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

Ανάλυση Συναισθήματος σε Δεδομένα του Κοινωνικού Δικτύου Twitter με Μεθόδους Μηχανικής Μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεώργιος Κ. Μαστραπάς

Επιβλέπων : Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2016

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

Ανάλυση Συναισθήματος σε Δεδομένα του Κοινωνικού Δικτύου Twitter με Μεθόδους Μηχανικής Μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεώργιος Κ. Μαστραπάς

Επιβλέπων : Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 19η Ιουλίου 2016.

.....	.....	.....
Στέφανος Κόλλιας	Κωνσταντίνος Καρπούζης	Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.	Διευθυντής Ερευνών Ε.Π.Ι.Σ.Ε.Υ. - Ε.Μ.Π.	Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2016

.....

Γεώργιος Κ. Μαστραπάς

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

Περίληψη

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι η ανάλυση συναισθήματος σε δεδομένα του κοινωνικού δικτύου Twitter με μεθόδους επιβλεπόμενης μηχανικής μάθησης. Ανάλυση συναισθήματος καλείται η αυτοματοποιημένη διαδικασία εξαγωγής πληροφοριών για την συναισθηματική πολικότητα ενός σώματος κειμένου και συχνά αναφέρεται εναλλακτικά ως εξόρυξη γνώμης. Αποτελεί ένα πεδίο έρευνας, που προσελκύει έντονο ενδιαφέρον τα τελευταία χρόνια εξαιτίας της μεγάλης επιρροής των κοινωνικών δικτύων στην καθημερινότητά μας, του αυτοματοποιημένου τρόπου που παρέχει για την ανάλυση της γραπτής πληροφορίας που αφθονεί σε διαδικτυακές πηγές αλλά και της σημαντικής προόδου που σημειώνεται τελευταία στα πεδία της μηχανικής μάθησης, της τεχνητής νοημοσύνης και της βαθιάς μάθησης.

Η αναγνώριση του συναισθήματος γίνεται σε δύο κατηγορίες, θετικό και αρνητικό συναίσθημα και για τις ανάγκες της εργασίας χρησιμοποιείται ένα σύνολο από περίπου 20,800 tweets με αντίστοιχες ετικέτες συναισθήματος. Προτείνεται μία μέθοδος προεπεξεργασίας των tweets που χειρίζεται όλους τους ειδικούς όρους που απαντώνται σε αυτά και εξετάζονται διάφοροι αλγόριθμοι επιβλεπόμενης μάθησης. Αυτοί είναι οι αλγόριθμοι ταξινόμησης κατά Bayes, ο αλγόριθμος k-Nearest Neighbors, η λογιστική παλινδρόμηση ή αλγόριθμος μέγιστης εντροπίας, οι μηχανές διανυσμάτων υποστήριξης, τα τεχνητά νευρωνικά δίκτυα και τα συνελικτικά νευρωνικά δίκτυα. Παράλληλα, εξετάζονται διάφοροι τρόποι εξαγωγής χαρακτηριστικών από δεδομένα κειμένου και συγκεκριμένα η κλασσική μέθοδος Bag-of-Words με τις παραλλαγές term occurrence, term frequency και tf-idf (term frequency - inverse document frequency) και οι διανυσματικές αναπαραστάσεις λέξεων που καλούνται word vectors. Μελετάμε νευρωνικά γλωσσικά μοντέλα όπως το word2vec και count-based μοντέλα όπως το GloVe. Οι διανυσματικές αναπαραστάσεις συντίθενται με διάφορους απλούς τρόπους αλλά και με τον αλγόριθμο doc2vec. Οι παραπάνω ιδέες αξιολογούνται όλες στο σύνολο δεδομένων.

Η εργασία καταλήγει στο συμπέρασμα πως οι κλασσικές τεχνικές ανάλυσης συναισθήματος όπως ο αλγόριθμος μέγιστης εντροπίας ή οι μηχανές διανυσμάτων υποστήριξης, με Bag-of-Words χαρακτηριστικά συμπεριφέρονται πολύ καλά στο πρόβλημα παρέχοντας γρήγορες υλοποιήσεις και αξιόπιστες επιδόσεις. Ωστόσο οι διανυσματικές αναπαραστάσεις λέξεων σε συνδυασμό με τεχνικές βαθιάς μάθησης που εκμεταλλεύονται την πληροφορία της σύνταξης ή σειράς των λέξεων, όπως τα συνελικτικά νευρωνικά δίκτυα, παρουσιάζουν καλύτερες επιδόσεις οδηγώντας την υπολογιστική κατανόηση φυσικού λόγου ένα βήμα πιο κοντά στην ανθρώπινη.

Λέξεις - Κλειδιά

Ανάλυση Συναισθήματος, Μηχανική Μάθηση, Twitter, Βαθιά Μάθηση, Συνελικτικά Νευρωνικά Δίκτυα, Διανυσματικές Αναπαραστάσεις Λέξεων, word2vec, GloVe

Abstract

The subject of this diploma thesis is sentiment analysis in Twitter data, using methods of supervised machine learning. Sentiment analysis is the automated process for extracting information about the sentiment polarity of a given body of text and is often, alternatively referred to, as opinion mining. It is a field of study that currently attracts a lot of academic attention due to the impact of social networks in our everyday life, the automated way it offers for analyzing the written information hugely available in web sources, and also because of the substantial progress being made in the last years, in the fields of machine learning, artificial intelligence and deep learning.

The detection of sentiment polarity is made in two broad categories of sentiment, namely positive and negative sentiment. For the purpose of this project, we use a labeled dataset of approximately 20,800 tweets with the respective labels. A preprocessing method for Twitter data is proposed, that handles all the special tokens found in tweets, and a number of supervised learning algorithms are examined. These are the Naive Bayes classifier, the k-Nearest Neighbors algorithm, the Logistic Regression or Maximum Entropy classifier, the Support Vector Machine, the Artificial Neural Network and last but not least the Convolutional Neural Network. Additionally, we examine various ways of extracting features from text and specifically the Bag-of-Words model with the variations term occurrence, term frequency and tf-idf (term frequency - inverse document frequency) and the distributed vector representations of words which are simply called word vectors. These word vectors include neural language models like word2vec and count-based models like GloVe. The vector representations are being composed in various simple ways but also using the doc2vec model. All the above ideas are being tested in our Twitter dataset.

The dissertation finally concludes that the simple and classic techniques for sentiment analysis like the Maximum Entropy algorithm or the Support Vector Machine, with Bag-of-Words features, perform really well and offer fast computing and reliable performance. However, word vectors combined with deep learning techniques that take advantage of syntax or word order, like the Convolutional Neural Network, perform better, leading computer understanding of natural language one step closer to human.

Keywords

Sentiment Analysis, Machine Learning, Twitter, Deep Learning, Convolutional Neural Networks, Word Vectors, word2vec, GloVe

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στο εργαστήριο Ψηφιακής Επεξεργασίας Εικόνων, Βίντεο και Πολυμέσων (Image, Video and Multimedia systems Lab - IVML) του Εθνικού Μετσόβιου Πολυτεχνείου, υπό την επίβλεψη του κυρίου Στέφανου Κόλλια, καθηγητή του Ε.Μ.Π.

Αρχικά, θα ήθελα να ευχαριστήσω τον κύριο Στέφανο Κόλλια για την εμπιστοσύνη που μου έδειξε και την ευκαιρία που μου έδωσε να ασχοληθώ με το ενδιαφέρον αντικείμενο της παρούσας εργασίας αλλά και για το γεγονός ότι, μέσα από τις διαλέξεις του, με ενέπνευσε να ασχοληθώ με τα πεδία της μηχανικής μάθησης και της επεξεργασίας φυσικού λόγου.

Παράλληλα, θα ήθελα να ευχαριστήσω τον διευθυντή ερευνών Ε.Π.Ι.Σ.Ε.Υ. - Ε.Μ.Π. κύριο Κώστα Καρπούζη για την καθοδήγηση και τις πολύτιμες συμβουλές που συνετέλεσαν στην εκπόνηση της εργασίας αυτής.

Τέλος, θα ήθελα να ευχαριστήσω θερμά τους φίλους μου, για όλες τις ωραίες στιγμές αυτά τα έξι χρόνια των σπουδών και φυσικά την οικογένειά μου, για την αμέριστη στήριξη που μου παρέχουν σε κάθε βήμα.

Περιεχόμενα

Κατάλογος Σχημάτων

[1] https://en.wikipedia.org/wiki/Twitter
[2] https://en.wikipedia.org/wiki/Support_vector_machine
[3] http://eblearn.sourceforge.net/beginner_tutorial2_train.html

[4] http://nlp.stanford.edu/projects/glove/
[5] http://cs224d.stanford.edu/

Κατάλογος Πινάκων

Interviewer: Dr. Poole, what's it like living for the better part of a year in such close proximity with HAL?

Dr. Frank Poole: Well, it's pretty close to what you said about him earlier. He is just like a sixth member of the crew. You very quickly get adjusted to the idea that he talks and you think of him really just as another person.

Interviewer: In talking to the computer one gets the sense that he is capable of emotional responses. For example, when I asked him about his abilities, I sensed a certain pride in his answer about his accuracy and perfection. Do you believe that HAL has genuine emotions?

Dave Bowman: Well, he acts like he has genuine emotions. Um, of course he's programmed that way to make it easier for us to talk to him. But as to whether he has real feelings is something I don't think anyone can truthfully answer.

Απόσπασμα από την ταινία "2001: A Space Odyssey"
Stanley Kubrick, 1968

1 Εισαγωγή

Στην παρούσα διπλωματική εργασία γίνεται μία εισαγωγή στο αντικείμενο της ***ανάλυσης συναισθήματος*** ή ***sentiment analysis*** όπως συχνά συναντάται στη διεθνή βιβλιογραφία. Η ανάλυση συναισθήματος είναι ταυτόσημη με την έννοια της ***εξόρυξης γνώμης*** (***opinion mining***) και αναφέρεται στη διαδικασία της αναγνώρισης του συναισθηματικού υπόβαθρου που χαρακτηρίζει ένα σώμα κειμένου. Ανάλυση συναισθήματος μπορεί να υπάρξει και σε διαφορετικού τύπου δεδομένα, όπως για παράδειγμα στη μουσική αλλά επί το πλείστον αναφέρεται σε δεδομένα κειμένου (***text data***).

Η ανάλυση συναισθήματος σαν πρόβλημα μπορεί να προσεγγιστεί με διαφορετικούς τρόπους που θα αναλυθούν στη συνέχεια, ωστόσο αυτή η εργασία πραγματεύεται την προσέγγιση της ***μηχανικής μάθησης*** ή ***machine learning*** στην αγγλική ορολογία. Η μηχανική μάθηση είναι ο κλάδος της ***επιστήμης των υπολογιστών*** (***computer science***) που δίνει στις μηχανές, δηλαδή στους υπολογιστές, τη δυνατότητα να μαθαίνουν από την εμπειρία με έναν τρόπο που προσομοιάζει και εμπνέεται από την λειτουργία του ανθρώπινου εγκεφάλου. Η μηχανική μάθηση συνδέεται στενά με άλλους κλάδους των μαθηματικών και της επιστήμης των υπολογιστών όπως η ***ταξινόμιση*** (***classification***), η ***αναγνώριση προτύπων*** (***pattern recognition***) και η ***τεχνητή νοημοσύνη*** (***artificial intelligence***).

Οι παραπάνω έννοιες έρχονται κοντά και βρίσκουν εφαρμογή σε δεδομένα κοινωνικών δικτύων όπως το Twitter.

Στο παρόν κεφάλαιο λοιπόν θα επιχειρήσουμε μία σύντομη εισαγωγή στην ανάλυση συναισθήματος, στη μηχανική μάθηση, στα κοινωνικά δίκτυα και στις ιδιαιτερότητές τους και τέλος στο πώς όλα τα παραπάνω συνδέονται και δημιουργούν μαζί ένα ενδιαφέρον πεδίο έρευνας που τα τελευταία χρόνια προσελκύει έντονα το ακαδημαϊκό ενδιαφέρον.

1.1 Ανάλυση Συναισθήματος

1.1.1 Περιγραφή και Κατηγορίες της Ανάλυσης Συναισθήματος

Η *ανάλυση συναισθήματος* αποτελεί μία υποπεριοχή της *ταξινόμησης κειμένου* (*text classification*) και αναφέρεται στη διαδικασία εξαγωγής πληροφορίων για τη συναισθηματική κατάσταση ενός χρήστη μέσα από το γραπτό λόγο του. Χρησιμοποιεί τεχνικές *επεξεργασίας φυσικού λόγου* (*natural language processing* - NLP), στατιστικές μεθόδους και μεθόδους μηχανικής μάθησης για την ταξινόμηση ενός κειμένου σε κλάσεις που εκφράζουν συναίσθημα.

Ένας πρώτος διαχωρισμός της ανάλυσης συναισθήματος γίνεται με βάση την ακριβή έννοια της συναισθηματικής κατάστασης που επιχειρεί να προσδιορίσει. Αυτή σύμφωνα με τη ***wikipedia***[6] μπορεί να αναφέρεται είτε στη γενικότερη συναισθηματική κατάσταση του συγγραφέα κατά τη συγγραφή του κειμένου (***affective state***), είτε στο συναίσθημα που μεταδίδεται σκόπιμα από τον συγγραφέα στον αναγνώστη μέσω του κειμένου, είτε στην στάση – άποψη – εκτίμηση του συγγραφέα σχετικά με κάποιο θέμα. Στις πρώτες δύο περιπτώσεις η ταξινόμηση μπορεί να γίνει σε κλάσεις που εκφράζουν συναισθήματα αντιληπτά από τον άνθρωπο. Επιχειρείται δηλαδή η αναγνώριση πραγματικών συναισθημάτων στο κείμενο όπως η χαρά, η λύπη και ο θυμός ή και καταστάσεων όπως η ειρωνεία. Η ταξινόμηση ωστόσο μπορεί να γίνει και σε γενικότερες κλάσεις όπως θετικό, αρνητικό και ουδέτερο συναίσθημα. Στην τελευταία περίπτωση, όπου εξετάζεται η στάση ως προς κάποιο θέμα, η ταξινόμηση γίνεται συνήθως σε δύο (θετική στάση, αρνητική στάση), τρείς (θετική, ουδέτερη, αρνητική) ή πέντε (θετική, μάλλον θετική, ουδέτερη, μάλλον αρνητική, αρνητική) κλάσεις.

Ένας άλλος διαχωρισμός γίνεται με βάση το μέγεθος του κειμένου που εξετάζεται. Έτσι μπορεί να αναζητείται το συναίσθημα ή η πολικότητά του (***polarity***) σε ένα ολοκληρο κείμενο (***document-based sentiment analysis***), σε μία πρόταση (***sentence-based sentiment analysis***) ή ακόμα και σε μεμονωμένες φράσεις (***feature/aspect-based sentiment analysis***) όταν αυτές αναφέρονται σε χαρακτηριστικά μίας οντότητας (***features of an entity***) ως προς τα οποία αναζητούμε το συναίσθημα. Στη συνέχεια δίνονται κάποια παραδείγματα για να γίνει πιο κατανοητός ο παραπάνω διαχωρισμός.

Document-based sentiment analysis

War movies have been biased to one side or the other. This movie does not make heroes or enemies of the German U-boat sailors. Instead, it grips the viewer with realistic depictions of what it was like to be a U-boat sailor for the Germans in WWII. It starts off with young (17 year old to 25 year old) who have been filled with propaganda about the war effort and glorious battle. After this young crew of immature sailors start to experience the true horrors of war, you can not only see, but experience with them the boredom, laughter, camaraderie, team work and death. In a world where you have no windows, where your ears have to be your

[6] https://en.wikipedia.org/wiki/Sentiment_analysis

eyes, where a cat and mouse game is played and the looser dies, these young men age 10 to 15 years It makes the viewer realize the horror of submarine warfare in WWII. The most realistic war movie I have ever seen.

Το παραπάνω παράδειγμα αποτελεί κριτική ταινίας από την ιστοσελίδα ***imdb***[7]. Πρόκειται σαφώς για μία θετική κριτική και αναζήτηση του συναισθήματος σε μία τέτοια περίπτωση, δηλαδή στην περίπτωση ενός ολοκληρωμένου κειμένου, συνιστά document-based sentiment analysis. Παρόμοια θα μπορούσε να αναζητηθεί και η στάση ενός αρθρογράφου προς ένα πολιτικό συμβάν μέσα από ένα άρθρο πολιτικής εφημερίδας ή και η στάση ενός χρήστη προς κάποιο προϊόν μέσα από κάποια διαδικτυακή κριτική.

Sentence-based sentiment analysis

It's so laddish and juvenile, only teenage boys could possibly find it funny.

Πρόκειται πάλι για μία κριτική ταινίας, η οποία όμως αυτή τη φορά συμπυκνώνεται σε μία πρόταση. Η αναγνώριση συναισθήματος σε αυτή την περίπτωση είναι αρκετά πιο δύσκολη καθώς τα δεδομένα είναι λιγότερα και συνήθως η σειρά και η σύνταξη των λέξεων παίζουν σημαντικό ρόλο. Τυπικά, sentence-based sentiment analysis μπορεί να υφίσταται και σε μικρό αριθμό προτάσεων. Η ανάλυση συναισθήματος σε tweets που θα απασχολήσει την παρούσα διπλωματική εργασία ανήκει σε αυτή την κατηγορία αφού τα tweets είναι κείμενα μικρού μήκους αποτελούμενα συνήθως από μία με δύο προτάσεις.

Feature/aspect-based sentiment analysis

The screen in this smartphone is beautiful with realistic colors and wide viewing angles, the camera though could perform better under low light.

Σε αυτή την περίπτωση έχουμε την κριτική ενός smartphone από το διαδίκτυο. Η οντότητα (entity) είναι το smartphone και τα χαρακτηριστικά της οντότητας είναι η οθόνη του και η κάμερά του. Σκοπός της feature/aspect-based sentiment analysis είναι η αναγνώριση του συναισθήματος στις φράσεις ***beautiful with realistic colors and wide viewing angles*** και ***could perform better under low light*** που η μεν αναφέρεται στην οθόνη και η δε στην κάμερα του κινητού τηλεφώνου.

1.1.2 Εφαρμογές

Η ανάλυση συναισθήματος είναι ένας σχετικά νέος τομέας έρευνας. Οι πρώτες απόπειρες έγιναν στις αρχές του 21ου αιώνα από τον Turney [27] και τους Pang και Lee [19] οι οποίοι επιχείρησαν να ταξινομήσουν μεγάλου μήκους κείμενα σε κατηγορίες ανάλογα με το συνολικό συναίσθημα που εκφράζουν. Μέχρι τότε είχε μελετηθεί επαρκώς το θέμα του ***topic classification*** (ταξινόμηση κειμένων με βάση το θέμα τους) αλλά όχι η ταξινόμηση με βάση το συναίσθημα. Ο Turney προσπάθησε να ταξινομήσει διαδικτυακές κριτικές αυτοκινήτων, ταινιών, τραπεζών και ταξιδιωτικών προορισμών χρησιμοποιώντας

[7] http://www.imdb.com/

στατιστικές μεθόδους ενώ οι Pang και Lee κριτικές ταινιών με τη βοήθεια κλασσικών αλγορίθμων μηχανικής μάθησης οι οποίοι είχαν δώσει καλά αποτελέσματα στο πεδίο του topic classification.

Δεκαπέντε χρόνια μετά η ανάλυση συναισθήματος αποτελεί ένα ενεργό πεδίο έρευνας που προσελκύει μεγάλο ενδιαφέρον και αυτό οφείλεται κατά βάση σε δύο παράγοντες:

- *Στην έκρηξη του διαδικτύου με την έλευση του Web 2.0.* Καθώς ζούμε στην εποχή του Internet, των υπολογιστών και των smartphones, δισεκατομμύρια άνθρωποι στον πλανήτη έχουν πρόσβαση στο διαδίκτυο και σε τεράστιους όγκους δεδομένων είτε εικόνων, είτε βίντεο, είτε δεδομένων κειμένου, τα λεγόμενα ***big data***. Το Internet έχει γίνει κομμάτι της καθημερινότητας των ανθρώπων οι οποίοι πλέον το χρησιμοποιούν για οποιαδήποτε πληροφορία χρειάζονται αλλά και για να κοινωνικοποηθούν μέσω κοινωνικών δικτύων. Τα κοινωνικά δίκτυα όπως το ***Facebook*** και το ***Twitter*** αλλά και το ***4chan*** ή το ***reddit***, που βασίζονται στην ανωνυμία των χρηστών, δίνουν στο μέσο χρήστη τη δυνατότητα να συνδεθεί και να επικοινωνήσει με οποιονδήποτε στον πλανήτη και να ανεβάσει δεδομένα στο διαδίκτυο. Από τη σκοπιά της ανάλυσης συναισθήματος σαν ερευνητικό πεδίο οι παραπάνω συνθήκες δίνουν στον ερευνητή πρόσβαση σε πολύ μεγάλους όγκους δεδομένων που βοηθούν την έρευνά του, σε αντίθεση με το τι επικρατούσε μερικά χρόνια πριν. Πλεόν ο καθένας μπορεί να έχει πρόσβαση σε κριτικές ταινιών στο imdb, κριτικές προϊόντων στο amazon, γνώμες και απόψεις στα κοινωνικά δίκτυα και πρακτικά οποιαδήποτε μορφή πληροφορίας. Ενδεικτική είναι η πρόοδος που έχει σημειωθεί τα τελευταία χρόνια στον τομέα της ***υπολογιστικής όρασης*** (***computer vision***) καθώς εταιρείες όπως η Google έχουν τη δυνατότητα να εφαρμόζουν αλγορίθμους σε μεγάλους όγκους δεδομένων.

- *Στην αύξηση των υπολογιστικών πόρων που είναι διαθέσιμοι στο μέσο χρήστη.* Η ανάλυση συναισθήματος αλλά και γενικότερα η μηχανική μάθηση εκτός από επαρκή δεδομένα απαιτούν και υπολογιστικούς πόρους. Η συνεχής αύξηση της επεξεργαστικής ισχύος των υπολογιστών διευρύνει συνεχώς τα όρια των δυνατοτήτων της τεχνολογίας. Πλέον επιτρέπει στον καθένα να υλοποιήσει και να τρέξει απαιτητικούς αλγορίθμους στον προσωπικό του υπολογιστή ή ακόμα και σε υπερυπολογιστές (***super computers***, ***gpu clusters***) τους οποίους διαθέτουν εταιρείες σε χρήστες. Χαρακτηριστικό των παραπάνω είναι η ***βαθιά μάθηση*** (***deep learning***). Τα τελευταία χρόνια οι ερευνητές κατάφεραν να υλοποιήσουν αποτελεσματικά, βαθιά νευρωνικά δίκτυα σε μεγάλους όγκους δεδομένων, με τα αποτελέσματα να είναι παραπάνω από ενθαρρυντικά, παρόλο που οι αλγόριθμοι αυτοί θεωρητικά είχαν θεμελιωθεί αρκετά χρόνια πριν.

Συνοψίζοντας, η πρόσβαση στα πρακτικά άπειρα δεδομένα του διαδικτύου και η τεχνολογική πρόοδος στον τομέα των υπολογιστών έχουν αναζωπυρώσει το ενδιαφέρον τα τελευταία χρόνια γύρω από τη μηχανική μάθηση και την τεχνητή νοημοσύνη. Η ανάλυση συναισθήματος πλέον γίνεται αποτελεσματικά, είναι εφικτή από το μέσο χρήστη και ήδη χρησιμοποιείται από εταιρείες, οργανισμούς και ερευνητές σε μεγάλη κλίμακα. Κάποιες από τις εφαρμογές, σύμφωνα με το [18] είναι οι παρακάτω:

- Οι εταιρείες χρησιμοποιούν αλγορίθμους ανάλυσης συναισθήματος σε διαδικτυακά δεδομένα για να εξάγουν πληροφορίες για την αποδοχή των προϊόντων τους από το σύνολο των καταναλωτών. Με αυτό τον τρόπο αντλούν αποτελεσματικά feedback για την ποιότητα των προϊόντων και των υπηρεσιών τους (business intelligence).

- Η ανάλυση συναισθήματος χρησιμοποιείται για την εξόρυξη πληροφοριών σχετικά με τη στάση της κοινής γνώμης σε διάφορα ζητήματα, πολιτικά, κοινωνικά ή οικονομικά. Η παραπάνω εφαρμογή συχνά αναφέρεται ως εκδημοκρατισμός των κοινωνικών δικτύων καθώς επιτρέπει την έκφραση της άποψης της πλειοψηφίας πάνω σε ένα θέμα. Δεδομένης της αποδοχής κοινωνικών δικτύων όπως το Twitter στη σημερινή κοινωνία, μπορούμε να πούμε ότι η πλειοψηφία των κοινωνικών δικτύων εκφράζει σε μεγάλο βαθμό και τη πλειοψηφία της κοινωνίας, χωρίς αυτό να σημαίνει φυσικά, ότι δεν απαιτείται και ιδιαίτερη προσοχή στην εξαγωγή τέτοιων συμπερασμάτων.

1.1.3 Προσεγγίσεις

Κατά κύριο λόγο η ανάλυση συναισθήματος προσεγγίζεται με δύο διαφορετικούς τρόπους. Ο πρώτος αναφέρεται ως lexicon-based και κάνει χρήση συναισθηματικών λεξικών για να αποδώσει συναισθηματική βαθμολογία σε λέξεις και φράσεις. Στη συνέχεια συνθέτει τις λέξεις και τις φράσεις για να προκύψει συναισθηματική βαθμολογία για όλο το προς εξέταση κείμενο. Μία απλή υλοποίηση της lexicon-based προσέγγισης, στην περίπτωση απλών προτάσεων, είναι η πρόσθεση των συναισθηματικών βαθμολογιών των λέξεων με τον τρόπο που υποδεικνύει το συντακτικό δέντρο της πρότασης. Έτσι ξεκινώντας από τα φύλλα, που αποτελούν λέξεις, γίνονται διαδοχικές αθροίσεις των βαθμολογιών μέχρι το τελικό σκορ στη ρίζα του δέντρου, που χαρακτηρίζει όλη την πρόταση. Ανάλογα με την τελική βαθμολογία η πρόταση χαρακτηρίζεται αρνητική ή θετική. Τέτοιες προσεγγίσεις συχνά περιέχουν και χειρισμό της άρνησης (negation handling). Καθώς η άρνηση αντιστρέφει το συναίσθημα της λέξης (cool - not cool), μπορεί για παράδειγμα η συναισθηματική βαθμολογία λέξεων που συντάσσονται με άρνηση απλά να αλλάζει πρόσημο. Διατηρείται δηλαδή η ένταση του συναισθήματος αλλά αντιστρέφεται η πολικότητα. Ένα ιδιαίτερα δημοφιλές λεξικό συναισθηματικής πολικότητας είναι το SentiWordNet [3], η εκδοχή του WordNet που αποδίδει συναισθηματική βαθμολογία σε λέξεις σε δύο επίπεδα, positive και negative.

Ο δεύτερος τρόπος προσέγγισης του προβλήματος της ανάλυσης συναισθήματος είναι η εξαγωγή χαρακτηριστικών και η χρήση μεθόδων μηχανικής μάθησης. Η προσέγγιση αυτή θα απασχολήσει την παρούσα εργασία και θα αναλυθεί εκτενώς στα επόμενα κεφάλαια. Ξεκινάμε, δίνοντας τον ορισμό της μηχανικής μάθησης.

1.2 Μηχανική Μάθηση

1.2.1 Γενικά

Όπως αναφέρθηκε προηγουμένως, η μηχανική μάθηση είναι ο κλάδος της επιστήμης των υπολογιστών που μελετά και κατασκευάζει αλγορίθμους και τεχνικές που δίνουν τη δυνατότητα στον υπολογιστή να μαθαίνει από την εμπειρία. Βασικό χαρακτηριστικό της μηχανικής μάθησης είναι ότι αναζητεί πρότυπα (***patterns***) και σχέσεις στα δεδομένα με σκοπό να τα μοντελοποιήσει και να κάνει προβλέψεις πάνω σε αυτά. Σύμφωνα με τον πρώιμο ορισμό που έδωσε ο Arthur Samuel το 1959 η μηχανική μάθηση είναι :

Πεδίο μελέτης που δίνει στους υπολογιστές τη δυνατότητα να μαθαίνουν χωρίς να έχουν προγραμματιστεί αυστηρά.

Ένας πιο επίσημος ορισμός προτάθηκε από τον Tom M. Mitchell και είναι ο ακόλουθος

Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από εμπειρία E ως προς μία κλάση εργασιών T και ένα μέτρο επίδοσης P, αν η επίδοσή του σε εργασίες της κλάσης T, όπως αποτιμάται από το μέτρο P, βελτιώνεται με την εμπειρία E.

Στο γενικό πλαίσιο της μηχανικής μάθησης, σε ένα σύστημα παρουσιάζονται *δεδομένα εκπαίδευσης* (***training data***) και το σύστημα με τη βοήθεια ενός αλγορίθμου μηχανικής μάθησης, επιχειρεί να εκπαιδευτεί πάνω στα δεδομένα αυτά. Κατά τη φάση αυτή, που ονομάζεται *εκπαίδευση* (***training***), το σύστημα ***μαθαίνει από την εμπειρία*** και σχηματίζει μοντέλα των δεδομένων. Στη συνέχεια του παρουσιάζονται *δεδομένα δοκιμής* (***testing data***) τα οποία δεν έχει συναντήσει κατά την εκπαίδευσή του και με βάση τις προβλέψεις που κάνει πάνω στα δεδομένα αυτά, αξιολογείται η επίδοσή του. Τα δεδομένα που τροφοδοτούν το σύστημα κατά την εκπαίδευση μπορεί να συνοδεύονται από κάποια επιθυμητή απόκριση, η οποία κατευθύνει το σύστημα, οπότε η μάθηση σε αυτή την περίπτωση ονομάζεται *επιβλεπόμενη* (***supervised***). Στην αντίθετη περίπτωση που το σύστημα δεν ενημερώνεται για την απόκριση που πρέπει να έχει στα δεδομένα, η μάθηση χαρακτηρίζεται *μη επιβλεπόμενη* (***unsupervised***). Στην υβριδική περίπτωση που μερικά μόνο από τα δεδομένα συνοδεύονται απο επιθυμητή έξοδο (συνήθως λίγα σε σχέση με το συνολικό αριθμό δεδομένων), έχουμε *ημι-επιβλεπόμενη μάθηση* (***semi-supervised***). Τέλος υπάρχει μία ακόμη μορφή μηχανικής μάθησης κατά την οποία το σύστημα καλείται να φέρει εις πέρας μία αποστολή μέσα σε ένα μεταβλητό περιβάλλον. Στο σύστημα παρουσιάζονται αρχικά αλλά και νέα δεδομένα καθ'όλη τη διάρκεια της προσπάθειάς του. Τα δεδομένα αυτά δε συνοδεύονται από κάποια επιθυμητή απόκριση αλλά το σύστημα λαμβάνει τελικώς ως feedback το πόσο καλά εκτέλεσε την αποστολή. Έτσι το σύστημα μέσα από τη διαδικασία αυτή μαθαίνει και αυτή η μορφή μάθησης χαρακτηρίζεται *ενισχυτική* (***reinforcement***).

Έχουμε λοιπόν τις εξής κατηγορίες μηχανικής μάθησης:

Επιβλεπόμενη Μάθηση – Supervised Learning

Στο σύστημα παρουσιάζονται δεδομένα εισόδου με τις αντίστοιχες επιθυμητές εξόδους και αυτό προσπαθεί να «μάθει» τον τρόπο με τον οποίο οι είσοδοι αντιστοιχίζονται στις εξόδους, έτσι ώστε μελλοντικά να προβλέπει την έξοδο σε νέα δεδομένα εισόδου. Χαρακτηριστικά παραδείγματα αλγορίθμων επιβλεπόμενης μάθησης είναι τα *τεχνητά νευρωνικά δίκτυα* (*artificial neural networks*), οι *μηχανές διανυσμάτων υποστήριξης* (*support vector machines*) και οι *ταξινομητές Bayes* (*Bayes classifiers*).

Σημειώνεται σε αυτό το σημείο ότι στην παρούσα εργασία θα μελετηθεί η επιβλεπόμενη μηχανική μάθηση και θα εξεταστούν μόνο αντίστοιχοι αλγόριθμοι. Εισαγωγή στην θεωρία μερικών απο τους αλγορίθμους αυτούς γίνεται στο κεφάλαιο 3.

Μη Επιβλεπόμενη Μάθηση – Unsupervised Learning

Σε αυτή τη μορφή μηχανικής μάθησης, το σύστημα εκπαιδεύεται με δεδομένα που δεν έχουν χαρακτηριστεί από επιθυμητή έξοδο (*unlabeled data*). Δεν υπάρχει δυνατότητα αξιολόγησης της επίδοσης του δικτύου αφού δεν είναι γνωστή η επιθυμητή απόκριση. Συνήθως σκοπός αυτού του είδους μηχανικής μάθησης είναι η ανακάλυψη κρυφών δομών, προτύπων ή συσχετίσεων πίσω από τα δεδομένα. Στην πιο απλή περίπτωση, μη επιβλεπόμενη μάθηση χρησιμοποιείται για τη *συσταδοποίηση* δεδομένων (*clustering*) δηλαδή το χωρισμό τους σε κατηγορίες, έτσι ώστε παρόμοια δεδομένα να ανήκουν στην ίδια κατηγορία. Τυπικοί αλγόριθμοι αυτού του είδους μηχανικής μάθησης είναι ο αλγόριθμος *συσταδοποίησης k-means* (*k-means clustering*) και ειδικοί τύποι νευρωνικών δικτύων που καλούνται *χάρτες αυτο-οργάνωσης* (*self-organizing maps - SOMs*).

Ημι-επιβλεπόμενη Μάθηση – Semi-supervised Learning

Αυτή είναι η υβριδική προσέγγιση της μηχανικής μάθησης που συνδυάζει την επιβλεπόμενη και την μη-επιβλεπόμενη μάθηση. Στο σύστημα, κατά την εκπαίδευσή του, παρουσιάζονται τόσο *labeled* όσο και *unlabeled* δεδομένα. Συνήθως τα labeled δεδομένα, αυτά δηλαδή που συνοδεύονται από επιθυμητή απόκριση, είναι αρκετά λιγότερα. Αυτό συμβαίνει γιατί στην πράξη τα labeled δεδομένα είναι δυσεύρετα σε αντίθεση με τα unlabeled που υπάρχουν σε αφθονία. Πιο αναλυτικά τα labeled δεδομένα απαιτούν κάποια ανθρώπινη παρέμβαση, δηλαδή κάποιον άνθρωπο να τα χαρακτηρίσει ως προς την επιθυμητή απόκριση. Αυτό σε μερικές περιπτώσεις μπορεί να γίνει με αυτοματοποιημένο τρόπο, όπως στην ανάλυση συναισθήματος σε κριτικές ταινιών στο διαδίκτυο όπου κάθε χρήστης μαζί με την κριτική του αφήνει και κάποια βαθμολογία (για παράδειγμα 1 έως 10 αστέρια). Συνήθως όμως πρέπει να γίνει χειροκίνητα από άνθρωπο ή ομάδα ανθρώπων. Γενικότερα αυτό είναι και το πλεονέκτημα της μη επιβλεπόμενης μάθησης έναντι της επιβλεπόμενης. Απαιτεί unlabeled δεδομένα στα οποία η πρόσβαση είναι εύκολη. Η επιβλεπόμενη μάθηση ωστόσο με επαρκή δεδομένα αποδίδει καλύτερα στην πράξη.

Ενισχυτική Μάθηση – Reinforcement Learning

Χαρακτηριστικά παραδείγματα ενισχυτικής μάθησης είναι τα *self-driving cars* και τα υπολογιστικά συστήματα που μαθαίνουν να παίζουν παιχνίδια όπως το σκάκι ή και πρόσφατα το Go[8].

1.2.2 Το Πρόβλημά μας

Μετά από τη σύντομη αυτή εισαγωγή στην έννοια της μηχανικής μάθησης, μπορούμε πλέον να σκιαγραφήσουμε το πρόβλημα που θα απασχολήσει αυτή την εργασία. Το πρόβλημα αυτό, όπως ειπώθηκε και παραπάνω, είναι ένα πρόβλημα επιβλεπόμενης μάθησης. Σκοπός είναι η δημιουργία ενός συστήματος που θα είναι σε θέση να αναγνωρίζει αυτόματα το συνολικό συναίσθημα πίσω από τις δημοσιεύσεις του κοινωνικού δικτύου Twitter. Ορθότερα, μας απασχολεί η πολικότητα του συναισθήματος, οπότε η ταξινόμηση θα γίνει σε δύο κλάσεις, θετικό και αρνητικό συναίσθημα. Στην προηγούμενη ενότητα έγινε ένας διαχωρισμός της ακριβούς έννοιας του συναισθήματος που πρέπει να προσδιοριστεί και συγκεκριμένα έαν αυτό εκφράζει συναισθηματική κατάσταση, εκτίμηση προς κάποιο θέμα ή συναίσθημα που μεταδίδεται στον αναγνώστη. Στην υλοποίησή μας, ζητούμενο είναι ο προσδιορισμός της πολικότητας σε κάθε περίπτωση. Για παράδειγμα η παρακάτω πρόταση

I feel awesome!

εκφράζει συναισθηματική κατάσταση. Σε αυτή την περίπτωση θα πρέπει να αναγνωρίζεται θετική πολικότητα. Η επόμενη πρόταση

This speaker-set is plain fantastic. Not the best sounding of course but you get so much value for your money.

περιέχει εκτίμηση για κάποιο θέμα και μάλιστα μικτό συναίσθημα. Η συνολική πολικότητα ωστόσο είναι θετική και το σύστημα μας θα πρέπει να την αναγνωρίζει.

1.3 Κοινωνικά Δίκτυα και Twitter

Το Twitter[9] είναι ένας από τους πλέον δημοφιλείς ιστότοπους κοινωνικής δικτύωσης που επιτρέπει στους χρήστες να κοινοποιήσουν μηνύματα μεγέθους το πολύ 140 χαρακτήρων που καλούνται tweets. Δημιουργήθηκε το Μάρτιο του 2006 και μέσα σε μικρό χρονικό διάστημα προσέλκυσε πολύ μεγάλο αριθμό χρηστών. Πλέον εξυπηρετεί περισσότερους από 310 εκατομμύρια ενεργούς χρήστες το μήνα[10].

Τα βασικά χαρακτηριστικά των μηνυμάτων που ανταλλάσονται μέσω Twitter είναι

- Μικρό μέγεθος και περιεκτικότητα, το πολύ 140 χαρακτήρες.

[8] https://deepmind.com/
[9] https://twitter.com/
[10] https://en.wikipedia.org/wiki/Twitter

- Ανεπίσημος λόγος, πολλές φορές χωρίς έμφαση στη σύνταξη και τη γραμματική ορθότητα.
- Πολύ συχνά περιέχουν συντομεύσεις και ακρωνύμια.
- Επιμηκυμένες λέξεις, επαναλήψεις σημείων στίξης και λέξεις γραμμένες μόνο με κεφαλαία, όλα προς απόδοση έμφασης.
- Ειδικά tokens όπως hashtags, usernames, URLs, retweets και emoticons.

Σχήμα 1.1 : Το λογότυπο του Twitter

Τα hashtags χαρακτηρίζονται από τον χαρακτήρα #, με τον οποίο πάντα ξεκινάνε, και την απουσία κενών. Μπορεί να δηλώνουν το θέμα του tweet, συναισθηματική κατάσταση ή και οτιδήποτε άλλο. Συνήθως χρησιμοποιούνται για την επισήμανση κάποιας τάσης (trend) μεταξύ των χρηστών. Το σώμα των hashtags μπορεί να περιέχει μόνο πεζούς χαρακτήρες αλλά επίσης συνηθίζεται να περιέχει κεφαλαία για τον διαχωρισμό των λέξεων καθώς δεν επιτρέπεται η χρήση κενών.

Τα usernames ξεκινάνε με τον χαρακτήρα @ και επισημαίνουν κάποιον χρήστη του δικτύου.

Το token RT δηλώνει retweet δηλαδή κοινοποίηση του μηνύματος άλλου χρήστη.

Τα URLs είναι ηλεκτρονικές διευθύνσεις που επισημαίνονται σε μηνύματα. Ξεκινάνε σχεδόν πάντα με `http://`, `https://` ή `www.`.

Τα emoticons είναι strings που προσομοιάζουν ανθρώπινες εκφράσεις με χαρακτήρες για να δηλώσουν συναισθηματική κατάσταση όπως χαρά, λύπη, θυμό, έκπληξη και γέλιο. Η πληροφορία που μεταφέρουν είναι πολύτιμη για το πρόβλημα της ανάλυσης συναισθήματος.

Στην ενότητα 2.2 περιγράφεται αναλυτικά η διαδικασία που ακολουθούμε για τον χειρισμό των παραπάνω ειδικών strings που απαντώνται στα tweets.

1.4 Λογισμικό

Η εργασία υλοποιείται σε python[11] με την εξαίρεση του συνελικτικού δικτύου που υλοποιείται με τη βοήθεια του deep learning framework *torch*[12]. Τα βασικότερα modules που χρησιμοποιεί η υλοποίησή μας είναι το nltk[13], για την επεξεργασία των text data, το numpy[14], βασικό εργαλείο της python για πράξεις μεταξύ πινάκων, το scikit-learn[15] για την αποδοτική εφαρμογή των αλγορίθμων μηχανικής μάθησης, εκτός του συνελικτικού δικτύου και του multi-layer perceptron, το re για την υλοποίηση των κανονικών εκφράσεων (regular expressions) που χρησιμοποιούνται στη φάση της προεπεξεργασίας και το pybrain[16] για την υλοποίηση του πολυεπίπεδου perceptron. Επιπλέον χρησιμοποιείται το gensim[17] για τις υλοποιήσεις των μοντέλων word2vec και doc2vec σε cython και το pyenchant[18], ένα απλό και γρήγορο module που παρέχει ένα λεξικό της αγγλικής γλώσσας.

1.5 Οργάνωση Κειμένου

Η οργάνωση του κειμένου γίνεται σε πέντε κεφάλαια με το πρώτο να αποτελεί την εισαγωγή στο αντικείμενο της εργασίας. Τα υπόλοιπα τέσσερα κεφάλαια οργανώνονται ως εξής

- Το κεφάλαιο 2 περιγράφει το σύνολο δεδομένων και την προεπεξεργασία που υφίσταται το σύνολο των tweets πριν γίνει η εφαρμογή των αλγορίθμων μηχανικής μάθησης. Η ενότητα 2.1 δίνει αναλυτικά τα τρία datasets που χρησιμοποιούνται και η 2.2 τα βήματα της προεπεξεργασίας.
- Στο κεφάλαιο 3 περιγράφεται η θεωρία των αλγορίθμων επιβλεπόμενης μηχανικής μάθησης που εξετάζονται από την υλοποίησή μας. Πιο αναλυτικά στην ενότητα 3.1 περιγράφονται οι αλγόριθμοι ταξινόμησης που βασίζονται στη θεωρία αποφάσεων κατά Bayes, στην ενότητα 3.2 ο αλγόριθμος k-Nearest Neighbors, στην 3.3 η γραμμική και η λογιστική παλινδρόμηση, στην 3.4 οι μηχανές διανυσμάτων υποστήριξης, στην 3.5 τα τεχνητά νευρωνικά δίκτυα και τέλος στην ενότητα 3.6 η αρχιτεκτονική του συνελικτικού δικτύου.
- Το κεφάλαιο 4 περιγράφει τη διαδικασία εξαγωγής χαρακτηριστικών από τα δεδομένα κειμένου. Στην ενότητα 4.1 μελετάμε τη μέθοδο Bag-of-Words, στην ενότητα 4.2 τις διανυσματικές αναπαραστάσεις λέξεων ή word vectors και τέλος, στην ενότητα 4.3 τη χρήση των word vectors για την αναπαράσταση κειμένου, δηλαδή συνόλου λέξεων.
- Τέλος, στο κεφάλαιο 5 παρουσιάζεται η υλοποίηση του συστήματος και τα πειραματικά αποτελέσματα, καθώς και σχολιασμός αυτών.

---

[11] https://www.python.org/
[12] http://torch.ch/
[13] http://www.nltk.org/
[14] http://www.numpy.org/
[15] http://scikit-learn.org/stable/
[16] http://pybrain.org/
[17] https://radimrehurek.com/gensim/
[18] http://pythonhosted.org/pyenchant/

2 Δεδομένα και Προεπεξεργασία

Ένα πρόβλημα των εφαρμογών επιβλεπόμενης μηχανικής μάθησης , είναι η εύρεση καλών, ποιοτικά και ποσοτικά, συνόλων δεδομένων που συνοδεύονται από επιθυμητές αποκρίσεις. Τέτοια σύνολα δεδομένων καλούνται labeled datasets και είναι απαραίτητα στην επιβλεπόμενη μάθηση εν αντιθέσει με την μη επιβλεπόμενη μάθηση που χρησιμοποιεί unlabeled δεδομένα. Η προσάρτηση επιθυμητών εξόδων σε δεδομένα απαιτεί στις περισσότερες των περιπτώσεων ανθρώπινη εργασία, δηλαδή ανθρώπους να χαρακτηρίζουν τα δεδομένα ένα προς ένα ως προς το περιεχόμενό τους. Αυτό, όπως είναι φυσικό, θέτει περιορισμούς τόσο στην ποιότητα όσο και στην ποσότητα των δεδομένων. Εξαιτίας του μη αυτοποιημένου τρόπου εξαγωγής των labels (επιθυμητές έξοδοι), labeled datasets με μέγάλο αριθμό δεδομένων είναι δυσεύρετα. Επίσης, πολλές φορές το περιεχόμενο των δεδομένων δεν είναι άμεσα προφανές ούτε στον άνθρωπο, με αποτέλεσμα να υπάρχουν ασυμφωνίες στο χαρακτηρισμό των δεδομένων (labeling of data) και να απαιτούνται περισσότεροι του ενός άνθρωποι (annotators) για αυτό τον χαρακτηρισμό.

Η υπερπληθώρα δεδομένων στο διαδίκτυο που σχολιάσαμε και στην εισαγωγή, συνιστά κυρίως unlabeled δεδομένα, ωστόσο το web2.0 δίνει δυνατότητες για την πλήρη ή μερική αυτοματοποίηση της διαδικασίας χαρακτηρισμού δεδομένων σε ορισμένες περιπτώσεις. Στο πεδίο της ανάλυσης συναισθήματος, για παράδειγμα, όταν εξετάζεται η συναισθηματική πολικότητα σε κριτικές από διαδικτυακές πηγές, η αυτοματοποίηση της διαδικασίας χαρακτηρισμού είναι εύκολη. Καθώς υπάρχουν πολλές διαδικτυακές πηγές, όπου χρήστες ανεβάζουν κριτικές προϊόντων και συνοδεύουν τις κριτικές αυτές από κάποια βαθμολογία (star rating) ο χαρακτηρισμός δεδομένων, επιμερίζεται σε όλους τους χρήστες και όχι σε μερικούς annotators. Έτσι στο διαδίκτυο συναντά κανείς πολλά datasets με κριτικές ταινιών από τις αντίστοιχες πηγές (imdb, rotten tomatoes), κριτικές προϊόντων (amazon), κριτικές ταξιδιωτικών προορισμών (trip advisor) κ.α. Επίσης σε δεδομένα

Twitter είναι δυνατόν να αυτοματοποιηθεί η διαδικασία χαρακτηρισμού της συναισθηματικής πολικότητας των tweets με αυτόματο χαραρακτηρισμό των tweets με θετικά emoticons ως θετικά και των αντίστοιχων με αρνητικά emoticons ως αρνητικά. Μία τέτοια μέθοδος καλείται noisy labeling (ή distant supervision) καθώς στις επιθυμητές εξόδους ενυπάρχει θόρυβος. Tweets με θετικά emoticons δεν είναι απαραίτητα θετικά και αντίστροφα.

Η συνολική συναισθηματική πολικότητα βέβαια ενός συνόλου λέξεων, ειδικά ενός τόσο μικρού όσο ένα tweet είναι πολλές φορές δύσκολο να χαρακτηριστεί απόλυτα. Οι annotators πολλές φορές διαφωνούν σε μικρό ποσοστό ακόμα και στον χαρακτηρισμό πολύ συγκεκριμένου περιεχομένου, όπως αντικείμενα σε εικόνες, συνεπώς σε κάτι πιο αφηρημένο όπως το συναίσθημα σε ένα tweet τα ποσοστά διαφωνίας είναι αρκετά μεγαλύτερα. Από τα παραπάνω προκύπτει ότι η συμφωνία των annotators είναι ένα στοιχείο που πρέπει να λαμβάνεται υπόψιν στην αξιολόγηση ενός sentiment analysis αλγορίθμου. Ποσοστό επιτυχίας 98% σε ένα πρόβλημα αναγνώρισης αντικειμένων, όταν το αντίστοιχο ποσοστό ενός ανθρώπου είναι 99% δείχνει ότι ο αλγόριθμος λειτουργεί ικανοποιητικά. Αντίθετα ποσοστό επιτυχίας 98% σε ένα πρόβλημα ανάλυσης συναισθήματος στο οποίο ο άνθρωπος συμφωνεί κατά 90% με τους χαρακτηρισμούς των annotators δηλώνει ότι ο αλγόριθμος ταξινόμησης λειτουργεί σωστά αλλά το πρόβλημα δεν επιλύεται ικανοποιητικά. Σε κάθε περίπτωση ο χαρακτηρισμός των δεδομένων πρέπει να γίνεται από πολλούς annotators για να εξασφαλίζεται η ποιότητα των δεδομένων και να εκπαιδεύονται συστήματα που συλλαμβάνουν την πραγματική διάσταση του προβλήματος.

Εξαιτίας των παραπάνω περιορισμών αλλά και των περιορισμών που θέτει το Twitter, η δημιουργία καλών συνόλων labeled δεδομένων είναι δύσκολη διαδικασία. Στην ενότητα 2.1 παρουσιάζεται το σύνολο δεδομένων που θα χρησιμοποιήσουμε για την υλοποίησή μας. Τα δεδομένα προέρχονται από τον ετήσιο διαγωνισμό SemEval ([16]), το σύνολο Stanford Twitter Sentiment ([4]) και την παραλλαγή του συνόλου αυτού που καλείται STS-Gold ([22]). Στην ενότητα 2.2 παρουσιάζουμε αναλυτικά τη διαδικασία προεπεξεργασίας των δεδομένων.

