



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Ανάλυση Συναισθήματος από Κείμενα με Χρήση Τεχνικών
Μηχανικής Μάθησης**

Ιωάννης Ε. Κερατζάκης

Επιβλέπων: Παναγιώτης Σταματόπουλος, Επίκουρος Καθηγητής

ΑΘΗΝΑ

ΟΚΤΩΒΡΙΟΣ 2019

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάλυση Συναισθήματος από Κείμενα με Χρήση Τεχνικών Μηχανικής Μάθησης

Ιωάννης Ε. Κερατζάκης

A.M.: M1615

ΕΠΙΒΛΕΠΩΝ: Παναγιώτης Σταματόπουλος, Επίκουρος Καθηγητής

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: Παναγιώτης Σταματόπουλος, Επίκουρος Καθηγητής
Δημήτριος Γουνόπουλος, Καθηγητής

Οκτώβριος 2019

ΠΕΡΙΛΗΨΗ

Στην παρούσα εργασία εξετάζουμε την εφαρμογή τεχνικών Μηχανικής Μάθησης με σκοπό την Ανάλυση Συναισθήματος - ΑΣ (Sentiment Analysis) από κείμενα. Αρχικά γίνεται μια επισκόπηση των κύριων σημείων της θεωρίας που διέπουν την επιστημονική περιοχή, έπειτα υλοποιείται ένα σύστημα ανάλυσης συναισθήματος με χρήση νευρωνικών δικτύων, και τέλος εξετάζονται ειδικότερες πειραματικές τεχνικές και αρχιτεκτονικές που οδηγούν στην αύξηση της συνολικής απόδοσης.

Η Ανάλυση Συναισθήματος και γενικότερα ο τομέας της Επεξεργασίας Φυσικής Γλώσσας – ΕΦΓ (Natural Language Processing- NLP) στον οποίο ανήκει, αποτελούν ενεργές ερευνητικές περιοχές με σημαντικά ευρήματα κάθε χρόνο. Σε μια εποχή που ο διαθέσιμος όγκος των δεδομένων αυξάνεται με σημαντικό ρυθμό και η ενσωμάτωση της Τεχνητής Νοημοσύνης – ΤΝ (Artificial Intelligence - AI) στην καθημερινότητα είναι γεγονός, είναι βέβαιο πως συστήματα όπως αυτό της παρούσας εργασίας θα έχουν ρόλο κλειδί σε πολλούς τομείς της τεχνολογίας.

Στα πρώτα δύο κεφάλαια γίνεται μια επισκόπηση του θεωρητικού υπόβαθρου της εργασίας. Ειδικότερα, στο πρώτο κεφάλαιο περιγράφονται οι βασικές έννοιες της Μηχανικής Μάθησης και της Ανάλυσης Συναισθήματος, και στο δεύτερο γίνεται μια παρουσίαση των νευρωνικών δικτύων. Οι δύο κατηγορίες νευρωνικών δικτύων που χρησιμοποιήθηκαν είναι τα Συνελικτικά Νευρωνικά Δίκτυα - ΣΝΔ (Convolutional Neural Networks - CNNs) και οι Αναδρομικές Μονάδες με Πύλες - ΑΜΠ (Gated Recurrent Units - GRUs), μια ειδική κατηγορία αναδρομικών νευρωνικών δικτύων.

Στο τρίτο κεφάλαιο κάνουμε μια σύντομη παρουσίαση στην προγραμματιστική βιβλιοθήκη TensorFlow, και συγκεκριμένα στις λειτουργίες της που χρησιμοποιήσαμε. Στο τέταρτο κεφάλαιο περιγράφεται η μεθοδολογία και η αρχιτεκτονική ταξινόμησης που χρησιμοποιείται για την ανάλυση συναισθήματος, καθώς και τα σύνολα δεδομένων πάνω στα οποία εκπαιδεύτηκε και αποτιμήθηκε το σύστημα. Τέλος, στο πέμπτο κεφάλαιο παρουσιάζονται τα αποτελέσματα που προέκυψαν από την εφαρμογή του συστήματος.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Μηχανική Μάθηση

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: επεξεργασία φυσικής γλώσσας, νευρωνικά δίκτυα, ανάλυση συναισθήματος, tensorflow, ταξινόμηση

ABSTRACT

In this thesis we study the usage of machine learning techniques for Sentiment Analysis of texts. First of all, the key theory is presented, followed by an implementation of a Sentiment Analysis system by using neural networks. Finally, more advanced architectures and techniques are examined with the goal of increasing the overall performance.

Sentiment Analysis and the field in which it belongs, Natural Language Processing, are both active research fields with important advances every year. In the era of Big Data and the integration of Artificial Intelligence into aspects of the everyday life, it's certain that systems like the one described in this thesis, will play an important role in many technological areas.

In the first two chapters, the basic theoretical concepts are explained. More precisely, in the first chapter the basic concepts of Machine Learning and Sentiment Analysis are described, and in the second chapter a presentation of neural networks is made. The two types of neural networks used are convolutional neural networks and gated recurrent units, a special type of recurrent neural networks.

In the third chapter we briefly present the TensorFlow library and more specifically the TensorFlow features that were utilized. In the fourth chapter the methodology and the classification architecture behind the sentiment analysis system is described, along with a look at the datasets that the system was trained and evaluated on. Finally, in the fifth chapter we present the results that were achieved and some useful conclusions.

SUBJECT AREA: Machine Learning

KEYWORDS: natural language processing, neural networks, sentiment analysis, tensorflow, classification

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω όλους εκείνους τους ανθρώπους που αισθάνομαι κοντά μου. Η συμβολή τους είναι ιδιαίτερα σημαντική. Επίσης, θα ήθελα από εδώ να ευχαριστήσω τον επιβλέποντα μου, καθηγητή κ. Σταματόπουλο για την εξαιρετική συνεργασία που είχαμε. Οι υποδείξεις του και η καθοδήγηση του ήταν καθοριστικές.

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΕΙΣΑΓΩΓΗ.....	12
1.1 Μηχανική Μάθηση.....	12
1.2 Ανάλυση Συναισθήματος.....	13
1.2.1 Βασικές έννοιες.....	14
1.2.2 Εφαρμογές της ΑΣ.....	15
2. ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ.....	17
2.1 Επισκόπηση λειτουργίας ΤΝΔ.....	17
2.2 Συνελικτικά Νευρωνικά Δίκτυα.....	20
2.3 Αναδρομικά Νευρωνικά Δίκτυα.....	24
2.3.1 Βασικές έννοιες.....	24
2.3.2 Αναδρομικές Μονάδες με Πύλες.....	28
2.4 Άλλα ζητήματα μηχανικής μάθησης.....	29
2.4.1 Διαχωρισμός συνόλου δεδομένων.....	29
2.4.2 Υποπροσαρμογή και υπερπροσαρμογή.....	30
3. ΕΙΣΑΓΩΓΗ ΣΤΟ TENSORFLOW.....	35
3.1 Γενικά.....	35
3.2 Δυνατότητες προεπεξεργασίας.....	35
3.3 Κατασκευή νευρωνικών δικτύων και εκπαίδευση.....	37
3.4 Γραφήματα εκπαίδευσης.....	41
4. ΘΕΜΑΤΑ ΕΠΕΞΕΡΓΑΣΙΑΣ ΦΥΣΙΚΗΣ ΓΛΩΣΣΑΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗΣ.....	43
4.1 Διανύσματα λέξεων.....	43
4.2 Σύνολα δεδομένων που χρησιμοποιήθηκαν.....	45
4.2.1 Το σύνολο δεδομένων Amazon review data.....	45
4.2.2 Το σύνολο δεδομένων Sentiment140.....	48
4.3 Προεπεξεργασία.....	50
4.4 Αρχιτεκτονική συστήματος ΑΣ.....	51
5. ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΧΟΛΙΑΣΜΟΣ.....	57
6. ΣΥΜΠΕΡΑΣΜΑΤΑ.....	61
ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ.....	62
ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ.....	64

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Μορφή συστήματος ταξινόμησης.....	13
Εικόνα 2.1: Μοντέλο απλού τεχνητού νευρώνα.....	17
Εικόνα 2.2: Λογιστική συνάρτηση.....	18
Εικόνα 2.3: Συνάρτηση υπερβολικής εφασπτομένης.....	18
Εικόνα 2.4: Διαχωριστική γραμμή ταξινόμησης για 2 κατηγορίες.....	19
Εικόνα 2.5: Ένα Τεχνητό Νευρωνικό Δίκτυο με 3 βασικά επίπεδα.....	19
Εικόνα 2.6: Ένα ΣΝΔ για αναγνώριση χειρόγραφων ψηφίων.....	21
Εικόνα 2.7: Συνάρτηση ενεργοποίησης γραμμικού ανορθωτή (ReLU).....	21
Εικόνα 2.8: Χάρτης χαρακτηριστικών για το παράδειγμα μας.....	22
Εικόνα 2.9: Συγκέντρωση μεγίστων για τον χάρτη χαρακτηριστικών.....	22
Εικόνα 2.10: ΑΝΔ με έξοδο σε κάθε χρονικό βήμα.....	25
Εικόνα 2.11: ΑΝΔ με σύνδεση της εξόδου με το επόμενο χρονικό βήμα.....	25
Εικόνα 2.12: ΑΝΔ με έξοδο μόνο στο τελευταίο χρονικό βήμα.....	26
Εικόνα 2.13: Αμφίδρομο ΑΝΔ. Διακρίνεται η ροή πληροφορίας προς δύο κατευθύνσεις	27
Εικόνα 2.14: Κλασική εκδοχή των ΑΜΠ. Διακρίνονται οι πύλες και οι πράξεις που λαμβάνουν χώρα.....	28
Εικόνα 2.15: Επίδραση της υποπροσαρμογής στην γραφική παράσταση απόδοσης- εποχών.....	30
Εικόνα 2.16: Επίδραση της υποπροσαρμογής στην συνάρτηση απώλειας	31
Εικόνα 2.17: Επίδραση της υπερπροσαρμογής στην επίδοση.....	32
Εικόνα 2.18: Επίδραση της υπερπροσαρμογής στην συνάρτηση απώλειας.....	32

Εικόνα 2.19: Παράδειγμα ικανοποιητικής εκπαίδευσης.....	33
Εικόνα 2.20: Μέθοδος εγκατάλειψης. Οι νευρώνες με X έχουν εξαιρεθεί από την εκπαίδευση.....	34
Εικόνα 3.1: Παράδειγμα γραφήματος ακρίβειας-εποχών.....	42
Εικόνα 3.2: Παράδειγμα γραφήματος συνάρτησης απώλειας-εποχών.....	42
Εικόνα 4.1: Μοντέλο συνεχούς σάκου λέξεων (CBOW).....	44
Εικόνα 4.2: Μοντέλο skip-gram (πρόβλεψη συμφραζομένων).....	44
Εικόνα 4.3: Κατανομή μήκους κριτικών ταινιών.....	46
Εικόνα 4.4: Κατανομή μήκους κριτικών για αντικείμενα σχετικά με αθλήματα.....	46
Εικόνα 4.5: Κατανομή μήκους για αντικείμενα σχετικά με παιχνίδια.....	47
Εικόνα 4.6: Κατανομή μήκους μηνυμάτων χρηστών.....	49
Εικόνα 4.7: Συχνότερες λέξεις σε θετικά μηνύματα χρηστών.....	49
Εικόνα 4.8: Συχνότερες λέξεις σε αρνητικά μηνύματα χρηστών.....	50
Εικόνα 4.9: Χρήση διανυσμάτων λέξεων από ΣΝΔ.....	51
Εικόνα 4.10: Αρχιτεκτονική ΣΝΔ για Ανάλυση Συναισθήματος.....	52
Εικόνα 4.11: Αρχιτεκτονική ΑΜΠ για Ανάλυση Συναισθήματος.....	53
Εικόνα 4.12: Στιγμιότυπο εκπαίδευσης όπως παρουσιάζεται από το Tensorflow.....	54
Εικόνα 4.13: Αποτέλεσμα ΑΣ για 2 μηνύματα Twitter.....	55
Εικόνα 4.14: Επίδραση συμφραζομένων στην ΑΣ.....	56

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 4.1: Κατανομή αριθμού αστεριών.....	47
Πίνακας 5.1: Συγκεντρωτικά αποτελέσματα.....	57
Πίνακας 5.2: Σύγκριση με άλλες μεθόδους μηχανικής μάθησης.....	57
Πίνακας 5.3: Επιρροή της σύνοψης στο αποτέλεσμα.....	58
Πίνακας 5.4: Τιμές παραμέτρων για το σύνολο Movies and TV.....	58
Πίνακας 5.5: Τιμές παραμέτρων για το σύνολο Toys and Games.....	59
Πίνακας 5.6: Τιμές παραμέτρων για το σύνολο Sports and Outdoors.....	59
Πίνακας 5.7: Τιμές παραμέτρων για το σύνολο Sentiment140.....	59
Πίνακας 5.8: Αποτελέσματα για το πρόβλημα 3 κατηγοριών.....	60

ΠΡΟΛΟΓΟΣ

Η εργασία αυτή διεξήχθη στο πλαίσιο του ΠΜΣ του τμήματος, και συγκεκριμένα στο πλαίσιο της 6ης κατεύθυνσης “Επεξεργασία-Μάθηση Σήματος και Πληροφορίας”. Επιλέχθηκε ένα θέμα με μεγάλο ερευνητικό ενδιαφέρον, η Ανάλυση Συναισθήματος, και υλοποιήθηκε σε προγραμματιστικό περιβάλλον Python με χρήση βοηθητικών βιβλιοθηκών.

Μέσα στο κείμενο έχει γίνει μια προσπάθεια να μεταφραστούν οι περισσότεροι όροι όπου ήταν δυνατό από την αγγλική γλώσσα στην γλώσσα συγγραφής, την ελληνική. Όπου δεν είναι αυτό εφικτό, αφέθηκαν ως έχουν και μερικές φορές εντός εισαγωγικών. Στις περιπτώσεις όπου έγινε μετάφραση, δίνεται ο ελληνικός όρος και εντός παρενθέσεων ο αγγλικός όρος από τον οποίο έγινε η μετάφραση. Στο κείμενο χρησιμοποιούνται επίσης και αρκετές συντομογραφίες και την αποφυγή επανάληψης μεγάλων όρων. Στο τέλος της εργασίας υπάρχουν πίνακες με τις μεταφράσεις των όρων και επεξήγηση των συντμήσεων για την διευκόλυνση του αναγνώστη.

Για λόγους αναγνωσιμότητας, τα τμήματα κώδικα και τα αποτελέσματα αυτών που βρίσκονται στο κεφάλαιο 3 περικλείονται από κενά, με διαφορετική γραμματοσειρά σε σχέση με το υπόλοιπο κείμενο.

1. ΕΙΣΑΓΩΓΗ

Για την ΤΝ έχουν δοθεί πολλοί ορισμοί κατά καιρούς. Σύμφωνα με το [1], ένας ορισμός που συνδυάζει τα σημαντικότερα στοιχεία από αυτούς είναι ο εξής: *“Τεχνητή Νοημοσύνη είναι ο τομέας των Υπολογιστών που ασχολείται με τη σχεδίαση και την υλοποίηση προγραμμάτων τα οποία είναι ικανά να μιμηθούν τις ανθρώπινες γνωστικές ικανότητες, εμφανίζοντας έτσι χαρακτηριστικά που αποδίδουμε συνήθως σε ανθρώπινη συμπεριφορά, όπως για παράδειγμα η επίλυση προβλημάτων, η αντίληψη μέσω της όρασης, η μάθηση, η εξαγωγή συμπερασμάτων, η κατανόηση φυσικής γλώσσας, κτλ.”*

Τα τελευταία χρόνια ο κλάδος της ΤΝ έχει σημειώσει σημαντική πρόοδο. Σε αυτό συμβάλλουν η καθιέρωση πολλών σημαντικών διεθνών συνεδρίων, η αύξηση των διαθέσιμων δεδομένων, η σημαντικότερη αύξηση της υπολογιστικής ισχύος καθώς και η ολοένα αυξανόμενη ενσωμάτωση της τόσο στην αγορά εργασίας όσο και στην καθημερινότητα.

Από τους κλάδους της ΤΝ με την μεγαλύτερη πρόοδο, είναι η Μηχανική Μάθηση (Machine Learning), η οποία χρησιμοποιείται και στην παρούσα εργασία.

1.1 Μηχανική Μάθηση

Η Αναγνώριση Προτύπων (Pattern Recognition) ασχολείται με την *“...ταξινόμηση αντικειμένων σε κατηγορίες ή κλάσεις”* [2]. Η Μηχανική Μάθηση είναι ένας υποτομέας της Αναγνώρισης Προτύπων, και αν και δεν ταυτίζονται, έχουν πολλά κοινά χαρακτηριστικά. Σκοπός της είναι η δημιουργία ενός συστήματος που έχει την ικανότητα να “μαθαίνει” και να ταξινομεί αντικείμενα. Δηλαδή δοθέντος ενός συνόλου αντικειμένων, τα οποία περιγράφονται από κάποια χαρακτηριστικά, ο στόχος μας είναι να τα αναθέσουμε στις κατάλληλες κατηγορίες στις οποίες ανήκουν. Τα αντικείμενα αυτά μπορούν να είναι πολλών διαφορετικών μορφών, όπως βίντεο, κείμενο, εικόνα, ήχος. Τα χαρακτηριστικά από τα οποία περιγράφονται εξαρτάται από την φύση των αντικειμένων. Για παράδειγμα, αν διαθέταμε ένα σύνολο από εικόνες, τα χαρακτηριστικά τους θα είναι ο δισδιάστατος ή τρισδιάστατος πίνακας εικονοστοιχείων (pixels) τους. Παρομοίως, εάν είχαμε ένα σύνολο από στοιχεία ασθενών με στόχο τον χαρακτηρισμό τους ως πιθανοί διαβητικοί ή όχι, τα δυνητικά χαρακτηριστικά τους θα μπορούσαν να είναι η τιμή του σακχάρου στο αίμα, το βάρος, ο μέσος όρος της αρτηριακής πίεσης, η ηλικία κτλ.

Πιο τυπικά έχουμε ένα σύνολο χαρακτηριστικών πλήθους n , που σχηματίζουν το διάνυσμα χαρακτηριστικών $x = [x_1, x_2, \dots, x_n]$.

Η αρχική μορφή που έχουν οι συνιστώσες του διανύσματος εξαρτάται άμεσα από το πρότυπο το οποίο περιγράφουν. Για ένα κείμενο οι συνιστώσες μπορούν να είναι οι λέξεις ή φράσεις που το αποτελούν, ενώ όταν το πρότυπο είναι ένας άνθρωπος που εξετάζεται για πιθανότητα διαβήτη, οι συνιστώσες είναι αριθμητικές τιμές. Ένα σημείο που πρέπει να τονιστεί εδώ είναι πως οι είσοδοι σε ένα σύστημα ταξινόμησης σπάνια έχουν την αρχική τους μορφή, καθώς προηγείται η προεπεξεργασία των δεδομένων ώστε να είναι σε μορφή αφενός κατανοητή από τον αλγόριθμο μηχανικής μάθησης (δηλαδή αριθμητική), και αφετέρου αποτελεσματική ως προς την απόδοση και την ακρίβεια του συστήματος.

Τα τρία κυριότερα είδη Μηχανικής Μάθησης είναι τα εξής:

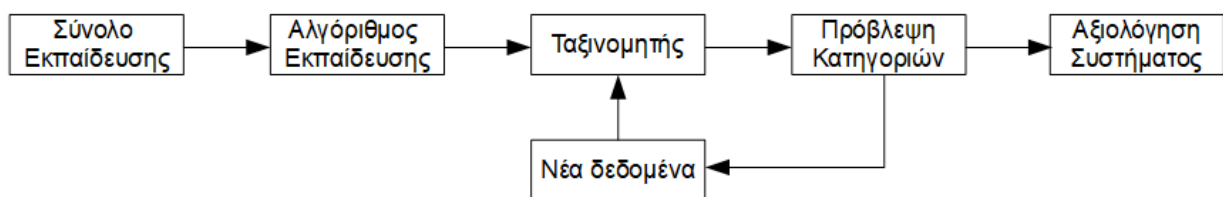
- Μάθηση υπό επίβλεψη (Supervised Learning)
- Μάθηση χωρίς επίβλεψη (Unsupervised Learning)
- Ενισχυτική Μάθηση (Reinforcement Learning)

Στην Μάθηση υπό επίβλεψη δίνεται στο σύστημα κάθε φορά ένα σύνολο προτύπων μαζί με τις κατηγορίες στις οποίες ανήκουν. Σκοπός του συστήματος είναι η κατασκευή ενός μοντέλου τέτοιου που μέσω της προσέγγισης μιας συνάρτησης, να βρει τον τρόπο με τον οποίο αντιστοιχίζονται νέα πρότυπα στις κατάλληλες κατηγορίες. Στην παρούσα εργασία τα μοντέλα που θα χρησιμοποιηθούν ανήκουν στην Μάθηση υπό επίβλεψη.

Στην Μάθηση χωρίς επίβλεψη, ο αλγόριθμος ανακαλύπτει μόνος του τις κατηγορίες στις οποίες μπορούν να ανήκουν τα πρότυπα, χωρίς την κατασκευή κάποιας συνάρτησης προσέγγισης. Οι κατηγορίες αυτές ονομάζονται και ομάδες, γι' αυτό και η Μάθηση χωρίς επίβλεψη συναντάται συχνά στην βιβλιογραφία με τον όρο ομαδοποίηση (ή συσταδοποίηση).

Η Ενισχυτική Μάθηση χρησιμοποιεί μια διαφορετική φιλοσοφία από τους προηγούμενους δύο τύπους μάθησης. Θεωρούμε πως το πρόβλημα περιγράφεται επαρκώς από ένα μοντέλο πράκτορα (agent) ο οποίος δρα σε ένα περιβάλλον (environment). Στον πράκτορα ανατίθενται οι δυνατές ενέργειες που μπορεί να κάνει, και ανάλογα με το αποτέλεσμα αυτών επιβραβεύεται. Η Ενισχυτική Μάθηση βρίσκεται πιο κοντά στην κλασική θεωρία της TN.

Στο σχήμα που ακολουθεί φαίνεται ένα τυπικό σύστημα ταξινόμησης για Μάθηση υπό επίβλεψη.



Εικόνα 1: Μορφή συστήματος ταξινόμησης [3]

Το σύνολο εκπαίδευσης (training dataset) περιλαμβάνει τα πρότυπα που θα χρησιμοποιηθούν για την κατασκευή του αλγορίθμου εκπαίδευσης. Τα υπόλοιπα τμήματα του συστήματος θα εξηγηθούν στην συνέχεια καθώς θα παρουσιαστούν η ΑΣ και τα νευρωνικά δίκτυα.

1.2 Ανάλυση Συναισθήματος

Η ΑΣ ασχολείται κυρίως με την αυτόματη εξαγωγή συναισθηματικών καταστάσεων και γνώμων με χρήση τεχνικών ΕΦΓ. Η ΕΦΓ είναι ένας κλάδος που συνδυάζει διάφορες επιστημονικές περιοχές όπως η Γλωσσολογία, η TN, η Επιστήμη των Υπολογιστών κ.α. με σκοπό την επεξεργασία και ανάλυση δεδομένων που προέρχονται από την φυσική γλώσσα. Οι κύριες πηγές τέτοιων δεδομένων είναι η ανθρώπινη φωνή, και όλα τα

κείμενα που έχουν συγγραφεί από ανθρώπους όπως βιβλία, άρθρα, κριτικές, ή οποιοδήποτε κείμενο συναντάται σε κοινωνικά δίκτυα.

Από μία οπτική, η ΑΣ μπορεί να θεωρηθεί ένα είδος ταξινόμησης κειμένων (text classification) όπου κατασκευάζεται ένας αλγόριθμος που αποδίδει αυτόματα κατηγορίες στα κείμενα που εισάγονται. Όμως η ΑΣ έχει αρκετές ιδιαιτερότητες οι οποίες πηγάζουν κυρίως από το γεγονός πως οι κλασικές περιπτώσεις κειμένων που είναι τα αντικείμενα επεξεργασίας της ΑΣ όπως οι κριτικές για μία υπηρεσία ή κάποιο προϊόν και οι δημοσιεύσεις σε μέσα κοινωνικής δικτύωσης, περιέχουν μεγάλες δόσεις συναισθηματικότητας και υποκειμενικότητας.

1.2.1 Βασικές έννοιες

Σύμφωνα με τον Liu [4], μία γνώμη μπορεί αρχικά να οριστεί ως μία δυάδα (t, s) όπου t είναι ο στόχος (target) της γνώμης (opinion) και s είναι το συναίσθημα (sentiment) που σχετίζεται με το συγκεκριμένο αντικείμενο. Οι πιο κλασικές περιπτώσεις που συναντώνται είναι ο διαχωρισμός του συναισθήματος σε θετικό, ουδέτερο και αρνητικό, ή η ανάθεση ενός σκορ, το οποίο συνήθως εκφράζεται σε μια αριθμητική κλίμακα (για παράδειγμα ως (N/M) όπου N ο αριθμός των αστεριών μιας κριτικής και M ο μέγιστος δυνατός αριθμός αστεριών που μπορεί να λάβει το αντικείμενο).

Ως παράδειγμα ας υποθέσουμε πως έχουμε την ακόλουθη κριτική για μία ταινία, στα αγγλικά:

“I really liked this movie. The direction was wonderful, there was lots of character development and the story was an intriguing and very creative. If you haven't watched this yet, I highly recommend that you do”. 5/5

Εδώ, ο στόχος της γνώμης είναι η ταινία, και το συναίσθημα είναι ο αριθμός των αστεριών, δηλαδή 5 σε σύνολο 5 αστεριών. Είναι εμφανές πως ο συγγραφέας της κριτικής ένιωσε πολύ θετικό συναίσθημα για αυτή την ταινία.

Ως άλλο ένα παράδειγμα ας υποθέσουμε το ακόλουθο τίβισμα (tweet) από το Twitter [5]:

“@some_random_user I really dislike the direction this country is going in. our government has been terrible so far and I fear for the worst”

Στόχος της γνώμης είναι η κυβέρνηση, και το συναίσθημα του χρήστη σχετικά με αυτήν είναι αρνητικό.

