Πρόβλεψη Τιμών Μετοχών με Χρήση Βαθιάς Μάθησης

Η Διπλωματική Εργασία
παρουσιάστηκε ενώπιον
του Διδακτικού Προσωπικού του
Πανεπιστημίου Αιγαίου

Σε Μερική Εκπλήρωση
των Απαιτήσεων για το Δίπλωμα του
Μηχανικού Πληροφοριακών και Επικοινωνιακών Συστημάτων

του
ΔΑΝΟΥΣΗ ΜΙΧΑΗΛ
ΣΕΠΤΕΜΒΡΙΟΣ 2019

Η ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΔΙΔΑΣΚΟΝΤΩΝ ΕΠΙΚΥΡΩΝΕΙ
ΤΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
ΤΟΥ ΔΑΝΟΥΣΗ ΜΙΧΑΗΛ:

ΣΤΑΜΑΤΑΤΟΣ ΕΥΣΤΑΘΙΟΣ , Επιβλέπων
Τμήμα Μηχανικών Πληροφοριακών και
Επικοινωνιακών Συστημάτων

ΠΑΠΑΣΑΛΟΥΡΟΣ ΑΝΔΡΕΑΣ, Μέλος
Τμήμα Μαθηματικών

ΓΚΟΥΜΟΠΟΥΛΟΣ ΧΡΗΣΤΟΣ, Μέλος
Τμήμα Μηχανικών Πληροφοριακών και
Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΣΕΠΤΕΜΒΡΙΟΣ 2019

ΠΕΡΙΛΗΨΗ

Αυτή η εργασία αφορά στην διερεύνηση των τρόπων με τους οποίους αλγόριθμοι μηχανικής μάθησης όπως αυτοί που υλοποιούν τεχνικές βαθιάς εκμάθησης (deep learning), μπορούν να βοηθήσουν στο να καταστούν πιο προβλέψιμες οι συμπεριφορές χρηματιστηριακών μετοχών, οι οποίες αποτελούν παράδειγμα μη γραμμικών συναρτήσεων που εξαρτώνται από μεγάλο αριθμό παραγόντων (γνωστές για την αβέβαιη φύση τους).

Ορίζοντας το σύστημά μας ως το σύνολο των χρηματιστηριακών οντοτήτων μιας αγοράς (τιμές μετοχών, δείκτες, κτλ.), χωρίζουμε την εργασία μας σε 2 μέρη τα οποία ασχολούνται με την μελέτη συστημάτων πρόβλεψης των συναρτήσεων τιμών των μετοχών τα οποία στηρίζονται σε δεδομένα ενδογενών και εξωγενών παραγόντων του συστήματος αντίστοιχα.

Πιο συγκεκριμένα στο πρώτο μέρος της, η εργασία αυτή μελετά τη διαφορά στην προβλεπτική δύναμη την οποία περιέχουν τα αριθμητικά δεδομένα υψηλής συχνότητας όπως αυτά που περιγράφονται στην έρευνα [5] με παρόμοια δεδομένα χαμηλής συχνότητας. Καταλήγουμε στο συμπέρασμα ότι η πληροφορία που μπορεί να εξαχθεί με τις τεχνικές βαθιάς μάθησης που περιγράφονται στην έρευνα [5], είναι πολύ πιο αδύναμη αν χρησιμοποιήσουμε δεδομένα χαμηλής συχνότητας αντί υψηλής.

Στο δεύτερο κομμάτι της εργασίας, κατασκευάζουμε ένα σύστημα πρόβλεψης το οποίο βασίζεται σε δεδομένα εξωγενών παραγόντων όπως ειδησεογραφικά άρθρα οικονομικού χαρακτήρα. Το σύστημα χρησιμοποιεί κάποιους από τους αλγόριθμους επεξεργασίας φυσικής γλώσσας που περιγράφονται στην έρευνα [25] ώστε να δημιουργήσει πυκνά διανύσματα νοήματος λέξεων των 100 διαστάσεων (Word embeddings) από κάποιες συντακτικές δομές (events) που εξάγονται από της προτάσεις του κειμένου, και περιέχουν 3 παραμέτρους (“δράστη”, “σχέση”, “αντικείμενο”).

Τα αποτελέσματα δείχνουν ότι τα συστήματα που μελετήσαμε στο δεύτερο κομμάτι της εργασίας, είναι πολύ ποιο αποδοτικά από αυτά του πρώτου.

ABSTRACT

This thesis investigates how machine learning algorithms, such as those that implement deep learning techniques, can help into making stock market behaviors more predictable, which are examples of nonlinear functions that depend from a large number of factors (known for their uncertain nature).

By defining our system as a set of stock market entities (stock quotes, indices, etc.), we divide our work into 2 parts that deal with the study of prediction systems of stock price functions based on intrinsic and extrinsic factors respectively.

More specifically, in its first part, this thesis studies the difference between the predictive power of high frequency numerical data such as those described in the research [5], and the predictive power of similar low frequency data. We conclude that the information that can be extracted by the deep learning techniques described in the research [5] is much weaker if we use low frequency instead of high frequency data.

In the second part of this thesis, we build a forecasting system that is based on data from exogenous factors such as economic news articles. The system uses some of the natural language processing algorithms described in the research [25] to generate dense word embeddings from some of the syntactic events extracted from the text sentences, containing 3 parameters (“actor”, “relationship”, “object”).

The results show that the systems we studied in the second part of the work are much more efficient than the first.

DANOUSIS MICHAIL

Department of Information and Communication Systems Engineering

UNIVERSITY OF THE AEGEAN

ΕΥΧΑΡΙΣΤΙΕΣ - ΑΦΙΕΡΩΣΕΙΣ

Ευχαριστώ θερμά, τον καθηγητή του Τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων, Καθηγητή Ε. Σταματάτο, που μου εμπιστεύτηκε και μου ανέθεσε την παρούσα διπλωματική εργασία, αλλά και για την πολύτιμη βοήθεια που μου παρείχε καθ' όλη τη διάρκεια της εργασίας για κάθε μικρό ή μεγάλο πρόβλημα που αντιμετώπισα. Επιπροσθέτως, ευχαριστώ τους καθηγητές της τριμελούς επιτροπής που δέχτηκαν να παραβρεθούν στην εξέταση της διπλωματικής μου εργασίας.

Τέλος, θα επιθυμούσα να ευχαριστήσω τους γονείς μου για την απεριόριστη συμπαράσταση, όχι μόνο κατά τη διάρκεια της διπλωματικής μου εργασίας, αλλά και σε όλη τη διάρκεια των σπουδών μου.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

ΚΕΦΑΛΑΙΟ 1 - ΕΙΣΑΓΩΓΗ

1.1 Μια σύντομη Ιστορία της Τεχνητής Νοημοσύνης

Όπως συμβαίνει ανά καιρούς ανάλογα με τις επιστημονικές αλλά και τεχνολογικές εξελίξεις κάποια πεδία της επιστήμης των υπολογιστών κερδίζουν το ενδιαφέρων των ερευνητών.

Τα τελευταία χρόνια ένα από αυτά τα πεδία είναι αυτό της Τεχνητής Νοημοσύνης. Το πεδίο της τεχνητής νοημοσύνης αποτελεί σημείο τομής μεταξύ πολλαπλών επιστημών όπως της πληροφορικής, της ψυχολογίας, της φιλοσοφίας, της νευρολογίας, της γλωσσολογίας και της επιστήμης μηχανικών.

Η ιδέα της δημιουργίας μιας μηχανής, ή ενός συστήματος το οποίο να διαθέτει νοημοσύνη η οποία να μιμείται την ανθρώπινη, ήταν ανέκαθεν κάτι το οποίο γοήτευε και απασχολούσε τον άνθρωπο.

Από τουλάχιστον την εποχή της αρχαίας Ελλάδας, μηχανικοί «Άνθρωποι» και τεχνητά όντα είχαν συλληφθεί από τη φαντασία των ανθρώπων. Μύθοι όπως αυτοί του Ηφαίστου που σχεδίασε τον χάλκινο γίγαντα Τάλο, ο οποίος προστάτευε το νησί της Κρήτης περιπολώντας το και απωθώντας τους εχθρούς του φανερώνουν το πόσο κοινές ήταν αυτές οι ιδέες ακόμα και σε εκείνα τα χρόνια.

Στα νεότερα χρόνια εφευρέτες όπως ο Λεονάρντο Ντα Βίντσι προσπάθησαν να ενσαρκώσουν αυτή την ιδέα με εφευρέσεις όπως τον «Ρομποτικό Ιππότη», χωρίς ωστόσο να πλησιάζουν σε μεγάλο βαθμό την ιδέα της νοημοσύνης στις εφευρέσεις τους.

Σε θεωρητικό επίπεδο από την άλλη κατά την περίοδο του 1600, σπουδαίοι μαθηματικοί και φιλόσοφοι όπως ο Τόμας Χόμπς και ο Ρενέ Ντεκάρντ, πίστευαν στην ιδέα ότι κάθε λογική σκέψη μπορεί να απεικονιστεί με τον ίδιο τρόπο που απεικονίζεται η άλγεβρα και η γεωμετρία. Μια ιδέα που είχε τις ρίζες της στον 4ο αιώνα προ Χριστού, πατέρας της οποίας ήταν ο Αριστοτέλης ο οποίος την ονόμασε συλλογιστική λογική.

Όπως δήλωσε ο Τόμας Χόμπς στο βιβλίο του Λεβιάθαν, «Όταν ένας άνθρωπος σκέφτεται, δεν κάνει τίποτα άλλο παρά να συλλάβει ένα συνολικό ποσό, από την προσθήκη των τεμαχίων ή να συλλάβει ένα υπόλοιπο, από την αφαίρεση ενός ποσού από το άλλο ... Και αν και σε κάποια πράγματα (όπως στους αριθμούς), εκτός από την προσθήκη και την αφαίρεση, οι άνθρωποι ονομάζουν και άλλες πράξεις, όπως πολλαπλασιασμό και διαίρεση. Στην πραγματικότητα είναι το ίδιο. Ο πολλαπλασιασμός, είναι μόνο η προσθήκη των ίσων πραγμάτων, και η διαίρεση, είναι να αφαιρούμε ένα πράγμα, όσο συχνά μπορούμε»

Προχωρώντας στο 1840 και μετά την Βιομηχανική επανάσταση, ο Charles Babbage σχεδιάζει μια εξαιρετικά καινοτόμα για την εποχή μηχανή, η οποία θεωρείται από πολλούς ο πρώτος προγραμματίσιμος Υπολογιστής και παρόλο που δεν ολοκληρώθηκε την τότε εποχή, άνοιξε τον δρόμο για τον σχεδιασμό μηχανών τέτοιου είδους. Αυτός είναι και ο λόγος για τον οποίο σήμερα θεωρείται από πολλούς ο «πατέρας του υπολογιστή».

Μια δεκαετία αργότερα ο Άγγλος μαθηματικός και Φιλόσοφος, George Boole έφερε την επανάσταση στο πεδίο των Υπολογιστών και έκανε τα πρώτα βήματα προς την «Τεχνητή Νοημοσύνη» όπως την έχουμε σήμερα στο μυαλό μας, δημιουργώντας την «Boolean Logic» και αντικαθιστώντας τον πολλαπλασιασμό με την πράξη «AND» και την πρόσθεση με την πράξη «OR».

Η ιδέα ωστόσο μπορούμε να πούμε απέκτησε το μέσο που θα τη βοηθούσε να κάνει τη μετάβασή της από τη σφαίρα της φαντασίας, στον φυσικό μας κόσμο, όταν το 1936 ο Άγγλος μαθηματικός, καθηγητής της λογικής και κρυπτογράφος, Άλαν Μάθισον Τούρινγκ εξέφρασε την ιδέα της «Καθολικής Μηχανής Τούρινγκ», μια μηχανή η οποία αντί να είναι προσχεδιασμένη για να εκτελεί κάποιους συγκεκριμένους υπολογισμούς για κάποια συγκεκριμένη εργασία, να είναι

σχεδιασμένη να μπορεί να υπολογίσει, οτιδήποτε υπολογίσιμο αν της δώσει κανείς τις κατάλληλες εντολές. Αυτή η ιδέα η οποία είναι η βάση του μοντέρνου Computing, έφερε στη ζωή μια πρώιμη μορφή ηλεκτρονικού υπολογιστή, σχεδιασμένο με ηλεκτρικούς ρότορες, ο οποίος δημιουργήθηκε με σκοπό την αποκρυπτογράφηση μηνυμάτων.

Περίπου μια δεκαετία αργότερα ο Τούρινγκ θα καθόταν απέναντι σε έναν συνάδελφό του παίζοντας σκάκι εκτελώντας ο ίδιος κινήσεις ενός αλγορίθμου σχεδιασμένου για μία μηχανή Τούρινγκ καθώς δεν υπήρχε ακόμα κάποιο μηχάνημα αρκετά ισχυρό για να μπορέσει να εκτελέσει τον αλγόριθμό του σε εύλογο χρονικό διάστημα. Φημολογείται επίσης πως ενώ ο αλγόριθμος έχασε από τον συνάδελφό του, κατάφερε να κερδίσει στη συνέχεια έχοντας αντίπαλο την γυναίκα του Champernowne. Η σύγχρονη ιδέα της Τεχνητής Νοημοσύνης είχε πλέον γεννηθεί και ήταν θέμα χρόνου με τον ρυθμό ανάπτυξης της υπολογιστικής δύναμης των Ηλεκτρονικών Υπολογιστών, συστήματα Τεχνητής Νοημοσύνης να καταφέρουν να νικήσουν τον άνθρωπο όχι μόνο στο σκάκι, άλλα και σε πολλά άλλα παιχνίδια περίπλοκης σκέψης. Με μια από τις κορυφαίες στιγμές αυτών των συστημάτων να λαμβάνει χώρα το 2016, όταν το σύστημα τεχνητής νοημοσύνης «**AlphaGo**» της εταιρείας Google κατάφερε να νικήσει τον παγκόσμιο πρωταθλητή Lee Sedol σε μια σειρά παιχνιδιών «Go» ένα παιχνίδι που χαρακτηρίζεται από την πολυπλοκότητα του πεδίου κινήσεων και στρατηγικών που μπορούν να εφαρμοστούν σε κάθε παρτίδα.

1.2 Η Τεχνητή Νοημοσύνη Σήμερα

Σήμερα η τεχνητή νοημοσύνη είναι κάτι που συναντάμε σε μία πληθώρα πληροφοριακών συστημάτων και εφαρμογών. Από συστήματα που χρησιμοποιούν τις κινήσεις χρηστών σε e-shops για να παρέχουν πιο εύστοχες διαφημίσεις, μέχρι ανακατασκευή παλιών ασπρόμαυρων ταινιών σε έγχρωμες. Οι εφαρμογές τεχνητής νοημοσύνης έχουν εισχωρήσει πλέον αρκετά στην καθημερινότητά μας φαίνεται πως τουλάχιστον στο κοντινό μέλλον αυτό το φαινόμενο μόνο αυξητική τάση μπορεί να έχει.

Κάποιοι από τους λόγους αυτής της άνθησης του επιστημονικού και εμπορικού ενδιαφέροντος πάνω στο πεδίο είναι, η αύξηση της υπολογιστικής δύναμης των υπολογιστών, αλλά και οι εξελίξεις που έχουν υπάρξει τα τελευταία χρόνια στην επιστήμη της νευρολογίας, οι οποίες μας βοήθησαν στο να αποκτήσουμε μια περισσότερο εμπεριστατωμένη γνώση για τους τρόπους λειτουργίας του εγκεφάλου με αποτέλεσμα να μπορέσουμε να αναπτύξουμε καλύτερους αλγόριθμους μίμησης της.

Το πεδίο της Τεχνητής Νοημοσύνης έχει πλέον πολλούς κλάδους ανάλογα με τον τρόπο εφαρμογής της και τους αλγόριθμους που χρησιμοποιούνται. Κάποια παραδείγματα θα μπορούσαν να είναι ο κλάδος της συμβολικής λογικής, τα expert systems και τα γραφήματα γνώσης.

Ωστόσο ίσως ο ποιο δημοφιλής αυτή τη περίοδο κλάδος της τεχνητής νοημοσύνης είναι αυτός της μηχανικής μάθησης, στον οποίο θα εμβαθύνουμε στο επόμενο κεφάλαιο της εργασίας.

1.3 Η Έρευνά μας

1.3.1 Το πεδίο της έρευνας

Η χρηματιστηριακή αγορά είναι ένας χώρος εμπορίου, του οποίου το κύριο εμπορεύσιμο αγαθό είναι τμήματα ιδιοκτησίας εταιριών (μετοχές), ή και άλλων περιουσιακών οντοτήτων. Ο χώρος αυτός, ο οποίος μπορεί να μελετηθεί ως ένα ενιαίο σύστημα οικονομικών μεταβλητών, επηρεάζει σε μεγάλο βαθμό την οικονομία και κατ' επέκταση έναν μεγάλο αριθμό κοινωνικών πεδίων.

Ωστόσο το σύστημα αυτό χαρακτηρίζεται από την πολυπλοκότητα της συμπεριφοράς του καθώς όπως θα δούμε και στη συνέχεια αναλυτικότερα, αυτή επηρεάζεται από έναν πολύ μεγάλο αριθμό παραγόντων. Το σύστημα αυτό είναι τόσο περίπλοκο και ο όγκος της πληροφορίας που πρέπει να διαχειριστεί κανείς για να αναλύσει τη συμπεριφορά του, τόσο μεγάλος, που μέχρι και τη

δεκαετία του 70 μία από τις σημαντικότερες έρευνες για την συμπεριφορά του συστήματος της χρηματιστηριακής αγοράς αναφέρει "Prices are determined randomly, it is impossible to outperform the market." [3]. Όμως στη συνέχεια η ανάπτυξη της επιστήμης των υπολογιστών και οι αλγόριθμοι τεχνητής νοημοσύνης απέδειξαν εμπειρικά ότι ο παραπάνω ισχυρισμός δεν ισχύει.

1.3.2 Περιγραφή έρευνας

Σε αυτή την εργασία θα επικεντρωθούμε στους τρόπους με τους οποίους η Μηχανική Μάθηση μπορεί να μας βοηθήσει να προβλέψουμε τη συμπεριφορά συναρτήσεων οι οποίες εξαρτώνται από ένα μεγάλο αριθμό παραγόντων όπως είναι οι συναρτήσεις τιμών χρηματιστηριακών μετοχών.

Πρόκειται αρχικά να εξετάσουμε αν η εφαρμογή των μοντέλων που χρησιμοποιήθηκαν στην έρευνα των [5] η οποία βασίστηκε σε χρηματιστηριακά δεδομένα μεγάλης συχνότητας, είναι εξίσου αποδοτική σε δεδομένα μικρότερης συχνότητας όπως για παράδειγμα τα ημερήσια δεδομένα.

Ειδικότερα στην έρευνά μας θα χρησιμοποιήσουμε ιστορικά δεδομένα από τιμές μετοχών του δείκτη S&P 500, κατά την περίοδο 10/2006 ως 11/2013, για να εκπαιδεύσουμε μέσω αλγορίθμων μηχανικής μάθησης και πιο συγκεκριμένα βαθιάς μάθησης (Deep Learning), μοντέλα τα οποία θα μπορούν να προβλέπουν όσο αποδοτικότερα γίνεται την μελλοντική κατεύθυνση των μετοχών. Η επιλογή της συγκεκριμένης χρονικής περιόδου γίνεται για λόγους συμβατότητας και συνεπώς δυνατότητας σύγκρισης των αποτελεσμάτων με τα μοντέλα πρόβλεψης στο δεύτερο κομμάτι της εργασίας.

Αφού καταλήξουμε σε κάποιο πόρισμα για την προβλεπτική δύναμη των δεδομένων χαμηλής και ψηλής συχνότητας, και την σχέση μεταξύ τους, θα προχωρήσουμε στο δεύτερο κομμάτι της εργασίας το οποίο αφορά την ανάπτυξη ενός συστήματος πρόβλεψης της κίνησης της αγοράς το οποίο ως δεδομένα εισόδου θα χρησιμοποιεί ειδησεογραφικά κείμενα οικονομικού ενδιαφέροντος.

1.3.3 Στόχοι

Στόχος της έρευνας αυτής είναι να προσφέρει μια εμπεριστατωμένη άποψη για το κατά πόσο οι σύγχρονοι αλγόριθμοι Μηχανικής Μάθησης είναι ικανοί να κάνουν σωστές προβλέψεις για την κίνηση της χρηματιστηριακής αγοράς, και το ποιο είδος δεδομένων είναι καταλληλότερο για να χρησιμοποιηθεί από τους αλγόριθμους αυτούς.

Η ανάλυση της συμπεριφοράς του συστήματος της αγοράς, και οι πιθανές προβλέψεις που μπορούν να γίνουν πάνω στην μελλοντική κίνησή του συνολικού αυτού συστήματος, μπορούν να αξιοποιηθούν ως προς τη βελτίωση κοινωνικών χώρων που εξαρτώνται από την οικονομία.

Θεωρούμε πως το προβλήματα που αντιμετωπίζουμε σε αυτή την εργασία έχει σε κάποιο βαθμό επεκτασιμότητα και σε άλλους τομείς που στηρίζονται σε αποφάσεις παρόμοιας φύσεως.

1.3.4 Τι πετύχαμε

Τα αποτελέσματα από τα πειράματα που διεξήχθησαν στην παρούσα εργασία μπορούν να συμπυκνωθούν σε 2 βασικά σημεία. Αρχικά φαίνεται ότι η συχνότητα των δεδομένων αυτής της φύσης παίζει πολύ σημαντικό ρόλο στην απόδοση του συστήματος πρόβλεψης. Αυτό πιθανώς οφείλεται στον χρόνο απόκρισης της ανθρώπινης απόφασης. Έπειτα στο δεύτερο κομμάτι της εργασίας διαπιστώσαμε ότι στα δεδομένα ημερήσιας συχνότητας, η πληροφορία που μπορεί να εξαχθεί από ειδησεογραφικά άρθρα έχει πολύ μεγαλύτερη προβλεπτική δύναμη από αυτήν που εξάγεται από τις αριθμητικές τιμές της μετοχής.

ΚΕΦΑΛΑΙΟ 2 - ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

2.1 Τι είναι Μηχανική Μάθηση

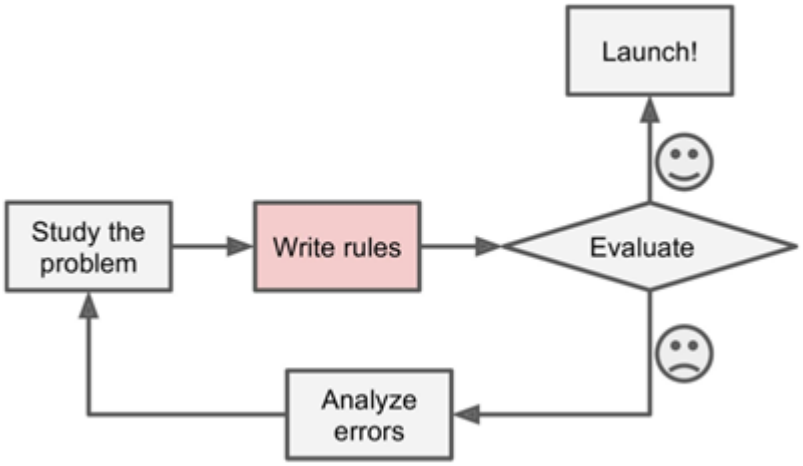
Αν και πολλοί επιστήμονες έχουν δώσει το δικό τους ορισμό για το τι είναι Μηχανική Μάθηση, θα δανειστούμε αυτόν του Tom Mitchell ο οποίος στο βιβλίο του [13] αναφέρει το εξής:

«Το πεδίο της μηχανικής μάθησης αφορά το θέμα του τρόπου κατασκευής προγραμμάτων ηλεκτρονικών υπολογιστών που αυτομάτως βελτιώνονται με την εμπειρία.

Ένα πρόγραμμα ηλεκτρονικού υπολογιστή λέγεται ότι μαθαίνει από την εμπειρία E σε σχέση με κάποια τάξη εργασιών T και μέτρηση απόδοσης P, εάν η απόδοσή του σε εργασίες στο T, όπως μετράται με P, βελτιώνεται με την εμπειρία E.»

Για να αναλύσουμε όμως περισσότερο την έννοια του Machine Learning θα μπορούσαμε να δανειστούμε ένα κομμάτι από την εισαγωγή του βιβλίου «Hands On Machine Learning with Sci-Kit & Tensorflow» [6] το οποίο βάζει τους αναγνώστες του στην έννοια λέγοντας:

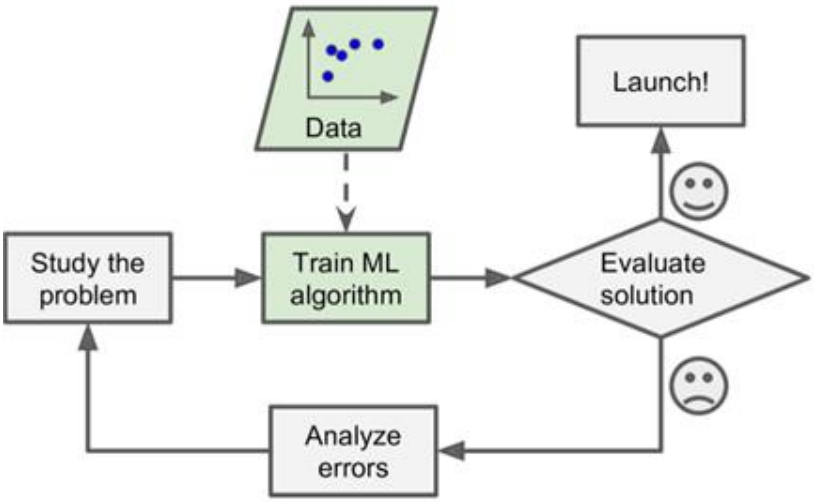
Ας θεωρήσουμε ένα απλό πρόβλημα που καλούμαστε να λύσουμε. Αν θα σχεδιάζαμε τη λύση χρησιμοποιώντας την παραδοσιακή μέθοδο φαίνεται στο παρακάτω σχήμα:



Εικόνα 2-1 Επεξήγηση Μηχανικής Μάθησης[6]

Αρχικά δηλαδή θα εξετάζαμε το προβλημα, θα γράφαμε ένα σετ κανόνων, θα αξιολογούσαμε τη λύση και στη συνέχεια αν είμασταν ικανοποιημένοι με το αποτέλεσμα θα καταλήγαμε στο τελικό αποτέλεσμα, διαφορετικά θα αναλύαμε τα λάθη μας και θα επαναλαμβάναμε τη διαδικασία.

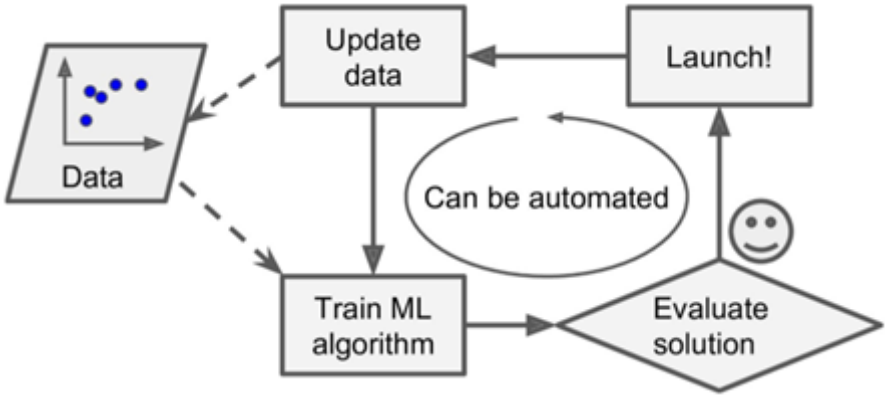
Αν όμως χρησιμοποιούσαμε μεθόδους μηχανικής μάθησης το διάγραμμά μας θα έμοιαζε ως εξής:



Εικόνα 2-2 Επεξήγηση Μηχανικής Μάθησης [6]

Δηλαδή μετά τη μελέτη του προβλήματος θα αφήναμε μια ML μέθοδο να εκπαιδευτεί μέσω δεδομένων ώστε να καταλήξει σε ένα σετ κανόνων η οποία θα πρέπει και πάλι να αξιολογηθεί.

Παρατηρούμε ότι το τελευταίο διάγραμμα έχει τη δυνατότητα να αυτοματοποιηθεί ως εξής:



Εικόνα 2-3 Επεξήγηση Μηχανικής Μάθησης [6]

Αυτή η παρατήρηση είναι πολύ σημαντική καθώς καθιστά δυνατή τη δημιουργία λύσης στο πρόβλημά μας η οποία έχει σχεδιαστεί εξ ολοκλήρου από τα δεδομένα και τον ML αλγόριθμο.

