ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Εντοπισμός / Κατηγοριοποίηση tweets με ψευδές περιεχόμενο με την χρήση μεθόδων μηχανικής μάθησης και βαθιάς μάθησης

Του φοιτητή

Ασημάκη Ξάνθη

Αρ. Μητρώου: 133978

Επιβλέπων

Ονοματεπώνυμο: Κωνσταντίνος Διαμαντάρας

Βαθμίδα: Καθηγητής ΔΙΠΑΕ

Ημερομηνία: 18/1/2020

Τίτλος Δ.Ε. : Εντοπισμός / Κατηγοριοποίηση tweets με ψευδές περιεχόμενο με την χρήση μεθόδων μηχανικής μάθησης και βαθιάς μάθησης

Κωδικός Δ.Ε. …

Ονοματεπώνυμο φοιτητή: Ασημάκης Ξάνθης

Ονοματεπώνυμο εισηγητή: Κωνσταντίνος Διαμαντάρας

Ημερομηνία ανάληψης Δ.Ε. : 15/11/2018

Ημερομηνία περάτωσης Δ.Ε. : 18/1/2020

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

ΠΡΟΛΟΓΟΣ

Οι ηλεκτρονικοί υπολογιστές από την στιγμή της δημιουργίας τους έγινε κατανοητό πως θα έχουν κεντρικό ρόλο στην ανάπτυξη και πρόοδο του ανθρώπου, όμως πόσο μεγάλο κανείς δεν μπορούσε να φανταστεί. Από την σύλληψη της ιδέας του Alan Turing για τον υπολογιστή γενικής χρήσης μέχρι σήμερα πολλά έχουν αλλάξει στον τρόπο που χρησιμοποιούμε τους υπολογιστές, από ογκώδεις μηχανές για αντιμετώπιση χρονοβόρων προβλημάτων έχουν πλέον μεταφερθεί στα σπίτια μας, στις παλάμες των χεριών μας, στα αυτοκίνητα που οδηγούμε και πλέον είναι δύσκολο να φανταστεί κανείς κάποια πτυχή της ζωής μας που δεν επηρεάζεται έστω και λίγο πλέον από αυτούς.

Μέσα από την χρήση των Η/Υ ο άνθρωπος έχει καταφέρει να βελτιώσει σε μεγάλο βαθμό την καθημερινότητά του καθώς με λίγες κινήσεις των χεριών του μπορεί να επικοινωνήσει με κάποιο άτομο στην άλλη πλευρά του πλανήτη, να κάνει αγορές μέσα από ηλεκτρονικά καταστήματα, να ενημερωθεί πιο γρήγορα από ποτέ και να καταναλώσει όσο περιεχόμενο επιθυμεί. Ωστόσο όλη αυτή η χρήση έχει ως αποτέλεσμα την δημιουργία πολύ μεγάλων ποσοτήτων δεδομένων. Για παράδειγμα κάθε λεπτό γράφονται 100.000 tweets, βίντεο συνολικής διάρκειας 48 ωρών γίνονται διαθέσιμα στο Youtube, 284.000 ενημερώσεις γίνονται στο Facebook, 571 καινούργιες ιστοσελίδες δημιουργούνται.

Αυτά τα δεδομένα δεν είναι πλέον διαχειρίσιμα από ανθρώπους λόγω του μεγέθους τους οπότε παρουσιάστηκε το πρόβλημα της επεξεργασίας τους και η λύση δόθηκε από τους υπολογιστές και πάλι. Με έμπνευση από την ίδια τη φύση και τη βοήθεια της μηχανικής μάθησης καταφέραμε να μετατρέψουμε τους ηλεκτρονικούς υπολογιστές από μηχανές που κάνουν πολύπλοκες μαθηματικές πράξεις σε μικρογραφίες ενός ανθρώπινου εγκεφάλου. Έτσι όπως είναι δύσκολο να βρεθεί κάποιο κομμάτι της ζωής που δεν επηρεάζεται από τους Η/Υ, το ίδιο δύσκολο είναι να βρεθεί κάποιος τομέας που δεν έχει χρησιμοποιηθεί ακόμα η Μηχανική Μάθηση.

Περίληψη

Η αναγνώριση και ταξινόμηση κειμένου αποτελεί ένα σχετικά καινούργιο τομέα στον χώρο της μηχανικής μάθησης, αλλά και πολλά υποσχόμενο ταυτόχρονα. Κατά την διάρκεια των Αμερικάνικων εκλογών του 2016 είχε παρατηρηθεί ραγδαία αύξηση στην κοινοποίηση ψευδών ειδήσεων στην κοινωνική πλατφόρμα του Tweeter, αύξηση τόσο μεγάλη που ένα στα τέσσερα tweet που περιείχαν σύνδεσμο προς πλατφόρμα ειδήσεων είχαν ψευδές περιεχόμενο [1]. Σε ένα κοινωνικό δίκτυο όπως το Twitter η εύρεση ψευδών ειδήσεων μπορεί να αποτελέσει πολύ δύσκολη διαδικασία λόγω του ότι κάθε άνθρωπος εκφράζεται διαφορετικά.

Στην διπλωματική εργασία αυτή, εφαρμόστηκαν συνελικτικά νευρωνικά δίκτυα σε συνδυασμό με δίκτυα μεγάλης βραχυπρόθεσμης μνήμης για την διαδικασία της αναγνώρισης ψευδών και αληθών ειδήσεων από την πλατφόρμα του Tweeter βάση κειμένου. Το σύνολο δεδομένων που χρησιμοποιήθηκε προέρχεται από την εργασία των Julio Amador, Axel Oehmichen και Miguel Molina, “Characterizing Political Fake News in Tweeter by its Metadata” και περιέχει 9019 tweets, την κατηγορία στην οποία ανήκουν και πληροφορίες σχετικά με το tweet και τον χρήστη που το δημοσίευσε. Η εργασία υλοποιήθηκε σε Python και το βασικό εργαλείο που χρησιμοποιήθηκε για την εκπαίδευση του μοντέλου είναι το framework Keras.

Τα τελικά αποτελέσματα που επιτεύχθηκαν είναι: 85% ακρίβεια(accuracy) με το ποσοστό των ψευδών ειδήσεων να είναι στο 23.75%. Η ευαισθησία(sensitivity) ανέρχεται στο 0.967 ενώ η ειδικότητα(specificity) και η εγγύτητα(precision) σε 0.473 και 0.85 αντίστοιχα.

ABSTRACT

Text recognition and classification is relatively new area of research in the field of machine learning, but also very pro,mising at the same time. During the 2016 US election, there was a steep increase in sharing fake news on the social platform of Tweeter, an increase so high that one out of four tweets containing a link to a news platform had fake content [1]. But finding fake news on a social network like Tweeter can be a very difficult process because every person expresses themselves differently.

In this thesis, convolutional neural networks were implemented in combination with long-term memory networks for the process of identifying fake and true news from the Tweeter platform based on text. The dataset used comes from the work of Julio Amador, Axel Oehmichen and Miguel Molina, “Characterizing Political Fake News in Tweeter by its Metadata” and contains 9019 tweets, the category they belong to, and information about the tweet and the user who posted it. . The work was implemented in Python and the key tool used to train the model is the Keras framework.

The final results achieved are: 85% accuracy with false news rate at 23.75%. The sensitivity is 0.967 while the specificity and precision are 0.473 and 0.85 respectively.

ΕΥΧΑΡΙΣΤΙΕΣ

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ.Κωνσταντίνο Διαμαντάρα για την πολύ καλή δουλεία που κάνει στο τμήμα ως καθηγητής του μαθήματος της Μηχανκής Μάθησης, καθώς αυτό αποτέλεσε σημαντικό ρόλο στην επιλογή της συγκεκριμένης διπλωματικής, και για την καθοδήγηση και υποστήριξη στην πραγματοποίηση αυτής της εργασίας με τίτλο: Εντοπισμός / Κατηγοριοποίηση tweets με ψευδές περιεχόμενο με την χρήση μεθόδων μηχανικής μάθησης και βαθιάς μάθησης. Σε προσωπικό επίπεδο θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου για την αμέριστη στήριξή τους, και περισσότερο συγκεκριμένα την ξαδέρφη μου την Χρύσα και τον αδερφό μου, τον Κωνσταντίνο.

Πίνακας περιεχομένων

Ευρετήριο Σχημάτων

Ευρετήριο Πινάκων

Κεφάλαιο 1

Εισαγωγή

Εισαγωγή

Σε αυτό το πρώτο κεφάλαιο θα γίνει μια εισαγωγική αναφορά στο πρόβλημα που μελετήθηκε, στο στόχο καθώς και στην δομή που ακολούθησε αυτή η διπλωματική εργασία.

Πρόλογος

Την σημερινή εποχή πολύ άνθρωποι χρησιμοποιούν πλατφόρμες κοινωνικής δικτύωσης όπως το Twitter για ενημέρωση όσο και για ψυχαγωγία. Μεγάλο πρόβλημα όμως έχει εμφανιστεί από την αύξηση των ψευδών ειδήσεων και τη μειωμένη εμπιστοσύνη στα ΜΜΕ με αρνητικές επιπτώσεις στην κοινωνία μας. Τί μπορεί όμως να θεωρηθεί ψευδή είδηση; Προφανώς μία ιστορία με παραπλανητικό περιεχόμενο αποτελεί ψευδή είδηση όμως τον τελευταίο καιρό με την αύξηση της ενσυνείδησης στον συγκεκριμένο τομέα χρησιμοποιείται συχνά η έννοια για την απόρριψη δεδομένων τα οποία είναι αντίθετα στην προτιμώμενη άποψη κάποιου.

Η σημασία της διάδοσης ψευδών ειδήσεων ήρθε στο προσκήνιο κυρίως κατά την διάρκεια των Αμερικάνικων εκλογών του 2016. Ο όρος ψευδή είδηση ή “fake news” χρησιμοποιήθηκε έντονα εκείνη την περίοδο για να περιγράψει ειδήσεις ή άρθρα που περιείχαν λανθασμένες ή παραπλανητικές ειδήσεις με στόχο την πόλωση της κοινής γνώμης, το κέρδος από την επισκεψιμότητα των ιστοσελίδων αλλά και πολλές φορές από άτομα που το θεωρούσαν αστείο ή που αναπαρήγαγαν ψευδές περιεχόμενό εν αγνοία τους νομίζοντας πως είναι αληθές.

Το Twitter βρέθηκε στο επίκεντρο έντονης κριτικής από την στιγμή που πήρε διαστάσεις το θέμα. Τα τελευταία χρόνια έχουν υλοποιηθεί μέτρα για την αντιμετώπιση του θέματος, έχει ήδη εισαχθεί συγκεκριμένη κατηγορία για την αναφορά κάποιου tweet όταν γίνει αναφορά από κάποιον χρήστη, ενώ έχει γίνει δημόσια τοποθέτηση σχετικά με την αναζήτηση για έναν πιο αυτόματο τρόπο αναγνώρισης των ψευδών ειδήσεων. Αδιαμφισβήτητα δεν θα είναι εύκολη διαδικασία καθώς ένα μοντέλο θα πρέπει να είναι πολιτικά αμέριστο εφόσον οι ψευδείς ειδήσεις μπορούν να εμφανιστούν και από τα 2 άκρα του φάσματος, καθώς επίσης το θέμα της εγκυρότητας μιας είδησης είναι δύσκολο να προσδιοριστεί. Για την επίλυση του θέματος θα πρέπει να γίνει κατανοητό τι είναι μία ψευδή είδηση και σε επόμενο κεφάλαιο θα γίνει αναφορά σε άλλες εργασίες που δοκίμασαν να αντιμετωπίσουν παρόμοια προβλήματα με την βοήθεια της μηχανικής μάθησης και της επεξεργασίας φυσικής γλώσσας.

1.1 Στόχος Διπλωματικής

Στόχος της συγκεκριμένης εργασίας είναι η εφαρμογή και η συγκριτική αξιολόγηση μεθόδων μηχανικής μάθησης για τον εντοπισμό ψευδών ειδήσεων σε κείμενα πεπερασμένου μεγέθους όπως είναι τα tweets ή κείμενα από οποιοδήποτε άλλο microblog με τα συγκεκριμένα χαρακτηριστικά. Για την υλοποίηση της εργασίας χρησιμοποιήθηκε η γλώσσα python με τις βιβλιοθήκες scikit.learn, Tensorflow και Keras για την προεπεξεργασία των δεδομένων και για την εφαρμογή των αλγορίθμων μηχανικής μάθησης και βαθιάς μάθησης. Για την υλοποίηση των παραπάνω χρησιμοποιήθηκε ένα σύνολο δεδομένων με tweets τα οποία αφορούν την εκλογική αναμέτρηση στις ΗΠΑ του 2016 και αφού έγινε η κατάλληλη προεπεξεργασία αναπτύχθηκε ένα υβριδικό δίκτυο με συνδυασμό συνελικτικού και επαναλαμβανόμενου νευρωνικού δικτύου πετυχαίνοντας ικανοποιητικά αποτελέσματα σε σχέση με την πολυπλοκότητα του προβλήματος.

1.2 Δομή Διπλωματικής

Το κεφάλαιο 1 αποτελεί το εισαγωγικό μέρος της εργασίας, στο οποίο παρουσιάζεται το πρόβλημα που προσπαθεί να επιλύσει καθώς και ο στόχος της. Στο κεφάλαιο 2 γίνεται μία βιβλιογραφική επισκόπηση όπου αναφέρονται διαφορετικές προσεγγίσεις από άλλες εργασίες που προσπάθησαν να αντιμετωπίσουν προβλήματα σχετικά με την επεξεργασία της φυσικής γλώσσας. Στο κεφάλαιο 3 αναπτύσσεται το θεωρητικό υπόβαθρο στο οποίο βασίστηκε η εργασία. Στο κεφάλαιο 4 αναπτύσσεται η μεθοδολογία που χρησιμοποιήθηκε για την αντιμετώπιση διαφόρων προβλημάτων που προέκυψαν και στο κεφάλαιο 5 παρουσιάζονται τα συμπεράσματα και αποτελέσματα των μοντέλων που αναπτύχθηκαν.

Επίλογος

Το κεφάλαιο αυτό αποτέλεσε εισαγωγή για το θέμα που θα παρουσιαστεί σε αυτή την διπλωματική εργασία καθώς έγινε αναφορά πως ο λόγος που προέκυψε η ανάγκη για την δημιουργία μοντέλων αναγνώρισης ψευδών ειδήσεων ήταν η μεγάλη αύξησή των ψευδών ειδήσεων και η χρήση τους για παραπληροφόρηση. Επίσης παρουσιάστηκαν τα κεφάλαια από τα οποία αποτελείται αυτή η εργασία.

Κεφάλαιο 2

Βιβλιογραφική Αναφορά

Εισαγωγή

Στο Δεύτερο κεφάλαιο θα γίνει μια αναφορά σε προηγούμενες εργασίες και έρευνες που προσπάθησαν να αντιμετωπίσουν προβλήματα σχετικά με την επεξεργασία της φυσικής γλώσσας. Επίσης θα γίνει μία σύντομη αναφορά στα πλεονεκτήματα των συνελικτικών και επαναλαμβανόμενων νευρωνικών δικτύων.

2.1 Βιβλιογραφική Αναφορά

Αδόμητη πληροφορία στη μορφή κειμένου υπάρχει παντού, όπως σε email, σε συζητήσεις κοινωνικών δικτύων, ηλεκτρονικές ιστοσελίδες, απαντήσεις σε ερωτηματολόγια και πολλά άλλα. Οπότε φυσικό είναι να έχουν αναπτυχθεί μοντέλα και τεχνικές μηχανικής μάθησης που είναι πιο συγκεκριμένα με το πρόβλημα που προσπαθούν να επιλύσουν, για αυτό το λόγο θα γίνει αναφορά κυρίως αλγορίθμους που αντιμετωπίζουν μεγάλο εύρος των προβλημάτων ή αυτούς που έκαναν μεγάλη πρόοδο σε έναν συγκεκριμένο τομέα. Πολύ σημαντικό ρόλο στην κατηγοριοποίηση κειμένου αποτελεί η προεργασία που θα γίνει στα δεδομένα πριν χρησιμοποιηθούν για εκπαίδευση κάποιου μοντέλου. Μία κυρίαρχη τεχνική που παρουσιάζει πολύ καλά αποτελέσματα είναι γνωστή ως Bag Of Words ή και BOW, από τα αρχικά της. Η BOW αποτελεί έναν τρόπο για την εξαγωγή χαρακτηριστικών με σκοπό την χρήση τους σε κάποιο μοντέλο μηχανικής μάθησης, η μέθοδος που ακολουθεί είναι πολύ απλή και χρειάζονται μόνο 2 παράμετροι, ένα λεξικό που υποδεικνύει τις γνωστές λέξεις και έναν τρόπο μέτρησης της παρουσίας των λέξεων, συνήθως η συχνότητα εμφάνισης τους σε ένα κείμενο. Λόγω της προσέγγισης αυτής η μόνη πληροφορία που απομένει είναι η συχνότητα των λέξεων οπότε συνήθως η BOW χρησιμοποιείται σε συνδυασμό με κάποιον ταξινομητή όπως ο Naive Bayes. Σε έναν διαγωνισμό του Kaggle [2] βασιζόμενο σε κριτικές του IMDB πέτυχε ακρίβεια πάνω από 99% θεωρώντας τα δεδομένα με κριτικές κάτω του 5 αρνητικές και πάνω από 7 θετικές. Επίσης μία από τις πρώτες εμπορικές εφαρμογές μηχανικής μάθησης βασίστηκε στον συνδυασμό BOW με Naive Bayes για την αναγνώριση ανεπιθύμητης ηλεκτρονικής αλληλογραφίας[3] πετυχαίνοντας ακρίβεια και ανάκληση σε ποσοστά 97,6% , 94,3% και 87,8% , 94,7% σε ανεπιθύμητη και επιθυμητή αλληλογραφία αντίστοιχα, ενώ λίγα χρόνια μετά το ποσοστό αυτό βελτιώθηκε με ακρίβεια πάνω από 99,5%. Ένας διαφορετικός τρόπος για την εξαγωγή χαρακτηριστικών αποτελεί η μέθοδος N-gram κατά την οποία οι λέξεις αποτελούν σύνολα φράσεων σταθερού ή δυναμικού μεγέθους με αποτέλεσμα να παράγονται προβλέψεις για την τελευταία λέξη του συνόλου βάση των προηγούμενων. Ο [4] χρησιμοποίησε N-grams με σκοπό την ταξινόμηση κειμένων ανάλογα με τον συγγραφέα και σε ένα dataset από 39 έγγραφα τριών συγγραφέων κατάφερε να επιτύχει 100% επιτυχία χρησιμοποιώντας στην συνέχεια τον ταξινομητή SVM. Μία ακόμα προσέγγιση στην

ταξινόμηση κειμένων είναι και η αντιμετώπιση των δεδομένων ως ένα σύνολο χαρακτήρων, αυτή την προσέγγιση ακολούθησε ο [5]. Χρησιμοποίησε σύνολα δεδομένων από το Yahoo Answers, σχόλια του Amazon, κριτικές του Yelp, άρθρα του Wikipedia και άρθρα από το Sogou και το AG και με μόνα στοιχεία τα κείμενα και τις επικεφαλίδες κατάφερε να επιτύχει το μικρότερο σφάλμα στα σύνολα δεδομένων από τα Yahoo, Amazon και Yelp, σε σύγκριση με δίκτυα BOW, N-grams και LSTM, χρησιμοποιώντας ένα συνελικτικό δίκτυο 9 στρωμάτων. Με αυτό τον τρόπο φαίνεται πως τα συνελικτικά δίκτυα μπορούν να έχουν ικανοποιητικά αποτελέσματα και μάλιστα πιθανόν να είναι καλύτερα σε κείμενα που έχουν παραχθεί από χρήστες, πράγμα πολύ σημαντικό σε σενάρια του πραγματικού κόσμου. Επίσης σημειώνεται πως τα συνελικτικά μοντέλα ανταποκρίνονται όλο και καλύτερα όσο πιο πολλά δεδομένα χρησιμοποιηθούν για την εκπαίδευσή τους. Μία ακόμα αποτελεσματική μέθοδος είναι η Word2Vec, η οποία ως βάση χρησιμοποιεί την λογική πως λέξεις με κοινό νόημα έχουν και κοινή σημασία και νόημα. Οι λέξεις μετατρέπονται σε διανύσματα με αποτέλεσμα να μπορούν να γίνουν μαθηματικές πράξεις μεταξύ τους και ως αποτέλεσμα να προκύπτει ένα αποτέλεσμα που να αντικατοπτρίζει την διαφορά τους. Οι [6] εφηύραν και χρησιμοποίησαν αυτή την μέθοδο σε ένα σύνολο κειμένων από 6 δισεκατομμύρια λέξεις και πέτυχαν 50% και 64.5% ακρίβεια στην σημασιολογική και συντακτική συσχέτιση των λέξεων με την χρήση λεξικού μόνο 30 χιλιάδων λέξεων. Το συγκεκριμένο μοντέλο αποτελεί μοντέλο μη εποπτευόμενης μάθησης καθώς εξαρτάται απολύτως από τις συσχετίσεις τις γλώσσας.

Την τελευταία δεκαετία τα επαναλαμβανόμενα νευρωνικά δίκτυα(ΕΝΔ) ή RNN έχουν δείξει πως παρουσιάζουν καλά αποτελέσματα σχετικά με την αναγνώριση κειμένου. Το 2007 το [7] ήταν από τα πρώτα RNN που κατάφεραν να κερδίσουν σε διαγωνισμούς αναγνώρισης προτύπων στην κατηγορία αναγνώρισης χειρόγραφων κειμένων. Το 2014 η [8] κατάφερε να μειώσει το σφάλμα στην αναγνώριση φωνής στον διαγωνισμό Switchboard Hub5'00 κατά 2,5% χωρίς την χρήση παραδοσιακών μεθόδων αναγνώρισης φωνής, ενώ στην συνέχεια οι [9] κατάφεραν να πετύχουν καλύτερα αποτελέσματα από τα παραδοσιακά βαθιά νευρωνικά δίκτυα(DNN) χρησιμοποιώντας δίκτυα μεγάλης βραχυπρόθεσμης μνήμης(LSTM). Την ίδια χρονιά η Google χρησιμοποίησε LSTM [10] δίκτυα στην αναγνώριση φωνής του Android και λίγους μήνες μετά το εισήγαγε στην αναζήτηση με φωνή στην περίφημη μηχανή αναζήτησής της. Στο τέλος του 2019 η Google πάλι δημοσίευσε ένα καινούργιο μοντέλο[12] που βασίζεται σε δίκτυα LSTM διπλής κατεύθυνσης το οποίο εκπαιδεύεται με την τεχνική μη εποπτευόμενης μάθησης καλύπτοντας ένα μέρος των λέξεων και προσπαθώντας στην συνέχεια να τις προβλέψει, ενώ υπάρχει και αντίστοιχη μοντελοποίηση για την πρόβλεψη προτάσεων. Το μοντέλο ονομάζεται BERT, από τα αρχικά του που σημαίνουν αμφίδρομες κωδικοποιημένες παραστάσεις από μετασχηματιστές, και έχει πετύχει τα καλύτερα αποτελέσματα στο GLUE benchmark.

2.2 Πλεονεκτήματα ΣΝΔ και ΕΝΔ

Τα συνελικτικά και επαναλαμβανόμενα νευρωνικά δίκτυα ανήκουν και αυτά στην κατηγορία των τεχνητών νευρωνικών δικτύων οπότε έχουν πολλές ομοιότητες, όμως τα τελευταία χρόνια τα παραδοσιακά νευρωνικά δίκτυα έχει αποδειχθεί πως υστερούν. Αρχικά ας δούμε τις ομοιότητές τους. Και τα τρία είδη δικτύων έχουν την ικανότητα να μαθαίνουν τα βάρη και την προκατάληψη των συνάψεών τους

- Οι νευρώνες δέχονται μία είσοδο, παράγουν ένα εσωτερικό γινόμενο και ακολουθεί μία μη γραμμική συνάρτηση για να παραχθεί η έξοδος

Ένα συνελικτικό δίκτυο έχει μία ή περισσότερες συνελικτικές μονάδες. Τέτοιου είδους μονάδες δέχονται ως είσοδο εξόδους από πολλές μονάδες του προηγούμενου στρώματος με αποτέλεσμα να δημιουργείται μία γειτονιά. Με αυτόν τον τρόπο οι μονάδες εισόδου μοιράζονται τα βάρη τους. Οι συνελικτικές μονάδες τόσο όσο και οι συγκεντρωτικές μονάδες είναι ιδιαίτερα επωφελής επειδή:

- Μειώνουν τις μονάδες επεξεργασίας(νευρώνες) σε ένα δίκτυο λόγω του ότι η αντιστοίχισή τους είναι πολλά-προς-ένα. Αυτό σημαίνει πως υπάρχουν λιγότεροι παράμετροι να εκπαιδευτούν και έτσι μειώνεται η πιθανότητα της υπερεκπαίδευσης λόγω του ότι το μοντέλο θα είναι λιγότερο πολύπλοκο.
- Λαμβάνουν πληροφορία από γειτονικά στοιχεία. Αυτό το χαρακτηριστικό είναι πολύ σημαντικό σε πολλές εφαρμογές όπως εικόνες, κείμενο, βίντεο και επεξεργασία λόγου καθώς τα γειτονικά στοιχεία συνήθως κατέχουν σχετικές πληροφορίες μεταξύ τους.

Τα RNN είναι δίκτυα τα οποία χωρίζονται σε 4 κατηγορίες. Ένα-προς-ένα είναι η πιο συνηθισμένη μορφή τους, κατά την οποία υπάρχει μία είσοδος, όπως μία λέξη ή μία εικόνα και παράγεται ως αποτέλεσμα μία έξοδος, όπως μία λέξη ή μία δυαδική τιμή. Αντίστοιχα υπάρχουν τα ένα-προς-πολλά, πολλά-προς-ένα και πολλά-προς-πολλά. Οι κύριες διαφορές με τα κλασσικά νευρωνικά δίκτυα είναι:

- Αντίθετα με τα κλασσικά νευρωνικά δίκτυα που προωθούν την πληροφορία προς τα επόμενα στρώματα, οι έξοδοι κάποιον νευρώνων χρησιμοποιούνται ως είσοδοι σε ένα επόμενο στρώμα. Ως αποτέλεσμα τα RNN είναι κατάλληλα για την ανάλυση ακολουθιακών δεδομένων καθώς μπορούν και μαθαίνουν τις συσχετίσεις ανάμεσα σε προηγούμενα δεδομένα με τα τωρινά.
- Επίσης τα παραδοσιακά νευρωνικά δίκτυα είναι περιορισμένα σε εισόδους με σταθερό μήκος, πράγμα το οποίο δεν ισχύει με τα RNN.

Επίλογος

Σε αυτό το κεφάλαιο έγινε μία αναφορά στα δημοφιλέστερα μοντέλα όσον αφορά την ταξινόμηση κειμένου και την ανάλυση συναισθημάτων βάση αυτών καθώς και μία περιληπτική αναφορά των πλεονεκτημάτων CNN και RNN σε σχέση με τα κλασσικά τεχνητά νευρωνικά δίκτυα(ANN).

Κεφάλαιο 3

Θεωρητικό Υπόβαθρο

Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει μία ιστορική αναφορά στα πρώτα βήματα των νευρωνικών δικτύων και στην συνέχεια θα εξεταστεί το αναγκαίο θεωρητικό υπόβαθρο και οι βάσεις πάνω στις οποίες βασίστηκαν και δημιουργήθηκαν τα τεχνητά νευρωνικά δίκτυα. Στην συνέχεια θα γίνει επεξήγηση της λειτουργίας των CNN και RNN καθώς και μερικές από τις περισσότερο επιτυχημένες υλοποιήσεις τους.

3.1 Ιστορική αναφορά

Πιστεύεται πως η μελέτη για την μοντελοποίηση των νευρωνικών δικτύων ξεκίνησε το 1943 από τον νευρολόγο Warren McCulloch και τον μαθηματικό Walter Pitts, οι οποίοι δημοσίευσαν μία εργασία σχετικά με το πώς πιθανόν λειτουργούν οι νευρώνες. Για να περιγράψουν πως δουλεύουν οι νευρώνες στον εγκέφαλο δημιούργησαν ένα απλό νευρωνικό δίκτυο χρησιμοποιώντας ηλεκτρικά κυκλώματα. Το 1949 ο Donald Hebb δήλωσε το γεγονός στο βιβλίο του “The Organization Of Behaviour” πως η σύνδεση μεταξύ 2 νευρώνων δυναμώνει όσο περισσότερο χρησιμοποιείται, υποστήριξε δηλαδή πως αν 2 νευρώνες ενεργοποιηθούν μαζί η μεταξύ τους σύνδεση ενισχύεται.

