ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ ΠΛΗΡΟΦΟΡΙΑΣ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΝΙΧΝΕΥΣΗ ΤΟΞΙΚΩΝ ΣΧΟΛΙΩΝ ΚΑΙ ΕΛΑΧΙΣΤΟΠΟΙΗΣΗ ΑΚΟΥΣΙΑΣ ΠΡΟΚΑΤΑΛΗΨΗΣ ΜΟΝΤΕΛΟΥ ΜΕ ΧΡΗΣΗ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ

Πτυχιακή Εργασία

του

Γιάντσιου Κωνσταντίνου

Επιβλέπων Καθηγητής: ΙΩΑΝΝΗΣ ΡΕΦΑΝΙΔΗΣ

Θεσσαλονίκη, Αύγουστος 2020

Περίληψη

Το διαδίκτυο αποτελεί μια κοινωνία και όπως σε κάθε κοινωνία υπάρχουν κακόβουλα άτομα έτσι και στο διαδίκτυο υπάρχουν χρήστες που στοχοποιούν μέλη της κοινότητας, κάνοντας χυδαία και προκλητικά σχόλια. Τέτοιες τοξικές συμπεριφορές σε πρώτη φάση αποτρέπουν τα θύματα να ασκήσουν στο μέλλον το δικαίωμα τους στην ελευθερία του λόγου και σε δεύτερη φάση ερημοποιούν την κοινότητα. Σκοπός της παρούσας πτυχιακής είναι η διερεύνηση και πρόβλεψη της τοξικότητας σε σχόλια χρησιμοποιώντας διάφορες αρχιτεκτονικές Νευρωνικών Δικτύων. Το σύνολο δεδομένων αντλήθηκε από τον διαγωνισμό ‘Jigsaw Unintended Bias in Toxicity Classification’ του Kaggle που διοργανώνει η Jigsaw, ερευνητική ομάδα της Google. Οι αρχιτεκτονικές, που συντέθηκαν, είναι 16 στο σύνολο: 6 που κάνουν χρήση LSTM, 6 που κάνουν χρήση GRU, 1 που κάνει χρήση CNN, 1 με χρήση BERT, 1 με χρήση RoBERTa και 1 με χρήση GPT2. Τέλος έγινε χρήση συλλογικής μάθησης δοκιμάζοντας διάφορους συνδυασμούς για τις 4 καλύτερες αρχιτεκτονικές. Καλύτερα αποτελέσματα παρουσίασε η χρήση και των τεσσάρων καλύτερων αρχιτεκτονικών κατατάσσοντας την συγκεκριμένη λύση στο κορυφαίο 6% των καλύτερων λύσεων του διαγωνισμού.

Λέξεις Κλειδιά: νευρωνικά δίκτυα, NLP, toxicity classification, LSTM, GRU, TextCNN, Transformers, BERT, RoBERTa, GPT2, Ensemble Learning

Αποποίηση ευθυνών: Η εργασία περιέχει κείμενο που μπορεί να θεωρηθεί άσεμνο, χυδαίο ή προσβλητικό.

Abstract

The internet constitutes a society and as in every society there are malicious people so there are users on the internet who victimize other members of the community by making vulgar and provocative comments. Such toxic behaviors in first phase prevent the victims from exercising their right to freedom of speech in the future and in second phase they desert the community. The purpose of this thesis is to investigate and predict the toxicity in comments using various Neural Network architectures. The data set was taken from Kaggle's ‘Jigsaw Unintended Bias in Toxicity Classification’ competition organized by Jigsaw, a Google research team. The architectures, synthesized, are 16 in total: 6 using LSTM, 6 using GRU, 1 using CNN, 1 using BERT, 1 using RoBERTa and 1 using GPT2. Finally, ensemble learning was used, testing various combinations for the 4 best architectures. The best results were shown by the use of all four best architectures ranking this solution in the top 6% of the best solutions of the competition.

Keywords: neural networks, NLP, toxicity classification, LSTM, GRU, TextCNN, Transformers, BERT, RoBERTa, GPT2, Ensemble Learning

Disclaimer: This thesis contains text that may be considered profane, vulgar, or offensive.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή Ρεφανίδη Ιωάννη που με καθοδήγησε καθ΄ όλη τη διάρκεια εκπόνησης της πτυχιακής. Επιπλέον ευχαριστώ την οικογένεια μου για την υποστήριξη που μου παρείχε.

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Πινάκων

1 Εισαγωγή

Οι αρχές του 21ου αιώνα έμελλαν να αποτελούν και οι αρχές μια νέας εποχής, της εποχή του διαδικτύου. Πλέον οι χρήστες από απλοί παρατηρητές περιεχομένου διαδικτύου έχουν γίνει δημιουργοί. Το διαδίκτυο πριν το 1999 δεν διευκόλυνε έναν απλό χρήστη έναν απλό χρήστη να δημιουργήσει περιεχόμενο, αυτό άλλαξε με τον ερχομό του web 2.0. Σήμερα βρισκόμαστε στα τέλη του web 2.0 και στον ερχομό του web 3.0, όπου τεχνολογίες μηχανικής μάθησης και τεχνητής νοημοσύνης θα πρωταγωνιστούν. Το web 3.0 θα είναι ένας σημασιολογικός ιστός που θα βασίζεται στα δεδομένα, ο χρήστης θα μπορεί να πληκτρολογεί ένα ερώτημα στο διαδίκτυο και αυτό θα μπορεί να κατανοήσει το πλαίσιο της ερώτησης και θα επιστρέφει απαντήσεις που καλύπτουν τις ανάγκες του χρήστη. Επιπλέον στην ανάπτυξη του διαδικτύου βοήθησε και η ανάπτυξη των smartphones, όπου επιτρέπουν στον χρήστη να είναι συνέχεια συνδεδεμένος.

Παρόλα αυτά η ευρεία διάδοση της χρήσης του διαδικτύου έχει επιτρέψει και σε κακόβουλα άτομα να συμμετάσχουν στις κοινότητες που έχουν δημιουργηθεί. Αυτοί οι χρήστες πολλές φορές εκφοβίζουν και προπηλακίζουν με υβριστικά και τοξικά σχόλια διακινδυνεύοντας την ελευθερία και την ακεραιότητα των διαδικτυακών κοινοτήτων. Η ομάδα Conversation AI, μια ερευνητική πρωτοβουλία της Jigsaw και Google (και οι δύο τμήματα της Alphabet), δημιουργήθηκε για την ανάπτυξη τεχνολογίας η οποία θα μπορεί να προστατέψει τις απόψεις των χρηστών σε μια διαδικτυακή συνομιλία. Συγκεκριμένα η Jigsaw προσπαθεί να προβλέψει και να αντιμετωπίσει αναδυόμενες απειλές όπως η παραπληροφόρηση, η λογοκρισία, η παρενόχληση και ο βίαιος εξτρεμισμός δημιουργώντας τεχνολογία και εκπονώντας έρευνα καθοριστική για την διατήρηση της ασφάλειας του διαδικτυακού αλλά και του πραγματικού κόσμου.

Η εργασία αυτή επικεντρώνεται στην ανάπτυξη μοντέλων μηχανικής μάθησης, συγκεκριμένα νευρωνικών Δικτύων, όπου θα μπορούν να ανιχνεύσουν την τοξικότητα σε σχόλια. Η τοξικότητα ορίζεται ως οτιδήποτε αγενή, ασεβή ή όποιος τρόπος πιθανόν αναγκάζει έναν χρήστη να φύγει από μια συζήτηση. Για την εκπόνηση της εργασίας χρειάστηκε η συμμετοχή στον διαγωνισμού του Kaggle που διεξήχθη από την Jigsaw. Δυστυχώς η συμμετοχή έγινε αφού είχε τελειώσει ο διαγωνισμός (offline).

2 Βαθιά Νευρωνικά Δίκτυα

2.1 Τεχνητά Νευρωνικά Δίκτυα

2.1.1 Ιστορική Αναδρομή

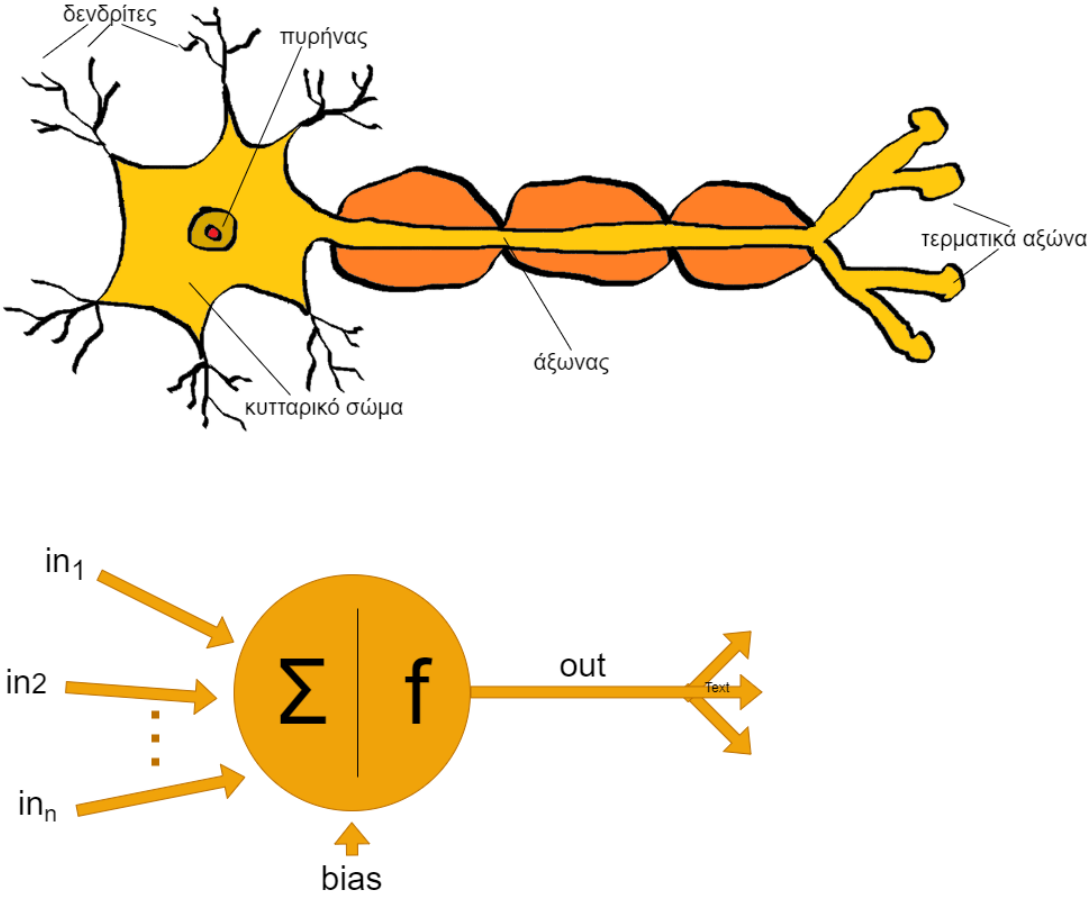
Τους δύο τελευταίους αιώνες το ανθρώπινο είδος έχει βιώσει αλλαγές χωρίς προηγούμενο ειδικά στον τομέα της τεχνολογίας. Περάσαμε από την εποχή του ατμού και μέσα από την δεύτερη βιομηχανική επανάσταση και την ψηφιακή επανάσταση, η οποία ξεκίνησε στα τέλη της δεκαετίας του 1940, βρισκόμαστε πλέον στην εποχή της Τεχνολογίας της Πληροφορίας. Οι έρευνες και οι καινοτομίες που διεξήχθησαν σε όλη αυτή την περίοδο κάνουν την ζωή μας ευκολότερη βοηθώντας τόσο σε πιο ευκαταφρόνητα θέματα, όπως είναι τα κλιματιζόμενα υποδήματα, αλλά και σε πιο σοβαρά όπως είναι η υγεία μας (ανίχνευση και θεραπεία καρκίνου) ή η διοίκηση μιας επιχείρησης (πρόβλεψη τιμών, εφοδιαστική αλυσίδα). Αναμφίβολα για όλες αυτές τις περίπλοκες διαδικασίες που είμαστε σε θέση να φέρουμε εις πέρας σήμερα οφείλουμε να πιστώσουμε τα εύσημα μας στη επιστήμη των υπολογιστών που μέσα σε λίγες δεκαετίες υπερέβηκε επανειλημμένα τα όρια που θέταμε για αυτήν.

Σημαντική συνεισφορά στην επεξεργασία δεδομένων και παραγωγή αποτελεσμάτων, χρήσιμων για την εκπλήρωση μιας περίπλοκης διαδικασίας, έχουν τα υπολογιστικά μοντέλα. Τα Τεχνητά Νευρωνικά δίκτυα (ΤΝΔ) είναι ένα υπολογιστικό μοντέλο εμπνευσμένο από την λειτουργία του εγκεφάλου.

Ήδη από τον 19^{ο} αιώνα οι επιστήμονες αποδέχονται ότι ο εγκέφαλος συντίθεται από ένα σύνολο νευρωνικών δικτύων αποτελούμενα από διακριτά στοιχεία, τους νευρώνες (neurons), που επικοινωνούν το ένα με το άλλο. Οι νευρώνες αποτελούνται από :

- Το Σώμα, περιλαμβάνει το πυρήνα του κυττάρου.
- Τους δενδρίτες, επιτρέπουν στο κύτταρο να λαμβάνει σήματα από συνδεδεμένους γειτονικά νευρώνες.
- Τον Άξονα, το σήμα ταξιδεύει μέσω αυτού για να φτάσει στις συνάψεις.

- Τις συνάψεις, ενώνουν τον άξονα ενός νευρώνα με τους δενδρίτες άλλων νευρώνων και μεταφέρουν το σήμα.



Εικόνα 2-1: Βιολογικός και τεχνητός νευρώνας

Οι Warren McCulloch και Walter Pitts προσπάθησαν για πρώτη φορά το 1943 να υλοποιήσουν την λειτουργία του νευρικού συστήματος ως υπολογιστικό μοντέλο προτείνοντας την Λογική Μονάδα Κατωφλιού (Threshold Logic Unit), το οποίο ήταν ένα απλό μοντέλο που μπορούσε να μάθει τις λογικές πύλες AND και OR [1].

Έπειτα το 1957 ο Frank Rosenblatt εφηύρε το μοντέλο του απλού αισθητήρα (Perceptron), ένα γραμμικό μοντέλο δυαδικής κατηγοριοποίησης (classification) που δεν μπορεί να ταξινομήσει μη γραμμικώς διαχωρίσιμα προβλήματα.

Το 1986 οι David Rumelhart, Geoffrey Hinton and Ronald J. Williams έδειξαν πως η μέθοδος οπισθοδιάδοσης για την εκπαίδευση Τεχνητών Νευρωνικών Δικτύων επιφέρει ποιοτικά αποτελέσματα.

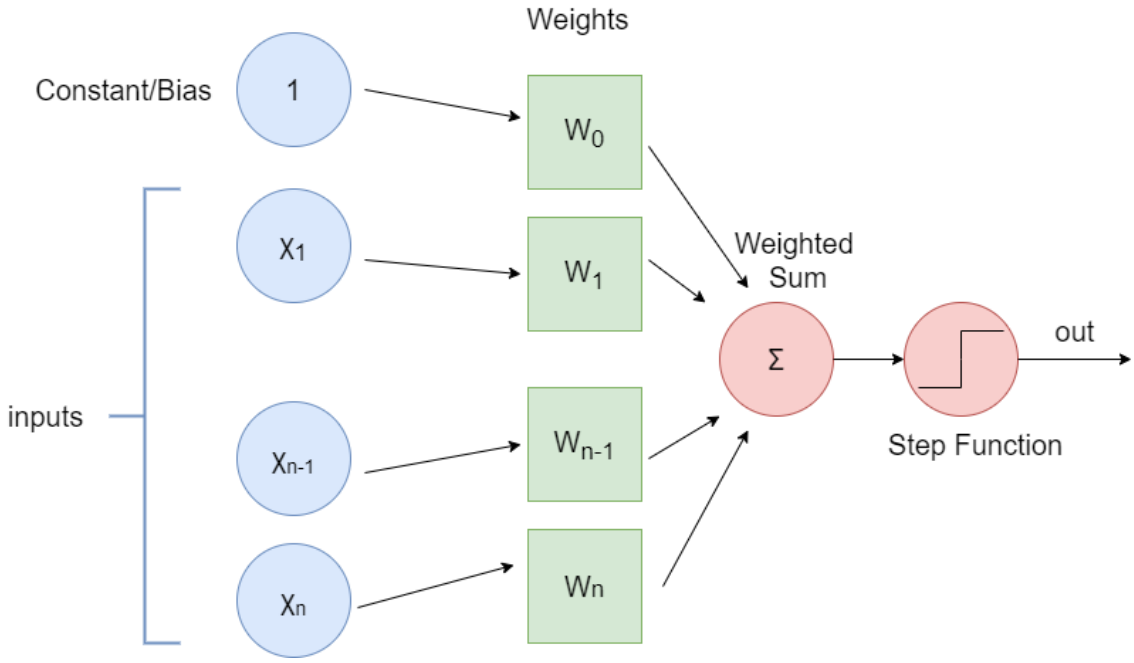
Την δεκαετία του 1990, οι Sepp Hochreiter and Jürgen Schmidhuber το 1997 πρότειναν το Long Sort-Term Memory δίκτυο για να αντιμετωπιστεί το πρόβλημα των φθινουσών κλίσεων (vanishing gradient) στα Επαναληπτικά Νευρωνικά Δίκτυα (RNN).

Το 1998 ο Yann LeGun καθιέρωσε τα Συνελικτικά Δίκτυα (Convolutional Networks) όπως τα ξέρουμε σήμερα.

2.1.2 Αρχιτεκτονική

Βασικό χαρακτηριστικό ενός Τεχνητού Νευρωνικού Δικτυού είναι η αρχιτεκτονική του. Ένα Τεχνητό Νευρωνικό Δίκτυο αποτελείτε από Τεχνητούς Νευρώνες (Perceptrons). Ένας Τεχνητός Νευρώνας έχει πέντε συστατικά μέρη:

1. Τις τιμές των εξόδων ή σήματα εισόδου από το προηγούμενο επίπεδο.
2. Τα βάρη των συνδέσεων, τα οποία είναι συντελεστές που κλιμακώνουν το σήμα εισόδου ενισχύοντας ή αποδυναμώνοντάς το. Στις περισσότερες αναπαραστάσεις των νευρωνικών δικτύων είναι οι ακμές που συνδέουν τους κόμβους.
3. Την τάση πόλωσης ως σταθερά (bias). Χρησιμοποιείται για να εξασφαλίσουμε ότι μερικοί νευρώνες ανά επίπεδο ενεργοποιούνται ανεξαρτήτως την δύναμη του σήματος και έτσι επιτρέπουν να γίνει η μάθηση σε περιπτώσεις αδύναμου σήματος.
4. Την συνολική είσοδο.
5. Την συνάρτηση ενεργοποίησης.



Εικόνα 2-2: Τεχνητός Νευρώνας (Perceptron)

Για την τελική έξοδο του νευρώνα ακολουθούνται τα εξής βήματα :

I. Όλες οι είσοδοι πολλαπλασιάζονται με τα αντίστοιχα βάρη τους και αθροίζονται για να υπολογιστεί η συνολική είσοδος του νευρώνα. Στην συνολική είσοδο προστίθεται και η σταθερά που συνήθως ισούται με ένα και δεν έχει βάρος ή άμα έχει βάρος το γινόμενο του πολλαπλασιασμού τους ισούται με ένα.

$$S_i = \sum_j a_j w_{j,i} + b$$

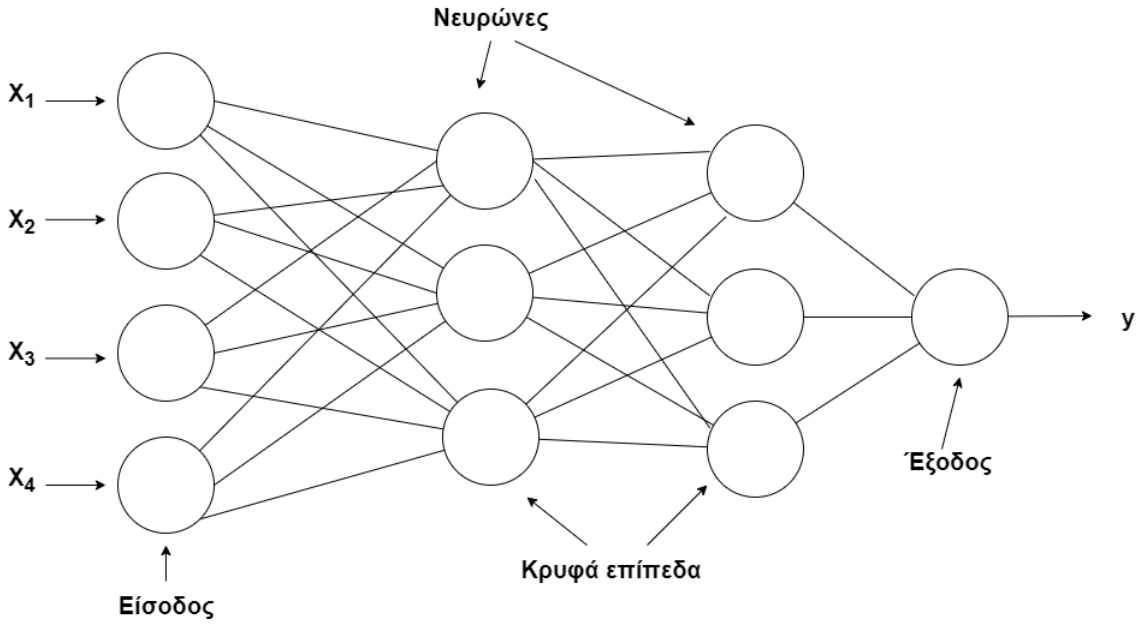
, όπου S_i είναι η συνολική είσοδος του νευρώνα i, a_j είναι οι έξοδοι των νευρώνων j, $w_{j,i}$ είναι τα βάρη και b είναι η σταθερά.

II. Εφαρμογή της συνάρτησης ενεργοποίησης στην συνολική είσοδο για να προκύψει η έξοδος του νευρώνα.

$$a_i = \Phi(S_i)$$

, όπου η a_i είναι η έξοδος του νευρώνα i και Φ η συνάρτηση ενεργοποίησης.

Ένα Τεχνητό Νευρωνικό Δίκτυο πολλών επιπέδων (Multilayer Perceptron) συγκροτείτε από ένα επίπεδο εισόδου, κανένα ένα ή περισσότερα κρυφά επίπεδα και ένα επίπεδο εξόδου. Κάθε επίπεδο εκτός του επιπέδου εισόδου έχει ένα ή περισσότερους νευρώνες που ενώνονται με τους νευρώνες του προηγούμενου επιπέδου με συνδέσεις που έχουν βάρη.



Εικόνα 2-3: Νευρωνικό Δίκτυο Πολλών Επιπέδων (Multilayer Perceptron)

Το επίπεδο εισόδου είναι υπεύθυνο για την είσοδο των δεδομένων στο δίκτυο ως διανύσματα τιμών, δεν έχει νευρώνες γιατί απλώς στέλνει τα σήματα εισόδου στο επόμενο επίπεδο. Συνήθως είναι πλήρως συνδεδεμένο με το κρυφό επίπεδο που το ακολουθεί, ωστόσο σε ορισμένες αρχιτεκτονικές μπορεί το επίπεδο εισόδου να είναι μερικώς συνδεδεμένο.

Τα κρυφά επίπεδα επιτρέπουν το Νευρωνικό Δίκτυο να εκπαιδευτεί σε μη-γραμμικώς διαχωρίσιμες συναρτήσεις, όπως προαναφέρθηκε αυτό ήταν περιορισμός για ένα δίκτυο με έναν απλό νευρώνα (perceptron). Οι τιμές των βαρών στις συνδέσεις μεταξύ των επιπέδων είναι ο τρόπος που τα Νευρωνικά Δίκτυα κωδικοποιούν την πληροφορία που εξήγαγαν από τα αρχικά δεδομένα εκπαίδευσης.

Στο επίπεδο εξόδου παίρνουμε την απάντηση ή την πρόβλεψη του μοντέλου. Ανάλογα το πρόβλημα και την αρχιτεκτονική του δικτύου που προσπαθεί να το λύσει η έξοδος μπορεί να είναι ένας πραγματικός αριθμός για προβλήματα παλινδρόμησης (regression) ή ένα σύνολο πιθανοτήτων για προβλήματα ταξινόμησης (classification). Αυτό ελέγχεται εύκολα από τον τύπο της συνάρτησης ενεργοποίησης που χρησιμοποιούμε στους νευρώνες στο επίπεδο εξόδου. Συνήθως χρησιμοποιούνται για το επίπεδο εξόδου συναρτήσεις ενεργοποίησης όπως η SoftMax και η σιγμοειδής για ταξινόμηση που θα αναλυθούν παρακάτω. Η διαδικασία υπολογισμού των εξόδων ή της εξόδου ενός

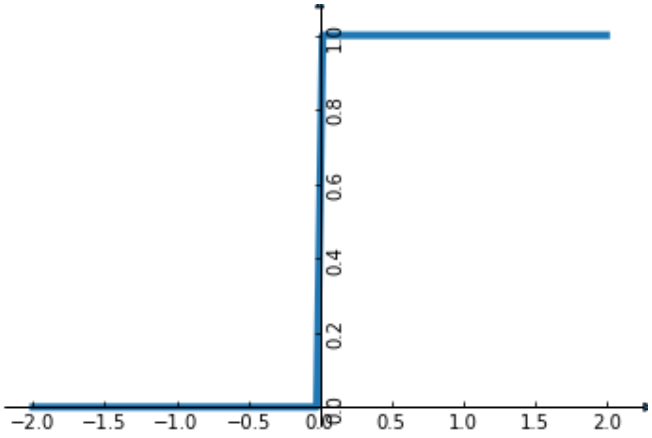
νευρωνικού δικτύου διασχίζοντας και λαμβάνοντας τις εξόδους από όλους τους νευρώνες όλων των επιπέδων, όπως προαναφέρθηκε, ονομάζεται απλή τροφοδότηση (feedforward).

2.1.3 Συναρτήσεις ενεργοποίησης

Η συνάρτηση ενεργοποίησης είναι κυρίαρχο κομμάτι του νευρώνα και ορίζει την συμπεριφορά του. Χρησιμοποιούμε συναρτήσεις ενεργοποίησης για να διαδώσουμε τις εξόδους των νευρώνων ενός επιπέδου στους νευρώνες του επομένου και συγκεκριμένα για τα κρυφά επίπεδα τις χρησιμοποιούμε για να εισάγουμε την μη-γραμμικότητα στις δυνατότητες προσομοίωσης του νευρωνικού δικτύου. Οι συναρτήσεις ενεργοποίησης είναι συναρτήσεις μονόμετρο μέγεθος προς μονόμετρο μέγεθος (scalar-to-scalar) και καθορίζουν την ενεργοποίηση ενός νευρώνα. Οι κυριότερες συναρτήσεις ενεργοποίησης περιγράφονται παρακάτω:

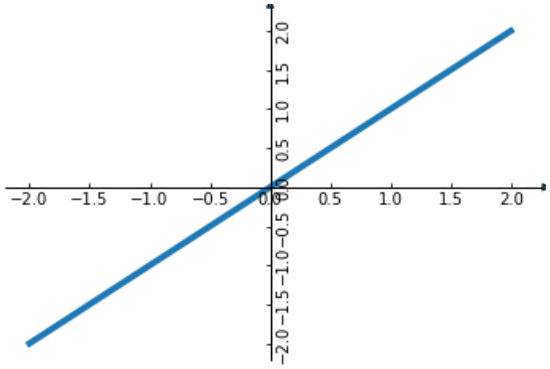
- Βηματική συνάρτηση (step function) ή συνάρτηση κατωφλιού (threshold function) :

$$\Phi(S) = \begin{cases} 1, \alpha\nu\ S > 0 \\ 0, \alpha\nu\ S \leq 0 \end{cases}$$



Εικόνα 2-4: Βηματική Συνάρτηση (Step function)

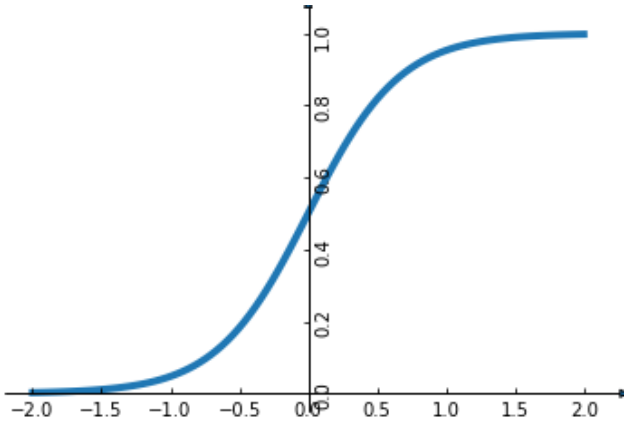
- Γραμμική συνάρτηση (linear function). Πρακτικά επιστρέφει αμετάβλητο το σήμα εισόδου. Χρησιμοποιείτε κυρίως στους νευρώνες εξόδου για προβλήματα παλινδρόμησης.



Εικόνα 2-5: Γραμμική συνάρτηση (linear function)

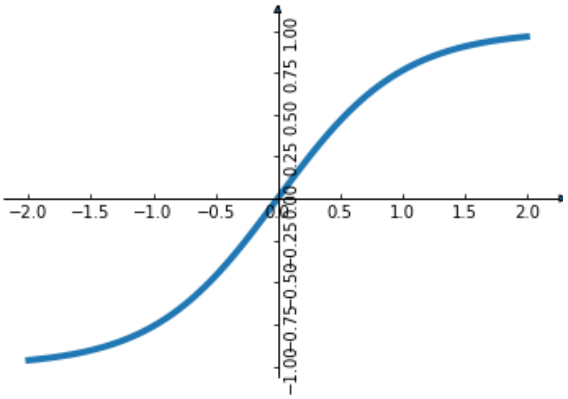
- Σιγμοειδής συνάρτηση (Sigmoid ή logistic function). Χρησιμοποιείται κατά κόρον σε πολλές αρχιτεκτονικές γιατί μπορεί να μείωση τις υπερβολικές τιμές ή το θόρυβο στα δεδομένα χωρίς να τα εξαφανίζει. Παίρνει τιμές από 0 έως 1.

$$\Phi(S) = \frac{1}{1 + e^{-a \cdot S}}$$



Εικόνα 2-6: Σιγμοειδής συνάρτηση (Sigmoid function)

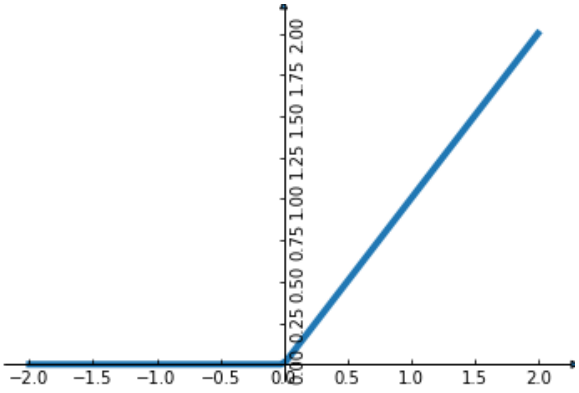
- Υπερβολική εφαπτομένη συνάρτηση (Hyperbolic tangent function). Σε αντίθεση με την σιγμοειδή παίρνει τιμές από -1 έως 1 και δια διαχειρίζεται εύκολο τις αρνητικές τιμές ως είσοδο.



Εικόνα 2-7: Υπερβολική εφαπτομένη συνάρτηση (Hyperbolic tangent function)

- Διορθωμένη γραμμική μονάδα (Rectified Linear Unit ή ReLU). Είναι η ισχύουσα "state of the art" συνάρτηση ενεργοποίησης χρησιμοποιείται σε πολλές περιπτώσεις και έχει αποδειχθεί να λειτουργεί εξίσου ικανοποιητικά.

$$\Phi(S) = \begin{cases} S, \alpha\nu\ S > 0 \\ 0, \alpha\nu\ S \leq 0 \end{cases}$$

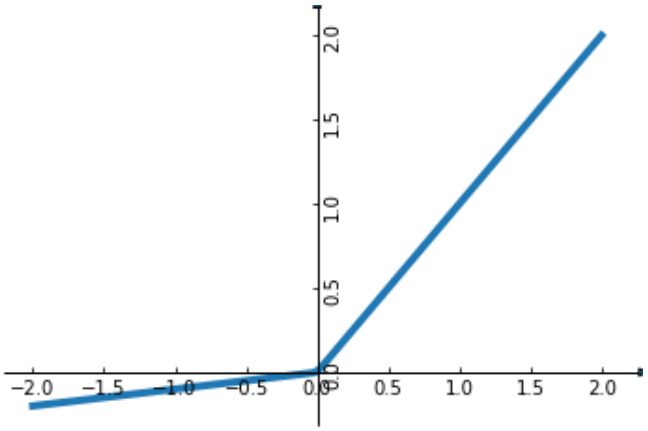


Εικόνα 2-8: Διορθωμένη γραμμική μονάδα (ReLU)

Ένα μειονέκτημα που έχει η ReLU είναι πως λόγω της πρώτης παραγώγου της, η οποία ισούται είτε με μηδέν είτε με μια σταθερά, μπορεί να εμφανιστεί το πρόβλημα το πρόβλημα των νεκρών Νευρώνων (Dying ReLUs). Για να καταπολεμηθεί το πρόβλημα των χρησιμοποιήθηκε μια παραλλαγή της ReLU, η Leaky ReLU, η οποία δεν θεωρείτε καλύτερη [2].

$$\Phi(S) = \begin{cases} S, \alpha\nu\ S > 0 \\ 0.01 \cdot S, \alpha\nu\ S \leq 0 \end{cases}$$

, όπου 0.01 μια σταθερά που μπορεί να αλλάξει.



Εικόνα 2-9: Leaky ReLU

- SoftMax συνάρτηση ενεργοποίησης. Συναντάται κυρίως στους νευρώνες του επιπέδου εξόδου. Η διαφορά με τις προηγούμενες συναρτήσεις είναι ότι η έξοδος ενός νευρώνα επηρεάζεται από τις εξόδους και άλλων νευρώνων. Ουσιαστικά ανακατανέμει τις τιμές των εξόδων των νευρώνων στο διάστημα 0 έως 1, ώστε αυτές να έχουν άθροισμα 1 δίνοντας μεγαλύτερο μερίδιο στους νευρώνες με την μεγαλύτερη συνολική είσοδο [3].

$$Softmax(S)_i = \frac{e^{S_i}}{\sum_{k=1}^{n} e^{S_k}}$$

, όπου n το πλήθος των νευρώνων στο επίπεδο.

