ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Χρήση Τεχνικών Βαθιάς Μηχανικής Μάθησης για την Αυτόματη Δημιουργία Περιγραφών Εικόνων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΓΕΩΡΓΙΟΥ ΒΑΣΙΛΑΚΗ

Επιβλέπων : Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2017

Η σελίδα αυτή είναι σκόπιμα λευκή.

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΈΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Χρήση Τεχνικών Βαθιάς Μηχανικής Μάθησης για την Αυτόματη Δημιουργία Περιγραφών Εικόνων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΓΕΩΡΓΙΟΥ ΒΑΣΙΛΑΚΗ

Επιβλέπων : Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 11η Σεπτεμβρίου 2017.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Ανδρέας-Γεώργιος Σταφυλοπάτης Καθηγητής Ε.Μ.Π.	Γεώργιος Στάμου Επ. Καθηγητής Ε.Μ.Π.	Παναγιώτης Τσανάκας Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2017

(Υπογραφή)

.....

ΓΕΩΡΓΙΟΣ ΒΑΣΙΛΑΚΗΣ

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

Περίληψη

Η αυτόματη περιγραφή του περιεχομένου μιας εικόνας αποτελεί ένα σημαντικό πρόβλημα στο πεδίο της τεχνητής νοημοσύνης, το οποίο συνδυάζει το επιστημονικό πεδίο της Όρασης Υπολογιστών με αυτό της Επεξεργασίας Φυσικής Γλώσσας.

Στην διπλωματική αυτή, υλοποιούμε και παρουσιάζουμε ένα μοντέλο, βασισμένο σε τεχνικές βαθιάς μηχανικής μάθησης, το οποίο συνδυάζει πρόσφατες προόδους στην Όραση Υπολογιστών και στην Μετάφραση Μηχανών και το οποίο είναι ικανό να δημιουργεί φυσικές προτάσεις οι οποίες περιγράφουν μια εικόνα. Πιο συγκεκριμένα, χρησιμοποιούμε έναν συνδυασμό Βαθιών Συνελικτικών Νευρωνικών Δικτύων (***CNNs***) και Ανατροφοδοτούμενων Νευρωνικών Δικτύων (***RNNs***), προκειμένου να πάρουμε το επιθυμητό αποτέλεσμα. Το μοντέλο μας εκπαιδεύεται έτσι ώστε να μεγιστοποιεί την πιθανότητα επιτυχίας της σωστής πρότασης περιγραφής, δεδομένης μιας εικόνας εισόδου.

Πειράματα σε μια μεγάλη βάση δεδομένων για εκπαίδευση, αξιολόγηση και έλεγχο λειτουργίας, όπως είναι η ***MSCOCO 2015*** την οποία και χρησιμοποιήσαμε, αποδεικνύουν την ακρίβεια του μοντέλου καθώς και την ευφράδεια της γλώσσας που μαθαίνει αποκλειστικά από περιγραφές εικόνων. Το μοντέλο μας είναι, συχνά, αρκετά ακριβές, γεγονός που επαληθεύουμε ποιοτικά και ποσοτικά.

Λέξεις Κλειδιά: Αυτόματη περιγραφή εικόνων, επεξεργασία εικόνων, συνελικτικά νευρωνικά δίκτυα, ανατροφοδοτούμενα νευρωνικά δίκτυα, γλωσσικό μοντέλο, επεξεργασία φυσικής γλώσσας

Η σελίδα αυτή είναι σκόπιμα λευκή.

Abstract

Automatically describing the content of an image is a fundamental problem in artificial intelligence that connects computer vision and natural language processing.

In this thesis, we implement and present a generative model based on deep learning techniques that combine recent advances in computer vision and machine translation and that can be used to generate natural sequences describing an image. More specifically, we use a combination of ***Convolutional Neural Networks*** along with ***Recurrent Neural Networks*** to get the desired results. The model is trained to maximize the likelihood of the target description sentence given the training image.

Experiments on a huge training dataset, like that of ***MSCOCO 2015*** that we used, show the accuracy of the model and the fluency of the language it learns solely from image descriptions. Our model is often quite accurate, which we verify both qualitatively and quantitatively.

Keywords: Image captioning, image processing, convolutional neural networks, recurrent neural networks, language model, natural language processing, LSTM

Η σελίδα αυτή είναι σκόπιμα λευκή.

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Ανδρέα-Γεώργιο Σταφυλοπάτη για την ευκαιρία που μου έδωσε να εκπονήσω την διπλωματική εργασία αυτήν, καθώς και για την εμπιστοσύνη που έδειξε εξ' αρχής στο πρόσωπό μου. Στη συνέχεια, οφείλω να ευχαριστήσω ιδιαιτέρως τον Θάνο Τάγαρη για την συνεχή καθοδήγησή του αλλά και για την άμεση ανταπόκρισή του σε οτιδήποτε χρειαζόμουν κατά τη διάρκεια της συγγραφής της παρούσας διπλωματικής εργασίας. Δίχως την σημαντική του συμβολή, η εκπόνηση της εργασίας δεν θα ήταν εφικτή. Τέλος, θα ήθελα να ευχαριστήσω μέσα από την καρδιά μου την οικογένεια μου αλλά και τους φίλους μου για την αγάπη, την υποστήριξη και την κατανόηση που μου παρείχαν καθ' όλα τα χρόνια της ακαδημαϊκής μου πορείας.

Γεώργιος Βασιλάκης
Σεπτέμβριος 2017

Περιεχόμενα

Κατάλογος Σχημάτων και Πινάκων

1

Εισαγωγή

1.1 Τεχνητή Νοημοσύνη

Τα τελευταία χρόνια ο κλάδος της τεχνητής νοημοσύνη έχει αρχίσει να εισβάλλει όλο και περισσότερο στην καθημερινότητά μας. Με τον όρο αυτόν, όπως τον αντιλαμβάνεται η επιστήμη των υπολογιστών, εννοούμε το πεδίο έρευνας εκείνο στο οποίο μια οποιαδήποτε συσκευή μπορεί να αντιληφθεί το περιβάλλον στο οποίο βρίσκεται και να λάβει τις απαραίτητες ενέργειες προκειμένου να μεγιστοποιήσει την πιθανότητα επιτυχίας της για ένα συγκεκριμένο στόχο. Χρησιμοποιώντας πιο απλές έννοιες, ο όρος "Τεχνητή Νοημοσύνη" βρίσκει εφαρμογή όταν μία μηχανή βρίσκεται σε θέση να μιμηθεί νοητικές λειτουργίες του ανθρώπινου μυαλού, όπως είναι η "μάθηση" και η "επίλυση προβλημάτων".

Ιστορικά, η τεχνητή νοημοσύνη εντοπίζεται ήδη από την αρχαιότητα, σε μύθους και ιστορίες που περιλαμβάνουν τεχνητά όντα προικισμένα με συναίσθηση, ενώ η μεγαλύτερη άνθηση του πεδίου αυτού εντοπίζεται περί τα 1940 μαζί με την εφεύρεση των προγραμματιζόμενων ψηφιακών υπολογιστών, μηχανών βασιζόμενων στην αυθαίρετη ουσία του μαθηματικού λογισμού. Φτάνοντας στο σήμερα, η τεχνητή νοημοσύνη μοιάζει να αποτελεί ένα αναπόσπαστο κομμάτι της επιστήμης των υπολογιστών.

1.1.1 Μηχανική Μάθηση

Ένας υποκλάδος της τεχνητής νοημοσύνης που τα τελευταία χρόνια γνωρίζει μεγάλη ανάπτυξη είναι αυτός της μηχανικής μάθησης. Ως μηχανική μάθηση, ορίζουμε το πεδίο εκείνο της επιστήμης των υπολογιστών, το οποίο δίνει την δυνατότητα στους υπολογιστές να "μαθαίνουν" χωρίς, ωστόσο, να προγραμματίζονται ρητά. Πρόκειται, επί της ουσίας, για ένα επιστημονικό πεδίο προερχόμενο από το συνδυασμό της μελέτης της αναγνώρισης προτύπων και της υπολογιστικής θεωρίας μάθησης στην τεχνητή νοημοσύνη, που ερευνά την μελέτη και την δημιουργία αλγορίθμων, οι οποίοι είναι σε θέση να "μαθαίνουν" και να κάνουν προβλέψεις βάσει δεδομένων, ξεπερνώντας, ουσιαστικά, τον ρητό, στατικό προγραμματισμό.

Η επιστήμη της μηχανικής μάθησης βρίσκει πλέον εφαρμογή σε διάφορα επιστημονικά πεδία που απαιτούν υπολογιστικές εργασίες, όπου ο σχεδιασμός και ο προγραμματισμός ρητών αλγορίθμων με ικανοποιητική απόδοση είναι δύσκολος ή ακόμη και αδύνατος. Παραδείγματα τέτοιων εφαρμογών είναι το φιλτράρισμα των e-mail, η ανίχνευση επιβλαβών εισβολών σε δίκτυα, η οπτική αναγνώριση χαρακτήρων και εφαρμογές στην όραση υπολογιστών.

1.2 Βαθιά Μηχανική Μάθηση & Νευρωνικά Δίκτυα

Μια άλλη αλγοριθμική προσέγγιση η οποία έχει προσελκύσει μεγάλο ενδιαφέρον τα τελευταία χρόνια, προερχόμενη από τον ευρύτερο κλάδο της μηχανικής μάθησης, είναι η Βαθιά Μηχανική Μάθηση η οποία βασίζεται στα Τεχνητά Νευρωνικά Δίκτυα.

Τα τεχνητά νευρωνικά δίκτυα (ΤΝΔ) είναι αρχιτεκτονικές αποτελούμενες από πλήθος διασυνδεδεμένων μονάδων - νευρώνων. Κάθε νευρώνας δέχεται στην είσοδό του σήματα, από το περιβάλλον, αν είναι νευρώνας εισόδου, ή από τις εξόδους των άλλων νευρώνων, και περιλαμβάνει μια συνάρτηση ενεργοποίησης, μέσω της οποίας παράγει ένα σήμα, με το οποίο τροφοδοτεί τους άλλους νευρώνες ή την έξοδο του δικτύου. Κάθε σύναψη μεταξύ δύο νευρώνων χαρακτηρίζεται από μια τιμή βάρους. Οι τιμές των βαρών των συνάψεων αποτελούν τη γνώση που είναι αποθηκευμένη στο δίκτυο και καθορίζουν τη λειτουργία του. Το υπολογιστικό μοντέλο ενός νευρωνικού δικτύου πολλαπλασιάζει τα σήματα-εισόδους κάθε νευρώνα με τα συναπτικά βάρη των συνδέσεων από τις οποίες αυτά διέρχονται και υπολογίζει το άθροισμα των γινομένων αυτών. Το άθροισμα που προκύπτει αποτελεί την είσοδο της συνάρτησης ενεργοποίησης του νευρώνα. Εάν x_i είναι η i-οστή είσοδος του k νευρώνα, w_{ki}: το i-οστό συναπτικό βάρος του k νευρώνα (από έναν αριθμό N συνδέσεων) και η συνάρτηση ενεργοποίησης του νευρώνα, τότε η έξοδος y_k του νευρώνα δίδεται από την εξίσωση:

$$y_k = \phi(\sum_{i=0}^{N-1} x_i w_{ki})$$

Οι πλέον συνήθεις μορφές της συνάρτησης ενεργοποίησης είναι η βηματική, η συνάρτηση προσήμου, η ταυτοτική συνάρτηση - στην περίπτωση γραμμικών νευρώνων – και η σιγμοειδής συνάρτηση.

Τα Νευρωνικά Δίκτυα είναι ένας όρος εμπνευσμένος από την κατανόηση της βιολογικής λειτουργίας του εγκεφάλου μας, αναπαριστώντας, ουσιαστικά, όλες αυτές τις διασυνδέσεις

μεταξύ των νευρώνων. Ωστόσο, σε αντίθεση με ένα βιολογικό εγκέφαλο όπου ένας νευρώνας μπορεί να συνδεθεί σε έναν οποιονδήποτε άλλο νευρώνα εντός μίας συγκεκριμένης απόστασης, τα τεχνητά νευρωνικά δίκτυα έχουν διακριτά επίπεδα, διακριτές συνδέσεις και διακριτές κατευθύνσεις εξάπλωσης των δεδομένων.

Μπορούμε, για παράδειγμα, να πάρουμε μια εικόνα και να την χωρίσουμε σε επιμέρους κομμάτια, τα οποία θα μπουν σαν είσοδοι στο πρώτο επίπεδο του νευρωνικού δικτύου και όπου κάποιοι μεμονωμένοι νευρώνες θα προωθήσουν τα δεδομένα στο δεύτερο επίπεδο. Το δεύτερο επίπεδο νευρώνων θα εκτελέσει κάποιες υπολογιστικές εργασίες και θα προωθήσει τα δεδομένα στο επόμενο επίπεδο, και πάει λέγοντας, ώσπου να φτάσουμε στο τελευταίο επίπεδο όπου παράγεται το τελικό αποτέλεσμα.

Κάθε νευρώνας αποδίδει ένα "βάρος" στα δεδομένα εισόδου του - πόσο "σωστά" ή πόσο "λάθος" είναι σχετικά με την εργασία που πρέπει να υλοποιηθεί. Το τελικό αποτέλεσμα, στη συνέχεια, καθορίζεται από το σύνολο αυτών των βαρών.

Γνωρίζοντας, πλέον, πως λειτουργούν τα τεχνητά νευρωνικά δίκτυα, μπορούμε να ορίσουμε την βαθιά μηχανική μάθηση ως την εφαρμογή "μαθησιακών διεργασιών" στα τεχνητά νευρωνικά δίκτυα τα οποία αποτελούνται από περισσότερα από ένα επίπεδα. Αποτελεί κομμάτι μιας ευρύτερης οικογένειας μεθόδων μηχανικής μάθησης βασιζόμενων στην αναπαράσταση δεδομένων, σε αντίθεση με αλγορίθμους επικεντρωμένους σε υπολογιστικές εργασίες.

Η βαθιά μηχανική μάθηση έχει ενεργοποιήσει αρκετές εφαρμογές της Μηχανικής Μάθησης και κατ' επέκταση του γενικότερου κλάδου της Τεχνητής Νοημοσύνης. Αυτόνομα αυτοκίνητα, καλύτερη προληπτική περίθαλψη, ακόμα και καλύτερες προτάσεις ταινιών βρίσκονται ήδη σε ισχύ ή διαφαίνονται στον ορίζοντα. Η τεχνητή νοημοσύνη μοιάζει να είναι το παρόν και το μέλλον, ενώ με την βοήθεια της Βαθιάς Μηχανικής Μάθησης, μπορεί να φτάσει σε καταστάσεις επιστημονικής φαντασίας που τόσα χρόνια απλά φανταζόμασταν.

1.2.1 Επιβλεπόμενη Μάθηση (Supervised Learning) Τεχνητών Νευρωνικών Δικτύων

Πολλά πρακτικά προβλήματα μπορούν να αναπαρασταθούν με μια απεικόνιση $f : X \to Y$ του υπολογιστή, όπου *X* είναι ο χώρος εισόδων και *Y* ο χώρος εξόδων. Για παράδειγμα, στην Οπτική Αναγνώριση, το *X* θα μπορούσε να είναι ο χώρος των εικόνων και *Y* το διάστημα [0, 1] που υποδεικνύει την πιθανότητα μια γάτα να εμφανίζεται κάπου στην εικόνα. Ωστόσο, τις περισσότερες φορές είναι αρκετά δύσκολο να ορίσουμε ακριβώς την

συνάρτηση *f*. Η ιδέα της Επιβλεπόμενης Μάθησης προσφέρει μια εναλλακτική προσέγγιση στο πρόβλημα αυτό, η οποία εκμεταλλεύεται το γεγονός ότι είναι σχετικά εύκολο να αποκτήσουμε παραδείγματα $(x, y)\epsilon X \times Y$ της επιθυμητής απεικόνισης. Στο παράδειγμά μας, αυτό θα αντιστοιχούσε στη συλλογή ενός συνόλου δεδομένων, το οποίο θα αποτελούνταν από εικόνες και την αντίστοιχη σημείωση του αν υπάρχει γάτα στην εικόνα ή όχι.

Συγκεκριμένα, υποθέτουμε ένα σύνολο δεδομένων εκπαίδευσης με *n* παραδείγματα $\{(x_1, y_1), ..., (x_n, y_n)\}$, τα οποία αποτελούνται από ανεξάρτητα και ταυτοτικά κατανεμημένα (*identically distributed*) δείγματα, από μια κατανομή δεδομένων *D*. Μπορούμε, τότε, να εκπαιδεύσουμε την απεικόνιση $f : X \to Y$ με το να ψάχνουμε μέσα από ένα σύνολο συναρτήσεων για την συνάρτηση εκείνη η οποία είναι η πιο σύμφωνη με τα παραδείγματά εκπαίδευσής μας. Πιο συγκεκριμένα, θεωρούμε μια συγκεκριμένη κλάση από συναρτήσεις *F* και επιλέγουμε μία συνάρτηση κόστους $L(\hat{y}, y)$, η οποία μετράει την διαφορά μεταξύ της πρόβλεψης $\hat{y}_i = f(x_i)$ για κάποιο $f\epsilon F$ και της πραγματικής τιμής y_i**.** Ο σκοπός μας είναι να να βρούμε την *f* εκείνη για την οποία ιδανικά ικανοποιείται η σχέση:

$$f^* = \arg\min_{f\epsilon F} E_{(x,y)\sim D} L(f(x), y)$$

Με άλλα λόγια, αναζητούμε μία συνάρτηση *f* η οποία θα ελαχιστοποιεί την αναμενόμενη απώλεια της κατανομής δεδομένων *D*. Σε πρακτικές εφαρμογές, μόλις βρούμε τη ζητούμενη *f*, μπορούμε να "ξεφορτωθούμε" τα δεδομένα εκπαίδευσης και να κρατήσουμε μόνον την συνάρτηση *f**, την οποία θα χρησιμοποιούμε για να κάνουμε την αντιστοίχιση των στοιχείων από το *X* στο *Y*.

Η πιο ευρέως χρησιμοποιούμενη συνάρτηση κόστους, την οποία χρησιμοποιούμε και εμείς στο σύστημά μας, ονομάζεται απώλεια διασχιζόμενης εντροπίας *(cross-entropy loss),* η οποία έχει τη μορφή:

$$L(\hat{y}, y) = -\sum_{k=1}^{K} y_k log\hat{y}_k = -log\hat{y}_{y=1}$$

,

όπου η πρώτη ισότητα είναι ο ορισμός της διασχιζόμενης εντροπίας μεταξύ δύο κατανομών $H(p, q) = -\sum_x p(x) log q(x)$ και η δεύτερη ισότητα απλοποιεί την έκφραση.

1.3 Επεξεργασία Εικόνων

Σύμφωνα με την επιστήμη της Απεικόνισης, ο όρος "Επεξεργασία Εικόνων" αναφέρεται στην επεξεργασία που υποβάλλονται οι εικόνες, με τη βοήθεια μαθηματικών λειτουργιών και κάνοντας χρήση οποιασδήποτε μορφής επεξεργασίας σήματος, χρησιμοποιώντας ως είσοδο μια εικόνα, μια σειρά από εικόνες ή ένα βίντεο. Η έξοδος της επεξεργασίας αυτής μπορεί να είναι μια εικόνα ή ένα σύνολο χαρακτηριστικών ή παραμέτρων που σχετίζονται με την εικόνα αυτή. Οι περισσότερες τεχνικές επεξεργασίας εικόνων περιλαμβάνουν απομόνωση των μεμονωμένων επιπέδων χρωμάτων μιας εικόνας, μεταχείρισή τους σαν δισδιάστατα σήματα και εφαρμογή συμβατικών τεχνικών επεξεργασίας σήματος σε αυτά.

1.3.1 Αυτόματη Δημιουργία Περιγραφών Εικόνων

Η αυτόματη περιγραφή του περιεχομένου μιας εικόνας αποτελεί ένα σημαντικό πρόβλημα της τεχνητής νοημοσύνης, το οποίο συνδέει το πεδίο της Όρασης Υπολογιστών με αυτό της Επεξεργασίας Φυσικής Γλώσσας. Η ικανότητα ενός συστήματος να μπορεί, αυτόματα, να δημιουργεί περιγραφές του περιεχομένου των εικόνων γεννώντας σωστές, συντακτικά και σημασιολογικά, προτάσεις, φαίνεται να είναι μία αρκετά απαιτητική πρόκληση, η οποία, ωστόσο, θα μπορούσε να έχει μεγάλη επίδραση, όπως για παράδειγμα στο να βοηθήσει ανθρώπους με μειωμένη όραση να αντιληφθούν καλύτερα το περιεχόμενο εικόνων που υπάρχουν στο διαδίκτυο. Το συγκεκριμένο πρόβλημα είναι πολύ δυσκολότερο από το άκρως μελετημένο, για παράδειγμα, πρόβλημα της ταξινόμησης μιας εικόνας ή από προβλήματα που σχετίζονται με εντοπισμό αντικειμένων σε εικόνες, με τα οποία ασχολείται κυρίως η κοινότητα της Όρασης Υπολογιστών. Και αυτό. διότι μία περιγραφή πρέπει, όχι μόνον να εντοπίσει τα επιμέρους αντικείμενα σε μία εικόνα, αλλά πρέπει, επίσης, να εκφράσει πώς αυτά τα αντικείμενα συνδέονται μεταξύ τους, αλλά με ποια χαρακτηριστικά και δραστηριότητες αυτά σχετίζονται. Επιπλέον, οι παραπάνω σημασιολογικές πληροφορίες πρέπει να εκφραστούν σε μία φυσική γλώσσα, όπως είναι τα Αγγλικά, που σημαίνει ότι χρειάζεται και ένα μοντέλο της φυσικής γλώσσας επιπρόσθετα από την οπτική κατανόηση της εικόνας.

1.3.1.1 Σχετικές Προσεγγίσεις

Οι περισσότερες προσεγγίσεις για τη λύση του προβλήματος αυτού προσπαθούν να ενοποιήσουν ήδη υπάρχουσες, επιμέρους λύσεις των προαναφερθέντων υποπροβλημάτων προκειμένου να μεταβούν από μία εικόνα στην περιγραφή της.

Οι προσεγγίσεις που έχουν ακολουθηθεί από την κοινότητα της Όρασης Υπολογιστών έχει παραδοσιακά οδηγήσει πολύπλοκα συστήματα, τα οποία αποτελούνται από πρωτόγονους οπτικούς αναγνωριστές *(visual primitive recognizers),* συνδυαζόμενοι με μια δομημένη επίσημη γλωσσα, όπως για παράδειγμα And-Or γράφοι ή λογικά συστήματα που, στην συνέχεια, μετατρέπονται σε φυσική γλώσσα μέσω συστημάτων βασισμένων σε κανόνες *(rule-based systems).* Τέτοια συστήματα, ωστόσο, είναι υπερβολικά ρητώς προγραμματιζόμενα , σχετικά εύθραυστα και χρησιμοποιούνται σε πολύ λίγους τομείς.

Το πρόβλημα της δημιουργίας περιγραφών περιεχομένων των εικόνων έχει προσελκύσει και το ενδιαφέρον των ανθρώπων που βρίσκονται στο πεδίο της Επεξεργασίας Φυσικής Γλώσσας. Τελευταίες πρόοδοι στην αναγνώριση και στον εντοπισμό αντικειμένων και χαρακτηριστικών, έχουν οδηγήσει στην χρησιμοποίησή τους σε συστήματα που "γεννούν" φυσική γλώσσα, αν και είναι κάπως περιορισμένα στην εκφραστικότητά τους. Τέτοιες προσεγγίσεις, ωστόσο, αν και είναι ικανές να εντοπίσουν τα επιμέρους αντικείμενα των εικόνων, δυσκολεύονται αρκετά στην δημιουργία του κειμένου.

1.3.1.2 Αντικείμενο Διπλωματικής

Στην παρούσα διπλωματική εργασία συνδυάζουμε βαθιά συνελικτικά δίκτυα ***(CNN)*** για την εξαγωγή χαρακτηριστικών των εικόνων και ανατροφοδοτούμενα δίκτυα ***(RNN)*** υπεύθυνα για την μοντελοποίηση των προτάσεων, ώστε να δημιουργήσουμε ένα ενιαίο δίκτυο το οποίο θα παράγει περιγραφές των εικόνων. Το μοντέλο που θα χρησιμοποιήσουμε είναι εμπνευσμένο από πρόσφατες επιτυχίες στην παραγωγή προτάσεων στο πεδίο της Μετάφρασης Μηχανών, με την διαφορά ότι αντί να ξεκινάμε με μία πρόταση σαν είσοδο, παρέχουμε στο σύστημά μας μια εικόνα επεξεργασμένη από ένα συνελικτικό δίκτυο ***(ConvNet)***.

2

Βαθιά Συνελικτικά Νευρωνικά Δίκτυα (CNN)

2.1 Ορισμός & Λειτουργία

Τα Βαθιά Συνελικτικά Νευρωνικά Δίκτυα (CNNs ή ConvNets) είναι νευρωνικές αρχιτεκτονικές δικτύων, ειδικά σχεδιασμένες για τη διαχείριση δεδομένων με κάποια χωρική τοπολογία (π.χ. εικόνες, βίντεο, φασματόγραμμα ήχου στην επεξεργασία φωνής, ακολουθίες χαρακτήρων σε κείμενο). Είναι μια υποκατηγορία των Τεχνητών Νευρωνικών Δικτύων τα οποία έχουν αποδειχθεί ότι είναι πολύ αποτελεσματικά σε πεδία όπως η αναγνώριση και ταξινόμηση εικόνων, με συγκεκριμένες επιτυχίες στην αναγνώριση προσώπων,αντικειμένων και φωτεινών σηματοδοτών, ενώ επίσης παρέχουν όραση σε ρομπότ και αυτοκινούμενα οχήματα. Τα δίκτυα αυτά αποτελούνται από νευρώνες οι οποίοι έχουν εκπαιδεύσιμα βάρη(weights) και κλίσεις(biases). Κάθε νευρώνας δέχεται κάποιες εισόδους, εκτελεί ένα εσωτερικό γινόμενο και, προαιρετικά, ακολουθείται από μία συνάρτηση μη-γραμμικότητας (non-linearity function). Το συνολικό δίκτυο, ωστόσο, εκφράζει μία ατομικά διαφορίσιμη συνάρτηση αποτελέσματος: από τα ανεπεξέργαστα *pixel* της εικόνας από τη μία, στα αποτελέσματα κλάσεων από την άλλη. Τέλος, ακολουθείται από μία συνάρτηση απωλειών.

2.2 Επισκόπηση Αρχιτεκτονικής

Ένα Βαθύ Συνελικτικό Νευρωνικό Δίκτυο αποτελείται από έναν αριθμό από συνελικτικά (*convolutional*) και υποδειγματοληπτικά (*subsampling*) επίπεδα, τα οποία, προαιρετικώς, ακολουθούνται από πλήρως συνδεδεμένα επίπεδα (*fully-connected layers*). Η είσοδος σε ένα

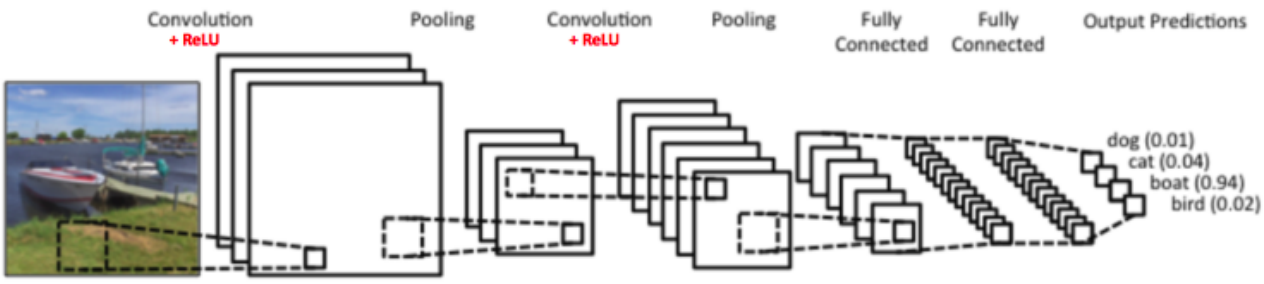
τέτοιο δίκτυο είναι, συνήθως, ένας πίνακας 3 διάστασεων, τον οποίον απο εδώ και πέρα θα αποκαλούμε τένσορα. Για παράδειγμα, μια εικόνα μπορεί να αναπαρασταθεί σαν έναν τένσορα 3 διαστάσεων H x W x 3, όπου το H (*height*) αντιστοιχεί στον αριθμό των pixel της εικόνας στον κάθετο άξονα, το W (*width*) αντιστοιχεί στον αριθμό των pixel της εικόνας στον οριζόντιο άξονα και το 3 αναφέρεται στα τρία κανάλια χρωμάτων RGB (*Red, Green, Blue*). Τένσορες με περισσότερες απο τρεις διαστάσεις μπορούν να διαχειριστούν από ένα *CNN* με παρόμοιο τρόπο. Η είσοδος, στη συνέχεια, περνάει ακολουθιακά από μια σειρά επεξεργασιών. Ένα βήμα επεξεργασίας, συνήθως, αποκαλείται επίπεδο, το οποίο θα μπορούσε να είναι ένα συνελικτικό επίπεδο (*convolutional layer*), ένα συγκεντρωτικό επίπεδο (*pooling layer*), ένα επίπεδο κανονικοποίησης (*normalization layer*), ένα πλήρως συνδεδεμένο επίπεδο (*fully connected layer*) ή ένα επίπεδο απωλειών (*loss layer*). Θα περιγράψουμε λεπτομερώς τα επίπεδα αυτά στην συνέχεια.