Το 1962 οι Widrow και Hoff εξέλιξαν μία διαδικασία μάθησης κατά την οποία σε περίπτωση που ένας νευρώνας παρουσιάζει μεγάλο σφάλμα τα συναπτικά βάρη του μπορούν να προσαρμόζονται με σκοπό να διανέμεται το σφάλμα στο δίκτυο. Παρά την ανάπτυξη των νευρωνικών δικτύων, στην συνέχεια δεκαετία του 1960 προέκυψε στασιμότητα στον τομέα τόσο λόγο της προσοχής που στράφηκε στην αρχιτεκτονική “von Neumman” όσο και από το γεγονός πως η αρχική τους επιτυχία οδήγησε σε υπερβολή των δυνατοτήτων τους. Το επόμενο βήμα πραγματοποιήθηκε το 1972 από τους Kohonen και Anderson οι οποίοι ενώ δούλευαν ξεχωριστά παρουσίασαν παρόμοιες ιδέες, με την χρήση μαθηματικών μήτρας σχεδίασαν ένα πίνακα από δίκτυα ADALINE στα οποία οι νευρώνες ενεργοποιούν πολλαπλές εξόδους και όχι μόνο μία. Το 1982 το ενδιαφέρον στο πεδίο αναζωπυρώθηκε, ο John Hopfield παρουσίασε μία εργασία στην οποία χρησιμοποιούσε νευρώνες με αμφίδρομες συνδέσεις, κάτι που δεν είχε ξαναδοκιμαστεί, με σκοπό την δημιουργία περισσότερο χρήσιμων δικτύων. Τα επόμενα χρόνια το κυρίως πρόβλημα ήταν πως θα γίνει επέκταση της λύσης των Widrow και Hoff σε περισσότερα του ενός στρώματα. Τρείς διαφορετικές ερευνητικές ομάδες προέβαλαν παρόμοιες ιδέες που πλέον ονομάζεται δίκτυα πίσω διάδοσης με αυτόν τον τρόπο τα νευρωνικά δίκτυα μπορούσαν να επεκταθούν πέρα από τα 2 στρώματα που χρησιμοποιούνταν μέχρι τότε. Από το 1990 και μετά ο τομέας των τεχνητών νευρωνικών δικτύων έχει παρουσιάσει τεράστια άνθηση και έχει μπει στην καθημερινή μας ζωή καθώς έχει βρεθεί πρακτική εφαρμογή του σε επιστήμες όπως τα μαθηματικά, η φυσική, η ψυχολογία, τα οικονομικά και πολλές ακόμα.

3.2 Βιολογική Έμπνευση

Το έργο στο επιστημονικό πεδίο των τεχνητών νευρωνικών δικτύων βασίστηκε, από τις απαρχές του, στο γεγονός ότι ο ανθρώπινος εγκέφαλος εκτελεί τους υπολογισμούς με εντελώς διαφορετικό τρόπο από τον συμβατικό ψηφιακό υπολογιστή. Ο εγκέφαλος είναι ένας εξαιρετικά πολύπλοκος παράλληλος υπολογιστής που έχει την δυνατότητα να οργανώνει τα δομικά του στοιχεία, γνωστά ως νευρώνες, με τρόπο ώστε να εκτελούν συγκεκριμένους υπολογισμούς με ταχύτητα πολλαπλάσια από αυτή του γρηγορότερου ψηφιακού υπολογιστή. Μερικές από τις λειτουργίες αυτές είναι:

- Αναγνώριση εικόνων (προσώπων, αντικειμένων, κ.λπ..).
- Αναγνώριση φωνής, κατανόηση και παραγωγή γλώσσας.
- Αυτόνομη πλοήγηση στον χώρο
- Λήψη αποφάσεων
- Ανάπτυξη στρατηγικών, επιλογή των καλύτερων με βάση διάφορα κριτήρια κόστους.
- Μάθηση αυτό-προσαρμογή σε νέο περιβάλλον - καταστάσεις.
- Αναγνώριση προτύπων
- Αντίληψη και έλεγχο της κίνησης [13]

Το νευρικό κύτταρο ή νευρώνας είναι το βασικό δομικό στοιχείο του εγκεφάλου. Ο νευρώνας είναι ένα μεγάλο σε μέγεθος κύτταρο το οποίο, ανατομικά, αποτελείται από τα εξής τμήματα: το σώμα, τις συνάψεις, τον άξονα και τους δενδρίτες. Κάθε ένα από αυτά τα τμήματα έχει και διαφορετικό ρόλο.

- Οι δενδρίτες είναι οι πύλες εισόδου του νευρώνα. Δέχονται ηλεκτρικά σήματα από άλλους νευρώνες
- Ο άξονας είναι η πύλη εξόδου του νευρώνα. Η λειτουργία του άξονα είναι να στέλνει σήματα προς τους υπόλοιπους γειτονικούς νευρώνες.
- Οι συνάψεις είναι τα σημεία ένωσης μεταξύ διακλαδώσεων του άξονα ενός νευρώνα και τον δενδριτών άλλων νευρώνων. Αποτελούνται κυρίως από ηλεκτροχημικό υλικό και μεταφέρουν την ηλεκτρική δραστηριότητα που έρχεται από τον άξονα. Το πλάτος της σύναψης, η απόσταση της από τον δενδρίτη και η πυκνότητα του ηλεκτροχημικού υλικού επηρεάζουν την ευκολία με την οποία η ηλεκτρική δραστηριότητα μεταδίδεται από τον άξονα στο δενδρίτη. Το ποσοστό της ηλεκτρικής δραστηριότητας που μεταδίδεται στον δενδρίτη ονομάζεται συναπτικό βάρος.

3.3 Το πρώτο τεχνητό νευρωνικό δίκτυο

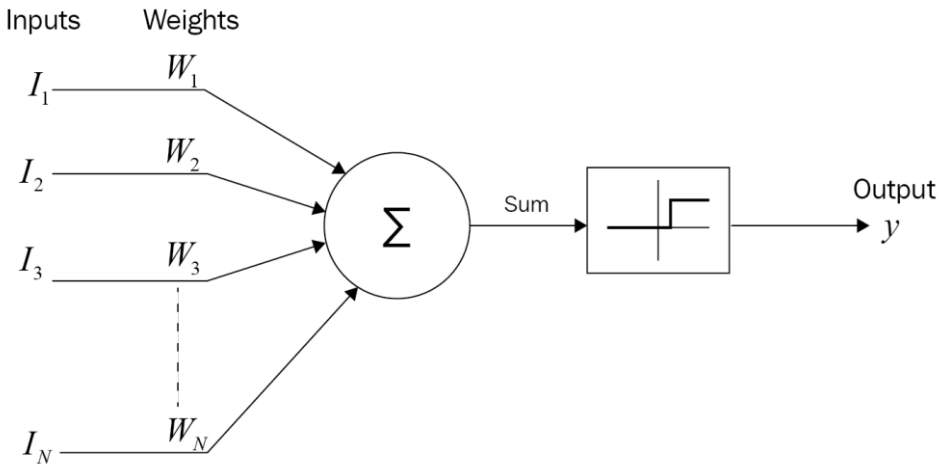
Το μοντέλο McCulloch-Pitts ήταν το πρώτο τεχνητό νευρωνικό δίκτυο. Με έμπνευση από τον βιολογικό νευρώνα οι δύο Αμερικανοί επιστήμονες περιέγραψαν ένα απλό μοντέλο της δραστηριότητας του νευρώνα.

Το μοντέλο χαρακτηρίζεται και αλλιώς ως ο πρώτος γραμμικός ταξινομητής και η κατάσταση του νευρώνα μπορεί να αποτελείται από μόνο 2 καταστάσεις :

- Y = 0 : Ο νευρώνας είναι αδρανής
- Y = 1 : Ο νευρώνας ενεργοποιείται στη μέγιστη συχνότητα

Ωστόσο ο ταξινομητής αυτός αντιμετώπιζε αρκετούς περιορισμούς στην λειτουργία του. Οι τιμές εισόδου και εξόδου μπορούν να είναι μόνο 0 ή 1, αν το άθροισμα του g είναι μεγαλύτερο από ένα κατώφλι τότε ο νευρώνας πυροβολεί, διαφορετικά παραμένει αδρανής.

Για να αντιμετωπιστούν κάποια από τα προβλήματα δημιουργήθηκε το μοντέλο perceptron από τους Minsky-Papert. Ξεπερνάει τους περιορισμούς του νευρώνα McCulloch-Pitts με την χρήση βαρών , ως μέτρο σπουδαιότητας, για τις εισόδους και ένα μηχανισμό για την εκπαίδευση αυτών των βαρών. Οι είσοδοι δεν είναι πλέον μόνο δυαδικές τιμές αλλά πραγματικοί αριθμοί.



Σχήμα 1 : Perceptron

Το μοντέλο παρουσιάζει πολλές ομοιότητες με τον νευρώνα M-P, ωστόσο οι συνάψεις περιγράφονται από συναπτικά βάρη (w_i) που είναι πραγματικοί αριθμοί, θετικοί για τις ενισχυτικές συνάψεις και αρνητικοί για τις ανασταλτικές συνάψεις. Το άθροισμα που προκύπτει ονομάζεται διέγερση του νευρώνα και δεδομένου πως οι είσοδοί του είναι της μορφής I_1, I_2, I_3,…..,I_n το φορτίο που δέχεται ο νευρώνας είναι:

$$u = Sum = I_1 w_1 + I_2 w_2 + \ldots + I_n w_n = \sum_{i=1}^{n} I_i w_i \quad (3.1)$$

Οπότε αν θεωρήσουμε το κατώφλι ενεργοποίησης ως ϑ η συνάρτηση ενεργοποίησης του νευρώνα μοιάζει κάπως έτσι:

$$f(u) = \begin{cases} 0, & u \leq \theta \\ 1, & u > \theta \end{cases} \text{ (3.2)}$$

Ωστόσο σύμφωνα με την σύμβαση αντί να τίθεται μία τιμή ϑ με το χέρι ως κατώφλι, προστίθεται ως μία ακόμα μεταβλητή εισόδου η οποία πάντα ισούται με μονάδα με το βάρος να είναι το ϑ , μετατρέποντας την εξίσωση:

$$u = u + \theta \text{ (3.3)}$$

Η πόλωση έχει ως αποτέλεσμα την αύξηση ή μείωση της δικτυακής διέγερσης της συνάρτησης ενεργοποίησης ανάλογα με το αν είναι θετική ή αρνητική. Έτσι η διέγερση που δέχεται πλέον ο νευρώνας μπορεί να αναπαρασταθεί ως:

$$u = Sum = I_1 w_1 + I_2 w_2 + \ldots + I_n w_n + \theta = \sum_{i=1}^{n} I_i w_i + \theta = \sum_{i=0}^{n} I_i w_i \text{(3.4)}$$

με $w_0 = \vartheta$ και $\chi_0 = 1$.

Οι συναρτήσεις ενεργοποίησης είναι πολύ σημαντικές για τα τεχνητά νευρωνικά δίκτυα για την αντιμετώπιση πολύπλοκων προβλημάτων. Κυρίως στόχος είναι η μετατροπή της εισόδου ενός νευρώνα σε ένα σήμα εξόδου που θα χρησιμοποιηθεί από τον επόμενο. Οι συναρτήσεις ενεργοποίησης μπορούν να χωριστούν σε 2 κατηγορίες, συναρτήσεις ενεργοποίησης με συνεχείς τιμές και συναρτήσεις ενεργοποίησης με εξόδους διακριτές τιμές:

Συναρτήσεις ενεργοποίησης:

- Βηματική Συνάρτηση -1/1 (Step Function)

  Με τύπο :

$$f(u) = \begin{cases} 0, & u \leq \theta \\ 1, & u > \theta \end{cases} \text{ (3.5)}$$

- Συνάρτηση κατωφλίου (Threshold Function)

  Με τύπο:

$$f(u) = \begin{cases} 0, & u \leq 0 \\ u, & 0 < u < 1 \\ 1, & x > 0 \end{cases} \text{ (3.6)}$$

- Συνεχής Συνάρτηση (Linear)

  Με τύπο:

$$f(u) = u \text{ (3.7)}$$

- Σιγμοειδής Συνάρτηση (Sigmoid)

  Με τύπο:

$$f(u) = \frac{1}{1+e^{-u}} \text{ (3.8)}$$

- Υπερβολική Εφαπτομένης (Hyperbolic Tangent)

  Με τύπο:

$$f(u) = \tanh(u) = \frac{1-e^{-u}}{1+e^{-u}} \text{ (3.9)}$$

- Συνάρτηση Ράμπας (ReLu)

  Με τύπο:

$$f(u) = \begin{cases} 0, & x \leq 0 \\ u, & x > 0 \end{cases} \text{ (3.10)}$$

- Συνάρτηση Ράμπας με διαρροή (Leaky ReLu)

  Με τύπο:

$$f(u) = \begin{cases} 0.1 * u, & x \leq 0 \\ u, & x > 0 \end{cases} \text{ (3.11)}$$

- Κανονικοποιημένη Εκθετική Συνάρτηση (Softmax)

  Με τύπο:

$$f(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \text{ (3.12)}$$

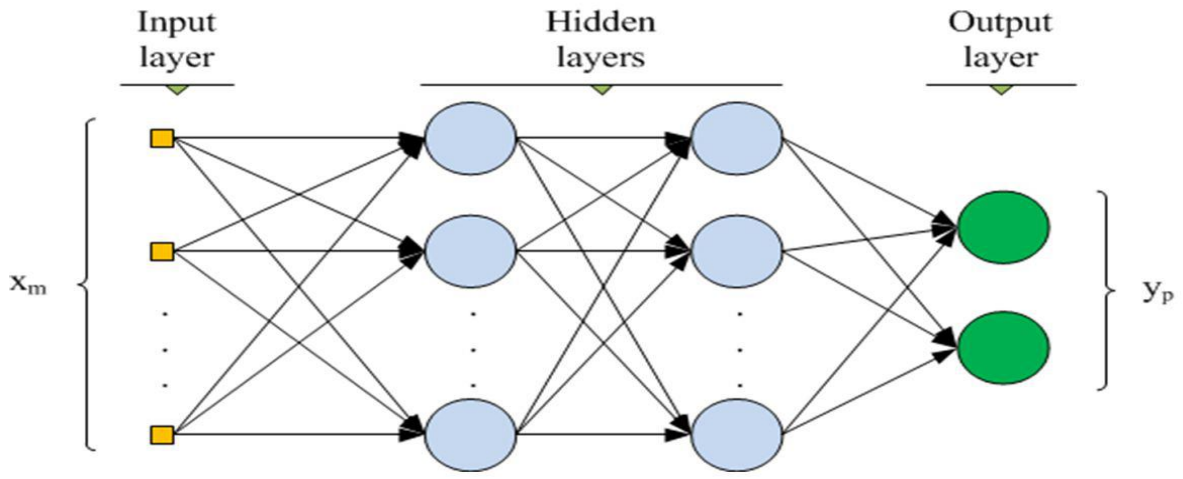
  Όπου:

  $z = \{I_0 w_0, I_1 w_1, \dots, I_n w_n\}$ kai $K = |z|$ (3.13)

Μέχρι τώρα σε αυτό το κεφάλαιο παρουσιάστηκε το μοντέλο M-P και το μοντέλο Perceptron. Μοντέλα με ένα μόνο νευρώνα είναι πολύ απλά στην λειτουργία τους και μπορούν να λύσουν μόνο τα ευκολότερα από τα προβλήματα. Ωστόσο αυτές οι δύο προσεγγίσεις δεν αποτελούν δίκτυα γιατί περιλαμβάνουν μόνο ένα νευρώνα. Οι περιορισμένες δυνατότητες των μοντέλων αυτών, όπως η αδυναμία επίλυσης της εξίσωσης XOR οδήγησε στην δημιουργία του Perceptron πολλαπλών επιπέδων.

3.4 Ο καθολικός ταξινομητής

Όπως είδαμε οι δυνατότητες αναπαράστασης διαχωριστικών επιφανειών είναι περιορισμένες στο δίκτυο Perceptron καθώς μόνο με ένα νευρώνα το δίκτυο μπορεί να αναπαραστήσει μόνο επίπεδες επιφάνειες. Ο περιορισμός αυτός αίρεται με την χρήση περισσότερων νευρώνων. Μία σειρά από νευρώνες ονομάζεται “στρώμα” και ένα νευρωνικό δίκτυο μπορεί να έχει πολλαπλά στρώματα. Η δυνατότητα των δικτύων πολλαπλών στρωμάτων προέρχεται από την ικανότητα να αναπαριστούν τα δεδομένα με τα οποία εκπαιδεύονται. Με αυτήν την έννοια τα νευρωνικά δίκτυα μαθαίνουν μία χαρτογράφηση και μαθηματικά έχει αποδειχθεί πως μπορούν να αναπαραστήσουν οποιαδήποτε συνάρτηση, για αυτό το λόγω θεωρούνται καθολικοί προσεγγιστές.



Σχήμα 2: Perceptron Πολλαπλών Στρωμάτων

Τα νευρωνικά δίκτυα πολλαπλών στρωμάτων ή αλλιώς MLP αποτελούνται 3 είδη στρωμάτων, το στρώμα εισόδου, τα κρυφά στρώματα και το στρώμα εξόδου. Το στρώμα εισόδου δεν αποτελείται από νευρώνες καθώς καμία λειτουργία δεν πραγματοποιείται, απλά προωθούνε τις τιμές εισόδου στο επόμενο στρώμα. Τα στρώματα μετά από το στρώμα εισόδου ονομάζονται κρυφά στρώματα γιατί δεν είναι κατευθείαν εκτεθειμένα στις εισόδους. Ένα δίκτυο είναι αρκετό να έχει μόνο ένα κρυφό στρώμα για να μπορεί να προσεγγίσει όσο είναι επιθυμητό οποιαδήποτε συνεχή συνάρτηση αρκεί να τηρεί τις εξής ιδιότητες:

- Να έχει αρκετούς κρυφούς νευρώνες, όσο πιο πολύπλοκο είναι το πρόβλημα τόσο περισσότεροι νευρώνες χρειάζονται για την αναπαράσταση της συνάρτησης.
- Οι νευρώνες του κρυφού στρώματος να έχουν την σιγμοειδή συνάρτηση ενεργοποίησης
- Ο νευρώνας εξόδου να έχει τη γραμμική συνάρτηση ενεργοποίησης

Το τελευταίο κρυφό στρώμα ονομάζεται στρώμα εξόδου και είναι υπεύθυνο για την έξοδο μίας τιμής ή ενός συνόλου τιμών, ανάλογα με το πρόβλημα που προσπαθεί να επιλυθεί. Η επιλογή συνάρτησης ενεργοποίησης στο στρώμα εξόδου περιορίζεται από το είδος του προβλήματος, δηλαδή :

- Για ένα πρόβλημα παλινδρόμησης μπορεί να υπάρχει μόνο ένας νευρώνας εξόδου χωρίς καμία συνάρτηση ενεργοποίησης
- Ένα δυαδικό πρόβλημα ταξινόμησης μπορεί να έχει ένα νευρώνα και να χρησιμοποιεί την σιγμοειδή συνάρτηση για να εξάγει μία τιμή ανάμεσα σε 0 και 1 η οποία αναπαριστά την πιθανότητα για την πρόβλεψη της κλάσης. Τότε με την χρήση κάποιας τιμής ως κατώφλι, π.χ. 0.5, η πιθανότητα μπορεί να μετατραπεί σε ακέραιο αριθμό 0, αν είναι κάτω από το κατώφλι και 1 διαφορετικά.
- Ένα πρόβλημα ταξινόμησης πολλαπλών τάξεων μπορεί να έχει πολλούς νευρώνες εξόδου, έναν για κάθε τάξη. Σε αυτή την περίπτωση η συνάρτηση ενεργοποίησης Softmax μπορεί να χρησιμοποιηθεί για να υπολογισθεί η πιθανότητα της πρόβλεψης να ανήκει σε κάποια από τις τάξεις.

Όπως είναι κατανοητό ένα δίκτυο MLP μόνο 2 στρώματα, ένα κρυφό και ένα στρώμα εξόδου, και την κατάλληλη διαμόρφωση μπορεί να προσεγγίσει οποιαδήποτε συνάρτηση, είτε είναι για ταξινόμηση είτε είναι για παλινδρόμηση.

3.5 Εκπαίδευση σε NLP

Αρχικά για να γίνει είσοδος των δεδομένων στο νευρωνικό δίκτυο πρέπει να γίνει η κατάλληλη προεργασία τους. Τα δεδομένα πρέπει να αναπαρασταθούν σε αριθμητική μορφή, ακόμα και αν είναι σε κατηγορηματική μορφή όπως το με τις τιμές “αρσενικό” θηλυκό”, μπορούν να μετατραπούν σε μία άλλη παράσταση που ονομάζεται *one hot encoding*. Μία καινούργια στήλη προστίθεται για κάθε διαφορετική τάξη και 0 ή 1 προστίθεται σε κάθε σειρά ανάλογα με την τιμή της τάξης για τη συγκεκριμένη σειρά. Δεδομένα όπως λέξεις μπορούν να μετατραπούν σε αριθμούς με ποικίλους τρόπους, όπως π.χ. την συχνότητα εμφάνισης μίας λέξης, στους οποίους θα γίνει αναφορά στην συνέχεια.

Ο πιο συνηθισμένος τρόπος εκπαίδευσης νευρωνικών δικτύων ονομάζεται κατάβαση δυναμικού. Στην συγκεκριμένη περίπτωση μία σειρά δεδομένων εμφανίζεται στο δίκτυο ως είσοδος, το δίκτυο επεξεργάζεται τις εισόδους ενεργοποιώντας νευρώνες και τελικά παράγει μία έξοδο, η διαδικασία αυτή ονομάζεται εμπρός τροφοδότηση και είναι παρόμοια με αυτήν που κάνει το δίκτυο μετά την εκπαίδευση με σκοπό την δημιουργία προβλέψεων. Η έξοδος που παρήγαγε συγκρίνετε με την κλάση στην οποία ανήκει, δηλαδή την αναμενόμενη έξοδο και το σφάλμα υπολογίζεται. Για τον υπολογισμό

του σφάλματος υπάρχουν διαφορετικές συναρτήσεις ωστόσο 2 είναι οι πιο δημοφιλείς, η εντροπία για προβλήματα ταξινόμησης και το μέσο τετραγωνικό σφάλμα για προβλήματα παλινδρόμησης. Η εντροπία υπολογίζεται από τον τύπο:

$$\mathrm{H(p,q)} = -\sum_x p(x) log q(x) \text{ (3.14)}$$

όπου x είναι το διάνυσμα της εισόδου, p(x) είναι η επιθυμητή πιθανότητα και q(x) είναι η πραγματική πιθανότητα του δικτύου. Ενώ το μέσω τετραγωνικό σφάλμα υπολογίζεται:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right) \text{ (3.15)}$$

Με *n* να είναι το σύνολο των δυνατών προβλέψεων, *Y* η επιθυμητή τιμή και $\hat{Y}$ η πρόβλεψη του μοντέλου.

Στην συνέχεια το σφάλμα διαδίδεται στο δίκτυο προς τα πίσω, δηλαδή από την έξοδο μέχρι το πρώτο κρυφό στρώμα, ένα στρώμα την φορά και τα βάρη ενημερώνονται ανάλογα με την ποσότητα που συνέβαλαν στον υπολογισμό του σφάλματος. Αν ο νευρώνας είναι στο τελευταίο στρώμα:

$$d_i(L) = (t_i - y_i) f'(u_i) \text{ (3.16)}$$

Και αν είναι σε οποιοδήποτε άλλο στρώμα:

$$d_i(l) = f'(u_i) \sum_{j=1}^{N(l+1)} w_{ji} d_j(l+1) \text{(3.17)}$$

Όπου $N(l+1)$ το πλήθος των νευρώνων στο στρώμα $l+1$, $f'(u_i)$ η παράγωγος της συνάρτησης ενεργοποίησης του νευρώνα και w_{ji} το διάνυσμα των βαρών του νευρώνα i στο στρώμα j.

Η διαδικασία επαναλαμβάνεται για κάθε παράδειγμα στο σύνολο δεδομένων εκπαίδευσης. Μία επανάληψη όλης της διαδικασίας ονομάζεται εποχή και κάθε δίκτυο μπορεί να εκπαιδευτεί για εκατοντάδες ή και χιλιάδες εποχές. Στην πραγματικότητα όμως ένα δίκτυο MLP είναι πολύ δύσκολο να επιλύσει πολύπλοκα προβλήματα και να πετύχει ποσοστά ακρίβειας παρόμοιας με έναν άνθρωπο, για αυτόν ακριβώς τον λόγω περαιτέρω ανάπτυξη στον τομέα των νευρωνικών δικτύων οδήγησε στην ανακάλυψη των συνελικτικών νευρωνικών δικτύων.

3.6 Συνελικτικά Δίκτυα

Τα συνελικτικά δίκτυα έχουν ενισχύσει στην ραγδαία ανάπτυξη των τεχνητών νευρωνικών δικτύων τα τελευταία χρόνια σε μεγάλο βαθμό, ωστόσο η ιστορία τους δεν είναι πρόσφατη. Η απαρχή έγινε το 1959 με την περαιτέρω εξέταση των βιολογικών νευρώνων. Οι David Hubel και Torsten Wiesel περιέγραψαν απλούς και πολύπλοκους νευρώνες που ανταποκρίνονταν σε οπτικά ερεθίσματα που εισέβαλαν μία γάτα, στην εργασία τους "*Receptive fields of single neurons in the cat's striate cortex*"[15]. Αυτό που παρατηρήθηκε ήταν πως συγκεκριμένοι νευρώνες ενεργοποιούνταν όταν τοποθετούταν ένα αντικείμενο σε συγκεκριμένη θέση ως προς τα μάτια μίας γάτας, όμως άλλοι νευρώνες ενεργοποιούνταν όταν το ίδιο αντικείμενο εμφανιζόταν σε άλλο σημείο της όρασης, τέτοιους νευρώνες τους ονόμασαν απλούς. Όμως υπήρχε και μία άλλη κατηγορία νευρώνων που ενεργοποιούνταν ανεξαρτήτως της θέσης του αντικειμένου στην εικόνα, αυτούς τους ονόμασαν πολύπλοκους. Πρότειναν οπότε πως οι πολύπλοκοι νευρώνες είναι αμετάβλητοι σε αλλαγές του χώρου, εφόσον το αντικείμενο παραμένει στο οπτικό πεδίο, επειδή μπορούσαν και άθροιζαν το αποτέλεσμα των απλών νευρώνων. Η ιδέα πως απλοί νευρώνες μπορούν να προστεθούν και να παράγουν περισσότερο πολύπλοκους συναντάται και στην ανθρώπινη όραση επίσης και αποτελεί την βάση για τον τρόπο λειτουργίας των συνελικτικών νευρωνικών δικτύων. Το 1980 ο Kunihiko Fukushima εμπνεύστηκε από την δουλειά τους και πρότεινε το συνελικτικό δίκτυο "*Neocognitron*". Χρησιμοποιούσε απλούς νευρώνες στην είσοδο και πολύπλοκους νευρώνες στην συνέχεια, με την

κύρια διαφορά τους να είναι πως μόνο οι συνάψεις των απλών νευρώνων μπορούσαν να τροποποιηθούν. Η ιδέα ήταν πως οι απλοί νευρώνες δέχονταν την πληροφορία και οι πολύπλοκοι ενεργούσαν ως ένα αφηρημένο στρώμα[16]. Το μοντέλο αυτό ενέπνευσε τελικά τους δημιουργούς του πρώτο συνελικτικού νευρωνικού δικτύου που μοιάζει αρκετά με τα σημερινά όμως αναπτύχθηκε το 1998 από τους Yann LeCun κ.α. Χρησιμοποίησαν 2 συνελικτικά στρώματα τα οποία στην συνέχεια συνέδεσαν με 2 στρώματα παρόμοια του NLP. Το μοντέλο χρησιμοποιήθηκε για την ταξινόμηση χειρόγραφων χαρακτήρων από το σύνολο δεδομένων *MNIST.* Παρατήρησαν πως οι επιδόσεις του μοντέλου βελτιώνονταν με όσο περισσότερα δεδομένα εκπαιδεύονταν, οπότε αφού χρησιμοποιήσαν 60.000 δεδομένα και 540.000 παραμορφωμένα δεδομένα, που δημιούργησαν οι ίδιοι από το αρχικό δείγμα, κατάφεραν να πετύχουν ποσοστό σφάλματος δοκιμής 0.8% από ένα σύνολο 10.000 αντικειμένων.

3.6.1 Η λειτουργία της συνέλιξης

Στα μαθηματικά η λειτουργία της συνέλιξης είναι μία πράξη σε 2 συναρτήσεις από την οποία προκύπτει μία τρίτη συνάρτηση, η οποία εκφράζει πώς το σχήμα της μίας μεταβάλει την άλλη. Ορίζεται ως το ολοκλήρωμα του γινομένου 2 συναρτήσεων, αφού η μία αντιστραφεί και μετατοπιστεί. Για την καλύτερη κατανόηση θα χρησιμοποιηθεί ένα παράδειγμα. Ας υποθέσουμε πως ρίχνουμε μία μπάλα από ένα ύψος στο έδαφος και σταματάει a μονάδες μακριά από την αρχική

τοποθεσία που προσγειώθηκε, κινούμενη σε μία διάσταση, με πιθανότητα $f(a)$ όπου f είναι η κατανομή πιθανότητας.

