ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Εφαρμογή μεθόδων Βαθιάς Μάθησης για την αναγνωριση συναισθήματος κειμένου

Του φοιτητή

Κερμιζίδη Ραφαήλ
Καθηγητής

Αρ. Μητρώου: 123838

Επιβλέπων καθηγητής

Διαμαντάρας Κωνσταντίνος ,

Θεσσαλονίκη 2017

Περίληψη

Το μοντέλο bag-of-words είναι μια απλουστευμένη αναπαράσταση που χρησιμοποιείται στην επεξεργασία της φυσικής γλώσσας και στην ανάκτηση πληροφοριών (IR). Σε αυτό το μοντέλο, ένα κείμενο (όπως μια πρόταση ή ένα έγγραφο) αναπαρίσταται ως η τσάντα (multiset) των λέξεων της, αγνοώντας τη γραμματική και ακόμη και την τάξη των λέξεων, διατηρώντας όμως την πολλαπλότητα. Το μοντέλο bag-of-words έχει επίσης χρησιμοποιηθεί για την όραση του υπολογιστή. Το μοντέλο bag-of-words χρησιμοποιείται συνήθως σε μεθόδους ταξινόμησης εγγράφων όπου η συχνότητα εμφάνισης κάθε λέξης χρησιμοποιείται ως χαρακτηριστικό γνώρισμα για την κατάρτιση ενός ταξινομητή. Στην πτυχιακή εργασία θα παρουσιαστούν η συγκεκριμένη μέθοδος για την αναγνώριση συναισθήματος κειμένου(θετικό / αρνητικό συναίσθημα) χρησιμοποιώντας ένα σύνολο από βιβλιοθήκες και κωδικοποιώντας το σε γλώσσα python χρησιμοποιώντας την βιβλιοθήκη tensorflow . Θα χρησιμοποιηθούν βαθιά νευρωνικά δίκτυα(Convolutional neural network) για την ανάλυση κειμένου. Θα αναφερθούμε σε έννοιες για να κατανοήσουμε καλύτερα τι είναι ένα νευρωνικό δίκτυο , γιατί χρειαζόμαστε νευρωνικά δίκτυα και συγκεκριμένα γιατί να μας κινεί το ενδιαφέρον η αναγνώριση συναισθήματος καθώς επίσης γιατί είναι χρήσιμος ο τομέας των νευρωνικών δικτύων.Τέλος θα κάνουμε λεπτομερής αναφορά σε αλγόριθμο αλλά και στα αποτελέσματα που εξάγαμε.

Περιεχόμενα

Εισαγωγή

Η αύξηση των μέσων κοινωνικής δικτύωσης, όπως τα ιστολόγια (blogs) και τα κοινωνικά δίκτυα (social networks) έχει στρέψει το ερευνητικό ενδιαφέρον στην ανάλυση συναισθήματος (sentiment analysis). Ο όρος ανάλυση συναισθήματος αναφέρεται στον αυτόματο εντοπισμό και εξαγωγή απόψεων, συναισθημάτων και διαθέσεων από έγγραφα κειμένου. Ο βασικός στόχος της ανάλυσης συναισθήματος είναι ο χαρακτηρισμός της πολικότητας ενός συγκεκριμένου κειμένου – αν η γνώμη που εκφράζεται σε αυτό ερμηνεύεται ως θετική, η αρνητική.Έχει αποδειχθεί πως οι παραδοσιακές προσεγγίσεις ταξινόμησης κειμένου (text classification) μπορεί να είναι αρκετά αποτελεσματικές όταν εφαρμόζονται στο πρόβλημα της ανάλυσης συναισθήματος. Μοντέλα όπως Naïve Bayes (NB), Maximum Entropy (ME) , μέθοδος Word2vec, Support Vector Machines (SVM) μπορούν να προσδιορίσουν το συναίσθημα των κειμένων με υψηλή ακρίβεια.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: sentiment , analysis , python , word2vec , network, neural, statistics

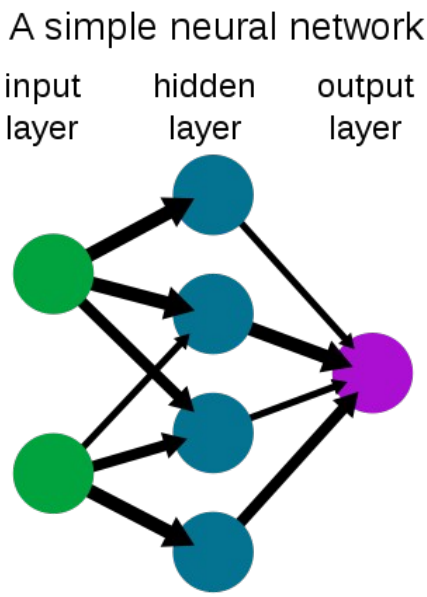
Νευρωνικά Δίκτυα

Εισαγωγή

Νευρωνικό δίκτυο ονομάζεται ένα κύκλωμα διασυνδεδεμένων νευρώνων . Στην περίπτωση βιολογικών νευρώνων, πρόκειται για ένα τμήμα νευρικού ιστού. Στην περίπτωση τεχνητών νευρώνων πρόκειτα για ένα αφηρημένο αλγοριθμικό κατασκεύασμα το οποίο εμπίπτει στον τομέα της υπολογιστικής νοημοσύνης, όταν στόχος του νευρωνικού δικτύου είναι η επίλυση κάποιου υπολογιστικούπροβλήματος, ή της υπολογιστικής νευροεπιστήμης, όταν στόχος είναι η υπολογιστική προσομοίωση της λειτουργίας των βιολογικών νευρωνικών δικτύων με βάση κάποιο μαθηματικό μοντέλο τους.Στην επιστήμη των υπολογιστών είναι ένα δίκτυο από απλούς υπολογιστικούς κόμβους (νευρώνες, νευρώνια), διασυνδεδεμένους μεταξύ

τους. Είναι εμπνευσμένο από το Κεντρικό Νευρικό Σύστημα (ΚΝΣ), το οποίο προσπαθεί να προσομοιώσει.Οι νευρώνες είναι τα δομικά στοιχεία του δικτύου. Κάθε τέτοιος κόμβος δέχεται ένα σύνολο αριθμητικών εισόδων από διαφορετικές πηγές (είτε από άλλους νευρώνες, είτε από το περιβάλλον), επιτελεί έναν υπολογισμό με βάση αυτές τις εισόδους και παράγει μία έξοδο. Η εν λόγω έξοδος είτε κατευθύνεται στο περιβάλλον, είτε τροφοδοτείται ως είσοδος σε άλλους νευρώνες του δικτύου. Υπάρχουν τρεις τύποι νευρώνων: οι νευρώνες εισόδου, οι νευρώνες εξόδου και οι υπολογιστικοί νευρώνες ή κρυμμένοι νευρώνες. Οι νευρώνες εισόδου δεν επιτελούν κανέναν υπολογισμό, μεσολαβούν απλώς ανάμεσα στις περιβαλλοντικές εισόδους του δικτύου και στους υπολογιστικούς νευρώνες. Οι νευρώνες εξόδου διοχετεύουν στο περιβάλλον τις τελικές αριθμητικές εξόδους του δικτύου. Οι υπολογιστικοί νευρώνες πολλαπλασιάζουν κάθε είσοδό τους με το αντίστοιχο συναπτικό βάρος και υπολογίζουν το ολικό άθροισμα των γινομένων. Το άθροισμα αυτό τροφοδοτείται ως όρισμα στη συνάρτηση ενεργοποίησης, την οποία υλοποιεί εσωτερικά κάθε κόμβος. Η τιμή που λαμβάνει η συνάρτηση για το εν λόγω όρισμα είναι και η έξοδος του νευρώνα για τις τρέχουσες εισόδους και βάρη.

Απλό νευρωνικό δίκτυο 1.1



1.1 Παράδειγμα τεχνικού νευρωνικού δικτύου

$$y_k = \phi\left(\sum_{i=0}^{N} x_{ki} w_{ki}\right)$$

Στον k-οστό νευρώνα υπάρχει ένα συναπτικό βάρος με ιδιαίτερη σημασία, το οποίοκαλείται πόλωση ή κατώφλι (bias, threshold). Η τιμή της εισόδου του είναι *πάντα* η μονάδα. Εάν το συνολικό άθροισμα από τις υπόλοιπες εισόδους του νευρώνα είναι μεγαλύτερο από την τιμή αυτή, τότε ο νευρώνας ενεργοποιείται. Εάν είναι μικρότερο, τότε ο νευρώνας παραμένει ανενεργός. Η ιδέα προέκυψε από τα βιολογικά νευρικά κύτταρα.

Συναρτήσεις ενεργοποίησης 1.2

Η συνάρτηση ενεργοποίησης μπορεί να είναι βηματική (step transfer function), γραμμική (linear transfer function), μη γραμμική (non-linear transfer function), στοχαστική (stochastic transfer function).

Βηματική:

$$\phi(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Βασικό μειονέκτημα ότι η παράγωγός της απειρίζεται

Γραμμική

$$\phi(x) = x$$

Μη γραμμική

Η μη γραμμική συνάρτηση ενεργοποίησης που χρησιμοποιείται συνήθως στα νευρωνικά δίκτυα καλείται σιγμοειδής συνάρτηση. Οι τυπικές σιγμοειδείς είναι δύο:

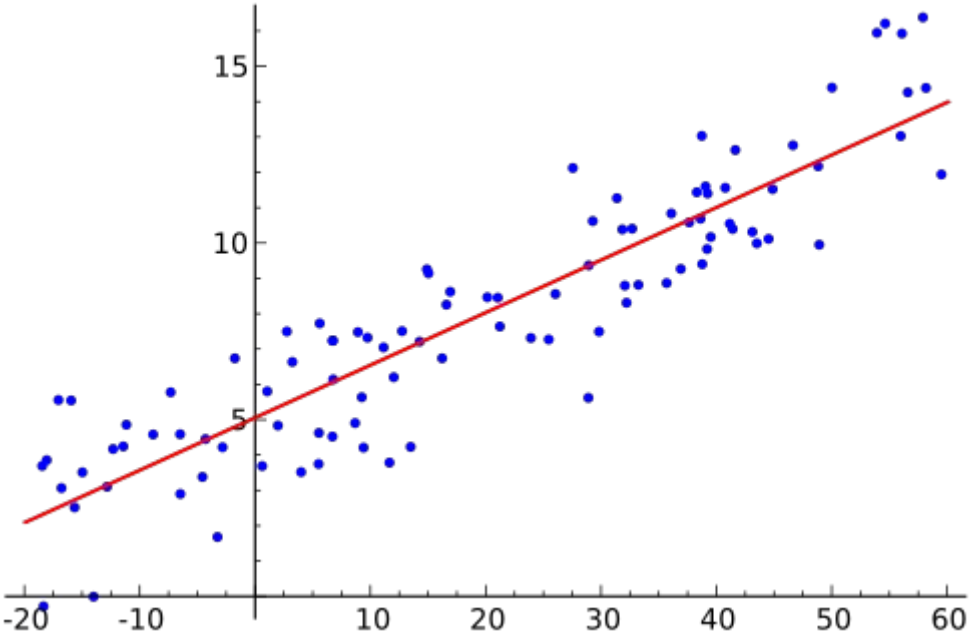
$$\phi(x) = \frac{1}{1 + e^{x}}$$

Λογιστική σιγμοειδής

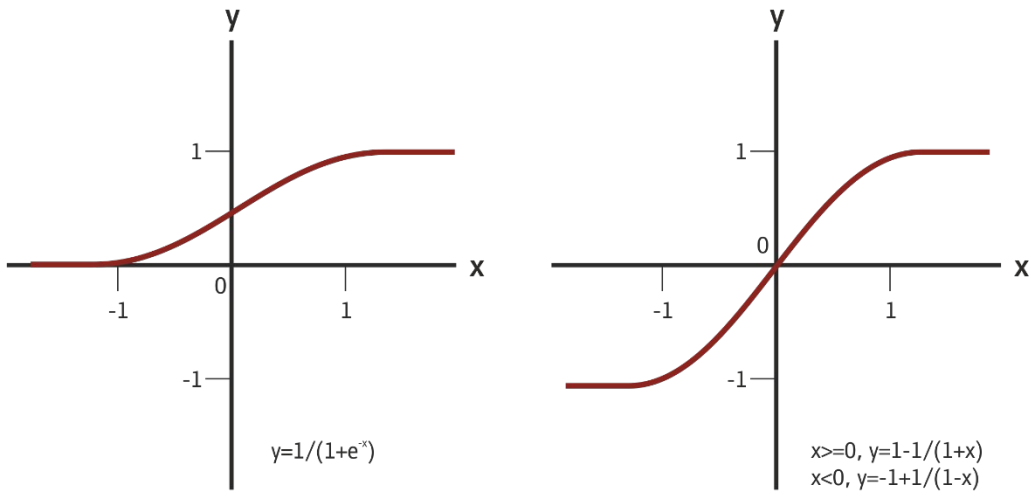
$$\phi(x) = \tanh x$$

Υπερβολική εφαπτομένη

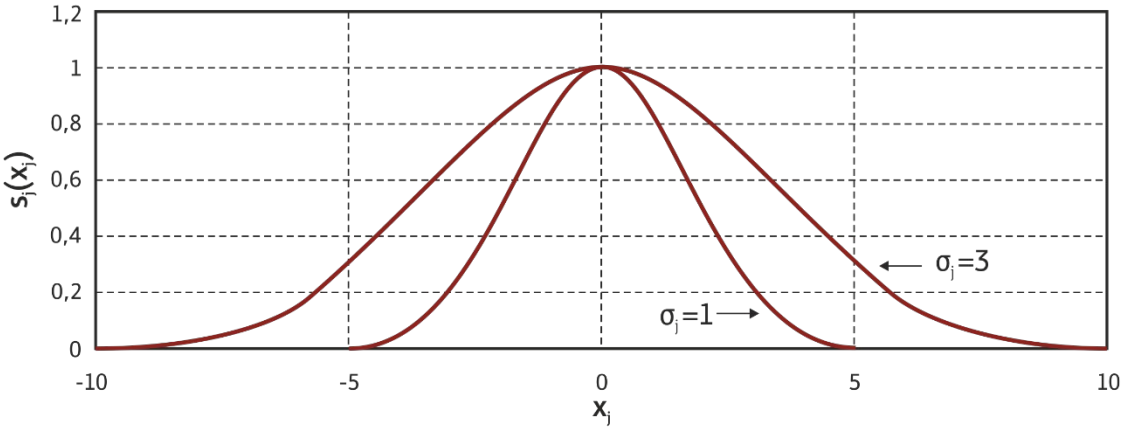
1.2 γραμμική παλινδρόμηση



1.2 συγμοειδής συναρτήσεις , γκαουσιανές συναρτήσεις



Σιγμοειδείς Συναρτήσεις



Γκαουσιανές Συναρτήσεις

Εφαρμογές 1.3

Τα τελευταία χρόνια έχει υπάρξει μία έκρηξη ενδιαφέροντος για τα νευρωνικά δίκτυα καθώς εφαρμόζονται με μεγάλη επιτυχία σε ένα ασυνήθιστα μεγάλο φάσμα τομέων της επιστήμης και της τεχνολογίας, όπως τα **χρηματοοικονομικά**, η **ιατρική**, η **επιστήμη μηχανικού**, η **γεωλογία**, η **φυσική**, η **ρομποτική**, η **επεξεργασία σήματος** κτλ. Στην πραγματικότητα, τα νευρωνικά δίκτυα εισάγονται οπουδήποτε τίθεται θέμα πρόβλεψης, ταξινόμησης ή ελέγχου. Η σαρωτική αυτή επιτυχία, μπορεί να αποδοθεί σε δύο βασικά στοιχεία: την ισχύ και την ευχρηστία.

- **Ισχύς**: Τα νευρωνικά δίκτυα είναι πολύ εξελιγμένες τεχνικές μη γραμμικής μοντελοποίησης, ικανές να μοντελοποιήσουν εξαιρετικά πολύπλοκες λειτουργίες. Η γραμμική μοντελοποίηση υπήρξε ευρέως διαδεδομένη για πολύ καιρό, δεδομένου ότι στα γραμμικά μοντέλα εφαρμόζονται πολύ γνωστές στρατηγικές βελτιστοποίησης. Στις συνήθεις, όμως, περιπτώσεις όπου η γραμμική προσέγγιση δεν ήταν έγκυρη, τα μοντέλα αυτά αποτύγχαναν αναλόγως. Τα νευρωνικά δίκτυα βέβαια, αν και επιτρέπουν τη μη γραμμικότητα μέσω χρήσης μη γραμμικών συναρτήσεων ενεργοποίησης, μεταθέτουν με τη σειρά τους το πρόβλημα στο ζήτημα της διάστασης (του πλήθους των διαφορετικών εισόδων και εξόδων), το οποίο αποτελεί αγκάθι στις προσπάθειες μοντελοποίησης μη γραμμικών συναρτήσεων με μεγάλο αριθμό μεταβλητών.

- **Ευχρηστία**: Τα νευρωνικά δίκτυα εκπαιδεύονται με παραδείγματα. Ο χρήστης συγκεντρώνει αντιπροσωπευτικά δεδομένα και στη συνέχεια, καθώς τα τροφοδοτεί συστηματικά στο δίκτυο μέσω των κατάλληλων αλγορίθμων εκπαίδευσης, το δίκτυο «αντιλαμβάνεται» αυτομάτως τη δομή των δεδομένων και η «γνώση» αυτή εκφράζεται ως κατάλληλες επιλογές συναπτικών βαρών. Επομένως το τελικό αποτέλεσμα της εκπαίδευσης με ένα συγκεκριμένο σύνολο παραδειγμάτων είναι ο προσδιορισμός των κατάλληλων βαρών του δικτύου. Ο χρήστης χρειάζεται να έχει κάποιες ουσιώδεις γνώσεις σχετικά με τον τρόπο επιλογής και προετοιμασίας των δεδομένων, τον τρόπο εκλογής του κατάλληλου νευρωνικού δικτύου και στο πως θα ερμηνευτούν τα αποτελέσματα. Παρά ταύτα, το επίπεδο των γνώσεων του χρήστη που απαιτούνται για μια επιτυχημένη εφαρμογή των νευρωνικών δικτύων, είναι πολύ χαμηλότερο συγκριτικά με κάποια περίπτωση που θα χρησιμοποιούνταν ορισμένες πιο παραδοσιακές, μη γραμμικές στατιστικές μέθοδοι.

Τα νευρωνικά δίκτυα είναι εφαρμόσιμα σχεδόν σε κάθε κατάσταση στην οποία ισχύει μια σχέση μεταξύ μεταβλητών πρόβλεψης (ανεξάρτητες, εισροές) και προβλεπόμενες μεταβλητές (εξαρτημένες, εκροές), ακόμα και όταν αυτή η σχέση είναι πολύ περίπλοκη για να αποδοθεί με τους συνηθισμένους όρους της «συσχέτισης» ή των «διαφόρων ομάδων». Ενδεικτικά αντιπροσωπευτικά παραδείγματα προβλημάτων στα οποία η ανάλυση των νευρωνικών δικτύων έχει εφαρμοστεί με επιτυχία είναι τα εξής:

- Ιατρική διάγνωση: Ένα ευρύ φάσμα ιατρικά συσχετιζόμενων ενδείξεων, όπως ο συνδυασμός της καρδιακής συχνότητας, τα επίπεδα των διαφόρων ου-

σιών στο αίμα, ο ρυθμός της αναπνοής μπορούν να παρακολουθηθούν. Η εκδήλωση μιας συγκεκριμένης ιατρικής κατάστασης, γίνεται να συσχετιστεί με ένα πολύπλοκο συνδυασμό μεταβολών σε ένα υποσύνολο μεταβλητών που παρακολουθούνται. Τα νευρωνικά δίκτυα έχουν χρησιμοποιηθεί για την αναγνώριση αυτού του προτύπου πρόβλεψης, ώστε να χορηγηθεί η κατάλληλη θεραπεία.

- Χρηματιστηριακές προβλέψεις: Οι διακυμάνσεις των τιμών των μετοχών και των χρηματιστηριακών δεικτών είναι ακόμα ένα παράδειγμα ενός πολύπλοκου, πολυδιάστατου, αλλά και σε ορισμένες περιπτώσεις εν μέρει ντετερμινιστικού φαινομένου. Τα νευρωνικά δίκτυα χρησιμοποιούνται από πολλούς τεχνικούς αναλυτές, ώστε να κάνουν προβλέψεις σχετικά με τις τιμές των μετοχών, βασιζόμενοι σε ένα μεγάλο αριθμό παραγόντων, όπως δηλαδή, τις προηγούμενες επιδόσεις άλλων αποθεμάτων και διαφόρων οικονομικών δεικτών.

- Πιστωτική ανάθεση: Μια ποικιλία από κομμάτια πληροφοριών, τα οποία είναι συνήθως γνωστά για ένα απαιτούμενο δάνειο. Για παράδειγμα, η ηλικία του αιτούντος, η εκπαίδευση, το επάγγελμα και πολλά άλλα στοιχεία που μπορεί να είναι διαθέσιμα. Μετά την εκπαίδευση ενός νευρωνικού δικτύου σε ιστορικά δεδομένα η ανάλυση μπορεί να εκτοπίσει τα πιο κατάλληλα και σχετικά χαρακτηριστικά και να τα χρησιμοποιήσει για την ταξινόμηση των αιτούντων ως χαμηλού ή υψηλού κινδύνου.

- Παρακολούθηση της κατάστασης των μηχανημάτων: Τα νευρωνικά δίκτυα μπορούν να συμβάλλουν στη μείωση του κόστους με την εξασφάλιση της πρόσθετης εμπειρογνωμοσύνης για τον προγραμματισμό προληπτικής συντήρησης των μηχανημάτων. Ένα νευρωνικό δίκτυο, λοιπόν, μπορεί να εκπαιδευτεί με τέτοιο τρόπο, ώστε να διακρίνει από τους ήχους τους οποίους παράγει μια μηχανή είτε αν εκτελεί κανονικά τις λειτουργίες της, είτε βρίσκεται στα πρόθυρα εμφάνισης οποιασδήποτε δυσλειτουργίας. Μετά από αυτήν την περίοδο εκπαιδευτικής κατάρτισης, η εμπειρία του ίδιου δικτύου είναι δυνατό να χρησιμοποιηθεί με σκοπό την προειδοποίηση ενός τεχνικού για κάποια επικείμενη βλάβη προτού συμβεί και ενδεχομένως προκαλέσει πολυδάπανες και απρόβλεπτες χρονικές καθυστερήσεις.

- Συστήματα διαχείρισης κινητήρα: Τα νευρωνικά δίκτυα έχουν χρησιμοποιηθεί για την ανάλυση των εισροών που δέχονται οι αισθητήρες ενός κινητήρα. Το νευρωνικό δίκτυο ελέγχει μια ποικιλία παραμέτρων με τις οποίες λειτουργεί ο κινητήρας, προκειμένου να επιτευχθεί ένας συγκεκριμένος στόχος. Για παράδειγμα, το δίκτυο αυτό επιχειρεί την ελαχιστοποίηση της κατανάλωσης των καυσίμων.

Για να είναι ευφικτή η ανάλυση και η εξαγωγή αποτελεσμάτων θα πρέπει να συγκεντρώσουμε ένα σύνολο από δεδομένα και να σκεφτούμε πως θα μπορέσουμε να δώσουμε σημασία σε αυτά.Στην συνέχεια θα εκπαιδεύσουμε το νευρωνικό δίκτυο και θα αξιολογήσουμε σε ποσοστό αν είναι αποτελεσματικό.

Εκπαίδευση νευρωνικών δικτύων

Εκπαίδευση

Μια από τις πιο βασικές ιδιότητες των Νευρωνικών Δικτύων είναι η ικανότητά τους για εκπαίδευση. Η εκπαίδευση αυτή επιτυγχάνεται μέσω της ανταλλαγής τιμών και βαρών, που αποσκοπεί στη βαθμιαία σύλληψη της πληροφορίας η οποία στη συνέχεια θα είναι διαθέσιμη προς ανάκτηση. Υπάρχουν, βέβαια, πολλοί αλγόριθμοι που η εφαρμογή τους έχει στόχο την προσαρμογή των τιμών των βαρών ενός Τεχνητού Νευρωνικού Δικτύου. Όλες οι μέθοδοι μάθησης μπορούν να καταταχτούν σε δύο κατηγορίες : τη **μάθηση με επίβλεψη**(supervised learning) και τη **μάθηση χωρίς επίβλεψη** (unsupervised learning).