Όπως μπορεί όμως να γίνει αντιληπτό, η μηχανική μάθηση είναι ένας καλός τρόπος για να μας βοηθήσει σε συγκεκριμένα προβλήματα. Μερικά από αυτά είναι:

- Προβλήματα για τα οποία οι υπάρχουσες λύσεις απαιτούν πολλούς χειρισμούς ή μακρούς καταλόγους κανόνων: ένας αλγόριθμος Machine Learning μπορεί συχνά να απλοποιήσει τον κώδικα και να αποδώσει καλύτερα.
- Πολύπλοκα προβλήματα για τα οποία δεν υπάρχει καλή λύση χρησιμοποιώντας μια παραδοσιακή προσέγγιση: οι καλύτερες τεχνικές Machine Learning μπορούν να βρουν μια λύση.
- Έντονα μεταβαλλόμενα περιβάλλοντα: ένα σύστημα Machine Learning μπορεί να προσαρμοστεί σε νέα δεδομένα.
- Απόσπαση πληροφοριών για σύνθετα προβλήματα και μεγάλα ποσά δεδομένων. [6]

2.2 Πως Λειτουργεί η Μηχανική Μάθηση

Για να λειτουργήσει ένας αλγόριθμος μηχανικής μάθησης χρειάζεται 2 στοιχεία.

- Δεδομένα
- Τι να κάνει με τα δεδομένα

Το πρώτο στοιχείο είναι πιο εύκολο να αναλυθεί καθώς τα δεδομένα παρόλο που μπορεί να περιλαμβάνουν πολλές μορφές και έννοιες, όλοι έχουμε μια γενική εικόνα για το τι μορφή μπορεί να έχουν. Οι μεγαλύτερες κατηγορίες στις οποίες μπορούν να χωριστούν τα δεδομένα κατάλληλα για μηχανική μάθηση είναι οι εξής:

- Διανύσματα: Ένα σύνολο χαρακτηριστικών (features), το οποίο μπορεί να αφορά αριθμητικές τιμές (ύψος, ηλικία, απόσταση, κτλ.), ή και κατηγορικές τιμές (χρώμα, είδος, κτλ.). Ένα παράδειγμα dataset με διανυσματικά δεδομένα είναι αυτό της παρακάτω εικόνας
- Πίνακες: Εικόνες, Γεωγραφικά δεδομένα, κτλ.
- Ακολουθίες Χαρακτήρων: Κείμενα, Ακολουθίες γονιδίων, κα.
- Δομημένα Αντικείμενα: XML Κείμενα, Γραφήματα.

2.2.1 Είδη αλγόριθμων Μηχανικής Μάθησης

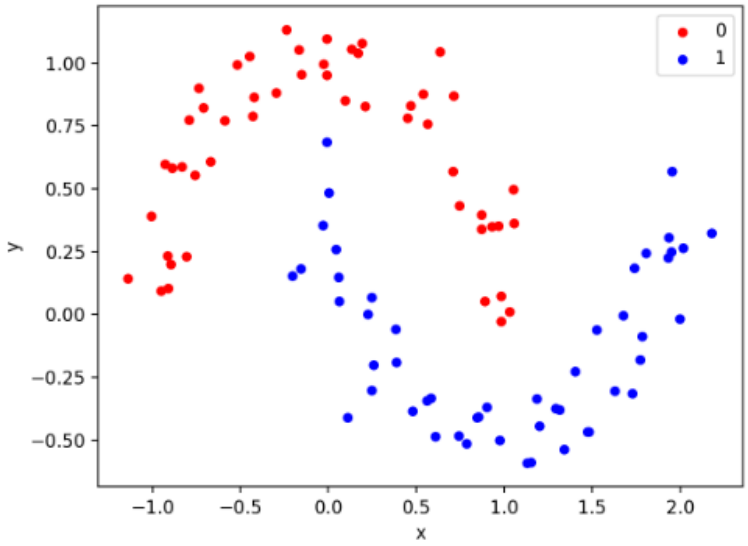
Το δεύτερο στοιχείο ωστόσο αποτελεί ένα πιο σύνθετο κομμάτι του πεδίου και είναι αυτό με το οποίο ασχολούνται οι περισσότεροι ερευνητές του τομέα της μηχανικής μάθησης. Το τι θα κάνει με τα δεδομένα, ή αλλιώς ποια θα είναι η λειτουργία του αλγορίθμου μπορεί να απαντηθεί ως εξής.

Οι Αλγόριθμοι μηχανικής μάθησης ποικίλουν στους τρόπους με τους οποίους λειτουργούν και στις προσεγγίσεις τους στα προβλήματα που προσπαθούν να λύσουν, ωστόσο η πιο συνηθισμένη κατηγοριοποίηση τους γίνεται με βάση το κατά πόσο εκπαιδεύτηκαν με ανθρώπινη επίβλεψη: Εποπτευόμενοι (Supervised), μη-εποπτευόμενοι (Unsupervised), ημί-εποπτευόμενοι (Unsupervised), και ενίσχυση μάθησης (Reinforcement Learning).

2.2.1.1 Εποπτευόμενοι

Εποπτευόμενους θεωρούμε τους αλγόριθμους μηχανικής μάθησης, οι οποίοι μαθαίνουν μία συνάρτηση σύνδεσης των δεδομένων εισαγωγής, με τα δεδομένα στόχου, μέσω δεδομένων που έχουν τη μορφή παραδειγμάτων αυτής της σύνδεσης (με ποιο input παίρνω ποιο output). Τα δεδομένα εισόδου έχουν τη μορφή διανύσματος (ένα σύνολο features), και τα δεδομένα εξόδου έχουν τη μορφή ετικέτας (label), η οποία υποδηλώνει την κλάση στην οποία ανήκει τα αντικείμενο με τα συγκεκριμένα χαρακτηριστικά.

Στην παρακάτω εικόνα φαίνεται ένα dataset διανυσμάτων με 2 χαρακτηριστικά (την τιμή x τους, και την τιμή y τους),το καθένα από τα οποία έχει ετικέτα (label) «0» ή «1».



Εικόνα 2-4 Παράδειγμα δεδομένων αλγόριθμου εποπτευόμενης μάθησης [6]

(ένας supervised αλγόριθμος μπορεί να εφαρμοστει στο παραπάνω dataset, ετσι ώστε να αναπτύξει ένα μοντέλο το οποίο θα κατατάσει ένα σημείο ως «0» ή «1», ανάλογα με τις τιμές των x και y του.)

Κάποια παραδείγματα τέτοιου είδους αλγορίθμων είναι:

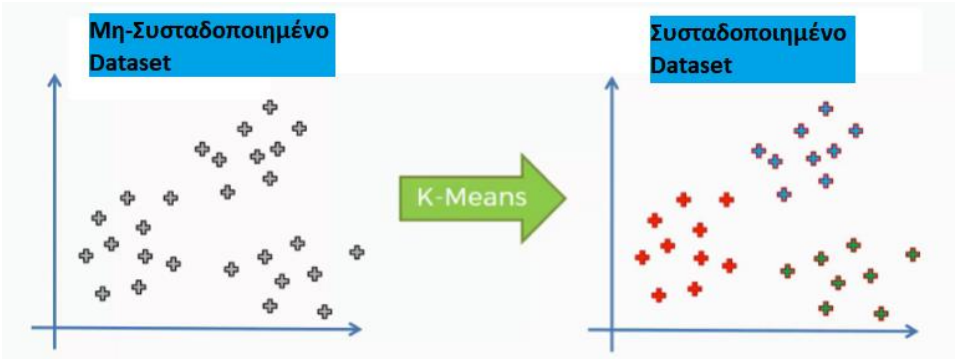
- k-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVMs)
- Decision Trees and Random Forests
- Neural networks

2.2.1.2 Μη-εποπτευόμενοι

Μη-εποπτευόμενους λέμε τους αλγόριθμους οι οποίοι καλούνται να εξάγουν συσχετίσεις (κ.α) για δεδομένα τα οποία δεν έχουν κατηγοριοποιηθεί. Συνήθως τέτοιου είδους αλγόριθμοι χρησιμοποιούνται για να συσταδοποιήσουν δεδομένα, ή να εξάγουν συσχετίσεις από αυτά. Κάποια παραδείγματα τέτοιων αλγορίθμων είναι:

- Συσταδοποίηση: k-Means, HCA
- Οπτικοποίηση και μείωση διαστάσεων: PCA, Kernel PCA
- Εξαγωγή κανόνων συσχέτισης: Apriori, Eclat

Στην παρακάτω εικόνα φαίνεται ένα παράδειγμα του αποτελέσματος της εφαρμογής του αλγορίθμου k-Means σε μη συσταδοποιημένα δεδομένα.

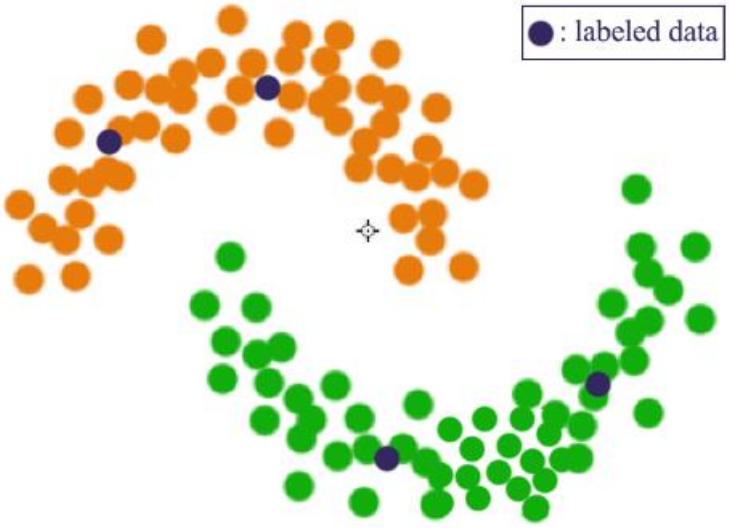


Εικόνα 2-5 Παράδειγμα μη-εποπτευόμενης μάθησης(https://towardsdatascience.com)

2.2.1.3 Ημι-εποπτευόμενοι

Ημί-Εποπτευόμενους ονομάζουμε τους αλγόριθμους οι οποίοι δουλεύουν με datasets τα οποία έχουν και ετικετοποιημένα δείγματα (labeled), αλλά τα περισσότερα συνήθως είναι μη-ετικετοποιημένα (unlabeled). Συνήθως τέτοιοι αλγόριθμοι είναι ο συνδυασμός εποπτευόμενων και μη-εποπτευόμενων αλγορίθμων. Χρησιμοποιούνται πολλές φορές για κατηγοριοποίηση σε δεδομένα τα οποία κατηγοριοποιούν χρήστες δειγματικά.

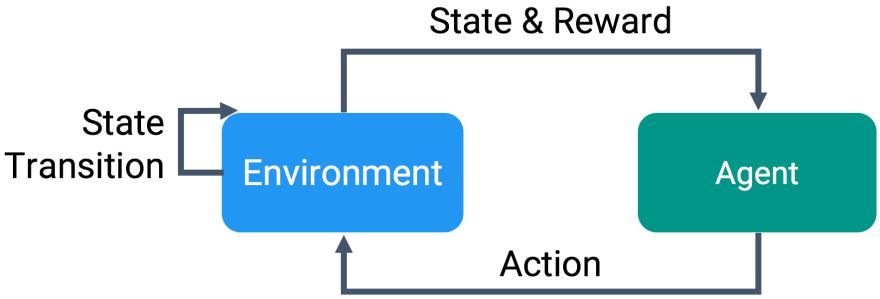
Ένα dataset Ημί-Εποπτευόμενου αλγορίθμου θα μπορούσε να είναι το παρακάτω:



Εικόνα 2-6 Δεδομένα ημί-εποπτευόμενης μάθησης(https://towardsdatascience.com)

2.2.1.4 Μάθηση με ενίσχυση

Μάθηση με ενίσχυση. Η βασική ιδέα πίσω από αυτού του είδους τους αλγόριθμους, είναι η χρήση κάποιων στοιχείων που ονομάζονται πράκτορες. Πράκτορας ονομάζεται μια οντότητα του αλγορίθμου, η οποία βάση ενός συστήματος ανταμοιβών και ποινών, οι οποίες καθορίζονται ανάλογα με τη κατάσταση του πράκτορα και του περιβάλλοντος στο οποίο δρα, αναπτύσσει όσο το δυνατόν πιο συμφέρουσες στρατηγικές σε βάθος χρόνου, μέσω της μεθόδου προσπάθειας και λάθους (trial and error).

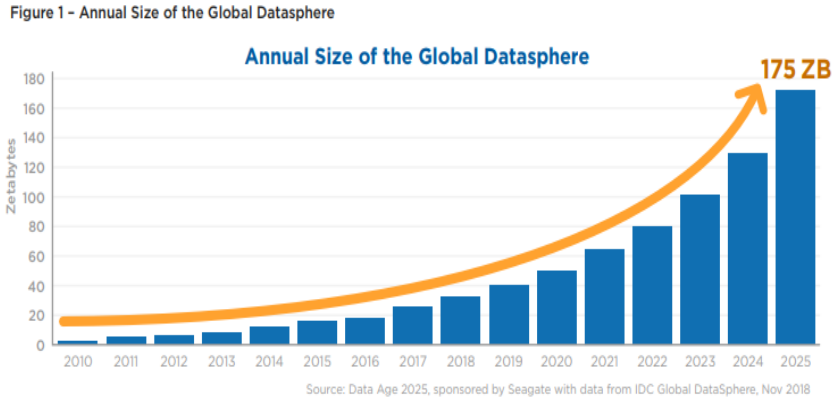


Εικόνα 2-7 Επεξηγηματικό σχήμα μάθησης με ενίσχυση[6]

2.3 Γιατί Μηχανική Μάθηση

Ο κύριος λόγος για τον οποίο το πεδίο της Μηχανικής Μάθησης έχει κερδίσει τόσο ερευνητικό και όχι μόνο ενδιαφέρον, είναι ότι για να είναι χρήσιμη η μηχανική μάθηση, δηλαδή να παράγουμε μοντέλα υψηλής απόδοσης σε λογικό χρονικό διάστημα, χρειαζόμαστε δύο πράγματα. Το ένα είναι υπολογιστική ισχύς και το άλλο είναι δεδομένα. Όπως γνωρίζουμε, η υπολογιστική ισχύς είναι κάτι που αυξάνεται σταθερά κατά τη διάρκεια των ετών. Το δεύτερο στοιχείο που χρειάζεται η Μηχανική Μάθηση για να είναι αποτελεσματική (δεδομένα) είναι κάτι που τα τελευταία χρόνια γνώρισε τεράστια ανάπτυξη. Συγκεκριμένα σε μία έρευνα που διεξήχθη το 2014 με τίτλο «Big Data Characteristics, Value Chain and Challenges» [14] αναφέρεται χαρακτηριστικά ότι το 2003, είχαν δημιουργηθεί 5 exabytes δεδομένων παγκοσμίως, ενώ 2014 το ποσό δεδομένων αυτό δημιουργούταν μόνο σε δύο ημέρες.

Η ποσότητα των δεδομένων στον ψηφιακό κόσμο έφθασε τα 2.72 zettabytes το 2012, το 2015 είχαν δημιουργηθεί παγκοσμίως παραπάνω από 8 zettabytes, ενώ σήμερα το σύνολο των zettabytes δεδομένων που υπάρχουν παγκοσμίως είναι περισσότερα από 33. Σύμφωνα με την εταιρεία «International Data Corporation (IDC)» το σύνολο των zettabytes παγκοσμίως αναμένετε να φτάσει τα 175 μέχρι το 2025. [17]



Εικόνα 2-8 Γράφημα συνολικού όγκο δεδομένων ανά χρόνο[17]

Τα δεδομένα λοιπόν που έχει στη διάθεσή του ο άνθρωπος, αυξάνονται συνεχώς σε όγκο και έτσι γίνεται όλο και δυσκολότερα διαχειρίσιμα από τον ανθρώπινο εγκέφαλο.

Η ανάγκη συνεπώς του ανθρώπου για εργαλεία που θα τον βοηθούν να «ανεβάζει» ψηλότερα στην «πυραμίδα της αξίας των δεδομένων», όπως ορίστηκε από τον R. L. Ackoff το 1988 (βλ 2-9) τα δεδομένα που έχει στη διάθεσή του, μεγαλώνει. Έτσι οι εφαρμογές της επιστήμης της μηχανικής μάθησης και τα προβλήματα που μπορεί να λύσει αυξάνονται ταχύρρυθμα.

Εικόνα 2-9 Πυραμίδα αξίας των δεδομένων[17]

2.4 Βαθιά εκμάθηση και Τεχνητά Νευρωνικά Δίκτυα

2.4.1 Εισαγωγή

Ένα από τα πλέον δημοφιλή και αναπτυσσόμενα υπο-πεδία της μηχανικής μάθησης είναι αυτό της βαθιάς εκμάθησης (Deep Learning).

Στην εισαγωγή του βιβλίου «Deep Learning» [26] ο συγγραφέας εξηγεί την έννοια της βαθιάς εκμάθησης λέγοντας πως είναι τρόπος με τον οποίο επιτρέπουμε στους ηλεκτρονικούς υπολογιστές, να καταλάβουν τον κόσμο μέσω της ιεραρχίας των εννοιών, οπού δηλαδή η κάθε έννοια είναι συνδυασμός απλούστερων εννοιών.

Σήμερα το πεδίο της βαθιάς εκμάθησης έχει σχεδόν ταυτιστεί με τη Μελέτη των Τεχνητών Νευρωνικών Δικτύων.

Ένας από τους σημαντικότερους λόγους αυτής της άνθησης του επιστημονικού και όχι μόνο ενδιαφέροντος πάνω στο πεδίο, εκτός της αύξηση της υπολογιστικής δύναμης, είναι και οι εξελίξεις που έχουν υπάρξει τα τελευταία χρόνια στην επιστήμη της νευρολογίας, οι οποίες μας βοήθησαν στο να αποκτήσουμε μια πιο εμπεριστατωμένη άποψη για τους τρόπους λειτουργίας του εγκεφάλου με αποτέλεσμα να μπορέσουμε να αναπτύξουμε καλύτερους αλγόριθμους «μίμησης» της.

Βιολογικά Νευρωνικά Δίκτυα

Στην κατανόηση της έννοιας των Τεχνητών Νευρωνικών Δικτύων, θα μπορούσε να βοηθήσει η παρακάτω σύνοψη της λειτουργίας ενός εγκέφαλου.

Όπως τα μυϊκά κύτταρα είναι το βασικό δομικό συστατικό του μυϊκού μας ιστού, έτσι και τα νευρικά κύτταρα ή νευρώνες είναι το βασικό δομικό συστατικό του νευρικού μας ιστού του οποίου το κέντρο είναι ο εγκέφαλος.

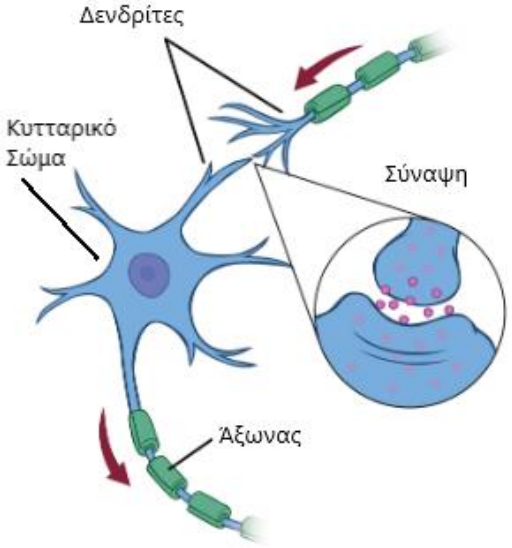
Παρόλο που όλοι οι νευρώνες διαφέρουν σε μέγεθος, σχήμα και χαρακτηριστικά ανάλογα με τη λειτουργία και το ρόλο τους, υπάρχουν τρία βασικά τμήματα ενός νευρώνα που είναι κοινά:

1. Οι δενδρίτες οι οποίοι ονομάστηκαν έτσι λόγω του σχήματός τους (έχουν δενδροειδή μορφή) βρίσκονται στην αρχή ενός νευρώνα και συμβάλλουν στην αύξηση της επιφάνειας του σώματος του κυττάρου. Αυτές οι μικροσκοπικές προεξοχές λαμβάνουν πληροφορίες από άλλους νευρώνες με τους οποίους ενώνονται μέσω συνάψεων και μεταδίδουν ηλεκτρική διέγερση στο κυτταρικό σώμα.

2. Το κυτταρικό σώμα, είναι το μέρος όπου τα σήματα από τους δενδρίτες συνδέονται και μεταφέρονται. Το κυτταρικό σώμα και ο πυρήνας δεν παίζουν ενεργό ρόλο στη μετάδοση του

νευρικού σήματος. Αντ 'αυτού, αυτές οι δύο δομές χρησιμεύουν στη διατήρηση του κυττάρου στη διατήρηση της λειτουργικότητας του νευρώνα.

3. Ο άξονας είναι η επιμήκης ίνα που εκτείνεται από το σώμα του κυττάρου έως το τέρμα του και μεταδίδει το νευρικό σήμα. Όσο μεγαλύτερη είναι η διάμετρος του άξονα, τόσο πιο γρήγορα μεταδίδει πληροφορίες.



Εικόνα 2-10 Το νευρικό σύστημα του ανθρώπου

Ο ανθρώπινος εγκέφαλος αποτελείται από περίπου 100 δισεκατομμύρια νευρώνες (και 10 φορές περισσότερα νευρογλοιακά κύτταρα, ένα άλλο είδος κυττάρων του νευρικού ιστού. Κάτι που ίσως βοήθησε στην διάδοση της δημοφιλής αλλά λανθασμένης άποψης ότι ο άνθρωπος χρησιμοποιεί μόνο το 10% του εγκεφάλου του, όπως αναφέρεται στο άρθρο [27]. Οι νευρώνες αυτοί συνδέονται μεταξύ τους (μέσω δεδριτών και άξονα) σχηματίζοντας ένα **Νευρωνικό Δίκτυο**.

Οι συνδέσεις των νευρώνων δημιουργούνται και τροποποιούνται μέσω μιας βιολογικής εφαρμογής της μεθόδου προσπάθειας και λάθους (trial and error).

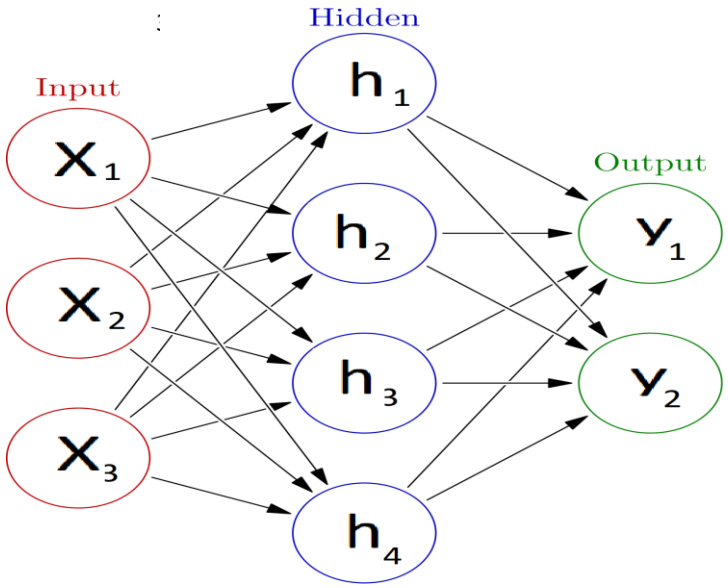
Το νευρικό σύστημα δίνει θετικές ή αρνητικές αναφορές στον εγκέφαλο, σε μορφή θετικών συναισθημάτων όπως η ευχαρίστηση ή αρνητικών όπως ο πόνος. Ο εγκέφαλος τροποποιεί τη δομή του αλλάζοντας τη δομή των συνάψεων μεταξύ των νευρώνων του, ενισχύοντας τις συνάψεις που καταλήγουν σε θετικά συναισθήματα και αποδυναμώνοντας αυτές για τις οποίες έλαβε αρνητικές αναφορές.

2.4.2 Τεχνητά Νευρωνικά Δίκτυα

Τεχνητά Νευρωνικά Δίκτυα αποκαλούμε το είδος αλγορίθμων Μηχανικής Μάθησης οι οποίοι προσπαθούν να μιμηθούν και να μοντελοποιήσουν τη συμπεριφορά του εγκεφάλου με σκοπό να λύσουν προβλήματα με τον τρόπο που θα τα έλυνε ένας εγκέφαλος, μεταδίδοντας δηλαδή σήματα από τον ένα νευρώνα στον άλλο με κάθε νευρώνα να συμψηφίζει τα σήματα των προηγούμενων «αποφασίζοντας» τι σήμα θα πρέπει να μεταδώσει στον επόμενο σε σειρά νευρώνα, καταλήγοντας έτσι σε κάποια αποτελέσματα (outputs). Αυτά τα αποτελέσματα στη συνέχεια

αξιολογούνται βάση των επιθυμητών αποτελεσμάτων που θα ορίσουμε, και η αξιολόγηση αυτή χρησιμοποιείται για την «εκπαίδευση» του δικτύου. Η εκπαίδευση του δικτύου γίνεται ουσιαστικά μέσω της αυξομείωσης των βαρών (της δύναμης δηλαδή των συνάψεων) μεταξύ των νευρώνων με σκοπό για συγκεκριμένα inputs να παίρνει καλύτερα outputs. Να μειώσει δηλαδή το error (διαφορά μεταξύ output και ιδανικού αποτελέσματος).

Στην παρακάτω εικόνα φαίνεται ένα απλό νευρωνικό δίκτυο με ένα κρυφό επίπεδο:



Εικόνα 2-11 Παράδειγμα νευρωνικού δικτύου

Οι νευρώνες X1, X2, X3 ανήκουν στο επίπεδο εισόδου, και είναι αυτοί που δίνουν το αρχικό ερέθισμα στο δίκτυο. Οι h1, h2, h3, h4 είναι νευρώνες κρυφού επιπέδου (hidden layer). Οι νευρώνες των κρυφών επιπέδων έχουν πάντα μια τιμή εισόδου και μία τιμή εξόδου.

Η **τιμή εισόδου** τους αποφασίζεται από το άθροισμα κάθε νευρώνα εισόδου(νευρώνα προηγούμενου επιπέδου) πολλαπλασιαζόμενο με το βάρος της σύνδεσης που ενώνει του τους 2 νευρώνες (W).

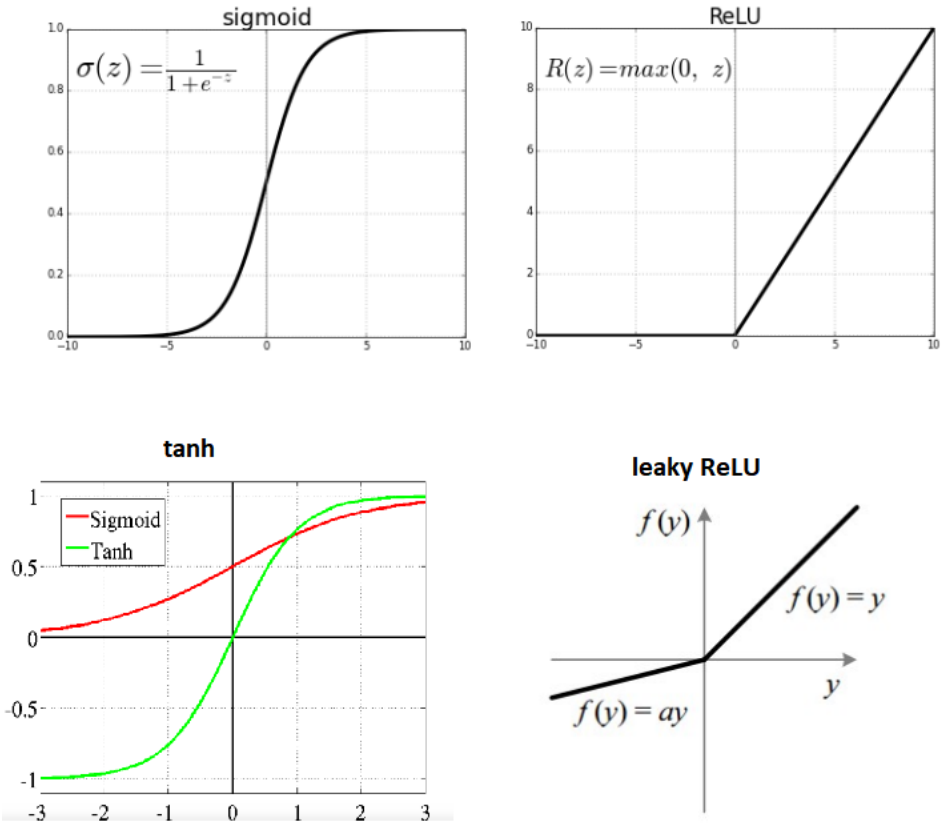
Για παράδειγμα στο παραπάνω νευρωνικό, η τιμή εισόδου του h1 θα υπολογίζονταν ως εξής: $h1(in) = X_1*Wx_{1,h1}+X_2*Wx_{2,h1}+X_3*Wx_{3,h1}$,όπου Wi,j το βάρος της σύνδεσης του νευρώνα i με το νευρώνα j.

Η **τιμή εξόδου** αντίστοιχα είναι το αποτέλεσμα της τιμής εισόδου του νευρώνα, αφού περάσει από μια συνάρτηση ενεργοποίησης (Activation Function A).

Στο παράδειγμα του νευρώνα h1 δηλαδή η τιμή εξόδου του θα υπολογίζονταν ως εξής: $h1_{(out)} = A(h1_{(in)})$.