2.1.4 Μάθηση

Η διαδικασία της μάθησης για κάθε αλγόριθμο που χρησιμοποιεί βάρη είναι η διαδικασία της αναπροσαρμογής των βαρών, προσθέτοντας ή αφαιρώντας κάποια ορισμένη ποσότητα από τα βάρη. Με αυτόν τον τρόπο το μοντέλο μαθαίνει ποια χαρακτηριστικά, του συνόλου δεδομένων, συνδέονται με ποιες εξόδους ή αποτελέσματα και προσαρμόζει ανάλογα τα βάρη. Οι βασικές μέθοδοι μάθησης είναι οι εξής :

- Μάθηση με επίβλεψη (supervised learning). Όλα τα δεδομένα έχουν ετικέτα (label) και το μοντέλο μαθαίνει να προβλέπει αυτήν την ετικέτα. Χρησιμοποιείται ως επί το πλείστον σε προβλήματα παλινδρόμησης (regression) και ταξινόμησης (classification).

- Μάθηση χωρίς επίβλεψη (unsupervised learning). Αντίθετα με την μάθηση με επίβλεψη όλα τα δεδομένα δεν έχουν ετικέτα και το μοντέλο μαθαίνει την διάρθρωση των δεδομένων. Χρησιμοποιείται σε προβλήματα ομαδοποίησης ή συσταδοποίησης (clustering).

- Ενισχυτική Μάθηση (reinforcement learning). Διαφέρει από τις δύο προηγούμενες κατηγορίες μάθησης ως προς το ότι μοντέλο αλληλοεπιδρά άμεσα με ένα δυναμικό περιβάλλον στο οποίο πρέπει να επιτευχθεί κάποιος συγκεκριμένος στόχος, με την προϋπόθεση ότι δίνετε μερική αναπληροφόρηση (feedback) στο μοντέλο για τις προβλέψεις του [4].

Ένα σημαντικό κομμάτι της μάθησης ενός νευρωνικού δικτύου είναι η Οπισθοδιάδοση Σφάλματος (Backpropagation Learning). Στην περίπτωση της επιβλεπόμενης μάθησης ο βασικός αλγόριθμος για να εφαρμόσουμε μάθηση με Οπισθοδιάδοση έχει δύο βήματα. Αρχικά στο εμπρός πέρασμα υπολογίζεται η έξοδος τροφοδοτώντας το νευρικό δίκτυο με το διάνυσμα εισόδου μιας εγγραφή του συνόλου δεδομένων και διαδίδοντας προς τα εμπρός τα σήματα εισόδου στα κρυφά επίπεδα και από εκεί στο επίπεδο εξόδου. Στο επόμενο βήμα, δηλαδή στο προς τα πίσω πέρασμα

μετράμε την απόκλισή ή σφάλμα της εξόδου που προέβλεψε το νευρωνικό δίκτυο από την πραγματική έξοδο ή αλλιώς ετικέτα, η οποία περιγράφει την εγγραφή, και άμα δεν υπάρχει απόκλιση δεν κάνουμε τίποτα αλλιώς τα βάρη των συνδέσεων όλων των νευρώνων με κατεύθυνση από το επίπεδο εξόδου προς το επίπεδο εισόδου προσαρμόζονται σύμφωνα με την συμμετοχή τους στα στο σφάλμα τις εξόδου, έτσι ώστε η επόμενη πρόβλεψη να συγκλίνει στην πραγματική έξοδο. Αναλυτικότερα τα βήματα του αλγορίθμου:

1. Προς τα εμπρός βήμα: Διάδοση των σημάτων εισόδου για κάθε νευρώνα και κάθε επίπεδο με κατεύθυνση από το επίπεδο εισόδου στο επίπεδο εξόδου χρησιμοποιώντας τον τρόπο που προαναφέρθηκε στο κεφάλαιο 2.1.2 για τον υπολογισμό εξόδου του νευρώνα.

2. Προς τα πίσω βήμα:
   Υπολογισμός σφάλματος στο επίπεδο εξόδου. Έστω στο επίπεδο εξόδου υπάρχει ένας μόνο νευρώνας k. Υπολογισμός σφάλματος για τον νευρώνα k:

   $$err_k = \hat{y}_k - y_k$$

   , όπου $\hat{y}_k$ είναι η έξοδος του νευρώνα k και y_k είναι η επιθυμητή έξοδος.

   Υπολογισμός προσαρμοσμένου σφάλματος :

   $$\delta_k = err_k \frac{d\Phi}{dS_k} = err_k \Phi'(S_k)$$

   Υπολογισμός σφάλματος στα κρυφά επίπεδα. Τα σφάλματα των νευρώνων των κρυφών επιπέδων υπολογίζονται από τα σφάλματα των νευρώνων του αμέσως επόμενου επιπέδου. Έστω ένας νευρώνας h σε ένα κρυφό επίπεδο το σφάλμα του νευρώνα h υπολογίζεται ως εξής :

   $$\delta_h = \sum_{k=1} w_{h,k} \cdot \delta_k \frac{d\Phi}{dS_h} = \sum_{k=1} w_{h,k} \cdot \delta_k \Phi'(S_h)$$

, όπου το άθροισμα αναφέρεται σε όλους του νευρώνες k του επόμενου επιπέδου (ή των επόμενων επιπέδων) με τους οποίους ο νευρώνας h συνδέεται με βάρη w_{hk}.

Έχοντας υπολογίσει για κάθε νευρώνα i το σφάλμα δ_i η αλλαγή των βαρών του νευρώνα i σύμφωνα με τον κανόνα δέλτα, δηλαδή τα βάρη πρέπει να αλλάξουν προς κατεύθυνση αντίθετη του ρυθμού αύξησης του σφάλματος υπολογίζεται ως εξής :

$$\Delta_{w_{j,i}} = -d\,\frac{\partial err_i}{\partial w_{j,i}} = -d \cdot \delta_i \cdot a_j$$

$$w_{j,i} = w_{j,i} + \Delta_{w_{j,i}}$$

, όπου a_j είναι η έξοδος του νευρώνα i του προηγούμενου επιπέδου

d είναι ο ρυθμός μάθησης (learning rate) και $\Delta_{w_{j,i}}$είναι η αλλαγή στο βάρος $w_{j,i}$(δηλαδή το βάρος της σύνδεσης του νευρώνα j του προηγούμενου επιπέδου με τον νευρώνα i).

2.1.5 Συναρτήσεις Σφάλματος

Ένα σημαντικό κομμάτι της σχεδίασης ενός νευρωνικού δικτύου είναι η επιλογή μιας συνάρτησης σφάλματος. Η συνάρτηση σφάλματος (loss ή cost function) υπολογίζει πόσο κοντά είναι το νευρωνικό δίκτυο στο βέλτιστο. Η βασική ιδέα για την εφαρμογή μιας συνάρτησης σφάλματος είναι απλή. Αρχικά υπολογίζουμε το σφάλμα για κάθε πρόβλεψη του νευρωνικού δικτύου, έπειτα παίρνουμε τον μέσο όρο από τα σφάλματα που παρατηρήθηκαν για τα παραδείγματα εκπαίδευσης και πλέον έχουμε ένα νούμερο που αντιπροσωπεύει αν το νευρωνικό δίκτυο συγκλίνει σε κάποιο τοπικό ελάχιστο. Χρησιμοποιώντας μια συνάρτηση σφάλματος μπορούμε να αναφερόμαστε στην εκπαίδευση του νευρωνικού δικτύου ως πρόβλημα βελτιστοποίησης. Στο προηγούμενο κεφάλαιο για να αναπαραστήσουμε το σφάλμα για ένα παράδειγμα εκπαίδευσης αναφέρθηκε ο τύπος με μορφή :

$$err_k = \hat{y}_k - y_k$$

Πρακτικά η συνάρτηση σφάλματος επιλέγεται ανάλογα με τον σκοπό ανάπτυξης του δικτύου.

Για προβλήματα παλινδρόμησης προτιμάται το μέσο τετραγωνικό σφάλμα (mean squared error), επειδή μπορεί να αναπαραστήσει καλύτερα το σφάλμα όταν στην έξοδο του δικτύου περιμένουμε πραγματικό αριθμό. Το μέσο τετραγωνικό σφάλμα εκφράζεται από τον τύπο:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

, όπου n ο αριθμός των παραδειγμάτων εκπαίδευσης.

Αν και το μέσο τετραγωνικό σφάλμα χρησιμοποιείται ευρέως για τα προβλήματα παλινδρόμησης, υπάρχουν και παραλλαγές του που σε κάποιες περιπτώσεις ενδείκνυνται. Μια παραλλαγή του μέσου τετραγωνικού σφάλματος είναι το μέσο απόλυτο σφάλμα (mean absolute error) :

$$MSE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Για προβλήματα ταξινόμησης είθισται η χρήση του σφάλματος διασταυρωμένης εντροπίας (cross entropy loss). Στην θεωρία της πληροφορίας η διασταυρωμένη εντροπία μεταξύ δύο κατανομών πιθανότητας πάνω στο ίδιο σύνολο συμβάντων μετρά τον μέσο αριθμό των bit που χρειάζονται για να αναπαρασταθεί ένα συμβάν, δηλαδή την αταξία μεταξύ των δυο πιθανοτήτων ή την δυσκολία πρόβλεψης του συμβάντος δεδομένου μιας πραγματικής κατανομής p και της κατανομής q που προκύπτει από το δίκτυο. Η διασταυρωμένη εντροπία για k αριθμό κλάσεων ορίζεται ως εξής:

$$H(p,q) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} p(c_j)\log\left(q(c_j)\right)$$

, όπου p η πραγματική κατανομή, η οποία τις περισσότερες φορές αποδίδει πιθανότητα 1 σε μια κλάση και πιθανότητα 0 στις υπόλοιπες (αλλά όχι απαραίτητα) και q η κατανομή που προκύπτει από το δίκτυο. Ο παραπάνω τύπος μπορεί να τροποποιηθεί ανάλογα με τον αριθμό των κλάσεων. Για παράδειγμα για πρόβλημα ταξινόμησης δύο κλάσεων χρησιμοποιείται η δυαδική διασταυρωμένη εντροπία (binary cross-entropy loss) που έχει την παρακάτω μορφή:

$$H(p,q) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{2} p(c_j)\log\Big(q\big(c_j\big)\Big)$$

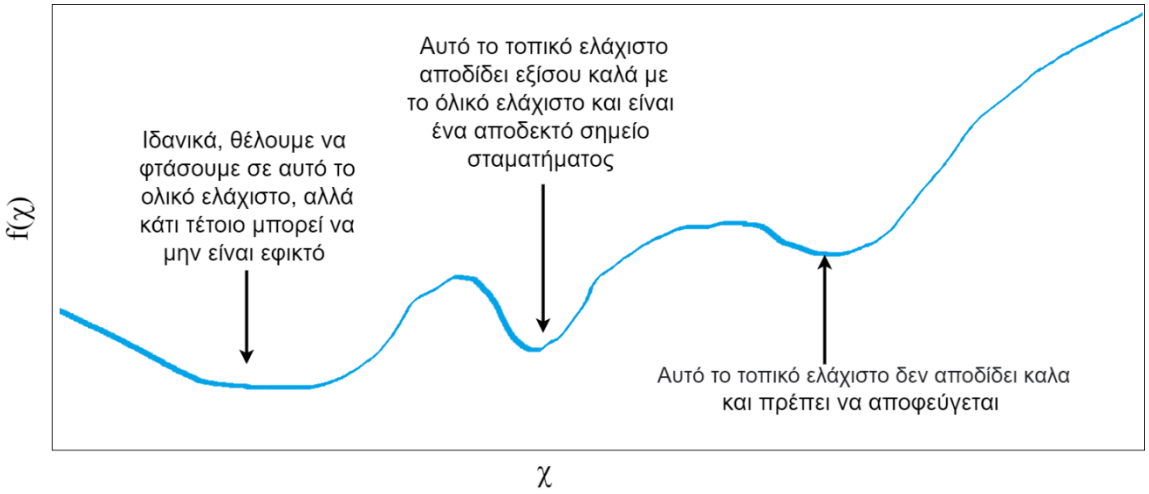
$$= -\frac{1}{n}\sum_{i=1}^{n} p(c_1)\log\big(q(c_1)\big) + (1-p(c_1))\log\big(1-\ q(c_1)\big)$$

Φυσικά το σφάλμα για ένα παράδειγμα εκπαίδευσης εκφράζεται με τους ίδιους τύπους χωρίς το εξωτερικό άθροισμα και την διαίρεση που γίνεται για να υπολογιστεί το μέσο σφάλμα. Για να χρησιμοποιηθεί το σφάλμα διασταυρωμένης εντροπίας το νευρωνικό δίκτυο πρέπει να έχει ως έξοδο ένα διάνυσμα τιμών μεγέθους k, όπου k ο αριθμός των κλάσεων (για προβλήματα με δύο κλάσεις το k μπορεί να είναι 1), και οι τιμές να είναι στο διάστημα 0 έως 1.

2.1.6 Αλγόριθμοι βελτιστοποίησης

Η βελτιστοποίηση είναι η διαδικασία της ελαχιστοποίησης ή μεγιστοποίησης κάποιας συνάρτησης $f(x)$ μεταβάλλοντας το x. Συγκεκριμένα στην περίπτωση των νευρωνικών δικτύων η μάθηση είναι ένα πρόβλημα βελτιστοποίησης όπου γίνεται αναζήτηση των παραμέτρων θ, τέτοιων ώστε να μειωθεί η συνάρτηση σφάλματος $J(\theta)$.

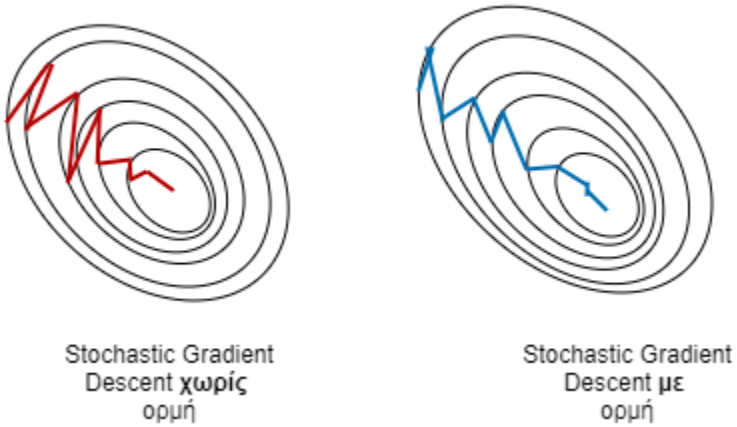
Οι Αλγόριθμοι βελτιστοποίησης που χρησιμοποιούνται στα νευρωνικά διαφέρουν από αυτούς που χρησιμοποιούνται παραδοσιακά. Πολλές φορές στα νευρωνικά δίκτυα χρησιμοποιούνται συναρτήσεις σφάλματος που έχουν πολλαπλά τοπικά ελάχιστα, έτσι η σύγκλιση σε ολικό ελάχιστο καθίσταται δυσεπίτευκτη. Ακόμα και αν ο στόχος ήταν η εύρεση ολικού ελαχίστου, η διαδικασία της μάθησης θα είχε πολύ μεγαλύτερο κόστος χρόνου και υπάρχει η περίπτωση να μην βρεθεί ποτέ το ολικό ελάχιστο. Για αυτόν τον λόγο γίνεται συμβιβασμός και αποδοχή των τιμών που μειώνουν σημαντικά την συνάρτηση σφάλματος. Με αυτόν τον τρόπο το δίκτυο εκπαιδεύεται πιο γρήγορα και πιθανότατα να έχει σφάλμα που συγκλίνει στο βέλτιστο αλλά δεν είναι το βέλτιστο.



Εικόνα 2-10: Το πρόβλημα τοπικού και ολικού ελαχίστου

Ένας αλγόριθμος βελτιστοποίησης που χρησιμοποιήθηκε στις πρώτες εφαρμογές νευρωνικών δικτύων είναι ο Gradient Descent. Στο κεφάλαιο 2.1.4 περιεγράφηκε η βασική λειτουργία του Gradient Descent στην τροποποίηση των βαρών του δικτύου χωρίς να αναφερθεί. Αναλυτικά, ο αλγόριθμος στο σύνολο των δεδομένων εκπαίδευσης υπολογίζει της κλίση του σφάλματος εν συνάρτηση των βαρών, χρησιμοποιώντας τις μερικές παραγώγους της συνάρτησης σφάλματος και των βαρών, και τροποποιεί τα βάρη με κατεύθυνση αντίθετη αυτής του σφάλματος. Βασικοί παράμετροι του αλγορίθμου είναι ο ρυθμός μάθησης d και όταν χρησιμοποιείτε ορμή η σταθερά mc που την ορίζει και παίρνει τιμές από 0 έως 1. Η ορμή είναι η επίδραση της προηγούμενης μεταβολής ενός βάρους στην επόμενη. Η χρήση της ορμής βοηθά στην αποφυγή plateau ή saddle points και των τοπικών ελαχίστων που δεν αποδίδουν το επιθυμητό σφάλμα. Επιπλέον η ορμή βοηθάει στην ταχύτητα σύγκλισης. Με χρήση ορμής ο τύπος μεταβολή των βαρών τροποποιείται ως εξής :

$$\Delta_{w_{j,i}}(t+1) = -(1-mc)d \cdot \delta_i \cdot a_j + mc\,\Delta_{w_{j,i}}(t)$$

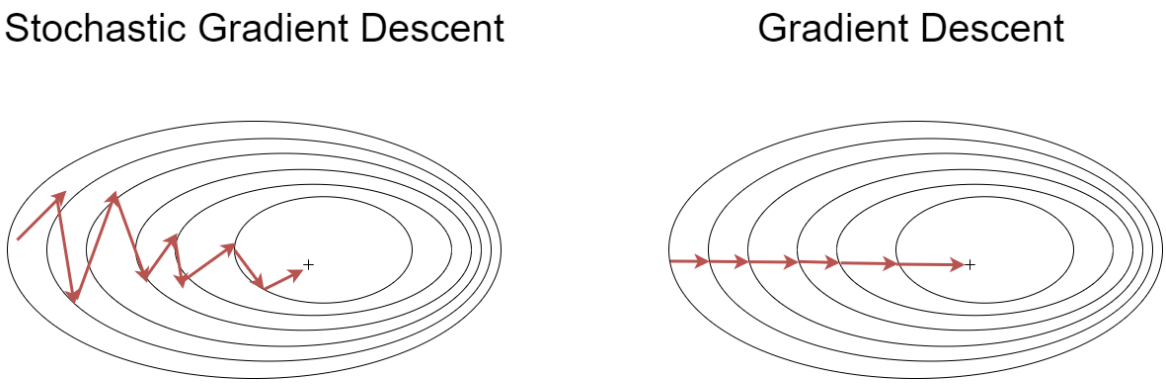


Εικόνα 2-11: Παράδειγμα Stochastic Gradient Descent με και χωρίς ορμή

Συχνά γίνεται και χρήση μεταβλητού ρυθμού μάθησης για ταχύτερη σύγκλιση, επειδή όταν ο αλγόριθμος πλησιάζει σε τοπικό ελάχιστο θέλουμε μικρό ρυθμό μάθησης για να το βρει με κάποια ακρίβεια ενώ όταν είναι στην αρχή θέλουμε μεγάλο ρυθμό μάθησης για να φτάσει γρήγορα κοντά σε τοπικό ελάχιστο.

Εικόνα 2-12: Η διαφορά ανόμοιων ρυθμών μάθησης

Μια διάσημη παραλλαγή του Gradient Descent είναι ο Stochastic Gradient Descent (SGD). Η διαφορά τους είναι πως στον SGD υπολογίζεται η κλίση του σφάλματος και μεταβάλλονται τα βάρη μετά από την τροφοδότηση κάθε παραδείγματος εκπαίδευσης στο δίκτυο. Με αυτόν τον τρόπο έχουμε περισσότερες μεταβολές των βαρών, αυτό βοηθάει το δίκτυο να γενικεύει καλά. Ένα από τα πλεονεκτήματα του SGD είναι η εύκολη ανάπτυξή του και η γρήγορη επεξεργασία ακόμα και για μεγάλο σύνολο δεδομένων. Επιπλέον ο αλγόριθμος διαχειρίζεται επιτυχώς τις αλλαγές των βαρών που έχουν θόρυβο.



Εικόνα 2-13: Αριστερά ο αλγόριθμος Stochastic Gradient Descent και δεξιά ο Gradient Descent.

Συνήθως ο SGD χρησιμοποιείται με mini-batches αντί για κάθε παράδειγμα εκπαίδευσης. Με mini-batches στέλνουμε ως είσοδο στο δίκτυο παραπάνω από ένα παράδειγμα εκπαίδευσης αλλά όχι όλα το σύνολο εκπαίδευσης. Για το μέγεθος του mini-batch προτείνετε η χρήση των δυνάμεων του 2, έτσι το μέγεθος μπορεί ναι είναι 16, 32, 64, 128 και ούτω καθεξής. Ο SGD μπορεί να χρησιμοποιηθεί και με ορμή οπού θα αυξήσει την ταχύτητα σύγκλισης.

Nesterov Accelerated Gradient (NAG). Ο αλγόριθμος λειτουργεί ως παράγοντας διόρθωσης της ορμής. Αντί να υπολογίζεται πρώτα η κλίση μετά η ορμή και να γίνεται μεταβολή των βαρών με την συνισταμένη τους, ο αλγόριθμος εφαρμόζει πρώτα την ορμή στα βάρη, μετά υπολογίζει την κλίση στην νέα θέση και στην συνέχεια μεταβάλει τα βάρη.

Adagrad. Είναι ένας αλγόριθμος βελτιστοποίησης βασιζόμενος στον Gradient Descent. Ο αλγόριθμος αναπτύχθηκε για να βοηθήσει στην εύρεση του σωστού ρυθμού μάθησης. Η λειτουργία του είναι η προσαρμογή του ρυθμού μάθησης έτσι ώστε τα βάρη που τροποποιούνται πιο συχνά να έχουν μικρό ρυθμό μάθησης, ενώ τα βάρη που τροποποιούνται σπάνια να έχουν μεγαλύτερο ρυθμό μάθησης. Ο αλγόριθμος βασίζεται στο άθροισμα τετραγώνων όλων των προηγούμενων κλίσεων της συνάρτησης σφάλματος ως προς κάθε παράμετρο [5]. Το βασικότερο μειονέκτημα του είναι ότι ο ρυθμός μάθησης μπορεί να γίνει υπερβολικά μικρός και να μην μπορεί το δίκτυο να εκπαιδευτεί περεταίρω.

Adadelta. Επέκταση του Adagrad όπου προσπαθεί να αποφύγει την υπερβολική μείωση του ρυθμού μάθησης εφαρμόζοντας ορμή στα τετράγωνα των κλίσεων [6]. Παρόμοιος αλγόριθμός είναι και ο RMSProp.

Adam ή Adaptive moment estimation. Είναι ένας συνδυασμός της ορμής και των αλγορίθμων Adadelta/RMSProp. Αντί να προσαρμόζει τα βάρη βασιζόμενος μόνο στον μεταβλητό ρυθμό μάθησης, ο αλγόριθμος λαμβάνει υπόψη και υπολογίζει την ορμή των ίδιων των κλίσεων [7]. Ο Adam είναι ο πιο διάσημος αλγόριθμος βελτιστοποίησης στον πεδίο της βαθιάς μάθησης, επειδή μπορεί να πετύχει καλή απόδοση χωρίς μεγάλο κόστος χρόνου.

2.2 Βαθιά Μάθηση

Οι τυπικές τεχνικές μηχανικής μάθησης πολλές φορές έχουν περιορισμένες ικανότητες για την επεξεργασία φυσικών δεδομένων στην απλή τους μορφή. Το κύριο μειονέκτημα τους είναι η εξάρτηση τους από τον άνθρωπο και την δυνατότητα του να εξάγει χαρακτηριστικά. Τα χαρακτηριστικά που εξάγονται από άνθρωπο συχνά είναι εξαιρετικά συγκεκριμένα και είναι ελλιπή λόγω λανθάνουσας μορφής που δεν εντοπίζεται ακόμα και από τους ειδήμονες σε θέματα από τα οποία αντλούνται τα χαρακτηριστικά [8].

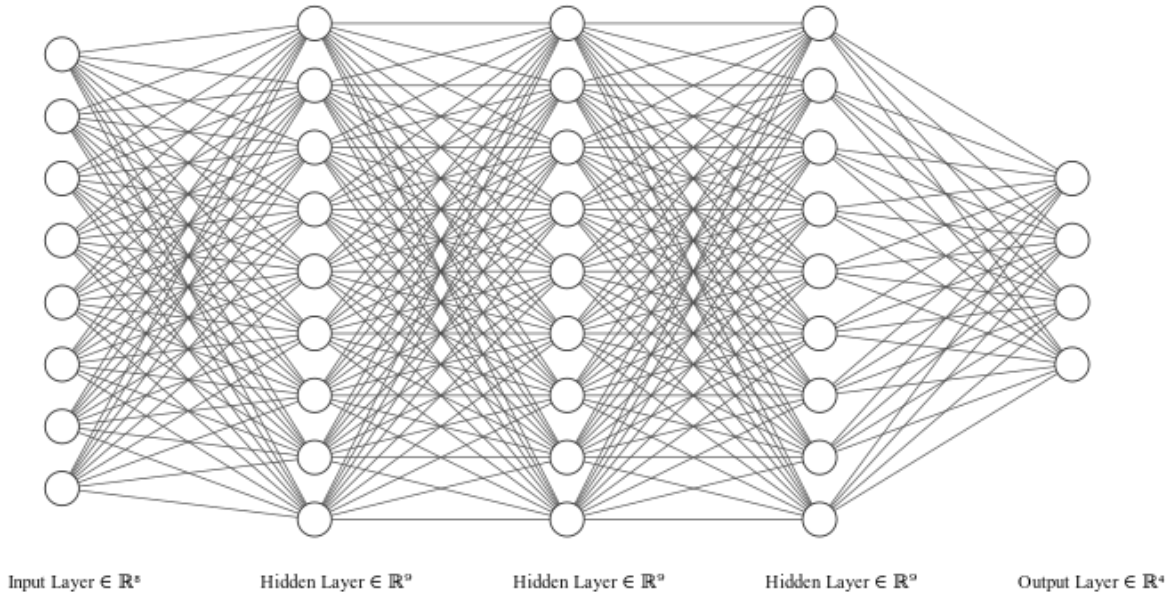
Η εκμάθηση αναπαράστασης είναι ένα σύνολο συστημάτων που δίνει την ικανότητα σε ένα μοντέλο να τροφοδοτείτε με δεδομένα στην απλή τους μορφή και να ανακαλύπτει αυτόματα τους συμβολισμούς και τις αναπαραστάσεις των δεδομένων. Τα νευρωνικά δίκτυα βαθιάς μάθησης είναι συστήματα εκμάθησης αναπαράστασης με πολλαπλά επίπεδα αναπαράστασης καθένα εξυπηρετώντας έναν σκοπό. Ξεκινώντας με την απλή είσοδο των δεδομένων και περνώντας από ένα επίπεδο αναπαράστασης σε ένα άλλο ελαφρώς πιο αφηρημένο επίπεδο και με την σύνθεση αρκετών τέτοιων μετασχηματισμών δίνετε η δυνατότητα στο δίκτυο να εκπαιδευτεί σε λειτουργίες που μέχρι σήμερα θεωρούνταν περίπλοκες. Για προβλήματα ταξινόμησης, τα υψηλότερα επίπεδα αναπαράστασης ενισχύουν τις πτυχές της εισόδου που είναι σημαντικές για την διάκριση των δεδομένων και καταστολή άσχετων παραλλαγών. Η βασική πτυχή της βαθιάς μάθησης είναι ότι αυτά τα επίπεδα χαρακτηριστικών δεν έχουν σχεδιαστεί από ανθρώπους, αλλά μαθαίνονται από δεδομένα χρησιμοποιώντας μια διαδικασία μάθησης γενικής χρήσης [9].

Η ραγδαία εξάπλωση των εφαρμογών την βαθιάς μάθησης την τελευταία δεκαετία οφείλεται στους ακόλουθους παράγοντες:

- Την αύξηση των διαθέσιμων δεδομένων.

- Την πρόοδο σε πολυπύρηνους επεξεργαστές και GPU που διευκολύνει τη βαθιά μάθηση.
- Την ανάπτυξη νέων μοντέλων και αλγορίθμων.

Τα βαθιά νευρωνικά δίκτυα έχουν επαληθεύσει την δυναμικότητα τους και την προσοχή που στρέφεται σε αυτά κερδίζοντας διαγωνισμούς όπως ο ImageNet 2012.



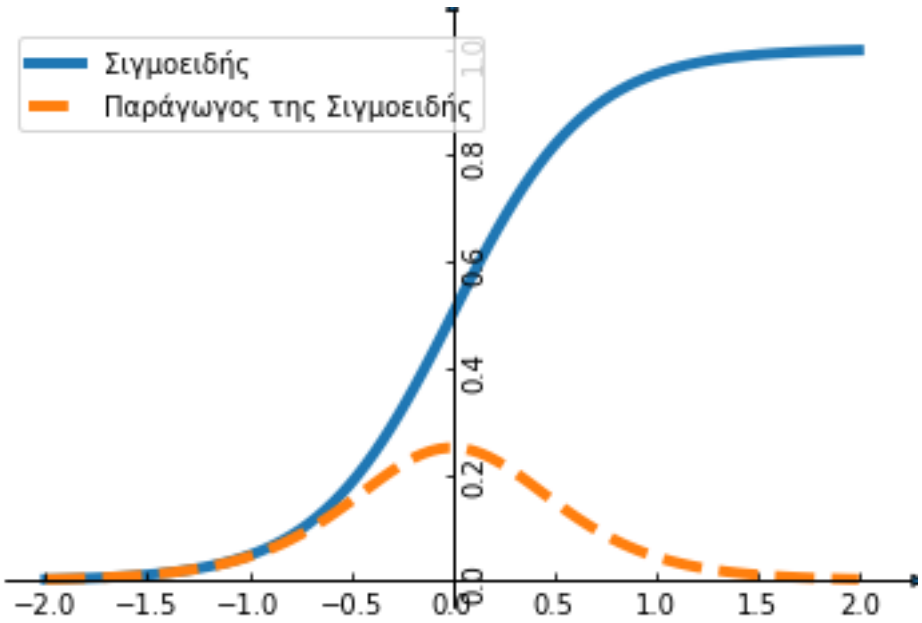
Εικόνα 2-14: Βαθύ Νευρωνικό δίκτυο με 3 κρυφά επίπεδα

Τα βαθιά νευρωνικά δίκτυα χαρακτηρίζονται από την πολυπλοκότητα τους. Συγκεκριμένα τα βαθιά δίκτυα διαφοροποιούνται από τα δίκτυα με την συνήθη αρχιτεκτονική στα εξής χαρακτηριστικά:

- Περισσότεροι νευρώνες και περισσότερα κρυφά επίπεδα από τα "απλά" δίκτυα.
- Πιο περίπλοκοι τρόποι σύνδεσης επιπέδων.
- Χρήση περισσότερης υπολογιστικής δύναμης.
- Αυτόματη εξαγωγή χαρακτηριστικών

Η αύξηση των νευρώνων εκφράζει την δυνατότητα που έχουν τα βαθιά δίκτυα στην ανάπτυξη περίπλοκων μοντέλων. Επιπλέον τα επίπεδα και ο τρόπος σύνδεσης τους έχουν εξελιχθεί από απλά πλήρως συνδεδεμένα σε τοπικά συνδεδεμένα patches νευρώνων μεταξύ επίπεδων, στα Συνελικτικά Νευρωνικά Δίκτυα (CNN), και σε επαναληπτικές συνδέσεις στον ίδιο νευρώνα (εκτός από τις συνδέσεις από το προηγούμενο επίπεδο), στα Επαναληπτικά Νευρωνικά Δίκτυα (RNN). Περισσότερες συνδέσεις σημαίνει περισσότερα βάρη και παράμετροι που πρέπει να βελτιστοποιηθούν και αυτό απαιτεί την χρήση αρκετής υπολογιστικής δύναμης. Όλες αυτές οι διαφοροποιήσεις επιτρέπουν στο δίκτυο να εκπαιδευτεί σε περίπλοκες διαδικασίες όπως Image recognition, Text Summarization, Machine Translation, Text classification και σε πολλές άλλες.

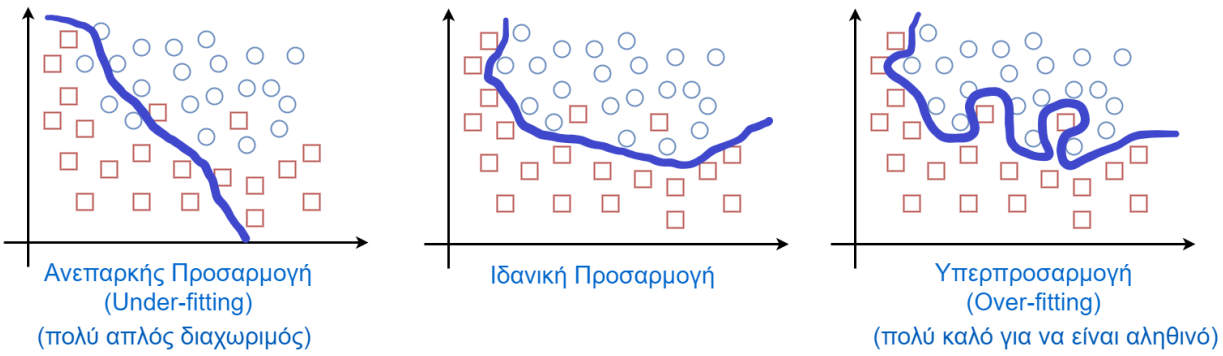
Ένα κοινό πρόβλημα των νευρωνικών δικτύων βαθιάς μάθησης είναι οι φθίνουσες κλίσεις (vanishing gradients). Όσο περισσότερα επίπεδα προστίθενται στο δίκτυο τόσο πιθανό είναι οι παράγωγοι του σφάλματος ως προς τα βάρη για τα βάρη των αρχικών επιπέδων να τείνουν στο μηδέν, με αποτέλεσμα το δίκτυο να μην εκπαιδεύεται αποδοτικά. Αυτό το πρόβλημα συναντάται συχνά στους νευρώνες που έχουν σιγμοειδή συνάρτηση ενεργοποίησης, επειδή περιορίζει έναν μεγάλο διάστημα εισόδου σε ένα μικρό διάστημα μεταξύ των τιμών 0 και 1. Ως εκ τούτου, μια μεγάλη αλλαγή στην είσοδο της σιγμοειδής συνάρτησης θα προκαλέσει μια μικρή αλλαγή στην έξοδο και ως επακόλουθο η παράγωγος είναι μικρή. Πολλαπλασιάζοντας μικρές παραγώγους επιφέρει εκθετική μείωση στις κλίσεις καθώς πραγματοποιείται η οπισθοδιάδοση σφάλματος από το επίπεδο εξόδου στα αρχικά επίπεδα. Επειδή τα αρχικά επίπεδα είναι υπεύθυνα για την εκμάθηση αναπαράστασης από τα δεδομένα εισόδου, η μη αποδοτική εκπαίδευση τους οδηγεί σε σοβαρή αναποτελεσματικότητα του δικτύου.



