



# ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Πανεπιστήμιο Αιγαίου  
Σχολή Θετικών Επιστημών  
Πρόγραμμα Μεταπτυχιακών Σπουδών:  
Στατιστική και Αναλογιστικά-Χρηματοοικονομικά Μαθηματικά

Νευρωνικά Δίκτυα και Εφαρμογές  
Εφαρμογή Τεχνητών Νευρωνικών Δικτύων και Στατιστικών  
Μοντέλων στην Πρόβλεψη Χρονοσειρών

Διπλωματική Εργασία  
του  
Δημήτριου Πήτα

**Επιβλέπων:** Ξανθόπουλος Στυλιανός  
Αναπληρωτής Καθηγητής Πανεπιστημίου Αιγαίου

**Μέλη Επιτροπής:** Νάστου Παναγιώτης  
Επίκουρος Καθηγητής Πανεπιστημίου Αιγαίου  
Χατζησπύρος Σπυρίδων  
Αναπληρωτής Καθηγητής Πανεπιστημίου Αιγαίου

Καρλόβασι, 06 Οκτωβρίου 2020



# Περιεχόμενα

<b>1</b>	<b>Εισαγωγή</b>	<b>5</b>
<b>2</b>	<b>Μηχανική Μάθηση και Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ)</b>	<b>7</b>
2.1	Εισαγωγή στην Μηχανική Μάθηση . . . . .	7
2.2	Μηχανική Μάθηση και Τεχνητά Νευρωνικά Δίκτυα . . . . .	8
2.3	Εισαγωγή στα Τεχνητά Νευρωνικά Δίκτυα . . . . .	8
2.3.1	Ιστορική Αναδρομή . . . . .	8
2.3.2	Βιολογικά Νευρωνικά Δίκτυα . . . . .	9
2.3.3	Το Μοντέλο του Τεχνητού Νευρώνα . . . . .	10
2.4	Συνάρτηση Ενεργοποίησης . . . . .	11
2.5	Αρχιτεκτονικές Τεχνητών Νευρωνικών Δικτύων . . . . .	14
2.5.1	Δίκτυα Πρόσθιας Τροφοδότησης Ενός Επιπέδου . . . . .	14
2.5.2	Πολυεπίπεδα Δίκτυα Πρόσθιας Τροφοδότησης . . . . .	15
2.5.3	Αναδρομικά Νευρωνικά Δίκτυα . . . . .	16
2.6	Το Μοντέλο του Στοιχειώδους Αισθητήρα (Perceptron) . . . . .	17
2.7	Κανόνας Μάθησης Perceptron . . . . .	18
2.8	Πολυεπίπεδο Δίκτυο Perceptron . . . . .	19
<b>3</b>	<b>Ανάλυση Χρονοσειρών</b>	<b>21</b>
3.1	Εισαγωγή στις Χρονοσειρές . . . . .	21
3.2	Βασικά Χαρακτηριστικά Χρονοσειρών . . . . .	22
3.3	Στατιστικά Μοντέλα Πρόβλεψης Στάσιμων Χρονοσειρών . . . . .	25
3.4	Στατιστικά Μοντέλα Πρόβλεψης Μη-Στάσιμων Χρονοσειρών . . . . .	26
3.5	Τρόποι Επιλογής του Καταλληλότερου Στατιστικού Μοντέλου . . . . .	27
<b>4</b>	<b>Περιγραφή της Εφαρμογής</b>	<b>29</b>
4.1	Υπάρχοντα Μοντέλα Πρόβλεψης με Χρήση Τεχνητών Νευρωνικών Δικτύων . . . . .	29
4.2	Πολυεπίπεδο Τεχνητό Νευρωνικό Δίκτυο . . . . .	30
4.3	Στατιστικά Μοντέλα . . . . .	34
4.3.1	ARMA . . . . .	34
4.3.2	ARIMA . . . . .	36
4.3.3	SARIMA . . . . .	40
4.4	Κώδικας Της Εργασίας . . . . .	44
4.4.1	Κώδικας Τεχνητού Νευρωνικού Δικτύου . . . . .	44
4.4.2	Κώδικας ARMA . . . . .	46
4.4.3	Κώδικας ARIMA . . . . .	47
4.4.4	Κώδικας SARIMA . . . . .	49



# Κεφάλαιο 1

## Εισαγωγή

Τα συστήματα λήψης αποφάσεων (decision making systems) είναι μια κατηγορία πληροφοριακών συστημάτων (information systems), η οποία υποστηρίζει εταιρείες και οργανισμούς στην διαδικασία λήψης αποφάσεων. Τα συστήματα αυτά διαχωρίζονται σε ανθρωποειδή (human-powered), αυτοματοποιημένα (automated) και σε ημι-αυτοματοποιημένα (semi-automated), ανάλογα με το αν η μηχανή ή ο άνθρωπος λαμβάνει την τελική απόφαση [32]. Επίκεντρο στην λήψη της ορθότερης απόφασης αποτελεί η συλλογή, η επεξεργασία και η ανάλυση δεδομένων. Γενικότερα, τα δεδομένα φωτίζουν ορισμένες άγνωστες πτυχές και παράγοντες που πολλές φορές είναι καθοριστικές στην λήψη αποφάσεων. Μέ την χρήση και την αξιοποίηση των δεδομένων και των πληροφοριών που αυτά παρέχουν δημιουργήθηκαν μοντέλα τα οποία κατέστησαν εφικτή την δημιουργία προβλέψεων με μεγαλύτερη εγγυρότητα. Μερικά τέτοια μοντέλα πρόβλεψης είναι τα τεχνητα νευρωνικά δίκτυα και τα στατιστικά μοντέλα. Έτσι, καταλήγουμε στο συμπέρασμα ότι με την χρήση και την αξιοποίηση των δεδομένων παρέχονται εγγυρότερες προβλέψεις, οι οποίες με την σειρά τους αποτελούν αρωγό των εταιρειών και των επιχειρήσεων στην λήψη αποφάσεων.

Η παρούσα εργασία πραγματεύεται τα δύο είδη μοντέλων πρόβλεψης που αναφέρθηκαν παραπάνω, δηλαδή των τεχνητών νευρωνικών δικτύων και των στατιστικών μοντέλων (ARMA, ARIMA, SARIMA), ενώ υλοποιείται μια εφαρμογή με σκοπό την δημιουργία προβλέψεων με βάση τα παραπάνω μοντέλα πρόβλεψης. Πιο συγκεκριμένα, στο δεύτερο κεφάλαιο παρατίθεται μια ιστορική αναδρομή των τεχνητών νευρωνικών δικτύων. Επιπρόσθετα, περιγράφεται η ιδέα, η δομή, η λειτουργία και οι αρχιτεκτονικές των τεχνητών νευρωνικών δικτύων, ενώ δίνεται ιδιαίτερη βάση στο μοντέλο του απλού αισθητήρα (Perceptron). Επίσης, παρουσιάζεται ο αλγόριθμος εκπαίδευσης του μοντέλου του αισθητήρα και τα πολυεπίπεδα δίκτυα Perceptron. Στο τρίτο κεφάλαιο περιγράφεται η έννοια της χρονοσειράς και τα βασικά χαρακτηριστικά της, ενώ αναλύονται τα στατιστικά μοντέλα ARMA, ARIMA, SARIMA και οι τρόποι επιλογής του καταλληλότερου μοντέλου για κάθε σειρά δεδομένων. Τέλος, στο τέταρτο κεφάλαιο σκιαγραφείται η ιδέα της εφαρμογής, σύμφωνα με την οποία σκοπός της εφαρμογής είναι να πραγματοποιηθούν προβλέψεις για επιλεγμένα σύνολα δεδομένων. Για τις προβλέψεις αυτές χρησιμοποιήθηκαν τέσσερα διαφορετικά σύνολα δεδομένων, όπου στο κάθε σύνολο δεδομένων εφαρμόστηκε ένα από τα μοντέλα πρόβλεψης που πραγματεύεται η εργασία, δηλαδή των τεχνητών νευρωνικών δικτύων και των τριών στατιστικών μοντέλων. Το τεχνητό νευρωνικό δίκτυο που επιλέχθηκε για την δημιουργία προβλέψεων είναι το πολυεπίπεδο δίκτυο του αισθητήρα. Κλείνοντας το κεφάλαιο παρουσιάζεται ο κώδικας που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής.



## Κεφάλαιο 2

# Μηχανική Μάθηση και Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ)

### 2.1 Εισαγωγή στην Μηχανική Μάθηση

Η μάθηση είναι μία από τις θεμελιώδεις ιδιότητες κάθε νοήμων οργανισμού και ιδιαίτερα του ανθρώπου. Από βρεφική ηλικία και για όλη του την ζωή, ο άνθρωπος αναζητάει συνεχώς την γνώση. Έτσι, μέσα από την επανάληψη και τις εκφάνσεις της καθημερινότητας κατορθώνει να την κατακτήσει. Η ανάγκη του ανθρώπου να μαθαίνει και να βελτιώνεται σε συνδυασμό με την ανάπτυξη της τεχνολογίας και τις απαιτήσεις της αγοράς για εξαγωγή πληροφορίας από δεδομένα μεγάλου όγκου, οδήγησε στην δημιουργία ενός νέου κλάδου της επιστήμης των υπολογιστών, ο οποίος είναι γνωστός ως Μηχανική Μάθηση (Machine Learning).

Με την εξέλιξη της μηχανικής μάθησης ως επιστημονικού κλάδου, της αποδόθηκαν διαχρονικά ποικίλοι ορισμοί με βάση την θέωρηση και οπτική, που σημαντικοί επιστήμονες ανέπτυξαν για αυτήν. Έτσι, το 1959 ο Arthur Samuel ορίζει την μηχανική μάθηση ως “Το πεδίο μελέτης που δίνει στους υπολογιστές την δυνατότητα να μαθαίνουν χωρίς να έχουν προγραμματιστεί ρητά” [2]. Μέσα από αυτό τον ορισμό, γίνεται κατανοητή η ικανότητα των μηχανών να μαθαίνουν με την επανάληψη και τα παραδείγματα, όπως ακριβώς ένας άνθρωπος μαθαίνει με την εμπειρία.

Στο σημείο αυτό, η έννοια της μάθησης των υπολογιστών με την εμπειρία, αποτελεί κλειδί για την διατύπωση ενός νέου πιο επίσημου ορισμού της μηχανικής μάθησης από τον Tom M. Mitchell (1997), σύμφωνα με τον οποίο: “Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από την εμπειρία  $E$  ως προς κάποια κλάση εργασιών  $T$  και μέτρο απόδοσης  $P$ , αν η απόδοση του σε εργασίες της κλάσης  $T$ , όπως αποτιμάται από το μέτρο  $P$ , βελτιώνεται με την εμπειρία  $E$ ” [1]. Με αυτό τον ορισμό ο Tom M. Mitchell δημιουργεί τις βάσεις για τον καθορισμό της Μηχανικής Μάθησης σε βασικό λειτουργικό πλαίσιο, χωρίς την χρήση γνωστικών όρων. Έτσι, με απλά λόγια λέμε ότι ένας υπολογιστής γίνεται πιο αποδοτικός και ότι μαθαίνει στηριζόμενος στην εμπειρία που αποκτάει μέσα από τους αλγόριθμους εκπαίδευσης του.

Η Μηχανική Μάθηση με βάση τον τρόπο με τον οποίο κάποιος υπολογιστής μαθαίνει και εκπαιδεύεται, διακρίνεται σε τρεις κατηγορίες:

- 1. Επιβλεπόμενη Μηχανική Μάθηση (Supervised Learning):** Στην επιβλεπόμενη μηχανική μάθηση, στον εκπαιδευόμενο υπολογιστή παρέχεται ένα σύνολο δεδομένων, το οποίο περιέχει τα δεδομένα εισόδου (inputs) και τα επιθυμητά δεδομένα εξόδου (outputs). Το σύνολο αυτό έχει τον ρόλο του “δάσκαλου” για τον εκπαιδευόμενο υπολογιστή, σύμφωνα με το οποίο ο υπολογιστής προσαρμόζεται και τελικά μαθαίνει. Πιο συγκεκριμένα, κατά την διαδικασία της εκπαίδευσης του, ο υπολογιστής δέχεται τις εισόδους και τις επιθυμητές εξόδους με στόχο να μάθει έναν γενικό αλγόριθμο, ο οποίος θα παράγει αποτελέσματα όσο το δυνατόν πιο κοντά στις επιθυμητές τιμές εξόδου [3]. Η επιβλεπόμενη μηχανική μάθηση διακρίνεται σε δύο κατηγορίες εφαρμογών, της Ταξινόμησης (Classification) και της Παλινδρόμησης (Regression). Στις εφαρμογές ταξινόμησης ο υπολογιστής εκπαιδεύεται να ταξινομεί σε κατηγορίες με βάση τα χαρακτηριστικά που του προσφέρονται από το σύνολο δεδομένων, ενώ στις εφαρμογές παλινδρόμησης εκπαιδεύεται να προβλέπει μελλοντικές τιμές. Μερικές ευρέως χρησιμοποιούμενες μέθοδοι επιβλεπόμενης μηχανικής μάθησης είναι τα τεχνητά νευρωνικά δίκτυα (Artificial Neural Networks), η γραμμική παλινδρόμηση (Linear Regression) και τα δέντρα αποφάσεων (Decision Trees).
- 2. Μη Επιβλεπόμενη Μηχανική Μάθηση (Unsupervised Machine Learning):** Στην μη επιβλεπόμενη μηχανική μάθηση, στον εκπαιδευόμενο υπολογιστή παρέχονται μόνο τα δεδομένα εισόδου (Inputs).

Έτσι, σε αντίθεση με την επιβλεπόμενη μηχανική μάθηση δεν υπάρχει η έννοια του “δάσκαλου”, όπου ο υπολογιστής στηριζόμενος στα δεδομένα εξόδου που του δίνονται, μαθαίνει. Σε αυτού του είδους την μάθηση, ο υπολογιστής προσπαθεί να αναγνωρίσει “μοτίβα” και δομές των δεδομένων εισόδου με σκοπό να εξάγει πληροφορίες και συμπεράσματα για αυτά [3]. Η επιβλεπόμενη μηχανική μάθηση διακρίνεται σε δύο κατηγορίες εφαρμογών, της Ομαδοποίησης (Clustering) και της Σχέσης (Association). Στις εφαρμογές ομαδοποίησης ο υπολογιστής προσπαθεί να ομαδοποιήσει τα δεδομένα σε κλάσεις, ενώ στις εφαρμογές σχέσης προσπαθεί να ανακαλύψει κανόνες, οι οποίοι περιγράφουν τα δεδομένα εισόδου. Μερικές μέθοδοι μη επιβλεπόμενης μηχανικής μάθησης είναι ο K-Means και ο Apriori.

- 3. Ενισχυτική Μάθηση (Reinforcement Learning):** Στην ενισχυτική μάθηση ο υπολογιστής δεν καθοδηγείται από κάποιον εκπαιδευτή, αλλά είναι ελεύθερος να ανακαλύψει τις ενέργειες, που θα του αποφέρουν το επιθυμητό αποτέλεσμα, μέσα από την αλληλεπίδραση του με το περιβάλλον. Η μέθοδος μάθησης αυτή είναι εμπνευσμένη από τις τεχνικές μάθησης με επιβράβευση και τιμωρία, που συναντώνται στην εκπαίδευση των έμβιων όντων (π.χ. σκύλος). Μία από τις πιο χαρακτηριστικές εφαρμογές Ενισχυτικής Μάθησης είναι τα συστήματα αυτόματης οδήγησης, όπου αυτοκίνητα μετακινούνται χωρίς κάποιος να τα οδηγεί.

Συνοψίζοντας, καταλήγουμε στο συμπέρασμα ότι η μηχανική μάθηση αποτελεί έναν καινοτόμο και σύγχρονο επιστημονικό πεδίο, το οποίο προσπαθεί να δώσει απάντηση στο ερώτημα του πατέρα της επιστήμης των υπολογιστών Alan Turing “Αν μια μηχανή μπορεί να πράξει, όπως ένα νοήμον όν μπορεί”.

## 2.2 Μηχανική Μάθηση και Τεχνητά Νευρωνικά Δίκτυα

Η μηχανική μάθηση προσφέρει τεχνικές επίλυσης προβλημάτων προσανατολισμένες στα δεδομένα (Data Driven), οι οποίες κατά κύριο λόγο βασίζονται στην ικανότητα των συστημάτων να προσαρμόζονται στα δεδομένα και όχι στην διαίσθηση των αναλυτών. Ο συνδυασμός του εύρους των ικανοτήτων και η αποτελεσματικότητα των εφαρμογών της μηχανικής μάθησης, αλλά και η αδυναμία των αυστηρά αλγοριθμικών μεθόδων να αντιμετωπίσουν προβλήματα υψηλότερου επιπέδου, όπως η ταξινόμηση, οδήγησε την επιστημονική κοινότητα και την αγορά να επενδύσουν στην περαιτέρω ανάπτυξη του κλάδου αυτού.

Μία ευρέως χρησιμοποιημένη μέθοδος μηχανικής μάθησης είναι τα Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ), τα οποία αποτελούν μια απόπειρα προσέγγισης των βιολογικών νευρωνικών δικτύων και της λειτουργίας του ανθρώπινου εγκεφάλου. Στόχος των τεχνητών νευρωνικών δικτύων είναι να συνδυάσουν την ικανότητα των υπολογιστών να εκτελούν γρήγορα και με ακρίβεια σύνθετες υπολογιστικές πράξεις, όπως επίσης και να αποθηκεύουν δεδομένα τεράστιου όγκου, με την ικανότητα του ανθρώπου να αναγνωρίζει αποτελεσματικά αντικείμενα και καταστάσεις. Τα νευρωνικά δίκτυα χρησιμοποιούνται για τη βελτίωση της διαδικασίας λήψης αποφάσεων από τους αναλυτές και απαντώνται σε ένα ευρύ φάσμα επιστημονικών περιοχών για την επίλυση προβλημάτων ταξινόμησης και πρόβλεψης. Μερικές αξιοσημείωτες περιοχές εφαρμογής των ΤΝΔ είναι τα Χρηματοοικονομικά, η Ιατρική, η Βιολογία και η Περιβαλλοντολογία. Στο σημείο αυτό αξίζει να σημειωθεί ότι τα ΤΝΔ μέσα από την απήχηση και την αποτελεσματικότητά τους στην αντιμετώπιση προβλημάτων λήψης αποφάσεων παρουσίασαν διαχρονικά ιδιαίτερη εξέλιξη. Έτσι, η ανάγκη για συνεχή βελτίωση τους, οδήγησε σε ποικίλους αλγόριθμους εκπαίδευσης και αρχιτεκτονικές, χαρακτηριστικά που θα μελετήσουμε αργότερα στην εργασία αυτή.

## 2.3 Εισαγωγή στα Τεχνητά Νευρωνικά Δίκτυα

### 2.3.1 Ιστορική Αναδρομή

Η ραγδαία ανάπτυξη των ηλεκτρονικών υπολογιστών σε συνδυασμό με την ανάγκη εξαγωγής πληροφοριών από μεγάλο όγκο δεδομένα, οδήγησε στην σταδιακή ανάπτυξη των τεχνητών νευρωνικών δικτύων, η οποία όμως γνώρισε την άνθιση μετά το 1980. Το πρώτο κύμα ενδιαφέροντος για τα νευρωνικά δίκτυα δημιουργήθηκε το 1943, όταν ο νευροφυσιολόγος Warren McCulloch και ο πρωτοετής φοιτητών Μαθηματικών Walter Pitts παρουσίασαν το πρώτο μοντέλο νευρωνικού δικτύου. Το μοντέλο McCulloch-Pitts περιγράφει πως λειτουργούν οι βασικές λειτουργίες του εγκεφάλου. Πιο συγκεκριμένα, παρουσιάζει πως απλά ηλεκτρικά κυκλώματα, τα οποία συνδέουν γραμμικές συναρτήσεις κατωφλιού (Linear Threshold functions), μπορούν να υλοποιούν λογικές πράξεις [5]. Με την εργασία τους αυτή οι ερευνητές προτείνουν τον νευρώνα ως βασική μονάδα ενός νευρωνικού δικτύου και εισάγουν για πρώτη φορά την έννοια της μνήμης στους υπολογιστές με την χρήση κλειστών διαδρομών του σήματος μέσα στο δίκτυο.

Το 1949 ο Donald Hebb με το βιβλίο του “The Organization of Behavior” υποστήριξε ότι η πληροφορία μπορεί να αποθηκευτεί μέσα στις συνδέσεις του δικτύου και εισήγαγε τον πρώτο κανόνα μάθησης (Learning Rule) για την μεταβολή των βαρών των συνδέσεων του δικτύου [6] [7]. Μερικά χρόνια αργότερα ο Minsky (1954) κατασκευάζει



τους πρώτους “νευρωνικούς υπολογιστές”, οι οποίοι είναι ικανοί να μεταβάλλουν τα βάρη των συνδέσεων τους αυτόματα.

Αποκορύφωμα όλων των παραπάνω εργασιών αποτέλεσε η εφεύρεση του μοντέλου του απλού αισθητήρα (Perceptron) από τον Frank Rosenblatt (1958). Το μοντέλο Perceptron μπορούσε μεταβάλλοντας τις τιμές των βαρών των συνδέσεων του, να εκπαιδευτεί στην ταξινόμηση σε κατηγορίες συγκεκριμένων παραδειγμάτων. Η σημαντικότητα του μοντέλου του απλού αισθητήρα στην εξέλιξη των Τεχνητών Νευρωνικών Δικτύων σημειώνεται στο γεγονός ότι έθεσε την βάση για τους αλγόριθμους μάθησης των μηχανών.

Στις αρχές της δεκαετίας του '60 οι Widrow και Hoff παρουσιάζουν την διαδικασία μάθησης “Ελάχιστου Μέσου Τετραγώνου του Σφάλματος”, γνωστή και ως κανόνας Δέλτα. Η μέθοδος μάθησης αυτή αποτελεί μια γενίκευση του κανόνα μάθησης του μοντέλου του απλού αισθητήρα και χρησιμοποιήθηκε στα μοντέλα ADALINE και MADALINE, που αναπτύχθηκαν εκείνη την περίοδο και χρησιμοποιούνταν για ταξινόμηση και για την πρόβλεψη του καιρού.

Το 1969 οι Minsky και Papert δημοσιεύουν το βιβλίο τους με τίτλο “The Perceptron”, στο οποίο αποδεικνύουν μαθηματικά ότι τα Τεχνητά Νευρωνικά Δίκτυα ενός επιπέδου τύπου απλού αισθητήρα παρουσιάζουν σημαντικούς περιορισμούς στην επίλυση συγκεκριμένων προβλημάτων [8]. Για το επόμενο 20 χρόνια οι περιορισμοί και οι δυσκολίες που δημιούργησαν αυτοί, απέτρεψαν πληθώρα ερευνητών να μελετήσουν τα Τεχνητά Νευρωνικά Δίκτυα.

Η αναγέννηση για τον κλάδο των τεχνητών νευρωνικών δικτύων ήρθε την δεκαετία του '80 μέσα από τις δημοσιεύσεις του John Hopfield και την μαθηματική απόδειξη του ότι ένα τεχνητό νευρωνικό δίκτυο πολλών επιπέδων μπορεί να αποθηκεύσει οποιαδήποτε πληροφορία. Στις δημοσιεύσεις του ο Hopfield έδωσε την θεωρητική θεμελίωση των δικτύων ανατροφοδότησης (Recurrent Neural Networks). Σημαντική πτυχή για την εξέλιξη των Τεχνητών Νευρωνικών Δικτύων αποτέλεσε η δημοσίευση του βιβλίου “Parallel Distributed Processing” από τους Rumelhart-McClelland το 1986 και η παρουσίαση μέσα σε αυτό του αλγόριθμου οπισθοδιάδοσης (Back-propagation Algorithm) [9]. Αποκορύφωμα της ανακάλυψης του αλγόριθμου οπισθοδιάδοσης αποτέλεσε η δημοσίευση του άρθρου με τίτλο “Learning Representations by Back-propagation Errors” από τους Hint, Rumelhart και Williams (1986), στο οποίο έδειξαν ότι τα νευρωνικά δίκτυα με πολλά κρυμμένα στρώματα μπορούν να εκπαιδευτούν αποτελεσματικά με μια απλή διαδικασία [11]. Μέσα από αυτή την ανακάλυψη τα νευρωνικά δίκτυα ξεπέρασαν την δυσκολία του μοντέλου του απλού αισθητήρα να εκπαιδευτεί σε μη γραμμικές λειτουργίες.

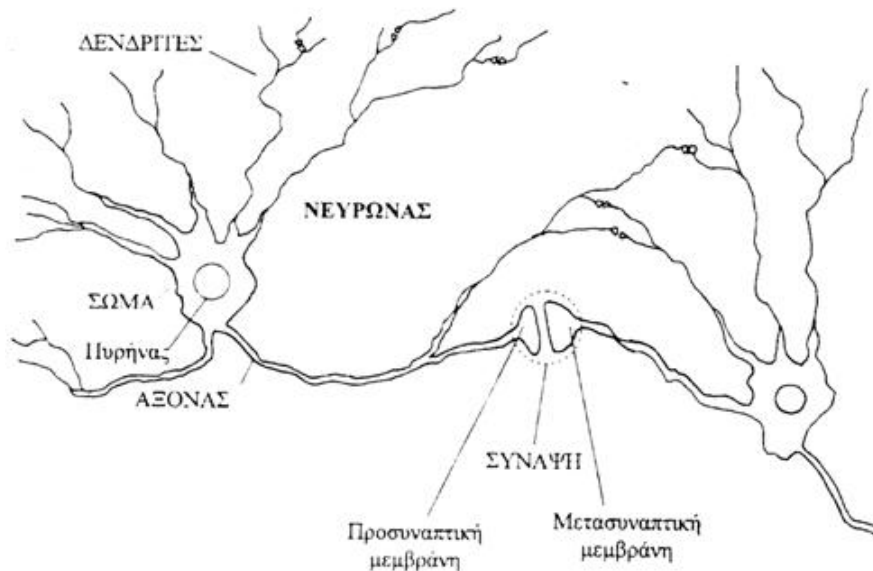
Ο αλγόριθμος οπισθοδιάδοσης αποτέλεσε σημείο αναφοράς για την αλματώδη εξέλιξη των τεχνητών νευρωνικών δικτύων μέχρι και την σημερινή εποχή, ωθώντας στην ανάπτυξη πολλών εφαρμογών, όπως η πρόβλεψη χρηματοοικονομικών χρονολογικών σειρών και η αναγνώριση παλμού ραντάρ και καταλήγοντας στην ανάπτυξη των βαθιών νευρωνικών δικτύων (Deep Neural Networks).

### 2.3.2 Βιολογικά Νευρωνικά Δίκτυα

Η ιδέα των τεχνητών νευρωνικών δικτύων φέρει την καταγωγή της στα βιολογικά νευρωνικά δίκτυα και την λειτουργία του ανθρώπινου εγκεφάλου. Έτσι, είναι σημαντικό να κατανοήσουμε την δομή και την λειτουργία των βιολογικών νευρωνικών δικτύων πριν μελετήσουμε την αντίστοιχη των τεχνητών νευρωνικών δικτύων.

Το νευρικό σύστημα του ανθρώπου έχει ως δομική μονάδα τον νευρώνα. Ο νευρώνας είναι ένα εξειδικευμένο είδος κυττάρου, που αποτελεί τον πυρήνα των συστημάτων επεξεργασίας πληροφοριών του ανθρώπινου κεντρικού νευρικού συστήματος και του εγκεφάλου. Ο ανθρώπινος εγκέφαλος είναι ένα τεράστιο δίκτυο διασυνδεδεμένων νευρώνων και υπολογίζεται ότι αποτελείται από περίπου 100 δισεκατομμύρια νευρώνες, καθένας από τους οποίους συνδέεται με περίπου 10.000 άλλους νευρώνες, μέσω των συνάψεων.

Η βασική δομή ενός βιολογικού νευρώνα δίνεται στο σχήμα 2.1. Γενικά, κάθε νευρώνας αποτελείται από ένα κυτταρικό σώμα, τον πυρήνα, τους δενδρίτες, τον άξονα και τις συνάψεις. Το σώμα επεξεργάζεται τις εισόδους που εισέρχονται στον νευρώνα και περιέχει τον πυρήνα. Οι δενδρίτες αποτελούν το σημείο εισόδου των σημάτων, που ο νευρώνας δέχεται από τους γειτονικούς του νευρώνες. Ο άξονας είναι μια μακριά κυλινδρική σύνδεση που μεταφέρει τα σήματα του νευρώνα στους δενδρίτες των γειτονικών του νευρώνων, αποτελεί δηλαδή το σημείο εξόδου των σημάτων του νευρώνα. Τα απειροελάχιστα κενά, στα οποία ενώνονται οι διακλαδώσεις του άξονα του νευρώνα με τους δενδρίτες των γειτονικών του νευρώνων, ονομάζονται συνάψεις. Τα σήματα που φτάνουν στις συνάψεις ενός νευρώνα και μεταφέρονται στους δενδρίτες των γειτονικών του νευρώνων έχουν την μορφή ηλεκτρικού παλμού. Η αμεσότητα μεταβίβασης τους από έναν νευρώνα σε έναν άλλο εξαρτάται από το πλάτος της σύναψης και από την πυκνότητα του ηλεκτροχημικού υλικού. Κατά την λειτουργία ένας νευρώνας είναι ενεργός ή ανενεργός ανάλογα με το αν μεταφέρει πληροφορία στους γειτονικούς νευρώνες. Έιδικότερα, ένας νευρώνας συλλέγει τα σήματα εισόδου



Σχήμα 2.1: Βιολογικός Νευρώνας  
Πηγή: [41]

που λαμβάνει από τους γειτονικούς του νευρώνες, τα αθροίζει στο σώμα του και στην συνέχεια εάν το άθροισμα των εισερχόμενων σημάτων ξεπερνάει μια τιμή καταστολής, γνωστή και ως κατώφλι (Threshold), δημιουργεί ένα παλμό απόκρισης, ο οποίος μεταφέρεται στους γειτονικούς του νευρώνες. Σε αντίθετη περίπτωση ο νευρώνας παραμένει σε καταστολή και δεν μεταφέρει την πληροφορία, δηλαδή είναι ανενεργός. Πολλοί νευρώνες μαζί, οι οποίοι συνδέονται μεταξύ τους μέσω των συνάψεων, δομούν ένα βιολογικό νευρωνικό δίκτυο, μέσα από το οποίο διέρχονται πληροφορίες για την ομαλή λειτουργία του ανθρώπινου οργανισμού.

