

Πανεπιστήμιο Πατρών  
Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας  
Υπολογιστών

Διπλωματική Εργασία  
**«Κατασκευή κειμένων φυσικής γλώσσας με μεθόδους  
βαθιάς μάθησης και ανατροφοδοτούμενα νευρωνικά δίκτυα»**

Ονοματεπώνυμο: Κρέπιας-Βλάχος Αλέξιος

AM:1019683

Αριθμός Διπλωματικής Εργασίας:

Επιβλέπων: Ευάγγελος Δερματάς

Πάτρα, Σεπτέμβριος 2020

# ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η Διπλωματική Εργασία με θέμα

**Κατασκευή κειμένων φυσικής γλώσσας με μεθόδους βαθιάς  
μάθησης και ανατροφοδοτούμενα νευρωνικά δίκτυα**

Του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας  
Υπολογιστών

Αλέξη Κρέπια-Βλάχου του Ευστρατίου

Αριθμός Μητρώου: 1019683

Παρουσιάστηκε δημόσια και εξετάστηκε στο Τμήμα Ηλεκτρολόγων Μηχανικών  
και Τεχνολογίας Υπολογιστών στις

...../...../.....

Ο Επιβλέπων

Ευάγγελος Δερματάς

Αναπληρωτής Καθηγητής

Ο Διευθυντής του Τομέα

Ιωάννης Μουρτζόπουλος

Καθηγητής

**Αριθμός Διπλωματικής Εργασίας:**

**Θέμα: « Κατασκευή κειμένων φυσικής γλώσσας με μεθόδους βαθιάς μάθησης  
και ανατροφοδοτούμενα νευρωνικά δίκτυα »**

**Φοιτητής:**

**Επιβλέπων:**

## Περιεχόμενα

Περίληψη .....	8
Abstract .....	9
Κεφάλαιο 1: Εισαγωγή .....	10
1.1 Αντικείμενο της εργασίας .....	10
1.2 Στόχος Εργασίας .....	12
1.3 Σημαντικότητα Εργασίας.....	13
1.4 Μεθοδολογία.....	16
Κεφάλαιο 2: Βασικές Έννοιες .....	17
2.1 Τεχνητή Νοημοσύνη .....	17
2.1.1 Ορισμός και βασικά στοιχεία.....	17
2.1.2 Τι είναι η Νοημοσύνη .....	17
2.1.3 Μέθοδοι και στόχοι της Τεχνητής Νοημοσύνης .....	21
2.2 Μηχανική Μάθηση .....	22
2.2.1 Ορισμός.....	22
2.2.2 Διαφορά Μηχανικής Μάθησης με τον Παραδοσιακό Προγραμματισμό .....	23
2.2.3 Λειτουργία μάθησης για έναν υπολογιστή .....	24
2.2.4 Κατηγοριοποίηση της Μηχανικής Μάθησης.....	24
2.2.5 Τα δεδομένα στην μηχανική μάθηση.....	26
Κεφάλαιο 3: Επεξεργασία Φυσικής Γλώσσας.....	29
3.1 Εισαγωγή .....	29
3.2 Ορισμός.....	29
3.3 Χρήση NLP .....	30

3.4 Κατανόηση της Γλώσσας από τους υπολογιστές .....	32
3.5 Τα βήματα της NLP για την κατανόηση των Αγγλικών.....	33
3.5.1 Βήμα 1: Κατάτμηση Πρότασης .....	33
3.5.2 Βήμα 2: Εξαγωγή Συμβόλων .....	34
3.5.3 Βήμα 3: Πρόβλεψη του μέρους του λόγου για κάθε σύμβολο .....	34
3.5.4 Βήμα 4: Καταχώρηση κειμένου.....	35
3.5.5 Βήμα 5: Αναγνώριση Διακοπτόμενων Λέξεων .....	36
3.5.6 Ανάλυση Εξαρτήσεων .....	37
3.5.7 Βήμα 6β: Εύρεση Φράσεων Ουσιαστικών .....	39
3.5.8 Βήμα 7: Αναγνώριση Ονοματικής Οντότητας (NER – Named Entity Recognition) ..	39
3.5.9 Βήμα 8: Ανάλυση Αναφοράς.....	41
Κεφάλαιο 4: Νευρωνικά Δίκτυα.....	43
4.1 Εισαγωγή και Ορισμός .....	43
4.1.1 Γενικά Στοιχεία .....	43
4.1.2 Ορισμός.....	43
4.2 Νευρώνες .....	44
4.3 Συνάρτηση Ενεργοποίησης.....	46
4.3.1 Εισαγωγή και Ορισμός .....	46
4.3.2 Τύποι Συναρτήσεων Ενεργοποίησης .....	47
4.4 Τρόπος Λειτουργίας Νευρωνικών Δικτύων.....	50
4.5 Τρόπος εκμάθησης Νευρωνικών Δικτύων(Μετάδοση προς τα πίσω).....	51
4.5.1 Εισαγωγικά Στοιχεία.....	51
4.5.2 Συνάρτηση Κόστους .....	52
4.5.3 Προσαρμογή των βαρών.....	52
4.6 Μαζική κλίση.....	53

4.7 Στοχαστική Κλίση (SGD – Stochastic Gradient Descent).....	54
4.8 Εκπαίδευση ενός Νευρωνικού Δικτύου με στοχαστική κλίση .....	55
4.9 Ανατροφοδοτούμενα Νευρωνικά Δίκτυα (Recurrent Neural Networks) .....	55
4.9.1 Εισαγωγικά Στοιχεία.....	55
4.9.2 Κοινή Χρήση Παραμέτρων .....	56
4.9.3 Βαθιά Ανατροφοδοτούμενα Νευρωνικά Δίκτυα .....	57
4.9.4 Αμφίδρομα RNN.....	58
4.9.5 Αναδρομικά Νευρωνικά Δίκτυα .....	59
4.9.6 Αποκωδικοποιητής/Αποκωδικοποιητής ακολουθίας σε ακολουθιακά RNN .....	60
Κεφάλαιο 5: TensorFlow .....	61
5.1 Ορισμός και Εισαγωγικά Στοιχεία.....	61
5.2 Λειτουργία TensorFlow .....	61
5.3 Πλεονεκτήματα TensorFlow.....	62
5.4 Παράδειγμα TensorFlow .....	63
5.5 Επιλογές για Φόρτωση δεδομένων στο TensorFlow .....	65
5.6 Δημιουργία Διοχέτευση TensorFlow.....	66
Βήμα 1: Δημιουργία Δεδομένων .....	66
5.7 Σύντομη Ιστορία του TensorFlow .....	68
5.8 Αρχιτεκτονική TensorFlow.....	68
Κεφάλαιο 6: Χρήση TensorFlow για παραγωγή Κειμένων του Ομήρου .....	70
6.1 TensorFlow για παραγωγή Φυσικής Γλώσσας .....	70
6.1.1 Το μοντέλο για Παραγωγή Φυσικής Γλώσσας.....	70
6.1.2 Χρήση του GRU στο TensorFlow. ....	72
6.2 Χρήση των Παραπάνω για Παραγωγή Κειμένων του Ομήρου .....	75
6.2.1 Προεπεξεργασία Κειμένου .....	75

6.2.2 Εκπαίδευση του Μοντέλου.....	75
6.2.3 Πειραματικά Αποτελέσματα.....	76
6.2.4 Σφάλμα εκπαίδευσης συναρτήσει της αρχιτεκτονικής του RNN.....	87
6.2.5 Εκπαίδευση με διαφορετικά μεγέθη δεδομένων.....	87
6.2.6 Παραγόμενο κείμενο.....	88
Βιβλιογραφία .....	89

## Περίληψη

Η βαθιά μάθηση είναι μία μέθοδος η οποία χρησιμοποιείται ευρύτατα σε πολλούς τομείς της σύγχρονης ζωής. Με τη μέθοδο αυτή δίνεται η δυνατότητα στους υπολογιστές να εκτελούν συγκεκριμένες εργασίες που απαιτούν ανθρώπινη ευφυΐα χωρίς ανθρώπινη παρέμβαση.

Αντικείμενο της παρούσας διπλωματικής εργασίας αποτελεί η περιγραφή της διαδικασίας χρήσης ανατροφοδοτούμενων νευρωνικών δικτύων για την παραγωγή φυσικής γλώσσας. Αρχικά αναλύεται το αντικείμενο της εργασίας και αναφέρονται βασικές έννοιες που θα μας απασχολήσουν σε όλη την εργασία (Τεχνητή Νοημοσύνη και Μηχανική Μάθηση). Στη συνέχεια, γίνεται μια εκτενής αναφορά για την επεξεργασία της φυσικής γλώσσας και για τα νευρωνικά δίκτυα. Τέλος επεξηγείται η βιβλιοθήκη της python TensorFlow και ο τρόπος με τον οποίο χρησιμοποιείται για την παραγωγή κειμένων του Ομήρου.



## **Abstract**

Deep learning is a method that is widely used in various sections of modern life. This method provides computers with the capability to do tasks, which require human intelligence, without the human intervention. The subject of this thesis is the description of natural language generation using recurrent neural networks. First of all, the subject and some basic concepts, which are going to be used throughout the whole thesis such as machine learning and artificial intelligence, are being analyzed. Next, we examine in detail neural networks and how natural language processing works. In the end, we explore TensorFlow, a python library, and the way which is used to generate Homer texts.

# Κεφάλαιο 1: Εισαγωγή

## 1.1 Αντικείμενο της εργασίας

Αντικείμενο της παρούσας εργασίας είναι η βιβλιογραφική περιγραφή της χρήσης ανατροφοδοτούμενων νευρωνικών δικτύων για την παραγωγή φυσικής γλώσσας. Συγκεκριμένα, θα περιγράψουμε πως λειτουργούν τα ανατροφοδοτούμενα νευρωνικά δίκτυα για την παραγωγή φυσικής γλώσσας. Επιπλέον, θα τροποποιήσουμε ένα πρόγραμμα, το οποίο είναι διαθέσιμο στο διαδίκτυο, για να δούμε πως με την χρήση βαθιάς μάθησης μπορούμε να έχουμε κείμενα βασισμένα στην Ομήρου Οδύσσεια και Ιλιάδα, τα οποία όμως δεν τα έγραψε κάποιος ποιητής αλλά τα δημιούργησε το πρόγραμμα.

Σε αυτό το σημείο κρίνεται απαραίτητο να παραθέσουμε κάποιους βασικούς όρους, τους οποίους θα συναντήσουμε σε όλη την παρούσα εργασία.

**Τεχνητή Νοημοσύνη:** Η τεχνητή νοημοσύνη (TN) είναι η προσομοίωση των διαδικασιών ανθρώπινης νοημοσύνης από μηχανήματα, ειδικά συστήματα υπολογιστών. Οι ειδικές εφαρμογές της TN περιλαμβάνουν συστήματα ειδικών, επεξεργασία φυσικών γλωσσών, αναγνώριση ομιλίας κ.α. Ο προγραμματισμός TN εστιάζεται σε τρεις γνωστικές δεξιότητες: μάθηση, συλλογισμός και αυτοδιόρθωση:

- **Μαθησιακές Διαδικασίες:** Αυτή η πτυχή του προγραμματισμού TN εστιάζει στην απόκτηση δεδομένων και στη δημιουργία κανόνων για τον τρόπο μετατροπής των δεδομένων σε εφαρμόσιμες πληροφορίες. Οι κανόνες, οι οποίοι ονομάζονται αλγόριθμοι, παρέχουν στις υπολογιστικές συσκευές οδηγίες βήμα προς βήμα για τον τρόπο ολοκλήρωσης μιας συγκεκριμένης εργασίας.
- **Διαδικασίες Συλλογισμού:** Αυτή η πτυχή του προγραμματισμού TN εστιάζεται στην επιλογή του κατάλληλου αλγόριθμου για την επίτευξη του επιθυμητού αποτελέσματος.
- **Διαδικασίες αυτοδιόρθωση:** Αυτή η πτυχή του προγραμματισμού TN έχει σχεδιαστεί για να βελτιώνει συνεχώς τους αλγορίθμους και να διασφαλίζει ότι παρέχουν τα πιο ακριβή αποτελέσματα. (Rouse, 2019)

**Μηχανική Μάθηση:** Η μηχανική μάθηση είναι μια εφαρμογή της τεχνητής νοημοσύνης, η οποία παρέχει τη δυνατότητα στα συστήματα να μαθαίνουν και να βελτιώνονται αυτόματα από την εμπειρία χωρίς να προγραμματίζονται ρητά. Η εκμάθηση της μηχανής εστιάζει στην ανάπτυξη προγραμμάτων υπολογιστών, τα οποία μπορούν να έχουν πρόσβαση σε δεδομένα και να τα χρησιμοποιούν για να μάθουν μόνες τους. Η διαδικασία μάθησης αρχίζει με παρατηρήσεις ή δεδομένα, όπως παραδείγματα, άμεση εμπειρία ή διδασκαλία, προκειμένου να αναζητηθούν πρότυπα στα δεδομένα και να ληφθούν καλύτερες αποφάσεις στο μέλλον με βάση τα παραδείγματα που παρέχουμε. Ο πρωταρχικός στόχος είναι να μπορούν οι υπολογιστές να μαθαίνουν αυτόματα χωρίς ανθρώπινη παρέμβαση ή βοήθεια και να προσαρμόζουν αναλόγως τις δράσεις τους. (What is Machine Learning? A definition, 2017)

**Βαθιά Μάθηση:** Η βαθιά μάθηση είναι μια λειτουργία της τεχνητής νοημοσύνης που μιμείται την λειτουργία του ανθρώπινου εγκεφάλου στην επεξεργασία δεδομένων και δημιουργεί πρότυπα για χρήση στον τομέα της λήψης αποφάσεων. Η βαθιά μάθηση είναι ένα υποσύνολο της μηχανικής μάθησης στην τεχνητή νοημοσύνη, η οποία διαθέτει δίκτυα που μπορούν να μαθαίνουν χωρίς επίβλεψη από δεδομένα, τα οποία είναι αδόμητα ή δεν περιέχουν ετικέτες. Η βαθιά μάθησης είναι επίσης γνωστή ως βαθιά νευρωνική μάθηση ή βαθύ νευρωνικό δίκτυο.

Η βαθιά μάθηση εξελίχθηκε παράλληλα με την ψηφιακή εποχή, η οποία προκάλεσε έκρηξη δεδομένων σε όλες τις μορφές και από κάθε περιοχή του κόσμου. Τα δεδομένα αυτά, γνωστά και ως μεγάλα δεδομένα, προέρχονται από πηγές όπως τα μέσα κοινωνικής δικτύωσης, οι μηχανές αναζήτησης στο διαδίκτυο, οι πλατφόρμες ηλεκτρονικού εμπορίου και οι διαδικτυακοί κινηματογράφοι, μεταξύ των άλλων. Αυτός ο τεράστιος όγκος δεδομένων, είναι άμεσα προσβάσιμος και μπορεί να χρησιμοποιηθεί από κοινού μέσω εφαρμογών fintech, όπως υπολογιστές νέφους. Ωστόσο, τα δεδομένα, τα οποία συνήθως δεν είναι δομημένα, είναι τόσο μεγάλα που θα μπορούσαν να χρειαστούν δεκαετίες για να τα κατανοήσουν οι άνθρωποι και να αντλήσουν σχετικές πληροφορίες. Οι εταιρείες έχουν συνειδητοποιήσει τις απίστευτες δυνατότητες που μπορεί να προκύψουν από την αποκάλυψη του απίστευτου πλούτου πληροφοριών και προσαρμόζονται όλα και περισσότερο σε συστήματα TN για αυτοματοποιημένη υποστήριξη. (Learning, 2019)

**Νευρωνικά Δίκτυα:** Τα νευρωνικά δίκτυα είναι ένα σύνολο αλγορίθμων, ανεπτυγμένων με βάση την λειτουργία του ανθρώπινου εγκεφάλου. Έχουν σχεδιαστεί για να αναγνωρίζουν πρότυπα. Ερμηνεύουν τα αισθητήρια δεδομένα μέσω ενός είδους μηχανικής αντίληψης, επισημάνσης ή συμπλέγματος ακατέργαστων εισόδων. Τα μοτίβα που αναγνωρίζουν είναι αριθμητικά που περιέχονται σε διανύσματα, στα οποία πρέπει να μετατρέπονται όλα τα πραγματικά δεδομένα, είτε πρόκειται για εικόνες, ήχο ή κείμενο. Τα νευρωνικά δίκτυα μας βοηθούν να ομαδοποιούμε και να ταξινομούμε. Μπορείτε να τα θεωρήσετε ως το επίπεδο δημιουργίας ομάδων και ταξινόμησης πάνω στα δεδομένα που αποθηκεύετε και διαχειρίζεστε. Βοηθούν στην ομαδοποίηση των μη επισημασμένων δεδομένων σύμφωνα με τις ομοιότητες μεταξύ των εισόδων παραδειγμάτων και ταξινομούν τα δεδομένα όταν έχουν ένα επισημασμένο σύνολο δεδομένων για εκπαίδευση. Τα νευρωνικά δίκτυα μπορούν επίσης να εξάγουν χαρακτηριστικά, τα οποία τροφοδοτούνται σε άλλους αλγορίθμους για δημιουργία ομάδων και για ταξινόμηση. Έτσι κάποιος μπορεί να παρομοιάσει τα νευρωνικά δίκτυα σαν συστατικά μέρη μεγαλύτερων εφαρμογών μηχανικής μάθησης που περιλαμβάνουν αλγορίθμους για ενίσχυση της μάθησης, της ταξινόμησης και της παλινδρόμησης. (Nicholson, n.d.)

Τις έννοιες της τεχνητής νοημοσύνης, της μηχανικής μάθησης, της βαθιάς μάθησης και των νευρωνικών δικτύων θα αναλύσουμε περαιτέρω στο επόμενο κεφάλαιο. Προς το παρόν θα δούμε άλλη μια έννοια με την οποία θα ασχοληθούμε στην παρούσα εργασία: την έννοια της Επεξεργασίας Φυσικής Γλώσσας:

**Η Επεξεργασία Φυσικής Γλώσσας (NLP – Natural Language Processing)**, είναι ένας κλάδος της τεχνητής νοημοσύνης, ο οποίος ασχολείται με την αλληλεπίδραση μεταξύ υπολογιστών και ανθρώπων χρησιμοποιώντας τη φυσική γλώσσα. Ο τελικός στόχος της επεξεργασίας φυσικής γλώσσας είναι η ανάγνωση, η αποκρυπτογράφηση και κατανόηση των ανθρώπινων γλωσσών. Οι περισσότερες τεχνικές επεξεργασίας φυσικής γλώσσας βασίζονται στην εκμάθηση των μηχανών για την απόκτηση νοήματος της ανθρώπινης γλώσσας.

Για την Επεξεργασία της Φυσικής γλώσσας θα μιλήσουμε εκτενέστερα στο Κεφάλαιο 3.

## **1.2 Στόχος Εργασίας**

Στόχος της συγκεκριμένης εργασίας είναι η πληροφόρηση του αναγνώστη της συγκεκριμένης διπλωματικής, για έναν από τους τρόπους, με τους οποίους μπορεί ένας υπολογιστής να παράγει

φυσική γλώσσα. Συγκεκριμένα, όπως αναφέραμε παραπάνω, θα χρησιμοποιήσουμε το πρόγραμμα για να περιγράψουμε την λειτουργία των νευρωνικών δικτύων και πιο συγκεκριμένα την λειτουργία των ανατροφοδοτούμενων νευρωνικών δικτύων. Επιπλέον, ένας ακόμα στόχος είναι να τροποποιήσουμε κατάλληλα το πρόγραμμα, έτσι ώστε να παράγει κείμενα της Ομήρου Οδύσσειας και Ιλιάδας (Το πρόγραμμα, το οποίο είναι διαθέσιμο σε εμάς, παράγει ποιήματα του Shakespeare).

## 1.3 Σημαντικότητα Εργασίας

Η διαδικασία της κατασκευής φυσικής γλώσσας, είναι μια απλούστατη μορφή Επεξεργασίας Φυσικής Γλώσσας (πολλές φορές η κατασκευή ενός μοντέλου γλώσσας, το οποίο επιτυγχάνει το πρόγραμμα συνεργάζεται με την διαδικασία της Επεξεργασίας Φυσικής Γλώσσας). Η επεξεργασία φυσικής γλώσσας είναι ένας πολύ σημαντικός κλάδος της τεχνητής νοημοσύνης διότι έχει πολλές καινοτόμες εφαρμογές στις μέρες μας. Μερικές από αυτές είναι στον τομέα των επιχειρήσεων:

### 1. Ανάλυση Συναισθήματος

Η ταξινόμηση του συναισθήματος ενός κειμένου, ενός σχολίου ή ενός άρθρου είναι ένα δύσκολο υποκειμενικό έργο ακόμα και για έναν άνθρωπο. Εκεί παρουσιάζεται το NLP. Ένα μοντέλο ανάλυσης συναισθήματος, μπορεί να μας πει εάν ένα κείμενο έχει αρνητική ή θετική πολικότητα. Το επίπεδο ταξινόμησης μπορεί να ποικίλει από ολόκληρο το έγγραφο μέχρι το επίπεδο των λέξεων. Με τη χρήση της ανάλυσης συναισθήματος, μπορούμε να ταξινομήσουμε πράγματα όπως τις κριτικές μιας εταιρείας μας ή των προϊόντων της. Μια άλλη χρήση της ανάλυσης συναισθήματος είναι η ταξινόμηση των απόψεων των ανθρώπων με βάση τα σχόλιά τους και τις δημοσιεύσεις τους στα μέσα κοινωνικής δικτύωσης. Μια πιο σύνθετη πρόκληση είναι η διαίρεση της πολικότητας σε τάξεις με μεγαλύτερη κατανομή, όπως πολύ θετική, θετική, ουδέτερη, αρνητική και πολύ αρνητική. Ένα μοντέλο εκμάθησης φυσικών γλωσσών μπορεί να εκτελέσει αυτά τα καθήκοντα για εμάς. Αυτά τα μοντέλα συνήθως, προσπαθούν να ορίσουν σχέσεις μεταξύ λέξεων και προτάσεων. Για την κατάρτιση ενός τέτοιου μοντέλου, απαιτείται ένας μεγάλος όγκος δεδομένων, όπου το κείμενο εκπαίδευσης έχει ήδη επισημανθεί με ένα κατάλληλο χαρακτηριστικό πολικότητας.

## **2. Επισημάνση Κειμένου**

Η ταξινόμηση κειμένου δεν σταματά στην ανάλυση συναισθήματος. Όπως ήδη αναφέρθηκε, τα μοντέλα επεξεργασίας φυσικών γλωσσών, μπορούν να ταξινομήσουν μεγάλα τμήματα κειμένου. Μπορούν να αποφασίσουν για ένα μήνυμα ηλεκτρονικού ταχυδρομείου, αν πρόκειται για ανεπιθύμητο ή όχι. Ένα καλά εκπαιδευμένο μοντέλο ταξινόμησης κειμένου, μπορεί επίσης να προσθέτει ετικέτες στο περιεχόμενο από τα πρακτορεία ειδήσεων. Αυτές οι τοποθεσίες web, μπορούν να δημοσιεύσουν μόνο βασικές ανακοινώσεις ή μεγάλα άρθρα για κάποιον. Χάρη στην ταξινόμηση του κειμένου, δεν χρειάζεται να ανησυχούν για τη χειροκίνητη σήμανση των άρθρων. Επιπλέον, τα μοντέλα NLP μπορούν να διακρίνουν τα υποκειμενικά από τα αντικειμενικά. Επίσης, είναι δυνατό να ταξινομηθούν ερωτήσεις, είτε ρωτούν για την τοποθεσία, το άτομο ή για αριθμητικές πληροφορίες. Μια από τις αρχιτεκτονικές ταξινόμησης κειμένου ονομάζεται Named Entity Identifier (NER). Μπορεί να βάλει ετικέτες σε οντότητες: π.χ. Η Google είναι εταιρεία.

## **3. Μετάφραση Κειμένου**

Καθώς η παγκοσμιοποίηση ανοίγει νέες αγορές και βάσεις χρηστών από διαφορετικές ηπείρους, η μετάφραση των γλωσσών γίνεται όλο και πιο σημαντική. Η αλήθεια είναι ότι υπάρχουν πολλά διαθέσιμα δεδομένα εκπαίδευσης μοντέλων μηχανικής μάθησης για να μεταφράσουμε λέξεις και ολόκληρα έγγραφα. Μόλις μια επιχείρηση αρχίσει να χρησιμοποιεί αυτά τα μοντέλα, το ανταγωνιστικό της πλεονέκτημα θα εκτοξευθεί στα ύψη και το σύνολο των πελατών της θα πολλαπλασιαστούν. Η επικοινωνία με τους ξένους πελάτες μιας επιχείρησης αλλά και τους πιθανούς επιχειρηματικούς εταίρους θα γίνει ευκολότερη με τη χρήση μοντέλων NLP για μετάφραση. Τα σύνθετα μοντέλα δημιουργούν μια πιο οργανική εμπειρία όσον αφορά την ανάγνωση (ή την ακοή ) μεταφρασμένου κειμένου. Κάθε πελάτης θα αισθάνεται ότι είναι σημαντικός επειδή η εταιρεία ενδιαφέρεται για την επικοινωνία στην γλώσσα του. Δεδομένων όλων αυτών, αξίζει να γίνει επένδυση στα μοντέλα μετάφρασης, ειδικά για τις διεθνείς επιχειρήσεις.

## **4. Απάντηση ερωτήσεων**

Η εξαιρετική εξυπηρέτηση πελατών αποτελεί κορυφαία προτεραιότητα για πολλές επιχειρήσεις. Είτε η επιχείρηση είναι κατάστημα ηλεκτρονικού εμπορίου είτε υπηρεσία

παροχής λύσεων IT, η ζωντανή υποστήριξη επί 24ώρου βάσεως, 7 μέρες την εβδομάδα είναι πολύτιμη. Ωστόσο, η πρόσληψη ανθρώπων για να είναι παρόντες μέρα και νύχτα γίνεται όλο και πιο δύσκολη. Οι χρήστες αναμένουν μια οργανωτική, μοναδική εμπειρία εξυπηρέτησης προς αυτόν. Εάν η επιχείρηση θέλει να αντικαταστήσει τις συνήθεις ερωτήσεις και να διατηρήσει υποστήριξη 24 ώρες την ημέρα, 7 ημέρες την εβδομάδα, η χρήση chatbots αλλά και ψηφιακών βοηθών, είναι μονόδρομος. Τα chatbots πρέπει να εκπαιδεύονται για μια συγκεκριμένη συμπεριφορά, έτσι χρειάζεται κάποια προετοιμασία πριν την ανάπτυξή τους. Διατίθενται προεκπαιδευμένα μοντέλα επεξεργασίας φυσικών γλωσσών για τα chatbots (όπως είναι το ULMFiT και το BERT), ωστόσο, με προσαρμοσμένα μοντέλα μπορεί να επιτευχθεί καλύτερη εμπειρία χρήστη.

## **5. Σύνθεση κειμένου σε ομιλία**

Όπως αναφέρθηκε παραπάνω, οι εικονικοί βοηθοί είναι μια καλή επιλογή εάν κάποιος θέλει περισσότερες υπηρεσίες στην επιχείρησή του. Η χρήση εξατομικευμένων εικονικών βοηθών γίνεται όλο και πιο τακτική. Έτσι, οι τεχνολογικοί γίγαντες δημιουργούν τους δικούς τους. Η Siri της Apple είναι μια από τις πιο γνωστές βοηθούς. Η Microsoft έχει την Cortana, η Amazon την Alexa κλπ. Κάθε τεχνολογικός γίγαντας αναπτύσσει τη δική του εφαρμογή και ερμηνεία ενός εικονικού βοηθού. Ωστόσο, η τεχνολογία που χρησιμοποιείται είναι σχεδόν η ίδια. Τα περισσότερα χαρακτηριστικά του ψηφιακού βοηθού βασίζονται στην ίδια τεχνολογία, η οποία χρησιμοποιεί εφαρμογές επεξεργασίας φυσικών γλωσσών. Για παράδειγμα, η επεξεργασία φυσικής γλώσσας χρησιμοποιείται για την ερμηνεία των οδηγιών των χρηστών, καθώς και για τη σύνθεση της ομιλίας από το γραπτό κείμενο. Όταν ο χρήστης μιλάει στον βοηθό, γνωρίζει τι είδους οδηγίες λέγονται βάση των μαθημένων μοτίβων ομιλίας. Μετά την μετάφραση της ομιλίας σε κείμενο, ο βοηθός θα εκτελέσει τις οδηγίες που έχει υποδείξει ο χρήστης. Παρόμοια λειτουργία εκτελείται, όταν ο ψηφιακός βοηθός λέει στον χρήστη τι να κάνει. Για την παραγωγή ήχου, ο οποίος μοιάζει με ανθρώπινο, το μοντέλο επεξεργασίας φυσικής γλώσσας μαθαίνει πως ακούγεται ο τονισμός για διαφορετικές λέξεις. Προσπαθεί επίσης να διακρίνει τον τρόπο με τον οποίο οι ερωτήσεις διαφέρουν από τις προταγές. Οι ψηφιακοί βοηθοί είναι πολύ σύνθετες υπηρεσίες, αλλά παρέχουν ανεκτίμητα πλεονεκτήματα όσον αφορά την αντιμετώπιση των πελατών.