Προς το παρόν θα δώσουμε μία αφηρημένη περιγραφή της δομής ενός βαθιού συνελικτικού δικτύου:

$$x^1 \rightarrow \left[w^1\right] \rightarrow x^2 \rightarrow \ldots \rightarrow x^{L-1} \rightarrow \left[w^{L-1}\right] \rightarrow x^L \rightarrow \left[w^L\right] \rightarrow z \quad (1)$$

Η παραπάνω εξίσωση (1) επεξηγεί πώς λειτουργεί ένα CNN ανά επίπεδο για προωθητικό πέρασμα (*forward pass*). Η είσοδος είναι το x^1, συνήθως μία εικόνα τριών διαστάσεων, η οποία επεξεργάζεται από το πρώτο επίπεδο, το οποίο είναι το πρώτο κουτάκι της παραπάνω εξίσωσης. Υποδηλώνουμε τις παραμέτρους που συμπεριλαμβάνονται στην επεξεργασία του πρώτου επιπέδου συλλογικά με τον τένσορα w^1 . Η έξοδος του πρώτου επιπέδου είναι το x^2 , το οποίο με την σειρά του λειτουργεί ως είσοδος στο δεύτερο επίπεδο επεξεργασίας.

Αυτός ο τρόπος επεξεργασίας συνεχίζεται ώσπου ολοκληρωθούν όλα τα επίπεδα επεξεργασίας του CNN, το οποίο δίνει σαν έξοδο το x^L. Ένα επιπλέον επίπεδο, ωστόσο, προστίθεται για μία *οπισθοδιάδοση σφάλματος (backward error propagation)*, μια μέθοδος που εκπαιδεύει "καλά" τις παραμέτρους του CNN και που θα την αναλύσουμε εκτενέστερα στην συνέχεια.



Σχήμα 2.1 Παράδειγμα αρχιτεκτονικής ενός βαθιού νευρωνικού συνελικτικού δικτύου που αποτελείται από 2 συνελικτικά επίπεδα, 2 συγκεντρωτικά επίπεδα και δύο πλήρως συνδεδεμένα δεχόμενο ως είσοδο μία εικόνα.

Αν υποθέσουμε ότι η εξίσωση (1) παρουσιάζει ένα πρόβλημα ταξινόμησης εικόνων με C κλάσεις, τότε μία συνήθης στρατηγική είναι να απεικονίζουμε την έξοδο x^L σαν ένα C-διάστατο διάνυσμα, του οποίου η *i-οστή* είσοδος κωδικοποιεί την πρόβλεψη (οπίσθια πιθανότητα το x^l να έρχεται από την *i-οστή* κλάση).

Το τελευταίο επίπεδο είναι ένα επίπεδο απωλειών. Αν υποθέσουμε ότι *t* είναι η αντίστοιχη πραγματική τιμή (ground-truth) για είσοδο x^l , τότε μπορεί να χρησιμοποιηθεί μία συνάρτηση κόστους ή απωλειών προκειμένου να μετρήσουμε την διαφορά μεταξύ της προβλέψεως του CNN x^L και της τιμής - στόχου *t*. Για παράδειγμα, μία απλή συνάρτηση απωλειών θα μπορούσε να είναι η εξής:

$$z = \frac{1}{2}(\|t - x^L\|)^2 \quad , \qquad (2)$$

αν και συνήθως χρησιμοποιούνται πιο πολύπλοκες συναρτήσεις απωλειών. Αυτή η συνάρτηση θα μπορούσε να χρησιμοποιηθεί σε ένα αναδρομικό πρόβλημα *(regression)*. Σε ένα πρόβλημα ταξινόμησης *(classification)*, η συνάρτηση απωλειών που συνήθως χρησιμοποιείται είναι αυτή της διασχιζόμενης εντροπίας (*cross-entropy*). Η πραγματική τιμή σε ένα πρόβλημα ταξινόμησης είναι μια κατηγορική μεταβλητή *t*. Αρχικά, μετατρέπουμε την κατηγορική αυτή μεταβλητή σε ένα C-διάστατο διάνυσμα. Τώρα, και το t και το x^L είναι συναρτήσεις μάζας πυκνότητας πιθανότητας και η απώλεια διασχιζόμενης εντροπίας μετράει την μεταξύ του απόσταση. Συνεπώς, μπορούμε να ελαχιστοποιήσουμε την διασχιζόμενη εντροπία. Η εξίσωση (1) μοντελοποιεί ρητά την συνάρτηση απωλειών σαν ένα επίπεδο απωλειών, του οποίου η επεξεργασία μοντελοποιείται σαν ένα κουτί με παραμέτρους w^L.

Αξίζει να σημειώσουμε ότι μερικά επίπεδα μπορεί να μην έχουν καθόλου παραμέτρους, που σημαίνει ότι το w^i μπορεί να είναι κενό για κάποια *i*. Ένα τέτοιο παράδειγμα είναι το επίπεδο *softmax*.

2.3 Επίπεδα Επεξεργασίας

Έχοντας κατανοήσει την αρχιτεκτονική των βαθιών συνελικτικών νευρωνικών δικτύων, θα συνεχίσουμε με την ανάλυση των διαφόρων επιπέδων επεξεργασίας, ξεκινώντας από το επίπεδο *ReLU*, το οποίο είναι το απλούστερο επίπεδο μεταξύ αυτών που θα αναλύσουμε στην συγκεκριμένη ενότητα. Πριν προχωρήσουμε στην ανάλυση, θα δώσουμε σημασία σε κάποια σύμβολα.

Έστω ότι βρισκόμαστε στο *l*-οστό επίπεδο, του οποίου η είσοδος είναι ένας τρισδιάστατος τένσορας x^l με το $x^l \epsilon \mathbb{R}^{H^l \times W^l \times D^l}$. Έτσι, χρειαζόμαστε μία τριπλέτα (i^l, j^l, d^l) προκειμένου να εντοπίσουμε κάποιο συγκεκριμένο στοιχείο του x^l . Η τριπλέτα (i^l, j^l, d^l) αναφέρεται σε ένα στοιχείο του x^l, το οποίο βρίσκεται στο d^l*-οστό* κανάλι και σε χωρική θέση (i^l, j^l). Στο *l-οστό* επίπεδο, μία συνάρτηση θα μετατρέψει την είσοδο $\boldsymbol{x^l}$ σε μία έξοδο $\boldsymbol{y}$, η οποία με τη σειρά της θα αποτελέσει την είσοδο του επόμενου επιπέδου. Συνεπώς, αξίζει να σημειώσουμε ότι το *y* και το x^{l+1} αναφέρονται, στην πραγματικότητα, στο ίδιο αντικείμενο.

2.3.1 Επίπεδο ReLU

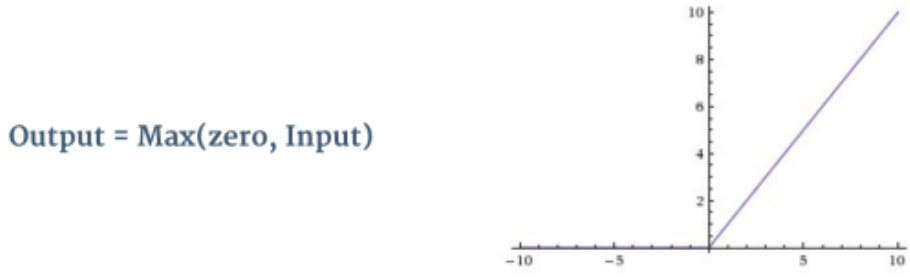
Το επίπεδο επεξεργασίας *ReLU* (*Rectified Linear Unit*) δεν επηρεάζει το μέγεθος της εισόδου, πράγμα που σημαίνει ότι το x^l και το y έχουν το ίδιο μέγεθος. Στην παργματικότητα, η ReLU μπορεί να θεωρηθεί σαν μια αποκοπή που εφαρμόζεται ατομικά σε κάθε στοιχείο της εισόδου:

$$y_{i,j,d} = max\left\{0, x^l_{i,j,d}\right\} \tag{3}$$

Στο επίπεδο αυτό δεν υπάρχουν παράμετροι και συνεπώς δεν υπάρχει ανάγκη για εκπαίδευση των παραμέτρων του επιπέδου αυτού.

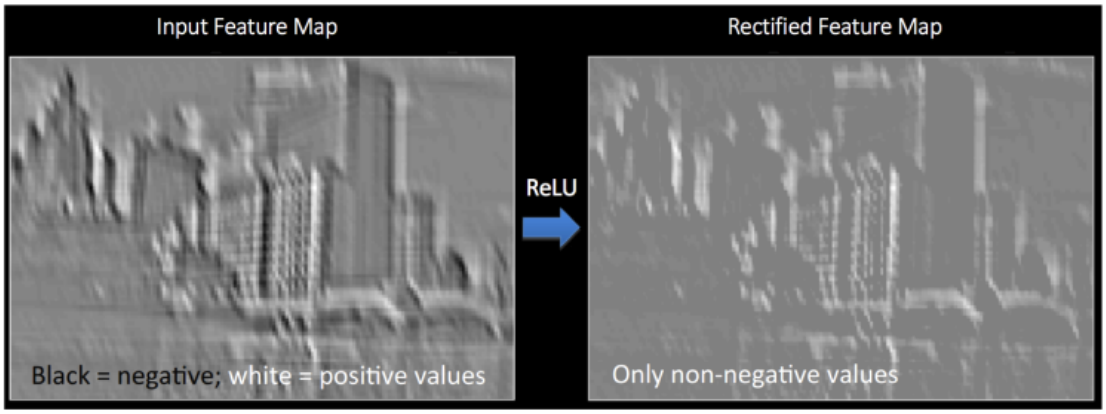
Ο σκοπός του επιπέδου ReLU είναι να αυξήσει την μη-γραμμικότητα του CNN. Εφ' όσον οι σημασιολογικές πληροφορίες μιας εικόνας (π.χ. ένας άνθρωπος και ένα σκυλί Χάσκι κάθονται δίπλα ο ένας στον άλλον σε έναν παγκάκι μέσα σε έναν κήπο) είναι, προφανώς, μια αρκετά μη-γραμμική απεικόνιση τιμών pixel στην είσοδο, θέλουμε η αντιστοίχηση της

εισόδου του CNN στην έξοδό του να είναι, επίσης, αρκετά μη-γραμμική. Η συνάρτηση ReLU, αν και αρκετά απλή, είναι μία μη-γραμμική συνάρτηση, όπως φαίνεται και στο Σχήμα 2.2.



Σχήμα 2.2 Η συνάρτηση ReLU

Αν θεωρήσουμε το $x_{i,j,d}^{l}$ σαν ένα από τα $H^l W^l D^l$ χαρακτηριστικά που εξάχθηκαν από τα επίπεδα 1 έως l-1 του CNN, τότε αυτό μπορεί να έχει θετική, αρνητική ή μηδενική τιμή. Για παράδειγμα, το $x_{i,j,d}^{l}$ μπορεί να είναι θετικό εάν μία περιοχή της εικόνας εισόδου έχει συγκεκριμένο μοτίβο (όπως το κεφάλι ενός σκύλου ή μιας γάτας) . Διαφορετικά, μπορεί να είναι αρνητικό ή μηδενικό εάν η περιοχή αυτή δεν παρουσιάζει τέτοια μοτίβα. Το επίπεδο ReLU θα θέσει όλες τις αρνητικές τιμές ίσες με το μηδέν, που σημαίνει ότι το $y_{i,j,d}^{l}$ θα ενεργοποιηθεί μόνο για εικόνες που διαθέτουν τέτοια μοτίβα σε μία συγκεκριμένη περιοχή. Διαισθητικά, αυτή η ιδιότητα είναι χρήσιμη για την αναγνώριση περίπλοκων μοτίβων και πραγμάτων. Για παράδειγμα, το να πούμε ότι μια εικόνα περιέχει "ένα κεφάλι γάτας" μόνο από το γεγονός ότι ένα συγκεκριμένο χαρακτηριστικό ενεργοποιήθηκε και το μοτίβο αυτού του χαρακτηριστικού μοιάζει με κεφάλι γάτας, είναι από μόνο του ασθενής απόδειξη. Ωστόσο, αν βρούμε πολλά ενεργοποιημένα χαρακτηριστικά μετά το επίπεδο ReLU των οποίων τα μοτίβα αντιστοιχούν σε κεφάλι γάτας, μπορούμε να πούμε με μεγαλύτερη αυτοπεποίθηση ότι όντως υπάρχει μια γάτα στην εικόνα εισόδου. Η λειτουργία της συνάρτησης ReLU μπορεί να γίνει πιο κατανοητή με την βοήθεια του σχήματος 2.3



Σχήμα 2.3 Λειτουργία της συνάρτησης ReLU

Υπάρχουν και άλλες συναρτήσεις που έχουν χρησιμοποιηθεί στα νευρωνικά δίκτυα για να παράγουν μη-γραμμικότητα , όπως είναι για παράδειγμα η λογιστική σιγμοειδής συνάρτηση. Ωστόσο, η συνάρτηση αυτή αποδίδει αρκετά χειρότερα από την ReLU στην εκπαίδευση των CNN.

2.3.2 Συνελικτικό Επίπεδο

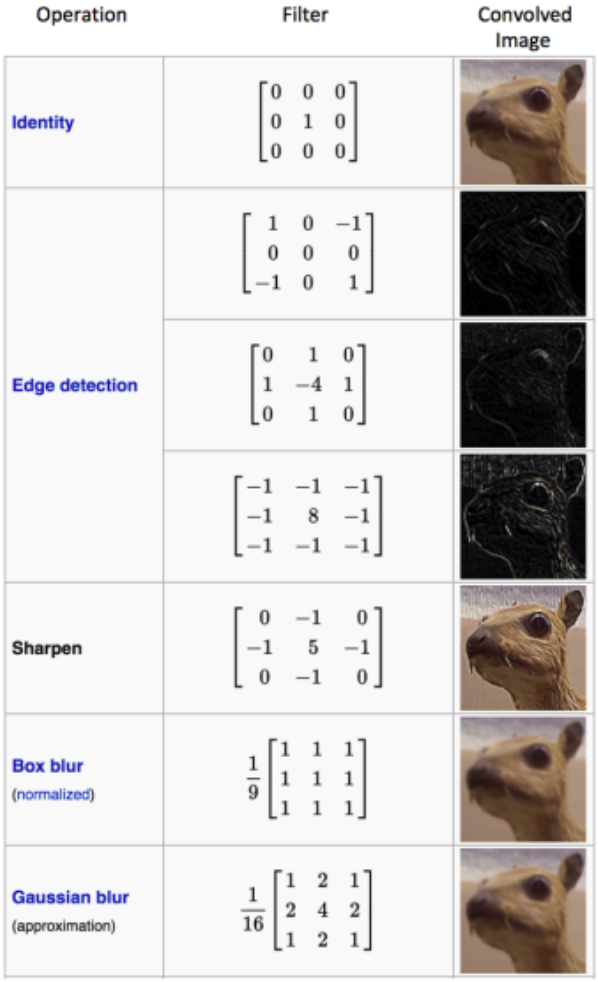
Το συνελικτικό επίπεδο είναι η βασική μονάδα κατασκευής ενός Συνελικτικού Δικτύου, το οποίο εκτελεί και τους πιο απαιτητικούς υπολογισμούς. Ο κύριος σκοπός του επιπέδου αυτού είναι η εξαγωγή χαρακτηριστικών από την εικόνα εισόδου.

Οι παράμετροι του επιπέδου αυτού αποτελούνται από ένα σύνολο από εκπαιδεύσιμα φίλτρα. Κάθε φίλτρο είναι χωρικά μικρό (ως προς το ύψος και το πλάτος), αλλά εκτείνεται σε όλο το βάθος του όγκου της εισόδου. Για παράδειγμα, ένα τυπικό φίλτρο στο πρώτο επίπεδο ενός Συνελικτικού Δικτύου μπορεί να έχει μέγεθος *5x5x3* (5 *pixels* για το πλάτος, 5 *pixels* για το ύψος και 3 για τον αριθμό των καναλιών μίας έγχρωμης εικόνας *RGB*). Κατά το προωθητικό πέρασμα συνελίσσουμε κάθε φίλτρο σε όλο τον όγκο της εισόδου και υπολογίζουμε τα εσωτερικά γινόμενα μεταξύ των τιμών του φίλτρου και των τιμών της εισόδου σε οποιαδήποτε θέση. Καθώς περνάμε το φίλτρο κατά ύψος και κατά πλάτος του πίνακα εισόδου, παράγεται ένας δισδιάστατος πίνακας ενεργοποίησης ο οποίος αποδίδει τις τιμές απόκρισης του φίλτρου σε κάθε χωρική θέση. Διαισθητικά, το δίκτυο θα θα εκπαιδευθεί σε φίλτρα τα οποία ενεργοποιούνται όταν βλέπουν κάποιον τύπο οπτικών χαρακτηριστικών όπως είναι η ακμή κάποιου προσανατολισμού ή η κηλίδα κάποιου χρώματος στο πρώτο επίπεδο. Έτσι έχουμε, πλέον, αποκτήσει ένα ολόκληρο σύνολο από φίλτρα σε κάθε συνελικτικό

επίπεδο, κάθε ένα από τα οποία θα παράγει έναν δισδιάστατο πίνακα ενεργοποίησης. Θα στοιβάξουμε αυτούς τους πίνακες ενεργοποίησης κατα την τρίτη δίασταση (βάθος) και εν τέλει θα αποκτήσουμε την τρισδιάστατη έξοδο.

Κάνοντας την παραλληλοποίηση με την λειτουργία του εγκεφάλου, μπορούμε να να φανταστούμε ότι κάθε τιμή του τρισδιάστατου πίνακα εξόδου μπορεί να μεταφραστεί σαν μία έξοδο ενός νευρώνα ο οποίος κοιτάζει μόνο μία μικρή περιοχή της εισόδου και μοιράζεται τις παραμέτρους του με όλους τους νευρώνες που βρίσκονται δεξιά και αριστερά του.

Στην συνέχεια θα αναλύσουμε τις λεπτομέρειες των συνδέσεων μεταξύ των νευρώνων, την διάταξή τους στο χώρο και το σχήμα μοιράσματος παραμέτρων που χρησιμοποιούν.



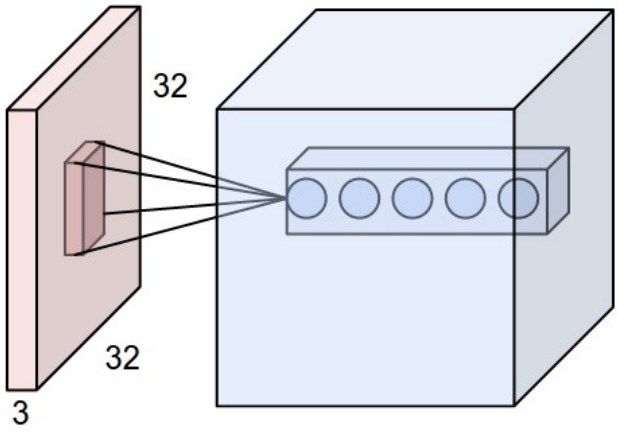
Σχήμα 2.4 Διαφορετικά φίλτρα εξάγουν διαφορετικά χαρακτηριστικά μιας εικόνας

2.3.2.1 Τοπική Σύνδεση Μεταξύ Νευρώνων

Όταν αντιμετωπίζουμε εισόδους πολλών διαστάσεων όπως οι εικόνες, δεν είναι πρακτική η σύνδεση όλων των νευρώνων με όλους του προηγούμενου όγκου. Αντιθέτως, θα θέλαμε να συνδέσουμε κάθε νευρώνα με μία μικρή μόνον περιοχή του όγκου εισόδου. Η χωρική επέκταση της διασύνδεσης αυτής, αποτελεί μία υπερπαράμετρο του δικτύου και ονομάζεται *δεκτικό πεδίο του νευρώνα* ή πιο απλά *μέγεθος φίλτρου*. Η επέκταση της διασύνδεσης αυτής κατα βάθος είναι πάντα ίση με το βάθος του όγκου εισόδου. Είναι σημαντικό να τονίσουμε την ασυμμετρία στο πως χειριζόμαστε τις χωρικές διαστάσεις (ύψος και πλάτος) και την διάσταση του βάθους: Οι συνδέσεις είναι τοπικές στο χώρο (κατά ύψος και κατά πλάτος), αλλά πάντα εφαρμόζεται κατά το συνολικό βάθος του όγκου εισόδου.

Για παράδειγμα, ας υποθέσουμε ότι ο όγκος εισόδου έχει διαστάσεις 32x32x3. Αν το δεκτικό πεδίο του νευρώνα είναι 5x5, τότε κάθε νευρώνας στο Συνελικτικό Επίπεδο θα έχει βάρη σε μία [5x5x3] περιοχή του όγκου εισόδου και συνολικά 5*5*3 = 75 τιμές βάρους.

Για ένα άλλο παράδειγμα ας υποθέσουμε ότι ο όγκος εισόδου είναι [16x16x20]. Τότε, χρησιμοποιώντας ένα μέγεθος φίλτρου 3x3, κάθε νευρώνας στο Συνελικτικό Επίπεδο θα έχει συνολικά 3*3*20 = 180 διασυνδέσεις με τον όγκο εισόδου. Αξίζει να δούμε και πάλι, ότι ενώ η σύνδεση είναι σε τοπικό επίπεδο (3x3), εφαρμόζεται σε όλο το βάθος εισόδου (20).



Σχήμα 2.5 Ένα παράδειγμα εισόδου διαστάσεων 32x32x3 και ένα παράδειγμα ενός συνόλου νευρώνων στο πρώτο συνελικτικό επίπεδο. Κάθε νευρώνας στο συνελικτικό επίπεδο συνδέεται μόνο με μία τοπική

περιοχή του όγκου εισόδου χωρικά, αλλά σε όλο το βάθος. Παρατηρούμε ότι υπάρχουν 5 νευρώνες κατά βάθος.

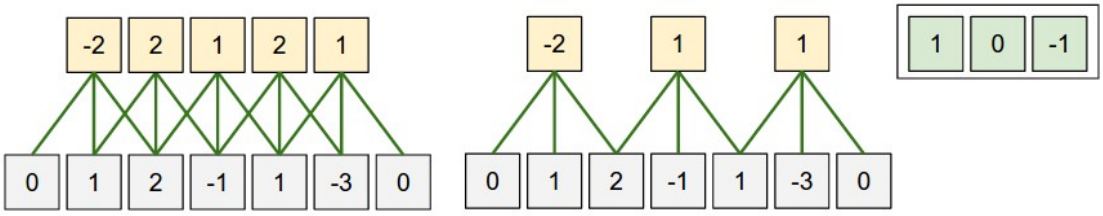
2.3.2.2 Χωρική Διάταξη

Στο κομμάτι αυτό θα δούμε πόσοι νευρώνες βρίσκονται στην έξοδο του συνελικτικού επιπέδου και πως αυτοί συνδέονται. Για τον λόγο αυτόν θα ορίσουμε τρεις υπερπαραμέτρους που ελέγχουν το μέγεθος του όγκου εξόδου: το ***βάθος***, το ***βήμα*** και το ***γέμισμα με μηδενικά***.

- Αρχικά, το ***βάθος*** του όγκου εξόδου αποτελεί μία υπερπαράμετρο που αντιστοιχεί στον αριθμό των φίλτρων που θα θέλαμε να χρησιμοποιήσουμε και κάθε ένα από τα οποία θα εξετάζει κάποιο διαφορετικό χαρακτηριστικό της εισόδου. Για παράδειγμα, αν το πρώτο συνελικτικό επίπεδο ενός δικτύου δέχεται σαν είσοδο μια εικόνα, τότε διαφορετικοί νευρώνες κατά την διάσταση του βάθους μπορεί να ενεργοποιηθούν παρουσία διαφόρων ακμών προσανατολισμού ή κηλίδων χρωμάτων. Θα αναφερόμαστε σε ένα σύνολο νευρώνων που εξετάζουν όλοι την ίδια περιοχή της εισόδου σαν μια ***στήλη βάθους***.
- Στη συνέχεια, θα ορίσουμε το ***βήμα*** με το οποίο "μετακινούμε" το φίλτρο. Είναι ο αριθμός των pixel κατά τον οποίο μετακινούμε τον πίνακα φίλτρου μας πάνω στον πίνακα εισόδου. Όταν το βήμα έχει την τιμή 1, τότε μετακινούμε το φίλτρο κατά 1 pixel τη φορά. Όταν το βήμα έχει την τιμή 2, τότε τα φίλτρα περνούν τον πίνακα εισόδου ανά 2 pixel την φορά. Σε αυτήν την περίπτωση, θα παραχθούν μικρότεροι, χωρικά, όγκοι εξόδου.
- Κάποιες φορές, όπως θα δούμε και στην συνέχεια, είναι βολικό να "γεμίζουμε" τον όγκο εισόδου με μηδενικά γύρω από τα σύνορα. Το μέγεθος αυτού του ***γεμίσματος με μηδενικά,*** αποτελεί μία υπερπαράμετρο, η οποία μας δίνει το δικαίωμα να ελέγχουμε το χωρικό μέγεθος των όγκων εξόδου.

Μπορούμε να υπολογίσουμε το χωρικό μέγεθος του όγκου εξόδου σαν μία συνάρτηση του μεγέθους του όγκου εισόδου, έστω ***W,*** του μεγέθους του δεκτικού πεδίου του συνελικτικού επιπέδου, έστω ***F***, του βήματος του οποίου εφαρμόζεται, έστω ***S*** και του αριθμού των μηδενικών τα οποία χρησιμοποιήθηκαν για γέμισμα στα σύνορα, έστω ***P*** . Τότε, η σχέση που

αποδίδει τον αριθμό των νευρώνων είναι ***(W - F + 2P)/S +1.*** Για παράδειγμα, για μία είσοδο 7x7 και για ένα φίλτρο 3x3, με βήμα 1 γέμισμα μηδενικών ίσο με το μηδέν, θα πάρουμε σαν έξοδο έναν 5x5 πίνακα. Ένα παράδειγμα ακολουθεί στο σχήμα 2.6



Σχήμα 2.6 Εικονογράφηση μιας χωρικής διάταξης. Στο συγκεκριμένο παράδειγμα υπάρχει μόνο μία χωρική διάσταση (x) ένας νευρώνας με δεκτικό πεδίο F=3, μέγεθος εισόδου W=5 και γέμισμα μηδενικών P=1.

