Νέες τεχνικές μηχανικής μάθησης για την επίλυση προβλημάτων ταξινόμησης

Γαζής Αθανάσιος

22 Φεβρουαρίου 2020

Περίληψη

Η παρούσα διπλωματική εργασία μελετά την χρήση Νευρωνικών Δικτύων για την ταξινόμηση κειμένων με κριτήριο το αν αποτελούν υβριστικά η/και προσβλητικα. Επίσης παρουσιάζονται δύο πιθανές εφαρμογές νευρωνικών δικτύων τέτοιου τύπου. Η δομή της εργασίας έχει ως εξής:

Το Κεφάλαιο 1 αποτελεί μια σύντομη εισαγωγή΄στους τομείς και τις εφαρμογές της τεχνητής νοημοσύνης και των νευρωνικών δικτύων. Στο Κεφάλαιο 2 παρουσιάζονται οι κυριότερες εξελίξεις στην ιστορία της τεχνητής νοημοσύνης. Στο Κεφάλαιο 3 παρουσιάζονται συνοπτικά οι βασικές αρχές λειτουργίας των νευρωνικών δικτύων. Στο Κεφάλαιο 4 αναφέρονται οι βασικές τεχνικές που χρησιμοποιούνται στην επεξεργασία φυσικής γλώσσας. Στο Κεφάλαιο 5 εξετάζονται τα δεδομένα που χρησιμοποιήθηκαν για την εκπαίδευση των δικτύων, οι μέθοδοι που χρησιμοποιήθηκαν και οι αρχιτεκτονικές τον δικτύων. Στο Κεφάλαιο 6 βλέπουμε αρχικά τα αποτελέσματα που πήραμε από την εκπαίδευση των δικτύων. Στη συνέχεια συγκρίνονται οι αναπαραστάσεις των λέξεων σε διανύσματα αναπτύχθηκαν και τέλος παρουσιάζονται δύο εφαρμογές των ταξινομητών σε δεδομένα που συλλέχθηκαν από το Twitter. Το Κεφάλαιο 7 αποτελεί τον επίλογο όπου παρατίθενται τα συμπεράσματα και σχόλια από όσα είδαμε στην παρούσα εργασία.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

Η κατασκευή μηχανών με την ικανότητα να επιλύουν προβλήματα που κλασσικά απαιτούν την ανθρώπινη νοημοσύνη για την επίλυση τους είναι μια επιδίωξη της ανθρωπότητας που προϋπάρχει των ηλεκτρονικών υπολογιστών. Τα τελευταία χρόνια βέβαια με την ανάπτυξη και την εκθετική πρόοδο των ηλεκτρονικών υπολογιστών η επιδίωξη αυτή των ανθρώπων έχει σε ένα βαθμό εκπληρωθεί.

Προβλήματα όπως η αναγνώριση προσώπων και αντικειμένων σε εικόνες/βίντεο και η αυτόνομη οδήγηση, τα οποία μέχρι πριν λίγες δεκαετίες άνηκαν στη σφαίρα της επιστημονικής φαντασίας και σήμερα αποτελούν πραγματικότητα παρά τις όποιες ατέλειες τους.

Η πρόοδος αυτή οφείλεται στη ταυτόχρονη ανάπτυξη της ισχύος των ηλεκτρονικών υπολογιστών καθώς και των αλγορίθμων που χρησιμοποιούνται για την επίλυση των προβλημάτων. Η υπολογιστική ισχύς ενός σύγχρονου οικιακού υπολογιστή υπερβαίνει αυτή ενός υπερυπολογιστή της δεκαετίας του 1980. Οι εξελίξεις αυτές κατέστησαν δυνατό να εφαρμοστούν προσεγγίσεις όπως αυτή των νευρωνικών δικτύων που παλαιότερα θεωρούνταν υπολογιστικά ασύμφορες.

Οι προσεγγίσεις αυτές αποδείχτηκαν αποτελεσματικές για την επίλυση εξαιρετικά πολύπλοκων προβλημάτων με αποτέλεσμα να υπάρξει μια ραγδαία ανάπτυξη στον τομέα των νευρωνικών δικτύων. Η συστηματική μελέτη έκανε δυνατή την ανάπτυξη εξειδικευμένων τεχνικών για

κάθε είδος προβλήματος και υπάρχουν συνεχώς καινούργιες εξελίξεις στον τομέα.

Ένα από τα προβλήματα που έχει εφαρμοστεί η μέθοδος των νευρωνικών δικτύων με πολλά υποσχόμενα αποτελέσματά είναι αυτό της επεξεργασίας φυσικής γλώσσας (Natural Language Processing). Στο πεδίο αυτό περιλαμβάνονται εφαρμογές που εκτείνονται από την εύρεση του θέματος ή συναισθημάτων που εκφράζονται σε ένα κείμενο μέχρι την παραγωγή κειμένου είτε με τη μορφή άρθρων είτε ως chatbots τα οποία είναι προγράμματα που έχουν την ικανότητα σε ένα βαθμό να μιμούνται ανθρώπους συνομιλητές. Οι εφαρμογές αυτές βρίσκονταν σε εμβρυικό στάδιο στις αρχές της δεκαετίας αλλά πλέον παρουσιάζουν συνεχώς καινούργια επιτεύγματα.

Στην παρούσα εργασία εξετάζεται ένα πρόβλημα που εντάσσεται στα προβλήματα επεξεργασίας φυσικής γλώσσας. Συγκεκριμένα θα εξετάζεται η απόδοση διαφορετικών μοντέλων νευρωνικών δικτύων στην ταξινόμηση σχολίων σε υβριστικά (τοξικά) ή όχι και στην συνέχεια κάποιες πιθανές εφαρμογές που μπορούν να έχουν αυτά τα μοντέλα σε δεδομένα που προέρχονται από τα μέσα κοινωνικής δικτύωσης. Για την σύγκριση της απόδοσης των νευρωνικών δικτύων χρησιμοποιήθηκαν και δύο μοντέλα βασισμένα στη μέθοδο Naive Bayes.

Κεφάλαιο 2

Ιστορία της Μηχανικής Μάθησης

Σε αυτό το κεφάλαιο θα παρουσιαστούν συνοπτικά καθοριστικά γεγονότα στην ιστορία της τεχνητής νοημοσύνης και των νευρωνικών δικτύων με χρονολογική σειρά. Η ιστορία των τεχνολογιών αυτών ξεκινά αρκετά νωρίτερα από την δεκαετία μας κατά την οποία γνώρισαν τεράστια άνθηση και πληθώρα εφαρμογών.

Ουσιαστικά η ιστορία της μηχανικής μάθησης χωρίζεται σε τρεις χρονικές περιόδους όπου στο ενδιάμεσο υπήρχε μείωση του ενδιαφέροντος για τον τομέα.

Η πρώτη περίοδος περιλαμβάνει τις δεκαετίες 1940-1950 περίοδος όπου ξεκινά και η επιστήμη των υπολογιστών. Ταυτόχρονα μπαίνουν και τα θεμέλια για τον τομέα της μηχανικής μάθησης. Η δεύτερη περίοδος εντοπίζεται στις αρχές της δεκαετίας του '80 όπου αναζωπυρώθηκε το ενδιαφέρον για τα νευρωνικά δίκτυα και την τεχνητή νοημοσύνη. Ως τρίτη περίοδος αναφερόμαστε στη σύγχρονη εποχή των νευρωνικών δικτύων και του Deep Learning η οποία μπορούμε να πούμε ότι ξεκίνησε γύρω στο 2012. Σε αυτή την περίοδο που διανύουμε βλέπουμε ταχεία ανάπτυξη κλάδου με αναρίθμητες πρακτικές εφαρμογές και συνεχή πρόοδο στην ικανότητες των αλγορίθμων να επιλύουν προβλήματα που μέχρι πριν λίγα χρόνια φάνταζε αδύνατο να επιλυθούν.

2.1 Πρώτη Περίοδος (1940-1950)

1943 Η ιστορία των Νευρωνικών δικτύων ξεκινά το 1943 όπου οι Warren Mcculloch και Walter Pitts [1] ,νευροφυσιολόγος και μαθηματικός αντίστοιχα, μελέτησαν σε εργασία τους πως λειτουργούν τα νευρικά δίκτυα του εγκεφάλου και αποφάσισαν να φτιάξουν ένα από ηλεκτρικά κυκλώματα.

1950 Ο Άγγλος μαθηματικός Alan Turing[2] δημοσιεύει την εργασία του Computing machines and intelligence όπου περιγράφει μια διαδικασία για να κρίνουμε αν μια μηχανή μπορεί να κατανοήσει την ανθρώπινη γλώσσα. Η διαδικασία αυτή αργότερα έγινε γνωστή ως Turing Test.

1956 Ο Αμερικάνος επιστήμονας της πληροφορικής Arthur Samuel[3] παρουσιάζει το πρώτο πρόγραμμα μηχανικής μάθησης το οποίο έχει τη δυνατότητα να μαθαίνει να παίζει το παιχνίδι ντάμα.

1956 Λαμβάνει χώρα το Dartmouth Workshop το οποίο ήταν ένα συνέδριο που θεωρείται πως καθιέρωσε την Τεχνητή Νοημοσύνη ως επιστημονικό πεδίο.

1958 Ο Αμερικανός ψυχολόγος F.Rosenblatt δημοσιεύει την εργασία του με τίτλο The Perceptron: A probalistic model for information storage and organization in the brain[5] οπού εισήγαγε την έννοια του perceptron το οποίο είναι ουσιαστικά ένα πολύ απλό τεχνητό νευρωνικό δίκτυο που έχει την ικανότητα να κάνει δυαδική ταξινόμηση.

1959 Οι ηλεκτρολόγοι μηχανικοί Bernard Widrow και Marcian Hoff[6] κατασκεύασαν δύο μοντέλα με ονόματα ADELINE και MADELINE (Multiple Adaptive Linear Elementes) τα οποία είχαν την δυνατότητα δοσμένης μίας ακολουθίας bit να προβλέπουν το επόμενο. Το μοντέλο MADELINE ήταν κατά μία έννοια το πρώτα σύστημα τεχνητής νοημοσύνης που εφαρμόστηκε σε πραγματικό πρόβλημα και συγκεκριμένα εφαρμόστηκε στη μείωση του θορύβου των τηλεφωνικών γραμμών.

Όλα τα παραπάνω γεγονότα αν και είχαν ακαδημαϊκό ενδιαφέρον είχαν πολύ λίγες εφαρμογές σε προβλήματα του πραγματικού κόσμου. Έτσι τις επόμενες δυο δεκαετίες έγινε ελάχιστη πρόοδος στον τομέα γεγονός που ίσως οφείλεται και στην περιορισμένη ισχύ που είχαν οι υπολογιστές της εποχής

2.2 Δεύτερη Περίοδος (1980)

1982 Το ενδιαφέρον για τα νευρωνικά δίκτυα αναζωπυρώνεται μετά από μια εργασία του John Hopfield[7] με τίτλο Neural networks and physical systems with emergent collective computational abilities όπου περιγράφει την κατασκευή νευρωνικών δικτύων διπλής κατεύθυνσης. Δηλαδή δίκτυα όπου η έξοδος τους επαναχρησιμοποιείται ως είσοδος με σκοπό την εκπαίδευση του δικτύου.

1986 Οι Rumelhart, Hinton και Williams[8] του τμήματος ψυχολογίας του πανεπιστημίου Stanford αποφασίζουν να επεκτείνουν ένα αλγόριθμο των Bernard Widrow και Marcian Hof και ουσιαστικά χρησιμοποίησαν τη μέθοδο Back Propagation που χρησιμοποιείται και σήμερα στην εκπαίδευση Νευρωνικών Δικτύων.

Τα χρόνια που ακολούθησαν αν και η έρευνα σε ακαδημαϊκό επίπεδο συνεχιζόταν δεν υπήρξε κάποια σημαντική πρόοδος με αποτέλεσμα ο κλάδος να συγκεντρώνει περιορισμένο ενδιαφέρον.

2.3 Τρίτη Περίοδος (2012-Σήμερα)

Από το 2012 και έπειτα οι εξελίξεις στον τομέα της τεχνητής νοημοσύνης είναι ραγδαίες. Βέβαια τα προηγούμενα χρόνια οι ακαδημαϊκές έρευνες συνεχίζονταν πάνω στο αντικείμενο αλλά από το σημείο αυτό και έπειτα ο τομέας αναπτύσσεται ταχύτατα.

Στην ραγδαία εξέλιξη του τομέα συνέβαλε και το ενδιαφέρον εταιριών κολοσσών της πληροφορικής που είχαν την δυνατότητα να διαθέσουν τεράστια ποσά με σκοπό την έρευνα πάνω στον τομέα.

Κάποια μεγάλα project που έδειξαν τις δυνατότητες της τεχνητής νοημοσύνης είναι τα παρακάτω.

2012 Οι ερευνητική ομάδα της Google, Google Brain[9], κατασκευάζει ένα νευρωνικό δίκτυο για την αναγνώριση μοτίβων σε εικόνες και βίντεο χρησιμοποιώντας μοντέλα μη επιβλεπόμενης μάθησης. Χρησιμοποιήθηκε για την αναγνώριση αντικειμένων σε βίντεο του Youtube.

2012 Οι Krizhevsky, Sutskever και Hinton[10] παρουσιάζουν το δίκτυο AlexNet το οποίο κερδίζει με μεγάλη διαφορά το διαγωνισμό ImageNet που έχει σκοπό την ταξινόμηση εικόνων. Η επικράτηση του δικτύου αυτού έδειξε τις δυνατότητες που έδινε η εκμετάλλευση των καρτών γραφικών των μοντέρνων υπολογιστών και των Convolutional Neural Networks στα προβλήματα που αφορούσαν την επεξεργασία εικόνας. Επίσης το δίκτυο AlexNet χρησιμοποίησε ως συνάρτηση ενεργοποίησης την συνάρτηση ReLu κι όχι την σιγμοειδή ή την υπερβολική εφαπτομένη όπως ήταν η συνήθης πρακτική έως τότε.

2015 Η Google αγοράζει την εταιρία DeepMind η οποία έχει φτιάξει προγράμματα ικανά να παίζουν video games σε επίπεδο αντίστοιχο των ανθρώπων και στην συνέχεια ανέπτυξε το μοντέλο AlphaGo που κατάφερε να κερδίσει επαγγελματίες παίχτες στο παιχνίδι Go, ένα παιχνίδι που θεωρείται από τα δυσκολότερα επιτραπέζια.

2015 Η Amazon δημιουργεί το Amazon Machine Learning Platform κάτι που δείχνει το ενδιαφέρον που έχουν οι μεγάλες εταιρίες για τον πεδίο της τεχνητής νοημοσύνης.

Από το 2015 εώς σήμερα οι εξελίξεις στον τομέα της τεχνητής νοημοσύνης είναι ραγδαίες και η πρόοδος συνεχής. Έτσι η τεχνητή νοημοσύνη από ένα ακαδημαϊκό αντικείμενο κατέστη ένα εργαλείο με πληθώρα εφαρμογών στον πραγματικό κόσμο.

Κεφάλαιο 3

Βασικές αρχές λειτουργίας Νευρωνικών Δικτύων

Τα σύγχρονα νευρωνικά δίκτυα μπορεί να είναι εξαιρετικά πολύπλοκα και να περιέχουν από χιλιάδες έως και εκατομμύρια παραμέτρους, οι βασικές αρχές λειτουργίας τους όμως είναι σχετικά απλές. Κάθε νευρωνικό δίκτυο ουσιαστικά είναι μια συνάρτηση που δέχεται μια είσοδο ,η οποία μπορεί να είναι μιας ή περισσότερων διαστάσεων, και παράγει μια έξοδο, οι οποία πάλι μπορεί να είναι μίας ή περισσότερων διαστάσεων.

Σκοπός αυτής της συνάρτησης είναι δεδομένης της εισόδου να πάρουμε την επιθυμητή έξοδο κάτι που επιτυγχάνεται με τη διαδικασία της εκπαίδευσης. Ως εκπαίδευση ορίζεται η διαδικασία κατά την οποία εντοπίζονται οι βέλτιστες τιμές των παραμέτρων του δικτύου ώστε να πάρουμε την επιθυμητή έξοδο. Για να το καταφέρουμε υπολογίζουμε την έξοδο δεδομένης της εισόδου και βλέπουμε πόσο ''απέχει'' από την επιθυμητή έξοδο και στην συνέχεια αναπροσαρμόζουμε τις παραμέτρους του δικτύου και ελέγχουμε ξανά την έξοδο.

