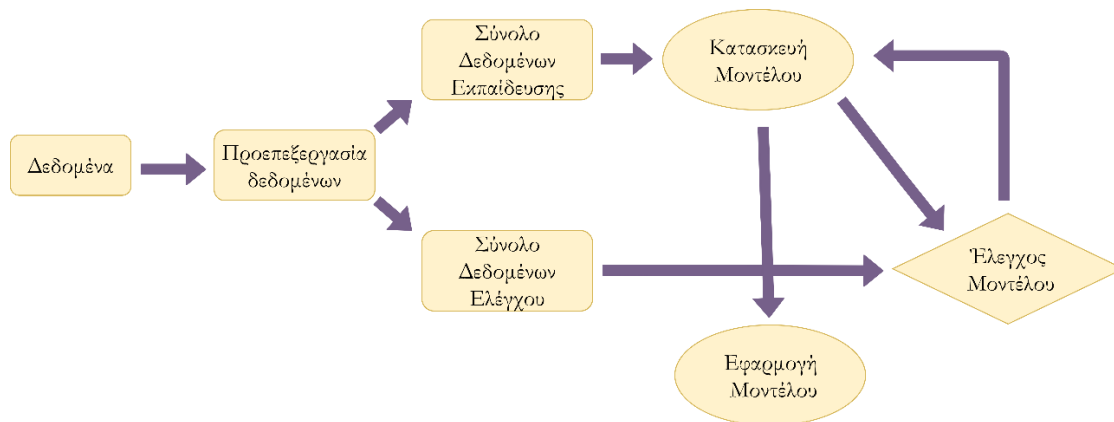
«Ένα πρόγραμμα υπολογιστή λέμε ότι μαθαίνει από την εμπειρία E ως προς κάποια κλάση εργασιών T και μέτρο απόδοσης P , αν η απόδοσή του σε εργασίες από το T , όπως μετρείται από το P , βελτιώνεται μέσω της εμπειρίας E .»

Οι *κλάσεις εργασιών* T συνήθως, περιγράφονται με βάση τον τρόπο που ο αλγόριθμος Μηχανικής Μάθησης επεξεργάζεται ένα *πρότυπο* (example). Για ένα σύστημα Μηχανικής Μάθησης, *πρότυπο* είναι ένα στοιχείο από τα δεδομένα (dataset) του προβλήματος, το οποίο έχει μια συγκεκριμένη αναπαράσταση (representation) την οποία μπορεί να καταλαβαίνει. Για την αναπαράσταση αυτήν, χρησιμοποιείται μια συλλογή *χαρακτηριστικών* (features). Τυπικά αναπαριστούμε ένα πρότυπο ως ένα διάνυσμα $x \in \mathbb{R}^n$ όπου κάθε στοιχείο x_i του διανύσματος, αποτελεί και ένα διαφορετικό χαρακτηριστικό. Έτσι, αν για παράδειγμα θέλουμε να σχεδιάσουμε ένα σύστημα Μηχανικής Μάθησης το οποίο θα μπορεί να αναγνωρίζει τα αντικείμενα σε μια φωτογραφία, η *κλάση εργασίας* T είναι το πρόβλημα ταξινόμησης-κατηγοριοποίησης (classification) των φωτογραφιών, *πρότυπα* είναι οι φωτογραφίες οι οποίες έχουν ως *χαρακτηριστικά* τα pixels τους. Αυτά τα χαρακτηριστικά θα δοθούν, για κάθε πρότυπο, ως είσοδος στον αλγόριθμο.

Για να αξιολογήσουμε τις ικανότητες ενός αλγορίθμου Μηχανικής Μάθησης, πρέπει να σχεδιάσουμε ένα ποσοτικό μέτρο της απόδοσής του. Συνήθως αυτό το μέτρο απόδοσης είναι συγκεκριμένο για την εργασία που εκτελείται από το σύστημα. Για εργασίες όπως η ταξινόμηση-κατηγοριοποίηση, συχνά μετράμε την *ακρίβεια* (accuracy) με την οποία ταξινομεί το μοντέλο. Η *ακρίβεια* είναι το ποσοστό των παραδειγμάτων για τα οποία το μοντέλο παράγει τη σωστή έξοδο σε σχέση με τον αρχικό αριθμό των παραδειγμάτων που κλήθηκε να επεξεργαστεί.

Η βασικότερη φάση κάθε αλγορίθμου Μηχανικής Μάθησης είναι η φάση της εκπαίδευσης (training phase), όπου ο αλγόριθμος χρησιμοποιεί ως είσοδο ένα σύνολο δεδομένων πάνω στα οποία εκπαιδεύεται (training set), προς επίτευξη του σκοπού του, δηλαδή τη δημιουργία νέας γνώσης. Την εκπαίδευση ακολουθεί η φάση της πιστοποίησης-ελέγχου (validation/testing phase) της παραγόμενης νέας γνώσης. Σε αυτήν την φάση υπολογίζεται η ακρίβεια και η αποδοτικότητα του αλγορίθμου με τη βοήθεια δεδομένων ελέγχου (test set) και εξάγονται όλα τα συμπεράσματα σχετικά με τον τρόπο που οι διάφοροι παράμετροι επηρεάζουν το τελικό μοντέλο. Τέλος, ο αλγόριθμος περνάει στο στάδιο εφαρμογής (application phase) δηλαδή η νέα γνώση δίνεται προς

χρήση σε εφαρμογές στις οποίες αυτή είναι απαραίτητη προκειμένου να λυθούν πραγματικά προβλήματα.



Εικόνα 2 Διάγραμμα ροής μοντέλου επιβλεπόμενης μάθησης

2.3 Είδη Μηχανικής Μάθησης

Εν γένει, ο τομέας της Μηχανικής Μάθησης αναπτύσσει τρεις τρόπους μάθησης, ανάλογους με τους τρόπους με τους οποίους μαθαίνει ο άνθρωπος: την *επιβλεπόμενη μάθηση*, τη *μη επιβλεπόμενη μάθηση* και την *ενισχυτική μάθηση*. Ωστόσο ιδιαίτερη αναφορά αξίζει να κάνουμε και στην *βαθιά μάθηση* αν και η τελευταία αποτελεί μια ειδική κατηγορία της μηχανικής μάθησης και κυρίως των νευρωνικών δικτύων και δεν μπορεί να χαρακτηριστεί ως μια επιπλέον κατηγορία της παραπάνω γενικής κατηγοριοποίησης.

2.3.1 Επιβλεπόμενη Μάθηση

Στην *επιβλεπόμενη μάθηση* (*Supervised Learning*) ή *μάθηση με παραδείγματα*, δοσμένων των μεταβλητών εισόδου X (ανεξάρτητες) και των μεταβλητών εξόδου Y (εξαρτημένες), το σύστημα καλείται να μάθει την συνάρτηση-στόχο Φ , η οποία χρησιμοποιείται για την πρόβλεψη των Y μέσω των X , δηλαδή την σχέση $Y = \Phi(X)$. Η συνάρτηση αυτή αποτελεί έκφραση του μοντέλου που περιγράφει τα δεδομένα του εκάστοτε προβλήματος. Ονομάζεται επιβλεπόμενη μάθηση καθώς είναι μια διαδικασία εκπαίδευσης που μπορεί να παρομοιαστεί με ένα «δάσκαλο» που επιτηρεί την μάθηση του συστήματος και διορθώνει τις προβλέψεις του με βάση τις γνωστές –

επιθυμητές εξόδους (Y). Στην επιβλεπόμενη μάθηση διακρίνονται δυο κατηγορίες προβλημάτων:

- Τα προβλήματα ταξινόμησης (*classification*), που σχετίζονται με την δημιουργία μοντέλων που προβλέπουν (κατηγοριοποιούν) μεταξύ διακριτών κλάσεων (π.χ. ταξινόμησης του συναισθήματος μιας πρότασης σε μια από τις κλάσεις θετικό ή αρνητικό)
- Τα προβλήματα παρεμβολής (*regression*), που σχετίζονται με την δημιουργία μοντέλων που προβλέπουν αριθμητικές τιμές σε συνεχή διαστήματα.

2.3.2 Μη Επιβλεπόμενη Μάθηση

Στην μη επιβλεπόμενη μάθηση (*Unsupervised Learning*) ή μάθηση από παρατήρηση, το σύστημα, δοσμένων μόνο των μεταβλητών εισόδου X , πρέπει να διερευνήσει και να ανακαλύψει συσχετίσεις για τα δοσμένα δεδομένα, δημιουργώντας κατάλληλα πρότυπα, χωρίς όμως να γνωρίζει τις επιθυμητές εξόδους. Τα προβλήματα στην μη επιβλεπόμενη μάθηση ομαδοποιούνται σε δύο κατηγορίες:

- Στα προβλήματα ομαδοποίησης (*clustering*): Σε ένα πρόβλημα ομαδοποίησης, ο αλγόριθμός μας προσπαθεί να ανακαλύψει εγγενείς ομαδοποιήσεις μεταξύ των δεδομένων στα οποία εφαρμόζεται (π.χ. ομαδοποιήσεις των πελατών με βάση την αγοραστική τους συμπεριφορά):
- Στα προβλήματα συσχέτισης (*association*): Σε ένα πρόβλημα συσχέτισης, ο αλγόριθμός μας προσπαθεί να ανακαλύψει κανόνες (rules) που χαρακτηρίζουν ένα μέρος των δεδομένων μας (π.χ. από ένα μεγάλο αριθμό αντικειμένων ζητείται να βρεθεί για ένα υποσύνολο τους, ποια αντικείμενα αγοράζονται μαζί)

2.3.3 Ενισχυτική Μάθηση

Στην ενισχυτική μάθηση (*Reinforcement Learning*), ο αλγόριθμος μαθαίνει μια στρατηγική ενεργειών μέσα από άμεση αλληλεπίδραση με το περιβάλλον. Χρησιμοποιείται κυρίως σε προβλήματα Σχεδιασμού (Planning), όπως για παράδειγμα ο έλεγχος κίνησης ρομπότ και η βελτιστοποίηση εργασιών σε εργοστασιακούς χώρους.

2.3.4 Βαθιά Μάθηση

Ένας ιδιαίτερος κλάδος της μηχανικής μάθησης, η βαθιά μάθηση, έχει αποδειχθεί ιδιαίτερα αποτελεσματικός τα τελευταία χρόνια. Η βαθιά μάθηση είναι μια οικογένεια αλγορίθμων εκμάθησης αναπαράστασης που χρησιμοποιεί σύνθετες αρχιτεκτονικές νευρωνικών δικτύων με μεγάλο αριθμό κρυφών επιπέδων, το καθένα από τα οποία αποτελείται από απλούς αλλά μη γραμμικούς μετασχηματισμούς (νευρώνες) τους οποίους εφαρμόζει στα δεδομένα

εισόδου. Δεδομένου ότι τα δομικά αυτά στοιχεία μετασχηματισμού παρουσιάζονται σε πολύ μεγάλο πλήθος στις αρχιτεκτονικές αυτές, δίνεται η δυνατότητα για την διαμόρφωση πολύ σύνθετων μοντέλων τα οποία μπορούν να επιλύσουν περίπλοκα και απαιτητικά προβλήματα ταξινόμησης, παλινδρόμησης και πολλές άλλες μαθησιακές εργασίες.

Υπάρχουν αρκετοί λόγοι που οδήγησαν στην ανάπτυξη της βαθιάς μάθησης και στην τοποθέτησή της στο κέντρο του τομέα της μηχανικής μάθησης τις τελευταίες δεκαετίες. Ένας λόγος, ίσως και ο κυριότερος, αντιπροσωπεύεται ασφαλώς από την πρόοδο στον τομέα του υλικού (hardware), με τη διαθεσιμότητα νέων επεξεργαστών, όπως μονάδες επεξεργασίας γραφικών (GPU), οι οποίες μείωσαν σημαντικά τον χρόνο που απαιτείται για την εκπαίδευση τέτοιων σύνθετων δικτύων, δεδομένης την υψηλής υπολογιστικής τους πολυπλοκότητας. Ένας άλλος λόγος είναι η συνεχώς αυξανόμενη διαθεσιμότητα σε σύνολα δεδομένων για την κατάρτιση ενός τέτοιου συστήματος, που απαιτείται για την εκπαίδευση αρχιτεκτονικών ορισμένου βάθους και με μεγάλες διαστάσεις στα δεδομένα εισόδου.

Η βαθιά μάθηση, ως προϊόν των νευρωνικών δικτύων, βασίζεται στον τρόπο με τον οποίο ο ανθρώπινος εγκέφαλος επεξεργάζεται πληροφορίες και μαθαίνει, ανταποκρινόμενος σε εξωτερικά ερεθίσματα. Αποτελεί ένα μοντέλο μηχανικής μάθησης, αποτελούμενο από διάφορα επίπεδα αναπαράστασης, όπου τα βαθύτερα επίπεδα λαμβάνουν ως είσοδο τα αποτελέσματα των προηγούμενων, δημιουργώντας όλο και πιο αφηρημένες αναπαραστάσεις της πληροφορίας.

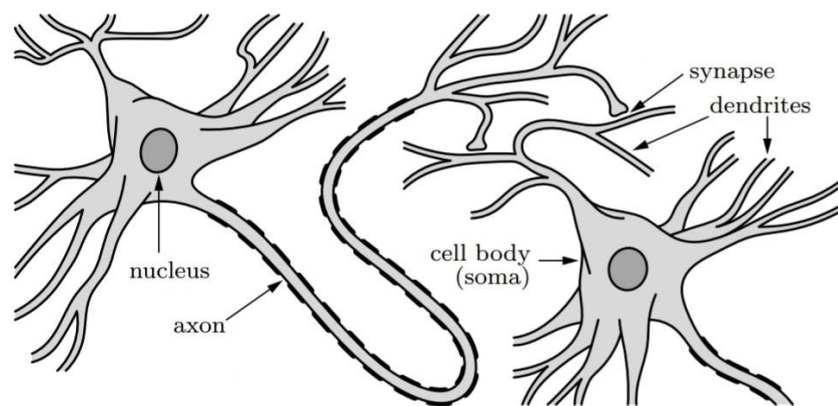
ΚΕΦΑΛΑΙΟ 3

ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Τα Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ) είναι μια περίπτωση συστήματος Μηχανική Μάθησης και αποτελούν υπολογιστικά-αλγοριθμικά μοντέλα επεξεργασίας δεδομένων, τα οποία αποτελούνται από ένα πλήθος τεχνητών νευρώνων και έχουν σαν σκοπό τη προσέγγιση και προσομοίωση της δομής και της λειτουργίας του ανθρώπινου εγκεφάλου.

3.1 Βιολογικά Νευρωνικά Δίκτυα

Ο εγκέφαλος είναι ένα εξαιρετικά πολύπλοκο, μη γραμμικό, παράλληλο σύστημα επεξεργασίας πληροφοριών, στο οποίο ο άνθρωπος οφείλει την ικανότητα του να σκέφτεται, να θυμάται και να επιλύει προβλήματα. Όπως είναι γνωστό από την Βιολογία, η δομική μονάδα του εγκεφάλου είναι οι νευρώνες. Ο τυπικός βιολογικός νευρώνας είναι ένα μεγάλο σε μέγεθος κύτταρο, το οποίο αποτελείται από το σώμα, τους δένδριτες και τον άξονα.



Εικόνα 3 Αναπαράσταση βιολογικού νευρώνα

Πιο συγκεκριμένα, το σώμα περιλαμβάνει τον πυρήνα του νευρώνα, οι δενδρίτες αποτελούν τα τμήματα μέσω των οποίων ο νευρώνας λαμβάνει τα σήματα από τους γειτονικούς τους, ενώ ο άξονας είναι η έξοδος του και το μέσο σύνδεσής του με άλλους νευρώνες. Σε ένα νευρωνικό δίκτυο, ένας νευρώνας συνδέεται με χιλιάδες άλλους, μέσω σημείων σύνδεσης που ονομάζονται *συνάψεις*. Καθώς η πληροφορία στο νευρωνικό δίκτυο μεταδίδεται μέσω ηλεκτρικών παλμών, οι συνάψεις έχουν την δυνατότητα να επιταχύνουν ή επιβραδύνουν την ροή των ηλεκτρικών φορτίων προς το σώμα του νευρώνα, καθορίζοντας έτσι το σήμα που θα φτάσει σε αυτόν. Τα ηλεκτρικά σήματα που εισέρχονται στο σώμα του μέσω των δενδριτών, συνδυάζονται και αν το αποτέλεσμα ξεπερνά κάποια τιμή κατωφλίου, ο νευρώνας πυροδοτείται και το σήμα διαδίδεται προς τους υπόλοιπους νευρώνες.

3.2 Τεχνητά Νευρωνικά Δίκτυα

Τα Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ) επεξεργάζονται τις πληροφορίες με παρόμοιο τρόπο που κάνει ο ανθρώπινος εγκέφαλος. Όπως ένα βιολογικό έτσι και ένα τεχνητό νευρωνικό δίκτυο, αποτελείται από ένα μεγάλο αριθμό εξαιρετικά διασυνδεδεμένων στοιχείων επεξεργασίας και αποθήκευσης, τους τεχνητούς νευρώνες, που εργάζονται παράλληλα για την επίλυση συγκεκριμένου προβλήματος. Οι δυνατότητες και ο τρόπος λειτουργίας των νευρώνων σε συνδυασμό με αυτήν την αρχιτεκτονική, δίνει στα ΤΝΔ κάποιες βασικές ιδιότητες, χάριν στις οποίες, αυτά βρίσκονται ανάμεσα στις πιο δημοφιλείς και αποδοτικές μορφές συστημάτων μηχανικής μάθησης. Αυτές οι ιδιότητες είναι:

- Ικανότητα μάθησης μέσω της εκπαίδευσης τους με παραδείγματα-δεδομένα, με αποτέλεσμα να βελτιώνουν την εμπειρία τους και να προσαρμόζονται στο περιβάλλον τους
- Ικανότητα γενίκευσης της γνώσης τους πάνω στα δεδομένα που εκπροσωπούν το πρόβλημα που πρέπει να επιλύσουν.
- Αυτό-οργάνωση, καθώς ένα τεχνητό νευρωνικό δίκτυο μπορεί να δημιουργήσει τη δική του οργάνωση των πληροφοριών που λαμβάνει κατά τη διάρκεια της εκμάθησης.
- Κατανεμημένος και παράλληλος υπολογισμός
- Ανοχή σε σφάλματα και θορυβώδεις εισόδους, με την έννοια ότι τόσο η κακή λειτουργία ή απώλεια ενός νευρώνα όσο και ελλείψεις-θορυβώδεις εισοδοί στο ΤΝΔ δεν επηρεάζουν την λειτουργία και την απόδοσή του.

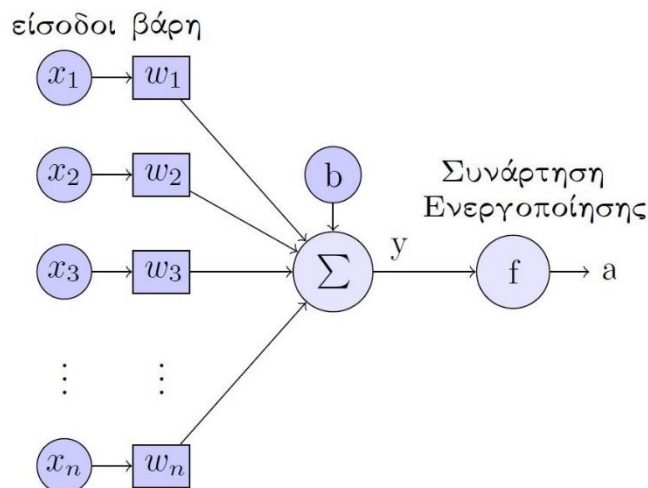
3.2.1 Μοντέλο Τεχνητού Νευρώνα

Όπως αναφέρθηκε παραπάνω, στην γενική τους εικόνα, οι νευρώνες σε ένα νευρωνικό δίκτυο (τεχνητό ή βιολογικό), διασυνδέονται μεταξύ τους μέσω συνάψεων και πυροδοτούνται όταν ένας γραμμικός συνδυασμός των εισόδων τους υπερβαίνει ένα κατώφλι.

Ο τεχνητός νευρώνας είναι ένα μαθηματικό μοντέλο αποθήκευσης και επεξεργασίας της πληροφορίας, το οποίο αποτελεί μια αφηρημένη προσέγγιση του βιολογικού νευρώνα. Πιο συγκεκριμένα αποτελείται από ένα πλήθος καναλιών εισόδου n , τις συνάψεις, μέσω των οποίων δέχεται τις εισόδους του x_i . Κάθε σύναψη του συσχετίζεται με ένα αριθμητικό βάρος W_i , τα λεγόμενα συναπτικά βάρη (σε ένα σύνθετο ΤΝΔ οι εισοδοι μπορεί να είναι τόσο δεδομένα που τροφοδοτούνται απευθείας στον νευρώνα όσο και έξοδοι από άλλους νευρώνες). Οι εισοδοι αφού πολλαπλασιαστούν με τα αντίστοιχα συναπτικά βάρη, πηγαίνουν σε έναν αθροιστή όπου και αθροίζονται, μαζί με μια αριθμητική τιμή πόλωσης b (bias). Το αποτέλεσμα y , τροφοδοτείται σε μια συνάρτηση ενεργοποίησης f , η έξοδος της οποία αποτελεί και την τελική έξοδο, γνωστή και ως *ενεργοποίηση* a του νευρώνα. Η μαθηματική έκφραση της λειτουργίας του νευρώνα είναι η εξής:

$$y = \sum_{i=1}^n w_i x_i + b = \sum_{i=0}^n w_i x_i = \mathbf{W}^T \mathbf{X}$$

$$\text{Έξοδος} = a = f(y)$$



Εικόνα 4 Μαθηματικό μοντέλο τεχνητού νευρώνα

Αυτό το μοντέλο νευρώνα αποτελεί το βασικό μοντέλο νευρώνα που δομεί κάθε είδους νευρωνικό. Η φιλοσοφία λειτουργίας αυτού το μοντέλου είναι η εξής:

- Τα βάρη κάθε σύναψης καθορίζουν το βαθμό συμμετοχής κάθε εισόδου, δηλαδή το πόσο πολύ ή το πόσο λίγο κάθε είσοδος θα επηρεάσει την ενεργοποίηση του νευρώνα.
- Η τιμή πόλωσης αποτελεί το αριθμητικό κατώφλι που θα πρέπει να ξεπεράσει η συνολική είσοδος του νευρωνικού προκειμένου ο νευρώνας να πυροδοτηθεί-ενεργοποιηθεί. Ωστόσο, τόσο η τιμή του κατωφλίου, όσο και το αν τελικά ο νευρώνας θα ενεργοποιηθεί ή όχι, θα καθοριστούν από την συνάρτηση ενεργοποίησης.
- Η τιμή $y = \sum_{i=1}^n W_i x_i + b$ του νευρώνα, μπορεί να πάρει πρακτικά τιμές από $-\infty$ έως $+\infty$, με αποτέλεσμα ο νευρώνας να μην μπορεί να ξέρει τα πραγματικά της όρια και κατ'επέκταση να μην υπάρχει κάποιο μέτρο σύγκρισης που θα καθορίζει αν αυτός θα ενεργοποιηθεί ή όχι. Το πρόβλημα αυτό, το λύνει η συνάρτηση ενεργοποίησης η οποία ουσιαστικά φράζει την y , δρα δηλαδή ως φίλτρο που την αντιστοιχίζει σε ένα συγκεκριμένο αριθμητικό διάστημα. Η πιο απλή συνάρτηση ενεργοποίησης είναι η βηματική (step function) αλλά καθώς δεν είναι διαφορίσιμη (κάτι το οποίο παίζει σημαντικό στην διαδικασία εκπαίδευσης και την απόδοση ενός νευρωνικού δικτύου) δεν χρησιμοποιείται σχεδόν ποτέ. Κάποιες από τις πιο δημοφιλείς συναρτήσεις ενεργοποίησης (Εικόνα 5) είναι οι:

- Σιγμοειδής (Sigmoid): $f y = \frac{1}{1+e^{-y}}$

- Υπερβολική Εφαπτομένη: $f y = \tanh y = \frac{1}{1+e^{-2y}} - 1$

- Rectified linear unit (ReLU): $f y = \max(0, y)$

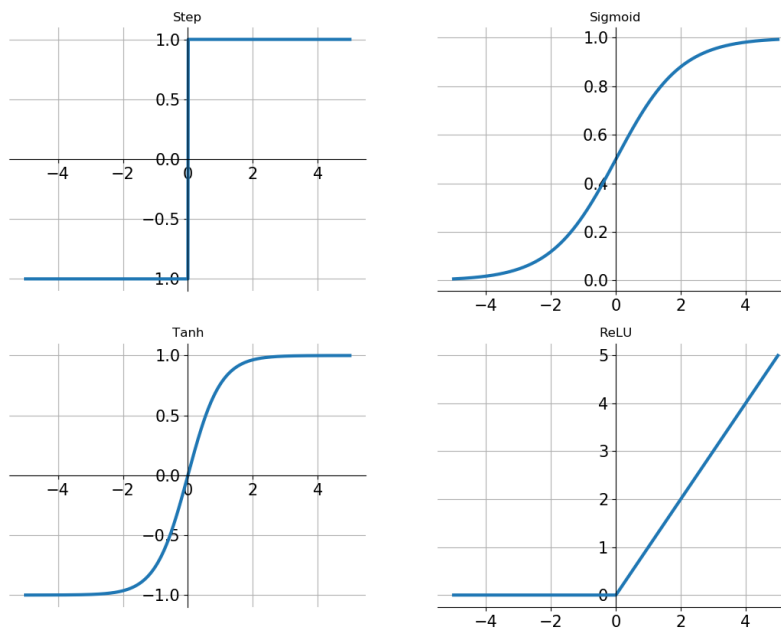
3.2.2 Δομές Τεχνητών Νευρωνικών Δικτύων

Το μοντέλο τεχνητού νευρώνα που παρουσιάστηκε παραπάνω, αποτελεί τον βασικό τεχνητό νευρώνα που δομεί την πλειοψηφία των τεχνητών νευρωνικών δικτύων. Με κριτήριο το πώς είναι συνδεδεμένοι οι νευρώνες μεταξύ τους, τα ΤΝΔ χωρίζονται σε δύο βασικές κατηγορίες:

- **Στα ΤΝΔ πρόσθιας τροφοδότησης (feedforward)**. Τα νευρωνικά δίκτυα πρόσθιας τροφοδότησης αποτελούν την πιο απλή αρχιτεκτονική νευρωνικών δικτύων. Σε αυτά τα δίκτυα οι νευρώνες οργανώνονται σε επίπεδα και κάθε νευρώνας συνδέεται μόνο με τους νευρώνες του μπροστινού γειτονικού του επιπέδου. Έτσι, η ροή της πληροφορίας σε αυτά τα

δίκτυα είναι μονής κατεύθυνσης. Ένα ΤΝΔ πρόσθιας τροφοδότησης ουσιαστικά αναπαριστά μια συνάρτηση των τρεχουσών εισόδων του.

- **Στα ΤΝΔ ανατροφοδότησης ή αναδρομικά ΤΝΔ (feedback ή recurrent).** Στα αναδρομικά νευρωνικά δίκτυα, σε αντίθεση με τα ΤΝΔ πρόσθιας τροφοδότησης, πληροφορία δεν ρέει μονόδρομα, αλλά οι έξοδοι του δικτύου ανατροφοδοτούνται πίσω στις εισόδους. Αυτό σημαίνει ότι τα επίπεδα ενεργοποίησης του δικτύου σχηματίζουν ένα δυναμικό σύστημα το οποίο μπορεί να ευσταθεί πλήρως, να παρουσιάζει ταλαντώσεις ή ακόμα και χαοτική συμπεριφορά.



Εικόνα 5 Οι συναρτήσεις ενεργοποίησης: Βηματική, Σιγμοειδής, ReLU, Υπερβολική Εφαπτομένη

3.2.3 Ανάκληση και Εκπαίδευση

Ο κύκλος ζωής ενός τεχνητού νευρωνικού δικτύου περιγράφεται από δύο βασικές λειτουργίες: την *ανάκληση* και την *εκπαίδευση*. Η ανάκληση (recall) είναι η διαδικασία κατά την οποία το νευρωνικό δίκτυο, για συγκεκριμένο διάνυσμα εισόδου, υπολογίζει το αντίστοιχο διάνυσμα εξόδου. Η εκπαίδευση (training) ή μάθηση του νευρωνικού, είναι η διαδικασία τροποποίησης της τιμής των βαρών του δικτύου, προκειμένου να τα προσαρμόσει με τέτοιο τρόπο ώστε για συγκεκριμένο διάνυσμα εισόδου, να παράγει την επιθυμητή έξοδο.

3.3 Τεχνητά Νευρωνικά Δίκτυα Πρόσθιας Τροφοδότησης

Στα ΤΝΔ πρόσθιας τροφοδότησης, η πληροφορία ρέει μονόδρομα από την είσοδο του νευρωνικού μέχρι την έξοδό του, χωρίς να ενεργοποιείται κανένας νευρώνας παραπάνω από μία φορά για συγκεκριμένη είσοδο. Σε αυτά υπάρχει ένα επίπεδο εισόδου, ένα επίπεδο εξόδου και προαιρετικά ένα ή περισσότερα ενδιάμεσα επίπεδα. Παρακάτω θα εξετάσουμε τις βασικές περιπτώσεις ΤΝΔ πρόσθιας τροφοδότησης.

3.3.1 Αισθητήρας Perceptron και Εκπαίδευση

Ο αισθητήρας perceptron αποτελεί την πιο απλή τοπολογία ΤΝΔ πρόσθιας τροφοδότησης χωρίς κρυφά επίπεδα, καθώς αποτελείται μονάχα από έναν τεχνητό νευρώνα ο οποίος δομεί το επίπεδο εξόδου και στον οποίο τροφοδοτούνται όλες οι εισοδοί. Η λειτουργία του perceptron περιγράφεται πλήρως από την λειτουργία του απλού τεχνητού νευρώνα που παρουσιάστηκε παραπάνω.