## **6. Σύνοψη Κειμένου**

Η σύνοψη κειμένου είναι μια δύσκολη διαδικασία ακόμα και για έναν άνθρωπο. Ωστόσο, η επεξεργασία φυσικής γλώσσας μπορεί να μας βοηθήσει να επιταχύνουμε την διαδικασία. Η κύρια πρόκληση είναι να δημιουργηθεί ένα κείμενο που θα είναι κατανοητό για τους ανθρώπους, διατηρώντας παράλληλα τις απαιτούμενες πληροφορίες. Εξακολουθεί να αποτελεί αναδυόμενη τεχνολογία, καθώς μεγάλο μέρος του γραπτού περιεχομένου, τόσο online όσο και offline, αναμένει επεξεργασία. Επίσης, η δημιουργία σύνοψης από όλα αυτά τα κείμενα είναι χρονοβόρα εργασία. (Zsargo, 2019)

## **1.4 Μεθοδολογία**

Για την ανάπτυξη του της εργασίας, θα χρησιμοποιηθούν πηγές από το διαδίκτυο. Θα γίνει αναζήτηση πηγών και βιβλίων σχετικών με το θέμα έτσι ώστε να γίνει μια σωστή και επιστημονικά τεκμηριωμένη εργασία. Επίσης θα γίνουν οι κατάλληλες αλλαγές στον κώδικα του προγράμματος που αναφέραμε παραπάνω, έτσι ώστε να παράγει κείμενα της οδύσσειας.



## Κεφάλαιο 2: Βασικές Έννοιες

Σε αυτό το κεφάλαιο θα αναλύσουμε περεταίρω κάποιες βασικές έννοιες η οποίες είναι θεμελιώδης για το αντικείμενο της παρούσας εργασίας. Αυτές είναι η έννοια της τεχνητής νοημοσύνης, της μηχανικής μάθησης αλλά και της βαθιάς μάθησης.

### 2.1 Τεχνητή Νοημοσύνη

#### 2.1.1 Ορισμός και βασικά στοιχεία

Τεχνητή νοημοσύνη (TN) καλούμε την ικανότητα ενός ηλεκτρονικού υπολογιστή ή ενός ρομπότ που ελέγχεται από υπολογιστή να εκτελεί εργασίες που σχετίζονται με ευφυή όντα. Ο όρος εφαρμόζεται συχνά στο σχέδιο ανάπτυξης συστημάτων, τα οποία διαθέτουν τις πνευματικές διεργασίες που χαρακτηρίζουν τους ανθρώπους, όπως η ικανότητα να βγάζουν λογικά συμπεράσματα, να βγάζουν νόημα, να γενικεύουν ή να μαθαίνουν από προηγούμενη εμπειρία. Από την ανάπτυξη του πρώτου ηλεκτρονικού υπολογιστή το 1940, έχει αποδειχθεί ότι οι υπολογιστές μπορούν να προγραμματιστούν για να εκτελούν πολύ περίπλοκα καθήκοντα (όπως για παράδειγμα η ανακάλυψη αποδείξεων για μαθηματικά θεωρήματα) με μεγάλη επάρκεια. Ωστόσο, παρά τις συνεχείς προόδους στην ταχύτητα επεξεργασίας των υπολογιστών και στη χωρητικότητα της μνήμης, δεν υπάρχουν ακόμα προγράμματα που να μπορούν να συνδυάσουν την ανθρώπινη ευελιξία με ευρύτερους τομείς ή με εργασίες που απαιτούν πολλές καθημερινές γνώσεις. Από την άλλη, ορισμένα προγράμματα έχουν επιτύχει τα επίπεδα απόδοσης των ανθρώπινων εμπειρογνομώνων κατά την εκτέλεση συγκεκριμένων καθηκόντων, έτσι ώστε η τεχνητή νοημοσύνη με αυτή την περιορισμένη έννοια να εντοπίζεται σε εφαρμογές τόσο διαφορετικές όπως η ιατρική διάγνωση, οι μηχανές αναζήτησης υπολογιστών αλλά και η αναγνώριση φωνής ή χειρόγραφου.

#### 2.1.2 Τι είναι η Νοημοσύνη

Σχεδόν όλη η ανθρώπινη συμπεριφορά αποδίδεται στην ευφυΐα, ενώ ακόμα και η πιο περίπλοκη συμπεριφορά των εντόμων δεν θεωρείται ως ένδειξη νοημοσύνης. Ας δούμε ποια είναι η διαφορά. Για το λόγο αυτό κρίνεται σκόπιμο να πάρουμε ως παράδειγμα την συμπεριφορά της σφήκας *Sphex ichneumoneus*. Όταν η θηλυκή σφήκα επιστρέψει στην φωλιά της με φαγητό, το αποθέτει πρώτα στο κατώφλι της, ελέγχει για εισβολείς στην φωλιά της, και μόνο τότε, εάν δεν βρει κάποιον εισβολέα, μεταφέρει το φαγητό της μέσα. Η πραγματική φύση της ενστικτώδους

συμπεριφοράς της σφήκας αποκαλύπτεται αν το φαγητό μετακινηθεί λίγα εκατοστά μακριά από την είσοδο της φωλιάς της ενώ είναι μέσα: καθώς το φαγητό μετακινείται θα επαναλάβει όλη την παραπάνω διαδικασία όσο το φαγητό αλλάζει θέση. Η ευφυΐα, πρέπει να περιλαμβάνει την ικανότητα προσαρμογής σε νέες συνθήκες.

Οι ψυχολόγοι γενικά δεν χαρακτηρίζουν την ανθρώπινη νοημοσύνη μόνο με ένα χαρακτηριστικό, αλλά με το συνδυασμό πολλών διαφορετικών ικανοτήτων. Η έρευνα στον τομέα της ΤΝ εστιάστηκε κυρίως στα ακόλουθα στοιχεία πληροφοριών: εκμάθηση, αιτιολογία, επίλυση προβλημάτων, αντίληψη και χρήση γλώσσας.

### **Εκμάθηση**

Υπάρχουν διάφορες μορφές μάθησης που εφαρμόζονται στην τεχνητή νοημοσύνη. Το πιο απλό είναι η μάθηση με δοκιμή και σφάλμα. Για παράδειγμα, ένα απλό πρόγραμμα υπολογιστή για την επίλυση προβλημάτων παιχνιδιών σκάκι, μπορεί να δοκιμάζει τυχαίες κινήσεις μέχρι να βρεθεί αντίπαλος. Στη συνέχεια, το πρόγραμμα μπορεί να αποθηκεύει την λύση με τη θέση, ώστε την επόμενη φορά που ο υπολογιστής συναντήσει την ίδια θέση να χρησιμοποιήσει και την αντίστοιχη λύση. Αυτή η απλή απομνημόνευση μεμονωμένων αντικειμένων και διαδικασιών, γνωστή ως εκμάθηση της σειράς, είναι σχετικά εύκολη στην εφαρμογή σε έναν υπολογιστή. Μεγαλύτερη πρόκληση είναι το πρόβλημα της εφαρμογής αυτού που ονομάζεται γενίκευση. Η γενίκευση περιλαμβάνει την εφαρμογή της εμπειρίας του παρελθόντος σε ανάλογες νέες καταστάσεις. Για παράδειγμα, ένα πρόγραμμα που μαθαίνει τον προηγούμενο χρόνο των απλών αγγλικών ρημάτων δεν θα μπορεί να παράγει τον προηγούμενο χρόνο μιας λέξης όπως το jump, εκτός αν στο παρελθόν έχει τροφοδοτηθεί με το ρήμα jump.

### **Αιτιολογία**

Η αιτιολογία είναι να συνάγονται συμπεράσματα ανάλογα με την κατάσταση. Τα συμπεράσματα ταξινομούνται είτε ως αφαιρετικά είτε ως επαγωγικά. Ένα παράδειγμα του πρώτου είναι: «Ο Φρεντ πρέπει να βρίσκεται είτε στο μουσείο είτε στην καφετέρια. Δεν είναι στην καφετέρια. Επομένως, βρίσκεται στο μουσείο.». Παράδειγμα της επαγωγικής ταξινόμησης συμπερασμάτων είναι: «Προηγούμενα ατυχήματα αυτού του είδους προκλήθηκαν από βλάβη οργάνων, επομένως το ατύχημα προκλήθηκε από βλάβη του οργάνου.». Η σημαντικότερη διαφορά μεταξύ αυτών των μορφών της συλλογιστικής είναι ότι, στην υπόθεση της αφαίρεσης, η αλήθεια της υπόθεσης

εγγυάται την αλήθεια του αποτελέσματος, ενώ στην επαγωγική περίπτωση η αλήθεια της υπόθεσης παρέχει υποστήριξη στο συμπέρασμα χωρίς να παρέχει απόλυτη βεβαιότητα. Η επαγωγική λογική είναι συνήθης στην επιστήμη, όπου τα δεδομένα συλλέγονται και αναπτύσσονται δοκιμαστικά μοντέλα για να περιγράψουν και να προβλέψουν την μελλοντική συμπεριφορά, έως ότου η εμφάνιση ανωμαλιών δεδομένων αναγκάσει το μοντέλο να αναθεωρήσει. Η συλλογιστική της αφαίρεσης, είναι συνήθης στα μαθηματικά και τη λογική, όπου οι πολύπλοκες δομές των θεωρημάτων δημιουργούνται από ένα μικρό σύνολο βασικών αξόνων και κανόνων.

Ο προγραμματισμός ηλεκτρονικών υπολογιστών έχει σημειώσει σημαντική επιτυχία, ιδίως με την εξαγωγή συμπερασμάτων. Ωστόσο, η πραγματική συλλογιστική δεν αφορά μόνο την εξαγωγή συμπερασμάτων, αλλά περιλαμβάνει και την εξαγωγή συμπερασμάτων σχετικά με την επίλυση ενός συγκεκριμένου καθήκοντος ή κατάστασης. Αυτό είναι ένα από τα δυσκολότερα προβλήματα που αντιμετωπίζει η ΤΝ.

### **Επίλυση Προβλημάτων**

Η επίλυση προβλημάτων, ιδίως στην τεχνητή νοημοσύνη, μπορεί να χαρακτηριστεί ως συστηματική αναζήτηση μέσω μιας σειράς πιθανών ενεργειών, προκειμένου να επιτευχθεί ένας προκαθορισμένος στόχος ή λύση. Οι μέθοδοι επίλυσης προβλημάτων υποδιαιρούνται σε ειδικό σκοπό και γενικό σκοπό. Μια μέθοδος ειδικού σκοπού είναι προσαρμοσμένη σε ένα συγκεκριμένο πρόβλημα και συχνά αξιοποιεί πολύ συγκεκριμένα χαρακτηριστικά της κατάστασης στην οποία εντάσσεται το πρόβλημα. Αντίθετα, μια μέθοδος γενικού σκοπού εφαρμόζεται σε ένα ευρύ φάσμα προβλημάτων. Μια τεχνική γενικής χρήσης που χρησιμοποιείται στην τεχνητή νοημοσύνη είναι η αναλυτική ανάλυση. Το πρόγραμμα επιλέγει ενέργειες από μια λίστα μέσων: στην περίπτωση ενός απλού ρομπότ αυτές μπορεί να αποτελούνται από PICKUP, PUTDOWN, MOVEFORWARD, MOVEBACK, MOVELEFT και MOVERIGHT, μέχρι να επιτευχθεί ο στόχος. Πολλά διαφορετικά προβλήματα έχουν επιλυθεί μέσω προγραμμάτων τεχνητής νοημοσύνης. Μερικά παραδείγματα είναι η εύρεση της νικηφόρου κίνησης ή ακολουθίας κινήσεων σε ένα επιτραπέζιο παιχνίδι, η επινόηση μαθηματικών αποδείξεων και ο χειρισμός εικονικών αντικειμένων σε έναν κόσμο που παράγεται από υπολογιστή.

## Αντίληψη

Στην αντίληψη, το περιβάλλον σαρώνεται μέσω διαφόρων αισθητηριακών οργάνων, πραγματικών ή τεχνητών, και το σκηνικό αποσυντίθεται σε ξεχωριστά αντικείμενα. Η ανάλυση περιπλέκεται από το γεγονός ότι ένα αντικείμενο μπορεί να φαίνεται διαφορετικό ανάλογα με την γωνία από την οποία κάποιος παρατηρητής το βλέπει, την κατεύθυνση και την ένταση του φωτισμού στη σκηνή και το βαθμό αντίθεσης του αντικειμένου με το γύρω πεδίο. Επι του παρόντος, η τεχνητή αντίληψη είναι αρκετά προηγμένη ώστε να επιτρέπει στους οπτικού αισθητήρες να αναγνωρίζουν άτομα, τα αυτόνομα οχήματα να οδηγούν με μέτριες ταχύτητες στον ανοιχτό δρόμο, και τα ρομπότ να περιφέρονται μέσω κτιρίων συλλέγοντας άδεια κουτάκια σόδας. Ένα από τα παλαιότερα συστήματα για την ενσωμάτωση της αντίληψης και της δράσης ήταν το FREDDY, ένα στατικό ρομπότ με κινούμενο τηλεοπτικό μάτι και ένα χέρι, κατασκευασμένο στο Πανεπιστήμιο του Εδιμβούργου της Σκωτίας, κατά την περίοδο 1966-1973 υπό την καθοδήγηση του Donald Mickie. Η FREDDY μπόρεσε να αναγνωρίσει μια ποικιλία αντικειμένων και μπορούσε να λάβει οδηγίες για τη συναρμολόγηση απλών αντικειμένων, όπως ένα αυτοκίνητο-παιχνίδι, από ένα σωρό τυχαίων εξαρτημάτων.

## Γλώσσα

Η γλώσσα είναι ένα σύστημα σημαδιών που έχουν νόημα. Υπό την έννοια αυτή, η γλώσσα δεν πρέπει να περιορίζεται στον προφορικό λόγο. Ένα σημαντικό χαρακτηριστικό των ολοκληρωμένων ανθρώπινων γλωσσών, σε αντίθεση με τα τιτιβίσματα πουλιών και τις πινακίδες κυκλοφορίας, είναι η παραγωγικότητά τους. Μια παραγωγική γλώσσα μπορεί να διαμορφώσει μια απεριόριστη ποικιλία προτάσεων. Είναι σχετικά εύκολο να γράψει κανείς προγράμματα υπολογιστών που φαίνονται ικανά, σε αυστηρά περιορισμένα πλαίσια, να ανταποκριθούν με άνεση σε ερωτήσεις και δηλώσεις σε μια ανθρώπινη γλώσσα. Αν και κανένα από τα προγράμματα αυτά δεν κατανοεί την γλώσσα, μπορεί, καταρχήν, να φτάσει στο σημείο όπου η γνώση μιας γλώσσας να συναγωνίζεται την γνώση ενός φυσιολογικού ανθρώπου. Σε αυτό το σημείο πρέπει να αναρωτηθούμε τι σημαίνει η πραγματική κατανόηση, εάν ακόμα και ένας ηλεκτρονικός υπολογιστής δεν κατανοεί πραγματικά την γλώσσα. Δεν υπάρχει καθολική συμφωνία για την απάντηση σε αυτό το ερώτημα. Σύμφωνα με μια θεωρία, το αν κάποιος κατανοεί κάτι ή όχι δεν εξαρτάται μόνο από τη συμπεριφορά του αλλά και από την ιστορία του:

για να μπορεί να πει κάποιος ότι καταλαβαίνει, πρέπει να έχει μάθει την γλώσσα και να έχει εκπαιδευτεί για να πάρει τη θέση του στη γλωσσική κοινότητα μέσω της αλληλεπίδρασης με άλλους χρήστες της γλώσσας.

### **2.1.3 Μέθοδοι και στόχοι της Τεχνητής Νοημοσύνης**

#### **Συμβολικές Προσεγγίσεις έναντι συνδεδεμένων προσεγγίσεων**

Η έρευνα στην TN ακολουθεί δύο διαφορετικές και, σε κάποιο βαθμό, ανταγωνιστικές μεθόδους, τη συμβολική (ή «από πάνω προς τα κάτω») προσέγγιση και την συνδετική προσέγγιση (ή «από κάτω προς τα πάνω»). Η προσέγγιση από πάνω προς τα κάτω επιδιώκει να αναπαράγει την ευφυία αναλύοντας τη γνώση ανεξάρτητα από τη βιολογική δομή του εγκεφάλου, όσον αφορά την επεξεργασία των συμβόλων-όταν πρόκειται για τη συμβολική ετικέτα. Η προσέγγιση από κάτω προς τα πάνω, από την άλλη πλευρά, περιλαμβάνει τη δημιουργία τεχνητών νευρωνικών δικτύων με στόχο την απομίμηση της δομής του εγκεφάλου. Για να απεικονίσουμε την διαφορά αυτών των προσεγγίσεων, θα εξετάσουμε την εργασία κατασκευής ενός συστήματος, εξοπλισμένου με οπτικό σαρωτή, ο οποίος αναγνωρίζει τα γράμματα του αλφάβητου. Μια προσέγγιση από κάτω προς τα πάνω συνήθως περιλαμβάνει την εκπαίδευση ενός τεχνητού νευρωνικού δικτύου παρουσιάζοντας γράμματα σε αυτό ένα προς ένα βελτιώνοντας σταδιακά την απόδοση του δικτύου (Η ρύθμιση αλλάζει την απόκριση των διαφορετικών νευρικών οδών σε διαφορετικά ερεθίσματα). Αντίθετα, μια προσέγγιση από πάνω προς τα κάτω συνήθως περιλαμβάνει τη σύνταξη ενός προγράμματος υπολογιστή, το οποίο συγκρίνει κάθε γράμμα με γεωμετρικές περιγραφές. Με απλά λόγια, οι νευρικές δραστηριότητες αποτελούν τη βάση της προσέγγισης από κάτω προς τα πάνω, ενώ οι συμβολικές περιγραφές αποτελούν τη βάση της προσέγγισης από πάνω προς τα κάτω.

#### **Ισχυρή TN, εφαρμοσμένη TN και γνωστική προσομοίωση**

Εφαρμόζοντας τις μεθόδους που περιγράφονται πιο πάνω, η έρευνα στον τομέα της TN επιχειρεί να επιτύχει έναν από τους τρεις στόχους: Ισχυρή TN, εφαρμοσμένη TN ή γνωστική προσομοίωση. Η ισχυρή TN στοχεύει στην κατασκευή μηχανών που σκέφτονται (Ο όρος ισχυρή TN καθιερώθηκε για αυτή την κατηγορία έρευνας το 1980 από τον φιλόσοφο John Searle του πανεπιστημίου της Καλιφόρνια στο Berkeley). Η τελική φιλοδοξία της ισχυρής τεχνητής νοημοσύνης είναι να παραχθεί ένα μηχάνημα του οποίου η συνολική πνευματική ικανότητα να

μην ξεχωρίζει από εκείνη του ανθρώπου. Μέχρι σήμερα η πρόοδος σε αυτό το εγχείρημα ήταν πενιχρή. Κάποιοι επικριτές αμφιβάλλουν εάν η έρευνα θα δημιουργήσει ακόμα και ένα σύστημα με τη συνολική πνευματική ικανότητα ενός μυρμηγκιού. Πράγματι, ορισμένοι ερευνητές που εργάζονται στα άλλα δύο τμήματα της TN θεωρούν ότι η ισχυρή TN δεν αξίζει να επιδιωχθεί.

Η εφαρμοσμένη TN, γνωστή και ως προηγμένη επεξεργασία πληροφοριών, έχει ως στόχο την παραγωγή εμπορικά βιώσιμων «έξυπνων» συστημάτων – για παράδειγμα ειδικά συστήματα ιατρικής διάγνωσης και συστήματα διαπραγμάτευσης αποθεμάτων. Η εφαρμοσμένη τεχνητή νοημοσύνη έχει σημειώσει σημαντική επιτυχία.

Στην γνωστική προσομοίωση, οι υπολογιστές χρησιμοποιούνται για να δοκιμάσουν θεωρίες για το πώς λειτουργεί το ανθρώπινο μυαλό (παράδειγμα: θεωρίες για το πώς οι άνθρωποι αναγνωρίζουν πρόσωπα ή θυμούνται αναμνήσεις.) Η γνωστική προσομοίωση είναι ήδη ένα ισχυρό εργαλείο τόσο στην νευροεπιστήμη όσο και στη γνωστική ψυχολογία. (Copeland, n.d.)

## 2.2 Μηχανική Μάθηση

### 2.2.1 Ορισμός

Σύμφωνα με τον Arthur Samuel, έναν πρωτοπόρο στον κλάδο της τεχνητής νοημοσύνης και των παιχνιδιών υπολογιστών, **Μηχανική Μάθηση** είναι ο κλάδος που δίνει στους υπολογιστές την δυνατότητα να «μάθουν» χωρίς να έχουν προγραμματιστεί ρητά.

Με απλά λόγια, με την Μηχανική Μάθηση μπορεί να γίνει αυτοματοποίηση και βελτίωση της ικανότητας των υπολογιστών να μαθαίνουν χωρίς να έχουν προγραμματιστεί στην πραγματικότητα. Αυτό σημαίνει ότι αφότου χρησιμοποιήσουν Μηχανικοί Μάθηση οι ηλεκτρονικοί υπολογιστές δεν θα κάνουν ενέργειες επειδή τους έχει προγραμματίσει κάποιος προγραμματιστής, αλλά επειδή έχουν μάθει. Η διαδικασία ξεκινάει με το να δίνεις σαν είσοδο καλά και ποιοτικά δεδομένα. Έτσι χρησιμοποιώντας αυτά τα δεδομένα σε διαφορετικούς αλγόριθμους δημιουργούνται τα μοντέλα μηχανικής μάθησης. Η επιλογή των αλγορίθμων εξαρτάται αποκλειστικά από το είδος των δεδομένων που έχουμε στην διάθεσή μας αλλά και από το είδος της εργασίας που θέλουμε να αυτοματοποιήσουμε.

**Παράδειγμα: Εκπαίδευση των μαθητών σε ένα διαγώνισμα**

Κατά την διάρκεια ενός διαγωνίσματος, οι μαθητές προσπαθούν να μάθουν τα πάντα για το μάθημα. Πριν το διαγώνισμα δίνουν στο μηχάνημα(εγκέφαλο) αρκετά και ποιοτικά δεδομένα (ερωτήσεις και απαντήσεις από διαφορετικά βιβλία ή σημειώσεις από καθηγητές). Στην πραγματικότητα εκπαιδεύουν τον εγκέφαλό τους και με είσοδο και με έξοδο. Για παράδειγμα, το είδος της λογικής που θα χρησιμοποιήσουν για να λύσουν ένα πρόβλημα είναι ένας τρόπος εκπαίδευσης εξόδου. Κάθε φορά λύνουν προβλήματα και βρίσκουν το πόσο καλά τα πήγανε (ευστοχία / βαθμολογία) με το να συγκρίνουν τις απαντήσεις που έδωσαν με τις απαντήσεις των προβλημάτων. Με αυτό τον τρόπο, σταδιακά η απόδοση των ανεβαίνει και είναι πιο σίγουρο ότι θα τα πάνε καλά στο διαγώνισμα.

Αυτός είναι ο τρόπος που κατασκευάζονται τα πραγματικά μοντέλα, εκπαιδεύεται η μηχανή με δεδομένα (είσοδο και έξοδο) και όταν γίνει αυτή η διαδικασία με αποτελεσματικό τρόπο, ακολουθεί δοκιμή στα δεδομένα (μόνο είσοδος) και έτσι παράγεται η βαθμολογία του μοντέλου συγκρίνοντας την απάντηση που έδωσε το μηχάνημα με την πραγματική έξοδο η οποία δεν έχει δοθεί κατά την διάρκεια της εκπαίδευσης.

## 2.2.2 Διαφορά Μηχανικής Μάθησης με τον Παραδοσιακό Προγραμματισμό



Εικόνα 4: Μηχανική Μάθηση και Παραδοσιακός Προγραμματισμός

Όπως φαίνεται και στην παραπάνω εικόνα η διαφορά της μηχανικής μάθησης με τον παραδοσιακό προγραμματισμό είναι πολύ απλή. Στον παραδοσιακό προγραμματισμό με την

παροχή δεδομένων αλλά και προγράμματος τα τρέχουμε στον υπολογιστή και μας παράγει κάποια έξοδο. Από την άλλη, στην μηχανική μάθηση ο υπολογιστής τροφοδοτείται με δεδομένα(είσοδο) αλλά και έξοδο, γίνεται η εκτέλεση στο μηχάνημα κατά την διάρκεια της εκπαίδευσης, και με βάση την εκπαίδευση παράγεται ένα καινούριο πρόγραμμα με την δικιά του λογική(πρόγραμμα) το οποίο μπορεί να αξιολογηθεί κατά την διάρκεια του ελέγχου.

### **2.2.3 Λειτουργία μάθησης για έναν υπολογιστή**

Παρακάτω θα δούμε πως λειτουργεί η μηχανική μάθηση:

1. Αρχικά, συλλέγονται παλιά δεδομένα σε μορφή κατάλληλη για επεξεργασία. Όσο πιο ποιοτικά είναι τα δεδομένα τόσο πιο κατάλληλα είναι για την μοντελοποίηση.
2. Επεξεργασία Δεδομένων: Μερικές φορές τα δεδομένα είναι σε μορφή που δεν έχουν δεχτεί επεξεργασία και υπάρχει πιθανότητα να λείπουν κάποιες τιμές. Οι τιμές που λείπουν σε αριθμητικά χαρακτηριστικά, όπως είναι η τιμή του σπιτιού μπορεί να αντικατασταθεί με την μέση τιμή του χαρακτηριστικού.
3. Στην συνέχεια γίνεται διαίρεση της πληροφορίας σε εκπαιδευτικά σύνολα, σύνολα διασταυρούμενη επικύρωσης και σύνολα δοκιμών. Η αναλογία μεταξύ των επιμέρους συνόλων πρέπει να είναι 6:2:2.
4. Έπειτα, γίνεται η δημιουργία μοντέλων χρησιμοποιώντας κατάλληλους αλγορίθμους και τεχνικές στα σύνολα δοκιμών.
5. Τέλος, το μοντέλο δοκιμάζεται με δεδομένα τα οποία δεν είχαν τροφοδοτηθεί στο μοντέλο κατά την διάρκεια της εκπαίδευσης και γίνεται η αξιολόγηση της απόδοσης.  
(Gupta, ML | What is Machine Learning ?, n.d.)

### **2.2.4 Κατηγοριοποίηση της Μηχανικής Μάθησης**

Οι υλοποιήσεις της μηχανικής μάθησης χωρίζονται σε 3 μεγάλες κατηγορίες:

#### **Επιτηρούμενη Μάθηση:**

Όταν ένας αλγόριθμος μαθαίνει από παραδείγματα δεδομένων και σχετικές απαντήσεις, οι οποίες μπορούν να αποτελούνται από αριθμητικές τιμές ή ετικέτες όπως είναι οι κλάσεις, για να μπορεί αργότερα να προβλεφθεί η σωστή απάντηση όταν τροφοδοτείται με νέα παραδείγματα, αυτό ανήκει στην κατηγορία της επιτηρούμενης μάθησης.



### **Μη Επιτηρούμενη Μάθηση:**

Όταν ένας αλγόριθμος μαθαίνει από απλά παραδείγματα χωρίς σχετιζόμενες απαντήσεις, αφήνουμε τον αλγόριθμο να αποφασίσει μόνος του τα πρότυπα δεδομένων. Αυτός ο τύπος αλγορίθμου τείνει να αναδομεί τα δεδομένα σε κάτι άλλο, όπως για παράδειγμα νέα χαρακτηριστικά που μπορεί να αντιπροσωπεύει μια κλάση ή μια νέα σειρά μη συσχετιζόμενων τιμών. Είναι αρκετά χρήσιμο στο να δίνει στον άνθρωπο πληροφορίες σχετικά με την σημασία των δεδομένων σε αλγόριθμους επιτηρούμενης μάθησης.

### **Ενίσχυση της μάθησης:**

Μερικές φορές τα δεδομένα που τροφοδοτούνται στον αλγόριθμο μηχανικής μάθησης δεν συνοδεύονται από ετικέτες. Ωστόσο υπάρχει δυνατότητα μαζί ένα παράδειγμα να ακολουθείται με θετική ή αρνητική ανατροφοδότηση με βάση την λύση που προτείνει ο αλγόριθμος. Αυτή η προσέγγιση καλείται Ενίσχυση της Μάθησης και συνδέεται με εφαρμογές για τις οποίες ο αλγόριθμος πρέπει να πάρει αποφάσεις και οι αποφάσεις αυτές έχουν κάποιες συνέπειες. Στον ανθρώπινο κόσμο αυτό είναι σαν να μαθαίνεις με δοκιμή και λάθος. Τα λάθη βοηθούν στο να μαθαίνεις επειδή έχουν μια ποινή (κόστος, απώλεια χρόνου κ.λπ.). Από αυτά τα λάθη μαθαίνουμε ότι μια πορεία δράσης είναι πιο πιθανό να πετύχει σε σχέση με άλλες. Ένα τέτοιο παράδειγμα ενισχυτικής μάθησης είναι όταν οι υπολογιστές μαθαίνουν να παίζουν παιχνίδια από μόνοι τους.

Σε αυτή την περίπτωση, η εφαρμογή παρουσιάζει στον αλγόριθμο παραδείγματα συγκεκριμένων καταστάσεων. Για παράδειγμα μια τέτοια κατάσταση μπορεί να είναι όταν ένας παίκτης βρίσκεται παγιδευμένος σε έναν λαβύρινθο ενώ προσπαθεί να αποφύγει κάποιον εχθρό. Η εφαρμογή ενημερώνει τον αλγόριθμο για την έκβαση των ενεργειών στις οποίες προβαίνει και η μάθηση γίνεται όσο προσπαθεί να αποφύγει τους κινδύνους.