2.3.2.3 Σύνοψη Συνελικτικού Επιπέδου

Συνοψίζοντας για το συνελικτικό επίπεδο, μπορούμε να πούμε τα εξής:

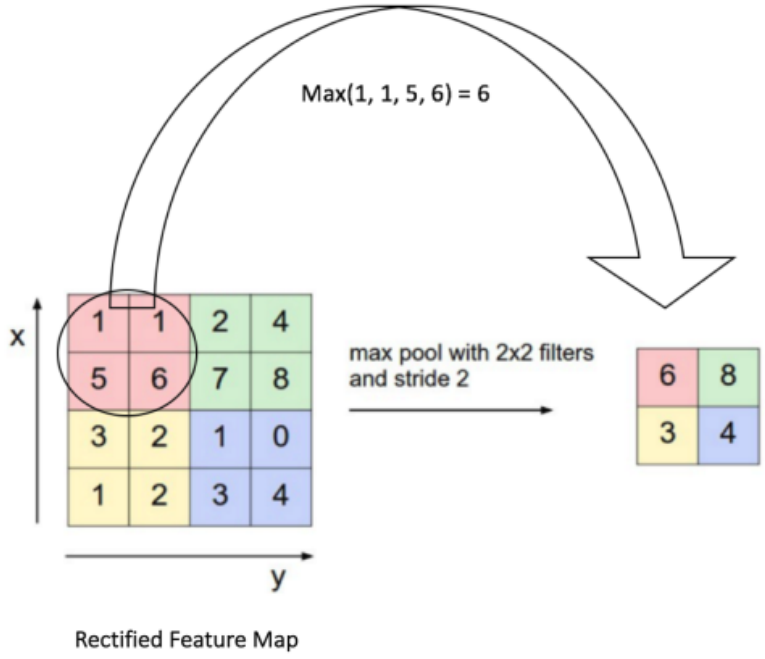
- Δέχεται ως είσοδο έναν όγκο μεγέθους $W_1 \times H_1 \times D_1$
- Απαιτεί 4 υπερπαραμέτρους:
    - Αριθμό φίλτρων ***K***,
    - Το χωρικό τους μέγεθος ***F***,
    - Το βήμα ***S***,
    - Τον αριθμό του γεμίσματος με μηδενικά ***P***.
- Παράγει έναν όγκο μεγέθους $W_2 \times H_2 \times D_2$, όπου:
    - $W_2 = (W_1 - F + 2P)S + 1$
    - $H_2 = (H_1 - F + 2P)S + 1$
    - $D_2 = K$

2.3.3 Συγκεντρωτικό Επίπεδο (Pooling Layer)

Το Συγκεντρωτικό Επίπεδο ***(Pooling Layer)*** είναι ένα επίπεδο το οποίο, συνήθως, εισάγεται μεταξύ διαδοχικών συνελικτικών επιπέδων σε μια αρχιτεκτονική ενός Συνελικτικού Δικτύου.

Η λειτουργία του έγκειται στην προοδευτική μείωση του χωρικού μεγέθους της αναπαράστασης, στην μείωση των παραμέτρων και υπολογισμών στο δίκτυο και, συνεπώς, στον έλεγχο της υπερπροσαρμογής ***(overfitting)***. Παρά τις όποιες χωρικές μειώσεις, το επίπεδο αυτό είναι σε θέση να διατηρεί τις πιο σημαντικές πληροφορίες της εισόδου.

Στο επίπεδο αυτό, ορίζουμε μια χωρική "γειτονιά" (για παράδειγμα, ένα παράθυρο 2x2) και επιλέγουμε να διατηρήσουμε μόνο το μεγαλύτερο στοιχείο από το διαμορφωμένο πίνακα μέσα στο παράθυρο. Αντί να επιλέξουμε το μεγαλύτερο στοιχείο, θα μπορούσαμε επίσης να επιλέξουμε την μέση τιμή των στοιχείων ***(Average Pooling)*** ή το άθροισμα όλων των στοιχείων μέσα στο παράθυρο. Στην πράξη έχει αποδειχθεί ότι αποτελεσματικότερα αποδίδει το ***Max Pooling***.



Σχήμα 2.7 Παράδειγμα εφαρμογής ενός συγκεντρωτικού επιπέδου χρησιμοποιώντας την λειτουργία MAX , χρησιμοποιώντας παράθυρο μεγέθους 2x2

Γενικότερα, για το συγκεντρωτικό επίπεδο μπορούμε να πούμε τα εξής:

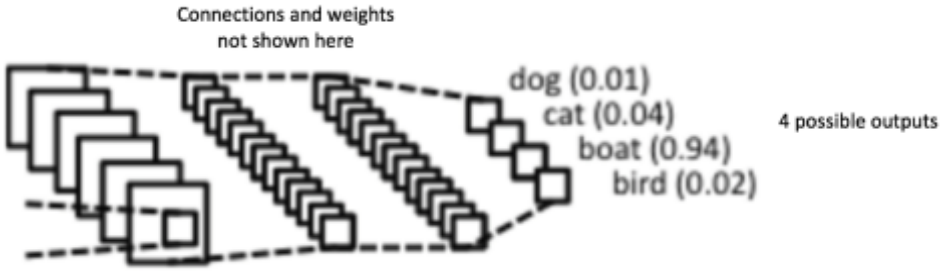
- Δέχεται ως είσοδο έναν όγκο μεγέθους $W_1 \times H_1 \times D_1$

- Απαιτεί δύο υπερπαραμέτρους:
    - Το χωρικό τους μέγεθος ***F***,
    - το βήμα ***S***
- Παράγει έναν όγκο μεγέθους $W_2 \times H_2 \times D_2$, όπου :
    - $W_2 = (W_1 - F)/S + 1$
    - $H_2 = (H_1 - F)/S + 1$
    - $D_2 = D_1$
- Εισάγει μηδενικές παραμέτρους αφού υπολογίζει μόνο μία σταθερή συνάρτηση της εισόδου

2.3.4 Πλήρως Συνδεδεμένο Επίπεδο (Fully-Connected Layer)

Όπως μπορούμε να καταλάβουμε από τον όρο Πλήρως Συνδεδεμένο Επίπεδο ***(Fully-Connected Layer),*** πρόκειται για ένα επίπεδο όπου κάθε νευρώνας σε ένα προηγούμενο επίπεδο συνδέεται με όλους τους νευρώνες του επόμενου επιπέδου. Πρόκειται για μία παραδοσιακή αρχιτεκτονική πολλών επιπέδων με νευρώνες, η οποία χρησιμοποιεί μια συνάρτηση ενεργοποίησης(συνήθως την softmax) στην έξοδό της.

Οι έξοδοι των συνελικτικών και συγκεντρωτικών επιπέδων αναπαριστούν χαρακτηριστικά υψηλών στρωμάτων. Ο σκοπός του πλήρως συνδεδεμένου επιπέδου είναι να χρησιμοποιήσει αυτά τα χαρακτηριστικά προκειμένου να κατηγοριοποιήσει την εικόνα εισόδου σε διάφορες κλάσεις, βασιζόμενο στο σύνολο δεδομένων που χρησιμοποιήθηκαν για εκπαίδευση. Για παράδειγμα, στο παράδειγμα που φαίνεται στο Σχήμα 2.8 παρατηρούμε ένα πλήρως συνδεδεμένο επίπεδο το οποίο κατηγοριοποιεί την εικόνα εισόδου σε τέσσερις κλάσεις με τις αντίστοιχες πιθανότητες.



Σχήμα 2.8 Παράδειγμα ενός πλήρως συνδεδεμένου δικτύου

Εκτός από την κατηγοριοποίηση που μπορεί να προσφέρει το επίπεδο αυτό, είναι επίσης ένας "φθηνός" τρόπος να εκπαιδεύσουμε το δίκτυό μας ώστε να μάθει μη-γραμμικούς συνδυασμούς των χαρακτηριστικών που παράγουν τα συνελικτικά και συγκεντρωτικά επίπεδα.

Το άθροισμα των πιθανοτήτων των κλάσεων που παράγει το πλήρως συνδεδεμένο επίπεδο είναι ένα(1) , πράγμα που μας το επιβεβαιώνει η συνάρτηση ενεργοποίησης softmax που χρησιμοποιούμε στην έξοδο του επιπέδου αυτού. Η συνάρτηση αυτή, δέχεται σαν είσοδο ένα διάνυσμα από τυχαίες πραγματικές τιμές και τις αντιστοιχίζει σε ένα διάνυσμα με τιμές από μηδέν έως ένα και με συνολικό άθροισμα των τιμών αυτών ίσο με ένα(1).

2.4 Κανονικοποίηση

2.4.1 Το πρόβλημα της Υπερπροσαρμογής

Η *γενίκευση (generalization)* στην Μηχανική Μάθηση αναφέρεται στο πόσο καλά μπορούν να αποδίδουν τα δίκτυα σε παραδείγματα τα οποία τα οποία δεν περιέχονταν στο εκπαιδευτικό σύνολο δεδομένων. Ο στόχος των περισσοτέρων μοντέλων Μηχανικής Μάθησης είναι να η "καλή" γενίκευση από το εκπαιδευτικό σύνολο δεδομένων, προκειμένου να κάνουν σωστές προβλέψεις στο μέλλον για δεδομένα που δεν είχαν ξαναδεί προηγουμένως.

Το πρόβλημα της *υπερπροσαρμογής* (*overfitting*), που αναφέρθηκε και προηγουμένως, εμφανίζεται όταν τα μοντέλα εκπαιδεύονται πολύ καλά στις λεπτομέρειες και στον θόρυβο του εκπαιδευτικού συνόλου δεδομένων, αλλά δεν μπορούν να γενικεύοσυν καλά και έτσι η απόδοσή τους είναι χαμηλή για δοκιμαστικά δεδομένα. Το συγκεκριμένο πρόβλημα εμφανίζεται πολύ συχνά όταν το εκπαιδευτικό σύνολο δεδομένων είναι αρκετά μικρό σε σχέση με τις παραμέτρους στις οποίες πρέπει να εκπαιδευθεί το μοντέλο και μπορεί να είναι και εκατομμύρια στον αριθμό τους.

Σχήμα 2.9 Σύμπτωμα υπερπροσαρμογής όπου το δοκιμαστικό λάθος είναι πολύ μεγαύτερο από το εκπαιδευτικό λάθος

2.4.2 Επίπεδα Κανονικοποίησης

Η *κανονικοποίηση (regularization)* αποτελεί μία λύση-κλειδί στο πρόβλημα της υπερπροσαρμογής. Επίσης, μερικές τεχνικές κανονικοποίησης μπορούν να χρησιμοποιηθούν προκειμένου να μειώσουν την χωρητικότητα του μοντέλου, διατηρώντας, παράλληλα, την ακρίβεια. Οι μέθοδοι που χρησιμοποιούνται προκειμένου να αντιμετωπιστεί το συγκεκριμένο πρόβλημα αναλύονται στην συνέχεια.

2.4.2.1 Αύξηση Εκπαιδευτικού Συνόλου Δεδομένων

Ένα υπερπροσαρμοσμένο μοντέλο *(overfitting model)* μπορεί να αποδώσει, σίγουρα, καλύτερα εάν ο αλγόριθμος εκπαίδευσης επεξεργάζεται περισσότερα δεδομένα εκπαίδευσης. Μπορεί ένα ήδη υπάρχον σύνολο δεδομένο να είναι περιορισμένο, υπάρχει, ωστόσο, η δυνατότητα για μερικά προβλήματα μηχαικής μάθησης, να παραχθούν νέα, συνθετικά δεδομένα. Παρά το γεγονός ότι δεν υπάρχει κάποια καθορισμένη "συνταγή" για τον τρόπο με τον οποίον θα πρέπει να παράγονται συνθετικά δεδομένα, η βασική αρχή είναι να επεκτείνουμε το ήδη υπάρχον σύνολο δεδομένων, εφαρμόζοντας λειτουργίες που αντανακλούν τις πραγματικές ποικιλίες όσο το δυνατόν καλύτερα.

2.4.2.2 Πρόωρο Σταμάτημα Εκπαίδευσης

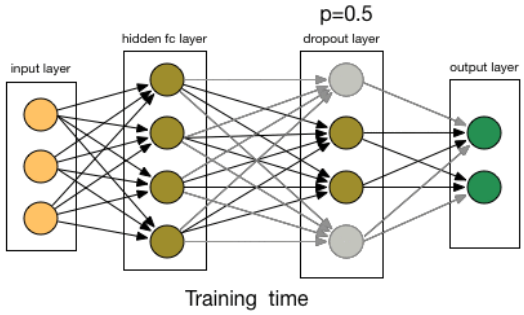
Μια άλλη τακτική που αντιμετωπίζει το πρόβλημα της υπερπροσαρμογής είναι αυτή του πρόωρου σταματήματος της εκπαιδευσης (Early Stopping). Με αυτόν τον τρόπο, διακόπτουμε την διαδικασία της εκπαίδευσης μόλις παρατηρήσουμε πτώση της απόδοσης του μοντέλου στο σύνολο δεδομένων επαλήθευσης.

Διαισθητικά, καθώς το μοντέλο βλέπει περισσότερα δεδομένα και μαθαίνει μοτίβα και συσχετίσεις, και το εκπαιδευτικό και το δοκιμαστικό λάθος *(training and testing error)* μειώνονται. Ωστόσο, μετά από πολλά περάσματα του εκπαιδευτικού συνόλου δεδομένων, το μοντέλο μπορεί να αρχίζει να υπερπροσαρμόζεται και να μαθαίνει και τον θόρυβο που υπάρχει στο εκπαιδευτικό σύνολο δεδομένων. Σε αυτήν την περίπτωση, ενώ το εκπαιδευτικό λάθος θα συνεχίζει να μειώνεται, το δοκιμαστικό λάθος θα αυξάνεται. Το πρόωρο σταμάτημα έγκειται στην εύρεση αυτής της σωστής στιγμής με το ελάχιστο δοκιμαστικό λάθος, όπως φαίνεται και στο σχήμα 2.10

Σχήμα 2.10 Εύρεση σημείου με ελάχιστο δοκιμαστικό λάθος

2.4.2.3 Περιορισμός Ενεργοποίησης (Dropout)

Μια πιο πρόσφατη προσέγγιση στην αντιμετώπιση του προβλήματος της υπερπροσαρμογής περιλαμβάνει τον περιορισμό ενεργοποίησης. Στο επίπεδο αυτό, σε κάθε επανάληψη της εκπαίδευσης, απενεργοποιούνται τυχαία κάποιοι νευρώνες του δικτύου μαζί με όλες τις εισερχόμενες και εξερχόμενες συνδέσεις. Στο Σχήμα 2.11 φαίνεται το επίπεδο περιορισμού ενεργοποίησης, όπου ο κάθε νευρώνας έχει πιθανότητα 50% να απενεργοποιηθεί.



Σχήμα 2.11 Εφαρμογή του περιορισμού ενεργοποίησης με πιθανότητα p=0.5

2.4.2.4 Κανονικοποίησεις L1 & L2

Οι κανονικοποιήσεις L1 & L2 (Regularization L1 & L2) βασίζονται στην υπόθεση ότι ένα μοντέλο με μικρά βάρη είναι κάπως απλούστερο από ένα δίκτυο με μεγάλα βάρη. Οι προαναφερθείσες κανονικοποιήσεις προσπαθούν να κρατήσουν τα βάρη σε μία μικρή τιμή ή σε μία που τείνει στο μηδέν εκτός και αν υπάρχουν μεγάλες κλίσεις (*gradients*) που αντικορούουν.

Κανονικοποίηση L2

- Για κάθε βάρος *w,* προστίθεται ένας όρος $\frac{1}{2}\lambda w^2$ στην συνάρτηση κόστους, προκειμένου να αποτρέψει το δίκτυο από το να μοντελοποιήσει πλήρως τα δεδομένα εκπαίδευσης. Τότε, η συνάρτηση κόστους γίνεται: $L(x,y)_{new} = L(x,y) + 0.5\lambda w^2$
- Τείνει να οδηγεί όλα τα βάρη σε μικρότερες τιμές

Κανονικοποίηση L1

- Για κάθε βάρος *w* προστίθεται ένας όρος $\lambda\,|w|$ στην συνάρτηση κόστους.
- Τείνει να οδηγεί κάποια βάρη κατ ευθείαν στο μηδέν, ενώ αφήνει κάποια άλλα να είναι μεγάλα.

2.5 Συνάρτηση Κόστους

Μία συνάρτηση κόστους σε ένα πρόβλημα επιβλεπόμενης μάθησης (*supervised learning*), υπολογίζει την συμβατότητα μεταξύ της πρόβλεψης του μοντέλου (π.χ. οι αντίστοιχες πιθανότητες σε ένα πρόβλημα κατηγοριοποίησης όπως αυτό που είδαμε προηγουμένως) και της πραγματικής της τιμής *(ground truth label)*. Η συνολική απώλεια δεδομένων παίρνει την μορφή ενός μέσου όρου απωλειών δεδομένων από κάθε ξεχωριστό παράδειγμα, πράγμα που σημαίνει ότι η συνάρτηση παίρνει την μορφή $L = \frac{1}{N}\sum_i L_i$, όπου ***N*** είναι ο αριθμός των δεδομένων εκπαίδευσης. Εάν θεωρήσουμε ότι $f = f(x_i; W)$ είναι οι ενεργοποιήσεις του

επιπέδου εξόδου σε ένα νευρωνικό δίκτυο, τότε, για το πρόβλημα της κατηγοριοποίησης που είδαμε προηγουμένως, μπορούμε να ορίσουμε ως συνάρτηση κόστους την εξής:

$$L_i = \sum_{j \neq y_i} max(0, f_j - f_{yi} + 1)$$

,

γνωστή και ως *SVM,* η οποία αποτελεί και την συνηθέστερη συνάρτηση κόστους.

Η δεύτερη πιο ευρέως χρησιμοποιούμενη επιλογή είναι ο Softmax κατηγοριοποιητής, ο οποίος χρησιμοποιεί την απώλεια διασχιζόμενης εντροπίας(cross-entropy loss) και έχει την μορφή:

$$L_i = -log(\frac{e^{f_{yi}}}{\sum_j e^{f_j}})$$

2.6 Βελτιστοποίηση

Για ένα πρόβλημα επιβλεπόμενης μάθησης, μπορούμε να μειώσουμε τον φόρτο εργασίας με το να λύσουμε ένα πρόβλημα βελτιστοποίησης της μορφής $\theta^* = \arg min_\theta g(\theta)$, όπου θ είναι ένα διάνυσμα παραμέτρων και g είναι μια συνάρτηση η οποία, συνήθως, συνδυάζει τη μέση απώλεια όλων των παραδειγμάτων και μία ποινή κανονικοποίησης.

Μια πιθανή προσέγγιση στο πρόβλημα αυτό θα ήταν να επιλέγουμε τυχαία θ έως ότου βρούμε το θ εκείνο για το οποίο ελαχιστοποιείται η $g(\theta)$. Ωστόσο, ένα τυπικό νευρωνικό δίκτυο που θέλουμε να εκπαιδεύσουμε μπορεί να έχει διανύσματα παραμέτρων με εκατομμύρια ή δισεκατομμύρια παραμέτρους και έτσι μέθοδοι σαν και αυτήν γίνονται υπολογιστικά αδύνατες.

Προκειμένου να βελτιώσουμε την αποτελεσματικότητα της βελτιστοποίησης, μπορούμε να κάνουμε κάποιες επιπρόσθετες υποθέσεις για την συνάρτηση *g*. Πιο συγκεκριμένα, αν περιοριστούμε στο να χρησιμοποιούμε μόνο διαφορίσιμες συναρτήσεις, τότε μπορούμε να υπολογίσουμε την κλίση *(gradient)* $\nabla_\theta g$ με οπισθοδιάδοση (θα αναλύσουμε αυτήν την μέθοδο στη συνέχεια). Η κλίση είναι είναι ένα διάνυσμα από μερικές παραγώγους, η οποία μας δίνει το πόσο απότομα μετακινείται η *g* κατά μήκος κάθε διάστασης του *θ*. Μπορούμε να χρησιμοποιήσουμε την κλίση αυτήν ώστε να μας δώσει μια κατεύθυνση για να αναζητήσουμε το βέλτιστο *θ*. Πιο συγκεκριμένα, μπορούμε να βελτιώσουμε την αναζήτηση του *θ* με το να

προσθέτουμε σε αυτό μια μικρή ποσότητα της αρνητικής κατεύθυνσης της κλίσης. Οι παρατηρήσεις αυτές μας οδηγούν στον αλγόριθμο της κατηφορικής κλίσης (*gradient descent*), ο οποίος εναλλάσσει τα δύο παρακάτω βήματα:

1. Υπολογισμός της κλίσης με οπισθοδιάδοση
2. Ενημέρωση παραμέτρων με την προσθήκη μιας μικρής ποσότητας της αρνητικής κατεύθυνσης της κλίσης

Σαν μια τελευταία σκέψη, επειδή τα σύνολα δεδομένων που χρησιμοποιούμε μπορεί να είναι πολύ μεγάλα στην πράξη, μπορούμε να υπολογίζουμε την κατηφορική κλίση για μια ομάδα *(batch)* παραδειγμάτων την φορά. Με τον τρόπο αυτόν, υλοποιούμε λιγότερες προσεγγιστικές ενημερώσεις αντί για περισσότερες και ακριβέστερες - μια στρατηγική που αποδίδει καλύτερα σε πρακτικές εφαρμογές. Ο αλγόριθμος που προκύπτει ονομάζεται Στοχαστική Κατηφορική Κλίση (***Stohastic Gradient Descent or SGD***) και αναλύεται ως εξής:

Αλγόριθμος1 Στοχαστική Κατηφορική Κλίση

1. Δεδομένου ενός αρχικού σημείου $\theta \epsilon dom g$
2. Δεδομένου ενός βήματος μεγέθους ϵ
3. Επανάλαβε
   a. Δειγματολήπτισε μια ομάδα από *m* παραδείγματα $\{(x_1, y_1), ..., (x_m, y_m)\}$ από τα δεδομένα εκπαίδευσης
   b. Υπολόγισε την κλίση $\bigtriangledown_\theta g(\theta) \approx \bigtriangledown_\theta \left[\frac{1}{m} \sum_{i=1}^{m} L(f_\theta(x_i), y_i) + R(f_\theta) \right]$ με την μέθοδο της οπισθοδιάδοσης
   c. Υπολόγισε την κατεύθυνση ενημέρωσης: $\bigtriangleup_\theta := -\epsilon \bigtriangledown_\theta g(\theta)$
   d. Εκτέλεση μία ενημέρωση παραμέτρων $\theta := \theta + \bigtriangleup\theta$

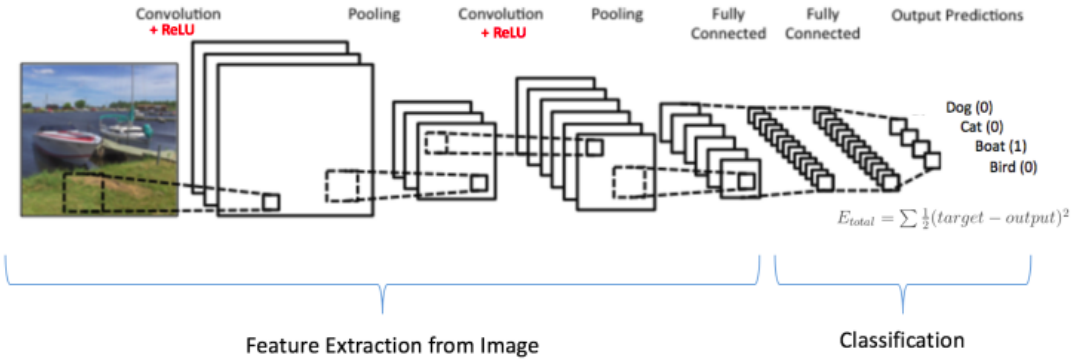
   έως ότου υπάρξει σύγκλιση

2.7 Εκπαίδευση

Στο κομμάτι αυτό θα παρουσιάσουμε μια διαισθητική ανάλυση του τρόπου με τον οποίον εκπαιδεύονται τα συνελικτικά νευρωνικά δίκτυα, εισάγοντας και την μέθοδο της οπισθοδιάδοσης ***(backpropagation)***.

Ως οπισθοδιάδοση ορίζεται η διαδικασία κατά την οποία μπορούμε, αποδοτικά, να υπολογίσουμε τις κλίσεις κλιμακωτών συναρτήσεων με βάση τις εισόδους τους. Ο αλγόριθμος της οπισθοδιάδοσης είναι μια αναδρομική εφαρμογή του αλυσιδωτού κανόνα, όπως τον γνωρίζουμε από τη θεωρία του Λογισμού. Από το προηγούμενο κεφάλαιο θυμόμαστε ότι η συνάρτηση για την οποία ενδιαφερόμαστε να υπολογίσουμε τις κλίσεις είναι η *g*, που δέχεται σαν είσοδο το σύνολο δεδομένων παραδειγμάτων (x_i, y_i) και τις παραμέτρους *θ*. Συγκεκριμένα, ενδιαφερόμαστε να υπολογίσουμε την κλίση $\nabla_\theta g$ αναφορικά με τις παραμέτρους *θ* προκειμένου να υλοποιήσουμε την ενημέρωση παραμέτρων. Γενικά, θα θέλαμε να πολλαπλασιάσουμε τις τοπικές παραγώγους όλων των ενδιάμεσων συναρτήσεων μεταξύ τους για να αποκτήσουμε την τελική παράγωγο της εξόδου σε σχέση με την είσοδο. Αν υποθέσουμε ότι είχαμε για είσοδο ένα διάνυσμα x_0 το οποίο μετατρέπουμε μέσα από μία σειρά συναρτήσεων $x_i = f_i(x_{i-1})$, όπου $i = 1, ..., k$. Υποθέτουμε ότι η κλίση υπάρχει και έτσι μπορούμε να υπολογίσουμε τον Ιακωβιανό πίνακα $\frac{\partial x_i}{\partial x}_{i-1}$ όλων των ενδιάμεσων μετασχηματισμών, που μας υποδεικνύει πως κάθε διάσταση εξόδου του x_i εξαρτάται από κάθε διάσταση εισόδου x_{i-1}. Με τον αλυσιδωτό κανόνα, η τελική κλιση, για την οποία κιόλας ενδιαφερόμαστε, είναι απλώς το γινόμενο πινάκων όλων των Ιακωβιανών πινάκων : $\frac{\partial x_k}{\partial x_0} = \prod_{i=1}^{k} \frac{\partial x_i}{\partial x}_{i-1}$.

Έχοντας, πλέον, κατά νου, πως λειτουργεί η μέθοδος της οπισθοδιάδοσης, μπορούμε να περιγράψουμε την γενική διαδικασία εκπαίδευσης ενός συνελικτικού δικτύου βασιζόμενοι στον παρακάτω αλγόριθμο:



Σχήμα 2.12 Εκπαίδευση ενός συνελικτικού νευρωνικού δικτύου από την αρχή μέχρι το τέλος

Αλγόριθμος 2 Εκπαίδευση Συνελικτικού Νευρωνικού Δικτύου

1. Αρχικοποιούμε όλα τα φίλτρα και τις παραμέτρους/βάρη με τυχαίες τιμές
2. Το δίκτυο δέχεται μια είσοδο, ας θεωρήσουμε μία εικόνα, η οποία περνάει περνάει προς τα εμπρός από όλα τα υπάρχοντα επίπεδα (συνελικτικά, συγκεντρωτικά κ.λ.π.) και υπολογίζει τις πιθανότητες εξόδου για κάθε κλάση.
    a. Ας υποθέσουμε το παράδειγμα του Σχήματος 2.12, με πιθανότητες εξόδου [0.2, 0.4, 0.1, 0.3]
    b. Εφ' όσον τα βάρη είναι αρχικοποιημένα με τυχαίες τιμές, οι πιθανότητες εξόδου θα έχουν και αυτές τυχαίες τιμές.
3. Υπολογίζουμε το συνολικό σφάλμα στο επίπεδο εξόδου(άθροισμα και των τεσσάρων κλάσεων)
    a. Συνολικό Σφάλμα = $\sum$ ½ (πιθανότητα στόχου - πιθανότητα εξόδου)^2
4. Στο βήμα αυτό θα χρησιμοποιήσουμε τη μέθοδο της οπισθοδιάδοσης(backpropagation) που αναφέραμε προηγουμένως, προκειμένου να υπολογίσουμε τις κλίσεις (gradients) του σφάλματος αναφορικά με όλα τα βάρη του δικτύου και στη συνέχεια θα εφαρμόσουμε την κατηφορική κλίση (gradient descent) για να ενημερώσουμε όλες τις τιμές των φίλτρων/βαρών και τις τιμές των παραμέτρων προκειμένου να ελαχιστοποιήσουμε το σφάλμα εξόδου ως εξής:
    a. Τα βάρη προσαρμόζονται ανάλογα με την συνεισφορά τους στο συνολικό σφάλμα

b. Όταν η ίδια εικόνα εισαχθεί σαν είσοδος ξανά, οι πιθανότητες εξόδου πιθανότατα να είναι [0.1, 0.1, 0.7, 0.1] οι οποίες βρίσκονται πιο κοντά στο διάνυσμα-στόχο [0, 0, 1, 0]
c. Αυτό σημαίνει ότι το δίκτυο έχει μάθει να κατηγοριοποιεί την συγκεκριμένη εικόνα σωστά με το να προσαρμόζει τα βάρη/φίλτρα ώστε να μειωθεί το σφάλμα εξόδου
d. Παράμετροι όπως ο αριθμός των φίλτρων, τα μεγέθη των φίλτρων, αρχιτεκτονική του δικτύου κ.λ.π., είναι προκαθορισμένα πριν από το βήμα 1 και δεν μεταβάλλονται κατά την διάρκεια της εκπαίδευσης - μόνο οι τιμές των φίλτρων και τα βάρη των συνδέσεων ενημερώνονται.