Για την εκπαίδευση του perceptron χρησιμοποιείται επιβλεπόμενη μάθηση. Η έξοδος του αισθητήρα perceptron συγκρίνεται με τις τιμές στόχους $t(p)$ κάθε προτύπου εκπαίδευσης p και μέσω του υπολογισμού του σφάλματος εξόδου υπολογίζονται και αναπροσαρμόζονται κατάλληλα τα συναπτικά βάρη w_0, w_1, \dots, w_n των εισόδων προκειμένου στο τέλος να παραχθεί η επιθυμητή έξοδος. Έτσι με τυχαία αρχικοποίηση του διανύσματος συναπτικών βαρών $\mathbf{W} = [w_0, w_1, \dots, w_n]$, συνάρτηση ενεργοποίησης f και δοσμένων των προτύπων εκπαίδευσης $\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \dots, \mathbf{X}^{(P)}$ και των αντίστοιχων στόχων $t^{(0)}, t^{(1)}, \dots, t^{(P)}$ ο επαναληπτικός αλγόριθμος εκπαίδευσης του perceptron είναι:

Μέχρι να συμπληρωθεί ο προκαθορισμένος αριθμός εποχών εκπαίδευσης ή δεν γίνει καμία μεταβολή στα συναπτικά βάρη:

Για κάθε πρότυπο εισόδου $\mathbf{X}^{(i)}$ με στόχο $t^{(i)}$:

1. Υπολόγισε την έξοδο $\mathbf{a} = \mathbf{f} \left(\sum_{j=0}^n w_j x_j^{(i)} \right)$

2. Εάν $\mathbf{a} = \mathbf{t}$, δεν γίνεται καμία αλλαγή στα συναπτικά βάρη

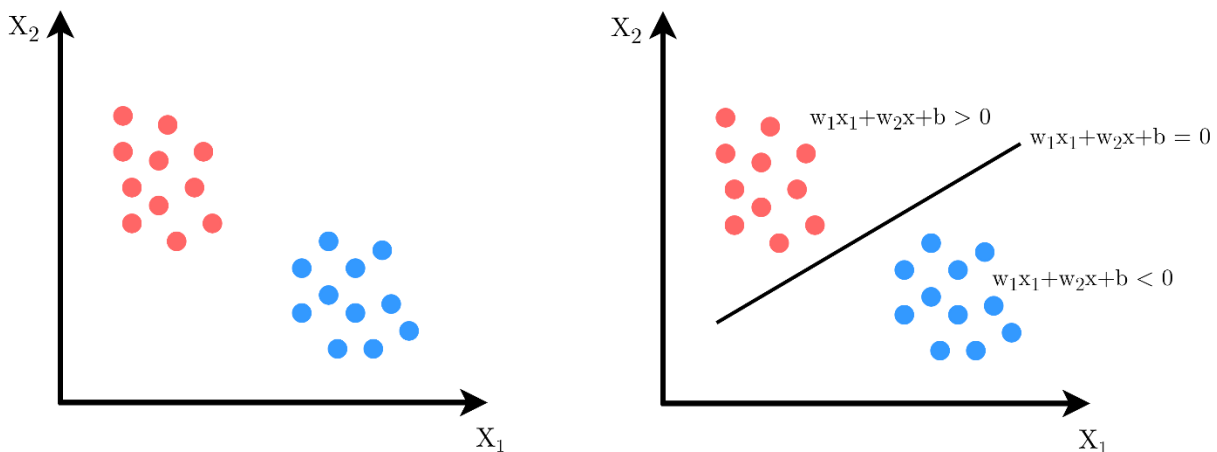
3. Εάν $\mathbf{a} \neq \mathbf{t}$, τότε μετέβαλλε τα βάρη κατά την ποσότητα $\Delta \mathbf{W} = \mathbf{d} \cdot \mathbf{t} - \mathbf{a} * \mathbf{X}$

Τα βάρη προσαρμόζονται ως εξής: $\mathbf{W}^k = \mathbf{W}^{k-1} + \Delta \mathbf{W}$, όπου \mathbf{W}^k τα βάρη της επανάληψης k

Στον παραπάνω αλγόριθμο, όπως και σε κάθε αλγόριθμο εκπαίδευσης ενός νευρωνικού δικτύου, ως *εποχή* ονομάζεται ένας πλήρης κύκλος χρήσης όλων των προτύπων εκπαίδευσης, ενώ η παράμετρος d ονομάζεται ρυθμός εκπαίδευσης (learning rate) και καθορίζει το ρυθμό μεταβολής των βαρών του νευρωνικού δικτύου.

3.3.2 Δίκτυο perceptron ως γραμμικός διαχωριστής

Σε ένα πρόβλημα ταξινόμησης μεταξύ δύο κλάσεων, στόχος του perceptron είναι να ταξινομήσει σωστά ένα σύνολο παραδειγμάτων σε μια από τις κλάσεις C_1, C_2 . Καθένα από αυτά τα παραδείγματα έχει πρότυπο με διάνυσμα χαρακτηριστικών $\mathbf{X} = [x_1, x_2, \dots, x_n]^T$, το οποίο αποτελεί την είσοδο του νευρώνα. Εξετάζοντας την ενεργοποίηση του νευρώνα $a = f(\mathbf{W}^T \mathbf{X})$, παρατηρούμε ότι η εξίσωση $\mathbf{W}^T \mathbf{X} = 0$ ορίζει ένα υπερεπίπεδο στο n -διάστατο χώρο εισόδου, όπου αν η είσοδος \mathbf{X} βρίσκεται στην μια πλευρά αυτού του επιπέδου, δηλαδή $y > 0$, τότε η βηματική ενεργοποίηση a του νευρώνα είναι 1, ενώ αν $y < 0$ η ενεργοποίηση είναι 0. Έτσι το perceptron μπορεί να λειτουργήσει σαν γραμμικός διαχωριστής μεταξύ των κλάσεων C_1, C_2 αν οι κλάσεις αυτές, στον n -διάστατο χώρο που ορίζουν τα παραδείγματά τους, μπορούν να διαχωριστούν γραμμικά, είναι δηλαδή γραμμικά διαχωρίσιμες.

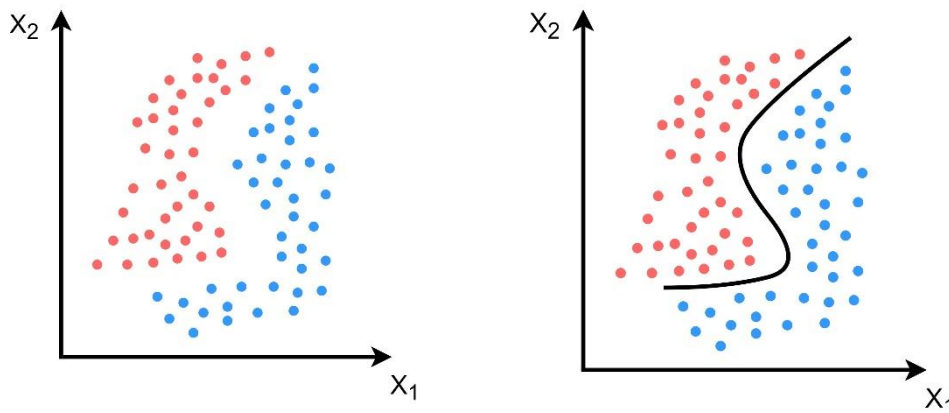


Εικόνα 6 Ταξινόμηση γραμμικά διαχωρίσιμων κλάσεων

3.3.3 Πολυεπίπεδα Δίκτυα Perceptron (Multi-Layer Perceptron)

Όπως φάνηκε παραπάνω, οι δυνατότητες των δικτύων με μόνο έναν νευρώνα, περιορίζονται στο να μπορούν αναπαριστούνε μόνο επίπεδες επιφάνειες, με αποτέλεσμα τα δίκτυα αυτά να είναι ικανά να διαχωρίσουν με αρκετά μεγάλη ακρίβεια κλάσεις, οι οποίες όμως θα πρέπει να είναι γραμμικά διαχωρίσιμες.

Παρόλ' αυτά, στην πλειοψηφία των προβλημάτων ταξινόμησης, ένα ΤΝΔ καλείται να ταξινομήσει πρότυπα τα οποία ανήκουν σε κλάσεις που διαχωρίζονται μεταξύ τους μη γραμμικά (Εικόνα 7). Για την αντιμετώπιση του προβλήματος αυτού, προστίθενται στο δίκτυο περισσότεροι νευρώνες, δημιουργώντας έτσι, πολυεπίπεδα ΤΝΔ, δηλαδή πιο σύνθετες δομές οι οποίες παράγουν και πιο σύνθετες, μη γραμμικές, επιφάνειες διαχωρισμού των κλάσεων. Τα δίκτυα αυτά είναι γνωστά ως πολυεπίπεδα δίκτυα perceptron (Multi-layer Perceptron ή MLP).



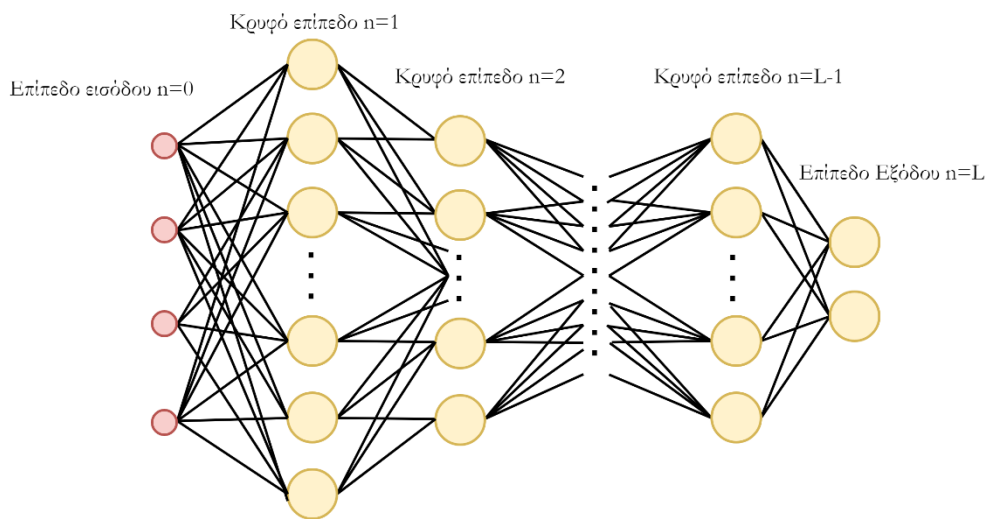
Εικόνα 7 Παράδειγμα μη γραμμικά διαχωρίσιμων κλάσεων

Οι νευρώνες στα πολυεπίπεδα δίκτυα perceptron ομαδοποιούνται και στοιβάζονται σε στρώματα-επίπεδα. Οι νευρώνες του ίδιου επιπέδου χρησιμοποιούν την ίδια συνάρτηση ενεργοποίησης, ενώ μεταξύ τους δεν υπάρχουν συνδέσεις. Συνδέσεις υπάρχουν μόνο μεταξύ των νευρώνων γειτονικών επιπέδων. Έτσι σχηματίζονται:

- Τα κρυφά επίπεδα (hidden layers). Τα επίπεδα αυτά είναι ουσιαστικά ομάδες νευρώνων που βρίσκονται ανάμεσα στην είσοδο και το επίπεδο εξόδου. Ένα πολυεπίπεδο δίκτυο perceptron μπορεί να έχει ένα ή περισσότερα κρυφά επίπεδα. Αυτό που τα ξεχωρίζει από το επίπεδο εξόδου, είναι ότι η έξοδος των νευρώνων των κρυφών επιπέδων αποτελεί πάντα είσοδο για τους νευρώνες των επόμενων επιπέδων (κρυφά ή επίπεδο εξόδου)

- Το επίπεδο εξόδου (output layer). Το επίπεδο αυτό, όπως γίνεται κατανοητό και από το όνομα, αποτελείται από νευρώνες, η έξοδος των οποίων αποτελεί και την τελική έξοδο του νευρωνικού.

Η είσοδος του νευρωνικού συνηθίζεται να θεωρείται και αυτό ως επίπεδο αλλά στην πραγματικότητα δεν αποτελείται από νευρώνες, παρό μόνο από τις εισόδους του νευρωνικού, οι οποίες, μέσω των συνάψεων των νευρώνων του πρώτου κρυφού επιπέδου, υπεισέρχονται στο νευρωνικό δίκτυο. Οι συνδέσεις μεταξύ επιπέδων είναι πυκνές με την έννοια ότι κάθε νευρώνας ενός επιπέδου, συνδέεται με όλους του νευρώνες του γειτονικού επιπέδου.



Εικόνα 8 Παράδειγμα πολυεπίπεδου τεχνητού νευρωνικού δικτύου

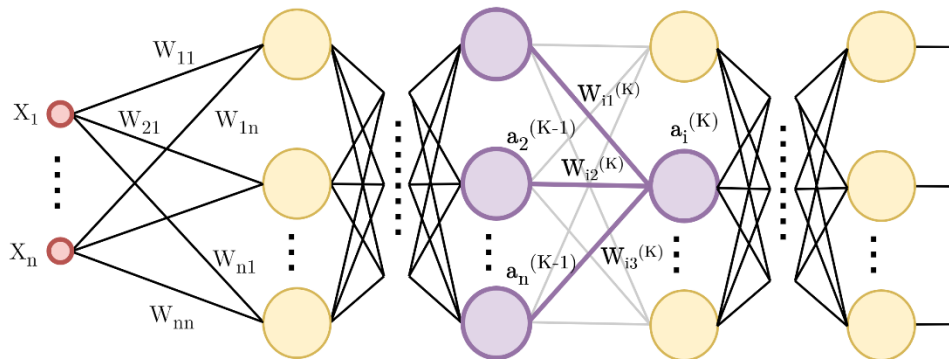
3.3.4 Ανάκληση ΤΝΔ Πρόσθιας Τροφοδότησης

Για να περιγράψουμε την ανάκληση των πολυεπίπεδων δικτύων perceptron, δηλαδή το πως υπολογίζεται η έξοδος του δικτύου δοσμένου του διανύσματος εισόδου, βασιζόμαστε στο μαθηματικό μοντέλο του απλού τεχνητού νευρώνα και χρησιμοποιούμε την εξής σημειογραφία:

- N_i , το πλήθος νευρώνων σε κάθε επίπεδο i
- $w_{ij}^{(K)}$, το συναπτικό βάρος από τον νευρώνα j του επιπέδου $K - 1$ στο νευρώνα i του επιπέδου K
- $f_i^{(K)}$, την συνάρτηση ενεργοποίησης του νευρώνα i του επιπέδου K
- $a_i^{(K)}$, την έξοδο-ενεργοποίηση του νευρώνα i του επιπέδου K

Έτσι η ενεργοποίηση του νευρώνα i του επιπέδου K υπολογίζεται ως εξής (Εικόνα 7):

$$\alpha_i^{(K)} = f_i^{(K)} \left(\sum_{j=1}^{N_{K-1}} w_{ij}^{(K)} a_j^{(K-1)} + b_i^{(K)} \right)$$



Εικόνα 9 Ενεργοποίηση του νευρώνα i στο επίπεδο K του δικτύου

Με βάση τα παραπάνω, η διαδικασία υπολογισμού της εξόδου του πολυεπίπεδου ΤΝΔ, ορίζεται με τον εξής αλγόριθμο:

Για κάθε επίπεδο K του ΤΝΔ από 1 έως L :

Για κάθε νευρώνα i στρώματος K από 1 έως N_K :

Υπολόγισε τη έξοδο του νευρώνα: $\alpha_i^{(K)} = f_i^{(K)} \left(\sum_{j=1}^{N_{K-1}} w_{ij}^{(K)} a_j^{(K-1)} + b_i^{(K)} \right)$

Αποθήκευσε την στο διάνυσμα: $[\alpha_1^{(K)}, \alpha_2^{(K)}, \dots, \alpha_{N_K}^{(K)}]$

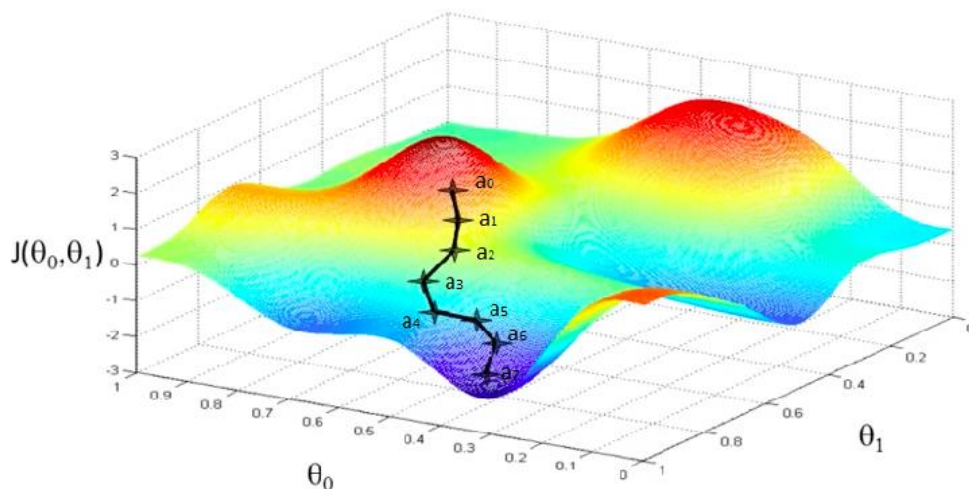
Επέστρεψε ως έξοδο το διάνυσμα: $[\alpha_1^{(L)}, \alpha_2^{(L)}, \dots, \alpha_{N_L}^{(L)}]$

3.3.5 Εκπαίδευση Πολυεπίπεδου Δικτύου Πρόσθιας Τροφοδότησης με τον αλγόριθμο Ανάστροφης Μετάδοσης Σφάλματος (Back-Propagation)

Στα πολυεπίπεδα νευρωνικά δίκτυα, όπως και στην περίπτωση του απλού perceptron, η διαδικασία της εκπαίδευσης βασίζεται στην κατάλληλη προσαρμογή των βαρών του δικτύου, προκειμένου η τελική έξοδος να προσεγγίζει όσο τον δυνατόν περισσότερο την αναμενόμενη έξοδο. Ο αλγόριθμος εκπαίδευσης των πολυεπίπεδων ΤΝΔ στηρίζεται στην εξής ιδέα: έχοντας ορίσει μια αντικειμενική συνάρτηση που θα περιγράφει το σφάλμα μεταξύ της εξόδου του δικτύου και της αναμενόμενης εξόδου, προσπαθούμε να ελαχιστοποιήσουμε την συνάρτηση αυτή ως προς τα βάρη του δικτύου, προσαρμόζοντάς τα ανάλογα με το πόσο συνεισέφεραν στο συνολικό σφάλμα.

Η πιο γνωστή και σε ευρεία χρήση μέθοδος, για την εύρεση του ελάχιστου μιας συνάρτησης σφάλματος στα νευρωνικά δίκτυα, είναι η μέθοδος κατάβασης δυναμικού (gradient descent). Η μέθοδος αυτή, είναι ένας επαναληπτικός αλγόριθμος βελτιστοποίησης πρώτης τάξης, ο οποίος βασίζεται στην παρατήρηση ότι αν μια συνάρτηση πολλών μεταβλητών $J(\mathbf{x})$ ορίζεται και είναι συνεχής στην περιοχή γύρω από ένα σημείο \mathbf{a} , τότε η συνάρτηση αυτή ελαχιστοποιείται γρηγορότερα προς την κατεύθυνση ελάττωσης της κλίσης της,

$$-\nabla_{\mathbf{x}} J(\mathbf{x}) = - \left[\frac{\partial J(\mathbf{x})}{\partial x_1}, \frac{\partial J(\mathbf{x})}{\partial x_2}, \dots, \frac{\partial J(\mathbf{x})}{\partial x_n} \right]$$



Εικόνα 10 Αναπαράσταση εύρεσης ελαχίστου με την μέθοδο κατάβασης δυναμικού

Δηλαδή, ακολουθώντας την αρνητική κλίση της επιφάνειας σφάλματος, κατευθυνόμαστε γρηγορότερα προς το ελάχιστο της. Σε κάθε βήμα του αλγορίθμου, το επόμενο σημείο προς την κατεύθυνση του ελαχίστου υπολογίζεται ως εξής:

$$\alpha_{n+1} = \alpha_n - \eta \nabla J \alpha_n ,$$

Με γ αρκετά μικρό έτσι ώστε: $J \alpha_n \geq J \alpha_{n+1}$.

Στην περίπτωση των νευρωνικών οι μεταβλητές της συνάρτησης σφάλματος J αποτελούν τα βάρη του δικτύου και επομένως η κλίση της θα είναι:

$$\nabla_{\mathbf{W}} J = \frac{\partial J}{\partial \mathbf{W}} = \left(\frac{\partial J(\mathbf{W})}{\partial w_1}, \frac{\partial J(\mathbf{W})}{\partial w_2}, \dots, \frac{\partial J(\mathbf{W})}{\partial w_n} \right)$$

Παρ' όλα αυτά η μέθοδος αυτή δεν μπορεί να εφαρμοστεί αυτούσια σε δίκτυα τα οποία έχουν κρυφά επίπεδα, καθώς δεν είναι γνωστή η επιθυμητή έξοδος σε κάθε νευρώνα των εσωτερικών επιπέδων του δικτύου, παρά μόνο στο επίπεδο εξόδου. Η λύση για το πρόβλημα αυτό, δίνεται από τον αλγόριθμο *ανάστροφης μετάδοσης σφάλματος* (*back propagation*), τον πιο γνωστό αλγόριθμο εκπαίδευσης τεχνητών νευρωνικών δικτύων και ο οποίος βασίζεται στην μέθοδο κατάβασης δυναμικού, επιτρέποντας όμως να καθοριστεί το ποσοστό του συνολικού σφάλματος που αντιστοιχεί στα βάρη κάθε νευρώνα, ακόμα και αν αυτός βρίσκεται σε κρυφό επίπεδο. Παρακάτω διατυπώνεται μαθηματικά και περιγράφεται ο αλγόριθμος ανάστροφης μετάδοσης σφάλματος.

Αλγόριθμος Ανάστροφης Μετάδοσης Σφάλματος

Δίνεται ένα νευρωνικό δίκτυο perceptron πολλών επιπέδων, με αριθμό επιπέδων K και ένα σύνολο δεδομένων εισόδου και επιθυμητής εξόδου (σύνολο δεδομένων εκπαίδευσης) πάνω στο οποίο το νευρωνικό δίκτυο θα εκπαιδευτεί:

$$(\mathbf{x}^1, \mathbf{d}^1), (\mathbf{x}^2, \mathbf{d}^2), \dots, (\mathbf{x}^P, \mathbf{d}^P)$$

$$\text{όπου, } \mathbf{x}^{(i)} = [x_1^i, x_2^i, \dots, x_n^i], \quad \mathbf{d}^{(i)} = [d_1^i, d_2^i, \dots, d_m^i]$$

Χρησιμοποιώντας ως συνάρτηση σφάλματος, την συνάρτηση summed squared error (SSE) μεταξύ εξόδου του δικτύου και αναμενόμενης εξόδου, η αντικειμενική συνάρτηση που προκύπτει είναι:

$$J_{tot} = \frac{1}{2} \sum_{i=1}^P \|\mathbf{d}^i - \mathbf{y}^i\|^2 = \frac{1}{2} \sum_{i=1}^P \sum_{j=1}^m (d_j^i - y_j^i)^2 \quad (1),$$

όπου, $\mathbf{y} = [y_1, y_2, \dots, y_m] = [a_1^{(K)}, a_2^{(K)}, \dots, a_m^{(K)}] = \mathbf{a}^K$, η έξοδος του δικτύου που αποτελείται από τις ενεργοποιήσεις των νευρώνων του επιπέδου εξόδου.

Για τον καθορισμό της συνεισφοράς του επιπέδου εξόδου στο συνολικό σφάλμα:

Υπολογίζουμε μεταβολή του συνολικού σφάλματος ως προς τις τιμές των βαρών των ενώσεων μεταξύ του επιπέδου εξόδου και του προηγούμενου επιπέδου:

$$\nabla J_{tot} = \frac{\partial J_{tot}}{\partial \mathbf{W}^{(K)}} = \frac{\partial}{\partial \mathbf{W}^{(K)}} \left(\frac{1}{2} \sum_{z=1}^p \| \mathbf{d}^z - \mathbf{y}^z \|^2 \right) = \frac{1}{2} \sum_{z=1}^p \frac{\partial (\mathbf{d}^z - \mathbf{y}^z)^2}{\partial \mathbf{W}^{(K)}} \quad (2)$$

όπου $\mathbf{W}^{(K)}$ συμβολίζουμε το διάνυσμα βαρών που χαρακτηρίζουν τις συνδέσεις του επιπέδου $K-1$ με το επίπεδο εξόδου K (επίπεδο εξόδου). Αν επικεντρωθούμε μόνο σε ένα πρότυπο του συνόλου δεδομένων εκπαίδευσης, τότε το σφάλμα εκεί είναι:

$$\frac{\partial J}{\partial \mathbf{W}^{(K)}} = \frac{1}{2} \frac{\partial \|\mathbf{d} - \mathbf{y}\|^2}{\partial \mathbf{W}^{(K)}} = -(\mathbf{d} - \mathbf{y}) \frac{\partial \mathbf{y}}{\partial \mathbf{W}^{(K)}} = -(\mathbf{d} - \mathbf{a}^K) \frac{\partial \mathbf{a}^K}{\partial \mathbf{W}^{(K)}}$$

Η ενεργοποίηση όμως ενός νευρώνα i στο επίπεδο K είναι:

$$a_i^K = f(u_i^K),$$

$$u_i^K = \sum_{v=1}^m w_{iv}^K a_v^{K-1} + b_i^{(K)}$$

Όπου $w_{ij}^{(K)}$ το βάρος σύνδεσης του νευρώνα j του στρώματος $K-1$ με το νευρώνα i του στρώματος K . Έτσι χρησιμοποιώντας τον κανόνα της αλυσίδας προκύπτει:

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{W}^{(K)}} &= (\mathbf{d} - \mathbf{a}^K) \frac{\partial \mathbf{a}^K}{\partial \mathbf{W}^{(K)}} = (\mathbf{d} - \mathbf{a}^K) \frac{\partial \mathbf{a}^K}{\partial \mathbf{u}^K} \frac{\partial \mathbf{u}^K}{\partial \mathbf{W}^{(K)}} = (\mathbf{d} - \mathbf{a}^K) \frac{\partial f(\mathbf{u}^K)}{\partial \mathbf{u}^K} \frac{\partial \mathbf{u}^K}{\partial \mathbf{W}^{(K)}} \\ &= (\mathbf{d} - \mathbf{a}^K) f'(\mathbf{u}^{(K)}) \mathbf{a}^{K-1} \end{aligned}$$

Και τελικά,

$$\boxed{\frac{\partial J}{\partial \mathbf{W}^{(K)}} = \boldsymbol{\delta}^K \mathbf{a}^{K-1}, \quad \mu\epsilon \quad \boldsymbol{\delta}^K = (\mathbf{d} - \mathbf{a}^K) f'(\mathbf{u}_i^K)} \quad (3)$$

Όπου $\boldsymbol{\delta}^K = (\mathbf{d} - \mathbf{a}^K) f'(\mathbf{u}_i^K)$ ονομάζεται παράγοντας δ και ουσιαστικά αποτελεί τον ρυθμό μεταβολής του σφάλματος ως προς την είσοδο των νευρώνων του επιπέδου K . Από την (1),

αθροίζοντας τα $\frac{\partial J}{\partial \mathbf{W}^K}$ για κάθε πρότυπο του συνόλου δεδομένων εκπαίδευσης, παίρνουμε την συνεισφορά του επιπέδου εξόδου στο συνολικό σφάλμα.

Για τον καθορισμό της συνεισφοράς των κρυφών επιπέδων στο συνολικό σφάλμα:

Όπως και προηγουμένως υπολογίζουμε μεταβολή του συνολικού σφάλματος ως προς τις τιμές των βαρών, τα οποία όμως τώρα αφορούν τις ενώσεις μεταξύ των κρυφών επιπέδων. Έτσι με παρόμοιο τρόπο όπως και πριν και με χρήση του κανόνα της αλυσίδας, για τα βάρη \mathbf{W}^{K-1} μεταξύ των επιπέδων $K-1$ και $K-2$ έχουμε:

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{W}^{K-1}} &= (\mathbf{d} - \mathbf{a}^K) \frac{\partial \mathbf{a}^K}{\partial \mathbf{W}^{K-1}} = (\mathbf{d} - \mathbf{a}^K) \frac{\partial \mathbf{a}^K}{\partial \mathbf{u}^K} \frac{\partial \mathbf{u}^K}{\partial \mathbf{W}^{K-1}} = \\ &= (\mathbf{d} - \mathbf{a}^K) f'(\mathbf{u}^K) \frac{\partial \mathbf{u}^K}{\partial \mathbf{W}^{K-1}} = \boldsymbol{\delta}^K \frac{\partial \mathbf{u}^K}{\partial \mathbf{a}^{K-1}} \frac{\partial \mathbf{a}^{K-1}}{\partial \mathbf{W}^{K-1}} = \boldsymbol{\delta}^K \frac{\partial \mathbf{u}^K}{\partial \mathbf{a}^{K-1}} \frac{\partial \mathbf{a}^{K-1}}{\partial \mathbf{u}^{K-1}} \frac{\partial \mathbf{u}^{K-1}}{\partial \mathbf{W}^{K-1}} \\ &= \boldsymbol{\delta}^K \mathbf{W}^K f'(\mathbf{u}^{(K-1)}) \frac{\partial \mathbf{u}^{K-1}}{\partial \mathbf{W}^{K-1}} = \boldsymbol{\delta}^K \mathbf{W}^K f'(\mathbf{u}^{(K-1)}) \mathbf{a}^{K-2} \end{aligned}$$

Και τελικά,

$$\frac{\partial J}{\partial \mathbf{W}^{K-1}} = \boldsymbol{\delta}^{K-1} \mathbf{a}^{K-2}, \quad \mu\epsilon \quad \boldsymbol{\delta}^{K-1} = \boldsymbol{\delta}^K \mathbf{W}^K f'(\mathbf{u}^{(K-1)}) \quad (4)$$

Η παραπάνω σχέση αφορά τον καθορισμό της συνεισφοράς των κρυφών επιπέδων στο σφάλμα του δικτύου και επομένως εφαρμόζεται ανάλογα και για τα βάρη \mathbf{W}^{K-2} , \mathbf{W}^{K-3} κ.ο.κ.

Από τις σχέσεις (3), (4) πλέον γνωρίζουμε τον τρόπο με τον οποίο μπορεί να υπολογιστεί η μεταβολή του σφάλματος σε σχέση με κάθε βάρος του νευρωνικού δικτύου και επομένως με χρήση της μεθόδου κατάβασης δυναμικού μπορούμε να μεταβάλλουμε τα βάρη με τέτοιον τρόπο ώστε η συνάρτηση σφάλματος να ελαχιστοποιείται. Έτσι, η μεταβολή των βαρών προς την κατεύθυνση αρνητικής κλίσης είναι:

$$w_{ij}^{l, k+1} = w_{ij}^{l, k} - \gamma \frac{\partial J}{\partial w_{ij}^{l, k}} \quad (5)$$

Όπου $w_{ij}^l(k)$ είναι το βάρος σύνδεσης του νευρώνα j του επιπέδου $l-1$ με τον νευρώνα i του επιπέδου l στην χρονική στιγμή (επανάληψη) k και γ ένας αριθμός αρκετά μικρός που ονομάζεται ρυθμός μάθησης.