Η διαδικασία που περιγράφηκε παραπάνω ουσιαστικά είναι η ελαχιστοποίηση μια συνάρτησης. Παρακάτω θα δούμε πως από μια απλή τέτοια διαδικασία κατασκευάζονται τα σύγχρονα νευρωνικά δίκτυα.

3.1 Εύρεση Ελάχιστου Συνάρτησης

Για την εύρεση των ελαχίστων τιμών μιας συνάρτησης υπάρχει η αναλυτική μέθοδος και οι υπολογιστικές μέθοδοι. Σε συναρτήσεις με μικρό αριθμό παραμέτρων χρησιμοποιούμε την αναλυτική μέθοδο. Σε περιπτώσεις που η συνάρτηση έχει μεγάλο αριθμό παραμέτρων καταφεύγουμε σε υπολογιστικές μεθόδους.

Στην αναλυτική μέθοδος που αναφέραμε αρχικά εντοπίζονται τα κρίσιμα σημεία της συνάρτησης και στην συνέχεια στην εξέτάζεται αν αυτά αποτελούν τοπικά μέγιστα ή ελάχιστα. Κρίσιμα σημεία ονομάζονται τα σημεία όπου οι πρώτες παράγωγοι μιας συνάρτησης μηδενίζονται. Για μια συνάρτηση δύο μεταβλητών τα κρίσιμα σημεία βρίσκονται με την επίλυση των εξισώσεων:

$$\frac{\partial f}{\partial x} = 0$$
$$\frac{\partial f}{\partial y} = 0$$

Στην συνέχεια για να εξετάσουμε αν το κρίσιμο σημείο (x_i, y_i) αποτελεί τοπικό ελάχιστο υπολογίζουμε την ποσότητα:

$$D = \frac{\partial^2 f}{\partial x^2}\frac{\partial f^2}{\partial y^2} - (\frac{\partial^2 f}{\partial x \partial y})^2$$

Και για να είναι τοπικό ελάχιστο ένα κρίσιμο σημείο πρέπει να ισχύουν οι σχέσεις:

$$D > 0$$
$$\frac{\partial^2 f}{\partial x^2} > 0$$

Σε περιπτώσεις που η συνάρτηση μας έχει παραπάνω μεταβλητές η παραπάνω διαδικασία αυξάνεται σε πολυπλοκότητα. Έτσι καταφεύγουμε σε υπολογιστικές μεθόδους.
Η πιο απλή μέθοδος που χρησιμοποείται στην εκπαίδευση νευρωνικών δικτύων είναι η μέθοδος Gradient Descent. Σε αυτή την μέθοδο χρειάζεται να υπολογίσουμε μόνο της παραγώγους πρώτης τάξης της συνάρτησης προς ελαχιστοποίηση. Τα βήματα που ακολουθούμε είναι τα εξής:

Βήμα 1: Ξεκινάμε με τυχαίες τιμές για τα (x, y) τις οποίες θα ονομάσουμε (x_0, y_0) και υπολογίζουμε τις τιμές των παραγώγων για τα (x_0, y_0) καθώς και την τιμή της συνάρτησης για αυτές τις τιμές.

Βήμα 2: Ενημερώνουμε τις τιμές των (x, y) σύμφωνα με τις παρακάτω σχέσεις:

$$x_1 = x_0 - a * \frac{\partial f}{\partial x_0}$$

$$y_1 = y_0 - a * \frac{\partial f}{\partial y_0}$$

Όπου a σταθερά που στη βιβλιογραφία που αναφέρεται στα νευρωνικά δίκτυα ονομάζεται learning rate. Η τιμή της επιλέγεται ανάλογα με το πρόβλημα αλλά συνήθως είναι ισχύει a<1. Η παραπάνω διαδικασία επαναλαμβάνεται για τις νέες τιμές (x_1, y_1) είτε για προκαθορισμένο αριθμό βημάτων είτε έως ότου υπάρξει στασιμότητα στις τιμές της συνάρτησης f(x,y). Τα κριτήρια για κριθεί αν η συνάρτηση εμφανίζει στασιμότητα εξαρτώνται από το εκάστοτε πρόβλημα.

3.2 Γραμμική παλινδρόμηση

Μία πρώτη προσέγγιση στους αλγόριθμους Μηχανικής Μάθησης μπορεί να γίνει χρησιμοποιώντας την τεχνική της γραμμικής παλινδρόμησης.

Το μοντέλο της γραμμικής παλινδρόμησης λειτουργεί ως εξής:
Για m ζεύγη τιμών $\{x_i, y_i\}$ αναζητούμε συνάρτηση:

$$f(x) = wx + b$$

Για την οποία θα ισχύει ιδανικά:

$$f(x_i) = y_i$$

Οι τιμές $f(x_i)$ ονομάζονται και προβλέψεις ή εκτιμήσεις οι οποίες συμβολίζονται και ως $\hat{y}_i$.

Στην πραγματικότητα αυτό δεν γίνεται ποτέ λόγω θορύβου στις μετρήσεις, εξαρτήσεων από άλλες μεταβλητές κ.ά.
Αυτό που κάνουμε τελικά είναι να βρούμε την εξίσωση που ελαχιστοποιεί το σφάλμα των προβλέψεων μας. Ένας τρόπος να ποσοτικοποιήσουμε το σφάλμα ώστε να το ελαχιστοποιήσουμε είναι να αθροίσουμε τα τετράγωνα των αποστάσεων (ώστε να αποφύγουμε μηδενισμό σε περίπτωσή αντίθετων τιμών) των προβλέψεων απο τις μετρήσεις και στην συνέχεια να ελαχιστοποιήσουμε το άθροισμα αυτό. Ορίζουμε την συνάρτηση:

$$L = \sum_{i=1}^{m} (y_i - f(x_i)^2$$

Χρησιμοποιώντας την παραπάνω συνάρτηση, ο αναλυτικός τρόπος για να βρούμε την καλύτερη δυνατή λύση είναι να λύσουμε το σύστημα:

$$\frac{\partial L}{\partial w} = 0$$
$$\frac{\partial L}{\partial b} = 0$$

Λύνοντας το σύστημα καταλήγουμε στις παρακάτω σχέσεις

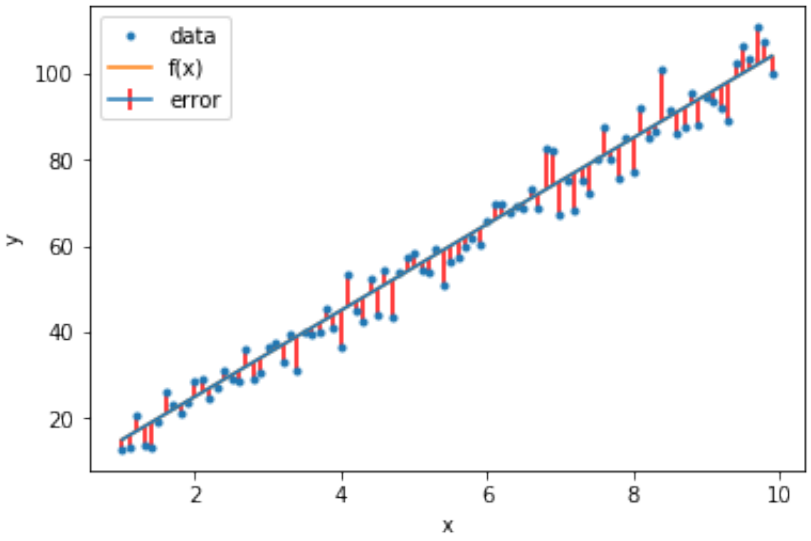
$$b = \frac{\sum_{i=1}^{m} y_i \sum_{i=1}^{m} x_i^2 - \sum_{i=1}^{m} x_i \sum_{i=1}^{m} x_i y_i}{m \sum_{i=1}^{m} x_i^2 - (\sum_{i=1}^{m} x_i)^2}$$

$$w = \frac{m \sum_{i=1}^{m} y_i x_i - \sum_{i=1}^{m} x_i \sum_{i=1}^{m} y_i}{m \sum_{i=1}^{m} x_i^2 - (\sum_{i=1}^{m} x_i)^2}$$

Όπου αντικαθιστώντας τις τιμές των x και y υπολογίζουμε τους συντελεστές.
'Ενας άλλος τρόπος να προσεγγίσουμε το πρόβλημα είναι να ξεκινήσουμε από τυχαίες τιμές των w και b και να υπολογίζουμε τις παραγώγους $\frac{\partial L}{\partial w}$ και $\frac{\partial L}{\partial b}$ και στην συνέχεια να ενημερώνουμε τις τιμές σύμφωνα με τις εξισώσεις:

$$w = w - \alpha * \frac{\partial L}{\partial w}$$
$$b = b - \alpha * \frac{\partial L}{\partial b}$$

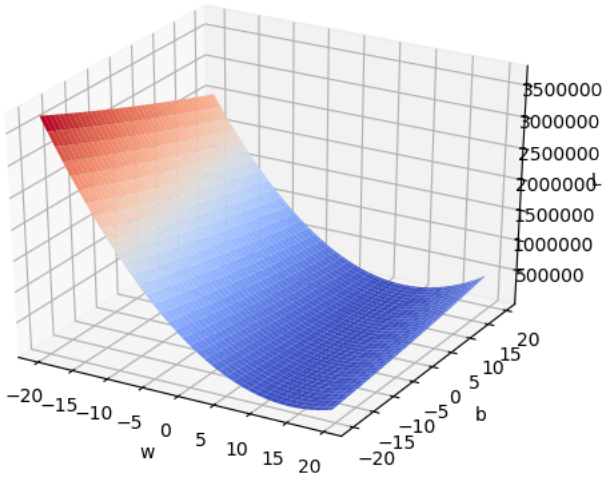
Ας συγκρίνουμε τις δυο μεθόδους χρησιμοποιώντας ένα τεχνητό σύνολο μετρήσεων.
Δημιουργήσαμε σημεία από το ένα ως το 10 με βήμα ένα και για να προσομοιώσουμε πραγματικές μετρήσεις προσθέσαμε γκαουσιανό θόρυβο.
Η συνάρτηση που χρησιμοποιούμε είναι η $y = 10x + 5$.



Σχήμα 3.1: Το διάγραμμα της συνάρτησης $y = 10x + 5, x \in [1, 10]$,οι "μετρήσεις" και το σφάλμα τους

Χρησιμοποιώντας την μέθοδο ελαχίστων τετραγώνων παίρνουμε τις τιμές $w = 10.0739, b = 3.8742$ στρογγυλοποιημένες στα 4 δεκαδικά.
Ας δούμε τώρα πως θα λειτουργήσει η υπολογιστική μέθοδος.

Η συνάρτηση κόστους L έχει την ακόλουθη γραφική παράσταση όπως στο παρακάτω σχήμα



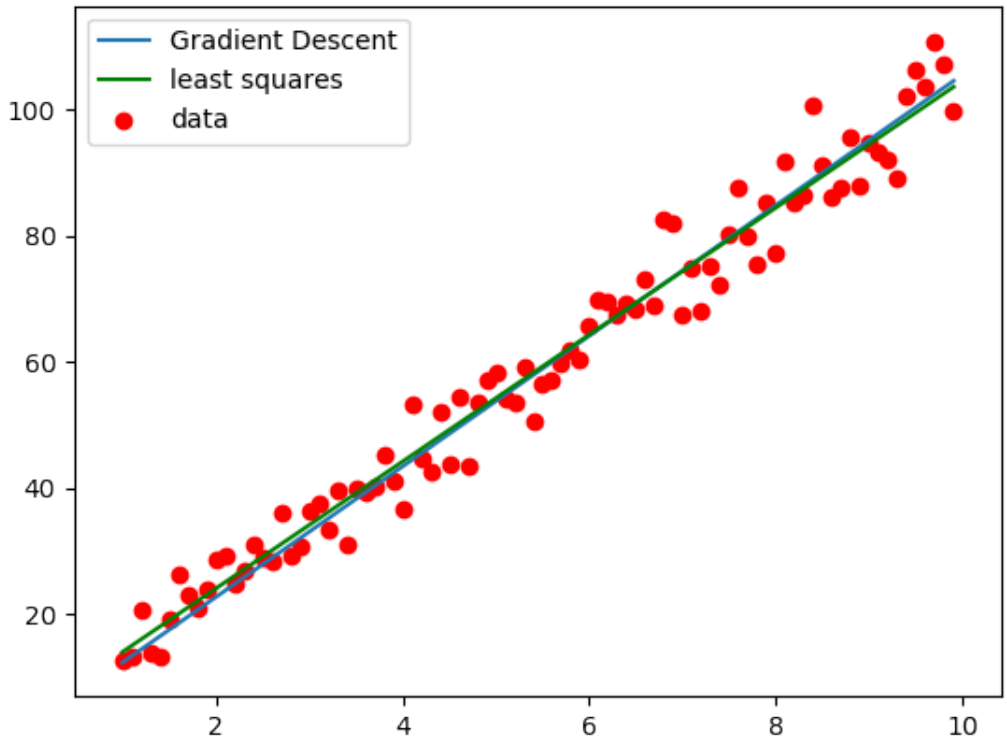
Σχήμα 3.2: Η συνάρτηση $L(w, b), w, b \in [-20, 20]$

Οι τιμές που παίρνουμε χρησιμοποιώντας τις σχέσεις

$$w = w - \alpha * \frac{\partial L}{\partial w}$$

$$b = b - \alpha * \frac{\partial L}{\partial b}$$

με αφετηρία το 0 και $\alpha = 0.01$ και 100 επαναλήψεις είναι $w = 10.3685, b = 1.9132$
Βλέπουμε μια σύγκριση στην προσαρμογή των δεδομένων που κάνει η μέθοδος ελαχίστων τετραγώνων καθώς και η μέθοδος Gradient Descent στο σχήμα 3.3



Σχήμα 3.3: Σύγκριση της προσαρμογής των δύο μεθόδων στα δεδομένα

Βλέπουμε ότι οι δύο μέθοδοι γραφικά απέδωσαν το ίδιο καλά ενώ η σύγκριση των παραμέτρων που μας απέδωσαν δίνει το πλεονέκτημα στη μέθοδο ελαχίστων τετραγώνων. Το πλεονέκτημα της υπολογιστικής μεθόδου δεν εντοπίζεται στην ακρίβεια αλλά στην ευκολότερη και καθολικότερη εφαρμογή της. Καθώς τα προβλήματα που αντιμετωπίζουμε και αυξάνουν σε πολυπλοκότητα είτε με την αύξηση των ανεξαρτήτων μεταβλητών είτε με την εμφάνιση μη γραμμικών εξαρτήσεων η εφαρμογή υπολογιστικών μεθόδων γίνεται απαραίτητη.

3.3 Λογιστική Παλινδρόμηση

Είδαμε στην περίπτωση της γραμμικής παλινδρόμησης ότι από ένα σύνολο δεδομένων που ακολουθούν μια γραμμική συνάρτηση μπορούμε να την υπολογίσουμε. Τώρα θα εξετάσουμε τι γίνεται στην περίπτωση που τα δεδομένα μας ανήκουν σε δύο κλάσεις.

Για την αντιμετώπιση τέτοιων προβλημάτων χρησιμοποιούμε την μέθοδο της Λογιστικής Παλινδρόμησης.Η λογιστική παλινδρόμηση είναι μια μέθοδος για την εύρεση μιας συνάρτησης $f(x_1, x_2, ...,$ όπου x_i η ανεξάρτητες μεταβλητές, που μπορεί ταξινομεί τα δεδομένα μας τα οποία σε δύο κλάσεις, που μπορούμε να συμβολίζουμε για παράδειγμα με τις τιμές $0, 1$.