Συνάρτηση ενεργοποίησης είναι ο μηχανισμός που αποφασίζει αν ο νευρώνας θα στείλει σήμα εξόδου, και αν ναι με τι ισχύ. Στα ΤΝΔ χρησιμοποιούνται διάφορες συναρτήσεις ενεργοποίησης οι οποίες επιλέγονται ανάλογα με την καταλληλόλητα των χαρακτηριστικών τους για το εκάστοτε πρόβλημα.

Κάποια παραδείγματα συναρτήσεων ενεργοποίησης θα μπορούσαν να είναι οι συναρτήσεις sigmoid, ReLU, tanh και leaky ReLU που απεικονίζονται παρακάτω:



Εικόνα 2-12 Συναρτήσεις ενεργοποίησης

Η ιδέα των Τεχνητών Νευρωνικών Δικτύων (Artificial Neural Networks) υπήρχε για αρκετά χρόνια (Πρώτος εκφραστής της θεωρείται ο D.O. Hebb το 1947, Hebbian Learning). Ωστόσο δεν ήταν πολύ δημοφιλής προς τους ερευνητές καθώς τα ANN δεν ήταν ιδιαίτερα αποτελεσματικά στην δημιουργία λύσεων για σύνθετα προβλήματα καθώς χρειάζονταν υπερβολική για την εποχή υπολογιστική ισχύ.

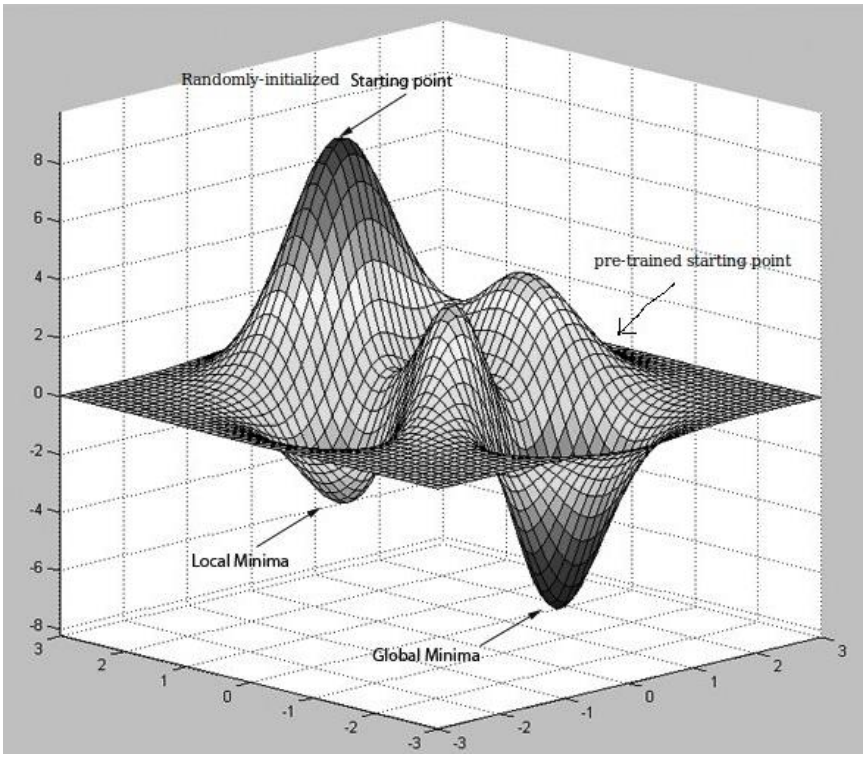
2.4.2.1 Εκπαίδευση Δικτύων

Κάτι που αναζωπύρωσε το ενδιαφέρον για τα νευρωνικά δίκτυα, ήταν ο αλγόριθμος **back-propagation** του Werbos (1975) που λύνει αποτελεσματικά το πρόβλημα κάνοντας εφικτή και αποδοτική την κατάρτιση των δικτύων πολλαπλών επιπέδων (Deep Neural Networks, DNN). Ο αλγόριθμος back-propagation διανείμει τον όρο σφάλματος στα στρώματα του DNN, τροποποιώντας τα βάρη σε κάθε κόμβο μέχρι να φτάσει στην αρχή του δικτύου. Το μεγαλύτερο προτέρημα του είναι ότι χρησιμοποιεί τον κανόνα αλυσίδας, μειώνοντας έτσι κατά πολύ την ποσότητα υπολογισμών που απαιτούνται για την εκπαίδευση του .

Ένας εξίσου σημαντικός αλγόριθμος που βοήθησε τα ANN να φτάσουν στο σημείο που βρίσκονται σήμερα είναι ο Gradient Descent, ο οποίος χρησιμοποιεί μερική παραγώγιση για το σύνολο των μεταβλητών του ANN έτσι ώστε να φτιάξει έναν νοητό πολυδιάστατο χώρο που να καμπυλώνει προς το σημείο ή τα σημεία για τα οποία το λάθος ελαχιστοποιείται. [7]

Έτσι το νευρωνικό δίκτυο κατά την εκμάθηση προσαρμόζει κάθε φορά τα βάρη του προς την κατεύθυνση του στόχου, και όχι τυχαία (brute force). Συνεπώς μειώνει κατά πολύ τον χρόνο που απαιτείται για την εκπαίδευση του ANN.

Στο παρακάτω σχήμα δίνεται ένα παράδειγμα απεικόνισης του χώρου (3^{ov} διαστάσεων σε αυτή τη περίπτωση) λάθους ενός νευρώνα που έχει 3 βάρη ως είσοδο.



Εικόνα 2-13 Χώρος λάθους νευρώνα με 3 βάρη σαν είσοδο (https://stackoverflow.com)

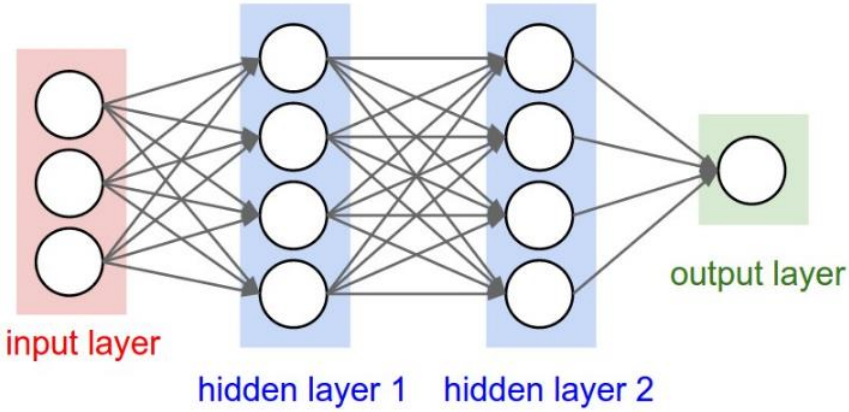
2.4.2.2 Αρχιτεκτονικές Δικτύων

Παρόλο που ένα δίκτυο Νευρώνων μπορεί να κατασκευαστεί με πολλούς τρόπους, ανάλογα με τον τρόπο που οι νευρώνες συνδέονται μεταξύ τους τρείς βασικοί τύποι δικτύων στους οποίους τα κατηγοριοποιούμε είναι οι εξής:

Feed Forward Network:

Λέμε ότι το δίκτυο είναι Feed Forward όταν η πληροφορία (σήμα) ξεκινά από το input layer συνεχίζει στα hidden layer και καταλήγει στο output layer, χωρίς αυτή η διαδρομή να διαγράφει πουθενά κάποιο κύκλο. Τα feed forward δίκτυα είναι αυτά που έχουν χρησιμοποιηθεί περισσότερο μέχρι σήμερα. Αυτό οφείλεται στην σχετική απλότητα στη δομή τους, κάτι που όχι μόνο τα κάνει πιο εύκολα στην ανάλυση και τροποποίησή τους, αλλά και στην εκπαίδευσή τους.

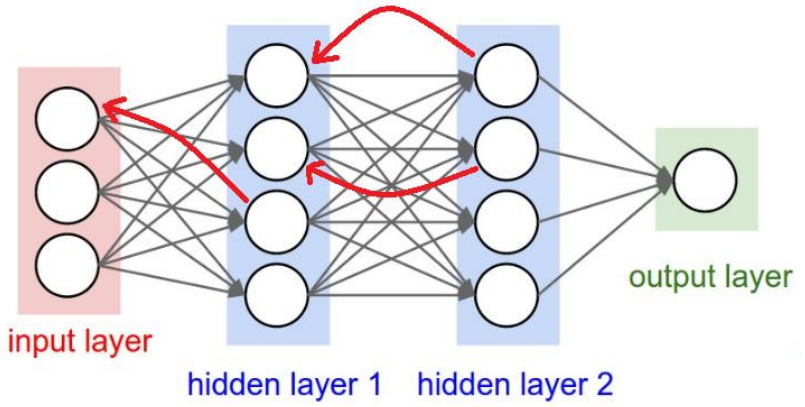
Στην παρακάτω εικόνα φαίνεται η δομή ενός τέτοιου δικτύου με τα βελάκια να υποδηλώνουν την κατεύθυνση της πληροφορίας.



Εικόνα 2-14 Παράδειγμα feed forward NN

Recurrent Network:

Το δίκτυο είναι ανατροφοδοτούμενο (Recurrent) όταν η πληροφορία ξεκινά από το input layer συνεχίζει στα hidden layer και καταλήγει στο output layer, αλλά μέσα σε αυτή τη διαδρομή διαγράφονται κάποιοι κύκλοι, η πληροφορία δηλαδή μπορεί να κατευθυνθεί από τα τελευταία επίπεδα, στα πρώτα σε κάποια σημεία. Παρακάτω, ένα παράδειγμα απλού Recurrent Δικτύου.



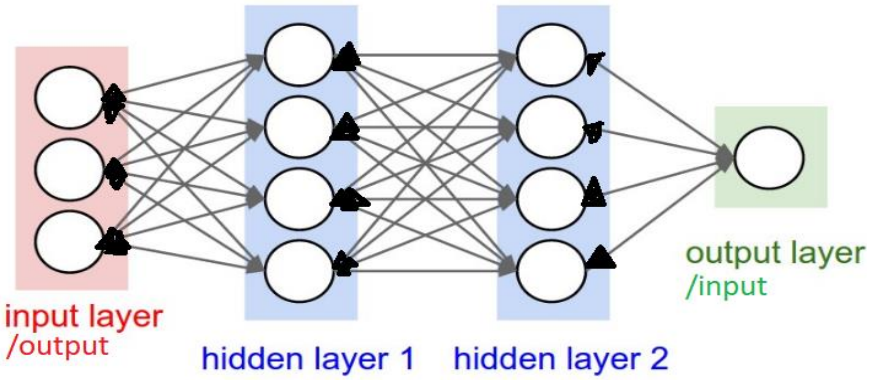
Εικόνα 2-15 Παράδειγμα recurrent NN

Χρησιμοποιούνται ιδιαίτερα όταν απαιτούμε από το δίκτυό μας να διαθέτει ένα είδος μνήμης. Ωστόσο το μειονέκτημά τους είναι ότι μέχρι τώρα η εκπαίδευσή τους θεωρείται ιδιαίτερα δύσκολη. Τα recurrent δίκτυα θεωρείται ότι έχουν τη δυνατότητα να προσομοιώσουν καλύτερα τη δομή και λειτουργία του εγκεφάλου, αφού ο εγκέφαλός μας έχει ανατροφοδοτούμενη (recurrent) δομή.

Ένα υποσύνολο των δικτιών RNN είναι τα δίκτυα LSTM (Long Short Term Memory), τα οποία παρουσιάστηκαν για πρώτη φορά το 1997, και έγιναν ιδιαίτερα δημοφιλή λόγω της λύσης που έδωσαν στο πρόβλημα εξαφανιζόμενης κλίσης [15]

Symmetric (Restricted Boltzman Machine):

Στο συμμετρικό δίκτυο η πληροφορία δεν έχει κατεύθυνση. Το input layer είναι και output, και το output layer είναι και input αντίστοιχα.



Εικόνα 2-16 Παράδειγμα συμμετρικού NN

Τα δίκτυα αυτά δεν χρησιμοποιούνται για προβλέψεις κλάσεων, αλλά για κατηγοριοποιήσεις (clustering). Ανήκουν στην κατηγορία «Unsupervised Learning».

ΚΕΦΑΛΑΙΟ 3 - ΕΠΕΞΕΡΓΑΣΙΑ ΦΥΣΙΚΗΣ ΓΛΩΣΣΑΣ

Ως γνωστών κατά συντριπτική πλειοψηφία η γνώση που έχει στη διάθεση του ο άνθρωπος μέχρι σήμερα βρίσκεται σε μορφή ελεύθερου κειμένου. Μια μορφή η οποία για τον άνθρωπο είναι πολύ γνώριμη και ευκολόχρηστη. Οι περισσότεροι άνθρωποι χρησιμοποιούν πληροφορίες αυτής της μορφής για να μορφωθούν, ενημερωθούν, επικοινωνήσουν, και γενικά να διαχειριστούν τη γνώση.

Ωστόσο για τα μηχανήματα τα οποία έχουν κατακλίσει τις ζωές μας σε διάστημα λίγο περισσότερο από μισό αιώνα μετά την εφεύρεσή τους, αυτό δεν ισχύει. Και αυτό γιατί οι υπολογιστές στη βάση τους είναι μηχανές που επικεντρώνονται στην ακρίβεια και στην ταχύτητα. Ένας υπολογιστής έχει σχεδιαστεί για να εκτελεί συγκεκριμένες αριθμητικές πράξεις με μεγάλη ταχύτητα, γι' αυτό και οι γλώσσες οι οποίες επιτρέπουν στον άνθρωπο να επεξεργαστεί τις λειτουργίες του, είναι αυστηρά δομημένες, και δεν επιτρέπουν λάθη ή παραλείψεις. Αντίθετα η φυσική γλώσσα του ανθρώπου είναι γεμάτη λάθη, πολυσήμαντες εκφράσεις και παραλήψεις ευκόλως εννοούμενων.

Αυτός είναι και ο λόγος για τον οποίο υπάρχει τόσο μεγάλη δυσκολία μοντελοποίησης και επεξεργασίας της υπάρχουσας γνώσης από αλγορίθμους ηλεκτρονικών υπολογιστών.

3.1 Τι είναι Επεξεργασία φυσικής γλώσσας(Natural Language Processing)

Η επεξεργασία φυσικής γλώσσας (NLP) είναι μια πτυχή της Τεχνητής Νοημοσύνης που βοηθά τους υπολογιστές να κατανοήσουν, να ερμηνεύσουν και να χρησιμοποιήσουν τις ανθρώπινες γλώσσες. Το NLP επιτρέπει στους υπολογιστές να επικοινωνούν με ανθρώπους χρησιμοποιώντας ανθρώπινη γλώσσα, επίσης παρέχει στους υπολογιστές τη δυνατότητα ανάγνωσης κειμένου, ακρόασης ομιλίας και ερμηνείας, και γενικά μπορούμε να πούμε ότι ο απώτερος σκοπός του κλάδου είναι να καλύψει το χάσμα μεταξύ ανθρώπου και υπολογιστή. Ωστόσο όπως αναφέραμε και προηγουμένως η διαδικασία «κατανόησης» και χειρισμού της φυσικής γλώσσας από τον υπολογιστή είναι ένα δύσκολο και πολυσύνθετο πρόβλημα.

Αρχικά τα προβλήματα επεξεργασίας φυσικής γλώσσας μπορούν να χωριστούν σε 2 βασικές κατηγορίες. Στα προβλήματα κατανόησης, και στα προβλήματα δημιουργίας φυσικής γλώσσας. Σε αυτή την εργασία η κατηγορία που θα μας απασχολήσει είναι η πρώτη. Τα στάδια για τη λύση ενός προβλήματος επεξεργασίας φυσικής γλώσσας φαίνονται στην παρακάτω εικόνα.

Εικόνα 3-1 Στάδια NLP

3.2 Λεκτική ανάλυση

Σε αυτό το στάδιο ο αρχικός όγκος του κειμένου χωρίζεται σε παραγράφους, οι οποίες χωρίζονται σε προτάσεις, οι οποίες χωρίζονται σε λέξεις (tokenization). Στη συνέχεια συνήθως οι λέξεις και τα σημεία στίξεως χωρίζονται.

3.3 Συντακτική ανάλυση

Ο σκοπός αυτής της φάσης είναι να σχεδιαστεί το συντακτικό νόημα (συντακτικό λεξικό) από το κείμενο. Η συντακτική ανάλυση ελέγχει το κείμενο προσπαθώντας να εξάγει το συντακτικό νόημα του, συμβουλευόμενη τους κανόνες της ανάλογης γραμματικής.

Το αντικείμενο που εκτελεί την συντακτική ανάλυση σε ένα κείμενο ονομάζεται συντακτικός αναλυτής (Parser). Ο Parser είναι υπεύθυνος για την συντακτική ανάλυση ή αλλιώς χαρτογράφηση του κειμένου. Στη συγκεκριμένη εργασία ο Parser που θα χρησιμοποιήσουμε στο δεύτερο κομμάτι είναι από την NLP βιβλιοθήκη “Open Information Extraction” του πανεπιστημίου του Stanford. Παρακάτω φαίνεται ένα παράδειγμα συντακτικής ανάλυσης από τον συγκεκριμένο Parser.

```
>>> parse, = dep_parser.raw_parse(
...     'The quick brown fox jumps over the lazy dog.'
... )
>>> print(parse.to_conll(4))  # doctest: +NORMALIZE_WHITESPACE
The     DT      4       det
quick   JJ      4       amod
brown   JJ      4       amod
fox     NN      5       nsubj
jumps   VBZ     0       ROOT
over    IN      9       case
the     DT      9       det
lazy    JJ      9       amod
dog     NN      5       nmod
.       .       5       punct
```

```
>>> print(parse.tree())  # doctest: +NORMALIZE_WHITESPACE
(jumps (fox The quick brown) (dog over the lazy) .)
```

Εικόνα 3-2 Παράδειγμα Parsing (https://nlp.stanford.edu/)

3.4 Σημασιολογική ανάλυση

Ο σκοπός της σημασιολογικής ανάλυσης είναι να εξαγάγει την ακριβή έννοια ή αλλιώς τη σημασία από το περιεχόμενο του κειμένου.

Σε αντίθεση με την λεκτική ανάλυση η οποία μπορεί επίσης να κάνει κάποια ανάλυση του νοήματος των λέξεων η σημασιολογική ανάλυση επικεντρώνεται σε μεγαλύτερα κομμάτια και προσπαθεί να εξάγει νόημα από τις ομαδοποιήσεις των tokens.

Η σημασιολογική ανάλυση είναι ένα στάδιο του NLP στο οποίο η ποικιλομορφία των διαφόρων προσεγγίσεων ακμάζει. Στην δική μας περίπτωση η μέθοδος που θα χρησιμοποιήσουμε για αυτό το βήμα είναι η μετατροπή των λέξεων από τις οντότητες φυσικής γλώσσας που μας ενδιαφέρουν, σε πυκνά διανύσματα νοήματος λέξεων (Word Embeddings). Η διαδικασία αυτή όπως και άλλες παρόμοιες διαδικασίες διανυσματοποίησης των λέξεων περιγράφονται και μελετώνται ενδελεχώς στην έρευνα [20]

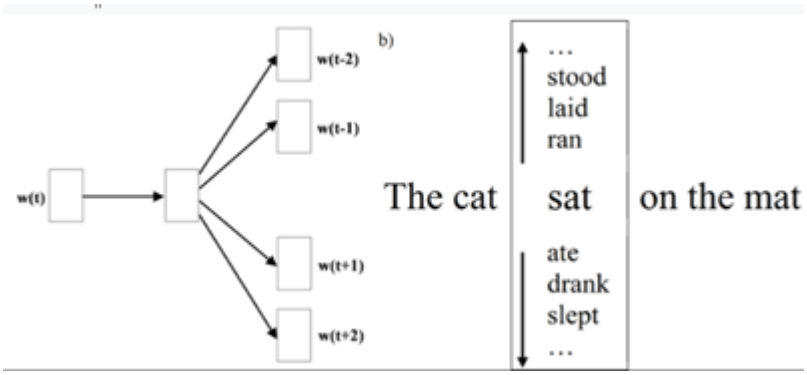
3.4.1 Πυκνά διανύσματα νοήματος λέξεων

Είναι η τεχνική σημασιολογικής ανάλυσης η οποία ορίζοντας έναν πολυδιάστατο διανυσματικό χώρο προσπαθεί να κατατάξει στη συνέχεια κάθε λέξη σε ένα σημείο σχετικό με το νόημά της, προσπαθώντας έτσι να ομαδοποιήσει τις λέξεις με παρόμοιο νόημα και να αποστασιοποιήσει της μη σχετιζόμενες λέξεις μεταξύ τους.

3.4.2 Skip-Gram

Ο αλγόριθμος που επιλέξαμε για να κάνει αυτού του είδους την κατανομή των λέξεων στον πολυδιάστατο διανυσματικό (100 διαστάσεων) χώρο που επιλέξαμε ονομάζεται skip-gram.

Ο Skip-gram χρησιμοποιείται για την πρόβλεψη των πιθανών κοντινών λέξεων για μια δεδομένη λέξη που χρησιμοποιείται ως είσοδος. Ένα επεξηγηματικό σχήμα θα μπορούσε να είναι το παρακάτω.



Εικόνα 3-3 Παράδειγμα λειτουργίας skip-gram (https://towardsdatascience.com)

Ο Skip-gram βασίζει τις προβλέψεις του στα αποτελέσματα ενός νευρωνικού δικτύου με ένα κρυφό επίπεδο το οποίο εκπαιδεύει τα βάρη του ανάλογα με τον σκοπό που περιγράψαμε προηγουμένως.

3.5 Γιατί η μελέτη της κίνησης των μετοχών ταιριάζει στη μηχανική μάθηση

Οι χρηματιστηριακές μετοχές είναι ένα αντικείμενο στο οποίο η μηχανική μάθηση όχι μόνο έχει εφαρμογή, άλλα κερδίζει συνεχώς έδαφος σε σχέση με τα παραδοσιακά πληροφοριακά συστήματα. Και αυτό γιατί η ευημερία των χρηματιστηριακών αγορών εξαρτάται από την ορθή πρόβλεψη της κίνησης της αγοράς. Κάτι το οποίο απαιτεί την εξαγωγή γνώσης μέσα από έναν ιδιαίτερα μεγάλο όγκο δεδομένων τον οποίο έχουν στη διάθεσή τους οι χρηματιστές. Το γεγονός αυτό καθιστά τα συστήματα μηχανικής μάθησης ιδανικά εργαλεία καθώς η εξαγωγή χρήσιμης πληροφορίας από μεγάλο όγκο δεδομένων ανήκει στην κατηγορία προβλημάτων στην οποία αυτά τα συστήματα χρησιμοποιούνται κατά κόρον.

Η συμπεριφορά της χρηματιστηριακής αγοράς μπορεί να θεωρηθεί ως μια συνάρτηση πολλαπλών μεταβλητών που τα αποτελέσματά της διατιμώνται από:

1. Οικονομικές μεταβλητές, όπως είναι τα επιτόκια, συναλλαγματικές ισοτιμίες, τιμές βασικών εμπορευμάτων κλπ.
2. Μεταβλητές για τη βιομηχανία, όπως οι ρυθμοί ανάπτυξης της βιομηχανικής παραγωγής και των τιμών καταναλωτή
3. Συγκεκριμένες μεταβλητές της εταιρείας, όπως αλλαγές στις πολιτικές της εταιρείας, καταστάσεις αποτελεσμάτων και μερισματικές αποδόσεις
4. Ψυχολογικές μεταβλητές των επενδυτών, όπως οι προσδοκίες των επενδυτών και οι επιλογές των θεσμικών επενδυτών
5. Πολιτικές μεταβλητές, όπως η εμφάνιση και η απελευθέρωση σημαντικών πολιτικών γεγονότων [28]

Η προβλεψιμότητα των μετοχών έχει μελετηθεί από πολλούς ερευνητές. Ενώ το επίπεδο προβλεψιμότητας μπορεί να διαφέρει μεταξύ των ερευνών, το γεγονός ότι η συμπεριφορά της χρηματιστηριακής αγοράς είναι σε κάποιο επίπεδο προβλέψιμή, αποτελεί κοινή αποδοχή. Οι πιο συνηθισμένες μέθοδοι πρόβλεψης απόδοσης μετοχών είναι οικονομετρικές ή στατιστικές μέθοδοι που βασίζονται στην ανάλυση προηγούμενων κινήσεων της αγοράς [9]

ΚΕΦΑΛΑΙΟ 4 - ΣΧΕΤΙΚΕΣ ΕΡΕΥΝΕΣ

4.1 Εισαγωγή

Πολλοί ερευνητές έχουν προσεγγίσει το θέμα της ανάλυσης και πρόβλεψης της κίνησης των μετοχών μέσω μεθόδων Μηχανικής Μάθησης με ποικίλους τρόπους. Αν ορίζαμε ως σύστημά μας, ένα σύστημα το οποίο να περιέχει μόνο αριθμητικές οντότητες της χρηματιστηριακής αγοράς (τιμές μετοχών, όγκος κεφαλαιοποιήσεων, δείκτες κτλ.)

Μπορούμε να χωρίσουμε τις έρευνες αυτού του αντικειμένου σε 3 βασικές κατηγορίες:

1) Έρευνες βασιζόμενες σε δεδομένα ενδογενών παραγόντων: θεωρούμε αυτές που μελετούν συστήματα πρόβλεψης που χειρίζονται πληροφορία που πηγάζει μόνο από το εσωτερικό της αγοράς, ιστορικές τιμές μετοχών, δείκτες των αγορών, στατιστικοί δείκτες (κινούμενοι μέσοι όροι, παλινδρομήσεις των τιμών, εποχικότητα κτλ.).

2) Έρευνες βασιζόμενες σε δεδομένα εξωγενών παραγόντων: είναι αυτές οι οποίες μελετούν συστήματα πρόβλεψης που βασίζονται σε πληροφορίες όπως απόψεις ατόμων για την αγορά, γεγονότα ειδήσεων, αναλύσεις οικονομολόγων, κοινή γνώμη της μάζας από κοινωνικά δίκτυα και άλλες πληροφορίες τέτοιου χαρακτήρα.

3) Συνδυασμός των 2 παραπάνω κατηγοριών.

Εικόνα 4-1 Σχήμα επεξήγησης του συστήματος μελέτης μας

4.2 Έρευνες βασιζόμενες σε δεδομένα ενδογενών παραγόντων

Authors (Year)	Data type (Num. of input features × lagged times)	Target output	Num. of samples (Training: Validation: Test)	Sampling period (Frequency)	Method	Performance measure
Enke and Mehdiyev (2013)	US S&P 500 index (20 × 1)	Stock price	361	Jan-1980 to Jan-2010 (daily)	Feature selection+fuzzy clustering+fuzzy NN	RMSE
Niaki and Hoseinzade (2013)	Korea KOSPI200 index (27 × 1)	Market direction (up or down)	3650 (8:1:1)	1-Mar-1994 to 30-Jun-2008 (daily)	Feature selection+ANN	statistical tests
Cervelló-Royo et al. (2015)	US Dow Jones index (1 × 10)	Market trend (bull/bear-flag)	91,307	22-May-2000 to 29-Nov-2013 (15-min)	Template matching	trading simulation
Patel, Shah, Thakkar, and Kotecha (2015)	India CNX and BSE indices (10 × 1)	Stock price	2393*	Jan-2003 to Dec-2012 (daily)	SVR+ {ANN, RF, SVR}	MAPE, MAE, rRMSE, MSE
T.-l. Chen and Chen (2016)	Taiwan TAIEX[a] and US NASDAQ[b] indices (27 × 20)	Market trend (bull-flag)	3818[a.] * 3412[b.] * (7:0:1)	7-Jan-1989 to 24-Mar-2004 (daily)	Dimension reduction+template matching	trading simulation
Chiang, Enke, Wu, and Wang (2016)	World 22 stock market indices ({3~5} × 1)	Trading signal (stock price)	756 (2:0:1)	Jan-2008 to Dec-2010 (daily)	Particle swarm optimization +ANN	trading simulation
Chourmouziadis and Chatzoglou (2016)	Greece ASE general index (8 × 1)	Portfolio composition (cash:stock)	3907*	15-Nov-1996 to 5-Jun-2012 (daily)	Fuzzy system	trading simulation
Qiu, Song, and Akagi (2016)	Japan Nikkei 225 index (71 × 1)	Stock return	237 (7:0:3)	Nov-1993 to Jul-2013 (monthly)	ANN+{genetic algorithm, simulated annealing}	MSE
Arévalo, Niño, Hernández, and Sandoval (2016)	US Apple stock (3× {2~15}+2)	Stock price	19,109 (17:0:3)	2-Sep-2008 to 7-Nov-2008 (1-minute)	Deep NN	MSE, directional accuracy
Zhong and Enke (2017)	US SPDR S&P 500 ETF (SPY) (60 × 1)	Market direction (up or down)	2518 (14:3:3)	1-Jun-2003 to 31-May-2013 (daily)	Dimension reduction+ANN	trading simulation, statistical tests
(Eunsuk Chong 2017)	Korea KOSPI 38 stock returns (38 × 10)	Stock return	73,041 (3:1:1)	4-Jan-2010 to 30-Dec-2014 (5-minute)	Data representation+deep NN	NMSE, RMSE, MAE, MI
(Thomas Fischer 2017)	**US S&P 500 index**	Stock return	~23.000(8:2)	1992 to 2015 (in time windows of 1000 days)	LSTM Networks	Trading simulation, statistical tests
Our research	**(US S&P 500 index + 15 stock) x 4**	Market direction (up or down) & Stock return	1783 * 4 (8:2)	October 2006 to November 2013 (daily)	Data representation+deep NN	NMSE, RMSE, MAE, MI + Accuracy(Up/Down%)

Εικόνα 4-2 Έρευνες ενδογενών παραγόντων

Ο πίνακας 1 μας δίνει συμπυκνωμένα δεδομένα πρόσφατων ερευνών για συστήματα πρόβλεψης χρηματιστηριακής αγοράς που βασίζονται σε ενδογενείς παράγοντες. Οι έρευνες που αναφέρονται είναι οι [31], [32], [33], [34], [22], [10], [11], [23], [19].