2.1 Το Σύνολο Δεδομένων

2.1.1 Τα Δεδομένα του SemEval-2016: Task 4

Ο ετήσιος διαγωνισμός SemEval[19] (International Workshop on Semantic Evaluation) περιλαμβάνει διάφορα προβλήματα επεξεργασίας κειμένου πάνω στα οποία διαγωνίζονται κάθε χρόνο ερευνητικές ομάδες από οργανισμούς, εταιρείες και πανεπιστήμια ανά τον κόσμο. Παρέχει labeled και unlabeled δεδομένα στις ομάδες, πάνω στα διάφορα προβλήματα και οι ομάδες καλούνται να υλοποιήσουν συστήματα προς επίλυση των προβλημάτων. Το 2016 ο διαγωνισμός περιελάμβανε 14 προβλήματα (tasks) με τέσσερα από αυτά, να αφορούν την ανάλυση συναισθήματος. Συγκεκριμένα το task 4 ήταν η ανάλυση

[19] http://alt.qcri.org/semeval2016/

συναισθήματος στο Twitter (Task 4: Sentiment Analysis in Twitter), το task 5 η aspect-based ανάλυση συναισθήματος (Task 5: Aspect-Based Sentiment Analysis), το task 6 η ανάλυση της εκτίμησης/στάσης για κάποιο θέμα σε δεδομένα Twitter (Task 6: Detecting Stance in Tweets) και το task 7 ο προσδιορισμός της έντασης συναισθήματος (sentiment intensity) σε αγγλικές και αραβικές φράσεις (Task 7: Determining Sentiment Intensity of English and Arabic Phrases).

Το πρόβλημα της ανάλυσης συναισθήματος στο Twitter, περιελάμβανε πέντε επιμέρους προβλήματα (subtasks), τα ακόλουθα

- Subtask A: Ταξινόμηση της συνολικής συναισθηματικής πολικότητας σε τρεις κατηγορίες, θετική, αρνητική ή ουδέτερη.
- Subtask B: Ταξινόμηση του συναισθήματος σε δύο κατηγορίες, θετικό ή αρνητικό, όταν κάθε tweet αναφέρεται σε κάποιο δοσμένο θέμα.
- Subtask C: Ταξινόμηση του συναισθήματος σε πέντε κατηγορίες, θετικό, αρνητικό, ουδέτερο, πολύ θετικό ή πολύ αρνητικό όταν κάθε tweet αναφέρεται σε κάποιο δοσμένο θέμα.
- Subtask D: Εκτίμηση της κατανομής σε δύο κλάσεις, θετική και αρνητική, δεδομένου ενός συνόλου από tweets που αναφέρονται σε συγκεκριμένο θέμα.
- Subtask E: Εκτίμηση της κατανομής σε πέντε κλάσεις, δεδομένου ενός συνόλου από tweets που αναφέρονται σε συγκεκριμένο θέμα.

Για τις ανάγκες της παρούσας διπλωματικής εργασίας χρησιμοποιήθηκαν τα ακόλουθα σύνολα δεδομένων που παρέχει το SemEval για την επίλυση του task4[20]

- Τα training, development και development-test δεδομένα του 2016 (SemEval-2016 task 4 train, dev and devtest data).
- Τα training δεδομένα του 2013 (SemEval-2013 task 2 train data).
- Τα development δεδομένα του 2013 (SemEval-2013 task 2 dev data).
- Τα development-test δεδομένα του 2013 (SemEval-2013 task 2 devtest data), μόνο αυτά που περιέχουν tweets και όχι τα SMS.
- Τα development-test δεδομένα του 2014 (SemEval-2014 task 9 devtest data), μόνο αυτά που περιέχουν tweets και όχι τα SMS.

Τα δεδομένα παρέχονται με τη μορφή αρχείων κειμένου που περιέχουν τις μοναδικές ταυτότητες (IDs) των tweets και τα labels για κάθε tweet. Με τη βοήθεια των αρχείων αυτών, των python scripts που παρέχει το SemEval και ενός Twitter account κατεβάσαμε τα δεδομένα απευθείας από το Twitter. Από το σύνολο των tweets αφαιρέθηκαν τα μη διαθέσιμα (Not Available) και τα tweets με χαρακτηρισμό objective ή neutral, δηλαδή κρατήσαμε μόνο τα positive και negative tweets. Τέλος διαγράφησαν tweets που παρουσιάζονταν δύο ή περισσότερες φορές. Τελικά προέκυψε το σύνολο δεδομένων με 12,895 tweets εκ των οποίων τα 9,594 είναι θετικά και τα 3,301 αρνητικά.

Αναλυτικές πληροφορίες για το task 4 του διαγωνισμού SemEval-2016 δίνονται στο [16], όπου παρουσιάζονται και τα αποτελέσματα με τα συστήματα που πέτυχαν την καλύτερη

[20] http://alt.qcri.org/semeval2016/task4/index.php?id=data-and-tools

επίδοση σε κάθε subtask. Επίσης, παραπέμπουμε στη δημοσίευση των Palogiannidi et al. [17] για την υλοποίηση του συστήματος που κέρδισε την πρώτη θέση στο subtask B που αναφέρεται στο ίδιο πρόβλημα (two-point scale sentiment analysis in Twitter) που αντιμετωπίζεται και στην παρούσα εργασία.

Σημειώνεται επίσης ότι τα αποτελέσματα της υλοποίησης μας, που θα παρουσιαστούν στο κεφάλαιο 5 δεν είναι δυνατό να τεθούν σε άμεση σύγκριση με τις επιδόσεις που αναφέρονται στο [16]. Μεταξύ άλλων οι λόγοι είναι οι ακόλουθοι :

- Ο διαγωνισμός SemEval θέτει κανόνες για την ακριβή χρήση του κάθε επιμέρους συνόλου δεδομένων. Αντιθέτως στην παρούσα υλοποίηση, τα tweets συγκεντρώνονται σε ένα σύνολο δεδομένων από το οποίο τυχαία επιλέγονται δεδομένα εκπαίδευσης και αξιολόγησης.
- Το μοντέλο που παρουσιάζεται εδώ, εκπαιδεύεται σε σημαντικά μικρότερο αριθμό tweets καθώς πολλά από τα δεδομένα του SemEval δεν ήταν διαθέσιμα από το Twitter και διαγράφησαν.
- Τα συστήματα που διαγωνίζονται αξιολογούνται σε συγκεκριμένο σύνολο δεδομένων (SemEval-2016 task 4 test data) το οποίο δεν είναι διαθέσιμο παρά μόνο στις ομάδες που διαγωνίζονται.
- Το σύνολο δεδομένων της υλοποίησης μας περιέχει tweets και από άλλα σύνολα δεδομένων όπως θα δούμε στη συνέχεια.

2.1.2 Τα Δεδομένα των STS και STS-Gold

Το σύνολο δεδομένων Stanford Twitter Dataset (STS) διαχωρίζεται σε δύο επιμέρους σύνολα, το STS-test και το STS-train. Το STS-test αποτελείται από 177 αρνητικά, 182 θετικά και 139 ουδέτερα tweets, σύνολο 498, τα οποία είναι όλα manually labeled. Το STS-train περιλαμβάνει περίπου 1.6 εκατομμύρια tweets τα οποία χαρακτηρίζονται ως θετικά ή αρνητικά ανάλογα με την παρουσία των αντίστοιχων emoticons. Μία τέτοια διαδικασία χαρακτηρισμού καλείται noisy labeling και παρά το γεγονός ότι είναι αυτοματοποιημένη, δεν είναι πλήρως αξιόπιστη καθώς tweets με θετικά emoticons είναι δυνατόν να εκφράζουν αρνητικό συναίσθημα και αντίστροφα. Τα STS-test και STS-train δεδομένα είναι διαθέσιμα μέσω του Sentiment140[21].

Το σύνολο δεδομένων STS-Gold προτείνεται από τους Saif et al. [22] και χρησιμοποιεί δεδομένα του αρχικού STS-train συνόλου, που περιέχουν δηλαδή τα emoticons, αλλά σε αντίθεση με το STS-train χαρακτηρίζονται από 3 annotators και όχι με βάση τα emoticons. Το τελικό STS-Gold σύνολο περιέχει εκείνα τα tweets στα οποία οι 3 annotators συμφωνούν για τη συνολική συναισθηματική πολικότητα και έτσι προκύπτουν 1,402 αρνητικά tweets, 632 θετικά και 77 ουδέτερα tweets. Μάλιστα, το STS-Gold dataset εξάγει και entities και εκτός του συνολικού συναισθήματος, περιέχει και sentiment labels ως προς τις οντότητες. Ωστόσο, για το πρόβλημα της παρούσας εργασίας

[21] http://help.sentiment140.com/for-students/

επικεντρωνόμαστε στο συνολικό συναίσθημα και χρησιμοποιούμε το αντίστοιχο dataset με τις ετικέτες συνολικής συναισθηματικής πολικότητας.

Για την κατασκευή του συνόλου δεδομένων, χρησιμοποιούμε τα θετικά και αρνητικά tweets των SemEval, STS-test και STS-Gold. Όλα τα tweets είναι manually labeled και όλα τα επιμέρους datasets ελέγχονται για διπλά tweets. Μετά την αφαίρεση κάποιων διπλών tweets καταλήγουμε στο σύνολο που περιγράφει ο πίνακας 2.1.

Όπως φαίνεται στον πίνακα 2.1 οι δύο κλάσεις στο σύνολό μας είναι ανισοπληθείς και μάλιστα με μεγάλη διαφορά αφού τα θετικά tweets είναι υπερδιπλάσια των αρνητικών. Τέτοια ανισορροπία στις κλάσεις μπορεί να οδηγήσει τους αλγορίθμους μηχανικής μάθησης σε φτωχά αποτελέσματα και γι'αυτό το λόγο συμπεριλαμβάνονται στο τελικό σύνολο δεδομένων αρνητικά tweets από το noisy labeled σύνολο STS-train. Συγκεκριμένα συμπεριλαμβάνονται 5,536 αρνητικά tweets που επιλέγονται τυχαία από τα 1.6 εκατομμύρια του συνόλου. Το τελικό σύνολο λοιπόν προκύπτει, όπως φαίνεται στον πίνακα 2.2.

	#θετικών tweets	#αρνητικών tweets	#tweets
SemEval-2016	9,594	3,301	12,895
STS-test	182	177	359
STS-Gold	632	1,394	2,026
Συνολικά	10,408	4,872	15,280

Πίνακας 2.1

	#θετικών tweets	#αρνητικών tweets	#tweets
SemEval-2016	9,594	3,301	12,895
STS-test	182	177	359
STS-Gold	632	1,394	2,026
STS-train	0	5,536	5,536
Συνολικά	10,408	10,408	20,816
Αφαίρεση διπλών	10,408	10,403	20,811

Πίνακας 2.2

Συνοψίζοντας, το σύνολο δεδομένων περιλαμβάνει θετικά και αρνητικά tweets από τα datasets SemEval-2016: task 4, STS-test, STS-Gold και STS-train. Περίπου το 75% του συνόλου είναι manually labeled και το 25% noisy labeled.

Τέλος, σημειώνεται ότι καθώς το STS-Gold περιλαμβάνει tweets από το STS-train τα οποία είναι στην αρχική τους μορφή, μαζί με τα emoticons, είναι πιθανό στο τελικό σύνολο να υπάρχουν σχεδόν ίδια tweets που διαφέρουν μόνο στο emoticon και διαφεύγουν της αναζήτησης ακριβώς ίδιων tweets. Ωστόσο η πιθανότητα να συμβαίνει αυτό είναι μικρή καθώς το STS-train περιέχει 1.6 εκατομμύρια tweets από τα οποία περίπου 2,000

επιλέγονται στο STS-Gold και περίπου 5,000 επιλέγουμε τυχαία για το σύνολό μας. Σε κάθε περίπτωση για να βεβαιωθούμε για την ποιότητα του συνόλου μας, πραγματοποιούμε αναζήτηση κοντινών tweets με βάση την απόσταση Levenshtein στο σύνολό μας. Η απόσταση Levenshtein μεταξύ δύο strings ορίζεται ως ο ελάχιστος αριθμός τροποποιήσεων ενός χαρακτήρα (αφαίρεση, προσθήκη και αντικατάσταση χαρακτήρα) που απαιτείται για την μετάβαση από το ένα string στο άλλο. Strings με μηδενική απόσταση Levenshtein είναι πανομοιότυπα, ενώ όσο μικρότερη είναι η απόσταση τόσο πιο όμοια είναι τα strings. Η αναζήτησή μας δεν φανέρωσε κάποιο τέτοιο ζήτημα, ωστόσο παρατηρήθηκε η περίπτωση των retweets. Εντοπίστηκαν λίγες περιπτώσεις όπου το αρχικό tweet και ένα retweet του αρχικού περιλαμβάνονται και τα δύο στο σύνολο δεδομένων.

2.2 Προεπεξεργασία

Για την εξαγωγή χαρακτηριστικών και την εφαρμογή αλγορίθμων μηχανικής μάθησης, τα δεδομένα πρέπει να υποστούν κατάλληλη επεξεργασία. Ειδικά στην περίπτωση του Twitter λόγω της ειδικής φύσης των tweets η διαδικασία της προεπεξεργασίας είναι ιδιαίτερα σημαντική. Ας δούμε ενδεικτικά τη μορφή μερικών tweets από το σύνολο δεδομένων

```
Gas by my house hit $3.39!!!! I'm going to Chapel Hill on Sat. :)

Looks like Andy the Android may have had a little too much fun yesterday.
http://t.co/...

@Jen I have studied all day but tomorrow I'm going out with friends! :D Omg
Jennette did?!!!! I'm gonna look! &lt;3

#NowPlaying: BEP, Ricky Martin and KT Tunstall! Great songs to get you
through your Sunday! Hate the rain!! http://t.co/...
```

Στα παραπάνω tweets φαίνονται μερικά από τα ιδιαίτερα tokens που χρίζουν ειδικής μεταχείρισης όπως το emoticon ":)", το url "http://t.co/...", το username "@Jen" και το hashtag "#NowPlaying". Στη συνέχεια περιγράφονται αναλυτικά τα βήματα προεπεξεργασίας που ακολουθούμε στην υλοποίηση μας, με τη σειρά που αυτά πραγματοποιούνται. Η προεπεξεργασία γίνεται με τη βοήθεια κανονικών εκφράσεων (regular expressions) και του re module της python. Αναζητούνται σειρές χαρακτήρων που επαληθεύουν συγκεκριμένες κανονικές εκφράσεις και αντικαθίστανται από strings ή άλλες κανονικές εκφράσεις.

Ηλεκτρονικές διευθύνσεις - URLs

Οι ηλεκτρονικές διευθύνσεις αντικαθίστανται με το string <url>. Για παράδειγμα

```
http://t.co/... → <url>
```

Οι ηλεκτρονικές διευθύνσεις συνήθως ξεκινάνε με τους χαρακτήρες `http://`, `https://` και `www.`.

Ειδικοί χαρακτήρες html

Στο σώμα κειμένου, ειδικά των tweets που προέρχονται από το SemEval περιέχονται διάφοροι ειδικοί χαρακτήρες, κατάλοιπο της html επεξεργασίας που υφίστανται τα tweets. Οι ειδικοί αυτοί χαρακτήρες αντικαθίστανται από τα σύμβολα που αναπαριστούν. Αναλυτικά

- Το string `&` αντικαθίσταται από τον χαρακτήρα `&`. Η διαδικασία εκτελείται δύο φορές.
- Τα strings `<` και `>` αντικαθίστανται από τους χαρακτήρες `<` και `>` αντίστοιχα.
- Το string `&quot;` αντικαθίσταται από τον χαρακτήρα `"`.
- Τα strings ` ` , `<p>` και `</p>` αντικαθίστανται από κενά.

Usernames

Τα usernames, που δηλώνονται στο Twitter από strings που αρχίζουν με το σύμβολο `@` και περιλαμβάνουν γράμματα, αριθμούς ή underscores, αντικαθίστανται απλά από τον όρο `<user>` καθώς δεν περιέχουν σημαντική πληροφορία που μπορεί να ωφελήσει την ανάλυση συναισθήματος. Για παράδειγμα `@Jen` → `<user>`.

Επαναλήψεις σημείων στίξης

Τα σημεία στίξης `.`, `!` και `?` όταν εμφανίζονται πάνω από μία φορά αντικαθίστανται από το αντίστοιχο σημείο στίξης και τον όρο `<repeat>`. Δηλαδή `!!!!` → `! <repeat>`.

Hashtags

Τα hashtags διακρίνονται από τη χρήση του χαρακτήρα `#`. Αντικαθίστανται από τον όρο `<hashtag>` και το σώμα του hashtag το οποίο διαχωρίζεται σε επιμέρους λέξεις. Στην περίπτωση που οι επιμέρους λέξεις διαχωρίζονται με κεφαλαία γράμματα, ο διαχωρισμός σε tokens γίνεται με τη βοήθεια κανονικών εκφράσεων. Σε αντίθετη περίπτωση ο διαχωρισμός γίνεται με αναδρομική αναζήτηση σε λεξικό. Για παράδειγμα

```
#NowPlaying → <hashtag> Now Playing
#Truth → <hashtag> Truth
#love → <hashtag> love
#soundsgood → <hashtag> sounds good
```

Μία εναλλακτική προσέγγιση για τον διαχωρισμό του σώματος του hashtag σε επιμέρους λέξεις που ενδεχομένως να έχει καλύτερα αποτελέσματα μπορεί να υλοποιηθεί με τη βοήθεια του αλγορίθμου Viterbi.

Emoticons

Η επιτυχής αναγνώριση των emoticons είναι μία απαιτητική εργασία εξαιτίας του θορύβου που ενυπάρχει στον τρόπο γραφής τους σε πραγματικά tweets. Ο τρόπος παρουσίασης ενός emoticon δεν είναι μοναδικός καθώς στην πράξη εμφανίζονται πολλές παραλλαγές που περιλαμβάνουν κενά, μικρά και κεφαλαία γράμματα, επιπλέον σημεία στίξης, επαναλήψεις γραμμάτων κ.α. Για παράδειγμα το emoticon :D που συναντάται συχνά μπορεί να εμφανιστεί με διάφορες μορφές όπως :d, : D, :ddd ή :DDD. Ο προσδιορισμός κανονικών εκφράσεων που μπορούν να συλλάβουν όλους τους δυνατούς τρόπους εμφάνισης ενός emoticon είναι δύσκολος, ενώ παράλληλα υπάρχει και το πρόβλημα των περιπτώσεων string που ταιριάζουν σε emoticon αλλά στην πράξη δεν είναι. Το ζήτημα αυτό προκύπτει στην προσπάθεια αναγνώρισης των :p, :P, :d και :D emoticons. Καθώς μετά την άνω κάτω τελεία υπάρχει γράμμα, είναι δυνατό τέτοιες σειρές χαρακτήρων να αντιστοιχούν σε απλό κείμενο, όπως για παράδειγμα στο περίπτωση ...:Don't... . Καθώς ο εντοπισμός των emoticons είναι πολύ σημαντικός για το πρόβλημα της ανάλυσης συναισθήματος, δώθηκε ιδιαίτερη βαρύτητα στην ανάπτυξη κανονικών εκφράσεων που αναγνωρίζουν αποτελεσματικά τα emoticons στο σώμα κειμένου. Οι διάφορες κανονικές εκφράσεις προέκυψαν μετά από πειραματισμό καθώς οι απλές κανονικές εκφράσεις για τον εντοπισμό των emoticons δεν λειτουργούσαν ικανοποιητικά στα δεδομένα μας.

Στη συνέχεια δίνεται μία αναλυτική λίστα με τα emoticons που αναγνωρίζει η υλοποίησή μας και τους όρους με τους οποίους τα αντικαθιστά.

Emoticon	Παραλλαγές	Αντικατάσταση με
<3	<33,<333,…	<heart>
</3	</33,</333,…	<broken_heart>
(y)	(yy),(yyy),…	<thumbs_up>
\m/	\mm/,\M/,…	<metal_salute>
:*	:-*,:-**,:**,…	<kiss>
o.O	O.o	<confused>
^_^	^__^,^___^,…	<smile>
-_-	-__-,-___-,…	<neutralface>
xD	xDD,XD,xd,…	<smile>
8-)	8-},8-],…	<smile>
B-)	B-},B-],b-),…	<cool>
:-)	-	<smile>
:-(	-	<sadface>
:P	-	<lolface>
:-|	-	<neutralface>

Πίνακας 2.3

Οι παραλλαγές των τεσσάρων τελευταίων περιπτώσεων είναι πάρα πολλές και διαφορετικές για να συμπεριληφθούν, έστω και ενδεικτικά, στην παραπάνω λίστα. Εναλλακτικά, μπορούμε να παρουσιάσουμε το σύνολο των emoticons που επιστρέφει η υλοποίησή μας από όλο το σύνολο δεδομένων. Στον πίνακα 2.4 φαίνονται όλα τα emoticons και το πλήθος εμφάνισής τους, που αναγνωρίζει το σύνολο των κανονικών εκφράσεων στο σύνολο των δεδομένων.

Emoticon	**Πλήθος**	**Emoticon**	**Πλήθος**	**Emoticon**	**Πλήθος**
:)	434	<33	4	:'''''(	1
<3	136	:\	4	)::::	1
:D	119	\m/	4	;)	1
;)	114	O.o	4	:-P	1
:(	103	=P	3	;))	1
:-)	57	=))	3	;P	1
:/	41	:)))	3	:'''''''')	1
(:	38	):	3	-_____-	1
:'(	22	(=	2	(((:	1
xD	21	=/	2	((:	1
:p	21	<333	2	:-D)	1
:))	21	:-))	2	;/	1
:P	16	:-\	2	8-)	1
;-)	16	:****	2	=)))	1
XD	15	o.O	2	:***	1
:')	14	:-p	2	>:)	1
:|	12	:(((((	2	>:(	1
=)	11	<3333	2	>:D	1
(;	10	/:	2	;}	1
-_-	9	xd	1	>:[	1
;D	9	=|	1	[;	1
:]	9	:(((((((((	1	>:|	1
:*	8	xDDD	1	:(((	1
=D	8	^___^	1	-____-	1
</3	7	;-))	1	-___-	1
=]	6	=p	1	=))))))	1
-__-	6	:DDD	1	:((	1
:-D	6	:((((	1	))):	1
^_^	6	|=	1	XDDD	1
:-(	6	<333333	1	:`(	1
:-/	5	xDD	1	:))))	1
:)	5	:///	1		

Πίνακας 2.4

Retweets - RTs

Τα string RT που δηλώνουν κοινοποίηση tweet άλλου χρήστη αντικαθίστανται με τον όρο <retweet>.

Αριθμοί

Αριθμοί, ημερομηνίες και ειδικά string που περιέχουν αριθμούς όπως ώρα και τιμή, αντικαθίστανται από το token <number>. Για παράδειγμα $3.39 → $ <number>.

Λέξεις γραμμένες αποκλειστικά με κεφαλαία - All Caps words

Οι λέξεις που περιέχουν μόνο κεφαλαία γράμματα και έχουν μήκος μεγαλύτερο ή ίσο των τριών χαρακτήρων αντικαθίστανται από τη λέξη στη lowercased εκδοχή της και τον όρο <allcaps>. Για παράδειγμα USA → usa <allcaps> και YEEEES → yeeees <allcaps>.

Σε αυτό το σημείο όλοι οι χαρακτήρες του σώματος κειμένου μετατρέπονται στους αντίστοιχους πεζούς.

Επιμηκυμένες λέξεις - Elongated words

Επιμηκυμένες χαρακτηρίζονται οι λέξεις στις οποίες κάποιοι χαρακτήρες επαναλαμβάνονται προς απόδοση έμφασης στο νόημα της λέξης. Η αναγνώριση επιμηκυμένων λέξεων έχει ιδιαίτερη σημασία αφενός γιατί συναντώνται πολύ συχνά σε δεδομένα κοινωνικών δικτύων και αφετέρου γιατί δίνουν χρήσιμη πληροφορία τουλάχιστον όσον αφορά το πρόβλημα της ανάλυσης συναισθήματος. Ο αποτελεσματικός εντοπισμός επιμηκυμένων λέξεων είναι δύσκολη εργασία όπως επίσης και η αντιστοίχισή τους σε πραγματικές λέξεις. Η διαδικασία που ακολουθούμε για τον χειρισμό επιμηκυμένων λέξεων συνοψίζεται στα ακόλουθα βήματα:

- Αρχικά το σώμα κειμένου, αφού έχει υποστεί την προεπεξεργασία μέχρι αυτό το σημείο, διαχωρίζεται σε επιμέρους tweets (κάθε γραμμή στο txt αρχείο αντιστοιχεί σε ένα tweet) και κάθε tweet διαχωρίζεται στα επιμέρους tokens. Το tokenization γίνεται με τη βοήθεια του TweetTokenizer του nltk που συμπεριφέρεται καλύτερα σε tweets από την απλή συνάρτηση word_tokenize του ίδιου module. Ωστόσο σε αυτό το σημείο της προεπεξεργασίας, και ένας απλός διαχωρισμός με βάση τα κενά, θα διαχώριζε επιτυχώς τις λέξεις σε κάθε tweet.

- Στη συνέχεια, υλοποιείται μία κανονική έκφραση που αναζητεί τρεις ή περισσότερες διαδοχικές εμφανίσεις του ίδιου γραμματικού χαρακτήρα σε strings και κάθε token του σώματος των tweets ελέγχεται για το αν επαληθεύει την κανονική έκφραση ή όχι. Τα tokens που επαληθεύουν την έκφραση θεωρούνται επιμηκυμένες λέξεις αφού στην πράξη δεν παρατηρούνται τρεις ή περισσότερες διαδοχικές εμφανίσεις του ίδιου χαρακτήρα σε λέξεις της αγγλικής γλώσσας.

- Οι επιμηκυμένες λέξεις αποθηκεύονται σε μία λίστα και για κάθε μία από αυτές ακολουθείται η εξής διαδικασία:

    - Οι χαρακτήρες που εμφανίζονται τρεις ή περισσότερες φορές αντικαθίστανται από τον αντίστοιχο χαρακτήρα δύο φορές. Ελέγχεται αν η λέξη είναι λέξη της αγγλικής.
    - Εάν όχι, οι διπλοί χαρακτήρες αντικαθίστανται από μία εμφάνιση του αντίστοιχου χαρακτήρα. Ελέγχεται πάλι αν η λέξη είναι λέξη της αγγλικής γλώσσας.
    - Εάν όχι, ελέγχονται όλες οι παραλλαγές της λέξης όπου μόνο ένας χαρακτήρας από αυτούς που εμφανίζονται πολλές φορές στην αρχική λέξη, εμφανίζεται δύο φορές.
    - Εάν ο έλεγχος επιστρέψει μία πραγματική λέξη η διαδικασία τερματίζει, διαφορετικά η επιμηκυμένη λέξη αντιστοιχίζεται στην παραλλαγή όπου όλοι οι χαρακτήρες εμφανίζονται μόνο μία φορά.

- Αφού κάθε επιμηκυμένη λέξη αντιστοιχηθεί σε μία νέα λέξη που μπορεί να είναι ή όχι πραγματική λέξη της αγγλικής γλώσσας, όλες οι εμφανίσεις της επιμηκυμένης λέξης στο σώμα κειμένου, αντικαθίστανται από τη νέα λέξη και το token <elong>.

Για παράδειγμα η λέξη soooo, σύμφωνα με τα παραπάνω υφίσταται τις εξής μετατροπές

soooo → soo → so → soo → so

με τη διαδικασία να σταματά στο τρίτο βήμα. Τελικά αντικαθίσταται ως εξής

soooo → so <elong>

Αντίστοιχα η λέξη yeeeeaaah

yeeeeaaah → yeeaah → yeah → yeeah → yeaah →yeah

yeeeeaaah → yeah <elong>

Οι έλεγχοι γίνονται με τη βοήθεια του pyenchant module της python που παρέχει μία γρήγορη υλοποίηση αγγλικού λεξικού.

Με τον χειρισμό των επιμηκυμένων λέξεων, ολοκληρώνεται η διαδικασία προεπεξεργασίας των tweets. Η προεπεξεργασία σε συνδυασμό με τον TweetTokenizer του nltk παρουσιάζει καλά αποτελέσματα στον διαχωρισμό των tweets. Σημειώνεται ότι οι όροι που επιλέχτηκαν να συνοδεύουν ιδιαίτερα tokens των tweets όπως τα <user>, <hashtag> και <elong>, δεν επιλέχτηκαν τυχαία, αλλά είναι σε συμφωνία με την προεπεξεργασία του μοντέλου GloVe στο Twitter, καθώς στην πορεία της υλοποίησης θα χρησιμοποιηθούν προ-εκπαιδευμένα word vectors του μοντέλου αυτού. Το μοντέλο GloVe δίνεται αναλυτικά στο κεφάλαιο 4.

Ενδεικτικά δίνεται η μορφή των tweets που παρουσιάστηκαν στην αρχή της ενότητας, μετά τη διαδικασία προεπεξεργασίας.

gas by my house hit $ <number> ! <repeat> i'm going to chapel hill on sat. <smile>

looks like andy the android may have had a little too much fun yesterday. <url>

<user> i have studied all day but tomorrow i'm going out with friends! <smile> omg jennette did? ! <repeat> i'm gonna look! <heart>

<hashtag> now playing : bep, ricky martin and kt tunstall! great songs to get you through your sunday! Hate the rain ! <repeat> <url>

3 Αλγόριθμοι Μηχανικής Μάθησης

Σε αυτό το κεφάλαιο θα επιχειρηθεί μία σύντομη, αλλά περιεκτική κατά το δυνατόν, θεωρητική εισαγωγή στους αλγορίθμους μηχανικής μάθησης που θα χρησιμοποιηθούν στη συνέχεια. Οι αλγόριθμοι αυτοί είναι αλγόριθμοι επιβλεπόμενης μάθησης και επιλύουν ουσιαστικά ένα πρόβλημα ταξινόμησης.

Ας δούμε όμως αρχικά το γενικό πλαίσιο του προβλήματος της ταξινόμησης.

Έστω σημεία ενός χώρου διάστασης n

$$\mathbf{x} = (x_1, x_2, \dots, x_n)$$

Τα $x_1, x_2 \dots$ ονομάζονται *χαρακτηριστικά* (*features*) και το $\mathbf{x}$ *διάνυσμα χαρακτηριστικών* (*feature vector*). Το σύνολο δεδομένων εκπαίδευσης αποτελείται από πολλά τέτοια διανύσματα στον n-διάστατο *χώρο χαρακτηριστικών* (*feature space*), κάθε ένα από τα οποία ανήκει σε μία από τις k κλάσεις

$$C_1, C_2, \dots, C_k \in C$$

όπου C το σύνολο των κλάσεων $\{C_1, C_2, \dots, C_k\}$.

Η κλάση κάθε σημείου είναι γνωστή και στόχος είναι η πρόβλεψη της κλάσης νέων σημείων.

Κάποιοι αλγόριθμοι όπως οι μηχανές διανυσμάτων υποστήριξης, αντιμετωπίζουν το πρόβλημα γεωμετρικά και αναζητούν διαμερίσεις του χώρου χαρακτηριστικών σε διαστήματα, ώστε σημεία του ίδιου διαστήματος να ανήκουν στην ίδια κλάση. Άλλοι αλγόριθμοι όπως οι *Μπεϋζιανοί ταξινομητές* (*Bayesian classifiers*) προσεγγίζουν το πρόβλημα στατιστικά όπως θα φανεί στη συνέχεια.

3.1 Αλγόριθμοι Ταξινόμησης Bayes

Οι αλγόριθμοι ταξινόμησης Bayes είναι ταξινομητές που βασίζονται στη *θεωρία αποφάσεων κατά Bayes* (*Bayesian Decision Theory*). Έστω ένα νέο διάνυσμα

$$\mathbf{x} = (x_1, x_2, \dots, x_n)$$

του οποίου η κλάση-κατηγορία αναζητείται. Ο Μπεϋζιανός ταξινομητής εν γένει υπολογίζει για το νέο αυτό διάνυσμα, τις πιθανότητες να ανήκει σε κάθε μία από τις κατηγορίες και το ταξινομεί τελικά στην κλάση-κατηγορία για την οποία η πιθανότητα είναι η μεγαλύτερη. Η πιθανότητα κάποιο διάνυσμα $\mathbf{x}$ να ανήκει στην κλάση $C_i\, i = 1,2, \dots, k$ δηλώνεται ως η δεσμευμένη πιθανότητα

$$P(C_i \mid \mathbf{x}) = P(C_i \mid x_1, x_2, \dots, x_n)$$

οπότε η απόφαση του ταξινομητή $\hat{y}$ για το διάνυσμα $\mathbf{x}$ είναι

$$\hat{y} = argmax_{\,i \in \{1,2,\dots,k\}} P(C_i \mid \mathbf{x})$$

Η προσέγγιση μέχρι στιγμής είναι γενική για τους στατιστικούς ταξινομητές, ωστόσο οι Μπεϋζιανοί κάνουν χρήση του θεωρήματος του Bayes για να υπολογίσουν αυτή ακριβώς την πιθανότητα. Το θεώρημα αυτό συνδέει την *posterior* πιθανότητα $P(A|B)$, του ενδεχομένου A δεδομένου του ενδεχομένου B, με την *prior* πιθανότητα $P(A)$. Συγκεκριμένα

$$posterior = \frac{prior \cdot likelihood}{evidence}$$

ή με όρους πιθανοτήτων

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)}$$

- $posterior$: $P(A|B)$ η πιθανότητα να συμβεί το ενδεχόμενο A δεδομένου ότι συμβαίνει το B.
- $prior$: $P(A)$ η πιθανότητα να συμβεί το A χωρίς προϋπόθεση.
- $evidence$: $P(B)$ η πιθανότητα να συμβεί το B χωρίς προϋπόθεση.
- $likelihood$: $P(B|A)$ η πιθανότητα να συμβεί το ενδεχόμενο B δεδομένου ότι συμβαίνει το A.

Με εφαρμογή αυτής της σχέσης στις παραπάνω προκύπτει

$$P(C_i|\mathbf{x}) = \frac{P(C_i) \cdot P(\mathbf{x}|C_i)}{P(\mathbf{x})}$$

$$\hat{y} = argmax_{i\in\{1,2,\dots,k\}} \frac{P(C_i) \cdot P(\mathbf{x}|C_i)}{P(\mathbf{x})}$$

Ο παρονομαστής της παραπάνω σχέσης μπορεί να απαλειφθεί αφού είναι ανεξάρτητος του i , είναι δηλαδή κοινός σε όλες τις πιθανότητες (για όλες τις κλάσεις). Η απόφαση του ταξινόμητη λοιπόν γίνεται

$$\hat{y} = argmax_{i\in\{1,2,\dots,k\}} P(C_i) \cdot P(\mathbf{x}|C_i)$$

όπου $P(C_i) \cdot P(\mathbf{x}|C_i) = P(C_i,\ \mathbf{x}) = P(C_i, x_1, x_2, \dots, x_n) = P(x_1, x_2, \dots, x_n, C_i)$.

Ο ταξινομητής λοιπόν είναι σε θέση να πάρει μία απόφαση για την κατηγορία στην οποία ανήκει το άγνωστο δείγμα $\mathbf{x}$ αλλά χρειάζεται τις πιθανότητες $P(C_i)$ και $P(\mathbf{x}|C_i)$ ή ορθότερα τις ***a priori*** πιθανότητες $P(C_i)$ για κάθε κλάση και τις ***συναρτήσεις πυκνότητας πιθανότητας*** (***probability density functions***) $P(\mathbf{x}|C_i)$. Οι παράμετροι αυτές είναι χαρακτηριστικές του μοντέλου και εκτιμούνται από τα δεδομένα εκπαίδευσης. Μάλιστα υπάρχουν διαφορετικοί τρόποι εκτίμησης αυτών των παραμέτρων οπότε προκύπτουν και διαφορετικοί ταξινομητές. Προτού όμως εξεταστούν αναλυτικά πρέπει να σημειωθεί ότι η σχετικά ακριβής εκτίμηση των συναρτήσεων πυκνότητας πιθανότητας είναι ακριβή υπολογιστικά διαδικασία και για αυτό το λόγο η ευρέως υιοθετημένη προσέγγιση κάνει σημαντικές παραχωρήσεις. Προκύπτει έτσι ο ***Απλοϊκός Ταξινομητής Bayes*** (***Naive Bayes Classifier***) ο οποίος υποθέτει ότι τα επιμέρους χαρακτηριστικά $x_1, x_2, \dots, x_n$ είναι στατιστικώς ανεξάρτητα. Με βάση αυτή την παραδοχή η από κοινού πιθανότητα γίνεται

$$\begin{aligned}
P(C_i) \cdot P(\mathbf{x}|C_i) &= P(x_1, x_2, \dots, x_n, C_i) = P(x_1|x_2, \dots, x_n, C_i) \cdot P(x_2, \dots, x_n, C_i) \\
&= P(x_1|x_2, \dots, x_n, C_i) \cdot P(x_2|x_3, \dots, x_n, C_i) \cdot P(x_3, \dots, x_n, C_i) \\
&= P(x_1|x_2, \dots, x_n, C_i) \cdot P(x_2|x_3, \dots, x_n, C_i) \cdot P(x_3|x_4, \dots, x_n, C_i) \cdot P(x_4, \dots, x_n, C_i) \\
&= \cdots \\
&= P(x_1|x_2, \dots, x_n, C_i) \cdot P(x_2|x_3, \dots, x_n, C_i) \cdot \dots \cdot P(x_{n-1}|x_n, C_i) \cdot P(x_n|C_i) \cdot P(C_i) \\
&= P(x_1|C_i) \cdot P(x_2|C_i) \cdot \dots \cdot P(x_n|C_i) \cdot P(C_i)
\end{aligned}$$

όπου στην τελευταία γραμμή έγινε η παραδοχή της ανεξαρτησίας

$$P\left(x_j \middle| x_{j+1}, \dots, x_n, C_i\right) = P(x_j|C_i)$$

Συνεπώς η απόφαση του απλοϊκού ταξινομητή Bayes για το άγνωστο δείγμα $\mathbf{x}$ είναι

$$\hat{y} = argmax_{i\in\{1,2,\dots,k\}} P(C_i) \cdot \prod_{j=1}^{n} P(x_j|C_i)$$

Ο απλοϊκός ταξινομητής Bayes, παρά τις παραδοχές ανεξαρτησίας είναι ένας εξαιρετικά δημοφιλής ταξινομητής και σημειώνει καλές επιδόσεις σε πολλά προβλήματα του πραγματικού κόσμου. Επίσης εκπαιδεύεται αποδοτικά εξαιτίας της απλότητάς του. Η εκπαίδευσή του, όπως προαναφέρθηκε, συνιστά ουσιατικά τη στατιστική εκτίμηση των συναρτήσεων πυκνότητας πιθανότητας $P(\mathbf{x}\,|C_i)$ δηλαδή των $P(x_j|C_i)$ για $j = 1,2,\dots,n$ και $i = 1,2,\dots,k$ μετά την θεώρηση της ανεξαρτησίας των χαρακτηριστικών. Η προσέγγιση αυτή βασίζεται στη θεώρηση κάποιας τυπικής κατανομής για τα δεδομένα και την εκτίμηση των παραμέτρων της κατανομής με τη μέθοδο *μέγιστης πιθανοφάνειας* (*maximum likelihood - ML*). Σχετικά με τις prior πιθανότητες $P(C_i)$, οι κλάσεις μπορούν να θεωρηθούν ισοπίθανες δηλαδή

$$P(C_i) = \frac{1}{\textit{αριθμός κλάσεων}} = \frac{1}{k}\ \textit{για κάθε}\ i = 1,2,\dots,k$$

ή να εκτιμηθούν με βάση το πλήθος των δεδομένων σε κάθε κλάση

$$P(C_i) = \frac{\textit{πλήθος δεδομένων στην κλάση}\ C_i}{\textit{συνολικό πλήθος δεδομένων}}$$

Gaussian Naive Bayes Classifier

Σε αυτή την περίπτωση τα δεδομένα θεωρείται ότι ακολουθούν κανονική κατανομή. Για κάθε χαρακτηριστικό x_j, $j = 1,2,\dots,n$ και κάθε κλάση C_i, $i = 1,2,\dots,k$ η συνάρτηση πυκνότητας πιθανότητας θεωρείται Gaussian, δηλαδή

$$P(x_j = x|C_i) = \frac{1}{\sqrt{2\pi\sigma_{ji}^2}}\exp(-\frac{(x-\mu_{ji})^2}{2\sigma_{ji}^2})$$

Οι παράμετροι μέσης τιμής μ_{ji} και διασποράς σ_{ji}^2 για κάθε χαρακτηριστικό και κάθε κλάση εκτιμούνται με τη μέθοδο μέγιστης πιθανοφάνειας και ορίζουν επαρκώς τις συναρτήσεις πυκνότητας πιθανότητας.

Multinomial Naive Bayes Classifier

Οι multinomial ταξινομητές Bayes σε αντίθεση με τους Gaussian δεν εφαρμόζονται σε συνεχή δεδομένα όπως ήταν η μέχρι τώρα παραδοχή αλλά είναι δημοφιλείς σε προβλήματα ταξινόμησης κειμένου και ειδικότερα στην τεχνική *Bag-of-Words* που είναι ένας κλασσικός τρόπος εξαγωγής χαρακτηριστικών από δεδομένα κειμένου. Στην προσέγγιση αυτή τα διανύσματα $\mathbf{x} = (x_1, x_2, \dots, x_n)$ είναι ιστογράμματα όπου κάθε χαρακτηριστικό εκφράζει συχνότητα εμφάνισης ενός ενδεχομένου. Για παράδειγμα το δείγμα

$$a = (1,5,4,0,2)$$

σε ένα τέτοιο πρόβλημα εκφράζει εμφάνιση του ενδεχομένου x_1 μία φορά, του x_2 πέντε φορές κ.ο.κ. Στην Bag-of-Words προσέγγιση τα ενδεχόμενα είναι λέξεις. Λεπτομερέστερη περιγραφή θα δωθεί στο κεφάλαιο 4, ωστόσο προς το παρόν σημειώνεται ότι τα χαρακτηριστικά είναι ακέραιοι αριθμοί και απεικονίζουν συχνότητα εμφάνισης ενδεχομένων. Η συνάρτηση πυκνότητας πιθανότητας είναι

$$P(\mathbf{x}|C_i) = \frac{(\sum_j x_j)!}{\prod_j x_j!} \prod_j p_{ij}^{x_j}$$

όπου p_{ij} είναι οι παράμετροι προς εκτίμηση.

Bernoulli Naive Bayes Classifier

Όπως και η προηγούμενη κατηγορία ταξινομητών έτσι και οι Bernoulli είναι δημοφιλείς για προβλήματα ταξινόμησης κειμένων. Θεωρούν οτι τα δεδομένα ακολουθούν κατανομή Bernoulli και συνεπώς χειρίζονται δυαδικά χαρακτηριστικά, δηλαδή κάθε χαρακτηριστικό x_j μπορεί να πάρει την τιμή 0 ή 1. Τα διανύσματα λοιπόν στο χώρο χαρακτηριστικών είναι ακολουθίες από 0 και 1, με το 0 σε κάποια θέση x_j να υποδεικνύει ότι το ενδεχόμενο j δεν συμβαίνει στο δείγμα $\mathbf{x}$ και το 1 το αντίθετο. Στην Bag-of-Words προσέγγιση, όπου τα ενδεχόμενα είναι στην ουσία η παρουσία λέξεων σε ένα κείμενο, αυτή η τεχνική χαρακτηρίζεται *term occurrence* εν αντιθέσει με αυτή που περιγράψαμε στον προηγούμενο ταξινομητή που χαρακτηρίζεται *term frequency*.

Στην περίπτωση λοιπόν που τα χαρακτηριστικά είναι δυαδικά μπορεί να εφαρμοστεί ο απλοϊκός ταξινομητής Bayes με Bernoulli εκτίμηση της συνάρτησης πυκνότητας πιθανότητας

$$P(\mathbf{x}|C_i) = \prod_{j=1}^{n} p_{ij}^{x_j} \cdot (1 - p_{ij})^{1-x_j}$$

όπου p_{ij} είναι η πιθανότητα η κλάση C_i να παράξει το ενδεχόμενο i.

3.2 Ο Αλγόριθμος k –Nearest Neighbors

Στην προηγούμενη ενότητα, είδαμε πώς οι Bayesian ταξινομητές αποφασίζουν για την κλάση ενός νέου δείγματος στο χώρο χαρακτηριστικών, με τον προσδιορισμό και τη μεγιστοποίηση της ***a priori*** πιθανότητας $P(C_i|\mathbf{x})$. Σε αυτό το σημείο θα εξεταστεί ένας αρκετά διαφορετικός αλγόριθμος ταξινόμησης, ο οποίος βασίζει την απόφαση του σε τοπολογικά κριτήρια, αντί πιθανοτικών. Ο αλγόριθμος ***k-Nearest Neighbors*** αποτελεί έναν από τους πιο απλούς αλγορίθμους μηχανικής μάθησης και βασίζεται στην έννοια της εγγύτητας. Ταξινομεί σημεία του χώρου χαρακτηριστικών στην κλάση που είναι η πιο κοινή μεταξύ των k πλησιέστερων σημείων εκπαίδευσης.

Ο Κανόνας του Πλησιέστερου Γείτονα – Nearest Neighbor Rule

Δεδομένων Ν διανυσμάτων εκπαίδευσης $\mathbf{x}_i$ $i = 1,2,\dots,N$ στο χώρο χαρακτηριστικών, κάθε νέο σημείο – δείγμα καταχωρείται στην κλάση του πλέον κοντινού σημείου εκπαίδευσης, με βάση κάποιο μέτρο απόστασης.

Ο αλγόριθμος k-Nearest Neighbors

Δεδομένων Ν διανυσμάτων εκπαίδευσης $\mathbf{x}_i$ $i = 1,2,\dots,N$ στο χώρο χαρακτηριστικών, κάθε νέο σημείο – δείγμα, καταχωρείται στην κλάση που πλειοψηφεί μεταξύ των k κοντινότερων στο δείγμα, με βάση κάποιο μέτρο απόστασης, σημείων εκπαίδευσης.

Ο αλγόριθμος ***k-Nearest Neighbors*** είναι η γενίκευση του *κανόνα του πλησιέστερου γείτονα* στα k πλησιέστερα σημεία στο σημείο δείγμα. Η παράμετρος k καθορίζεται από τον χρήστη και επιλέγεται συνήθως μέσω πειραμάτων (***hyperparameter optimization***) αφού η επίδοση διαφορετικών k εξαρτάται από την γενικότερη διάταξη των σημείων εκπαίδευσης και δεν απαντά σε συγκεκριμένους κανόνες. Εν γένει μεγαλύτερες τιμές k μπορούν να βελτιώσουν το αποτέλεσμα της ταξινόμησης αλλά και να συμπεριλάβουν στα σημεία-ψηφοφόρους απομακρυσμένα δείγματα (***outliers***) με αρνητικές συνέπειες. Ωστόσο στην περίπτωση δύο κλάσεων αποφεύγονται άρτιες τιμές k για να μην υπάρχει ισοπαλία στην ψήφο της κλάσης. Με την ίδια λογική, στη γενική περίπτωση K κλάσεων αποφεύγονται πολλαπλάσια του K δηλαδή $k = K, k = 2K, k = 3K \dots$

Ως μετρική απόστασης μπορεί να χρησιμοποιηθεί οποιαδήποτε μαθηματικά θεμελιωμένη απόσταση σημείων σε n-διάστατο χώρο. Οι συνηθέστερες επιλογές είναι η *Ευκλείδεια απόσταση* (***Euclidean Distance***), η *απόσταση Minkowski* (***Minkowski Distance***), η *απόσταση Mahalanobis* (***Mahalanobis Distance***) και η *απόσταση Hamming* (***Hamming Distance***), που μετρά αποστάσεις ανάμεσα σε strings, σε εφαρμογές επεξεργασίας και ταξινόμησης κειμένου.

Ευκλείδια απόσταση

Η ευκλείδια απόσταση μεταξύ δύο σημείων $\mathbf{p}$ και $\mathbf{q}$ στο n-διάστατο χώρο με $\mathbf{p} = (p_1, p_2, \dots, p_n)$ και $\mathbf{q} = (q_1, q_2, \dots, q_n)$ ορίζεται ως

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

Minkowski απόσταση

$$d(\mathbf{p}, \mathbf{q}) = \left(\sum_{i=1}^{n} |p_i - q_i|^{\rho} \right)^{\frac{1}{\rho}}$$

Για $\rho = 2$ η απόσταση Minkowski ταυτίζεται με την ευκλείδια.

Mahalanobis απόσταση

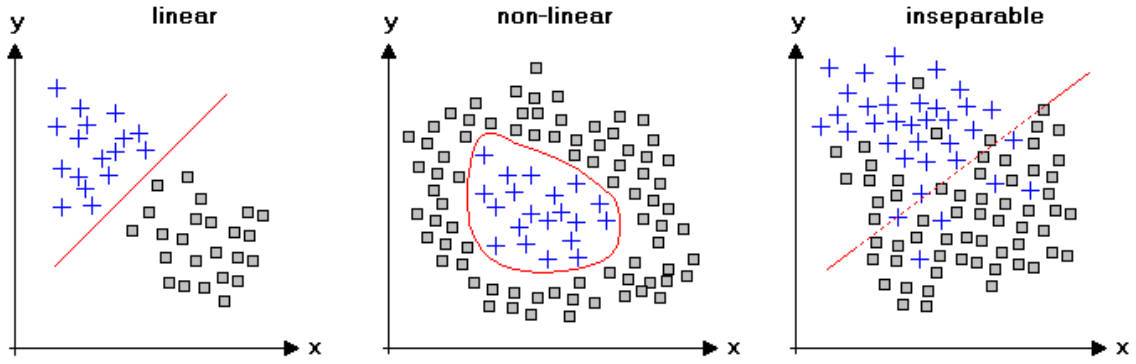
Η απόσταση Mahalanobis μεταξύ σημείου $\mathbf{p} = (p_1, p_2, \dots, p_n)^T$ και κατανομής με μέση τιμή $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_n)^T$ και μήτρα διασποράς $\mathbf{S}$ ορίζεται ως

$$D_M(\mathbf{p}) = \sqrt{(\mathbf{p} - \boldsymbol{\mu})^T \mathbf{S} (\mathbf{p} - \boldsymbol{\mu})}$$

Ο αλγόριθμος του πλησιέστερου γείτονα είναι πολύ απλός στη σύλληψη και την υλοποίησή του, αλλά υπολογιστικά ακριβός για μεγάλα σύνολα δεδομένων εκπαίδευσης. Για κάθε νέο σημείο απαιτείται υπολογισμός των αποστάσεων του από ολά τα σημεία εκπαίδευσης. Ωστόσο είναι εύκολα παραλληλοποιήσιμος και για το λόγο αυτό σύνήθως η υλοποίησή του είναι αρκετά γρήγορη. Επίσης διαφέρει από τους υπόλοιπους αλγορίθμους που θα εξετάσουμε καθώς στην ουσία δεν περιλαμβάνει φάση εκπαίδευσης. Τέλος επηρεάζεται αρνητικά από κλάσεις δεδομένων που δεν είναι ισορροπημένες. Όταν μία κλάση περιλαμβάνει πολύ περισσότερα δεδομένα από τις υπόλοιπες είναι πιθανότερο να υπερισχύσει σε μία διαδικασία ψήφου.