Μάθηση με επίβλεψη: Η μάθηση αυτή είναι μια διαδικασία η οποία συνδυάζει έναν εξωτερικό εκπαιδευτή και τη συνολική ή γενικευμένη πληροφορία. Κάποιες από τις μεθόδους οι οποίες συγκαταλέγονται σε αυτή την κατηγορία είναι η μάθηση με διόρθωση σφάλματος, η στοχαστική μάθηση. Παραδείγματα τα οποία αντιπροσωπεύουν την μάθηση με επίβλεψη συμπεριλαμβάνουν αποφάσεις για το πότε θα πρέπει να σταματήσει η διαδικασία εκπαίδευσης, αποφάσεις αναφορικά με τη συχνότητα παρουσίασης στο δίκτυο τα πρότυπα εκπαίδευσης και η παρουσίαση προόδου του δικτύου. Η μάθηση με επίβλεψη χωρίζεται σε δύο ακόμα κατηγορίες: στη **δομική** (structural) και στην **προσωρινή** (temporal) εκμάθηση. Οι αλγόριθμοι οι οποίοι βρίσκονται στην πρώτη κατηγορία, χρησιμοποιούνται για την εύρεση της βέλτιστης σχέσης μεταξύ εισόδων και εξόδων για κάθε ξεχωριστό ζευγάρι προτύπων. Παραδείγματα της δομικής εκμάθησης αποτελούν η αναγνώριση και η κατηγοριοποίηση προτύπων, ενώ παραδείγματα της προσωρινής εκμάθησης η πρόβλεψη και ο έλεγχος.

Μάθηση χωρίς επίβλεψη: Οι αλγόριθμοι της εν λόγω μάθησης αναφέρονται ως αυτό-οργανώμενοι (self-organized) και είναι διαδικασίες οι οποίες δεν απαιτούν να είναι παρών ένας «εξωτερικός» δάσκαλος ή επιβλέπων. Βασίζονται, μάλιστα, μόνο σε τοπική πληροφορία καθ' όλη τη διάρκεια της εκπαίδευσης του Τεχνητού Νευρωνικού Δικτύου. Οι συγκεκριμένοι αλγόριθμοι οργανώνουν τα δεδομένα και ανακαλύπτουν τις σημαντικές συλλογικές ιδιότητες. Για παράδειγμα, αλγόριθμοι εκπαίδευσης χωρίς επίβλεψη είναι ο αλγόριθμος Hebbian, ο διαφορικός αλγόριθμος Hebbian και ο Min-Max αλγόριθμος.

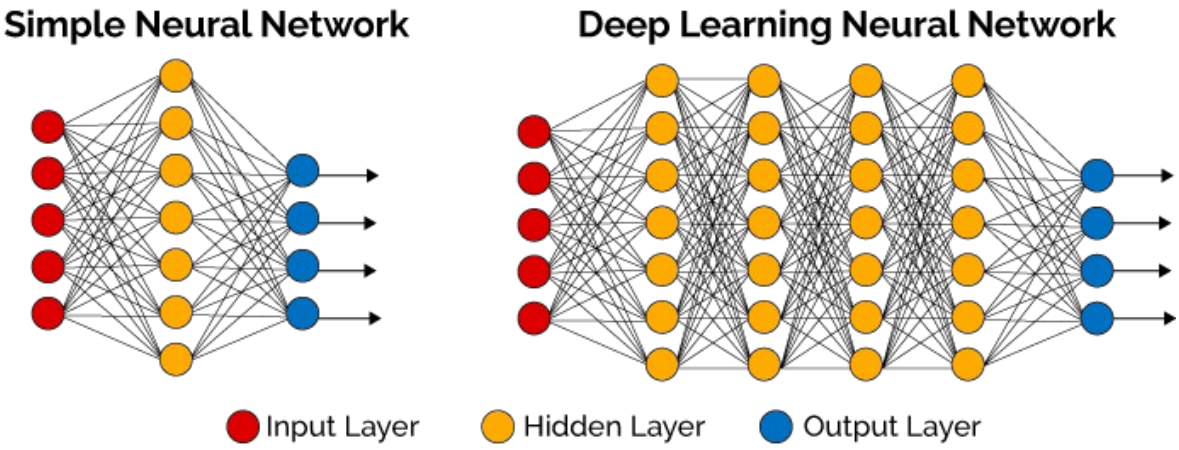
Βαθιά νευρωνικά δίκτυα για αναγνώριση προτύπων

Τα τελευταία χρόνια υπάρχει ραγδαία ανάπτυξη του Διαδικτύου. Όλο και περισσότεροι άνθρωποι ασχολούνται με την αναζήτηση περιεχομένου στο διαδίκτυο, την αγορά και πώληση προϊόντων καθώς επίσης και με την ανταλλαγή απόψεων για προϊόντα και υπηρεσίες που αγοράζουν. Οι χρήστες του Διαδικτύου δεν είναι πλέον παθητικοί αποδέκτες πληροφοριών. Μεγάλο ποσοστό χρηστών χρησιμοποιεί κοινωνικά δίκτυα και διάφορα forum για να ανταλλάξει απόψεις και να αξιολογήσει διάφορα προϊόντα και υπηρεσίες. Καθώς ολοένα και περισσότεροι χρήστες

του Διαδικτύου αναρτούν κριτικές με τα προϊόντα ή τις υπηρεσίες που χρησιμοποιούν σε κοινωνικά δίκτυα, forum και ηλεκτρονικά καταστήματα πλέον υπάρχει πολύ μεγάλος όγκος πληροφοριών σε ηλεκτρονική μορφή σχετικά με τα συναισθήματα των ανθρώπων. Η ανάγκη ανάλυσης και κατά συνέπεια αξιοποίησης του όγκου της πληροφορίας αυτής οδήγησε στην εμφάνιση της Ανάλυσης Συναισθήματος (Sentiment Analysis).

Ενδεικτική εικόνα ανάλυσης συναισθημάτων

1.3



1.4 Απλό νευρωνικό δίκτυο – Συνελικτικό νευρωνικό δίκτυο

Αναγνώριση προτύπων

Η αναγνώριση προτύπων (Pattern Recognition) είναι ένα επιστημονικό πεδίο με στόχο την ανάπτυξη αλγορίθμων για την αυτοματοποιημένη απόδοση κάποιας τιμής ή διακριτικού στοιχείου σε εισαγόμενα δεδομένα, συνήθως κωδικοποιημένα ως αλληλουχίες αριθμών. Κατ' αυτόν τον τρόπο, ενδεικτικά, τα δεδομένα αυτόματα ταξινομούνται σε κατηγορίες ή διαχωρίζονται σε ομάδες με βάση κάποια κριτήρια, ακόμα και υπό την παρουσία θορύβου ο οποίος δυσκολεύει την αναγνώριση, ωθώντας συνήθως τα δεδομένα να μοιάζουν περισσότερο τυχαία απ' όσο πραγματικά είναι. Το ερευνητικό ενδιαφέρον για την αναγνώριση προτύπων έχει τις ρίζες του στη δεκαετία του 1960, κατά την πρώτη περίοδο ανάπτυξης της πληροφορικής και, ειδικότερα, της τεχνητής νοημοσύνης.

Οι άνθρωποι και οι ευφυείς οργανισμοί έχουν την ικανότητα να ταυτοποιούν πραγματικά δεδομένα χρησιμοποιώντας τις αισθήσεις τους και την αντιληπτική τους ικανότητα προκειμένου να λάβουν τις κατάλληλες αποφάσεις ώστε να επιβιώσουν στο περιβάλλον τους. Μία μηχανή, όπως ένας ηλεκτρονικός υπολογιστής, πρέπει να εκπαιδευθεί κατάλληλα ώστε να αναγνωρίζει πρότυπα (patterns) και να τα κατηγοριοποιεί αυτόματα σε κατηγορίες. Ανάλογα με την εφαρμογή, γίνεται κατάταξη των αντικειμένων σε κλάσεις με τη βοήθεια αλγορίθμων ταξινόμησης.

Η μηχανική μάθηση αναφέρεται στον σχεδιασμό αλγορίθμων για τη δημιουργία ενός αυτόματου συστήματος που θα αποκτά γνώση βασιζόμενο σε εμπειρικά δεδομένα. Η έννοια της επιβλεπόμενης μάθησης έχει ιδιαίτερη σημασία. Δεδομένης μίας υπάρχουσας συλλογής αντικειμένων για τα οποία είναι γνωστή κλάση (ή κατηγορία) στόχος είναι να βρεθεί μία συνάρτηση μεταβλητών που θα περιγράφει το μοντέλο της κλάσης. Η επιτυχία πρόβλεψης αξιολογείται με ένα νέο σύνολο δεδομένων.

Βασική ορολογία

- Σύνολο Εκπαίδευσης: Σύνολο εισαγόμενων δεδομένων στο οποίο έχουν εκχωρηθεί εκ των προτέρων ετικέτες.
- Στιγμιότυπο: Εισαγόμενο αντικείμενο στο οποίο αναμένουμε να αποδοθεί μία τιμή από το σύστημα.
- Γνώρισμα: Χαρακτηριστικό του στιγμιότυπου βάσει του οποίου θα γίνει η ταξινόμηση των νέων αντικειμένων.

Τυπική Διαδικασία

1. Καθορισμός τύπου δεδομένων.

2. Δημιουργία συνόλου εκπαίδευσης.

   Το σύνολο εκπαίδευσης πρέπει να είναι αντιπροσωπευτικό της πραγματικής φύσης του προβλήματος. Από μετρήσεις ή εξωτερική ανθρώπινη παρέμβαση αναθέτουμε στα δεδομένα του σώματος τις ετικέτες.

3. Επιλογή κατάλληλων γνωρισμάτων.

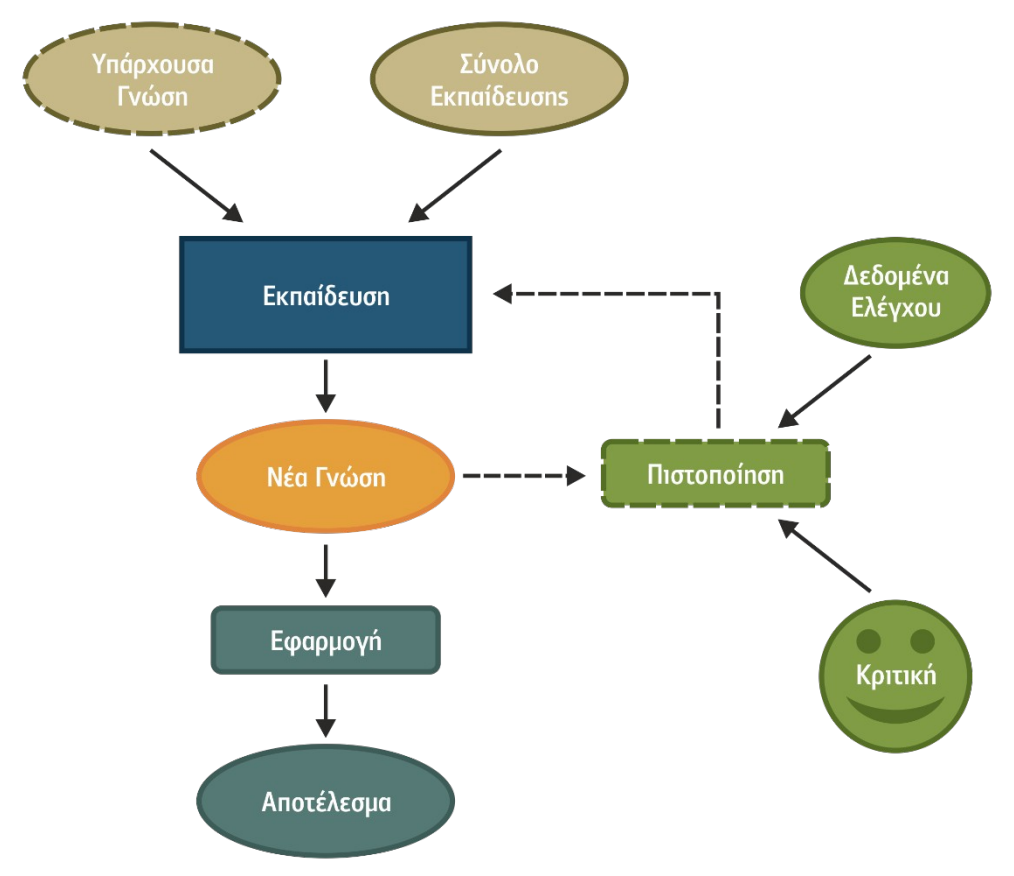
Μετατρέπουμε κάθε νέο αντικείμενο σε διάνυσμα γνωρισμάτων προκείμενου να ακολουθήσει η ταξινόμησή του. Είναι σημαντικό να επιλεγούν τα γνωρίσματα εκείνα που περιέχουν την απαραίτητη πληροφορία για την σωστή ανάθεση κλάσης ενώ παράλληλα ο αριθμός τους πρέπει να είναι διαχειρίσιμος υπολογιστικά. Επίσης πρέπει να γνωρίζουμε τον τύπο τιμής κάθε ιδιότητας (ονομαστικής (nominal), τακτικής (ordinal), αναλογικής (ratio)).

4. Επιλογή αλγορίθμου εκπαίδευσης

Υπάρχει πληθώρα αλγορίθμων και η επιλογή πρέπει να γίνει ανάλογα με το ποιος κρίνεται πιο αποδοτικός για τη συγκεκριμένη κατηγορία προβλημάτων.

5. Αξιολόγηση της διαδικασίας.

Μετά την ολοκλήρωση της εκπαίδευσης η ακρίβεια της παραγόμενης σχέσης θα πρέπει να αξιολογηθεί με ένα σύνολο αξιολόγησης.



2.1 διάγραμμα ακολουθίας υλοποίησης

Τομείς Χρησιμότητας

Μάρκετινγκ

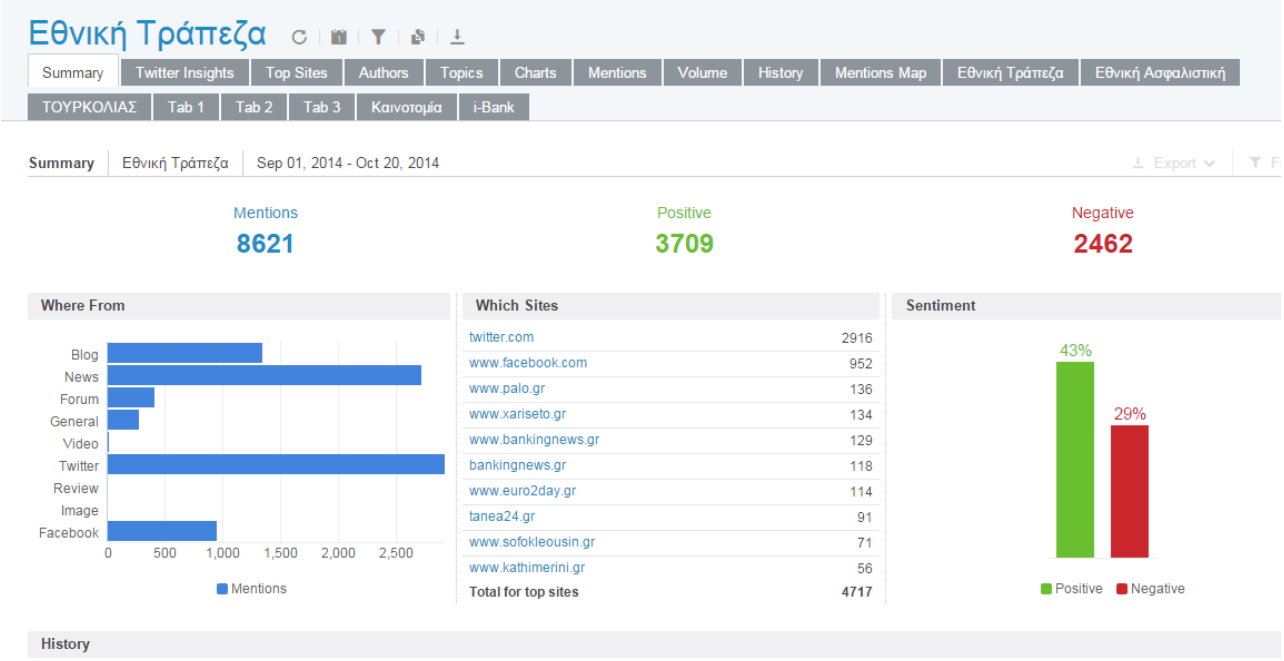
Οι ακαδημαϊκοί ερευνητές εντοπίζουν μεγάλο ενδιαφέρον στις τεχνικές προκλήσεις που παρουσιάζει η ανάλυση συναισθήματος

Έρευνα

Η ανάλυση συναισθήματος σε δεδομένα μεγάλου όγκου χρηστών του διαδικτύου κερδίζει ολοένα και μεγαλύτερο έδαφος στο ακαδημαϊκό χώρο καθώς και στον επιχειρηματικό χώρο.Οι επιχειρηματίες εντοπίζουν το ενδιαφέρον τους στις πολλά υποσχόμενες προοπτικές της. Καινοτόμες επιχειρήσεις ασχολούνται με την εξόρυξη γνώσης μέσα από τις αξιολογήσεις χρηστών σε ηλεκτρονικά καταστήματα και κοινωνικά δίκτυα με βάση την ανάλυση συναισθήματος. Τέλος, υπάρχει μεγάλη επιρροή καταναλωτών και χρηστών του διαδικτύου από τις κριτικές που υπάρχουν στον ηλεκτρονικό κόσμο πριν λάβουν κάποια απόφαση για αγορά μιας υπηρεσίας ή ενός προϊόντος.

Χρηματιστήριο

Είναι ωφέλιμο να γνωρίζουμε και να μπορούμε να προβλέψουμε σε πολύ μικρό χρονικό διάστημα αν ένας χρηματιστής σχολιάσει μια πρόγνωση σχετικά με τις μελλοντικές τιμές (αν είναι αξιόλογο το σχόλιο του ή όχι) .Έτσι τα συμπεράσματα μας θα είναι πιο σίγουρα αν βέβαια και ο αλγόριθμος μας αξιολογεί σωστά τα αποτελέσματα.



2.2 Παράδειγμα αξιολόγησης θετικών και αρνητικών προτάσεων από διάφορες ιστοσελίδες

Λεκτική γραμματική

Ο ορισμός μιας γλώσσας προγραμματισμού συχνά περιλαμβάνει ένα σύνολο κανόνων που ορίζουν τον συντακτικό αναλυτή. Αυτοί οι κανόνες συνήθως είναι κανονικές εκφράσεις και περιγράφουν το σύνολο των επιτρεπτών ακολουθιών χαρακτήρων που μπορούν να χρησιμοποιηθούν για να σχηματίσουν μεμονωμένες λεκτικές μονάδες (tokens ή lexemes). Ένα πρόγραμμα συντακτικής ανάλυσης αναγνωρίζει συμβολοσειρές. Για κάθε συμβολοσειρά που εντοπίζει, το πρόγραμμα ακολουθεί κάποια ενέργεια.

Στις γλώσσες προγραμματισμού που χωρίζουν τμήματα κώδικα (**blocks**) με μονάδες (όπως το "{" και το "}"), και όχι με στοίχιση, οι χαρακτήρες κενού (white space) ορίζονται και αυτοί από μια κανονική έκφραση, επηρεάζοντας έτσι την αναγνώριση των άλλων μονάδων, χωρίς όμως το ίδιο το κενό να αναγνωρίζεται σαν μονάδα. Δηλαδή, σε αυτές τις γλώσσες, οι χαρακτήρες κενού δεν είναι σημαντικοί.

Λεκτική μονάδα

Μια **λεκτική μονάδα** (**token**) είναι μια ακολουθία από χαρακτήρες που έχουν καταχωρηθεί ανάλογα με τους κανόνες ως κάποιο σύμβολο (π.χ., IDENTIFIER, NUMBER, COMMA). Η διαδικασία σχηματισμού λεκτικών μονάδων από ένα ρεύμα εισόδου χαρακτήρων ονομάζεται **tokenization** και ο λεκτικός αναλυτής κατηγοριοποιεί τις μονάδες ανάλογα με τον τύπο συμβόλου τους. Μια μονάδα μπορεί να μοιάζει με οτιδήποτε μπορεί να χρησιμοποιηθεί για την επεξεργασία ενός ρεύματος εισόδου κειμένου ή ενός αρχείου κειμένου.Ένας λεκτικός αναλυτής γενικά δεν επεξεργάζεται συνδυασμούς λεκτικών μονάδων αλλά αυτό αποτελεί έργο του συντακτικού αναλυτή. Για παράδειγμα, ένας κλασικός λεκτικός αναλυτής αναγνωρίζει κάθε παρένθεση σαν ξεχωριστή λεκτική μονάδα, αλλά δεν ελέγχει αν κάθε παρένθεση που ανοίγει αντιστοιχεί σε μια παρένθεση που κλείνει.

Έστω η εξής έκφραση στη γλώσσα προγραμματισμού C:

```
Sum = 3 + 2;
```

Οι λεκτικές μονάδες εμφανίζονται στον εξής πίνακα:

Lexeme	Τύπος λεκτικής μονάδας
sum	Αναγνωριστικό (identifier)
=	Τελεστής ανάθεσης
3	Ακέραια τιμή
+	Τελεστής πρόσθεσης
2	Ακέραια τιμή
;	Τέλος εντολής

2.3 παράδειγμα αναφορικά

Tokenization ονομάζεται η διαδικασία με την οποία τεμαχίζεται μια συμβολοσειρά χαρακτήρων εισόδου σε ξεχωριστά μέρη, καθώς και η κατηγοριοποίηση αυτών των μερών. Οι λεκτικές μονάδες που προκύπτουν προωθούνται στη συνέχεια για περαιτέρω επεξεργασία. Η διαδικασία αυτή μπορεί να θεωρηθεί μέρος της συντακτικής ανάλυσης της εισόδου.

Έστω για παράδειγμα, η συμβολοσειρά:

```
The quick brown fox jumps over the lazy dog
```

Αν και ένας ομιλητής της Αγγλικής θα χώριζε την παραπάνω πρόταση με βάση τα κενά, αυτό δεν συμβαίνει γενικά κατά τη συντακτική ανάλυση. Η είσοδος αποτελείται από 43 χαρακτήρες και πρέπει ρητά να χωριστεί σε 9 λεκτικές μονάδες χρησιμοποιώντας σαν διαχωριστικό το κενό (π.χ. ίσο με τη συμβολοσειρά " " ή την κανονική έκφραση /\s{1}/).

Οι λεκτικές μονάδες θα μπορούσαν επίσης να αναπαρασταθούν ως XML:

```xml
<sentence>
  <word>The</word>
  <word>quick</word>
  <word>brown</word>
  <word>fox</word>
  <word>jumps</word>
  <word>over</word>
  <word>the</word>
  <word>lazy</word>
  <word>dog</word>
</sentence>
```

2.3.1 παράδειγμα σε xml

Μπορούμε να μελετήσουμε την εξέλιξη της ανάλυσης συναισθήματος από τα αναλυτικά διακριτικά (tokens) ή δομικά στοιχεία και τις πληροφορίες που σχετίζονται με αυτά τα διακριτά. Μπορούμε να κατηγοριοποιήσουμε τις υπάρχουσες προσεγγίσεις σε τέσσερις κατηγορίες. Η επισήμανση των λέξεων κλειδιών,η λεξιλογική συγγένεια, οι στατιστικές μέθοδοι και οι τεχνικές που βασίζονται στην έννοια.

Η επισήμανση των λέξεων: αν και μπορεί να χαρακτηριστεί ως η πιο αφελής προσέγγιση η οικονομία και η προσβασιμότητα του εντοπισμού των λέξεων κλειδιών την καθιστούν δημοφιλή. Αυτή η προσέγγιση κατατάσσει κείμενο ανάλογα με τις λέξεις κλειδιά: χαρούμενος, λυπημένος, φοβισμένος, βαρετά κτλ.

Η λεξιλογική συγγένεια: Αυτή η προσέγγιση δεν ανιχνεύει προφανείς λέξεις αλλά αποδίδει λέξεις με πιθανή συγγένεια με συγκεκριμένα συναισθήματα. Για παράδειγμα η λέξει «ατύχημα» παρουσιάζει κατά 75% μια αρνητική επιρροή. Αυτή η προσέγγιση ξεπερνά την προσέγγιση επισήμανση των λέξεων αλλά έχει και τα μειονεκτήματά της. Αρχικά αναιρεί εκφράσεις της μορφής «Απέφυγα ένα ατύχημα» καθώς θα την εμφανίσει κατά 75% αρνητική και δεν καταλαβαίνει γλωσσικές ιδιομορφίες της γλώσσας (I met my girlfriend by accident).