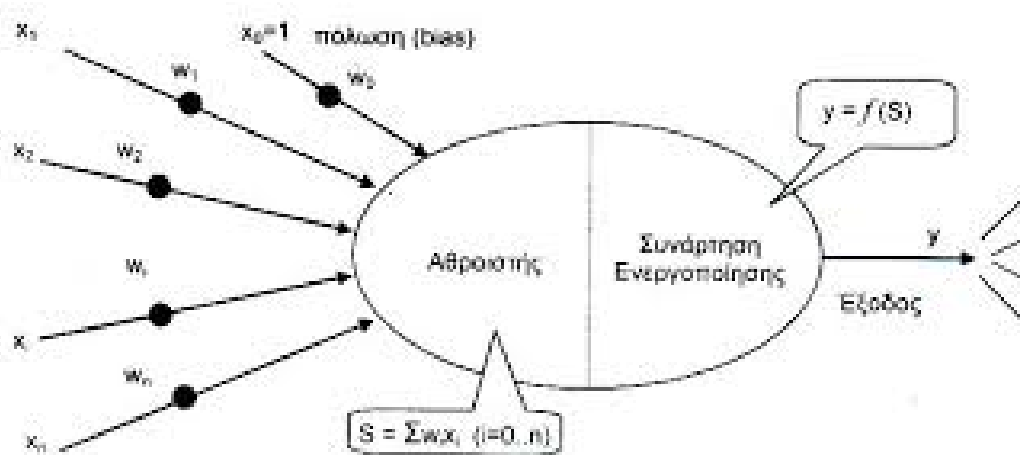
### 2.3.3 Το Μοντέλο του Τεχνητού Νευρώνα

Όπως έχουμε κάνει ήδη γνωστό, η ιδέα ανάπτυξης των τεχνητών νευρωνικών δικτύων στηρίζεται στην δομή και την λειτουργία των βιολογικών νευρωνικών δικτύων. Έτσι, όμοια με τα βιολογικά νευρωνικά δίκτυα, ένα τεχνητό νευρωνικό δίκτυο είναι ένα δίκτυο από νευρώνες διασυνδεδεμένους μεταξύ τους. Ένας βιολογικός νευρώνας λαμβάνει στους δένδριτες του τα σήματα εισόδου σε μορφή ηλεκτρικών παλμών, τα επεξεργάζεται και τελικά τα μεταφέρει μέσω του άξονα στους γειτονικούς του νευρώνες, με την προϋπόθεση ότι βρίσκεται σε ενεργή κατάσταση.

Στο αντίστοιχο μαθηματικό μοντέλο, ο τεχνητός νευρώνας υποδέχεται τις τιμές εισόδου σε μορφή διανύσματος  $(x_1, x_2, \dots, x_n)$ . Το ρόλο των συνάψεων και άρα το κριτήριο για την επιρροή, που κάθε τιμή εισόδου έχει στον νευρώνα, τον έχουν τα συναπτικά βάρη  $w_i$ . Τα συναπτικά βάρη είναι ένα διάνυσμα τιμών της μορφής  $(w_0, w_1, w_2, \dots, w_n)$ , όπου  $w_0$  είναι η τιμή πόλωσης ή μεροληψία του εκάστοτε νευρώνα. Τα συναπτικά βάρη παίρνουν θετικές ή αρνητικές τιμές, ανάλογα με την διεγερτική ή ανασταλτική λειτουργία της "σύναψης". Η πόλωση έχει πάντα σαν τιμή εισόδου την μονάδα και αποτελεί την τιμή επιρροής του κάθε νευρώνα. Η επεξεργασία των εισόδων στο μοντέλο του τεχνητού νευρώνα γίνεται όπως και στον βιολογικό νευρώνα στο σώμα του. Στον τεχνητό νευρώνα το σώμα χωρίζεται σε δύο μέρη: τον αθροιστή (Sum) και την συνάρτηση ενεργοποίησης (Activation Function). Αρχικά, ο αθροιστής αθροίζει τα επηρεασμένα από τα βάρη σήματα εισόδου και παράγει ένα άθροισμα της μορφής:

$$S = \sum_{i=1}^n (x_i \cdot w_i + w_0)$$

Στην συνέχεια, η τιμή του αθροίσματος  $S$  εισάγεται στην συνάρτηση ενεργοποίησης  $f$  του νευρώνα και έτσι υπολογίζεται η τιμή εξόδου  $y$  που παράγει. Η συνάρτηση ενεργοποίησης αποτελεί ένα είδος φίλτρου των σημάτων εισόδου του νευρώνα. Ένα τεχνητός νευρώνας ανάλογα με την τιμή εξόδου που παράγει, μπορεί να είναι ενεργός, δηλαδή να μεταφέρει τον παλμό του στους γειτονικούς του νευρώνες ή αδρανής. Η τιμή εξόδου ενός νευρώνα είναι μοναδική και όταν νευρώνας είναι ενεργός μεταφέρεται σε όλους τους γειτονικούς του νευρώνες. Η δομή του μοντέλου του τεχνητού νευρώνα απεικονίζεται στο σχήμα 2.2.



Πηγή: [34]

Σχήμα 2.2: Το Μοντέλο του Τεχνητού Νευρώνα

## 2.4 Συνάρτηση Ενεργοποίησης

Η συνάρτηση ενεργοποίησης αποτελεί σημαντικό κομμάτι για την επεξεργασία του σήματος, για την έξοδο που ο κάθε νευρώνας παράγει και κατ' επέκταση και για το τεχνητό νευρωνικό δίκτυο. Αυτό οφείλεται στο γεγονός ότι διαφορετικές συναρτήσεις ενεργοποίησης εξάγουν διαφορετικές εξόδους. Μερικές από τις πιο διαδεδομένες συναρτήσεις ενεργοποίησης είναι οι παρακάτω:

- **Σιγμοειδής Συνάρτηση (Sigmoid Function):** Η σιγμοειδής συνάρτηση έχει γραφική παράσταση σχήματος S-καμπύλης και είναι μη-γραμμική συνάρτηση. Ο γενικός της τύπος είναι της μορφής:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

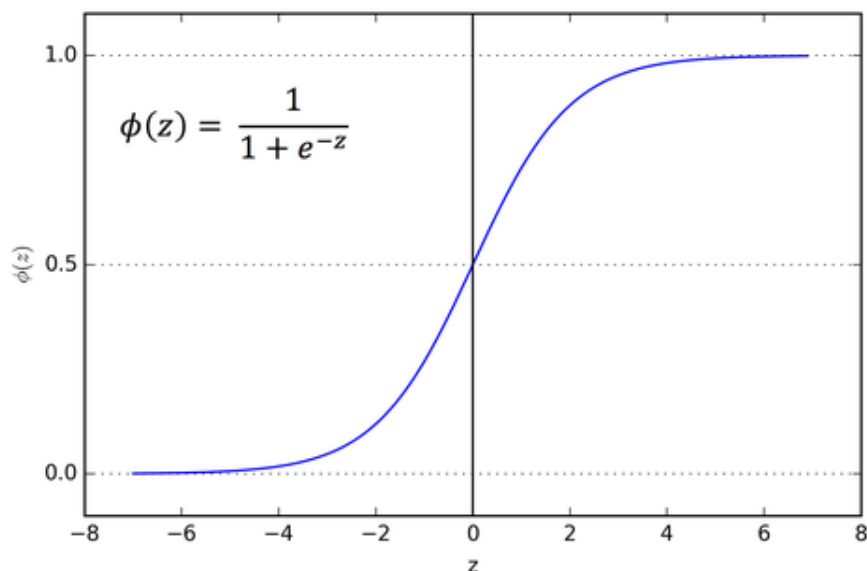
,όπου  $x_0$  είναι το κεντρικό σημείο της καμπύλης,  $L$  η μέγιστη τιμή της καμπύλης και  $k$  η παράμετρος του ρυθμού ανάπτυξης της καμπύλης. Η συνάρτηση αυτή έχει πεδίο ορισμού όλο το σύνολο των πραγματικών αριθμών  $(-\infty, +\infty)$  και σύνολο τιμών το σύνολο  $(0, L)$ . Ανάλογα με τις τιμές που παίρνουν οι παράμετροι της η σιγμοειδής συνάρτηση μπορεί να πάρει κάποιες συγκεκριμένες μορφές. Οι πιο χαρακτηριστικές σιγμοειδείς συναρτήσεις είναι η λογιστική συνάρτηση (Logistic Function) και η συνάρτηση υπερβολικής εφαπτομένης (Hyperbolic Tangent).

- **Λογιστική Συνάρτηση (Logistic Function):** Για  $L=1$ ,  $k=1$ ,  $x_0=0$  ο γενικός τύπος της σιγμοειδούς συνάρτησης γίνεται:

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

Η λογιστική συνάρτηση είναι μία από τις πιο διαδεδομένες συναρτήσεις ενεργοποίησης στα τεχνητά νευρωνικά δίκτυα και όπως και η γενική σιγμοειδή συνάρτηση είναι μη-γραμμική. Όμοια με την σιγμοειδή συνάρτηση το πεδίο ορισμού της λογιστικής συνάρτησης είναι το σύνολο των πραγματικών αριθμών  $(-\infty, +\infty)$ , ενώ το σύνολο τιμών της είναι το  $(0, 1)$ , αφού το  $L=1$ . Επίσης, είναι μια συνεχής και διαφορίσιμη συνάρτηση στο πεδίο ορισμού της.

Μερικά από τα πλεονεκτήματα της χρήσης της λογιστικής συνάρτησης ως συνάρτηση ενεργοποίησης στα νευρωνικά δίκτυα είναι ότι παράγει κανονικοποιημένες τιμές εξόδου, δηλαδή τιμές μεταξύ του  $(0, 1)$  και ότι για τιμές εισόδου μικρότερες του  $-2$  και μεγαλύτερες του  $+2$ , αλλά όχι πολύ μικρές ή πολύ μεγάλες τιμές, παράγει ακριβείς προβλέψεις, καθώς η τιμή εξόδου του νευρώνα είναι πολύ κοντά στο  $0$  ή στο  $1$  αντίστοιχα. Από την άλλη πλευρά, η χρήση της λογιστικής συνάρτησης παρουσιάζει και ορισμένα μειονεκτήματα, όπως ότι είναι υπολογιστικά σύνθετη και ότι δεν είναι συμμετρική γύρω από το μηδέν, γεγονός που σημαίνει ότι οι τιμές των εξόδων των νευρώνων που παράγονται είναι ίδιες. Επιπρόσθετα, για πολύ μεγάλες ή πολύ μικρές τιμές εισόδου η κλίση του νευρώνα είναι σχεδόν μηδενική με συνέπεια να διακόπτεται η εκπαίδευση του νευρωνικού δικτύου κατά την εκτέλεση του αλγόριθμου οπισθοδιάδοσης [12].



Πηγή: [40]

Σχήμα 2.3: Λογιστική Συνάρτηση

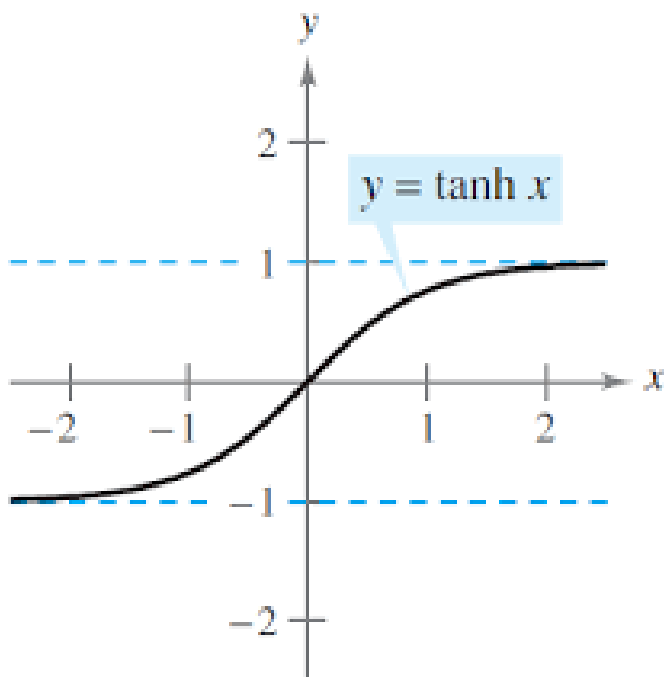
- **Συνάρτηση Υπερβολικής Εφαπτομένης (Hyperbolic Tangent, tanh):** Μια ακόμα ευρέως χρησιμοποιημένη στα τεχνητά νευρωνικά δίκτυα συνάρτηση ενεργοποίησης από την οικογένεια των σιγμοειδών συναρτήσεων είναι η υπερβολική εφαπτομένη. Ο τύπος της υπερβολικής εφαπτομένης είναι της μορφής:

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

Η υπερβολική εφαπτομένη είναι μη-γραμμική συνάρτηση και όμοια με την λογιστική είναι συνεχής και διαφορίσιμη στο πεδίο ορισμού της. Το πεδίο ορισμού της είναι το σύνολο των πραγματικών αριθμών  $(-\infty, +\infty)$ , ενώ το σύνολο τιμών της είναι το διάστημα  $(-1, 1)$ .

$$f(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} = \frac{1}{2} + \frac{1}{2} \cdot \tanh(x/2)$$

Από την παραπάνω εξίσωση παρατηρούμε την μαθηματική σχέση μεταξύ της λογιστικής συνάρτησης και της υπερβολικής εφαπτομένης. Είναι φανερό ότι οι δύο συναρτήσεις είναι παρόμοιες. Παρόλα αυτά η υπερβολική εφαπτομένη είναι συμμετρική γύρω από το μηδέν σε αντίθεση με την λογιστική. Αυτό έχει ως αποτέλεσμα να παράγονται διαφορετικοί έξοδοι από τα προηγούμενα επίπεδα νευρώνων, οι οποίες θα τροφοδοτηθούν στα επόμενα επίπεδα. Επιπρόσθετα, η κλίση της υπερβολικής εφαπτομένης είναι πιο απότομη σε σχέση με την αντίστοιχη της λογιστικής. Γενικότερα η υπερβολική εφαπτομένη προτιμάται από την λογιστική συνάρτηση, καθώς οι κλίσεις της δεν περιορίζονται σε ένα συγκεκριμένο εύρος τιμών [12].



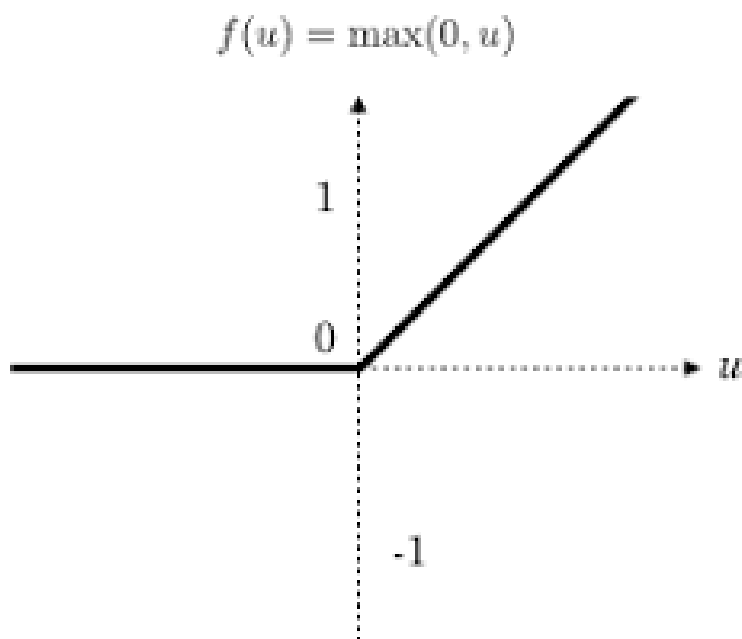
Πηγή: [35]

Σχήμα 2.4: Συνάρτηση Υπερβολικής Εφαπτομένης

- **Συνάρτηση Ράμπας (Rectified Linear Unit, ReLU):** Η συνάρτηση μπάρας είναι μια μη-γραμμική συνάρτηση ενεργοποίησης, η οποία χρησιμοποιείται ευρέως στα τεχνητά νευρωνικά δίκτυα. Επίσης, είναι συνεχής και διαφορίσιμη στο πεδίο ορισμού της. Το πεδίο ορισμού της είναι το σύνολο των πραγματικών αριθμών  $(-\infty, +\infty)$ , ενώ το σύνολο τιμών της είναι για  $x < 0$  το 0 και για  $x \geq 0$  το  $x$ . Ο τύπος της συνάρτησης μπάρας είναι της μορφής:

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

Η συνάρτηση μπάρας θεωρείται πιο αποτελεσματική από άλλες συναρτήσεις ενεργοποίησης, καθώς δεν ενεργοποιούνται όλοι οι νευρώνες ενός δικτύου ταυτόχρονα, αλλά ορισμένοι κάθε κάποιο χρονικό διάστημα. Επίσης θεωρείται υπολογιστικά απλή, διότι μοιάζει αρκετά με γραμμική συνάρτηση. Το σημαντικότερο μειονέκτημα της συνάρτησης ράμπας είναι ότι σε ορισμένες περιπτώσεις η κλίση της είναι μηδέν με αποτέλεσμα τα βάρη να μην προσαρμόζονται κατά την εκπαίδευση του δικτύου [12].



Πηγή: [42]

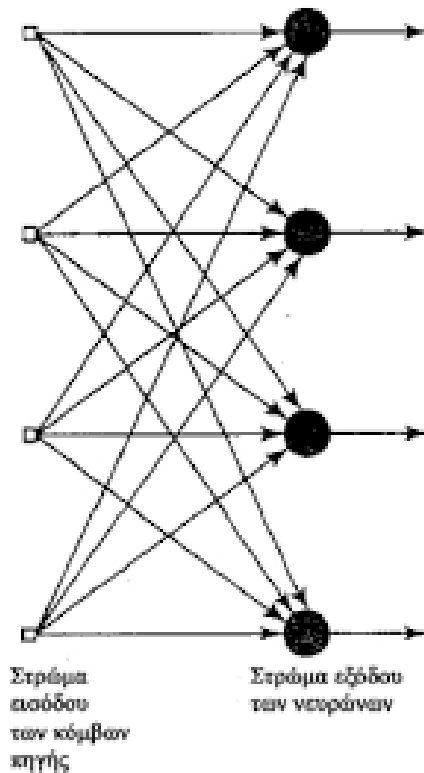
Σχήμα 2.5: Συνάρτηση Ράμπας

## 2.5 Αρχιτεκτονικές Τεχνητών Νευρωνικών Δικτύων

Τα τεχνητά νευρωνικά δίκτυα αναπτύχθηκαν με στόχο να προσομοιάσουν την λειτουργία του ανθρώπινου εγκεφάλου και να συνδυάσουν τις ικανότητες των ηλεκτρονικών υπολογιστών με αυτές του ανθρώπου. Οι ηλεκτρονικοί υπολογιστές έχουν την δυνατότητα να υπολογίζουν με ακρίβεια σύνθετες υπολογιστικές πράξεις, όπως επίσης και να αποθηκεύουν μεγάλου όγκου δεδομένα, ενώ από την άλλη πλευρά ο άνθρωπος μπορεί να αναγνωρίζει αντικείμενα και καταστάσεις. Τα πρώτα τεχνητά νευρωνικά δίκτυα ήταν απλά δίκτυα ενός νευρώνα, τα οποία χρησιμοποιούνταν για τον υπολογισμό λογικών πράξεων, όπως το μοντέλο McCulloch-Pitts. Ο στόχος της προσομοίωσης της λειτουργίας του ανθρώπινου εγκεφάλου και η ανάγκη για συλλογή πληροφοριών από μεγάλου όγκου δεδομένα οδήγησαν σε ορισμένες σημαντικές ανακαλύψεις στον τομέα των τεχνητών νευρωνικών δικτύων, όπως οι κανόνες μάθησης του Hebb, το μοντέλο του αισθητήρα (Perceptron) και ο αλγόριθμος οπισθοδιάδοσης. Έτσι, σταδιακά αναπτύχθηκαν τα δίκτυα πολλαπλών επιπέδων. Με τα δίκτυα πολλαπλών επιπέδων ο άνθρωπος ερχόταν ένα βήμα πιο κοντά στην λειτουργία του ανθρώπινου εγκεφάλου, καθώς πλέον τα νευρωνικά δίκτυα είχαν την ικανότητα να εκπαιδεύονται, να μαθαίνουν και τελικά να διακρίνουν σε κατηγορίες δεδομένα ή να προβλέπουν τιμές. Στην σημερινή εποχή, έχουν αναπτυχθεί πολλά είδη τεχνητών νευρωνικών δικτύων, τα οποία διαχωρίζονται με βάση την λειτουργία, την μέθοδο εκπαίδευσης και την αρχιτεκτονική τους. Με τον όρο αρχιτεκτονική ενός τεχνητού νευρωνικού δικτύου ορίζεται ο τρόπος διάταξης των συνδέσεων των νευρώνων, ο αριθμός, ο τύπος τους και ο τρόπος διάδοσης της πληροφορίας μέσα στο δίκτυο. Μερικές από τις πιο χαρακτηριστικές αρχιτεκτονικές τεχνητών νευρωνικών δικτύων είναι τα δίκτυα πρόσθιας τροφοδότησης ενός επιπέδου, τα πολυεπίπεδα δίκτυα πρόσθιας τροφοδότησης και τα αναδρομικά δίκτυα.

### 2.5.1 Δίκτυα Πρόσθιας Τροφοδότησης Ενός Επιπέδου

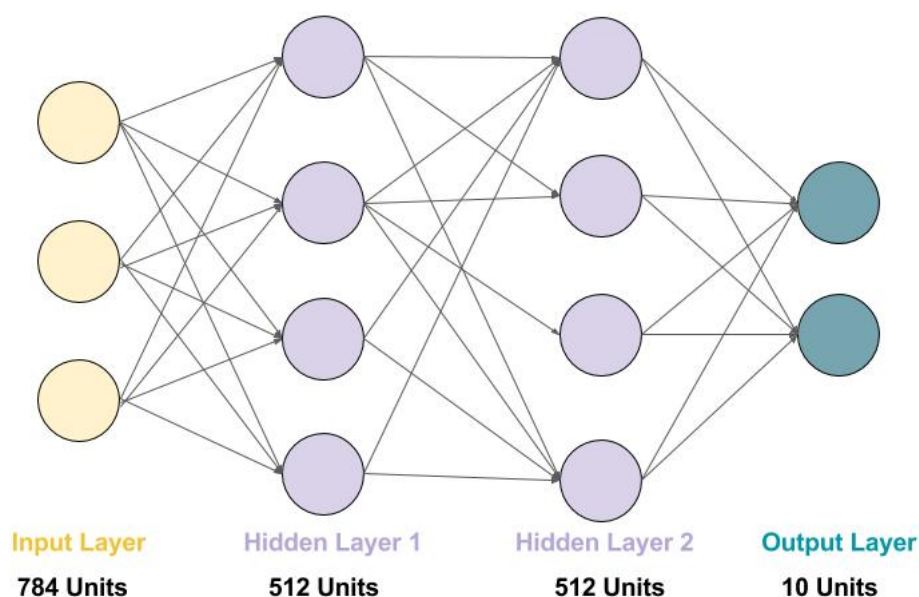
Τα δίκτυα πρόσθιας τροφοδότησης ενός επιπέδου (One Layer Feed-forward Neural Networks) παρουσιάζουν την πιο απλή δομή που ένα δίκτυο μπορεί να έχει. Η δομή τους περιλαμβάνει δύο επίπεδα: το επίπεδο εισόδου και το επίπεδο εξόδου. Το επίπεδο εισόδου αποτελείται από νευρώνες εισόδου, οι οποίοι δεν εκτελούν κάποιο υπολογισμό, αλλά χρησιμοποιούνται για την συλλογή των δεδομένων εισόδου και την μεταφορά τους προς το επίπεδο εξόδου. Στα δίκτυα πρόσθιας τροφοδότησης το ηλεκτρικό σήμα μεταφέρεται μονόδρομα από το επίπεδο εισόδου προς το επίπεδο εξόδου. Έτσι, η πληροφορία μετά την εισαγωγή της σε μορφή διανύσματος στο επίπεδο εισόδου, μεταφέρεται προς το επίπεδο εξόδου, το οποίο αποτελείται από τους νευρώνες εξόδου. Οι νευρώνες εξόδου είναι υπολογιστικοί νευρώνες και υπολογίζουν την τελική τιμή που το δίκτυο παράγει. Η διαδικασία υπολογισμού της τελικής τιμής που ένας νευρώνας παράγει, περιγράφηκε στην ενότητα του μοντέλου του τεχνητού νευρώνα. Η αρχιτεκτονική ενός δικτύου πρόσθιας τροφοδότησης ενός επιπέδου, απεικονίζεται στο σχήμα 2.6.



Σχήμα 2.6: Δίκτυο Πρόσθιας Τροφοδότησης ενός Επιπέδου  
 Πηγή: [35]

### 2.5.2 Πολυεπίπεδα Δίκτυα Πρόσθιας Τροφοδότησης

Τα πολυεπίπεδα δίκτυα πρόσθιας τροφοδότησης (Multilayer Feed-forward Neural Networks) δομήθηκαν σε απόλυτη συνάφεια με τα δίκτυα πρόσθιας τροφοδότησης ενός επιπέδου. Έτσι, στα δίκτυα αυτά υπάρχουν τα επίπεδα εισόδου και εξόδου, ενώ διαφοροποιούνται από τα δίκτυα πρόσθιας τροφοδότησης ενός επιπέδου με την προσθήκη ενός ή περισσότερων κρυφών επιπέδων ανάμεσα στα επίπεδα εισόδου και εξόδου. Όμοια, με τα δίκτυα ενός επιπέδου στα πολυεπίπεδα δίκτυα το ηλεκτρικό σήμα μεταφέρεται μονόδρομα από το επίπεδο εισόδου προς το επίπεδο εξόδου και μέσα από τα κρυφά επίπεδα. Το επίπεδο εισόδου στα πολυεπίπεδα δίκτυα πρόσθιας τροφοδότησης αποτελείται από μη-υπολογιστικούς νευρώνες, οι οποίοι συλλέγουν και μεταφέρουν τα δεδομένα εισόδου προς τα κρυφά επίπεδα. Τα κρυφά επίπεδα αποτελούνται από νευρώνες, οι οποίοι ονομάζονται κρυφοί και όπως οι νευρώνες του επιπέδου εξόδου είναι υπολογιστικοί. Μετά τον υπολογισμό της τιμής εξόδου των νευρώνων του κρυφού επιπέδου οι τιμές εξόδου αυτές εισέρχονται ως τιμές εισόδου στους νευρώνες του επιπέδου εξόδου, αφού πρώτα πολλαπλασιαστούν με τα αντίστοιχα συναπτικά βάρη. Τελικώς, οι νευρώνες του επιπέδου εξόδου υπολογίζουν τις τελικές τιμές που το νευρωνικό δίκτυο εξάγει. Η προσθήκη των κρυφών επιπέδων ήταν καθοριστική στην εξέλιξη των τεχνητών νευρωνικών δικτύων, καθώς με την υπολογιστική ισχύ που προσέφεραν τα δίκτυα ήταν πλέον ικανά να υπολογίζουν πιο σύνθετες μη γραμμικές συναρτήσεις και άρα να κατηγοριοποιούν πιο σύνθετα δεδομένα. Τέλος, σε αυτού του είδους τα δίκτυα δεν υπάρχει ουδεμία ανάδραση ανάμεσα στους νευρώνες του ίδιου επιπέδου ή σε νευρώνες διαφορετικών επιπέδων. Η αρχιτεκτονική ενός πολυεπίπεδου δικτύου πρόσθιας τροφοδότησης απεικονίζεται στο σχήμα 2.7.

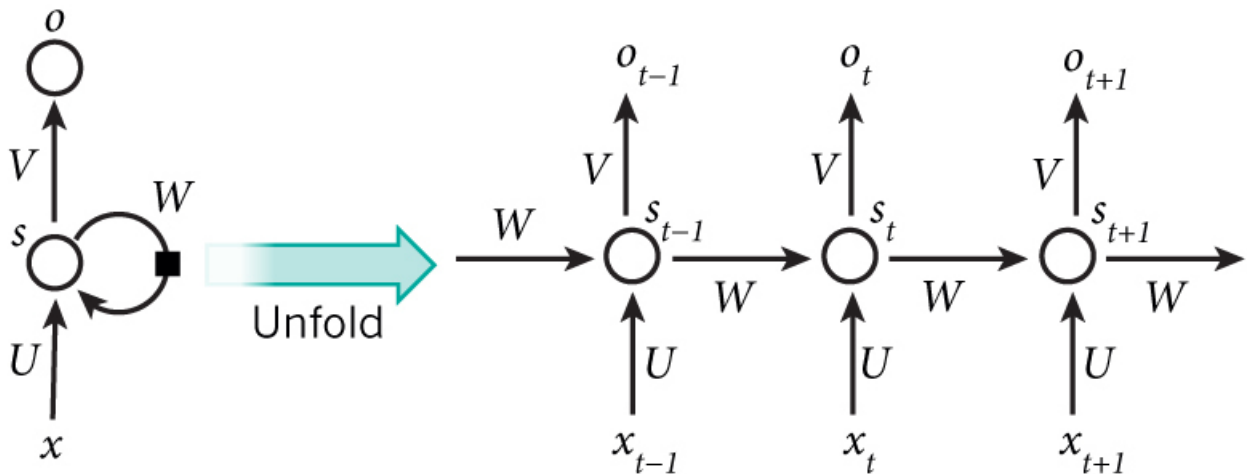


Σχήμα 2.7: Πολυεπίπεδο Δίκτυο Πρόσθιας Τροφοδότησης  
Πηγή: [43]

### 2.5.3 Αναδρομικά Νευρωνικά Δίκτυα

Στην αρχιτεκτονική των δικτύων πρόσθιας τροφοδότησης γίνεται η υπόθεση ότι όλες οι εισοδοί είναι ανεξάρτητες μεταξύ τους. Με την ανάπτυξη των τεχνητών νευρωνικών δικτύων παρατηρήθηκε ότι η υπόθεση αυτή παρουσιάζει αρκετά κενά στην λειτουργία των νευρωνικών δικτύων, όπως η αδυναμία μεταφοράς χαρακτηριστικών από το ένα επίπεδο νευρώνων στο άλλο ή ακόμα και μεταξύ των νευρώνων του ίδιου επιπέδου. Έτσι, μέσα από αυτή την υπόθεση χάνεται η αναγκαία για την αποτελεσματικότερη μάθηση, κληρονομικότητα. Η ανακάλυψη των αναδρομικών νευρωνικών δικτύων (Recurrent Neural Networks) συνέβαλε σημαντικά στην αντιμετώπιση της έλλειψης κληρονομικότητας. Το χαρακτηριστικό που ξεχωρίζει τα αναδρομικά νευρωνικά δίκτυα από τα πρόσθιας τροφοδότησης είναι ότι περιέχουν τουλάχιστον μία ανάδραση ανάμεσα στους νευρώνες του ίδιου επιπέδου ή διαφορετικών επιπέδων. Η ιδιότητα αυτή προσφέρει στα αναδρομικά δίκτυα την ικανότητα αντίληψης του χρόνου. Πιο συγκεκριμένα, τα δίκτυα αυτά μπορούν να χρησιμοποιηθούν σε εφαρμογές με δυναμικά συστήματα για να προβλέψουν την έξοδο της στιγμής  $t+1$  γνωρίζοντας τα δεδομένα προηγούμενων στιγμών  $t$ ,  $t-1$ . Επιπλέον, στα αναδρομικά νευρωνικά δίκτυα με την ύπαρξη της ανάδρασης μεταξύ των νευρώνων εισάγεται η έννοια της “μνήμης” του δικτύου. Λόγω των δύο παραπάνω ιδιοτήτων τους, τα αναδρομικά δίκτυα χρησιμοποιούνται για να μοντελοποιούν και να αναπαραστήσουν χρονικά μεταβαλλόμενα μοντέλα. Γενικότερα, τα αναδρομικά δίκτυα είναι δομημένα όμοια με τα πολυεπίπεδα δίκτυα πρόσθιας τροφοδότησης, δηλαδή αποτελούνται από τα επίπεδα εισόδου, εξόδου και τα ενδιάμεσα κρυφά επίπεδα, ενώ η πληροφορία κατευθύνεται από το επίπεδο εισόδου προς το επίπεδο εξόδου όχι όμως μονόδρομα. Στα δίκτυα αυτά εκτός από τα κλασική ροή που γνωρίζουμε από τα δίκτυα πρόσθιας τροφοδότησης, το ηλεκτρικό σήμα του εκάστοτε νευρώνα ανατροφοδοτείται στον εαυτό του ή σε νευρώνα του ίδιου ή προηγούμενου επιπέδου. Η αρχιτεκτονική ενός αναδρομικού νευρωνικού δικτύου απεικονίζεται στο σχήμα 2.8.