### **Ημι-επιτηρούμενη Μάθηση**

Σε αυτό το είδος μηχανικής μάθησης ο αλγόριθμος τροφοδοτείται με μη ολοκληρωμένα δεδομένα. Αυτή είναι μια ιδιαίτερη περίπτωση της αρχής της Μεταγωγής όπου ολόκληρο το σύνολο των στιγμιότυπων των προβλημάτων είναι γνωστό κατά την διάρκεια της εκμάθησης, με εξαίρεση ότι λείπει το κομμάτι των στόχων.

Ένα άλλο είδος κατηγοριοποίησης εργασιών μηχανικής μάθησης υπάρχει και έχει σχέση με την επιθυμητή έξοδο ενός συστήματος μηχανικής μάθησης:

### **Ταξινόμηση:**

Όταν οι εισόδοι χωρίζονται σε δύο ή περισσότερες κατηγορίες και ο εκπαιδευόμενος πρέπει να δώσει ένα μοντέλο, το οποίο δίνει εισόδους που δεν έχουν προβλεφθεί σε μια ή περισσότερες κλάσεις. Αυτό συνήθως γίνεται με βάση την επιτηρούμενη μάθηση. Το φιλτράρισμα ανεπιθύμητων είναι ένα τέτοιο παράδειγμα όπου η είσοδος είναι ένα email και οι κλάσεις είναι «Ανεπιθύμητο» ή «Μη ανεπιθύμητο».

### **Παλινδρόμηση**

Είναι και αυτή μια περίπτωση επιτηρούμενης μάθησης. Σε αυτή την περίπτωση η έξοδος είναι συνεχής και όχι διακριτή.

### **Συσταδοποίηση:**

Είναι μια τεχνική όπου ένα σύνολο εισόδων διαιρείται σε ομάδες. Σε αντίθεση με την ταξινόμηση, οι ομάδες δεν είναι γνωστές εκ των προτέρων και αυτό έχει σαν αποτέλεσμα αυτές οι εργασίες είναι μη επιτηρούμενες. (Gupta, ML | Introduction to Data in Machine Learning, n.d.)

## **2.2.5 Τα δεδομένα στην μηχανική μάθηση**

**Δεδομένα:** Μπορεί να είναι οτιδήποτε δεν έχει υποστεί επεξεργασία όπως είναι τιμές, κείμενο, ήχο ή εικόνα που δεν έχει ερμηνευτεί και αναλυθεί. Τα δεδομένα είναι το πιο σημαντικό στην ανάλυση δεδομένων, την μηχανική μάθηση και την τεχνητή νοημοσύνη. Χωρίς τα δεδομένα, δεν μπορεί να γίνει εκπαίδευση των μοντέλων και όλη η έρευνα που έχει γίνει πάει στράφι. Μεγάλες εταιρίες ξοδεύουν πολλά χρήματα μόνο για να συγκεντρώνουν μεγάλες ποσότητες συγκεκριμένων δεδομένων. Παράδειγμα: Το Facebook απέκτησε το WhatsApp πληρώνοντας 19 εκατομμύρια. Αυτό έγινε διότι η εταιρεία ήθελε να έχει πρόσβαση σε δεδομένα χρηστών που μπορεί η Facebook να μην είχε πρόσβαση αλλά η WhatsApp να είχε. Αυτά τα δεδομένα είναι σημαντικά για την Facebook διότι με την συγκέντρωσή τους η εταιρεία έχει την δυνατότητα να βελτιώσει τις υπηρεσίες της.

**Πληροφορίες:** Οι πληροφορίες που έχουν ερμηνευτεί και διαχειριστεί και έχουν κάποια σημασία για τους χρήστες.

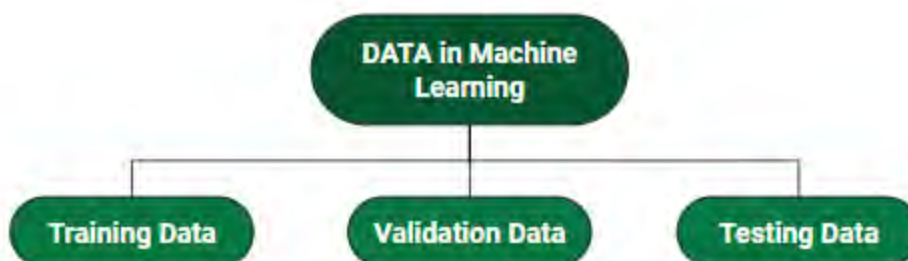
**Γνώση:** Συνδυασμός πληροφοριών, εμπειριών και μάθησης. Έχει σαν αποτέλεσμα την ευαισθητοποίηση ή την δημιουργία ιδεών για ένα άτομο ή οργανισμό.



Εικόνα 5: Δεδομένα Πληροφορία, Γνώση

### Δεδομένα και Μηχανική Μάθηση

- Δεδομένα Εκπαίδευσης: Τα δεδομένα που χρησιμοποιούμε για να εκπαιδεύσουμε το μοντέλο μας. Αυτά είναι τα δεδομένα όπου βλέπει το μοντέλο μας.
- Δεδομένα επικύρωσης: Τα δεδομένα που χρησιμοποιούνται για να γίνεται μια συχνή αξιολόγηση του μοντέλου.
- Δεδομένα δοκιμής: Όταν το μοντέλο που έχει δημιουργηθεί εκπαιδευτεί πλήρως, τα δεδομένα δοκιμής παρέχουν αμερόληπτη αξιολόγηση. Όταν τροφοδοτούνται οι είσοδοι με δεδομένα δοκιμής, το διαθέσιμο μοντέλο προβλέπει κάποιες τιμές (χωρίς να βλέπει την πραγματική έξοδο). Μετά την πρόβλεψη, γίνεται αξιολόγηση του μοντέλου με το να συγκρίνονται με την πραγματική έξοδο. Με αυτόν τον τρόπο αξιολογούμε το μοντέλο και βλέπουμε πόσο έχει μάθει από τις εμπειρίες τις οποίες έχει λάβει σαν εκπαιδευτικά δεδομένα κατά την διάρκεια της εκπαίδευσης.



Εικόνα 6: Δεδομένα στην μηχανική μάθηση

### *Παράδειγμα:*

Υπάρχει ο ιδιοκτήτης ενός καταστήματος, ο οποίος έκανε μια έρευνα για την οποία έχει μια μεγάλη λίστα από ερωτήσεις και απαντήσεις τις οποίες ρωτήθηκε από τους πελάτες. Η λίστα με τις ερωτήσεις και απαντήσεις είναι τα **δεδομένα**. Κάθε φορά που θέλει να βγάλει ένα συμπέρασμα, δεν είναι δυνατόν να ανατρέχει κάθε μια ερώτηση από τις χιλιάδες που υπάρχουν διαθέσιμες για να βρει κάτι σχετικό, διότι αυτό αποτελεί μια χρονοβόρα διαδικασία και καθόλου χρήσιμη. Για να λύσει αυτό το πρόβλημα και να κάνει αυτή την δουλειά πιο εύκολη, τα δεδομένα μετατρέπονται μέσω λογισμικού, υπολογισμών, γραφημάτων κ.α. Αυτή η παρεμβολή από υποβληθείσα δεδομένα είναι η **πληροφορία**. Έτσι τα δεδομένα είναι απαραίτητα για την πληροφορία. Η γνώση παίζει ρόλο στην διαφοροποίηση δύο ατόμων που έχουν τις ίδιες πληροφορίες. Στην πραγματικότητα δεν αποτελεί τεχνικό όρο, αλλά συνδέεται με την διαδικασία της ανθρώπινης σκέψης.

### **Ιδιότητες Δεδομένων**

- Όγκος: Κλίμακα δεδομένων. Με τον παγκόσμιο πληθυσμό να αυξάνεται και την τεχνολογική έκθεση, τεράστια δεδομένα δημιουργούνται κάθε δευτερόλεπτο.
- Ποικιλία: Διαφορετικές μορφές δεδομένων – υγεία, εικόνες, βίντεο και ήχος.
- Ταχύτητα: Ρυθμός ροής και δημιουργίας δεδομένων.
- Τιμή: Σημασία των δεδομένων όσον αφορά τις πληροφορίες που μπορούν να συνάγουν οι ερευνητές από αυτά.

Ειλικρίνεια: Ασφάλεια και ορθότητα των δεδομένων στα οποία εργαζόμαστε.(ML | Introduction to Data in Machine Learning, n.d.)

## Κεφάλαιο 3: Επεξεργασία Φυσικής Γλώσσας

### 3.1 Εισαγωγή

Όλα όσα εκφράζουμε (είτε προφορικά είτε γραπτά) φέρουν τεράστιες ποσότητες πληροφοριών. Το θέμα που επιλέγουμε, ο τόνος μας, η επιλογή λέξεων, τα πάντα προσθέτει κάποιο είδος πληροφορίας το οποίο μπορεί να ερμηνευτεί με διάφορους τρόπους. Θεωρητικά, μπορούμε να κατανοήσουμε και ακόμη να προβλέψουμε την ανθρώπινη συμπεριφορά χρησιμοποιώντας αυτές τις πληροφορίες. Αλλά υπάρχει ένα πρόβλημα: ένα άτομο μπορεί να δημιουργήσει εκατοντάδες ή χιλιάδες λέξεις σε μια δήλωση, όπου κάθε πρόταση έχει την αντίστοιχη πολυπλοκότητα. Εάν θέλετε να κλιμακώσετε και να αναλύσετε αρκετές εκατοντάδες, χιλιάδες ή εκατομμύρια ανθρώπους ή δηλώσεις σε μια δεδομένη περιοχή, τότε οδηγούμαστε σε μια κατάσταση η οποία είναι δύσκολα διαχειρίσιμη από τον άνθρωπο. Τα δεδομένα που παράγονται από συνομιλίες, δηλώσεις ή ακόμη και tweets είναι παραδείγματα μη δομημένων δεδομένων. Τα μη δομημένα δεδομένα δεν χωράνε σωστά στην παραδοσιακή δομή γραμμών και στηλών των σχεσιακών βάσεων δεδομένων και αντιπροσωπεύουν τη συντριπτική πλειονότητα των διαθέσιμων δεδομένων στον πραγματικό κόσμο. Είναι μπερδεμένα και είναι μια διαδικασία που απαιτεί πολύ επεξεργασία για να τα χειριστεί κανείς. Ωστόσο, χάρη στις προόδους σε κλάδους όπως η μηχανική μάθηση, συντελείται μια μεγάλη επανάσταση σε αυτό το θέμα. Σήμερα, δεν πρόκειται πλέον για την προσπάθεια ερμηνείας ενός κειμένου ή μιας ομιλίας με βάση τις λέξεις-κλειδιά του (του παλιού μηχανικού τρόπου), αλλά για την κατανόηση του νοήματος πίσω από αυτές τις λέξεις (του γνωστικού τρόπου). Με αυτόν τον τρόπο είναι δυνατό να ανιχνευθούν μεταφορικές φράσεις όπως ειρωνεία, ή ακόμη να γίνει ανάλυση συναισθήματος.

### 3.2 Ορισμός

Η επεξεργασία φυσικής γλώσσας (Natural Language Processing - NLP) είναι ένα πεδίο της τεχνητής νοημοσύνης, το οποίο δίνει στις μηχανές τη δυνατότητα να διαβάζουν, να κατανοούν και να βγάζουν κάποιο νόημα από τις ανθρώπινες γλώσσες.

Πρόκειται για έναν κλάδο που επικεντρώνεται στην αλληλεπίδραση μεταξύ της επιστήμης των δεδομένων και της ανθρώπινης γλώσσας, και επεκτείνεται σε πολλές βιομηχανίες. Σήμερα, η NLP σημειώνει ραγδαία άνοδο χάρη στις τεράστιες βελτιώσεις στην πρόσβαση σε δεδομένα και

στην αύξηση της υπολογιστικής ισχύος, οι οποίες επιτρέπουν στους επαγγελματίες να επιτύχουν σημαντικά αποτελέσματα σε τομείς όπως η υγειονομική περίθαλψη, τα μέσα ενημέρωσης, η χρηματοδότηση και οι ανθρώπινοι πόροι κ.α.

### 3.3 Χρήση NLP

Με απλά λόγια, η NLP αντιπροσωπεύει τον αυτόματο χειρισμό της φυσικής ανθρώπινης γλώσσας, όπως του προφορικού λόγου ή του κειμένου.

Η NLP μπορεί να σας βοηθήσει με πολλές εργασίες και τα πεδία εφαρμογής φαίνεται να αυξάνονται σε καθημερινή βάση. Ας αναφέρουμε ορισμένα παραδείγματα:

- Η NLP επιτρέπει την αναγνώριση και την πρόβλεψη ασθενειών με βάση τα ηλεκτρονικά μητρώα υγείας και την ομιλία του ίδιου του ασθενούς. Αυτή η ικανότητα διερευνάται σε συνθήκες υγείας που κυμαίνονται από καρδιαγγειακές παθήσεις έως κατάθλιψη, ακόμη και σχιζοφρένεια. Για παράδειγμα, το Amazon Comhend Medical είναι μια υπηρεσία που χρησιμοποιεί NLP για την εξαγωγή παθήσεων, φαρμάκων και αποτελεσμάτων θεραπείας από σημειώσεις ασθενών, αναφορές κλινικών δοκιμών και άλλα ηλεκτρονικά αρχεία υγείας.
- Οι οργανισμοί μπορούν να προσδιορίσουν τι λένε οι πελάτες για μια υπηρεσία ή ένα προϊόν εντοπίζοντας και εξάγοντας πληροφορίες σε πηγές όπως τα μέσα κοινωνικής δικτύωσης. Αυτή η ανάλυση συναισθήματος μπορεί να παρέχει πολλές πληροφορίες σχετικά με τις επιλογές των πελατών.
- Ένας εφευρέτης στην IBM ανέπτυξε έναν γνωστικό βοηθό που λειτουργεί σαν εξατομικευμένη μηχανή αναζήτησης μαθαίνοντας τα πάντα για εσάς και στη συνέχεια σας θυμίζει ένα όνομα, ένα τραγούδι ή οτιδήποτε δεν μπορείτε να θυμηθείτε τη στιγμή που το χρειάζεστε.
- Εταιρείες όπως η Yahoo και η Google φιλτράρουν και ταξινομούν τα email σας με NLP αναλύοντας κείμενο σε μηνύματα ηλεκτρονικού ταχυδρομείου που υπάρχουν στους διακομιστές τους και διακόπτουν την ανεπιθύμητη αλληλογραφία πριν καν εισέλθουν στα Εισερχόμενά σας.

- Για να βοηθήσει στον εντοπισμό πλαστών ειδήσεων, η ομάδα NLP στο MIT ανέπτυξε ένα νέο σύστημα για να καθορίσει εάν μια πηγή είναι ακριβής ή πολιτικά μεροληπτική, εντοπίζοντας εάν μια πηγή ειδήσεων μπορεί να θεωρηθεί αξιόπιστη ή όχι.
- Η Αλέξα της Amazon και η Siri της Apple είναι παραδείγματα έξυπνων φωνητικών προγραμμάτων που χρησιμοποιούν την NLP για να απαντήσουν σε φωνητικές ερωτήσεις και να κάνουν τα πάντα όπως να βρουν ένα συγκεκριμένο κατάστημα, να μας πουν την πρόγνωση καιρού, να προτείνουν την καλύτερη διαδρομή για το γραφείο ή να ανάψουν τα φώτα στο σπίτι.
- Το να έχουμε μια εικόνα για το τι συμβαίνει και για τι μιλούν οι άνθρωποι μπορεί να είναι πολύ πολύτιμο για κάποιες εταιρίες. Η NLP χρησιμοποιείται για την παρακολούθηση ειδήσεων, αναφορών, σχολίων σχετικά με τις δραστηριότητες μιας εταιρίας. Αυτές οι πληροφορίες μπορούν να ενσωματωθούν σε έναν αλγόριθμο συναλλαγών για να παράγουν τεράστια κέρδη.
- Η NLP χρησιμοποιείται επίσης τόσο στο στάδιο της έρευνας όσο και στο στάδιο της επιλογής όσον αφορά την εργασία προσλήψεων ταλέντων. Αυτό γίνεται με τον προσδιορισμό των δεξιοτήτων των υποψηφίων και την πρόβλεψη των προοπτικών τους πριν αρχίσουν να δραστηριοποιούνται στην αγορά εργασίας.

Η NLP σημειώνει ιδιαίτερη άνθηση στον κλάδο υγείας. Αυτή η τεχνολογία βελτιώνει την παροχή φροντίδας, τη διάγνωση ασθενειών και μειώνει το κόστος, ενώ οι οργανισμοί υγειονομικής περίθαλψης χρησιμοποιούν όλο και περισσότερο τα ηλεκτρονικά μητρώα υγείας. Το γεγονός ότι η κλινική τεκμηρίωση μπορεί να βελτιωθεί σημαίνει ότι οι ασθενείς μπορούν να κατανοηθούν καλύτερα και να ωφεληθούν μέσω καλύτερης υγειονομικής περίθαλψης. Ο στόχος θα πρέπει να είναι η βελτιστοποίηση της εμπειρίας τους, και αρκετοί οργανισμοί εργάζονται ήδη σε αυτό. Εταιρείες όπως η Winterlight Labs βελτιώνουν σημαντικά τη θεραπεία της νόσου του Αλτσχάιμερ παρακολουθώντας τη νοητική δυσλειτουργία μέσω του προφορικού λόγου και μπορούν επίσης να υποστηρίξουν κλινικές δοκιμές και μελέτες για ένα ευρύ φάσμα διαταραχών του κεντρικού νευρικού συστήματος. Ακολουθώντας μια παρόμοια προσέγγιση, το

Πανεπιστήμιο του Στάνφορντ ανέπτυξε το Woebot, ένα ρομπότ θεραπευτή, το οποίο έχει σκοπό να βοηθήσει ανθρώπους με άγχος και άλλες διαταραχές.

Αλλά υπάρχει σοβαρή διαμάχη γύρω από το θέμα. Πριν από μερικά χρόνια, η Microsoft απέδειξε ότι αναλύοντας μεγάλα δείγματα ερωτημάτων μηχανών αναζήτησης, μπορούσαν να αναγνωρίσουν χρήστες του διαδικτύου που πάσχουν από καρκίνο του παγκρέατος ακόμη και πριν λάβουν διάγνωση της νόσου. Πώς θα αντιδρούσαν οι χρήστες σε μια τέτοια διάγνωση; Και τι θα συνέβαινε αν εξεταζόσουν ως ψευδές θετικό αποτέλεσμα; (που σημαίνει ότι μπορεί να διαγνωστεί με την ασθένεια ακόμα και αν δεν την έχετε). Αυτό θυμίζει την περίπτωση των Τάσεων Google Flu, οι οποίες το 2009 ανακοινώθηκαν ότι ήταν σε θέση να προβλέψουν τη γρίπη, αλλά αργότερα εξαφανίστηκαν λόγω της χαμηλής ακρίβειας και της αδυναμίας της να ανταποκριθεί στα προβλεπόμενα ποσοστά. Η NLP μπορεί να είναι το κλειδί για μια αποτελεσματική κλινική υποστήριξη στο μέλλον, αλλά υπάρχουν ακόμα πολλές προκλήσεις που πρέπει να αντιμετωπιστούν βραχυπρόθεσμα. (Yse, 2019)

### 3.4 Κατανόηση της Γλώσσας από τους υπολογιστές

Όσο υπάρχουν οι υπολογιστές, οι προγραμματιστές προσπαθούν να γράψουν προγράμματα, τα οποία να καταλαβαίνουν γλώσσες όπως τα αγγλικά. Ο λόγος είναι αρκετά προφανής. οι άνθρωποι γράφουν πράγματα εδώ και χιλιάδες χρόνια και θα ήταν πραγματικά χρήσιμο αν ένας υπολογιστής μπορούσε να διαβάσει και να καταλάβει όλα αυτά τα δεδομένα. Οι υπολογιστές δεν μπορούν ακόμα να καταλάβουν πραγματικά τα αγγλικά με τον τρόπο που τα καταλαβαίνουν οι άνθρωποι, αλλά μπορούν ήδη να κάνουν πολλά.

Η διαδικασία ανάγνωσης και κατανόησης των αγγλικών είναι πολύ περίπλοκη και αυτό δεν σημαίνει καν ότι τα αγγλικά δεν ακολουθούν λογικούς και συνεπείς κανόνες. Για παράδειγμα, τι σημαίνει αυτός ο τίτλος ειδήσεων:

*“Environmental regulators grill business owner over illegal coal fires.”*

Οι ρυθμιστικές αρχές ανακρίνουν έναν ιδιοκτήτη επιχείρησης για την παράνομη καύση άνθρακα; Ή μήπως οι ρυθμιστές μαγειρεύουν κυριολεκτικά τον ιδιοκτήτη της επιχείρησης; Όπως μπορείτε να δείτε, η ανάλυση αγγλικών με έναν υπολογιστή θα είναι περίπλοκη. Το να κάνει κάποιος οτιδήποτε περίπλοκο στον τομέα της μηχανικής μάθησης συνήθως σημαίνει να



κατασκευάζει έναν "αγωγό". Η ιδέα είναι να χωριστεί ένα πρόβλημά σε πολύ μικρά κομμάτια και στη συνέχεια να χρησιμοποιηθεί η μηχανική μάθηση για να λυθεί κάθε μικρότερο κομμάτι ξεχωριστά. Στη συνέχεια, συνδέοντας διάφορα μοντέλα εκμάθησης μηχανών που τροφοδοτούνται το ένα με το άλλο, μπορούν να γίνουν πιο περίπλοκα πράγματα. Και αυτή ακριβώς είναι η στρατηγική που θα χρησιμοποιήσουμε παρακάτω. Θα αναλύσουμε τη διαδικασία κατανόησης των αγγλικών σε μικρά κομμάτια και θα δούμε πώς λειτουργεί το καθένα.

### 3.5 Τα βήματα της NLP για την κατανόηση των Αγγλικών

Ας δούμε αυτό το απόσπασμα κειμένου από την Wikipedia:

*“London is the capital and most populous city of England and the United Kingdom. Standing on the River Thames in the south east of the island of Great Britain, London has been a major settlement for two millennia. It was founded by the Romans, who named it Londinium.”*

Η παράγραφος αυτή περιέχει πολλά χρήσιμα στοιχεία. Θα ήταν υπέροχο αν ένας υπολογιστής μπορούσε να διαβάσει αυτό το κείμενο και να καταλάβει ότι το Λονδίνο είναι πόλη, το Λονδίνο βρίσκεται στην Αγγλία, το Λονδίνο ιδρύθηκε από τους Ρωμαίους και ούτω καθεξής. Αλλά για να φτάσουμε εκεί, πρέπει πρώτα να διδάξουμε στον υπολογιστή μας τις πιο βασικές έννοιες της γραπτής γλώσσας και μετά να προχωρήσουμε.

Στο υπόλοιπο της παραγράφου θα δούμε τα βήματα που πρέπει να ακολουθήσουμε για να εκτελέσουμε την προαναφερθείσα εργασία.

#### 3.5.1 Βήμα 1: Κατάτμηση Πρότασης

Το πρώτο βήμα του «αγωγού» είναι η διάσπαση του κειμένου σε ξεχωριστές προτάσεις. Αυτό μας δίνει το εξής:

1. *“London is the capital and most populous city of England and the United Kingdom.”*
2. *“Standing on the River Thames in the south east of the island of Great Britain, London has been a major settlement for two millennia.”*
3. *“It was founded by the Romans, who named it Londinium.”*

Μπορούμε να υποθέσουμε ότι κάθε πρόταση στα αγγλικά είναι μια ξεχωριστή σκέψη ή ιδέα. Θα είναι πολύ πιο εύκολο να γραφεί ένα πρόγραμμα για να κατανοηθεί μία πρόταση παρά να κατανοηθεί μια ολόκληρη παράγραφος. Η κωδικοποίηση ενός μοντέλου «Κατάτμηση πρότασης» μπορεί να είναι τόσο απλή όσο η διαίρεση προτάσεων κάθε φορά που βλέπετε ένα σημείο στίξης. Αλλά οι σύγχρονοι αλγόριθμοι NLP συχνά χρησιμοποιούν πιο σύνθετες τεχνικές που λειτουργούν ακόμα και όταν ένα έγγραφο δεν έχει μορφοποιηθεί καλά.

### 3.5.2 Βήμα 2: Εξαγωγή Συμβόλων

Τώρα που χωρίσαμε το έγγραφό μας σε προτάσεις, μπορούμε να τις επεξεργαστούμε μία-μία. Ας ξεκινήσουμε με την πρώτη πρόταση του εγγράφου μας:

*“London is the capital and most populous city of England and the United Kingdom.”*

Το επόμενο βήμα στον αλγόριθμό μας είναι να σπάσουμε αυτή την πρόταση σε ξεχωριστές λέξεις ή διακριτικά. Αυτό ονομάζεται «εξαγωγή συμβόλων». Αυτό είναι το αποτέλεσμα:

*“London”, “is”, “the”, “capital”, “and”, “most”, “populous”, “city”, “of”, “England”, “and”, “the”, “United”, “Kingdom”, “.”*

Η δημιουργία συμβόλων είναι εύκολη στα αγγλικά. Απλά θα χωρίσουμε τις λέξεις όποτε υπάρχει κενό ανάμεσά τους. Και θα αντιμετωπίσουμε επίσης τα σημεία στίξης ως ξεχωριστά διακριτικά, καθώς η στίξη έχει επίσης νόημα.

### 3.5.3 Βήμα 3: Πρόβλεψη του μέρους του λόγου για κάθε σύμβολο

Στη συνέχεια, θα κοιτάμε κάθε σύμβολο και θα προσπαθήσουμε να μαντέψουμε το μέρος του λόγου του, δηλαδή αν πρόκειται για ουσιαστικό, ρήμα, επίθετο και ούτω καθεξής. Η γνώση του ρόλου κάθε λέξης στην πρόταση θα μας βοηθήσει να αρχίσουμε να καταλαβαίνουμε για τι μιλάει η πρόταση. Μπορούμε να το κάνουμε αυτό ταΐζοντας κάθε λέξη (και μερικές επιπλέον λέξεις γύρω της για το περιβάλλον) σε ένα προ εκπαιδευμένο μοντέλο ταξινόμησης κατά μέρος του λόγου:



Το μοντέλο εύρεσης του μέρους του λόγου εκπαιδεύτηκε αρχικά δίνοντάς του εκατομμύρια αγγλικές προτάσεις με το μέρος του λόγου κάθε λέξης να έχει ήδη επισημανθεί και με βάση αυτά τα δεδομένα το μοντέλο έμαθε να αναπαράγει αυτή τη συμπεριφορά.

Λάβετε υπόψη ότι το μοντέλο βασίζεται πλήρως στα στατιστικά στοιχεία. Στην πραγματικότητα δεν καταλαβαίνει τι σημαίνουν οι λέξεις με τον ίδιο τρόπο όπως οι άνθρωποι. Ξέρει απλώς πώς να μαντέψει ένα μέρος του λόγου βάσει παρόμοιων προτάσεων και λέξεων που έχει δει στο παρελθόν. Μετά την επεξεργασία ολόκληρης της πρότασης, θα έχουμε ένα αποτέλεσμα όπως αυτό:

<b>London</b>	<b>is</b>	<b>the</b>	<b>capital</b>	<b>and</b>	<b>most</b>	<b>populous ...</b>
Proper Noun	Verb	Determiner	Noun	Conjunction	Adverb	Adjective

Με αυτές τις πληροφορίες, μπορούμε ήδη να αρχίσουμε να δίνουμε ένα πολύ βασικό νόημα. Για παράδειγμα, μπορούμε να δούμε ότι τα ουσιαστικά της πρότασης περιλαμβάνουν το “London” και το “Capital”, οπότε η πρόταση μάλλον μιλάει για το Λονδίνο.

#### 3.5.4 Βήμα 4: Καταχώρηση κειμένου

Στα αγγλικά (και στις περισσότερες γλώσσες), οι λέξεις εμφανίζονται με διαφορετικές μορφές. Δείτε αυτές τις δύο προτάσεις:

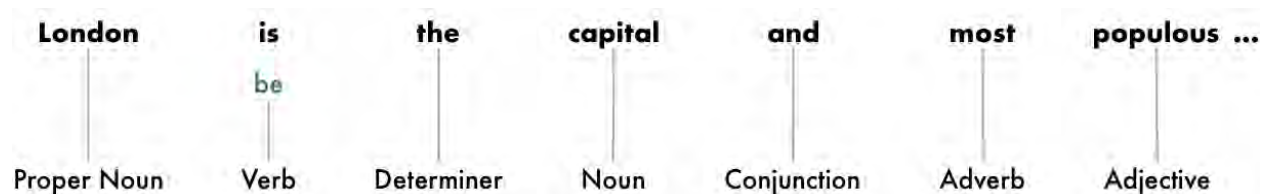
*I had a pony.*

*I had two ponies.*

Και οι δύο προτάσεις μιλούν για το ουσιαστικό πόνυ, αλλά χρησιμοποιούν διαφορετική κλίση. Όταν εργάζεστε με κείμενο σε έναν υπολογιστή, είναι χρήσιμο να γνωρίζετε τη βασική μορφή κάθε λέξης έτσι ώστε να γνωρίζετε ότι και οι δύο προτάσεις μιλούν για την ίδια έννοια.