Με βάση τα παραπάνω ο αλγόριθμος ανάστροφης μετάδοσης σφάλματος διατυπώνεται ως εξής:

1. Αρχικοποίηση του δικτύου με τυχαία βάρη
2. Για κάθε εποχή εκπαίδευσης επανέλαβε:
 - 2.1. Για κάθε πρότυπο εκπαίδευσης από 1 έως P επανέλαβε:
 - 2.1.1. Υπολόγισε τις εξόδους-ενεργοποιήσεις κάθε επιπέδου του δικτύου
 - 2.1.2. Υπολόγισε τους παράγοντες δ για τους m νευρώνες του επιπέδου εξόδου (K) με βάση την (3)
 - 2.1.3. Ανάστροφα, για κάθε κρυφό επίπεδο του δικτύου από $K-1$ ως 1 επανέλαβε:
 - 2.1.3.1. Υπολόγισε τους παράγοντες δέλτα για όλους τους νευρώνες του κρυφού επιπέδου, με βάση την (4).
 - 2.1.4. Ανανέωσε όλα τα βάρη του δικτύου με βάση την (5)
 - 2.2. Υπολόγισε το σφάλμα εξόδου
 - 2.3. Αν το σφάλμα είναι μικρότερο από ϵ επέστρεψε τα βάρη του δικτύου.
 - 2.4. Αν η εποχή είναι η τελευταία επέστρεψε το σφάλμα.

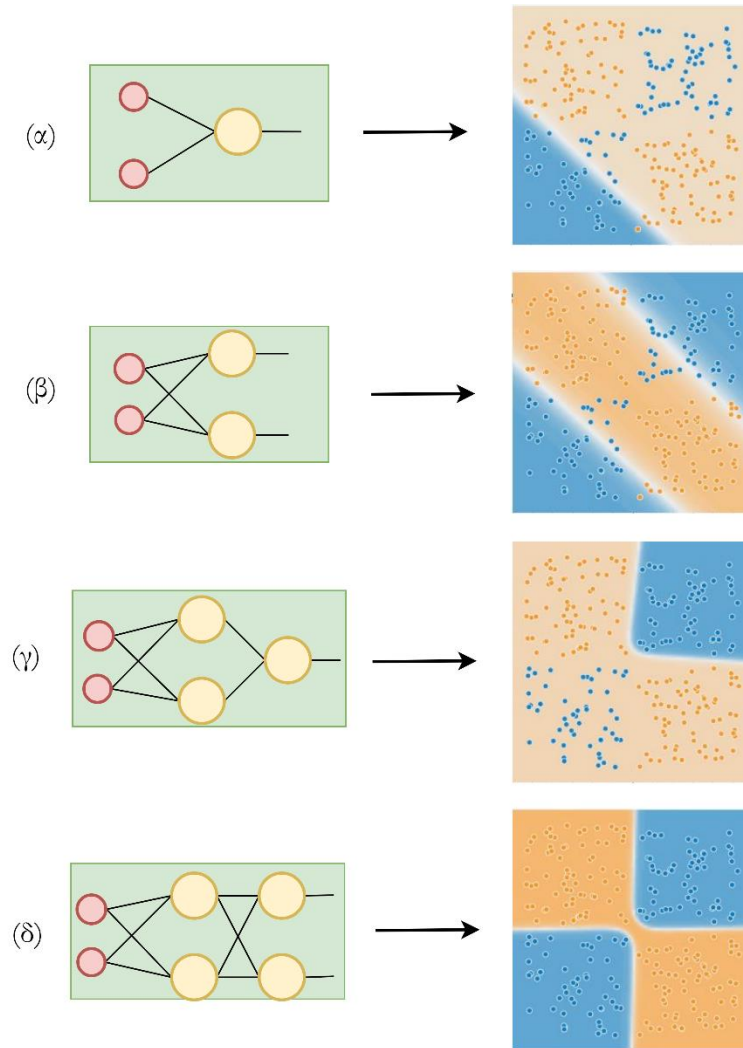
Η προσπάθεια του αλγορίθμου ανάστροφης μετάδοσης σφάλματος για ελαχιστοποίηση, μπορεί να θεωρηθεί σαν μια αναζήτηση του ολικού ελαχίστου της συνάρτησης σφάλματος, η οποία έχει σαν παραμέτρους τις τιμές των βαρών. Η διόρθωση που γίνεται κάθε φορά προσπαθεί να ελαχιστοποιήσει το σφάλμα διαλέγοντας να κάνει εκείνες τις αλλαγές που φαίνεται να το μειώνουν τοπικά.

3.3.6 ΤΝΔ Πρόσθιας Τροφοδότησης Πολλών Επιπέδων ως καθολικοί προσεγγιστές

Σύμφωνα με το *Θεώρημα Καθολικής Προσέγγισης* που διατυπώνεται στην μαθηματική θεωρία των τεχνητών νευρωνικών δικτύων, ένα πολυεπίπεδο δίκτυο perceptron (MLP) με δύο επίπεδα (ένα κρυφό επίπεδο και το επίπεδο εξόδου) με πεπερασμένο αριθμό νευρώνων, είναι ικανό να προσεγγίσει οποιαδήποτε συνεχή συνάρτηση με οσοδήποτε μικρό σφάλμα, διατηρώντας μονάχα, ήπιους περιορισμούς σχετικά με τις συναρτήσεις ενεργοποίησης που χρησιμοποιεί οι οποίες θα πρέπει να είναι φραγμένες, μονότονα αύξουσες και συνεχείς π.χ. η σιγμοειδής συνάρτηση.

Στα πλαίσια του θεωρήματος αυτού, εξετάζουμε μια τυχαία κατανομή παραδειγμάτων, καθένα από τα οποία έχει πρότυπο δύο διαστάσεων (x_1, x_2) και ανήκει σε μία από τις δύο μη γραμμικά διαχωρίσιμες κλάσεις. Στην Εικόνα 11 αναπαριστούμε στο επίπεδο τα αποτελέσματα

ταξινόμησης τεσσάρων διαφορετικών ΤΝΔ που έχουν εκπαιδευτεί πάνω σε αυτό το σύνολο δεδομένων, σχεδιάζοντας τις διαχωριστικές επιφάνειες που δημιουργεί το καθένα.



Εικόνα 11 Απεικόνιση διαχωριστικών επιφανειών που δημιουργούν πολυεπίπεδα δίκτυα perceptron με διαφορετικό πλήθος νευρώνων και κρυφών επιπέδων

3.4 Αναδρομικά Τεχνητά Νευρωνικά Δίκτυα

Τα αναδρομικά νευρωνικά δίκτυα (RNN) είναι δημοφιλή μοντέλα που έχουν δείξει μεγάλη αποτελεσματικότητα κυρίως σε προβλήματα επεξεργασίας φυσικής γλώσσας (NLP). Η ιδέα πίσω από τα αυτά είναι ότι κάνουν χρήση διαδοχικών πληροφοριών. Σε ένα παραδοσιακό νευρωνικό δίκτυο υποθέτουμε ότι όλες οι εισοδοί (και οι έξοδοι) είναι ανεξάρτητες μεταξύ τους. Αλλά για αρκετά προβλήματα αυτό αποτελεί μια πολύ κακή προσέγγιση καθώς πολλές φορές θέλουμε να προβλέψουμε ή να κατηγοριοποιήσουμε οντότητες βασιζόμενοι στην αλληλουχία των εισόδων.

Τα αναδρομικά νευρωνικά δίκτυα αποτελούν μια άλλη κατηγορία τεχνητών νευρωνικών δικτύων, των οποίων το βασικό τους σώμα αποτελείται από ένα πολυεπίπεδο δίκτυο perceptron (MLP). Ωστόσο, το θεμελιώδες χαρακτηριστικό τους που τα διαφοροποιεί από τα ΤΝΔ πρόσθιας τροφοδότησης είναι ότι περιέχουν τουλάχιστον μια σύνδεση ανάδρασης μεταξύ των νευρώνων ίδιων ή προηγούμενων επιπέδων, με αποτέλεσμα οι ενεργοποιήσεις των νευρώνων αυτών να μπορούν να ρέουν κυκλικά προς τα πίσω. Έτσι καθώς πλέον η ενεργοποίηση των νευρώνων ενός επιπέδου μπορεί να συσχετιστεί με μια προηγούμενη ενεργοποίηση του, εισάγεται η έννοια του χρόνου, όπου μια παρελθοντική στιγμή $t-1$ μπορεί να συσχετιστεί και να συνδυαστεί αντίστοιχα με την χρονική στιγμή t . Αυτό σημαίνει ότι τα αναδρομικά δίκτυα, σε αντίθεση με τα ΤΝΔ πρόσθιας τροφοδότησης, μπορούν να υποστηρίξουν στην εσωτερική δομή τους βραχυπρόθεσμη μνήμη συσχετίζοντας προηγούμενες εισόδους με μελλοντικές, ένα χαρακτηριστικό που τα κάνει πολύ πιο ενδιαφέροντα ως μοντέλα του ανθρώπινου εγκεφάλου, αλλά ταυτόχρονα και πιο δυσνόητα και σύνθετα.

3.4.1 Αρχιτεκτονικές Αναδρομικών Δικτύων

Τα αναδρομικά δίκτυα έχουν ένα πολύ πλούσιο ρεπερτόριο αρχιτεκτονικών διατάξεων καθώς υπάρχει μια μεγάλη ποικιλία επιλογών σχεδιασμού σχετικά με την πυκνότητα των συνδέσεων ανάδρασης—δηλαδή σχετικά με το αν μια ανάδραση μεταβιβάζεται σε έναν ή περισσότερους νευρώνες—, αν οι αναδράσεις θα μεταβιβάζονται στο ίδιο ή σε προηγούμενο επίπεδο κτλ.

Στα πλαίσια αυτής της διπλωματικής, θα παρουσιάσουμε μόνο τις βασικές αρχιτεκτονικές που θα μας απασχολήσουν για την ανάπτυξη των μοντέλων μας. Αρχικά, σε όλες τις αρχιτεκτονικές που θα παρουσιαστούν, θα θεωρήσουμε πλήρεις συνδέσεις στις αναδράσεις, δηλαδή η ανάδραση που ξεκινάει από ένα νευρώνα και μεταβιβάζεται σε προηγούμενο επίπεδο, τροφοδοτείται σε όλους του νευρώνες αυτού του επιπέδου. Στην συνέχεια κατηγοριοποιούμε τις

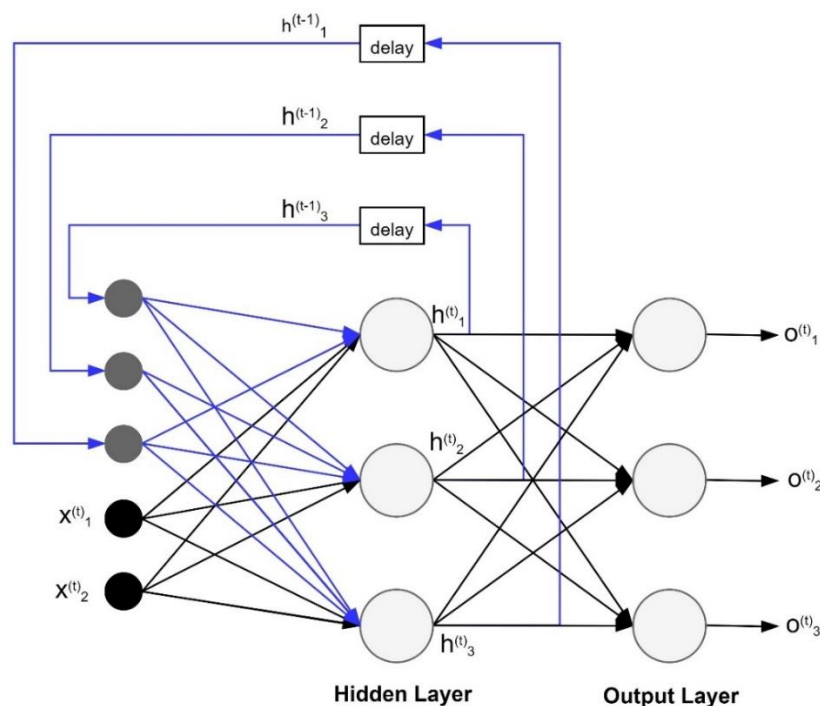
αρχιτεκτονικές αυτές με βάση τον αριθμό των κρυφών επιπέδων και με βάση τον αριθμό των εξόδων.

Με βάση τον αριθμό των κρυφών επιπέδων

Όπως είχε αναφερθεί, το κύριο σώμα των αναδρομικών νευρικών δικτύων αποτελείται από ένα πολυεπίπεδο perceptron (MLP), μέσα στο οποίο περιλαμβάνονται οι αναδράσεις του δικτύου. Όπως κάθε MLP έτσι και ένα αναδρομικό νευρωνικό δίκτυο μπορεί να αποτελείται από ένα ή περισσότερα κρυφά επίπεδα. Με βάση το κριτήριο αυτό έχουμε τις εξής κατηγορίες δικτύων.

- **Αναδρομικά Δίκτυα Ενός (Κρυφού) Επιπέδου**

Το σώμα αυτών των αναδρομικών δικτύων αποτελείται από ένα κρυφό επίπεδο νευρώνων και από ένα επίπεδο εξόδου. Οι αναδράσεις ξεκινάνε μόνο από τους νευρώνες του κρυφού επιπέδου και καταλήγουν πάλι σε αυτούς. Όπως οι πρόσθιες συνδέσεις μεταξύ των νευρώνων των διάφορων επιπέδων του νευρωνικού δικτύου χαρακτηρίζονται από συναπτικά βάρη, έτσι και οι συνδέσεις των αναδράσεων αποτελούνται τα δικά τους βάρη. Στην παρακάτω εικόνα φαίνεται μια λεπτομερής αναπαράσταση των συνδέσεων σε ένα αναδρομικό δίκτυο ενός κρυφού επιπέδου.



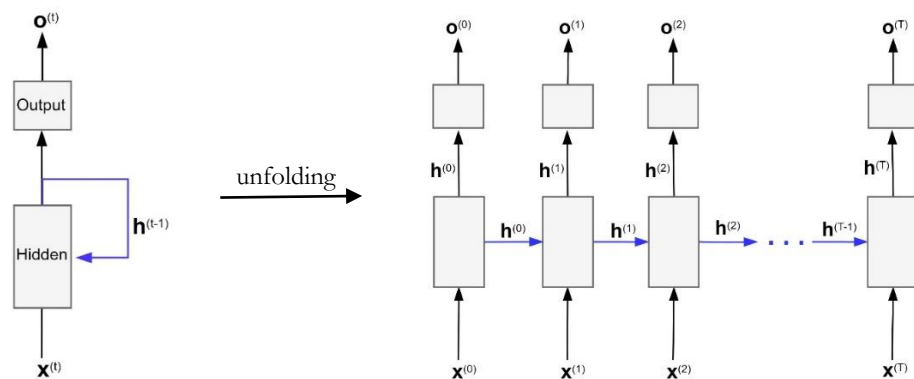
Εικόνα 12 Πλήρης αναπαράσταση συνδέσεων αναδρομικού δικτύου

Στα αναδρομικά δίκτυα εισάγεται η έννοια του διακριτού χρόνου όπου η ενεργοποίηση των νευρώνων ενός επιπέδου (χρονική στιγμή t) μπορεί να συσχετιστεί με μια προηγούμενη ενεργοποίηση του (χρονική στιγμή $t-1$). Για τον λόγο αυτό, στις συνδέσεις των αναδράσεων εισάγεται μια μονάδα καθυστέρησης (delay) ο ρόλος της οποία είναι να συγκρατεί προσωρινά τις ενεργοποιήσεις των νευρώνων που συμμετέχουν στην ανάδραση, προκειμένου αυτές να τροφοδοτηθούν και να επεξεργαστούν σε επόμενο βήμα.

Η σημειογραφία που θα χρησιμοποιηθεί, όπως φαίνεται και στην Εικόνα 11, για την περιγραφή της λειτουργίας ενός αναδρομικού δικτύου είναι η εξής:

- $\mathbf{X}^t = [x_1^t, x_2^t, \dots, x_I^t]$ είναι το διάνυσμα των εισόδων του δικτύου την χρονική στιγμή t .
- $\mathbf{H}^t = [h_1^t, h_2^t, \dots, h_H^t]$ είναι το διάνυσμα που περιέχει τις ενεργοποιήσεις των νευρώνων του κρυφού επιπέδου την χρονική στιγμή t . Αποτελεί την «μνήμη» του αναδρομικού δικτύου και υπολογίζεται συνδυάζοντας την τρέχουσα είσοδο με την ενεργοποίηση του κρυφού επιπέδου του προηγούμενου χρονικού βήματος.
- $\mathbf{O}^t = [o_1^t, o_2^t, \dots, o_H^t]$ είναι το διάνυσμα που περιέχει τις εξόδους του δικτύου την χρονική στιγμή t .

Λόγω της περίπλοκης δομής που ένα αναδρομικό δίκτυο μπορεί να εμφανίσει λόγω της παρουσίας συνδέσεων ανάδρασης, συνηθίζεται για την γραφική απεικόνισή του να χρησιμοποιείται μια απλοποιημένη ισοδύναμη αναπαράσταση. Η απλοποιημένη αναπαράσταση για το αναδρομικό δίκτυο με ένα κρυφό επίπεδο είναι η εξής:



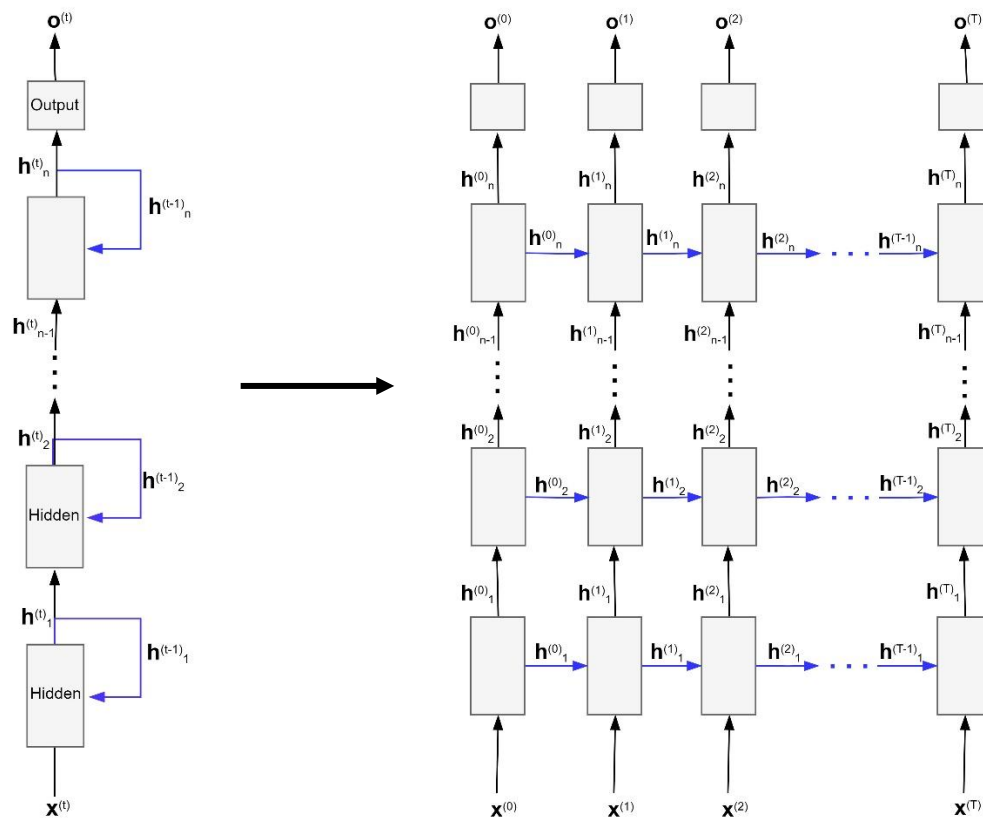
Εικόνα 13 Απλοποιημένη αναπαράσταση αναδρομικού δικτύου και unfolding του δικτύου

Για όλα τα αναδρομικά δίκτυα που θα παρουσιαστούν στην συνέχεια θα χρησιμοποιηθεί η απλοποιημένη αναπαράστασή τους.

Τέλος, πολύ συχνή και κοινή πρακτική είναι να αναπαρίσταται η εξέλιξη του αναδρομικού δικτύου μέσα στον χρόνο η οποία είναι γνωστή ως «ξεδιπλώμα» (unfolding) του δικτύου κατά μήκος της ακολουθίας των εισόδων. Μέσω του ξεδιπλώματος πετυχαίνουμε να έχουμε καλύτερη εικόνα της δυναμικής λειτουργίας ενός αναδρομικού δικτύου και των εισόδων και εξόδων του κάθε χρονική στιγμή.

- **Αναδρομικά Δίκτυα Πολλών (Κρυφών) Επιπέδων**

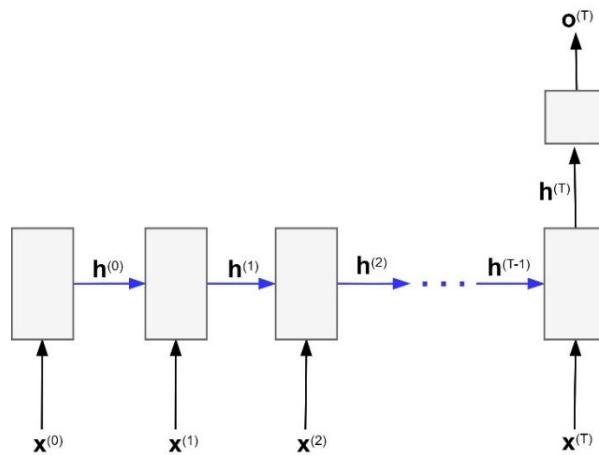
Τα αναδρομικά νευρωνικά δίκτυα πολλών επιπέδων είναι παρόμοια με αυτά του ενός επιπέδου ως προς τις αναδράσεις τους και την εξέλιξή τους στον χρόνο, με την διαφορά ότι σε αυτά τώρα στοιβάζονται πολλά κρυφά επίπεδα.



Εικόνα 14 Πολυεπίπεδο αναδρομικό δίκτυο

Με βάση τον αριθμό των εξόδων

Ανάλογα με το πρόβλημα που καλείται να επιλύσει κάθε φορά ένα αναδρομικό νευρωνικό δίκτυο αντίστοιχα προσαρμόζεται και ο αριθμός των εξόδων του. Για παράδειγμα σε ένα πρόβλημα ταξινόμησης συναισθήματος μιας πρότασης, οι λέξεις τροφοδοτούνται διαδοχικά ως εισοδοι στο δίκτυο (διαδοχικές χρονικές στιγμές) και αφού έχουν δοθεί όλες οι λέξεις, στο τελευταίο βήμα παράγεται η έξοδος. Οπότε το αναδρομικό δίκτυο περιορίζεται σε μία μόνο έξοδο στο τελευταίο βήμα (Εικόνα 15). Αντίθετα σε ένα πρόβλημα μετάφρασης ενός κειμένου, όπου για κάθε λέξη που εισάγεται, το δίκτυο θα πρέπει να παράγει και την αντίστοιχη μεταφρασμένη λέξη, σε κάθε βήμα-εισοδο του δικτύου υπάρχει και μια έξοδος (Εικόνα 13).



Εικόνα 15 Αναδρομικό δίκτυο με μία έξοδο στο τελευταίο επίπεδο

3.4.2 Ανάκληση Αναδρομικού ΤΝΔ

Για την περιγραφή της ανάκληση των αναδρομικών νευρωνικών δικτύων, θα χρησιμοποιήσουμε ένα αναδρομικό δίκτυο ενός επιπέδου, ωστόσο η περιγραφή αυτή μπορεί να γενικευθεί κατάλληλα και για τις υπόλοιπες αρχιτεκτονικές που αναφέρθηκαν παραπάνω.

Η ανάκληση ενός αναδρομικού νευρωνικού δικτύου είναι η ίδια με αυτή ενός perceptron πολλαπλών επιπέδων με ένα μόνο κρυφό στρώμα, με την διαφορά ότι οι ενεργοποιήσεις φθάνουν στο κρυφό στρώμα τόσο από την τρέχουσα εξωτερική είσοδο όσο και από τις ενεργοποιήσεις του κρυφού επιπέδου από το προηγούμενο χρονικό διάστημα. Θεωρούμε μια ακολουθία εισόδων μήκους T , $\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2 \dots, \mathbf{x}^T]$ σε αναδρομικό δίκτυο με I εισόδους, H πλήθος νευρώνων κρυφού επιπέδου και K νευρώνες εξόδου. Αν x_i^t είναι η είσοδος i δικτύου την χρονική στιγμή t ,

y_j^t η είσοδος του νευρώνα j την χρονική στιγμή t και h_j^t, o_j^t αντίστοιχα οι ενεργοποιήσεις για τους νευρώνες j του κρυφού και του επιπέδου εξόδου την χρονική στιγμή t , η ανάκληση του δικτύου δίνεται ως εξής:

$$y_h^t = \sum_{i=1}^I w_{ih} x_i^t + \sum_{h'=1}^H w_{h'h} h_{h'}^{t-1} \quad (1)$$

$$h_h^t = f(y_h^t) \quad (2)$$

$$y_o^t = \sum_{h'=1}^H w_{h'o} h_{h'}^t$$

$$o_k^t = g(y_o^t)$$

όπου f, g μη γραμμικές συναρτήσεις ενεργοποίησης των νευρώνων του κρυφού και του επιπέδου εξόδου.

Η πλήρης ακολουθία των ενεργοποιήσεων των νευρώνων του κρυφού επιπέδου μπορεί να υπολογιστεί ξεκινώντας για $t=1$ και εφαρμόζοντας αναδρομικά τις (1) και (2) καθώς αυξάνουμε το t σε κάθε βήμα. Σημειώνεται, ότι για να μπορεί να επιτευχθεί αυτό, απαιτείται η επιλογή των αρχικών τιμών $h_{h'}^0$ σχετικά με την κατάσταση του αναδρομικού δικτύου πριν αυτό λάβει οποιαδήποτε είσοδο. Οι αρχικές τιμές αυτές μπορούν να πάρουν τόσο μηδενικές όσο και μη μηδενικές τιμές.

Η ανάκληση του αναδρομικού νευρωνικού δικτύου σε μορφή μητρώων είναι η εξής :

$$\mathbf{H}^t = f(\mathbf{W}_{IH} \mathbf{X}^t + \mathbf{W}_{H'H} \mathbf{H}^{t-1}) \quad (3)$$

$$\mathbf{O}^{(t)} = g(\mathbf{W}_{HO} \mathbf{H}^t)$$

\mathbf{W}_{IH} : διάνυσμα βαρών μεταξύ εισόδου και κρυφού επιπέδου.

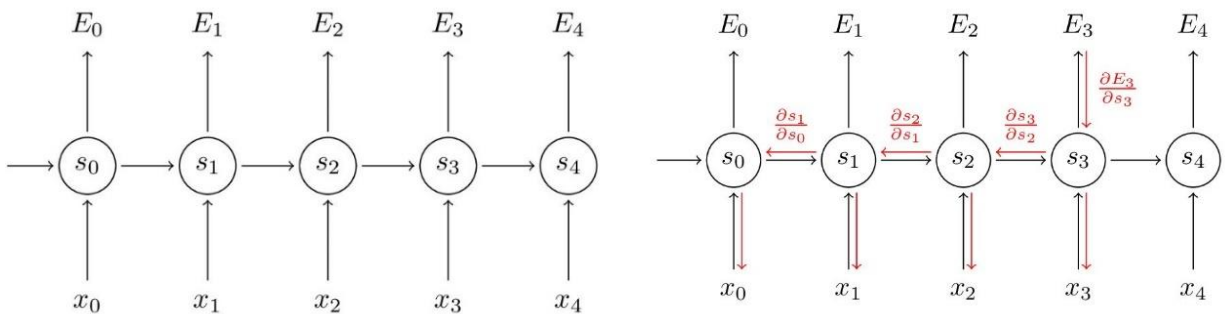
$\mathbf{W}_{H'H}$: διάνυσμα βαρών μεταξύ κρυφού επιπέδου μεταξύ χρονικών στιγμών t και $t-1$.

\mathbf{W}_{HO} : διάνυσμα βαρών μεταξύ κρυφού επιπέδου και επιπέδου εξόδου.

3.4.3 Εκπαίδευση Αναδρομικού Δικτύου με τον αλγόριθμο οπισθοδιάδοσης σφάλματος μέσα στον χρόνο (Back-Propagation Through Time)

Παρόμοια με ένα νευρωνικό δίκτυο πρόσθιας τροφοδότησης, ένα αναδρομικό νευρωνικό δίκτυο προκειμένου να εκπαιδευτεί, απαιτεί έναν κατάλληλο αλγόριθμο εκπαίδευσης. Ένας από τους πιο διαδεδομένους αλγορίθμους εκπαίδευσης αναδρομικών δικτύων είναι ο αλγόριθμος οπισθοδιάδοσης σφάλματος μέσα στον χρόνο (Back-Propagation Through Time ή BPTT), ο οποίος αποτελεί προέκταση του απλού αλγορίθμου οπισθοδιάδοσης σφάλματος (Back-Propagation ή BP). Όπως και στον απλό BP έτσι και στον BPTT υπολογίζονται κατάλληλα οι παράγωγοι της συνάρτησης σφάλματος ως προς τις παραμέτρους του δικτύου εφαρμόζοντας επανειλημμένα τον κανόνα της αλυσίδας με σκοπό τελικά να ανανεωθούν τα βάρη του δικτύου σύμφωνα με την βελτιστοποίηση της κατάβασης δυναμικού. Η διαφορά έγκειται στο γεγονός ότι στα αναδρομικά δίκτυα οι ενεργοποιήσεις των κρυφών επιπέδων επηρεάζονται χρονικά και από παρελθοντικές ενεργοποιήσεις τους. Αυτό δημιουργεί την ανάγκη η μεταβολή του σφάλματος να υπολογιστεί σε σχέση και με προηγούμενες χρονικές στιγμές.

Διαισθητικά ο BPTT δρα πάνω στο ξεδιπλωμένο αναδρομικό δίκτυο το οποίο μπορεί να προσεγγιστεί σαν ένα βαθύ δίκτυο πρόσθιας τροφοδότησης, όπου κάθε επίπεδο του είναι και μια διαφορετική χρονική στιγμή και αποτελείται από ένα χρονικό βήμα εισόδου, ένα αντίγραφο του αρχικού δικτύου και ένα χρονικό βήμα εξόδου. Τα σφάλματα υπολογίζονται για κάθε χρονικό βήμα, αθροίζονται και μέσω του αλγορίθμου υπολογίζονται κατάλληλα οι μεταβολές λαμβάνοντας υπ' όψιν της εξαρτήσεις που υπάρχουν κατά μήκος όλη της χρονικής ακολουθίας, προκειμένου να ανανεωθούν ανάλογα τα βάρη.