Σχήμα 2.8: Αναδρομικό Νευρωνικό Δίκτυο  
Πηγή: [46]

Ένα δίκτυο μπορεί να είναι πλήρως συνδεδεμένο ή μερικώς συνδεδεμένο. Σε ένα πλήρως συνδεδεμένο δίκτυο οι νευρώνες του κάθε επιπέδου του συνδέονται με όλους τους νευρώνες του επόμενου επιπέδου, ενώ σε ένα μερικώς συνδεδεμένο δίκτυο συνδέονται με ορισμένους νευρώνες του επόμενου επιπέδου [6].

## 2.6 Το Μοντέλο του Στοιχειώδους Αισθητήρα (Perceptron)

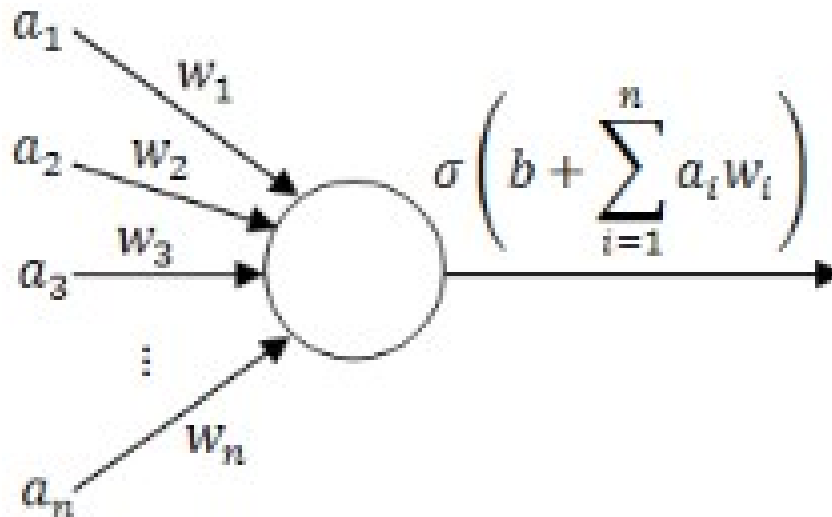
Το μοντέλο του στοιχειώδους αισθητήρα, γνωστό και ως μοντέλο Perceptron είναι από τα πρώτα τεχνητά νευρωνικά δίκτυα που αναπτύχθηκαν. Προτάθηκε το 1958 από τον Rosenblatt και αποτέλεσε έναυσμα για την δημιουργία νέων πιο σύνθετων τεχνητών νευρωνικών δικτύων με μεγαλύτερες δυνατότητες. Το μοντέλο αυτό αποτελεί ένα από τα πιο απλά τεχνητά νευρωνικά δίκτυα που σχεδιάστηκαν ποτέ, καθώς αποτελείται μόνο από έναν νευρώνα, ενώ η λειτουργία και η δομή του είναι ίδια με αυτή του τεχνητού νευρώνα που μελετήσαμε σε προηγούμενη ενότητα της εργασίας. Επίσης, χρησιμοποιούνταν ως δυαδικός γραμμικός ταξινομητής για τα δεδομένα εισόδου του. Παρά τα αρκετά μειονεκτήματα που αποδείχθηκε ότι το μοντέλο του στοιχειώδους αισθητήρα παρουσίαζε αποτέλεσε σημαντικό κομμάτι στην εξέλιξη των τεχνητών νευρωνικών δικτύων, καθώς συνέβαλε στην ανάπτυξη των αλγόριθμων μάθησης.

Το μοντέλο του αισθητήρα μπορεί να θεωρηθεί ως δίκτυο πρόσθιας τροφοδότησης ενός επιπέδου, όπου ο μοναδικός νευρώνας του δικτύου αποτελεί το επίπεδο εξόδου και επεξεργασίας των δεδομένων. Η δομή του μοντέλου Perceptron δίνεται στο σχήμα 2.9. Ο νευρώνας του μοντέλου υποδέχεται το διάνυσμα των τιμών εισόδου, οι οποίες πολλαπλασιάζονται με τα αντίστοιχα συναπτικά βάρη. Κάθε τιμή εισόδου  $x_i$  συνδέεται με τον νευρώνα μέσω μιας τιμής βάρους  $w_i$ , η οποία παίζει τον ρόλο των βιολογικών συνάψεων και φανερώνει την επιρροή που το κάθε σήμα έχει στον νευρώνα. Όμοια με το μοντέλο του τεχνητού νευρώνα στο διάνυσμα των συνοπτικών βαρών υπάρχει η πόλωση του νευρώνα  $w_0$ , η οποία φανερώνει την μεροληψία του νευρώνα να διεγερθεί ή όχι. Γενικότερα, τα συναπτικά βάρη παίρνουν αρνητικές ή θετικές τιμές, ανάλογα με την διεγερτική ή ανασταλτική λειτουργία της “σύναψης”. Στην συνέχεια, ο αθροιστής του νευρώνα αθροίζει τα γινόμενα των εισόδων με τα βάρη παράγοντας ένα άθροισμα της μορφής:

$$S = \sum_{i=1}^n (x_i \cdot w_i + w_0)$$

Το άθροισμα  $S$  εισάγεται στην συνάρτηση κατωφλιού Heaviside, με ένα συγκεκριμένο κατώφλι  $\theta$ . Σύμφωνα με την συνάρτηση ενεργοποίησης αυτή συγκρίνουμε την τιμή του αθροίσματος  $S$  με την τιμή κατωφλιού  $\theta$ . Αν  $S > \theta$  τότε ο αισθητήρας ενεργοποιείται και θεωρούμε ότι πυροδοτεί, ενώ αν  $S < \theta$  ο αισθητήρας παραμένει αδρανής. Πιο συγκεκριμένα, αν ο αισθητήρας είναι ενεργός η τιμή εξόδου του είναι 1, ενώ αν παραμένει αδρανής είναι 0.

Συνοψίζοντας, γίνεται εύκολα κατανοητό ότι η ενεργητικότητα του αισθητήρα εξαρτάται από τα βάρη των συνδέσεων, τις τιμές των εισόδων και την τιμή του κατωφλιού ( $\theta$ ). Έτσι, θεωρούμε ότι η πληροφορία βρίσκεται στα συναπτικά βάρη των συνδέσεων, τα οποία μεταβάλλονται συνεχώς κατά την εκπαίδευση του μοντέλου [10].



Σχήμα 2.9: Το Μοντέλο του Στοιχειώδους Αισθητήρα  
Πηγή: [45]

## 2.7 Κανόνας Μάθησης Perceptron

Το δίκτυο Perceptron άσκησε την μεγαλύτερη επιρροή από οποιοδήποτε από τα αρχικά νευρωνικά δίκτυα που αναπτύχθηκαν. Η σημαντική του επιρροή οφείλεται κατά κύριο λόγο στην ανάπτυξη των αλγορίθμων μάθησης. Ο κανόνας μάθησης του στοιχειώδους αισθητήρα είναι μια βελτιωμένη έκδοση του κανόνα μάθησης Hebb. Στόχος των κανόνων μάθησης γενικότερα και του κανόνα μάθησης Perceptron ειδικότερα είναι παροχή μεγαλύτερης ακρίβειας προβλέψεων. Για την επίτευξη του στόχου αυτού ο κανόνας μάθησης τροποποιεί τα βάρη του νευρωνικού δικτύου με στόχο να μειωθεί το σφάλμα που δημιουργείται. Το σφάλμα αυτό υπολογίζεται βρίσκοντας την διαφορά μεταξύ της τιμής εξόδου που παράγει το δίκτυο και της τιμής στόχου του δικτύου. Ο αλγόριθμος μάθησης που παρουσιάζεται παρακάτω είναι κατάλληλος τόσο για δυαδικά (binary), όσο και για διπολικά (bipolar) δεδομένα εισόδου, με διπολική τιμή στόχου, καθορισμένο κατώφλι  $\theta$  και ρυθμιζόμενη μεροληψία, ενώ δεν επηρεάζεται από τις αρχικές τιμές των βαρών και του ποσοστού μάθησης (learning rate) [31].

- **Βήμα 1:** Αρχικοποίηση των βαρών και της μεροληψίας. Για απλότητα προτείνεται η αρχικοποίηση των βαρών και της μεροληψίας ως μηδέν και ο ορισμός του ποσοστού μάθησης με μια τιμή από το σύνολο  $(0,1]$ .
- **Βήμα 2:** Όσο η συνθήκη τερματισμού είναι λανθασμένη, ακολουθήσε τα βήματα 3-7.
- **Βήμα 3:** Για κάθε ζευγάρι εκπαίδευσης  $s:t$ , κάνε τα βήματα 4-6.
- **Βήμα 4:** Καθόρισε τις ενεργοποιήσεις των μονάδων εισόδου ως  $x_i = s_i$
- **Βήμα 5:** Υπολόγισε την τιμή εξόδου  $y_i = b + \sum_{i=1}^n (x_i \cdot w_i)$
- **Βήμα 6:** Ανανέωσε τα βάρη και την μεροληψία εαν υπάρχει σφάλμα, δηλαδή αν  $y \neq t$ , τότε  $w_i(new) = w_i(old) + atx_i$  και  $b(new) = b(old) + at$ .
- **Βήμα 7:** Συνθήκη τερματισμού: Αν τα βάρη δεν αλλάζουν τιμή, τότε σταμάτα την εκπαίδευση, αλλιώς συνέχισε στο βήμα 3.

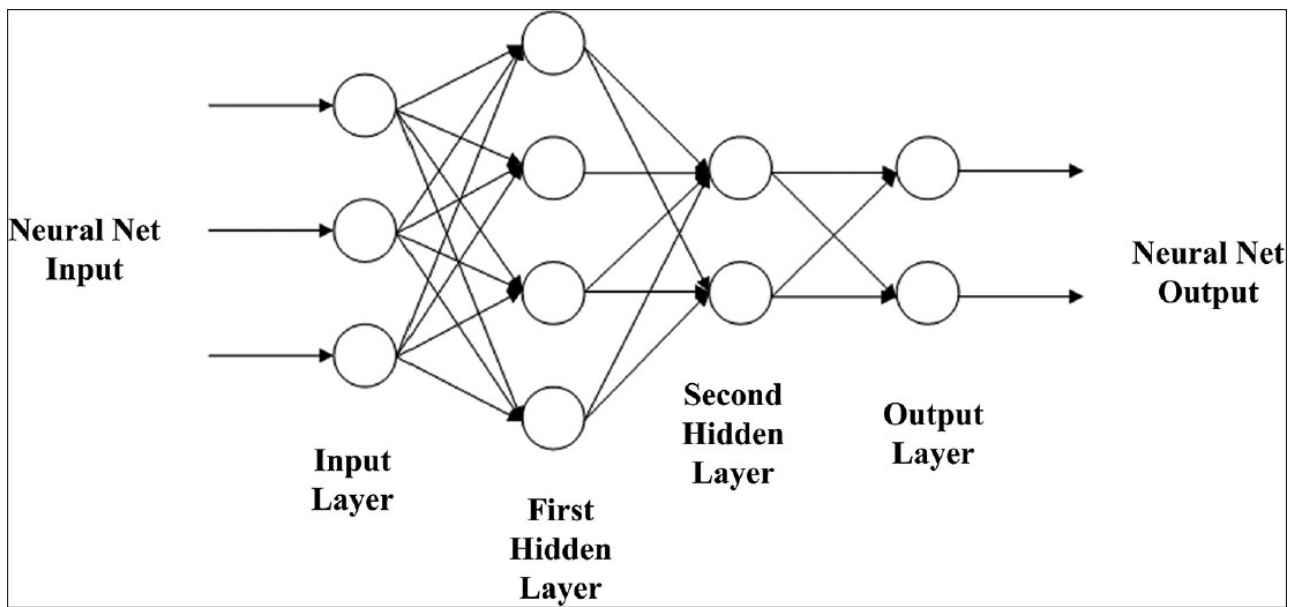
Στο σημείο αυτό αξίζει να σημειωθεί ότι ανανεώνονται μόνο τα βάρη που συνδέονται με ενεργές εισόδους ( $x_i \neq 0$ ). Επίσης, τα βάρη ανανεώνονται όταν η τιμή εξόδου που παράγει το δίκτυο διαφέρει από την τιμή στόχου και άρα υπάρχει σφάλμα. Αυτό σημαίνει ότι όσο περισσότερα δεδομένα εκπαίδευσης παράγουν σωστές προβλέψεις τόσο λιγότερη εκπαίδευση χρειάζεται το δίκτυο.

## 2.8 Πολυεπίπεδο Δίκτυο Perceptron

Η αδυναμία του μοντέλου του απλού αισθητήρα να επιλύσει απλά μη γραμμικά προβλήματα οδήγησε στην δημιουργία των πολυεπίπεδων δικτύων Perceptron (MultiLayer Perceptron (MLP)). Τα πολυεπίπεδα δίκτυα Perceptron αποτελούν δίκτυα πρόσθιας τροφοδότησης (Feed-forward Neural Networks), δηλαδή δίκτυα στα οποία η διάδοση του σήματος είναι μονόδρομη και δεν υφίσταται ανάδραση του. Τα MLP δίκτυα βασίζονται στην δομή και την λειτουργία τόσο του μοντέλου του στοιχειώδους αισθητήρα, όσο και των γνωστών μας πολυεπίπεδων δικτύων πρόσθιας τροφοδότησης. Η ιδέα πίσω από την οποία κρύβεται η γέννηση των πολυεπίπεδων δικτύων Perceptron είναι η προσθήκη των κρυφών επιπέδων στο δίκτυο. Πιο συγκεκριμένα, τα δίκτυα αυτά αποτελούνται από τα επίπεδα εισόδου, εξόδου και ένα ή περισσότερα κρυφά επίπεδα. Στα MLP δίκτυα ως επίπεδα πρόσθιας τροφοδότησης, το επίπεδο εισόδου αποτελείται από μη-υπολογιστικούς νευρώνες, οι οποίοι μεταφέρουν τα δεδομένα εισόδου προς τα κρυφά επίπεδα. Τα κρυφά επίπεδα αποτελούνται από υπολογιστικούς νευρώνες, οι οποίοι λαμβάνουν από το επίπεδο εισόδου τα δεδομένα σε μορφή διανύσματος και στην συνέχεια με κατάλληλη επεξεργασία, ίδια με αυτή του στοιχειώδους αισθητήρα, εξάγουν τις τελικές εξόδους τους. Στην συνέχεια, αυτές οι εξόδους εισάγονται ως είσοδοι στο επόμενο κρυφό επίπεδο, όπου πραγματοποιείται η ίδια διαδικασία. Τελικώς, το ηλεκτρικό σήμα φτάνει στο επίπεδο εξόδου, το οποίο όμοια με τα κρυφά επίπεδα αποτελείται από υπολογιστικούς νευρώνες. Οι νευρώνες αυτοί ακολουθώντας την γνωστή μας διαδικασία επεξεργασίας του στοιχειώδους αισθητήρα παράγουν τις τελικές τιμές  $y$  του μοντέλου. Στο σχήμα 2.10 απεικονίζεται η αρχιτεκτονική ενός πλήρως συνδεδεμένου MLP δικτύου με δύο κρυφά επίπεδα.

Τα πολυεπίπεδα δίκτυα Perceptron παρουσιάζουν ιδιαίτερα πιο πολύπλοκη δομή και λειτουργία από το μοντέλο του στοιχειώδους αισθητήρα. Μερικά από τα κυριότερα χαρακτηριστικά αυτά είναι τα εξής:

- Στα MLP δίκτυα οι υπολογιστικοί νευρώνες των επιπέδων χρησιμοποιούν μη-γραμμικές συναρτήσεις ενεργοποίησης, οι οποίες είναι συνεχείς και παραγωγίσιμες στο πεδίο ορισμού τους. Αντίθετα, στο μοντέλο Perceptron, η συνάρτηση ενεργοποίησης είναι γραμμική και μη παραγωγίσιμη στο πεδίο ορισμού της. Μερικές από τις πιο διαδεδομένες συναρτήσεις ενεργοποίησης στα δίκτυα MLP παρουσιάστηκαν σε προηγούμενη ενότητα, όπως η λογιστική συνάρτηση. Η διαφοροποίηση αυτή στο είδος συνάρτησης ενεργοποίησης των δύο δικτύων οφείλεται στην διατύπωση των Minsky και Papert το 1969, σύμφωνα με την οποία τα μονοεπίπεδα δίκτυα Perceptron είναι ικανά να μαθαίνουν αποκλειστικά γραμμικώς διαχωρίσιμα προβλήματα. Σε προέκταση της διατύπωσης αυτής συνεπάγεται άμεσα ότι τα πολυεπίπεδα δίκτυα με γραμμική συνάρτηση ενεργοποίησης μπορούν όμοια να μάθουν γραμμικώς διαχωρίσιμα προβλήματα, καθώς ο συνδυασμός γραμμικών συναρτήσεων είναι εξίσου γραμμική συνάρτηση. Έτσι, οι ερευνητές για να υπερκεράσουν το εμπόδιο αυτό στην εξέλιξη των τεχνητών νευρωνικών δικτύων εισήγαγαν τα πολυεπίπεδα δίκτυα και την χρήση μη-γραμμικών συναρτήσεων ενεργοποίησης. Με τα νέα πολυεπίπεδα δίκτυα μπορούσαν να αντιμετωπίσουν πιο σύνθετα μη-γραμμικώς διαχωρίσιμα προβλήματα.
- Οι συναρτήσεις ενεργοποίησης που χρησιμοποιούνται στους υπολογιστικούς νευρώνες των MLP δικτύων είναι μη-γραμμικές, καθώς με την χρήση γραμμικών συναρτήσεων ενεργοποίησης είναι αδύνατος ο υπολογισμός πιο σύνθετων συνεχών και διαφορίσιμων συναρτήσεων. Επιπρόσθετα, η προσθήκη των κρυφών επιπέδων και η χρήση μη-γραμμικών διαφορίσιμων συναρτήσεων ενεργοποίησης προσδίδει στα δίκτυα MLP μεγαλύτερη υπολογιστική ισχύ.
- Το χαρακτηριστικό των δικτύων αυτού του τύπου να χρησιμοποιούν διαφορίσιμες συναρτήσεις, οφείλεται στο γεγονός ότι οι περισσότεροι κανόνες εκπαίδευσης στηρίζονται σε μεθόδους βελτιστοποίησης που χρησιμοποιούν παραγώγους. Για τον λόγο αυτό, γραμμικές συναρτήσεις όπως η συνάρτηση κατωφλιού του μοντέλου του αισθητήρα δεν μπορούν να χρησιμοποιηθούν στα δίκτυα MLP.



Σχήμα 2.10: Πολυεπίπεδο Δίκτυο Perceptron με δύο κρυφά επίπεδα  
Πηγή: [44]

# Κεφάλαιο 3

## Ανάλυση Χρονοσειρών

### 3.1 Εισαγωγή στις Χρονοσειρές

Σε πολλούς επιστημονικούς κλάδους και ειδικότερα της Στατιστικής επιδιώκεται σε διάφορες μελέτες να δημιουργηθούν μοντέλα τα οποία εκφράζουν την εξάρτηση ενός μεγέθους σε σχέση με άλλα μεγέθη. Τα μοντέλα αυτά αναπτύσσονται με βάση την γνώση των ερευνητών για το πρόβλημα που μελετάνε και ονομάζονται μοντέλα βασικών αρχών (First Principles). Παρόλα αυτά αρκετές φορές οι ερευνητές δεν γνωρίζουν τέτοιες σχέσεις για την ανάπτυξη μοντέλων βασικών αρχών ή δεν θέλουν να δημιουργήσουν τέτοιου είδους μοντέλα και η μελέτη των φαινομένων γίνεται με βάση την παρατήρηση των σχετικών μεγεθών. Τέτοιου είδους μοντέλα ονομάζονται εμπειρικά ή παραγόμενα από τα δεδομένα (Empirical or Data Driven Models). Στην παρούσα εργασία θα ασχοληθούμε με τέτοιου είδους μοντέλα, τα οποία δεν προϋποθέτουν γνώση του προβλήματος, αλλά στηρίζονται αποκλειστικά στα δεδομένα και ειδικότερα σε δεδομένα με χρονική διάταξη (χρονοσειρές) [18]. Πιο συγκεκριμένα, αναφερόμαστε σε προβλήματα στα οποία ένα μέγεθος αλλάζει τιμές με τον χρόνο και μελετάται η εξάρτηση του μεγέθους αυτού σε κάποια χρονική στιγμή  $t$ , σε σχέση με τις τιμές του ίδιου μεγέθους σε προηγούμενες χρονικές στιγμές  $t-1, t-2, \dots$ . Έτσι, λοιπόν εξετάζεται η εξέλιξη της διαδικασίας που παράγει το παρατηρούμενο μέγεθος [20].

Με τον όρο χρονοσειρά ή χρονολογική σειρά ονομάζεται μια ακολουθία  $x_t : t = 0, 1, 2, \dots$ , όπου κάθε  $x_t$  εκφράζει την κατά την χρονική στιγμή  $t$  κατάσταση ενός συστήματος, το οποίο εξελίσσεται στο χρόνο κατά τυχαίο εν γένει τρόπο [20].

Κάθε φαινόμενο που εξελίσσεται στον χρόνο είναι μια στοχαστική διαδικασία, έτσι συνεπάγεται ότι οι χρονοσειρές αποτελούν στοχαστικές διαδικασίες [16].

Γενικότερα, στις χρονοσειρές οι παρατηρήσεις της υπό εξέταση μεταβλητής παρουσιάζουν προκαθορισμένα και ίσα χρονικά διαστήματα, δηλαδή φέρουν σταθερό χρονικό βήμα ή αλλιώς σταθερή δειγματοληψία. Παρόλα αυτά σε ορισμένα προβλήματα ο χρόνος δειγματοληψίας δεν είναι σταθερός και σε αυτές τις περιπτώσεις χρειάζεται ειδική επεξεργασία της χρονοσειράς για να γίνει ανάλυση. Ένα παράδειγμα μεταβλητού χρόνου δειγματοληψίας είναι οι ημερήσιες τιμές των μετοχών ενός χρηματιστηριακού δείκτη, καθώς σε αυτή την περίπτωση μεσολαβούν Σαββατοκύριακα και αργίες, όπου το χρηματιστήριο παραμένει κλειστό. Σε αυτή την περίπτωση ορίζεται ως χρόνος αναφοράς ο οικονομικός χρόνος συναλλαγών και όχι ο φυσικός χρόνος, ενώ ακόμα θεωρείται σταθερό το χρονικό βήμα μιας οικονομικής ημέρας (Παρασκευή - Δευτέρα) [19] [20].

Το αντικείμενο το οποίο μελετάει τις χρονοσειρές είναι η ανάλυση χρονοσειρών. Η ανάλυση χρονοσειρών στοχεύει:

1. Στην αναζήτηση μαθηματικού μοντέλου που να εξηγεί και να περιγράφει το υπό μελέτη φαινόμενο.
2. Στην αναζήτηση κατάλληλου μοντέλου (όχι απαραίτητα του αληθινού), το οποίο να παράγει καλές προβλέψεις.

Στην παρούσα εργασία θα εξετάσουμε την αναζήτηση κατάλληλων μοντέλων με σκοπό την παραγωγή προβλέψεων σε διάφορα σύνολα δεδομένων. Γενικά, τα μοντέλα προβλέψεων των χρονοσειρών διακρίνονται σε γραμμικά ή μη-γραμμικά ανάλογα με το αν η συνάρτηση, η οποία περιγράφει το εκάστοτε μοντέλο είναι γραμμική ή όχι. Επιπλέον, διακρίνονται σε υποκειμενικές/ποιοτικές (subjective/qualitative) και σε αντικειμενικές/ποσοτικές (objective/quantitative). Στα υποκειμενικά μοντέλα οι προβλέψεις εφαρμόζονται από έμπειρους αναλυτές, οι οποίοι χρησιμοποιούν κατά κύριο λόγο την προσωπική τους κρίση, ενώ από την άλλη πλευρά στα αντικειμενικά μοντέλα οι προβλέψεις στηρίζονται αποκλειστικά σε κάποιο μαθηματικό μοντέλο. Τέλος, τα μοντέλα πρόβλεψης χρονοσειρών

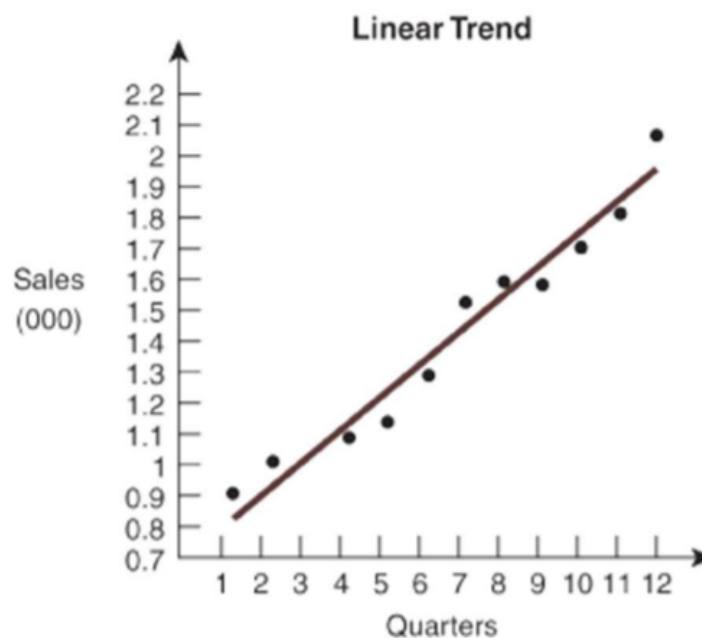
διαχωρίζονται σε αιτιατά και μη-αιτιατά. Αιτιατά αποκαλούνται τα μοντέλα, στα οποία οι προβλέψεις μια μεταβλητής βασίζονται σε μια οικονομική ή στατιστική σχέση, η οποία συνδέει την μεταβλητή αυτή με άλλες σχετιζόμενες μεταβλητές. Χαρακτηριστικό παράδειγμα τέτοιου είδους μοντέλου πρόβλεψης χρονοσειρών είναι τα οικονομετρικά μοντέλα. Μη-αιτιατά ονομάζονται τα μοντέλα, τα οποία για τις προβλέψεις τους στηρίζονται αποκλειστικά στις προηγούμενες τιμές της ίδιας χρονοσειράς [15].

### 3.2 Βασικά Χαρακτηριστικά Χρονοσειρών

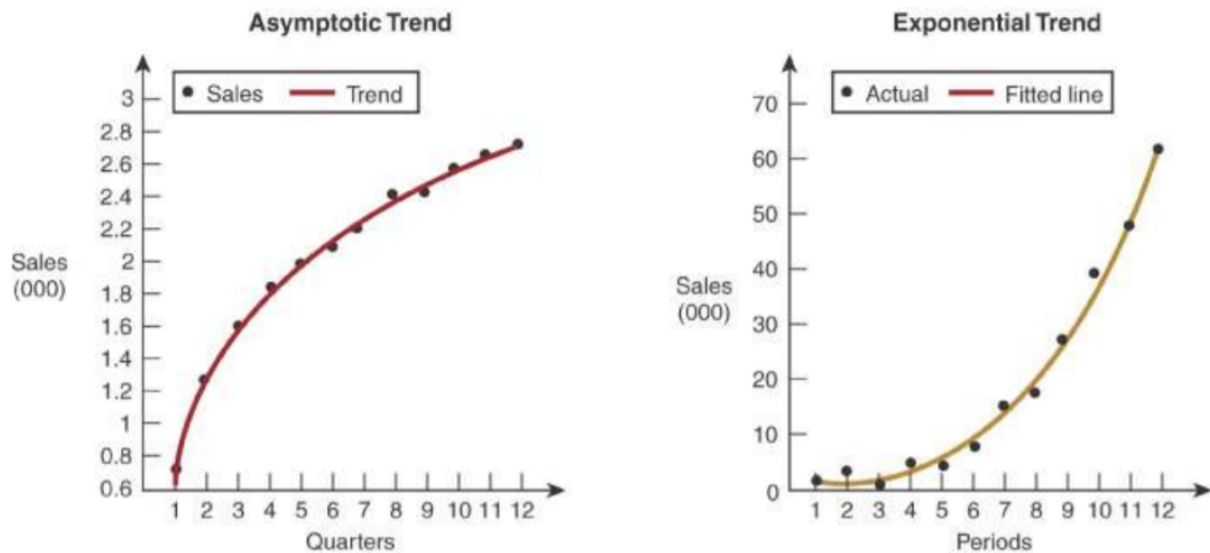
Οι πραγματικές χρονοσειρές παρουσιάζουν μια γενική μορφή, σύμφωνα με την οποία αποτελούνται από επιμέρους συνιστώσες. Έτσι, μια χρονοσειρά  $Y_t$  μπορεί να αποτελείται από τους παράγοντες της τάσης  $T_t$ , της εποχικότητας  $S_t$ , της κυκλικότητας  $C_t$  και των σφαλμάτων  $e_t$ , χωρίς αυτό να σημαίνει ότι πρέπει να τους περιέχει όλους ταυτόχρονα. Η γενική μορφή μιας χρονοσειράς είναι η εξής:

$$Y_t = T_t + S_t + C_t + e_t$$

- **Τάση (Trend):** Με τον όρο τάση αναφέρεται η διάθεση των δεδομένων μιας χρονοσειράς να παρουσιάζουν σταδιακές μετατοπίσεις σε υψηλότερες ή χαμηλότερες τιμές για μεγάλες χρονικές περιόδους. Η προδιάθεση των δεδομένων σε αυξητικές ή μειωτικές τάσεις οφείλεται σε μακροπρόθεσμους παράγοντες, όπως η τεχνολογία και οι προτιμήσεις των καταναλωτών. Μερικά από τα πιο συνηθισμένα είδη τάσης είναι η γραμμική τάση (Linear Trend), η εκθετική τάση (Exponential Trend) και η ασυμπτωτική τάση (Asymptotic Trend). Στα σχήματα 3.1 και 3.2 απεικονίζονται διαγράμματα χρονοσειρών με τάση [24].