Διαφορετικά, ο υπολογιστής καταλαβαίνει τις λέξεις “pony” και “ponies” σαν δύο τελείως διαφορετικές λέξεις. Στην NLP, καλούμε αυτή την διαδικασία ληματοποίηση, δηλαδή τον προσδιορισμό της πιο βασικής μορφής ή του λήμματος κάθε λέξης της πρότασης. Το ίδιο ισχύει και για τα ρήματα. Μπορούμε επίσης να αποτιμήσουμε τα ρήματα βρίσκοντας τη ρίζα τους, μη συζευγμένη μορφή τους. Οπότε το “I had two ponies” γίνεται “I [have] two [pony]”. Η ληματοποίηση γίνεται συνήθως με την ύπαρξη ενός πίνακα αναζήτησης των μορφών λήμματος των λέξεων με βάση το μέρος του λόγου τους και πιθανώς με την ύπαρξη κάποιων συνηθισμένων κανόνων για το χειρισμό λέξεων που δεν έχετε ξαναδεί.

Ας δούμε πώς φαίνεται η πρόταση μας αφότου η ληματοποίηση προσθέσει τη ριζική μορφή του ρήματος μας:

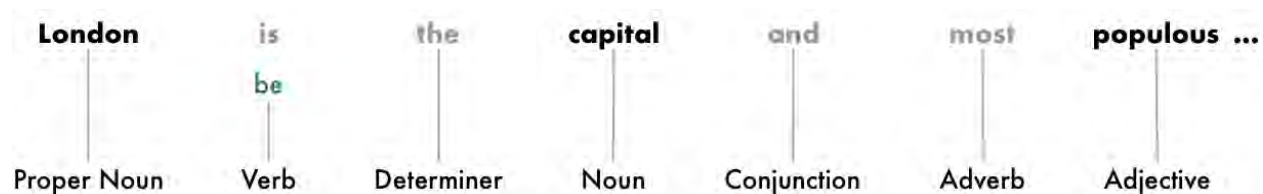


Η μόνη αλλαγή που κάναμε ήταν να μετατρέψουμε το “is” σε “be”.

### 3.5.5 Βήμα 5: Αναγνώριση Διακοπτόμενων Λέξεων

Στη συνέχεια, θέλουμε να εξετάσουμε τη σημασία κάθε λέξης στην πρόταση. Τα αγγλικά έχουν πολλές λέξεις που συχνά εμφανίζονται όπως “and”, “the” και “a”. Κατά τη στατιστική σχετική με το κείμενο, αυτές οι λέξεις εισάγουν πολύ θόρυβο, καθώς εμφανίζονται πολύ συχνότερα από άλλες λέξεις. Ορισμένοι αγωγοί NLP θα τους επισημάνουν ως λέξεις διακοπής (stopwords) — δηλαδή, λέξεις που μπορεί να θέλετε να φιλτράρετε πριν κάνετε οποιαδήποτε στατιστική ανάλυση.

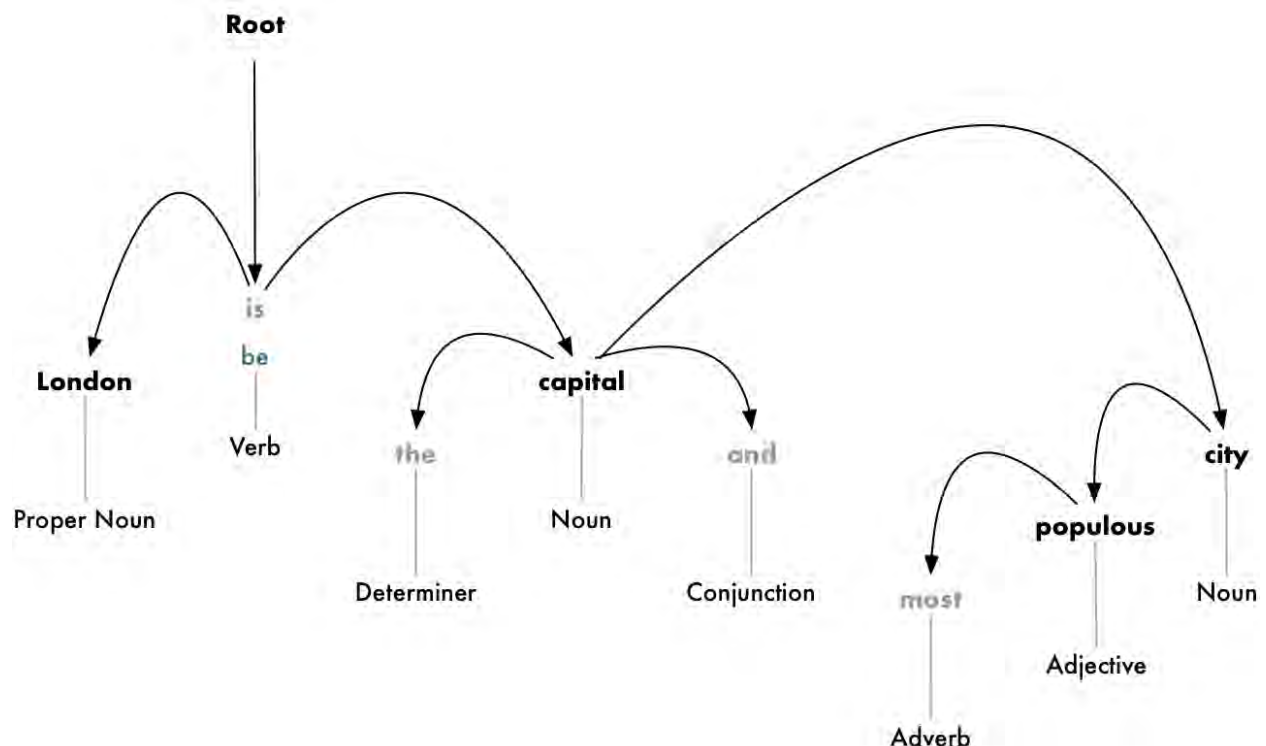
Παρακάτω βλέπουμε πώς φαίνεται η πρόταση μας με τις λέξεις διακοπής που αποκαλύπτονται:



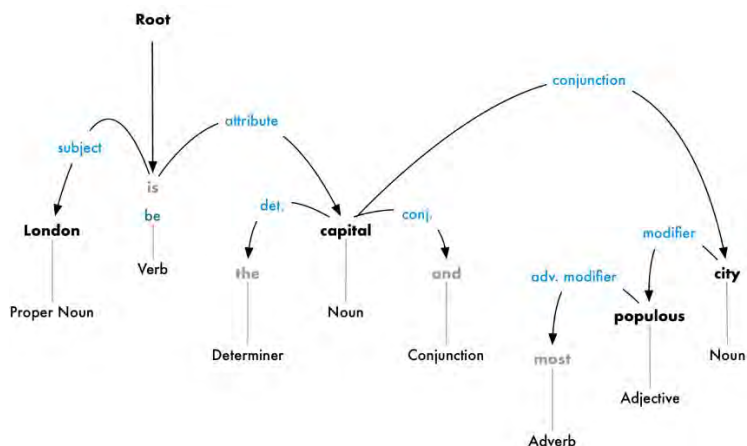
Οι λέξεις διακοπής προσδιορίζονται συνήθως με έλεγχο μιας λίστας γνωστών διακοπόμενων λέξεων. Αλλά δεν υπάρχει τυποποιημένος κατάλογος διακοπόμενων λέξεων που να είναι κατάλληλος για όλες τις εφαρμογές. Η λίστα των λέξεων που θα αγνοήσετε μπορεί να διαφέρει ανάλογα με την εφαρμογή σας. Για παράδειγμα, αν δημιουργείτε μια μηχανή αναζήτησης συγκροτημάτων, θέλετε να βεβαιωθείτε ότι δεν αγνοείτε τη λέξη “The”. Επειδή όχι μόνο η λέξη “The” εμφανίζεται σε πολλά ονόματα συγκροτημάτων, υπάρχει ένα διάσημο ροκ συγκρότημα του 1980 που ονομάζεται The!

### 3.5.6 Ανάλυση Εξαρτήσεων

Το επόμενο βήμα είναι να καταλάβουμε πώς όλες οι λέξεις στην πρόταση σχετίζονται μεταξύ τους. Αυτό ονομάζεται ανάλυση εξαρτήσεων. Στόχος είναι να δημιουργηθεί ένα δέντρο που εκχωρεί μία γονική λέξη σε κάθε λέξη της πρότασης. Η ρίζα του δέντρου θα είναι το κύριο ρήμα της πρότασης. Παρακάτω βλέπουμε πώς θα φαίνεται η αρχή του δέντρου ανάλυσης για την πρόταση μας:



Αλλά μπορούμε να πάμε ένα βήμα παραπέρα. Εκτός από τον προσδιορισμό της γονικής λέξης κάθε λέξης, μπορούμε επίσης να προβλέψουμε τον τύπο της σχέσης που υπάρχει μεταξύ αυτών των δύο λέξεων:



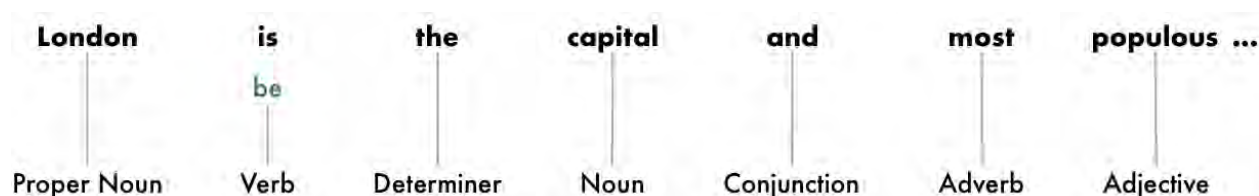
Αυτό το δέντρο ανάλυσης μας δείχνει ότι το θέμα της πρότασης είναι το ουσιαστικό “London” και έχει μια σχέση “Be” με το “capital”. Επιτέλους ξέρουμε κάτι χρήσιμο. Το Λονδίνο είναι πρωτεύουσα! Και αν ακολουθούσαμε το πλήρες δέντρο ανάλυσης για την πρόταση (πέρα από αυτό που φαίνεται), θα ανακαλύπταμε ακόμα και ότι το Λονδίνο είναι η πρωτεύουσα του Ηνωμένου Βασιλείου. Όπως ακριβώς προβλέψαμε τα μέρη του λόγου νωρίτερα χρησιμοποιώντας ένα μοντέλο μηχανικής μάθησης, η ανάλυση εξαρτήσεων λειτουργεί επίσης τροφοδοτώντας λέξεις σε ένα μοντέλο μηχανικής μάθησης και βγάζοντας ένα αποτέλεσμα. Ωστόσο, η ανάλυση των εξαρτήσεων των λέξεων είναι ιδιαίτερα πολύπλοκη εργασία και θα απαιτούσε ένα ολόκληρο κεφάλαιο για να εξηγηθεί λεπτομερώς. Το 2016, η Google κυκλοφόρησε έναν νέο αναλυτή εξαρτήσεων, με το όνομα Parsey McParseface, ο οποίος ξεπέρασε τα προηγούμενα σημεία αναφοράς χρησιμοποιώντας μια νέα προσέγγιση βαθιάς μάθησης που γρήγορα εξαπλώθηκε σε όλο τον κλάδο. Έπειτα ένα χρόνο αργότερα, κυκλοφόρησαν ένα ακόμα νεότερο μοντέλο που ονομάζεται ParseySaurus, το οποίο βελτίωσε τα πράγματα περαιτέρω. Με άλλα λόγια, οι τεχνικές ανάλυσης εξακολουθούν να αποτελούν έναν ενεργό τομέα έρευνας και συνεχώς μεταβαλλόμενες και βελτιούμενες. Είναι επίσης σημαντικό να θυμόμαστε ότι πολλές αγγλικές προτάσεις είναι ασαφείς και πραγματικά δύσκολο να αναλυθούν. Σε αυτές τις περιπτώσεις, το μοντέλο θα κάνει μια εικασία με βάση ποια αναλυμένη εκδοχή της πρότασης φαίνεται πιθανότερη, αλλά αυτή η διαδικασία δεν είναι τέλεια και μερικές

φορές το μοντέλο θα είναι λάθος. Αλλά με την πάροδο του χρόνου τα μοντέλα NLP θα συνεχίσουν να βελτιώνονται στην εκτέλεση της λειτουργίας της ανάλυση του κειμένου.

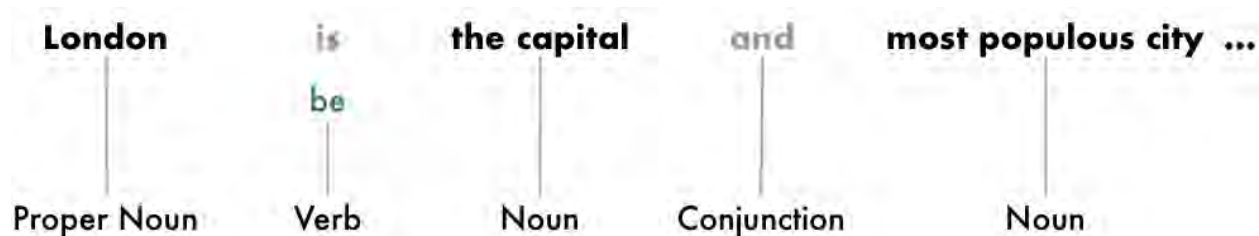
### 3.5.7 Βήμα 6β: Εύρεση Φράσεων Ουσιαστικών

Μέχρι στιγμής, έχουμε αντιμετωπίσει κάθε λέξη της πρότασης μας ως ξεχωριστή οντότητα. Αλλά μερικές φορές είναι πιο λογικό να ομαδοποιούμε τις λέξεις που αντιπροσωπεύουν μία ιδέα ή ένα πράγμα. Μπορούμε να χρησιμοποιήσουμε τις πληροφορίες από το δέντρο ανάλυσης εξαρτήσεων για να ομαδοποιήσουμε αυτόματα λέξεις που μιλούν για το ίδιο πράγμα.

Για παράδειγμα, αντί για αυτό:



Μπορούμε να ομαδοποιήσουμε τις νέες φράσεις ουσιαστικών για να το δημιουργήσουμε αυτό:



Το αν θα κάνουμε αυτό το βήμα ή όχι εξαρτάται από τον τελικό μας στόχο. Αλλά είναι συχνά ένας γρήγορος και εύκολος τρόπος για να απλοποιήσουμε την πρόταση εάν δεν χρειαζόμαστε επιπλέον λεπτομέρειες για το ποιες λέξεις είναι επίθετα και αντ' αυτού ενδιαφερόμαστε περισσότερο για την εξαγωγή ολοκληρωμένων ιδεών.

### 5.5.8 Βήμα 7: Αναγνώριση Ονοματικής Οντότητας (NER – Named Entity Recognition)

Τώρα που έχουμε κάνει όλη αυτή τη διαδικασία, μπορούμε επιτέλους να προχωρήσουμε πέρα από τη γραμματική του δημοτικού και να αρχίσουμε να εξάγουμε ιδέες. Στην πρόταση μας, έχουμε τα ακόλουθα ουσιαστικά:

**London** **is** **the** **capital** **and** **most** **populous** **city** **of** **England** **and** **the** **United Kingdom**.

**London** is the capital and most populous city of **England** and the **United Kingdom**.

Geographic Entity
Geographic Entity
Geographic Entity

Ακολουθούν μερικά από τα είδη αντικειμένων που μπορεί να επισημάνει ένα τυπικό σύστημα NER:

- Το NER έχει πολλές χρήσεις, καθώς διευκολύνει την εξαγωγή δομημένων δεδομένων από το κείμενο.



### 5.5.9 Βήμα 8: Ανάλυση Αναφοράς

Σε αυτό το σημείο, έχουμε ήδη μια χρήσιμη αναπαράσταση της πρότασης μας. Γνωρίζουμε τα μέρη του λόγου για κάθε λέξη, πώς σχετίζονται οι λέξεις μεταξύ τους και ποιες λέξεις μιλούν για ονομαστικές οντότητες. Ωστόσο, εξακολουθούμε να έχουμε ένα μεγάλο πρόβλημα. Τα αγγλικά είναι γεμάτα αντωνυμίες, δηλαδή λέξεις όπως αυτός, αυτή, και αυτό. Αυτές είναι συντομεύσεις που χρησιμοποιούμε αντί να γράφουμε ονόματα ξανά και ξανά σε κάθε πρόταση. Οι άνθρωποι μπορούν να παρακολουθούν τι αντιπροσωπεύουν αυτές οι λέξεις με βάση το περιεχόμενο. Αλλά το μοντέλο NLP μας δεν γνωρίζει τι σημαίνουν οι αντωνυμίες, επειδή εξετάζει μόνο μία πρόταση τη φορά. Ας δούμε την τρίτη πρόταση στο έγγραφό μας:

*“It was founded by the Romans, who named it Londinium.”*

Αν το αναλύσουμε αυτό με τον αγωγό NLP μας, θα ξέρουμε ότι το "αυτό" ιδρύθηκε από τους Ρωμαίους. Αλλά είναι πολύ πιο χρήσιμο να γνωρίζουμε ότι το "Λονδίνο" ιδρύθηκε από τους Ρωμαίους. Ως άνθρωπος που διαβάζει αυτή την πρόταση, μπορείς εύκολα να καταλάβεις ότι "αυτό" σημαίνει "Λονδίνο". Ο στόχος της ανάλυσης αναφοράς είναι να βρεθεί αυτή η ίδια αντιστοίχιση με την παρακολούθηση προφορών σε προτάσεις. Θέλουμε να καταλάβουμε όλες τις λέξεις που αναφέρονται στην ίδια οντότητα.

Ιδού το αποτέλεσμα της Ανάλυσης Αναφοράς στο έγγραφό μας για τη λέξη "London":

**London** is the capital and most populous city of England and the United Kingdom. Standing on the River Thames in the south east of the island of Great Britain, **London** has been a major settlement for two millennia. **It** was founded by the Romans, who named it Londinium.

Με τις πληροφορίες αναφοράς σε συνδυασμό με το δέντρο ανάλυσης και τις πληροφορίες ονοματισμένης οντότητας, θα πρέπει να μπορούμε να εξάγουμε πολλές πληροφορίες από το έγγραφο. Το ψήφισμα περί παραπομπής είναι ένα από τα δυσκολότερα βήματα που πρέπει να υλοποιήσουμε. Είναι ακόμα πιο δύσκολο από την ανάλυση πρότασης. Οι πρόσφατες πρόοδοι στη βαθιά μάθηση οδήγησαν σε νέες προσεγγίσεις, οι οποίες είναι πιο ακριβείς, αλλά δεν είναι ακόμα τέλειες.

Αφού είδαμε πως λειτουργεί κάθε βήμα στο Pipeline του NLP ας δούμε ένα σχεδιάγραμμα με τα συνολικά βήματα που πρέπει να ακολουθήσει κάποιος:



(Geitgey, 2018)

# Κεφάλαιο 4: Νευρωνικά Δίκτυα

## 4.1 Εισαγωγή και Ορισμός

### 4.1.1 Γενικά Στοιχεία

Στη βαθιά μάθηση, ένα μοντέλο υπολογιστή μαθαίνει να εκτελεί εργασίες ταξινόμησης απευθείας από εικόνες, κείμενο ή ήχο. Τα μοντέλα βαθιάς μάθησης μπορούν να επιτύχουν κορυφαία ακρίβεια, μερικές φορές υπερβαίνοντας τις επιδόσεις των ανθρώπων. Τα μοντέλα εκπαιδεύονται χρησιμοποιώντας ένα μεγάλο σύνολο επισημασμένων δεδομένων και αρχιτεκτονικών νευρωνικού δικτύου που περιέχουν πολλά επίπεδα.

Τα μοντέλα βαθιάς μάθησης μπορούν να χρησιμοποιηθούν για μια ποικιλία σύνθετων εργασιών:

- Τεχνητά νευρωνικά δίκτυα (ANN - Artificial Neural Networks) για παλινδρόμηση και ταξινόμηση
- Συνελκτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks - CNN) για μηχανική όραση
- Αναδρομικά νευρωνικά δίκτυα (Recurrent Neural Networks - RNN) για ανάλυση χρονολογικών σειρών
- Αυτοοργανωμένοι χάρτες για την εξαγωγή δυνατοτήτων
- Μηχανές Deep Boltzmann για συστήματα προτάσεων
- Αυτόματοι κωδικοποιητές για συστήματα προτάσεων

### 4.1.2 Ορισμός

Τα Τεχνητά Νευρωνικά Δίκτυα είναι ένα πρότυπο επεξεργασίας πληροφοριών που εμπνέεται από τον τρόπο με τον οποίο το βιολογικό νευρικό σύστημα, όπως ο εγκέφαλος, επεξεργάζεται πληροφορίες. Αποτελείται από μεγάλο αριθμό ιδιαίτερα διασυνδεδεμένων στοιχείων επεξεργασίας (νευρώνες) που εργάζονται από κοινού για την επίλυση ενός συγκεκριμένου προβλήματος.

Στην συνέχεια του κεφαλαίου θα αναφερθούμε στις εξής έννοιες:

1. Νευρώνες
2. Συναρτήσεις ενεργοποίησης
3. Τύποι συναρτήσεων ενεργοποίησης
4. Τρόπος Λειτουργίας Νευρωνικών Δικτύων
5. Τρόπος εκμάθησης Νευρωνικών Δικτύων(Μετάδοση προς τα πίσω)
6. Κλίση
7. Στοχαστική κλίση
8. Εκπαίδευση Νευρωνικού Δικτύου με στοχαστική κλίση

## 4.2 Νευρώνες

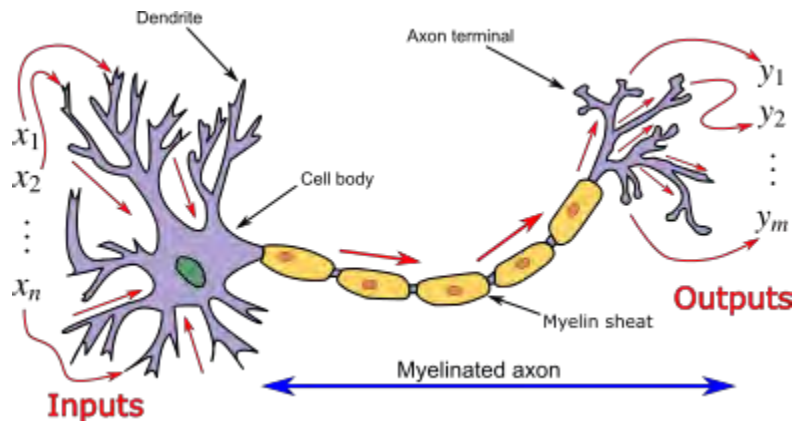
Οι Βιολογικοί Νευρώνες (που ονομάζονται επίσης νευρικά κύτταρα) ή απλά οι νευρώνες είναι οι βασικές μονάδες του εγκεφάλου και του νευρικού συστήματος, τα κύτταρα που είναι υπεύθυνα για τη λήψη αισθητήριων εισροών από τον εξωτερικό κόσμο, για την επεξεργασία τους έξοδο τους αποτελέσματος της επεξεργασίας τους.

Ένας βιολογικός νευρώνας περιέχει:

- **Σώμα κυττάρων:** Το σώμα του πυρήνα του νευρώνα περιέχει τον πυρήνα και πραγματοποιεί βιοχημικό μετασχηματισμό που είναι απαραίτητος για τη ζωή των νευρώνων.
- **Δενδρίτες:** Κάθε νευρώνας έχει λεπτές, τριχοειδείς σωληνωτές δομές (επεκτάσεις) γύρω του. Διακλαδώνονται σε ένα δέντρο γύρω από το σώμα του κελιού. Δέχονται εισερχόμενα σήματα.
- **Άξονας:** Είναι μια μεγάλη, λεπτή, με δομή σωλήνα που λειτουργεί σαν γραμμή μετάδοσης.
- **Συνάψεις:** Οι νευρώνες συνδέονται μεταξύ τους σε μια πολύπλοκη χωρική διάταξη. Όταν ο άξονας φτάσει στον τελικό του προορισμό, υποδιαιρείται πάλι. Στο τέλος του

άξονα υπάρχουν εξαιρετικά σύνθετες και εξειδικευμένες δομές που ονομάζονται συνάψεις. Η σύνδεση μεταξύ δύο νευρώνων πραγματοποιείται σε αυτές τις συνάψεις.

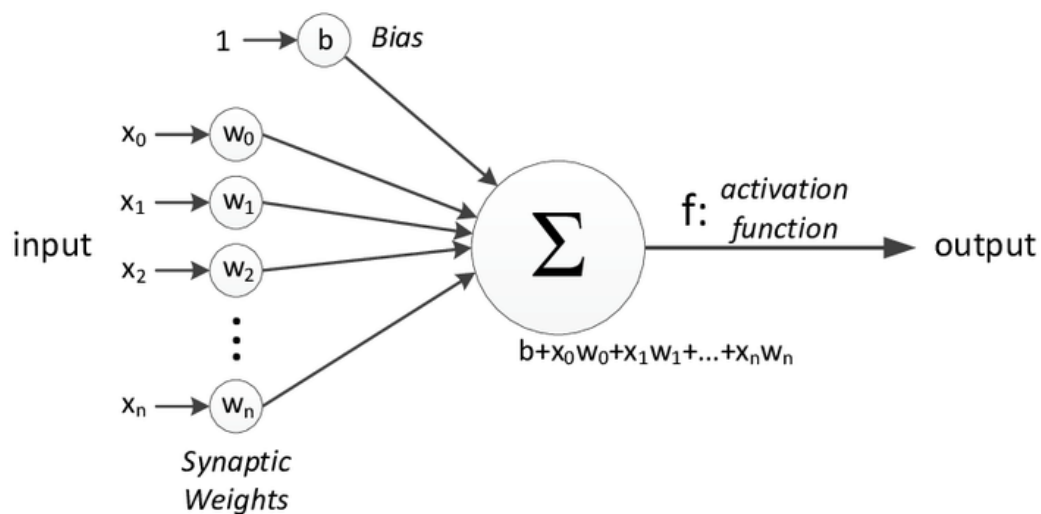
- Οι δενδρίτες λαμβάνουν είσοδο μέσω των συνάψεων άλλων νευρώνων. Ο οργανισμός επεξεργάζεται αυτά τα εισερχόμενα σήματα με την πάροδο του χρόνου και μετατρέπει την επεξεργασμένη τιμή σε έξοδο, η οποία αποστέλλεται σε άλλους νευρώνες μέσω του αξόνων και των συνάψεων.



Εικόνα 1: Η δομή ενός βιολογικού νευρώνα

Το ακόλουθο διάγραμμα απεικονίζει το γενικό μοντέλο ενός τεχνητού νευρωνικού δικτύου, η λειτουργία του οποίου είναι εμπνευσμένη από έναν βιολογικό νευρώνα.

Ένα νευρωνικό δίκτυο ενός επιπέδου δίνει μία μόνο έξοδο.



Στο παραπάνω σχήμα, μία μόνο παρατήρηση,  $x_0, x_1, x_2, x_3 \dots x(n)$  αντιπροσωπεύει διάφορες εισόδους (ανεξάρτητες μεταβλητές) στο δίκτυο. Κάθε μία από αυτές τις εισόδους πολλαπλασιάζονται με ένα βάρος σύνδεσης ή μια συνάρτηση. Τα βάρη απεικονίζονται ως  $w_0, w_1, w_2, w_3 \dots w(n)$ . Το βάρος δείχνει την ισχύ ενός συγκεκριμένου κόμβου.

$b$  είναι μια τιμή πόλωσης. Μια τιμή πόλωσης σάς επιτρέπει να μετακινήσετε την συνάρτηση ενεργοποίησης προς τα πάνω ή προς τα κάτω. Στην απλούστερη περίπτωση, οι τιμές αυτές αθροίζονται, τροφοδοτούνται με μια συνάρτηση ενεργοποίησης για να παράγουν ένα αποτέλεσμα και το αποτέλεσμα αυτό αποστέλλεται ως έξοδο.

Μαθηματικά,  $x_1.w_1 + x_2.w_2 + x_3.w_3 \dots x_n.w_n = \sum x_i.w_i$

Σε αυτή την φάση εφαρμόζεται η συνάρτηση ενεργοποίησης  $\phi(\sum x_i.w_i)$ .

## 4.3 Συνάρτηση Ενεργοποίησης

### 4.3.1 Εισαγωγή και Ορισμός

Η συνάρτηση ενεργοποίησης είναι σημαντική για ένα νευρωνικό δίκτυο να μάθει και να καταλάβει κάτι πραγματικά περίπλοκο. Ο κύριος σκοπός τους είναι να μετατρέψουν ένα σήμα εισόδου ενός κόμβου σε σήμα εξόδου. Αυτό το σήμα εξόδου χρησιμοποιείται ως είσοδος στο επόμενο επίπεδο στη στοίβα.

#### Ορισμός:

«Η συνάρτηση ενεργοποίησης αποφασίζει εάν ένας νευρώνας θα πρέπει να ενεργοποιηθεί ή όχι υπολογίζοντας το σταθμισμένο άθροισμα και προσθέτοντας περαιτέρω πόλωση σε αυτόν. Το κίνητρο είναι να εισαχθεί η μη γραμμικότητα στην έξοδο ενός νευρώνα.»