Εικόνα 16 Ανάστροφη μετάδοση σφάλματος σε αναδρομικό δίκτυο

Για αναδρομικό δίκτυο ενός επιπέδου με μήκος ακολουθίας T :

Αν J_t είναι το σφάλμα σε κάθε χρονικό βήμα t , τότε το συνολικό σφάλμα που θέλουμε να ελαχιστοποιήσουμε είναι:

$$J = \sum_t J_t(\mathbf{o}_t, \mathbf{d}_t) \quad (1)$$

Μεταβολή του σφάλματος ως προς τα βάρη W_{HO} :

Από την (1) και τις σχέσεις (3) του προηγούμενου κεφαλαίου έχουμε:

$$\frac{\partial J}{\partial W_{HO}} = \sum_t \frac{\partial J_t}{\partial W_{HO}} = \sum_t \frac{\partial J_t}{\partial \mathbf{o}} \frac{\partial \mathbf{o}}{\partial W_{HO}} = \sum_t \frac{\partial J_t}{\partial \mathbf{o}} \frac{\partial \mathbf{o}}{\partial \mathbf{y}_o} \frac{\partial \mathbf{y}_o}{\partial W_{HO}}$$

Μεταβολή του σφάλματος ως προς τα βάρη $W_{H'H}$:

Για το σφάλμα μια συγκεκριμένης χρονικής στιγμής t έχουμε:

$$\frac{\partial J_t}{\partial W_{H'H}} = \frac{\partial J_t}{\partial \mathbf{o}_t} \frac{\partial \mathbf{o}_t}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial W_{H'H}}$$

Η ενεργοποίηση του κρυφού \mathbf{h}_t εν μέρη εξαρτάται από τις ενεργοποιήσεις του ίδιου επιπέδου για τις προηγούμενες χρονικές στιγμές $t-1, t-2, \dots, t-k$. Επιπλέον, καθώς τα βάρη $W_{H'H}$ μοιράζονται σε όλα τα χρονικά βήματα, ο κανόνας της αλυσίδας μπορεί να εφαρμοστεί κατά μήκος όλης της ακολουθίας του ξεδιπλωμένου αναδρομικού δικτύου. Επομένως η μερική παράγωγος ως προς $W_{H'H}$ μπορεί να επεκταθεί ως εξής:

$$\frac{\partial J_t}{\partial W_{H'H}} = \frac{\partial J_t}{\partial \mathbf{o}_t} \frac{\partial \mathbf{o}_t}{\partial \mathbf{h}_t} \left(\frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_{t-1}} \frac{\partial \mathbf{h}_{t-1}}{\partial \mathbf{h}_{t-2}} \dots \frac{\partial \mathbf{h}_{k+1}}{\partial \mathbf{h}_k} \right) \frac{\partial \mathbf{h}_k}{\partial W_{H'H}} = \frac{\partial J_t}{\partial \mathbf{o}_t} \frac{\partial \mathbf{o}_t}{\partial \mathbf{h}_t} \left(\prod_{m=k+1}^t \frac{\partial \mathbf{h}_m}{\partial \mathbf{h}_{m-1}} \right) \frac{\partial \mathbf{h}_k}{\partial W_{H'H}}$$

Έτσι κατά το χρονικό βήμα t μπορούμε να υπολογίσουμε την μεταβολή της συνάρτησης σφάλματος ως προς την έξοδο \mathbf{o}_t και κατ' επέκταση, μέσω της οπισθοδιάδοσης σφάλματος μέσα στον χρόνο από την στιγμή t έως 0, υπολογίζουμε την μεταβολή ως τα βάρη $W_{H'H}$ (Εικόνα 16). Επομένως, αν λάβουμε υπόψη μόνο την έξοδο \mathbf{o}_t στο βήμα t μπορούμε να δώσουμε την ακόλουθη κλίση ως προς $W_{H'H}$:

$$\frac{\partial J_t}{\partial W_{H'H}} = \sum_{k=1}^t \frac{\partial J_t}{\partial \mathbf{o}_t} \frac{\partial \mathbf{o}_t}{\partial \mathbf{h}_t} \left(\prod_{m=k+1}^t \frac{\partial \mathbf{h}_m}{\partial \mathbf{h}_{m-1}} \right) \frac{\partial \mathbf{h}_k}{\partial W_{H'H}}$$

Αθροίζοντας τις μεταβολές του σφάλματος ως προς $W_{H'H}$ για όλα τα χρονικά βήματα μέσω της (1), παίρνουμε:

$$\frac{\partial J}{\partial \mathbf{W}_{H'H}} = \sum_t \sum_{k=1}^t \frac{\partial J_t}{\partial \mathbf{o}_t} \frac{\partial \mathbf{o}_t}{\partial \mathbf{h}_t} \left(\prod_{m=k+1}^t \frac{\partial \mathbf{h}_m}{\partial \mathbf{h}_{m-1}} \right) \frac{\partial \mathbf{h}_k}{\partial \mathbf{W}_{H'H}}$$

Μεταβολή του σφάλματος ως προς τα βάρη \mathbf{W}_{IH} :

Με παρόμοιο τρόπο όπως και πριν μπορούμε να καταλήξουμε η μεταβολή του σφάλματος ως προς \mathbf{W}_{IH} είναι:

$$\frac{\partial J}{\partial \mathbf{W}_{IH}} = \sum_t \sum_{k=1}^t \frac{\partial J_t}{\partial \mathbf{o}_t} \frac{\partial \mathbf{o}_t}{\partial \mathbf{h}_t} \left(\prod_{m=k+1}^t \frac{\partial \mathbf{h}_m}{\partial \mathbf{h}_{m-1}} \right) \frac{\partial \mathbf{h}_k}{\partial \mathbf{W}_{IH}}$$

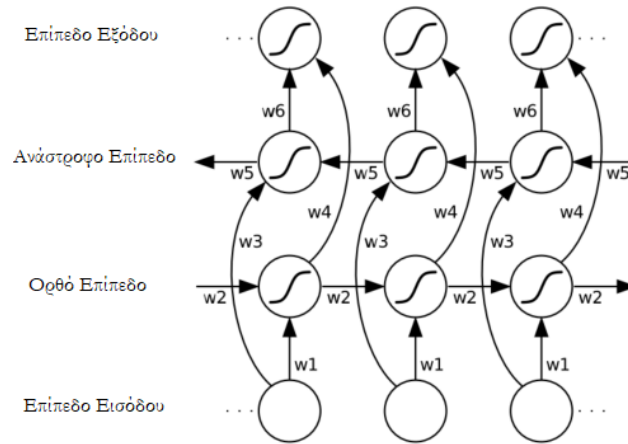
3.4.4 Αμφίδρομα αναδρομικά δίκτυα

Σε πολλά προβλήματα όπου ένα αναδρομικό δίκτυο καλείται να ταξινομήσει ακολουθίες στοιχείων με χρονική εξάρτηση, όπως ένα κείμενο, είναι χρήσιμο να έχουμε πρόσβαση τόσο σε παρελθοντικό όσο και σε μελλοντικό νοηματικό περιεχόμενο. Ωστόσο, δεδομένου ότι οι τυπική λειτουργία των απλών αναδρομικών δικτύων ακολουθεί συγκεκριμένη χρονική σειρά, αγνοείται το μελλοντικό περιεχόμενο. Πιθανές λύσεις σε αυτό το πρόβλημα είναι η χρήση παραθύρου που συμπεριλαμβάνει μελλοντικές εισόδους και τις εισάγει στην τρέχουσα είσοδο του δικτύου ή χρήση καθυστέρησης μεταξύ των εισόδων και των εξόδων-στόχων, παρέχοντας έτσι στο δίκτυο κάποια χρονικά βήματα για να επεξεργαστεί τις μελλοντικές εισόδους. Παρόλα αυτά, αυτές οι μέθοδοι έχουν αποδειχθεί αναποτελεσματικές και καμία δεν επιλύει την ασυμμετρία που μπορεί να παρουσιάζουν οι παρελθοντικές και οι μελλοντικές εισοδοι μεταξύ τους.

Τα αμφίδρομα αναδρομικά νευρωνικά δίκτυα (bidirectional recurrent networks) προσφέρουν μια πιο κομψή λύση. Η βασική ιδέα των αμφίδρομων αναδρομικών δικτύων είναι να παρουσιάσουν κάθε ακολουθία από τα δεδομένα εκπαίδευσης τόσο προς τα εμπρός και όσο προς τα πίσω, κάνοντας χρήση δύο χωριστών αναδρομικών κρυφών στρωμάτων, τα οποία αμφότερα είναι συνδεδεμένα με το ίδιο στρώμα εξόδου. Αυτή η δομή με αυτόν τον τρόπο, παρέχει στο επίπεδο εξόδου ένα πλήρες στιγμιότυπο των παρελθοντικών και των μελλοντικών εισόδων για κάθε χρονικό βήμα στην ακολουθία εισόδου. Ένα ξεδιπλωμένο αμφίδρομο αναδρομικό δίκτυο φαίνεται στην εικόνα 17.

Η ανάκληση ενός αμφίδρομου αναδρομικού δικτύου είναι το ίδιο όπως και για ένα αναδρομικό δίκτυο μονής κατεύθυνσης, με την διαφορά ότι η ακολουθία εισόδου παρουσιάζεται

σε αντίθετες κατευθύνσεις προς τα δύο κρυφά στρώματα και το στρώμα εξόδου δεν ενημερώνεται μέχρι να επεξεργαστούν τα δύο κρυφά στρώματα ολόκληρη την ακολουθία εισόδου. Αρχικά υπολογίζεται η έξοδος h του κρυφού επιπέδου ορθής φοράς, στη συνέχεια η έξοδος h του κρυφού επιπέδου ορθής φοράς και τέλος υπολογίζεται η τελική έξοδος διασχίζοντας τις εξόδους του ορθού επιπέδου από $t = 1$ έως T και του ανάστροφου από $t = T$ έως 1 .



Εικόνα 17 Αμφίδρομο αναδρομικό δίκτυο

Η μαθηματική αναπαράσταση την ανάκλησης ενός αμφίδρομου αναδρομικού δικτύου είναι η εξής:

$$\overrightarrow{h}^{(t)} = f \left(\mathbf{W}_{Ih} \mathbf{X}^{(t)} + \mathbf{W}_{hh} \overrightarrow{h}^{(t-1)} + \mathbf{b}_h \right)$$

$$\overleftarrow{h}^t = f \left(\mathbf{W}_{Ih} \mathbf{X}^t + \mathbf{W}_{hh} \overleftarrow{h}^{t-1} + \mathbf{b}_h \right)$$

$$\mathbf{o}^{(t)} = g \left(\mathbf{W}_{hO} \overrightarrow{h}^{(t)} + \mathbf{W}_{hO} \overleftarrow{h}^{(t)} + \mathbf{b}_o \right)$$

3.5 Μέθοδοι Βελτιστοποίησης

Η μέθοδος κατάβασης δυναμικού (gradient descent) είναι ένας από τους πιο δημοφιλείς αλγορίθμους για την βελτιστοποίηση συναρτήσεων και μακράν ο πιο κοινός τρόπος για τη βελτιστοποίηση των νευρωνικών δικτύων. Μέσω της μεθόδου κατάβασης δυναμικού επιτυγχάνεται η ελαχιστοποίηση της αντικειμενικής συνάρτησης (συνάρτησης κόστους) $J(\boldsymbol{\theta})$, η οποία αποτελεί συνάρτηση των παραμέτρων $\boldsymbol{\theta}$ του νευρωνικού δικτύου, ανανεώνοντας αυτές τις παραμέτρους προς στην αντίθετη κατεύθυνση της κλίσης $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ της αντικειμενικής συνάρτησης. Στην μέθοδο χρησιμοποιείται και ο συντελεστής η ο οποίος ονομάζεται ρυθμός εκμάθησης και καθορίζει το μέγεθος των βημάτων που παίρνουμε για να φτάσουμε σε ένα (τοπικό) ελάχιστο. Με άλλα λόγια, ακολουθούμε την κατεύθυνση της κλίσης της επιφάνειας που δημιουργείται από την αντικειμενική συνάρτηση προς τα κάτω μέχρι να καταλήξουμε σε μια κοιλάδα.

3.5.1 Παραλλαγές της μεθόδου κατάβασης δυναμικού

Υπάρχουν τρεις παραλλαγές της μεθόδου κατάβασης δυναμικού οι οποίες διαφέρουν στην ποσότητα των δεδομένων που χρησιμοποιούν κάθε φορά για να υπολογίσουν την κλίση της αντικειμενικής συνάρτησης και την ανανέωση των βαρών. Ανάλογα με το πλήθος των δεδομένων, γίνεται συμβιβασμός μεταξύ της ακρίβειας με την οποία ανανεώνονται οι παράμετροι του δικτύου και του χρόνου που απαιτείται για την ανανέωση αυτών και κατ' επέκταση του χρόνου για την τελική σύγκλιση του προς εκπαίδευση μοντέλου.

3.5.1.1 Κατάβαση δυναμικού κατά παρτίδες (Batch Gradient Descent)

Στην πιο απλή μορφή της, η απλή μέθοδος κατάβασης δυναμικού, γνωστή και ως *κατάβαση δυναμικού κατά παρτίδες (batch gradient descent)*, ανανεώνει κατάλληλα τα βάρη του δικτύου αφού πρώτα υπολογίσει το κόστος και τις κλίσεις για όλα δεδομένα που βρίσκονται στο σύνολο εκπαίδευσης.

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \eta J(\boldsymbol{\theta}_n)$$

Η απαίτηση του να υπολογίζει και να κρατάει τις κλίσεις για ολόκληρο το σύνολο δεδομένων προκειμένου να ανανεώσει τις παραμέτρους στο τέλος της κάθε εποχής εκπαίδευσης, μπορεί να καταλήξει να γίνει πολύ αργή και κοστοβόρα ειδικά σε σύνολα εκπαίδευσης τα οποία είναι μεγάλα σε όγκο και δεν χωράνε στην μνήμη.

3.5.1.2 Στοχαστική κατάβαση δυναμικού (Stochastic Gradient Descent)

Η στοχαστική κατάβαση δυναμικού ακολουθεί τελείως αντίθετη προσέγγιση σε σχέση με την κατάβαση δυναμικού κατά παρτίδες. Αντί να ανανεώνει τις παραμέτρους μετά το πέρασμα ολόκληρου του συνόλου δεδομένων, τις ανανεώνει για κάθε πρότυπο εκπαίδευσης $\mathbf{x}^{(i)}$ με στόχο $\mathbf{y}^{(i)}$, το οποίο επιλέγεται τυχαία κάθε φορά. Σύμφωνα με αυτήν την στοχαστική εκδοχή της κατάβασης δυναμικού, δεν υπολογίζεται ολόκληρη η κλίση $\nabla_{\theta} J(\theta_n)$, αλλά η κλίση $\nabla_{\theta} J_i(\theta_n, \mathbf{x}^i, \mathbf{y}^i)$ και η ανανέωσή γίνεται ως εξής:

$$\theta_{n+1} = \theta_n - \eta \nabla_{\theta} J_i(\theta_n; \mathbf{x}^i, \mathbf{y}^i)$$

Η κατάβαση δυναμικού κατά παρτίδες εκτελεί περιττούς υπολογισμούς για μεγάλα σύνολα δεδομένων, διότι επαναπροσδιορίζει κλίσεις για παρόμοια παραδείγματα πριν την από κάθε ενημέρωση παραμέτρων. Η στοχαστική κατάβαση δυναμικού απομακρύνει αυτή την πλεονασματικότητα εκτελώντας μία ενημέρωση τη φορά. Επομένως εποδεικνύεται πολλές φορές και ειδικά σε περιπτώσεις μεγάλων δεδομένων εκπαίδευσης πολύ ταχύτερη.

Οι συχνές ενημερώσεις της στοχαστικής κατάβασης δυναμικού παρουσιάζουν μεγάλη διακύμανση κάτι που προκαλεί και έντονες μεταβολές στην αντικειμενική συνάρτηση. Οι έντονες διακυμάνσεις της στοχαστικής κατάβαση δυναμικού, αφενός, της επιτρέπουν να μεταπηδήσει σε νέα και ενδεχομένως καλύτερα τοπικά ελάχιστα, αφετέρου όμως αυτό τελικά μπορεί να περιπλέξει τη τελική σύγκλιση. Ωστόσο, έχει αποδειχθεί ότι όταν μειώνουμε αργά τον ρυθμό εκμάθησης, η στοχαστική κατάβαση δυναμικού δείχνει την ίδια συμπεριφορά με αυτήν της κατάβασης δυναμικού κατά παρτίδες και σχεδόν σίγουρα συγκλίνει σε ένα τοπικό ή ολικό ελάχιστο για μη κυρτή και κυρτή βελτιστοποίηση αντίστοιχα.

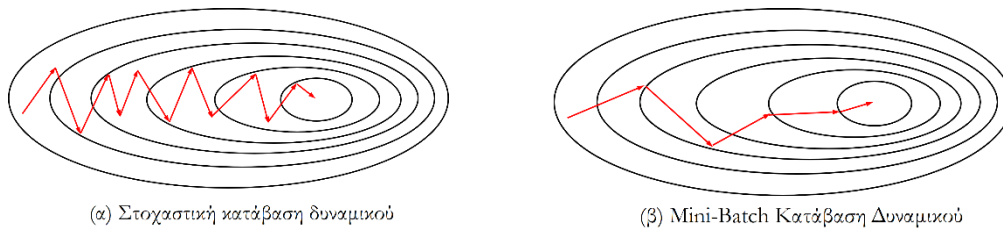
3.5.1.3 Κατάβαση δυναμικού κατά μικρές παρτίδες (Mini-Batch Gradient Descent)

Η κατάβαση δυναμικού κατά μικρές παρτίδες αποτελεί ένα συνδυασμός των μεθόδων κατάβασης δυναμικού κατά παρτίδες και στοχαστικής κατάβασης δυναμικού, επιλέγοντας κάθε φορά μια μικρή παρτίδα μεγέθους n από το σύνολο δεδομένων εκπαίδευσης και ανανεώνοντας τις παραμέτρους του δικτύου μετά από κάθε τέτοια παρτίδα. Η ανανέωση γίνεται ως εξής:

$$\theta_{n+1} = \theta_n - \eta \nabla_{\theta} J(\theta_n; \mathbf{x}^{i:i+n}, \mathbf{y}^{i:i+n})$$

Με αυτόν τον τρόπο επιτυγχάνεται: α) η μείωση της διακύμανσης των ενημερώσεων των παραμέτρων από την οποία πάσχει η στοχαστική κατάβαση δυναμικού, γεγονός που μπορεί να

οδηγήσει σε πιο σταθερή σύγκλιση. και β) ο υπολογισμός της κλίσης ως προς μια μικρή παρτίδα εξακολουθεί να είναι πολύ γρήγορος και αποτελεσματικός ακόμα και σε μεγάλα σύνολα εκπαίδευσης, αντιμετωπίζοντας έτσι την αδυναμία της απλής κατάβασης δυναμικού κατά παρτίδες. Τα κοινά μεγέθη μίνι-παρτίδων κυμαίνονται μεταξύ 50 και 256, αλλά μπορούν να ποικίλουν για διαφορετικές εφαρμογές. Η κατάβαση δυναμικού κατά μικρές παρτίδες είναι τυπικά ο πιο συνηθισμένος αλγόριθμος επιλογής κατά την εκπαίδευση ενός νευρικού δικτύου και ο όρος *στοχαστική κατάβαση δυναμικού* συνηθίζεται να χρησιμοποιείται και όταν χρησιμοποιούνται μικρές παρτίδες.



Εικόνα 18 Σύγκριση σύγκλισης μεθόδων (α) στοχαστικής κατάβασης δυναμικού και (β) mini-batch κατάβασης δυναμικού

3.5.2 Ορμή (Momentum)

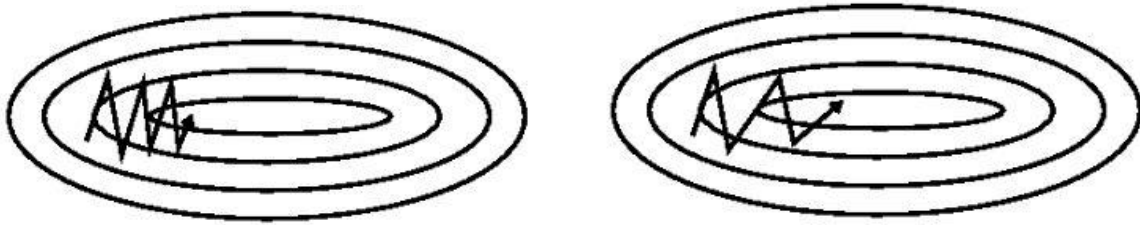
Υπάρχουν περιπτώσεις όπου η επιφάνεια της αντικειμενικής συνάρτησης παρουσιάζει περιοχές με την μορφή «μακριών χαράδρων» με απότομα τοιχώματα στα πλάγια, μέσω των οποίων η συνάρτηση βελτιστοποίησης θα πρέπει να περάσει προκειμένου να φτάσει στο επιθυμητό βέλτιστο ελάχιστο. Σε τέτοιες περιπτώσεις η τυποποιημένη *στοχαστική κατάβαση δυναμικού* τείνει να ταλαντεύεται κατά μήκος της στενής χαράδρας αφού η αρνητική κλίση θα δείχνει προς μία από τις απότομες πλευρές και όχι κατά μήκος της χαράδρας όπου βρίσκεται το βέλτιστο. Αυτή η μορφολογία παρουσιάζεται αρκετά συχνά κοντά στο βέλτιστο ελάχιστο με αποτέλεσμα η *στοχαστική κατάβαση δυναμικού* να οδηγεί σε πολύ αργή σύγκλιση. Η *ορμή* είναι μια μέθοδος για την ταχύτερη ώθηση της της αντικειμενικής συνάρτησης σε τέτοιες «χαράδρες». Αυτό το καταφέρει προσθέτοντας ένα ποσοστό της ανανέωσης των παραμέτρων από προηγούμενο βήμα στον τρέχον βήμα. Η *στοχαστική κατάβαση δυναμικού με ορμή* ανανεώνει κάθε παράμετρο ως εξής:

$$\mathbf{u}_t = \gamma \mathbf{u}_{t-1} + \eta \nabla_{\theta} J(\theta; \mathbf{x}^i, \mathbf{y}^i)$$

$$\theta = \theta - \mathbf{u}_t$$

Στην παραπάνω σχέση \mathbf{u} είναι το τρέχον διάνυσμα ταχύτητας το οποίο έχει τις ίδιες διαστάσεις με το διάνυσμα θ και γ ο συντελεστής ορμής. Ο ρυθμός εκμάθησης έχει την ίδια λειτουργία όπως και πριν ωστόσο όταν χρησιμοποιείται σε συνδυασμό με την ορμή μπορεί να χρειαστεί να πάρει

μικρότερες τιμές καθώς το μέτρο της κλίσης θα είναι μεγαλύτερο. Ο συντελεστής ορμής $\gamma \in (0,1]$ καθορίζει ουσιαστικά πόσες επαναλήψεις πριν σε τι ποσοστό οι προηγούμενες κλίσεις θα συμμετέχουν στην τρέχουσα ανανέωση των παραμέτρων. Μέσω της βελτιστοποίησης της ορμής, ενισχύονται οι διαστάσεις των οποίων η κλίση τους δείχνει πάντα προς την ίδια κατεύθυνση και ελαττώνονται αυτές που οι διαστάσεις τους αλλάζουν κατευθύνσεις. Έτσι μειώνονται σημαντικά οι ταλαντεύσεις της αντικειμενικής συνάρτησης και επιτυγχάνεται γρηγορότερη σύγκλιση προς το ζητούμενο ελάχιστο.



Εικόνα 19 Σύγκριση σύγκλισης μεθόδων (αριστερά) απλής στοχαστικής κατάβασης δυναμικού και (δεξιά) στοχαστικής κατάβασης δυναμικού με ορμή

3.5.3 Ο αλγόριθμος βελτιστοποίησης Adagrad

Ο αλγόριθμος *Adagrad* είναι ένας αλγόριθμος βελτιστοποίησης βασισμένος στην βελτιστοποίηση κατάβασης δυναμικού ο οποίος προσαρμόζει τον ρυθμό εκμάθησης στις εκάστοτε παραμέτρους κάνοντας μικρότερες ανανεώσεις για παραμέτρους που σχετίζονται με χαρακτηριστικά που εμφανίζονται πιο συχνά και μεγαλύτερες ανανεώσεις για παραμέτρους που σχετίζονται με λιγότερο συχνά χαρακτηριστικά.

Προηγούμεως, εκτελούσαμε μια κοινή ενημέρωση για όλες τις παραμέτρους θ , καθώς για κάθε παράμετρο θ_i χρησιμοποιούσαμε κοινό και σταθερό ρυθμό εκμάθησης η . Ο αλγόριθμος *Adagrad* χρησιμοποιεί διαφορετικό ρυθμό εκμάθησης για κάθε παράμετρο θ_i σε κάθε χρονικό βήμα t και επομένως η σχέση που περιγράφει την ανανέωση για μια παράμετρο είναι η εξής:

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \varepsilon}} \cdot \nabla_{\theta} J(\theta_{t,i})$$

όπου $G_{t,ii} \in \mathbb{R}^{d \times d}$ είναι ένας διαγώνιος πίνακας όπου κάθε στοιχείο i, i της διαγωνίου είναι το άθροισμα των τετραγώνων των κλίσεων ως προς την παράμετρο θ_i μέχρι το χρονικό βήμα t , ενώ ε είναι ένας πολύ μικρός αριθμός προκειμένου να αποφευχθεί τυχόν διαίρεση με το μηδέν.

Ένα από τα κύρια οφέλη του Adagrad είναι ότι εξαλείφει την ανάγκη για συντονισμό του ρυθμού εκμάθησης με το χέρι. Οι περισσότερες υλοποιήσεις χρησιμοποιούν μια προεπιλεγμένη τιμή 0,01

και αρχούνται σε αυτό. Η κύρια αδυναμία του *Adagrad* είναι η συσσώρευση των τετραγωνικών κλίσεων στον παρονομαστή. Δεδομένου ότι κάθε προστιθέμενος όρος είναι θετικός, το συσσωρευμένο ποσό αυξάνεται ολοένα και περισσότερο κατά τη διάρκεια της εκπαίδευσης. Αυτό με τη σειρά του αναγκάζει τον ρυθμό εκμάθησης να συρρικνώνεται σε κάθε διάσταση και τελικά να γίνεται απεριορίστα μικρός, οπότε ο αλγόριθμος δεν είναι πλέον σε θέση να αποκτήσει επιπλέον γνώση.

3.5.4 Ο αλγόριθμος βελτιστοποίησης Adadelta

Ο αλγόριθμος *Adadelta* αποτελεί επέκταση του αλγορίθμου βελτιστοποίησης *Adagrad* και επιδιώκει να μειώσει την επιθετική, μονοτονική μείωση του ρυθμού εκμάθησης του δεύτερου. Η λογική του βασίζεται στο αντί να συσσωρεύει όλες τις παρελθοντικές τετραγωνικές κλίσεις, ο αλγόριθμος *Adadelta* περιορίζει το παράθυρο των συσσωρευμένων κλίσεων του παρελθόντος σε κάποιο σταθερό μέγεθος w .

Ωστόσο, αντί απλά να αποθηκεύει με μη αποδοτικό τρόπο τα τετράγωνα των w προηγούμενων κλίσεων, ο αλγόριθμος *Adadelta* ορίζει αναδρομικά το άθροισμα των κλίσεων ως τον ελαττούμενο μέσο όρο των παρελθοντικών τετραγωνισμένων κλίσεων. Ο τρέχον μέσος όρος $E[g^2]_t$ την χρονική στιγμή t , εξαρτάται επομένως (ως ένα ποσοστό γ , παρόμοια με την λογική του συντελεστή ορμής) μόνο από τις προηγούμενες και την τρέχουσα κλίσεις:

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1 - \gamma) g_t^2$$

όπου παραπάνω συμβολίσαμε ως g_t το διάνυσμα των κλίσεων $\nabla_{\theta} J(\theta_t)$ για την χρονική στιγμή t , ενώ ο συντελεστής γ που χρησιμοποιήθηκε παίρνει τιμές παρόμοιες με αυτές του συντελεστή ορμής. Επιπλέον το διάνυσμα ανανέωσης το συμβολίζουμε τώρα ως $\Delta\theta_t$ και έτσι για παράδειγμα η ανανέωση της στοχαστικής κατάβασης δυναμικού γράφεται:

$$\begin{aligned} \Delta\theta_t &= -\eta \cdot g_t \\ \theta_{t+1} &= \theta_t + \Delta\theta_t \end{aligned}$$

Με αντίστοιχο τρόπο, με βάση ότι αναφέρθηκε παραπάνω, το διάνυσμα ανανέωσης του αλγορίθμου βελτιστοποίησης *Adadelta* είναι το εξής:

$$\Delta\theta_t = -\frac{\eta}{\sqrt{E[g^2]_t + \varepsilon}} g_t = -\frac{\eta}{RMS[g]_t} g_t$$

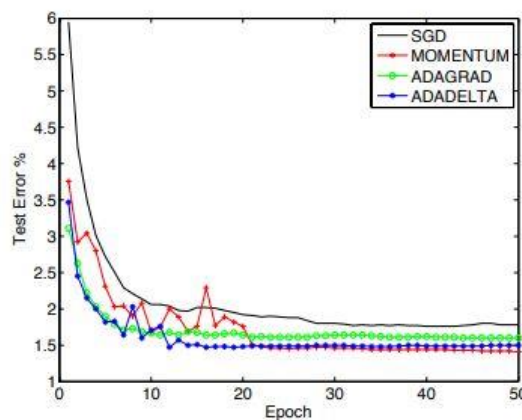
Ωστόσο ένα θέμα που επισημάνθηκε από τους συγγραφείς του αλγορίθμου *Adadelta* όσον αφορά την παραπάνω σχέση ανανέωσης και το οποίο παρουσιάζεται και στις ανανεώσεις των αλγορίθμων *στοχαστικής κατάβασης δυναμικού, ορμής και Adagrad*, είναι η ασυνέπεια που υπάρχει στις μονάδες. Κατά την εξέταση των ενημερώσεων $\Delta\theta$, που εφαρμόζονται στις παραμέτρους θ , οι μονάδες πρέπει να ταιριάζουν. Δηλαδή, εάν η παράμετρος θ είχε κάποιες υποθετικές μονάδες, οι αλλαγές στην παράμετρο $\Delta\theta$ πρέπει να είναι αλλαγές με τις μονάδες. Ωστόσο σύμφωνα με τις σχέσεις ανανέωσης:

$$\text{μονάδες του } \Delta\theta \propto \text{μονάδες του } g \propto \text{μονάδες του } \frac{\partial J}{\partial \theta} \propto \frac{1}{\text{μονάδες του } \theta}$$

Παρατηρώντας αυτή την αναντιστοιχία των μονάδων, θα πρέπει να σκεφτούμε τους όρους που πρέπει να προσθέσουμε στο διάνυσμα ανανέωσης του *Adadelta*, για να ταιριάζουν οι μονάδες της ενημέρωσης με τις μονάδες των παραμέτρων. Καθώς η μέση τετραγωνική ρίζα (RMS) των κλίσεων προηγούμενων χρονικών στιγμών υπάρχει ήδη στον παρονομαστή της σχέσης ανανέωσης, θεωρούμε μια μέτρηση της ποσότητας $\Delta\theta$ για τον αριθμητή. Καθώς η $\Delta\theta_t$ δεν είναι γνωστή για το τρέχον χρονικό βήμα, θεωρούμε ότι η καμπυλότητα τοπικά είναι ομαλή και προσεγγίζουμε την ποσότητα $\Delta\theta_t$ υπολογίζοντας την εκθετικά ελαττούμενη RMS τιμή για ένα παράθυρο μεγέθους w για τις παρελθοντικές $\Delta\theta$. Έτσι τελικά, η ανανέωση των παραμέτρων για τον αλγόριθμο *Adadelta*, γίνεται ως εξής:

$$\Delta\theta_t = -\frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} g_t$$

$$\theta_{t+1} = \theta_t + \Delta\theta_t$$



Εικόνα 20 Σύγκριση σύγκλισης αλγορίθμων, SGD, Ορμής, Adagrad, Adadelta

3.5.5 Ο αλγόριθμος βελτιστοποίησης RMSprop

Ο αλγόριθμος RMSprop αναπτυσσόταν ανεξάρτητα από τον αλγόριθμο Adadelta και περίπου την ίδια χρονική περίοδο, προσπαθώντας και αυτός να λύσει το πρόβλημα της πολύ επιθετικής μείωσης του ρυθμού εκμάθησης του αλγορίθμου Adagrad. Η λύση στην οποία κατέληξε, είναι σχεδόν πανομοιότυπη με αυτήν του Adadelta, δίνοντας ενημέρωση:

$$E[\mathbf{g}^2]_t = 0,9[\mathbf{g}^2]_{t-1} + 0,1\mathbf{g}_t^2$$
$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[\mathbf{g}^2]_t + \varepsilon}} \mathbf{g}_t$$

Φαίνεται ότι και RMSprop προσπαθεί να λύσει το πρόβλημα, προτείνοντας την διαίρεση του ρυθμού εκμάθησης με έναν εκθετικά ελαττούμενο μέσο όρο τετραγωνισμένων κλίσεων. Εδώ προτείνεται ο όρος της ορμής να είναι κοντά στο 0.9 και αντίστοιχα η αρχική τιμή του ρυθμού εκμάθησης κοντά στο 0.1.