Σχήμα 3.1: Διάγραμμα Χρονοσειράς με Γραμμική Τάση  
Πηγή: [24]



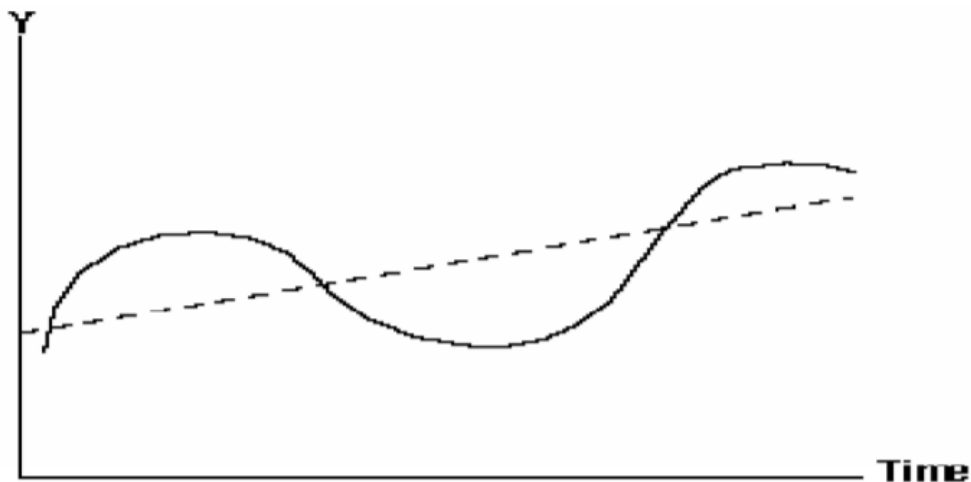
Σχήμα 3.2: Διάγραμμα Ασυμπτωτικής και Εκθετικής Τάσης  
Πηγή: [24]

- Εποχικότητα (Seasonality):** Με τον όρο εποχικότητα αναφέρονται οι περιοδικές διακυμάνσεις, οι οποίες εμφανίζονται κατά την διάρκεια επαναληπτικών ημερολογιακών περιόδων με διάρκεια επανάληψης μικρότερη του ενός έτους. Τα είδη της εποχικότητας συνήθως είναι ημερήσια, εβδομαδιαία, μηνιαία, τριμηνιαία και ετήσια ανάλογα με το πόσο συχνά εμφανίζονται οι διακυμάνσεις στα δεδομένα. Ένα παράδειγμα εποχικότητας αποτελεί η αύξηση της πώλησης παγωτών το καλοκαίρι και η μείωση της τον χειμώνα. Στο σχήμα 3.3 απεικονίζεται το διάγραμμα μιας χρονοσειράς με εποχικότητα [24].



Σχήμα 3.3: Διάγραμμα Χρονοσειράς με Εποχικότητα  
Πηγή: [24]

- **Κυκλικότητα (Cyclical):** Η συνιστώσα της κυκλικότητας αναφέρεται σε μακροπρόθεσμες διακυμάνσεις, οι οποίες εμφανίζονται γύρω από μια μακροπρόθεσμη τάση μιας χρονοσειράς. Ένα παράδειγμα εμφάνισης κυκλικότητας είναι ο εμπορικός κύκλος. Στο σχήμα 3.4 απεικονίζεται το διάγραμμα μιας χρονοσειράς, η οποία εμφανίζει κυκλικότητα [24].



Σχήμα 3.4: Διάγραμμα Χρονοσειράς με Κυκλικότητα  
Πηγή: [24]

#### Στασιμότητα:

Εκτός από τους παραπάνω συντελεστές από τους οποίους αποτελούνται οι χρονοσειρές, ένα χαρακτηριστικό που θα πρέπει να μελετάτε κατά την ανάλυση και την αναζήτηση του κατάλληλου μοντέλου που περιγράφει την εκάστοτε χρονοσειρά είναι η στασιμότητα. Γενικότερα, οι χρονοσειρές διακρίνονται σε στάσιμες και μη-στάσιμες ανάλογα με το αν τα σημαντικά τους χαρακτηριστικά (μέση τιμή και διασπορά) παραμένουν αναλλοίωτες στον χρόνο ή όχι. Πιο συγκεκριμένα, μια χρονοσειρά ονομάζεται αυστηρώς στάσιμη όταν οι κατανομές της για κάθε τάξη (ή ισοδύναμα όλες οι ροπές) είναι σταθερές στον χρόνο. Συνήθως στην ανάλυση χρονοσειρών εξετάζεται η στασιμότητα μιας χρονοσειράς με την χρήση του ασθενούς ορισμού της στασιμότητας. Έτσι, μια χρονοσειρά ονομάζεται ασθενώς στάσιμη όταν η μέση τιμή και η διακύμανση της δεν μεταβάλλονται με τον χρόνο και η συνδιακύμανση της σε δύο χρονικά σημεία της εξαρτάται αποκλειστικά από την απόσταση ανάμεσα σε αυτά τα χρονικά σημεία [16] [18].

Ιδανικά, για την μοντελοποίηση πρέπει η χρονοσειρά να είναι στάσιμη. Παρόλα αυτά στα πραγματικά δεδομένα οι στάσιμες χρονοσειρές είναι ιδιαίτερα σπάνιες. Έτσι, υπάρχουν διάφορες τεχνικές για την μετατροπή μη-στάσιμων χρονοσειρών σε στάσιμες. Η πιο χαρακτηριστική μέθοδος μετατροπής μη-στάσιμης χρονοσειράς σε στάσιμη είναι η μέθοδος των διαφορών (Difference Stationary Process). Σύμφωνα με αυτή την μέθοδο οι πρώτες διαφορές ή μεγαλύτερης τάξης διαφορές των τιμών της χρονοσειράς μετατρέπουν την νέα χρονοσειρά που προκύπτει από τις διαφορές των παρατηρήσεων της αρχικής χρονοσειράς σε στάσιμη. Οι χρονοσειρές που ανήκουν σε αυτή την κατηγορία ονομάστηκαν από τους Nelson και Plosser το 1982 στάσιμες σε διαφορές. Τέτοιες χρονοσειρές είναι οι περισσότερες χρηματοοικονομικές. Ο τελεστής διαφορών (Backshift Operator) εφαρμόζεται σε μια χρονοσειρά και υπολογίζει τις διαφορές για τις τιμές της ανα δύο. Συνήθως, οι χρονοσειρές μετατρέπονται σε στάσιμες μετά την πρώτη διαφορά των τιμών τους και σπάνια μετά την δεύτερη.

Για τον έλεγχο της στασιμότητας των χρονοσειρών υπάρχουν διάφοροι έλεγχοι και εμπειρικοί τρόποι, όπως η οπτική απεικόνιση μιας χρονοσειράς. Μια χρονοσειρά χωρίς μεγάλη διασπορά στις τιμές της, χωρίς εμφανή εποχικότητα και τάση είναι στάσιμη, ενώ μια χρονοσειρά που δεν πληρεί αυτές τις προϋποθέσεις είναι μη-στάσιμη. Παρόλα αυτά η οπτική απεικόνιση των χρονοσειρών ως μέθοδος ελέγχου στασιμότητας είναι υποκειμενική, καθώς περιέχει την προσωπική κρίση του κάθε αναλυτή και άρα ελλοχεύει ο κίνδυνος λανθασμένης απόφασης. Έτσι, στην εργασία αυτή θα επικεντρωθούμε στον έλεγχο στασιμότητας Augmented Dickey-Fuller (adf test). Στο adf test η μηδενική υπόθεση του ελέγχου  $H_0$  υποστηρίζει ότι στην υπό έλεγχο χρονοσειρά υπάρχει μοναδική ρίζα, ενώ η εναλλακτική υπόθεση  $H_1$  υποστηρίζει ότι η χρονοσειρά είναι στάσιμη [25].



### 3.3 Στατιστικά Μοντέλα Πρόβλεψης Στάσιμων Χρονοσειρών

Μέσα από την ενασχόληση των επιστημόνων με την ανάλυση χρονοσειρών έχουν αναπτυχθεί πολλά μοντέλα για την περιγραφή χρονοσειρών και για την δημιουργία προβλέψεων. Τα μοντέλα αυτά αφορούν στάσιμες και μη-στάσιμες χρονοσειρές, ενώ ένα μοντέλο μπορεί να περιγράψει κατάλληλα μια μη-στάσιμη χρονοσειρά, αλλά να μην περιγράψει κατάλληλα μια άλλη μη-στάσιμη χρονοσειρά. Μερικά από τα πιο χαρακτηριστικά μοντέλα πρόβλεψης στάσιμων χρονοσειρών είναι τα παρακάτω:

- **Μοντέλο Κινητού Μέσου (Moving Average, MA(q)):** Το μοντέλο αυτό είναι ένα από τα πιο διαδεδομένα και απλοϊκά μοντέλα πρόβλεψης και περιγραφής χρονοσειρών. Το μοντέλο κινητού μέσου απλά παραθέτει ότι η επόμενη παρατήρηση της χρονοσειράς  $Y_t$  είναι η μέση τιμή των  $T$  πιο πρόσφατων προηγούμενων παρατηρήσεων της, δηλαδή έχουμε:

$$Y_{t+1} = \frac{Y_1 + Y_2 + \dots + Y_t}{T} = m + e_t - u_1 e_{t-1} - u_2 e_{t-2} - \dots - u_q e_{t-q}$$

,όπου  $m$  είναι ο μέσος όρος της χρονοσειράς,  $u_1, u_2, \dots, u_q$  είναι οι παράμετροι του κινητού μέσου,  $e_t, e_{t-1}, e_{t-2}, \dots, e_{t-j}$  είναι τα σφάλματα (υπόλοιπα λευκού θορύβου) και  $j = (0, q)$  το πλήθος της υστέρησης.

Κάθε φορά που προστίθεται μία παρατήρηση στο δείγμα παραλείπεται η πιο παλιά, έτσι ώστε υπάρχει σταθερή περίοδος υπολογισμού του μέσου. Από αυτή την διαδικασία το μοντέλο αυτό ονομάστηκε μοντέλο κινητού μέσου. Τέλος, παρά την απλότητα του το μοντέλο κινητού μέσου παρουσιάζει αρκετά καλά αποτελέσματα και αποτέλεσε ένα σημαντικό βήμα για την εξέλιξη των μοντέλων πρόβλεψης χρονοσειρών [15] [25] [26].

- **Αυτοπαλίνδρομο Μοντέλο (Autoregressive, AR(p)):** Το αυτοπαλίνδρομο μοντέλο και το μοντέλο κινητού μέσου αποτελούν χαρακτηριστικά παραδείγματα μη-αιτιατών μοντέλων πρόβλεψης. Όπως έχει αναφερθεί στα μοντέλα αυτά οι προβλέψεις παράγονται με βάση τις παρελθοντικές τιμές της χρονοσειράς. Πιο συγκεκριμένα, το αυτοπαλίνδρομο μοντέλο αποτελεί μια παλινδρόμηση της χρονοσειράς με τον εαυτό της, όπου η τιμή της την χρονική στιγμή  $t$  εξαρτάται από τις προηγούμενες τιμές της με κάποια υστέρηση (lag). Ο μέγιστος αριθμός υστέρησης (max lag) του εκάστοτε αυτοπαλίνδρομου μοντέλου ορίζεται ως μια παράμετρος  $p$ . Η γενική σχέση ενός αυτοπαλίνδρομου μοντέλου είναι η εξής:

$$Y_t = a_1 Y_{t-1} + a_2 Y_{t-2} + \dots + a_p Y_{t-p} + e_t$$

,όπου  $a_1, a_2, \dots, a_p$  είναι οι παράμετροι αυτοπαλίνδρομου μοντέλου,  $e_t$  τα σφάλματα (υπόλοιπα λευκού θορύβου),  $Y_t, Y_{t-1}, Y_{t-2}, \dots, Y_{t-j}$  οι τιμές της χρονοσειράς με τις παρελθοντικές τιμές της και  $j = (0, q)$  το πλήθος των υστέρησης της χρονοσειράς [19] [25].

- **Μικτό Μοντέλο (ARMA(p, q)):** Το μικτό αυτοπαλίνδρομο κινητού μέσου μοντέλο αποτελεί συνδυασμό των δυο μοντέλων πρόβλεψης χρονοσειρών που περιγράφηκαν παραπάνω, δηλαδή του μοντέλου AR και του μοντέλου MA. Το μοντέλο αυτό αναφέρεται αποκλειστικά σε στάσιμες χρονοσειρές και είναι ιδιαίτερα χρήσιμο στην ερμηνεία μεγάλου όγκου δεδομένων, κυρίως από τον κλάδο της οικονομίας και άλλων θετικών επιστημών. Το γενικό μοντέλο ARMA παρουσιάστηκε πρώτη φορά το 1951 από τον Peter Whittle στην διατριβή του με τίτλο "Hypothesis Testing in Time Series Analysis", ενώ έγινε ευρέως γνωστό από τους George E. P. Box και Gwilym Jenkins με την παρουσίαση του βιβλίου τους το 1970. Ένα μοντέλο ARMA μπορεί να περιγράψει μια χρονοσειρά είτε αποκλειστικά ως μοντέλο AR, είτε αποκλειστικά ως μοντέλο MA, είτε ως συνδυασμός και των δύο ARMA. Η γενική εξίσωση ενός αυτοπαλίνδρομου μοντέλου είναι η εξής:

$$Y_t = d + a_1 Y_{t-1} + a_2 Y_{t-2} + \dots + a_p Y_{t-p} + e_t - u_1 e_{t-1} - u_2 e_{t-2} - \dots - u_q e_{t-q}$$

,όπου  $d$  είναι η σταθερά της εξίσωσης του μοντέλου και όλα τα υπόλοιπα είναι τα χαρακτηριστικά των AR και MA μοντέλων αντίστοιχα, των οποίων οι γενικές εξισώσεις περιγράφηκαν παραπάνω [15] [27].

### 3.4 Στατιστικά Μοντέλα Πρόβλεψης Μη-Στάσιμων Χρονοσειρών

Στις προηγούμενες ενότητες περιγράφηκαν οι κατηγορίες των στάσιμων και μη-στάσιμων χρονοσειρών. Στην διαδικασία της ανάλυσης χρονοσειρών οι αναλυτές επιδιώκουν να αναλύουν στάσιμες χρονοσειρές ή τουλάχιστον χρονοσειρές, οι οποίες μπορούν να γίνουν στάσιμες με διάφορες τεχνικές (π.χ. Backshift Operator). Η προτίμηση σε αυτό το είδος χρονοσειρών οφείλεται στα χαρακτηριστικά τους (σταθερή μέση τιμή και διακύμανση) και στην αδυναμία των ερευνητών να περιγράψουν με μαθηματικά μοντέλα μη-στάσιμες χρονοσειρές. Τα πιο χαρακτηριστικά μοντέλα πρόβλεψης μη-στάσιμων χρονοσειρών είναι τα παρακάτω:

- **Ολοκληρωμένο Μικτό Μοντέλο ARIMA(p, d, q):** Το μικτό μοντέλο ARIMA αποτελεί συνδυασμό των μοντέλων AR και MA, όπως το μοντέλο ARMA που περιγράφηκε προηγουμένως, με την διαφορά ότι οι χρονοσειρές που περιγράφει είναι μη-στάσιμες και μετατρέπονται σε στάσιμες με την μέθοδο των διαφορών. Τέτοιου είδους χρονοσειρές συνήθως εμφανίζουν τάση, ενώ παραδείγματα τέτοιων χρονοσειρών είναι αυτή του ακαθάριστου εγχώριου προϊόντος (Α.Ε.Π.) και των δημόσιων δαπανών. Στο μικτό μοντέλο ARIMA εκτός από τους συντελεστές των μοντέλων AR και MA, υπάρχει και ο συντελεστής των διαφορών που απαιτούνται, ώστε να μετατραπεί η σειρά σε στάσιμη. Ένα μοντέλο ARIMA για την περιγραφή μιας χρονοσειράς μπορεί να μην περιέχει καθόλου έναν από τους συντελεστές AR και MA, αλλά μπορεί να αποτελείται και από τους δύο. Η γενική εξίσωση ενός μικτού μοντέλου ARIMA είναι η εξής:

$$(1 - \sum_{i=1}^p a_i L^i)(1 - L)^d Y_t = (1 + \sum_{i=1}^q u_i L^i) e_t$$

,όπου  $L$  είναι ο Backshift Operator,  $a_i$  είναι οι παράμετροι του αυτοπαλίνδρομου μοντέλου,  $u_i$  είναι οι παράμετροι του κινητού μέσου,  $d$  ο αριθμός των διαφορών που απαιτούνται για να μετατραπεί η χρονοσειρά σε στάσιμη και  $e_t$  είναι τα σφάλματα [15] [28].

- **Εποχιακό Ολοκληρωμένο Μικτό Μοντέλο SARIMA(p, d, q)(P, D, Q):** Το εποχιακό μικτό μοντέλο SARIMA αποτελεί μια ειδική περίπτωση του μικτού μοντέλου ARIMA που περιγράφηκε προηγουμένως, για μη-στάσιμες χρονοσειρές με εποχικότητα. Έτσι, το μοντέλο SARIMA αποτελείται από τους συντελεστές του ARIMA με την προσθήκη τριών εποχικών συντελεστών. Πιο συγκεκριμένα, στο μοντέλο αυτό προστίθενται τρεις υπερπαράμετροι για τον προσδιορισμό του εποχικού αυτοπαλίνδρομου συντελεστή, του εποχικού συντελεστή κινητού μέσου και του εποχικού συντελεστή διαφορών. Επιπρόσθετα, προστίθεται μια ξεχωριστή υπερπαράμετρος για το προσδιορισμό της περιόδου της εποχικότητας (π.χ.  $s=12$  μήνες αν η χρονοσειρά παρουσιάζει ετήσια εποχικότητα). Όμοια, με το μοντέλο ARIMA, το μοντέλο SARIMA για την περιγραφή μιας χρονοσειράς μπορεί να μην περιέχει έναν ή και περισσότερους από τους συντελεστές AR και MA (είτε τους εποχιακούς, είτε τους μη-εποχιακούς), αλλά μπορεί να τους περιέχει και όλους. Η γενική εξίσωση ενός εποχιακού μικτού μοντέλου SARIMA είναι η εξής:

$$A(L)F(L^s)(1 - L)^d(1 - L^s)^D Y_t = u(L)u(L^s)e_t$$

,όπου  $D$  ο αριθμός που πρέπει να εφαρμοστεί η τεχνική των διαφορών στο εποχικό τμήμα για να γίνει στάσιμη η χρονοσειρά,  $s$  η περίοδος της εποχικότητας,  $d$  ο αριθμός που πρέπει να εφαρμοστεί η τεχνική των διαφορών στο μη-εποχικό τμήμα ώστε να γίνει στάσιμη η χρονοσειρά και  $L$  ο Backshift Operator. Επίσης,  $A(L)$  είναι ο μη-εποχικός αυτοπαλίνδρομος συντελεστής,  $F(L^s)$  ο εποχικός αυτοπαλίνδρομος συντελεστής,  $u(L)$  ο μη-εποχικός συντελεστής κινητού μέσου,  $u(L^s)$  ο εποχικός συντελεστής κινητού μέσου και  $e_t$  τα σφάλματα.

Το μοντέλο SARIMA σε σχέση με τον μοντέλο ARIMA προσπαθεί να παράγει πιο αξιόπιστες προβλέψεις, χαρτογραφώντας την εποχικότητα και την τάση που πολλές πραγματικές χρονοσειρές παρουσιάζουν [15].

### 3.5 Τρόποι Επιλογής του Καταλληλότερου Στατιστικού Μοντέλου

Στην σημερινή εποχή υπάρχουν δύο τρόποι επιλογής του μοντέλου που παριστάνει πιο ικανοποιητικά μια χρονοσειρά. Οι τρόποι αυτοί είναι είτε με την χρήση διάφορων γλωσσών προγραμματισμού, όπως η Python, όπου με έτοιμα πακέτα (βλ. κώδικα εργασίας) και το κριτήριο  $AIC$  ως μέσο σύγκρισης των υποψήφιων μοντέλων επιλέγεται το καταλληλότερο μοντέλο, είτε με τον παραδοσιακό τρόπο επιλογής του μοντέλου με την προσωπική κρίση του ερευνητή (μεθοδολογία Box-Jenkins) [16].

- **Μεθοδολογία Box-Jenkins:** Οι Box-Jenkins παρουσίασαν το 1976 την μεθοδολογία τους για την εύρεση του στατιστικού μοντέλου ARIMA, το οποίο παριστάνει πιο ικανοποιητικά την χρονοσειρά από την οποία προέρχεται το δείγμα μας. Η μέθοδος αυτή περιλαμβάνει τρία στάδια, ενώ χρησιμοποιείται και για την εύρεση μοντέλων ARMA και SARIMA.
  1. **Ταυτοποίηση:** Στο συγκεκριμένο στάδιο πραγματοποιείται εξειδίκευση του μοντέλου με βάση τις πληροφορίες του δείγματος. Πιο συγκεκριμένα, καθορίζονται οι τιμές των παραμέτρων  $p$ ,  $d$ ,  $q$ , δηλαδή εξετάζεται η παράμετρος  $d$  των διαφορών που χρειάζονται για να μετατραπεί η χρονοσειρά σε στάσιμη, εφόσον δεν είναι ήδη. Επιπρόσθετα, εξετάζεται η τάξη  $p$  της αυτοπαλίνδρομης διαδικασίας και η τάξη  $q$  της διαδικασίας κινητού μέσου. Στο στάδιο αυτό για να διαπιστωθεί η στασιμότητα της χρονοσειράς εξετάζεται η δειγματική συμπεριφορά της συνάρτησης αυτοσυσχέτισης. Έτσι, αν οι αυτοσυσχετίσεις συγκλίνουν γρήγορα στο μηδέν, τότε η χρονοσειρά μάλλον είναι στάσιμη. Από την άλλη πλευρά, αν οι αυτοσυσχετίσεις φθίνουν με αργό ρυθμό, τότε η χρονοσειρά πιθανόν να μην είναι στάσιμη.
  2. **Εκτίμηση:** Στο στάδιο της εκτίμησης εκτιμώνται οι συντελεστές του αυτοπαλίνδρομου μοντέλου και του μοντέλου κινητού μέσου. Οι μέθοδοι εκτίμησης, που καθορίζουν τον αριθμό των υπο εκτίμηση παραμέτρων ποικίλουν ανάλογα με την μορφή και την τάξη του μοντέλου. Χαρακτηριστική μέθοδος εκτίμησης είναι με την χρήση των διαγραμμάτων αυτοσυσχέτισης ACF και μερικής αυτοσυσχέτισης PACF.
  3. **Διαγνωστικός Έλεγχος:** Στο στάδιο του διαγνωστικού ελέγχου εξετάζεται η ικανότητα του μοντέλου που καθορίστηκε στα προηγούμενα στάδια να περιγράφει επαρκώς την υπό ανάλυση χρονοσειρά. Ο έλεγχος της καταλληλότητας του μοντέλου γίνεται με βάση τα παρακάτω κριτήρια:
    - Την σημαντικότητα των συντελεστών του μοντέλου
    - Την σταθερότητα των συντελεστών του μοντέλου
    - Τις ιδιότητες των καταλοίπων (πρέπει να είναι ασυσχέτιστα)
    - Την προβλεπτική ικανότητα του μοντέλου
  4. **Πρόβλεψη:** Μετά την ολοκλήρωση των τριών παραπάνω σταδίων ακολουθεί η πρόβλεψη για τις μελλοντικές τιμές της χρονοσειράς, με βάση το ιδανικότερο μοντέλο που εκτιμήθηκε στα προηγούμενα στάδια.
- **Με την χρήση προγραμματιστικών πακέτων:** Στην σημερινή εποχή με την ανάπτυξη της τεχνολογίας των υπολογιστών αναπτύχθηκαν σταδιακά διάφορες γλώσσες προγραμματισμού και πακέτα, τα οποία είναι ικανά να πραγματοποιούν όλες τις παραπάνω διαδικασίες αυτόματα. Με την χρήση των πακέτων αυτών ελαχιστοποιείται η πιθανότητα λανθασμένης επιλογής μοντέλου από τον αναλυτή και αυξάνεται η ακρίβεια στις μελλοντικές προβλέψεις (βλ. Κεφάλαιο 4).

### Διάγραμμα Αυτοσυσχέτισης (ACF):

ACF είναι η (πλήρης) συνάρτηση αυτοσυσχέτισης, η οποία υπολογίζει τις τιμές της αυτοσυσχέτισης οποιασδήποτε χρονοσειράς με τις τιμές της με υστέρηση (lag). Σχεδιάζοντας τις τιμές αυτοσυσχέτισης αυτές προκύπτει το διάγραμμα ACF. Με απλά λόγια το διάγραμμα ACF περιγράφει πόσο καλά η σημερινή τιμή της χρονοσειράς σχετίζεται με τις τιμές της τις προηγούμενες χρονικές στιγμές. Μια χρονοσειρά μπορεί να έχει τάση, εποχικότητα, κυκλικότητα και κατάλοιπα. Η συνάρτηση αυτοσυσχέτισης συνυπολογίζει αυτές τις συνιστώσες, καθώς υπολογίζει τις αυτοσυσχετίσεις και για αυτό ονομάζεται και πλήρης συνάρτηση αυτοσυσχέτισης. Στο διάγραμμα αυτοσυσχέτισης μιας χρονοσειράς στον άξονα  $x$  απεικονίζεται ο αριθμός των υστερήσεων (lag) και στον άξονα  $y$  οι συσχετίσεις [22] [29].

### Διάγραμμα Μερικής Αυτοσυσχέτισης (PACF):

PACF είναι η συνάρτηση μερικής αυτοσυσχέτισης, η οποία αντί να υπολογίζει τις συσχετίσεις της τωρινής τιμής της χρονοσειράς με τις προηγούμενες τιμές της (με υστέρηση), υπολογίζει την συσχέτιση των καταλοίπων (τα οποία παραμένουν μετά την αφαίρεση των επιδράσεων που εξηγούνται από τις προηγούμενες υστερήσεις) με την επόμενη τιμή υστέρησης (lag). Για τον λόγο αυτό η συνάρτηση αυτή ονομάζεται μερικής αυτοσυσχέτισης και όχι πλήρης, καθώς αφαιρούνται διαπιστωμένες μεταβολές πριν υπολογιστεί η επόμενη συσχέτιση [29].

Για την εκτίμηση των παραμέτρων  $p$ ,  $q$  των μοντέλων AR και MA κατά την μεθοδολογία Box-Jenkins με την χρήση των διαγραμμάτων ACF και PACF υπάρχουν ορισμένοι εμπειρικοί κανόνες. Οι αναλυτές στηρίζονται στους κανόνες αυτούς, ώστε να εκτιμήσουν τις παραμέτρους του ιδανικότερου μοντέλου για την εκάστοτε χρονοσειρά. Μερικοί από αυτούς τους κανόνες είναι οι παρακάτω:

- Αν το διάγραμμα PACF μιας χρονοσειράς, η οποία έχει γίνει στάσιμη με την μέθοδο των διαφορών εμφανίζει απότομη διακοπή και η αυτοσυσχέτιση της χρονοσειράς με την πρώτη υστέρηση  $\text{lag}(1)$  είναι θετική, τότε είναι πιθανό η χρονοσειρά αυτή να περιγράφεται κατάλληλα με την προσθήκη ενός αυτοπαλίνδρομου όρου AR στο μοντέλο. Η υστέρηση στην οποία υφίσταται η απότομη διακοπή στο διάγραμμα PACF είναι και ο αριθμός της παραμέτρου  $p$  του αυτοπαλίνδρομου όρου του μοντέλου [30].
- Αν το διάγραμμα ACF μιας χρονοσειράς, η οποία έχει γίνει στάσιμη με την μέθοδο των διαφορών εμφανίζει απότομη διακοπή και η αυτοσυσχέτιση της χρονοσειράς με την πρώτη υστέρηση  $\text{lag}(1)$  είναι αρνητική, τότε είναι πιθανό η χρονοσειρά που μελετάτε να περιγράφεται κατάλληλα με την προσθήκη ενός όρου κινητού μέσου MA στο μοντέλο. Η υστέρηση στην οποία υφίσταται η απότομη διακοπή στο διάγραμμα ACF είναι και ο αριθμός της παραμέτρου  $q$  του όρου του κινητού μέσου που προστίθεται στο μοντέλο [30].

## Κεφάλαιο 4

# Περιγραφή της Εφαρμογής

Στο κεφάλαιο αυτό γίνεται περιγραφή της εφαρμογής που υλοποιήθηκε στα πλαίσια της διπλωματικής εργασίας με τίτλο “Εφαρμογή Τεχνητών Νευρωνικών Δικτύων και Στατιστικών Μοντέλων στην Πρόβλεψη Χρονοσειρών”. Η βασική ιδέα της εφαρμογής κινήθηκε σε δύο άξονες. Ο πρώτος άξονας αφορά την επιλογή, την δημιουργία και την εκπαίδευση του καταλληλότερου πολυεπίπεδου δικτύου Perceptron με βάση ένα σύνολο δεδομένων και τέλος την δημιουργία μελλοντικών προβλέψεων για το σύνολο δεδομένων αυτό. Από την άλλη πλευρά, ο δεύτερος άξονας πραγματεύεται την επιλογή, την δημιουργία και την εκπαίδευση των καταλληλότερων στατιστικών μοντέλων (ARMA, ARIMA, SARIMA) με βάση τρία νέα σύνολα δεδομένων και τέλος την δημιουργία μελλοντικών προβλέψεων για τα σύνολα δεδομένων που μελετήθηκαν.

