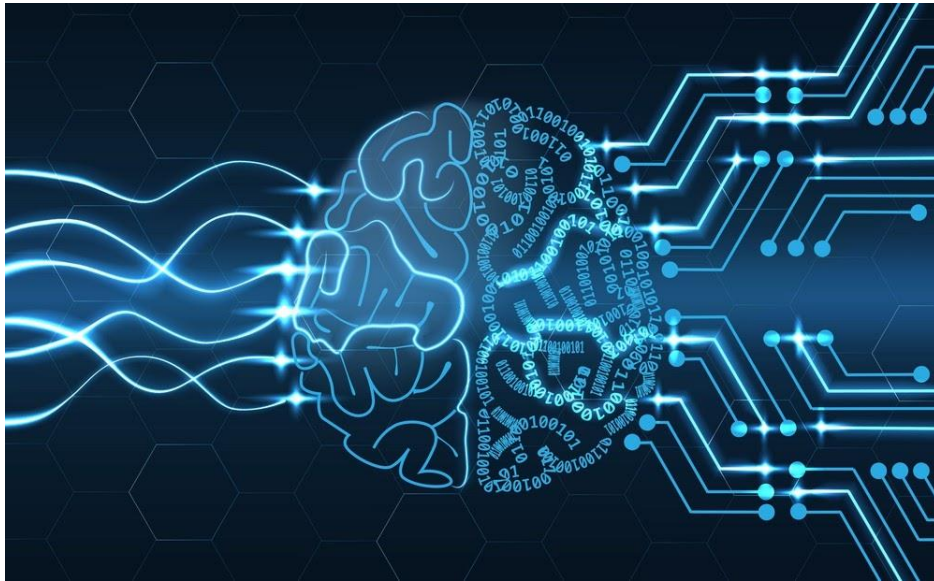




ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
Π.Μ.Σ. ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΚΑΙ ΥΠΗΡΕΣΙΕΣ



Διπλωματική Εργασία:

**Μελέτη νευρωνικών δικτύων και εφαρμογές στην πρόβλεψη
χρονοσειρών στο χρηματιστήριο**

Μελάς Χρήστος

Επιβλέπων : Μιχαήλ Φιλιππάκης

Αθήνα Φεβάρης 2020

Περιεχόμενα

ΠΕΡΙΛΗΨΗ	4
ΕΙΣΑΓΩΓΗ	5
ΜΕΡΟΣ 1: ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	6
1.1. Εισαγωγή στη Μηχανική Μάθηση	6
1.2. Είδη Μηχανικής Μάθησης	6
1.3. Αρχιτεκτονικές Νευρωνικών Δικτύων	7
1.3.1. Multilayer Perceptrons (MLPs)	9
1.3.2. Restricted Boltzmann Machines (RBMs)	11
1.3.3. Deep Belief Networks (DBFs)	12
1.3.4. Autoencoders	13
1.3.5. Convolutional Neural Networks (CNNs)	14
1.3.6. Recurrent Neural Networks (RNNs)	16
1.3.7. Long-Short Term Memory Neural Networks (LSTMNs)	17
1.3.8. Hopfield Neural Networks (HNNs)	19
1.3.9. Generative Adversarial Networks (GANs)	20
1.3.10. Gated Recurrent Units (GRUs)	21
ΜΕΡΟΣ 2: ΧΡΟΝΟΣΕΙΡΕΣ	22
2.1. Εισαγωγή στις Χρονοσειρές	22
2.2. Βασικά Χαρακτηριστικά Χρονοσειρών	23
2.2.1. Τάση (Trend)	23
2.2.2. Κυκλικότητα (Cyclic)	24
2.2.3. Ακραίες τιμές (Outliers)	25
2.2.4. Εποχικότητα (Seasonality)	25
2.3. Ανάλυση Χρονοσειρών	26
ΜΕΡΟΣ 3: ΠΕΙΡΑΜΑΤΙΚΟ ΣΚΕΛΟΣ	27
3.1. Βασική Ιδέα	27
3.2. Διερεύνηση Δεδομένων (Exploratory Analytics)	27
3.2.1. Τα Δεδομένα (Dataset)	27
3.2.2. Αρχική Επεξεργασία Δεδομένων	28
3.2.3. Οπτικοποίηση Χρονοσειρών	29
3.3. Μοντελοποίηση Νευρωνικών Δικτύων	32
3.3.1. Τεχνητά Νευρωνικά Δίκτυα προς Μοντελοποίηση	32
3.3.2. Keras Configuration	33
3.4. Multilayer Perceptron (MLP)	33

3.4.1. Αποτελέσματα MLP για το Facebook	34
3.4.2. Αποτελέσματα MLP για τη Google.....	36
3.4.3. Αποτελέσματα MLP για την Amazon	38
3.5. Long-Short Term Memory Network (LSTMN).....	40
3.5.1. Αποτελέσματα LSTMN για το Facebook.....	40
3.5.2. Αποτελέσματα LSTMN για τη Google	42
3.5.3. Αποτελέσματα LSTMN για την Amazon.....	43
3.6. Convolutional Neural Network (CNN)	45
3.6.1. Αποτελέσματα CNN για το Facebook.....	45
3.6.2. Αποτελέσματα CNN για τη Google	47
3.6.3. Αποτελέσματα CNN για την Amazon.....	48
ΣΥΜΠΕΡΑΣΜΑΤΑ	50
ΜΕΛΛΟΝΤΙΚΟΙ ΣΤΟΧΟΙ.....	51
ΒΙΒΛΙΟΓΡΑΦΙΑ	52

ΠΕΡΙΛΗΨΗ

Τα τελευταία χρόνια τα τεχνητά νευρωνικά δίκτυα παρουσιάζουν μία συνεχή αλματώδη ανάπτυξη και αποτελούν την αιχμή του δόρατος του επιστημονικού πεδίου της μηχανικής μάθησης. Η χρήση τους έχει αρχίσει να διευρύνεται, πέραν των στενών ορίων της επίλυσης προβλημάτων επεξεργασίας δεδομένων εικόνας και ήχου, σε πληθώρα τεχνολογικών εφαρμογών του σύγχρονου τεχνολογικού και οικονομικού περιβάλλοντος στο οποίο ζούμε. Μία από αυτές τις σύγχρονες εφαρμογές των τεχνητών νευρωνικών δικτύων, είναι η επεξεργασία, η ανάλυση και η πρόβλεψη χρηματιστηριακών χρονοσειρών.

Με την παρούσα διπλωματική εργασία θα επιχειρήσουμε αρχικά να παρουσιάσουμε τις βασικότερες αρχιτεκτονικές νευρωνικών δικτύων. Στη συνέχεια θα προσπαθήσουμε να μοντελοποιήσουμε εκείνα που χρησιμοποιούνται ευρέως για την ανάλυση χρονοσειρών χρηματιστηριακού τύπου, χρησιμοποιώντας τη γλώσσα προγραμματισμού Python. Τέλος με τη χρήση πραγματικών δεδομένων θα προσπαθήσουμε να αξιολογήσουμε τον τρόπο με τον οποίο λειτουργούν και την απόδοσή τους, με απώτερο στόχο την εξαγωγή χρήσιμων συμπερασμάτων.

ΕΙΣΑΓΩΓΗ

Όπως μπορεί να γίνει εύκολα αντιληπτό όσο γιγαντώνεται η χρήση του διαδικτύου μέσω της αύξησης των χρηστών αλλά και των δυνατοτήτων που παρέχονται προς αυτούς, τόσο αυξάνονται και οι πληροφορίες, δηλαδή τα δεδομένα, που διακινούνται σε αυτό. Ειδικά τη τελευταία δεκαετία, με την είσοδο στη ζωή μας των μέσων κοινωνικής δικτύωσης, η αυξητική τάση συντελείται με γεωμετρική πρόοδο. Απόρροια του γεγονότος αυτού είναι η συνεχής αναζήτηση τρόπων και εργαλείων για τη βέλτιστη αξιοποίηση του ολοένα και αυξανόμενου όγκου δεδομένων μέσω του επιστημονικού πεδίου της Επιστήμης των Δεδομένων.

Ένα από τα δομικά στοιχεία της Επιστήμης των Δεδομένων είναι το επιστημονικό πεδίο της Μηχανικής Μάθησης, το οποίο μεταφράζεται ως η προσπάθεια να αποδοθεί στους υπολογιστές η ικανότητα να εκτελούν εντολές χωρίς να έχουν προγραμματιστεί ρητά για αυτές, με τη χρήση μαθηματικών αλγορίθμων. Όταν οι αλγόριθμοι αυτοί προσομοιάζουν ένα δίκτυο από απλούς υπολογιστικούς κόμβους που συνδέονται μεταξύ τους και παραπέμπουν ως προς τη δομή τους στη μορφολογία του ανθρώπινου κεντρικού νευρικού συστήματος, τότε ονομάζονται τεχνητά νευρωνικά δίκτυα.

Το παγκοσμιοποιημένο και τρομερά ευμετάβλητο οικονομικό περιβάλλον το οποίο έχει διαμορφωθεί σε συνδυασμό με το πλήθος των διαθέσιμων δεδομένων, έχει ως αποτέλεσμα να χρησιμοποιούνται όλο και περισσότερο τα τεχνητά νευρωνικά δίκτυα σε διάφορους τομείς του χρηματοπιστωτικού τομέα, προκειμένου να προβλέψουν την πορεία των διεθνών κεφαλαιουχικών αγορών. Στην παρούσα διπλωματική εργασία θα επιχειρηθεί η χρήση τεχνητών νευρωνικών δικτύων για την ανάλυση χρονολογικών δεδομένων που αφορούν μετοχές εταιρειών εισηγμένες στο χρηματιστήριο, με απώτερο σκοπό τη βέλτιστη δυνατή μελλοντική πρόβλεψη.

Αρχικά θα παρουσιάσουμε τις βασικότερες αρχιτεκτονικές των τεχνητών νευρωνικών δικτύων, στη συνέχεια θα αναφέρουμε τις βασικές ιδιότητες και τα κυριότερα χαρακτηριστικά των χρονοσειρών και στο τέλος θα προσπαθήσουμε να μοντελοποιήσουμε τις σημαντικότερες αρχιτεκτονικές τεχνητών νευρωνικών δικτύων, που χρησιμοποιούνται για την ανάλυση χρονοσειρών, με απώτερο σκοπό να αξιολογήσουμε την απόδοσή τους στην πρόβλεψη της εξέλιξης χρηματιστηριακών δεδομένων.

ΜΕΡΟΣ 1: ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

1.1. Εισαγωγή στη Μηχανική Μάθηση

Με τον όρο Μηχανική Μάθηση αναφερόμαστε στο σύνολο των μαθηματικών αλγορίθμων που χρησιμοποιούνται για την εκπαίδευση υπολογιστικών συστημάτων, τα οποία επιφορτίζονται με τη διεκπεραίωση καθημερινών εργασιών που μέχρι πρότινος εκτελούνταν συνήθως με τη χρήση ανθρώπινου έργου. Ουσιαστικά ο πυρήνας της Μηχανικής Μάθησης είναι ο αλγόριθμος εκπαίδευσης, δηλαδή ένας αλγόριθμος που έχει την ικανότητα να μάθει χρησιμοποιώντας διάφορες πηγές δεδομένων και να αυξάνει με την πάροδο του χρόνου αυτή του την ικανότητα μέσω επαναληπτικών διεργασιών.

Η Μηχανική Μάθηση αποτελεί ένα από τα επιστημονικά πεδία που απαρτίζουν το γνωστικό αντικείμενο της Τεχνητής Νοημοσύνης και μας δίνει τη δυνατότητα να αντιμετωπίσουμε πολύπλοκα προβλήματα, η επίλυση των οποίων είναι πολύ δύσκολη με τη χρήση προγραμμάτων που γράφονται και σχεδιάζονται αποκλειστικά από ανθρώπους. Από επιστημονική άποψη η Μηχανική Μάθηση παρουσιάζει πολύ μεγάλο ενδιαφέρον, καθώς η ανάπτυξη της κατανόησης μας για αυτή συνεπάγεται με την ανάπτυξη της κατανόησης μας για τις αρχές που διέπουν τη νοημοσύνη γενικότερα.

Η Μηχανική Μάθηση τέθηκε σε εφαρμογή για πρώτη φορά σαν ιδέα από τον Arthur Samuel το 1952 στην IBM, με τη δημιουργία ενός προγράμματος που παρατηρούσε και πρότεινε τις βέλτιστες κινήσεις για το παιχνίδι της ντάμας και το οποίο γινόταν όλο και καλύτερο στις προβλέψεις με το πέρασμα του χρόνου. Τις επόμενες δεκαετίες που ακολούθησαν το πρώτο αυτό πείραμα η ανάπτυξη της Μηχανικής Μάθησης υπήρξε αλματώδης με τη δημιουργία συνεχώς νέων μοντέλων όπως ο αλγόριθμός του Rosenblatt Perceptron το 1957, ο αλγόριθμος Backpropagation το 1981, τα δέντρα απόφασης και ο αλγόριθμος ID3 το 1986, ο αλγόριθμος Support Vector Machines το 1995 και τα πολύπλοκα μοντέλα που ανήκουν στο φάσμα της βαθιάς μάθησης (deep learning και διαδραματίζουν σημαίνοντα ρόλο στις τεχνολογικές εξελίξεις των ημερών μας.

Διανύοντας την εποχή της έκρηξης των Μεγάλων Δεδομένων η Μηχανική Μάθηση, με την υποστήριξη της ολοένα και αυξανόμενης υπολογιστικής ισχύς που μας προσφέρει ο τεχνολογικός κλάδος της νεφούπολογιστικής (cloud computing), χρησιμοποιείται σε ένα ευρύ φάσμα δραστηριοτήτων όπως στις μηχανές αναζήτησης, στον τομέα της υγείας, στο ηλεκτρονικό εμπόριο, στην εξερεύνηση του διαστήματος, στη ρομποτική, στη διαχείριση πληροφοριών κ.α..

Η ολοένα και αυξανόμενη χρήση της Μηχανικής Μάθησης είναι απόρροια των συγκριτικών της πλεονεκτημάτων όπως των ικανοτήτων της να επεξεργάζεται δεδομένα και να κάνει προβλέψεις σε πραγματικό χρόνο, να δέχεται δεδομένα από πολλαπλές πηγές, να παράγει πολυδιάστατη οπτική στα δεδομένα, να παίρνει αποφάσεις και να προσαρμόζεται στις εκάστοτε αλλαγές που προκύπτουν [1], [2].

1.2. Είδη Μηχανικής Μάθησης

Το επιστημονικό πεδίο της Μηχανικής Μάθησης αποτελείται από τρεις διαφορετικές τεχνικές προσεγγίσεις, όσον αφορά την εκπαίδευση των αλγορίθμων, οι οποίες

αντικατοπτρίζουν τους τρόπους με τους οποίους μαθαίνει ο άνθρωπος. Αυτές είναι η επιβλεπόμενη μάθηση, η μη επιβλεπόμενη μάθηση και η ενισχυτική μάθηση.

Επιβλεπόμενη Μάθηση (Supervised Learning) είναι η διαδικασία κατά την οποία οι αλγόριθμοι της Μηχανικής Μάθησης δημιουργούν συναρτήσεις οι οποίες προβάλουν συγκεκριμένες εισόδους (σύνολο εκπαίδευσης) σε γνωστές επιθυμητές εξόδους, με τον τελικό στόχο της να είναι η γενίκευση των αποτελεσμάτων αυτών και σε εισόδους με άγνωστη έξοδο. Το συγκεκριμένο είδος Μηχανικής Μάθησης χρησιμοποιείται κυρίως για την αντιμετώπιση προβλημάτων ταξινόμησης (classification), πρόγνωσης (prediction) και διερμηνείας (interpretation).

Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning) είναι η διαδικασία κατά την οποία οι αλγόριθμοι της Μηχανικής Μάθησης μοντελοποιούν ένα σύνολο παρατηρήσεων, οι οποίες διαδραματίζουν το ρόλο των δεδομένων εισόδου, χωρίς να έχουν γνώση των επιθυμητών εξόδων. Οι παραπάνω αλγόριθμοι χρησιμοποιούνται κυρίως για την αντιμετώπιση προβλημάτων ομαδοποίησης (clustering) και ανάλυσης συσχετισμών (association analysis).

Ενισχυτική Μάθηση (Reinforcement Learning) είναι η διαδικασία κατά την οποία οι αλγόριθμοι Μηχανικής Μάθησης αλληλεπιδρούν με το περιβάλλον με πολύ άμεσο τρόπο και εκπαιδεύονται σε μία συγκεκριμένη στρατηγική ενεργειών. Το συγκεκριμένο είδος Μηχανικής Μάθησης χρησιμοποιείται κυρίως για την επίλυση προβλημάτων σχεδιασμού (planning), όπως η διαχείριση των αποφάσεων ενός συστήματος αυτόνομης οδήγησης [3].

Ορισμένοι αλγόριθμοι Μηχανικής Μάθησης είναι εμπνευσμένοι από τον τρόπο λειτουργίας των βιολογικών συστημάτων μάθησης που χρησιμοποιούνται από τους έμβιους οργανισμούς, κυρίως του ανθρώπινου εγκεφάλου και τα οποία αποτελούνται από ένα πολύπλοκο δίκτυο υπερσυνδεδεμένων νευρώνων. Οι αλγόριθμοι αυτοί ονομάζονται **Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks)** και αποτελούνται από ένα πυκνό σύνολο υπερσυνδεδεμένων απλών μονάδων, με την κάθε μονάδα να παίρνει έναν αριθμό από εισόδους πραγματικών τιμών και αφού τις επεξεργαστεί, παράγει μία έξοδο πραγματικής τιμής.

Η διαδικασία αυτή πραγματοποιείται στα τρία επίπεδα ενός τυπικού Τεχνητού Νευρωνικού Δικτύου και τα οποία είναι το επίπεδο εισόδου, το κρυμμένο και το επίπεδο εξόδου. Υπάρχουν Τεχνητά Νευρωνικά Δίκτυα τα οποία μπορούν να έχουν περισσότερα επίπεδα, ειδικά όσον αφορά τον αριθμό των κρυμμένων επιπέδων τους, όπως και πολύ μεγάλο αριθμό νευρώνων. Η μελέτη και η δημιουργία των Τεχνητών Νευρωνικών Δικτύων που παρουσιάζουν πολύ μεγάλη πολυπλοκότητα ως προς τη δομή του συνθέτουν ένα υποείδος της Μηχανικής Μάθησης, το οποίο ονομάζεται **Βαθιά Μάθηση (Deep Learning)** [4].

1.3. Αρχιτεκτονικές Νευρωνικών Δικτύων

Προκειμένου να κατανοήσουμε περαιτέρω τις ιδιότητες και τα πλεονεκτήματα των νευρωνικών δικτύων κρίνεται απαραίτητη η αναλυτικότερη μελέτη των σημαντικότερων αρχιτεκτονικών νευρωνικών δικτύων που χρησιμοποιούνται ευρέως τα τελευταία χρόνια, καθώς και των επιμέρους χαρακτηριστικών τους.

Όλα τα τυπικά τεχνητά νευρωνικά δίκτυα, εκτός από τα τρία βασικά επίπεδα (εισόδου, κρυμμένο και εξόδου) που αναφέρθηκαν προηγουμένως, περιλαμβάνουν και άλλες παραμέτρους που καθορίζουν τον τρόπο με τον οποίο λειτουργούν και

συμπεριφέρονται. Δύο τέτοιες βασικές παράμετροι είναι οι συναρτήσεις **εισαγωγής και ενεργοποίησης**, οι οποίες ελέγχουν τις λειτουργίες των επιμέρους μονάδων που συνθέτουν τα επίπεδα του νευρωνικού δικτύου και συνήθως είναι οι ίδιες σε κάθε επίπεδο του νευρωνικού δικτύου.

Με τον όρο εισαγωγή στον κόμβο ενός νευρωνικού δικτύου εννοούμε το σταθμικό άθροισμα των αποτελεσμάτων εξόδου των κόμβων που συνδέονται μαζί του.

Μία τυπική **συνάρτηση εισαγωγής** έχεις συνήθως την παρακάτω μορφή:

$$\text{net} = \sum_j w_{ij} x_j + \mu_i$$

Όπου το net_i περιγράφει το αποτέλεσμα που έχουν τα στοιχεία εισόδου του δικτύου x_i (σταθμισμένα από τα βάρη w_{ij}) στη μονάδα i . Ακόμη με w_{ij} συμβολίζουμε τα βάρη που συνδέουν τον νευρώνα j με το νευρώνα i , με x_j το αποτέλεσμα εξόδου της μονάδας j και με μ_i το κατώφλι για το νευρώνα i . Με τον όρο κατώφλι ονομάζουμε την τιμή εισόδου που λειτουργεί σαν σημείο αναφοράς σε ένα κόμβο εν απουσία άλλων τιμών εισόδου.

Στη συνέχεια κάθε μονάδα του τεχνητού νευρωνικού δικτύου παίρνει τις τιμές εισόδου που περιγράφονται από τη μεταβλητή net_i και εφαρμόζει μία **συνάρτηση ενεργοποίησης** πάνω σε αυτές. Για παράδειγμα το αποτέλεσμα της j^{th} μονάδας ή αλλιώς τιμή ενεργοποίησης, υπολογίζεται από τον τύπο:

$$g(\sum w_{ij} x_i)$$

Όπου $g(\cdot)$ είναι η συνάρτηση ενεργοποίησης και το x_i είναι το αποτέλεσμα της i^{th} μονάδας που είναι συνδεδεμένη με τη μονάδα j . Υπάρχει ένας σημαντικός αριθμός μη γραμμικών συναρτήσεων που χρησιμοποιούνται ως συναρτήσεις ενεργοποίησης νευρωνικών δικτύων, αυτές όμως που χρησιμοποιούνται πιο συχνά είναι σιγμοειδείς συναρτήσεις της μορφής:

$$g(\text{netinput}) = [1 + e^{-\text{netinput}}]^{-1}$$

Είναι προφανές ότι η συνάρτηση ενεργοποίησης έχει μεγάλο αντίκτυπο στη συμπεριφορά και την απόδοση ενός νευρωνικού δικτύου, ενώ τα χαρακτηριστικά της ποικίλουν ανάλογα την περίπτωση. Ο κύριος σκοπός της συνάρτησης ενεργοποίησης είναι να καθορίσει τις απομακρυσμένες τιμές των αποκτηθέντων μονάδων εισόδου πίσω σε ένα διάστημα εμπιστοσύνης όπως το $[0,1]$ ή το $[-1,1]$.

Επομένως συνοψίζοντας καταλήγουμε στο συμπέρασμα ότι τα συνθετικά μέρη που απαρτίζουν την αρχιτεκτονική και τη δομή ενός τυπικού νευρωνικού δικτύου είναι τα παρακάτω:

- Ο αριθμός των κρυμμένων επιπέδων: Ένα νευρωνικό δίκτυο μπορεί να έχει ένα ή περισσότερα κρυμμένα επίπεδα, τα οποία δίνουν τη δυνατότητα στο νευρωνικό δίκτυο να βρίσκει τα στοιχεία που σχετίζονται μεταξύ τους. Ακόμη και δίκτυα με ένα ή δύο κρυμμένα επίπεδα εάν έχουν σημαντικό αριθμό νευρώνων αποδίδουν πολύ καλά.
- Ο αριθμός των κρυμμένων κόμβων: Δεν υπάρχει απόλυτα συγκεκριμένος τρόπος που μπορεί να ακολουθήσει κάποιος για να επιλέξει το βέλτιστο αριθμό κρυμμένων κόμβων ενός τεχνητού νευρωνικού δικτύου. Η επιλογή αυτή

απαιτεί τη διενέργεια πειραματικών δοκιμών και αξιολόγηση της αποδοτικότητας του τεχνητού νευρωνικού δικτύου ανά δοκιμή. Υπάρχουν ορισμένα εργαλεία που μπορούν να μας βοηθήσουν μερικές φορές στην προσέγγιση του σωστού αριθμού των κρυμμένων κόμβων, όπως ο κανόνας της Γεωμετρικής Πυραμίδας.