4.2.1 Η έρευνα «Forecasting daily stock market return using dimensionality reduction» [23]

Στόχος αυτής της έρευνας είναι να παρουσιάσει μια αποδοτική διαδικασία μηχανικής μάθησης για την πρόβλεψη της ημερήσιας κατεύθυνσης μετοχών της πλατφόρμας S&P 500. Η μέθοδος στη οποία εστιάζει η έρευνα είναι η «μείωση διαστάσεων» (Dimensionality Reduction).

Τι είναι η μείωση διαστάσεων;

Στα προβλήματα ταξινόμησης μηχανικής μάθησης, υπάρχουν συχνά πάρα πολλοί παράγοντες βάσει των οποίων γίνεται η τελική ταξινόμηση. Αυτοί οι παράγοντες είναι μεταβλητές που ονομάζονται χαρακτηριστικά (features). Όσο μεγαλύτερος είναι ο αριθμός των features, τόσο πιο δύσκολα γίνεται η απεικόνιση του training set και κατά συνέπεια η εργασία σε αυτό. Μερικές φορές, τα περισσότερα από αυτά τα χαρακτηριστικά συσχετίζονται και επομένως είναι περιττά. Σε αυτό το σημείο οι αλγόριθμοι μείωσης των διαστάσεων βρίσκουν εφαρμογή. Η μείωση των

διαστάσεων είναι η διαδικασία μείωσης του αριθμού των τυχαίων μεταβλητών που εξετάζονται, επιτυγχάνοντας ένα σύνολο κύριων μεταβλητών. Μπορεί να χωριστεί σε επιλογή features (όταν από το αρχικό σύνολο των features επιλέγουμε ένα υποσύνολο τους) και εξαγωγή features (όταν από το αρχικό σύνολο features καταλήγουμε σε ένα σύνολο εντελώς καινούριων features).

Η εμπειρική αυτή έρευνα μελετά την εφαρμογή τριών αλγορίθμων της κατηγορίας «Dimensionality Reduction» οι οποίοι είναι:

1) Principal Component Analysis (PCA)
2) Fuzzy Robust Principal Component Analysis(FRPCA)
3) Kernel-based Principal Component Analysis(KPCA)

Στην αρχή του κειμένου αναφέρονται τα δημοφιλέστερα εργαλεία για την πρόβλεψη της κίνησης της αγοράς. Υποστηρίζοντας ότι υπάρχουν 2 βασικές κατηγορίες στις οποίες μπορούν να χωριστούν οι τρόποι που χρησιμοποιούνται για την πρόβλεψη οικονομικών χρονοσειρών, ο univariate και ο multivariate.

Στη Univariate ανάλυση, σαν είσοδος χρησιμοποιείται μόνο η ίδια η χρονοσειρά ενώ στη Multivariate ανάλυση, οι μεταβλητές εισόδου μπορούν να είναι και διάφοροι άλλοι τύποι δεδομένων. Αναφέρει διάφορα εργαλεία αρχίζοντας από κάποια στατιστικής προσέγγισης, όπως το Autoregressive Moving Average(ARMA), ή το Autoregressive Integrated Moving Average (ARIMA), και καταλήγοντας στα εργαλεία μηχανικής μάθησης τα οποία ανήκουν στην κατηγορία multivariate, δίνοντας έμφαση στα Νευρωνικά Δίκτυα τα οποία όπως υποστηρίζεται είναι ιδιαίτερα δημοφιλή.

Σε αυτή την έρευνα χρησιμοποιούνται Ιστορικά δεδομένα της πλατφόρμας S&P 500. Συγκεκριμένα ως output χρησιμοποιείται η τιμή κλεισίματος της μετοχής του S&P 500(SPY), μαζί με ένα σύνολο 60 οικονομικών παραγόντων, που χρησιμοποιούνται ως πιθανά features. Χρησιμοποιούνται δεδομένα από συνολικά 2518 ημέρες, από τον Ιούνιο του 2003 ως τον Μάιο του 2013.

Αρχικά υπάρχει μία προ επεξεργασία των δεδομένων. Τα δεδομένα εξομαλύνονται αφαιρούνται πιθανοί «outliers» και γεμίζονται τα κενά υπάρχουν στα features με αναμενόμενες τιμές. Αφού τα δεδομένα εξομαλυνθούν και κανονικοποιηθούν, υπάγονται σε «Dimensionality Reduction» μέσω μιας από τις τρείς μεθόδους που αναφέρθηκαν αρχικά.

Μετά από αυτή τη διαδικασία, τα «μειωμένα» features χρησιμοποιούνται σε input νευρώνες ενός Νευρωνικού δικτύου με ένα «hidden layer» αποτελούμενο από 10 νευρώνες, και ένα «output layer» με 2 νευρώνες οι οποίοι αντιπροσωπεύουν της κλάσεις «Πάνω» και «Κάτω» ανάλογα με την κατεύθυνση της τιμής κλεισίματος την επόμενη μέρα.

Εφαρμόζοντας PCA στα δεδομένα ο αριθμός των features μειώνεται από 60 σε 37 features το καθένα από τα οποία αποτελεί έναν γραμμικό συνδυασμό των 60, εξηγώντας και πάλι σχεδόν το 100% του dataset.

Η ίδια διαδικασία επαναλαμβάνεται και με την εφαρμογή των αλγορίθμων FRPCA & KPCA. Μετά την χρήση των τριών μεθόδων μείωσης διαστάσεων και την εκπαίδευση του Νευρωνικού Δικτύου για κάθε μία περίπτωση αντίστοιχα, τα αποτελέσματα που προέκυψαν μετρήθηκαν βάση του ποσοστού της σωστής πρόβλεψης της κατεύθυνσης της τιμής κλεισίματος την επόμενη μέρα.

Το συμπέρασμα ήταν ότι παρόλο που η διαφορά ήταν μικρή, η μέθοδος PCA επικράτησε ελαφρά σε ακρίβεια σε σχέση με τις άλλες 2. Τα αποτελέσματα από τα ποσοστά ορθής πρόβλεψης σε 4 διαφορετικά σύνολα δεδομένων (training,validation,testing & total) φαίνονται στον παρακάτω πίνακα:

Table 3

The *ANN* classification results of the 36 transformed data sets based on three *PCAs*

PCs	PCA				FPCA				KPCA			
	training	validation	testing	total	training	validation	testing	total	training	validation	testing	total
1	54.8	53.6	56.8	54.9	54.8	53.3	57	54.9	55.3	53.3	57	55.2
3	55.2	53.3	57.3	55.2	55.2	53.8	56.8	55.2	55.8	53.6	57	55.6
6	54.9	53.6	57.3	55	57.1	53.6	57	56.6	55.6	53.3	57	55.5
10	56.4	54.6	57.3	56.3	57.1	56.5	56.8	57	56.7	54.6	58.1	56.6
15	56.3	53.3	57.6	56	55.3	55.4	57.8	55.7	56	54.9	57.6	56
22	55.2	54.6	58.1	55.5	56.2	54.9	57.8	56.2	56.6	56	57.8	56.7
26	55.1	53.1	58.1	55.2	56.8	56.5	58.6	57	55.4	54.1	57.8	55.6
31	57.5	57.3	58.1	57.5	56.2	54.4	59.2	56.4	55.7	54.1	57.3	55.7
34	56.2	56	57.3	56.4	56	53.8	58.1	56	55.5	54.4	56.8	55.5
37	55	54.4	57	55.2	56.3	54.1	57.8	56.2	55.7	53.1	57.3	57.6
40	56.2	56.2	56.2	56.2	56	54.1	57.8	56	55.8	59.2	57.6	56.6
60	57.5	54.1	58.1	57.1	56.5	54.4	57.3	56.3	57.4	54.9	58.4	57.1

Εικόνα 4-3 Αποτελέσματα έρευνας [23]

Στο τέλος της έρευνας παρουσιάστηκε μία προσομοίωση ανταλλαγής μετοχών κατά την οποία διαπιστώθηκε ότι μια στρατηγική βασισμένη στη μέθοδο ANN-PCA έχει σημαντικά μεγαλύτερα «risk-adjusted» κέρδη, από μία στρατηγική αγοράς Αμερικανικών κρατικών ομολόγων ωρίμανσης ενός μηνός.

4.2.2 Η έρευνα «Deep Learning networks for stock market analysis and prediction: Methodology, Data representations, and case studies» [5]

Η έρευνα αυτή μελετά την χρήση Αλγορίθμων Βαθιάς εκμάθησης για την ανάλυση και πρόβλεψη της κίνησης της χρηματιστηριακής αγοράς. Εξετάζει την απόδοση των αλγορίθμων αυτών δίνοντας μεγάλη βαρύτητα στον τρόπο αναπαράστασης των δεδομένων, καθώς όπως αναφέρεται στο άρθρο η απόδοση ενός αλγορίθμου βαθιάς εκμάθησης εξαρτάται σε μεγάλο βαθμό από την αναπαράσταση του data set. Στην εμπειρική έρευνα αυτή για την αναπαράσταση των δεδομένων χρησιμοποιούνται τέσσερεις τρόποι, εκτός από τη «μη επεξεργασία» τους (raw data), εφαρμόζονται τρείς «unsupervised» μέθοδοι εξαγωγής features (Principal Component Analysis, Auto-Encoders & restricted Boltzman Machine) τις οποίες θα αναλύσουμε παρακάτω. Τα δεδομένα της έρευνας αποτελούνται από τις τιμές μετοχών της αγοράς «Korea KOPSI» ανά 5 λεπτά στο διάστημα 4/1/2010 ως 30/12/2014 . Επίσης εξετάζεται και η επιρροή στην απόδοση που μπορεί να έχει ο συνδυασμός στατιστικών μεθόδων (Αυτοπαλινδρόμηση) με τα νευρωνικά δίκτυα.

Μέθοδοι Εξαγωγής Features:

1) Principal Component Analysis (PCA)

είναι μια τεχνική που χρησιμοποιείται για την ταυτοποίηση ενός μικρότερου αριθμού μη συσχετισμένων μεταβλητών γνωστών ως κύριων συνιστωσών από ένα μεγαλύτερο σύνολο δεδομένων.[16]

2) Autoencoders

Ένας αυτόματος κωδικοποιητής έχει τρία βασικά μέρη: έναν κωδικοποιητή, έναν κωδικό και έναν αποκωδικοποιητή. Τα αρχικά δεδομένα πηγαίνουν σε ένα κωδικοποιημένο αποτέλεσμα και τα επόμενα στρώματα του δικτύου το επεκτείνουν σε τελική έξοδο [2]

3) Restricted Boltzman Machine

Η περιορισμένη μηχανή Boltzman ονομάζεται περιορισμένη λόγω της έλλειψης επικοινωνίας μεταξύ των στρωμάτων (layers) του μοντέλου.[8]

Δεδομένα:

Για την έρευνα αυτή χρησιμοποιούνται οι τιμές 38 μετοχών της «Korea KOPSI market» ανά 5 λεπτά την περίοδο 4/1/2010 ως 30/12/2014. Τα δεδομένα δηλαδή αποτελούνται από 73,041 τιμές για κάθε μία από τις 38 μετοχές (εξαιρούνται οι πρώτες 10 τιμές καθώς τις χρειαζόμαστε για να κατασκευάσουμε το πρώτο input κάθε μέρας (η αρχική δομή των input δεδομένων είναι οι δέκα προηγούμενες τιμές κάθε μετοχής, το input δηλαδή του νευρωνικού δικτύου χωρίς την εφαρμογή μεθόδων εξαγωγής features είναι 38*10 δηλαδή 380 τιμές εισόδου).

Μεθοδολογία:

Αρχικά τα δεδομένα μεταμορφώνονται μέσω της εφαρμογής μιας από τις μεθόδους εξαγωγής features (αυτό το βήμα παραλείπεται στην περίπτωση των μη-επεξεργασμένων δεδομένων) σε ένα καινούριο Data Set με αριθμό χαρακτηριστικών ανάλογο με το output της μεθόδου.

Στη συνέχεια όλα τα features κανονικοποιούνται και τα δεδομένα χωρίζονται σε training (80% των δεδομένων) και test set (20% των δεδομένων). Επιπλέον το τελευταίο 20% του training set χωρίζεται σε validation set για λόγους αποφυγής overfitting.

Πριν την εφαρμογή του Νευρωνικού Δικτύου στα διάφορα set αναπαράστασης που έχουν δημιουργηθεί, οι ερευνητές χρησιμοποιούν διάφορες μεθόδους για την απόσπαση πληροφοριών σχετικά με την προβλεπτική δύναμη που περιέχεται στα δεδομένα. Για παράδειγμα χρησιμοποιούν τη στατιστική μέθοδο «logistic regression» με σκοπό να κάνουν κάποιες απλές προβλέψεις για την κατεύθυνση της τιμής της μετοχής στο επόμενο χρονικό βήμα, και να μπορέσουν στη συνέχεια να έχουν ένα επιπλέον μέτρο σύγκρισης για τα μοντέλα της έρευνάς τους.

Νευρωνικό Δίκτυο:

Το νευρωνικό δίκτυο που χρησιμοποιείται στην έρευνα αποτελείται από ένα input layer, με αριθμό νευρώνων ανάλογο με τον αριθμό των features της αναπαράστασης των δεδομένων, δύο κρυμμένα layers και ένα output layer με νευρώνες που αντιπροσωπεύουν τις κλάσεις πάνω ή κάτω (την κατεύθυνση δηλαδή της κάθε μετοχής στο επόμενο χρονικά βήμα).

Η συνάρτηση ενεργοποίησης που χρησιμοποιείται είναι η «ReLU» καθώς προσφέρει γρηγορότερη εκμάθηση σε σχέση με την «sigmoid» χωρίς να επηρεάζεται η απόδοση του δικτύου σύμφωνα με τους [21].

Αξιολόγηση απόδοσης:

Για την αξιολόγηση των αποτελεσμάτων χρησιμοποιούνται τα «metrics» NMSE (Normalized Mean Squared Error), RMSE (Rooted Mean Squared Error),MAE (Mean Absolute Error).

Χρησιμοποιώντας τα παραπάνω «metrics» συγκρίνονται τα αποτελέσματα τον 5 αναπαραστάσεων των δεδομένων, τα αποτελέσματα ενός δικτύου με 2 hidden layer, και τα αποτελέσματα ενός απλού μοντέλου αυτοπαλινδρόμησης(AR(10)).

Τα αποτελέσματα για το μέτρο NMSE φαίνονται στον παρακάτω πίνακα:

Table 5
Normalized mean squared error in the training set (04-Jan-2010~24-Dec-2013).

Stock ID	AR(10)	ANN (RawData)	DNN (RawData)	DNN (PCA380)	DNN (RBM400)	DNN (AE400)
1	0.9810	1.0000	0.9246	0.9336	0.9401	0.9241
2	0.9746	0.9999	0.9314	0.9370	0.9453	0.9343
3	0.9908	0.9985	0.9379	0.9444	0.9479	0.9429
4	0.9816	1.0000	0.9234	0.9244	0.9304	0.9224
5	0.9817	1.0000	0.9431	0.9511	0.9546	0.9452
6	0.9396	0.9674	0.9436	0.9481	0.9490	0.9528
7	0.9288	0.9798	0.8854	0.8952	0.9122	0.8849
8	0.9618	1.0000	0.9198	0.9189	0.9228	0.9184
9	0.9750	1.0000	0.9671	0.9654	0.9661	0.9622
10	0.9164	0.9797	0.8537	0.8669	0.8699	0.8560
11	0.9322	0.9714	0.9389	0.9394	0.9458	0.9400
12	0.9816	0.9999	0.9010	0.9095	0.9165	0.9076
13	0.9771	0.9999	0.8975	0.9061	0.9143	0.8988
14	0.9929	1.0000	0.9644	0.9655	0.9676	0.9672
15	0.9902	1.0000	0.9371	0.9430	0.9487	0.9386
16	0.9492	0.9667	0.9489	0.9550	0.9621	0.9514
17	0.9837	0.9971	0.9218	0.9273	0.9269	0.9240
18	0.9685	1.0000	0.9142	0.9226	0.9297	0.9161
19	0.9571	1.0000	0.9440	0.9455	0.9444	0.9460
20	0.8747	0.9745	0.9089	0.9166	0.9160	0.9121
21	0.9871	1.0000	0.9635	0.9675	0.9683	0.9662
22	0.8737	0.9787	0.7496	0.7539	0.7832	0.7497
23	0.9695	1.0000	0.9545	0.9553	0.9570	0.9559
24	0.9681	1.0000	0.9427	0.9410	0.9432	0.9392
25	0.9645	0.9631	0.9584	0.9613	0.9612	0.9562
26	0.9936	0.9809	0.9670	0.9673	0.9664	0.9687
27	0.9887	1.0000	0.9344	0.9413	0.9463	0.9384
28	0.9444	0.9785	0.8781	0.8800	0.8856	0.8777
29	0.9559	0.9998	0.9546	0.9555	0.9573	0.9528
30	0.9731	1.0001	0.9197	0.9216	0.9244	0.9206
31	0.9880	1.0000	0.9450	0.9512	0.9551	0.9466
32	0.9737	1.0000	0.9473	0.9504	0.9558	0.9468
33	0.9376	0.9706	0.9347	0.9432	0.9462	0.9384
34	0.9539	0.9799	0.9133	0.9191	0.9223	0.9162
35	0.9841	0.9999	0.9448	0.9497	0.9548	0.9482
36	0.9848	1.0008	0.9675	0.9688	0.9693	0.9676
37	0.9281	0.9785	0.8148	0.8159	0.8429	0.8107
38	0.9720	1.0000	0.9293	0.9306	0.9406	0.9278
Average	0.9626	0.9912	0.9244	0.9287	0.9340	0.9256

Εικόνα 4-4 Αποτελέσματα έρευνας [5]

Τα συμπεράσματα που εξάγονται είναι ότι το Dnn είναι αποδοτικότερο από το AR(10) για οποιανδήποτε αναπαράσταση των δεδομένων. Με την αναπαράσταση των Raw Data να έχει τα καλύτερα αποτελέσματα.

Στη συνέχεια εξετάζεται η εφαρμογή συνδυασμού των μεθόδων AR(10) και DNN, και συμπεραίνεται ότι συνδυάζοντας το AR(10) με το DNN, χρησιμοποιώντας τα υπολειπόμενα του AR(10) ως input για το DNN, μπορούμε να βελτιώσουμε ελάχιστα την απόδοση της πρόβλεψης.

Στον παρακάτω πίνακα φαίνεται το μέσο NMSE για κάθε μέθοδο που εξετάστηκε από την έρευνα:

Method	Data representation for DNN			
	(RawData)	(PCA380)	(RBM400)	(AE400)
AR(10)	0.9655			
ANN	0.9937	0.9990	0.9982	0.9976
DNN	0.9629	0.9660	0.9702	0.9638
AR-DNN	0.9622	0.9625	0.9628	0.9621
	(−0.0033)	(−0.0030)	(−0.0027)	(−0.0034)
DNN-AR	0.9643	0.9650	0.9682	0.9648
	(0.0013)	(−0.0010)	(−0.0020)	(0.0010)

Εικόνα 4-5 Συνοπτικά αποτελέσματα [5]

Το άρθρο αυτό τελειώνοντας επισημαίνει ότι σε μελλοντικές έρευνες θα μπορούσε να μελετηθεί το κατά πόσο η εισαγωγή επιπλέον features που είναι γνωστό ότι περιέχουν πληροφορία για την μελλοντική τιμή της μετοχής.

4.2.3 Η Έρευνα «Deep learning with long short-term memory networks for financial market predictions» [19]

Το αντικείμενο αυτού του άρθρου είναι η ανάλυση της απόδοσης των «LTSM Neural Nets» σε θέματα ανάλυσης και πρόβλεψης της κατεύθυνσης χρηματιστηριακών μετοχών.

Τα LSTM(long Short Term Memory) δίκτυα είναι ένα ειδικό είδος RNN, ικανό να μαθαίνει μακροπρόθεσμες εξαρτήσεις. Αναφέρθηκαν για πρώτη φορά στο άρθρο [18].

Στην περίληψη του άρθρου επίσης αναφέρεται πως παρόλο που τα συγκεκριμένα δίκτυα ενδείκνυνται για την εκμάθηση ακολουθιών, παραδόξως δεν χρησιμοποιούνται τόσο συχνά σε προβλήματα πρόβλεψης οικονομικών χρονοσειρών.

Η έρευνα αυτή εφαρμόζει ένα δίκτυο LSTM στα δεδομένα από την πλατφόρμα S&P 500 από την αρχή των δεδομένων διαθέσιμων στη πλατφόρμα (1992), ως και το 2015. Η χρήση του συγκεκριμένου Data Set γίνεται εκτός των άλλων και για λόγους συμβατότητας με την έρευνα[4] η οποία όπως αναφέραμε προηγουμένως μελετά την απόδοση των αλγορίθμων «Random Forest», «Deep Neural Nets» και «Logistic Regression» στο συγκεκριμένο Data Set.

Παρατηρείται ότι το LSTM δίκτυο που εξετάζει η έρευνα, αποδίδει καλύτερα από τους συγκρινόμενους αλγόριθμους της έρευνας [4] με επιστροφή 0.47 ανά ημέρα κατά την περίοδο 1992-2009, αλλά από την χρονιά 2010 και έπειτα το πλεονέκτημά του χάνεται και καταλήγει σε μηδενικό κέρδος μετά την εφαρμογή των κοστών συναλλαγής.

Μεθοδολογία:

Στο άρθρο αυτό, η μεθοδολογία που περιγράφεται είναι περιληπτικά η παρακάτω:

Αρχικά κατασκευάζονται οι ακολουθίες που απαιτούνται για το training του LSTM δικτύου ως εξής:

Οι 23.000 περίπου ημερήσιες τιμές κάθε μετοχής χωρίζονται σε 23 περιόδους μελέτης με 1000 ημερήσιες τιμές σε κάθε περίοδο, στη συνέχεια κάθε περίοδος αφού κανονικοποιηθεί χωρίζεται σε training και testing data, και για κάθε σετ φτιάχνονται οι ακολουθίες 240 κινούμενων ημερών για κάθε μετοχή οι οποίες εισάγονται ως input στο LSTM δίκτυο το οποίο καταλήγει σε 2 νευρώνες εξόδου μέσω των οποίων εξάγει 2 πιθανότητες ως αποτέλεσμα, την πιθανότητα η τιμή της μετοχής για την επόμενη μέρα να είναι μεγαλύτερη από την προηγούμενη (κλάση 1) ή την πιθανότητα να είναι μικρότερη απ' τη προηγούμενη (κλάση 2).

Εικόνα από [19]

Εικόνα 4-6 Επεξήγηση προ-επεξεργασίας δεδομένων [19]

Το δίκτυο που χρησιμοποιήθηκε περιγράφεται όπως αναφέρεται και στα άρθρα [29], [30]

LSTM δίκτυο

Αποτελείται από ένα input layer , ένα ή περισσότερα hidden layer και ένα output layer. Το input layer αποτελείται από έναν νευρώνα στον οποίο εισάγεται η τιμή της μετοχής για 240 χρονικά βήματα που υπάρχουν στην ακολουθία.

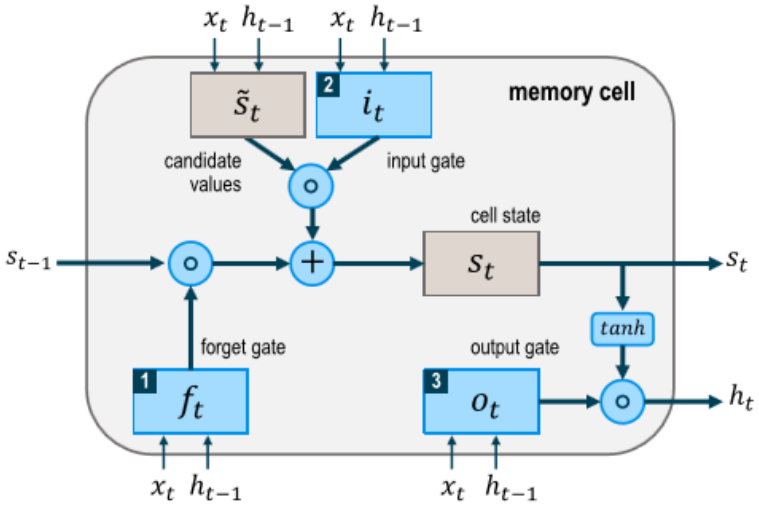
Το output layer αποτελείται από αριθμό νευρώνων ίσο με τις κλάσεις που θέλουμε να μπορεί να αντιπροσωπεύσει το αποτέλεσμα, δηλαδή 2 (πάνω, αν η μετοχή θα έχει ανοδική πορεία, ή κάτω αν η μετοχή θα έχει καθοδική πορεία).

Το hidden layer αποτελείται από 25 memory cells τα οποία περιγράφονται στο άρθρο ως εξής:

Το κάθε memory cell αποτελείται από 3 πύλες (πύλη input, πύλη output, πύλη forget), οι οποίες είναι υπεύθυνες για την κατάσταση του. Κάθε χρονική στιγμή t κάθε μία από αυτές τις πύλες παίρνει σαν είσοδο ένα στοιχείο της ακολουθίας των 240 στοιχείων του input διανύσματος, καθώς και την τιμή output του cell για την προηγούμενη χρονική στιγμή t-1.

- Η **πύλη input** είναι αυτή που αποφασίζει για το ποιες πληροφορίες θα εισαχθούν στο cell.
- Η **πύλη forget** καθορίζει το ποιες πληροφορίες θα διαγραφούν από το cell.
- Η **πύλη output** καθορίζει ποιες πληροφορίες από το cell θα χρησιμοποιηθούν ως output.

Εικ. Από (Thomas Fisher, 2017)



Εικόνα 4-7 Επεξήγηση LSTM δικτύου [19]

Εκπαίδευση

Για την εκπαίδευση του δικτύου εφαρμόζεται μία συνάρτηση «dropout» στις input πύλες των memory cells και στις recurrent συνδέσεις του δικτύου επιλέγοντας την τιμή 0.1 ως πιθανότητα drop. Αυτό είναι μια μέθοδος για να αποφευχθεί το overfitting του δικτύου. Επίσης χρησιμοποιείται η μέθοδος του πρόωρου σταματήματος (early stopping) χωρίζοντας το σετ εκπαίδευσης σε training και validation set και ελέγχοντας το αν είναι συμφέρων για το μοντέλο να συνεχιστεί η εκπαίδευση του. Με αυτόν τον τρόπο οι ερευνητές στοχεύουν στη περαιτέρω μείωση του κινδύνου του overfitting.

Στη συνέχεια το άρθρο δίνει μία σύντομη περιγραφή των μοντέλων (Random Forest, Deep Neural Network, και Logistic Regression) με τα οποία πρόκειται να συγκριθούν τα αποτελέσματα του LSTM δικτύου τους.

Αναφέροντας ότι για το μοντέλο Random Forest ακολούθησε την έρευνα του [4].

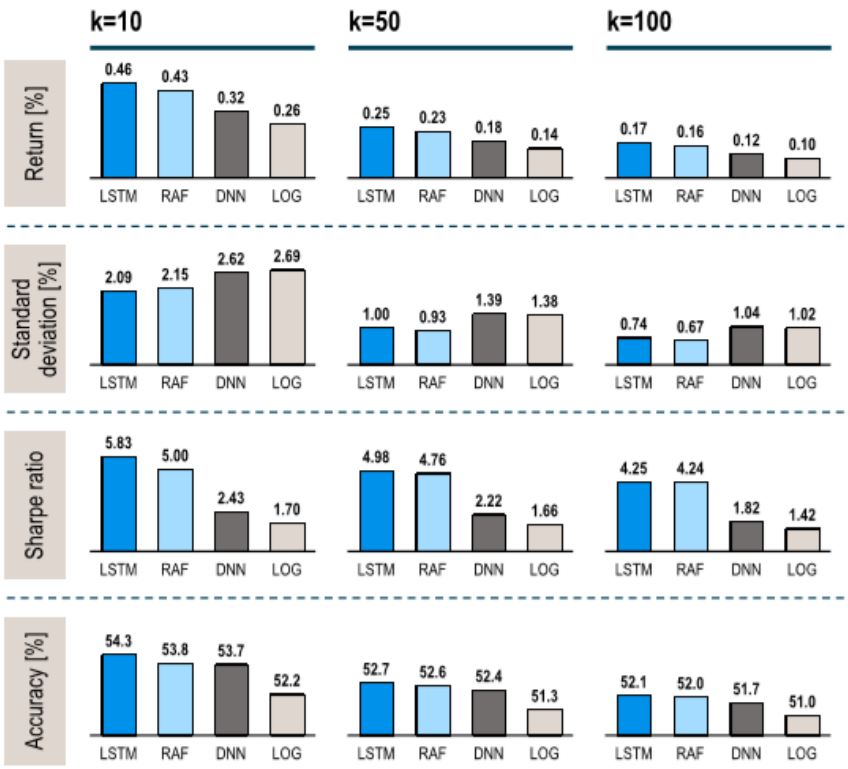
Για το μοντέλο Deep Neural Network διευκρινίζει ότι χρησιμοποιήθηκε ένα δίκτυο με 31 νευρώνες στο input layer, 31 στο 1ο hidden layer, 10 στο 2ο hidden, 5 στο 3ο hidden και 2 output νευρώνες για τις κλάσεις «UP» και «DOWN». Επίσης χρησιμοποιούνται 2 μέθοδοι regularization (dropout function και L1 matrix).