Η παρούσα εργασία υλοποιήθηκε στην γλώσσα προγραμματισμού Python με χρήση της διαδικτυακής πλατφόρμας Google Drive και του Google Colaboratory.

### 4.1 Υπάρχοντα Μοντέλα Πρόβλεψης με Χρήση Τεχνητών Νευρωνικών Δικτύων

Στην σημερινή εποχή, τα συστήματα λήψης αποφάσεων και η προσπάθεια εξαγωγής έγγυρων προβλέψεων είναι συνυφασμένα με την έννοια της ανάπτυξης και της προόδου τόσο της επιστήμης, όσο και των εταιρειών. Έτσι, τα μοντέλα πρόβλεψης συμβάλλουν καταλυτικά στην λήψη αποφάσεων από τους ερευνητές και τις εταιρείες. Ο παροξυσμός για την συλλογή, ανάλυση και εξαγωγή προβλέψεων και συμπερασμάτων μέσα από τα δεδομένα προέρχεται από την ανάγκη τόσο των επιστημόνων, όσο και των εταιρειών να έχουν πρόσβαση σε πληροφορίες με όσο το δυνατόν μεγαλύτερη ακρίβεια γίνεται. Η διεθνοποίηση της οικονομίας και η ανάγκη για κέρδος ώθησε σταδιακά στην προσπάθεια δημιουργίας προβλέψεων για τους χρηματιστηριακούς δείκτες. Έτσι, με την πάροδο των χρόνων αναπτύχθηκαν ποικίλες τεχνικές προβλέψεων, όπως η βασική ανάλυση (Fundamental Analysis), η τεχνική ανάλυση (Technical Analysis), τα μοντέλα μηχανικής μάθησης και ειδικότερα τα τεχνητά νευρωνικά δίκτυα. Γενικότερα, οι χρηματιστηριακοί δείκτες παρουσιάζουν μη-γραμμικά και χαοτικά μοτίβα, τα οποία μπορεί να επηρεάζονται άμεσα ή έμμεσα από πολλές μεταβλητές [39]. Το γεγονός αυτό καθιστά αδύνατο να μελετηθούν και κατέπεκταση να δημιουργηθούν προβλέψεις για αυτούς με τα απλά συμβατά μοντέλα προβλέψεων. Από την άλλη πλευρά, τα τεχνητά νευρωνικά δίκτυα είναι ικανά να εκπαιδεύονται και να μαθαίνουν μη-γραμμικά, χαοτικά συστήματα, παρέχοντας εγγυρότερες προβλέψεις. Έτσι, λόγω της ικανότητας αυτής δημιουργήθηκαν αρκετά διαφορετικά μοντέλα προβλέψεων με χρήση τεχνητών νευρωνικών δικτύων. Μερικά από τα οποία περιγράφονται παρακάτω:

Ένα από τα μοντέλα που χρησιμοποιήθηκαν για την παροχή προβλέψεων για τους δείκτες του χρηματιστηρίου περιγράφεται στο άρθρο με τίτλο “Price Prediction of Share Market Using Artificial Neural Network (ANN)” [37]. Στο άρθρο αυτό οι ερευνητές χρησιμοποιούν ένα πολυεπίπεδο δίκτυο πρόσθιας τροφοδότησης (Feedforward Network) με ένα επίπεδο εισόδου, ένα επίπεδο εξόδου και ένα κρυφό επίπεδο. Το επίπεδο εισόδου όπως και το κρυφό επίπεδο αποτελούνται από 5 νευρώνες αντίστοιχα, ενώ το επίπεδο εξόδου αποτελείται από έναν νευρώνα, ο οποίος τελικώς παράγει και την τελική πρόβλεψη. Ως εισοδοί του νευρωνικού δικτύου εισάγονται 5 μεταβλητές, οι οποίες είναι ο γενικός δείκτης (General Index), ο λόγος P/E, η καθαρή αξία του ενεργητικού (Net Asset Value), τα κέρδη ανά μετοχή (Earnings per Share) και το Volume. Το νευρωνικό δίκτυο αυτό εκπαιδεύεται με τον αλγόριθμο οπισθοδιάδοσης, γνωστό και ως Backpropagation Algorithm. Πριν την διαδικασία της εκπαίδευσης του νευρωνικού δικτύου οι εισοδοί των 5 μεταβλητών που αναφέρθηκαν προηγουμένως κανονικοποιούνται και στην συνέχεια εισάγονται στο δίκτυο. Με την εισαγωγή των κανονικοποιημένων εισόδων στον νευρωνικό δίκτυο ξεκινάει το στάδιο της εκπαίδευσης. Έτσι, οι εισοδοί που εισάγονται στο δίκτυο πολλαπλασιάζονται με τα αντίστοιχα βάρη και

εισάγονται στους νευρώνες του κρυφού επιπέδου του δικτύου, όπου ο κάθε νευρώνας του κρυφού επιπέδου παράγει μια έξοδο. Οι νευρώνες του κρυφού επιπέδου του δικτύου αυτού παρουσιάζουν την ίδια συνάρτηση ενεργοποίησης. Οι έξοδοι του κρυφού επιπέδου θεωρούνται οι εισόδοι για το επίπεδο εξόδου. Με την ίδια διαδικασία και όπως από το επίπεδο εισόδου προς το κρυφό επίπεδο, οι έξοδοι του κρυφού επιπέδου πολλαπλασιάζονται με τα αντίστοιχα βάρη και εισάγονται στο επίπεδο εξόδου, όπου και παράγεται η έξοδος. Στην συνέχεια, υπολογίζεται το σφάλμα μεταξύ της τιμής που παράγει το δίκτυο και της τιμής στόχου. Με βάση το σφάλμα αυτό το νευρωνικό δίκτυο τροποποιεί τα βάρη των επιπέδων του με σκοπό το μέσο τετραγωνικό σφάλμα  $mse$  του να είναι μηδέν ή πολύ κοντά σε αυτό. Όταν κατά την διαδικασία εκπαίδευσης το μέσο τετραγωνικό σφάλμα πλησιάζει το μηδέν ή παίρνει την τιμή μηδέν, τότε η διαδικασία της εκπαίδευσης σταματάει. Με το πέρας της διαδικασίας της εκπαίδευσης υπάρχει το στάδιο της πρόβλεψης. Στο στάδιο αυτό εισάγονται στο εκπαιδευμένο δίκτυο τα δεδομένα εισόδου με βάση τα οποία οι ερευνητές θέλουν να πραγματοποιηθεί η πρόβλεψη για την τιμή του χρηματιστηρίου και αυτό με την σειρά του παράγει την τελική τιμή πρόβλεψης.

Στο άρθρο με τίτλο “Stock Market Forecasting Using Artificial Neural Networks” χρησιμοποιείται ένα μοντέλο τεχνητού νευρωνικού δικτύου για την δημιουργία προβλέψεων για τις μετοχές της αυτοκινητοβιομηχανίας Iran Khodro [38]. Στο άρθρο αυτό οι ερευνητές χρησιμοποιούν ένα πολυεπίπεδο νευρωνικό δίκτυο με ένα επίπεδο εισόδου, ένα επίπεδο εξόδου και τρία κρυφά επίπεδα. Το επίπεδο εισόδου αποτελείται από 22 νευρώνες, το επίπεδο εξόδου από 4, ενώ τα κρυφά επίπεδα αποτελούνται από 22, 9 και 4 νευρώνες αντίστοιχα. Στο νευρωνικό δίκτυο εισάγονται ως εισόδοι 22 μεταβλητές, οι οποίες σχετίζονται με την τιμή της υπό εξέταση μετοχής. Μερικές από τις μεταβλητές εισόδου είναι η ρευστότητα (Liquidity), η προστιθέμενη αξία (Added Value) και η αξία παραγωγής βιομηχανικών χώρων εργασίας. Οι 4 νευρώνες από τους οποίους αποτελείται το επίπεδο εξόδου του δικτύου παράγουν 4 εξόδους (μια έξοδο ο κάθε νευρώνας). Οι έξοδοι που παράγονται είναι η τελευταία τιμή των μετοχών της Iran Khodro, οι αλλαγές στην τιμή των μετοχών της Iran Khodro, η χαμηλότερη τιμή της μετοχής της Iran Khodro, η υψηλότερη τιμή της μετοχής της Iran Khodro. Το νευρωνικό δίκτυο εκπαιδεύεται, λαμβάνοντας υπόψη τις εισερχόμενες εισόδους, τα δεδομένα στην έξοδο και με την χρήση του αλγόριθμου εκπαίδευσης Train GD. Όμοια με όλα τα νευρωνικά δίκτυα και σε αυτό το μοντέλο οι ερευνητές κανονικοποίησαν τα δεδομένα εισόδου πριν την εκπαίδευση του νευρωνικού δικτύου, ενώ ως μέτρο ακρίβειας των προβλέψεων του δικτύου χρησιμοποιήθηκε το μέσο τετραγωνικό σφάλμα ( $mse$ ).

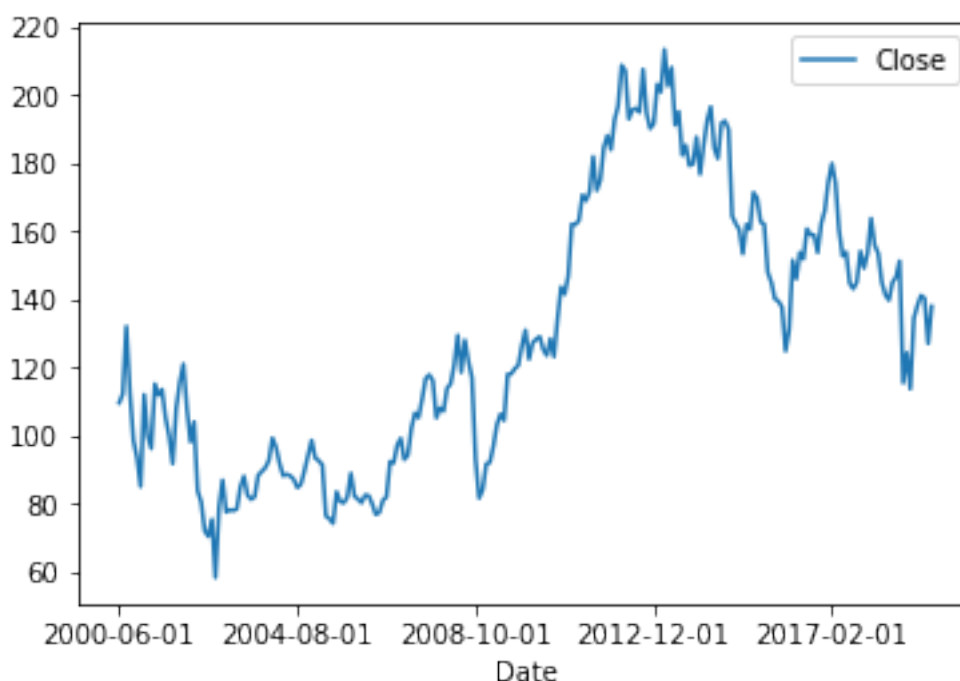
## 4.2 Πολυεπίπεδο Τεχνητό Νευρωνικό Δίκτυο

Στο παρόν εδάφιο θα γίνει η παρουσίαση μιας πολύ απλής προσέγγισης για πρόβλεψη της συμπεριφοράς της μετοχής της IBM εκπαιδεύοντας ένα πολυεπίπεδο νευρωνικό δίκτυο Perceptron, του οποίου η δομή σχεδιάστηκε εξετάζοντας ένα σύνολο με παραμέτρους. Για την επιλογή, την εκπαίδευση και τελικά την δημιουργία μελλοντικών προβλέψεων με βάση το καταλληλότερο πολυεπίπεδο δίκτυο Perceptron χρησιμοποιήθηκε το διάλυμα παρατηρήσεων  $df$ . Το διάλυμα παρατηρήσεων αυτό προέρχεται από την ηλεκτρονική σελίδα Yahoo Finance [23] και αποτελείται από 7 διανύσματα παρατηρήσεων από 241 παρατηρήσεις. Το πρώτο διάνυσμα παρατηρήσεων περιλαμβάνει τις ημερομηνίες καταγραφής των παρατηρήσεων, ενώ τα υπόλοιπα είναι με την σειρά τα εξής: Open, High, Low, Close, Adj Close, Volume. Η παρούσα εφαρμογή πραγματεύεται τις τιμές κλεισίματος της μετοχής, δηλαδή την στήλη με όνομα Close. Για να έχουμε πρόσβαση μόνο στο διάλυμα των παρατηρήσεων των τιμών κλεισίματος της IBM χρησιμοποιούμε κατάλληλη εντολή (βλ. ενότητα 4.4.1), με την οποία ορίζουμε το διάλυμα παρατηρήσεων αυτό ως  $w$ .

Στο πρώτο στάδιο της εφαρμογής το διάλυμα παρατηρήσεων  $w$  χωρίζεται σε δυό μικρότερα διανύσματα παρατηρήσεων, όπου το πρώτο  $a$  αποτελείται από τις πρώτες 230 παρατηρήσεις του  $w$  (παλαιότερες προς νεότερες) και το δεύτερο  $b$  από την 230 παρατήρηση του  $w$  και μέχρι την 241. Στο διάλυμα παρατηρήσεων  $a$ , η πρώτη παρατήρηση αντιστοιχεί στην τιμή κλεισίματος της μετοχής της IBM στις 01/06/2000, η δεύτερη παρατήρηση αντιστοιχεί στην τιμή κλεισίματος της μετοχής της IBM στις 01/07/2000, ενώ η παρατήρηση 230 αντιστοιχεί στην τιμή κλεισίματος της μετοχής της IBM στις 01/07/2019. Από την άλλη πλευρά, στο διάλυμα παρατηρήσεων  $b$  η παρατήρηση 230 αντιστοιχεί στην τιμή κλεισίματος της μετοχής της IBM στις 01/07/2019, η παρατήρηση 231 στην τιμή κλεισίματος της μετοχής της IBM στις 01/08/2019 και η παρατήρηση 241 στην τιμή κλεισίματος της μετοχής της IBM στις 01/06/2020. Η διαδικασία αυτή γίνεται με σκοπό το διάλυμα παρατηρήσεων  $b$  να χρησιμοποιηθεί ως μέτρο σύγκρισης μεταξύ των παρατηρήσεων του και των τιμών που θα προβλεφθούν από το νευρωνικό μας δίκτυο. Στη συνέχεια, με τη χρήση κατάλληλης συνάρτησης στην γλώσσα προγραμματισμού Python (βλ. ενότητα 4.4.1) [36] το διάλυμα παρατηρήσεων  $a$  οργανώνεται κατά τριάδες ως εξής: οι 2 πρώτες παρατηρήσεις αντιστοιχούν στις τιμές  $(t - 1, t)$  αντίστοιχα και η τρίτη παρατήρηση αντιστοιχεί στην τιμή στόχο  $t + 1$ . Έπειτα, η δεύτερη και η τρίτη παρατήρηση αντιστοιχούν στις τιμές  $(t - 1, t)$ , ενώ η τέταρτη παρατήρηση αντιστοιχεί στην τιμή στόχο  $t + 1$ . Η διαδικασία αυτή συνεχίζεται με τον ίδιο τρόπο για όλες τις παρατηρήσεις του διανύσματος παρατηρήσεων

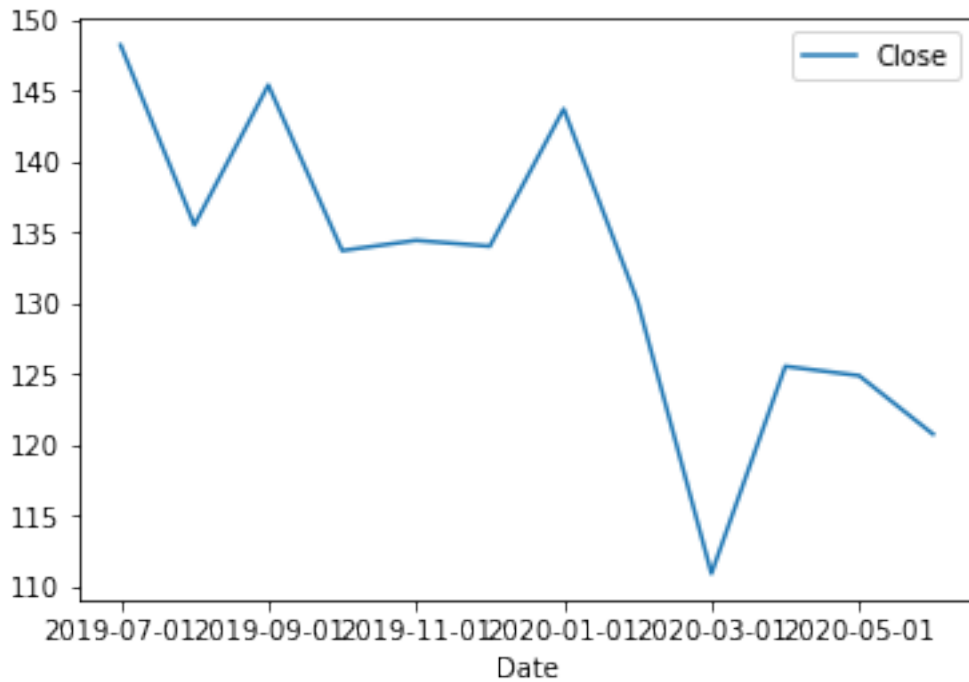
a. Με αυτό τον τρόπο κατασκευάζεται το σύνολο των pattern που θα χρησιμοποιηθούν για να επιλεγεί και να εκπαιδευθεί το νευρωνικό δίκτυο με το οποίο θα δημιουργηθούν οι μελλοντικές μας προβλεψεις. Ουσιαστικά, το σύνολο των pattern αποτελείται από δύο επιμέρους διανύσματα παρατηρήσεων  $\mathbf{X}$  και  $\mathbf{t}$ , όπου το  $\mathbf{X}$  αποτελείται από διανύσματα μεγέθους 2 και έχει μέγεθος 228 και το  $\mathbf{t}$  αποτελείται από 228 παρατηρήσεις, οι οποίες είναι οι τιμές στόχου. Η κάθε μία τιμή στόχου του  $\mathbf{t}$  αντιστοιχεί με την σειρά σε ένα διατεταγμένο ζεύγος του  $\mathbf{X}$ , δηλαδή το πρώτο διατεταγμένο ζεύγος του  $\mathbf{X}$  αντιστοιχεί στην πρώτη παρατήρηση του διανύσματος παρατηρήσεων  $\mathbf{t}$  και ούτω καθεξής. Στη συνέχεια, το σύνολο των pattern χωρίζεται με αναλογία 80:20 σε δύο νέα υποσύνολα τα pattern εκπαίδευσης και τα pattern ελέγχου. Πιο συγκεκριμένα, τα pattern εκπαίδευσης αποτελούνται από τα 182 πρώτα διατεταγμένα ζεύγη του διανύσματος παρατηρήσεων  $\mathbf{X}$  και από τις 182 πρώτες παρατηρήσεις του διανύσματος παρατηρήσεων  $\mathbf{t}$ . Από την άλλη πλευρά, τα pattern ελέγχου αποτελούνται από τα υπόλοιπα 46 διατεταγμένα ζεύγη του διανύσματος παρατηρήσεων  $\mathbf{X}$  και τις υπόλοιπες 46 παρατηρήσεις του διανύσματος παρατηρήσεων  $\mathbf{t}$ . Τελικά, με τον διαχωρισμό του συνόλου των pattern, δημιουργήθηκαν τα υποσύνολα  $\mathbf{X}_{\text{train}}$ ,  $\mathbf{X}_{\text{test}}$ ,  $\mathbf{t}_{\text{train}}$  και  $\mathbf{t}_{\text{test}}$ . Στο σημείο αυτό τα νέα υποσύνολα κανονικοποιούνται με σκοπό οι τιμές των παρατηρήσεων τους να βρίσκονται στο διάστημα (0,1).

Στο σχήμα 4.1 απεικονίζεται το διάγραμμα της χρονοσειράς των μετοχών κλεισίματος της IBM (a), ενώ στο σχήμα 4.2 απεικονίζεται το διάγραμμα της χρονοσειράς (b).



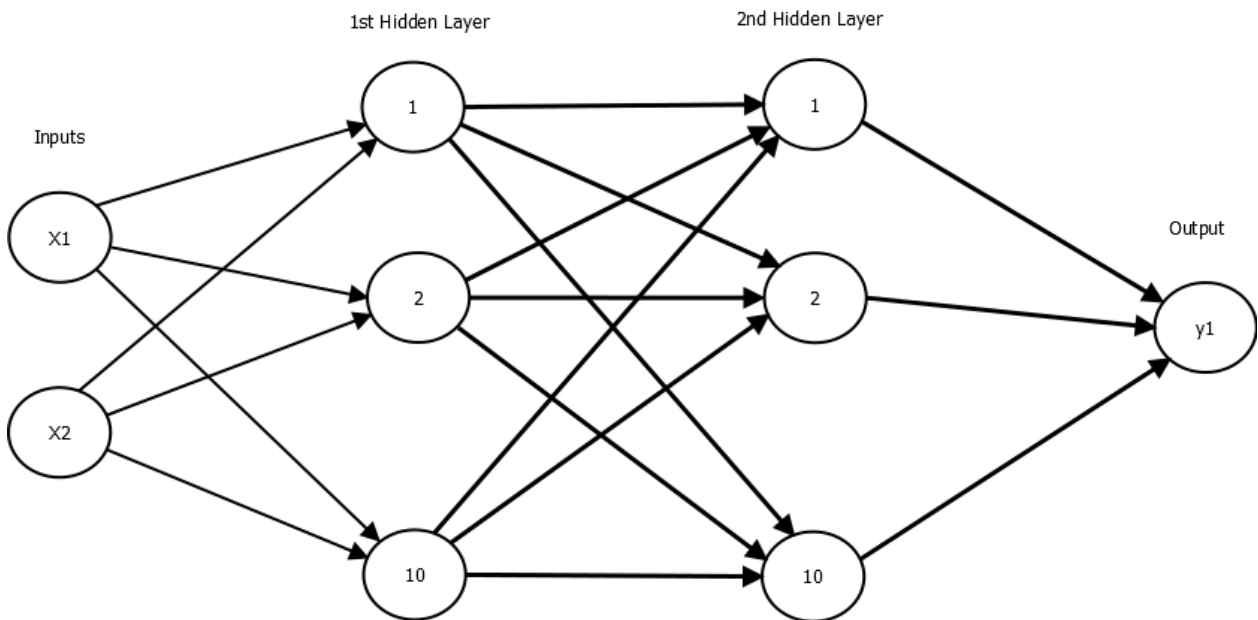
Σχήμα 4.1: Διάγραμμα Μετοχών Κλεισίματος IBM (a)

Στο δεύτερο στάδιο της εφαρμογής επιλέγεται και εκπαιδεύεται με την βοήθεια κατάλληλων εντολών στην python το καταλληλότερο πολυεπίπεδο δίκτυο Perceptron. Για την επιλογή του καταλληλότερου μοντέλου ελέγχθηκαν ορισμένοι παράμετροι των τεχνητών νευρωνικών δικτύων, όπως ο αριθμός των κρυφών επιπέδων, ο αριθμός των νευρώνων από τον οποίο αποτελείται το κάθε κρυφό επίπεδο, η συνάρτηση ενεργοποίησης των νευρώνων, ο solver της βελτιστοποίησης των βαρών για την υποστήριξη του αλγόριθμου μάθησης και η μεταβλητή alpha (παράμετρος ποινής για τον όρο κανονικοποίησης). Ειδικότερα, η επιλογή για τον αριθμό των κρυφών επιπέδων και του αριθμού των νευρώνων από τους οποίους θα αποτελείται κάθε κρυφό επίπεδο έγινε ανάμεσα στις εξής περιπτώσεις: ένα κρυφό επίπεδο με 10 νευρώνες, ένα κρυφό επίπεδο με 20 νευρώνες, δύο κρυφά επίπεδα με 10 νευρώνες το κάθε κρυφό επίπεδο και δύο κρυφά επίπεδα με 20 νευρώνες το πρώτο επίπεδο και 10 νευρώνες το δεύτερο επίπεδο. Οι συναρτήσεις ενεργοποίησης που εξετάστηκαν είναι η γραμμική, η λογιστική, η υπερβολική εφαπτομένη και συνάρτηση ράμπας, ενώ οι solver που εξετάστηκαν είναι ο stochastic gradient descent (sgd), ο lbfgs και ο adam. Τέλος, ελέγχθηκε η παράμετρος alpha για τις τιμές 0.00005, 0.0005. Τελικά, με κατάλληλες εντολές στην python τα pattern εκπαίδευσης διοχετεύονται 1-1 σε όλα τα νευρωνικά δίκτυα που δημιουργούνται από όλους τους πιθανούς συνδυασμούς των παραπάνω παραμέτρων. Τα νευρωνικά δίκτυα αυτά εκπαιδεύονται και τελικά επιλέγεται το νευρωνικό δίκτυο με το μικρότερο μέσο τετραγωνικό σφάλμα (mse).



Σχήμα 4.2: Διάγραμμα Μετοχών Κλεισίματος IBM ( $\beta$ )

Από όλα τα πιθανά πολυεπίεδα νευρωνικά δίκτυα Perceptron αυτό με το μικρότερο μέσο τετραγωνικό σφάλμα, δηλαδή το καταλληλότερο είναι αυτό που αποτελείται από δύο κρυφά επίπεδα από 10 νευρώνες το καθένα, που χρησιμοποιεί ως συνάρτηση ενεργοποίησης την συνάρτηση ράμπας (relu), ως solver τον sgd και τέλος παίρνει την τιμή  $5e - 05$  ως alpha. Η αρχιτεκτονική του καταλληλότερου νευρωνικού δικτύου απεικονίζεται στο σχήμα 4.3, ενώ κατασκευάστηκε με την εφαρμογή Dia.



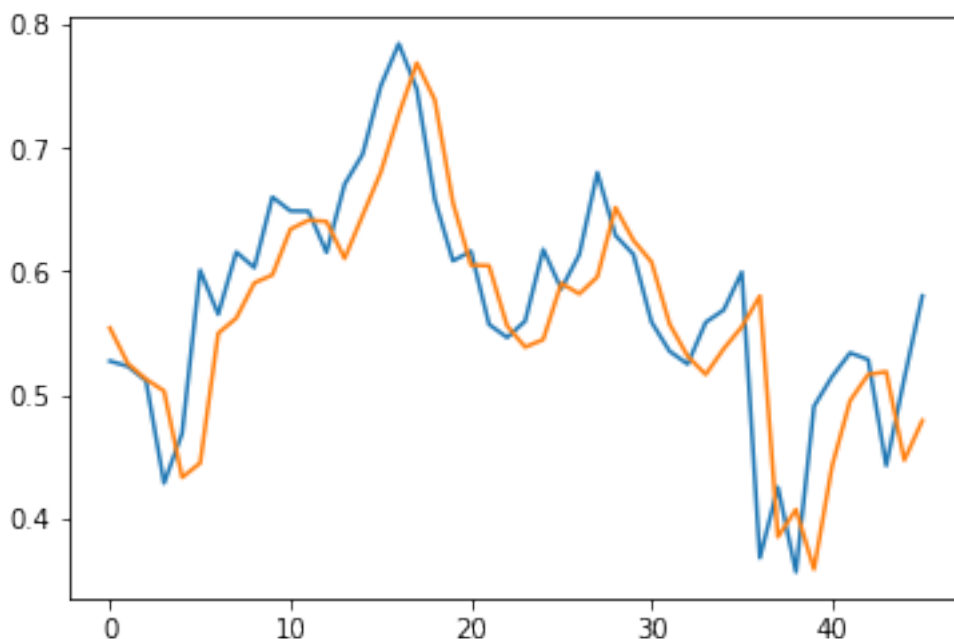
Σχήμα 4.3: Η Αρχιτεκτονική του Καταλληλότερου νευρωνικού δικτύου

Κατά την διαδικασία εκπαίδευσης του νευρωνικού δικτύου τα pattern εκπαίδευσης εισάγονται 1-1 στο νευρωνικό δίκτυο. Αρχικά, η κάθε μία παρατήρηση του πρώτου διατεταγμένου ζεύγους του  $\mathbf{X}_{train}$  πολλαπλασιάζεται με τα βάρη που έχουν τον ρόλο της σύναψης των νευρώνων του πρώτου κρυφού επιπέδου και το γινόμενο αυτό εισάγεται στους νευρώνες του πρώτου κρυφού επιπέδου. Οι 10 νευρώνες του πρώτου κρυφού επιπέδου υπολογίζουν μια τιμή εξόδου ο καθένας, η οποία πολλαπλασιάζεται με τα βάρη των νευρώνων του δεύτερου κρυφού επιπέδου.



Τελικά, οι 10 νευρώνες του δεύτερου κρυφού επιπέδου παράγουν μια έξοδο  $y$ . Στη συνέχεια, υπολογίζεται το μέσο τετραγωνικό σφάλμα της τιμής  $\psi$  και της αντίστοιχης τιμής στόχου του pattern εκπαίδευσης. Με την βοήθεια του  $\text{sgd}$  υπολογίζεται η πρώτη παράγωγος του μέσου τετραγωνικού σφάλματος που υπολογίστηκε προηγουμένως. Στην συνέχεια, με σκοπό η πρώτη παράγωγος του μέσου τετραγωνικού σφάλματος να παίρνει την τιμή μηδέν ή όσο γίνεται πιο κοντά στο μηδέν τροποποιούνται τα βάρη των νευρώνων των επιπέδων. Έπειτα, το δεύτερο διατεταγμένο ζεύγος του  $\mathbf{X}_{\text{train}}$  εισάγεται στο νευρωνικό δίκτυο και υλοποιούνται τα ίδια βήματα μέχρι και την τροποποίηση των βαρών των νευρώνων. Όταν εισαχθούν όλα τα pattern εκπαίδευσης στο νευρωνικό δίκτυο λέμε ότι έχει εκτελεστεί μία εποχή. Το νευρωνικό δίκτυο εκπαιδεύεται μέχρι τα βάρη των νευρώνων του να μην αλλάζουν τιμή.