ΚΕΦΑΛΑΙΟ 4

ΔΙΚΤΥΑ ΜΑΚΡΑΣ ΒΡΑΧΥΠΡΟΘΕΣΜΗΣ ΜΝΗΜΗΣ

Τα δίκτυα Μακράς Βραχυπρόθεσμης Μνήμης (LSTM) είναι μια κατηγορία αναδρομικών δικτύων τα οποία έχουν τραβήξει το ενδιαφέρον και έχουν γίνει πολύ δημοφιλή τα τελευταία χρόνια. Τα δίκτυα LSTMs δεν έχουν ριζικά διαφορετική αρχιτεκτονική από τα κλασικά αναδρομικά δίκτυα, αλλά χρησιμοποιούν μια διαφορετική και πιο σύνθετη μεθοδολογία για τον υπολογισμό της κρυφής τους κατάσταση. Η πέραση τους και η επιτυχία τους σε πολλά προβλήματα οφείλεται στην αποτελεσματικότητά τους να ξεπερνάνε το πρόβλημα των μακρινών εξαρτήσεων, κάτι στο οποίο τα απλά αναδρομικά δίκτυα υστερούνε αρκετά.

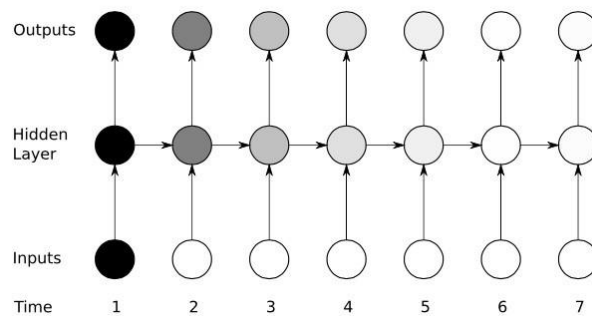
4.1 Το πρόβλημα των μακρινών εξαρτήσεων στα απλά αναδρομικά νευρωνικά δίκτυα

Ένα από τα προτερήματα των αναδρομικών νευρωνικών δικτύων είναι η ιδιότητα τους να συνδέουν προηγούμενες εισόδους με την ελάχιστοτε τρέχουσα, κάτι που τα κάνει πολύ δημοφιλή όταν πρόκειται για επεξεργασία δεδομένων τα οποία παρουσιάζουν χρονική εξάρτηση μεταξύ τους. Τέτοια παραδείγματα είναι περιπτώσεις προβλημάτων επεξεργασίας φυσικής γλώσσας, όπου κάθε λέξη αποτελεί μια είσοδο στο δίκτυο και επιθυμούμε οι εξαρτήσεις μεταξύ των λέξεων, μέσα στην πρόταση, να διατηρούνται όσο μακριά και βρίσκονται μεταξύ τους. Έτσι μπορούμε να χρησιμοποιούμε την πληροφορία των συμφραζομένων και να συγκρατούμε όσο το δυνατόν περισσότερο νόημα με αποτέλεσμα στο τέλος να έχουμε την δυνατότητα να συμπεραίνουμε για το συναίσθημα του κειμένου, να προβλέψουμε την επόμενη λέξη κ.α.

Ωστόσο, όταν πρόκειται για μαρκινές εξαρτήσεις, δηλαδή για εξαρτήσεις μεταξύ δυο χρονικών βημάτων που απέχουν αρκετά μεταξύ τους, τα αναδρομικά δίκτυα αδυνατούν να τις διατηρήσουν. Αυτό οφείλεται στο γεγονός ότι κατά την διάρκεια της εκπαίδευσης, για να καθοριστεί το σφάλμα και επομένως και η εξάρτηση της εξόδου της χρονικής στιγμής t σε σχέση με την είσοδο της χρονικής στιγμής $t-n$, θα παρεμβληθεί γινόμενο n παραγώγων σύμφωνα με τον κανόνα της αλυσίδας.

$$\underbrace{\frac{\partial h_t}{\partial_{t-1}} \frac{\partial h_{t-1}}{\partial_{t-2}} \dots \frac{\partial h_{t-n+1}}{\partial_{t-n}}}_n$$

Αυτό έχει ως αποτέλεσμα, όταν οι κλίσεις $\frac{\partial h_t}{\partial_{t-1}}$ είναι μικρές, δηλαδή της τάξης 10^{-1} ή και μικρότερες, το συνολικό γινόμενο να τείνει στο μηδέν, με αποτέλεσμα η επιρροή μια αρχικής χρονικής στιγμής σε μια μαρκινή μεταγενέστερη στιγμή, ουσιαστικά να εξαφανίζεται. Αυτό το φαινόμενο είναι γνωστό και ως πρόβλημα των εξαφανιζόμενων κλίσεων (Vanishing Gradient Problem). Αυτό το πρόβλημα αποτελεί μιας από τις αδυναμίες των απλών αναδρομικών δικτύων καθώς εισάγει περιορισμούς στο βάθος των αρχιτεκτονικών που καταδικάζουν τα μοντέλα σε κορεσμό των επιδόσεων τους.



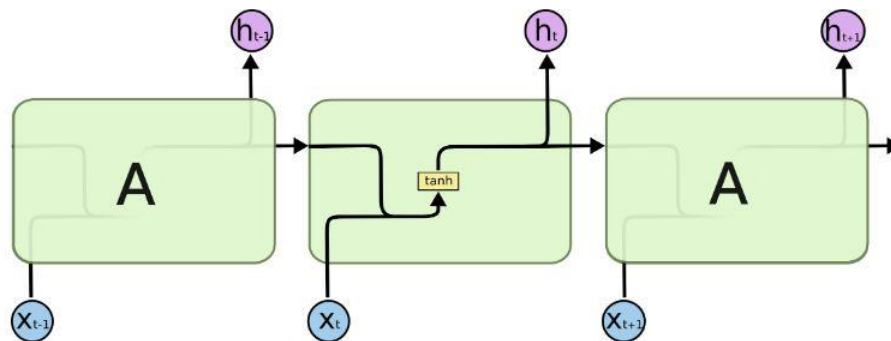
Εικόνα 21 Το πρόβλημα μείωσης της κλίσης για τα απλά αναδρομικά δίκτυα

. Η σκίαση των κόμβων στο ξεδιπλωμένο δίκτυο υποδεικνύει την εναισθησία τους στις εισόδους κατά την πρώτη χρονικά είσοδο (όσο πιο σκούρα είναι η σκιά, τόσο μεγαλύτερη είναι η εναισθησία). Η εναισθησία αποσυντίθεται με την πάροδο του χρόνου καθώς οι νέες εισόδοι αντικαθιστούν τις ενεργοποιήσεις του κρυμμένου στρώματος και το δίκτυο «ξεχνά» τις πρώτες εισόδους.

4.2 Τα Δίκτυα Μακράς Βραχυπρόθεσμης Μνήμης

4.2.1 Γενικά

Όπως έχει αναφερθεί, τα δίκτυα Μακράς Βραχυπρόθεσμης Μνήμης (LSTM) αποτελούν μια ειδική κατηγορία αναδρομικών νευρωνικών δικτύων τα οποία καταφέρνουν να ξεπερνούν τον πρόβλημα των μακρινών εξαρτήσεων. Είναι σχεδιασμένα με τέτοιο τρόπο προκειμένου να επιτυγχάνουν να διατηρούν ένα πιο σταθερό σφάλμα κατά την ανάστροφη μετάδοσή του μέσα στον χρόνο, μαθαίνοντας έτσι ακόμα και μέσα από πολύ μεγάλες ακολουθίες χρονικών βημάτων. Η δομική διαφορά τους που τα ξεχωρίζει από τα απλά αναδρομικά νευρωνικά δίκτυα, έγκειται στην αρχιτεκτονική του κρυφού τους επιπέδου. Σε ένα απλό αναδρομικό δίκτυο, το κρυφό επίπεδο του δικτύου, αποτελείται από ένα επίπεδο νευρώνων με μια καθορισμένη συνάρτηση ενεργοποίησης.



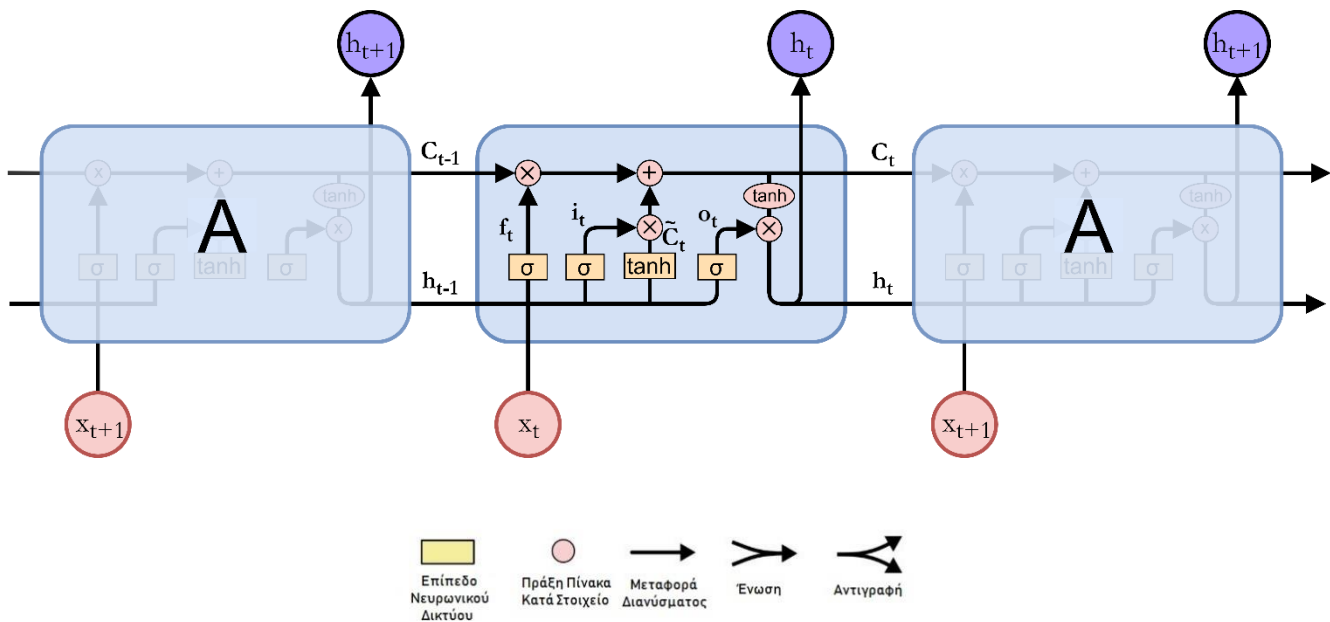
Εικόνα 22 Κρυφό επίπεδο απλού αναδρομικού δικτύου

Το κρυφό επίπεδο των δικτύων LSTM, το οποίο συνηθίζεται να αποκαλείται κύτταρο LSTM, ακολουθεί μια πιο σύνθετη δομή καθώς, αντί να έχει ένα απλό νευρωνικό δίκτυο, διαθέτει τέσσερα τα οποία αλληλοεπιδρούν μεταξύ τους. Μέσω αυτών, οι πληροφορίες μπορούν να αποθηκευτούν, να γραφτούν ή να διαβαστούν όπως τα δεδομένα στη μνήμη ενός υπολογιστή. Το κύτταρο του LSTM λαμβάνει αποφάσεις σχετικά με το τι πρέπει να αποθηκεύει και πότε πρέπει να επιτρέπεται η ανάγνωση, η εγγραφή και η διαγραφή, μέσω κατάλληλων πυλών που ανοίγουν και κλείνουν.

4.2.2 Το κύτταρο μακράς βραχυπρόθεσμης μνήμης

Το κύτταρο μακράς βραχυπρόθεσμης μνήμης γνωστό απλά και ως κύτταρο μνήμης που δομεί τα ομώνυμα αναδρομικά νευρωνικά δίκτυα αποτελείται από τα εξής βασικά δομικά στοιχεία: 4 μονό-επίπεδα νευρωνικά δίκτυα με συγκεκριμένες συναρτήσεις ενεργοποίησης, 3 πολλαπλασιαστές, έναν αθροιστή και την κατάσταση του κυττάρου. Η κατάσταση του κυττάρου (C_t) μπορεί να θεωρηθεί σαν μια γραμμή διαβίβασης που διατρέχει όλο το σώμα του κυττάρου, συμμετέχει στην ανάδραση και στην οποία το κύτταρο μνήμης μπορεί να γράφει και να διαγράφει πληροφορία διαμορφώνοντας έτσι την κατάσταση του την οποία την μεταβιβάζει σε κάθε επόμενο χρονικό βήμα.

Η πληροφορία γράφεται και διαγράφεται από την κατάσταση του κυττάρου μέσω δομών που ονομάζονται πύλες. Οι πύλες δημιουργούνται από τον συνδυασμό ενός μονό-επίπεδου νευρωνικού δικτύου και ενός πολλαπλασιαστή. Ο σκοπός μιας πύλης είναι να ρυθμίζει την πληροφορία που ρέει προς την κατάσταση του κυττάρου. Για αυτό τον λόγο το επίπεδο του νευρωνικού δικτύου που την δομεί συνηθίζεται να έχει ως συνάρτηση ενεργοποίησης την σιγμοειδή συνάρτηση η οποία φράζεται μεταξύ 0 και 1. Έτσι παράγοντας αριθμούς μεταξύ 0 και 1 καθορίζεται το ποσοστό πληροφορίας που θα περάσει. Στο κύτταρο μακράς βραχυπρόθεσμης μνήμης υπάρχουν τρεις πύλες: η πύλη εισόδου (input gate), η πύλη εξόδου (output gate) και η πύλη διαγραφής μνήμης (forget gate).



Εικόνα 23 Δομή γρήγορου επιπέδου κυττάρου LSTM

4.2.3 Λειτουργία κυττάρου μακράς βραχυπρόθεσμης μνήμης

Η λειτουργία του κυττάρου LSTM θα αναλυθεί σε στάδια καθένα από τα οποία αντιπροσωπεύει μια ξεχωριστή του λειτουργία. Κάθε λειτουργία θα περιγράφεται και από την αντίστοιχη μαθηματική της αναπαράσταση.

- Στάδιο Διαγράφης μνήμης (Forgetting)

Το πρώτο βήμα στην λειτουργία του κυττάρου μνήμης, είναι να αποφασιστεί ποια πληροφορία πρόκειται να μην συμπεριληφθεί στην κατάσταση κυττάρου. Αυτό επιτυγχάνεται μέσω της πύλης διαγραφής μνήμης, όπου λαμβάνοντας το σήμα ανάδρασης h_{t-1} και την τρέχουσα είσοδο x_t , το νευρωνικό δίκτυο με την σιγμοειδή ενεργοποίηση παράγει έναν αριθμό μεταξύ 0 και 1. Η έξοδος αυτή πολλαπλασιάζεται με κάθε αριθμό του διανύσματος C_{t-1} της προηγούμενης κατάστασης, ρυθμίζοντας έτσι το ποια πληροφορία θα «ξεχαστεί». Η μαθηματική διατύπωση αυτού του σταδίου είναι η εξής:

$$f_t = \sigma(W_f [h_{t-1}, x_t] + b_f)$$

- Στάδιο Εισόδου

Στη συνέχεια θα καθοριστεί ποια πληροφορία από την είσοδο $[h_{t-1}, x_t]$ θα αποθηκευτεί στην κατάσταση του κυττάρου μνήμης. Αυτή η λειτουργία υλοποιείται σε δύο βήματα. Αρχικά η είσοδος τροφοδοτείται στην πύλη εισόδου η οποία καθορίζει ποιες τιμές θα ανανεωθούν. Ταυτόχρονα τροφοδοτείται και σε ένα μονό-επίπεδο νευρωνικό δίκτυο με συνάρτηση ενεργοποίησης την υπερβολική εφαπτομένη (\tanh) το οποίο εξάγει τις νέες συμπιεσμένες τιμές \tilde{C}_t που είναι υποψήφιος για να προστεθούν στην κατάσταση του κυττάρου.

$$\begin{aligned} i_t &= \sigma(W_i [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \end{aligned}$$

- Στάδιο Ανανέωσης Κατάστασης Κυττάρου

Στο στάδιο αυτόν η κατάσταση του κυττάρου μνήμης C_{t-1} ανανεώνεται με τις εξόδους των προηγούμενων δύο σταδίων δημιουργώντας την κατάσταση C_t . Πιο συγκεκριμένα, μέσω του πολλαπλασιαστή την πύλης διαγραφής μνήμης, η C_{t-1} αρχικά συνδυάζεται με την f_t για να ενημερωθεί για το ποια πληροφορία πρέπει να ξεχαστεί. Στη συνέχεια αθροίζεται με την έξοδο του σταδίου Εισόδου $i_t * \tilde{C}_t$, η οποία περιέχει τις νέες υποψήφιος τιμές, κλιμακωμένες

κατά το πόσο αποφασίστηκε να ενημερώσουμε την τρέχουσα κατάσταση. Η μαθηματική διατύπωση αυτού του σταδίου είναι η εξής:

$$C_t = f_t + i_t * \tilde{C}_t$$

- Στάδιο Εξόδου

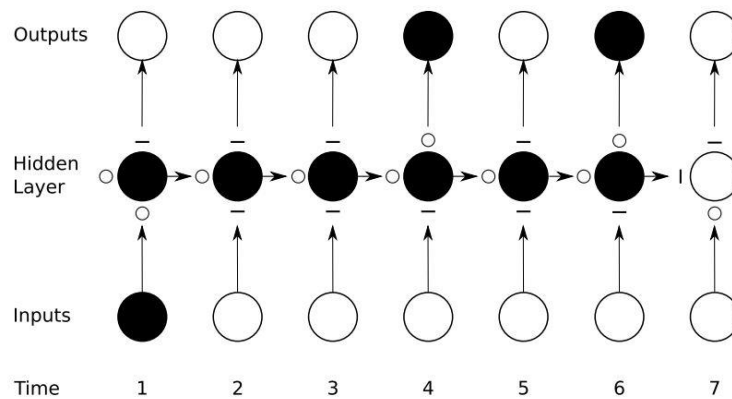
Στο τελευταίο στάδιο καθορίζεται η τελική έξοδος h_t του κυττάρου LSTM. Η έξοδος αυτή θα βασιστεί στην τρέχουσα κατάσταση του κυττάρου C_t και θα αποτελεί μια φιλτραρισμένη εκδοχή της. Αρχικά συμπιέζουμε την κατάσταση C_t μέσω της συνάρτησης \tanh . Στη συνέχεια η είσοδος $[h_{t-1}, x_t]$ περνάει από το νευρωνικό δίκτυο της πύλης εξόδου με σιγμοειδή ενεργοποίηση και το αποτέλεσμα αυτό (o_t) συνδυάζεται μέσω του πολλαπλασιαστή με την συμπιεσμένη κατάσταση κυττάρου, αποφασίζοντας έτσι ποια μέρη της κατάστασης αυτής θα συμμετέχουν στην τελική έξοδο.

$$\begin{aligned} o_t &= \sigma W_o [h_{t-1}, x_t] + b_o \\ h_t &= \tanh(o_t) \end{aligned}$$

4.2.4 Αρχιτεκτονικές και Εκπαίδευση Δικτύων Μακράς Βραχυπρόθεσμης Μνήμης

Τα δίκτυα LSTM ως αναδρομικά δίκτυα μπορούν να ακολουθήσουν όλες τις αρχιτεκτονικές που σχηματίζει και ένα απλό αναδρομικό δίκτυο. Έτσι υπάρχουν τα δίκτυα LSTM ενός επιπέδου, τα δίκτυα LSTM πολλών επιπέδων, τα αμφίδρομα LSTM (BLSTM) δίκτυα καθώς και αυτά με μία ή περισσότερες εξόδους ανάλογα με την φύση του προβλήματος που καλούνται να επιλύσουν κάθε φορά.

Η εκπαίδευση των δικτύων LSTM σε αντιστοιχία με τα απλά αναδρομικά δίκτυα μπορεί να επιτευχθεί βασιζόμενο στον βελτιστοποιητή κατάβασης δυναμικού με ανάστροφη μετάδοση σφάλματος μέσα στον χρόνο (BPTT). Ο στόχος είναι να ανανεωθούν κατάλληλα τα βάρη των τεσσάρων μονό-επιπέδων νευρωνικών δικτύων που διαθέτει το κύτταρο LSTM, προκειμένου αυτό δεχόμενο μια ακολουθία εισόδων να μάθει πλήρως και ακριβώς το τι να διαγράφει από την μνήμη, τι να εισάγει και τι να εξάγει σε κάθε χρονικό βήμα. Έτσι θα μπορεί αποτελεσματικά να διατηρεί όλες τις αναγκαίες μακρινές και μη εξαρτήσεις στην δοσμένη ακολουθία.



Εικόνα 24 Απεικόνιση της διατήρησης των μακρών εξαρτήσεων σε δίκτυο LSTM

Όπως και στην Εικόνα 20, η σκίαση των κόμβων υποδεικνύει την εναισθησία του δικτύου στις εισόδους την πρώτη φορά. Σε αυτή την περίπτωση οι μαύροι κόμβοι είναι μέγιστα εναισθητοί και οι λευκοί κόμβοι πλήρως ανελαστικοί. Η κατάσταση των πυλών εισόδου, διαγραφής μνήμης και εξόδου εμφανίζονται από κάτω, από τα αριστερά και από πάνω από το κρυφό επίπεδο αντίστοιχα. Για λόγους απλότητας, όλες οι πύλες είναι είτε εντελώς ανοιχτές («O») είτε εντελώς κλειστές («-»). Η μνήμη του LSTM «θυμάται» την πρώτη είσοδο όσο είναι ανοιχτή η θύρα διαγραφής μνήμης και η πύλη εισόδου είναι κλειστή. Η εναισθησία του επιπέδου εξόδου μπορεί να τροποποιηθεί από την πύλη εξόδου χωρίς να επηρεαστεί το κύτταρο LSTM.

ΚΕΦΑΛΑΙΟ 5

ΤΟ ΕΡΓΑΛΕΙΟ TENSORFLOW

5.1 Εισαγωγή

Το TensorFlow (<https://www.tensorflow.org/>) είναι μια βιβλιοθήκη λογισμικού βασισμένο στο προγραμματιστικό μοντέλο ροής δεδομένων (dataflow programming paradigm), που έχει αναπτυχθεί από την Google Brain Team του ερευνητικού οργανισμού Machine Learning Intelligence της Google, με σκοπό να διευκολύνει την διεξαγωγή έρευνας και την υλοποίηση αλγορίθμων και συστημάτων μηχανικής μάθησης. Το TensorFlow έχει σχεδιαστεί ώστε να συνδυάζει την υπολογιστική άλγεβρα με τεχνικές βελτιστοποίησης υπολογίζοντας γρήγορα και αποδοτικά μαθηματικές εκφράσεις που απαιτούν μεγάλο χρόνο εκτέλεσης λόγω της υψηλής υπολογιστικής τους πολυπλοκότητας.



Τα κύρια χαρακτηριστικά του περιλαμβάνουν:

- Τον εύκολο ορισμό, την βελτιστοποίηση και τον αποδοτικό υπολογισμό μαθηματικών εκφράσεων που περιλαμβάνουν πίνακες υψηλών διαστάσεων (τανυστές, tensors).
- Την εύκολη κατασκευή και τον προγραμματισμό μοντέλων βαθιάς μάθησης και την άμεση χρήση τεχνικών μηχανικής μάθησης μέσω των προγραμματιστικών βιβλιοθηκών που υποστηρίζονται.
- Διαφανής χρήση των μονάδων υπολογισμού γραφικών (GPU) για την επιτάχυνση των υπολογισμών καθώς και αυτοματοποίηση της διαχείρισης και βελτιστοποίηση της μνήμης και των δεδομένων που χρησιμοποιούνται. Επιπλέον παρέχεται η δυνατότητα ο ίδιος κώδικας να εκτελεστεί είτε σε CPU είτε σε GPU και αφήνεται στο TensorFlow να υπολογίσει ποια μέρη του υπολογισμού πρέπει να μεταφερθούν στη GPU και ποια όχι.
- Δυνατότητα υψηλής κλιμάκωσης των υπολογισμών σε πολλά μηχανήματα και τεράστια υποστηρίξη τεράστιων συνόλων δεδομένων

5.2 Το προγραμματιστικό μοντέλο του TensorFlow

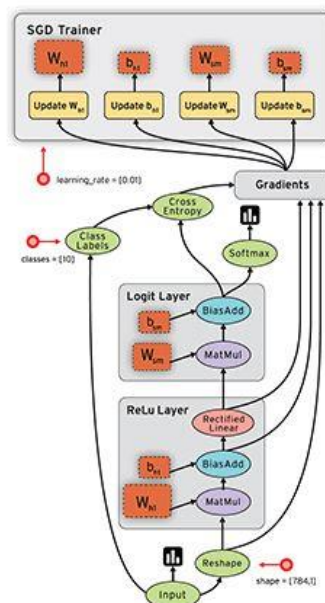
Το TensorFlow στον πυρήνα του είναι βασισμένο στο προγραμματιστικό πρότυπο της *Ροής Δεδομένων* (dataflow programming), σύμφωνα με το οποίο ένα πρόγραμμα μοντελοποιείται και αναπαρίσταται ως κατευθυνόμενος γράφος των δεδομένων που ρέουν μεταξύ πράξεων-λειτουργιών. Υιοθετώντας τον Γράφο Ροής Δεδομένων (Data Flow Graph) ως μοντέλο εκτέλεσης, καταφέρει να διαχωρίσει την διαδικασία σχεδιασμού της ροής δεδομένων (κατασκευή γράφου και ροής) από την εκτέλεση του (CPU, GPU ή συνδυασμός τους), χρησιμοποιώντας μια μοναδική προγραμματιστική διεπαφή η οποία αποκρύπτει κάθε πολυπλοκότητα.

5.2.1 Γράφοι Ροής Δεδομένων (Data Flow Graphs)

Μια εφαρμογή μηχανικής μάθησης είναι το αποτέλεσμα επαναλαμβανόμενων υπολογισμών σύνθετων μαθηματικών εκφράσεων. Στο TensorFlow, ένας υπολογισμός περιγράφεται με την χρήση των *Γράφων Ροής Δεδομένων*, όπου κάθε κόμβος στον γράφο αναπαριστά ένα στιγμιότυπο μιας μαθηματικής πράξης και κάθε ακμή του γράφου αποτελεί έναν πολυδιάστατο πίνακα-σύνολο δεδομένων (τανυστής) πάνω στο οποίο εφαρμόζονται οι μαθηματικές πράξεις αυτές. Πιο συγκεκριμένα, οι κόμβοι και οι ακμές διαχειρίζονται από το TensorFlow ως εξής:

- **Κόμβος** (node): Κάθε *κόμβος* αναπαριστά ένα στιγμιότυπο μιας μαθηματικής πράξης. Κάθε τέτοια πράξη έχει τουλάχιστον μία είσοδο και καμία ή περισσότερες εξόδους.
- **Ακμή** (edge): Στο TensorFlow οι ακμές είναι δύο τύπων:

- *Απλές Ακμές:* Οι απλές ακμές μεταφέρουν τανυστές όπου η έξοδος μιας πράξης (κόμβος) γίνεται είσοδος σε επόμενη πράξη.
- *Ειδικές Ακμές:* Οι ακμές αυτές δεν μεταφέρουν δεδομένα μεταξύ της εξόδου ενός κόμβου και της εισόδου κάποιου άλλου αλλά αντίθετα αναπαριστούν μια εξάρτηση ελέγχου μεταξύ δύο κόμβων. Αν υποθέσουμε ότι έχουμε δύο κόμβους A, B, η *ειδική ακμή* που τους συνδέει, υποδεικνύει ότι ο B θα αρχίσει την λειτουργία του αφού τελειώσει ο A την δικιά του. Έτσι οι ακμές αυτές χρησιμοποιούνται στον *Γράφο Ροής Δεδομένων* για να δηλώσουν την εξάρτηση και ποια ενέργεια από δύο κόμβους εφαρμόστηκε στα δεδομένα (τανυστές) πρώτα.



Εικόνα 26 Αναπαράσταση νευρωνικού επιπέδου ως γράφος στο TensorFlow

Πιο λεπτομερειακά, κάποια από τα βασικά χαρακτηριστικά του Γράφου Ροής Δεδομένων είναι τα εξής:

- **Πράξη (Operation):** Η *πράξη* αντιπροσωπεύει έναν αφηρημένο υπολογισμό π.χ. άθροισμα, γινόμενο πινάκων κλπ. Μια πράξη διαχειρίζεται τανυστές και είναι πολυμορφική δηλαδή μπορεί να δράσει πάνω σε τανυστές διαφορετικών τύπων. Για παράδειγμα, η πρόσθεση μεταξύ δύο τανυστών που έχουν ως στοιχεία ακέραιους των 32 bit (int32 tensors), μπορεί να εφαρμοστεί σε τανυστές με αριθμούς κινητής υποδιαστολής (float tensors) κ.ο.κ.