5. Επαναλαμβάνουμε τα βήματα 2-4 με όλες τις διαθέσιμες εικόνες του εκπαιδευτικού συνόλου δεδομένων

Τα βήματα που περιγράψαμε παραπάνω ***εκπαιδεύουν*** ένα συνελικτικό δίκτυο. Πρακτικά, αυτό σημαίνει ότι όλα τα βάρη και οι παράμετροι του δικτύου έχουν βελτιστοποιηθεί ώστε να μπορούν να κατηγοριοποιούν σωστά εικόνες από το εκπαιδευτικό σύνολο δεδομένων.

Όταν μία νέα εικόνα εισαχθεί σαν είσοδος στο δίκτυο, αυτό θα εκτελέσει μια εμπρόσθια εξάπλωση από όλα τα επιμέρους επίπεδα και θα παράξει μία πιθανότητα για κάθε κλάση που υπάρχει. Αν το εκπαιδευτικό σύνολο δεδομένων που έχουμε είναι αρκετά μεγάλο, το δίκτυο θα γενικεύσει αρκετά καλά και σε νέες εικόνες και θα τις κατηγοριοποιήσει σε σωστές κατηγορίες.

2.8 Γνωστά Συνελικτικά Νευρωνικά Δίκτυα

Στο κομμάτι αυτό θα παρουσιάσουμε μερικά, ήδη υπάρχοντα, γνωστά συνελικτικά δίκτυα που χρησιμοποιούνται ευρέως στον χώρο της Βαθιάς Μηχανικής Μάθησης, και τα οποία επιλέγονται βάσει τον τύπο του εκάστοτε προβλήματος που θέλουμε να επιλύσουμε.

- ***LeNet*** : Αποτελεί την πρώτη επιτυχημένη εφαρμογή Συνελικτικών Δικτύων που αναπτύχθηκε από τον Yann LeCun την δεκαετία του 1990. Η συγκεκριμένη αρχιτεκτονική χρησιμοποιήθηκε κυρίως για αναγνώριση κωδικών, ψηφίων κ.λ.π.

- ***AlexNet*** : Το συγκεκριμένο δίκτυο ήταν το πρώτο το οποίο έκανε τα Συνελικτικά Δίκτυα διάσημα στο χώρο της Όρασης Υπολογιστών. Το δίκτυο αυτό είχε μία πολύ παρόμοια αρχιτεκτονική με αυτή του LeNet, ωστόσο, ήταν βαθύτερο, μεγαλύτερο και είχε πολλά συνελικτικά επίπεδα, στοιβαγμένα το ένα πάνω στο άλλο, που είχε αποτελέσει μια πρωτοποριακή τεχνική.
- ***ZF Net*** : Το δίκτυο αυτό αποτελεί μια βελτίωση του AlexNet, διορθώνοντας κάποιες υπερπαραμέτρους της αρχιτεκτονικής. Πιο συγκεκριμένα, επεκτείνανε το μέγεθος του μεσαίου συνελικτικού επιπέδου, ενώ παράλληλα έκαναν το βήμα και το μέγεθος φίλτρου του πρώτου επιπέδου μικρότερα. Το συγκεκριμένο δίκτυο απέσπασε την πρώτη θέση στον διαγωνισμό ILSVRC 2013.
- ***GoogleNet*** : Νικητής του διαγωνισμού ILSVRC 2014 αναδείχθηκε το συγκεκριμένο δίκτυο το οποίο αναπτύχθηκε από την Google. Το βασικό πλεονέκτημά του έναντι στα προηγούμενα μοντέλα, έγκειται στην δραματική μείωση των παραμέτρων του δικτύου. Πιο συγκεκριμένα, ο αριθμός των παραμέτρων μειώθηκε στα 4 εκατομμύρια σε σύγκριση με το AlexNet που είχε 60 εκατομμύρια παραμέτρους.
- ***VGGNet*** : Το συγκεκριμένο δίκτυο, αν και απέσπασε την δεύτερη θέση στον διαγωνισμό ILSVRC 2014, ήταν αντίστοιχα αποτελεσματικό με το GoogleNet. Η βασική συνεισφορά του ήταν στο γεγονός ότι το βάθος ενός δικτύου είναι ένα σημαντικό συστατικό για την καλή απόδοση.
- ***ResNet*** : Το δίκτυο αυτό ήταν το νικητήριο δίκτυο για τον διαγωνισμό ILSVRC 2015. Εισήγαγε για πρώτη φορά παραλειπούμενες συνδέσεις (skip connections) και εκτεταμένη χρήση κανονικοποίησης. Η αρχιτεκτονική του δικτύου αυτού, επίσης, δεν περιέχει πλήρως συνδεδεμένο επίπεδο στο τέλος του δικτύου.

3

Ανατροφοδοτούμενα Νευρωνικά Δίκτυα (RNN)

3.1 Ορισμός & Λειτουργία

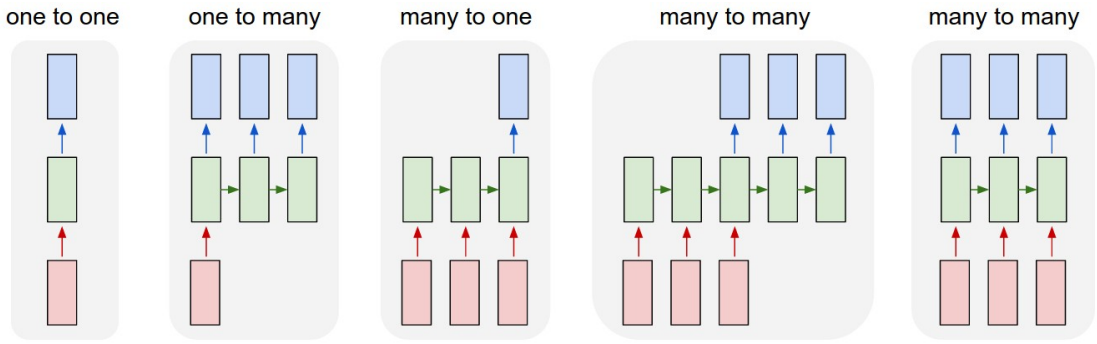
Σε πολλές πρακτικές εφαρμογές της μηχανικής μάθησης, ο χώρος εισόδων ή εξόδων περιέχει ακολουθίες. Για παράδειγμα, οι προτάσεις συχνά μοντελοποιούνται σαν ακολουθίες λέξεων, όπου κάθε λέξη μπορεί να αναπαρασταθεί σαν ένα *διάνυσμα μοναδικού άσσου*(*one hot-vector*), που σημαίνει ότι το διάνυσμα αυτό έχει παντού μηδενικά εκτός από μία θέση που έχει άσσο και αντιστοιχεί στην λέξη ενός λεξικού που θέλουμε να αναπαραστήσουμε. Ένα *ανατροφοδοτούμενο νευρωνικό δίκτυο* (*Recurrent Neural Network*) είναι μία διάταξη συνδέσεων που επεξεργάζονται μια ακολουθία από διανύσματα $\{x_1, ..., x_T\}$ χρησιμοποιώντας μια ανατροφοδοτούμενη φόρμουλα της μορφής $h_t = f_\theta(h_{t-1}, x_t)$, όπου f είναι μία συνάρτηση την οποία θα περιγράψουμε αναλυτικότερα στην συνέχεια, ενώ οι ίδιες παράμετροι θ χρησιμοποιούνται σε κάθε βήμα, πράγμα που μας δίνει το δικαίωμα να επεξεργαζόμαστε ακολουθίες με έναν τυχαίο αριθμό διανυσμάτων. Το κρυφό διάνυσμα h_t μπορεί να ερμηνευθεί σαν μια τρέχουσα σύνοψη όλων των διανυσμάτων *x* μέχρι εκείνο το βήμα και η ανατροφοδοτούμενη φόρμουλα ενημερώνει την σύνοψη αυτήν βασιζόμενη στο επόμενο διάνυσμα. Είναι σύνηθες είτε να χρησιμοποιούμε $h_0 = \underset{0}{\rightarrow}$, ή να συμπεριφερόμαστε στο h_0 σαν παράμετρο και να μαθαίνει την αρχική κρυφή κατάσταση. Η ακριβής μαθηματική μορφή της επανάληψης $(h_{t-1}, x_t) \rightarrow h_t$ διαφέρει από μοντέλο σε μοντέλο και θα τα αναλύσουμε στην συνέχεια.

Τα ανατροφοδοτούμενα νευρωνικά δίκτυα διαφέρουν από τα απλά προωθητικά δίκτυα, υπό την έννοια ότι προωθούν την έξοδό τους ξανά προς στην είσοδό τους. Μπορούμε, να

φανταστούμε, επομένως, ότι τα δίκτυα αυτά διαθέτουν μνήμη. Το να προσθέτουμε μνήμη, ωστόσο, σε ένα δίκτυο, έχει κάποιο σκοπό : Υπάρχουν πληροφορίες μέσα στις ακολουθίες εισόδου, και τις οποίες χρησιμοποιούν τα ανατροφοδοτούμενα νευρωνικά δίκτυα προκειμένου να εκτελούν εργασίες που τα απλά προωθητικά δίκτυα δεν μπορούν.

3.1.1 Απλούστερη Μορφή Ανατροφοδοτούμενων Νευρωνικών Δικτύων (Vanilla RNN)

Η απλούστερη μορφή ενός ανατροφοδοτούμενου νευρωνικού δικτύου χρησιμοποιεί μία επανάληψη της μορφής $tanh\left(W\begin{pmatrix}x_t\\h_{t-1}\end{pmatrix}\right)$, που σημαίνει ότι το προηγούμενο κρυφό διάνυσμα και η τρέχουσα είσοδος έχουν ενωθεί και έχουν μετασχηματισθεί γραμμικά από τις παραμέτρους *W*. Αξίζει να παρατηρήσουμε ότι αυτό είναι ισοδύναμο με το να γράφαμε $h_t = tanh\left(W_{xh}x_t + W_{hh}h_{t-1}\right)$, όπου οι δύο πίνακες W_{xh}, W_{hh} ενωμένοι με οριζόντιο τρόπο είναι ισοδύναμοι με τον παραπάνω πίνακα *W.* Οι εξισώσεις αυτές παραλείπουν την επιπρόσθετη κλίση (*bias*) για λόγους συντομίας. Η μη γραμμική συνάρτηση *tanh* μπορεί, επίσης, να αντικατασταθεί από μια ReLU συνάρτηση, όπως αυτήν που είδαμε στο δεύτερο κεφάλαιο. Αν τα διανύσματα εισόδου x_t έχουν διάσταση *D* και οι κρυφές καταστάσεις διάσταση *H*, τότε ο *W* είναι ένας πίνακας μεγέθους $[H \times (D + H)]$. Ερμηνεύοντας την εξίσωση αυτήν, οι νέες κρυφές καταστάσεις σε κάθε χρονικό βήμα αποτελούν μία γραμμική συνάρτηση των στοιχείων x_t, h_{t-1} και συνοστίζονται, στην συνέχεια, από μη-γραμμικότητα.



Σχήμα 3.1 Ένα σύνηθες νευρωνικό δίκτυο στα αριστερά μπορεί να δεχθεί σαν είσοδο ένα διάνυσμα (κόκκινο), να το μετασχηματίσει μέσω ενός κρυφού επιπέδου (πράσινο) και εν τέλει να παράγει ένα διάνυσμα εξόδου (μπλε). Στα διαγράμματα αυτά, τα κουτιά παριστάνουν διανύσματα ενώ τα βέλη υποδηλώνουν λειτουργικές εξαρτήσεις. Τα ανατροφοδοτούμενα δίκτυα μας επιτρέπουν να επεξεργαζόμαστε ακολουθίες διανυσμάτων, όπως για παράδειγμα: 1) στην έξοδο, 2)στην είσοδο, ή 3)και στις δύο πλευρές, είτε σειριακά ή παράλληλα, όπως φαίνεται στα υπόλοιπα σχήματα.

3.2 Εκπαίδευση

Για την εκπαίδευση των ανατροφοδοτούμενων νευρωνικών δικτύων χρησιμοποιούμε τεχνικές παρεμφερείς με αυτές που είδαμε στο προηγούμενο κεφάλαιο και οι οποίες χρησιμοποιήθηκαν για την εκπαίδευση των συνελικτικών νευρωνικών δικτύων.

3.2.1 Οπισθοδιάδοση στο Χρόνο (Backpropagation Through Time)

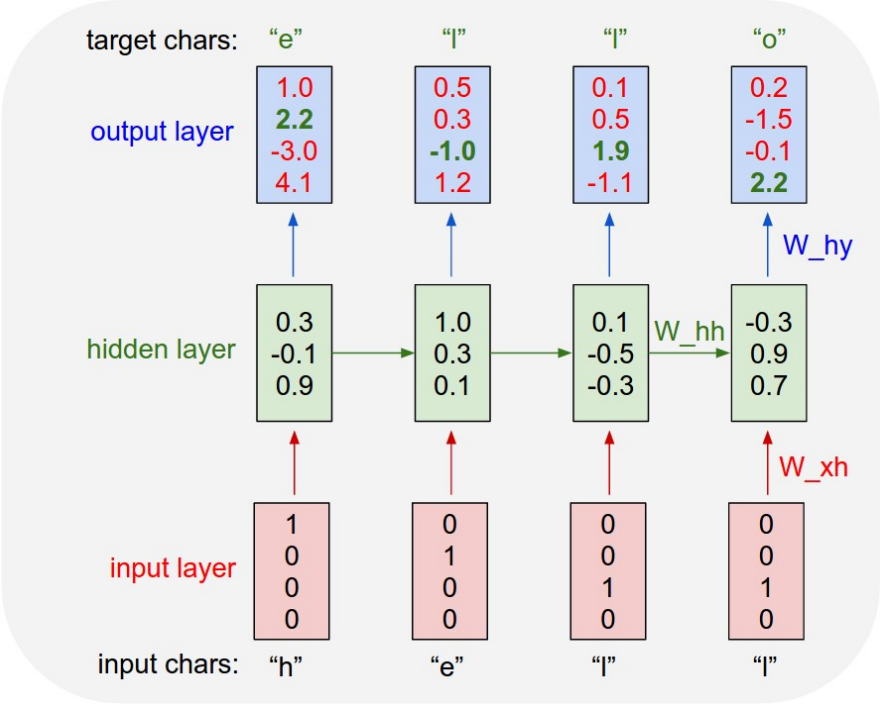
Όπως έχουμε επισημάνει, ο βασικός σκοπός των επαναλαμβανόμενων νευρωνικών δικτύων είναι να κατηγοριοποιούν επακριβώς ακολουθιακές εισόδους. Προκειμένου να επιτευχθεί αυτός ο στόχος, κάνουμε χρήση του σφάλματος της οπισθοδιάδοσης *(backpropagation error)* και της κατηφορικής κλίσης *(gradient descent)*.

Η μέθοδος της οπισθοδιάδοσης στα προωθητικά δίκτυα κινείται προς τα πίσω, από το τελικό σφάλμα μέσω των εξόδων, των βαρών και των εισόδων κάθε κρυφού επιπέδου, αποδίδοντας "ευθύνη" σε αυτά τα βάρη για ένα ποσοστό του λάθους, μέσω του υπολογισμού των μερικών παραγώγων $-\frac{\partial E}{\partial W}$, ή της σχέσης μεταξύ των λόγων της αλλαγής τους. Οι παράγωγοι αυτές, στην συνέχεια, χρησιμοποιούνται από τον εκπαιδευτικό μας κανόνα, την κατηφορική

κλίση, προκειμένου να προσαρμόσουμε τα βάρη προς τα πάνω ή προς τα κάτω, αναλόγως το προς τα που μειώνεται το σφάλμα.

Τα επαναλαμβανόμενα νευρωνικά δίκτυα βασίζονται σε μία επέκταση της οπισθοδιάδοσης που ονομάζεται *οπισθοδιάδοση στον χρόνο* (*backpropagation through time, or BTT*). Στην περίπτωση αυτήν, ο χρόνος εκφράζεται απλά σαν μια καλώς ορισμένη, διατεταγμένη σειρά υπολογισμών, συνδέοντας το ένα χρονικό βήμα με το επόμενο.

Τα νευρωνικά δίκτυα, είτε επαναλαμβανόμενα είτε όχι, είναι απλές, φωλιασμένες, σύνθετες συναρτήσεις της μορφής $f\left(g\left(h\left(x\right)\right)\right)$. Η προσθήκη του στοιχείου του χρόνου, απλώς επεκτείνει τη σειρά των λειτουργιών για τις οποίες υπολογίζουμε τις παραγώγους με τον αλυσιδωτό κανόνα.



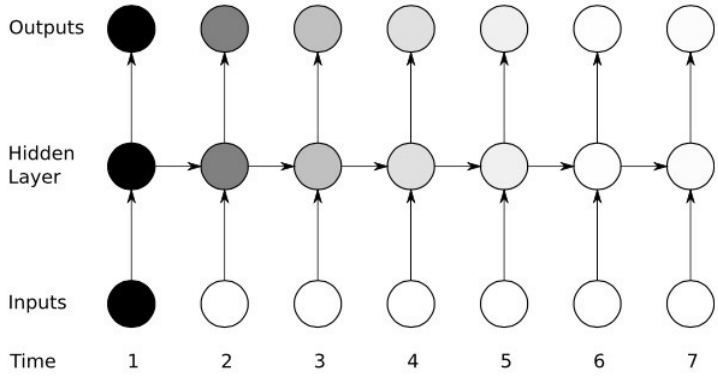
Σχήμα 3.2 Ένα παράδειγμα χρήσης ενός RNN σαν ένα επιπέδου χαρακτήρα, γλωσσικό μοντέλο. Η ακολουθία εκπαίδευσης είναι "hello" και το λεξικό μας έχει 4 χαρακτήρες h,e,l,o. Οι είσοδοι είναι one-hot διανύσματα των χαρακτήρων "h,e,l,l" και θέλουμε το RNN να προβλέψει τον επόμενο χαρακτήρα στην ακολουθία σε κάθε χρονικό βήμα. Το RNN έχει μια τρισδιάστατη κρυφή κατάσταση (πράσινο) και υπάρχουν 4 διαστάσεις στα διανύσματα εξόδου (μπλε), τα οποία ερμηνεύονται ως

αποτελέσματα για τον επόμενο χαρακτήρα. Η συνάρτηση κόστους και η κλίση θα ενθαρρύνουν τα αποτελέσματα των σωστών χαρακτήρων να αυξηθούν και τα υπόλοιπα αποτελέσματα να μειωθούν.

3.3 Το Πρόβλημα των Εξαφανιζόμενων/Ανατινασσόμενων Κλίσεων (Vanishing/Exploding Gradients)

Τα ανατροφοδοτούμενα νευρωνικά δίκτυα, όπως τα περισσότερα νευρωνικά δίκτυα, έχουν δημιουργηθεί εδώ και πολλά χρόνια. Το πρόβλημα των Εξαφανιζόμενων Κλίσεων παρουσιάσθηκε στις αρχές του 1990 και αποτέλεσε σημαντική τροχοπέδη στην απόδοση των επαναλαμβανόμενων νευρωνικών δικτύων.

Όπως μια ευθεία γραμμή εκφράζει μια αλλαγή στον άξονα *x* παράλληλα με μια αλλαγή στον άξονα *y*, έτσι και η κλίση εκφράζει μια αλλαγή στα βάρη αναφορικά με την αλλαγή στο σφάλμα. Αν δεν μπορούμε να γνωρίζουμε την κλίση, δεν μπορούμε να προσαρμόσουμε τα βάρη σε μία κατεύθυνση που θα μειώσει το σφάλμα και, ως εκ τούτου το δίκτυό μας σταματάει να μαθαίνει.

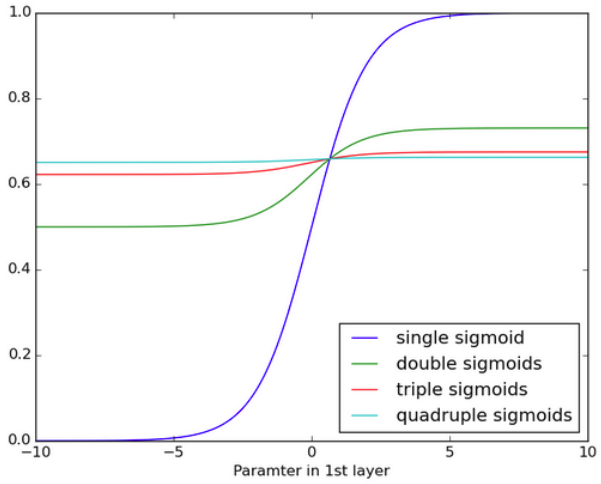


Σχήμα 3.3 Το πρόβλημα των εξαφανιζόμενων κλίσεων

Τα ανατροφοδοτούμενα νευρωνικά δίκτυα αποσκοπούν στο να καθιερώσουν συνδέσεις μεταξύ μιας τελικής εξόδου και γεγονότων, πολλά χρονικά βήματα πριν περιοριστούν, διότι είναι αρκετά δύσκολο να ξέρουν από πριν πόση σημασία να αποδώσουν σε μεμονωμένες εισόδους. Για αυτό, ευθύνεται, εν μέρει, το γεγονός ότι οι πληροφορίες που κυλούν μέσω των νευρωνικών δικτύων περνάνε από πολλά στάδια πολλαπλασιασμών.

Επειδή τα επίπεδα και τα χρονικά βήματα βαθιών νευρωνικών δικτύων συνδέονται μεταξύ τους μέσω πολλαπλασιασμών, οι παράγωγοι είναι επιρρεπείς στην εξαφάνιση ή στην "ανατίναξη".

Στο παρακάτω σχήμα μπορούμε να δούμε τα αποτελέσματα της εφαρμογής μιας σιγμοειδούς συνάρτησης πολλαπλές φορές, όπου η κλίση σταδιακά εξαφανίζεται.



Σχήμα 3.4 Αποτελέσματα εφαρμογής της σιγμοειδούς συνάρτησης πολλαπλές φορές

3.4 Διαρκείς Μονάδες με Μικρή Περίοδο Μνήμης (Long Short-Term Memory Units ή LSTMs)

Στα μέσα του 1990, μια παραλλαγή των ανατροφοδοτούμενων νευρωνικών δικτύων έκαναν την εμφάνισή τους, με το όνομα ***Διαρκείς Μονάδες με Μικρή Περίοδο Μνήμης (Long Short-Term Memory Units ή LSTMs*).** Τα δίκτυα αυτά αποτέλεσαν τη λύση στο πρόβλημα των Εξαφανιζόμενων/Ανατινασσόμενων Κλίσεων, όπως αυτό περιγράφηκε παραπάνω.

Τα LSTMs βοηθάνε στην διατήρηση του σφάλματος το οποίο μπορεί να οπισθοδιαδοθεί μέσω του χρόνου και των επιπέδων. Με το να διατηρούμε ένα πιο σταθερό σφάλμα, δίνουμε την δυνατότητα στα επανατροφοδοτούμενα δίκτυα να μαθαίνουν για πολλά βήματα χρόνου και ως εκ τούτου να ανοίγουν ένα δίαυλο προκειμένου να συνδέουν τις αιτίες και τα αποτελέσματα ξεχωριστά.

Τα LSTMs περιέχουν πληροφορίες που εκτείνονται πέρα από την κανονική ροή του επανατροφοδοτούμενου δικτύου σε ένα κελί με πύλες *(gated cell)*. Οι πληροφορίες μπορούν

να αποθηκευτούν, να γραφτούν ή να διαβαστούν από ένα κελί *(cell)*, με τρόπο παρόμοιο με αυτόν που γίνεται και με τα δεδομένα σε μια μνήμη ενός υπολογιστή. Το κελί παίρνει αποφάσεις για το τι θα κρατήσει, για το θα διαβαστεί, τι θα γραφτεί και τι θα διαγραφεί, μέσω πυλών που ανοίγουν και κλείνουν. Σε αντίθεση, όμως, με την ψηφιακή αποθήκευση στους υπολογιστές, οι πύλες αυτές είναι αναλογικές, υλοποιημένες με πολλαπλασιασμό ανα στοιχείο από σιγμοειδείς συναρτήσεις, που βρίσκονται στο εύρος 0-1. Η χρήση των αναλογικών πυλών παρουσιάζει το πλεονέκτημα έναντι των ψηφιακών στο ότι είναι διαφορίσιμες και ως εκ τούτου κατάλληλες για οπισθοδιάδοση.

Οι πύλες αυτές λειτουργούν με βάση τα σήματα που λαμβάνουν, και παρόμοια με τους νευρώνες ενός νευρωνικού δικτύου, εμποδίζουν ή αφήνουν να περάσουν πληροφορίες βάσει της σημασίας τους, τις οποίες φιλτράρουν με ένα δικό τους σύνολο βαρών. Τα βάρη αυτά, όπως και τα βάρη που προσαρμόζουν τις κρυφές καταστάσεις και τις καταστάσεις εισόδου, μεταβάλλονται μέσω της διαδικασίας της εκπαίδευσης των ανατροφοδοτούμενων νευρωνικών δικτύων. Πράγμα που σημαίνει, ότι τα κελιά μαθαίνουν πότε να αφήσουν μία πληροφορία να περάσει, πότε να την εμποδίσουν ή να την διαγράψουν μέσω της επαναληπτικής διαδικασίας των προβλέψεων, της οπισθοδιάδοσης σφάλματος και της ενημέρωσης βαρών μέσω της κατηφορικής κλίσης. Για να κατανοήσουμε καλύτερα πως λειτουργούν τα LSTMs , ας δούμε πώς υπολογίζεται η κρυφή κατάσταση s_t :

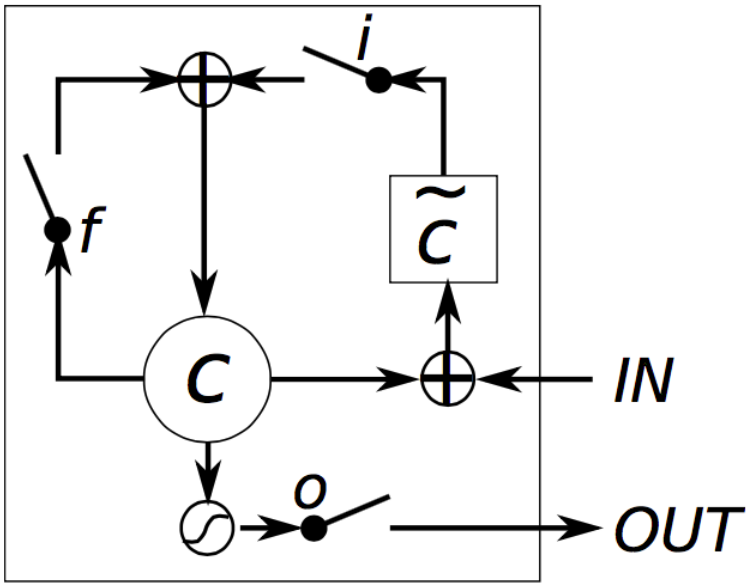
$$
\begin{aligned}
i &= \sigma(x_t U^i + s_{t-1} W^i) \\
f &= \sigma(x_t U^f + s_{t-1} W^f) \\
o &= \sigma(x_t U^o + s_{t-1} W^o) \\
g &= \ tanh(x_t U^g + s_{t-1} W^g) \\
c_t &= c_{t-1} \circ f + g \circ i \\
s_t &= \tanh(c_t) \circ o
\end{aligned}
$$

Ας εξηγήσουμε, όμως, τις παραπάνω σχέσεις:

- *i, f, o* ονομάζονται αντίστοιχα οι πύλες εισόδου *(input gates)*, οι πύλες διαγραφής *(forget gates),* και οι πύλες εξόδου (*output gates)*. Ονομάζονται πύλες, διότι η σιγμοειδής συνάρτηση συμπιέζει τις τιμές των διανυσμάτων αυτών μεταξύ 0 και 1 και με τον πολλαπλασιασμό ανά στοιχείο με ένα άλλο διάνυσμα, μπορούν να καθορίσουν πόση πληροφορία από το άλλο διάνυσμα επιθυμούν να κρατήσουν. Η πύλη εισόδου καθορίζει πόση πληροφορία από την νέα υπολογισμένη κατάσταση με βάση την τρέχουσα είσοδο επιθυμεί να κρατήσει. Η πύλη διαγραφής καθορίζει πόση

πληροφορία από την προηγούμενη κατάσταση θα κρατηθεί και τέλος η πύλη εξόδου αποφασίζει πόση πληροφορία από την εσωτερική κατάσταση θέλει να εκθέσει στο εξωτερικό δίκτυο. Όλες οι πύλες έχουν τις ίδιες διαστάσεις d_s , που είναι το μέγεθος της κρυφής κατάστασης.