Επεκτείνοντας τον προηγούμενο ορισμό, μία γνώμη μπορεί να οριστεί ως η πλειάδα τεσσάρων στοιχείων [4] (o, s, h, t) όπου o είναι το αντικείμενο της γνώμης, s το συναίσθημα, h εκείνος που εκφράζει την γνώμη και t η χρονική στιγμή η οποία εκφράστηκε η γνώμη. Αυτός ο ορισμός έχει πιο πρακτική αξία όσον αφορά την αποθήκευση γνώμων σε βάσεις δεδομένων, αλλά μάλλον δεν έχει και τόση χρησιμότητα για την ΑΣ καθώς είναι αδιάφορο το ποιος και πότε εκφράζει την γνώμη στις περισσότερες περιπτώσεις. Εξάιρεση αποτελούν συστήματα ΑΣ όπου λαμβάνονται υπόψιν τα συμφραζόμενα (context) σε επίπεδο χρήστη στα μέσα κοινωνικής δικτύωσης.

Η ΑΣ μπορεί γενικά να γίνει σε 3 επίπεδα [4]:

- Επίπεδο εγγράφου (Document-level)
- Επίπεδο πρότασης (Sentence-level)
- Επίπεδο πτυχής (Aspect-level)

Στο επίπεδο εγγράφου το συναίσθημα εξάγεται συνολικά από όλο το κείμενο. Αυτό είναι μια αρκετά απλοποιημένη προσέγγιση καθώς είναι συχνό, ειδικά σε κριτικές προϊόντων να εκφράζονται διαφορετικές γνώμες σχετικά με διάφορες πτυχές. Για παράδειγμα, σε μια κριτική ενός κινητού ο ιδιοκτήτης του μπορεί να εκφράστηκε θετικά σχετικά με την επεξεργαστική του ισχύ και τον αποθηκευτικό χώρο, αλλά να ήταν απογοητευμένος από την μπαταρία. Η ΑΣ σε επίπεδο εγγράφου αυτό που θα κατορθώσει είναι να εξάγει το συνολικό “συναίσθημα” που αποκόμισε ο χρήστης από το κινητό, αδιαφορώντας για τις λεπτομέρειες. Είναι η απλούστερη μορφή ΑΣ και αν και όχι βέλτιστη, δίνει γενικά καλά αποτελέσματα, όπως στα [6], [7]. Η παρούσα εργασία θα πραγματοποιήσει ΑΣ σε επίπεδο εγγράφου.

Στο επίπεδο πρότασης επιλέγονται προτάσεις από το κείμενο και εξάγεται το συναίσθημα που σχετίζεται με κάθε μία από αυτή. Η ανάλυση αυτή είναι πιο απαιτητική από την προηγούμενη περίπτωση και συνδέεται με την ταξινόμηση υποκειμενικότητας (subjectivity classification) [8] όπου το σύστημα προσπαθεί να προσδιορίσει αν αυτό που εκφράζεται στο κείμενο είναι ένα αντικειμενικό γεγονός ή μια αυστηρά προσωπική άποψη του συγγραφέα.

Τέλος, στο επίπεδο πτυχής γίνεται μια προσπάθεια από το σύστημα να εντοπιστούν όλες οι πτυχές του προϊόντος που περιγράφονται μαζί με το συναίσθημα με το οποίο σχετίζονται. Στο παράδειγμα της κριτικής για ένα κινητό, σκοπός του συστήματος θα ήταν ο εντοπισμός όλων των χαρακτηριστικών για τα οποία εξέφρασε γνώμη ο ιδιοκτήτης (μπαταρία, κάμερα, επεξεργαστής, πιθανώς και άλλα) και η εξαγωγή των αντίστοιχων συναισθημάτων. Είναι η πιο αναλυτική μορφή ΑΣ και συνάμα η πιο δύσκολη, καθώς χρειάζεται αρκετά εξειδικευμένες αρχιτεκτονικές νευρωνικών δικτύων για καλά αποτελέσματα. Ένα άλλο πρόβλημα είναι και η μορφή των κριτικών καθώς πολλές φορές δυσχεραίνεται η λειτουργία του συστήματος από την περιπλοκότητα της φυσικής γλώσσας. Σχετικό παράδειγμα τέτοιας ανάλυσης και αρχιτεκτονικής που την επιτυγχάνει υπάρχει στο [9].

1.2.2 Εφαρμογές της ΑΣ

Τα τελευταία χρόνια η ΑΣ έχει αρχίσει να χρησιμοποιείται όλο και περισσότερο λόγω της εύκολης πρόσβασης στο διαδίκτυο και του σημαντικού όγκου δεδομένων που κυκλοφορεί και δημιουργείται καθημερινά σε αυτό.

Ένας σημαντικός τομέας που βρίσκει απήχηση η ΑΣ είναι η υποστήριξη πελατών και συγκεκριμένα η άποψη των πελατών (customer feedback). Οι επιχειρήσεις χρησιμοποιούν αυτοματοποιημένα συστήματα τα οποία επιτρέπουν στους χρήστες να υποβάλλουν τις γνώμες τους για τα προϊόντα ή τις υπηρεσίες που χρησιμοποίησαν, είτε σε κάποια φόρμα είτε σε email. Το σύστημα έπειτα εκτελεί την ανάλυση και τροφοδοτεί την επιχείρηση με στατιστικά σχετικά με την ικανοποίηση των πελατών. Η όλη διαδικασία γίνεται αυτόματα, και έτσι υπάρχει σημαντικό κέρδος σε ανθρώπινους

πόρους και οικονομικό κεφάλαιο. Σύνολα δεδομένων που δημιουργήθηκαν με τέτοιο τρόπο χρησιμοποιούνται και στην παρούσα εργασία.

Μία παραλλαγή και προέκταση του προηγούμενου παραδείγματος είναι η χρήση της ΑΣ σε συστήματα προτάσεων (recommender systems) ως συστατικό μάρκετινγκ. Καθώς ο χρήστης περιηγείται στο διαδίκτυο ή αγοράζει προϊόντα σε ένα ηλεκτρονικό καλάθι, του εμφανίζονται συστάσεις για άλλα προϊόντα που ενδέχεται να του αρέσουν, σύμφωνα με όσα έχει αγοράσει έως τώρα ή τις ιστοσελίδες που έχει δει. Η συνεισφορά της ΑΣ εδώ είναι πως ο χρήστης ερωτάται για την ικανοποίηση του μέχρι τη στιγμή της πρότασης νέων αντικειμένων, και ανάλογα με τις απαντήσεις του το σύστημα εκτελεί μια ανάλυση. Αυτό μπορεί να γίνει και με την μορφή μιας σύντομης κριτικής την οποία υποβάλλει ο χρήστης. Ένα τέτοιο παράδειγμα παρουσιάζεται στο [10] όπου συνδυάζεται η ΑΣ και το συνεργατικό φιλτράρισμα (collaborative filtering).

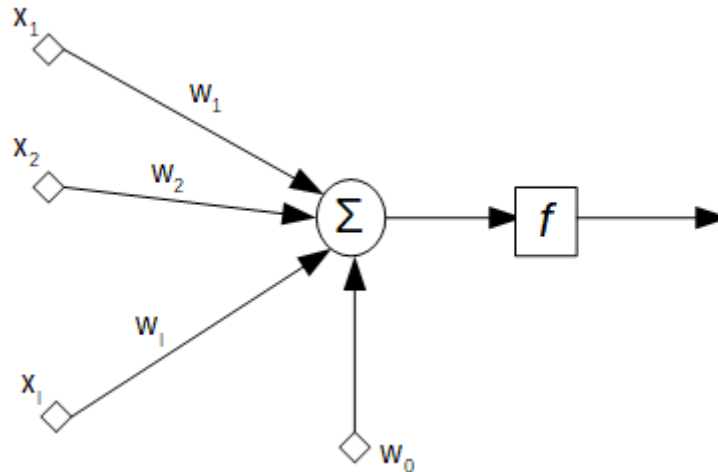
Με την ραγδαία εξάπλωση των κοινωνικών δικτύων η ΑΣ είναι ιδιαίτερα χρήσιμη στην ανάλυση δεδομένων που έχουν προέλθει από αυτά. Χαρακτηριστικό παράδειγμα είναι η ανάλυση των πολιτικών προτιμήσεων των χρηστών κοινωνικών δικτύων όπως στο [11], ακόμα και για την πρόβλεψη εκλογικών αποτελεσμάτων. Σημαντικό επίσης παράδειγμα χρήσης της ΑΣ είναι η ανίχνευση καταθλιπτικών συμπεριφορών όπως γίνεται στο [12]. Η χρήση τέτοιων συστημάτων από κοινωνικούς φορείς μπορεί να έχει σημαντικά θετικά κοινωνικά αποτελέσματα. Φυσικά, τα κοινωνικά δίκτυα μπορούν να συνδυαστούν με το πρώτο παράδειγμα που αναφέραμε. Δεν είναι σπάνιο οι εταιρείες να ζητούν στο Twitter την γνώμη των χρηστών και να χρησιμοποιούν αυτοματοποιημένα συστήματα ΑΣ στα απαντητικά τιτιβίσματα τους.

Τέλος, άλλη μία εφαρμογή των συστημάτων ΑΣ είναι η χρήση τους σε διαλογικά συστήματα και έξυπνους βοηθούς. Πράκτορες που χρησιμοποιούν την ΑΣ είναι σε θέση να κατανοούν καλύτερα τις ανάγκες των χρηστών, και έτσι υπάρχει καλύτερη αλληλεπίδραση [13].

2. ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

2.1 Επισκόπηση λειτουργίας ΤΝΔ

Το βασικό δομικό στοιχείο των απλών Τεχνητών Νευρωνικών Δικτύων- ΤΝΔ (artificial neural network- ANN) είναι ο τεχνητός νευρώνας (perceptron) που φαίνεται στην ακόλουθη εικόνα, η αρχική μορφή του οποίου προτάθηκε στο [14]. Αυτός βασίζεται στον νευρώνα των McCulloch και Pitts [18]. Η έμπνευση για τους τεχνητούς νευρώνες προήλθε από την μελέτη των βιολογικών νευρώνων του ανθρώπινου εγκεφάλου, που λειτουργούν με παρόμοιο τρόπο.



Εικόνα 2.1: Μοντέλο απλού τεχνητού νευρώνα. Πηγή: [2], σελ.109

Οι συνιστώσες του διανύσματος εισόδου (όπως περιγράψαμε στην ενότητα 1) πολλαπλασιάζονται με τα αντίστοιχα βάρη και στη συνέχεια αθροίζονται. Δηλαδή αν y η έξοδος του δικτύου, έχουμε:

$$y = \sum_{i=1}^l x_i w_i + w_0$$

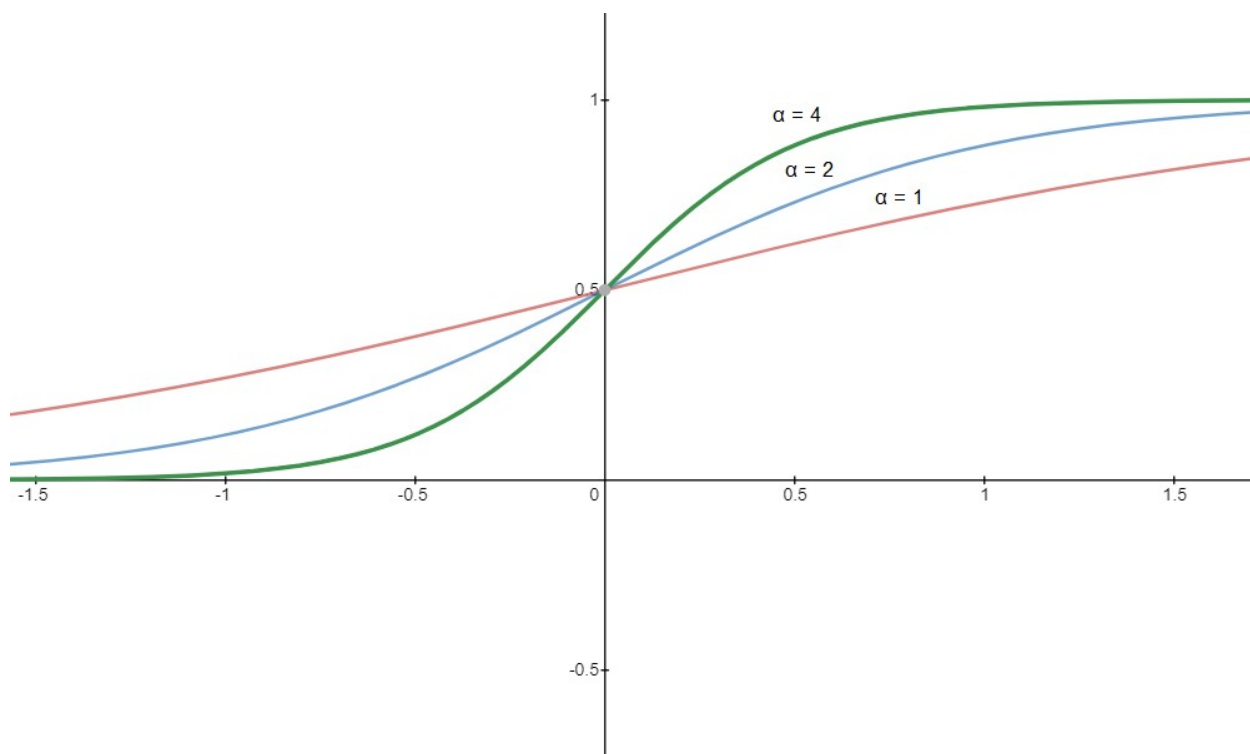
Το w_0 είναι μια ποσότητα που ονομάζεται πόλωση (bias), Συνήθως ενσωματώνεται στους υπολογισμούς, προσαυξάνοντας το διάνυσμα εισόδου με την μονάδα. Έπειτα σε αυτό το αποτέλεσμα εφαρμόζεται μια συνάρτηση που καλείται συνάρτηση ενεργοποίησης (activation function). Μερικές από τις κλασικές περιπτώσεις συναρτήσεων ενεργοποίησης είναι η βηματική ($f(x) = -1$ για $x < 0$ και $f(x) = 1$ για $x > 0$, δεν ορίζεται στο 0 συνήθως), η λογιστική (logistic function):

$$f(x) = \frac{1}{1 + e^{-ax}}$$

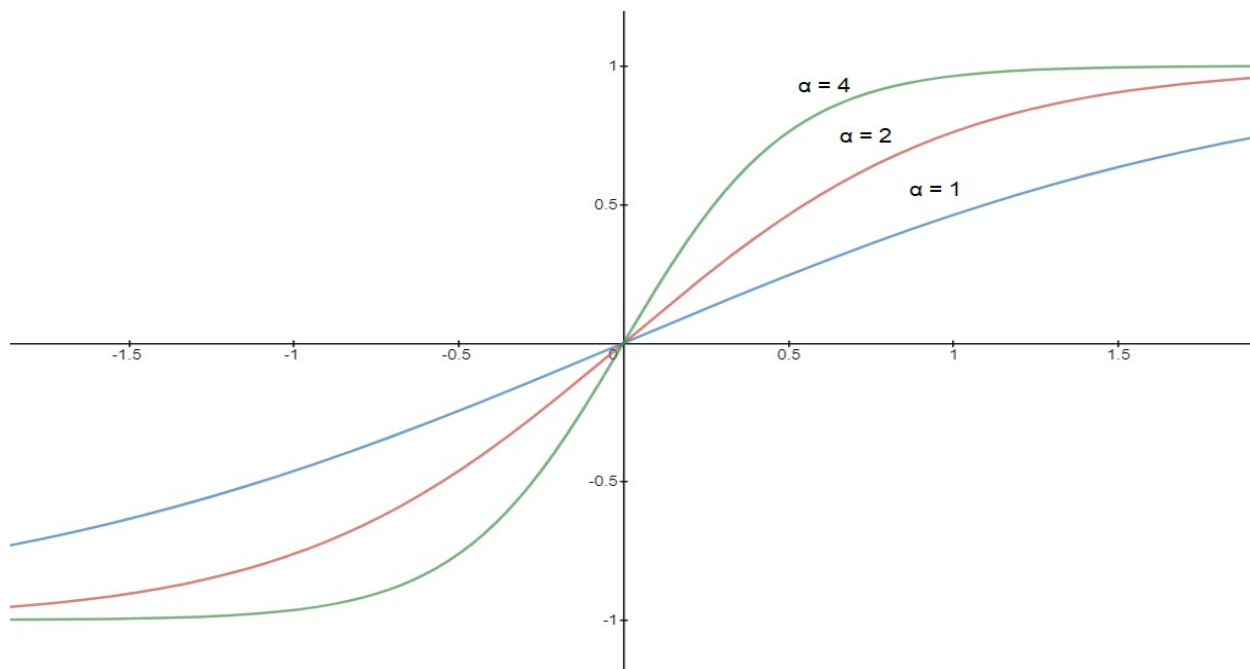
και η συνάρτηση υπερβολικής εφαπτομένης (hyperbolic tangent function):

$$f(x) = c \tanh\left(\frac{ax}{2}\right) \quad [2]$$

Οι τελευταίες δύο συναρτήσεις ανήκουν στην κατηγορία των σιγμοειδών (sigmoid) συναρτήσεων και έχουν χρήσιμες ιδιότητες κατά την εκπαίδευση, όπως παραγωγισιμότητα.



Εικόνα 2.2: Λογιστική συνάρτηση

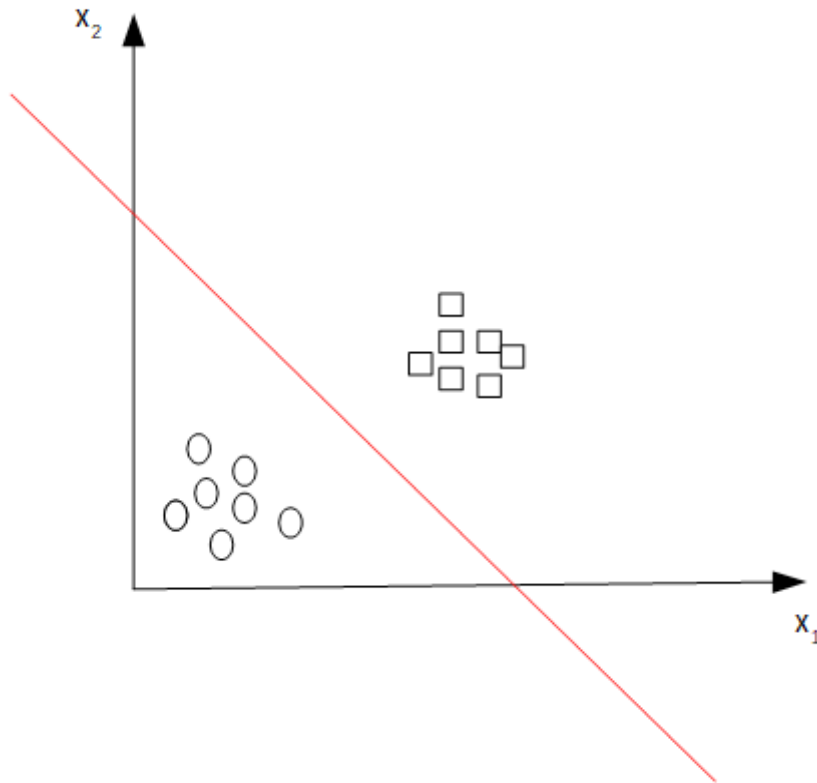


Εικόνα 2.3: Συνάρτηση υπερβολικής εφαπτομένης

Ο τεχνητός νευρώνας μπορεί να κατασκευάσει ένα υπερεπίπεδο (hyperplane) της μορφής:

$$g(x) = w^T x + w_0 = 0$$

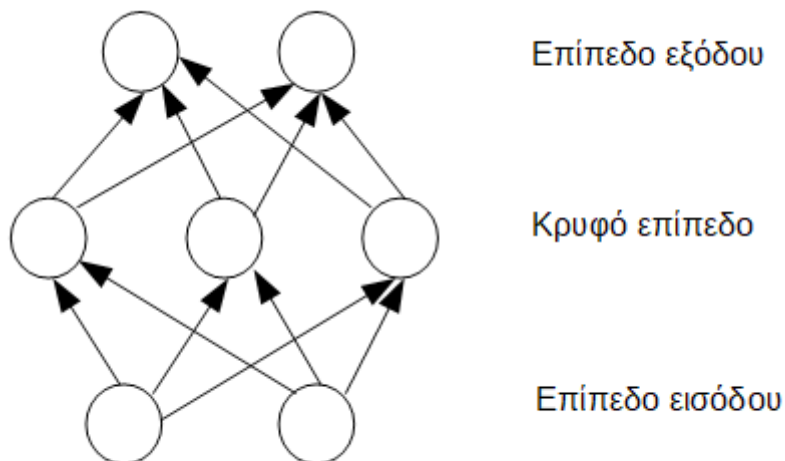
Αυτό το υπερεπίπεδο εκφράζει μια γραμμή που έχει την δυνατότητα να διαχωρίζει διανύσματα που ανήκουν σε δύο διαφορετικές κατηγορίες. Ένα παράδειγμα εμφανίζεται στην επόμενη εικόνα, για πρότυπα δύο διαστάσεων. Η ιδέα επεκτείνεται και σε περισσότερες διαστάσεις, που είναι εξάλλου και η συνηθισμένη περίπτωση.



Εικόνα 2.4: Διαχωριστική γραμμή ταξινόμησης για 2 κατηγορίες

Ο υπολογισμός των βαρών που ορίζουν την διαχωριστική γραμμή γίνεται μέσω ενός αλγορίθμου μάθησης. Ένας τέτοιος αλγόριθμος περιγράφεται στο [14].

Χρησιμοποιώντας πολλούς νευρώνες μπορούν να σχηματιστούν ΤΝΔ.



Εικόνα 2.5: Ένα Τεχνητό Νευρωνικό Δίκτυο με 3 βασικά επίπεδα

Το πρώτο επίπεδο είναι εκείνο στο οποίο εισάγονται τα δεδομένα προς επεξεργασία και ονομάζεται επίπεδο εισόδου. Μετά ακολουθούν ένα ή περισσότερα κρυφά επίπεδα τα οποία βοηθούν στην καλύτερη προσέγγιση του διαχωριστικού υπερεπιπέδου. Τέλος, στο επίπεδο εξόδου λαμβάνουμε την συνολική απόκριση του δικτύου σε συνάρτηση με την είσοδο. Σε αυτό το επίπεδο μοντελοποιούνται οι πραγματικές και επιθυμητές κατηγορίες, με την μορφή διανυσμάτων στο $\{0,1\}$. Για παράδειγμα, μια δυνατή έξοδος του δικτύου θα μπορούσε να είναι η $[1,0]$ που σηματοδοτεί ότι ο πρώτος νευρώνας στο επίπεδο εξόδου ενεργοποιείται ενώ ο δεύτερος όχι.

Η πιο συνηθισμένη περίπτωση είναι όλοι οι νευρώνες κάθε επιπέδου να συνδέονται με όλους του επόμενου επιπέδου. Τέτοια δίκτυα ονομάζονται πλήρως συνδεδεμένα, και όταν η ροή πληροφορίας είναι μόνο προς τα μπροστά, εμπρόσθιας τροφοδότησης (feedforward). Δίκτυα στα οποία υπάρχουν συνδέσεις μεταξύ νευρώνων του ίδιου επιπέδου ονομάζονται αναδρομικά (recurrent). Μία κατηγορία τους, οι ΑΜΠ περιγράφονται σε επόμενη ενότητα και χρησιμοποιούνται στην παρούσα εργασία.

Μπορεί να αποδειχτεί [16] ότι τα ΤΝΔ με ένα κρυφό επίπεδο και σιγμοειδείς συναρτήσεις ως συνάρτηση ενεργοποίησης, μπορούν να προσεγγίσουν συνεχείς συναρτήσεις, και επομένως μπορούν να χρησιμοποιηθούν για ταξινόμηση.

Τα ΤΝΔ και οι παραλλαγές τους (ΣΝΔ, ΑΜΠ) εκπαιδεύονται με τον αλγόριθμο της οπισθοδιάδοσης [17]. Η βασική ιδέα πίσω από τον αλγόριθμο είναι η διαρκής αναπροσαρμογή των βαρών μέσω της παραγωγού (συνήθως χρησιμοποιείται η μέθοδος καθόδου με βάση την κλίση- gradient descent) ώστε να ελαχιστοποιείται το σφάλμα μεταξύ της επιθυμητής εξόδου του δικτύου και της πραγματικής. Αυτό εκφράζεται μέσα από μια συνάρτηση απώλειας (loss function).

2.2 Συνελικτικά Νευρωνικά Δίκτυα

Τα ΣΝΔ είναι ένας ιδιαίτερος τύπος νευρωνικών δικτύων που είναι πολύ αποτελεσματικά όταν χρησιμοποιούνται για δεδομένα που μπορούν να έχουν μια χωροταξική αναπαράσταση. Χρησιμοποιήθηκαν για ταξινόμηση για πρώτη φορά το 1989 [20], ενώ βασίζονται στο neocognitron [21]. Κλασική περίπτωση χρήσης τους είναι η ανάλυση εικόνων, αλλά με κατάλληλη επεξεργασία, όπως θα δούμε στο κεφάλαιο 4, μπορούν να χρησιμοποιηθούν και στην ανάλυση κειμένων. Η βασική έννοια με την οποία σχετίζονται είναι η πράξη της συνέλιξης.

Η συνέλιξη μπορεί να είναι είτε διακριτή είτε συνεχής. Σύμφωνα με το [19], η διακριτή συνέλιξη (discrete convolution) ορίζεται ως:

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m]$$

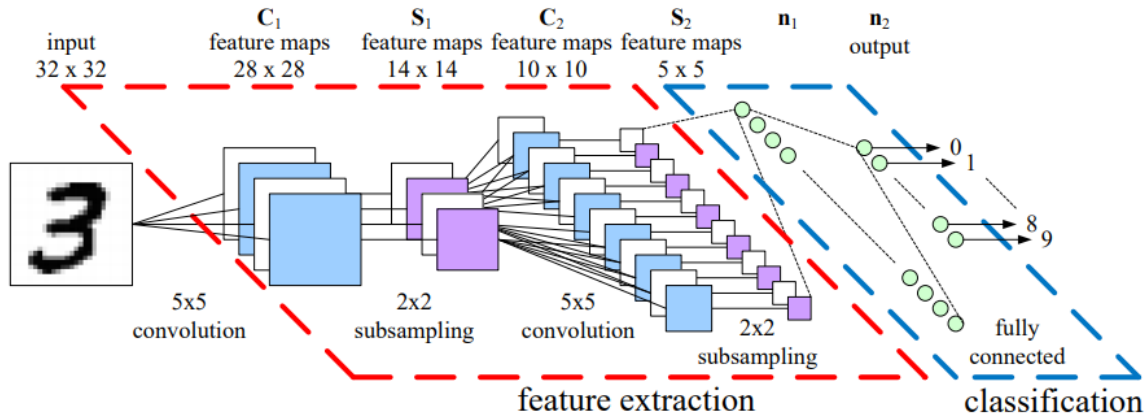
Ενδιαφέρον χαρακτηριστικό της συνέλιξης είναι η ιδιότητα της αντιμεταθετικότητας:

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[n-m]g[m]$$

Στην περίπτωση των ΣΝΔ, η συνάρτηση f αντιστοιχεί στον πίνακα που δίνεται ως είσοδο στο δίκτυο, και η συνάρτηση g καλείται πυρήνας (kernel) ή φίλτρο.