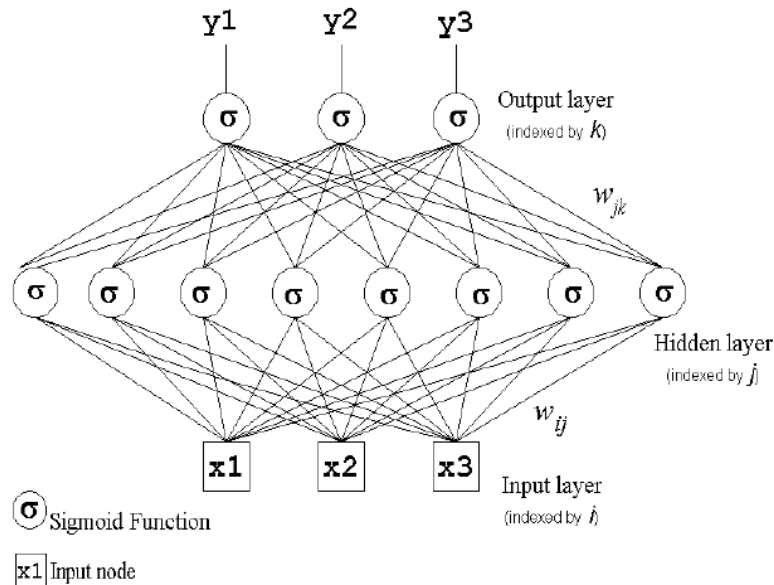
- Ο αριθμός των κόμβων εξόδου: Ο αριθμός των κόμβων εξόδου ενός νευρωνικού δικτύου είναι σημαντικός, καθώς επηρεάζει σε μεγάλο βαθμό την απόδοση του. Νευρωνικά δίκτυα με με πολλές εξόδους, ειδικά αν αυτές απέχουν αρκετά μεταξύ τους, αποδίδουν χειρότερα σε σχέση με νευρωνικά δίκτυα που έχουν μία μοναδική έξοδο.
- Η συνάρτηση ενεργοποίησης και η συνάρτηση εισαγωγής: Οι συναρτήσεις ενεργοποίησης είναι μαθηματικές φόρμουλες που καθορίζουν το αποτέλεσμα της διαδικασίας επεξεργασίας των στοιχείων που εισάγονται στο νευρώνα μέσω των συναρτήσεων εισαγωγής. Συνήθως χρησιμοποιούνται ως συναρτήσεις ενεργοποίησης μη γραμμικές συναρτήσεις όπως η λογιστική συνάρτηση και η υπερβολικά εφασπτομένη συνάρτηση στον άξονα x (\tanh).

Ο σκοπός των συναρτήσεων ενεργοποίησης είναι να αποτρέψουν τις ακραίες τιμές στα αποτελέσματα εξόδου, έτσι ώστε να μην αδρανοποιηθεί το τεχνητό νευρωνικό δίκτυο αναστέλλοντας τη διαδικασία της εκπαίδευσης του. Συναρτήσεις ενεργοποίησης όπως οι σιγμοειδείς χρησιμοποιούνται συχνά στην εκπαίδευση των τεχνητών νευρωνικών δικτύων, γιατί είναι μη γραμμικές και συνεχώς διαφοροποιήσιμες, χαρακτηριστικά που τις αναγάγουν σε ιδανικές επιλογές για χρήση σε τεχνητά νευρωνικά δίκτυα [5].

Είναι εύκολο να αντιληφθεί κανείς ότι δεν υπάρχει μόνο μία συγκεκριμένη αρχιτεκτονική όσον αφορά τα τεχνητά νευρωνικά δίκτυα, αν και όλες οι αρχιτεκτονικές περιλαμβάνουν τα παραπάνω συνθετικά μέρη. Κάθε μία αρχιτεκτονική έχει τα δικά της πλεονεκτήματα και μειονεκτήματα και διαφέρει από τις άλλες ως προς τον τρόπο που επεξεργάζεται τις πληροφορίες και παράγει αποτελέσματα. Επομένως αναλόγως τη φύση του προβλήματος που καλείται να λύσει το τεχνητό νευρωνικό δίκτυο που πρέπει να κατασκευάσουμε, πρέπει να επιλέξουμε αρχικά την κατάλληλη αρχιτεκτονική. Στη συνέχεια ακολουθούν οι πιο γνωστές αρχιτεκτονικές τεχνητών νευρωνικών δικτύων που χρησιμοποιούνται για την επίλυση μιας πληθώρας προβλημάτων η πολυπλοκότητα των οποίων αυξάνεται συνεχώς.

1.3.1. Multilayer Perceptrons (MLPs)

Ο Multilayer Perceptron (MLP) είναι ο πιο διαδεδομένος και συχνά χρησιμοποιούμενος τύπος αρχιτεκτονικής τεχνητών νευρωνικών δικτύων. Τα σήματα μεταδίδονται κατά μήκος του τεχνητού νευρωνικού δικτύου πάντα προς μία μόνο κατεύθυνση, από την είσοδο προς την έξοδο χωρίς επανάληψη. Για το λόγο αυτό η αρχιτεκτονική αυτή ονομάζεται και εναλλακτικά Feed-Forward Network (FFN). Τα βασικά πλεονεκτήματα των MLPs είναι ότι είναι εύκολοι στη χρήση και μπορούν να προσεγγίσουν οποιοδήποτε χάρτη εισόδου-εξόδου, ενώ τα μειονεκτήματά τους είναι ότι μαθαίνουν αργά και απαιτούν μεγάλο όγκο δεδομένων για την εκπαίδευση τους.



Η αρχιτεκτονική ενός Multilayer Perceptron [6].

Ο MLP χρησιμοποιεί μη γραμμικές συναρτήσεις ενεργοποίησης, με μοναδική εξαίρεση τις πολυωνυμικές συναρτήσεις, ενώ για την εκπαίδευση των μοντέλων τεχνητών νευρωνικών δικτύων που κατασκευάζονται με βάση αυτόν χρησιμοποιούνται αλγόριθμοι εκπαίδευσης όπως ο Backpropagation. Οι συναρτήσεις που έχουν επικρατήσει να χρησιμοποιούνται σήμερα ως συναρτήσεις ενεργοποίησης για τον MLP είναι η μονοπολική σιγμοειδής (λογιστική) συνάρτηση (1) και η διπολική σιγμοειδής συνάρτηση (2), οι οποίες έχουν την παρακάτω μορφή.

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

$$f(s) = \frac{1-e^{(-\alpha)s}}{1+e^{(-\alpha)s}} \quad (2)$$

Τα βασικά στάδια για το σχεδιασμό και τη λειτουργία ενός MLP τεχνητού νευρωνικού δικτύου είναι η επιλογή του αριθμού των κρυμμένων επιπέδων του δικτύου, η επιλογή του αριθμού των νευρώνων σε κάθε κρυμμένο επίπεδο, η επιλογή της βέλτιστης αλγοριθμικής λύσης που ελαχιστοποιεί τα σφάλματα, η εφαρμογή της λύσης αυτής σε ένα εύλογο χρονικό διάστημα και ο τελικός έλεγχος του δικτύου για την αποφυγή του φαινομένου της υπερπροσαρμογής (overfitting).

Αφού ακολουθηθούν τα παραπάνω βήματα, απώτερος στόχος είναι να μειωθεί όσο το δυνατό περισσότερο σφάλμα πρόβλεψης του τεχνητού νευρωνικού δικτύου με την κατάλληλη επιλογή βαρών και ορίων. Τον ρόλο αυτό των αναλαμβάνουν οι αλγόριθμοι εκπαίδευσης που επιλέγονται για να εκπαιδεύσουν το δίκτυο, χρησιμοποιώντας την ιστορικότητα των περιπτώσεων που έχουν συγκεντρώσει μέχρι στιγμής και συγκρίνοντας τα αποτελέσματά τους με συγκεκριμένες τιμές εξόδου. Οι αποκλίσεις που παρατηρούνται συγκεντρώνονται σε μία συνάρτηση υπολογισμού του σφάλματος του νευρωνικού δικτύου, όπως η συνάρτηση του αθροίσματος των τετραγωνικών σφαλμάτων των αποτελεσμάτων εξόδου.

Επίσης για την αντιμετώπιση της πλειοψηφίας των προβλημάτων που καλείται να λύσει ο MLP η χρήση ενός κρυμμένου επιπέδου είναι υπεραρκετή. Δύο κρυμμένα επίπεδα χρησιμοποιούνται συνήθως σε περιπτώσεις μοντελοποίησης δεδομένων που

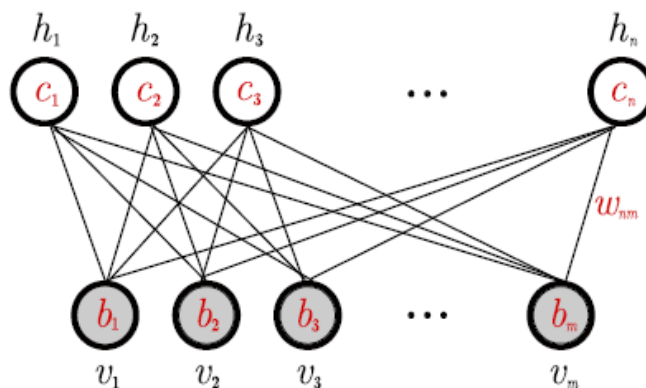
παρουσιάζουν ασυνέχεια, ενώ δεν υπάρχει θεωρητικά κανένας λόγος για τη χρήση παραπάνω από δύο κρυμμένων επιπέδων.

Τέλος η επιλογή του κατάλληλου αριθμού νευρώνων που θα υπάρχουν στα κρυμμένα επίπεδα του MLP καθορίζει σε πολύ σημαντικό βαθμό τη λειτουργία του και το τελικό αποτέλεσμα που παράγει. Επιλέγοντας πολύ μικρό αριθμό νευρώνων ρισκάρουμε να έχουμε φαινόμενα υποπροσαρμογής (underfitting) στο μοντέλο μας, αφού θα αδυνατούν να εντοπίσουν όλα τα διαθέσιμα σήματα. Αντιθέτως επιλέγοντας πολύ μεγάλο αριθμό νευρώνων ρισκάρουμε να αυξηθεί σημαντικά ο χρόνος εκπαίδευσης του δικτύου, αλλά και να έχουμε φαινόμενα υπερπροσαρμογής (overfitting), δηλαδή τα διαθέσιμα δεδομένα να μην επαρκούν για την σωστή εκπαίδευση των νευρώνων [7].

1.3.2. Restricted Boltzmann Machines (RBMs)

Restricted Boltzmann Machine (RBM) ονομάζουμε μία αρχιτεκτονική νευρωνικών δικτύων που μπορεί να ερμηνευθεί ως στοχαστικό νευρωνικό δίκτυο και είναι μία από τις δημοφιλέστερες αρχιτεκτονικές βαθιάς μάθησης, λόγω της ικανότητας της να μαθαίνει την πιθανότητα της κατανομής των τιμών εισόδου με τη χρήση των μεθόδων τόσο της εποπτευόμενης όσο και της μη εποπτευόμενης μάθησης [8].

Οι νευρώνες ενός RBM χωρίζονται στις κατηγορίες των ορατών και των κρυμμένων και σχηματίζουν τη μορφή ενός αμφίδρομου γράφου, χωρίς όμως να υπάρχει σύνδεση μεταξύ των νευρώνων που βρίσκονται στο ίδιο επίπεδο [9]. Το ορατό επίπεδο αποτελείται από τους νευρώνες που είναι άμεσα μετρήσιμοι στον RBM, δηλαδή διαδραματίζουν τον ρόλο των μονάδων εισόδου. Αντίθετα οι κρυμμένοι νευρώνες στο μη ορατό επίπεδο δεν συνδέονται άμεσα με τα δεδομένα εκπαίδευσης και ο σκοπός τους επικεντρώνεται στη μοντελοποίηση των σχέσεων εξάρτησης που παρατηρούνται μεταξύ των συνθετικών μερών του μοντέλου. Ουσιαστικά οι κρυφοί νευρώνες εξάγονται κατά κάποιον τρόπο από τους ορατούς νευρώνες, λειτουργώντας ως μη γραμμικοί ανιχνευτές [10].



Η αρχιτεκτονική ενός Restricted Boltzmann Machine με n κρυμμένες και m ορατές μεταβλητές [11].

Η συνδυασμένη κατανομή πιθανότητας ενός τυπικού RBM μπορεί να οριστεί με την κατανομή του Gibbs:

$$p(v, h) = \frac{1}{Z} e^{-E(v, h)}$$

Με την συνάρτηση ενέργειας $E(v, h)$ να έχει την παρακάτω μορφή:

$$E(v, h) = \sum_{i=1}^n \sum_{j=1}^m w_{ij} h_j v_i - \sum_{j=1}^m b_j v_j - \sum_{i=1}^n c_i h_i \quad [8].$$

Όπου τα n και m είναι οι αριθμοί των κρυμμένων και των φανερών μονάδων, τα v_i και τα h_j είναι οι σταθερές των φανερών και κρυμμένων μονάδων i και j αντίστοιχα, b_i και c_j είναι σφάλματα με κανονικές τιμές που αντιστοιχούν στην i στη φανερή και j στη κρυμμένη μονάδα και με τη μεταβλητή w_{ij} συμβολίζονται οι πραγματικές τιμές των βαρών που συνδέουν τις φανερές με τις κρυμμένες μονάδες. Το z είναι η σταθερά κανονικοποίησης, δηλαδή το άθροισμα όλων των πιθανών συνδυασμών για το $e^{-E(v,h)}$ προκειμένου να εξασφαλίσουμε ότι όλες οι κατανομές πιθανότητας αθροίζουν στο 1.

Ο περιορισμός που υφίσταται μεταξύ των στρωμάτων του RBM καθιστά τις κρυμμένες μεταβλητές του εξαρτημένες από τις σταθερές των φανερών μεταβλητών του και το αντίστροφο. Προκειμένου να διευκολύνουμε τη μοντελοποίηση της κατανομής πιθανότητας της κάθε μεταβλητής, μπορούμε να τις υπολογίσουμε εύκολα με τις παρακάτω σχέσεις [11].

$$p(h|v) = \prod_{j=1}^m p(h_j | v)$$

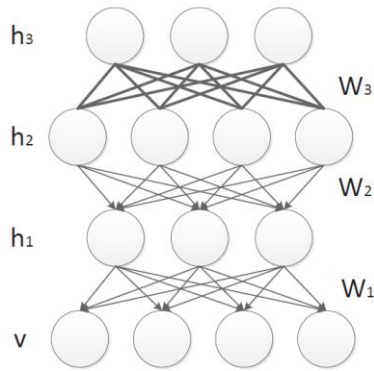
$$p(v|h) = \prod_{i=1}^n p(v_i | h)$$

1.3.3. Deep Belief Networks (DBFs)

Ως Belief Networks ονομάζουμε τα νευρωνικά δίκτυα που αποτελούνται από στρώματα στοχαστικών δυαδικών μονάδων που ενώνονται μεταξύ τους μέσω σταθμισμένων συνδέσεων. Ο σκοπός των δικτύων αυτών είναι να καταφέρουν να συμπεραίνουν την κατάσταση των στοχαστικών δυαδικών μονάδων που δεν έχουν παρατηρηθεί ακόμα προσαρμόζοντας κατάλληλα τα βάρη μεταξύ των μονάδων αυτών και λειτουργώντας με βάση τα δεδομένα που έχουν ήδη παρατηρηθεί [12].

Με το όνομα Deep Belief Networks (DBFs) αναφέρουμε την αρχιτεκτονική νευρωνικών δικτύων που αποτελείται από πολυεπίπεδα Belief Networks. Ουσιαστικά ένα DBF αποτελείται από ένα σύνολο RBMs τοποθετημένα το ένα πάνω στο άλλο, δομή που εμπνεύστηκε πρώτος ο Hinton το 2006 και αποτελεί το πρώτο μοντέλο βαθιάς μάθησης που εκπαιδεύτηκε με επιτυχία [13].

Η βασική ιδέα πίσω από αυτή την αρχιτεκτονική είναι η χρήση ανά επίπεδο μιας μεθόδου μη εποπτευόμενης μάθησης με σκοπό την προετοιμασία των αρχικών βαρών πριν την κανονική τους εκπαίδευση στο τεχνητό νευρωνικό δίκτυο [14]. Αυτό πρακτικά σημαίνει ότι τα χαρακτηριστικά του κάθε επιπέδου μεταφέρονται στο αμέσως επόμενο, ενώ όπως γίνεται εύκολα κατανοητό τα χαμηλότερα RBMs εκπαιδεύονται πρώτα από αυτά που βρίσκονται ψηλότερα στην ιεραρχική δομή του DBN. Σε σύγκριση με ένα μόνο RBM το DBN μπορεί να αυξήσει το άνω άκρο της λογαριθμικής πιθανότητας του μοντέλου, γεγονός που έχει ως αποτέλεσμα καλύτερες δυνατότητες εκπαίδευσης.



Η αρχιτεκτονική ενός Deep Belief Network , όπου με h συμβολίζονται τα κρυμμένα επίπεδα, με v τα ορατά επίπεδα και w τα βάρη [9].

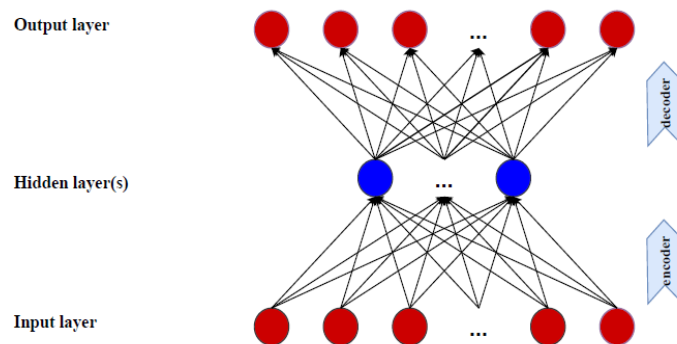
Ένα DBL με L αριθμό κρυμμένων επιπέδων έχει κοινή συνάρτηση κατανομής μεταξύ των ορατών επιπέδων του v και των κρυμμένων του επιπέδων h^l με l = 1, 2, 3,L της παρακάτω μορφής:

$$p(v, h^1, \dots, h^L) = p(v|h^1) \prod_{l=1}^{L-2} p(h^l|h^{l+1})p(h^{L-1}, h^L) \quad [8].$$

1.3.4. Autoencoders

Με τον όρο Autoencoder ονομάζεται μία συγκεκριμένη αρχιτεκτονική τεχνητών νευρωνικών δικτύων, η οποία έχει ως πυρήνα της την προσπάθεια των δεδομένων εξόδου να προσεγγίσουν τα δεδομένα εισόδου. Ένας τυπικός Autoencoder αποτελείται συνήθως από τρία επίπεδα, με το επίπεδο εισόδου να έχει τον ίδιο αριθμό νευρώνων με το επίπεδο εξόδου. Υπάρχουν δύο βασικά είδη Autoencoders οι γραμμικοί Autoencoders και οι μη γραμμικοί Autoencoders αναλόγως με το είδος της συνάρτησης ενεργοποίησης που χρησιμοποιούν [15].

Οι μεταβλητές εισόδου περνούν μέσα από έναν αριθμό νευρώνων στο κρυμμένο επίπεδο, ο οποίος είναι μικρότερος σε σχέση με τα άλλα δύο επίπεδα, προσπαθώντας να δημιουργήσουν εκεί μία συμπιεσμένη αναπαράστασή τους (encoding) και η οποία στη συνέχεια αποσυμπιέζεται και χαρτογραφείται στο επίπεδο εξόδου (decoding).



Η αρχιτεκτονική ενός τυπικού Autoencoder με ένα κρυμμένο επίπεδο , όπου τα επίπεδα εισόδου και εξόδου είναι πανομοιότυπα [16].

Λόγω του γεγονότος ότι δεν χρησιμοποιούνται στη συγκεκριμένη διεργασία άλλες μεταβλητές εκτός από τα δεδομένα εισόδου συμπεραίνουμε ότι αυτή η αρχιτεκτονική εφαρμόζει τη μέθοδο της μη εποπτευόμενης μάθησης [16].

Η διαδικασία κωδικοποίησης (encoding) και αποκωδικοποίησης (decoding) ενός Autoencoder μπορεί να περιγραφεί από τις παρακάτω εξισώσεις:

Κατά την κωδικοποίηση:

$$y' = f(w \cdot x + b)$$

Όπου w και b αντιστοιχούν στις παραμέτρους που ρυθμίζονται, f είναι η συνάρτηση ενεργοποίησης, x είναι το διάνυσμα εισόδου και y είναι η κρυμμένη αναπαράσταση.

Κατά την αποκωδικοποίηση,

$$x' = f(w' \cdot y' + c)$$

Όπου w' είναι ο ανάστροφος πίνακας του w , το c το στατιστικό σφάλμα του επιπέδου εξόδου και το x' είναι η ανασχηματισμένη είσοδος στο επίπεδο εξόδου. Οι παράμετροι του Autoencoder μπορούν να αναβαθμιστούν με τις παρακάτω σχέσεις:

$$w_{new} = w - \frac{\eta \theta E}{\theta w}$$

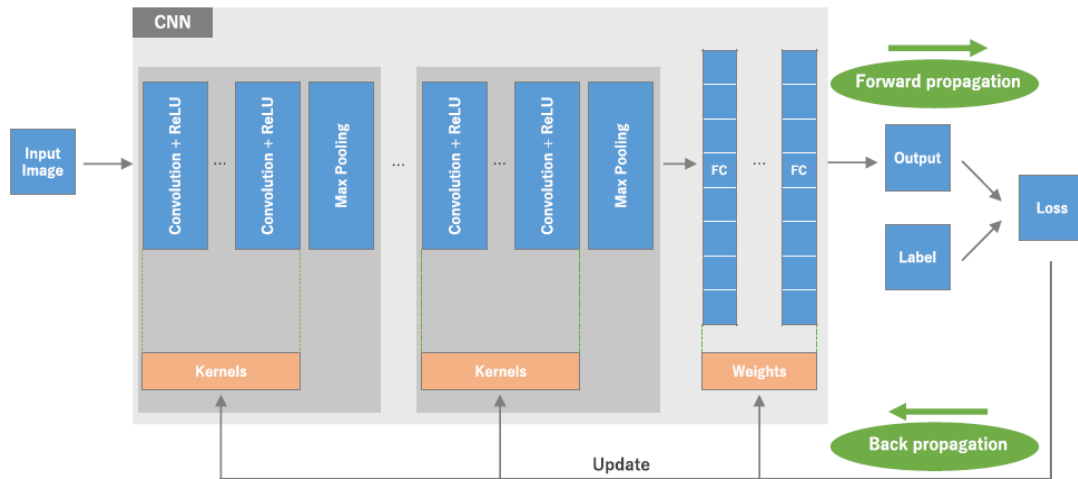
$$b_{new} = b - \frac{\eta \theta E}{\theta b}$$

Όπου w_{new} και b_{new} είναι οι αναβαθμισμένες παράμετροι για τα w και b αντίστοιχα στο τέλος της τρέχουσας επανάληψης και E είναι το σφάλμα ανασχηματισμού των στοιχείων εισόδου στο επίπεδο εξόδου [8].

1.3.5. Convolutional Neural Networks (CNNs)

Το Convolutional Neural Network (CNN) είναι μία από τις δημοφιλέστερες και ευρεία χρησιμοποιημένες αρχιτεκτονικές τεχνητών νευρωνικών δικτύων, ειδικά σε προβλήματα που σχετίζονται με την αναγνώριση και ανάλυση εικόνων. Η ονομασία του προέρχεται από τη συνέλιξη (convolution) κι η οποία είναι μία μαθηματική γραμμική διεργασία μεταξύ πινάκων [17]. Η πηγή έμπνευσης για τη συγκεκριμένη αρχιτεκτονική ήταν ο φυσικός οπτικός μηχανισμός των ζωντανών πλασμάτων, μέσω του οποίου ανιχνεύουν το φως στα δεκτικά πεδία τους με τη χρήση συγκεκριμένων κυττάρων στον οπτικό τους μανδύα [18].