Αν δεν εφαρμόσουμε τη συνάρτηση ενεργοποίησης, τότε το σήμα εξόδου θα είναι απλά γραμμική συνάρτηση (πολυωνυμική συνάρτηση βαθμού 1). Τώρα, μια γραμμική συνάρτηση είναι εύκολο να επιλυθεί, αλλά είναι περιορισμένες στην πολυπλοκότητά τους, έχοντας λιγότερη ισχύ. Χωρίς συνάρτηση ενεργοποίησης, το μοντέλο μας δεν μπορεί να μάθει και να μοντελοποιήσει περίπλοκα δεδομένα όπως εικόνες, βίντεο, ήχο, ομιλία, κλπ.

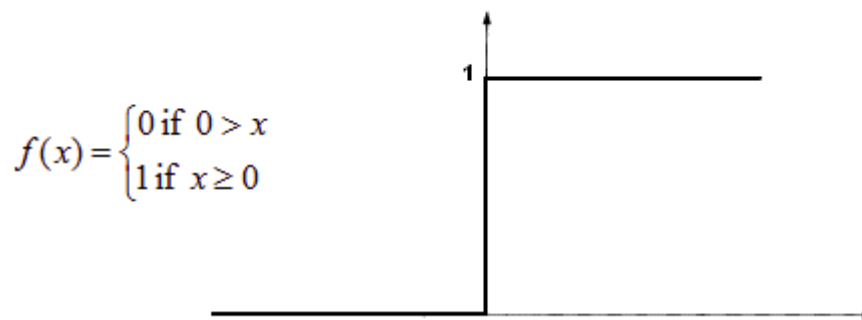
Το θέμα τώρα είναι γιατί χρησιμοποιούμε μη γραμμικότητα:

Μη γραμμικές συναρτήσεις είναι εκείνες που έχουν βαθμό μεγαλύτερο από 1 και έχουν καμπυλότητα. Ένα νευρωνικό δίκτυο χρειάζεται μάθει και να αναπαριστά σχεδόν οτιδήποτε και οποιαδήποτε αυθαίρετη σύνθετη συνάρτηση που χαρτογραφεί μια είσοδο στην έξοδο. Ένα νευρωνικό δίκτυο μπορεί να μάθει και να υπολογίζει οποιαδήποτε συνάρτηση.

#### 4.3.2 Τύποι Συναρτήσεων Ενεργοποίησης

##### 1. Συνάρτηση ενεργοποίησης κατωφλίου — (Συνάρτηση Δυαδικού Βήματος)

Μια Συνάρτηση Δυαδικού Βήματος είναι μια συνάρτηση ενεργοποίησης βάσει κατωφλίου. Εάν η τιμή εισόδου είναι πάνω ή κάτω από ένα συγκεκριμένο όριο, ο νευρώνας ενεργοποιείται και στέλνει ακριβώς το ίδιο σήμα στο επόμενο επίπεδο.



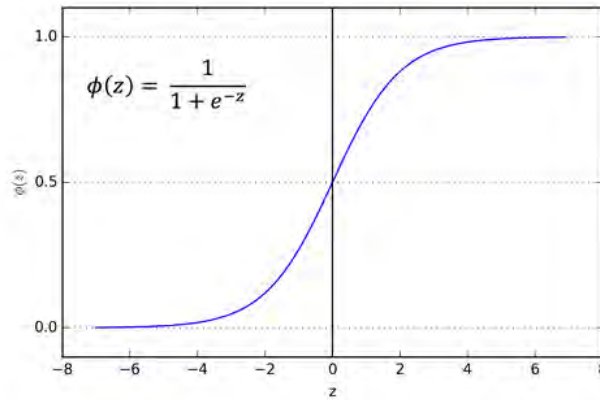
Εικόνα 2: Συνάρτηση Δυαδικού Βήματος

Μια συνάρτηση δυαδικού βήματος:

Συνάρτηση ενεργοποίησης  $A = \text{"ενεργοποιημένη"}$  αν  $Y > \text{κατώφλι}$  αλλιώς όχι ή  $A=1$  εάν  $Y > \text{κατώφλι}$  διαφορετικά 0. Το πρόβλημα με αυτή τη συνάρτηση είναι η δημιουργία ενός δυαδικού κατηγοριοποιητή (1 ή 0), αλλά αν θέλετε πολλαπλοί από αυτούς τους νευρώνες να συνδεθούν για να φέρουν περισσότερες κλάσεις, Κατηγορία1, Κατηγορία2, Κατηγορία3, κ.λπ. Σε αυτήν την περίπτωση, όλοι οι νευρώνες θα δώσουν 1.

##### 2. Σιγμοειδής Συνάρτηση Ενεργοποίησης — (Λογιστική Συνάρτηση)

Μια σιγμοειδής συνάρτηση είναι μια μαθηματική συνάρτηση με χαρακτηριστική καμπύλη σχήματος "S" ή σιγμοειδή καμπύλη που κυμαίνεται μεταξύ 0 και 1, επομένως χρησιμοποιείται για μοντέλα όπου πρέπει να προβλέψουμε την πιθανότητα ως έξοδο.



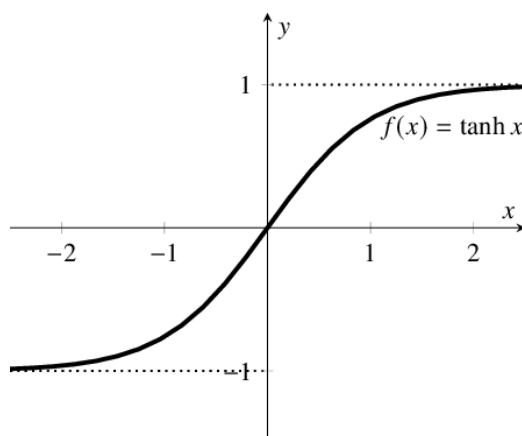
Εικόνα 3: Σιγμοειδής Συνάρτηση

### Σιγμοειδής καμπύλη

Η σιγμοειδής συνάρτηση είναι παραγωγίσιμη, το οποίο σημαίνει ότι μπορούμε να βρούμε την κλίση της καμπύλης σε οποιαδήποτε 2 σημεία. Το μειονέκτημα της σιγμοειδής συνάρτησης ενεργοποίησης είναι ότι μπορεί να προκαλέσει εμπλοκή του νευρωνικού δικτύου κατά την εκπαίδευση, αν παρέχεται ισχυρή αρνητική είσοδος.

### 3. Υπερβολική Συνάρτηση Εφαπτομένης (tanh)

Είναι παρόμοιο με την σιγμοειδή αλλά καλύτερη σε απόδοση. Είναι μη γραμμική στη φύση, οπότε μπορούμε να στοιβάξουμε επίπεδα. Η συνάρτηση κυμαίνεται μεταξύ (- 1,1).



Εικόνα 4: Υπερβολική Συνάρτηση Εφαπτομένης



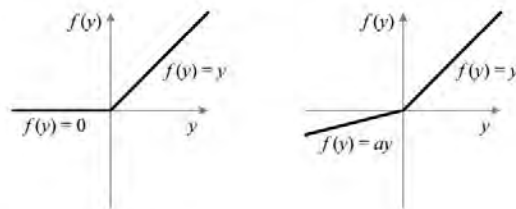
Το κύριο πλεονέκτημα αυτής της συνάρτησης είναι ότι οι ισχυρές αρνητικές είσοδοι θα χαρτογραφηθούν σε αρνητική έξοδο και μόνο οι είσοδοι με μηδενικές τιμές θα χαρτογραφηθούν σε σχεδόν μηδενικές εξόδους. Έτσι, είναι λιγότερο πιθανό να κολλήσει κατά τη διάρκεια της εκπαίδευσης.

#### 4. Διορθωμένες Γραμμικές Μονάδες – (Rectified Linear Units - ReLu)

Η ReLu είναι η πιο χρησιμοποιούμενη συνάρτηση ενεργοποίησης στα CNN και ANN, η οποία κυμαίνεται από μηδενική έως άπειρη.  $[0, \infty)$ .

##### Relu

Δίνει σαν έξοδο το  $x$  αν το  $x$  είναι θετικό, 0 διαφορετικά. Φαίνεται να έχει το ίδιο πρόβλημα των γραμμικών συναρτήσεων καθώς είναι γραμμική στον θετικό άξονα. Η ReLu έχει μη γραμμικό χαρακτήρα και ο συνδυασμός του ReLu είναι επίσης μη γραμμικός. Στην πραγματικότητα, πρόκειται για έναν καλό προσεγγιστή και κάθε λειτουργία μπορεί να προσεγγιστεί με ένα συνδυασμό ReLu. Η τιμή ReLu είναι 6 φορές βελτιωμένη έναντι της συνάρτησης υπερβολικής εφαπτομένης. Θα πρέπει να εφαρμόζεται μόνο σε κρυφά επίπεδα ενός νευρωνικού δικτύου. Έτσι, για το επίπεδο εξόδου χρησιμοποιούμε τη συνάρτηση softmax για το πρόβλημα ταξινόμησης και για το πρόβλημα παλινδρόμησης χρησιμοποιούμε μια γραμμική συνάρτηση. Εδώ ένα πρόβλημα είναι ότι μερικές διαβαθμίσεις είναι εύθραυστες κατά τη διάρκεια της εκπαίδευσης και μπορεί να βγάλει λάθος αποτέλεσμα. Προκαλεί μια ενημέρωση βάρους που θα την κάνει να μην ενεργοποιηθεί ποτέ ξανά σε κανένα σημείο δεδομένων. Βασικά η ReLu θα μπορούσε να οδηγήσει σε νεκρούς νευρώνες. Για να διορθωθεί το πρόβλημα των νεκρών νευρώνων, παρουσιάστηκε η Leaky ReLu. Έτσι, η Leaky ReLu παρουσιάζει μια μικρή κλίση για να κρατήσει ζωντανές τις ενημερώσεις. Η Leaky ReLu κυμαίνεται από  $-\infty$  έως  $+\infty$ .

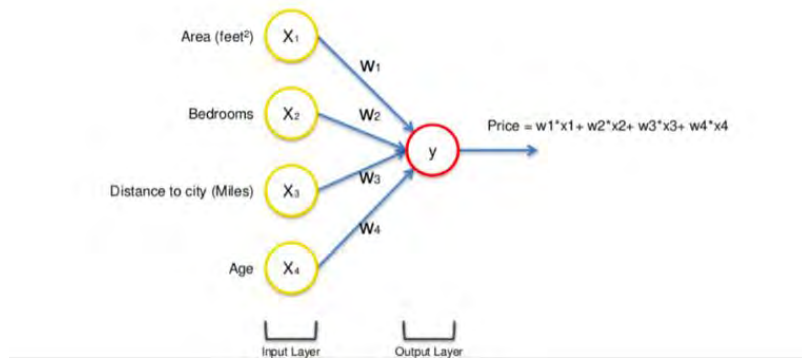


Εικόνα 5: ReLu εναντίον Leaky ReLu

Η διαρροή αυξάνει το εύρος της συνάρτησης ReLu. Συνήθως, η τιμή του  $\alpha$  είναι 0,01 ή περισσότερο. Όταν το  $\alpha$  δεν είναι 0,01, τότε ονομάζεται Randomized ReLu.

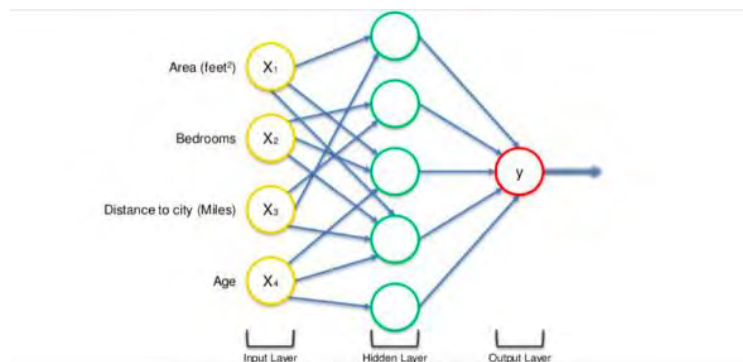
## 4.4 Τρόπος Λειτουργίας Νευρωνικών Δικτύων

Ας πάρουμε το παράδειγμα της τιμής ενός ακινήτου και ας ξεκινήσουμε με τη συγκέντρωση διαφορετικών παραγόντων σε μία σειρά δεδομένων: Περιοχή, υπνοδωμάτια, απόσταση από πόλη και ηλικία.



Εικόνα 6: Είσοδος και έξοδος Νευρωνικού Δικτύου

Οι τιμές εισόδου διέρχονται από τις σταθμισμένες συνάψεις απευθείας στο επίπεδο εξόδου. Θα αναλυθούν και οι τέσσερις, θα εφαρμοστεί μια συνάρτηση ενεργοποίησης και θα παραχθούν τα αποτελέσματα. Αυτό είναι αρκετά απλό, αλλά υπάρχει ένας τρόπος να ενισχύσουμε την ισχύ του Νευρωνικού Δικτύου και να αυξήσουμε την ακρίβειά του με την προσθήκη ενός κρυμμένου επιπέδου που βρίσκεται μεταξύ των επιπέδων εισόδου και εξόδου.



Εικόνα 7: Νευρωνικό Δίκτυο με κρυφό επίπεδο ανάμεσα στο επίπεδο εισόδου και στο επίπεδο εξόδου

Τώρα στο παραπάνω σχήμα, και οι 4 μεταβλητές συνδέονται με νευρώνες μέσω μιας σύναψης. Ωστόσο, δεν σταθμίζονται όλες οι συνάψεις. είτε θα έχουν τιμή 0 είτε μια μη μηδενική τιμή. Στην συγκεκριμένη περίπτωση η μη μηδενική τιμή υποδεικνύει τη σπουδαιότητα. Οι μηδενικές τιμές θα απορριφθούν.

Ας πάρουμε το παράδειγμα της Περιοχής και της Απόστασης προς την Πόλη, οι τιμές των οποίων δεν είναι μηδέν για τον πρώτο νευρώνα, που σημαίνει ότι είναι σταθμισμένοι και μετράνε στον πρώτο νευρώνα. Οι άλλες δύο μεταβλητές, Υποδομάτια και Ηλικία δεν είναι σταθμισμένες και έτσι δεν λαμβάνονται υπόψη από τον πρώτο νευρώνα. Μπορεί να αναρωτιέστε γιατί αυτός ο πρώτος νευρώνας εξετάζει μόνο δύο από τις τέσσερις μεταβλητές. Σε αυτή την περίπτωση, στην αγορά ακινήτων συνηθίζεται τα μεγαλύτερα σπίτια να γίνονται φθηνότερα όσο πιο μακριά είναι από την πόλη. Αυτό είναι ένα βασικό γεγονός. Αυτό που μπορεί να κάνει αυτός ο νευρώνας είναι να ψάχνει συγκεκριμένα για ιδιότητες που είναι μεγάλες αλλά δεν είναι τόσο μακριά από την πόλη. Από εδώ προέρχεται η δύναμη των νευρωνικών δικτύων. Υπάρχουν πολλοί από αυτούς τους νευρώνες, καθένας από τους οποίους κάνει παρόμοιους υπολογισμούς με διαφορετικούς συνδυασμούς αυτών των μεταβλητών. Μόλις ικανοποιηθεί αυτό το κριτήριο, ο νευρώνας εφαρμόζει τη συνάρτηση ενεργοποίησης και εκτελεί τους υπολογισμούς του. Ο επόμενος νευρώνας προς τα κάτω μπορεί να έχει σταθμισμένες συνάψεις της Απόστασης από την πόλη και τα υποδομάτια. Με τον τρόπο αυτό οι νευρώνες λειτουργούν και αλληλοεπιδρούν με πολύ ευέλικτο τρόπο, επιτρέποντάς τους να αναζητήσουν συγκεκριμένα πράγματα και επομένως, να κάνουν μια ολοκληρωμένη αναζήτηση για οτιδήποτε.

## **4.5 Τρόπος εκμάθησης Νευρωνικών Δικτύων(Μετάδοση προς τα πίσω)**

### **4.5.1 Εισαγωγικά Στοιχεία**

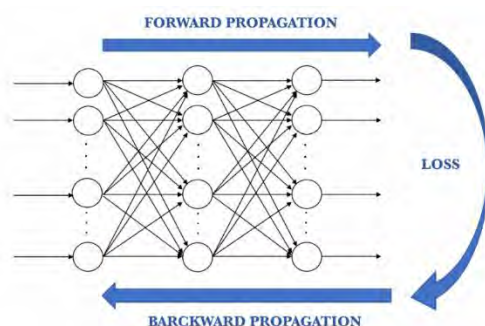
Η εξέταση μιας αναλογίας μπορεί να είναι χρήσιμη για την κατανόηση των μηχανισμών ενός νευρωνικού δικτύου. Η μάθηση σε ένα νευρικό δίκτυο σχετίζεται στενά με το πώς μαθαίνουμε στην κανονική μας ζωή και τις δραστηριότητές μας. Εκτελούμε μια ενέργεια και είτε γίνεται δεκτή είτε διορθωνόμαστε από εκπαιδευτή ή προπονητή για να καταλάβουμε πώς να γίνουμε καλύτεροι σε μια συγκεκριμένη εργασία. Παρομοίως, τα νευρωνικά δίκτυα απαιτούν εκπαιδευτή προκειμένου να περιγράψουν τι θα έπρεπε να έχει παραχθεί ως απάντηση στις εισόδους. Με

βάση τη διαφορά μεταξύ της πραγματικής τιμής και της προβλεπόμενης τιμής, υπολογίζεται και επιστρέφεται μέσω του συστήματος μια τιμή σφάλματος που ονομάζεται Συνάρτηση κόστους.

#### 4.5.2 Συνάρτηση Κόστους

**Συνάρτηση κόστους:** Το μισό της τετραγωνισμένης διαφοράς μεταξύ πραγματικής τιμής και τιμής εξόδου.

Για κάθε επίπεδο του δικτύου, αναλύεται η συνάρτηση κόστους και χρησιμοποιείται για την προσαρμογή του ορίου και των βαρών για την επόμενη είσοδο. Στόχος μας είναι να ελαχιστοποιήσουμε τη συνάρτηση του κόστους. Όσο χαμηλότερη είναι η συνάρτηση κόστους, τόσο πιο κοντά είναι η πραγματική τιμή στην προβλεπόμενη τιμή. Με τον τρόπο αυτό, το σφάλμα μειώνεται οριακά σε κάθε εκτέλεση, καθώς το δίκτυο μαθαίνει πώς να αναλύει τιμές. Τροφοδοτούμε τα δεδομένα που προκύπτουν σε ολόκληρο του νευρωνικό δίκτυο. Οι σταθμισμένες συνάψεις που συνδέουν τις μεταβλητές εισόδου με το νευρώνα είναι το μόνο πράγμα που ελέγχουμε. Εφόσον υπάρχει διαφορά μεταξύ της πραγματικής τιμής και της προβλεπόμενης τιμής, πρέπει να προσαρμόσουμε αυτές τις τιμές. Μόλις τις τροποποιήσουμε λίγο και λειτουργήσουμε το νευρωνικό δίκτυο ξανά, θα παραχθεί μια νέα συνάρτηση κόστους, μικρότερη από την τελευταία. Επαναλαμβάνουμε αυτήν τη διαδικασία μέχρι να ελαχιστοποιήσουμε την συνάρτηση κόστους σε όσο το δυνατόν μικρότερη τιμή.



Εικόνα 8: Μετάδοση προς τα πίσω

Η διαδικασία που περιγράφεται ανωτέρω είναι γνωστή ως Μετάδοση προς τα πίσω και εφαρμόζεται συνεχώς μέσω δικτύου έως ότου η τιμή σφάλματος ελαχιστοποιηθεί.

#### 4.5.3 Προσαρμογή των βαρών

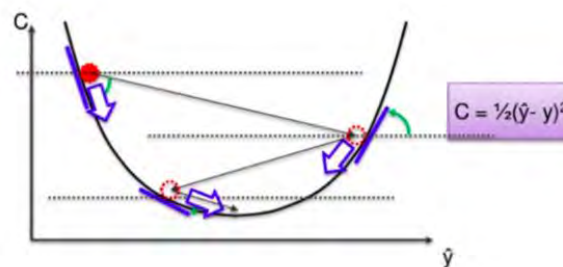
Υπάρχουν βασικά 2 τρόποι για την προσαρμογή των βαρών:

## 1. Ωμή βία

Ιδανική για δίκτυο προώθησης τροφοδοσίας ενός επιπέδου. Εδώ παίρνετε μια σειρά από πιθανά βάρη. Σε αυτή τη μέθοδο, θέλουμε να εξαλείψουμε όλα τα άλλα βάρη εκτός από αυτό που βρίσκεται ακριβώς στο κάτω μέρος της καμπύλης σχήματος U. Το βέλτιστο βάρος μπορεί να βρεθεί χρησιμοποιώντας απλές τεχνικές εξάλειψης. Αυτή η διαδικασία εξάλειψης λειτουργεί εάν έχετε ένα βάρος για βελτιστοποίηση. Σε περίπτωση που έχετε σύνθετο νευρωνικό δίκτυο με πολλούς αριθμούς βαρών, τότε αυτή η μέθοδος αποτυγχάνει. Η εναλλακτική προσέγγιση που έχουμε ονομάζεται Μαζική Κλίση.

### 4.6 Μαζική κλίση

Πρόκειται για έναν πρώτης τάξεως επαναληπτικό αλγόριθμο βελτιστοποίησης και η ευθύνη του είναι να βρει την ελάχιστη τιμή κόστους (ζημία) κατά τη διαδικασία εκπαίδευσης του μοντέλου με διαφορετικά βάρη ή την ενημέρωση των βαρών.



Εικόνα 9: Μαζική Κλίση

#### Κλίση

Στην Κλίση, αντί να επεξεργαζόμαστε ένα βάρος την φορά, και να παίρνουμε κάθε λάθος βάρος, κοιτάμε τη κλίση των συναρτήσεων.

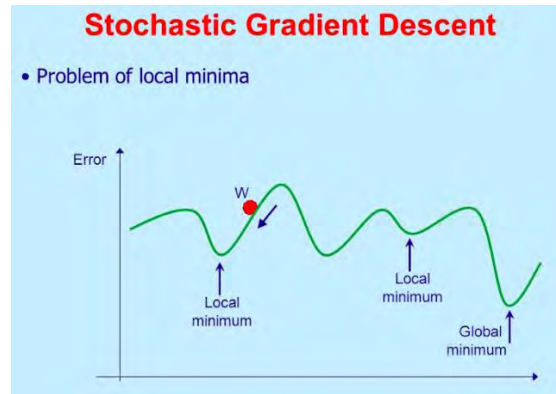
*Αν η κλίση είναι αρνητική, αυτό σημαίνει ότι κατεβαίνουμε την καμπύλη.*

*Αν η κλίση είναι θετική, δεν κάνουμε τίποτα*

Με αυτόν τον τρόπο εξαλείφεται ένας μεγάλος αριθμός λανθασμένων βαρών. Για παράδειγμα, αν έχουμε 3 εκατομμύρια δείγματα, πρέπει να κάνουμε την διαδικασία 3 εκατομμύρια φορές. Οπότε βασικά πρέπει να υπολογισθεί κάθε κόστος 3 εκατομμύρια φορές.

## 4.7 Στοχαστική Κλίση (SGD – Stochastic Gradient Descent)

Η παραπάνω μέθοδος λειτουργεί καλά όταν έχουμε μια κυρτή καμπύλη όπως ακριβώς στο παραπάνω σχήμα. Αλλά αν δεν έχουμε κυρτή καμπύλη, η μέθοδος αποτυγχάνει. Ως "στοχαστικό" νοείται ένα σύστημα ή μια διαδικασία που συνδέεται με τυχαία πιθανότητα. Ως εκ τούτου, στην στοχαστική κλίση, επιλέγονται τυχαία μερικά δείγματα αντί του συνόλου των δεδομένων για κάθε επανάληψη.



Εικόνα 10: Στοχαστική Κλίση

### Στοχαστική κλίση

Στο SGD, παίρνουμε μία σειρά δεδομένων τη φορά, το τρέχουμε μέσω του νευρωνικού δικτύου και μετά προσαρμόζουμε τα βάρη. Για τη δεύτερη επανάληψη, την τρέχουμε και μετά συγκρίνουμε τη συνάρτηση κόστους και στη συνέχεια προσαρμόζουμε ξανά τα βάρη. Και ούτω καθεξής. Το SGD μας βοηθά να αποφύγουμε το πρόβλημα των τοπικών ελαχίστων. Είναι πολύ πιο γρήγορο από την κλίση, επειδή εκτελείται κάθε επανάληψη τη φορά και δεν χρειάζεται να φορτώνει όλα τα δεδομένα στη μνήμη για να εκτελεί υπολογισμούς. Ένα πράγμα που πρέπει να σημειωθεί είναι ότι, καθώς το SGD είναι γενικά πιο ασυνήθιστο από την τυπική κλίση, χρειάζεται συνήθως μεγαλύτερος αριθμός επαναλήψεων για να φτάσει στα ελάχιστα, λόγω της τυχαιότητας της καθόδου του. Αν και απαιτεί μεγαλύτερο αριθμό επαναλήψεων για να φτάσει στα ελάχιστα από την τυπική κλίση. Εξακολουθεί να είναι υπολογιστικά πολύ λιγότερο ακριβή από την τυπική κλίση. Ως εκ τούτου, στα περισσότερα σενάρια, προτιμάται το SGD για τη βελτιστοποίηση ενός αλγορίθμου εκμάθησης.

## 4.8 Εκπαίδευση ενός Νευρωνικού Δικτύου με στοχαστική κλίση

- 1 Αρχικοποιήστε τυχαία τα βάρη σε μικρούς αριθμούς κοντά στο 0 αλλά όχι στο 0.
- 2 Εισαγάγετε την πρώτη παρατήρηση του συνόλου δεδομένων σας στο επίπεδο εισόδου, κάθε χαρακτηριστικό σε έναν κόμβο.
- 3 Μετάδοση προς τα εμπρός: Από τα αριστερά προς τα δεξιά, οι νευρώνες ενεργοποιούνται με τρόπο τέτοιο ώστε η επίδραση της ενεργοποίησης κάθε νευρώνα να περιορίζεται από τα βάρη. Μετάδοση των ενεργοποιήσεων μέχρι τη λήψη της προβλεπόμενης τιμής.
- 4 Συγκρίνετε το προβλεπόμενο αποτέλεσμα με το πραγματικό αποτέλεσμα και μετρήστε το παραγόμενο σφάλμα (συνάρτηση κόστους).
- 5 Μετάδοση πίσω: από τα δεξιά προς τα αριστερά, το σφάλμα μεταδίδεται ξανά. Ενημερώστε τα βάρη ανάλογα με το βαθμό ευθύνης που έχουν για το σφάλμα. Το ποσοστό μάθησης αποφασίζει πόσο θα ενημερώσουμε τα βάρη.
- 6 Επαναλάβετε τα βήματα 1 έως 5 και ενημερώστε τα βάρη μετά από κάθε παρατήρηση (Ενισχυτική μάθηση)
- 7 Όταν όλη η εκπαίδευση περάσει από το Νευρωνικό Δίκτυο, εδώ έχουμε ένα epoch. Επανάληψη της διαδικασίας για περισσότερα epochs. (Chauhan, 2019)

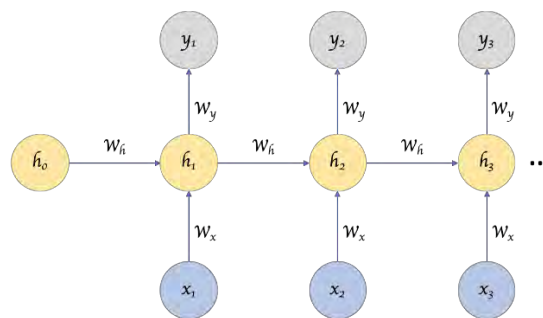
## 4.9 Ανατροφοδοτούμενα Νευρωνικά Δίκτυα (Recurrent Neural Networks)

### 4.9.1 Εισαγωγικά Στοιχεία

Σε αυτή την παράγραφο θα αναφερθούμε σε μια κατηγορία νευρωνικών δικτύων, τα ανατροφοδοτούμενα νευρωνικά δίκτυα.

Ένα απλό νευρωνικό δίκτυο λαμβάνει ως είσοδο ένα σταθερό μέγεθος διανύσματος που περιορίζει τη χρήση του σε καταστάσεις που περιλαμβάνουν είσοδο τύπου "σειράς" χωρίς προκαθορισμένο μέγεθος. Το Ανατροφοδοτούμενο Νευρωνικό Δίκτυο θυμάται το παρελθόν και οι αποφάσεις του επηρεάζονται από όσα έμαθε από το παρελθόν. Σημείωση: Τα βασικά δίκτυα τροφοδοσίας "θυμούνται" πράγματα επίσης, αλλά θυμούνται πράγματα που έμαθαν κατά τη διάρκεια της εκπαίδευσης. Για παράδειγμα, ένας ταξινομητής εικόνων μαθαίνει πώς μοιάζει το "1" κατά τη διάρκεια της εκπαίδευσης και στη συνέχεια χρησιμοποιεί αυτή τη γνώση για να ταξινομήσει τα πράγματα στην παραγωγή. Ενώ τα RNN μαθαίνουν παρόμοια ενώ

εκπαιδεύονται, επιπρόσθετα, θυμούνται πράγματα που μαθαίνουν από προηγούμενες εισόδους, ενώ παράγουν εξόδους. Η διαδικασία αυτή αποτελεί μέρος του δικτύου. Τα RNN μπορούν να πάρουν ένα ή περισσότερα διανύσματα εισόδου και να παράγουν ένα ή περισσότερα διανύσματα εξόδου και οι έξοδοι δεν επηρεάζονται μόνο από τα βάρη που εφαρμόζονται σε εισόδους όπως ένα κανονικό νευρωνικό δίκτυο, αλλά και από ένα "κρυφό" διάνυσμα κατάστασης που αναπαριστά το περιβάλλον με βάση προηγούμενες εισόδους/εξόδους. Έτσι, η ίδια είσοδος θα μπορούσε να παράγει διαφορετική έξοδο ανάλογα με προηγούμενες εισόδους στη σειρά.



