



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Πολυτεχνική Σχολή
Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Διπλωματική Εργασία

**Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές
βαθιάς μάθησης σε δεδομένα μεγάλου όγκου**

Τριπολίτης Ιωάννης-Νικόλαος
AM:1062662

Επιβλέπων :	Σιούτας Σπυρίδων,	Καθηγητής
Συνεπιβλέποντες:	Μακρής Χρήστος,	Αναπληρωτής Καθηγητής
	Τσώλης Δημήτρης,	Αναπληρωτής Καθηγητής

Πάτρα, 2023

**Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε
δεδομένα μεγάλου όγκου**

**Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε
δεδομένα μεγάλου όγκου**

© Copyright συγγραφής Τριπολίτης Ιωάννης-Νικόλαος, 2023

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Ηλεκτρονικών
Υπολογιστών &

Πληροφορικής του Πανεπιστημίου Πατρών δεν υποδηλώνει απαραίτητως και
αποδοχή των απόψεων του

συγγραφέα εκ μέρους του Τμήματος

**Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε
δεδομένα μεγάλου όγκου**

Ευχαριστίες

Καταρχάς θα ήθελα να ευχαριστήσω την οικογένεια μου για την υποστήριξη που μου παρείχαν όλα αυτά τα χρόνια χωρίς τους οποίους είμαι βέβαιος δεν θα είχα καταφέρει απολύτως τίποτα. Επίσης θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Σπυρίδωνα Σιούτα που μου επέτρεψε να αναλάβω αυτήν την διπλωματική εργασία καθώς επίσης και τον Καρρά Αριστείδη για την συνεχή βοήθεια που μου παρείχε καθ' όλη τη διάρκεια της εκπόνησης της διπλωματικής μου εργασίας. Τέλος θα ήθελα να ευχαριστήσω και τους φίλους μου, για την ψυχολογική υποστήριξη και συμβουλευτική τους συμμετοχή.

**Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε
δεδομένα μεγάλου όγκου**

Περίληψη

Τα στατιστικά μοντέλα σε συνδυασμό με την μηχανική μάθηση έχουν απόκτηση μεγάλη σημασία στην σύγχρονη ζωή αφού πλέον είναι πανταχού παρών. Σε αυτή την διπλωματική έχει γίνει μια προσπάθεια επίλυσης του προβλήματος πρόβλεψης τιμών χρησιμοποιώντας ως δεδομένα τις τιμές διάφορων κρυπτονομισμάτων χειρίζοντας τις ως μια χρονοσειρά χρησιμοποιώντας τις για την κατασκευή και εκπαίδευση βαθιών νευρωνικών δικτύων για πρόβλεψη των μελλοντικών τους τιμών. Στην αρχή δίνεται μια εξήγηση το τι είναι οι χρονοσειρές και οι διάφοροι τρόποι που μπορείς να τις προβλέψεις καθώς επίσης και το γιατί καταλήξαμε στην χρήση βαθιών νευρωνικών δικτύων για τα συγκεκριμένα δεδομένα, στην συνέχεια αναλύονται τα νευρωνικά δίκτυα πιο συγκεκριμένα τα mlp, rnn, lstm και lstm διπλής κατεύθυνσης ως εργαλεία τα οποία είναι στη διάθεση των πιο σύγχρονων τεχνολογιών αιχμής και χρησιμοποιούνται καθημερινά για διάφορα πράγματα, όπως και η πρόβλεψη του χρηματιστήριου ή και των κρυπτονομισμάτων. Έπειτα από την αναλυτική παρουσίαση των παραπάνω θα πραγματοποιηθεί μια ανάλυση πάνω στις χρονομέτρες των κρυπτονομισμάτων για την εξαγωγή απαραίτητων συμπερασμάτων για την κατασκευή των βαθιών νευρωνικών δικτύων τα οποία θα χρησιμοποιηθούν για τις προβλέψεις τους. Στο τέλος θα γίνει μια σύγκριση των προβλέψεων του κάθε νευρωνικού δικτύου.

Λέξεις – Κλειδιά :

Νευρωνικά Δίκτυα, Εξόρυξη δεδομένων, Πρόβλεψη χρονοσειρών, recurrent neural network, long short-term memory network, κρυπτονομίσματα

**Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε
δεδομένα μεγάλου όγκου**

Abstract

Statistical models combined with machine learning have gained importance when they have become ubiquitous in modern life. This diploma thesis has a target to provide a solution to the price prediction problem using as a dataset cryptocurrencies treating them like time series and using them to create and teach deep neural networks for forecasting their prices. In the first part of the dissertation an introduction to time series and the procedure of forecasting it is given and why in the end deep neural networks were chosen as the tool that best fits the job of doing so, Neural networks in particular mlp, rnn lstm and last but not least bi-lstm are then analyzed as tools that are at the disposal of the most modern cutting-edge technologies and are used daily for lots of things, like the forecasting of stocks and cryptocurrencies. After an analytical presentation of the above an exploratory data analysis is done on the time series to best discern how to build the deep neural networks that are going to be used for the forecasting. In the end a comparison is made between all the predictions of the deep neural networks.

Keywords:

Neural networks, Data mining, Time Series forecasting, recurrent neural network, long short-term memory network, cryptocurrencies

**Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε
δεδομένα μεγάλου όγκου**

Table of contents

Contents

Chapter 1 Εισαγωγή.....	19
Chapter 2 Χρονοσειρές	20
2.1 Βασικά Χαρακτηριστικά μια χρονοσειράς.....	20
2.1.1 Τι είναι μια χρονοσειρά	20
2.1.2 Από τι απαρτίζεται μια χρονοσειρά.....	20
2.1.3 Στασιμότητα.....	21
2.2 Η τεχνική της ολίσθησης παραθύρου.....	23
2.3 Στατιστική ανάλυση	25
2.4 Στρατηγικές προβλέψεις χρονοσειρών	26
2.4.1 Αφελής πρόβλεψη	26
2.4.2 Η οικογένεια των μοντέλων Arima	27
2.4.3 Εκθετική εξομάλυνση	29
2.4.4 Γραμμική παλινδρόμηση	30
2.4.5 Νευρωνικά δίκτυα	32
Chapter 3 Βαθιά νευρωνικά δίκτυα	33
3.1 Εισαγωγή.....	33
3.2 Τεχνητό Νευρωνικό Δίκτυο.....	34
3.2.1 Τι είναι ένας νευρώνας.....	34
3.2.2 Τι είναι ένα βαθύ νευρωνικό δίκτυο και πόες οι διάφορες του με ένα απλό	35
3.3 Εκπαίδευση τεχνητού νευρωνικού δικτύου	36
3.3.1 Τύποι μηχανικής μάθησης.....	36
3.3.2 Αλγόριθμοι βελτιστοποίησης	37
3.3.3 Υπερεκπαίδευση και υποεκπαίδευση	40
3.4 Αρχιτεκτονικές νευρωνικών δικτύων.....	40
3.4.1 Πολυεπίπεδα Perceptrons (multilayer Perceptrons).....	41
3.4.2 Επαναλαμβανόμενο νευρωνικό δίκτυο (RNN)	42
3.4.3 Νευρωνικά Δίκτυα Μακράς Βραχύχρονης Μνήμης (LSTM)	43
3.4.4 Bidirectional LSTMS	46
3.5 Υπερπαραμέτροι.....	47
3.5.1 Η συνάρτηση ενεργοποίησης	47

**Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε
δεδομένα μεγάλου όγκου**

3.5.2 Τεχνικές κανονικοποίησης	52
3.5.3 Συνάρτηση σφάλματος	54
Chapter 4 Ανάλυση και επεξεργασία δεδομένων εισόδου	58
4.1 Εισαγωγή.....	58
4.2 Βιβλιοθήκες(modules) που χρησιμοποιήθηκαν για επεξεργασία και ανάλυση.....	60
4.3 Ανάλυση δεδομένων.....	61
4.4 Επεξεργασία δεδομένων	66
Chapter 5 Πρόβλεψη χρονοσειρών	69
5.1 Εισαγωγή.....	69
5.1.1 Tensors.....	69
5.1.2 Tensorflow	70
5.2 Αρχιτεκτονικές νευρωνικών δικτύων.....	71
5.2.1 MLP	72
5.2.2 LSTM.....	74
5.2.3 Bilstm	76
5.3 Αποτελέσματα και συγκρίσεις.....	78
Chapter 6 Συμπεράσματα & προτάσεις	80

**Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε
δεδομένα μεγάλου όγκου**

Table of figures

Εικόνα 2.1 Οι φάσεις μια χρονοσειράς	21
Εικόνα 2.2 Οπτικοποίηση του λευκού θορύβου ως χρονοσειρά	22
Εικόνα 2.3 ACF και PACF για κατανάλωση καυτού νερού	25
Εικόνα 2.4 Παράδειγμα πρόβλεψης του αφελούς μοντέλου	27
Εικόνα 2.5 Παράδειγμα πρόβλεψης ενός Sarima(Seasonal auto-regressive intergrated moving average model).....	29
Εικόνα 2.6 Παράδειγμα πρόβλεψης εκθετικής εξομάλυνσης	30
Εικόνα 2.7 Παράδειγμα πρόβλεψης γραμμικής παλινδρόμησης.....	32
Εικόνα 3.1 Η οικογένεια της τεχνητής νοημοσύνης.....	33
Εικόνα 3.2 Το σχήμα ενός νευρώνα	34
Εικόνα 3.3 Διάφορες ενός νευρωνικού δικτύου με ένα βαθύ νευρωνικό δίκτυο	35
Εικόνα 3.4 Τα είδη μηχανικής μάθησης.....	37
Εικόνα 3.5 Οπτικοποίηση της κλίσης καθόδου(gradient descent)	38
Εικόνα 3.6 Διαφορά μεταξύ μικρού και μεγάλου ρυθμού μάθησης	38
Εικόνα 3.7 Οπτικοποίηση του προβλήματος της υπερεκπαίδευσης και υποεκπαίδευσης ..	40
Εικόνα 3.8 Οπτικοποίηση σε γράφημα ενός πολυεπίπεδα perceptrons(MLP).....	42
Εικόνα 3.9 Οπτικοποίηση σε γράφημα ενός επαναλαμβανομένου νευρωνικού δικτύου	43
Εικόνα 3.10 Οπτικοποίηση ενός lstm επιπέδου.....	44
Εικόνα 3.11 Διαφορά αποκωδικοποιητή Lstm και Bilstm.....	46
Εικόνα 3.12 Εξίσωση και γραφική παράσταση της συνάρτησης sigmoid.....	48
Εικόνα 3.13 Εξίσωση και γραφική παράσταση της συνάρτησης Tanh.....	49
Εικόνα 3.14 Εξίσωση και γραφική παράσταση της συνάρτησης Relu	50
Εικόνα 3.15 Εξίσωση και γραφική παράσταση της Βηματικής συνάρτησης.....	51
Εικόνα 3.16 Εξίσωση και γραφική παράσταση της γραμμικής συνάρτησης ενεργοποίησης	52
Εικόνα 3.17 Οπτικοποίηση της πρόωρης διακοπής	53
Εικόνα 3.18 Οπτικοποίηση της τεχνικής Dropout σε ένα βαθύ νευρωνικό δίκτυο	54
Εικόνα 4.1 Συγκέντρωση δεδομένων για τα κρυπτονομίσματα.....	58
Εικόνα 4.2 Κώδικας python για κατασκευή διαγράμματος του Bitcoin	58
Εικόνα 4.3 Κώδικας python για κατασκευή διαγράμματος του Ethereum.....	58
Εικόνα 4.4 Η χρονοσειρά του Bitcoin από το 2017 μέχρι σήμερα.....	59
Εικόνα 4.5 Η χρονοσειρά του Ethereum από το 2017 μέχρι σήμερα	59
Εικόνα 4.6 Κώδικας python για τον υπολογισμό της ποσοστιαίας αλλαγής ανά χρονικό βήμα των κρυπτονομισμάτων	61
Εικόνα 4.7 Κώδικας Python για την κατασκευή ιστογράμματος των ρυθμών αλλαγής του Bitcoin	61
Εικόνα 4.8 Ο ρυθμός αλλαγής της τιμής του Bitcoin ως ιστόγραμμα σε σχέση με το ακολουθούθαγε κανονική κατανομή	62
Εικόνα 4.9 Κώδικας Python για την κατασκευή ιστογράμματος των ρυθμών αλλαγής του Ethereum	63
Εικόνα 4.10 Ο ρυθμός αλλαγής της τιμής του Ethereum ως ιστόγραμμα σε σχέση με το ακολουθούθαγε κανονική κατανομή	63
Εικόνα 4.11 Κώδικας Python για έλεγχο στασιμότητας του Bitcoin	64
Εικόνα 4.12 Κώδικας Python για έλεγχο στασιμότητας του Ethereum	65

**Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε
δεδομένα μεγάλου όγκου**

Εικόνα 4.13 Κώδικας Python για έλεγχο στασιμότητας του ρυθμού αλλαγής του Bitcoin ...	65
Εικόνα 4.14 Κώδικας Python για έλεγχο στασιμότητας του ρυθμού αλλαγής του Ethereum	66
Εικόνα 4.15 Κώδικας Python για την κατάλληλη δημιουργία των training και Test sets	67
Εικόνα 4.16 Κώδικας Python για την κατάλληλη δημιουργία των training και Test sets	67
Εικόνα 4.17 Σωστή και λάθος διαχώριση δεδομένων χρονοσειράς για εκπαίδευση και δοκιμή	68
Εικόνα 5.1 Σύγκριση Tensor με διαφορετικά Data containers	70
Εικόνα 5.2 Γράφημα για το πως λειτουργεί η ροή δεδομένων στην βιβλιοθήκη Tensorflow.	70
Εικόνα 5.3 Κώδικας Python για υλοποίηση του βαθιού νευρωνικού δικτύου αρχιτεκτονικής MLP	72
Εικόνα 5.4 Κώδικας Python για την δημιουργία γραφήματος αρχιτεκτονικής βαθιού νευρωνικού δικτύου τύπου MLP	72
Εικόνα 5.5 Το βαθύ νευρωνικό δίκτυο που υλοποιήθηκε στα πλαίσια υλοποίησης της διπλωματικής αρχιτεκτονικής MLP	73
Εικόνα 5.6 Κώδικας Python για υλοποίηση του βαθιού νευρωνικού δικτύου αρχιτεκτονικής LSTM	74
Εικόνα 5.7 Κώδικας Python για την δημιουργία γραφήματος αρχιτεκτονικής βαθιού νευρωνικού δικτύου τύπου LSTM	74
Εικόνα 5.8 Το βαθύ νευρωνικό δίκτυο που υλοποιήθηκε στα πλαίσια υλοποίησης της διπλωματικής αρχιτεκτονικής LSTM	75
Εικόνα 5.9 Κώδικας Python για υλοποίηση του βαθιού νευρωνικού δικτύου αρχιτεκτονικής BI-LSTM	76
Εικόνα 5.10 Κώδικας Python για την δημιουργία γραφήματος αρχιτεκτονικής βαθιού νευρωνικού δικτύου τύπου BI-LSTM	76
Εικόνα 5.11 Το βαθύ νευρωνικό δίκτυο που υλοποιήθηκε στα πλαίσια υλοποίησης της διπλωματικής αρχιτεκτονικής Bilstm	77
Εικόνα 5.12 Κώδικας Python για την οπτικοποίηση των προβλέψεων σε σχέση με τις πραγματικές τιμές των κρυπτονομισμάτων	78
Εικόνα 5.13 Σύγκριση των προβλέψεων με την πραγματική τιμή του Ethereum	78
Εικόνα 5.14 Σύγκριση των προβλέψεων με την πραγματική τιμή του Bitcoin	79

**Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε
δεδομένα μεγάλου όγκου**

Συντομογραφίες

NN	Neural Networks
MLP	Multilayer Perceptron
RNN	Recurrent Neural Network
LSTM	Long Short-term Memory
Bi-Lstm	Bidirectional Lstm
ADF	Augmented Dickey-Fuller
ACF	Autocorrelation function
PACF	Partial autocorrelation function
Arima	Autoregressive Integrated moving average
Btc	Bitcoin
Eth	Ethereum
Scikit-Learn	Sklearn

**Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε
δεδομένα μεγάλου όγκου**

Chapter 1 Εισαγωγή

Τα κρυπτονομίσματα είναι εικονικά νομίσματα τα οποία βασίζονται πάνω στην τεχνολογία blockchain. Έχουν αποκτήσει διαδεδομένη υιοθεσία στην αγορά από την εισαγωγή του bitcoin το 2009 το πιο δημοφιλές κρυπτονόμισμα μέχρι τώρα. Αν και οι επενδύσεις έχουν επικερδές αποδόσεις, οι πανταχού παρόντες διακυμάνσεις στα περισσότερα κρυπτονομίσματα καθιστούν τέτοιες επενδύσεις προκλητικές και επικίνδυνες. Για αυτούς τους λόγους, οι επενδυτές απαιτούν μια μέθοδο πρόβλεψης η οποία να συλλάβει αποτελεσματικά τις διακυμάνσεις των τιμών των κρυπτονομισμάτων για να ελαχιστοποιήσουν τον κίνδυνο και να αυξήσουν τα κέρδη τους(Roy et al., 2018) .

Η πρόβλεψη τάσεων σε περιουσιακά στοιχεία όπως οι μετοχές ή τα κρυπτονομίσματα είναι ένα πολύ δύσκολο έργο(Catania et al., n.d.; Tarasova et al., 2020) Ο στόχος αυτής της διπλωματικής εργασίας είναι η κατανόηση για το πως μπορούμε να χρησιμοποιήσουμε βαθιά νευρωνικά δίκτυα για την πρόβλεψη χρονοσειρών, πιο συγκεκριμένα τα δεδομένα τα οποία διαδέχτηκαν για να αναπαραστήσουν τις χρονοσειρές μας είναι τα κρυπτονομίσματα. Τα κρυπτονομίσματα είναι συχνά στο κέντρο προσοχής για αυτό και επιλέχθηκαν καθώς φαινόταν πολύ ενδιαφέρον πως τα βαθιά νευρωνικά δίκτυα θα μπορούσαν να ανταπεξέλθουν στην κατανόηση μοτίβων τόσο ασταθών δεδομένων. Ο συγγραφέας δεν πιστεύει, ούτε κατά διάνοια, ότι κάποιος μπορεί να προβλέψει τις μελλοντικές τιμές των κρυπτονομισμάτων ή παρομοίων δεδομένων κατασκευάζοντας κάποια βαθιά νευρωνικά δίκτυα. Ο συγγραφέας όμως πιστεύει πως τα βαθιά νευρωνικά δίκτυα όπως και η μηχανική μάθηση είναι πολύ χρήσιμα εργαλεία για την κατανόηση των μοτίβων τέτοιου είδους δεδομένων.