Παρόλο που υπάρχουν πάρα πολλές διαφορετικές αρχιτεκτονικές CNN, τα βασικά τους γνώρισμα είναι πανομοιότυπα. Ένα τυπικό CNN αποτελείται από μία στοιβάδα διαφορετικών επιπέδων όπως τα συνελκτικά επίπεδα (convolutional layers), τα συγκεντρωτικά επίπεδα (pooling layers) και τα πλήρως συνδεδεμένα επίπεδα (fully connected layers). Η απόδοση ενός CNN μοντέλου, που λειτουργεί υπό συγκεκριμένους πυρήνες και βάρη, υπολογίζεται με τη χρήση μιας συνάρτησης απώλειας μέσω της μεθόδου forward propagation σε ένα σύνολο δεδομένων εκπαίδευσης, ενώ οι παράμετροι που εκπαιδεύονται όπως οι πυρήνες και τα βάρη ενημερώνονται σύμφωνα με την τιμή απώλειας μέσω της μεθόδου Backpropagation και του αλγόριθμου βελτιστοποίησης Gradient Descent.



Η αρχιτεκτονική ενός τυπικού CNN με convolutional layers, pooling layers, fully connected layers (FC) και rectified linear units (ReLU) [19].

Το συνελικτικό επίπεδο (convolution layer) είναι η καρδιά της αρχιτεκτονικής ενός CNN, αφού είναι το επίπεδο στο οποίο διεξάγεται η εξαγωγή των χαρακτηριστικών των δεδομένων εισόδου και αποτελείται από ένα συνδυασμό γραμμικών και μη γραμμικών διεργασιών όπως η συνέλιξη και η συνάρτηση ενεργοποίησης.

Το συγκεντρωτικό επίπεδο (pooling layer) είναι το επίπεδο που διεξάγεται η δειγματοληψία κατά την οποία μειώνεται η διαστασιοποίηση των ανά επίπεδο χαρτογραφημένων χαρακτηριστικών (feature maps) με απώτερο σκοπό τη μείωση των εκπαιδευόμενων παραμέτρων που ακολουθούν. Αξίζει να σημειώσουμε ότι στο συγκεντρωτικό επίπεδο (pooling layer) δεν υπάρχουν παράμετροι προς εκπαίδευση και τον αντίστοιχο ρόλο τον διαδραματίζουν υπερπαραμέτροι όπως το μέγεθος του φιλτραρίσματος και το βήμα.

Τα τελικά χαρτογραφημένα χαρακτηριστικά μετατρέπονται σε ένα μονοδιάστατο πίνακα αριθμών και συνδέονται σε ένα ή περισσότερα πλήρως συνδεδεμένα επίπεδα (fully-connected layers), στα οποία η κάθε είσοδος είναι συνδεδεμένη με κάθε έξοδο μέσω ενός εκπαιδευόμενου βάρους. Στη συνέχεια τα χαρακτηριστικά που θα εξαχθούν από τα συγκεντρωτικά επίπεδα (convolutional layers) και θα δειγματοληφθούν από τα συγκεντρωτικά επίπεδα (pooling layers), θα χαρτογραφηθούν από ένα υποσύνολο των πλήρως συνδεδεμένων επιπέδων (fully connected layers) στις τελικές εξόδους του δικτύου. Ένα παράδειγμα αυτής της διαδικασίας είναι οι πιθανότητες της κάθε κλάσης σε προβλήματα ταξινόμησης. Το τελευταίο πλήρως συνδεδεμένο επίπεδο (fully connected layer) συνήθως έχει τον ίδιο αριθμό κόμβων εξόδου με των αριθμό των κλάσεων, ενώ κάθε ένα από αυτά ακολουθείται συνήθως από μία μη γραμμική συνάρτηση όπως τα ReLU [19].

Η σχέση που υπολογίζει μαθηματικά την αξία ενός χαρακτηριστικού στη θέση (i,j) στο k -στο χαρτογραφημένο χαρακτηριστικό του l -στου επιπέδου, $z_{i,j,k}^l$, είναι η παρακάτω:

$$z_{i,j,k}^l = w_k^l \cdot x_{i,j}^l + b_k^l$$

Με τα w_k^l και b_k^l να συμβολίζουν το διανυσματικό βάρος και το σφάλμα του k -στου φίλτρου στο l -στο επίπεδο και με το $x_{i,j}^l$ την παρτίδα των δεδομένων εισόδου που τοποθετημένα στη θέση (i,j) του l -στου επιπέδου.

Η αξία ενεργοποίησης $a_{i,j,k}^l$ του συνελκτικού (convolutional) χαρακτηριστικού $z_{i,j,k}^l$ μπορεί να υπολογιστεί με την παρακάτω μαθηματική σχέση:

$$a_{i,j,k}^l = a(z_{i,j,k}^l)$$

Με τις συνηθέστερες συναρτήσεις ενεργοποίησης να είναι οι σιγμοειδείς, tanh και ReLU. Η μείωση της διαστασιοποίησης που διεξάγεται στο συγκεντρωτικό επίπεδο (pooling layer) υπολογίζεται με την ενεργοποίηση της συγκεντρωτικής συνάρτησης pool (.) για κάθε χαρτογραφημένο χαρακτηριστικό $a_{i,j,k}^l$ και έχει την παρακάτω μορφή:

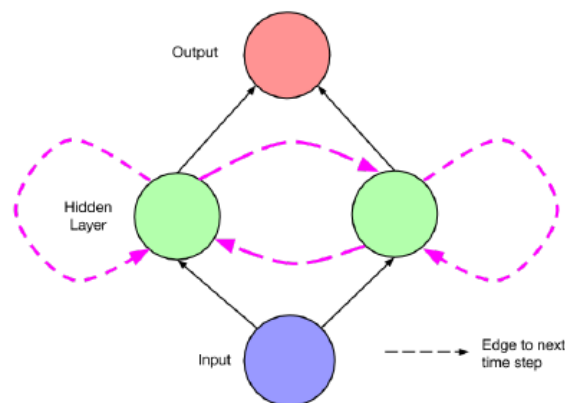
$$y_{i,j,k}^l = pool(a_{m,n,k}^l), \forall (m, n) \in R_{ij}$$

Με το R_{ij} να συμβολίζει την γειτονικό χώρο γύρω από τη θέση (i,j). Τέλος αν υποθέσουμε ότι έχουμε N επιθυμητούς συσχετισμούς εισόδου-εξόδου $\{(x^{(n)}, y^{(n)}); n [1, \dots, N]\}$, με $x^{(n)}$ να συμβολίζεται το n-στο δεδομένο εισόδου, με $y^{(n)}$ να συμβολίζεται η αντίστοιχη τιμή στόχου, με $o^{(n)}$ να συμβολίζεται το αποτέλεσμα εξόδου και με το θ να ενεργοποιούνται όλες οι παράμετροι του CNN (βάρη, σφάλματα κλπ), τότε η συνάρτηση απώλειας του CNN μπορεί να υπολογιστεί ακολούθως [18] :

$$L = \frac{1}{N} \sum_{n=1}^N l(\theta; y^{(n)}, o^{(n)})$$

1.3.6. Recurrent Neural Networks (RNNs)

Τα Recurrent Neural Networks(RNN) ανήκουν στην οικογένεια των Feed-Forward Neural Networks και έχουν ως κύριο γνώρισμα τους τη δυνατότητα να επεκτείνονται σε γειτονικά χρονικά στάδια. Με αυτό τον τρόπο ένα κόμβος του δικτύου έχει τη δυνατότητα σε κάθε χρονική στιγμή να παίρνει τα τρέχοντα δεδομένα εισόδου, παράλληλα με τις τιμές των κρυμμένων κόμβων, συλλέγοντας έτσι πληροφορίες από προηγούμενες χρονικές στιγμές [8].



Η αρχιτεκτονική ενός τυπικού RNN. Σε κάθε χρονικό βήμα t η ενεργοποίηση προωθείται κατά μήκος των στερεών άκρων όπως σε ένα feedforward δίκτυο. Οι διακεκομμένες άκρες συνδέουν μία πηγή κόμβο τη χρονική στιγμή t σε ένα κόμβο στόχο σε κάθε επόμενη χρονική στιγμή $t + 1$ [20].

Τα RNNs ενδέχεται να μη σχηματίζουν κύκλους μεταξύ των συμβατικών τους άκρων. Ωστόσο τα άκρα τους που συνδέουν παρακείμενα χρονικά βήματα, ονομάζονται επαναλαμβανόμενα άκρα και μπορούν να σχηματίζουν κύκλους. Σε αυτούς τους κύκλους συμπεριλαμβάνονται και οι κύκλοι με μοναδιαίο μήκος, οι οποίοι είναι ουσιαστικά αυτοσυνδέσεις ενός κόμβου με τον εαυτό του κατά τη διάρκεια του χρόνου.

Η λειτουργία ενός τυπικού RNN μπορεί να περιγραφεί ακολούθως: Τη χρονική στιγμή t κόμβοι με επαναλαμβανόμενες άκρες λαμβάνουν δεδομένα εισόδου από τη τρέχουσα μονάδα δεδομένου (data point) $x^{(t)}$ καθώς επίσης και τιμές $h^{(t-1)}$ από τον κρυμμένο κόμβο, που αφορούν την κατάσταση του δικτύου την αμέσως προηγούμενη στιγμή.

Το αποτέλεσμα $y^{(t)}$ τη χρονική στιγμή t υπολογίζεται με βάσει τις δοθείσες τιμές $h^{(t)}$ του κρυμμένου κόμβου τη χρονική στιγμή t . Οι εισροές $x^{(t-1)}$ τη χρονική στιγμή $t-1$ μπορούν να επηρεάσουν το αποτέλεσμα $y^{(t)}$ από τη χρονική στιγμή t και έπειτα με την πάροδο των επαναλαμβανόμενων συνδέσεων.

Δύο εξισώσεις καθορίζουν όλους του υπολογισμούς που απαιτούνται για τη στάθμιση σε κάθε χρονικό βήμα του περάσματος προς τα εμπρός ενός τυπικού RNN όπως το παραπάνω.

$$h^{(t)} = \sigma(W^{hx} \cdot x^{(t)} + W^{hh}h^{(t-1)} + b_h)$$

$$\widehat{y^{(t)}} = \text{softmax}(W^{yh}h^{(t)} + b_y)$$

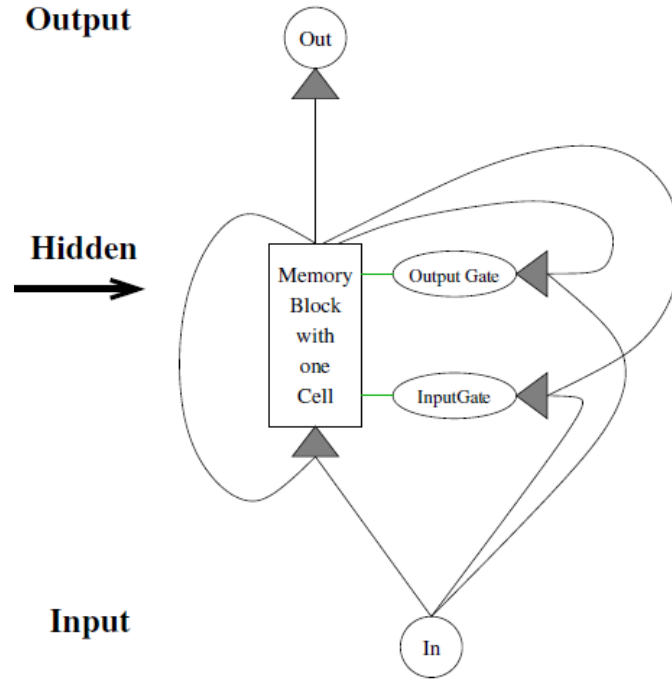
Η μεταβλητή W^{hx} συμβολίζει τον πίνακα των συμβατικών βαρών μεταξύ των εισροών και του κρυμμένου επιπέδου και η μεταβλητή W^{hh} συμβολίζει τον πίνακα των επαναλαμβανόμενων βαρών μεταξύ του κρυμμένου επιπέδου και του εαυτού του σε παρακείμενα χρονικά βήματα. Τέλος τα διανύσματα b_h και b_y είναι παράμετροι των σφαλμάτων που επιτρέπουν σε κάθε κόμβο να μάθει μία αντιστάθμιση [20].

1.3.7. Long-Short Term Memory Neural Networks (LSTMNs)

Το Long-Short Term Memory τεχνητό νευρωνικό δίκτυο (LSTMN) είναι ένας συγκεκριμένος τύπος Recurrent Neural Network, το οποίο δημιουργήθηκε με σκοπό να μοντελοποιεί τις χρονικές αλληλουχίες και τις μεγάλες εύρους εξαρτήσεις αυτών με μεγαλύτερη ακρίβεια από τα συμβατικά RNNs [21]. Η βασική μονάδα στο κρυμμένο επίπεδο ενός LSTMN τεχνητού νευρωνικού δικτύου είναι το μπλοκ μνήμης (memory block), το οποίο αντικαθιστά τις κρυφές μονάδες των κοινών RNN. Ένα μπλοκ μνήμης περιέχει ένα ή περισσότερα κελιά μνήμης (memory cells) και ένα ζευγάρι από προσαρμοστικές, πολλαπλασιαστικές μονάδες πύλης, οι οποίες χορηγούν είσοδο στις εισροές και εκροές όλων των κελιών του μπλοκ. Τα μπλοκ μνήμης επιτρέπουν στα κελιά να μοιράζονται τις ίδιες πύλες, έτσι ώστε να μειώνεται ο αριθμός των προσαρμοστικών παραμέτρων.

Κάθε κελί μνήμης έχει στον πυρήνα του μία επανειλημμένα αυτοσυνδεδεμένη γραμμική μονάδα με το όνομα Constant Error Carousel(CEC), η ενεργοποίηση της οποίας ονομάζεται κατάσταση κελιού (cell state). Το CEC λύνει το πρόβλημα της κατάρτησης σφάλματος με τον εξής τρόπο: όταν υπάρχει απουσία νέων εισροών ή σημάτων σφάλματος στο κελί, τότε η τοπική οπίσθια ροή σφάλματος του CEC παραμένει σταθερή. Το CEC προστατεύεται ταυτόχρονα από την εμπρόσθια ροή ενεργοποίησης και την οπίσθια ροή σφάλματος, χάριν της συμβολής των πυλών εισόδου και εξόδου αντίστοιχα. Όταν οι πύλες είναι κλειστές (ενεργοποίηση γύρω στο

μηδέν) άσχετες εισροές και θόρυβος δεν μπαίνουν στο κελί, με αποτέλεσμα να μη διαταράσσεται το υπόλοιπο δίκτυο [22].



Η αρχιτεκτονική ενός τυπικού LSTMN δικτύου με μπλοκ μνήμης στο κρυμμένο επίπεδο (ένα μόνο είναι διακριτό) [22].

Ένα LSTMN τεχνητό νευρωνικό δίκτυο λογαριάζει μία αντιστοίχιση από μία ακολουθία εισόδου $x = (x_1, x_2, \dots, x_T)$ σε μία ακολουθία εξόδου $y = (y_1, y_2, \dots, y_T)$, υπολογίζοντας τις μονάδες ενεργοποίησης του δικτύου χρησιμοποιώντας τις παρακάτω εξισώσεις επαναληπτικά από το $t=1$ μέχρι το T :

$$i_t = \sigma(W_{ix} \cdot x_t + W_{im} \cdot m_{t-1} + W_{ic} \cdot c_{t-1} + b_i)$$

$$f_t = \sigma(W_{fx} \cdot x_t + W_{fm} \cdot m_{t-1} + W_{fc} \cdot c_{t-1} + b_f)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g(W_{cm} x_t + W_{cm} m_{t-1} + b_c)$$

$$o_t = \sigma(W_{ox} \cdot x_t + W_{om} \cdot m_{t-1} + W_{oc} \cdot c_t + b_o)$$

$$m_t = o_t \odot h(c_t)$$

$$y_t = \varphi(W_{ym} \cdot m_t + b_y)$$

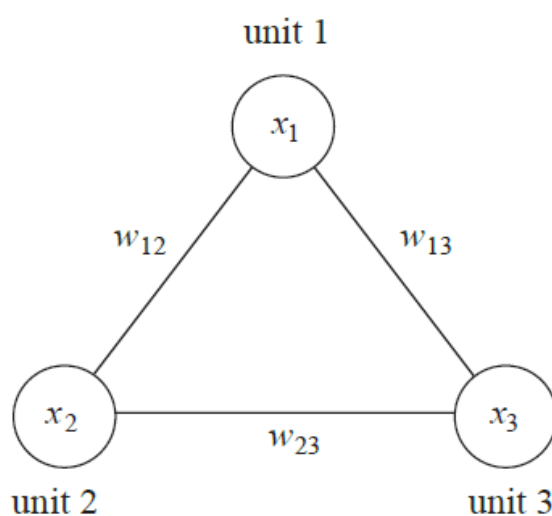
Όπου οι W όροι ενεργοποιούν τα βάρη των πινάκων (π.χ. W_{ix} είναι ο πίνακας των βαρών από την πύλη εισόδου στην είσοδο), τα W_{ic} , W_{fc} , W_{oc} είναι διαγώνιοι πίνακες βαρών για τις περιστροφές, οι b όροι ενεργοποιούν τα διανυσματικά σφάλματα (π.χ. b_i είναι το διανυσματικό σφάλμα της πύλης εισόδου), σ είναι η λογιστική σιγμοειδής

συνάρτηση και τα i, f, o, c είναι τα διανύσματα για την πύλη εισόδου, την πύλη απώλειας μνήμης, την πύλη εξόδου και ενεργοποίησης κελιού αντίστοιχα, τα οποία όλα έχουν το ίδιο μέγεθος με το διάνυσμα ενεργοποίησης κελιού m , με \odot συμβολίζεται το element-wise προϊόν των διανυσμάτων, το g και το h είναι οι συναρτήσεις ενεργοποίησης εισόδου και εξόδου των κελιών και φ είναι η συνάρτηση ενεργοποίησης εξόδου του δικτύου [21].

1.3.8. Hopfield Neural Networks (HNNs)

Η ικανότητα δικτύων υψηλής διασύνδεσης απλών μη γραμμικών αναλογικών επεξεργαστών (νευρώνων) για την επίλυση περίπλοκων προβλημάτων βελτιστοποίησης, επιδείχθηκε σε μία σειρά από δημοσιεύσεις από τον Hopfield το 1984 και τον Tank το 1986 [23]. Ένα Hopfield Neural Network (HNN) αποτελείται από n πλήρως συνδεδεμένους νευρώνες, οι οποίοι όμως δεν είναι αυτοσυνδεδεμένοι. Αυτό σημαίνει ότι κάθε νευρώνας είναι συνδεδεμένος σε όλους τους υπόλοιπους $n-1$ νευρώνες, αλλά όχι με τον εαυτό του. Ακόμη υποθέτουμε ότι οι ανεξάρτητες μονάδες διατηρούν τις ανεξάρτητες φάσεις τους μέχρι να επιλεγούν για μία νέα επικαιροποίηση.

Επίσης αυτό το τεχνητό νευρωνικό δίκτυο έχει συμμετρικά βάρη όπως το w_{ij} , που συμβολίζει την ενδοσύνδεση μεταξύ νευρώνα i και νευρώνα j και το οποίο είναι ίσο με το w_{ji} που συμβολίζει την ενδοσύνδεση μεταξύ νευρώνα j και νευρώνα i . Η απουσία αυτοσύνδεσης στο τεχνητό νευρωνικό δίκτυο έχει ως αποτέλεσμα να μην υπάρχει μόνιμη ανατροφοδότηση των νευρώνων με τις ίδιες τους τις τιμές φάσης [24].



Η αρχιτεκτονική ενός τυπικού HNN δικτύου που αποτελείται από τρεις μονάδες [25].

Οι συνδέσεις σε ένα Hopfield τεχνητό νευρωνικό δίκτυο το οποίο έχει n μονάδες μπορούν να αναπαρασταθούν με τη χρήση ενός πίνακα των βαρών $W = \{ w_{ij} \}$ διαστάσεων $n \times n$ με μηδενική διαγώνιο. Η συμμετρία του πίνακα βαρών και η μηδενική διαγώνιος είναι απαραίτητες συνθήκες για τη σύγκλιση ενός ασύγχρονου πλήρους συνδεδεμένου δικτύου σε μία σταθερή κατάσταση. Στις μονάδες ενός HNN μπορεί να εκχωρηθεί μία τιμή κατώφλι (threshold) θ , η οποία είναι διάφορη του μηδενός.

Σε αυτή την περίπτωση κάθε μονάδα του δικτύου που επιλέγεται για επικαιροποίηση υιοθετεί τη φάση 1 αν η διέγερση που υφίσταται είναι μεγαλύτερη του θ , αλλιώς υιοθετεί την φάση -1. Αυτός αποτελεί ουσιαστικά τον κανόνα ενεργοποίησης για του perceptrons, επομένως μπορούμε να οδηγηθούμε στο συμπέρασμα ότι το HNN είναι και λειτουργεί ως ένα ασύγχρονο επαναλαμβανόμενο δίκτυο από perceptrons.

Η συνάρτηση ενέργειας ενός HNN αποτελούμενο από μονάδες με κατώφλια (thresholds) θ μεγαλύτερα του μηδενός μπορεί να υπολογιστεί όπως παρακάτω:

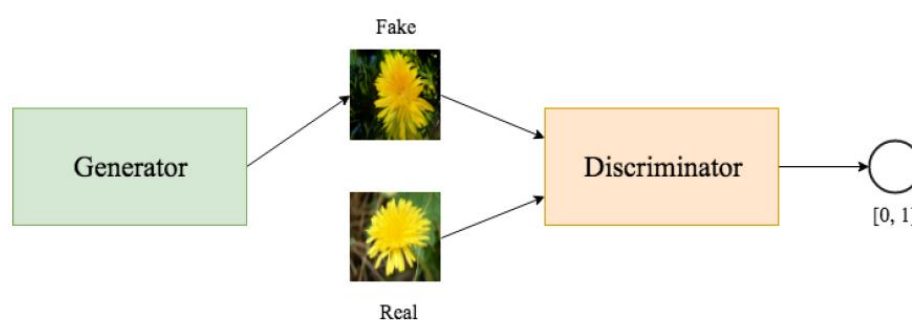
$$E(x) = -\frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n w_{ij} x_i x_j + \sum_{i=1}^n \theta_i x_i$$

Με τη μεταβλητή x να συμβολίζει τη φάση του νευρωνικού δικτύου μία συγκεκριμένη χρονική στιγμή [25].

1.3.9. Generative Adversarial Networks (GANs)

Τα Generative Adversarial Networks (GANs) είναι μία αρχιτεκτονική βαθιάς μάθησης, κύριο χαρακτηριστικό της οποίας είναι ότι αποτελείται από δύο μοντέλα. Τα δύο αυτά μοντέλα είναι το γενετικό (generative) μοντέλο G και το μεροληπτικό (discriminative) μοντέλο D , τα οποία εκπαιδεύονται ταυτόχρονα [26].

Ο στόχος της συγκεκριμένης αρχιτεκτονικής, την οποία παρουσίασε πρώτος ο Goodfellow, είναι να εκπαιδευτεί το γενετικό μοντέλο με τέτοιο τρόπο ώστε να μεγιστοποιήσει την πιθανότητα να υποπέσει σε λάθος το μεροληπτικό μοντέλο. Μία μοναδική λύση μπορεί να επιτευχθεί στο χώρο λειτουργίας όπου το γενετικό μοντέλο ανακτά την κατανομή των δεδομένων εκπαίδευσης και το μεροληπτικό μοντέλο δίνει αποτελέσματα με πιθανότητα 50% για κάθε δείγμα.