Στις προηγούμενες ενότητες είδαμε δύο διαφορετικούς τρόπους προσέγγισης του προβλήματος της ταξινόμησης. Ο πρώτος βασίστηκε στην εκτίμηση των συναρτήσεων πυκνότητας πιθανότητας των δεδομένων εκπαίδευσης για κάθε κλάση και την ταξινόμηση νέων σημείων άγνωστης κλάσης στην πιο πιθανή κλάση. Ο δεύτερος τρόπος βασίστηκε στην τοπολογία των σημείων εκπαίδευσης και στην ταξινόμηση νέων σημείων στην κλάση που ανήκουν οι κοντινότεροι γείτονές του. Στο υπόλοιπο κεφάλαιο θα δούμε έναν τρίτο τρόπο προσέγγισης που επιχειρεί να μοντελοποιήσει αναλυτικά τα δεδομένα και να τα διαχωρίσει με κάποια υπερεπιφάνεια απόφασης (***decision hyperplane***). Αρχικά θα εξεταστούν *γραμμικοί ταξινομητές* (***linear classifiers***) που παράγουν γραμμικά όρια απόφασης και μπορούν να διαχωρήσουν επιτυχώς δεδομένα που είναι *γραμμικά διαχωρίσιμα* (***linearly separable***). Στη συνέχεια θα επιχειρήσουμε μία επέκταση των αλγορίθμων αυτών με σκοπό

να είναι σε θέση να διαχωρίσουν μη γραμμικά δεδομένα. Ενδεικτικά παραθέτουμε το σχήμα 3.1.



Σχήμα 3.1

Στην εικόνα αριστερά παρατηρούμε γραμμικά διαχωρίσιμα δεδομένα. Στόχος μας είναι να υπολογίσουμε αναλυτικά την εξίσωση της κόκκινης ευθείας γραμμής που μπορεί να διαχωρίσει τα δεδομένα. Η ευθεία αυτή γραμμή είναι στη γενικότερη περίπτωση το υπερεπίπεδο απόφασης που σε χώρο δύο διαστάσεων εκφυλίζεται σε ευθεία. Στη μεσαία εικόνα φαίνονται δεδομένα που είναι αδύνατο να διαχωριστούν επιτυχώς από μία ευθεία γραμμή. Απαιτείται τώρα μία καμπύλη γραμμή ή στη περίπτωση περισσότερων διαστάσεων μία υπερεπιφάνεια απόφασης. Η καμπύλη αυτή υπολογίζεται αναλυτικά με τη βοήθεια ενός *μη γραμμικού ταξινομητή* (*nonlinear classifier*). Η τελευταία εικόνα παρουσιάζει δεδομένα αδιαχώριστα (*inseparable*). Στην πράξη μπορούν να διαχωριστούν από μία πολύπλοκη καμπύλη γραμμή αλλά με αυτό τον τρόπο το μοντέλο είναι πιο πιθανό να μοντελοποιεί θόρυβο στα δεδομένα παρά πραγματικά δεδομένα. Αυτό το πρόβλημα καλείται *overfitting*.

3.3 Γραμμική και Λογιστική Παλινδρόμηση

Η πιο απλή περίπτωση γραμμικού μοντέλου είναι ο αλγόριθμος *Linear Regression* και είναι ταυτόσημος με την έννοια των *ελαχίστων τετραγώνων* (*least squares*). Σε ένα n-διάστατο χώρο αναζητείται η εξίσωση ενός υπερεπιπέδου ώστε να ελαχιστοποιείται το *μέσο τετραγωνικό σφάλμα* (*mean square error*). Σε αυτό το σημείο αξίζει να επισημανθεί η διαφορά μεταξύ παλινδρόμησης (*regression*) και ταξινόμησης (*classification*). Στην πρώτη περίπτωση, δεν υπάρχουν κλάσεις και το ζητούμενο είναι η αναλυτική σχέση της ανεξάρτητης μεταβλητής με τις εξαρτημένες. Στην περίπτωση δύο διαστάσεων x και y αναζητείται η σχέση $y = f(x)$ ή $y = ax + b$ για γραμμική παλινδρόμηση και προβλέπεται η τιμή του y για νέα x. Στην ταξινόμηση αναζητούμε τη θέση ενός νέου σημείου (x, y) ως προς την γραμμή απόφασης και την καταχώρηση του σε μία εκ των δύο κλάσεων. Στην πράξη το πρώτο σκέλος είναι κοινό, δηλαδή με βάση τα δεδομένα σημεία κατασκευάζεται το γραμμικό μοντέλο. Το δεύτερο σκέλος διαφέρει. Στην παλινδρόμηση δίνονται x και ζητούνται οι τιμές y και στην ταξινόμηση δίνονται ζεύγη (x, y) και ζητείται η θέση τους ως πρός το μοντέλο.

Προτού προχωρήσουμε στη μαθηματική θεμελίωση της γραμμικής παλινδρόμησης ας αναφέρουμε ότι πλεονέκτημα των γραμμικών ταξινομητών είναι η απλότητά τους και οι μικρές υπολογιστικές απαιτήσεις. Μειονέκτημα αποτελεί ο περιορισμός τους σε γραμμικά διαχωρίσιμα δεδομένα. Μειονέκτημα ειδικά της γραμμικής παλινδρόμησης αποτελεί ο υποβέλτιστος διαχωρισμός των δεδομένων για τις ανάγκες της ταξινόμησης. Το τελευταίο πρόβλημα επιχειρεί να επιλύσει ο αλγόριθμος SVM που θα εξεταστεί στη συνέχεια.

Δοθέντος ενός διανύσματος $\mathbf{x}$ η έξοδος του γραμμικού ταξινομητή είναι

$$y = \mathbf{w}^T\mathbf{x} + w_0$$

όπου $\mathbf{w} = (w_1, w_2, \dots, w_n)$ το *διάνυσμα βαρών* (*weight vector*) ή ισοδύναμα

$$y = \ \mathbf{w}^T\mathbf{x}$$

με επαύξηση των διανυσμάτων $\mathbf{x}$ με 1 στην αρχή τους και $\mathbf{w} = (w_0, w_1, w_2, \dots, w_n)$. Για τις ανάγκες της ταξινόμησης θεωρούμε δύο κλάσεις και θέλουμε την έξοδο να παίρνει τιμές ± 1 ανάλογα με την κλάση του σημείου.

Με τις παραπάνω θεωρήσεις το μέσο τετραγωνικό σφάλμα που επιδιώκουμε να ελαχιστοποιήσουμε είναι

$$J(\mathbf{w}) = E[|y - \mathbf{x}^T\mathbf{w}|^2]$$

όπου ο τελεστής E δηλώνει μέση τιμή των σφαλμάτων στα N δείγματα εκπαίδευσης. Αναζητείται το διάνυσμα βαρών που ελαχιστοποιεί την παραπάνω ποσότητα, δηλαδή

$$\widehat{\boldsymbol{w}} = argmin_{\mathbf{w}} J(\mathbf{w})$$

Με παραγώγιση του κριτηρίου ελαχιστοποίησης έχουμε

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 2E[\mathbf{x}(y - \mathbf{x}^T\mathbf{w})]$$

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 0 \ \Rightarrow \ R_{\mathbf{x}}\widehat{\boldsymbol{w}} = E[y\mathbf{x}] \Rightarrow \ \widehat{\boldsymbol{w}} = R_{\mathbf{x}}^{-1}E[\mathbf{x}y]$$

όπου $R_{\mathbf{x}}$ ο πίνακας αυτοσυσχέτισης (*autocorrelation matrix*) του διανύσματος $\mathbf{x}$ και $E[y\mathbf{x}]$ η ετεροσυσχέτιση του $\mathbf{x}$ με την έξοδο y. Πιο αναλυτικά

$$R_{\mathbf{x}} = E[\mathbf{x}\mathbf{x}^T] = \begin{bmatrix} E[x_1x_1] & E[x_1x_2] & \dots & E[x_1x_n] \\ E[x_2x_1] & E[x_2x_2] & \dots & E[x_2x_n] \\ \vdots & \vdots & \vdots & \vdots \\ E[x_nx_1] & E[x_nx_2] & \dots & E[x_nx_n] \end{bmatrix}$$

$$E[y\mathbf{x}] = E[\begin{bmatrix} x_1 y \\ x_2 y \\ \vdots \\ x_n y \end{bmatrix}]$$

Εάν λοιπόν ο πίνακας αυτοσυσχέτισης είναι αντιστρέψιμος, το υπερεπίπεδο προκύπτει σαν μία λύση ενός συνόλου γραμμικών εξισώσεων.

Στην πράξη, ο απλός ταξινομητής Linear Regression δε θα χρησιμοποιηθεί, καθώς οι μηχανές διανυσμάτων υποστήριξης που θα δούμε στην επόμενη ενότητα αντιμετωπίζουν το πρόβλημα με τον ίδιο τρόπο αλλά βέλτιστα κατασκευάζοντας το υπερεπίπεδο με το μέγιστο περιθώριο. Ωστόσο θα χρησιμοποιηθεί ο ταξινομητής Logistic Regression, ο οποίος είναι δημοφιλής για προβλήματα επεξεργασίας κειμένου. Ο ταξινομητής αυτός έχει στενή σχέση με τον ταξινομητή μέγιστης εντροπίας (Maximum Entropy Classifier).

Logistic Regression

Έστω ένα σύνολο εκπαίδευσης με N δείγματα, $D = \{(\mathbf{x}_i, y_i): i = 1,2, \dots, N\}$ και ένα πρόβλημα δύο κλάσεων με $y = 0$ ή $y = 1$. Η πιθανότητα ένα σημείο $\mathbf{x}$ να ανήκει στην κλάση $y = 1$ μοντελοποιείται ως εξής

$$P(y = 1|\mathbf{x}, \mathbf{w}) = P_1(\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{wx})}$$

Τότε η πιθανότητα το σημείο να ανήκει στην κλάση $y = 0$ δίνεται από τη σχέση

$$P(y = 0|\mathbf{x}, \mathbf{w}) = P_0(\mathbf{x}) = 1 - P_1(\mathbf{x}) = \frac{\exp(-\mathbf{wx})}{1 + \exp(-\mathbf{wx})}$$

Η συνάρτηση

$$f(z) = \frac{1}{1 + \exp(-z)}$$

καλείται λογιστική συνάρτηση και δανείζει το όνομά της σε αυτή τη μέθοδο ταξινόμησης. Περισσότερα για τη λογιστική συνάρτηση θα δούμε στην ενότητα των νευρωνικών δικτύων. Το διάνυσμα $\mathbf{w}$ είναι το διάνυσμα βαρών και σκοπός είναι ο προσδιορισμός του μέσα από την ελαχιστοποίηση κάποιας συνάρτησης κόστους.

Ο φυσικός λογάριθμος του λόγου των δύο πιθανοτήτων είναι γραμμική συνάρτηση αφού

$$\log\left(\frac{P_0(\mathbf{x})}{P_1(\mathbf{x})}\right) = \log(\exp(-\mathbf{wx})) = -\mathbf{wx}$$

ή

$$\log\left(\frac{P_1(\mathbf{x})}{P_0(\mathbf{x})}\right) = \log\left(\frac{1}{\exp(-\mathbf{wx})}\right) = \mathbf{wx}$$

Η ταξινόμηση γίνεται πιθανοτικά, όπως στην περίπτωση του αλγορίθμου Naive Bayes και κάθε σημείο καταχωρείται στην κλάση στην οποία το μοντέλο δίνει μεγαλύτερη πιθανότητα. Δηλαδή για να καταχωρηθεί το σημείο $\mathbf{x}$ στην κλάση $y = 1$ πρέπει

$$P_1(\mathbf{x}) > P_0(\mathbf{x}) \Rightarrow \frac{P_1(\mathbf{x})}{P_0(\mathbf{x})} > 1 \Rightarrow \log\left(\frac{P_1(\mathbf{x})}{P_0(\mathbf{x})}\right) > 0 \Rightarrow \mathbf{wx} > 0$$

Όμοια, για να γίνει ταξινόμηση στην κλάση $y = 0$ πρέπει

$$P_0(\mathbf{x}) > P_1(\mathbf{x}) \Rightarrow \frac{P_0(\mathbf{x})}{P_1(\mathbf{x})} > 1 \Rightarrow \log\left(\frac{P_0(\mathbf{x})}{P_1(\mathbf{x})}\right) > 0 \Rightarrow -\mathbf{wx} > 0 \Rightarrow \mathbf{wx} < 0$$

Ο λόγος αυτός λοιπόν, καθορίζει τη μορφή του υπερεπιπέδου απόφασης και για αυτό ο ταξινομητής Logistic Regression είναι γραμμικός. Η λογιστική συνάρτηση αντιστοιχεί διανύσματα του χώρου χαρακτηριστικών στο διάστημα $[0,1]$ και γι'αυτό χρησιμοποιείται συχνά για την μοντελοποίηση πιθανοτήτων. Η γενίκευσή της σε περισσότερες κλάσεις καλείται συνάρτηση softmax.

Για τον προσδιορισμό των βαρών $\mathbf{w}$ χρησιμοποιείται η μέθοδος Maximum Likelihood δηλαδή ως συνάρτηση κόστους ορίζεται η πιθανοφάνεια των δεδομένων και στόχος είναι η μεγιστοποίησή της. Ο λογάριθμος της συνάρτησης πιθανοφάνειας (log-likelihood) είναι

$$\log\big(P(D|\mathbf{w})\big) = \log\left(\prod_{i=1}^{N} P(\mathbf{x}_i, y_i|\mathbf{w})\right) = \sum_{i=1}^{N} \log\big(P(\mathbf{x}_i, y_i|\mathbf{w})\big) = \sum_{i=1}^{N} \log(P(y_i|\mathbf{x}_i, \mathbf{w}) \cdot P(\mathbf{x}_i|\mathbf{w}))$$

Η πιθανότητα $P(\mathbf{x}_i|\mathbf{w})$ είναι ίση με $P(\mathbf{x}_i)$ καθώς $\mathbf{x}_i$ και $\mathbf{w}$ είναι ανεξάρτητα. Οπότε τα βάρη $\mathbf{w}$ εκτιμούνται βάσει της σχέσης

$$\widehat{\mathbf{w}} = argmax_{\mathbf{w}} \log\big(P(D|\mathbf{w})\big) = argmax_{\mathbf{w}} \left(\sum_{i=1}^{N} \log\big(P(y_i|\mathbf{x}_i, \mathbf{w}) \cdot P(\mathbf{x}_i)\big)\right)$$

$$\widehat{\mathbf{w}} = argmax_{\mathbf{w}} \left(\sum_{i=1}^{N} \log P(y_i|\mathbf{x}_i, \mathbf{w})\right)$$

Η πιθανότητα όμως εντός του λογαρίθμου ορίστηκε προηγουμένως και είναι

$$P(y_i|\mathbf{x}_i, \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{wx}_i)} = \widehat{y_i} \ , \alpha\nu \ y_i = 1$$

$$P(y_i|\mathbf{x}_i, \mathbf{w}) = 1 - \widehat{y_i} \ , \alpha\nu \ y_i = 0$$

Ενσωματώνοντας τις δύο εξισώσεις σε μία, έχουμε

$$P(y_i|\mathbf{x}_i, \mathbf{w}) = \hat{y}_i^{\,y_i} \cdot (1 - \hat{y}_i)^{1-y_i}$$

και η σχέση προσδιορισμού των βαρών, πλέον γίνεται

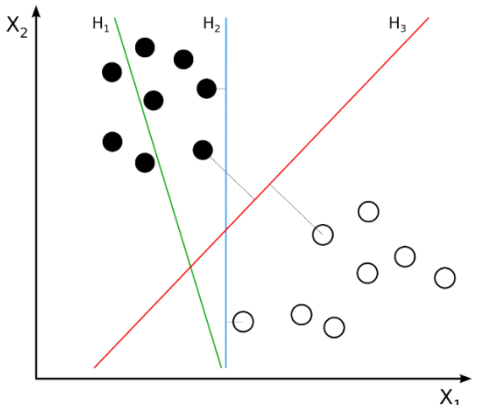
$$\hat{\mathbf{w}} = argmax_{\mathbf{w}} \left(\sum_{i=1}^{N} \log(\hat{y}_i^{\,y_i} \cdot (1 - \hat{y}_i)^{1-y_i}) \right)$$

$$\hat{\mathbf{w}} = argmax_{\mathbf{w}} \left(\sum_{i=1}^{N} (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)) \right)$$

Η τελευταία σχέση εκφράζει την εντροπία των κατανομών και έτσι ο ταξινομητής ονομάζεται εναλλακτικά ταξινομητής μέγιστης εντροπίας. Ο προσδιορισμός των βαρών μπορεί να γίνει με διάφορες μεθόδους βελτιστοποίησης. Εκτενής ανάλυση των πιθανοτικών μοντέλων ταξινόμησης παρουσιάζεται στο [2].

3.4 Μηχανές Διανυσμάτων Υποστήριξης

Η *μηχανή διανυσμάτων υποστήριξης* (*support vector machine - SVM*) είναι ένας γραμμικός ταξινομητής που διαχωρίζει δεδομένα δύο κλάσεων σε ένα χώρο πολλών διαστάσεων. Η διαφορά του από τον απλό γραμμικό ταξινομητή μέσου τετραγωνικού σφάλματος, που είδαμε προηγουμένως, είναι ότι κατασκευάζει το υπερεπίπεδο διαχωρισμού με τέτοιο τρόπο ώστε να βρίσκεται στη μεγαλύτερη δυνατή απόσταση από τα κοντινότερα σημεία εκπαίδευσης και των δύο κλάσεων. Εναλλακτικά, από τα άπειρα υπερεπίπεδα που διαχωρίζουν τα γραμμικά διαχωρίσιμα δεδομένα, επιλέγει αυτό με το *μέγιστο περιθώριο* και ως προς τις δύο κλάσεις. Στο σχήμα 3.2 φαίνονται μερικά γραμμικά διαχωρίσιμα δεδομένα σε χώρο δύο διαστάσεων.



Σχήμα 3.2

Η πράσινη ευθεία δεν διαχωρίζει τα δεδομένα. Η μπλε γραμμή τα διαχωρίζει αλλά υποβέλτιστα. Η κόκκινη γραμμή είναι το όριο που κατασκευάζει η μηχανή διανυσμάτων υποστήριξης που αφήνει το μέγιστο δυνατό περιθώριο και στις δύο κλάσεις ώστε νέα δεδομένα να έχουν μικρότερη πιθανότητα να προκαλέσουν σφάλμα. Αυτή η ευθεία είναι πιο αξιόπιστη, αφού γενικεύει καλύτερα σε νέα δεδομένα.

Η κατασκευή ταξινομητών που δε διαχωρίζουν απλά τα δεδομένα, αλλά επιχειρούν να τα διαχωρίσουν με τέτοιο τρόπο ώστε να συμπεριφέρονται καλύτερα σε νέα δεδομένα βασίζεται στην αρχή της ***απόδοσης γενίκευσης*** (***generalization performance***). Το υπερεπίπεδο καλείται ***υπερεπίπεδο μέγιστου περιθωρίου*** (***maximum margin hyperplane***) και ο ίδιος ο SVM ταξινομητής, ***ταξινομητής μέγιστου περιθωρίου*** (***maximum margin classifier***). Τα πλησιέστερα σημεία στο υπερεπίπεδο, από τα οποία η απόσταση θέλουμε να είναι μέγιστη ονομάζονται ***διανύσματα υποστήριξης*** (***support vectors***) και ο ρόλος τους στην κατασκευή του υπερεπιπέδου θα γίνει σαφής στη συνέχεια.

Ο αλγόριθμος SVM επεκτείνεται και σε περισσότερες των δύο κλάσεων, αλλά και σε δεδομένα που δεν είναι γραμμικά διαχωρίσιμα με τη ***μέθοδο πυρήνα*** (***kernel method***). Στη μέθοδο αυτή τα δεδομένα μετασχηματίζονται μη γραμμικά σε ένα χωρο περισσότερων διαστάσεων στον οποίο είναι γραμμικά διαχωρίσιμα. Ο απλός αλγόριθμος SVM προτάθηκε αρχικά το 1963 από τους Vapnik και Chernovenkis και η επέκταση του με τη μέθοδο πυρήνα το 1992 από τους Boser, Guyon και Vapnik.

Στη συνέχεια θα εξετάσουμε αναλυτικά τον αλγόριθμο και το πρόβλημα ***κυρτής βελτιστοποίησης*** (***convex optimization***) στο οποίο καταλήγει. Από τις διάφορες μεθόδους που έχουν προταθεί για την επίλυση του προβλήματος βελτιστοποίησης θα παρουσιαστεί η πιο κοινή μέθοδος που βασίζεται στη ***δυικότητα Lagrange*** (***Lagrangian duality***) και τις συνθήκες ***Karush-Kuhn-Tucker***. Στη δεύτερη υποενότητα θα συζητήσουμε για το ***τέχνασμα πυρήνα*** (***Kernel trick***).

3.4.1 Η Γραμμική Περίπτωση

Έστω το δείγμα εκπαίδευσης $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ όπου $\mathbf{x}_i$ είναι το i-οστό διάνυσμα εκπαίδευσης και y_i η αντίστοιχη επιθυμητή έξοδος. Η έξοδος θεωρείται $+1$ για τη μία κλάση και -1 για την άλλη κλάση και τα δεδομένα θεωρούνται γραμμικά διαχωρίσιμα. Μία επιφάνεια απόφασης που διαχωρίζει τα δεδομένα είναι η

$$\mathbf{w}^T\mathbf{x} + b = 0 \text{ ή } g(\mathbf{x}) = 0 \text{ με } g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$

με $\mathbf{x}$ ένα διάνυσμα εκπαίδευσης, $\mathbf{w}$ το διάνυσμα βαρών που αναζητείται και b η ***πόλωση*** (***bias***) που στα προηγούμενα είχε χαρακτηριστεί w_0 και επίσης αναζητείται. Για να ταξινομεί το υπερεπίπεδο αυτό τα δεδομένα εκπαίδευσης επιτυχώς, πρέπει να ισχύει

- Αν $y_i = +1$ τότε $\mathbf{w}^T\mathbf{x}_i + b \geq 0$ (1)
- Αν $y_i = -1$ τότε $\mathbf{w}^T\mathbf{x}_i + b < 0$

Το διάστημα ανάμεσα στο υπερεπίπεδο απόφασης και τα πλησιέστερα σημεία εκπαίδευσης και των δύο κλάσεων, δηλαδή το περιθώριο, συμβολίζεται ρ και στόχος είναι η εύρεση του βέλτιστου υπερεπιπέδου $\mathbf{w}_0$, b_0 που μεγιστοποιεί το περιθώριο αυτό. Αν προβάλουμε ένα διάνυσμα $\mathbf{x}$ πάνω στο βέλτιστο υπερεπίπεδο τότε το διάνυσμα συναρτήσει της προβολής του $\mathbf{x}_p$ δίνεται από τη σχέση

$$\mathbf{x} = \mathbf{x}_p + r\frac{\mathbf{w}}{\|\mathbf{w}\|}$$

αφού το $\mathbf{w}$ είναι κάθετο στο υπερεπίπεδο, με το βαθμωτό μέγεθος r να είναι η κάθετη απόσταση του σημείου $\mathbf{x}$ από το υπερεπίπεδο. Πολλαπλασιάζοντας την παραπάνω σχέση με $\mathbf{w}^T$ και προσθέτοντας την πόλωση b παίρνουμε

$$\mathbf{w}^T\mathbf{x} = \mathbf{w}^T\mathbf{x}_p + r\frac{\mathbf{w}^T\mathbf{w}}{\|\mathbf{w}\|}$$

$$\mathbf{w}^T\mathbf{x} + b = \mathbf{w}^T\mathbf{x}_p + b + r\frac{\|\mathbf{w}\|^2}{\|\mathbf{w}\|}$$

$$g(\mathbf{x}) = g(\mathbf{x}_p) + r\|\mathbf{w}\|$$

Όμως $g(\mathbf{x}_p) = 0$ καθώς η προβολή $\mathbf{x}_p$ βρίσκεται πάνω στο υπερεπίπεδο. Συνεπώς

$$g(\mathbf{x}) = r\|\mathbf{w}\|$$

$$r = \frac{g(\mathbf{x})}{\|\mathbf{w}\|} \tag{2}$$

Δοθέντος λοιπόν ενός σημείου $\mathbf{x}$ το μέτρο της κάθετης απόστασής του από το υπερεπίπεδο $g(\mathbf{x}) = 0$ είναι r και δίνεται από την τελευταία σχέση. Μέχρι στιγμής οι περιορισμοί για τα $\mathbf{w}$, b είναι οι σχέσεις (1) δηλαδή να ταξινομούν σωστά τα δεδομένα. Μπορούμε να κλιμακώσουμε τις παραμέτρους αυτές ώστε να ταξινομούν σωστά τα δεδομένα και η συνάρτηση $g(\mathbf{x})$ να παίρνει τις τιμές ± 1 στα κοντινότερα σημεία δηλαδή

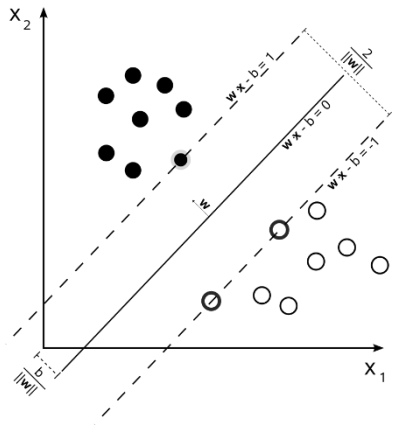
- Αν $y_i = +1$ τότε $\mathbf{w}^T\mathbf{x}_i + b \geq 1$ (3)
- Αν $y_i = -1$ τότε $\mathbf{w}^T\mathbf{x}_i + b \leq -1$

Αυτό μπορεί να γίνει αφού τα δεδομένα είναι γραμμικά διαχωρίσιμα και έχουν μεταξύ τους περιθώριο, οσοδήποτε μικρό και αν είναι.

Οι σχέσεις (3) αποτελούν τους περιορισμούς του βέλτιστου υπερεπιπέδου τώρα. Τα σημεία του δείγματος εκπαίδευσης $\mathbf{x}_s$ για τα οποία οι παραπάνω σχέσεις ισχύουν με ισότητα καλούνται διανύσματα υποστήριξης. Η κάθετη απόσταση των διανυσμάτων υποστήριξης από το βέλτιστο υπερεπίπεδο, βάσει της σχέσης (2) ισούται με

$$r = \frac{g(\mathbf{x}_s)}{\|\mathbf{w}\|} = \pm\frac{1}{\|\mathbf{w}\|}$$

Το περιθώριο λοιπόν δίνεται από τη σχέση



Σχήμα 3.3

$$\rho = 2|r| = \frac{2}{\|\mathbf{w}\|}$$

Στο σχήμα 3.3 στα αριστερά φαίνονται κάποια από τα γεωμετρικά χαρακτηριστικά της ευθείας που παράγει ο SVM ταξινομητής σε δεδομένα δύο διαστάσεων. Αναλυτικά, φαίνονται τα διανύσματα υποστήριξης, η ευθεία που διαχωρίζει τα δεδομένα και το διάνυσμα $\mathbf{w}$ που είναι κάθετο σε αυτή, οι ευθείες που ικανοποιούν τις ισότητες των σχέσεων (3) και τέλος το περιθώριο ρ.

Καταλήγουμε έτσι στο πρόβλημα βελτιστοποίησης του αλγορίθμου ταξινόμησης SVM. Δοθέντος του συνόλου εκπαίδευσης $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ αναζητούνται $\mathbf{w}$, b τέτοια ώστε

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \; για \; i = 1,2,\dots,N \;\; (οι \; σχέσεις \; (3) \; σε \; μία)$$

$$να \; ελαχιστοποιείται \; η \; συνάρτηση \; J(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 = \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

αφού ελαχιστοποίηση της νόρμας $\|\mathbf{w}\|$ ισοδυναμεί με μεγιστοποίηση του περιθωρίου ρ.

$$\mathbf{w}_0 = argmin_{\mathbf{w}} J(\mathbf{w})$$

$$υπό \; τους \; περιορισμούς \; y_i(\mathbf{w}_0^T\mathbf{x}_i + b_0) \geq 1 \; για \; i = 1,2,\dots,N$$

Αυτό είναι ένα μη γραμμικό πρόβλημα βελτιστοποίησης και χαρακτηρίζεται γενικά κυρτής βελτιστοποίησης καθώς η συνάρτηση $J(\mathbf{w})$ είναι κυρτή. Η λύση τέτοιων προβλημάτων είναι μοναδική και βέλτιστη. Οι περιορισμοί είναι γραμμικές ανισότητες και για την εύρεση των $\mathbf{w_0}$, b_0 δηλαδή του βέλτιστου υπερεπιπέδου, χρησιμοποιούνται οι συνθήκες *Karush-Kuhn-Tucker* (συνθήκες ΚΚΤ). Οι συνθήκες αυτές αποτελούν γενίκευση της μεθόδου των πολλαπλασιαστών Lagrange σε περιπτώσεις περιορισμών ανισότητας. Η συνθήκη ελαχιστοποίησης και οι περιορισμοί ενσωματώνονται στη συνάρτηση Lagrange

$$L(\mathbf{w}, b, \boldsymbol{\lambda}) = J(\mathbf{w}) - \sum_{i=1}^{N} \lambda_i[y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1] \qquad (4)$$

όπου λ_i είναι οι μη αρνητικοί πολλαπλασιαστές Lagrange. Οι συνθήκες ΚΚΤ τότε στο σημείο όπου εμφανίζεται το ελάχιστο της Lagrangian συνάρτησης είναι

- $\frac{\partial L(\mathbf{w},b,\boldsymbol{\lambda})}{\partial \mathbf{w}} = 0 \;\Rightarrow\; \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} - \sum_{i=1}^{N} \lambda_i y_i \mathbf{x}_i = 0 \;\Rightarrow\; \mathbf{w} = \sum_{i=1}^{N} \lambda_i y_i \mathbf{x}_i$ (5)

- $\frac{\partial L(\mathbf{w},b,\boldsymbol{\lambda})}{\partial b} = 0 \;\Rightarrow\; 0 - \sum_{i=1}^{N} \lambda_i y_i = 0 \;\Rightarrow\; \sum_{i=1}^{N} \lambda_i y_i = 0$ (6)

- $\lambda_i \geq 0, \gamma\iota\alpha\ i = 1,2,\dots,N$

- $\lambda_i[y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1] = 0, \gamma\iota\alpha\ i = 1,2,\dots,N$

Η τελευταία σχέση υποδηλώνει ότι οι πολλαπλασιαστές λ_i , για σημεία $\mathbf{x}_i$ στα οποία ισχύει $y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \neq 0$, πρέπει να είναι μηδενικοί. Ποιά είναι όμως αυτά τα σημεία; Όπως γίνεται εύκολα κατανοητό είναι τα σημεία στα οποία ισχύει $y_i(\mathbf{w}^T\mathbf{x}_i + b) \neq 1$ δηλαδή $g(\mathbf{x}_i) \neq 1$ ή $g(\mathbf{x}_i) \neq -1$. Όλα τα σημεία συνεπώς του συνόλου εκπαίδευσης πλην των σημείων που καλέσαμε διανύσματα υποστήριξης. Σε αυτά τα σημεία, αφού ισχύει $y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 = 0$ οι πολλαπλασιαστές Lagrange σύμφωνα με την τελευταία συνθήκη ΚΚΤ μπορούν να πάρουν θετικές τιμές. Έχουμε λοιπόν μία έκφραση για το βέλτιστο $\mathbf{w}$ από τη σχέση (5), μία σχέση που συνδέει τους πολλαπλασιαστές Lagrange και το συμπέρασμα ότι αυτοί είναι μη μηδενικοί μόνο στα διανύσματα υποστήριξης. Στην παρούσα φάση, χωρίς να προβούμε σε εκτενή μαθηματική ανάλυση του θέματος της βελτιστοποίησης Lagrange, αναφέρουμε ότι από το παραπάνω πρόβλημα βελτιστοποίησης, που χαρακτηρίζεται *πρωτεύον* (*primal problem*), κατασκευάζεται ένα *δυικό πρόβλημα* του πρωτεύοντος (*dual problem*) με βάση τη δυικότητα Lagrange. Ισοδύναμα το δυικό πρόβλημα καλείται δυική αναπαράσταση κατά Wolfe. Αναλύοντας τη συνάρτηση Lagrange που ορίσαμε προηγουμένως παίρνουμε

$$(4) \Rightarrow L(\mathbf{w},b,\lambda) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \lambda_i y_i \mathbf{w}^T\mathbf{x}_i - b\sum_{i=1}^{N} \lambda_i y_i + \sum_{i=1}^{N} \lambda_i$$

$$(4) \overset{(6)}{\Rightarrow} L(\mathbf{w},b,\lambda) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \lambda_i y_i \mathbf{w}^T\mathbf{x}_i + \sum_{i=1}^{N} \lambda_i \quad (7)$$

$$(5) \Rightarrow \mathbf{w}^T\mathbf{w} = \mathbf{w}^T \sum_{i=1}^{N} \lambda_i y_i \mathbf{x}_i = \sum_{i=1}^{N} \lambda_i y_i \mathbf{w}^T\mathbf{x}_i = \sum_{i=1}^{N} \lambda_i y_i (\sum_{j=1}^{N} \lambda_j y_j \mathbf{x}_j^T)\mathbf{x}_i$$

$$\mathbf{w}^T\mathbf{w} = \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i\lambda_j y_i y_j \mathbf{x}_j^T\mathbf{x}_i \quad (8)$$

$$(7) \overset{(8)}{\Rightarrow} L(\boldsymbol{\lambda}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i\lambda_j y_i y_j \mathbf{x}_j^T\mathbf{x}_i - \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i\lambda_j y_i y_j \mathbf{x}_j^T\mathbf{x}_i + \sum_{i=1}^{N} \lambda_i$$

$$L(\boldsymbol{\lambda}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i\lambda_j y_i y_j \mathbf{x}_j^T\mathbf{x}_i + \sum_{i=1}^{N} \lambda_i$$

Στο δυικό πρόβλημα πλέον, το ζητούμενο είναι η μεγιστοποίηση της συνάρτησης $L(\boldsymbol{\lambda})$ ως προς $\boldsymbol{\lambda}$. Δηλαδή

Δυικό Πρόβλημα

Δοθέντος του συνόλου εκπαίδευσης $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ *αναζητούνται οι συντελεστές* $\lambda_i\ i = 1,2,\dots,N$ *που μεγιστοποιούν τη συνάρτηση*

$$L(\lambda) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y_i y_j \mathbf{x}_j^T\mathbf{x}_i + \sum_{i=1}^{N}\lambda_i$$

ικανοποιώντας παράλληλα τους περιορισμούς

$$\lambda_i \geq 0, \gamma\iota\alpha\ i = 1,2,\dots,N$$

$$\sum_{i=1}^{N}\lambda_i y_i = 0$$

Από το παραπάνω πρόβλημα υπολογίζονται οι συντελεστές $\lambda_i, i = 1,2,\dots,N$. Όπως όμως επισημάνθηκε προηγουμένως μη μηδενικούς συντελεστές έχουμε μόνο στα διανύσματα υποστήριξης. Δηλαδή από τα N σημεία εκπαίδευσης μόνο σε ένα υποσύνολο πλήθους $N_s \leq N$ προκύπτουν μη μηδενικοί λ_i. Αφού υπολογιστούν οι πολλαπλασιαστές μπορούμε να επιστρέψουμε στη σχέση (5) του πρωτεύοντος προβλήματος για να προσδιορίσουμε τελικά τη βέλτιστη λύση $\mathbf{w_0}$

$$\mathbf{w}_0 = \sum_{i=1}^{N}\lambda_i y_i\,\mathbf{x}_i = \sum_{i=1}^{N_s}\lambda_i y_i\,\mathbf{x}_i$$

Το b_0 μπορεί να υπολογιστεί σε κάποιο διάνυσμα υποστήριξης $\mathbf{x}_s$. Για αυτά τα σημεία ισχύει

$$y_i(\mathbf{w}_0^T\mathbf{x}_s + b_0) - 1 = 0 \ \Rightarrow\ y_i(\mathbf{w}_0^T\mathbf{x}_s + b_0) = 1 \ \Rightarrow\ b_0 = \frac{1}{y_i} - \mathbf{w}_0^T\mathbf{x}_s \ \Rightarrow$$

$$b_0 = \frac{1}{y_i} - \sum_{i=1}^{Ns}\lambda_i y_i \mathbf{x}_i^T\mathbf{x}_s$$

Στην πράξη υπολογίζεται σε όλα τα διανύσματα υποστήριξης και τελικά λαμβάνεται ο μέσος όρος όλων ως b_0.

Σημειώνεται, τέλος ένα ενδιαφέρον χαρακτηριστικό αυτού του τύπου βελτιστοποίησης. Στη συνάρτηση Lagrange του δυικού προβλήματος παρατηρούμε ότι τα διανύσματα εκπαίδευσης υπεισέρχονται με τη μορφή εσωτερικών γινομένων $\mathbf{x}_j^T\mathbf{x}_i$. Η συνάρτηση κόστους δηλαδή δεν εξαρτάται από τη διάσταση του χώρου χαρακτηριστικών. Η σημασία της

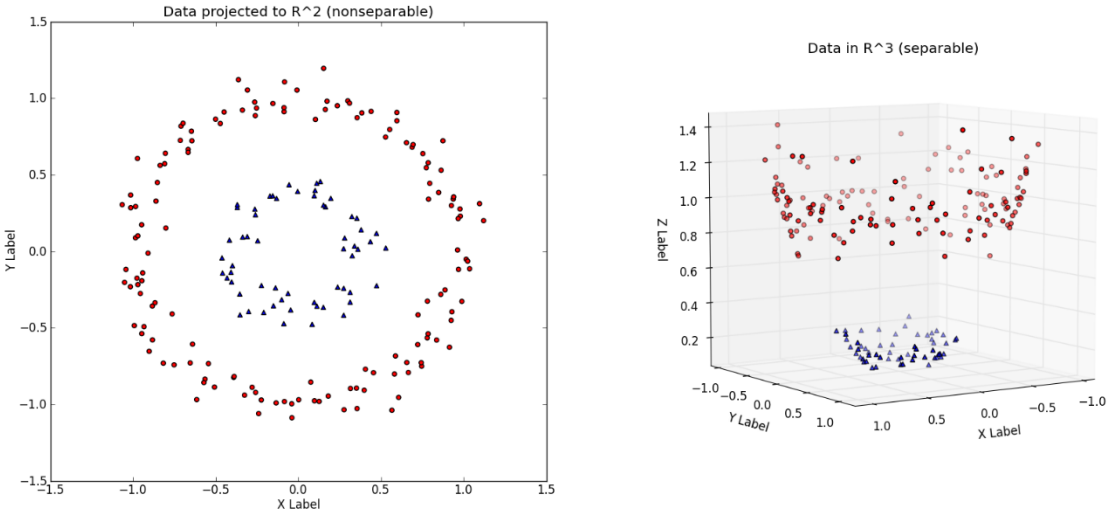
ιδιότητας αυτής θα γίνει πιο κατανοητή στα επόμενα που θα επιχειρήσουμε να διαχωρήσουμε μη γραμμικά διαχωρίσιμες κλάσεις με τη μέθοδο πυρήνα.

3.4.2 Μέθοδοι Πυρήνα

Έστω η περίπτωση προτύπων που δεν είναι γραμμικά διαχωρίσιμα. Τέτοια δεδομένα όπως έγινε φανερό και σε προηγούμενο σχήμα δε μπορούν να διαχωριστούν επιτυχώς από μία ευθεία γραμμή ή ένα υπερεπίπεδο σε χώρους άνω των δύο διαστάσεων, αλλά απαιτούν μία καμπύλη ή υπερεπιφάνεια γενικότερα. Ωστόσο ο αναλυτικός υπολογισμός μίας υπερεπιφάνειας που διαχωρίζει τα δεδομένα, όπως γίνεται αντιληπτό, είναι ένα αρκετά δύσκολο αν όχι ανεπίλυτο πρόβλημα. Πως θα μπορούσε να επιλυθεί ένα τέτοιο πρόβλημα ταξινόμησης;

Η γενική ιδέα είναι ο μετασχηματισμός του χώρου χαρακτηριστικών σε ένα χώρο υψηλότερης διάστασης. Τα δεδομένα μπορεί να μην είναι γραμμικά διαχωρίσιμα στον αρχικό χώρο αλλά είναι πιθανό να είναι διαχωρίσιμα στον τελικό χώρο. Στο σχήμα 3.4 δίνεται ένα σχηματικό παράδειγμα.

Στα αριστερά βλέπουμε δεδομένα σε ένα χώρο δύο διαστάσεων τα οποία δεν είναι γραμμικά διαχωρίσιμα. Στα δεξιά τα δεδομένα μετασχηματίζονται με κάποιο μη γραμμικό τρόπο σε ένα χώρο τριών διαστάσεων. Πλέον είναι γραμμικά διαχωρίσιμα αφού μπορούν να διαχωριστούν με ένα επίπεδο στον τρισδιάστατο χώρο. Ωστόσο εγείρεται το εξής ερώτημα: στο παράδειγμα απαιτείται μία αύξηση 50% στη διάσταση του χώρου και γενικότερα για να καταστούν τα δεδομένα γραμμικώς διαχωρίσιμα απαιτείται συνήθως μία γενναιόδωρη αύξηση της διάστασης του χώρου χαρακτηριστικών. Πόσο εφικτό είναι υπολογιστικά να γίνει η ταξινόμηση σε ένα χώρο τόσο μεγαλύτερης διάστασης;



Σχήμα 3.4

Η απάντηση δίνεται από τη σημείωση στο τέλος της προηγούμενης υποενότητας. Τα δεδομένα εκπαίδευσης υπεισέρχονται στο πρόβλημα βελτιστοποίησης σαν ζεύγη με τη μορφή εσωτερικών γινομένων. Εάν προσδιορίζαμε κάποιο συγκεκριμένο είδος μετασχηματισμού που αντιστοιχεί απευθείας εσωτερικά γινόμενα του χώρου μικρής διάστασης σε εσωτερικά γινόμενα του χώρου μεγάλης διάστασης με κάποιο μη γραμμικό τρόπο το πρόβλημα θα ήταν πολύ πιο απλό. Τέτοιοι μετασχηματισμοί καλούνται ***συναρτήσεις πυρήνα*** ή πιο απλά ***πυρήνες*** (***kernels***). Ας προχωρήσουμε λοιπόν στη μαθηματική θεμελίωση της παραπάνω ιδέας.

Έστω η μη γραμμική απεικόνιση

$$\mathbf{x} \in \mathbb{R}^n \xrightarrow{\varphi} \mathbf{z} \in \mathbb{R}^m \ , \qquad m > n$$

όπου ο χώρος $\mathbb{R}^m$ είναι χώρος ***Hilbert*** δηλαδή χώρος με ορισμένη πράξη εσωτερικού γινομένου που μπορεί να έχει ακόμα και άπειρη διάσταση. Το εσωτερικό γινόμενο σε αυτό το χώρο δηλώνεται ως

$$\langle \varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2) \rangle = \langle \mathbf{z}_1, \mathbf{z}_2 \rangle = K(\mathbf{x}_1, \mathbf{x}_2)$$

και είναι κάποια συμμετρική και συνεχής συνάρτηση $K(\mathbf{x}_1, \mathbf{x}_2)$ αρκεί αυτή να ικανοποιεί τη συνθήκη

$$\int_C \int_C K(\mathbf{x}_1, \mathbf{x}_2) g(\mathbf{x}_1) g(\mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2 \geq 0$$

για κάθε $g(\mathbf{x}), \mathbf{x} \in C \subset \mathbb{R}^n$ τέτοια ώστε

$$\int_C g(\mathbf{x})^2 d\mathbf{x} < +\infty$$

όπου C ένα συμπαγές και πεπερασμένο υποσύνολο του $\mathbb{R}^n$.

Τα παραπάνω αποτελούν το ***θεώρημα του Mercer*** και δηλώνουν πως αν μία συνεχής και συμμετρική συνάρτηση $K(\mathbf{x}_1, \mathbf{x}_2)$ ικανοποιεί τις παραπάνω προϋποθέσεις τότε ορίζει ένα εσωτερικό γινόμενο σε κάποιο χώρο Hilbert. Η διάσταση αυτού του χώρου δεν είναι γνωστή και μπορεί να είναι και άπειρη. Επίσης ούτε ο ακριβής μετασχηματισμός φ είναι γνωστός αλλά η συνάρτηση $K(\mathbf{x}_1, \mathbf{x}_2)$ είναι όλα όσα χρειαζόμαστε για να υλοποιήσουμε τον ταξινομητή SVM στο νέο χώρο. Η συνάρτηση αυτή καλείται πυρήνας και ο χώρος, ***χώρος Hilbert αναπαραγωγού πυρήνα*** (***Reproducing Kernel Hilbert Space - RKHS***). Το δυικό πρόβλημα βελτιστοποίησης του αλγορίθμου SVM στο νέο χώρο υψηλότερης διάστασης είναι πλέον το ακόλουθο.

Δυικό Πρόβλημα

Δοθέντος του συνόλου εκπαίδευσης $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ *αναζητούνται οι συντελεστές* $\lambda_i\ i = 1,2,\dots,N$ *που μεγιστοποιούν τη συνάρτηση*

$$L(\lambda) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^{N}\lambda_i$$

ικανοποιώντας παράλληλα τους περιορισμούς

$$0 \le \lambda_i \le C\ , \gamma\iota\alpha\ i = 1,2,\dots,N$$

$$\sum_{i=1}^{N}\lambda_i y_i = 0$$

Αφού προσδιοριστούν οι συντελεστές, στη συνέχεια, όπως και στη γραμμική περίπτωση προσδιορίζονται τα $\mathbf{w_0}$ και b_0. Αυτά ορίζουν το υπερεπίπεδο στο νέο χώρο. Στον αρχικό χώρο χαρακτηριστικών το υπερεπίπεδο αντιστοιχίζεται σε πολύπλοκη υπερεπιφάνεια λόγω της μη γραμμικής φύσης του μετασχηματισμού $\boldsymbol{\varphi}$.

Τέλος επισημαίνονται οι συνηθέστερες συναρτήσεις πυρήνα που χρησιμοποιούνται σε εφαρμογές. Είναι συναρτήσεις που ικανοποιούν τις συνθήκες του θεωρήματος Mercer και αντιστοιχίζουν δεδομένα σε χώρους υψηλότερης διάστασης. Ωστόσο όπως προείπαμε η αντίστοιχιση αυτή δεν είναι ρητά προσδιορισμένη. Μεταφερόμαστε με μη γραμμικό τρόπο σε ένα νέο χώρο υψηλότερης διάστασης αλλά δεν υπολογίζονται ρητά σημεία στο νέο χώρο παρά μόνο εσωτερικά γινόμενα.

Πολυωνυμικός Πυρήνας – Polynomial Kernel

$$K(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1^T\mathbf{x_2} + 1)^p\ , \qquad p > 0$$

Συνάρτηση Πυρήνα Ακτινωτής Βάσης – Radial Base Function Kernel (RBF)

$$K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2\sigma^2})$$

Πυρήνας Υπερβολικής Εφαπτομένης – Hyperbolic Tangent Kernel

$$K(\mathbf{x}_1, \mathbf{x}_2) = \tanh(\beta\mathbf{x}_1^T\mathbf{x_2} + \gamma)$$

Σε αυτό το σημείο ολοκληρώνεται η σύντομη θεωρητική εισαγωγή στις μηχανές διανυσμάτων υποστήριξης. Μία αναλυτικότερη παρουσιάση των μηχανών διανυσμάτων υποστήριξης και των υπερπαραμέτρων τους δίνεται στο [26].

Στη συνέχεια θα εξεταστούν τα νευρωνικά δίκτυα, αλγόριθμοι ταξινόμησης που επίσης μετασχηματίζουν δεδομένα σε χώρους υψηλότερης διάστασης με μη γραμμικό τρόπο. Διαφέρουν ωστόσο στον τρόπο προσέγγισης και στον τρόπο που εκπαιδεύονται πάνω στο σύνολο εκπαίδευσης.

3.5 Τεχνητά Νευρωνικά Δίκτυα

Τα *τεχνητά νευρωνικά δίκτυα* (*Artificial Neural Networks - ANNs*) είναι υπολογιστικά μοντέλα που εμπνέονται από βιολογικές διαδικασίες μάθησης. Απαρτίζονται από απλούς υπολογιστικούς κόμβους που καλούνται *νευρώνες* (*neurons*) οι οποίοι συνδέονται μεταξύ τους δημιουργώντας ένα δίκτυο. Κάθε διασύνδεση νευρώνων στο δίκτυο χαρακτηρίζεται από κάποιο βάρος w_i που προσαρμόζεται κατά τη φάση εκπαίδευσης του δικτύου. Η εκπαίδευση συνολικά συνιστά τη διαδικασία προσαρμογής όλων των βαρών του δικτύου. Τα παραπάνω απαντώνται γενικά στις διάφορες κατηγορίες νευρωνικών δικτύων, ωστόσο σε αυτή την ενότητα θα μας απασχολήσουν μερικές μόνο υλοποιήσεις τους. Συγκεκριμένα θα μελετήσουμε το *perceptron πολλών επιπέδων* (*Multi-Layer Perceptron - MLP*) και έπειτα το *συνελικτικό νευρώνικο δίκτυο* (*convolutional neural network*). Όπως θα φανεί στη συνέχεια το δεύτερο ακολουθεί μία παρόμοια προσέγγιση με το πρώτο αλλά κάνει χρήση συνελίξεων αντί εσωτερικών γινομένων.

Τα παραπάνω δίκτυα είναι επιβλεπόμενης μάθησης, ωστόσο νευρωνικά μοντέλα συναντάμε και σε μη επιβλεπόμενη μάθηση με χαρακτηριστικότερο παράδειγμα τους χάρτες αυτό-οργάνωσης του Kohonen.

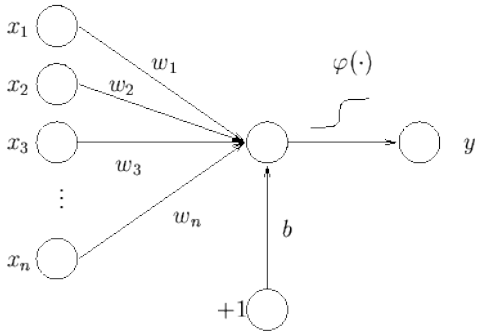
Τα νευρωνικά μοντέλα μάθησης βασίζονται στη λειτουργία του ανθρώπινου εγκεφάλου. Αν και το κατά πόσο προσομοιάζουν τη διαδικασία βιολογικής μάθησης πιστά, είναι πολλές φορές υπό αμφισβήτηση, οι βασικές τους αρχές είναι αυτές που πιστεύεται ότι διέπουν το κεντρικό νευρικό σύστημα. Η μη γραμμικότητα, η παραλληλία, το μεγάλο πλήθος νευρώνων, οι πολύπλοκες διασυνδέσεις είναι όλα χαρακτηριστικά νευρωνικών δικτύων, βιολογικών και τεχνητών. Στην πράξη πάντως, τα νευρωνικά δίκτυα έχουν επιδείξει υψηλές επιδόσεις σε διάφορες εφαρμογές. Ειδικά την τελευταία δεκαετία με τη συνεχή αύξηση των υπολογιστικών πόρων, δίνεται η δυνατότητα σε πολύπλοκα νευρωνικά δίκτυα να εκπαιδευτούν σε μεγάλους όγκους δεδομένων. Η *βαθιά μάθηση* (*deep learning*), δηλαδή η εκπαίδευση νευρωνικών δικτύων με πολλά κρυφά επίπεδα σημειώνει πλέον τα κορυφαία αποτελέσματα στα πεδία της *υπολογιστικής όρασης* (*computer vision*) και της *αναγνώρισης φωνής* (*speech recognition*).