Στο σημείο αυτό πραγματοποιείται έλεγχος του καταλληλότερου νευρωνικού δικτύου. Για τον έλεγχο του νευρωνικού δικτύου χρησιμοποιούνται ως είσοδοι σε αυτό τα διατεταγμένα ζεύγη ελέγχου  $\mathbf{X}_{\text{test}}$ . Τα ζεύγη αυτά εισάγονται 1-1 όπως και στην διαδικασία εκπαίδευσης και κάθε ένα από αυτά παράγει μία έξοδο, η οποία αντιστοιχεί στην τιμή  $t + 1$ . Έτσι, καθώς τα διατεταγμένα ζεύγη ελέγχου είναι 46 το νευρωνικό δίκτυο παράγει 46 εξόδους. Έπειτα, υπολογίζεται το μέσο τετραγωνικό σφάλμα (mse) μεταξύ των 46 εξόδων που παράχθηκαν με βάση το  $\mathbf{X}_{\text{test}}$  και τις τιμές στόχου του  $t_{\text{test}}$ . Τελικώς, η τιμή του μέσου τετραγωνικού σφάλματος μεταξύ των δύο συνόλων είναι 0.003937534685401186. Στο σχήμα 4.4 απεικονίζονται τα διαγράμματα των  $t_{\text{test}}$  (μπλε γραμμή) και των τιμών ελέγχου που παράγει το καταλληλότερο νευρωνικό δίκτυο (πορτοκαλί γραμμή). Από το διάγραμμα 4.4 παρατηρούμε ότι οι τιμές ελέγχου ακολουθούν παρομοίω μοτίβο με τις τιμές στόχου μετατοπισμένο δεξιάτερα.



Σχήμα 4.4: Τα Διαγράμματα των τιμών ελέγχου που παράγει το καταλληλότερο νευρωνικό και του  $t_{\text{test}}$

Έχοντας ελέγξει την αποτελεσματικότητα του καταλληλότερου νευρωνικού δικτύου προβλέπουμε τις μελλοντικές τιμές των μετοχών κλεισίματος της IBM από τις 01/07/19 και για κάθε μήνα μέχρι τις 01/10/20. Για να υλοποιήσουμε αυτές τις προβλέψεις αρχικά δημιουργούμε το διατεταγμένο ζεύγος με τις δύο τελευταίες κανονικοποιημένες τιμές του  $t_{\text{test}}$ , δηλαδή τις τιμές κλεισίματος στις 01/05/19 και στις 01/06/19. Στη συνέχεια, εισάγουμε αυτό το διατεταγμένο ζεύγος στο εκπαιδευμένο νευρωνικό δίκτυο και αυτό προβλέπει μια τιμή, η οποία αντιστοιχεί στην τιμή κλεισίματος στις 01/07/19. Για να προβλέψουμε την τιμή κλεισίματος στις 01/08/19 δημιουργούμε το διατεταγμένο ζεύγος με τις τιμές κλεισίματος στις 01/06/19 και στις 01/07/19, που ουσιαστικά προβλέψαμε στο προηγούμενο βήμα. Για να φτάσουμε τις μελλοντικές προβλέψεις μέχρι τις 01/10/20, δημιουργούμε με τον ίδιο τρόπο τα εκάστοτε διατεταγμένα ζεύγη. Καθώς οι μελλοντικές τιμές που προβλέπει το νευρωνικό δίκτυο είναι κανονικοποιημένες χρησιμοποιούμε κατάλληλη εντολή της python, η οποία επιστρέφει τις κανονικοποιημένες προβλέψεις στην αρχική κλίμακα των παρατηρήσεων. Στο παρακάτω πίνακα διακρίνονται οι προβλέψεις που παράγει το νευρωνικό δίκτυο και οι πραγματικές τιμές της μετοχής κλεισίματος της IBM.

Ημερομηνία	Μελλοντικές προβλέψεις	Πραγματικές τιμές
Ιουν-19	137,8999941	148,240005
Ιουλ-19	132,6284478	135,529999
Αυγ-19	132,2863709	145,419998
Σεπ-19	130,3910664	133,729996
Οκτ-19	128,932175	134,449997
Νοε-19	127,2979774	134,039993
Δεκ-19	126,0337348	143,729996
Ιαν-20	124,9692836	130,149994
Φεβ-20	124,0963242	110,93
Μαρ-20	123,3746489	125,559998
Απρ-20	122,2883332	124,9
Μαϊ-20	121,8637831	120,769997
Ιουν-20	12142504356	124,900002
Ιουλ-20	120,9793047	120,769997
Αυγ-20	120,5257683	122,940002
Σεπ-20	120,0643615	123,309998
Οκτ-20	119,5949418	?

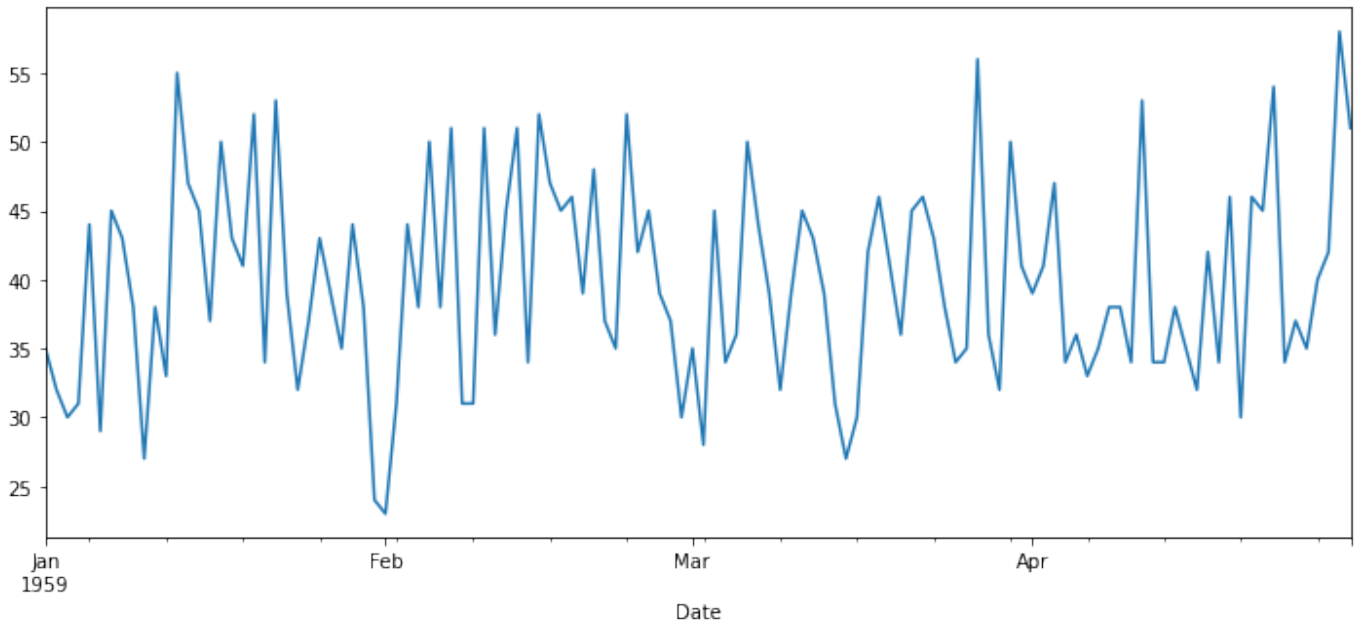
### 4.3 Στατιστικά Μοντέλα

Για την επιλογή, την δημιουργία και την εκπαίδευση των καταλληλότερων στατιστικών μοντέλων χρησιμοποιήθηκαν τρία σύνολα δεδομένων, ένα για το κάθε μοντέλο. Τα σύνολα δεδομένων αυτά είναι από το διαδικτυακό μάθημα του Jose Portilla με τίτλο “Python for Time Series Data Analysis” της ηλεκτρονικής πλατφόρμας Udemy [22]. Τα μοντέλα που χρησιμοποιήθηκαν για την δημιουργία μελλοντικών προβλέψεων με βάση τα τρία σύνολα δεδομένων που μελετήθηκαν είναι τα ARMA, ARIMA και SARIMA (βλ. Κεφάλαιο 3).

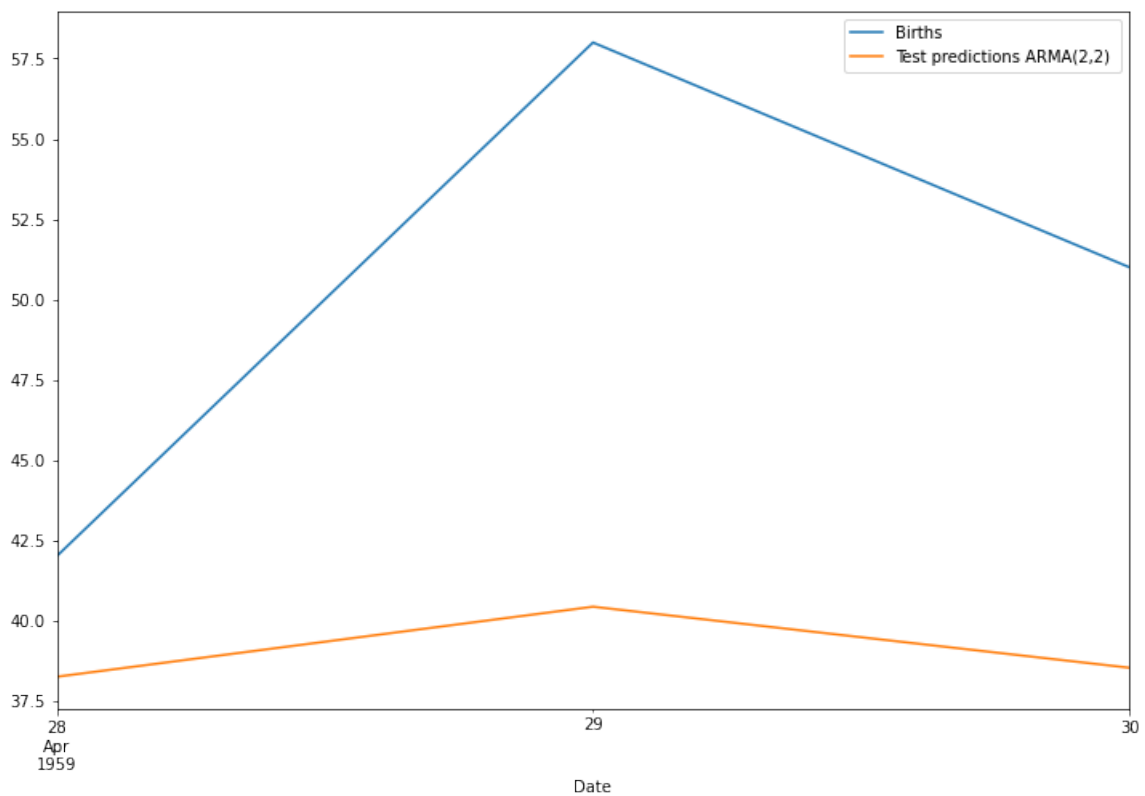
#### 4.3.1 ARMA

Για το μοντέλο ARMA χρησιμοποιήθηκε ένα σύνολο δεδομένων, το οποίο περιέχει τον ημερήσιο αριθμό γεννήσεων νοεγνών θηλυκού γένους από τη 01/01/1959 έως 30/04/1959. Ειδικότερα, το σύνολο δεδομένων αυτό αποτελείται από 120 παρατηρήσεις και δύο στήλες: η πρώτη στήλη περιλαμβάνει τις ημερομηνίες καταγραφής των γεννήσεων, ενώ η δεύτερη στήλη περιλαμβάνει τον αριθμό των ημερήσιων γεννήσεων.

Στο πρώτο στάδιο της ανάλυσης της παραπάνω χρονοσειράς, δημιουργήθηκε το διάγραμμα της, με σκοπό να διακριθούν τυχόν ιδιαίτερα χαρακτηριστικά της όπως η στασιμότητα, η τάση και η εποχικότητα. Στο σχήμα 4.4 απεικονίζεται το διάγραμμα της χρονοσειράς των γεννήσεων. Με βάση το διάγραμμα της, η χρονοσειρά αυτή φαίνεται να έχει σταθερή μέση τιμή και διακύμανση (στάσιμη), ενώ παρατηρείται ότι μάλλον δεν έχει τάση και εποχικότητα. Στο σημείο αυτό πραγματοποιήθηκε ο έλεγχος Dickey-Fuller με σκοπό να επιβεβαιωθεί η στασιμότητα της χρονοσειράς. Εκτελώντας τον έλεγχο στασιμότητας Dickey-Fuller η τιμή p-value είναι **4.3735454059910584e-17** < 0.05, γεγονός που σημαίνει ότι απορρίπτουμε την μηδενική υπόθεση και άρα η χρονοσειρά μας είναι πράγματι στάσιμη. Με την επιβεβαίωση της στασιμότητας της χρονοσειράς και με την γνώση μας για τα στατιστικά μοντέλα (βλ. κεφάλαιο 3) κατευθυνόμαστε στην χρήση του μοντέλου ARMA. Έτσι, με την εντολή της Python `auto_arima(df1['Births'], seasonal = False).summary()` αναζητείται το καταλληλότερο μοντέλο ARMA με βάση τα δεδομένα όλης της χρονοσειράς. Από την παραπάνω εντολή προκύπτει ότι το καταλληλότερο μοντέλο, με βάση τον εκτιμητή Kalman Filter είναι το ARMA(2,2), δηλαδή το μοντέλο με υστέρηση πρώτης και δεύτερης τάξης αυτοπαλινδρόμου όρου και κινητού μέσου. Έπειτα, διαχωρίζεται το σύνολο δεδομένων σε δεδομένα εκπαίδευσης (train test) και δεδομένα ελέγχου (test set). Πιο συγκεκριμένα, το πλήθος των δεδομένων εκπαίδευσης είναι 117, ενώ των δεδομένων ελέγχου είναι 3. Στο σημείο αυτό, το μοντέλο ARMA(2,2) δομείται και εκπαιδεύεται με βάση τα δεδομένα εκπαίδευσης. Στην συνέχεια, με βάση το μοντέλο που εκπαιδεύτηκε προηγουμένως, δημιουργούνται οι προβλέψεις για τις αντίστοιχες τιμές του συνόλου ελέγχου. Οι προβλέψεις αυτές συγκρίνονται σχηματικά με τα αληθινά δεδομένα ελέγχου στο διάγραμμα 4.6. Από το διάγραμμα 4.6 παρατηρείται ότι οι προβλέψεις (πορτοκαλί γραμμή) του εκπαιδευμένου με τα δεδομένα εκπαίδευσης μοντέλου δεν αποτυπώνουν με ακρίβεια τις πραγματικές τιμές του συνόλου ελέγχου. Επιπρόσθετα, παρατηρείται ότι το Root Mean Square Error (rmse) δεν είναι ικανοποιητικά μικρότερο από τις μέσες τιμές των προβλέψεων και του συνόλου ελέγχου, οι οποίες δεν παρουσιάζουν κοντινές τιμές. Πιο συγκεκριμένα, `rmse=12.625851824366249`, `prediction mean=39.07013493532798`, `test set mean=50.333333333333336`. Έτσι, το μοντέλο θεωρείται ότι μάλλον δεν προβλέπει ικανοποιητικά.

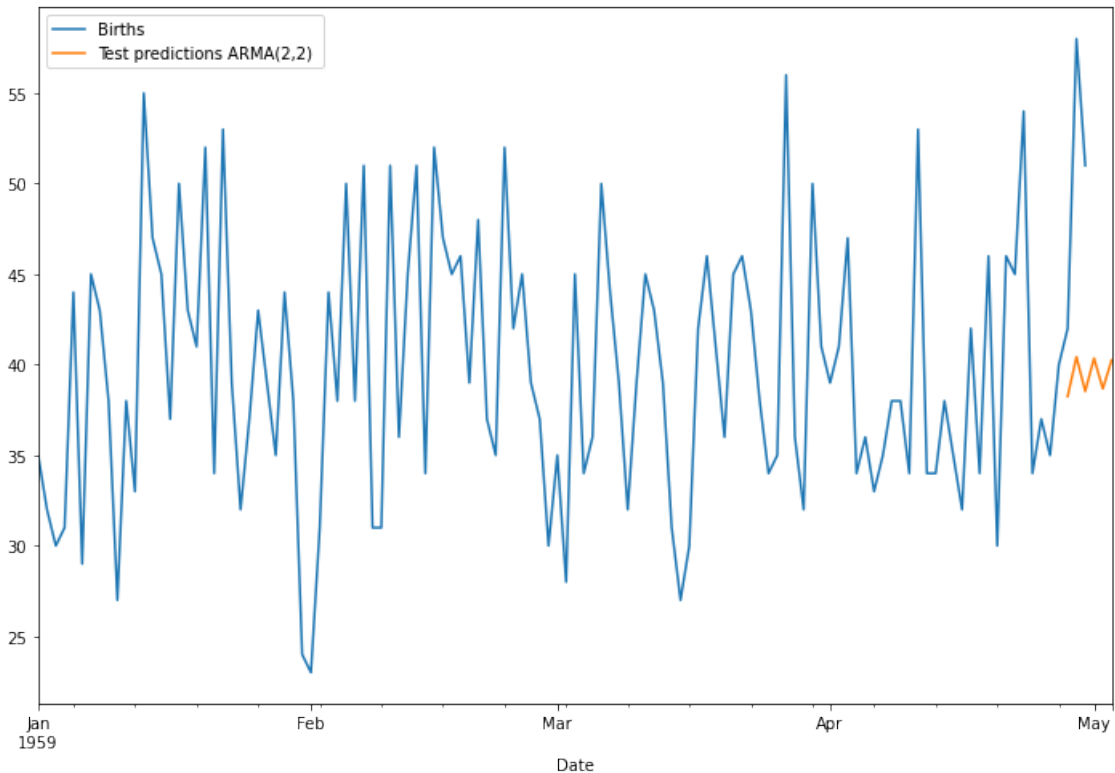


Σχήμα 4.5: Διάγραμμα Χρονοσειράς Γεννήσεων



Σχήμα 4.6: Διάγραμμα Σύγκρισης Μεταξύ των Δεδομένων Ελέγχου και των Αντίστοιχων Προβλεψεων

Στο σημείο αυτό πραγματοποιούνται μελλοντικές προβλέψεις για την χρονοσειρά μας. Ειδικότερα, πραγματοποιούνται οι μελλοντικές προβλέψεις για τις επόμενες τρεις ημέρες εκτός του συνόλου δεδομένων της χρονοσειράς. Η χρονοσειρά και οι μελλοντικές προβλέψεις διακρίνονται στο διάγραμμα 4.7.



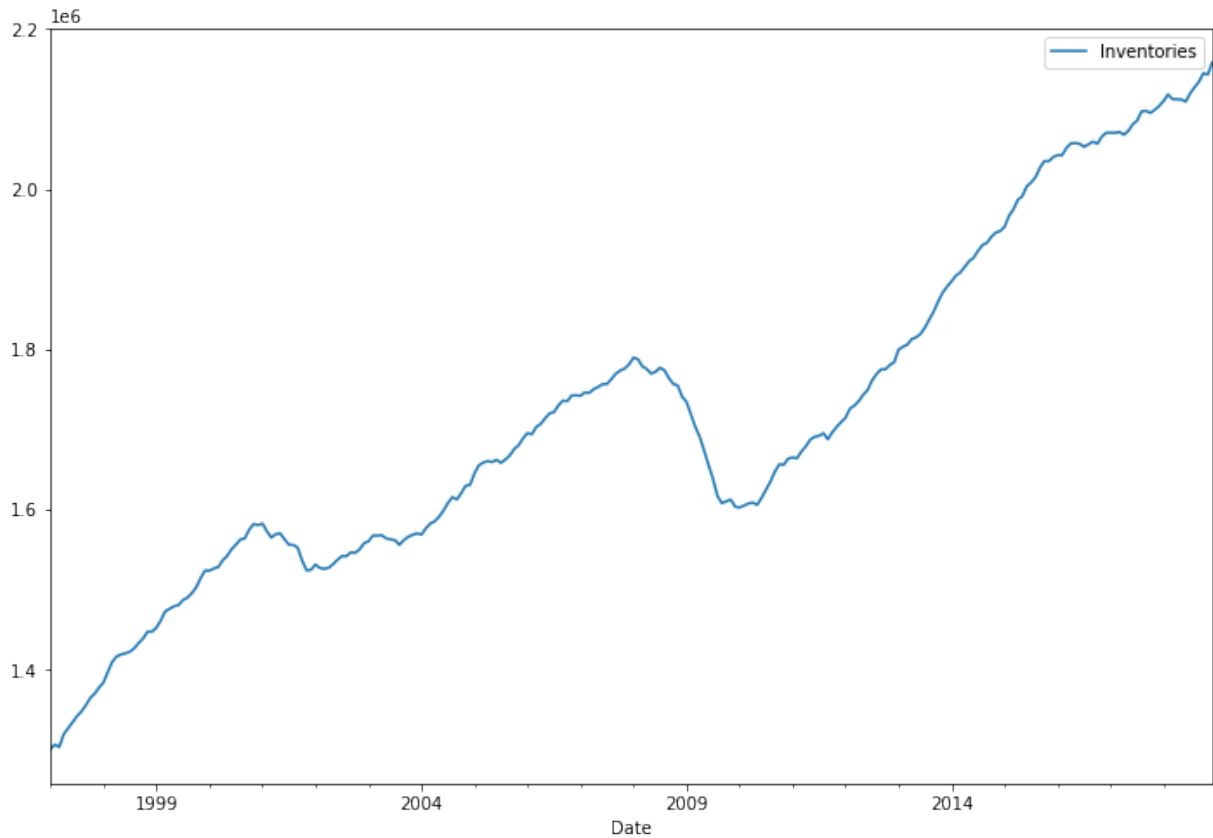
Σχήμα 4.7: Διάγραμμα Μελλοντικών Προβλέψεων ARMA(2,2) Μοντέλου

### 4.3.2 ARIMA

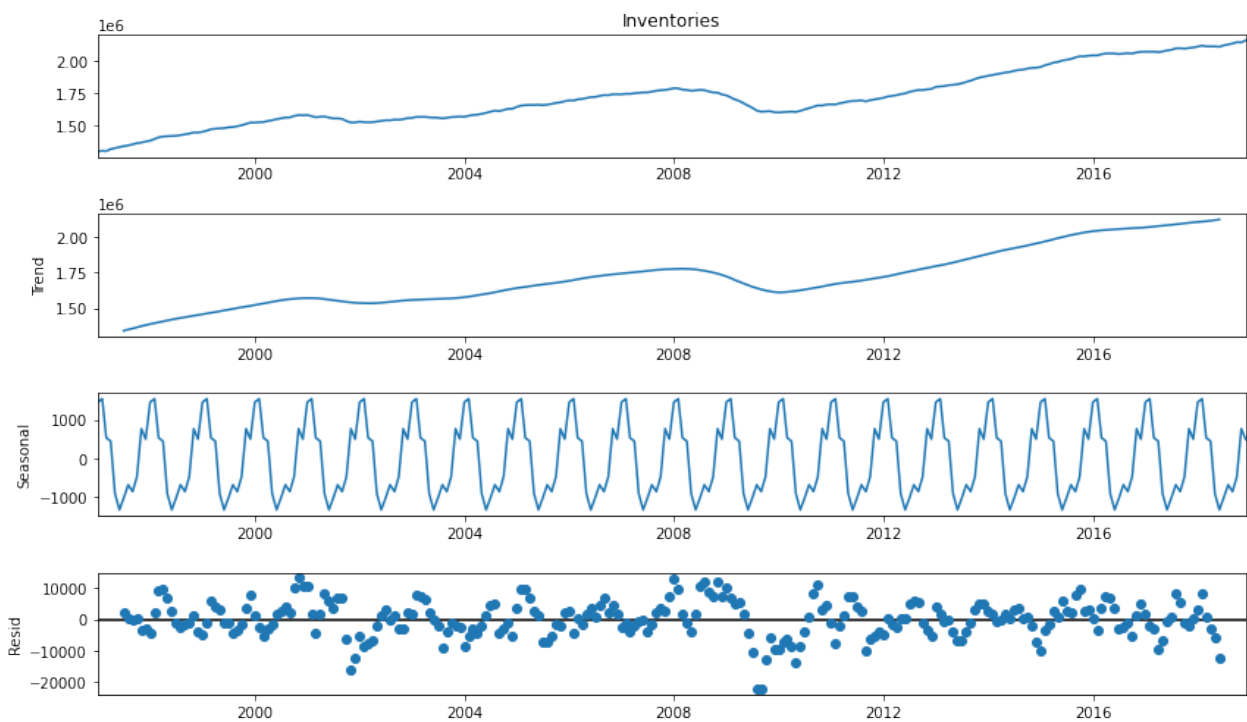
Για το μοντέλο ARIMA χρησιμοποιήθηκε ένα σύνολο δεδομένων, το οποίο περιέχει τον αριθμό εμπορικών αποθεμάτων για τις πρώτες 12 ημέρες κάθε Ιανουαρίου από τη 01/01/1997 έως 12/01/2018. Ειδικότερα, το σύνολο δεδομένων αυτό αποτελείται από 264 παρατηρήσεις και δύο στήλες: η πρώτη στήλη περιλαμβάνει τις ημερομηνίες καταγραφής των αποθεμάτων, ενώ η δεύτερη στήλη περιλαμβάνει τον αριθμό των αποθεμάτων.

Στο πρώτο στάδιο της ανάλυσης της παραπάνω χρονοσειράς, δημιουργήθηκε το διάγραμμα της, με σκοπό να διακριθούν τυχόν ιδιαίτερα χαρακτηριστικά της όπως η στασιμότητα, η τάση και η εποχικότητα. Στο σχήμα 4.8 απεικονίζεται το διάγραμμα της χρονοσειράς των εμπορικών αποθεμάτων. Με βάση το διάγραμμα της, η χρονοσειρά αυτή φαίνεται να μην έχει σταθερή μέση τιμή και διακύμανση (μη-στάσιμη), ενώ παρατηρείται ότι πιθανόν να έχει τάση, αλλά όχι εποχικότητα. Στο σημείο αυτό πραγματοποιήθηκε ο έλεγχος Dickey-Fuller με σκοπό να επιβεβαιωθεί η μη-στασιμότητα της χρονοσειράς. Εκτελώντας τον έλεγχο στασιμότητας Dickey-Fuller η τιμή p-value είναι **0.9506515302298977** > 0.05, γεγονός που σημαίνει ότι αποδεχόμαστε την μηδενική υπόθεση και άρα η χρονοσειρά είναι πράγματι μη-στάσιμη. Με την επιβεβαίωση της μη-στασιμότητας της χρονοσειράς και με την γνώση μας για τα στατιστικά μοντέλα (βλ. κεφάλαιο 3) κατευθυνόμαστε στην χρήση του μοντέλου ARIMA, καθώς από το διάγραμμα της χρονοσειράς παρατηρείται η έλλειψη εποχικότητας και αποκλείεται το εποχικό μοντέλο SARI-MA. Πριν αναζητηθεί το καταλληλότερο μοντέλο για την περιγραφή της χρονοσειράς, δημιουργείται το διάγραμμα Decompose (διάγραμμα 4.9) με σκοπό να εξετάσουμε τις συνιστώσες της χρονοσειράς. Το διάγραμμα Decompose αποτελείται από τέσσερα μικρότερα διαγράμματα. Στο πρώτο διάγραμμα απεικονίζεται η χρονοσειρά, στο δεύτερο διάγραμμα απεικονίζεται η πιθανή τάση της υπό μελέτη χρονοσειράς, στο τρίτο διάγραμμα απεικονίζεται η πιθανή εποχικότητα της χρονοσειράς και στο τέταρτο διάγραμμα απεικονίζονται τα κατάλοιπα της χρονοσειράς. Από το διάγραμμα Decompose της χρονοσειράς διακρίνουμε την ύπαρξη αυξητικής τάσης στα αποθέματα με την παροδο του χρόνου. Επίσης, διακρίνεται ότι η χρονοσειρά παρουσιάζει μια εποχικότητα, η οποία λόγω του μικρού εύρους των τιμών της (άξονας y) δεν λαμβάνεται υπόψη. Τέλος, στο διάγραμμα των καταλοίπων δεν παρατηρείται κάποιο μοτίβο.