Η συνάρτηση της λογιστικής παλινδρόμησης για δύο ανεξάρτητες μεταβλητές ορίζεται ως εξής:

$$f(x_1, x_2) = \frac{1}{1 + e^{-(w_1 * x_1 + w_2 * x_2 + b)}}$$

,όπου τα $w_1, w_2, b \in \Re$

Όπως και στην Γραμμική παλινδρόμηση κι εδώ χρειαζόμαστε μια συνάρτηση κόστους που να μετρά το πόσο σωστά είναι τα αποτελέσματα μας. Η συνάρτηση που θα χρησιμοποιήσουμε εδώ είναι η logistic loss η οποία για N μετρήσεις ορίζεται ως εξής:

$$L = -\frac{1}{N} \sum_{i=1}^{N} (y_i log f(x_i) + (1 - y_i) log(1 - f(x_i))$$

Για να βρούμε την κατάλληλη f για το πρόβλημα μας θα πρέπει να ελαχιστοποιήσουμε την συνάρτηση κόστους $L(w_1, w_2, b)$. Αυτό θα το κάνουμε με την υπολογιστική μέθοδο που χρησιμοποιήσαμε και πιο πάνω. Για να χρησιμοποιήσουμε την παραπάνω μέθοδο όμως πρέπει να υπολογίσουμε τις παραγώγους της L. Χρησιμοποιώντας τον κανόνα τις αλυσίδας παίρνουμε:

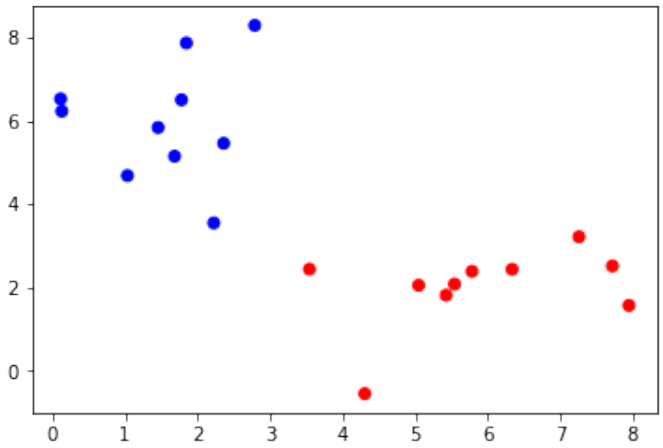
$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial f}\frac{\partial f}{\partial z}\frac{\partial z}{\partial w}$$

Η παραπάνω σχέση ισχύει αντίστοιχα για τα w_1, w_2, b.
Έτσι έχουμε:

$$\frac{\partial L}{\partial f} = -\frac{1}{N}\sum_{i=1}^{N}(y_i\frac{1}{f(x_i)} - (1-y_i)\frac{1}{1-f(x_i)})$$

$$\frac{\partial f}{\partial z} = f(z)(1-f(z))$$
$$\frac{\partial z}{\partial w_i} = x_i, \frac{\partial z}{\partial b} = 1$$

Έτσι μπορούμε να χρησιμοποιήσουμε τη υπολογιστική μέθοδο από που είδαμε και στην γραμμική παλινδρόμηση με σκοπό να βρούμε ένα ελάχιστο τις L.
Για να δοκιμάσουμε την Λογιστική παλινδρόμηση δημιουργήσαμε ένα δοκιμαστικό σετ μετρήσεων. Αυτό το φτιάξαμε παίρνοντας δύο σημεία στο επίπεδο x-y και παίρνοντας τυχαία σημεία κοντά τους.

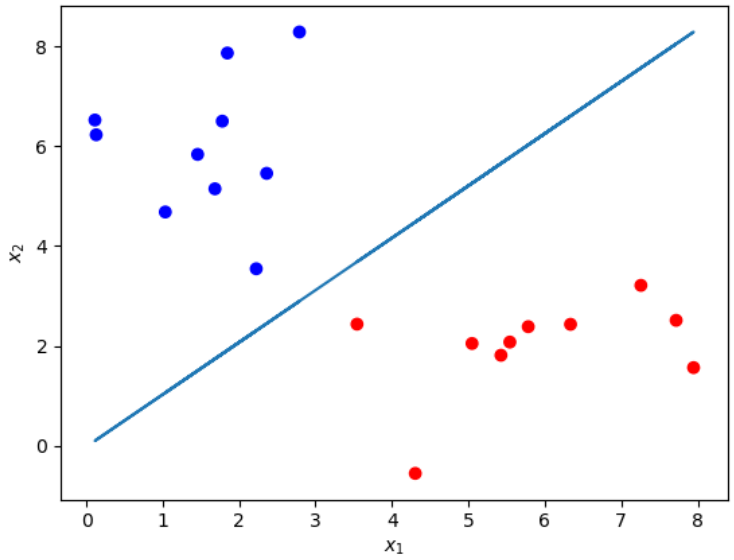


Σχήμα 3.4: Πήραμε τα σημεία (5,1) και (1,5) και τυχαία σημεία γκαουσιανής κατανομής με μέση τιμή και τυπική απόκλιση 1 και τους αναθέσαμε κλάσεις όπως φαίνεται σύμφωνα με τα χρώματα.

Αυτό που μένει είναι να εντοπίσουμε τα w_1, w_2, b για τα οποία θα παίρνουμε τα σωστά αποτελέσματα Τρέχοντας τον παραπάνω αλγόριθμο για 100 επαναλήψεις παίρνουμε τις τιμές $w_1 = 2.0725, w_2 = -1.9851, b = -0.0256$ και βάζοντας τις στην εξίσωση

$$f(x_1, x_2) = \frac{1}{1 + e^{-(w_1 * x_1 + w_2 * x_2 + b)}}$$

παίρνουμε σωστά αποτελέσματα για κάθε κλάση. Για άλλα 20 τυχαία σημεία που δημιουργήσαμε πήραμε σωστές προβλέψεις για τις κλάσεις.



Σχήμα 3.5: Τα σημεία μαζί με την ευθεία $w_1 x_1 + w_2 x_2 + b = 0$. Βλέπουμε ότι ευθεία που βρήκαμε δρα ως "σύνορο" μεταξύ των περιοχών τον δύο κλάσεων

3.4 Softmax Regression

Στη λογιστική παλινδρόμηση τα δεδομένα άνηκαν σε δύο κλάσεις. Στις περιπτώσεις που τα δεδομένα ανήκουν σε περισσότερες από δύο

κλάσεις χρησιμοποιείται η μέθοδος Softmax Regression. Ο συμβολισμός που επιλέχθηκε για τις παραμέτρους του προβλήματος είναι διαφορετικός από τις προηγούμενες ενότητες και θα φανεί χρήσιμος και στην περιγραφή των νευρωνικών δικτύων.

Η τεχνική Softmax Regression ουσιαστικά αποτελεί επέκταση της Logistic Regression. Στην προηγούμενη περίπτωση είχαμε δύο κλάσεις οπότε μια ευθεία αρκούσε για να διαχωρίσει τις κλάσεις (σχήμα 3.5). Στην περίπτωση που έχουμε k κλάσεις θα χρειαστούμε k ευθείες.

Θεωρούμε πως τα δεδομένα αποτελούνται από m μετρήσεις n ανεξάρτητων μεταβλητών οπότε αντιστοιχούν σε ένα πίνακα $\mathbf{X}$ διαστάσεων (mxn). Κάθε γραμμή του πίνακα $\mathbf{X}$ που στο εξής στα αναφέρουμε ως **μέτρηση** μπορούμε να την συμβολίσουμε ως ένα πίνακα-γραμμή διαστάσεων (1xn), επίσης θα συμβολίζουμε την i μέτρησή ως $x^{(i)}$. Για να ανατρέχουμε στα στοιχεία του πίνακα-γραμμή θα χρησιμοποιούμε το δείκτη j. Έτσι για παράδειγμα αν θέλουμε να ανατρέξουμε στη τρίτη ανεξάρτητη μεταβλητή της δέκατης μέτρησης πρέπει να πάμε στο στοιχείο $x_3^{(10)}$. Για κάθε πίνακα γραμμή $x^{(i)}$ μπορούμε να υπολογίσουμε τον επίσης πίνακα-γραμμή $z^{(i)}$ σύμφωνα με την εξίσωση:

$$z^{(i)} = x^{(i)} * w + b$$

Όπου οι πίνακες w και b:

$$w = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1k} \\ \vdots & \ddots & \vdots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{nk} \end{bmatrix}, b = \begin{bmatrix} b_1 & \dots & b_k \end{bmatrix}$$

Κάθε γραμμή του πίνακα w μαζί με το αντίστοιχο στοιχείο το πίνακα b αποτελούν τα βάρη που είχαμε δει στη λογιστική παλινδρόμηση. Με άλλα λόγια η Softmax Regression μοιάζει να αποτελείται από k διαφορετικές λογιστικές παλινδρομήσεις.
Μόλις υπολογίσουμε το z στην συνέχεια υπολογίσουμε τον πίνακα-γραμμή:

$$p^{(i)} = \frac{e^{z^{(i)}}}{\sum_{j=1}^{k} e^{z_j^{(i)}}}$$

Όπου καθένα από τα k στοιχεία του πίνακα p_i αντιστοιχεί στην πιθανότητα δίνει ο ταξινομητής μας στο i στοιχείο των μετρήσεων μας να ανήκει στην k κλάση.
Ορίσαμε την συνάρτηση του ταξινομητή τώρα ας ορίσουμε και μια συνάρτηση κόστους. Η συνάρτηση κόστους που χρησιμοποιούμε είναι η γενίκευση της συνάρτησης κόστους της λογιστικής παλινδρόμησης:

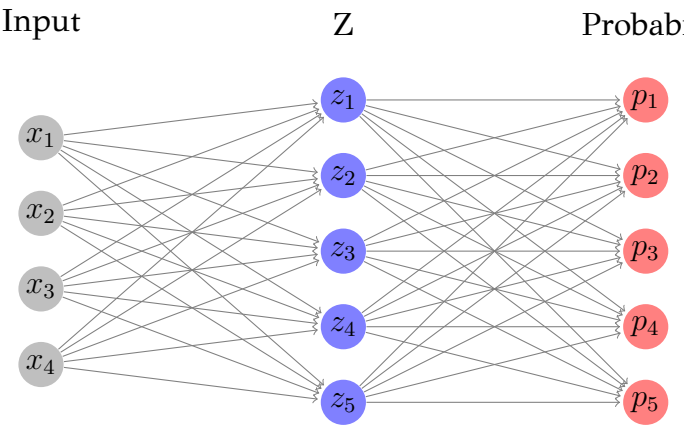
$$L = -(\frac{1}{m})\sum_{i=1}^{m}\sum_{j=1}^{k}(y_j^{(i)}log(p_j^{(i)})$$

Και τώρα μένει να βρούμε τις παραγώγους της L ώστε να μπορούμε να εφαρμόσουμε έναν αλγόριθμο εύρεσης ελάχιστου. Οι παράγωγοι που θέλουμε να υπολογίσουμε είναι οι $\frac{\partial L}{\partial w_{lj}}$ και $\frac{\partial L}{\partial b_j}$
Για την αποφυγή σύγχυσης με τους δείκτες έχουμε:
Για τις μετρήσεις χρησιμοποιούμε το δείκτη i και για πλήθος τους το m
Για τις κλάσεις χρησιμοποιούμε το δείκτη j και στο πλήθος τους αναφερόμαστε με k
Τέλος για τον αριθμό των ανεξαρτήτων μεταβλητών θα χρησιμοποιήσουμε το δείκτη l και για το πλήθος τους όπως έχουμε αναφέρει τον n.
Έτσι για τις πιο πάνω παραγώγους έχουμε:

$$\frac{\partial L}{\partial w_{lj}} = -\frac{1}{m}\sum_{i=1}^{m}(p_j^{(i)} - y_j^{(i)})x_l$$

$$\frac{\partial L}{\partial b_j} = -\frac{1}{m}\sum_{i=1}^{m}(p_j^{(i)} - y_j^{(i)})$$

'Ετσι μπορούμε να χρησιμοποιήσουμε το αλγόριθμο που είδαμε πιο πάνω για να βρούμε ένα ελάχιστο τις συνάρτησης



Σχήμα 3.6: Σχηματική αναπαράσταση της Softmax Regression. Η είσοδος(Input) εδώ έχει διάσταση ίση με 4 και το τα δεδομένα ανήκουν σε 5 κλάσεις. $x^{(i)}$ δημιουργείται το διανυσμα $z^{(i)}$ και στην συνέχεια υπολογίζονται οι πιθανότητες για κάθε κλάση στο διάνυσμα $p^{(i)}$.Το διάνυσμα x έχει διάσταση n ,όσες και η ανεξάρτητες μετβλητές και τα διανύσματα $z^{(i)}$,$p^{(i)}$ έχουν διάσταση k, όσες και η κλάσεις του προβλήματος.

3.5 Multi Layer Perceptron

Είδαμε στις προηγούμενες ενότητες πως με την τεχνική Softmax Regression μπορούμε να φτιάξουμε ένα ταξινομητή που να ταξινομεί τα δεδομένα μας σε περισσότερες από μία κλάσεις. Η τεχνική αυτή όμως λειτουργεί μόνο στην περίπτωση που τα δεδομένα διαχωρίζονται από γραμμικές συναρτήσεις. Τώρα θα δούμε την τεχνική Multi-Layer Perceptron(MLP) η οποία ουσιαστικά αποτελεί ένα Τεχνητό Νευρωνικό Δίκτυο και μπορεί να ταξινομεί δεδομένα που εμφανίζουν μη γραμμική συμπεριφορά.

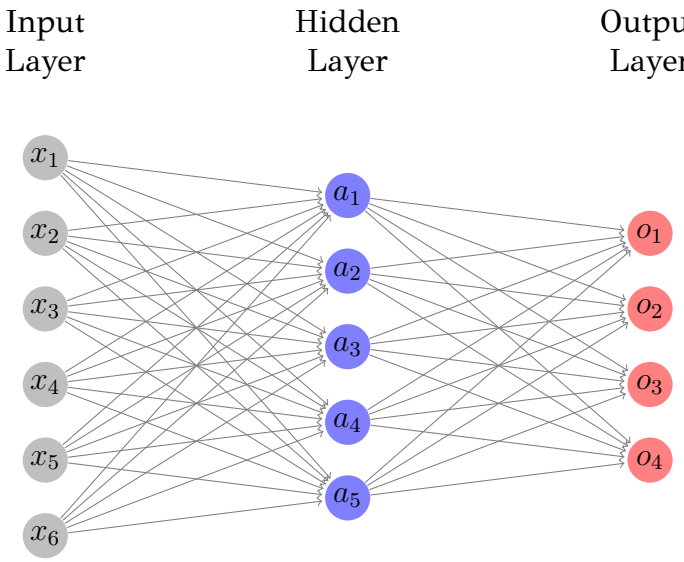
Ένα παράδειγμα αρχιτεκτονικής MLP με δύο Layers έχει ως εξής:

1ο Επίπεδο(Είσοδος): Το πρώτο επίπεδο δέχεται την είσοδο από τα δεδομένα τα οποία ονομάζουμε $x_j^{(i)}$

2ο Επίπεδο: Στο δεύτερο επίπεδο τα δεδομένα μας ακολουθούν αρχικά τον μετασχηματισμό:$z^{(i)} = x^{(i)} * w_h + b_h$ όπου το w_h είναι ένας πίνακας με βάρη διάστασης (n, d) όπου το n είναι ο αριθμός των

ανεξαρτήτων μεταβλητών των δεδομένων μας και το d ο αριθμός των νευρώνων όπου το επιλέγουμε εμείς. Στη συνέχεια κάθε στοιχείο του πίνακα z μετασχηματίζεται σύμφωνα με μια συνάρτηση την οποία ονομάζουμε συνάρτηση ενεργοποίησης $A(z^{(i)})^{(i)}$. Υπάρχουν διάφορες συναρτήσεις που χρησιμοποιούνται για αυτό το σκοπό αλλά πρέπει οπωσδήποτε να είναι κάποια μη γραμμική συνάρτηση. Κάθε νευρώνας έχει έξοδο $a_j^{(i)}$ όπου $j \in (1, d), i \in (1, m)$

3ο Επίπεδο(Έξοδος): Εδώ χρησιμοποιούμε ως είσοδο τα a_j του προηγούμενου επιπέδου και ακολουθούμε πάλι την ίδια διαδικασία με ποιο πάνω $z^{(i)} = a^{(i)} * w_o + b_o$ όπου το w_o εδώ έχει διάσταση (dxk) όπου k ο αριθμός των κλάσεων. Στη συνέχεια χρησιμοποιούμε πάλι συνάρτηση ενεργοποίησης στα z αλλά στην έξοδο θέλουμε να η συνάρτηση ενεργοποίησης να έχει φραγμένες τιμές. Έτσι χρησιμοποιούμε είτε τη softmax είτε τη σιγμοειδή η μπορούμε να χρησιμοποιήσουμε την tanh(z).