Το τελευταίο μοντέλο που περιγράφεται είναι ένα απλό μοντέλο Λογιστικής Παλινδρόμησης το οποίο χρησιμοποιείται με τις προεπιλογές που παρέχονται από τη Python βιβλιοθήκη sci-kit.

Για την σύγκριση, μετά την κατηγοριοποίηση που κάνει το κάθε μοντέλο στις μετοχές για κάθε χρονική στιγμή τους (από t=241 ως t=1000), για κάθε χρονική στιγμή t οι μετοχές ταξινομούνται κατά φθίνουσα σειρά ανάλογα με την πιθανότητα της «UP» κλάσης που προβλέφθηκε για τη κάθε μία.

Στη συνέχεια σε ένα long-short portfolio που αποτελείται από έναν αριθμό μετοχών ίσο με 2κ, επιλέγονται οι κ top μετοχές και οι k low μετοχές.

Επιλέγοντας τη θέση long για τις top μετοχές, και τη θέση short για τις low μετοχές για διάφορα κ, τα αποτελέσματα είναι αυτά που φαίνονται στον παρακάτω πίνακα:



Εικόνα 4-8 Αποτελέσματα [19]

Όπως είναι αναμενόμενο όσο μικρότερο είναι το κ τόσο πιο εύστοχη είναι η long και η short θέση για τις top και low μετοχές αντίστοιχα ασχέτος το μοντέλο που χρησιμοποιείται.

Από τον παραπάνω πίνακα φαίνεται ότι το μοντέλο LSTM δικτύου υπερνικά σε αποτελεσματικότητα όλα τα υπόλειπα.

Αναλύοντας περισσότερο τα αποτελέσματα οι ερευνητές παρατηρούν ότι η απόδοση του μοντέλου τείνει να μειώνεται όσο προχωρά χρονικά η χρονική περίοδος στην οποία εφαρμόζεται. Αυτό όπως υποστιρίζεται στο άρθρο συμβαίνει λόγω της διάδοσης των μεθόδων machine learning στον χώρο της ανταλαγής μετοχών, με αποτέλεσμα το μοντέλο που μελετάται να έχει ουσιαστικά μεγαλύτερο επίπεδο ανταγωνισμού όσο προχωρά η χρονική περίοδος. Αντίστοιχη τάση μείωσης της απόδοσης παρατηρείται και στο Random Forest μοντέλο, ωστόσο η απόδοση του LSTM μοντέλου είναι σταθερά καλύτερη από αυτή του Random Forest καθόλη τη διάρκεια της περιόδου μελέτης εκτός της περιόδου της παγκόσμιας οικονομικής κρίσης του 2008.

Περιληπτικά, τα συμπεράσματα των ερευνητών είναι ότι η βαθιά εκμάθηση σε μορφή Lstm δικτύων είναι ένα πεδίο έρευνας που ταιριάζει ιδιαίτερα στον χώρο της ανταλλαγής χρηματιστηριακών μετοχών, και ότι πιστεύουν πως περεταίρω έρευνες προς αυτή τη κατεύθυνση είναι ιδιαίτερα υποσχόμενες.

4.3 Έρευνες βασιζόμενες σε δεδομένα εξωγενών παραγόντων

4.3.1 Using Structured Events to Predict Stock Price Movement: An Empirical Investigation [24]

Σε αυτή την εμπειρική μελέτη οι ερευνητές σχεδίασαν ένα σύστημα πρόβλεψης της κίνησης μετοχών, το οποίο βασίζεται σε δομημένα events για τις προβλέψεις τους.

Η συγκεκριμένη μέθοδος χρησιμοποιεί ειδησεογραφικά άρθρα οικονομικού χαρακτήρα ως αρχικά δεδομένα. Σε αυτά στη συνέχεια μέσω της βιβλιοθήκης "Open IE" του πανεπιστημίου "Stanford" κάνει συντακτική ανάλυση της κάθε πρότασης στα κείμενα, με σκοπό να εξάγει δομημένα Events.

Τα δομημένα events περιγράφονται στο κείμενο της έρευνας ως οντότητες που αποτελούνται από τρείς παραμέτρους: τον δράστη, την σχέση, και το υποκείμενο.

Αφού εξάγουν κάθε event από τα κείμενα, προχωρούν σε μια διαδικασία γενικοποίησής τους.

Η γενικοποίηση των παραμέτρων του κάθε event αποτελείται από 2 βήματα: 1.Αρχικά, κατασκευάζουν ένα εργαλείο μορφολογικής ανάλυσης με βάση το πρότυπο WordNet [12] για να εξαγάγει τις απλοποιημένες μορφές των λέξεων.

2. Στη συνέχεια, εντάσσουν το κάθε ρήμα της 2ης παραμέτρου "Σχέση" σε κλάσεις, με σκοπό να γενικευτεί και άλλο το περιεχόμενο του event.

Μετά τη γενικοποίηση τα events έχουν πάρει πλέον την τελική τους μορφή. Η έρευνα αυτή δημιουργεί στη συνέχεια κάποια μοντέλα πρόβλεψης, που βασιζόμενα στην τελική μορφή των events εκπαιδεύονται στο να προβλέπουν για κάθε δείγμα, αν η τιμή της μετοχής την ημέρα στόχου, θα είναι ανοδική η καθοδική σε σχέση με την παρούσα τιμή.

Τα μοντέλα που ορίστηκαν είναι: Support Vector Machines(SVM) , και απλό νευρωνικό δίκτυο 2 κρυφών επιπέδων (NN).

Επίσης σαν εναλλακτική αρχική ανάλυση ορίστηκε και η τεχνική bag of words, η οποία αντλεί από τα κείμενα τον αριθμό των λέξεων των οποίων το νόημα είναι γνωστό ως θετικό ή αρνητικό.

Παρακάτω φαίνονται τα αποτελέσματα της έρευνάς τους συγκρίνοντας 4 μοντέλα πρόβλεψης.

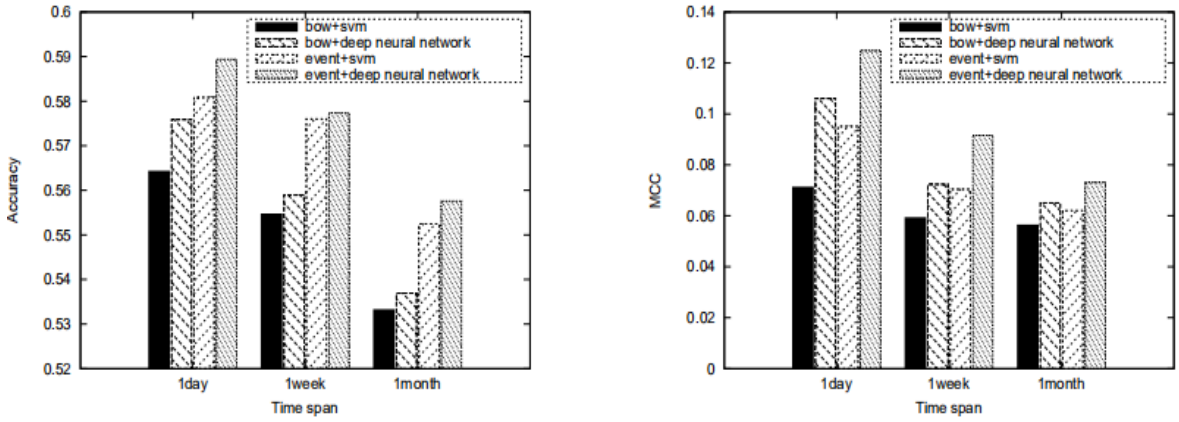
1) Bag Of Words to SVM (BOW_SVM)

2) Bag Of Words to NN (BOW_NN)

3) Event to SVM (event_SVM)

4) Events to NN (event_NN)

Ως προς την αποτελεσματικότητά τους σε προβλέψεις, ημέρας, εβδομάδας και μήνα.



Εικόνα 4-9 Αποτελέσματα [24]

Όπως φαίνεται η τεχνική ανάλυσης Bag Of Words αποδίδει χειρότερα και στα 2 μοντέλα σε σχέση με την εξαγωγή events. Επίσης παρατηρούμε ότι σε όλες τις περιπτώσεις το NN μοντέλο είναι ακριβέστερο του SVM.

Στη συνέχεια η έρευνα επεκτείνεται στο αν τα Events που εξάγονται μέσα από το σώμα του κάθε κειμένου είναι εξίσου σημαντικά με αυτά που εξάγονται από τον τίτλο του. Παρακάτω ο πίνακας αποτελεσμάτων:

	title	content	content + title	bloomberg title + title
Acc	59.60%	54.65%	56.83%	59.64%
MCC	0.1683	0.0627	0.0852	0.1758

Εικόνα 4-10 Σύνοψη αποτελεσμάτων [24]

Παρατηρούμε ότι το καλύτερο αποτέλεσμα επιτυγχάνεται με τη χρήση μόνο τίτλου του εκάστοτε κειμένου. Ακολουθεί χρήση του τίτλου και του περιεχόμενου ταυτόχρονα, και τελευταίο σε απόδοση είναι το αποτέλεσμα που επιτυγχάνεται με τη χρήση μόνο του περιεχομένου του κειμένου.

Τα συμπεράσματα της έρευνας είναι ότι η χρήση της μεθόδου εξαγωγής events με τον τρόπο που προτάθηκε σε αυτή την έρευνα είναι πιο αποτελεσματική στην εκπαίδευση μοντέλων πρόβλεψης της κίνησης των μετοχών, από ότι η χρήση της τεχνικής Bag Of Words. Επίσης συμπεραίνεται το ότι η ποιότητα του περιεχομένου των προτάσεων είναι πιο σημαντική από την ποσότητα, αφού η χρήση μόνο των τίτλων για εξαγωγή events αποδεικνύονται αποτελεσματικότεροι παρά τον μικρό σε μέγεθος όγκο πληροφορίας.

4.3.2 Deep Learning for Event-Driven Stock Prediction[25]

Η συγκεκριμένη μελέτη η οποία στηρίζεται σε μεγάλο βαθμό στην ακριβώς προηγούμενη που αναφέραμε εξετάζει και πάλι συστήματα πρόβλεψης μετοχών που στηρίζονται σε ειδησεογραφικά κείμενα οικονομικού ενδιαφέροντος.

Στη μεθοδολογία τους οι συγγραφείς περιγράφουν την εξαγωγή events με τη χρήση της βιβλιοθήκης “Open IE” από κείμενα των ειδησεογραφικών πρακτορείων “Bloomberg” και “Reuters”.

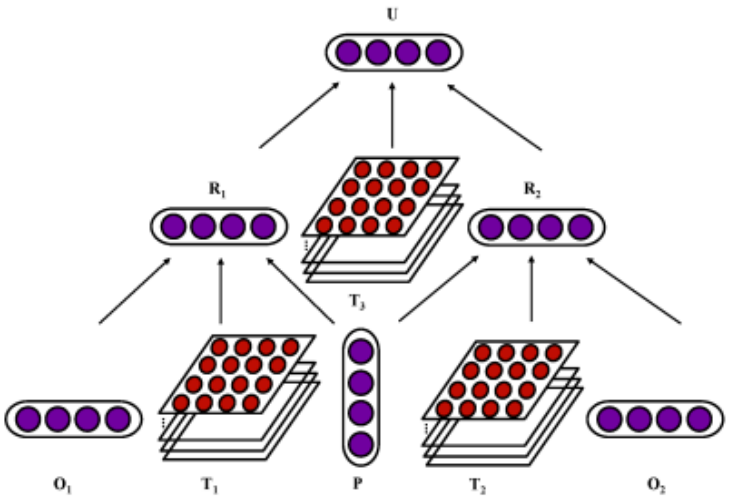
Στη συνέχεια ωστόσο η έρευνα διαφοροποιείται από την προηγούμενη που περιγράψαμε, προτείνοντας μια αναπαράσταση κάθε μιας από τις παραμέτρους των events ως συμπυκνωμένο διάνυσμα νοήματος (Word Embedding) των 100 διατάσεων.

Συγκεκριμένα στη μεθοδολογία αναφέρεται πως χρησιμοποιώντας τον αλγόριθμο Skipgram [20] κατασκευάζουν ένα λεξικό το οποίο συνδέει κάθε λέξη που περιέχει, με το αντίστοιχο διάνυσμα νοήματός της.

Με αυτόν τον τρόπο δημιουργούν 1 διάνυσμα για κάθε λέξη, κάθε παραμέτρου του event (παράμετροι event: O1, P, O2).

Στη συνέχεια χρησιμοποιούν τον μέσο όρο τον διανυσμάτων των λέξεων σε κάθε παράμετρο, έτσι ώστε να καταλήξουν σε μία αναπαράσταση 3 * 100 για κάθε event.

Εν συνεχεία χρησιμοποιώντας τα διανύσματα αυτά, αλλά και σκοπίμως φθαρμένες μορφές τους εκπαιδεύουν ένα “Neural Tensor Network” στην αναπαράσταση της αρχικής μορφής Διανυσμάτων νοήματος λέξεων (Word Embeddings, 3 * 100) σε ένα τελικό διάνυσμα αναπαράστασης νοήματος του Event (Event Embedding (U), 1 * 100). Παρακάτω ένα επεξηγηματικό σχήμα του δικτύου:



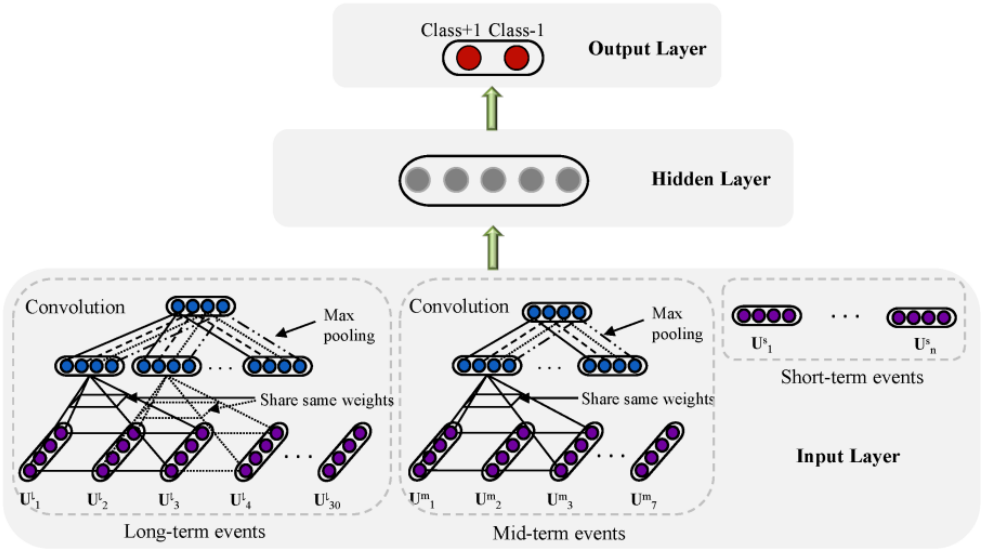
Εικόνα 4-11 Επεξήγηση Neural Tensor Network [25]

Όπου O1 η πρώτη παράμετρος του event (Υποκείμενο ή Δράστης), P η δεύτερη (Σχέση), O2 η τρίτη (Αντικείμενο).

Αφού λοιπόν δημιουργήσουν τα event από τα αρχικά κείμενα, τα ομαδοποιούν για κάθε μέρα παίρνοντας τον μέσο όρο μέρας για κάθε μία τιμή από τις 100 συνολικά του διανύσματος. Φτιάχνοντας έτσι μια χρονοσειρά από διανύσματα νοήματος event, για την περίοδο 2/10/2006 - 21/11/2013.

Εν τέλη τα δεδομένα αυτά εισάγονται σε 3 συνελικτικά νευρωνικά δίκτυα, τα οποία παίρνουν ως είσοδο τα διανύσματα του προηγούμενου μήνα, τα διανύσματα της προηγούμενης βδομάδας, και το διάνυσμα της προηγούμενης μέρας, με στόχο τον συνδυασμό τον εξόδων τους για την τελική πρόβλεψη της κλάσης (Άνοδος ή Κάθοδος) της μετοχής για την επόμενη ημέρα.

Παρακάτω φαίνεται το σχέδιο που επεξηγεί τη δομή του νευρωνικού δικτύου αυτού.



Εικόνα 4-12 Επεξήγηση Convolutional NN [25]

Στη συνέχεια συγκρίνουν το μοντέλο αυτό ως προς την ικανότητά του να προβλέπει σωστά την κατεύθυνση κίνησης του χρηματιστηριακού δείκτη SP500, με άλλα 6 απλούστερα μοντέλα.

Τα αποτελέσματα της σύγκρισης παρατίθενται στον παρακάτω πίνακα:

	Acc	MCC
Luss and d'Aspremont [2012]	56.42%	0.0711
Ding et al. [2014] (E-NN)	58.94%	0.1649
WB-NN	60.25%	0.1958
WB-CNN	61.73%	0.2147
E-CNN	61.45%	0.2036
EB-NN	62.84%	0.3472
EB-CNN	**65.08%**	**0.4357**

Εικόνα 4-13 Σύνοψη αποτελεσμάτων [25]

Επεξήγηση των μοντέλων σύγκρισης:

1. (Luss 2012)

Αυτή η έρευνα χρησιμοποιεί τη τεχνική bag of words για την ανάλυση των κειμένων, και ως μοντέλο μηχανικής μάθησης προτείνει το Support Vector Machines (SVM)

2. [24]

Αφορά την έρευνα που περιγράψαμε προηγουμένως, η οποία χρησιμοποιεί απλουστευμένες μορφές των αρχικών events, και ως μοντέλο χρησιμοποιεί ένα νευρωνικό δίκτυο 2 επιπέδων.

3. WB-NN

Τα διανύσματα των νοημάτων λέξεως για τις 3 παραμέτρους (3 * 100), με μοντέλο πρόβλεψης απλό νευρωνικό δίκτυο 2 επιπέδων.

4. WB-CNN

Το ίδιο με το μοντέλο 3, με τη διαφορά ότι σε αυτό χρησιμοποιείται συνελικτικό δίκτυο αντί για απλό.

5. E-CNN

Η απλοποιημένη μορφή από το μοντέλο 2, με μοντέλο πρόβλεψης ένα συνελικτικό δίκτυο παρόμοιο με αυτό που περιγράφηκε προηγουμένως.

6. EB-NN

Τα διανύσματα νοήματος των events, σε απλό νευρωνικό δίκτυο.

7. EB-CNN

Τα διανύσματα νοήματος των events, σε συνελικτικό νευρωνικό δίκτυο.

Τα αποτελέσματα της έρευνας είναι εντυπωσιακά. Μάλιστα υποστηρίζεται ότι το μοντέλο αυτό σε συνδυασμό με μία απλή στρατηγική ανταλλαγής μετοχών, πετυχαίνει τα διπλάσια ποσοστά κέρδους από ότι τα αντίστοιχα της έρευνας (Luss 2012).

ΚΕΦΑΛΑΙΟ 5 - ΣΧΕΔΙΑΣΜΟΣ ΤΗΣ ΔΙΚΗΣ ΜΑΣ ΕΡΕΥΝΑΣ

5.1 Εισαγωγή

Η έρευνά μας θα μπορούσε να χωριστεί σε δύο μέρη.

Στην αρχή της (στο πρώτο δηλαδή μέρος), η μελέτη μας επικεντρώθηκε στην διερεύνηση του αν μοντέλα πρόβλεψης ενδογενών παραγόντων που χρησιμοποιούν δεδομένα χρονοσειρών μεγάλης συχνότητας, έχουν εφαρμογή σε δεδομένα μικρότερης συχνότητας.

Το δεύτερο κομμάτι της έρευνάς μας, εφόσον τα αποτελέσματα του πρώτου ήταν αποθαρρυντικά ως προς την απόδοση των μοντέλων πρόβλεψης, είχε να κάνει με την υλοποίηση ενός συστήματος το οποίο χρησιμοποιεί εξωγενής παράγοντες, και συγκεκριμένα ειδησεογραφικά άρθρα οικονομικού ενδιαφέροντος, για να φτάσει σε πρόβλεψη.

5.2 Πρώτο Μέρος έρευνας. Μελέτη για την επιρροή της συχνότητας των δεδομένων των μετοχών, στην αποτελεσματικότητα του μοντέλου πρόβλεψης.

5.2.1 Εισαγωγή πρώτου μέρους

Η απόδοση ενός ANN (χρονικά και ποιοτικά) εξαρτάται από πολλούς παράγοντες, δομή δικτύου, μέγεθος διαστάσεων δεδομένων, μέγεθος dataset, συναρτήσεις κόστους και εκπαίδευσης, συνάρτηση ενεργοποίησης (activation function).

Ένας από τους σημαντικότερους είναι η μορφή των δεδομένων που θα του δώσει κανείς ως input στο δίκτυο.

Στη δημοσίευση [5], η οποία διερευνά την απόδοση deep learning αλγορίθμων στην πρόβλεψη χρηματιστηριακών δεικτών σε high frequency δεδομένα (ανά 5 λεπτά), συγκρίνοντας διαφόρων ειδών επεξεργασίες πάνω στο input dataset, παρατηρείται ότι στο μέσο όρο τους, τα μη επεξεργασμένα input (raw dataset) έχουν καλύτερη απόδοση σε σχέση με κάποιες δημοφιλής τεχνικές προ επεξεργασίας του dataset (PCA, Autoencoders, Restricted Boltzman Machine).

Επίσης η παραπάνω έρευνα υποστηρίζει πως η απόδοση του δικτύου ενισχύεται σε ένα μικρό βαθμό αν σε αυτό εισαχθούν τα υπολειπόμενα (residuals) από ένα μοντέλο αυτοπαλινδρόμησης το οποίο θα εφαρμοστεί πάνω στα raw input dataset.

Στη παρούσα έρευνα εξετάζεται αρχικά το κατά πόσο τα αποτελέσματα της έρευνας [5] έχουν αντίκρισμα και σε δεδομένα λιγότερο υψηλής συχνότητας όπως τα ημερήσια. Επίσης ερευνάται διεξοδικά το κατά πόσο η στατιστική προ επεξεργασία των δεδομένων μπορεί να βοηθήσει τις τεχνικές βαθιάς μάθησης. Για να επιτευχθεί αυτός ο σκοπός εξετάζουμε την απόδοση του νευρωνικού δικτύου όταν εισάγονται σε αυτό τα αποτελέσματα δημοφιλών στατιστικών μοντέλων όπως τα «κινούμενος μέσος όρος» και «κινούμενος μέσος όρος αυτοπαλινδρόμησης» όπως θα δούμε στη συνέχεια.

Για την υλοποίηση της έρευνάς μας επιλέξαμε τη γλώσσα προγραμματισμού Python καθώς είναι μία από τις πιο πλήρης σε βιβλιοθήκες Machine Learning γλώσσες, και η δημοφιλέστερη αυτή τη στιγμή γλώσσα στον σχεδιασμό τέτοιου είδους εφαρμογών.

Εργαλεία που χρησιμοποιήσαμε:

Γλώσσα προγραμματισμού: Python

Βιβλιοθήκες:

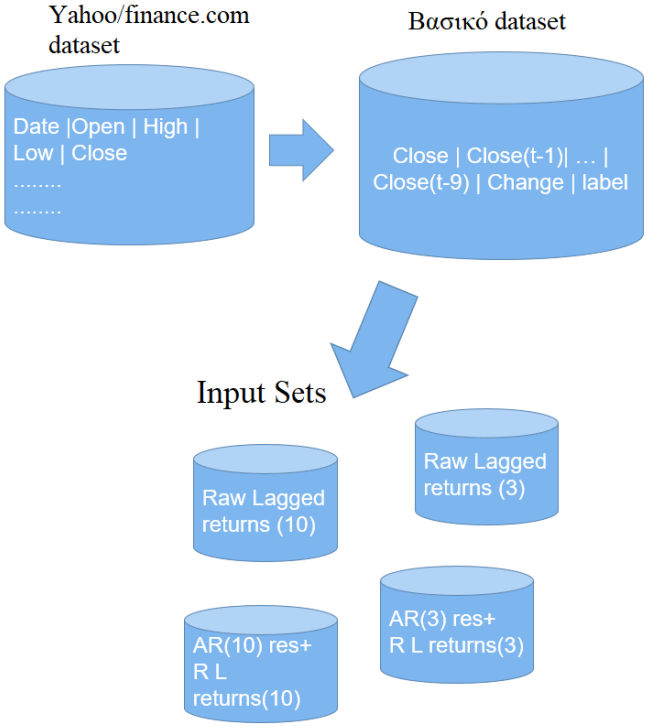
- **Numpy** (Για την εκτέλεση μεγάλου μέρους των αριθμητικών λειτουργιών του κώδικα)
- **Pandas** (Για φόρτωση dataset και προ επεξεργασία)
- **Scikit-learn** (Για τη χρήση Machine Learning αλγορίθμων που προσφέρει, αλλά και συναρτήσεις «Validation» των μοντέλων)
- **Tensorflow** (Για τις λειτουργίες που εκτελούν τα Νευρωνικά δίκτυα σε χαμηλό επίπεδο)
- **Keras** (Που χρησιμοποιεί ως back-end την tensorflow)

5.2.2 Data preparation

Αρχικά ως dataset επιλέξαμε δεδομένα από χρηματιστηριακούς δείκτες της πλατφόρμας S&P 500 λόγω της ποικιλίας σε μετοχές αλλά και της πληρότητας του συγκεκριμένου dataset. Τα δεδομένα έχουν εξαχθεί από την ιστοσελίδα (yahoo/finance.com).

Όπως προαναφέρθηκε, το input ενός νευρωνικού δικτύου αλλά και γενικώς ενός αλγόριθμου μηχανικής μάθησης είναι πολύ κρίσιμο στοιχείο όσο αναφορά την απόδοσή του τόσο στον χρόνο εκπαίδευσής όσο και στην ακρίβεια των μετέπειτα αποτελεσμάτων που θα παρέχει το μοντέλο.

Χρησιμοποιήσαμε 2 διαφορετικές μεθόδους προ επεξεργασίας των δεδομένων μας. Αξιοποιώντας το στατιστικό μοντέλο αυτοπαλινδρόμησης AR(10), καθώς και των συνδυασμό του με τα μη-επεξεργασμένα δεδομένα, σχηματίσαμε πρωτίστως ένα αρχικό dataset, το οποίο σε κάθε sample είχε τα απαραίτητα δεδομένα για να δημιουργήσουμε στη συνέχεια 4 σετ δεδομένων εισόδου, τα οποία στη συνέχεια χρησιμοποιούνται για την εκπαίδευση αλλά και την αξιολόγηση της απόδοσης του νευρωνικού μας δικτύου.



Εικόνα 5-1 Τρόπος δημιουργίας των δεδομένων εισόδου 1ου μέρους

Η πρώτη μέθοδος επεξεργασίας που χρησιμοποιήθηκε ήταν η κατασκευή του input data set ως ένας αριθμός προηγούμενων τιμών της κάθε μετοχής του dataset. Η δεύτερη είχε να κάνει με την εφαρμογή του στατιστικού μοντέλου αυτοπαλινδρόμησης.

Δημιουργήσαμε αρχικά 17 Dataset σε μορφή csv αρχείων, από τα οποία τα πρώτα 16 αντιστοιχούσαν σε 16 μετοχές του S&P500 που επιλέξαμε, και το 17ο αντιστοιχούσε στον ίδιο τον δείκτη S&P500.

Κάθε sample σε αυτά τα αρχεία, περιείχε τις τιμές «Close» της κάθε μετοχής για τις 10 προηγούμενες εργάσιμες ημέρες. Επίσης περιείχε τα υπόλοιπα (Residuals) του αποτελέσματος ενός μοντέλου αυτοπαλινδρόμησης που εφαρμόστηκε με τις 10 καθυστερημένες τιμές του close ως input, και το close της επόμενης ημέρας ως output. Τέλος το Sample περιείχε την τιμή «Close» της επόμενης ημέρας, την ποσοστιαία διαφορά της με την προηγούμενη αντίστοιχη τιμή, και μία ετικέτα η οποία παίρνει την τιμή 1 όταν η τιμή της μετοχής ανέβηκε, και 0 όταν η τιμή της μετοχής κατέβηκε.