Μετά την πρώτη ρίψη σηκώνουμε την μπάλα και την ρίχνουμε από ένα διαφορετικό ύψος πάνω από το σημείο που είχε αρχικά σταματήσει. Η πιθανότητα η μπάλα να σταματήσει να κυλάει b μονάδες μακριά από το σημείο που την ρίξαμε είναι $g(b)$, όπου g είναι μία διαφορετική κατανομή πιθανότητας λόγω του διαφορετικού ύψους της ρίψης. Αν δεχθούμε πως η απομάκρυνση της μπάλας στην πρώτη ρίψη είναι σταθερή και ισούται με a μονάδες, τότε για να καλύψει μία συνολική απόσταση c η απόσταση της δεύτερης ρίψης θα πρέπει να είναι πάλι σταθερή στο b, με $a + b = c$. Οπότε η πιθανότητα να συμβεί αυτό είναι $f(a) \cdot f(b)$. Με συγκεκριμένους αριθμούς το παράδειγμα έχει ως εξής. Θέλουμε η μπάλα να καλύψει την συνολική απόσταση c που ισούται με 3. Αν την πρώτη φορά που σταματήσει έχει απομακρυνθεί $a = 2$ τότε την δεύτερη φορά πρέπει να κυλήσει $b = 1$ με σκοπό να καλύψει την απόσταση $a + b = 3$. Η πιθανότητα αυτή ισούται με $f(2) \cdot f(1)$. Ωστόσο αυτός δεν είναι ο μόνος τρόπος να πάρουμε ως αποτέλεσμα την συνολική απόσταση 3. Η μπάλα μπορεί να κυλήσει 1 μονάδα την πρώτη φορά και 2 μονάδες την δεύτερη, ή 3 μονάδες την πρώτη φορά και 0 την δεύτερη, δηλαδή το a και b μπορούν να πάρουν οποιεσδήποτε τιμές αρκεί το άθροισμά τους να είναι πάντα 3. Για να βρεθεί η συνολική πιθανότητα της μπάλας να διανύσει μία απόσταση c δεν γίνεται να θεωρηθεί μόνο ένας από τους πιθανούς τρόπους να διανύσει απόσταση c. Θεωρούμε όλες οι πιθανότητες το c να διαχωριστεί σε 2 ρίψεις a και b και προσθέτουμε την πιθανότητα της κάθε λύσης:

$$f(0) \cdot g(3) + f(1) \cdot g(2) + f(2) \cdot g(1) + \cdots \quad (3.18)$$

Και γνωρίζοντας από πριν πως η πιθανότητα κάθε περίπτωσης $a + b = c$ είναι $f(a) \cdot f(b)$ οπότε προσθέτοντας κάθε δυνατή πιθανότητα η συνολική πιθανότητα μπορεί να δηλωθεί ως:

$$(f * g)(c) = \sum_{a+b=c} f(a) \cdot g(b) \quad (3.19)$$

Και αν γίνει η αντικατάσταση $b = c - a$:

$$(f * g)(c) = \sum_{a} f(a) \cdot g(c - a) \quad (3.20)$$

Για να συνδέσουμε τα νευρωνικά δίκτυα με τη θεωρία της συνέλιξης θα χρησιμοποιηθεί ένα ακόμα παράδειγμα. Ας θεωρήσουμε ένα στρώμα νευρώνων με εισόδους x_n και εξόδους y_n όπου κάθε νευρώνας μπορεί να αναπαρασταθεί ως $A = \sigma(w_0 x_0 + w_1 x_1 + \ldots + b)$ με $x_0, x_1, \ldots$ είναι οι είσοδοι του νευρώνα και $w_1, w_2, \ldots$ τα βάρη του. Ένα αρνητικό βάρος προτρέπει τον νευρώνα από το να ενεργοποιηθεί ενώ ένα θετικό τον προτρέπει. Αυτή την σύνδεση των νευρώνων και ποια από τα

βάρη τους είναι πανομοιότυπα, είναι που θα χειριστούν οι συνελίξεις. Ως τώρα παρουσιάζονται οι νευρώνες ως ένα άθροισμα του γινομένου των βαρών τους με τις εισόδους τους, είναι στην πραγματικότητα πολύ πιο σύνηθες να εμφανίζονται τα βάρη όλων των νευρώνων ενός στρώματος σε ένα πίνακα, όπου κάθε σειρά αποτελεί και τα βάρη ενός νευρώνα για κάθε είσοδό του. Και επειδή κανονικά όλοι οι νευρώνες συνδέονται με όλες τις εισόδους ο πίνακας είναι :

$$W = \begin{bmatrix} W_{0,0} & W_{0,1} & W_{0,2} & W_{0,3} & \cdots \\ W_{1,0} & W_{1,1} & W_{1,2} & W_{1,3} & \cdots \\ W_{2,0} & W_{2,1} & W_{2,2} & W_{2,3} & \cdots \\ W_{3,0} & W_{3,1} & W_{3,2} & W_{3,3} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \quad (3.21)$$

Ο πίνακας των βαρών όμως για ένα συνελικτικό δίκτυο μοιάζει κάπως διαφορετικός, επειδή υπάρχουν πολλά αντίγραφα του ίδιου νευρώνα τα βάρη επαναλαμβάνονται και εμφανίζονται σε διαφορετικές θέσεις, και επειδή οι νευρώνες δεν συνδέονται με όλες τις πιθανές εισόδους υπάρχουν πολλά μηδενικά

$$W = \begin{bmatrix} w_0 & w_1 & 0 & 0 & \cdots \\ 0 & w_{1,0} & w_1 & 0 & \cdots \\ 0 & 0 & w_0 & w_1 & \cdots \\ 0 & 0 & 0 & w_0 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \quad (3.22)$$

Ο πολλαπλασιασμός με τον πίνακα (3.22) είναι όπως η συνέλιξη με $[\dots, 0, w_1, w_2, 0, \dots]$. Η ολίσθηση της συνάρτησης είναι σε διαφορετικές θέσεις αντιστοιχεί με την ύπαρξη νευρώνων σε αυτές τις θέσεις. Στο παράδειγμα η συνέλιξη πραγματοποιείται σε μία διάσταση όμως είναι δυνατόν να υπάρξει συνέλιξη σε πολλές διαστάσεις. Αν θεωρήσουμε στο παράδειγμα με την μπάλα πως η μετατόπιση a_1, b_1 πραγματοποιείται σε μία διάσταση ενώ η μετατόπιση $a_2 b_2$ σε μία άλλη, η εξίσωση παραμένει ίδια με την (3.19) μόνο που τώρα οι μεταβλητές a, b και c είναι διανύσματα οπότε ο τυπικός ορισμός της είναι:

$$(f * g)(c_1, c_2) = \sum_{a_1, a_2} f(a_1, a_2) \cdot g(c_1 - a_1, c_2 - a_2) \quad (3.20)$$

Όπου c_1 και c_2 οι συνολικές μεταβολές σε κάθε διάσταση.[18]

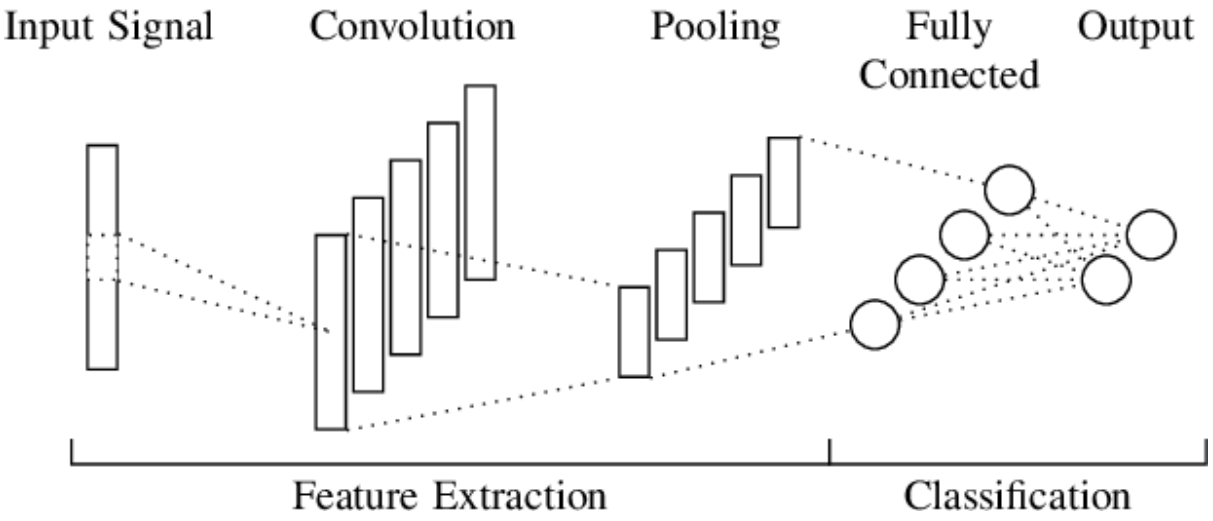
3.6.2 Οργάνωση Συνελικτικών δικτύων

Τα Συνελικτικά Νευρωνικά Δίκτυα (ΣΝΔ) ή στα Αγγλικά Convolutional Neural Networks (CNN), χωρίζονται σε δυο μεγάλες κατηγορίες, τα Αβαθή Νευρωνικά Δίκτυα (Shallow Neural Networks) και τα Βαθιά Νευρωνικά Δίκτυα (Deep Neural Networks). Ένα συνελικτικό επίπεδο (convolutional layer) είναι ουσιαστικά ένα σύνολο από νευρώνες που εκτελούν συνέλιξη των φίλτρων που έχουν προκαθοριστεί, με την εικόνα-διάνυσμα που δέχονται στην είσοδο. Κάθε επίπεδο μπορεί να περιλαμβάνει νευρώνες που εκτελούν συνέλιξη, διαδικασίες pooling, εισαγωγή μη γραμμικότητας ή ακόμη και κανονικοποίηση, ενώ έχει διακριτές εισόδους και εξόδους. Οι διαστάσεις των φίλτρων που περιλαμβάνουν, ο αριθμός τους και το βάθος τους (αριθμός καναλιών) μπορεί να διαφέρει σημαντικά ανάλογα με το πρόβλημα. Κατά τον παραδοσιακό τρόπο αναγνώρισης αντικειμένων σε εικόνα ή βίντεο, εικόνες ή αλληλουχία εικόνων (βίντεο) δίνονται σαν είσοδο σε κάποιο σύστημα προκειμένου να εκτελεστεί ο αλγόριθμος αναγνώρισης. Συνήθως δε, τα χαρακτηριστικά που χρησιμοποιούνται για την αναγνώριση είναι κατά βάση χαρακτηριστικά επιλεγμένα από ανθρώπινο παράγοντα, δηλαδή δεν μεσολαβεί κάποια διαδικασία εκμάθησης μέσω της οποίας εκπαιδεύεται ένας ταξινομητής για την εκπλήρωση του σκοπού της ταξινόμησης. Έτσι, από νωρίς έγινε γνωστό ότι το κλειδί στην αναγνώριση και ταξινόμηση, είναι η επιλογή σωστών χαρακτηριστικών και η καλή περιγραφή τους. Το ερώτημα που προέκυψε λοιπόν, συνοψίζεται στην ανάγκη για εύρεση μιας διαδικασίας εκμάθησης αυτών των χαρακτηριστικών, χωρίς την ανάγκη παρέμβασης του ανθρώπινου παράγοντα. Μια καλή προσέγγιση και λύση αυτού του προβλήματος, δίνουν τα CNN μέσω της εξαγωγής χαρακτηριστικών από ενδιάμεσα κρυφά επίπεδα. Όπως προαναφέρθηκε τα νευρωνικά δίκτυα χωρίζονται σε Αβαθή και Βαθιά νευρωνικά δίκτυα. Στην περίπτωση των βαθιών νευρωνικών δικτύων, ο αριθμός των συνελίξεων μπορεί να είναι σημαντικά αυξημένος και τα χαρακτηριστικά διανύσματα που προκύπτουν να έχουν ελαττωμένες τις διαστάσεις σε μεγάλο ποσοστό, χωρίς όμως να χάνουν την ουσία της πληροφορίας που περιλαμβάνουν. Όσο πιο “βαθύ” είναι το επίπεδο το οποίο εξετάζουμε, τόσο πιο “γενικά” είναι τα χαρακτηριστικά και πιο αναλλοίωτα. Γενικά ένα συνελικτικό δίκτυο αποτελείται από 3 είδη στρωμάτων: τα συνελικτικά στρώματα, τα στρώματα συγκέντρωσης και τα πλήρως συνδεδεμένα στρώματα. Παρά το ότι δεν υπάρχει κάποιος κανόνας για την σειρά με την οποία εισέρχονται τα στρώματα για να σχηματίσουν ένα δίκτυο συνηθίζεται να μπαίνουν πρώτα τα συνελικτικά και συγκεντρωτικά στρώματα και στο τέλος τα πλήρως συνδεδεμένα στρώματα. Καλό είναι εδώ να γίνει κατανοητό πως κάποια στρώματα έχουν ρυθμιζόμενες παραμέτρους και κάποια άλλα έχουν στατικές παραμέτρους. Τα συνελικτικά δίκτυα επειδή αποτελούν μία ειδική μορφή των νευρώνων Perceptron έχουν ρυθμιζόμενες παραμέτρους στην μορφή των βαρών και της προκατάληψης, ενώ το ίδιο ισχύει και για τους νευρώνες των πλήρως συνδεδεμένων δικτύων. Αντίθετα λόγω της λειτουργίας που επιτελούν, τα στρώματα συγκέντρωσης είναι στατικά και δεν διαφοροποιούνται κατά την διάρκεια της εκπαίδευσης του δικτύου.

Όταν τα προβλήματα που καλείται να επιλύσει ένα νευρωνικό δίκτυο περιλαμβάνουν εισόδους με μεγάλο μήκος ή πολλές διαστάσεις τότε το σύνολο των βαρών σε ένα πλήρως συνδεδεμένο δίκτυο

μπορεί να αποδειχθεί ιδιαίτερα μεγάλο με αποτέλεσμα να χρειάζονται υπολογιστές με πολύ μεγάλη επεξεργαστική ισχύ για την εκπαίδευσή του. Αντίθετα ένας νευρώνας μπορεί να συνδεθεί μόνο με ένα μικρό σύνολο από την είσοδο που μπορεί να χαρακτηριστεί ως γειτονιά. Το σύνολο αυτό αποτελεί μία υπερπαράμετρο του δικτύου και χαρακτηρίζεται ως το πεδίο όρασης ενός νευρώνα

Ένα συνελικτικό στρώμα αποτελείται από ένα σύνολο από ρυθμιζόμενα φίλτρα. Κάθε φίλτρο είναι μικρό χωροταξικά όμως επεκτείνεται σε όλο το μέγεθος της εισόδου. Για παράδειγμα ένα φίλτρο στο πρώτο στρώμα ενός συνελικτικού δικτύου μπορεί να έχει μέγεθος 3×1 (3 λέξεις ή γράμματα πλάτος και 1 ύψος, λόγου του ότι το κείμενο αποτελεί μονοδιάστατη είσοδο) ή $5 \times 5 \times 3$ (για εικόνες, με 5 pixel πλάτος και μήκος και 3 βάθος λόγω χρωματικών καναλιών RGB). Κατά την διάρκεια του προς τα εμπρός περάσματος ολισθαίνει το φίλτρο σε όλο το μήκος και πλάτος της εισόδου πραγματοποιώντας έτσι την συνέλιξη και δημιουργώντας ένα μονοδιάστατο πίνακα από το εσωτερικό γινόμενο του φίλτρου σε κάθε θέση στην οποία βρέθηκε στο διάνυσμα εισόδου. Με αυτόν τον τρόπο ο νευρώνας ενεργοποιείται όταν εισέρχονται δεδομένα όπως ένα σύνολο από γράμματα που δημιουργούν μία λέξη μέσα σε μία πρόταση ή μία γωνία με συγκεκριμένη κλίση από τα pixel μίας εικόνας. Τελικά στα επόμενα συνελικτικά στρώματα μαθαίνουν να αναγνωρίζουν ολόκληρες εκφράσεις μέσα σε προτάσεις ή μοτίβο όπως ένα κύκλος μέσα σε μία εικόνα.

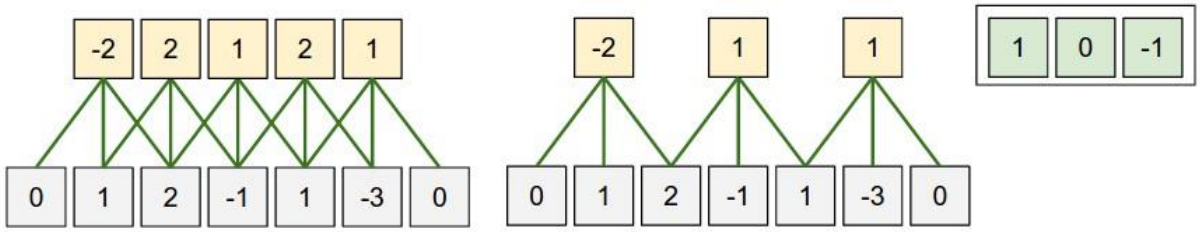


Σχήμα 3 : Συνελικτικό δίκτυο με συνέλιξη και συγκέντρωση μίας διάστασης και πλήρως συνδεδεμένο δίκτυο.

Η έξοδος των συνελικτικών στρωμάτων εξαρτάται από τρείς παράγοντες, το βάθος, το βήμα του φίλτρου και το γέμισμα του περιθωρίου.

- Το βάθος αποτελεί υπερπαράμετρο του δικτιού και είναι ίδια για κάθε νευρώνα σε ένα στρώμα. Αντιστοιχεί στον αριθμό των διαφορετικών φίλτρων που θα χρησιμοποιηθούν, κάθε ένα από αυτά θα εκπαιδευτεί στην εκμάθηση ενός διαφορετικού χαρακτηριστικού
- Το βήμα βάση του οποίου γίνεται η κύλιση του φίλτρου σε όλο το φάσμα της εισόδου. Αν τεθεί σε 1 τότε το φίλτρο προχωράει μία λέξη ή ένα pixel την κάθε φορά. Αν τεθεί σε μεγαλύτερο αριθμό τότε η έξοδος του νευρώνα παράγει μικρότερο σε μέγεθος αποτέλεσμα από την είσοδο.

- Το γέμισμα είναι μία τεχνική που χρησιμοποιείται κυρίως στην ανάλυση εικόνων με σκοπό το μέγεθος της εξόδου να είναι παρόμοιο με το μέγεθος της εισόδου. Στην τεχνική αυτή προστίθενται επιπλέον μηδενικές γραμμές και στήλες με σκοπό να μην επηρεάζεται το αποτέλεσμα.



Σχήμα 4 : Εφαρμογή φίλτρου σε νευρώνα μίας διάστασης

Στο σχήμα 6 φαίνεται η εφαρμογή φίλτρου 3×1 με βάρη $W = 1 \quad 0 \quad -1$. Στο πρώτο παράδειγμα το βήμα είναι 1 οπότε το φίλτρο περνάει πάνω από κάποια χαρακτηριστικά παραπάνω από μία φορακαι έχει χρησιμοποιηθεί και γέμισμα $P = 1$. Με αυτό τον τρόπο παράγεται αποτέλεσμα ίδιας διάστασης με την είσοδο. Στο δεύτερο παράδειγμα χρησιμοποιείται βήμα 2 δίνοντας έξοδο μεγέθους 3. Η έξοδος ενός νευρώνα μπορεί να υπολογισθεί από τον τύπο :

$$\frac{W - F + 2P}{S} + 1 \ (3.21)$$

Αντίστοιχα για εισόδους 2 διαστάσεων όπως για παράδειγμα εικόνες ο τύπος χωρίζεται σε:

$$\frac{W - F_w + 2P}{S_w} + 1 \ (3.22)$$

Με F_w και S_w να είναι το μέγεθος και το βήμα του φίλτρου ως προς το πλάτος και

$$\frac{W - F_h + 2P}{S_h} + 1 \ (3.23)$$

Με F_h και S_h να είναι το μέγεθος και το βήμα του φίλτρου ως προς το ύψος

Όπου W είναι το μέγεθος της εισόδου, F το φίλτρο που εφαρμόζεται, P το γέμισμα και S το βήμα του φίλτρου. Σημαντικό επίσης είναι να αναφερθεί πως οι νευρώνες ενός στρώματος μοιράζονται τα φίλτρα, αυτό βασίζεται στην λογική πως αν ένα χαρακτηριστικό είναι χρήσιμο να το δίκτυο σε ένα

μέρος της εισόδου τότε θα είναι χρήσιμο να αναγνωριστεί σε οποιοδήποτε σημείο της εισόδου. Με αυτόν τον τρόπο επιτυγχάνεται η μείωση του χρόνου για την εκπαίδευση του δικτύου.[20]

3.6.3 Μη γραμμικότητα

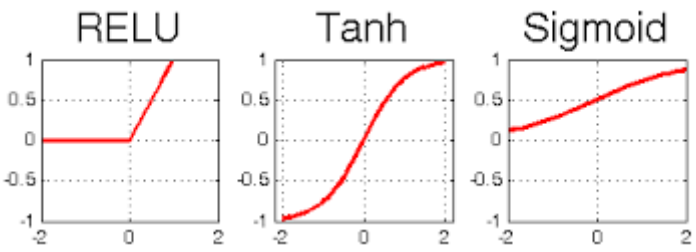
Η εισαγωγή της μη γραμμικότητας (non-linearity) δίνει σοβαρό πλεονέκτημα στα CNN έναντι άλλων γνωστών μεθόδων αντιμετώπισης πολλών προβλημάτων. Η γραμμική συνάρτηση δεν περιορίζει σε την έξοδο του νευρώνα σε κανένα εύρος τιμών. Σαν παράδειγμα αναφέρουμε την περίπτωση ενός μη γραμμικού συστήματος και την προσπάθεια πρόβλεψης αυτού, όπου οι γραμμικές μέθοδοι αδυνατούν να δώσουν καλά αποτελέσματα. Οπότε οι πιο δημοφιλείς συναρτήσεις ενεργοποίησης έχουν γίνει πλέον οι μη γραμμικές. Οι περισσότερο συνηθισμένες είναι οι Σιγμοειδής, η υπερβολική εφαπτομένη και η Rectified Linear Units (ReLU). Δεδομένου πως η εκπαίδευση γίνεται των δικτύων γίνεται με την χρήση της μεθόδου Back Propagation οι συναρτήσεις ενεργοποίησης της υπερβολικής εφαπτομένης και σιγμοειδούς εμφανίζουν κάποια μειονεκτήματα. Αρχικά η σιγμοειδής συνάρτηση:

- Δεν είναι γύρω από το μηδέν οπότε κάνει την βελτιστοποίηση δυσκολότερη. Έχει μικρή σύγκλιση
- Πάσχει από το πρόβλημα της εξάλειψης της παραγώγου.

Ενώ η υπερβολική εφαπτομένη έχει τιμές γύρω από το μηδέν εξακολουθεί να έχει παρόμοια προβλήματα με την σιγμοειδή, κάτι που κάνει την εκπαίδευση ιδιαίτερα δύσκολη. Σε αντίθεση η ReLU κερδίζει ολοένα και περισσότερο έδαφος καθώς έχει αποδειχθεί πολύ περισσότερο αποτελεσματική. Συγκεκριμένα η ReLU :

- Είναι πιο απλή και περισσότερο γρήγορη
- Μειώνει την πιθανότητα εξάλειψης της παραγώγου
- Λόγω του ότι όλες οι αρνητικές τιμές μετατρέπονται σε μηδέν ενεργοποιούνται μόνον όσοι νευρώνες χρειάζονται, κάνοντας την έτσι πιο αποδοτική στα CNN

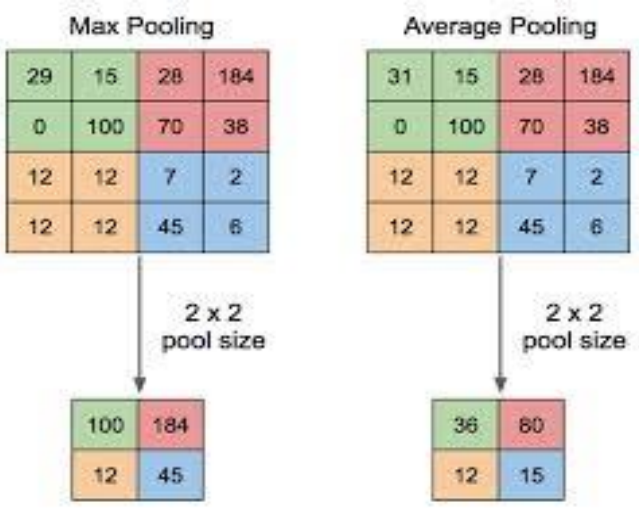
Το τελευταίο χαρακτηριστικό πολλές φορές μπορεί να λειτουργήσει και αρνητικά προκαλώντας κάποια ανανέωση των βαρών που θα έχει ως αποτέλεσμα να μην μπορεί να ενεργοποιηθεί ποτέ ξανά αυτός ο νευρώνας, ουσιαστικά σκοτώνοντας τους. Ωστόσο έχει αποδειχθεί πως παράγει πολύ καλά αποτελέσματα και είναι ιδιαίτερα δημοφιλής ως συνάρτηση ενεργοποίησης στους κρυφούς νευρώνες.



Σχήμα 5 : Γραφική απεικόνιση των συναρτήσεων ενεργοποίησης ReLU, Sigmoid και Tanh.

3.6.4 Συγκεντρωτικά Στρώματα

Τα επίπεδα συγκέντρωσης στα CNN συνοψίζουν τις τιμές γειτονικών ομάδων νευρώνων και παράγουν ως αποτέλεσμα μία μόνο τιμή. Η διαδικασία μπορεί να χαρακτηριστεί και ως υποδειγματοποίηση και έχει ως αποτέλεσμα να μειωθούν οι διαστάσεις της αναπαράστασης της πληροφορίας στα επόμενα στρώματα ώστε να μειωθεί ο αριθμός των συνολικών παραμέτρων και να μειωθούν οι υπολογιστικές απαιτήσεις του δικτύου. Μία κοινή μορφή συγκεντρωτικού στρώματος αποτελεί ένα φίλτρο με διαστάσεις 2×2 και βήμα 2. Για καλύτερη κατανόηση μπορεί κανείς να φανταστεί το στρώμα ως ένα πλέγμα με τις παραπάνω διαστάσεις που κυλάει πάνω από τις εξόδους γειτονικών συνελικτικών νευρώνων και εφαρμόζει μία από τρείς διαφορετικές συναρτήσεις, την συνάρτηση μεγίστου, μέσου όρου, ελαχίστου. Στην συνάρτηση μεγίστου αν υποθέσουμε ένα συγκεντρωτικό στρώμα με τις παραπάνω προδιαγραφές τότε για κάθε υπό-πίνακα 2×2 παράγεται ως αποτέλεσμα ένας στοιχείο το οποίο έχει το χαρακτηριστικό να είναι το μεγαλύτερο σε σχέση με τα υπόλοιπα. Αξιοσημείωτο είναι να αναφερθεί πως στην συγκεκριμένη περίπτωση δεν υπάρχει πολλαπλή επικάλυψη κάποιου στοιχείου από το πλέγμα, ενώ αν ένα χρησιμοποιούνταν ένα πλέγμα με ίδιες διαστάσεις αλλά βήμα 1 τότε θα κάποια στοιχεία θα συνέβαλαν σε 2 διαφορετικές θέσεις του πλέγματος. Αντίστοιχα η συνάρτηση του μέσου όρου δέχεται ως ορίσματα τα 4 στοιχεία του πλέγματος και παράγει τον μέσο όρο στέλνοντας το αποτέλεσμα στο επόμενο στρώμα. Το πλεονέκτημα στην χρήση της μέσης τιμής είναι πως όλες οι τιμές συμβάλουν στην είσοδο του επόμενου στρώματος σε αντίθεση με την χρήση μεγίστου ή ελαχίστου. Η συνάρτηση της μέσης τιμής έχει χρησιμοποιηθεί αρκετά ωστόσο τα τελευταία χρόνια έχει αποδειχθεί πως υπολείπει της μέγιστης τιμής η οποία λειτουργεί καλύτερα στην πράξη. Ενώ η συνάρτηση του ελαχίστου είναι ακριβώς αντίθετη με την συνάρτηση του μεγίστου και χρησιμοποιείται σε προβλήματα ελαχιστοποίησης.



Σχήμα 6 : Παράδειγμα συγκεντρωτικού στρώματος με φίλτρο 2×2 και βήμα 2. Στο πρώτο παράδειγμα χρησιμοποιείται η συνάρτηση μεγίστου παίρνοντας το αποτέλεσμα από 4 αριθμούς και προωθώντας το στο επόμενο στρώμα. Ενώ στο δεύτερο σχήμα γίνεται χρήση της συνάρτησης μέσου όρου.