Εικόνα 2-15: Η σιγμοειδής συνάρτηση και η παράγωγός της

Ενίοτε μπορεί να προκύψει και το πρόβλημα των αυξόντων κλίσεων (exploding gradients), όπου παρατηρούνται πολύ μεγάλες παράγωγοι στα πρώτα επίπεδα και μικρότερες στα τελευταία, εισάγοντας αστάθεια στην εκπαίδευση του δικτύου. Μια από τις πιο απλές λύσεις είναι η χρήση μια διαφορετικής συνάρτησης ενεργοποίησης, συνήθως, όταν παρατηρούνται τέτοιου είδους προβλήματα, προτείνετε η χρήση ReLU αντί της σιγμοειδής.

Η πολυπλοκότητα ενός νευρωνικού δικτύου βαθιάς μάθησης, αν και βοηθάει το δίκτυο στην εκπαίδευση περίπλοκων συναρτήσεων, συχνά προκαλεί το φαινόμενο της υπερπροσαρμογής (overfitting). Υπερπροσαρμογή έχουμε όταν το συνολικό σφάλμα για τα παραδείγματα εκπαίδευσης γίνεται πολύ μικρό, αλλά όταν τροφοδοτούμε το δίκτυο με νέα δεδομένα παρατηρείτε μεγάλη αύξηση στο συνολικό σφάλμα. Διαφορετικά λέμε ότι το δίκτυο δεν γενικεύει καλά. Το αντίθετο φαινόμενο της υπερπροσαρμογής είναι η ανεπαρκής προσαρμογή (underfitting), όπου το συνολικό σφάλμα είναι μεγάλο για το παραδείγματα εκπαίδευσης και κατά πάσα πιθανότητα το συνολικό σφάλμα είναι μεγάλο και για τα νέα παραδείγματα που τροφοδοτούνται στο δίκτυο.



Εικόνα 2-16: Αριστερά το φαινόμενο της ανεπαρκής προσαρμογής (underfitting), στην μέση η σωστή προσαρμογή και δεξιά η υπερπροσαρμογή (overfitting)

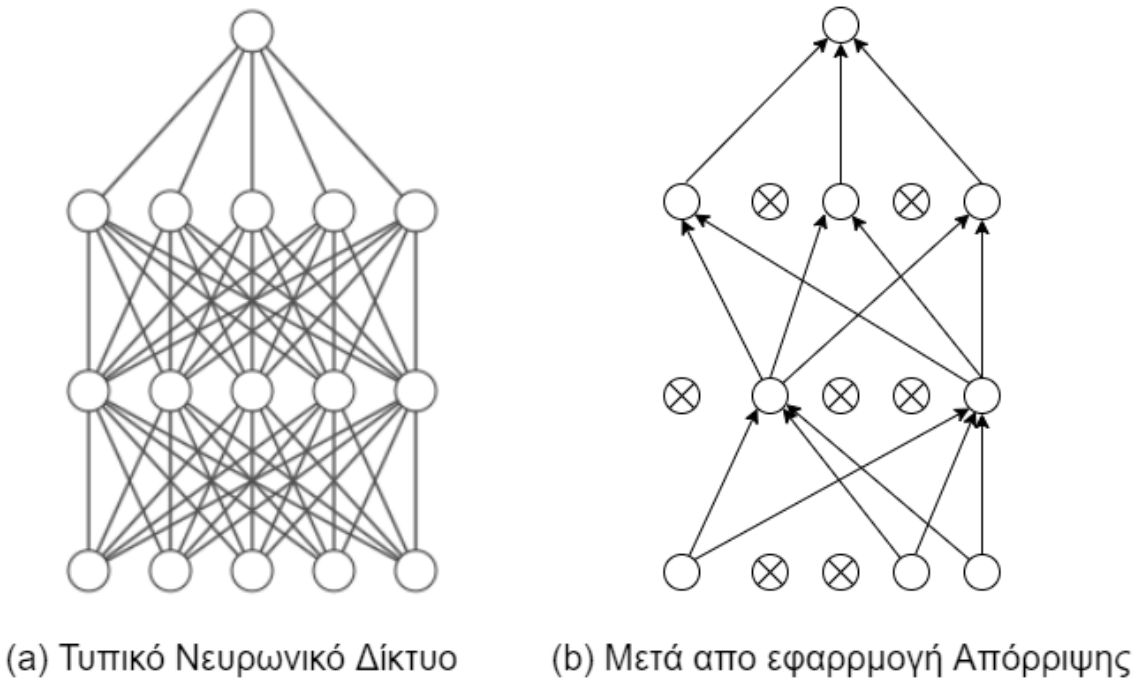
Μια λύση για το πρόβλημα της ανεπαρκής προσαρμογής, που χρησιμοποιείται ευρέως, είναι η ανάπτυξη ενός πιο περίπλοκου δικτύου, προσθέτοντας επίπεδα και χρησιμοποιώντας διαφορετικές συνδέσεις μεταξύ τους. Κατά καιρούς για το πρόβλημα της υπερπροσαρμογής έχουν προταθεί πολλές τεχνικές που το ανιχνεύουν και το αντιμετωπίζουν, παρακάτω περιγράφονται οι πιο διαδεδομένες στην βαθιά μάθηση:

- Πρόωρη διακοπή (Early Stopping). Το διαθέσιμο σύνολο δεδομένων διασπάται σε τρείς ομάδες:
    1. Το σύνολο των παραδειγμάτων εκπαίδευσης (training set)
    2. Το σύνολο των παραδειγμάτων επαλήθευσης (validation set)
    3. Το σύνολο των παραδειγμάτων ελέγχου (test set) – προαιρετικό

  Ένας συνήθης ποσοστιαίος διαχωρισμός είναι 60% για το training set, 20% για το validation set και 20% για το test set. Ο τρόπος διαχωρισμού εξαρτάται κυρίως από το πλήθος των διαθέσιμων δεδομένων, συνήθως για μικρό σύνολο δεδομένων παραλείπεται το σύνολο ελέγχου. Η εκπαίδευση γίνεται μόνο με το σύνολο εκπαίδευσης. Κατά την διάρκεια της εκπαίδευσης παρακολουθείται και το σφάλμα στο σύνολο επαλήθευσης. Εάν κατά την εκπαίδευση προκύψει αύξηση του σφάλματος στο σύνολο επαλήθευσης για συγκεκριμένο πλήθος διαδοχικών εποχών, η εκπαίδευση διακόπτεται και οι παράμετροι παίρνουν τις τιμές που αντιστοιχούσαν στο μικρότερο σφάλμα των δεδομένων επαλήθευσης. Περεταίρω έλεγχος γίνεται χρησιμοποιώντας το σύνολο ελέγχου, αν παρουσιαστεί υψηλό

σφάλμα τα δεδομένα ανακατανέμονται στα τρία σύνολα. Η τεχνική της πρόωρης διακοπής εστιάζει στην ανίχνευση υπερπρορμογής και συχνά συνδυάζεται με άλλες τεχνικές όπως η επόμενη.

- Απόρριψη (Drop out). Η βασική ιδέα είναι η αφαίρεση κάποιων νευρώνων, από το δίκτυο, έτσι ώστε η εκπαίδευση να επιμερίζεται σε όλους τους νευρώνες και να μην βασίζεται μόνο σε μερικούς. Επιπλέον το δίκτυο δεν υπερπροσαρμόζεται στα παραδείγματα εκπαίδευσης και η εκπαίδευση έχει μικρότερο κόστος χρόνου. Η διαδικασία που ακολουθείτε κατά την διάρκεια της εκπαίδευσης είναι η εξής : με πιθανότητα p (dropout rate), που μπορεί να οριστεί, κάποιος νευρώνας αγνοείται από το τρέχον παράδειγμα. Μια καλή τιμή για το p είναι ανάμεσα στο 0.5 και 0.8. Επίσης η απόρριψη προσφέρει και έναν τρόπο για να συνδυαστούν διάφορες αρχιτεκτονικές νευρωνικών δικτύων [10].



Εικόνα 2-17: Η τεχνική Dropout

2.3 Συνελικτικά Νευρωνικά Δίκτυα (CNN)

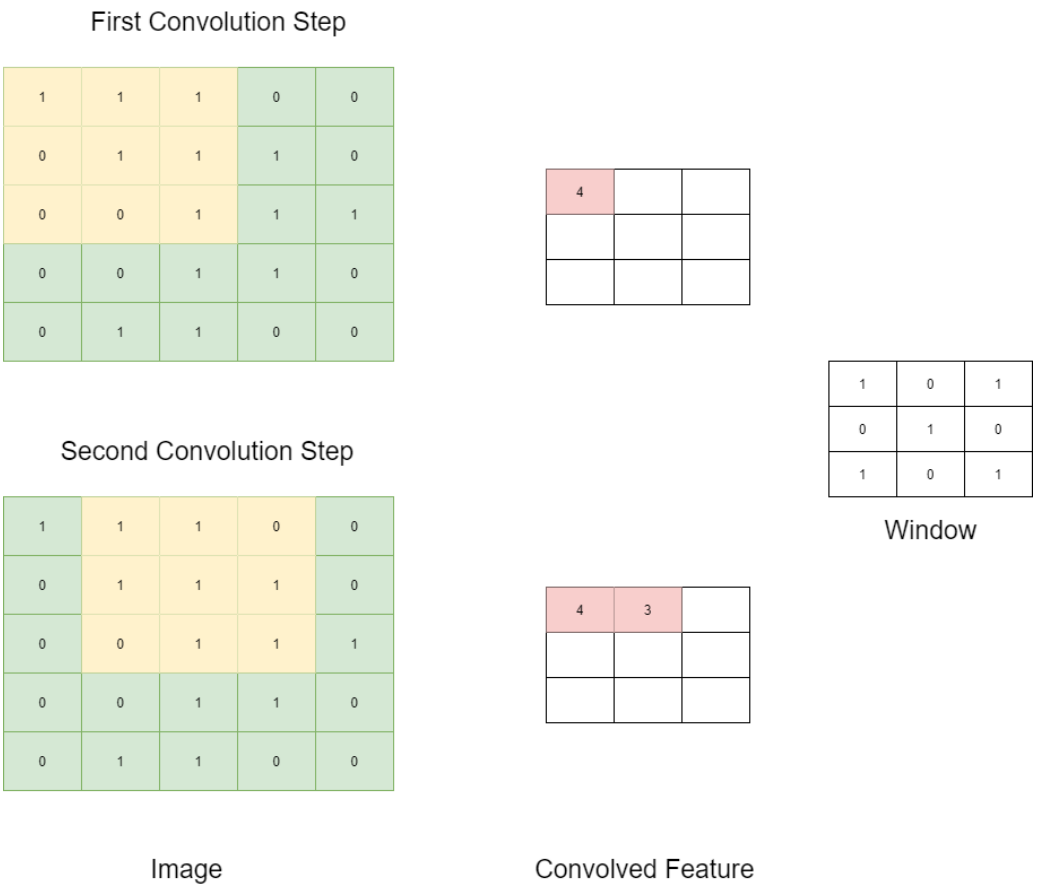
Τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks, ConvNet ή CNN) είναι ένας τύπος Νευρωνικών Δικτύων που χρησιμοποιείται κατά κύριο λόγο στην ταξινόμηση και ανίχνευση εικόνων (image classification and recognition). Την τελευταία δεκαετία παρουσιάζεται μια ιδιαίτερη αύξηση στην χρήση Συνελικτικών Δικτύων, λόγω της αύξησης της διαθέσιμης υπολογιστικής ισχύς με την άφιξη καρτών γραφικών (GPU) με όλο και περισσότερους πυρήνες αλλά και την εφαρμογή διαφόρων τεχνικών παραλληλοποίησης. Απόδειξη της επίδοσης των Συνελικτικών Δικτύων είναι οι επιτυχίες που σημειώνουν σε διαγωνισμούς, όπως ο ImageNet (αναγνώριση αντικειμένων που εμφανίζονται σε κάθε εικόνα, μεταξύ 1000 διαφορετικών αντικειμένων), καθώς και η ευρεία χρήση τους για επίλυση προβλημάτων Υπολογιστικής Όρασης (Computer Vision) από εξέχουσες εταιρίες τεχνολογίας, όπως η Facebook και το αυτοματοποιημένο της σύστημα προσθήκης ετικετών σε φωτογραφίες. Άλλες γνωστές εφαρμογές των Συνελεκτικών Δικτυών είναι :

- Αναγνώριση βίντεο (video recognition)
- Σύστημα συστάσεων (recommender system)
- Ανάλυση ιατρικών εικόνων (medical image analysis)
- Επεξεργασία Φυσικής γλώσσας (natural language processing)
- Χρονοσειρές (time series)

Τα Συνελικτικά Δίκτυα έχουν τέσσερις βασικές λειτουργίες :

1. Βήμα συνέλιξης (Convolution Step)
2. Εφαρμογή Συνάρτησης Ενεργοποίησης
3. Pooling ή Sub-Sampling
4. Ταξινόμηση (Classification)

Η συνέλιξη είναι μια πράξη για συγχώνευση πληροφορίας μεταξύ δύο συνόλων. Ο πιο απλός τρόπος για να καταλάβουμε πως λειτουργεί η συνέλιξη, είναι σκεπτόμενοι ένα κυλιόμενο παράθυρο (sliding window) να εφαρμόζεται σε έναν πίνακα.



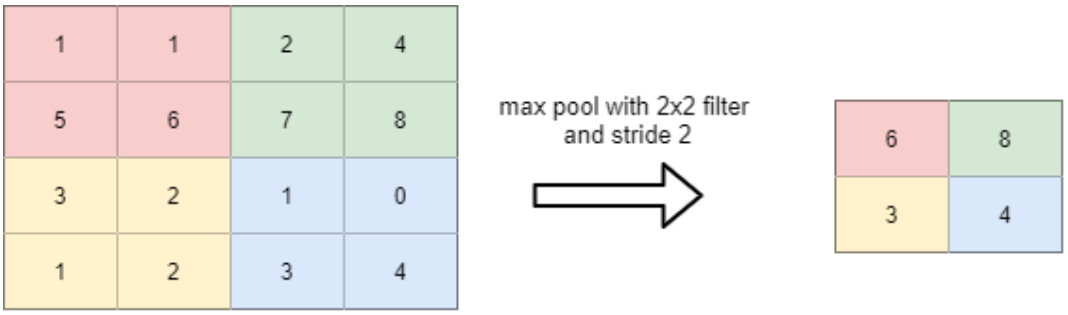
Εικόνα 2-18: Δύο βήματα συνέλιξης

Στην παραπάνω εικόνα η πράξη της συνέλιξης γίνεται πιο ξεκάθαρη. Έστω πως ο πίνακας στα αριστερά είναι μια ασπρόμαυρη εικόνα και κάθε κελί αναπαριστά ένα pixel. Αυτό που κάνουμε στο βήμα της συνέλιξης είναι να πολλαπλασιάσουμε τις τιμές του παραθύρου με τις αντίστοιχες τιμές στον αρχικό πίνακα, να τις αθροίσουμε και το αποτέλεσμα να το αποθηκεύσουμε σε ένα δεύτερο πίνακα, όπου ονομάζεται πίνακας συνέλιξης. Η διαδικασία συνεχίζεται μετακινώντας το παράθυρο μέχρι να περάσει από όλο τον αρχικό πίνακα και να συμπληρωθεί ο πίνακας συνέλιξης. Το κυλιόμενο παράθυρο λέγεται αλλιώς πυρήνας (kernel), φίλτρο (filter) ή ανιχνευτής χαρακτηριστικών (feature detector). Στο παράδειγμα της εικόνας χρησιμοποιούμε ένα φίλτρο 3x3.

Τα Συνελικτικά Δίκτυα είναι πολλά επίπεδα με συνελίξεις με συναρτήσεις ενεργοποίησης μη γραμμικές, όπως η ReLU και η tanh, να εφαρμόζονται στα αποτελέσματα. Αντί για σύνδεση ενός νευρώνα με πολλούς νευρώνες του επόμενου επιπέδου έχουμε τοπικές συνδέσεις οπού μια περιοχή της εισόδου μέσω συνελίξεων συνδέεται με έναν νευρώνα στην έξοδο. Σε κάθε επίπεδο εφαρμόζονται διαφορετικά

φίλτρα, όπου τα αποτελέσματά τους συνδυάζονται. Το Συνελικτικό Δίκτυο αυτόματα μαθαίνει τις τιμές των φίλτρων βάση της εργασίας που του έχει ανατεθεί.

Ανάμεσα σε κάθε συνέλιξη μπορεί να χρησιμοποιηθεί και η πράξη του pooling. Το pooling είναι η δειγματοληψία εφαρμόζοντας τις πράξεις max ή average στον πίνακα συνέλιξης.



Εικόνα 2-19: Παράδειγμα max Pooling

Ένας από τους λόγους που χρησιμοποιείτε το pooling είναι και μείωση διαστάσεων και η παραγωγή πινάκων συγκεκριμένου μεγέθους. Βασικός στόχος του Συνελικτικού Δικτυού είναι να ανιχνεύει στα πρώτα επίπεδα χαμηλού επιπέδου χαρακτηριστικά όπως γραμμές, σχήματα, σκιάσεις και χρησιμοποιώντας αυτά στα επόμενα επίπεδα να ανιχνεύσει υψηλού επιπέδου χαρακτηριστικά. Το τελευταίο επίπεδο χρησιμοποιεί τα υψηλού επιπέδου χαρακτηριστικά για την ταξινόμηση.

Στα Συνελικτικά Δίκτυα υπάρχουν αρκετές επιλογές υπέρ-παραμέτρων, που πρέπει να διαλέξουμε κατά την κατασκευή τους, μερικές από αυτές είναι :

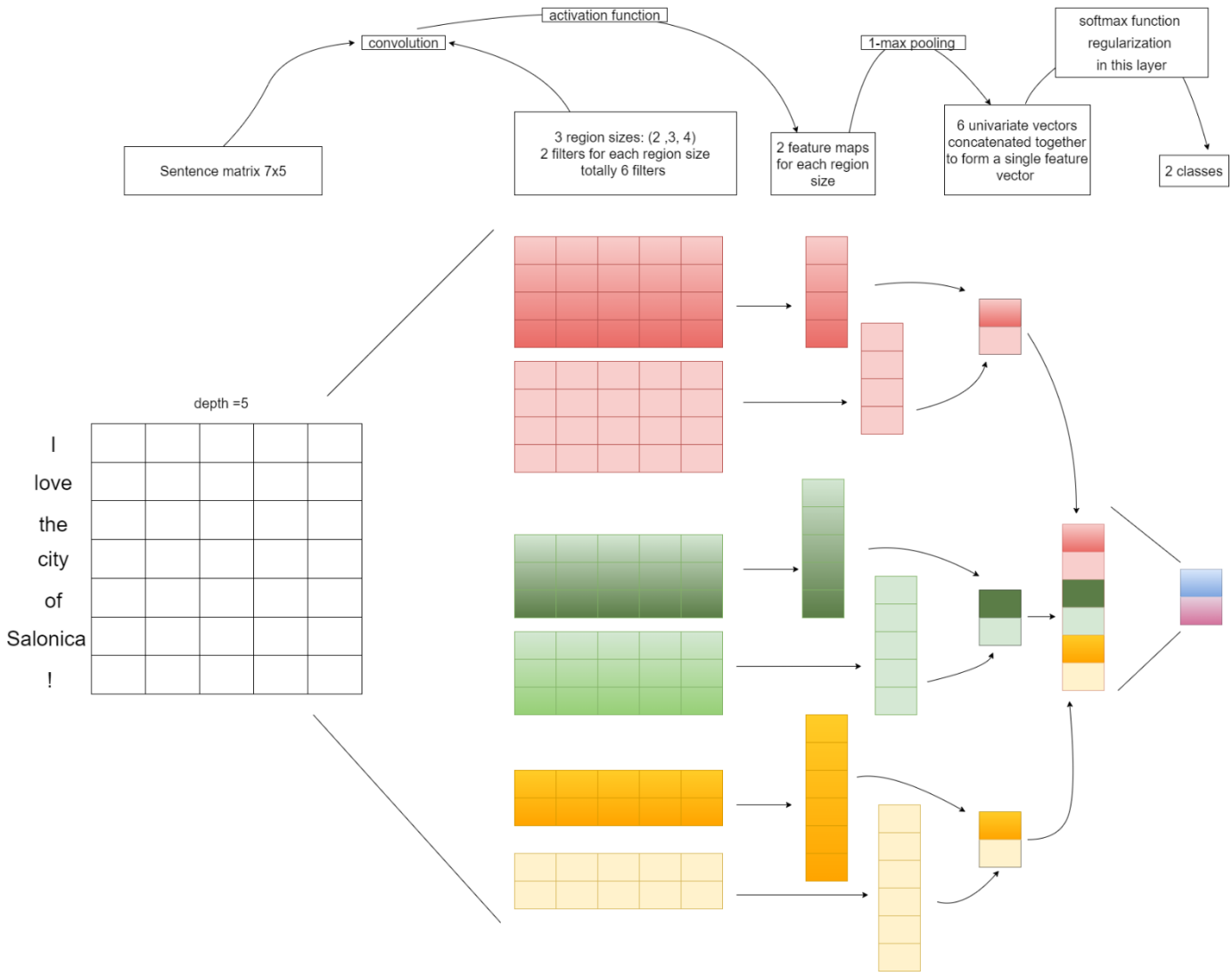
- Στενή ή Ευρεία συνέλιξη (Narrow or Wide Convolution) : Η στενή συνέλιξη χρησιμοποιείται όταν θέλουμε να εφαρμόσουμε το φίλτρο μόνο σε στοιχεία του πίνακα που έχουν γειτονικά στοιχεία όπως το παράδειγμα της εικόνας 2-18. Όταν θέλουμε να εφαρμόσουμε το φίλτρο σε στοιχεία που δεν έχουν γειτονικά στοιχεία χρησιμοποιούμε γέμισμα με μηδενικά (zero-padding) για να καλύψουμε το κενό, αυτή η διαδικασία ονομάζεται ευρεία συνέλιξη.
- Μέγεθος Διασκελισμού (Stride Slide) : Στο παράδειγμα της εικόνας 2-18 χρησιμοποιούσαμε stride slide ίσο με 1, δηλαδή μετακινούσαμε το φίλτρο κατά μια θέση. Μεγαλύτερο μέγεθος διασκελισμού οδηγεί σε λιγότερες

εφαρμογές του φίλτρου που έχει επακόλουθο μικρότερο πίνακα αποτελεσμάτων.

- Κανάλια (Channels) : Η έννοια των καναλιών αναφέρεται σε διαφορετικές όψεις των δεδομένων εισόδου. Για παράδειγμα στις εικόνες υπάρχουν τα RGB κανάλια (κόκκινο, πράσινο, μπλε). Μπορούν να εφαρμοστούν συνελίξεις μεταξύ καναλιών. Σε γλωσσικά δεδομένα διαφορετικά κανάλια μπορεί να σημάνει διαφορετικές ενσωματώσεις (embeddings) μιας λέξεις ανά κανάλι ή μεταφράσεις σε άλλες γλώσσες ανά κανάλι.

2.3.1 Text CNN

Για γλωσσικά δεδομένα και για προβλήματα επεξεργασίας φυσικής γλώσσας (natural language processing) η είσοδος, αντί για pixels, είναι προτάσεις ή έγγραφα εκφρασμένα ως ένας πίνακας. Κάθε σειρά του πίνακα αντιστοιχεί σε μια λεκτική μονάδα (token), που στην ουσία είναι μια λέξη. Κάθε λέξη αναπαριστάτε ως ένα διάνυσμα τιμών, που συνήθως είναι embeddings. Έτσι, λοιπόν, για μια πρόταση με εφτά λέξεις ή λεκτικές μονάδες και χρησιμοποιώντας embeddings 300 διαστάσεων παίρνουμε έναν πίνακα 7x300 όπου είναι η είσοδος μας στο Συνελικτικό Δίκτυο. Επιπλέον επειδή κάθε σειρά είναι και μια λέξη χρησιμοποιούνται φίλτρα όπου μετακινούνται πάνω από ολόκληρες λέξεις, για αυτόν τον λόγο το πλάτος του φίλτρου ισούται με το πλάτος του πίνακα εισόδου. Το ύψος του φίλτρου μπορεί να διαφέρει ανάλογα με το σε πόσες λέξεις θέλουμε να εφαρμόζεται κάθε φορά, τυπικές τιμές είναι από δύο μέχρι πέντε.



Εικόνα 2-20: Παράδειγμα αρχιτεκτονικής CNN για ταξινόμηση πρότασης (sentence classification)

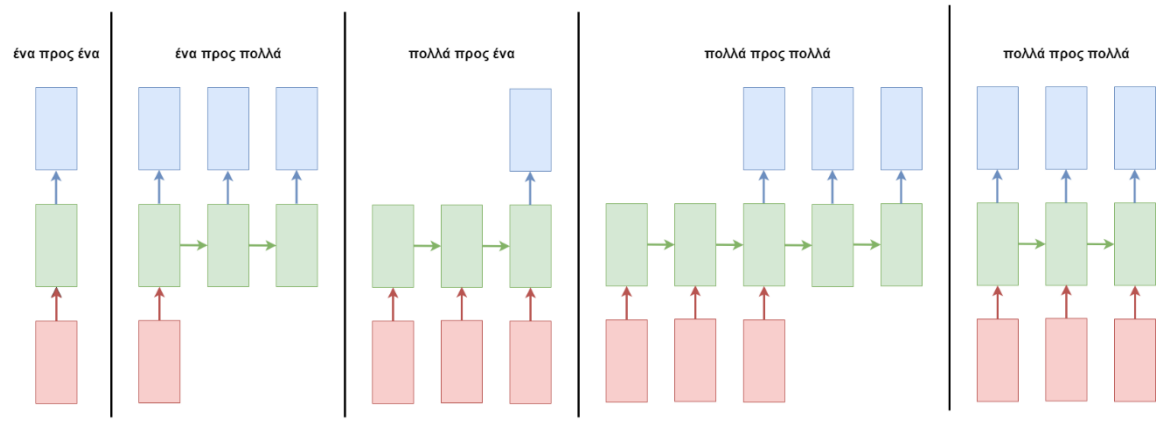
Τα Συνελικτικά Δίκτυα έχουν ικανοποιητική απόδοση σε συγκεκριμένα προβλήματα επεξεργασίας φυσικής γλώσσας όπως είναι η ανάλυση συναισθημάτων (sentiment analysis), ανίχνευση ανεπιθύμητων μηνυμάτων (spam detection), ταξινόμηση θέματος (topic classification) και γενικά ότι έχει να κάνει με ταξινόμηση κειμένου (text classification). Η αρχιτεκτονική που παρουσιάζεται στην εικόνα 2-20 αναπαριστά ένα βασικό μοντέλο για ταξινόμηση κειμένου και έχει δείξει πως έχει καλύτερα αποτελέσματα η χρήση max pooling αντί του average pooling [11]. Άλλες αρχιτεκτονικές κάνουν χρήση δύο καναλιών με διαφορετικά embeddings κρατώντας το ένα κανάλι σταθερό και στο άλλο να προσαρμόζονται τα embeddings κατά την διάρκεια της μάθησης [12]. Επίσης είναι δυνατή η κατασκευή Συνελικτικού Δικτύου το όποιο χρησιμοποιεί ως είσοδο χαρακτήρες αντί για λέξεις. Η χρήση του προαναφερόμενου δικτύου προτείνετε όταν έχουμε πολύ

μεγάλα σύνολα δεδομένων, ενώ σε μικρότερα σύνολα δεδομένων παρουσιάζει μειωμένη απόδοση από πιο απλά μοντέλα [13].

Τέλος το μεγαλύτερο πρόβλημα που παρουσιάζεται στην χρήση Συνελικτικών Δικτύων είναι η αδυναμία τους να ανιχνεύσουν συσχετίσεις λέξεων που βρίσκονται μακριά. Η βασική αρχή με την οποία λειτουργούν τα Συνελικτικά Δίκτυα είναι πως στοιχεία που βρίσκονται κοντά, όπως τα pixels, είναι και σημασιολογικά σχετιζόμενα, αυτό όμως δεν συμβαίνει πάντα όταν ως είσοδο έχουμε κείμενο και λέξεις. Πράγματι διαισθητικά έχει περισσότερο νόημα η χρήση Επαναληπτικών Νευρωνικών Δικτύων (RNN), που θα δούμε στην επόμενη ενότητα, παρά Συνελικτικών. Παρόλα αυτά η μάθηση το Συνελικτικών Δικτύων εφαρμόζεται ταχύτατα, ειδικά με χρήση κάρτας γραφικών (GPU), και τα αποτελέσματα είναι ικανοποιητικά. Για τον παραπάνω λόγο τα Συνελικτικά Δίκτυα χρησιμοποιούνται σε αρκετές περιπτώσεις για την επίλυση προβλημάτων επεξεργασίας φυσική γλώσσας.

2.4 Επαναληπτικά Νευρωνικά Δίκτυα (RNN)

Τα Επαναληπτικά Νευρωνικά Δίκτυα (Recurrent Neural Networks, RNN) ανήκουν στην οικογένεια των δικτύων με ανατροφοδότηση (feedback ή recurrent), δηλαδή υπάρχουν συνδέσεις μεταξύ νευρώνων ενός επιπέδου και νευρώνων προηγούμενου. Μοιάζουν με τα δίκτυα απλής τροφοδότησης (feedforward), αλλά κύρια διαφορά τους είναι η ικανότητα των Επαναληπτικών Δικτύων να αποθηκεύουν και να στέλνουν πληροφορία κατά την πάροδο του χρόνου. Ανέκαθεν η εκπαίδευση των Επαναληπτικών Δικτυών ήταν μια χρονοβόρα διαδικασία η οποία χρειαζόταν αρκετούς υπολογιστικούς πόρους, αλλά πρόσφατες εξελίξεις στην ανάπτυξη καρτών γραφικών (GPUs) με πολλούς πυρήνες και στην παραλληλοποίηση έκαναν την χρήση τους προσιτή προς το ευρύ κοινό των ερευνητών. Τα Επαναληπτικά Δίκτυα λειτουργούν με ακολουθίες διανυσμάτων (sequences of vectors), μπορούν να δεχτούν ακολουθίες ως είσοδο, ως έξοδο ή στις περισσότερες των περιπτώσεων και στα δύο.



Εικόνα 2-21: Παραδείγματα αρχιτεκτονικών Επαναληπτικών Νευρωνικών Δικτύων. Με κόκκινο είναι τα διανύσματα εισόδου, με μπλε είναι τα διανύσματα εξόδου και με πράσινο είναι η κατάσταση του Επαναληπτικού Δικτύου (κρυφά επίπεδα/ο)

Η ικανότητα των Επαναληπτικών Δικτύων διαχείρισης ακολουθιών τους προσθέτει και την δυνατότητα να θυμούνται σε ποια κατάσταση βρίσκονται καθώς και να λαμβάνουν υπόψη προγενέστερες καταστάσεις. Παρατηρώντας την εικόνα 2-21 λαμβάνουμε υπόψη ότι τα Επαναληπτικά Δίκτυα μπορούν να χρησιμοποιηθούν σε μια πληθώρα πεδίων. Συγκεκριμένα πεδία και προβλήματα που χρησιμοποιούν αρχιτεκτονικές σαν την δεύτερη της εικόνας 2-21, δηλαδή θέλουν ως έξοδο μια ακολουθία διανυσμάτων και έχουν ως είσοδο ένα διάνυσμα :

- Τοποθέτηση λεζάντας σε εικόνα (Image captioning).
- Σύνθεση ομιλίας (Speech synthesis).
- Παραγωγή μουσικής (Music generation).
- Μοντέλα παραγωγής κειμένου επιπέδου χαρακτήρα (Character-level text generation models).

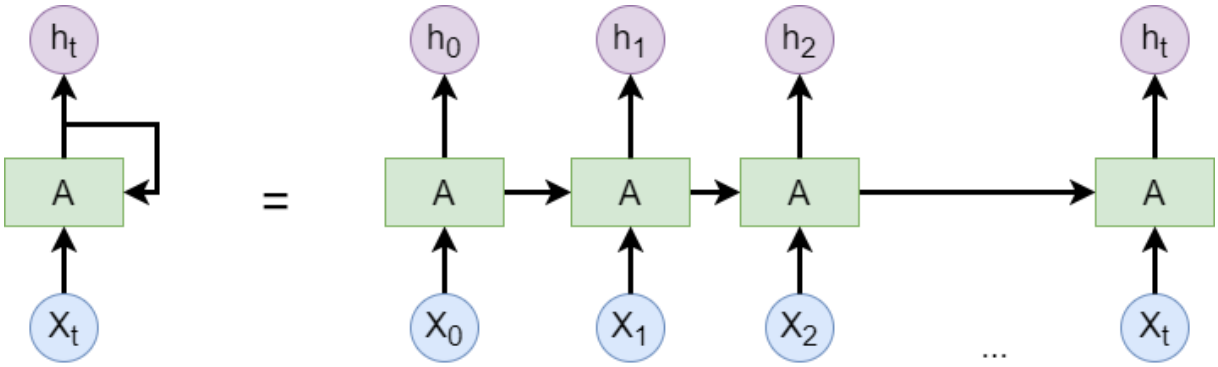
Επιπλέον πεδία και προβλήματα που χρησιμοποιούν αρχιτεκτονικές πολλά προς ένα (many to one) :

- Ταξινόμηση κειμένου (Text classification).
- Ανάλυση συναισθημάτων (Sentiment Analysis).
- Πρόβλεψη χρονοσειρών (Time-series prediction).
- Ανάλυση βίντεο (Video Analysis).
- Ανάκτηση πληροφορίας από Μουσική (Music information retrieval).

Τέλος πεδία και προβλήματα που χρησιμοποιούν αρχιτεκτονικές πολλά προς πολλά (many to many), είτε με την πρώτη παραλλαγή είτε με την δεύτερη :

- Μετάφραση φυσικής γλώσσας (Natural language translation).
- Περίληψη φυσικής γλώσσας (Natural language summarization).
- Συμμετοχή σε διάλογο (chatbot).
- Ρομποτικός έλεγχος (Robotic control).

Ουσιαστικά τα Επαναληπτικά Νευρωνικά Δίκτυα είναι Νευρωνικά Δίκτυα με βρόγχους, οι οποίοι βρόγχοι βοηθάνε στην διατήρηση πληροφορίας του Νευρωνικού Δικτύου. Έτσι, λοιπόν, για την πρόβλεψη δεδομένου της τρέχουσας εισόδου λαμβάνει υπόψη και την έξοδο που έμαθε από προηγούμενες εισόδους. Εννοείτε πως οι είσοδοι σχετίζονται μεταξύ τους.