Εικόνα 11: Ένα Ανατροφοδοτούμενο Νευρωνικό Δίκτυο, με κρυφή κατάσταση που προορίζεται να μεταφέρει σχετικές πληροφορίες από ένα στοιχείο εισόδου της σειράς σε άλλα.

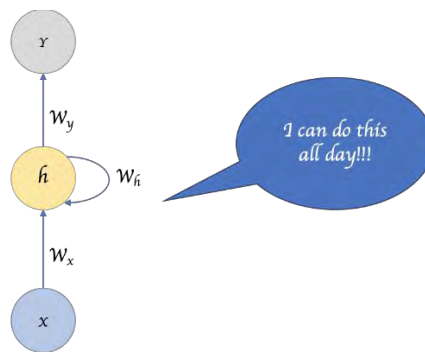
Συνοπτικά, σε ένα βασικό νευρωνικό δίκτυο, ένα διάνυσμα εισόδου σταθερού μεγέθους μετατρέπεται σε ένα διάνυσμα εξόδου σταθερού μεγέθους. Ένα τέτοιο δίκτυο γίνεται "ανατροφοδοτούμενο" όταν εφαρμόζετε επανειλημμένα τους μετασχηματισμούς σε μια σειρά δεδομένων εισόδου και δημιουργείτε μια σειρά διανυσμάτων εξόδου. Δεν υπάρχει προκαθορισμένος περιορισμός στο μέγεθος του διανύσματος. Και, εκτός από το να παράγουμε την έξοδο που είναι συνάρτηση της εισόδου και της κρυφής κατάστασης, ενημερώνουμε την κρυφή κατάσταση με βάση την είσοδο και την χρησιμοποιούμε στην επεξεργασία της επόμενης εισόδου.

#### 4.9.2 Κοινή Χρήση Παραμέτρων

Μπορεί να έχετε παρατηρήσει άλλη σημαντική διαφορά μεταξύ του σχήματος 1 και του σχήματος 3. Στο παρελθόν, εφαρμόζονται πολλαπλά διαφορετικά βάρη στα διάφορα μέρη ενός στοιχείου εισόδου που παράγουν ένα κρυφό νευρώνα επιπέδου, ο οποίος με τη σειρά του μετασχηματίζεται χρησιμοποιώντας επιπλέον βάρη για να παράγει ένα αποτέλεσμα. Στα απλά νευρωνικά δίκτυα φαίνεται να υπάρχουν πολλά βάρη. Ενώ σε ένα RNN (Εικόνα 11), φαίνεται ότι εφαρμόζουμε βάρη ξανά και ξανά σε διαφορετικά στοιχεία της σειράς εισόδου. Σε ένα απλό



νευρωνικό δίκτυο αναφερόμαστε στην "μία" μεμονωμένη είσοδο, ενώ RNN αντιπροσωπεύει πολλαπλές εισόδους από μια σειρά. Ωστόσο, ενστικτωδώς, καθώς αυξάνεται ο αριθμός των εισόδων, δεν θα έπρεπε να αυξηθεί και ο αριθμός των βαρών που διακυβεύονται; Μοιραζόμαστε παραμέτρους σε όλες τις εισόδους στην εικόνα 11. Εάν δεν μοιραζόμαστε παραμέτρους σε όλες τις εισόδους, τότε το δίκτυο μετατρέπεται σε απλό νευρωνικό δίκτυο όπου κάθε κόμβος εισόδου απαιτεί δικά του βάρη. Αυτό εισάγει τον περιορισμό ότι το μήκος των εισόδων πρέπει να καθορίζονται και αυτό καθιστά αδύνατη τη μόχλευση εισόδων τύπου σειράς όταν τα μήκη διαφέρουν και δεν είναι πάντα γνωστά.

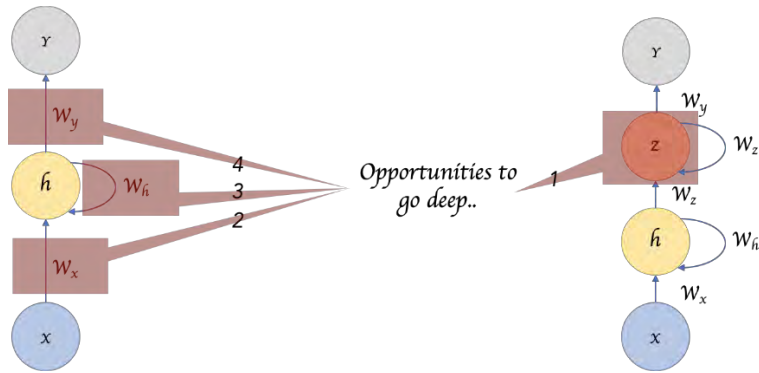


Εικόνα 12: Η κοινή χρήση παραμέτρων βοηθά στην απαλλαγή από τους περιορισμούς μεγέθους

Αλλά αυτό που φαινομενικά χάνουμε σε αξία εδώ, το κερδίζουμε πίσω εισάγοντας την "κρυφή κατάσταση" που συνδέει τη μία είσοδο στην άλλη. Η κρυφή κατάσταση αποτυπώνει τη σχέση που μπορεί να έχουν οι γείτονες μεταξύ τους σε μια σειριακή είσοδο και συνεχίζει να αλλάζει σε κάθε βήμα, και έτσι ουσιαστικά κάθε είσοδος υπόκειται σε διαφορετική μετάβαση.

#### 4.9.3 Βαθιά Ανατροφοδοτούμενα Νευρωνικά Δίκτυα

Αν και είναι καλό ότι η εισαγωγή της κρυφής κατάστασης μας επέτρεψε να εντοπίσουμε αποτελεσματικά τη σχέση μεταξύ των εισόδων, υπάρχει κάποιος τρόπος να φτιάξουμε ένα RNN "βαθύ" και να κερδίσουμε τις πολυεπίπεδες αφαιρέσεις και αναπαραστάσεις που κερδίζουμε μέσω του "βάθους" σε ένα τυπικό νευρωνικό δίκτυο;



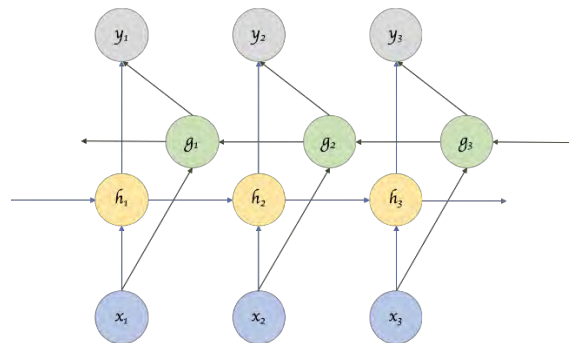
Εικόνα 13: Μπορούμε να αυξήσουμε το βάθος σε τρία πιθανά σημεία σε ένα τυπικό RNN

Ακολουθούν τέσσερις πιθανοί τρόποι για να προσθέσετε βάθος.

- 1 Ίσως το πιο προφανές από όλα, είναι η προσθήκη κρυφών καταστάσεων, η μία πάνω στην άλλη, τροφοδοτώντας την παραγωγή της μίας στην άλλη.
- 2 Μπορούμε επίσης να προσθέσουμε επιπλέον μη γραμμικά κρυφά επίπεδα μεταξύ της εισόδου στην κρυφή κατάσταση
- 3 Μπορούμε να αυξήσουμε το βάθος στην κρυφή μετάβαση
- 4 Μπορούμε να αυξήσουμε το βάθος στη μετάβαση κρυφής εξόδου.

#### 4.9.4 Αμφίδρομα RNN

Μερικές φορές δεν φτάνει η μάθηση από το παρελθόν για να προβλέψουμε το μέλλον, αλλά πρέπει επίσης να κοιτάξουμε στο μέλλον για να διορθώσουμε το παρελθόν. Σε καθήκοντα αναγνώρισης ομιλίας και αναγνώρισης χειρόγραφου, όπου θα μπορούσε να υπάρχει σημαντική ασάφεια με ένα μόνο μέρος της συμβολής, συχνά χρειάζεται να γνωρίζουμε τι ακολουθεί για να κατανοήσουμε καλύτερα το περιεχόμενο και να εντοπίσουμε το παρόν.

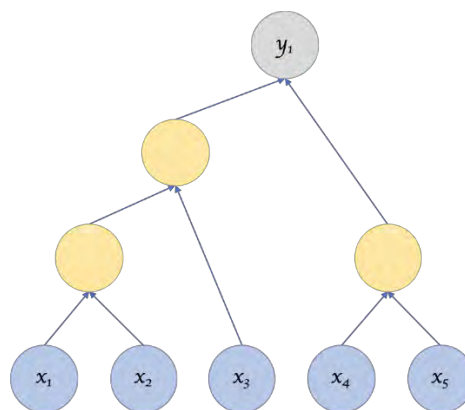


Εικόνα 14: Αμφίδρομα RNN

Αυτό θέτει την προφανή πρόκληση του κατά πόσο πρέπει να εξετάσουμε το μέλλον, διότι αν πρέπει να περιμένουμε να δούμε όλες τις εισόδους, τότε ολόκληρη η διαδικασία θα είναι δαπανηρή. Για τα καθήκοντα NLP, όπου οι εισόδοι τείνουν να είναι διαθέσιμες, είναι πιθανό να μπορούμε να εξετάζουμε ολόκληρες προτάσεις ταυτόχρονα. Επίσης, ανάλογα με την εφαρμογή, εάν η ευαισθησία σε άμεσους και πιο κοντινούς γείτονες είναι μεγαλύτερη από τις εισόδους που έρχονται πιο μακριά, μια παραλλαγή που κοιτάζει μόνο προς ένα περιορισμένο μέλλον/παρελθόν μπορεί να διαμορφωθεί στο μοντέλο.

#### 4.9.5 Αναδρομικά Νευρωνικά Δίκτυα

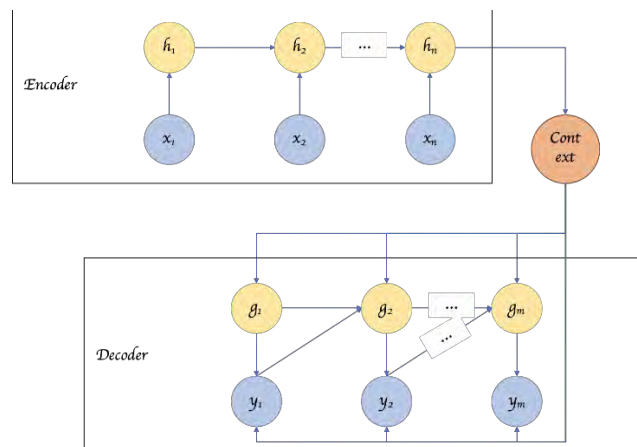
Ένα ανατροφοδοτούμενο νευρωνικό δίκτυο αναλύει τις εισόδους με διαδοχικό τρόπο. Ένα αναδρομικό νευρωνικό δίκτυο είναι παρόμοιο στο ότι οι μεταβάσεις εφαρμόζονται επανειλημμένα στις εισόδους, αλλά όχι απαραίτητα με διαδοχικό τρόπο. Τα Αναδρομικά Νευρωνικά Δίκτυα είναι μια γενικότερη μορφή Ανατροφοδοτούμενων Νευρωνικών Δικτύων. Μπορεί να λειτουργήσει σε οποιαδήποτε ιεραρχική δομή δέντρου. Ανάλυση μέσω κόμβων εισόδου, συνδυασμός θυγατρικών κόμβων στους γονικούς κόμβους και συνδυασμός τους με άλλους θυγατρικούς/γονικούς κόμβους για τη δημιουργία δομής που μοιάζει με δέντρο. Τα Ανατροφοδοτούμενα νευρωνικά δίκτυα κάνουν το ίδιο, αλλά η δομή εκεί είναι αυστηρά γραμμική. δηλαδή, τα βάρη εφαρμόζονται στον πρώτο κόμβο εισόδου, στη συνέχεια στον δεύτερο, στον τρίτο κ.λπ.



Εικόνα 15: Αναδρομικό Νευρωνικό Δίκτυο

#### 4.9.6 Αποκωδικοποιητής/Κωδικοποιητής ακολουθίας σε ακολουθιακά RNN

Ο αποκωδικοποιητής/κωδικοποιητής ή η ακολουθία σε RNN ακολουθίας χρησιμοποιούνται πολύ στις μεταφραστικές υπηρεσίες. Η βασική ιδέα είναι ότι υπάρχουν δύο RNN, ένας κωδικοποιητής που συνεχίζει να ενημερώνει την κρυφή κατάστασή του και παράγει μια τελική έξοδο. Στη συνέχεια, τροφοδοτείται στον αποκωδικοποιητή, ο οποίος μεταφράζει αυτό το πλαίσιο σε μια σειρά εξόδων. Μια άλλη σημαντική διαφορά σε αυτή τη διάταξη είναι ότι το μήκος της ακολουθίας εισόδου και το μήκος της ακολουθίας εξόδου δεν χρειάζεται απαραίτητα να είναι το ίδιο. (Venkatachalam, 2019)



Εικόνα 16: Αποκωδικοποιητής/κωδικοποιητής ή η ακολουθία σε RNN

## Κεφάλαιο 5: TensorFlow

### 5.1 Ορισμός και Εισαγωγικά Στοιχεία

Το TensorFlow είναι μια βιβλιοθήκη σε Python, ανοιχτού κώδικα για αριθμητικές υπολογιστικές εργασίες, που κάνει την μηχανική μάθηση ταχύτερη και ευκολότερη. Η μηχανική μάθηση είναι ένας πολύπλοκος κλάδος. Αλλά η εφαρμογή μοντέλων μηχανικής μάθησης είναι λιγότερο τρομακτική και δύσκολη από ό,τι ήταν μέχρι τώρα, χάρη σε framework μηχανικής μάθησης, όπως είναι το TensorFlow της Google, που διευκολύνουν τη διαδικασία απόκτησης δεδομένων, εκπαίδευσης μοντέλων, εξυπηρέτησης προβλέψεων και βελτίωσης των μελλοντικών αποτελεσμάτων. Το TensorFlow, που δημιουργήθηκε από την ομάδα του Google Brain, είναι μια βιβλιοθήκη ανοιχτού κώδικα για αριθμητικούς υπολογισμούς και μηχανική μάθηση μεγάλης κλίμακας. Το TensorFlow συγκεντρώνει μια σειρά μοντέλων μηχανικής μάθησης, βαθιάς μάθησης και αλγορίθμων και τα καθιστά χρήσιμα. Χρησιμοποιεί Python για να παρέχει ένα πρακτικό front-end API για τη δημιουργία εφαρμογών με framework, ενώ εκτελεί αυτές τις εφαρμογές σε υψηλής απόδοσης C++. Το TensorFlow μπορεί να εκπαιδεύει και να εκτελεί βαθιά νευρωνικά δίκτυα για ταξινόμηση χειρόγραφων ψηφίων, αναγνώριση εικόνας, ενσωμάτωση λέξεων, επαναλαμβανόμενα νευρωνικά δίκτυα, μοντέλα αλληλουχίας προς ακολουθία για μηχανική μετάφραση, επεξεργασία φυσικής γλώσσας και προσομοιώσεις βασισμένες σε μερικές διαφορικές εξισώσεις(PDE - Partial Differential Equation). Το καλύτερο από όλα είναι ότι το TensorFlow υποστηρίζει την πρόβλεψη παραγωγής σε κλίμακα, χρησιμοποιώντας με τα ίδια μοντέλα που χρησιμοποιούνται για την εκπαίδευση.

### 5.2 Λειτουργία TensorFlow

Το TensorFlow επιτρέπει στους προγραμματιστές να δημιουργούν γραφήματα ροής δεδομένων, δηλαδή δομές που περιγράφουν τον τρόπο με τον οποίο τα δεδομένα μετακινούνται μέσα σε ένα γράφημα ή μια σειρά κόμβων επεξεργασίας. Κάθε κόμβος του γραφήματος αντιπροσωπεύει μια μαθηματική λειτουργία και κάθε σύνδεση ή άκρο μεταξύ κόμβων είναι ένας πολυδιάστατος πίνακας δεδομένων ή ένας ελκυστήρας. Το TensorFlow παρέχει όλα αυτά τα εργαλεία στον προγραμματιστή μέσω της γλώσσας Python. Η Python είναι εύκολη στην εκμάθηση και να δουλέψεις μαζί της και παρέχει πρακτικούς τρόπους για να εκφράσεις πώς μπορούν να συνδυαστούν οι αφαιρέσεις υψηλού επιπέδου. Οι κόμβοι και οι δείκτες στο TensorFlow είναι

αντικείμενα της Python και οι εφαρμογές TensorFlow είναι οι ίδιες εφαρμογές σε Python. Οι πραγματικές μαθηματικές λειτουργίες, ωστόσο, δεν εκτελούνται στην Python. Οι βιβλιοθήκες μετασχηματισμών που είναι διαθέσιμες μέσω του TensorFlow είναι γραμμένες ως δυαδικά αρχεία C++ υψηλής απόδοσης. Η Python απλά κατευθύνει την κυκλοφορία μεταξύ των κομματιών. Οι εφαρμογές του TensorFlow μπορούν να εκτελούνται στις περισσότερες κατηγορίες μηχανημάτων: τοπικό μηχάνημα, σύμπλεγμα στο cloud, συσκευές iOS και Android, CPU ή GPU. Εάν χρησιμοποιείτε το ίδιο cloud της Google, μπορείτε να εκτελέσετε το TensorFlow στο προσαρμοσμένο πυρίτιο της Google TensorFlow Processing Unit (TPU) για περαιτέρω επιτάχυνση. Τα μοντέλα που δημιουργούνται από το TensorFlow, ωστόσο, μπορούν να υλοποιηθούν στις περισσότερες συσκευές όπου θα χρησιμοποιηθούν για την εξυπηρέτηση προβλέψεων.

Το TensorFlow 2.0, που δημοσιεύθηκε τον Οκτώβριο του 2019, βελτίωσε το framework με πολλούς τρόπους, βασιζόμενο στην ανάδραση των χρηστών, για να διευκολύνει την εργασία με αυτό(π.χ. χρησιμοποιώντας το σχετικά απλό API Keras για εκπαίδευση μοντέλων) και με καλύτερες επιδόσεις. Η κατανεμημένη εκπαίδευση είναι πιο εύκολη στην εκτέλεση χάρη στο νέο API και η υποστήριξη για το TensorFlow Lite επιτρέπει την υλοποίηση μοντέλων σε μεγαλύτερη ποικιλία πλατφορμών. Ωστόσο, ο κώδικας που έχει συνταχθεί για προηγούμενες εκδόσεις του TensorFlow πρέπει να ξαναγραφτεί για να επωφεληθεί στο μέγιστο τα νέα χαρακτηριστικά TensorFlow 2.0.

## 5.3 Πλεονεκτήματα TensorFlow

Το μοναδικό μεγάλο όφελος που παρέχει το TensorFlow για την ανάπτυξη της μηχανικής μάθησης είναι η αφαίρεση. Αντί να ασχοληθεί με τις λεπτομέρειες της μοναδικότητας της εφαρμογής αλγορίθμων ή να βρει κατάλληλους τρόπους για να σταλθεί η έξοδος μιας λειτουργίας στην είσοδο μιας άλλης, ο προγραμματιστής μπορεί να επικεντρωθεί στη γενική λογική της εφαρμογής. Το TensorFlow φροντίζει για τις λεπτομέρειες στο παρασκήνιο. Το TensorFlow προσφέρει επιπλέον δυνατότητες για προγραμματιστές που πρέπει να εντοπίσουν τα σφάλματα. Η λειτουργία εκτέλεσης προκαθορισμένου προγράμματος σάς επιτρέπει να αξιολογείτε και να τροποποιείτε κάθε λειτουργία γραφήματος ξεχωριστά και με διαφάνεια, αντί να κατασκευάζετε ολόκληρο το γράφημα ως ένα μεμονωμένο αδιαφανές αντικείμενο και να το

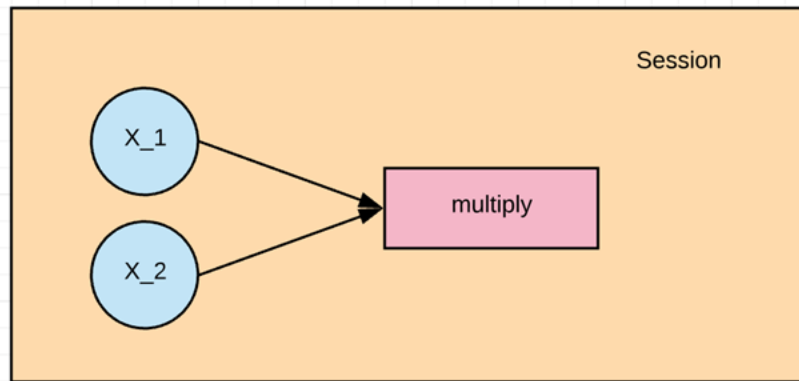
αξιολογείτε όλα ταυτόχρονα. Η οικογένεια απεικόνισης του TensorBoard σάς επιτρέπει να ελέγξετε και να διαμορφώσετε τον τρόπο με τον οποίο τα γραφήματα εκτελούνται μέσω ενός διαδραστικού πίνακα εργαλείων βασισμένου στο διαδίκτυο. Η Google όχι μόνο τροφοδότησε τον ταχύ ρυθμό ανάπτυξης πίσω από το έργο, αλλά δημιούργησε πολλές σημαντικές προσφορές γύρω από το TensorFlow, οι οποίες διευκολύνουν την υλοποίηση και την ευκολότερη χρήση: το προαναφερθέν πυρίτιο TPU για ταχύτερη απόδοση στο cloud της Google, έναν διαδικτυακό κόμβο για την κοινή χρήση μοντέλων που έχουν δημιουργηθεί με το framework. Τέλος, ορισμένες λεπτομέρειες της εφαρμογής του TensorFlow καθιστούν δύσκολη την απόκτηση εντελώς καθοριστικών αποτελεσμάτων εκπαίδευσης σε πρότυπα για ορισμένες θέσεις εκπαίδευσης. Μερικές φορές ένα μοντέλο που έχει εκπαιδευτεί σε ένα σύστημα διαφέρει ελαφρώς από ένα μοντέλο που έχει εκπαιδευτεί σε ένα άλλο, ακόμη και όταν έχουν λάβει τα ίδια ακριβώς δεδομένα. Οι λόγοι γι' αυτό είναι πολλοί, π.χ. ο τρόπος με τον οποίο εμφανίζονται οι τυχαίοι αριθμοί, ή ορισμένες μη προσδιοριστικές συμπεριφορές κατά τη χρήση της GPU). (Yegulalp, 2019)

## 5.4 Παράδειγμα TensorFlow

Σε αυτή την παράγραφο θα δούμε ένα πολύ απλό παράδειγμα TensorFlow.

```
import numpy as np
import tensorflow as tf
```

Στην πρώτη γραμμή του κώδικα, εισαγάγαμε το TensorFlow ως tf. Με την Python, είναι κοινή πρακτική να χρησιμοποιείται ένα σύντομο όνομα για μια βιβλιοθήκη. Το πλεονέκτημα είναι να αποφεύγετε να πληκτρολογείτε το πλήρες όνομα της βιβλιοθήκης όταν πρέπει να τη χρησιμοποιήσουμε. Για παράδειγμα, μπορούμε να εισαγάγουμε τη ροή του TensorFlow ως tf, και να καλέσουμε tf όταν θέλουμε να χρησιμοποιήσουμε μια συνάρτηση TensorFlow. Ας εξασκήσουμε την στοιχειώδη ροή εργασιών του TensorFlow με ένα απλό παράδειγμα. Ας δημιουργήσουμε ένα υπολογιστικό γράφημα που πολλαπλασιάζει δύο αριθμούς μαζί. Στη διάρκεια του παραδείγματος, θα πολλαπλασιάσουμε τα X\_1 και X\_2 μαζί. Το TensorFlow θα δημιουργήσει έναν κόμβο για τη σύνδεση της λειτουργίας. Στο παράδειγμά μας, λέγεται πολλαπλασιασμός. Όταν προσδιοριστεί το γράφημα, οι υπολογιστικοί μηχανισμοί του TensorFlow θα πολλαπλασιάσουν τα X\_1 και X\_2.



Εικόνα 17: Παράδειγμα TensorFlow, Πολλαπλασιασμός Αριθμών

Τέλος, θα εκτελέσουμε ένα TensorFlow session το οποίο θα εκτελέσει το υπολογιστικό γράφημα με τις τιμές X\_1 και X\_2 και θα εκτυπώσει το αποτέλεσμα του πολλαπλασιασμού. Ας ορίσουμε τους κόμβους εισόδου X\_1 και X\_2. Όταν δημιουργούμε έναν κόμβο στο TensorFlow, πρέπει να επιλέγουμε το είδος του κόμβου που θα δημιουργήσουμε. Οι κόμβοι X1 και X2 θα είναι ένας κόμβος placeholder. Το placeholder εκχωρεί μια νέα τιμή κάθε φορά που κάνουμε έναν υπολογισμό.

### Βήμα 1: Ορισμός Μεταβλητών:

```
X_1 = tf.placeholder(tf.float32, name = "X_1")  
X_2 = tf.placeholder(tf.float32, name = "X_2")
```

Όταν δημιουργούμε ένα placeholder, πρέπει να περάσουμε τον τύπο δεδομένων που θα προσθέτει αριθμούς εδώ, ώστε να μπορούμε να χρησιμοποιήσουμε έναν τύπο δεδομένων κινητής υποδιαστολής, ας χρησιμοποιήσουμε το tf.float32. Πρέπει επίσης να δώσουμε σε αυτόν τον κόμβο ένα όνομα. Αυτό το όνομα θα εμφανιστεί όταν δούμε τις γραφικές απεικονίσεις του μοντέλου μας. Ας ονομάσουμε αυτόν τον κόμβο X\_1 περνώντας μια παράμετρο που ονομάζεται X\_1 και τώρα ας ορίσουμε την X\_2 με τον ίδιο τρόπο.

### Βήμα 2: Ορισμός του Υπολογισμού

```
multiply = tf.multiply(X_1, X_2, name = "multiply")
```



Τώρα μπορούμε να ορίσουμε τον κόμβο που εκτελεί τη λειτουργία πολλαπλασιασμού. Στο TensorFlow μπορούμε να το κάνουμε αυτό δημιουργώντας έναν κόμβο `tf.multiply`. Θα μεταβιβάσουμε τους κόμβους `X_1` και `X_2` στον κόμβο πολλαπλασιασμού. Δίνει εντολή στο TensorFlow να συνδέσει αυτούς τους κόμβους στο υπολογιστικό γράφημα, έτσι του ζητάμε να τραβήξει τις τιμές από το `x` και το `y` και να πολλαπλασιάσει το αποτέλεσμα. Επίσης θα δώσουμε στον κόμβο πολλαπλασιασμού ένα όνομα(`multiply`).

### Βήμα 3: Εκτέλεση Λειτουργίας

Για την εκτέλεση λειτουργιών στο γράφημα, πρέπει να δημιουργήσουμε μια περίοδο λειτουργίας. Στο TensorFlow, αυτό γίνεται από το `tf.Session()`. Τώρα που έχουμε μια περίοδο λειτουργίας μπορούμε να ζητήσουμε από τη μια περίοδο λειτουργίας να εκτελέσει λειτουργίες στο υπολογιστικό μας γράφημα καλώντας την μια περίοδο λειτουργίας. Για να εκτελέσουμε τον υπολογισμό, πρέπει να χρησιμοποιήσουμε την εκτέλεση. Όταν εκτελείται η λειτουργία προσθήκης, θα δει ότι το πρόγραμμα θα πάρει τις τιμές των κόμβων `X_1` και `X_2`, επομένως πρέπει επίσης να τροφοδοτήσουμε τιμές για τις `X_1` και `X_2`. Μπορούμε να το κάνουμε αυτό παρέχοντας μια παράμετρο που ονομάζεται `feed_dictionary`. Περνάμε την τιμή 1,2,3 στο `X_1` και 4,5,6 στο `X_2`. Εκτυπώνουμε τα αποτελέσματα με την εντολή `print(result)`. Θα πρέπει να δείτε 4, 10 και 18 για `1x4`, `2x5` και `3x6`.

## 5.5 Επιλογές για Φόρτωση δεδομένων στο TensorFlow

Το πρώτο βήμα πριν από την εκπαίδευση ενός αλγόριθμου μηχανικής μάθησης είναι η φόρτωση των δεδομένων. Υπάρχουν δύο κοινά στοιχεία για τη φόρτωση δεδομένων:

1. Φόρτωση δεδομένων στη μνήμη: Είναι η απλούστερη μέθοδος. Φορτώνετε όλα τα δεδομένα σας στη μνήμη με την χρήση πίνακα. Μπορείτε να γράψετε κώδικα Python. Αυτές οι γραμμές κώδικα δεν σχετίζονται με το TensorFlow.
2. Διοχέτευση δεδομένων TensorFlow. Το TensorFlow διαθέτει ενσωματωμένο API που σας βοηθάει να φορτώσετε τα δεδομένα, να εκτελέσετε τη λειτουργία και να τροφοδοτήσετε εύκολα τον αλγόριθμο μηχανικής μάθησης. Αυτή η μέθοδος λειτουργεί πολύ καλά, ειδικά όταν έχετε ένα μεγάλο όγκο δεδομένων. Για παράδειγμα, οι εγγραφές εικόνων είναι γνωστό ότι είναι τεράστιες και δεν χωράνε στη μνήμη. Η διοχέτευση δεδομένων διαχειρίζεται τη μνήμη από μόνη της.