Μάλιστα το δεδομένου ενός φίλτρου συγκέντρωσης με διαστάσεις F (το πλάτος και μήκος είναι πάντα ίδια γιατί το πλέγμα είναι τετραγωνικό), βήμα S και είσοδο με διαστάσεις $W \times H$ οι διαστάσεις που προκύπτουν μπορούν να υπολογιστούν από τον εξής τύπο:

$$W_2 = \frac{W_1 - F}{S} + 1 \quad (3.24)$$

και

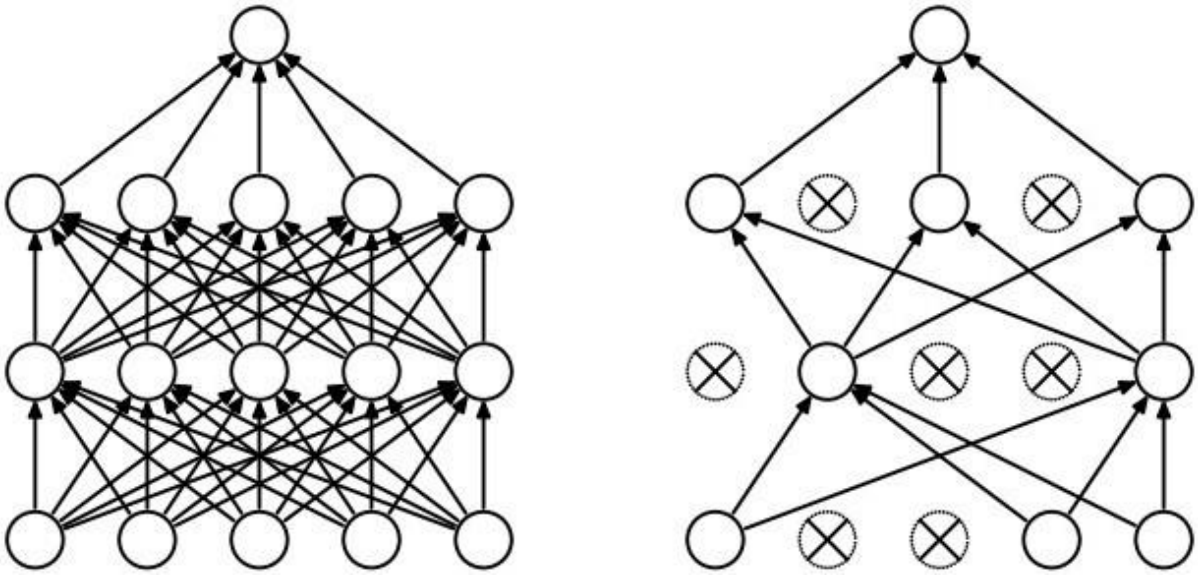
$$H_2 = \frac{H_2 - F}{S} + 1 \quad (3.25)$$

όπου $W_2 \times H_2$ οι διαστάσεις που προκύπτουν ως είσοδοι για το επόμενο στρώμα.[20]

3.6.5 Επίπεδο Κανονικοποίησης

Υπάρχουν διάφορες μέθοδοι για να μειώσουμε τα σφάλματα εκπαίδευσης ενός Νευρωνικού Δικτύου, όπως για παράδειγμα να συγκρίνουμε τις προβλέψεις πολλών και διαφορετικών μοντέλων. Για μεγάλα Νευρωνικά Δίκτυα των οποίων η εκπαίδευση μπορεί να κρατήσει αρκετές ημέρες αυτή η μέθοδος είναι χρονοβόρα και σε ορισμένες περιπτώσεις ίσως και αδύνατη. Προς αντιμετώπιση αυτού του προβλήματος έχει αναπτυχθεί μια μέθοδος σύγκρισης μοντέλων αρκετά αποτελεσματική, το υπολογιστικό κόστος της οποίας είναι πολύ χαμηλό. Η μέθοδος αυτή, η οποία ονομάζεται “Dropout” συνίσταται την ανάθεση ως “0” της εξόδου κάθε κρυφού νευρώνα με πιθανότητα 0.5. Οι νευρώνες που συμμετέχουν με αυτό τον τρόπο στο “Dropout” δεν συνεισφέρουν στη διάδοση προς τα εμπρός των σημάτων εκπαίδευσης και δεν συμμετέχουν στην διαδικασία του back-propagation. Έτσι, κάθε φορά που μια είσοδος παρουσιάζεται στο Δίκτυο, εκείνο χρησιμοποιεί διαφορετική αρχιτεκτονική αλλά όλες αυτές οι αρχιτεκτονικές μοιράζονται τα ίδια βάρη. Αυτή η τεχνική μειώνει τις περίπλοκες συν-προσαρμογές των νευρώνων κι έτσι το Δίκτυο αποκτά τη δυνατότητα εκμάθησης ισχυρότερων χαρακτηριστικών. Κατά τη διαδικασία της επαλήθευσης (test) χρησιμοποιούμε όλους τους νευρώνες πολλαπλασιάζοντας όμως τις εξόδους τους με 0.5 προκειμένου να πάρουμε τον γεωμετρικό μέσο των κατανομών πρόβλεψης. Τέλος, αναφέρουμε ότι η χρήση του Dropout σχεδόν διπλασιάζει τον αριθμό των επαναλήψεων που απαιτούνται για σύγκλιση.[21] Τρείς χρήσιμες παρατηρήσεις από την χρήση της κανονικοποίησης είναι:

- Αναγκάζει ένα νευρωνικό δίκτυο να μάθει περισσότερο εύρωστα χαρακτηριστικά που είναι περισσότερο χρήσιμα σε σύνδεση με τυχαίες ομαδοποιήσεις με άλλους νευρώνες.
- Το “Dropout” σχεδόν διπλασιάζει την εκπαίδευση που χρειάζεται ένα δίκτυο για να συγκλίνει, ωστόσο μειώνει σημαντικά τον χρόνο εκπαίδευσης κάθε εποχής.
- Δεδομένου πως υπάρχουν H κρυφοί νευρώνες μπορούν να υπάρξουν 2^H πιθανά δίκτυα.



Σχήμα 7 : Απεικόνιση επιπέδου κανονικοποίησης. Στα αριστερά ένα πλήρως συνδεδεμένο δίκτυο, ενώ δεξιά η μετατροπή μετά την χρήση Dropout.

3.6.6 Πλήρως συνδεδεμένο στρώμα

Ο ρόλος ενός πλήρως συνδεδεμένου στρώματος είναι να δεχθεί τα αποτελέσματα από τα συνελικτικά και συγκεντρωτικά στρώματα και να τα χρησιμοποιήσει ώστε να εκπαιδεύσει τις παραμέτρους του να κατηγοριοποιούν τις εισόδους σε τάξεις. Οι νευρώνες σε ένα πλήρως συνδεδεμένο στρώμα συνδέονται με όλους τους νευρώνες του προηγούμενου στρώματος, όπως δηλαδή στα απλά τεχνητά νευρωνικά δίκτυα. Οι έξοδοι των συνελικτικών στρωμάτων μετατρέπονται σε ένα διάνυσμα από τιμές, κάθε μία από τις οποίες αντιπροσωπεύει την πιθανότητα να υπάρχει ένα συγκεκριμένο χαρακτηριστικό στην είσοδο. Οι τιμές στην συνέχεια πολλαπλασιάζονται με τα βάρη και παράγεται το αποτέλεσμα που χρησιμοποιείται από την συνάρτηση ενεργοποίησης για να “πυροβολήσει” ή όχι ο νευρώνας. Δεδομένου της συνάρτησης ενεργοποίησης η δραστηριότητα του νευρώνα μπορεί να υπολογιστεί από έναν πολλαπλασιασμό του πίνακα των βαρών με τον πίνακα των εισόδων και προσθέτοντας την προκατάληψη ως αντιστάθμιση. Το τελευταίο στρώμα ενός συνελικτικού δικτύου είναι πάντα ένα πλήρως συνδεδεμένο και μπορεί να αποτελείται από διαφορετικό αριθμό νευρώνων κάθε φορά, ο αριθμός αυτός συνδέεται όμως με το πρόβλημα το οποίο προσπαθεί να αντιμετωπίσει το δίκτυο. Μία προσέγγιση είναι το τελευταίος στρώμα να αποτελείται από μόνο ένα νευρώνα, χρησιμοποιείται κυρίως σε δυαδικά προβλήματα ταξινόμησης όπου με την χρήση μίας τιμής ως κατώφλι ενεργοποιείται ή όχι ο νευρώνας αντιπροσωπεύοντας την μία κλάση αν ενεργοποιηθεί και την άλλη αντίθετα. Παραπάνω νευρώνες χρησιμοποιούνται συνήθως σε προβλήματα που οι κλάσεις είναι περισσότερες από 2, δηλαδή αν οι κλάσεις είναι 5 τότε τόσοι είναι και οι νευρώνες με κάθε έναν να αντιπροσωπεύει και μία κλάση.

3.6.7 Back Propagation σε CNN

Ας υποθέσουμε ότι με δ^{l+1} συμβολίζεται το σφάλμα για το $l+1$ επίπεδο στο δίκτυο με συνάρτηση κόστους $J(W,b;x,y)$ όπου (W,b) είναι το διάνυσμα των βαρών και η προκατάληψη και (x,y) οι είσοδοι και οι τάξεις που ανήκουνε αντίστοιχα. Αν το επίπεδο l είναι πυκνά συνδεδεμένο με το $l+1$ επίπεδο τότε το σφάλμα για το επίπεδο l υπολογίζεται με τον ακόλουθο τύπο:

$$\delta^{l} = \left(\left(W^{l}\right)^{T}\delta^{l+1}\right)\cdot f'(z^{l}) \text{ (3.26)}$$

Όπου $f'\left(z^{l}\right)$ είναι η παράγωγος της συνάρτησης ενεργοποίησης και οι κλίσεις είναι:

$$\nabla_{w^{l}}J(W,b;x,y) = \delta^{l+1}\left(a^{l}\right)^{T} \text{ (3.27)}$$

$$\nabla_{b^{l}}J(W,b;x,y) = \delta^{l+1} \text{ (3.28)}$$

Εάν το επίπεδο l είναι συνελικτικό τότε το σφάλμα διαδίδεται:

$$\delta_{k}^{l} = upsample\left(\left(W_{k}^{l}\right)^{T}\delta_{k}^{l+1}\right)\cdot f'(z_{k}^{l}) \text{ (3.29)}$$

Και ο αριθμός k αντιστοιχεί τον αριθμό του φίλτρου. Το σφάλμα πρέπει να διαδοθεί μέσω του επιπέδου συγκέντρωσης υπολογίζοντάς το ως προς κάθε νευρώνα εισερχόμενου από το επίπεδο συγκέντρωσης:

$$\nabla_{w_{k}^{l}}J(W,b;x,y) = \sum_{i=1}^{m}\left(a_{i}^{l}\right) * rot(\delta_{k}^{l+1},2) \text{ (3.30)}$$

$$\nabla_{w_{k}^{l}}J(W,b;x,y) = \sum_{a,b}(\delta_{k}^{l+1})_{a,b} \text{ (3.31)}$$

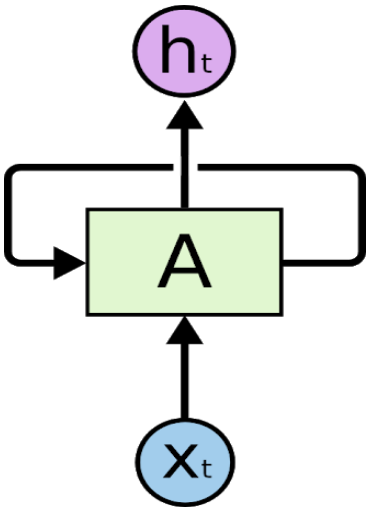
Με a^{l} είναι η είσοδος στο επίπεδο l και a είναι η εικόνα εισόδου.

3.7 Επαναλαμβανόμενα νευρωνικά δίκτυα

Τα επαναλαμβανόμενα νευρωνικά δίκτυα(RNN), που είναι γνωστά επίσης ως δίκτυα ανατροφοδότησης ανήκουν σε μία κατηγορία τεχνητών νευρωνικών δικτύων στα οποία οι σχέσεις μεταξύ των νευρώνων σχηματίζουν ένα κύκλο, οι έξοδοι των νευρώνων ταξιδεύουν ταυτόχρονα με κατεύθυνση μπροστά και πίσω δημιουργώντας βρόγχους στο δίκτυο. Με αυτό τον τρόπο δημιουργείται μία εσωτερική κατάσταση του δικτύου που επιτρέπει την ανάπτυξη δυναμικής χρονικής συμπεριφοράς. Αντίθετα με τα δίκτυα εμπρός τροφοδότησης, τα RNN μπορούν να χρησιμοποιήσουν την εσωτερική τους μνήμη ώστε να επεξεργαστούν αυθαίρετες αλληλουχίες πληροφορίας. Το δίκτυο Hopfield θεωρείται ένα από τα πρώτα RNN παρά το γεγονός πως δεν αποτελεί δίκτυο γενικής χρήσης και έχει περισσότερο ιστορικό ενδιαφέρων καθώς δεν σχεδιάστηκε για την επεξεργασία ακολουθιακής πληροφορίας. Κάθε νευρώνας δέχεται 2 εισόδους, η μία είναι οι πραγματικές είσοδοι για τις οποίες θέλουμε να βγάλουμε αποτελέσματα και η άλλη είσοδος είναι η έξοδος των υπόλοιπων νευρώνων. Εφευρέθηκε το 1982 από τον Hopfield και αν οι νευρώνες εκπαιδευτούν βάση την μάθησης Hebbian τότε το δίκτυο μπορεί να αποθηκεύσει πληροφορία η οποία είναι ανεκτική στην αλλοίωση της σύνδεσης των νευρώνων. Το 1993 ο Jeffrey L. Elman βασίστηκε στην παραπάνω αρχιτεκτονική για να δημιουργία του δικτύου Elman. Ένα δίκτυο 3 στρωμάτων x,γκαι z τα οποία είναι ακολουθιακά, με την προσθήκη ενός συνόλου μονάδων γενικού πλαισίου m. Υπάρχουν συνδέσεις από το μέσο κρυφό στρώμα προς αυτές τις μονάδες με βάρος σταθερό και ίσο με την μονάδα. Κατά την διάρκεια της εκπαίδευσης οι είσοδοι μεταφέρονται όπως σε ένα κανονικό νευρωνικό δίκτυο και στη συνέχεια ένας κανόνας μάθησης χρησιμοποιείται για την εκπαίδευση των βαρών. Οι ειδικές μονάδες γενικού πλαισίου κρατούν σταθερές τις τιμές τους δίνοντας την δυνατότητα στο δίκτυο να γνωρίζει την προηγούμενη κατάστασή του πετυχαίνοντας πολύ καλύτερα αποτελέσματα σε εργασίες όπως η πρόβλεψη επόμενης ακολουθίας από ένα δίκτυο MLP. Τα τελευταία χρόνια τα δίκτυα μακριάς-βραχείας μνήμης έχουν φέρει επανάσταση στον τομέα της αναγνώρισης ομιλίας καταφέρνοντας καλύτερα αποτελέσματα από οποιαδήποτε άλλη αρχιτεκτονική. Γενικά τα δίκτυα αυτά έχουν τα κατάλληλα εργαλεία για να πετυχαίνουν πολύ καλά αποτελέσματα σε προβλήματα ταξινόμησης, προ-επεξεργασίας και δημιουργίας προβλέψεων σε δεδομένα που είναι σε χρονική σειρά στον πραγματικό κόσμο. Η βασική ιδέα πίσω από αυτά τα δίκτυα είναι πως μπορούν να διατηρήσουν πληροφορία που χρησιμοποιήθηκε σε προηγούμενα βήματα για να επιτύχουν την έννοια του πλαισίου σε δεδομένα όπως το κείμενο.

3.7.1 Αρχιτεκτονική Επαναλαμβανόμενων νευρωνικών δικτύων

Το επαναλαμβανόμενο νευρωνικό δίκτυο (RNN) είναι ένας αλγόριθμος βαθιάς μάθησης, ο οποίος χρησιμοποιείται για τη μοντελοποίηση πληροφοριών σε σειριακή μορφή. Ο [22] όρισε το νευρωνικό δίκτυο ως ένα υπολογιστικό σύστημα το οποίο αποτελείται από απλά στοιχεία επεξεργασίας, με μεγάλο βαθμό διασύνδεσης, τα οποία διαχειρίζονται τις πληροφορίες με δυναμική ανταπόκριση στις εξωτερικές εισόδους. Το βασικό πλεονέκτημα αυτών των νευρωνικών δικτύων είναι η δυνατότητα μοντελοποίησης της επόμενης κατάστασης με βάση την προηγούμενη και σε μερικές παραλλαγές επιπλέον πλεονέκτημα είναι η μνήμη που διαθέτουν από παλαιότερες καταστάσεις. Τα RNN είναι εξαιρετικά ισχυρά όταν επιφορτίζονται με την επίλυση εργασιών νευρογλωσσικού προγραμματισμού, έχοντας πετύχει τον στόχο τους σε προβλήματα αναγνώρισης φωνής [23] και μηχανικής μετάφρασης [24].



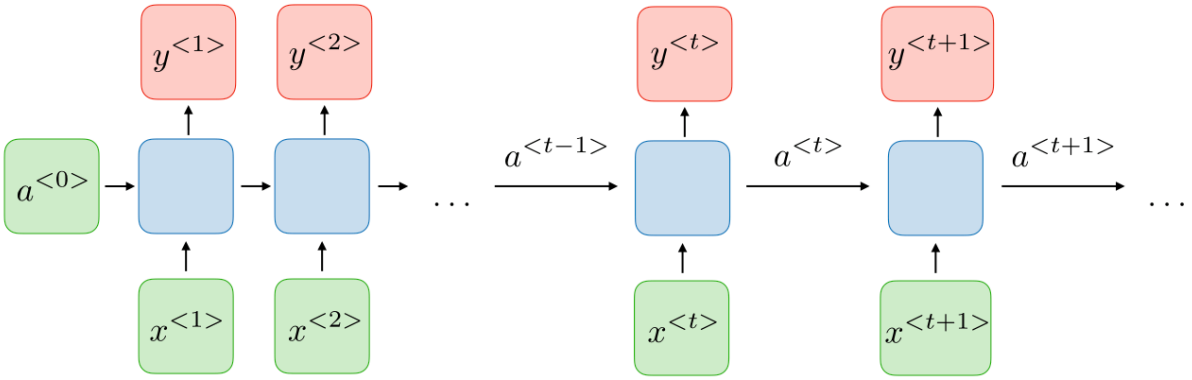
Σχήμα 8 : Διπλωμένο RNN

Η πρότυπη μορφή ενός RNN μπορεί να περιγραφή ως : δεδομένης μίας σειράς από διανύσματα εισόδου $(x_1, \dots, x_T)$ το δίκτυο παράγει μία ακολουθία από κρυφές καταστάσεις στους νευρώνες του$(y_1, \dots, y_T)$ και μία σειρά από εξόδους $(\hat{y}_1, \dots, \hat{y}_T)$ εκτελώντας συνεχώς τις ακόλουθες συναρτήσεις:

$$y_t = \mathrm{g}_1(W_{yx}X_t + W_{yy}y_{t-1} + b_y) \text{ (3.32)}$$

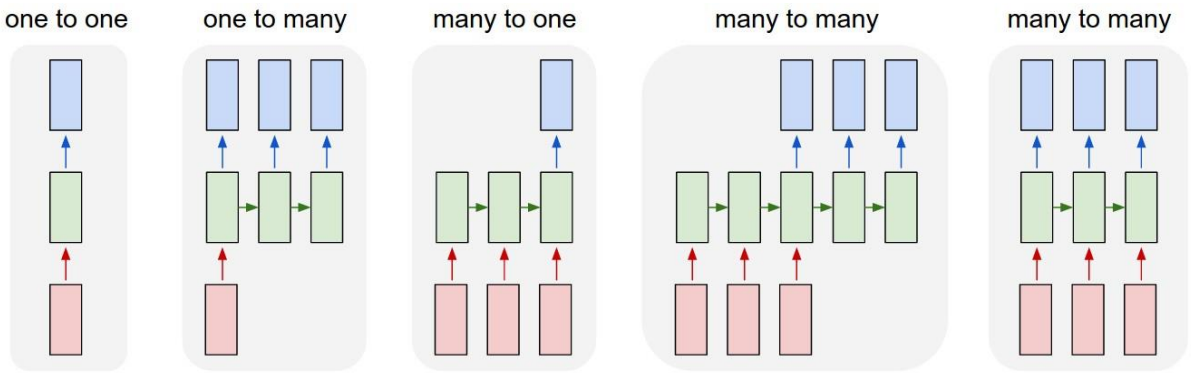
$$\hat{y}_t = g_2(W_{\hat{y}a}a_t + b_{\hat{y}}) \text{ (3.33)}$$

Στις παραπάνω συναρτήσεις το διάνυσμα W_{yx} αποτελεί το διάνυσμα βαρών για την είσοδο του νευρώνα όπως στα συνήθη νευρωνικά δίκτυα, το διάνυσμα W_{yy} είναι τα βάρη για την είσοδο από το προηγούμενο χρονικό βήμα του νευρώνα και $W_{\hat{y}a}$ τα βάρη για την έξοδο του νευρώνα προς το επόμενο στρώμα. Αντίστοιχα b_y και $b_{\hat{y}}$ είναι η προκατάληψη για την είσοδο από το προηγούμενο στρώμα και για την έξοδο, ενώ για την στιγμή $t = 1$ η συνάρτηση $W_{yy}y_{t-1}$ αντικαθίσταται από ένα ειδικό διάνυσμα πόλωσης a_{init} και εφαρμόζεται κανονικά η συνάρτηση ενεργοποίησης g_1.



Σχήμα 9 : Ξετυλιγμένο νευρωνικό δίκτυο με πολλές εισόδους και πολλές εξόδους. Ολόκληρο το διάνυσμα x μπορεί να αποτελεί μία πρόταση και κάθε x_n να είναι μία λέξη. Στο πρώτο βήμα δέχεται είσοδο x_1 και μία σταθερή αρχική τιμή a_0 και παράγει ως έξοδο y^1 και a^1

Τα επαναλαμβανόμενα νευρωνικά δίκτυα αποτελούν εντελώς ξεχωριστή κατηγορία από τα απλά νευρωνικά δίκτυα ή τα συνελικτικά δίκτυα γιατί δεν έχουν τους περιορισμούς αυτών των δικτύων. Τα διανύσματα εισόδου και εξόδου δεν χρειάζεται να είναι σταθερού μεγέθους αλλά και τα υπολογιστικά βήματα που συντελούνται μπορούν να είναι μεταβλητά. Τα RNN μπορούν να έχουν διανύσματα στις εισόδους, στις εξόδους και σε κάποιες περιπτώσεις και στα 2.



Σχήμα 10 : Διαφορετικέ υλοποιήσεις σε RNN

Όπως φαίνεται στο σχήμα 12 κάθε τετράγωνο είναι και ένα διάνυσμα και τα βέλη αποτελούν συναρτήσεις. Τα διανύσματα εισόδου είναι σε κόκκινο, τα διανύσματα εξόδου σε μπλέ και τα πράσινα διανύσματα αντιπροσωπεύουν την κρυφή κατάσταση του RNN. Από αριστερά προς δεξιά: (1) Η πιο απλή μορφή επεξεργασίας, χωρίς απαραίτητα να είναι RNN, με σταθερού μεγέθους εισόδους και εξόδους (π.χ. κατηγοριοποίηση εικόνων). (2) Πολλαπλές εξόδους από μία είσοδο (π.χ. υποτιτλισμός εικόνων, δέχεται μία εικόνα ως είσοδο και παράγει κείμενο για την περιγραφή της). (3) Πολλαπλές είσοδοι με μία έξοδο (π.χ. ανάλυση συναισθημάτων, όπου μία πρόταση δίνεται ως είσοδος και πρέπει να κατηγοριοποιηθεί ως θετική ή αρνητική). (4) Πολλαπλές ακολουθίες εισόδων και εξόδων (π.χ. Μετάφραση κειμένου, δέχεται ως είσοδο μία πρόταση στα αγγλικά και προσπαθεί να την μεταφράσει στα Ελληνικά). (5) Ισάριθμοι είσοδοι και έξοδοι (π.χ. κατηγοριοποίηση βίντεο, όπου έχει ως σκοπό την δημιουργία ετικετών για κάθε εικόνα)[24]

3.7.2 Εκπαίδευση σε RNN

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 49 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. Η συνάρτηση κόστους δεδομένου του χρόνου t ορίζεται ως:

$$E(y, \hat{y}) = -\frac{1}{N}\sum_t y_t log\hat{y_t} \quad (3.34)$$

Για ευκολία μπορεί να μετατραπεί σε:

$$E_t(y_t, \hat{y_t}) = -y_t log\hat{y_t} \quad (3.35)$$

Όπου y_t είναι η σωστή λέξη στο χρονικό βήμα t και $\hat{y}_t$ η πρόβλεψη του δικτύου. Στην συνέχεια θα θεωρήσουμε για καλύτερη κατανόηση των μαθηματικών υπολογισμών κάθε διάνυσμα βαρών με διαφορετικό γράμμα με V να είναι το διάνυσμα βαρών για την έξοδο του νευρώνα, W το διάνυσμα για την κρυφή κατάσταση και U τα βάρη για την είσοδο του νευρώνα σε κάθε χρονικό βήμα. Κάθε πρόταση αποτελεί μία είσοδο για το σύστημα οπότε το συνολικό σφάλμα είναι η άθροιση των σφαλμάτων από κάθε χρονικό βήμα. Σκοπός πάντα είναι να υπολογιστεί η κλίση της συνάρτησης του σφάλματος σχετικά με τις παραμέτρους U, V, W ώστε να βελτιωθούν οι παράμετροι με την χρήση Στοχαστικής Κατάβασης Δυναμικού(Stochastic Gradient Decent).Έτσι όπως αναφέραμε πως προσθέτουμε τα σφάλματα προσθέτουμε και την κλίση του σφάλματος σε κάθε χρονικό βήμα:

$$\frac{\partial E}{\partial W} = \sum_t \frac{\partial E_t}{\partial W} \quad (3.36)$$

Για την συνέχεια θα χρησιμοποιηθεί το σφάλμα E_3 με είσοδο X_3και συνάρτηση ενεργοποίησης S_3, όπου 3 είναι το χρονικό βήμα. Για τον υπολογισμό των κλίσεων χρησιμοποιείται ο αλυσιδωτός κανόνας της διαφοροποίησης:

$$\frac{\partial E}{\partial V} = \frac{\partial E_3}{\partial \hat{y}_3}\frac{\partial \hat{y}_3}{\partial z_3}\frac{\partial z_3}{\partial V} = (\hat{y}_3 - y_3) \otimes s_3 \ (3.37)$$

Όπου $z_3 = Vs_3$ και $\otimes$ το εξωτερικό γινόμενο των διανυσμάτων.

Από την σχέση (3.37) προκύπτει πως η κλίση της συνάρτησης του σφάλματος εξαρτάται από τις μεταβλητές των τιμών $\hat{y}_3, y_3, s_3$. Οπότε αν είναι γνωστές οι τιμές τους η κλίση υπολογίζεται από ένα πολλαπλασιασμό πινάκων. Ωστόσο η κατάσταση είναι λίγο διαφορετική λόγω του s_3. Το όπως φαίνεται και από την (3.32) εξαρτάται από το s_2, το οποί με την σειρά του εξαρτάται από το s_1 και ούτω καθεξής. Αυτό έχει ως αποτέλεσμα να μην μπορεί να θεωρηθεί το s_3ως σταθερά. Οπότε εφαρμόζοντας τον αλυσιδωτό κανόνα στην σχέση (3.36) προκύπτει:

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial W}\frac{\partial \hat{y}_3}{\partial s_3}\frac{\partial s_3}{\partial s_k}\frac{\partial s_k}{\partial W} \ (3.38)$$

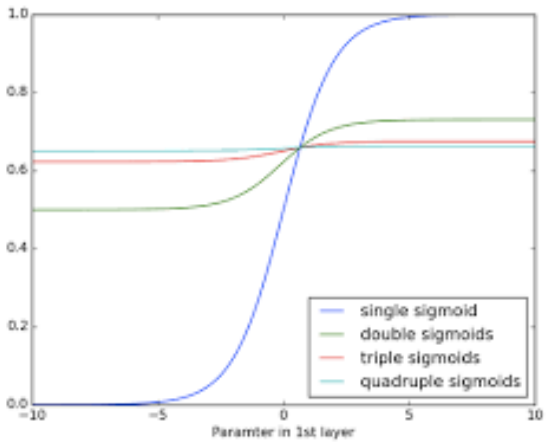
Για καλύτερη κατανόηση ένα ξετύλιχτο RNN με t χρονικά βήματα μπορεί να θεωρηθεί και ως ένα νευρωνικό δίκτυο με t στρώματα όπου οι παράμετροι W, U, V είναι κοινοί ανάμεσα στα στρώματα. Η σημαντική διαφορά με ένα σύνηθες νευρωνικό δίκτυο είναι πως οι παράμετροι των βαρών δεν είναι κοινοί ανάμεσα στα στρώματα οπότε δεν υπάρχει η ανάγκη για τον υπολογισμό της συνολικής κατάβασης, σε αντίθεση με τα επαναλαμβανόμενα νευρωνικά δίκτυα όπου πρέπει να προστεθούν οι κλίσεις μέχρι το αρχικό στρώμα για την βελτιστοποίηση των παραμέτρων.