Σε αυτή την διπλωματική εργασία θα πραγματοποιηθεί μια επεξήγηση, για το τι είναι οι χρονοσειρές, οι διάφοροι τρόποι οι οποίοι θα μπορούσαν να υλοποιηθούν για την πρόβλεψη τους και στην συνέχεια θα χρησιμοποιηθούν η γλώσσα python και πιο συγκεκριμένα, κυρίως οι βιβλιοθήκες Keras και Sklearn για την δημιουργία βαθιών νευρωνικών δικτύων, τα οποία θα είναι ικανά να προβλέψουν, (όσο το δυνατόν καλύτερα γίνεται), τις μελλοντικές τιμές κρυπτονομισμάτων, χρησιμοποιώντας τις τιμές που είχαν στο παρελθόν ως δεδομένα για να εκπαιδεύσουμε τα βαθιά νευρωνικά δίκτυα. Θα χρησιμοποιηθεί η τεχνική της ολίσθησης παραθύρου, για την μετατροπή της χρονοσειράς σε ένα πρόβλημα εποπτευόμενης μάθησης. Θα κατασκευαστούν και θα εξηγηθούν εκτεταμένα 3 είδη νευρωνικών δικτύων τα mlp, rnn και τέλος το Lstm. Εν τελεί θα γίνει μια σύγκριση των προβλέψεων των παραπάνω νευρωνικών δικτύων.

Chapter 2 Χρονοσειρές

2.1 Βασικά Χαρακτηριστικά μια χρονοσειράς

2.1.1 Τι είναι μια χρονοσειρά

Οι χρονοσειρές ουσιαστικά είναι χρονικά διατεταγμένες παρατηρήσεις X_t καταγεγραμμένες την χρονική στιγμή t . Οι χρονομέτρες μπορούν να διαχωριστούν σε δυο κατηγορίες ανάλογα με την φύση του t , αν οι παρατηρήσεις καταγράφονται σε διακεκριμένες χρονικές στιγμές τότε η χρονοσειρά είναι διακριτή αλλιώς αν καταγράφονται συνεχόμενα σε κάποια πέρασμα του χρόνου τότε είναι συνέχεις. Στην περίπτωση των κρυπτονομισμάτων προφανώς οι χρονοσειρές θα είναι διακριτές.

2.1.2 Από τι απαρτίζεται μια χρονοσειρά

Τα κυριότερα στοιχεία που απαρτίζουν μια χρονοσειρά είναι:

- **Tt: τάση** - Η τάση αναπαριστά την μακροχρόνια αλλαγή του επιπέδου της χρονοσειράς. (reviews & 2019, n.d.) Αυτή η αλλαγή μπορεί να είναι ανόδων ή καθόδων. Αν αυτή η αλλαγή είναι συστηματική σε μια κατεύθυνση τότε η τάση είναι μονοτονική. Ένα κλασσικό παράδειγμα ανοδικής τάσης σε μια χρονοσειρά είναι η μηνιαία εκπομπή του διοξειδίου του άνθρακα στην ατμόσφαιρα και ένα παράδειγμα καθοδικής τάσης είναι οι ετήσιες γεννήσεις στην Ελλάδα τις τελευταίες δεκαετίες .
- **St: εποχικότητα** - Η βραχυπρόθεσμη επανάληψη κύκλων με συγκεκριμένη συχνότητα. Ένα κλασσικό παράδειγμα εποχικότητας είναι η άφιξη των τουριστών στην Ελλάδα. (Hylleberg et al., 1993)
- **Ct: κυκλικότητα** – Η κυκλικότητα όπως και η εποχικότητα έχει επαναληπτική φύση αλλά η συχνότητα στην συγκεκριμένη περίπτωση δεν είναι συγκεκριμένη, συνήθως μεγαλύτερη από 1 χρόνο (Bry et al., 1971)
- **Rt: Το υπολειπόμενο κομμάτι** το οποίο καλύπτει οτιδήποτε δεν βρίσκεται στα τρία παραπάνω
- **Τέλος** μια χρονοσειρά είναι είτε προσθετικού μοντέλου είτε πολλαπλασιαστικού μοντέλου ανάλογα με την διακύμανση της. Αν η διακύμανση παραμένει σταθερή ανεξάρτητα από

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

την τιμή της χρονοσειράς τότε είναι προσθετικού μοντέλου αλλιώς είναι πολλαπλασιαστικού.(Nwogu et al., 2019)

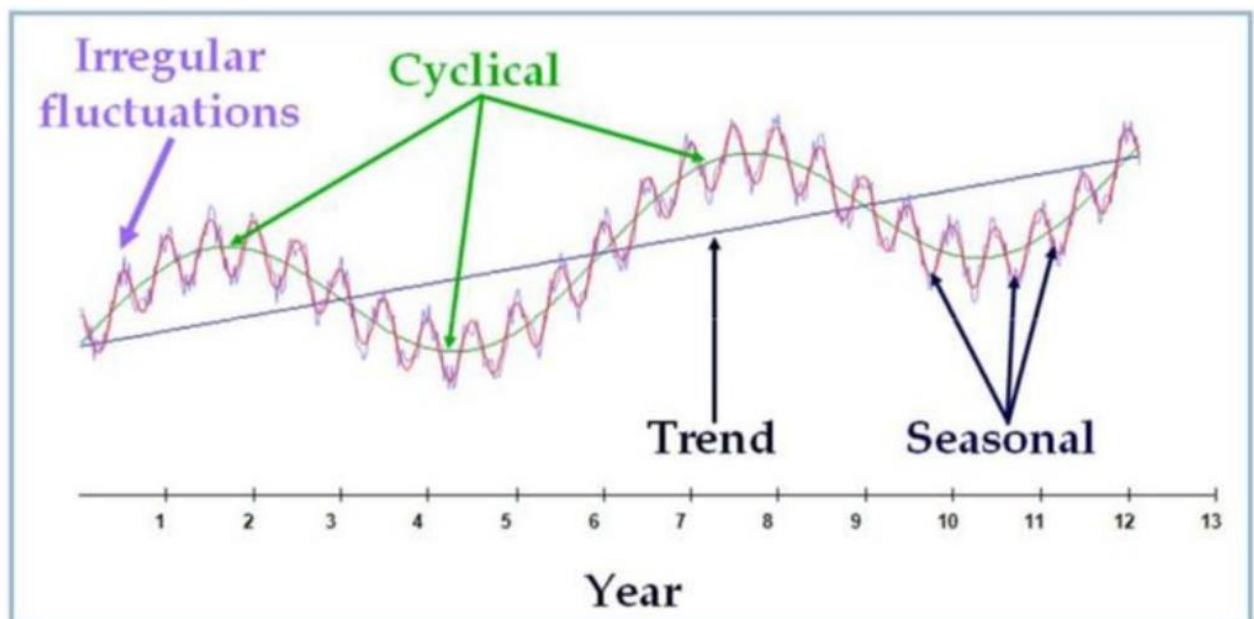
Το προσθετικό μοντέλο χρονοσειράς

$$y_t = T_t + S_t + C_t + R_t \quad (2.1)$$

Το πολλαπλασιαστικό μοντέλο χρονοσειράς

$$y_t = T_t * S_t * C_t * R_t \quad (2.2)$$

Μια οπτικοποίηση των παραπάνω στοιχείων φαίνεται παρακάτω



Εικόνα 2.1 Οι φάσεις μια χρονοσειράς

2.1.3 Στασιμότητα

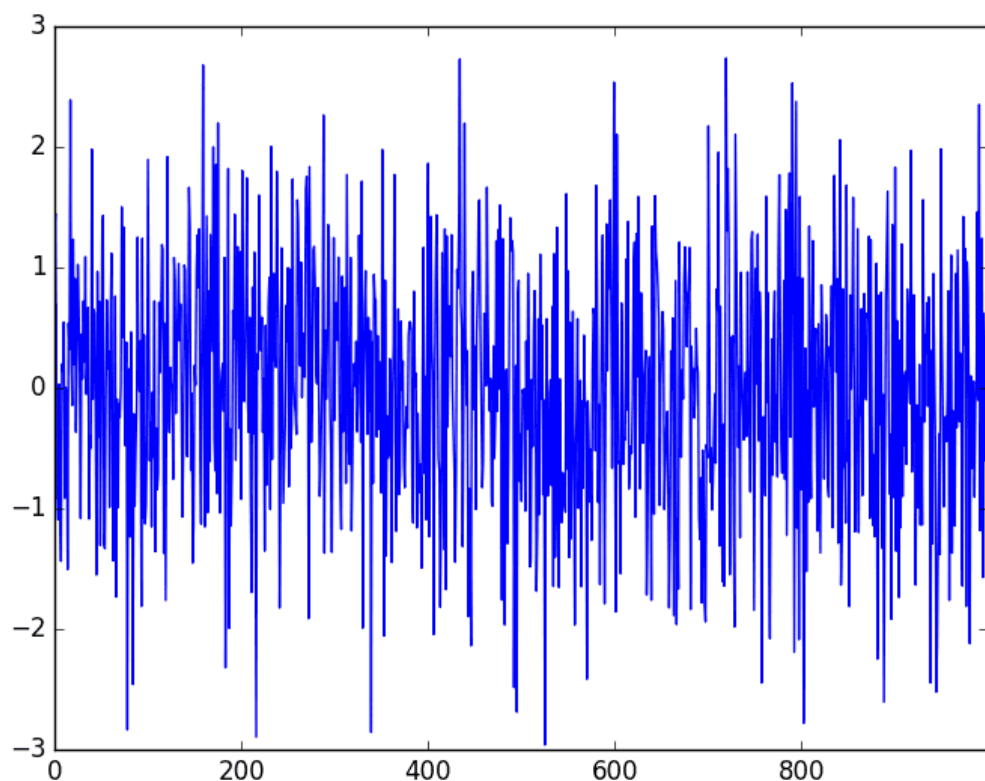
Η στασιμότητα περιγράφει την έννοια πως τα στατιστικά χαρακτηριστικά της χρονοσειράς δεν αλλάζουν με τον χρόνο, εξάλλου είναι πιο εύκολο να προβλέψεις το μέλλον, αν το

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

μέλλον μοιάζει με το παρόν. (Manuca et al., n.d.) Με μαθηματικούς ορους μια χρονοσειρά είναι στάσιμη όταν έχει:

- Σταθερό μέσο ορό
- Σταθερή διακύμανση
- Συνδιακύμανση ανεξάρτητη από τον χρόνο

Ένα παράδειγμα μιας στάσιμης χρονοσειράς είναι ο λευκός θόρυβος, μια οπτικοποίηση φαίνεται παρακάτω.



Εικόνα 2.2 Οπτικοποίηση του λευκού θορύβου ως χρονοσειρά

Ο πιο απλός τρόπος για να καταλάβει κάποιος αν μια χρονοσειρά είναι στάσιμη είναι με το να την απεικονίσει και να αναζητήσει αν έχει κάποια είδους τάση ή εποχικότητα αν δεν υπάρχει κανένα από τα δυο τότε η χρονοσειρά είναι στάσιμη. Όμως το να

Πρόβλεψη της τιμής των κρουπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

προσπαθεί κάποιος να καθορίσει την στασιμότητα με αυτό τον τρόπο είναι ένα αμφίβολο έργο. Για αυτό το λόγο χρησιμοποιείται συνήθως το Augmented Dickey-Fuller test το οποίο είναι τύπου μοναδιαίας ρίζας στατιστικού test.

2.2 Η τεχνική της ολίσθησης παραθύρου

Τα δεδομένα χρονοσειρών μπορούν να παραφραστούν ως δεδομένα εποπτευομένης μάθησης. Αυτό μπορεί να επιτευχθεί με τον μετασχηματισμό των δεδομένων ώστε να παρομοιαστούν ως πρόβλημα εποπτευομένης μάθησης αν μας έχει δοθεί μια ακολουθία αριθμών μιας χρονοσειράς. Αυτό μπορεί να επιτευχθεί χρησιμοποιώντας τα προηγούμενα χρονικά βήματα ως μεταβλητές εισόδου και χρησιμοποιώντας το επόμενο χρονικό βήμα ως μεταβλητή εξόδου. (Sciences & 1995, n.d.) Έστω ότι έχουμε την παρακάτω χρονοσειρά:

Χρόνος	Τιμή
1	109
2	182
3	5
4	93
5	99

Μπορούμε να ανακατασκευάσουμε την χρονοσειρά μας χρησιμοποιώντας την τιμή του προηγούμενου βήματος για να προβλέψουμε το επόμενο, κάνοντας αυτό η παραπάνω χρονοσειρά μετατρέπεται σε:

X	Y
?	109
109	182
182	5

Πρόβλεψη της τιμής των κρουπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

5	93
93	99
99	?

Από την παραπάνω μετατροπή των δεδομένων μπορούμε να συμπεράνουμε τα εξής:

- Η έξοδος του προηγούμενου χρονικού βήματος έχει γίνει η είσοδος του επομένου χρονικού βήματος
- Η σειρά των τιμών έχει παραμείνει και πρέπει πάντοτε να παραμένει η ίδια
- Δεν έχουμε τιμή εισόδου για την πρώτη τιμή της χρονοσειράς μας που σημαίνει δεν μπορούμε να την χρησιμοποιήσουμε για να εκπαιδεύσει, συνεπώς η πρώτη σειρά θα διαγραφεί
- Θα πρέπει εμείς να προβλέψουμε χρησιμοποιώντας όλα τα προηγούμενα βήματα ως “δάσκαλο” την τιμή της τελευταίας εισόδου

Στο παραπάνω παράδειγμα χρησιμοποιούμε μόνο το ακριβώς προηγούμενο βήμα για να προβλέψουμε το επόμενο, στην πραγματικότητα όμως θα μπορούμε να χρησιμοποιούσαμε περισσότερα από ένα βήματα για την πρόβλεψη του επομένου. Το ποσό πίσω θα πάμε, δηλαδή το ποσά βήματα θα χρησιμοποιήσουμε ονομάζεται lag στις χρονοσειρές. Το παραπάνω παράδειγμα με lag=2 θα μετατρεπόταν σε:

X1	X2	Y
?	?	109
?	109	182
109	182	5
5	93	99
93	99	Y1
99	Y1	Y2

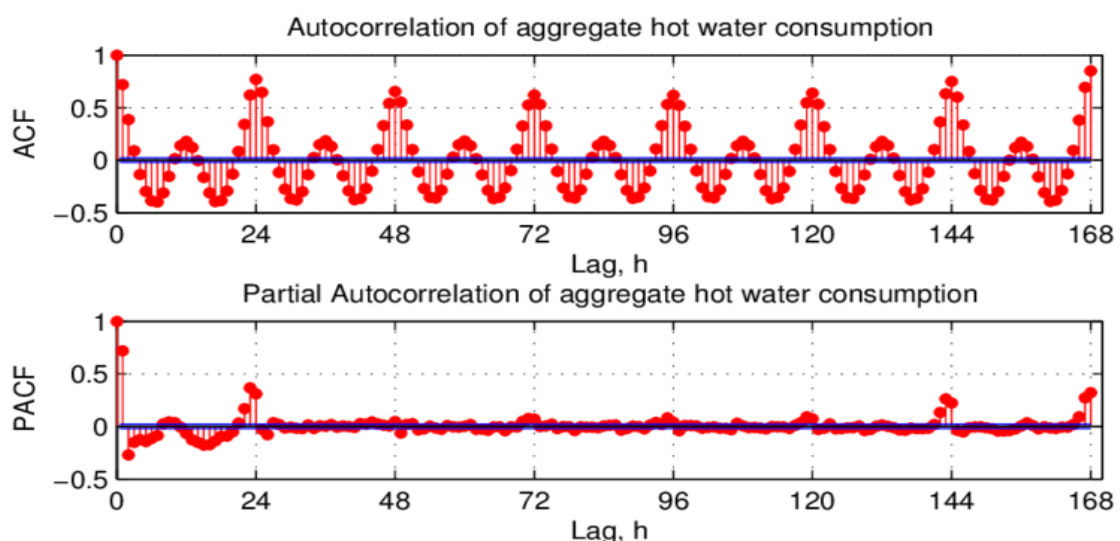
Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

Y1	Y2	Y3
----	----	----

Όπως ακριβώς και στο παράδειγμα με $\text{lag}=1$ οι πρώτες 2 γραμμές μας είναι άχρηστες καθώς δεν γνωρίζουμε τις εισόδους για να προβλεπτή αυτή η έξοδος αρά θα διαγραφτούν, συμπεραίνουμε λοιπόν ότι οι πρώτες $r=\text{lag}$ σειρές των δεδομένων δεν μας είναι χρήσιμες στην εκπαίδευση μας.

2.3 Στατιστική ανάλυση

Η αυτοσυσχέτιση εξηγεί την σχέση μεταξύ παρατηρήσεων της ίδιας μεταβλητής σε διαφορετικά χρονικά βήματα και αναπαριστά τον βαθμό ομοιότητας της χρονοσειράς με τον εαυτό της, κάποια χρονικά βήματα πιο πίσω. Η συνάρτηση αυτοσυσχέτισης (autocorrelation function) συνήθως χρησιμοποιείται για να υπολογιστούν οι αυτοσυσχετίσεις μιας χρονοσειράς με τον εαυτό της σε παρελθοντικά χρονικά βήματα, μας επιστρέφει τιμές τις οποίες τις κάνουμε plot με κάποιο διάστημα εμπιστοσύνης για να περιγράψουμε πως η παρόντα τιμή της χρονοσειράς σχετίζεται με τις τιμές της στο παρελθόν. Σε κάποια πράγματα όμως η μερικώς αυτοσυσχέτιση μας δίνει μια καλύτερα εικόνα της χρονοσειράς μας δείχνει την άμεση αυτοσυσχέτιση της χρονοσειράς με ένα παρελθοντικό βήμα της, δηλαδή την συσχέτιση που δεν έχει εξηγηθεί είδη από τις παρατηρήσεις των μικρότερων lags. (Engel et al., 2012) Για παράδειγμα η μερικώς αυτοσυσχέτιση για $\text{lag}=4$ είναι μόνο η συσχέτιση που δεν έχει εξηγηθεί είδη από τα lags 1,2 και 3 αντίστοιχα. Παρακάτω φαίνεται ένα παράδειγμα όσον αφορά το ποσό μπορούν να διαφέρουν η αυτοσυσχέτιση και η μερικώς αυτοσυσχέτιση (η μπλε γραμμή αναπαριστά τον δείκτη εμπιστοσύνης).



Εικόνα 2.3 ACF και PACF για κατανάλωση καυτού νερού

Πρόβλεψη της τιμής των κρουπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

Η μερικώς αυτοσυσχέτιση στις χρονοσειρές υπολογίζεται από την συνάρτηση μερικώς αυτοσυσχέτισης (partial autocorrelation function) ή αλλιώς PACF.