- το g είναι μια "υποψήφια" κρυφή κατάσταση που υπολογίζεται βάσει την τρέχουσα είσοδο και την προηγούμενη κρυφή κατάσταση.
- *το* c_t είναι η εσωτερική μνήμη της μονάδας. Είναι ένας συνδυασμός της προηγούμενης μνήμης c_{t-1} πολλαπλασιασμένη με την πύλη διαγραφής και της προσφάτως υπολογισμένης κρυφής κατάστασης g, πολλαπλασιασμένης με την πύλη εισόδου.
- Δεδομένης της μνήμης c_t, τελικώς υπολογίζουμε την κρυφή κατάσταση εξόδου s_t με πολλαπλασιασμό της μνήμης με την πύλη εξόδου.



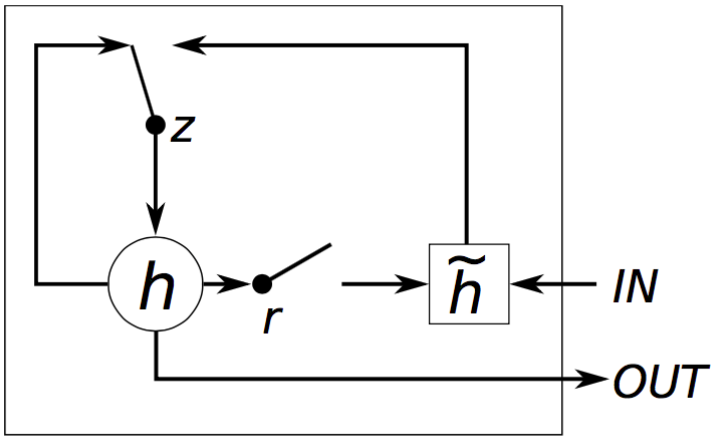
Σχήμα 3.5 Αρχιτεκτονική ενός δικτύου LSTM με τις πύλες που περιγράψαμε παραπάνω

3.5 Ανατροφοδοτούμενες Μονάδες με Πύλες (Gated Recurrent Units ή GRUs)

Μια άλλη μορφή επανατροφοδοτούμενων δικτύων τα οποία έδωσαν λύση στο πρόβλημα των εξαφανιζόμενων κλίσεων ήταν τα δίκτυα με ονομασία *Επανατροφοδοτούμενες Μονάδες με Πύλες (Gated Recurrent Units ή GRUs).* Ένα *GRU* είναι τυπικά ένα *LSTM* χωρίς την πύλη

εξόδου, που σημαίνει ότι καταγράφονται όλα τα περιεχόμενα από το κελί της μνήμης στο ευρύτερο δίκτυο σε χρονικό βήμα. Οι εξισώσεις που περιγράφουν την λειτουργία ενός GRU είναι ως εξής :

$$z = \sigma(x_t U^z + s_{t-1} W^z)$$
$$r = \sigma(x_t U^r + s_{t-1} W^r)$$
$$h = tanh(x_t U^h + (s_{t-1} \circ r) W^h)$$
$$s_t = (1 - z) \circ h + z \circ s_{t-1}$$



Σχήμα 3.6 Αρχιτεκτονική ενός GRU

4

Υλοποίηση &

Σχεδιασμός Μοντέλου

Στο κεφάλαιο αυτό θα αναλύσουμε το νευρωνικό και πιθανοτικό μοντέλο το οποίο χρησιμοποιήσαμε προκειμένου να παράγουμε περιγραφές του περιεχομένου των εικόνων.

Τελευταίες πρόοδοι στην πιθανοτική μετάφραση μηχανών έχουν αποδείξει ότι, αν έχουμε στη διάθεσή μας ένα ισχυρό ακολουθιακό μοντέλο, είναι πιθανό να επιτύχουμε βέλτιστα αποτελέσματα, με το να μεγιστοποιήσουμε την πιθανότητα της σωστής μετάφρασης, δεδομένης μιας ακολουθίας εισόδου. Τα μοντέλα αυτά κάνουν χρήση επαναλαμβανόμενων νευρωνικών δικτύων, τα οποία κωδικοποιούν το μήκος της μεταβλητής εισόδου σε ένα-σταθερών διαστάσεων- διάνυσμα και χρησιμοποιούν την κωδικοποίηση αυτήν για να την αποκωδικοποιήσουν, στην συνέχεια, στην επιθυμητή ακολουθία εξόδου. Είναι φυσικό, επομένως, να χρησιμοποιήσουμε και εμείς μια παρεμφερή προσέγγιση για το δικό μας πρόβλημα, όπου, δεδομένης μιας εικόνας εισόδου (αντί για μια ακολουθία εισόδου), εφαρμόζουμε την ίδια διαδικασία "μετάφρασης" της εικόνας σε μια περιγραφή, όπως φαίνεται στο παράδειγμα του Σχήματος 4.1.

Σχήμα 4.1 Παράδειγμα λειτουργίας του μοντέλου που δημιουργήσαμε το οποίο, αυτόματα, δημιουργεί περιγραφές του περιεχομένου των εικόνων

4.1 Επισκόπηση Αρχιτεκτονικής Μοντέλου

Το μοντέλο που χρησιμοποιήσαμε, όπως αναφέραμε και προηγουμένως, αποτελεί ένα κλασικό παράδειγμα ενός νευρωνικού δικτύου κωδικοποίησης-αποκωδικοποίησης *(encoder-decoder neural network)*: Αρχικά, κωδικοποιεί την εικόνα που δέχεται σαν είσοδο σε μια αναπαράσταση ενός διανύσματος με σταθερό μήκος και στην συνέχεια αποκωδικοποιεί την αναπαράσταση αυτήν σε μια περιγραφή φυσικής γλώσσας.

Σκοπός δημιουργίας του μοντέλου μας είναι να μεγιστοποιήσουμε την πιθανότητα μιας σωστής περιγραφής, δεδομένης μιας εικόνας, χρησιμοποιώντας την παρακάτω σχεση:

$$\theta^* = \arg\max_{\theta} \sum_{(I,S)} logp(S|I;\theta)$$

,

όπου θ είναι οι παράμετροι του μοντέλου μας, I είναι μια εικόνα εισόδου και S η σωστή περιγραφή της εικόνας. Εφ' όσον το S αποτελεί μία πρόταση, το μέγεθός του δεν έχει όρια. Έτσι, είναι σύνηθες να εφαρμόζουμε τον κανόνα αλυσίδας προκειμένου να μοντελοποιήσουμε την ομαδική πιθανότητα των $S_0, ..., S_N$, όπου N είναι το μήκος ενός συγκεκριμένου παραδείγματος, ως :

$$log(p(S|I) = \sum_{t=0}^{N} logp(S_t|I, S_0, ..., S_{t-1})$$

,

όπου αφήσαμε το θ εκτός για λόγους ευκολίας.

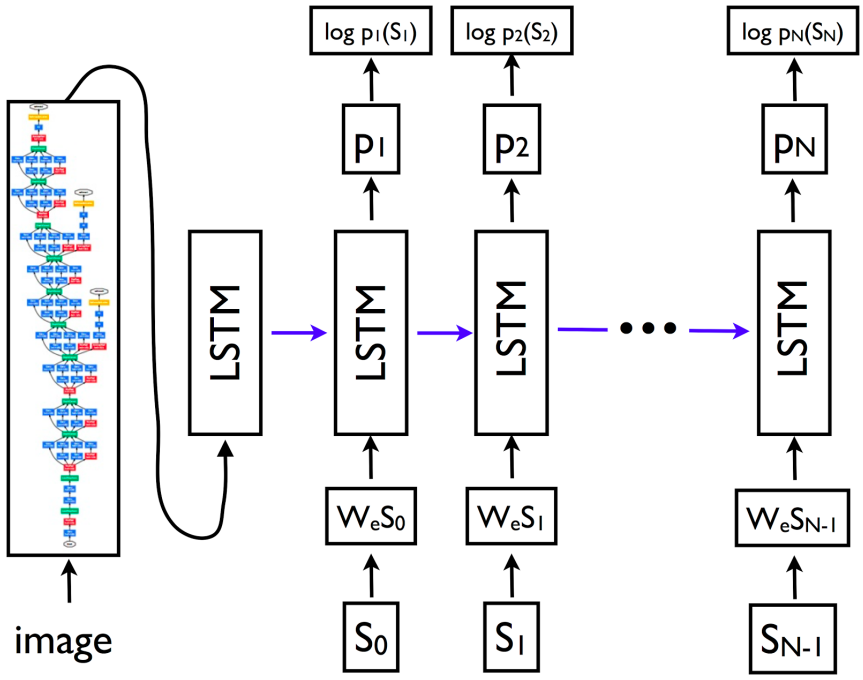
Για την κωδικοποίηση της εικόνας χρησιμοποιήσαμε ένα Βαθύ Συνελικτικό Νευρωνικό Δίκτυο (*CNN*), όπως το είδαμε στο δεύτερο κεφάλαιο, καθώς όπως έχουμε ήδη αναφέρει, τα δίκτυα αυτά είναι ευρέως χρησιμοποιούμενα σε εφαρμογές ανίχνευσης και αναγνώρισης αντικειμένων και έχουν αποδειχθεί ότι αποδίδουν βέλτιστα σε σχέση με τις ήδη υπάρχουσες μεθόδους.

Ο αποκωδικοποιητής που χρησιμοποιήσαμε ήταν ένα δίκτυο Διαρκούς Μονάδας με Μικρή Περίοδο Μνήμης (*LSTM*). Τέτοια δίκτυα χρησιμοποιούνται ευρέως για εφαρμογές

μοντελοποίησης ακολουθιών, όπως είναι η μοντελοποίηση γλώσσας και η μετάφραση μηχανών. Το κομμάτι αυτό του συστήματος είναι και αυτό που παράγει την έξοδό μας.

Οι λέξεις που εμφανίζονται στις περιγραφές και αποτελούν και είσοδο στο *LSTM* αναπαριστώνται με τη βοήθεια ενός εμφυτευμένου μοντέλου (*embedding model*), όπως αυτό θα το αναλύσουμε στην συνέχεια.

Στο παρακάτω σχήμα φαίνεται η γενική αρχιτεκτονική του μοντέλου που χρησιμοποιήσαμε και του οποίου τα συστατικά μέρη και οι τεχνικές που χρησιμοποιήθηκαν αναλύονται στις επόμενες παραγράφους.



Σχήμα 4.2 Αρχιτεκτονική του μοντέλου που χρησιμοποιήσαμε

4.2 Αναπαράσταση Εικόνων

Όπως έχουμε ήδη αναφέρει, το πρώτο στάδιο του μοντέλου που χρησιμοποιήσαμε είναι ένα Βαθύ Συνελικτικό Νευρωνικό Δίκτυο. Ο σκοπός του δικτύου αυτού είναι να κωδικοποιήσει την εικόνα εισόδου, εξάγοντας, ουσιαστικά, συγκεκριμένα χαρακτηριστικά από αυτήν, τα οποία, στη συνέχεια, προωθούνται στο LSTM με τη μορφή ενός διανύσματος σατθερών διαστάσεων.

Για τις ανάγκες της υλοποίησης του μοντέλου μας, χρησιμοποιήσαμε ένα γνωστό Συνελικτικό Νευρωνικό Δίκτυο, το οποίο ονομάζεται ***Inception V3***. Ο λόγος της επιλογής του συγκεκριμένου δικτύου έχει να κάνει με την εφαρμογή πρόσφατων τεχνικών *κανονικοποίησης ομάδων παραδειγμάτων* (*batch normalization*), καθώς και ότι επίσης , το συγκεκριμένο δίκτυο, απέσπασε την πρώτη θέση στον διαγωνισμό κατηγοριοποίησης εικόνων *ILSVRC 2014*. Ακόμη, έχει αποδειχθεί ότι μπορεί να γενικεύσει εύκολα και σε άλλες εφαρμογές, όπως την κατηγοριοποίηση τοπίων, χρησιμοποιώντας τεχνικές μεταφερόμενης μάθησης. Στη συνέχεια, παρουσιάζεται αναλυτικότερα το συγκεκριμένο δίκτυο.

4.2.1 Χαρακτηριστικά του δικτύου Inception V3

Το Βαθύ Συνελικτικό Δίκτυο *Inception V3* είναι ένα δίκτυο δημιουργημένο από τους ανθρώπους της *Google*. Οι βασικές αρχές που διέπουν το συγκεκριμένο δίκτυο και οι οποίες το κάνουν αποδοτικό σε σύγκριση με άλλα παρεμφερή δίκτυα είναι οι εξής:

- Αποφυγή συμφόρησης (*bottleneck*)- ιδιαίτερα στην αρχή του δικτύου.
- Αναπαραστάσεις υψηλών διαστάσεων, καθώς είναι ευκολότερο να προοδεύσουν τοπικά σε ένα δίκτυο.
- Υλοποίηση χωρικής συγκέντρωσης (*spatial aggregation*) σε μικρών διαστάσεων εμφυτεύσεις. Για παράδειγμα, πριν την υλοποίηση μιας 3x3 συνέλιξης, μπορούμε να μειώσουμε τις διαστάσεις της αναπαράστασης της εισόδου, χωρίς να αντιμετωπίσουμε κάποιες σοβαρές επιπτώσεις.
- Ισορροπία μεταξύ του πλάτους και του βάθους του δικτύου.

Ένα, ακόμη, σημαντικό χαρακτηριστικό που κάνει το συγκεκριμένο δίκτυο αρκετά αποδοτικό είναι η ***παραγοντοποίηση σε μικρότερες συνελίξεις (factorization into smaller convolutions)***. Οι συνελίξεις με μεγάλα χωρικά φίλτρα (π.χ. 5x5 ή 7x7) τείνουν να είναι δυσανάλογα “ακριβές” όσον αφορά τους υπολογισμούς. Για παράδειγμα, μια 5x5 συνέλιξη με n φίλτρα, σε ένα πλέγμα με m φίλτρα είναι 25/9 = 2.78 φορές πιο “ακριβή” υπολογιστικά από ότι μια 3x3 συνέλιξη με τον ίδιο αριθμό φίλτρων. Οι μεγαλύτερες συνελίξεις, ωστόσο, είναι γνωστό ότι μπορούν να εντοπίζουν καλύτερα τις εξαρτήσεις μεταξύ των δεδομένων και ως εκ τούτου να αυξάνουν την εκφραστικότητα των εξόδων. Για τον λόγο αυτόν, αποδείχθηκε ότι η αντικατάσταση μιας μεγαλύτερης συνέλιξης με έναν αριθμό από μικρότερες συνελίξεις, μπορεί να αποδώσει το ίδιο καλά, κοστίζοντας, ωστόσο, πολύ λιγότερο υπολογιστικά. Στο

παράδειγμά μας, η 5x5 συνέλιξη μπορεί να αντικατασταθεί από δύο 3x3 συνελίξεις παρουσιάζοντας τα ίδια αποτελέσματα.

4.2.2 Αρχιτεκτονική του δικτύου Inception V3

Με βάση τις βασικές αρχές και τεχνικές σχεδιασμού που περιγράψαμε στην προηγούμενη ενότητα, στην παράγραφο αυτή, θα παρουσιάσουμε την ακριβή αρχιτεκτονική του δικτύου *Inception V3*.

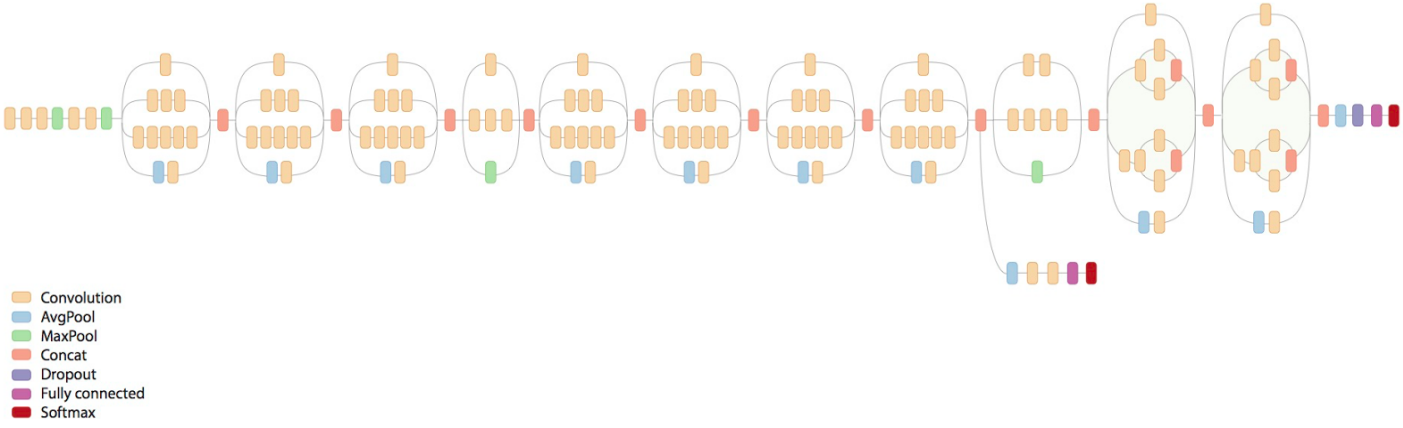
Όπως φαίνεται και στο Σχήμα 4.3, η αρχιτεκτονική του δικτύου βασίζεται σε επιμέρους μονάδες (*modules*) . Πιο αναλυτικά, το δίκτυο αποτελείται από τα εξής επεξεργαστικά επίπεδα με τη σειρά που αναγράφονται) :

- Τρία (3) συνελικτικά επίπεδα, μεγέθους 3x3
- Ένα (1) συγκεντρωτικό επίπεδο μεγίστου, μεγέθους 3x3
- Ένα (1) συνελικτικό επίπεδο, μεγέθους 1x1
- Ένα (1) συνελικτικό επίπεδο, μεγέθους 1x1
- Ένα (1) συνελικτικό επίπεδο, μεγέθους 3x3
- Ένα (1) συγκεντρωτικό επίπεδο μεγίστου (Max Pooling), μεγέθους 3x3
- Τρεις (3), ίδιου τύπου, μονάδες *Inception,* κάθε μία από τις οποίες αποτελείται από:
    - Τέσσερα (4) συνελικτικά επίπεδα, μεγέθους 1x1
    - Δύο (2) συνελικτικά επίπεδα, μεγέθους 3x3
    - Ένα (1) συνελικτικό επίπεδο, μεγέθος 5x5
    - Ένα (1) συγκεντρωτικό επίπεδο μέσου όρου (Average Pooling), μεγέθους 3x3
- Μία (1) μονάδα *Inception*, η οποία αποτελείται από:
    - Τρία (3) συνελικτικά επίπεδα, μεγέθους 3x3
    - Ένα (1) συνελικτικό επίπεδο, μεγέθους 1x1
    - Ένα (1) συγκεντρωτικό επίπεδο μεγίστου, μεγέθους 3x3
- Τέσσερις (4) μονάδες *Inception*, κάθε μία από τις οποίες αποτελείται από:
    - Τέσσερα (4) συνελικτικά επίπεδα, μεγέθους 1x1
    - Τρία (3) συνελικτικά επίπεδα, μεγέθους 1x7
    - Τρία (3) συνελικτικά επίπεδα, μεγέθους 7x1
    - Ένα (1) συγκεντρωτικό επίπεδο μέσου όρου, μεγέθους 3x3
- Μία (1) μονάδα *Inception*, η οποία αποτελείται από:
    - Δύο (2) συνελικτικά επίπεδα, μεγέθους 1x1
    - Δύο (2) συνελικτικά επίπεδα, μεγέθους 3x3

    - Ένα (1) συνελικτικό επίπεδο, μεγέθους 1x7
    - Ένα (1) συνελικτικό επίπεδο μεγέθους 7x1
    - Ένα (1) συγκεντρωτικό επίπεδο μεγίστου, μεγέθους 3x3
- Δύο (2) μονάδες *Inception*, κάθε μια από τις οποίες αποτελείται από:
    - Τέσσερα (4) συνελικτικά επίπεδα, μεγέθους 1x1
    - Δύο (2) συνελικτικά επίπεδα, μεγέθους 1x3
    - Δύο (2) συνελικτικά επίπεδα, μεγέθους 3x1
    - Ένα (1) συνελικτικό επίπεδο, μεγέθους 3x3
    - Ένα (1) συγκεντρωτικό επίπεδο μέσου όρου, μεγέθους 3x3
- Ένα (1) επίπεδο περιορισμού ενεργοποίησης (*Dropout*)
- Ένα (1) πλήρως συνδεδεμένο επίπεδο (*Fully-Connected*)
- Ένα (1) επίπεδο κατηγοριοποίησης (*Softmax Classifier*)

Αξίζει να αναφερθεί το γεγονός ότι το συγκεκριμένο δίκτυο πετυχαίνει 21.2% *top-1* και 5.6% *top-5* σφάλματος για αξιολόγηση μονού πλαισίου, με ένα υπολογιστικό κόστος της τάξης των 5 δισεκατομμυρίων πολλαπλασιασμών-προσθέσεων ανά συμπερασμό και με τη χρήση λιγότερων από 25 εκατομμυρίων παραμέτρων. Η εκπαίδευση του δικτύου αυτού έχει γίνει σε χίλιες (1000) κλάσεις, που σημαίνει ότι είναι ικανό να αναγνωρίζει χίλια (1000) διαφορετικά αντικείμενα.

Στην συνέχεια παρατίθεται μια οπτικοποίηση της αρχιτεκτονικής του δικτύου *Inception V3*, όπως αυτό περιγράφηκε παραπάνω.



Σχήμα 4.3 Οπτικοποίηση της αρχιτεκτονικής του δικτύου Inception V3

4.2.3 Ενσωμάτωση του Inception V3 στο Σύστημά μας

Όπως έχουμε ήδη αναφέρει, το δίκτυο *Inception V3* αποτελεί το πρώτο συστατικό κομμάτι του συνολικού μας συστήματος. Είναι το κομμάτι στο οποίο τροφοδοτούμε την εικόνα εισόδου, της οποίας θέλουμε να παράγουμε την τελική περιγραφή.

Το δίκτυο αυτό, λοιπόν, δέχεται σαν είσοδο μια έγχρωμη εικόνα διαστάσεων ***299x299,*** η οποία μεταφράζεται σε έναν πίνακα διαστάσεων ***299x299x3***, όπου το 299x299 αναφέρεται στις διαστάσεις της εικόνας (ύψος και πλάτος της εικόνας αντίστοιχα) μετρούμενες σε εικονοστοιχεία (*pixels*), ενώ το 3 αναφέρεται στο κανάλι χρωμάτων *RGB (Red-Green-Blue)*. Η εικόνα, στην συνέχεια, περνάει από τα διάφορα επεξεργαστικά στάδια που αναφέραμε παραπάνω, εξάγοντας έτσι, κρίσιμα χαρακτηριστικά της εικόνας.

Αξίζει να σημειώσουμε σε αυτό το σημείο ότι δεν μας ενδιαφέρει να μας πει το *Inception V3* τι περιέχεται στην εικόνα ρητά, καθώς δεν πρόκειται για ένα πρόβλημα κατηγοριοποίησης εικόνας. Αυτό που μας ενδιαφέρει είναι να πάρουμε κάποια σημαντικά χαρακτηριστικά της εικόνας τα οποία θα προωθηθούν στο *LSTM*. Για τον λόγο αυτόν, δεν λαμβάνουμε την έξοδο

του δικτύου αυτού από το τελευταίο επιμέρους επίπεδό του, το οποίο είναι ένας κατηγοριοποιητής *Softmax*, καθώς η έξοδος του επιπέδου αυτού θα μας έδινε μία πιθανοτική κατανομή μεταξύ των κλάσεων στο οποίο είναι εκπαιδευμένο. Για τον ίδιο λόγο, δεν χρειαζόμαστε και το προτελευταίο στάδιο του δικτύου που είναι ένα πλήρως συνδεδεμένο επίπεδο. Αντ' αυτού, λαμβάνουμε την έξοδό μας από το τρίτο από το τέλος επίπεδο, το οποίο είναι ένα επίπεδο περιορισμού ενεργοποίησης. Από το επίπεδο αυτό, παίρνουμε έναν πίνακα τιμών, διαστάσεων ***8x8x2048***, ο οποίος, πλέον, αποτελεί την αναπαράσταση της εικόνας εισόδου. Έπειτα, με χρήση μιας ισοπεδωτικής συνάρτησης (*flatten*) μετασχηματίζουμε την αναπαράσταση της εικόνας, δίνοντάς της διαστάσεις μέγεθος_ομάδας_παραδειγμάτων *(batch_size)* x 2048.

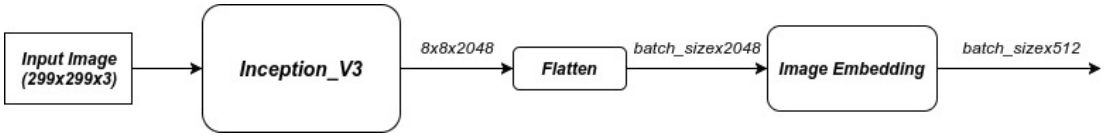
4.2.4 Εμφύτευση Εικόνων (Image Embedding)

Ο πίνακας εξόδου του δικτύου *Inception V3*, όπως έχουμε ήδη αναφέρει, θα αποτελέσει την είσοδο στο δίκτυο *LSTM*, το οποίο είναι υπεύθυνο για την παραγωγή λογικών προτάσεων περιγραφής των εικόνων. Κατά τη διάρκεια της εκπαίδευσης, όπως θα δούμε και στο επόμενο κεφάλαιο, η εικόνα αποτελεί την είσοδο **μόνο** στο πρώτο βήμα λειτουργίας του *LSTM*. Στα επόμενα βήματα, η είσοδος του δικτύου *LSTM* είναι μια αναπαράσταση της εκάστοτε προηγούμενης λέξης (αναλύεται παρακάτω) σε μία πρόταση που δημιουργείται εκείνη τη στιγμή.

Όπως γίνεται κατανοητό, οι διαστάσεις της αναπαράστασης της εικόνας, όπως επίσης και οι διαστάσεις των αναπαραστάσεων των λέξεων πρέπει να είναι ίσες, προκειμένου το δίκτυο *LSTM* να λειτουργήσει σωστά. Για τον λόγο αυτόν ακριβώς, αλλά και για λόγους μείωσης του υπολογιστικού κόστους και καλύτερης διαχείρισης των δεδομένων, θα πρέπει να απεικονίσουμε την αναπαράσταση της εικόνας στις ίδιες διαστάσεις με αυτές των λέξεων. Όπως θα δούμε και στην συνέχεια, οι λέξεις θα απεικονίζονται σε διανύσματα 512 διαστάσεων, και, επομένως θα πρέπει να μετασχηματίσουμε την αναπαράσταση της εικόνας σε διαστάσεις *batch_size x 512*.

Για τον μετασχηματισμό αυτόν θα χρειαστεί να προσθέσουμε ένα πλήρως συνδεδεμένο επίπεδο, όπως αυτά που αναλύσαμε στο Κεφάλαιο 2, αμέσως μετά της χρήσης της ισοπεδωτικής συνάρτησης, το οποίο θα έχει 512 εξόδους και θα δέχεται ως είσοδο την έξοδο της συνάρτησης αυτής. Η έξοδος του επιπέδου αυτού θα είναι ένας πίνακας διαστάσεων

batch_size x 512. Στο Σχήμα 4.4 μπορούμε να λάβουμε μια γενική εικόνα του τρόπου που αναπαριστούνται οι εικόνες στο σύστημά μας.