Μία αφαιρετική απεικόνιση της αρχιτεκτονικής ενός GAN. Η γεννήτρια του γενετικού μοντέλου παράγει σύνθετες εικόνες, ενώ ο διευκρινιστής του μεροληπτικού μοντέλου παίρνει τις εικόνες σαν δεδομένα εισόδου και εξάγει την πιθανότητα που αποδίδει στην εικόνα ως προς το να είναι πραγματική [27].

Η διαδικασία αυτή μπορεί να παρομοιαστεί ως ένα παιχνίδι μέγιστου-ελάχιστου μεταξύ των δύο μοντέλων, καθώς τα γενετικά μοντέλα παράγουν αντιφατικά παραδείγματα, ενώ τα μεροληπτικά μοντέλα προσπαθούν να αναγνωρίσουν σωστά. Απώτερος σκοπός και των δύο μοντέλων είναι να βελτιώσουν την αποτελεσματικότητά τους, έτσι ώστε τα αντιφατικά παραδείγματα να ταυτίζονται με τα αρχικά παραδείγματα [27].

1.3.10. Gated Recurrent Units (GRUs)

Ως Gated Recurrent Unit (GRU) ονομάζουμε την αρχιτεκτονική τεχνητών νευρωνικών δικτύων που παρουσιάστηκε για πρώτη φορά το 2014 από τον Cho και το κύριο χαρακτηριστικό της είναι η δυνατότητα της να προσαρμόζει την κάθε μονάδα ενός επαναλαμβανόμενου (recurrent) νευρωνικού δικτύου, με στόχο την απαθανάτιση των εξαρτήσεων που παρατηρούνται σε διαφορετικές χρονικές περιόδους [28].

Όπως και ο LSTMN σχεδιάστηκε με σκοπό να επαναφέρει ή να αναβαθμίζει το περιεχόμενο της μνήμης του. Κάθε GRU έχει μία πύλη επαναφοράς r_t^j και μία πύλη αναβάθμισης z_t^j , οι οποίες αντιστοιχούν στις πύλες μνήμης και εισόδου ενός LSTMN. Σε αντίθεση όμως με το LSTMN το GRU εκθέτει πλήρως το περιεχόμενο της μνήμης του σε κάθε χρονικό σημείο και ισορροπεί μεταξύ του προηγούμενου περιεχομένου της μνήμης του και του νέου περιεχομένου της μνήμης του χρησιμοποιώντας αυστηρά ολοκλήρωση διαρροής (leaky integration), παρόλο που η προσαρμοστική χρονική σταθερά του ελέγχεται από την πύλη αναβάθμισης z_t^j .

Στο χρονικό σημείο t , η σταθερά h_t^j του j -στού GRU υπολογίζεται με την παρακάτω σχέση:

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j\bar{h}_t^j$$

Όπου τα h_{t-1}^j και h_t^j αντικατοπτρίζουν το προηγούμενο περιεχόμενο της μνήμης και το νέο υποψήφιο περιεχόμενο μνήμης αντίστοιχα. Η πύλη αναβάθμισης z_t^j ελέγχει πόση από την προηγούμενη μνήμη θα απολεσθεί και πόση από τη νέα μνήμη θα προστεθεί στο δίκτυο. Η πύλη αναβάθμισης υπολογίζεται με βάσει τις προηγούμενες καταστάσεις h_{t-1}^j και την τρέχουσα μεταβλητή εισόδου x_t σύμφωνα με τον παρακάτω τύπο:

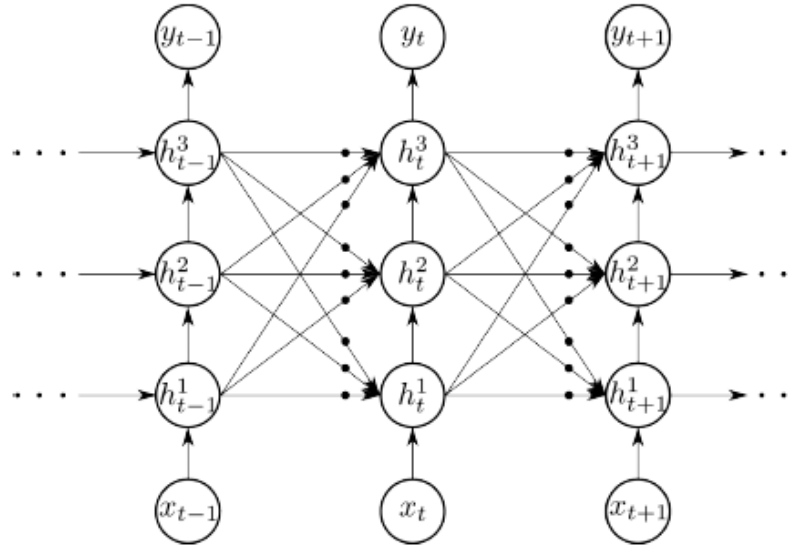
$$z_t^j = \sigma(W_z x_t + U_z h_{t-1}^j)$$

Ενώ το περιεχόμενο της νέα μνήμης h_t^j υπολογίζεται με την παρακάτω συνάρτηση

$$\bar{h}_t^j = \tanh(W_{xt} + r_t^j \odot U h_{t-1}^j)$$

Με το \odot να συμβολίζει έναν element-wise πολλαπλασιασμό και το \tanh την υπερβολική συνάρτηση.

Ένα από τα βασικά χαρακτηριστικά του GRU είναι ότι οι καταστάσεις του προηγούμενου βήματος h_{t-1}^j ρυθμίζονται από τις πύλες επαναφοράς r_t^j , με αποτέλεσμα να μπορεί να



Μία απεικόνιση της αρχιτεκτονικής ενός Gated Recurrent Neural Network [29].

αγνοήσει τις προηγούμενες κρυφές καταστάσεις, όποτε αυτό φαντάζει απαραίτητο, λαμβάνοντας υπόψιν τις προηγούμενες κρυφές καταστάσεις και την τρέχουσα είσοδο:

$$r_t = \sigma(W_r x_t + U_r h_{t-1})$$

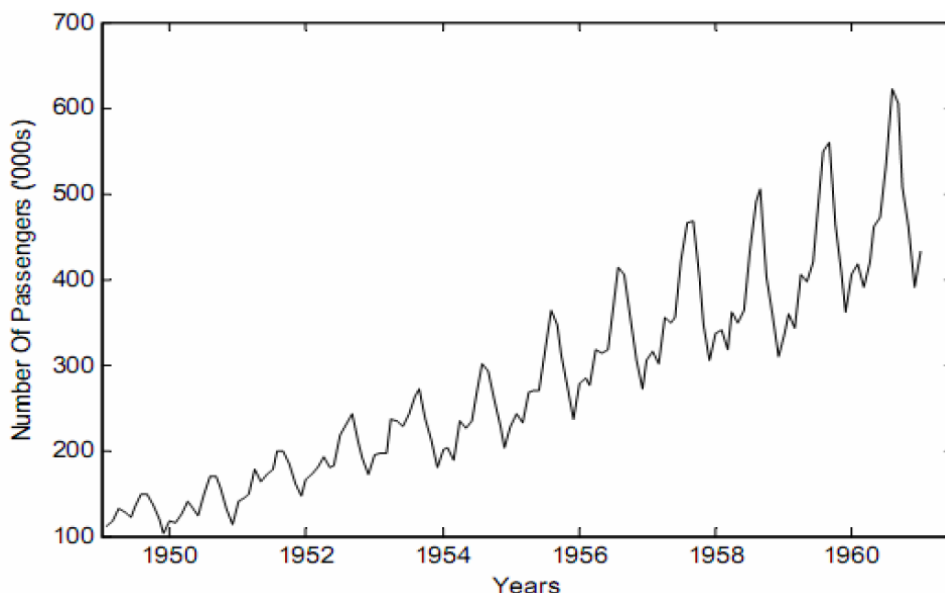
Αυτός ο μηχανισμός αναβάθμισης δίνει στο GRU τη δυνατότητα να αιχμαλωτίζει μακροχρόνιες εξαρτήσεις. Όποτε ένα προηγούμενα ανιχνευμένο χαρακτηριστικό ή περιεχόμενο μνήμης θεωρηθεί σημαντικό για μελλοντική χρήση από το τεχνητό νευρωνικό δίκτυο, η πύλη αναβάθμισης θα κλείσει μεταφέροντας το τρέχον περιεχόμενο της μνήμης διαμέσου πολλαπλών χρονικών βημάτων. Τέλος ο μηχανισμός επαναφοράς (reset) βοηθάει το GRU να χρησιμοποιεί τη χωρητικότητα του με αποδοτικό τρόπο, δίνοντας του τη δυνατότητα να επανέρχεται στην αρχική του κατάσταση όταν το ανιχνευμένο χαρακτηριστικό δεν είναι πλέον απαραίτητο [29].

ΜΕΡΟΣ 2: ΧΡΟΝΟΣΕΙΡΕΣ

2.1. Εισαγωγή στις Χρονοσειρές

Με τον όρο χρονοσειρά αναφερόμαστε σε ένα σύνολο παρατηρήσεων x_t , κάθε μία από τις οποίες καταγράφεται σε μία συγκεκριμένη στιγμή t . Χρονοσειρές διακριτού χρόνου ονομάζονται εκείνες οι χρονοσειρές στις οποίες το σύνολο του χρόνου T_0 στο οποίο γίνονται οι παρατηρήσεις είναι ένα διακριτό σύνολο, όπως για παράδειγμα η τιμή μιας μετοχής ανά ημέρα, που οι παρατηρήσεις καταγράφονται σε προκαθορισμένα χρονικά διαστήματα. Αντίθετα συνεχείς χρονοσειρές ονομάζονται εκείνες που δημιουργούνται από τη συνεχή καταγραφή παρατηρήσεων πάνω σε κάποιο χρονικό διάστημα, όπως για παράδειγμα η συνεχόμενη καταγραφή της θερμοκρασίας [30]. Όταν μία χρονοσειρά περιλαμβάνει καταγραφές μόνο μίας μεταβλητής ονομάζεται univariate, ενώ όταν αποτελείται από τις καταγραφές περισσότερων μεταβλητών ονομάζεται multivariate.

Η μεταβλητή που παρατηρείται σε μία διακριτή χρονοσειρά υποτίθεται ότι υπολογίζεται ως μία συνεχής μεταβλητή με τη χρήση της πραγματικής αριθμητικής της κλίμακας. Ακόμη μία συνεχής χρονοσειρά μπορεί να μετασχηματιστεί πολύ εύκολα σε μια διακριτή χρονοσειρά, με τη συγχώνευση των δεδομένων για συγκεκριμένο χρονικό διάστημα [31].



Μία μηνιαία απεικόνιση των επιβατών των διεθνών αερογραμμών μεταξύ Ιανουαρίου 1949 και Δεκεμβρίου 1960 [31].

Ο συνηθέστερος τρόπος για να αναπαραστήσουμε μία χρονοσειρά είναι με τη χρήση γραφημάτων, στα οποία οι παρατηρήσεις αναπαράγονται διαγραμματικά σε συνάρτηση με το χρόνο. Ένα κλασικό τέτοιο παράδειγμα αποτελεί το παραπάνω διάγραμμα.

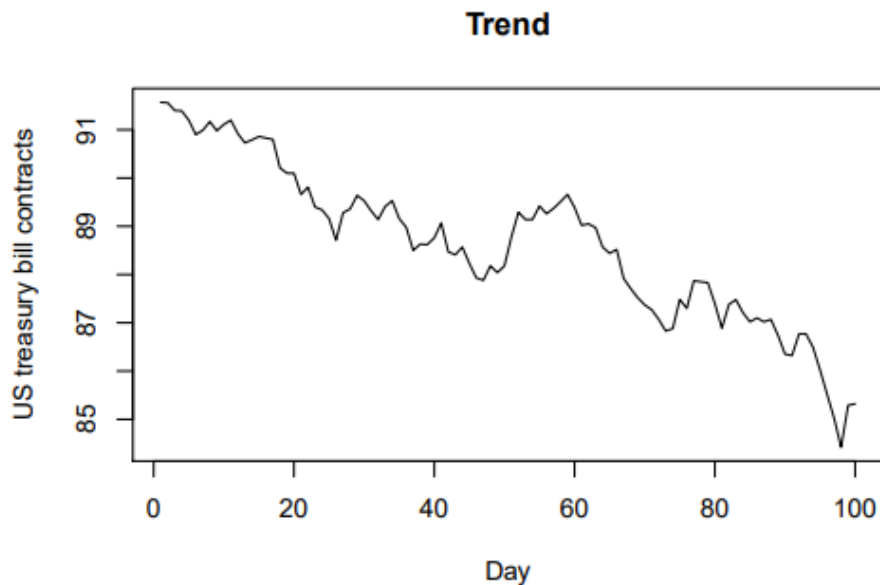
2.2. Βασικά Χαρακτηριστικά Χρονοσειρών

Τα χαρακτηριστικά ή συστατικά μιας χρονοσειράς ονομάζονται οι διαφορετικοί τύποι των διακυμάνσεων που τη συνθέτουν. Τα βασικά χαρακτηριστικά των χρονοσειρών είναι η τάση (trend), η κυκλικότητα (cyclic), η εποχικότητα (seasonal) και οι ακραίες τιμές (outliers) [32].

2.2.1. Τάση (Trend)

Με τον όρο τάση αναφερόμαστε στην αργή απόκλιση που παρατηρείται στο μέσο επίπεδο των τιμών μιας χρονοσειράς και η οποία συντελείται σε μακροπρόθεσμο χρονικό ορίζοντα. Το φαινόμενο αυτό σχετίζεται με τα δομικά χαρακτηριστικά της εκάστοτε χρονοσειράς, που υπόκειται σε εξέταση και μπορεί να έχει είτε ανοδικό, είτε πτωτικό, είτε σταθερό προσανατολισμό.

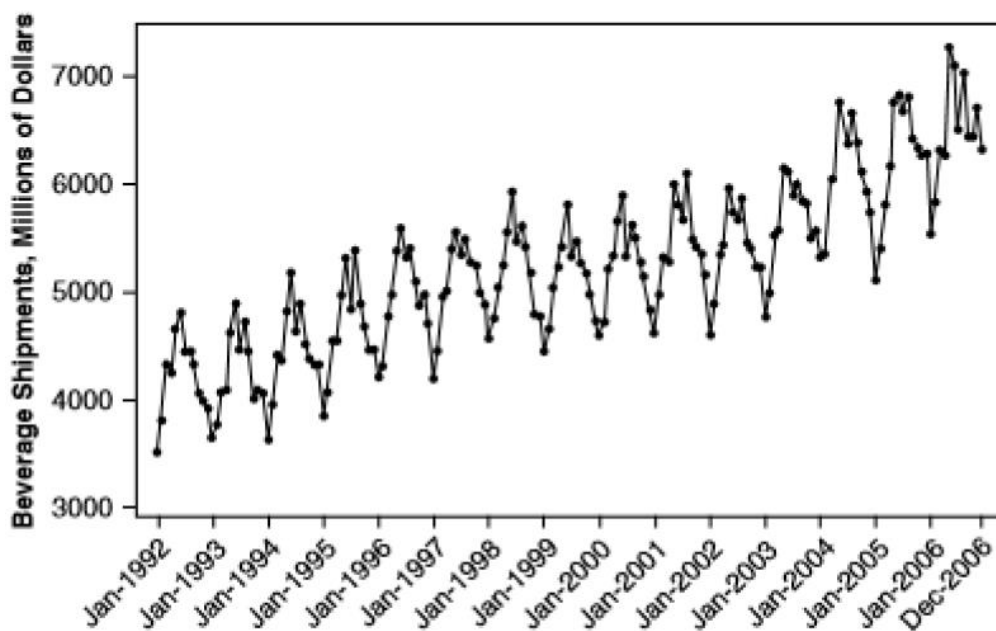
Επίσης για να καταφέρουμε να εξάγουμε ασφαλή συμπεράσματα για το αν μία χρονοσειρά παρουσιάζει τάση ή όχι, είναι απαραίτητη η ύπαρξη επαρκούς αριθμού παρατηρήσεων και χρονικού διαστήματος [33], [34]. Στο παρακάτω διάγραμμα οπτικοποιείται αρκετά ευδιάκριτα το συγκεκριμένο χαρακτηριστικό.



Ημερήσια απεικόνιση των αμερικανικών ομολόγων με καθοδική τάση [35].

2.2.2. Κυκλικότητα (Cyclic)

Η κυκλικότητα είναι μία κατά προσέγγιση περιοδική ταλάντωση, που χαρακτηρίζεται από εναλλασσόμενες περιόδους μεγένθυσης και συρρίκνωσης. Τα χρονικά διαστήματα που παρατηρείται αυτό το φαινόμενο είναι αρκετά μεγάλα, όπως για παράδειγμα ανά πέντε ή ανά δέκα χρόνια. Η κυκλικότητα είναι ένα χαρακτηριστικό που εμφανίζεται με αρκετά μεγάλη συχνότητα σε οικονομικές χρονοσειρές και μπορεί να σχετίζεται με οικονομικά φαινόμενα, όπως υφέσεις και κρίσεις χρέους. Στο παρακάτω διάγραμμα παραγωγής ποτού, παρατηρούμε έναν κυκλικό σχηματισμό που εμφανίζεται ανά τριετία[33], [34].

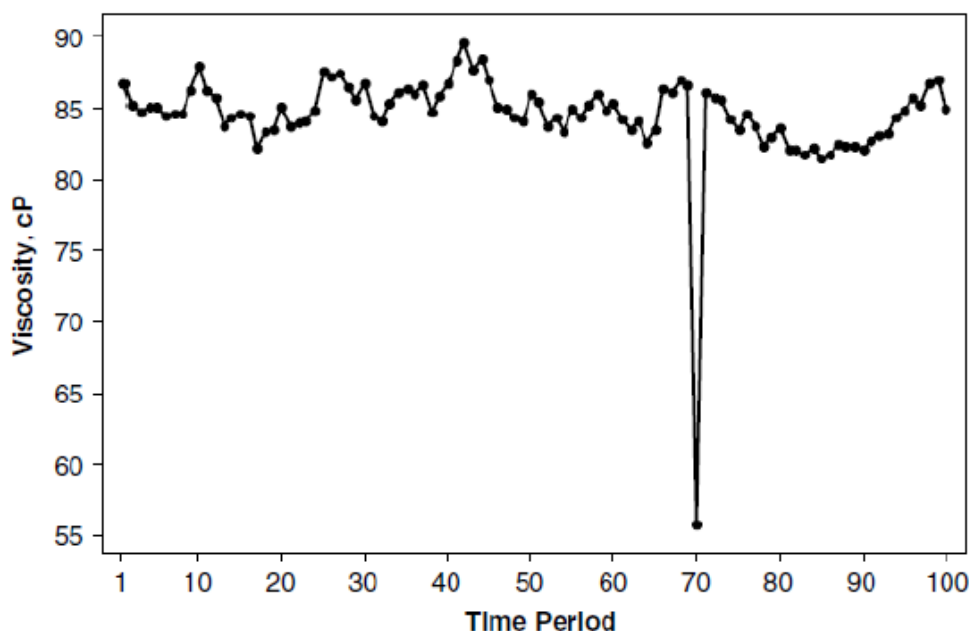


Μηνιαίες αποστολές εμπορεύματος από παραγωγό ποτού στις ΗΠΑ [34].

2.2.3. Ακραίες τιμές (Outliers)

Οι ακραίες τιμές είναι οι παρατηρήσεις που ξεχωρίζουν σε μεγάλο βαθμό από τις υπόλοιπες στο διάγραμμα μιας χρονοσειράς και ουσιαστικά υποδεικνύουν απότομες αλλαγές στο μοτίβο της συμπεριφοράς της. Οι ακραίες τιμές είναι συνήθως αδύνατο να προβλεφθούν και η εμφάνισή τους δεν διαρκεί ιδιαίτερα πολύ χρονικά. Χρειάζεται ιδιαίτερη προσοχή ή μεταχείριση των ακραίων τιμών κατά τη διάρκεια μοντελοποίησης μιας χρονοσειράς, αφού σχεδόν πάντα τείνουν να επηρεάζουν σημαντικά τα αποτελέσματα αυτής της διαδικασίας.

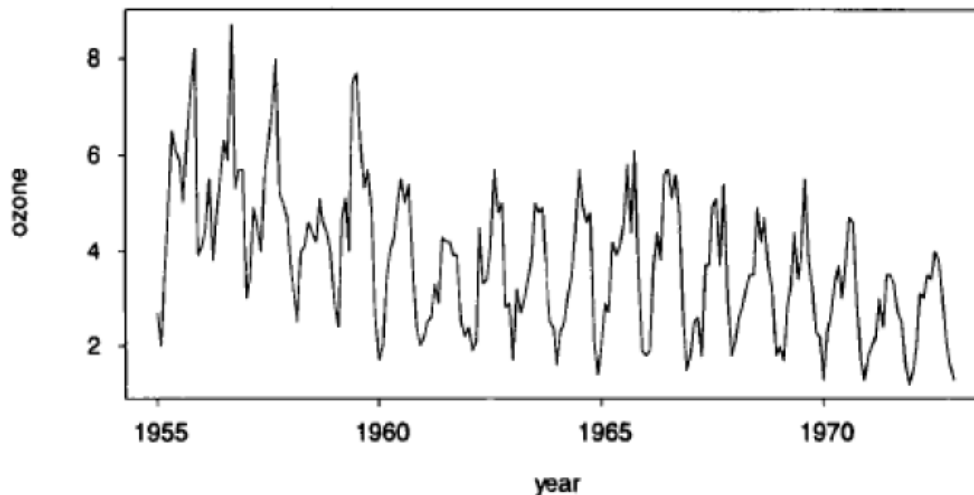
Δεν υπάρχει συγκεκριμένη προσέγγιση ως προς τον χειρισμό των ακραίων τιμών μιας χρονοσειράς, καθώς κάθε περίπτωση είναι διαφορετική και η ερμηνεία τους έγκειται πολλές φορές στην εμπειρία και τη γνωστική ικανότητα του παρατηρητή. Πολλές φορές οι ακραίες τιμές έχουν εύκολα αναγνωρίσιμα αίτια αφού μπορεί να σχετίζονται με το ξέσπασμα φυσικών καταστροφών και με την εμφάνιση απρόβλεπτων κοινωνικών ή ιστορικών γεγονότων, όπως το ξέσπασμα μιας απεργίας σε ένα εργοστάσιο παραγωγής. Στο διάγραμμα που ακολουθεί καταγράφεται μία δυσλειτουργία με τη μορφή ακραίας τιμής στη χρονολογική παρουσίαση μιας διαδικασίας [33], [34].



Διάγραμμα αναγνώσεων ιξώδους χημικής διεργασίας με αισθητήρα δυσλειτουργίας [34].

2.2.4. Εποχικότητα (Seasonality)

Ο όρος εποχικότητα αναφέρεται στις χρονοσειρές για να προσδιορίσει τις εποχικές διακυμάνσεις που παρατηρούνται σε αυτές και συνήθως αναπαριστούν τις επιδράσεις κλιματικών και θεσμικών γεγονότων που επαναλαμβάνονται λίγο πολύ κάθε χρόνο [33]. Στο παρακάτω διάγραμμα παρατηρείται ότι το ατμοσφαιρικό όζον παρουσιάζει έντονη εποχικότητα, η οποία είναι υψηλή κατά τους καλοκαιρινούς μήνες και χαμηλή τον χειμώνα.



Μηνιαίες ενδείξεις όζοντος στο Λος Άντζελες [34].