2.4 Στρατηγικές προβλέψεις χρονοσειρών

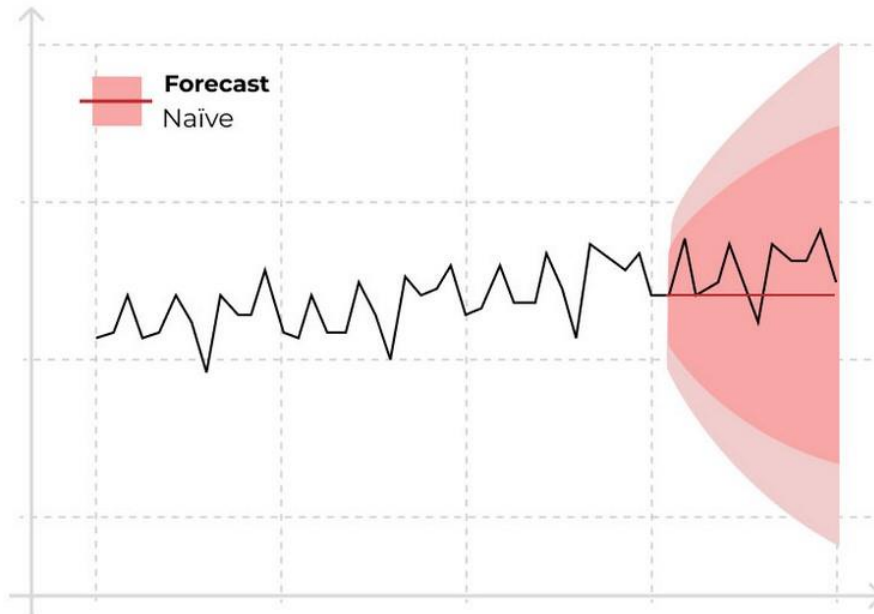
2.4.1 Αφελής πρόβλεψη

Η πιο απλή μέθοδος που μπορεί να χρησιμοποιηθεί για την πρόβλεψη μιας χρονοσειράς είναι αυτή της αφελής πρόβλεψης. (and & 2017, n.d.) Το αφελές μοντέλο χρησιμοποιείται συνήθως ως η γραμμή αναφοράς και δεν χρειάζεται καμία απολύτως εκπαίδευση, αυτό συμβαίνει διότι στην πραγματικότητα το μόνο πράγμα το οποίο κάνει είναι να θεωρεί πως η τιμή στο επόμενο χρονικό βήμα θα είναι ακριβώς η ίδια με την τιμή του προηγούμενου χρονικού βήματος. Η φόρμουλα είναι η εξής:

$$y_{t+1} = y_t \quad (2,3)$$

Μπορεί να ακούγεται απλό αλλά σε ένα ανοιχτό σύστημα όπως για παράδειγμα το χρηματιστήριο είναι ιδιαίτερα δύσκολο να πέτυχεις καλύτερη πρόβλεψη από αυτή του αφελούς μοντέλου. Ένα παράδειγμα πρόβλεψης του αφελούς μοντέλου φαίνεται παρακάτω:

NAÏVE MODEL



Εικόνα 2.4 Παράδειγμα πρόβλεψης του αφελούς μοντέλου

2.4.2 Η οικογένεια των μοντέλων Arima

Τα μοντέλα της τάξης Arima αν και απλά παρέχουν μια πολύ αξιόπιστη μέθοδο για πρόβλεψη χρονοσειρών. (Ariyo et al., 2014; Newbold, 1983) Arima είναι ένα ακρώνυμο το οποίο σημαίνει Autoregressive integrated Moving average, το ακρώνυμο αυτό περιγράφει κάθε κύριο χαρακτηριστικό του ιδίου του μοντέλου. Συνοπτικά, είναι:

- AR: Autoregression είναι ένα μοντέλο το οποίο χρησιμοποιεί την εξαρτώμενη σχέση μεταξύ μιας παρατήρησης με ένα πλήθος παρατηρήσεων σε προηγούμενα χρονικά βήματα. Ουσιαστικά το Ar μοντέλο προβλέπει τιμές ως το άθροισμα των προηγούμενων παρατηρήσεων πολλαπλασιασμένες με ένα σταθερό αριθμό και περιγράφεται από την παρακάτω εξίσωση για lag=2:

$$X_t = c + \varphi_1 * X_{t-1} + \varphi_2 * X_{t-2} + E_t \quad (2,4)$$

C: σταθερός αριθμός

φ_1, φ_2 : αριθμητικοί συντελεστές για lag=1 και lag=2 αντίστοιχα

E_t : Η διαφορά μεταξύ της πρόβλεψης και της πραγματικής τιμής

Πρόβλεψη της τιμής των κρουπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

- I: Integrated η αλλιώς ολοκλήρωση οπου είναι η χρήση παραγωγίσεων για να κάνουμε την χρονοσειρά μας στάσιμη
- MA: Moving average είναι εννοιολογικά μια γραμμική παλινδρόμηση την τωρινής τιμής της χρονοσειράς έναντι του τωρινού και των προηγούμενων σφαλμάτων. Έστω lag=2 τότε το μοντέλο περιγράφεται από την παρακάτω εξίσωση:

- $$R_t = C + \theta_1 * E_{t-1} + \theta_2 * E_{t-2} + E_t \quad (2,5)$$

C: σταθερός αριθμός

θ_1, θ_2 : αριθμητικοί συντελεστές για τις τιμές του πρώτου και δευτέρου lag αντίστοιχα

E_t : Υπόλοιπο για το τωρινό βήμα

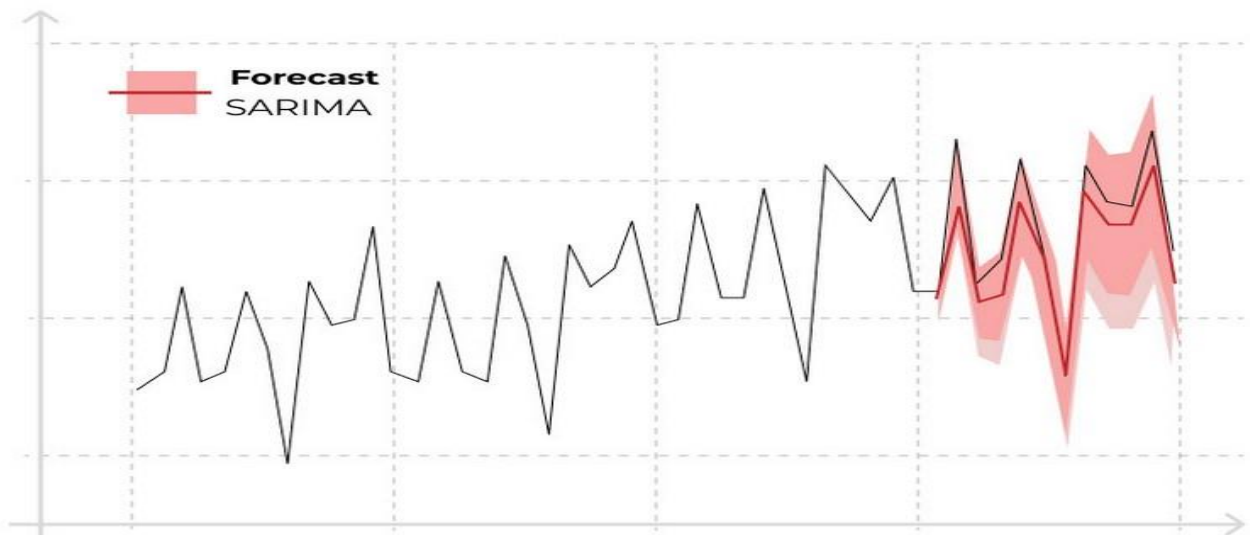
E_t : Υπόλοιπα για τα εκάστοτε χρονικά βήματα του παρελθόντος

Κάθε ένα από τα παραπάνω ορίζεται ρητά ως παράμετρο του ARIMA μοντέλου.

Ένα ελάττωμα του μοντέλου όμως είναι πως δεν μπορεί να μεταχειρίζεται εποχιακά δεδομένα το οποίο λύνεται βεβαίως με μια παραλλαγή του το SARIMA.

Ένα παράδειγμα πρόβλεψης με SARIMA φαίνεται παρακάτω:

SEASONAL AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (SARIMA)



Εικόνα 2.5 Παράδειγμα πρόβλεψης ενός Sarima(Seasonal auto-regressive intergrated moving average model)

2.4.3 Εκθετική εξομάλυνση

Η εκθετική εξομάλυνση είναι μια οικογένεια προβλεπτικών μεθόδων, τα οποία σταθμισμένους μέσους ορούς παλιών παρατηρήσεων για να κάνουν προβλέψεις. Η κυρίως ιδέα είναι να δοθεί μεγαλύτερη σημασία στις πιο πρόσφατες παρατηρήσεις, συνεπώς όσο παλιώνουν οι παρατηρήσεις η σημασία τους γίνεται εκθετικά χαμηλότερη. Υπάρχουν τρεις κυρίως μέθοδοι εκθετική εξομάλυνσης για πρόβλεψη χρονοσειρών.

Η μονή εκθετική εξομάλυνση η οποία λαμβάνει μόνο μια παράμετρο E , η οποία επίσης ονομάζεται συντελεστής εξομάλυνσης και συνήθως έχει τιμή από 0 έως 1. Όσο μεγαλύτερη η τιμή του E τόσο περισσότερο το μοντέλο εστιάζει μόνο στις πρόσφατες παρατηρήσεις αγνοώντας τις παλαιότερες. Βέβαια αυτό το μοντέλο δεν λαμβάνει υπόψιν την τάση και την εποχικότητα της χρονοσειράς για αυτό και υπάρχουν και τα υπόλοιπα μοντέλα εκθετικής εξομάλυνσης.

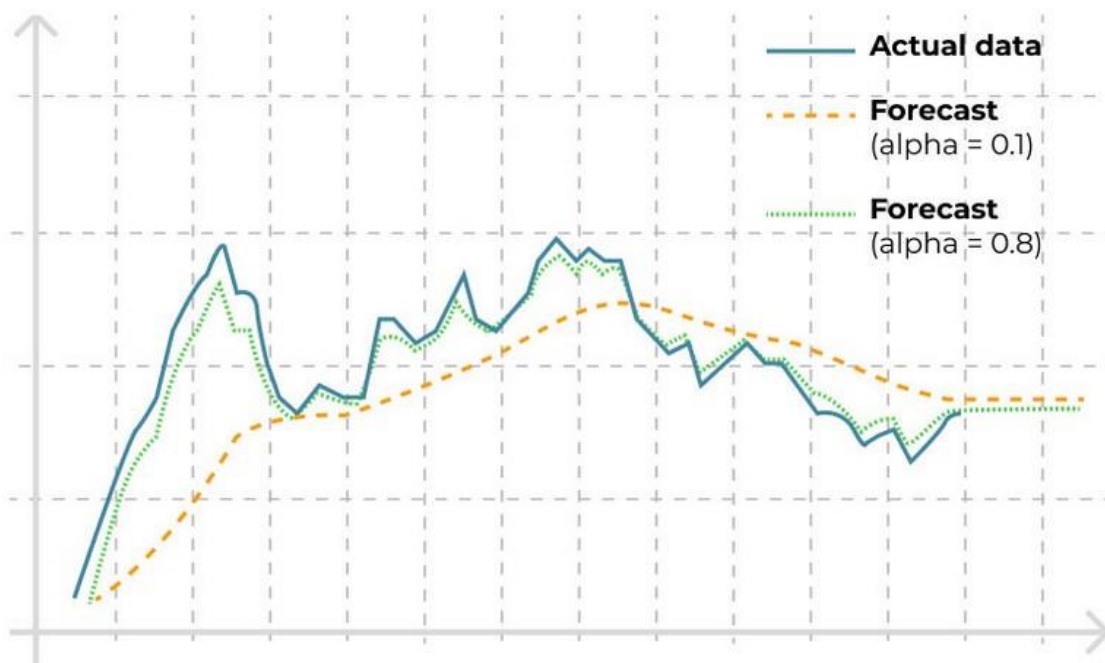
Η διπλή εκθετική εξομάλυνση έρχεται για λύση το πρόβλημα της τάσης, σε αυτό το μοντέλο εκτός από την παράμετρο E χρησιμοποιεί και την παράμετρο T η οποία ελέγχει την επιρροή της τάσης. Αναλόγως με το αν η τάση είναι γραμμική η εκθετική η μέθοδο προσαρμόζεται σε προσθετική και πολλαπλασιαστική αντίστοιχα. Το μόνο που μένει τώρα είναι να συμπεριλάβουμε και την εποχικότητα και αυτό ακριβώς κάνει η τριπλή εκθετική εξομάλυνση.

Η τριπλή εκθετική εξομάλυνση η αλλιώς μέθοδο Holts Winters ονομασμένη από τους δημιουργούς της Charles Holt και Peter Winters λαμβάνει υπόψιν και την εποχικότητα. Σε

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

αυτή χρησιμοποιείται άλλος ένας παράγοντας εξομάλυνσης ο S ο οποίος ελέγχει την επιρροή της εποχικότητας και ακριβώς όπως και ο T μπορεί να προσαρμοστεί για γραμμική και εκθετική εποχικότητα. Η τριπλή εκθετική εξομάλυνση είναι η πιο προχωρημένη εκδοχή και μπορεί να προσαρμοστεί για να αναπτυχθούν μοντέλα μονής ή και διπλής εκθετικής εξομάλυνσης. (de Livera et al., 2011)

EXPONENTIAL SMOOTHING MODEL



Εικόνα 2.6 Παράδειγμα πρόβλεψης εκθετικής εξομάλυνσης

2.4.4 Γραμμική παλινδρόμηση

Η γραμμική παλινδρόμηση είναι μια απλή στατιστική τεχνική η οποία μπορεί να χρησιμοποιηθεί ως προβλεπτικό μοντέλο, ουσιαστικά αυτό που κάνει είναι να δημιουργεί μια σχέση μεταξύ του στόχου μεταβλητής και διάφορων ανεξάρτητων μεταβλητών. Η εξίσωση της γραμμικής παλινδρόμησης μπορεί να αναπαράσταθεί ως:

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

$$Y = \theta^T x \quad (2,6)$$

- θ αντιπροσωπεύει το διάνυσμα των παραμέτρων του μοντέλου
- x αντιπροσωπεύει το διάνυσμα των χαρακτηριστικών του μοντέλου

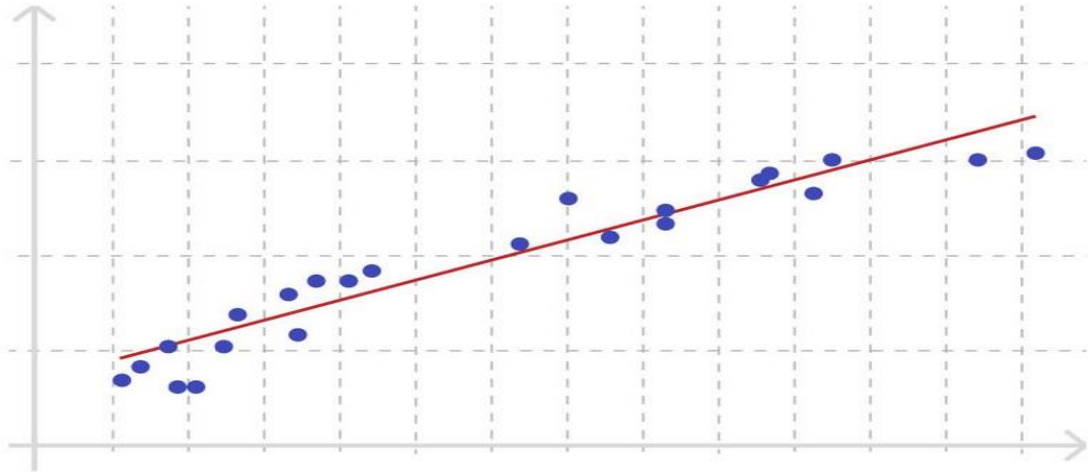
Για να βρεθούν οι παράμετροι θ θα χρησιμοποιηθεί ο αλγόριθμος gradient descent

Η γραμμική κάνει κάποιες υποθέσεις όμως που τις περισσότερες φορές δεν ισχύουν για τα δεδομένα και στα σιγουρά δεν θα ισχύουν στην περίπτωση των κρυπτονομισμάτων ,αυτές είναι:

- Γραμμικότητα: Η προβλεπόμενη τιμή της εξαρτώμενης μεταβλητής y είναι μια γραμμική εξίσωση των ανεξάρτητων μεταβλητών
- Ανεξαρτησία: Τα υπόλοιπα του εκπαιδευμένου μοντέλου είναι ανεξάρτητα το ένα από το άλλο
- Ομοσκεδαστικότητα: Η διακυμάνσεις των υπολοίπων είναι σταθερές για εξαρτημένες μεταβλητές
- Κανονικότητα: Τα σφάλματα έχουν κανονική κατανομή

Ένα παράδειγμα γραμμικής παλινδρόμησης φαίνεται παρακάτω:

LINEAR REGRESSION



Εικόνα 2.7 Παράδειγμα πρόβλεψης γραμμικής παλινδρόμησης

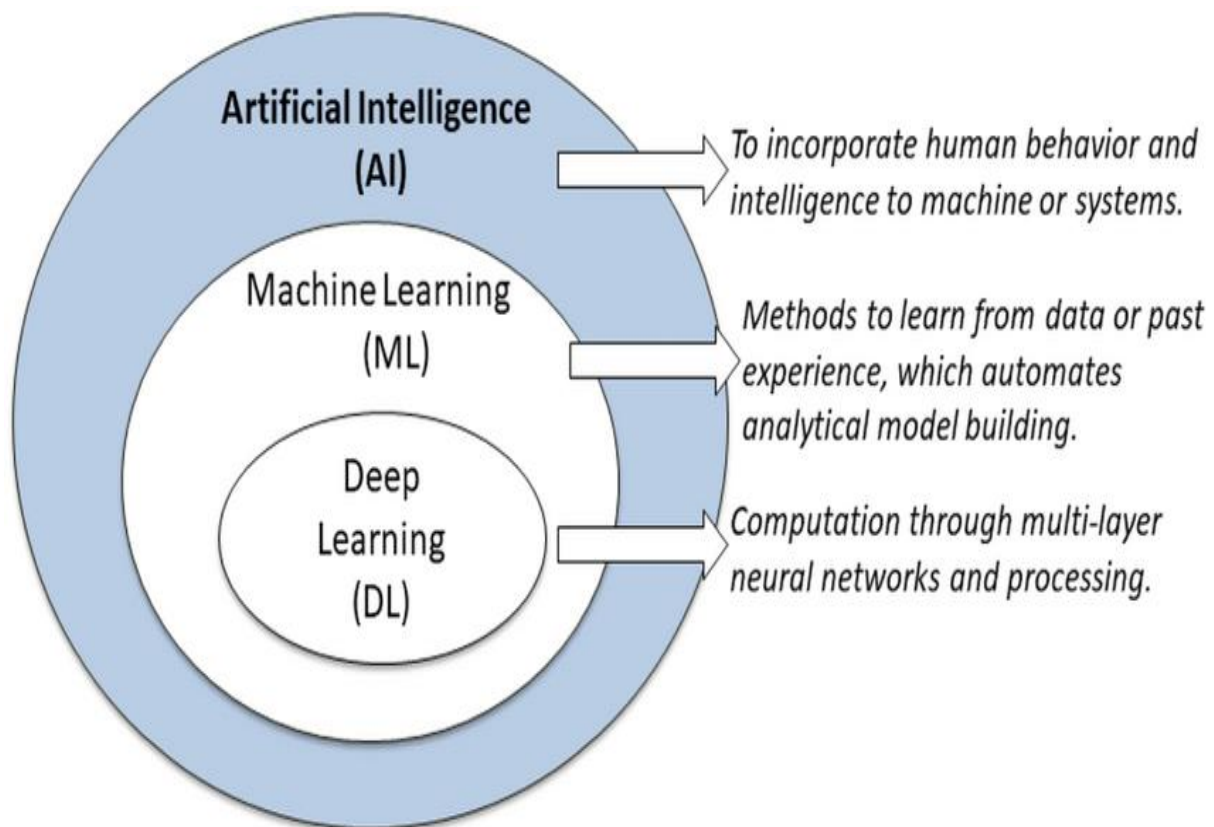
2.4.5 Νευρωνικά δίκτυα

Τα νευρωνικά δίκτυα αποτελούν εδώ και χρονιά μια τεχνολογία αιχμής και μπορούν να χρησιμοποιηθούν για να αντιμετωπίσουν διάφορα προβλήματα, ένα από αυτά είναι και η πρόβλεψη χρονομέτρων. (Journal & 2018, n.d.) Πιο συγκεκριμένα τα βαθιά νευρωνικά δίκτυα τα οποία προτιμώνται καθώς τα πολύ μεγάλα δίκτυα είναι αργά στην εκπαίδευση, χρησιμοποιούν την αυτοπαλινδρόμηση για να κάνουν προβλέψεις. Μπορούν να σχεδιαστούν με διάφορους τρόπους όπως MLP, CNN, RNN, LSTM οι οποίοι θα μελετηθούν εκτεταμένα στο επόμενο κεφάλαιο αυτής της διπλωματικής, μάλιστα όπως και όλα τα υπόλοιπα στάδια για την δημιουργία και εκπαίδευση ενός βαθιού νευρωνικού δικτύου για την επίλυση του προβλήματος μας, δηλαδή την πρόβλεψη των κρυπτονομισμάτων. Τα κυρία πλεονεκτήματα των βαθιών νευρωνικών δικτύων σε σχέση με τους υπόλοιπους μεθόδους προβλέψεις είναι πως δεν χρειάζεται να γίνει κάποια υπόθεση για την εφαρμογή τους και έχουν την ικανότητα να εντοπίσουν σημαντικές γενικευτικές δυνάμεις. Τέλος τα lstm είναι ιδανικά για διαδοχικά δεδομένα όπως οι χρονοσειρές που τα καθιστά την καλύτερη ελπίδα μας για την ιδανική πρόβλεψη.