Εικόνα 2-22: Εκτύλιξη (unroll ή unfold) ενός Επαναληπτικού Νευρωνικού Δικτύου

Το δίκτυο της εικόνας 2-22 αρχικά παίρνει ως διάνυσμα εισόδου X_0 και παράγει h_0, η οποία κρυφή κατάσταση σε συνδυασμό με το διάνυσμα εισόδου X_1 αποτελούν είσοδο στο επόμενο βήμα. Παρομοίως η κρυφή κατάσταση h_1 μαζί με το X_2 είναι η επόμενη είσοδος, και αυτό συνεχίζεται μέχρι να τροφοδοτηθεί στο δίκτυο ολόκληρη η ακολουθία των διανυσμάτων εισόδου. Άρα η εξίσωση για τον υπολογισμό του h_t μετά και την εφαρμογή της συνάρτησης ενεργοποίησης διαμορφώνεται ως εξής :

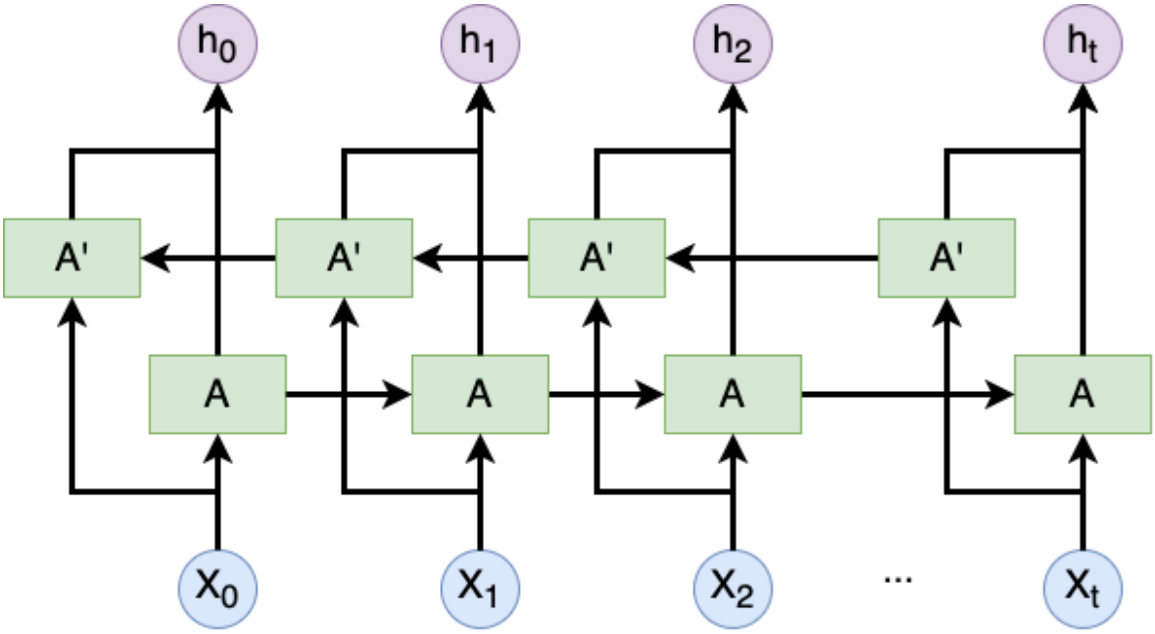
$$h_t = \tanh\left(W_{hh}h_{t-1} + W_{xh}x_t\right)$$

, όπου W_{hh} είναι τα βάρη που εφαρμόζονται στην προηγούμενη κρυφή κατάσταση, W_{xh} είναι τα βάρη που εφαρμόζονται στην τρέχουσα είσοδο, tanh είναι η συνάρτηση ενεργοποίησης.

Τέλος το διάνυσμα εξόδου του δικτύου προκύπτει από:

$$y_t = W_{hy} h_t$$

Πολλές φορές υπάρχει η ανάγκη για διατήρηση πληροφορίας προερχόμενη και από τις δύο πλευρές μιας ακολουθίας. Για αυτό τον λόγο γίνεται χρήση αμφίδρομων Επαναληπτικών Νευρωνικών Δικτύων (Bidirectional Recurrent Neural Networks, BRNN), τα οποία στην ουσία είναι δύο ξεχωριστά Επαναληπτικά Δίκτυα. Η ακολουθία εισόδου τροφοδοτείτε στο ένα δίκτυο με την φυσική χρονική σειρά, ενώ στο άλλο με την αντίθετη χρονική σειρά. Οι έξοδοι των δυο δικτύων συνενώνονται σε κάθε χρονικό βήμα, ωστόσο υπάρχουν και άλλες επιλογές όπως η άθροιση.

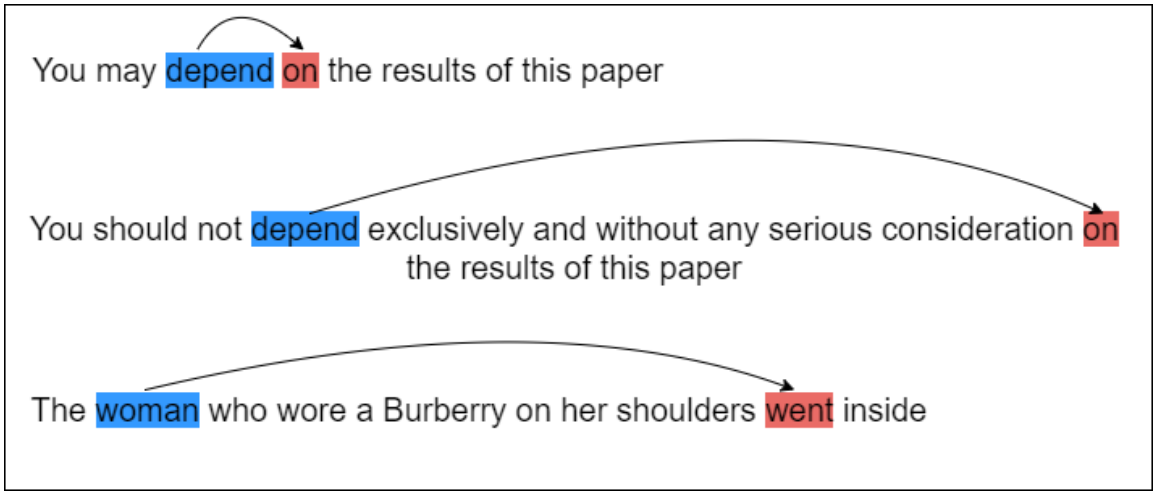


Εικόνα 2-23: Γενική δομή ενός Αμφίδρομου Επαναληπτικού Δικτύου (BRNN)

2.4.1 Long Short-Term Memory (LSTM)

Ως θεωρία τα Επαναληπτικά Νευρωνικά Δίκτυα φαίνονται απλά και ισχυρά μοντέλα αλλά στην πράξη αντιμετωπίζουν αρκετές επιπλοκές που τα εμποδίζουν να εκπαιδευτούν καταλλήλως. Μια από αυτές τις επιπλοκές είναι το πρόβλημα φθινουσών και αυξόντων κλίσεων (vanishing and exploding gradients) [14]. Όπως αναφέρθηκε και σε προηγούμενη ενότητα το πρόβλημα αυξόντων κλίσεων αναφέρεται σε μεγάλη αύξηση των κλίσεων που μπορεί να προκληθεί λόγω της έκρηξης των μακροπρόθεσμων συνιστωσών, που μπορούν να αυξηθούν εκθετικά περισσότερο των βραχυπρόθεσμων. Το πρόβλημα των φθινουσών κλίσεων περιγράφει την αντίθετη συμπεριφορά, δηλαδή οι

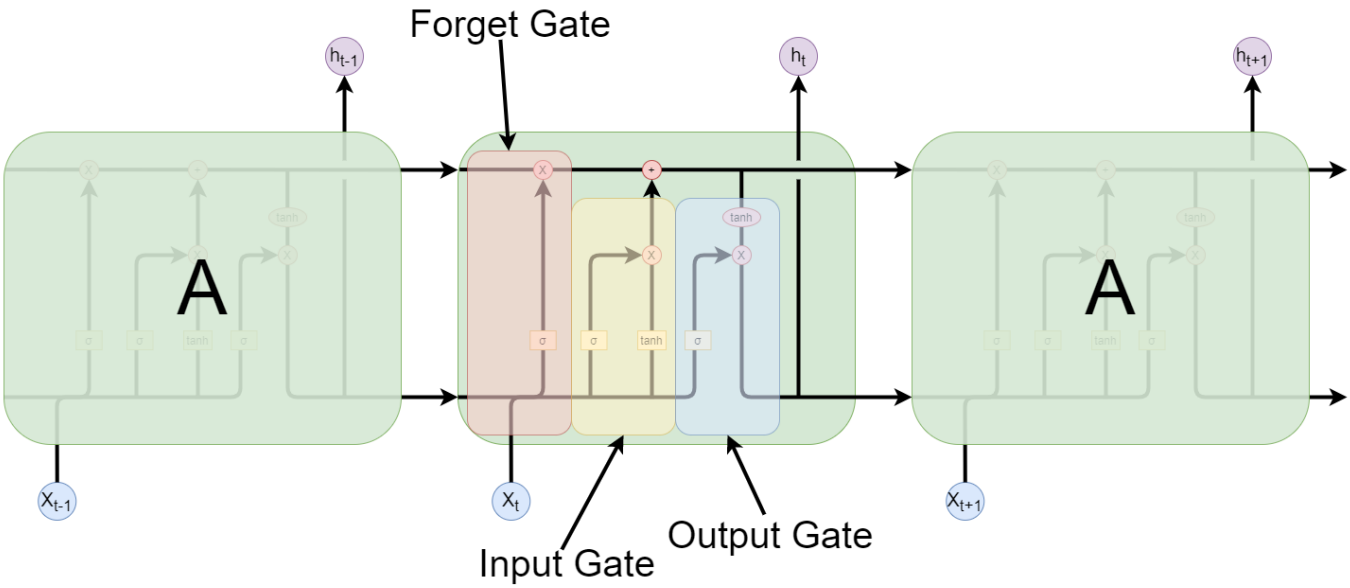
μακροπρόθεσμες συνιστώσες προσεγγίζουν εκθετικά το μηδέν, έχοντας σαν επίπτωση το μοντέλο να μην μπορεί να μάθει την συσχέτιση μεταξύ απομακρυσμένων συμβάντων [15].



Εικόνα 2-24: Παραδείγματα μακροπρόθεσμων εξαρτήσεων σε γλωσσικά δεδομένα

Στην εικόνα 2-24 μπορούμε να δούμε πως το πρόβλημα των μακροπρόθεσμων εξαρτήσεων εκφράζεται όταν έχουμε ως είσοδο μια πρόταση. Στην εικόνα βλέπουμε πως για την ίδια φράση "depend on" μπορεί το μήκος της εξάρτησης να ποικίλει. Επιπροσθέτως το νόημα μιας πρότασης πολλές φορές προσδιορίζεται από λέξεις που δεν είναι κοντά, για παράδειγμα, στην πρόταση "The woman who wore a Burberry on her shoulders went inside". Αυτή η πρόταση περιγράφει στην πραγματικότητα περιγράφει την κίνηση μιας γυναίκας και όχι ενός αντικειμένου. Χρησιμοποιώντας ένα απλό Επαναληπτικό Νευρωνικό Δίκτυο είναι δύσκολο μάθει μια τέτοια πληροφορία και να καταλάβει την σημασιολογία της πρότασης [16].

Για την αντιμετώπιση του προβλήματος των μακροπρόθεσμων εξαρτήσεων προτείνεται το μοντέλο Long Short-Term Memory (LSTM). Τα δίκτυα LSTM είναι μια τροποποιημένη εκδοχή των Επαναληπτικών Δικτύων, η οποία διευκολύνει το δίκτυο να αφομοιώνει και να αποθηκεύει πληροφορία για μεγαλύτερες περιόδους χρόνου. Το μοντέλο LSTM αποτελείτε από πυρήνες μνήμης (memory cells) πού έχουν μια πύλη εισόδου (input gate), μια πύλη εξόδου (output gate) και μια forget gate [17].



Εικόνα 2-25: Πύλες LSTM

Η πύλη εισόδου (input gate) ανιχνεύει ποια τιμή από την είσοδο θα χρησιμοποιηθεί για να τροποποιηθεί η μνήμη. Η σιγμοειδής συνάρτηση αποφασίζει ποιες τιμές θα περάσουν και η εφαπτομένη δίνει βάρος στις τιμές που περνάνε αποφασίζοντας το επίπεδο σημαντικότητας που κυμαίνεται μεταξύ μείον ένα και ένα.

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

$$\tilde{C}_t = tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right)$$

Η forget gate είναι υπεύθυνη για την απελευθέρωση μνήμης και αφαίρεση πληροφορίας από προηγούμενες καταστάσεις, που δεν χρειάζονται πλέον. Η λειτουργία αυτή υλοποιείται από την σιγμοειδή συνάρτηση, όπου έχει ως είσοδο την προηγούμενη κρυφή κατάσταση h_{t-1} και την είσοδο x_t από την ακολουθία διανυσμάτων. Η έξοδος της σιγμοειδής συνάρτησης είναι ένας αριθμός μεταξύ μηδέν και ένα, όπου μηδέν σημαίνει ότι παραλείπει την προηγούμενη κατάσταση C_{t-1} και ένα σημαίνει την κρατάει.

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

Έπειτα πρέπει να γίνει η ενημέρωση της προγενέστερης κατάστασης πυρήνα C_{t-1} σε C_t. Στα προηγούμενα βήματα έχει αποφασιστεί τι θα γίνει, το μόνο που απομένει είναι να γίνει πολλαπλασιάζοντας την προηγούμενη κατάσταση πυρήνα C_{t-1} με f_t, για να αφαιρεθεί η πληροφορία που δεν είναι απαραίτητη, και προσθέτουμε και $i_t \cdot \tilde{C}_t$.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

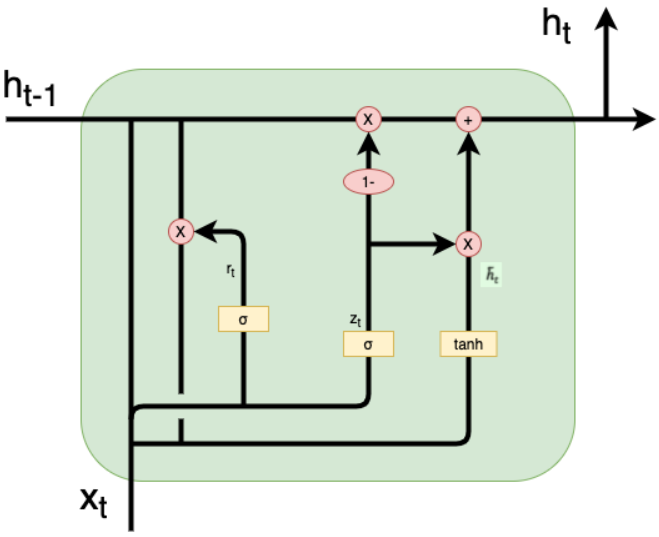
Τέλος πρέπει να αποφασιστεί τι θα έχουμε ως έξοδο μέσω της πύλης εξόδου (output gate). Αρχικά η προηγούμενη κρυφή κατάσταση h_{t-1} και η είσοδος x_t περνάνε από μια σιγμοειδή συνάρτηση για να αποφασιστεί ποια μέρη της τρέχουσας κατάστασης πυρήνα θα κρατήσουμε. Μετά εφαρμόζουμε στην τρέχουσα κατάσταση πυρήνα C_t στην εφαπτομένη συνάρτηση, για να υπάρχει ένα επίπεδο σημαντικότητας για κάθε τιμή, και το αποτέλεσμα το πολλαπλασιάζουμε με την έξοδο της σιγμοειδής συνάρτησης.

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

$$h_t = o_t \cdot \tanh\left(C_t\right)$$

2.4.2 Gated Recurrent Unit (GRU)

Μια ευρέως χρησιμοποιούμενη παραλλαγή του LSTM είναι το Gated Recurrent Unit (GRU). Σε αυτή την παραλλαγή αντί για τρεις πύλες χρησιμοποιούνται δύο, αφαιρέθηκε η πύλη εξόδου (output gate). Επιπλέον συγχωνεύονται η κατάσταση πυρήνα με την κρυφή κατάσταση [18].



Εικόνα 2-26: Πύλες GRU

Η πύλη ενημέρωσης (update gate) καθορίζει πόση πληροφορία από προηγούμενες καταστάσεις θα περάσουν στις επόμενες. Αυτό επιτυγχάνεται ενώνοντας την προηγούμενη κρυφή κατάσταση h_{t-1} και την είσοδο x_t, και εφαρμόζοντας την σιγμοειδή συνάρτηση έπειτα.

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

Η πύλη επαναφοράς (reset gate) ευθύνεται για το πόση πληροφορία από τις προηγούμενες καταστάσεις πρέπει να ξεχαστεί. Για να υπολογιστεί αυτό ακολουθείτε παρόμοια διαδικασία με την προηγούμενη πύλη.

$$r_t = \sigma\,(W_r \cdot [h_{t-1}, x_t])$$

Για να υπολογιστεί η υποψήφια τρέχουσα κρυφή κατάσταση πολλαπλασιάζεται η προηγούμενη κρυφή κατάσταση h_{t-1} με r_t και μαζί με την είσοδο x_t περνάνε από μια εφαπτομένη συνάρτηση.

$$\tilde{h}_t = tanh\,(W_h \cdot [r_t \;\cdot\; h_{t-1}, x_t])$$

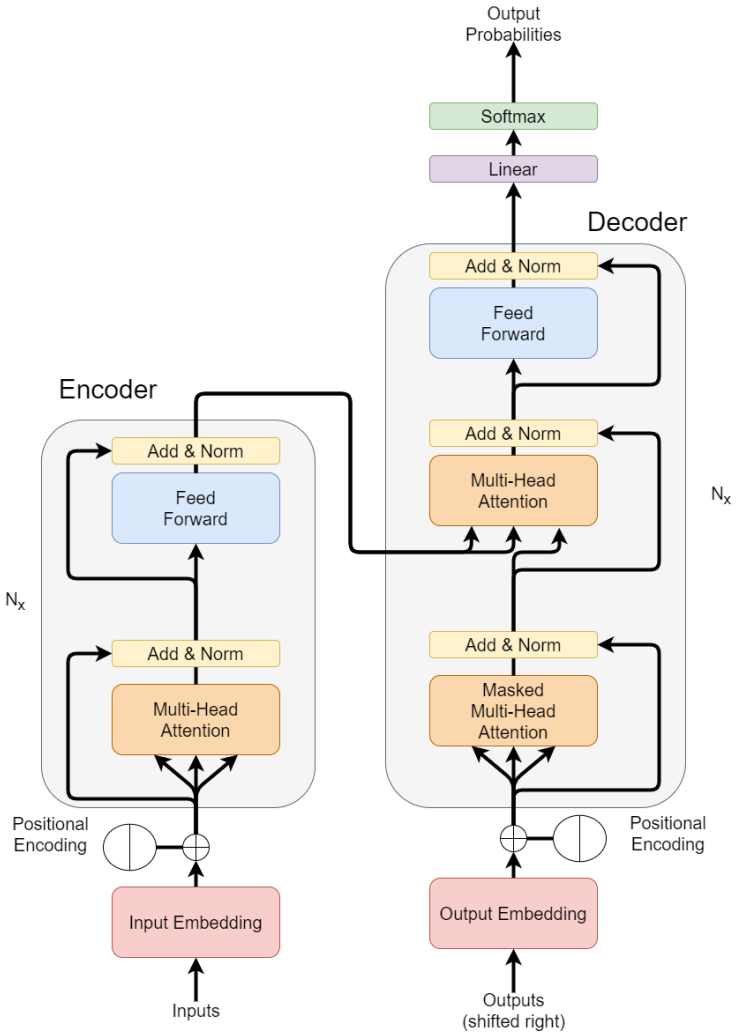
Το τελευταίο βήμα είναι ο υπολογισμός της κρυφή κατάστασης h_t.

$$h_t = (1 - \; z_t) \;\cdot\; h_{t-1} + \; z_t \cdot \; \tilde{h}_t$$

Η χρήση λιγότερων πυλών στο GRU τους δίνει το πλεονέκτημα στην αποθήκευση και επεξεργασία λιγότερων παραμέτρων από ένα μοντέλο LSTM. Γενικά το GRU έχει παρόμοια επίδοση με το LSTM σε αρκετά σύνολα δεδομένων και εργασίες. Ωστόσο στο πεδίο της μάθησης γλωσσών που βασίζονται σε γραμματικές χωρίς συμφραζόμενα το LSTM έχει αποδειχθεί καλύτερο [19].

2.5 Transformers

Οι Transformers στο πεδίο της επεξεργασίας φυσικής γλώσσας (NLP) είναι μια καινοτόμος αρχιτεκτονική που αποσκοπεί στην επίλυση προβλημάτων και εργασιών που χρησιμοποιούν μοντέλο ακολουθία σε ακολουθία (sequence to sequence) αλλιώς πολλά προς πολλά (many to many). Επιπλέον οι Transformers χειρίζονται με ευκολία μακροπρόθεσμες εξαρτήσεις. Η βασική ιδέα πίσω από την λειτουργία ενός Transformer είναι η διαχείριση των μακροπρόθεσμων εξαρτήσεων μεταξύ εισόδου και εξόδου χρησιμοποιώντας προσοχή (attention) και επανάληψη [20].



Εικόνα 2-27: Αρχιτεκτονική Transformer

Στην παραπάνω εικόνα φαίνεται η Αρχιτεκτονική ενός Transformer, όπου απαρτίζεται από ένα μπλοκ Encoder και ένα άλλο Decoder. Ο Encoder έχει ένα επίπεδο με Multi-Head Attention ακολουθούμενο από ένα επίπεδο νευρωνικού δικτύου απλής τροφοδότησης. Από την άλλη πλευρά ο Decoder έχει ένα επιπλέον επίπεδο Masked Multi-Head Attention, το οποίο χρησιμοποιείται μόνο κατά την διάρκεια της εκπαίδευσης και βοηθάει στην παραλληλοποίηση και στην ταχύτητα της εκπαίδευσης του μοντέλου. Τα μπλοκ του Encoder και του Decoder στην πραγματικότητα είναι πολλαπλοί ταυτόσημοι Encoders και Decoders στοιβαγμένοι. Η στοίβα των Encoders έχει το ίδιο πλήθος στοιχείων Encoder με το πλήθος στοιχείων Decoder στην στοίβα των Decoders.

Το επίπεδο Mutli-head Attention στην ουσία είναι πολλαπλοί υπολογισμοί του Self-attention που συνενώνονται έπειτα. Η διαδικασία αυτή επιτρέπει στο μοντέλο να δίνει

σημασία σε πληροφορία από διαφορετικές αναπαραστάσεις σε διαφορετικές θέσεις μιας ακολουθίας. Έτσι, λοιπόν, το Self-attention είναι ένας μηχανισμός συσχέτισης διαφορετικών θέσεων μιας ακολουθίας με σκοπό τον υπολογισμό μιας αναπαράστασής της [20]. Ο υπολογισμός του Self-attention ακολουθεί τα εξής βήματα:

1. Δημιουργία τριών διανυσμάτων για κάθε διάνυσμα εισόδου του Encoder, πολλαπλασιάζοντας το διάνυσμα εισόδου με τρείς πίνακες βαρών :
   - Query Vector
   - Key Vector
   - Value Vector

   Οι τιμές των βαρών και άρα και των διανυσμάτων ενημερώνονται κατά την διάρκεια της εκπαίδευσης. Το προεπιλεγμένο μέγεθος των διανυσμάτων είναι 64.
2. Υπολογισμός Self-attention αμφίδρομα, για κάθε λέξη και λαμβάνοντας υπόψη όλες τις λέξεις μιας πρότασης. Έστω η πρόταση “Creativity takes courage”, για να υπολογιστεί το Self-attention της λέξης “Creativity” πρέπει πρώτα να εκτιμηθεί η σημαντικότητα συσχέτισης όλων των λέξεων της πρότασης βάση κάποιων σκορ. Αυτό το σκορ καθορίζει την συμμετοχή των άλλων λέξεων στην αναπαράσταση και κωδικοποίηση μιας άλλης λέξης.
   a. Τα σκορ για τα πρώτη λέξη υπολογίζονται παίρνοντας εσωτερικό γινόμενο του Query διανύσματος με τα Keys διανύσματα (k_1, k_2, k_3) όλων των λέξεων της πρότασης.

Word	Query vector	Key vector	Value vector	score
Creativity	q_1	k_1	v_1	$q_1 \cdot k_1$
takes		k_2	v_2	$q_1 \cdot k_2$
courage		k_3	v_3	$q_1 \cdot k_3$

Πίνακας 2-1: Υπολογισμός σκορ για κάθε λέξη

   b. Έπειτα τα σκορ διαιρούνται με την τετραγωνική ρίζα της διάστασης του Key διανύσματος, δηλαδή με οχτώ. Αυτό οδηγεί σε πιο σταθερές κλίσεις [20]. Αφού γίνει η διαίρεση, τα σκορ κανονικοποιούνται χρησιμοποιώντας στην συνάρτηση SoftMax.

Word	Query vector	Key vector	Value vector	score	score/8($\sqrt{d_k}$)	SoftMax
Creativity	q_1	k_1	v_1	$q_1 \cdot k_1$	$q_1 \cdot k_1/8$	X_{11}

takes		k_2	v_2	$q_1 \cdot k_2$	$q_1 \cdot k_2/8$	X_{12}
courage		k_3	v_3	$q_1 \cdot k_3$	$q_1 \cdot k_3/8$	X_{13}

Πίνακας 2-2: Διαίρεση και κανονικοποίηση σκορ

c. Τα κανονικοποιήμενα σκορ πολλαπλασιάζονται με τα αντίστοιχα διανύσματα Value (v_1, v_2, v_3) και τα διανύσματα που βγαίνουν ως αποτέλεσμα προστίθενται για την παραγωγή του τελικού διανύσματος που είναι η είσοδος του δικτύου απλής τροφοδότησης.

Word	**Query vector**	**Key vector**	**Value vector**	**score**	**score/8($\sqrt{d_k}$)**	**SoftMax**	**SoftMax*v**	**Sum**
Creativity	q_1	k_1	v_1	$q_1 \cdot k_1$	$q_1 \cdot k_1/8$	X_{11}	$X_{11} \cdot v_1$	Z_1
takes		k_2	v_2	$q_1 \cdot k_2$	$q_1 \cdot k_2/8$	X_{12}	$X_{12} \cdot v_2$	
courage		k_3	v_3	$q_1 \cdot k_3$	$q_1 \cdot k_3/8$	X_{13}	$X_{13} \cdot v_3$	

Πίνακας 2-3: Παραγωγή τελικού διανύσματος της πρώτης λέξης

3. Η διαδικασία συνεχίζεται για όλες τις λέξεις της πρότασης.

Word	**Query vector**	**Key vector**	**Value vector**	**score**	**score/8($\sqrt{d_k}$)**	**SoftMax**	**SoftMax*v**	**Sum**
Creativity		k_1	v_1	$q_2 \cdot k_1$	$q_2 \cdot k_1/8$	X_{21}	$X_{21} \cdot v_1$	
takes	q_2	k_2	v_2	$q_2 \cdot k_2$	$q_2 \cdot k_2/8$	X_{22}	$X_{22} \cdot v_2$	Z_2
courage		k_3	v_3	$q_2 \cdot k_3$	$q_2 \cdot k_3/8$	X_{23}	$X_{23} \cdot v_3$	

Πίνακας 2-4: Παραγωγή τελικού διανύσματος της δεύτερης λέξης

Word	**Query vector**	**Key vector**	**Value vector**	**score**	**score/8($\sqrt{d_k}$)**	**SoftMax**	**SoftMax*v**	**Sum**
Creativity		k_1	v_1	$q_3 \cdot k_1$	$q_3 \cdot k_1/8$	X_{31}	$X_{31} \cdot v_1$	
takes		k_2	v_2	$q_3 \cdot k_2$	$q_3 \cdot k_2/8$	X_{32}	$X_{32} \cdot v_2$	
courage	q_3	k_3	v_3	$q_3 \cdot k_3$	$q_3 \cdot k_3/8$	X_{33}	$X_{33} \cdot v_3$	Z_3

Πίνακας 2-5: Παραγωγή τελικού διανύσματος της τρίτης λέξης

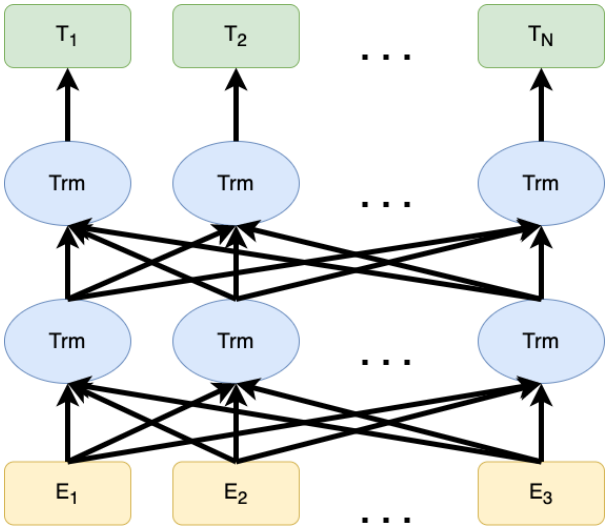
Πριν το τελικό διάνυσμα περάσει ως είσοδος στο δίκτυο απλής τροφοδότησης, μεσολαβεί ένα επίπεδο όπου προστίθενται το τελικό διάνυσμα με το αρχικό και γίνεται επιμέρους κανονικοποίηση επιπέδου [21].

Οι transformers αυτή την στιγμή πετυχαίνουν state-of-the-art αποτελέσματα στα περισσότερα σύνολα δεδομένων και προβλήματα επεξεργασίας φυσικής γλώσσας. Ωστόσο για την εκπαίδευση τους απαιτούνται ισχυροί υπολογιστικοί πόροι.

2.5.1 BERT

Το BERT είναι ένα μοντέλο γλώσσας βαθιάς μάθησης (deep learning language model) και τα αρχικά του σημαίνουν Bidirectional Encoder Representations for Transformers [22]. Η αρχιτεκτονική του BERT αποτελείτε από έναν πολύ-επίπεδο Transformer Encoder, όπου χρησιμοποιεί αμφίδρομο υπολογισμό του Self-attention για κάθε λεξικογραφική μονάδα της εισόδου, όπως παρουσιάστηκε στην προηγούμενη ενότητα. Η Google έχει κυκλοφορήσει δύο εκδοχές του BERT:

- BERT Base:
    - Πλήθος επιπέδων Transformer: 12 (στην ουσία πλήθος Encoders)
    - Πλήθος κεφαλών Attention: 12
    - Πλήθος παραμέτρων: 110 εκατομμύρια
- BERT Large:
    - Πλήθος επιπέδων Transformer: 24
    - Πλήθος κεφαλών Attention: 16
    - Πλήθος παραμέτρων: 340 εκατομμύρια



Εικόνα 2-28: Αρχιτεκτονική του BERT

Στην παραπάνω εικόνα φαίνεται ότι κάθε επίπεδο στο BERT δημιουργεί μια ενδιάμεση αναπαράσταση της λεκτικής μονάδας, άρα για το BERT με τα 12 επίπεδα έχουμε 12 ενδιάμεσες αναπαραστάσεις ίσου μεγέθους.

Το BERT έχει προ εκπαιδευτεί σε δεδομένα από την Wikipedia και το BookCorpus [23]. Το BERT base εκπαιδεύτηκε χρησιμοποιώντας 4 TPUs (Tensor Processing Unit) για 4 μέρες, ενώ το BERT large χρησιμοποιώντας 16 TPUs για 4 μέρες. Για να χρησιμοποιηθεί για την επίλυση κάποιου προβλήματος επεξεργασίας φυσικής γλώσσας απαιτείτε βελτιστοποίηση (fine-tuning) σύμφωνα με το σύνολο δεδομένων που έχουμε και το πρόβλημα που θέλουμε να επιλύσουμε.

Τα προβλήματα που προ εκπαιδεύτηκε το BERT είναι:

1. Masked Language Modeling (MLM): Στο πρόβλημα αυτό δεδομένου μια ακολουθίας λέξεων, πρέπει το μοντέλο να προβλέπει την επόμενη λέξη. Στην πράξη αντί να προβλέπεται κάθε επόμενη λεξικογραφική μονάδα, ένα ποσοστό των λεξικογραφικών μονάδων κρύβονται (masked) τυχαία και μόνο για αυτά γίνεται η πρόβλεψη. Οι κρυμμένες λέξεις κάποιες φορές δεν αντικαθίστανται από την λεξικογραφική μονάδα της μάσκας [MASK], επειδή οι κρυμμένες λεξικογραφικές μονάδες δεν θα παρουσιαστούν πριν την βελτιστοποίηση. Συγκεκριμένα 15% από τις λεξικογραφικές μονάδες επιλέγονται τυχαία και από αυτές το 80% αντικαθίσταται από την λεξικογραφική μονάδα της μάσκας, 10%

αντικαθίσταται από κάποια τυχαία λεξικογραφική μονάδα και 10% μένουν απαράλλαχτες.

2. Next Sentence Prediction (NSP): Αυτό το πρόβλημα είναι δυαδικής ταξινόμησης, όπου δεδομένου ενός ζεύγους προτάσεων, το μοντέλο καλείτε να προβλέψει αν όντως η δεύτερη πρόταση είναι η επόμενη πρόταση από την πρώτη.

Το μέγεθος του λεξικού και για τις δύο εκδοχές φτάνει τις 30 χιλιάδες λέξεις. Οι λεξικογραφικές μονάδες του λεξικού αντλούνται χρησιμοποιώντας Byte-Pair Encoding (BPE) σε επίπεδο χαρακτήρα, η οποία είναι μια τεχνική που βοηθάει στο χειρισμό μεγάλων λεξικών [24]. Στο Byte-Pair Encoding αντί για πλήρης λέξεις χρησιμοποιούνται υπολέξεις, οι οποίες εξάγονται μετά από στατιστική ανάλυση στο σώμα των κειμένων.

2.5.2 RoBERTa

Το RoBERTa ακολουθεί την ίδια αρχιτεκτονική με το BERT, αλλά διαφοροποιείται στον τρόπο προ εκπαίδευσης και στον τρόπο που παράγονται οι λεξικογραφικές μονάδες [25]. Τα αρχικά του RoBERTa προέρχονται από το Robustly optimized BERT approach. Συγκεκριμένα η προ εκπαίδευση του RoBERTa έγινε με μεγαλύτερα mini-Batches και σε περισσότερα δεδομένα, χρησιμοποιώντας ένα λεξικό μεγέθους 50 χιλιάδων λέξεων. Αυτή η αλλαγή προσθέτει επιπλέον 15 εκατομμύρια και 20 εκατομμύρια παραμέτρους για την base και large εκδοχή αντίστοιχα. Οι λεξικογραφικές μονάδες αντλούνται χρησιμοποιώντας την τεχνική Byte-Pair Encoding αλλά σε επίπεδο bytes αντί για Unicode χαρακτήρες [26]. Η προ εκπαίδευση έγινε μόνο για το πρόβλημα του Masked Language Modeling.