Θα περιγράψουμε την λειτουργία των ΣΝΔ με ένα παράδειγμα χρησιμοποιώντας την εικόνα από το [22] για την αναγνώριση χειρόγραφων ψηφίων. Στην περίπτωση των κειμένων όπως για την ΑΣ δεν υπάρχουν σημαντικές μεταβολές, καθώς το μόνο που χρειάζεται να γίνει είναι να μετασχηματιστούν τα δεδομένα κατάλληλα όπως θα

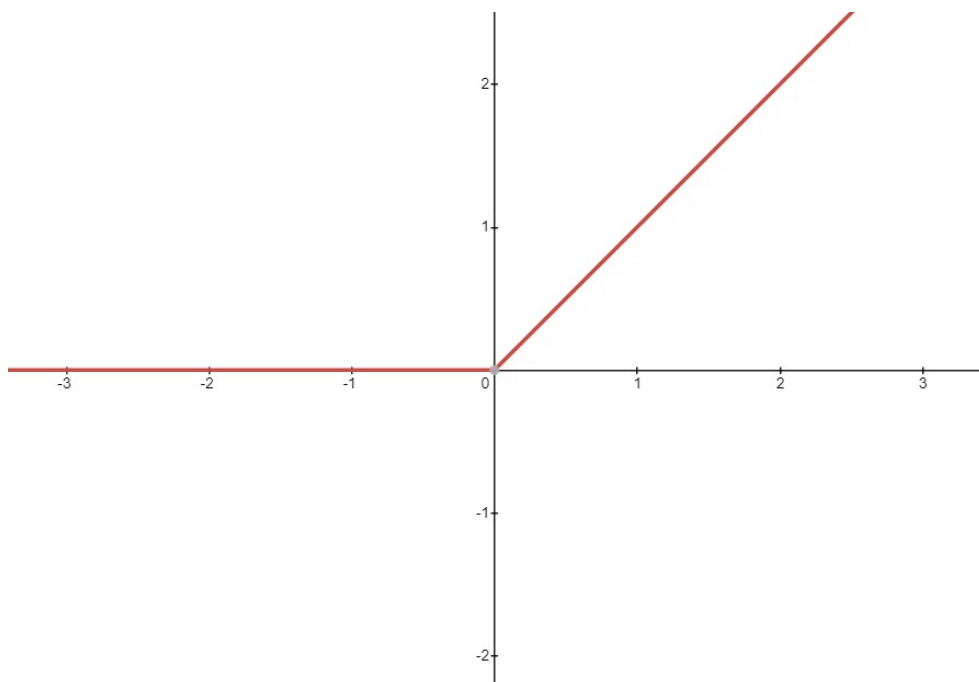
περιγράψουμε στο κεφάλαιο 4. Οι αλλαγές θα είναι στην είσοδο και στην έξοδο καθώς οι υπόλοιπες παράμετροι του δικτύου δεν εξαρτώνται άμεσα από την εφαρμογή.



Εικόνα 2.6: Ένα ΣΝΔ για αναγνώριση χειρόγραφων ψηφίων [22]

Στο δίκτυο δίνονται ως εισόδοι δυαδικές εικόνες σε μορφή πινάκων $N \times M$, οι οποίες σχηματίζουν τα διανύσματα εισόδου. Οι βασικές λειτουργίες που εκτελεί το δίκτυο είναι:

- Συνέλιξη. Το αποτέλεσμα της συνέλιξης του διανύσματος εισόδου με τους πυρήνες ονομάζεται και χάρτης χαρακτηριστικών (feature map).
- Χρήση του γραμμικού ανορθωτή (rectified linear unit – ReLU) στον χάρτη χαρακτηριστικών ως συνάρτηση ενεργοποίησης. Αυτός ορίζεται ως $f(x) = \max(0, x)$.
- Υποδειγματοληψία μέσω της συγκέντρωσης (pooling).
- Ταξινόμηση στο επίπεδο εξόδου, συνήθως (αλλά όχι απαραίτητα) με χρήση πλήρως διασυνδεδεμένων επιπέδων.



Εικόνα 2.7: Συνάρτηση ενεργοποίησης γραμμικού ανορθωτή (ReLU)

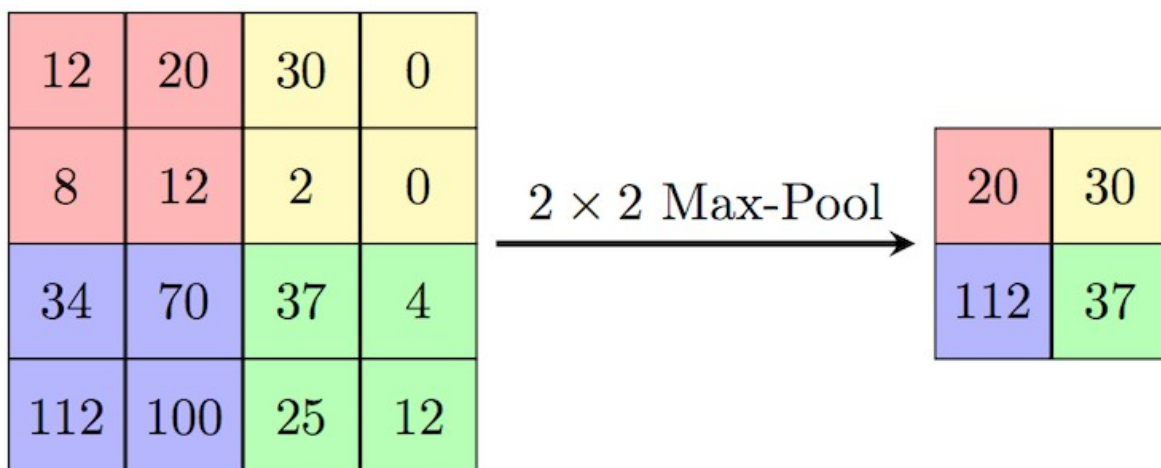
Είναι συχνό το φαινόμενο ένα ΣΝΔ να αποτελείται από πολλά επίπεδα συνέλιξης και συγκέντρωσης. Τέτοια δίκτυα αποκαλούνται βαθιά δίκτυα (deep networks) και η μάθηση με αυτά βαθιά μάθηση (deep learning).

Σε αυτό το σημείο θα εξηγήσουμε καλύτερα τι λειτουργία επιτελεί το επίπεδο συγκέντρωσης. Ας υποθέσουμε πως ο χάρτης χαρακτηριστικών που προέκυψε από την συνέλιξη είναι ο εξής:

12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

Εικόνα 2.8: Χάρτης χαρακτηριστικών για το παράδειγμα μας [23]

Μία σημαντική παράμετρος για την συγκέντρωση είναι το βήμα (stride). Αυτό καθορίζει κατά πόσα εικονοστοιχεία θα πραγματοποιείται υποδειγματοληψία. Αν υποθέσουμε πως πραγματοποιούμε υποδειγματοληψία μεγίστου ανά 2 pixel τότε θα γίνουν οι ακόλουθες αντιστοιχίες:



Εικόνα 2.9: Συγκέντρωση μεγίστων για τον χάρτη χαρακτηριστικών [23]

Δηλαδή εξετάζουμε τον χάρτη χαρακτηριστικών σε περιοχές-παράθυρα των 2×2 εικονοστοιχείων, και από κάθε τετράδα διατηρούμε τον μέγιστο αριθμό. Με αυτό τον τρόπο, εκτός από μείωση διάστασης πετυχαίνουμε και την αξιοποίηση της χωρικής πληροφορίας κάθε περιοχής, διατηρώντας το σημαντικότερο στοιχείο. Εάν λείπει πληροφορία στις άκρες, τότε συνήθως συμπληρώνεται με την διαδικασία γεμίσματος

(padding) η οποία προσθέτει είτε μηδενικά είτε κάποια άλλη τιμή που ορίζεται από παράμετρο.

Το αποτέλεσμα της υποδειγματοληψίας συνήθως συνδέεται σε ένα κρυφό επίπεδο όπως αυτά των κλασικών ΤΝΔ αλλά αυτό δεν είναι απαραίτητο. Εάν έχουμε παρόμοια ή ίδια απόδοση χωρίς το κρυφό επίπεδο, μπορούμε να χρησιμοποιήσουμε το επίπεδο εξόδου αμέσως μετά το επίπεδο υποδειγματοληψίας για να αποφύγουμε περιττούς υπολογισμούς.

Για το παράδειγμα της αναγνώρισης χειρόγραφων ψηφίων που αναφέραμε παραπάνω, καθώς και στις περισσότερες εφαρμογές ΕΦΓ όπως η ΑΣ, οι τιμές που έχουν οι νευρώνες εξόδου έπειτα από την εφαρμογή της συνάρτησης ενεργοποίησης ερμηνεύονται ως πιθανότητες. Στην συνηθισμένη περίπτωση όπου η επιθυμητή κατηγορία είναι μόνο μία μεταξύ 3 ή περισσότερων κατηγοριών (multiclass classification, για παράδειγμα ένα ψηφίο από τα 10 ή ένα συναίσθημα από τα 3 βασικά-θετικό, αρνητικό, ουδέτερο), χρησιμοποιείται η λεγόμενη συνάρτηση ενεργοποίησης softmax [24]:

$$y_k = \frac{e^{(a_k)}}{\sum_j e^{(a_j)}}$$

όπου a το διάνυσμα που δίνεται ως είσοδος στο νευρώνα εξόδου. Ο παρονομαστής εκφράζει την άθροιση των συνιστωσών, σε εκθετική μορφή. Η συνάρτηση αυτή έχει την ικανότητα να κανονικοποιεί το αποτέλεσμα στο διάστημα $[0,1]$, και μάλιστα οι νευρώνες εξόδου θα αθροίζονται στη μονάδα. Αυτό έρχεται σε πλήρη συμφωνία με αυτό που αναφέραμε πριν, πως οι έξοδοι του ΣΝΔ θα ερμηνεύονται ως πιθανότητες. Ο νευρώνας που θα ενεργοποιηθεί θα είναι αυτός με την μεγαλύτερη πιθανότητα. Μάλιστα, η συνάρτηση softmax χρησιμοποιείται πολύ συχνά με την συνάρτηση κόστους διεντροπίας (cross-entropy). Για την περίπτωση των πολλών κατηγοριών αυτή μπορεί να οριστεί με τον τύπο:

$$CE = -\sum_{i=1}^C y_i \ln(y'_i)$$

όπου y είναι η έξοδος του ΣΝΔ και y' η επιθυμητή έξοδος, και C ο αριθμός των κλάσεων. Αυτός ο τύπος ισχύει για ένα πρότυπο. Συνήθως στην μηχανική μάθηση η εκπαίδευση γίνεται σε κύκλους που ονομάζονται εποχές (epochs). Μέσα σε μια εποχή το ΣΝΔ μπορεί να τροφοδοτείται με κάθε δείγμα και να ανανεώνει τα βάρη του μετά την εμφάνιση του, ή (πιο συνηθισμένα) να δίνονται στο δίκτυο ένας καθορισμένος αριθμός δειγμάτων, προτού εκτελεστεί η διαδικασία ανανέωσης βαρών. Η δεύτερη περίπτωση ονομάζεται και εκπαίδευση με δέσμες (batch mode). Τότε μπορούμε να μεταβάλλουμε τον τύπο της συνάρτησης κόστους:

$$CE = \frac{-1}{N} \sum_N \sum_{i=1}^C y_i \ln(y'_i)$$

δηλαδή το κόστος υπολογίζεται συνολικά για όλα τα δείγματα της δέσμης. Εάν έχουμε μόνο δύο κλάσεις (binary cross-entropy, BCE) ο τύπος μπορεί να εκφυλιστεί ως εξής:

$$BCE = \frac{-1}{N} \sum_N [y \ln(y') + (1-y) \ln(1-y')]$$

Αυτός ο τύπος χρησιμοποιείται σε συνδυασμό με μια σιγμοειδή συνάρτηση. Για την αποφυγή ενδεχομένων προβλημάτων με τους υπολογισμούς των λογαρίθμων, υιοθετείται η παραδοχή $0 \ln 0 = 0$.

Ο συνδυασμός της συνάρτησης κόστους διεντροπίας με σιγμοειδείς συναρτήσεις ενεργοποίησης οδηγεί σε μία μορφή λογιστικής παλινδρόμησης [25] (logistic regression). Εκτιμάται δηλαδή η πιθανοτική κατανομή του μοντέλου του ΣΝΔ και ενεργοποιείται ο νευρώνας με την μέγιστη πιθανότητα.

Εκτός από την χρήση τους στην επεξεργασία εικόνας, τα ΣΝΔ έχουν δώσει πολύ καλά αποτελέσματα και στην ΑΣ όπως στο [26], και είναι ο ένας από τους 2 τύπους ΤΝΔ που θα χρησιμοποιήσουμε στην εργασία αυτή. Εκπαιδεύονται σύμφωνα με τον αλγόριθμο της οπισθοδιάδοσης.

2.3 Αναδρομικά Νευρωνικά Δίκτυα

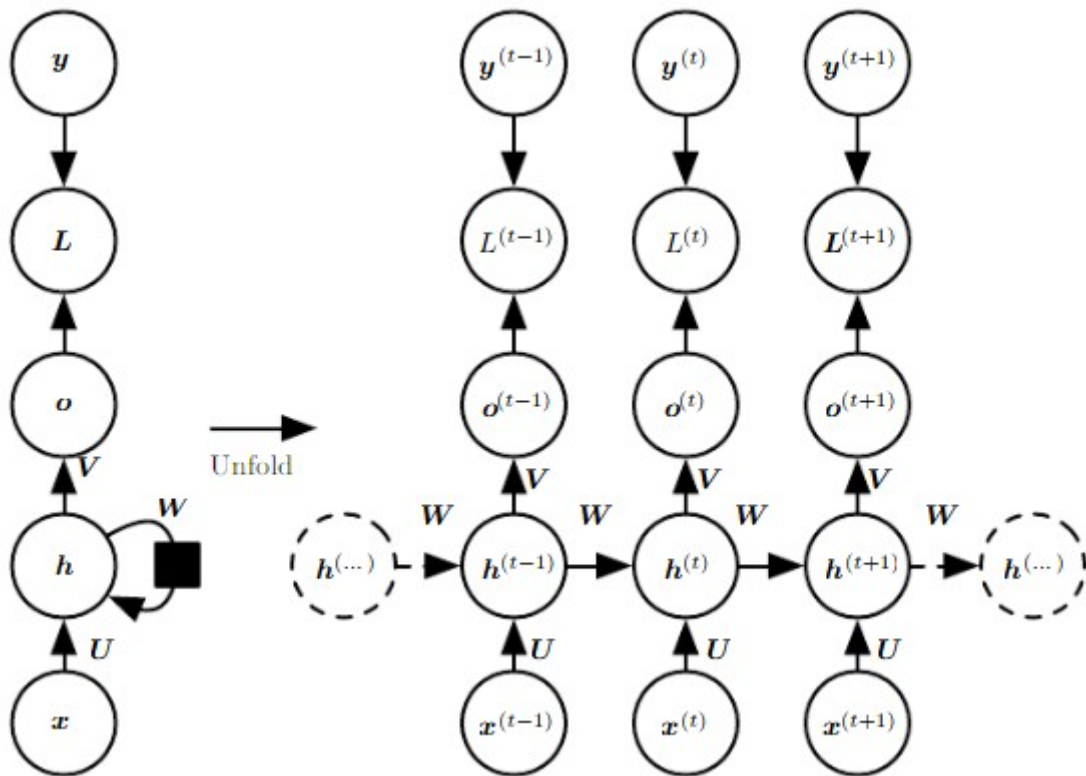
2.3.1 Βασικές έννοιες

Σε αντίθεση με τα ΣΝΔ και τα απλά ΤΝΔ, τα Αναδρομικά Νευρωνικά Δίκτυα- ΑΝΔ (Recurrent Neural Networks-RNN) περιέχουν συνδέσεις μεταξύ νευρώνων του ίδιου επιπέδου. Έτσι μπορούν να μοντελοποιήσουν ακολουθίες όπως κείμενα πολύ αποτελεσματικά. Ως παράδειγμα, ας υποθέσουμε πως έχουμε την πρόταση “Αυτός ο σκηνοθέτης δεν είναι καλός”. Ιδανικά, ένα δίκτυο θα έπρεπε να μπορεί να καταλάβει αφενός πως η λέξη καλός αναφέρεται στον σκηνοθέτη, και αφετέρου πως πριν την λέξη “καλός” προηγείται η λέξη “δεν”. Ένα ΑΝΔ μπορεί να το πετύχει αυτό, χρησιμοποιώντας την έννοια των χρονικών βημάτων (timesteps), με κάθε χρονικό βήμα να αναφέρεται σε μια λέξη (αλλά όχι απαραίτητα), ή γενικότερα ένα συγκεκριμένο σημείο της ακολουθίας .

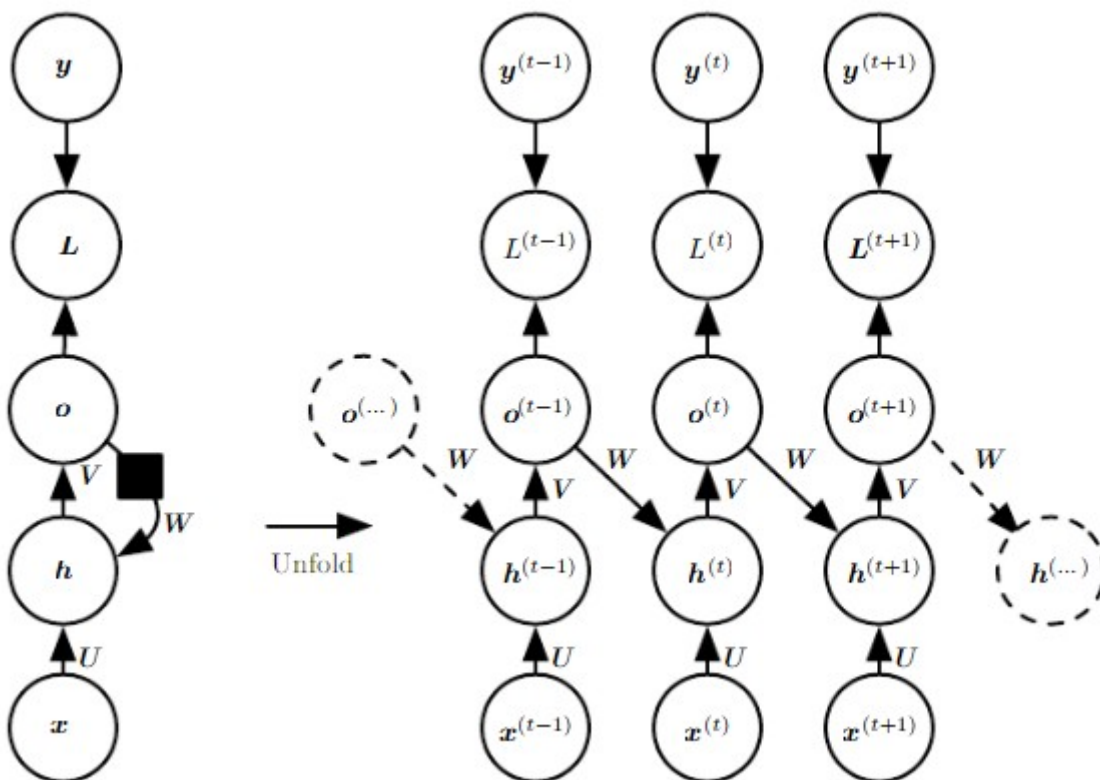
Ανεξαρτήτως της ειδικότερης αρχιτεκτονικής τους, οι πιο συνηθισμένες κατηγορίες τους είναι [27]:

- ΑΝΔ που παράγουν μία έξοδο σε κάθε χρονικό βήμα, και έχουν συνδέσεις μεταξύ των νευρώνων του ίδιου κρυφού επιπέδου.
- ΑΝΔ που παράγουν μία έξοδο σε κάθε χρονικό βήμα, αλλά η αναδρομικότητα προέρχεται από τη σύνδεση της εξόδου του χρονικού βήματος με την κρυφή κατάσταση του επόμενου.
- ΑΝΔ που παράγουν μόνο μία έξοδο, αφού πρώτα επεξεργαστούν όλη την ακολουθία εισόδου.

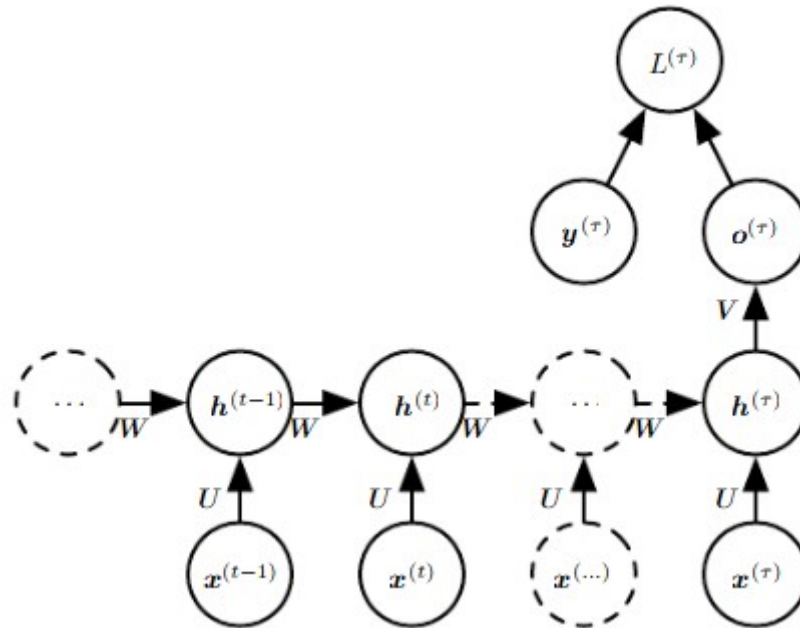
Τα ΑΝΔ μπορούν να οπτικοποιηθούν είτε ως ένας γράφος με μια μήτρα βαρών που επαναλαμβάνεται για συγκεκριμένο αριθμό χρονικών βημάτων, είτε να ξεδιπλωθούν ώστε να φαίνονται όλες οι εμπλεκόμενοι παράμετροι σε κάθε χρονικό βήμα. Αυτό φαίνεται στις εικόνες που ακολουθούν, οι οποίες δείχνουν τις 3 κατηγορίες που μόλις αναφέρθηκαν.



Εικόνα 2.10: ΑΝΔ με έξοδο σε κάθε χρονικό βήμα [27]



Εικόνα 2.11: ΑΝΔ με σύνδεση της εξόδου με το επόμενο χρονικό βήμα [27]



Εικόνα 2.12: ΑΝΔ με έξοδο μόνο στο τελευταίο χρονικό βήμα [27]

Στους γράφους αυτούς:

- W είναι η μήτρα βαρών που διαδίδεται στους νευρώνες του κρυφού επιπέδου
- U είναι η μήτρα βαρών μεταξύ των εισόδων και του κρυφού επιπέδου
- V είναι η μήτρα βαρών μεταξύ κρυφού επιπέδου και εξόδου
- $o^{(t)}$ είναι η έξοδος στο χρονικό βήμα t
- $h^{(t)}$ είναι η κρυφή κατάσταση στο χρονικό βήμα t
- $y^{(t)}$ είναι η επιθυμητή έξοδος
- $L^{(t)}$ είναι η συνάρτηση κόστους, όπως αυτή της διεντροπίας που περιγράψαμε πριν

Είναι σημαντικό πως η μήτρα βαρών W διαδίδεται στους νευρώνες του ίδιου επιπέδου. Μέσω αυτής της παραμέτρου υπάρχει διάδοση πληροφορίας και έτσι το δίκτυο μπορεί να κάνει συσχέτιση της πληροφορίας την χρονική στιγμή $t+1$, με αυτή που εμφανίστηκε την χρονική στιγμή $t-1$. Θυμίζουμε εδώ πως η έννοια “χρόνος” δεν είναι κυριολεκτική, αλλά μεταφορική και αναφέρεται στα τμήματα της ακολουθίας.

Μπορούμε λοιπόν να γράψουμε [27] πως:

$$h^{(t)} = f(h^{(t-1)}, x^{(t)}; \theta)$$

Δηλαδή η κατάσταση του δικτύου σε ένα δεδομένο χρονικό βήμα εξαρτάται από την προηγούμενη κρυφή κατάσταση και την τρέχουσα είσοδο, ως προς μια παράμετρο θ (όπως είναι το βάρος W). Πιο αναλυτικά:

$$a^{(t)} = b + Wh^{(t-1)} + Ux^{(t)}$$

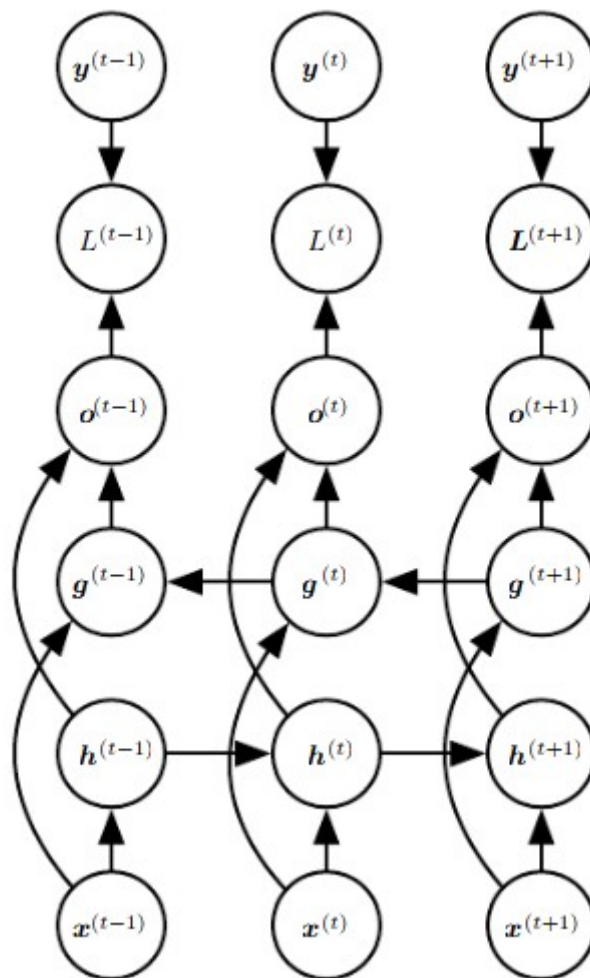
$$h^{(t)} = \tanh(a^{(t)})$$

$$o^{(t)} = c + Vh^{(t)}$$

$$y^{(t)} = \text{softmax}(o^{(t)})$$

Αυτές οι εξισώσεις χρησιμοποιούνται από $t = 1$ έως μια χρονική στιγμή $t = \tau$.