Οι στατιστικές μέθοδοι: Είναι σημασιολογικά αδύναμες που σημαίνει ότι έχουν μικρή προγνωστική αξία. Αυτό έχει ως αποτέλεσμα η προσέγγιση αυτή να λειτουργεί σωστά μόνο όταν λαμβάνει επαρκώς μεγάλη εισαγωγή κειμένου. Έτσι θα μπορούσαμε να πούμε πως οι στατιστικές μέθοδοι ενώ είναι σε θέση να ταξινομούν με βάση το συναίσθημα το κείμενο ενός χρήστη σε επίπεδο σελίδας δεν λειτουργεί καλά σε μικρότερα κείμενα όπως προτάσεις και σχόλια χρηστών.

Οι τεχνικές βασισμένες στην έννοια: Οι τεχνικές αυτές χρησιμοποιούν οντολογίες διαδικτύου ή σημασιολογικά δίκτυα ώστε να επιτευχθεί σημασιολογική ανάλυση.

Εξόρυξη δεδομένων

Εξόρυξη δεδομένων (ή ανακάλυψη γνώσης από βάσεις δεδομένων)[1] είναι η εξεύρεση μιας (ενδιαφέρουσας, αυτονόητης, μη προφανούς και πιθανόν χρήσιμης) πληροφορίας ή προτύπων από μεγάλες βάσεις δεδομένων με χρήση αλγορίθμων ομαδοποίησης ή κατηγοριοποίησης και των αρχών της στατιστικής, της τεχνητής νοημοσύνης, της μηχανικής μάθησης και των συστημάτων βάσεων δεδομένων. Στόχος της εξόρυξης δεδομένων είναι η πληροφορία που θα εξαχθεί και τα πρότυπα που θα προκύψουν να έχουν δομή κατανοητή προς τον άνθρωπο έτσι ώστε να τον βοηθήσουν να πάρει τις κατάλληλες αποφάσεις.

Ο πραγματικός στόχος της εξόρυξης δεδομένων είναι η αυτόματη ή ημιαυτόματη ανάλυση μεγάλων ποσοτήτων δεδομένα για την εξαγωγή κάποιου ενδιαφέροντος προτύπου που ήταν άγνωστο μέχρι εκείνη τη στιγμή, όπως ομάδες από εγγραφές

δεδομένων (συσταδοποίηση), ασυνήθιστες εγγραφές (*anomaly detection*) και εξαρτήσεις (κανόνες συσχετίσεων). Αυτό συνήθως συμπεριλαμβάνει τη χρήση βάσης δεδομένων όπως χωρικά ευρετήρια. Αυτά τα πρότυπα ύστερα μπορούν να θεωρηθούν ως μία περιγραφή των δεδομένων εισαγωγής και να χρησιμοποιηθούν για περαιτέρω ανάλυση ή για παράδειγμα στην εκμάθηση μηχανής και στην προγνωστική ανάλυση. Για παράδειγμα, η εξόρυξη δεδομένων θα μπορούσε να προσδιορίσει πολλαπλά σύνολα στα δεδομένα, τα οποία μπορούν να χρησιμοποιηθούν μετά για να εξασφαλίσουν περισσότερο ακριβή αποτελέσματα από ένα σύστημα υποστήριξης αποφάσεων. Παρότι η συλλογή δεδομένων και η προετοιμασία δεδομένων, αλλά και η ερμηνεία των αποτελεσμάτων και εκθέσεων δεν αποτελούν μέρος της εξόρυξης δεδομένων, παρ' όλα αυτά ανήκουν στην ανακάλυψη γνώσης από βάσεις δεδομένων σαν κάποια επιπρόσθετα βήματα.

Άλλοι σχετικοί όροι της εξόρυξης δεδομένων είναι οι *data dredging*, *data fishing* και *data snooping*, που αναφέρονται στην χρήση μεθόδων της εξόρυξης δεδομένων για να πάρουν δείγματα από μεγαλύτερη συλλογή δεδομένων που είναι (ή μπορεί να είναι) πολύ μικρά για αξιόπιστα στατιστικά συμπεράσματα που έγιναν σχετικά με τη εγκυρότητα των προτύπων που ανακαλύφθηκαν. Αυτές οι μέθοδοι, επίσης, μπορούν να χρησιμοποιηθούν για την δημιουργία νέων υποθέσεων προς εξέταση έναντι μεγαλύτερων συλλογών δεδομένων.

Η διαδικασία ανακάλυψης γνώσης από βάσεις δεδομένων(KDD) συνήθως ορίζεται από τα εξής στάδια:

1. Συλλογή
2. Προεπεξεργασία
3. Μετασχηματισμός
4. Εξόρυξη δεδομένων
5. Ερμηνεία/Αξιολόγηση

Υπάρχουν όμως κι άλλες παραλλαγές για τον ορισμό των σταδίων αυτών σύμφωνα και με το CRoss Industry Standard Process for Data Mining (CRISP-DM) όπου τα στάδια έχουν ως εξής:

1. Κατανόηση Θέματος
2. Κατανόηση δεδομένων
3. Προετοιμασία δεδομένων
4. Μοντελοποίηση
5. Αξιολόγηση
6. Ανάπτυξη ή απλοποιημένη διαδικασία όπως
    1. Προ-επεξεργασία
    2. Εξόρυξη δεδομένων
    3. Επικύρωση αποτελέσματος.

Ιατρική

Τα τελευταία χρόνια, η εξόρυξη δεδομένων χρησιμοποιείται ευρέως στους τομείς της ιατρικής, όπως η βιοϊατρική ,το DNA ,η γενετική και η φαρμακευτική. Στον τομέα της γενετικής, ο σκοπός είναι να κατανοήσουμε την χαρτογράφηση της σχέσης μεταξύ της μεταβολής των ακολουθιών του ανθρώπινου DNA και την προδιάθεση στην αρρώστια. Η εξόρυξη δεδομένων είναι ένα σημαντικό εργαλείο που μπορεί να βοηθήσει στην βελτίωση της διάγνωσης, της πρόληψης και της θεραπείας των ασθενειών.

- Εξαιτίας της αύξησης των βιοϊατρικών ερευνών, η μεγάλη κλίμακα γονιδιακών προτύπων και λειτουργιών πρέπει να εξετασθεί. Τα εργαλεία της εξόρυξης δεδομένων μπορούν να βοηθήσουν σε μεγάλο βαθμό για να μελετήσουμε την σύσταση του DNA και να βρούμε ποικίλα πρότυπα και λειτουργίες αυτού.
- Ένας από τους κύριους στόχους που σχετίζεται με την ανάλυση δεδομένων του DNA είναι η σύγκριση ποικίλων ακολουθιών και η αναζήτηση ομοιοτήτων μεταξύ των δεδομένων του DNA. Η σύγκριση κυρίως περιλαμβάνει την γονιδιακή ακολουθία υγιών και βλαβερών ιστών για να βρει την διαφορά ανάμεσα σε αυτούς τους δύο τύπους. Αυτό μπορεί να επιτευχθεί ανακτώντας τις τάξεις υγιών αλλά και βλαβερών γονιδιακών ακολουθιών και μετά βρίσκοντας τις συχνά εμφανιζόμενες μορφές των δύο τάξεων. Αυτή η ανάλυση βοηθάει στο να βρίσκουμε τις ομοιότητες και τις διαφορές στις γενετικές ακολουθίες.
- Στην βιοϊατρική , ερευνάται αν οι περισσότερες ασθένειες προκαλούνται από ένα συνδυασμό των γονιδίων. Η μέθοδος της συσχέτισης χρησιμοποιείται για να καθορίσει την συνύπαρξη ομάδων των γονιδίων και επίσης μπορούμε να εξετάσουμε την αλληλεπίδραση και την σχέση μεταξύ των γονιδίων.
- Τα εργαλεία της οπτικοποίησης παίζουν επίσης ένα σημαντικό ρόλο στην εξόρυξη δεδομένων στην βιοϊατρική. Τα εργαλεία αυτά μπορούν να παρουσιάσουν πολύπλοκες δομές γονιδίων σε γράφους, δένδρα και αλυσίδες. Η οπτική παρουσίαση βοηθάει στην καλύτερη κατανόηση αυτών των δομών για ανακάλυψη γνώσης και εξερεύνηση των δεδομένων.
- Υπάρχουν διάφοροι συνδυασμοί γονιδίων που συμβάλλουν στις ασθένειες, αλλά αυτά τα γονίδια ενεργοποιούνται σε διαφορετικά επίπεδα. Η ανάλυση μονοπατιού (path analysis[5]) χρησιμοποιείται για να συνδέει διαφορετικά γονίδια με διαφορετικά στάδια κατά την εξέλιξη της ασθένειας. Η ανάλυση μονοπατιού διαδραματίζει ένα σπουδαίο ρόλο στην γενετική.

Οικονομία

Άλλος τομέας που εφαρμόζεται η εξόρυξη δεδομένων είναι η οικονομία. Τα οικονομικά δεδομένα κυρίως συλλέγονται από τράπεζες και από άλλους οικονομικούς οργανισμούς. Τα δεδομένα αυτά συνήθως είναι αξιόπιστα, ολοκληρωμένα και έχουν υψηλή ποιότητα και απαιτούν συστηματική μέθοδο για την ανάλυση αυτών. Η συνεισφορά της εξόρυξης δεδομένων στην επιστήμη της οικονομίας συναντάται

στην συλλογή και κατανόηση των δεδομένων, στην βελτίωση δεδομένων (data refinement), στην δημιουργία και εκτίμηση ενός μοντέλου και στην ανάπτυξη αυτού. Η σωστή ανάλυση των οικονομικών δεδομένων μας διευκολύνει στο να παίρνουμε καλύτερες αποφάσεις ενεργώντας σύμφωνα με την ανάλυση της αγοράς. Τα εργαλεία και οι τεχνικές της εξόρυξης δεδομένων βοηθούν στο να αναλύσουμε τα οικονομικά δεδομένα με τους παρακάτω τρόπους:

- Τα δεδομένα που συλλέγονται από διάφορα οικονομικά ινστιτούτα, όπως οι τράπεζες ,συγκεντρώνονται αρχικά στην αποθήκη δεδομένων (data warehouse). Οι τεχνικές της πολυδιάστατης ανάλυσης δεδομένων χρησιμοποιούνται για την ανάλυση τέτοιων δεδομένων που συλλέγονται στην αποθήκη δεδομένων για τις γενικές ιδιότητές του.
- Μία άλλη εφαρμογή της εξόρυξης δεδομένων σχετίζεται με την πρόβλεψη αποπληρωμής δανείου και πολιτικές πίστωσης του πελάτη. Μέθοδοι της εξόρυξης όπως η επιλογή χαρακτηριστικών (feature selection) βοηθάει στην ταυτοποίηση ποικίλων χαρακτηριστικών όπως το επίπεδο εισοδήματος του πελάτη, την εξόφληση ανάλογα με τα έσοδα, την πιστωτική του ιστορία κτλ. Με την επεξεργασία αυτών των χαρακτηριστικών, η τράπεζα μπορεί να αποφασίσει για τις πολιτικές δανειοδότησης βάσει των σχετικά χαμηλών κινδύνων. Οι τεχνικές της συσταδοποίησης και της ταξινόμησης βοηθούν τα οικονομικά ινστιτούτα να ομαδοποιούν διάφορους πελάτες που έχουν κοινά χαρακτηριστικά. Η αποτελεσματική συσταδοποίηση και οι μέθοδοι φιλτραρίσματος βοηθούν τις τράπεζες να ταυτοποιούν μία ομάδα πελατών, να συσχετίζουν ένα νέο πελάτη με την παρούσα ομάδα και να τους παρέχουν κοινά οφέλη.
- Τα εργαλεία της εξόρυξης δεδομένων βοηθούν τα οικονομικά ινστιτούτα να αναγνωρίζουν τις απάτες και τα εγκλήματα από παραποιημένα δεδομένα από τις διάφορες βάσεις δεδομένων και από το ιστορικό συναλλαγών που έγιναν από τους πελάτες. Οι τεχνικές οπτικοποίησης βοηθούν στην παρουσίαση δεδομένων με διαφορετικές μορφές, όπως γράφοι που βασίζονται σε συγκεκριμένα γνωρίσματα. Προβάλλοντας τα δεδομένα από διάφορες οπτικές γωνίες, η τράπεζα δύναται να διακρίνει τους πελάτες που έχουν επιχειρήσει παράνομες πράξεις και μετά μια λεπτομερής έρευνα αυτών των ύποπτων περιπτώσεων βοηθάει στην εξιχνίαση των απατών και των εγκλημάτων.

Τηλεπικοινωνία

Η τηλεπικοινωνιακή βιομηχανία αναπτύσσεται πολύ γρήγορα όπως και η τεχνολογία. Αυτές τις μέρες οι τηλεπικοινωνιακές υπηρεσίες έχουν επεκταθεί από τοπικές και μεγάλης απόστασης τηλεπικοινωνίες, στην χρήση φαξ, συσκευές τηλεειδοποίησης, κινητό τηλέφωνο, και ηλεκτρονικό ταχυδρομείο. Εξαιτίας των εξελίξεων στις τηλεπικοινωνιακές τεχνολογίες και για να δουλέψουν αποτελεσματικά αυτές οι τεχνολογίες, οι τεχνικές της εξόρυξης δεδομένων ενσωματώνονται σε αυτές τις τεχνολογίες για να παράγουν αποδοτικά αποτελέσματα. Η εξόρυξη δεδομένων βοηθάει στην διάκριση τηλεπικοινωνιακών προτύπων, καταπολέμησης παράνομων δραστηριοτήτων, και επίσης βοηθάει στην καλύτερη χρήση των πόρων και στη βελτίωση της ποιότητας των υπηρεσιών. Η εξόρυξη δεδομένων βελτιώνει τις τηλεπικοινωνιακές υπηρεσίες με τους εξής τρόπους:

- Τα τηλεπικοινωνιακά δεδομένα που συλλέγονται, περιλαμβάνουν τον τύπο κλήσης, την τοποθεσία του καλούντος και του κληθέντος, τον χρόνο κλήσης,

την διάρκεια κλήσης κλπ. Η πολυδιάστατη ανάλυση βοηθά στον προσδιορισμό και στην σύγκριση του φορτιού του συστήματος, κίνηση δεδομένων, και κέρδος κλπ. Η ανάλυση μπορεί να δείξει διαγράμματα και γράφους των πόρων του συστήματος, του προορισμού κλπ κάνοντας χρήση των εργαλείων οπτικοποίησης της εξόρυξης δεδομένων. Τέτοια εργαλεία όπως η συσχετισμένη οπτικοποίηση και η συσταδοποίηση παρέχουν χρήσιμες υπηρεσίες στην ανάλυση των δεδομένων τηλεπικοινωνίας.

- Το κυρίως πρόβλημα που αντιμετωπίστηκε από την βιομηχανία τηλεπικοινωνιών είναι οι παράνομες δραστηριότητες. Αυτές οι δραστηριότητες μπορεί να έχουν να κάνουν με σκόπιμες κλήσεις κατά την ώρα αιχμής, περιοδικές κλήσεις κ.α. με αποτέλεσμα να επιδρούν αρνητικά στην επίδοση του δικτύου επικοινωνιών. Μέθοδοι όπως η συσταδοποίηση και η ανάλυση ακραίων τιμών, συνεισφέρει στην ανίχνευση παράνομων προτύπων βελτιώνοντας την αποτελεσματικότητα των υπηρεσιών τηλεπικοινωνίας.
- Εκμεταλλευόμενοι τα εργαλεία της εξόρυξης δεδομένων είναι δυνατή η δημιουργία προφίλ των πελατών και ο εντοπισμός βλαβών στο δίκτυο.
- Τέλος, η ανάλυση συσχετιζόμενων και ακολουθιακών προτύπων ενθαρρύνει την προώθηση νέων και ποικίλων υπηρεσιών τηλεπικοινωνίας.

Συμπέρασμα

Οι εκτεταμένες αλλαγές στην υιοθέτηση και χρησιμοποίηση των νέων τεχνολογιών στις μεγάλες αλλά και στις μικρές επιχειρήσεις έχει ως αποτέλεσμα την συγκέντρωση μεγάλου αριθμού δεδομένων από τις οικονομικές συναλλαγές. Είναι ευθύνη του αναλυτή να αναλύσει αυτές τις συναλλαγές και να εντοπίσει τις απάτες και τα λάθη μέσα σε αυτές. Λόγω των αλλαγών των τάσεων μέσα στην επιχείρηση, είναι δύσκολο να επεξεργαστείς και να αναλύσεις τα δεδομένα με παλαιές μεθόδους. Οι περιορισμοί που εμφανίζουν αυτές οι μέθοδοι μας έχουν οδηγήσει στην εκμετάλλευση των εργαλείων της εξόρυξης για καλύτερα και περισσότερο αξιόπιστα αποτελέσματα.

Συλλογή δεδομένων από το διαδίκτυο

Υπάρχουν τρόποι για να μπορέσουμε να αντλήσουμε ένα σύνολο από δεδομένα που βρίσκονται στο διαδίκτυο εύκολα και γρήγορα. Για παράδειγμα μπορείτε:

- Να πάρετε τα δεδομένα αυτά από διαδικτυακές διεπαφές (ή API) που παρέχονται από διαδικτυακές βάσεις δεδομένων και πολλές σύγχρονες εφαρμογές (όπως το Twitter, το Facebook και πολλές άλλες). Με αυτόν τον τρόπο μπορείτε να έχετε εύκολη πρόσβαση σε δεδομένα κυβερνητικού ή εμπορικού περιεχομένου, καθώς και σε δεδομένα από μέσα κοινωνικής δικτύωσης.

- Να εξάγετε πληροφορίες από αρχεία PDF. Πρόκειται για δύσκολο εγχείρη-

μα, καθώς το PDF αποτελεί γλώσσα εκτυπωτών και δε συγκρατεί πολλές πληροφορίες σχετικά με τη δομή των δεδομένων, οι οποίες εμπεριέχονται σε ένα έγγραφο. Αν και η ανάκτηση στοιχείων από αρχείο PDF δεν εξετάζεται σε αυτό το βιβλίο, υπάρχουν διάφορα εργαλεία και βοηθητικά βίντεο.

Scrapping Websites

Είστε σε μια ιστοσελίδα, βλέπετε έναν ενδιαφέροντα πίνακα και προσπαθείτε να τον αντιγράψετε στο Excel για να προσθέσετε αργότερα κάποια νούμερα ή να τον αποθηκεύσετε για μελλοντική χρήση. Ωστόσο, αυτό δε συμβαίνει πάντα ή μπορεί οι πληροφορίες που ζητάτε να βρίσκονται διασκορπισμένες σε πολλές διαφορετικές σελίδες. Καθώς η αντιγραφή με το χέρι είναι κουραστική, είναι πιο λογικό να χρησιμοποιήσετε έναν κώδικα.Το πλεονέκτημα του scraping είναι ότι εφαρμόζεται σε κάθε ιστοσελίδα, από την πρόγνωση του καιρού μέχρι τις κρατικές δαπάνες, ακόμα κι αν η σελίδα δε διαθέτει διεπαφή για πρόσβαση σε ανεπεξέργαστα δεδομένα.

Υπάρχουν φυσικά και περιορισμοί στο scrapping. Μερικοί από τους παράγοντες που δυσκολεύουν τη διαδικασία αυτή είναι οι εξής:

- Κακοφορμισμένος κώδικας HTML με καθόλου ή ελάχιστες πληροφορίες σχετικά με τη δομή (όπως σε παλαιότερες κυβερνητικές ιστοσελίδες).
- Συστήματα πιστοποίησης που προορίζονται για να εμποδίζουν την αυτόματη πρόσβαση (όπως οι κώδικες CAPTCHA και τα paywalls).
- Συστήματα session-based που χρησιμοποιούν cookies από το πρόγραμμα περιήγησης για να καταγράφουν τις δραστηριότητες του χρήστη.
- Έλλειψη μιας ολοκληρωμένης καταχώρησης των αντικειμένων και πιθανές αναζητήσεις με μπαλαντέρ χαρακτήρες.
- Απαγόρεψη πρόσβασης στα δεδομένα από τους διαχειριστές των εξυπηρετητών

Ένα πρόσθετο είδος περιορισμών αποτελούν τα νομικά εμπόδια: κάποιες χώρες αναγνωρίζουν τα δικαιώματα στις βάσεις δεδομένων. Ως εκ τούτου, έχετε περιορισμένο δικαίωμα αναπαραγωγής ήδη δημοσιευμένων πληροφοριών στο διαδίκτυο. Μπορείτε βέβαια να αγνοήσετε αυτήν την παράμετρο και να ενεργήσετε πάραυτα. Αυτό όμως εξαρτάται από τη δικαιοδοσία σας –μπορεί να έχετε ειδικά δικαιώματα ως δημοσιογράφος. Το scraping σε διαθέσιμα κυβερνητικά δεδομένα επιτρέπεται, καλό θα ήταν όμως να είστε σίγουρος, πριν τα δημοσιεύσετε. Εμπορικοί οργανισμοί –και κάποιες ΜΚΟ- είναι λιγότερο ανεκτικοί και μπορεί να ισχυριστούν ότι «σαμποτάρετε» την πολιτική τους. Άλλες πληροφορίες παραβιάζουν την ιδιωτικότητα του πολίτη και καταστρατηγούν νόμους σχετικά με το απόρρητο δεδομένων ή την επαγγελματική ηθική.

Εργαλεία που βοηθούν το Scrape

Τα προγράμματα για την ανάκτηση τεράστιου όγκου πληροφοριών από ιστοσελίδες αφθονούν, ενώ περιλαμβάνουν πρόσθετα φυλλομετρητή και κάποιες διαδικτυακές υπηρεσίες. Ανάλογα με το φυλλομετρητή σας, εργαλεία όπως το Readability (για την ανάκτηση του κειμένου από μια σελίδα) ή το DownThemAll (για να κατεβάζετε πολλά αρχεία ταυτόχρονα) αυτοματοποιούν κουραστικές διαδικασίες, ενώ το πρόσθετο Scraper του Chrome δημιουργήθηκε ακριβώς για την ανάκτηση πίνακα από ιστοσελίδες. Τα πρόσθετα του κατασκευαστικού λογισμικού όπως το FireBug (για τον Firefox, ενώ υπάρχει ήδη το αντίστοιχο πρόγραμμα για τα Chrome, Safari και IE) σας επιτρέπουν να καταγράψετε πώς ακριβώς είναι η δομή μιας ιστοσελίδας και τι μηνύματα ανταλλάσσονται μεταξύ του προγράμματος περιήγησής σας και του εξυπηρετητή.

Το ScraperWiki είναι μια ιστοσελίδα που σας παρέχει τη δυνατότητα να κωδικοποιήσετε scrapers σε πολλές διαφορετικές γλώσσες προγραμματισμού, όπως οι Python, Ruby και PHP. Αν θέλετε να αποφύγετε το δύσκολο εγχείρημα της δημιουργίας περιβάλλοντος προγραμματισμού στον υπολογιστή σας και θέλετε να ξεκινήσετε κατευθείαν το scraping, τότε αυτός είναι ο τρόπος. Άλλες διαδικτυακές υπηρεσίες, όπως το Google Spreadsheets και το Yahoo! Pipes σας επιτρέπουν επίσης να ανακτήσετε κάποιες πληροφορίες από ιστοσελίδες.

Πως λειτουργεί ένας Web Scraper

Οι Web Scrapers είναι συνήθως μικρά κομμάτια κώδικα γραμμένα σε μια γλώσσα προγραμματισμού όπως οι Python, Ruby ή PHP. Η επιλογή της γλώσσας εξαρτάται σε μεγάλο βαθμό από την κοινότητα στην οποία έχετε πρόσβαση: εάν κάποιος στο γραφείο τύπου ή στην πόλη σας δουλεύει ήδη με μια από αυτές τις γλώσσες, καλό θα ήταν να χρησιμοποιήσετε την ίδια.

Παρόλο που μπορεί κάποια εύκολα προαναφερθέντα εργαλεία να είναι χρήσιμα στην αρχή, η πραγματική δυσκολία του scraping βρίσκεται στον εντοπισμό των σωστών σελίδων και των σωστών στοιχείων μέσα σε αυτές, προκειμένου να ανακτήσετε τις πληροφορίες που επιθυμείτε. Οι διαδικασίες αυτές δε σχετίζονται με τον προγραμματισμό, αλλά με την κατανόηση της δομής της ιστοσελίδας και της βάσης δεδομένων.

Κατά την προβολή μιας ιστοσελίδας, το πρόγραμμα περιήγησής σας θα χρησιμοποιήσει δύο τεχνολογίες: την HTTP, για να επικοινωνήσει με τον εξυπηρετητή και να ζητήσει συγκεκριμένες πληροφορίες όπως έγγραφα, εικόνες ή βίντεο, και την HTML, τη γλώσσα κατασκευής ιστοσελίδων.