3.7.3 Το πρόβλημα των Εξαφανιζόμενων/Ανατινασσόμενων Κλίσεων (Vanishing/Exploding Gradients)

Τα ανατροφοδοτούμενα νευρωνικά δίκτυα, όπως τα περισσότερα νευρωνικά δίκτυα, έχουν δημιουργηθεί εδώ και πολλά χρόνια. Παρά το ότι η εκπαίδευσή τους φαίνεται εύκολη με την χρήση της οπισθοδιάδοσης μέσα στον χρόνο, στην πραγματικότητα η σχέση μεταξύ των παραμέτρων των RNN είναι πολύ ασταθής κάτι που κάνει την μέθοδο της κατάβασης δυναμικού ιδιαίτερα αναποτελεσματική. Το 1991 ο [25] απέδειξε πως η κλίση εξαφανίζεται ή πιο σπάνια εκτινάσσεται εκθετικά κατά την μέθοδο της οπισθοδιάδοσης μέσα στον χρόνο και χρησιμοποίησε τα αποτελέσματά του για να υποστηρίξει πως τα RNN δεν είναι ικανά να “μάθουν” συσχετίσεις μεγάλων αποστάσεων στα διανύσματα εισόδου. 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. 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

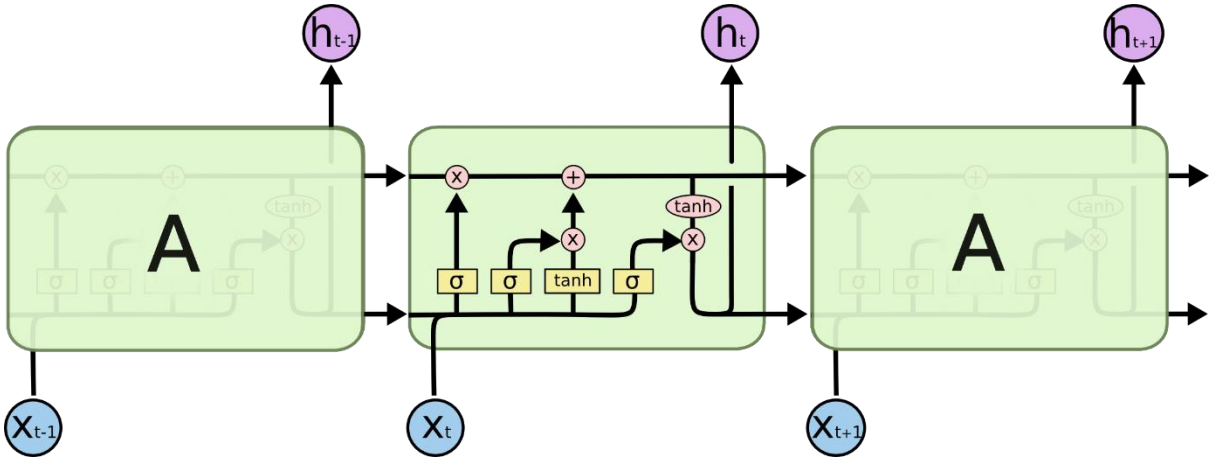
μεμονωμένες εισόδους. Για αυτό, ευθύνεται, εν μέρει, το γεγονός ότι οι πληροφορίες που κυλούν μέσω των νευρωνικών δικτύων περνάνε από πολλά στάδια πολλαπλασιασμών. Επειδή τα επίπεδα και τα χρονικά βήματα βαθιών νευρωνικών δικτύων συνδέονται μεταξύ τους μέσω πολλαπλασιασμών, οι παράγωγοι είναι επιρρεπείς 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 11 : Αποτέλεσμα εφαρμογής της σιγμοειδούς συνάρτησης πολλαπλές φορές

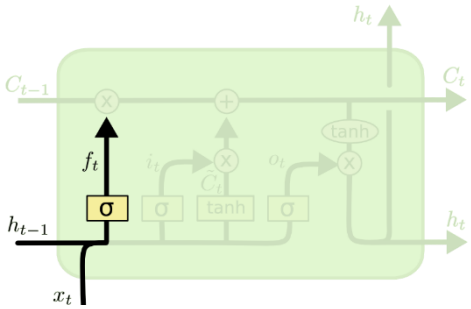
3.7.4 Βραχυπρόθεσμες Μονάδες Μνήμης Μακράς Διάρκειας(Long Short-Term Memory Units)

Τα δίκτυα LSTM είναι μία ειδική κατηγορία RNN τα οποία είναι ικανά να μάθουν μακροπρόθεσμες εξαρτήσεις. Η ιδέα για την αρχιτεκτονική αυτών των δικτύων εισήχθη από τους Hochreiter και Schmidhuber το 1997[25] και στην συνέχεια αναπτύχθηκαν και έγιναν δημοφιλή από πληθώρα επιστημόνων σε ακόλουθες εργασίες. Πλέον έχει αποδειχθεί πως είναι πολύ αποτελεσματικά σε μεγάλο εύρος προβλημάτων και χρησιμοποιούνται για πολλούς σκοπούς. Τα LSTM έχουν σχεδιαστεί με ένα γνώμονα κατά νου και αυτό είναι να αποφύγουν το πρόβλημα των μακροπρόθεσμων εξαρτήσεων. Η απομνημόνευση πληροφορίας για μεγάλο χρονικό διάστημα είναι η προκαθορισμένη συμπεριφορά τους. Όλα τα επαναλαμβανόμενα νευρωνικά δίκτυα έχουν ένα σύνολο από επαναλαμβανόμενες μονάδες που σχηματίζουν μία αλυσίδα, στα πιο κλασσικά από αυτά οι μονάδες τους θα αποτελούνται στην ουσία από μία απλή συνάρτηση όπως η υπερβολική εφαπτομένη ή η σιγμοειδής. Οι μονάδες στα LSTM στρώματα επίσης συνδέονται ακολουθώντας η μία την άλλη σαν αλυσίδα ωστόσο με διαφορετική δομή. Αντί να υπάρχει μόνον μία συνάρτηση ενεργοποίησης υπάρχουν 4, οι οποίες αλληλοεπιδρούν με έναν ιδιαίτερο τρόπο



Σχήμα 12 : Ξετυλιγμένος νευρώνας δικτύου LSTM

Το κλειδί για τη λειτουργία των δικτύων LSTM είναι η γραμμή που διέρχεται στο σχήμα 12 από την κορυφή του νευρώνα. Η γραμμή αυτή αποτελεί την κατάσταση του νευρώνα και διέρχεται κατά μήκος ολόκληρης της αλυσίδας με λίγες μόνο αλληλεπιδράσεις οπότε δεν είναι δύσκολο να μείνει ανεπηρέαστη η πληροφορία που μεταφέρει για μεγάλο χρονικό διάστημα. Το δίκτυο ωστόσο πάντα μπορεί να προσθέσει ή να αφαιρέσει πληροφορία για με την χρήση δομών που ονομάζονται πύλες. Οι δομές αυτές αποτελούνται από μία σιγμοειδή συνάρτηση σ και ένα πολλαπλασιασμό, η συνάρτηση σ μπορεί να έχει ως έξοδο τιμές από 0 μέχρι 1, όπου για 0 επιτρέπει την κατάσταση του νευρώνα να παραμείνει αμετάλλακτη εντελώς, ενώ το 1 να διαγραφεί και να ξαναδημιουργηθεί βάση των εισόδων της τωρινής μονάδας. Το πρώτο βήμα σε ένα δίκτυο LSTM είναι η απόφαση που πρέπει να ληφθεί για την διατήρηση ή όχι και σε τί ποσοστό από την προηγούμενη κατάσταση του νευρώνα.

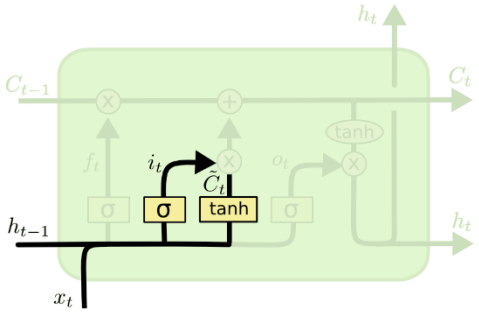


Σχήμα 13 : Το πρώτο στρώμα του LSTM, ονομάζεται αλλιώς και πύλη λήθης ή forget gate, καθώς εδώ αποφασίζεται σε τι ποσοστό θα διατηρηθεί η κατάσταση του νευρώνα αμετάλλακτη

Η απόφαση αυτή παίρνετε από μία σιγμοειδή συνάρτηση η οποία δέχεται ως εισόδους σε χρόνο t, την κρυφή κατάσταση του νευρώνα από την προηγούμενη χρονική στιγμή h_{t-1} και την τωρινή είσοδο x_t ,δημιουργεί μία αλληλουχία και παράγει μία έξοδο μέσω την συνάρτησης ενεργοποίησης σ :

$$f_t = s(W_f \cdot [h_{t-1}, x_t] + b_f) \text{ (3.39)}$$

Στην συνέχεια όταν γίνει η ανανέωση της κατάστασης του νευρώνα θα συζητηθεί η χρήση της f_t. Για την καλύτερη κατανόηση θα χρησιμοποιηθεί ένα παράδειγμα. Ας υποθέσουμε πως ένα δίκτυο προσπαθεί να προβλέψει την επόμενη λέξη βάση όλων των προηγούμενων. Σε μία τέτοια κατάσταση το δίκτυο μπορεί να χρειαστεί να συμπεριλάβει το γένος του τωρινού θέματος ώστε να χρησιμοποιήσει τις σωστές αντωνυμίες, ενώ όταν υπάρξει αλλαγή θέματος το δίκτυο θα πρέπει να ξεχάσει το προηγούμενο γένος. Στο επόμενο βήμα θα πρέπει να αποφασιστεί ποια κομμάτια πληροφορίας από την είσοδο του νευρώνα (h_{t-1}, x_t) θα επηρεάσουν την κατάσταση του C_t. Στο παράδειγμα που αναφέρθηκε θα πρέπει το μοντέλο να μάθει το γένος του καινούργιου θέματος έχοντας ξεχάσει πλέον το παλιό.



Σχήμα 14 : Το δεύτερο στρώμα του LSTM, ή αλλιώς και πύλη εισόδου, σε αυτό το σημείο επιλέγονται οι καινούργιες είσοδοι και σε τι ποσοστό θα επηρεάσουν την κατάσταση του νευρώνα η κάθε μία

Αρχικά γίνεται η χρήση μίας σιγμοειδούς συνάρτησης για την επιλογή των τιμών που θα ενημερωθούν :

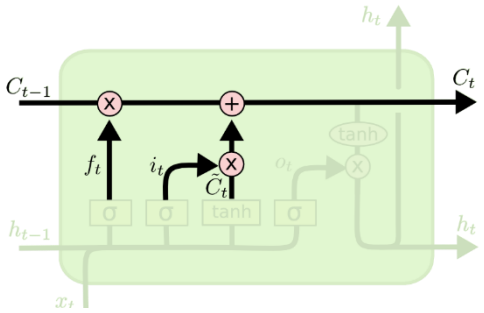
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \text{ (3.40)}$$

και στην συνέχεια μία συνάρτηση υπερβολικής εφαπτομένης δημιουργεί ένα διάνυσμα με τιμές που πιθανόν να ενημερώσουν την κατάσταση του νευρώνα:

$$\widetilde{C_t} = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \text{ (3.41)}$$

Με την $\widetilde{C_t}$ έχουμε τις τιμές που θέλουμε να ενημερώσουν την κατάσταση του νευρώνα και σε ποιο βαθμό επίσης, για να παραχθεί η καινούργια κατάσταση C_t. Στο παράδειγμα που αναφέρθηκε αυτό είναι το βήμα που θα ξεχαστεί η πληροφορία του προηγούμενου γένους και θα προστεθεί καινούργια πληροφορία. Για να υπολογιστεί κατάσταση του νευρώνα C_t η f_t, που είχε υπολογισθεί στο πρώτο βήμα, πολλαπλασιάζεται με την προηγούμενη κατάσταση του νευρώνα και στην συνέχεια το γινόμενο των i_t και $\widetilde{C_t}$ προστίθεται στο αποτέλεσμα:

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \text{ (3.42)}$$



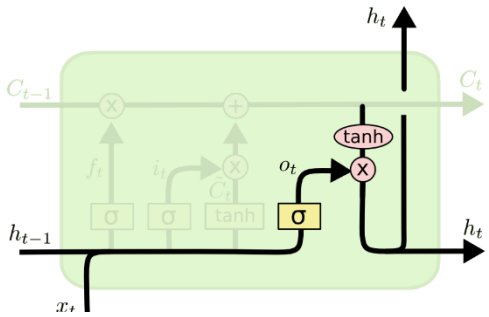
Σχήμα 15 : Η ανανέωση της κατάστασης του νευρώνα

Τελικά πρέπει να αποφασιστεί η έξοδος του νευρώνα η οποία βασίζεται σε μία φιλτραρισμένη εκδοχή της κατάστασής του. Σκοπός της πύλης εξόδου είναι να αποφασιστεί ποιο κομμάτι πληροφορίας θα χρησιμοποιηθεί από την τωρινή κρυφή κατάσταση για την ενημέρωση της κατάστασης του νευρώνα την επόμενη χρονική στιγμή t. Το στρώμα εξόδου o_tέχει παρόμοια είσοδο με το στρώμα εισόδου, αλλά με το δικό του διάνυσμα βαρών:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (3.43)$$

ωστόσο στην συνέχεια πολλαπλασιάζεται με τη τωρινή κατάσταση του νευρώνα για την δημιουργία της καινούργιας κρυφής κατάστασης:

$$h_t = o_t * \tanh(C_t) \quad (3.44)$$

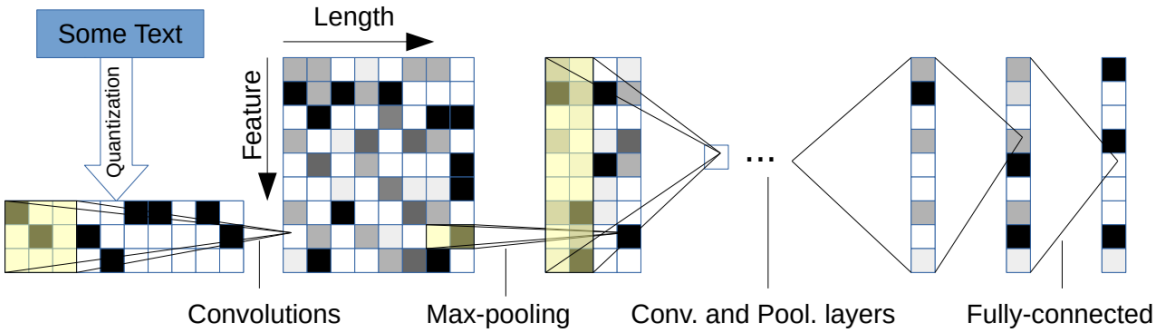


Σχήμα 16 : Πύλη εξόδου

3.8 Γνωστά Δίκτυα NLP

Character Level CNN

Το δίκτυο αναπτύχθηκε το 2015 από τον Yann LeCun και χρησιμοποιούσε τους χαρακτήρες των προτάσεων για την σωστή κατάταξη τους. Το δίκτυο δεχόταν μόνον 70 χαρακτήρες οι οποίοι αποτελούνταν από 26 γράμματα, 10 ψηφία, και 33 ακόμη χαρακτήρες και το κενό.



Σχήμα 17 : Αρχιτεκτονική Character lever CNN

Δύο διαφορετικά δίκτυα σχεδιάστηκαν το ένα πιο βαθύ από το άλλο, και στα 2 γίνεται χρήση 6 κρυφών συνελικτικών στρωμάτων όπου στα 2 πρώτα και στο τελευταίο γίνεται χρήση pooling με μέγεθος 3. Στο μεγάλο δίκτυο γίνεται χρήση από 1024 νευρώνες σε κάθε στρώμα ενώ στο μικρό 256.

Layer	Large Feature	Small Feature	Kernel	Pool
1	1024	256	7	3
2	1024	256	7	3
3	1024	256	3	N/A
4	1024	256	3	N/A
5	1024	256	3	N/A
6	1024	256	3	3

Σχήμα 18 : Αρχιτεκτονική CNN στρωμάτων

Στο τέλος για την εξαγωγή των τάξεων γίνεται χρήση 3 πλήρως συνδεδεμένων στρωμάτων.

Layer	Output Units Large	Output Units Small
7	2048	1024
8	2048	1024
9	Depends on the problem	

Σχήμα 19 : Αρχιτεκτονική Πλήρως συνδεδεμένου δικτύου

Η εκπαίδευση έγινε από κείμενα του συνόλου δεδομένων από AG και Sogou και κριτικές του Yelp, Amazon και Yahoo. Κατάφερε να πετύχει σφάλμα 4.93% στις κριτικές του Amazon και καλύτερα αποτελέσματα από τα μοντέλα όπως τα BoW, ngrams, Bag-of-Means και LSTM σε κριτικές από Yahoo

και Yelp. Από τα αποτελέσματα φάνηκε πως τα CNN ίσως να είναι καλύτερα στην αντιμετώπιση προβλημάτων σχετικά με κείμενα γραμμένα χωρίς ιδιαίτερη προσοχή καθώς ένας άνθρωπος γράφει πολύ πιο πρόχειρα μία κριτική στο Amazon παρά ένα άρθρο.

BoW

Το Bag-of-Words είναι ένας αλγόριθμος που μετράει πόσες φορές εμφανίστηκε σε ένα κείμενο μία λέξη και μετά χρησιμοποιεί αυτήν την πληροφορία σε ένα απλό νευρωνικό δίκτυο για την ταξινόμηση των κειμένων. Το BoW δημιουργεί μία λίστα με λέξεις και τον αριθμό εμφάνισης σε κάθε κείμενο, μετατρέποντας τις λέξεις ουσιαστικά σε διανύσματα όπου κάθε γραμμή είναι λέξη και κάθε στήλη είναι κείμενο. Κάθε κείμενο καταλήγει να αντιπροσωπεύεται από στήλες ίσου μεγέθους, από τις λέξεις χωρίς καμία πληροφορία σχετικά με την θέση των λέξεων.

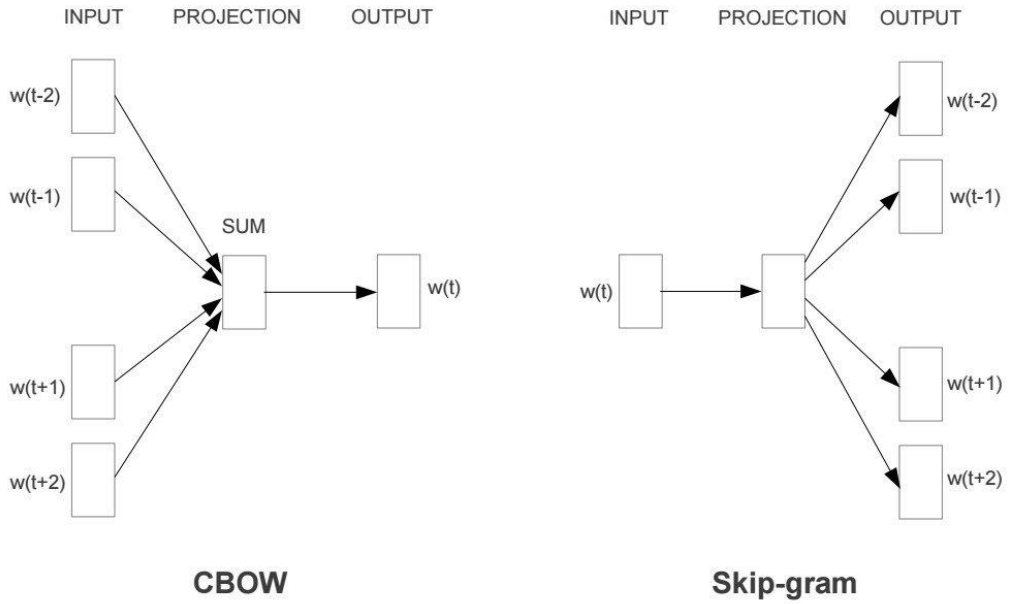
	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

Σχήμα 20 : Bag-of-Words

Πριν εισαχθούν στο νευρωνικό δίκτυο κάθε διάνυσμα με τις μετρήσεις των λέξεων κανονικοποιείται με τέτοιο τρόπο ώστε όλα τα στοιχεία του διανύσματος να έχουν άθροισμα μονάδα. Οπότε η συχνότητα της κάθε λέξης μετατρέπεται στην πιθανότητα να εμφανιστεί αυτή η λέξη στο κείμενο, μεγαλύτερες πιθανότητες ενεργοποιούν νευρώνες στο δίκτυο και επηρεάζουν την ταξινόμηση του κειμένου.

Word2Vec

Το μοντέλο Word2Vec αναπτύχθηκε από την Google το 2013 και έχει πολύ καλύτερα αποτελέσματα από προηγούμενες προσπάθειες στον τομέα της επεξεργασίας φυσικής γλώσσας. Αποτελείται από 2 στρώματα μόνον στρώματα και ακολουθεί 2 διαφορετικές προσεγγίσεις στην αναζήτηση για το γενικό πλαίσιο μίας πρότασης ή μίας λέξης. Στην πρώτη τεχνική που ακολουθεί(CBOW), δέχεται ως εισόδους τις λέξεις μίας πρότασης εκτός από μία, και δεδομένων των εισόδων προσπαθεί να προσαρμόσει τα βάρη του ώστε να προβλέπει την λέξη που αγνοείται. Στην δεύτερη τεχνική γίνεται το ανάποδο, δέχεται δηλαδή ως είσοδο μία λέξη και προσπαθεί να υπολογίσει τα συμφραζόμενα κάτω από τα οποία βρίσκεται.



Σχήμα 21 : Word2Vec

Σκοπός αυτού του μοντέλου είναι να ομαδοποιήσει λέξεις με παρόμοια σημασία στις διαστάσεις των διανυσμάτων που τις αντιπροσωπεύουν. Δεδομένου πως θα χρησιμοποιηθούν αρκετά δεδομένα για την εκπαίδευση το μοντέλο μπορεί να κάνει αρκετά ακριβείς προβλέψεις για την σημασία μίας λέξης βασισμένες στις προηγούμενες εμφανίσεις της. Εκπαιδεύτηκε στο σύνολο δεδομένων Google Vocab και κατάφερε να βρει συσχετίσεις όπως στις οποίες ισχύουν σχέσεις όπως ότι είναι το διάνυσμα "άντρας" για το διάνυσμα "αγόρι" είναι η "γυναίκα" για το "κορίτσι" αλλά και περισσότερο πολύπλοκες που δεν είναι η μία καθρέπτης της άλλης όπως "Πρωθυπουργός" – "Δύναμη" = "Πρόεδρος" και "Άνθρωπος" – "Ζώο" = "Ηθική".

B.E.R.T (Bidirectional Encoder Representations from Transformers)

Το μοντέλο με όνομα B.E.R.T αποτελεί αυτή την στιγμή το μοντέλο με την καλύτερη επίδοση σε 2 διαγωνισμούς σχετικά με την ανάλυση της φυσικής γλώσσας. Στον διαγωνισμό SQuAD v1.1 κατάφερε επίδοση 93.2%, ενώ την ίδια στιγμή η ανθρώπινη επίδοση είναι στο 91.2%, και στο Glue Benchmark κατάφερε να βελτιώσει την καλύτερη μέση επίδοση κατά 6.7% με απόδοση 81,9%, ο διαγωνισμός αυτός αποτελείται από 9 διαφορετικές δοκιμασίες σε προβλήματα επεξεργασίας φυσικής γλώσσας. Για να κατανοήσει κάποιος καλύτερα το μοντέλο αρκεί να δει το όνομά του, και όπως υποδηλώνεται χρησιμοποιεί αμφίδρομες κωδικοποιημένες αναπαραστάσεις από μετασχηματιστές για να επιτύχει τα αποτελέσματά του. Το μεγαλύτερο πλεονέκτημα του μοντέλου είναι πως καταφέρνει να μάθει συσχετίσεις μεταξύ των λέξεων σε μία πρόταση οι οποίες δεν είναι απαραίτητα κοντά η μία στην άλλη, όμως αρχικά καλό θα ήταν να δούμε πως προ επεξεργάζεται τα δεδομένα το μοντέλο πριν δημιουργήσει συσχετίσεις μεταξύ τους. Χρησιμοποιεί μία τεχνική που ονομάζεται ακολουθία σε ακολουθία(seq2seq). Κάθε πρόταση αποτελείται από μία ακολουθία από σημεία, στην πιο απλή

περίπτωση τα σημεία αυτά είναι οι λέξεις. Συνήθως γίνεται η χρήση μίας συνάρτησης για τον διαχωρισμό της πρότασης σε σημεία λαμβάνοντας υπόψιν την θέση της λέξης μέσα στην πρόταση και τα σημεία στίξης, μία από τις πιο γνωστές βιβλιοθήκες είναι η nltk. Ωστόσο η μάθηση πάνω σε σημεία που προέρχονται από λέξεις έχει κάποια μειονεκτήματα. Ένα σύνηθες πρόβλημα είναι η επιλογή λέξεων από ένα πολύ μεγάλο λεξικό, για παράδειγμα αν το μοντέλο προσπαθεί να μαντέψει την επόμενη λέξη από ένα σύνολο λέξεων που ο αριθμός του είναι απαγορευτικός, αναγκαστικά πρέπει να τηρηθεί κάποιο όριο από πόσες λέξεις θα επιλέξει ο αλγόριθμος. Ένα ακόμα πρόβλημα είναι πως το μοντέλο τελικά δεν μαθαίνει πως λέξεις που έχουν ίδιο κορμό έχουν και κοντινή σημασία, για παράδειγμα η λέξη “μαθαίνει” με την λέξη “μάθηση” θεωρούνται διαφορετικές ενώ έχουν κοινή ρίζα. Ένας τρόπος για την αντιμετώπιση του πρώτου προβλήματος είναι η χρήση υπολέξεων. Για παράδειγμα η λέξη ”μαθαίνει” μπορεί να χωριστεί σε “μαθ” και ”αινει”, ενώ η λέξη “μάθηση” σε “μαθ” και “ηση”, αυτό επιτρέπει στο μοντέλο να μπορεί να μάθει καινούργιες λέξεις πιο εύκολα αλλά και να έχει μικρότερο λεξικό. Επειδή δεν υπάρχει κάποιο μέρος αυτού του δικτύου το οποίο να κρατά την πληροφορία της θέσης πρέπει να προστεθεί τεχνητά επιπλέον πληροφορία σχετικά με την θέση των δεδομένων στην ακολουθία. Αυτή η δουλειά επιτυγχάνεται από την μονάδα κωδικοποίησης θέσης. Η πληροφορία θα μπορούσε να είναι απλά ένας αύξων αριθμός ωστόσο δημιουργούνται προβλήματα από αυτήν την προσέγγιση, όπως κατά την διαδικασία της γενίκευσης το μοντέλο μπορεί να συναντούσε προτάσεις μεγαλύτερες απ’ ότι κατά την εκπαίδευση. Μία άλλη λύση θα ήταν οι τιμές να είναι ανάμεσα σε ένα εύρος τιμών όπως $[0,1]$ όμως έτσι δεν θα ξέραμε πόσες λέξεις υπάρχουν σε όλο το διάστημα και η διαφορά ανάμεσα σε 2 θα ήταν μεταβλητή. Η λύση που προτάθηκε είναι πως η πληροφορία πρέπει να αναπαρασταθεί από ένα διάνυσμα και όχι μόνο μία τιμή. Αν θεωρηθεί πως p_t είναι το διάνυσμα για την θέση t και f η συνάρτηση που παράγει το p_t ορίζεται ως εξής:

$$p_t = f(t)^i = \begin{cases} \sin(w_k \cdot t), & i = 2k \\ \cos(w_k \cdot t), & i = 2k+1 \end{cases} \quad (3.45)$$

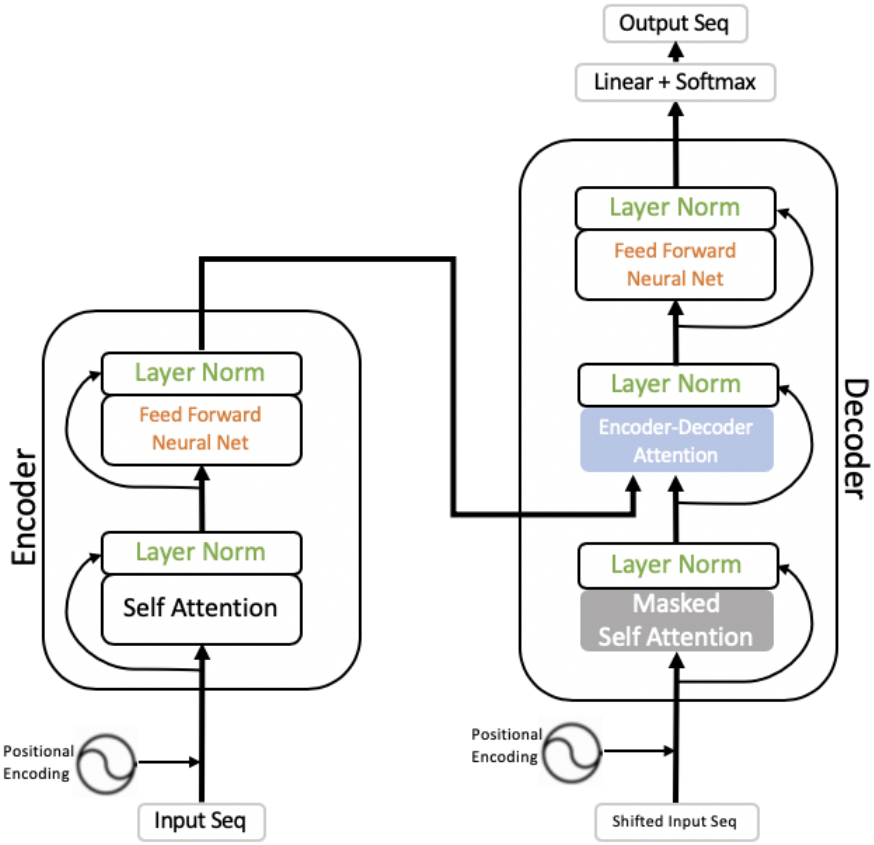
Όπου

$$\mathrm{w}_k = \frac{1}{10000^{\frac{2k}{d}}} \quad (3.46)$$

με d να είναι οι διαστάσεις του διανύσματος κωδικοποίησης.

Όπως φαίνεται από την εξίσωση η συχνότητα μειώνεται κατά μήκος των διαστάσεων του διανύσματος δημιουργώντας μία γεωμετρική ακολουθία από το 2π μέχρι το $10000 \cdot 2\pi$ στο μήκος κύματος. Για να αντιληφθεί κανείς πως η ακολουθία ημιτόνων και συνημίτονων μπορεί να αναπαραστήσει θέση αρκεί κανείς θα σκεφτεί πως αναπαριστώνται οι αριθμοί σε δυαδική μορφή. Το λιγότερο σημαντικό bit αλλάζει τιμή κάθε αριθμό, το δεύτερο λιγότερο σημαντικό bit κάθε 2 αριθμούς και ούτω καθεξής. Ωστόσο η χρήση πραγματικών αριθμών θα αποτελούσε σπατάλη χώρου σε έναν κόσμο με αριθμούς κινητής υποδιαστολής οπότε χρησιμοποιούνται οι συνεχείς αντικαταστάτες τους, οι συναρτήσεις ημιτόνου και συνημίτονου. Στην συνέχεια χρησιμοποιείται ένα στρώμα κανονικοποίησης, του οποίου η σημαντική λειτουργία είναι πως επιτελεί κανονικοποίηση των εισόδων σε όλα τα χαρακτηριστικά, αντίθετα με την κανονικοποίηση παρτίδας στην οποία

γίνεται κανονικοποίηση κάθε χαρακτηριστικού σε ολόκληρη την παρτίδα.