Εκτός από αυτές τις περιπτώσεις, υπάρχει και η περίπτωση να θέλουμε το δίκτυο να μπορεί να εκμεταλλευτεί την διάδοση πληροφορίας και προς τα πίσω. Τέτοια δίκτυα παρουσιάστηκαν το 1997 [28] και λέγονται αμφίδρομα (bidirectional). Πρακτικά, είναι συνδυασμός ενός ΑΝΔ με ροή πληροφορίας προς τα μπροστά, και ενός ΑΝΔ με ροή πληροφορίας προς τα πίσω.



Εικόνα 2.13: Αμφίδρομο ΑΝΔ [27]. Διακρίνεται η ροή πληροφορίας προς δύο κατευθύνσεις

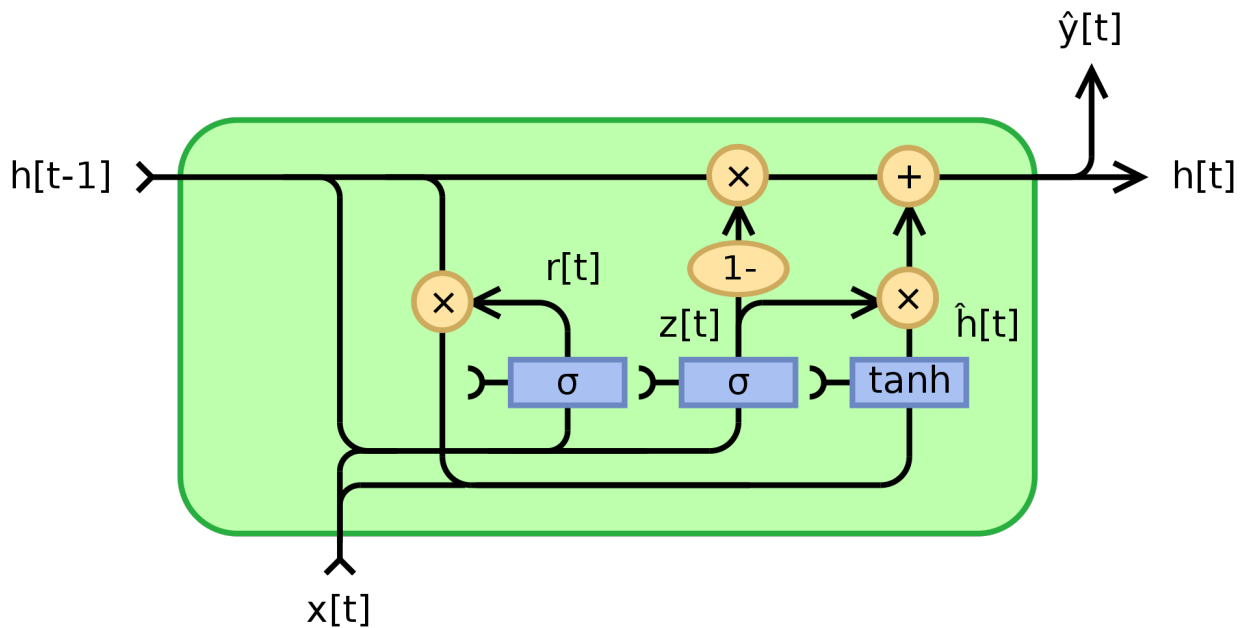
Τα αμφίδρομα ΑΝΔ έχουν παρουσιάσει καλά αποτελέσματα στην ΑΣ [29], και τα χρησιμοποιούμε και στην εργασία μας.

Τα ΑΝΔ εκπαιδεύονται με τον αλγόριθμο οπισθοδιάδοσης μέσα από τον χρόνο (Backpropagation through time -BPTT) [31]. Αρχικά ξετυλίγεται το δίκτυο σε έναν συγκεκριμένο αριθμό χρονικών βημάτων, και έπειτα γίνεται οπισθοδιάδοση του σφάλματος στα χρονικά βήματα.

2.3.2 Αναδρομικές Μονάδες με Πύλες

Οι ΑΜΠ είναι μια ειδική κατηγορία ΑΝΔ που παρουσιάστηκαν το 2014 [30]. Λειτουργούν με έναν μηχανισμό από πύλες ο οποίος ελέγχει την ροή της πληροφορίας μέσα στο δίκτυο, πετυχαίνοντας καλύτερη μοντελοποίηση ακολουθιών. Έχουν χρησιμοποιηθεί με μεγάλη επιτυχία σε εφαρμογές ΑΣ [32].

Υπάρχουν αρκετές παραλλαγές των ΑΜΠ αλλά η συνηθέστερη είναι αυτή που διακρίνεται στην εικόνα που ακολουθεί.



Εικόνα 2.14: Κλασική εκδοχή των ΑΜΠ. Διακρίνονται οι πύλες και οι πράξεις που λαμβάνουν χώρα [33]

Σε αυτό το μπλοκ διάγραμμα έχουμε δύο βασικές πύλες:

- Πύλη ενημέρωσης z (update gate). Αυτή η πύλη ελέγχει το πόση πληροφορία από την προηγούμενη κρυφή κατάσταση h θα μεταδοθεί στην τωρινή κρυφή κατάσταση. Είναι έτσι μια μορφή “μνήμης”
- Πύλη επαναφοράς r (reset gate). Αυτή η πύλη ελέγχει το κατά πόσο είναι σχετική η πληροφορία της προηγούμενης κρυφής κατάστασης, και η αντικατάσταση αυτής με πληροφορία από την τωρινή είσοδο.

Η λειτουργία της ΑΜΠ μπορεί να γίνει περισσότερο κατανοητή αν γράψουμε τις σχετικές εξισώσεις, σύμφωνα με το φορμαλισμό του διαγράμματος και το [30]. Για την χρονική στιγμή $t=0$ υποτίθεται πως η έξοδος της κρυφής κατάστασης είναι αρχικά 0:

$$r_t = \sigma(W_r x_t + U_r h_{t-1})$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1})$$

Το “ σ ” υποδηλώνει την σιγμοειδή συνάρτηση ενεργοποίησης. Όπως και στα απλά ΑΝΔ, οι όροι U και W περιγράφουν τα βάρη.

Τότε, η νέα κρυφή κατάσταση του δικτύου μπορεί να δοθεί ως εξής:

$$h_t = z_t h_{t-1} + (1 - z_t) h'_t$$

όπου:

$$h'_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1}))$$

Το σύμβολο \odot δηλώνει την πράξη του γινομένου Hadamard, όπου πραγματοποιείται πολλαπλασιασμός πινάκων στοιχείο προς στοιχείο. Στις εξισώσεις αυτές δύναται να προστεθεί και η παράμετρος της πόλωσης.

Όταν η πύλη επαναφοράς έχει τιμή κοντά στο 0, η ΑΜΠ αγνοεί την πληροφορία από την προηγούμενη κρυφή κατάσταση.

Τα κλασικά ΑΝΔ έχουν το πρόβλημα της παραγωγού που “εξαφανίζεται” (vanishing gradient problem) [34]. Δηλαδή, σε όσο μεγαλύτερο αριθμό χρονικών βημάτων εφαρμόζεται ο αλγόριθμος ΒΡΤΤ, τόσο μικρότερες γίνονται οι παράγωγοι λόγω των πολλαπλασιασμών που εμπλέκονται. Έτσι, παρουσιάζουν αδυναμία στο να απομνημονεύουν εξαρτήσεις με μεγάλη απόσταση από την τωρινή. Για παράδειγμα ένα κλασικό ΑΝΔ ανέλυε την ακόλουθη πρόταση:

“Compared to all his other works, this movie seemed terrible. This is strange, coming from a director with a high number of masterpieces”.

Το δίκτυο θα παρουσίαζε αδυναμία να καταλάβει σε ποιο πράγμα αναφέρεται η λέξη “masterpieces”, και αυτό θα είχε αντίκτυπο στα αποτελέσματα της ΑΣ. Αυτό το πρόβλημα περιορίζεται σημαντικά με την χρήση ΑΜΠ. Μέσω του μηχανισμού των πυλών, οι ΑΜΠ μπορούν να αναλύσουν πληροφορίες που έχουν πολύ μεγαλύτερη απόσταση μεταξύ τους, από ότι τα απλά ΑΝΔ.

Οι ΑΜΠ μπορούν να συνδυαστούν και με το επίπεδο συγκέντρωσης που περιγράψαμε για τα ΣΝΔ.

2.4 Άλλα ζητήματα μηχανικής μάθησης

2.4.1 Διαχωρισμός συνόλου δεδομένων

Όπως αναφέραμε στο πρώτο κεφάλαιο, στην μάθηση υπό επίβλεψη το σύστημα δέχεται πρότυπα μαζί με τις κατηγορίες στις οποίες ανήκουν. Ο σκοπός είναι αφού χρησιμοποιηθεί κάποιος αλγόριθμος για να μάθει τον τρόπο διαχωρισμού τους, να μπορεί να χρησιμοποιηθεί στη συνέχεια για ταξινόμηση νέων, αγνώστων προτύπων.

Ως εκ τούτου, πραγματοποιείται διαχωρισμός του συνόλου δεδομένων σε σύνολο εκπαίδευσης και σε σύνολο δοκιμής. Το σύστημα εκπαιδεύεται πάνω στο σύνολο εκπαίδευσης, και πραγματοποιεί έπειτα δοκιμές στο σύνολο δοκιμής. Δηλαδή, για να γίνουμε πιο συγκεκριμένοι, τα βάρη των νευρωνικών δικτύων που περιγράφηκαν προηγουμένως υπολογίζονται σύμφωνα με το σύνολο εκπαίδευσης. Έπειτα τα ίδια βάρη χρησιμοποιούνται για τον υπολογισμό της εξόδου του δικτύου, όταν παρουσιαστεί κάποιο πρότυπο από το σύνολο δοκιμής.

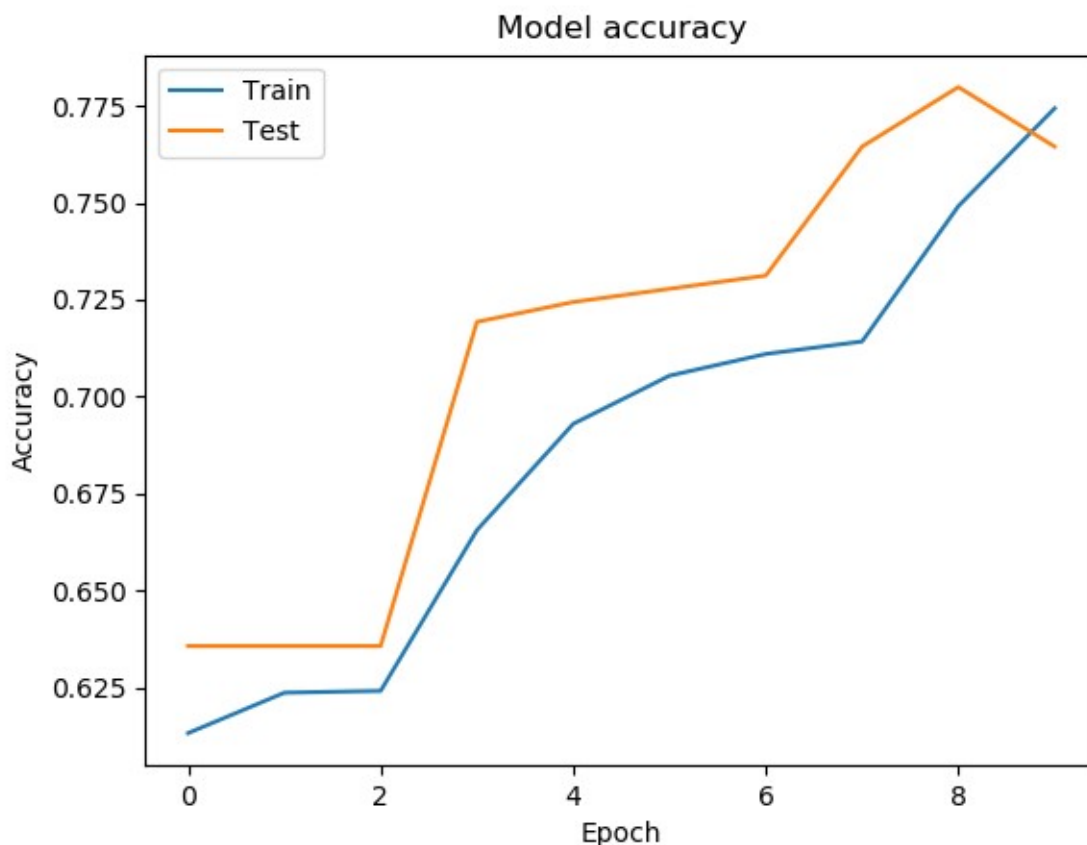
Πολλές φορές, όπως και στην εργασία μας, χρησιμοποιείται και άλλο ένα σύνολο, το σύνολο επαλήθευσης (validation set). Αυτό το σύνολο χρησιμοποιείται ουσιαστικά ως σύνολο δοκιμής ξεχωριστό από τα υπόλοιπα, κατά την διάρκεια της εκπαίδευσης, και βοηθάει στην επιλογή του βέλτιστου μοντέλου προς εκπαίδευση. Λειτουργεί ως μια αμερόληπτη εκτιμήτρια για την βελτιστοποίηση των εμπλεκόμενων υπερπαραμέτρων (hyperparameter). Οι υπερπαραμέτροι είναι οι ιδιότητες του συστήματος (και

συγκεκριμένα στην περίπτωση μας των νευρωνικών δικτύων) οι οποίες μπορούν να τροποποιηθούν, με επίδραση στο αποτέλεσμα. Τέτοιες είναι ο αριθμός των κρυφών επιπέδων, ο αριθμός των νευρώνων σε κάθε επίπεδο κ.ά. . Στα ΣΝΔ υπερπαραμέτροι είναι και ο αριθμός των φίλτρων που χρησιμοποιείται, και το είδος της συγκέντρωσης που ακολουθεί την συνέλιξη.

Δεν υπάρχει κάποια απόλυτη μέθοδος που προσδιορίζει τα ποσοστά των δεδομένων που ανήκουν στα σύνολα που περιγράψαμε. Συνηθισμένοι διαχωρισμοί είναι 80/10/10, 70/10/20, 70/20/10, 60/40 (χωρίς σύνολο επαλήθευσης). Το ποιος διαχωρισμός θα χρησιμοποιηθεί εξαρτάται από τον αριθμό των διαθέσιμων προτύπων και πιθανώς και από τον αριθμό των χαρακτηριστικών που έχει το καθένα. Αν διαθέτουμε μόλις 100 πρότυπα, ο διαχωρισμός σε 60 για εκπαίδευση και 40 για δοκιμή μάλλον δεν θα οδηγήσει σε καλά αποτελέσματα. Τον λόγο θα τον δούμε στην αμέσως επόμενη ενότητα.

2.4.2 Υποπροσαρμογή και υπερπροσαρμογή

Δύο σημαντικά ζητήματα που έχουν επίδραση στην απόδοση ενός συστήματος ΑΣ και γενικότερα ταξινόμησης είναι η υποπροσαρμογή (underfitting) και υπερπροσαρμογή (overfitting).

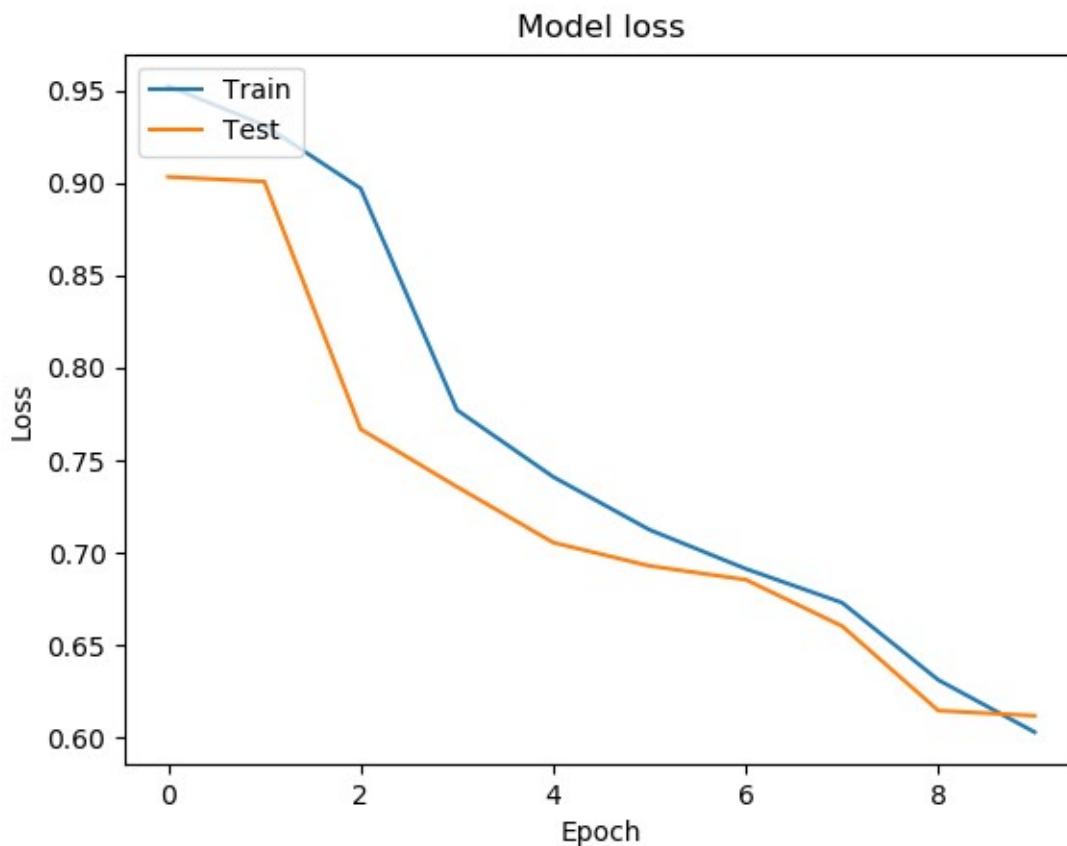


Εικόνα 2.15: Επίδραση της υποπροσαρμογής στην γραφική παράσταση απόδοσης-εποχών

Υποπροσαρμογή έχουμε όταν το μοντέλο δεν έχει εκπαιδευθεί αρκετά ή η αρχιτεκτονική του δεν επαρκεί για να μπορέσει να μάθει καλές παραμέτρους που περιγράφουν επαρκώς το πρόβλημα. Η επίδοση του μοντέλου πιθανώς θα είναι υψηλότερη στο σύνολο δοκιμής (χωρίς αυτό όμως να εγγυάται κάτι για την ικανότητα γενίκευσης του σε

νέα δεδομένα), και θα παρουσιάζει χαμηλές επιδόσεις συγκριτικά στο σύνολο εκπαίδευσης.

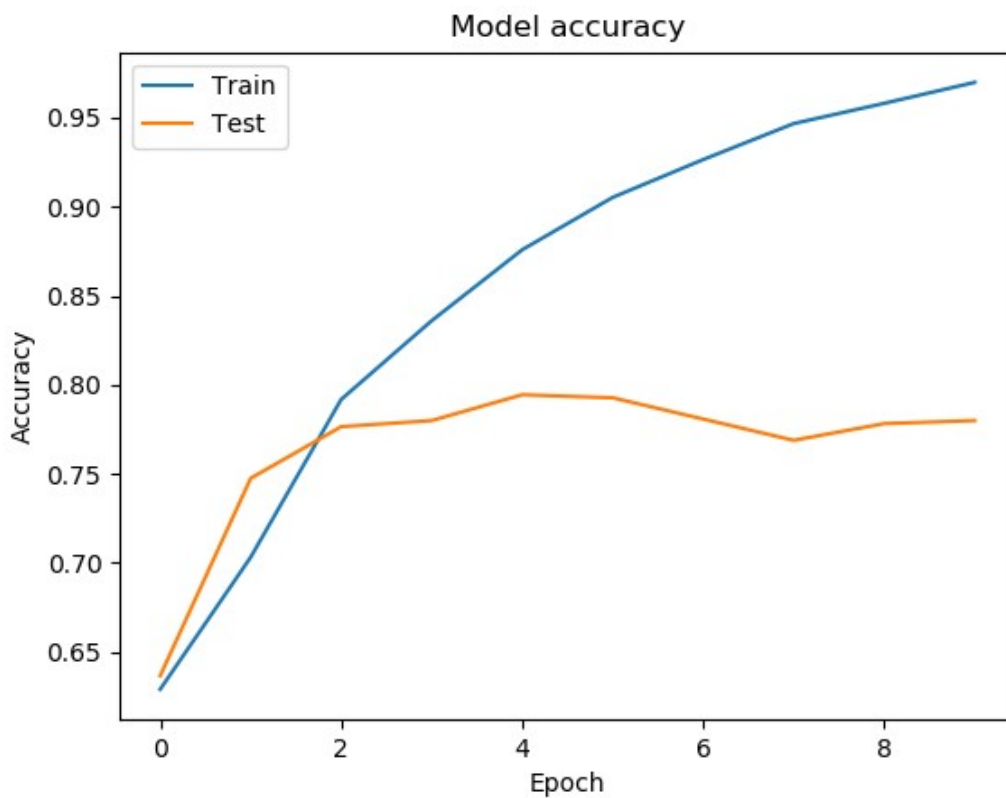
Ενώ φαινομενικά φαίνεται να μην υπάρχει πρόβλημα αφού το σημαντικότερο κριτήριο αξιολόγησης είναι η επίδοση στο σύνολο δοκιμής, είναι σημαντικό να τονίσουμε πως σε μια σωστή εκπαίδευση το σύστημα θα είχε εξαρχής καλύτερη επίδοση στο σύνολο εκπαίδευσης, το οποίο με την σειρά του θα προκαλούσε μια μετατόπιση προς το καλύτερο και στο σύνολο δοκιμής.



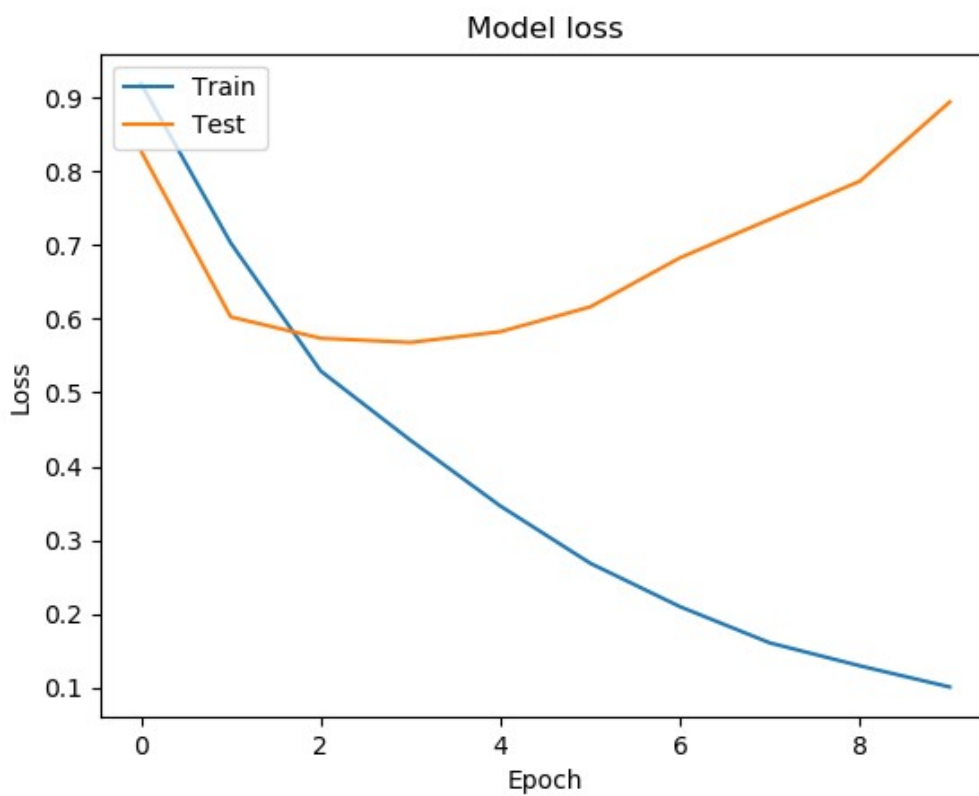
Εικόνα 2.16: Επίδραση της υποπροσαρμογής στην συνάρτηση απώλειας

Αντίθετα υπερπροσαρμογή έχουμε όταν το μοντέλο έχει προσαρμοστεί υπερβολικά καλά στα δεδομένα εκπαίδευσης, με συνέπεια να έχει μάθει μόνο αυτά καλά και να έχει αδυναμία στην ικανότητα γενίκευσης. Έτσι, η απόδοση του είναι υψηλότερη στο σύνολο εκπαίδευσης ενώ αρκετά χαμηλότερη.

Η υπερπροσαρμογή είναι πολύ συχνότερη από την υποπροσαρμογή λόγω της φύσης της μηχανικής μάθησης, και είναι αρκετά δυσκολότερη η αντιμετώπιση της. Μάλιστα, τα ΑΝΔ λόγω της πολυπλοκότητας της αρχιτεκτονικής τους έχουν την τάση να υπερπροσαρμόζονται αρκετά εύκολα. Οι ΑΜΠ που περιγράψαμε προηγουμένως βοηθούν στην αντιμετώπιση του προβλήματος μέσω του μηχανισμού των πυλών.