Chapter 3 Βαθιά νευρωνικά δίκτυα

3.1 Εισαγωγή

Από τα πρώτα χρονιά που δημιουργήθηκαν οι υπολογιστές γίνεται μια συνεχής προσπάθεια να βρεθεί κάποιος τρόπος, ώστε να γίνουν ικανά να παρέχουν κατάλληλες εξόδους για συγκεκριμένες εισόδους. Την τελευταία δεκαετία λόγω της εξέλιξης της τεχνολογίας, επιτεύχθηκε μια σημαντική εξέλιξη στον τομέα των νευρωνικών δικτύων διότι παρέχουν ακριβώς αυτή την “μαγική” ιδιότητα δηλαδή να παρέχουν μια λύση σε κάτι στο οποίο δεν μπορούσε να βρεθεί μια. Οι πρώτες ιδέες ξεκίνησαν από τους βιολογικούς οργανισμούς και ιδιαίτερα από το νευρικό σύστημα του ανθρώπου, αλλά η μελέτη και η χρήση τους έχει προχωρήσει πολύ περισσότερο σήμερα λειτουργούν με παρόμοιο τρόπο με τον εγκέφαλο του ανθρώπου. Το νευρωνικό δίκτυο μαθαίνει, εκπαιδεύεται, απομνημονεύει, ξεχνάει και θυμάται με την πάροδο του χρόνου. Η βαθιά μάθηση ανήκει στην ευρύτερη οικογένεια την μηχανικής μάθησης, η οποία με την σειρά της θεωρείται τμήμα της τεχνητής νοημοσύνης.



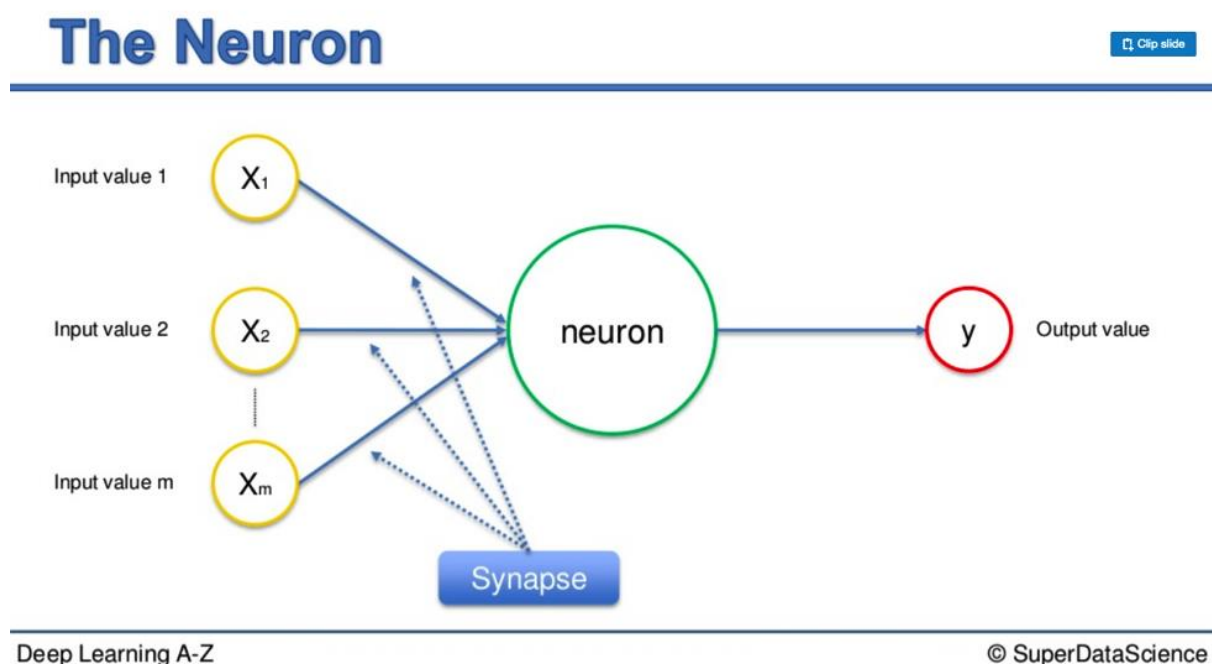
Εικόνα 3.1 Η οικογένεια της τεχνητής νοημοσύνης

3.2 Τεχνητό Νευρωνικό Δίκτυο

Είναι ένα πρόγραμμα που προσμοιάζει την οργάνωση και λειτουργία βιολογικών νευρώνων, συνήθως αποτελείται από διαφορά επίπεδα όπου σε αυτά γίνεται η επεξεργασία. Η μονάδα επεξεργασία είναι είτε ένας νευρώνας είτε μια ομάδα νευρώνων και η δουλειά της είναι να μαζέψει πληροφορία από άλλες μονάδες ή από τα δεδομένα εκπαίδευσης να εκτελέσει έναν υπολογισμό και να διαδώσει το αποτέλεσμα σε άλλες μονάδες. Η επίδραση μια μονάδες σε μια άλλη εξαρτάται από το βάρος διασύνδεσης τους.

3.2.1 Τι είναι ένας νευρώνας

Έστω ένας νευρώνας με m εισόδους, αυτό συνεπάγεται πως θα έχει και m βάρη η αλλιώς συνάψεις ένα για κάθε είσοδο και φυσικά ένα bias. Θα γίνει μια τυχαία αρχικοποίηση στα βάρη και στο bias του νευρώνα για να μπορέσει η γίνει η εκπαίδευση. Οι εισοδοι θα είναι είτε τα δεδομένα εκπαίδευσης είτε έξοδοι από άλλους νευρώνες. Τα βάρη και το bias αλλάζουν κατά την διάρκεια της εκπαίδευσης στόχος της οποίας είναι να βρει τις καταλληλότερες τιμές που μπορούν να λάβουν.



Εικόνα 3.2 Το σχήμα ενός νευρώνα

Παρακάτω φαίνεται η εξίσωση που περιγράφει τον νευρώνα πιο πάνω:

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

$$y = \sum_{i=1}^m (w_i + x_i) + b \quad (3,1)$$

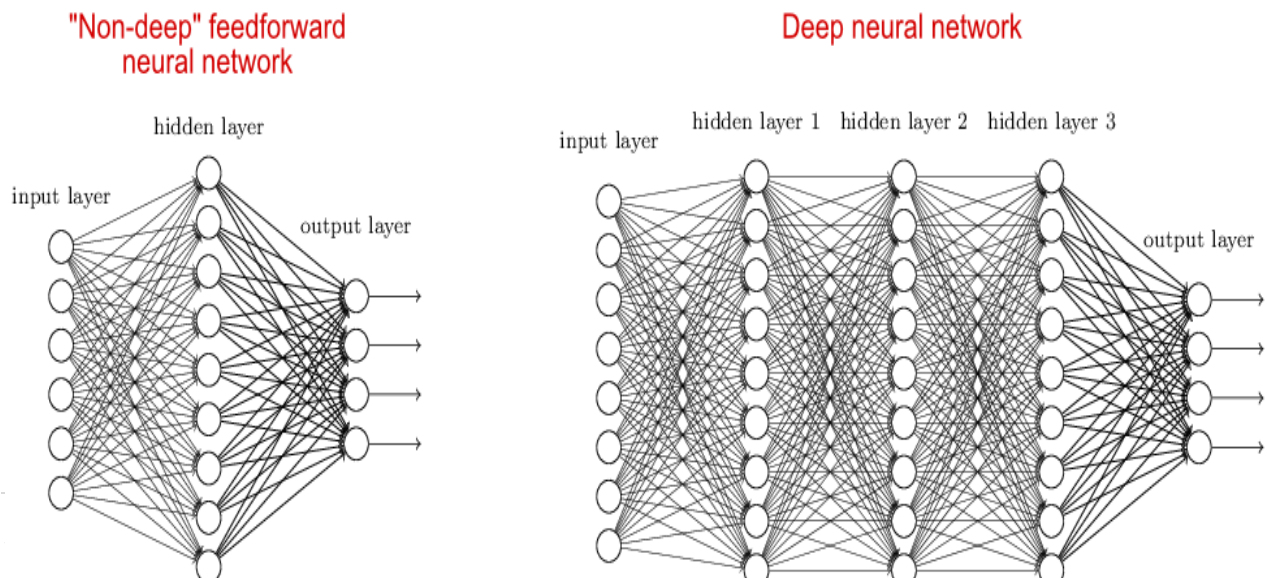
3.2.2 Τι είναι ένα βαθύ νευρωνικό δίκτυο και πότες οι διαφορές του με ένα απλό

Τα νευρωνικά δίκτυα έχουν επίπεδα τα οποία αποτελούνται από πολλούς νευρώνες, ουσιαστικά το κάθε επίπεδο είναι μια ομάδα νευρώνων. Κάθε νευρώνας στο ίδιο επίπεδο παίρνει την ίδια είσοδο η οποία είναι είτε τα δεδομένα εκπαίδευσης αν είναι το πρώτο κρυφό επίπεδο είτε η έξοδοι του προηγούμενου επιπέδου και αναλόγως των βαρών και bias του παρέχει μια έξοδο, η έξοδος του επιπέδου είναι σύνολο όλων των εξόδων των νευρώνων του επιπέδου.

Ένα νευρωνικό δίκτυο αποτελείται από

- Επίπεδο εισόδου(input layer)
- Κρυφό επίπεδο(hidden layer)
- Επίπεδο εξόδου(output layer)

Το βαθύ νευρωνικό δίκτυο είναι απλώς ένα προωθητικό δίκτυο με πολλά κρυμμένα επίπεδα. Τα νευρωνικά δίκτυα μπορεί να είναι είτε προωθητικά είτε επαναλαμβανόμενα και η διαφορά τους είναι πως τα προωθητικά δεν έχουν βρόχους στο γράφημα τους, που σημαίνει πως μπορούν να οργανωθούν σε επίπεδα. Ένα νευρωνικό δίκτυο με μόνον ένα κρυφό επίπεδο ονομάζεται ρηχό, αντιθέτως ένα νευρωνικό δίκτυο με περισσότερο από ένα κρυφό επίπεδο ονομάζεται βαθύ.



Εικόνα 3.3 Διάφορες ενός νευρωνικού δικτύου με ένα βαθύ νευρωνικό δίκτυο

3.3 Εκπαίδευση τεχνητού νευρωνικού δικτύου

3.3.1 Τύποι μηχανικής μάθησης

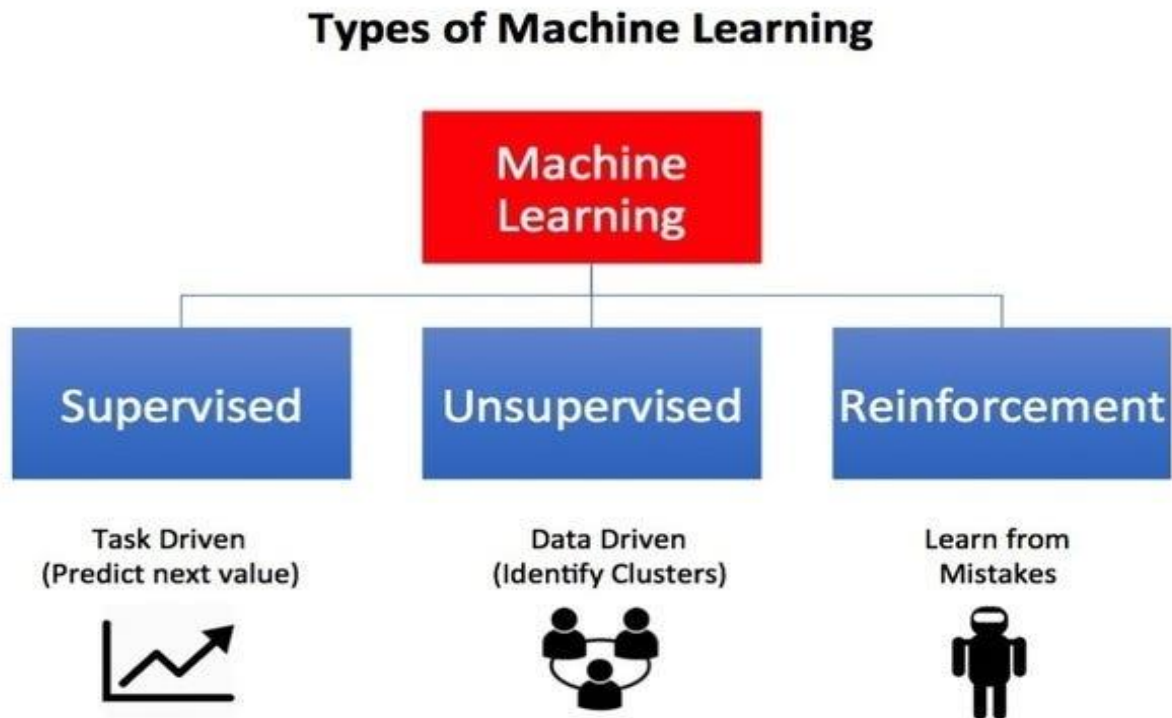
Για την δημιουργία ενός συστήματος το οποίο μπορεί να μάθει και να εκτιμήσει την ευκρίνεια ενός μοντέλου, τα δεδομένα εισόδου συνήθως χωρίζονται σε δεδομένα εκπαίδευσης και δεδομένα αξιολόγησης. Τα δεδομένα εκπαίδευσης χρησιμοποιούνται για την εύρεση των καταλληλότερων παραμέτρων του επιλεγμένου μοντέλου, παράμετροι όπως βαρύ και bias που αναφέρθηκαν. Τα δεδομένα αξιολόγησης υπάρχουν για να δούμε πως τα πηγαίνει το μοντέλο σε δεδομένα τα οποία δεν χρησιμοποιήθηκαν κατά την εκπαίδευση. Ο τρόπος διαχείρισης των δεδομένων για την εκπαίδευση του νευρωνικού μπορεί να χωριστεί σε τρεις κατηγορίες

- Επιβλεπόμενη μάθηση(Supervised learning)
- Μη επιβλεπόμενη μάθηση(Unsupervised learning)
- Ενισχυτική μάθηση(Reinforcement learning)

Στην επιβλεπόμενη μάθηση τα δεδομένα εκπαίδευσης είναι με ετικέτα, δηλαδή τα δεδομένα έχουν μια είσοδο και μια επιθυμητή έξοδο. Κατά την διάρκεια της εκπαίδευσης το δίκτυο μαθαίνει πατέντες ώστε να γίνει ικανό να πετυχαίνει τους επιθυμητούς στόχους. Σε αυτή την διπλωματική χρησιμοποιηθεί αυτή η μέθοδος εκπαίδευσης για τα νευρωνικά μας.

Στην μη επιβλεπόμενη μάθηση όλα τα δεδομένα μας είναι εισόδου και δεν αντιστοιχίζονται με κάποια έξοδο, ο στόχος του μοντέλου αυτή την φορά είναι να βρει τις σχέσεις μεταξύ των στοιχείων εισόδου και να τα ομαδοποιήσει αναλόγως. Συνήθως χρησιμοποιείται όταν επιθυμούμε να εντοπίσουμε πατέντες οι οποίες δεν είναι προφανείς με πρώτη ματιά ή και όταν δεν έχουμε τρόπο να δημιουργήσουμε ετικέτα εξόδου για τα δεδομένα εκπαίδευσης.

Στην ενισχυτική συμπεριφορά υπάρχει ένα προεπιλεγμένο σύστημα επιβράβευσης, σύμφωνα με αυτό το σύστημα το μοντέλο ψάχνει και βρίσκει όλες τις δυνατές καταστάσεις και επιλεγεί εκείνη η οποία μεγιστοποιεί την απόδοση.