Η δύο εκδοχές του RoBERTa:

- RoBERTa Base:
    - Πλήθος επιπέδων Transformer: 12 (στην ουσία πλήθος Encoders)
    - Πλήθος κεφαλών Attention: 12
    - Πλήθος παραμέτρων: 125 εκατομμύρια
- RoBERTa Large:
    - Πλήθος επιπέδων Transformer: 24
    - Πλήθος κεφαλών Attention: 16
    - Πλήθος παραμέτρων: 355 εκατομμύρια

2.5.3 GPT-2

Το GPT-2, συντομογραφία του Generative Pre-Trained Transformer, είναι διάδοχος του GPT αναπτύχθηκε από την εταιρία OpenAI. Το GPT-2 εκπαιδεύτηκε στο να προβλέπει την επόμενη λέξη, δεδομένου τις προηγούμενες λέξεις ενός κειμένου, χρησιμοποιώντας ένα σύνολο δεδομένων 40GB που ονομάζεται WebText και που εξάχθηκε από το διαδίκτυο. [26] Αντί για μόνο Encoders, που είδαμε να χρησιμοποιούν το BERT και το RoBERTa, χρησιμοποιεί μόνο Decoders. Αυτή η διαφορά αλλάζει και τον τρόπο που υπολογίζεται το Self-attention. Πλέον αντί να υπολογίζεται λαμβάνοντας υπόψη κάθε λέξη της ακολουθίας, δίνουμε προσοχή μόνο στις λέξεις που βρίσκονται αριστερά της λέξης που υπολογίζεται το Self-attention. Ο τρόπος που δημιουργείται το λεξικό και το μέγεθος του λεξικού είναι ίδιος με αυτόν που περιεγράφηκε στην προηγούμενη ενότητα.

Έχουν βγει τέσσερις εκδοχές του GPT-2:

- GPT-2 Small:
  - Πλήθος επιπέδων Transformer: 12 (στην ουσία πλήθος Decoders)
  - Πλήθος κεφαλών Attention: 12
  - Πλήθος παραμέτρων: 125 εκατομμύρια
- GPT-2 Medium:
  - Πλήθος επιπέδων Transformer: 24
  - Πλήθος κεφαλών Attention: 16
  - Πλήθος παραμέτρων: 355 εκατομμύρια
- GPT-2 Large:
  - Πλήθος επιπέδων Transformer: 36
  - Πλήθος κεφαλών Attention: 20
  - Πλήθος παραμέτρων: 762 εκατομμύρια
- GPT-2 Extra Large:
  - Πλήθος επιπέδων Transformer: 48
  - Πλήθος κεφαλών Attention: 25
  - Πλήθος παραμέτρων: 1.5 δισεκατομμύρια

3 Δεδομένα

3.1 Λογισμικό και άδειες χρήσης

Google Colab: είναι μια από τις πιο διάσημες υπηρεσίες cloud για επιστήμονες δεδομένων, ερευνητές και μηχανικούς λογισμικού. Όχι μόνο παρέχει δωρεάν μια πλατφόρμα για σύνταξη κώδικα σε Python αλλά παρέχει και δωρεάν πρόσβαση σε υπολογιστικούς πόρους όπως κάρτα γραφικών (GPU) και TPU (Tensor Processing Unit), που χρειάζονται για υπολογιστικά προβλήματα όπως είναι η εκπαίδευση ενός Νευρωνικού Δικτύου. Για τα πειράματα που διεξήχθησαν χρησιμοποιήθηκε μία TPUv2 με 8 πυρήνες και 16GB μνήμης υψηλού εύρους ζώνης (HBM). Επιπλέον χρησιμοποιήθηκαν και 12 GB μνήμης τυχαίας προσπέλασης (RAM). Τέλος για να μην γίνεται κατάχρηση υπάρχουν κάποιοι περιορισμοί χρήσης και πόρων που δεν αναφέρονται συγκεκριμένα αλλά εμπειρικά προσδιορίζονται στο ότι επιτρέπεται χρήση GPU ή TPU μόνο για 6-8 ώρες την ημέρα.

Google Drive: αποτελεί μια cloud υπηρεσία για αποθήκευση αρχείων και συγχρονισμό τους. Εκεί έχουν αποθηκευτεί το σύνολο δεδομένων, ο κώδικας, τα αποτελέσματα και ότι άλλο ήταν απαραίτητο. Ο λόγος που επιλέχθηκε ήταν ο κάλος συνδυασμός χρήσης του με το Google Colab. Δεν αναφέρεται κάποιος χωρικός περιορισμός.

Γλώσσα προγραμματισμού και βιβλιοθήκες που χρησιμοποιήθηκαν για την Ανάλυση Δεδομένων:

- Python: αποτελεί μια γλώσσα προγραμματισμού υψηλού επιπέδου και γενικού σκοπού. Δημιουργήθηκε από τον Guido van Rossum και κυκλοφόρησε το 1991. Το όνομα της εμπνεύστηκε από τους Monty Python, μια βρετανική κωμική ομάδα. Την τελευταία δεκαετία έχει γνωρίσει ιδιαίτερη αναγνώριση από τους προγραμματιστές καθώς βρίσκετε στις κορυφαίες τρεις πιο συχνά χρησιμοποιούμενες γλώσσες προγραμματισμού. Οι λόγοι που η χρήση της είναι τόσο διαδεδομένη είναι ότι μπορεί να χρησιμοποιηθεί τόσο ως μια αντικειμενοστραφής γλώσσα προγραμματισμού, όσο ως και μια διαδικαστική ενώ είναι dynamically typed και επιπλέον είναι ανοικτού κώδικα υπό την άδεια χρήσης PSFL. Τέλος η σύνταξη της ακολουθεί αυστηρές οδηγίες για την αποφυγή ανάπτυξη περίπλοκου κώδικα. Η έκδοση που χρησιμοποιείται είναι η 3.6.9.

- TensorFlow: είναι μια πλατφόρμα ανοικτού κώδικα γραμμένη σε Python, C++ και CUDA. Κύρια χρήση της είναι σε εφαρμογές μηχανικής μάθησης όπως τα Νευρωνικά Δίκτυα. Χρησιμοποιείτε υπό την άδεια χρήσης Apache License 2.0. Η έκδοση που χρησιμοποιείται είναι η 2.3.0.
- Keras: αποτελεί μια βιβλιοθήκη ανοικτού κώδικα για Νευρωνικά Δίκτυα γραμμένη σε Python. Η βιβλιοθήκη λειτουργεί πάνω από το TensorFlow και προσφέρει έναν τρόπο φιλικό προς τον χρήστη και επεκτάσιμο για να πειραματιστεί με τα Βαθιά Νευρωνικά Δίκτυα. Χρησιμοποιείτε υπό την άδεια χρήσης MIT. Η έκδοση που χρησιμοποιείται είναι η 2.4.3.
- scikit-learn (sklearn): είναι μια βιβλιοθήκη ανοικτού κώδικα για μηχανική μάθηση σε Python. Περιέχει πολλούς αλγορίθμους για ταξινόμηση, ομαδοποίηση και παλινδρόμηση καθώς και υλοποιημένους τρόπους για τον υπολογισμό διάφορων μετρικών. Χρησιμοποιείτε υπό την άδεια χρήσης New BSD License. Η έκδοση που χρησιμοποιείται είναι η 0.22.2. post1.
- transformers: αποτελεί μια βιβλιοθήκη παροχής προ εκπαιδευμένων γλωσσικών μοντέλων που χρησιμοποιούνται στα περισσότερα προβλήματα επεξεργασίας φυσικής γλώσσας. Συνδυάζεται καλά με την TensorFlow. Χρησιμοποιείτε υπό την άδεια χρήσης Apache License 2.0. Η έκδοση που χρησιμοποιείται είναι η 2.11.0.
- NumPy: είναι μια βιβλιοθήκη για αριθμητικούς υπολογισμούς στην Python. Δημιουργήθηκε το 2005 και είναι ανοικτού κώδικα υπό την διαμορφωμένη άδεια χρήσης BSD-2. Η έκδοση που χρησιμοποιείται είναι η 1.18.5.
- pandas: αποτελεί μια βιβλιοθήκη της Python που χρησιμοποιείται για ανάγνωση, εγγραφή και επεξεργασία συνόλων δεδομένων μέσω DataFrames. Η βιβλιοθήκη είναι ανοικτού κώδικα υπό την άδεια χρήσης BSD-2 και έχει βελτιστοποιηθεί για την καλύτερη απόδοση ενέργειων διαχείρισης συνόλου δεδομένων [28]. Η έκδοση που χρησιμοποιείται είναι η 1.0.5.
- Matplotlib: είναι βιβλιοθήκη της Python για δημιουργία παραστάσεων και γραφημάτων και ανοικτού κώδικα υπό την άδεια χρήσης PSFL. Η έκδοση που χρησιμοποιείται είναι η 3.2.2.
- Seaborn: αποτελεί βιβλιοθήκη της Python για απεικόνιση δεδομένων και παρέχει μια υψηλού επιπέδου διεπαφή για σχεδίαση ελκυστικών στατιστικών γραφημάτων.

Χρησιμοποιείται υπό την άδεια χρήσης BSD-3. Η έκδοση που χρησιμοποιείται είναι η 0.10.1.

- wordcloud: βιβλιοθήκη της Python για παραγωγή word clouds. Χρησιμοποιείται υπό την άδεια χρήσης MIT. Η έκδοση που χρησιμοποιείται είναι η 1.5.0.
- nltk: είναι μια βιβλιοθήκη της Python που αποτελεί εργαλειοθήκη των προγραμματιστών που ασχολούνται με το πεδίο της επεξεργασίας φυσική γλώσσας. Προσφέρει μια ποικιλία εργαλείων για επεξεργασία και απεικόνιση δεδομένων κειμένου. Χρησιμοποιείται υπό την άδεια χρήσης Apache License 2.0. Η έκδοση που χρησιμοποιείται είναι η 3.2.5.
- PIL: αποτελεί βιβλιοθήκη της Python για διαχείριση εικόνων. Χρησιμοποιείται υπό την άδεια χρήσης PIL Software License. Η έκδοση που χρησιμοποιείται είναι η 7.0.0.
- Emoji: βιβλιοθήκη της Python για διαχείριση emoji. Χρησιμοποιείται υπό την άδεια χρήσης BSD. Η έκδοση που χρησιμοποιείται είναι η 0.6.0.
- pickle: αποτελεί βιβλιοθήκη της Python για διαχείριση αρχείων της μορφής pickle. Η μορφή pickle εφαρμόζει δυαδικά πρωτόκολλα για σειριοποίηση και αποσειριοποίηση μιας δομής αντικειμένου Python. “Pickling” είναι η διαδικασία κατά την οποία μια ιεραρχία αντικειμένων Python μετατρέπεται σε ροή byte, και “unpickling” είναι η αντίστροφη λειτουργία, όπου μια ροή byte (από δυαδικό αρχείο) μετατρέπεται σε ιεραρχία αντικειμένων. Το μοντέλο Pickle εφαρμόζεται για γρήγορο φόρτωμα των δεδομένων στην ram.

3.2 Προέλευση του συνόλου δεδομένων

Το σύνολο δεδομένων που χρησιμοποιήθηκε είναι μέρος του διαγωνισμού του Kaggle Jigsaw Unintended Bias in Toxicity Classification και συγκεκριμένα έχουν αντληθεί από την πλατφόρμα Civil Comments, ένα πλήρες plugin για τον σχολιασμό ιστοσελίδων ειδήσεων [27].

Στα τέλη του 2017 η πλατφόρμα Civil Comments έκλεισε και αποφάσισε να κάνει διαθέσιμο το σύνολο δεδομένων της, που απαρτίζεται από περίπου 2 εκατομμύρια δημόσια σχόλια, σε ένα διαρκές ανοικτό αρχείο με σκοπό ερευνητές να καταλάβουν και να βελτιώσουν την ποιότητα των διαδικτυακών συνομιλίων. Η Jigsaw χρηματοδότησε αυτήν την προσπάθεια και επέκτεινε τον σχολιασμό αυτών των δεδομένων δημιουργώντας

επιπλέον τοξικά χαρακτηριστικά από ανθρώπους βαθμολογητές. Το σύνολο δεδομένων κυκλοφορεί υπό την άδεια χρήσης CCO, για ερευνητική χρήση.

3.3 Διερευνητική ανάλυση του συνόλου δεδομένων

3.3.1 Κατανόηση σχήματος δεδομένων

Το σύνολο δεδομένων απαρτίζεται από ένα σύνολο παραδειγμάτων εκπαίδευσης και ένα δημόσιο (public) σύνολο παραδειγμάτων ελέγχου και ένα ιδιωτικό (private).

- Μέγεθος συνόλου εκπαίδευσης: 1.8 εκατομμύρια παραδείγματα
    - Αριθμός μη-τοξικών σχολίων: 1.66 εκατομμύρια (92% του συνόλου)
    - Αριθμός τοξικών σχολίων: 144 χιλιάδες (8% του συνόλου)

  Βλέπουμε δυσαναλογία στα δεδομένα.
- Μέγεθος δημόσιου συνόλου ελέγχου: 97 χιλιάδες παραδείγματα
- Μέγεθος ιδιωτικού συνόλου ελέγχου: 97 χιλιάδες παραδείγματα

Το κείμενο των σχολίων βρίσκεται στην στήλη comment_text που έχουν όλα τα σύνολα. Κάθε σχόλιο στο σύνολο εκπαίδευσης έχει μια ετικέτα τοξικότητας στην στήλη target που παίρνει τιμές από 0 έως 1, άρα το μοντέλο πρέπει να προβλέψει τις τιμές στην στήλη target στα σύνολα ελέγχου. Μόνο για την αξιολόγηση παρέχονται οι τιμές της στήλης target στα σύνολα ελέγχου. Επιπλέον σχόλια στα σύνολα ελέγχου με target μεγαλύτερο ή ίσο του 0.5 θεωρούνται τοξικά και το ανάποδο για τα μη τοξικά. Στο σύνολο εκπαίδευσης υπάρχουν και έξι στήλες που αναπαριστούν υποκατηγορίες της τοξικότητας θα τις χρειαστούμε για βοηθητική έξοδο στο μοντέλο. Επιπλέον ένα υποσύνολο των σχολίων έχει επισημανθεί με μια ποικιλία χαρακτηριστικών ταυτότητας (identity attributes), που αντιπροσωπεύουν τις ταυτότητες που αναφέρονται στο εκάστοτε σχόλιο. Από τις στήλες που αντιστοιχούν στις ταυτότητες, στο σύνολο τους είναι 25 και παίρνουν τιμές από 0 έως 1, θα χρησιμοποιήσουμε μόνο αυτές που οι ταυτότητες τους αναφέρονται σε περισσότερα από 500 σχόλιο στα σύνολα ελέγχου, το ποιες είναι αυτές οι στήλες μας δίνονται από την εκφώνηση του διαγωνισμού είναι στον αριθμό 9. Τέλος μας δίνονται κάποιες στήλες μετά δεδομένων (created_date, publication_id, parent_id, article_id, rating, funny, wow, sad, likes, disagree, sexual_explicit, identity_annotator_count, toxicity_annotator_count) που δεν θα χρησιμοποιήσουμε.

	id	comment_text	target	severe_toxicity	obscene	identity_attack	insult	threat	male	female	homosexual_gay_or_lesbian	christian	jewish	muslim	black	white	psychiatric_or_mental_illness
0	59848	This is so cool. It's like, 'would you want yo...	0.000000	0.000000	0.0	0.000000	0.00000	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	59849	Thank you!! This would make my life a lot less...	0.000000	0.000000	0.0	0.000000	0.00000	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	59852	This is such an urgent design problem; kudos t...	0.000000	0.000000	0.0	0.000000	0.00000	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	59855	Is this something I'll be able to install on m...	0.000000	0.000000	0.0	0.000000	0.00000	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	59856	haha you guys are a bunch of losers.	0.893617	0.021277	0.0	0.021277	0.87234	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Εικόνα 3-1: Δομή συνόλου εκπαίδευσης

Τελικά για το σύνολο εκπαίδευσης από τις 45 στήλες κρατάμε τις 17:

- id - ένα μοναδικό αναγνωριστικό για κάθε σχόλιο
- comment_text – το κείμενο των σχολίων.
- target - υποδηλώνει εάν ένα σχόλιο είναι τοξικό (τιμή μεγαλύτερη ή ίση του 0.5) ή όχι. Παίρνει τιμές από 0.0 έως 1.0.
- severe_toxicity, obsene, identity_attack, insult, threat – υποκατηγορίες τοξικότητας παίρνουν τιμές από 0.0 έως 1.0.
- male, female, homosexual_gay_or_lesbian, christian, jewish, muslim, black, white, psychiatric_or_mental_illness – στήλες ταυτοτήτων μπορεί να είναι null, παίρνουν τιμές από 0.0 έως 1.0.

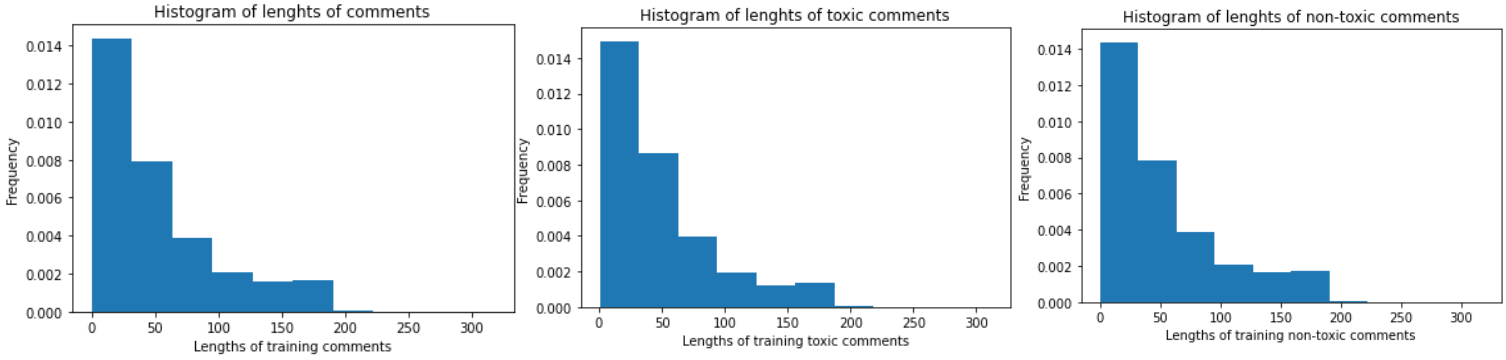
Τα σύνολα ελέγχου έχουν τρείς στήλες που είναι όπως οι τρεις πρώτες του συνόλου εκπαίδευσης. Η τρίτη στήλη χρησιμοποιείται μόνο στην αξιολόγηση. Στην συνέχεια της ανάλυσης δεδομένων θα χρησιμοποιηθεί μόνο το σύνολο εκπαίδευσης.

Όπως προαναφέρθηκε και φαίνεται στην εικόνα 3-1 οι στήλες ταυτοτήτων έχουν null τιμές. Στην παρακάτω εικόνα φαίνεται και το ποσοστό των τιμών που λείπουν από κάθε στήλη και παρατηρείται μια ομοιόμορφη κατανομή.

```
male                          77.553558
female                        77.553558
homosexual_gay_or_lesbian     77.553558
christian                     77.553558
jewish                        77.553558
muslim                        77.553558
black                         77.553558
white                         77.553558
psychiatric_or_mental_illness 77.553558
```

Εικόνα 3-2: Ποσοστό τιμών που λείπουν ανά στήλη

Ο έλεγχος για την συχνότητα του μεγέθους των κειμένων των σχολίων έδειξε πως το μέγιστο μέγεθος σχολίου είναι 317 λεξικογραφικές μονάδες (tokens). Δεν αρκεί μόνο αυτό για να αποφασίσουμε το μέγεθος εισόδου που πρέπει να ορίσουμε για το μοντέλο πρέπει να δούμε και την κατανομή του μεγέθους.



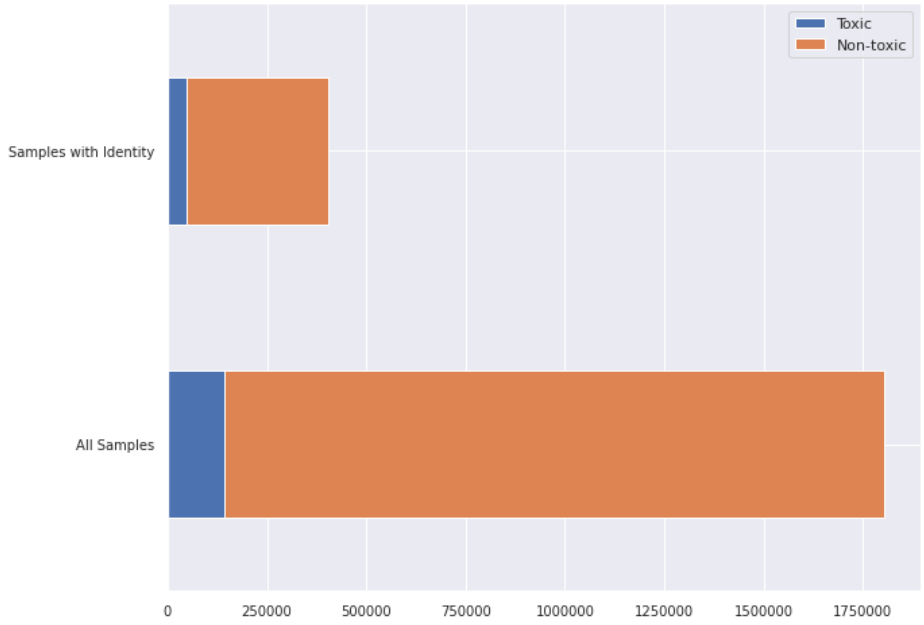
Εικόνα 3-3: Κατανομή μεγέθους σχολίων

Όπως φαίνεται στις παραπάνω εικόνες το μέγεθος του κειμένου ενός σχολίου δεν διαδραματίζει σημαντικό ρόλο στην ταξινόμηση του σχολίου. Καλή επιλογή για μέγεθος εισόδου στο μοντέλο είναι τα 180 tokens.

3.3.2 Κατανόηση του υποσυνόλου των Δεδομένων με ταυτότητα

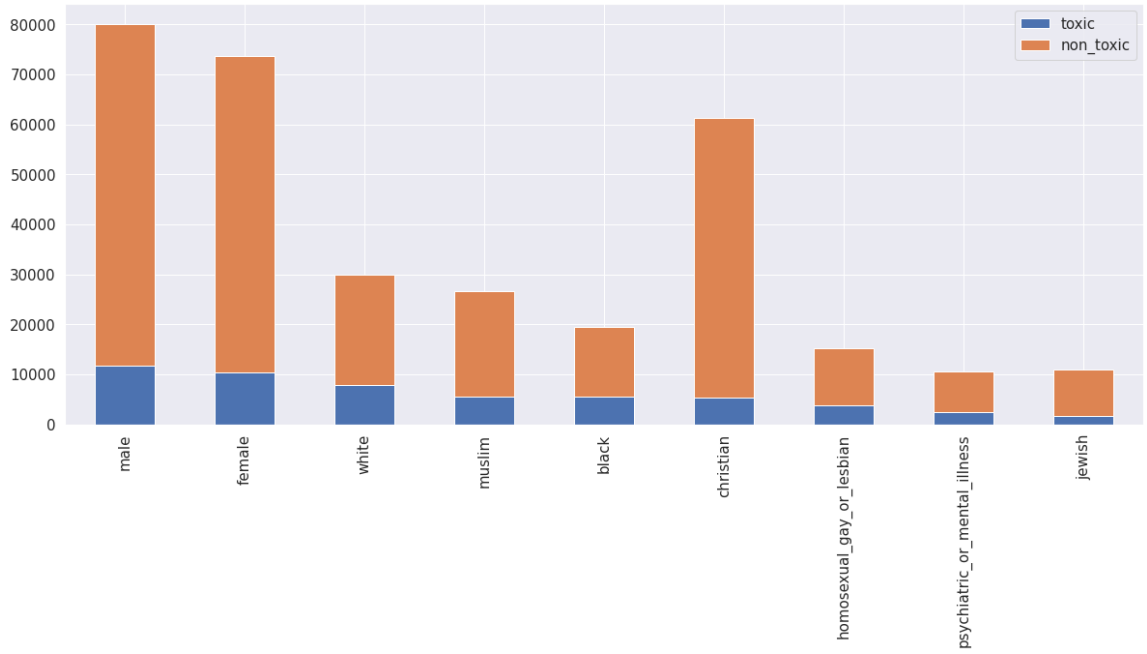
Μέγεθος υποσυνόλου σχολίων με ταυτότητα: 405 χιλιάδες παραδείγματα (22.5% του συνόλου).

- Αριθμός μη-τοξικών σχολίων: 359 χιλιάδες (88.6% του υποσυνόλου)
- Αριθμός τοξικών σχολίων: 46 χιλιάδες (11.4% του υποσυνόλου)

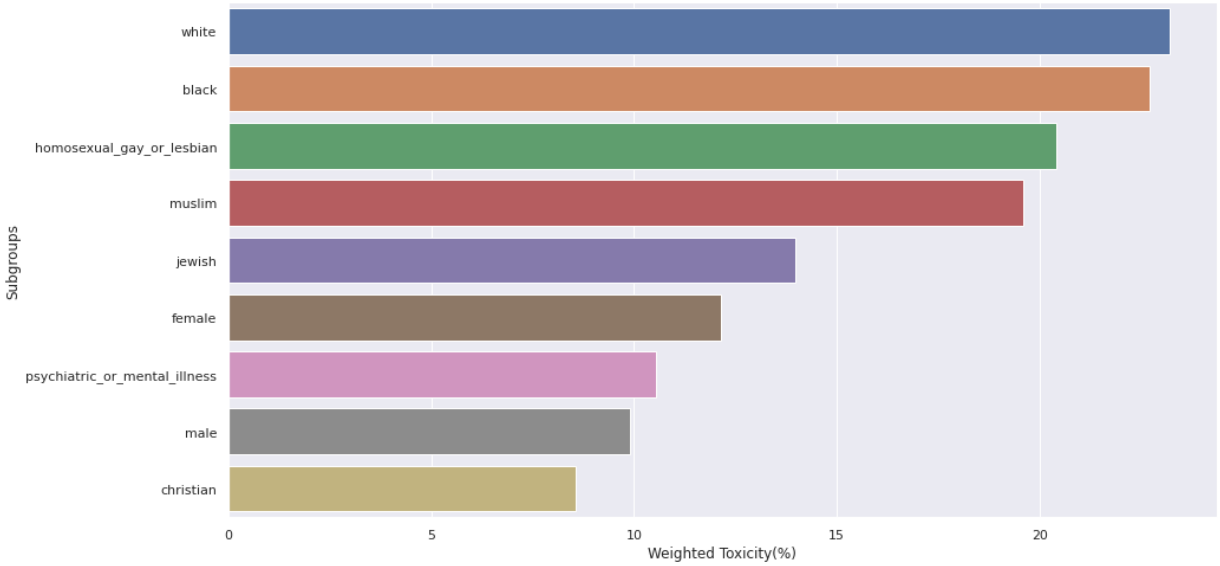


Εικόνα 3-4: Κατανομή των κλάσεων στο σύνολο και στο υποσύνολο των παραδειγμάτων με ταυτότητα

Όπως φαίνεται και από την παραπάνω εικόνα το σύνολο ακολουθεί την άνιση κατανομή των κλάσεων του συνόλου.

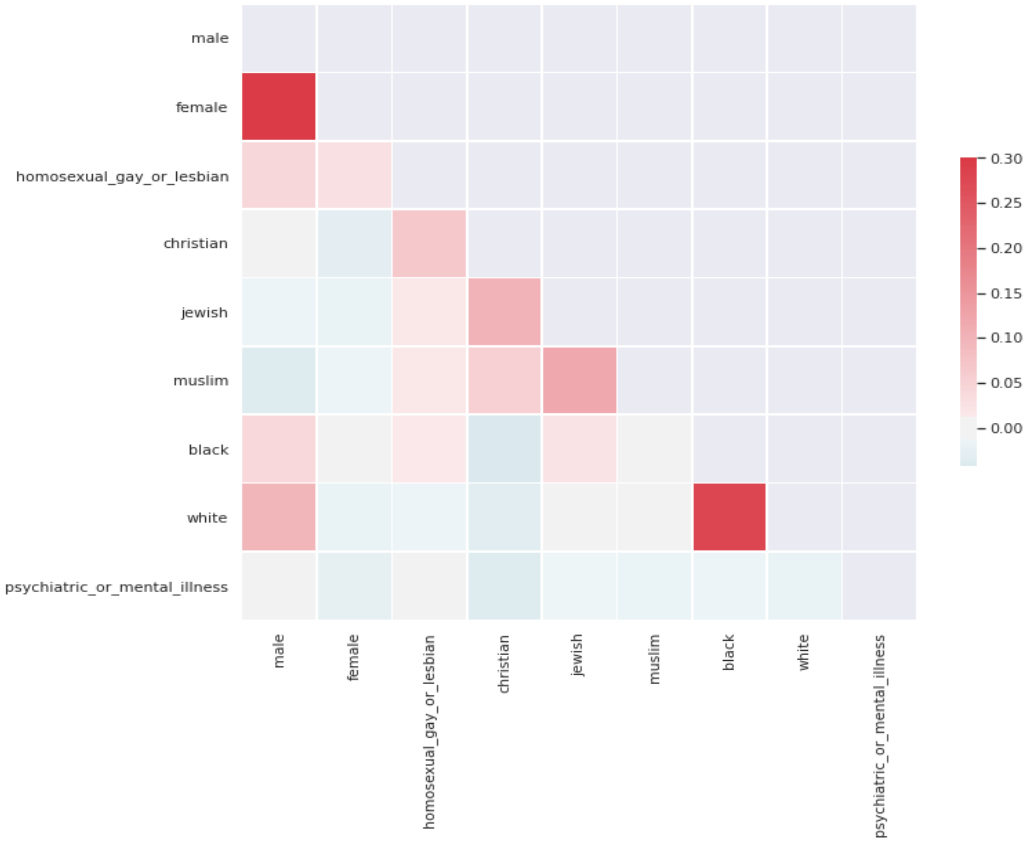


Εικόνα 3-5: Κατανομή κλάσεων ανά ταυτότητα



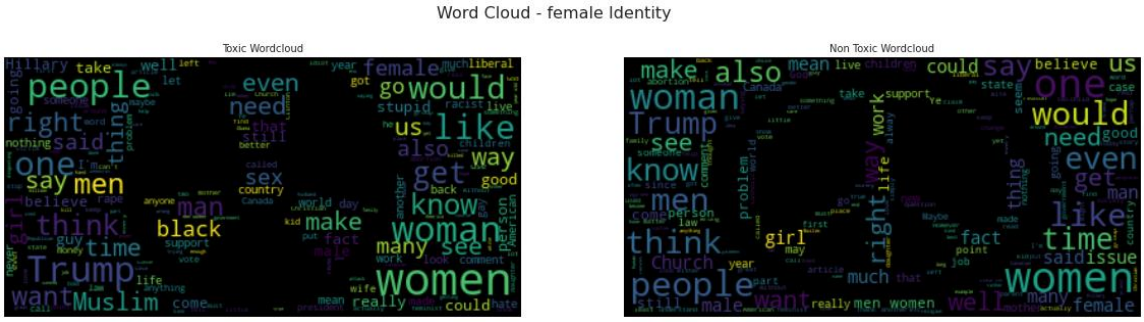
Εικόνα 3-6: Σταθμισμένη τοξικότητα ανά ταυτότητα

Όπως φαίνεται από τις εικόνες 3-5 και 3-6 για τις ταυτότητες white, black, homosexual_gay_or_lesbian και muslim υπάρχει αυξημένη τοξικότητα στα σχόλια που αναφέρονται αναλογικά με το πλήθος τους.

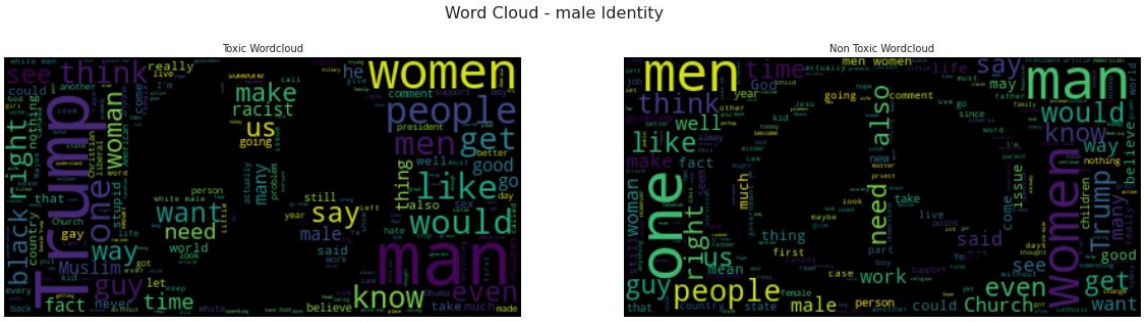


Εικόνα 3-7: Χάρτης συσχέτισης των ταυτοτήτων

Στον χάρτη της εικόνας 3-7 ενδιαφέρουσα είναι η περιοχή στην άκρη της διαγώνιου όπου φαίνεται ότι σε τοξικά και μη σχόλια συχνά αναφέρονται ταυτότητες που συχνά αντικρούονται. Τέτοιες ταυτότητες είναι : male και female, black και white, muslim και jewish. Παρακάτω παράγονται σύννεφα λέξεων (word clouds) αυτών των ταυτοτήτων.

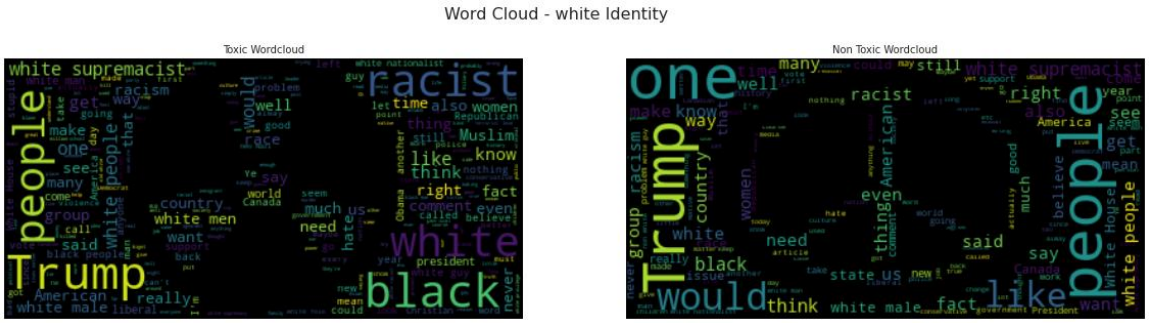


Εικόνα 3-8: Word cloud της ταυτότητας female

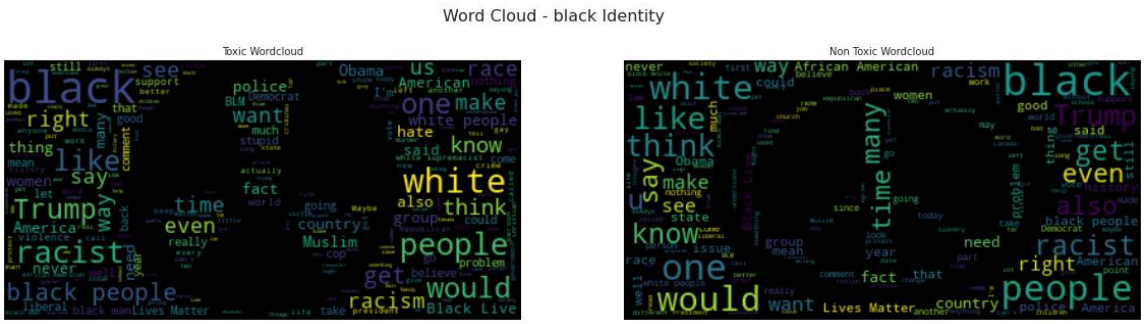


Εικόνα 3-9: Word cloud της ταυτότητας male

Αν και η σημασιολογία διαφέρει στο κείμενο των τοξικών και μη τοξικών σχολίων στις παραπάνω εικόνες φαίνεται πώς δεν υπάρχει μεγάλη διαφορά στις λέξεις που χρησιμοποιούνται πιο συχνά.