Σχήμα 3.7: Σχηματική αναπαράσταση MLP. Input Layer είναι είσοδος του MLP, στην Hidden Layer ακολουθεί τον μετασχηματισμό $a^{(i)} = A(x^{(i)} * w_h + b_h)$ και στην Output Layer των μετασχηματισμό $o^{(i)} = A(a^{(i)} * w_o + b_o)$

Όπως και στις προηγούμενες τεχνικές έτσι κι εδώ θα χρειαστούμε μια συνάρτηση κόστους. Συνήθως ανάλογα με την επιλογή την συνάρτησης ενεργοποίησης επιλέγουμε και την συνάρτηση κόστους. Για παράδειγμα αν επιλέξουμε τη σιγμοειδή συνάρτησή ως συνάρτηση ενεργοποίησης μια επιλογή συνάρτησης κόστους είναι η Mean Squarred Error-(MSE) αν επιλέξουμε την Softmax ως συνάρτηση ενεργοποίησης τότε μπορούμε να επιλέξουμε την Cross Entropy που είδαμε και στη Softmax Regression. Μόλις ορίσουμε και την συνάρτηση κόστους μετά μένει εκπαιδεύσουμε το δίκτυο ώστε να βρούμε τις κατάλληλες παραμέτρους w,b για το πρόβλημα μας. Ας δούμε πως ορίζεται πρόβλημα στην περίπτωση που έχουμε το πιο πάνω δίκτυο με συνάρτηση ενεργοποίησης την σιγμοειδή και συνάρτηση κόστους την MSE.
Έχουμε δύο πίνακες με βάρη για τα Hidden και Output επίπεδα τους οποίους θα συμβολίζουμε $W^{(1)}$ και$W^{(2)}$. Οπότε για παράδειγμα το πρώτο στοιχείο της δεύτερης γραμμής του δεύτερου πίνακα θα συμβολίζεται $w_{21}^{(2)}$.

Αυτή είναι η δομή του δικτύου. Εκτός όμως από την δομή πρέπει να δούμε και πώς θα το εκπαιδεύσουμε. Ο συνήθης και πιο απλός τρόπος είναι με τη μέθοδο Gradient Descent που είδαμε και πιο πάνω.
Ουσιαστικά αυτό που έχουμε να κάνουμε είναι να υπολογίσουμε όλες τις παραγώγους $\frac{\partial L}{\partial w_{ij}^{l}}$. Μπορεί να φαίνεται αρκετά περίπλοκο αλλά μπορούμε χρησιμοποιώντας τον κανόνα τις αλυσίδας στις παραγώγους και το γράφημα του Νευρωνικού Δικτύου να βρούμε τις παραγώγους που πρέπει να υπολογίσουμε.
'Ενας τρόπος για να σκεφτούμε πως να υπολογίσουμε τις παραγώγους είναι να πάμε στο γράφο του δικτύου και να βρούμε όλες τις πιθανές διαδρομές που καταλήγουν στο βάρος που θέλουμε. Για παράδειγμα έστω ότι θέλουμε να υπολογίσουμε την παράγωγο του βάρους w_{11}^{1} το οποίο στο γράφημα αναπαριστάται από το βέλος που πηγαίνει από το x_1 στο a_1. Αρχικά πρέπει να εντοπίσουμε όλες τις πιθανές διαδρομές από το L στο w και στη συνέχεια τις αντίστοιχες παραγώγους. Δηλαδή θα έχουμε

$$\frac{\partial L}{\partial w_{11}} = \sum_{i=1}^{k} (\frac{\partial L}{\partial o_i} \frac{\partial o_i}{\partial a_1} \frac{\partial a_1}{\partial w_{11}})$$

Η παραπάνω άθροιση για όλα το k μας δείχνει το ότι πρέπει να ακο-

λουθήσουμε όλες τις πιθανές διαδρομές από το L στο w_{11}

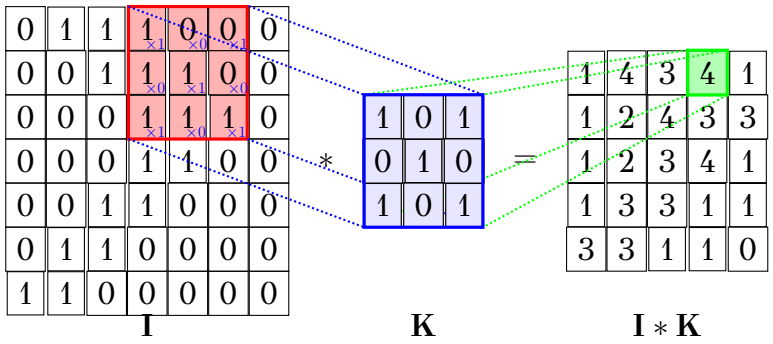
3.6 Convolutional Neural Networks (CNN's)

Σε όλα τα παραπάνω η είσοδος στους αλγορίθμους είναι διάνυσμα. Σε προβλήματα υπολογιστικής όρασης όπου τα δεδομένα αποτελούνται από εικόνες ή βίντεο χρησιμοποιούνται δίκτυα που ονομάζονται Convolutional Neural Networks η λειτουργία των οποίων εμπνεύστηκε από την λειτουργία των οπτικών νευρώνων και χρησιμοποιούν την μαθηματική πράξη της συνέλιξης.

Εκτός όμως από τα πιο γνωστά Convolutional Neural Networks που χρησιμοποιούνται στα προβλήματα υπολογιστικής όρασης, τα οποία ονομάζονται 2D CNN's έχουν αναπτυχθεί και δικτύα τα οποία λειτουργούν με παρόμοιο τρόπο και χρησιμοποιούνται σε προβλήματα που τα δεδομένα αποτελούνται από χρονόσειρες ή σε προβλήματα επεξεργασίας φυσικής γλώσσας τα οποία ονομάζονται 1D CNN's.

3.6.1 2D CNN

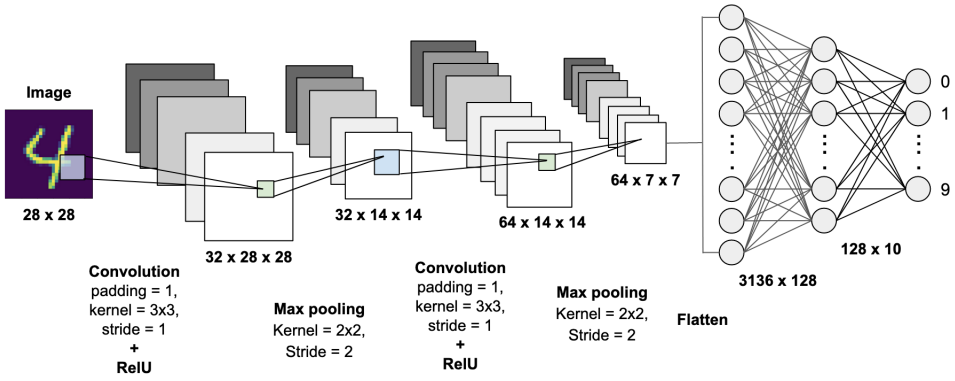
Όπως προαναφέρθηκε τα Convolutional Neural Networks εμπνεύστηκαν από την λειτουργία των οπτικών νευρώνων του εγκεφάλου. Κάποιοι από τους νευρώνες του οπτικού φλοιού έχουν την ικανότητα να εντοπίζουν τις κορυφές(edges) και σταδιακά πιο σύνθετα αντικείμενα σε μία εικόνα. Έχει βρεθεί πως τις λειτουργίες αυτές εκτελούν τα CNN's καταφέρνοντας έτσι να εντοπίζουν πρόσωπα ή αντικείμενα σε εικόνες.



Σχήμα 3.8: Παραδειγμα 2D συνέλιξης. Εδώ ο Kernel έχει σταθερά βάρη.

Στο σχήμα 3.8 βλέπουμε πως λειτουργεί η συνέλιξη. Ο πίνακας **I** είναι ή είσοδος (input) ο πίνακας **K** είναι ο πίνακας της συνέλιξης(Kernel) και ο πίνακας $\mathbf{I} * \mathbf{K}$ το αποτέλεσμα τις συνέλιξης.Η μέθοδος που ακολουθείται συνήθως είναι πως ό πίνακας kernel ''σαρώνει'' την είσοδο από αριστερά προς τα δεξιά και μόλις φτάσει στο τέλος ξεκινά την ίδια διαδικασία στην επόμενη σειρά.

Δύο παράμετροι που συναντούμε στα CNN's είναι τα Stride και Padding, Η παράμετρος Stride αναφέρεται στο βήμα ουσιαστικά του Kernel πάνω στην είσοδο. Padding είναι η διαδικασία όπου επεκτείνουμε τον αρχικό πίνακα ώστε το αποτέλεσμα τις συνέλιξης να διατηρήσει τις διαστάσεις του αρχικού πίνακα.



Σχήμα 3.9: Convolution Neural Netcwork για την αναγνώριση χειρόγραφων ψηφίων, πηγή εικόνας:towardsdatascience.com/

Στο σχήμα 3.9 βλέπουμε την αρχιτεκτονική ενός CNN το οποίο αναγνωρίζει χειρόγραφα ψηφία. Θεωρητικά στα πρώτα layers εντοπίζονται οι κορυφές και όσο πιο βαθιά προχωράμε πιο σύνθετα χαρακτηριστικά ώστε να διακριθεί ποιο είναι το ψηφίο που υπάρχει στην εικόνα. Μετά από τα Convolutional Layers ακολουθούν τα συμβατικά layers όπως και στο Multi-Layer Perceptron ώστε να παραχθεί το αποτέλεσμα τις ταξινόμησης.

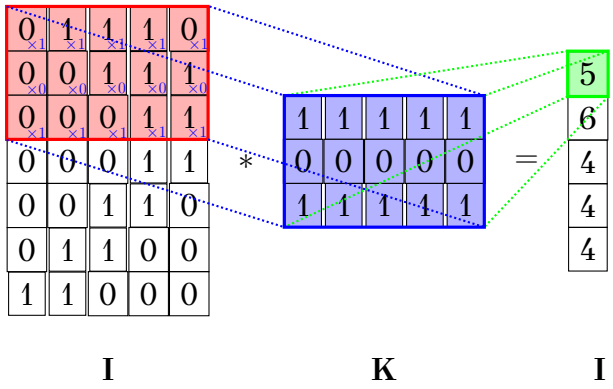
Έτσι βλέπουμε ότι η διαφορά των CNN τα νευρωνικά δίκτυα όπως το Multi-Layer Perceptron είναι η χρήση των Convolutional Layers τα οποία έχουν την ικανότητα να εντοπίζουν χαρακτηριστικά σε μία εικόνα.

3.6.2 1D Convolution

Η επιτυχία των 2D CNN's σε προβλήματα επεξεργασίας εικόνας και βίντεο ενέπνευσε την δημιουργία των 1D CNN's σε διαφορετικού τύπου προβλήματα. Δίκτυα τέτοιου τύπου που βρέθηκε ότι έχουν εφαρμογή στη επεξεργασία φυσικής γλώσσας ή σε δεδομένα που αποτελούνται από χρονοσειρές[11].

Οι διαφορές μεταξύ των δύο είναι ότι εδώ το πλάτος του kernel είναι ίσο με αυτό της εισόδου το οποίο συνήθως ισούται με τον αριθμό των

μεταβλητών κάθε μέτρησης. Επίσης στα 1D CNN'S η συνέλιξη λαμβάνει χώρα μόνο στην μία διάσταση, δηλαδή ο πίνακας kernel "σαρώνει" μόνο κατακόρυφα την είσοδο.



Σχήμα 3.10: Παράδειγμα της 1D συνέλιξης. Εδώ ο Kernel έχει σταθερά βάρη.

Στο σχήμα 3.10 βλέπουμε πως λειτουργεί η 1D συνέλιξη.Ο πίνακας **I** είναι ή είσοδος (input) ο πίνακας **K** είναι ο πίνακας της συνέλιξης(Kernel) και ο πίνακας $\mathbf{I} * \mathbf{K}$ το αποτέλεσμα τις συνέλιξης. Στα νευρωνικά δίκτυα οι τιμές του Kernel είναι βάρη που καθορίζονται κατά την διαδικασία της εκπαίδευσης.

Κεφάλαιο 4

Βασικές Τεχνικές στη Επεξεργασία φυσικής γλώσσας

Η επεξεργασία φυσικής γλώσσας εμφανίζει μία πρόσθετη δυσκολία σε σχέση με άλλους τομείς της τεχνητής νοημοσύνης η οποία πηγάζει από την φύση των δεδομένων προς επεξεργασία. Στις περιπτώσεις όπου τα δεδομένα προς επεξεργασία αποτελούν αρχεία ήχου, εικόνας ή βίντεο υπάρχει συσχέτιση μεταξύ μιας μετρούμενης φυσικής ποσότητας και τις τιμής που την αντιπροσωπεύει στο αρχείο προς επεξεργασία.Για παράδειγμα το χρώμα και η φωτεινότητα ενός αντικειμένου αντιπροσωπεύονται από τρεις τιμές σε κάθε pixel που καταλαμβάνει το αντικείμενο στην εικόνα, κάτι που μπορούμε να εκμεταλλευτούμε διότι μία απότομη αλλαγή στο χρώμα η στη φωτεινότητα μπορεί να σημαίνει ότι υπάρχει ένα διαφορετικό αντικείμενο στην εικόνα. Κάτι τέτοιο δεν είναι δυνατό να γίνει σε αρχεία κείμενου οπότε πρέπει να βρεθούν τεχνικές αναπαράστασης των λέξεων ώστε τυχόν συγγένεια σε νόημα δύο λέξεων να αντιπροσωπεύεται από ''συγγένεια'' των αναπαραστάσεων.

Τεχνικές για την αναπαράσταση των λέξεων είχαν ξεκινήσει να αναπτύσσονται πριν από την σύγχρονη εποχή της τεχνητής νοημοσύνης αλλά τα τελευταία χρόνια έχουν δημιουργηθεί πολλά καινούργια εργαλεία που έχουν παρουσιάσει ενδιαφέροντα αποτελέσματα.

Σε αυτό το κεφάλαιο θα αναφερθούμε σε κάποιες από τις πιο βασικές μεθόδους που χρησιμοποιούνται γι αυτό το σκοπό.

4.1 Δημιουργία Λεξιλογίου

Πρώτου αναφέρουμε πιο σύνθετες προσεγγίσεις αξίζει να αναφέρουμε την κατασκευή λεξιλογίου από τα δεδομένα προς επεξεργασία. Σε κάθε λέξη που εμφανίζεται στα δεδομένα μας αντιστοιχίζεται ένας αριθμός. Αν μια λέξη εμφανίζεται πάνω από μια φορές αντιστοιχεί στον ίδιο αριθμό.Έτσι ένα κείμενο μετατρέπεται σε μια ακολουθία αριθμών οι οποία μπορεί να επεξεργαστεί από τον υπολογιστή.

Σε αυτή την προσέγγιση δεν υπάρχει κάποια συσχέτιση μεταξύ του αριθμού που αντιστοιχείται σε κάθε λέξη και του νοήματος της λέξης αλλά είναι ένα χρήσιμο εργαλείο διότι μπορούμε με αυτή την προσέγγιση να μετατρέψουμε μια πρόταση σε μία ακολουθία αριθμών, δηλαδή διάνυσμα. που στη συνέχεια μπορούμε να το δώσουμε ως είσοδο σε ένα νευρωνικό δίκτυο.

Η διαδικασία που ακολουθείται για να μετατραπεί μία πρόταση σε διάνυσμα φαίνεται στο παρακάτω σχήμα:

$$This\ is\ a\ sentence \rightarrow \begin{bmatrix} This \\ is \\ a \\ sentence \end{bmatrix} \rightarrow \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix}$$

Κάθε πρόταση χωρίζεται στις επιμέρους λέξεις τις και στην συνέχεια αυτές αντικαθιστώνται από τον αριθμό που τους έχουμε αντιστοιχίσει.