- **Πυρήνας (Kernel):** Ο πυρήνας καθορίζει την πραγματική υλοποίηση της πράξης, έχοντας διαφορετικές υλοποιήσεις (για την ίδια πράξη) ανάλογα με την συσκευή στην οποία θα εκτελεστεί (CPU ή GPU)
- **Συνεδρία (Session):** Κάθε φορά που ένα πρόγραμμα-πελάτης πρέπει να επικοινωνήσει με το TensorFlow, πρέπει να δημιουργηθεί μια συνεδρία μεταξύ αυτού του προγράμματος (συνήθως ένα πρόγραμμα σε γλώσσα Python ή σε όποια άλλη γλώσσα υποστηρίζεται αντίστοιχη διεπαφή) και του περιβάλλοντος λειτουργίας C++ του Tensorflow. Αφού η συνεδρία δημιουργηθεί για τον συγκεκριμένο πελάτη, δημιουργείται ένα αρχικός γράφος ο οποίος είναι άδειος. Η συνεδρία, που ουσιαστικά αποτελεί ένα αντικείμενο της Python (`tf.Session`), παρέχει πρόσβαση σε συσκευές στο τοπικό μηχάνημα καθώς και σε απομακρυσμένες συσκευές μέσω του κατανεμημένου περιβάλλοντος λειτουργίας του Tensorflow. Επιπλέον αποθηκεύει σε κρυφή μνήμη πληροφορίες σχετικά με τον Γράφο (αντικείμενο `tf.Graph`) παρέχοντας έτσι την δυνατότητα να τρέχουμε αποδοτικά τους ίδιους υπολογισμούς πολλές φορές.

5.2.2 Τα πλεονεκτήματα των Γράφων Ροής Δεδομένων

Το TensorFlow βελτιστοποιεί τους υπολογισμούς του βάσει της συνδεσιμότητας του γράφου. Κάθε γράφος διατηρεί το δικό του σύνολο εξαρτήσεων από κόμβους. Όταν η είσοδος του κόμβου B επηρεάζεται από την έξοδο του κόμβου A, λέμε ότι ο B είναι εξαρτώμενος από τον κόμβο A. Ονομάζουμε την εξάρτηση άμεση όταν οι δύο εξαρτώμενοι κόμβοι συνδέονται κατευθείαν με ακμή, ενώ έμμεση σε κάθε άλλη περίπτωση.

Το γεγονός ότι μπορούμε πάντα να αναγνωρίζουμε εύκολα τα σύνολα εξαρτήσεων για κάθε κόμβο σε έναν γράφο, αποτελεί βασικό χαρακτηριστικό για το πώς θα δομήσουμε τους υπολογισμούς βάσει γράφου. Η δυνατότητα εντοπισμού εξαρτήσεων μεταξύ των μονάδων του μοντέλου μας, μας επιτρέπει τόσο να διανείμουμε τους υπολογισμούς μας μεταξύ των διαθέσιμων πόρων όσο και να αποφεύγουμε την εκτέλεση περιττών υπολογισμών άσχετων υποσυνόλων, με αποτέλεσμα ταχύτερο, αποδοτικότερο και πιο αποτελεσματικό τρόπο υπολογισμού. Συνοπτικά τα πλεονεκτήματα της χρήσης Γράφου Ροής Δεδομένων, τα οποία εκμεταλλεύεται το TensorFlow κατά την εκτέλεση των προγραμμάτων είναι:

- **Παράλληλισμός:** Χρησιμοποιώντας σαφώς ορισμένες ακμές που αντιπροσωπεύουν εξαρτήσεις μεταξύ των λειτουργιών, είναι εύκολο για το σύστημα να εντοπίσει λειτουργίες που μπορούν να εκτελούνται παράλληλα.
- **Κατανεμημένη Εκτέλεση:** Χρησιμοποιώντας σαφώς ορισμένες ακμές που αντιπροσωπεύουν τις τιμές που ρέουν μεταξύ των λειτουργιών των κόμβων, είναι δυνατό για το TensorFlow να χωρίσει και να κατανείμει το πρόγραμμά σας σε πολλαπλές συσκευές (CPUs, GPUs)

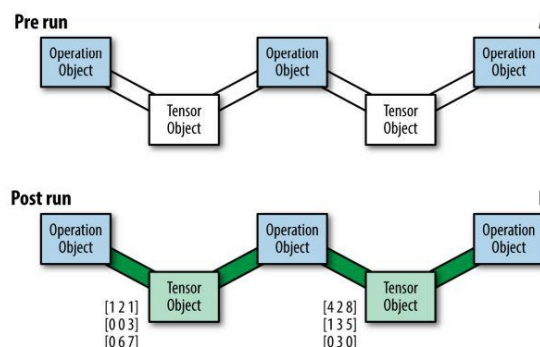
συνδεδεμένες σε διαφορετικές μηχανές. Το TensorFlow εισάγει την απαραίτητη επικοινωνία και τον συντονισμό μεταξύ των συσκευών

- **Μεταγλώττιση:** Ο μεταγλωττιστής XLA του TensorFlow μπορεί να χρησιμοποιήσει τις πληροφορίες από τον Γράφο Ροής Δεδομένων για να δημιουργήσει γρηγορότερο κώδικα, όπως για παράδειγμα, συγχωνεύοντας γειτονικές λειτουργίες.
- **Φορητότητα:** Ο Γράφος Ροής Δεδομένων είναι μια αναπαράσταση ανεξάρτητη από την γλώσσα του κώδικα που έχει χρησιμοποιηθεί για την δημιουργία του μοντέλου μας. Παρέχεται έτσι η δυνατότητα να δημιουργήσουμε ένα Γράφος Ροής Δεδομένων στη Python και στην συνέχεια να το αποθηκεύσουμε και να το επαναφέρουμε σε ένα πρόγραμμα C++.

5.3 Τανυστές, Γράφοι, Συνεδρίες

Όταν κατασκευάζουμε έναν κόμβο του γράφου, στην πραγματικότητα δημιουργούμε ένα στιγμιότυπο μιας πράξης. Αυτές οι πράξεις δεν παράγουν πραγματικές τιμές (μέχρι την στιγμή που ο γράφος θα εκτελεστεί), αλλά αναφορές στα μελλοντικά αποτελέσματα που πρόκειται να υπολογίσουν, τις οποίες παρέχουν σαν «σκυτάλη» στην είσοδο των επόμενων κόμβων. Το TensorFlow έχει σχεδιαστεί έτσι ώστε να δημιουργείται ο αρχικός σκελετός του γράφου με όλα τα συστατικά του και μέχρι το σημείο εκείνο να μην ρέει καμία πληροφορία ή να μην εκτελείται κανένας υπολογισμός. Μόνο κατά την εκτέλεση, όταν δημιουργούμε τη *συνεδρία*, τα δεδομένα εισέρχονται στο γράφημα ως *τανυστές* και πραγματοποιούνται οι υπολογισμοί. Με αυτόν τον τρόπο, οι υπολογισμοί μπορούν να είναι πολύ πιο αποδοτικοί, λαμβάνοντας υπόψη ολόκληρη τη δομή γραφημάτων.

Σε αυτήν την ενότητα θα περιγράψουμε και θα εξηγήσουμε του τανυστές και στην συνέχεια θα δημιουργήσουμε έναν απλό γράφο που υλοποιεί μια στοιχειώδη αριθμητική παράσταση και θα τον εκτελέσουμε. Η υλοποίηση θα γίνει στην γλώσσα Python, η οποία αποτελεί και την γλώσσα με την πιο δημοφιλή και πλούσια προγραμματιστική διεπαφή (API) όσον αφορά το TensorFlow.

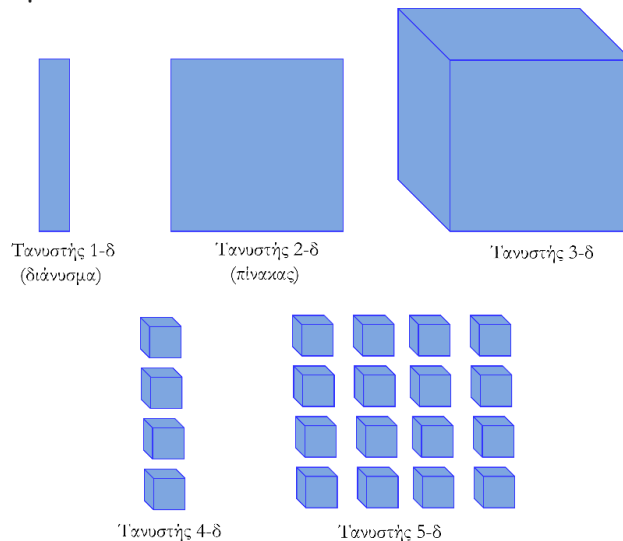


Εικόνα 27 Αφηρημένη απεικόνιση του γράφου, πριν και μετά την εκτέλεση της συνεδρίας.

5.3.1 Τανυστές (Tensors)

Οι τανυστές είναι η βασική δομή δεδομένων στο Tensorflow. Όπως έχει ήδη αναφερθεί, οι τανυστές αναπαριστούν τις ακμές συνδέσεων στο Γράφο Ροής Δεδομένων ή διαφορετικά, την πληροφορία που ρέει μέσα στις ακμές. Ένας τανυστής ουσιαστικά αποτελεί έναν πολυδιάστατο πίνακα ή λίστα και προσδιορίζεται από τρία βασικά χαρακτηριστικά: τον βαθμό (rank), το σχήμα (shape) και το τύπο (type).

- Η διάσταση κάθε τανυστή περιγράφεται από μια μονάδα η οποία ονομάζεται **βαθμός**. Έτσι ένας απλός διδιάστατος πίνακας, είναι ένας τανυστής με βαθμό 2, ενώ ένα διάνυσμα είναι ένα τανυστής με βαθμό 1.
- Το **σχήμα** ενός τανυστή περιγράφει το μέγεθος της κάθε διάστασης του. Για παράδειγμα, σε έναν πίνακα 2 διαστάσεων, δηλώνει το πλήθος των γραμμών και των στηλών που έχει ο πίνακας αυτός.
- Τέλος, ο **τύπος** του τανυστή, αποτελεί τον τύπο των στοιχείων (int32, float κλπ.) που περιέχει ο τανυστής.



Εικόνα 28 Γραφική απεικόνιση τανυστών με μία έως πέντε διαστάσεις

5.3.2 Σταθερές, Μεταβλητές και Placeholder

Το TensorFlow προκειμένου να μπορεί να εισάγει δεδομένα μέσα στον γράφο καθώς και να τα αποθηκεύσει χρησιμοποιεί τρεις τύπους κόμβων καθένας με διαφορετική λειτουργία και χρησιμότητα. Οι κόμβοι αυτοί είναι οι εξής:

- **tf.constant:** η λειτουργία του κόμβου αυτού είναι παρόμοια με την λειτουργία της σταθεράς σε ένα πρόγραμμα. Ο κόμβος αυτός κατά την δημιουργία του, αρχικοποιείται με μια τιμή η οποία παραμένει σταθερή από την αρχή μέχρι το τέλος λειτουργίας του γράφου.

Ένα παράδειγμα δημιουργίας ενός κόμβου constant:

```
>>> a = tf.constant(<αρχική-τιμή>, dtype=<προαιρετικός τύπος>)
```

- **tf.placeholder:** η λειτουργία του κόμβου αυτού είναι να τροφοδοτεί δεδομένα στον γράφο. Για παράδειγμα, κατά την εκπαίδευση ενός νευρωνικού δικτύου, ο κόμβος αυτός έχει σαν ρόλο να δέχεται τα δεδομένα από το σύνολο εκπαίδευσης και να τα προωθεί μέσα στον γράφο. Αυτό επιτυγχάνεται καθώς όπως κάθε τανυστής, έτσι και ο κόμβος placeholder μπορεί να χρησιμοποιηθεί ως είσοδος σε επόμενο κόμβο. Κατά την κατασκευή του, θα πρέπει υποχρεωτικά να δηλώνεται ο τύπος των δεδομένων που δέχεται.

```
>>> a = tf.placeholder(dtype=<τύπος_δεδομένων>)
```

- **tf.Variable:** Ο κόμβος αυτός, όπως και μια μεταβλητή σε ένα πρόγραμμα, αποθηκεύει και διατηρεί δεδομένα, τα οποία μπορούν να μεταβάλλονται κατά την διάρκεια εκτέλεσης του γράφου. Η προσθήκη μιας μεταβλητής στον γράφο γίνεται με την κατασκευή ενός στιγμιότυπου της κλάσης Variable του TensorFlow. Κατά την κατασκευή της, απαιτείται η αρχικοποίηση της με μία τιμή, η οποία μπορεί να είναι ένας τανυστής οποιουδήποτε τύπου και σχήματος.

```
>>> a = tf.Variable(<αρχική-τιμή>, name=<προαιρετικό-όνομα>)
```

Μετά την κατασκευή της, ο τύπος και το σχήμα της μένουν σταθερά και δεν μπορούν να μεταβληθούν. Η τιμή της μπορεί να αλλάξει χρησιμοποιώντας την μέθοδο assign ή κάποια παρόμοια μέθοδο. Όπως κάθε τανυστής, έτσι και οι μεταβλητές που έχουν κατασκευαστεί με την κλάση Variable, μπορούν να χρησιμοποιηθούν και ως είσοδοι σε επόμενους κόμβους.

5.3.3 Δημιουργία Γράφου

Πριν ξεκινήσουμε να δημιουργούμε οποιαδήποτε μορφή γράφου θα πρέπει πρώτα στο πρόγραμμά μας να εισάγουμε την βιβλιοθήκη του TensorFlow για την Python.

```
>>> import tensorflow as tf
```

Αμέσως μετά την εισαγωγή της βιβλιοθήκης, αρχικοποιείται ένας προκαθορισμένος γράφος, ο οποίος αρχικά είναι κενός. Ότι κόμβους δημιουργούμε στην συνέχεια θα συσχετίζονται αυτόματα και θα προστίθενται σε αυτόν τον γράφο.

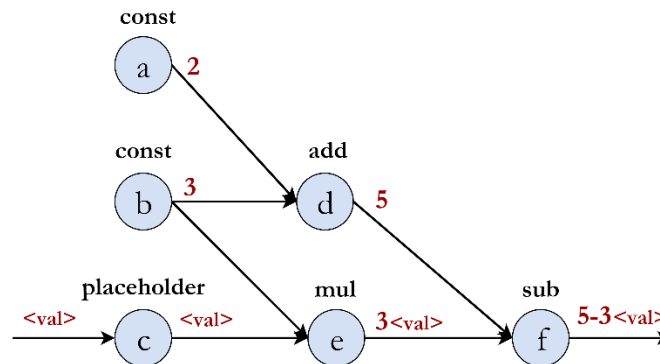
Αρχικά, με την βοήθεια των τύπων κόμβων που παρουσιάσαμε παραπάνω για την διαχείριση των δεδομένων στο γράφο, θα δημιουργήσουμε τρεις κόμβους στους οποίους θα αναθέσουμε τιμές και θα τους συνδέσουμε με επόμενους κόμβους που θα παρουσιαστούν παρακάτω:

```
>>> a = tf.constant(2, dtype=tf.int32)
>>> b = tf.constant(3, dtype=tf.int32)
>>> c = tf.placeholder(dtype=tf.int32, shape=[1])
```

Εδώ, οι κόμβοι *a* και *b* έχουν σταθερή τιμή, ενώ ο *c* είναι placeholder και θα λάβει την είσοδο του κατά την εκτέλεση του γράφου.

Στην συνέχεια, καθένας από τους επόμενους κόμβους παίρνει ως είσοδο την έξοδο κάποιου από τους προηγούμενους κόμβους, δομώντας έτσι έναν γράφο που αναπαριστά μια απλή αριθμητική παράσταση:

```
>>> d = tf.add(a,b)
>>> e = tf.multiply(b,c)
>>> f = tf.subtract(d,e)
```



Εικόνα 29 Παράδειγμα γράφου

5.3.4 Δημιουργία και εκτέλεση Συνεδρίας

Μετά την ολοκλήρωση της περιγραφής και της κατασκευής του γράφου, είμαστε έτοιμοι να εκτελέσουμε του υπολογισμούς που αναπαριστώνται από αυτόν. Για τον σκοπό αυτό θα πρέπει

να δημιουργήσουμε μια *συνεδρία* και στην συνέχεια να την εκτελέσουμε. Αυτό το επιτυγχάνουμε με τον παρακάτω κώδικα:

```
>>> sess = tf.Session()
>>> out = sess.run(f, {c:7})
>>> sess.close()
>>> print(out)
>>> -16
```

Ένα αντικείμενο `Session` είναι κομμάτι της προγραμματιστικής διεπαφής (API) του Tensorflow το οποίο δημιουργεί την επικοινωνία μεταξύ των αντικειμένων της Python μαζί με τα δεδομένα που έχουμε από τη δικιά μας πλευρά και του πραγματικού υπολογιστικού συστήματος στο οποίο έχει δεσμευτεί η μνήμη για τα αντικείμενα που εμείς ορίσαμε, έχουν αποθηκευτεί ενδιάμεσες μεταβλητές και από το οποίο τελικά μας επιστρέφονται τα αποτελέσματα. Η *συνεδρία* αφού δημιουργηθεί, στην συνέχεια εκτελείται καλώντας την μέθοδο `.run()`. Εκεί είναι που εισάγεται ο κόμβος του γράφου του οποίου επιθυμούμε να υπολογίσουμε την έξοδο, καθώς και τα δεδομένα που θέλουμε να τροφοδοτήσουμε στον γράφο, στην περίπτωση που στο σύνολο εξαρτήσεων υπάρχει και κόμβος `placeholder`. Όταν κληθεί, η μέθοδος αυτή υπολογίζει ένα σύνολο υπολογισμών στο γράφο με τον ακόλουθο τρόπο: ξεκινάει από τον κόμβο του οποίου η έξοδος ζητήθηκε και στην συνέχεια με βάση τις εξαρτήσεις μεταξύ των κόμβων, ξεδιπλώνει προς τα πίσω τον γράφο καθορίζοντας το ποιος κόμβος πρέπει να υπολογιστεί. Στην περίπτωση μας, ζητήσαμε να υπολογιστεί ο κόμβος `f` όταν στον κόμβο `c` δίνουμε τον ακέραιο αριθμό 7 μέσω της εντολής: `sess.run(f, {c:7})`. Αφού ολοκληρωθούν οι υπολογισμοί μας, είναι καλή πρακτική να κλείνουμε την *συνεδρία* προκειμένου να αποδεσμεύουμε του πόρους του συστήματος.

ΚΕΦΑΛΑΙΟ 6

ΠΡΟΣΕΓΓΙΣΗ ΠΡΟΒΛΗΜΑΤΟΣ

Με βάση το θεωρητικό υπόβαθρο που θεμελιώθηκε στα προηγούμενα κεφάλαια πάνω στα νευρωνικά δίκτυα, σε αυτό και στο επόμενο το κεφάλαιο θα παρουσιαστούν και θα αναλυθούν η προσέγγιση και η υλοποίηση που θα ακολουθήσει η παρούσα διπλωματική, για την ανάλυση και την ταξινόμηση του συναισθήματος ενός κειμένου.

6.1 Αναδρομικά Νευρωνικά Δίκτυα και Ανάλυση Συναισθήματος

Όπως έχει αναφερθεί σε προηγούμενο κεφάλαιο, ένα πολυεπίπεδο δίκτυο perceptron μπορεί να απεικονίζει μόνο από ένα διάνυσμα εισόδου σε ένα διάνυσμα εξόδου, ενώ ένα αναδρομικό δίκτυο, μπορεί κατά κύριο λόγο να απεικονίζει ολόκληρο το ιστορικό των εισόδων του σε κάθε έξοδο. Το κλειδί για αυτήν την ιδιότητα είναι οι συνδέσεις ανάδρασης ενός τέτοιου δικτύου, οι οποίες επιτρέπουν στο να συγκρατείται ένα είδος «μνήμης» των προηγούμενων εισόδων στην εσωτερική κατάσταση του δικτύου και επομένως να επηρεάζεται ανάλογα η έξοδος του.

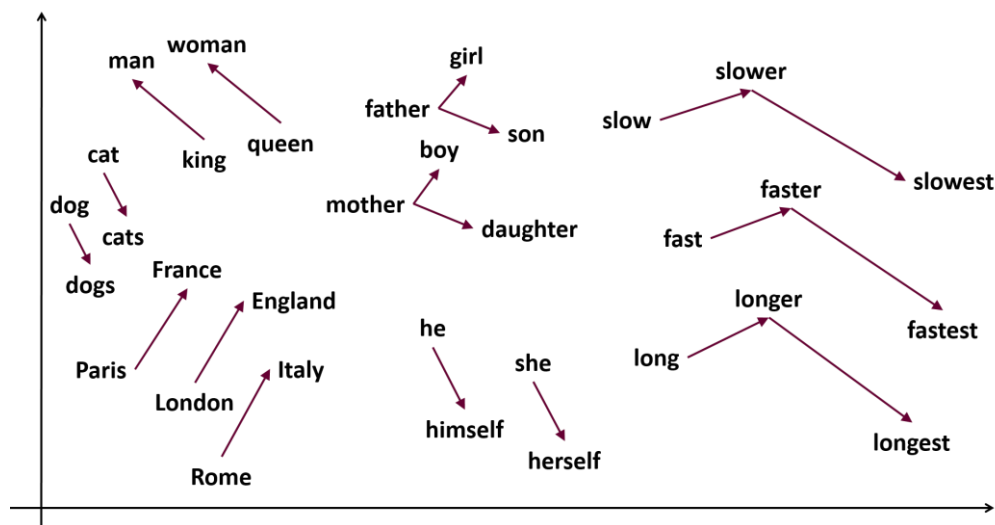
Παρόμοια ακολουθιακή συμπεριφορά παρουσιάζει και ο ανθρώπινος εγκέφαλος. Ο άνθρωπος κατά την ομιλία του δεν ξεκινάει να σκέφτεται για κάθε λέξη από την αρχή. Αντίστοιχα, όταν διαβάζει μια πρόταση αντιλαμβάνεται το νόημα της, βασισμένος στην κατανόηση των λέξεων που έχει ήδη διαβάσει.

Στο πρόβλημα ταξινόμησης του συναισθήματος μιας πρότασης ή και ενός κειμένου, μια δομή που μπορεί να προσεγγίσει την ακολουθιακή λειτουργία της ανθρώπινη σκέψης για την κατανόηση του συναισθήματός της, όπως ένα αναδρομικό νευρωνικό δίκτυο, μπορεί να έχει σημαντικά πλεονεκτήματα και κατ' επένταση μεγάλη αποτελεσματικότητα σε αυτό το πρόβλημα ταξινόμησης. Επιπλέον, όταν θέλουμε να εξάγουμε το συναισθηματικό από ένα κείμενο μεγάλου μήκους, το αναδρομικό δίκτυο που θα χρησιμοποιήσουμε θα πρέπει να έχει μνήμη αρκετά μεγάλη, ώστε να διατηρεί εξαρτήσεις ακόμα και μεταξύ λέξεων που βρίσκονται σε μακρινές θέσεις, προκειμένου να συγκρατεί όσο τον δυνατόν περισσότερο νόημα από το κείμενο αυτό.

Στην βάση των όσων αναφέρθηκαν παραπάνω, η προσέγγιση της παρούσας διπλωματικής, θα βασιστεί σε αρχιτεκτονικές από αναδρομικά δίκτυα μακράς βραχυπρόθεσμης μνήμης προκειμένου να εξεταστεί η απόδοση και η αποτελεσματικότητά τους.

6.2 Αναπαράσταση Λέξεων Στον Διανυσματικό Χώρο

Πρώτο βήμα για την επίλυση του προβλήματος ταξινόμησης του συναισθήματος ενός κειμένου, είναι ο τρόπος με τον οποίο θα αναπαρασταθούν οι λέξεις, προκειμένου αυτές να δοθούν ως είσοδος στο εκάστοτε σύστημα με μορφή που αυτό να μπορεί να τις καταλαβαίνει. Οι λέξεις θα πρέπει να λάβουν τέτοια κωδικοποίηση-αναπαράσταση, ώστε να γίνονται αντιληπτές από το σύστημα και να μπορούν να επεξεργαστούν από αυτό, ενώ ταυτόχρονα, η αναπαράσταση αυτή να μπορεί να αναδείξει τις σημασιολογικές διαφορές και ομοιότητες μεταξύ των λέξεων. Μια γενικευμένη μοντελοποίηση της φυσικής γλώσσας η οποία να γίνεται κατανοητή από ένα σύστημα, σε παρόμοιο βαθμό που την αντιλαμβάνεται ο άνθρωπος είναι σχεδόν αδύνατη να πραγματοποιηθεί. Ωστόσο μπορεί να γίνει μια μερική μοντελοποίηση των συνιστωσών της. Μια από τις πιο συνηθισμένες και καλύτερες μοντελοποιήσεις, ειδικά όταν πρόκειται για εισόδους σε νευρωνικά δίκτυα, είναι η αναπαράσταση των λέξεων στον διανυσματικό χώρο. Συγκεκριμένα, η κάθε λέξη περιγράφεται από ένα διάνυσμα πραγματικών αριθμών n διαστάσεων. Τα διανύσματα είναι διαμορφωμένα με τέτοιο τρόπο ώστε λέξεις που σχετίζονται σημασιολογικά μεταξύ τους, να αναπαριστώνται και από διανύσματα που έχουν αντίστοιχα κοντινές τιμές στις διαστάσεις τους.



Εικόνα 30 Παράδειγμα αναπαράστασης λέξεων ως διανύσματα

Τα διανύσματα αυτά είναι δυνατό να βελτιστοποιηθούν σε τέτοιο βαθμό ώστε να μπορούν να αναδείξουν σημασιολογικές συσχετίσεις μεταξύ σημασιολογικά κοντινών λέξεων ακόμα και μέσω των μεταξύ τους πράξεων. Μερικά σχετικά παραδείγματα είναι $\text{France} - \text{Italy} + \text{Rome} = \text{Paris}$ καθώς και $\text{King} - \text{Man} + \text{Woman} = \text{Queen}$, $\text{software} - \text{building} + \text{architect} = \text{programmer}$.

Η αναπαράσταση των λέξεων μέσω διανυσμάτων έχει αποδειχθεί πολύ χρήσιμη για κάθε είδους προβλήματος επεξεργασία φυσικής γλώσσας και κατά συνέπεια και για το πρόβλημα ταξινόμησης συναισθήματος κειμένου. Για την δημιουργία των διανυσμάτων υπάρχουν αρκετές προσεγγίσεις. Στα πλαίσια της διπλωματικής αυτής, θα χρησιμοποιηθεί η μέθοδος GloVe (Global Vectors for Word Representation) που έχει αναπτυχθεί από το πανεπιστήμιο του Stanford.

6.2.1 GloVe (Global Vectors for Word Representation)

Η μέθοδος GloVe είναι ένας αλγόριθμος μάθησης χωρίς επίβλεψη για την δημιουργία διανυσματικών αναπαραστάσεων για λέξεις. Το μοντέλο GloVe μαθαίνει διανύσματα λέξεων εξετάζοντας λέξεις που συν-εμφανίζονται μέσα στο σώμα ενός κειμένου.

Πριν εκπαιδευσουμε το πραγματικό μοντέλο, χρειάζεται να κατασκευάσουμε έναν πίνακα συν-εμφάνισης X , όπου κάθε στοιχείο $X_{i,j}$ είναι ένας αριθμός που αντιπροσωπεύει την συχνότητα με την οποία η λέξη i εμφανίζεται στην γειτονιά της λέξης j . Η γειτονιά κάθε λέξης αντιπροσωπεύεται από ένα παράθυρο n θέσεων. Για να χτίσουμε τον πίνακα X διατρέχουμε μια φορά όλο το κείμενο και στη συνέχεια χρησιμοποιούμε αυτά τα δεδομένα συν-εμφάνισης στη θέση του πραγματικού κειμένου. Το μοντέλο μας το κατασκευάζουμε με βάση μόνο τις τιμές που συλλέχθηκαν στον πίνακα X . Αφού δημιουργήσουμε τον πίνακα X , τα διανύσματα τα παράγουμε θέτοντας αρχικά τον χαλαρό περιορισμό:

$$\vec{w}_i^T \vec{w}_j + b_i + b_j = \log X_{ij}$$

όπου b_i , b_j είναι διανύσματα πόλωσης για τις λέξεις i, j αντίστοιχα. Διαισθητικά τα διανύσματα θέλουμε να φτιαχτούν διατηρώντας την χρήσιμη πληροφορία για το πώς κάθε ζευγάρι λέξεων i, j εμφανίζεται μέσα στο κείμενο. Αυτό επιτυγχάνεται ελαχιστοποιώντας την αντικειμενική συνάρτηση J , η οποία υπολογίζει το άθροισμα των τετραγώνων των σφαλμάτων βασισμένη στον παραπάνω περιορισμό βεβαρυνόμενο με μια συνάρτηση f :

$$J = \sum_i \sum_j f(X_{ij})(\overline{w}_i^T \overline{w}_j + b_i + b_j - \log X_{ij})^2$$

Η συνάρτηση f λειτουργεί ως βάρος στην αντικειμενική συνάρτηση και έχει ως ρόλο, να προστατέψει την αλλοίωση της από ζευγάρια λέξεων με μεγάλη συχνότητα συν-εμφάνισης, δηλαδή με πολύ υψηλές τιμές X_{ij} .

Για τον ρόλο αυτό επιλέγεται η εξής συνάρτηση:

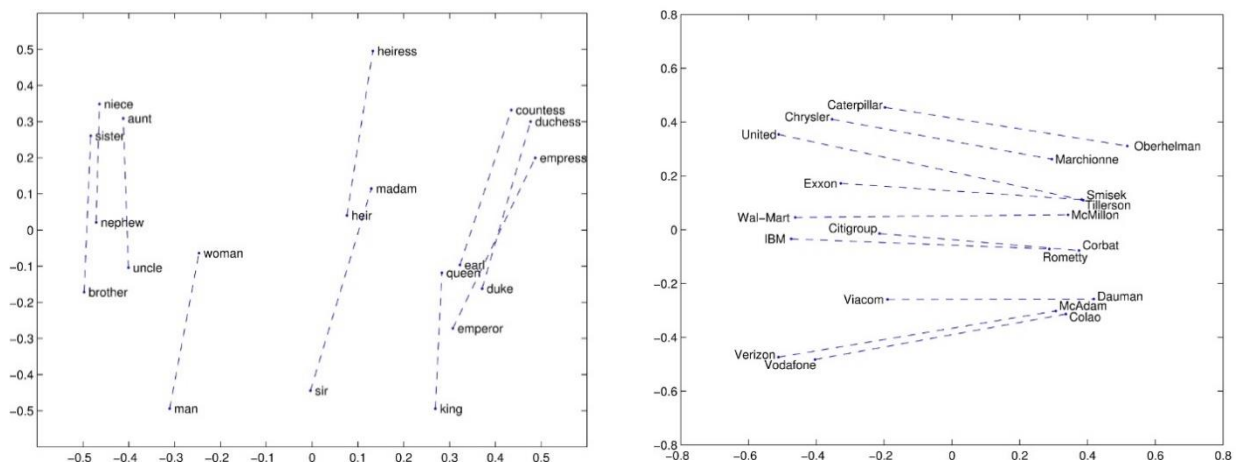
$$f(X_{ij}) \begin{cases} \left(\frac{X_{ij}}{x_{max}}\right)^a, & \text{αν } X_{ij} < x_{max} \\ 1, & \text{διαφορετικά} \end{cases}$$

Γνωρίζοντας την αντικειμενική συνάρτηση, θα υπολογιστούν κατάλληλα οι κλίσεις της J ως προς τις παραμέτρους $\overline{w}_i, \overline{w}_j, b_i, b_j$ προκειμένου αυτές να ανανεωθούν κατάλληλα.