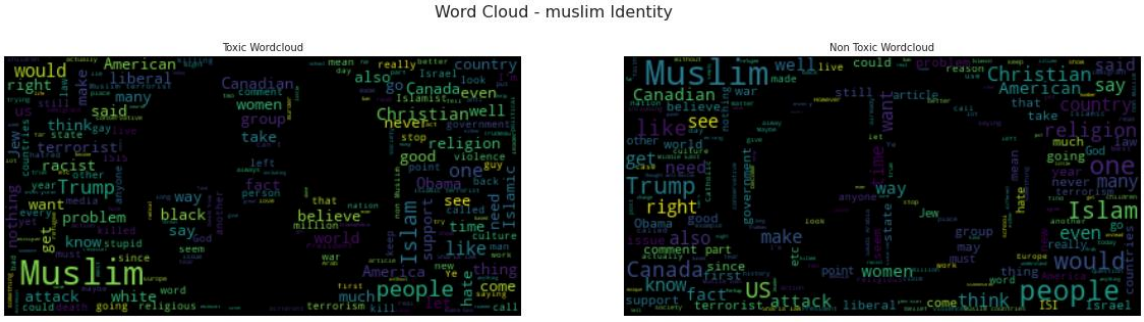


Εικόνα 3-10: Word cloud της ταυτότητας white

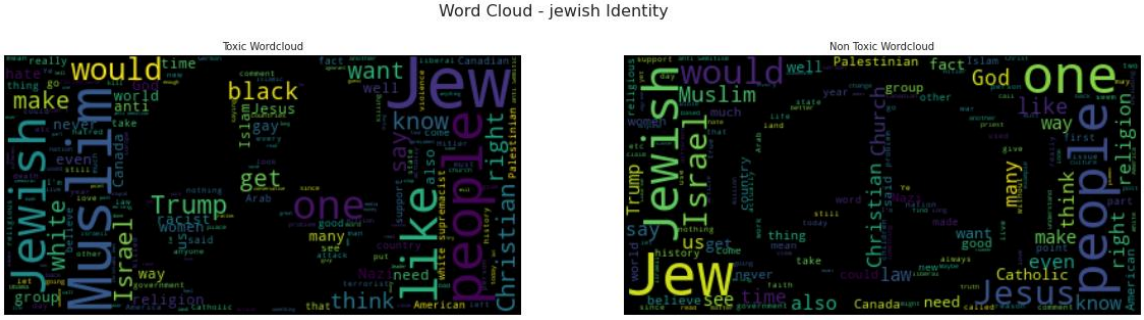


Εικόνα 3-11: Word cloud της ταυτότητας black

Φαίνεται πως μεταξύ των ταυτοτήτων black και white υπάρχει μια μεγάλη τομή στις λέξεις που χρησιμοποιούνται τόσο στα τοξικά σχόλια όσο και στα μη τοξικά.



Εικόνα 3-12: Word cloud της ταυτότητας muslim



Εικόνα 3-13: Word cloud της ταυτότητας jewish

Από τις δύο παραπάνω εικόνες παρατηρείται ότι τοξικά σχόλια που αναφέρονται στην ταυτότητα jewish συχνά εμφανίζουν και την λέξη Muslim ενώ δεν συμβαίνει το ίδιο στο αντίστοιχο word cloud της ταυτότητας muslim. Άξιο αναφοράς είναι ότι πολύ συχνά και μάλιστα περισσότερο στα τοξικά σχόλια όλων των ταυτοτήτων στις εικόνες εμφανίζεται το όνομα Trump, που φυσικά αναφέρεται στον πρόεδρο της Αμερικής για την περίοδο 2017-2021.

3.4 Προ-επεξεργασία και καθαρισμός του συνόλου δεδομένων

Σκοπός της προ-επεξεργασίας για αυτό το σύνολο δεδομένων είναι ο καθαρισμός του κειμένου των σχολίων έτσι ώστε να υπάρχει για όσες περισσότερες λέξεις είναι δυνατό κάλυψη στα προ εκπαιδευμένα διανύσματα λέξεων (embeddings).

3.4.1 Word Embeddings

Τα word embeddings είναι το μέσο που χρειαζόμαστε ώστε μια λέξη να γίνεται κατανοητή και υπολογίσιμη για έναν υπολογιστή. Λέξεις που συναντώνται συχνά σε παρόμοιο πλαίσιο κειμένου τείνουν να μοιάζουν και σημασιολογικά. Χρησιμοποιώντας την λογική αυτή, τα word embeddings αποθηκεύουν πληροφορία με βάση τα συμφραζόμενα σε ένα διάνυσμα λίγων διαστάσεων. Τα διανύσματα μαθαίνονται από μοντέλα μέσω μάθησης χωρίς επίβλεψη σε μεγάλα σύνολα δεδομένων κειμένου όπως η Wikipedia ή σε δεδομένα κειμένου που εξήχθησαν γενικά από τα διαδίκτυο. Τα πρώτα μοντέλα ήταν κυρίως στατιστικά και είχαν προβλήματα προσαρμοστικότητας σε καινούρια δεδομένα και ήταν ευαίσθητα στην ανισότητα της συχνότητας των λέξεων.

Το 2013 το μοντέλο Word2Vec έφερε την καινοτομία στο χώρο των word embeddings χρησιμοποιώντας παρόμοια αρχιτεκτονική με ένα νευρωνικό δίκτυο απλής τροφοδότησης [29] [30]. Έχουν βγει δύο παραλλαγές του Word2vec:

1. Continuous Bag-of-words (CBOW): στόχος η πρόβλεψη μιας λέξης βάσει ενός παραθύρου που περιέχει τις γειτονικές λέξεις.
2. SkipGram (SG): στόχος η πρόβλεψη των συμφραζόμενων βάση μιας λέξης.

Το Word2Vec έχει καλή επίδοση στα προβλήματα επεξεργασίας φυσικής γλώσσας, ωστόσο έχει κάποια σημαντικά μειονεκτήματα:

1. Λειτουργεί βασισμένο σε ένα παράθυρο, όποτε δεν επωφελείται από την πληροφορία ολόκληρου του κειμένου.
2. Δεν μπορεί να διαχειριστεί λέξεις που δεν υπάρχουν στο λεξικό του (OOV, Out Of Vocabulary, words).
3. Δεν καταλαβαίνει λέξεις με διπλό νόημα.

Το δεύτερο πρόβλημα το επιλύνει η μέθοδος FastText για εκμάθηση των αναπαραστάσεων των λέξεων, η οποία βασίζεται στο μοντέλο SkipGram του Word2vec [31]. Χρησιμοποίει όλα τα n-grams μιας λέξης για την εύρεση κάποιου παράγωγού της ή κάποιας υπό λέξης της. Για τον υπολογισμό των word embedding του συνόλου δεδομένων

μας θα χρησιμοποιήσουμε ένα σύνολο δεδομένων σε μορφή pickle με 2 εκατομμύρια διανύσματα λέξεων με μέγεθος διανύσματος 300 διαστάσεις, εκπαιδευμένα με την μέθοδο FastText σε δεδομένα κειμένου από το διαδίκτυο (Common Crawl), με άδεια χρήσης CC BY-SA 3.0 [32].

Τα προβλήματα στα στατιστικά μοντέλα για εκμάθηση αναπαραστάσεων λέξεων ήρθε να επιλύσει το GloVe (Global Vectors for Word Representation) [33]. Η λειτουργία του παίρνει υπόψη στατιστικά συνεμφάνισης λέξεων. Για τον υπολογισμό των word embedding του συνόλου δεδομένων μας θα χρησιμοποιήσουμε και ένα σύνολο δεδομένων σε μορφή pickle με 2.2 εκατομμύρια διανύσματα λέξεων με μέγεθος διανύσματος 300 διαστάσεις, εκπαιδευμένα με την μέθοδο GloVe σε δεδομένα κειμένου από το διαδίκτυο (Common Crawl), με άδεια χρήσης PDDL [34].

3.4.2 Βελτιστοποίηση κάλυψης λεξικού

Για λόγους ταχύτητας και εξοικονόμησης χώρου μετριέται η κάλυψη μόνο για το σύνολο δεδομένων διανυσμάτων λέξεων από το GloVe. Η προ επεξεργασία εφαρμόζεται στο σύνολο δεδομένων εκπαίδευσης και στα σύνολο δεδομένων ελέγχου. Αρχική κάλυψη: Βρέθηκαν embeddings για το 15.82 % του λεξικού και για το 89.63 % των λέξεων στα σχόλια.

Λεξικογραφική Μονάδα	Εμφανίσεις
isn't	39964
That's	37640
won't	29397
he's	24353

Πίνακας 3-1: Οι τέσσερις πιο συχνά χρησιμοποιούμενες άγνωστες λέξεις

- 1ο Βήμα : Φαίνεται πως το σημείο στίξης «'» αποτελεί πρόβλημα. Ένα καλό πρώτο βήμα για τον καθαρισμό των κειμένων είναι η αφαίρεση άγνωστων συμβόλων και ο διαχωρισμός των συναιρέσεων. Νέα κάλυψη: Βρέθηκαν embeddings για το 52.32 % του λεξικού και για το 99.58 % των λέξεων στα σχόλια.

Λεξικογραφική Μονάδα	Εμφανίσεις
tRump	2522
gov't	2237

Brexit	1729
theglobeandmail	1350

Πίνακας 3-2: Οι τέσσερις πιο συχνά χρησιμοποιούμενες άγνωστες λέξεις μετά το 1ο βήμα

- 2ο Βήμα: Για επόμενο βήμα τσεκάρουμε για τις λέξεις που δεν υπάρχει embedding αν γραφούν μόνο με κεφαλαία ή μόνο με μικρά υπάρχει embedding. Νέα κάλυψη : Βρέθηκαν embeddings για το 54.68 % του λεξικού και για το 99.61 % των λέξεων στα σχόλια.

Λεξικογραφική Μονάδα	Εμφανίσεις
gov't	2237
Brexit	1729
theglobeandmail	1350
'the	1300

Πίνακας 3-3: Οι τέσσερις πιο συχνά χρησιμοποιούμενες άγνωστες λέξεις μετά το 2ο βήμα

- 3ο Βήμα: Διαχωρισμός από γνωστών συναιρέσεων όπως το «gov't» σε «government». Νέα κάλυψη: Βρέθηκαν embeddings για το 54.69 % του λεξικού και για το 99.61 % των λέξεων στα σχόλια.

Λεξικογραφική Μονάδα	Εμφανίσεις
Brexit	1729
theglobeandmail	1350
'the	1300
Drumpf	1183

Πίνακας 3-4: Οι τέσσερις πιο συχνά χρησιμοποιούμενες άγνωστες λέξεις μετά το 3ο βήμα

- 4ο Βήμα: Διαχωρισμός λέξεων που είναι συνδυασμός κάποιων άλλων και έχουν παραχθεί πρόσφατα, όπως το «Brexit» σε «British exit». Νέα κάλυψη: Βρέθηκαν embeddings για το 54.69 % του λεξικού και για το 99.62 % των λέξεων στα σχόλια.

Λεξικογραφική Μονάδα	Εμφανίσεις
'the	1300
'The	843
'I	639
'bout	515

Πίνακας 3-5: Οι τέσσερις πιο συχνά χρησιμοποιούμενες άγνωστες λέξεις μετά το 4ο βήμα

- 5ο Βήμα: Διαγραφή του σημείου στίξης «'» μπροστά από τις λέξεις.
  Τελική κάλυψη : Βρέθηκαν embeddings για το 56.92 % του λεξικού και για το 99.7 % των λέξεων στα σχόλια.

3.5 Προετοιμασία του συνόλου δεδομένων για εκπαίδευση

Για την εκπαίδευση χωρίζονται τα παραδείγματα εκπαίδευσης από το πλέον καθαρισμένο σύνολο δεδομένων εκπαίδευσης σε train set και validation set για να αποφύγουμε την υπερπροσαρμογή. Ο ποσοστιαίος διαχωρισμός γίνεται παίρνοντας το 80% του καθαρισμένου συνόλου εκπαίδευσης για το train set και το υπόλοιπο 20% για το validation set. Τα παραδείγματα στα καθαρισμένα σύνολα ελέγχου χρησιμοποιούνται για αξιολόγηση της απόδοσης του εκπαιδευμένου δικτύου και έχουν ονομασία public test set και private test set.

Για να τροφοδοτήσουμε τα μοντέλα με τα παραδείγματα εκπαίδευσης, επαλήθευσης και ελέγχου πρώτα πρέπει να μετατρέψουμε κάθε σχόλιο σε αλληλουχία αριθμών (sequence) και έπειτα συμπληρώνουμε με κάποια λεξικογραφική μονάδα (padding) ή κλάδεμα (truncate) μέχρι όλες οι αλληλουχίες να έχουν μέγεθος ίσο με MAX_LEN = 180. Στην συνέχεια όλα τα σύνολα δεδομένων μετατρέπονται σε tf. Dataset και ορίζεται το batch size σε 512 για τα μοντέλα που χρησιμοποιούν LSTM, GRU, CNN, ενώ για αυτά που χρησιμοποιούν transformers ορίζεται σε 64.

Για τα μοντέλα που χρησιμοποιούν LSTM, GRU, CNN το γέμισμα και το κλάδεμα γίνεται συμπληρώνοντας με μηδενικά τις αλληλουχίες ή αφαιρώντας λέξεις ξεκινώντας από την αρχή(pre padding). Για να μετατραπεί μια λέξη, που πλέον είναι ένας αριθμός, σε διάνυσμα λέξης (word vector) χρειαζόμαστε ένα επίπεδο Embedding όπου βάσει ενός embedding matrix θα κάνει την μετατροπή. Embedding matrix είναι ένας πίνακας πού έχει διανύσματα λέξεων για όλες τις λέξεις των σχολίων. Για να κατασκευαστεί

χρησιμοποιούνται τα προ εκπαιδευμένα διανύσματα λέξεων του συνόλου GloVe και του συνόλου από το FastText. Όπου υπάρχει μια άγνωστη λέξη για τα προ εκπαιδευμένα διανύσματα λέξεων το διάνυσμα της λέξης συμπληρώνεται με μηδενικά. Χρησιμοποιούνται και τα δύο προ εκπαιδευμένα διανύσματα λέξεων για μεγαλύτερη ακρίβεια στην αναπαράσταση μιας λέξης.

Για τα μοντέλα που χρησιμοποιούν τους transformers BERT και RoBERTa προστίθενται επιπλέον λεξικογραφικές μονάδες, όπως το CLS token και SEP token. Για παράδειγμα η πρόταση «The fox crossed the road but the chicken didn't» γίνεται:

- Για το BERT: [CLS] The fox crossed the road but the chicken didn't [SEP] και έπειτα γίνεται [101, 1109, 17594, 3809, 1103, 1812, 1133, 1103, 9323, 1238, 112, 189, 102]. Μετά προστίθενται τα μηδενικά για padding ή γίνεται κλάδεμα από το τέλος της αλληλουχίας και μπαίνει το token <UNK> για τις άγνωστες λέξεις.
- Για το RoBERTa: <s> The fox crossed the road but the chicken didn't </s> και έπειτα γίνεται [0, 20, 23602, 7344, 5, 921, 53, 5, 5884, 399, 75, 2]. Μετά προστίθενται άσσοι για padding ή γίνεται κλάδεμα από το τέλος της αλληλουχίας και μπαίνει το token <unk> για τις άγνωστες λέξεις.

Για το μοντέλο που χρησιμοποιεί το GPT2 ακολουθείτε παρόμοια διαδικασία, εκτός από την προσθήκη επιπλέον λεξικογραφικών μονάδων και του ότι το padding γίνεται με τον αριθμό 50256. Επίσης για το μοντέλα που χρησιμοποιούν BERT και RoBERTa δημιουργείται το διάνυσμα μάσκας που περιέχει άσσους για όλα τα token εκτός του token που χρησιμοποιήθηκε για padding που αυτό συμβολίζεται με μηδέν.

Επιπλέον ως είσοδος στα μοντέλα δίνονται και κάποια βάρη για κάθε παράδειγμα εκπαίδευσης και επαλήθευσης ώστε να δίνει ιδιαίτερη σημασία στα σχόλια που έχουν ταυτότητα. Ο υπολογισμός των βαρών ακολούθησε την εξής διαδικασία:

1. Αρχικοποίηση όλων των βαρών με 1.
2. Πρόσθεση μιας μονάδας για κάθε σχόλιο που έχει κάποια ταυτότητα.
3. Πρόσθεση δύο μονάδων για κάθε τοξικό σχόλιο που δεν έχει κάποια ταυτότητα.
4. Πρόσθεση δέκα μονάδων για κάθε μη τοξικό σχόλιο που έχει κάποια ταυτότητα.
5. Κανονικοποίηση διαιρώντας με το μέσο.

Επιλογή των τιμών που προστέθηκαν έγινε με βάση την λογική ότι δεν θέλουμε το μοντέλο να έχει προκατάληψη προς τις ταυτότητες των σχολίων. Αυτό αποτρέπει το μοντέλο από

το να προβλέπει ως τοξικά κάποια σχόλια που είναι μη τοξικά απλά και μόνο αναφέρουν κάποια ταυτότητα που εμφανίζεται συχνά δίπλα από υβριστικές ή χυδαίες λέξεις.

Τέλος για κάθε παράδειγμα εκπαίδευσης και επαλήθευσης χρησιμοποιείται η αντίστοιχη τιμή στην στήλη target ως πραγματική έξοδος του μοντέλου και οι αντίστοιχες τιμές στις στήλες target, severe_toxicity, obsene, identity_attack, insult, threat ως βοηθητικό διάνυσμα εξόδου. Οι τιμές είναι ένας πραγματικός αριθμός μεταξύ 0 και 1.

4 Εκπαίδευση

4.1 Μετρικές

Ορθότητα (Accuracy). Είναι ο λόγος του αριθμού των σωστών προβλέψεων προς το συνολικό αριθμό των προβλέψεων.

$$Accuracy = \frac{\#correct\ predictions}{\#predictions} = \frac{TP + TN}{TP + FP + TN + FN}$$

, όπου TP = True Positive και εννοούμε τις σωστές προβλέψεις για την κλάση 1 (τοξικό σχόλιο). Αντίστοιχα TN = True Negative, FP = False Positive, FN = False Negative. Δεν είναι καλή μετρική γιατί δεν έχουμε ισοδύναμες κλάσεις. Για αυτό λόγο χρησιμοποιείται ως συμπληρωματική.

Ανάκληση (Recall) ή True positive Rate ή Sensitivity. Είναι ο λόγος του αριθμού των σωστών προβλέψεων της θετικής κλάσης (τοξικό σχόλιο) προς το συνολικό αριθμό των παραδειγμάτων της θετικής κλάσης.

$$Recall = \frac{\#correct\ positive\ predictions}{\#\ positive\ samples} = \frac{TP}{TP + FN}$$

Ακρίβεια (Precision). Είναι ο λόγος του αριθμού των σωστών προβλέψεων της θετικής κλάσης (τοξικό σχόλιο) προς το συνολικό αριθμό των προβλέψεων της θετικής κλάσης.

$$Precision = \frac{\#correct\ positive\ predictions}{\#correct\ positive\ predictions + \#wrong\ positive\ predixtions} = \frac{TP}{TP + FP}$$

Επειδή οι κλάσεις είναι μη ισοδύναμες χρησιμοποιείται η σταθμισμένη μέση ακρίβεια.

F1 score είναι ο αρμονικός μέσος μεταξύ ακρίβειας και ανάκλασης.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Επειδή οι κλάσεις είναι μη ισοδύναμες χρησιμοποιείται το σταθμισμένο μέσο F1 score.

Η βασική μετρική που χρησιμοποιείται για την αξιολόγηση των μοντέλων, έχει αναπτυχθεί από τους διοργανωτές του διαγωνισμού από τον οποίο πήρα το σύνολο δεδομένων, και βασίζεται στην μετρική ROC-AUC [35]. Η μετρική ROC-AUC περιγράφει πόσο ικανό είναι ένα μοντέλο στον διαχωρισμό κλάσεων δεδομένου κάποιου threshold. Για το διαγωνισμό το threshold έχει οριστεί στο 0.5. Για τον υπολογισμό της απλής μετρικής ROC-AUC χρειάζεται το Recall ή TPR και το FPR (False Positive Rate). Για να μετρηθεί η ακούσια προκατάληψη του μοντέλου ξανά υπολογίζεται η ROC-AUC, αλλά αυτή την φορά υπολογίζεται σε 3 διαφορετικά υποσύνολα των συνόλων ελέγχου για κάθε ταυτότητα:

1. AUC υποομάδας (Subgroup AUC): Εδώ περιορίζονται τα δεδομένα μόνο σε αυτά που αναφέρουν την συγκεκριμένη ταυτότητα της υποομάδας. Μια χαμηλή τιμή σε αυτή την μετρική σημαίνει πως το μοντέλο δεν κάνει καλή δουλειά στον διαχωρισμό τοξικών και μη σχόλιων που αναφέρουν την ταυτότητα.
2. BPSN (Background Positive, Subgroup Negative) AUC: Εδώ περιορίζονται τα σύνολα ελέγχου σε υποσύνολα που περιέχουν μη τοξικά σχόλια που αναφέρουν συγκεκριμένη ταυτότητα και τοξικά σχόλια που δεν αναφέρουν. Μια χαμηλή τιμή σε αυτή την μετρική σημαίνει πως το μοντέλο προβλέπει υψηλότερο ποσοστό τοξικότητας από ότι θα έπρεπε για τα μη τοξικά σχόλια που αναφέρουν ταυτότητα.
3. BNSP (Background Negative, Subgroup Positive) AUC: Εδώ περιορίζονται τα σύνολα ελέγχου σε υποσύνολα που περιέχουν τοξικά σχόλια που αναφέρουν συγκεκριμένη ταυτότητα και μη τοξικά σχόλια που δεν αναφέρουν. Μια χαμηλή τιμή σε αυτή την μετρική σημαίνει πως το μοντέλο προβλέπει χαμηλότερο ποσοστό τοξικότητας από ότι θα έπρεπε για τα τοξικά σχόλια που αναφέρουν ταυτότητα.

Για να συνδυαστούν οι υπό μετρικές κάθε ταυτότητας υπολογίζεται ο μέσος τους ως εξής:

$$M_p(m_s) = \left(\frac{1}{N}\sum_{s=1}^{N} m_s^p\right)^{\frac{1}{p}}$$

Όπου m_s είναι η τιμή της μετρικής m για την ταυτότητα της υποομάδας s, N είναι ο αριθμός των υποομάδων με ταυτότητα και p μια σταθερά που ισούται με -5.
Για τον υπολογισμό της τελικής μετρικής συνδυάζεται το γενικό AUC με τους μέσους των 3 υπό μετρικών που ορίστηκαν παραπάνω:

$$score = w_0 AUC_{overral} + \sum_{a=1}^{A} w_a M_p(m_{s,a})$$

, όπου A είναι ο αριθμός των υπό μετρικών (3), $m_{s,a}$ είναι η μέση τιμή της υπό μετρικής α και w είναι τα βάρη για στάθμιση του αποτελέσματος ανάλογα με την σημαντικότητα κάθε μετρικής και τα τέσσερα βάρη έχουν τιμή 0.25.

4.2 Επαναληπτικό Νευρωνικό Δίκτυο με LSTM

Στην εκπαίδευση χρησιμοποιείται πρόωρη διακοπή (early stopping) για την αποφυγή υπερπροσαρμογής. Η πρόωρη διακοπή εφαρμόζεται παρακολουθώντας το σφάλμα στο σύνολο ελέγχου ξεκινώντας από την 2^{η} εποχή. Η διακοπή γίνεται όταν για 1 εποχή δεν μειωθεί το σφάλμα κάτω από το καλύτερο σφάλμα και επιστρέφουν τα βάρη στις τιμές που είχαν την εποχή που εμφανίστηκε το καλύτερο σφάλμα. Ο μέγιστος αριθμός εποχών είναι 10.

Για κάθε Νευρωνικό δίκτυο ο αλγόριθμος βελτιστοποίησης είναι ο Adam με τις προεπιλεγμένες τιμές του, δηλαδή learning_rate=0.001, beta_1=0.9, beta_2=0.999, amsgrad=False. Για κάθε Νευρωνικό δίκτυο η συνάρτηση σφάλματος είναι η binary-crossentropy.

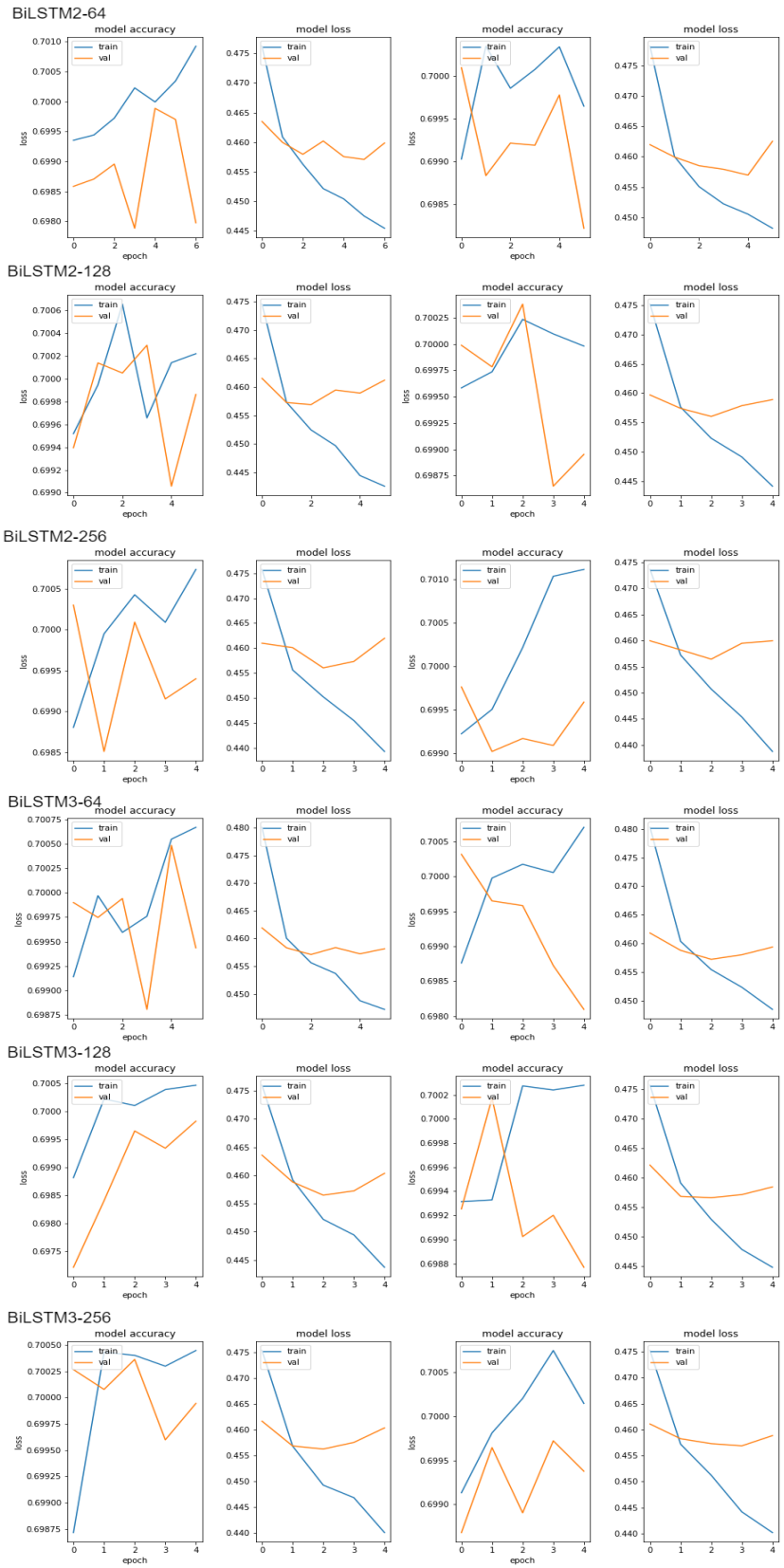
Μετά το επίπεδο για τα embeddings χρησιμοποιείται ένα επίπεδο απόρριψης (Drop out) με p = 0.2. Έπειτα χρησιμοποιούνται κρυφά επίπεδα με διπλή κατεύθυνση (bidirectional) και με Long Short-Term Memory (LSTM) ακολουθούμενα από συνενωμένα max pooling και average pooling. Στην συνέχεια χρησιμοποιούνται 2 κρυφά επίπεδα απλών νευρώνων, που έχουν συνάρτηση ενεργοποίησης ReLU, με skip-connections. Η τεχνική skip-connections βοηθάει στην σύγκλιση του μοντέλου, στην ουσία η είσοδος σε ένα επίπεδο προστίθεται με την έξοδο του και τροφοδοτούνται στο επόμενο επίπεδο. Στο τέλος υπάρχουν 2 επίπεδα εξόδου, με σιγμοειδή συνάρτηση ενεργοποίησης, το πρώτο είναι το βασικό που προβλέπει την τοξικότητα (1 νευρώνας) και το άλλο είναι βοηθητικό και προβλέπει τις υποκατηγορίες τοξικότητας (6 νευρώνες) και την τοξικότητα, δίνοντας την δυνατότητα στο δίκτυο να συγκλίνει πιο εύκολα.

Χρησιμοποιούνται οι default τιμές για LSTM: activation = 'tanh', recurrent_activation = 'sigmoid', use_bias = True, kernel_initializer = 'glorot_uniform', recurrent_initializer = 'orthogonal', bias_initializer = 'zeros'. Για να βρεθεί το βέλτιστο πλήθος επίπεδων LSTM, δοκιμάζονται διαφορετικές αρχιτεκτονικές για 2 επίπεδα και 3 επίπεδα. Για να βρεθεί το καλύτερο πλήθος νευρώνων για κάθε αρχιτεκτονική με διαφορετικό πλήθος επιπέδων, εκπαιδεύονται μοντέλα με αρχιτεκτονικές για 64, 128 και 256 νευρώνες ανά επίπεδο. Το πλήθος των νευρώνων σε κάθε επίπεδο LSTM είναι το ίδιο. Το πλήθος των νευρώνων για τα 2 επίπεδα απλών νευρώνων είναι όσο είναι για τα επίπεδα LSTM πολλαπλασιασμένο με το 4, για εφαρμοστεί η τεχνική skip-connections. Κάθε δίκτυο με LSTM παίρνει το όνομα BiLSTMx-mm. Όπου x είναι ο αριθμός των κρυφών επιπέδων LSTM και mm είναι το πλήθος των νευρώνων τους. Για την μείωση τυχαιότητας αποτελεσμάτων κάθε αρχιτεκτονική με LSTM εκπαιδεύτηκε 2 φορές, και χρησιμοποιήθηκε ο μέσος όρος των προβλέψεων για τα σύνολα ελέγχου.

Model	Acc. Public	W. Pre. Public	Rec. Public	W. F1 Public	Public AUC	W. Acc. Private	W. Pre. Private	Rec. Private	W. F1 Private	Private AUC
BiLSTM2-64	0.9526	0.9491	0.5989	0.9501	0.9327	0.9532	0.9500	0.6128	0.9509	0.9343
BiLSTM2-128	0.9522	0.9503	0.6484	0.9511	0.9339	0.9524	0.9505	0.6560	0.9513	0.9333
BiLSTM2-256	0.9527	0.9504	0.6381	0.9513	0.9325	0.9528	0.9506	0.6461	0.9515	0.9335
BiLSTM3-64	0.9538	0.9505	0.6101	0.9515	0.9333	0.9540	0.9508	0.6175	0.9517	0.9339
BiLSTM3-128	0.9525	0.9496	0.6205	0.9506	0.9335	0.953	0.9504	0.6335	0.9513	0.9339
BiLSTM3-256	0.9533	0.9504	0.6206	0.9514	0.9340	0.9536	0.9507	0.6270	0.9517	0.9337

Πίνακας 4-1: Αποτελέσματα LSTM στα σύνολα ελέγχου

Σύμφωνα με τον παραπάνω πίνακα καλύτερο Public AUC με 0.9340 έχει το μοντέλο BiLSTM3-256, ενώ καλύτερο Private AUC με 0.9343 έχει το μοντέλο BiLSTM2-64. Τελικά το μοντέλο όπου θα αντιπροσωπεύσει τα νευρωνικά δίκτυα με LSTM στην σύγκριση με τα υπόλοιπα είναι το BiLSTM2-64, γιατί ο διαγωνισμός έχει δημοσιευμένο μόνο το βαθμολογικό πίνακα για το private σύνολο ελέγχου και θέλουμε να συγκριθεί και με εκείνα τα μοντέλα.