Παρακάτω φαίνεται η εικόνα του αρχείου που αντιστοιχεί στον δείκτη S&P500:

	A	B	C	D	E
1	date	Close	Close -1	Close -2	Close
2	########	1318.03	1325.18	1317.64	132
3	########	1314.78	1318.03	1325.18	131
4	########	1326.37	1314.78	1318.03	132
5	########	1336.35	1326.37	1314.78	131
6	########	1336.59	1336.35	1326.37	131
7	########	1338.88	1336.59	1336.35	132
8	########	1335.85	1338.88	1336.59	133
9	########	1331.32	1335.85	1338.88	133
10	########	1334.11	1331.32	1335.85	133
11	########	1350.2	1334.11	1331.32	133
12	########	1353.22	1350.2	1334.11	133
13	########	1349.59	1353.22	1350.2	133
14	########	1350.66	1349.59	1353.22	13
15	########	1353.42	1350.66	1349.59	135
16	########	1349.95	1353.42	1350.66	134

K	L	M	N	O	P
Close -9	RES_3	RES_10	Next Clos	Change	label
1298.92	-0.44805	-0.44251	1294.02	0.018217	1
1299.54	-0.50882	-0.62579	1298.92	0.012063	1
1313	-0.56636	-0.42675	1299.54	0.020228	1
1318.07	-0.39444	-0.58921	1313	0.017473	1
1316.28	-0.02264	-0.10292	1318.07	0.013856	1
1319.66	-0.16596	0.093336	1316.28	0.01688	1
1321.18	-0.42653	-0.22126	1319.66	0.01212	1
1317.64	-0.35652	-0.37576	1321.18	0.007616	1
1325.18	-0.37329	-0.34545	1317.64	0.012345	1
1318.03	-0.38517	-0.48297	1325.18	0.018531	1
1314.78	-0.11698	-0.23347	1318.03	0.026005	1
1326.37	-0.338	-0.29615	1314.78	0.025793	1
1336.35	-0.76622	-0.43856	1326.37	0.017984	1
1336.59	-0.57474	-0.54242	1336.35	0.012613	1
1338.88	-0.27209	-0.37734	1336.59	0.009897	1

Εικόνα 5-2 Παράδειγμα αρχικού Data set 1ου μέρους

Έχοντας λοιπόν δημιουργήσει αυτά τα 16 βασικά αρχεία, είμασταν πλέον σε θέση να εκπαιδεύσουμε τα Deep Learning μοντέλα μας, επιλέγοντας κάθε φορά τα κατάλληλα features ανάλογα με τις ανάγκες καθ' ενός εκ των τεσσάρων ειδών input που θα μελετήσουμε και συνδυάζοντας τα δεδομένα των αρχείων μας όπου χρειαζόμαστε περισσότερες από μία μετοχές στην είσοδο του δικτύου μας.

Κάνοντας μια σύνοψη, τα αρχικά δεδομένα μας φαίνονται στον παρακάτω πίνακα:

Πίνακας 1 Αριθμητικά στοιχεία δεδομένων 1ου μέρους

Αρχικό Dataset	16 μετοχές + SP500
Samples κάθε Data set	1812
Χρονική περίοδος	21/09/2006 ως 29/11/2013
features των sample	16

Train-Test Split	80%-20%

5.2.3 Έλεγχος Προβλεπτικής δύναμης των δεδομένων

Όπως αναφέρθηκε και νωρίτερα η επιλογή των δεδομένων εισαγωγής σε ένα μοντέλο μηχανικής μάθησης είναι μια διαδικασία μείζονος σημασίας για τη μετέπειτα απόδοσή του. Για αυτό το λόγο πριν επιλέξουμε τα δεδομένα που θα χρησιμοποιήσουμε, εφαρμόσαμε σε αυτά κάποια τεστ, απλούστερων μοντέλων από αυτά των νευρωνικών δικτύων, με σκοπό να λάβουμε κάποιες ενδείξεις σε σχέση με της δυνατότητες των συγκεκριμένων δεδομένων όσο αφορά τη χρήση τους ως input σε πιο πολύπλοκα μοντέλα στη συνέχεια.

Λογιστική παλινδρόμηση είναι μία μέθοδος μηχανικής μάθησης η οποία ειδικεύεται σε προβλήματα 2αδικής κατηγοριοποίησης.

Έχοντας ως είσοδο ένα σετ δεδομένων σε διανυσματική μορφή (X_1,X_2,X_3,…X_v), παράγει μια έξοδο δυαδικής μορφής (y)

Μια τυπική συνάρτηση λογιστικής παλινδρόμησης φαίνεται παρακάτω:

Εξίσωση 1 συνάρτηση λογιστικής παλινδρόμησης

$$Y_i = \beta_0 + \beta_1 X_{1i} + \cdots + \beta_k X_{ki} + \varepsilon_i$$

Όπου τα β είναι συντελεστές οι οποίοι υπολογίζονται κατά τη διάρκεια της εκπαίδευσης του μοντέλου.

Η εκπαίδευση ενός μοντέλου λογιστικής παλινδρόμησης μπορεί να γίνει με ποικίλους τρόπους, στη συγκεκριμένη εργασία ωστόσο χρησισιμοποιούμε τη Νευτόνια μέθοδο. Για την κατηγοριοποίηση τον αποτελεσμάτων σε 2 κλάσεις χρησιμοποιούμε μια συνάρτηση υπόθεσης h, οπού h=sigmoid function, την οποία περιγράψαμε και προηγουμένως στο τμήμα των συναρτήσεων ενεργοποίησης.

Χρησιμοποιώντας την συνάρτηση υπόθεσης μπορούμε να υπολογίσουμε την πιθανότητα το διάνυσμα εισόδου μας, να δίνει ως έξοδο την κλάση 1, και την πιθανότητα να δίνει ως έξοδο την κλάση 0.

Εξίσωση 2 πιθανότητες κλάσεων 1 και 0

$$P(y = 1|\, x; \theta) = h_\theta(x)$$

$$P(y = 0|\, x; \theta) = 1 - h_\theta(x)$$

Όπου P(y=1|x;θ) η πιθανότητα το διάνυσμα χ να δώσει έξοδο 1, και P(y=1|x;θ), η πιθανότητα το διάνυσμα χ να δώσει έξοδο 0.

Συνυπολογίζοντας τις 2 παραπάνω σχέσεις καταλήγω σε μία ενοιαία σχέση για τον υπολογισμό της πιθανότητας της κλάσης όπως φαίνεται παρακάτω:

Εξίσωση 3 τελικός τύπος πιθανότητας λογιστικής παλινδρόμησης

$$P(y|\, x; \theta) = h_\theta(x)^y (1 - h_\theta(x))^{1-y}$$

Υπολογίζοντας με την παραπάνω σχέση την πιθανότητα για κάθε sample στα δεδομένα μου και συγρίνοντας τη με την πραγματική κλάση στην οποία ανήκει το καθένα, καταλήγω σε μία

συνολική ποσότητα η οποία αντιπροσωπεύει την απόσταση τον προβλέψεων του μοντέλου λογιστικής παλινδρόμησης, με τις πραγματικές τιμές των δεδομένων στόχου μας. Η μέθοδος του νεύτωνα χρησιμοποιείται για να μειώσει αυτήν τη ποσότητα.

Είναι ένας αλγόριθμος ο οποίος αποσκοπεί στην εύρεση των ριζών του πολυώνυμου χρησιμοποιώντας μερική παραγώγιση των μεταβλητών εισόδου, για αποφασίσει την κατεύθυνση του επόμενου βήματος εκμάθησης. Η συνάρτηση ενημέρωσης των βαρών (θ) έχει την παρακάτω μορφή:

Εξίσωση 4 Ενημέρωση βαρών

$$\hat{x}_{n+1} = \hat{x}_n - f(\hat{x}_n) * \nabla f(\hat{x}_n)^{-1}$$

Οπού χ^n οι παράμετροι του διανύσματος εισόδου, και f(x^n) η συνάρτηση διαβάθμισης.

Χρησιμοποιώντας τη βιβλιοθήκη της python "pandas"

Η **pandas** είναι μια open source βιβλιοθήκη που παρέχει μια μεγάλη ποικιλία συναρτήσεων και εργαλείων για την χειραγώγηση των σετ δεδομένων. Είναι εύκολη στη χρήση και αυτή τη στιγμή θεωρείται ένα από (αν όχι το) καλύτερα εργαλεία ανάλυσης δεδομένων για τη γλώσσα προγραμματισμού Python.

Εισάγουμε τα δεδομένα μας από τα αρχεία .csv μας, σε pandas dataframes, έτσι ώστε να μπορέσουμε να τα επεξεργαστούμε. Στη συνέχεια μετατρέψαμε τα δεδομένα των dataframes σε δομές πινάκων της βιβλιοθήκης numpy.

Η **NumPy**, η οποία αντιπροσωπεύει την Αριθμητική Python, είναι μια βιβλιοθήκη που αποτελείται από αντικείμενα πολυδιάστατης συστοιχίας (πολυδιάστατους πίνακες), και μια συλλογή συναρτήσεων για την επεξεργασία αυτών των συστοιχιών. Χρησιμοποιώντας NumPy, μπορούν να εκτελεστούν μαθηματικές και λογικές πράξεις σε συστοιχίες. Η βιβλιοθήκη είναι δομημένη πάνω στη γλώσσα C, κάτι που της δίνει πλεονέκτημα όσο άφορά την ταχύτητα εκτέλεσης των διαδικασιών.

Για το μοντέλο λογιστικής παλινδρόμησης, χρησιμοποιήσαμε τη βιβλιοθήκη **scikit-learn**. Η scikit-learn είναι μία από τις δημοφιλέστερες βιβλιοθήκες Python, για μηχανική μάθηση. Έχει τη μεγαλύτερη ποικιλία σε συναρτήσεις αλγορίθμων τέτοιου είδους καθώς και πολλές άλλες λειτουργίες για την προ επεξεργασία των δεδομένων και την αξιολόγηση των μοντέλων.

Και συγκεκριμένα τα πακέτα που φαίνονται στην παρακάτω εικόνα:

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
```

Εικόνα 5-3 imports 1ου μέρους εργασίας

Χρησιμοποιώντας την κλάση LogisticRegression λοιπόν δημιουργήσαμε ένα μοντέλο λογιστικής παλινδρόμησης, για κάθε μετοχή του Dataset.

Στη συνέχεια με τη βοήθεια της συνάρτησης fit εκπαιδεύσαμε το κάθε μοντέλο δινοντάς του ως διάνυσμα εισόδου τις τιμές κλεισίματος της μετοχής για τις 10 προηγούμενες ημέρες και έξοδο την κλάση της επόμενης ημέρας. Οι κλάσεις της επόμενης ημέρας ήταν δυαδικής φύσεως.

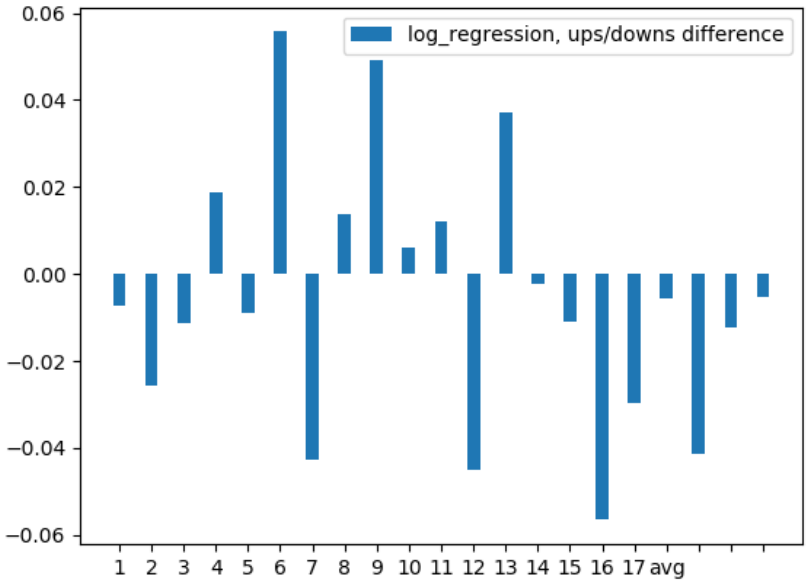
Η κλάση '0' αντιπροσώπευε την περίπτωση που η τιμή κλεισίματος της μετοχής την επόμενη ημέρα πέφτει.

Η κλάση ‘1’ αντιπροσώπευε την περίπτωση που η τιμή κλεισίματος της μετοχής την επόμενη ημέρα ανεβαίνει.

Επίσης ορίσαμε ένα επιπλέον μοντέλο αναφοράς (baseline), το οποίο προβλέπει την κλάση του κάθε sample επιλέγοντας σταθερά την κλάση με το μεγαλύτερο ποσοστό στο test set.

Η διαφορά στα αποτελέσματα των μοντέλων λογιστικής παλινδρόμησης που εκπαιδεύτηκαν για την κάθε μετοχή, με αυτά του μοντέλου αναφοράς παρουσιάζονται παρακάτω

2006-10 εως 2013-11 SP500 dataset:



Εικόνα 5-4 Διάγραμμα διαφοράς Log_reg & Baseline 1

Ups Average: 52.1%

Avg Accuracy score: 51.6%

Λαμβάνοντας υπ-όψην λοιπόν τα αποτελέσματα της λογιστικής παλινδρόμισης που εφαρμόσαμε στο dataset, έχουμε ενδείξεις ότι τα συγκεκριμένα δεδομένα είναι ενδεχομένως αδύναμα όσο αφορά τη προβλεπτική τους δύναμη.

Για του λόγου το αληθές ωστόσο το συγκεκριμένο dataset θα χρησιμοποιηθεί αργότερα σε κάποια από τα πιο περίπλοκα από αυτό της λογιστικής παλινδρόμησης μοντέλα τα οποία εξετάζονται σε αυτή την εργασία.

Εφόσον λάβαμε αυτήν την ένδειξη για τα συγκεκριμένα δεδομένα, κατάφύγαμε στην αναζήτηση ενός επιπλέον dataset με μεγαλύτερη πληρότητα και όγκο δεδομένων που πιθανώς θα μπορούσε να μας βοηθήσει να εξάγουμε καλύτερα αποτελέσματα κατά την εφαρμογή του παραπάνω μοντέλου.

Καταλήξαμε σε ένα dataset ημερήσιων δεδομένων το οποίο αποτελείται από 17 μετοχές της Αμερικανικής αγοράς οι οποίες έχουν διαθέσιμες ιστορικές τιμές για μεγαλύτερο χρονικό διάστημα από αυτό που μελετήσαμε. Παρακάτω φαίνεται το κομμάτι κώδικά με το οποίο φορτώνουμε τα καινούρια 17 csv αρχεία.