Η αρχή της μελέτης τέτοιων μοντέλων έγινε το 1943 όταν οι McCulloch και Pitts πρότειναν ένα υπολογιστικό μοντέλο του βιολογικού νευρωνικού δικτύου και έθεσαν τα θεμέλια της έρευνας. Στα τέλη της δεκαετίας ο Donald Hebb θεμελίωσε τη *Χεμπιανή μάθηση* (*Hebbian learning*) που έδωσε την ιδέα για προσαρμόσιμα βάρη στις διασυνδέσεις του δικτύου. Το 1958 ο Frank Rosenblatt πρότεινε το *perceptron*, ένα νευρωνικό δίκτυο δύο επιπέδων,

ικανό να ταξινομήσει επιτυχώς γραμμικά διαχωρίσιμα δεδομένα. Η έρευνα στο πεδίο των νευρωνικών δικτύων έμεινε στάσιμη τα επόμενα χρόνια καθώς απαιτούσαν υπολογιστικούς πόρους που ξεπερνούσαν τους διαθέσιμους ενώ έλειπε και ένας αποτελεσματικός αλγόριθμος εκπαίδευσης. Τουλάχιστον μέχρι το 1975 όταν και ο Paul Werbos πρότεινε τον ***αλγόριθμο οπισθοδιάδοσης*** (***Backpropagation algorithm***), έναν αποτελεσματικό τρόπο εκπαίδευσης δικτύων πολλών επιπέδων. Αυτά τα δίκτυα είναι η γενίκευση του απλού perceptron και χρησιμοποιούν ***κρυφά επίπεδα*** (***hidden layers***) για να αντιστοιχίσουν δεδομένα σε χώρους υψηλότερης διάστασης, μία προσέγγιση που θυμίζει τη μέθοδο πυρήνα του αλγορίθμου SVM. Στα επόμενα θα επιχειρήσουμε μία σύντομη εισαγωγή στα νευρωνικά δίκτυα που θα χρησιμοποιήσουμε στο πλαίσιο αυτής της εργασίας. Θα μιλήσουμε αρχικά για το perceptron του Rosenblatt, στη συνέχεια για το perceptron πολλων επιπέδων και τον αλγόριθμο backpropagation και τέλος για τα συνελικτικά δίκτυα.

3.5.1 Το Perceptron του Rosenblatt

Το μοντέλο perceptron είναι η πιο απλή υλοποίηση του ***δικτύου εμπρόσθιας διάδοσης*** (***feed-forward network***). Απαρτίζεται από ένα μόνο νευρώνα, παρόμοιο με το νευρώνα που περιέγραψαν αρχικά οι McCulloch και Pitts. Στα δίκτυα εμπρόσθιας διάδοσης κάθε διάνυσμα εκπαίδευσης τροφοδοτείται στην αρχή του δικτύου και μέσα από τους επιμέρους υπολογισμούς του δικτύου προκύπτει μία ή περισσότερες έξοδοι. Ας δούμε αρχικά το μοντέλο του νευρώνα.



Σχήμα 3.5

Στο σχήμα 3.5 φαίνεται ένας νευρώνας που τροφοδοτείται από μία είσοδο $\mathbf{x} = (x_1, x_2, \dots, x_n)$. Κάθε επιμέρους χαρακτηριστικό της εισόδου πολλαπλασιάζεται με ένα βάρος $w_i, i = 1,2, \dots, n$ (***synaptic weight***) και στη συνέχεια όλα τα γινόμενα αθροίζονται και προστίθεται και ένας όρος πόλωσης b. Το αποτέλεσμα v περνάει από μία μη γραμμική συνάρτηση $\varphi(x)$ και τελικά προκύπτει η έξοδος y. Δηλαδή

$$v = x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + b = \mathbf{w}^T \mathbf{x} + b$$

$$y = \varphi(v) = \varphi(\mathbf{w}^T \mathbf{x} + b)$$

όπου $\mathbf{w} = (w_1, w_2, \dots, w_n)$.

Στόχος είναι η ταξινόμηση των N δειγμάτων εκπαίδευσης $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$ σε δύο κλάσεις. Εδώ γίνεται διαφοροποίηση των μεγεθών d_i που είναι οι επιθυμητές αποκρίσεις από τα y_i που είναι οι έξοδοι του δικτύου στα διάφορα δεδομένα εκπαίδευσης. Όπως και στα προηγούμενα οι δύο κλάσεις χαρακτηρίζονται από $d_i = \pm 1$ άρα και $y_i = \pm 1$. Συνεπώς η συνάρτηση φ πρέπει να έχει δύο δυνατές εξόδους, $+1$ και -1. Μία τέτοια συνάρτηση είναι η συνάρτηση προσήμου που παράγει $+1$ άν η είσοδος είναι θετική και -1 αν είναι αρνητική. Σε αυτή την περίπτωση η έξοδος του δικτύου είναι

$$y = sgn(\mathbf{w}^T\mathbf{x} + b) = \begin{cases} +1\ , & \alpha\nu\ \ \mathbf{w}^T\mathbf{x} + b > 0 \\ -1\ , & \alpha\nu\ \ \mathbf{w}^T\mathbf{x} + b < 0 \end{cases}$$

δηλαδή το μοντέλο perceptron παράγει στην ουσία ένα υπερεπίπεδο που διαχωρίζει τα δεδομένα σε δύο κλάσεις. Τα $\mathbf{w}$ και b είναι οι παράμετροι που πρέπει να προσδιοριστούν και υπολογίζονται από τον αλγόριθμο μάθησης του perceptron ή απλούστερα τον αλγόριθμο perceptron. Στα προηγούμενα είδαμε τρόπους κατασκευής υπερεπιπέδων για ταξινόμηση γραμμικά διαχωρίσιμων δεδομένων. Είδαμε την προσέγγιση της ελαχιστοποίησης του μέσου τετραγωνικού σφάλματος στον αλγόριθμο γραμμικής παλινδρόμησης και την προσέγγιση του αλγορίθμου SVM. Τώρα θα δούμε την προσέγγιση των νευρωνικών δικτύων στο πρόβλημα βελτιστοποίησης. Η προσέγγιση αυτή θα γενικευτεί στη συνέχεια σε δίκτυα πολλών επιπέδων για να προκύψει ο αλγόριθμος οπισθοδιάδοσης σφάλματος.

Ο αλγόριθμος perceptron

Το perceptron εκπαιδεύεται με τον εξής τρόπο: αρχικοποιούνται οι παράμετροι προς εκτίμηση σε κάποια τυχαία τιμή και στη συνέχεια οι παράμετροι αυτές ενημερώνονται σύμφωνα με κάποιο κανόνα ενημέρωσης μετά από κάθε διάνυσμα εκπαίδευσης. Κάθε σημείο εκπαίδευσης λοιπόν τροφοδοτείται στο δίκτυο και προκύπτει μία έξοδος. Αν η έξοδος συμφωνεί με την επιθυμητή, το διάνυσμα βαρών και η πόλωση παραμένουν αμετάβλητα. Αν η έξοδος του δικτύου και η επιθυμητή δεν είναι ίδιες, παράμετροι και πόλωση ενημερώνονται σε μία νέα τιμή. Αυτή η νέα τιμή χρησιμοποιείται στην τροφοδότηση του επόμενου διανύσματος εκπαίδευσης. Η διαδικασία τελειώνει όταν τροφοδοτηθούν στο δίκτυο όλα τα πρότυπα εκπαίδευσης. Συχνά όμως τα δεδομένα τροφοδοτούνται περισσότερες φορές στο δίκτυο και κάθε πλήρης κύκλος όλου του συνόλου εκπαίδευσης χαρακτηρίζεται *εποχή*. Έτσι το perceptron μπορεί να εκπαιδεύεται για πολλές εποχές. Επίσης κριτήριο τερματισμού της διαδικασίας εκπαίδευσης μπορεί να αποτελεί και ο μηδενισμός του σφάλματος ταξινόμησης. Δηλαδή η εκπαίδευση συνεχίζεται για πολλές εποχές μέχρις ότου τα σημεία εκπαίδευσης ταξινομούνται όλα σωστά.

Οι παράμετροι προς προσδιορισμό σε κάποιο χρονικό βήμα t είναι οι $\mathbf{w}(t) = [w_1(t), w_2(t), \dots, w_n(t)]^T$ και $b(t)$. Οι δύο παράμετροι ενσωματώνονται σε μία, με μία τεχνική που είδαμε και προηγουμένως, με επαύξηση των διανυσμάτων εκπαίδευσης στην αρχή τους με $+1$. Πλέον έχουμε

$$D = \{(\mathbf{x}_i, d_i)\}_{i=1}^{N} \text{ το σύνολο εκπαίδευσης}$$

$$\mathbf{x}_i = [+1, x_1, x_2, \dots, x_n]\,, \gamma\iota\alpha\ i = 1,2, \dots, N$$

$$\mathbf{w}(t) = [b(t), w_1(t), w_2(t), \dots, w_n(t)]^T$$

$$v_i(t) = \mathbf{w}^T(t)\mathbf{x}_i$$

$$y_i(t) = sgn[v_i(t)] = sgn[\mathbf{w}^T(t)\mathbf{x}_i]$$

και ο κανόνας ενημέρωσης γίνεται

$$\begin{gathered}\mathbf{w}(t+1) = \mathbf{w}(t)\,, \alpha\nu\ y_i = d_i \\ \mathbf{w}(t+1) = \mathbf{w}(t) + \eta\mathbf{x}_i\,, \alpha\nu\ y_i(t) = -1\ \kappa\alpha\iota\ d_i = 1 \\ \mathbf{w}(t+1) = \mathbf{w}(t) - \eta\mathbf{x}_i\,, \alpha\nu\ y_i(t) = 1\ \kappa\alpha\iota\ d_i = -1\end{gathered}$$

ή πιο απλά

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta[d_i - y_i(t)]\mathbf{x}_i$$

Η παράμετρος η καλείται *ρυθμός μάθησης* (*learning rate*) και καθορίζει την ευαισθησία με την οποία πραγματοποιούνται αλλαγές στα βάρη. Μεγάλες τιμές ρυθμού μάθησης αναγκάζουν το δίκτυο να ταλαντώνεται γύρω από τη λύση χωρίς να την προσεγγίζει ενώ αντίθετα μικρές τιμές επιβραδύνουν τη διαδικασία μάθησης. Η παράμετρος η είναι πάντα θετική και παίρνει τιμές $0 < \eta \leq 1$. Μπορεί να διατηρείται σταθερή κατά τη διάρκεια της εκπαίδευσης αλλά συνήθως ελαττώνεται μετά από κάθε εποχή.

Ο ψευδοκώδικας του αλγορίθμου perceptron είναι ο παρακάτω

1. *Αρχικοποίησε το διάνυσμα βαρών* $\mathbf{w}$. Συνήθως επιλέγεται $\mathbf{w}(0) = \mathbf{0}$ ή τυχαίες μικρές τιμές για τα επιμέρους βάρη του $\mathbf{w}$. Υπενθυμίζουμε ότι πλέον η πόλωση έχει ενσωματωθεί στο διάνυσμα βαρών.

2. *Αρχικοποίησε το ρυθμό μάθησης* η.

3. *Αρχικοποίησε το σφάλμα στο 0.* $err(0) = 0$

4. *Για κάθε διάνυσμα εκπαίδευσης* $\mathbf{x}_i$ *του συνόλου* D *ακολούθησε τα παρακάτω βήματα:*

   - *Ενεργοποίηση*: Υπολόγισε την έξοδο του δικτύου

   $$y_i(t) = sgn[\mathbf{w}^T(t)\mathbf{x}_i]$$

   - *Ενημέρωση*: Ενημέρωσε τα βάρη με τον κανόνα

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta[d_i - y_i(t)]\mathbf{x}_i$$

- *Ενημέρωση του σφάλματος*:

$$err(t+1) = err(t) + |d_i - y_i(t)|$$

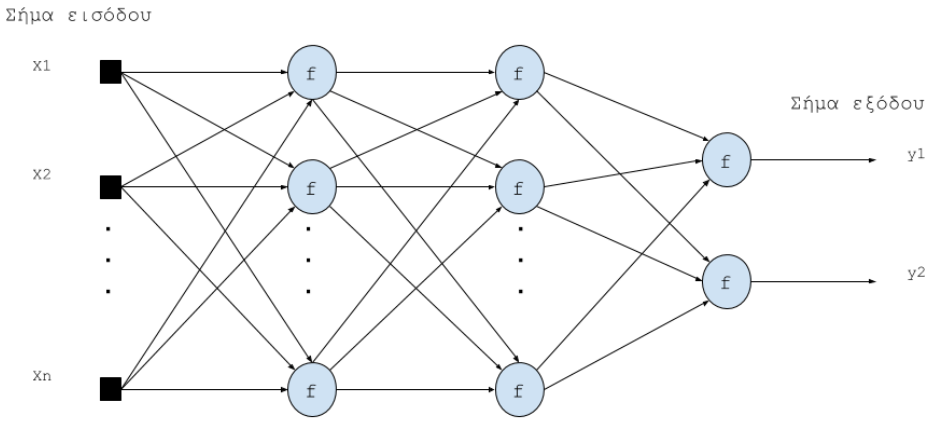
- *Αύξησε το* t *κατά 1.*

4. *Εάν* $\frac{1}{N} err(t) < \gamma$ *ή* $\frac{t}{N} = max\_epochs$ *τερμάτισε. Αλλιώς μηδένισε το λάθος, ενημέρωσε το ρυθμό μάθησης και επανέλαβε το 3.*

Ο παραπάνω αλγόριθμος είναι μία πολύ κοινή υλοποίηση του Perceptron που χρησιμοποιεί και κριτήρια τερματισμού όπως ο μέγιστος αριθμός εποχών και το ελάχιστο σφάλμα. Πριν προχωρήσουμε στο πολυεπίπεδο perceptron αναφέρουμε ότι ο αλγόριθμος perceptron συγκλίνει αποδεδειγμένα σε λύση μετά από πεπερασμένο αριθμό βημάτων ή εποχών. Για τη λεπτομερή απόδειξη της σύγκλισης παραπέμπουμε τον αναγνώστη στο [6].

3.5.2 Το Perceptron Πολλών Επιπέδων

Το Perceptron Πολλών Επιπέδων είναι η γενικότερη περίπτωση του δικτύου εμπρόσθιας διάδοσης. Αποτελεί γενίκευση του απλού perceptron που είδαμε στην προηγούμενη ενότητα και η βασική διαφορά έγκειται στον αριθμό επιπέδων. Στο perceptron του Rosenblatt είχαμε στην πράξη ένα επίπεδο, αυτό της εξόδου με ένα μόνο νευρώνα ενώ το perceptron πολλών επιπέδων προσθέτει τουλάχιστον ένα ακόμα επίπεδο ανάμεσα στην είσοδο και στην έξοδο, που καλείται *κρυφό επίπεδο* (*hidden layer*). Με ένα κρυφό επίπεδο έχουμε το perceptron δύο επιπέδων, με δύο κρυφά επίπεδα, το perceptron τριών επιπέδων κ.ο.κ. Ωστόσο η γενίκευση είναι άμεση, γιαυτό και εδώ εξετάζεται το perceptron πολλών επιπέδων. Η αρχιτεκτονική ενός MLP με δύο κρυφά επίπεδα δίνεται στο παρακάτω σχήμα.



Σχήμα 3.6

Παρατηρούμε το σήμα εισόδου που τροφοδοτείται στην αρχή, τα δύο κρυφά επίπεδα και τελικά το επίπεδο εξόδου που έχει δύο νευρώνες. Στο επίπεδο εξόδου ο αριθμός νευρώνων ταυτίζεται με το πλήθος των κλάσεων. Συνεπώς το παραπάνω δίκτυο ταξινομεί δεδομένα ενός χώρου n-διαστάσεων σε δύο κλάσεις. Στα κρυφά επίπεδα ο αριθμός νευρώνων είναι αυθαίρετος και αποτελεί επιλογή του σχεδιαστή. Ουσιαστικά κατά τη μετάβαση από την είσοδο στο πρώτο κρυφό επίπεδο τα δεδομένα του αρχικού χώρου χαρακτηριστικών μετασχηματίζονται σε ένα νέο χώρο του οποίου η διάσταση είναι ίση με τον αριθμό νευρώνων στο επίπεδο αυτό. Ομοίως κατά τη μετάβαση από το πρώτο στο δεύτερο κρυφό επίπεδο τα δεδομένα μετασχηματίζονται από το χώρο του πρώτου στο χώρο του δεύτερου και τελικά στο χώρο της εξόδου κατά την τελική μετάβαση. Αν υποθέσουμε δηλαδή h_1 νευρώνες στο πρώτο κρυφό επίπεδο και h_2 νευρώνες στο δεύτερο τα δεδομένα μετασχηματίζονται από τον αρχικό χώρο n διαστάσεων, σε χώρο h_1 διαστάσεων και στη συνέχεια σε χώρο h_2 διαστάσεων. Σαν σχεδιαστές δεν έχουμε καθαρή εικόνα των χώρων αυτών ούτε γνωρίζουμε από πριν τη διάσταση που πρέπει να έχουν οι χώροι αυτοί για να ταξινομηθούν σωστά τα δεδομένα, αλλά με την παραπάνω αρχιτεκτονική εμπιστευόμαστε το δίκτυο να ανακαλύψει κρυφές δομές και patterns στα δεδομένα που δεν είναι εμφανή στον αρχικό χώρο. Κάθε βέλος-σύναψη συνοδεύεται από κάποιο βάρος και επίσης κάθε νευρώνας έχει και μία βοηθητική πόλωση που δεν φαίνεται στο σχήμα. Το perceptron πολλών επιπέδων είναι στην πράξη πολλά απλά perceptrons διασυνδεδεμένα προς τα εμπρός και ταξινομημένα σε επίπεδα. Όπως και στα προηγούμενα, έτσι και εδώ, το δίκτυο εκπαιδεύται προσαρμόζοντας τα συναπτικά βάρη του και τις πολώσεις του. Η εκπαίδευση γίνεται με τον αλγόριθμο οπίσθιας διάδοσης σφάλματος που θα δούμε στη συνέχεια.

Το δίκτυο MLP αποτελεί ένα ισχυρό μοντέλο που μπορεί να διαχωρίζει δεδομένα που δεν είναι γραμμικά διαχωρίσιμα. Όσο περισσότερα τα κρυφά επίπεδα και όσο περισσότεροι οι νευρώνες, τόσο πιο πολύπλοκες είναι οι αναπαραστάσεις των αρχικών δεδομένων και τόσο μεγαλύτερη η δύναμη του δικτύου στο να ταξινομεί δεδομένα.

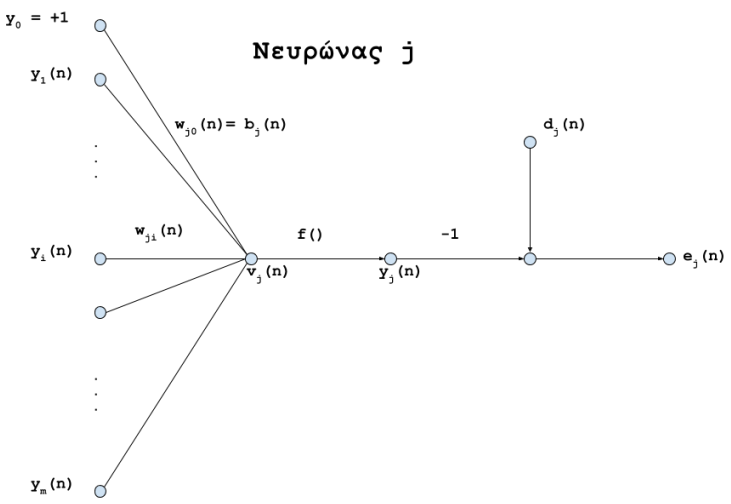
Η συνάρτηση f που σημειώνεται στο σχήμα είναι η μη γραμμική συνάρτηση ενεργοποίησης του νευρώνα. Όπως είδαμε στο απλό perceptron ο σκοπός της συνάρτησης ήταν να αντιστοιχεί τη συνεχή είσοδο σε κβαντισμένες εξόδους -1 και 1 και για το λόγο αυτό χρησιμοποιήσαμε τη συνάρτηση προσήμου $sgn(x)$. Η ίδια λογική ισχύει και εδώ, θέλουμε δηλαδή κάθε νευρώνας, είτε εξόδου είτε κρυφού επιπέδου, να αντιστοιχεί την είσοδο του στις τιμές -1 και 1 ή 0 και 1. Όπως όμως θα δείξουμε στη συνέχεια ο αλγόριθμος Backpropagation απαιτεί η συνάρτηση αυτή να είναι διαφορίσιμη. Συνεπώς δεν μπορούμε να χρησιμοποιήσουμε την ασυνεχή $sgn(x)$. Στην πράξη χρησιμοποιούνται συναρτήσεις που προσεγγίζουν τη συμπεριφορά της συνάρτησης προσήμου αλλά είναι συνεχείς. Τέτοιες συναρτήσεις καλούνται *σιγμοειδείς* (*sigmoid functions*). Για έξοδο 0 και 1 χρησιμοποιείται η *λογιστική συνάρτηση* (*logistic function*)

$$f(x) = \frac{1}{1 + \exp(-ax)} \ , a > 0$$

και για έξοδο -1 και 1 η *συνάρτηση υπερβολικής εφαπτομένης* (*hyperbolic tangent function*)

$$f(x) = a \tanh(bx) \ , a, b > 0$$

Ας αναλύσουμε τώρα το δίκτυο μαθηματικά. Για τις ανάγκες της ανάλυσης θα αλλάξουμε ελαφρά τους συμβολισμούς. Θεωρούμε λοιπόν ότι το σύνολο εκπαίδευσης αποτελείται από N πρότυπα και κάθε πρότυπο δηλώνεται ως $\mathbf{x}(n)$ με $n = 1,2,\dots,N$ και συνοδεύεται από επιθυμητή απόκριση $\mathbf{d}(n)$. Η απόκριση σε αντίθεση με τα προηγούμενα είναι διάνυσμα γιατί στη γένικη περίπτωση το νευρωνικό δίκτυο έχει περισσότερες των δύο εξόδων και μπορεί να ταξινομήσει σε παραπάνω από δύο κλάσεις. Το σύνολο εκπαίδευσης λοιπόν είναι $D = \{(\mathbf{x}(n), \mathbf{d}(n))\}_{n=1}^{N}$. Ας επικέντρωθούμε τώρα σε ένα νευρώνα του δικτύου. Το σχήμα 3.7 απεικονίζει ένα νευρώνα j του δικτύου να τροφοδοτείται από σήματα που παράγει ένα επίπεδο, κρυφό ή εισόδου, στα αριστερά του.



Σχήμα 3.7

Τα μεγέθη $y_i(n)$ για $i = 0,1,\dots,m$ είναι οι έξοδοι των νευρώνων του προηγούμενου επιπέδου κατά την εξέταση του διανύσματος $\mathbf{x}(n)$. Για κάθε είσοδο, δηλαδή για κάθε n η επαύξηση y_0 παραμένει σταθερή. Τα $w_{ji}(n)$ είναι τα βάρη που συνδέουν το προηγούμενο επίπεδο με τον νευρώνα μας για τη «χρονική στιγμη» n. Σαν συνέπεια της επαύξησης των προτύπων εισόδου το $w_{j0}(n)$ είναι στην πράξη η πόλωση του νευρώνα $b_j(n)$. Το σήμα $v_j(n)$ είναι το τοπικό πεδίο στον νευρώνα j πάνω στο οποίο εφαρμόζεται η μη γραμμική συνάρτηση για να προκύψει η έξοδος $y_j(n)$.

Όπως είδαμε και στην πιο απλή περίπτωση του perceptron το τοπικό πεδίο δίνεται από τη σχέση

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n)$$

και η έξοδος

$$y_j(n) = f\left(v_j(n)\right) = f\left(\sum_{i=0}^{m} w_{ji}(n) y_i(n)\right)$$

Στη συνέχεια, στο σχήμα, δηλώνεται ο προσδιορισμός του σφάλματος στο νευρώνα j δηλαδή

$$e_j(n) = d_j(n) - y_j(n)$$

Το σφάλμα αυτό όμως μπορεί να υπολογιστεί μόνο αν ο νευρώνας είναι νευρώνας εξόδου αφού εκεί υπάρχει διαθέσιμη η επιθυμητή απόκριση. Θα επανέλθουμε σε αυτό στη συνέχεια. Για την ώρα ας ορίσουμε το συνολικό σφάλμα που παρουσιάζει το δίκτυο με

απώτερο σκοπό να το ελαχιστοποιήσουμε. Η στιγμιαία ενέργεια του σφάλματος σε ένα νευρώνα j ορίζεται ως

$$\mathcal{E}_j(n) = \frac{1}{2} e_j^2(n)$$

και η συνολική στιγμιαία ενέργεια σφάλματος

$$\mathcal{E}(n) = \sum_{j \in C} \mathcal{E}_j(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n)$$

όπου C το σύνολο των νευρώνων στην έξοδο. Για όλο το σύνολο εκπαίδευσης η μέση ενέργεια σφάλματος ή μέσο τετραγωνικό σφάλμα

$$E = \frac{1}{N} \sum_{n=1}^{N} \mathcal{E}(n) = \frac{1}{2N} \sum_{n=1}^{N} \sum_{j \in C} e_j^2(n)$$

Το νευρωνικό δίκτυο εκπαιδεύεται, τροποποιώντας τα βάρη του w_{ij} με σκοπό την ελαχιστοποίηση του παραπάνω συνολικού σφάλματος. Στην πράξη τροποποιεί τα βάρη του στην κατεύθυνση που ελαχιστοποιεί το στιγμιαίο σφάλμα για να ελαχιστοποιηθεί τελικά το συνολικό. Συγκεκριμένα μετά από την προς τα εμπρός διάδοση (feed-forward) ενός διανύσματος $\mathbf{x}(n)$ αλλάζει τα βάρη του για το επόμενο διάνυσμα εκπαίδευσης προς την κατεύθυνση που ελαχιστοποιεί τη στιγμιαία ενέργεια σφάλματος $\mathcal{E}(n)$. Αυτός είναι ο αλγόριθμος Backpropagation που θα δούμε στη συνέχεια.

3.5.3 Ο Αλγόριθμος Backpropagation

Η ιδέα του αλγορίθμου μάθησης, όπως δώθηκε συνοπτικά και προηγουμένως, είναι να αλλάζουμε τα βάρη μετά από κάθε πρότυπο εισόδου προς την κατεύθυνση που μειώνεται η συνολική στιγμιαία ενέργεια σφάλματος σε όλους τους νευρώνες εξόδου. Μαθηματικά αυτό αποτυπώνεται στην εξίσωση

$$w_{ji}(n+1) = w_{ji}(n) - \eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)}$$

όπου η είναι ο ρυθμός μάθησης. Η εξίσωση αφορά ένα νευρώνα εξόδου j ενώ το βάρος $w_{ji}(n)$ τη σύναψη από ένα νευρώνα i του προηγούμενου επιπέδου προς τον j. Η παράγωγος στο δεξί μέλος της εξίσωσης μπορεί να αναλυθεί με τον κανόνα αλυσίδας.

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \cdot \frac{\partial e_j(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} \cdot \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

με

$$\mathcal{E}(n) = \frac{1}{2}\sum_{j\in C} e_j^2(n) \quad \Rightarrow \quad \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} = e_j(n)$$

$$e_j(n) = d_j(n) - y_j(n) \quad \Rightarrow \quad \frac{\partial e_j(n)}{\partial y_j(n)} = -1$$

$$y_j(n) = f\left(v_j(n)\right) \quad \Rightarrow \quad \frac{\partial y_j(n)}{\partial v_j(n)} = f'\left(v_j(n)\right)$$

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n) \quad \Rightarrow \quad \frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n)$$

συνεπώς

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = -e_j(n) f'\left(v_j(n)\right) y_i(n)$$

Αντικαθιστούμε στην τελευταία εξίσωση το μέγεθος $e_j(n)f'(v_j(n))$ με μία νέα μεταβλητή $\delta_j(n)$, που καλείται τοπική κλίση. Έχουμε λοιπόν

$$w_{ji}(n+1) = w_{ji}(n) + \eta \delta_j(n) y_i(n)$$

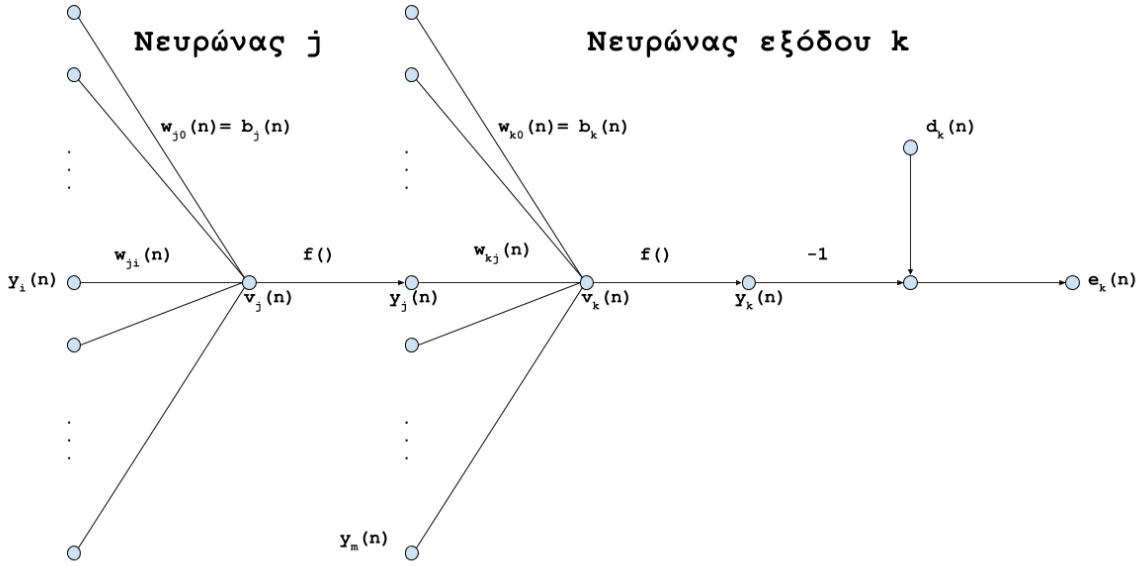
$$\text{με } \delta_j(n) = e_j(n) f'(v_j(n))$$

Με την παραπάνω εξίσωση λοιπόν μπορούμε να ενημερώσουμε το βάρος w_{ji} που καταλήγει σε ένα νευρώνα εξόδου j από ένα νευρώνα i προηγούμενου επιπέδου. Όλα τα μεγέθη της εξίσωσης είναι γνωστά αφού όπως είπαμε οι νευρώνες εξόδου τροφοδοτούνται με τις επιθυμητές αποκρίσεις $d_j(n)$ με τη βοήθεια των οποίων υπολογίζουμε τα σφάλματα $e_j(n)$. Τι γίνεται όμως με τους νευρώνες στα κρυφά επίπεδα, που δεν έχουν επιθυμητή απόκριση; Η παραπάνω εξίσωση δε μπορεί να εφαρμοστεί σε αυτή την περίπτωση και για αυτό το λόγο καταφεύγουμε στην οπισθοδιάδοση σφάλματος. Για να γίνει κατανοητή η έννοια αυτή, ας ξεκινήσουμε με την αναπαράσταση ενός κρυφού νευρώνα. Στο σχήμα 3.8 φαίνεται ένας κρυφός νευρώνας j που συνδέεται προς τα εμπρός με ένα νευρώνα εξόδου k και προς τα πίσω με ένα νευρώνα i.

Η λογική είναι παρόμοια με τα προηγούμενα. Θέλουμε να ενημερώσουμε το βάρος $w_{ji}(n)$ στη κατεύθυνση που ελαχιστοποιείται η συνολική στιγμιαία ενέργεια σφάλματος. Δηλαδή όπως και πριν

$$w_{ji}(n+1) = w_{ji}(n) - \eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)}$$

Τώρα όμως η σχέση ανάμεσα στην $\mathcal{E}(n)$ και το βάρος $w_{ji}(n)$ δεν είναι τόσο απλή. Στην προηγούμενη περίπτωση κάθε βάρος $w_{ji}(n)$ επηρέαζε την έξοδο, άρα και το σφάλμα, μόνο ενός νευρώνα εξόδου j. Τώρα το βάρος $w_{ji}(n)$ συμμετέχει στην έξοδο ενός νευρώνα



Σχήμα 3.8

j στο κρυφό επίπεδο ο οποίος όμως στη συνέχεια συνδέεται με όλους τους κόμβους εξόδου επηρεάζοντας τα σφάλματα σε όλους τους νευρώνες στην έξοδο. Μαθηματικά αυτό εκφράζεται ως

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial}{\partial w_{ji}(n)}\left[\frac{1}{2}\sum_{k\in C} e_k^2(n)\right] = \frac{1}{2}\sum_{k\in C}\frac{\partial e_k^2(n)}{\partial w_{ji}(n)} = \frac{1}{2}\sum_{k\in C} 2e_k(n)\frac{\partial e_k(n)}{\partial w_{ji}(n)} = \sum_{k\in C} e_k(n)\frac{\partial e_k(n)}{\partial w_{ji}(n)}$$

όπου C είναι το σύνολο των νευρώνων στην έξοδο όπως το ορίσαμε και προηγουμένως. Πλέον φαίνεται πιο καθαρά η διαφορά. Στην παραγώγιση του αθροίσματος των τετραγωνικών σφαλμάτων στις εξόδους ως προς το βάρος, στην προηγούμενη περίπτωση το βάρος επηρέαζε το σφάλμα μόνο σε μία έξοδο οπότε η εξίσωση δεν είχε άθροισμα. Τώρα όμως το βάρος συμμετέχει σε όλες τις εξόδους γι'αυτό και προκύπτει το άθροισμα.

Η παράγωγος της τελευταίας σχέσης αναλύεται και πάλι με τον κανόνα της αλυσίδας

$$\frac{\partial e_k(n)}{\partial w_{ji}(n)} = \frac{\partial e_k(n)}{\partial y_k(n)}\cdot\frac{\partial y_k(n)}{\partial v_k(n)}\cdot\frac{\partial v_k(n)}{\partial y_j(n)}\cdot\frac{\partial y_j(n)}{\partial v_j(n)}\cdot\frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

$$e_k(n) = d_k(n) - y_k(n) \;\Rightarrow\; \frac{\partial e_k(n)}{\partial y_k(n)} = -1$$

$$y_k(n) = f\big(v_k(n)\big) \;\Rightarrow\; \frac{\partial y_k(n)}{\partial v_k(n)} = f'\big(v_k(n)\big)$$

$$v_k(n) = \sum_{j=0}^{m} w_{kj}(n)y_j(n) \;\Rightarrow\; \frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n)$$

$$y_j(n) = f\left(v_j(n)\right) \;\Rightarrow\; \frac{\partial y_j(n)}{\partial v_j(n)} = f'\left(v_j(n)\right)$$

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n) \;\Rightarrow\; \frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n)$$

Συνεπώς

$$\frac{\partial e_k(n)}{\partial w_{ji}(n)} = -f'\big(v_k(n)\big) w_{kj}(n)\, f'\left(v_j(n)\right) y_i(n)$$

και

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = -\sum_{k\in C} e_k(n) f'\big(v_k(n)\big)\, w_{kj}(n)\, f'\left(v_j(n)\right) y_i(n)$$

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = -\, f'\left(v_j(n)\right) y_i(n) \sum_{k\in C} e_k(n) f'\big(v_k(n)\big)\, w_{kj}(n)$$

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} == -\, f'\left(v_j(n)\right) y_i(n) \sum_{k\in C} \delta_k(n)\, w_{kj}(n)$$

Αν λοιπόν ορίσουμε την τοπική κλίση $\delta_j(n)$ του κρυφού νευρώνα ως

$$\delta_j(n) = f'(v_j(n)) \sum_{k\in C} \delta_k(n)\, w_{kj}(n)$$

θα πάρουμε

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = -y_i(n)\delta_j(n)$$

και η έκφραση για την ανανέωση των βαρών θα είναι η ίδια ανεξαρτήτως αν ο νευρώνας είναι στην έξοδο ή σε κρυφό επίπεδο. Δηλαδή στη γενική περίπτωση το βάρος w_{ji} ενημερώνεται με τον κανόνα

$$w_{ji}(n+1) = w_{ji}(n) + \eta\delta_j(n) y_i(n)$$

όπου

$$\delta_j(n) = e_j(n) f'(v_j(n))\,, \textit{αν ο νευρώνας } j \textit{ είναι στην έξοδο}$$

$$\delta_j(n) = f'\left(v_j(n)\right) \sum_{k\in C} \delta_k(n)\, w_{kj}(n)\;\,, \textit{αν ο νευρώνας } j \textit{ είναι κρυφός}$$

Οι παραπάνω εξισώσεις ισχύουν για κάθε νευρώνα στο δίκτυο. Για την απόδειξη των σχέσεων περιοριστήκαμε σε νευρώνες στην έξοδο και σε νευρώνες του κρυφού επιπέδου που προηγείται της εξόδου. Ωστόσο η λογική είναι η ίδια και οι σχέσεις γενικεύονται για νευρώνες κάθε επιπέδου. Αν για παράδειγμα έχουμε ένα δίκτυο με τρία κρυφά επίπεδα και ένα επίπεδο εξόδου, για τους νευρώνες του δεύτερου κρυφού επιπέδου ισχύουν οι ίδιες σχέσεις όπου οι τοπικές κλίσεις $\delta_k(n)$ αναφέρονται στους νευρώνες του τρίτου κρυφού επιπέδου. Αρχικά λοιπόν προσδιορίζονται οι τοπικές κλίσεις στο επίπεδο εξόδου και αυτές διαδίδονται αναδρομικά προς τα πίσω στο δίκτυο επίπεδο προς επίπεδο. Με τη βοήθεια των κλίσεων στο επίπεδο εξόδου υπολογίζονται οι κλίσεις στο τελευταίο κρυφό επίπεδο, έπειτα με τη βοήθεια αυτών οι κλίσεις στο προτελευταίο επίπεδο και η διαδικασία συνεχίζεται ώσπου να φτάσουμε στο πρώτο κρυφό επίπεδο. Αυτή είναι η φάση της οπισθοδιάδοσης σφάλματος και είναι η δεύτερη φάση του αλγόριθμου Backpropagation. Η πρώτη είναι η απλά η εφαρμογή των διανυσμάτων εισόδου στο δίκτυο, με σταθερά τα βάρη, για τον υπολογισμό των $y_i(n)$ για κάθε νευρώνα i.

Παρατηρήσεις

Στους παραπάνω υπολογισμούς υπεισέρχεται η παράγωγος της συνάρτησης f. Για αυτό το λόγο απαιτήσαμε στην αρχή η συνάρτηση αυτή να είναι συνεχής και διαφορίσιμη. Όπως αναφέρθηκε οι συνηθέστερες επιλογές είναι η συνάρτηση υπερβολικής εφαπτομένης και η λογιστική συνάρτηση.

Η διαδίκασια που περιγράψαμε, όπου τα βάρη ενημερώνονται μετά από κάθε πρότυπο εισόδου, καλείται *on-line μάθηση*. Η εναλλακτική προσέγγιση καλείται *μαζική μάθηση* και σε αυτή το δίκτυο προπονείται σε εποχές. Κάθε εποχή είναι μία πλήρης εφαρμογή του συνόλου εκπαίδευσης στο δίκτυο και τα βάρη ενημερώνονται μετά από κάθε εποχή.

Ο ρυθμός μάθησης όπως και στο απλό perceptron μπορεί να είναι σταθερός ή να μειώνεται συνήθως μετά από κάθε εποχή. Μεγάλες τιμές μπορεί να κάνουν το δίκτυο ασταθές ενώ μικρές τιμές επιβραδύνουν τη διαδικασία μάθησης. Για την αποφυγή της αστάθειας πολλές φορές χρησιμοποιείται ένας *όρος ορμής* στον κανόνα αναπροσαρμογής των βαρών όπως φαίνεται στην παρακάτω σχέση

$$w_{ji}(n+1) = w_{ji}(n) + \alpha w_{ji}(n-1) + \eta\delta_j(n)y_i(n)$$

όπου α παράμετρος που καθορίζεται από τον εκπαιδευτή και καλείται *σταθερά ορμής*.

Σύνοψη του αλγορίθμου Backpropagation

Ο αλγόριθμος εκπαίδευσης ενός νευρωνικού δικτύου πολλών επιπέδων πάνω στα δεδομένα $D = \{(\mathbf{x}(n), \mathbf{d}(n))\}_{n=1}^{N}$ δίνεται συνοπτικά ως εξής:

1. *Αρχικοποίηση των βαρών*. Τα βάρη μεταξύ όλων των νευρώνων αρχικοποιούνται τυχαία. Συνήθως επιλέγεται ομοιόμορφη κατανομή μηδενικής μέσης τιμής και κατάλληλης διασποράς. Αν υπάρχει πρότερη γνώση, δηλαδή το δίκτυο έχει εκπαιδευτεί στο παρελθόν σε δεδομένα του ίδιου προβλήματος, χρησιμοποιούνται τα τελικά βάρη που προέκυψαν από την προηγούμενη εκπαίδευση.

2. *Η αρχή μίας εποχής*. Για κάθε πρότυπο εκπαίδευσης $\boldsymbol{x}(n)$ εκτελούνται οι δύο φάσεις:

2.1 *Διάδοση προς τα εμπρός*. Το πρότυπο εισόδου τροφοδοτείται στο δίκτυο από αριστερά. Με βάση τα υπάρχοντα βάρη πραγματοποιούνται οι υπολογισμοί προς τα εμπρός, επίπεδο προς επίπεδο, για όλους τους νευρώνες. Τελικά προκύπτουν στο επίπεδο εξόδου οι έξοδοι του δικτύου. Για κάθε νευρώνα i αποθηκεύονται οι τιμές της εξόδου του $y_i(n)$ και του πεδίου $v_i(n)$.

2.2 *Διάδοση προς τα πίσω*. Από τις εξόδους και τις επιθυμητές αποκρίσεις στο επίπεδο εξόδου, υπολογίζονται τα σφάλματα $e_j(n)$, $j \in C$. Προσδιορίζονται με τη σειρά τους οι τοπικές κλίσεις στο επίπεδο εξόδου από τη σχέση

$$\delta_j(n) = e_j(n) f'\left(v_j(n)\right)$$

και με τη βοήθεια αυτών, οι τοπικές κλίσεις σε όλα τα επίπεδα από δεξιά προς τα αριστερά με τη σχέση

$$\delta_j(n) = f'\left(v_j(n)\right) \sum_{k \in H} \delta_k(n)\, w_{kj}(n)$$

όπου H το επίπεδο στα δεξιά του επιπέδου που βρίσκεται ο νευρώνας j. Αφού υπολογιστούν και αποθηκευτούν οι τοπικές κλίσεις για όλους τους νευρώνες, τα βάρη ενημερώνονται με τον κανόνα

$$w_{ji}(n+1) = w_{ji}(n) + \alpha w_{ji}(n-1) + \eta \delta_j(n) y_i(n)$$

όπου α η σταθερά ορμής και η ο ρυθμός μάθησης.

3. *Το τέλος μίας εποχής*. Αν το συνολικό σφάλμα σε όλο το δείγμα εκπαίδευσης είναι μικρότερο από μία τιμή γ ή αν συμπληρώθηκε ο μέγιστος αριθμός εποχών max_epochs η διαδικασία τερματίζει. Διαφορετικά επιστρέφει στο βήμα 2.

Στο κεφάλαιο 4 του [6] παρουσιάζεται εκτενής ανάλυση του perceptron πολλαπλών επιπέδων και του βασικού ζητήματος της προσαρμογής του δικτύου (tuning), δηλαδή του καθορισμού των παραμέτρων για την αποφυγή overfitting και την επιτυχή εφαρμογή σε πραγματικά δεδομένα .

3.6 Συνελικτικά Νευρωνικά Δίκτυα

Τα συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks - CNNs) αποτελούν ειδική μορφή του δικτύου εμπρόσθιας διάδοσης, και εμπνέονται από τους μηχανισμούς επεξεργασίας της οπτικής πληροφορίας, που συναντώνται στους ζωντανούς οργανισμούς. Δημιουργήθηκαν αρχικά για εφαρμογές υπολογιστικής όρασης, ωστόσο τα τελευταία χρόνια εφαρμόζονται επιτυχώς σε διάφορους κλάδους της επιστήμης των υπολογιστών όπως

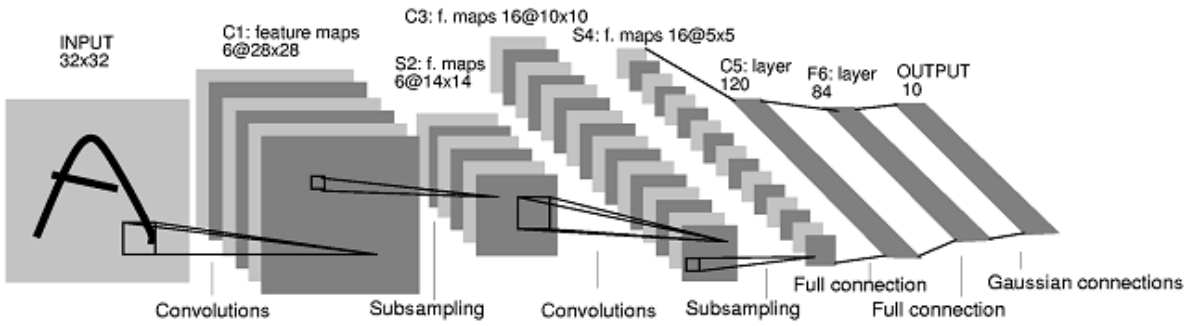
στην επεξεργασία φωνής και σε υποπεριοχές της επεξεργασίας φυσικού λόγου, όπως η ανάλυση συναισθήματος. Το συνελικτικό δίκτυο οργανώνεται σε επίπεδα, και αντικαθιστά τις διασυνδέσεις εσωτερικού γινομένου με συνελίξεις και επίπεδα υποδειγματοληψίας (pooling ή subsampling layers).

Μία βασική διαφορά του συνελικτικού δικτύου από το perceptron πολλών επιπέδων είναι ότι εφαρμόζει τοπικό φιλτράρισμα (local filtering) αντί ολικού. Ας υποθέσουμε μία greyscale εικόνα με μέγεθος 32x32 pixels. Η εικόνα αυτή χαρακτηρίζεται από 1024 pixels και για να κατασκευαστεί ένα πολυεπίπεδο perceptron που μπορεί να εκπαιδεύεται σε τέτοιες εικόνες απαιτούνται 1024 νευρώνες στο επίπεδο εισόδου, αυθαίρετος αριθμός νευρώνων στο κρυφό ή στα κρυφά επίπεδα και τέλος κάποιος αριθμός νευρώνων στην έξοδο που συνήθως αντιστοιχεί στις κλάσεις στις οποίες θέλουμε να κατατάξουμε την πληροφορία της εικόνας. Καθώς το MLP είναι fully connected ο αριθμός των διασυνδέσεων γίνεται υπερβολικά μεγάλος και η εκπαίδευση ενός τέτοιου δικτύου με πολλά κρυφά επίπεδα ή σε μεγάλο αριθμό δεδομένων γίνεται πρακτικά αδύνατη. Στα συνελικτικά δίκτυα αντίθετα, κάθε νευρώνας συνδέεται με ένα υποσύνολο των νευρώνων του προηγούμενου επιπέδου. Ο συνδυασμός του υποσυνόλου των νευρώνων του προηγούμενου επιπέδου με τα βάρη συνιστά ουσιαστικά μία πράξη συνέλιξης.

Στο σχήμα 3.9 παρουσιάζεται η αρχιτεκτονική του δικτύου LeNet-5, ενός κλασσικού συνελικτικού δικτύου που αναγνωρίζει χειρόγραφους χαρακτήρες και χειρίζεται greyscale εικόνες μεγέθους 32x32.

Αρχικά παρατηρούμε την είσοδο. Στην είσοδο εφαρμόζονται 6 διαφορετικά φίλτρα και μέσω συνέλιξης προκύπτουν 6 διαφορετικές αναπαραστάσεις της αρχικής εικόνας, μεγέθους 28x28. Οι αναπαραστάσεις αυτές καλούνται χάρτες χαρακτηριστικών (feature maps) και οι διαστάσεις τους καθορίζονται από το μέγεθος των φίλτρων, το οποίο είναι επιλογή του σχεδιαστή. Στην πράξη υπεισέρχονται πολώσεις και μη γραμμικές συναρτήσεις στις αναπαραστάσεις αυτές, τις οποίες θα δούμε αναλυτικά στη συνέχεια.

Έπειτα εφαρμόζεται υποδειγματοληψία μέσω του pooling layer και προκύπτουν 6 χάρτες χαρακτηριστικών μικρότερου μεγέθους. Η διαδικασία συνεχίζεται με εφαρμογή 16 φίλτρων στην έξοδο της πρώτης υποδειγματοληψίας, εφαρμογή ενός επιπλέον επιπέδου υποδειγματοληψίας και τέλος εφαρμογή fully connected επιπέδων για την παραγωγή των εξόδων. Ας δούμε αναλυτικά τη μορφή των διαφόρων επιπέδων του συνελικτικού δικτύου.



Σχήμα 3.9

Είσοδος

Η είσοδος του συνελικτικού δικτύου είναι μία εικόνα διάστασης $W \times H$ όπου με W ορίζεται το πλάτος της εικόνας και με H το ύψος. Στην περίπτωση έγχρωμων εικόνων που αναλύονται σε τρία κανάλια R,G και B η είσοδος είναι ένας πίνακας $W \times H \times D$ όπου D είναι το βάθος της εικόνας με $D = 3$. Τέτοια αντικείμενα που αποτελούν πολυδιάστατες γενικεύσεις του απλού πίνακα καλούνται τανυστές (tensors).Η διάσταση του τανυστή συχνά αναφέρεται ως τάξη για να αποφεύγεται η σύγχυση με την απλή έννοια της διάστασης. Έτσι ο απλός πίνακας είναι ένας tensor τάξης 2 και το διάνυσμα ένας tensor τάξης 1. Γενικά η είσοδος του συνελικτικού δικτύου είναι ένας tensor τάξης 3 με διαστάσεις $W \times H \times D$ όπου $D = 1$ για greyscale εικόνες και $D = 3$ για RGB εικόνες.