2.3. Ανάλυση Χρονοσειρών

Με τον όρο ανάλυση χρονοσειρών αναφερόμαστε στη διαδικασία κατά την οποία μοντελοποιούμε μία χρονοσειρά σε μία συγκεκριμένη χρονική αλληλουχία και οι αντίστοιχες παράμετροι της υπολογίζονται με τη χρήση των τιμών των ήδη γνωστών δεδομένων [31]. Ο απώτερος σκοπός αυτής της διαδικασίας είναι η ανάδειξη χαρακτηριστικών της χρονοσειράς που δεν είναι εμφανή σε πρώτη ανάγνωση και κυρίως η πρόβλεψη της μελλοντικής ποσοτικής εξέλιξης των παρατηρήσεων που αποτελούν τη χρονοσειρά.

Συνεπώς η μοντελοποίηση των χρονοσειρών είναι ουσιαστικά μια διαδικασία που έχει τα χαρακτηριστικά της παλινδρόμησης και μπορεί να εξεταστεί ως τέτοια και για την αξιολόγηση της απόδοσης ενός μοντέλου χρησιμοποιούνται συνήθως μέτρα όπως η ρίζα του μέσου τετραγωνικού σφάλματος (RMSE). Ένα βασικό στάδιο της διαδικασίας της μοντελοποίησης είναι εξέταση και η αναγνώριση διαφόρων χαρακτηριστικών της χρονοσειράς, όπως τα τέσσερα που αναφέρθηκαν παραπάνω και ο κατάλληλος χειρισμός τους.

Η διαδικασία κατά την οποία αφαιρούμε από τη χρονοσειρά κάποιο από τα τέσσερα βασικά χαρακτηριστικά της (τάση, κυκλικότητα, εποχικότητα, ακραίες τιμές) ονομάζεται **decomposition** και είναι πολύ σημαντική, καθώς η ύπαρξη τους μπορεί να επηρεάσει σε μεγάλο βαθμό την προβλεπτική ικανότητα του εκάστοτε μοντέλου. Επίσης είναι σημαντικό να ελέγξουμε τη χρονοσειρά για την ύπαρξη λευκού θορύβου, δηλαδή η αναμενόμενη ή μέση διακύμανση σε οποιαδήποτε στιγμή της χρονοσειράς να είναι σταθερή. Η ύπαρξη λευκού θορύβου στη χρονοσειρά υποδηλώνει ότι κάθε παρατήρηση δε σχετίζεται με τις υπόλοιπες στη χρονολογική ακολουθία.

Ακόμη μία βασική ιδιότητα των χρονοσειρών είναι η στασιμότητα των δεδομένων. Με τον όρο στασιμότητα αναφερόμαστε σε μία χρονοσειρά που η κοινή κατανομή των παρατηρήσεων της δεν μεταβάλλεται με την πάροδο του χρόνου και η ύπαρξη της είναι αρκετά σημαντική για την καλή προβλεπτική ικανότητα ενός μοντέλου [36]. Οι χρονοσειρές που αποτελούνται από πραγματικές παρατηρήσεις τείνουν να παρουσιάζουν μικρό βαθμό στασιμότητας και προκειμένου να αντιμετωπιστεί αυτό το πρόβλημα εφαρμόζεται σε αυτές η μέθοδος decomposition. Αυτό γίνεται γιατί πολλές φορές η μη ύπαρξη στασιμότητας σε μία χρονοσειρά υποδηλώνει την ύπαρξη ενός ή

περισσότερων εκ των βασικών χαρακτηριστικών της. Ακόμη μία γνωστή μέθοδος για την αντιμετώπιση του προβλήματος της έλλειψης στασιμότητας ενός μοντέλου είναι η κανονικοποίηση των δεδομένων.

Τα βασικά μοντέλα που χρησιμοποιούνται για την ανάλυση χρονοσειρών είναι μοντέλα παλινδρόμησης όπως για παράδειγμα τα Moving Average (MA) models, Auto Regressive (AR) models, Auto Regressive Moving Average (ARMA) models και τα Auto Regressive Integrated Moving Average (ARIMA) models. Τα τελευταία όμως χρόνια με την τεράστια αύξηση των διαθέσιμων δεδομένων σε συνδυασμό με την ανάλογη αύξηση της διαθέσιμης υπολογιστικής ισχύς έχουν αρχίσει να χρησιμοποιούνται μοντέλα βασισμένα σε αλγόριθμους μηχανικής μάθησης και αρχιτεκτονικές νευρωνικών δικτύων όπως ο k-Nearest Neighbor (kNN), τα Decision Trees [37], οι Multilayer Perceptrons και τα Recurrent Neural Networks [38], [39].

ΜΕΡΟΣ 3: ΠΕΙΡΑΜΑΤΙΚΟ ΣΚΕΛΟΣ

3.1. Βασική Ιδέα

Ο κύριος στόχος του τρίτου μέρους είναι να εξετάσουμε τον τρόπο με τον οποίο λειτουργούν και τα αποτελέσματα τα οποία παράγουν οι βασικότερες αρχιτεκτονικές τεχνητών νευρωνικών δικτύων, που χρησιμοποιούνται για την παρατήρηση και μελέτη των τιμών των μετοχών που διαπραγματεύονται στις χρηματιστηριακές αγορές. Ουσιαστικά πρόκειται για μία προσπάθεια ανάλυσης χρονοσειρών, οι παρατηρήσεις των οποίων αντιπροσωπεύουν την εξέλιξη της τιμής συγκεκριμένων πραγματικών μετοχών, με τη χρήση τεχνητών νευρωνικών δικτύων.

Η διαδικασία αυτή θα χωριστεί σε τρία μέρη. Το πρώτο μέρος περιλαμβάνει την παρουσίαση του συνόλου δεδομένων (dataset), την επιλογή των χρονοσειρών που θα εξετάσουμε και τον αρχικό έλεγχο του ως προς τα χαρακτηριστικά τους (exploratory analytics). Το δεύτερο περιλαμβάνει την επιλογή των αρχιτεκτονικών νευρωνικών δικτύων που θα χρησιμοποιήσουμε, την κατασκευή τους και την πραγματοποίηση πειραμάτων για την αξιολόγηση της απόδοσης τους. Το τρίτο και τελευταίο μέρος περιλαμβάνει την παρουσίαση των συμπερασμάτων τα οποία βγήκαν από την παρατήρηση των αποτελεσμάτων των πειραμάτων του δεύτερου μέρους.

Στο πρώτο μέρος το εργαλείο που θα χρησιμοποιήσουμε είναι η **R**, η οποία είναι αρκετά μία γλώσσα στατιστικού προγραμματισμού και πιο συγκεκριμένα την πλατφόρμα R-studio στην έκδοση 3.5.3. Στο δεύτερο μέρος θα χρησιμοποιήσουμε τη γενική και πολύ διαδεδομένη γλώσσα προγραμματισμού **python** στην έκδοση 3.7.4 και πιο συγκεκριμένα την πλατφόρμα της **Anaconda** στην έκδοση 2019.10, η οποία είναι κατάλληλη για ανάλυση δεδομένων. Η χρήση της Anaconda είναι επιβεβλημένη προκειμένου να χρησιμοποιηθούν οι βιβλιοθήκες **TensorFlow** και **Keras**, οι οποίες είναι απαραίτητες για τη δημιουργία των τεχνητών νευρωνικών δικτύων.

3.2. Διερεύνηση Δεδομένων (Exploratory Analytics)

3.2.1. Τα Δεδομένα (Dataset)

Το αρχικό σύνολο δεδομένων (dataset), δεδομένα του οποίου θα χρησιμοποιήσουμε για την εκπαίδευση και την αξιολόγηση των τεχνητών νευρωνικών δικτύων,

περιλαμβάνει τα ιστορικά δεδομένα για πέντε τεχνολογικές εταιρείες που είναι εισηγμένες στον αμερικάνικο χρηματιστηριακό δείκτη Nasdaq και βρίσκεται ανεβασμένο στο δημοφιλή ιστότοπο, με αντικείμενο την ανάλυση δεδομένων, kaggle [40].

Οι εταιρείες για τις οποίες περιλαμβάνει πληροφορίες είναι η Microsoft, η Apple, η Google, η Amazon και το Facebook. Το σύνολο δεδομένων (dataset) είναι σε μορφή csv, έχει μέγεθος 1,92 megabytes και έχει 41.518 εγγραφές, οι οποίες κατανέμονται στο χρονικό διάστημα μεταξύ 05-02-1971 και 23-08-2019. Οι στήλες έχουν τα ονόματα name, close, open, high, low και volume. Η στήλη name έχει τα ονόματα των εταιρειών, η στήλη close έχει την τιμή της μετοχής κάθε εταιρείας στο κλείσιμο του δείκτη, η στήλη open έχει την τιμή της μετοχής κάθε εταιρείας στο άνοιγμα του δείκτη, η στήλη high έχει την υψηλότερη τιμή της μετοχής της εκάστοτε εταιρείας ανά ημερήσια διαπραγμάτευση, η στήλη low έχει τη χαμηλότερη τιμή της μετοχής της εκάστοτε εταιρείας ανά ημερήσια διαπραγμάτευση και η στήλη volume έχει τον αριθμό των μετοχών που διακινήθηκαν ανά εταιρεία.

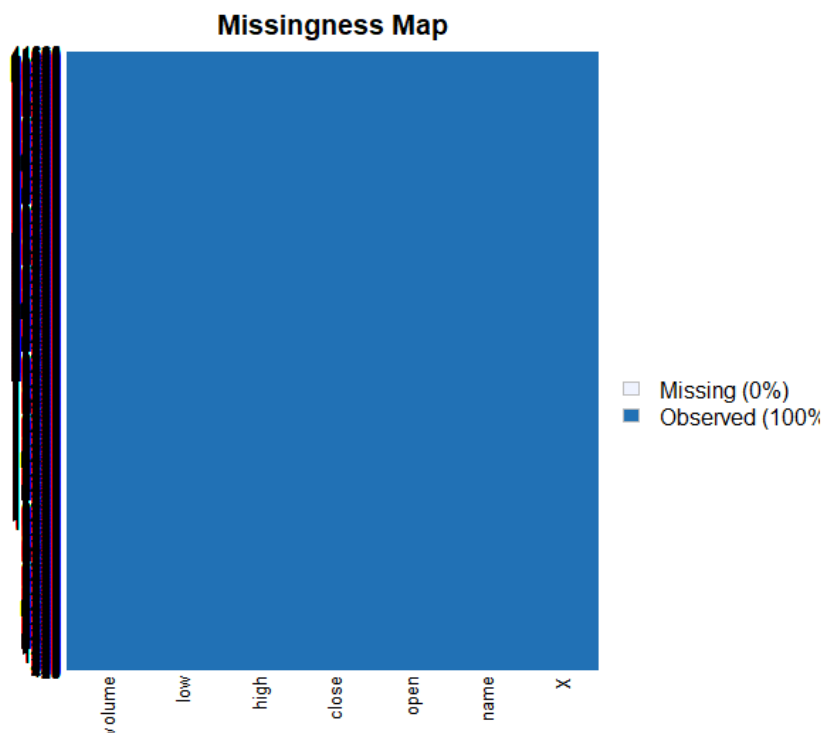
Υπάρχει και άλλη μία στήλη η οποία περιέχει τις ημερομηνίες των παρατηρήσεων και η οποία δε φέρει όνομα. Η συχνότητα των καταγραφών είναι ημερήσια και ακολουθεί προφανώς τις εργάσιμες ημέρες λειτουργίας του δείκτη διαπραγμάτευσης, δηλαδή Δευτέρα με Παρασκευή με εξαίρεση τις επίσημες αμερικανικές αργίες.

3.2.2. Αρχική Επεξεργασία Δεδομένων

Τα πρώτα βήματα που πρέπει να ακολουθήσουμε για την επεξεργασία ενός αρχείου που περιέχει χρονολογικά δεδομένα είναι με τη σειρά η εισαγωγή του αρχείου στο προγραμματιστικό περιβάλλον που θα το επεξεργαστούμε, ο έλεγχος του αρχείου για για απολεσθέντες τιμές (missing values), ο ορισμός των χρονοσειρών και η οπτικοποίηση της αρχικής τους μορφής.

Για να υλοποιήσουμε τα παραπάνω βήματα θα χρησιμοποιήσουμε τις βιβλιοθήκες **dplyr**, **ggplot**, **plotly** και **amelia** που υπάρχουν στο περιβάλλον της R. Η dplyr είναι μία από τις βασικότερες βιβλιοθήκες της R για το μετασχηματισμό δεδομένων που είναι σε μορφή dataframe, επιτρέποντας μας να κάνουμε πολύ εύκολα ομαδοποιήσεις, διαχωρισμούς, εξαγωγές και αποκοπές δεδομένων από το αρχικό αρχείο. Οι βιβλιοθήκες ggplot και plotly είναι απαραίτητες για την υλοποίηση σχεδόν των περισσότερων οπτικοποιήσεων που μπορούν να πραγματοποιηθούν στο περιβάλλον της R.

Επομένως μετά την εισαγωγή του συνόλου δεδομένων (dataset), σειρά έχει ο έλεγχος για την ύπαρξη απολεσθέντων τιμών (missing values). Ο έλεγχος αυτός θα γίνει με τη χρήση της βιβλιοθήκης amelia και τη δημιουργία ενός missingness map. Το missingness map είναι ένα διάγραμμα το οποίο οπτικοποιεί τις απολεσθέντες τιμές (missing values) του συνόλου δεδομένων (dataset) και δίνει πληροφορίες για τον τρόπο που κατανέμονται σε περίπτωση που υπάρχουν.



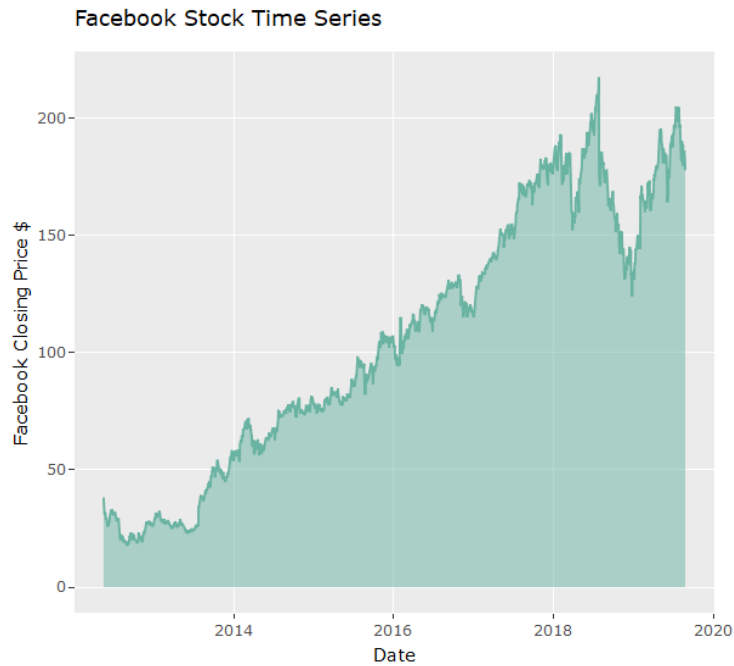
To missingness map για το σύνολο των δεδομένων (dataset).

Το missingness map μας πληροφορεί ότι δεν υπάρχουν καθόλου απολεσθέντες τιμές (missing values) στο σύνολο δεδομένων (dataset). Αυτή είναι μία πολύ σημαντική πληροφορία, καθώς οι απολεσθέντες τιμές (missing values) δημιουργούν προβλήματα στη περαιτέρω επεξεργασία των δεδομένων και θα έπρεπε να αντικατασταθούν ή να απαλειφθούν από το σύνολο των δεδομένων (dataset). Η πιο δόκιμη προσέγγιση από τις δύο σε τέτοιες περιπτώσεις είναι αντικατάσταση των απολεσθέντων τιμών (missing values) με ένα στατιστικό μέγεθος, όπως η μέση τιμή, της μεταβλητής με την οποία σχετίζονται.

3.2.3. Οπτικοποίηση Χρονοσειρών

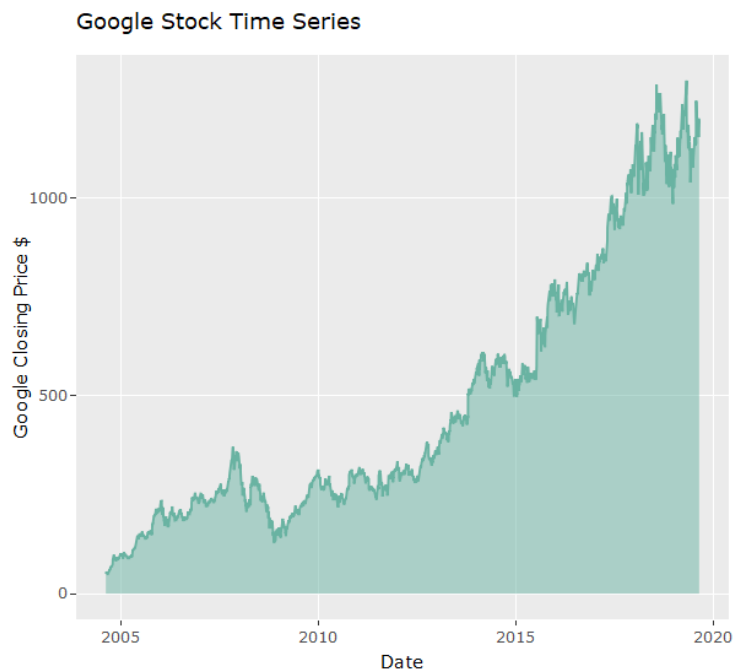
Οι χρονοσειρές που θα δημιουργήσουμε, προκειμένου να χρησιμοποιηθούν για την εκπαίδευση των νευρωνικών δικτύων στη συνέχεια, είναι τρεις και απεικονίζουν την εξέλιξη της τιμής κλεισίματος της μετοχής (close) των εταιρειών Facebook, Google και Amazon. Η τιμή κλεισίματος της μετοχής είναι το σημαντικότερο χαρακτηριστικό που περιέχει το σύνολο δεδομένων (dataset), αφού αντικατοπτρίζει την πορεία της μετοχής, άρα και της εταιρείας, σε ημερήσια βάση μετά το κλείσιμο των συναλλαγών στο δείκτη Nasdaq.

Αρχικά θα δημιουργήσουμε τη χρονοσειρά για την τιμή κλεισίματος της μετοχής της εταιρείας Facebook, δημιουργώντας μία μεταβλητή η οποία θα έχει 1828 παρατηρήσεις που κατανέμονται στο χρονικό διάστημα μεταξύ 18-05-2012 και 23-08-2019. Η μορφή της χρονοσειράς απεικονίζεται καθαρά στο παρακάτω διάγραμμα.



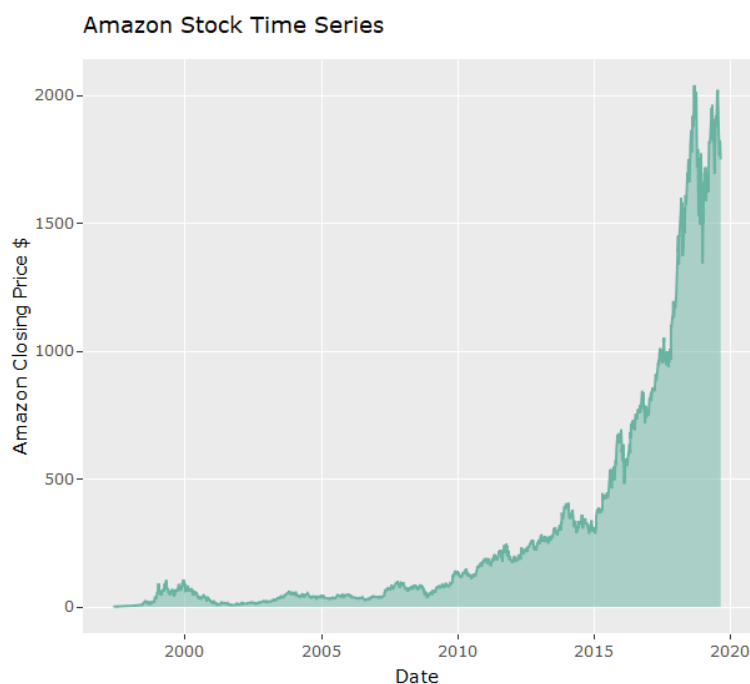
Η διαγραμματική απεικόνιση της τιμής κλεισίματος (close) της μετοχής της εταιρείας Facebook σε συνάρτηση με το χρόνο.

Στη συνέχεια θα δημιουργήσουμε τη χρονοσειρά για την τιμή κλεισίματος της μετοχής της εταιρείας Google, δημιουργώντας μία μεταβλητή η οποία θα έχει 3780 παρατηρήσεις που κατανέμονται στο χρονικό διάστημα μεταξύ 19-08-2004 και 23-08-2019. Η μορφή της χρονοσειράς απεικονίζεται καθαρά στο παρακάτω διάγραμμα.



Η διαγραμματική απεικόνιση της τιμής κλεισίματος της μετοχής (close) της εταιρείας Google σε συνάρτηση με το χρόνο.

Τέλος θα δημιουργήσουμε τη χρονοσειρά για τη τιμή κλεισίματος της μετοχής της εταιρείας Amazon, δημιουργώντας μία μεταβλητή η οποία θα έχει 5607 παρατηρήσεις που κατανέμονται στο χρονικό διάστημα μεταξύ 15-05-1997 και 23-08-2019. Η μορφή της χρονοσειράς απεικονίζεται καθαρά στο παρακάτω διάγραμμα.



Η διαγραμματική απεικόνιση της τιμής κλεισίματος της μετοχής (close) της εταιρείας Amazon σε συνάρτηση με το χρόνο.

Γνωρίζοντας πλέον τη διαγραμματική μορφή των τριών χρονοσειρών που αφορούν τις εταιρείες Facebook, Google και Amazon, μπορούμε να τις εξετάσουμε για την ύπαρξη των επιμέρους χαρακτηριστικών τους. Θα εξετάσουμε αν παρουσιάζουν τάση, κυκλικότητα, εποχικότητα, ακραίες τιμές και αν εφαρμόσουμε τη μέθοδο απαλοιφής χαρακτηριστικών (decomposition) σε κάποια από αυτές.

Αρχικά μπορούμε να παρατηρήσουμε αρκετά εύκολα ότι καμία από τις τρεις χρονοσειρές δεν παρουσιάζει το φαινόμενο της κυκλικότητας και της εποχικότητας. Επίσης ξεκάθαρα δεν υφίσταται η ύπαρξη ακραίων τιμών στη χρονοσειρά του Facebook και της Google. Όσον αφορά την Amazon οι τελευταίες παρατηρήσεις της χρονοσειράς, εκ πρώτης όψεως, φαίνονται ως ακραίες τιμές (outliers) σε σχέση με τις αρχικές της παρατηρήσεις. Σε δεύτερη ανάγνωση όμως παρατηρούμε ότι από το 2010 και μετά η αύξηση της τιμής κλεισίματος της Amazon είναι αλματώδης. Τέλος σε όλες τις χρονοσειρές είναι ευδιάκριτη η μεγάλη αυξητική τάση της τιμής κλεισίματος, αν και υπάρχουν σημαντικές διακυμάνσεις σε αρκετά χρονικά διαστήματα.

Ο διαγραμματικός έλεγχος των τριών χρονοσειρών μπορεί να μας δίνει εύκολα αρκετά σημαντικές πληροφορίες και με πολύ παραστατικό τρόπο, δεν αρκεί όμως για να καταλήξουμε σε με απόλυτη βεβαιότητα στην εφαρμογή ή όχι της μεθόδου απαλοιφής

χαρακτηριστικών (decomposition) σε κάποια από τις χρονοσειρές. Επομένως για να βεβαιωθούμε ότι οι χρονοσειρές της τιμής των μετοχών των τριών εταιρειών παρουσιάζουν στασιμότητα θα κάνουμε Dickey-Fuller tests για καθεμία από αυτές. Σκοπός του Dickey-Fuller test είναι να παράξει ένα μικρό p-value, ώστε να απορρίψουμε την αρχική υπόθεση μας ότι η χρονοσειρά δεν είναι σταθερή. Τα τελικά αποτελέσματα είναι p-value ίσο με 0.2763 για τη χρονοσειρά του Facebook, p-value ίσο με 0.7831 για τη χρονοσειρά της Google, ενώ για την Amazon το p-value είναι υψηλότερο από το μέγιστο που τυπώνει το τεστ (λογικά μεγαλύτερο της μονάδας).