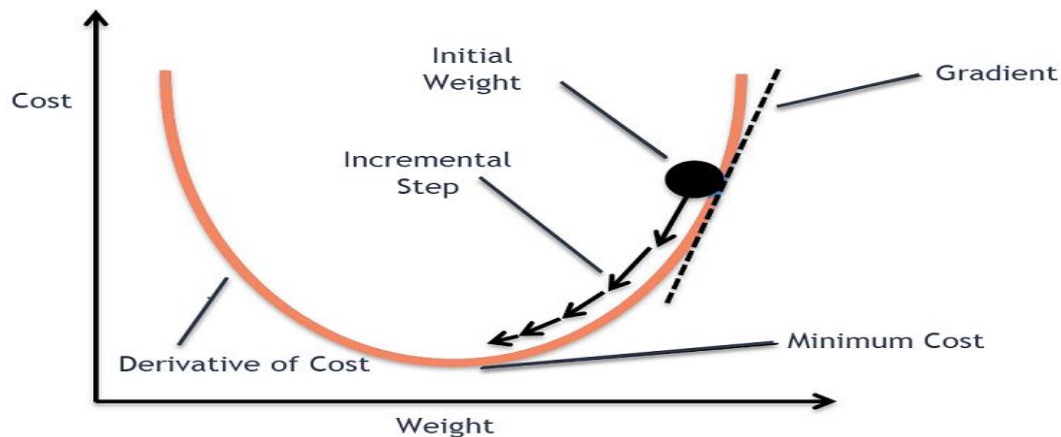


Εικόνα 3.4 Τα είδη μηχανικής μάθησης

3.3.2 Αλγόριθμοι βελτιστοποίησης

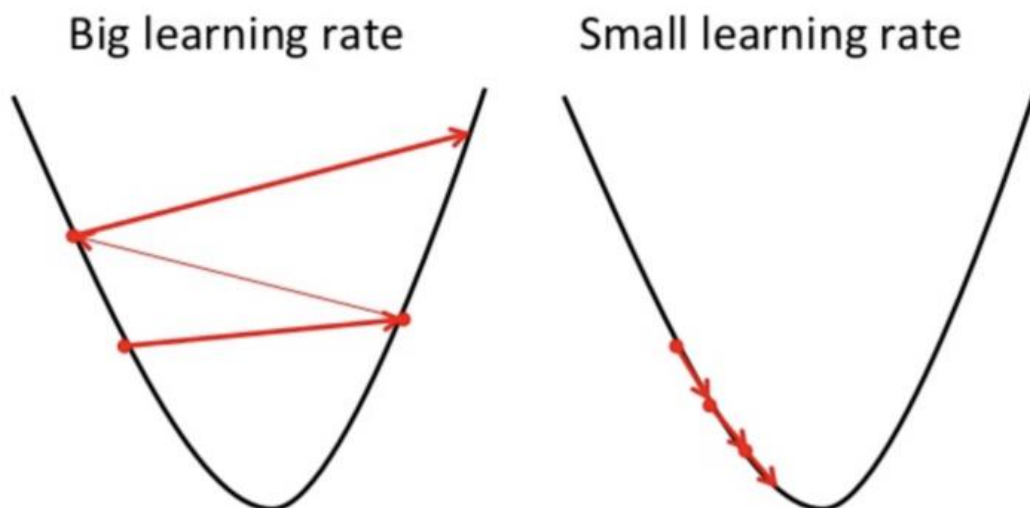
Οι αλγόριθμοι βελτιστοποίησης είναι είδη αλγορίθμων οι οποίοι ειδικεύονται στην εύρεση της βέλτιστης λύσης σε ένα πρόβλημα. Όσον αφορά το δικό μας πρόβλημα δηλαδή την εύρεση των κατάλληλων βαρών και του bias για το νευρωνικό μας δίκτυο οι δυο πιο συνηθισμένοι αλγόριθμοι είναι η μέθοδος κλίσης καθόδου και ο αλγόριθμος πίσω διαδρομής.

- **Κλίσης Καθόδου(gradient descent)**



Εικόνα 3.5 Οπτικοποίηση της κλίσης καθόδου(gradient descent)

Μπορεί να παρομοιαστεί με την κατάβαση ενός λόφου όπου σκοπός είναι το μικρότερο μονοπάτι. Έχοντας για παράδειγμα τα βαρύ τα οποία τα έχουμε αρχικοποίηση με τυχαίες τιμές και θέλοντας να τα ελαχιστοποιήσουμε ο αλγόριθμος gradient descent αρχίζει βήμα βήμα να κατεβαίνει το λόφο δηλαδή να ελαχιστοποιεί τις τιμές τους. Το ποσό μεγάλα η μικρά θα είναι τα βήματα αυτά εξαρτάται από το learning rate ή ρυθμό μάθησης. Πρέπει η επιλογή του ρυθμού μάθησης να γίνει με μεγάλη προσοχή διότι αν είναι πολύ μεγάλος τότε υπάρχει κίνδυνος να μην βρεθεί το ελάχιστο ενώ αν είναι πολύ μικρός μπορεί να χρειαστεί ένα πολύ μεγάλο πλήθος βημάτων για να βρεθεί το ελάχιστο.(Chowdhury et al., n.d.)



Εικόνα 3.6 Διαφορά μεταξύ μικρού και μεγάλου ρυθμού μάθησης

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

Μπορεί να περιγραφεί από την παρακάτω εξίσωση:

$$b = a - \gamma \nabla f(a) \quad (3,2)$$

b: Η επόμενη θέση των βαρών

a: Η τωρινή θέση των βαρών

Το μείον συμβολίζει το κομμάτι της ελαχιστοποίησης του αλγορίθμου

γ : Ο ρυθμός μάθησης που έχει οριστεί

$\nabla f(a)$: Η κατεύθυνση την πιο απότομης κατάβασης

- **Αλγόριθμος πίσω διάδοσης(backpropagation)**

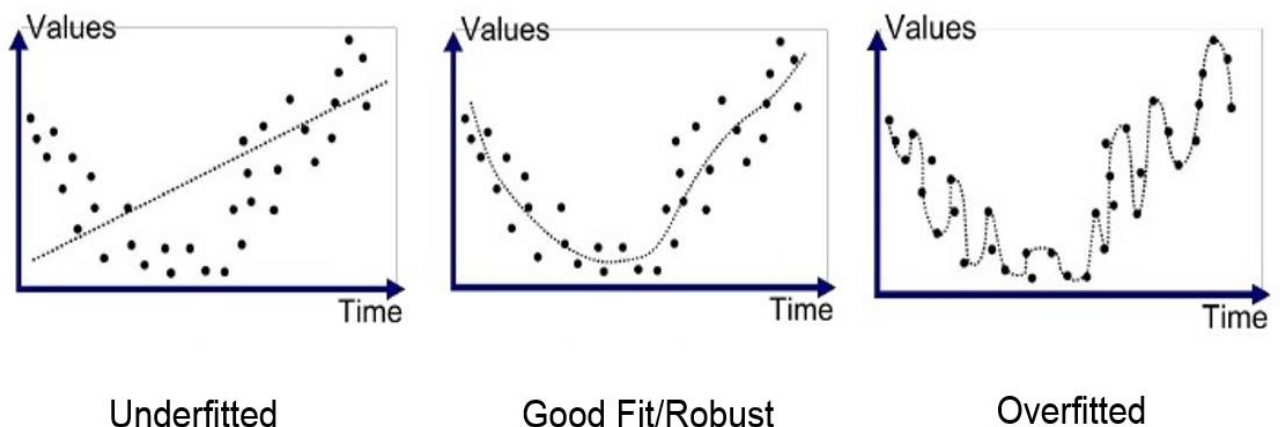
Μπορεί να χωριστεί σε δυο στάδια την πασά προς τα εμπρός(forward pass) και την πασά προς τα πίσω(backward pass).

- Forward pass: Σε αυτό το στάδιο τα δεδομένα εισόδου πέρανε από το επίπεδο εισόδου μέχρι και το επίπεδο εξόδου ώστε να γίνει μια πρόβλεψη. Καθώς περνάει η πληροφορία από τα διαφορά επίπεδα τα βαρύ και το bias αλλάζουν, αφότου γίνει η πρόβλεψη συγκρίνεται με την τιμή του στόχου και η διαφορά τους χρησιμοποιείται για να υπολογιστή η συνάρτηση λάθους(loss function).
- Backward pass: Σε αυτό το στάδιο η ροή της πληροφορίας αντιστρέφεται, αρχίζει η διάδοση του λάθους από το επίπεδο εξόδου μέχρι να φτάσει στο επίπεδο εισόδου, ακριβώς όπως και στο forward pass καθώς περνάει η πληροφορία από τα επίπεδα τα βαρύ και το bias αλλάζουν.

Τα δυο παραπάνω στάδια επαναλαμβάνονται μέχρι να ικανοποιηθεί μια συνθήκη εξόδου του loop συνήθως το πλήθος των επαναλήψεων η το μέγεθος του λάθους είναι αυτή η συνθήκη.

3.3.3 Υπερεκπαίδευση και υποεκπαίδευση

Η υπερεκπαίδευση ή αλλιώς overfitting συμβαίνει όταν απομνημονεύει τις πατέντες των δεδομένων χωρίς να τις κατανοεί, στην συγκεκριμένη περίπτωση το μοντέλο μας φαίνεται να είναι δουλεύει πολύ καλά διότι πετυχαίνει πολύ καλή ακρίβεια στα δεδομένα εκπαίδευσης αλλά δεν μπορεί να ανταποκριθεί σε όλα τα υπόλοιπα δεδομένα δηλαδή σε δεδομένα που δεν γνωρίζουμε είδη την λύση που είναι και ο κύριος λόγος που εκπαιδεύουμε ένα νευρωνικό δίκτυο, για να μας βρει μια λύση σε τέτοιου είδους δεδομένα καθιστώντας συνεπώς το δίκτυο μας άχρηστο. Μια λύση στο πρόβλημα είναι η εκπαίδευση του μοντέλου με περισσότερα δεδομένα αλλά αυτά δεν είναι πάντα διαθέσιμα, μια άλλη είναι η απλοποίηση της αρχιτεκτονικής του νευρωνικού μας ή και με διάφορες τεχνικές γενικοποίησης οι οποίες θα αναλυθούν πιο μετά στα πλαίσια αυτής της διπλωματικής. Βέβαια πρέπει να είμαστε προσεκτικοί γιατί αν το παρακάνουμε μπορεί να έχουμε να αντιμετωπίσουμε το αντίθετο πρόβλημα δηλαδή την υποεκπαίδευση (underfitting) σε αυτή την περίπτωση το νευρωνικό δεν έχει εκπαιδευτεί με σωστό τρόπο και είναι ανίκανο να κατανοήσει τις συνδέσεις μεταξύ δεδομένων εισόδου και τις επιθυμητές εξόδους τους.



Εικόνα 3.7 Οπτικοποίηση του προβλήματος της υπερεκπαίδευσης και υποεκπαίδευσης

3.4 Αρχιτεκτονικές νευρωνικών δικτύων

Η επιλογή της αρχιτεκτονικής ενός νευρωνικού δικτύου είναι υψίστης σημασίας, όταν λεμέ αρχιτεκτονική αναφερόμαστε στο βάθος, πλάτος και τις συνδέσεις των κόμβων του

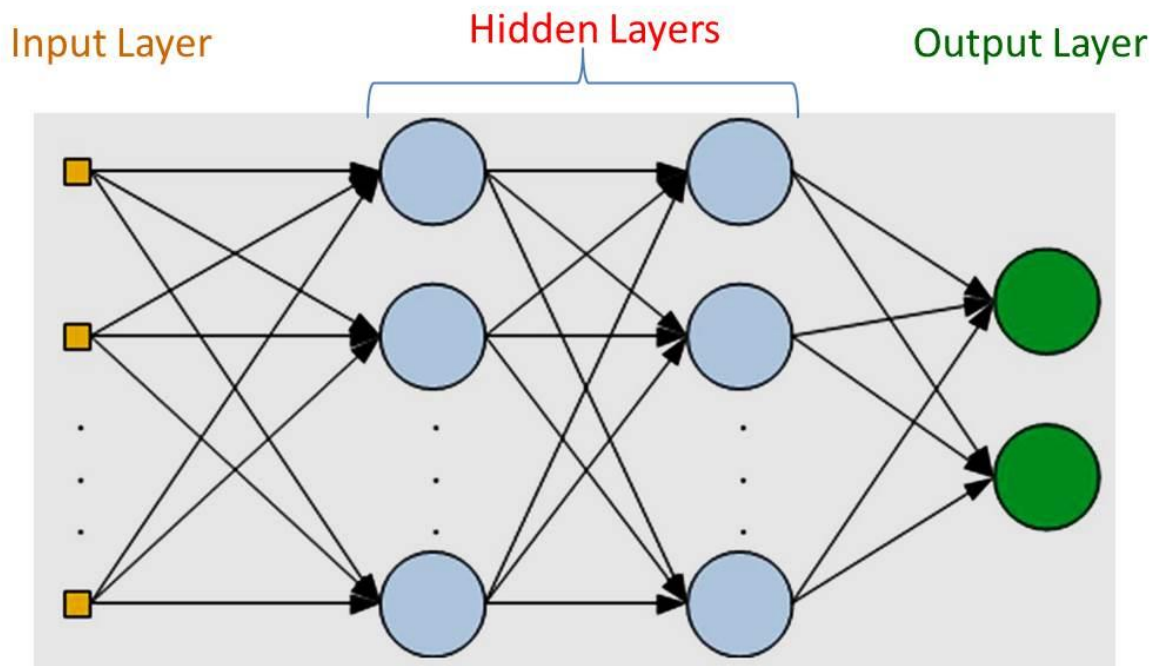
δικτύου.(Dayhoff, 1990; Luo et al., n.d.) Παρακάτω παρουσιάζονται τα πιο δημοφιλή είδη βαθιών νευρωνικών δικτύων.

3.4.1 Πολυεπίπεδα Perceptrons (multilayer Perceptrons)

Είναι ένα εμπρός τροφοδότησης τεχνητό νευρωνικό δίκτυο, το οποίο αποτελείται από τουλάχιστον τρία επίπεδα.(Taud & Mas, 2018) Προφανώς όπως έχει είδη αναφερθεί αν αποτελείται από τρία επίπεδα είναι ένα ρηχό νευρωνικό δίκτυο και αν αποτελείται από παραπάνω από τρία είναι ένα βαθύ νευρωνικό δίκτυο. Όλα τα επίπεδα είναι πλήρως συνδεδεμένα και κάθε νέο επίπεδο είναι ένα σύνολο μη γραμμικών συναρτήσεων ενός αθροίσματος όλων των εξόδων του προηγούμενου επιπέδου. Τα δεδομένα ρέουν από το επίπεδο εισόδου προς το επίπεδο εξόδου. . Οι κυριότερες περιπτώσεις χρήσης του MLP είναι η ταξινόμηση προτύπων, η αναγνώριση, η πρόβλεψη και η προσέγγιση. Όσον αφορά την πρόβλεψη αν και είναι τα πιο απλοϊκά είδη νευρωνικών δικτύων έχουν τα έξεις θετικά χαρακτηριστικά:

- Ανθεκτικότητα στην φασαρία
- Μη γραμμικότητα
- Πολυμεταβλητές εισόδους
- Πολυβηματικές προβλεψεις

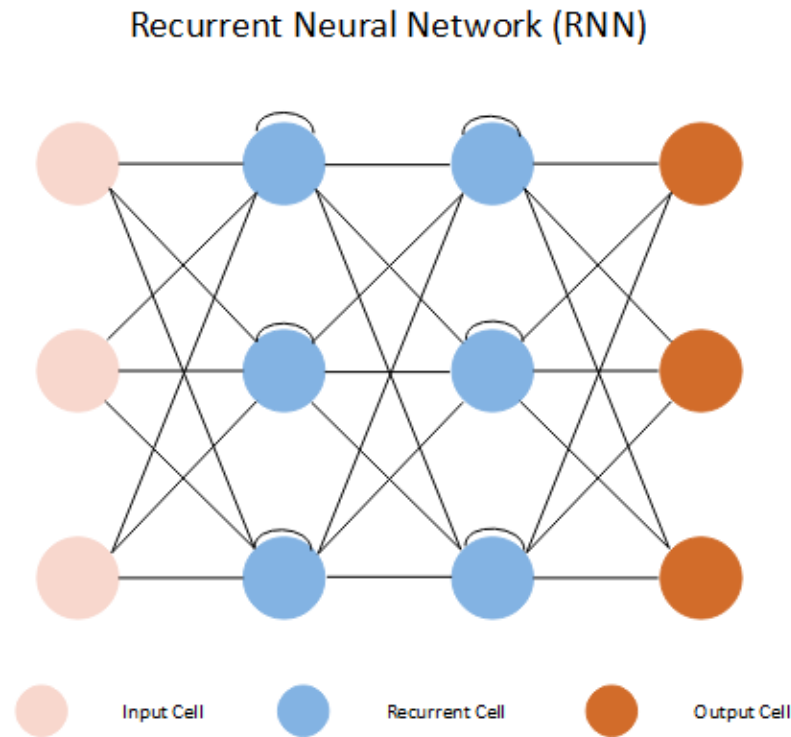
Το μεγαλύτερο μειονέκτημα τους είναι πως πρέπει να δοθεί ρητά η χρονική εξάρτηση εισόδου με προβλέψεις δηλαδή το lag



Εικόνα 3.8 Οπτικοποίηση σε γράφημα ενός πολυεπίπεδα perceptrons (MLP)

3.4.2 Επαναλαμβανόμενο νευρωνικό δίκτυο (RNN)

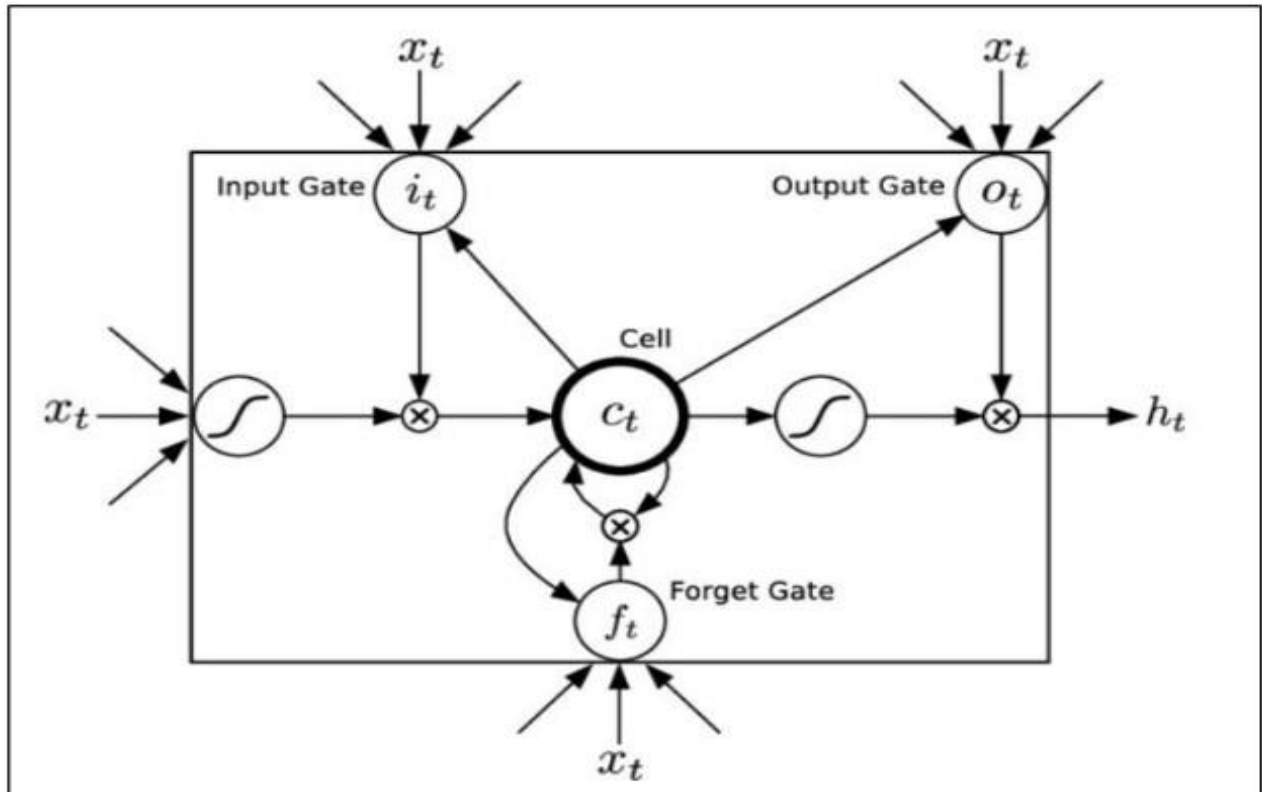
Ένα επαναλαμβανόμενο νευρωνικό δίκτυο είναι μπορεί να εμφανίσει κύκλους στις συνδέσεις των κόμβων του. (Phenomena & 2020, n.d.) Έχει την ικανότητα να επιδεικνύει χρονική δυναμική συμπεριφορά διότι η έξοδος από μερικούς κόμβους μπορεί να επηρεάσει την επακόλουθη είσοδο στους ίδιους τους κόμβους. Δέχονται διαδοχικά δεδομένα η δεδομένα χρονοσειρών και χρησιμοποιούνται κυρίως για προβλέψεις χρονοσειρών, αναγνώριση ομιλίας και επεξεργασία φυσικής γλώσσας. Το μεγαλύτερο πλεονέκτημα τους είναι ότι μπορούν να έχουν δυναμικό lag συνεπώς δεν χρειάζεται να δοθεί ρητά στην αρχή όπως στα πολυεπίπεδα perceptrons όπως και ότι υποστηρίζουν με φυσικό τρόπο χρονοσειρές ως δεδομένα εισόδου. Στην πραγματικότητα όμως εμφανίζουν ένα πολύ σημαντικό πρόβλημα αυτό των μακροχρονίων εξαρτήσεων, τα προβλήματα της εξαφανιζόμενης ή εκρηκτικής κλίσης εμφανίζονται όταν η κλίση γίνεται πολύ μεγάλη ή πολύ μικρή κατά την εκτέλεση του αλγορίθμου πίσω διαδρομής. Αυτό εμφανίζεται στα RNN διότι έχουν επαναλαμβανόμενες συνδέσεις οι οποίες τους επιτρέπουν να αποθηκεύουν πληροφορίες από προηγούμενα χρονικά βήματα.