Σχήμα 22 : Κωδικοποιητής - Αποκωδικοποιητής του μοντέλου Bert

Επίσης κάθε μονάδα κωδικοποιητή – αποκωδικοποιητή όπως φαίνεται στο σχήμα 20 έχει και ένα πλήρως συνδεδεμένο δίκτυο στο τέλος, Το δίκτυο αυτό αποτελείται από 2 στρώματα με την συνάρτηση ενεργοποίησης ReLU ανάμεσά τους. Μία ακόμα τεχνική που χρησιμοποιείται από το μοντέλο είναι η αυτό-προσοχή(self-attention). Στην τεχνική της προσοχής γίνεται χρήση 2 προτάσεων όπου μετατρέπονται σε πίνακα με την μία να σχηματίζει τις στήλες και την άλλη τις γραμμές. Με αυτόν τον τρόπο δημιουργούνται συνδέσεις μεταξύ των προτάσεων αναγνωρίζοντας έτσι το σχετικό περιεχόμενο, διαδικασία που έχει πολύ καλά αποτελέσματα στην μετάφραση προτάσεων από μία γλώσσα σε μία άλλη. Ωστόσο δεν είναι αναγκαία συνθήκη η χρήση διαφορετικών προτάσεων και αυτό εκμεταλλεύεται η αυτό-προσοχή, κατά την οποία γίνεται χρήση της ίδιας πρότασης και στις 2 διαστάσεις του πίνακα και μαθαίνοντας έτσι συσχετίσεις μεταξύ λέξεων της ίδιας πρότασης. Για παράδειγμα το πρόβλημα της σύνδεσης των αντωνυμιών με τις λέξεις στις οποίες αναφέρονται είναι πολύ παλιό και το οποίο έχει οδηγήσει στην δημιουργία της ακόλουθης κωμικής ακολουθίας που μιμείται την κραυγή ερώτημα και την απάντηση σε μία διαδήλωση:

$$ΤΙ\ ΘΕΛΟΥΜΕ;$$

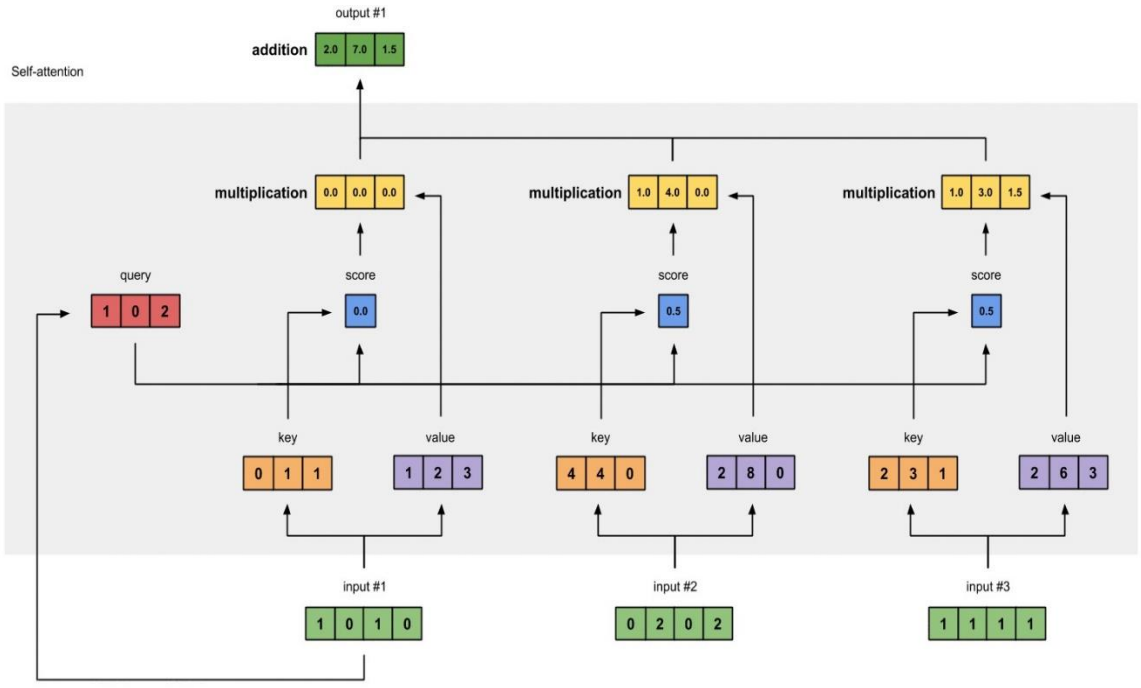
$$Επεξεργασία\ φυσικής\ γλώσσας!$$

$ΠΟΤΕ\ ΤΗΝ\ ΘΕΛΟΥΜΕ;$

$Συγνώμη, πότε\ θέλουμε\ τι;$

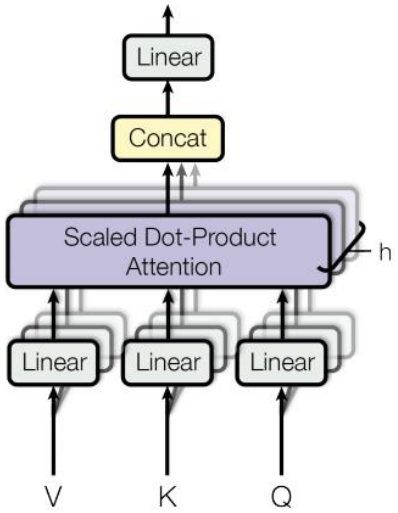
Ένα τεχνητό νευρωνικό δίκτυο το οποίο χρησιμοποιεί μεθόδους προσοχής και αυτό-προσοχής μπορεί να μάθει σε ποιο κομμάτι κειμένου αναφέρεται το «$τι$». Δηλαδή μπορεί να ξεχωρίσει τον θόρυβο που είναι η μεταξύ τους λέξεις και αντ' αυτού να συνδέσει 2 λέξεις που σχετίζονται μεταξύ τους χωρίς οι ίδιες να δείχνουν με κάποιο τρόπο η μία στην άλλη. Η προσέγγιση αυτή είναι πολύ διαφορετική σε σχέση με την αντίθετη στο ρεύμα ροή των ακολουθιακών νευρωνικών δικτύων και την προσεγγιστική λογική των συνελικτικών δικτύων.

Το πετυχαίνει αυτό χωρίζοντας το διάνυσμα εισόδου σε 3 διανύσματα, κάθε ένα από τα οποία προκύπτει από το γινόμενο της εισόδου (I) με ένα διαφορετικό διάνυσμα βαρών. Τα διανύσματα αυτά ονομάζονται Κλειδί (K), Τιμή (V) και Ερώτημα (Q). Στην συνέχεια υπολογίζονται η απόδοση προσοχής για κάθε είσοδο, σχετικά με τον εαυτό τους και τις υπόλοιπες εισόδους και αυτό γίνεται με τον υπολογισμό του εσωτερικού γινομένου του K με το Q, εδώ να αναφερθεί πως το Q που χρησιμοποιείται σε αυτό το βήμα είναι της εισόδου. Στις αποδόσεις που υπολογίστηκαν παραπάνω εφαρμόζεται η συνάρτηση υπερβολικής εφαπτομένης και το αποτέλεσμα πολλαπλασιάζεται με το V. Τελικά προστίθενται τα διανύσματα που προέκυψαν από το τελευταίο βήμα και υπολογίζεται η αυτό-προσοχή της εισόδου. Οι διαστάσεις των Q και V πρέπει πάντα να είναι ίδιες λόγω του εσωτερικού γινομένου που υπολογίζεται το score ωστόσο το V μπορεί να είναι διαφορετικό σε μέγεθος, άρα το αποτέλεσμα θα ακολουθεί τις διαστάσεις του V.



Σχήμα 23 : Διαδικασία της αυτό-προσοχής με παράδειγμα με 3 εισόδους, όπου στο τέλος υπολογίζεται η προσοχή της πρώτης εισόδου

Ο μετατροπέας χρησιμοποιεί ύστερα την τεχνική του Multi-Head Attention, αυτό σημαίνει πως υπολογίζει την προσοχή h φορές με διαφορετικούς πίνακες βαρών και στη ενώνει τα αποτελέσματα. Το αποτέλεσμα κάθε μίας από τους παράλληλους υπολογισμούς ονομάζεται κεφαλή(head).



Σχήμα 24 : Υπολογισμός παράλληλων κεφαλών και ένωσή τους.

Αυτό έχει ως αποτέλεσμα ένα πίνακα με διαστάσεις $(input\_length) \cdot (h * d\_v)$. Στην συνέχεια ένα γραμμικό στρώμα με βάρη W^0 με διαστάσεις $(h * d\_v) \cdot (emdedding\_dimension)$ θα εφαρμοστεί με ένα τελικό αποτέλεσμα με διαστάσεις $(input\_length) \cdot (emdedding\_dimension)$:

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^0 \quad (3.47)$$

Όπου

$$head_i = Attention(QW_i, KW_i, VW_i) \ (3.48)$$

Κάθε κεφαλή μπορεί να χαρακτηριστεί από 3 διαφορετικές προβολές που δίνονται από τους πίνακες:

$$W_i^K[d_{model}, d_k]$$
$$W_i^Q[d_{model}, d_k] \ (3.49)$$
$$W_i^V[d_{model}, d_v]$$

Για να υπολογιστεί η κεφαλή θα χρησιμοποιηθεί η είσοδος X και θα προβληθεί επάνω στους πίνακες των βαρών:

$$XW_i^K = K_i[input\_length, d_k]$$
$$XW_i^Q = Q_i[input\_length, d_k] \quad (3.50)$$
$$XW_i^V = V_i[input\_length, d_v]$$

Οπότε όταν έχουν υπολογιστεί οι τιμές K, Q, V η προσοχή μπορεί να υπολογιστεί ως:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (3.51)$$

Τα Q και K είναι διαφορετικές προβολές των εισόδων οπότε μπορεί κανείς να αντιληφθεί πως το γινόμενό των προβολών είναι ένας τρόπος για να μετρηθεί η ομοιότητά τους.

Στην πλευρά του αποκωδικοποιητή χρησιμοποιείται η μασκαρισμένη αυτό-προσοχή(Masked Self Attention). Σε κάθε σημείο μία λέξη μπορεί να εξαρτάται και από τις προηγούμενες και τις επόμενες μίας πρότασης. Για παράδειγμα στην πρόταση «Είδα μία _____ να κυνηγάει ένα ποντίκι» θα βάζαμε την λέξη «γάτα» ως την περισσότερο πιθανή. Οπότε καθώς γίνεται το encoding πρέπει να γνωρίζει όλη την πρόταση ενώ κατά την διαδικασία του decoding, όπου προσπαθεί να προβλέψει την επόμενη λέξη στην πρόταση, λογικά, δεν θα πρέπει να ξέρει ποιες λέξεις έρχονται μετά, αυτό

επιτυγχάνεται με τον πολλαπλασιασμό των εισόδων των επόμενων τιμών με 0 ώστε η πρόβλεψη να βασίζεται μόνο στις προηγούμενες λέξεις. Μία ακόμη τεχνική εκπαίδευσης είναι η πρόβλεψη επόμενης πρότασης όπου το μοντέλο μαθαίνει σχέσεις μεταξύ προτάσεων. Στην διαδικασία της μάθησης το μοντέλο λαμβάνει ζευγάρια από προτάσεις ως είσοδο και μαθαίνει να προβλέπει αν η δεύτερη πρόταση από το ζευγάρι είναι πραγματικά η ακόλουθη πρόταση μέσα στην αρχική παράγραφο. Κατά την εκπαίδευση του μοντέλου και οι 2 τεχνικές εκπαιδεύονται ταυτόχρονα ελαχιστοποιώντας έτσι την συνάρτηση κόστους των 2 στρατηγικών. Όπως φαίνεται στο σχήμα 18 το ενδιάμεσο στρώμα προσοχής δέχεται 2 εισόδους, η μία προέρχεται από το επίπεδο του Masked Self-Attention που βρίσκεται στον ίδιο τον αποκωδικοποιητή και η άλλη από την έξοδο του κωδικοποιητή. Αντί να παραχθούν τα Q, K και V από την είσοδο του Decoder όπως έγινε στα προηγούμενα στρώματα Self-Attention, θα αξιοποιηθεί και η τελική έξοδος του Encoder E και η έξοδος του Masked Self-Attention D. Το E είναι ένας πίνακας με διαστάσεις *input_length * embedding_dimension* και το D, πίνακας με διαστάσεις *target_length * embedding_dimension,* από την έξοδο του στρώματος κανονικοποίησης μετά το Masked Self-Attention. Θα χρησιμοποιηθούν πάλι πίνακες βαρών με διαστάσεις παρόμοιες όπως στο στρώμα αυτό-προσοχής κατά την διαδικασία της κωδικοποίησης $W_i^K[d_{model}, d_k], W_i^Q[d_{model}, d_k], W_i^V[d_{model}, d_v]$ όμως αυτή τη φορά η προβολή που παράγει το Q_i θα γίνει με την χρήση των εισόδων D ενώ για τον υπολογισμό των K και V θα χρησιμοποιηθεί το E:

$$DW_i^Q = Q_i[target\_length, d_k]$$
$$EW_i^K = K_i[input\_length, d_k] \quad (3.51)$$

$$XW_i^V = V_i[input\_length, d_v]$$

Για κάθε κεφαλή $i = 1, \dots, h$.

Οπότε όπως έχει γίνει αντιληπτό στο μοντέλο οι κωδικοποιητές μετατρέπουν τις εισόδους σε καλά ορισμένα διανύσματα, γι' αυτό το λόγω στον BERT base 12 γίνεται η χρήση 12 κεφαλών h, 6 στρωμάτων κωδικοποίησης που το κάθε στρώμα αποτελείται από τις μονάδες αυτό-προσοχής και πλήρους συνδεδεμένου δικτύου, ενώ στην συνέχεια υπάρχουν επίσης 6 στρώματα αποκωδικοποιητών σε κάθε ένα από τα οποία αντιστοιχεί μία μονάδα μασκαρεμένης αυτό-προσοχής, μία μονάδα αυτό-προσοχής και ένα πλήρως συνδεδεμένο δίκτυο. Στον BERT Large 24 χρησιμοποιούνται 12 στρώματα κωδικοποίησης και αποκωδικοποίησης με 16 διαφορετικές κεφαλές και 340 εκατομμύρια παραμέτρους. Ο BERT έχει επιτύχει πολύ καλά αποτελέσματα σε κάποια συγκεκριμένα προβλήματα όμως από τότε που αποδείχθηκε πως η χρήσης Transformers επιτυγχάνει καλύτερα αποτελέσματα σε δοκιμασίες NLP πολύ τον αντέγραψαν. Το Facebook κάνει χρήση του RoBERTa το οποίο ακολουθεί την ίδια στρατηγική με τον BERT. Για να βελτιώσουν την διαδικασία της μάθησης αφαίρεσαν την πρόβλεψη επόμενης πρότασης και εισήγαγαν την τεχνική δυναμικού μασκαρέματος ώστε σε κάθε εποχή διαφορετικό κομμάτι της εισόδου να είναι κρυφό. Ο DistilBERT αναπτύχθηκε από το HugginFace, έχει παρόμοια αρχιτεκτονική απ' ότι ο BERT και πετυχαίνει το ίδιο ποσοστό ευστοχίας στο GLUE dataset χρησιμοποιώντας όμως τις μισές παραμέτρους από τον BERT. Το σκεπτικό είναι πως αφού ένα μεγάλο πλήρως συνδεδεμένο δίκτυο έχει εκπαιδευτεί, οι διανομές πιθανοτήτων των εξόδων του μπορούν να προσεγγιστούν από ένα μικρότερο δίκτυο. Ο XLM/mBERT που αναπτύχθηκε πάλι από το Facebook χρησιμοποιεί την μέθοδο κωδικοποίησης byte-pair encoding

και μία διαγλωσσική τεχνική μάθησης με τον BERT για να μάθει σχέσεις μεταξύ λέξεων διαφορετικών γλωσσών. Το μοντέλο παρουσιάζει πολύ καλά αποτελέσματα σε προβλήματα ταξινομήσεων πολλαπλών γλωσσών και βελτιώνει πολύ τα ποσοστά στην διαδικασία της μετάφρασης όταν για αρχικές τιμές χρησιμοποιείται ένα προ-εκπαιδευμένο δίκτυο. Ομαδικά δημιουργημένο από την ερευνητική ομάδα της Google το ερευνητικό ινστιτούτο της Toyota, ο ALBERT αποτελεί την βελτιωμένη έκδοση του BERT η οποία είναι πιο μικρή ελαφριά και περισσότερο ικανή από τον ίδιο. Ο ALBERT χρησιμοποιεί τις ίδιες παραμέτρους σε όλα τα στρώματα του, που σημαίνει πως όλες οι παράμετροι των πλήρων συνδεδεμένων δικτύων και της προσοχής μοιράζονται. Επίσης όσον αφορά την εκπαίδευση ο ALBERT έχει την δικιά του μέθοδο εκπαίδευσης που ονομάζεται SOP(Sentence Over Prediction), αντίθετα με την πρόβλεψη επόμενης πρότασης, η οποία σύμφωνα με τους συγγραφείς συγκρούει την πρόβλεψη της έννοιας και της συνοχής της επόμενης πρότασης.

Επίλογος

Σε αυτό το κεφάλαιο έγινε μία ιστορική αναφορά στα πρώτα βήματα των νευρωνικών δικτύων και στην συνέχεια θα εξετάστηκε το αναγκαίο θεωρητικό υπόβαθρο και οι βάσεις πάνω στις οποίες βασίστηκαν και δημιουργήθηκαν τα τεχνητά νευρωνικά δίκτυα. Τέλος έγινε επεξήγηση της λειτουργίας των CNN και RNN καθώς και μερικές από τις περισσότερο επιτυχημένες υλοποιήσεις τους.

Κεφάλαιο 4

Frameworks

Εισαγωγή

Τα Frameworks είναι πολύ σημαντικά καθώς επιτρέπουν στους προγραμματιστές να εστιάσουν στην βελτιστοποίηση και ανάπτυξη ενός αλγορίθμου και όχι σε βασικά τμήματα χαμηλού επιπέδου όπως την παρουσίαση ενός συστήματος που απλά δουλεύει. Σε αυτό το κεφάλαιο θα γίνει μία αναφορά στα Frameworks που χρησιμοποιήθηκαν στην εργασία αυτή και σε κάποια τα οποία είναι ευρέως γνωστά και έχουν βοηθήσει την ανάπτυξη των τεχνητών νευρωνικών δικτύων.

4.1 Frameworks

Tensoflow

Αδιαμφισβήτητα ένα από τα καλύτερα και πιο διαδεδομένα frameworks, το Tensorflow έχει ενσωματωθεί και χρησιμοποιείται από εταιρίες γίγαντες όπως η Airbus, Twitter, IBMκαι πολλές ακόμη κυρίως λόγω της ευλύγιστης αρχιτεκτονικής του συστήματός. Η πιο διαδεδομένη χρήση του έχει να κάνει με την εφαρμογή του στην μετάφραση της google σε συνδυασμό με την επεξεργασία φυσικής γλώσσας, ταξινόμηση ή περίληψη κειμένων, αναγνώριση και πρόβλεψη κειμένων, ομιλίας ή εικόνων. Το Tensorflow δημιουργήθηκε αρχικά από την google για εσωτερική χρήση και έγινε πρώτη φορά διαθέσιμο από το δημόσια το 2015 κάτω από την άδεια ανοιχτού λογισμικού Apache

2.0. Η πρώτη επίσημη να έκδοση βγήκε το 2016 και χρησιμοποιούσε μία ειδική μονάδα επεξεργασίας την οποία ονόμασε tensor και εκφράζει στατιστικά διαγράμματα ροής δεδομένων και το όνομά του προκύπτει από τις διεργασίες που εκτελούνται από τα νευρωνικά δίκτυα πάνω σε αυτούς τους πίνακες. Το framework είναι μπορεί και εκμεταλλεύεται παράλληλη επεξεργασία καθώς μπορεί να τρέχει πάνω σε πολλούς πυρήνες, αλλά είναι πιο αποδοτικό όταν χρησιμοποιείται με κάρτες γραφικών λόγω των μαθηματικών πράξεων αλλά και επειδή μπορεί να εκμεταλλεύεται τεχνολογίες όπως η CUDA της NVIDIA. Υποστηρίζει μεγάλη πληθώρα από τρίτα frameworks και διαφορετικές γλώσσες προγραμματισμού όπως Python, C++ και R.

Caffe

Το Caffe είναι ένα framework βαθιάς μηχανικής μάθησης και υποστηρίζεται από διεπαφές όπως C, C++, Python, MATLAB, μέχρι και την γραμμή εντολών των windows. Έχει γίνει ιδιαίτερα δημοφιλές λόγω της ταχύτητας και την εύκολης εφαρμογής του στην μοντελοποίηση συνελικτικών νευρωνικών δικτύων. Το μεγαλύτερο πλεονέκτημά της χρήσης της βιβλιοθήκης για την C++ έχει να κάνει με την πρόσβαση σε ένα μεγάλο σύνολο από ήδη εκπαιδευμένα δίκτυα από το ‘Caffe Model Zoo’ τα οποία είναι προ-εκπαιδευμένα και μπορούν να χρησιμοποιηθούν κατευθείαν. Το Caffe είναι πολύ δημοφιλές για την επεξεργασία εικόνων ωστόσο δεν έχει τόσο λεπτομερή στρώματα όπως το Tensorflow και η υποστήριξη για επαναλαμβανόμενα δίκτυα είναι φτωχή και πρέπει να γίνει σε χαμηλό επίπεδο με κάποια γλώσσα προγραμματισμού.

Microsoft Cognitive Toolkit

Δημοφιλές λόγω της εύκολης εκπαίδευση και του συνδυασμού μοντέλων μεταξύ κατανεμητών το Microsoft Cognitive Toolkit (πρωτύτερα γνωστό ως CNTK) είναι ένα framework ανοικτής πηγής για την εκπαίδευση βαθιών νευρωνικών δικτύων. Αποδίδει αποτελεσματικά συνελικτικά δίκτυα για εκπαίδευση σε εικόνες, κείμενο και ομιλία ενώ παρόμοια με το Caffee υποστηρίζει διεπαφές όπως η Python, C++ και γραμμής εντολών. Λόγω του υψηλού επιπέδου αφαιρετικότητας είναι εύκολη η μοντελοποίηση νευρωνικών δικτύων και θεωρείται πως έχει καλύτερη απόδοση και επεκτασιμότητα από άλλα frameworks, όπως το Tensorflow όταν λειτουργεί σε διαφορετικά μηχανήματα. Σε σύγκριση με το Caffe κατά την δημιουργία πολύπλοκων δικτύων οι χρήστες δεν χρειάζεται να υλοποιήσουν στρώματα σε κάποια γλώσσα προγραμματισμού λόγω των καλά δομημένων δομικών στοιχείων του. Το CNTK υποστηρίζει τόσο CNN όσο και RNN και είναι οπότε ικανό για την διαχείριση προβλημάτων εικόνων, κειμένου και ομιλίας.

Pytorch/Torch

Το Torch είναι ένα επιστημονικό υπολογιστικό framework που παρέχει υποστήριξη για αλγορίθμους μηχανικής μάθησης. Είναι βασισμένο στη γλώσσα προγραμματισμού Lua και χρησιμοποιείται από εταιρίες όπως το Facebook,Twitter και Google. Παρέχει υποστήριξη για βιβλιοθήκες CUDA και C/C++ για την παράλληλη επεξεργασία και φτιάχτηκε με σκοπό να είναι ευέλικτο και να μπορεί να μπορεί να κλιμακωθεί σε μεγάλο βαθμό. Τον τελευταίο καιρό το Pytorch έχει ενσωματωθεί σε μεγάλο βαθμό από την κοινότητα της βαθιάς μηχανικής μάθησης και θεωρείται αντίπαλο δέος του Tensorflow. Το

PyTorch είναι βασικά μία πόρτα για την χρήση του Torch και δημιουργία βαθιών νευρωνικών δικτύων που είναι πολύ ψηλά σε βαθμό πολυπλοκότητας. Αντίθετα με το Torch τρέχει σε python οπότε οποιοσδήποτε που γνωρίζει σε βασικό βαθμό την γλώσσα μπορεί να δημιουργήσει τα δικά του μοντέλα. Δεδομένης της αρχιτεκτονικής του PyTorch η διαδικασία για την δημιουργία νευρωνικών δικτυών είναι περισσότερο απλή και διαφανής σε σχέση με το Torch.

MxNet

Το MxNet είναι ένα framework που υποστηρίζει Python, R, C++ και Julia και σχεδιάστηκε εξολοκλήρου με μέριμνα για υψηλή απόδοση, ελαστικότητα και παραγωγικότητα. Το μεγαλύτερο πλεονέκτημά του είναι πως δίνει την δυνατότητα για ανάπτυξη νευρωνικών δικτύων σε διαφορετικές γλώσσες προγραμματισμού χωρίς την ανάγκη για εκμάθηση μίας καινούργιας από την αρχή. Έχει γραφτεί σε C++ και CUDA οπότε υπάρχει υποστήριξη για πολλές διαφορετικές κάρτες γραφικών και σε συνδυασμό με την ελαστικότητά είναι μη αναλώσιμο για πολλές εταιρίες, για αυτό η Amazon χρησιμοποιεί αυτό το Framework ως βιβλιοθήκη για μηχανική μάθηση. Το MxNet υποστηρίζει δίκτυα όλων των ειδών όπως LSTM, RNN και CNN και είναι γνωστό για την χρήση του σε προβλήματα εικόνων, κειμένων και NLP.

Theano

Theano είναι μία μαθηματική βιβλιοθήκη που αναπτύχθηκε για την Python και παρέχει βελτιστοποιημένο compiler για την επεξεργασία και υπολογισμό μαθηματικών συναρτήσεων και ειδικά πινάκων. Στην Theano οι υπολογισμοί γίνονται με παρόμοιο συντακτικό με την βιβλιοθήκη Numpy της Python και μεταγλωττίζονται για να τρέξουν τόσο σε επεξεργαστή όσο και σε κάρτα γραφικών. Είναι ένα framework ανοικτού κώδικα που αναπτύχθηκε αρχικά από το πανεπιστήμιου το Montreal για αλγορίθμους μάθησης και η πρώτη του έκδοση έγινε δημόσια διαθέσιμη το 2017. Το αρνητικό είναι πως για δεν παρέχει κάποιο υψηλό επίπεδο αφαιρετικότητας αναγκάζοντας έτσι τον χτίστη να δημιουργήσει από την αρχή το δίκτυο, όπως την αρχιτεκτονική, τα στρώματα, την ενεργοποίηση και τον μηχανισμό μάθησης, πάνω όμως σε ένα σύνολο από διανυσματικές συναρτήσεις κάνοντας το μοντέλο περισσότερο αποδοτικό.

Keras

Το Keras είναι μία βιβλιοθήκη ανοικτής πηγής που είναι γραμμένη σε Python. Είναι δυνατόν να χρησιμοποιηθεί σε συνδυασμό με Tensorflow, Theano, R, ή CNTK. Σχεδιάστηκε για τον εύκολο πειραματισμό με βαθιά νευρωνικά δίκτυα και εστιάζει στην φιλική προς τον χρήση αρχιτεκτονική του. Δημιουργήθηκε από το πρόγραμμα ONEIROS(Open-ended Neuro-Electronic Intelligent Robot Operating System) με κύριο συγγραφέα και υποστηρικτή τον François Chollet, που είναι μηχανικός της Google. Το 2017 η ομάδα Tensorflow της Google πρόσθεσε το Keras στο βασικό πυρήνα του δικού της Framework, ενώ ο συγγραφέας του υποστήριξε πως το Keras δημιουργήθηκε περισσότερο ως διεπαφή παρά ως αυτόνομο framework. Παρέχει ένα υψηλού επιπέδου σύνολο από κατασκευάστηκες μονάδες κάνοντας πολύ εύκολο τον σχεδιασμό βαθιών νευρωνικών δικτύων παρά την πολυπλοκότητα που περιέχει από πίσω. Περιέχει μία μεγάλη ποικιλία από στρώματα,

συναρτήσεις ενεργοποίησης, μηχανισμούς μάθησης και ένα σύνολο εργαλείων που κάνουν πιο εύκολο τον χειρισμό δεδομένων όπως εικόνες και κείμενα. Υποστηρίζει CNN και RNN και πολλά χρήσιμα στρώματα όπως συγκεντρωτικά, κανονικοποίησης και dropout και μπορεί να χρησιμοποιηθεί σε κάρτες γραφικών σε συνδυασμό με CUDA.

DeepLeaning4J

Το DeepLearing4J έχει αναπτυχθεί σε Java και Scala και υποστηρίζει και άλλες υλοποιήσεις σε JVM επίσης. Έχει εγκριθεί ευρέως στον εμπορικό τομέα ως μία διανεμημένη πλατφόρμα μηχανικής μάθησης με μεγαλύτερο πλεονέκτημα πως μπορεί κανείς να χρησιμοποιήσει ολόκληρο το οικοσύστημα της Java για ανάπτυξη μοντέλων μηχανικής μάθησης και μπορεί να είναι διαχειρίσιμο πάνω από το Hadoop και Spark για την παράλληλη επεξεργασία του φόρτου σε πολλούς υπολογιστές. Υποστηρίζει μεγάλη πληθώρα από νευρωνικά δίκτυα όπως CNN,RNN και LSTM και επειδή έχει αναπτυχθεί σε Java είναι πολύ γρηγορότερο και αποδοτικό από αντίστοιχες υλοποιήσεις σε Python, σχετικά με αναγνώριση εικόνων μπορεί να είναι όσο γρήγορο είναι και το Caffe. Το framework εκδόθηκε το 2017 και είναι λογισμικό ανοικτής πηγής και μπορεί να τρέξει τόσο σε επεξεργαστές όσο και σε κάρτες γραφικών καθώς υποστηρίζει CUDA.

Επίλογος

Σε αυτό το κεφάλαιο έγινε μία συνοπτική αναφορά στα Frameworks που χρησιμοποιήθηκαν στην εργασία αυτή και σε κάποια τα οποία είναι ευρέως γνωστά και έχουν βοηθήσει την ανάπτυξη των τεχνητών νευρωνικών δικτύων.

Κεφάλαιο 5

Σύνολο Δεδομένων

Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει αναφορά στο σύνολο δεδομένων που χρησιμοποιήθηκε για την εξαγωγή των αποτελεσμάτων, τα προβλήματα τα οποία βρέθηκαν κατά την επεξεργασία τους καθώς και τρόποι αντιμετώπισης τους.