Για το scrape σε ιστοσελίδες, πρέπει να γνωρίζετε κάποια πράγματα σχετικά με τα διαφορετικά είδη στοιχείων που βρίσκονται σε ένα έγγραφο HTML. Για παράδειγμα, το στοιχείο <table> περιλαμβάνει έναν πίνακα, ο οποίος έχει <tr> στοιχεία (σειρές πίνακα) για τις σειρές του που με τη σειρά τους περιλαμβάνουν <td> (δεδομένα πίνακα) για κάθε κελί. Το πιο κοινό στοιχείο που θα συναντήσετε είναι το που ουσιαστικά περιλαμβάνει κάθε στοιχείο περιεχομένου. Ο ευκολότερος τρόπος για να καταλάβετε τη σημασία των στοιχείων αυτών είναι με το πρόγραμμα developer toolbar στο πρόγραμμα περιήγησής σας: έτσι θα μπορέσετε να δείτε οποιαδήποτε ιστοσελίδα μαζί με τον υποκείμενο κώδικά της.Οι ετικέτες μοιάζουν με βιβλιοστάτες, καθώς σηματοδοτούν την αρχή και το τέλος μιας ενότητας. Για παράδειγμα, η ετικέτα συμβολίζει την αρχή μιας φράσης με πλάγιους χαρακτήρες ενώ το συμβολίζει το τέλος της. Εύκολο.

Γλώσσα προγραμματισμού υλοποίησης

Θα χρησιμοποιήσουμε την γλώσσα python. Πρώτα όμως θα περιγράψουμε τι είναι η γλώσσα python. Η Python είναι μια υψηλού επιπέδου γλώσσα προγραμματισμού η οποία δημιουργήθηκε από τον Ολλανδό Γκβίντο βαν Ρόσσουμ (Guido van Rossum) το 1990. Ο κύριος στόχος της είναι η αναγνωσιμότητα του κώδικά της και η ευκολία χρήσης της και το συντακτικό της επιτρέπει στους προγραμματιστές να εκφράσουν έννοιες σε λιγότερες γραμμές κώδικα απ'ότι θα ήταν δυνατόν σε γλώσσες όπως η C++ ή η Java. Διακρίνεται λόγω του ότι έχει πολλές βιβλιοθήκες που διευκολύνουν ιδιαίτερα αρκετές συνηθισμένες εργασίες και για την ταχύτητα εκμάθησης της.

Οι διερμηνευτές της Python είναι διαθέσιμοι για εγκατάσταση σε πολλά λειτουργικά συστήματα, επιτρέποντας στην Python την εκτέλεση κώδικα σε ευρεία γκάμα συστημάτων. Χρησιμοποιώντας εργαλεία τρίτων, όπως το Py2exe ή το Pyinstaller, ο κώδικας της Python μπορεί να πακεταριστεί σε αυτόνομα εκτελέσιμα προγράμματα για μερικά από τα πιο δημοφιλή λειτουργικά συστήματα, επιτρέποντας τη διανομή του βασισμένου σε Python λογισμικού για χρήση σε αυτά τα περιβάλλοντα χωρίς να απαιτείται εγκατάσταση του διερμηνευτή της Python.

Η Python αναπτύσσεται ως ανοιχτό λογισμικό (open source) και η διαχείρισή της γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation. Ο κώδικας διανέμεται με την άδεια Python Software Foundation License η οποία είναι συμβατή με την GPL. Το όνομα της γλώσσας προέρχεται από την ομάδα άγγλων κωμικών Μόντυ Πάιθον.

Για τη συγγραφή προγραμμάτων είναι απαραίτητος ένας κειμενογράφος ή ακόμα καλύτερα ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment - IDE), το οποίο είναι ένα ειδικό λογισμικό για την ανάπτυξη εφαρμογών. Η Python έρχεται μαζί με ένα εύχρηστο περιβάλλον ανάπτυξης με την ονομασία IDLE. Τα αρχικά του έρχονται από τις λέξεις Interactive DeveLopment Environment και είναι γραμμένο σε Python από τον Guido van Rossum. Χρησιμοποιεί τη βιβλιοθήκη γραφικών Tkinter, οπότε μπορεί να εκτελεσθεί σε περιβάλλον Linux, Windows και Mac OS X. Το IDLE μας δίνει τη δυνατότητα να χρησιμοποιήσουμε διαδραστικά τον διερμηνευτή της γλώσσας, να γράψουμε και να επεξεργαστούμε προγράμματα, να τα αποθηκεύσουμε σε αρχεία, να τα εκτελέσουμε, να κάνουμε αποσφαλμάτωση.Εμείς θα χρησιμοποιήσουμε το περιβάλλον spyder.

Γιατί Python;

- Γρήγορη προτυποποίηση
- Προγραμματισμός στον παγκόσμιο ιστό
- Scripting
- Εκπαίδευση
- Επιστήμη
- Εφαρμογές με γραφική διεπαφή

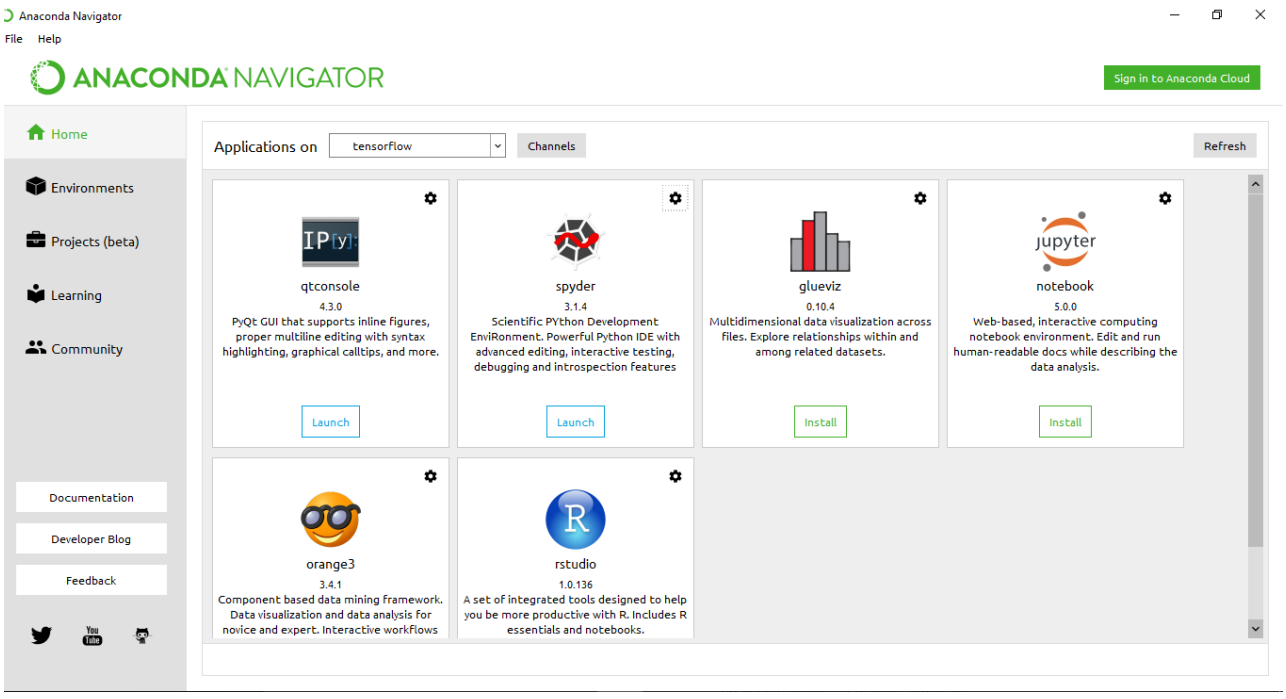
Η Python είναι μια εύκολη στην εκμάθηση, ισχυρή γλώσσα προγραμματισμού. Έχει αποδοτικές δομές δεδομένων υψηλού επιπέδου και μια απλή αλλά αποτελεσματική προσέγγιση στον αντικειμενοστρεφή προγραμματισμό. Η κομψή σύνταξη της Python και οι δυναμικοί τύποι της, μαζί με τη λειτουργία της ως διερμηνευόμενης (αντί μεταγλωττιζόμενης) γλώσσας, την καθιστούν την ιδανική γλώσσα για δημιουργία σεναρίων εντολών και για ταχεία ανάπτυξη εφαρμογών σε πολλούς τομείς και στις περισσότερες πλατφόρμες.

Η **Python** είναι μια από εκείνες τις σπάνιες γλώσσες που ισχυρίζονται ότι είναι και απλές και ισχυρές. Θα εκπλαγείτε ευχάριστα από την ευκολία με την οποία θα συ-

γκεντρώνεστε στην λύση ενός προβλήματος, παρά στο συντακτικό και στην δομή της γλώσσας στην οποία προγραμματίζετε. Η Python είναι μια απλή και μινιμαλιστική γλώσσα. Το διάβασμα ενός καλού προγράμματος σε Python είναι σαν το διάβασμα των Αγγλικών, αλλά πολύ αυστηρών Αγγλικών! Αυτή η ομοιότητα της Python με ψευδοκώδικα είναι ένα από τα πιο ισχυρά σημεία της. Σας επιτρέπει να συγκεντρώνεστε στην λύση του προβλήματος, αντί στην ίδια την γλώσσα.

ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

Θα χρησιμοποιήσουμε το Anaconda Navigator. Το Anaconda Navigator είναι ένα γραφικό περιβάλλον εργασίας γραφικών (GUI) που περιλαμβάνεται στο Anaconda® και μας επιτρέπει να ξεκινήσουμε εφαρμογές και να διαχειριστούμε εύκολα πακέτα, περιβάλλοντα και κανάλια conda χωρίς να χρησιμοποιήσουμε εντολές γραμμής εντολών. Μπορούμε να διαμορφώσουμε το Navigator για να αναζητήσουμε πακέτα στο Anaconda Cloud ή σε ένα τοπικό κατάστημα Anaconda. Διατίθεται για Windows, macOS και Linux.



3.1 Ενδεικτική εικόνα

Η επίσημη ιστοσελίδα για κατέβασμα και εγκατάσταση της εφαρμογής Anaconda Navigator

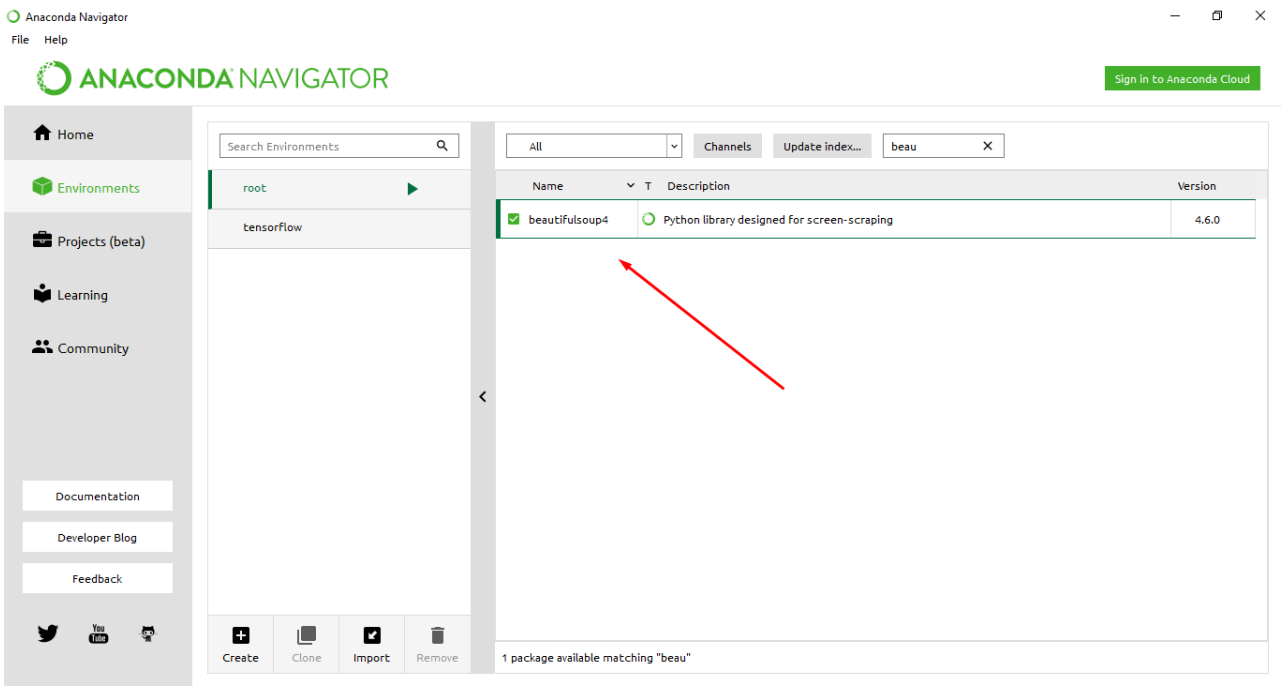
Υλοποίηση αλγορίθμου

Scrapping ενδεικτικό παράδειγμα

Θα χρησιμοποιηθεί η βιβλιοθήκη BeautifulSoup

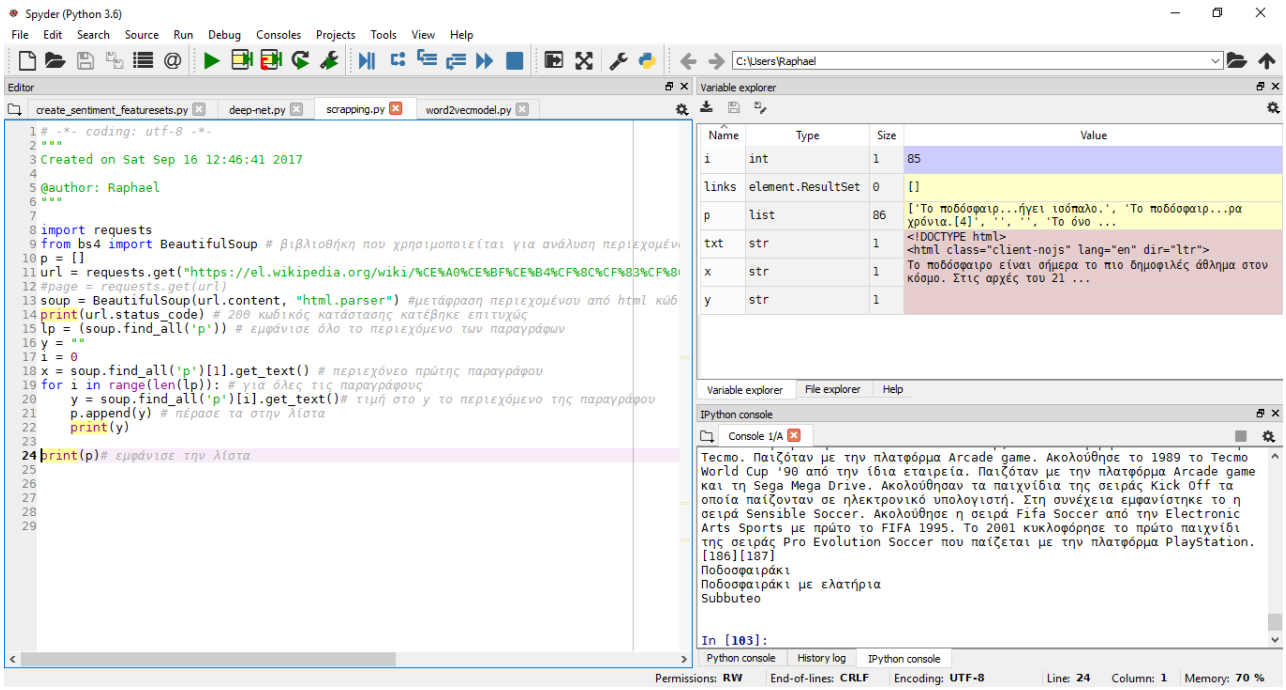
Η βιβλιοθήκη Beautiful Soup είναι μια βιβλιοθήκη της Python για την εξαγωγή δεδομένων από αρχεία HTML και XML. Λειτουργεί με τον αγαπημένο σας αναλυτή για να παρέχει ιδιωματικούς τρόπους πλοήγησης, αναζήτησης και τροποποίησης του δέντρου ανάλυσης. Συνήθως εξοικονομεί ώρες προγραμματισμου ή ημέρες εργασίας.Αυτή την στιγμή βρίσκεται στην έκδοση 4 η οποία και ενδείκνυται για την δημιουργία νέων projects.

Είναι αναγκαίο να γίνει import της βιβλιοθήκης BeautifulSoup.Στην προκειμένη περίπτωση την έχω ήδει εγκαταστήσει.



3.2 Παραδειγμα εγκατάστασης της βιβλιοθήκης BeautifulSoup

Στη συνέχεια θα ανοίξουμε το spyder το περιβάλλον εργασίας μας δηλαδη για να γράψουμε τον κώδικα.



3.3 Το πρόγραμμα spyder με τον κώδικα και τα αποτελέσματα

Παρακάτω παραθέτω τον κώδικα για επεξήγηση

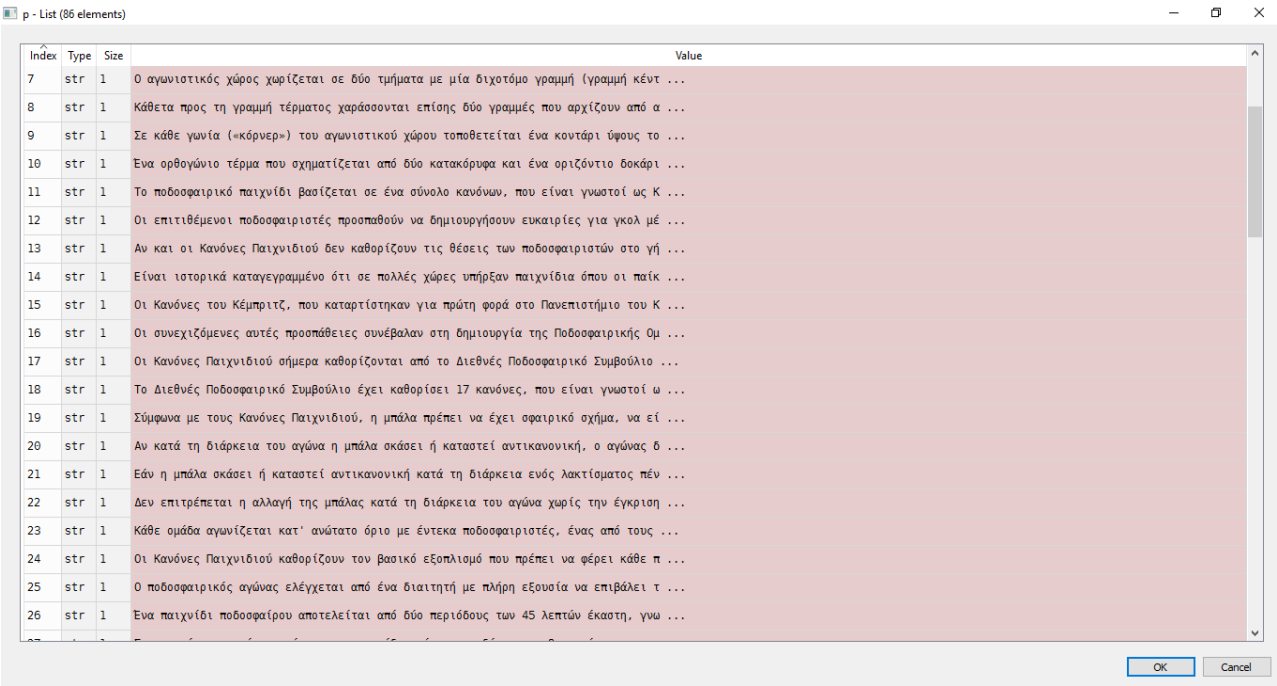
```
# -*- coding: utf-8 -*-
"""
Created on Sat Sep 16 12:46:41 2017

@author: Raphael
"""

import requests
from bs4 import BeautifulSoup
p = []
url = requests.get("https://el.wikipedia.org/wiki/%CE
#page = requests.get(url)
soup = BeautifulSoup(url.content, "html.parser")
print(url.status_code)
lp = (soup.find_all('p'))
y = ""
i = 0
x = soup.find_all('p')[1].get_text()
for i in range(len(lp)):
    y = soup.find_all('p')[i].get_text()
    p.append(y)
    print(y)

print(p)
```

3.4 Κώδικας παράδειγμα για scrapping απο την wikipedia

Βήματα ανάλυσης του κώδικα

- Εισάγουμε την βιβλιοθήκη requests η οποία θα χρησιμοποιηθεί για την εισαγωγή της ιστοσελίδας από την οποία θα κατεβάσουμε το περιεχόμενο
- Εισάγουμε την απο την bs4 την μέθοδο BeautifulSoup
- Δημιουργούμε μια λίστα κενή
- Μετάφραση περιεχομένου απο τον html κώδικα
- Εκτύπωση του status code (αν είναι 200 κατέβηκε σωστά)
- Εμφάνησε όλο το περιεχόμενο των παραγράφων χρησιμοποιώντας την μέθοδο find_all
- Για όλες τις παραγράφους
- Κράτα σε μια μεταβλητή το περιεχόμενο κάθε παραγράφου και κάνε append σε λίστα
- Εκτύπωση της λίστας

Το scrapping είναι μιά πολύ σημαντική τεχνική που μας γλυτώνει από χρόνο και κόπο.Πλέον είμαστε ετοιμα για sentiment analysis



3.5 Εμφάνιση της λίστας

Μοντέλο bag-of-words και Δημιουργία λεξικού

Το μοντέλο bag-of-words είναι μια απλουστευμένη αναπαράσταση που χρησιμοποιείται στην επεξεργασία της φυσικής γλώσσας και στην ανάκτηση πληροφοριών (IR). Σε αυτό το μοντέλο, ένα κείμενο (όπως μια πρόταση ή ένα έγγραφο) αναπαρίσταται ως η τσάντα (multiset) των λέξεων της, αγνοώντας τη γραμματική και ακόμη και τη σειρά των λέξεων, διατηρώντας όμως την πολλαπλότητα. Το μοντέλο bag-of-words έχει επίσης χρησιμοποιηθεί για όραση στον υπολογιστή .

Το μοντέλο bag-of-words χρησιμοποιείται συνήθως σε μεθόδους ταξινόμησης εγγράφων όπου η συχνότητα εμφάνισης κάθε λέξης χρησιμοποιείται ως χαρακτηριστικό γνώρισμα για την κατάρτιση ενός ταξινομητή .

Στην δική μας περίπτωση θα χρησιμοποιήσουμε το μοντέλο bag-of-words.Επειδή δεν μπορούμε να αξιολογήσουμε τις προτάσεις ως ένα σύνολο από λέξεις θα πρέπει πρώτα να δημιουργήσουμε ένα λεξικό έτσι ώστε να αντιστοιχήσουμε κάθε λέξη με ένα διάνυσμα.

Για να πειραματιστούμε θα χρησιμοποιήσουμε 2 αρχεία όπου το ένα περιέχει προτάσεις που εκφράζουν θετικό συναίσθημα και το άλλο αρνητικό συναίσθημα. Κάθε αρχείο αποτελείται από 5.000 προτάσεις περίπου.