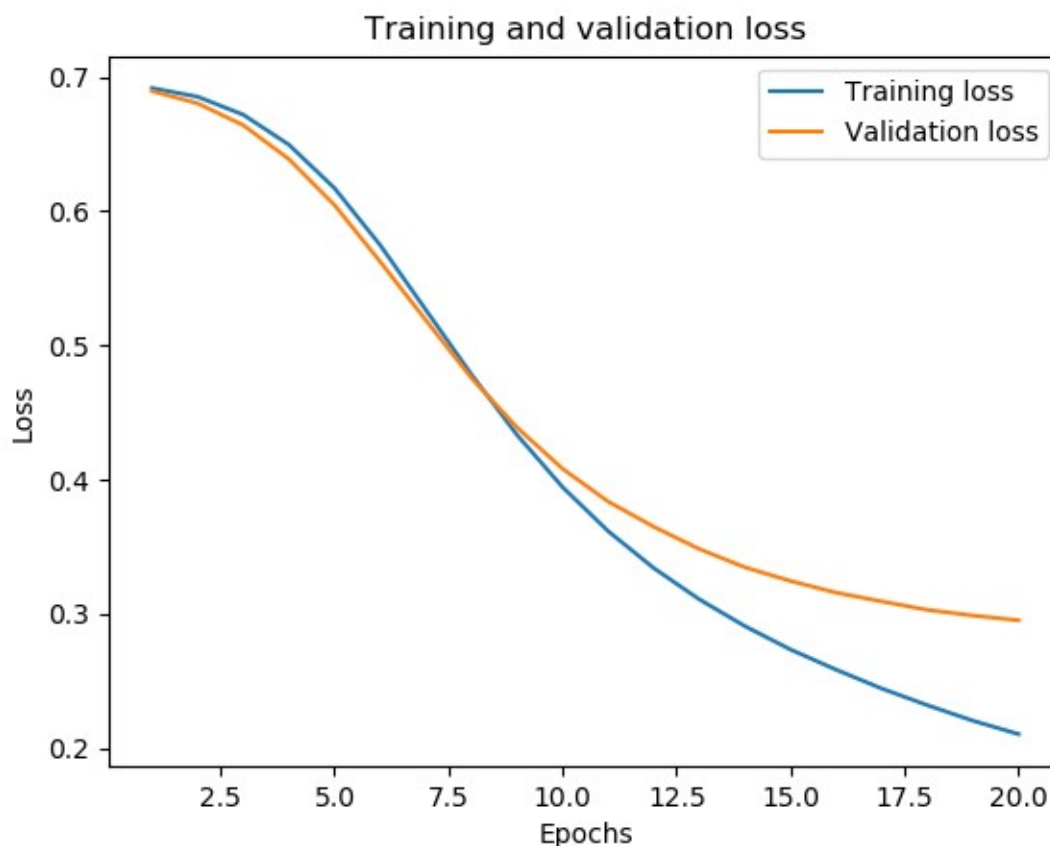


Εικόνα 2.17: Επίδραση της υπερπροσαρμογής στην επίδοση



Εικόνα 2.18: Επίδραση της υπερπροσαρμογής στην συνάρτηση απώλειας

Το ζητούμενο σε μια καλή εκπαίδευση είναι η επίδοση του συστήματος στο σύνολο εκπαίδευσης να είναι παραπλήσια με το σύνολο δοκιμής.



Εικόνα 2.19: Παράδειγμα ικανοποιητικής εκπαίδευσης

Μερικοί απλοί τρόποι για να αντιμετωπίσουμε την υποπροσαρμογή είναι:

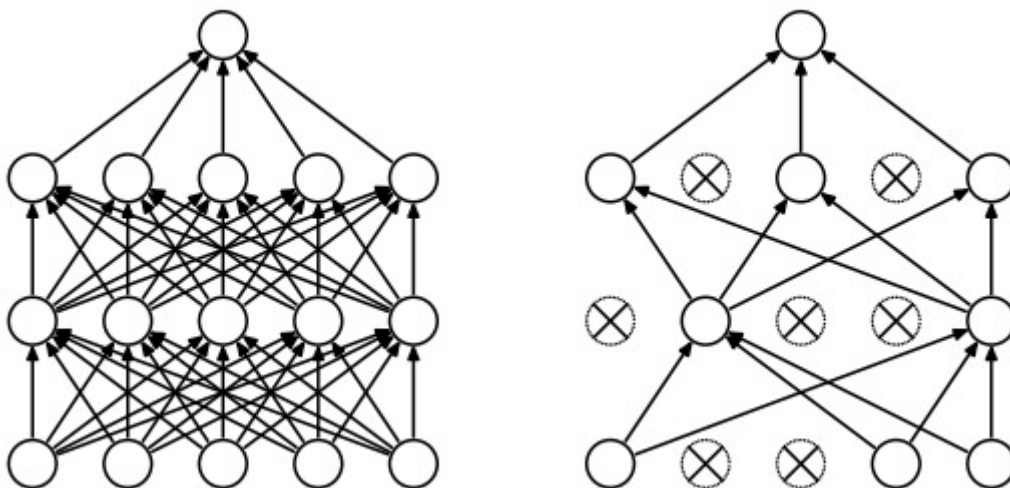
- Να αφήσουμε το δίκτυο να εκπαιδευθεί για παραπάνω αριθμό εποχών. Αυτό θα έχει ουσιαστικό αποτέλεσμα μόνο αν έχει γίνει σωστή επιλογή των παραμέτρων του μοντέλου.
- Να κάνουμε το μοντέλο πιο περίπλοκο, με τρόπους όπως η αύξηση των επιπέδων ή των νευρώνων ανά κρυφό επίπεδο. Συνήθως αυτό αρκεί για να απαλείψει το πρόβλημα.

Για την υπερπροσαρμογή έχουν προταθεί διάφοροι τρόποι αντιμετώπισης. Μερικοί από αυτούς είναι:

- Να μειώσουμε την πολυπλοκότητα του μοντέλου, κάνοντας περικοπή στον αριθμό των κρυφών επιπέδων και των νευρώνων.
- Να σταματήσουμε την εκπαίδευση νωρίς. Στην εικόνα 2.17 φαίνεται πως μετά από περίπου 5 εποχές υπάρχει μια στασιμότητα ή μια πτωτική πορεία στην επίδοση, οπότε δεν χρειάζεται παραπάνω εκπαίδευση το σύστημα. Η πιο συνηθισμένη παράμετρος η οποία καθορίζει το πότε θα σταματήσει η εκπαίδευση είναι η επίδοση στο σύνολο δοκιμής (ή η πορεία της συνάρτησης κόστους στο ίδιο σύνολο).
- Να επιλέξουμε καλύτερα χαρακτηριστικά που θα χρησιμοποιηθούν ως είσοδο στο δίκτυο. Θα επανέλθουμε σε αυτό το ζήτημα στο επόμενο κεφάλαιο.

- Να συλλέξουμε περισσότερα δεδομένα. Αυτή είναι μια αρκετά αποτελεσματική μέθοδος καθώς βοηθάει σημαντικά στην ικανότητα γενίκευσης αλλά δυστυχώς σπάνια είναι εφικτό να χρησιμοποιηθεί καθώς η συλλογή σωστών δεδομένων και η δημιουργία συνόλων κατάλληλων για μηχανική μάθηση είναι μια χρονοβόρα και περίπλοκη διαδικασία.
- Να χρησιμοποιήσουμε κάποια πιο προχωρημένη μέθοδο όπως την διασταυρωμένη επικύρωση (cross validation) ή την μέθοδο εγκατάλειψης (dropout).
- Να χρησιμοποιήσουμε ένα σύνολο επαλήθευσης όπως περιγράψαμε στην ενότητα 2.4.1.

Καθώς η μέθοδος διασταυρωμένης επικύρωσης έχει απαγορευτικό υπολογιστικό κόστος για μεγάλους όγκους δεδομένων όπως στην εργασία μας, θα χρησιμοποιήσουμε την μέθοδο εγκατάλειψης η οποία δίνει πολύ καλά αποτελέσματα [35]. Κατά την μέθοδο εγκατάλειψης, σε κάθε κύκλο εκπαίδευσης εξαιρείται με τυχαίο τρόπο από την μάθηση ένα ποσοστό νευρώνων, σταθερό ή μεταβλητό (συνήθως σταθερό). Αυτό εμποδίζει την υπερπροσαρμογή καθώς ουσιαστικά εκπαιδεύονται πολλά μικρότερα δίκτυα αντί για ένα μεγάλο.



Εικόνα 2.20: Μέθοδος εγκατάλειψης. Οι νευρώνες με X έχουν εξαιρεθεί από την εκπαίδευση. Πηγή: [35]

Προφανώς, οι νευρώνες που έχουν εξαιρεθεί από την εκπαίδευση δεν έχουν και τις αντίστοιχες συνάψεις.

3. ΕΙΣΑΓΩΓΗ ΣΤΟ TENSORFLOW

3.1 Γενικά

Το TensorFlow ([43], διαθέσιμο στο <https://tensorflow.org/>) είναι ολοκληρωμένη πλατφόρμα μηχανικής μάθησης ανοιχτού κώδικα που αναπτύχθηκε από την ομάδα Google Brain. Σκοπός του εγχειρήματος ήταν η ανάπτυξη μιας βιβλιοθήκης που έχει καλή κλιμάκωση σε κατανεμημένα συστήματα, απλότητα στη χρήση και υποστήριξη των πιο πρόσφατων μεθόδων.

Ένα σημαντικό χαρακτηριστικό του TensorFlow είναι πως επιτρέπει την χρήση της βιβλιοθήκης CUDA της NVIDIA, σε περίπτωση που ο χρήστης διαθέτει συμβατή κάρτα γραφικών NVIDIA στο σύστημα του. Αυτό οδηγεί σε σημαντική επιτάχυνση της διαδικασίας εκπαίδευσης, καθώς οι πράξεις που γίνονται τυπικά κατά την εκπαίδευση μπορούν να υλοποιηθούν με πολύ αποτελεσματικό τρόπο σε κάρτες γραφικών.

Σε χαμηλότερο επίπεδο το TensorFlow βασίζεται στους υπολογιστικούς γράφους (computational graphs). Ο χρήστης της βιβλιοθήκης στις αρχικές εκδόσεις της καθόριζε την μορφή του γράφου και τις πράξεις οι οποίες θα υλοποιούνταν σε αυτόν. Ο γράφος αυτός μπορεί να αντιστοιχεί στην αρχιτεκτονική ενός ΤΝΔ όπως αυτά που περιγράψαμε στο προηγούμενο κεφάλαιο.

Με την πάροδο των ετών έγιναν σημαντικές βελτιώσεις στην πλατφόρμα και με την εισαγωγή της έκδοσης 2.0 ενσωματώθηκε η βιβλιοθήκη υψηλού επιπέδου Keras [52] στο TensorFlow. Η βιβλιοθήκη Keras παρέχει σημαντικές ρουτίνες διαθέσιμες στον χρήστη που βοηθάνε στην προεπεξεργασία, τον προσδιορισμό του μοντέλου μηχανικής μάθησης, την εκπαίδευση κ.ά. Στις επόμενες ενότητες θα παρουσιάσουμε τις βασικότερες λειτουργίες του TensorFlow-Keras που χρησιμοποιήσαμε, μαζί με ενδεικτικά τμήματα κώδικα από τα πηγαία αρχεία της εργασίας που υλοποιούν αυτές τις λειτουργίες. Μερικές από τις λειτουργίες που θα αναφέρουμε εδώ εξηγούνται καλύτερα από την θεωρητική και μαθηματική οπτική τους στο κεφάλαιο 4, στο οποίο παραπέμπεται ο αναγνώστης για μια βαθύτερη ανάλυση.

3.2 Δυνατότητες προεπεξεργασίας

Το πακέτο preprocessing περιέχει πολλές δυνατότητες προεπεξεργασίας. Μία από αυτές είναι και η μέθοδος Tokenizer για την μετατροπή κειμένων σε ακολουθίες από αριθμούς, αφού πρώτα γίνει διαχωρισμός λέξεων:

```
# Train a keras tokenizer and transform reviews
print('Training tokenizer...')
tokenizer=keras.preprocessing.text.Tokenizer(num_words=top_words
)
tokenizer.fit_on_texts(x_train)
print('Word indexes: ', tokenizer.word_index)
```

Εδώ αρχικοποιούμε τον Tokenizer με την παράμετρο num_words, η οποία δηλώνει τον μέγιστο αριθμό λέξεων που θέλουμε να διατηρήσουμε. Μπορούμε να δούμε το λεξιλόγιο που σχηματίζεται το οποίο περιλαμβάνει αρχικώς όλες τις λέξεις:

```
'adoloph': 216857, 'wisnston': 216858, 'semblances': 216859,  
's3100': 216860, 'wakeling': 216861, 'kramerjean': 216862,  
'grandfatherarthur': 216863, 'butlerhelen': 216864, ...
```

Η παράμετρος `num_words` θα χρησιμοποιηθεί στη συνέχεια στις μεθόδους `fit_on_texts` και `texts_to_sequences` οι οποίες μετατρέπουν τα κείμενα σε ακολουθίες ακεραίων. Η μέθοδος `fit_on_texts` ενημερώνει το λεξιλόγιο του `Tokenizer` σύμφωνα με την παράμετρο `num_words`.

```
# More preprocessing. Each review is transformed to a series of  
integers where each integer represents the  
# associated word's frequency in the entire corpus.  
print('Transforming texts to vectors...')  
print('Sample text: ',x_train[0])  
x_train = tokenizer.texts_to_sequences(x_train)  
x_test = tokenizer.texts_to_sequences(x_test)  
print('Sample vectorized text', x_train[0])
```

το οποίο θα μας δώσει αποτέλεσμα:

```
Sample text: Different Take on Stoker's Vamps! Director Ken  
Russell, who a few years before directed Altered States, brings  
in a young Hugh Grant and a few English actors to put together a  
wild and funny story...
```

```
Sample vectorized text: [286, 206, 19, 6084, 201, 2921, 1652,  
33, 3, 181, 115, 147, 469, 4756, 1459, 687, 7, 3, 171, 3829,  
968, 2, 3, 181, 544, 194, ...]
```

Βλέπουμε πως οι λέξεις έχουν αντιστοιχηθεί πλέον σε αριθμούς. Σημειώνουμε εδώ πως τα `x_train`, `x_test` είναι δομές δεδομένων τύπου λίστας της Python που περιέχουν τα σύνολα δοκιμής και εκπαίδευσης αντίστοιχα. Ο διαχωρισμός έχει γίνει με την εντολή `train_test_split` της βιβλιοθήκης `scikit-learn` [48].

Μέσω του πακέτου `pickle` της Python μπορούμε να αποθηκεύσουμε το μοντέλο που κατασκεύασε η μέθοδος `Tokenizer`:

```
# Save tokenizer to disk  
with open('twitter_tokenizer.pickle', 'wb') as handle:  
    pickle.dump(tokenizer, handle, protocol=pickle.HIGHEST_PROTOCOL)
```

Για την σωστή χρήση της συνάρτησης απώλειας διεντροπίας, χρειάζεται να γίνει μετατροπή των κλάσεων σε κωδικοποίηση με μηδενικά και άσσους (`one-hot encoding`). Έστω πως λαμβάνουμε την περίπτωση των 5 κλάσεων και πως η επιθυμητή κατηγορία είναι η 4η. Με την χρήση της λειτουργίας `to_categorical` του `Keras` μπορούμε να κάνουμε την μετατροπή ως εξής:

```
# Convert to one hot encoding for use with categorical
crossentropy.
print('Target before one-hot encoding: ', y_train[0])
y_train = keras.utils.to_categorical(y_train, num_classes=5)
y_test = keras.utils.to_categorical(y_test, num_classes=5)
print('Target after encoding: ', y_train[0])
```

που θα μας δώσει αποτέλεσμα:

```
Target before one-hot encoding: 3
Target after encoding: [0. 0. 0. 1. 0.]
```

Αυτό γίνεται για όλα τα κείμενα. Σημειώνουμε εδώ πως η αρίθμηση των δεικτών στην Python ξεκινάει από το 0 οπότε οι 5 κατηγορίες κωδικοποιούνται ως 0, 1, 2, 3, 4.

Μπορούμε επίσης να ορίσουμε ένα μέγιστο μήκος για όλα τα κείμενα μας με την χρήση της εντολής `pad_sequences` ως εξής:

```
# Pad each sequence to a fixed size
print('Padding sequences to a fixed size of ', maxlen)
x_train = keras.preprocessing.sequence.pad_sequences(x_train,
maxlen=maxlen)
x_test = keras.preprocessing.sequence.pad_sequences(x_test,
maxlen=maxlen)
print('Train dataset shape is ', x_train.shape)
```

Η παράμετρος `maxlen` καθορίζει το μέγιστο μήκος των ακολουθιών. Για παράδειγμα μπορούμε να ορίσουμε στην αρχή του πηγαίου αρχείου `maxlen = 200`. Μετά την εκτέλεση της εντολής όλα τα κείμενα θα έχουν μήκος 200, είτε με συμπλήρωση μηδενικών αν είναι μικρότερου μήκους, είτε με περικοπή των πλεοναζόντων λέξεων. Πράγματι:

```
Padding sequences to a fixed size of 200
Train dataset shape is (240000, 200)
```

το οποίο δηλώνει πως έχουμε 240.000 κείμενα στο σύνολο εκπαίδευσης με μήκος 200 το καθένα.

3.3 Κατασκευή νευρωνικών δικτύων και εκπαίδευση

Μέσω του Keras Sequential API (Application programming interface) μπορούμε να ορίσουμε σε υψηλό επίπεδο την αρχιτεκτονική του δικτύου και τις εμπλεκόμενες παραμέτρους, καθώς και πρόσθετες δυνατότητες. Αρχίζουμε προσδιορίζοντας το πότε είναι καλό να σταματήσει η εκπαίδευση, αν αρχίσει να παρουσιάζεται υπερπροσαρμογή:

```

# Build Tensorflow-Keras GRU Model. Use a bidirectional GRU for
higher accuracy
# Dropout is used for regularization
# Stop training early if that is desired.
    callbacks = [
        keras.callbacks.EarlyStopping(
            # Stop training when `val_loss` is no longer
improving
            monitor='val_loss',
            min_delta=1e-2,
            # Also wait for at least 3 epochs before deciding
whether to stop training
            patience=3,
            verbose=1)
    ]

```

Μέσω της λειτουργίας `callbacks` του Keras μπορούμε να παρακολουθήσουμε την εκπαίδευση και να την σταματήσουμε κατάλληλα. Η παράμετρος `monitor` καθορίζει ποια τιμή θα χρησιμοποιηθεί για την λήψη της σχετικής απόφασης. Εδώ έχουμε δηλώσει πως θα παρακολουθείται η τιμή της συνάρτησης απώλειας στο σύνολο επαλήθευσης. Η παράμετρος `min_delta` αναφέρεται στην μικρότερη αποδεκτή μεταβολή της τιμής που παρακολουθείται. Η παράμετρος `patience` δηλώνει τον αριθμό των εποχών που πρέπει να περάσουν ώστε να ληφθεί η απόφαση για την διακοπή της εκπαίδευσης.

Ας δούμε τώρα ένα παράδειγμα για την κατασκευή ενός συστήματος που βασίζεται σε ΑΜΠ για ΑΣ:

```

model = tf.keras.Sequential([
    tf.keras.layers.Embedding(top_words, embedding_size,
input_length=maxlen, mask_zero=False),
    tf.keras.layers.Dropout(0.45),
    tf.keras.layers.Bidirectional(
        tf.keras.layers.GRU(units=64, activation='tanh',
recurrent_activation='sigmoid', unroll=False,
        use_bias=True, recurrent_dropout=0, reset_after=True,
return_sequences=True)),
    tf.keras.layers.GlobalMaxPooling1D(),
    tf.keras.layers.Dropout(0.45),
    tf.keras.layers.Dense(5, activation='softmax')
])
# Save model architecture to file

```

```
keras.utils.plot_model(model, to_file='images/modelGRU.png',  
show_shapes=True, show_layer_names=True)
```

Αρχικά δηλώνουμε πως χρησιμοποιούμε το Sequential API. Στη συνέχεια:

- Κατασκευάζουμε ένα επίπεδο εκπαίδευσης διανυσμάτων λέξεων (εξηγούνται αναλυτικά στο επόμενο κεφάλαιο). Οι παράμετροι που λαμβάνει αυτό το επίπεδο είναι ο αριθμός των λέξεων που θα μετατραπούν σε διανύσματα (top_words), η διάσταση των διανυσμάτων (embedding_size), το μήκος των κειμένων εισόδου (input_length) και η παράμετρος mask_zero που αναφέρεται στην εσωτερική κωδικοποίηση των διανυσμάτων.
- Εφαρμόζουμε ένα επίπεδο εγκατάλειψης νευρώνων με πιθανότητα 0.45.
- Κατασκευάζουμε ένα αμφίδρομο ANΔ το οποίο λειτουργεί όπως περιγράψαμε στο προηγούμενο κεφάλαιο. Το είδος του ANΔ είναι η ΑΜΠ (tf.keras.layers.GRU). Οι παράμετροι εντός της παρένθεσης είναι ο αριθμός των κελιών ΑΜΠ, οι συναρτήσεις ενεργοποίησης που χρησιμοποιεί, η παράμετρος unroll που δηλώνει εάν θα ξεδιπλωθεί ο υπολογιστικός γράφος της ΑΜΠ, η χρήση πόλωσης, το ποσοστό εγκατάλειψης στο αναδρομικό σκέλος, το πότε θα εφαρμοστεί η πύλη επαναφοράς και το αν θα επιστρέφεται όλη η ακολουθία σε κάθε χρονικό βήμα. Για την λειτουργία ταχύτερης εκπαίδευσης απαιτούνται οι παράμετροι να έχουν οριστεί όπως στο τμήμα κώδικα μας.
- Χρησιμοποιούμε ένα επίπεδο συγκέντρωσης (GlobalMaxPooling1D) το οποίο εφαρμόζει ολική συγκέντρωση μεγίστων για κάθε χρονικό βήμα.
- Χρησιμοποιούμε πάλι ένα επίπεδο εγκατάλειψης νευρώνων με πιθανότητα 0.45.
- Κατασκευάζουμε το επίπεδο εξόδου με 5 νευρώνες και την συνάρτηση ενεργοποίησης softmax.

Με την εντολή plot_model μπορούμε να αποθηκεύσουμε την αρχιτεκτονική του συστήματος σε ένα αρχείο εικόνας.

Το αντίστοιχο τμήμα κώδικα για τα ΣΝΔ είναι:

```
model = tf.keras.Sequential([  
    tf.keras.layers.Embedding(top_words, embedding_size,  
input_length=maxlen, mask_zero=False),  
    tf.keras.layers.Dropout(0.5),  
    tf.keras.layers.SeparableConv1D(filters=filters,  
kernel_size=kernel_size, bias_initializer='random_uniform',  
depthwise_initializer='random_uniform', padding='same',  
activation='relu', strides=stride),  
    tf.keras.layers.SeparableConv1D(filters=filters,  
kernel_size=kernel_size, bias_initializer='random_uniform',  
depthwise_initializer='random_uniform', padding='same',  
activation='relu', strides=stride),  
    tf.keras.layers.GlobalMaxPooling1D(),
```

```
tf.keras.layers.Dropout(0.5),
tf.keras.layers.Dense(5, activation='softmax')
])
```

Η διαφορά με το τμήμα κώδικα για τις ΑΜΠ είναι ότι αντί για την εντολή `tf.keras.layers.GRU` χρησιμοποιούμε την `tf.keras.layers.SeparableConv1D`. Η παράμετρος `filters` αναφέρεται στον αριθμό των φίλτρων που χρησιμοποιούνται, η παράμετρος `kernel_size` στο μέγεθος του πυρήνα, η παράμετρος `bias_initializer` στην αρχικοποίηση της πόλωσης (εδώ σύμφωνα με την κανονική κατανομή), η παράμετρος `depthwise_initializer` στην αρχικοποίηση του πυρήνα της συνέλιξης βάθους, η παράμετρος `padding` στην μέθοδο γεμίσματος (εδώ χρησιμοποιείται γέμισμα με μηδενικά), η παράμετρος `activation` στην μορφή της συνάρτησης ενεργοποίησης και η παράμετρος `stride` στο μέγεθος του βήματος ολίσθησης.

Αφού έχουμε ορίσει την αρχιτεκτονική του δικτύου μας μπορούμε να ξεκινήσουμε την εκπαίδευση:

```
# Multiclass problem-> should use categorical_crossentropy, and
adam or rmsprop preferably.
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])

# Can stop training early by removing the # in the callbacks
comment.
history = model.fit(x_train,
                   y_train,
                   epochs=epochs,
                   batch_size=batch_size,
                   callbacks=callbacks,
                   validation_split=0.2,
                   verbose=1)

results = model.evaluate(x_test, y_test, verbose=1)
```

Η εντολή `model.compile` σχηματίζει τον υπολογιστικό γράφο του TensorFlow σύμφωνα με όσα έχουν δηλωθεί προηγουμένως. Η παράμετρος `loss` αναφέρεται στην συνάρτηση απώλειας, η παράμετρος `optimizer` στο είδος του αλγορίθμου εκπαίδευσης και η συνάρτηση `metrics` στην μετρική αποτίμησης του ταξινομητή.

Η εντολή `model.fit` εφαρμόζει την αρχιτεκτονική του συστήματος στο σύνολο εκπαίδευσης. Οι λίστες `x_train`, `y_train` περιλαμβάνουν τα κείμενα εκπαίδευσης με τα αντίστοιχα συναισθήματα, η παράμετρος `epochs` καθορίζει τον αριθμό των εποχών εκπαίδευσης, η παράμετρος `batch_size` το μέγεθος της δέσμης εκπαίδευσης, η παράμετρος `callbacks` το πότε θα σταματήσουμε την εκπαίδευση εάν το επιθυμούμε (εδώ θα γίνει σύμφωνα με όσα περιγράψαμε προηγουμένως), και τέλος η παράμετρος

`validation_split` ορίζει το ποσοστό του συνόλου δεδομένων που θα χρησιμοποιηθεί ως σύνολο επαλήθευσης.

Η εντολή `model.evaluate` αξιολογεί το σύστημα στο σύνολο δοκιμής `x_test`, `y_test`. Μπορούμε επίσης να αποθηκεύσουμε το εκπαιδευμένο δίκτυο για μελλοντική χρήση:

```
model.save('twitter_CNN.h5')
```

Με την εντολή `model.load` σε μεταγενέστερη στιγμή φορτώνουμε το μοντέλο που αποθηκεύσαμε.