Η αποτελεσματικότητα με την οποία το GloVe αναπαριστά τις λέξεις στον διανυσματικό χώρο αποτυπώνεται στις εξής ιδιότητες:

- Η Ευκλείδεια απόσταση μεταξύ δύο λέξεων- διανυσμάτων παρέχει μια αποτελεσματική μέθοδο για την μέτρηση και ανάδειξη της γλωσσικής ή νοηματικής ομοιότητας των λέξεων αυτών. Βρίσκοντας έτσι του κοντινούς γείτονες κάθε λέξης με βάση αυτήν την μετρική μπορεί να αναδειχθεί η νοηματική συγγένεια και μεταξύ λέξεων που απουσιάζουν από το καθημερινό λεξιλόγιο των ανθρώπων και ανήκουν σε μια πιο επιστημονική ορολογία.
- Οι μετρικές ομοιότητας που χρησιμοποιούνται για την εύρεση των γειτονικών λέξεων, παράγουν έναν μόνο αριθμό που ποσοτικοποιεί την ομοιότητα τους. Υπάρχουν ωστόσο περιπτώσεις, όπου η ποσοτικοποίηση της ομοιότητας δύο λέξεων-διανυσμάτων, τείνει να γίνει προβληματική καθώς τις περισσότερες φορές, οι λέξεις επιδεικνύουν πιο σύνθετες σχέσεις που δεν μπορούν να εκφραστούν απλά με έναν αριθμό. Ένα παράδειγμα είναι οι λέξεις άντρας, γυναίκας, οι οποίες μπορούν να χαρακτηριστούν τόσο ως όμοιες (μικρή απόσταση μεταξύ τους) καθώς περιγράφουν έναν άνθρωπο, όσο και ως αντίθετες αφού αντιπροσωπεύουν διαφορετικά φύλα. Προκειμένου να εκφραστεί και να ποσοτικοποιηθεί αυτή η διαφορά μεταξύ των δύο λέξεων απαιτείται ένα μοντέλο που να συνδυάζει περισσότερους από έναν αριθμούς για κάθε ζευγάρι. Ένας φυσικός και απλός υποψήφιος για ένα διευρυμένο σύνολο διακριτών αριθμών είναι η διανυσματική διαφορά μεταξύ των δυο λέξεων. Το GloVe έχει σχεδιαστεί έτσι ώστε οι διαφορές μεταξύ των διανυσμάτων να καταγράφουν όσο το δυνατόν περισσότερο το νόημα που δίνεται από την αντιπαράθεση

δύο λέξεων. Η υποκείμενη έννοια που διακρίνει τον άνθρωπο από τη γυναίκα, δηλαδή το φύλο, μπορεί να προσδιοριστεί ισοδύναμα από διάφορα άλλα ζεύγη λέξεων, όπως ο βασιλιάς και η βασίλισσα ή ο αδελφός και η αδελφή. Για να δηλωθεί αυτή την παρατήρηση μαθηματικά, θα περιμέναμε ότι οι διαφορές διανυσμάτων άνδρας - γυναίκας, βασιλιάς - βασίλισσας και αδελφής - αδελφής θα ήταν ίσως ίσες. Αυτή η ιδιότητα και άλλα ενδιαφέροντα πρότυπα μπορούν να παρατηρηθούν στο παραπάνω σύνολο οπτικοποιήσεων από τα αποτελέσματα του GloVe.



Εικόνα 31 Αποτελέσματα αναπαράτασης της μεθόδου GloVe

6.3 Επιλεγμένες Αρχιτεκτονικές Βαθιάς Μάθησης με Αναδρομικά Νευρωνικά Δίκτυα

Στην προηγούμενη ενότητα αναλύθηκε η μέθοδος που επιλέξαμε για να αναπαραστήσουμε τις λέξεις προκειμένου αυτές να δοθούν ως είσοδος σε ένα νευρωνικό δίκτυο. Στην παρούσα ενότητα θα περιγράψουμε τα μοντέλα βαθιάς μάθησης νευρωνικών δικτύων τα οποία επιλέχθηκαν να υλοποιηθούν στα πλαίσια αυτής της διπλωματικής προκειμένου να εξεταστεί η απόδοση τους στο πρόβλημα της συναισθηματικής ταξινόμησης κειμένων. Στην γενική τους εικόνα, κάθε μοντέλο δέχεται ως είσοδο μια ακολουθία από τις λέξεις ενός κειμένου και δίνει ως έξοδο μια κατανομή πιθανότητας που προσδιορίζει το ποσοστό σύμφωνα με το οποίο αυτή η ακολουθία να ανήκει σε καθεμία από τις κατηγορίες: θετικό, αρνητικό σχόλιο.

6.3.1 Επίπεδα Αρχιτεκτονικής

Η δομή των μοντέλων αυτών, ως μοντέλα βαθιάς μάθησης, αποτελείται από μια αλληλουχία λειτουργικών στρωμάτων-επιπέδων καθένα από τα οποία επιτελεί μια συγκεκριμένη λειτουργία. Αναλόγως με την ελάχιστη αρχιτεκτονική που ακολουθείται, τα επίπεδα αυτά ενώνονται μεταξύ τους με μια προκαθορισμένη σειρά, προκειμένου στο τέλος να δημιουργήσουν το τελικό μοντέλο που θα ταξινομή το συναίσθημα ενός κειμένου.

Στις παρούσες αρχιτεκτονικές που έχουν επιλεγθεί, μπορούν να παρατηρηθούν τρία βασικά λειτουργικά επίπεδα: το επίπεδο του αναδρομικού δικτύου LSTM, το softmax επίπεδο και το pooling επίπεδο.

6.3.1.1 Επίπεδο LSTM

Το επίπεδο LSTM, είναι ουσιαστικά το επίπεδο το οποίο περιλαμβάνει το αναδρομικό δίκτυο που χρησιμοποιείται στο μοντέλο μας.

Όπως είχε αναφερθεί παραπάνω, στόχος είναι να αξιοποιηθούν οι ιδιότητες των αναδρομικών δικτύων σε εισόδους με ακολουθιακό χαρακτήρα αλλά ταυτόχρονα να διατηρούνται και οι μακρινές εξαρτήσεις που παρουσιάζονται μεταξύ τους. Στην βάση των παραπάνω, τα αναδρομικά δίκτυα που θα συμπεριληφθούν στο μοντέλο μας, θα χρησιμοποιούν στο κρυφό τους επίπεδο το κύτταρο μακράς βραχυπρόθεσμης μνήμης (LSTM).

Το δίκτυο μακράς βραχυπρόθεσμης μνήμης, μπορεί να θεωρηθεί ως το επίπεδο εισόδου του μοντέλου μας, καθώς είναι αυτό στο οποίο τροφοδοτούνται οι λέξεις του ελάχιστου κειμένου. Πιο συγκεκριμένα, η κάθε λέξη αναπαρασταίνεται ως διάνυσμα σύμφωνα με την διανυσματική αναπαράσταση του GloVe και τροφοδοτείται στο δίκτυο, με την σειρά που εμφανίζεται μέσα

στο κείμενο. Έτσι κάθε επόμενη λέξη αποτελεί και την επόμενη χρονικά είσοδο του δικτύου μακράς βραχυπρόθεσμης μνήμης.

Στο επίπεδο LSTM, ως αυτοτελές λειτουργικό επίπεδο της γενικής αρχιτεκτονικής, το αναδρομικό δίκτυο μακράς βραχυπρόθεσμης μνήμης εμφανίζεται με παραλλαγές στην δομή του προκειμένου να εξυπηρετεί τις ελάχιστες ανάγκες. Έτσι, μπορεί να εμφανιστεί με μία ή περισσότερες εξόδους καθώς και ως απλό ή αμφίδρομο (BLSTM) αναδρομικό δίκτυο.

6.3.1.2 Επίπεδο Softmax

Στα προβλήματα ταξινόμησης που καλούνται να λύσουν τα νευρωνικά δίκτυα (όπως είναι και το πρόβλημα που έχουμε να επιλύσουμε στην παρούσα διπλωματική), θα πρέπει να βρούμε μια μοντελοποίηση της εξόδου που να περιγράφει τα αποτελέσματα αυτής της ταξινόμησης. Το επίπεδο Softmax, όπως θα αναλυθεί και παρακάτω, αποτελείται από μια συστοιχία νευρώνων με συνάρτηση ενεργοποίησης την συνάρτηση softmax. Αυτό το επίπεδο είναι και η έξοδος του μοντέλου μας, το οποίο επιτυγχάνει να δημιουργήσει μια κατηγορική κατανομή για τις κατηγορίες που περιλαμβάνει το πρόβλημα μας.

Εν συντομία, μια κατηγορική κατανομή είναι η γενίκευση της κατανομής Bernoulli σε πολλές κλάσεις-κατηγορίες. Η κατανομή Bernoulli είναι μια διακριτή κατανομή πιθανότητας που μοντελοποιεί την έκβαση ενός μοναδικού πειράματος ή μιας απλής παρατήρησης μιας τυχαίας μεταβλητής με δύο αποτελέσματα. Η κατηγορική κατανομή, επεκτείνει φυσικά την κατανομή Bernoulli και αποτελεί μια διακριτή πιθανοτική κατανομή που περιγράφει τα πιθανά αποτελέσματα μια τυχαίας μεταβλητής που μπορεί να ανήκει σε K κατηγορίες, προσδιορίζοντας συγκεκριμένα την πιθανότητα με την οποία μπορεί να ανήκει σε καθεμία από αυτές τις κατηγορίες. Η συνάρτηση softmax ορίζεται ως εξής:

$$\sigma z = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$$

όπου $j = 1 \dots K$. Όπως φαίνεται, η λειτουργία της συνάρτησης αυτής είναι να δέχεται ένα διάνυσμα z , K διαστάσεων με αυθαίρετες πραγματικές τιμές και να το κανονικοποιεί σε ένα διάνυσμα πιθανοτήτων σz του οποίου τα στοιχεία έχουν άθροισμα ίσο με 1.

Με βάση τα παραπάνω, ένα νευρωνικό δίκτυο που ειτελεί πιθανοτική ταξινόμηση (probabilistic classification) μεταξύ K κατηγοριών, θα έχει επίπεδο εξόδου που αποτελείται από K νευρώνες και οι οποίοι θα έχουν ως συνάρτηση ενεργοποίησης την συνάρτηση softmax. Έτσι καθένας από τους νευρώνες αυτούς θα παράγει ως έξοδο και μια διαφορετική πιθανότητα p , με $0 \leq p \leq 1$, η

οποία θα αντιπροσωπεύει την πιθανότητα με την οποία η ταξινομούμενη οντότητα ανήκει στην εκάστοτε κατηγορία.

Για την διαδικασία της εκπαίδευσης νευρωνικών δικτύων με επίπεδο softmax ως έξοδο, θα πρέπει να χρησιμοποιηθεί κατάλληλη συνάρτηση σφάλματος. Η συνάρτηση αυτή, η οποία έχει χρησιμοποιηθεί ως συνάρτηση σφάλματος και στα παρόντας μοντέλα, είναι η συνάρτηση Cross-Entropy, και το σφάλμα υπολογίζεται ως εξής:

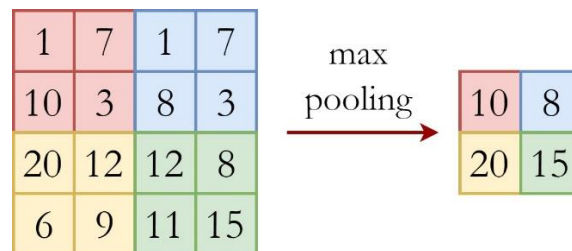
$$H(y, \hat{y}) = -y \log y - (1 - y) \log(1 - y)$$

6.3.1.3 Επίπεδο Pooling

Pooling είναι μία τεχνική που χρησιμοποιείται συχνά σε δίκτυα βαθιάς μάθησης και κυρίως στα συνελικτικά δίκτυα (convolutional networks). Η βασική του λειτουργία είναι να εφαρμόζεται πάνω σε ταχυστές με μεγάλο πλήθος στοιχείων, με σκοπό να μειώσει το μέγεθος των διαστάσεών τους, οδηγώντας σε αποδοτικότερους και πιο γρήγορους υπολογισμούς, διατηρώντας ταυτόχρονα όμως την συνεκτικότητα της πληροφορίας του αρχικό ταχυστή.

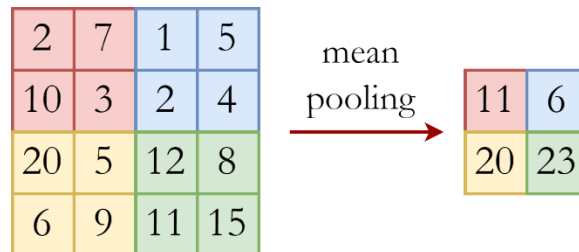
Στην περίπτωση των δύο διαστάσεων, η τεχνική pooling κάνει χρήση ενός πίνακα με διαστάσεις P_h (ύψος), P_w (πλάτος), γνωστό και ως φίλτρο pooling. Το φίλτρο αυτό, το ολισθαίνει στην κατεύθυνση της διάστασης του πλάτους, κατά S (strides) θέσεις την φορά, πάνω στον ζητούμενο πίνακα A διαστάσεων H, W εφαρμόζοντας μια διαδικασία που μοιάζει με δειγματοληψία του αρχικού πίνακα. Η διαδικασία αυτή κατά κύριο λόγο μπορεί να είναι:

- συγκράτηση του μεγαλύτερου από τα στοιχεία του πίνακα A , τα οποία ταυτόχρονα ορίζονται και περιλαμβάνονται από το φίλτρο pooling. Η διαδικασία αυτή είναι γνωστή ως max pooling



Εικόνα 32 Max-pooling φίλτρου 2x2 με 2 strides

- εξαγωγή της μέσης τιμής από τα στοιχεία του πίνακα A , τα οποία ταυτόχρονα ορίζονται και περικλείονται από το φίλτρο pooling. Η διαδικασία αυτή είναι γνωστή ως mean pooling ή average pooling.



Εικόνα 33 Mean-pooling φίλτρου 2×2 με 2 strides

Ο πίνακας που δημιουργείται ανάλογα με την εκάστοτε τεχνική pooling που χρησιμοποιείται, έχει νέες διαστάσεις οι οποίες προκύπτουν ως εξής:

$$W_{new} = \frac{W - P_w}{S} + 1$$

$$H_{new} = \frac{H - P_h}{S} + 1$$

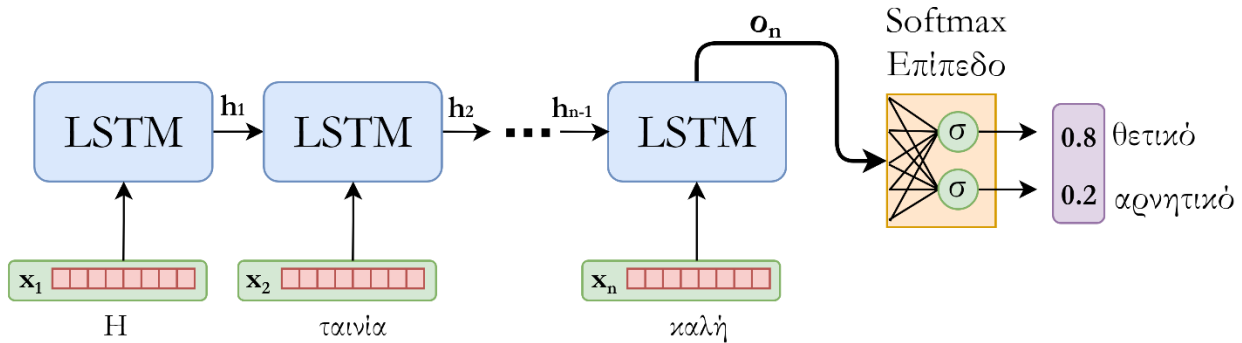
6.2.2 Αρχιτεκτονικές

Με βάση τα λειτουργικά επίπεδα που αναλύθηκαν, παρακάτω θα παρουσιαστούν οι αρχιτεκτονικές των μοντέλων βαθιάς μάθησης που προτείνει και υλοποιεί η παρούσα διπλωματική.

6.3.2.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ I

Η αρχιτεκτονική αυτή, αποτελεί την πιο απλή προσέγγιση βάση της οποίας το μοντέλο μας αποτελείται από το LSTM επίπεδο που περιέχει το αναδρομικό δίκτυο μακράς βραχυπρόθεσμης μνήμης και το επίπεδο softmax. Σε κάθε χρονικό βήμα του αναδρομικού δικτύου, του δίνεται ως είσοδο η επόμενη σε σειρά λέξη από το εξεταζόμενο κείμενο. Αφού το κείμενο έχει «διαβαστεί» από το δίκτυο LSTM, χρησιμοποιείται μόνο η έξοδος του τελευταίου χρονικού

βήματος o_n , στην οποία περιλαμβάνεται όλη η πληροφορία του ιστορικού (προηγούμενων λέξεων) που βρίσκεται αποθηκευμένη ως αριθμοί στα βάρη του κρυφού επιπέδου του LSTM. Η έξοδος o_n δίνεται ως είσοδος στο softmax επίπεδο, το οποίο παράγει την κατανομή πιθανοτήτων σύμφωνα με την οποία προσδιορίζεται η πιθανότητα με την οποία, το προς εξέταση κείμενο έχει θετικό ή αρνητικό συναίσθημα.



Εικόνα 34 Απεικόνιση Αρχιτεκτονικής I

Τα μεγέθη που χαρακτηρίζουν το παραπάνω μοντέλο είναι:

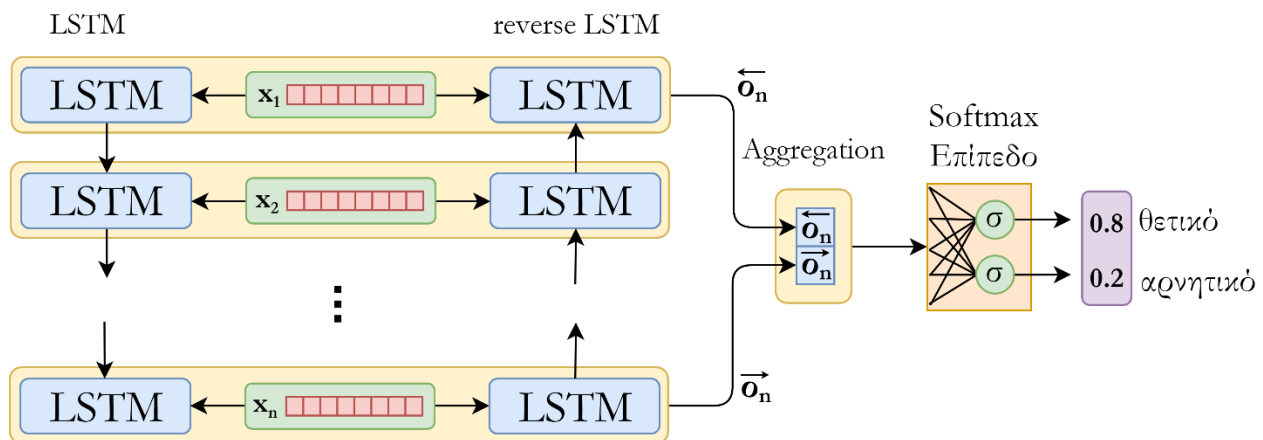
- Το μέγεθος X των διανυσμάτων αναπαράστασης των λέξεων σύμφωνα με την μέθοδο GloVe.
- Το μέγεθος H του κρυφού επιπέδου του κυττάρου LSTM
- Τα βάρη των συναπτικών συνδέσεων εισόδου του επιπέδου softmax, που αναπαριστώνται ως πίνακας μεγέθους $W = H \times 2$ (σειρές x στήλες).

6.3.2.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ II

Η δεύτερη αρχιτεκτονική είναι παρόμοια με την προσέγγιση που ακολουθεί η αρχική, με την διαφορά ότι στο επίπεδο LSTM υπάρχει τώρα ένα αμφίδρομο αναδρομικό δίκτυο μακράς βραχυπρόθεσμης μνήμης (Bidirectional LSTM ή BLSTM). Ένα δίκτυο BLSTM ουσιαστικά προκύπτει από ένα αμφίδρομο αναδρομικό δίκτυο (που παρουσιάστηκε στην ενότητα 3.4) το οποίο στο κρυφό του επίπεδο κάνει χρήση του κυττάρου LSTM. Σκοπός είναι να αποκομίσουμε τα οφέλη που παρέχει ένα αμφίδρομο αναδρομικό δίκτυο, συγκρατώντας επιπλέον νοηματική

πληροφορία μέσω της επεξεργασία ενός κειμένου με την ορθή και την αντίστροφη φορά, αλλά ταυτόχρονα να μπορούμε να διατηρούμε όλες τις μακρινές εξαρτήσεις μεταξύ των λέξεων.

Με παρόμοια λογική όπως και στην αρχική αρχιτεκτονική, οι έξοδοι που συγκροτούνται από το δίκτυο BLSTM είναι η πρώτη και η τελευταία. Η πρώτη χρονικά έξοδος του BLSTM ουσιαστικά περιέχει την τελευταία έξοδο από το ανάστροφο LSTM \vec{o}_n (τελικό αποτέλεσμα από την ανάστροφη επεξεργασία του κειμένου) και η τελευταία έξοδος του BLSTM περιέχει την τελευταία έξοδο του ορθού σε φορά LSTM \overleftarrow{o}_n .



Εικόνα 35 Απεικόνιση Αρχιτεκτονικής II

Οι έξοδοι $\vec{o}_n, \overleftarrow{o}_n$ συνενώνονται ως ένα διάνυσμα και εισάγονται στο επίπεδο softmax για την τελική πιθανοτική κατανομή συναισθήματος του εξεταζόμενου κειμένου.

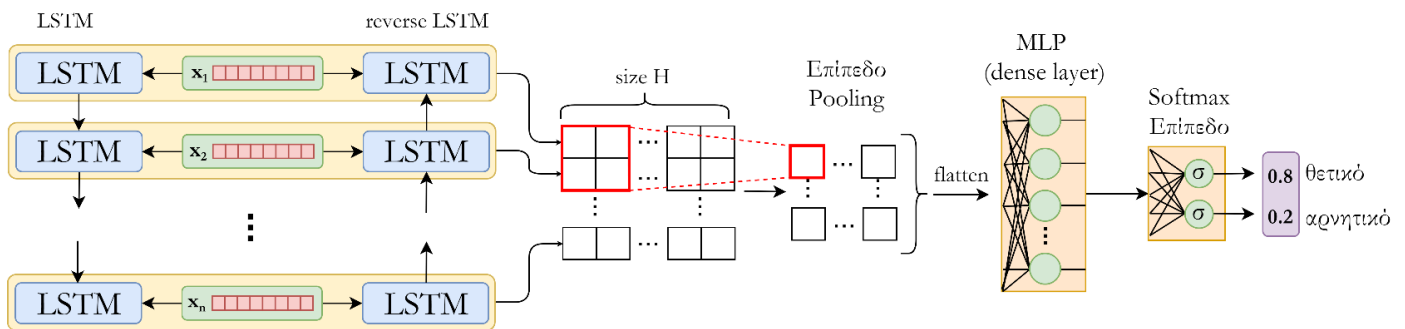
Τα μεγέθη που χαρακτηρίζουν το παραπάνω μοντέλο είναι:

- Το μέγεθος X των διανυσμάτων αναπαράστασης των λέξεων σύμφωνα με την μέθοδο GloVe.
- Το μέγεθος H των κρυφών επιπέδων των κυττάρων LSTM (ορθού και ανάστροφου)
- Τα βάρη των συναπτικών συνδέσεων εισόδου του επιπέδου softmax, που αναπαριστώνται ως πίνακας μεγέθους $W = 2H \times 2$ (σειρές x στήλες).

6.3.2.3 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΙΙΙ

Η τρίτη αρχιτεκτονική αποτελεί την τελευταία καθώς και την πιο σύνθετη προσέγγιση από τις προηγούμενες δύο. Το επίπεδο του LSTM, είναι πανομοιότυπο με αυτό της Αρχιτεκτονικής ΙΙ καθώς αποτελείται από έναν αμφίδρομο δίκτυο μακράς βραχυπρόθεσμης μνήμης. Ωστόσο, στην αρχιτεκτονική αυτή, δεν χρησιμοποιείται μόνο η πρώτη και η τελευταία έξοδος του BLSTM, αλλά όλες οι εξοδοί από κάθε χρονικό βήμα του δικτύου. Έτσι, αν το κρυφό επίπεδο των LSTM του αμφίδρομου δικτύου έχει μέγεθος H , η έξοδος του BLSTM αποτελούμενη από όλες τις εξόδους της ακολουθίας, θα είναι πίνακας δύο διαστάσεων μεγέθους, $L \times H$, όπου L το στατικό μήκος της ακολουθίας.

Δεδομένης της μεγάλης διάστασης που μπορεί να λάβει η έξοδος αυτή, το επόμενο στάδιο στο οποίο τροφοδοτείται είναι το επίπεδο pooling. Το στάδιο αυτό, όπως αναφέρθηκε και στην περιγραφή του, έχει ως λειτουργία να μειώσει τα διαστασιολογικά μεγέθη της δοσμένης εισόδου αλλά ταυτόχρονα να διατηρήσει την πληροφορία που περιέχεται. Στο pooling στάδιο, μπορεί να χρησιμοποιηθεί τόσο mean pooling όσο και max pooling. Αν και η απόδοση της κάθε τεχνικής εξαρτάται από τη φύση του προβλήματος, είναι κοινή παραδοχή πως το max pooling, διατηρώντας μόνο τα χαρακτηριστικά μέγιστης τιμής, εξαιρεί μεγάλο μέρος της πληροφορίας, σε αντίθεση με τον mean pooling που διατηρεί πληροφορία από όλα τα χαρακτηριστικά.



Εικόνα 36 Απεικόνιση Αρχιτεκτονικής ΙΙΙ

Το αποτέλεσμα του σταδίου pooling είναι ένα πίνακας με σημαντικά μειωμένος μέγεθος διαστάσεων. Ο πίνακας αυτός, αφού «σοπεδωθεί» (flatten) σε ένα διάνυσμα δίνεται στην είσοδο ενός πολυεπίπεδου perceptron (MLP) με μόνο ένα επίπεδο. Λόγω των πυκνών συνδέσεων του στα βάρη εισόδου, το επίπεδο αυτό αναφέρεται και ως πυκνό επίπεδο (dense layer). Ο λόγος χρήσης του, είναι συνδυαστεί με το επίπεδο εξόδου και να παρεμβληθεί μεταξύ αυτού και του επιπέδου pooling προκειμένου να αυξήσει την εκφραστικότητα και τις ικανότητες

αναπαράστασης του δικτύου. Ανάλογα με το ελάχιστο εκπαιδευόμενο μοντέλο Αρχιτεκτονικής III, το πλήθος των νευρώνων αυτού του επιπέδου επιλέχτηκε να είναι ίσο με το μέγεθος του κρυφού επιπέδου του δικτύου μακριάς βραχυπρόθεσμης μνήμης.

Τα μεγέθη που χαρακτηρίζουν το παραπάνω μοντέλο είναι:

- Το μέγεθος X των διανυσμάτων αναπαράστασης των λέξεων σύμφωνα με την μέθοδο GloVE.
- Το μέγεθος H των κρυφών επιπέδων των κυττάρων LSTM (ορθού και ανάστροφου)
- Το μέγεθος των διαστάσεων του φίλτρου pooling P_w, P_h καθώς και το μέγεθος S των strides
- Το μέγεθος του πίνακα που προκύπτει από την διαδικασία του pooling με διαστάσεις H_{new}, W_{new} . Οι τιμές αυτών υπολογίζονται βάση των σχέσεων που παρουσιάστηκαν στην περιγραφή του επιπέδου pooling. Το διάνυσμα που δημιουργείται από την ισοπέδωση (flatten) του πίνακα έχει μέγεθος $V = H_{new} \times W_{new}$.
- Τα βάρη των συναπτικών συνδέσεων εισόδου του πυκνού επιπέδου, που αναπαριστώνται ως πίνακας μεγέθους $W_d = N \times V$ (σειρές x στήλες), όπου N το πλήθος των νευρώνων του επιπέδου αυτού
- Τα βάρη των συναπτικών συνδέσεων εισόδου του επιπέδου softmax, που αναπαριστώνται ως πίνακας μεγέθους $W_o = N \times 2$ (σειρές x στήλες).

6.4 Εκπαίδευση

6.4.1 Επιλεγμένες Μέθοδοι Γενίκευσης

6.4.1.1 Early Stopping

Κατά την εκπαίδευση νευρωνικών δικτύων, μία από τις υπερπαραμέτρους οι οποίες πρέπει να ληφθούν υπόψιν είναι το πλήθος των εποχών εκπαίδευσης του δικτύου, δηλαδή το πόσοι πλήρεις κύκλοι θα γίνουν στο σύνολο δεδομένων εκπαίδευσης προκειμένου να φτάσουμε στο επιθυμητό επίπεδο εκπαίδευσης. Ωστόσο, υπάρχει ο κίνδυνος, αν ο αριθμός των εποχών είναι μεγάλος, να παρατηρηθούν φαινόμενα υπερπροσαρμογής (overfitting) στο σύνολο εκπαίδευσης, με αποτέλεσμα το εκπαιδευόμενο μοντέλο να χάνει την γενικότητα του (generalization) και η απόδοση του σε άλλα σύνολα δεδομένων να είναι μειωμένη.

Για να αποφευχθεί το πρόβλημα αυτό, στην εκπαίδευση των μοντέλων μας, χρησιμοποιούμε την τεχνική early stopping. Βάση αυτής, χρησιμοποιούμε ένα σύνολο δεδομένων γνωστό και ως

validation set, πάνω στο οποίο εξετάζουμε την απόδοση του μοντέλου μας κατά την διάρκεια εκπαίδευσης. Ανεξαρτήτως της απόδοσης στο σύνολο δεδομένων εκπαίδευσης, αν η απόδοση του δικτύου στο validation set αρχίσει να μειώνεται, αυτό αποτελεί δείγμα μείωσης της γενικότητας του δικτύου και η εκπαίδευση σταματάει.

6.4.1.2 Dropout νευρώνων

Μια άλλη τεχνική που θα χρησιμοποιηθεί για να διασφαλίσουμε την γενικότητα των μοντέλων και να αποφύγουμε φαινόμενα προσαρμογής είναι η τεχνική dropout. Η λειτουργία της βασίζεται στο γεγονός, ότι κατά την εκπαίδευση σύνθετων δικτύων, οι νευρώνες μπορεί να αναπτύξουν αλληλεξαρτήσεις μεταξύ τους κάτι το οποίο περιορίζει την ατομική τους απόδοση οδηγώντας σε υπερβολική προσαρμογή στα δεδομένα εκπαίδευσης. Η τεχνική dropout αποτελεί μια μέθοδο κανονικοποίησης (regularization) σύμφωνα με την οποία, σε κάθε βήμα της εκπαίδευσης τυχαία νευρώνες απενεργοποιούνται από το δίκτυο. Συγκεκριμένα ορίζεται μια πιθανότητα p , βάση της οποίας ένας νευρώνας θα απενεργοποιηθεί με πιθανότητα $(1-p)$ ενός θα παραμείνει ενεργός με πιθανότητα p . Στα μοντέλα μας, dropout εφαρμόζεται στους νευρώνες των κρυφών επιπέδων των δικτύων LSTM καθώς και στο πολυεπίπεδο perceptron της *Αρχιτεκτονικής III*.