4.2 Document Term Matrix

Μία τεχνική που χρησιμοποιείται πολύ συχνά όταν έχουμε δεδομένα που αποτελούνται από ένα σύνολο κειμένων είναι η Document Term Matrix. Σε αυτή την τεχνική δημιουργείται ένας πίνακας με διαστάσεις (M,N), όπου M είναι αριθμός των κειμένων που αποτελούν τα δεδομένα μας και N ο αριθμός των λέξεων που εμφανίζονται σε αυτό, όπως γίνεται και στην κατασκευή του λεξιλογίου. Ο πίνακας αυτός μετρά πόσες φορές εμφανίζεται η κάθε λέξη στο κάθε ένα από τα κείμενα.

Για παραδείγμα έχουμε τα παρακάτω κείμενα:

$$\boldsymbol{D1}: \textit{This is the first sentence}$$

$$\boldsymbol{D2}: \textit{This is the second sentence}$$

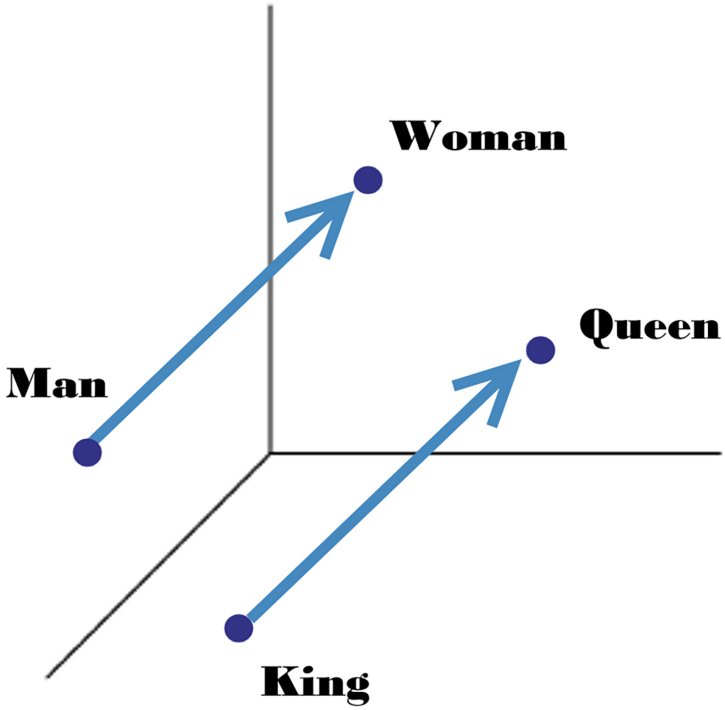
Έτσι ο πίνακας κατασκευάζεται ως εξής:

	This	is	the	first	second	sentence
D1	1	1	1	1	0	1
D2	1	1	1	0	1	1

4.3 Word2Vec

Η μέθοδος Word2Vec αναπτύχθηκε τον Mikolov[12] το 2013 στα πλαίσια ερευνών τις Google. Είναι μία μέθοδος αναπαράστασης των λέξεων ενός σετ δεδομένων σε διανύσματα τα οποία ιδανικά θα εμφανίζουν συσχετίσεις που φανερώνουν νοηματική σχέση των λέξεων που αντιπροσωπεύουν.

Ένα παράδειγμα των συσχετίσεων που περιμένουμε να εμφανιστούν φαίνεται στο Σχήμα 4.1 όπου η διαφορά των διανυσμάτων που αντιπροσωπεύουν τις λέξεις βασιλιάς-βασίλισσα θα αντιστοιχεί στη διαφορά των διανυσμάτων άνδρας-γυναίκα.



Σχήμα 4.1: Παράδειγμα της ιδανικής λειτουργίας της αναπαράστασης Word2Vec , πηγή εικόνας

Η μέθοδος αυτή βασίζεται στην Distributional hypothesis που συνοπτικά μας λέει πως λέξεις που σχετίζονται νοηματικά τείνουν να εμφανίζονται μαζί σε κείμενα.

Υπάρχουν δύο προσεγγίσεις για να κατασκευαστεί η αναπαράσταση Word2Vec. Η Common Bag Of Words και η Skip Gram. Και οι δύο χρησιμοποιούν ένα ρηχό νευρωνικό δίκτυο. Η διαφορά τους έγκειται στο ότι στην πρώτη χρησιμοποιείται ως είσοδος μια λέξη και η έξοδος πρέπει να είναι οι λέξεις που εμφανίζονται δίπλα σε αυτή τη λέξη και στη δεύτερη γίνεται το ανάποδο δηλαδή δίνονται ως είσοδος οι λέξεις που εμφανίζονται δίπλα από μία άλλη και πρέπει να προβλεφθεί η λέξη που λείπει.

4.4 Embedding Layer

Ένας ακόμη τρόπος να φτιάξουμε την αναπαράσταση των λέξεων σε διανύσματα είναι η μέθοδος Embedding Layer. Σε αυτή τη μέθοδο η αναπαράσταση εκπαιδεύεται ταυτόχρονα με το νευρωνικό δίκτυο στο οποίο θα χρησιμοποιηθεί.

Σε αυτή την μέθοδο αρχικά τα διανύσματα τις αναπαράστασεις έχουν τυχαίες τιμές οι οποίες αλλάζουν κατά την διάρκεια της εκπαίδευσης του δικτύου όπως και οι υπόλοιπες παράμετροι του δικτύοθ

Στο Σχήμα 4.2 βλεπουμε πως κάθε λέξη αρχικά αντικαθίσταται από τον αριθμό που έχει στο λεξιλόγιο και στην συνέχεια μετατρέπεται σε διάνυσμα το οποίο έχει αρχικά τυχαίες τιμές.

$$This\ is\ a\ sentence \rightarrow \begin{bmatrix} This \\ is \\ a \\ sentence \end{bmatrix} \rightarrow \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} \xrightarrow[Embedding]{} \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \end{bmatrix}$$

Σχήμα 4.2: Κάθε λέξη αντιστοιχεί σε μία γραμμή του πίνακα

Κεφάλαιο 5

Περιγραφή προβλήματος-τεχνικών

Στο κεφάλαιο αυτό θα εξετάσουμε τα δεδομένα που χρησιμοποιήθηκαν, τις αρχιτεκτονικές των δικτύων και τις τεχνικές που χρησιμοποιήθηκαν στην εκπαίδευση τους. Η μεθοδολογία σε ένα βαθμό βασίστηκε στην εργασία των Georgakopoulos et. al.[13]. Εκτός από τις προσεγγίσεις όπου χρησιμοποιήθηκαν νευρωνικά δίκτυα εφαρμόστηκαν και δύο παραλλαγές της μεθόδου Naive Bayes με σκοπό την σύγκριση της απόδοσης των νευρωνικών δικτύων με μεθόδους βασισμένες στην θεωρία των πιθανοτήτων και τις στατιστικής.

5.1 Δεδομένα

Τα δεδομένα που χρησιμοποιήθηκαν σε αυτή την εργασία δημοσιεύθηκαν στο site Kaggle στα πλαίσια ενός διαγωνισμού για την κατασκευή μοντέλων ταξινόμησης. Τον διαγωνισμό ξεκίνησαν οι εταιρίες Google και Alphabet με σκοπό να βρεθεί ένας τρόπος όχι μόνο να κατηγοριοποιούνται σχόλια σε υβριστικά ή όχι αλλά και τι τύπου προσβολή περιέχουν.

Τα δεδομένα προέρχονται από σχόλια στην Wikepedia και έχουν ταξινομηθεί σε κατηγορίες από ανθρώπους.

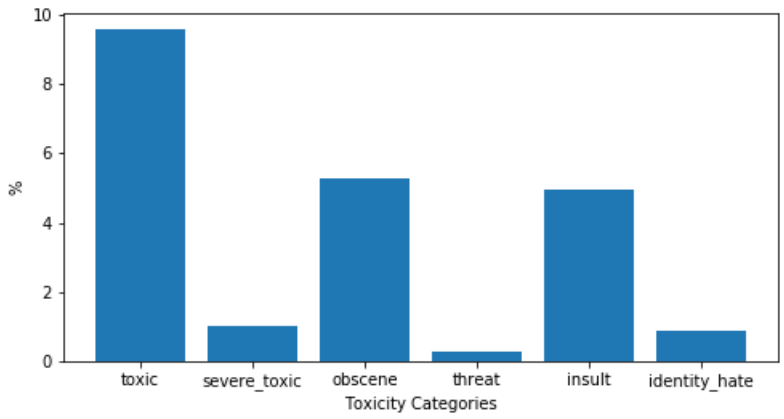
Έτσι τα δεδομένα μας αποτελούν ένα πίνακα όπου ως στήλες έχει ένα id για κάθε σχόλιο, τον κείμενο του σχολίου και στην συνεχεία ακολουθούν οι κατηγορίες τοξικότητας. Με 0 σημειώνεται αν ένα σχόλιο δεν εντάσσεται σε αυτή την κατηγορία και με 1 αν εντάσσεται.

Ένα δείγμα των δεδομένων φαίνεται στο Σχήμα 5.1.

	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0
5	00025465d4725e87	"\n\nCongratulations from me as well, use the ...	0	0	0	0	0	0
6	0002bcb3da6cb337	COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK	1	1	1	0	1	0
7	00031b1e95af7921	Your vandalism to the Matt Shirvington article...	0	0	0	0	0	0
8	00037261f536c51d	Sorry if the word 'nonsense' was offensive to ...	0	0	0	0	0	0
9	00040093b2687caa	alignment on this subject and which are contra...	0	0	0	0	0	0

Σχήμα 5.1: Δείγμα των δεδομένων προς επεξεργασία

Παρακάτω βλέπουμε την σχετική συχνότητα των κατηγοριών στα δεδομένα



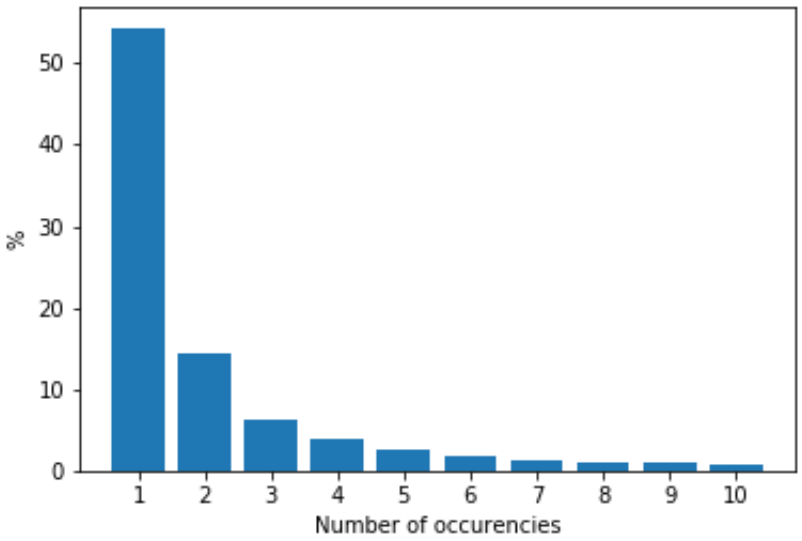
Σχήμα 5.2: Συχνότητες των κατηγοριών τοξικότητας.

Στην συγκεκριμένη εργασία δεν εξετάσαμε τις ξεχωριστές κατηγόριες αλλά αν κάθε σχόλιο ανήκει έστω σε μία από αυτές. Στην περίπτωση που ανήκε έστω σε μία το σχόλιο χαρακτηριζόταν τοξικό.

Σχήμα 5.3: Συχνότητα συνολικής τοξικότητας στα δεδομένα. Βλέπουμε ότι το 90% είναι μη τοξικά.

Στο Σχήμα 5.3 βλέπουμε ότι περίπου το 90% των σχολίων είναι μη τοξικά κάτι που πρέπει να λάβουμε υπόψη μας όταν επιλέξουμε μετρική για να μετρήσουμε την επιτυχία του ταξινόμητη. Στο συγκεκριμένο σετ δεδομένων μία πρόβλεψη πως όλα τα σχόλια είναι μη τοξικά θα έχει ακρίβεια σχεδόν 90%. Άρα πέραν της ακρίβειας πρέπει να βρούμε κι άλλον τρόπο να μετρήσουμε την επιτυχία του ταξινομητή μας.

Άλλη μία ιδιαιτερότητα των δεδομένων είναι η εμφάνιση λέξεων για μια μόνο φορά. Συγκεκριμένα πάνω από το 50% των λέξεων στα δεδομένα μας εμφανίζονται μόνο μία φορά. Οι λέξεις αυτές μπορεί να είναι ειδικοί χαρακτήρες όπως emojis ή ορθογραφικά λάθη ή παραφράσεις λέξεων με σκοπό να μην εντοπίζονται αυτόματα. Τέτοιες παραφράσεις είναι για παράδειγμα η αντικατάσταση του α με το @.



Σχήμα 5.4: Ποσοστό λέξεων που εμφανίζονται έως και δέκα φορές

Στο διάγραμμα παραπάνω βλέπουμε το ποσοστό των λέξεων που εμφανίζονται εώς και 10 φορές στα δεδομένα. Αθροιστικά αποτελούν το 87.76% των λέξεων.

Θα μπορούσαμε να επιλέξουμε να αγνοήσουμε τις λέξεις που εμφανίζονται μόνο μία φορά αλλά σε πολλές περιπτώσεις στο διαδίκτυο όπου εμφανίζονται προσβλητικά σχόλια οι χρήστες επιλέγουν να μην γράψουν κανονικά την λέξη αλλά να την παραφράσουν. Έτσι λήφθηκαν υπ'οψιν όλες οι λέξεις.

5.2 Naive Bayes

Για να έχουμε ένα μέτρο σύγκρισης της απόδοσης των νευρωνικών δικτύων εφαρμόστηκε η μέθοδος Naive Bayes[14] για την ταξινόμηση των σχολίων. Συγκεκριμένα εφαρμόστηκαν δυο παραλλαγές της μεθόδου, μια όπου γίνεται χρήση της πολυωνυμικής κατανομής και μια όπου γίνεται χρήση της κατανομής Bernoulli.

Η μέθοδος Naive Bayes αποτελεί μια μέθοδο ταξινόμησης βασισμένη στο θεώρημα του Bayes. Το θεώρημα αυτό χρησιμοποιείται στην θεωρία των πιθανοτήτων για τον υπολογισμό της δεσμευμένης πιθανότητας δηλαδή της πιθανότητας ενός ενδεχομένου υπο την προϋπόθεση ότι έχει πραγματοποιηθεί ένα άλλο ενδεχόμενο.
Η δεσμευμένη πιθανότητα κατά Bayes υπολογίζεται από τον τύπο:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

$P(A \mid B)$: Η πιθανότητα να πραγματοποιείται το ενδεχόμενο Β δεδομένου ότι έχει πραγματοποιηθεί το ενδεχόμενο Α.
$P(B \mid A)$:Η πιθανότητα να πραγματοποιείται το ενδεχόμενο Α δεδομένου ότι έχει πραγματοποιηθεί το ενδεχόμενο Β.
$P(A), P(B)$: Η ανεξάρτητες πιθανότητες να πραγματοποιηθούν τα ενδεχόμενα Α και Β.

Μπορούμε να χρησιμοποιήσουμε το θεώρημα αυτό για να φτιάξουμε ένα δυαδικό ταξινομητή ως εξής: Έστω ότι τα δεδομένα αποτελούνται από διανύσματα $x^i = (x_1, x_2....x_n)$ καθένα από τα οποία ανήκει σε μία κλάση $y^i \in \{0, 1\}$.
Η πιθανότητα της κλάσης του x^i δίνεται από την σχέση:

$$P(y^i \mid x^i) = \frac{P(x^i \mid y^i)P(y^i)}{P(x^i)} = \frac{P((x_1, x_2....x_n)^i \mid y^i)P(y^i)}{P(x^i)}$$

Θεωρώντας πως οι συνιστώσες του x^i είναι ανεξάρτητες μεταξύ τους η παραπάνω σχέση μπορεί να γραφεί ως:

$$P(y^i \mid x^i) = \frac{\prod_{j=1}^{N} P(x_j^i \mid y^i)P(y^i)}{P(x^i)}$$

Άπό την παραπάνω σχέση τα $P(y^i)$ και $P(x^i)$ είναι γνωστά από τα δεδομένα. Την κατανομή τον όρο. $P(x^i \mid y^i)$ την επιλέγουμε με βάση το εκάστοτε πρόβλημα. Στην συγκεκριμένη εργασία χρησιμοποιήθηκαν οι πολυωνυμική και η κατανομή Bernoulli.