Σχήμα 4.4 Αναπαράσταση των εικόνων του μοντέλου μας

4.3 Αναπαράσταση Λέξεων

Κατά την διαδικασία της *εκπαίδευσης* (*training*), αλλά και αυτήν του *συμπερασμού* (*inference*), έχουμε την ανάγκη για αποδοτική επεξεργασία και χρησιμοποίηση των λέξεων που αποτελούν κομμάτια της τελικής περιγραφής που παράγεται. Για τον λόγο αυτόν, όπως γίνεται εύκολα αντιληπτό, η αναπαράσταση των λέξεων σαν ακολουθία χαρακτήρων (*strings*), δυσκολεύει αρκετά την αποδοτική λειτουργία του συστήματός μας. Έτσι, προκειμένου να απλοποιήσουμε την λειτουργία του μοντέλου μας, θα αναπαραστήσουμε τις λέξεις σαν ***ακέραιους αριθμούς***, οι οποίοι, ακολούθως, θα εμφυτευθούν σε ***διανύσματα σταθερού μήκους***, όπως θα αναλύσουμε στην συνέχεια. Ας δούμε αναλυτικότερα τα επιμέρους βήματα της αναπαράστασης των λέξεων.

4.3.1 Δημιουργία Λεξικού

Οι λέξεις οι οποίες θα χρησιμοποιηθούν προκειμένου να παραχθεί μία τελική πρόταση περιγραφής μιας εικόνας, όπως γίνεται εύκολα αντιληπτό, θα πρέπει να προέρχονται από ένα σύνολο λέξεων, το οποίο θα λειτουργεί σαν ***λεξικό (vocabulary)***.

Το σύνολο δεδομένων εκπαίδευσης το οποίο έχουμε στην διάθεσή μας και το οποίο θα δούμε αναλυτικότερα στο επόμενο κεφάλαιο, αποτελείται από έναν αριθμό εικόνων και από τις αντίστοιχες περιγραφές τους σε ακολουθίες χαρακτήρων στην αγγλική γλώσσα.

Το *λεξικό* που δημιουργήσαμε αποτελείται από όλες τις διαφορετικές λέξεις που εμφανίζονται στις περιγραφές των εικόνων του συνόλου δεδομένων εκπαίδευσης, με τον μόνο περιορισμό, ότι για να εισαχθεί μία λέξη στο λεξικό μας, θα πρέπει να εμφανίζεται τουλάχιστον τέσσερις (4) φορές, συνολικά, στις περιγραφές των εικόνων του συνόλου δεδομένων μας. Για την διευκόλυνση δημιουργίας του λεξικού μας, μετασχηματίσαμε τις περιγραφές των εικόνων σε

λίστες από λέξεις, χωρισμένες με κόμμα μεταξύ τους (*comma-separated*), όπως φαίνεται στο παράδειγμα που ακολουθεί:

$$A\ person\ on\ a\ beach\ flying\ a\ kite\ .\ \rightarrow [A, person, on, a, beach, flying, a, kite, .]$$

Αξίζει, επίσης, να σημειώσουμε ότι κάθε πρόταση συνοδεύεται από δύο ειδικούς χαρακτήρες <*S*> και </*S*>, οι οποίοι υποδεικνύουν την αρχή και το τέλος μιας πρότασης, αντίστοιχα, και οι οποίοι συμπεριλαμβάνονται στο λεξικό μας, όπως επίσης και τα σημεία στίξης.

Συνολικά, το λεξικό μας αποτελείται από *11.519 λέξεις*, **ταξινομημένες σε φθίνουσα σειρά**, με βάση τη συχνότητα εμφάνισής τους στις περιγραφές του συνόλου δεδομένων εκπαίδευσης που έχουμε στην διάθεσή μας. Η μορφή που παίρνει, επομένως, το λεξικό μας είναι σαν και αυτήν που παρουσιάζεται στο Σχήμα 4.5.

Αρ.Σειράς	*Λέξεις*	*Αρ.Εμφανίσεων*
0	*a*	*969108*
1	*<S>*	*586368*
2	*</S>*	*586368*
. . .	. . .	. . .
11.519	*moguls*	*4*

Σχήμα 4.5 Αναπαράσταση του λεξικού που δημιουργήσαμε

4.3.2 Αντιστοίχιση Λέξεων σε Ακεραίους

Προκειμένου, όπως έχουμε ήδη αναφέρει, να διευκολύνουμε την χρήση των λέξεων στις διαδικασίες της εκπαίδευσης και του συμπερασμού του μοντέλου μας, θα αναπαραστήσουμε τις λέξεις με ακέραιους αριθμούς. Οι λέξεις που απαιτούν αυτήν την αντιστοίχιση, όπως καταλαβαίνουμε, είναι μόνο οι λέξεις που περιέχονται στο λεξικό. Η αντιστοίχιση που επιλέξαμε να κάνουμε είναι αρκετά απλή και αποτελείται από τον εξής κανόνα:

- ***Κάθε λέξη του λεξικού αναπαρίσταται με τον αντίστοιχο ακέραιο αριθμό της σειράς στην οποία βρίσκεται η λέξη αυτή, μέσα στο λεξικό.***

Έτσι, για το παράδειγμα του Σχήματος 4.5, η λέξη "*a"* θα αναπαρασταθεί με τον ακέραιο αριθμό 0, η λέξη "<*S*>" με τον ακέραιο αριθμό 1 και η λέξη "*moguls*" με τον ακέραιο αριθμό

11.519. Στο παράδειγμα που ακολουθεί φαίνονται κάποιες χαρακτηριστικές αντιστοιχίσεις λέξεων σε ακεραίους που έγιναν.

$on \rightarrow 21362$
$is \rightarrow 97322$
$people \rightarrow 41672$
$, \rightarrow 43921$
$street \rightarrow 30173$
$bacon \rightarrow 424$
$east \rightarrow 29$

4.3.3 Εμφύτευση Λέξεων σε Διανύσματα (Word Embedding Vectors)

Πολλά συστήματα και τεχνικές Επεξεργασίας Φυσικής Γλώσσας χειρίζονται τις λέξεις σαν ατομικές μονάδες, χωρίς να υπάρχει η έννοια της της ομοιότητας μεταξύ των λέξεων, καθώς αυτές αναπαριστώνται σαν δείκτες σε ένα λεξικό, όπως δηλαδή έχουμε περιγράψει την διαδικασία αναπαράστασης των λέξεων μέχρι τώρα. Θα είχαμε πολλούς καλούς λόγους να μείνουμε σε αυτήν την επιλογή αναπαράστασης καθώς το σύστημά μας θα παρουσίαζε απλότητα και στιβαρότητα.

Ωστόσο, όλες οι απλές τεχνικές φτάνουν στα όριά τους σε αρκετές εφαρμογές, όπως ισχύει και στην δική μας περίπτωση. Για τον λόγο αυτόν, οι λέξεις, οι οποίες πλέον αναπαρίστανται ως ακέραιοι αριθμοί, θα επιλέγουμε να αναπαριστώνται με ***διανύσματα εμφύτευσης σταθερού μήκους (word embedding vectors)***. Ας δούμε, όμως, αναλυτικότερα τι είναι αυτά τα διανύσματα.

Η βασική ιδέα των διανυσμάτων αυτών είναι να μετατρέπουν τις λέξεις σε διανύσματα σταθερού μήκους, τα οποία περιέχουν πραγματικούς αριθμούς. Η μετατροπή αυτή είναι απαραίτητη, καθώς οι περισσότεροι αλγόριθμοι μηχανικής μάθησης, συμπεριλαμβανομένων και των βαθιών νευρωνικών δικτύων, απαιτούν οι είσοδοι να είναι διανύσματα πραγματικών τιμών. Πέραν από το γεγονός, ότι αυτή η αναπαράσταση είναι σύμφωνη με τους αλγόριθμους εκπαίδευσης, έχει άλλες δύο σημαντικές και πλεονεκτικές ιδιότητες:

- ***Μείωση Διαστάσεων (Dimensionality Reduction)*** - μια πιο αποδοτική αναπαράσταση
- ***Σημασιολογική Ομοιότητα Περιεχομένου Μεταξύ Λέξεων (Contextual Similarity***) - μια πιο εκφραστική αναπαράσταση

Όσον αφορά την ιδιότητα της *μείωσης των διαστάσεων*, γνωρίζουμε ότι ήδη υπάρχουσες τεχνικές αναπαράστασης, όπως οι *Σακούλες Λέξεων (Bag-of-Words),* καταλήγουν σε *one-hot*

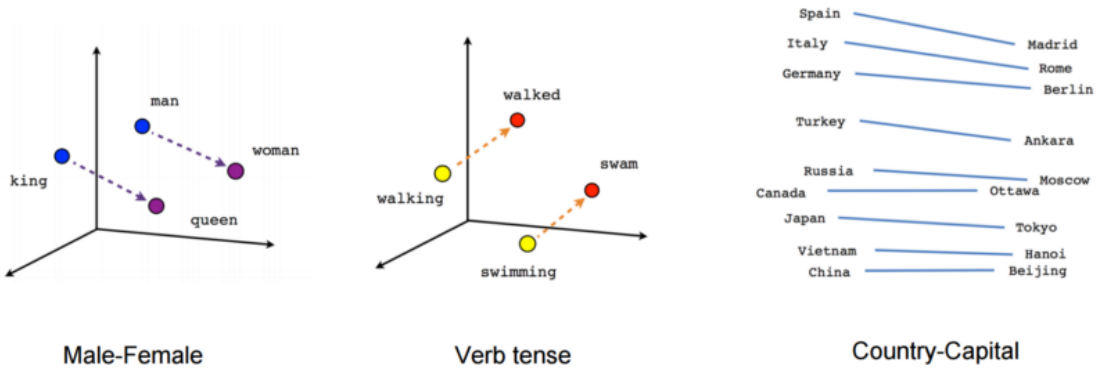
διανύσματα τα οποία έχουν αρκετά μεγάλες διαστάσεις, των οποίων ο αριθμός είναι ίσος με το μέγεθος του λεξικού. Σκοπός των διανυσμάτων εμφύτευσης είναι να δημιουργηθούν αναπαραστάσεις των λέξεων, των οποίων το μέγεθος θα είναι ανεξάρτητο από το μέγεθος του λεξικού και επίσης θα έχουν και αρκετά μικρότερο αριθμό διαστάσεων.

Τα διανύσματα εμφύτευσης, επίσης, χρησιμοποιούνται για συντακτική ανάλυση, με την έννοια ότι εξάγουν νόημα από το κείμενο, προκειμένου να προωθήσουν την καλύτερη κατανόηση της φυσικής γλώσσας. Για να είναι ικανό ένα γλωσσικό μοντέλο, σαν το δικό μας, να προβλέπει την φυσική σημασία μιας πρότασης, θα πρέπει να γνωρίζει και την *σημασιολογική ομοιότητα του περιεχομένου μεταξύ των λέξεων*. Για παράδειγμα, σαν αποτέλεσμα των διανυσμάτων αυτών, θα περιμέναμε να βρούμε λέξεις που σχετίζονται με τα φρούτα, όπως "γινομένα", "χυμώδη", "φαγωμένα" να βρίσκονται αρκετά κοντά τους σημασιολογικά, αλλά για μια λέξη σαν την "αεροπλάνο", δεν θα περιμέναμε να βρίσκεται σε κοντινή απόσταση με τις προαναφερθείσες.

Τα διανύσματα που δημιουργούνται από τις εμφυτεύσεις λέξεων, είναι ικανά να διατηρούν τις ομοιότητες αυτές, έτσι ώστε λέξεις που, τακτικά, εμφανίζονται κοντά μεταξύ τους στις προτάσεις, να βρίσκονται, επίσης, κοντά στο χώρο των διανυσμάτων.

Μια σύντομη, έτσι, απάντηση στο τι είναι τα διανύσματα εμφύτευσης λέξεων, θα μπορούσαμε να πούμε ότι: ***Είναι μια μέθοδος κατασκευής, χαμηλών διαστάσεων, διανυσμάτων αναπαράστασης λέξεων, τα οποία διατηρούν μια σημασιολογική ομοιότητα περιεχομένου μεταξύ των λέξεων.***

Στο σχήμα 4.6 μπορούμε να δούμε κάποιες σημασιολογικές ομοιότητες μεταξύ λέξεων. Για παράδειγμα, η λέξη "king" και η λέξη "queen" θα βρεθούν αρκετά κοντά λόγω της σχέσης άνδρα-γυναίκας που έχουν μεταξύ τους. Επίσης, οι λέξεις "swimming" και "swam" θα βρεθούν σε κοντινή απόσταση, αφού το μοντέλο αυτό θα έχει εντοπίσει τη σχέση που συνδέει τα δύο ρήματα και είναι ότι αποτελούν το ίδιο, ουσιαστικά, ρήμα σε άλλο χρόνο.



Σχήμα 4.6 Σημασιολογικές ομοιότητες μεταξύ κάποιων διανυσμάτων εμφύτευσης

Ειδικότερα στο μοντέλο μας, οι διαστάσεις των διανυσμάτων εμφύτευσης που χρησιμοποιήσαμε είναι *512*. Αυτό σημαίνει ότι κάθε λέξη - στην οποία αναφερόμαστε σαν ακέραιο - αναπαρίσταται με ένα διάνυσμα μεγέθους *512*, το οποίο περιέχει πραγματικές τιμές, οι οποίες εκφράζουν την σημασιολογική ομοιότητα της λέξης αυτής με κάποιες άλλες λέξεις. Έτσι, το σύνολο των απεικονίσεων των λέξεων σε διανύσματα υλοποιείται με τη χρήση ενός δισδιάστατου πίνακα, οποίος έχει διαστάσεις *11.519 x 512*, όπου ο αριθμός 11.519 αναφέρεται στον αριθμό των λέξεων που υπάρχουν στο λεξικό, ενώ το 512 είναι οι διαστάσεις των διανυσμάτων εμφύτευσης, όπως τα αναλύσαμε και τα οποία περιέχουν πραγματικές τιμές. Κάθε σειρά του πίνακα εμφύτευσής μας θα έχει την παρακάτω μορφή του παραδείγματος:

$$W(\text{"}cat\text{"}) = (0.2, -0.4, 0.7, ...)$$

4.4 Γεννήτρια Προτάσεων Βασισμένη σε LSTM

Στην παράγραφο αυτή θα αναλύσουμε το πιο σημαντικό, ίσως, κομμάτι του συστήματός μας το οποίο είναι ένα δίκτυο *LSTM*, σαν και εκείνα που περιγράφηκαν στο τρίτο κεφάλαιο και το οποίο είναι υπεύθυνο για την δημιουργία λογικών προτάσεων περιγραφής του περιεχομένου των εικόνων.

Η επιλογή του δικτύου *LSTM*, έγινε λόγω της ικανότητας των δικτύων αυτών να αντιμετωπίζουν με αποτελεσματικότητα το πρόβλημα των εξαφανιζόμενων και ανατινασσόμενων κλίσεων, όπως το είδαμε στο Κεφάλαιο 3 - μια μεγάλη πρόκληση στο σχεδιασμό και στην εκπαίδευση ανατροφοδοτούμενων νευρωνικών δικτύων.

Η λειτουργία του δικτύου αυτού, όπως ήδη αναφέραμε, έγκειται στην παραγωγή προτάσεων περιγραφών εικόνων. Όπως μπορούμε να δούμε και από το Σχήμα 4.2, το δίκτυο *LSTM*

δέχεται σαν είσοδο την εικόνα μόνο στο πρώτο χρονικό του βήμα. Έπειτα, η εικόνα **δεν** ξανατροφοδοτείται στο δίκτυο *LSTM*. Οι επόμενες είσοδοι του δικτύου αυτού, είναι διανύσματα εμφύτευσης λέξεων, τα οποία αποτελούν κομμάτια της πρότασης που θα δημιουργηθεί τελικά αλλά και οι κρυφές καταστάσεις των προηγούμενων χρονικών βημάτων. Πιο συγκεκριμένα, σε κάθε χρονικό βήμα, παρέχεται σαν είσοδος στο *LSTM* το διάνυσμα εμφύτευσης της λέξης εκείνης την οποία προέβλεψε το δίκτυο αυτό ότι θα αποτελεί μέρος της τελικής πρότασης, στο ακριβώς προηγούμενο χρονικό βήμα. Αυτό σημαίνει, ότι αν για παράδειγμα μέχρι την χρονική στιγμή *t-1*, η πρόταση η οποία έχει δημιουργηθεί από το δίκτυο *LSTM* είναι $"A\ man\ sitting"$, τότε, την χρονική στιγμή *t,* η είσοδος που θα δοθεί στο δίκτυο *LSTM* θα είναι το διάνυσμα εμφύτευσης της λέξης $"sitting"$, όπως επίσης και η κρυφή κατάσταση του χρονικού βήματος *t-1*. Η ένωση των επιμέρους λέξεων που παράγονται από το *LSTM,* με την σειρά που αυτές παράγονται, θα αποτελέσει την τελική περιγραφή της εικόνας εισόδου.

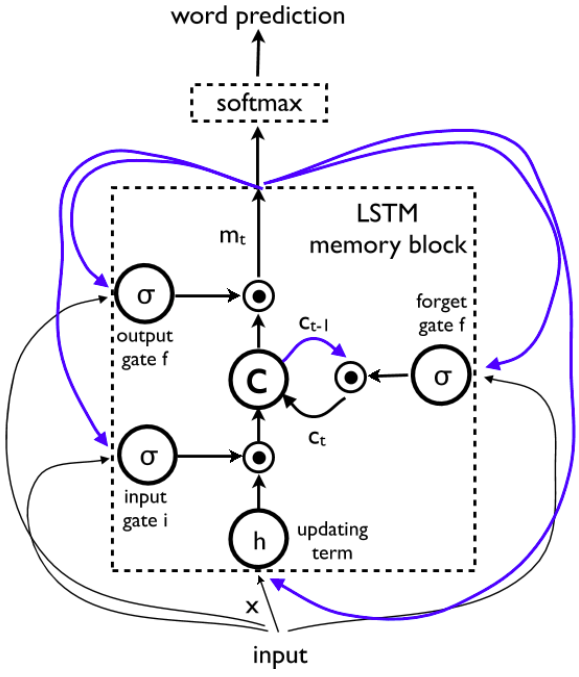
Η πρόβλεψη της επόμενης λέξης υλοποιείται με την βοήθεια ενός επιπέδου *Softmax*, το οποίο δέχεται σαν είσοδο την έξοδο ενός πλήρως συνδεδεμένου επιπέδου, όπως αυτό αναλύεται στην συνέχεια*,* και παράγει μια πιθανοτική κατανομή των λέξεων. Επειδή η είσοδος του *LSTM* έχει *512* διαστάσεις, όπως ήδη έχουμε δει, *512* διαστάσεις θα έχει και η έξοδός του. Όμως, εμείς αυτό που, ουσιαστικά, επιθυμούμε είναι μια πιθανοτική κατανομή μεταξύ των πραγματικών λέξεων και όχι μεταξύ διανυσμάτων εμφύτευσης των λέξεων, προκειμένου να επιλέγουμε την πιο πιθανή λέξη, με βάση την κατανομή, σε κάθε χρονικό βήμα. Για ,ακριβώς, αυτόν τον λόγο, έχουμε προσθέσει ένα επιπλέον *πλήρως-συνδεδεμένο επίπεδο επεξεργασίας (fully-connected layer),* του οποίου η λειτουργία είναι να αντιστοιχίζει τις *512* εξόδους του δικτύ *LSTM*, σε *11.519* εξόδους *(logits)*, κάθε μια από τις οποίες αντιστοιχίζεται και σε μια λέξη του λεξικού μας. Έτσι, τελικά, θα αποκτήσουμε μια πιθανοτική κατανομή μεταξύ 11.519 λέξεων μέσω του κατηγοριοποιητή *Softmax*, και όπως θα δούμε από τον αλγόριθμο συμπερασμού στο επόμενο κεφάλαιο, επιλέγεται η λέξη εκείνη που έχει την μεγαλύτερη πιθανότητα να αποτελεί την επόμενη λέξη σε μια πρόταση.

Παρουσιάζουμε, στην συνέχεια, αναλυτικότερα τον αλγόριθμο λειτουργίας του δικτύου *LSTM*.

Αλγόριθμος 1 Λειτουργία του δικτύου LSTM

1. Δεδομένης μια εικόνας εισόδου *I*, εκτέλεσε ένα βήμα λειτουργίας του *LSTM* και κάνε μια πρόβλεψη
2. Πρόσθεσε την πρόβλεψη στην πρόταση *S*
3. Μέχρι να σχηματισθεί μια ολοκληρωμένη πρόταση *S,* επανάλαβε:
    a. Δώσε σαν είσοδο στο *LSTM* την κρυφή κατάσταση του προηγούμενου βήματος
    b. Δώσε σαν είσοδο στο *LSTM* το διάνυσμα εμφύτευσης της πρόβλεψης του δικτύου την προηγούμενη χρονική στιγμή
    c. Εκτέλεσε ένα βήμα λειτουργίας του *LSTM* και κάνε νέα πρόβλεψη

Ο πυρήνας του μοντέλου *LSTM* είναι ένα κελί μνήμης *c* το οποίο κωδικοποιεί την γνώση του τι είδους είσοδοι έχουν παρατηρηθεί μέχρι εκείνο το χρονικό βήμα (βλ. Σχήμα 4.7).



Σχήμα 4.7 LSTM: το κομμάτι μνήμης περιέχει ένα κελί c, το οποίο ελέγχεται από τρεις πύλες. Με μπλε συμβολίζουμε τις ανατροφοδοτούμενες συνδέσεις - η έξοδος m, τη χρονική στιγμή t-1, ανατροφοδοτείται στην μνήμη τη χρονική στιγμή t μέσω τριών πυλών: Η τιμή του κελιού ανατροφοδοτείται μέσω της πύλης

διαγραφής, η λέξη που προβλέφθηκε τη χρονική στιγμή t-1, μαζί με την έξοδο μνήμης m ανατροφοδοτούνται την χρονική στιγμή t στο επίπεδο Softmax για να γίνει νέα πρόβλεψη

Η συμπεριφορά του κελιού ελέγχεται από πύλες -επίπεδα, τα οποία εφαρμόζονται πολλαπλασιαστικά και, έτσι, μπορούν είτε να κρατήσουν μια τιμή από το επίπεδο πυλών αν η πύλη έχει την τιμή 1, ή να την μηδενίσουν αν η πύλη έχει την τιμή 0. Πιο συγκεκριμένα, τρεις πύλες χρησιμοποιούνται οι οποίες ελέγχουν αν θα πρέπει να ξεχάσουν την τρέχουσα τιμή του κελιού (forget gate *f*), αν θα πρέπει να διαβάσουν την είσοδό του (input gate *i*) και αν θα πρέπει να παράγουν μια νέα τιμή κελιού (output gate *o*). Αναλυτικότερα, οι διευκρινίσεις των πυλών και των κελιών ενημέρωσης και εξόδου έχουν ως εξής:

$$
\begin{aligned}
i_t &= \sigma(W_{ix}x_t + W_{im}m_{t-1}) \\
f_t &= \sigma(W_{fx}x_t + W_{fm}m_{t-1}) \\
o_t &= \sigma(W_{ox}x_t + W_{om}m_{t-1}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot h(W_{cx}x_t + W_{cm}m_{t-1}) \\
m_t &= o_t \odot c_t \\
p_{t+1} &= Softmax(m_t)
\end{aligned}
,
$$

όπου $\odot$ αντιπροσωπεύει το γινόμενο με μία τιμή πύλης και οι διάφοροι πίνακες *W* είναι εκπαιδευμένες παράμετροι. Τέτοιες πολλαπλασιαστικές πύλες, μας δίνουν την δυνατότητα να εκαπδιεύουμε δίκτυα *LSTM* αποδοτικά, καθώς οι πύλες αυτές αντιμετωπίζουν αποτελεσματικά τις εξαφανιζόμενες κλίσεις. Οι μη-γραμμικές συναρτήσεις είναι η σιγμοειδής (*σ*) και η υπερβολική εφαπτομένη *h*. Η τελευταία εξίσωση m_t, είναι εκείνη η οποία χρησιμοποιείται για να τροφοδοτήσει το πλήρως συνδεδεμένο επίπεδο, το οποίο με τη σειρά του θα τροφοδοτήσει το *Softmax* επίπεδο, ώστε να παράξει μια κατανομή πιθανοτήτων p_t όλων των λέξεων του λεξικού.

4.5 Προγραμματιστικές Πλατφόρμες & Εργαλεία

Το μοντέλο μας υλοποιήθηκε στην αντικειμενοστραφή γλώσσα προγραμματισμού ***Python***, χρησιμοποιώντας τις βιβλιοθήκες ***NumPy*** και ***Tensorflow***.

Ειδικότερα για την βιβλιοθήκη του *Tensorflow*, στην οποία βασίστηκε η υλοποίησή μας, αποτελεί μια ανοιχτού κώδικα (*open source*) προγραμματιστική βιβλιοθήκη για αριθμητικούς υπολογισμούς, χρησιμοποιώντας γράφους ροής δεδομένων *(data flow graphs)*. Οι κόμβοι σε έναν γράφο αναπαριστούν μαθηματικές λειτουργίες, ενώ οι ακμές του γράφου αναπαριστούν πολυδιάστατους πίνακες δεδομένων (*τένσορες*), οι οποίοι επικοινωνούν μεταξύ τους. Αυτή η ευέλικτη αρχιτεκτονική μας δίνει το δικαίωμα να παρατάξουμε υπολογισμούς σε περισσότερες από μια CPUs ή GPUs με ένα μόνο API. Η βιβλιοθήκη του *Tensorflow*, δημιουργήθηκε από ερευνητές και μηχανικούς της *Google*, για τις ανάγκες έρευνας πάνω σε συστήματα Μηχανικής και Βαθιάς Μάθησης.

Ειδικότερα, στο σύστημά μας, ο γράφος Tensorflow αποτελείται από:

- Το Βαθύ Συνελικτικό Δίκτυο *Inception V3*
- Το πλήρως συνδεδεμένο επίπεδο εμφύτευσης της εικόνας
- Τον πίνακα εμφύτευσης των διανυσμάτων των λεξεων
- Το δίκτυο *LSTM*
- Το πλήρως συνδεδεμένο επίπεδο αντιστοίχισης των διανυσμάτων εμφύτευσης στις λέξεις του λεξικού
- Τον κατηγοριοποιητή *Softmax* στην έξοδο του *πλήρως συνδεδεμένου επιπέδου*

Για τις ανάγκες της απεικόνισης των περιγραφών των εικόνων ως λίστες από λέξεις, όπως περιγράψαμε στην Ενότητα 4.3, χρησιμοποιήσαμε το έτοιμο *API* ***nltk.tokenize*** από την ***Εργαλειοθήκη Φυσικής Γλώσσας (Natural Language ToolKit ή NLTK),*** το οποίο έκανε ακριβώς αυτήν την λειτουργία.

5

Εκπαίδευση Μοντέλου και Αποτελέσματα

Στο κεφάλαιο αυτό θα αναλύσουμε τις πηγές από τις οποίες αντλήσαμε τα δεδομένα μας, τους αλγορίθμους και τις τεχνικές που χρησιμοποιήθηκαν προκειμένου να εκπαιδεύσουμε το μοντέλο μας, καθώς και τα αποτελέσματα που παρουσίασε το σύστημά μας.

5.1 Βάση Δεδομένων MSCOCO 2015

Το σύνολο δεδομένων με βάση το οποίο υλοποιήσαμε τις λειτουργίες της εκπαίδευσης, της αξιολόγησης και του ελέγχου λειτουργίας είναι το σύνολο δεδομένων ***MSCOCO 2015.*** Πρόκειται για ένα σύνολο δεδομένων το οποίο χρησιμοποιήθηκε για τον διαγωνισμό ***COCO Captioning 2015***.

5.1.Χαρακτηριστικά Δεδομένων

Η επιλογή του συγκεκριμένου συνόλου δεδομένων έγινε λόγω του μεγάλου και ποικίλου αριθμού παραδειγμάτων εκπαίδευσης. Πιο συγκεκριμένα, το σύνολο δεδομένων εκπαίδευσης περιελάμβανε:

- ***91 κατηγορίες συνηθισμένων αντικειμένων***
    - *82 από τις κατηγορίες αυτές περιείχαν πάνω από 5000 περιγραφές η κάθε μία*
- ***2500000 συνολικές περιγραφές για 328000 εικόνες***
- ***7,7 περιγραφές για κάθε αντικείμενο κατά μέσο όρο***

Τα δεδομένα, όπως επίσημα παρέχονται από την ιστοσελίδα της *COCO*, αποτελούνται από τις εικόνες οι οποίες είναι σε μορφή ***JPEG***, μαζί με τις περιγραφές τους οι οποίες βρίσκονται σε αρχεία ***JSON***. Κάθε εικόνα έχει τουλάχιστον 5 περιγραφές, ενώ κάποιες εικόνες έχουν και παραπάνω. Η μορφή στην οποία παρέχονται οι περιγραφές είναι η εξής :

```
annotation {
    "id"                : int,
    "image_id"          : int,
    "caption"           : str,
}
```

όπου ***id*** είναι ο αναγνωριστικός αριθμός της περιγραφής, ***image_id*** είναι ο αναγνωριστικό αριθμός της εικόνας στην οποίαν αναφέρεται η περιγραφή και ***caption*** είναι η περιγραφή της εικόνας αυτής. Μαζί με τις εικόνες και περιγραφές, προσφέρεται και ένα ***API*** το οποίο διευκολύνει την διαχείριση και επεξεργασία των εικόνων, το οποίο, όμως, εμείς δεν χρειαστήκαμε.