Επίπεδο Συνέλιξης - Convolutional Layer

Το επίπεδο συνέλιξης περιλαμβάνει K φίλτρα κάθε ένα από τα οποία εφαρμόζεται στην είσοδο του επιπέδου. Αν θεωρήσουμε ότι η είσοδος έχει βάθος $D = 1$, είναι δηλαδή ένας απλός πίνακας ή μία εικόνα, τότε γίνεται διδιάστατη συνέλιξη του κάθε φίλτρου με αυτόν τον πίνακα και προκύπτουν K διαφορετικοί πίνακες ή εικόνες δηλαδή K χάρτες χαρακτηριστικών. Τα φίλτρα έχουν όλα τις ίδιες διαστάσεις και συνήθως είναι συμμετρικά, οπότε κάθε φίλτρο είναι διάστασης $F \times F$ με το F να καλείται απλά μέγεθος των φίλτρων. Ωστόσο τα K φίλτρα έχουν διαφορετικές παραμέτρους τις οποίες μαθαίνει το δίκτυο μέσα από την εκπαίδευση. Στην περίπτωση που η είσοδος έχει βάθος D μεγαλύτερο του 1, δηλαδή αποτελείται από D εικόνες το κάθε φίλτρο εφαρμόζεται με διαφορετικά βάρη σε κάθε μία από τις εικόνες και προκύπτουν D διαφορετικοί πίνακες οι οποίοι αθροίζονται για να προκύψει τελικά ένας πίνακας. Πάλι λοιπόν παράγονται K χάρτες χαρακτηριστικών. Γενικά όταν η είσοδος έχει διάσταση $W \times H \times D$, κάθε φίλτρο του επιπέδου συνέλιξης έχει διάσταση $F \times F$ για κάθε κανάλι εισόδου ή ισοδύναμα διάσταση $F \times F \times D$. Επίσης στον πίνακα που προκύπτει από κάθε φίλτρο εφαρμόζεται μία πόλωση b (σε κάθε στοιχείο) και έπειτα κάθε στοιχείο περνά από μία μη γραμμική συνάρτηση. Η συνάρτηση αυτή συνήθως είναι η ReLU (Rectified Linear Unit) που ορίζεται ως

$$ReLU(x) = \max(0, x)$$

Εκτός του μεγέθους και του πλήθους των φίλτρων, το συνελικτικό επίπεδο δέχεται δύο επιπλέον παραμέτρους. Οι παράμετροι αυτές είναι το βήμα της συνέλιξης S (stride) που καθορίζει τη μετακίνηση του φίλτρου στο σώμα της εικόνας (για την απλή συνέλιξη το βήμα είναι 1) και το μέγεθος του zero-padding P. Οι δύο παράμετροι όπως γίνεται κατανοητό επηρεάζουν το μέγεθος των εικόνων που προκύπτουν μετά την εφαρμογή των φίλτρων.

Συνοψίζοντας το συνελικτικό επίπεδο δέχεται στην είσοδο έναν tensor τάξης 3 και διάστασης $W_1 \times H_1 \times D_1$. Χαρακτηρίζεται από K φίλτρα κάθε ένα από τα οποία είναι διάστασης $F \times F$. Το μέγεθος F, το stride S ,το zero-padding P και το πλήθος φίλτρων K αποτελούν όλα υπερπαραμέτρους του επιπέδου. Το κάθε φίλτρο αποτελείται από τα βάρη w_{ijk} με $i = 1,2,\dots,F$, $j = 1,2,\dots,F$ και $k = 1,2,\dots,D_1$. Για κάθε φίλτρο, γίνεται συνέλιξη κάθε καναλιού του φίλτρου με το αντίστοιχο κανάλι της εισόδου, με τον τρόπο που υποδεικνύουν

τα P και S και τα αποτελέσματα αθροίζονται. Έτσι κάθε φίλτρο παράγει έναν πίνακα διάστασης $W_2 \times H_2$ τα στοιχεία του οποίου αθροίζονται με μία σταθερή πόλωση (η πόλωση είναι σταθερή για ένα φίλτρο αλλά διαφορετική μεταξύ των φίλτρων) και διέρχονται από μία μη γραμμική συνάρτηση. Ισοδύναμα στην έξοδο προκύπτει ένας tensor $3^{ης}$ τάξης και διάστασης $W_2 \times H_2 \times K$. Μάλιστα τα μεγέθη W_2 και H_2 δίνονται από τις σχέσεις

$$W_2 = \frac{W_1 - F + 2P}{S} + 1$$

$$H_2 = \frac{H_1 - F + 2P}{S} + 1$$

Επίπεδο Υποδειγματοληψίας - Subsampling or Pooling Layer

Το pooling layer τοποθετείται συνήθως μετά από κάθε συνελικτικό επίπεδο και ο σκοπός του είναι η μείωση των διαστάσεων της εξόδου του συνελικτικού επιπέδου. Δέχεται στην είσοδο έναν tensor διάστασης $W_1 \times H_1 \times D_1$ (D_1 είναι το πλήθος των φίλτρων στο συνελικτικό επίπεδο που προηγείται) και για κάθε κανάλι της εισόδου εκτελεί μία διαδικασία pooling στον πίνακα διάστασης $W_1 \times H_1$. Η διαδικασία αυτή μπορεί να είναι average pooling, L2-norm pooling αλλά συνήθως εκτελείται max pooling. Ουσιαστικά ένα συμμετρικό παράθυρο διατρέχει τον πίνακα με κάποιο βήμα και σε κάθε θέση κρατά τη μεγαλύτερη τιμή που συναντά. Οι παράμετροι του pooling layer είναι το μέγεθος του παραθύρου F και το βήμα S που συνήθως ταυτίζεται με το F για να μην υπάρχουν επικαλύψεις. Το max pooling επίπεδο επιστρέφει έναν tensor διάστασης $W_2 \times H_2 \times D_1$ με

$$W_2 = \frac{W_1 - F}{S} + 1$$

$$H_2 = \frac{H_1 - F}{S} + 1$$

Σημειώνεται ότι τα επίπεδα υποδειγματοληψίας δεν εισάγουν προσαρμόσιμες παραμέτρους.

Πλήρως-συνδεδεμένο Επίπεδο - Fully-connected Layer

Μετά από αρκετές διαδοχικές εμφανίσεις συνελικτικών και pooling επιπέδων συνηθίζεται να συμπεριλαμβάνονται fully-connected επίπεδα όπως αυτά που χρησιμοποιούνται στο πολυεπίπεδο perceptron. Η έξοδος του προηγούμενου επιπέδου που συνήθως είναι pooling επίπεδο συνδέεται πλήρως με κάθε νευρώνα του fully-connected επιπέδου. Για παράδειγμα στο σχήμα 3.9 μετά το δεύτερο επίπεδο υποδειγματοληψίας προκύπτουν 16 χάρτες χαρακτηριστικών μεγέθους 5×5 δηλαδή $16 \times 5 \times 5 = 450$ νευρώνες, κάθε ένας από τους οποίους συνδέεται με κάθε έναν από τους 120 νευρώνες του fully connected επιπέδου.

Όμοια οι 120 νευρώνες συνδέονται πλήρως με τους 80 νευρώνες του επόμενου επιπέδου και τελικά οι 80 νευρώνες με τους 10 της εξόδου.

Σημειώσεις

1. Η διάταξη των νευρώνων στα συνελικτικά και pooling επίπεδα είναι τρισδιάσταση σε αντίθεση με τη μονοδιάστατη διάταξη νευρώνων στα επίπεδα του πολυεπίπεδου perceptron. Ας θεωρήσουμε ένα συνελικτικό επίπεδο μεγέθους $W_2 \times H_2 \times K$ που συνδέεται με ένα προηγούμενο επίπεδο μεγέθους $W_1 \times H_1 \times D_1$. Το επίπεδο αυτό περιέχει $W_2 \cdot H_2 \cdot K$ νευρώνες οργανωμένους τρισδιάστατα στις 3 διαστάσεις, πλάτος, ύψος και βάθος. Κάθε νευρώνας συνδέεται με $F \cdot F \cdot D_1$ νευρώνες του προηγούμενου επιπέδου. Νευρώνες που βρίσκονται στο ίδιο βάθος δηλαδή αναφέρονται στο ίδιο φίλτρο μπορεί να συνδέονται με διαφορετικούς νευρώνες του προηγούμενου επιπέδου αλλά μοιράζονται τα ίδια βάρη.

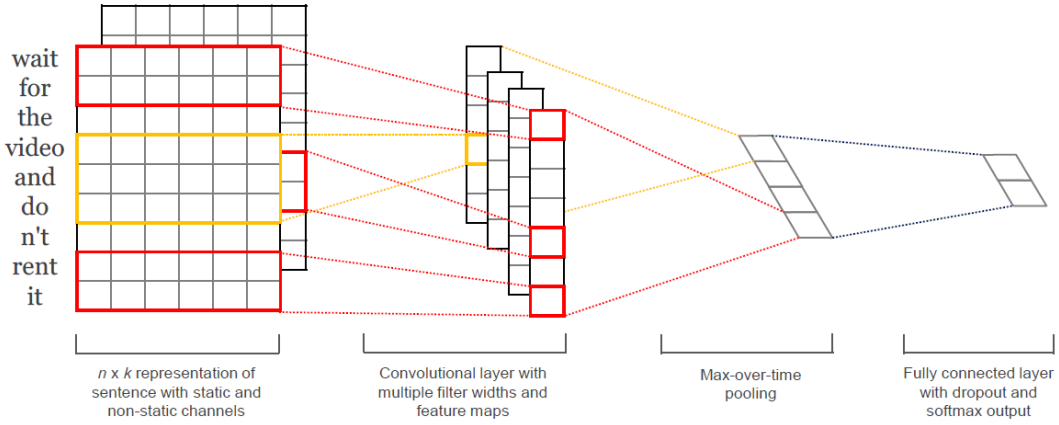
2. Τα βάρη των συνελικτικών δικτύων είναι τα στοιχεία του κάθε φίλτρου, η πόλωση για κάθε φίλτρο και τα συναπτικά βάρη των fully-connected επιπέδων. Προσαρμόζονται με τον αλγόριθμο backpropagation και τη βοήθεια τεχνικών stochastic gradient descent.

3. Για την αποφυγή overfitting χρησιμοποιούνται διάφορες τεχνικές όπως cross-validation, weight decay και αποκοπή συνδέσεων (μέθοδος dropout).

Συνελικτικά Δίκτυα στην Επεξεργασία Φυσικού Λόγου

Στην παρούσα εργασία θα εξεταστεί η υλοποίηση ενός συνελικτικού δικτύου για το πρόβλημα της ανάλυσης συναισθήματος. Συνελικτικά δίκτυα έχουν χρησιμοποιηθεί σε εφαρμογές sentiment analysis τα τελευταία χρόνια με αρκετά καλές επιδόσεις. Οι Santos και Gatti στο [23] προτείνουν τη χρήση διανυσματικών αναπαραστάσεων χαρακτήρων και λέξεων (character and word embeddings) για την τροφοδότηση ενός συνελικτικού δικτύου με δύο συνελικτικά επίπεδα, που είναι σε θέση να εκπαιδεύεται σε μικρές προτάσεις και να τις ταξινομεί με βάση το συναίσθημα. Σημειώνουν πολύ καλές επιδόσεις στο Stanford Twitter Sentiment dataset (STS) που είδαμε στο κεφάλαιο 2 αλλά και στο Stanford Sentiment Treebank dataset που περιέχει κριτικές ταινιών. Μία πιο απλή προσέγγιση προτείνεται στο [8] από τον Yoon Kim ο οποίος χρησιμοποιώντας μόνο διανυσματικές αναπαραστάσεις λέξεων και ένα απλό σχετικά συνελικτικό δίκτυο σημειώνει state-of-the-art επιδόσεις στο κλάσσικο movie review dataset των Pang και Lee. Μία εκτενής μελέτη των διανυσματικών αναπαραστάσεων λέξεων ή word vectors θα γίνει στην ενότητα 4.2, ωστόσο προς το παρόν αναφέρεται ότι είναι αναπαραστάσεις που αντιστοιχίζουν λέξεις σε διανύσματα συγκεκριμένης διάστασης. Μεταφράζουν τη σημασιολογική σχέση των λέξεων σε γραμμικές ιδιότητες του διανυσματικού χώρου. Στο σχήμα 3.10 δίνεται η αρχιτεκτονική του δικτύου που προτείνεται από τον Yoon Kim.

Το δίκτυο αποτελείται από ενα συνελικτικό επίπεδο, ένα επίπεδο pooling και τέλος ένα fully-connected επίπεδο που δίνει δύο εξόδους. Οι δύο έξοδοι αντιστοιχούν στις δύο κλάσεις στις οποίες ταξινομούνται οι προτάσεις δηλαδή θετικό και αρνητικό συναίσθημα. Η είσοδος του δικτύου είναι ένας πίνακας $n \times k$ όπου οι γραμμές αντιστοιχούν στις λέξεις της πρότασης και κάθε γραμμή είναι το word vector της αντίστοιχης λέξης.



Σχήμα 3.10

Στην πράξη η διάσταση n είναι σταθερή για όλα τα δείγματα εισόδου καθώς ο πίνακας στην είσοδο υφίσταται zero-padding μέχρι το μήκος της πρότασης με τις περισσότερες λέξεις στο σύνολο των δεδομένων.

Έστω $\mathbf{x}_i \in \mathbb{R}^k$ η k −διάστατη διανυσματική αναπαράσταση της λέξης w_i και μία πρόταση n λέξεων $w_{1:n} = \{w_1, w_2, \dots, w_n\}$. Η πρόταση αυτή αναπαρίσταται από τον πίνακα

$$\mathbf{x}_{1:n} = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \dots \oplus \mathbf{x}_n$$

όπου $\oplus$ συμβολίζει την πράξη της συνένωσης διανυσμάτων (concatenation). Γενικότερα η έκφραση $\mathbf{x}_{i:i+j}$ ορίζει τη συνένωση των διανυσμάτων $\mathbf{x}_i$, $\mathbf{x}_{i+1}$,…, $\mathbf{x}_{i+j}$. Κάθε φίλτρο $\mathbf{w} \in \mathbb{R}^{hk}$ εφαρμόζεται σε ένα παράθυρο h λέξεων για την παραγωγή ενός χαρακτηριστικού c_i σύμφωνα με τη σχέση

$$c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b)$$

όπου b είναι ο παράγοντας πόλωσης του φίλτρου και f μία μη γραμμική συνάρτηση. Το χαρακτηριστικό c_1 παράγεται με εφαρμογή του φίλτρου στο παράθυρο $\mathbf{x}_{1:h}$, το χαρακτηριστικό c_2 με εφαρμογή στο παράθυρο $\mathbf{x}_{2:h+1}$ και η διαδικασία συνεχίζει μέχρι την παραγωγή του χαρακτηριστικού c_{n-h+1} στο παράθυρο $\mathbf{x}_{n-h+1:n}$. Με αυτό τον τρόπο προκύπτει ο χάρτης των χαρακτηριστικών

$$\mathbf{c} = (c_1, c_2, \dots, c_{n-h+1})$$

ο οποίος στη συνέχεια διέρχεται από ένα επίπεδο max pooling που κρατά τη μέγιστη τιμή του χάρτη, δηλαδή

$$\hat{c} = \max(\mathbf{c})$$

Με βάση τα παραπάνω εξάγεται ένα χαρακτηριστικό από ένα φίλτρο. Το συνελικτικό επίπεδο χρησιμοποιεί πολλά φίλτρα με διαφορετικά βάρη $\mathbf{w}$ και διαφορετικά ύψη h και συγκεκριμένα η υλοποίηση που προτείνεται από τον Yoon Kim χρησιμοποιεί 300 διαφορετικά φίλτρα, 100 με $h = 3$, 100 με $h = 4$ και 100 με $h = 5$.

Το δίκτυο εκπαιδεύεται με την μέθοδο stochastic gradient descent και τον Adadelta κανόνα ενημέρωσης των βαρών. Επίσης για την υλοποίηση της μη γραμμικότητας χρησιμοποιείται η ReLU συνάρτηση ενώ παράλληλα εφαρμόζεται η μέθοδος dropout για την αποφυγή overfitting.

Τέλος σημειώνεται ότι τα διανύσματα των λέξεων σε ένα συνελικτικό δίκτυο μπορούν να ενημερώνονται ακριβώς όπως τα βάρη των φίλτρων , δηλαδή προς την κατεύθυνση που ελαχιστοποιείται κάποια συνάρτηση κόστους. Έτσι στο [8] προτείνονται τέσσερις διαφορετικές εκδοχές του παραπάνω μοντέλου που σχετίζονται με τα word vectors και τον τρόπο που το συνελικτικό δίκτυο τα χειρίζεται. Οι εκδοχές αυτές είναι οι παρακάτω.

CNN-rand : Το δίκτυο αρχικοποιεί τυχαία τα word vectors και τα ενημερώνει κατά την διάρκεια της εκπαίδευσης.

CNN-static : Το δίκτυο χρησιμοποιεί προ-εκπαιδευμένα (pretrained) word vectors και τα διατηρεί αμετάβλητα κατά τη διάρκεια της εκπαίδευσης

CNN-nonstatic : Το δίκτυο χρησιμοποιεί pretrained word vectors και τα ενημερώνει κατά τη φάση της εκπαίδευσης.

CNN-multichannel : Χρησιμοποιούνται δύο κανάλια στην είσοδο όπου σε κάθε δείγμα εκπαίδευσης το ένα κανάλι απεικονίζει την πρόταση χρησιμοποιώντας τα pretrained στατικά διανύσματα και το δεύτερο κανάλι χρησιμοποιώντας τα διανύσματα που ενημερώνονται ταυτόχρονα.

Το συνελικτικό δίκτυο που προτείνει ο Yoon Kim σημειώνει state-of-the-art αποτελέσματα στο πεδίο του sentiment analysis παρά την απλότητά του. Στο κεφάλαιο 5 θα δωθούν αναλυτικά οι λεπτομέρειες της υλοποίησής του όπως θα γίνει και για τους υπόλοιπους αλγορίθμους μηχανικής μάθησης που εξετάσαμε στο κεφάλαιο αυτο.

4 Εξαγωγή Χαρακτηριστικών σε Δεδομένα Κειμένου

Στο προηγούμενο κεφάλαιο εξετάστηκε το πρόβλημα της ταξινόμησης. Δεδομένων κάποιων σημείων σε ένα n-διάστατο χώρο συνοδευόμενων από κάποια κλάση, το ζητούμενο είναι η υλοποίηση ενός μαθηματικού μοντέλου που θα έχει τη δυνατότητα να ταξινομεί επιτυχώς νέα σημεία στη σωστή κλάση. Για το σκοπό αυτό είδαμε αλγορίθμους επιβλεπόμενης μάθησης που μαθαίνουν από τα δεδομένα εκπαίδευσης, όπου η μάθηση στην ουσία είναι η χρήση των δεδομένων αυτών για την υλοποίηση κάποιας διαμέρισης του χώρου σε περιοχές, ώστε σημεία της ίδιας περιοχής να αντιστοιχούν στην ίδια κλάση. Ακόμα και ταξινομητές όπως ο Naive Bayes, το πολυεπίπεδο perceptron και ο αλγόριθμος k-Nearest Neighbors στην πράξη εκτελούν τέτοια διαμέριση παρόλο που δεν γίνεται άμεσα προφανές από τη θεωρία.

Τα παραπάνω λοιπόν είναι μαθηματικά εργαλεία που εφαρμόζονται σε οποιοδήποτε πρόβλημα ταξινόμησης. Όταν όμως θέλουμε να ταξινομήσουμε αντικείμενα σε κλάσεις, προτού εφαρμοστεί ο αλγόριθμος ταξινόμησης, είναι απαραίτητο να αναπαραστήσουμε με κάποιο τρόπο τα αντικείμενα αυτά σε διανύσματα ενός χώρου χαρακτηριστικών. Η διαδικασία αυτή καλείται *εξαγωγή χαρακτηριστικών* (***feature extraction***) και το γενικότερο πεδίο μελέτης ***feature engineering***. Μπορεί να είναι μία πολύ απλή διαδικασία αλλά μπορεί να είναι και εξίσου σύνθετη. Για παράδειγμα, ας υποθέσουμε ότι θέλουμε να ταξινομήσουμε αυτόματα φρούτα στις διάφορες υποκατηγορίες. Φαίνεται απολύτως λογικό να χρησιμοποιήσουμε για χαρακτηριστικά το μέγεθος και το χρώμα και πιθανώς να έχουμε απόλυτη επιτυχία στην ταξινόμηση νέων δεδομένων. Παρόμοια κατάσταση συναντάμε στην ταξινόμηση ιατρικών δεδομένων όπως για παράδειγμα αν θέλουμε να ταξινομήσουμε καρκινικά κύτταρα σε καλοήθη ή κακοήθη. Σε τέτοιες περιπτώσεις, η επιστημονική γνώση καθοδηγεί την επιλογή χαρακτηριστικών. Αν δηλαδή το αντίστοιχο επιστημονικό πεδίο έχει επίγνωση του προβλήματος και γνωρίζει τι είναι αυτό που διαφοροποιεί τις κλάσεις

είναι εύκολο να εξάγουμε χαρακτηριστικά απλά και να πετύχουμε καλή απόδοση στην ταξινόμηση. Στην περίπτωση των καρκινικών κυττάρων τα χαρακτηριστικά αυτά θα μπορούσαν να είναι το μέγεθος του κυττάρου ή άλλα δεδομένα που υποδεικνύει η έρευνα στο πεδίο της ιατρικής. Τι συμβαίνει όμως όταν το πρόβλημα δεν είναι τόσο απλό και δεν γνωρίζουμε τι είναι αυτό που διαφοροποιεί τις κλάσεις; Για παράδειγμα πότε μία πρόταση εκφράζει θετικό συναίσθημα και πότε αρνητικό; Ποιά είναι αυτά τα χαρακτηριστικά που δίνουν σε ένα κομμάτι κειμένου θετική ή αρνητική πολικότητα συναισθήματος; Πότε μία φράση εκφράζει ειρωνία;

Σε τέτοιες περιπτώσεις αυτό που μπορούμε να κάνουμε είναι να δώσουμε στον ταξινομητή ακατέργαστα δεδομένα (*raw data*) χωρίς feature engineering και να του αναθέσουμε την εύρεση higher level χαρακτηριστικών που μπορεί να διαφοροποιούν τις κλάσεις. Ειδικά τα νευρωνικά δίκτυα βασίζονται στη λογική της αναζήτησης τέτοιων χαρακτηριστικών με τη βοήθεια των κρυφών επιπέδων. Τα παραπάνω γίνονται κατανοητά με ένα παράδειγμα υπολογιστικής όρασης. Ας υποθέσουμε ότι θέλουμε να ταξινομήσουμε εικόνες προσώπων σε κατηγορίες ανάλογα με το συναίσθημα που εκφράζουν. Ας περιοριστούμε σε δύο κλάσεις, χαρά και λύπη. Η μία προσέγγιση, αυτή του feature engineering είναι να αναζητήσουμε χαρακτηριστικά στην εικόνα που διαισθητικά πιστεύουμε ότι καθορίζουν το αποτέλεσμα. Για παράδειγμα, η καμπύλη των χειλιών, το σχήμα των ματιών και οι γωνίες του προσώπου. Εναλλακτικά μπορούμε να χρησιμοποιήσουμε σαν χαρακτηριστικά τα pixel της εικόνας και να αφήσουμε το δίκτυο να ανακαλύψει τα υψηλού επιπέδου χαρακτηριστικά. Αυτή είναι και η βασική προσέγγιση στην επεξεργασία εικόνων με συνελικτικά δίκτυα και τα αποτελέσματα ([9]) δείχνουν ότι τα δίκτυα αυτά μπορούν να ανιχνεύουν πολύπλοκες δομές στις εικόνες και να ταξινομούν με μεγάλη επιτυχία οτιδήποτε, από χειρόγραφους χαρακτήρες έως πρόσωπα σε φωτογραφίες (Facebook automated tagging[22]).

Τα παραπάνω είναι ενδεικτικά και του προσανατολισμού της παρούσας διπλωματικής εργασίας. Θα εστιάσουμε στη χρήση αλγορίθμων μηχανικής μάθησης με raw data κειμένου και όχι σε τεχνικές feature engineering. Τέτοιες τεχνικές όπως η χρήση negation και τα λεξικά συναισθήματος δώθηκαν συνοπτικά στο κεφάλαιο 1.

Σε αυτό το κεφάλαιο λοιπόν θα δούμε τρόπους εξαγωγής χαρακτηριστικών από δεδομένα κειμένου. Υπενθυμίζεται ότι το πρόβλημα είναι η ταξινόμηση ενός συνόλου από tweets σε δύο κλάσεις, θετικό ή αρνητικό συναίσθημα. Συνεπώς το ζητούμενο είναι η αντιστοίχιση κάθε tweet σε μία διανυσματική αναπαράσταση $\mathbf{x} = (x_1, x_2, \dots, x_n)$ όπου n η διάσταση του χώρου χαρακτηριστικών.

4.1 Bag-of-Words

Ο πλέον απλός τρόπος εξαγωγής χαρακτηριστικών από δεδομένα κειμένου είναι η μέθοδος Bag-of-Words. Με τη μέθοδο αυτή λέξεις, φράσεις, προτάσεις και κείμενα αναπαρίστανται με αραιά διανύσματα μεγάλου μήκους σε ένα διανυσματικό χώρο υψηλής διάστασης. Όπως δηλώνει και το όνομα, το κομμάτι κειμένου αντιμετωπίζεται σαν ένας σάκος με λέξεις.

[22] https://en.wikipedia.org/wiki/DeepFace

Αυτό σημαίνει πρακτικά ότι αγνοούμε τη σειρά των λέξεων και ενδιαφερόμαστε μόνο για την παρουσία ή τη συχνότητα λέξεων στο κείμενο. Άμεσα λοιπόν γίνεται αντιληπτή η αδυναμία της Bag-of-Words προσέγγισης, αφού η σειρά των λέξεων μπορεί να είναι καθοριστικής σημασίας όχι μόνο για τις ανάγκες της ανάλυσης συναισθήματος αλλά και για κάθε τύπου εξαγωγή πληροφοριών από κείμενο. Ωστόσο είναι ιδιαίτερα δημοφιλής για την απλοϊκότητά της και τα ικανοποιητικά αποτελέσματα που δίνει ειδικά στο πεδίο της ανάλυσης συναισθήματος και του topic classification ([28]).

4.1.1 Term Occurrence και Term Frequency

Αρχικά από το σύνολο όλων των *εγγράφων* (***documents***) δημιουργείται το *λεξικό* (***vocabulary***) που περιέχει όλες τις λέξεις που εμφανίζονται σε όλα τα έγγραφα. Έπειτα κάθε έγγραφο αναπαρίσταται από ένα διάνυσμα του οποίου κάθε επιμέρους χαρακτηριστικό αντιστοιχεί σε μία λέξη και παίρνει τιμές ανάλογα με την παρουσία ή τη συχνότητα της λέξης στο έγγραφο. Έτσι οι τιμές ενός χαρακτηριστικού μπορεί να είναι 0 και 1 για την παρουσία λέξης ή κάθε φυσικός αριθμός για τη συχνότητα λέξης. Η πρώτη προσέγγιση καλείται ***term occurrence*** και η δεύτερη ***term frequency***. Για παράδειγμα ας υποθέσουμε το σύνολο S των τριών εγγράφων που αφορούν κριτικές ταινιών

d_1: *light , cute and forgettable .*

d_2: *. . . hypnotically dull .*

d_3: *journalistically dubious , inept and often lethally dull .*

Το λέξικο σε αυτή την περίπτωση είναι το σύνολο V όλων των λέξεων w που συναντάμε σε κάθε έγγραφο d. Δηλαδή

$$V(S) = \{w | w \in d \; \forall \; d \in S\}$$

Στο παράδειγμά μας το λεξικό είναι

$V(S) =$ [‘light’ , ‘,’ , ‘cute’ , ‘and’ , ‘forgettable’ , ‘.’ , ‘hypnotically’ , ‘dull’ , ‘journalistically’ , ‘dubious’ , ‘inept’ , ‘often’ , ‘lethally’]

Οι λέξεις-όροι που περιέχονται στο λεξικό, αντιστοιχούν στα χαρακτηριστικά. Έτσι οι term occurrence αναπαραστάσεις των εγγράφων d είναι

$$\mathbf{x}_{to}(d_1) = [1,1,1,1,1,1,0,0,0,0,0,0,0]$$

$$\mathbf{x}_{to}(d_2) = [0,0,0,0,0,1,1,1,0,0,0,0,0]$$

$$\mathbf{x}_{to}(d_3) = [0,1,0,1,0,1,0,1,1,1,1,1,1]$$

και αντίστοιχα οι term frequency

$$\mathbf{x}_{tf}(d_1) = [1,1,1,1,1,1,0,0,0,0,0,0,0]$$

$$\mathbf{x}_{tf}(d_2) = [0,0,0,0,0,4,1,1,0,0,0,0,0]$$

$$\mathbf{x}_{tf}(d_3) = [0,1,0,1,0,1,0,1,1,1,1,1,1]$$

Παρατηρούμε ότι οι δύο αναπαραστάσεις είναι ίδιες για τα d_1 και d_3 καθώς σε αυτά δεν υπάρχουν λέξεις που εμφανίζονται δύο φορές. Γενικότερα σε περιπτώσεις εγγράφων μικρού μήκους (sentence-based sentiment analysis) οι δύο αναπαραστάσεις τείνουν να παράγουν παρόμοια διανύσματα καθώς οι λέξεις σε μικρά έγγραφα επαναλαμβάνονται πιο σπάνια.

Η διανυσματική αναπαράσταση έχει διάσταση ίση με το μέγεθος του λεξικού δηλαδή

$$n = |V(S)|$$

όπου ο τελεστής $|A|$ δηλώνει τον πληθυσμό (cardinality) του συνόλου A. Στην περίπτωση λοιπόν ενός πραγματικού συνόλου δεδομένων όπου τα δείγματα είναι μερικές χιλιάδες, όπως γίνεται αντιληπτό, το μέγεθος του λεξικού μπορεί να γίνει δυσθεώρητα μεγάλο. Σε μία τέτοια περίπτωση προκύπτουν διανύσματα με μέγεθος που καθιστά οποιαδήποτε υπολογιστική εργασία αδύνατη. Γι'αυτό το λόγο στην πράξη ποτέ δεν χρησιμοποιείται όλο το λεξικό για την εξαγωγή χαρακτηριστικών παρά μόνο οι όροι του με τη μεγαλύτερη συχνότητα σε ολόκληρο το S. Έτσι ο χρήστης καθορίζει τη διάσταση των διανυσμάτων επιλέγοντας τους πιο συχνούς όρους σε όλο το σώμα κειμένου (*text corpus*). Συνηθισμένη επιλογή αποτελούν οι 1,000 με 5,000 πιο συχνοί όροι οπότε προκύπτουν διανύσματα με 1,000-5,000 διαστάσεις.

Ήδη από το παραπάνω απλό παράδειγμα φάνηκε ότι κάποιοι συνηθισμένοι όροι εμφανίζονται αρκετά συχνά. Στο έγγραφο d_2 ο όρος '.' εμφανίζεται τέσσερις φορές. Τέτοιοι όροι που είναι αρκετά συνήθεις καλούνται *stopwords* και σε πολλές εφαρμογές αφαιρούνται από το λεξικό ή δεν χρησιμοποιούνται σαν χαρακτηριστικά, προκειμένου να προκύπτουν ισορροπημένες διανυσματικές αναπαραστάσεις. Το πρόβλημα αυτό δεν είναι τόσο εμφανές στην περίπτωση κειμένων μικρού μήκους αλλά σε document-based εφαρμογές μπορεί να πλήξει την απόδοση. Μία εναλλακτική της πλήρους αφαίρεσης των stopwords είναι η αναπαράσταση tf-idf (*term frequency – inverse document frequency*), παραλλαγή της term frequency αναπαράστασης, σύμφωνα με την οποία κάθε χαρακτηριστικό, ισούται με την συχνότητα του όρου στο έγγραφο πολλαπλασιασμένης με κάποιο βάρος που σχετίζεται με τη συχνότητα του όρου σε όλο το σώμα κειμένου.

Συνοπτικά οι διάφορες Bag-of-Words αναπαραστάσεις δίνονται μαθηματικά ως εξής:

Έστω S ένα σύνολο από m έγγραφα, όπου το κάθε έγγραφο συμβολίζεται ως d_i , $i = 1,2,\dots,m$. Κατασκευάζεται ένα λεξικό $V(S) = \{w_j, j = 1,2,\dots,n\}$ με τους n πιο συχνούς όρους στο σύνολο S. Οι τρεις αναπαραστάσεις που περιγράψαμε είναι οι ακόλουθες

term occurrence : $\mathbf{x}_{to}(d_i) = (x_1, x_2, \dots, x_n)$ όπου $x_j = \begin{cases} 0 \ , & \alpha\nu\ f(w_j, d_i) = 0 \\ 1 \ , & \alpha\nu\ f(w_j, d_i) > 0 \end{cases}$

term frequency : $\mathbf{x}_{tf}(d_i) = (x_1, x_2, \dots, x_n)$ όπου $x_j = f(w_j, d_i)$

tf-idf : $\mathbf{x}_{tf-idf}(d_i) = (x_1, x_2, \dots, x_n)$ όπου $x_j = f(w_j, d_i) \cdot \log\frac{m}{k_j}$ με $k_j = |\{d_i \in S : w_j \in d_i\}|$

όπου $f(w_j, d_i)$ είναι η συχνότητα εμφάνισης του όρου w_j στο έγγραφο d_i και k_j το πλήθος των εγγράφων του συνόλου S στα οποία εμφανίζεται ο όρος w_j.

4.1.2 Όροι και n-grams

Στην προηγούμενη υποενότητα δώθηκε μία περιγραφή των αναπαραστάσεων term frequency και term occurrence όπου θεωρήσαμε σαν όρους (terms), απλές λέξεις και σημεία στίξης. Η γενίκευση της λογικής αυτής μας προτρέπει να χρησιμοποιήσουμε και ακολουθίες δύο, τριών ή και παραπάνω λέξεων σαν όρους. Τέτοιες ακολουθίες διαδοχικών δεδομένων καλούνται n-grams. Στην περίπτωσή μας, όπου επεξεργαζόμαστε δεδομένα κειμένου, τα n-grams είναι στην πράξη ακολουθίες διαδοχικών λέξεων. Έτσι η ακολουθία δύο λέξεων καλείται ***bigram***, η ακολουθία τριών λέξεων ***trigram***, τεσσάρων ***fourgram*** κ.ο.κ. Επίσης η απλή λέξη δηλαδή το n-gram για $n = 1$ καλείται ***unigram***.

Η λογική είναι παρόμοια με τα προηγούμενα. Για κάθε περίπτωση n-gram κατασκευάζεται το αντίστοιχο λεξικό από το σώμα κειμένου και σαν όροι επιλέγονται οι συχνότεροι σε κάθε λεξικό. Έπειτα προσδιορίζονται οι αναπαραστάσεις με τρόπο πανομοιότυπο με την περίπτωση απλών λέξεων. Αν το n-gram υπάρχει στο έγγραφο τότε το αντίστοιχο χαρακτηριστικό παίρνει τιμή διάφορη του 0. Επισημαίνεται ότι τα n-grams είναι ακολουθίες λέξεων οπότε αναζητείται στο έγγραφο η ίδια ακριβώς ακολουθία και όχι οι μεμονωμένες λέξες σε τυχαίες θέσεις.

Για παράδειγμα, ας υποθέσουμε το παρακάτω σώμα κειμένου S

d_1: *tender yet lacerating and darkly funny fable .*

d_2: *a taut , intelligent psychological drama .*

Τα λεξικά που προκύπτουν για τα n-grams με $n = 1,2,3$ είναι

$V_{uni}(S)$ = [‘tender’ , ‘yet’ , ‘lacerating’ , ‘and’ , ‘darkly’ , ‘funny’ , ‘fable’ , ‘.’ , ‘a’ , ‘taut’ , ‘,’ , ‘intelligent’ , ‘psychological’ , ‘drama’]

$V_{bi}(S)$ = [‘tender yet’ , ‘yet lacerating’ , ‘lacerating and’ , ‘and darkly’ , ‘darkly funny’ , ‘funny fable’ , ‘fable .’ , ‘a taut’ , ‘taut ,’ , ‘, intelligent’ , ‘intelligent psychological’ , ‘psychological drama’ , ‘drama .’]

$V_{tri}(S)$ = ['tender yet lacerating' , 'yet lacerating and' , 'lacerating and darkly'
, 'and darkly funny' , 'darkly funny fable' , 'funny fable .' , 'a taut ,'
, 'taut , intelligent' , ', intelligent psychological'
, 'intelligent psychological drama' , 'psychological drama .']

Εάν παραδείγματος χάριν επιλέξουμε για χαρακτηριστικά τους όρους

$V(S)$ = ['tender' , 'and' , 'darkly' , 'psychological drama' , 'darkly funny fable']

τότε οι term occurrence αναπαραστάσεις θα είναι

$$\mathbf{x}_{to}(d_1) = [1,1,1,0,1]$$

$$\mathbf{x}_{to}(d_2) = [0,0,0,1,0]$$

Στη γενική περίπτωση ενός πραγματικού συνόλου δεδομένων κατασκευάζονται τα διάφορα λεξικά, στη συνέχεια επιλέγονται οι πιο συχνοί όροι σε κάθε λεξικό και προκύπτει ένας κατάλογος όρων που θα χρησιμοποιηθούν σαν χαρακτηριστικά. Έπειτα κάθε έγγραφο αναπαρίσταται με τις μεθόδους term occurrence, term frequency ή tf-idf σαν ένα αραιό διάνυσμα μήκους ίσου με τον αριθμό όρων που χρησιμοποιούνται σαν χαρακτηριστικά.

Η Bag-of-Words προσέγγιση, που είδαμε μέχρι τώρα, είναι ο απλούστερος τρόπος εξαγωγής χαρακτηριστικών από κείμενο. Το μειονέκτημα της μεθόδου όταν αφορά αναπαραστάσεις εγγράφων, είναι ότι αγνοεί τη σειρά των λέξεων. Ακόμα και με τη χρήση n-grams για χαρακτηριστικά, το πρόβλημα παραμένει γιατί από όλα τα n-grams που εμφανίζονται στο σώμα κειμένου ελάχιστα τελικά επιλέγονται για χαρακτηριστικά. Εάν λοιπόν δύο προτάσεις, περιέχουν τις ίδιες λέξεις αλλά με διαφορετική σειρά, τότε οι αναπαραστάσεις τους με την μέθοδο που είδαμε προηγουμένως θα είναι πανομοιότυπες εκτός και αν κάποιο n-gram που χρησιμοποιείται σαν χαρακτηριστικό βρίσκεται στη μία και όχι στην άλλη, κάτι που φυσικά δεν είναι καθόλου συνηθισμένο. Για παράδειγμα

an infection killing white blood cells

white blood cells killing an infection

Οι δύο αυτές φράσεις αποτελούνται από τις ίδιες λέξεις αλλά έχουν διαφορετική σύνταξη. Το νόημα είναι διαμετρικά αντίθετο, ωστόσο οι Bag-of-words αναπαραστάσεις πανομοιότυπες.

Βέβαια η σειρά των λέξεων δεν είναι το μοναδικό πρόβλημα της απλοϊκής αυτής θεώρησης. Στην περίπτωση αναπαράστασης λέξεων, μία μοναδική λέξη στη μέθοδο Bag-of-Words αναπαρίσταται από ένα διάνυσμα με 1 στην αντίστοιχη θέση της λέξης στο λεξικό και 0 σε όλες τις άλλες θέσεις. Η αναπαράσταση αυτή καλείται ***one-hot encoding***. Μία διαφορετική λέξη θα είναι ένα διάνυσμα ίδιου μήκους με 1 σε κάποια άλλη θέση και 0 αλλού. Έτσι, οποιεσδήποτε δύο λέξεις του λεξικού θα έχουν ευκλείδια απόσταση στο χώρο χαρακτηριστικών ίση με $\sqrt{2}$ και εσωτερικό γινόμενο 0. Δηλαδή οι θέσεις και οι αποστάσεις των λέξεων δεν μας δίνουν καμία πληροφορία για το νόημα των λέξεων και τη σημασιολογική, συντακτική ή γραμματική συνάφεια διαφορετικών λέξεων. Με άλλα λόγια

δημιουργείται ένας διανυσματικός χώρος ο οποίος δεν παρέχει πληροφορίες για τις σημασιολογικές σχέσεις μεταξύ λέξεων. Κατ' επέκταση η αναπαράσταση εγγράφων δε δίνει εγγυήσεις ότι έγγραφα με παρόμοιο νόημα θα αντιστοιχίζονται σε κοντινές θέσεις του χώρου χαρακτηριστικών.

Πέρα από το μοντέλο Bag-of-Words

Αντλώντας έμπνευση από τα μειονεκτήματα αυτά καταλήγουμε στο συμπέρασμα ότι απαιτείται ένα διαφορετικό μαθηματικό μοντέλο για την περιγραφή δεδομένων κειμένου. Χρειαζόμαστε ένα τρόπο να απεικονίζουμε λέξεις, φράσεις, προτάσεις και ολόκληρα κείμενα σε ένα διανυσματικό χώρο λίγων διαστάσεων, έτσι ώστε κοντινά σημεία στο χώρο να αντιπροσωπεύουν παρόμοιο νόημα (***semantic similarity***). Επίσης είναι χρήσιμο, για υπολογιστικούς λόγους, τα διανύσματα να είναι πυκνά (***dense***) με συνεχή χαρακτηριστικά και όχι διακριτά και αραιά όπως στην περίπτωση των αναπαραστάσεων term frequency ή term occurrence. Επιπλέον οι αναπαραστάσεις αυτές πρέπει να έχουν καθολικό χαρακτήρα έτσι ώστε να μπορούν να χρησιμοποιηθούν σε ποικίλες εφαρμογές επεξεργασίας κειμένου. Είτε πρόκειται για sentiment analysis, είτε για machine translation οι αναπαραστάσεις πρέπει να είναι ίδιες όπως ακριβώς και στην επεξεργασία εικόνας τα pixel χρησιμοποιούνται σαν χαρακτηριστικά ανεξάρτητα από την επιμέρους εφαρμογή.

Τέτοιες διανυσματικές αναπαραστάσεις στη ξενόγλωσση βιβλιογραφία απαντώνται με τους όρους ***embeddings***, ***representations*** ή απλά ***vectors***. Συγκεκριμένα για τις αναπαραστάσεις λέξεων, συναντάμε τους όρους ***word embeddings***, ***word representations*** ή ***word vectors***.

Η ιδέα της αναπαράστασης λέξεων με σημεία ενός διανυσματικού χώρου μικρής διάστασης είναι σχετικά παλιά, ωστόσο τα τελευταία χρόνια έχει γίνει σημαντική πρόοδος με την ανάπτυξη αλγορίθμων που έχουν την ικανότητα να εξάγουν υψηλής ποιότητας αναπαραστάσεις από πολύ μεγάλους όγκους κειμένου. Ως υψηλής ποιότητας αναπαραστάσεις (***high quality word embeddings***) χαρακτηρίζονται οι αναπαραστάσεις εκείνες που συλλαμβάνουν τις πολυεπίπεδες συντακτικές και σημασιολογικές σχέσεις μεταξύ των λέξεων και τις μεταφράζουν σε γραμμικές ιδιότητες του χώρου αναπαραστάσεων ([14]). Συμπληρωματικά λοιπόν με πριν, που κάναμε λόγο για μοντέλα που αναπαριστούν λέξεις παρόμοιας σημασίας σε κοντινά σημεία, τα μοντέλα που θα δούμε στη συνέχεια θα είναι σε θέση να ανακαλύπτουν σχέσεις μεταξύ των λέξεων σε πολλά επίπεδα και όχι απλά νοηματική ομοιότητα. Έτσι θα δούμε διανυσματικές αναπαραστάσεις με ιδιότητες όπως

$$\mathbf{x}_{king} - \mathbf{x}_{man} \cong \mathbf{x}_{queen} - \mathbf{x}_{woman}$$
$$\mathbf{x}_{big} - \mathbf{x}_{bigger} \cong \mathbf{x}_{strong} - \mathbf{x}_{stronger}$$
$$\mathbf{x}_{greece} - \mathbf{x}_{athens} \cong \mathbf{x}_{france} - \mathbf{x}_{paris}$$

Θα επανέλθουμε με λεπτομέρειες σε αυτό το θέμα στη συνέχεια. Προς το παρόν σημειώνεται ότι ο εναλλακτικός τρόπος της Bag-of-Words προσέγγισης στην αναπαράσταση δεδομένων κειμένου είναι η χρήση των διανυσματικών αυτών αναπαραστάσεων, οι οποίες από εδώ και στο εξής θα καλούνται word vectors. Στα επόμενα θα δούμε πως μπορούμε να δημιουργήσουμε τέτοιες διανυσματικές αναπαραστάσεις

ενσωματώνοντας πληροφορία για τις σημασιολογικές και συντακτικές σχέσεις των διαφόρων λέξεων.

4.2 Word Vectors

"You shall know a word by the company it keeps"
John Rupert Firth – English Linguist 1957:11

Σύμφωνα με την θεωρία των word vectors κάθε λέξη αντιστοιχίζεται σε ένα διάνυσμα ενός χώρου μικρής διάστασης. Τα διανύσματα αυτά είναι πυκνά, με πραγματικές τιμές, η διάσταση τυπικά κυμαίνεται μέχρι μερικές εκατοντάδες και η αντιστοίχιση γίνεται με τρόπο τέτοιο ώστε οι σχέσεις μεταξύ των λέξεων να μετατρέπονται σε ιδιότητες του χώρου των word vectors. Για την δημιουργία των διανυσματικών αναπαραστάσεων χρησιμοποιούνται κατά κύριο λόγο δύο μέθοδοι. Η πρώτη χρησιμοποιεί τεχνικές ***μείωσης διαστατικότητας*** (***dimensionality reduction***) σε co-occurrence πίνακες και η δεύτερη χρησιμοποιεί ***νευρωνικά γλωσσικά μοντέλα*** (***neural language models***). Στην πρώτη θα αναφερθούμε εισαγωγικά στην ενότητα 4.2.1 και στην δεύτερη στην ενότητα 4.2.2 όπου θα εξετάσουμε τον αλγόριθμο word2vec. Σε κάθε περίπτωση η εκπαίδευση των word vectors είναι μη επιβλεπόμενη και οι πληροφορίες για τις σχέσεις μεταξύ λέξεων προκύπτουν από τον τρόπο που οι λέξεις συντάσσονται σε πραγματικά κείμενα. Τα μοντέλα που παράγουν word vectors τροφοδοτούνται απλά από κείμενο χωρίς επίβλεψη και μοντελοποιούν τις λέξεις βασιζόμενα στο context και στον τρόπο γειτνίασης διαφορετικών λέξεων.

4.2.1 Lexical Co-occurrence

Ο co-occurrence πίνακας σέ ένα σώμα κειμένου είναι ένας πίνακας που περιέχει πληροφορίες για την συνύπαρξη λέξεων στο κείμενο. Στην απλούστερη περίπτωση, είναι τετραγωνικός και συμμετρικός και κάθε γραμμή του όπως και κάθε στήλη του αντιστοιχούν σε έναν όρο από το λεξικό. Αν η γραμμή i αντιστοιχεί στον όρο w_i και η στήλη j στον όρο w_j τότε ο co-occurrence πίνακας είναι

$$\mathbf{C} = \{c_{ij}\} \text{ με } c_{ij} = f(w_i, w_j, S)$$

όπου $f(w_i, w_j, S)$ η συχνότητα εμφάνισης της ακολουθίας λέξεων $w_i w_j$ ή της $w_j w_i$ σε όλο το σώμα κειμένου. Ας υποθέσουμε για παράδειγμα το σύνολο S των προτάσεων

I like machine learning.

I like sentiment analysis.

I enjoy learning.

Το λεξικό μας είναι

$V(S) =$ ['I' , 'like' , 'machine' , 'learning' , 'sentiment' , 'analysis' , 'enjoy' , '.']

	I	like	machine	learning	sentiment	analysis	enjoy	.
I	0	2	0	0	0	0	1	0
like	2	0	1	0	1	0	0	0
machine	0	1	0	1	0	0	0	0
learning	0	0	1	0	0	0	1	1
sentiment	0	1	0	0	0	1	0	0
analysis	0	0	0	0	1	0	0	1
enjoy	1	0	0	1	0	0	0	0
.	0	0	0	1	0	1	0	0

Πίνακας 4.1

Ο πίνακας co-occurrence δίνεται στον πίνακα 4.1.

Με τη μέθοδο SVD (Singular Value Decomposition) ο παραπάνω αραιός πίνακας μπορεί να παραγοντοποιηθεί ως εξής

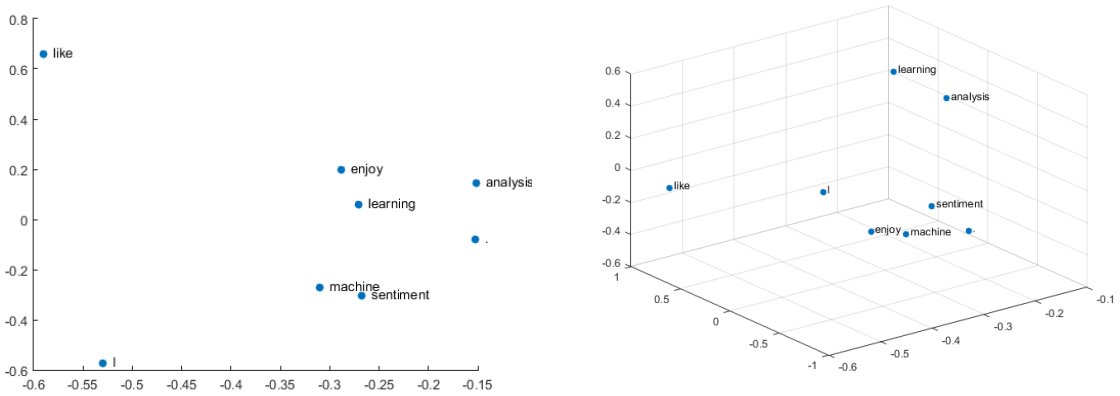
$$\mathbf{C} = \mathbf{U}\Sigma\mathbf{V}$$

$$\mathbf{C} = \begin{bmatrix} \begin{bmatrix} \mathbf{u_1} \end{bmatrix} & \begin{bmatrix} \mathbf{u_2} \end{bmatrix} & \dots & \begin{bmatrix} \mathbf{u_n} \end{bmatrix} \end{bmatrix} \cdot \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix} \cdot \begin{bmatrix} [& \mathbf{v_1} &] \\ [& \mathbf{v_2} &] \\ & \vdots & \\ [& \mathbf{v_n} &] \end{bmatrix}$$

όπου $\mathbf{\Sigma}$ διαγώνιος πίνακας, σ_i καλούνται *ιδιάζουσες τιμές* (*singular values*), $\mathbf{u_i}$ *αριστερά ιδιάζοντα διανύσματα* (*left-singular vectors*) και $\mathbf{v_i}$ *δεξιά ιδιάζοντα διανύσματα* (*right-singular vectors*). Επιλέγοντας τις k μεγαλύτερες ιδιάζουσες τιμές και τα ιδιάζοντα διανύσματα από αριστερά και δεξιά που αντιστοιχούν σε αυτές, παίρνουμε την ***προσέγγιση βαθμού k του πίνακα $\boldsymbol{C}$ με το μικρότερο σφάλμα κατά τη νόρμα Frobenius***. Αυτό πρακτικά σημαίνει ότι αν ο αρχικός πίνακας είναι $m \times l$, δηλαδή περιέχει l στήλες m διαστάσεων ή m γραμμές l διαστάσεων, επιλέγοντας τις k μεγαλύτερες ιδιάζουσες τιμές και τα αντίστοιχα k διανύσματα $\mathbf{u}$ και k διανύσματα $\mathbf{v}$, τα k διανύσματα $\mathbf{u}$ δίνουν την k-διάστατη προσέγγιση των m γραμμών και τα k διανύσματα $\mathbf{v}$ την k-διάστατη προσέγγιση των l στηλών. Η παραγοντοποίηση SVD λοιπόν είναι μία μέθοδος μείωσης της διαστατικότητας.