Από τα παραπάνω αποτελέσματα είναι σίγουρο ότι και οι τρεις χρονοσειρές δεν είναι στάσιμες και σε περίπτωση που θα θέλαμε να δημιουργήσουμε τα παραδοσιακά μοντέλα πρόβλεψης χρονοσειρών, όπως το ARIMA, η εφαρμογή της διαδικασίας απαλοιφής χαρακτηριστικών (decomposition) θα ήταν απαραίτητη.

Μία όμως από τις βασικές ιδιότητες των τεχνητών νευρωνικών δικτύων είναι η ικανότητα τους να εντοπίζουν μοτίβα και αλληλουχίες σε μη επεξεργασμένα δεδομένα. Επομένως όσων αφορά τη διαδικασία της απαλοιφής χαρακτηριστικών για τις χρονοσειρές (decomposition) στα τεχνητά νευρωνικά δίκτυα υπάρχουν δύο προσεγγίσεις. Η πρώτη προσέγγιση αναφέρει ότι η διαδικασία της απαλοιφής χαρακτηριστικών (decomposition) δεν είναι απαραίτητο να εφαρμόζεται στα μοντέλα τεχνητών νευρωνικών δικτύων καθολικά [41], ενώ η δεύτερη υποστηρίζει ότι η εφαρμογή της απαλοιφής χαρακτηριστικών (decomposition) στα μοντέλα τεχνητών νευρωνικών δικτύων μπορεί να βελτιώσει την προβλεπτική τους ικανότητα σε σημαντικό βαθμό [42].

Κατά την άποψη μου δεν υπάρχει σωστή ή λάθος προσέγγιση, καθώς εξαρτάται από τον σκοπό για τον οποίο δημιουργούμε το εκάστοτε μοντέλο τεχνητού νευρωνικού δικτύου. Αν για παράδειγμα θέλουμε να εξετάσουμε το μέγιστο των δυνατοτήτων ενός τεχνητού νευρωνικού δικτύου στην επεξεργασία δεδομένων, τότε δεν χρειάζεται να εφαρμοστεί η μέθοδος απαλοιφής χαρακτηριστικών (decomposition). Σε περίπτωση όμως που θέλουμε να εστιάσουμε στην προβλεπτική ικανότητα του τεχνητού νευρωνικού δικτύου, με τα αρχικά αποτελέσματα του να μην μας ικανοποιούν, τότε μπορούμε να εφαρμόσουμε τη μέθοδο απαλοιφής χαρακτηριστικών (decomposition) για να τα βελτιώσουμε.

3.3. Μοντελοποίηση Νευρωνικών Δικτύων

3.3.1. Τεχνητά Νευρωνικά Δίκτυα προς Μοντελοποίηση

Για τη δημιουργία μοντέλων νευρωνικών δικτύων, προκειμένου να αξιολογήσουμε τον τρόπο με τον οποίο λειτουργούν και προβλέπουν χρησιμοποιώντας τις χρονοσειρές που δημιουργήσαμε προηγουμένως, θα επιλέξουμε τις τρεις πιο διαδεδομένες αρχιτεκτονικές τεχνητών νευρωνικών δικτύων για την ανάλυση univariate χρονοσειρών. Αυτές οι τρεις είναι οι Multilayer Perceptrons (MLP), τα Recurrent Neural Networks (RNN) στην πιο ειδική μορφή τους τα Long-Short Term Memory Networks (LSTM) και τα Convolutional Neural Networks (CNN). Τα μοντέλα θα κατασκευαστούν με χρήση του API Keras και θα αξιολογηθούν με τη μέθοδο cross-validation και τη χρήση ενός στατιστικού μέτρου που θα υποδηλώνει το συνολικό σφάλμα του μοντέλου να προβλέψει σωστά τις παρατηρήσεις του εκάστοτε συνόλου δεδομένων προς εκπαίδευση (training set).

3.3.2. Keras Configuration

Το Keras είναι ένα από τα βασικότερα και πιο εύχρηστα API που μπορεί να χρησιμοποιήσει κάποιος για τη δημιουργία τεχνητών νευρωνικών δικτύων στη γλώσσα προγραμματισμού python και το οποίο βασίζεται στην πλατφόρμα μηχανικής μάθησης (machine learning) TensorFlow. Περιλαμβάνει αρκετές βιβλιοθήκες που απλοποιούν τη διαδικασία δημιουργίας ενός νευρωνικού δικτύου και οι οποίες παρέχουν τη δυνατότητα στον χρήστη να κατασκευάζει και να παραμετροποιεί το κάθε ένα επίπεδο του τεχνητού νευρωνικού δικτύου ξεχωριστά.

Οι βασικές παράμετροι που συνθέτουν ένα τεχνητό νευρωνικό δίκτυο στο Keras είναι το είδος του μοντέλου (model), η συνάρτηση ενεργοποίησης (activation), ο μεταγλωττιστής του νευρωνικού δικτύου (compiler), η συνάρτηση εκπαίδευσης (optimizer) και οι επιλογές εκπαίδευσης του μοντέλου (fit).

Στο κέρας υπάρχουν δύο επιλογές ως προς το είδος του μοντέλου του νευρωνικού δικτύου. Το ένα είναι το Sequential model [43] και το άλλο το Functional API model [44]. Για τη δημιουργία των μοντέλων νευρωνικών δικτύων στη συνέχεια θα χρησιμοποιήσουμε αποκλειστικά το Sequential model, το οποίο ουσιαστικά είναι μία γραμμική στίβα πολλών επιπέδων.

Ως προς τη συνάρτηση ενεργοποίησης υπάρχουν αρκετές επιλογές όπως η exponential linear unit (elu), η softmax, η scaled exponential linear unit (selu), η softplus, η softsign, η relu, η tanh, η sigmoid, η hard_sigmoid, η exponential και η linear. Η πιο διαδεδομένη από τις παραπάνω είναι η relu και αυτή χρησιμοποιήσουμε ως συνάρτηση ενεργοποίησης σε όλα τα μοντέλα που θα δημιουργήσουμε.

Ο μεταγλωττιστής (compiler) του εκάστοτε τεχνητού νευρωνικού δικτύου μας δίνει τη δυνατότητα να επιλέξουμε τον κατηγοριοποιητή (optimizer) που θα είναι υπεύθυνος για την εκπαίδευση του νευρωνικού δικτύου και το στατιστικό μέτρο που θα αξιολογήσει την απόδοση του κάθε μοντέλου. Οι διαθέσιμοι κατηγοριοποιητές (optimizers) στο keras είναι ο Stochastic Gradient Descent (Sgd), ο RMSprop, ο Adagrad, ο Adadelta, ο Adam, ο Adamax και ο Nadam. Οι Adam και Sgd είναι οι πιο διαδεδομένοι κατηγοριοποιητές (optimizers) και μαζί με τον RMSprop θα τους χρησιμοποιήσουμε για την εκπαίδευση των μοντέλων που θα δημιουργήσουμε. Επίσης με τη χρήση του Mean Square Error (MSE) θα αξιολογήσουμε την απόδοση των μοντέλων των τεχνητών νευρωνικών δικτύων, την λειτουργία των οποίων θα εξετάσουμε.

Τέλος η επιλογή fit μας δίνει τη δυνατότητα να επιλέξουμε τον αριθμό των epochs, δηλαδή το πόσες φορές ολόκληρο το σύνολο δεδομένων (dataset) θα περάσει διαμέσου του τεχνητού νευρωνικού δικτύου και το batch size, δηλαδή ο αριθμός των παραδειγμάτων εκπαίδευσης που υπάρχουν σε κάθε παρτίδα (batch). Με λίγα λόγια με το batch size χωρίζουμε το σύνολο δεδομένων (dataset) για να το εισάγουμε στο τεχνητό νευρωνικό δίκτυο, καθώς δε μπορεί να εισαχθεί ολόκληρο με τη μία.

3.4. Multilayer Perceptron (MLP)

Η πρώτη αρχιτεκτονική τεχνητού νευρωνικού δικτύου που θα εξετάσουμε ως προς την λειτουργία της με τη χρήση χρηματιστηριακών δεδομένων σε μορφή χρονοσειράς μίας μεταβλητής (univariate) είναι αυτή του MLP. Για να το επιτύχουμε αυτό θα κατασκευάσουμε ένα στάνταρ μοντέλο και θα τρέξουμε αρκετά πειράματα με διαφορετικές παραμέτρους (configuration), χρησιμοποιώντας και τα τρεις χρονοσειρές

που δημιουργήσαμε. Είναι απαραίτητο να αναφέρουμε ότι δεν υπάρχει κάποιος τρόπος να ξέρουμε τις βέλτιστες παραμέτρους πριν πραγματοποιήσουμε πολλές δοκιμές, ώστε να οδηγηθούμε σε ασφαλή συμπεράσματα.

Ο MLP που θα χρησιμοποιήσουμε θα είναι ένα απλό τεχνητό νευρωνικό δίκτυο, το οποίο θα αποτελείται από τρία επίπεδα. Τα τρία επίπεδα θα είναι ένα επίπεδο εισόδου, ένα κρυμμένο επίπεδο και ένα επίπεδο εξόδου. Μετά από πολλές δοκιμές καταλήξαμε ότι τα βέλτιστα αποτελέσματα για τον MLP παράγονται με τη χρήση 12 νευρώνων, με το διαχωρισμό των συνόλων δεδομένων (dataset) σε 70% το σύνολο δεδομένων εκπαίδευσης (training set) και 30% το σύνολο δεδομένων αξιολόγησης (test set) και με το ρυθμό εισαγωγής δεδομένων στο τεχνητό νευρωνικό δίκτυο (batch size) να ισούται με 50.

Αυτά τα νούμερα θα είναι ίδια σε όλα τα πειράματα που θα παρουσιάσουμε στη συνέχεια, όπως και η συνάρτηση ενεργοποίησης, η οποία θα είναι η relu. Ουσιαστικά οι παράμετροι που θα αλλάζουν σε κάθε πείραμα είναι ο αριθμός των χρονικών επαναλήψεων (epochs) και από έναν αριθμό πειραμάτων και μετά ο κατηγοριοποιητής (optimizer).

Ακόμη θα δημιουργήσουμε μία συνάρτηση η οποία θα παίρνει δύο μεταβλητές, με τη μία να είναι το σύνολο δεδομένων (dataset) σε μορφή numpy array και η άλλη να είναι η μεταβλητή look_back, η οποία θα αντιπροσωπεύει των αριθμό των προηγούμενων χρονικών βημάτων που χρησιμοποιήθηκαν ως μεταβλητές εισόδου για την πρόβλεψη της μελλοντικής χρονικής περιόδου και η αρχική της τιμή θα ισούται με μονάδα. Με αυτό τον τρόπο θα δημιουργηθεί ένα σύνολο δεδομένων (dataset) όπου με X θα συμβολίζεται η τιμή κλεισίματος της μετοχής τη χρονική στιγμή t και με Y η τιμή κλεισίματος της μετοχής την αμέσως επόμενη χρονική στιγμή t+1.

Επίσης από τη στιγμή που θα μετασχηματιστεί το σύνολο δεδομένων (dataset) σε numpy array και η χρονοσειρά μας είναι μίας μεταβλητής (univariate) δεν υπάρχει λόγος να κρατήσουμε τις ημερομηνίες του συνόλου δεδομένων (dataset), αφού πλέον το πρόβλημα πρόβλεψης χρονοσειράς έχει μετατραπεί σε πρόβλημα πρόβλεψης παλινδρόμησης με συνεχόμενο αριθμό παρατηρήσεων που αντικατοπτρίζουν συνεχόμενες ημέρες χρονικά. Τέλος τα δεδομένα μας θα κανονικοποιηθούν με τη χρήση του MinMaxScaler από τη βιβλιοθήκη Sklearn και θα χρησιμοποιηθεί ο αλγόριθμος Backpropagation κατά την εκπαίδευση του τεχνητού νευρωνικού δικτύου, η οποία εφαρμόζεται αυτοματοποιημένα από το Keras.

3.4.1. Αποτελέσματα MLP για το Facebook

Τα αποτελέσματα του MLP στη χρονοσειρά της τιμής κλεισίματος της μετοχής της εταιρείας Facebook παρουσιάζονται στον παρακάτω πίνακα.

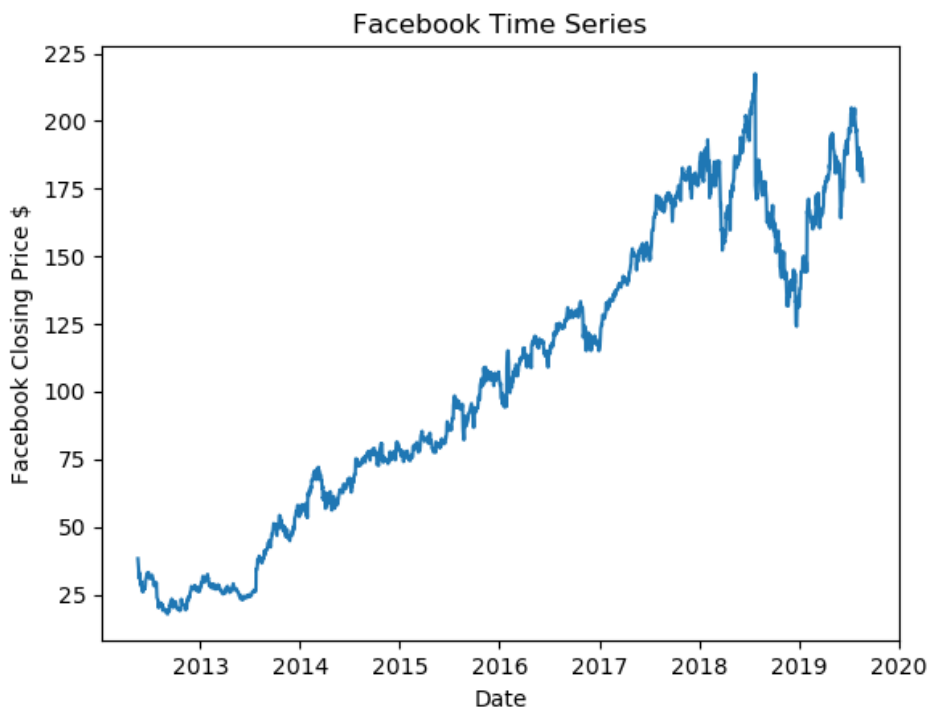
TEST	EPOCHS	OPTIMIZER	MSE	RMSE
1	200	ADAM	12.69	3.56
2	400	ADAM	12.47	3.53
3	600	ADAM	12.47	3.53
4	800	ADAM	12.65	3.56

5	200	SGD	14.55	3.81
6	400	SGD	12.80	3.58
7	600	SGD	12.78	3.58
8	800	SGD	12.81	3.58
9	200	RMSPROP	12.49	3.53
10	400	RMSPROP	13.41	3.66
11	600	RMSPROP	13.29	3.65
<u>12</u>	<u>800</u>	<u>RMSPROP</u>	<u>12.46</u>	<u>3.53</u>

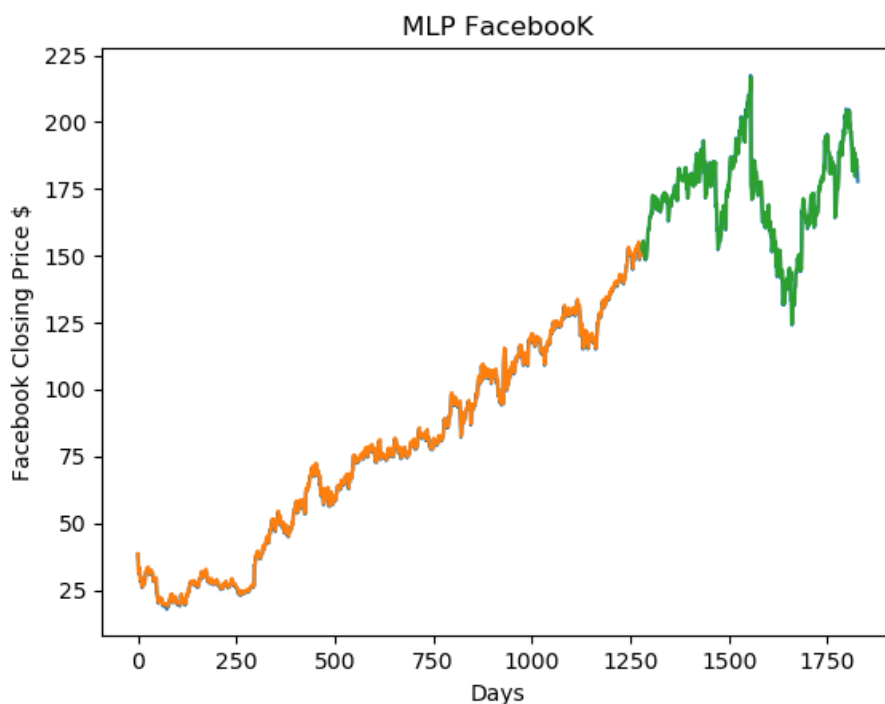
Παρόλο που οι διαφορές είναι σχετικά μικρές, το μικρότερο MSE το παράγει ο κατηγοριοποιητής (optimizer) RMSprop στα 800 epochs. Το αποτέλεσμα αυτό μας υποδηλώνει ότι σε αντίθεση με τους άλλους τρεις κατηγοριοποιητές (optimizers), που αρχίζουν να παρουσιάζουν το φαινόμενο του κορεσμού (overfitting) στα δεδομένα μετά τα 600 epochs, ο RMSprop δίνει το καλύτερο αποτέλεσμα τότε.

Χρησιμοποιώντας τις παραμέτρους του τεστ 12, που παρουσιάζει το μικρότερο MSE, θα παρουσιάσουμε διαγραμματικά τα αποτελέσματα του σε σύγκριση με την αρχική χρονοσειρά. Για να το επιτύχουμε αυτό θα χρησιμοποιήσουμε τις βιβλιοθήκες pandas και matplotlib.

Τα διαγραμματικά αποτελέσματα του τεστ 12 παρουσιάζονται παρακάτω.



Η πραγματική χρονοσειράς της τιμής κλεισίματος της μετοχής της εταιρείας Facebook.



Η χρονοσειρά που παράγεται από τον MLP για την τιμή κλεισίματος της μετοχής της εταιρείας Facebook. Με πορτοκαλί απεικονίζονται οι παρατηρήσεις του training set και με πράσινο οι παρατηρήσεις του test set και με μπλε οι πραγματικές παρατηρήσεις.

Ουσιαστικά η διαγραμματική απεικόνιση αποτυπώνει ξεκάθαρα τα πολύ καλά αποτελέσματα που παρουσιάστηκαν και παραπάνω. Το μοντέλο ακολουθεί με μεγάλη ακρίβεια την πορεία της πραγματικής χρονοσειράς.

3.4.2. Αποτελέσματα MLP για τη Google

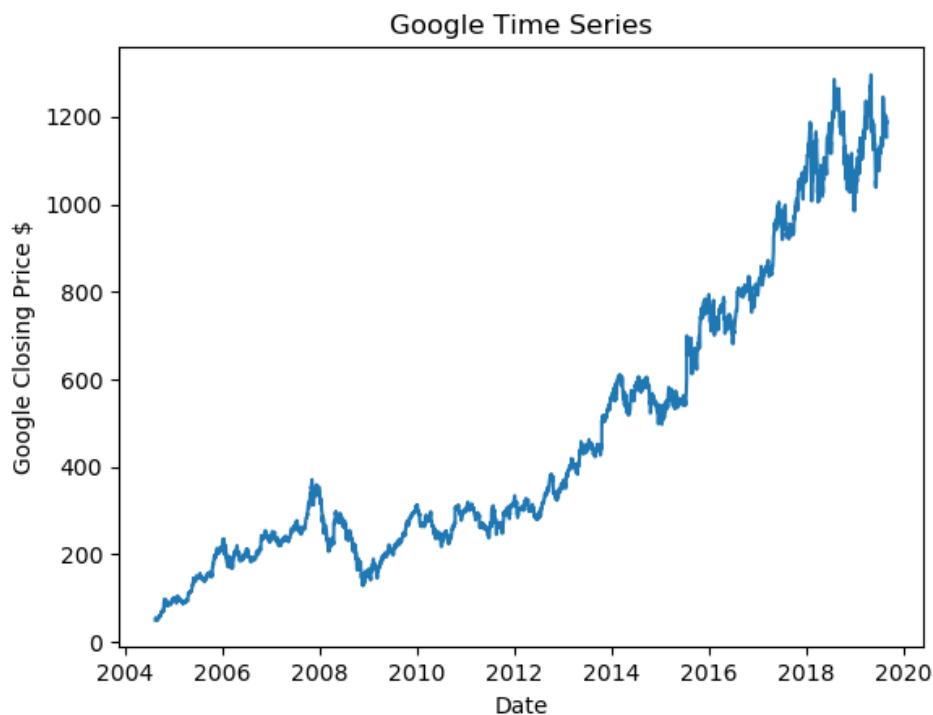
Τα αποτελέσματα του MLP στη χρονοσειρά της τιμής κλεισίματος της μετοχής της εταιρείας Google παρουσιάζονται στον παρακάτω πίνακα.

TEST	EPOCHS	OPTIMIZER	MSE	RMSE
1	200	ADAM	335.91	18.33
2	400	ADAM	333.48	18.26
3	600	ADAM	316.78	17.80
4	800	ADAM	294.18	17.15
5	200	SGD	8489.33	92.14
6	400	SGD	477.65	21.86

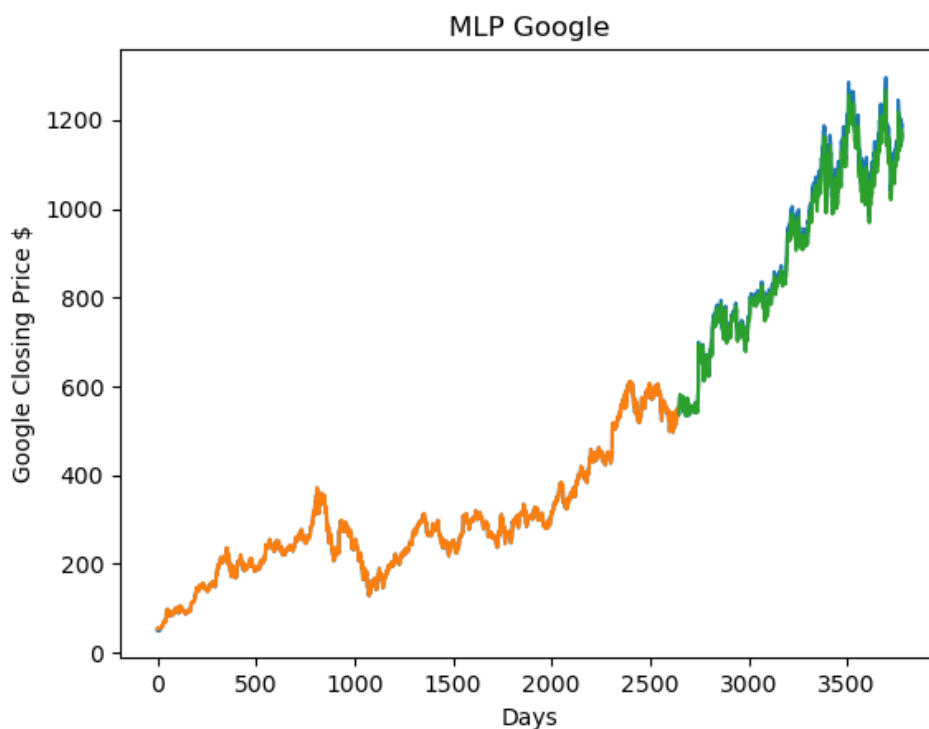
7	600	SGD	450.21	21.22
8	800	SGD	448.80	21.18
9	200	RMSPROP	277.69	16.66
10	400	RMSPROP	327.67	18.10
11	600	RMSPROP	338.76	18.41
<u>12</u>	<u>800</u>	<u>RMSPROP</u>	<u>246.49</u>	<u>15.70</u>

Οι διακυμάνσεις που παρατηρούνται στο συγκεκριμένο σύνολο δεδομένων (dataset) είναι πιο ευδιάκριτες σε σχέση με το προηγούμενο, με τον κατηγοριοποιητή (optimizer) που να σημειώνει το μικρότερο MSE να είναι κι εδώ ο RMSprop στα 800 epochs. Αντίθετα την χειρότερη επίδοση την καταγράφει ο κατηγοριοποιητής (optimizer) Sgd στα 200 epochs με MSE 8489.33.