5.1.2 Προεπεξεργασία Δεδομένων

Προκειμένου να χειριστούμε τα δεδομένα με περισσότερη ευκολία και αποδοτικότητα, χρειάστηκε να κάνουμε κάποια επεξεργασία πάνω σε αυτά.

Όπως ήδη αναλύσαμε στο Κεφάλαιο 4, το συνελικτικό νευρωνικό δίκτυο *Inception V3*, δέχεται εικόνες μεγέθους *299x299x3*. Έτσι, η πρώτη επεξεργασία που χρειάστηκε να κάνουμε είναι να μετασχηματίσουμε όλες τις εικόνες έτσι ώστε να αποκτήσουν διαστάσεις *299x299x3*, ώστε να βρίσκονται σε συμβατό μέγεθος με την είσοδο του δικτύου *Inception V3*.

Το άλλο σκέλος επεξεργασίας που εφαρμόσαμε, αφορούσε τον τρόπο με τον οποίον θα προωθούνται τα δεδομένα στον υπολογιστικό μας γράφο κατά την διαδικασία της εκπαίδευσης.

Τα δυαδικά αρχεία (*binary files*), είναι συνήθως πιο εύκολα στην διαχείρισή τους, καθώς δεν χρειάζονται προσδιορισμό των διαφορετικών αρχείων εκείνων που περιέχουν τις εικόνες και εκείνων που περιέχουν τις περιγραφές. Με την αποθήκευση των δεδομένων σε δυαδική μορφή, ουσιαστικά, τα αποθηκεύουμε σε ένα κομμάτι μνήμης, ενώ γίνεται αρκετά αποδοτικότερο το διάβασμά τους.

Για τους λόγους αυτούς, επιλέξαμε να αναπαραστήσουμε τα δεδομένα μας (εικόνες και περιγραφές) σε μια δυαδική μορφή, η οποία είναι απόλυτα συμβατή με την βιβλιοθήκη *TensorFlow*, την ***TFRecord*** *(TensorFlow Record)*.

Πρόκειται, ουσιαστικά, για μια δυαδική μορφή αρχείου, η οποία ενσωματώνει τις εικόνες και τις περιγραφές τους σε ένα αρχείο, χωρίς να χρειάζεται να διαβάζουμε χωριστά σαν είσοδο την εικόνα και την περιγραφή της.

Στην υλοποίησή μας, δημιουργήσαμε *256* κατανεμημένα *TFRecord* αρχεία για την εκπαίδευση του μοντέλου μας, κάθε ένα από τα οποία περιείχε περίπου *2300* εγγραφές.

Κάθε εγγραφή μέσα σε ένα αρχείο *TFRecord* , είναι ένα σειριοποιημένο ***SequenceExample Proto*** αρχείο, το οποίο αποτελείται αποκλειστικά από ακριβώς ένα ζευγάρι εικόνας-περιγραφής. Αξιοσημείωτο είναι το γεγονός ότι, επειδή κάθε εικόνα στο αρχικό μας σύνολο δεδομένων έχει πολλές περιγραφές (συνήθως 5), κάθε εικόνα αντιγράφεται αρκετές φορές στα αρχεία *TFRecord*. Κάθε *SequenceExample Proto* αρχείο περιέχει τα παρακάτω πεδία:

context:

image/image_id: *integer MSCOCO image identifier*

image/data: *string containing JPEG encoded image in RGB colorspace*

feature_lists:

image/caption: *list of strings containing the (tokenized) caption words*

image/caption_ids: *list of integer ids corresponding to the caption words*

5.2 Εκπαίδευση Μοντέλου

Στην ενότητα αυτή θα ασχοληθούμε με την διαδικασία της εκπαίδευσης του συστήματός μας, αναλύοντας τον ακριβή αλγόριθμο που χρησιμοποιήθηκε προκειμένου να εκπαιδευτεί το σύστημά μας, αλλά και την επιλογή των υπερπαραμέτρων που κάναμε, ώστε να βελτιστοποιήσουμε την διαδικασία αυτή.

5.2.1 Εκπαιδευόμενες Μεταβλητές

Έχοντας περιγράψει όλα τα συστατικά κομμάτια του συστήματος το οποίο υλοποιήσαμε, ήρθε η στιγμή να δούμε ποια από τα κομμάτια αυτά αποτέλεσαν μέρος της συνολικής εκπαίδευσης του συστήματος.

Οι προκλήσεις σχεδιασμού και επιλογής των μεταβλητών εκπαίδευσης που αντιμετωπίσαμε κατά την διάρκεια δημιουργίας του συστήματός μας ήταν αρκετές και είχαν να κάνουν κυρίως με το πρόβλημα της υπερπροσαρμογής.

Παρ' όλα αυτά, εφαρμόσαμε τεχνικές οι οποίες ήταν ικανές να αντιμετωπίσουν την υπερπροσαρμογή.

Αρχικά, ο πιο προφανής τρόπος να αντιμετωπίσουμε το πρόβλημα αυτό, ήταν να αρχικοποιήσουμε τα βάρη του Συνελικτικού Νευρωνικού Δικτύου μας *Inception V3*, με τα βάρη ενός ήδη εκπαιδευμένου μοντέλου σε αρκετά μεγάλα σύνολα δεδομένων, όπως είναι το *ImageNet,* το οποίο είναι ικανό να αναγνωρίζει και να κατηγοριοποιεί αντικείμενα που βρίσκονται μέσα σε εικόνες. Το εφαρμόσαμε σε όλα μας τα παραδείγματα και βοήθησε αρκετά, κυρίως στον τομέα της *γενίκευσης*. Αυτό σημαίνει, ότι τα βάρη του *Inception V3* αρχικοποιήθηκαν με ήδη εκπαιδευμένα βάρη και δεν μεταβλήθηκαν καθόλου κατά την διαδικασία πρώτου σταδίου της εκπαίδευσης.

Άλλο ένα σύνολο βαρών τα οποία θα μπορούσαν να αρχικοποιηθούν με κάποια ήδη εκπαιδευμένα βάρη και να μην συμμετέχουν στη διαδικασία της εκπαίδευσης, ήταν εκείνα του πίνακα εμφύτευσης των διανυσμάτων των λέξεων (*word embedding vectors*). Υπάρχουν αρκετά, ήδη εκπαιδευμένα, μοντέλα τα οποία είναι σε θέση να αναπαριστούν εκατομμύρια λέξεις με σταθερού μήκους διανύσματα, όπως είναι το *word2vec*, ωστόσο παρατηρήθηκε ότι η χρήση ενός τέτοιου μοντέλου δεν εμφάνιζε κάποια βελτίωση και, έτσι, για λόγους απλότητας, προτιμήθηκε να μην συμπεριληφθεί στο σύστημά μας, αλλά να εκπαιδεύσουμε τα βάρη του πίνακα αυτού κανονικά. Έτσι, όπως γίνεται αντιληπτό, τα βάρη του πίνακα εμφύτευσης των λέξεων σε διανύσματα σταθερού μήκους αποτέλεσαν μεταβλητές οι οποίες συμμετείχαν στην διαδικασία της εκπαίδευσης.

Τέλος, χωρίς να υπάρχει διαφορετική επιλογή, τα βάρη των δύο πλήρως συνδεδεμένων επιπέδων επεξεργασίας, τα οποία βρίσκονται στην έξοδο του Συνελικτικού Νευρωνικού Δικτύου και στην είσοδο του κατηγοριοποιητή *Softmax* μετά το *LSTM*, αποτέλεσαν μεταβλητές οι οποίες χρειάστηκαν εκπαίδευση, όπως έγινε, φυσικά, και με τα βάρη του *LSTM* δικτύου. Συνοψίζοντας, τα βάρη τα οποία εκπαιδεύτηκαν στο πρώτο στάδιο της εκπαίδευσης ήταν τα εξής:

- Τα **βάρη του πλήρως συνδεδεμένου επιπέδου**, το οποίο βρίσκεται στην έξοδο του *CNN* δικτύου και αντιστοιχίζει τις διαστάσεις του πίνακα που περιέχει χαρακτηριστικά της εικόνας από 2048 σε 512

- Τα **βάρη του πίνακα εμφύτευσης** των λεξεων σε διανύσματα σταθερού μήκους, ο οποίος μετατρέπει τις λέξεις σε διανύσματα πραγματικών τιμών
- Τα **βάρη του δικτύου *LSTM*** το οποίο είναι υπεύθυνο για την δημιουργία προτάσεων περιγραφών των εικόνων
- Τα **βάρη του πλήρως συνδεδεμένου επιπέδου**, το οποίο βρίσκεται στην είσοδο του κατηγοριοποιητή *Softmax* μετά το *LSTM*, το οποίο αντιστοιχίζει τις 512 διαστάσεις εξόδου του *LSTM*, στις 11.519 διαστάσεις, που είναι το μέγεθος του λεξικού μας

Αξίζει να σημειωθεί ότι το πρώτο στάδιο της εκπαίδευσης διήρκησε ένα εκατομμύριο βήματα, κάθε βήμα από τα οποία αφορούσε την εκπαίδευση του συστήματος σε μια ομάδα εικόνων μεγέθους *32*.

Με την ολοκλήρωση του πρώτου σταδίου της εκπαίδευσης, αποφασίσαμε να βελτιώσουμε την απόδοση του συστήματός μας, με το να τρέξουμε την εκπαίδευση για ακόμη ***435460*** βήματα, αυτήν την φορά εκπαιδεύοντας όμως και τις μεταβλητές του ***Inception V3***.

5.2.2 Αλγόριθμος Εκπαίδευσης

Ο γενικός σκοπός της εκπαίδευσης του συστήματός μας είναι να εκπαιδευθεί το *LSTM* δίκτυό μας έτσι ώστε να μπορεί να προβλέπει κάθε λέξη μιας πρότασης περιγραφής μιας εικόνας, αφού πρώτα έχει δει την εικόνα καθώς και όλες τις προηγούμενες λέξεις οι οποίες αναπαριστώνται με διανύσματα εμφύτευσης, όπως ήδη έχουμε δει στο προηγούμενο κεφάλαιο.

Για τον λόγο αυτό, είναι βολικό να σκεφτούμε το δίκτυο *LSTM* σε μια ξεδιπλωμένη μορφή σαν και αυτήν του Σχήματος 4.2., στην οποία δημιουργείται ένα αντίγραφο της μνήμης του *LSTM* για την εικόνα και ένα για κάθε λέξη της πρότασης, έτσι ώστε όλα τα *LTSM's* να μοιράζονται τις ίδιες παραμέτρους και η έξοδος m_{t-1} του *LSTM* τη χρονική στιγμή *t-1* να προωθείται στο *LSTM* τη χρονική στιγμή *t*. Στην ξεδιπλωμένη μορφή του *LSTM* όλες οι ανατροφοδοτούμενες συνδέσεις μετασχηματίζονται σε προωθητικές προς τα εμπρός συνδέσεις. Αναλυτικότερα, αν ορίσουμε με *I* την εικόνα εισόδου και με $S = (S_0, ..., S_N)$ μια πραγματική πρόταση περιγραφής της εικόνας, τότε η "ξεδιπλωμένη" διαδικασία διαβάζει:

$$x_{-1} = CNN(I)$$
$$x_t = W_e S_t, t\epsilon \{0...N-1\}$$
$$p_{t+1} = LSTM(x_t), t\epsilon \{0...N-1\}$$

, όπου με S_t αναπαριστούμε την κάθε λέξη σαν ακέραιο. Επίσης, όπως έχουμε ήδη αναφέρει, με S_0 ορίζουμε μια ειδική λέξη που υποδηλώνει την αρχή μιας πρότασης, ενώ με S_N, , ορίζουμε μια ειδική λέξη που υποδηλώνει το τέλος μιας πρότασης περιγραφής. Η εικόνα και οι λέξεις απεικονίζονται στον ίδιο χώρο διαστάσεων με την βοήθεια του *CNN* και του πλήρως συνδεδεμένου επιπέδου για την εικόνα, και με τις εμφυτεύσεις λέξεων σε διανύσματα W_e για τις λέξεις.

Αξίζει να σημειώσουμε ότι η εικόνα τροφοδοτείται μόνο μία φορά στο *LSTM,* την στιγμή της εκκίνησης του δικτύου μάλιστα, προκειμένου να πληροφορήσει το *LSTM* για το περιεχόμενο της εικόνας. Το να τροφοδοτούμε την εικόνα σε κάθε χρονικό βήμα λειτουργίας του *LSTM*, συνεισφέρει μόνο στην εκμετάλλευση του θορύβου και ως εκ τούτου στην ενίσχυση του προβλήματος της υπερπροσαρμογής.

Η συνάρτηση απωλειών που χρησιμοποιούμε είναι το άθροισμα των αρνητικών λογαριθμικών ενδεχομένων της σωστής λέξης σε κάθε χρονικό βήμα και ορίζεται ως εξής:

$$L(I,S) = -\sum_{t=1}^{N} logp_t(S_t)$$

Η παραπάνω συνάρτηση απωλειών ελαχιστοποείται σε σχέση με όλες τις παραμέτρους που χρειάζονται εκπαίδευση και που αναλύσαμε στην προηγούμενη ενότητα.

Ας δούμε, όμως, αναλυτικότερα και διαισθητικότερα τα βήματα τα οποία εκτελούνται κατά την διαδικασία της εκπαίδευσης.

Αλγόριθμος 1 Εκπαίδευση του συνολικού μοντέλου

1. Δεδομένης μιας εικόνας εισόδου *I*, πέρασέ την από το δίκτυο ***CNN***
2. Προώθησε την έξοδο του *CNN* στο ***πλήρως συνδεδεμένο επίπεδο***, προκειμένου να μετασχηματίσει τις διαστάσεις της εικόνας
3. Προώθησε την έξοδο του πλήρως συνδεδεμένου επιπέδου στην ***είσοδο του LSTM***
4. Βρες τα ***διανύσματα εμφύτευσης σταθερού μήκους της κάθε λέξης*** της πραγματικής πρότασης περιγραφής της εικόνας μέσω του ***πίνακα εμφύτευσης*** και προώθησέ τα στην ***είσοδο του LSTM***
5. Μέχρι να παραχθεί μια ολοκληρωμένη πρόταση περιγραφής της εικόνας I, επανάλαβε:
    a. Αν *t=0*:

i. Δώσε σαν είσοδο στο *LSTM* την έξοδο του πλήρως συνδεδεμένου επιπέδου η οποία περιέχει χαρακτηριστικά της εικόνας

b. Διαφορετικά:

i. Δώσε σαν είσοδο στο *LSTM* το διάνυσμα εμφύτευσης της πραγματικής λέξης του χρονικού βήματος *t-1*

c. Τρέξε ένα χρονικό βήμα *LSTM* και κάνε μια πρόβλεψη για την επόμενη λέξη

d. Υπολόγισε την τιμή της συνάρτησης απωλειών ***L(I,S)*** μεταξύ της πιθανοτικής που προβλέφθηκε από το δίκτυο *LSTM* και της πραγματικής λέξης

e. Μέσω της τεχνικής της ***οπισθοδιάδοσης σφάλματος (error backpropagation)*** και με την βοήθεια του ***βελτιστοποιητή SGD (Stohastic Gradient Descent)***, ανανέωσε τις τιμές των εκπαιδευόμενων μεταβλητών, έτσι ώστε να κάνουν μια πιο σωστή πρόβλεψη την επόμενη φορά

5.2.3 Επιλογή Υπερπαραμέτρων Εκπαίδευσης

Προκειμένου το σύστημά μας να παράγει όσο το δυνατόν καλύτερα αποτελέσματα, θα έπρεπε να εκπαιδευθεί όσο το δυνατόν καλύτερα. Για να συμβεί αυτό, σημαντικό ρόλο στην εκπαίδευση του συστήματος παίζει η σωστή επιλογή των υπερπαραμέτρων. Για να καταλήξουμε στις συγκεκριμένες επιλογές που θα παρουσιάσουμε στην συνέχεια, χρειάστηκε να δοκιμάσουμε διάφορες τιμές, να αναζητήσουμε τιμές που έχουν ήδη χρησιμοποιηθεί σε αντίστοιχα συστήματα και έχουν παρουσιάσει ικανοποιητικά αποτελέσματα και στην συνέχεια να κάνουμε τις δικές μας επιλογές. Έτσι, οι τιμές των υπερπαραμέτρων στις οποίες καταλήξαμε είναι οι εξής:

- Ο αριθμός εικόνων οι οποίες θα επεξεργάζονται ταυτόχρονα από το σύστημά μας (***batch size***) να είναι ίσος με **32**.
- Σαν βελτιστοποιητή της εκπαίδευσής μας, επιλέξαμε τον ***SGD***.
- Όπως ήδη αναφέραμε, ο αριθμός των μονάδων του LSTM δικτύου, όπως επίσης και οι διαστάσεις εξόδου του πίνακα εμφύτευσης ήταν ίσες με ***512***.
- Ο ρυθμός εκμάθησης (***learning rate***) του συστήματος μας στο πρώτο στάδιο ξεκινούσε από την τιμή ***2*** και σταδιακά έφτασε σχεδόν σε μηδενική τιμή χρησιμοποιώντας μια φθίνουσα συνάρτηση.

- Ο ρυθμός εκμάθησης ***(learning rate)*** των μεταβλητών του ***Inception V3*** είχε την τιμή ***0.0005***.
- Δεδομένου ότι το σύνολο δεδομένων εκπαίδευσής μας αποτελούταν από 586363 εικόνες με τις περιγραφές τους, οι ***εποχές (epochs)*** στις οποίες εκπαιδεύθηκε το σύστημά μας ήταν περίπου ***80***.

5.3 Έλεγχος Λειτουργίας (Inference)

Υπάρχουν αρκετές τεχνικές οι οποίες μπορούν να χρησιμοποιηθούν για να δημιουργηθούν προτάσεις περιγραφής δεδομένης μιας εικόνας εισόδου, χρησιμοποιώντας το μοντέλο μας. Η πρώτη τεχνική είναι αυτή της ***Δειγματοληψίας (Sampling)***, όπου απλώς δειγματοληπτούμε την πρώτη λέξη σύμφωνα με την πιθανότητα p_1 , στην συνέχεια παρέχουμε το διάνυσμα εμφύτευσης της λέξης που αντιστοιχεί στην πιθανότητα p_1 και συνεχίζουμε με αυτόν τον τρόπο έως ότου συναντήσουμε τον χαρακτήρα που υποδηλώνει το τέλος της πρότασης ή η πρόταση φτάσει ένα μέγιστο μήκος, το οποίο έχουμε ορίσει εμείς.

Η δεύτερη τεχνική ονομάζεται ***Ακτινική Αναζήτηση (Beam Search)***. Στην μέθοδο αυτήν, επαναληπτικά θεωρούμε το σύνολο των *k* καλύτερων προτάσεων μέχρι την χρονική στιγμή *t* ως υποψήφιες για να δημιουργήσουμε προτάσεις μεγέθους *t+1* και από αυτές κρατάμε μόνον τις *k* καλύτερες. Αυτό αποτυπώνεται καλύτερα ως : $S = \arg\max_{S'} p(S'|I)$.

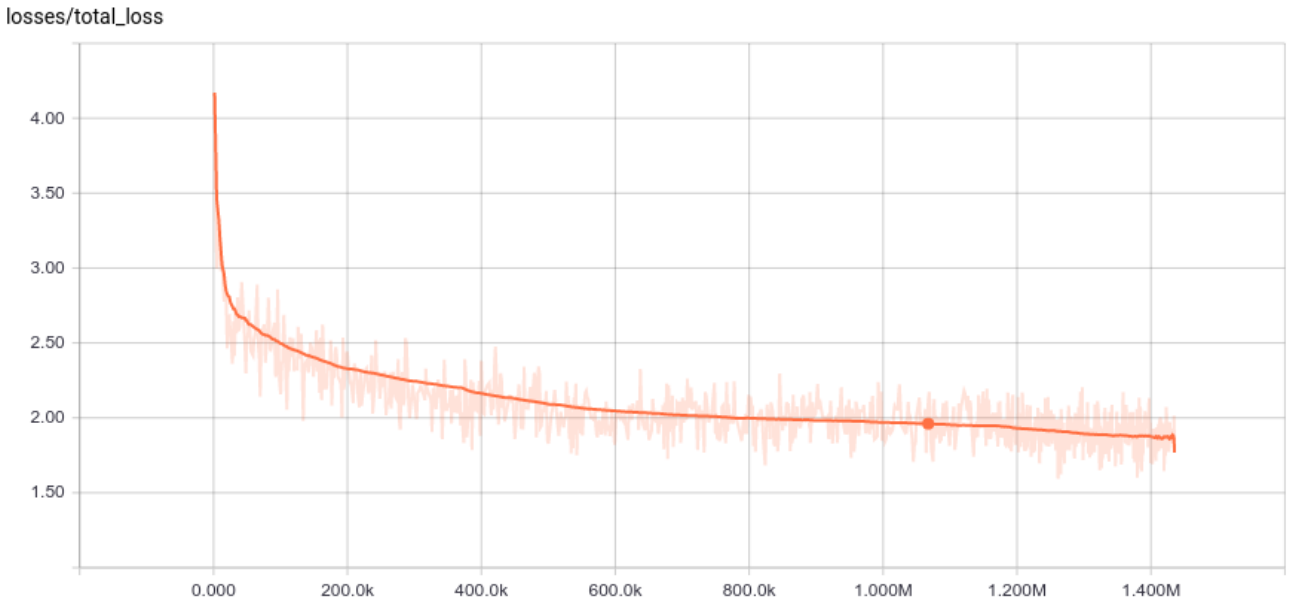
Στην υλοποίησή μας χρησιμοποιήσαμε την μέθοδο της *Δειγματοληψίας*..

5.4 Αποτελέσματα-Μετρήσεις

Η εκπαίδευση του συστήματος μας διήρκησε 7 ημέρες, με τις 4 πρώτες να αφιερώνονται αποκλειστικά στο πρώτο στάδιο της εκπαίδευσης ενώ η διαδικασία fine-tuning των μεταβλητών του Inception V3 διήρκησε άλλες 3 ημέρες. Η εκπαίδευση έγινε πάνω στην κάρτα γραφικών ***Nvidia GeForce GTX 1060 6GB***.

Τα αποτελέσματα που παρουσίασε το σύστημά μας ήταν αρκετά ικανοποιητικά, με τις περιγραφές των εικόνων που δημιουργούσε να βρίσκονται πολύ κοντά στις πραγματικές. Στη συνέχεια παρουσιάζονται γραφήματα με τις τιμές που πήραν διάφορες μεταβλητές κατά τη διάρκεια της εκπαίδευσης καθώς και την γραφική παράσταση των συνολικών απωλειών μεταξύ της περιγραφής που γεννούσε το σύστημά μας και της πραγματικής περιγραφής.

- ***Μεταβλητή Συνολικών Απωλειών (Total Loss)***



Πίνακα 5.1 Γραφική παράσταση των συνολικών απωλειών κατά την διάρκεια της εκπαίδευσης

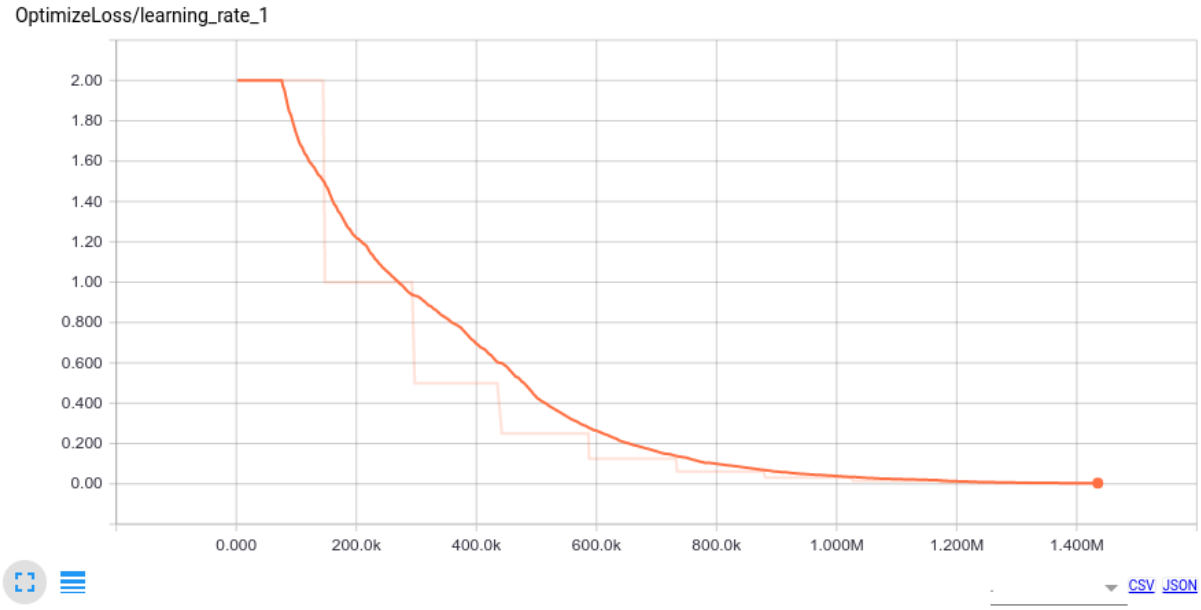
Στην παραπάνω γραφική παράσταση φαίνονται οι τιμές που παίρνει η μεταβλητή των συνολικών απωλειών κατά την διάρκεια της εκπαίδευσης. Όπως είναι φυσιολογικό, όταν ξεκινάει η εκπαίδευση η μεταβλητή αυτή παίρνει πολύ μεγάλες τιμές, καθώς το σύστημά μας δεν έχει εκπαιδευθεί καθόλου και έτσι οι περιγραφές που δημιουργεί απέχουν αρκετά από τις πραγματικές.

Καθώς αυξάνονται τα βήματα της εκπαίδευσης, όπως μπορούμε να δούμε, η τιμή της μεταβλητής αυτής παρουσιάζει μια μεγάλη βελτίωση μέχρι τα 600 χιλιάδες βήματα όπου για περίπου 300 χιλιάδες βήματα ακόμη, παρουσιάζει μια σταθερή αλλά καθοδική τάση.

Στα 1.100.000 βήματα βλέπουμε ότι η καθοδική αυτή τάση αρχίζει να παίρνει μια πιο απότομη κλίση. Ο λόγος που συμβαίνει αυτό, είναι διότι σε αυτό το σημείο ξεκινάει το δεύτερο στάδιο της εκπαίδευσης στο οποίο εκπαιδεύονται και οι μεταβλητές του *Inception V3* μαζί με τις υπόλοιπες μεταβλητές.

Η ελάχιστη τιμή στην οποία φτάνει η μεταβλητή αυτή είναι περίπου στο 1.7.