Στην συνέχεια της ανάλυσης της χρονοσειράς με την εντολή της Python `auto_arima(df2['Inventories'], start_p = 0, start_q = 0, max_p = 4, max_q = 3, seasonal = False, trace = True).summary()` αναζητείται το καταλληλότερο μοντέλο ARIMA με βάση τα δεδομένα όλης της χρονοσειράς και περιορίζοντας τις παρεμέτρους p και q στα σύνολα (0,4) και (0,3) αντίστοιχα. Από την παραπάνω εντολή προκύπτει ότι το καταλληλότερο μοντέλο, με βάση τον εκτιμητή Kalman Filter είναι το ARIMA(1,1,1), δηλαδή το μοντέλο με υστέρηση πρώτης τάξης αυτοπαλίνδρουμου

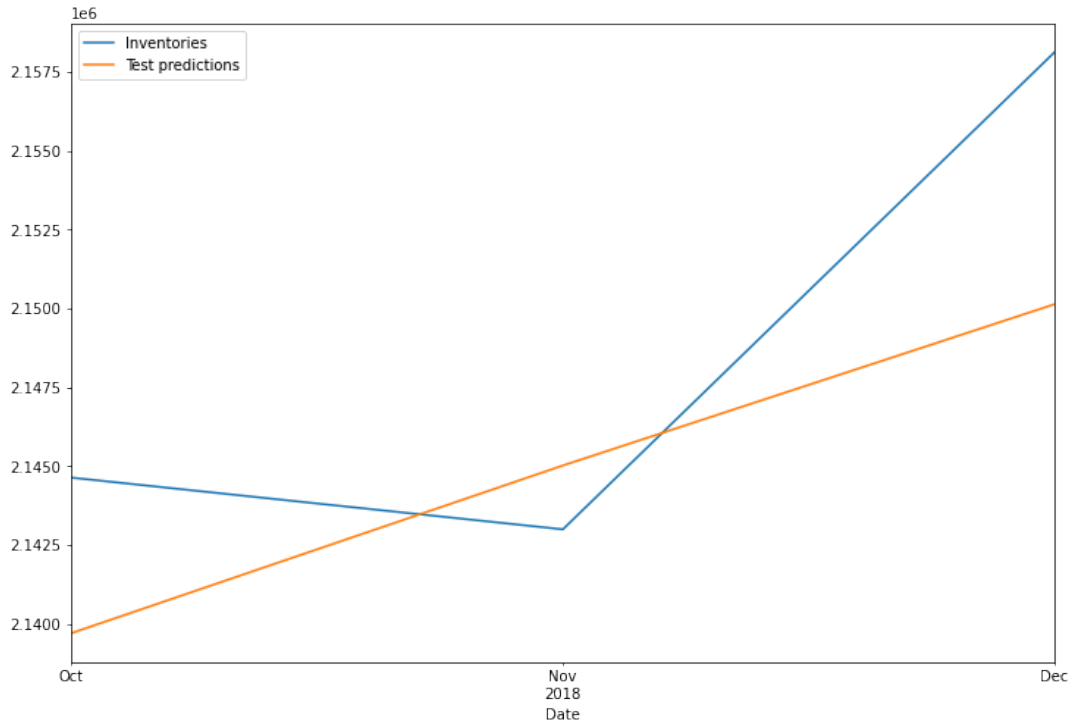


Σχήμα 4.8: Διάγραμμα Χρονοσειράς Εμπορικών Αποθέματων

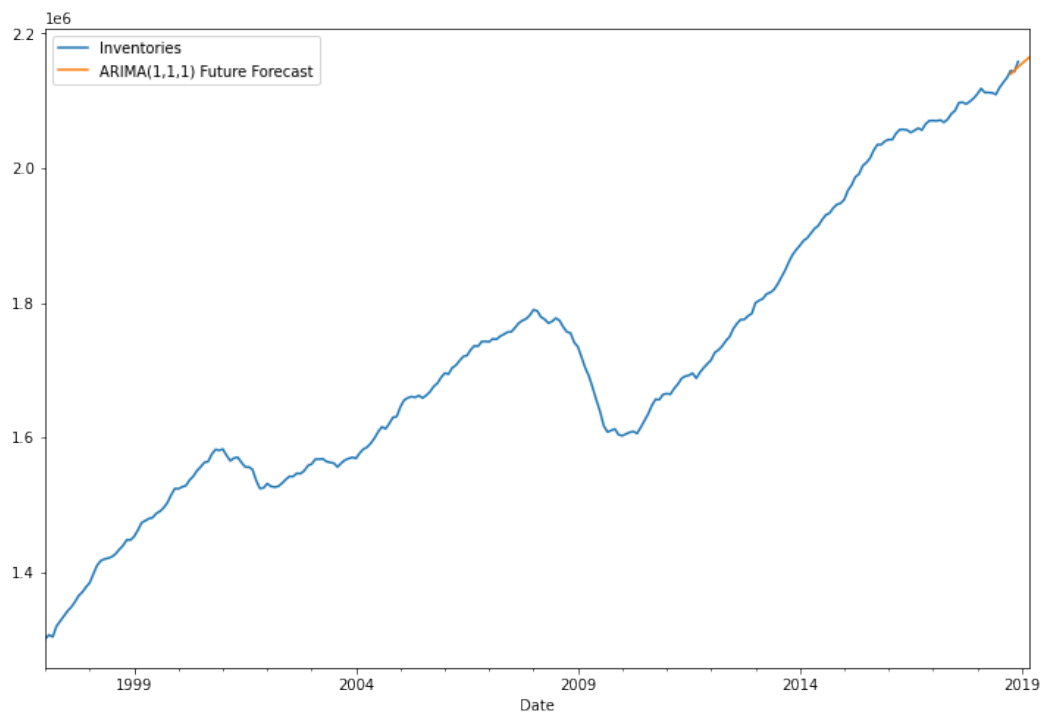


Σχήμα 4.9: Διάγραμμα Decompose

όρου και κινητού μέσου και που χρειάζεται να εφαρμοστεί μία φορά η μέθοδος των διαφορών στην χρονοσειρά ώστε αυτή να γίνει στάσιμη. Έπειτα, διαχωρίζεται το σύνολο δεδομένων σε σύνολο εκπαίδευσης (train test) και σύνολο ελέγχου (test set), όπου το πλήθος των δεδομένων εκπαίδευσης είναι 261, ενώ των δεδομένων ελέγχου είναι 3.



Σχήμα 4.10: Διάγραμμα Σύγκρισης Μεταξύ των Δεδομένων Ελέγχου και των Αντίστοιχων Προβλεψεων



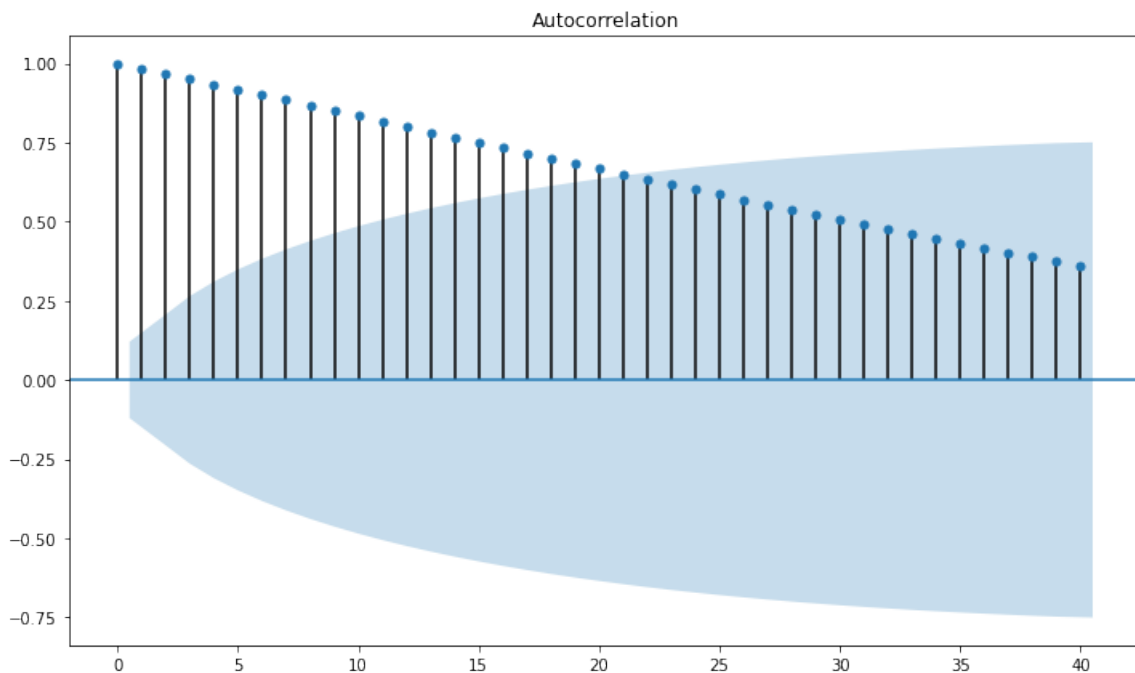
Σχήμα 4.11: Διάγραμμα Μελλοντικών Προβλέψεων ARIMA(1,1,1) Μοντέλου

Το μοντέλο ARIMA(1,1,1) δομείται και εκπαιδεύεται με βάση τα δεδομένα εκπαίδευσης. Στην συνέχεια, με βάση το μοντέλο που εκπαιδεύτηκε προηγουμένως, δημιουργούνται οι προβλέψεις για τις αντίστοιχες τιμές του συνόλου ελέγχου. Οι προβλέψεις αυτές συγκρίνονται σχηματικά με τα αληθινά δεδομένα ελέγχου στο διάγραμμα 4.10. Από το διάγραμμα 4.10 παρατηρείται ότι οι προβλέψεις (πορτοκαλί γραμμή) του εκπαιδευμένου με τα δεδομένα εκπαίδευσης μοντέλου αποτυπώνουν ικανοποιητικά τις πραγματικές τιμές του συνόλου ελέγχου. Η παρατήρηση αυτή επιβεβαιώνεται υπολογίζοντας το Root Mean Square Error (rmse) και συγκρίνοντας το με τις μέσες τιμές των προβλέψεων

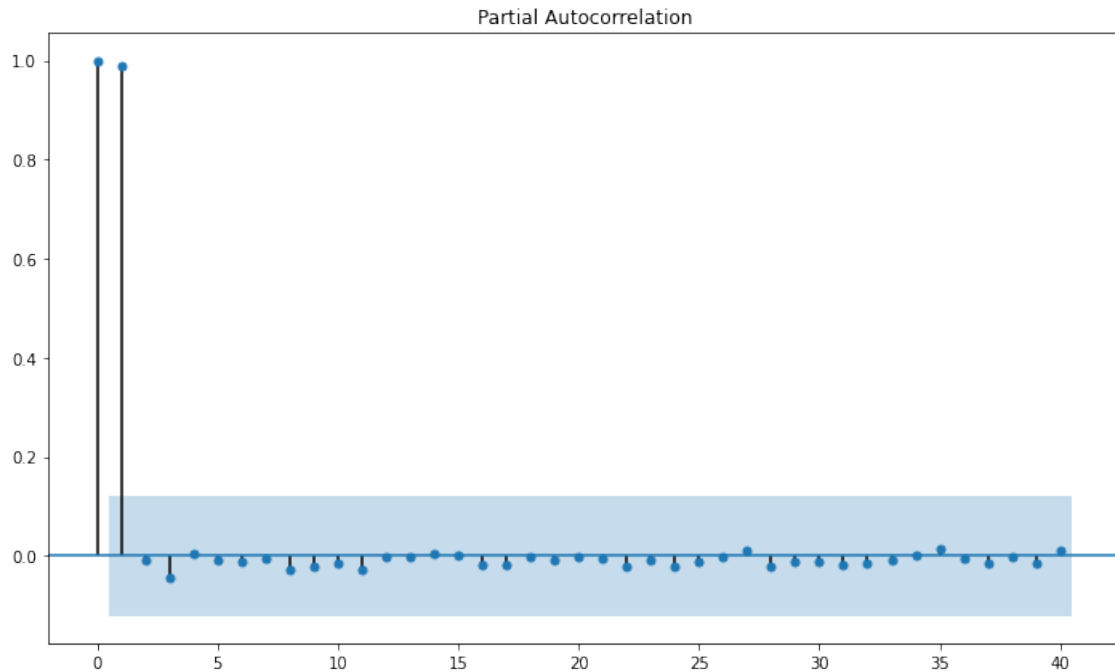
και του συνόλου ελέγχου. Με την σύγκριση αυτή παρατηρείται ότι η τιμή του rmse είναι κατά πολύ μικρότερη και από τις δύο μέσες τιμές. Ειδικότερα,  $rmse=5540.510231261458$  ,  $prediction\ mean=2144956.0655194274$ ,  $test\ set\ mean=2148585.0$ . Έτσι, το μοντέλο θεωρείται ότι προβλέπει ικανοποιητικά.

Στο σημείο αυτό πραγματοποιούνται μελλοντικές προβλέψεις για την χρονοσειρά μας. Ειδικότερα, πραγματοποιούνται οι μελλοντικές προβλέψεις για τις επόμενες τρεις ημέρες του Ιανουαρίου του 2019 εκτός του συνόλου δεδομένων της χρονοσειράς. Η χρονοσειρά και οι μελλοντικές προβλέψεις διακρίνονται στο διάγραμμα 4.11.

Με την ανάλυση της χρονοσειράς και ειδικότερα με την χρήση του ελέγχου στασιμότητας Dickey-Fuller παρατηρήθηκε ότι η χρονοσειρά αυτή είναι μη-στάσιμη. Παρόλα αυτά όπως αναφέρθηκε στο κεφάλαιο 3 για να μπορεί το μοντέλο ARIMA να αναπαριστήσει κατάλληλα μια χρονοσειρά πρέπει αυτή να είναι ήδη στάσιμη ή να μετατραπεί σε στάσιμη με την βοήθεια διάφορων τεχνικών. Στην παραπάνω ανάλυση η συνάρτηση *auto\_arima* αναφέρει ότι για να μετατραπεί η υπο μελέτη χρονοσειρά σε στάσιμη χρειάζεται να εφαρμοσθεί η μέθοδος των διαφορών μία φορά ( $d = 1$ ). Στην σημερινή εποχή η χρήση των γλωσσών προγραμματισμού προσφέρει στους αναλυτές εύκολα και γρήγορα αποτελέσματα όσον αφορά τις παραμέτρους των εκάστοτε μοντέλων, όμως εκτός από την εύκολη λύση των γλωσσών η αναζήτηση των κατάλληλων παραμέτρων μπορεί να γίνει και εμπειρικά. Έτσι, εκτελέστηκε μία φορά η μέθοδος των διαφορών στα δεδομένα της χρονοσειράς και στην συνέχεια εφαρμόστηκε ο έλεγχος στασιμότητας Dickey-Fuller στα δεδομένα της νέας αυτής χρονοσειράς. Από τον έλεγχο προέκυψε ότι η τιμή p-value είναι ίση με **0.010548311489190486** < 0.05, γεγονός που σημαίνει ότι η χρονοσειρά που δημιουργήθηκε μετά την εφαρμογή της μεθόδου των διαφορών μετατράπηκε σε στάσιμη. Επιπρόσθετα, από την συνάρτηση *auto\_arima* προκύπτει ότι το μοντέλο ARIMA παρουσιάζει AR( $p=1$ ) και MA( $q=1$ ). Από το διάγραμμα 4.13 μερικής αυτοσυσχέτισης (PACF) και από γνωστό εμπειρικό κανόνα προκύπτει ότι μετά την πρώτη υστέρηση (lag) του διαγράμματος υπάρχει απότομη κλίση προς την τιμή μηδέν. Έτσι, εμπειρικά αναφέρεται ότι πιθανόν το μοντέλο που αναπαριστά καταλληλότερα την χρονοσειρά πρέπει να περιέχει αυτοπαλίνδρομο όρο πρώτης τάξης, όπως ακριβώς προέκυψε και με την βοήθεια της python. Στο σχήμα 4.12 απεικονίζεται το διάγραμμα αυτοσυσχέτισης (ACF).



Σχήμα 4.12: Διάγραμμα Αυτοσυσχέτισης ACF



Σχήμα 4.13: Διάγραμμα Μερικής Αυτοσυσχέτισης PACF

### 4.3.3 SARIMA

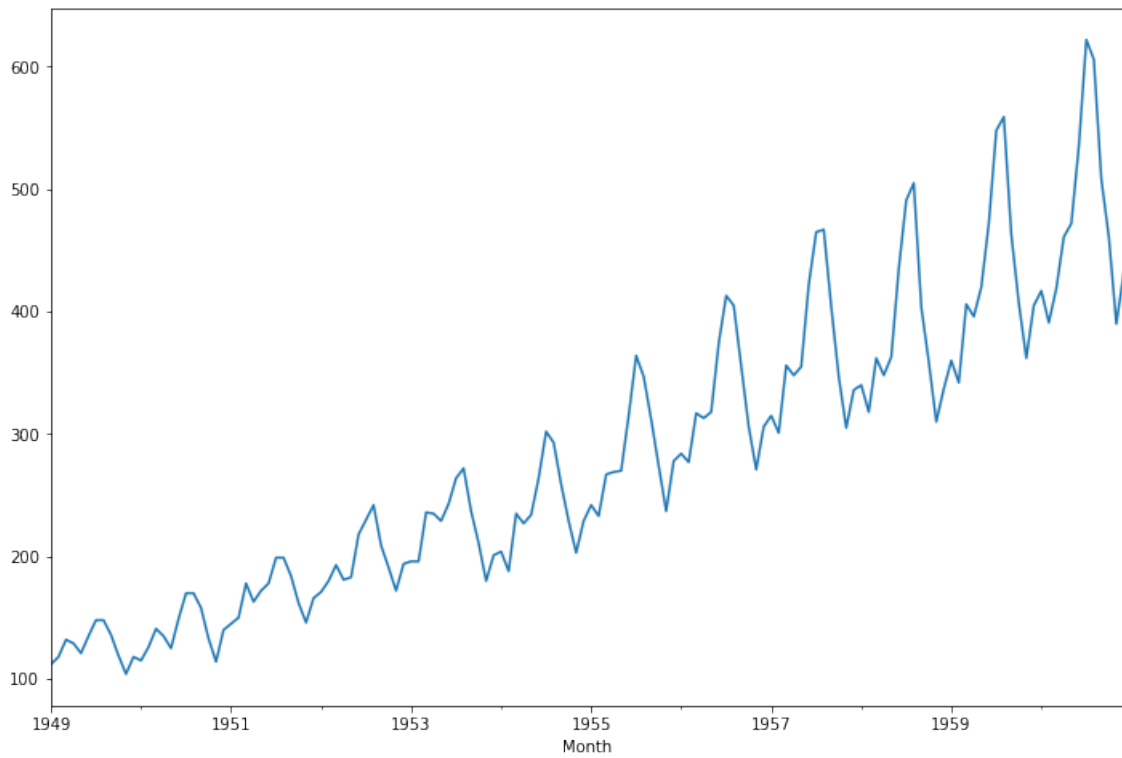
Για το μοντέλο SARIMA χρησιμοποιήθηκε το γνωστό σύνολο δεδομένων με τον μηνιαίο αριθμό επιβατών σε χιλιάδες των αμερικάνικων αερογραμμών από τη 01/1949 έως 12/1960. Ειδικότερα, το σύνολο δεδομένων αυτό αποτελείται από 144 παρατηρήσεις και δύο στήλες: η πρώτη στήλη περιλαμβάνει τον μήνα και το έτος καταγραφής των επιβατών, ενώ η δεύτερη στήλη περιλαμβάνει τον μηνιαίο αριθμό των επιβατών σε χιλιάδες των αμερικάνικων αερογραμμών.

Στο πρώτο στάδιο της ανάλυσης της παραπάνω χρονοσειράς, δημιουργήθηκε το διάγραμμα της, με σκοπό να διακριθούν τυχόν ιδιαίτερα χαρακτηριστικά της όπως η στασιμότητα, η τάση και η εποχικότητα. Στο σχήμα 4.14 απεικονίζεται το διάγραμμα της χρονοσειράς του αριθμού των επιβατών των αμερικάνικων αερογραμμών. Με βάση το διάγραμμα της, η χρονοσειρά φαίνεται να μην έχει σταθερή μέση τιμή και διακύμανση (μη-στάσιμη), ενώ παρατηρείται ότι πιθανόν να έχει τάση και εποχικότητα. Στο σημείο αυτό πραγματοποιήθηκε ο έλεγχος Dickey-Fuller με σκοπό να επιβεβαιωθεί η μη-στασιμότητα της χρονοσειράς. Εκτελώντας τον έλεγχο στασιμότητας Dickey-Fuller η τιμή p-value είναι  $0.991880243437641 > 0.05$ , γεγονός που σημαίνει ότι αποδεχόμαστε την μηδενική υπόθεση και άρα ότι η χρονοσειρά είναι πράγματι μη-στάσιμη.

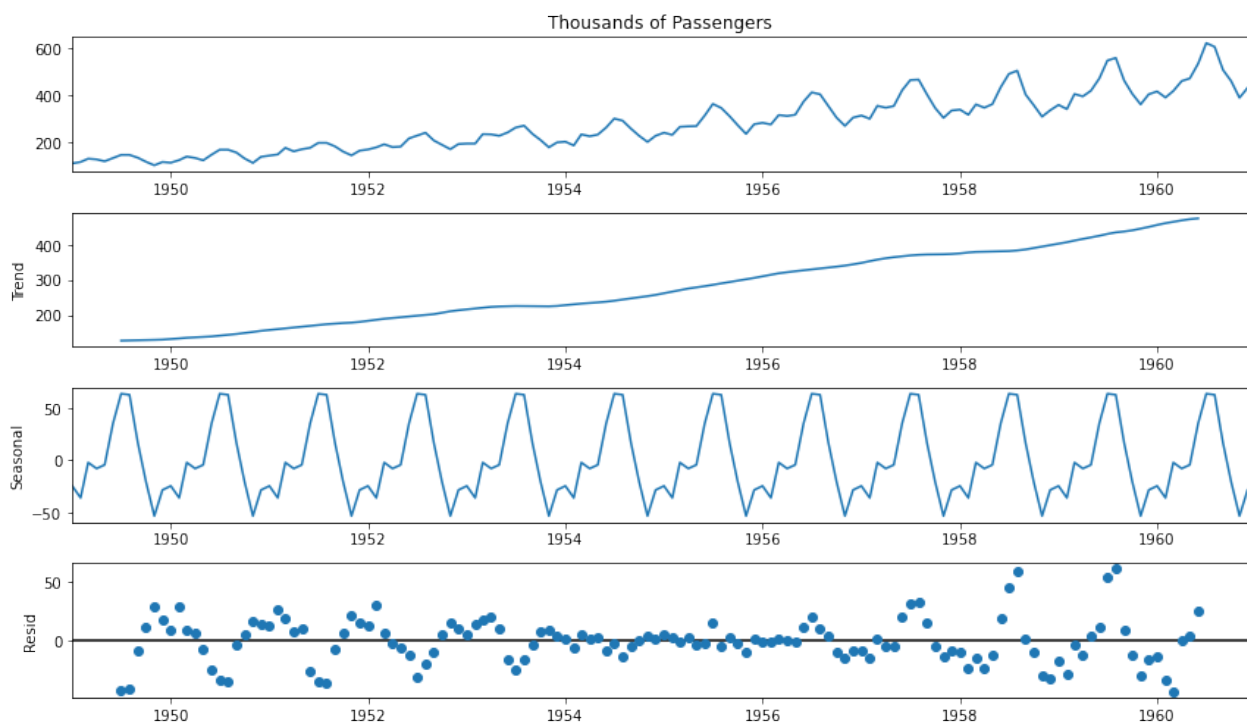
Στην συνέχεια της ανάλυσης της χρονοσειράς δημιουργείται το διάγραμμα Decompose (διάγραμμα 4.15), σύμφωνα με το οποίο παρατηρείται η ύπαρξη αυξητικής τάσης στην χρήση αεροπλάνου με την παροδο του χρόνου, καθώς και η ύπαρξη εποχικότητας. Με την επιβεβαίωση της μη-στασιμότητας της χρονοσειράς μέσω του ελέγχου Dickey-Fuller και των συμπερασμάτων που προέκυψαν από το διάγραμμα Decompose, κατευθυνόμαστε προς την χρήση του εποχικού μοντέλου SARIMA.

Για την αναζήτηση του καταλληλότερου εποχικού μοντέλου SARIMA, δηλαδή του μοντέλου που περιγράφει καλύτερα τη χρονοσειρά, χρησιμοποιείται η ακόλουθη εντολή της Python `auto_arima(df2['ThousandsofPassengers'], start_p = 1, start_q = 1, max_p = 4, max_q = 4, seasonal = True, m = 12).summary()`. Σύμφωνα με την εντολή αυτή το καταλληλότερο μοντέλο SARIMA αναζητείται με βάση τα δεδομένα όλης της χρονοσειράς, περιορίζοντας τις παραμέτρους p και q στα σύνολα (1,4) και (1,4) αντίστοιχα, ορίζοντας την εποχικότητα ως ετήσια ( $m=12$ ). Τελικώς, το καταλληλότερο μοντέλο, με βάση τον εκτιμητή Kalman Filter είναι το SARIMA(0,1,1)(2,1,1,12), δηλαδή το μοντέλο όπου για το μη-εποχικό μέρος αποτελείται από τον όρο με υστέρηση πρώτης τάξης κινητού μέσου και που χρειάζεται να εφαρμοστεί μία φορά η μέθοδος των διαφορών στην χρονοσειρά ώστε να γίνει στάσιμη, ενώ για το εποχικό μέρος αποτελείται από τους πρώτης και δεύτερης υστέρησης αυτοπαλινδρομους όρους και τον όρο πρώτης υστέρησης κινητού μέσου όρου, ενώ τέλος χρειάζεται να εφαρμοσθεί και μία φορά η μέθοδος των διαφορών για να μετατραπεί η εποχική συνιστώσα της χρονοσειράς σε στάσιμη. Έπειτα, διαχωρίζεται το σύνολο δεδομένων σε σύνολο εκπαίδευσης (train test) και σύνολο ελέγχου (test set), όπου το πλήθος των δεδομένων εκπαίδευσης





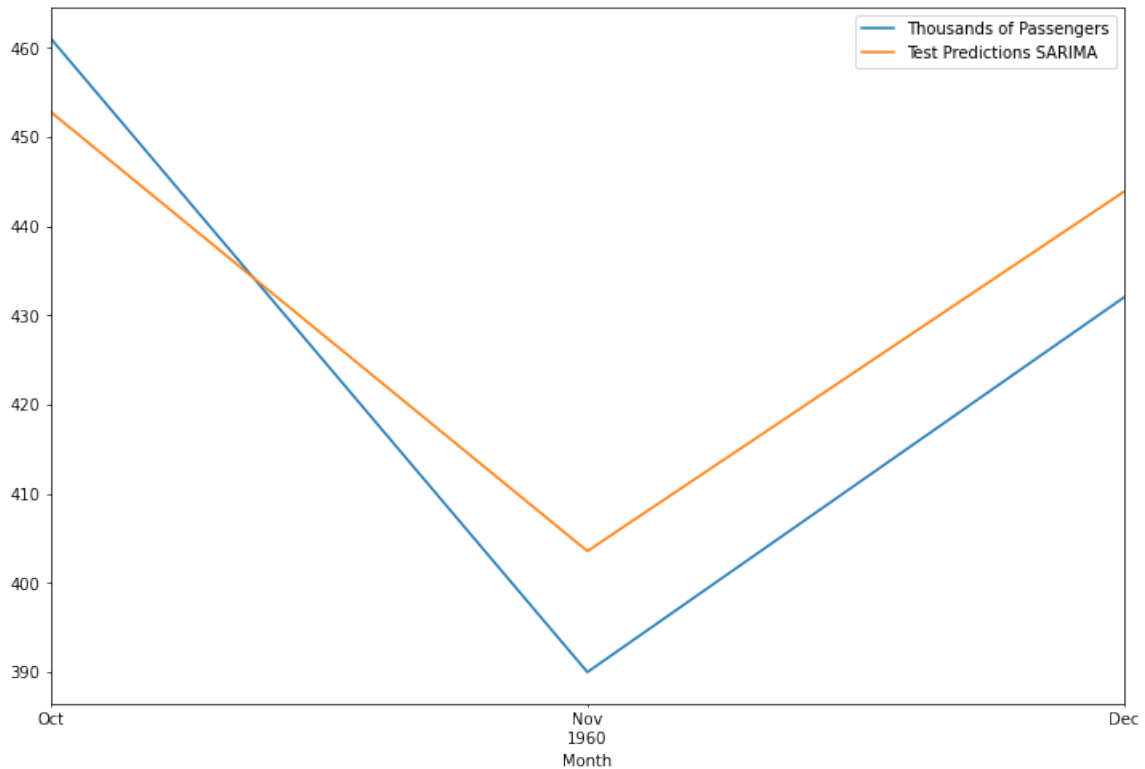
Σχήμα 4.14: Διάγραμμα Χρονοσειράς Επιβατών Αμερικάνικων Αερογραμμών



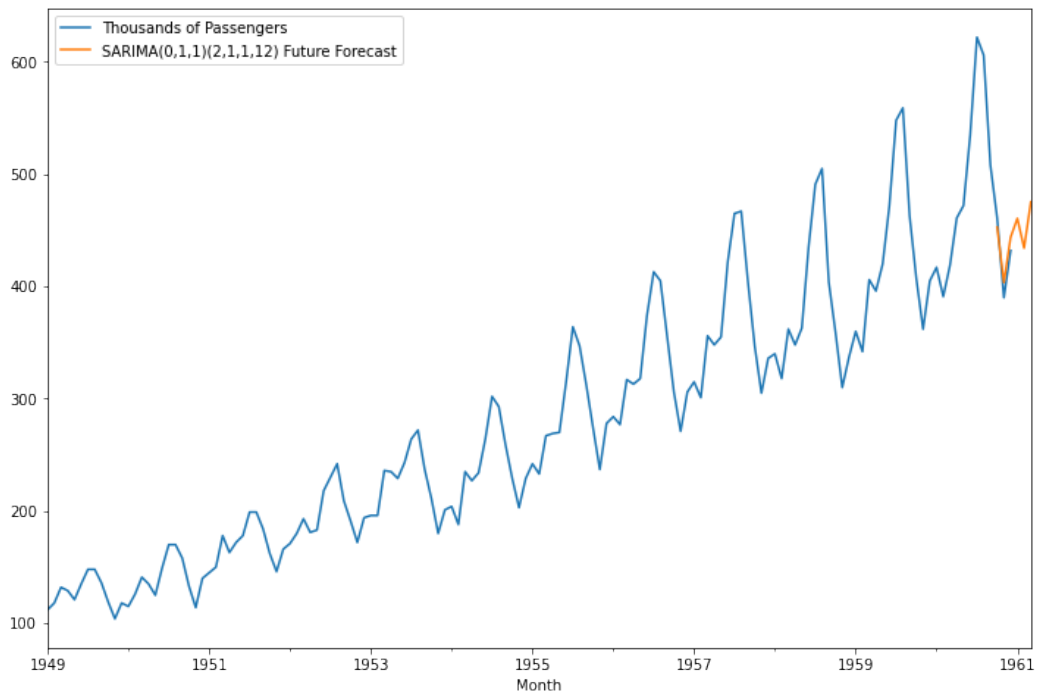
Σχήμα 4.15: Διάγραμμα Decompose

είναι 141 και των δεδομένων ελέγχου είναι 3. Στο σημείο αυτό, το μοντέλο  $SARIMA(0,1,1)(2,1,1,12)$  δομείται και εκπαιδεύεται με βάση τα δεδομένα εκπαίδευσης. Στην συνέχεια, με βάση το μοντέλο που εκπαιδεύτηκε προηγουμένως, δημιουργούνται οι προβλέψεις για τις αντιστοιχες τιμες του συνόλου ελέγχου. Οι προβλέψεις αυτές συγκρίνονται σχηματικά με τα αληθινα δεδομένα ελέγχου στο διάγραμμα 4.16. Από το διάγραμμα 4.16 παρατηρείται ότι οι προβλέψεις (πορτοκαλί γραμμή) του εκπαιδευμένου με τα δεδομένα εκπαίδευσης μοντέλου αποτυπώνουν πολύ

ικανοποιητικά τις πραγματικές τιμές του συνόλου ελέγχου. Η παρατήρηση αυτή επιβεβαιώνεται υπολογίζοντας το Root Mean Square Error (rmse) και συγκρίνοντας το με τις μέσες τιμές των προβλέψεων και του συνόλου ελέγχου. Με την σύγκριση αυτή παρατηρείται ότι η τιμή του rmse είναι κατά πολύ μικρότερη και από τις δύο μέσες τιμές. Ειδικότερα,  $rmse=(test\ set)$ ,  $prediction\ mean=433.39602382466416$ ,  $test\ set\ mean=427.6666666666667$ . Έτσι, το μοντέλο θεωρείται ότι προβλέπει ικανοποιητικά.



Σχήμα 4.16: Διάγραμμα Σύγκρισης Μεταξύ των Δεδομένων Ελέγχου και των Αντίστοιχων Προβλεψεων



Σχήμα 4.17: Διάγραμμα Μελλοντικών Προβλέψεων SARIMA(0,1,1)(2,1,1,12) Μοντέλου

Στο σημείο αυτό πραγματοποιούνται μελλοντικές προβλέψεις για την χρονοσειρά μας. Ειδικότερα, πραγματοποιούνται οι μελλοντικές προβλέψεις για τις επόμενες τρεις μήνες εκτός του συνόλου δεδομένων της χρονοσειράς. Η χρονοσειρά και οι μελλοντικές προβλέψεις διακρίνονται στο διάγραμμα 4.17.