Εικόνα 3.9 Οπτικοποίηση σε γράφημα ενός επαναλαμβανομένου νευρωνικού δικτύου

3.4.3 Νευρωνικά Δίκτυα Μακράς Βραχύχρονης Μνήμης (LSTM)

Είναι μια κατηγορία των RNN και ειδικεύεται στην εξαφάνιση του προβλήματος των μακροχρονίων εξαρτήσεων. (Staudemeyer & Morris, 2019; Yu et al., n.d.) Έχει ταυτόχρονα μακροπρόθεσμη και βραχυπρόθεσμη μνήμη όπως φαίνεται και στο όνομα του, η αρχιτεκτονική του LSTM αποτελείται από τρία κομμάτια κάθε ένα υπεύθυνο για διαφορετική λειτουργία. Το πρώτο κομμάτι είναι υπεύθυνο για να επιλέξει αν η πληροφορία που έρχεται αξίζει να απομνημονευθεί ή είναι αδιάφορη, το δεύτερο κομμάτι είναι υπεύθυνο για να μάθει την πληροφορία εισόδου και το τρίτο κομμάτι είναι υπεύθυνο για να εκσυγχρονίσει την πληροφορία από το ένα χρονικό βήμα στο επόμενο. Όπως ακριβώς και τα RNN έτσι και το LSTM έχει μια κρυφή κατάσταση $H(t-1)$ του προηγούμενου χρονικού βήματος και H_t του τωρινού χρονικού βήματος. Επιπρόσθετα έχει και μια κατάσταση κελύφους και το τωρινό και προηγούμενα χρονικό βήμα C_t και $C(t-1)$ αντίστοιχα.



Εικόνα 3.10 Οπτικοποίηση ενός *Lstm* επιπέδου

Εξίσωση forget gate

$$f_t = \sigma(x_t * U_f * H(t-1) * W_f) \quad (3,3)$$

- x_t : είσοδος του τωρινού χρονικού βήματος
- U_f : βαρύ που σχετίζονται με την είσοδο
- $H(t-1)$: Η κρυφή κατάσταση του προηγούμενου χρονικού βήματος
- W_f : Ο πίνακας βαρών της κρυφής κατάστασης

Πρόβλεψη της τιμής των κρουπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

Εν τελεί για να προκύψει το ft πραγματοποιείται μια σιγμοειδής συνάρτηση η οποία δίνει ένα αποτέλεσμα από το 0 έως και το 1. Όσο μικρότερο το ft τόσο περισσότερα ξεχνιούνται αν το ft είναι 1 τότε δεν ξεχνιέται τίποτα.

Εξίσωση input gate

$$It = \sigma(xt * Ui + H(t-1) * Wi) \quad (3,4)$$

- xt : είσοδος του τωρινού χρονικού βήματος
- Ui : βαρύ του πίνακα εισόδου
- $H(t-1)$: Η κρυφή κατάσταση του προηγούμενου χρονικού βήματος
- Wi : βαρύ πίνακα εισόδου της κρυφής κατάστασης

Για άλλη μια φορά χρησιμοποιούμε την σιγμοειδή συνάρτηση για να πάρουμε ένα αποτέλεσμα από το 0 έως και το 1. Πλέον η νέα πληροφορία για το Ct είναι μια συνάρτηση του xt και $H(t-1)$. Χρησιμοποιείται η \tanh αρά το αποτέλεσμα πλέον είναι από -1 έως και 1, αν το αποτέλεσμα είναι αρνητικό η πληροφορία αφαιρείται από το Ct αλλιώς αν είναι θετική προστίθεται.

Εξίσωση νέας πληροφορίας

$$Nt = \tanh(xt * Uc + H(t-1) * Wc) \quad (3,5)$$

Εξίσωση εκσυγχρονιστής κελύφους

$$Ct = ft * C(t-1) + It * Nt \quad (3,6)$$

Εξίσωση πύλης εξόδου

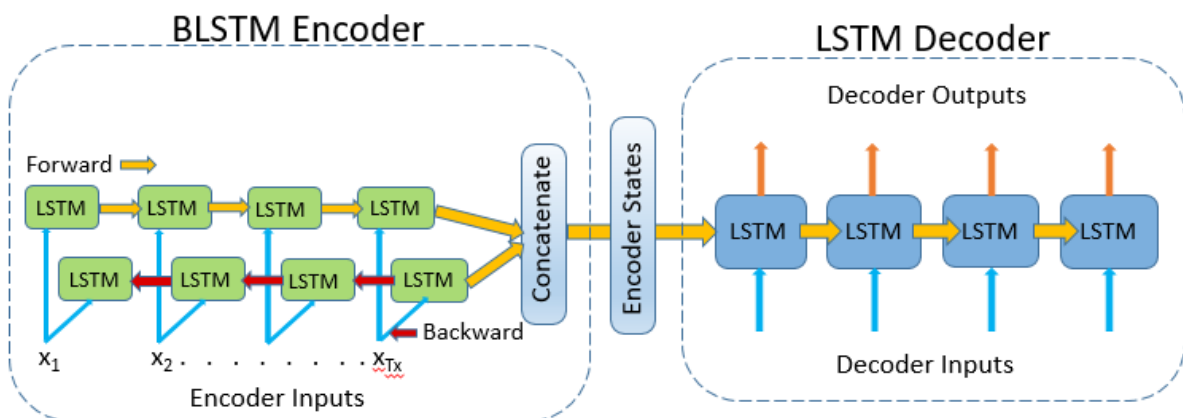
$$O_t = \sigma(x_t * U_0 + H(t-1) * W_0) \quad (3,7)$$

Για άλλη μια φορά η τιμή θα είναι από το 0 έως και το 1 λόγω της σιγμοειδής συνάρτησης. Τέλος θέλουμε την τιμή της τωρινής κρυφής κατάστασης η οποία δίνεται από την παρακάτω εξίσωση.

$$h_t = O_t * \tanh(Ct) \quad (3,8)$$

3.4.4 Bidirectional LSTMS

Τα LSTM διπλής κατεύθυνσης είναι και αυτά ένα είδος RNN, τα οποία έχουν την ιδιαιτερότητα ότι επεξεργάζονται δεδομένα από δυο κατεύθυνσης, από εμπρός και από πίσω.(Huang et al., 2015) Στα παραδοσιακά LSTM η πληροφορία κυλάει από το παρελθόν προς το μέλλον και κάνει προβλέψεις με αυτόν τον τρόπο. Όμως στα LSTM διπλής κατεύθυνσης το δίκτυο λαμβάνει υπόψιν και το μέλλον το οποίο του επιτρέπει να καταλαβαίνει εξαρτήσεις και από τις δυο κατεύθυνσης. Αποτελείται από δυο LSTM το ένα λαμβάνει την είσοδο από εμπρός και το άλλο από πίσω, αυτό του δίνει πληροφορίες για τα παρόντα αλλά και για τα μελλοντικά χρονικά βήματα. Συνήθως τέτοιοι είδους δίκτυα χρησιμοποιούνται για ανάλυση συναισθήματος(sentiment analysis), αυτόματη μετάφραση και ονομαστικές αναγνωρίσεις οντοτήτων. Είναι ακόμα πιο αποδοτικά στην αντιμετώπιση μακροχρονίων εξαρτήσεων από τα απλά LSTM καθώς ενσωματώνουν πληροφορίες και από τις δυο κατεύθυνσης και κάνουν πιο περίπλοκες προβλέψεις για ακολουθία δεδομένα.



Εικόνα 3.11 Διαφορά αποκωδικοποιητή Lstm και Bilstm

3.5 Υπερπαραμέτροι

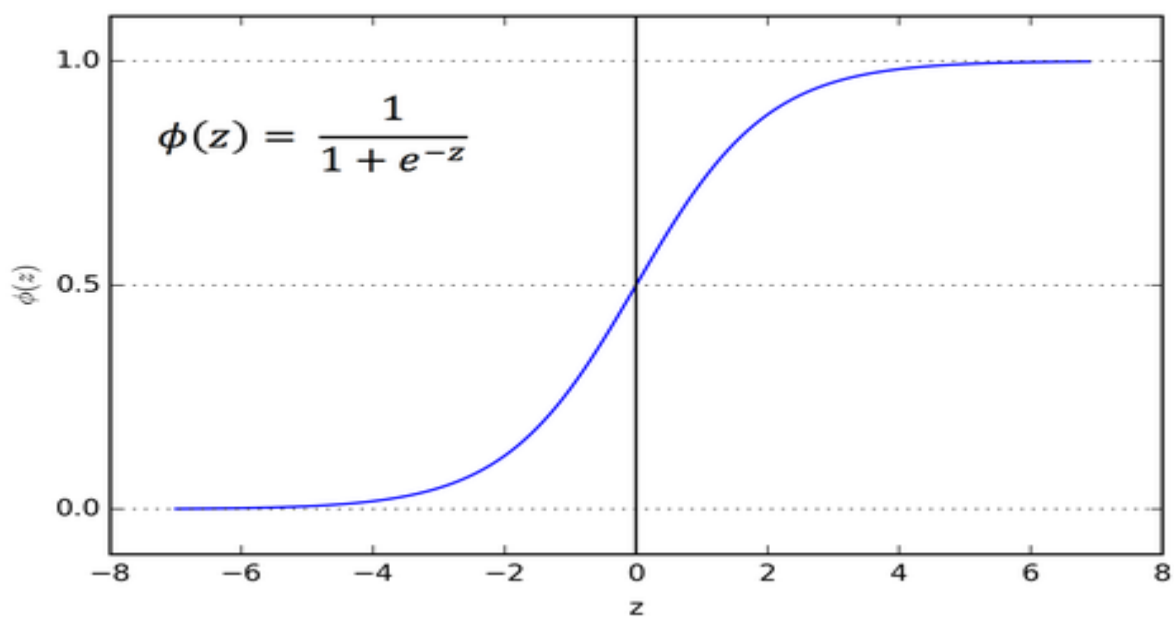
Οι υπερπαραμέτροι από τον αλγόριθμο εκμάθησης του μοντέλου όταν μαθαίνει αλλά δεν είναι κομμάτι του τελικού μοντέλου. Είναι εξωτερικοί του μοντέλου και η αρχικοποίηση του γίνεται κυρίως πριν αλλά και μερικές φορές κατά την διάρκεια της φάσης της εκπαίδευσης του μοντέλου. Η σωστή επιλογή τους έχει αξιοσημείωτες επιπτώσεις στο αν και ποσό καλά ένα μοντέλο θα εκπαιδευτεί και ποιες εν τελεί θα είναι οι προβλέψεις του. Σημαντικοί υπερπαραμέτροι για ένα μοντέλο είναι:

- Η συνάρτηση ενεργοποίησης
- Τεχνικές κανονικοποίησης
- Συνάρτηση σφάλματος
- Πλήθος εποχών
- Μέγεθος παρτίδας(batch size)
- Learning rate
- Πλήθος επιπέδων και πλήθος νευρώνων για κάθε επίπεδο

3.5.1 Η συνάρτηση ενεργοποίησης

Η συνάρτηση ενεργοποίησης αποφασίζει αν ένας νευρώνας θα ενεργοποιηθεί ή όχι, με άλλα λόγια αποφασίζει αν η είσοδος του νευρώνα στο δίκτυο υπάρχει λογίως να υπάρχει ή όχι. Δεν υπάρχει εγχειρίδιο για το ποια συνάρτηση ενεργοποίησης είναι η καλύτερη αλλά υπάρχουν διάφορα είδη συναρτήσεων ενεργοποίησης και η χρήση και η χρήση της κάθε μίας εξαρτάται από το πρόβλημα. Εφαρμόζεται είτε στην έξοδο ενός νευρώνα είτε στην έξοδο ενός επιπέδου νευρώνων και συνηθίζεται να χρησιμοποιείται η ίδια συνάρτηση ενεργοποίησης για όλα τα κρυφά επίπεδα. Ακολουθούν οι συναρτήσεις ενεργοποίησης που τείνουν να χρησιμοποιούνται το περισσότερο.

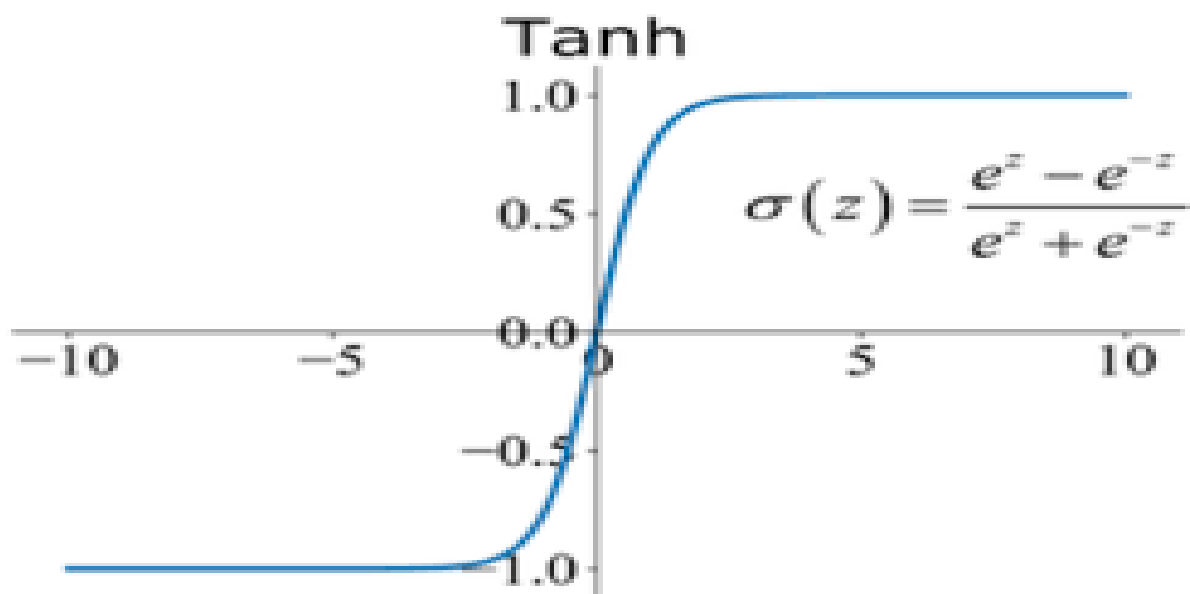
Sigmoid Function



Εικόνα 3.12 Εξίσωση και γραφική παράσταση της συνάρτησης sigmoid

Αυτή η συνάρτηση δέχεται μια τιμή ως είσοδο και δίνει ως έξοδο τιμές από το 0 έως και το 1, όσο πιο θετική είναι η είσοδο τόσο πιο κοντά θα είναι η έξοδο στο 1 και όσο πιο αρνητική τόσο πιο κοντά στο 0. Είναι ιδιαίτερα χρήσιμη για μοντέλα που θέλουμε να προβλέψουμε πιθανότητα ως έξοδο καθώς μας δίνει έξοδο στα ίδια πλαίσια τιμών.

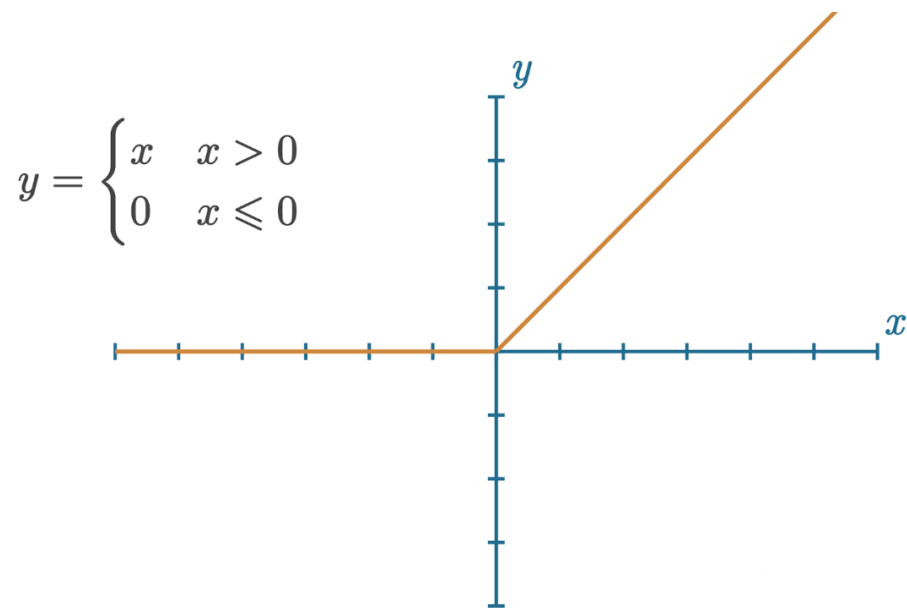
Tanh or hyperbolic tangent Function



Εικόνα 3.13 Εξίσωση και γραφική παράσταση της συνάρτησης Tanh

Είναι και αυτή μια λογιστική συνάρτηση όπως και η σιγμοειδής, αλλά αυτή μας δίνει τιμές από το -1 έως και το 1. Το κύριο πλεονέκτημα της από την σιγμοειδή είναι ότι οι αρνητικές τιμές θα λάβουν αρνητικές εξόδους και οι τιμές εισόδου που είναι μηδέν θα λάβουν τιμή κοντά στο 0.