Ποια Λύση χρησιμοποιούμε ανάλογα με την περίπτωση:

### **Φόρτωση δεδομένων στη μνήμη**

Εάν το σύνολο δεδομένων σας δεν είναι πολύ μεγάλο, δηλαδή μικρότερο από 10 gigabyte, μπορείτε να χρησιμοποιήσετε την πρώτη μέθοδο. Τα δεδομένα μπορούν να χωρέσουν στη μνήμη. Μπορείτε να χρησιμοποιήσετε μια διάσημη βιβλιοθήκη που ονομάζεται Pandas για να εισαγάγετε αρχεία CSV.

### **Φόρτωση δεδομένων με διοχέτευση TensorFlow**

Η δεύτερη μέθοδος λειτουργεί καλύτερα εάν έχετε ένα μεγάλο σύνολο δεδομένων. Για παράδειγμα, εάν έχετε ένα σύνολο δεδομένων 50 gigabyte και ο υπολογιστής σας διαθέτει μόνο 16 gigabyte μνήμης, τότε ο υπολογιστής θα καταρρεύσει. Σε αυτή την περίπτωση, πρέπει να κατασκευάσετε μια διοχέτευση TensorFlow. Η διοχέτευση θα φορτώσει τα δεδομένα μαζικά ή σε μικρά μπλοκ. Κάθε παρτίδα θα ωθείται στην διοχέτευση και θα είναι έτοιμη για την εκπαίδευση. Η κατασκευή μιας διοχέτευσης είναι μια εξαιρετική λύση, επειδή σας επιτρέπει να χρησιμοποιείτε παράλληλες υπολογιστικές εργασίες. Αυτό σημαίνει ότι το TensorFlow θα εκπαιδεύσει το μοντέλο σε πολλαπλούς επεξεργαστές. Ενισχύει τη χρήση υπολογιστών για την εκπαίδευση ισχυρού νευρωνικού δικτύου.

Με λίγα λόγια, εάν έχετε ένα μικρό σύνολο δεδομένων, μπορείτε να φορτώσετε τα δεδομένα στη μνήμη με τη βιβλιοθήκη Pandas. Εάν έχετε ένα μεγάλο σύνολο δεδομένων και θέλετε να χρησιμοποιήσετε πολλαπλούς επεξεργαστές, τότε θα είστε πιο άνετοι να εργαστείτε με τη διοχέτευση TensorFlow.

## **5.6 Δημιουργία Διοχέτευση TensorFlow**

Στο προηγούμενο παράδειγμα, προσθέτουμε με μη αυτόματο τρόπο τρεις τιμές για τις X\_1 και X\_2. Τώρα θα δούμε πώς θα φορτώσουμε τα δεδομένα στο TensorFlow.

### **Βήμα 1: Δημιουργία Δεδομένων**

Πρώτα απ' όλα, ας χρησιμοποιήσουμε τη βιβλιοθήκη numpy για να δημιουργήσουμε δύο τυχαίες τιμές.

```
import numpy as np
```

```
x_input = np.random.sample((1,2))  
print(x_input)
```

```
[[0.8835775 0.23766977]]
```

## **Βήμα 2: Δημιουργία placeholder**

Όπως και στο προηγούμενο παράδειγμα, δημιουργούμε ένα placeholder με το όνομα `x`. Σε περίπτωση που φορτώσουμε έναν πίνακα με μόνο δύο τιμές, μπορούμε να γράψουμε το σχήμα ως `shape=[1,2]`.

## **Βήμα 3: Ορισμός της μεθόδου για το σύνολο των δεδομένων**

Στη συνέχεια, πρέπει να ορίσουμε το σύνολο δεδομένων όπου μπορούμε να συμπληρώσουμε την τιμή του placeholder `x`. Πρέπει να χρησιμοποιήσουμε τη μέθοδο `tf.data.Dataset.from_tensor_slices`.

```
dataset = tf.data.Dataset.from_tensor_slices(x)
```

## **Βήμα 4: Δημιουργία Διοχέτευσης**

Στο τέταρτο βήμα, πρέπει να αρχικοποιήσουμε τη διοχέτευση όπου θα ρέουν τα δεδομένα. Πρέπει να δημιουργήσουμε ένα πρόγραμμα επανάληψης με `make_initiable_iterator`. Τον ονομάσαμε "Επαναλήπτη". Στη συνέχεια, πρέπει να καλέσουμε αυτόν τον επαναλήπτη για να τροφοδοτήσει την επόμενη ροή δεδομένων, `get_next`. Ονομάστε αυτό το βήμα `get_next`. Σημειώστε ότι στο παράδειγμά μας, υπάρχει μόνο μία ροή δεδομένων με μόνο δύο τιμές.

```
iterator = dataset.make_initializable_iterator()  
get_next = iterator.get_next()
```

## **Βήμα 5: Εκτέλεση της Λειτουργίας**

Το τελευταίο βήμα είναι παρόμοιο με το προηγούμενο παράδειγμα. Αρχίζουμε μια περίοδο λειτουργίας, και εκτελούμε τον επαναλήπτη. Τροφοδοτούμε το `feed_dictionary` με την τιμή που δημιουργείται από το `numpy`. Αυτές οι δύο τιμές θα συμπληρώσουν το placeholder `x`. Στη συνέχεια, θα εκτελεστεί το `get_next` για να εκτυπωθεί το αποτέλεσμα.

```
with tf.Session() as sess:
```

```
# feed the placeholder with data
sess.run(iterator.initializer, feed_dict={ x: x_input })
print(sess.run(get_next)) # output [ 0.52374458  0.71968478]
```

```
[0.8835775  0.23766978]
```

## 5.7 Σύντομη Ιστορία του TensorFlow

Πριν μερικά χρόνια, η βαθιά μάθηση άρχισε να ξεπερνά όλους τους άλλους αλγόριθμους εκμάθησης μηχανής επειδή έδινε μια τεράστια ποσότητα δεδομένων. Η Google είδε ότι μπορούσε να χρησιμοποιήσει αυτά τα βαθιά νευρωνικά δίκτυα για να βελτιώσει τις υπηρεσίες της:

- Gmail
- Φωτογραφία
- μηχανή αναζήτησης Google

Έτσι έχτισαν ένα πλαίσιο που ονομάζεται TensorFlow για να επιτρέπουν στους ερευνητές και τους προγραμματιστές να συνεργάζονται σε ένα μοντέλο τεχνητής νοημοσύνης. Μόλις αναπτύχθηκε και κλιμακώθηκε, επέτρεψε σε πολλούς ανθρώπους να το χρησιμοποιήσουν. Δημοσιεύτηκε για πρώτη φορά στα τέλη του 2015, ενώ η πρώτη σταθερή έκδοση εμφανίστηκε το 2017. Είναι ανοικτού κώδικα με άδεια Apache Open Source. Μπορείτε να το χρησιμοποιήσετε, να το τροποποιήσετε και να αναδιανεμίτε την τροποποιημένη έκδοση έναντι αντιτίμου χωρίς να πληρώσετε τίποτα στο Google.

## 5.8 Αρχιτεκτονική TensorFlow

Η αρχιτεκτονική του TensorFlow λειτουργεί σε τρία μέρη:

1. Προεπεξεργασία των δεδομένων
2. Κατασκευή του μοντέλου
3. Εκπαίδευση και εκτίμηση του μοντέλου

Ονομάζεται TensorFlow επειδή δέχεται την είσοδο ως έναν πολυδιάστατο πίνακα, γνωστή και ως tensor. Μπορείτε να δημιουργήσετε ένα είδος διαγράμματος ροής λειτουργιών (που ονομάζεται Graph) τον οποίο θέλετε να εκτελέσετε σε αυτήν την είσοδο. Η είσοδος γίνεται από τη μία άκρη, και μετά ρέει μέσα από αυτό το σύστημα πολλαπλών λειτουργιών και βγαίνει από την άλλη άκρη ως έξοδος. (What is TensorFlow? Introduction, Architecture & Example, n.d.)

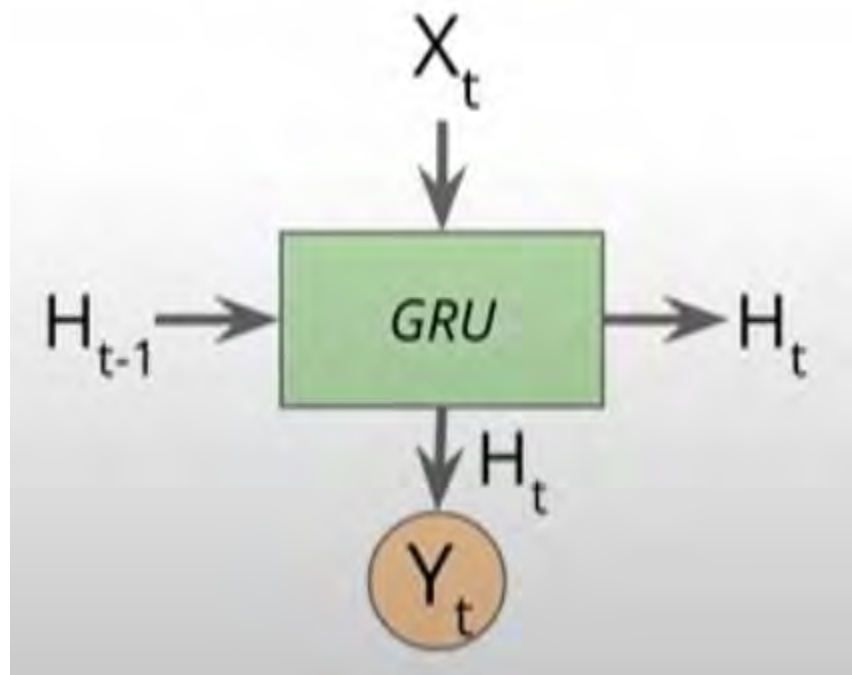
## Κεφάλαιο 6: Χρήση TensorFlow για παραγωγή Κειμένων του Ομήρου

### 6.1 TensorFlow για παραγωγή Φυσικής Γλώσσας

Για την παραγωγή Φυσικής γλώσσας χρησιμοποιούμε το TensorFlow. Σε αυτή την παράγραφο θα δούμε το μοντέλο που χρησιμοποιούμε για την παραγωγή φυσικής γλώσσας και στην συνέχεια θα δούμε τις εντολές με τις οποίες πετυχαίνουμε τον στόχο μας. Για την παραγωγή Φυσικής γλώσσας θα χρησιμοποιήσουμε Ανατροφοδοτούμενα Νευρωνικά Δίκτυα.

#### 6.1.1 Το μοντέλο για Παραγωγή Φυσικής Γλώσσας

Για την παραγωγή φυσικής γλώσσας, θα εισάγουμε χαρακτήρες σε κάθε κελί και αυτό σαν έξοδο θα «μαντεύει» τον επόμενο χαρακτήρα που θα μπει στο κείμενο που θέλουμε να παράγουμε. Στην συγκεκριμένη φάση θα χρησιμοποιήσουμε το λεγόμενο GRU (Gated recurrent Unit). Την δομή που βλέπουμε στην διπλανή εικόνα την μελετήσαμε στο κεφάλαιο 5, όταν αναφερθήκαμε στα ανατροφοδοτούμενα νευρωνικά δίκτυα. Ας δούμε συνοπτικά τι περιλαμβάνει:



Εικόνα 18: Δομή GRU

- Μια είσοδο  $X_t$  (Γράμμα)
- Μια έξοδο  $Y_t$  (Το επόμενο γράμμα του κειμένου μας)
- Μια κατάσταση που υπολογίστηκε στο προηγούμενο βήμα ( $H_{t-1}$ )

- Την νέα κατάσταση που υπολογίζει το τρέχων βήμα, η οποία τροφοδοτείται στο επόμενο βήμα και χρησιμοποιείται για τον υπολογισμό της εξόδου του τρέχοντος βήματος.

Γενικά το GRU υπολογίζει την έξοδο χρησιμοποιώντας τις παρακάτω εξισώσεις:

$X = X_t | H_{t-1}$  : Στην ουσία η είσοδος αποτελείται από την είσοδο που δίνεται συνενωμένη με την κατάσταση του προηγούμενου βήματος.

$$z = \sigma(X \cdot W_z + b_z)$$

$$r = \sigma(X \cdot W_r + b_r)$$

Οι παραπάνω δύο εξισώσεις αποτελούν επίπεδο νευρωνικού δικτύου όπως είδαμε στα προηγούμενα κεφάλαια. Χρησιμοποιεί τα βάρη και της πολώσεις κάθε επιπέδου και επιπλέον χρησιμοποιεί την σιγμοειδή συνάρτηση ενεργοποίησης.

$$X' = X_t | r * H_{t-1}$$

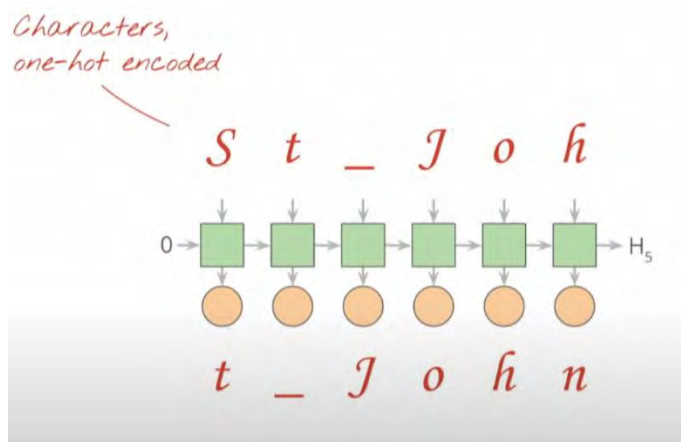
$$X'' = \tanh(X' \cdot W_c + b_c)$$

$$H_t = (1 - z) * H_{t-1} + z * X''$$

$$Y_t = \text{softmax}(H_t \cdot W + b)$$

Ας δούμε σχηματικά πως γίνεται η παραγωγή φυσικής γλώσσας:

όπως βλέπουμε στο διπλανό σχήμα αυτό που κάνουμε είναι να τροφοδοτούμε με χαρακτήρα το μοντέλο που περιγράψαμε παραπάνω και αυτό «μαντεύει» τον επόμενο χαρακτήρα. Η μαντεψιά γίνεται χρησιμοποιώντας τις εξισώσεις που παραθέσαμε παραπάνω. Αυτό που πρέπει να παρατηρήσουμε είναι ότι η αρχική κατάσταση  $H_{-1}$  είναι 0 και στην συνέχεια



Εικόνα 19: : Παραγωγή Φυσικής Γλώσσας με την Χρήση του GRU

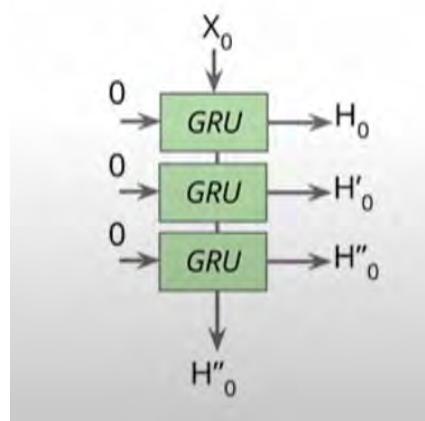
η κατάσταση αυτή υπολογίζεται σε κάθε βήμα και την χρησιμοποιεί το επόμενο βήμα σαν είσοδο μαζί με το γράμμα.

### 6.1.2 Χρήση του GRU στο TensorFlow.

Για να χρησιμοποιήσουμε το GRU στο TensorFlow αρχικά εισάγουμε την εντολή:

```
cell = (tf.nn.rnn.cell.GRUCell(CELLSIZE)).
```

Με την παραπάνω εντολή αρχικοποιούμε ένα κελί GRU με ένα δεδομένο μέγεθος κελιού. Η συγκεκριμένη εντολή ορίζει εσωτερικά τα βάρη και τις πολώσεις που θα χρησιμοποιήσει. Εμείς όμως θέλουμε να χρησιμοποιήσουμε βαθιά ανατροφοδοτούμενα νευρωνικά δίκτυα. Για να γίνει αυτό αρκεί να «στοιβάξουμε» 3 τέτοια κελιά το ένα πάνω στο άλλο. Με βάση αυτό η είσοδος του επόμενου θα είναι η έξοδος του προηγούμενου κελιού. Στην διπλανή εικόνα βλέπουμε ένα παράδειγμα όπου για το βήμα χρησιμοποιούμε 3 στρώματα. Για να γίνει αυτό χρησιμοποιούμε την παρακάτω εντολή στο TensorFlow:



Εικόνα 20: Το πρώτο βήμα Υπολογισμού της Φυσικής Γλώσσας Χρησιμοποιώντας 3 στρώματα

```
mcell = tf.nn.rnn_cell.MultiRNNCell([cell] * NLAYERS, state_is_tuple = False)
```

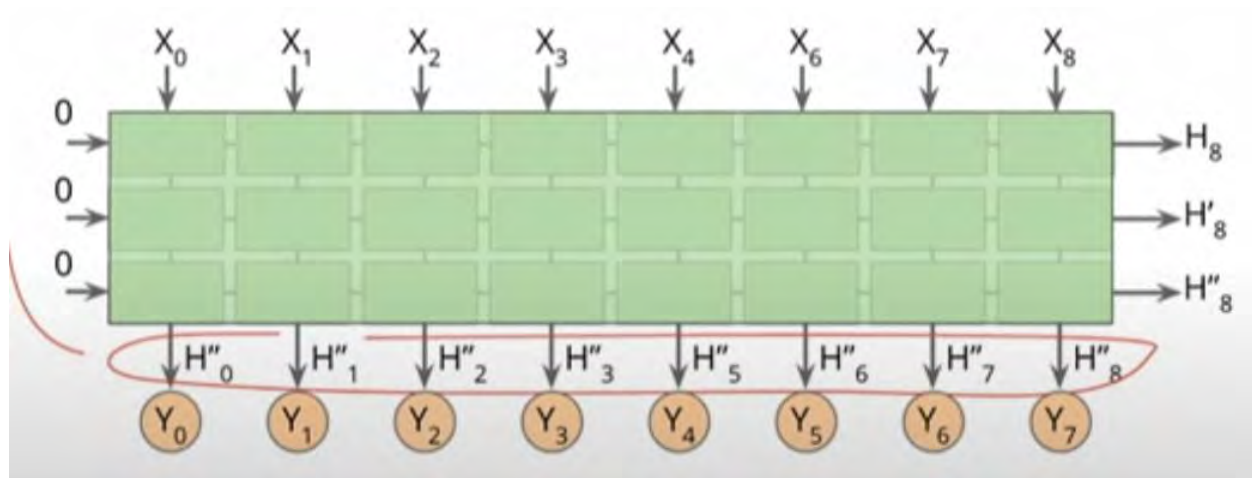
Η μεταβλητή που μας ενδιαφέρει στην παραπάνω εντολή είναι η NLAYERS η οποία ορίζει σε πόσα στρώματα θα «στοιβαχθεί» κάθε βήμα. Στο παράδειγμα που βλέπουμε στην Εικόνα 20 NLAYERS = 3. Το αποτέλεσμα αυτού του «στοιβάγματος» θα είναι ένα νέο κελί. Τέλος με την παρακάτω εντολή ξετυλίγουμε το μοντέλο για όσους χαρακτήρες περιέχει το κείμενο, με το οποίο θέλουμε να εκπαιδεύσουμε το μοντέλο (το μοντέλο εκπαιδεύεται με ομάδες από ακολουθίες δεδομένων (κάθε δέσμη έχει ίδιο αριθμό χαρακτήρων). Αυτό σημαίνει ότι στην εκπαίδευση δεν τροφοδοτούμε το δίκτυο με όλο το κείμενο εξαρχής):

```
Hr, H = tf.nn.dynamic_rnn(mcell, X, initial_state = Hin)
```

Για την εφαρμογή του στρώματος της softmax (για τον υπολογισμό της τελικής εξόδου). Εδώ πρέπει να θυμηθούμε ότι όλα τα βήματα μοιράζονται τα βάρη και τις πολώσεις. Αυτό που μπορούμε να κάνουμε είναι να πάρουμε όλες τις εξόδους από μια δέσμη δεδομένων και να τις



τροφοδοτήσουμε σε ένα στρώμα softmax. Στην συνέχεια αφού έχουν γίνει οι κατάλληλοι υπολογισμοί θα γίνει μετατροπή των εξόδων στην κατάλληλη μορφή (αυτή που είχαμε πριν). Αυτό φαίνεται σχηματικά στην παρακάτω εικόνα:



Εικόνα 21: Υπολογισμός των Εξόδων (softmax)

Στο TensorFlow αυτό γίνεται χρησιμοποιώντας τις παρακάτω 3 εντολές:

$$H_f = \text{tf.reshape}(H_r, [-1, \text{CELLSIZE}]) \quad (1)$$

$$Y_{\text{logits}} = \text{layers.linear}(H_f, \text{ALPHASIZE}) \quad (2)$$

$Y = \text{tf.nn.softmax}(Y_{\text{logits}}) \quad (3)$  Η εντολή (2) υπολογίζει τα σταθμισμένα αθροίσματα και η (3) υπολογίζει την συνάρτηση ενεργοποίησης (softmax).

Επίσης, με την παρακάτω εντολή υπολογίζουμε την συνάρτηση απώλειας:

$$\text{loss} = \text{tf.nn.softmax\_cross\_entropy\_with\_logits}(Y_{\text{logits}}, Y\_)$$

Ας δούμε τώρα τον τρόπο με τον οποίο η TensorFlow μας επιτρέπει να ορίζουμε την είσοδο του προγράμματός μας (το κείμενο). Αυτό που θέλουμε να κάνουμε είναι να δώσουμε σαν είσοδο δεσμίδες από ακολουθίες χαρακτήρων σταθερού μεγέθους. Αυτό που θέλουμε να κάνουμε στους χαρακτήρες είναι το λεγόμενο One Hot Encoding. Αυτό σημαίνει ότι κάθε χαρακτήρας θα μετατραπεί σε ένα διάνυσμα μεγέθους  $n$  (όπου  $n$  το μέγεθος του αλφάβητου). Η θέση κάθε

στοιχείου του διανύσματος θα αντιστοιχεί σε κάθε γράμμα του αλφάβητου και το περιεχόμενο του στοιχείου θα είναι η πιθανότητα να συναντήσουμε το συγκεκριμένο γράμμα (0 έως 1).

Όσον αφορά τις καταστάσεις, η κάθε μια από τις οποίες είναι ένα διάνυσμα, το μέγεθος του οποίου θα είναι ίσο με το μέγεθος του κελιού (GRU) \* τον αριθμό των στρωμάτων (CELLSIZE \* NLAYERS). Ας δούμε πως θα είναι το μοντέλο μας:

$$Xd = tf.placeholder(tf.uint8, [None, None])$$

Με την παραπάνω εντολή ορίζουμε ένα placeholder για την δέσμη ακολουθιών χαρακτήρων.

$$X = tf.one\_hot(X, ALPHASIZE, 1.0, 0.0)$$

Με αυτή την εντολή κάθε χαρακτήρας μετατρέπεται σε διάνυσμα μεγέθους ALPHASIZE (το μέγεθος του αλφάβητου που χρησιμοποιούμε).

$$Yd\_ = tf.placeholder(tf.uint8, [None, None])$$

Με αυτή την εντολή δημιουργείται ένα placeholder, στο οποίο θα αποθηκευτούν οι σωστές απαντήσεις. Οι σωστές απαντήσεις θα είναι δέσμες ακολουθιών χαρακτήρων (κλασσικά κάθε ακολουθία αποτελείται από συγκεκριμένο πλήθος χαρακτήρων το οποίο είναι ίδιο με την είσοδο). Στην συνέχεια κάνουμε one hot encode και για την έξοδο με την παρακάτω εντολή:

$$Y\_ = tf.one\_hot(Y\_ , ALPHASIZE, 1.0, 0.0)$$

Τέλος, χρειαζόμαστε ένα placeholder για την αρχική κατάσταση. Όπως είπαμε και παραπάνω το μέγεθος της αρχικής κατάστασης θα είναι CELLSIZE \* NLAYERS. Οπότε αυτό το πετυχαίνουμε με την παρακάτω εντολή:

$$Hin = tf.placeholder(tf.float32, [None, CELLSIZE * NLAYERS])$$

Για να βρούμε ποιος χαρακτήρας θα επιλεγεί με βάση της πιθανότητες χρησιμοποιούμε την παρακάτω εντολή:

$$predictions = tf.argmax(Y, 1)$$

Η συνάρτηση `argmax` δίνει την θέση με το με το μεγαλύτερο αριθμό στο διάνυσμα. Στην συγκεκριμένη περίπτωση, ο αριθμός είναι μια πιθανότητα και η θέση είναι ο κωδικός ASCII ενός χαρακτήρα.

```
predictions = tf.reshape(predictions, [batchsize - 1])
```

Με την παραπάνω εντολή γίνεται πάλι η μετατροπή σε δέσμες χαρακτήρων(είδαμε πιο πριν ότι για τον υπολογισμό της softmax συνάρτησης ενεργοποίησης το είχαμε αναιρέσει αυτό).

Τέλος, με την παρακάτω εντολή λέμε στο TensorFlow να βελτιστοποιήσει το δίκτυο για να γίνει η εκπαίδευση:

```
train_step = tf_train.AdamOptimizer(1e - 3).minimize(loss).
```

## 6.2 Χρήση των Παραπάνω για Παραγωγή Κειμένων του Ομήρου

Μέχρι τώρα είδαμε πως μπορούμε γενικά να κάνουμε παραγωγή φυσικής γλώσσας. Σε αυτή την παράγραφο θα δούμε πως κάναμε παραγωγή κειμένων Του Ομήρου δίνοντας σαν είσοδο κείμενα στα νέα Ελληνικά. Για να γίνει όμως αυτό έπρεπε πρώτα να κάνουμε μια Προεπεξεργασία στα κείμενα που βρήκαμε στο διαδίκτυο.

### 6.2.1 Προεπεξεργασία Κειμένου

Το πρώτο βήμα της προεπεξεργασίας είναι η διαγραφή των γραμμών που περιείχαν Αρχαία Ελληνικά. Αυτό έγινε διότι σε πρώτη φάση η εκπαίδευση του δικτύου και κατά συνέπεια το κείμενο που θα παράγεται, θα είναι στα Νέα Ελληνικά. Στην συνέχεια στο κείμενο που παράχθηκε από την παραπάνω διαδικασία, αφαιρέσαμε κάποιους χαρακτήρες, οι οποίοι ήταν στα αγγλικά ενώ έπρεπε να είναι στα Ελληνικά. Στην συνέχεια για το αλφάβητο του κειμένου μας κρατήσαμε το 99.9% των χαρακτήρων. Αυτό σημαίνει, ότι το αλφάβητο των 85 χαρακτήρων μειώθηκε σε ένα λεξικό των 63 χαρακτήρων.

### 6.2.2 Εκπαίδευση του Μοντέλου

Το μοντέλο μας το εκπαιδεύσαμε σε 9 συστήματα (9 διαφορετικούς συνδυασμούς παραμέτρων). Οι παράμετροι που αλλάζαμε είναι: το μέγεθος του κελιού και ο αριθμός των στρωμάτων (κάθε μια από αυτές τις παραμέτρους τις αναλύσαμε στην αρχή του κεφαλαίου όταν περιγράφαμε το μοντέλο μας).

Γενικά οι Συνδυασμού που κάναμε είναι οι εξής:

- Μέγεθος Κελιού: 128, Αριθμός Στρωμάτων: 1
- Μέγεθος Κελιού: 256, Αριθμός Στρωμάτων: 1
- Μέγεθος Κελιού: 512, Αριθμός Στρωμάτων: 1
- Μέγεθος Κελιού: 128, Αριθμός Στρωμάτων: 2
- Μέγεθος Κελιού: 256, Αριθμός Στρωμάτων: 2
- Μέγεθος Κελιού: 512, Αριθμός Στρωμάτων: 2
- Μέγεθος Κελιού: 128, Αριθμός Στρωμάτων: 3
- Μέγεθος Κελιού: 256, Αριθμός Στρωμάτων: 3
- Μέγεθος Κελιού: 512, Αριθμός Στρωμάτων: 3

Στην συνέχεια χρησιμοποιώντας checkpoints από τα παραπάνω παράγουμε ένα κείμενο 10000 λέξεων. Οι λέξεις του παραπάνω κειμένου ανήκουν σε μια από τις 3 κατηγορίες:

- Παλιές Λέξεις Κειμένου(Υπήρχαν στο Αρχικό Κείμενο)
- Καινούριες Λέξεις Κειμένου που Υπάρχουν σε ένα Ελληνικό Λεξικό που βρήκαμε στο GitHub
- Καινούριες Λέξεις που δεν Υπάρχουν στο λεξικό

Για το παραπάνω φτιάξαμε πρόγραμμα, το οποίο κατηγοριοποιεί τις λέξεις. Τα αποτελέσματα θα τα δείτε στην επόμενη ενότητα.