5.1 Σύνολο Δεδομένων

Το σύνολο δεδομένων που χρησιμοποιήθηκε προέρχεται από την εργασία “Characterizing Political Fake News in Twitter by its Meta-Data” [30]. Στην εργασία αυτή τα δεδομένα συγκεντρώθηκαν από με την χρήση όρων αναζήτησης που έχουν σχέση με τις προεδρικές εκλογικές που διεξήχθησαν στην Αμερική στις 8 Νοεμβρίου του 2016. Ποιο συγκεκριμένα έγινε αναζήτηση με την χρήση του Twitter API με όρους αναζήτησης και ετικέτες(hashtags) : #MyVote2016, #ElectionDay, #electionnight, @realDonaldTrump και @HillaryClinton, τα δεδομένα προήλθαν μόνον από την ημέρα των εκλογών. Από το σύνολο τους αφαιρέθηκαν τα διπλότυπα και επίσης 1 ειδικός κατηγοριοποίησε τα δεδομένα

ως προς το αν περιέχουν ψευδές περιεχόμενο ή όχι. Επιπλέον αφαιρέθηκαν όσα δεδομένα δεν θεωρείται πως είχαν μεγάλο αντίκτυπο, θεωρήθηκε δηλαδή πως αν ένα Tweet δεν έχει γίνει re-tweet περισσότερο από 1000 φορές δεν θα έπρεπε να συμπεριληφθεί στο τελικό σύνολο δεδομένων. Το τελικό αποτέλεσμα περιείχε αρχικά 1.785.855 Tweets από τα οποία μόνον τα 1327 είχαν διαδοθεί περισσότερο από 1000 φορές. Την κατηγοριοποίηση ανέλαβε ένα άτομο ώστε να διατηρηθεί συνέπεια σε όλο το σύνολο των δεδομένων.

	is_fake_news	text
0	0	@realDonaldTrump you are full of shit!
1	0	@realDonaldTrump you're fucking retarded
2	0	@realDonaldTrump You are the stupidest man on planet earth.
3	0	@realDonaldTrump I am continually amazed and terribly inspired by you! If such an idiot can make a ton of money, here's hope for me yet!
4	0	Hey @realDonaldTrump You Mad Bro?
5	0	@realDonaldTrump Go fuck yourself.
6	0	@realDonaldTrump If you hate America so much, you should run for President and fix things
7	0	@realDonaldTrump you asshole STFU abt military you know NOTHING. Men like YOU assault & think it's a woman's fault. As a VET you sicken me
8	0	'@arcuate: dude that's freaking cool as heck RT @realDonaldTrump wind turbine blades will slice 14 million birds and bats to death in 10 yrs
9	0	β€@realDonaldTrump: I would like to extend my best wishes to all, even the haters and losers, on this special date, September 11th.β€
10	0	Beautiful morning walk in Hyde Park #London with @HillaryClinton. http://t.co/0aQNITRkvA
11	0	Retweet if you Agree w/ @HillaryClinton: http://t.co/cjmOvmyKkV #HillaryClinton @Faith4Hillary @VOTE_HILLARY @VOTE_CLINTON @ReadyForHillary
12	0	'@johnnyhabit: @realDonaldTrump why the fuck do I even follow you? Because you're addicted to genius, asshole!
13	0	'@KEEMSTARx: If @realDonaldTrump was president, middle class would make more $. I honestly think he could keep Jobs in America! So true-easy
14	0	Haha @HillaryClinton just came up and asked 'Are you guys The Vamps' ??? We are all confused and impressed. She was sooooo nice!
15	0	@realDonaldTrump Do you pay an angry 12 year old boy to handle your twitter account?
16	0	@realDonaldTrump Sadly your racism is poisoning your barely functioning brain. You are a terrible person.
17	0	Hillary Clinton has stolen our innovative WikiLeaks twitter logo design. Compare: @WikiLeaks vs @HillaryClinton http://t.co/mifka4mXf4
18	0	yaaas @hillaryclinton
19	0	It's clear @realdonaldtrump been racist..saying our 'African American president' it's plain to see! We won't miss u when u perish!
20	0	There's an important choice to be made today; vote wisely: - Favourite for milk first - Retweet for milk last #ElectionDay
21	0	@realDonaldTrump has been a great friend for many years. We don't need another politician, we need a businessman like Mr. Trump! Trump 2016
22	0	'@USARestoring: @HillaryClinton's toast. Dems had better get the'B Team' off the bench. @TGowdySC for Attorney General under President Trump
23	0	Not sure why @megynkelly blasts @realDonaldTrump for objectifying woman & treating women as objects. REALLY Megyn? http://t.co/RPcFjsn7Mb
24	0	i mean @realDonaldTrump if you're so good at negotiating how come you couldn't get @DonaldTrump
25	0	'@AmyMek Every Time I see @realDonaldTrump address a crowd I want to start chanting USA, USA, USA! #AmericanPride is Back #Trump2016'
26	0	If @realDonaldTrump only keeps 10% of his promises that will still be 10 times more than the @GOP @RNC has in 20 years.
27	0	.@realDonaldTrump a real hero would have actually written 'Fuckface'. Coward.
28	0	Fuck @realDonaldTrump
29	0	Which clown would you rather see as President? retweet for Bozo the clown, Favorite for donald trump @realDonaldTrump http://t.co/J7DkeAhmfk
30	0	Hey @realDonaldTrump you saw a kid lost in New York and didn't tell an adult. How can we trust you to be president? https://t.co/FUp7wgUA2c
31	0	@realDonaldTrump Thank you for tweeting me. So honored to support you. https://t.co/S2SUaD3v4K

Σχήμα 25 : Σύνολο δεδομένων

Από τα 1327, 136 αναγνωρίστηκαν με περιεχόμενο που πιθανόν να περιέχει ψευδές πληροφορίες και τα υπόλοιπα χαρακτηρίστηκαν ως αληθή αντικείμενα, ωστόσο σημειώνεται πως η κατηγορία στην οποία εντάσσεται κάθε αντικείμενο δεν μπορεί να χαρακτηριστεί ως γεγονός λόγω της ασάφειας των ψευδών ειδήσεων και του ανθρώπινου παράγοντα. Τα δεδομένα περιείχαν και άλλες πληροφορίας όπως το όνομα του χρήστη που δημοσίευσε το Tweet, την ώρα, τον αριθμό των re-tweets κ.α. όμως σε αυτά δεν γίνεται αναφορά καθώς αυτή η εργασία επικεντρώνεται στην αναγνώριση των ψευδών ειδήσεων από το κείμενο μόνο.

Ένα δεύτερο σύνολο δεδομένων που ήταν διαθέσιμο από αυτήν την εργασία περιείχε 9001 αντικείμενα τα οποία είχαν κατηγοριοποιηθεί από 2 άτομα το καθένα. Αυτό το σύνολο δημιουργήθηκε από τους συγγραφείς του [30] με σκοπό να κατηγοριοποιηθούν τα tweets σε

διαφορετικές κατηγορίες όσον αφορά το περιεχόμενο αλλά και να γίνει διαχωρισμός σε αληθής και ψευδής πληροφορία. Το αρνητικό με αυτήν την προσέγγιση είναι πως πολλά αντικείμενα κατέληξαν να ενταχθούν σε μία κατηγορία από τον ένα ταξινομητή και σε άλλη από τον δεύτερο. Επίσης σε αυτό το σύνολο δεδομένων οι ταξινομητές είχαν την δυνατότητα να μην εντάξουν κάποιο αντικείμενο ως προς το περιεχόμενό του και να το χαρακτηρίσουν ως άγνωστο(unknown). Το πρόβλημα που εμπίπτει είναι λοιπόν πως θα γίνει ο σύμπτυξη των 2 ταξινομήσεων σε μία.

Μία προσέγγιση είναι να δεχτούμε ως έγκυρα δεδομένα μόνον αυτά στα οποία οι 2 ταξινομητές συμφωνούσαν πλήρως, να διαγράφαμε τα άγνωστα και να διεξάγουμε τα πειράματα με τα έγκυρα δεδομένα που απέμεναν. Δεδομένης αυτής της προσέγγισης τα δεδομένα με τα οποία θα είχαμε μείνει θα ήταν στο σύνολο τους 6641 από τα οποία τα 158 είχαν χαρακτηριστεί να έχουν ψευδές περιεχόμενο και τα υπόλοιπα αληθές. Με αυτόν τον τρόπο όμως έχει δημιουργηθεί ένα σύνολο δεδομένων του οποίου οι 2 κλάσεις υπάρχουν σε μεγάλη ανισορροπία, για την ακρίβεια υπάρχει 1 αντικείμενο με ψευδές περιεχόμενο για κάθε 48 με αληθές.

Μία διαφορετική λύση θα αποτελούσε η χρήση μόνο του ενός εκ των 2 ταξινομητών. Βάση αυτής την λογικής όμως θα πρέπει να επιλεγεί ποιος από τους 2 θα είναι ο καταλληλότερος και αυτό θα γινόταν δεδομένου της εγκυρότητας των αποτελεσμάτων τους αλλά και της συνέπειας των κριτηρίων με των οποίων γίνεται η ταξινόμηση. Με αυτόν τον τρόπο το μοντέλο θα εκπαιδευτεί περισσότερο αποδοτικά και θα έχει καλύτερα αποτελέσματα στην διαδικασία της δοκιμής. Επειδή οι κατηγορίες είναι 2 και αρκετά ανισόρροπες είναι εύκολο να γίνει μία κριτική πάνω στους 2 ταξινομητές κοιτάζοντας απλά τα δεδομένα από την κλάση με τα λιγότερα στοιχεία, δηλαδή αυτά που έχουν ταξινομηθεί ως ψευδή. Μία αρχική παρατήρηση που είναι εύκολο να γίνει είναι η μεγάλη διαφοροποίηση στην ταξινόμηση ως προς τα ψευδή στοιχεία, με τον πρώτο ταξινομητή να εντάσσει 1729 αντικείμενα ως ψευδή και τον δεύτερο 404 και τα κοινά τους όπως είδαμε από πάνω ανέρχονται στα 158! Κοιτάζοντας τα δεδομένα, μία πρόταση με διαφορετική κατηγοριοποίηση είναι η ακόλουθη: "*If @realDonaldTrump was president, middle class would make more $. I honestly think he could keep Jobs in America!*". Ο πρώτος ταξινομητής κατέληξε να θεωρεί την πρόταση ως ψευδές περιεχόμενο ενώ ο δεύτερος όχι. Σε αυτήν την πρόταση το περιεχόμενο δεν είναι απαραίτητα ψευδές αλλά εμπίπτει στον τρόπο που το δέχεται ο καθένας. Η πρόταση μπορεί να ληφθεί σαν προσωπική δήλωση των απόψεων ενός ατόμου, αυτή είναι και η δική μου ιδέα, ή ως μία δήλωση με ψευδές περιεχόμενο λόγω του ότι δεν υπάρχει καμία ένδειξη πως είναι αληθής. Ωστόσο στην συνέχεια μπορεί να παρατηρηθεί το αντικείμενο με το ακόλουθο κείμενο: "Every Time I see @realDonaldTrump address a crowd I want to start chanting USA, USA, USA! #AmericanPride is Back #Trump2016". Σε αυτήν την περίπτωση ο δεύτερος ταξινομητής χαρακτήρισε το περιεχόμενο ως ψευδές ενώ ο πρώτος ως αληθές. Τι μπορεί να εξαχθεί ως αποτέλεσμα για το ερώτημα ποιος ταξινομητής πρέπει να επιλεγεί; Το κύριο πρόβλημα που εμπίπτει από την χρήση των 2 αυτών παραδειγμάτων είναι πως κανένας από τους 2 ταξινομητές δεν χαρακτήριζε τα αντικείμενα με συνέπεια, δηλαδή αντικείμενα που είναι αμφιλεγόμενα ως προς το περιεχόμενο να ταξινομούνται σταθερά ως ψευδή ή αληθή. Ένα ακόμα παράδειγμα που είναι κάπως περισσότερο ανησυχητικό είναι το εξής: "Hey @realDonaldTrump your bad boy son cut off the tail of an elephant! Why??? http://t.co/47qW3kztuu". Αν ακολουθήσει κανείς τον σύνδεσμο του κειμένου θα δει μία εικόνα του γιού του Donald Trump να στέκεται δίπλα σε ένα νεκρό ελέφαντα με ένα μαχαίρι στο χέρι, κρατώντας την ουρά του και έχοντας ένα όπλο να ακουμπάει πάνω στο σώμα του. Από τα όσα δείχνει η φωτογραφία είναι εύκολο να χαρακτηριστεί το

περιεχόμενο ως αληθές ωστόσο ο πρώτος ταξινομητής μόνο το θεωρεί έτσι, με τον δεύτερο να το χαρακτηρίζει ως ψευδές. Αν παρατηρήσει κανείς καλύτερα την εικόνα μπορεί να δει πως το μαχαίρι που κρατάει είναι πολύ μικρό και καθαρό για να έχει προηγηθεί κάποιου τέτοιου είδους πράξη, ωστόσο το μήνυμα που περνάει η συγκεκριμένη φωτογραφία είναι αδιαμφισβήτητα συνδεδεμένο με το σχόλιο που έγινε στο συγκεκριμένο tweet. Σε αυτό το δεύτερο παράδειγμα φαίνεται πως μπορεί να επηρεάσει ο ανθρώπινος παράγοντας την ταξινόμηση.

Τελικά η τακτική που ακολουθήθηκε για επιλογή της κλάσης από αυτό το δεύτερο σύνολο είναι διαφορετική. Όπως αναφέρθηκε αρχικά τα δεδομένα μπορούσαν να χαρακτηριστούν ως άγνωστα πέρα από αληθή και ψευδή, κάτι το οποίο δεν μας βοηθάει για την χρήση σε αλγόριθμο με δυαδικά αποτελέσματα. Για αυτό τον λόγο έγινε η επιλογή όσο στοιχεία είχαν κατηγοριοποιηθεί ως άγνωστα από τους 2 ταξινομητές να εξαιρεθούν από το σύνολο. Στην συνέχεια όσο είχαν κατηγοριοποιηθεί ως άγνωστα από τον ένα και αληθή ή ψευδή από τον άλλο επιλέχθηκε ο χαρακτηρισμός που εμπίπτει σε μία από τις 2 κατηγορίες, με αυτόν τον τρόπο λύθηκε το πρόβλημα των αγνώστων. Τα αντικείμενα που είχαν εισαχθεί στην ίδια κατηγορία και από τους 2 ταξινομητές κράτησαν την κατηγορία τους ενώ όσα είχαν διαφορετική εξαιρέθηκαν από το τελικό αποτέλεσμα. Τελικά το σύνολο δεδομένων που προέκυψε αποτελούνταν από 7224 στοιχεία από τα οποία τα 242 ήταν χαρακτηρισμένα ως ψευδή και τα υπόλοιπα ως αληθή.

5.2 Προεπεξεργασία

Η προ επεξεργασία των δεδομένων είναι συνήθως το πρώτο βήμα κατά την διαδικασία της επεξεργασίας φυσικής γλώσσας με πιθανές επιπτώσεις στην τελική απόδοση. Οι [31] χρησιμοποίησαν 2 διαφορετικά μοντέλα, ένα CNN και ένα υβριδικό με CNN και LSTM στρώματα και 4 τρόπους προ επεξεργασίας. Ο πιο συνηθισμένος τρόπος προ επεξεργασίας είναι ο διαχωρισμός των προτάσεων σε λέξεις και αντιμετώπισης κάθε λέξης σαν διαφορετικό στοιχείο. Οι επιπλέον τρόποι που χρησιμοποιηθήκαν είναι η μετατροπή όλων των χαρακτήρων σε πεζά (lowercase) , η λημματοποίηση(lemmatization), και η χρήση πολλαπλών λέξεων ως ένα στοιχείο για εξαγωγή χαρακτηριστικών(multiword embedding).

Η μετατροπή όλων των γραμμάτων σε πεζά είναι ο πιο απλός τρόπος προ επεξεργασίας κειμένου και αποτελείται από την μετατροπή των κεφαλαίων σε πεζά, οπότε για παράδειγμα το κείμενο:

“Η Microsoft αναγνωρίζει το πρόβλημα με την τελευταία αναβάθμιση των Windows 10 και εξηγεί πως να το φτιάξετε”

θα μετατραπεί σε:

“η Microsoft αναγνωρίζει το πρόβλημα με την τελευταία αναβάθμιση των windows 10 και εξηγεί πως να το φτιάξετε”

Λόγω της απλότητας του η μετατροπή όλων των χαρακτήρων σε πεζούς έχει γίνει ιδιαίτερα δημοφιλής πρακτική σε προβλήματα NLP και έχει διαδοθεί σε αρκετές βιβλιοθήκες και frameworks. Το θετικό αντίκτυπο που έχει μία τέτοια μετατροπή είναι πολύ εύκολο να γίνει κατανοητό καθώς μειώνεται σε μεγάλο βαθμό το μέγεθος των λέξεων που πρέπει να μάθει το δίκτυο, αν σκεφτεί κανείς πως μόνο στην αρχή κάθε πρότασης η λέξη που έχει ένα κεφαλαίο γράμμα θεωρείται ξεχωριστό

στοιχείο από την ίδια λέξη μέσα στην πρόταση που οι χαρακτήρες έχουν γραφτεί όλοι με πεζά. Αντίθετα όμως με τη θετική ιδιότητά της να μειώνει την διασπορά στο σύνολο των λέξεων μπορεί να επηρεάσει και αρνητικά το δίκτυο. Για παράδειγμα η λέξη *Windows* που στην συγκεκριμένη περίπτωση αντιπροσωπεύει το όνομα του λογισμικού και η λέξη *windows*, δηλαδή τα παράθυρα, αντιπροσωπεύουν το ίδιο στοιχείο πλέον.

Για γραμματικούς λόγους στα κείμενα χρησιμοποιούνται διαφορετικές μορφές της ίδιας λέξης όπως organize, organizes και organizing. Επιπλέον υπάρχουν ομοιότητες στην σημασία μεταξύ λέξεων με παρόμοια παράγωγα όπως democracy. Ο στόχος της λημματοποίησης και της εξάλειψης είναι να μειώσουν τις μορφές με την οποία εφαρμόζεται μία λέξη με όμοια σημασία σε μία μόνο μορφή. Για παράδειγμα:

car, car's, cars, cars' → car

am, are, is → be

Το αποτέλεσμα μίας τέτοιας μετατροπής θα έχει ως αποτέλεσμα κάτι παρόμοιο με:

the boy's cars are different colors → the boy car be differ color

Ωστόσο οι 2 αυτές παρόμοιες τεχνικές διαφέρουν στην μεθοδολογία που ακολουθούν για να επιτύχουν τον στόχο τους. Η εξάλειψη συνήθως κόβει τους τελευταίους χαρακτήρες μίας λέξης με την ελπίδα πως θα έχει το σωστό αποτέλεσμα, συνήθως πετυχαίνοντας το, και πολλές φορές συμπεριλαμβάνει τη διαγραφή των παράγωγων επισημάνσεων. Με την λημματοποίηση συνήθως ακολουθείται μία πιο ορθολογική προσέγγιση. Γίνεται χρήση ενός λεξικού και με την μορφολογική ανάλυση της λέξης αφαιρείται η κατάληξη της και επιστρέφεται το στοιχείο όπως υπάρχει στο λεξικό που είναι γνωστό και ως λήμμα. Για παράδειγμα αν υποθέσουμε πως σε μία πρόταση υπάρχει η λέξη *saw,* η εξάλειψη πιθανώς να επιστρέψει μονάχα *s* ενώ η λημματοποίηση *see* ή *saw*, ανάλογα ποια λέξη υπάρχει στο λεξικό και αν έχει χρησιμοποιηθεί ως επίθετο ή ρήμα. Σκοπός και των 2 τεχνικών είναι να μειωθεί η διασπορά των λέξεων καθώς διαφορετικά μορφές μίας λέξης μπορεί να εμφανίζονται λιγότερο συχνά από την ίδια την λέξη. Ωστόσο αυτό μπορεί να έχει αντίκτυπο καθώς δεν λαμβάνονται υπόψη διαφορετικά παράγωγα της ίδιας λέξης.

Η τελευταία τεχνική προ επεξεργασίας έχει να κάνει με την ομαδοποίηση συγκεκριμένων στοιχείων κάτω από ένα στοιχείο αν βρεθούν κάτω από κάποιες προϋποθέσεις:

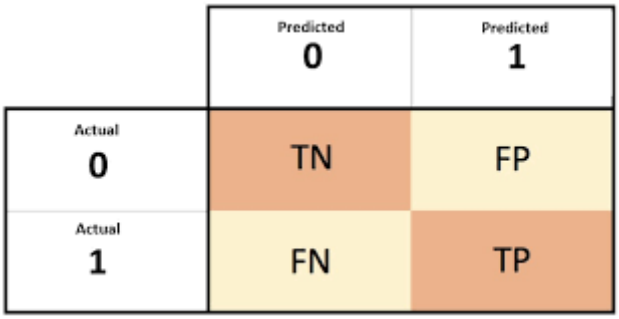
"Η Microsoft αναγνωρίζει το πρόβλημα με την τελευταία αναβάθμιση των Windows_10 και εξηγεί πως να το φτιάξετε"

Η προσέγγισή αυτή έχει να κάνει ιδιοσυγκρασιακή φύση των εκφράσεων, *Windows 10* στο παράδειγμα αυτό. Η σημασία των εκφράσεων αυτών συνήθως είναι δύσκολο να βρεθεί στα ξεχωριστά στοιχεία που την απαρτίζουν και ως αποτέλεσμα η αντιμετώπισή τους ως ένα στοιχείο μπορεί να οδηγήσει σε καλύτερη εκπαίδευση του μοντέλου. Ως αποτέλεσμα αυτού μοντέλα ενσωμάτωσης όπως το Word2Vec και το GloVe χρησιμοποιούν στοιχεία πολλών λέξεων ταυτόχρονα με αυτόνομα στοιχεία στα προ εκπαιδευμένα μοντέλα τους.

5.3 Ανισορροπία κλάσεων

Όπως είναι εμφανές από τα δεδομένα που παρουσιάστηκαν οι 2 κλάσεις υπάρχουν σε σχετικά μεγάλη ανισορροπία μεταξύ τους, για την ακρίβεια υπάρχουν περίπου 3.3 αντικείμενα που έχουν χαρακτηριστεί ως ψευδή για κάθε 100 αντικείμενα που βλέπει το μοντέλο. Κατά την εκπαίδευση αλλά και την ανάκληση αυτό μπορεί να προκαλέσει παρερμήνευση των αποτελεσμάτων. Για

παράδειγμα είναι πολύ εύκολο το μοντέλο να μάθει να ταξινομεί τα πάντα ως αληθή και ως αποτέλεσμα θα έχει ποσοστό ακρίβειας πάνω από 96.5%. Για αρχή όμως ας δούμε με ποιους τρόπους γίνεται η μέτρηση της αποτελεσματικότητας ενός μοντέλου.



Σχήμα 26 : Πίνακας Σύγχυσης(Confusion Matrix)

Όπως φαίνεται στο σχήμα 38 στις γραμμές είναι οι πραγματικές κλάσεις στις οποίες ανήκουν τα αντικείμενα και οι στήλες είναι οι κλάσεις στις οποίες ταξινομήθηκαν. Έτσι τα αντικείμενα που ταξινομήθηκαν διαγώνια από αριστερά προς δεξιά έχουν ταξινομηθεί σωστά ενώ τα υπόλοιπα λάθος. Οι συντομογραφίες που χρησιμοποιούνται αντιπροσωπεύουν τα δεδομένα που ταξινομήθηκαν έτσι :

TN = True Negative (ταξινομήθηκαν στην κλάση 0, και όντως ανήκουν σε αυτήν)

TP = True Positive (ταξινομήθηκαν στην κλάση 1, και όντως ανήκουν σε αυτήν)

FN = False Negative (ταξινομήθηκαν στην κλάση 0, αλλά ανήκουν στην κλάση 1)

FP = False Positive (ταξινομήθηκαν στην κλάση 1, αλλά ανήκουν στην κλάση 0)

Στα δυαδικά προβλήματα δεν υπάρχει κάποιος κανόνας που να ορίζει ποια κλάση θα αντιπροσωπεύει τα Positive και ποια τα Negative, και είναι περισσότερο ονομασίες που μπορούν να μετατραπούν ώστε να αντιπροσωπεύουν καλύτερα την κάθε περίσταση, όπως για παράδειγμα στην εργασία αυτή υπάρχουν αληθής και ψευδής ειδήσεις αντί για θετική και αρνητική κλάση. Οι μετρήσεις πάνω στις οποίες βασίζεται η απόδοση του μοντέλου βασίζονται στα TN,TP,FN και FP και είναι οι ακόλουθες:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.1)$$

$$Precision = \frac{TP}{TP + FP} \quad (5.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (5.3)$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (5.4)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (5.5)$$

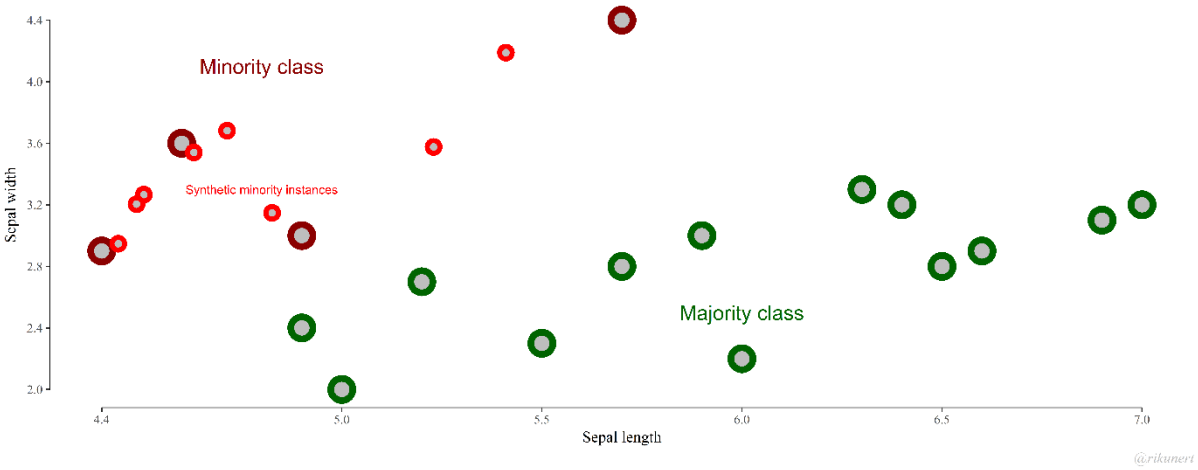
$$Specificity = \frac{TN}{TN + FP} \quad (5.6)$$

Σύμφωνα με τα παραπάνω το μοντέλο μας λοιπόν θα είχε accuracy σε ποσοστό 96.7%, η ανάκληση(recall) για την κλάση αληθών ειδήσεων θα ήταν 100% που είναι και το καλύτερο δυνατό αποτέλεσμα, ενώ για τις ψευδείς ειδήσεις θα ήταν 0%, οπότε το μοντέλο δεν θα είχε καλά αποτελέσματα για αυτήν την κλάση. Ενώ το F1 δεν θα ήταν υπολογίσιμο για τις ψευδείς ειδήσεις και για τις αληθείς θα ήταν 0.98. Οπότε λόγω της μεγάλης ανισορροπίας μεταξύ των κλάσεων, κοιτάζοντας τον πίνακα σύγχυσης θα οδηγούσε αρχικά σε λάθος συμπεράσματα. Σε ένα πρόβλημα ταξινόμησης οι κλάσεις δεν περιέχουν ποτέ ακριβώς τον ίδιο αριθμό αντικειμένων καθώς πάντα τα δεδομένα πρώτα συλλέγονται και στην συνέχεια κατηγοριοποιούνται από κάποιον ειδικό. Ωστόσο υπάρχουν μερικοί συνηθισμένοι τρόποι για την εξισορρόπηση κλάσεων σε τέτοιου είδους προβλήματα.

Μία από αυτές τις τεχνικές ονομάζεται under sampling(υποδειγματοποίηση) κατά την οποία επιλέγονται όλα τα αντικείμενα από την κλάση που αποτελεί μειονότητα ενώ γίνεται δειγματοληψία αντικειμένων από την υπερέχουσα κλάση με σκοπό στο τέλος να προκύψει ένα πιο ισορροπημένο σύνολο δεδομένων. Η υπερδειγματοληψία είναι μία άλλη τεχνική στην οποία συμβαίνει το αντίθετο, δηλαδή αντιγράφονται μερικά αντικείμενα της μικρότερης κλάσης με σκοπό να αυξηθούν τα ποσοστά της ως προς το σύνολο. Ένας άλλος τρόπος είναι με την δημιουργία κάποιων συνθετικών δεδομένων στην μικρότερη κλάση είναι με την δημιουργία συνθετικών αντικειμένων για την μικρότερη κλάση αντί για αντιγραφή τους. Η περισσότερο διαδεδομένη τεχνική ονομάζεται SMOTE(Synthetic Minority Oversampling Technique) [32]. Βάση της SMOTE τα δεδομένα παρουσιάζονται σε ένα σύστημα αξόνων και επιλέγοντας τους κοντινότερους γείτονες από

αντικείμενα της ίδιας κλάσης δημιουργούνται τεχνητές εγγραφές με τιμές που βρίσκονται στον μεταξύ χώρο των γειτονικών αντικειμένων.

Σύνολο Δεδομένων



Σχήμα 27 : SMOTE

Όλες αυτές οι προσεγγίσεις αποτελούν διαφορετικούς τρόπους για την επεξεργασία των δεδομένων εισόδου ωστόσο υπάρχουν και διαφορετικές τεχνικές. Μέχρι τώρα θεωρούσαμε έναν ταξινομητή με υψηλή ακρίβεια δεδομένου πως το κόστος του σφάλματος και των 2 κλάσεων είναι ισάξιο. Στο παράδειγμά μας δηλαδή θεωρούσαμε πως η λάθος ταξινόμηση μίας ψευδής είδησης έχει το ίδιο κόστος με την λάθος ταξινόμηση μίας αληθής είδησης. Αυτό σημαίνει πως τα σφάλματα των 2 κλάσεων είναι συμμετρικά και αυτό θα ήταν το ιδανικό. Ωστόσο στην συγκεκριμένη περίπτωση είναι δυνατόν να προσθέσουμε περισσότερο βάρος στα αντικείμενα που αντιπροσωπεύουν την μειονότητα, έτσι με αυτόν τον τρόπο κατά την διάρκεια της εκπαίδευσης τα βάρη μπορούν και να επανα-υπολογίζονται περισσότερο βέλτιστα ως προς την μειονεκτική κλάση.