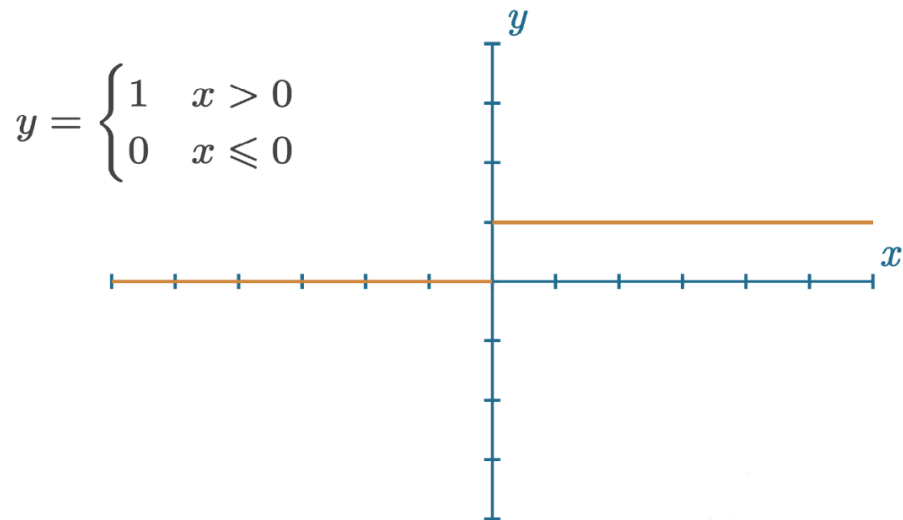
Relu Function



Εικόνα 3.14 Εξίσωση και γραφική παράσταση της συνάρτησης Relu

Είναι η πιο ευρέως χρησιμοποιούμενη συνάρτηση ενεργοποίησης στον κόσμο αυτή την στιγμή διότι προσφέρει ταχύτητα και αποτελεσματικότητα στο μοντέλο. Παρουσιάζει μη γραμμικότητα σε ένα μοντέλο βαθιάς μάθησης και βοηθάει στην επίλυση της εξαφανιζόμενης κλίσης κάθοδου.

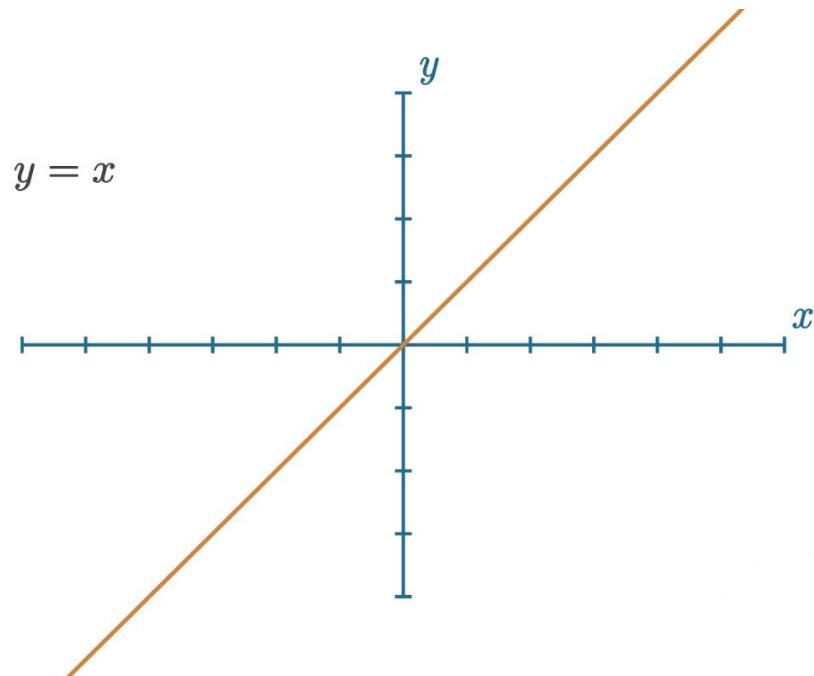
Binary step Function



Εικόνα 3.15 Εξίσωση και γραφική παράσταση της Βηματικής συνάρτησης

Αυτή η συνάρτηση ενεργοποίησης συγκρίνει την τιμή εισόδου με ένα κατώφλι, αν η τιμή είναι μεγαλύτερη από το κατώφλι τότε ο νευρώνας «πυροδοτείται» αλλιώς όχι.

Linear activation Function



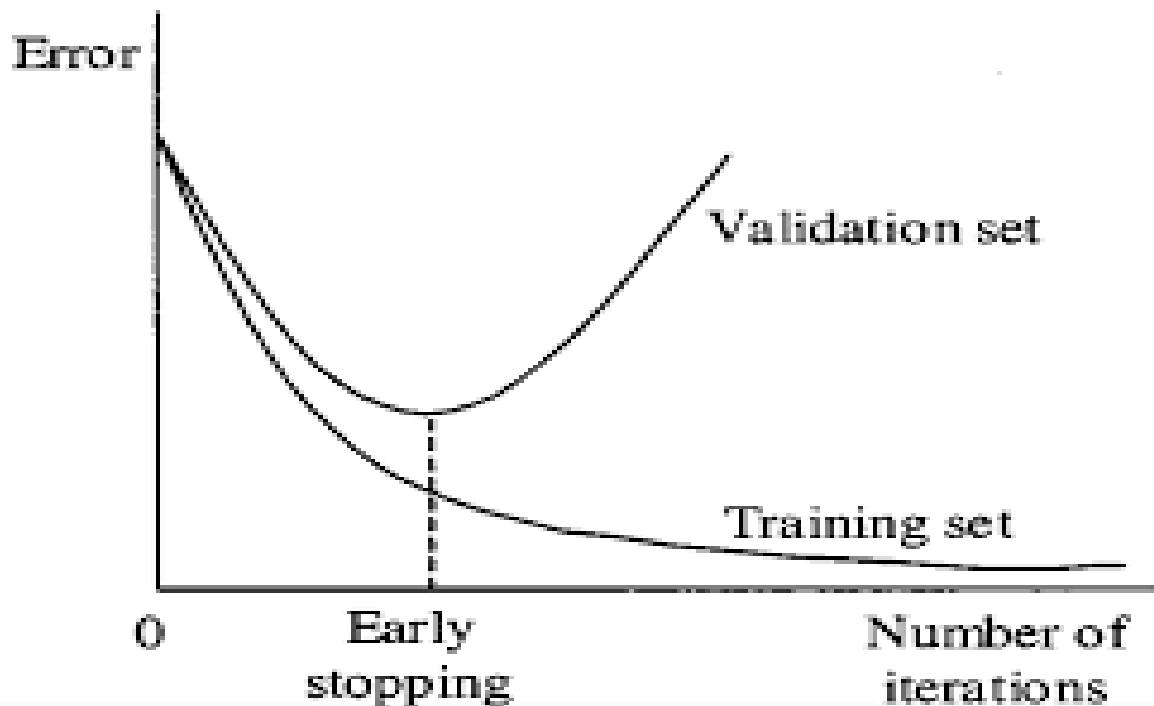
Εικόνα 3.16 Εξίσωση και γραφική παράσταση της γραμμικής συνάρτησης ενεργοποίησης

Επίσης γνωστή και ως συνάρτηση ταυτότητας σε αυτή την περίπτωση η έξοδος του νευρώνα είναι ανάλογη της εισόδου. Συνήθως εφαρμόζεται ως έξοδος του τελευταίου επιπέδου σε μοντέλα παλινδρόμησης.

3.5.2 Τεχνικές κανονικοποίησης

Πρόκειται για τεχνικές όπου μπορούν να βοηθήσουν το μοντέλο να αποφύγει το πρόβλημα της υπερεκπαίδευσης και κατά συνέπεια να βελτιώσουν αισθητά την ευκρίνεια του μοντέλου όταν αντιμετωπίζει καινούρια δεδομένα.

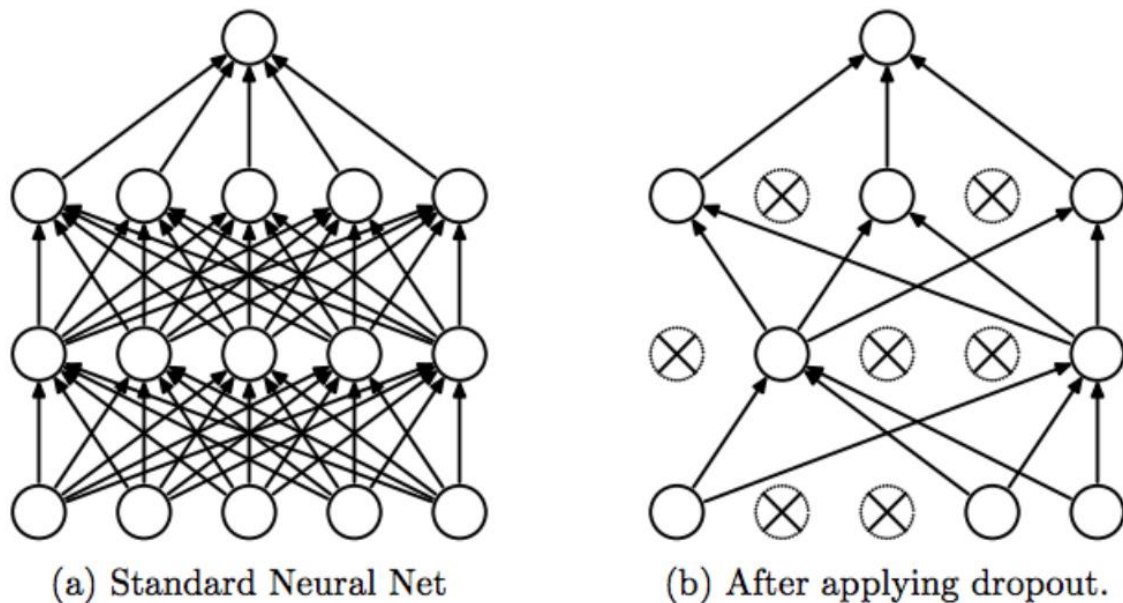
Πρόωρη διακοπή(Early Stopping)



Εικόνα 3.17 Οπτικοποίηση της πρόωρης διακοπής

Είναι μια τεχνική η οποία σταματάει την εκπαίδευση του νευρωνικού μοντέλου όταν μια παρακολουθούμενη ποσότητα έχει σταματήσει να βελτιώνεται , για παράδειγμα ακόμα και αν παρατηρείται ασθενής βελτίωση στην ευκρίνεια του μοντέλου στα δεδομένα εκπαίδευσης μερικές φορές μπορεί η ευκρίνεια στα δεδομένα ελέγχου να μειώνεται όπου είναι η πραγματική μετρική που μας ενδιαφέρει, κάτι τέτοιο δείχνει πως το μοντέλο επηρεάζεται από υπερεκπαίδευση, συνεπώς είναι καλύτερο να σταματήσει η εκπαίδευση πρόωρα ώστε να αποφευχθεί αυτό το φαινόμενο. Η πρόωρη διακοπή είναι μια από τις ευρέως εφαρμοζόμενες μεθόδους εθνοποίησης στην βαθιά μάθηση λόγω της απλότητάς της και της αποτελεσματικότητας της στην μείωση του χρόνου εκπαίδευσης.(Ji et al., n.d.)

Dropout



Εικόνα 3.18 Οπτικοποίηση της τεχνικής Dropout σε ένα βαθύ νευρωνικό δίκτυο

Κατά την εκπαίδευση του μοντέλου κάποιοι από τους νευρώνες τυχαία αγνοούνται ή «εγκαταλείπονται» κατά συνέπεια αυτό κάνει τα επίπεδα να μοιάζουν και να συμπεριφέρονται λες και είχαν διαφορετικό πλήθος κόμβων και συνδέσεων αναλόγως ποιοι νευρώνες θα αγνοηθούν. Αυτό έχει ως στόχο κάθε ενημέρωση των βαρών σε ένα επίπεδο κατά την εκπαίδευση σε ένα επίπεδο να πραγματοποιείται με διαφορετική οπτική ματιά. Η εγκαταλείπει είναι αποτελεσματική διότι κάνει την διαδικασία εκπαίδευση θορυβώδη κάτι που υποχρεώνει του νευρώνες ενός επιπέδου να έχουν περισσότερη η λιγότερη υπευθυνότητα για τις εισόδους και βοηθά στην καταπολέμηση της υπερεκπαίδευσης. (Srivastava et al., n.d.; Toronto & 2013, 2013)

3.5.3 Συνάρτηση σφάλματος

Οι επιλογή της κατάλληλης συνάρτησης σφάλματος είναι ζωτικής σημασίας για τα νευρωνικά διότι είναι άμεσα υπεύθυνες για το ποσά καλά θα εκπαιδευτεί ένα νευρωνικό δίκτυο στα δεδομένα που του δίνονται. Αυτές λοιπόν οι συναρτήσεις συγκρίνουν τον στόχο μας με την πρόβλεψη και μπορούν να διαχωριστούν σε δυο κυρίες κατηγορίες αναλόγως με το τι προσπαθεί να προβλέψει το νευρωνικό δίκτυο, σε παλινδρομικές συναρτήσεις σφάλματος και σε κατηγορηματικές συναρτήσεις σφάλματος.

παλινδρομικές συναρτήσεις σφάλματος

Χρησιμοποιούνται σε προβλήματα παλινδρόμησης και οι δυο πιο συχνά εφαρμοζόμενες είναι το σφάλμα αθροίσματος τετράγωνων (Sum of Squares error) , το μέσο τετραγωνικό λάθος (Mean squared error) και το μέσο απολυτό λάθος (Mean absolute error)

Σφάλμα αθροίσματος τετράγωνων

Αναπαριστά των τετραγωνικών λαθών επίσης γνωστό και ως υπολειπόμενα άθροισμα τετράγωνων και είναι η διαφορά μεταξύ του στόχου και τις πρόβλεψης. Περιγράφεται από την παρακάτω εξίσωση:

$$SSE = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (3,7)$$

- Y_i : Οι στόχοι που προσπαθεί να προβλέψει το μοντέλο
- \hat{Y}_i : Οι τιμές που προβλέπει το μοντέλο

Μέσο τετραγωνικό λάθος

Υπολογίζει τον μέσο ορό των τετράγωνων των σφαλμάτων μεταξύ του στόχου και των προβλέψεων και περιγράφεται από την παρακάτω εξίσωση:

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (3,8)$$

- n: το πλήθος των συγκρίσεων

Ωστόσο παρουσιάζει ένα μεγάλο μειονέκτημα είναι πολύ ευαίσθητη σε ακραίες περιπτώσεις δηλαδή όταν η πρόβλεψη είναι πολύ μακριά από τον στόχο.

Μέσο απολυτό λάθος

Υπολογίζει τον μέσο ορό των απολυτών διαφορών των στόχων με τις προβλέψεις του μοντέλου, χρησιμοποιείται ως εναλλακτική του μέσου τετραγωνικού λάθους διότι αντιμετωπίζει καλύτερα τις ακραίες περιπτώσεις και περιγράφεται από την παρακάτω εξίσωση:

$$MAE = \frac{\sum_{i=1}^n |\hat{Y}_i - Y_{i_i}|}{n} \quad (3,9)$$

Κατηγορηματικές συναρτήσεις σφάλματος

Χρησιμοποιούνται σε κατηγορηματικά προβλήματα και οι δυο πιο συχνά εφαρμοζόμενες είναι η δυαδική διασταυρούμενη εντροπία (Binary cross-Entropy) και κατηγορηματική διασταυρούμενη εντροπία (Categorical cross-Entropy)

Δυαδική διασταυρούμενη εντροπία

Χρησιμοποιείται από ένα μοντέλο το οποίο δέχεται μια είσοδο και προσπαθεί να την κατατάξει σε μια από δυο κατηγορίες και περιγράφεται από την παρακάτω εξίσωση:

$$H_p(q) = -\frac{1}{N} \sum_1^n y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \quad (3,10)$$

Κατηγορηματική διασταυρουμένη εντροπία

Είναι μια εναλλακτική της δυαδικής διασταυρουμένης εντροπίας με την μονή διαφορά πως έχουμε να κάνουμε με περισσότερες από δυο κατηγορίες και περιγράφεται από την παρακάτω εξίσωση:

$$Loss = - \sum_{i=1}^n y_i \cdot \log \hat{y}_i \quad (3,11)$$

Chapter 4 Ανάλυση και επεξεργασία δεδομένων εισόδου

4.1 Εισαγωγή

Τα δεδομένα αυτής της διπλωματικής είναι η καθημερινή τιμή που είχαν τα κρυπτονομίσματα bitcoin και Ethereum από τις 9 Νοέμβριου του 2017 (δηλαδή την πρώτη ημέρα που έχουμε διαθέσιμα δεδομένα για το eth) μέχρι και σήμερα και περιγράφονται από τις παρακάτω γραφικές παραστάσεις.