Στο παράδειγμά μας ο πίνακας $\mathbf{C}$ είναι τετραγωνικός ($m = n$) και συμμετρικός. Σε αυτή την περίπτωση από την SVD ανάλυση τα διανύσματα $\mathbf{u}$ και $\mathbf{v}$ προκύπτουν ίδια δηλαδή $\mathbf{u_i} = \mathbf{v_i}$ για $i = 1,2, \dots, n$. Έτσι, επιλέγοντας τις k μεγαλύτερες ιδιάζουσες τιμές και τα αντίστοιχα k διανύσματα αριστερά και δεξιά, καθώς αυτά ταυτίζονται καταλήγουμε με k διανύσματα συνολικά, διάστασης $m = n$. Αυτό φαίνεται λογικό γιατι στην περίπτωση ενός co-occurrence πίνακα τα ίδια αντικείμενα αντιστοιχίζονται σε γραμμές και στήλες οπότε συνολικά έχουμε $m = n$ αντικείμενα να περιγράψουμε εν αντιθέσει με την περίπτωση μη

τετραγωνικού πίνακα όπου πρέπει να περιγραφούν $m+n$. Από τον αρχικό λοιπόν 8×8 πίνακα που εκφράζει την συνύπαρξη 8 λέξεων στο corpus, με την βοήθεια της SVD παραγοντοποίησης, μπορούμε να πάρουμε k-διάστατες ($k<8$) προσεγγίσεις των στηλών ή των γραμμών, δηλαδή των λέξεων. Εναλλακτικά παράγονται k-διάστατα word vectors.



Σχήμα 4.1

Στο σχήμα 4.1 φαίνονται οι διανυσματικές αναπαραστάσεις που προκύπτουν από τον πίνακα 4.1 για $k=2$ και $k=3$.

Η μείωση της διαστατικότητας του co-occurrence πίνακα είναι μία απλή μέθοδος για την εξαγωγή διανυσματικών αναπαραστάσεων λέξεων. Ωστόσο μπορεί να επεκταθεί και σε αναπαραστάσεις εγγράφων. Σε αυτή την περίπτωση ο co-occurrence πίνακας είναι μη τετραγωνικός και κάθε γραμμή του αφορά μία λέξη ή γενικότερα έναν όρο (n-gram) ενώ κάθε στήλη ένα έγγραφο. Κάθε στοιχείο του πίνακα αναφέρεται στη συνύπαρξη όρων με έγγραφα δηλαδή κάθε στήλη είναι μία term occurrence, term frequency ή tf-idf αναπαράσταση. Συνεπώς με την SVD παραγοντοποίηση παίρνουμε k-διάστατες προσεγγίσεις όρων και εγγράφων. Η μέθοδος αυτή είναι ευρύτερα γνωστή ως *Λανθάνουσα Σημασιολογική Ανάλυση* (*Latent Semantic Analysis - LSA*).

Η παραπάνω μέθοδος εξαγωγής διανυσματικών αναπαραστάσεων δώθηκε συνοπτικά για να γίνει μία σύνδεση των word vectors με τη μέθοδο Bag-of-Words και να γίνει πιο σαφής η ιδέα της απεικόνισης λέξεων σε διανυσματικούς χώρους με τη χρήση στατιστικών μετρήσεων για την γειτνίασή τους. Στη συνέχεια θα εξεταστεί ο αλγόριθμος word2vec, ένα αρκετά πιο σύνθετο σύστημα που ανήκει στην κατηγορία των νευρωνικών γλωσσικών μοντέλων. Αυτά τα μοντέλα είναι νευρωνικά δίκτυα που εκπαιδεύονται χωρίς επίβλεψη σε δεδομένα κειμένου. Δεδομένων κάποιων λέξεων σε ένα παράθυρο του συνολικού κειμένου, προσπαθούν να προβλέψουν μία νέα λέξη και προσαρμόζουν τα βάρη τους με στόχο τη μεγιστοποίηση της softmax πιθανότητας η νέα λέξη να είναι η λέξη που συναντάται στα δεδομένα.

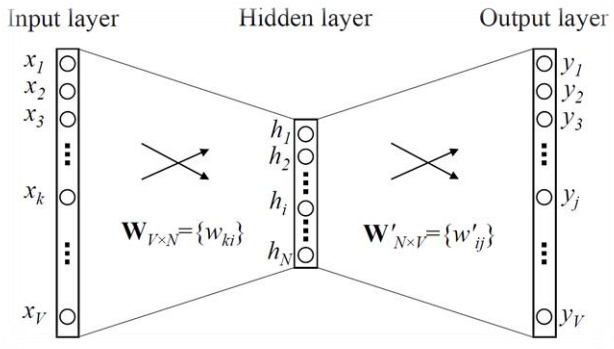
4.2.2 Το Μοντέλο word2vec

Το μοντέλο word2vec είναι ένα νευρωνικό γλωσσικό μοντέλο (neural language model) που χρησιμοποείται για την απεικόνιση λέξεων σε διανυσματικούς χώρους μικρής διάστασης. Προτάθηκε από τους Mikolov et al. [12] το 2013 και προσέλκυσε το ακαδημαϊκό ενδιαφέρον εξαιτίας της ποιότητας των διανυσματικών αναπαραστάσεων που παράγει. Το μοντέλο word2vec είναι ικανό να ανακαλύπτει πολύπλοκες σημασιολογικές και συντακτικές σχέσεις μεταξύ των λέξεων και να τις μετατρέπει σε γραμμικές ιδιότητες του διανυσματικού χώρου. Γενικά πρόκειται για ένα window-based γλωσσικό μοντέλο πρόβλεψης που βασίζεται στην αρχιτεκτονική ενός απλού νευρωνικού δικτύου.

Οι Mikolov et al. στο [12] δίνουν δύο διαφορετικές υλοποιήσεις του μοντέλου word2vec όπου η πρώτη καλείται ***Continuous Bag-of-Words*** (CBOW) και βασίζεται στην πρόβλεψη της κεντρικής λέξης (***central word***) δεδομένων των λέξεων γύρω από αυτή (***context words***) και η δεύτερη ***Skip-Gram*** (SG) και βασίζεται στην πρόβλεψη των context words δεδομένης της λέξης στο κέντρο. Για την θεωρητική ανάλυση του αλγορίθμου στη συνέχεια, δανείζονται στοιχεία από την δημοσίευση του Xin Rong [21] πάνω στον αλγόριθμο word2vec.

4.2.2.1 Continuous Bag-of-Words

Δεδομένου ενός σώματος κειμένου, ο αλγόριθμος εργάζεται σε παράθυρα όπου κάθε παράθυρο περιλαμβάνει μία κεντρική λέξη και c λέξεις δεξιά και αριστερά της κεντρικής. Από το σώμα κειμένου κατασκευάζεται το λεξικό που υποθέτουμε ότι έχει μήκος V. Σε κάθε λέξη του λεξικού αρχικά αντιστοιχίζονται τυχαία διανύσματα μήκους N και ο σκοπός είναι η εκπαίδευση των διανυσμάτων αυτών έτσι ώστε να ελαχιστοποιείται ένα κόστος πρόβλεψης. Η πρόβλεψη στην περίπτωση CBOW αφορά την κεντρική λέξη δεδομένων των γειτονικών λέξεων. Η εκπαίδευση γίνεται με την βοήθεια ***stochastic gradient descent*** και ***backpropagation*** μεθόδων Η αρχιτεκτονική του δικτύου για την απλή περίπτωση μίας context λέξης φαίνεται στο σχήμα 4.2.



Σχήμα 4.2

Η είσοδος του δικτύου είναι η one-hot αναπαράσταση $\mathbf{x}$ της context λέξης w_I, δηλαδή ένα διάνυσμα μήκους V με 1 σε μία θέση και 0 σε όλες τις υπόλοιπες. Το κρυφό επίπεδο έχει μήκος N. Η έξοδος έχει μήκος V και είναι ένα διάνυσμα πιθανοτήτων. Κάθε έξοδος y_j είναι η πιθανότητα η κεντρική λέξη να είναι η j λέξη του λεξικού δεδομένου ότι η context λέξη είναι στην είσοδο. Οι πιθανότητες αυτές προσδιορίζονται με τη βοήθεια της *softmax συνάρτησης*. Μεταξύ των επιπέδων υπάρχουν τα βάρη $\mathbf{W}_{V\times N} = \{w_{ki}\}$ και $\mathbf{W'}_{N\times V} = \{w'_{ij}\}$, όπου τα μεν μετασχηματίζουν την είσοδο στο κρυφό επίπεδο, τα δε το κρυφό επίπεδο στην έξοδο.

Έστω ότι η context λέξη είναι η k λέξη του λεξικού. Τότε για την αναπαράσταση $\mathbf{x}$ θα ισχύει $x_k = 1$ και $x_{k'} = 0$ για $k' \neq k$ οπότε ο πολλαπλασιασμός της εισόδου με τα βάρη $\mathbf{W}$ θα δίνει την k γραμμή του πίνακα βαρών.

$$\mathbf{h} = (h_1, h_2, \dots, h_N) = \mathbf{x}^T\mathbf{W} = \mathbf{W}_{(k,\cdot)} = \mathbf{v}_k = \mathbf{v}_{w_I} \tag{1}$$

Ο πίνακας $\mathbf{W}$ δηλαδή, περιέχει N-διάστατα διανύσματα $\mathbf{v}_k$ για κάθε λέξη του λεξικού και με την εφαρμογή μιας one-hot αναπαράστασης στην είσοδο για την context λέξη w_k, το αντίστοιχο διάνυσμα $\mathbf{v}_k$ προωθείται στο κρυφό επίπεδο. Στη συνέχεια το κρυφό επίπεδο μετασχηματίζεται στο τοπικό πεδίο $\mathbf{u}$

$$\mathbf{u} = \mathbf{W'}^T \cdot \mathbf{h} \quad \Rightarrow \quad u_j = \mathbf{v'}_{w_j}{}^T \cdot \mathbf{h} \tag{2}$$

και η έξοδος $\mathbf{y}$ προκύπτει από την εφαρμογή της μη γραμμικής softmax συνάρτησης στο τοπικό πεδίο, δηλαδή

$$\mathbf{y} = softmax(\mathbf{u}) \quad \Rightarrow \quad y_j = \frac{\exp(u_j)}{\sum_{j'=1}^{V}\exp(u_{j'})} \tag{3}$$

Η έξοδος y_j όπως είπαμε εκφράζει την πιθανότητα η κεντρική λέξη w_O να είναι η λέξη w_j του λεξικού δεδομένου ότι η context λέξη w_I είναι η λέξη w_k.

$$y_j = P(w_O = w_j | w_I = w_k) = \frac{\exp(u_j)}{\sum_{j'=1}^{V}\exp(u_{j'})}$$

Αντικαθιστώντας τις σχέσεις (1) και (2) στην (3) παίρνουμε

$$P(w_O = w_j | w_I = w_k) = P(w_j | w_I) = \frac{\exp(\mathbf{v'}_{w_j}{}^T \cdot \mathbf{v}_{w_I})}{\sum_{j'=1}^{V}\exp(\mathbf{v'}_{w_j}{}^T \cdot \mathbf{v}_{w_I})} \tag{4}$$

Ας επισημάνουμε σε αυτό το σημείο κάποια βασικά χαρακτηριστικά του μοντέλου. Το μοντέλο αναθέτει σε κάθε λέξη του λεξικού δύο διανύσματα, το $\mathbf{v}$ που είναι κομμάτι των βαρών $\mathbf{W}$ και το $\mathbf{v'}$ των βαρών $\mathbf{W'}$. Δεδομένου ότι η context λέξη ή αλλιώς η λέξη εισόδου, είναι κάποια λέξη w_I του λεξικού, η πιθανότητα η κεντρική λέξη, ή αλλιώς λέξη εξόδου, w_O να είναι η w_j του λεξικού δίνεται από την (4). Επειδή τα τονούμενα διανύσματα $\mathbf{v'}$ στην (4)

αναφέρονται πάντα σε λέξεις εξόδου, χαρακτηρίζονται *διανύσματα εξόδου* (***output vectors***) και αντίστροφα τα $\mathbf{v}$ *διανύσματα εισόδου* (***input vectors***). Κάθε λέξη w λοιπόν χαρακτηρίζεται από ένα διάνυσμα εισόδου $\mathbf{v}_w$ το οποίο υπεισέρχεται στους υπολογισμούς όταν η λέξη δεν είναι στο context και παράλληλα από ένα διάνυσμα εξόδου $\mathbf{v}'_w$ που καλείται στην εξίσωση (4) όταν η λέξη είναι στο context. Ο σκοπός του δικτύου είναι η προσαρμογή των διανυσμάτων αυτών με κριτήριο τη μεγιστοποίηση της πιθανότητας παρατήρησης της πραγματικής κεντρικής λέξης. Μετά την ολοκλήρωση της εκπαίδευσης τα διανύσματα αυτά θα αποτελούν τα word vectors. Συνεπώς η διάσταση των word vectors είναι N και είναι υπερπαράμετρος του δικτύου.

Το κριτήριο της εκπαίδευσης είναι η μεγιστοποίηση της πιθανότητας

$$P(w_O = w_{j^*}|w_I) = y_{j^*} = \frac{\exp(\mathbf{v}'_{w_{j^*}}{}^T \cdot \mathbf{v}_{w_I})}{\sum_{j'=1}^{V} \exp(\mathbf{v}'_{w_{j'}}{}^T \cdot \mathbf{v}_{w_I})}$$

όπου με j^* δηλώνεται ο δείκτης της λέξης, που πράγματι παρατηρείται σαν κεντρική, στο λεξικό. Η μεγιστοποίηση της παραπάνω πιθανότητας ισοδυναμεί με μεγιστοποίηση του λογαρίθμου της. Συνεπώς για λόγους διευκόλυνσης των πράξεων, το κριτήριο γίνεται

$$\max\left(\log\left(P(w_O = w_{j^*}|w_I)\right)\right) \Rightarrow$$

$$\max(\log(y_{j^*})) \Rightarrow$$

$$\max\left(\log\left(\frac{\exp\left(\mathbf{v}'_{w_{j^*}}{}^T \cdot \mathbf{v}_{w_I}\right)}{\sum_{j'=1}^{V} \exp\left(\mathbf{v}'_{w_{j'}}{}^T \cdot \mathbf{v}_{w_I}\right)}\right)\right) \Rightarrow$$

$$\max\left(\log\left(\exp\left(\mathbf{v}'_{w_{j^*}}{}^T \cdot \mathbf{v}_{w_I}\right)\right) - \log\left(\sum_{j'=1}^{V} \exp\left(\mathbf{v}'_{w_{j'}}{}^T \cdot \mathbf{v}_{w_I}\right)\right)\right) \Rightarrow$$

$$\max\left(\mathbf{v}'_{w_{j^*}}{}^T \cdot \mathbf{v}_{w_I} - \log\left(\sum_{j'=1}^{V} \exp\left(\mathbf{v}'_{w_{j'}}{}^T \cdot \mathbf{v}_{w_I}\right)\right)\right) \Rightarrow$$

$$\max\left(u_{j_*} - \log\left(\sum_{j'=1}^{V} \exp(u_{j'})\right)\right)$$

Οι παράμετροι του δικτύου είναι τα διανύσματα $\mathbf{v}_w$ και $\mathbf{v}'_w$ για κάθε λέξη w του λεξικού ή εναλλακτικά οι πίνακες $\mathbf{W} = \{w_{ki}\}$ και $\mathbf{W}' = \{w'_{ij}\}$. Όπως υπαγορεύει η αρχή gradient descent οι παράμετροι ενημερώνονται προς την κατεύθυνση που ελαχιστοποιείται η συνάρτηση κόστους. Ορίζοντας σαν συνάρτηση κόστους την

$$\mathcal{E} = -\left(u_{j_*} - \log\left(\sum_{j'=1}^{V} \exp(u_{j'}) \right) \right) \tag{5}$$

δηλαδή το αρνητικό της παραπάνω ποσότητας, πλέον κριτήριο γίνεται η ελαχιστοποίηση της $\mathcal{E}$. Οι παράγωγοι της συνάρτησης $\mathcal{E}$ ως προς τα βάρη είναι

$$\frac{\partial \mathcal{E}}{\partial w'_{ij}} = \frac{\partial \mathcal{E}}{\partial u_j} \cdot \frac{\partial u_j}{\partial w'_{ij}} \tag{6}$$

όπου από την (5) έχουμε

$$\frac{\partial \mathcal{E}}{\partial u_j} = -t_j + \frac{\partial}{\partial u_j} \log\left(\sum_{j'=1}^{V} \exp(u_{j'}) \right) = -t_j + \frac{\exp(u_j)}{\sum_{j'=1}^{V} \exp(u_{j'})} = -t_j + y_j = e_j \tag{7}$$

με $t_j = \begin{cases} 0\,, & \alpha\nu\, j \neq j^* \\ 1\,, & \alpha\nu\, j = j^* \end{cases}$. Επίσης από την (2)

$$u_j = \mathbf{v}'^{\,T}_{w_j} \cdot \mathbf{h} = \sum_{i=1}^{N} w'_{ij} h_i \quad \Rightarrow \quad \frac{\partial u_j}{\partial w'_{ij}} = h_i$$

Επιστρέφοντας στην (6) παίρνουμε

$$\frac{\partial \mathcal{E}}{\partial w'_{ij}} = e_j h_i$$

και η ανανέωση των βαρών θα γίνεται σύμφωνα με τον κανόνα

$$w'_{ij}(n+1) = w'_{ij}(n) - \eta e_j h_i$$

όπου η ο ρυθμός μάθησης. Επεκτείνοντας την παραπάνω σχέση για τα διανύσματα $\mathbf{v}'_{w_j}$ ο κανόνας προσαρμογής γίνεται

$$\mathbf{v}'_{w_j}(n+1) = \mathbf{v}'_{w_j}(n) - \eta e_j \mathbf{h} \tag{8}$$

Η ενημέρωση των βαρών w_{ki} γίνεται με τη βοήθεια της παραγώγου του κόστους $\mathcal{E}$ ως προς τα βάρη. Το βάρος w_{ki} συνεισφέρει στην παραγωγή του λειτουργικού σήματος h_i το οποίο με τη σειρά του συνεισφέρει σε όλα τα τοπικά πεδία u_j στην έξοδο. Το ίδιο πρόβλημα αντιμετωπίστηκε στον αλγόριθμο Backpropagation για την ενημέρωση βαρών σε κρυφούς νευρώνες.

$$\frac{\partial \mathcal{E}}{\partial w_{ki}} = \frac{\partial \mathcal{E}}{\partial h_i} \cdot \frac{\partial h_i}{\partial w_{ki}} = \frac{\partial h_i}{\partial w_{ki}} \cdot \sum_{j=1}^{V} \frac{\partial \mathcal{E}}{\partial u_j} \cdot \frac{\partial u_j}{\partial h_i}$$

Από τις σχέσεις (7), (2) και (1) έχουμε

$$\frac{\partial \mathcal{E}}{\partial u_j} = e_j \qquad \frac{\partial u_j}{\partial h_i} = w'_{ij} \qquad \frac{\partial h_i}{\partial w_{ki}} = 1$$

Επομένως

$$\frac{\partial \mathcal{E}}{\partial w_{ki}} = \sum_{j=1}^{V} e_j \, w'_{ij}$$

και η ανανέωση των βαρών w_{ki} και των διανυσμάτων $\mathbf{v}_{w_I}$ γίνεται ως εξής

$$w_{ki}(n+1) = w_{ki}(n) - \eta \sum_{j=1}^{V} e_j \, w'_{ij}$$

$$\mathbf{v}_{w_I}(n+1) = \mathbf{v}_{w_I}(n) - \eta \mathbf{EH} \tag{9}$$

όπου **EH** είναι το διάνυσμα N διαστάσεων του οποίου κάθε στοιχείο είναι η παράγωγος που προσδιορίσαμε και χρησιμοποιείται στην ενημέρωση βαρών, δηλαδή

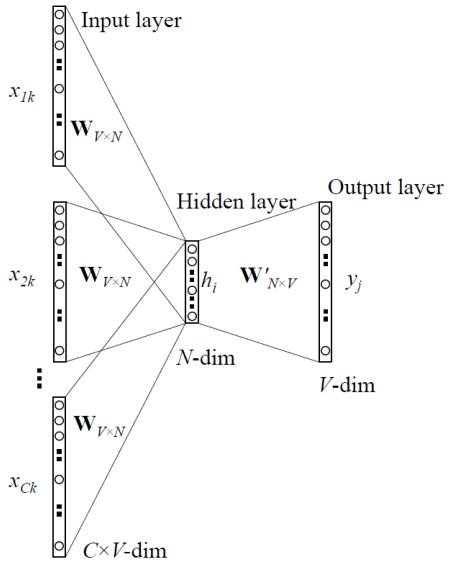
$$EH_i = \sum_{j=1}^{V} e_j \, w'_{ij}$$

Όπως προκύπτει από τις εξισώσεις, όλα τα διανύσματα εξόδου $\mathbf{v}'_{w_j}$ για κάθε λέξη w_j του λεξικού, πολλαπλασιάζονται με το αντίστοιχο σφάλμα e_j και προστίθενται για να προκύψει το διάνυσμα **EH**. Συνεπώς σε κάθε παράθυρο, το διάνυσμα εισόδου της context λέξης μετατοπίζεται στην αρνητική κατεύθυνση του σταθμισμένου μέσου όλων των διανυσμάτων εξόδου του λεξικού. Όσο μεγαλύτερο το σφάλμα για κάποια λέξη w_j τόσο μεγαλύτερη η συνεισφορά του διανύσματος εξόδου της συγκεκριμένης λέξης στην κατεύθυνση που θα κινηθεί το διάνυσμα εισόδου τις context λέξης.

Οι σχέσεις (8) και (9) αποτελούν τον κανόνα ενημέρωσης των βαρών του δικτύου με βάση την μέθοδο gradient descent. Παρουσιάζουν σημαντικά προβλήματα υλοποίησης τα οποία θα συζητηθούν στη συνέχεια. Προτού αναφερθούμε σε αυτά, εξετάζουμε την γενικότερη περίπτωση πολλών context λέξεων.

Η γενικότερη περίπτωση του CBOW μοντέλου word2vec

Στο σχήμα 4.3 φαίνεται η αρχιτεκτονική του δικτύου για C context λέξεις. Παρατηρούμε ότι το δίκτυο δεν αλλάζει κατά την μετάβαση από το κρυφό επίπεδο στην έξοδο. Για την μετάβαση από την είσοδο στο κρυφό επίπεδο, πλέον δεν αντιγράφεται το διάνυσμα εισόδου της context λέξης στο κρυφό επίπεδο όπως πριν, αλλά με γραμμικό τρόπο τα διανύσματα εισόδου και των C context λέξεων προστίθενται, σταθμίζονται και μεταβιβάζονται στο κρυφό επίπεδο.



Σχήμα 4.3

Δηλαδή

$$\mathbf{h} = \frac{1}{C}\mathbf{W}(\mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_C)$$

$$\mathbf{h} = \frac{1}{C}(\mathbf{v}_{w_1} + \mathbf{v}_{w_2} + \cdots + \mathbf{v}_{w_C})$$

Σημειώνεται ότι όπως και πριν οι αναπαραστάσεις **x** των context λέξεων είναι one-hot.

Η συνάρτηση κόστους προς μεγιστοποίηση πλέον γίνεται

$$\log\left(P(w_O = w_{j^*}|w_{I,1}, w_{I,2}, \dots, w_{I,C})\right) = \log(y_{j^*}) = \log\left(\frac{\exp(u_{j^*})}{\sum_{j'=1}^{V}\exp(u_{j'})}\right) =$$

$$= u_{j^*} - \log\left(\sum_{j'=1}^{V}\exp(u_{j'})\right) = \mathbf{v}'_{w_{j^*}}{}^T \cdot \mathbf{h} - \log\left(\sum_{j'=1}^{V}\exp\left(\mathbf{v}'_{w_{j'}}{}^T \cdot \mathbf{h}\right)\right)$$

Επιδιώκεται η ελαχιστοποίηση της αντίθετης συνάρτησης $\mathcal{E}$

$$\mathcal{E} = \log\left(\sum_{j'=1}^{V}\exp\left(\mathbf{v}'_{w_{j'}}{}^T \cdot \mathbf{h}\right)\right) - \mathbf{v}'_{w_{j'}}{}^T \cdot \mathbf{h}$$

Η συνάρτηση κόστους είναι ίδια με την προηγούμενη περίπτωση αλλά τώρα το διάνυσμα **h** ορίζεται διαφορετικά. Παίρνοντας παραγώγους της συνάρτησης $\mathcal{E}$ εύκολα καταλήγουμε στις ίδιες σχέσεις με τα προηγούμενα

$$\frac{\partial \mathcal{E}}{\partial w'_{ij}} = e_j h_i$$

$$\frac{\partial \mathcal{E}}{\partial w_{ki}} = \frac{\partial \mathcal{E}}{\partial h_i}\cdot\frac{\partial h_i}{\partial w_{ki}} = \frac{\partial h_i}{\partial w_{ki}}\cdot\sum_{j=1}^{V}\frac{\partial \mathcal{E}}{\partial u_j}\cdot\frac{\partial u_j}{\partial h_i}$$

αλλά με τη διαφοροποίηση

$$\mathbf{h} = \frac{1}{C}\mathbf{W}(\mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_C) \quad \Rightarrow \quad \frac{\partial h_i}{\partial w_{ki}} = \frac{1}{C}$$

Στην περίπτωση λοιπόν του context C λέξεων, τα βάρη και τα διανύσματα ενημερώνονται με τους ακόλουθους κανόνες

$$w'_{ij}(n+1) = w'_{ij}(n) - \eta e_j h_i$$

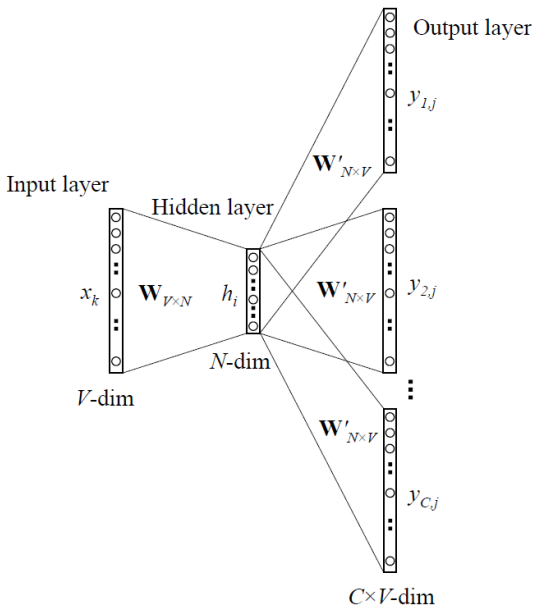
$$\mathbf{v}'_{w_j}(n+1) = \mathbf{v}'_{w_j}(n) - \eta e_j \mathbf{h}$$

$$w_{ki}(n+1) = w_{ki}(n) - \frac{1}{C}\eta \sum_{j=1}^{V} e_j \, w'_{ij}$$

$$\mathbf{v}_{w_{I,c}}(n+1) = \mathbf{v}_{w_{I,c}}(n) - \frac{1}{C}\eta \mathbf{EH}$$

4.2.2.2 Skip-Gram

Στην Skip-Gram εκδοχή του μοντέλου word2vec το νευρωνικό δίκτυο εκπαιδεύεται στην πρόβλεψη των context λέξεων δεδομένης της κεντρικής λέξης. Η αρχιτεκτονική του δικτύου για C context λέξεις δίνεται στο σχήμα 4.4.



Σχήμα 4.4

Η είσοδος είναι η one-hot αναπαράσταση $\mathbf{x}$ της κεντρικής λέξης η οποία πολλαπλασιάζεται με τον $V \times N$ πίνακα βαρών $\mathbf{W}$. Ο πίνακας αυτός περιέχει N-διάστατα διανύσματα $\mathbf{v}_w$ για κάθε λέξη του λεξικού και συνεπώς ο πολλαπλασιασμός του με την one-hot αναπαράσταση $\mathbf{x}$ απλώς αντιγράφει στο κρυφό επίπεδο το διάνυσμα της λέξης εισόδου $\mathbf{v}_{w_I}$. Συνεπώς, αν θεωρήσουμε ότι η λέξη στην είσοδο είναι η k λέξη του λεξικού, δηλαδή το $\mathbf{x}$ είναι διάνυσμα μηδενικών με 1 μόνο στη θέση k, τότε

$$\mathbf{h} = \mathbf{W}_{(k,\cdot)} = \mathbf{v}_k = \mathbf{v}_{w_I}$$

Η έξοδος είναι ένα διάνυσμα διάστασης $C \cdot V$ που περιέχει πιθανότητες για κάθε context λέξη και για όλες τις λέξεις του λεξικού. Έτσι ως $y_{c,j}$ χαρακτηρίζεται η πιθανότητα η context λέξη c να είναι η λέξη j του λεξικού με $c = 1,2,\ldots,C$ και $j = 1,2,\ldots,V$. Η πιθανότητα αυτή ορίζεται και πάλι ως η softmax συνάρτηση του τοπικού πεδίου. Δηλαδή

$$y_{c,j} = P(w_{O,c} = w_j | w_I) = \frac{\exp(u_{c,j})}{\sum_{j'=1}^{V} \exp(u_{c,j'})}$$

Τα τοπικά πεδία παράγονται κατά τη μετάβαση από το κρυφό επίπεδο στην έξοδο. Μεσολαβεί ο πίνακας βαρών $\mathbf{W}'$ που περιέχει τα $\mathbf{v}'_w$ διανύσματα για κάθε λέξη του λεξικού. Όπως δείχνει και το σχήμα ο πίνακας αυτός χρησιμοποιείται για την παραγωγή των τοπικών πεδίων ανεξάρτητα από την context λέξη στην οποία αναφερόμαστε δηλαδή τα πεδία $u_{c,j}$ είναι ανεξάρτητα του c. Έχουμε λοιπόν

$$\mathbf{u}_c = \mathbf{u} = \mathbf{W}'^T \mathbf{h} \quad \Rightarrow \quad u_{c,j} = u_j = \mathbf{v}'^{\,T}_{w_j} \cdot \mathbf{h}$$

Ο στόχος της εκπαίδευσης στο SG μοντέλο είναι η μεγιστοποίηση της πιθανότητας οι context λέξεις να είναι οι λέξεις που πράγματι παρατηρούνται στο δείγμα. Ορίζοντας ως $\mathcal{E}$ το αρνητικό του λογαρίθμου της παραπάνω πιθανότητας, όπως εργαστήκαμε και προηγουμένως, κριτήριο γίνεται πλέον η ελαχιστοποίηση της ποσότητας $\mathcal{E}$.

$$\mathcal{E} = -\log\left(P(w_{O,1}, w_{O,2}, \dots, w_{O,C} | w_I)\right) = -\log\left(\prod_{c=1}^{C} P(w_{O,c} | w_I)\right) = -\sum_{c=1}^{C} \log\left(P(w_{O,c} | w_I)\right)$$

$$\mathcal{E} = -\sum_{c=1}^{C} \log\left(\frac{\exp(u_{c,j^*})}{\sum_{j'=1}^{V} \exp(u_{c,j'})}\right) = -\sum_{c=1}^{C} \left(\log(\exp(u_{j_c^*})) - \log\left(\sum_{j'=1}^{V} \exp(u_{j'})\right)\right)$$

$$\mathcal{E} = -\sum_{c=1}^{C} \left(u_{j_c^*} - \log\left(\sum_{j'=1}^{V} \exp(u_{j'})\right)\right) = C \cdot \log\left(\sum_{j'=1}^{V} \exp(u_{j'})\right) - \sum_{c=1}^{C} u_{j_c^*}$$

Με $u_{j_c^*}$ δηλώνεται το τοπικό πεδίο στο νευρώνα j_c^* που αντιστοιχεί στην λέξη που παρατηρείται στο δείγμα για το context c. Στη συνέχεια λαμβάνονται οι παράγωγοι της συνάρτησης $\mathcal{E}$ ως προς τα βάρη. Δουλεύοντας όπως στα προηγούμενα προκύπτει

$$\frac{\partial \mathcal{E}}{\partial w'_{ij}} = \sum_{c=1}^{C} \frac{\partial \mathcal{E}}{\partial u_{c,j}} \cdot \frac{\partial u_{c,j}}{\partial w'_{ij}} = \sum_{c=1}^{C} (y_{c,j} - t_{c,j}) h_i = \sum_{c=1}^{C} e_{c,j} h_i$$

και

$$\frac{\partial \mathcal{E}}{\partial w_{ki}} = \sum_{j=1}^{V} \left(\sum_{c=1}^{C} e_{c,j}\right)$$

Με βάση λοιπόν τα παραπάνω η προσαρμογή των βαρών γίνεται σύμφωνα με τις εξισώσεις

$$w'_{ij}(n+1) = w'_{ij}(n) - \eta \sum_{c=1}^{C} e_{c,j} h_i$$

$$\mathbf{v}'_{w_j}(n+1) = \mathbf{v}'_{w_j}(n) - \eta \sum_{c=1}^{C} e_{c,j} \mathbf{h}$$

$$w_{ki}(n+1) = w_{ki}(n) - \eta \sum_{j=1}^{V} (\sum_{c=1}^{C} e_{c,j})\, w'_{ij}$$

$$\mathbf{v}_{w_I}(n+1) = \mathbf{v}_{w_I}(n) - \eta \mathbf{EH}$$

όπου

$$\mathbf{EH}_i = \sum_{j=1}^{V} (\sum_{c=1}^{C} e_{c,j})\, w'_{ij}$$

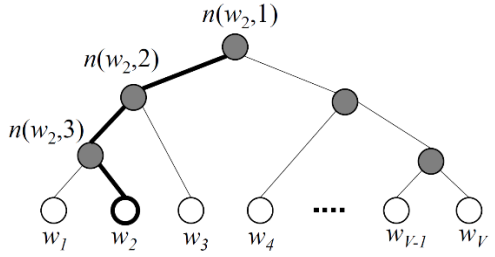
4.2.3 Μείωση της Πολυπλοκότητας του Αλγορίθμου word2vec

Σε αυτό το σημείο θα εξεταστεί η υπολογιστική πολυπλοκότητα του αλγορίθμου word2vec. Στην προηγούμενη ενότητα είδαμε το μοντέλο σαν ένα ρηχό νευρωνικό δίκτυο με ένα κρυφό επίπεδο που προσαρμόζει τα word vectors με σκοπό τη μεγιστοποίηση της πιθανότητας οι λέξεις να εμφανίζονται με τη σειρά που πράγματι εμφανίζονται στο δείγμα κειμένου που χρησιμοποιείται για την εκπαίδευση. Το μοντέλο word2vec αναθέτει σε κάθε λέξη w του λεξικού δύο διανύσματα, το διάνυσμα εισόδου $\mathbf{v}_w$ και το διάνυσμα εξόδου $\mathbf{v}'_w$ και στις εξισώσεις που δόθηκαν δίνεται ο κανόνας προσαρμογής των διανυσμάτων αυτών. Μία προσεκτική ματιά στις εξισώσεις αυτές αποκαλύπτει ένα σοβαρό πρόβλημα του αλγορίθμου. Στην περίπτωση Continuous Bag-of-Words έχουμε στην είσοδο τις context λέξεις και στην έξοδο την κεντρική λέξη. Οι εξισώσεις υπαγορεύουν την ενημέρωση των διανυσμάτων εισόδου για τις context λέξεις και την ενημέρωση των διανυσμάτων εξόδου για όλες τις λέξεις του λεξικού. Στο Skip-Gram μοντέλο η είσοδος είναι η κεντρική λέξη και η έξοδος οι context λέξεις. Όμοια απαιτείται η ενημέρωση του διανύσματος εισόδου μόνο για την κεντρική λέξη και η ενημέρωση των διανυσμάτων εξόδου για όλες τις λέξεις του λεξικού. Η προσαρμογή λοιπόν των διανυσμάτων εξόδου είναι πολύ ακριβή υπολογιστικά και καθιστά αδύνατη την εφαρμογή του αλγορίθμου σε μεγάλα σώματα κειμένου με λεξικό χιλιάδων ή εκατομμυρίων λέξεων. Λύση σε αυτό το πρόβλημα δίνουν οι δύο εναλλακτικές τεχνικές που προτείνονται από τους Mikolov et al. στο [13] και καλούνται Hierarchical Softmax και Negative Sampling.

4.2.3.1 Hierarchical Softmax

Hierarchical Softmax καλείται ένας εναλλακτικός τρόπος υπολογισμού της softmax πιθανότητας που κάνει χρήση δυαδικών δέντρων. Στο σχήμα 4.5 φαίνεται η δενδρική δομή για τον υπολογισμό των softmax πιθανοτήτων λέξεων ενός λεξικού πλήθους V.

Κάθε λέξη του λεξικού είναι ένα φύλλο του δέντρου. Το δέντρο έχει V φύλλα και $V-1$ εσωτερικούς κόμβους. Το δέντρο είναι δυαδικό και για κάθε φύλλο υπάρχει ένα μοναδικό μονοπάτι που οδηγεί από την ρίζα σε αυτό. Με $n(w,j)$ συμβολίζεται ο j κόμβος στο μονοπάτι από τη ρίζα προς το φύλλο w. Με $L(w)$ συμβολίζεται το μήκος του μονοπατιού από τη ρίζα μέχρι το φύλλο w και με $ch(n)$ το αριστερό παιδί του κόμβου n. Για παράδειγμα $L(w_3)=3$ και $ch\big(n(w_2,2)\big)=n(w_2,3)$. Σε αυτή την διάταξη η πιθανότητα εμφάνισης του φύλλου w είναι



Σχήμα 4.5

$$P(w)=\prod_{j=1}^{L(w)-1}\sigma([\![n(w,j+1)=ch(n(w,j))]\!]\cdot \mathbf{w}_{n(w,j)}^{T}\mathbf{h})$$

όπου $\mathbf{w}_{n(w,j)}$ είναι μία διανυσματική αναπαράσταση του κόμβου $n(w,j)$, $\mathbf{h}$ η έξοδος του κρυφού επιπέδου, σ η λογιστική συνάρτηση, και $[\![a]\!]$ μία ειδική συνάρτηση με τιμή 1 εάν η πρόταση α είναι αληθής και -1 εάν είναι ψευδής. Στην περίπτωση του μοντέλου word2vec το επίπεδο hierarchical softmax εφαρμόζεται μετά το κρυφό επίπεδο, στη θέση του απλού softmax που είδαμε μέχρι τώρα. Η πιθανότητα αναφέρεται στην πιθανότητα μία λέξη του λεξικού να έιναι η λέξη εξόδου, δηλαδή η κεντρική λέξη στην περίπτωση CBOW και μία εκ των context λέξεων στην περίπτωση SG. Η παραπάνω σχέση λαμβάνοντας υπόψιν και τους συμβολισμούς της προηγούμενης ενότητας για το μοντέλο word2vec γίνεται

$$P(w_O=w)=\prod_{j=1}^{L(w)-1}\sigma([\![n(w,j+1)=ch(n(w,j))]\!]\cdot {\mathbf{v}'_{n(w,j)}}^{T}\mathbf{h}) \quad (1)$$

Παρατηρούμε αρχικά ότι πλέον διανύσματα εξόδου $\mathbf{v}'$ ανατίθενται στους $V-1$ κόμβους και όχι στις λέξεις του λεξικού. Επίσης για τον υπολογισμό της πιθανότητας για μία λέξη του λεξικού απαιτούνται $L(w)-1$ βήματα και όχι άθροιση σε όλο το λεξικό για την κανονικοποίηση της softmax όπως φαίνεται στη σχέση (1). Αναφορικά με το διάνυσμα $\mathbf{h}$

στο μοντέλο word2vec όπως είδαμε, είνα $\mathbf{h} = \mathbf{v}_{w_I}$ για την SG εκδοχή και $\mathbf{h} = \frac{1}{C}\sum_{c=1}^{C}\mathbf{v}_{w_{I,c}}$ για την CBOW.

Για τον υπολογισμό της πιθανότητας του φύλλου w όπως υποδεικνύει η (1) ακολουθείται το μονοπάτι που οδηγεί από τη ρίζα στο w και για κάθε πέρασμα κόμβου κατά τη διάρκεια του μονοπατιού, υπολογίζεται το εσωτερικό γινόμενο του διανύσματος εξόδου του κόμβου με το κρυφό επίπεδο. Τελικά όλα τα εσωτερικά γινόμενα που προκύπτουν εφαρμόζονται στη λογιστική συνάρτηση και τα αποτελέσματα πολλαπλασιάζονται για να προκύψει η τελική πιθανότητα. Επίσης, όπως δηλώνει η συνάρτηση $[\![\cdot]\!]$ και η εσωτερική της συνθήκη, όταν γίνεται μετάβαση από κόμβο σε αριστερό παιδί υπολογίζουμε το εσωτερικό γινόμενο αυτούσιο ενώ όταν μετάβαση γίνεται προς το δεξί παιδί του υπό εξέταση κόμβου, υπολογίζεται το αρνητικό του εσωτερικού γινομένου.

Ας επιστρέψουμε στην πλέον απλή περίπτωση του CBOW μοντέλου word2vec με μία context λέξη για να δούμε πώς η εισαγωγή του νέου hierarchical softmax επιπέδου επηρεάζει την πολυπλοκότητα της ενημέρωσης των διανυσμάτων εξόδου. Αρχικά εξετάζουμε την συνάρτηση κόστους υπό το νέο πρίσμα. Όπως είχαμε δει προηγουμένως

$$\mathcal{E} = -\log\big(P(w_O = w|w_I)\big) = -\log\left(\prod_{j=1}^{L(w)-1}\sigma\big([\![n(w,j+1) = ch\big(n(w,j)\big)]\!]\cdot \mathbf{v}'_{n(w,j)}{}^{T}\mathbf{h}\big)\right)$$

$$\mathcal{E} = -\sum_{j=1}^{L(w)-1}\log\Big(\sigma\big([\![n(w,j+1) = ch\big(n(w,j)\big)]\!]\cdot \mathbf{v}'_{n(w,j)}{}^{T}\mathbf{h}\big)\Big)$$

Τα εσωτερικά γινόμενα $\mathbf{v}'_{n(w,j)}{}^{T}\mathbf{h}$ είναι ανάλογα των τοπικών πεδίων που είχαν οριστεί στην απλή softmax περίπτωση. Για την ενημέρωση των βαρών κατεφεύγουμε και πάλι στις παραγώγους της συνάρτησης κόστους ως προς τα διανύσματα εισόδου και εξόδου. Λαμβάνουμε

$$\frac{\partial\mathcal{E}}{\partial\mathbf{v}'_{n(w,j)}} = \frac{\partial\mathcal{E}}{\partial\mathbf{v}'_{n(w,j)}{}^{T}\mathbf{h}}\cdot\frac{\partial\mathbf{v}'_{n(w,j)}{}^{T}\mathbf{h}}{\partial\mathbf{v}'_{n(w,j)}} =$$

$$= \left(-\frac{\sigma'\big([\![n(w,j+1) = ch\big(n(w,j)\big)]\!]\cdot \mathbf{v}'_{n(w,j)}{}^{T}\mathbf{h}\big)\cdot[\![n(w,j+1) = ch\big(n(w,j)\big)]\!]}{\sigma\big([\![n(w,j+1) = ch\big(n(w,j)\big)]\!]\cdot \mathbf{v}'_{n(w,j)}{}^{T}\mathbf{h}\big)}\right)\cdot\mathbf{h}$$

Κάνοντας χρήση της ιδιότητας $\sigma'(x) = \sigma(x)\big(1-\sigma(x)\big)$ της λογιστικής συνάρτησης η παραπάνω σχέση γίνεται

$$\frac{\partial\mathcal{E}}{\partial\mathbf{v}'_{n(w,j)}} = -\Big(1-\sigma\big([\![n(w,j+1) = ch\big(n(w,j)\big)]\!]\cdot \mathbf{v}'_{n(w,j)}{}^{T}\mathbf{h}\big)\Big)[\![n(w,j+1) = ch\big(n(w,j)\big)]\!]\mathbf{h}$$

Εάν $[\![n(w,j+1)=ch(n(w,j))]\!]=1$ τότε

$$\frac{\partial \mathcal{E}}{\partial \mathbf{v}'_{n(w,j)}}=\left(\sigma\left({\mathbf{v}'_{n(w,j)}}^{T}\mathbf{h}\right)-1\right)\mathbf{h}$$

Αντίστοιχα άν $[\![n(w,j+1)=ch(n(w,j))]\!]=-1$ τότε

$$\frac{\partial \mathcal{E}}{\partial \mathbf{v}'_{n(w,j)}}=\left(1-\sigma\left(-{\mathbf{v}'_{n(w,j)}}^{T}\mathbf{h}\right)\right)\mathbf{h} \xrightarrow{\sigma(-x)=1-\sigma(x)} \frac{\partial \mathcal{E}}{\partial \mathbf{v}'_{n(w,j)}}=\sigma\left({\mathbf{v}'_{n(w,j)}}^{T}\mathbf{h}\right)\mathbf{h}$$

Ορίζοντας μεταβλητές t_j για κάθε εσωτερικό κόμβο ως

$$t_j=1\ \alpha\nu\ [\![n(w,j+1)=ch(n(w,j))]\!]=1$$

$$t_j=0\ \alpha\nu\ [\![n(w,j+1)=ch(n(w,j))]\!]=-1$$

παίρνουμε

$$\frac{\partial \mathcal{E}}{\partial \mathbf{v}'_{n(w,j)}}=\left(\sigma\left({\mathbf{v}'_{n(w,j)}}^{T}\mathbf{h}\right)-t_j\right)\mathbf{h}$$

και ο κανόνας ενημέρωσης γίνεται

$$\mathbf{v}'_{n(w,j)}(n+1)=\mathbf{v}'_{n(w,j)}(n)-\eta\left(\sigma\left({\mathbf{v}'_{n(w,j)}}^{T}\mathbf{h}\right)-t_j\right)\mathbf{h}$$

Για την ενημέρωση των διανυσμάτων εισόδου παίρνουμε

$$\frac{\partial \mathcal{E}}{\partial \mathbf{v}_{w_I}}=\frac{\partial \mathcal{E}}{\partial \mathbf{h}}\cdot\frac{\partial \mathbf{h}}{\partial \mathbf{v}_{w_I}}=\frac{\partial \mathbf{h}}{\partial \mathbf{v}_{w_I}}\sum_{j=1}^{L(w)-1}\frac{\partial \mathcal{E}}{\partial {\mathbf{v}'_{n(w,j)}}^{T}\mathbf{h}}\cdot\frac{\partial {\mathbf{v}'_{n(w,j)}}^{T}\mathbf{h}}{\partial \mathbf{h}}=$$

$$=\frac{\partial \mathbf{h}}{\partial \mathbf{v}_{w_I}}\sum_{j=1}^{L(w)-1}\left(\sigma\left({\mathbf{v}'_{n(w,j)}}^{T}\mathbf{h}\right)-t_j\right)\mathbf{v}'_{n(w,j)}$$

Στην περίπτωση της μίας context λέξης ισχύει $\mathbf{h}=\mathbf{v}_{w_I}$ δηλαδή $\frac{\partial \mathbf{h}}{\partial \mathbf{v}_{w_I}}=1$, συνεπώς

$$\frac{\partial \mathcal{E}}{\partial \mathbf{v}_{w_I}}=\sum_{j=1}^{L(w)-1}\left(\sigma\left({\mathbf{v}'_{n(w,j)}}^{T}\mathbf{h}\right)-t_j\right)\mathbf{v}'_{n(w,j)}$$

και τα διανύσματα εισόδου ενημερώνονται με τον κανόνα

$$\mathbf{v}_{w_I}(n+1) = \mathbf{v}_{w_I}(n) - \eta \sum_{j=1}^{L(w)-1} \left(\sigma\left(\mathbf{v}'_{n(w,j)}{}^{T}\mathbf{h}\right) - t_j\right)\mathbf{v}'_{n(w,j)}$$

Με την ίδια μεθοδολογία προκύπτουν και οι εξισώσεις ενημέρωσης των word vectors στη γενικότερη περίπτωση CBOW αλλά και στην SG εκδοχή του αλγορίθμου word2vec. Αυτό που πρέπει να τονιστεί ωστόσο είναι η πολύ σημαντική μείωση στην πολυπλοκότητα του μοντέλου με την χρησιμοποίηση της hierarchical softmax πιθανότητας. Στην απλή περίπτωση softmax είδαμε ότι το διάνυσμα της λέξης εισόδου εισέρχεται στο δίκτυο και στη συνέχεια υπολογίζονται για όλες τις λέξεις του λεξιλογίου τα τοπικά πεδία και μέσω αυτών αντίστοιχες έξοδοι για κάθε λέξη. Στη συνέχεια ενσωματώνονται οι παράμετροι t_j για να προσδιοριστούν τα λάθη πάλι για κάθε λέξη και ενημερώνονται όλα τα διανύσματα εξόδου με βάση τα λάθη αυτά. Πλέον διανύσματα εξόδου αποδίδονται στους εσωτερικούς κόμβους της δενδρικής διάταξης και όπως φάνηκε στα παραπάνω χρησιμοποιούνται οι παράμετροι t_j σε κάθε κόμβο του μονοπατιού που οδηγεί στην πραγματική λέξη, για την ενημέρωση των διανυσμάτων μόνο αυτών των κόμβων. Συνολικά σε κάθε στιγμιότυπο εκπαίδευσης και για κάθε λέξη του δείγματος εκπαίδευσης η πολυπλοκότητα γίνεται $O(\log V)$ και όχι $O(V)$ όπως στην απλή softmax πιθανότητα. Η μείωση αυτή στην πολυπλοκότητα αυξάνει αισθητά την ταχύτητα του αλγορίθμου και επιτρέπει την εκπαίδευση σε μεγάλα σύνολα δεδομένων.

Στην επόμενη ενότητα θα αναφερθεί μία ακόμη εναλλακτική που καλείται negative sampling. Βασίζεται στην ενημέρωση μέρους αντί του συνόλου των διανυσμάτων εξόδου.