Παράδειγμα εφαρμογής

```
1) John likes to watch movies. Mary likes movies too.
```

```
(2) John also likes to watch football games.
```

Με βάση αυτά τα δύο έγγραφα κειμένου, ένας κατάλογος κατασκευάζεται ως εξής:

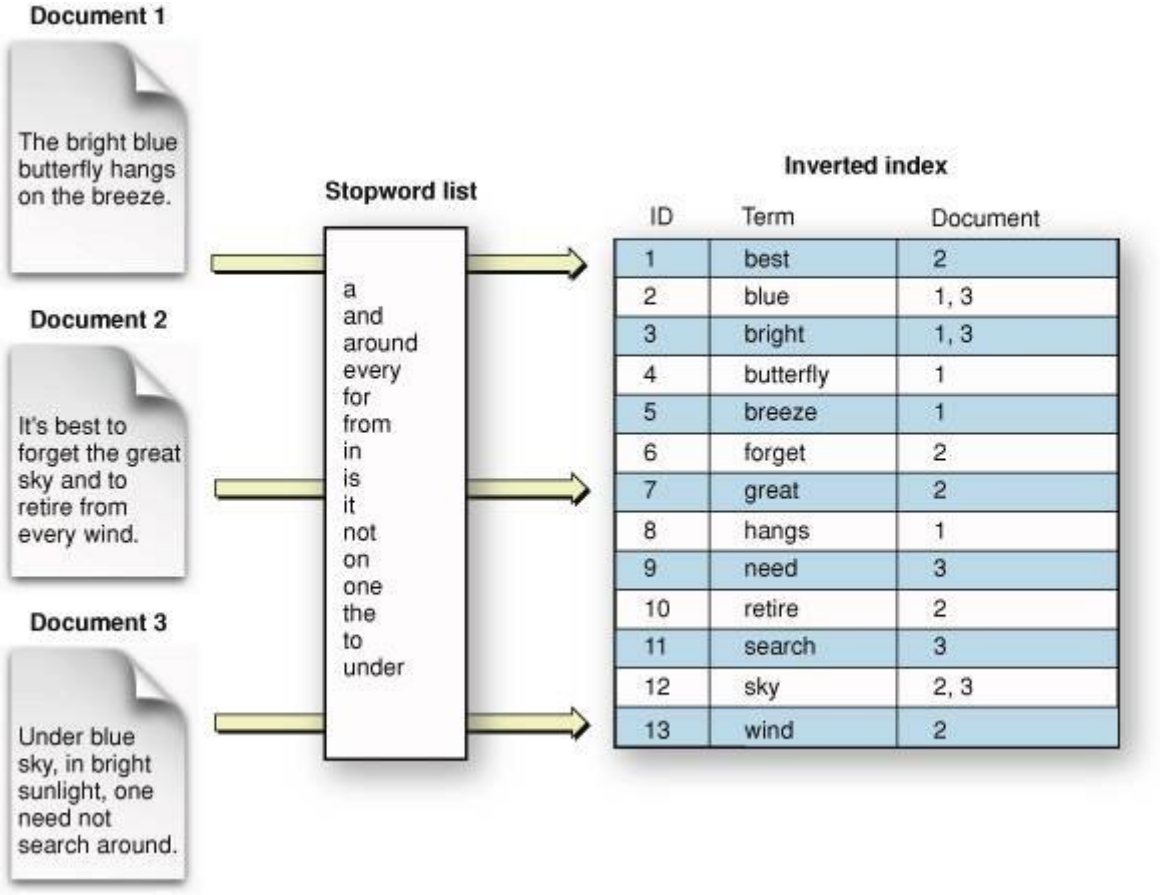
```
[
    "John",
    "likes",
```

```
    "to",
    "watch",
    "movies",
    "Mary",
    "too",
    "also",
    "football",
    "games"
]
```

Στην πράξη, το μοντέλο Bag-of-words χρησιμοποιείται κυρίως ως εργαλείο δημιουργίας χαρακτηριστικών. Αφού μετατρέψουμε το κείμενο σε "σακούλα λέξεων", μπορούμε να υπολογίσουμε διάφορα μέτρα για να χαρακτηρίσουμε το κείμενο. Ο συνηθέστερος τύπος χαρακτηριστικών ή χαρακτηριστικών που υπολογίζονται από το μοντέλο Bag-of-words είναι η μακροχρόνια συχνότητα, δηλαδή ο αριθμός των φορών που εμφανίζεται ένας όρος στο κείμενο. Για το παραπάνω παράδειγμα, μπορούμε να κατασκευάσουμε τις ακόλουθες δύο λίστες για να καταγράψουμε τις συχνότητες όρων όλων των διακριτών λέξεων:

```
( 1 ) [ 1 , 2 , 1 , 1 , 2 , 1 , 1 , 0 , 0 , 0 ]
( 2 ) [ 1 , 1 , 1 , 1 , 0 , 0 , 1 , 1 , 1 ]
```

Κάθε καταχώρηση των λιστών αναφέρεται στην καταμέτρηση της αντίστοιχης καταχώρησης στη λίστα (αυτή είναι και η παράσταση ιστόγραμμα). Για παράδειγμα, στην πρώτη λίστα (η οποία αντιπροσωπεύει το έγγραφο 1), οι δύο πρώτες καταχωρήσεις είναι "1,2". Η πρώτη καταχώρηση αντιστοιχεί στη λέξη "John", η οποία είναι η πρώτη λέξη στη λίστα και η τιμή της είναι "1" επειδή εμφανίζεται το "John" στο πρώτο έγγραφο 1 φορά. Ομοίως, η δεύτερη καταχώρηση αντιστοιχεί στη λέξη "likes" που είναι η δεύτερη λέξη στη λίστα και η τιμή της είναι "2" επειδή το "likes" εμφανίζεται στο πρώτο έγγραφο 2 φορές. Αυτή η αναπαράσταση λίστας (ή διανυσμάτων) δεν διατηρεί τη σειρά των λέξεων στις αρχικές προτάσεις, η οποία είναι μόνο το κύριο χαρακτηριστικό του μοντέλου Bag-of-words. Αυτό το είδος αναπαράστασης έχει αρκετές επιτυχημένες εφαρμογές, όπως για παράδειγμα το φιλτράρισμα του ηλεκτρονικού ταχυδρομείου .



3.6 Παράδειγμα μοντέλου bag-of-words

Αυτό που θα προσπαθήσουμε να κάνουμε εδώ είναι να χρησιμοποιήσουμε ένα νευρωνικό δίκτυο για να εντοπίσουμε σωστά το συναίσθημα. Πρώτον, τα δεδομένα μας είναι σε μορφή γλώσσας / λέξης, όχι αριθμητικής μορφής, η οποία πρέπει να μετατραπεί σε φορέα χαρακτηριστικών. Έτσι λοιπόν, αρχίζουμε να σκεφτόμαστε πώς θα μετατρέψουμε τα λόγια σε αριθμούς και στη συνέχεια θα κάνουμε μια

δεύτερη συνειδητοποίηση: τα κείμενά μας μπορεί να μην έχουν το ίδιο μήκος λέξεων ή χαρακτήρων. Πρόκειται για μια μεγάλη υπόθεση, δεδομένου ότι χρειαζόμαστε όλα τα χαρακτηριστικά γνωρίσματα να έχουν ακριβώς το ίδιο μήκος που πηγαίνει στην κατάρτιση, και φυσικά για την κατάρτιση.

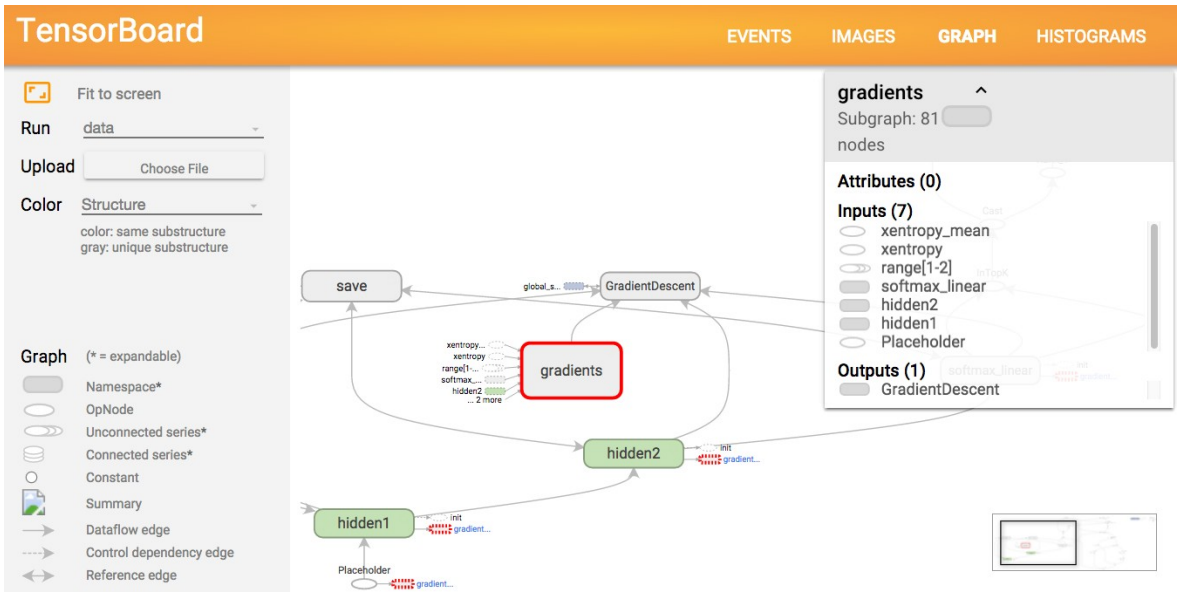
Μια επιλογή που έχουμε είναι να συντάξουμε μια λίστα με όλες τις μοναδικές λέξεις στο σύνολο εκπαίδευσης. Ας πούμε ότι είναι 3.500 μοναδικές λέξεις. Αυτές οι λέξεις είναι το λεξικό μας. Τώρα, δημιουργούμε ένα διάνυσμα, το διάνυσμα κατάρτισης των μηδενικών που έχει μέγεθος 1x3500, και στη συνέχεια έχουμε μια λίστα όλων των μοναδικών λέξεων που είναι επίσης 1x3500. Από εδώ, για κάθε λέξη που περιλαμβάνεται στο πρότυπο δείγμα μας, ελέγξουμε να δούμε αν είναι στο μοναδικό διάνυσμα λέξεων μας. Αν ναι, η τιμή του δείκτη αυτής της λέξης στον

μοναδικό ευρετήριο λέξεων ορίζεται στο 1 στο διάνυσμα εκπαίδευσης. Αυτό είναι ένα πολύ απλό μοντέλο (bag of words model)

Tensorflow

Το TensorFlow ™ είναι μια βιβλιοθήκη λογισμικού ανοιχτού κώδικα για αριθμητικούς υπολογισμούς χρησιμοποιώντας γραφήματα ροής δεδομένων. Οι κόμβοι στο γράφημα αντιπροσωπεύουν μαθηματικές λειτουργίες, ενώ οι άκρες των γραφημάτων αντιπροσωπεύουν τις πολυδιάστατες συστοιχίες δεδομένων (tensors) που επικοινωνούν μεταξύ τους. Η ευέλικτη αρχιτεκτονική επιτρέπει την ανάπτυξη υπολογισμών σε μία ή περισσότερες CPU ή GPU σε επιτραπέζιο υπολογιστή, διακομιστή ή κινητή συσκευή με ένα μόνο API. Το TensorFlow αναπτύχθηκε αρχικά από ερευνητές και μηχανικούς που εργάζονται στην Ομάδα Εγκεφάλου της Google στο ερευνητικό οργανισμό Machine Intelligence της Google για σκοπούς διερεύνησης της μηχανικής μάθησης και της έρευνας σε βαθιά νευρωνικά δίκτυα, αλλά το σύστημα είναι αρκετά γενικό ώστε να μπορεί να εφαρμοστεί σε ευρύ φάσμα άλλων τομέων καλά.

TensorBoard: Graph Visualization



3.7 Visualization of a TensorFlow graph.

Convolutional Neural Network (CNN)

Στη μηχανική μάθηση, ένα συνελικτικό νευρωνικό δίκτυο (CNN, ή ConvNet) είναι μια κλάση βαθιών τεχνητών νευρωνικών δικτύων με τροφοδοσία προς τα εμπρός που έχει εφαρμοστεί με επιτυχία στην ανάλυση οπτικών εικόνων.

Τα CNNs χρησιμοποιούν μια παραλλαγή πολλαπλών στρώσεων perceptrons που έχουν σχεδιαστεί για να απαιτούν ελάχιστη προεπεξεργασία. Είναι επίσης γνωστά ως αμεταβλητά τεχνητά νευρωνικά δίκτυα μεταβλητής μεταβλητής (SIAN), βασισμένα στην αρχιτεκτονική κοινής βαρύτητας και στα χαρακτηριστικά μεταβλητής μεταβλητότητας.

Τα συγκλινόμενα δίκτυα εμπνεύστηκαν από βιολογικές διεργασίες στις οποίες το πρότυπο σύνδεσης μεταξύ νευρώνων εμπνέεται από την οργάνωση του ζωτικού οπτικού φλοιού. Οι μεμονωμένοι φλοιώδεις νευρώνες ανταποκρίνονται σε ερεθίσματα μόνο σε μια περιορισμένη περιοχή του οπτικού πεδίου που είναι γνωστή ως το δεκτικό πεδίο. Τα δεκτικά πεδία διαφορετικών νευρώνων επικαλύπτονται εν μέρει έτσι ώστε να καλύπτουν ολόκληρο το οπτικό πεδίο.

Τα CNN χρησιμοποιούν σχετικά μικρή προεπεξεργασία σε σύγκριση με άλλους αλγόριθμους ταξινόμησης εικόνων. Αυτό σημαίνει ότι το δίκτυο μαθαίνει τα φίλτρα που κατασκευάστηκαν με παραδοσιακούς αλγόριθμους. Αυτή η ανεξαρτησία από

τις προηγούμενες γνώσεις και την ανθρώπινη προσπάθεια στον σχεδιασμό χαρακτηριστικών είναι ένα σημαντικό πλεονέκτημα.

Ένα CNN αποτελείται από ένα στρώμα εισόδου και εξόδου, καθώς και από πολλαπλά κρυμμένα στρώματα. Τα κρυμμένα στρώματα είναι συνελικτικά, συγκεντρώνονται ή συνδέονται πλήρως.

Τα περιστροφικά στρώματα εφαρμόζουν μια λειτουργία συνέλισης στην είσοδο, μεταφέροντας το αποτέλεσμα στο επόμενο στρώμα. Η συνέλιξη προσομοιώνει την απόκριση ενός μεμονωμένου νευρώνα σε οπτικά ερεθίσματα .

Κάθε συνελικτικός νευρώνας επεξεργάζεται τα δεδομένα μόνο για το δεκτικό πεδίο. Το πλακίδιο επιτρέπει στα CNN να ανέχονται τη μετάφραση της εικόνας εισόδου (π.χ. μετάφραση, περιστροφή, προοπτική παραμόρφωση)

Αν και τα πλήρως συνδεδεμένα πρωτεύοντα νευρωνικά δίκτυα μπορούν να χρησιμοποιηθούν για να μάθουν χαρακτηριστικά καθώς και για ταξινόμηση δεδομένων, δεν είναι πρακτικό να εφαρμοστεί αυτή η αρχιτεκτονική στις εικόνες. Ένας πολύ μεγάλος αριθμός νευρώνων θα ήταν απαραίτητος ακόμη και σε μια ρηχή αρχιτεκτονική (αντίθετη από βαθιά). Η λειτουργία συνέλιξης φέρνει μια λύση σε αυτό το πρόβλημα καθώς μειώνει τον αριθμό των ελεύθερων παραμέτρων, επιτρέποντας στο δίκτυο να είναι βαθύτερο με λιγότερες παραμέτρους . Με άλλα λόγια, επιλύει τα προβλήματα εξαφάνισης ή έκρηξης στην εκπαίδευση παραδοσιακών νευρωνικών δικτύων πολλαπλών στρώσεων με πολλά στρώματα με τη χρήση backpropagation.

Τα συνεργατικά δίκτυα μπορούν να περιλαμβάνουν τοπικά ή σφαιρικά στρώματα συγκέντρωσης [απαιτούμενη διαύγαση], τα οποία συνδυάζουν τις εξόδους των συστοιχιών νευρώνων σε ένα στρώμα σε έναν μόνο νευρώνα στο επόμενο στρώμα .Για παράδειγμα, η μέγιστη συγκέντρωση χρησιμοποιεί τη μέγιστη τιμή από το καθένα από ένα σύμπλεγμα νευρώνων στο προηγούμενο επίπεδο. Ένα άλλο παράδειγμα είναι η μέση συγκέντρωση, η οποία χρησιμοποιεί τη μέση τιμή από κάθε μία από μια ομάδα νευρώνων στο προηγούμενο στρώμα.

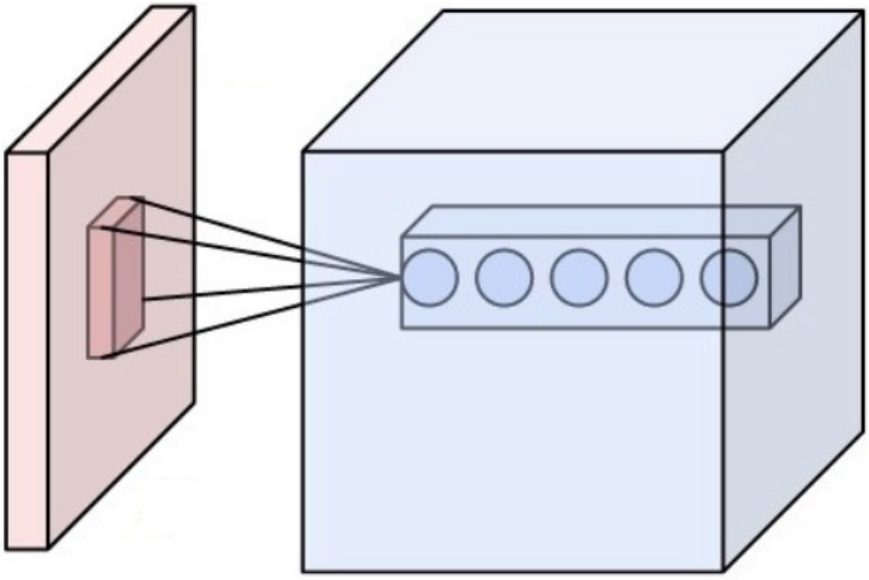
Τα πλήρως συνδεδεμένα στρώματα συνδέουν κάθε νευρώνα σε ένα στρώμα με κάθε νευρώνα σε ένα άλλο στρώμα. Είναι κατ 'αρχήν το ίδιο με το παραδοσιακό νευρωνικό δίκτυο perceptron πολλαπλών στρώσεων (MLP).

Βάρη

Τα CNN μοιράζονται τα βάρη σε συνελικτικά στρώματα, πράγμα που σημαίνει ότι το ίδιο φίλτρο (τράπεζα βάρους) χρησιμοποιείται για κάθε πεδίο δεκτών στο στρώμα. αυτό μειώνει το αποτύπωμα μνήμης και βελτιώνει την απόδοση [πώς;].

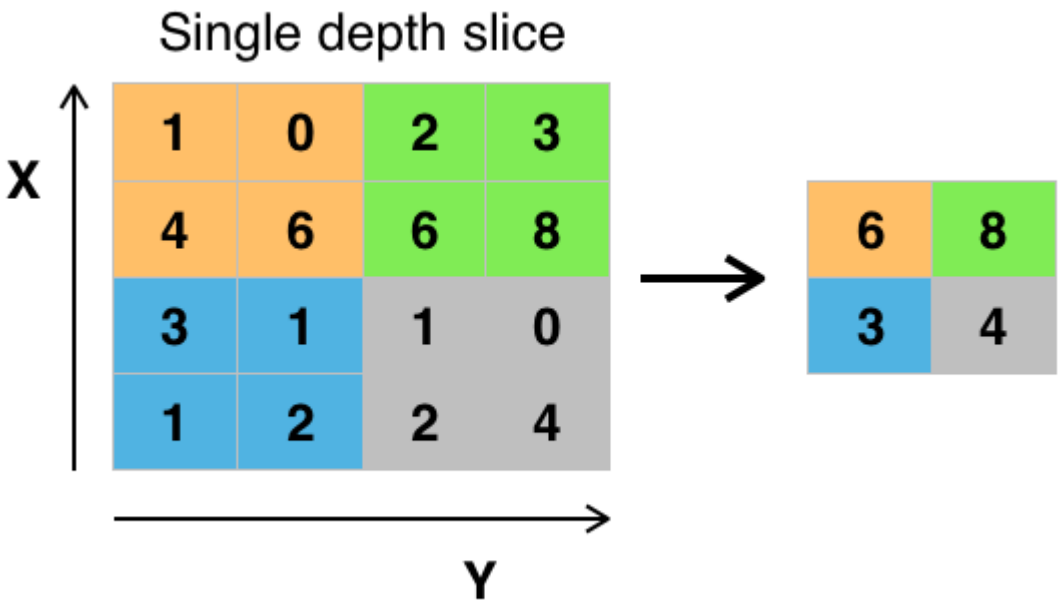
Το συνελικτικό στρώμα είναι το κεντρικό δομικό στοιχείο ενός CNN. Οι παράμετροι του στρώματος αποτελούνται από ένα σύνολο μαθηματικών φίλτρων (ή πυρήνων), τα οποία έχουν ένα μικρό πεδίο υποδοχής, αλλά εκτείνονται σε όλο το βάθος του όγκου εισόδου. Κατά τη διάρκεια της προώθησης, κάθε φίλτρο περιστρέφεται σε όλο το πλάτος και το ύψος του όγκου εισόδου, υπολογίζοντας το προϊόν κουκκίδων μεταξύ των καταχωρήσεων του φίλτρου και της εισόδου και δημιουργώντας ένα χάρτη 2-διαστάσεων ενεργοποίησης αυτού του φίλτρου. Ως αποτέλεσμα, το δίκτυο μαθαίνει φίλτρα που ενεργοποιούνται όταν ανιχνεύει κάποιο συγκεκριμένο τύπο χαρακτηριστικού σε κάποια χωρική θέση στην είσοδο.

Η στοίβαξη των χάρτων ενεργοποίησης για όλα τα φίλτρα κατά μήκος της διάστασης βάθους αποτελεί τον πλήρη όγκο εξόδου του στρώματος συνέλιξης. Κάθε είσοδος στον όγκο εξόδου μπορεί έτσι να ερμηνευτεί ως έξοδος ενός νευρώνα που κοιτάζει μια μικρή περιοχή στην είσοδο και μοιράζεται τις παραμέτρους με τους νευρώνες στον ίδιο χάρτη ενεργοποίησης.



3.8 Οι νευρώνες ενός συνελικτικού στρώματος (μπλε), συνδεδεμένοι στο δεκτικό τους πεδίο (κόκκινο)

ΣΤΡΩΜΑ ΣΥΓΚΕΝΤΡΩΣΗΣ



3.9 Max pooling με φίλτρο 2x2 και βήμα = 2

Μια άλλη σημαντική έννοια των CNN είναι η συγκέντρωση, η οποία είναι μια μορφή μη γραμμικής δειγματοληψίας. Υπάρχουν πολλές μη γραμμικές λειτουργίες για την υλοποίηση της συγκέντρωσης μεταξύ των οποίων η μέγιστη συγκέντρωση είναι η συνηθέστερη. Διαχωρίζει την εικόνα εισόδου σε ένα σύνολο μη επικαλυπτόμενων ορθογωνίων και, για κάθε τέτοια υποπεριοχή, εξάγει το μέγιστο. Η διαίσθηση είναι ότι η ακριβής τοποθεσία ενός χαρακτηριστικού είναι λιγότερο σημαντική από την ακατέργαστη θέση του σε σχέση με άλλα χαρακτηριστικά. Το στρώμα συγκέντρωσης χρησιμεύει για τη σταδιακή μείωση του χωροταξικού μεγέθους της αναπαράστασης, τη μείωση του αριθμού των παραμέτρων και του αριθμού των υπολογισμών στο δίκτυο και, ως εκ τούτου, τον έλεγχο της υπερφόρτωσης. Είναι συνηθισμένο να εισάγετε περιοδικά ένα στρώμα συγκέντρωσης μεταξύ διαδοχικών συνθετικών στρώσεων σε μια αρχιτεκτονική του CNN. Η λειτουργία συγκέντρωσης παρέχει μια άλλη μορφή μεταβλητής μεταβλητότητας.

Το στρώμα συγκέντρωσης λειτουργεί ανεξάρτητα σε κάθε φέτα βάθους της εισόδου και το μετατρέπει ξανά χωρικά. Η πιο συνηθισμένη μορφή είναι ένα στρώμα συγκέντρωσης με φίλτρα μεγέθους 2x2 που εφαρμόζεται με ένα βήμα 2 υποδειγμάτων σε κάθε διάκενο βάθους στην είσοδο κατά 2 κατά μήκος τόσο του πλάτους όσο και του ύψους, απορρίπτοντας το 75% των ενεργοποιήσεων. Σε αυτή την περίπτωση, κάθε μέγιστη λειτουργία είναι πάνω από 4 αριθμούς. Η διάσταση βάθους παραμένει αμετάβλητη.

Εκτός από τη μέγιστη συγκέντρωση, οι μονάδες συγκέντρωσης μπορούν να χρησιμοποιήσουν άλλες λειτουργίες, όπως μέση συγκέντρωση ή συγκέντρωση L2-norm. Η μέση συγκέντρωση χρησιμοποιήθηκε συχνά ιστορικά, αλλά πρόσφατα

έπαψε να είναι υπέρ της σύγκρισης με τη μέγιστη συγκέντρωση, η οποία λειτουργεί καλύτερα στην πράξη.

Λόγω της επιθετικής μείωσης του μεγέθους της αναπαράστασης, η τάση είναι προς τη χρήση μικρότερων φίλτρων ή για την απόλυτη απόρριψη της στρώσης συγκέντρωσης .