Στην περίπτωση της πολυωνυμικής κατανομής επιλέγουμε την κατανομή

$$P(x^i \mid y^i) = \frac{(\sum_{j=1}^{n} x_j^i)!}{\prod_{j=1}^{n} x_j^i!} \prod_{j=1}^{n} p_j^{x_j^i}$$

Όπου p_j η πιθανότητα να εμφανίζεται η λέξη με δείκτη j στα κείμενα μας.

Στην περίπτωση της πολυωνυμικής κατανομής για την ταξινόμηση κείμενων κατασκευάζεται ο πίνακας Document Term Matrix ο οποίος χρησιμοποιείται ως είσοδος στον αλγόριθμο. Η μεταβλητή x_j^i εδώ συμβολίζει το πόσες φορές εμφανίστηκε η λέξη με δείκτη j από στο i στοιχείο των δεδομένων μας.

Στην περίπτωση της κατανομής Bernoulli η κατανομή που επιλέγεται είναι η

$$P(x^i \mid y^i) = \prod_{j=1}^{n} p_j^{x_j^i}(1 - p_j^{(1-x_j)})$$

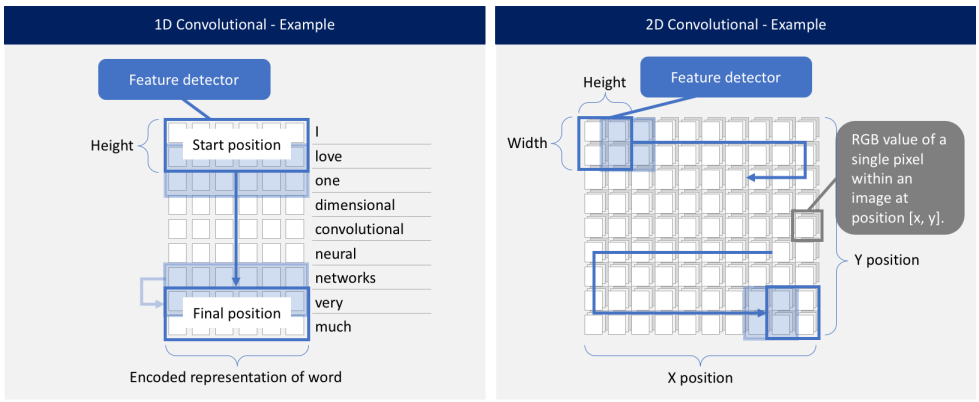
Στην περίπτωση της κατανομής Bernoulli για την ταξινόμηση κείμενων ως είσοδος χρησιμοποιείται μια τροποποιημένη εκδοχή του Document Term Matrix όπου δεν μετράμε των αριθμό εμφανίσεων κάθε λέξης σε κάθε έγγραφο αλλά αν αυτό εμφανίστηκε ή όχι κάνοντας την μεταβλητή x_j^i να παίρνει τις τιμές (0,1).

5.3 Αρχιτεκτονική δικτύων - Μέθοδος εκπαίδευσης

Για την επίλυση του προβλήματος χρησιμοποιήθηκαν δύο τεχνικές αναπαραστάσεων των λέξεων σε διανύσματα, μια αναπαράσταση Word-2Vec και μια με χρήση Embedding Layer. Τέλος επιλέχθηκε ως Baseline Model η επιλογή όλα τα σχόλια να είναι μη τοξικά για να εκτιμήσουμε την ικανότητα των αλγορίθμων να διακρίνουν τα τοξικά σχόλια.

5.3.1 Αρχιτεκτονική δικτύων

Αν και σε προβλήματα επεξεργασίας κειμένου χρησιμοποιείται συνήθως η μέθοδος 1D-Convolution στη παρούσα εργασία χρησιμοποιήθηκε και η 2D-Convolution που συνήθως χρησιμοποιείται σε προβλήματα ταξινόμησης εικόνας.



Σχήμα 5.5: Σύγκριση 1D και 2D Convolution, πηγή εικόνας

Βλέπουμε στο σχήμα 5.5 τις διαφορές μεταξύ της μεθόδου 1D &2D Convolution. Στο πρόβλημα που εξετάζουμε στην παρούσα εργασία η διαφορά αυτή σημαίνει πως στην πρώτη περίπτωση σε κάθε βήμα της συνέλιξης συμμετέχουν όλες οι συνιστώσες ενώ στην δεύτερη περίπτωση σαρώνονται τμηματικά.

Και στις δύο αναπαραστάσεις για να υπάρχει ένα μέτρο σύγκρισης επιλέχθηκαν ίσης διάστασης διανύσματα τα οποία ήταν διάστασης (1,20). Το μήκος των σχολίων περιορίστηκε στις 120 λέξεις. Για τα σχόλια με μικρότερο μήκος χρησιμοποιήθηκε τεχνική Zero Padding ώστε τα δεδομένα μας να έχουν όλα τις ίδιες διαστάσεις. Έπειτα από αυτή της επεξεργασία κάθε σχόλιο μετατράπηκε σε ένα πίνακα διαστάσεων (120,20).

Στους πίνακες 5.1 έως 5.4 παρουσιάζονται οι αρχιτεκτονικές των δικτύων. Όλα τα δίκτυα έχουν το ίδιο αριθμό Convolutional Layer ώστε να συγκρίνουμε την αποτελεσματικότητα των διαφορετικών προσεγγίσεων

και όχι του βάθους των δικτύων

Η αρχιτεκτονικές των δικτύων είναι οι εξής:

Layer	Shape
Input	(120,20)
Conv1D	10
Max Pooling1D	3
Conv1D	10
Max Pooling	3
Flatten	-
Dropout	0.2
Dense	100
Dense	32
Dense	1

Πίνακας 5.1: 1D Convolutional Network, Word2Vec Embedding

Layer	Shape
Input	(120,1)
Embedding Layer	(120,20)
Conv1D	10
Max Pooling1D	3
Conv1D	10
Max Pooling	3
Flatten	-
Dropout	0.2
Dense	100
Dense	32
Dense	1

Πίνακας 5.2: 1D Convolutinal Network, Embedding Layer

Layer	Shape
Input Layer	(120,20,1)
Conv2D	(5,10,5)
Max Pooling2D	(3,3)
Conv2D	(3,3,3)
Max Pooling2D	(2,2)
Dropout	0.2
Dense	100
Dense	32
Dense	1

Πίνακας 5.3: 2D Convolutional Network, Word2Vec Embedding

Layer	Shape
Input Layer	(120,1)
Embedding Layer	(120,20)
Reshape	(120,20,1)
Conv2D	(5,10,5)
Max Pooling2D	(3,3)
Conv2D	(3,3,3)
Max Pooling2D	(2,2)
Dropout	0.2
Dense	100
Dense	32
Dense	1

Πίνακας 5.4: 2D Convolutional Network, Embedding Layer

5.3.2 Μέθοδος Εκπαίδευσης ADAM

Ο αλγόριθμος ADAM είναι είναι ένας αλγόριθμος εκπαίδευσης που χρησιμοποιείται σε δίκτυα Deep Learning. Όπως όλοι οι αλγόριθμοι εκπαίδευσης χρησιμοποιεί τις παραγώγους της συνάρτησης κόστους για να βρει τις βέλτιστες τιμές των βαρών αλλά έχει δύο σημαντικές διαφορές σε σχέση με την μέθοδο Gradient Descent.

Πρώτη διαφορά είναι ότι δεν υπολογίζει τις παραγώγους για το σύνολο των δεδομένων αλλά για τυχαία επιλεγμένα υποσύνολο. Για το λόγο αυτό ονομάζεται στοχαστικός. Με την χρήση αυτής της τεχνικής επιταχύνεται κατα πολύ η διαδικασία εκπαίδευσης για είναι πολύ λιγότερες οι απαιτούμενες πράξεις που πρέπει να γίνουν.

Η δεύτερη διαφορά είναι πως δεν χρησιμοποιεί σταθερά ως learning rate αντ' αυτού η τιμή της μεταβάλλεται ξεχωριστά για τις παραμέτρους του δικτύου. Έτσι το δίκτυο εκπαιδεύεται γρηγορότερα σε σχέση με μια συμβατική μέθοδο όπως η Gradient Descent.

5.3.3 Συνάρτηση Κόστους και μετρικές

Η συναρτήση κόστους που επιλέχθηκε είναι η Cross Entropy. Η συνάρτηση αυτή χρησιμοποιείται σε προβλήματα ταξινόμησης και για προβλήματα με μόνο δύο κλάσσεις υπολογίζεται από τον τύπο:

$$L = -\frac{1}{m}\sum_{i=1}^{m}(y_i log(p_i) + (1 - y_i) log(1 - p_i))$$

Όπου i είναι το στοιχείο των δεδομένων μας p_i είναι η πρόβλεψη του δικτύου για το στοιχείο αυτό, m το πλήθος των δεδομένων και y_i κλάση του συγκεκριμένου στοιχείου.
Ιδανικά αν το μοντέλο μας έχει ακρίβεια 100% η τιμή της cross entropy θα έπρεπε να είναι 0.

Για τη μέτρηση της απόδοσης των δικτύων επιλέχθηκαν δύο μετρικές. Η **ακρίβεια (Accuracy)** και η **Area Under Curve(AUC)**. Η ακρίβεια μετρά το ποσοστό των δεδομένων για το οποίο έγινα σωστές προβλέψεις. Η μετρική Area Under Curve λαμβάνει υπ' όψιν την και τα ψευδή θετικά και αληθή θετικά αποτελέσματα ώστε να δούμε κατά πόσο είναι ό ταξινομητής είναι ικανός να διακρίνει τις κλάσεις. Η τιμή της μετρικής αυτής κυμαίνεται από 0.5 έως 1 όπου 0.5 έχει ένας ταξινομητής που δεν καταφέρνει να διαχωρίσει της κλάσεις και 1 ένας ταξινομητής που διαχωρίζει τέλεια χωρίς να έχει ψευδή θετικά αποτελέσματα. Για παράδειγμα εάν σε ένα σετ δεδομένων το 95% ανήκει στη μία κλάση αν υποθέταμε πως όλα τα δεδομένα ανήκουν σε αυτή την κλάση θα είχαμε

ακρίβεια 95% αλλά η τιμή της AUC θα ήταν 0.5. Έτσι θα γνωρίζαμε η ταξινόμηση δεν λειτουργεί σωσταά.

Κεφάλαιο 6

Αποτελέσματα

Το κεφάλαιο αυτό είναι χωρισμένο σε τρείς ενότητες. Αρχικά εξετάζουμε την απόδοση των ταξινομητών που περιγράφηκαν στο προηγούμενο κεφάλαιο στο Dataset που προέρχεται από το Kaggle. Στη συνέχεια εξετάζονται οι δύο τεχνικές αναπαράστασης που χρησιμοποιήθηκαν. Τέλος στην τελευταία ενότητα εξετάζονται δύο εφαρμογές των ταξινομητών. Στην πρώτη εφαρμογή εξετάστηκε αν υπάρχει κάποια συσχέτιση μεταξύ των τοξικών απαντήσεων σε Tweets του κόμματων των συντηρητικών του Η/Β με το περιεχόμενο των Tweets και στην δεύτερη εφαρμογή αν υπάρχει κάποια συσχέτιση της τοξικότητας των Tweets με την δημοτικότητα του κόμματος, όπως αυτή αποτυπώνεται στις δημοσκοπήσεις.

6.1 Αποτελέσματα Ταξινόμησης Σχολίων

Όπως αναφέρθηκε και στο Κεφάλαιο 5 εξετάστηκαν τέσσερα μοντέλα νευρωνικών δικτύων και δύο μοντέλα όπου έγινε χρήση της μεθόδου Naive Bayes. Τα νευρωνικά δίκτυα εκπαιδεύτηκαν για 10 εποχές με τη μέθοδο ADAM. Τα δεδομένα χωρίστηκαν σε δύο με μέρη το ένα εκ των οποίων χρησιμοποιήθηκε για την εκπαίδευση των δικτύων και το άλλο για τον έλεγχο της απόδοσης τους.

Η συνάρτηση κόστους που χρησιμοποιήθηκε είναι η Cross Entropy και οι μετρικές που χρησιμοποιήθηκαν είναι η Accuracy(Ακρίβεια) και Area Under Curve(AUC).

Στην συνέχεια παρατίθονται πίνακες με τα αποτελέσματα των ταξινομητών.

Metrics	Cross Entropy	Accuracy	AUC
Baseline	-	0.90	0.5
Naive Bayes(Bern)	-	0.87	0.83
Naive Bayes(Multi)	-	0.95	0.84
Word2Vec 1D	0.24	0.93	0.89
Word2Vec 2D	0.24	0.92	0.88
Embedding 1D	0.24	0.98	0.99
Embedding 2D	0.24	0.96	0.95

Πίνακας 6.1: Αποτελέσματα ταξινόμησης για το Training set.

Στο πίνακα 6.1 βλέπουμε πως η προσέγγιση Naive Bayes με χρήση της κατανομής Bernoulli πέτυχε χαμηλότερη ακρίβεια από το Baseline. Βλέπουμε όμως ότι στην μετρική Area Under Curve η προσέγγιση αυτή έχει συγκρίσιμη τιμή με τις υπόλοιπες προσεγγίσεις. Επίσης η προσέγγιση Naive Bayes με την πολυωνυμική κατανομή πέτυχε αποτελέσματα συγκρίσιμα με αυτά των νευρωνικών δικτύων.

Συγκρίνοντας τις προσεγγίσεις των νευρωνικών δικτύων βλέπουμε υπεροχή των δικτύων όπου έγινε χρήση της Embedding Layer σε σχέση με την προεκπαιδευμένη αναπαράσταση Word2Vec. Μεταξύ των δύο προσεγγίσεων των νευρωνικών που χρησιμοποιούν την Embedding Layer καλύτερη απόδοση είχε η μέθοδος 1D Convolution.

Metrics	Cross Entropy	Accuracy	AUC
Baseline	-	0.90	0.5
Naive Bayes(Bern)	-	0.87	0.81
Naive Bayes(Multi)	-	0.95	0.81
Word2Vec 1D	0.20	0.93	0.91
Word2Vec 2D	0.24	0.92	0.89
Embedding 1D	0.24	0.95	0.92
Embedding 2D	0.23	0.94	0.93

Πίνακας 6.2: Αποτελέσματα, Test set

Στον πίνακα 6.2 βλέπουμε τα αποτελέσματα των ταξινομητών για το Test set. Παρατηρούμε ότι και οι προσεγγίσεις Naive Bayes εμφάνισαν σχετικά σταθερή απόδοση στο Training set και Test set. Πάλι όπως και πριν τα νευρωνικά δίκτυα εμφάνισαν καλύτερη απόδοση σε σχέση με τις προσεγγίσεις Naive Bayes ιδίως ως προς την μετρική Area Under Curve κάτι που σημαίνει ότι δίνουν λιγότερα ψευδή θετικά και ψευδή αρνητικά αποτελέσματα.

Όσον αφορά τα νευρωνικά δίκτυα βλέπουμε πως εμφανίζεται η ίδια σειρά στις αποδόσεις με training set. Και αυτή την περίπτωση τα μοντέλα με Embedding Layer απέδωσαν καλύτερα από τις αναπαραστάσεις Word2Vec και τα η μέθοδος 1D Convolution καλύτερα από την 2D Convolution. Βέβαια φαίνεται μεγαλύτερο ρόλο παίζει ο τύπος της αναπαράστασης παρά ο τύπος της συνέλιξης στην απόδοση, από την στιγμή που η 2D συνέλιξη με χρήση embedding layer ξεπέρασε την 1D συνέλιξη με αναπαράσταση Word2Vec.