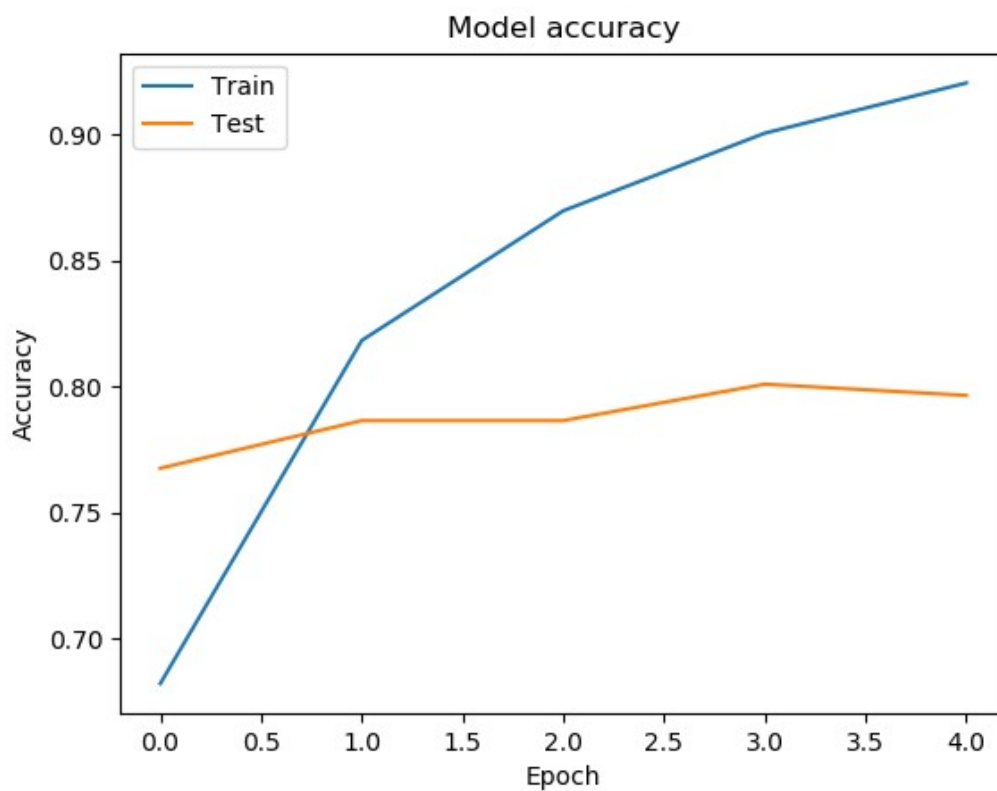
3.4 Γραφήματα εκπαίδευσης

Αν και δεν αποτελεί μέρος του TensorFlow, για λόγους πληρότητας περιγράφουμε εδώ το πώς μπορούμε να οπτικοποιήσουμε την πορεία της εκπαίδευσης του δικτύου με την βοήθεια της βιβλιοθήκης Matplotlib [49].

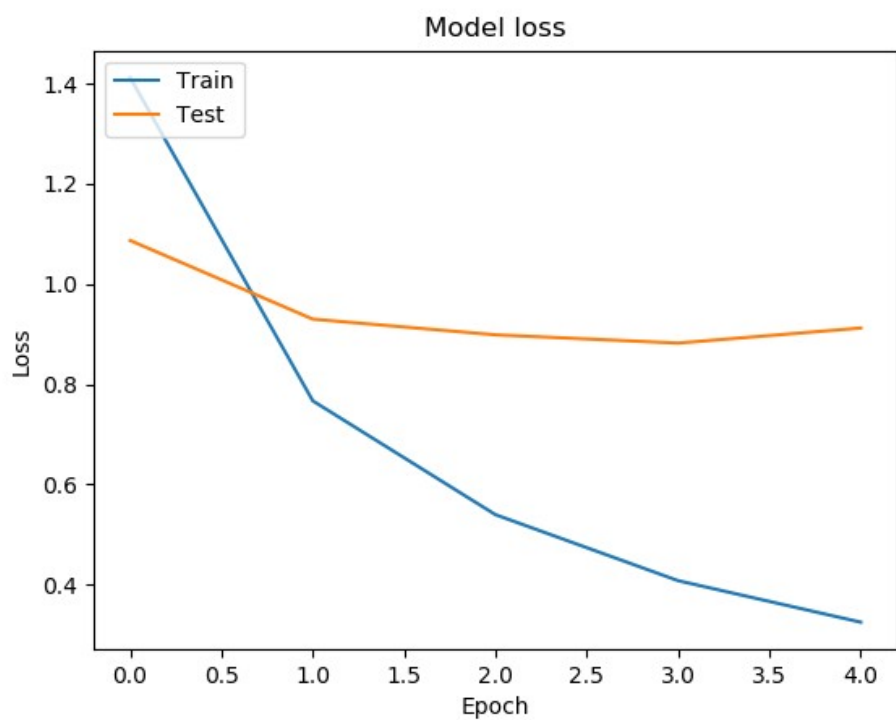
```
# Plot training & validation accuracy values
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()

# Plot training & validation loss values
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```

Μέσα από το αντικείμενο `history` του Keras έχουμε πρόσβαση στα στοιχεία που χρειαζόμαστε. Τα γραφήματα που κατασκευάζονται είναι όπως αυτά που δείξαμε στο κεφάλαιο 2. Περιλαμβάνουν την πορεία της συνάρτησης απώλειας και της ακρίβειας του ταξινομητή, ως προς τις εποχές εκπαίδευσης.



Εικόνα 3.1: Παράδειγμα γραφήματος ακρίβειας-εποχών



Εικόνα 3.2: Παράδειγμα γραφήματος συνάρτησης απώλειας-εποχών

4. ΘΕΜΑΤΑ ΕΠΕΞΕΡΓΑΣΙΑΣ ΦΥΣΙΚΗΣ ΓΛΩΣΣΑΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗΣ

4.1 Διανύσματα λέξεων

Ένα σύστημα ΑΣ και γενικότερα μηχανικής μάθησης δεν μπορεί να χρησιμοποιήσει τα δεδομένα στην αρχική τους μορφή τις περισσότερες φορές. Αυτό είναι σίγουρο για τα κείμενα, καθώς τα δίκτυα που περιγράψαμε δεν μπορούν να χρησιμοποιήσουν λέξεις ως είσοδο. Χρειάζεται να αναπαρασταθούν σε κάποια αριθμητική μορφή. Δηλαδή, αν έχουμε μια συλλογή C από N κείμενα (κριτικές, tweets):

$$C = (d_1, d_2, \dots, d_N)$$

τότε μπορούμε να γράψουμε πως κάθε κείμενο από αυτά αποτελείται από τις λέξεις-διανύσματα:

$$d = [w_1, w_2, \dots, w_k]$$

Το ζήτημα τώρα είναι με ποια μορφή θα κωδικοποιηθεί κάθε λέξη στο σύνολο των κειμένων. Μία συνηθισμένη πρακτική είναι η χρήση κωδικοποίησης με μηδενικά και άσσους όπου [38] σε ένα λεξιλόγιο μεγέθους V , κάθε λέξη αναπαρίσταται με $V-1$ μηδενικά και έναν άσσο. Ας υποθέσουμε το εξής απλό παράδειγμα: Έχουμε την πρόταση "This is good". Με αυτή την αναπαράσταση, θα μπορούσαμε να γράψουμε:

$$this = [1, 0, 0]$$

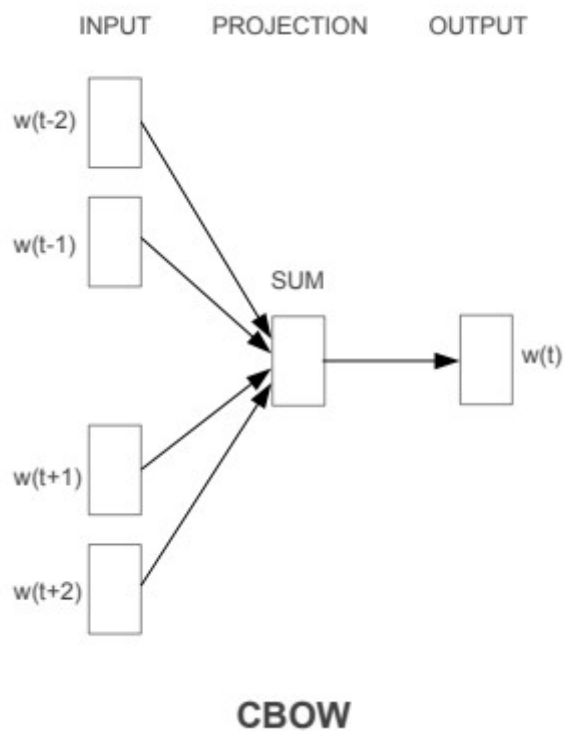
$$is = [0, 1, 0]$$

$$good = [0, 0, 1]$$

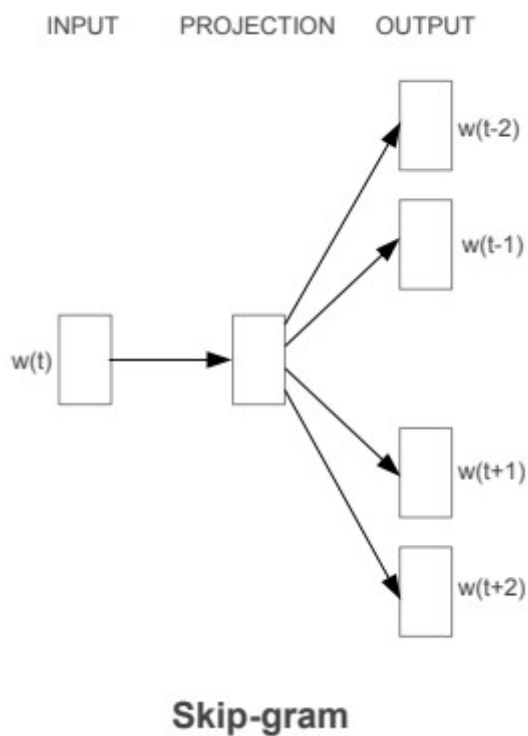
Αυτή η προσέγγιση αν και απλή έχει ένα σημαντικό μειονέκτημα: Το μέγεθος του λεξιλογίου στις περισσότερες εφαρμογές και σύνολα δεδομένων της ΑΣ είναι γιγαντιαίο συγκριτικά. Είναι πολύ συχνό να συναντάμε λεξιλόγια της τάξης των 50.000+ λέξεων όπως και στην εργασία μας. Τα διανύσματα που θα προέκυπταν θα ήταν πολύ αραιά, και θα είχαμε σημαντικές απαιτήσεις τόσο σε υπολογιστικό κόστος όσο και σε μνήμη. Αρκεί να σκεφτούμε πως μόλις ένα κείμενο 80 λέξεων θα απεικονιζόταν σε διάσταση 80 x 50.000.

Μία πολύ καλή λύση είναι το μοντέλο των διανυσμάτων λέξεων (word embeddings) το οποίο εκμεταλλεύεται τα συμπραζόμενα [39]. Σε αυτό κάθε λέξη απεικονίζεται ως διάνυσμα μιας σταθερής διάστασης N του οποίου οι συνιστώσες είναι πραγματικοί αριθμοί, και με τέτοιο τρόπο ώστε λέξεις με παραπλήσια σημασία να καταλήγουν σε παραπλήσια σημεία στο διανυσματικό χώρο [36]. Το 2013 προτάθηκε ο αλγόριθμος word2vec [36] ο οποίος δίνει έναν μηχανισμό αποτελεσματικής εκπαίδευσης ενός δικτύου το οποίο έχει ως στόχο την κατασκευή τέτοιων διανυσμάτων.

Η μία προσέγγιση είναι να προβλέπουμε μια λέξη σύμφωνα με τα συμπραζόμενα της εντός ενός παραθύρου παρατήρησης με συγκεκριμένο μήκος. Αυτό είναι το μοντέλο συνεχούς σάκου λέξεων (continuous bag-of-words – CBOW). Στην ακόλουθη εικόνα φαίνεται μια αρχιτεκτονική που υλοποιεί ένα τέτοιο μοντέλο. Από την άλλη, μπορούμε να προβλέψουμε τα συμπραζόμενα μιας συγκεκριμένης λέξης, πάλι εντός ενός παραθύρου παρατήρησης. Αυτό είναι το μοντέλο πρόβλεψης συμπραζομένων (skip-gram).



Εικόνα 4.1: Μοντέλο συνεχούς σάκου λέξεων (CBOW) [36]



Εικόνα 4.2: Μοντέλο skip-gram (πρόβλεψη συμφραζομένων) [36]

Το μοντέλο πρόβλεψης συμφραζομένων είναι αρκετά πρακτικό λόγω της αρχιτεκτονικής του. Σύμφωνα με το [37], δοθείσης μιας ακολουθίας λέξεων w_1, w_2, \dots, w_T στόχος είναι η μεγιστοποίηση της λογαριθμικής πιθανότητας:

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log(p(w_{t+j}|w_t))$$

όπου c το μέγεθος του παραθύρου παρατήρησης. Οι πιθανότητες μπορούν να οριστούν ως:

$$p(w_O|w_I) = \frac{e^{(u_{w_O}^T u_{w_I})}}{\sum_{w=1}^W e^{(u_w^T u_{w_I})}}$$

όπου W το μέγεθος του λεξιλογίου, u και u' οι διανυσματικές αναπαραστάσεις της λέξης w στην είσοδο και την έξοδο του δικτύου αντίστοιχα. Επειδή ο προηγούμενος τύπος δεν έχει καλή κλιμάκωση για μεγάλα λεξιλόγια, μπορεί να χρησιμοποιηθεί η τεχνική της αρνητικής δειγματοληψίας (negative sampling) η οποία έχει την εξής συνάρτηση-στόχο:

$$\log \sigma(u_{w_O}^T u_{w_I}) + \sum_{i=1}^k E_{w_i \sim P_n(w)} [\log \sigma(-u_{w_i}^T u_{w_I})]$$

Εδώ διακρίνεται η λέξη-στόχος w_O από δείγματα που προκύπτουν από την κατανομή θορύβου $P_n(w)$.

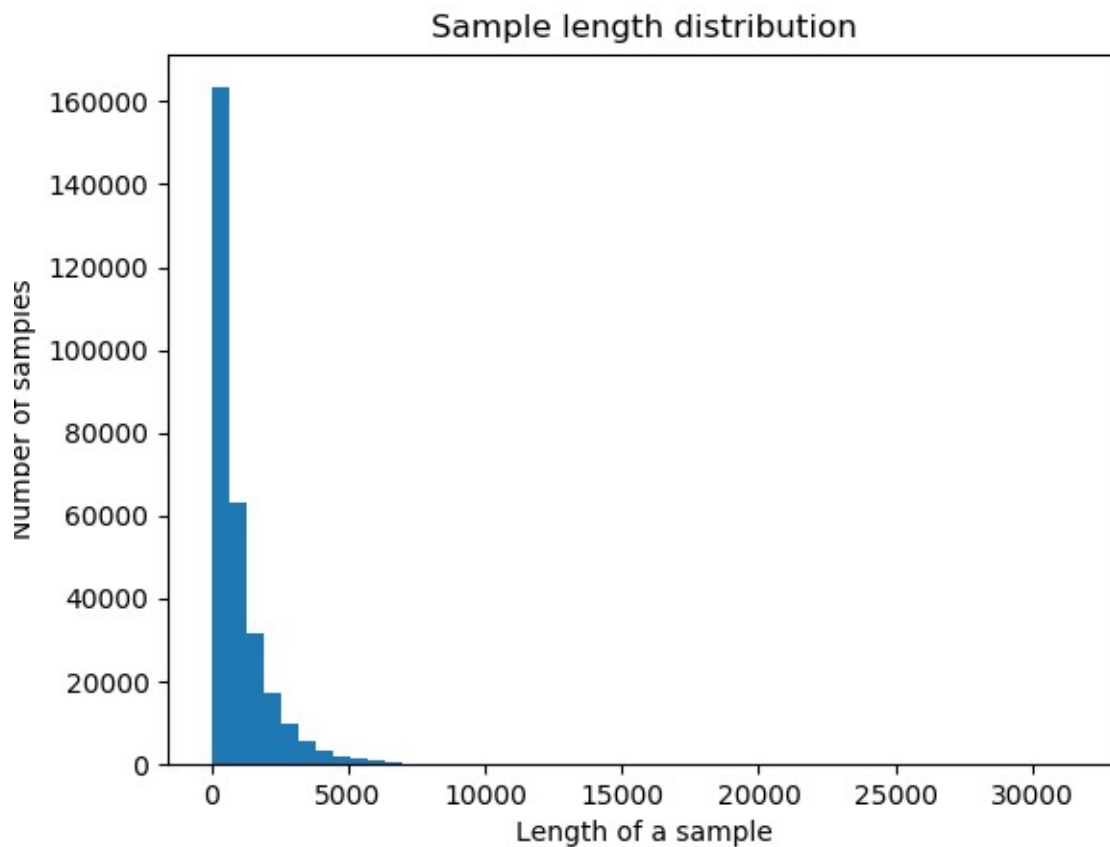
Η διάσταση των διανυσμάτων που προκύπτουν κυμαίνεται γενικά. Κλασικές τιμές είναι μεταξύ 50-300, αλλά η καλύτερη εκτίμηση της παραμέτρου μόνο πειραματικά μπορεί να γίνει γιατί εξαρτάται άμεσα από το σύνολο δεδομένων που διαθέτουμε.

4.2 Σύνολα δεδομένων που χρησιμοποιήθηκαν

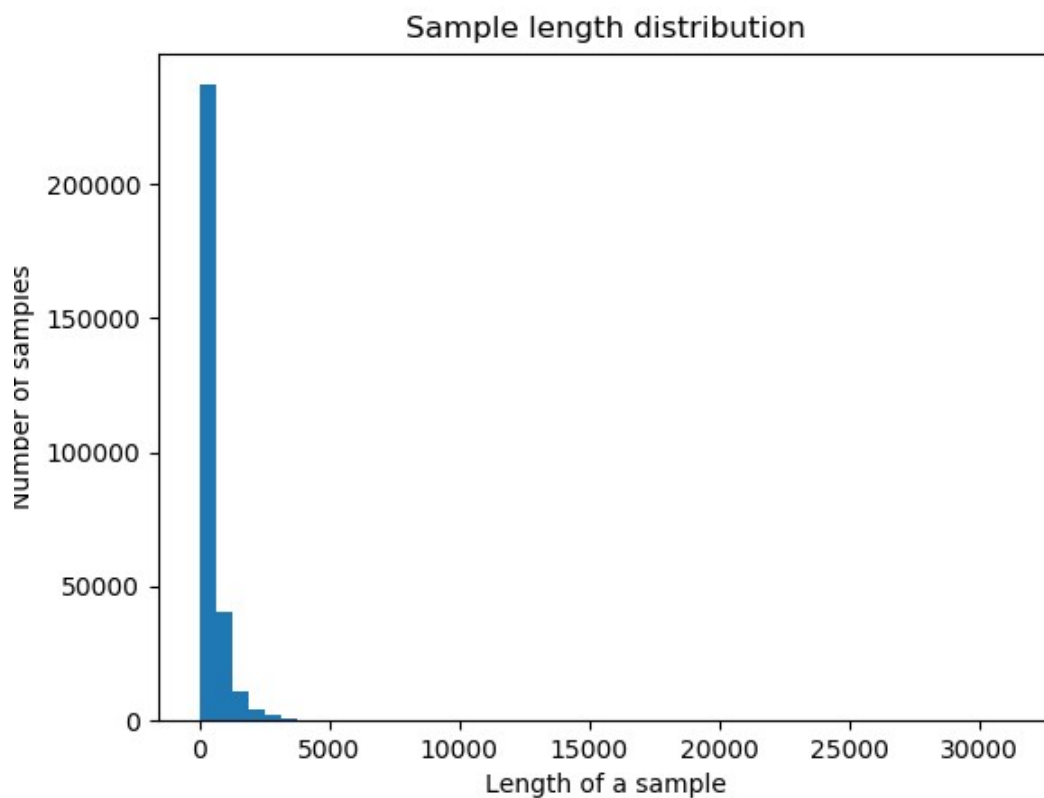
4.2.1 Το σύνολο δεδομένων Amazon review data

Το σύνολο δεδομένων Amazon review data [40] είναι ένα τεράστιο σύνολο δεδομένων της ιστοσελίδας διαδικτυακών αγορών Amazon με κριτικές από πολλές κατηγορίες προϊόντων οι οποίες συλλέχθηκαν κατά το διάστημα 1996-2014. Στην ιστοσελίδα <http://jmcauley.ucsd.edu/data/amazon/> βρίσκεται ελεύθερα διαθέσιμο ένα υποσύνολο του αρχικού συνόλου που προσφέρεται μεταξύ άλλων για εφαρμογές ΑΣ. Αυτό το υποσύνολο έχουμε χρησιμοποιήσει και στην εργασία.

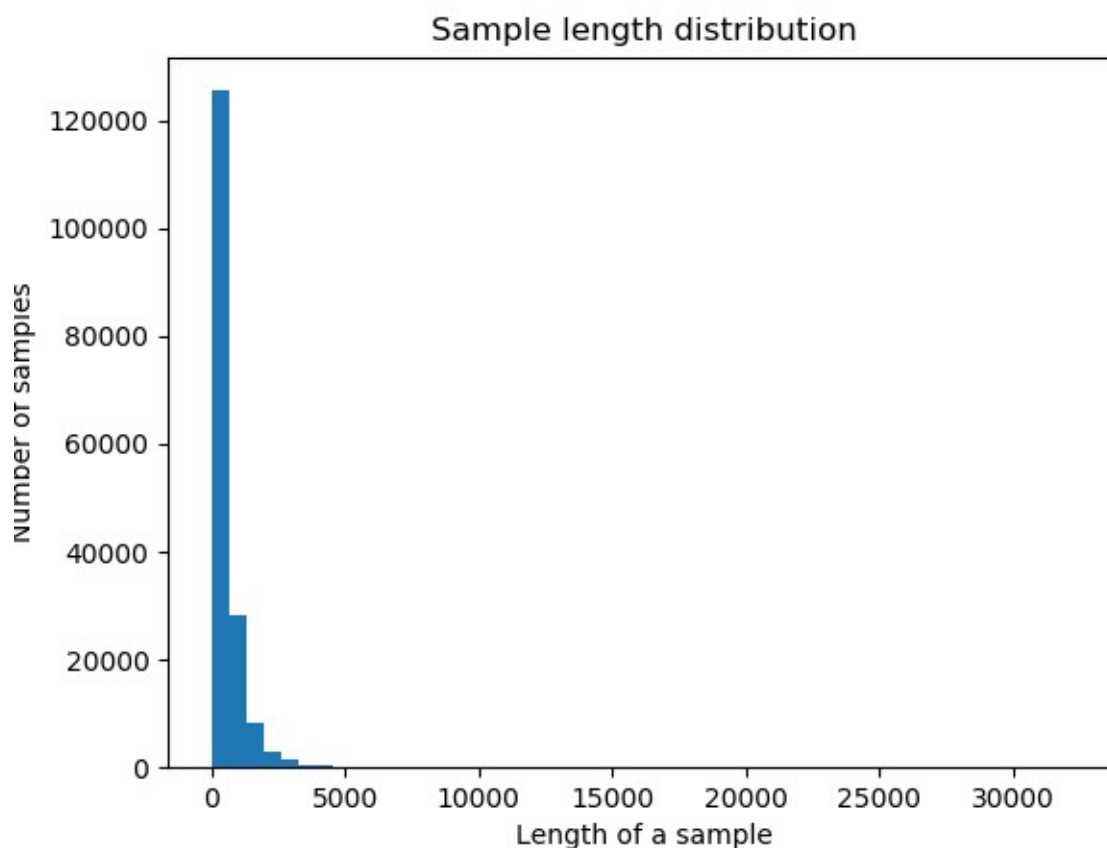
Από τις διάφορες υποκατηγορίες επιλέξαμε τρεις οι οποίες έχουν χρησιμοποιηθεί και στην βιβλιογραφία για συστήματα ΑΣ [41]. Αυτές είναι οι Movies and TV (κριτικές ταινιών και σειρών), Toys and Games (κριτικές για παιχνίδια) και Sports and Outdoors (κριτικές για αντικείμενα σχετικά με αθλήματα). Από τις 1.697.533 κριτικές της πρώτης κατηγορίας διατηρήσαμε 300.000 κριτικές ταινιών, και χρησιμοποιήσαμε όλες τις διαθέσιμες κριτικές από τις άλλες 2 (167.597 και 296.337 αντίστοιχα). Στις εικόνες που ακολουθούν παρέχονται μερικά διαγράμματα για την κατανομή του μήκους των κριτικών για κάθε κατηγορία.



Εικόνα 4.3: Κατανομή μήκους κριτικών ταινιών



Εικόνα 4.4: Κατανομή μήκους κριτικών για αντικείμενα σχετικά με αθλήματα



Εικόνα 4.5: Κατανομή μήκους για αντικείμενα σχετικά με παιχνίδια

Κάθε προϊόν βαθμολογείται με 1 έως 5 αστέρια σύμφωνα με όσα είπαμε στην ενότητα 1.2.1. Στον παρακάτω πίνακα φαίνεται η κατανομή του αριθμού των κριτικών ανά αστέρια.

Πίνακας 4.1: Κατανομή αριθμού αστεριών

	1 Αστέρι	2 Αστέρια	3 Αστέρια	4 Αστέρια	5 Αστέρια
Movies and TV	13.671	13.632	28.845	64.782	179.070
Sports and Outdoors	9.045	10.204	24.071	64.809	188.208
Toys and Games	4.707	6.298	16.357	37.445	102.790

Το μήκος του λεξιλογίου για την πρώτη κατηγορία είναι 256.916 καθώς περιλαμβάνει πολλές λέξεις οι οποίες συναντώνται αποκλειστικά στην εκάστοτε ταινία όπως τα

ονόματα χαρακτήρων. Για την κατηγορία παιχνιδιών έχουμε 81.768 και για την κατηγορία με αθλητικά είδη έχουμε 124.241.

Βλέπουμε ότι υπάρχει έντονη ανισορροπία κλάσεων (class imbalance), δηλαδή μερικές κατηγορίες έχουν πολλά περισσότερα πρότυπα από άλλες, ιδίως οι θετικές αξιολογήσεις. Αυτό είναι δυστυχώς ένα αρκετά συχνό πρόβλημα στην ΑΣ καθώς οι χρήστες που αποκόμισαν πολύ θετικές εντυπώσεις για ένα προϊόν, είναι αρκετά πιθανό να το επαινέσουν στο διαδίκτυο. Η ανισορροπία κλάσεων δημιουργεί προβλήματα στην απόδοση του ταξινομητή αλλά τις περισσότερες φορές είναι αναπόφευκτη. Υπάρχουν τεχνικές όπως η υποδειγματοληψία των συχνότερων κλάσεων αλλά αυτές με την σειρά τους μπορούν να έχουν επίπτωση στην ικανότητα γενίκευσης του συστήματος. Και αυτό διότι μειώνουμε τον αριθμό των διαθέσιμων προτύπων για εκπαίδευση.

Μια λογική που εφαρμόσαμε για τον περιορισμό του προβλήματος είναι η δημιουργία μιας δεύτερης εκδοχής του συνόλου δεδομένων, με 3 κατηγορίες αυτή τη φορά αντί για 5. Οι κριτικές των 1 και 2 αστεριών συγχωνεύτηκαν σε μία κατηγορία με “αρνητικό” συναίσθημα, οι κριτικές των 3 αστεριών αποτελούν το “ουδέτερο” ενώ οι υπόλοιπες σχηματίζουν την κατηγορία του “θετικού” συναισθήματος. Στο κεφάλαιο 4 το οποίο περιέχει την παρουσίαση των αποτελεσμάτων θα δούμε τις επιδράσεις που είχε αυτή η προσέγγιση στην απόδοση.

Εκτός από το κείμενο της κριτικής και τον αριθμό των αστεριών το σύνολο δεδομένων περιέχει και άλλα πεδία. Το πιο ενδιαφέρον από αυτά είναι το πεδίο “summary” που περιέχει μια σύνοψη της κριτικής από τον χρήστη. Συνενώνοντας την σύνοψη με το κείμενο κάθε κριτικής, εκμεταλλευόμαστε πολύ χρήσιμη πληροφορία την οποία μπορούν να αξιοποιήσουν τα δίκτυα που περιγράψαμε στο προηγούμενο κεφάλαιο.

4.2.2 Το σύνολο δεδομένων Sentiment140

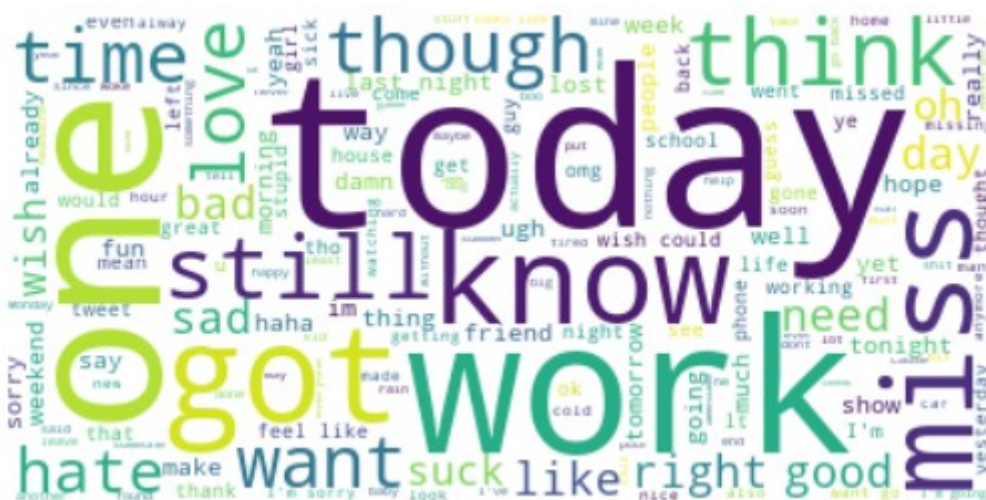
Το σύνολο δεδομένων Sentiment140 ή Stanford Twitter Sentiment Corpus ([42], <http://help.sentiment140.com/for-students>) είναι ένα σύνολο από 1.600.000 τιβίσματα από την ιστοσελίδα Twitter τα οποία σχετίζονται το καθένα με είτε θετικό είτε αρνητικό συναίσθημα. Για την εργασία μας χρησιμοποιήσαμε τα μισά από αυτά, δηλαδή 800.000. Η κατανομή είναι σχεδόν απόλυτα ισορροπημένη, με κάθε κατηγορία να περιέχει περίπου 400.000 μηνύματα χρηστών.

Τα μηνύματα στο Twitter είναι αρκετά πιο ανεπίσημα και με διαφορετική χρήση της γλώσσας σε σύγκριση με το σύνολο δεδομένων Amazon. Για παράδειγμα, ένα τυχαίο πρότυπο εδώ είναι το :

“@Torn13 holy shiznee! thats quite a chunk of time. the sad thing bout me bein happy is i used to do it all the time last yr. “

ενώ μια τυχαία κριτική ταινίας είναι:

““This is a charming version of the classic Dicken's tale. Henry Winkler makes a good showing as the \"Scrooge\" character. Even though you know what will happen this version has enough of a change to make it better than average. If you love A Christmas Carol in any version, then you will love this.”



Εικόνα 4.8: Συχνότερες λέξεις σε αρνητικά μηνύματα χρηστών

4.3 Προεπεξεργασία

Στην βιβλιογραφία έχουν προταθεί πολλοί διαφορετικοί τρόποι για την προεπεξεργασία των κειμένων σε συστήματα μηχανικής μάθησης. Η αποτελεσματικότητα αυτών εξαρτάται άμεσα από το σύνολο δεδομένων που διαθέτουμε κάθε φορά καθώς και από τον αλγόριθμο μάθησης που χρησιμοποιείται. Παραθέτουμε εδώ την προεπεξεργασία που κάναμε στα κείμενα μας:

- Εκτελέσαμε διαχωρισμό λέξεων (tokenization). Κατά τον διαχωρισμό λέξεων, το σύστημα κατασκευάζει το λεξιλόγιο από την συλλογή κειμένων. Οι όροι δεν είναι μόνο λέξεις. Για παράδειγμα, ένας κλασικός διαχωρισμός λέξεων στην πρόταση “This is it (or is it?)” θα κατασκευάζει την λίστα όρων [this, is, it, (, or, ?,)].
- Από τα αρχικά λεξιλόγια, διατηρήσαμε έναν συγκεκριμένο αριθμό N από λέξεις, σύμφωνα με την συχνότητα τους. Αυτή είναι μια παράμετρος η οποία μπορεί να έχει σημαντική επίδραση στην ικανότητα γενίκευσης του ταξινομητή, καθώς από αυτό το μειωμένο λεξιλόγιο θα υπολογιστούν τα διανύσματα λέξεων. Όταν το σύστημα θα κληθεί να αναλύσει το συναίσθημα ενός άγνωστου κειμένου, όποια λέξη δεν αναγνωρίζει θα την αγνοήσει τελείως. Από την άλλη, αν το N είναι υπερβολικά μεγάλο τότε υπάρχει κίνδυνος υπερπροσαρμογής επειδή τα διανύσματα που θα σχηματιστούν θα εξαρτώνται υπερβολικά πολύ από τα συμφραζόμενα του συνόλου εκπαίδευσης. Για τις δοκιμές μας, το N κυμαινόταν κυρίως μεταξύ 10.000 και 50.000.
- Για την κατασκευή των διανυσμάτων λέξεων, όλοι οι όροι στα κείμενα αντικαταστάθηκαν με αριθμούς, χρησιμοποιώντας την συχνότητα τους στο λεξιλόγιο.
- Αναθέσαμε σε όλες τις κριτικές και τα μηνύματα ένα σταθερό μήκος . Αυτό είναι σημαντικό καθώς τα νευρωνικά δίκτυα απαιτούν όλες οι είσοδοι να έχουν το ίδιο μήκος. Κείμενα με περισσότερους όρους περικόπηκαν, ενώ κείμενα μικρότερα από το όριο συμπληρώθηκαν με μηδενικά.

- Για το σύνολο δεδομένων Sentiment140 αφαιρέσαμε τους υπερσυνδέσμους που είναι πολύ συνηθισμένοι στο Twitter χωρίς να προσδίδουν κάποια συναισθηματική πληροφορία.
- Η αναλογία διαχωρισμού των δεδομένων είναι 60/20/20 για το σύνολο δεδομένων Amazon, και 70/20/10 για το σύνολο δεδομένων Sentiment140. Τα ποσοστά αναφέρονται στο σύνολο εκπαίδευσης/δοκιμής/επαλήθευσης αντίστοιχα.

Εκτός από τους υπερσυνδέσμους, στα μηνύματα του Twitter είναι συχνοί πολλοί όροι όπως “@”, λέξεις με λίγα γράμματα οι οποίες είναι συνήθως συντμήσεις και διάφορα σύμβολα όπως “:-O” “:-/”. Ενώ θεωρητικά μπορούν να αφαιρεθούν, το πρόβλημα είναι πως πολλοί από αυτούς συμβαίνει να επηρεάζουν το συναίσθημα του μηνύματος. Είναι προτιμότερο να αφήσουμε την διαλογή να γίνει από το στάδιο διαχωρισμού λέξεων και την κατασκευή των διανυσμάτων λέξεων, ακόμα και αν αυτό σημαίνει ελαφρώς αυξημένη υπολογιστική πολυπλοκότητα.

Το ίδιο συμβαίνει και με τις αχρείαστες λέξεις (stop words). Αυτές είναι λέξεις που συναντώνται πολύ συχνά (am, it, are κ.ά.) και είναι συνηθισμένο να αφαιρούνται στην κατηγοριοποίηση κειμένων. Ο τρόπος κατασκευής τέτοιων λιστών όμως μπορεί να ενέχει κινδύνους [44]. Με τον τρόπο που δουλεύουν τα ΣΝΔ, οι ΑΜΠ και τα διανύσματα λέξεων, υπάρχει σημαντικός κίνδυνος απώλειας συναισθηματικής πληροφορίας λόγω έλλειψης συμφραζομένων. Γι’αυτό δεν τις αφαιρέσαμε.

4.4 Αρχιτεκτονική συστήματος ΑΣ

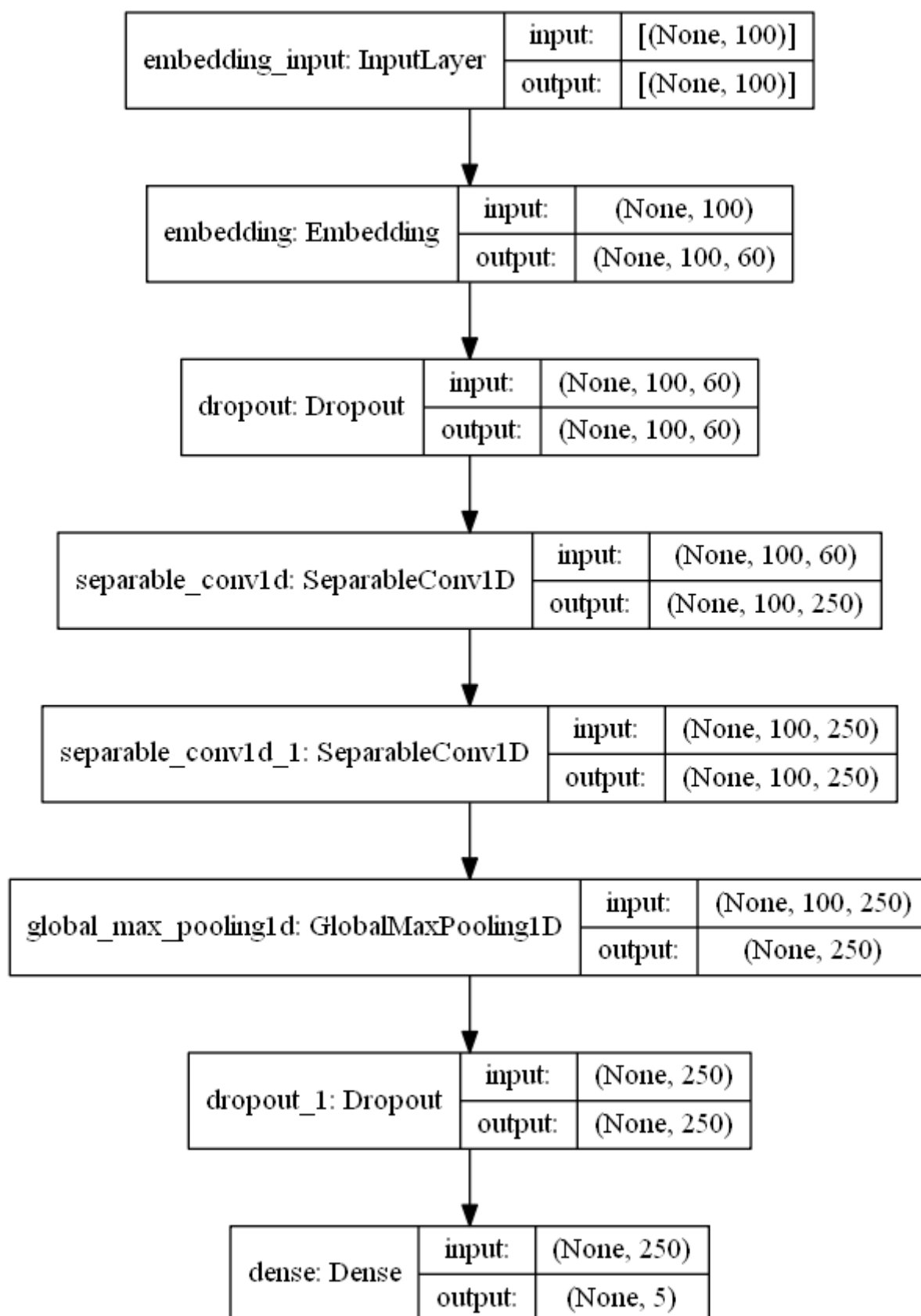
Επανερχόμαστε στο σημείο αυτό στη λειτουργία των ΣΝΔ. Για την κατασκευή του χάρτη χαρακτηριστικών μπορούμε να θέσουμε σε στήλες τα διανύσματα λέξεων και έπειτα να εφαρμοστεί κανονικά η πράξη της συνέλιξης. Για παράδειγμα έστω η πρόταση “I liked it” και έστω πως τα διανύσματα που προέκυψαν είναι [0.2,0.5,0.8], [0.3,0.7,0.9] και [0.1,0.4,0.6]. Τότε:

Επίπεδο εισόδου

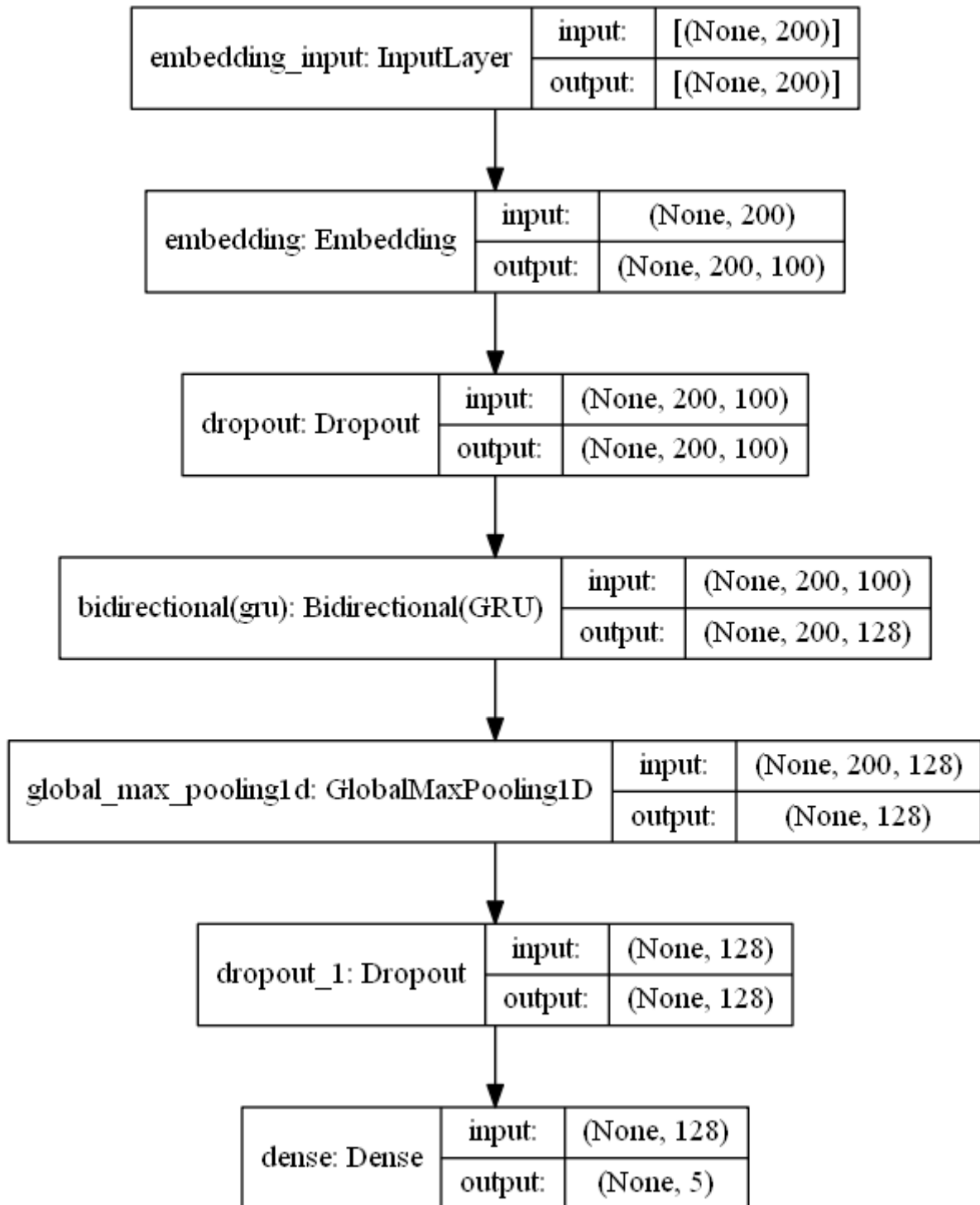
I	0.2	0.5	0.8
liked	0.3	0.7	0.9
it	0.1	0.4	0.6

Εικόνα 4.9: Χρήση διανυσμάτων λέξεων από ΣΝΔ

Στις εικόνες που ακολουθούν παραθέτουμε τα διαγράμματα αρχιτεκτονικής του συστήματος όπως δόθηκαν από το TensorFlow για την περίπτωση 5 κλάσεων.



Εικόνα 4.10: Αρχιτεκτονική ΣΝΔ για Ανάλυση Συναισθήματος



Εικόνα 4.11: Αρχιτεκτονική ΑΜΠ για Ανάλυση Συναισθήματος

Αυτά τα διαγράμματα αφορούν το σύνολο δεδομένων Amazon και τις 5 κατηγορίες. Οι παράμετροι στις παρενθέσεις μπορούν να αλλάξουν καθώς η εικόνα δείχνει την τιμή τους για μια συγκεκριμένη δοκιμή. Η συνάρτηση ενεργοποίησης στο επίπεδο εξόδου είναι η softmax. Για τις 3 κατηγορίες έχουμε προφανώς 3 νευρώνες εξόδου αντί για 5. Για το σύνολο δεδομένων Sentiment140 έχουμε 1 νευρώνα εξόδου και τη σιγμοειδή συνάρτηση ενεργοποίησης.

Για τα ΣΝΔ χρησιμοποιήσαμε μια υπολογιστικά αποτελεσματική παραλλαγή της συνέλιξης, την διαχωρίσιμη συνέλιξη (separable convolution) [45][46]. Σύμφωνα με το [45], οι κλασικές συνέλιξεις έχουν υπολογιστικό κόστος :

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F$$

όπου M ο αριθμός των καναλιών εισόδου, N ο αριθμός των καναλιών εξόδου, K ο δείκτης του πυρήνα και F ο δείκτης του χάρτη χαρακτηριστικών.

Στην διαχωρίσιμη συνέλιξη αρχικά εφαρμόζεται ένα φίλτρο για κάθε κανάλι εισόδου (συνέλιξη βάθους- depthwise convolution) και έπειτα εφαρμόζεται 1×1 συνέλιξη (συνέλιξη σημείου- pointwise convolution). Αυτή δημιουργεί έναν γραμμικό συνδυασμό σύμφωνα με την έξοδο του προηγούμενου επιπέδου. Στην γενική περίπτωση η συνέλιξη βάθους μπορεί να γραφτεί ως:

$$G_{k,l,m} = \sum_{i,j} K'_{i,j,m} \cdot F_{k+i-1,l_j-1,m}$$

όπου K' ο πυρήνας της συνέλιξης βάθους με μέγεθος $D_K * D_K * M$. Το m -οστό φίλτρο του πυρήνα εφαρμόζεται στο m -οστό κανάλι εισόδου για να σχηματιστεί ο χάρτης χαρακτηριστικών. Η συνέλιξη βάθους έχει κόστος:

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F$$

Συνολικά η διαχωρίσιμη συνέλιξη έχει κόστος:

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F$$

Το κέρδος σε σύγκριση με την κανονική συνέλιξη μπορεί να εκφραστεί ως:

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2}$$

Συνάρτηση ενεργοποίησης για τα ΣΝΔ είναι ο γραμμικός ανορθωτής και για τις ΑΜΠ είναι η υπερβολική εφαπτομένη και η σιγμοειδής. Η πιθανότητα εγκατάλειψης τέθηκε στο 0.45 για τα διανύσματα λέξεων και τις ΑΜΠ, και στο 0.5 για την περίπτωση των ΣΝΔ. Για την εκπαίδευση χρησιμοποιείται ο αλγόριθμος Adam [47] και η συνάρτηση κόστους είναι η διεντροπία που περιγράψαμε στο κεφάλαιο 2. Ο αριθμός των εποχών εκπαίδευσης είναι 10 αλλά στις περισσότερες υποπεριπτώσεις σταματάμε την εκπαίδευση όταν διαπιστώσουμε πως η ακρίβεια στο σύνολο επαλήθευσης μειώνεται αντί να αυξάνεται για κάποιο χρονικό διάστημα, ώστε να αποφύγουμε περιττή υπερπροσαρμογή.