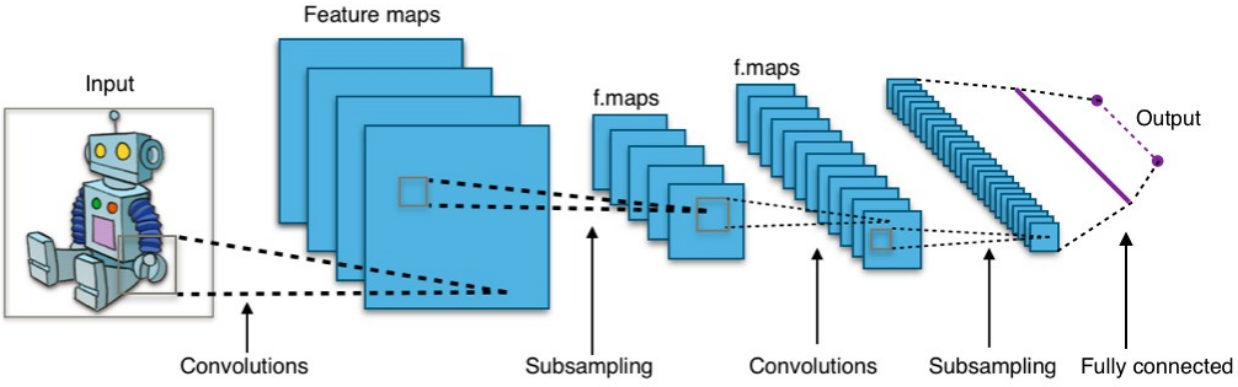
RoI συγκέντρωση σε μέγεθος 2x2. Σε αυτό το παράδειγμα η πρόταση περιοχής (μια παράμετρος εισόδου) έχει μέγεθος 7x5.
Η περιοχή συγκέντρωσης συμφερόντων (γνωστή επίσης ως συγκέντρωση RoI) είναι μια παραλλαγή της μέγιστης συγκέντρωσης, στην οποία το μέγεθος εξόδου είναι σταθερό και το παραλληλόγραμμο εισόδου είναι μια παράμετρος.

Η συγκέντρωση είναι ένα σημαντικό συστατικό των συνελικτικών νευρωνικών δικτύων για ανίχνευση αντικειμένων βασισμένο στην αρχιτεκτονική Fast R-CNN .

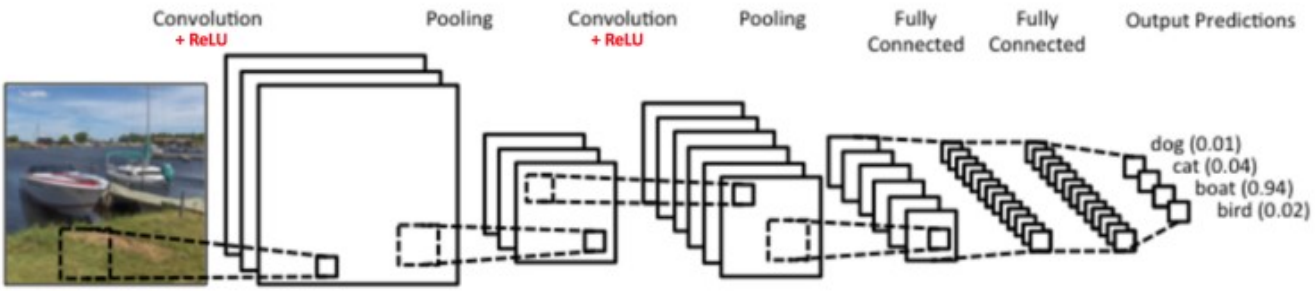
Το ReLU είναι η συντομογραφία των διορθωμένων γραμμικών μονάδων. Αυτή η στρώση εφαρμόζει τη συνάρτηση ενεργοποίησης μη-κορεσμού f (x) = max (0, x) f (x) = max (0, x). Αυξάνει τις μη γραμμικές ιδιότητες της συνάρτησης απόφασης και του συνολικού δικτύου χωρίς να επηρεάζει τα δεκτά πεδία του στρώματος συνέλιξης εφαπτομένη κορεσμού και η λειτουργία .Το ReLU από άλλες λειτουργίες, διότι εκπαιδεύει το νευρικό δίκτυο αρκετές φορές ταχύτερα χωρίς σημαντική ποινή στην ακρίβεια γενίκευσης.

Πλήρως συνδεδεμένο στρώμα

Πλήρως συνδεδεμένο στρώμα Τέλος, μετά από αρκετά στρώματα συνένωσης και μέγιστης συγκέντρωσης, ο συλλογισμός υψηλού επιπέδου στο νευρικό δίκτυο γίνεται μέσω πλήρως συνδεδεμένων στρωμάτων. Οι νευρώνες σε ένα πλήρως συνδεδεμένο στρώμα έχουν συνδέσεις σε όλες τις ενεργοποιήσεις στο προηγούμενο στρώμα, όπως φαίνεται στα κανονικά νευρωνικά δίκτυα. Επομένως, οι ενεργοποιήσεις τους μπορούν να υπολογιστούν με πολλαπλασιασμό μήτρας που ακολουθείται από μετατόπιση μεροληψίας.



3.9.1 Τυπική CNN αρχιτεκτονική



3.9.2 Παράδειγμα αρχιτεκτονικής CNN

Τα αποτελέσματα στηρίζονται στον πίνακα σύγχυσης (confusion matrix).

Predictive Model: Evaluation

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

predictive result / classification		actual result / classification: yes	actual result / classification: no
	yes	**tp** (true positive)	**fp** (false positive) ← Type 1 error
	no	**fn** (false negative)	**tn** (true negative)

$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{Recall} = \frac{tp}{tp + fn}$$

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \qquad \text{True Negative Rate} = \frac{tn}{tn + fp}$$

4.7 Confusion matrix

Υλοποίηση

Έχουμε μια λίστα απο λέξεις [καλά , οθόνης , κατασκευή , παίζω]. Και ξεκινάμε να εκπαιδεύσουμε την πρόταση <<Το κινητό έχει καλά χαρακτηριστικά οθόνης>> . Δημιουργούμε πρώτα το διάνυσμα εκπαιδεύσεως μας ώστε να είναι ένας φορέας μηδενικών που έχει το ίδιο μέγεθος με τον μοναδικό κατάλογο λέξεων. Αυτό θα ήταν ένα 1x4: [0 0 0 0]. Τώρα, επαναλαμβάνουμε μέσα από όλους τους κόσμους σε αυτή την πρόταση δείγματος και, αν βρίσκονται στον μοναδικό κατάλογο λέξεων, κάνουμε την τιμή του δείκτη στον φορέα εκπαίδευσης ίση με 1. Δεδομένου ότι η λέξη καλά (ευρετήριο: 0) και η λέξη οθόνης(δείκτης: 1) είναι στη μοναδική λίστα λέξεων, και δεν υπάρχουν άλλοι, το νέο μας διάνυσμα χαρακτηριστικών κατάρτισης είναι [1 1 0 0]. Στη συνέχεια, είτε τα δεδομένα είναι θετικά είτε αρνητικά και πάλι, εδώ, θα χρησιμοποιήσουμε μόνο μια θερμή κωδικοποίηση και έχουμε τον φορέα ετικέτας να είναι [POS, NEG], όπου τα θετικά δεδομένα είναι [1,0] και τα αρνητικά δεδομένα θα είναι [0,1].

Για την προεργασία θα χρησιμοποιηθεί η βιβλιοθήκη NLTK

Βιβλιοθήκες που χρησιμοποιήθηκαν

- **NLTK**

Το Toolkit για τη Φυσική Γλώσσα, ή πιο συχνά το NLTK, είναι μια σουίτα βιβλιοθηκών και προγραμμάτων για συμβολική και στατιστική επεξεργασία φυσικής γλώσσας (NLP) για αγγλικά γραμμένα στη γλώσσα προγραμματισμού Python. Αναπτύχθηκε από τον Steven Bird και τον Edward Loper στο Τμήμα Επιστήμης Υπολογιστών και Πληροφορίας του Πανεπιστημίου της Πενσυλβανίας . Το NLTK περιλαμβάνει γραφικές επιδείξεις και δείγματα δεδομένων. Συνοδεύεται από ένα βιβλίο που εξηγεί τις υποκείμενες έννοιες πίσω από τις εργασίες επεξεργασίας γλώσσας που υποστηρίζονται από το σύνολο εργαλείων .Το NLTK προορίζεται να υποστηρίξει την έρευνα και τη διδασκαλία σε NLP ή σε στενά συνδεδεμένους τομείς, συμπεριλαμβανομένης της εμπειρικής γλωσσολογίας, της γνωστικής επιστήμης, της τεχνητής νοημοσύνης, της ανάκτησης πληροφοριών και της μηχανικής μάθησης. Το NLTK χρησιμοποιήθηκε με επιτυχία ως εργαλείο διδασκαλίας, ως μεμονωμένο εργαλείο μελέτης, και ως πλατφόρμα για τη δημιουργία πρωτοτύπων και την κατασκευή ερευνητικών συστημάτων. Υπάρχουν 32 πανεπιστήμια στις ΗΠΑ και 25 χώρες που χρησιμοποιούν το NLTK στα μαθήματα τους. Το NLTK υποστηρίζει τις λειτουργίες ταξινόμησης, tokenization, starking, tagging, parsing και σημασιολογικής αιτιολογίας.

Την βιβλιοθήκη NLTK την χρησιμοποιούμε για να κάνουμε Λεκτική ανάλυση . Λεκτική ανάλυση είναι η διαδικασία που μετατρέπει μια ακολουθία από χαρακτήρες σε μια ακολουθία από λεκτικές μονάδες (tokens). Ένα πρόγραμμα ή συνάρτηση που κάνει λεκτική ανάλυση ονομάζεται λεκτικός αναλυτής (*lexical analyzer*, *lexer* ή *scanner*). Ένας λεκτικός αναλυτής συχνά αποτελεί μια συνάρτηση που καλείται από ένα συντακτικό αναλυτή ή κάποια άλλη συνάρτηση.

Μέσω της βιβλιοθήκης NLTK θα κάνουμε tokenization για να μπορέσουμε να χωρίσουμε το κείμενο σε λέξεις. Μετά το tokenization θα κάνουμε lemmatizer. Το

lemmatizer χρησιμοποιείται για να αντιστοιχίσει ίδιες λέξεις που έχουν παρόμοιο νόημα σε μία έτσι ώστε να δημιουργήσουμε ένα ποιοτικό λεξικό το οποίο να μην είναι μεγάλο.Όσο μεγαλύτερο τόσο λιγότερα ποιοτικό.

Για να την χρησιμοποιήσουμε καταρχάς την βιβλιοθήκη NLTK πρέπει να γράψουμε στην κονσόλα :

Import nltk
nltk.download ()

Αυτές είναι κάποιες από τις απαραίτητες εισαγωγές που πρέπει να κάνουμε.

```
import nltk
from nltk.tokenize import word_tokenize
import numpy as np
import random
import pickle
from collections import Counter
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()
hm_lines = 1000000
```

4.1 Εισαγωγή απαραίτητων μεθόδων και βιβλιοθηκών

Αυτές είναι μόνο ορισμένες απαραίτητες εισαγωγές. Το NLTK έχει εξηγηθεί,η βιβλιοθήκη numpy είναι μια βιβλιοθήκη για τη γλώσσα προγραμματισμού Python, προσθέτοντας υποστήριξη για μεγάλους πολυδιαστατους πίνακες μαζί με μια μεγάλη συλλογή μαθηματικών λειτουργιών υψηλού επιπέδου για να λειτουργούν πάνω σε αυτούς τους πίνακες.Η random θα χρησιμοποιηθεί για την ανακατάταξη των δεδομένων, ο μετρητής θα χρησιμοποιηθεί για τη διαλογή των πιο κοινών λειμμάτων και το pickle για να σώσει τη διαδικασία έτσι ώστε να μην χρειαζόμαστε να το κάνουμε κάθε φορά. Ορίζουμε τον λειματοποιητή και στη συνέχεια ορίζουμε την τιμή.

hm_lines. 100.000: η διαδικασία θα χρησιμοποιηθεί για όλες τις γραμμές των dataset.Υπάρχουν πάνω από 10.000 γραμμές.

```
def create_lexicon(pos,neg):

    lexicon = []
    with open(pos,'r',encoding = "utf8") as f:
        contents = f.readlines()
        for l in contents[:hm_lines]:
            all_words = word_tokenize(l)
            lexicon += list(all_words)

    with open(neg,'r',encoding = "utf8") as f:
        contents = f.readlines()
        for l in contents[:hm_lines]:
            all_words = word_tokenize(l)
            lexicon += list(all_words)

    lexicon = [lemmatizer.lemmatize(i) for i in lexicon]
    w_counts = Counter(lexicon)
    l2 = []
    for w in w_counts:

        if 1000 > w_counts[w] > 50:
            l2.append(w)
    print(len(l2))
    return l2
```

4.2 Κώδικας για την δημιουγία του λεξικού

Εδώ αρχίζει η λειτουργία. Δημιουργούμε μια μέθοδο η οποία δέχεται 2 παραμέτρους.Τα αρχεία με τις θετικές και αρνητικές προτάσεις. Διαβάζουμε την κάθε γραμμή. Για κάθε γραμμή συμβολίζουμε τις λέξεις και τις προσθέτουμε στο λεξικό. Σε αυτό το σημείο, το λεξικό μας είναι απλά ένας κατάλογος κάθε λέξης στα δεδομένα εκπαίδευσης. Εάν είχατε ένα τεράστιο σύνολο δεδομένων, πολύ μεγάλο για να χωρέσει στη μνήμη σας, τότε θα χρειαστεί να ρυθμίσετε την τιμή hm_lines εδώ, για να περάσετε απλώς στον πρώτο αριθμό hm_lines γραμμών ανά αρχείο. Τώρα πρέπει ακόμα να αφαιρέσουμε τα διπλότυπα. Επίσης, δεν χρειαζόμαστε πραγματικά σούπερ κοινά λόγια, ούτε πολύ ασυνήθιστα λόγια. Για παράδειγμα, λέξεις όπως "α", "και", ή "ή" δεν πρόκειται να μας δώσουν μεγάλη αξία σε αυτό το απλό μοντέλο "bag of words", οπότε δεν τις θέλουμε. Οι ασυνήθιστες λέξεις δεν πρόκειται να είναι πολύ χρήσιμες, καθώς θα ήταν πιθανότατα τόσο σπάνιες ώστε η ίδια η παρουσία τους να στρεβλώνει τα αποτελέσματα. Μπορούμε να προσπαθήσουμε να παίξουμε με αυτό για να δούμε αν είμαστε σωστοί σε αυτήν την πεποίθηση.

Τώρα θα πρέπει να δημιουργήσουμε μια μέθοδο η οποία να μπορεί να αναζητεί στο λεξικό και να μπορεί να αντιστοιχεί την λέξη από το dataset στο λεξικό μου μόλις δημιουργήσαμε.

```
def sample_handling(sample,lexicon,classification):

    featureset = []

    with open(sample,'r',encoding="utf8") as f:
        contents = f.readlines()
        for l in contents[:hm_lines]:
            current_words = word_tokenize(l.lower())
            current_words = [lemmatizer.lemmatize(i) for i in current_words]
            features = np.zeros(len(lexicon))
            for word in current_words:
                if word.lower() in lexicon:
                    index_value = lexicon.index(word.lower())
                    features[index_value] += 1

            features = list(features)
            featureset.append([features,classification])

    return featureset
```

4.3 μέθοδος για αντιστοίχηση λέξεων από dataset με λέξη στο λεξικό

Η μέθοδος αυτή δέχεται σαν παραμέτρους το δείγμα δηλαδή το dataset την λέξη και το classification το οποίο θα το δούμε παρακάτω πως θα το χρησιμοποιήσουμε.

Αυτό θα επαναληφθεί μέσω του αρχείου "δείγματος" που επιλέγουμε. Στην περίπτωσή μας, αυτό είναι το pos.txt ή το neg.txt. Περνάμε επίσης το λεξικό και την ταξινόμηση του φακέλου που περνάει. Από εδώ, tokenizes το δείγμα αρχείο με λέξη, στη συνέχεια lemmatizes τις λέξεις. Τώρα, ξεκινάμε με έναν πίνακα numpy.zeros που είναι το μήκος του λεξικού. Τώρα αρχίζουμε να επαναλαμβάνουμε τις λεγόμενες λέξεις προσθέτοντας 1 στην τιμή του δείκτη στον πίνακα χαρακτηριστικών που είναι ο ίδιος δείκτης της λέξης στο λεξικό. Από εδώ, εφαρμόζουμε αυτό στο σύνολο των χαρακτηριστικών μας. Όταν τελειώσετε, επιστρέφουμε ολόκληρο το πράγμα. Αυτή η λειτουργία θα εκτελεστεί δύο φορές. μία φορά για τα θετικά και μία για τα αρνητικά.

Η συνάρτηση create_feature_sets_and_labels χρησιμοποιείται για να συνοψήσουμε όλα όσα έχουμε αναφέρει μέχρι στιγμής. Δημιουργούμε το λεξικό εδώ με βάση τα δεδομένα ωμής δειγματοληψίας που έχουμε, τότε δημιουργούμε τα πλήρη χαρακτηριστικά με βάση τα αρχεία τους, το λεξικό και μετά τις ταξινομήσεις.Στη συνέχεια, θέλουμε να ανακατέψουμε αυτά τα δεδομένα, να μετατρέψουμε σε έναν πολύπλοκο πίνακα και στη συνέχεια να δημιουργήσουμε τα σύνολα εκπαίδευσης και δοκιμών. Από εδώ, επιστρέφουμε τα δεδομένα σε μεμονωμένες μεταβλητές. Τώρα είμαστε έτοιμοι να προχωρήσουμε και να προσπαθήσουμε να το τρέξουμε αυτό.

```
def create_feature_sets_and_labels(pos,neg,test_size = 0.1):
    lexicon = create_lexicon(pos,neg)
    features = []
    features += sample_handling('pos.txt',lexicon,[1,0])
    features += sample_handling('neg.txt',lexicon,[0,1])
    random.shuffle(features)
    features = np.array(features)

    testing_size = int(test_size*len(features))

    train_x = list(features[:,0][:-testing_size])
    train_y = list(features[:,1][:-testing_size])
    test_x = list(features[:,0][-testing_size:])
    test_y = list(features[:,1][-testing_size:])

    return train_x,train_y,test_x,test_y


if __name__ == '__main__':
    train_x,train_y,test_x,test_y = create_feature_sets_and_labels('neg.txt','pos.txt')

    with open('sentiment_set.pickle','wb') as f:
        pickle.dump([train_x,train_y,test_x,test_y],f)
```

Αυτό που πρέπει να κάνουμε τώρα είναι να μπορέσουμε να φορτώσουμε τα αρχεία μας.Θα εισάγουμε από το προηγούμενο αρχείο μας την create_feature_sets_and_labels.

```
import tensorflow as tf
import numpy as np
from create_sentiment_featuresets import create_feature_sets_and_labels
```

Στην συνέχεια θα δώσουμε την τιμή 400 το σύνολο των κόμβων που θα δεχθεί το στρώμα

Το μέγεθος παρτίδας(batch_size) ορίζει τον αριθμό των δειγμάτων που πρόκειται να διαδοθούν μέσω του δικτύου.

Για παράδειγμα, ας πούμε ότι έχετε 1050 δείγματα εκπαίδευσης και θέλετε να ρυθμίσετε το batch_size ίσο με 100. Ο αλγόριθμο παίρνει πρώτα 100 δείγματα (από το 1ο έως το 100ο) από το σύνολο δεδομένων εκπαίδευσης και το δίκτυο εκπαίδευσης. Στη συνέχεια παίρνει τα δεύτερα 100 δείγματα (από 101η έως 200η) και πάλι το δίκτυο εκπαίδευσης.

```
train_x,train_y,test_x,test_y = create_feature_sets_and_labels('pos.txt','neg.txt')
```

```
nodes_hl1 = 400
nodes_hl2 = 400
nodes_hl3 = 400
#nodes_hl4 = 1000

n_classes = 2;

batch_size = 50;

x = tf.placeholder('float',[None,len(train_x[0])]);
y = tf.placeholder('float');
```

Εισάγουμε στο χ ένα σύμβολο θέσης που θα τροφοδοτείται πάντα από το λεξικό αυτό κάθε φορά αυξάνεται κατά 1

```
Tensor("Placeholder:0", shape=(?, 423), dtype=float32)
```

Τώρα θα δημιουργήσουμε το cnn model με παράμετρο το (χ).Το μοντέλο περιλαμβάνει 3 κρυφά στρώματα.όπου η έξοδος των νευρώνων από το 1^0 στρώμα γίνεται είσοδος στο $2^ο$ κρυφό στρώμα και η έξοδος του $2^ο$ στρώματος γίνεται είσοδος στο $3^ο$ κρυφό στρώμα και εν συνεχεία η έξοδος του είναι το στρώμα εξόδου .

Matmul: παίρνει 2 πίνακες ως ορίσματα και επιστρέφει το γινόμενο 2 πινάκων

Relu:

Ορίζεται ως f (x) = max (0, x) επομένως δεν είναι διαφοροποιήσιμο.
Το παράγωγο του ReLU είναι πολύ απλό! Απλούστερο από το sigmoid, το οποίο είναι x(1-x).1 εάν x> 0 Αλλιώς 0.

Είναι η απλούστερη μη γραμμική συνάρτηση που χρησιμοποιούμε κυρίως σε κρυμμένα στρώματα.

```
def cnn_model(data):
    hl1_layer = {'weights':tf.Variable(tf.random_normal([len(train_x[0]),nodes_hl1])),
                 'biases': tf.Variable(tf.random_normal([nodes_hl1]))}

    hl2_layer = {'weights':tf.Variable(tf.random_normal([nodes_hl1,nodes_hl2])),
                 'biases': tf.Variable(tf.random_normal([nodes_hl2]))}

    hl3_layer = {'weights':tf.Variable(tf.random_normal([nodes_hl2,nodes_hl3])),
                 'biases': tf.Variable(tf.random_normal([nodes_hl3]))}

   # hl4_layer = {'weights':tf.Variable(tf.random_normal([nodes_hl3,nodes_hl4])),
            #     'biases': tf.Variable(tf.random_normal([nodes_hl4]))}


    output_l = {'weights':tf.Variable(tf.random_normal([nodes_hl3,n_classes])),
                 'biases': tf.Variable(tf.random_normal([n_classes])),}


    l1 = tf.add(tf.matmul(data,hl1_layer['weights']),hl1_layer['biases'])
    l1 = tf.nn.relu(l1)

    l2 = tf.add(tf.matmul(l1,hl2_layer['weights']),hl2_layer['biases'])
    l2 = tf.nn.relu(l2)

    l3 = tf.add(tf.matmul(l2,hl3_layer['weights']),hl3_layer['biases'])
    l3 = tf.nn.relu(l3)

   # l4 = tf.add(tf.matmul(l3,hl4_layer['weights']),hl4_layer['biases'])
   # l4 = tf.nn.relu(l4)

    output = tf.matmul(l3,output_l['weights']) + output_l['biases']

    return output
```

```
def train_text_classification(x):
    prediction = cnn_model(x)
    cost = tf.reduce_mean( tf.nn.softmax_cross_entropy_with_logits(logits=prediction,labels=y)
    optimazer = tf.train.AdamOptimizer().minimize(cost)

    hm_epochs = 200

    with tf.Session() as sess:
        sess.run(tf.global_variables_initializer())

        for epoch in range(hm_epochs):
            epoch_loss = 0

            i = 0;
            while i< len(train_x):

                start = i;
                end = i + batch_size;


                batch_x = np.array(train_x[start:end])
                batch_y = np.array(train_y[start:end])


                _,c = sess.run([optimazer,cost] , feed_dict={x:batch_x , y:batch_y})
                epoch_loss+=c
                i += batch_size

            correct = tf.equal(tf.arg_max(prediction,1),tf.arg_max(y,1))
```

Και τώρα θα εκπαιδεύσουμε το μοντέλο.η μέθοδος αυτή θα δεχτεί ως παράμετρο το χ και ξεκινάει η εκπαίδευση.κάνουμε prediction. Prediction είναι μια ποικιλία στατιστικών τεχνικών από την προγνωστική μοντελοποίηση, την εκμάθηση μηχανών και την εξόρυξη δεδομένων, που αναλύουν τα τρέχοντα και τα ιστορικά δεδομένα για να κάνουν προβλέψεις για μελλοντικά ή άγνωστα συμβάντα.

Optimizer είναι ο βελτιστοποιητής κλίσης. Δείχνει πώς κάποιος δημιουργεί μια παρουσία της βασικής κλάσης βελτιστοποίησης. Η τεκμηρίωση της κατηγορίας βάσης βελτιστοποίησης εξηγεί τι κάνουν οι μέθοδοι.