Στην εργασία αυτή δοκιμάστηκαν όλες οι παραπάνω μέθοδοι ωστόσο 2 από αυτές αποδείχθηκαν περισσότερο χρήσιμες, η υποδειγματοποίηση και η αντιστάθμιση του βάρους των κλάσεων. Για την δεύτερη τα αποτελέσματα δεν είναι τόσο ισχυρά και περισσότερες δοκιμές σε μεγαλύτερο σύνολο δεδομένων θα πρέπει να διενεργηθούν. Με τις υπόλοιπες τεχνικές δεν παρατηρήθηκαν καλύτερα αποτελέσματα από την βάση του μοντέλου, με την βάση να είναι το ποσοστό της πλειοψηφικής κλάσης.

Επίλογος

Σε αυτό το κεφάλαιο έγινε αναφορά στο σύνολο δεδομένων που χρησιμοποιήθηκε για την εξαγωγή των αποτελεσμάτων, τα προβλήματα τα οποία βρέθηκαν κατά την επεξεργασία τους καθώς και τρόποι αντιμετώπισης τους.

Κεφάλαιο 6

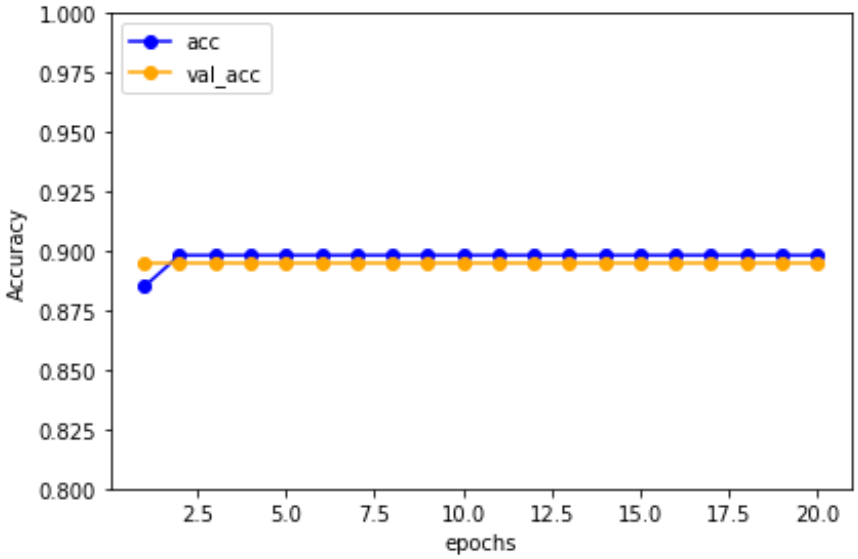
Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει παρουσίαση των αποτελεσμάτων των μοντέλων που δοκιμάστηκαν. Θα παρουσιαστούν οι μετρήσεις πάνω στα αποτελέσματα και διαγράμματα που παρουσιάζουν την διαδικασία την μάθησης.

6.1 Αποτελέσματα

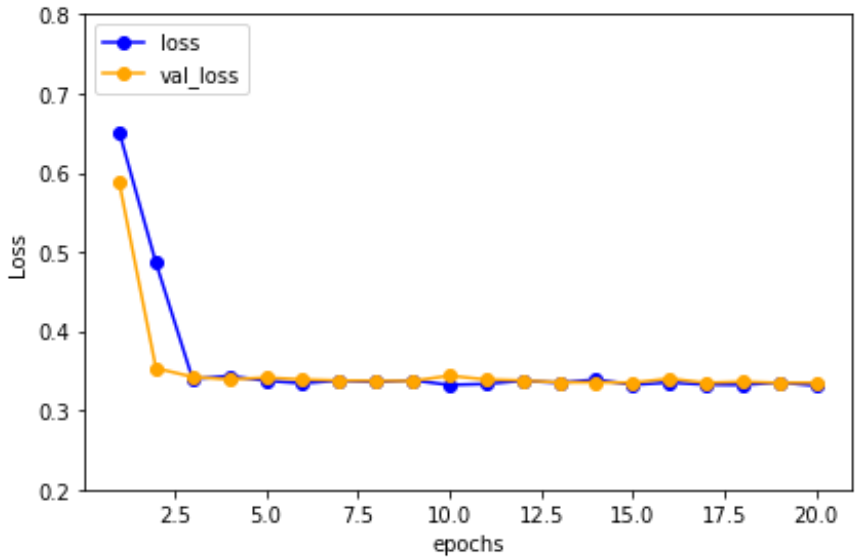
Character Level CNN

Για το συγκεκριμένο δίκτυο η προ-επεξεργασία που διενεργήθηκε αφορούσε την διαγραφή χαρακτήρων που δεν εμπεριέχονται στην κωδικοποίηση ASCII και στην συνέχεια μετατροπή των Tweet σε πίνακες διαστάσεων 140 επί 95. 140 που ήταν το μέγιστο επιτρεπόμενο όριο χαρακτήρων στο Tweeter και 95 το σύνολο των χαρακτήρων στην κωδικοποίηση ASCII. Από την μετατροπή αυτή σε one-hot-vectors έγινε και η αρχικοποίηση του Embedding στρώματος αφού χρησιμοποιήθηκε ο δείκτης όπου κάθε χαρακτήρας είχε την μονάδα στον πίνακα. Το δίκτυο ήταν όπως παρουσιάστηκε στο κεφάλαιο 3 ενώ στο τελευταίο στρώμα χρησιμοποιήθηκε η Σιγμοειδής συνάρτηση ενεργοποίησης και ένα πυκνό στρώμα 2 νευρώνων, κάθε νευρώνας αντιπροσώπευε την κάθε κλάση. Η εκπαίδευση έγινε με την χρήση της μεθόδου k-fold cross-validation για την οποία χρησιμοποιήθηκαν 3 folds.



Σχήμα 28 : Ακρίβεια σε Character Level CNN

Το υψηλότερο ποσοστό ακρίβειας κατά την εκπαίδευση ήταν 0.8935 και κατά την ανάκληση 0.8898.



Σχήμα 29 : Σφάλμα σε Character Level CNN

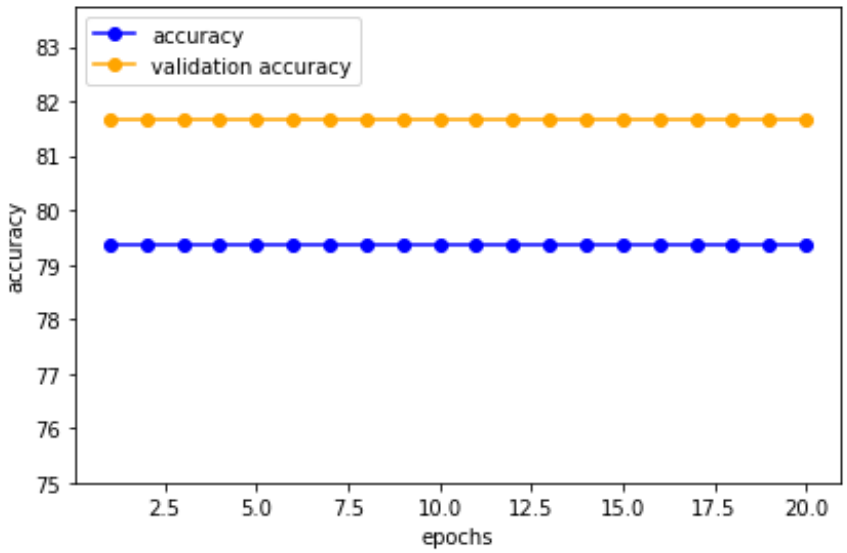
Το ελάχιστο σφάλμα που επιτεύχθηκε στην εκπαίδευση ήταν 0.3458 ενώ για την ανάκληση 0.3495.

Για την εκπαίδευση χρησιμοποιήθηκε η συνάρτηση της κατάβασης δυναμικού με παράγοντα μάθησης 0.1. Υπόλοιπες συναρτήσεις που χρησιμοποιήθηκαν όπως Adam και rmsprop οδήγησαν το μοντέλο σε υπερ-εκπαίδευση με αποτέλεσμα υψηλότερο σφάλμα και μικρότερη ακρίβεια στην ανάκληση.

Εκπαίδευση Συσχετίσεων Word2Vec

Στο συγκεκριμένο μοντέλο έγινε δημιουργία των συσχετίσεων από το ίδιο σύνολο δεδομένων και χρησιμοποίησή τους για την εξαγωγή συμπερασμάτων και την κωδικοποίηση των λέξεων.

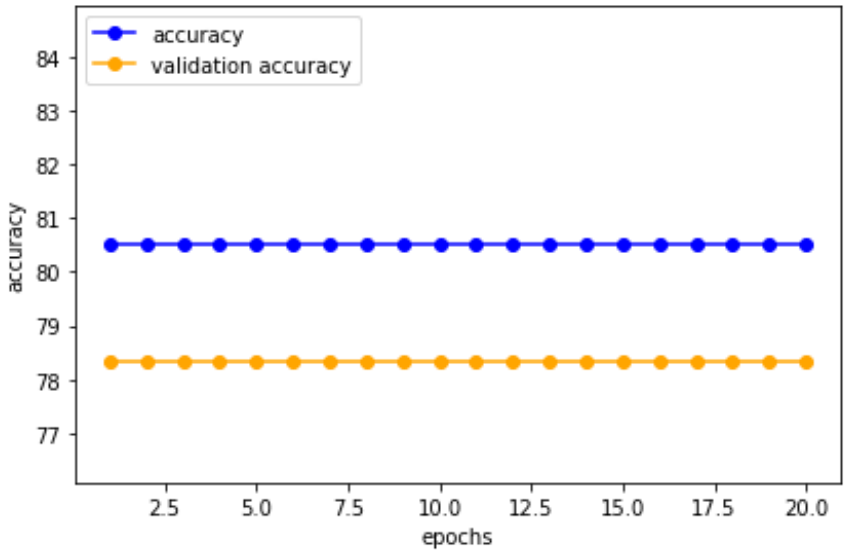
Η προ επεξεργασία που ακολουθήθηκε εδώ ήταν η μετατροπή όλων των χαρακτήρων σε πεζούς και η αντικατάσταση όλων των συνδέσμων διαδικτύου με την συντομογραφία http. Χρησιμοποιήθηκε το μοντέλο Word2Vec από την βιβλιοθήκη genism από το οποίο παράχθηκαν 100 διανύσματα για κάθε λέξη που εμφανίστηκε πάνω από μία φορά και παίρνοντας υπόψιν τις γειτονικές μόνο λέξεις. Στην συνέχεια για την κωδικοποίηση των λέξεων σε διανύσματα χρησιμοποιήθηκε ο μέσος όρος όλων των στηλών και το αποτέλεσμα αντιπροσώπευε την λέξη. Αρχικά ως ταξινομητής χρησιμοποιήθηκε ο SVM(Support Vector Machines) με την σιγμοειδή συνάρτηση ενεργοποίησης όμως το μοντέλο δεν ήταν ικανό το αποτέλεσμα που έπιανε το μοντέλο ήταν η απόλυτη βάση, δηλαδή το ποσοστό επιτυχίας ήταν όσο και το ποσοστό των αληθών ειδήσεων, κατηγοριοποιώντας όλες τις ψευδείς ειδήσεις λάθος:



Σχήμα 30 : Ακρίβεια του SVM

Έγινε επίσης χρήση βαρών για τις 2 κλάσεις όμως το μοντέλο δεν κατάφερε να πετύχει καλύτερα αποτελέσματα.

Επίσης έγινε χρήση του AdaBoostClassifier με την χρήση του ταξινομητή DecisionTreeClassifier. Χρησιμοποιήθηκαν 1000 νευρώνες και ρυθμός μάθησης(learning_rate) 1 ωστόσο και εδώ τα αποτελέσματα είναι παρόμοια με τον SVM. Έγιναν δοκιμές με διαφορετικό αριθμό νευρώνων, πιο συγκεκριμένα δοκιμάστηκαν τα σύνολα 100,200,1000,2000 και 5000 νευρώνες ωστόσο ενώ το μοντέλο με 1000 νευρώνες και πάνω μπορούσε να πετύχει 100% ακρίβεια στην εκπαίδευση το αποτέλεσμα στην γενίκευση δεν ήταν καλό αποδεικνύοντας πως το μοντέλο είχε φτάσει στο σημείο υπερ-εκπαίδευσης.



Σχήμα 31 : Ακρίβεια AdaBoost

B.E.R.T.

Για την προ-επεξεργασία του B.E.R.T. έγινε μόνο η μετατροπή όλων των χαρακτήρων που αντιπροσωπεύουν συνδέσμους διαδικτύου σε http και η μετατροπή του κειμένου σε tokens βάση του TreebankWordTokenizer της βιβλιοθήκης nltk. Το μοντέλο που χρησιμοποιήθηκε είναι το βασικό με τα δώδεκα στρώματα καθώς ο υπολογιστής που κατέχω δεν έχει ικανοποιητικό υλικό για να τρέξει το μεγαλύτερο μοντέλο. Ακολουθήθηκε ο οδηγός από το [34] και χρησιμοποιήθηκε batch_size 16 με το προ-εκπαιδευμένο μοντέλο για πεζούς χαρακτήρες. Δυστυχώς το μοντέλο δεν πέτυχε καλά αποτελέσματα ταξινομώντας όλα τα δεδομένα ως αληθή:

1	0.8396243	0.1603757
2	0.9090578	0.090942174
3	0.84330267	0.15669732
4	0.915113	0.084887035
5	0.8518958	0.14810413
6	0.91534454	0.084655486
7	0.9151553	0.08484469
8	0.9014406	0.09855942
9	0.91538227	0.08461775
10	0.9162326	0.08376738
11	0.7944244	0.20557557
12	0.80942446	0.19057558
13	0.9139401	0.086059935
14	0.9079935	0.09200644
15	0.91512305	0.084876925
16	0.9129987	0.08700131
17	0.830311	0.16968897
18	0.9103613	0.08963867
19	0.7937704	0.20622967
20	0.8265731	0.17342699
21	0.7980985	0.20190154
22	0.8048762	0.1951238
23	0.915291	0.08470906
24	0.91666013	0.08333985
25	0.814578	0.18542199
26	0.8118246	0.18817538
27	0.8217389	0.17826113
28	0.91612273	0.08387723
29	0.8154413	0.18455872
30	0.84799993	0.15200001
31	0.91494226	0.08505773
32	0.91500336	0.084996596
33	0.8209585	0.17904152
34	0.8016084	0.19839163
35	0.91641486	0.08358511
36	0.8304604	0.16953959
37	0.91499746	0.08500249
38	0.91631866	0.08368134
39	0.82601005	0.17398992
40	0.79615605	0.203844
41	0.80172205	0.1982779
42	0.9139267	0.08607332
43	0.91552335	0.08447667
44	0.915564	0.084436
45	0.8382833	0.16171671

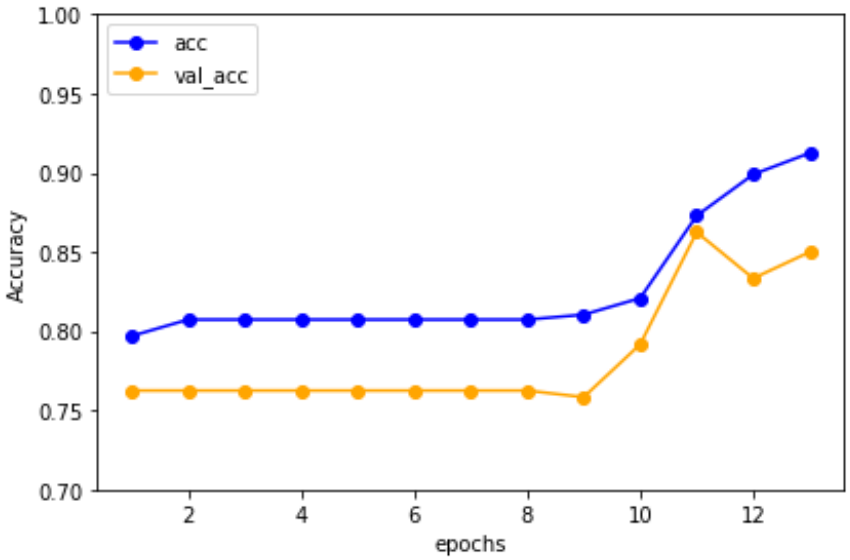
Σχήμα 32 : Αποτελέσματα BERT

Το μοντέλο εξάγει τα αποτελέσματα σε ένα αρχείο tsv όπου κάθε στήλη αντιπροσωπεύει την κάθε κλάση και όπως φαίνεται όλα τα δεδομένα ταξινομήθηκαν με ποσοστό πάνω από 70% ως αληθή. Οι δοκιμές έγιναν με 1200 από τα 7224 στοιχεία από τα οποία 242 ήταν ψευδείς ειδήσεις. Έγιναν επιπλέον δοκιμές με περισσότερα και λιγότερα στοιχεία από το σύνολο όμως το μοντέλο έπιανε πάντα την απόλυτη βάση. Πιθανόν αυτό να οφείλεται στο ότι το σύνολο δεδομένων είναι πολύ μικρό και τα στοιχεία την μίας κλάσης πολύ λιγότερα.

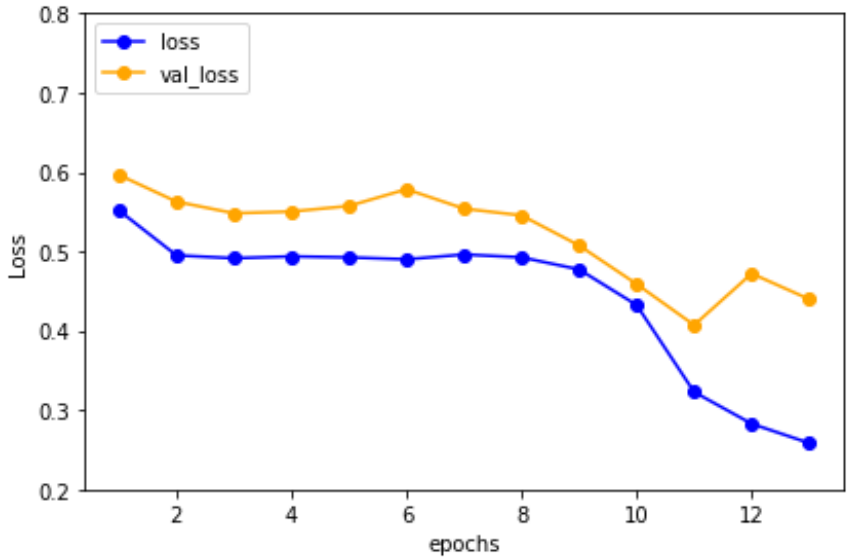
Υβριδικό Δίκτυο

Σε αυτό το δίκτυο έγινε χρήση προ-εκπαιδευμένων αναπαραστάσεων των λέξεων με τα οποία εκπαιδεύτηκαν ένα συνελικτικό στρώμα, ένα συγκεντρωτικό στρώμα, ένα στρώμα LSTM και δύο πλήρη συνδεδεμένα στρώματα, το τελευταίο από τα οποία αποτελούνταν από 2 νευρώνες και λειτουργεί ως στρώμα εξόδου. Στην αρχή του δικτύου έγινε χρήση των προ-εκπαιδευμένων αναπαραστάσεων GloVe(Global Vectors for Word Representations) της αντίστοιχης εργασίας του Stanford [36] με τις αναπαραστάσεις των 100 διανυσμάτων ανά λέξη και αμέσως μετά ενός στρώματος κανονικοποίησης dropout με ρυθμό 0.5. Για την προ-επεξεργασία έγινε η μετατροπή σε πεζά, αντικατάσταση συνδέσμων δικτύου με του χαρακτήρες “http” και η μετατροπή σε αριθμούς από το TreebankWordTokenizer. Η συνάρτηση κόστους που χρησιμοποιήθηκε ήταν η binary cross entropy από την βιβλιοθήκη του keras και για την βελτιστοποίηση τα καλύτερα αποτελέσματα είχε ο Adam. Για την εύρεση των καλύτερων παραμέτρων χρησιμοποιήθηκε η μέθοδος του grid search με παραμέτρους τον αριθμό των νευρώνων στα CNN και LSTM στρώματα και τον βελτιστοποιητή. Δεν παρατηρήθηκε κάποια ιδιαίτερη διαφορά ανάμεσα τιμές 100-500 για του νευρώνες ωστόσο αν οι τιμές ανέβαιναν παραπάνω το δίκτυο αδυνατούσε πλέον να ξεπεράσει την βάση και επιπλέον ο Adam παρείχε τα καλύτερα αποτελέσματα για αυτό και προτιμήθηκε. Τελικά τα αποτελέσματα που παρουσιάζονται έγιναν με την χρήση 200 νευρώνων στα CNN και LSTM στρώματα, 1000 νευρώνες στο πλήρες συνδεδεμένο στρώμα και συνέλιξη μεγέθους 3.

Για την εκπαίδευση επιλέχθηκε ένα υποσύνολο του γενικού συνόλου το οποίο αποτελούνταν από 1200 στοιχεία από τα οποία έγινε εκπαίδευση με το 80% αυτών και 20% για την γενίκευση. Μετά την επιλογή των 1200 στοιχείων, έγιναν επαναλαμβανόμενες δοκιμές πάνω σε αυτά αφού διαχωρίζονταν ξανά σε σύνολα εκπαίδευσης και γενίκευσης. Τα αποτελέσματα που προέκυψαν ήταν από την εξαγωγή των μέσων όρων των αποτελεσμάτων της επαναλαμβανόμενης διαδικασίας:



Σχήμα 33 : Ακρίβεια στο υβριδικό δίκτυο



Σχήμα 34 : Σφάλμα στο υβριδικό δίκτυο

Ο μέσος όρος των στοιχείων γενίκευση ήταν 183 αληθή και 57 ψευδή από τα οποία 177 από τα αληθή και 27 από τα ψευδή ταξινομήθηκαν σωστά. Αυτό οδηγεί σε ένα ποσοστό ακρίβειας 85% με βάση που αγγίζει ανέρχεται στο 76.25%. Το specificity ισούται με 0,473, το F1 score με 0.6 ενώ το recall για τα ψευδή στοιχεία είναι 0.82. Λόγω του ότι τα δεδομένα που αντιπροσώπευαν την αληθή κλάση τα στοιχεία είναι διαφορετικά σε κάθε επανάληψη του αλγορίθμου τα αποτελέσματα παρουσιάζουν ένα ποσοστό διαφοροποίησης επίσης.

6.2 Συμπεράσματα

Έγιναν δοκιμές με 4 διαφορετικά δίκτυα, το Character Level CNN, το Word2Vec, το B.E.R.T. και ένα υβριδικό δίκτυο με συνδυασμό LSTM και CNN. Σε κάθε δίκτυο ακολουθήθηκε διαφορετική προσέγγιση όσον αφορά την προεπεξεργασία του συνόλου των δεδομένων όπως αναφέρθηκε στην ανάλυση του προηγούμενου κεφαλαίου, ενώ έγινε χρήση του Dropout για αποφυγή της υπερεκπαίδευσης και τεχνητής αύξησης των δεδομένων για την εξαγωγή καλύτερων αποτελεσμάτων. Τα τελικά αποτελέσματα για την ανάκληση των 4 δικτύων είναι:

	Loss	Accuracy	Specificity
Character Level CNN	0.4460	81.25%	-
Word2Vec	0.6895	70%	-
B.E.R.T	0.4789	84%	-
Hybrid	0.4071	85%	0.4736

Πίνακας 1 : Αποτελέσματα

Το σφάλμα του B.E.R.T είναι υπολογισμένο με την μέθοδο categorical cross entropy ενώ των υπολοίπων βάση της binary cross entropy. Το υβριδικό μοντέλο κατάφερε γενικά καλά αποτελέσματα δεδομένου του μικρού συνόλου δεδομένων. Λόγω του μικρού συνόλου των δεδομένων και του τυχαίου διαχωρισμού σε εκπαίδευση και ανάκληση, η κλάση με τις ψευδείς ειδήσεις αποτελούσε συνήθως το 15% με 30% του συνόλου εκπαίδευσης, οπότε παρά το γεγονός πως τα πρώτα 3 μοντέλα υπόκεινται σε υπερεκπαίδευση παρουσιάζουν διαφορετικά αποτελέσματα λόγω της διακύμανσης των αναλογιών.

6.3 Περαιτέρω Ανάπτυξη

Τέλος παρουσιάζονται κάποιοι στόχοι για την περαιτέρω ανάπτυξη της πτυχιακής εργασίας:

- Αρχικά το μοντέλο θα μπορούσε να δοκιμαστεί σε κάποιο μεγαλύτερο σε πλήθος σύνολο δεδομένων με καλά ορισμένες τάξεις από κάποια αυθεντία.
- Η χρήση κάποιου διαφορετικού δικτύου για την παραγωγή καλύτερων αποτελεσμάτων. Πρόσφατα η Microsoft ανακοίνωσε το MT-DNN το οποίο βασίζεται στην αρχιτεκτονική του BERT όμως πετυχαίνει σημαντικά καλύτερα αποτελέσματα σε παρόμοιες εργασίες.
- Χρήση διαφορετικών Frameworks όπως το ML-NET της Microsoft

Επίλογος

Σε αυτό το κεφάλαιο έγινε αναφορά στα αποτελέσματα που παρουσίασαν διαφορετικά δίκτυα πάνω στη αναγνώριση ψευδών ειδήσεων στο σύνολο δεδομένων που χρησιμοποιήθηκε. Καθώς και μία αναφορά σε επόμενα βήματα τα οποία θα μπορούσαν να ακολουθηθούν για περαιτέρω έρευνα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Alexandre Bovet and Henran A.Manske “Influence of fake news in Twitter during the 2016 US presidential election” in Nature Communication Journal, 2019

[2] Kaggle leaderboard: Bag of words Meets Bags of Popcorn. Available: https://www.kaggle.com/c/word2vec-nlp-tutorial/leaderboard

[3] Mehran Sahami Susan Dumaisy David Heckermany and Eric Horvitzy, “A Bayesian Approach to Filtering Junk E-Mail” in AAAI Workshop on Learning for Text Categorization, July 1998

[4] [4] Grigori Sidorov , Francisco Velasquez , Efstathios Stamatatos , Alexander Gelbukh and Liliana Chanona-Hernández, “Syntactic N-grams as Machine Learning Features for Natural Language Processing” Published in Expert Syst. Appl. 2014

[5] Xiang Zhang, Junbo Zhao and Yann LeCun, “Character-level Convolutional Networks for Text Classification” in Proceedings of the 28th International Conference on Neural Information Processing Systems, December 2015

[6] Tomas Mikolov, Kai Chen, Creg Corrado and Jeffrey Dean, “Efficient Estimation of Word Representations in Vector Space”, in Proceedings of the International Conference on Learning Representations, January 2013

[7] Alex Graves and Jurgen Schmidhuber, “Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks” from “Guide to OCR for Arabic Scripts”, pp.545-552

[8] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates and Andrew Y. Ng, “Deep Speech: Scaling up end-to-end speech recognition”, December 2014

[9] Xiangang Li, Xihong Wu, “Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition”, in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, April 2015

[10] Heiga Zen and Has¸im Sak, “Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis”, in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, in April 2015

[11] Haşim Sak, Andrew Senior, Kanishka Rao, Françoise Beaufays and Johan Schalkwyk, “Google voice search: faster and more accurate”, Google AI Blog, September 24, 2015

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, October 2018

[13] Simon Haykin, “Neural Networks and Learning Machines”, Pearson Education Ltd 2009

[14] Διαμαντάρας, Κ. (2007). Τεχνητά Νευρωνικά Δίκτυα.

[15] D. H. Hubel and T. N. Wiesel, “Receptive fields of single neurones in the cat's striate cortex”, *The Journal of Phycology*, pp 574-591, 1959

[16] Kunihiko Fukushima, “Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position”, *Biol. Cybernetics 36*, pp 192-202, 1980

[17] Yann Lecun, Y.Bengio, Leon Bottou, Patrick Haffner, “Gradient-Based Learning Applied to Document Recognition”, in Proceedings of the IEEE, December 1998

[18] Github, “Understanding Convolutions”, 13 July 2014

[19] Παπαδολοπουλος Αθανασιος, “Συνελικτικά Νευρωνικά Δίκτυα στην Υπολογιστική Όραση”, 22 Μαΐου 2016

[20] http://cs231n.github.io/convolutional-networks/

[21] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, in Advances in neural information processing systems, January 2012

[22] A. Graves, Navdeep Jaitly, “Towards end-to-end speech recognition with recurrent neural networks”, January 2014

[23] Martin Sundermeyer, Ralf Schluter and Hermann Ney, “The RWTH Aachen University Neural Network Language Modeling Toolkit”, 15th Annual Conference of the International Speech Communication Association, 15 September 2014

[24] http://karpathy.github.io/2015/05/21/rnn-effectiveness/

- [25] Sepp Hochreiter and Jurgen Schmidhuber, “LONG SHORT-TERM MEMORY”, Neural Computation, December 1997

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, “Attention Is All You Need”, NIPS, June 2017

[27] https://towardsdatascience.com/transformer-attention-is-all-you-need-1e455701fdd9

[28] https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

[29] https://marutitech.com/top-8-deep-learning-frameworks/

[30] Julio Amador, Axel Oehmichen and Miguel Molina-Solana, “Characterizing Political Fake News in Twitter by its Meta-Data”, December 2017

[31] Jose Camacho-Collados and Mohammad Taher Pilehvar, “On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis”, July 2017

[32] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall and W. Philip Kegelmeyer “SMOTE: Synthetic Minority Over-sampling Technique” in Journal of Artificial Intelligence Research January 2002

[33] https://radimrehurek.com/gensim/models/word2vec.html

[34] https://github.com/google-research/bert

[35] https://nlp.stanford.edu/projects/glove/

[36] Sashank J. Reddi, Satyen Kale & Sanjiv Kumar, “On the convergence of Adam & Beyond”, May 2018