## 4.4 Κώδικας Της Εργασίας

Στην ενότητα αυτή περιγράφεται ο κώδικας που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής. Ορισμένες από τις εντολές που χρησιμοποιήθηκαν στο κομμάτι του κώδικα, ο οποίος αφορά τα στατιστικά μοντέλα προέρχονται από το διαδικτυακό μάθημα του Jose Portilla με τίτλο “Python for Time Series Data Analysis” της ηλεκτρονικής πλατφόρμας του Udemy [22], καθώς και από την ηλεκτρονική ιστοσελίδα [33].

### 4.4.1 Κώδικας Τεχνητού Νευρωνικού Δικτύου

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import pyplot
```

```
df=pd.read_csv ('ibmclose1.csv',sep=',')
df
```

Με τις παραπάνω εντολές εισάγεται το διάνυσμα των παρατηρήσεων των μετοχών της IBM με τις αντίστοιχες ημερομηνίες τους.

```
a=df[:230]
a
b=df[229:]
b
```

Από το αρχικό διάνυσμα παρατηρήσεων **df** δημιουργούνται δύο νέα διανύσματα παρατηρήσεων το **a** και το **b**.

```
w=a['Close']
w
```

Ορίζει ως **w** το διάνυσμα παρατηρήσεων **a**, που περιέχει τις τιμές κλεισίματος των μετοχών της IBM.

```
a.plot(kind='line',x='Date',y='Close')
b.plot(kind='line',x='Date',y='Close')
```

Δημιουργεί τα διαγράμματα των χρονοσειρών των μετοχών κλεισίματος της IBM (**a**) και (**b**).

```
from numpy import array
def split_sequence(sequence, n_steps):
    X, t = list(), list()
    for i in range(len(sequence)):
        end_ix = i + n_steps
        if end_ix > len(sequence)-1:
            break
        seq_x, seq_t = sequence[i:end_ix], sequence[end_ix]
        X.append(seq_x)
        t.append(seq_t)
    return array(X), array(t)
```

```
z=w.values.tolist()
n_steps=2
X, t = split_sequence(z, n_steps)
for i in range(len(X)):
    print(X[i], t[i])
```

Με την χρήση των παραπάνω εντολών δημιουργείται μια συνάρτηση, η οποία δημιουργεί και τυπώνει δύο διανύσματα παρατηρήσεων τα **X** και **t**, όπως αναφέρθηκε και στην ενότητα 4.2. Η συνάρτηση αυτή προέρχεται από την ηλεκτρονική ιστοσελίδα [36].

```
X_train=X[:182,:]
X_test=X[182:,:]
t_train=t[:182]
t_test=t[182:]
```

Με τις παραπάνω εντολές διαχωρίζονται με αναλογία 80:20 τα διανύσματα παρατηρήσεων **X** και **t** σε δεδομένα εκπαίδευσης και ελέγχου.

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler_t=MinMaxScaler()
```

Δημιουργούνται δύο scaler, οι οποίοι στις επόμενες εντολές χρησιμοποιούνται για την κανονικοποίηση με την μέθοδο Min-Max των διανυσμάτων παρατηρήσεων  $\mathbf{X}_{\text{train}}$ ,  $\mathbf{X}_{\text{test}}$ ,  $\mathbf{t}_{\text{train}}$  και  $\mathbf{t}_{\text{test}}$ .

```
X_train1=scaler.fit_transform(X_train)
X_test1=scaler.transform(X_test)
t_train1=scaler_t.fit_transform(t_train.reshape(-1,1))[:,0]
t_test1=scaler_t.transform(t_test.reshape(-1,1))[:,0]
```

Με αυτές τις εντολές οι κανονικοποιητές που δημιουργήθηκαν παραπάνω κανονικοποιούν τα διανύσματα παρατηρήσεων  $\mathbf{X}_{\text{train}}$ ,  $\mathbf{X}_{\text{test}}$ ,  $\mathbf{t}_{\text{train}}$  και  $\mathbf{t}_{\text{test}}$ .

```
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import TimeSeriesSplit
from sklearn.model_selection import GridSearchCV
from matplotlib import pyplot
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_squared_error
import pandas as pd
mlpr = MLPRegressor(max_iter=7000)
param_list = {"hidden_layer_sizes": [(10,),(20,),(10,10),(20,10)], "activation": ["identity",
                                         "logistic", "tanh", "relu"], "solver": ["lbfgs",
                                         "sgd", "adam"], "alpha": [0.00005,0.0005]}
gridCV = GridSearchCV(estimator=mlpr, param_grid=param_list)
```

Οι εντολές αυτές ορίζουν το νευρωνικό δίκτυο που θέλουμε να χρησιμοποιήσουμε, το οποίο είναι το πολυεπίπεδο νευρωνικό δίκτυο Perceptron. Επίσης, δημιουργούν μία λίστα με τις παραμέτρους, τις οποίες θέλουμε να εξετάσουμε ως προς την καταλληλότητα με βάση τα δεδομένα που εισάγονται στο νευρωνικό δίκτυο. Τέλος, δημιουργούν τον έλεγχο με όνομα gridCV, ο οποίος εξετάζει ποιο πολυεπίπεδο δικτύου Perceptron έχει το μικρότερο μέσο τετραγωνικό σφάλμα με βάση τις υπερπαραμέτρους της λίστας που εξετάζουμε.

```
gridCV.fit(X_train1,t_train1)
```

Με βάση τα κανονικοποιημένα pattern εκπαίδευσης εκπαιδεύονται τα πολυεπίπεδα δίκτυα Perceptron με όλους τους πιθανούς συνδυασμούς των υπερπαραμέτρων που έχουν οριστεί στην λίστα.

```
gridCV.best_estimator_
```

Εμφανίζει τις υπερπαραμέτρους του καταλληλότερου πολυεπίπεδου δικτύου Perceptron, δηλαδή το δίκτυο με το μικρότερο μέσο τετραγωνικό σφάλμα.

```
predicted=gridCV.best_estimator_.predict(X_test1)
predicted
```

Προβλέπει και τυπώνει τις τιμές της μετοχής κλεισίματος της IBM με βάση το  $\mathbf{X}_{\text{test}}$ .

```
from sklearn.metrics import mean_squared_error as mse
mse(t_test1,predicted)
```

Υπολογίζει το μέσο τετραγωνικό σφάλμα μεταξύ του  $\mathbf{X}_{\text{test}}$  και των τιμών στόχου  $\mathbf{t}_{\text{test}}$ .

```
plt.plot(t_test1)
plt.plot(predicted)
```

Δημιουργεί τα διαγράμματα των  $\mathbf{t}_{\text{test}}$  και των τιμών ελέγχου που προβλέπει το καταλληλότερο νευρωνικό δίκτυο.

```
arr=[ [0.44312534, 0.51351695] ]
arr1=[ [0.51351695, 0.47950478] ]
arr2=[ [0.47950478, 0.47729769] ]
arr3=[ [0.47729769, 0.46506913] ]
arr4=[ [0.46506913, 0.45565632] ]
arr5=[ [0.45565632, 0.44511243] ]
arr6=[ [0.44511243, 0.4369555] ]
arr7=[ [0.4369555, 0.43008763] ]
arr8=[ [0.43008763, 0.42445527] ]
arr9=[ [0.42445527, 0.419799] ]
arr10=[ [0.419799, 0.4159591] ]
arr11=[ [0.4159591, 0.41279006] ]
arr12=[ [0.41279006, 0.41005085] ]
arr13=[ [0.41005085, 0.40722009] ]
arr14=[ [0.40722009, 0.40434417] ]
arr15=[ [0.40434417, 0.40141794] ]
arr16=[ [0.40141794, 0.39844093] ]
arr17=[ [0.39844093, 0.39541222] ]
```

Δημιουργεί τα διατεταγμένα ζεύγη, που εισάγονται στο εκπαιδευμένο νευρωνικό για να δημιουργηθούν οι μελλοντικές προβλέψεις.

```
pred=gridCV.best_estimator_.predict(arr)
pred2=gridCV.best_estimator_.predict(arr2)
pred3=gridCV.best_estimator_.predict(arr3)
pred4=gridCV.best_estimator_.predict(arr4)
pred5=gridCV.best_estimator_.predict(arr5)
pred6=gridCV.best_estimator_.predict(arr6)
pred7=gridCV.best_estimator_.predict(arr7)
pred8=gridCV.best_estimator_.predict(arr8)
pred9=gridCV.best_estimator_.predict(arr9)
pred10=gridCV.best_estimator_.predict(arr10)
pred11=gridCV.best_estimator_.predict(arr11)
pred12=gridCV.best_estimator_.predict(arr12)
pred13=gridCV.best_estimator_.predict(arr13)
pred14=gridCV.best_estimator_.predict(arr14)
pred15=gridCV.best_estimator_.predict(arr15)
pred16=gridCV.best_estimator_.predict(arr16)
```

Με αυτή την εντολή προβλέπονται οι μελλοντικές τιμές κλεισίματος της μετοχής της IBM, έχοντας ως εισόδους τα παραπάνω διατεταγμένα ζεύγη.

```
scaler.inverse_transform(arr1)
scaler.inverse_transform(arr2)
scaler.inverse_transform(arr3)
scaler.inverse_transform(arr4)
scaler.inverse_transform(arr5)
scaler.inverse_transform(arr6)
scaler.inverse_transform(arr7)
scaler.inverse_transform(arr8)
scaler.inverse_transform(arr9)
scaler.inverse_transform(arr10)
scaler.inverse_transform(arr11)
scaler.inverse_transform(arr12)
scaler.inverse_transform(arr13)
scaler.inverse_transform(arr14)
scaler.inverse_transform(arr15)
scaler.inverse_transform(arr16)
scaler.inverse_transform(arr17)
```

Με την παραπάνω εντολή μετατρέπουμε στην αρχική κλίμακα των τιμών κλεισίματος της μετοχής της IBM τα κανονικοποιημένα διατεταγμένα ζεύγη. Έτσι, υπολογίζουμε τις μελλοντικές προβλέψεις μας στην πραγματική κλίμακα των αρχικών παρατηρήσεων.

#### 4.4.2 Κώδικας ARMA

```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings("ignore")
from statsmodels.tsa.arima_model import ARMA, ARIMA, ARMAResults, ARIMAResults
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
!pip install pmdarima
from pmdarima import auto_arima
from datetime import datetime
```

```
df1=pd.read_csv('DailyTotalFemaleBirths.csv',index_col='Date',parse_dates=True)
df1.index.freq='D'
df1=df1[:120]
```

Με τις 3 παραπάνω εντολές εισάγεται το σύνολο δεδομένων που χρησιμοποιείται, ορίζεται ως ημερήσια η συχνότητα της χρονοσειράς και αποκόπτονται από όλο το σύνολο δεδομένων οι πρώτες 120 παρατηρήσεις. Από αυτό το σημείο και μετά η χρονοσειρά αποτελείται από τις 120 παρατηρήσεις και ορίζεται ως df1.

```
df1['Births'].plot(figsize=(12,5))
```

Δημιουργείται το διάγραμμα της χρονοσειράς.

```
from statsmodels.tsa.stattools import adfuller
adfuller(df1['Births'],autolag='AIC')
```

Εκτελείται ο έλεγχος στασιμότητας Dickey-Fuller και υπολογίζεται η τιμή p-value, μέσω της οποίας επιβεβαιώνεται ή όχι η μηδενική υπόθεση.

```
auto_arima(df1['Births'], seasonal=False).summary()
```

Η συνάρτηση `auto_arima` του πακέτου `rmдарima` βρίσκει με βάση ολόκληρη την χρονοσειρά το στατιστικό μοντέλο που περιγράφει καταλληλότερα την χρονοσειρά, το οποίο είναι το ARMA(2,2).

```
train=df1.iloc[:117]
test=df1.iloc[117:]
```

Δημιουργούνται τα σύνολα εκπαίδευσης και ελέγχου αντίστοιχα.

```
model=ARMA(train['Births'], order=(2,2))
```

Δημιουργείται με βάση το σύνολο εκπαίδευσης το μοντέλο ARMA(2,2).

```
results=model.fit()
results.summary()
```

Το μοντέλο που δημιουργήθηκε εκπαιδεύεται, ενώ η εντολή `summary()` προσφέρει ορισμένες χρήσιμες πληροφορίες για το εκπαιδευμένο μοντέλο, όπως η τιμή του κριτηρίου AIC και οι συντελεστές των όρων του μοντέλου.

```
start=len(train)
end=len(train)+len(test)-1
```

Ορίζεται ως αρχή το μήκος του συνόλου εκπαίδευσης και ως τέλος το άθροισμα του μήκους των συνόλων ελέγχου και εκπαίδευσης μείον την μονάδα.

```
predictions=results.predict(len(train), len(train)+len(test)-1, typ='levels').rename('ARMA(2,2) Predictions')
```

Με την παραπάνω εντολή υπολογίζονται με βάση το εκπαιδευμένο μοντέλο οι τιμές πρόβλεψης (εντός του συνόλου δεδομένων) από την αρχή μέχρι το τέλος που ορίστηκε στις δύο προηγούμενες γραμμές.

```
predictions00=results0.predict(len(train0), len(train0)+len(test0)+2, typ='levels').rename('Test predictions ARMA(2,2)')
```

Με την παραπάνω εντολή το εκπαιδευμένο μοντέλο εκτός από τις προβλεψεις εντός του συνόλου δεδομένων, που χρησιμοποιούνται για την διαδικασία ελέγχου του μοντέλου, προβλέπει και τρεις μελλοντικές τιμές εκτός του συνόλου δεδομένων.

```
test['Births'].plot(figsize=(12,8), legend=True)
predictions.plot(legend=True)
```

Δημιουργείται το διάγραμμα με τις γραφικές παραστάσεις των δεδομένων ελέγχου και των προβλέψεων που υπολογίστηκαν παραπάνω.

```
from statsmodels.tools.eval_measures import rmse, mse
error_df1=rmse(test['Births'], predictions)
test.mean()
predictions.mean()
```

Υπολογίζεται το `rmse` μεταξύ των δεδομένων ελέγχου και των προβλέψεων που παράχθηκαν. Επίσης, υπολογίζονται οι μέσες τιμές των δεδομένων ελέγχου και των προβλέψεων.

```
df1['Births'].plot(legend=True, figsize=(12,8))
predictions00.plot(legend=True)
```

Δημιουργείται το διάγραμμα ολόκληρης της χρονοσειράς και των μελλοντικών προβλέψεων που υπολογίστηκαν.

### 4.4.3 Κώδικας ARIMA

```
df2=pd.read_csv('TradeInventories.csv', index_col='Date', parse_dates=True)
df2.index.freq='MS'
```

Με τις παραπάνω εντολές εισάγεται το σύνολο δεδομένων που χρησιμοποιείται και ορίζεται ως μηνιαία η συχνότητα της χρονοσειράς.

```
df2.plot(figsize=(12,8))
```

Δημιουργείται το διάγραμμα της χρονοσειράς.

```
adfuller(df2['Inventories'], autolag='AIC')
```

Εκτελείται ο έλεγχος στασιμότητας Dickey-Fuller και υπολογίζεται η τιμή p-value, μέσω της οποίας επιβεβαιώνεται ή όχι η μηδενική υπόθεση.

```
from statsmodels.tsa.seasonal import seasonal_decompose
result=seasonal_decompose(df2['Inventories'], model='add')
from pylab import rcParams
rcParams['figure.figsize']=(12,7)
result.plot();
```

Δημιουργείται το διάγραμμα Decompose.

```
!pip install pyramid-arima
from pyramid.arima import auto_arima
auto_arima(df2['Inventories'], start_p=0, start_q=0, max_p=4, max_q=3, seasonal=False, trace=True).
summary()
```

Η συνάρτηση *auto\_arima* του στατιστικού πακέτου *pyramid.arima*, όμοια με τον κώδικα του μοντέλου ARMA, βρίσκει με βάση ολόκληρη την χρονοσειρά το στατιστικό μοντέλο που περιγράφει καταλληλότερα την χρονοσειρά, το οποίο είναι το ARIMA(1,1,1). Για την συγκεκριμένη χρονοσειρά έχουν προστεθεί κάποιοι περιορισμοί σε ορισμένες παραμέτρους του μοντέλου.

```
train_df2=df2.iloc[:261]
test_df2=df2.iloc[261:]
```

Δημιουργούνται τα σύνολα εκπαίδευσης και ελέγχου αντίστοιχα.

```
model_df2=ARIMA(train_df2['Inventories'], order=(1,1,1))
results_df2=model_df2.fit()
results_df2.summary()
```

Δημιουργείται και εκπαιδύεται το μοντέλο ARIMA(1,1,1) με βάση το σύνολο εκπαίδευσης που δημιουργήθηκε στις δύο παραπάνω εντολές. Με την τρίτη εντολή παρέχονται ορισμένες χρήσιμες πληροφορίες για το μοντέλο, όπως η τιμή AIC και οι συντελεστές των όρων του.

```
start_df2=len(train_df2)
end_df2=len(train_df2)+len(test_df2)-1
```

Ορίζεται ως αρχή το μήκος του συνόλου εκπαίδευσης και ως τέλος το άθροισμα του μήκους των συνόλων ελέγχου και εκπαίδευσης μείον την μονάδα.

```
predictions_df2=results_df2.predict(start_df2, end_df2, typ='levels').rename('ARIMA(1,1,1)
Predictions ARIMA')
```

Με την παραπάνω εντολή υπολογίζονται με βάση το εκπαιδευμένο μοντέλο οι τιμές πρόβλεψης (εντός του συνόλου δεδομένων) από την αρχή μέχρι το τέλος που ορίστηκε στις δύο προηγούμενες γραμμές.

```
predictions1=results_df2.predict(len(train_df2), len(train_df2)+len(test_df2)+2, typ='levels').
rename('ARIMA(1,1,1) Future Forecast')
```

Με την παραπάνω εντολή το εκπαιδευμένο μοντέλο εκτός από τις προβλέψεις εντός του συνόλου δεδομένων, που χρησιμοποιούνται για την διαδικασία ελέγχου του μοντέλου, προβλέπει και τρεις μελλοντικές τιμές εκτός του συνόλου δεδομένων.

```
test_df2['Inventories'].plot(legend=True, figsize=(12,8))
predictions_df2.plot(legend=True)
```

Δημιουργείται το διάγραμμα με τις γραφικές παραστάσεις των δεδομένων ελέγχου και των προβλέψεων που υπολογίστηκαν παραπάνω.

```
from statsmodels.tools.eval_measures import rmse, mse
error_df2=rmse(test_df2['Inventories'], predictions_df2)
test_df2['Inventories'].mean()
predictions_df2.mean()
```

Υπολογίζεται το *rmse* μεταξύ των δεδομένων ελέγχου και των προβλέψεων που παράχθηκαν. Επίσης, υπολογίζονται οι μέσες τιμές των δεδομένων ελέγχου και των προβλέψεων.

```
df2['Inventories'].plot(legend=True, figsize=(12,8))
predictions1.plot(legend=True)
```



Δημιουργείται το διάγραμμα ολόκληρης της χρονοσειράς και των μελλοντικών προβλέψεων που υπολογίστηκαν.

```
from statsmodels.tsa.statespace.tools import diff
df2['Diff_1']=diff(df2['Inventories'],k_diff=1)
df2['Diff_1'][1:]
adfuller(df2['Diff_1'][1:])
```

Με τις παραπάνω εντολές εφαρμόζεται μία φορά η μέθοδος των διαφορών στα δεδομένα της χρονοσειράς και στην συνέχεια στην νέα χρονοσειρά που προκύπτει υλοποιείται ο έλεγχος στασιμότητας Dickey-Fuller.

```
plot_acf(df2['Inventories'],lags=40);
plot_pacf(df2['Inventories'],lags=40);
```

Δημιουργούνται τα διαγράμματα αυτοσυσχέτισης και μερικής αυτοσυσχέτισης αντίστοιχα.

#### 4.4.4 Κώδικας SARIMA

```
import pandas as pd
import numpy as np
from statsmodels.tsa.statespace.sarimax import SARIMAX, SARIMAXResults
!pip install pmdarima
from pmdarima import auto_arima
```

```
df3=pd.read_csv('airline_passengers.csv',index_col='Month',parse_dates=True)
df3.index.freq='MS'
```

Με τις παραπάνω εντολές εισάγεται το σύνολο δεδομένων που χρησιμοποιείται και ορίζεται ως μηνιαία η συχνότητα της χρονοσειράς.

```
df3['Thousands of Passengers'].plot(figsize=(12,8))
```

Δημιουργείται το διάγραμμα της χρονοσειράς.

```
from statsmodels.tsa.stattools import adfuller
adfuller(df3['Thousands of Passengers'],autolag='AIC')
```

Εκτελείται ο έλεγχος στασιμότητας Dickey-Fuller και υπολογίζεται η τιμή p-value, μέσω της οποίας επιβεβαιώνεται ή όχι η μηδενική υπόθεση.

```
from statsmodels.tsa.seasonal import seasonal_decompose
result=seasonal_decompose(df3['Thousands of Passengers'],model='add')
from pylab import rcParams
rcParams['figure.figsize']=(12,7)
result.plot();
```

Δημιουργείται το διάγραμμα Decompose.

```
auto_arima(df3['Thousands of Passengers'],start_p=1,start_q=1,max_p=4,max_q=4,seasonal=True,m=12).summary()
```

Η συνάρτηση *auto\_arima* του πακέτου *pyramid.arima*, βρίσκει με βάση ολόκληρη την χρονοσειρά το στατιστικό μοντέλο που περιγράφει καταλλήλότερα την χρονοσειρά, το οποίο είναι το SARIMA(0,1,1)(2,1,1,12). Για την συγκεκριμένη χρονοσειρά έχουν προστεθεί κάποιοι περιορισμοί σε ορισμένες παραμέτρους του μοντέλου.

```
train=df3.iloc[:141]
test=df3.iloc[141:]
```

Δημιουργούνται τα σύνολα εκπαίδευσης και ελέγχου αντίστοιχα.

```
model=SARIMAX(train['Thousands of Passengers'],order=(0,1,1),seasonal_order=(2,1,1,12))
results=model.fit()
results.summary()
```

Δημιουργείται και εκπαιδύεται το μοντέλο SARIMA(0,1,1)(2,1,1,12) με βάση το σύνολο εκπαίδευσης που δημιουργήθηκε στις δύο παραπάνω εντολές. Με την τρίτη εντολή παρέχονται ορισμένες χρήσιμες πληροφορίες για το μοντέλο, όπως η τιμή AIC και οι συντελεστές των όρων του.

```
start=len(train)
end=len(train)+len(test)-1
```

Ορίζεται ως αρχή το μήκος του συνόλου εκπαίδευσης και ως τέλος το άθροισμα του μήκους των συνόλων ελέγχου και εκπαίδευσης μείον την μονάδα.

```
predictions=results.predict(start,end,typ='levels').rename('SARIMA(0,1,1)(2,1,1,12)
Predictions SARIMA')
```

Με την παραπάνω εντολή υπολογίζονται με βάση το εκπαιδευμένο μοντέλο οι τιμές πρόβλεψης (εντός του συνόλου δεδομένων) από την αρχή μέχρι το τέλος που ορίστηκε στις δύο προηγούμενες γραμμές.

```
predictions31=results3.predict(len(train3),len(train3)+len(test3)+2,typ='levels').rename('
SARIMA(0,1,1)(2,1,1,12) Future Forecast')
```

Με την παραπάνω εντολή το εκπαιδευμένο μοντέλο εκτός από τις προβλεψεις εντός του συνόλου δεδομένων, που χρησιμοποιούνται για την διαδικασία ελέγχου του μοντέλου, προβλέπει και τρεις μελλοντικές τιμές εκτός του συνόλου δεδομένων.

```
test['Thousands of Passengers'].plot(legend=True,figsize=(12,8))
predictions.plot(legend=True)
```

Δημιουργείται το διάγραμμα με τις γραφικές παραστάσεις των δεδομένων ελέγχου και των προβλέψεων που υπολογίστηκαν παραπάνω.

```
from statsmodels.tools.eval_measures import rmse,mse)
error=rmse(test['Thousands of Passengers'],predictions)
test['Thousands of Passengers'].mean()
predictions.mean()
```

Υπολογίζεται το rmse μεταξύ των δεδομένων ελέγχου και των προβλέψεων που παράχθηκαν. Επίσης, υπολογίζονται οι μέσες τιμές των δεδομένων ελέγχου και των προβλέψεων.

```
df3['Thousands of Passengers'].plot(legend=True,figsize=(12,8))
predictions31.plot(legend=True)
```

Δημιουργείται το διάγραμμα ολόκληρης της χρονοσειράς και των μελλοντικών προβλέψεων που υπολογίστηκαν.

# Βιβλιογραφία

- [1] Tom M. Mitchell, *Machine Learning*, Burr Ridge, IL: McGraw Hill 45.37 (1997): 870-877
- [2] Arthur L. Samuel, *Some Studies in Machine Learning Using the Game of Checkers*, IBM Journal of Research and Development 3.3 (1959): 210-229
- [3] Kevin P. Murphy, *Machine learning: A Probabilistic Perspective*, MIT press (2012)
- [4] Ian Goodfellow, Yoshua Bengio and Aaron Courville, *Deep Learning*, MIT press (2016)
- [5] Mark JM. Bishop, *History and Philosophy of Neural Networks*, (2015)
- [6] Αχιλλέας Ζαπράνης, *Financial and Neural Systems*, Εκδόσεις Κλειδάριθμος (2005)
- [7] Donald O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*, Psychology Press (2005)
- [8] Marvin L. Minsky and Seymour A. Papert, *A Review of Perceptrons: An Introduction to Computational Geometry*, MIT press Cambridge Mass (1969)
- [9] James L. McClelland, David E. Rumelhart, and PDP Research Group, *Parallel distributed processing*, Explorations in the Microstructure of Cognition 2 (1986): 216-271
- [10] Πάνος Αργυράκης, *Neural Networks and Applications*, Εκδόσεις Ελληνικό Ανοικτό Πανεπιστήμιο (2001)
- [11] Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams, *Learning Representations by Back-Propagating Errors* Nature Publishing Group 323.6088 (1986): 533-536
- [12] Sagar Sharma, *Activation Functions in neural Networks*, Towards Data Science (2017)
- [13] Matt W. Gardner, and S. R. Dorling, *Artificial Neural Networks (The Multilayer Perceptron) — A Review of Applications in the Atmospheric Sciences*, Atmospheric environment 32.14-15 (1998): 2627-2636
- [15] Σοφία Δημέλη, *Σύγχρονες Μέθοδοι Ανάλυσης Χρονολογικών Σειρών*, Εκδόσεις ΟΠΑ, Αθήνα (2013)
- [16] Αλέξανδρος Καραγρηγορίου, *Σημειώσεις του μαθήματος: Ανάλυση Χρονοσειρών*, Πανεπιστήμιο Αιγαίου (2019)
- [18] Κουγιουμτζής Δημήτριος, *Σημειώσεις του μαθήματος: Ανάλυση Χρονοσειρών*, Α.Π.Θ. (2019)
- [19] Λευτέρης Ι. Θαλασσινός, *Ανάλυση Χρονολογικών Σειρών*, Εκδόσεις Α. Σταμούλης, Πειραιάς (1991)
- [20] [http://www.math.ntua.gr/~kokolakis/SEMFE/TimeSeries\\_Ch\\_1.pdf](http://www.math.ntua.gr/~kokolakis/SEMFE/TimeSeries_Ch_1.pdf)
- [21] <http://users.auth.gr/dkugiu/Teach/DataAnalysis/Chp6.pdf>
- [22] <https://www.udemy.com/course/python-for-time-series-data-analysis/>
- [23] <https://finance.yahoo.com/>
- [24] <https://eclass.emt.ihu.gr/modules/document/file.php/DBA203/>
- [25] <https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bf>
- [26] [https://en.wikipedia.org/wiki/Moving-average\\_model](https://en.wikipedia.org/wiki/Moving-average_model)
- [27] <https://en.wikipedia.org/wiki/Autoregressive>
- [28] [https://en.wikipedia.org/wiki/Autoregressive\\_integrated\\_moving\\_average](https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average)

- [29] <https://towardsdatascience.com/significance-of-acf-and-pacf-plots-in-time-series-analysis-2fa11a5d>
- [30] <https://people.duke.edu/~rnau/411arim3.htm>
- [31] Laurene Fausett, *Fundamental of Neural Networks, Architectures, Algorithms and Applications: Printice Hall int.* (1986)
- [32] <https://www.igi-global.com/dictionary/decision-making-systems/68813>
- [33] <https://medium.com/@josemarcialportilla/using-python-and-auto-arma-to-forecast-seasonal-time-series>
- [34] Θεοδόση-Κόκκινου Λάουρα, Διπλωματική εργασία με θέμα “Τεχνητά Νευρωνικά Δίκτυα και Εφαρμογές στα Συστήματα Αυτόματου Ελέγχου”, Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, Πανεπιστήμιο Πατρών (2013)
- [35] Δημήτριος Πήτας, Πτυχιακή εργασία με θέμα “Τεχνητά Νευρωνικά Δίκτυα και Εφαρμογή”, Σχολή Θετικών Επιστημών, Τμήμα Μαθηματικών, Πανεπιστήμιο Αιγαίου (2018)
- [36] <https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/>
- [37] Zabir Haider Khan, Tasnim Sharmin Alin, and Md Akter Hussain, textitPrice prediction of share market using artificial neural network (ANN), *International Journal of Computer Applications*, 22.2 (2011): 42-47.
- [38] Parsinejad Shahbaz, Bagheri Ahmad, Ebrahimi Atani Reza, et all, textitStock Market Forecasting Using Artificial Neural Networks, *European Online Journal of Natural and Social Sciences: Proceedings*, 2.3 (s) (2014): pp-2404.
- [39] Ramon Lawrence, *Using neural networks to forecast stock market prices*, University of Manitoba, 333 (1997): 2006-2013.
- [40] <https://dothanblog.wordpress.com/2019/12/29/batch-normalization/>
- [41] [http://psi-gr.tripod.com/choc\\_20\\_app\\_neuro.html](http://psi-gr.tripod.com/choc_20_app_neuro.html)
- [42] <https://aiandi.fr/comment-fonctionne-un-reseau-de-neurones/>
- [43] <https://www.learnopencv.com/image-classification-using-feedforward-neural-network-in-keras/>
- [44] [http://www.jpathinformatics.org/viewimage.asp?img=JPatholInform\\_2014\\_5\\_1\\_9\\_129442\\_f7.jpg](http://www.jpathinformatics.org/viewimage.asp?img=JPatholInform_2014_5_1_9_129442_f7.jpg)
- [45] <https://stackoverflow.com/questions/50394921/keras-explanation-number-of-nodes-in-input-layer>
- [46] <https://programmersought.com/article/4868807525/>