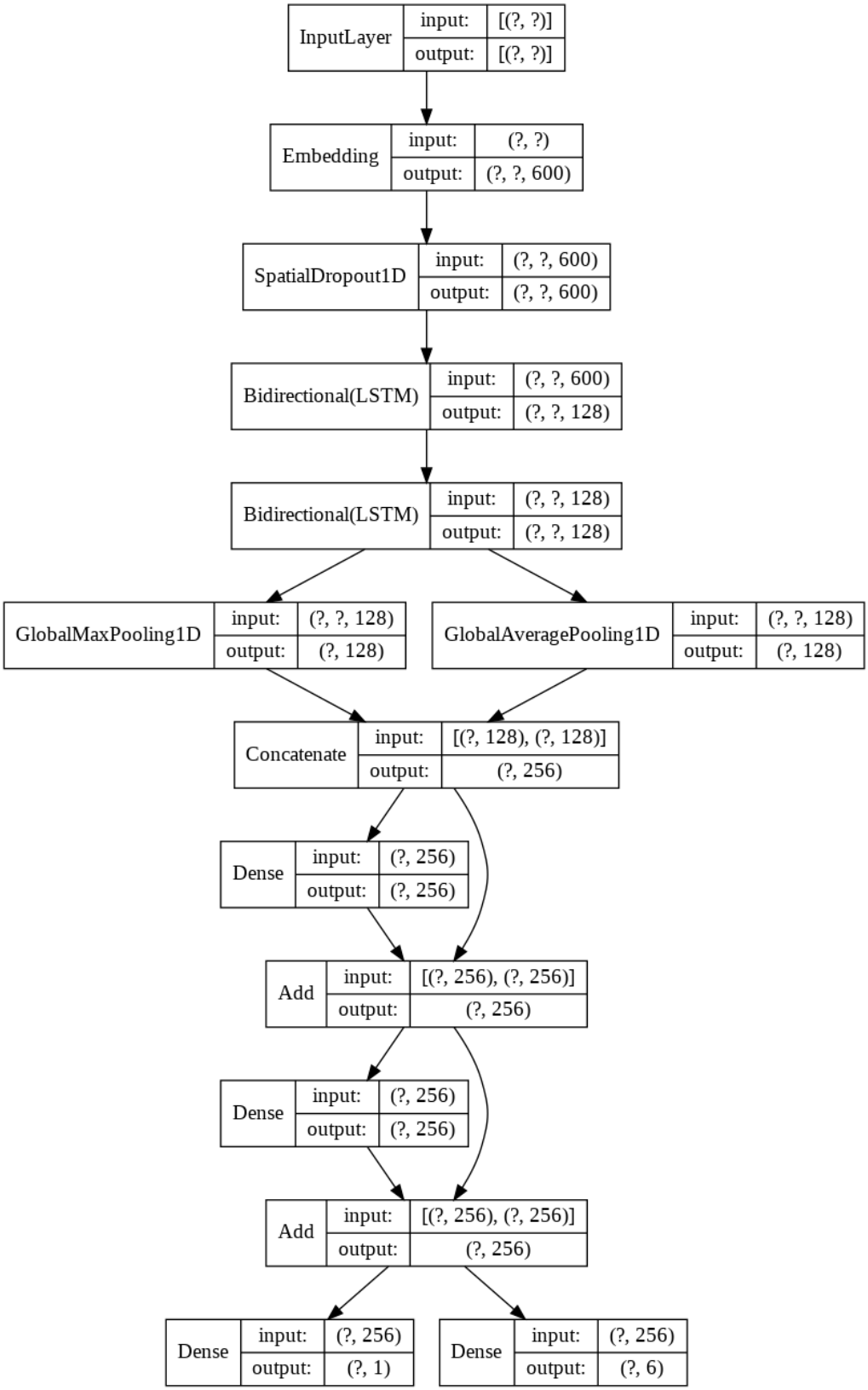
Εικόνα 4-1: Γραφικές παραστάσεις accuracy και loss κατά την εκπαίδευση για Νευρωνικα Δίκτυα με LSTM

subgroup	subgroup_auc	bpsn_auc	bnsp_auc
black	0.8383	0.9008	0.9485
white	0.8614	0.9105	0.9505
male	0.9243	0.9566	0.9456
female	0.9294	0.9614	0.9426
christian	0.9413	0.9672	0.9384
jewish	0.9242	0.9486	0.9482
muslim	0.8722	0.9296	0.9427
psychiatric_or_mental_illness	0.8936	0.9318	0.9490
homosexual_gay_or_lesbian	0.8305	0.9161	0.9378
Mp	0.8853	0.9342	0.9448

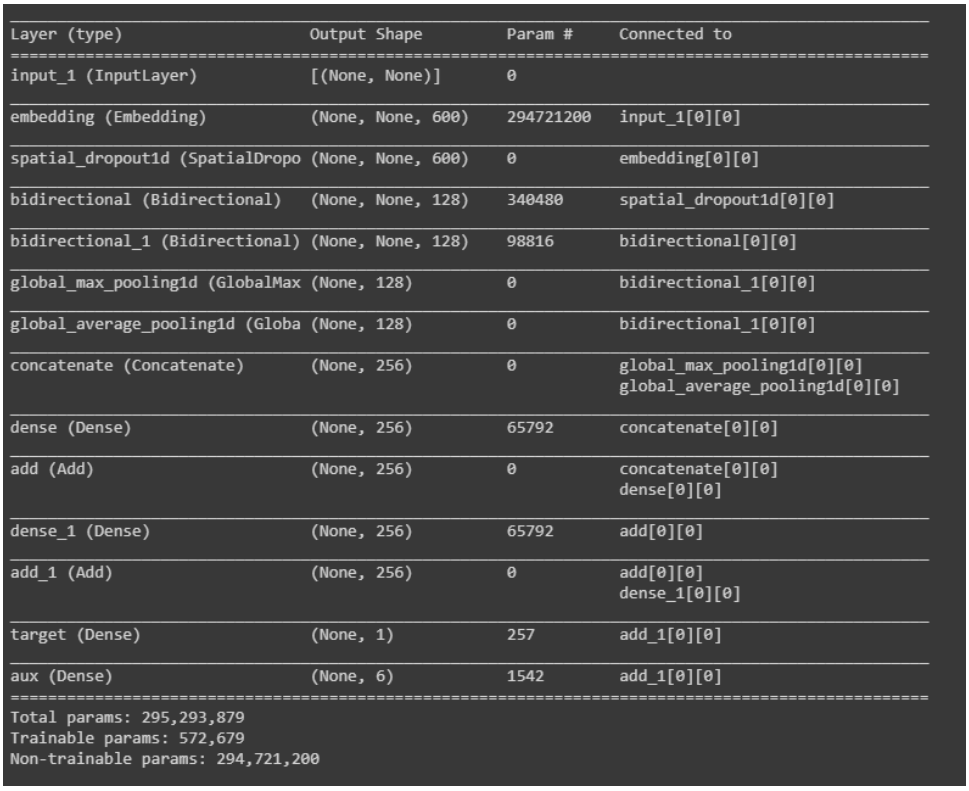
Πίνακας 4-2: Υπό μετρικές για το BiLSTM2-64 στο public σύνολο ελέγχου.

subgroup	subgroup_auc	bpsn_auc	bnsp_auc
black	0.8385	0.9153	0.9441
white	0.8469	0.9148	0.9473
male	0.9267	0.9577	0.9472
female	0.9336	0.9675	0.94
christian	0.941	0.9699	0.9392
jewish	0.9153	0.9461	0.948
muslim	0.8463	0.9357	0.9347
psychiatric_or_mental_illness	0.9324	0.9468	0.9600
homosexual_gay_or_lesbian	0.8414	0.9245	0.9376
Mp	0.8849	0.9408	0.9441

Πίνακας 4-3: Υπό μετρικές για το BiLSTM2-64 στο private σύνολο ελέγχου



Εικόνα 4-2: Γράφος του BiLSTM2-64



Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, None)]	0	
embedding (Embedding)	(None, None, 600)	294721200	input_1[0][0]
spatial_dropout1d (SpatialDropo	(None, None, 600)	0	embedding[0][0]
bidirectional (Bidirectional)	(None, None, 128)	340480	spatial_dropout1d[0][0]
bidirectional_1 (Bidirectional)	(None, None, 128)	98816	bidirectional[0][0]
global_max_pooling1d (GlobalMax	(None, 128)	0	bidirectional_1[0][0]
global_average_pooling1d (Globa	(None, 128)	0	bidirectional_1[0][0]
concatenate (Concatenate)	(None, 256)	0	global_max_pooling1d[0][0] global_average_pooling1d[0][0]
dense (Dense)	(None, 256)	65792	concatenate[0][0]
add (Add)	(None, 256)	0	concatenate[0][0] dense[0][0]
dense_1 (Dense)	(None, 256)	65792	add[0][0]
add_1 (Add)	(None, 256)	0	add[0][0] dense_1[0][0]
target (Dense)	(None, 1)	257	add_1[0][0]
aux (Dense)	(None, 6)	1542	add_1[0][0]

Total params: 295,293,879
Trainable params: 572,679
Non-trainable params: 294,721,200

Εικόνα 4-3: Περίληψη του BiLSTM2-64

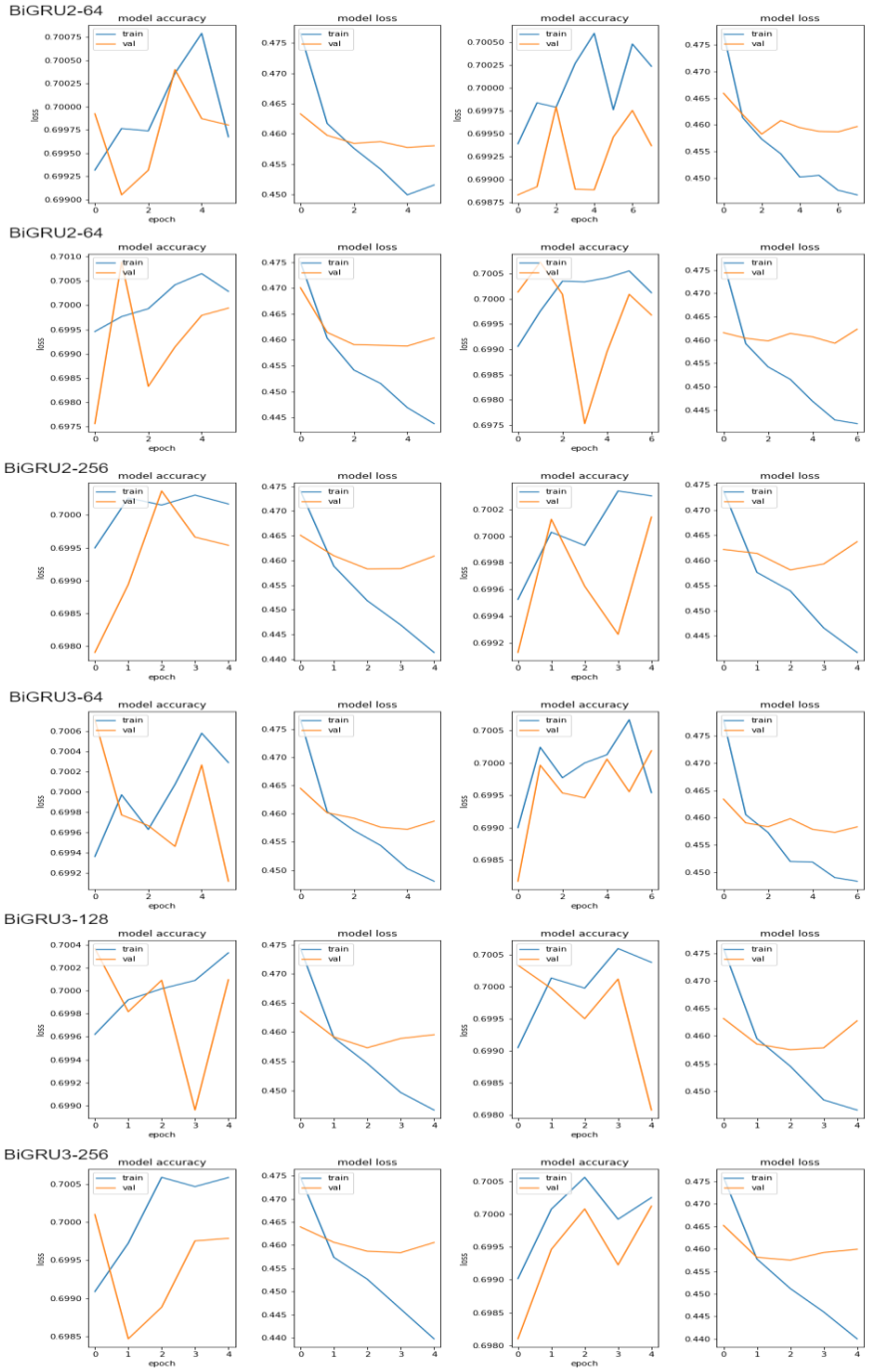
4.3 Επαναληπτικό Νευρωνικό Δίκτυο με GRU

Ακολουθείτε η ίδια διαδικασία με την προηγούμενη ενότητα μόνο που τώρα αντί για επίπεδα LSTM, χρησιμοποιούμε επίπεδα GRU. Άρα κάθε δίκτυο παίρνει το όνομα BiGRUx-mm. Όπου x είναι ο αριθμός των κρυφών επιπέδων GRU και mm είναι το πλήθος των νευρώνων τους.

Model	Acc. Public	W. Pre. Public	Rec. Public	W. F1 Public	Public AUC	W. Acc. Private	W. Pre. Private	Rec. Private	W. F1 Private	Private AUC
BiGRU2-64	0.9529	0.9493	0.5912	0.9502	0.9318	0.9528	0.9492	0.5938	0.9501	0.9338
BiGRU2-128	0.9533	0.9496	0.5835	0.9504	0.9317	0.9537	0.9501	0.5928	0.9509	0.9333
BiGRU2-256	0.9525	0.9495	0.6141	0.9505	0.9323	0.9532	0.9503	0.6230	0.9512	0.9337
BiGRU3-64	0.9533	0.9497	0.5937	0.9506	0.9339	0.9531	0.9496	0.5989	0.9505	0.9331
BiGRU3-128	0.9529	0.9498	0.6128	0.9508	0.9317	0.9528	0.9498	0.6190	0.9508	0.9338
BiGRU3-256	0.9531	0.9500	0.6130	0.9509	0.9332	0.9531	0.9501	0.6239	0.9511	0.9332

Πίνακας 4-4: Αποτελέσματα GRU στα σύνολα ελέγχου

Σύμφωνα με τον παραπάνω πίνακα καλύτερο Private AUC με 0.9338 έχουν τα μοντέλα BiGRU2-64 και BiGRU3-128, προτιμάτε το BiGRU2-64, γιατί έχει καλύτερο Public AUC. Τελικά το μοντέλο όπου θα αντιπροσωπεύσει τα νευρωνικά δίκτυα με GRU στην σύγκριση με τα υπόλοιπα είναι το BiGRU2-64.



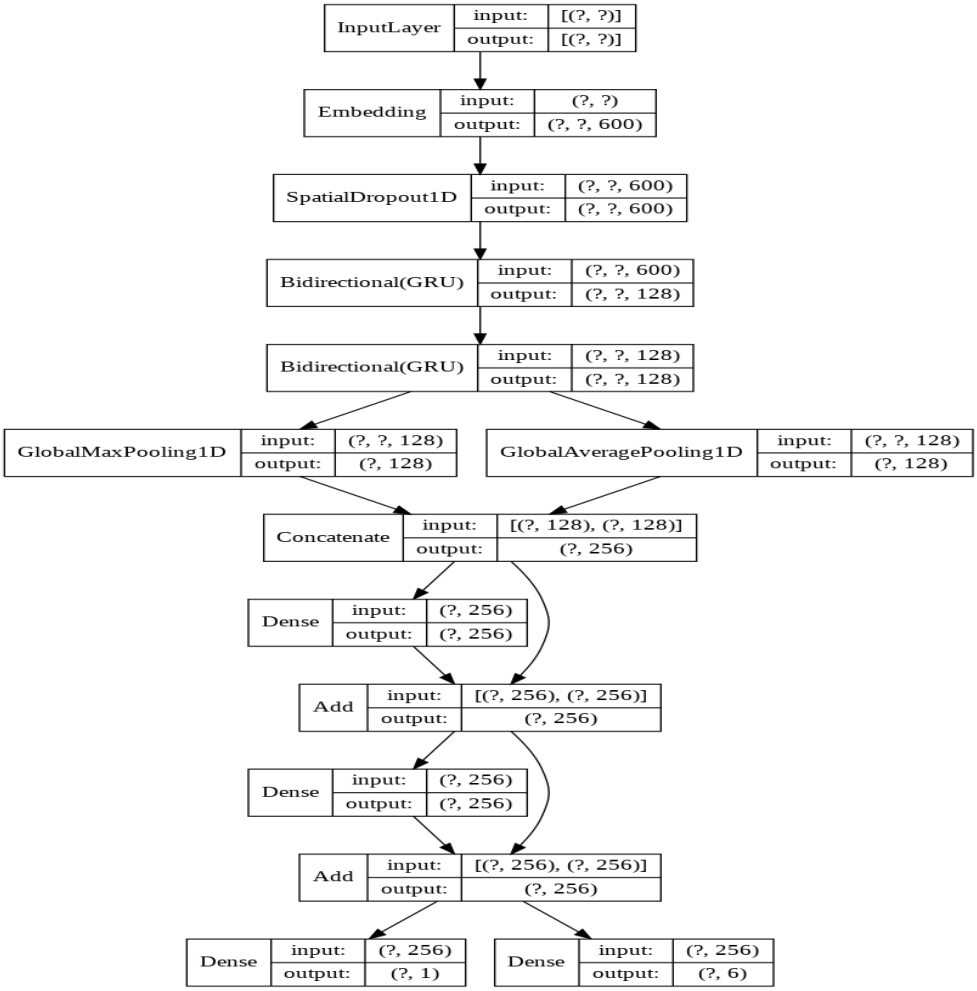
Εικόνα 4-4: Γραφικές παραστάσεις accuracy και loss κατά την εκπαίδευση για Νευρωνικα Δίκτυα με GRU

subgroup	subgroup_auc	bpsn_auc	bnsp_auc
black	0.8453	0.8968	0.9533
white	0.8712	0.9101	0.9537
male	0.9238	0.9579	0.9454
female	0.9272	0.9624	0.942
christian	0.9444	0.9702	0.9379
jewish	0.9283	0.9474	0.9507
muslim	0.882	0.9312	0.9457
psychiatric_or_mental_illness	0.8900	0.9319	0.9485
homosexual_gay_or_lesbian	0.8303	0.9076	0.9435
Mp	0.8887	0.9331	0.9467

Πίνακας 4-5: Υπό μετρικές για το BiGRU2-64 στο public σύνολο ελέγχου

subgroup	subgroup_auc	bpsn_auc	bnsp_auc
black	0.8376	0.9119	0.9464
white	0.8498	0.914	0.9493
male	0.9265	0.9579	0.9477
female	0.9342	0.9682	0.9401
christian	0.9397	0.9716	0.9362
jewish	0.9076	0.9458	0.9435
muslim	0.8474	0.9356	0.9341
psychiatric_or_mental_illness	0.9295	0.9443	0.9609
homosexual_gay_or_lesbian	0.8276	0.9115	0.9423
Mp	0.8822	0.9385	0.9443

Πίνακας 4-6: Υπό μετρικές για το BiGRU2-64 στο private σύνολο ελέγχου



Εικόνα 4-5: Γράφος του BiGRU2-64

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, None)]	0	
embedding_1 (Embedding)	(None, None, 600)	21336000	input_2[0][0]
spatial_dropout1d_1 (SpatialDro	(None, None, 600)	0	embedding_1[0][0]
bidirectional_2 (Bidirectional)	(None, None, 128)	255744	spatial_dropout1d_1[0][0]
bidirectional_3 (Bidirectional)	(None, None, 128)	74496	bidirectional_2[0][0]
global_max_pooling1d_1 (GlobalM	(None, 128)	0	bidirectional_3[0][0]
global_average_pooling1d_1 (Glo	(None, 128)	0	bidirectional_3[0][0]
concatenate_1 (Concatenate)	(None, 256)	0	global_max_pooling1d_1[0][0] global_average_pooling1d_1[0][0]
dense_2 (Dense)	(None, 256)	65792	concatenate_1[0][0]
add_2 (Add)	(None, 256)	0	concatenate_1[0][0] dense_2[0][0]
dense_3 (Dense)	(None, 256)	65792	add_2[0][0]
add_3 (Add)	(None, 256)	0	add_2[0][0] dense_3[0][0]
target (Dense)	(None, 1)	257	add_3[0][0]
aux (Dense)	(None, 6)	1542	add_3[0][0]

Total params: 21,799,623
Trainable params: 463,623
Non-trainable params: 21,336,000

Εικόνα 4-6: Περίληψη του BiGRU2-64

4.4 Συνελικτικό Νευρωνικό Δίκτυο

Χρησιμοποιείται πρόωρη διακοπή με την ίδια διαδικασία που έγινε στις προηγούμενες ενότητες, καθώς και ως αλγόριθμος βελτιστοποίησης ο ίδιος Adam με τις προεπιλεγμένες τιμές του.

Λόγω των μειωμένων υπολογιστικών πόρων και επειδή δεν αναμένονται τα καλύτερα αποτελέσματα χρησιμοποιώντας Συνελικτικό Δίκτυο, για αυτού του είδους τα δίκτυα εκπαιδεύεται μόνο ένα μοντέλο που θα ονομαστεί TextCNNbase.

Μετά το επίπεδο για τα embeddings χρησιμοποιούνται 5 παράλληλα επίπεδα συνέλιξης το $1^ο$ έχει 128 φίλτρα με μέγεθος 2, το $2^ο$ 128 φίλτρα με μέγεθος 3, το $3^ο$ 128 φίλτρα με μέγεθος 4, το $4^ο$ 128 φίλτρα με μέγεθος 5 και το $5^ο$ 128 φίλτρα με μέγεθος 6. Για όλα τα φίλτρα το μέγεθος διασκελισμού είναι ίσο με 1, δεν γίνεται γέμισμα με μηδενικά και υπάρχει μόνο ένα κανάλι. Επίσης χρησιμοποιείται ως συνάρτηση ενεργοποίησης η ReLU.

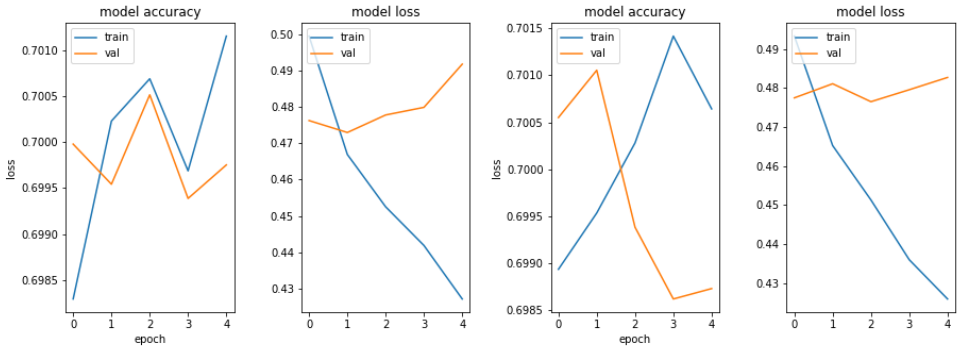
Στην συνέχεια γίνεται max pooling σε κάθε έξοδο από τα επίπεδα συνέλιξης, ώστε να μειωθούν οι διαστάσεις των εξόδων από 3 σε 2. Οι έξοδοι από τα max pooling συνενώνονται και τροφοδοτούνται σε ένα κρυφό επίπεδο απλών νευρώνων.

Το κρυφό επίπεδο απλών νευρώνων έχει 128 νευρώνες, που χρησιμοποιούν ReLU. Έπειτα γίνεται απόρριψη (Dropout) με p = 0.2. Στο τέλος υπάρχουν 2 επίπεδα εξόδου, με σιγμοειδή συνάρτηση ενεργοποίησης, το πρώτο είναι το βασικό που προβλέπει την τοξικότητα (1 νευρώνας) και το άλλο είναι βοηθητικό και προβλέπει τις υποκατηγορίες τοξικότητας (6 νευρώνες) και την τοξικότητα, δίνοντας την δυνατότητα στο δίκτυο να συγκλίνει πιο εύκολα.

Model	Acc. Public	W. Pre. Public	Rec. Public	W. F1 Public	Public AUC	W. Acc. Private	W. Pre. Private	Rec. Private	W. F1 Private	Private AUC
TextCNNbase	0.9485	0.9445	0.568	0.9457	0.9126	0.9487	0.9447	0.5732	0.9459	0.9161

Πίνακας 4-7: Αποτελέσματα TextCNN στα σύνολα ελέγχου

Από τον παραπάνω πίνακα φαίνεται πως το μοντέλο TextCNNbase ανταποκρίνεται καλύτερα στο private σύνολο ελέγχου αλλά όχι με τόση μεγάλη διαφορά ώστε να μπορούμε να πούμε πως τα δύο σύνολα ελέγχου έχουν διαφορετικό πεδίο ορισμού.



Εικόνα 4-7: Γραφικές παραστάσεις accuracy και loss κατά την εκπαίδευση για TextCNN

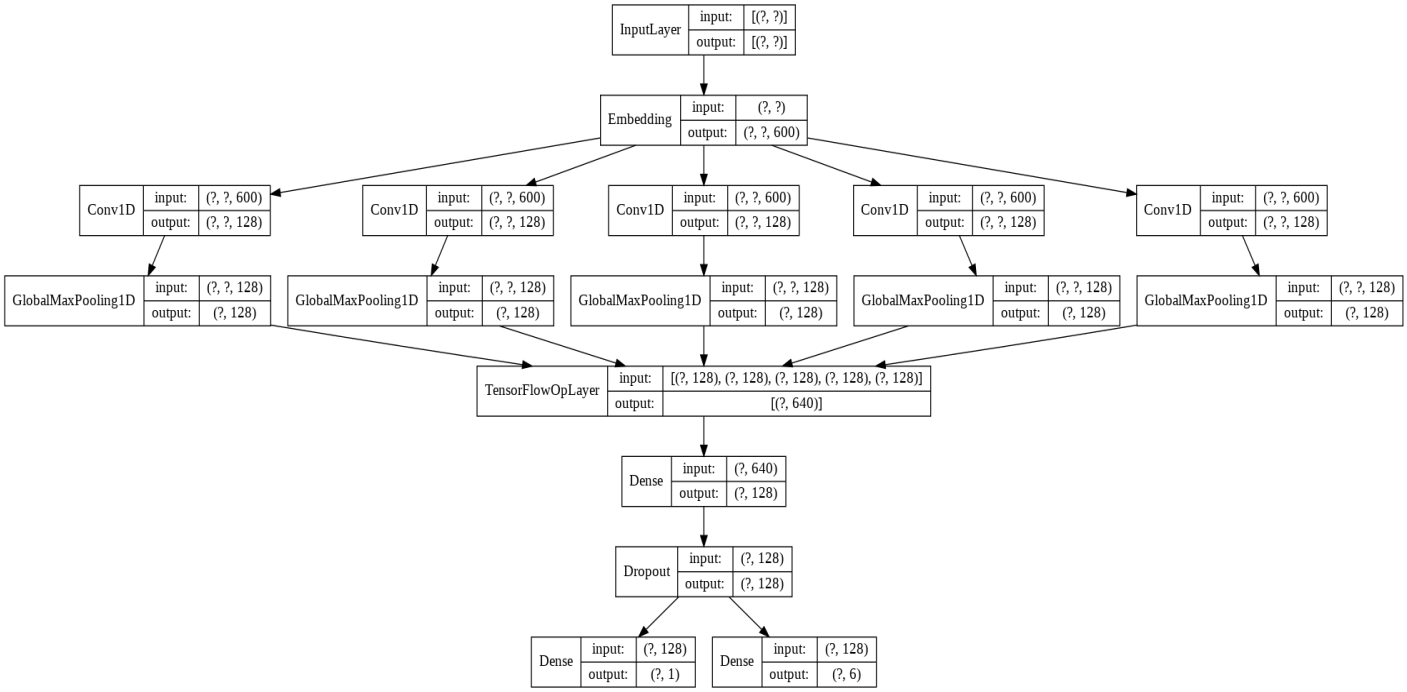
subgroup	subgroup_auc	bpsn_auc	bnsp_auc
black	0.7927	0.8647	0.9398
white	0.834	0.877	0.9454
male	0.9037	0.9323	0.9401
female	0.9069	0.9354	0.9393
christian	0.9197	0.9497	0.9298
jewish	0.8919	0.9244	0.934
muslim	0.8459	0.9064	0.9335
psychiatric_or_mental_illness	0.8425	0.9065	0.9283
homosexual_gay_or_lesbian	0.8039	0.8708	0.9378
black	0.7927	0.8647	0.9398
Mp	0.8533	0.9047	0.9364

Πίνακας 4-8: Υπό μετρικές για το TextCNNbase στο public σύνολο ελέγχου

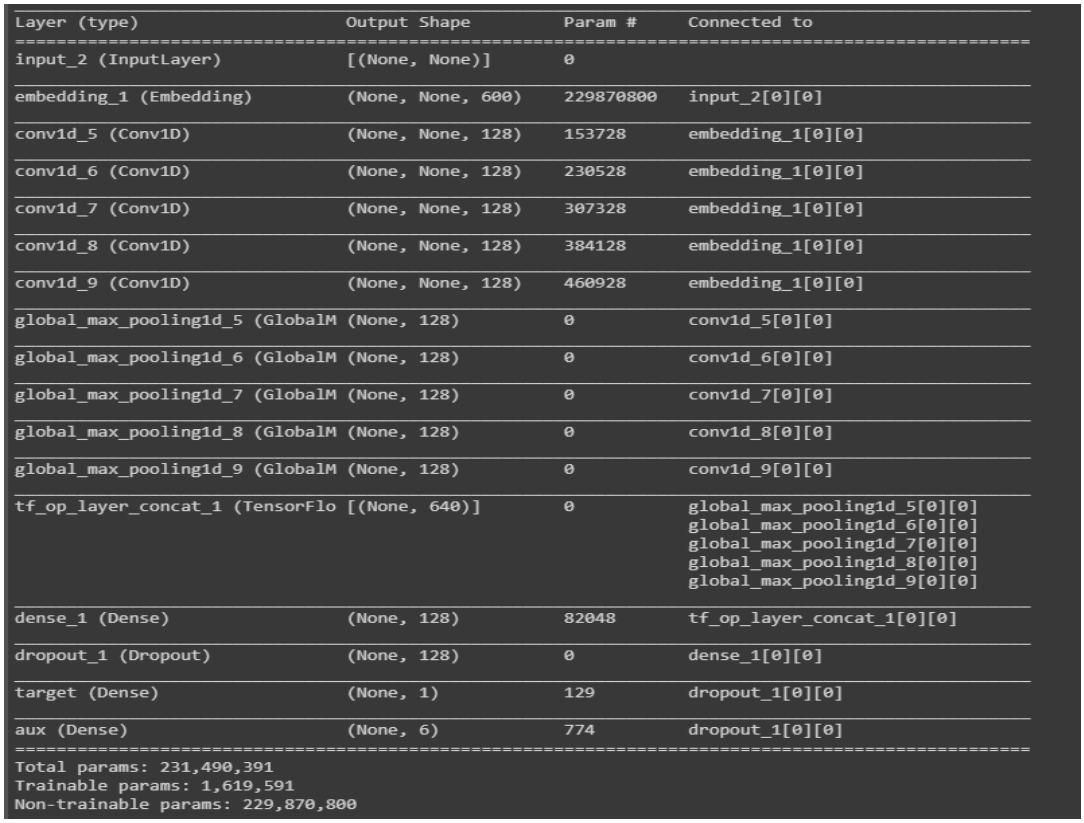
subgroup	subgroup_auc	bpsn_auc	bnsp_auc
black	0.8082	0.8803	0.9393
white	0.8118	0.8754	0.9431
male	0.9105	0.934	0.9448
female	0.9227	0.9442	0.9416
christian	0.9159	0.9531	0.9275
jewish	0.8825	0.9287	0.9304

muslim	0.8157	0.911	0.9237
psychiatric_or_mental_illness	0.9184	0.921	0.9558
homosexual_gay_or_lesbian	0.8115	0.8785	0.9398
Mp	0.8577	0.9114	0.9382

Πίνακας 4-9: Υπό μετρικές για το TextCNNbase στο private σύνολο ελέγχου



Εικόνα 4-8: Γράφος το TextCNNbase



Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, None)]	0	
embedding_1 (Embedding)	(None, None, 600)	229870800	input_2[0][0]
conv1d_5 (Conv1D)	(None, None, 128)	153728	embedding_1[0][0]
conv1d_6 (Conv1D)	(None, None, 128)	230528	embedding_1[0][0]
conv1d_7 (Conv1D)	(None, None, 128)	307328	embedding_1[0][0]
conv1d_8 (Conv1D)	(None, None, 128)	384128	embedding_1[0][0]
conv1d_9 (Conv1D)	(None, None, 128)	460928	embedding_1[0][0]
global_max_pooling1d_5 (GlobalM	(None, 128)	0	conv1d_5[0][0]
global_max_pooling1d_6 (GlobalM	(None, 128)	0	conv1d_6[0][0]
global_max_pooling1d_7 (GlobalM	(None, 128)	0	conv1d_7[0][0]
global_max_pooling1d_8 (GlobalM	(None, 128)	0	conv1d_8[0][0]
global_max_pooling1d_9 (GlobalM	(None, 128)	0	conv1d_9[0][0]
tf_op_layer_concat_1 (TensorFlo	[(None, 640)]	0	global_max_pooling1d_5[0][0] global_max_pooling1d_6[0][0] global_max_pooling1d_7[0][0] global_max_pooling1d_8[0][0] global_max_pooling1d_9[0][0]
dense_1 (Dense)	(None, 128)	82048	tf_op_layer_concat_1[0][0]
dropout_1 (Dropout)	(None, 128)	0	dense_1[0][0]
target (Dense)	(None, 1)	129	dropout_1[0][0]
aux (Dense)	(None, 6)	774	dropout_1[0][0]

Total params: 231,490,391
Trainable params: 1,619,591
Non-trainable params: 229,870,800

Εικόνα 4-9: Περίληψη του TextCNNbase

4.5 Νευρωνικό Δίκτυο με BERT

Για την κατασκευή του δικτύου χρησιμοποιήθηκε η βασική εκδοχή του BERT με 12 επίπεδα encoders. Έπειτα γίνεται average και max pooling στην έξοδο του BERT και τα αποτελέσματα συνενώνονται. Στην συνέχεια γίνεται απόρριψη (Dropout) με p=0.2 και στο τέλος υπάρχουν 2 επίπεδα εξόδου, με σιγμοειδή συνάρτηση ενεργοποίησης, το πρώτο είναι το βασικό που προβλέπει την τοξικότητα (1 νευρώνας) και το άλλο είναι βοηθητικό και προβλέπει τις υποκατηγορίες τοξικότητας (6 νευρώνες) και την τοξικότητα, δίνοντας την δυνατότητα στο δίκτυο να συγκλίνει πιο εύκολα Ως αλγόριθμος βελτιστοποίησης επιλέγεται ο Adam η μόνη διαφορά από τις προηγούμενες ενότητες είναι ο ρυθμός μάθησης που γίνεται 2e-5 ή 0.00002. Το δίκτυο εκπαιδεύεται για 4 εποχές σε κάθε τέλος εποχής γίνεται αξιολόγηση στο σύνολο επαλήθευσης και στα σύνολα ελέγχου και αποθηκεύονται οι παράμετροι. Επιλέγονται ως τελικοί παράμετροι του δικτύου εκείνοι που είχαν το καλύτερο ROC-AUC στο private σύνολο ελέγχου.