```
53184/107261 [=====>.....] - ETA: 40s - loss: 0.4423 - accuracy: 0.8490
53312/107261 [=====>.....] - ETA: 40s - loss: 0.4419 - accuracy: 0.8491
53440/107261 [=====>.....] - ETA: 40s - loss: 0.4418 - accuracy: 0.8491
53568/107261 [=====>.....] - ETA: 39s - loss: 0.4417 - accuracy: 0.8492
53696/107261 [=====>.....] - ETA: 39s - loss: 0.4412 - accuracy: 0.8494
53824/107261 [=====>.....] - ETA: 39s - loss: 0.4411 - accuracy: 0.8494
53952/107261 [=====>.....] - ETA: 39s - loss: 0.4410 - accuracy: 0.8495
54080/107261 [=====>.....] - ETA: 39s - loss: 0.4407 - accuracy: 0.8495
54208/107261 [=====>.....] - ETA: 39s - loss: 0.4405 - accuracy: 0.8494
54336/107261 [=====>.....] - ETA: 39s - loss: 0.4401 - accuracy: 0.8496
54464/107261 [=====>.....] - ETA: 39s - loss: 0.4399 - accuracy: 0.8496
54592/107261 [=====>.....] - ETA: 39s - loss: 0.4395 - accuracy: 0.8498
```

Εικόνα 4.12: Στιγμιότυπο εκπαίδευσης όπως παρουσιάζεται από το TensorFlow

Κατά την εκπαίδευση το TensorFlow δίνει πληροφορίες για το πόσα κείμενα έχουν επεξεργαστεί, την τρέχουσα τιμή της συνάρτησης κόστους και την τρέχουσα τιμή της

μετρικής απόδοσης. Χρησιμοποιούμε την μετρική της ακρίβειας η οποία δείχνει το ποσοστό των σωστών προβλέψεων που έκανε το σύστημα ΑΣ.

Συνοψίζοντας, κατά την εκπαίδευση το σύστημα:

- Πραγματοποιεί διαχωρισμό λέξεων και κρατάει τις N συχνότερες από αυτές
- Αντικαθιστά κάθε λέξη στο κείμενο με την συχνότητα εμφάνισης της
- Φροντίζει ώστε να έχουν όλες οι κριτικές/μηνύματα το ίδιο μήκος
- Υπολογίζει το διάνυσμα κάθε λέξης σύμφωνα με τα συμφραζόμενα της
- Το νευρωνικό δίκτυο (ΣΝΔ ή ΑΜΠ) επεξεργάζεται τα κείμενα που εισάγονται
- Το αποτέλεσμα των υπολογισμών του προηγούμενου επιπέδου περνάει στο επίπεδο συγκέντρωσης όπου διατηρείται η μέγιστη τιμή από κάθε χρονικό βήμα (ή κάθε χάρτη χαρακτηριστικών για τα ΣΝΔ)
- Το επίπεδο συγκέντρωσης συνδέεται με τους νευρώνες εξόδου οι οποίοι ενεργοποιούνται σύμφωνα με την συνάρτηση ενεργοποίησης
- Υπολογίζεται η απόκλιση από την επιθυμητή έξοδο σύμφωνα με την συνάρτηση κόστους διεντροπίας και αναπροσαρμόζονται τα βάρη κατάλληλα αφού παρουσιαστούν στο δίκτυο αρκετά κείμενα ώστε να συμπληρωθεί μία δέσμη (ιδανικά 64 στην περίπτωση μας)
- Η διαδικασία επαναλαμβάνεται μέχρι να τελειώσουν οι εποχές ή να διαπιστωθεί πως δεν χρειάζεται παραπάνω εκπαίδευση.

Κατά την ανάλυση συναισθήματος σε άγνωστα κείμενα το σύστημα:

- Φροντίζει ώστε να έχουν όλα το ίδιο μήκος
- Αντικαθιστά κάθε λέξη με το διάνυσμα της όπως υπολογίστηκε στην εκπαίδευση. Αν κάποια λέξη δεν περιλαμβάνεται στο λεξιλόγιο, αγνοείται ή της δίνεται αυθαίρετη τιμή.
- Υπολογίζει την έξοδο του δικτύου και εμφανίζει τις πιθανότητες για κάθε συναίσθημα. Ανάλογα με την περίπτωση, έχουμε είτε 3 συναίσθημα (θετικό, ουδέτερο, αρνητικό) σύμφωνα με το σύστημα ΑΣ του συνόλου δεδομένων Amazon είτε 2 (θετικό, αρνητικό) σύμφωνα με το σύστημα ΑΣ του συνόλου δεδομένων Sentiment140.

Παρουσιάζουμε τώρα στην πράξη τη λειτουργία του συστήματος. Έστω 2 υποτιθέμενα μηνύματα Twitter (άγνωστα για το σύστημα): “this is a perfect day for going out”, “missed the bus...again”. Είναι εμφανές πως το πρώτο μήνυμα έχει θετικό συναίσθημα ενώ το δεύτερο αρνητικό (προφανώς θεωρείται πως είναι αρνητικό το να έχει χάσει ο χρήστης το λεωφορείο). Το αποτέλεσμα της ΑΣ είναι:

```
this is a perfect day for going out Negative: 7.002% Positive: 92.998%
missed the bus...again Negative: 93.920% Positive: 6.080%
```

Εικόνα 4.13: Αποτέλεσμα ΑΣ για 2 μηνύματα Twitter

το οποίο είναι σύμφωνο με την υπόθεση που κάναμε. Είναι ενδιαφέρον να δούμε και την επίδραση που έχουν τα συμφραζόμενα σε μια πρόταση όσον αφορά την ανάλυση συναισθήματος. Αφαιρώντας την λέξη “missed” λαμβάνουμε:



the bus...again Negative: 49.651% Positive: 50.349%

Εικόνα 4.14: Επίδραση συμφραζομένων στην ΑΣ

Παρατηρούμε πως τα ποσοστά είναι σχεδόν ίσα το οποίο δηλώνει πως το συναίσθημα του μηνύματος κυμαίνεται προς το “ουδέτερο”, αν και δεν υπάρχει τυπικά ως ξεχωριστή κατηγορία. Αυτό είναι λογικό καθώς το μήνυμα από μόνο του δεν φαίνεται να εμπεριέχει κάποιο συναίσθημα.

Βέβαια, το σύστημα ίσως να μην δίνει καλά αποτελέσματα σε περιπτώσεις έντονης χρήσης σαρκασμού. Η χρήση ειρωνείας προσθέτει πολύ έντονο θόρυβο στα συστήματα ΑΣ και απαιτεί αρκετά εξειδικευμένες τεχνικές για την μερική αντιμετώπιση της, και αποτελεί ενεργό ερευνητικό θέμα [4]. Υπάρχουν πολλά ζητήματα που καθιστούν την ανάπτυξη ενός γενικευμένου συστήματος ΑΣ αρκετά δύσκολη.

Παραθέτουμε εδώ επίσης ένα παράδειγμα με 2 υποθετικές κριτικές ταινιών. Για λόγους χώρου δεν γίνεται να δείξουμε την εικόνα με το αποτέλεσμα παρά μόνο γραπτώς.

'Overall, I really enjoyed this movie and I can certainly see why it scored so high.'

'The direction is alright but nothing too impressive. Three stars are probably enough in general.'

Το σύστημα έδωσε Negative: 0.405% Neutral: 1.097% Positive: 98.498% για την πρώτη και Negative: 31.840% Neutral: 56.480% Positive: 11.680% για την δεύτερη, που είναι λογικά αποτελέσματα.

5. ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΧΟΛΙΑΣΜΟΣ

Παραθέτουμε αρχικά τα συγκεντρωτικά βέλτιστα αποτελέσματα για τα δύο σύνολα δεδομένων (5 κατηγορίες για το Amazon) και τα δύο είδη νευρωνικών δικτύων με στρογγυλοποίηση στα 3 ψηφία.

Πίνακας 5.1: Συγκεντρωτικά αποτελέσματα

	Sentiment 140	Movies and TV	Sports and Outdoors	Toys and Games
ΣΝΔ	0.816	0.686	0.721	0.687
ΑΜΠ	0.813	0.687	0.723	0.701

Όπως είναι λογικό, στην περίπτωση των 2 κατηγοριών το σύστημα τα πάει καλύτερα από ότι στις 5, αλλά κοιτώντας και την πρόσφατη βιβλιογραφία ([41]) είναι σε καλά επίπεδα. Για δοκιμή συγκρίναμε την απόδοση των νευρωνικών δικτύων με πιο παραδοσιακές μεθόδους μηχανικής μάθησης που έχουν παραγκωνιστεί τα τελευταία χρόνια όπως ο απλοϊκός ταξινομητής Bayes (Naive Bayes) και οι μηχανές διανυσματικής υποστήριξης (Support Vector Machines-SVM) τους οποίους υλοποιήσαμε με την βοήθεια της βιβλιοθήκης scikit-learn [48].

Πίνακας 5.2: Σύγκριση με άλλες μεθόδους μηχανικής μάθησης

	Movies and TV	Sports and Outdoors	Toys and Games
ΣΝΔ	0.686	0.721	0.687
ΑΜΠ	0.687	0.723	0.701
Naive Bayes	0.6203	0.6648	0.6425
SVM	0.5648	0.6617	0.6396

Είναι εμφανές πως οι παραδοσιακοί αλγόριθμοι μηχανικής μάθησης δεν μπορούν να ανταπεξέλθουν το ίδιο καλά σε εφαρμογές ΑΣ. Ο μεγάλος όγκος δεδομένων ευνοεί σημαντικά τα νευρωνικά δίκτυα τα οποία έχουν στην διάθεση τους περισσότερα δεδομένα για την βελτιστοποίηση των βαρών. Επίσης οι μηχανές διανυσματικής υποστήριξης χρειάζονται πάρα πολύ χρόνο για την εκπαίδευση στα σύνολα δεδομένων που χρησιμοποιήσαμε καθώς η πολυπλοκότητα τους εξαρτάται άμεσα από τον αριθμό των διαθέσιμων προτύπων.

Ενδιαφέρον είναι επίσης να δούμε την επιρροή της σύνοψης στο αποτέλεσμα για την δύσκολη περίπτωση των 5 κατηγοριών όπως αναφέραμε στο προηγούμενο κεφάλαιο.

Πίνακας 5.3: Επιρροή της σύνοψης στο αποτέλεσμα

	Movies and TV	Movies and TV(με σύνοψη)	Toys and Games	Toys and Games (με σύνοψη)	Sports and Outdoors	Sports and Outdoors (με σύνοψη)
ΣΝΔ	0.673	0.686	0.671	0.687	0.704	0.721
ΑΜΠ	0.677	0.687	0.697	0.701	0.696	0.723

Ανάλογα με την κατηγορία παρατηρήσαμε μια αύξηση της τάξης του 1-3% στην απόδοση. Δεδομένου του αριθμού των κατηγοριών αυτή η αύξηση δεν είναι αμελητέα. Τα νευρωνικά δίκτυα που χρησιμοποιήσαμε είναι σε θέση να εκμεταλλευτούν την πρόσθετη πληροφορία που δίνει η συγκόλληση της σύνοψης με την αρχή κάθε κριτικής, προσθέτοντας παραπάνω χρήσιμα συμφραζόμενα.

Υπάρχουν πολλές παράμετροι οι τιμές των οποίων μπορούν να μεταβληθούν για να δώσουν διαφορετικά αποτελέσματα. Αυτές είναι ο αριθμός των διαστάσεων των διανυσμάτων λέξεων, το μέγιστο μήκος κάθε κειμένου, ο αριθμός των νευρώνων, ο αριθμός των φίλτρων και το μέγεθος του πυρήνα για τα ΣΝΔ , το μέγεθος της δέσμης. Μετά από αρκετές δοκιμές διαπιστώσαμε πως οι περισσότερες από αυτές, εφόσον είναι εντός λογικών πλαισίων δεν έχουν ιδιαίτερη επίδραση στο αποτέλεσμα. Για λόγους πληρότητας παραθέτουμε τους πίνακες που δείχνουν τις βέλτιστες τιμές που προέκυψαν για κάθε σύνολο δεδομένων και για τα 2 είδη νευρωνικών δικτύων που χρησιμοποιήσαμε. Σημειώνουμε πως όσον αφορά τις αποκλειστικές παραμέτρους, για τα ΣΝΔ καταλήξαμε σε 250 φίλτρα, μέγεθος πυρήνα 3x3 και βήμα ολίσθησης 1. Για τις ΑΜΠ καταλήξαμε σε 64 κελιά.

Πίνακας 5.4: Τιμές παραμέτρων για το σύνολο Movies and TV

	Λεξιλόγιο	Μέγιστο μήκος	Διάσταση διανυσμάτων	Μέγεθος δέσμης
ΣΝΔ	10000	200	60	64
ΑΜΠ	15000	200	100	64

Πίνακας 5.5: Τιμές παραμέτρων για το σύνολο Toys and Games

	Λεξιλόγιο	Μέγιστο μήκος	Διάσταση διανυσμάτων	Μέγεθος δέσμης
ΣΝΔ	50000	100	256	64
ΑΜΠ	20000 (ή 50000)	100	100 (ή 256)	64

Πίνακας 5.6: Τιμές παραμέτρων για το σύνολο Sports and Outdoors

	Λεξιλόγιο	Μέγιστο μήκος	Διάσταση διανυσμάτων	Μέγεθος δέσμης
ΣΝΔ	10000	100	60	64
ΑΜΠ	20000	100	128	64

Πίνακας 5.7: Τιμές παραμέτρων για το σύνολο Sentiment140

	Λεξιλόγιο	Μέγιστο μήκος	Διάσταση διανυσμάτων	Μέγεθος δέσμης
ΣΝΔ	15000	50	64	32
ΑΜΠ	20000	75	128	32

Σε αυτή την περίπτωση υπάρχει μια εμφανής διαφοροποίηση στο μέγεθος δέσμης και στο μέγιστο μήκος λόγω της διαφορετικής φύσης των μηνυμάτων από χρήστες του Twitter σε σύγκριση με κείμενα κριτικών.

Αναφέραμε στο προηγούμενο κεφάλαιο πως για το σύνολο δεδομένων Amazon μπορούμε να μετατρέψουμε το πρόβλημα από πρόβλημα 5 κατηγοριών σε πρόβλημα 3 κατηγοριών. Από πρακτική πλευρά, από την στιγμή που μας ενδιαφέρει το συναίσθημα δεν είναι απαραίτητο να χρησιμοποιούμε όλες τις κατηγορίες για να εξάγουμε το συναίσθημα. Στον επόμενο πίνακα παρουσιάζονται τα αποτελέσματα για αυτή την εκδοχή του προβλήματος.

Πίνακας 5.8: Αποτελέσματα για το πρόβλημα 3 κατηγοριών

	Movies and TV	Sports and Outdoors	Toys and Games
ΣΝΔ	0.872	0.897	0.88
ΑΜΠ	0.867	0.885	0.879

Τα αποτελέσματα είναι πολύ καλά και αυτό φάνηκε και στο παράδειγμα με τις 2 ταινίες που παρουσιάστηκε στο τέλος του προηγούμενου κεφαλαίου.

Τέλος, αξίζει να σημειωθεί πως η χρήση της μεθόδου εγκατάλειψης και της έγκαιρης διακοπής της εκπαίδευσης βοήθησε σημαντικά στην αποφυγή της υπερπροσαρμογής η οποία είναι σχεδόν δεδομένη χωρίς κάποια μέθοδο κανονικοποίησης. Όπου αυτή εμφανίστηκε, ήταν σε πολύ μικρό βαθμό και χωρίς αξιολογες συνέπειες.

Αν και οι διαφορές δεν είναι ιδιαίτερα μεγάλες, υπάρχει η τάση το σύστημα να αποδίδει λίγο χειρότερα σε κριτικές ταινιών σε σύγκριση με τις άλλες 2 κατηγορίες, όσον αφορά το σύνολο δεδομένων Amazon. Αυτό είναι λογικό λόγω της αρκετά έντονης υποκειμενικής φύσης των κριτικών που αφορούν ταινίες, σε σύγκριση με προϊόντα.

6. ΣΥΜΠΕΡΑΣΜΑΤΑ

Φτάνοντας στο τέλος της εργασίας μπορούμε να κάνουμε μια ανασκόπηση όσων είδαμε. Υλοποιήσαμε ένα σύστημα Ανάλυσης Συναισθήματος από κείμενα το οποίο μπορεί να εξάγει από 2 έως 5 διαφορετικές κατηγορίες συναισθήματος, ανάλογα με το σύνολο δεδομένων που χρησιμοποιείται. Χρησιμοποιήσαμε πολύ πρόσφατες τεχνικές τόσο για την κατάλληλη προεπεξεργασία των όρων των κειμένων όσο και για το τμήμα που αφορά την μηχανική μάθηση. Τα Συνελικτικά Νευρωνικά Δίκτυα και οι Αναδρομικές Μονάδες με Πύλες έχουν δείξει την ισχύ τους στην Ανάλυση Συναισθήματος πολλές φορές τα τελευταία χρόνια και το ερευνητικό πεδίο συνεχώς εμπλουτίζεται με πολλές καινούργιες αρχιτεκτονικές οι οποίες βελτιώνουν τις υπάρχουσες. Αυτή η τάση φάνηκε και στην εργασία μας.

Πετύχαμε πολύ καλά αποτελέσματα τόσο στο σύνολο δεδομένων που αφορούσε κριτικές ταινιών και προϊόντων, όσο και στο σύνολο δεδομένων που αφορούσε μηνύματα χρηστών του Twitter. Τα δίκτυα που υλοποιήσαμε έδειξαν καλές επιδόσεις και στις δύσκολες περιπτώσεις που είναι τα μηνύματα στο Twitter και οι κριτικές ταινιών, εφαρμογές που έχουν πολύ έντονο το στοιχείο της υποκειμενικότητας .

Το πεδίο της Ανάλυσης Συναισθήματος είναι πραγματικά συναρπαστικό λόγω της άμεσης σχέσης που έχει με το ανθρώπινο στοιχείο. Υπάρχουν πάρα πολλές εφαρμογές στις οποίες μπορεί να φανεί χρήσιμη η Ανάλυση Συναισθήματος, και ενώ αναφέραμε μερικές μέσα στο κείμενο, φυσικά δεν εξαντλούνται εκεί. Η αύξηση της επεξεργαστικής ισχύος, ο ραγδαία αυξανόμενος όγκος δεδομένων λόγω του διαδικτύου και της υπολογιστικής νέφους, και η ολοένα μεγαλύτερη σύμπλευση της Τεχνητής Νοημοσύνης με την καθημερινότητα του ανθρώπου θα φέρουν στην επιφάνεια πολλές ακόμα δυνατότητες της Ανάλυσης Συναισθήματος.

Υπάρχουν ζητήματα σχετικά με την Ανάλυση Συναισθήματος τα οποία δεν εξερευνήθηκαν επαρκώς καθώς ξεφεύγουν από το πλαίσιο της έκτασης μιας μεταπτυχιακής εργασίας. Η διαχείριση του σαρκασμού, η ταξινόμηση υποκειμενικότητας, η εξαγωγή συναισθήματος για τα επιμέρους θέματα-χαρακτηριστικά στα οποία αναφέρεται μια κριτική και η χρήση παραπάνω του ενός μέσου (για παράδειγμα κείμενο μαζί με εικόνα) είναι μερικά από αυτά. Το ερευνητικό πεδίο είναι ενεργό και είναι βέβαιο ότι θα βλέπουμε συνεχή πρόοδο πάνω σε αυτά τα ζητήματα.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	Ελληνικός Όρος
Sentiment Analysis	Ανάλυση Συναισθήματος
Artificial Intelligence	Τεχνητή Νοημοσύνη
Convolutional Neural Networks	Συνελικτικά Νευρωνικά Δίκτυα
GRU	Αναδρομικές Μονάδες με Πύλες
Text Classification	Ταξινόμηση Κειμένων
Pattern Recognition	Αναγνώριση Προτύπων
Machine Learning	Μηχανική Μάθηση
Collaborative Filtering	Συνεργατικό Φιλτράρισμα
Agent	Πράκτορας
Supervised Learning	Μάθηση υπό επίβλεψη
Unsupervised Learning	Μάθηση χωρίς επίβλεψη
Reinforcement Learning	Ενισχυτική Μάθηση
Training Dataset	Σύνολο Εκπαίδευσης
Subjectivity Classification	Ταξινόμηση Υποκειμενικότητας
Recommender Systems	Συστήματα Προτάσεων
Document-level	Επίπεδο εγγράφου
Sentence-level	Επίπεδο πρότασης
Aspect-level	Επίπεδο πτυχής
Tweet	Τιτίβισμα
Artificial Neural Network	Τεχνητό Νευρωνικό Δίκτυο
Activation Function	Συνάρτηση ενεργοποίησης
Bias	Πόλωση
Perceptron	Τεχνητός νευρώνας
Hyperplane	Υπερεπίπεδο
Sigmoid function	Σιγμοειδής συνάρτηση
Feedforward	Εμπρόσθιας τροφοδότησης
Recurrent Neural Network	Αναδρομικό Νευρωνικό Δίκτυο
Loss function	Συνάρτηση απώλειας
Gradient descent	Κάθοδος με βάση την κλίση
Discrete convolution	Διακριτή συνέλιξη
Rectified Linear Unit	Γραμμικός Ανορθωτής
Feature map	Χάρτης χαρακτηριστικών
Pooling	Συγκέντρωση
Deep learning	Βαθιά μάθηση
Multiclass classification	Ταξινόμηση πολλών κατηγοριών
Padding	Γέμισμα
Cross-entropy	Διεντροπία
Batch training	Εκπαίδευση με δέσμες
Logistic regression	Λογιστική παλινδρόμηση
Bidirectional RNN	Αμφίδρομα αναδρομικά νευρωνικά δίκτυα
Hyperparameter	Υπερπαραμέτρος
Overfitting	Υπερπροσαρμογή
Underfitting	Υποπροσαρμογή
Cross Validation	Διασταυρωμένη Επικύρωση
Dropout	Μέθοδος εγκατάλειψης
Word embeddings	Διανύσματα λέξεων
Continuous bag-of-words	Συνεχής σάκος λέξεων

Skip-gram model	Μοντέλο πρόβλεψης συμπραζομένων
Negative sampling	Αρνητική δειγματοληψία
Class imbalance	Ανισορροπία κλάσεων
Tokenization	Διαχωρισμός λέξεων
Separable convolution	Διαχωρίσιμη συνέλιξη
Depthwise convolution	Συνέλιξη βάθους
Pointwise convolution	Συνέλιξη σημείου
Computational graph	Υπολογιστικός γράφος

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

ΑΣ	Ανάλυση Συναισθήματος
ΕΦΓ	Επεξεργασία Φυσικής Γλώσσας
ΤΝ	Τεχνητή Νοημοσύνη
ΣΝΔ	Συνελικτικά Νευρωνικά Δίκτυα
ΑΜΠ	Αναδρομικές Μονάδες με Πύλες
ΤΝΔ	Τεχνητά Νευρωνικά Δίκτυα
ReLU	Rectified Linear Unit
ΓΑ	Γραμμικός Ανορθωτής
ΑΝΔ	Αναδρομικά Νευρωνικά Δίκτυα
BCE	Binary cross-entropy
RNN	Recurrent Neural Networks
CBOW	Continuous bag-of-words
SVM	Support Vector Machines
API	Application programming interface

ΑΝΑΦΟΡΕΣ

- [1] Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας και Η. Σακελλαρίου, *Τεχνητή Νοημοσύνη*, 3η Έκδοση, Εκδόσεις Πανεπιστημίου Μακεδονίας, 2011.
- [2] Σ. Θεοδωρίδης και Κ. Κουτρούμπας, *Αναγνώριση Προτύπων*, 4η Έκδοση, Εκδόσεις Π.Χ. Πασχαλίδης, 2012.
- [3] Ι. Κερατζάκης, «Ταξινόμηση Κειμένων με χρήση Αναδρομικών Νευρωνικών Δικτύων», Πτυχιακή Εργασία, Τμήμα Πληροφορικής, Πανεπιστήμιο Πειραιώς, 2017.
- [4] B. Liu, "Sentiment analysis and opinion mining", *Synthesis lectures on human language technologies* 5.1, 2012, pp. 1-167.
- [5] Twitter, <https://twitter.com/>. [Προσπελάστηκε 7/9/19]
- [6] D. Tang, B. Qin και T. Liu, "Learning semantic representations of users and products for document level sentiment classification", *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 1014-1023.
- [7] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng και C. Potts, "Learning word vectors for sentiment analysis", *Proc. of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, Association for Computational Linguistics, 2011, pp. 142-150.
- [8] J. Wiebe, R. F. Bruce και T.P. O'Hara, "Development and use of a gold-standard data set for subjectivity classifications", *Proc. of the 37th annual meeting of the Association for Computational Linguistics*, 1999, pp. 246-253.
- [9] Y. Wang, M. Huang και L. Zhao, "Attention-based LSTM for aspect-level sentiment classification", *Proc. of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 606-615.
- [10] C.W Leung, C.S. Chan και L.F.Chung, "Integrating collaborative filtering and sentiment analysis: A rating inference approach", *Proc. of the ECAI 2006 workshop on recommender systems*, 2006, pp. 62-66.
- [11] A. Ceron, L.Curini, M.S. Iacus και G. Porro, "Every tweet counts? How sentiment analysis of social media can improve our knowledge of citizens' political preferences with an application to Italy and France", *New Media & Society*, vol. 16,no. 2,2014, pp. 340-358.
- [12] X. Wang, C.Zhang, Y. Ji, L. Sun, L. Wu και Z. Bao, "A depression detection model based on sentiment analysis in micro-blog social networks", *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, 2013, pp. 201-213.
- [13] C.Clavel και Z.Callejas, "Sentiment Analysis: From Opinion Mining to Human-Agent Interaction", *IEEE Transactions on Affective Computing*, vol. 7,no. 1, 2015, pp. 74-93.
- [14] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain", *Psychological Review*, vol.65, no. 6, 1958, pp. 386-408.
- [15] Desmos graphic calculator, <https://www.desmos.com/calculator>. [Προσπελάστηκε 10/9/19]
- [16] G. Cybenko, "Approximation by superpositions of a sigmoidal function", *Mathematics of control, signals and systems*, vol.2, no. 4, pp. 303-314.
- [17] P. Werbos, "Beyond Regression: "New tools for Prediction and Analysis in the Behavioral Sciences" ", Διδακτορική Διατριβή, Harvard University, 1974.
- [18] W. S. McCulloch και W. Pitts, "A logical calculus of the ideas immanent in nervous activity", *The bulletin of mathematical biophysics*, vol. 5, no. 4, 1943, pp. 115-133.
- [19] B. S. Damelin και W. Jr Miller, *The mathematics of signal processing*, vol.48, Cambridge University Press, 2012.
- [20] Y. LeCun, B. Boser, S. J. Denker, D. Henderson, E. R. Howard, W. Hubbard και D.L.Jackel, "Backpropagation applied to handwritten zip code recognition", *Neural computation*, vol. 1, no. 4, 1989, pp. 541-551.
- [21] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position", *Biological cybernetics*, vol. 36, no. 4, 1980, pp. 193-202.
- [22] M. Peemen, B. Mesman και H. Corporaal, "Efficiency optimization of trainable feature extractors for a consumer platform", *International Conference on Advanced Concepts for Intelligent Vision Systems*, Springer, 2011, pp. 293-304.
- [23] MaxpoolSample2.png,<https://computersciencewiki.org/index.php/File:MaxpoolSample2.png> [Προσπελάστηκε 10/9/19]
- [24] C. M. Bishop, *Pattern recognition and machine learning*, springer, 2006.
- [25] S. Dreiseitl και L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review", *Journal of Biomedical Informatics*, vol. 35, no. 5-6, 2002, pp. 352-359
- [26] Y. Kim, "Convolutional Neural Networks for Sentence Classification", *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

- [27] I. Goodfellow, Y. Bengio και A. Courville, *Deep learning*, MIT press, 2016.
- [28] M. Schuster και K. Paliwal, "Bidirectional recurrent neural networks", *IEEE Transactions on Signal Processing*, vol. 45, no. 11, 1997, pp. 2673-2681.
- [29] A. Mousa και B. Schuller, "Contextual Bidirectional Long Short-Term Memory Recurrent Neural Network Language Models: A Generative Approach to Sentiment Analysis", *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 2017, pp. 1023-1032.
- [30] K. Cho, B. V. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk και Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation", *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
- [31] P. Werbos, "Backpropagation through time: what it does and how to do it", *Proceedings of the IEEE*, vol. 78, no. 10, 1990, pp. 1550-1560.
- [32] D. Tang, B. Qin και T. Liu, "Document Modeling with Gated Recurrent Neural Network for Sentiment Classification", *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1422-1432.
- [33] Gated Recurrent Unit, fully gated version, https://en.wikipedia.org/wiki/Gated_recurrent_unit#/media/File:Gated_Recurrent_Unit_base_type.svg από τον χρήστη Jeblad - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=66225938> [Προσπελάστηκε 13/9/2019],
- [34] Y. Bengio, P. Simard και P. Frasconi, "Learning long-term dependencies with gradient descent is difficult", *IEEE Transactions on Neural Networks*, vol. 5, no. 2, 1994, pp. 157-166.
- [35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever και R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting", *The journal of machine learning research*, vol. 15, no. 1, 2014, pp. 1929-1958.
- [36] T. Mikolov, K. Chen, G. Corrado και J. Dean, "Efficient estimation of word representations in vector space.", *arXiv preprint arXiv:1301.3781*, 2013.
- [37] T. Mikolov, I. Sutskever, K. Chen, G. Corrado και J. Dean, "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*, 2013, pp. 3111-3119.
- [38] D. Jurafsky και J. H. Martin, *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River, NJ: Pearson Prentice Hall, 2009.
- [39] Y. Bengio, R. Ducharme, P. Vincent και C. Jauvin, "A neural probabilistic language model", *Journal of machine learning research*, vol. 3, Feb, 2003, pp. 1137-1155.
- [40] J. McAuley, C. Targett, Q. Shi, και A. V. D. Hengel, "Image-Based Recommendations on Styles and Substances", *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2015, pp. 43-52.
- [41] S. Ma, X. Sun, J. Lin και X. Ren, "A Hierarchical End-to-End Model for Jointly Improving Text Summarization and Sentiment Classification", *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018.
- [42] A. Go, R. Bhayani και L. Huang, "Twitter sentiment classification using distant supervision", *CS224N Project Report, Stanford*, 2009.
- [43] M. Abadi et al., "Tensorflow: A system for large-scale machine learning", *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265-283.
- [44] J. Nothman, H. Qin και R. Yurchak, "Stop Word Lists in Free Open-source Software Packages", *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, 2018, pp. 7-12.
- [45] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto και H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications", *arXiv preprint arXiv:1704.04861*, 2017.
- [46] L. Sifre και S. Mallat, "Rigid-motion scattering for image classification", *Διδακτορική Διατριβή*, Ecole Polytechnique, CMAP, 2014.
- [47] P. D. Kingma και J. Ba, "Adam: A method for stochastic optimization, Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)", 2015.
- [48] F. Pedregosa et al., "Scikit-learn: Machine learning in Python", *Journal of machine learning research*, vol. 12(Oct), 2011, pp. 2825-2830.
- [49] J. D. Hunter, "Matplotlib: A 2D Graphics Environment", *Computing in Science & Engineering*, vol. 9, no. 3, 2007, pp. 90-95.
- [50] S. V. D. Walt, S. C. Colbert και G. Varoquax, "The NumPy Array: A Structure for Efficient Numerical Computation", *Computing in Science & Engineering*, vol. 13, no. 2, 2011, pp. 22-30.
- [51] W. McKinney, "Data structures for statistical computing in python", *Proceedings of the 9th Python in Science Conference*, 2010, pp. 51-56.
- [52] F. Chollet, Keras, 2015. Διαθέσιμο στο <https://github.com/fchollet/keras>