4.2.3.2 Negative Sampling

Οι Mikolov et al. στο [13] προτείνουν σαν εναλλακτική της χρήσης hierarchical softmax, την εκπαίδευση του γλωσσικού μοντέλου με συνάρτηση κόστους την

$$\mathcal{E} = -\log\sigma\left(\mathbf{v}'_{w_O}{}^{T}\mathbf{h}\right) - \sum_{w_i \in W_{neg}} \log\sigma(-\mathbf{v}'_{w_i}{}^{T}\mathbf{h})$$

όπου $\mathbf{v}'_{w_O}$ είναι το διάνυσμα εξόδου της λέξης που παρατηρείται στην πράξη στα δεδομένα και $\mathbf{v}'_{w_i}$ τα διανύσματα εξόδου των λέξεων w_i οι οποίες δειγματοληπτούνται από το λεξικό. Δηλαδή από όλες τις λέξεις του λεξικού, στη συνάρτηση κόστους συνυπολογίζονται μόνο η λέξη w_O και κάποιες λέξεις w_i. Για αυτό το λόγο η λέξη w_O καλείται θετικό δείγμα (positive sample) γιατί παρατηρείται στην πράξη και δειγματοληπτείται πάντα και οι λέξεις w_i αρνητικά δείγματα (negative samples). Η μέθοδος καλείται negative sampling.

Η παραπάνω συνάρτηση κόστους δίνει τελικά την παρακάτω ενημέρωση δανυσμάτων

$$\mathbf{v}'_{w_j}(n+1) = \mathbf{v}'_{w_j}(n) - \eta\left(\sigma\left(\mathbf{v}'_{w_j}{}^{T}\mathbf{h}\right) - t_j\right)\mathbf{h}$$

για $w_j \in \{w_O\} \cup W_{neg}$.

Συνεπώς στην περίπτωση negative sampling μόνο η πραγματική λέξη w_O και μερικές λέξεις $w_i \in W_{neg}$ συμμετέχουν στην εκπαίδευση και υφίστανται ενημέρωση διανυσμάτων εξόδου. Οι λέξεις w_i προκύπτουν με δειγματοληψία του λεξικού σύμφωνα με κάποια κατανομή $P_n(w)$. Εμπειρικά στο [13] προτείνεται η χρήση unigram κατανομής υψωμένης στη δύναμη ¾.

Μία πιο λεπτόμερης ανάλυση της negative sampling εκδοχής του μοντέλου word2vec δίνεται από τους Goldberg και Levy στο [5].

4.2.4 Το Μοντέλο GloVe

Σύμφωνα με τους Pennington, Socher και Manning [20] ένα μειονέκτημα του αλγορίθμου word2vec είναι η μη ενσωμάτωση πληροφορίας, στις διανυσματικές αναπαραστάσεις, για τα συνολικά στατιστικά στοιχεία του σώματος κειμένου. Υποδεικνύουν δύο γενικούς τρόπους εξαγωγής διανυσματικών αναπαραστάσεων, όπου ο πρώτος εκμεταλλεύεται τα συνολικά στατιστικά στοιχεία του σώματος κειμένου και εμπλέκει συνήθως μεθόδους παραγοντοποίησης του co-occurrence πίνακα, ενώ ο δεύτερος εκμεταλλεύται τοπική πληροφορία με τη χρήση παραθύρων κειμένου. Ο ίδιος διαχωρισμός αναφέρθηκε και στην παρούσα εργασία. Στην ενότητα 4.2.1 είδαμε την απλή μέθοδο εξαγωγής διανυσμάτων από τον co-occurrence πίνακα που ανήκει στην πρώτη κατηγορία και στις ενότητες 4.2.2 και 4.2.3 το window-based μοντέλο word2vec που ανήκει στην δεύτερη. Στην ίδια δημοσίευση οι συγγραφείς προτείνουν ένα μοντέλο που χρησιμοποιεί πληροφορία από στατιστικές μετρικές στο σύνολο των δεδομένων και εκπαιδεύεται με μεθόδους gradient descent. Μάλιστα παρουσιάζεται και μία θεωρητική ανάλυση που θέλει το μοντέλο αυτό να έχει στην ουσία στενή σχέση με το μοντέλο word2vec των Mikolov et al. ([12]) . Το μοντέλο αυτό χαρακτηρίζεται GloVe από την συνένωση των λέξεων Global Vectors και παράγει διανυσματικές αναπαραστάσεις ενσωματώνοντας συνολική (global) πληροφορία.

Η θεωρητική ανάλυση του αλγορίθμου GloVe είναι απλούστερη της αντίστοιχης του μοντέλου word2vec. Το μοντέλο GloVe χρησιμοποιεί τον πίνακα co-occurrence του συνόλου δεδομένων αλλά δεν εφαρμόζει κάποια μέθοδο μείωσης της διαστατικότητας. Αντίθετα χρησιμοποιεί gradient descent με σκοπό την ελαχιστοποίηση μίας συνάρτησης κόστους όπως τα νευρωνικά γλωσσικά μοντέλα. Αποτελεί δηλαδή ένα συνδυασμό των δύο διαφορετικών τρόπων εξαγωγής διανυσματικών αναπαραστάσεων.

Αρχικά από το σύνολο των δεδομένων κατασκεύαζεται ο co-occurrence πίνακας **X**. Κάθε στοιχείο του πίνακα X_{ij} αναπαριστά ένα μέτρο του πόσο συχνά η λέξη w_j εμφανίζεται στη γειτονιά της λέξης w_i. Σε αντίθεση με τον απλό co-occurrence πίνακα της ενότητας 4.2.1, το context είναι μεγαλύτερο των δύο λέξεων και αποτελεί υπερπαράμετρο του μοντέλου. Επίσης επιλογή του χρήστη είναι η στάθμιση της συνεισφοράς των πιο απομακρυσμένων λέξεων στο αντίστοιχο στοιχείο X_{ij}. Ο υπολογισμός του πίνακα **X** είναι μία απαιτητική υπολογιστικά εργασία για σύνολα δεδομένων μεγέθους ανάλογου με αυτά που χρησιμοποιεί ο αλγόριθμος word2vec, ωστόσο ο υπολογισμός αυτός γίνεται μόνο μία φορά κατά τη διάρκεια του αλγορίθμου GloVe. Αφού υπολογιστεί ο πίνακας **X**, το μοντέλο GloVe κάνει

χρήση της μεθόδου Stochastic Gradient Descent για την εκπαίδευση διανυσματικών αναπαραστάσεων, με συνάρτηση κόστους την

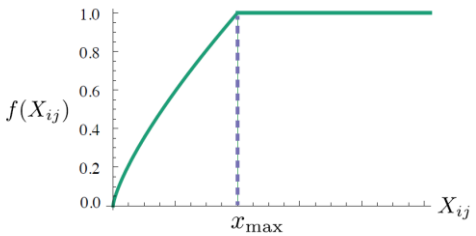
$$J = \sum_{i=1}^{V}\sum_{j=1}^{V} f(X_{ij})\left(\mathbf{v}_i^T\tilde{\mathbf{v}}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

όπου $\mathbf{v}_i$ είναι η διανυσματική αναπαράσταση της λέξης w_i, $\tilde{\mathbf{v}}_j$ η διανυσματική αναπαράσταση της λέξης w_j, b_i και $\tilde{b}_j$ βαθμωτές πολώσεις και X_{ij} το ij στοιχείο του πίνακα $\mathbf{X}$. Σημειώνεται ότι όπως το μοντέλο word2vec, έτσι και το μοντέλο GloVe χρησιμοποιεί δύο διανυσματικές αναπαραστάσεις για κάθε λέξη του λεξικού, με τη μία να αναφέρεται στην περίπτωση που η λέξη είναι στο context και την άλλη στην περίπτωση που η λέξη είναι κεντρική. Η συνάρτηση f ορίζεται στα θετικά του άξονα x, είναι αύξουσα και ικανοποιεί δύο ανάγκες του μοντέλου. Εξαλείφει την απροσδιοριστία $\log 0$ στην περίπτωση που $X_{ij} = 0$, παίρνοντας τιμή $f(0) = 0$ και κατωφλιώνει τις μεγάλες τιμές X_{ij} που εμφανίζονται για πολύ συχνές λέξεις. Η συνάρτηση που προτείνεται στο [20] είναι η

$$f(x) = \begin{cases} \left(\dfrac{x}{x_{max}}\right)^a, x < x_{max} \\ 1, x \geq x_{max} \end{cases}$$

και για καλύτερα αποτελέσματα, εμπειρικά προτείνεται η τιμή $a = 3/4$, η ίδια τιμή δηλαδή που δίνει και τα καλύτερα αποτελέσματα στην negative sampling εκδοχή του αλγορίθμου word2vec, όπου εκεί, όπως είδαμε, χρησιμοποιείται στην κατανομή $P_n(w)$ για τη δειγματοληψία των διανυσμάτων εξόδου. Στο σχήμα 4.6 φαίνεται η παραπάνω συνάρτηση για $\alpha = 3/4$.

Για την εφαρμογή gradient descent υπολογίζονται οι παράγωγοι της τετραγωνικής συνάρτησης κόστους ως προς τις παραμέτρους, που είναι οι διανυσματικές αναπαραστάσεις και οι πολώσεις. Σημειώνεται ότι στην πράξη εφαρμόζεται συνήθως μία παραλλαγή της μεθόδου stochastic gradient descent που καλείται *AdaGrad* (Adaptive Gradient Descent) και σύμφωνα με την οποία, οι παράμετροι x_i ενός προβλήματος βελτιστοποίησης ενημερώνονται βάσει του κανόνα



Σχήμα 4.6

$$x_{t+1,i} = x_{t,i} - \eta\frac{g_{t,i}}{\sqrt{\sum_{\tau=1}^{t-1} g_{\tau,i}^2}}$$

όπου $x_{t+1,i}$ η παράμετρος x_i τη στιγμή $t+1$, $x_{t,i}$ η ίδια παράμετρος τη στιγμή t, η ο ρυθμός μάθησης, $g_{t,i}$ η παράγωγος της συνάρτησης κόστους ως προς την παράμετρο x_i τη στιγμή t και $g_{\tau,i}$ η ίδια παράγωγος τη στιγμή τ.

Δηλαδή η παράμετρος ενημερώνεται με βάση την παράγωγο του κόστους διά του τετραγώνου του αθροίσματος των παραγώγων σε όλα τα προηγούμενα χρονικά βήματα.

Το μοντέλο GloVe εκπαιδεύεται αποδοτικά σε μεγάλα σύνολα δεδομένων εξαιτίας της μεθόδου stochastic gradient descent αλλά και γιατί ο πίνακας **X** είναι αραιός και η συνάρτηση κόστους εμπεριέχει ουσιατικά μόνο τα μη μηδενικά στοιχεία του. Έτσι μοιράζεται το ίδιο πλεονέκτημα με το μοντέλο word2vec, την δυνατότητα δηλαδή εκπαίδευσης σε μεγάλους όγκους κειμένου που επιτρέπει την εκμάθηση ποιοτικών word vectors. Στην δημοσίευση του μοντέλου ([20]) περιλαμβάνεται και μία αναλυτική παρουσίαση της υπολογιστικής πολυπλοκότητας του αλγορίθμου.

Η συνάρτηση κόστους του μοντέλου GloVe προκύπτει μαθηματικά από την απαίτηση μετάφρασης των νοηματικών σχέσεων των λέξεων σε γραμμικές ιδιότητες του χώρου αναπαραστάσεων. Όπως θα φανεί στην επόμενη ενότητα το μοντέλο word2vec αλλά και το μοντέλο GloVe, αναπαριστούν πολύπλοκες λεκτικές σχέσεις με τη μορφή διαφορών στο χώρο των word vectors. Ο λόγος που ένα νευρωνικό μοντέλο είναι σε θέση να μαθαίνει τέτοιες χρήσιμες ιδιότητες δεν είναι μαθηματικά προφανής. Αντίθετα στην περίπτωση του μοντέλου GloVe δίνεται η ακριβής μαθηματική αιτιολογία που το μοντέλο έχει αυτή τη δυνατότητα και μάλιστα ο αλγόριθμος θεμελιώνεται πάνω σε αυτή τη δυνατότητα.

Ας υποθέσουμε τις λέξεις ice και steam. Θα μπορούσε κάποιος να ισχυριστεί ότι οι δύο λέξεις είναι κοντά νοηματικά καθώς αποτελούν θερμοδυναμικές καταστάσεις του νερού. Οι δύο λέξεις όμως συμβολίζουν δύο αντίθετες καταστάσεις του νερού. Ιδανικά λοιπόν θέλουμε να κατασκευάσουμε διανυσματικές αναπαραστάσεις για τις δύο λέξεις, οι οποίες ναι μεν να είναι κοντά αλλά ταυτόχρονα να ενσωματώνουν τη σχέση που έχουν οι δύο λέξεις, που είναι μία σχέση φυσικής κατάστασης ή υγρού-αερίου. Για να γίνει αυτό επιτυχώς στο [20] προτείνεται η χρήση των co-occurrence ratios των λέξεων στο σώμα κειμένου. Αυτό γίνεται προφανές με τη βοήθεια του πίνακα που ακολουθεί.

	$x = solid$	$x = gas$	$x = water$	$x = random$
$P(x\|ice)$	*μεγάλη*	*μικρή*	*μεγάλη*	*μικρή*
$P(x\|steam)$	*μικρή*	*μεγάλη*	*μεγάλη*	*μικρή*
$\frac{P(x\|ice)}{P(x\|steam)}$	*μεγάλη*	*μικρή*	$\cong 1$	$\cong 1$

Πίνακας 4.2

Η πιθανότητα οι λέξεις solid και ice να συνυπάρχουν σε ένα context αναμένουμε να είναι μεγάλη. Όμοια η πιθανότητα να συνυπάρχουν οι λέξεις gas και steam, οι ice και water και οι steam και water. Οι υπόλοιπες πιθανότητες αναμένουμε να είναι μικρές. Οι λόγοι των παραπάνω πιθανοτήτων για οποιαδήπτε λέξη του λεξικού θα παίρνουν τιμή περίπου ίση με 1 και μόνο για τις λέξεις solid και gas θα παίρνουν μεγάλη ή μικρή τιμή. Δηλαδή η πληροφορία για τη σχέση των λέξεων ice και steam βρίσκεται σε εκείνες τις λέξεις του λεξικού για τις οποίες ο παραπάνω λόγος πιθανοτήτων παίρνει μεγάλη ή μικρή τιμή. Ένα λογικό επιχείρημα λοιπόν είναι να ενσωματωθεί η πληροφορία των λόγων των πιθανοτήτων στη διαφορά των διανυσμάτων. Το γενικό μαθηματικό μοντέλο που αναπαριστά την

ενσωμάτωση της βαθμωτής πληροφορίας του λόγου των πιθανοτήτων σε διανυσματικό χώρο και που εμπλέκει τρία διανύσματα είναι

$$F(\mathbf{v}_i, \mathbf{v}_j, \tilde{\mathbf{v}}_k) = \frac{P_{ik}}{P_{jk}}$$

Η λέξη w_k θεωρείται ότι βρίσκεται στο context όπως αντιμετωπίστηκε και στον πίνακα. Για την ενσωμάτωση της πληροφορίας στη διαφορά των διανυσμάτων των λέξεων w_i και w_j το επόμενο βήμα είναι η αναζήτηση μίας συνάρτησης F ώστε

$$F(\mathbf{v}_i - \mathbf{v}_j, \tilde{\mathbf{v}}_k) = \frac{P_{ik}}{P_{jk}}$$

Για λόγους διευκόλυνσης μπορούμε να αναζητήσουμε μία βαθμωτή συνάρτηση F αρκεί και το όρισμα της να είναι βαθμωτό. Συνεπώς περιοριζόμαστε στην απλή απεικόνιση

$$F\left((\mathbf{v}_i - \mathbf{v}_j)^T \tilde{\mathbf{v}}_k\right) = \frac{P_{ik}}{P_{jk}} = \frac{X_{ik}/X_i}{X_{jk}/X_j}$$

όπου η πιθανότητα P_{ik} είναι ο αριθμός των συνυπάρξεων X_{ik} προς τον αριθμό εμφάνισης X_i της λέξης w_i στο σώμα κειμένου. Τέλος για να είναι καλά ορισμένη η συνάρτηση F στην περίπτωση εναλλαγής των context λέξεων με κεντρικές προτείνεται η συνάρτηση σαν ομομορφισμός

$$F\left((\mathbf{v}_i - \mathbf{v}_j)^T \tilde{\mathbf{v}}_k\right) = \frac{F(\mathbf{v}_i^T \tilde{\mathbf{v}}_k)}{F(\mathbf{v}_j^T \tilde{\mathbf{v}}_k)} = \frac{P_{ik}}{P_{jk}} = \frac{X_{ik}/X_i}{X_{jk}/X_j}$$

Βάσει των παραπάνω περιορισμών που τέθηκαν, οι εξισώσεις καταλήγουν στην μοναδική λύση $F(x) = \exp(x)$. Οπότε

$$\exp(\mathbf{v}_i^T \tilde{\mathbf{v}}_k) = \frac{X_{ik}}{X_i} \quad \Rightarrow \quad \mathbf{v}_i^T \tilde{\mathbf{v}}_k = \log(X_{ik}) - \log(X_i)$$

Ο παράγοντας $\log(X_i)$ μπορεί να ενσωματωθεί σε μία πόλωση b_i καθώς είναι ανεξάρτητος του k. Αφού προστεθεί η πόλωση b_i για λόγους συμμετρίας πρέπει να προστεθεί και πόλωση $\tilde{b}_k$ οπότε καταλήγουμε τελικά στη σχέση

$$\mathbf{v}_i^T \tilde{\mathbf{v}}_k + b_i + \tilde{b}_k - \log(X_{ik}) = 0$$

Ελαχιστοποιώντας λοιπόν το κριτήριο

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})\left(\mathbf{v}_i^T \tilde{\mathbf{v}}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

προσεγγίζεται ο μηδενισμός των σχέσεων $\mathbf{v}_i^T \tilde{\mathbf{v}}_j + b_i + \tilde{b}_j - \log X_{ij}$ που είναι το ζητούμενο για να ισχύουν οι προηγούμενες σχέσεις και συνεπώς και η ενσωμάτωση πληροφορίας από τους λόγους των πιθανοτήτων στις διαφορές των διανυσματικών αναπαραστάσεων.

Τονίζεται ότι τα παραπάνω δεν αποτελούν μαθηματική απόδειξη αλλά επεξήγηση των μαθηματικών πίσω από την ικανότητα τέτοιων μοντέλων να μαθαίνουν αναπαραστάσεις που αντιστοιχούν πολυεπίπεδες νοηματικές σχέσεις σε διαφορές στο χώρο διανυσμάτων. Δικαιολογούν την ικανότητα του μοντέλου GloVe να εκπαιδεύει τέτοιες αναπαραστάσεις αλλά όχι την αντίστοιχη του μοντέλου word2vec. Ωστόσο αν θεωρήσουμε ότι τα δύο μοντέλα έχουν στενή μαθηματική σχέση, όπως δηλώνεται στο [20] τότε ίσως εξηγείται και η αντίστοιχη ικανότητα του μοντέλου word2vec.

Όπως είδαμε λοιπόν το μοντέλο GloVe αποτελεί ένα συνδυασμό μεθόδων παραγοντοποίησης πίνακα και window-based μεθόδων. Επιπλέον οι Goldberg και Levy στο [11] επεξηγούν πως τα νευρωνικά γλωσσικά μοντέλα στην πράξη εκτελούν παραγοντοποίηση πίνακα που δεν είναι άμεσα προφανής. Συνεπώς ο διαχωρισμός των μεθόδων εξαγωγής word vectors στις δύο παραπάνω κατηγορίες δεν είναι απόλυτα αυστηρός. Ένας εναλλακτικός διαχωρισμός είναι σε count-based και prediction μοντέλα, όπου τα πρώτα χρησιμοποιούν τα co-occurrence counts σε όλο το σώμα κειμένου όπως το μοντέλο GloVe και τα δεύτερα προσπαθούν να προβλέψουν λέξεις δεδομένου ενός context παραθύρου όπως το word2vec. Παραπέμπουμε στο [1] για μία συστηματική σύγκριση των δύο κατηγοριών.

4.2.5 Ιδιότητες των Μοντέλων word2vec και GloVe

Τα μοντέλα word2vec και GloVe παράγουν διανυσματικές αναπαραστάσεις λέξεων σε ένα χώρο μικρής διάστασης d. Οι τιμές του d συνήθως κυμαίνονται από 25 έως 500. Όπως έγινε φανερό από τη θεωρία και τα δύο μοντέλα παράγουν δύο αναπαραστάσεις για κάθε λέξη του λεξικού, μία για τη λέξη στο context και μία για τη λέξη όταν ορίζει το context. Οι τελικές διανυσματικές αναπαραστάσεις συνήθως προκύπτουν από την άθροιση των δύο διανυσμάτων, δηλαδή

$$\mathbf{x}_i = \mathbf{v}_i + \mathbf{v}_i'$$

με τους συμβολισμούς του μοντέλου word2vec ή

$$\mathbf{x}_i = \mathbf{v}_i + \tilde{\mathbf{v}}_i$$

με τους συμβολισμούς του μοντέλου GloVe.

Υπερπαράμετροι των μοντέλων αποτελούν το context c, η συνάρτηση f στο GloVe, η επιλογή CBOW ή SG στο word2vec, negative sampling ή hierarchical softmax επίσης στο word2vec, η στάθμιση απομακρυσμένων context λέξεων στο GloVe, η δύναμη της unigram κατανομής στο word2vec για την περίπτωση negative sampling αλλά και λοιπές παράμετροι της εκπαίδευσης όπως οι εποχές και ο ρυθμός μάθησης.

Στην παρούσα ενότητα θα παρουσιαστούν οι ενδιαφέρουσες ιδιότητες των word vectors μέσω ενδεικτικών γραφημάτων και πινάκων. Οι εικόνες που ακολουθούν προέρχονται από τις δημοσιεύσεις του μοντέλου word2vec ([12] και [13]) και από τον ιστότοπο του μοντέλου GloVe[23]. Στην περίπτωση γραφημάτων προκύπτουν από τη μείωση της διάστασης των word vectors σε δύο διαστάσεις με μεθόδους μείωσης διαστατικότητας.

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Πίνακας 4.3

Στον πίνακα 4.3 παρουσιάζονται μερικές ενδιαφέρουσες σχέσεις που ανακαλύπτει μία υλοποίηση του μοντέλου word2vec. Σε κάθε γραμμή τα διανύσματα των λέξεων στην πρώτη στήλη αφαιρούνται, το αποτέλεσμα προστίθεται στην πρώτη λέξη της δεύτερης, τρίτης και τέταρτης στήλης και προκύπτει ένα νέο σημείο στο χώρο. Μετά το σύμβολο : σημειώνεται η λέξη της οποίας το διάνυσμα είναι πιο κοντά στο νέο αυτό σημείο. Το μοντέλο δηλαδή αποτυπώνει σε διαφορές διανυσμάτων τις διάφορες νοηματικές σχέσεις της πρώτης στήλης και προκύπτουν λέξεις με την ίδια νοηματική σχέση ως προς μία άλλη λέξη.

Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna	Hanoi	airline Lufthansa	Moscow	Juliette Binoche
Check crown	Ho Chi Minh City	carrier Lufthansa	Volga River	Vanessa Paradis
Polish zolty	Viet Nam	flag carrier Lufthansa	upriver	Charlotte Gainsbourg
CTK	Vietnamese	Lufthansa	Russia	Cecile De

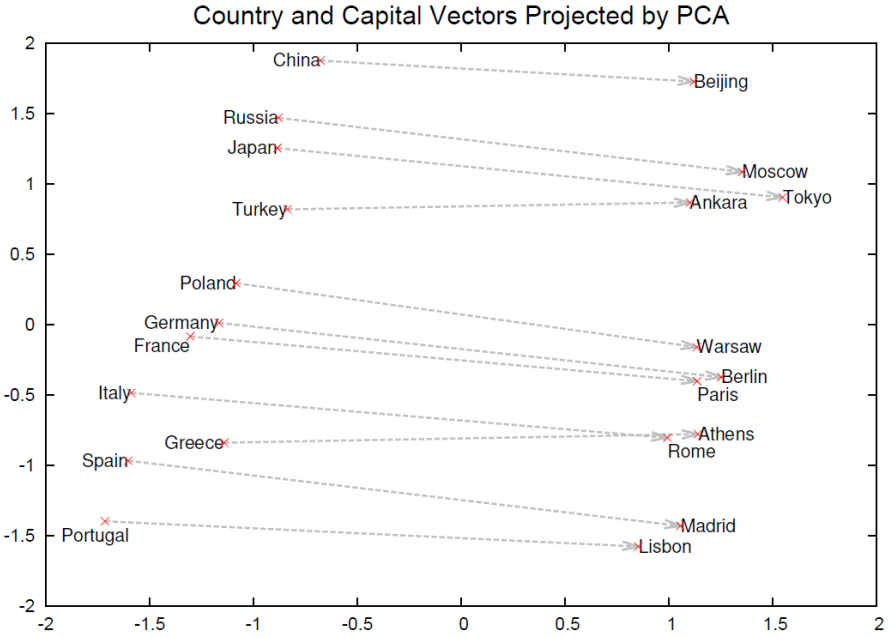
Πίνακας 4.4

Στον πίνακα 4.4 πάλι μία υλοποίηση του μοντέλου word2vec συνδέει νοηματικά λέξεις με τα αθροίσματα των διανυσμάτων διαφορετικών λέξεων. Σε κάθε στήλη τα διανύσματα των δύο λέξεων της πρώτης γραμμής προστίθενται και προκύπτει ένα νέο σημείο στο χώρο αναπαραστάσεων. Στη συνέχεια παρουσιάζονται οι τέσσερις λέξεις ή φράσεις των οποίων οι διανυσματικές αναπαραστάσεις είναι οι κοντινότερες στο νέο σημείο. Για τη γενίκευση των word vectors σε "phrase" vectors γίνεται συζήτηση στο [13].

Στο σχήμα 4.7 απεικονίζονται γραφικά οι σχέσεις χωρών με τις πρωτεύουσές τους, που παράγει η εκπαίδευση ενός μοντέλου word2vec. Παρατηρούμε ότι το διάνυσμα

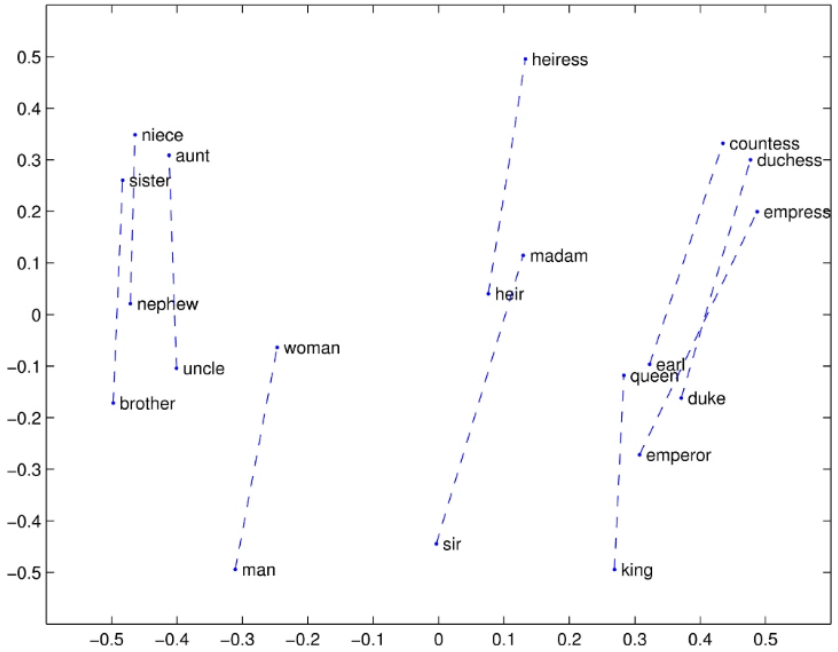
[23] http://nlp.stanford.edu/projects/glove/

$\mathbf{x}_{\pi\rho\omega\tau\epsilon\acute{\upsilon}ουσα} - \mathbf{x}_{\chi\acute{\omega}\rho\alpha}$ έχει σχεδόν σταθερό μέτρο και κατεύθυνση στη διδιάσταση προσέγγιση του χώρου αναπαραστάσεων. Η σχέση δηλαδή χώρα-πρωτεύουσα έχει αποτυπωθεί σε ένα διάνυσμα.

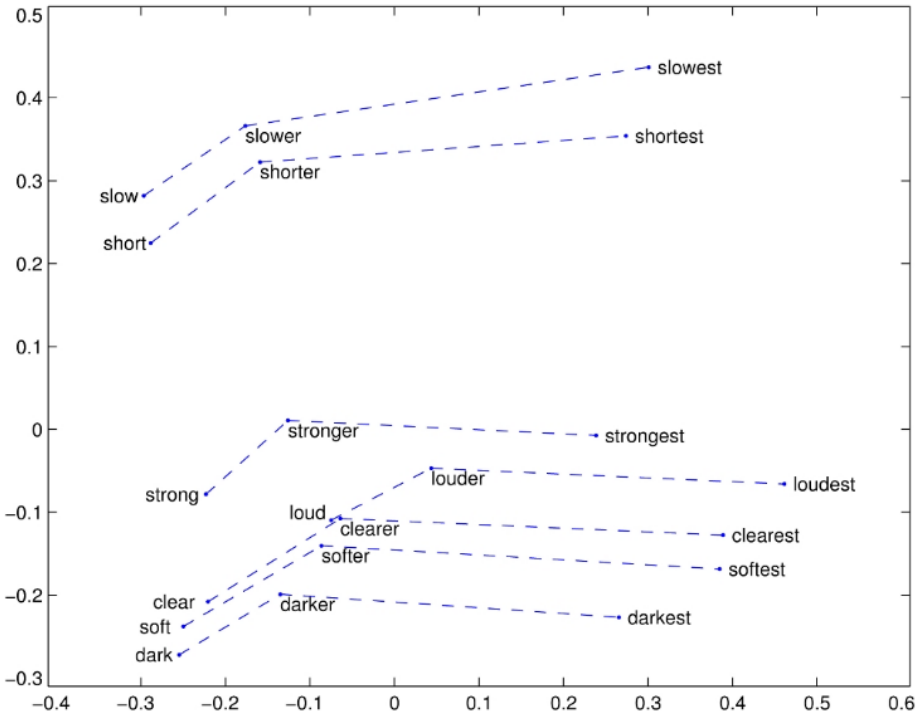


Σχήμα 4.7

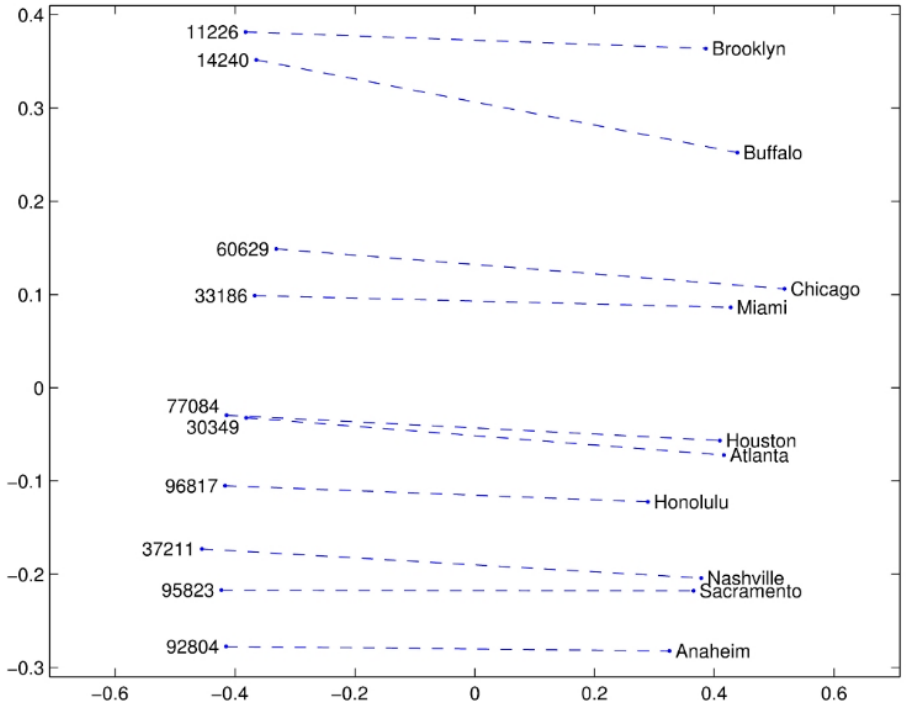
Στα σχήματα 4.8, 4.9 και 4.10 δίνονται ανάλογα γραφήματα από το μοντέλο GloVe για τη σχέση φύλου, τη σχέση επιθέτου-συγκριτικού-υπερθετικού βαθμού και τη σχέση πόλης-ταχυδρομικού κώδικα αντίστοιχα.



Σχήμα 4.8



Σχήμα 4.9



Σχήμα 4.10

4.3 Διανυσματικές Αναπαραστάσεις Κειμένου

Στην ενότητα 4.2 είδαμε μαθηματικά μοντέλα που παράγουν διανυσματικές αναπαραστάσεις λέξεων. Στην παρούσα ενότητα θα μελετήσουμε τρόπους για την συγχώνευση των word vectors με σκοπό την αναπαράσταση συνόλων λέξεων, όπως φράσεις, προτάσεις ή και ολόκληρα κείμενα. Υπενθυμίζεται ότι το πρόβλημα που αντιμετωπίζεται είναι η ταξινόμηση των tweets σε κλάσεις, οπότε μας ενδιαφέρει η αποτελεσματική αναπαράσταση του κάθε tweet σε διάνυσμα. Η εξαγωγή διανυσματικών αναπαραστάσεων για σύνολα λέξεων μπορεί να γίνει με διάφορους τρόπους, γενικότερα όμως αποτελεί ένα πεδίο έρευνας, που είναι μεν ενεργό αλλά δεν έχει επιδείξει σημαντικά αποτελέσματα.

Το πρόβλημα είναι αρκετά σύνθετο και αναφέρεται ουσιαστικά στην εύρεση μίας συνάρτησης που θα απεικονίζει το έγγραφο σε ένα διάνυσμα, δεδομένων των διανυσμάτων των λέξεων που περιέχονται στο έγγραφο. Ουσιαστικά αναζητείται η σχέση με την οποία συνδυάζονται οι λέξεις για να αποδοθεί νόημα στο σύνολό τους. Μία λογική πρόταση θα ήταν ο συνδυασμός των λέξεων με βάση την συντακτική τους δομή. Εφαρμόζοντας έναν parser πάνω σε μία πρόταση, παίρνουμε τα Part-Of-Speech tags (POS tags) των λέξεων και μία δενδρική δομή (parse tree) που απεικονίζει τη σύνταξη της πρότασης. Ξεκινώντας από τα φύλλα του δέντρου που αντιπροσωπεύουν λέξεις και δεδομένων των word vectors, συνδυάζουμε τις διανυσματικές αναπαραστάσεις με βάση το δέντρο για να καταλήξουμε στη ρίζα. Η τελική διανυσματική αναπαράσταση στη ρίζα του δέντρου αντιπροσωπεύει την πρόταση.

Στα επόμενα θα δούμε μερικούς τρόπους συνδυασμού των word vectors για την εξαγωγή document vectors (ή sentence vectors ή paragraph vectors). Αρχικά θα παρουσιαστούν κάποιοι απλοί τρόποι όπως πρόσθεση και συνένωση (concatenation) των word vectors, στη συνέχεια θα δοθεί για λόγους πληρότητας η ειδική περίπτωση απεικόνισης των δεδομένων κειμένου που απαιτεί ένα συνελικτικό δίκτυο και τέλος το μοντέλο doc2vec των Le και Mikolov ([10]), μία επέκταση του μοντέλου word2vec.

4.3.1 Απλές Μέθοδοι Συνδυασμού των Word Vectors

Στο [15] οι Mitchell και Lapata παρουσιάζουν το θεωρητικό υπόβαθρο πίσω από τη συγχώνευση του νοήματος διαφόρων λέξεων για την παραγωγή του νοήματος του συνόλου τους και συγκρίνουν διάφορες απλές μαθηματικές τεχνικές συγχώνευσης διανυσματικών αναπαραστάσεων, ανάλογα με την επίδοσή τους σε σύνολα δεδομένων που αφορούν τη συνάφεια φράσεων. Κάποιες από αυτές θα παρουσιαστούν και εδώ και θα χρησιμοποιηθούν στην υλοποίησή μας.

Έστω ένα έγγραφο d που αποτελείται από τις λέξεις w_i , $i = 1,2,\dots,n$. Θεωρούνται γνωστά τα m-διάστατα word vectors των λέξεων $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{ij}, \dots, v_{im})$, $i = 1,2,\dots,n$ και αναζητείται τη διανυσματική αναπαράσταση του εγγράφου $\mathbf{v}_d$.

Πρόσθεση - Addition

Δεδομένου ότι ο χώρος αναπαραστάσεων έχει γραμμικές ιδιότητες που αντικατοπτρίζουν τις σχέσεις των λέξεων, είναι λογικό να θεωρηθεί η πρόσθεση των word vectors ως ένας καλός τρόπος αναπαράστασης του εγγράφου. Δηλαδή

$$\mathbf{v}_d = \sum_{i=1}^{n} \mathbf{v}_i \quad , \quad v_{dj} = \sum_{i=1}^{n} v_{ij}$$

Επίσης μπορούμε να διαιρέσουμε με τον αριθμό των word vectors στο έγγραφο και να οριστεί το διάνυσμα του εγγράφου ως το κεντρικό σημείο των word vectors στο χώρο αναπαραστάσεων

$$\mathbf{v}_d = \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_i \quad , \quad v_{dj} = \frac{1}{n}\sum_{i=1}^{n} v_{ij}$$

Σταθμισμένη Πρόσθεση - Weighted Addition

Για τη μείωση της συνεισφοράς των διανυσμάτων των πολύ συχνών λέξεων, το διάνυσμα του εγγράφου μπορεί να οριστεί ως ο σταθμισμένος μέσος των word vectors όπου τα βάρη είναι ίσα με τους συντελεστές tf-idf (term frequency - inverse document frequency - ενότητα 4.1). Αναλυτικά

$$\mathbf{v}_d = \sum_{i=1}^{n} \alpha_i \mathbf{v}_i \quad , \quad v_{dj} = \sum_{i=1}^{n} a_i v_{ij}$$

ή

$$\mathbf{v}_d = \frac{1}{n}\sum_{i=1}^{n} \alpha_i \mathbf{v}_i \quad , \quad v_{dj} = \frac{1}{n}\sum_{i=1}^{n} a_i v_{ij}$$

όπου

$$\alpha_i = f(w_i, d) \log \frac{N}{k_i}$$

και $f(w_i, d)$ η συχνότητα εμφάνισης του όρου w_i στο έγγραφο, N ο συνολικός αριθμός εγγράφων και k_i ο αριθμός εγγράφων στα οποία εμφανίζεται ο όρος w_i. Σταθμισμένη άθροιση των επιμέρους word vectors μπορεί να γίνει και με διαφορετικού τύπου βάρη, για παράδειγμα βάρη που ευνοούν συγκεκριμένα μέρη του λόγου. Έτσι αν θέλουμε τα ρήματα, για παράδειγμα, να συμμετέχουν κατά μεγαλύτερο ποσοστό στο διάνυσμα του εγγράφου μπορούμε να τους αναθέσουμε μεγαλύτερα βάρη από τα υπόλοιπα μέρη του λόγου.

Πολλαπλασιασμός - Multiplication

Το διάνυσμα εγγράφου προκύπτει από τον στοιχείο προς στοιχείο πολλαπλασιασμό των word vectors.

$$v_{dj} = \prod_{i=1}^{n} v_{ij}$$

ή

$$v_{dj} = \frac{1}{n} \prod_{i=1}^{n} v_{ij}$$

Σημειώνεται ότι οι παραπάνω τρόποι παράγουν διάνυσμα εγγράφου ίδιας διάστασης με τα επιμέρους word vectors. Όπως και η μέθοδος Bag-of-Words αγνοούν την σειρά των λέξεων.

Συνένωση - Concatenation

Τα word vectors τοποθετούνται στη σειρά, το ένα μετά το άλλο, ανάλογα με τη σειρά που έχουν οι λέξεις στο έγγραφο και προκύπτει ένα νέο διάνυσμα μήκους $n \cdot m$. Η τεχνική αυτή δεν είναι πρακτική για μεγάλα έγγραφα, παρά μόνο για φράσεις και περιπτώσεις λίγων προτάσεων. Διατηρεί την πληροφορία της σειράς των λέξεων αλλά παράγει διανύσματα διαφορετικού μήκους αφού σε κάθε έγγραφο το πλήθος λέξεων n είναι διαφορετικό.

$$\mathbf{v}_d = (v_{11}, v_{12}, \dots, v_{1m}, v_{21}, v_{22}, \dots, v_{2m}, \dots, v_{i1}, v_{i2}, \dots, v_{im}, \dots, v_{n1}, v_{n2}, \dots, v_{nm})$$

Για να προκύψουν διανύσματα ίδιου μεγέθους συχνά εφαρμόζεται zero-padding, δηλαδή επαύξηση των διανυσμάτων με μηδενικά στο τέλος, μέχρι το μήκος του μεγαλύτερου διανύσματος.

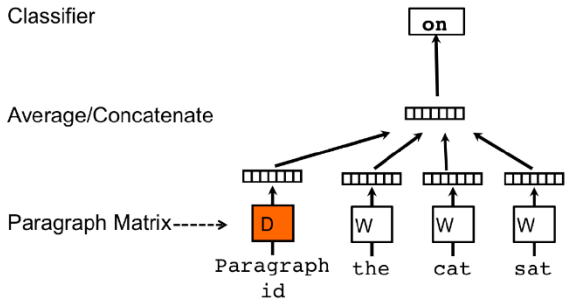
Η είσοδος του συνελικτικού δικτύου

Το συνελικτικό δίκτυο σε εφαρμογές επεξεργασίας φυσικού λόγου, όπως είδαμε στην ενότητα 3.6 δέχεται στην είσοδο ένα $n \times m$ πίνακα όπου κάθε γραμμή αντιστοιχεί στο διάνυσμα μίας λέξης. Η πρώτη γραμμή είναι το διάνυσμα της πρώτης λέξης, η δεύτερη το διάνυσμα της δεύτερης λέξης κ.ο.κ .

4.3.2 Το Μοντέλο doc2vec

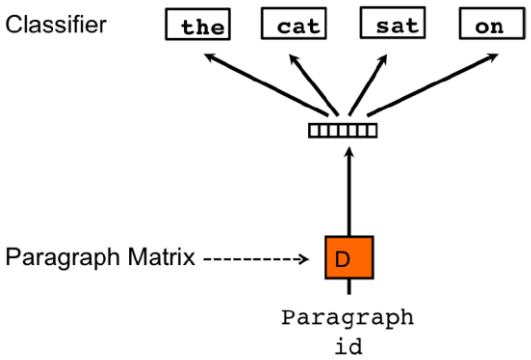
Οι Le και Mikolov στο [10] προτείνουν ένα νευρωνικό μοντέλο για την εκπαίδευση διανυσματικών αναπαραστάσεων κειμένου που αποτελεί άμεση γενίκευση του μοντέλου word2vec. Ονομάζουν το μοντέλο Paragraph Vector (PV) ωστόσο συχνά το ίδιο μοντέλο αναφέρεται με το όνομα doc2vec, ονομασία που καταδεικνύει τη στενή σχέση του με το μοντέλο word2vec αλλά και τη δυνατότητα του μοντέλου να παράγει αναπαραστάσεις για οποιασδήποτε μορφής έγγραφα, από φράσεις και προτάσεις μέχρι παραγράφους και ολόκληρα κείμενα.

Η βασική διαφορά του μοντέλου με το word2vec είναι ότι χρησιμοποιεί τις διανυσματικές αναπαραστάσεις κειμένου (document vectors) μαζί με τα word vectors για να κάνει προβλέψεις για το σημασιολογικό πλαίσιο και εκπαιδεύει και τα δύο με κριτήριο την ελαχιστοποίηση του κόστους πρόβλεψης. Ας υποθέσουμε ένα σώμα κειμένου που οργανώνεται σε M έγγραφα και περιέχει N διαφορετικές λέξεις. Το μοντέλο αρχικοποιεί τυχαία διανυσματικές αναπαραστάσεις διάστασης d για κάθε ένα από τα M έγγραφα και κάθε μία από τις N λέξεις. Έπειτα διατρέχει το σώμα κειμένου με παράθυρα, όπως και το μοντέλο word2vec και σε κάθε παράθυρο χρησιμοποιεί τα διανύσματα των λέξεων του παραθύρου αλλά και το διάνυσμα του εγγράφου στο οποίο αναφέρεται το παράθυρο για να προβλέψει την λέξη που ακολουθεί μετά το παράθυρο. Έτσι τα διανύσματα των λέξεων μοιράζονται σε όλο το κείμενο και τα διανύσματα των εγγράφων μοιράζονται μεταξύ παραθύρων του ίδιου εγγράφου. Η προσέγγιση αυτή καλείται Distributed Memory (PV-DM) και δίνεται σχηματικά στο σχήμα 4.11.



Σχήμα 4.11

Μία εναλλακτική προσέγγιση καλείται Distributed Bag-of-Words (PV-DBOW) και σχηματικά αναπαρίσταται στο σχήμα 4.12. Σε κάθε παράθυρο το δίκτυο τροφοδοτείται από το διάνυσμα του εγγράφου, στο οποίο ανήκει το παράθυρο, και οι παράμετροι προσαρμόζονται ώστε το δίκτυο να μπορεί να προβλέπει επιτυχώς τυχαία επιλεγμένες λέξεις του παραθύρου. Οι δύο διαφορετικές προσεγγίσεις εμπνέονται από τις παραλλαγές Skip-Gram και Continuous Bag-of-Words του μοντέλου word2vec. Σε κάθε περίπτωση το μοντέλο εκπαιδεύεται με τρόπο εντελώς αντίστοιχο του μοντέλου word2vec και παράγει word vectors και document vectors σταθερής διάστασης d. Με αυτό τον τρόπο τα έγγραφα απεικονίζονται στον ίδιο διανυσματικό χώρο με τις λέξεις και ιδανικά, ενσωματώνουν όλη τη σημασιολογική πληροφορία που προέρχεται από τις λέξεις που τα συνθέτουν αλλά και από τη σειρά των λέξεων, τουλάχιστον στην τεχνική Distributed Memory.



Σχήμα 4.12

Μετά την εκτίμηση των document vectors, αυτά μπορούν να χρησιμοποιηθούν σαν χαρακτηριστικά σε προβλήματα ταξινόμησης. Οι Le και Mikolov αναφέρουν καλές επιδόσεις σε προβλήματα sentiment analysis.

5 Υλοποίηση και Αποτελέσματα

Μετά την απαραίτητη θεωρητική εισαγωγή των κεφαλαίων 3 και 4 είμαστε πλέον σε θέση να δώσουμε αναλυτικά την υλοποίηση του συστήματός μας για το πρόβλημα της ανάλυσης συναισθήματος στο σύνολο δεδομένων που περιγράψαμε στο κεφάλαιο 2.

5.1 Η Υλοποίηση

Το σύνολο δεδομένων που περιγράψαμε στην ενότητα 2.1 αποτελείται από 20,811 tweets εκ των οποίων τα 10,408 είναι θετικά και τα 10,403 αρνητικά. Αρχικά το σύνολο των tweets υφίσταται προεπεξεργασία σύμφωνα με τη διαδικασία της ενότητας 2.2. Στη συνέχεια κάθε tweet διαχωρίζεται στις επιμέρους λέξεις - tokenization - και προκύπτουν 20,811 λίστες λέξεων. Συνολικά τα δεδομένα περιέχουν 452,400 λέξεις-unigrams, 431,589 bigrams και 410,778 trigrams. Από το σύνολο των n-grams εξάγονται 25,096 μοναδικά unigrams, 175,005 μοναδικά bigrams και 320,132 μοναδικά trigrams. Στους πίνακες 5.1-5.3 δίνονται ενδεικτικά τα 50 συχνότερα unigrams (χωρίς τις stopwords), τα 30 συχνότερα bigrams και τα 10 συχνότερα trigrams που απαντώνται στο σώμα των tweets.

Από τις 20,811 λίστες λέξεων εξάγονται χαρακτηριστικά και έτσι κάθε tweet αντιστοιχίζεται σε ένα σύνολο χαρακτηριστικών και μία ετικέτα θετικού ή αρνητικού συναισθήματος. Το σύνολο των tweets χωρίζεται σε δεδομένα εκπαίδευσης (training data) και δεδομένα δοκιμής (testing data) με λόγο διαχωρισμού 0.9 .

unigram	πλήθος	unigram	πλήθος	unigram	πλήθος
.	19,315	-	1,469	love	826
<number>	10,112	<elong>	1,265	(	793
,	8,778	going	1,259	friday	789
<user>	8,605	&	1,228	)	787
!	8,491	get	1,226	/	771
<allcaps>	7,133	see	1,194	new	766
<repeat>	6,915	night	1,143	can't	762
<hashtag>	3,739	like	1,128	don't	754
<url>	3,722	good	1,089	sunday	751
?	2,538	go	1,067	got	730
tomorrow	2,401	time	1,046	back	727
“	2,090	it's	1,036	want	657
:	1,848	'	977	tonight	651
may	1,831	st	951	still	649
day	1,781	<smile>	950	saturday	646
th	1,649	today	927	last	636
i'm	1,647	one	882		

Πίνακας 5.1

bigram	πλήθος	bigram	πλήθος
. <repeat>	4,955	at the	701
! <repeat>	1,720	<allcaps> !	696
<number> th	1,638	to see	574
the <number>	1,339	, i	569
in the	1,191	to be	543
<repeat> url	1,098	<number> nd	538
. i	951	, but	534
going to	902	i have	510
for the	868	. <hashtag>	505
<number> st	861	<allcaps> .	490
of the	855	tomorrow .	467
on the	835	will be	443
<user> <user>	822	. <url>	433
<user> i	748	<number> <number>	415
to the	742	may be	415

Πίνακας 5.2

Οι αλγόριθμοι μηχανικής μάθησης εκπαιδεύονται στα χαρακτηριστικά και τις ετικέτες των tweets εκπαίδευσης και αξιολογούνται σε αυτά των tweets δοκιμής. Η μετρική επίδοσης που χρησιμοποιείται είναι απλά η ακρίβεια accuracy καθώς η ταξινόμηση γίνεται σε δύο κλάσεις. Τυπικά ορίζεται ως

trigram	πλήθος
. <repeat> <url>	1,050
the <number> th	549
the <number> st	311
. <repeat> i	277
<user> <user> <user>	237
<allcaps> ! <repeat>	230
on the <number>	192
the <number> nd	186
for the <number>	182
going to be	177

Πίνακας 5.3

$$accuracy = \frac{\textit{αριθμώς σωστά ταξινομημένων δειγμάτων στο σύνολο δοκιμής}}{\textit{συνολικός αριθμός δειγμάτων στο σύνολο δοκιμής}}$$

Στην ενότητα 5.2 παρουσιάζονται τα αποτελέσματα της ταξινόμησης για διάφορους τρόπους εξαγωγής χαρακτηριστικών και τους διάφορους αλγορίθμους μηχανικής μάθησης που είδαμε αναλυτικά στο κεφάλαιο 3. Αναλυτικά οι αλγόριθμοι της υλοποίησης είναι οι ακόλουθοι

Gaussian Naive Bayes

Ο απλοϊκός ταξινομητής Bayes με γκαουσιανή εκτίμηση των συναρτήσεων πυκνότητας πιθανότητας. Υλοποιείται με τη βοήθεια του sklearn module.

```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
```

Multinomial Naive Bayes

Ο Naive Bayes ταξινομητής με θεώρηση multinomial κατανομών. Απαιτεί διακριτά χαρακτηριστικά αλλά στην πράξη εκπαιδεύεται και με συνεχή.

```
from sklearn.naive_bayes import MultinomialNB
classifier = MultinomialNB()
```

Bernoulli Naive Bayes

Ο απλοϊκός ταξινομητής κατά Bayes με Bernoulli εκτίμηση των κατανομών των χαρακτηριστικών. Απαιτεί δυαδικά χαρακτηριστικά όπως είδαμε στην ενότητα 3.1 ωστόσο εκπαιδεύεται και σε συνεχή χαρακτηριστικά με αντιστοίχιση θετικών τιμών στο 1 και αρνητικών τιμών στο 0 (binarize threshold = 0).

```
from sklearn.naive_bayes import BernoulliNB
classifier = BernoulliNB()
```

Και οι τρεις παραπάνω ταξινομητές εκτιμούν τις prior πιθανότητες στο σύνολο δεδομένων, δηλαδή

$$P(C_i) = \frac{\textit{πλήθος δεδομένων στην κλάση } C_i}{\textit{συνολικό πλήθος δεδομένων}} \, , i = 1,2$$

k –Nearest Neighbors

Ο αλγόριθμος ταξινόμησης των k πλησιέστερων γειτόνων. Υλοποιείται με το sklearn και εξετάζονται τιμές $k = 1,3,5$ και 7. Χρησιμοποιούνται ίσα βάρη για όλους τους γείτονες δηλαδή κάθε γείτονας συνεισφέρει το ίδιο στην ψηφοφορία και σαν μετρική απόστασης επιλέγεται η απόσταση Minkowski για $p = 2$ που ισοδυναμεί με την ευκλείδια απόσταση.