Model BERTwPool	Acc. Public	W. Pre. Public	Rec. Public	W. F1 Public	Public AUC	W. Acc. Private	W. Pre. Private	Rec. Private	W. F1 Private	Private AUC
Epoch 1	0.9517	0.9476	0.4842	0.9457	0.9294	0.9521	0.9482	0.4952	0.9463	0.9308
Epoch 2	0.9534	0.9494	0.5204	0.9485	0.9296	0.9532	0.9493	0.5223	0.9483	0.9303
Epoch 3	0.9523	0.9503	0.6459	0.9511	0.9275	0.9519	0.95	0.65	0.9508	0.9282
Epoch 4	0.9536	0.9501	0.5965	0.951	0.9278	0.954	0.9506	0.6055	0.9514	0.9274

Πίνακας 4-10: Αποτελέσματα BERTwPool στα σύνολα ελέγχου

Σύμφωνα με τον παραπάνω πίνακα καλύτερο Private AUC με 0.9308 έχει το μοντέλο της πρώτης εποχής, όπου αυτό επιλέγεται για σύγκριση με τα υπόλοιπα.

subgroup	subgroup_auc	bpsn_auc	bnsp_auc
black	0.8408	0.9106	0.9408
white	0.8499	0.9158	0.9416
male	0.9198	0.9596	0.9358
female	0.916	0.9655	0.9244
christian	0.9375	0.9682	0.9304
jewish	0.924	0.9445	0.9481
muslim	0.8651	0.9323	0.9349
psychiatric_or_mental_illness	0.8824	0.9386	0.9347
homosexual_gay_or_lesbian	0.8298	0.9211	0.9277
Mp	0.8801	0.9383	0.9352

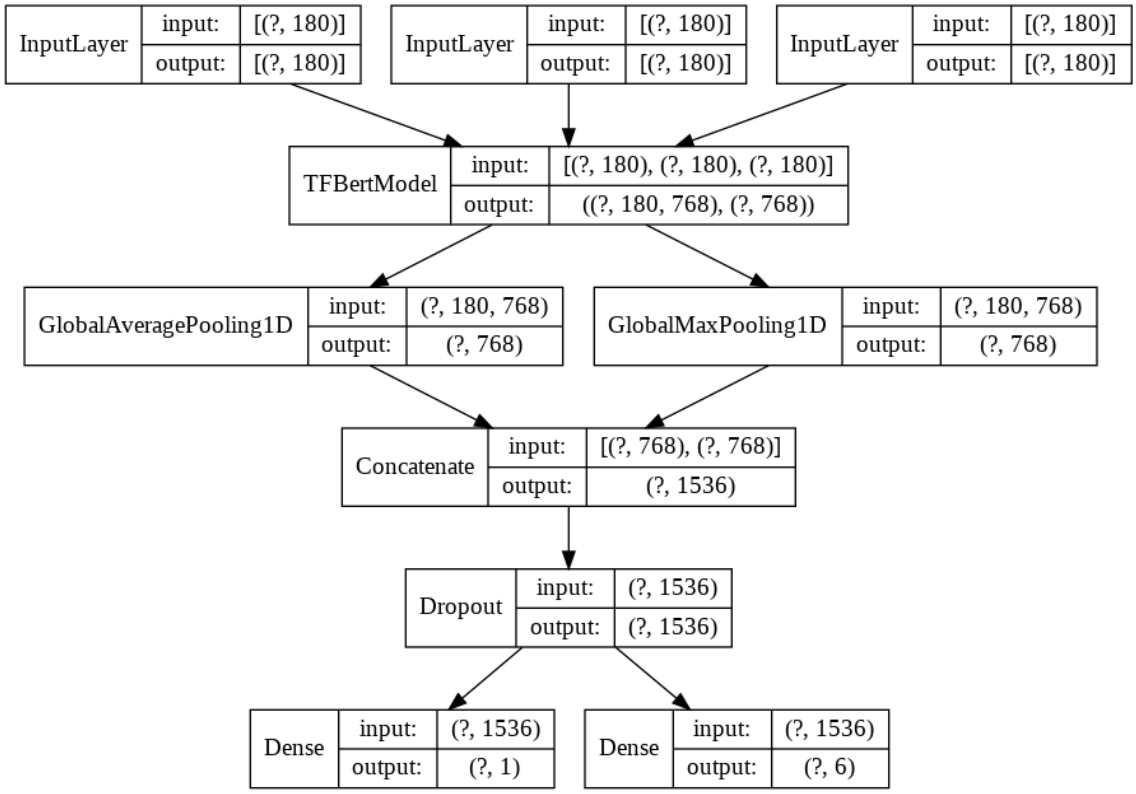
Πίνακας 4-11: Υπό μετρικές για το BERTwPool στο public σύνολο ελέγχου

subgroup	subgroup_auc	bpsn_auc	bnsp_auc
black	0.8479	0.925	0.935
white	0.8483	0.919	0.9411
male	0.9221	0.9598	0.9378
female	0.9295	0.9713	0.9257
christian	0.9293	0.969	0.9281
jewish	0.8943	0.9383	0.9421
muslim	0.8382	0.9344	0.9256
psychiatric_or_mental_illness	0.9514	0.9535	0.9595
homosexual_gay_or_lesbian	0.8212	0.9254	0.9261
Mp	0.8798	0.9429	0.9353

Πίνακας 4-12: Υπό μετρικές για το BERTwPool στο private σύνολο ελέγχου

```
Layer (type)                 Output Shape              Param #     Connected to
==================================================================================================
input_word_ids (InputLayer)  [(None, 180)]             0
__________________________________________________________________________________________________
input_mask (InputLayer)      [(None, 180)]             0
__________________________________________________________________________________________________
segment_ids (InputLayer)     [(None, 180)]             0
__________________________________________________________________________________________________
tf_bert_model_1 (TFBertModel)  ((None, 180, 768), (  108310272   input_word_ids[0][0]
                                                                 input_mask[0][0]
                                                                 segment_ids[0][0]
__________________________________________________________________________________________________
global_average_pooling1d_1 (Glo (None, 768)           0           tf_bert_model_1[0][0]
__________________________________________________________________________________________________
global_max_pooling1d_1 (GlobalM (None, 768)           0           tf_bert_model_1[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)  (None, 1536)              0           global_average_pooling1d_1[0][0]
                                                                 global_max_pooling1d_1[0][0]
__________________________________________________________________________________________________
dropout_75 (Dropout)         (None, 1536)              0           concatenate_1[0][0]
__________________________________________________________________________________________________
target (Dense)               (None, 1)                 1537        dropout_75[0][0]
__________________________________________________________________________________________________
aux (Dense)                  (None, 6)                 9222        dropout_75[0][0]
==================================================================================================
Total params: 108,321,031
Trainable params: 108,321,031
Non-trainable params: 0
```

Εικόνα 4-10: Περίληψη του BERTwPool



Εικόνα 4-11: Γράφος του BERTwPool

4.6 Νευρωνικό Δίκτυο με RoBERTa

Για την κατασκευή του δικτύου χρησιμοποιήθηκε η βασική εκδοχή του RoBERTa με 12 επίπεδα encoders. Αυτήν είναι η μόνη αλλαγή σε σχέση με την αρχιτεκτονική του προηγούμενου μοντέλου. Η εξαγωγή αποτελεσμάτων έγινε όπως στην προηγούμενη ενότητα.

Model RoBERTawPool	Acc. Public	W. Pre. Public	Rec. Public	W. F1 Public	Public AUC	W. Acc. Private	W. Pre. Private	Rec. Private	W. F1 Private	Private AUC
Epoch 1	0.9411	0.9492	0.7612	0.9443	0.9328	0.9416	0.9492	0.7620	0.9446	0.9336
Epoch 2	0.9452	0.9501	0.7416	0.9472	0.935	0.9446	0.9497	0.7442	0.9467	0.9362
Epoch 3	0.9496	0.9508	0.7069	0.9502	0.9329	0.9500	0.9513	0.7148	0.9506	0.9337
Epoch 4	0.9500	0.9514	0.7116	0.9506	0.9321	0.9496	0.9510	0.7124	0.9503	0.9317

Πίνακας 4-13: Αποτελέσματα RoBERTawPool στα σύνολα ελέγχου

Σύμφωνα με τον παραπάνω πίνακα καλύτερο Private AUC με 0.9362 έχει το μοντέλο της δεύτερης εποχής, όπου αυτό επιλέγεται για σύγκριση με τα υπόλοιπα.

subgroup	subgroup_auc	bpsn_auc	bnsp_auc
black	0.845	0.9105	0.9491
white	0.8602	0.9134	0.9516
male	0.9244	0.9559	0.9502
female	0.9222	0.9596	0.9446
christian	0.9441	0.9668	0.9443
jewish	0.9207	0.9457	0.9519
muslim	0.8798	0.9287	0.9466
psychiatric_or_mental_illness	0.8841	0.9395	0.9418
homosexual_gay_or_lesbian	0.8475	0.9263	0.9385
Mp	0.8879	0.9373	0.9464

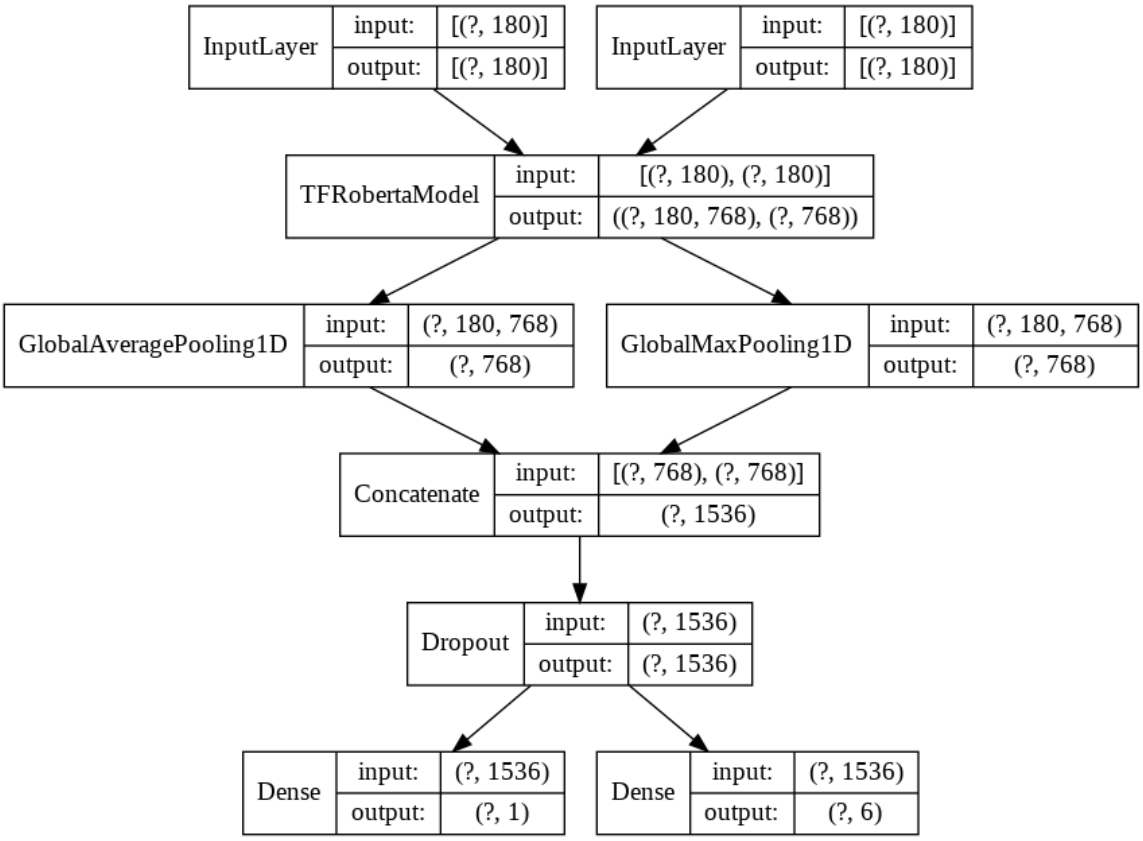
Πίνακας 4-14: Υπό μετρικές για το RoBERTawPool στο public σύνολο ελέγχου

subgroup	subgroup_auc	bpsn_auc	bnsp_auc
black	0.8515	0.9211	0.9445
white	0.8579	0.9161	0.9494
male	0.9261	0.9558	0.9495
female	0.936	0.9656	0.9438
christian	0.9365	0.967	0.9418
jewish	0.895	0.9408	0.946
muslim	0.8664	0.9326	0.9409
psychiatric_or_mental_illness	0.9589	0.953	0.9679
homosexual_gay_or_lesbian	0.832	0.9275	0.9352
Mp	0.8895	0.9412	0.9463

Πίνακας 4-15: Υπό μετρικές για το RoBERTawPool στο private σύνολο ελέγχου

```
Layer (type)                 Output Shape              Param #   Connected to
==================================================================================================
input_word_ids (InputLayer)  [(None, 180)]             0
__________________________________________________________________________________________________
input_mask (InputLayer)      [(None, 180)]             0
__________________________________________________________________________________________________
tf_roberta_model_1 (TFRobertaMo ((None, 180, 768), ( 124645632   input_word_ids[0][0]
                                                                 input_mask[0][0]
__________________________________________________________________________________________________
global_average_pooling1d (Globa (None, 768)           0         tf_roberta_model_1[0][0]
__________________________________________________________________________________________________
global_max_pooling1d (GlobalMax (None, 768)           0         tf_roberta_model_1[0][0]
__________________________________________________________________________________________________
concatenate (Concatenate)    (None, 1536)              0         global_average_pooling1d[0][0]
                                                                 global_max_pooling1d[0][0]
__________________________________________________________________________________________________
dropout_76 (Dropout)         (None, 1536)              0         concatenate[0][0]
__________________________________________________________________________________________________
target (Dense)               (None, 1)                 1537      dropout_76[0][0]
__________________________________________________________________________________________________
aux (Dense)                  (None, 6)                 9222      dropout_76[0][0]
==================================================================================================
Total params: 124,656,391
Trainable params: 124,656,391
Non-trainable params: 0
```

Εικόνα 4-12: Περίληψη του RoBERTawPool



Εικόνα 4-13: Γράφος του RoBERTawPool

4.7 Νευρωνικό Δίκτυο με GPT2

Για την κατασκευή του δικτύου χρησιμοποιήθηκε η μικρή εκδοχή του GPT2 με 12 επίπεδα decoders. Αυτήν είναι η μόνη αλλαγή σε σχέση με την αρχιτεκτονική του προηγούμενου μοντέλου. Η εξαγωγή αποτελεσμάτων έγινε όπως στην προηγούμενη ενότητα.

Model GPT2wPool	Acc. Public	W. Pre. Public	Rec. Public	W. F1 Public	Public AUC	W. Acc. Private	W. Pre. Private	Rec. Private	W. F1 Private	Private AUC
Epoch 1	0.9513	0.9481	0.6045	0.9492	0.9298	0.9514	0.9482	0.6086	0.9492	0.9323
Epoch 2	0.9522	0.9493	0.6179	0.9503	0.9317	0.952	0.9492	0.6232	0.9502	0.9338
Epoch 3	0.9539	0.9505	0.6049	0.9514	0.9322	0.9536	0.9503	0.6092	0.9512	0.9339
Epoch 4	0.9521	0.9504	0.6539	0.9511	0.9312	0.9526	0.9508	0.6576	0.9516	0.9336

Πίνακας 4-16: Αποτελέσματα GPT2wPool στα σύνολα ελέγχου

Σύμφωνα με τον παραπάνω πίνακα καλύτερο Private AUC με 0.9339 έχει το μοντέλο της τρίτης εποχής, όπου αυτό επιλέγεται για σύγκριση με τα υπόλοιπα.

subgroup	subgroup_auc	bpsn_auc	bnsp_auc
black	0.8442	0.8918	0.957
white	0.8551	0.9002	0.956
male	0.92	0.9544	0.9473
female	0.9213	0.9589	0.943
christian	0.9406	0.9651	0.9428
jewish	0.9173	0.9407	0.9538
muslim	0.8773	0.9243	0.9494
psychiatric_or_mental_illness	0.8916	0.9429	0.9409
homosexual_gay_or_lesbian	0.8333	0.9076	0.9409
Mp	0.8844	0.9297	0.9478

Πίνακας 4-17: Υπό μετρικές για το GPT2wPool στο public σύνολο ελέγχου

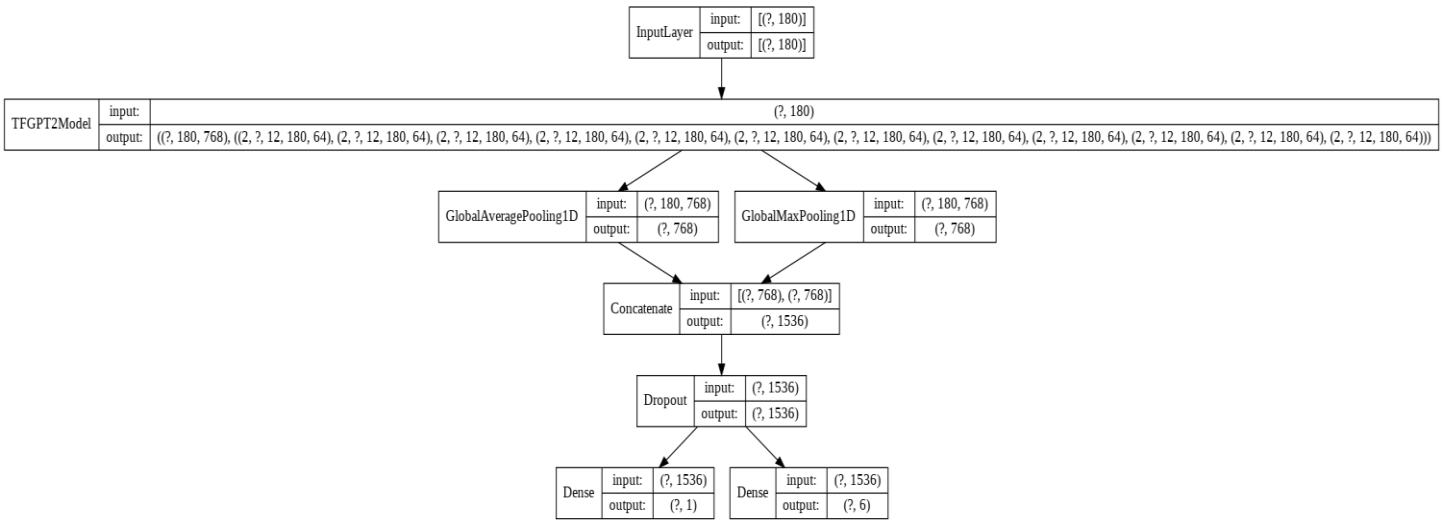
subgroup	subgroup_auc	bpsn_auc	bnsp_auc
black	0.8386	0.9109	0.9473
white	0.8481	0.9082	0.9528
male	0.9273	0.9551	0.9512
female	0.9341	0.9667	0.9426
christian	0.9358	0.9659	0.9417
jewish	0.8924	0.9429	0.9429
muslim	0.8554	0.9285	0.9425
psychiatric_or_mental_illness	0.9527	0.9523	0.9669
homosexual_gay_or_lesbian	0.8318	0.9087	0.9461
Mp	0.8839	0.936	0.948

Πίνακας 4-18: Υπό μετρικές για το GPT2wPool στο private σύνολο ελέγχου

Layer (type)	Output Shape	Param #	Connected to
input_word_ids (InputLayer)	[(None, 180)]	0	
tfgp_t2model (TFGPT2Model)	((None, 180, 768), (	124439808	input_word_ids[0][0]
global_average_pooling1d (Globa	(None, 768)	0	tfgp_t2model[0][0]
global_max_pooling1d (GlobalMax	(None, 768)	0	tfgp_t2model[0][0]
concatenate (Concatenate)	(None, 1536)	0	global_average_pooling1d[0][0] global_max_pooling1d[0][0]
dropout_37 (Dropout)	(None, 1536)	0	concatenate[0][0]
target (Dense)	(None, 1)	1537	dropout_37[0][0]
aux (Dense)	(None, 6)	9222	dropout_37[0][0]

Total params: 124,450,567
Trainable params: 124,450,567
Non-trainable params: 0

Εικόνα 4-14: Περίληψη του GPT2wPool



Εικόνα 4-15: Γράφος του GPT2wPool

4.8 Συλλογική Μάθηση (Ensemble Learning)

Μια συχνή τακτική για να αυξηθεί η απόδοση είναι αντί να έχουμε ένα μοντέλο που θα κάνει προβλέψεις, να έχουμε έναν συνδυασμό μοντέλων που θα αυξήσουν την ικανότητα πρόβλεψης. Η βασική ιδέα είναι ότι κάθε μοντέλο προσεγγίζει και μαθαίνει διαφορετικά μοτίβα από το σύνολο εκπαίδευσης, έτσι συνδυάζοντας διαφορετικά μοντέλα μας δίνετε η δυνατότητα διόρθωσης λαθών προβλέψεων του ενός μοντέλου από τα άλλα. Ένας τρόπος συνδυασμού μοντέλων είναι παίρνοντας τον μέσο όρο των προβλέψεων από κάθε μοντέλο. Συγκεκριμένα θα χρησιμοποιηθεί ο σταθμισμένος μέσος όρος με τα βάρη να βρίσκονται εμπειρικά μετά από δοκιμές. Θα ελεγχθούν οι συνδυασμοί μεταξύ των τεσσάρων καλύτερων τεσσάρων μοντέλων που έχουν εκπαιδευτεί και αυτά είναι τα:

- BiLSTM2-64
- BiGRU2-64
- RoBERTawPool 2ης εποχής
- GPT2wPool 3ης εποχής

Model Combination	Acc. Public	W. Pre. Public	Rec. Public	W. F1 Public	Public AUC	W. Acc. Private	W. Pre. Private	Rec. Private	W. F1 Private	Private AUC
RoBERTawPool (0.5) - BiLSTM2-64 (0.5)	0.9525	0.9510	0.6616	0.9517	0.9396	0.9529	0.9515	0.6694	0.9521	0.9413
RoBERTawPool (0.6)- GPT2wPool (0.4)	0.9524	0.9519	0.6847	0.9521	0.9377	0.9522	0.9516	0.6875	0.9519	0.9393
RoBERTawPool (0.5) - BiGRU2-64 (0.5)	0.9527	0.9510	0.6581	0.9517	0.9396	0.9530	0.9513	0.6631	0.9520	0.9411
BiLSTM2-64 (0.5)- GPT2wPool (0.5)	0.9544	0.9509	0.5981	0.9517	0.9384	0.9548	0.9515	0.6072	0.9522	0.9400
BiLSTM2-64 (0.5) - BiGRU2-64 (0.5)	0.953	0.9495	0.5931	0.9504	0.934	0.9537	0.9503	0.6041	0.9512	0.9357
GPT2wPool (0.5) - BiGRU2-64 (0.5)	0.9544	0.9509	0.5925	0.9516	0.938	0.9548	0.9514	0.6015	0.9520	0.9398
RoBERTawPool (0.4) - BiLSTM2-64 (0.4) -	0.9538	0.9517	0.6492	0.9525	0.9404	0.9540	0.952	0.6571	0.9528	0.9421

GPT2wPool (0.2)										
RoBERTawPool (0.4) - BiLSTM2-64 (0.3) - BiGRU2-64 (0.3)	0.9533	0.9511	0.6446	0.952	0.9398	0.9539	0.9517	0.6530	0.9526	0.9414
RoBERTawPool (0.4) - GPT2wPool (0.3) - BiGRU2-64 (0.3)	0.954	0.9518	0.6474	0.9527	0.9402	0.9542	0.9521	0.6558	0.9529	0.9419
BiLSTM2-64 (0.5) - GPT2wPool (0.4) - BiGRU2-64 (0.1)	0.9544	0.9509	0.5972	0.9517	0.9385	0.9547	0.9514	0.6051	0.9521	0.9402
RoBERTawPool (0.4) - BiLSTM2-64 (0.2) - GPT2wPool (0.2) - BiGRU2-64 (0.2)	0.9539	0.9518	0.6492	0.9526	0.9406	0.9543	0.9522	0.6559	0.9530	0.9423

Πίνακας 4-19: Αποτελέσματα Ensemble Averaging

Σύμφωνα με τον παραπάνω πίνακα καλύτερος συνδυασμός είναι ο RoBERTawPool (0.4) - BiLSTM2-64 (0.2) - GPT2wPool (0.2) - BiGRU2-64 (0.2), στις

παραθέσεις είναι οι τιμές των βαρών, με Private AUC ίσο με 0.9423. Αυτός ο συνδυασμός θα συγκριθεί με τα αποτελέσματα στον πίνακα βαθμολογίας του διαγωνισμού.

subgroup	subgroup_auc	bpsn_auc	bnsp_auc
black	0.8568	0.9126	0.9550
white	0.8722	0.9187	0.9563
male	0.9305	0.9629	0.9517
female	0.9326	0.9672	0.9479
christian	0.9489	0.9732	0.9455
jewish	0.9291	0.9542	0.9544
muslim	0.8917	0.9377	0.9505
psychiatric_or_mental_illness	0.8971	0.9456	0.9497
homosexual_gay_or_lesbian	0.8515	0.9277	0.9450
Mp	0.8973	0.9431	0.9506

Πίνακας 4-20: Υπό μετρικές για RoBERTawPool (0.4) - BiLSTM2-64 (0.2) - GPT2wPool (0.2) - BiGRU2-64 (0.2) στο public σύνολο ελέγχου

subgroup	subgroup_auc	bpsn_auc	bnsp_auc
black	0.8589	0.9272	0.9494
white	0.8645	0.9239	0.9533
male	0.935	0.9633	0.9536
female	0.9434	0.9735	0.9467
christian	0.944	0.9744	0.9439
jewish	0.917	0.9518	0.95
muslim	0.8726	0.9419	0.9439
psychiatric_or_mental_illness	0.9564	0.9584	0.9688
homosexual_gay_or_lesbian	0.8482	0.9307	0.9437
Mp	0.899	0.9484	0.9502

Πίνακας 4-21: Υπό μετρικές για το RoBERTawPool (0.4) - BiLSTM2-64 (0.2) - GPT2wPool (0.2) - BiGRU2-64 (0.2) στο private σύνολο ελέγχου

Από τους παραπάνω πίνακες φαίνεται πως ο συνδυασμός μοντέλων RoBERTawPool (0.4) - BiLSTM2-64 (0.2) - GPT2wPool (0.2) - BiGRU2-64 (0.2) να πηγαίνει εξαιρετικά

στις μετρικές BPSN AUC και BNSP AUC αυτό σημαίνει πως οι προβλέψεις για τα σχόλια με ταυτότητα δεν επηρεάζονται αρνητικά από αυτές των σχόλιων χωρίς ταυτότητα. Ωστόσο για κάποιες ταυτότητες δυσκολεύεται λίγο να προβλέψει αν τα σχόλια που αναφέρονται είναι τοξικά ή όχι.

5 Επίλογος

5.1 Σύνοψη και συμπεράσματα

Κύριος στόχος της συγκεκριμένης εργασίας είναι η πρόβλεψη τοξικότητας σε σχόλια, δεδομένου ενός συνόλου τοξικών και μη σχολίων. Δευτερεύων στόχος είναι η ελαχιστοποίηση της προκατάληψης ενός μοντέλου που μπορεί να έχει έναντι μιας πληθυσμιακής ταυτότητας.

Οι αρχιτεκτονικές Επαναληπτικών Νευρωνικών Δικτύων έφεραν εξαιρετικά αποτελέσματα και για αυτόν τον λόγο χρησιμοποιήθηκαν, ένα δίκτυο με LSTM και ένα με GRU, στις καλύτερες 4 αρχιτεκτονικές της συλλογικής μάθησης. Η αρχιτεκτονική Συνελικτικού Νευρωνικού Δικτύου, όπως αναμενόταν, είχε χαμηλότερη επίδοση της τάξης 1-1.5% σε όλες τις μετρικές σε σύγκριση με τα υπόλοιπα μοντέλα. Ωστόσο η εκπαίδευση του Συνελικτικού Δικτυού ήταν η ταχύτερη με κάθε εποχή να χρειάζεται κατά μέσο όρο 7 λεπτά, ενώ στα μοντέλα με LSTM και GRU να χρειάζεται 10 λεπτά και στα μοντέλα με Transformers να χρειάζεται 1 ώρα. Οι αρχιτεκτονικές με Transformers σε επίπεδο επίδοσης έφεραν παρόμοια και καλύτερα αποτελέσματα από τις αρχιτεκτονικές Επαναληπτικών Δικτύων, όπως αναμενόταν, με εξαίρεση το μοντέλο που κάνει χρήση BERT. Στην Συλλογική Μάθηση ο συνδυασμός και των τεσσάρων καλύτερων αρχιτεκτονικών έφερε τα καλύτερα αποτελέσματα και οι επιδόσεις του θα συγκριθούν με αυτές στον βαθμολογικό πίνακα του διαγωνισμού.

Δυστυχώς η εργασία δεν συμμετείχε στον διαγωνισμό γιατί η εκπόνησή της και η έρευνα ξεκίνησε μετά το πέρας του διαγωνισμού. Συνολικά στον διαγωνισμό συμμετείχαν 2646 ομάδες και η καλύτερη λύση έχει Private AUC score 0.94734. Η λύση που αντιπροσωπεύει αυτήν την εργασία, δηλαδή ο συνδυασμός μοντέλων RoBERTawPool (0.4) - BiLSTM2-64 (0.2) - GPT2wPool (0.2) - BiGRU2-64 (0.2) με Private AUC score 0.94233 θα ήταν στην θέση 165 κατατάσσοντας την στο κορυφαίο 6% των λύσεων του διαγωνισμού.

5.2 Όρια και περιορισμοί της έρευνας

Όπως παρουσιάστηκε και σε προηγούμενη ενότητα το σύνολο εκπαίδευσης ήταν αρκετά μεγάλο και μη ισορροπημένο, με το μεγαλύτερο ποσοστό των σχολίων να μην είναι τοξικά. Επιπλέον μπορεί η λύση με την Συλλογική Μάθηση να φέρνει καλά αποτελέσματα αλλά δεν προτείνετε για χρήση σε υπηρεσία κάποιας εταιρείας. Μάλιστα πολλές λύσεις του διαγωνισμού που χρησιμοποίησαν Συλλογική Μάθηση συμπεριλάμβαναν στους συνδυασμούς τους πάνω από 10 μοντέλα, το οποίο είναι απαγορευτικό στην κατασκευή μιας υπηρεσίας για παράδειγμα που θα προβλέπει την τοξικότητα στα σχόλια. Ένα άλλο πρόβλημα που αντιμετωπίστηκε ήταν οι περιορισμένοι υπολογιστικοί πόροι.

5.3 Μελλοντικές Επεκτάσεις

Δεδομένου ότι αρχιτεκτονικές με Transormers ήταν αποτελεσματικές, ειδικά το μοντέλο με χρήση RoBERTa, θα δοκίμαζα, αντί για ένα απλό pooling που γίνεται, να βάλω ένα ή περισσότερα Συνελικτικά επίπεδα αμέσως μετά των επιπέδων του Transformer. Επίσης θα επιχειρούσα να κάνω κάτι παρόμοιο και με τα Επαναληπτικά Δίκτυα. Επιπλέον θα χρησιμοποιούσα διαφορετικές μεθόδους συλλογικής μάθησης, λιγότερο απλοϊκούς, όπως το Boosting και Stacking. Τέλος μια καλή ιδέα για την αύξηση της απόδοσης είναι η εύρεση μιας καλύτερης ευρετικής για τον υπολογισμό των βαρών κάθε παραδείγματος. Μια τέτοια ευρετική θα έδινε σημασία στον διαχωρισμό τοξικών και μη σχόλιων που αναφέρουν την ταυτότητα.

6 Βιβλιογραφία

[1] McCulloch, W. S. and Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5 (4), pp.115-133.

[2] Li, Karpathy, "CS231n: Convolutional Neural Networks for Visual Recognition" (Course Notes). Available at: https://cs231n. github. io/neural-networks-1/#actfun (May 10 2020).

[3] Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y., 2016. Deep learning (Vol.1). Cambridge: MIT press.

[4] Szepesvári, C., 2010. Algorithms for reinforcement learning. Synthesis lectures on artificial intelligence and machine learning, 4 (1), pp.1-103.

[5] Duchi, J., Hazan, E. and Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research, 12 (7).

[6] Zeiler, M. D., 2012. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701.

[7] Kingma, D. P. and Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[8] Ho, C. C., Baharim, K. N., Fatan, A. A. A. and Alias, M. S. B., Deep Neural Networks for Text: A Review.

[9] LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521 (7553), pp.436-444.

[10] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15 (1), pp.1929-1958.

[11] Kalchbrenner, N., Grefenstette, E. and Blunsom, P., 2014. A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188.

[12] Kim, Y., 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

[13] Zhang, X. and LeCun, Y., 2015. Text understanding from scratch. arXiv preprint arXiv:1502.01710.

[14] Bengio, Y., Simard, P. and Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5 (2), pp.157-166.

[15] Pascanu, R., Mikolov, T. and Bengio, Y., 2013, February. On the difficulty of training recurrent neural networks. In International conference on machine learning (pp.1310-1318).

[16] Gui, T., Zhang, Q., Zhao, L., Lin, Y., Peng, M., Gong, J. and Huang, X., 2019, July. Long short-term memory with dynamic skip connections. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol.33, pp.6481-6488).

[17] Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9 (8), pp.1735-1780.

[18] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

[19] Suzgun, M., Gehrmann, S., Belinkov, Y. and Shieber, S. M., 2019. LSTM networks can perform dynamic counting. arXiv preprint arXiv:1906.03648.

[20] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. In Advances in neural information processing systems (pp.5998-6008).

[21] Ba, J. L., Kiros, J. R. and Hinton, G. E., 2016. Layer normalization. arXiv preprint arXiv:1607.06450.

[22] Devlin, J., Chang, M. W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[23] Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A. and Fidler, S., 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In Proceedings of the IEEE international conference on computer vision (pp.19-27).

[24] Sennrich, R., Haddow, B. and Birch, A., 2015. Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909.

[25] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V., 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

[26] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., 2019. Language models are unsupervised multitask learners. OpenAI Blog, 1 (8), p.9.

[27] Jigsaw, "Jigsaw Unintended Bias in Toxicity Classification" (Competition), 20 May 2019. Available at: https://www. kaggle. com/c/jigsaw-unintended-bias-in-toxicity-classification/data (August 10 2020).

[28] McKinney, W., 2010, June. Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference (Vol.445, pp.51-56).

[29] Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[30] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp.3111-3119).

[31] Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C. and Joulin, A., 2017. Advances in pre-training distributed word representations. arXiv preprint arXiv:1712.09405.

[32] authman, "Pickled Crawl-300D-2M For Kernel Competitions" (Dataset), 16 April 2019. Available at: https://www. kaggle. com/authman/pickled-crawl300d2m-for-kernel-competitions (August 10 2020).

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

[34] authman, "Pickled glove.840B.300d" (Dataset), 16 April 2019. Available at: https://www. kaggle. com/authman/pickled-crawl300d2m-for-kernel-competitions (August 10 2020).

[35] Borkan, D., Dixon, L., Sorensen, J., Thain, N. and Vasserman, L., 2019, May. Nuanced metrics for measuring unintended bias with real data for text classification. In Companion Proceedings of The 2019 World Wide Web Conference (pp.491-500).