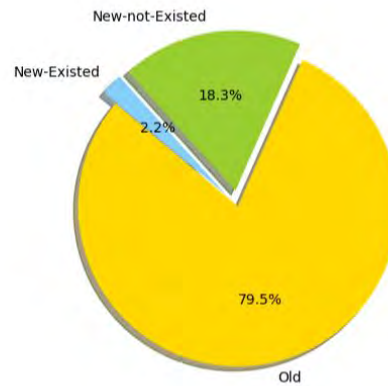
### **6.2.3 Πειραματικά Αποτελέσματα**

Ας δούμε για κάθε σύστημα το αποτέλεσμα της κατηγοριοποίησης για το αλφάβητο των 63 λέξεων και το αλφάβητο των 85 λέξεων:

- Αλφάβητο: 63 χαρακτήρες, Μέγεθος Κελιού: 128, Αριθμός Στρωμάτων: 1

Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Ανάμα, ανάματα, ανάρια, ανάριους, άρια, καλιά, κάματα, κάντρα, κάντρο, κολλά, κάρτα, κοντό, κουτά, πάλα, πάνα, πάρια, πουρά, σκότισα, σταρά, ταυ, τρένα.

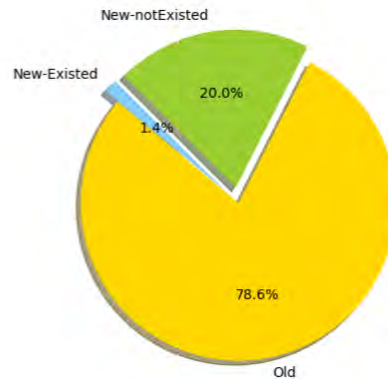


Εικόνα 22: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 63 χαρακτήρων, Μέγεθος Κελιού: 128, Αριθμός Στρωμάτων: 1

- Αλφάβητο: 85 χαρακτήρες, Μέγεθος Κελιού: 128, Αριθμός Στρωμάτων: 1

Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Ανάμα, ανάματα, ανάρια, απόντα, καρά, κάρτα, κάστορα, κουτά, κουτό, παράστησε, πολλού, πουλά, πουρά, τρίςουν.

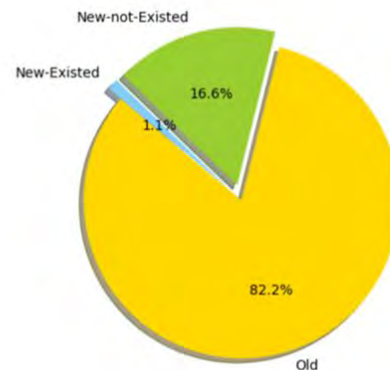


Εικόνα 23: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 85 χαρακτήρων, Μέγεθος Κελιού: 128, Αριθμός Στρωμάτων: 1

- Αλφάβητο: 63 χαρακτήρες, Μέγεθος Κελιού: 256, Αριθμός Στρωμάτων: 1

Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Άλλης, ανάματα, ανάσκητο, απάντα, άρχονταν, κάμα, καντάρια, καρά, καράτι, κάρτα, κουμάρια, κουτά, πάρια, πόρε.

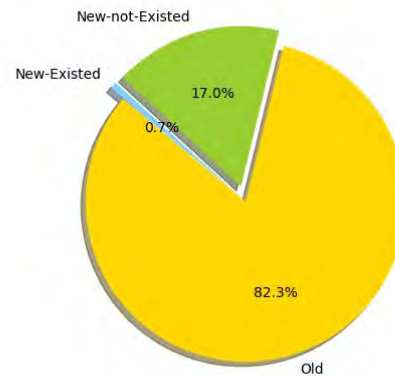


Εικόνα 24: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 63 χαρακτήρων, Μέγεθος Κελιού: 256, Αριθμός Στρωμάτων: 1

- Αλφάβητο: 85 χαρακτήρες, Μέγεθος Κελιού: 256, Αριθμός Στρωμάτων: 1

Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Ανάσκητε, άρια, καλού, κάμα, κάστορα, κουτά, μάρα, παραστούν, πάρια, χαλά.

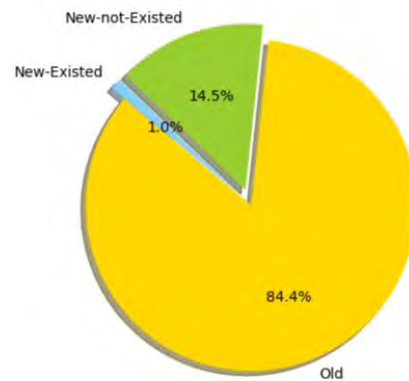


Εικόνα 25: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 85 χαρακτήρων, Μέγεθος Κελιού: 256, Αριθμός Στρωμάτων: 1

- Αλφάβητο: 63 χαρακτήρες, Μέγεθος Κελιού: 512, Αριθμός Στρωμάτων: 1

Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Άλλοις, ανάστερους, αντρειωμένη, γην, γύρη, κάμα, κάνατε, καντάρι, καράτι, μακρούς, μούρα, πάτερα, πατριά, πόλε, σταρά, στέκι, τόνο.

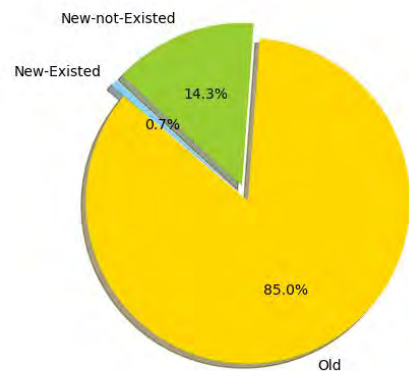


Εικόνα 26: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 63 χαρακτήρων, Μέγεθος Κελιού: 512, Αριθμός Στρωμάτων: 1

- Αλφάβητο: 85 χαρακτήρες, Μέγεθος Κελιού: 512, Αριθμός Στρωμάτων: 1

Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Ανάλατο, ανάματα, απάντα, αυλό, γην, έπαινε, κάμα, πλούσιους, πολύμορφο, πόρε, χέλι.

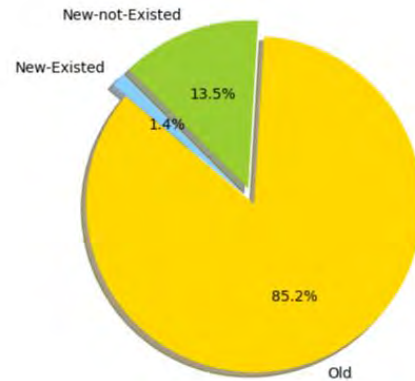


Εικόνα 27: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 85 χαρακτήρων, Μέγεθος Κελιού: 512, Αριθμός Στρωμάτων: 1

- Αλφάβητο: 63 χαρακτήρες, Μέγεθος Κελιού: 128, Αριθμός Στρωμάτων: 2

Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Γάνα, γην, κάλο, κάμα, κάματα, καράτια, κάρμα, κάστορα, κουτά, μακρά, μανάρια, μίνα, πάλα, πόλε, πουρά, σκοτισμένο.

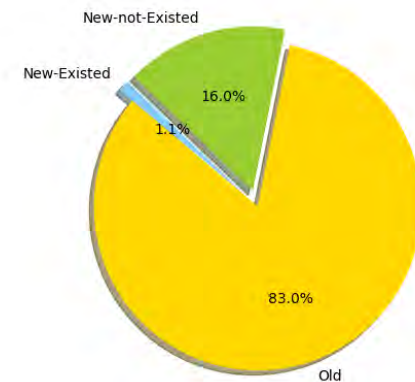


Εικόνα 28: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 63 χαρακτήρων, Μέγεθος Κελιού: 128, Αριθμός Στρωμάτων: 2

- Αλφάβητο: 85 χαρακτήρες, Μέγεθος Κελιού: 128, Αριθμός Στρωμάτων: 2

Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Άντρο, απόλεμα, απόλεμο, απόντα, γυρνούσες, κάμα, καρά, κοντό, κουμάρι, ον, πάλα, πέτα, πόλε, πολλού, πρότερα, σκέτα.



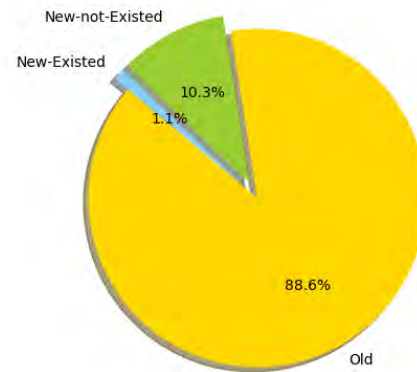
Εικόνα 29: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 85 χαρακτήρων, Μέγεθος Κελιού: 128, Αριθμός Στρωμάτων: 2



- Αλφάβητο: 63 χαρακτήρες, Μέγεθος Κελιού: 256, Αριθμός Στρωμάτων: 2

Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Ανάρια, αντρείων, απόλεμο, καίρια, κάμα, κάπο, καράτια, κοντά, μέσες, παλιέ, παράδες, πατάρια, πέτασε, πόλε, πόλο, πόρε, πότο, στήθι.

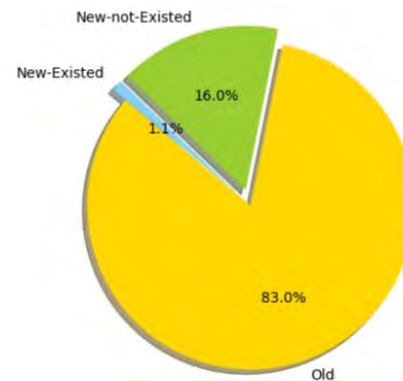


Εικόνα 30: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 63 χαρακτήρων, Μέγεθος Κελιού: 256, Αριθμός Στρωμάτων: 2

- Αλφάβητο: 85 χαρακτήρες, Μέγεθος Κελιού: 256, Αριθμός Στρωμάτων: 2

Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Άντρο, απόλεμα, απολεμο, απόντα, γυρνούσες, κάμα, καρά, κοντό, κουμάρι, όν, πάλα, πέτα, πόλε, πολλού, πρότερα, σκέτα.

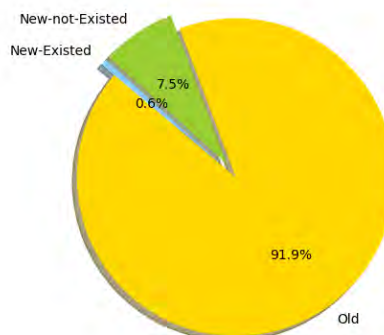


Εικόνα 31: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 85 χαρακτήρων, Μέγεθος Κελιού: 256, Αριθμός Στρωμάτων: 2

- Αλφάβητο: 63 χαρακτήρες, Μέγεθος Κελιού: 512, Αριθμός Στρωμάτων: 2

Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Ανάγει, άνθους, απάντα, απολογήθηξαν, καλούσε, κάμα, κάπο, κράνος, μακρούς, πρόβαλα, σκαλιά, σκοτώσετε.

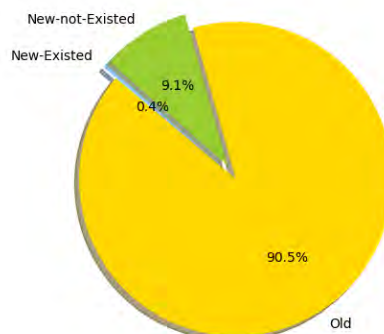


Εικόνα 32: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 63 χαρακτήρων, Μέγεθος Κελιού: 512, Αριθμός Στρωμάτων: 2

- Αλφάβητο: 85 χαρακτήρες, Μέγεθος Κελιού: 512, Αριθμός Στρωμάτων: 2

Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Ανάγει, απόλεσε, κάπο, κουράγια, κρατερό, πλούσιους, τάχη.

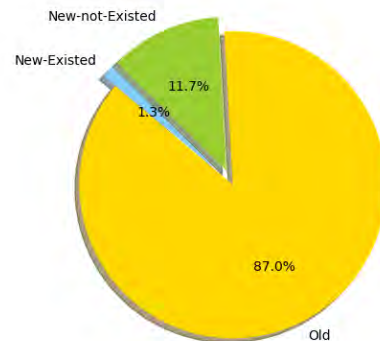


Εικόνα 33: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 85 χαρακτήρων, Μέγεθος Κελιού: 512, Αριθμός Στρωμάτων: 2

- Αλφάβητο: 63 χαρακτήρες, Μέγεθος Κελιού: 128, Αριθμός Στρωμάτων: 3

Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Άλλης, ανάμα, ανάματος, ανάμερους, απόν, γάνα, γύρη, κάμα, καρά, μακρό, πάλα, πάνα, πατάρια, πατριά, ροιά.

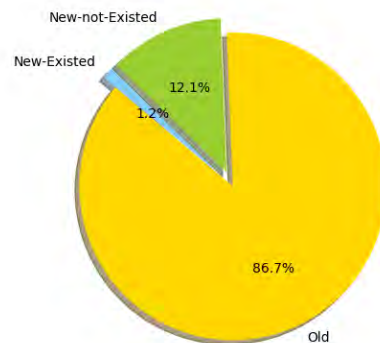


Εικόνα 34: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 63 χαρακτήρων, Μέγεθος Κελιού: 128, Αριθμός Στρωμάτων: 3

- Αλφάβητο: 85 χαρακτήρες, Μέγεθος Κελιού: 128, Αριθμός Στρωμάτων: 3

Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Αυτέ, γην, γύρας, κάμα, καράτι, κάρμα, κουμάρι, κουτό, μακρά, παλέτα, πάνα, πάρια, πατριά, πλάτης, πούρα, χάλι, χάλια.

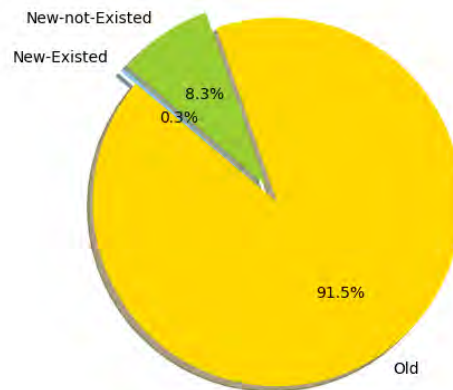


Εικόνα 35: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 85 χαρακτήρων, i

- Αλφάβητο: 63 χαρακτήρες, Μέγεθος Κελιού: 256, Αριθμός Στρωμάτων: 3

Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Ανάματα, ανάστροφο, κουράγια, μονομάχους, πόλε.

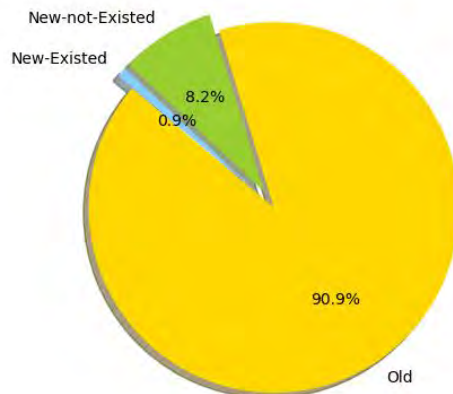


Εικόνα 36: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 63 χαρακτήρων, Μέγεθος Κελιού: 256, Αριθμός Στρωμάτων: 3

- Αλφάβητο: 85 χαρακτήρες, Μέγεθος Κελιού: 256, Αριθμός Στρωμάτων: 3

Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Ανάμα, ανέμη, απόλεμο, απόντα, γην, κάμα, κάμπη, κάντρο, πατριά, πόλε, χι.

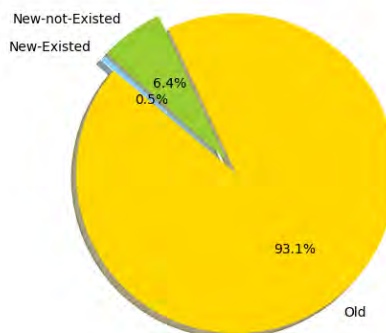


Εικόνα 37: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 85 χαρακτήρων, Μέγεθος Κελιού: 256, Αριθμός Στρωμάτων: 3

- Αλφάβητο: 63 χαρακτήρες, Μέγεθος Κελιού: 512, Αριθμός Στρωμάτων: 3

Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Άλογε, απολογήθηκε, αυλό, έστρεψε, κουράγια, πολυκάτεχους, σκαλιά.

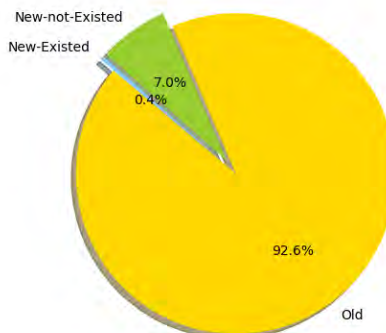


Εικόνα 38: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 63 χαρακτήρων, Μέγεθος Κελιού: 512, Αριθμός Στρωμάτων: 3

- Αλφάβητο: 85 χαρακτήρες, Μέγεθος Κελιού: 512, Αριθμός Στρωμάτων: 3

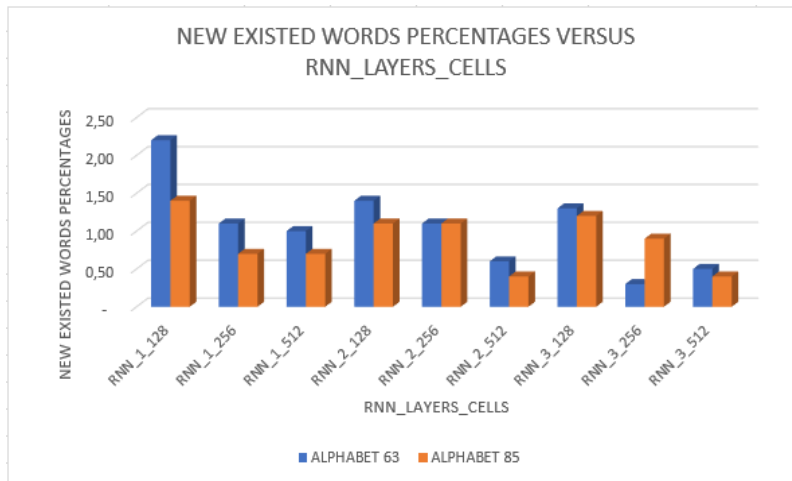
Μερικές από τις καινούργιες δόκιμες λέξεις που παρήγαγε το νευρωνικό δίκτυο και υπήρχαν στο λεξικό ήταν:

Άλογε, απολογήθηκε, αυλό, έστρεψε, κουράγια, πολυκάτεχους, σκαλιά.

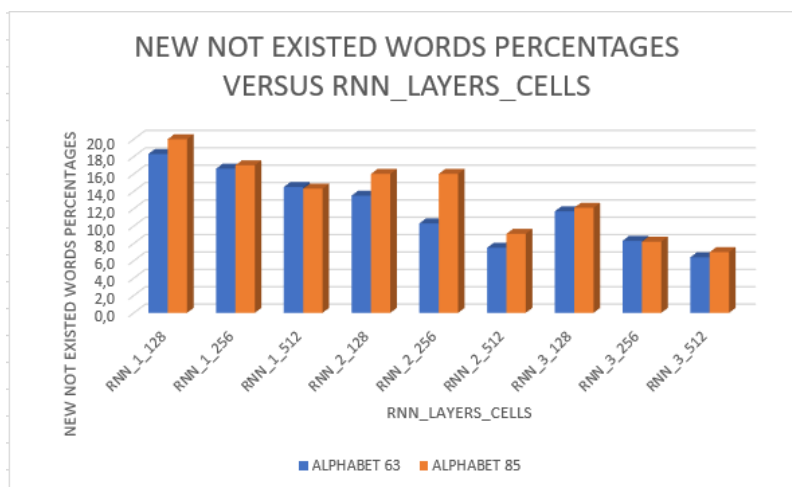


Εικόνα 39: Αποτέλεσμα Εκτέλεσης για Αλφάβητο 85 χαρακτήρων, Μέγεθος Κελιού: 512, Αριθμός Στρωμάτων: 3

Τα συνολικά αποτελέσματα φαίνονται στις παρακάτω εικόνες.



Εικόνα 40 Καινούργιες λέξεις που υπάρχουν συναρτήσει των συστημάτων νευρωνικών δικτύων

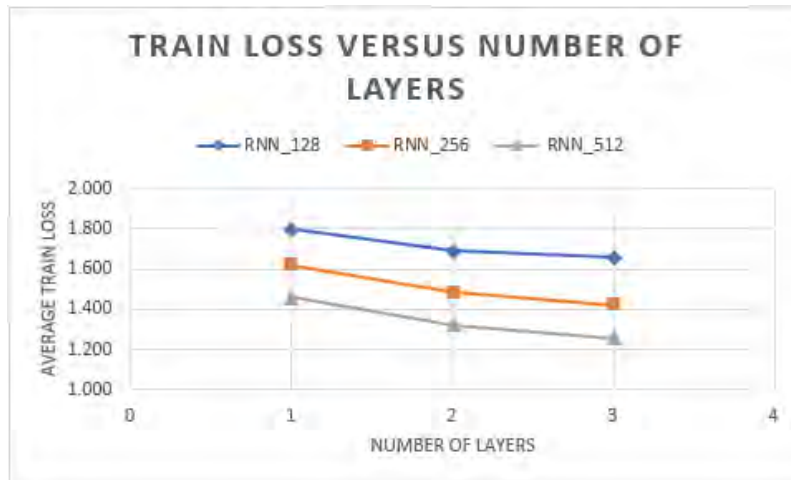


Εικόνα 41 Καινούργιες λέξεις που δεν υπάρχουν συναρτήσει των συστημάτων νευρωνικών δικτύων

Με βάση τα παραπάνω παρατηρούμε ότι, σε όλα τα συστήματα εκτός από ένα το ποσοστό των καινούργιων λέξεων που υπάρχουν στο αλφάβητο με τα 63 γράμματα είναι μεγαλύτερο από το ποσοστό των καινούργιων λέξεων που υπάρχουν στο αλφάβητο με τα 85 γράμματα καθώς επίσης και το ποσοστό των καινούργιων λέξεων που δεν υπάρχουν στο αλφάβητο με τα 63 γράμματα είναι μικρότερο από το ποσοστό των καινούργιων λέξεων που δεν υπάρχουν στο αλφάβητο με τα 85 γράμματα.

### 6.2.4 Σφάλμα εκπαίδευσης συναρτήσει της αρχιτεκτονικής του RNN

Για κάθε μια από τις αρχιτεκτονικές του RNN υπολογίσαμε το σφάλμα εκπαίδευσης με σκοπό να βρούμε τις παραμέτρους που μας εξασφαλίζουν το καλύτερο δυνατό παραγόμενο κείμενο.

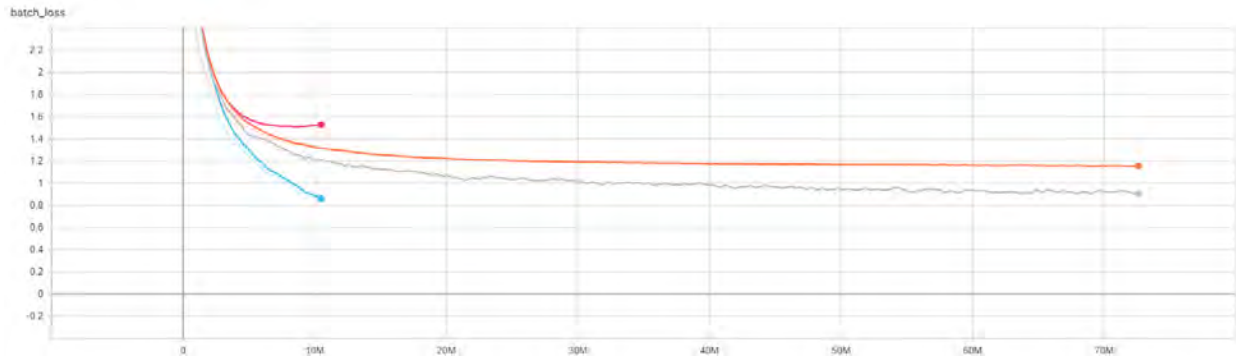


Εικόνα 42 Σφάλμα εκπαίδευσης συναρτήσει του αριθμού των επιπέδων

Από την παραπάνω εικόνα συμπεραίνουμε ότι όσο αυξάνουμε τον αριθμό των επιπέδων αλλά και το μέγεθος του κελιού τόσο βελτιώνεται και το σφάλμα εκπαίδευσης. Το καλύτερο παραγόμενο κείμενο με βάση την συνάρτηση σφάλματος έχουμε για 3 επίπεδα και μέγεθος κελιού 512.

### 6.2.5 Εκπαίδευση με διαφορετικά μεγέθη δεδομένων

Προκειμένου να αναδείξουμε το overfitting που δημιουργείται όταν έχουμε ελλιπή δεδομένα τρέξαμε το καλύτερο μοντέλο, αυτό με το μικρότερο σφάλμα εκπαίδευσης, για δύο διαφορετικά μεγέθη δεδομένων. Αρχικά τρέξαμε το μοντέλο χρησιμοποιώντας μόνο την Οδύσσεια ενώ στη συνέχεια τροφοδοτήσαμε στο μοντέλο μας εκτός από την Οδύσσεια, την Ιλιάδα. Τα αποτελέσματα φαίνονται στο παρακάτω διάγραμμα.



Εικόνα 43 batch loss vs steps

Από την παραπάνω εικόνα φαίνεται ότι χρησιμοποιώντας μικρό αριθμό δεδομένων το μοντέλο μας οδηγείται σε overfitting καθώς οι γραφικές παραστάσεις της ακρίβειας της εκπαίδευσης και της ακρίβειας της επικύρωσης αρχίζουν να αποκλίνουν ενώ τροφοδοτώντας το κείμενο με όλα τα κείμενα της Ιλιάδας και της Οδύσσειας πετυχαίνουμε καλύτερο αποτέλεσμα.

### 6.2.6 Παραγόμενο κείμενο

Παρακάτω παρατίθεται ένα απόσπασμα από το κείμενο που παρήγαγε το μοντέλο με τη καλύτερη ακρίβεια.

“Κι ως στο καλύβι του Αγαμέμνονα, το γιο του Ατρέα τριγύριζαν,  
κι οι άλλοι απ’ τον πόλεμο τον έστελναν το μάτι απά στο στήθος,  
κι αυτοί στον κάμπο ευκήθη ο Αντίλοχος, του Αινεία το γιο τον άντρα.  
  
Κι ως τα 'δενε, το μακρογίστιωτο κοντάρι του με βιάση,  
και στην καρδιά του, κι αναμέριζε το σπίτι του Οδυσσέα,  
τη γη την πατρική του εκράτησε το μακροσαγιτάρη:  
  
« Πατέρα Δία, τον Αγαμέμνονα τον πολυδοκατάτη  
και το Μηριόνη την αρμήνια σου και τον Αγαμέμνονα,  
τον αντρειωμένο γιο, πολύποθος τον Αίαντα να πατήσει.  
  
Με το καλύβι μου δεν άφηγε και την καρδιά σου εσένα,  
και το καλό σου το καλύτερο να πεί να μην το μάθεις.»”



## Βιβλιογραφία

- Chauhan, N. S. (2019, October 3). *Introduction to Artificial Neural Networks(ANN)*. Ανάκτηση από Towards Data Science: <https://towardsdatascience.com/introduction-to-artificial-neural-networks-ann-1aea15775ef9>
- Copeland, B. (χ.χ.). *Artificial intelligence*. Ανάκτηση από Encyclopaedia Britannica: <https://www.britannica.com/technology/artificial-intelligence>
- Geitgey, A. (2018, July 18). *Natural Language Processing is Fun*. Ανάκτηση από medium: <https://medium.com/@ageitgey/natural-language-processing-is-fun-9a0bff37854e>
- Gupta, M. (χ.χ.). *ML | Introduction to Data in Machine Learning*. Ανάκτηση από GeeksforGeeks: <https://www.geeksforgeeks.org/ml-introduction-data-machine-learning/>
- Gupta, M. (χ.χ.). *ML | What is Machine Learning ?* Ανάκτηση από GeeksforGeeks: <https://www.geeksforgeeks.org/ml-machine-learning/>
- Learning, D. (2019, April 30). *MARSHALL HARGRAVE*. Ανάκτηση από Investopedia: <https://www.investopedia.com/terms/d/deep-learning.asp>
- Nicholson, C. (χ.χ.). *A Beginner's Guide to Neural Networks and Deep Learning*. Ανάκτηση από pathmind: <https://pathmind.com/wiki/neural-network>
- Rouse, M. (2019, December). *artificial intelligence*. Ανάκτηση από SearchEnterpriseAI: <https://searchenterpriseai.techtarget.com/definition/AI-Artificial-Intelligence>
- Venkatachalam, M. (2019, March 1). *Recurrent Neural Networks*. Ανάκτηση από towards data science: <https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce>
- What is Machine Learning? A definition*. (2017, March 7). Ανάκτηση από ExpertSystem: <https://expertsystem.com/machine-learning-definition/>
- What is TensorFlow? Introduction, Architecture & Example*. (χ.χ.). Ανάκτηση από Guru99: <https://www.guru99.com/what-is-tensorflow.html>

Yegulalp, S. (2019, June 18). *What is TensorFlow? The machine learning library explained.*

Ανάκτηση από InfoWorld: <https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html>

Yse, D. L. (2019, January 15). *Your Guide to Natural Language Processing (NLP).* Ανάκτηση

από towards data science: <https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1>

Zsargo, B. (2019, August 8). *Top 5 applications of Natural Language Processing.* Ανάκτηση

από ChatbotsLife: <https://chatbotslife.com/top-5-applications-of-natural-language-processing-d45409c711e3>