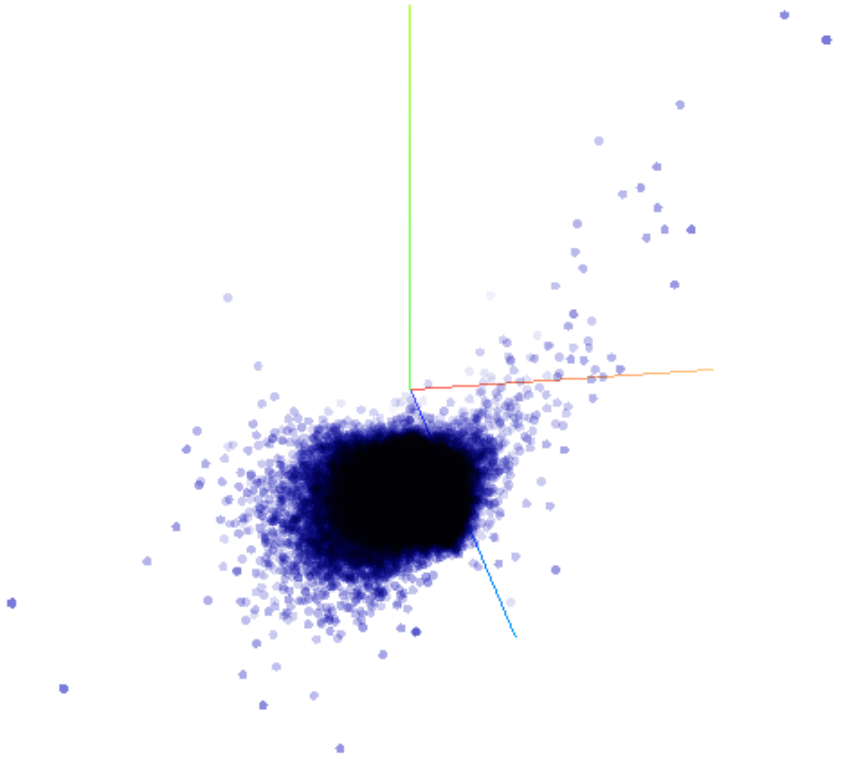
Από τα παραπάνω μοντέλα επιλέχθηκε το 1D μοντέλο με την Embedding Layer για την ταξινόμηση σχολίων που συλλέχθηκαν από το Twitter τα οποία θα δούμε στην ενότητα 6.3.

6.2 Σύγκριση Word2Vec και Embedding Layer

Είδαμε ότι χρησιμοποιήθηκαν δύο τεχνικές για την αναπαράσταση των λέξεων σε διανύσματα. Στη μία κατασκευάστηκε η αναπαράσταση με την μέθοδο Word2Vec, πριν εκπαιδευτούν, τα δίκτυα και στην δεύτερη έγινε χρήση Embedding Layer και η αναπαράσταση εκπαιδεύτηκε ταυτόχρονα με τα νευρωνικά δίκτυα. Στην ενότητα αυτή θα εξετάσουμε και θα συγκρίνουμε τις δύο μεθόδους στο κατά πόσο πέτυχαν να ομαδοποιήσουν τις λέξεις με βάση το νόημα τους.

Ένας τρόπος να ελέγξουμε αν κατάφεραν να ομαδοποιήσουν οι αναπαραστάσεις τις λέξεις είναι να δούμε τις θέσεις τους στο διανυσματικό χώρο. Αν και αυτό είναι αδύνατο για ένα χώρο είκοσι διαστάσεων, όσες είναι οι διαστάσεις των διανυσμάτων των αναπαραστάσεων που επι-

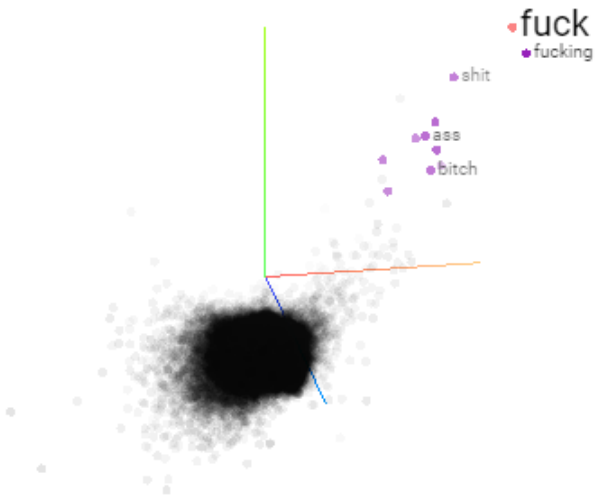
λέχθηκαν, μπορούμε να εφαρμόσουμε την μέθοδο Principal Component Analysis ώστε να προβάλουμε αυτόν το χώρο στις τρείς διαστάσεις.



Σχήμα 6.1: Προβολή σε τρισδιάστατο χώρο της αναπαράστασης που δημιουργήθηκε μέσω τις Embedding Layer. Κάθε σημείο αντιστοιχεί σε ένα διάνυσμα με αρχή την αρχή των αξόνων.

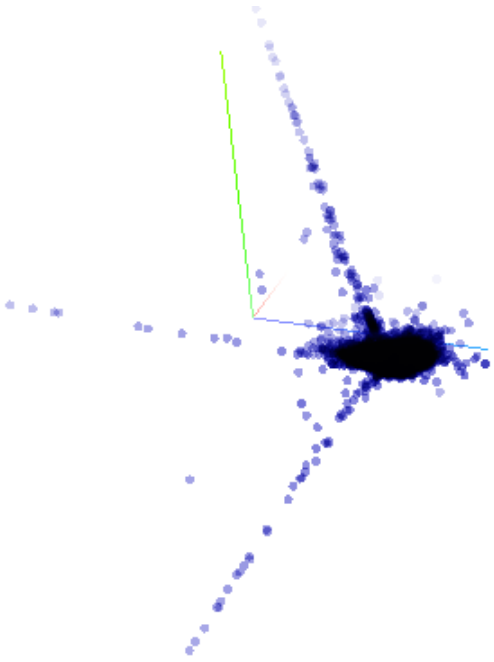
Στο σχήμα 6.1 βλέπουμε την αναπαράσταση που δημιουργήθηκε από την Embedding Layer του νευρωνικού δικτύου με 1D συνέλιξη. Είναι φανερό πως η πλειοψηφία των λέξεων έχει συγκεντρωθεί στην περιοχή που απεικονίζεται ως μαύρη στο σχήμα. Αυτό εξηγείται αν λάβουμε υπ' όψιν μας πως οι αναπαραστάσεις λέξεων σε διανύσματα βασίζονται στις γειτονικές λέξεις για την αντιστοίχηση τους σε διανύσματα. Έτσι τα διανύσματα που αντιστοιχούν σε λέξεις με μια ή δυο εμφανίσεις έχουν τιμές κοντά στις αρχικές τυχαίες τιμές που τους ανατέθηκαν κι όχι τιμές που να βασίζονται στο περιεχόμενο τους. Η μόνη περιοχή που φαίνεται να ξεχωρίζει είναι ο βραχίονας που εμφανίζεται να εκτείνεται

με κατεύθυνσή πάνω και δεξιά. Την περιοχή αυτή θα εξετάσουμε στη συνέχεια.



Σχήμα 6.2: Οι λέξεις του σχήματος 6.1 με επιλογή στον βραχίονα που εμφανιζόταν πάνω δεξία. Βλέπουμε πως η περιοχή αυτή καταλαμβάνεται από λέξεις που εμφανίζοντια πολύ συχνά σε κείμενα με υβριστικό περιεχόμενο.

Στο σχήμα 6.2 βλέπουμε τον βραχίονα που αναφέραμε προηγουμένως. Κοιτώντας τις λέξεις που εμφανίζονται σε αυτόν βλέπουμε πως είναι λέξεις με υβριστικό περιεχόμενο. Αν και η αναπαράσταση δεν ομαδοποίησε λέξεις γενικού περιεχομένου βλέπουμε πως ξεχώρισε από το σύνολο λέξεις με υβριστικό περιοχόμενο, κάτι που ταυτίζεται με σκοπό του δικτύου για το οποίο εκπαιδεύτηκε.



Σχήμα 6.3: Προβολή της αναπαράστασης Word2Vec.

Στο σχήμα 6.3 βλέπουμε την προβολή της Word2Vec αναπαράστασης. Αν και στο σχήμα διακρίνονται τρεις διακριτοί βραχίονες σε κανέναν από αυτούς δεν εμφανίζονται λέξεις με σύνδεση στο περιεχόμενο. Επίσης όπως και στην προηγούμενη αναπαράσταση η πλειοψηφία των λέξεων είναι συγκεντρωμένη στη μαύρη περιοχή που εμφανίζεται στο σχήμα. Ο λόγος που συμβαίνει αυτό όπως και πριν είναι η εμφάνιση πολλών λεξέων πολύ λίγες φορές μέσα στο δεδομένα που χρησιμοποιήθηκαν.

Συγκρίνοντας τις δύο μεθόδους βλέπουμε η αναπαράσταση που δημιουργήθηκε από την Embedding Layer "κληρονόμησε" την ιδιότητα του δικτύου για το οποίο δημιουργήθηκε και πέτυχε να ομαδοποιήσει λέξεις που εμφανίζονται σε υβριστικά σχόλια. Αυτό προφανώς οφείλεται στο ότι η αναπαράσταση εκπαιδεύτηκε μαζί με το δίκτυο που είχε ως σκοπό την διάκριση των σχολίων σε υβριστικά η όχι.

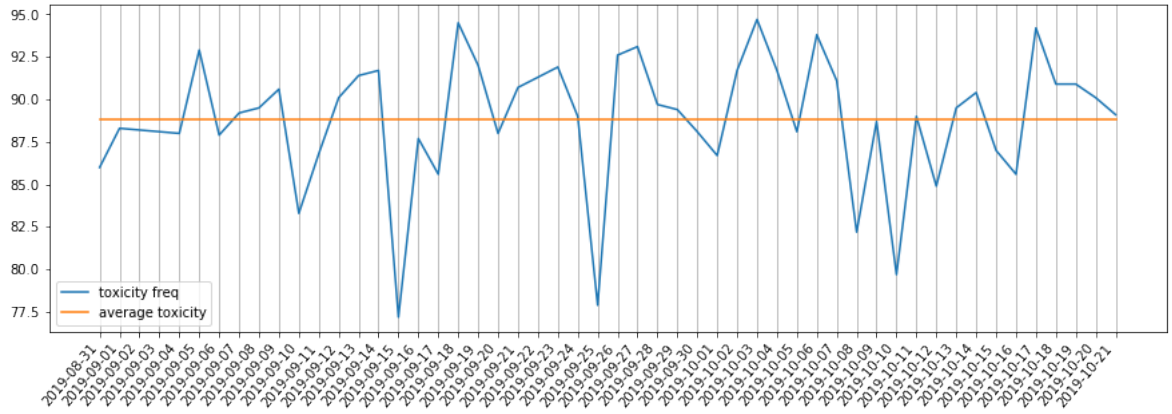
Πέραν από αυτή την διαφορά καμία από τις δύο αναπαραστάσεις δεν κατάφερε να ξεχωρίσει τις λέξεις με κάποιο άλλο τρόπο.

6.3 Εφαρμογή Εκπαιδευμένου δικτύου σε σχόλια του Twitter

Σε αυτή την ενότητα θα δούμε δύο εφαρμογές των ταξινομητών σε δεδομένα που συλλεχθηκαν από το Twitter. Σκοπός είναι να διερευνήσουμε πιθανές δυνατότητες που μπορεί να προσφέρουν τέτοιου τύπου ταξινομητές σε προβλήματα του πραγματικού κόσμου.

Τα δεδομένα που συγκεντρώθηκαν αποτελούν απαντήσεις σε δημοσιεύσεις στο Twitter του κόμματος των συντηρητικών του Ηνωμένου Βασιλείου για την χρονική περίοδο από 31/8 έως και 21/10. Μετά την συγκέντρωση των δεδομένων εξετάστηκαν δύο υποθέσεις. Πρώτον εάν υπάρχει κάποια συσχέτιση του περιεχομένου των δημοσιεύσεων με την τοξικότητα των απαντήσεων σε αυτές και δεύτερον αν υπάρχει κάποια συσχέτιση της τοξικότητας των απαντήσεων με τα αποτελέσματα των δημοσκοπήσεων.

Η διαδικασία που ακολουθήθηκε μετά την συγκέντρωση των δεδομένων ήταν η εξής. Για κάθε μέρα στην χρονική περίοδο που εξετάστηκε συγκεντρωνόταν 1000 σχόλια που ήταν το ανώτερο όριο που επιτρέπει η API του Twitter. Τα σχόλια αυτά περνούσαν από τον ταξινομητή όπου τα κατέτασσε σε κατηγορίες (τοξικά-μη τοξικά). Στην συνέχεια για κάθε μέρα υπολογίσαμε την συχνότητα των μη τοξικών σχολίων ($\frac{N_{non-toxic}}{N_{total}}$). Η τιμή αυτή στα διαγράμματα αναφέρεται ως Toxicity Frequency.



Σχήμα 6.4: Η πορεία της τοξικότητας την περίοδο 31/8 έως 21/10 και ο μέσος όρος ως σημείο αναφοράς.

Στο σχήμα 6.4, βλέπουμε πως μεταβάλλεται της Toxicity Frequency των απαντήσεων στις αναρτήσεις. Στο γράφημα παρουσιάζεται και ο μέσος όρος ως σημείο αναφοράς. Βλέπουμε πως υπάρχουν μέρες με μεγάλη μείωση ή αυξηση της τοξικότητας. Αυτές οι ημερομηνίες θα αποτελέσουν τα σημεία ενδιαφέροντως όπου και θα εστιάσουμε στις επόμενες υποενότητες.

6.3.1 Μεταβολή τοξικότητας και Tweets

Στην υποενότητα αυτή θα εστιάσουμε σε ημερομηνίες όπου φαίνονται στο σχήμα 6.4 κατά τις οποιές εμφανίστηκαν εξαιρετικά πολλά η εξαιρετικά λίγα τοξικά σχόλια και θα εξετάσουμε αν υπάρχει κάποιο κοινό θέμα που αναφέρεται σε αυτά τα tweets.

Σχήμα 6.5: Tweet των συντηρητικών στις 17/10. Ο Πρωθυπουργός της Αγγλίας Boris Johnson διαπραγματεύτηκε καινούργια συμφωνία για το Brexit και υπήρξε σημαντική μείωση στα τοξικά tweets.

Σχήμα 6.6: Στις 10/10 υπήρξε μια μείωση στα μη τοξικά σχόλια. Την μέρα αυτή ο λογαριασμός των συντηρητικών δεν έκανε tweet σχετικό με το Brexit αλλά αναφέρθηκε στην επίθεση σε εβραϊκή συναγωγή στην Γερμανία.

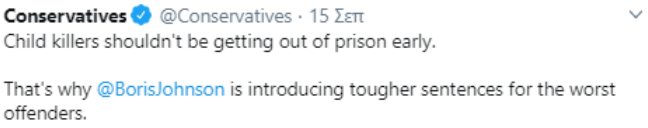
Σχήμα 6.7: Στις 8/10 υπήρξε μια μείωση στα μη τοξικά Tweet έπειτα από τις παρακάτω αναρτήσεις που ανέφεραν ότι θα φέρουν εις πέρας το Brexit στις 31/10 χωρίς βέβαια να έχει υπάρξει συμφωνία.

(α′) Tweet των συντηρητικών στις 6/10 (β′) Tweet των συντηρητικών στις 3/10

Σχήμα 6.8: Στις 3&6/9 βλέπουμε ότι υπήρξε αύξηση στα μη τοξικά σχόλια. Και στις δύο αυτές μέρες έγιναν tweets για διεκπεραίωση του Brexit με την πρόταση καινούργιας συμφωνίας από τον Πρωθυπουργό της Μ.Βρετανίας.

Σχήμα 6.9: Παρόμοια αντίδραση με τις 8/10 βλέπουμε ότι υπήρξε και στις 25/9 όταν έπειτα από tweet για την διεκπεραίωσή του Brexit στις 31/10 υπήρξε μεγάλη μείωση των μη τοξικών tweet.