```
[dataset_path + 'aet.csv',

   dataset_path + 'cat.csv',
   dataset_path + 'cop.csv',
   dataset_path + 'dis.csv',
   dataset_path + 'gis.csv',
   dataset_path + 'mat.csv',
   dataset_path + 'ko.csv',
   dataset_path + 'mas.csv',
   dataset_path + 'noc.csv',
   dataset_path + 'oxy.csv',
   dataset_path + 'pki.csv',
   dataset_path + 'rok.csv',
   dataset_path + 'uis.csv',
   dataset_path + 'wfc.csv',
   dataset_path + 'wmt.csv',
   dataset_path + 'xom.csv',
   dataset_path + 'xrx.csv',]
```

Εικόνα 5-5 Φόρτωση μετοχών επιπλέον dataset

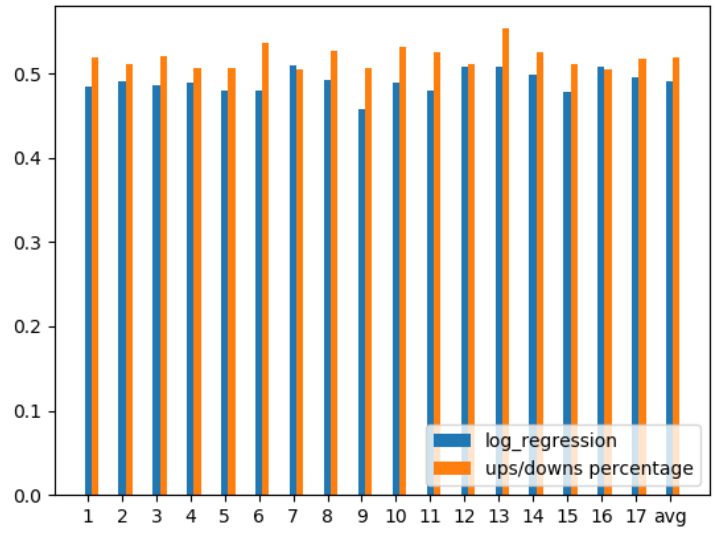
Τα στατιστικά στοιχεία αυτού του επιπλέον dataset που χρησιμοποιήσαμε φαίνονται παρακάτω.

Πίνακας 2 Αριθμητικά στοιχεία επιπλέον dataset 1ου μέρους

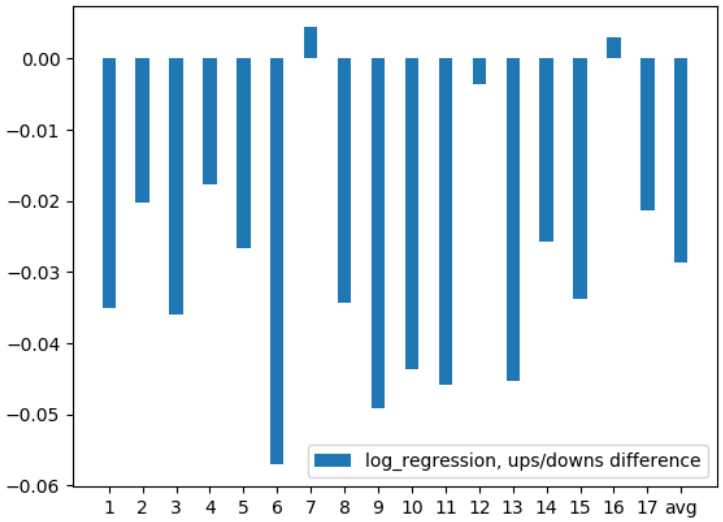
Αρχικό Dataset	17 μετοχές
Samples κάθε Data set	8035
Χρονική περίοδος	01/01/1985 ως 01/01/2017
features των sample	16
Train-Test Split	80%-20%

Με τη μέθοδο που χρησιμοποιήσαμε στο προηγούμενο σετ, εκπαιδεύσαμε 17 logistic regression models (ένα για κάθε μετοχή του dataset), τα αποτελέσματα που προέκυψαν περιγράφονται παρακάτω:

Εφαρμόσαμε το μοντέλο σε όλο το ημερομηνιακό εύρος των δεδομένων. Συγκεκριμένα η περίοδος που επιλέξαμε ήταν από την ημέρα 1985-1-1 εώς την ημέρα 2017-1-1. Τα αποτελέσματα για αυτήν τη περίοδο φαίνονται παρακάτω.



Εικόνα 5-6 Διάγραμμα απόδοσης Log_reg & Baseline 2



Εικόνα 5-7 Διάγραμμα διαφοράς Log_reg & Baseline 2

Ups Average: 0.5186945514606405

Avg Accuracy score: 0.4900246169673365

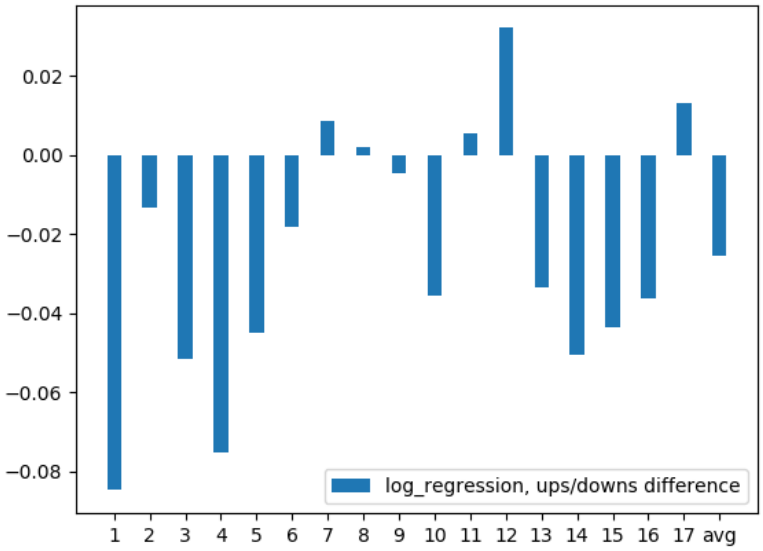
Τα αποτελέσματα για το επιπλέον αυτό dataset ήταν εξίσου αποθαρρυντικά, πιθανώς ένας από τους υπαίτιους λόγους να είναι ότι η δομή της χρηματιστηριακής αγοράς είναι δυναμική, και ο τρόπος συμπεριφοράς της αλλάζει παράλληλα με τον χρόνο.

Αφού λοιπόν έχουμε χωρίσει τα δεδομένα μας σε train και τεστ με σειριακό τρόπο. Δηλαδή στο training set εισάγουμε ένα ποσοστό από την αρχή της συνολικής περιόδου μελέτης, και στο test set εισάγουμε ένα ποσοστό από το τέλος της συνολικής περιόδου.

Είναι πιθανό οι πληροφορίες που εξάγονται από την περίοδο του training set να μην έχουν μεγάλη ισχύ στο test set, πράγμα που οδηγεί σε αδύναμες τελικές προβλέψεις.

Για να δοκιμάσουμε αυτήν την υπόθεση εφαρμόσαμε το παραπάνω τεστ για διαφορετικές περιόδους του συγκεκριμένου dataset.

Παρακάτω παρουσιάζεται το διάγραμμα του μέσου όρου της διαφοράς μεταξύ μοντέλου αναφοράς και μοντέλου λογιστικής παλινδρόμησης μετά από την εκτέλεση του συγκεκριμένου πειράματος για 10 διαφορετικές τυχαίες περιόδους μέσα στο dataset.



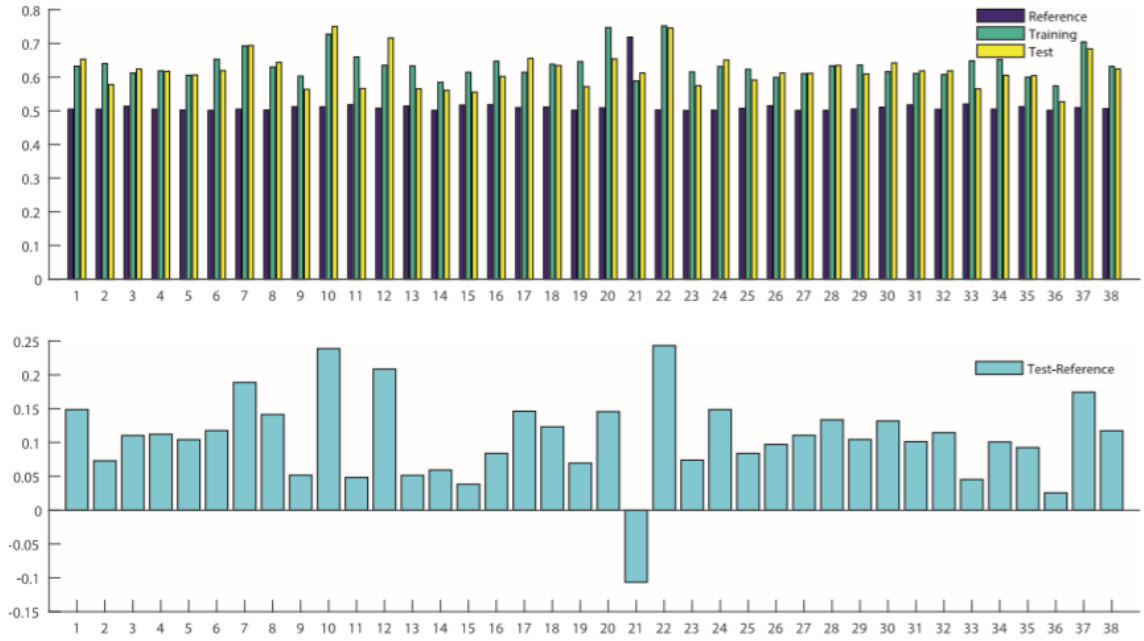
Εικόνα 5-8 Διάγραμμα διαφοράς Log_reg & Baseline 3

Ups Average: 0.5212418300653594

Avg Accuracy score: 0.49597523219814255

Παρατηρώ ότι η μέση τιμή του μοντέλου αναφοράς μας, ξεπερνά την μέση τιμή του μοντέλου logistic regression συνεπώς η περίοδος εφαρμογής πιθανότατα δεν είναι ο υπαίτιος λόγος των αδύναμων αποτελεσμάτων μας.

Στο αντίστοιχο πείραμα που εκτελέστηκε στην έρευνα [5] για την περίοδο που αναφέρουμε στο (4.2.2) τα αποτελέσματα ήταν σαφώς καλύτερα από τα μέχρι τώρα δικά μας. Παρακάτω παραθέτουμε τον αντίστοιχο πίνακα της δημοσίευσής τους.



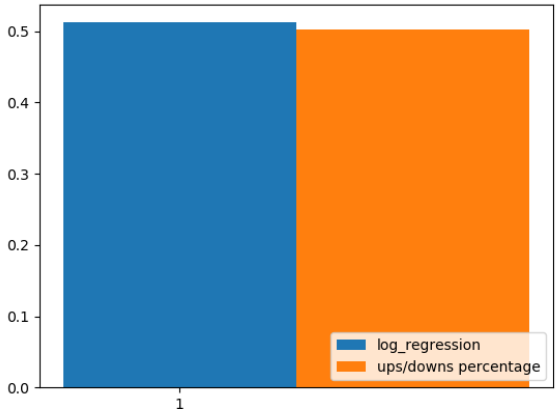
Εικόνα 5-9 Απόδοση κ Διαφορά Log_reg & Baseline [5]

Στον παραπάνω πίνακα φαίνονται τα αποτελέσματα του αντίστοιχου πειράματος με το προηγούμενο δικό μας με την διαφορά της επιπλέον μπάρας στο γράφημα που ανιστοιχεί στην επιτυχία του training set.

Όπως μπορούμε εύκολα να διακρίνουμε τα ποσοστά επιτυχίας πρόβλεψης κλάσης (UP/DOWN) που πετυχαίνει το μοντέλο λογιστικής παλινδρόμησης είναι κατά πολύ μεγαλύτερο από αυτό που πετυχαίνει το «reference», το οποίο είναι το αντιστοιχο μοντέλο σύγκρισης που περιγράψαμε προηγουμένως, δηλαδή η συνεχής επιλογή της κλάσης με το μεγαλύτερο ποσοστό στο test set.

Για να εξετάσουμε και με δικά μας πειράματα το αν τα παραπάνω αποτελέσματα οφείλονται στο είδος των δεδομένων, δηλαδή στη συχνότητα των τιμών του dataset, εφαρμόζουμε ακριβώς το ίδιο πείραμα αλλά με αρχικά δεδομένα τις τιμές μετοχής μιας τράπεζας ανα 1 λεπτό.

Παρακάτω φαίνετια η απόδοση των 2 μοντέλων μετά από 10 εφαρμογές τους σε διαφορετικές περιόδους του dataset.



Εικόνα 5-10 Απόδοση Log_reg & Baseline 1min. dataset

Ups Average: 0.502

Avg Accuracy score: 0.5125187781672509

Τα παραπάνω τεστ, μας παρέχουν ενδείξεις για τη δύναμη του ημερησιου dataset με καθυστερημένες επιστροφές τιμών κλεισίματος. Περιμένουμε ότι τα αποτελέσματα από τα πιο περίπλοκα μοντέλα μηχανικής μάθησης, θα υστερούν σε απόδοση σε σχέση με αυτά που παρουσιάζονται στην έρευνα [5].

Στη συνέχεια λοιπόν εξετάσαμε τους ελαφρώς πιο περίπλοκους τρόπους προεπεξεργασίας του αρχικού μας dataset, που προτείνονται [5].

Σε πρώτη φάση εξετάσαμε την εφαρμογή του αλγόριθμου μηχανηκής μάθησης «Random Forest» σε δεδομένα 10 καθυστερημένων επιστροφών της τιμής της μετοχής, ο οποίος σε αντίθεση με τους περισσότερους αλγορίθμους του είδους του, έχει την ικανότητα να μην παθαίνει υπερτροφοδότηση ανεξάρτητα με τον όγκο και τις συνθήκες της εκμάθησης. Έπειτα αναπτύξαμε ένα νευρωνικό δίκτυο πρόσθιας τροφοδότησης για την συνέχιση των πειραμάτων.

5.2.4 Τεχνητό Νευρωνικό Δίκτυο

Στη συνέχεια, και σαν δεύτερη φάση των πειραμάτων μας, χρησιμοποιήσαμε και τις 4 τιμές των μετοχών με διαφορετικούς αριθμούς καθυστερημένων τιμών και εστιάσαμε στην ικανότητα του νευρωνικού μας να προβλέπει την κλάση του Sample (UP/DOWN), έτσι ώστε αφενός να επεκτείνουμε την έρευνα στα μοντέλα ενδογενών παραγόντων, αλλά και αφετέρου να αποκτήσουμε ένα μέτρο σύγκρισης για τα αποτελέσματα του δεύτερου μέρους της εργασίας.

Κατασκευάσαμε το νευρωνικό Δίκτυο αρχικά ακολουθώντας τα χαρακτηριστικά του Νευρωνικού το οποίο αναφέρεται επίσης στην έρευνα [5], για να διερευνήσουμε το κατά πόσο τα αποτελέσματα του στο δικό του (high frequency) dataset συνάδουν με τα δικά μας αποτελέσματα από το low frequency dataset.

Το νευρωνικό δίκτυο αποτελείται από:

- 1 επίπεδο εισόδου (X) με αριθμό νευρώνων ίσο με τον αριθμό των features στα samples του κάθε ενός από τα Dataset που έχουμε δημιουργήσει.
- 1 κρυφό επίπεδο (H) με αριθμό νευρώνων ίσο με τον αριθμό νευρώνων εισόδου διά 2.
- 1 κρυφό επίπεδο (h) με αριθμό νευρώνων ίσο με τον αριθμό νευρώνων εισόδου διά 4.
- 1 επίπεδο εξόδου (y) με αριθμό νευρώνων ίσο με τον αριθμό των μετοχών εισόδου.

Για τα πειράματά μας παραμετροποιήσαμε το δίκτυο όπως φαίνεται παρακάτω.

Παραμετροποίηση Δικτύου:

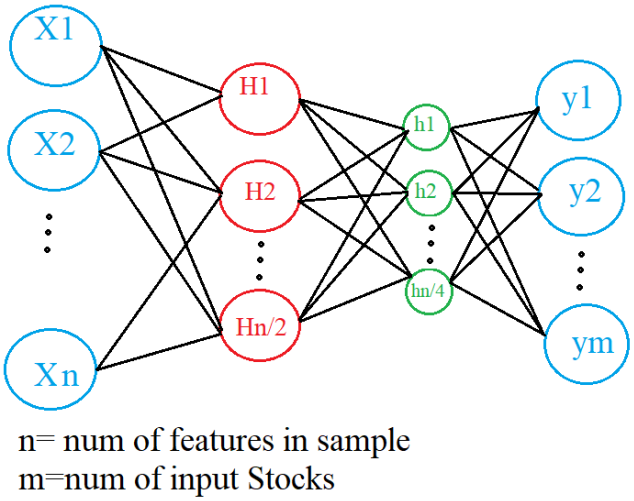
Αντικειμενική συνάρτηση = MSE minimize

Ρυθμός μάθησης = 0.01

Epochs = 3000

Regularization = L2 Matrix

Συνάρτηση ενεργοποίησης : Κρυφά επίπεδα “ReLU”, Επίπεδο εξόδου “Sigmoid”



Εικόνα 5-11 Νευρωνικό Δίκτυο 1ου μέρους

Εκπαιδεύσαμε το Νευρωνικό μας Δίκτυο ακολουθώντας την γενική μεθοδολογία των [5], με απώτερο σκοπό να καταλήξουμε σε ένα τελικό συμπέρασμα για την διαφορά προβλεπτικής δύναμης που εμπεριέχεται στα 2 διαφορετικά είδη δεδομένων (υψηλής και χαμηλής συχνότητας).

5.2.5 Raw lagged returns

Εξετάσαμε 4 περιπτώσεις όσο αφορά το πλήθος των καθυστερημένων τιμών που θα εισαχθούν ως είσοδος στο νευρωνικό δίκτυό μας.

- 1 καθυστερημένη τιμή
- 3 καθυστερημένες τιμές
- 5 καθυστερημένες τιμές
- 10 καθυστερημένες τιμές

Συγχωνέψαμε τα 15 αρχεία των μετοχών μας έτσι ώστε να δημιουργήσουμε δεδομένα εισόδου με τιμές επιστροφών κάθε μετοχής, για 1, 3, 5 ή 10 καθυστερημένες χρονικές στιγμές.

Έτσι καταλήξαμε σε 4 σετ δεδομένων εισόδου με μορφή:

1. [X1.τ(0), X2.τ(0),…, X15.τ(0)]
2. [X1.τ(0), X1.τ(-1), X1.τ(-2), X2.τ(0),…, X15.τ(0),… X15.τ(-2)]
3. [X1.τ(0), X1.τ(-1),…, X1.τ(-4), X2.τ(0),…, X15.τ(0),… X15.τ(-4)]
4. [X1.τ(0), X1.τ(-1),… X1.τ(-9), X2.τ(0),…, X15.τ(0),…. X15.τ(-9)]

για κάθε τ που ανήκει στο (11/2006 - 10/2013).

5.2.6 Αποτελέσματα

Για την αξιολόγηση της απόδοσης τον συστημάτων μας τα μέτρα που χρησιμοποιήσαμε είναι τα:

- ACCURACY: Ποσοστό επιτυχίας στη πρόβλεψη κλάσης (UP/DOWN).

- Mse(Mean Squared Error): Η μέση τετραγωνισμένη απόσταση μεταξύ πρόβλεψης του μοντέλου, και της τιμής του στόχου (target).

Μετά την εκτέλεση των παραπάνω βημάτων, εισάγαμε τα 4 input sets στο νευρωνικό δίκτυο χωρίζοντας τα προηγουμένως σε σετ Εκπαίδευσης (80%) και σετ ελέγχου (20%).

Για εξαγωγή ασφαλέστερων συμπερασμάτων εκπαιδεύσαμε το κάθε μοντέλο 10 φορές χρησιμοποιώντας κάθε φορά διαφορετική σειρά (shuffles) στα samples των δεδομένων.

Παρακάτω παρουσιάζεται ο πίνακας των συγκεντρωτικών αποτελεσμάτων.

1 hidden layer NN, Multi-Stock Net

Lag(.)	accuracy	mse
10	0.493	0.006
5	0.5027	0.00055
3	0.513	0.00041
1	0.505	0.00036

2 hidden layer NN, Multi-Stock Net

Lag(.)	accuracy	mse
10	0.509	0.00037
5	0.5176	0.00034
3	0.514	0.00034
1	0.516	0.00034

1 hidden layer NN, Single-Stock Net

Lag(.)	accuracy	mse
10	0.536	~0
5	0.536	~0
3	0.536	~0
1	0.536	~0

2 hidden layer NN, Single-Stock Net

Lag(.)	accuracy	mse
10	0.536	~0
5	0.536	~0
3	0.536	~0
1	0.536	~0

(3.000 epochs, Learning Step= 0.1)

Εικόνα 5-12 Σύνοψη αποτελεσμάτων 1ου μέρους

5.2.7 Auto Regressive Model

Καθώς υποστηρίζεται ότι τα αποτελέσματα του μοντέλου βελτιώνονται ελαφρώς με τη χρήση των υπολοίπων από ένα μοντέλο αυτοπαλινδρόμησης του «lag(10)» μοντέλου, ως είσοδο στο νευρωνικό δίκτυο, τρέξαμε το πείραμα μας για ακόμα 2 input sets. Αυτή τη φορά ενισχύσαμε το Lag(10) και το Lag(3) μοντέλο μας με τα residuals από την εφαρμογή του AR(10) μοντέλου (RES(10)). Τα επιπλέον input set μας ήταν:

1. [X1.τ(0), X1.τ(-1), X1.τ(-2), X2.τ(0),…, X15.τ(0),… X15.τ(-2)]
2. [X1.τ(0), X1.τ(-1),… X1.τ(-9),RES(10)1.τ, X2.τ(0),…, X15.τ(0),…. X15.τ(-9) RES(10)15.τ]

Η ολοκλήρωση των επιπλέον αυτών πειραμάτων έδειξε ότι τα αποτελέσματα δεν δείχνουν να βελτιώνονται, αλλά ούτε και να αλλάζουν γενικώς.

Όπως λοιπόν φαίνεται στα παραπάνω αποτελέσματα, η προβλεπτική δύναμη της τιμής κλεισίματος είναι μεγαλύτερη στα high frequency data

Υποθέτουμε ότι μια αιτία για το παραπάνω γεγονος μπορεί να αποτελεί το ότι ο αθρώπινος-εξωγενής παράγοντας ο οποίος για το παραπάνω πείραμά μας είναι αστάθμιτος, επιρεάζει σε πολύ μεγαλύτερο βαθμό τις τιμές της ημέρας, καθώς ο χρόνος μεταξύ των τιμών είναι αρκετός για να παρθούν ανθρώπινες αποφάσεις, αντίθετα οι μικροδομές της αγοράς σε υψηλής συχνότητας τιμές, οδηγούνται κυρίως από αλγορίθμους.

5.3 Δεύτερο μέρος έρευνας, μοντέλο πρόβλεψης βασιζόμενο σε εξωγενείς παράγοντες

5.3.1.1 NLP και Χρηματιστηριακές αγορές

Ως γνωστών, κάθε μέρα μέσα στο διαδίκτυο κινείται μεγάλος όγκος πληροφορίας που αφορά τη χρηματιστηριακή αγορά. Ωστόσο εκτός από τις αριθμητικές πληροφορίες (στιγμιαίες τιμές, μέσοι όροι, οικονομικοί δείκτες κτλ.), υπάρχει και ένα άλλο είδος πληροφορίας, το οποίο πολύ συχνά είναι και το καθοριστικότερο στον τρόπο κίνησης της αγοράς. Αναφερόμαστε στα κείμενα που γράφονται καθημερινά, και αφορούν τον χώρο. Κείμενα όπως άρθρα ανάλυσης δεικτών, αναρτήσεις ατόμων σχετικά με τον κλάδο και αντίστοιχα σχόλια, αλλά και στην περίπτωσή μας κείμενα ειδήσεων οικονομικού ενδιαφέροντος.

Όλη αυτή η πληροφορία ωστόσο, παρόλο που υπάρχει και κινείται ελεύθερα σε μεγάλο βαθμό στο διαδίκτυο η διαχείριση, επεξεργασία και χρησιμοποίησή της από τον άνθρωπο είναι μία δύσκολη διαδικασία, λόγο κυρίως του όγκο της ο οποίος είναι συνήθως κλίμακας μεγαλύτερης από αυτή που μπορούμε να επεξεργαστούμε σαν άτομα. Αυτός είναι και ο λόγος για τον οποίο ίσως μια μέθοδος λύσης βασισμένη σε NLP συστήματα, να είναι ιδανική για το πρόβλημά μας.

5.3.1.2 Περιγραφή της μεθοδολογίας

Αφού τα αποτελέσματα από το πρώτο κομμάτι της έρευνας ήταν αποθαρρυντικά ως προς την απόδοση των μοντέλων που στηρίχθηκαν σε αυτή τη μεθοδολογία.

Αποφασίσαμε να στραφούμε στη χρησιμοποίηση ειδήσεων οικονομικού ενδιαφέροντος, και αξιοποίηση συστημάτων NLP για τη δημιουργία μοντέλων πρόβλεψης. Χρησιμοποιώντας ως data set ένα σύνολο από ειδησεογραφικά άρθρα οικονομικού ενδιαφέροντος, από μια περίοδο περίπου 7 χρόνων, στα οποία εφαρμόσαμε μια σειρά από NLP αλγορίθμους τους οποίους θα αναλύσουμε στη συνέχεια, φτιάξαμε 5 μοντέλα πρόβλεψης, τα μοντέλα αυτά αναλύονται και συγκρίνονται ως προς την απόδοση στα επόμενα κεφάλαια.

5.3.2 Αρχικό Dataset

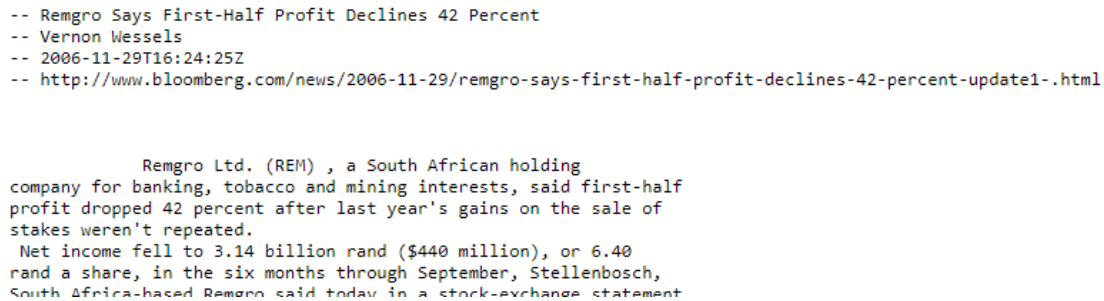
Παρακάτω παραθέτουμε τα αριθμητικά δεδομένα του dataset που θα περιγράψουμε παρακάτω

Πίνακας 3 Αριθμητικά δεδομένα dataset 2ου μέρους

Χρονική Περίοδος	2/10/2006 ως 21/11/2013
Αρχικά Κείμενα	452.284
Τίτλοι	447.378
Events	25.874
Avg WB	1.004
Avg WB labeled	720
Train-Test Split	80%-20%
Train-Validation Split	80%-20%

Ως αρχικό dataset χρησιμοποιήσαμε τα άρθρα ειδήσεων που χρησιμοποιήθηκαν στις έρευνες [24] και [25] οι οποίες αναφέρονται και στη συνέχεια της εργασίας. Το συγκεκριμένο σύνολο δεδομένων αποτελείται από 452.284 άρθρα τα οποία αποτελούν ειδήσεις μέσα στο διάστημα 02/10/2006 με 21/11/2013, 2 του μεγάλου ειδησεογραφικού πρακτορείου (bloomberg.com n.d.)

Το κάθε άρθρο περιλαμβάνει, τίτλο, ημερομηνία δημοσιοποίησης, συγγραφέα του άρθρου, και τέλος το κείμενο.



Εικόνα 5-13 Παράδειγμα κειμένου

5.3.3 Εξαγωγή γεγονότων (Event Extraction)

Όπως αναφέρεται στην έρευνά [24], Η εξαγωγή γεγονότων από ένα κείμενο μπορεί να βοηθήσει στην επεξεργασία νοήματος ελεύθερου κειμένου.

Γεγονότα, όπως ορίζονται και στην μελέτη [24] και [25] είναι σύνολα από tokens τα οποία εξάγονται από το ελεύθερο κείμενο, και έχουν την μορφή: («οντότητα 1», «σχέση», «οντότητα 2»). Η κάθε «οντότητα» μπορεί να αποτελείται από μία ή περισσότερες λέξεις και αποτελεί το υποκείμενο ή αντικείμενο μιας δράσης, η οποία με τη σειρά της ορίζεται στην παράμετρο «σχέση».

Για παράδειγμα για την πρόταση: «Ο διευθύνων σύμβουλος της εταιρίας Google ανακοίνωσε την παραίτησή του στο ΔΣ της εταιρίας.»

Μπορούμε να εξάγουμε ένα γεγονός με τη μορφή:

(O1, P, O2), όπου:

O1: Διευθύνων σύμβουλος της εταιρίας Google

P : ανακοίνωσε την παραίτησή του

O2: ΔΣ της εταιρίας.

Για την εξαγωγή γεγονότων από τα κείμενα χρησιμοποιήσαμε το εργαλείο «ReVerb» το οποίο δημιουργήθηκε από τους [1] χρησιμοποιώντας την open source βιβλιοθήκη του πανεπιστημίου Stanford, “Stanford IE” . Το Εργαλείο είναι γραμμένο σε γλώσσα προγραμματισμού Java και λειτουργεί ως εξής:

Για κάθε πρόταση χρησιμοποιεί την βιβλιοθήκη “Stanford Open Information Extraction” για να κάνει συντακτική ανάλυση. Στη συνέχεια βάση της ανάλυσης αυτής αποφασίζει αν μπορεί να εξάγει ένα γεγονός (της μορφής που περιγράψαμε παραπάνω). Αν η συντακτική ανάλυση προβλέπει την ύπαρξη γεγονότος, από αυτό εξάγεται ένας πίνακας με πληροφορίες του γεγονότος όπως: το κείμενο εξαγωγής, τον αριθμό της πρότασης από την οποία εξάχθηκε, την πιθανότητα το γεγονός να έχει εξαχθεί σωστά, τις παραμέτρους του γεγονότος, την συντακτική ανάλυση του γεγονότος, και την απλοποιημένη μορφή των παραμέτρων (μετά από αφαίρεση stop words και παρόμοιες διαδικασίες).

```
$ echo "Bananas are an excellent source of potassium." |
    ./reverb -q | tr '\t' '\n' | cat -n
 1  stdin
 2  1
 3  Bananas
 4  are an excellent source of
 5  potassium
 6  0
 7  1
 8  1
 9  6
10  6
11  7
12  0.9999999997341693
13  Bananas are an excellent source of potassium .
14  NNS VBP DT JJ NN IN NN .
15  B-NP B-VP B-NP I-NP I-NP I-NP I-NP O
16  bananas
17  be source of
18  potassium
```

Εικόνα 5-14 Παράδειγμα χρήσης Reverb

Όπως αναφέρεται στην έρευνα [24] μπορούμε να πετύχουμε μεγαλύτερη ακρίβεια ενός μοντέλου τέτοιου είδους, χρησιμοποιώντας μόνο τους τίτλους των κειμένων.

Εφαρμόζοντας το ReVerb λοιπόν στους τίτλους των άρθρων από τα κείμενα του dataset μας, εξάγουμε αρχικά γεγονότα “events”.

Για να έχουμε ωστόσο καλύτερο ποσοστό σε γεγονότα που έχουν εξαχθεί σωστά, φιλτράραμε τις συνολικές εξαγωγές event του reverb χρησιμοποιώντας την τιμή “confidence” της κάθε εξαγωγής, την τιμή της πιθανότητας δηλαδή που μας παρείχε το ReVerb για το αν το γεγονός έχει εξαχθεί σωστά.

Το σύνολο των τελικών γεγονότων προς χρήση συνεπώς ήταν 25.874 .

5.3.4 Word embeddings(WB)

Όπως αναφέρθηκε και προηγουμένως υπάρχει ένα μεγάλο χάσμα μεταξύ του τρόπου λειτουργίας της της φυσικής γλώσσας και της γλώσσας των H/Y, καθώς η φυσική γλώσσα είναι ρευστή και ασαφής, σε αντίθεση με τη γλώσσα μηχανής.

Συνεπώς θα πρέπει να βρεθεί ένας αποδοτικός τρόπος αναπαράστασης του νοήματος των λέξεων.

Για αυτό το πρόβλημα επιλέξαμε να συμβουλευτούμε την έρευνα [25] στην οποία οι ερευνητές χρησιμοποιούν τον αλγόριθμο «skip-gram» [20] και δημιουργούν συμπυκνωμένα διανύσματα τα οποία αναπαριστούν την κάθε λέξη σε έναν N-διάστατο χώρο ο οποίος έχει ως στόχο να ομαδοποιήσει γεωμετρικά της λέξεις με παρόμοιο νόημα.

Στην έρευνά μας λοιπόν χρησιμοποιούμε τον αλγόριθμο skip-gram μέσω της βιβλιοθήκης «Word2Vec», η οποία είναι μία από τις πιο πλήρεις βιβλιοθήκες στην αναπαράσταση λέξεων με τη μορφή διανύσματος, για να δημιουργήσουμε word embeddings, ή αλλιώς νοήματα λέξεων σε μορφή διανύσματος.

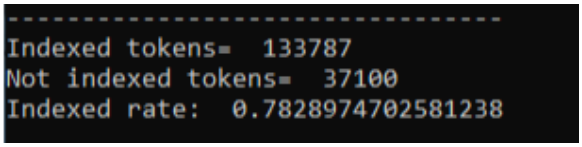
Χρησιμοποιώντας τα κείμενα του dataset, και χωρίζοντάς τα σε προτάσεις, δημιουργήσαμε ένα «λεξικό» αναπαράστασης λέξεων σε διανύσματα των 100, με μέγεθος λεξικού περίπου 700.000 λέξεις.

Δημιουργία WB dataset

Χρησιμοποιώντας το λεξικό που φτιάξαμε λοιπόν, και εφαρμόζοντάς το σε κάθε τα event από τους τίτλους των άρθρων, δημιουργήσαμε ένα σύνολο από word embeddings.

Για κάθε λέξη κάθε event, αν το embedding της υπήρχε στο λεξικό τότε δημιουργούμε ένα διάνυσμα των 100 διαστάσεων με τους αριθμούς που βρίσκονται στο λεξικό, αν αντίθετα η λέξη δεν υπάρχει στο λεξικό δημιουργούμε ένα διάνυσμα 100 διατάσεων με κεντρική (μέση) τιμή στη θέση κάθε αριθμού.

Το ποσοστό των λέξεων από τα events οι οποίες βρισκόταν στο λεξικό ήταν ~78%



Εικόνα 5-15 Ποσοστό μετατροπής λέξεων σε WB

Στη συνέχεια για ομαδοποιήσαμε τα embeddings μας ως προς τις 3 παραμέτρους του κάθε event (O1 , P, O2), ακολουθώντας και πάλι τη μεθοδολογία [25].

Μετά από αυτό το βήμα συνεπώς έχουμε δημιουργήσει ένα dataset, το οποίο για κάθε sample περιέχει την Ημερομηνία του event και το «μέσο» word embeddings κάθε παραμέτρου του event. Παρακάτω φαίνεται η μορφή του dataset σε αρχείο csv.

	A	B	C	D	E	F	G	H	I	J	K
1	date	0	0.0.1	0.0.2	0.0.3	0.0.4	0.0.5	0.0.6	0.0.7	0.0.8	0.0.9
2	########	0.389495	-0.10208	1.010095	-2.22525	-0.12566	-1.00048	-0.04079	3.501391	1.589972	1.248
3	########	2.334112	-1.77702	0.49549	-1.39564	0.319414	3.971314	-2.03963	4.168391	4.129178	-4.32
4	########	2.334112	-1.77702	0.49549	-1.39564	0.319414	3.971314	-2.03963	4.168391	4.129178	-4.32
5	########	0.029547	-0.93484	-0.7985	0.190842	-1.69287	0.523265	-5.53054	0.738107	-2.33831	-3.37
6	########	-1.30592	-0.44584	-1.99646	-1.22859	-1.27871	3.342243	-0.11103	-0.17443	1.353917	1.209
7	########	-0.39394	2.940721	-2.86108	-0.20798	-1.99647	2.469187	-0.38668	0.455732	0.221486	3.469
8	########	-0.39394	2.940721	-2.86108	-0.20798	-1.99647	2.469187	-0.38668	0.455732	0.221486	3.469
9	########	-1.32729	0.455439	0.287863	-1.46216	0.353356	-0.96913	-2.51852	-0.72999	-0.05167	-0.7
10	########	0.998671	0.689857	-0.02998	0.366409	0.582893	-0.1131	-0.90119	-1.18072	0.400353	-1.85
11	########	0	0	0	0	0	0	0	0	0	
12	########	0.362837	1.57078	1.372277	-0.14281	-0.11867	-0.4574	-0.28762	-0.14077	-1.03979	-0.77
13	########	0.362837	1.57078	1.372277	-0.14281	-0.11867	-0.4574	-0.28762	-0.14077	-1.03979	-0.77
14	2/7/2007	0.342361	2.270382	-0.25611	0.321489	-0.43989	-0.93147	-2.00822	-2.13612	-0.10851	0.260

Εικόνα 5-16 Παράδειγμα WB dataset

5.3.5 Δημιουργία τελικού dataset

Συνδυάζοντας το dataset που αναφέρθηκε προηγουμένως, και τις ιστορικές τιμές, του χρηματιστηριακού δείκτη S&P500, τις οποίες αποκτήσαμε από την ιστοσελίδα finance/yahoo.com, δημιουργήσαμε ένα τελικό dataset, το οποίο χρησιμοποιούμε για τα πειράματά μας. Το κάθε sample του dataset, περιλαμβάνει: ημερομηνία του event, μέσο διανύσματα νοήματος λέξεων (word embedding) 100 διαστάσεων για τα O1, P και O2 αντίστοιχα, τιμή ανοίγματος, τιμή κλεισίματος, υψηλότερη τιμή, χαμηλότερη τιμή, τιμή κλεισίματος, τιμή κλεισίματος επόμενης ημέρας, ποσοστιαία διαφορά τιμών κλεισίματος, ετικέτα (1 αν η τιμή κλεισίματος είναι μεγαλύτερη την επόμενη μέρα, 0 αν είναι μικρότερη). Για τα samples που δεν υπήρχε τιμή στην επόμενη τιμή κλεισίματος (δεν ήταν εργάσιμη η επόμενη ημέρα) συμπληρώσαμε τις αριθμητικές τιμές του με «-1». Έτσι το dataset μας περιείχε 1004

samples, από τα οποία μόνο τα 720 είχαν αριθμητικές τιμές διαφορετικές του -1. Παρακάτω η εικόνα του dataset σε αρχείο csv:

	A	B	C	D	E	F	…	KO	KP	KQ	KR	KS	KT	KU	KV
1		0	1	2	3			299	300	Open	High	Low	Close	Change	label
2	0	########	-0.05572	-0.2942	1.524013	-1.20		35464	-0.39299	1382.5	1388.61	1379.33	1385.72	0.005333	1
3	1	########	-1.74997	-0.17579	0.66493	-2.38		02314	-0.00023	1385.43	1388.92	1377.31	1378.33	-0.00186	0
4	2	########	-0.41167	-0.12329	0.700421	1.275		09721	-1.5784	1402.69	1407.89	1402.26	1406.09	0.003656	1
5	3	########	-2.65022	-0.34954	-1.97309	0.134		13486	0.744511	-1	-1	-1	-1	-1	-1
6	4	########	-2.9142	5.028496	-3.38901	-1.49		04428	0.11286	1427.08	1431.81	1420.65	1422.48	-0.00216	0
7	5	########	-1.63125	0.174616	-0.749	-1.36		47533	0.844413	1408.7	1415.99	1405.32	1414.85	-0.00634	0
8	6	########	0.071992	0.182265	0.225828	0.234		25049	0.940854	1430.59	1432.96	1424.21	1426.37	-0.0029	0
9	7	########	0	0	0			20634	-1.11903	1427.96	1440.14	1427.96	1440.13	0.01127	1
10	8	########	0.35058	1.03588	0.637621	-0.23		18337	1.410885	1428.65	1441.61	1424.78	1438.24	-0.00535	0
11	9	2/7/2007	1.26503	1.201032	1.110038	1.888		18573	3.109349	1447.41	1452.99	1446.44	1450.02	0.001179	1
12	10	########	-0.72656	-0.78925	1.965636	3.492		26936	4.142725	1438	1439.11	1431.44	1433.37	-0.0076	0
13	11	########	-1.22869	0.731312	-0.44881	-2.50		55964	-3.18945	1455.15	1457.97	1453.19	1456.81	0.000872	1
14	12	########	0.476351	0.609513	-1.22844	0.877		66548	0.3182	1406.23	1406.23	1377.71	1377.95	-0.00669	0
15	13	########	0.008591	-0.13536	-0.84836	0.639		27036	1.919311	1417.17	1426.24	1413.27	1422.53	0.001174	1
16	14	########	1.353248	0.593262	1.649266	1.525		35238	-3.74079	1494.21	1497.32	1488.67	1494.07	0.007831	1
17	15	########	1.244751	0.206223	-3.83886	-0.21		84703	-0.33379	1515.55	1521.8	1512.02	1518.11	-0.00798	0
18	16	########	-3.81537	0.882913	0.774483	0.06		0	0	1530.19	1535.56	1528.26	1530.62	-0.00374	0
19	17	########	-1.22578	1.269406	-1.31627	0.721		06414	0.953309	1507.64	1515.53	1503.35	1509.12	0.010682	1
20	18	########	-1.07289	0.127942	-1.23394	-0.88		47654	-0.67312	1492.62	1506.8	1484.18	1506.34	0.000418	1
21	19	7/6/2007	-0.62442	-0.38456	-0.27585	0.811		09624	0.358848	1524.96	1532.4	1520.47	1530.44	-0.00092	0
22	20	########	0.25904	1.469412	-0.29073	-0.67		44013	-0.0278	1549.2	1549.2	1533.67	1546.17	-0.00447	0
23	21	########	-0.23731	-0.79517	-1.56026	3.950		79133	-0.88322	1553.19	1553.19	1529.2	1534.1	-0.00487	0
24	22	########	-1.5938	1.059816	-1.39743	-1.32		23624	0.372917	1453.09	1462.02	1429.74	1453.64	0.000495	1
25	23	########	-0.20731	-0.10025	-0.29244	-0.02		39129	0.319266	1445.94	1451.75	1430.54	1445.55	-0.00109	0
26	24	########	-0.26274	1.197788	-3.23112	-1.33		16058	1.231697	1445.55	1455.32	1439.76	1447.12	-0.01171	0
27	25	########	-0.86808	1.521263	-0.41307	-0.88		59813	0.836939	1479.36	1479.36	1465.98	1466.79	0.023473	1
28	26	9/5/2007	0.965543	0.326081	2.06303	0.222		47121	-0.70675	1488.76	1488.76	1466.34	1472.29	-0.00425	0

Εικόνα 5-17 Παράδειγμα τελικού dataset 2ου μέρους

5.3.6 Μοντέλα Πρόβλεψης

Χρησιμοποιώντας το τελικό Dataset που περιγράφουμε παραπάνω, δημιουργήσαμε 3 input sets τα οποία στη συνέχεια χρησιμοποιήσαμε στα μοντέλα πρόβλεψης.

1. Αριθμητικές τιμές του SP500 (Num)

   ορίσαμε ως δεδομένα εισόδου τα: τιμές ανοίγματος (στήλη 301), κλεισίματος (στήλ 304), υψηλότερες (στήλη 302), χαμηλότερες (στήλη 303).

2. Word Embeddings (WB)

   Ορίσαμε ως δεδομένα εισόδου τις στήλες 1 ως 300.

3. Word Embeddings + Αριθμητικές τιμές του SP500 (WB_Num)

   Ορίσαμε ως δεδομένα εισόδου τις στήλες 1 ως 304.