Τα αποτελέσματα του τεστ 12 που καταγράφει το μικρότερο MSE παρουσιάζονται διαγραμματικά παρακάτω.



Η πραγματική χρονοσειρά της τιμής κλεισίματος της μετοχής της εταιρίας Google.



Η χρονοσειρά που παράγεται από τον MLP για την τιμή κλεισίματος της μετοχής της εταιρείας Google. Με πορτοκαλί απεικονίζονται οι παρατηρήσεις του training set και με πράσινο οι παρατηρήσεις του test set και με μπλε οι πραγματικές παρατηρήσεις.

Η διαγραμματική απεικόνιση του MLP για τη Google είναι ελαφρά χειρότερη από ότι του Facebook, κάτι που είναι απόλυτα λογικό, λόγω του μεγαλύτερου MSE. Αυτό αντικατοπτρίζεται και από την αμυδρή εμφάνιση της πραγματικής χρονοσειράς στο παραπάνω διάγραμμα, πίσω από τη χρονοσειρά που παράγει το μοντέλο.

3.4.3. Αποτελέσματα MLP για την Amazon

Τα αποτελέσματα του MLP στη χρονοσειρά της τιμής κλεισίματος της μετοχής της εταιρείας Amazon παρουσιάζονται στον παρακάτω πίνακα.

TEST	EPOCHS	OPTIMIZER	MSE	RMSE
1	200	ADAM	1404.05	37.47
2	400	ADAM	1440.20	37.95
3	600	ADAM	1370.97	37.03
4	800	ADAM	1549.80	39.37
5	200	SGD	770391	877.72
6	400	SGD	634376.61	796.48
7	600	SGD	468701.95	684.62

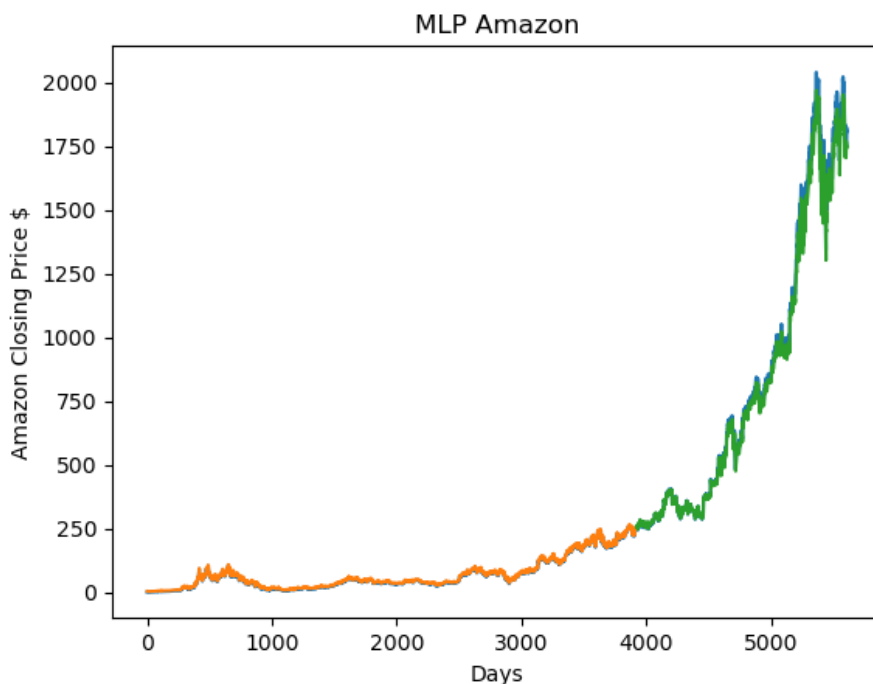
8	800	SGD	300930	548.47
9	200	RMSPROP	1571.81	39.65
10	400	RMSPROP	1470.37	38.35
<u>11</u>	<u>600</u>	<u>RMSPROP</u>	<u>1343.93</u>	<u>36.66</u>
12	800	RMSPROP	1443.56	37.99

Στο συγκεκριμένο σύνολο δεδομένων (dataset) παρατηρούνται τα μεγαλύτερα σφάλματα από τα τρία. Ο Rmsprop συνεχίζει να παράγει τα καλύτερα αποτελέσματα στο τεστ 11 με μικρότερο MSE 1343.93 στα 600 epochs, ακολουθεί ο Adam με 1370.97 στα 600 epochs, ενώ οι επιδόσεις του SGD είναι πολύ κακές σε όλο το φάσμα των δοκιμών.

Τα αποτελέσματα του τεστ 11 που καταγράφει το μικρότερο MSE παρουσιάζονται διαγραμματικά παρακάτω.



Η πραγματική χρονοσειρά της τιμής κλεισίματος της μετοχής της εταιρείας Amazon.



Η χρονοσειρά που παράγεται από τον MLP για την τιμή κλεισίματος της μετοχής της εταιρείας Amazon. Με πορτοκαλί απεικονίζονται οι παρατηρήσεις του training set και με πράσινο οι παρατηρήσεις του test set και με μπλε οι πραγματικές παρατηρήσεις.

3.5. Long-Short Term Memory Network (LSTMN)

Η δεύτερη αρχιτεκτονική τεχνητών νευρωνικών δικτύων που θα χρησιμοποιήσουμε για να αξιολογήσουμε την απόδοση της, ως την ικανότητα της να κάνει προβλέψεις σε χρονολογικά δεδομένα που αντιπροσωπεύουν τιμές μετοχών, είναι LSTMN. Το μοντέλο LSTMN που κατασκευάσαμε είναι ακριβώς το ίδιο με εκείνο του MLP, με μοναδικές διαφορές την ποσόστωση του συνόλου εκπαίδευσης (training set) και του συνόλου αξιολόγησης (test test) και ο αριθμός των παραδειγμάτων εκπαίδευσης ανά παρτίδα (batch size). Μετά από πολλές δοκιμές καταλήξαμε ότι το LSTMN χρειάζεται μεγαλύτερο όγκο δεδομένων προς εκπαίδευση και πιο συγκεκριμένα το 80% του συνόλου δεδομένων (dataset), ενώ το βέλτιστο batch size ισούται με 15.

3.5.1. Αποτελέσματα LSTMN για το Facebook

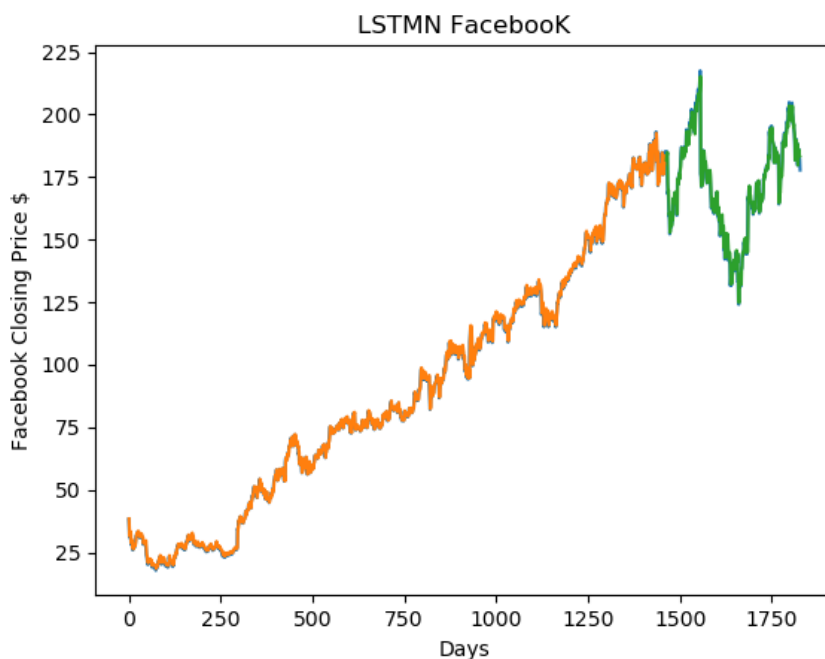
Τα αποτελέσματα του LSTMN στη χρονοσειρά της τιμής κλεισίματος της μετοχής της εταιρείας Facebook παρουσιάζονται στον παρακάτω πίνακα.

TEST	EPOCHS	OPTIMIZER	MSE	RMSE
1	200	ADAM	15.67	3.96
2	400	ADAM	15.87	3.98
3	600	ADAM	15.52	3.94

4	800	ADAM	16.32	4.04
5	200	SGD	24.49	4.95
6	400	SGD	20.78	4.56
7	600	SGD	19.20	4.38
8	800	SGD	18.22	4.27
9	200	RMSPROP	15.60	3.95
10	400	RMSPROP	15.70	3.96
11	600	RMSPROP	16.79	4.10
12	800	RMSPROP	20.52	4.53

Το χαμηλότερο MSE είναι 15.52 και προκύπτει από το τεστ 3 και τον κατηγοριοποιητή (optimizer) Adam στα 600 epochs. Οι διαφορές είναι πολύ μικρές με τους υπόλοιπους κατηγοριοποιητές (optimizers), ενώ τα πολύ χαμηλά MSE υποδηλώνουν την πολύ καλή λειτουργία του LSTMN στις προβλέψεις που παράγει με το συγκεκριμένο σύνολο δεδομένων (dataset).

Τα αποτελέσματα του τεστ 3 που καταγράφει το μικρότερο MSE παρουσιάζονται διαγραμματικά παρακάτω.



Η χρονοσειρά που παράγεται από τον LSTMN για την εταιρεία Facebook. Με πορτοκαλί απεικονίζονται οι παρατηρήσεις του training set και με πράσινο οι παρατηρήσεις του test set και με μπλε οι πραγματικές παρατηρήσεις.

Στο διάγραμμα αποτυπώνονται με ευκρίνεια οι πολύ καλές προβλέψεις του LSTMN με τις παραμέτρους (configurations) του τεστ 3. Οι προβλέψεις του μοντέλου ακολουθούν με πολύ μεγάλη ακρίβεια τις πραγματικές τιμές της χρονοσειράς.

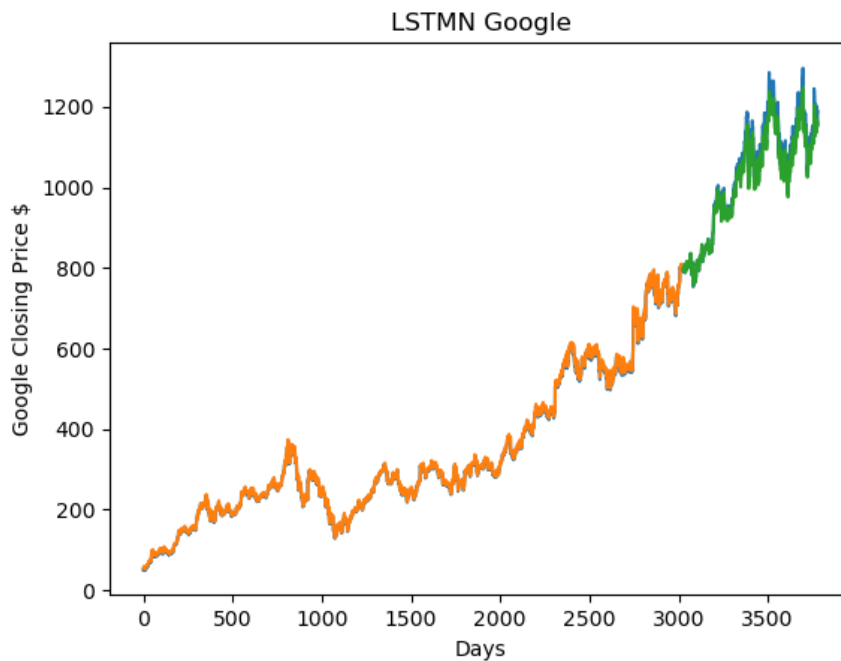
3.5.2. Αποτελέσματα LSTMN για τη Google

Τα αποτελέσματα του LSTMN στη χρονοσειρά της τιμής κλεισίματος της μετοχής της εταιρείας Google παρουσιάζονται στον παρακάτω πίνακα.

TEST	EPOCHS	OPTIMIZER	MSE	RMSE
1	200	ADAM	639.73	25.29
2	400	ADAM	633.49	25.17
3	600	ADAM	622.74	24.95
4	800	ADAM	848.98	29.14
5	200	SGD	4229.40	65.03
6	400	SGD	3855	62.09
7	600	SGD	3569.31	59.74
8	800	SGD	3342.72	57.82
9	200	RMSPROP	782.64	27.98
10	400	RMSPROP	796.05	28.21
11	600	RMSPROP	1107.28	33.28
<u>12</u>	<u>800</u>	<u>RMSPROP</u>	<u>548.53</u>	<u>23.42</u>

Το χαμηλότερο MSE είναι 548,53 και προκύπτει από το τεστ 12 και τον κατηγοριοποιητή (optimizer) RMSprop στα 800 epochs. Παρατηρούμε ότι το MSE είναι αρκετά μεγαλύτερο σε σχέση με το προηγούμενο σύνολο δεδομένων (dataset) σε όλους τους κατηγοριοποιητές (optimizers), με τον Sgd να καταγράφει τις χειρότερες επιδόσεις.

Τα αποτελέσματα του τεστ 12 που καταγράφει το μικρότερο MSE παρουσιάζονται διαγραμματικά παρακάτω.



Η χρονοσειρά που παράγεται από τον LSTMN για την εταιρεία Google. Με πορτοκαλί απεικονίζονται οι παρατηρήσεις του training set και με πράσινο οι παρατηρήσεις του test set και με μπλε οι πραγματικές παρατηρήσεις.

Στο διάγραμμα παρατηρούμε ότι οι προβλέψεις του LSTMN με τις βέλτιστες παραμέτρους του τεστ 12 είναι καλές αφού ακολουθούν με αρκετά καλή αναλογία τις τιμές των παρατηρήσεων της πραγματικής χρονοσειράς, αλλά είναι σαφώς χειρότερες με τις επιδόσεις του μοντέλου στο προηγούμενο σύνολο δεδομένων (dataset).

3.5.3. Αποτελέσματα LSTMN για την Amazon

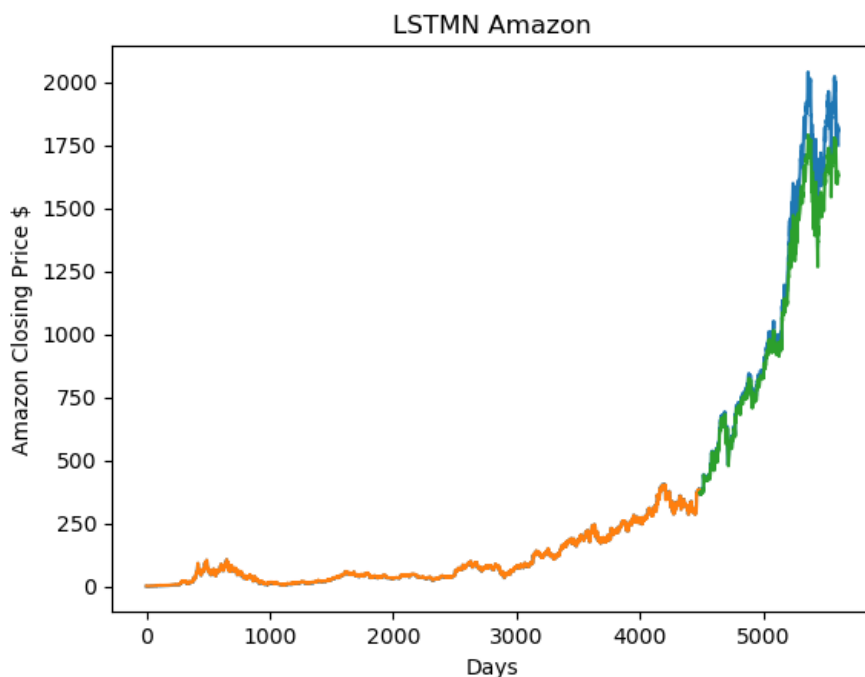
Τα αποτελέσματα του LSTMN στη χρονοσειρά της τιμής κλεισίματος της μετοχής της εταιρείας Amazon παρουσιάζονται στον παρακάτω πίνακα.

TEST	EPOCHS	OPTIMIZER	MSE	RMSE
1	200	ADAM	10449.73	102.22
2	400	ADAM	17255.57	131.36
3	600	ADAM	27448.93	165.68
4	800	ADAM	44221.19	210.29
5	200	SGD	567987.36	753.65
6	400	SGD	128281.25	358.16
7	600	SGD	39211.46	198.02

8	800	SGD	29707	172.36
9	200	RMSPROP	30325.64	174.14
10	400	RMSPROP	57796.07	240.41
11	600	RMSPROP	87949.48	296.56
12	800	RMSPROP	114440.91	338.29

Το καλύτερο αποτέλεσμα είναι στο τεστ 1 με το MSE να κυμαίνεται στο 10449.73, στα 200 epochs και με κατηγοριοποιητή (optimizer) τον Adam. Τα αποτελέσματα του LSTMN στο σύνολο δεδομένων (dataset) της Amazon είναι τα χειρότερα ανάμεσα στα τρία σύνολα δεδομένων (datasets), με το MSE να αυξάνεται κατακόρυφα. Επίσης ο Sgd συνεχίζει να δίνει τα χειρότερα αποτελέσματα και σε αυτό το σύνολο δεδομένων (dataset).

Τα αποτελέσματα του τεστ 1 που καταγράφει το μικρότερο MSE παρουσιάζονται διαγραμματικά παρακάτω.



Η χρονοσειρά που παράγεται από τον LSTMN για την εταιρεία Amazon. Με πορτοκαλί απεικονίζονται οι παρατηρήσεις του training set και με πράσινο οι παρατηρήσεις του test set και με μπλε οι πραγματικές παρατηρήσεις.

Στο διάγραμμα παρατηρούμε ότι ενώ ο LSTMN ενώ παράγει πολύ καλά αποτελέσματα στις αρχικές παρατηρήσεις του test set, μετά τις 5000 παρατηρήσεις αρχίζει να μην ακολουθεί σωστά την πορεία της πραγματικής χρονοσειράς. Αποτυγχάνει δηλαδή στη

σωστή πρόβλεψη των παρατηρήσεων που παρουσιάζουν τις υψηλότερες τιμές κλεισίματος, γεγονός που συνάδει με τη σημαντική αύξηση του MSE.

3.6. Convolutional Neural Network (CNN)

Η αρχιτεκτονική των CNN διαφέρει από τις αρχιτεκτονικές των δυο προηγούμενων τεχνητών νευρωνικών δικτύων. Επομένως το μοντέλο που θα χρησιμοποιήσουμε θα έχει ορισμένες διαφορές σε σχέση με τα προηγούμενα. Το ακολουθιακό CNN θα έχει τρία μονοδιάστατα συνελκτικά (convolutional) επίπεδα για την εισαγωγή των δεδομένων, ένα μονοδιάστατο συγκεντρωτικό επίπεδο (pooling) για τον διαχωρισμό των πιο σημαντικών στοιχείων εξόδου, ένα πολύ πυκνό (dense) από άποψη νευρώνων επίπεδο για την ερμηνεία των χαρακτηριστικών που έχουν εξαχθεί από το μοντέλο και τέλος ένα επίπεδο που θα συγκεντρώνει το πολυεπίπεδο χαρακτηριστικά που έχουν παραχθεί σε ένα ενιαίο διάνυσμα.

Μετά από αρκετές δοκιμές, καταλήξαμε ότι ο βέλτιστος αριθμός των φίλτρων για τα συνελκτικά επίπεδα (convolutional layers) είναι το 64, ενώ από τη στιγμή που τα αρχικά δεδομένα που θα εισάγουμε στο μοντέλο είναι μονοδιάστατα ορίσαμε το μέγεθος των κέρνελ (kernels) ίσο με τη μονάδα. Τα δεδομένα, όπως και στα δύο προηγούμενα μοντέλα, θα πρέπει να έχουν την μορφή numpy array και το σύνολο δεδομένων (dataset) θα χωριστεί σε ποσοστό 80% ως σύνολο εκπαίδευσης (training set) και 20% ως σύνολο αξιολόγησης (test set). Επίσης το χρονικό βήμα (timestep) θα είναι κι αυτό ίσο με τη μονάδα και για κάθε σύνολο δεδομένων (dataset) θα οριστεί μία μεταβλητή k , η οποία θα αντιπροσωπεύει τον αριθμό των παρατηρήσεων που αντιστοιχούν στο εκάστοτε σύνολο εκπαίδευσης (training set). Ακόμη ο αριθμός των νευρώνων στο προτελευταίο επίπεδο θα είναι ίσος με 100 και σε όλα τα συνελκτικά επίπεδα (convolutional layers) θα χρησιμοποιηθεί η relu ως συνάρτηση ενεργοποίησης.

Η υπόλοιπη πειραματική διαδικασία για την αξιολόγηση του μοντέλου παραμένει ίδια ως προς τις παραμέτρους (configurations). Θα χρησιμοποιήσουμε τους ίδιους ακριβώς κατηγοριοποιητές (optimizers) με τον ίδιο αριθμό epochs για τη διεξαγωγή και παρουσίαση του ίδιου αριθμού τεστ και το μέτρο σύγκρισης για κάθε δοκιμή που θα πραγματοποιήσουμε θα είναι το MSE. Η μοναδική αλλαγή θα είναι ως προς τη διαγραμματική παρουσίαση των αποτελεσμάτων, η οποία θα επικεντρωθεί στα χρονικά διαστήματα των παρατηρήσεων του συνόλου αξιολόγησης (test set) και οι οποίες θα προβληθούν σε αντιπαραβολή με τις τιμές των αντίστοιχων παρατηρήσεων του αρχικού συνόλου δεδομένων (dataset).

3.6.1. Αποτελέσματα CNN για το Facebook

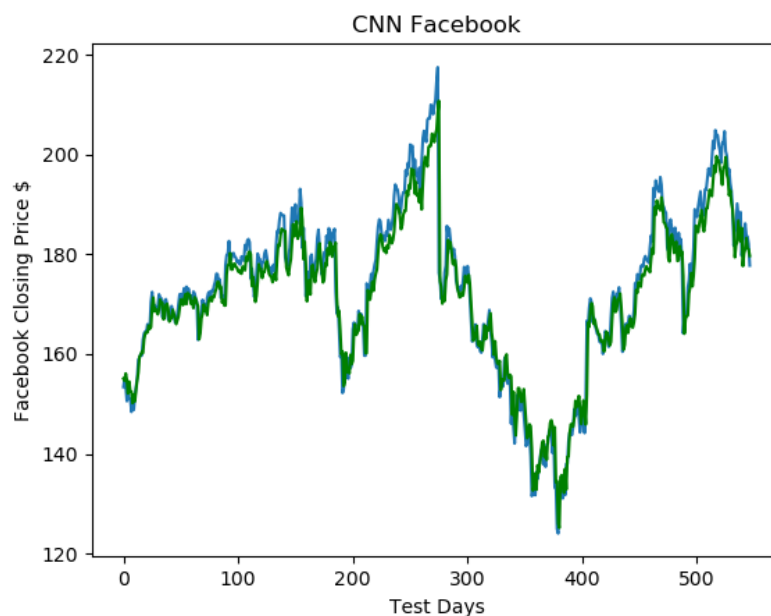
Τα αποτελέσματα του CNN στη χρονοσειρά της τιμής κλεισίματος της μετοχής της εταιρείας Facebook παρουσιάζονται στον παρακάτω πίνακα.