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=k)
```

Logistic Regression ή Max Entropy

Ο αλγόριθμος λογιστικής παλινδρόμησης ή μέγιστης εντροπίας. Υλοποιείται επίσης με το sklearn module και χρησιμοποιεί τη βιβλιοθήκη liblinear για βελτιστοποίηση.

```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression()
```

Stochastic Gradient Descent

Είναι ένας πλήρως παραμετροποιήσιμος γραμμικός ταξινομητής του sklearn που εκπαιδεύεται με τη μέθοδο stochastic gradient descent. Ανάλογα με τις τιμές των παραμέτρων μπορεί να συμπεριφερθεί σαν γραμμική παλινδρόμηση, γραμμική μηχανή διανυσμάτων υποστήριξης, λογιστική παλινδρόμηση αλλά και άλλους γραμμικούς ταξινομητές. Στην υλοποίησή μας εξετάζεται η SVM εκδοχή του. Είναι αρκετά πιο γρήγορος από τις κλασσικές SVM υλοποιήσεις εξαιτίας του τρόπου εκπαίδευσής του.

```
from sklearn.linear_model import SGDClassifier
classifier = SGDClassifier(loss='hinge', penalty='l2')
```

SVC

Ο κλασσικός ταξινομητής SVM γραμμικού πυρήνα. Χρησιμοποιεί τη βιβλιοθήκη libsvm για βελτιστοποίηση.

```
from sklearn.svm import SVC
```

```
classifier = SVC(kernel='linear')
```

LinearSVC

Ο ταξινομητής SVM γραμμικού πυρήνα που χρησιμοποιεί τη βιβλιοθήκη liblinear για την εκπαίδευση. Ειναι σημαντικά πιο γρήγορος από τον απλό SVC σε δεδομένα πολλών διαστάσεων.

```
from sklearn.svm import LinearSVC
classifier = LinearSVC()
```

NuSVC

SVM ταξινομητής γραμμικού πυρήνα με μία παράμετρο ελέγχου του αριθμού των διανυσμάτων υποστήριξης. Η βελτιστοποίηση γίνεται με τη βιβλιοθήκη libsvm. Είναι πιο γρήγορος από τον απλό SVC αλλά όχι το ίδιο γρήγορος με τον LinearSVC.

```
from sklearn.svm import NuSVC
classifier = NuSVC()
```

SVC rbf kernel

Ο SVM ταξινομητής με rbf πυρήνα. Η παράμετρος $\gamma = \frac{1}{2\sigma^2}$ επιλέγεται ίση με

$$\gamma = \frac{1}{\textit{αριθμός χαρακτηριστικών}}$$

καθώς δίνει τα βέλτιστα αποτελέσματα στο σύνολο των πειραμάτων.

```
from sklearn.svm import SVC
classifier = SVC(kernel='rbf', gamma=1/no_features)
```

SVC polynomial kernel

Ο SVM ταξινομητής με πολυωνυμικό πυρήνα και $p = 3$.

```
from sklearn.svm import SVC
classifier = SVC(kernel='poly', degree=3)
```

Multi-Layer Perceptron

Ένα πολυεπίπεδο perceptron με ένα κρυφό επίπεδο, dim νευρώνες στην είσοδο, $2 \cdot dim$ κρυφούς νευρώνες και 2 νευρώνες εξόδου, όπου dim ο αριθμός των χαρακτηριστικών.

Εκπαιδεύεται με τον αλγόριθμο backpropagation σε 25 εποχές με ρυθμό μάθησης 0.01 και σταθερά ορμής 0.01. Υλοποιείται με τη βοήθεια του pybrain module.

CNN rand, static, nonstatic και multichannel

Οι τέσσερις παραλλαγές του συνελικτικού δικτύου που προτείνει ο Yoon Kim στο [8] και περιγράφονται στην ενότητα 3.6. Υλοποιείται στο torch framework. Η εκπαίδευση γίνεται με 10-fold cross validation στα δεδομένα εκπαίδευσης που αποτελούν το 90% του συνόλου για 25 εποχές και η αξιολόγηση στα υπόλοιπα δεδομένα που αποτελούν το σύνολο δοκιμής.

5.2 Αποτελέσματα

5.2.1 Bag-of-Words και Ανάλυση Συναισθήματος

Σε αυτή την ενότητα δίνονται τα αποτελέσματα της εφαρμογής των διαφόρων αλγορίθμων επιβλεπόμενης μάθησης για την μέθοδο εξαγωγής χαρακτηριστικών Bag-of-Words. Στον πίνακα 5.4 φαίνονται τα αποτελέσματα για τις τρεις τεχνικές term occurrence, term frequency και tf-idf και με χρήση 1,000 και 2,000 unigrams σαν χαρακτηριστικά. Η αξιολόγηση γίνεται με τον μέσο όρο των validation scores σε 10-fold cross validation του συνόλου των δεδομένων.

Algorithms	**Term Occurrence**		**Term Frequency**		**Tf-Idf**	
	1,000u	2,000u	1,000u	2,000u	1,000u	2,000u
Gaussian Naive Bayes	79.35	78.65	79.13	78.60	78.93	78.72
Multinomial Naive Bayes	82.42	83.54	81.70	83.01	82.37	83.24
Bernoulli Naive Bayes	82.89	83.81	82.89	83.82	82.84	83.69
Nearest Neighbors $k = 1$	64.19	63.34	62.92	63.76	66.29	63.50
Nearest Neighbors $k = 3$	62.89	60.67	63.28	63.04	65.36	60.46
Nearest Neighbors $k = 5$	61.90	59.21	63.39	62.44	63.71	57.14
Nearest Neighbors $k = 7$	61.05	57.66	62.84	62.32	61.55	55.21
SGD	82.57	83.41	82.38	82.11	80.81	82.74
Logistic Regression	**85.40**	**86.08**	**85.04**	**85.96**	**85.01**	84.76
Linear SVC	85.30	85.39	85.02	85.11	84.94	84.79
SVC rbf kernel	77.66	76.22	76.97	76.59	84.72	**84.85**

Πίνακας 5.4

Ο αλγόριθμος Max Entropy σημειώνει καλές επιδόσεις στα διακριτά χαρακτηριστικά και οι SVMs στα συνεχή χαρακρτηριστικά tf-idf. Οι αλγόριθμοι Nearest Neighbors δίνουν φτωχά αποτελέσματα, όπως είναι λογικό καθώς σε αυτή τη μέθοδο εξαγωγής

χαρακτηριστικών η απόσταση στο χώρο χαρακτηριστικών δεν έχει άμεση σχέση με τη σημασιολογική εγγύτητα των εγγράφων. Ο Bernoulli Naive Bayes ταξινομητής δίνει συστηματικά καλύτερα αποτελέσματα από τους υπόλοιπους αλγορίθμους Bayes και ο Gaussian Naive Bayes δεν φαίνεται να πλεονεκτεί ούτε στα συνεχή χαρακτηριστικά tf-idf. Στον πίνακα 5.5 φαίνεται πως επηρεάζεται η επίδοση των ταξινομητών από την αύξηση του λεξικού σε 5,000 και 10,000 unigrams. Αυτή τη φορά η αξιλόγηση γίνεται με το μέσο όρο των validation scores σε 4-fold cross validation του συνόλου των δεδομένων.

Ο αλγόριθμος Logistic Regression εξακολουθεί να σημειώνει τα καλύτερα αποτελέσματα. Στην tf-idf εξαγωγή χαρακτηριστικών όλοι οι ταξινομητές πλήττονται από την μεγάλη αύξηση της διάστασης των διανυσμάτων εκπαίδευσης, εκτός από τον Bernoulli Naive Bayes που παρουσιάζει καλή επίδοση σε αυτή την περίπτωση.

Στους πίνακες 5.6 με 5.8 συμπεριλαμβάνονται bigrams και trigrams στο λεξικό. Η αξιολόγηση γίνεται και πάλι, όπως στον πίνακα 5.4, με το μέσο όρο των validation scores σε 10-fold cross validation.

Algorithms	**Term Occurrence**		**Term Frequency**		**Tf-Idf**	
	5000u	10000u	5000u	10000u	5000u	10000u
Gaussian Naive Bayes	70.35	66.46	69.31	66.30	68.72	64.27
Multinomial Naive Bayes	84.22	84.95	83.26	84.14	82.80	81.55
Bernoulli Naive Bayes	84.39	84.53	83.78	84.69	**83.36**	**84.18**
SGD	84.14	84.05	82.93	82.25	81.60	82.99
Logistic Regression	**86.77**	**86.88**	**85.84**	**86.47**	82.06	83.82
Linear SVC	84.58	84.63	83.57	84.87	79.04	80.48

Πίνακας 5.5

Algorithms	**Term Occurrence**	
	1000u+500b	1000u+500b+100t
Gaussian Naive Bayes	81.05	80.94
Multinomial Naive Bayes	81.46	81.25
Bernoulli Naive Bayes	81.75	81.25
SGD	82.67	82.48
Logistic Regression	**85.37**	**85.17**
Linear SVC	85.09	85.02

Πίνακας 5.6

Algorithms	Term Frequency	
	1000u+500b	1000u+500b+100t
Gaussian Naive Bayes	80.84	80.65
Multinomial Naive Bayes	80.96	80.78
Bernoulli Naive Bayes	81.81	81.28
SGD	81.49	81.92
Logistic Regression	**85.08**	**85.00**
Linear SVC	84.73	84.63

Πίνακας 5.7

Algorithms	Tf - Idf	
	1000u+500b	1000u+500b+100t
Gaussian Naive Bayes	80.90	80.73
Multinomial Naive Bayes	81.69	81.60
Bernoulli Naive Bayes	81.76	81.42
SGD	81.53	81.42
Logistic Regression	**84.90**	**84.95**
Linear SVC	84.66	84.79

Πίνακας 5.8

Συμπερασματικά, μπορούμε να πούμε πως οι τεχνικές term occurrence και term frequency αποδίδουν καλύτερα από την tf-idf. Μεταξύ των δύο δεν υπάρχουν σημαντικές διαφορές καθώς τα έγγραφα έχουν μικρό μήκος (sentence-based sentiment analysis) και οι δύο αναπαραστάσεις τείνουν να ταυτίζονται. Οι αλγόριθμοι k-Nearest Neighbors αποτελούν κακή σχεδιαστική επιλογή για τη μέθοδο Bag-of-Words όπως και οι SVMs πυρήνα πέραν του γραμμικού. Μόνο ο πυρήνας rbf παρουσιάζει καλά αποτελέσματα και μόνο σε συνεχή χαρακτηριστικά tf-idf. Οι SVMs γραμμικού πυρήνα παρουσιάζουν ανταγωνιστικές επιδόσεις σε όλες τις περιπτώσεις και ειδικά ο αλγόριθμος LinearSVC αποτελεί καλή επιλογή καθώς δεν υστερεί των υπολοίπων και εκπαιδεύεται σημαντικά πιο γρήγορα. Ο Bernoulli Naive Bayes ταξινομητής δίνει καλά αποτελέσματα σε όλες τις περιπτώσεις και ξεχωρίζει όταν η διάσταση των δεδομένων γίνεται πολύ μεγάλη (>5,000) ενώ ο Max Entropy ταξινομητής είναι η καλύτερη επιλογή για Bag-of-Words χαρακτηριστικά σύμφωνα με τα παραπάνω αποτελέσματα. Τέλος η ταξινόμηση στο σύνολο των δεδομένων μας δεν φαίνεται να επηρεάζεται θετικά από την χρήση n-grams σαν χαρακτηριστικά.

Σε γενικές γραμμές ο συνδυασμός των Bag-of-Words χαρακτηριστικών με τους ταξινομητές Naive Bayes, Max Entropy και SVM δίνει πολύ καλά αποτελέσματα για το πρόβλημα της ανάλυσης συναισθήματος σε κείμενο. Στο ίδιο συμπέρασμα καταλήγουν και ένα πλήθος ερευνητικών δημοσιεύσεων με πλέον χαρακτηριστικά παραδείγματα την δουλειά των Pang και Lee [19] και την πιο πρόσφατη των Wang και Manning [28].

5.2.2 Word Vectors και Ανάλυση Συναισθήματος

Σε αυτή την ενότητα θα εξεταστεί ένας εντελώς διαφορετικός τρόπος εξαγωγής χαρακτηριστικών που θεμελιώθηκε θεωρητικά στις ενότητες 4.2 και 4.3. Θα χρησιμοποιηθούν pretrained word vectors του μοντέλου GloVe στο Twitter και θα εξεταστούν διάφοροι τρόποι σύνθεσής τους για την εξαγωγή χαρακτηριστικών για τα tweets. Τα pretrained word vectors παράγονται μέσω εκπαίδευσης του μοντέλου GloVe σε ένα unsupervised σώμα από 2 δισεκατομμύρια tweets με 27 δισεκατομμύρια tokens και διατίθενται από την αντίστοιχη σελίδα του Stanford NLP σε εκδοχές 25, 50, 100 και 200 διαστάσεων. Το λεξικό περιλαμβάνει 1.2 εκατομμύρια tokens και συγκρίνεται με το δικό μας λεξικό των 25,096 unigrams για να προκύψουν word vectors για τις λέξεις του δικού μας συνόλου. Λέξεις που απαντώνται στο δικό μας σύνολο δεδομένων αλλά όχι στο λεξικό των pretrained word vectors του GloVe αντιστοιχίζονται σε μηδενικά διανύσματα. Αρχικά εξετάζονται οι διάφοροι τρόποι σύνθεσης των word vectors δηλαδή πρόσθεση με κανονικοποίηση, πρόσθεση χωρίς κανονικοποίηση, tf-idf πρόσθεση, πολλαπλασιασμός με ή χωρίς κανονικοποίηση και concatenation για τα διανύσματα 25 διαστάσεων. Τα αποτελέσματα φαίνονται στον πίνακα 5.9. Η αξιολόγηση γίνεται με το μέσο όρο των validation scores σε 10-fold cross validation όλου του συνόλου. Για το δίκτυο MLP χρησιμοποιείται το 10% του συνόλου σαν test dataset και η εκπαίδευση γίνεται με 10-fold cross validation στο υπόλοιπο σύνολο για 25 εποχές.

Ο πολλαπλασιασμός των word vectors δίνει φτωχά αποτελέσματα κοντά στο λόγο τυχαίας ταξινόμησης (random classifying 50% για ισοπληθείς κλάσεις) και για αυτό το λόγο δεν παρουσιάζεται στα αποτελέσματα. Η συνένωση των διανυσμάτων δίνει διανύσματα διάστασης $25 \cdot 61 = 1525$ διαστάσεων καθώς το tweet με τα περισσότερα tokens στο σύνολό μας περιέχει 61 λέξεις. Η εκπαίδευση των ταξινομητών SVC, NuSVC, SVC rbf kernel, SVC polynomial kernel του sklearn αλλά και του Multi-Layer Perceptron σε συνεχή δεδομένα τόσο μεγάλης διάστασης είναι υπερβολικά αργή και επίσης δεν παρουσιάζονται αποτελέσματα.

Η πρόσθεση των word vectors με ή χωρίς κανονικοποίηση δίνει τα καλύτερα αποτελέσματα και η τεχνική αυτή υιοθετείται και για τις υπόλοιπες διαστάσεις. Τα αποτελέσματα δίνονται στον πίνακα 5.10. Η αξιολόγηση γίνεται με τον ίδιο τρόπο.

Είναι αξιοσημείωτο το πώς η απλή πρόσθεση διανυσματικών αναπαραστάσεων που κωδικοποιούν τη γενικότερη σημασιολογική σχέση μεταξύ λέξεων, αποδίδει τόσο καλά στο πρόβλημα της ανάλυσης συναισθήματος. Τα word vectors προκύπτουν από unsupervised σώμα κειμένου και δεν είναι task specific για το πρόβλημα της ανάλυσης συναισθήματος, δηλαδή δεν είναι προσαρμοσμένα ώστε να συλλαμβάνουν τη σχέση των λέξεων σε επίπεδο συναισθηματικής πόλωσης. Αντ'αυτού είναι globally tuned (αφού προέρχονται από unsupervised κείμενο) για να μπορούν να χρησιμοποιηθούν σε διάφορα προβλήματα επεξεργασίας φυσικού λόγου. Εξαιτίας των παραπάνω λόγων, είναι ενδιαφέρον το πως η πρόσθεση των word vectors δίνει καλά αποτελέσματα συγκρίσιμα της μεθόδου Bag-of-Words. Σε αυτό το σημείο, σημειώνεται ότι είναι δυνατόν να προκύψουν task specific διανυσματικές αναπαραστάσεις με μία παραλλαγή του μοντέλου word2vec ώστε να εκπαιδεύεται και σε supervised δεδομένα. Μία τέτοια προσέγγιση, για sentiment specific διανυσματικές αναπαραστάσεις εξετάζεται από τους Tang et al. στο [25] (SSWE - Sentiment Specific Word Embeddings).

Algorithms	GloVe Twitter Pre-Trained Word Vectors dim = 25			
	Addition Norm	Addition No Norm	Tf-Idf Addition	Concatenation
Gaussian Naive Bayes	73.94	72.34	67.84	50.02
Nearest Neighbor $k = 1$	70.48	71.97	70.74	58.38
Nearest Neighbor $k = 3$	73.86	74.68	73.55	56.68
Nearest Neighbor $k = 5$	75.37	76.09	74.98	55.38
Nearest Neighbor $k = 7$	76.11	76.99	75.57	54.23
SGD	73.17	75.16	68.95	73.02
Logistic Regression	78.73	79.53	77.10	**77.96**
SVC	78.77	79.49	77.10	-
Linear SVC	78.74	79.42	77.17	75.45
Nu SVC	80.13	80.16	**78.90**	-
SVC rbf kernel	**80.38**	**80.53**	78.88	-
SVC polynomial kernel	78.59	79.31	76.84	-
Multi-Layer Perceptron	78.44	79.04	76.09	-

Πίνακας 5.9

Algorithms	GloVe Twitter Pre-Trained Word Vectors merged by addition with no normalization		
	dim = 50	dim = 100	dim = 200
Gaussian Naive Bayes	72.61	72.46	71.83
Nearest Neighbor $k = 1$	73.60	74.39	74.07
Nearest Neighbor $k = 3$	76.30	76.94	76.32
Nearest Neighbor $k = 5$	77.42	78.10	77.02
Nearest Neighbor $k = 7$	78.26	78.93	77.37
SGD	75.64	78.51	78.76
Logistic Regression	81.89	83.41	84.50
SVC	81.90	83.38	84.49
Linear SVC	81.83	83.38	84.51
Nu SVC	81.89	83.26	83.96
SVC rbf kernel	**82.70**	**84.72**	**85.81**
SVC polynomial kernel	81.86	83.44	84.30
Multi-Layer Perceptron	80.59	82.05	83.46

Πίνακας 5.10

Όσον αφορά τους αλγορίθμους μηχανικής μάθησης, ο SVM πυρήνα rbf δίνει τα καλύτερα αποτελέσματα, παρουσιάζοντας συστηματικά μία μικρή βελτίωση έναντι των γραμμικών SVMs και του Max Entropy ταξινομητή. Ο αλγόριθμος k-Nearest Neighbors

συμπεριφέρεται πολύ καλύτερα σε σχέση με τα Bag-of-Words χαρακτηριστικά αφού πλέον ο χώρος χαρακτηριστικών κωδικοποιεί σημασιολογική σχέση και η απόσταση σημείων στο χώρο έχει νόημα. Τέλος, πρέπει να σημειωθεί, ότι είναι δυνατόν να προκύψουν καλύτερα αποτελέσματα ακόμα πιο κοντά ή και καλύτερα από την Bag-of-Words μέθοδο με καλύτερο tuning του νευρωνικού δικτύου.

Στη συνέχεια εξετάζουμε τα μοντέλα word2vec και doc2vec. Εκπαιδεύεται ένα μοντέλο με τη βοήθεια του module της python gensim, το οποίο παράγει word vectors για κάθε λέξη του λεξικού μας και με τη βοήθεια αυτών, document vectors για κάθε tweet του συνόλου. Στη συνέχεια τα document vectors των tweets μαζί με τις ετικέτες προωθούνται στους διάφορους ταξινομητές. Πήραμε τα καλύτερα αποτελέσματα για Distributed Bag-of-Words αρχιτεκτονική, μέγεθος παραθύρου 10 και negative sampling ενημέρωση βαρών. Στον πίνακα 5.11 παρουσιάζονται αποτελέσματα για διαστάσεις 300, 400 και 500 των διανυσματικών αναπαραστάσεων λέξεων και εγγράφων. Η αξιολόγηση γίνεται και πάλι όπως στα προηγούμενα.

Algorithms	**Document features with doc2vec**		
	dim = 300	dim = 400	dim = 500
Gaussian Naive Bayes	76.60	77.32	76.04
Nearest Neighbor $k = 1$	63.02	60.79	62.19
Nearest Neighbor $k = 3$	63.69	61.84	58.82
Nearest Neighbor $k = 5$	62.01	61.39	56.13
Nearest Neighbor $k = 7$	61.45	59.80	53.88
SGD	73.03	75.88	75.50
Logistic Regression	78.92	79.79	79.69
Linear SVC	78.86	79.85	79.63
Nu SVC	78.99	**80.67**	**80.22**
SVC rbf kernel	78.68	80.25	79.34
Multi-Layer Perceptron	**79.42**	76.02	79.10

Πίνακας 5.11

Τα αποτελέσματα δεν είναι το ίδιο ικανοποιητικά με την άθροιση των pretrained vectors του μοντέλου GloVe και αυτό οφείλεται κατά κύριο λόγο στο μικρό μέγεθος του συνόλου δεδομένων. Στην παραπάνω υλοποίηση τόσο τα word vectors όσο και τα document vectors εκπαιδεύονται στο σύνολο των 20,811 tweets. Το σώμα κειμένου είναι υπερβολικά μικρό για την εκπαίδευση και των δύο διανυσματικών αναπαραστάσεων αλλά ειδικά των word vectors. Τα pretrained word vectors του μοντέλου GloVe, για παράδειγμα εκπαιδεύονται σε 2 δισεκατομμύρια tweets. Οι Le και Mikolov στο [10] σημειώνουν state-of-the-art αποτελέσματα στην ανάλυση συναισθήματος με το μοντέλο doc2vec, σε χαρακτηριστικά σύνολα δεδομένων αλλά εκπαιδεύουν τα document vectors σε πολύ μεγαλύτερο αριθμό εγγράφων (100,000 έγγραφα στο IMDB Dataset των Maas et al. και 239,232 labeled φράσεις στο Stanford Sentiment Treebank Dataset των Socher et al.). Επίσης υπάρχει η δυνατότητα αρχικοποίησης των word vectors στο μοντέλο doc2vec με τα pretrained word

vectors του μοντέλου word2vec, που έχουν εκπαιδευτεί σε δεδομένα από το Google News με περίπου 100 δισεκατομμύρια λέξεις συνολικά, για επίτευξη καλύτερων αποτελεσμάτων.

5.2.3 CNNs και Ανάλυση Συναισθήματος

Στην τελευταία ενότητα των αποτελεσμάτων παρουσιάζεται η υλοποίηση του συνελικτικού δικτύου του Yoon Kim για ταξινόμηση κειμένων. Εξετάζονται οι τέσσερις εκδοχές του μοντέλου και για word vectors χρησιμοποιούνται τόσο τα pretrained word vectors του μοντέλου word2vec 300 διαστάσεων όσο και τα pretrained word vectors του μοντέλου GloVe στο Twitter 200 διαστάσεων. Χρησιμοποιούνται 100 feature maps για παράθυρα ύψους $h = 3$, 100 για $h = 4$ και 100 για $h = 5$. Το δίκτυο εκπαιδεύεται με 10-fold cross validation σε τυχαία επιλογή του 90% των tweets και αξιολογείται στο υπόλοιπο 10% για 25 εποχές. Στους πίνακες 5.12 και 5.13 φαίνονται οι μέσοι όροι των test accuracies.

	GloVe Twitter pretrained dim = 200 avg test accuracy
CNN - rand	86.06
CNN - static	87.12
CNN - nonstatic	**87.50**
CNN - multichannel	86.74

Πίνακας 5.12

	word2vec pretrained dim=300 avg test accuracy
CNN - static	87.77
CNN - nonstatic	**88.09**

Πίνακας 5.13

5.3 Επίλογος

Στην ενότητα 5.2 παρουσιάσαμε τα αποτελέσματα για το πρόβλημα της ανάλυσης συναισθήματος, των αλγορίθμων μηχανικής μάθησης του κεφαλαίου 3 και των μεθόδων εξαγωγής χαρακτηριστικών του κεφαλαίου 4 στο σύνολο δεδομένων του Twitter. Τα διάφορα μοντέλα παρουσιάζουν επιδόσεις που κυμαίνονται στο 80-85% accuracy στα δεδομένα δοκιμής και τις υψηλότερες επιδόσεις πετυχαίνουν ο αλγόριθμος Max Entropy με χαρακτηριστικά Bag-of-Words, οι SVMs με pretrained word vectors στο Twitter και το συνελικτικό δίκτυο με accuracy της τάξης 86 με 88%.

Σχετικά με τον αλγόριθμο k-Nearest Neighbors μπορούμε να συμπεράνουμε ότι υστερεί έναντι των υπολοίπων στο πρόβλημα της ανάλυσης συναισθήματος και γενικά σε προβλήματα ταξινόμησης κειμένου. Ο χώρος χαρακτηριστικών δε διαθέτει τις απαραίτητες ιδιότητες ώστε τα τοπολογικά κριτήρια του αλγορίθμου k-Nearest Neighbors να δώσουν ανταγωνιστικές επιδόσεις.

Οι Μπεϋζιανοί ταξινομητές παρέχουν πολύ γρήγορη υλοποίηση, είναι απλοί και συνεργάζονται καλά με τα χαρακτηριστικά Bag-of-Words. Είναι ιδιαίτερα δημοφιλείς σε εργασίες επεξεργασίας κειμένου και τα αποτελέσματα το επιβεβαιώνουν. Ξεχωρίζει ο ταξινομητής Bayes με Bernoulli εκτίμηση των posterior πιθανοτήτων.

Οι μηχανές διανυσμάτων υποστήριξης δίνουν τις καλύτερες επιδόσεις, όταν τα pretrained word vectors αθροίζονται για την αναπαράσταση εγγράφων. Ο πυρήνας rbf εκμεταλλεύεται το γεγονός ότι τα δεδομένα δεν είναι γραμμικά διαχωρίσιμα και ξεπερνά τους SVMs γραμμικού πυρήνα σε αυτή την περίπτωση.

Το πολυεπίπεδο perceptron επίσης δίνει καλά αποτελέσματα στην περίπτωση άθροισης των word vecors αλλά υστερεί γενικά των υπολοίπων αλγορίθμων. Είναι πιθανό με κατάλληλη προσαρμογή του δικτύου και προσθήκη περισσότερων κρυφών επιπέδων να επιτευχθούν καλύτερα αποτελέσματα.

Ο αλγόριθμος Max Entropy ή Logistic Regression δίνει συστηματικά πολύ καλά αποτελέσματα και όταν συνδυάζεται με χαρακτηριστικά Bag-of-Words, πετυχαίνει την καλύτερη επίδοση στο πειραμά μας μεταξύ των αλγορίθμων κλασσικής μηχανικής μάθησης.

Σχετικά με τις μεθόδους εξαγωγής χαρακτηριστικών, η Bag-of-Words μέθοδος παραμένει μία πολύ καλή επιλογή για το πεδίο της ανάλυσης συναισθήματος. Τουλάχιστον στη δική μας υλοποίηση, δεν φαίνεται να επηρεάζεται θετικά από την χρήση n-grams στο λεξικό ενώ ένα λεξικό με 2,000 έως 5,000 unigrams είναι αρκετό για καλές επιδόσεις. Όσον αφορά τα word vectors η απλή πρόσθεσή τους για την αναπαράσταση εγγράφων δίνει τα καλύτερα αποτελέσματα. Όπως αναφέρθηκε και στην αντίστοιχη ενότητα, είναι αξιοσημείωτο το γεγονός ότι τα word vectors, που προκύπτουν από unsupervised εκπαίδευση σε μεγάλα σύνολα δεδομένων κειμένου, ενσωματώνουν ιδιότητες με τέτοιο τρόπο ώστε απλή άθροισή τους να δίνει πολύ καλά αποτελέσματα σε προβλήματα πάνω στα οποία δεν έχουν εκπαιδευτεί. Το μοντέλο doc2vec, που επίσης εξετάζεται, δίνει φτωχά αποτελέσματα, αν ληφθεί υπόψιν και η πολυπλοκότητα του, και μία πιθανή αιτία για αυτό είναι το μικρό σύνολο δεδομένων.

Επιπλέον σημειώνεται, ότι accuracy της τάξης του 85 με 88% είναι ασυνήθιστο, βάσει της βιβλιογραφίας, σε εφαρμογές document-based sentiment analysis. Ενδεικτικά στο movie review dataset των Pang και Lee, που περιλαμβάνει κριτικές ταινιών μήκους μερικών προτάσεων, οι αλγόριθμοι που εξετάσαμε δίνουν accuracy της τάξης του 75 με 77% και μόνο state-of-the-art deep learning μοντέλα προσεγγίζουν το 85 με 88%. Αυτό συμβαίνει εξαιτίας της αναγκαιότητας συνυπολογισμού της σειράς των λέξεων σε τόσο μικρά κείμενα για τον επιτυχή προσδιορισμό της συνολικής συναισθηματικής πολικότητας, κάτι που τα κλασσικά μοντέλα και οι κλασσικές μέθοδοι εξαγωγής χαρακτηριστικών δεν είναι σε θέση να κάνουν. Από τα αποτελέσματα του πειράματός μας λοιπόν, συμπεραίνουμε ότι η ανάλυση συναισθήματος σε δεδομένα του Twitter ίσως είναι μία πιο "εύκολη" εργασία. Τα δεδομένα του Twitter έχουν ιδιαιτερότητες που αν το σύστημα που υλοποιείται τις

εκμεταλλευτεί σωστά, βοηθούν σημαντικά την εκτίμηση του συναισθήματος. Αυτό φυσικά υποδεικνύει την αναγκαιότητα κατάλληλης προεπεξεργασίας. Η διαδικασία της προεπεξεργασίας, για τον χειρισμό των ιδιαίτερων tokens είναι ιδιαίτερα σημαντική για το πρόβλημα της ανάλυσης συναισθήματος σε δεδομένα του Twitter.

Το συνελικτικό δίκτυο είναι ένα πολύ ικανό μοντέλο και στην πράξη είναι το μόνο από αυτά που εξετάζονται που συνυπολογίζει τη σειρά των λέξεων (αν εξαιρέσουμε την παράθεση των word vectors η οποία όμως δεν δίνει καλά αποτελέσματα). Ταυτόχρονα είναι και ένα πολύπλοκο μοντέλο και η προσαρμογή του στα δεδομένα εκπαίδευσης είναι απαιτητική διαδικασία. Τέτοια μοντέλα σε μικρά σύνολα δεδομένων, είναι πολύ πιθανό να πάσχουν από overfitting οπότε σε κάθε περίπτωση είναι απαραίτητη η χρήση τεχνικών για την αποφυγή του. Ειδικά στο συνελικτικό δίκτυο χρησιμοποιούμε k-fold cross validation για την εκπαίδευση και τεχνικές dropout για την ενημέρωση των βαρών.

Συνοψίζοντας, τα αποτελέσματα της παρούσας διπλωματικής εργασίας μας οδηγούν στα εξής συμπεράσματα :

- Η διαδικασία της προεπεξεργασίας είναι βασικό κομμάτι για οποιαδήποτε μορφή εξαγωγής πληροφοριών από δεδομένα του Twitter, και ειδικά για το πρόβλημα της ανάλυσης συναισθήματος καθώς οι ειδικοί όροι όπως τα hashtags, τα emoticons και οι επιμηκυμένες λέξεις δίνουν πολύτιμη πληροφορία για το συναίσθημα.
- Οι κλασσικές μέθοδοι ανάλυσης συναισθήματος όπως ο αλγόριθμος Max Entropy και οι Μπεϋζιανοί ταξινομητές, με Bag-of-Words χαρακτηριστικά, παρέχουν καλά αποτελέσματα και γρήγορες υλοποιήσεις για το πρόβλημα της ανάλυσης συναισθήματος. Ειδικά σε δεδομένα του Twitter, αν γίνει η κατάλληλη προεπεξεργασία μπορούν να δώσουν accuracy της τάξης του 85%.
- Οι διανυσματικές αναπαραστάσεις λέξεων είναι ένα πολύτιμο εργαλείο για την επεξεργασία κειμένου. Μεταφράζουν σημασιολογικές και συντακτικές σχέσεις μεταξύ λέξεων σε γραμμικές ιδιότητες ενός διανυσματικού χώρου. Είναι ανεξάρτητες εφαρμογής, και ακόμα και απλή άθροισή τους δίνει συγκρίσιμα αποτελέσματα με τις κλασσικές μεθόδους, στο πρόβλημα της ανάλυσης συναισθήματος.
- Για την υπέρβαση του άνω ορίου που θέτουν οι κλασσικές μέθοδοι μάθησης (για παράδειγμα 75 με 77% στο movie review dataset και 85% στο δικό μας σύνολο) απαιτούνται πολύπλοκα μοντέλα, όπως το συνελικτικό δίκτυο που συνυπολογίζουν τη σειρά των λέξεων και χρησιμοποιούν word vectors. Το συνελικτικό δίκτυο που εξετάσαμε δίνει accuracy της τάξης του 88% στο σύνολο δεδομένων και σύμφωνα με το [8], 81% στο movie review dataset.

Μελλοντικοί Προσανατολισμοί Έρευνας

Η υλοποίηση πολύπλοκων νευρωνικών μοντέλων μάθησης είναι η κυρίαρχη ερευνητική τάση τα τελευταία χρόνια στα πεδία της μηχανικής μάθησης και της τεχνητής νοημοσύνης. Ο κλάδος αυτός , που ευρέως αναφέρεται ως βαθιά μάθηση (deep learning) έχει δώσει σημαντικά αποτελέσματα στα πεδία της υπολογιστικής όρασης, της επεξεργασίας φωνής και της ενισχυτικής μάθησης και τελευταία και στην επεξεργασία φυσικού λόγου. Με το μοντέλο word2vec και την αποδοτική υλοποίηση των νευρωνικών γλωσσικών μοντέλων που

παράγουν διανυσματικές αναπαραστάσεις λέξεων ([12], [13], [14] και [20]) το επίκεντρο της έρευνας μετατοπίστηκε στις αναπαραστάσεις αυτές και στην υλοποίηση πολύπλοκων νευρωνικών δικτύων που τις εκμεταλλεύονται για την επίλυση δύσκολων προβλημάτων της επεξεργασίας φυσικού λόγου, όπως η αυτόματη μετάφραση (machine translation), η αυτόματη απάντηση ερωτήσεων (question answering) και φυσικά η ανάλυση συναισθήματος. Παρά την επιτυχία των word vectors, η αναπαράσταση εγγράφων και φράσεων στον ίδιο διανυσματικό χώρο, με τρόπο που να ενσωματώνονται εξίσου ενδιαφέρουσες ιδιότητες, είναι ακόμη ένα ενεργό πεδίο έρευνας.

Το συνελικτικό δίκτυο είναι ένα μοντέλο που εμπνέεται από την επεξεργασία της πληροφορίας στον οπτικό φλοιό των βιολογικών οργανισμών και εξαρχής θεμελιώθηκε με γνώμονα την απόδοση σε προβλήματα επεξεργασίας εικόνων. Δέχεται raw data, δηλαδή απλά τα pixel της εικόνας και λόγω της ιδιαίτερης αρχιτεκτονικής του είναι σε θέση να ανακαλύπτει αναπαραστάσεις για τις δομές που περιλαμβάνει μία εικόνα. Ξεκινώντας από τα πρώτα κρυφά επίπεδα, μαθαίνει να αναγνωρίζει απλά σχήματα όπως γωνίες και ακμές και στη συνέχεια συνδυασμούς των δομών αυτών και συνδυασμούς των συνδυασμών για να ανακαλύψει τελικά πολύπλοκες δομές όπως πρόσωπα και να ικανοποιήσει τις ανάγκες ενός προβλήματος ταξινόμησης.

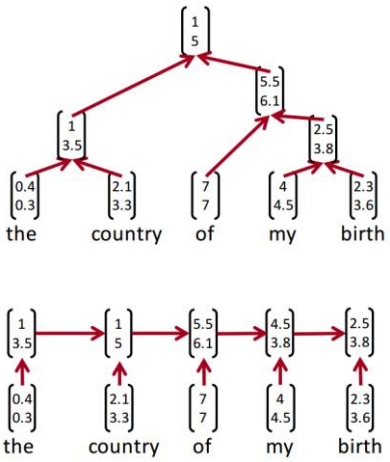
Μεταφέροντας την ίδια λογική στην επεξεργασία φυσικού λόγου, είναι λογικό να αναζητήσουμε ένα μοντέλο, που δέχεται raw data κειμένου, δηλαδή word vectors για κάθε λέξη, καθώς όπως είδαμε αποτελούν γενικές αναπαραστάσεις, και προσπαθεί να ανακαλύψει τις πολύπλοκες δομές μεταξύ των λέξεων. Στο συνελικτικό δίκτυο αυτό γίνεται με πράξεις συνέλιξης. Πως θα μπορούσε ενα δίκτυο να συμπεριφερθεί ανάλογα σε δεδομένα κειμένου;

Δύο δημοφιλή μοντέλα, που εμπνέονται από τη διαδικασία κατανόησης του λόγου, όπως το συνελικτικό δίκτυο εμπνέεται από τη διαδικασία επεξεργασίας της οπτικής πληροφορίας, είναι τα νευρωνικά δίκτυα με επανατροφοδότηση (recurrent neural networks) και τα αναδρομικά νευρωνικά δίκτυα (recursive neural networks). Τα δίκτυα με επανατροφοδότησση συχνά αναφέρονται ως αναδρομικά στην ελληνική βιβλιογραφία, ωστόσο εδώ θα χρησιμοποιήσουμε τους παραπάνω όρους για να μην υπάρχει σύγχηση μεταξύ των δύο διαφορετικών μοντέλων.

Τα δίκτυα με επανατροφοδότηση είναι γενικά διατάξεις νευρώνων, όπου επιτρέπονται συνδέσεις προς τα πίσω που δημιουργούν κατευθυνόμενους κυκλικούς γράφους. Σε τέτοια δυναμικά δίκτυα οι νευρώνες έχουν “μνήμη”, μία εσωτερική κατάσταση που σχετίζεται με προηγούμενα δεδομένα. Εξαιτίας αυτής της ιδιαιτερότητας, μοντελοποιούν δεδομένα με χρονική ή ακολουθιακή συσχέτιση, όπως για παράδειγμα ο φυσικός λόγος. Η διαδικασία εκπαίδευσης τέτοιων δικτύων παρουσιάζει σημαντικές δυσκολίες, ωστόσο έχουν χρησιμοποιηθεί επιτυχώς σαν γλωσσικά μοντέλα αλλά και σε εφαρμογές όπως η ανάλυση συναισθήματος. Στο [7] συγκεκριμένα, παρουσιάζεται ένα τέτοιο δίκτυο για τον προσδιορισμό της υποκειμενικότητας του συναισθήματος σε φράσεις που στοχεύουν σε κάποια οντότητα (entity).

Στα αναδρομικά δίκτυα, οι νευρώνες οργανώνονται βάσει κάποιας δομής ή δέντρου, με τα ίδια βάρη να μοιράζονται μεταξύ των νευρώνων διαφορετικών επιπέδων της δομής. Ουσιαστικά είναι νευρωνικά δίκτυα που οργανώνονται σε δενδρικές δομές και τα δίκτυα με επανατραφοδότηση αποτελούν υποπερίπτωσή τους, καθώς οργανώνονται με τη μορφή

αλυσίδας που είναι υποπερίπτωση δέντρου. Ο παραπάνω ισχυρισμός επεξηγείται στο σχήμα 5.1.



Σχήμα 5.1

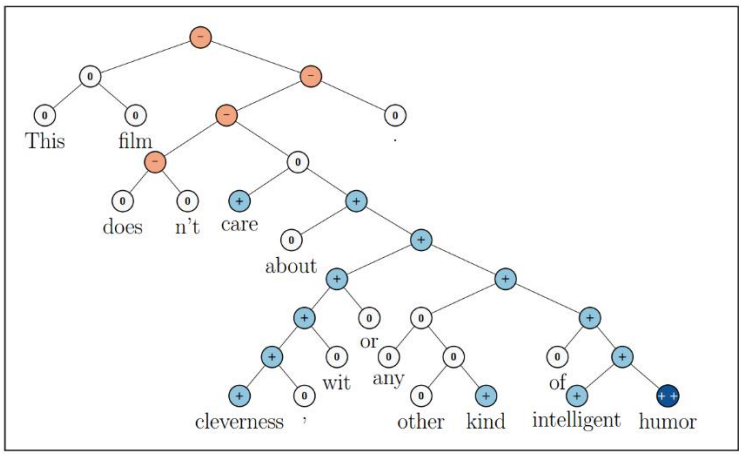
Οι Socher et al. [24] προτείνουν τη χρήση ενός recursive νευρωνικού δικτύου για το πρόβλημα της ανάλυσης συναισθήματος. Το δίκτυο καλείται recursive neural tensor network - RNTN και εκπαιδεύεται σε προτάσεις με ετικέτες συναισθήματος, για τις οποίες επίσης δίνονται τα συντακτικά δέντρα και ετικέτες συναισθήματος σε κάθε κόμβο του δέντρου. Το δίκτυο αυτό εξαιτίας της δομής του αλλά και της πληροφορίας για την συντακτική σύνθεση των λέξεων, πάνω στην οποία εκπαιδεύεται, έχει τη δυνατότητα να μαθαίνει κανόνες για την σύνθεση των word vectors και να αποδίδει συναισθηματικό σκορ στους συνδυασμούς τους. Σε νέες προτάσεις, κατασκευάζει το συντακτικό δέντρο βάσει των κανόνων που έχει μάθει, και αποδίδει συναίσθημα σε κάθε κόμβο για να προκύψει τελικά η συναισθηματική πολικότητα ολόκληρης της πρότασης.

Στο [24] εισάγεται ένα νέο σύνολο δεδομένων που βασίζεται στο κλασσικό movie review dataset, αλλά περιέχει τα συντακτικά δέντρα των κριτικών ταινιών και συναισθηματικές πολικότητες για τους κόμβους. Σημείωνεται accuracy 86% στο σύνολο αυτό, που καλείται Stanford Sentiment Treebank, με το μοντέλο RNTN.

Το μοντέλο αυτό λοιπόν, έχει τα χαρακτηριστικά και την αρχιτεκτονική, ώστε να συνυπολογίζει όχι απλά τη σειρά, αλλά και τον τρόπο σύνθεσης των λέξεων για την απόδοση νοήματος στο σύνολό τους. Χωρίς πρότερη γνώση, το μοντέλο μαθαίνει το ρόλο λέξεων όπως but, not και very στη συναισθηματική πολικότητα των φράσεων που σχηματίζουν με άλλες λέξεις. Στο σχήμα 5.2 δίνεται ενδεικτικά το συντακτικό δέντρο που κατασκευάζει το μοντέλο για μία πρόταση του συνόλου δεδομένων και τα επιμέρους συναισθηματικά σκορ σε κλίμακα 5 σημείων (--,-,0,+,++).

Συμπερασματικά, μπορούμε να πούμε ότι ο αναγκαίος προσανατολισμός της έρευνας, στο πεδίο της ανάλυσης συναισθήματος, αλλά και γενικά στο πεδίο της επεξεργασίας φυσικού λόγου, τη δεδομένη στιγμή, είναι η υλοποίηση ευφυών compositional μοντέλων, που συνθέτουν τις διανυσματικές αναπαραστάσεις λέξεων με τρόπο που να αποδίδεται το πραγματικό νόημα στο σύνολό τους, είτε αυτό είναι μία φράση, είτε μία πρόταση, είτε ένα

ολόκληρο κείμενο. Φυσικά, το πρόβλημα είναι δύσκολο, με επεκτάσεις στην τεχνητή νοημοσύνη, τη γλωσσολογία (linguistics) και τη γνωσιακή επιστήμη (cognitive science). Οι διανυσματικές αναπαραστάσεις λέξεων ωστόσο, αποδεικνύουν ότι υπολογιστικά μοντέλα μπορούν να συλλάβουν το πολυεπίπεδο νόημα, την αμφισημία και την πολυπλοκότητα που χαρακτηρίζει γενικότερα τον φυσικό λόγο.



Σχήμα 5.2

Βιβλιογραφία

[1] Marco Baroni, Georgiana Dinu and Germán Kruszewski, *Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors,* ACL (1), 2014.

[2] Christopher M. Bishop, *Pattern Recognition and Machine Learning,* Springer, 2006.

[3] Andrea Esuli and Fabrizio Sebastiani, *SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining*, Proceedings of LREC. Vol. 6. 2006.

[4] Alec Go, Richa Bhayani and Lei Huang, *Twitter Sentiment Classification using Distant Supervision*, CS224N Project Report, Stanford, 2009.

[5] Yoav Goldberg and Omer Levy, *word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method,* arXiv preprint arXiv:1402.3722, 2014.

[6] Simon S. Haykin, *Neural Networks and Learning Machines,* 3rd edition, Upper Saddle River, NJ, USA:: Pearson, 2009.

[7] Ozan İrsoy and Claire Cardie, *Opinion Mining with Deep Recurrent Neural Networks*, Empirical Methods in Natural Language Processing, 2014.

[8] Yoon Kim, *Convolutional Neural Networks for Sentence Classification*, arXiv preprint arXiv: 1408.5882, 2014.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, *Imagenet Classification with Deep Convolutional Neural Networks*, Advances in Neural Information Processing Systems, 2012.

[10] Quoc Le and Tomas Mikolov, *Distributed Representations of Sentences and Documents*, International Conference on Machine Learning, 2014.

[11] Omer Levy and Yoav Goldberg, *Neural Word Embedding as Implicit Matrix Factorization,* Advances in Neural Information Processing Systems, 2014.

[12] Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean, *Efficient Estimation of Word Representations in Vector Space*, arXiv preprint arXiv: 1301.3781, 2013.

[13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean, *Distributed Representations of Words and Phrases and their Compositionality*, Advances in Neural Information Processing Systems, 2013.

[14] Tomas Mikolov, Wen-tau Yih and Geoffrey Zweig, *Linguistic Regularities in Continuous Space Word Representations*, HLT-NAACL. Vol. 13. 2013.

[15] Jeff Mitchell and Mirella Lapata, *Composition in Distributional Models of Semantics*, Cognitive science 34.8: 1388-1429, 2010.

[16] Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani and Veselin Stoyanov, *SemEval-2016 Task 4: Sentiment Analysis in Twitter*, Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), San Diego, US. 2016.

[17] Elisavet Palogiannidi, Athanasia Kolovou, Fenia Christopoulou, Filippos Kokkinos, Elias Iosif, Nikolaos Malandrakis, Harris Papageorgiou, Shrikanth Narayanan and Alexandros Potamianos, *Tweester at SemEval-2016 Task 4: Sentiment Analysis in Twitter using Semantic-Affective Model Adaptation*, Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), San Diego, US. 2016.

[18] Bo Pang and Lillian Lee, *Opinion Mining and Sentiment Analysis*, Foundations and Trends in Information Retrieval 2(1-2), pp 1-135, 2008.

[19] Bo Pang, Lillian Lee and Shivakumar Vaithyanathan, *Thumbs Up? Sentiment Classification using Machine Learning Techniques*, Proceedings of the ACL-02 conference on Empirical Methods in Natural Language Processing-Volume 10. Association for Computational Linguistics, 2002.

[20] Jeffrey Pennington, Richard Socher and Christopher D. Manning, *GloVe: Global Vectors for Word Representation,* Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP). Vol. 14, 2014.

[21] Xin Rong, *word2vec Parameter Learning Explained*, arXiv preprint arXiv: 1411.2738, 2014.

[22] Hassan Saif, Miriam Fernandez, Yulan He and Harith Alani, *Evaluation Datasets for Twitter Sentiment Analysis: A survey and a new dataset, the STS-Gold*, 2013.

[23] Cícero Nogueira Dos Santos and Maíra Gatti, *Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts*, COLING. 2014.

[24] Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng and Christopher Potts, *Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank*, Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP). Vol. 1631, 2013.

[25] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu and Bing Qin, *Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification*, ACL (1). 2014.

[26] Sergios Theodoridis and Konstantinos Koutroumbas, *Pattern Recognition*, 4th edition, Academic Press, 2009.

[27] Peter D. Turney, *Thumbs up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*, Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002.

[28] Sida Wang and Christopher D. Manning, *Baselines and Bigrams: Simple, Good Sentiment and Topic Classification*, Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2. Association for Computational Linguistics, 2012.