```
#numeric
X = df.iloc[:,302:306]

#WB
X = df.iloc[:,2:302]

#WB+numeric
X = df.iloc[:,2:306]


y = df['label']
```

Εικόνα 5-18 Κώδικας: Ορισμός X & y

Έπειτα κανονικοποιήσαμε τις τιμές χρησιμοποιώντας τις λειτουργίες της κλάσης Preprocessing της βιβλιοθήκης sk-learn για να απαλείψουμε τυχών ανωμαλίες μεταξύ των feature στα input set όπως φαίνεται στην εικόνα:

```
# normalization
x = X.values #returns a numpy array
min_max_scaler = preprocessing.StandardScaler()
x_scaled = min_max_scaler.fit_transform(x)
X = pd.DataFrame(x_scaled)
```

Εικόνα 5-19 Κώδικας: Κανονικοποίηση

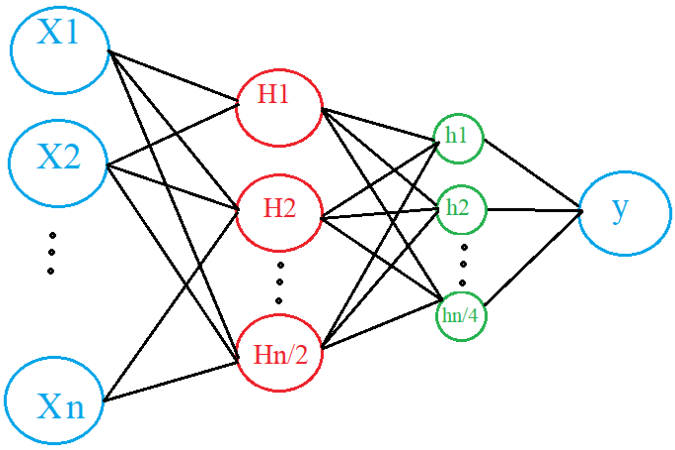
Εν συνεχεία, χωρίσαμε τα δεδομένα σε εκπαίδευσης και ελέγχου, με ποσοστά 80% και 20% αντίστοιχα ανακατεύοντας την σειρά των samples έτσι ώστε στη συνέχεια των πειραμάτων, τρέχοντας την εκπαίδευση για πολλά διαφορετικά «ανακατέματα» των samples να μπορέσουμε να βγάλουμε πιο υγιή συμπεράσματα.

```
#split into train and test
X_train, X_test, y_train, y_test = train_test_split(
                    X, y, test_size=0.20, random_state=43)
```

Εικόνα 5-20 Κώδικας: Train-Test Split

Το **νευρωνικό** δίκτυο αποτελείται από:

- 1 επίπεδο εισόδου (X) με αριθμό νευρώνων ίσο με τον αριθμό των features στα samples του κάθε ενός από τα Dataset που έχουμε δημιουργήσει, συνεπώς:
- 1 κρυφό επίπεδο (H) με αριθμό νευρώνων ίσο με τον αριθμό νευρώνων εισόδου διά 2.
- 1 κρυφό επίπεδο (h) με αριθμό νευρώνων ίσο με τον αριθμό νευρώνων εισόδου διά 4.
- 1 επίπεδο εξόδου (y) με αριθμό νευρώνων ίσο με 1.



Εικόνα 5-21 Νευρωνικό Δίκτυο 2ου μέρους

Για τα πειράματά μας παραμετροποιήσαμε το δίκτυο όπως φαίνεται παρακάτω.

Παραμετροποίηση Δικτύου:

Αντικειμενική συνάρτηση = “Binary Cross Entropy”

Optimizer = “adam”

Ρυθμός μάθησης = 0.01

Epochs = 4000

Batch size = 200

Early Stopping = monitor: “val_loss”, patience: 100

Check Point = Keep best “val_acc” model

Regularization = L1 + L2 Matrix

Συνάρτηση ενεργοποίησης :

- Κρυφά επίπεδα “ReLU”
- Επίπεδο εξόδου “Sigmoid”

Για την υλοποίηση του νευρωνικού δικτύου, χρησιμοποιήσαμε την βιβλιοθήκη Keras, η οποία χρησιμοποιεί την ήδη εγκατεστημένη στο σύστημά μας βιβλιοθήκη Tensorflow ως backend.

Παρακάτω παραθέτουμε τα σημαντικότερα κατά τη γνώμη μας κομμάτια κώδικα του νευρωνικού δικτύου:

```
import os
from keras.models import Sequential
from keras.layers import Dense
import pandas as pd
import numpy as np
from numpy import loadtxt
from keras.models import load_model
from keras.layers import Dropout
from keras.callbacks import EarlyStopping, ModelCheckpoint
from keras import regularizers
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
```

Εικόνα 5-22 Κώδικας: Imports 2ου μέρους

```
#-------------------------hyper params-------------------------------------

#input shape
input_neurons = np.shape(X)[1]
hidden_neurons = round(input_neurons/2)

#training
epochs = 4001
batch_size = 200

#callbacks
es = EarlyStopping(monitor='val_loss', mode='min',patience=20)
mc = ModelCheckpoint('best_model.h5', monitor='val_accuracy',
                     mode='max', verbose=1, save_best_only=True)

#regularization
kernel_reg = regularizers.l2(0.001)
activity_reg = regularizers.l1(0.001)
```

Εικόνα 5-23 Κώδικας: Υπερ-παράμετροι

```
#------------------------define and train model--------------------------



model = Sequential()
model.add(Dense(hidden_neurons, input_dim=input_neurons, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(round(hidden_neurons/2), input_dim=hidden_neurons, activation='relu',
                kernel_regularizer=kernel_reg,activity_regularizer=activity_reg))
model.add(Dense(1, activation='sigmoid'))


# compile the keras model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])


# callbacks=[es,mc]
# fit the keras model on the dataset
print('-------------------------------------------------------')
print(X_train[0:10])
print(y_train[0:10])
print('-------------------------------------------------------')

history = []
history = model.fit(X_train, y_train, epochs=epochs, batch_size=batch_size,
    validation_split=0.3, callbacks=[es,mc] )
```

Εικόνα 5-24 Κώδικας: Ορισμός Νευρωνικού δικτύου

5.3.7 Αποτελέσματα

Για την αξιολόγηση της απόδοσης τον συστημάτων μας το μέτρο που χρησιμοποιήσουμε είναι το

ACCURACY: Ποσοστό επιτυχίας στη πρόβλεψη κλάσης (UP/DOWN).

Έχοντας φτιάξει τα 3 input sets που περιγράψαμε παραπάνω, εκπαιδεύσαμε το δίκτυο 10 φορές για κάθε σετ, κάνοντας κάθε φορά διαφορετική κατανομή των samples «ανακάτεμα». Με αυτόν τον τρόπο μπορούμε να εξάγουμε καλύτερα και πιο ασφαλή συμπεράσματα για τις αποδόσεις των μοντέλων χρησιμοποιώντας τα στατιστικά στοιχεία των αποτελεσμάτων από τις 10 εκπαιδεύσεις για κάθε set.

Το σύστημά στο οποίο εκτελέσαμε τα πειράματά μας είναι ένας σταθερός υπολογιστής οικιακής χρήσης. Τα τεχνικά χαρακτηριστικά του είναι:

CPU: AMD RYZEN 7 1700X 3.80GHZ 8-CORE BOX

GPU: Gigabyte GeForce GTX1050 Ti 4GB G1 Gaming

RAM: 8gb ddr4

Επιπλέον για την εκπαίδευση του δικτύου χρησιμοποιούμε την έκδοση Tensorflow-gpu που χρησιμοποιεί την επεξεργαστική ισχύ της κάρτας γραφικών.

Για μπορέσουμε να εξάγουμε επιπλέον συμπεράσματα από τα πειράματά μας, παράλληλα εκπαιδεύσαμε επιπλέον 3 μοντέλα πρόβλεψης, τα οποία στηρίζονται στους αλγόριθμους Μηχανικής Μάθησης, «Random Forest», «Logistic Regression», και «Support Vector Machines (SVM)».

Τέλος ως Baseline χρησιμοποιήσαμε ένα υποτιθέμενο μοντέλο πρόβλεψης που προτείνεται από την έρευνα [5] και περιγράψαμε επίσης στο πρώτο μέρος της εργασίας, το οποίο επιλέγει ως πρόβλεψη την πλειοψηφία της UP ή DOWN(0/1) κλάσης στο test set.

Σημειώστε: ότι υπο κανονικές συνθήκες το μοντέλο δεν δύναται να γνωρίζει την κλάση πλειοψηφίας στο test set παρά μόνο στο training set (με κίνδυνο αυτή στη συνέχεια να αλλάξει στη συνέχεια, στα δεδομένα ελέγχου).

Χρησιμοποιώντας λοιπόν όλους τους πιθανούς συνδυασμούς, input set και αλγόριθμου, έχουμε τα παρακάτω τελικά μοντέλα:

- Αριθμητικές Τιμές σε αλγόριθμο Random Forest (NUM_RF)
- Αριθμητικές Τιμές σε αλγόριθμο Logistic Regression (NUM_Log_Reg)
- Αριθμητικές Τιμές σε αλγόριθμο SVM (NUM_SVM)
- Αριθμητικές Τιμές σε αλγόριθμο Neural Net (NUM_NN)

- Word Embeddings σε αλγόριθμο Random Forest (WB_RF)
- Word Embeddings σε αλγόριθμο Logistic Regression (WB_Log_Reg)
- Word Embeddings σε αλγόριθμο SVM (WB_SVM)
- Word Embeddings σε αλγόριθμο Neural Net (WB_NN)

- Word Embeddings + Αριθμητικές Τιμές σε αλγόριθμο Random Forest (WB_NUM_RF)
- Word Embeddings + Αριθμητικές Τιμές σε αλγόριθμο Logistic Regression(WB_NUM_Log_Reg)
- Word Embeddings + Αριθμητικές Τιμές σε αλγόριθμο SVM (WB_NUM_SVM)
- Word Embeddings + Αριθμητικές Τιμές σε αλγόριθμο Neural Net (WB_NUM_NN)

Εκπαιδεύοντας λοιπόν το δίκτυο, αλλά και τα υπόλοιπα 3 απλούστερα μοντέλα πρόβλεψης, για κάθε ένα από τα 3 input sets συνθέσαμε τους παρακάτω πίνακες αποτελεσμάτων, οι οποίοι δείχνουν την ακρίβεια που πέτυχε το κάθε μοντέλο στο test set σε κάθε ένα από τα 10 διαφορετικά Shuffles με τα οποία εκπαιδεύτηκε, καθώς και τον μέσο όρο ακρίβειας κάθε μοντέλου.

NUM (Numeric values)

Πίνακας 4 Αποτελέσματα 2ου μέρους για μοντέλο NUM

Shuffle seed	Random Forest	Logistic regression	SVM	NN	Baseline
26	47	53	49	53	53
41	53	54	54	54	54
42	54	57	57	57	57
57	55	54	54	54	54

43	52	57	54	57	57
16	59	59	59	59	59
25	49	53	53	53	53
38	56	60	58	60	60
62	54	53	53	53	53
70	55	55	55	55	55
Average	53.4	55.5	54.6	55.5	55.5

WB (Word embeddings)

Πίνακας 5 Αποτελέσματα 2ου μέρους για μοντέλο WB

Shuffle seed	Random Forest	Logistic regression	SVM	NN	Baseline
26	51	55	53	60	53
41	55	50	52	57	54
42	62	52	57	60	57
57	53	55	54	57	54
43	58	49	55	53	57
16	59	58	56	59	59
25	52	56	52	55	53
38	58	59	56	58	60
62	55	48	53	54	53
70	54	55	57	61	55
Average	55.7	53.7	54.5	57.4	55.5

WB_NUM (Word embeddings + Numeric values)

Πίνακας 6 Αποτελέσματα 2ου μέρους για μοντέλο WB_NUM

Shuffle seed	Random Forest	Logistic regression	SVM	NN	Baseline
26	53	55	53	59	53
41	55	51	53	55	54
42	60	54	56	62	57
57	55	56	53	55	54
43	55	49	54	55	57
16	58	58	56	54	59
25	54	56	53	53	53
38	59	58	57	56	60

62	54	49	53	53	53
70	55	55	57	60	55
Average	55.8	54.1	54.5	56.2	55.5

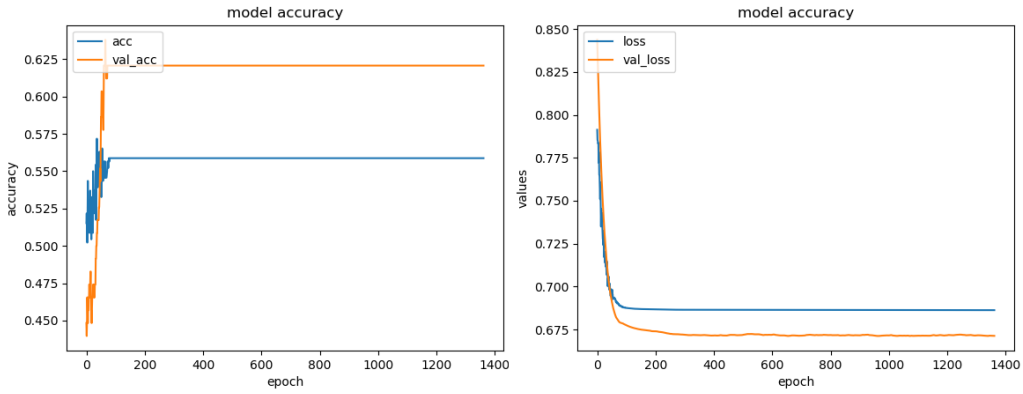
*Shuffle seed: ο αριθμός που δίνεται ως «σπόρος» στην rand συνάρτηση για να γίνει το ανακάτεμα (Shuffle) των samples.

Όπως είδαμε τα καλύτερα αποτελέσματα επιτυγχάνονται από το μοντέλο WB, το οποίο πετυχαίνει μέσο όρο ακρίβειας 57.4% ακολουθούμενα από αυτά του WB_NUM με 56.2% ενώ χαμηλότερο μέσο όρο πετυχαίνει το μοντέλο NUM.

Αν θα θέλαμε να ερμηνεύσουμε τα αποτελέσματα θα μπορούσαμε να πούμε ότι από τους παραπάνω πίνακες ένα από τα συμπεράσματα που εξάγουμε είναι το ότι φαίνεται πως η πληροφορία που εξάγεται από τα κείμενα ειδήσεων που περιγράψαμε είναι κατά πολύ χρησιμότερη από ότι οι απλές ημερήσιες αριθμητικές τιμές των μετοχών. Επίσης φαίνεται από τη διαφορά WB με WB_NUM πως το νευρωνικό δίκτυο παραπλανείται από την εξτρα πληροφορία που επιχειρούμε να εισάγουμε. Ωστόσο τα αποτελέσματα του WB_NUM_RF υποδεικνύουν ότι ίσως να υπαρχει πιθανότητα βελτίωσης του WB_NUM_NN μοντέλου. Τα αποτελέσματα του WB_NN μοντέλου είναι υποδιέστερα σε σχέση με αυτά που περιγράφονται στην έρευνα [25] για το αντίστοιχο μοντέλο. Αυτό είναι αναμενόμενο καθώς τα πειράματά μας πιθανώς να υστερούν σε παραμετροποίηση του δικτύου ή σε πληρότητα του λεξικού Word Embeddins που περιγράψαμε στην αρχή του κεφαλαίου. Τέλος παρατηρώντας τον 1ο πίνακα βλέπουμε ότι τα αποτελέσματα είναι πολύ αδύναμα σε σχέση με τους άλλους 2.

Παρακάτω παραθέτουμε κάποια διαγράμματα των εκπαιδεύσεων που θεωρήσαμε πιο αντιπροσωπευτικές για κάθε μοντέλο, τα οποία δείχνουν το πως εξελίσονται τα μέτρα “accuracy” και “loss” (binary crossentropy) στο training και στο validation set, ανάλογα με τον αριθμό των epochs.

NUM_NN

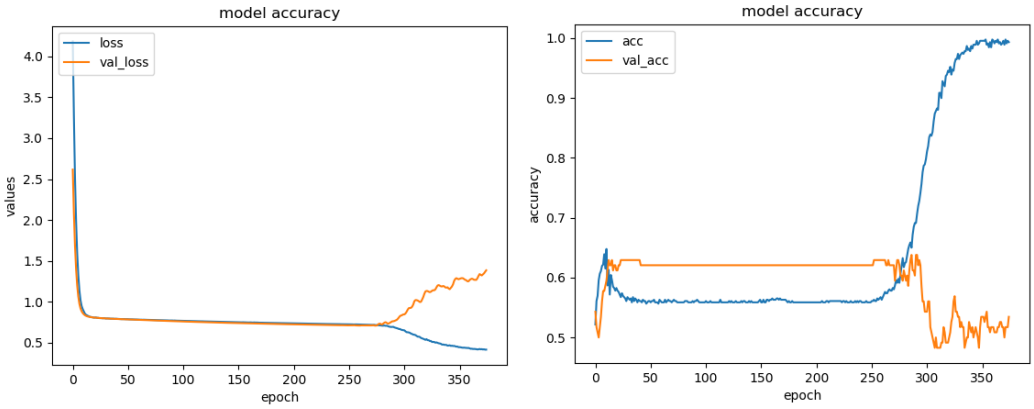


Εικόνα 5-25 Διαγράμματα accuracy & loss NUM_NN(53%)

Βλέπουμε στα παραπάνω γραφήματα, ότι παρόλο που στα πρώτα 100 epochs περίπου υπάρχει έντονη διακύμανση του accuracy και κατακόρυφη πτώση του loss. Στη συνέχεια όμως τα μεγέθη σταθεροποιούνται απότομα και δεν παρατηρείται άλλη αλλαγή μέχρις ότου η callback συνάρτηση

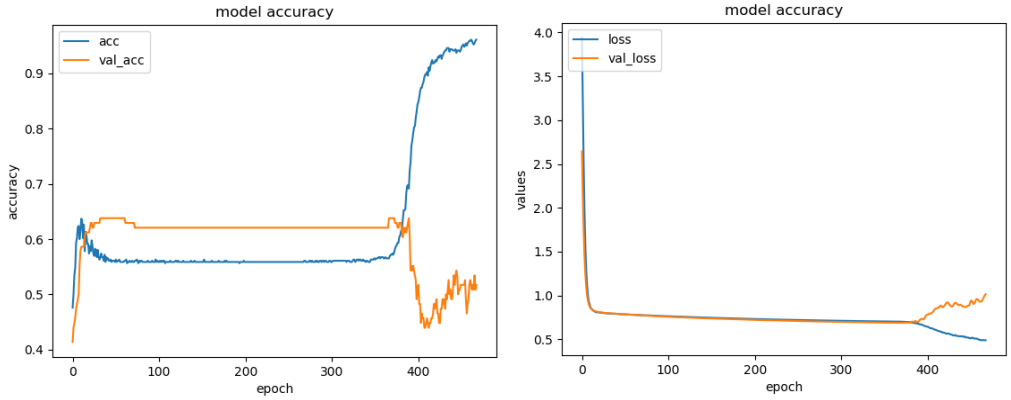
«Early stopping» σταματήσει το training. Από αυτή τη συμπεριφορά συμπεραίνουμε ότι η εκπαίδευση του δικτύου βρίσκει πολύ γρήγορα ένα τοπικό ελάχιστο στο χώρο αναζήτησης που ορίζεται από την αντικειμενική συνάρτηση(Binary cross entropy), από το οποίο δεν μπορεί να ξεφύγει. Το τοπικό ελάχιστο που φαίνεται να βρίσκει είναι η λύση του να επιλέγεται συνεχώς η κλάση πλειοψηφίας.

WB_NN



Εικόνα 5-26 Διαγράμματα accuracy & loss WB_NN(60%)

WB_NUM_NN



Εικόνα 5-27 Διαγράμματα accuracy & loss WB_NUM_NN(58%)

Αυτές οι περιπτώσεις γραφημάτων είναι πολύ διαφορετικές από τις προηγούμενες του μοντέλου NUM_NN. Εδώ βλέπουμε ότι και στις 2 περιπτώσεις, μετά την πρώτη διακύμανση το δίκτυο δείχνει να σταθεροποιείται, ωστόσο στη συνέχεια το accuracy στο training set αυξάνεται απότομα παρασύροντας μαζί του μέχρι ένα σημείο το accuracy στο validation set. Παρόλο που το δίκτυο δείχνει στη συνέχεια ότι πέφτει σε υπερτροφοδότηση, καθώς βέπουμε ότι η ψαλίδα μεταξύ train και validation accuracy μεγαλώνει απότομα με την callback συνάρτηση checkpoint, μας παρέχεται η δυνατότητα να επιλέξουμε το στιγμιότυπο του δικτύου με το μεγαλύτερο ποσοστό στο validation accuracy, το οποίο είναι μία κορυφή της πορτοκαλί γραμμής πάνω από τα 300 και

400 epochs αντίστοιχα. Με αυτή τη μέθοδο κατάφέρνουμε να επιτύχουμε ακρίβεια ίση με 60% στη συνέχεια, στο test set.

ΚΕΦΑΛΑΙΟ 6 - ΣΥΜΠΕΡΑΣΜΑΤΑ

Ανακεφαλαιώνοντας η παρούσα εργασία ως στόχο είχε αρχικά να διερευνήσει την διαφορά στην απόδοση ενός μοντέλου πρόβλεψης βαθιάς μάθησης το οποίο έχει ως δεδομένα εισόδου αριθμητικές τιμές μετοχής, όταν σε αυτό εισαχθούν δεδομένα υψηλής και χαμηλής συχνότητας. Μετά από μια σειρά πειραμάτων τα οποία ήταν και σε μεγάλο βαθμό συμβατά με τα πειράματα της έρευνας [5] έτσι ώστε να μπορούν να γίνουν συγκρίσεις, το συμπέρασμα που αποκομίσαμε ήταν ότι τα δεδομένα χαμηλής συχνότητας (ανά μία μέρα) αυτού του είδους, υστερούν πολύ σε προβλεπτική δύναμη από τα δεδομένα υψηλής (ανά 5 λεπτά).

Όπως αναφέραμε και προηγουμένως αυτό το γεγονός ίσως να οφείλεται στην αδυναμία του ανθρώπου να επεξεργάζεται αρκετά δεδομένα και να δρα αναλόγως στο επίπεδο συχνότητας των 5 λεπτών. Αυτό σημαίνει ότι σε μεγάλες συχνότητες ο ανθρώπινος (αλλά και γενικά ο εξωγενής) παράγοντας έχει μικρότερη επιρροή στη συμπεριφορά του συστήματος. Αυτό με τη σειρά του σημαίνει ότι τα δεδομένα ενδογενών παραγόντων είναι υπαίτια της συμπεριφοράς του συστήματος σε μεγαλύτερο βαθμό και κατ' επέκταση έχουν μεγαλύτερη προβλεπτική δύναμη.

Ένας άλλος πιθανώς λόγος για την αδυναμία πρόβλεψης που παρατηρήθηκε στα συστήματα του πρώτου μέρους της εργασίας είναι και η έλλειψη όγκου στα δεδομένα, αφού κάθε χρόνος έχει 250 εργάσιμες, πράγμα που σημαίνει ότι ακόμα και σε 10 χρόνια καταγραφής των τιμών μιας μετοχής, το dataset θα περιέχει μόλις 2500 samples.

Προχωρώντας στο δεύτερο κομμάτι της εργασίας ο παραπάνω ισχυρισμός μας για την αδυναμία των εξωγενών παραγόντων στα δεδομένα υψηλής συχνότητας δυναμώνει, αφού μετά την ολοκλήρωση των πειραμάτων μας, παρατηρήσαμε ότι τα δεδομένα εισόδου που δημιουργούνται από πληροφορίες εξωγενών παραγόντων (ειδησεογραφικά κείμενα οικονομικού ενδιαφέροντος) πετυχαίνουν σταθερά μεγαλύτερη ακρίβεια πρόβλεψης στην κατεύθυνση κίνησης της αγοράς.

Τελειώνοντας πιστεύουμε ότι η απόδοση των μοντέλων WB_NN και WB_NUM_NN που παρουσιάσαμε μπορεί εύκολα να βελτιωθεί με μια πιο προσεκτική παραμετροποίηση του δικτύου, ή με τη βελτίωση του Word Embedding λεξικού που περιγράψαμε στα προηγούμενα κεφάλαια.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1]Anthony Fader, Michael Schmitz, Robert Bart, Stephen Soderland, Oren Etzioni. *Identifying Relations for Open Information Extraction.* Edinburgh, 2011.

[2]Baldi, Pierre. «Autoencoders, Unsupervised Learning, and Deep Architectures.» *JMLR: Workshop and Conference Proceedings* 2012.

[3]Malkiel, BG. «A random walk down Wall Street. W. W.» *Norton & Co*, 1973.

[4]Christopher Krauss, Xuan Anh Do, Nicolas Huck. «Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500.» *European Journal of Operational Research*, 2016.

[5]Eunsuk Chong, Chulwoo Han, Frank C. Park. «Deep learning networks for stock market analysis and prediction: Methodology, data presentation and case studies.» *Elsevier*, 2017.

[6]Géron, Aurélien. *Hands-On MachineLearning with Scikit-Learn and TensorFlow.* United States of America: O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2017.

[7]HECHT-NIELSEN, ROBERT. «Theory of the Backpropagation Neural Network.» 1992.

[8]Hugo Larochelle, Michael Mandel, Razvan Pascanu, Yoshua Bengio. «Learning Algorithms for the Classification Restricted Boltzmann Machine.» 2012.

[9]J. G. Agrawal, Dr. V. S. Chourasia, Dr. A. K. Mittra. «State-of-the-Art in Stock Prediction Techniques .» 2013.

[10]Konstandinos Chourmouziadis, Prodromos D. Chatzoglou. «An intelligent short term stock trading fuzzy system for assisting investors in portfolio management.» *Expert Systems With Applications*, 2015.

[11]Luss, Ronny & d'Aspremont, Alexandre. «Predicting Abnormal Returns From News Using Text Classification.» 2012.

[12]Miller, George A. «WordNet: a lexical database for English.» *Communications of the ACM*, 1995.

[13]Mitchell, Tom M. *Machine Learning.* 1997.

[14]Rabiul Islam Jony, Rakibul Islam Rony Musfiqur Rahman, Abiduzzaman Rahat. «Big Data Characteristics, Value Chain and Challenges .» 2016.

[15]Hochreiter, S. «The vanishing gradient problem during learning recurrent neural nets and problem solutions.» *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 1998.

[16]Ringnér, Markus. «What is principal component analysis?» *Nature biotechnology* ,2008.

[17]Rydning, David Reinsel – John Gantz – John. «The Digitization of the World From Edge to Core.» *Framingham: International Data Corporation* 2018.

[18]Schmidhuber, Sepp Hochreiter and Jürgen. «Long Short-Term Memory.» *Neural computation* 1997.

[19]Thomas Fischer, Christopher Krauss. «Deep learning with long-short memory networks for financial market predictions.» *Elsevier*, 2017.

[20]Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. «Efficient Estimation of Word Representations in Vector Space.» 2013.

[21]Vinod Nair, Geoffrey E. Hinton. «Rectified Linear Units Improve Restricted Boltzmann Machines.» 2010.

[22]Wen-Chyuan Chiang, David Enke , Tong Wu , Renzhong Wang. «An adaptive stock index trading decision support system.» 2016 .

[23]Xiao Zhong, David Enke. «Forecasting daily stock market return using dimensionality reduction.» *Expert Systems With Applications*, 2016.

[24]XiaoDing, Yue Zhang, Ting Li, Junwen Duan. «UsingStructuredEventstoPredictStockPriceMovement: AnEmpiricalInvestigation .» *Association for Computational Linguistics* 2014.

[25]XiaoDing, YueZhang, TingLiu, JunwenDuan. «DeepLearningforEvent-DrivenStockPrediction .» *International Joint Conference on Artificial Intelligence* 2015.

[26]I Goodfellow, Y Bengio, A Courville. *Deep learning.* 2016.

[27]Herculano-Houzel, S. «Do you know your brain? A survey on public neuroscience literacy at the closing of the decade of the brain.» *The Neuroscientist*, 2002.

[28]David Enke, Suraphan Thawornwong. «The use of data mining and neural networks for forecasting stock market returns.» *Expert Systems with applications*2005.

[29]Graves, A. «Generating sequences with recurrent neural networks.» *arxiv.org,*2013.

[30]Andrew M. Dai, Christopher Olah, Quoc V. Le. «Document Embedding with Paragraph Vectors.» 2015.

[31]D Enke, N Mehdiyev. «Stock Market Prediction Using a Combination of Stepwise Regression Analysis, Differential Evolution-based Fuzzy Clustering, and a Fuzzy Inference Neural Network.» *Intelligent Automation & Soft Computing* 2013.

[32]STA Niaki, S Hoseinzade. «Forecasting S&P 500 index using artificial neural networks and design of experiments.» *Journal of Industrial Engineering International*2013.

[33]J Patel, S Shah, P Thakkar, K Kotecha. «Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques.» *Expert Systems with Applications*2015.

[34]T Chen, F Chen. «An intelligent pattern recognition model for supporting investment decisions in stock market.» *Information Sciences*2016.