TEST	EPOCHS	OPTIMIZER	MSE	RMSE
1	200	ADAM	22.71	4.76
2	400	ADAM	24.60	4.96
3	600	ADAM	23.02	4.79

4	800	ADAM	44.58	6.67
5	200	SGD	713.43	26.71
6	400	SGD	668.35	28.85
7	600	SGD	656.26	25.61
8	800	SGD	652.96	25.55
<u>9</u>	<u>200</u>	<u>RMSPROP</u>	<u>16.57</u>	<u>4.07</u>
10	400	RMSPROP	34.61	5.88
11	600	RMSPROP	44.07	6.63
12	800	RMSPROP	72.30	8.50

Το χαμηλότερο MSE επιτυγχάνεται στο τεστ 9 με τη χρήση του κατηγοριοποιητή (optimizer) RMSprop στα 200 epochs. Πολύ καλά αποτελέσματα παράγει και ο κατηγοριοποιητής (optimizer) Adam, ενώ τα αποτελέσματα του κατηγοριοποιητή (optimizer) Sgd είναι απογοητευτικά σε σύγκριση με των άλλων δύο. Παρατηρούμε ότι στον Adam και στον RMSprop μετά τα 200 epochs παρουσιάζεται ως ένα βαθμό το φαινόμενο του overfitting, το οποίο είναι εντονότερο στον Adam.

Τα αποτελέσματα του τεστ 9 οπτικοποιούνται στο παρακάτω διάγραμμα.



Το χρονικό στιγμιότυπο του τέλους της χρονοσειράς για την τιμή κλεισίματος της μετοχής της εταιρείας Facebook. Με πράσινο χρώμα συμβολίζονται οι προβλέψεις που παρήγαγε το μοντέλο και με μπλε χρώμα οι πραγματικές τιμές.

Στο διάγραμμα αντικατοπτρίζεται η καλή επίδοση του μοντέλου για το συγκεκριμένο σύνολο δεδομένων (dataset), με τις προβλέψεις του να ακολουθούν τις πραγματικές

τιμές της χρονοσειράς με αρκετά πιστό τρόπο. Το μοντέλο χάνει ως ένα βαθμό τις υψηλότερες τιμές της μετοχής που σημειώνονται ανάμεσα στις 200 με 300 ημέρες και κοντά στις 500 ημέρες, όχι όμως και τις χαμηλότερες τιμές που σημειώνονται κοντά στις 400 ημέρες.

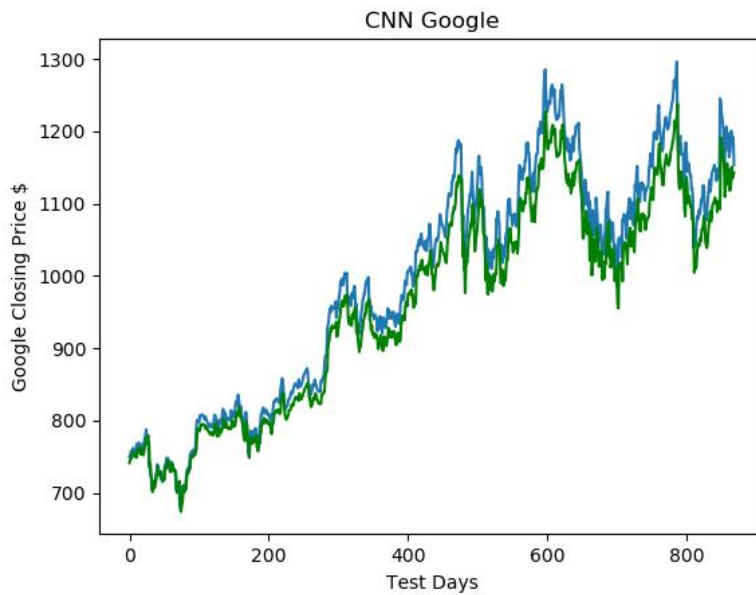
3.6.2. Αποτελέσματα CNN για τη Google

Τα αποτελέσματα του CNN στη χρονοσειρά της τιμής κλεισίματος της μετοχής της εταιρείας Google παρουσιάζονται στον παρακάτω πίνακα.

TEST	EPOCHS	OPTIMIZER	MSE	RMSE
1	200	ADAM	3904.95	62.48
2	400	ADAM	13755.1	117.28
3	600	ADAM	4023.93	63.43
4	800	ADAM	14418.61	120.07
5	200	SGD	87606.04	295.98
6	400	SGD	84348.09	290.42
7	600	SGD	82894.44	287.91
8	800	SGD	82054.12	286.45
9	200	RMSPROP	2324.29	48.21
10	400	RMSPROP	2275.27	47.69
11	600	RMSPROP	2165.71	46.53
<u>12</u>	<u>800</u>	<u>RMSPROP</u>	<u>1354.19</u>	<u>36.79</u>

Το χαμηλότερο MSE είναι 1354.19 και επιτυγχάνεται στο τεστ 12 με τη χρήση του κατηγοριοποιητή (optimizer) RMSprop στα 800 epochs. Ο Adam με MSE 3904.95 στο τεστ 1 να ακολουθεί ενώ τα αποτελέσματα του optimizer Sgd είναι για άλλη μία φορά χειρότερα σε σύγκριση με των άλλων δύο.

Τα αποτελέσματα του τεστ 12 οπτικοποιούνται στο παρακάτω διάγραμμα.



Το χρονικό στιγμιότυπο του τέλους της χρονοσειράς για την τιμή κλεισίματος της μετοχής της εταιρείας Google. Με πράσινο χρώμα συμβολίζονται οι προβλέψεις που παρήγαγε το μοντέλο και με μπλε χρώμα οι πραγματικές τιμές.

Τα μεγαλύτερα σφάλματα που παράγει το μοντέλου του CNN στο σύνολο δεδομένων (dataset) της Google, σε σχέση με το προηγούμενο σύνολο δεδομένων (dataset), αποτυπώνονται στο παραπάνω διάγραμμα με αρκετά ευδιάκριτο τρόπο. Οι προβλέψεις του μοντέλου ακολουθούν σωστά τις τιμές των πραγματικών παρατηρήσεων ως προς τις τάσεις που παρουσιάζουν, αλλά υπολείπονται σε ακρίβεια σχεδόν στο σύνολο του χρονικού διαστήματος.

3.6.3. Αποτελέσματα CNN για την Amazon

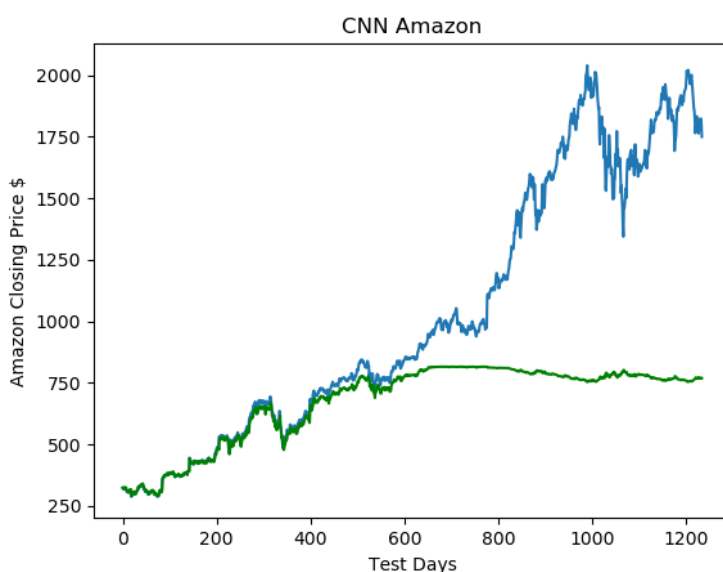
Τα αποτελέσματα του CNN στη χρονοσειρά της τιμής κλεισίματος της μετοχής της εταιρείας Amazon παρουσιάζονται στον παρακάτω πίνακα.

TEST	EPOCHS	OPTIMIZER	MSE	RMSE
1	200	ADAM	310582.94	557.29
2	400	ADAM	369978.56	608.25
3	600	ADAM	422991.44	650.37
4	800	ADAM	486737.6	697.66
5	200	SGD	344491.84	586.93
6	400	SGD	344556.2	586.98
7	600	SGD	344642.9	587.06

8	800	SGD	344598.06	587.02
9	200	RMSPROP	672792.4	820.23
10	400	RMSPROP	691995.8	831.86
11	600	RMSPROP	691493	831.56
12	800	RMSPROP	686713	828.68

Παρατηρούμε ότι σε όλα τα παραπάνω τεστ τα παραγόμενα σφάλματα είναι υπερβολικά μεγάλα. Όλοι οι κατηγοριοποιητές (optimizers) έχουν πολύ κακή απόδοση, παρόλα αυτά τη μη χείριστη την παρουσιάζει ο Adam στο τεστ 1, με MSE 210582 στα 200 epochs. Είναι προφανές ότι τα τόσο μεγάλα και καθολικά σφάλματα οδηγούν στο συμπέρασμα ότι το μοντέλο θα αδυνατεί να προβλέψει σωστά σε πολύ μεγάλο βαθμό.

Τα αποτελέσματα του τεστ 1 οπτικοποιούνται στο παρακάτω διάγραμμα.



Το χρονικό στιγμιότυπο του τέλους της χρονοσειράς για την τιμή κλεισίματος της μετοχής της εταιρείας Amazon. Με πράσινο χρώμα συμβολίζονται οι προβλέψεις που παρήγαγε το μοντέλο και με μπλε χρώμα οι πραγματικές τιμές.

Η μικρή προβλεπτική ικανότητα του μοντέλου, όπως αποτυπώθηκε και από τα πολύ μεγάλα MSE του προηγούμενου πίνακα, είναι πασιφανής στο παραπάνω διάγραμμα. Μετά τις 600 ημέρες το μοντέλο χάνει εντελώς την πορεία της πραγματικής χρονοσειράς και απλώς δίνει μία σχεδόν σταθερή τιμή κλεισίματος για τις υπόλοιπες ημέρες.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Αξιολογώντας τα αποτελέσματα της πειραματικής διαδικασίας οδηγούμαστε στο ξεκάθαρο συμπέρασμα ότι η αρχιτεκτονική των τεχνητών νευρωνικών δικτύων που παράγει τα καλύτερα αποτελέσματα ως προς την παραγωγή προβλέψεων σε χρονολογικά δεδομένα μετοχών, είναι ο Multilayer Perceptron. Το συμπέρασμα αυτό αποδεικνύεται από τις επιδόσεις του μοντέλου που βασίστηκε στη συγκεκριμένη αρχιτεκτονική τεχνητών νευρωνικών δικτύων και το οποίο παρήγαγε σταθερά το μικρότερο μέσο τετραγωνικό σφάλμα (mse) σε όλα τα σύνολα δεδομένων (dataset) που χρησιμοποιήσαμε. Το αμέσως επόμενο μοντέλο από τον MLP ως προς τις επιδόσεις, είναι αυτό που βασίστηκε στην αρχιτεκτονική των Long-Short Term Memory Neural Networks, ενώ τη χειρότερη επίδοση την κατέγραψε το μοντέλο που βασίστηκε στην αρχιτεκτονική των Convolutional Neural Networks.

Ως προς τους κατηγοριοποιητές (optimizers), τα καλύτερα αποτελέσματα τα παρήγαγε ο RMSprop, με τον Adam να καταγράφει σχεδόν εφάμιλλες ή και καλύτερες επιδόσεις σε ορισμένα tests. Οι επιδόσεις του Sgd μπορούν να χαρακτηριστούν σταθερά πολύ κακές, αφού στην καλύτερη των περιπτώσεων παρήγαγε σφάλματα σχεδόν διπλάσιου μεγέθους σε σχέση με του άλλους δύο κατηγοριοποιητές (optimizers).

Επίσης είναι πολύ σημαντικό να αναφέρουμε τις μεγάλες διακυμάνσεις στα σφάλματα ανάμεσα στα τρία σύνολα δεδομένων (datasets). Τα μικρότερα σφάλματα παρατηρήθηκαν στο σύνολο δεδομένων (dataset) της εταιρείας Facebook, ενώ τα μεγαλύτερα παρατηρήθηκαν στο σύνολο δεδομένων (dataset) της εταιρείας Amazon. Το γεγονός αυτό είναι απολύτως λογικό, αφού το σύνολο δεδομένων (dataset) του Facebook είχε τις λιγότερες παρατηρήσεις και τη μικρότερη διακύμανση από τα τρία σύνολα δεδομένων (datasets), ενώ το σύνολο δεδομένων (dataset) της Amazon είχε τις περισσότερες παρατηρήσεις και τη μεγαλύτερη διακύμανση από τα τρία σύνολα δεδομένων (datasets). Επομένως το γεγονός αυτό μας οδηγεί στο συμπέρασμα ότι ένα μοντέλο τεχνητού νευρωνικού δικτύου που μπορεί να παράγει πολύ καλά αποτελέσματα για ένα συγκεκριμένο σύνολο χρονολογικών δεδομένων, να μην μπορεί να ανταπεξέλθει εξίσου καλά σε άλλο σύνολο και να χρειάζεται παραμετροποίηση ή και αντικατάσταση με κάποιο άλλο πιο λειτουργικό.

Εν κατακλείδι τα τεχνητά νευρωνικά δίκτυα, αν και άρχισαν να αναπτύσσονται ραγδαία για να αντιμετωπίσουν προβλήματα επεξεργασίας δεδομένων εικόνας και ήχου, μπορούν να χρησιμοποιηθούν με επιτυχία στην επεξεργασία και πρόβλεψη χρηματιστηριακών χρονοσειρών. Η εξέλιξη αυτή είναι πολύ σημαντική γιατί δίνει επιπλέον εργαλειακές επιλογές για τη μελέτη τέτοιων φαινομένων οικονομικού τύπου, πέραν των παραδοσιακών μοντέλων παλινδρόμησης, οι οποίες μπορούν να εφαρμοστούν ευκολότερα και πιο άμεσα πολλές φορές. Αυτό προκύπτει από το γεγονός ότι τα μοντέλα των τεχνητών νευρωνικών δικτύων παρήγαγαν καλά αποτελέσματα χωρίς την εφαρμογή κλασικών μεθόδων επεξεργασίας των δεδομένων, όπως η διαδικασία της απαλοιφής των χαρακτηριστικών της χρονοσειράς (decomposition), οι οποίες είναι απαραίτητες σε προγενέστερες μεθόδους.

ΜΕΛΛΟΝΤΙΚΟΙ ΣΤΟΧΟΙ

Αν και ο γνωστικός στόχος της συγκεκριμένης διπλωματικής εργασίας σε γενικές γραμμές επετεύχθη, σίγουρα θα μπορούσαν να γίνουν ορισμένα παραπάνω πράγματα σε μελλοντικό χρόνο. Αρχικά θα μπορούσαμε να εφαρμόσουμε τη μέθοδο της απαλοιφής των χαρακτηριστικών των χρονοσειρών (decomposition) για να βελτιώσουμε ακόμη περισσότερο τα αποτελέσματα των μοντέλων που δημιουργήσαμε. Ακόμη θα μπορούσαμε σε αρκετά μακροπρόθεσμο ορίζοντα να χρησιμοποιήσουμε και τις υπόλοιπες μεταβλητές που υπήρχαν στο αρχικό σύνολο δεδομένων (dataset) και να προχωρήσουμε στην κατασκευή μοντέλων τεχνητών νευρωνικών δικτύων για την ανάλυση πολυμεταβλητών (multivariate) χρονοσειρών. Τέλος θα ήταν ίσως εφικτή η διεξαγωγή περισσότερων πειραμάτων με τη μοντελοποίηση περισσότερων αρχιτεκτονικών τεχνητών νευρωνικών δικτύων.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Sodhi, Pinky and Awasthi, Naman and Sharma, Vishal, Introduction to Machine Learning and Its Basic Application in Python (January 6, 2019). Proceedings of 10th International Conference on Digital Strategies for Organizational Success.
- [2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville (The MIT Press, November 18 2016). Deep Learning, chapter 5.
- [3] Αικατερίνη Γεωργούλη (2015 Αθήνα). Μηχανική Μάθηση, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών, κεφάλαιο 4.
- [4] Tom M. Mitchell (March 1, 1997). Machine Learning, McGraw-Hill Science/Engineering/Math, chapter 4.
- [5] Girish Kumar Jha. Artificial Neural Networks, Indian Agricultural Research Institute PUSA, New Delhi-110 012.
- [6] Gaurang Panchal, Amit Ganatra, Y P Kosta and Devyani Panchal, Behaviour Analysis of Multilayer Perceptrons with Multiple Hidden Neurons and Hidden Layers (April 2011). International Journal of Computer Theory and Engineering Vol. 3, No. 2.
- [7] Marius-Constantin Popescu, Valentina E. Balas, Liliana Perescu-Popescu, Nikos Mastorakis, Multilayer Perceptron and Neural Networks (July 2009). Article in WSEAS Transactions on Circuits and Systems, Issue 7, Volume 8.
- [8] Saptarshi Sengupta, Sanchita Basak, Pallabi Saikia, Sayak Paul, Vasilios Tsalavoutis, Frederick Atiah, Vadlamani Ravi, Alan Peters, A Review of Deep Learning with Special Emphasis on Architectures, Applications and Recent Trends (6 February, 2020). Article number 105596 on Knowledge-Based Systems.
- [9] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu and Fuad E. Alsaadi, A survey of deep neural network architectures and their applications (December 2016). Article in Neurocomputing Vol. 234.
- [10] Indraneel Pole, An Advance Seminar Paper on Restricted Boltzmann Machines for course Computational Intelligence, Matrikel Nr: 1063525.
- [11] Asja Fischer and Christian Igel, An Introduction to Restricted Boltzmann Machine (September 3-6, 2012). Conference Paper in 17th Iberoamerican Congress, CIARP 2012, Buenos Aires, Argentina.
- [12] Classification with Deep Belief Networks, HussamHebbo Jae Won Kim.
- [13] Azhagammal Alagarsamy, Bharathiyar, Dr. K. Ruba Soundar, A Survey Paper on Deep Belief Network For Big Data (September-October 2018), International Journal of Computer Engineering & Technology (IJCET) Volume 9, Issue 5.
- [14] Andreas Karathanasopoulos, University of Dubai, Modelling and trading commodities with a new deep belief network (May 18 2017). Economics and Business Letters 6(2), 28-34, 2017.

- [15] Lei Le, Andrew Patterson and Martha White, Supervised autoencoders: Improving generalization performance with unsupervised regularizers (December 2018). Conference Paper in Proceedings of the 32nd International Conference on Neural Information Processing.
- [16] Shihao Gu, Bryan Kelly and Dacheng Xiu, Autoencoder Asset Pricing Models (September 30, 2019). The University of Chicago Booth School of Business, Working Paper No. 19-24.
- [17] Saad Albawi, Tareq Abed Mohammed and Saad Al-Zawi, Understanding of Convolutional Neural Network (August 2017). Conference Paper in The International Conference on Engineering and Technology at Antalya Turkey.
- [18] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Li Wang, Gang Wang, Jianfei Cai, Tsuhan Chen, Recent Advances in Convolutional Neural Networks (May 2018). Research article in Pattern Recognition Vol. 77.
- [19] Rikiya Yamashita^{1,2} & Mizuho Nishio^{1,3} & Richard Kinh Gian Do² & Kaori Togashi, Convolutional neural networks: an overview and application in radiology (22 June, 2018). Article in Insights into Imaging Vol. 9.
- [20] Zachary C. Lipton, John Berkowitz, Charles Elkan, A Critical Review of Recurrent Neural Networks for Sequence Learning (17 October, 2015). Article in ArXiv.
- [21] Hasim Sak, Andrew Senior, Françoise Beaufays, Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling. Conference Article in Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH 2014.
- [22] Felix Gers (Thesis Number 2366-2001 EPFL). Long Short-Term Memory in Recurrent Neural Networks.
- [23] Gintaras v. Reklaitis, Athanasios G. Tsiрукis, Manoel F. Tenorio, Generalized hopfield networks and nonlinear optimization (January 1989). Article in NIPS'89: Proceedings of the 2nd International Conference on Neural Information Processing.
- [24] Narotam Singh, Amita Kapoor, Cloud Hopfield neural network: Analysis and simulation (August 2015). Conference paper in ICACCI 2015.
- [25] R. Rojas: Neural Networks A Systematic Introduction, Springer-Verlag, Berlin 1996, Chapter 13.
- [26] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative Adversarial Nets (December 2014). Article in NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2.
- [27] Christian Bodnar-University of Manchester, Text to Image Synthesis Using Generative Adversarial Networks (May, 2018). A final year report submitted for the degree of BSc Computer Science.
- [28] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio, Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling (December 11

2014). Paper presented in NIPS 2014 Deep Learning and Representation Learning Workshop.

[29] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, Yoshua Bengio, Gated Feedback Recurrent Neural Networks (July 2015). Article in ICML'15: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37.

[30] Peter J. Brockwell, Richard A. Davis, Introduction to Time Series and Forecasting, second edition Springer, Chapter 1.

[31] Ratnadip Adhikari, R. K. Agrawal. An Introductory Study on Time Series Modeling and Forecasting, LAMPERT Academic Publishing, Chapter 2.

[32] Estela Bee Dagum, Time Series Modeling and Decomposition (2010). Article in Statistica Vol. 70, No 4.

[33] Björn Fischer, Decomposition of Time Series Comparing Different Methods in Theory and Practice, Version 2.1 March/April 1995 EUROSTAT, Chapter 3.

[34] Γεωργία Καλαμβόκη, Μέθοδοι Πρόβλεψης Χρονοσειρών στην Ελληνική Οικονομία (Φλεβάρης 2017). Μεταπτυχιακή Διπλωματική Εργασία για το Δ.Π.Μ.Σ. Μαθηματικά των Υπολογιστών και των Αποφάσεων.

[35] Andrius Buteikis, Time Series with Trend and Seasonality components, lecture at Faculty of Mathematics and Informatics, Institute of Applied Mathematics, Vilniaus Universitetas.

[36] Bashirahamad Fardin Momin, Gaurav Chavan, Univariate Time Series Models for Forecasting Stationary and Non-stationary Data: A Brief Review. Conference Paper in Smart Innovation, March 2017.

[37] Nesreen K. Ahmed, Amir F. Atiya, Neamat El Gayar, Hisham El-Shishiny, An Empirical Comparison of Machine Learning Models for Time Series Forecasting (August 2010). Article in Econometrics Reviews Vol.29(5-6).

[38] Bohdan M. Pavlyshenko, Machine-Learning Models for Sales Time Series Forecasting (18 January, 2019). This paper is an extended version of conference paper: Using Stacking Approaches for Machine Learning Models. In Proceedings of the 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP).

[39] Hansika Hewamalage, Christoph Bergmeir, Kasun Bandara, Recurrent Neural Networks for Time Series Forecasting: Current Status and Future Directions. Paper published in ArXiv on September 02 2019.

[40] <https://www.kaggle.com/abdullahmu/big-five-stocks>

[41] G. Peter Zang, Min Qi, Neural network forecasting for seasonal and trend time series (16 January 2005). Article in European Journal of Operational Research Vol. 160, Issue 2.

[42] Kasun Bandara, Christoph Bergmeir, Hansika Hewamalage, LSTM-MSNet: Leveraging Forecasts on Sets of Related Time Series with Multiple Seasonal Patterns. Paper published in ArXiv on September 02 2019.

[43] <https://keras.io/getting-started/sequential-model-guide/>

[44] <https://keras.io/models/model/>