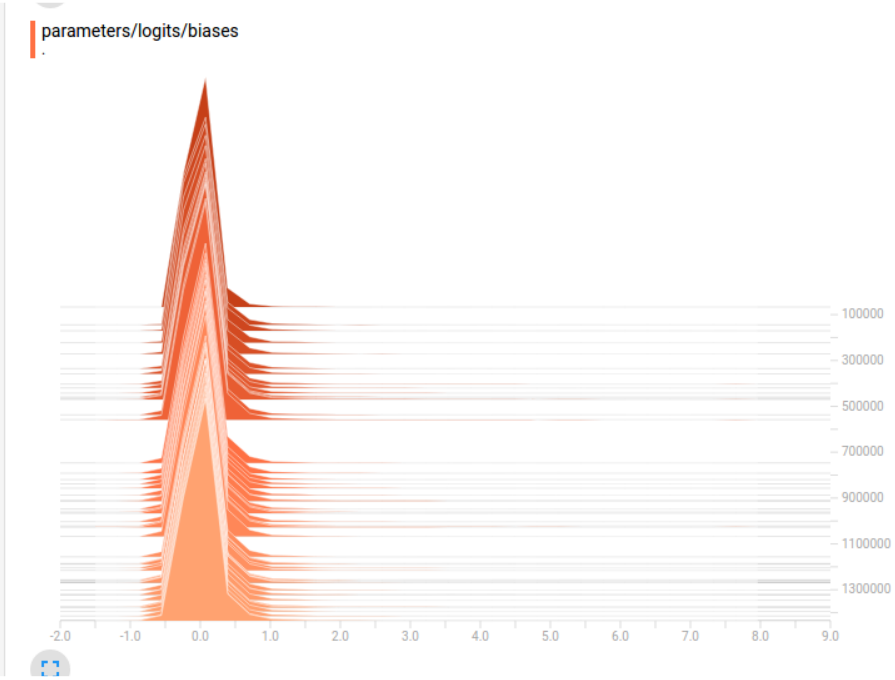
- ***Ρυθμός Εκμάθησης (Learning Rate)***



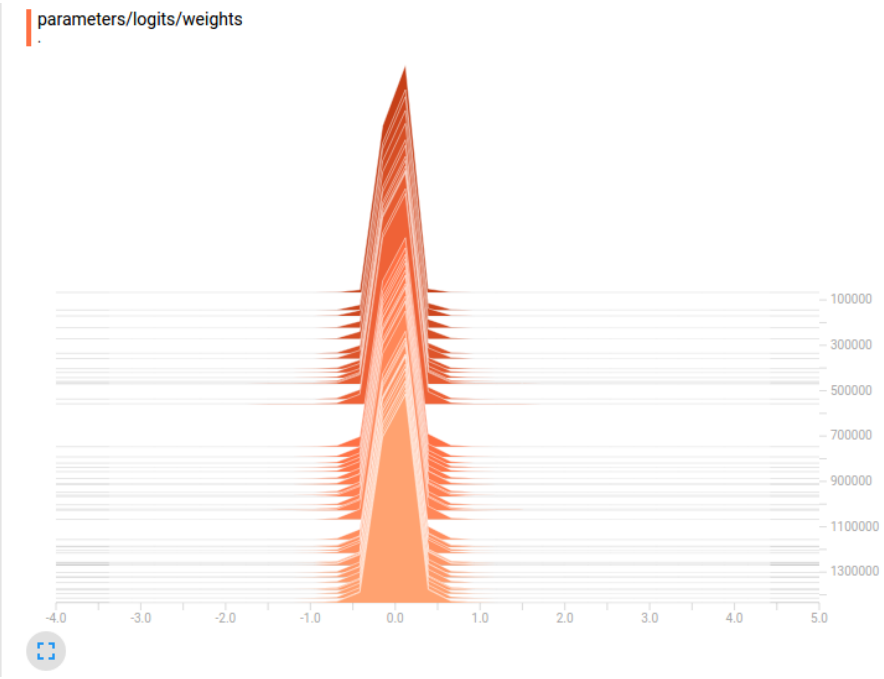
Πίνακας 5.2 Γραφική παράσταση της μεταβλητής που αφορά τον ρυθμό εκμάθησης

Όπως ήδη έχουμε αναφέρει, η μεταβλητή αυτή ξεκινά με την τιμή 2, ενώ σταδιακά μειώνεται μέχρι να πάρει τιμές που πλησιάζουν αρκετά στο μηδέν.

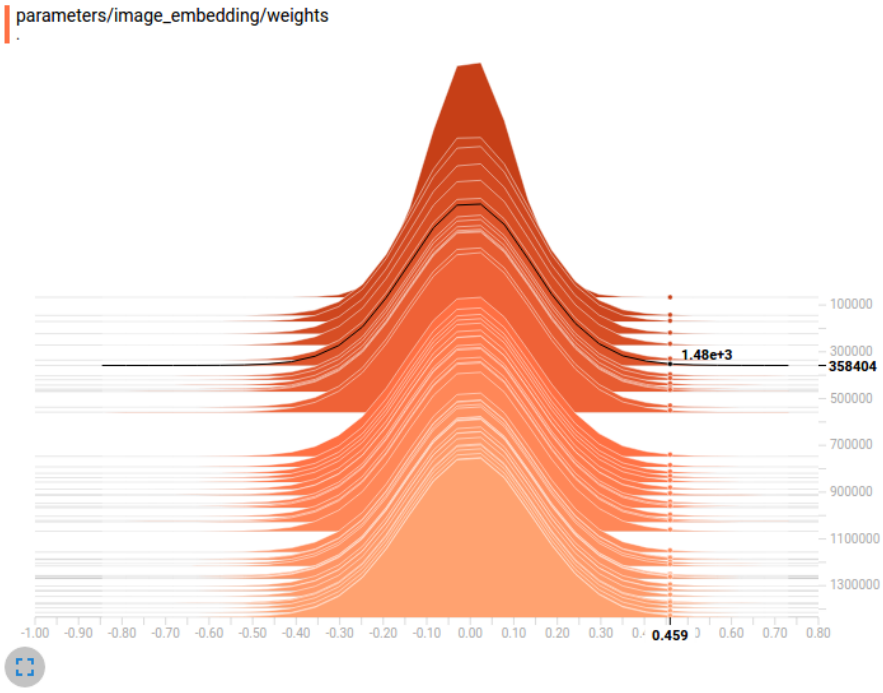
Στη συνέχεια, παρουσιάζουμε ιστογράμματα από τις τιμές που πήραν οι εκπαιδευόμενες μεταβλητές του πρώτου σταδίου.



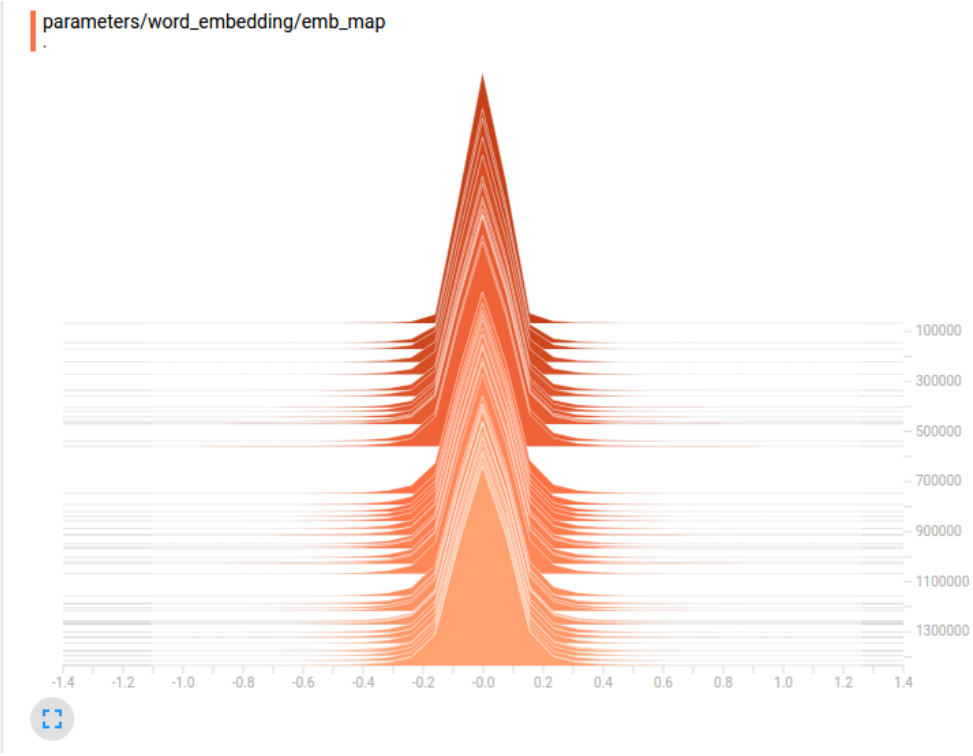
Πίνακας 5.3 Κατανομή τιμών των biases του τελευταίου επιπέδου του συστήματος



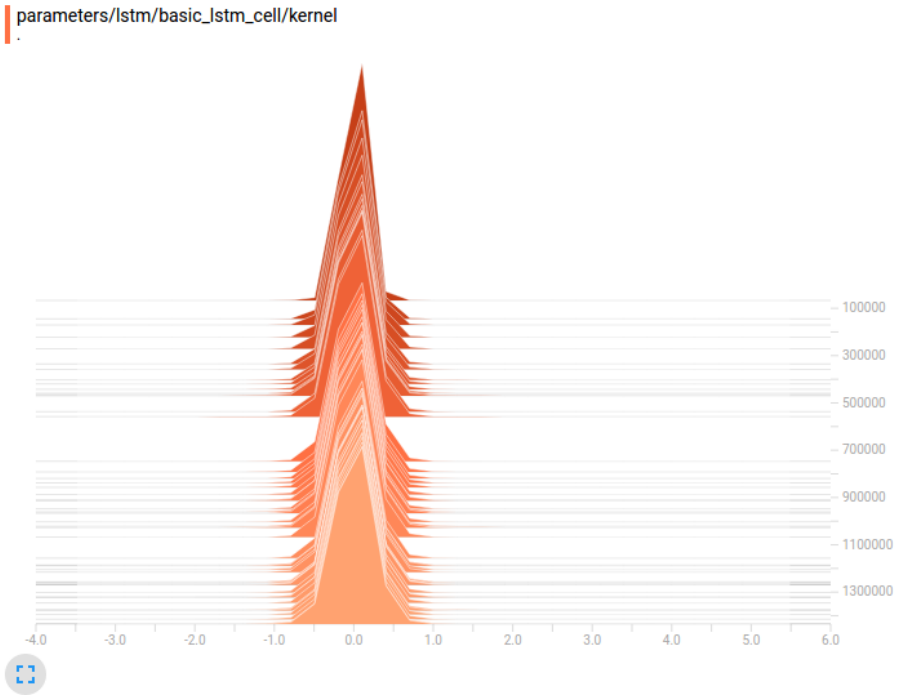
Πίνακας 5.4 Κατανομή των βαρών του τελευταίου επιπέδου του συστήματος



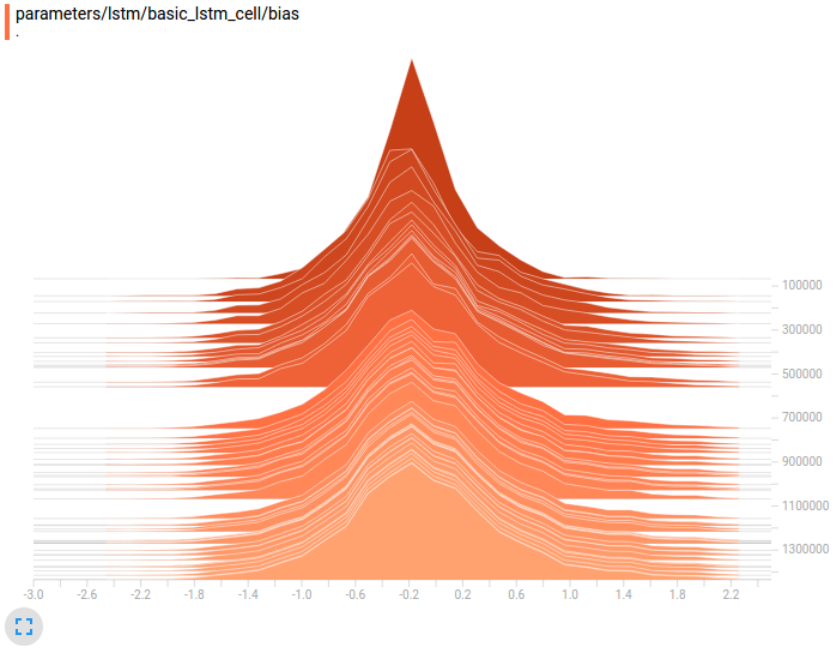
Πίνακας 5.5 Κατανομή των βαρών των εμφυτεύσεων των λέξεων



Πίνακας 5.6 Κατανομή των τιμών των διανυσμάτων εμφύτευσης των λέξεων



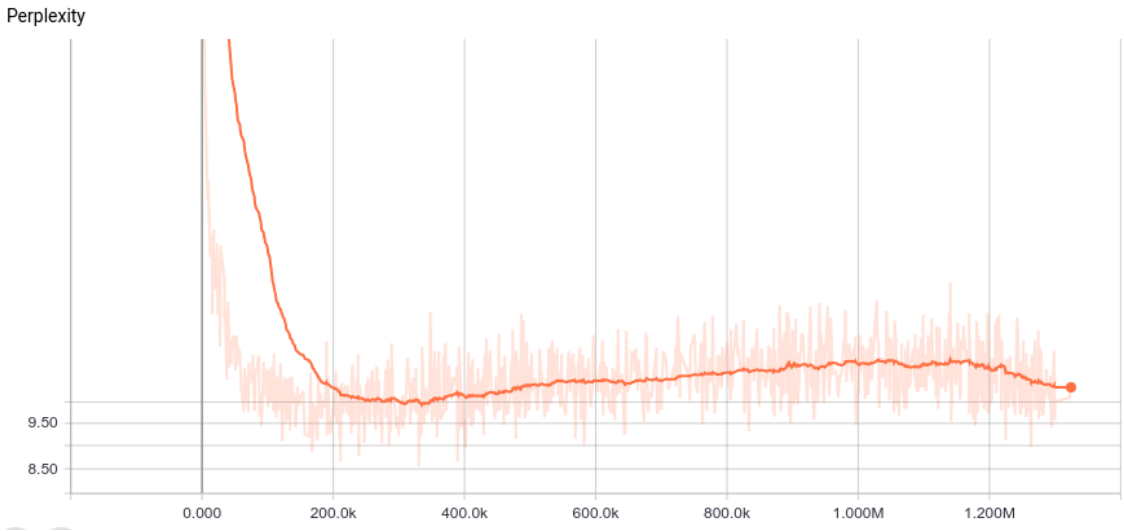
Πίνακας 5.7 Κατανομή των βαρών του δικτύου LSTM



Πίνακας 5.8 Κατανομή των biases του δικτύου LSTM

5.5 Αξιολόγηση

Για την αξιολόγηση του συστήματός μας, σημαντικό ρόλο παίζουν οι δύο γραφικές παραστάσεις που παρουσιάζουμε στην συνέχεια.



Πίνακας 5.9 Γραφική παράσταση που δείχνει την σύγχυση μεταξύ των πραγματικών περιγραφών και των περιγραφών που δημιουργεί το σύστημά μας.

Η παραπάνω γραφική παράσταση παρουσιάζει την τιμή μιας σημαντικής μεταβλητής που ονομάζεται ***σύγχυση (perplexity)*** και ορίζει την ανικανότητα του συστήματος να παρουσιάζει σωστές περιγραφές. Είναι η βασική ποσότητα, με βάση την οποία μπορούμε να αξιολογήσουμε το σύστημά μας και, όπως είναι φυσικό, όσο μικρότερη η τιμή της τόσο το καλύτερο.

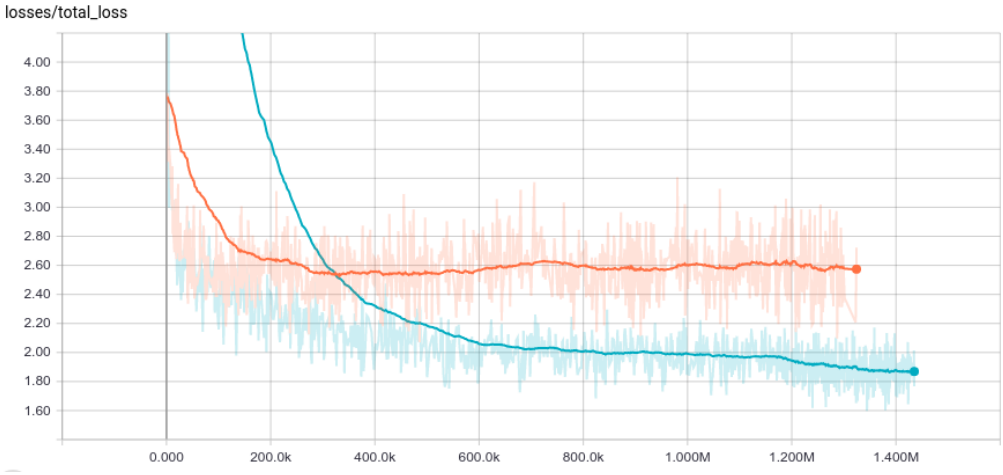
Πρόκειται, επί της ουσίας, για μια μετρική η οποία μας δείχνει το πόσο καλά μια κατανομή πιθανοτήτων ή ένα πιθανοτικό μοντέλο κάνει καλές προβλέψεις σε ένα δείγμα, ενώ συνήθως είναι η βασική μετρική η οποία χρησιμοποιείται για την σύγκριση συστημάτων. Μικρές τιμές της σύγχυσης αυτής υποδεικνύουν ότι το μοντέλο κάνει καλές προβλέψεις.

Μαθηματικά, ορίζεται ως ο γεωμετρικός μέσος όρος με βάρη των αντιστρόφων των πιθανοτήτων και ο μαθηματικός της τύπος είναι:

$$exp\sum_{x} p(x)log_e 1/p(x)$$

Παρατηρούμε ότι η τιμής της ποσότητας αυτής καταλήγει να είναι περίπου στο δέκα, που αποτελεί μια αρκετά ικανοποιητική τιμή.

Μια άλλη γραφική παράσταση που παρουσιάζει αρκετό ενδιαφέρον είναι αυτή που φαίνεται παρακάτω και δείχνει πως κυμαίνονται οι απώλειες εκπαίδευσης και οι απώλειες αξιολόγησης σε ένα γράφημα.



Πίνακας 5-10 Γραφική παράσταση των απωλειών εκπαίδευσης σε σχέση με τις απώλειες αξιολόγησης

Με το μπλε χρώμα φαίνονται οι απώλειες εκπαίδευσης και με το πορτοκαλί χρώμα οι απώλειες αξιολόγησης.

Το ενδιαφέρον στοιχείο σε αυτήν την γραφική παράσταση είναι ότι ενώ η γραφική παράσταση των απωλειών εκπαίδευσης παρουσιάζουν μια καθοδική τάση καθ' όλη την διάρκεια της εκπαίδευσης, δεν φαίνεται να ισχύει το ίδιο και για τις απώλειες αξιολόγησης.

Αντιθέτως, φαίνεται ότι μετά τα 300 χιλιάδες βήματα, οι απώλειες αξιολόγησης παρουσιάζουν μια σταθερή πορεία. Αυτό, βέβαια, φαίνεται και από την γραφική παράσταση της σύγχυσης, όπου από ένα σημείο και μετά δεν εμφάνιζε κάποια βελτίωση.

Αντιθέτως, όπως είναι φυσιολογικό, οι απώλειες εκπαίδευσης πέφτουν συνεχώς και θα συνεχίσουν να πέφτουν μέχρι να τελειώσει η εκπαίδευση, διατρέχοντας όμως μεγάλο κίνδυνο να εμφανίσουν το πρόβλημα της υπερπροσαρμογής.

Στη συνέχεια, παρουσιάζουμε τρία χαρακτηριστικά παραδείγματα λειτουργίας του συστήματός μας, όπου για την εικόνες εισόδου που φαίνονται παρακάτω, δημιουργήθηκαν αρκετά ακριβείς περιγραφές.

1) a baseball player swinging a bat at a ball (p=0.004845)

Εικόνα 5.11 Παράδειγμα λειτουργίας του συστήματός μας

1) a man riding a snowboard down a snow covered slope (p=0.012535)

Εικόνα 5-12 Παράδειγμα λειτουργίας του συστήματός μας

1) **a white plate topped with meat and vegetables (p=0.002463)**

Εικόνα 5-13 Παράδειγμα Λειτουργίας τους Συστήματός μας

Ωστόσο, όπως ήταν εξάλλου αναμενόμενο, το σύστημά μας δεν είναι τέλειο και κάποιες εικόνες ενδέχεται να το ξεγελάσουν, όπως αυτή που φαίνεται στην συνέχεια.

1) **a person riding a surf board on a body of water (p=0.000622)**

Εικόνα 5-14 Παράδειγμα λανθασμένης πρόβλεψης του συστήματός μας

Ενώ στην πραγματικότητα βλέπουμε μια δορυφορική λήψη ενός κομματιού της γης, η περιγραφή η οποία δημιουργείται δεν βγάζει κάποιο νόημα. Ο βασικός λόγος που συμβαίνει αυτό είναι διότι δεν υπήρχαν αρκετές αντίστοιχες εικόνες στο σύνολο δεδομένων μας και έτσι το σύστημα ίσως αντιμετωπίζει για πρώτη φορά μια δορυφορική λήψη. Παρατηρούμε, όμως, ότι παρά την κακή περιγραφή, το σύστημά μας καταφέρνει να εντοπίσει την ύπαρξη μεγάλης ποσότητας νερού.

6

Επίλογος

6.1 Σύνοψη και Συμπεράσματα

Στην διπλωματική εργασία αυτήν, παρουσιάσαμε ένα σύστημα νευρωνικών δικτύων το οποίο μπορεί, αυτόματα, να βλέπει μια εικόνα και να δημιουργεί μια λογική περιγραφή της στην Αγγλική γλώσσα. Το σύστημά μας βασίζεται σε ένα συνελικτικό νευρωνικό δίκτυο το οποίο κωδικοποιεί την εικόνα εισόδου σε μια αναπαράσταση ενός σταθερού-μήκους διάνυσμα, το οποίο ακολουθείται από ένα ανατροφοδοτούμενο νευρωνικό δίκτυο, υπεύθυνο για την δημιουργία της αντίστοιχης πρότασης περιγραφής της εικόνας. Το μοντέλο εκπαιδεύτηκε έτσι ώστε να μεγιστοποιεί την πιθανότητα να παράγει μια σωστή πρόταση περιγραφής δεδομένης μιας εικόνας εισόδου. Πειράματα σε αρκετά μεγάλα σύνολα δεδομένων απέδειξαν τη σταθερότητα του συστήματός μας ποιοτικά και ποσοτικά.

6.2 Μελλοντικές επεκτάσεις

Παρά το γεγονός ότι δημιουργήσαμε ένα αρκετά σταθερό και αποδοτικό σύστημα, το οποίο παρουσιάζει ικανοποιητικά αποτελέσματα, υπάρχουν ακόμα αρκετές κατευθύνσεις κατά τις οποίες το σύστημά μας θα μπορούσε να βελτιωθεί περαιτέρω.

- ***Επιπλέον Εναρμόνιση Μοντέλου Εικόνας (Image Model Fine Tuning)*** : Μια μελλοντική επέκταση της διπλωματικής αυτής, θα μπορούσε να περιλαμβάνει κάποια ακόμα βήματα εκπαίδευσης του συστήματος, συμπεριλαμβάνοντας ωστόσο τις παραμέτρους του συνελικτικού νευρωνικού δικτύου στην εκπαίδευσης, έτσι ώστε να εναρμονιστούν με το υπόλοιπο σύστημα. Αναμένουμε αυτή η αλλαγή στο σύστημα να επιφέρει βελτίωση των αποτελεσμάτων.
- ***Προγραμματισμένη Δειγματοληψία (Scheduled Sampling)*** : Κατά την διάρκεια εκπαίδευσης του συστήματός μας, το δίκτυο *LSTM,* όπως ήδη έχουμε δει, κάνει πρόβλεψη για την επόμενη λέξη μιας πρότασης περιγραφής με βάση την προηγούμενη πραγματική λέξη της πρότασης αυτής και όχι της λέξης την οποία

προέβλεψε στο προηγούμενο χρονικό βήμα. Αυτό είναι εφικτό, καθώς στο σύνολο δεδομένων εκπαίδευσης έχουμε στην διάθεσή μας την εικόνα και την περιγραφή αυτής. Κατά τη λειτουργία του ελέγχου, ωστόσο, αυτό, όπως καταλαβαίνουμε, δεν είναι εφικτό, καθώς δεν έχουμε στην διάθεσή μας την πραγματική πρόταση. Έτσι, το μοντέλο μας κάνει πρόβλεψη για την επόμενη λέξη με βάση την λέξη την οποία προέβλεψε στο προηγούμενο χρονικό βήμα. Έτσι, παρατηρείται μια ασυνέπεια μεταξύ του τρόπου που γίνεται η εκπαίδευση και αυτού του ελέγχου της λειτουργίας του συστήματος. Για τον λόγο αυτόν, μια μέθοδος που θα μπορούσε να βελτιώσει τη απόδοση του συστήματος είναι η εκπαίδευση με Προγραμματισμένη Δειγματοληψία [2], στην οποία το σύστημα, κατά την διάρκεια της εκπαίδευσης, θα επιλέγει τυχαία, με βάση μια πιθανοτική συνάρτηση, αν θα προβλέπει την επόμενη λέξη μιας πρότασης με βάση την πραγματική προηγούμενη ή με βάση την λέξη την οποία εκείνο προέβλεψε στο προηγούμενο χρονικό βήμα, προκειμένου να υπάρχει μια συμφωνία με την πραγματική λειτουργία του μοντέλου μετά το πέρας της εκπαίδευσης.

- ***Επέκταση Χρήσης Συστήματος*** : Μια πιθανή κατεύθυνση κατά την οποία το σύστημά μας θα μπορούσε να εξελιχθεί, είναι να έχουμε ένα σύστημα το οποίο θα μπορεί να κάνει πιο στοχευμένες περιγραφές - είτε με το να αποδίδει τις περιγραφές αυτές σε συγκεκριμένες ιδιότητες και τοπία της εικόνας ή με το να μπορεί να δίνει απαντήσεις σε συγκεκριμένες ερωτήσεις χρηστών. Περαιτέρω έρευνα θα μπορούσε να γίνει ώστε το συγκεκριμένο σύστημα να βρει εφαρμογή σε επιστημονικά πεδία, όπως είναι η Ρομποτική.

7

Βιβλιογραφία

[1] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan:" Show and Tell: Lessons learned from the 2015 MSCOCO Image Captioning Challenge",2016

[2] S. Bengio, O. Vinyals, N. Jaitly, and N.Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in Advances in Neural Information Processing Systems, NIPS, 2015

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in ICLR, 2013.

[4] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, "Every picture tells a story: Generating sentences from images," in ECCV, 2010.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, "Rethinking the Inception Architecture for Computer Vision", 2015

[6] T. Lin, J. Hays, M. Maire, P. Perona, S. Belongie, D. Ramanan, L. Bourdev, C.L. Zitnick, R. Girshick, P. Dollar, "Microsoft COCO: Common Objects in Context", 2015

[7] "Using the RNN API in Tensorflow" https://medium.com/@erikhallstrm/tensorflow-rnn-api-2bb31821b185

[8] "Using the LSTM API in Tensorflow" https://medium.com/@erikhallstrm/using-the-tensorflow-lstm-api-3-7-5f2b97ca6b73

[9] "COCO-Common Objects in Context"
http://mscoco.org/dataset/#download

[10] "Tensorflow Mechanics 101"
https://www.tensorflow.org/get_started/mnist/mechanics

[11] "Tensorboard:Vizualizing Learning"
https://www.tensorflow.org/get_started/summaries_and_tensorboard

[12] "Artificiall Intelligence"
https://en.wikipedia.org/wiki/Artificial_intelligence

[13] "Machine Learning"
https://en.wikipedia.org/wiki/Machine_learning

[14] "Image Processing"
https://en.wikipedia.org/wiki/Image_processing

[15] "Convolutional Neural Networks for Visual Recognition"
http://cs231n.github.io/convolutional-networks/

[16] "CNN's"
https://cs.nju.edu.cn/wujx/paper/CNN.pdf

[17] "An Intuitive Explanation of Convolutional Neural Networks"
https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

[18] "A Quick Introduction to Neural Networks"
https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/

[19] "Regularization in Deep Learning"
https://chatbotslife.com/regularization-in-deep-learning-f649a45d6e0

[20] "NLTK Tokenize Package"
http://www.nltk.org/api/nltk.tokenize.html

[21] "Glossary of Deep Learning:Word Embedding"
https://medium.com/deeper-learning/glossary-of-deep-learning-word-embedding-f90c3cec34ca

[22] "Tensorflow"
https://www.tensorflow.org/

[23] "Reading Data - Tensorflow"
https://www.tensorflow.org/programmers_guide/reading_data

[24] "TFRecords Guide"
http://warmspringwinds.github.io/tensorflow/tf-slim/2016/12/21/tfrecords-guide/

[25] Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem, "Re-evaluating Automatic Metrics for Image Captioning", 2016