6.4.2 Βελτιστοποίηση

Η εκπαίδευση των μοντέλων μας θα βασιστεί σε μια βελτιστοποιημένη εκδοχή της κατάβασης δυναμικού. Η βελτιστοποίηση αυτή, συγκεκριμένα είναι γνωστή ως Adam και αποτελεί μια στοχαστική βελτιστοποίηση που κάνει χρήση μόνο παραγώγων πρώτης τάξης με μικρές απαιτήσεις σε μνήμη. Ουσιαστικά αποτελεί ένας συνδυασμός των πλεονεκτημάτων που προσφέρει η μέθοδος Adagrad και η μέθοδος RMSprop. Πέρα από την αποθήκευση ενός εκθετικά ελαττούμενου μέσου όρου των τετραγώνων των προηγούμενων κλίσεων u_t , η μέθοδος Adam επιπρόσθετα αποθηκεύει και έναν ελαττούμενο μέσο όρο των προηγούμενων κλίσεων m_t , όπως γίνεται και στην μέθοδο Ορμής. Ο υπολογισμός των u_t και m_t γίνεται ως εξής:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) \nabla_{\theta} f_t(\theta_{t-1}) \\ u_t &= \beta_2 u_{t-1} + (1 - \beta_2) \nabla_{\theta}^2 f_t(\theta_{t-1}) \end{aligned}$$

Τα m_t , u_t καθώς αρχικοποιούνται ως μηδενικά διανύσματα παρατηρήθηκε ότι παραμένουν πολωμένα προς το 0, ειδικά στα πρώτα βήματα όπου οι ρυθμοί ελάττωσης είναι μικροί. Για την αντιμετώπιση αυτής της πόλωσης υπολογίστηκαν οι διορθωμένες εκτιμήσεις τους:

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t}$$
$$\widehat{u}_t = \frac{u_t}{1 - \beta_2^t}$$

Έτσι η ανανέωση των παραμέτρων, όπως υπολογίζεται και στους Adadelta και RMSprop είναι:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\widehat{u}_t} + \varepsilon} \widehat{m}_t$$

ΚΕΦΑΛΑΙΟ 7

ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ

7.1 Το σύνολο δεδομένων

Το σύνολο δεδομένων που χρησιμοποιήθηκε για την εκπαίδευση αλλά και την εξέταση της απόδοσης των μοντέλων ανάλυσης συναισθήματος, είναι μια συλλογή από κριτικές ταινιών της βάσης IMDb και το οποίο έχει δημιουργηθεί και συλλεχθεί από τρίτη πηγή (<http://ai.stanford.edu/~amaas/data/sentiment/>). Τα δεδομένα αυτά έχουν επιλεγεί προσεκτικά ώστε να περιέχουν κριτικές με αρικτά πολωμένο συναίσθημα. Συνολικά, το σύνολο δεδομένων αυτών περιέχει 50.000 ταινίες από τις οποίες οι 25.000 παρέχονται ως σύνολο εκπαίδευσης και οι υπόλοιπες 25.000 ως σύνολο ελέγχου, ενώ για κάθε ταινία δεν υπάρχουν πάνω από 30 κριτικές.

Για τις ανάγκες της διπλωματικής αυτής, συγκεκριμένα χρησιμοποιήθηκαν 25.000 κριτικές για την εκπαίδευση των μοντέλων, ενώ από σύνολο των 25.000 κριτικών ελέγχου κρατήθηκαν οι 20.000 για τον έλεγχο, ενώ οι υπόλοιπες 5.000 χρησιμοποιήθηκαν ως σύνολο validation για την εξυπηρέτηση του early stopping.

7.2 Μοντελοποίηση και Πειράματα

7.2.1 Διανύσματα λέξεων

Όπως έχει αναφερθεί και στο κεφάλαιο της προσέγγισης προβλήματος, για την διανυσματική αναπαράσταση των λέξεων έχει χρησιμοποιηθεί η μέθοδος GloVe.

Για την χρήση των διανυσμάτων που θα αναπαραστήσουν τις λέξεις των κριτικών του συνόλου δεδομένων, γίνεται χρήση των διανυσματικών αναπαραστάσεων που έχουν ήδη εκπαιδευτεί από τους δημιουργούς της μεθόδου. Αυτές αποτελούν αρχεία .txt οι οποίες έχουν την εξής δομή: σε κάθε γραμμή του αρχείου, αρχικά αναγράφεται η λέξη, στην συνέχεια ακολουθεί ένα κενό (space) και τέλος ακολουθεί το διάνυσμα αναπαράστασης λέξης ως σειρά από αριθμούς. Το αρχείο των εκπαιδευμένων λέξεων-διανυσμάτων που χρησιμοποιήθηκε, είναι μια συλλογή από 400.000 λέξεις καθεμία από τις οποίες αναπαριστάνεται με ένα διάνυσμα 50 διαστάσεων.

Η διαδικασία επεξεργασίας του αρχείου και φόρτωσης των διανυσμάτων στην μνήμη σε κάθε νέα εκπαίδευση ενός μοντέλου, θα ήταν τρομερά χρονοβόρα. Για τον λόγο αυτό, με χρήση της βιβλιοθήκης NumPy της γλώσσας Python, μετά το διάβασμα του αρχείου, τα διανύσματα φορτώθηκαν σε έναν πίνακα (numpy array), του οποίου το στιγμιότυπο αποθηκεύτηκε σε αρχείο τύπου .npy. Για κάθε επόμενη φορά που επιθυμούσαμε να φορτώσουμε τις λέξεις-διανύσματα για τις ανάγκες της διπλωματικής, γινόταν χρήση αυτού του αρχείου (για τον κώδικα επεξεργασίας του αρχείου με τις εκπαιδευμένες λέξεις διανύσματα μπορείτε να ανατρέξετε στο Παράρτημα).

Πιο συγκεκριμένα, για την διανυσματική αναπαράσταση των κειμένων χρησιμοποιώντας τις λέξεις-διανύσματα ακολουθήθηκε η εξής προ-επεξεργασία:

1. Αρχικά μέσω της επεξεργασίας του αρχείου με τις προ-εκπαιδευμένες λέξεις-διανύσματα, όπως αναφέρθηκε και παραπάνω, δημιουργήθηκαν και αποθηκεύτηκαν: μία λίστα (numpy.array) με τις λέξεις του αρχείου και μία λίστα με τα αντίστοιχα διανύσματα διατηρώντας την αντιστοιχία θέσεων μεταξύ των δύο λιστών.
2. Στην συνέχεια ορίστηκε ένα σταθερό μέγεθος ακολουθίας (sequence size). Το μέγεθος αυτό ουσιαστικά προσδιορίζει το πλήθος των λέξεων που θα διαβαστεί από κάθε κείμενο κριτικής. Έτσι βάση αυτού του μεγέθους ακολουθίας, διαβάζονται οι λέξεις ενός κειμένου και κάθε λέξη αναπαριστάνεται ως αριθμός. Ο αριθμός αυτός αντιστοιχεί στον αριθμό της θέσης της αντίστοιχης λέξης στην αποθηκευμένη λίστα λέξεων που δημιουργήθηκε στο πιο πάνω βήμα. Αν το πλήθος λέξεων του κειμένου είναι μικρότερο από το ορισμένο μέγεθος ακολουθίας, τότε οι υπόλοιπες θέσεις συμπληρώνονται με 0, το οποίο αντιστοιχίζεται με τον χαρακτήρα <PAD> στη λίστα λέξεων. Εφαρμόζοντας την παραπάνω διαδικασία για συγκεκριμένο μέγεθος ακολουθίας για όλες τις κριτικές, δημιουργείται ένας πίνακας διαστάσεων (πλήθος κριτικών x μέγεθος ακολουθίας) όπου κάθε γραμμή είναι μια διαφορετική κριτική. Αυτός ο πίνακας αποθηκεύεται με τον ίδιο τρόπο όπως οι λίστες στο βήμα 1, προκειμένου να το χρησιμοποιούμε σε κάθε νέα εκπαίδευση μοντέλου. Η αντιστοιχία μεταξύ των στοιχείων (λέξεις) του πίνακα κειμένων και των διανυσμάτων που περιέχονται στην αποθηκευμένη λίστα διανυσμάτων την αναλαμβάνει συγκεκριμένος κόμβος στον κώδικα του Tensorflow.

7.2.3 Πειράματα

Για την διεξαγωγή των πειραμάτων, δημιουργήθηκαν μοντέλα τα οποία είναι βασισμένα στις τρεις αρχιτεκτονικές που παρουσιάστηκαν στο προηγούμενο κεφάλαιο. Σε κάθε αρχιτεκτονική αντιστοιχούν παραπάνω από ένα μοντέλα στα οποία παραλλάσσονται οι τιμές βασικών παραμέτρων. Οι βασικές παράμετροι οι οποίες μεταβλήθηκαν προκειμένου να δημιουργηθούν και να εκπαιδευτούν τα μοντέλα αυτά και να εξεταστεί η απόδοση τους ως ταξινομητές συναισθήματος φαίνονται στον παρακάτω πίνακα:

Υπερπαράμετροι	Τιμές
Μέγεθος σταθερής ακολουθίας	250, 500, 1000
Μέγεθος κρυφού επιπέδου LSTM	64, 128, 256

Πίνακας 1 Πίνακας μεταβαλλόμενων υπερπαραμέτρων

Έτσι για καθεμία από τις τιμές 250, 500, 1000 του μεγέθους ακολουθίας, δημιουργήθηκαν μοντέλα και από τις τρεις αρχιτεκτονικές καθένα από τα οποία διατηρεί διαφορετικό μέγεθος κρυφού επιπέδου για το δίκτυο μακράς βραχυπρόθεσμης μνήμης. Επισημαίνεται ότι, όπως αναφέρθηκε και στην περιγραφή της *Αρχιτεκτονικής III*, τα μοντέλα που την υλοποιούν, μεταβάλλοντας το μέγεθος του κρυφού επιπέδου του LSTM μεταβάλλουν μαζί και το μέγεθος του πυκνού επιπέδου (dense layer) που περιλαμβάνουν.

Ωστόσο, πέρα από τις προαναφερθείσες υπερπαραμέτρους, υπάρχουν και άλλες οι οποίες χαρακτηρίζουν τα εξεταζόμενα μοντέλα οι οποίες θα μπορούσαν να μεταβληθούν οδηγώντας σε διαφορετικά αποτελέσματα. Παρόλα αυτά, στα πλαίσια την διπλωματικής αυτής, οι παράμετροι αυτές θα παραμείνουν αμετάβλητες. Στον παρακάτω πίνακα αναγράφονται οι σχετικές παράμετροι.

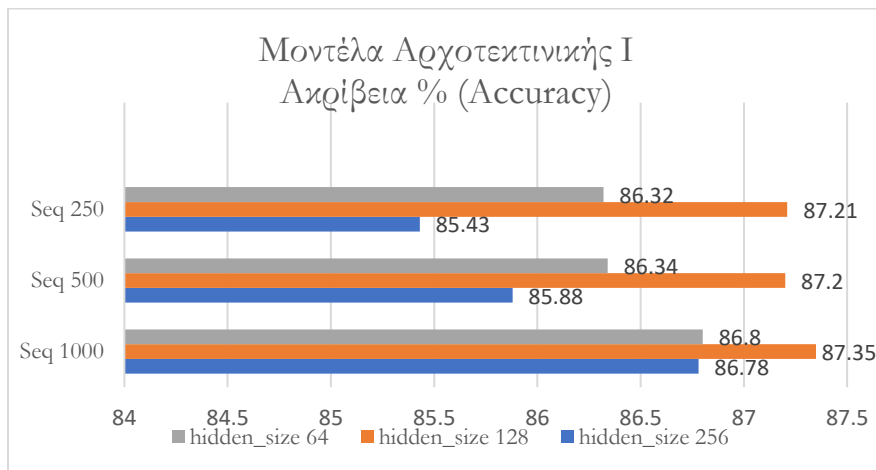
Υπερπαράμετροι	Τιμές
Διαστάσεις Word Vector	50
Batch Size	25
Dropout	0.75
Προκαθορισμένο Learning Rate	0.01
Μέγεθος Pooling	2 x 2
Pooling Strides	2
Pooling	Mean Pooling

Πίνακας 2 Πίνακας σταθερών υπερπαραμέτρων

7.3 Αποτελέσματα μοντέλων

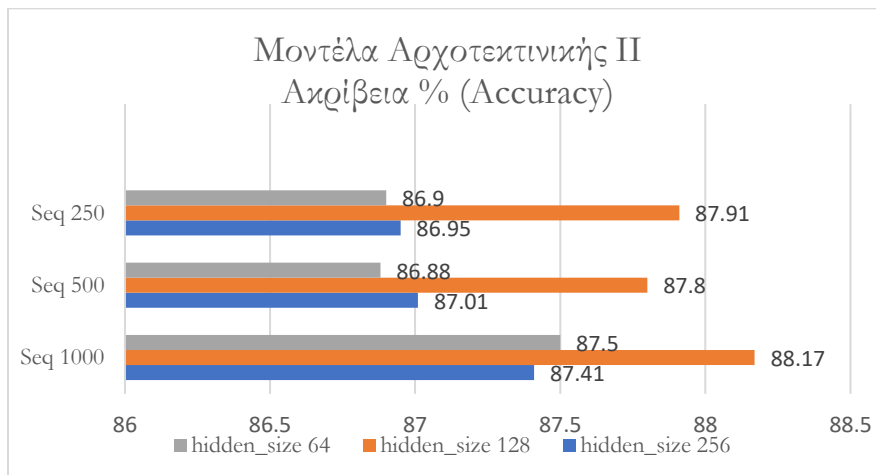
Αφού τα μοντέλα μας δημιουργηθούν και εκπαιδευτούν, θα πρέπει ελεγχτούν ως προς την ικανότητά τους να ταξινομήσουν το συναίσθημα ενός κειμένου, εξετάζοντας την απόδοσή τους πάνω στις κριτικές του συνόλου ελέγχου. Στην ενότητα αυτή, παρουσιάζονται τα αποτελέσματα της απόδοσης των 27 μοντέλων μας. Την απόδοσή τους την μετράμε ως ακρίβεια (accuracy) ταξινόμησης δηλαδή ως τον λόγο των κριτικών που ταξινομήθηκαν σωστά ως προς τις συνολικό πλήθος των κριτικών ελέγχου:

- **Ακρίβεια ταξινόμησης συναισθήματος μοντέλων βασισμένα στην Αρχιτεκτονική I**



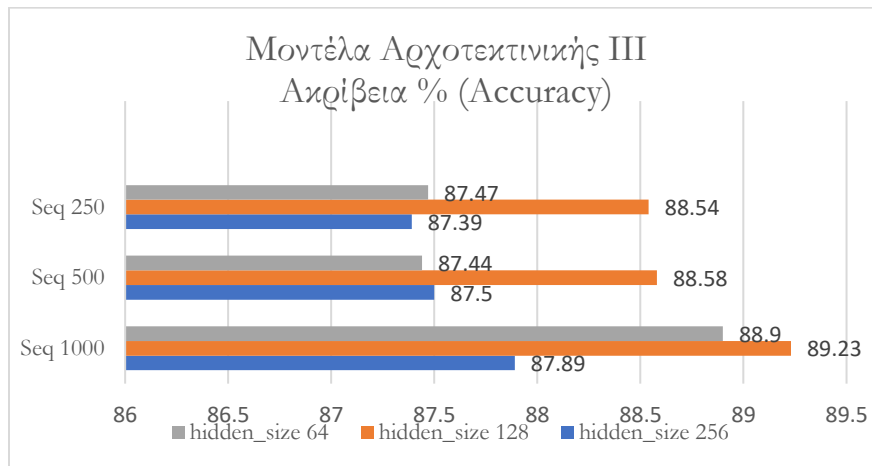
Εικόνα 37 Αποτελέσματα στην ταξινόμηση συναισθήματος μοντέλων Αρχιτεκτονικής I

- **Ακρίβεια ταξινόμησης συναισθήματος μοντέλων βασισμένα στην Αρχιτεκτονική II**



Εικόνα 38 Αποτελέσματα στην ταξινόμηση συναισθήματος μοντέλων Αρχιτεκτονικής II

- Ακρίβεια ταξινόμησης συναισθήματος μοντέλων βασισμένα στην Αρχιτεκτονική III



Εικόνα 39 Αποτελέσματα στην ταξινόμηση συναισθήματος μοντέλων Αρχιτεκτονικής III

7.4 Αξιολόγηση και συμπεράσματα

Από τα αποτελέσματα που παρουσιάστηκαν στην παραπάνω ενότητα, βλέπουμε ότι έχουμε πάρει αρκετά επιθυμητές επιδόσεις στην ικανότητα των μοντέλων μας να ταξινομήνουν το συναίσθημα κριτικών ταινιών. Από τις τρεις εξεταζόμενες αρχιτεκτονικές, φαίνεται ότι τα μοντέλα που βασίζονται στην τρίτη, παρουσιάζουν καλύτερη ακρίβεια στην ταξινόμηση συναισθήματος στις κριτικές των δεδομένων ελέγχου.

Όσον αφορά την το μέγεθος του κρυφού επιπέδου, σχεδόν σε όλες τις περιπτώσεις παρατηρείται πως, ανεξαρτήτως της αρχιτεκτονικής και των υπόλοιπων παραμέτρων, τα μοντέλα των οποίων το κρυφό επίπεδο του δικτύου τους μακράς βραχυπρόθεσμης μνήμης έχει μέγεθος 128, ξεπερνάνε σε επίδοση όλα τα υπόλοιπα. Εδώ αξίζει να παρατηρηθεί πως τα μοντέλα με αντίστοιχο μέγεθος κρυφού επιπέδου 256 αν και θα περιμέναμε να αποδίδει καλύτερα από τα υπόλοιπα λόγω αυξημένης αναπαραστατικής δύναμης, παρατηρούμε πως τελικά είναι αυτό με την χαμηλότερη ακρίβεια η οποία σε κάποιες περιπτώσεις φτάνει κοντά στις 2 μονάδες διαφορά από το αντίστοιχο καλύτερο. Αυτό θα μπορούσε να αποδοθεί στην αργότερη σύγκλιση που παρουσιάζει ένα τέτοιο δίκτυο, με αποτέλεσμα να μην προλάβει να φτάσει στο ελάχιστο του σφάλματος του στον ίδιο χρόνο εκπαίδευσης με ένα αντίστοιχο μικρότερου μεγέθους.

Σχετικά με την παράμετρο του μήκους ακολουθίας και εδώ τα αποτελέσματα είναι ξεκάθαρα, με την ακολουθία μήκους 1000 να είναι αυτή που αποδίδει καλύτερα σε κάθε

περίπτωση από μοντέλα μικρότερης ακολουθίας καθώς περιλαμβάνει περισσότερες λέξεις και άρα περισσότερη πληροφορία. Ο λόγος για τον οποίο οι ακολουθίες με μήκη 250 και 500 παρουσιάζουν παρόμοια απόδοση μπορεί να οφείλεται στο γεγονός ότι το πλήθος κειμένων με λέξεις περισσότερες από 250 και λιγότερες από 500 είναι μικρά σε αριθμό, με αποτέλεσμα αυτές οι δύο ακολουθίες να καλύπτουν προσεγγιστικά το ίδιο εύρος κειμένων.

Συνολικά τα αποτελέσματα που πήραμε από τις παραπάνω αρχιτεκτονικές είναι αρκετά ικανοποιητικά με χειρότερο ποσοστό ταξινόμησης κοντά στο 85% και καλύτερο κοντά στο 90%. Με επιπλέον παραμετροποίηση, δοκιμή διανυσμάτων μεγαλύτερων διαστάσεων όπως 100 ή 200 καθώς και με πειραματισμό στο μέγεθος ακολουθίας και το μέγεθος του κρυφού επιπέδου του δικτύου μακράς βραχυπρόθεσμης μνήμης, υπάρχει πιθανότητα να επιτευχθεί και ποσοτό μεγαλύτερο του 90%. Επιπλέον ενδιαφέρουσα επέκταση θα ήταν η δοκιμή των εν λόγω μοντέλων σε διαφορετικού είδους σύνολο δεδομένων, δηλαδή σε άλλα κείμενα με συναισθηματική φόρτιση (π.χ. tweets) για να εξεταστούν και εκεί οι δυνατότητές τους.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Κωνσταντίνος Διαμανταράς. Τεχνητά Νευρωνικά Δίκτυα, Κλειδάριθμος 2007

Βλαχάβας Ιωάννης. Τεχνητή Νοημοσύνη, Εκδόσεις Πανεπιστημίου Μακεδονίας, 2019

Yu, Zhou, et al. "Using bidirectional LSTM recurrent neural networks to learn high-level abstractions of sequential features for automated scoring of non-native spontaneous speech." Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on. IEEE, 2015.

Alessia, D., Ferri, F., Grifoni, P., & Guzzo, T. (2015). Approaches, tools and applications for sentiment analysis implementation. *International Journal of Computer Applications*, 125(3).

Sharma, A., & Dey, S. (2012, October). A comparative study of feature selection and machine learning techniques for sentiment analysis. In *Proceedings of the 2012 ACM research in applied computation symposium* (pp. 1-7). ACM.

Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H., & Xu, B. (2016). Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011, June). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1* (pp. 142-150). Association for Computational Linguistics.

A. Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. Textbook, Studies in Computational Intelligence, Springer, 2012

Tom Hope, Yehezkel S. Resheff, Itay Lieder. *Learning TensorFlow A Guide to Building Deep Learning Systems*, OREILLY, 2017

Giancarlo Zaccane. *Getting Started with TensorFlow*, Packt, 2016

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929-1958.

Wang, J., Yu, L. C., Lai, K. R., & Zhang, X. (2016). Dimensional sentiment analysis using a regional CNN-LSTM model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (Vol. 2, pp. 225-230).

Rosenthal, S., Farra, N., & Nakov, P. (2017). SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*(pp. 502-518).

Baziotis, C., Pelekis, N., & Doulkeridis, C. (2017). Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*(pp. 747-754).

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016, November). Tensorflow: a system for large-scale machine learning. In *OSDI* (Vol. 16, pp. 265-283).

ΠΑΡΑΡΤΗΜΑ

Στο παράρτημα αυτό βρίσκεται ο κώδικας με τον οποίο υλοποιήθηκαν τα μοντέλα βαθιάς μάθησης που πραγματεύεται η παρούσα διπλωματική. Συγκεκριμένα, παρουσιάζονται: η συνάρτηση για το διάβασμα του αρχείου των διανυσμάτων λέξεων και την φόρτωση τους, η συνάρτηση για το διάβασμα όλων των κριτικών και την αποθήκευσή τους ως πίνακες τύπου `numpy.array` και η συνάρτηση που δημιουργεί το μοντέλο για τις αρχιτεκτονικές I, II. Για τον πλήρη κώδικα, επισκεφτείτε την σελίδα https://gitlab.com/thodorisf/diploma_thesis_code

Συνάρτηση φόρτωσης διανυσμάτων-λέξεων

```
import numpy as np

def loadWordVectors(embeddingFile):
    vecs = []
    words = []
    print('Loading words list and word vectors from embedding file ' +
embeddingFile)
    with open(embeddingFile, "r", encoding='utf-8') as file:
        for line in file:
            data = line.strip().split(" ")
            words.append(data[0])
            vecs.append([float(i) for i in data[1:]])
    word_vectors = np.array(vecs)

    np.save("./wordsList", word_vectors)
    np.save("./word-vectors", word_vectors)

    return words, word_vectors
```

Συνάρτηση δημιουργίας πίνακα κριτικών με σταθερό μήκος ακολουθίας

```
import re
import numpy as np
import os

#paths for training and testing data set
positiveDirectory_train = './aclImdb/train/pos/'
negativeDirectory_train = './aclImdb/train/neg/'
positiveDirectory_test = './aclImdb/test/pos/'
negativeDirectory_test = './aclImdb/test/neg/'

#clear the given text from punctutation and symbols leaving only alphanumeric
characters
strip_special_chars = re.compile("[^A-Za-z0-9 ]+")

def cleanSentences(string):
    string = string.lower().replace("<br />", " ")
    return re.sub(strip_special_chars, "", string.lower())

#load the words of each review (positive and negative) in an numpy array
def loadTrainingReviews(type, number_of_reviews, wordList, maxSeqLength):
    reviews = np.zeros((number_of_reviews, maxSeqLength), dtype='int32')
    reviewIndex = 0

    if (type == 'train'):
        positiveDirectory = positiveDirectory_train
        negativeDirectory = negativeDirectory_train
    elif (type == 'test'):
        positiveDirectory = positiveDirectory_test
        negativeDirectory = negativeDirectory_test
    else:
        print("Wrong type of data set! Type must be 'train' or 'test'")
        return

    print(' -Loading positive reviews...')
    posDirectory = os.fsencode(positiveDirectory)
    for file in os.listdir(posDirectory):
        filename = os.fsdecode(file)
        if filename.endswith(".txt"):
            # print (positiveDirectory+filename)
            with open(positiveDirectory+filename, 'r', encoding='utf-8') as review:

                wordIndex = 0
                review_words = cleanSentences(review.readline()).split()
                for word in review_words:
```

```
#print (word)
try:
    #print(wordList.index(word))
    reviews[reviewIndex][wordIndex] = wordList.index(word)
except ValueError:
    reviews[reviewIndex][wordIndex] = 399999 #Vector for unkown
# print(wordIndex)
wordIndex +=1
if wordIndex >= maxSeqLength:
    break
reviewIndex += 1
else:
    continue

print(' -Loading negative reviews...')
negDirectory = os.fsencode(negativeDirectory)
for file in os.listdir(negDirectory):
    filename = os.fsdecode(file)
    if filename.endswith(".txt"):
        #print (positiveDirectory_train+filename)
        with open(negativeDirectory+filename, 'r', encoding='utf-8') as review:

            wordIndex = 0
            review_words = cleanSentences(review.readline()).split()
            for word in review_words:
                # print (word)
                try:
                    reviews[reviewIndex][wordIndex] = wordList.index(word)
                except ValueError:
                    reviews[reviewIndex][wordIndex] = 399999 #Vector for unkown
            # print(wordIndex)
            wordIndex +=1
            if wordIndex >= maxSeqLength:
                break
            reviewIndex += 1
        else:
            continue

np.save("./reviews-"+str(maxSeqLength)+"-"+type, reviews)

return reviews
```

```

def CreateModel(hidden_layer_size, num_of_sentiment_classes, vector_dimensions,
maximum_sequence_length, word_vectors, batch_size, architecture):

    # <<Placeholders>>
    input_data = tf.placeholder(tf.int32, [None, maximum_sequence_length], name="input_data")
    labels = tf.placeholder(tf.float32, [None, num_of_sentiment_classes], name="labels")
    keep_prob = tf.placeholder_with_default(0.75, shape=(), name="keep_prob")

    # The actual length of each input sequence in the current batch
    length = tf.cast(tf.reduce_sum(tf.sign(input_data), 1), tf.int32)

    # <<Embedding Layer>>
    rnn_input = tf.nn.embedding_lookup(word_vectors, input_data)

    if (architecture == '1'):

        # <<LSTM RNN>>
        lstmCells = [tf.contrib.rnn.BasicLSTMCell(size) for size in hidden_layer_size]
        lstmCells_with_drops = [tf.contrib.rnn.DropoutWrapper(cell=lstm, output_keep_prob=keep_prob)
for lstm in
                        lstmCells]
        stackedLstm = tf.contrib.rnn.MultiRNNCell(lstmCells_with_drops)
        initial_state = stackedLstm.zero_state(batch_size=batch_size, dtype=tf.float32)
        rnn_outputs, _ = tf.nn.dynamic_rnn(cell=stackedLstm, inputs=tf.cast(rnn_input, tf.float32),
sequence_length=length,
initial_state=initial_state, dtype=tf.float32)
        last_rnn_output = tf.gather_nd(rnn_outputs, tf.stack([tf.range(batch_size), length - 1],
axis=1))

        # <<Softmax Layer>
        weights = tf.Variable(tf.truncated_normal([hidden_layer_size[-1], num_of_sentiment_classes]))
        bias = tf.Variable(tf.constant(0.1, shape=[hidden_layer_size[-1]]))
        logits = (tf.matmul(last_rnn_output, weights) + bias)

        # apply softmax to prediction and use cross entropy as cost function
        softmax_error = tf.nn.softmax_cross_entropy_with_logits(logits=logits, labels=labels)
        loss = tf.reduce_mean(softmax_error, name="loss")

        # add optimizer to train the network based on the loss function
        optimizer = tf.train.AdamOptimizer().minimize(loss)

        return optimizer

    elif (architecture == '2'):

        # <<LSTM RNN>>
        lstmCell_fw = tf.contrib.rnn.BasicLSTMCell(hidden_layer_size)
        lstmCell_bw = tf.contrib.rnn.BasicLSTMCell(hidden_layer_size)

        lstmCell_fw_with_drops = tf.contrib.rnn.DropoutWrapper(cell=lstmCell_fw,
output_keep_prob=keep_prob)
        lstmCell_bw_with_drops = tf.contrib.rnn.DropoutWrapper(cell=lstmCell_bw,
output_keep_prob=keep_prob)

        initial_state_fw = lstmCell_fw_with_drops.zero_state(batch_size=batch_size, dtype=tf.float32)
        initial_state_bw = lstmCell_bw_with_drops.zero_state(batch_size=batch_size, dtype=tf.float32)

        brnn_outputs, _ = tf.nn.bidirectional_dynamic_rnn(cell_fw=lstmCell_fw_with_drops,
cell_bw=lstmCell_bw_with_drops,
                                                    inputs=tf.cast(rnn_input, tf.float32),

```

```
initial_state_fw=initial_state_fw,                sequence_length=length,
dtype=tf.float32)                                initial_state_bw=initial_state_bw,

    first_brnn_output = tf.gather_nd(brnn_outputs, tf.stack([tf.range(batch_size), 0], axis=1))
    last_brnn_output = tf.gather_nd(brnn_outputs, tf.stack([tf.range(batch_size), length - 1],
axis=1))

    # <<>Softmax Layer>
    weights= tf.Variable(tf.truncated_normal(2*[hidden_layer_size[-1], num_of_sentiment_classes]))
    bias = tf.Variable(tf.constant(0.1, shape=[2*hidden_layer_size[-1]]))
    logits = (tf.matmul(np.concatenate((first_brnn_output, last_brnn_output), axis=None), weights)
+ bias)

    # apply softmax to prediction and use cross entropy as cost function
    softmax_error = tf.nn.softmax_cross_entropy_with_logits(logits=logits, labels=labels)
    loss = tf.reduce_mean(softmax_error, name="loss")

    # add optimizer to train the network based on the loss function
    optimizer = tf.train.AdamOptimizer().minimize(loss)

    return optimizer

else: #for the complete code visit https://gitlab.com/thodorisf/diploma\_thesis\_code
```