Παρατηρήστε εδώ ότι χρησιμοποιούμε μόνο το i για να επαναλάβουμε τα δεδομένα και η σειρά μας είναι όπου και αν είμαστε με i, στην τιμή i + batch_size. Αυτό είναι πραγματικά το μόνο που πρέπει να κάνουμε για να κάνουμε παρτίδες. Τώρα, με αυτές τις παρτίδες, κάνουμε ό, τι άλλο είναι το ίδιο.

```
accuracy = tf.reduce_mean(tf.cast(correct,'float'))

argmax_prediction = tf.argmax(prediction, 1)
argmax_y = tf.argmax(y, 1)

TP = tf.count_nonzero(argmax_prediction * argmax_y, dtype=tf.float32)
TN = tf.count_nonzero((argmax_prediction - 1) * (argmax_y - 1), dtype=tf.float32)
FP = tf.count_nonzero(argmax_prediction * (argmax_y - 1), dtype=tf.float32)
FN = tf.count_nonzero((argmax_prediction - 1) * argmax_y, dtype=tf.float32)

precision = (TP / (TP+FP))
recall = (TP / TP+FN)
F1_score  = ( 2 * (precision * recall) / (precision + recall))
#accuracy2 = (TP+TN)/(TP+TN+FP+FN)
miss_rate = (FN/(FN+TP))
FOR = (FN/(FN+TN))

print('EPOCH' , epoch , 'completed out of' ,hm_epochs , 'loss:',epoch_loss )


print('Accuracy:', accuracy.eval({x:test_x, y:test_y}),'%')
#print('Accuracy 2:',accuracy2.eval({x:test_x, y:test_y}),'%')
print('Precision:',precision.eval({x:test_x, y:test_y}),'%')
print('Recall:',recall.eval({x:test_x, y:test_y}),'%')
print('F1_score:',F1_score.eval({x:test_x, y:test_y}))
print('Miss Rate:',miss_rate.eval({x:test_x, y:test_y}))
print('False omission Rate:',FOR.eval({x:test_x, y:test_y}))
```

Σε αυτόν τον κώδικα πέρνουμε τις μετρήσεις μας. Συγκεκριμένα για accuracy, Precision, Recall, F1_score, Miss Rate, False omission Rate

tf.argmax: Επιστρέφει τον δείκτη με τη μεγαλύτερη τιμή κατά μήκος του άξονα ενός τανυστή

Tf.reduce_mean Υπολογίζει τον μέσο όρο των στοιχείων στις διαστάσεις του τανυστή.

tf.cast: Κατατάσσει έναν τανυστή σε νέο τύπο.

tf.equal: Επιστρέφει την τιμή αλήθειας του (x == y) element-wise. Επαναφέρει την τιμή ακεραιότητας του (x == y) element-wise.

Στατιστικά στοιχεία και αποτελέσματα

Η ακρίβεια (επίσης αποκαλούμενη θετική τιμή πρόβλεψης) είναι το κλάσμα των σχετικών περιπτώσεων μεταξύ των ανακτημένων περιπτώσεων, ενώ η ανάκληση (επίσης γνωστή ως ευαισθησία) είναι το κλάσμα των σχετικών περιπτώσεων που έχουν ανακτηθεί στο σύνολο των σχετικών περιπτώσεων. Επομένως, τόσο η ακρίβεια όσο και η ανάκληση βασίζονται σε κατανόηση και μέτρηση της συνάφειας.

Precision

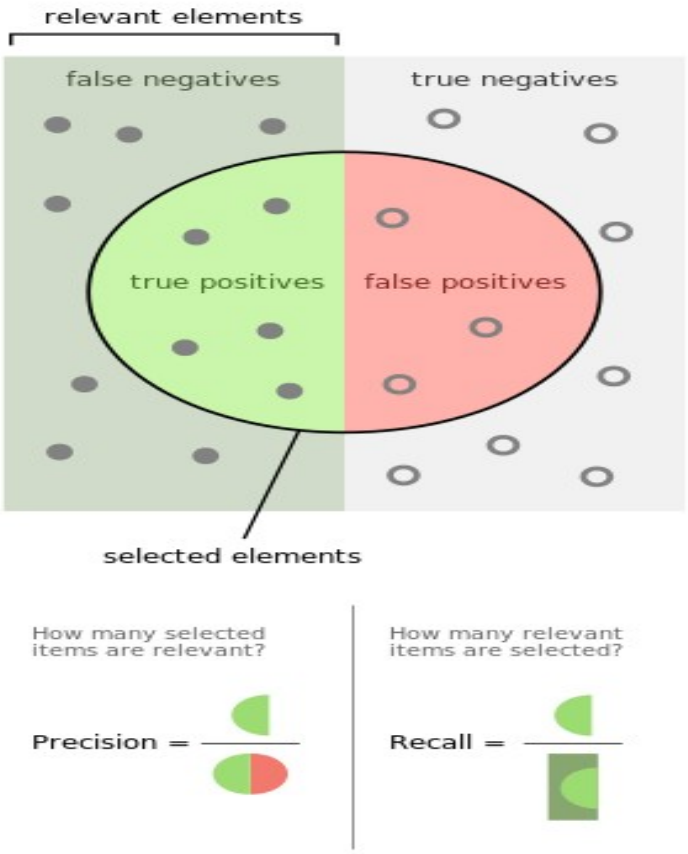
Recall

$$\text{Precision} = \frac{tp}{tp + fp}$$

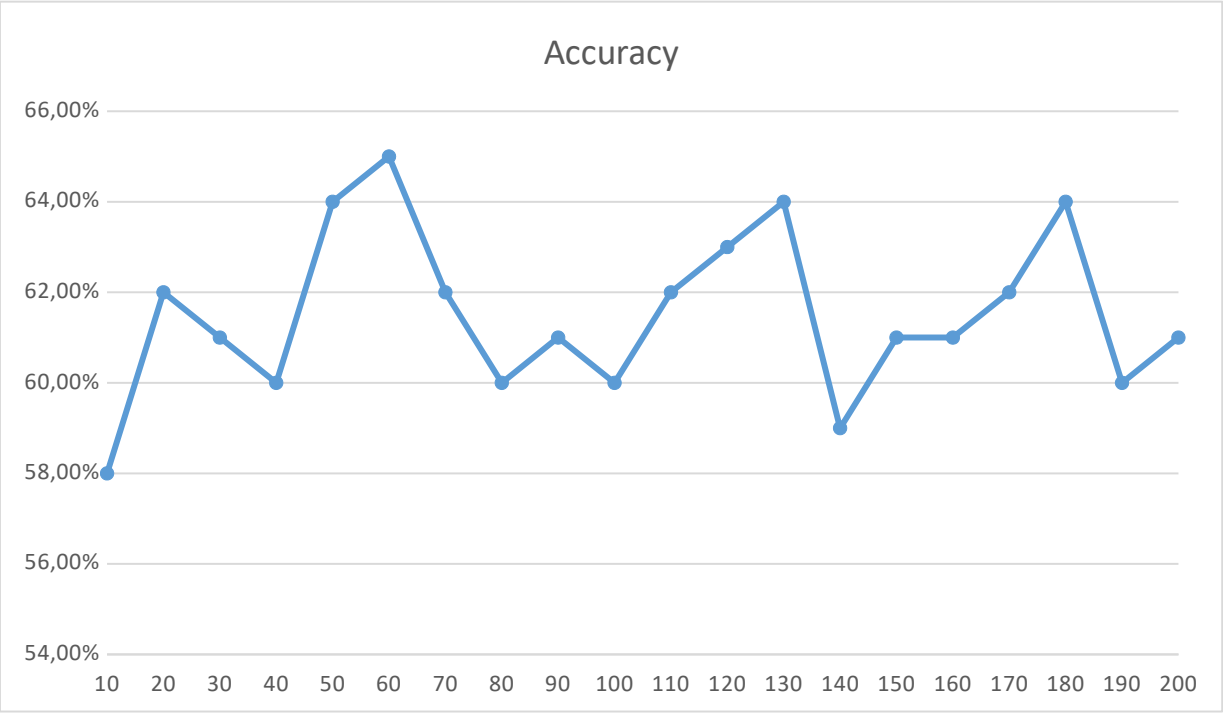
$$\text{Recall} = \frac{tp}{tp + fn}$$

Ένα μέτρο που συνδυάζει την ακρίβεια και την ανάκληση είναι ο αρμονικός μέσος όρος ακρίβειας και ανάκλησης, το παραδοσιακό μέτρο F ή το ισορροπημένο αποτέλεσμα F:

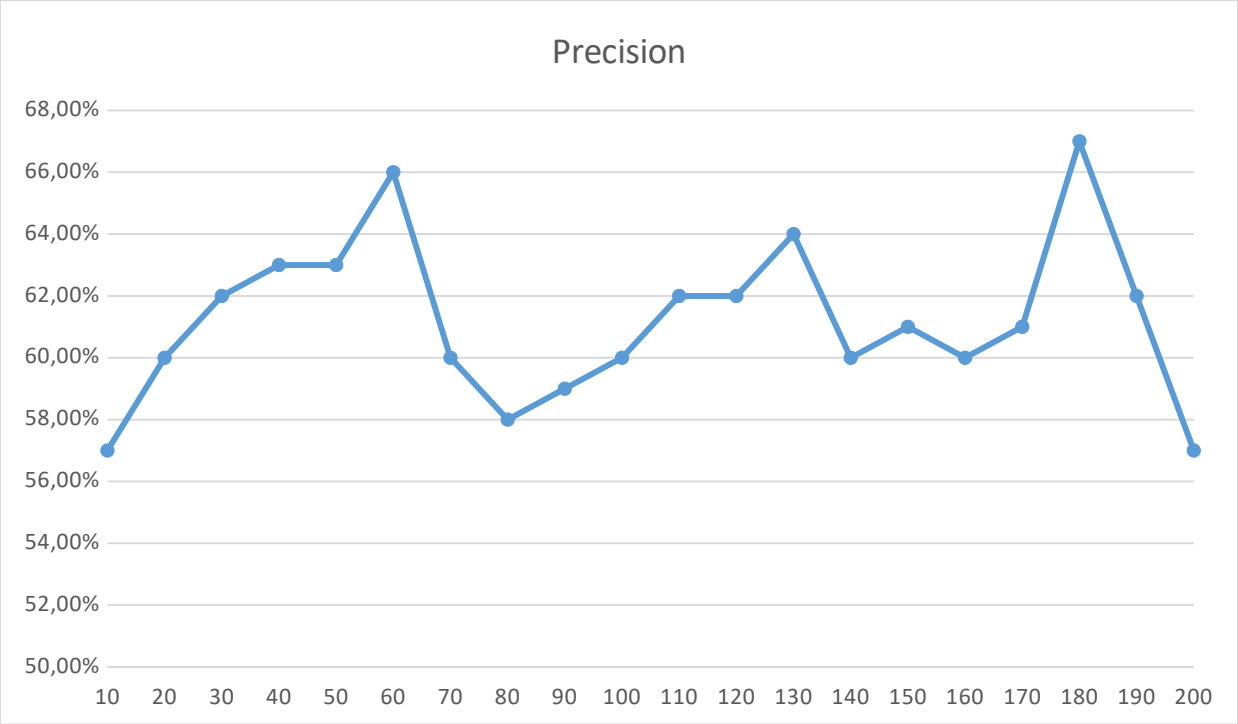
$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$



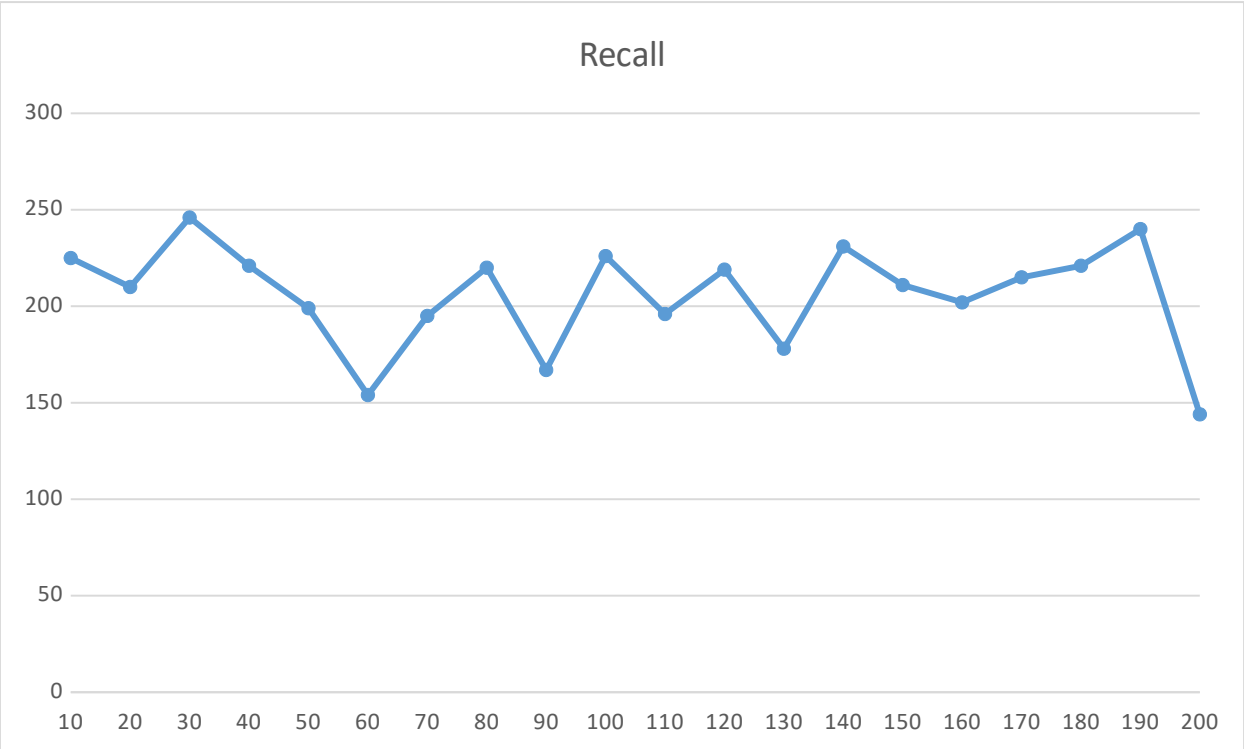
Accuracy: περίπου 67%



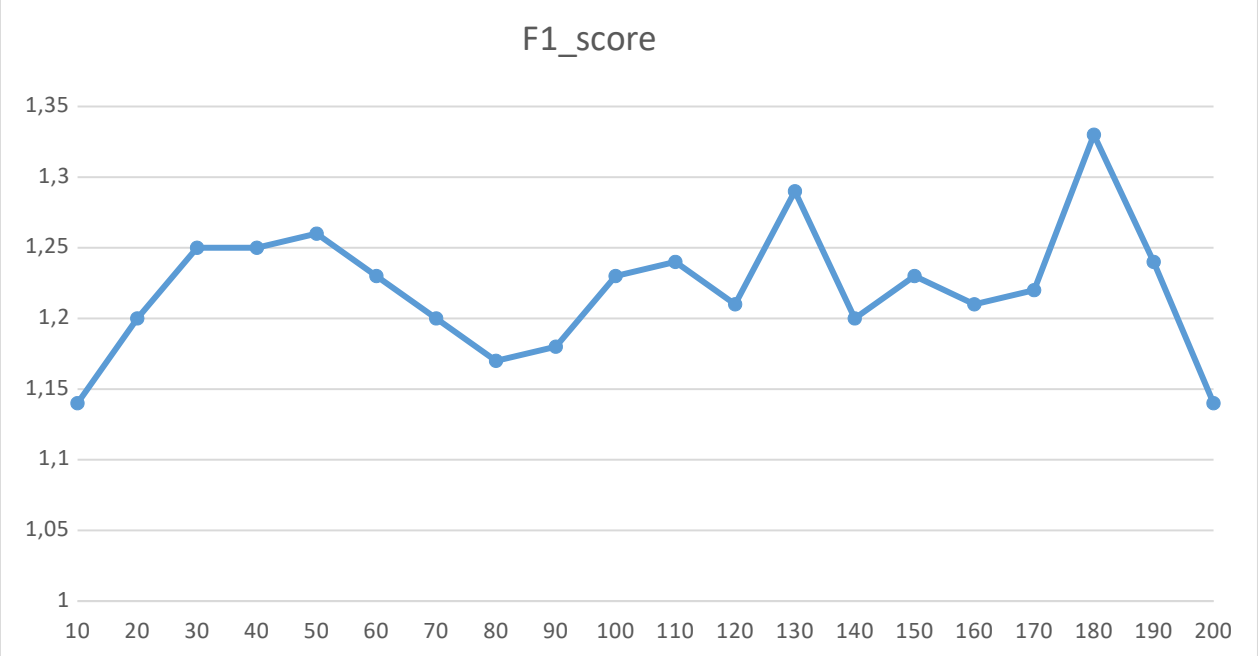
Precision: περίπου 67%



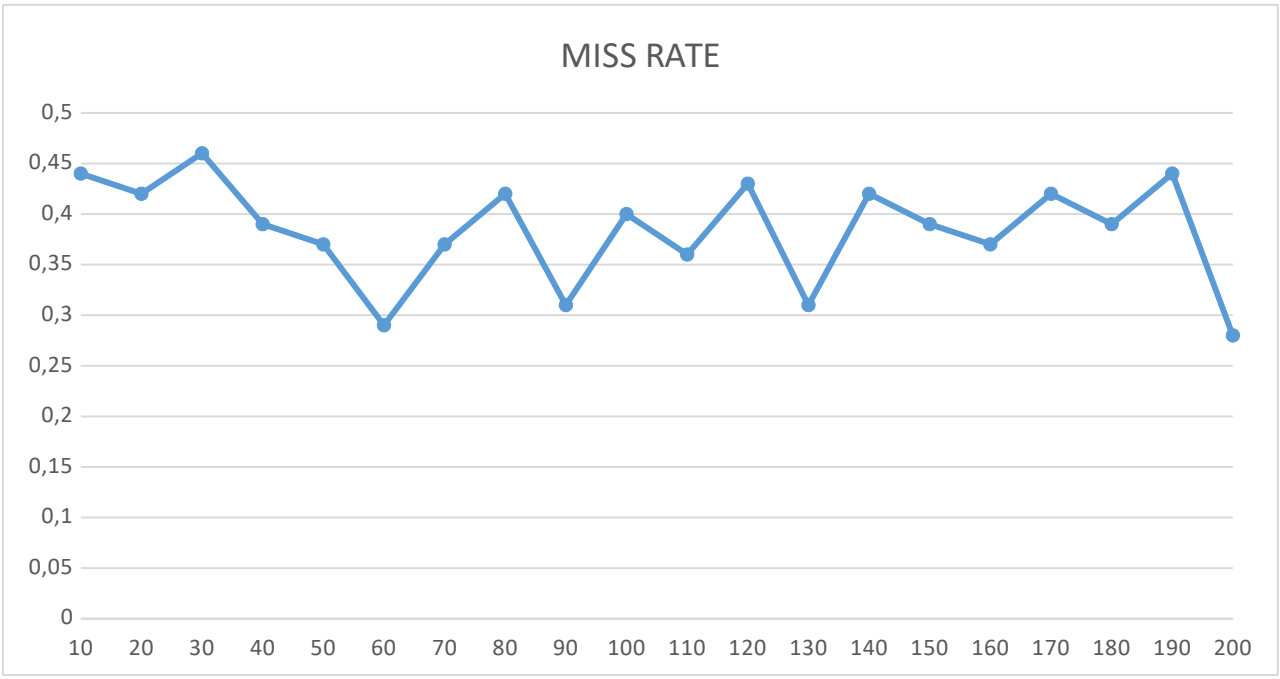
Recall



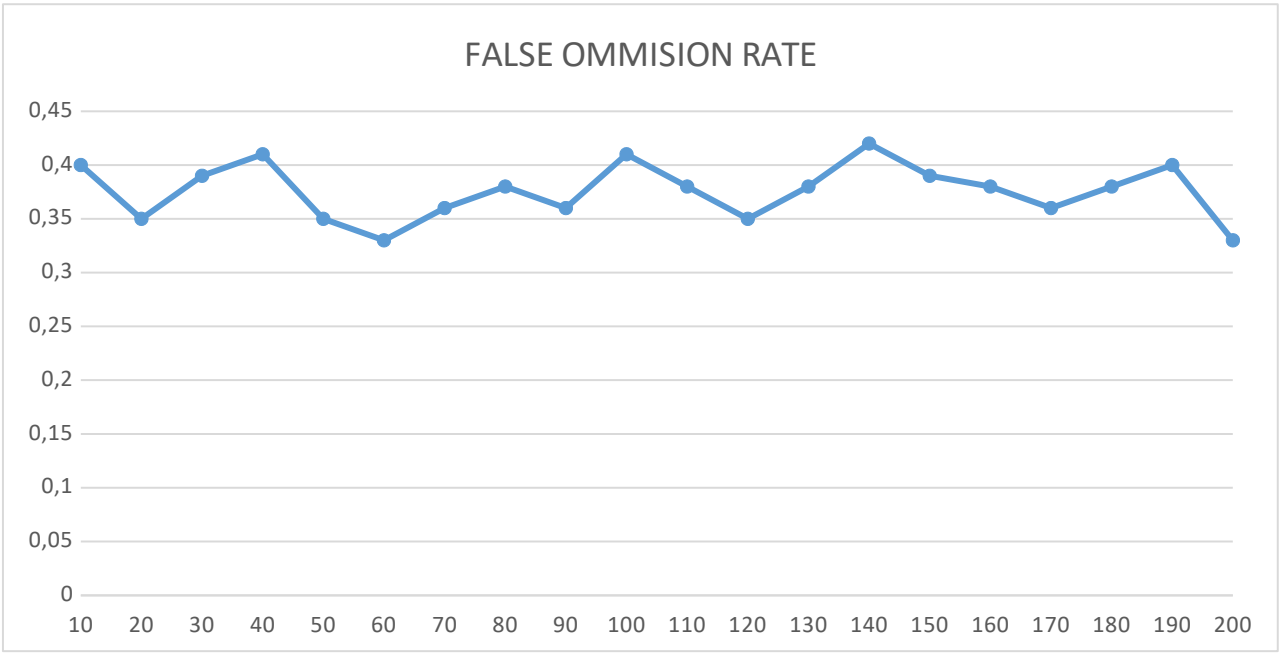
F1_score



MISS RATE



FALSE OMMISION RATE



Μοντέλο Word2vec Gensim

Το Word2vec είναι μια ομάδα σχετικών μοντέλων που χρησιμοποιούνται για την παραγωγή ενσωματωμένων λέξεων .Αυτά τα μοντέλα είναι ρηχά, δίφυλλα νευρωνικά δίκτυα που εκπαιδεύονται για να ανοικοδομήσουν τα γλωσσικά πλαίσια των λέξεων. Το Word2vec παίρνει ως είσοδο ένα μεγάλο κορμό του κειμένου και παράγει ένα χώρο διανύσματος , τυπικά από αρκετές εκατοντάδες διαστάσεις , με κάθε μοναδική λέξη στο σώμα να αντιστοιχεί σε έναν αντίστοιχο φορέα στον χώρο. Οι φορείς λέξης τοποθετούνται στον χώρο διανυσμάτων έτσι ώστε οι λέξεις που μοιράζονται κοινά πλαίσια στο σώμα να βρίσκονται σε στενή εγγύτητα μεταξύ τους στο χώρο.
Το Word2vec δημιουργήθηκε από μια ομάδα ερευνητών με επικεφαλής τον Tomas Mikolov στο Google . Ο αλγόριθμος αναλύθηκε στη συνέχεια και εξηγήθηκε από άλλους ερευνητές.Οι φορείς ενσωμάτωσης που δημιουργήθηκαν χρησιμοποιώντας τον αλγόριθμο Word2vec έχουν πολλά πλεονεκτήματα σε σύγκριση με παλαιότερους αλγορίθμους όπως η Latent Semantic Analysis .

Αλγόριθμος εκπαίδευσης

Ένα μοντέλο Word2vec μπορεί να εκπαιδευτεί με ιεραρχική softmax ή και αρνητική δειγματοληψία. Για να προσεγγιστεί η πιθανότητα καταγραφής υπό όρους που ένα μοντέλο επιδιώκει να μεγιστοποιήσει, η ιεραρχική μέθοδος softmax χρησιμοποιεί ένα δέντρο Huffman για να μειώσει τον υπολογισμό. Η αρνητική μέθοδος δειγματοληψίας, από την άλλη πλευρά, προσεγγίζει το πρόβλημα μεγιστοποίησης ελαχιστοποιώντας την πιθανότητα καταγραφής των δειγμάτων αρνητικών περιπτώσεων. Σύμφωνα με τους συγγραφείς, η ιεραρχική softmax λειτουργεί καλύτερα για σπάνιες λέξεις, ενώ η αρνητική δειγματοληψία λειτουργεί καλύτερα για συχνές λέξεις και καλύτερα με διανύσματα χαμηλής διαστάσεως. Καθώς οι εποχές εκπαίδευσης αυξάνονται, η ιεραρχική softmax σταματά να είναι χρήσιμη.