Σχήμα 6.10: Στις 18/9 βλέπουμε μια αύξηση των μη τοξικών tweets. Την μέρα αυτή οι Συντηρητικοί έκαναν tweet για την αυξηση των μισθών και την ανάπτυξη της Βρετανικής οικονομίας.



Σχήμα 6.11: Στις 15/9 βλέπουμε μια μεγάλη μείωση των τοξικών σχολίων. Την μέρα αυτή οι Συντηρητικοί δεν έκαναν κάποιο tweet σχετικά με το Brexit αλλά ανέφεραν πως αυστηροποιούν τις ποινές για δολοφόνους παιδιών.

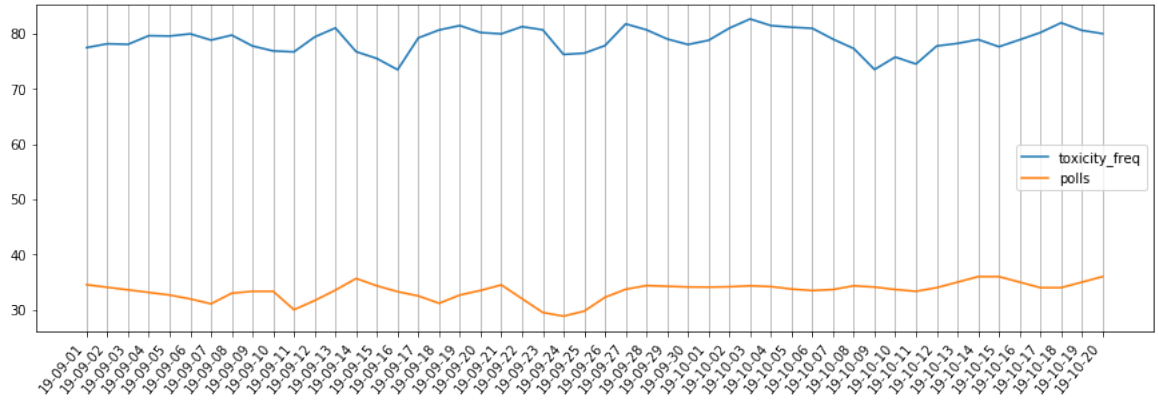
Στις αναρτήσεις που εξετάσαμε βλέπουμε ότι όταν στα tweets αναφερόταν η νέα πρόταση συμφωνίας για έξοδο από την Ευρωπαϊκή Ένωση είχαμε μείωση στα τοξικά tweets. Επίσης στις μέρες με μεγάλη αύξηση τις τοξικότητας των tweets αναφερόταν πως η κυβέρνηση του Η.Β θα προχωρούσε σε διεκπεραίωση του Brexit με ή χωρίς συμφωνία. Επίσης σε δύο περιπτώσεις που έγινε αναφορά σε ειδεχθή εγκλήματα όπως η επίθεση στην εβραϊκή συναγωγή ή σε φόνους παιδιών υπήρξε αύξηση των τοξικών tweets.

6.3.2 Μεταβολή τοξικότητας και δημοσκοπήσεις

Η δευτερη εφαρμογή που εξετάστηκε ήταν σύνδεση της τοξικότητας των tweets με τα αποτελέσματα των δημοσκοπήσεων. Τα δεδομένα για τις δημοσκοπήσεις στην Μεγάλη Βρετανία συλλέχθηκαν από την ιστοσελίδα Britain Elects η οποία συγκεντρώνει δεδομένα από πιστοποιημένες εταιρίες δημοσκοπήσεων στην Μεγάλη Βρετανία.

Επειδή η τιμή της toxicity frequency μετρά τα την συχνότητα μη τοξικών tweets περιμένουμε αν υπάρχει συσχέτιση αυτή να είναι θετική. Δηλαδή η αύξηση της πρόθεσης ψήφου να συμπίπτει με αύξηση της τιμής toxicity frequency που σημαίνει πως έχουμε μείωση των τοξικών tweets. Για να εξεταστεί αν υπάρχει αυτή η συσχέτιση αρχικά παρουσιάστηκαν στο ίδιο διάγραμμα η πορεία της τοξικότητας των tweets και η πορεία των δημοσκοπήσεων. Στο διάγραμμα αυτό θε περιμέναμε οι δύο τιμές να εμφανίζουν ταύχρονη αύξηση ή μείωση.

Στο σχήμα 6.12, παρουσιάζονται στο ίδιο διάγραμμα η πορεία της toxicity frequency που αναφέρθηκε πιο πάνω μαζί με τη πορεία των δημοσκοπήσεων.

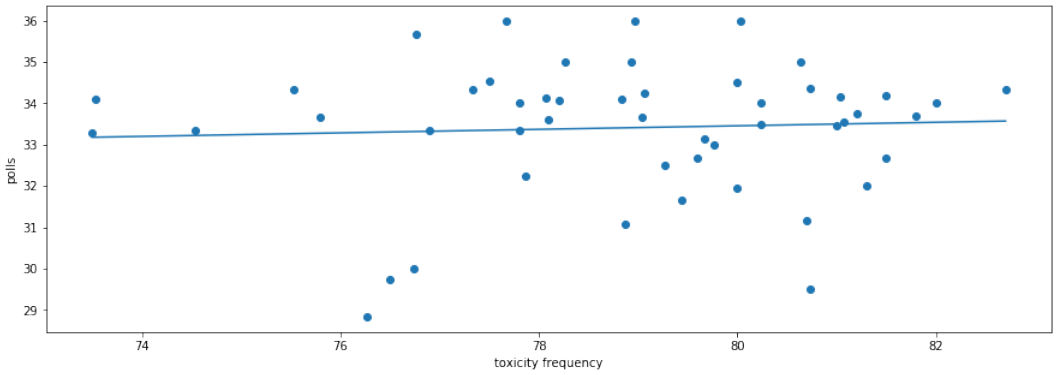


Σχήμα 6.12: Η πορεία των δημοσκοπήσεων μαζί με την πορεία της toxicity frequency

Στο σχήμα 6.12 βλέπουμε τη εξέλιξη των αποτελεσμάτων των δημοσκοπήσεων (polls) και της τιμής toxicity frequency. Δεν φαίνεται να

υπάρχει κάποια ξεκάθαρη συσχέτιση μεταξύ των μεταβλητών.

Στην συνέχεια κατασκευάστηκε το διάγραμμα διασποράς των δύο μεταβλητών. Αν όπως προαναφέρθηκε οι δύο τιμές εμφανίζουν θετική συσχέτιση θα πρέπει να συνδέονται από μια σχέση της μορφής $y = a * x + b$, όπου x, y οι μεταβλητές, a θετική σταθερά και b σταθερά.



Σχήμα 6.13: Διάγραμμα διασποράς της toxicity frequency σε συνάρτηση με τα αποτελέσματα των δημοσκοπήσεων(polls) και η ευθεία που προσαρμόστηκε σε αυτά

Στο σχήμα 6.13 βλέπουμε το διάγραμμα διασποράς των αποτελεσμάτων των δημοσκοπήσεων και της τιμής toxicity frequency. Για να φανεί αν υπάρχει κάποια συσχέτιση χρησιμοποιήθηκε η μέθοδος ελαχίστων τετραγώνων ώστε να δούμε αν υπάρχει κάποια ευθεία της μορφής $y = a * x + b$ που να ακολουθούν τα δεδομένα μας. Βλέπουμε πως δεν υπάρχει κάποια ξεκάθαρη συσχέτιση. Βέβαια πρέπει να αναφέρουμε πως την περίοδο που συλλέχθηκαν τα δεδομένα δεν υπήρχε κάποια ξεκάθαρη ανοδική ή καθοδική τάση η οποία θα καθιστούσε προφανέστερη την συσχέτιση των δύο μεταβλητών.

Κεφάλαιο 7

Επίλογος

Στην παρούσα εργασία εξετάστηκαν μέθοδοι ταξινόμησης βασισμένες στην τεχνική των νευρωνικών δικτύων, οι τεχνικές αναπαράστασης λέξεων σε διανύσματα που χρησιμοποιήθηκαν καθώς και δύο πιθανές εφαρμογές που μπορεί να έχει ένας τέτοιος ταξινομητής σε προβλήματατα του πραγματικού κόσμου,

Όσον αφορά τις αποδόσεις των ταξινομητών είδαμε πως οι προσεγγίσεις των νευρωνικών δικτύων ξεπέρασαν σε απόδοση τις πιο κλασσικές προσεγγίσεις βασισμένες στην στατιστική (Naive Bayes) κάτι που δικαιολογεί την χρήση τους σε προβλήματα επεξεργασίας φυσικής γλώσσας. Επίσης είδαμε πως η μέθοδος όπου έγινε χρήση Embedding Layer απέδωσε καλύτερα σε σχέση με την προεκπαιδευμένη αναπαράσταση Word2Vec κάτι που ίσως σχετίζεται με την επιτυχία των αναπαραστάσεων να ομαδοποιήσουν τις λέξεις. Τέλος είδαμε οι ταξινομητές όπου χρησιμοποιήθηκε 2D συνέλιξη είχαν απόδοση συγκρίσιμη με αυτή της 1D συνέλιξης. Αυτό μας έδειξε πως τα νευρωνικά δίκτυα λειτουργουν ακόμη και όταν τα διανύσματα εξετάζονται τμηματικά κι όχι ως σύνολο όπως στην 1D συνέλιξη. Η επιτυχία των 2D δικτύων θα μπορούσε να σημαίνει πως είναι δυνατόν να χρησιμομποιήσουμε διανυσματικούς χώρους μικρότερης διάστασης ή τεχνικές μείωσες των διαστάσεων των διανυσμάτων (π.χ. Principal Component Analysis) μειώνοντας τον υπολογιστικό φόρτο με σχετικά μικρή μείωση στην απόδοση.

Εξετάζοντας τις αναπαραστάσεις είδαμε πως και οι δύο προσεγγίσεις απέτυχαν να ομαδοποιήσουν τις λέξεις με βάση το περιεχόμενο κάτι που όπως προαναφέρθηκε πιθανότητα οφείλεται στον μικρό αριθμό εμφανίσεων των λέξεων στα δεδομένα μας. Αυτό που αξίζει να αναφερθεί είναι πως η αναπαράσταση που κατασκευαστηκε απο την Embedding Layer ουσιαστικά ''κληρονόμησε'' τις ιδιοτητες του δικτύου για το οποίο εκπαιδεύτηκε ξεχωρίζοντας λέξεις με υβριστικό περιεχόμενο.

Στην υποενότητα των εφαρμογών εξετάστηκε αν υπάρχει κάποιο κοινο θέμα στα tweets τις ημέρες που εμφανιζόταν μεγάλη αύξηση ή μείωση στα τοξικά tweets και αν σχετίζεται αυτή η μεταβολή με τα αποτελέσματα των δημοσκοπήσεων. Όσον αφορά το πρώτο μέρος φάνηκε πως όταν το κόμμα των Συντηρητικών αναφερόταν πως θα φέρει εις πέρας το Brexit υπήρχε μείωση των τοξικών tweets και όταν αναφερόταν πως αυτό θα γίνει χωρίς συμφωνία υπήρχε αύξηση των τοξικών tweets. Όσον αφορά την συσχέτιση της τοξικότητας με τα αποτελέσματα των δημοσκοπήσεων δεν φάνηκε να υπάρχει κάποια ΄συσχέτιση αν και η σχετική σταθερότητα στα αποτελέσματα των δημοσκοπήσεων δεν βοηθά στην αποκάλυψη μιας τέτοιας συσχέτισης.

Αν και χρειάζεται περαιτέρω μελέτη για να εξαχθούν ασφαλή συμπεράσματα μπορούμε να πούμε πως η χρήση νευρωνικών δικτύων για την ταξινόμηση κειμένων δίνει πολύ καλά αποτελέσματα. Κάποιες πιθανές προεκτάσεις τις παρούσας εργασίας θα μπορούμε να είναι η δυνατότητα μείωσης των διαστάσεων του διανυσματικού χώρου των αναπαραστάσεων και η επιδραση αυτού στα αποτελέσματα όπως και η μελέτη της αποτελεσματικότητας των εφαρμογών που παρουσιάστηκαν σε μεγαλύτερα χρονικά πλαίσια ώστε να καταστούν προφανέστερες οι συσχετίσεις που υπάρχουν.

Παράρτημα

Software

Όλα τα προγράμματα που εκτελέσθηκαν για το σκοπό αυτής της εργασίας γράφτηκαν σε Python 3 υπό μορφή Notebook στο περιβάλλον του Google Colab.
Για τον κομμάτι των νευρωνικών δικτύων χρησιμοποιήθηκαν οι βιβλιοθήκες Tensorflow και Keras.
Στα μοντέλα όπου έγινε χρήση του αλγορίθμου Naive Bayes χρησιμοποιήθηκε η βιβλιοθήκη Sklearn.
Η αναπαράσταση Word2Vec υλοποιήθηκε χρησιμοποιώντας την βιβλιοθήκη Gensim.
Τα δεδομένα συγκεντρώθηκαν από το Twitter με τη χρήση του API του Twitter και της βιβλιοθήκης tweetpy.
Links για τα μοντέλα των νευρωνικών δικτύων:

Word2Vec 1D
Word2Vec 2D
Embedding 1D
Embedding 2D

Links για τα μοντέλα Naive Bayes:
Bernoulli Naive Bayes
Multinomial Naive Bayes

Το πρόγραμμα που συνέλεξε τα δεδομένα από το twitter:
Tweet Miner

Και το πρόγραμμα που έγινε ή ταξινόμηση των σχολίων και παράχθη-

καν τα γραφήματα:
Tweet Classifier

Βιβλιογραφία

[1] McCulloch W.S., Pitts W. (1943), ”A logical calculus of the ideas immanent in nervous activity”. Bulletin of Mathematical Biophysics 5: 115. https://doi.org/10.1007/BF02478259

[2] Turing A. M. (1950), ”Computing Machinery and Intelligence. Mind 49: 433-460.

[3] Samuel A. L. (1959), ”Some Studies in Machine Learning Using the Game of Checkers,” IBM Journal of Research and Development, vol. 3, no. 3, pp. 210-229, . doi: 10.1147/rd.33.0210

[4] Moor J. (2006), ”The Dartmouth College Artificial Intelligence Conference: The Next Fifty Years.”, AI Magazine vol. 27

[5] Rosenblatt F. (1958), ”The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain”, Psychological Review 65-386

[6] Widrow B., Hoff M. (1962),””Associative Storage and Retrieval of Digital Information in Networks of Adaptive ’Neurons’”. Biological Prototypes and Synthetic Systems: Volume 1 Proceedings of the Second Annual Bionics Symposium

[7] Hopfield J.J. (1982), ”Neural networks and physical systems with emergent collective computational abilities”. Proceedings of the National Academy of Sciences 79 (8) 2554-2558; DOI: 10.1073/pnas.79.8.2554

[8] Rumelhart D., Hinton, G. , Williams, R. Learning (1986) representations by back-propagating errors. Nature 323, 533–536 . https://doi.org/10.1038/323533a0

[9] Le Q.V., Ranzato M., Monga R., Devin M., Chen K., Corrado G.S., Dean J., Ng A.Y, ”Building High-level (2012) Features Using Large Scale Unsupervised Learning”, 29 th International Conference on Machine Learning

[10] Krizhevsky A.,Sutskever I., Hinton, G.E. (2012), ”ImageNet Classification with Deep Convolutional Neural Networks”, Advances in Neural Information Processing Systems 25 1097-1105

[11] Kiranyaz s., Avci O., Abdeljaber O.,Ince T., Gabbouj M., Inman D.J.(2019), ”1D Convolutional Neural Networks and Applications: A Survey”, International Conference on Acoustics, Speech and Signal Processing (ICASSP)

[12] Mikolov T., Chen K., Corrado G,, Dean J. (2013), ”Efficient Estimation of Word Representations in Vector Space”, CoRR abs/1301.3781

[13] Georgakopoulos S.V., Tasoulis S.K., Vrahatis A.G., Plagianakos V.P. (2018), ”Convolutional Neural Networks for Toxic Comment Classification”, SETN ’18: Proceedings of the 10th Hellenic Conference on Artificial Intelligence

[14] McCallum A., Nigam K.(1998), A comparison of event models for Naive Bayes text classification , Association for the Advancement of Artificial Intelligence