Υποδειγματοληψία

Οι λέξεις υψηλής συχνότητας συχνά παρέχουν ελάχιστες πληροφορίες. Μπορούν να υπογραμμιστούν λέξεις με συχνότητα πάνω από ένα συγκεκριμένο όριο για να αυξηθεί η ταχύτητα εκπαίδευσης.

Διαστάσεις

Η ποιότητα της ενσωμάτωσης λέξεων αυξάνεται με μεγαλύτερη διαστασιολόγηση. Αλλά αφού φτάσει σε κάποιο σημείο, το οριακό κέρδος θα μειωθεί. Συνήθως, η διαστατικότητα των διανυσμάτων ορίζεται μεταξύ 100 και 1.000.

Παράθυρο πλαισίου

Το μέγεθος του παραθύρου περιβάλλοντος καθορίζει πόσες λέξεις πριν και μετά από μια δεδομένη λέξη θα συμπεριληφθούν ως λέξεις-κλειδιά της λέξης. Σύμφωνα με τη σημείωση των συγγραφέων, η συνιστώμενη τιμή είναι 10 για skip-gram και 5 για CBOW.

Παράμετροι και ποιότητα μοντέλου

Η χρήση διαφορετικών παραμέτρων μοντέλου και διαφορετικών μεγεθών σωμάτων μπορεί να επηρεάσει σημαντικά την ποιότητα ενός μοντέλου word2vec. Η ακρίβεια μπορεί να βελτιωθεί με διάφορους τρόπους, συμπεριλαμβανομένης της επιλογής της αρχιτεκτονικής μοντέλου (CBOW ή Skip-Gram), αύξηση του συνόλου δεδομένων εκπαίδευσης, αύξηση του αριθμού των διανυσματικών διαστάσεων και αύξηση του μεγέθους του παραθύρου των λέξεων που εξετάζονται από τον αλγόριθμο. Κάθε μία από αυτές τις βελτιώσεις έρχεται με το κόστος της αυξημένης υπολογιστικής πολυπλοκότητας και ως εκ τούτου την αύξηση του χρόνου παραγωγής του μοντέλου.
Στα μοντέλα που χρησιμοποιούν μεγάλα σωματίδια και μεγάλο αριθμό διαστάσεων, το μοντέλο skip-gram αποδίδει την υψηλότερη συνολική ακρίβεια και παράγει με συνέπεια την υψηλότερη ακρίβεια στις σημασιολογικές σχέσεις, ενώ παράλληλα δίνει την υψηλότερη συντακτική ακρίβεια στις περισσότερες περιπτώσεις. Ωστόσο, το CBOW είναι λιγότερο υπολογιστικά δαπανηρό και δίνει παρόμοια αποτελέσματα ακρίβειας.
Η ακρίβεια αυξάνεται συνολικά καθώς αυξάνεται ο αριθμός των χρησιμοποιούμενων λέξεων και καθώς ο αριθμός των διαστάσεων αυξάνεται. Ο Mikolov αναφέρει ότι ο διπλασιασμός του ποσού των δεδομένων εκπαίδευσης οδηγεί σε ισοδύναμη αύξηση της υπολογιστικής πολυπλοκότητας διπλασιάζοντας τον αριθμό διαστάσεων του φορέα.

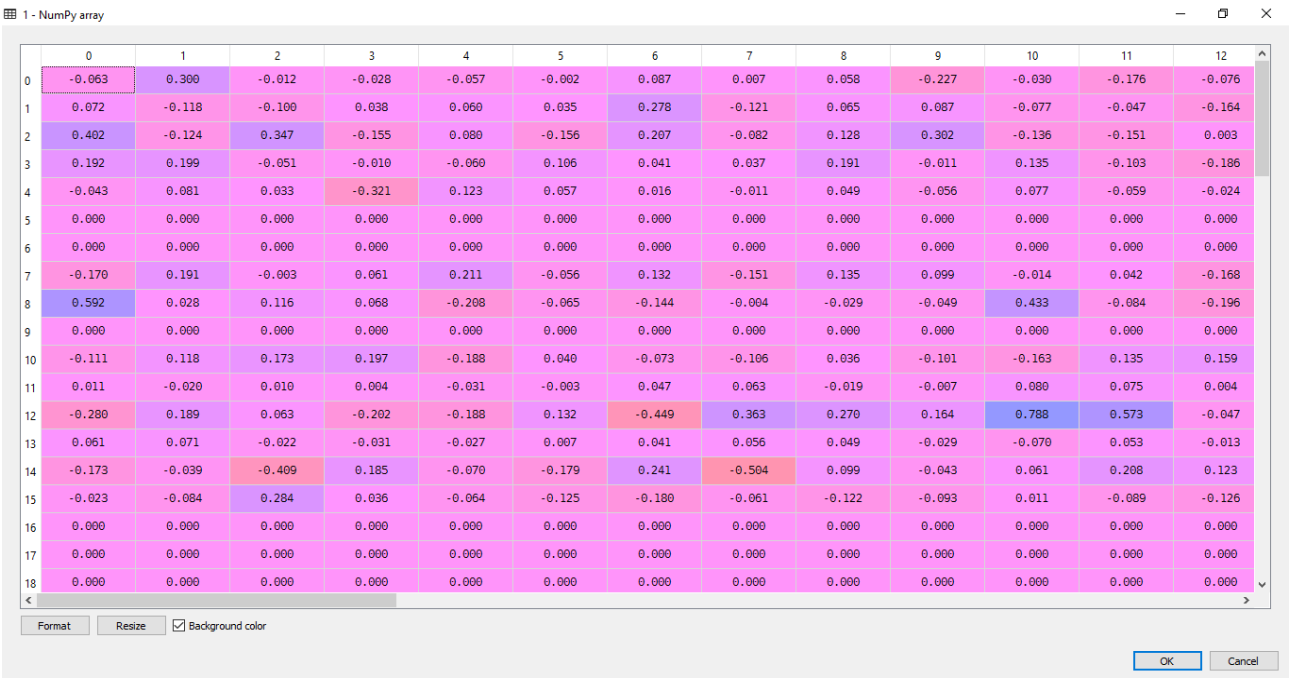
ΘΑ χρησιμοποιήσουμε ένα μοντέλο που έχει εκπαιδευτεί σε word2vec με size= 300 από εταιρία στην θεσσαλονίκη(msensis) σε συνεργασία με τον επιβλέποντα καθηγητή μου.

```
Created on Tue Sep 12 22:39:11 2017

@author: Raphael
"""

from gensim.models.word2vec import Word2Vec
import numpy as N

from gensim import utils
from gensim.models.doc2vec import LabeledSentence
from gensim.models import Doc2Vec
from gensim.models.word2vec import LineSentence


#Author Kermizidis Rafail
```

Φορτώνουμε τις εξής μεθόδους από την βιβλιοθήκη genism

Βήματα ηλοποίησης

- Φορτώνουμε το μοντέλο μέσω της μεθόδου Word2Vec.load()
- Δημιουργούμε έναν πίνακα κενό(θα κάνουμε append τα vectors)
- Δημιουργούμε έναν πίνακα 1χ300.Θα τον χρειαστούμε διότι αν μια λέξη δεν υπάρχει στο μοντέλο τότε η τιμή θα είναι μηδενικά σε όλο το διάνυσμα
- Διαβάζουμε το αρχείο κειμένου στο οποίο έχει γίνει lemmatizer (το οποίο δόθηκε από τον επιβλέποντα)
- Διαβάζουμε τις γραμμές
- Για κάθε γραμμή δημιουργούμε έναν πίνακα 80χ300 γιατί οι προτάσεις διαφέρουν σε μέγεθος(πχ μια πρόταση μπορεί να αποτελείται από 30 λέξεις ένω μια άλλη από 65 λέξεις) οπότε βάζω max 80
- Χωρίζω τις λέξεις από την κάθε γραμμή
- Για κάθε λέξη που έχει η γραμμή
- Ενάν η λέξη υπάρχει στο μοντέλο τότε δώσε μου το διάνυσμα της λέξης και κάνε append στο array 80x300 (αυτο θα δημιουργείται για κάθε πρόταση)
- Αν δεν υπάρχει τότε δώσε ως τιμή 1χ300 όλα μηδενικά
- Append στον πίνακα που δημιουγήσαμε
- Κλείσιμο αρχείου



	0	1	2	3	4	5	6	7	8	9	10	11	12
0	-0.063	0.300	-0.012	-0.028	-0.057	-0.002	0.087	0.007	0.058	-0.227	-0.030	-0.176	-0.076
1	0.072	-0.118	-0.100	0.038	0.060	0.035	0.278	-0.121	0.065	0.087	-0.077	-0.047	-0.164
2	0.402	-0.124	0.347	-0.155	0.080	-0.156	0.207	-0.082	0.128	0.302	-0.136	-0.151	0.003
3	0.192	0.199	-0.051	-0.010	-0.060	0.106	0.041	0.037	0.191	-0.011	0.135	-0.103	-0.186
4	-0.043	0.081	0.033	-0.321	0.123	0.057	0.016	-0.011	0.049	-0.056	0.077	-0.059	-0.024
5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
7	-0.170	0.191	-0.003	0.061	0.211	-0.056	0.132	-0.151	0.135	0.099	-0.014	0.042	-0.168
8	0.592	0.028	0.116	0.068	-0.208	-0.065	-0.144	-0.004	-0.029	-0.049	0.433	-0.084	-0.196
9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
10	-0.111	0.118	0.173	0.197	-0.188	0.040	-0.073	-0.106	0.036	-0.101	-0.163	0.135	0.159
11	0.011	-0.020	0.010	0.004	-0.031	-0.003	0.047	0.063	-0.019	-0.007	0.080	0.075	0.004
12	-0.280	0.189	0.063	-0.202	-0.188	0.132	-0.449	0.363	0.270	0.164	0.788	0.573	-0.047
13	0.061	0.071	-0.022	-0.031	-0.027	0.007	0.041	0.056	0.049	-0.029	-0.070	0.053	-0.013
14	-0.173	-0.039	-0.409	0.185	-0.070	-0.179	0.241	-0.504	0.099	-0.043	0.061	0.208	0.123
15	-0.023	-0.084	0.284	0.036	-0.064	-0.125	-0.180	-0.061	-0.122	-0.093	0.011	-0.089	-0.126
16	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
17	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
18	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

p - List (2520 elements)

Index	Type	Size	
0	float64	(80, 300)	array([[0. , 0. , 0. , ..., 0. , 0. ...
1	float64	(80, 300)	array([[-0.06271242, 0.29980266, -0.01186956, ..., 0.07682225, 0.2802 ...
2	float64	(80, 300)	array([[-0.10801255, -0.00639449, 0.02545316, ..., 0.07658723, -0.0240 ...
3	float64	(80, 300)	array([[-0.06271242, 0.29980266, -0.01186956, ..., 0.07682225, 0.2802 ...
4	float64	(80, 300)	array([[-0.22615868, -0.71884167, -0.33257911, ..., -0.18185058, 0.1150 ...
5	float64	(80, 300)	array([[0.0145386 , 0.07841848, 0.09937925, ..., -0.0492854 , -0.0111 ...
6	float64	(80, 300)	array([[0.03402223, -0.01265975, 0.17039065, ..., -0.07448008, -0.1070 ...
7	float64	(80, 300)	array([[0.00727892, 0.00766001, -0.06748195, ..., -0.39962199, -0.0565 ...
8	float64	(80, 300)	array([[0.24569324, 0.22555979, 0.13058084, ..., -0.07368037, -0.2333 ...
9	float64	(80, 300)	array([[0.41523463, -0.0194673 , -0.01976342, ..., -0.07741632, -0.0128 ...
10	float64	(80, 300)	array([[-0.06271242, 0.29980266, -0.01186956, ..., 0.07682225, 0.2802 ...
11	float64	(80, 300)	array([[0.01874935, 0.01654268, 0.20776652, ..., 0.02541579, 0.0309 ...
12	float64	(80, 300)	array([[-0.09465854, 0.06671555, -0.16259922, ..., -0.20592405, 0.1359 ...
13	float64	(80, 300)	array([[-0.19004087, 0.34245244, -0.04628177, ..., -0.1909374 , 0.0947 ...
14	float64	(80, 300)	array([[-0.00814763, 0.1134551 , 0.17975131, ..., 0.06468888, -0.0399 ...
15	float64	(80, 300)	array([[-0.24339798, 0.14638294, 0.10543439, ..., 0.21854979, 0.0880 ...
16	float64	(80, 300)	array([[0.20511919, 0.03662315, 0.40390655, ..., 0.00826834, -0.0256 ...
17	float64	(80, 300)	array([[-0.00814763, 0.1134551 , 0.17975131, ..., 0.06468888, -0.0399 ...
18	float64	(80, 300)	array([[-0.06271242, 0.29980266, -0.01186956, ..., 0.07682225, 0.2802 ...
19	float64	(80, 300)	array([[0.06290901, 0.16301295, 0.04537436, ..., -0.13596888, -0.0057 ...
			array([[-0.06271242, 0.29980266, -0.01186956, ..., 0.07682225,

```
model = Word2Vec.load('WikiEl_300dmc10w5.model');
p  = []
a = N.zeros([1,300])
size = 300
with open("MOBILE_LEM_TRAIN_s_1_0",encoding='ISO-8859-1') as file:

    for line in file:
        array = N.zeros([80,300])
        line = line.strip();
        word= line.split();


        for x in range(0, len(word)-1,1):


                    if word[x] in model.vocab:

                          word[x] = model[word[x]].reshape(1,size)
                          array[x,:] = word[x];



                    else:
                          word[x] = a
                          array[x,:] = word[x];


        p.append(array)

file.close();
```

Συμπέρασμα

Εν κατακλείδι για να έχουμε καλά αποτελέσματα είναι αναγκαίο να δημιουργήσουμε ένα ποιοτικό λεξικό και να εκπαιδεύσουμε τον αλγόριθμο πάνω σε μεγάλα dataset (εκατοντάδων γραμμών).Όσο πιο ποιοτικό είναι το λεξικό τόσο καλύτερα αποτελέσματα θα έχουμε.

Αναφορές

1. LeCun, Yann. "LeNet-5, convolutional neural networks". Retrieved 16 November2013.

2. Zhang, Wei (1988). "Shift-invariant pattern recognition neural network and its optical architecture". Proceedings of annual conference of the Japan Society of Applied Physics.

3. Zhang, Wei (1990). "Parallel distributed processing model with local space-invariant interconnections and its optical architecture". Applied Optics. **29** (32).

4. BMatusugu, Masakazu; Katsuhiko Mori; Yusuke Mitari; Yuji Kaneda (2003). "Subject independent facial expression recognition with robust face detection using a convolutional neural network" (PDF). Neural Networks. **16** (5): 555–559. Doi: 10.1016/S0893-6080(03)00115-1. Retrieved 17 November 2013.

5. Van den Oord, Aaron; Dieleman, Sander; Schrauwen, Benjamin (2013-01-01). Burges, C. J. C.; Bottou, L.; Welling, M.; Ghahramani, Z.; Weinberger, K. Q., eds. Deep content-based music recommendation (PDF). Curran Associates, Inc. pp. 2643–2651.

6. Collobert, Ronan; Weston, Jason (2008-01-01). "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning". Proceedings of the 25th International Conference on Machine Learning. ICML '08. New York, NY, USA: ACM: 160–167. ISBN 978-1-60558-2054. Doi: 10.1145/1390156.1390177.

7. "Convolutional Neural Networks (LeNet) – DeepLearning 0.1 documentation". DeepLearning 0.1. LISA Lab. Retrieved 31 August 2013.

8. Habibi, Aghdam, Hamed. Guide to convolutional neural networks: a practical application to traffic-sign detection and classification. Heravi, Elnaz Jahani, Cham, Switzerland. ISBN 9783319575490. OCLC 987790957.

9. Ciresan, Dan; Ueli Meier; Jonathan Masci; Luca M. Gambardella; Jurgen Schmidhuber (2011). "Flexible, High Performance Convolutional Neural Networks for Image Classification" (PDF). Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two. **2**: 1237–1242. Retrieved 17 November 2013.

10. Krizhevsky, Alex. "ImageNet Classification with Deep Convolutional Neural Networks"(PDF). Retrieved 17 November 2013.

11. Ciresan, Dan; Meier, Ueli; Schmidhuber, Jürgen (June 2012). "Multi-column deep neural networks for image classification". 2012 IEEE Conference on Computer Vision and Pattern Recognition. New York, NY: Institute of Electrical and Electronics Engineers (IEEE): 3642–3649. ISBN 978-1-4673-1226-4. OCLC 812295155. ArXiv: 1202.2745v1 . Doi: 10.1109/CVPR.2012.6248110. Retrieved 2013-12-09.

12. Le Callet, Patrick; Christian Viard-Gaudin; Dominique Barba (2006). "A Convolutional Neural Network Approach for Objective Video Quality Assessment" (PDF). IEEE Transactions on Neural Networks. **17** (5): 1316–1327. PMID 17001990. Doi: 10.1109/TNN.2006.879766. Retrieved 17 November 2013.

13. Hubel, D. H.; Wiesel, T. N. (1968-03-01). "Receptive fields and functional architecture of monkey striate cortex". The Journal of Physiology. **195** (1): 215–243. ISSN 0022-3751. PMC 1557912 . PMID 4966457. Doi: 10.1113/jphysiol.1968.sp008455.

14. LeCun, Yann; Bengio, Yoshua; Hinton, Geoffrey (2015). "Deep learning". Nature. **521**(7553): 436–444. PMID 26017442. Doi: 10.1038/nature14539.

15. Fukushima, Kunihiko (1980). "Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position" (PDF). Biological Cybernetics. **36** (4): 193–202. PMID 7370364. Doi: 10.1007/BF00344251. Retrieved 16 November 2013.

16. David E. Rumelhart; Geoffrey E. Hinton; Ronald J. Wiliams (1986). "Chapter 8: Learning

Internal Representations by ErrorPropagation". In Rumelhart, David E.; McClelland, James.L. Parallel Distributed Processing, Volume 1 (PDF). MIT Press. pp. 319–362. ISBN 9780262680530.

17. *Homma, Toshiteru; Les Atlas; Robert Marks II (1988). "An Artificial Neural Network for Spatio-Temporal Bipolar Patters: Application to Phoneme Classification" (PDF). Advances in Neural Information Processing Systems. **1**: 31–40.*
18. *LeCun, Yann; Léon Bottou; Yoshua Bengio; Patrick Haffner (1998). "Gradient-based learning applied to document recognition" (PDF). Proceedings of the IEEE. **86** (11): 2278–2324. Doi: 10.1109/5.726791. Retrieved October 7, 2016.*
19. S. Behnke. Hierarchical Neural Networks for Image Interpretation, volume 2766 of Lecture Notes in Computer Science. Springer, 2003.
20. Simard, Patrice, David Steinkraus, and John C. Platt. "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis." In ICDAR, vol. 3, pp. 958-962. 2003.
21. *Zhang, Wei (1991). "Error Back Propagation with Minimum-Entropy Weights: A Technique for Better Generalization of 2-D Shift-Invariant NNs". Proceedings of the International Joint Conference on Neural Networks.*
22. *Zhang, Wei (1991). "Image processing of human corneal endothelium based on a learning network". Applied Optics. **30** (29).*
23. *Zhang, Wei (1994). "Computerized detection of clustered microcalcifications in digital mammograms using a shift-invariant artificial neural network". Medical Physics. **21** (4).*
24. Daniel Graupe, Ruey Wen Liu, George S Moschytz."Applications of neural networks to medical signal processing". In Proc. 27th IEEE Decision and Control Conf., pp. 343-347, 1988.
25. https://pythonprogramming.net
26. https://python programming.net/preprocessing-tensorflow-deep-learning-tutorial/
27. https://stackoverflow.com/
28. https://www.tensorflow.org/
29. https://docs.anaconda.com/anaconda/navigator/
30. https://www.dataquest.io/blog/web-scraping-tutorial-python/
31. https://www.crummy.com/software/BeautifulSoup/bs4/doc/
32. Fukunaga, Keinosuke (1990). *Introduction to Statistical Pattern Recognition* (2nd έκδοση). Boston: Academic Press. ISBN 0-12-269851-7.
33. Bishop, Christopher (2006). *Pattern Recognition and Machine Learning*. Berlin: Springer. ISBN 0-387-31073-8.
34. Koutroumbas, Konstantinos. Theodoridis, Sergios (2008). *Pattern Recognition* (4th έκδοση). Boston: Academic Press. ISBN 1-59749-272-8.
35. Hornegger, Joachim. Paulus, Dietrich W. R. (1999). *Applied Pattern Recognition: A Practical Introduction to Image and Speech Processing in C++* (2nd έκδοση). San Francisco: Morgan Kaufmann Publishers. ISBN 3-528-15558-2.
36. Schuermann, Juergen (1996). *Pattern Classification: A Unified View of Statistical and Neural Approaches*. New York: Wiley. ISBN 0-471-13534-8.
37. Godfried T. Toussaint, επιμ. (1988). *Computational Morphology*. Amsterdam: North-Holland Publishing Company.
38. Kulikowski, Casimir A... Weiss, Sholom M. (1991). *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert*

Systems. Machine Learning. San Francisco: Morgan Kaufmann Publishers. ISBN 1-55860-065-5.

39. Ελένη Γολέμη.,(2010).Κρυπτογραφία & Εξόρυξη Δεδομένων.Ανακτήθηκε στις 16 Ιουλίου από http://nemertes.lis.upatras.gr/jspui/bitstream/10889/4791/1/ergasia-golemie.pdf
40. Kantardzic, Mehmed (2003). *Data Mining: Concepts, Models, Methods, and Algorithms*. John Wiley & Sons. ISBN 0-471-22852-4. OCLC 50055336.
41. Fayyad, Usama (1996). «From Data Mining to Knowledge Discovery in Databases». Ανακτήθηκε στις 2012-07-16.
42. Simmi Bagga., Dr. G.N. Singh., (2012).Applications of Data Mining.Ανακτήθηκε στις 19 Απριλίου ,2012 από http://www.ijsett.com/images/P5.pdf
43. path analysis
44. Γούλου Ζωή.,(2010). Εφαρμογή μεθόδων εξόρυξης δεδομένων στη διαχείριση πελατειακών σχέσεων. Ανακτήθηκε στις 18 Ιουλίου από http://dspace.lib.uom.gr/bitstream/2159/14808/6/GoulouZoiMsc2012.pdf
45. *Compiling with C# and Java*, Pat Terry, 2005, ISBN 032126360X624
46. *Algorithms + Data Structures = Programs*, Niklaus Wirth, 1975, ISBN 0-13-022418-9
47. *Compiler Construction*, Niklaus Wirth, 1996, ISBN 0-201-40353-6
48. Sebesta, R. W. (2006). Concepts of programming languages (Seventh edition) pp. 177. Boston: Pearson/Addison-Wesley.
49. Haykin, S. (1999) *Neural Networks: A Comprehensive Foundation*, Prentice Hall, ISBN 0-13-2733501
50. Διαμαντάρας, Κ. (2007) *Τεχνητά Νευρωνικά Δίκτυα*, Κλειδάριθμος, ISBN 9604610805
51. Ματσατσίνης Ν., Συστήματα Υποστήριξης Αποφάσεων, Εκδόσεις Νέων Τεχνολογιών, 2010
52. http://statsoft.com/textbook/neural-networks/
53. Powers, David M W (2011). *"Evaluation: From Precision, Recall and F-Measure to ROC, Unforcedness, Markedness & Correlation"* (PDF). Journal of Machine Learning Technologies. 2 (1): 37–63.
54. Perruchet, P.; Peereman, R. (2004). "The exploitation of distributional information in syllable processing". J. Neurolinguistics. 17 (2–3): 97–119. *Doi*: *10.1016/s0911-6044(03)00059-9*.
55. Powers, David M. W. (2012). "The Problem with Kappa". Conference of the European Chapter of the Association for Computational Linguistics (EACL2012) Joint ROBUS-UNSUP Workshop.
56. Fawcett, Tom (2006). *"An Introduction to ROC Analysis"* (PDF). Pattern Recognition Letters. 27 (8): 861–874. *Doi*: *10.1016/j.patrec.2005.10.010*.
57. Ting, Kai Ming (2011). *Encyclopedia of machine learning*. Springer. *ISBN 978-0-387-30164-8*.