Λαμβάνονται με τον παρακάτω κώδικα:

```
crypto = 'BTC'
crypto2 = 'ETH'
against_crypto = 'USD'
start = dt.date(2017, 11, 9)
end = dt.date.today()

df1 = yf.download(f'{crypto}-{against_crypto}', start=start, end=end)
df2 = yf.download(f'{crypto2}-{against_crypto}', start=start, end=end)

[*****100%*****] 1 of 1 completed
[*****100%*****] 1 of 1 completed
```

Εικόνα 4.1 Συγκέντρωση δεδομένων για τα κρυπτονομίσματα

Και οπτικοποιούνται με τον παρακάτω κώδικα:

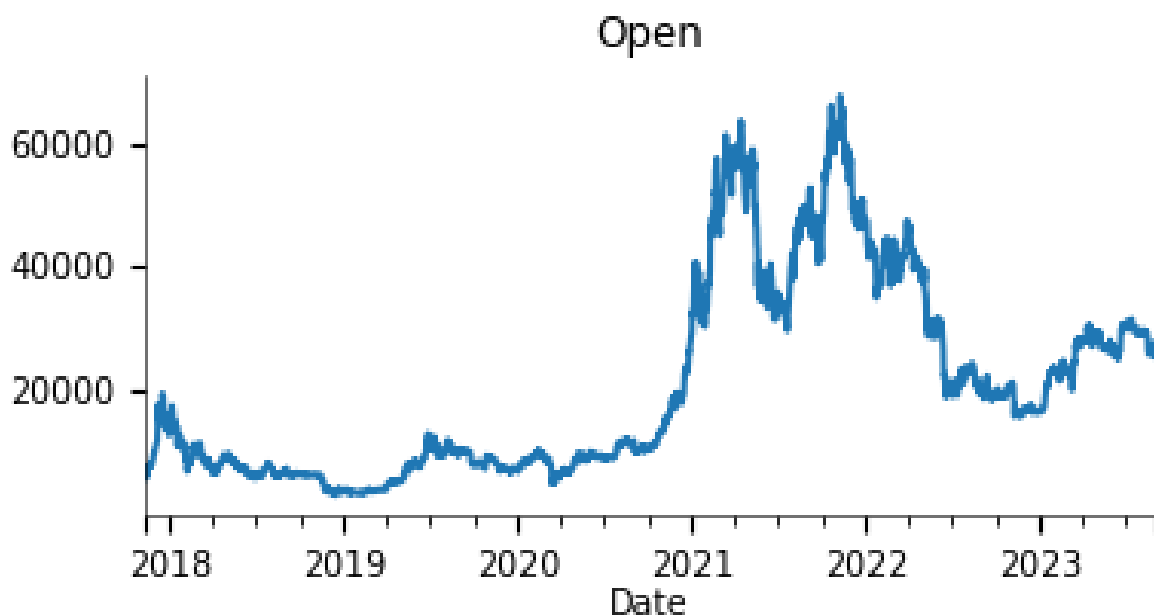
```
btc['Open'].plot(figsize=(20,8),color='blue')
plt.show()
```

Εικόνα 4.2 Κώδικας python για κατασκευή διαγράμματος του Bitcoin

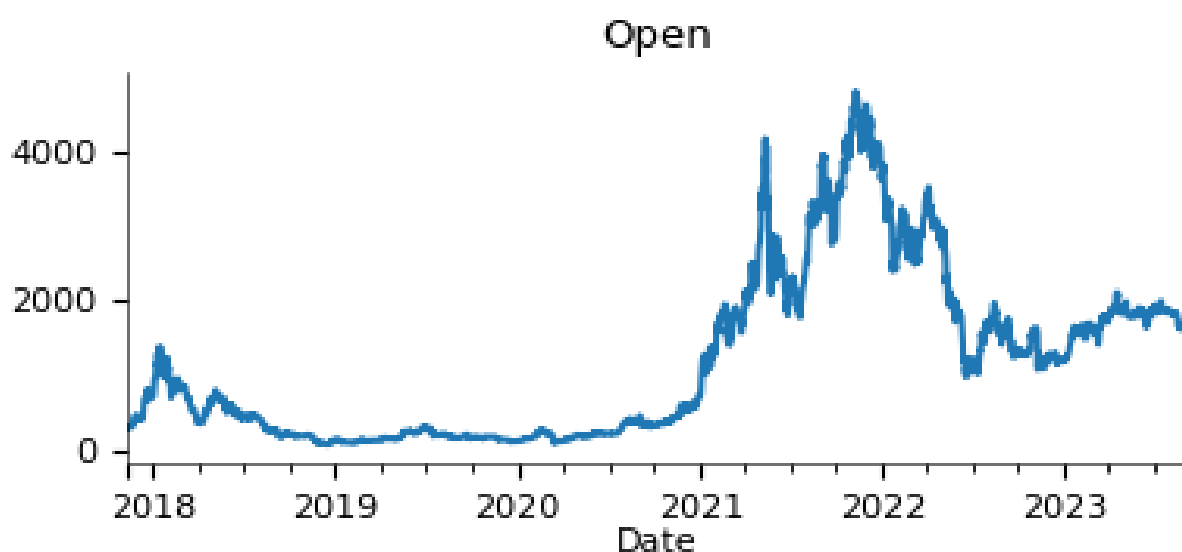
```
eth['Open'].plot(figsize=(20,8),color='blue')
plt.show()
```

Εικόνα 4.3 Κώδικας python για κατασκευή διαγράμματος του Ethereum

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου



Εικόνα 4.4 Η χρονοσειρά του Bitcoin από το 2017 μέχρι σήμερα



Εικόνα 4.5 Η χρονοσειρά του Ethereum από το 2017 μέχρι σήμερα

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

Για να επεξεργαστούν τα νευρωνικά δίκτυα τις παραπάνω χρονοσειρές πρέπει πρώτα να εφαρμοστεί η τεχνική του κινουμένου παραθύρου όπου έχει είδη εξηγηθεί σε προηγούμενο κεφάλαιο ώστε να μπορούν να διαχειριστούν ως δεδομένα παλινδρόμησης. Σε αυτό το κεφάλαιο θα εξηγηθεί ο τρόπος με τον οποίο επεξεργάστηκαν τα αρχικά δεδομένα και πως επιλέχθηκαν υπερπαραμέτροι όπως το lag κάνοντας ανάλυση δεδομένων.

4.2 Βιβλιοθήκες(modules) που χρησιμοποιήθηκαν για επεξεργασία και ανάλυση

- Numpy: Βιβλιοθήκη που παρέχει ένα αντικείμενο πολυδιάστατου πίνακα και προσφέρει διάφορους τρόπους για να τον διαχειριστείς.
- Pandas: Βιβλιοθήκη που χρησιμοποιείται για σύνολα δεδομένων, έχει συναρτήσεις για ανάλυση, κάθαρση, εξερεύνηση και χειραγώγηση δεδομένων.
- Matplotlib: Βιβλιοθήκη που χρησιμοποιείται για την δημιουργία οπτικοποιήσεων.
- Functools: Module για συναρτήσεις υψηλότερης τάξης που εφαρμόζεται σε άλλες συναρτήσεις, στα πλαίσια της διπλωματικής χρησιμοποιήθηκε για χειραγώγηση δεδομένων.
- Math: Module το οποίο παρέχει συναρτήσεις οι οποίες είναι χρήσιμες στην θεωρία αριθμών.
- Yfinance: Βιβλιοθήκη η οποία επιτρέπει την εύκολη εγκατάσταση δεδομένων από το Yahoo Finance, χρησιμοποιήθηκε για την εγκατάσταση των χρονοσειρών των κρυπτονομισμάτων.
- Datetime: Module το οποίο παρέχει classes για διαχείριση ημερομηνιών και χρόνου.
- Statsmodels: Είναι ένα πακέτο της python το οποίο χρησιμοποιείται για εξερεύνηση δεδομένων, υπολογισμό στατιστικών μοντέλων και υπολογισμό στατιστικών υποθέσεων, χρησιμοποιήθηκε για την ανάλυση δεδομένων.
- Scipy: Βιβλιοθήκη η οποία χρησιμοποιείται για την επίλυση μαθηματικών και επιστημονικών προβλημάτων, χρησιμοποιήθηκε για την ανάλυση δεδομένων.

4.3 Ανάλυση δεδομένων

Για την σωστή επιλογή υπερπαραμέτρων είναι απαραίτητο να γνωρίζουμε τα δεδομένα τα οποία έχουμε να διαχειριστούμε και αυτό ακριβώς καθιστά την ανάλυση των δεδομένων απαραίτητο στάδιο για μια καλή πρόβλεψη. Αρχικά έχει χρησιμοποιηθεί η μέθοδος describe ώστε να αποκτήσουμε μια γενική ιδέα για τα δεδομένα μας, στην συνέχεια έχει υπολογιστεί η κατά ποσοστό αλλαγή της τιμής των χρονομέτρων ανά χρονικό βήμα δηλαδή το return τους.

Τα return του BTC και ETH αντίστοιχα υπολογίζονται με τον παρακάτω κώδικα:

```
3] btc_return = round(np.log(btc).diff() * 100, 2)
   btc_return.dropna(inplace = True)
   eth_return = round(np.log(eth).diff() * 100, 2)
   eth_return.dropna(inplace = True)
```

Εικόνα 4.6 Κώδικας python για τον υπολογισμό της ποσοστιαίας αλλαγής ανά χρονικό βήμα των κρυπτονομισμάτων

Και η οπτικοποίηση του ιστογράμματος για τον ρυθμό αλλαγής του BTC παράγεται με τον παρακάτω κώδικα:

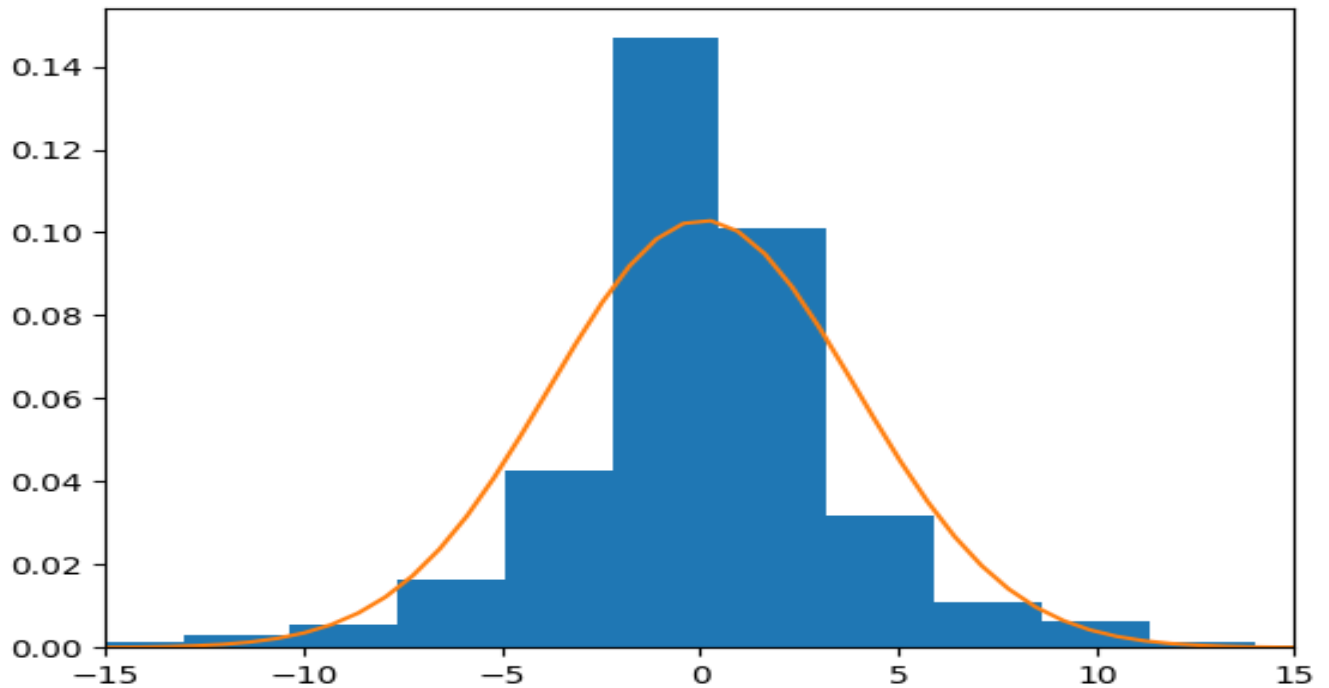
```
2] n, minmax, mean, var, skew, kurt = stats.describe(btc_return)
   mini, maxi = minmax
   std = var ** .5

   x = norm.rvs(mean, std, n)

3] plt.hist(btc_return, bins =25, density = True)
   b = np.linspace(mini, maxi, 100)
   plt.plot(b,stats.norm.pdf(b, mean, std))
   plt.xlim(-15,15);
```

Εικόνα 4.7 Κώδικας Python για την κατασκευή ιστογράμματος των ρυθμών αλλαγής του Bitcoin

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου



Εικόνα 4.8 Ο ρυθμός αλλαγής της τιμής του Bitcoin ως ιστόγραμμα σε σχέση με το ακολουθούμε κανονική κατανομή

Από το παραπάνω ιστόγραμμα του return του bitcoin μπορούμε να συμπεράνουμε πως η χρονοσειρά μας συγκεντρώνεται πιο συχνά σε πολύ μικρές αλλαγές ανά χρονικό βήμα από ότι αν ακολουθούμε κανονική κατανομή το return αλλά επίσης οι ακραίες αλλαγές της τιμής είναι πιο συχνές και πιο ακραίες.

Αντιστοίχως ο κώδικας για την οπτικοποίηση του ρυθμού αλλαγής του ETH:

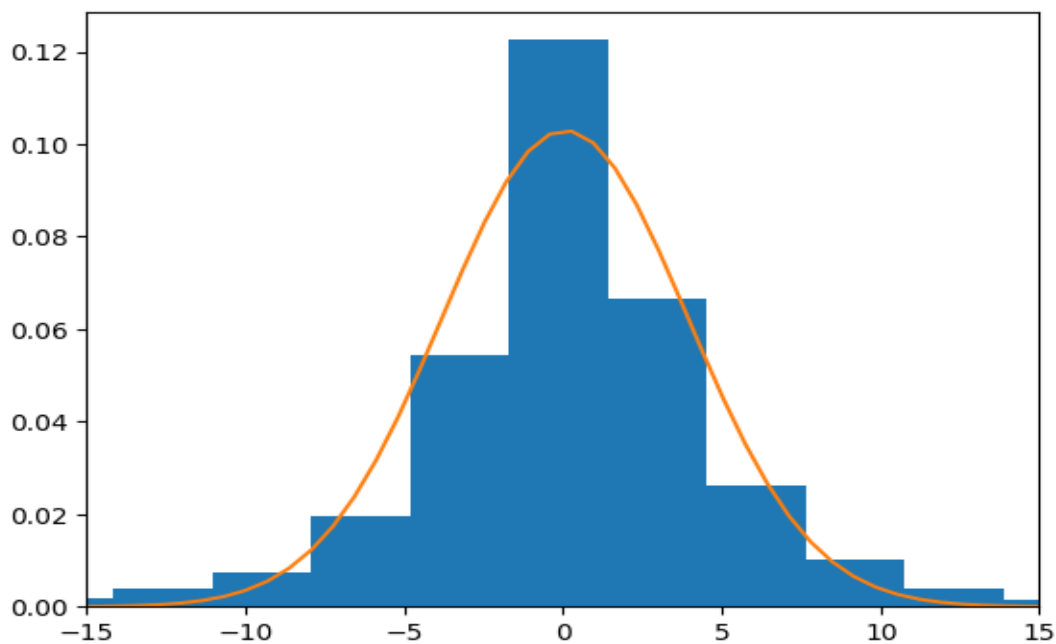
Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

```
n, minmax, mean, var, skew, kurt = stats.describe(eth_return)
mini, maxi = minmax
std = var ** .5

x = norm.rvs(mean, std, n)

plt.hist(eth_return, bins =25, density = True)
b = np.linspace(mini, maxi, 100)
plt.plot(b,stats.norm.pdf(b, mean, std))
plt.xlim(-15,15);
```

Εικόνα 4.9 Κώδικας Python για την κατασκευή ιστογράμματος των ρυθμών αλλαγής του Ethereum



Εικόνα 4.10 Ο ρυθμός αλλαγής της τιμής του Ethereum ως ιστόγραμμα σε σχέση με το ακολουθούμε κανονική κατανομή

Το Eth ακολουθεί την ίδια λογική με το btc αλλά είναι ακόμα πιο πολύ συγκεντρωμένο στο κέντρο και στις άκρες δηλαδή είναι πιο απότομο στις αλλαγές του από το btc αν έχει αλλαγή αλλά συνήθως δεν θα έχει αλλαγή η αξία του.

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

Έχει επίσης εξετασθή το αν τα κρυπτονομίσματα χρησιμοποιώντας το Dickey Fuller test και τα αποτελέσματα ήταν αυτά που περίμενε κανένας βλέποντας απλώς ένα γράφημα των τιμών στο χρόνο δηλαδή πως δεν είναι στασιμά. Τα return από την άλλη μεριά είναι πάντα στασιμά

Ο κώδικας για την εξέταση στασιμότητας στα κρυπτονομίσματα καθώς και στα return τους παρουσιάζεται παρακάτω:

```
from statsmodels.tsa.stattools import adfuller

adfullerbtc = adfuller(btc)
print('ADF statistic for btc : %f' % adfullerbtc[0])
print('p-value for btc: %f' % adfullerbtc[1])
print('Critical Values for btc:')
for key, value in adfullerbtc[4].items():
    print('\t%s: %.3f' % (key, value))

if adfullerbtc[0] < adfullerbtc[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")

ADF statistic for btc : -1.514316
p-value for btc: 0.526418
Critical Values for btc:
    1%: -3.433
    5%: -2.863
    10%: -2.568
Failed to Reject Ho - Time Series is Non-Stationary
```

Εικόνα 4.11 Κώδικας Python για έλεγχο στασιμότητας του Bitcoin

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

```
adfullereth = adfuller(eth)
print('ADF statistic for eth: %f' % adfullereth[0])
print('p-value for eth: %f' % adfullereth[1])
print('Critical Values:')
for key, value in adfullereth[4].items():
    print('\t%s: %.3f' % (key, value))

if adfullereth[0] < adfullereth[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")

ADF statistic for eth: -1.491117
p-value for eth: 0.537901
Critical Values:
    1%: -3.433
    5%: -2.863
   10%: -2.567
Failed to Reject Ho - Time Series is Non-Stationary
```

Εικόνα 4.12 Κώδικας Python για έλεγχο στασιμότητας του Ethereum

```
adfullerbtc_return = adfuller(btc_return)
print('ADF statistic for btc : %f' % adfullerbtc_return[0])
print('p-value for btc: %f' % adfullerbtc_return[1])
print('Critical Values for btc:')
for key, value in adfullerbtc_return[4].items():
    print('\t%s: %.3f' % (key, value))

if adfullerbtc_return[0] < adfullerbtc_return[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")

ADF statistic for btc : -31.750068
p-value for btc: 0.000000
Critical Values for btc:
    1%: -3.433
    5%: -2.863
   10%: -2.567
Reject Ho - Time Series is Stationary
```

Εικόνα 4.13 Κώδικας Python για έλεγχο στασιμότητας του ρυθμού αλλαγής του Bitcoin

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

```
adfullereth_return = adfuller(eth_return)
print('ADF statistic for btc : %f' % adfullereth_return[0])
print('p-value for btc: %f' % adfullereth_return[1])
print('Critical Values for btc:')
for key, value in adfullereth_return[4].items():
    print('\t%s: %.3f' % (key, value))

if adfullereth_return[0] < adfullereth_return[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")

ADF statistic for btc : -13.848589
p-value for btc: 0.000000
Critical Values for btc:
    1%: -3.433
    5%: -2.863
    10%: -2.567
Reject Ho - Time Series is Stationary
```

Εικόνα 4.14 Κώδικας Python για έλεγχο στασιμότητας του ρυθμού αλλαγής του Ethereum

4.4 Επεξεργασία δεδομένων

Τα δεδομένα είναι οι τιμές που είχαν τα κρυπτονομίσματα ανά χρονικό βήμα, όμως ένα νευρωνικό Δίκτυο το οποίο εκπαιδεύεται με την μέθοδο της επιβλεπόμενης μάθησης, ως είσοδο (training data) δέχεται δεδομένα που έχουν την μορφή πινάκων (arrays). Για την δημιουργία αυτών των πινάκων έχει χρησιμοποιηθεί η τεχνική ολίσθησης παραθύρου με lag=60, το lag επιλέχθηκε πειραματικά καθώς συμπεράναμε από τα γραφήματα acf, pacf των χρονοσειρών πως πρόκειται για random walk χρονοσειρές όπου η επιλογή του lag δεν μπορεί να βρεθεί από την ανάλυση των χρονοσειρών. Επίσης έχει πραγματοποιηθεί scaling στην χρονοσειρά για την διευκόλυνση της εκπαίδευσης του νευρωνικού και τέλος πριν την αρχή της εκπαίδευσης έχει διαχωριστεί η χρονοσειρά σε 80% training και 20 % testing ώστε να δημιουργηθεί ένα αυθαίρετο κομμάτι που διαχωρίζεται το παρόν με το μέλλον καθώς δεν πρέπει να ταίςουμε πληροφορίες στο μοντέλο για το μέλλον.

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

Ο κώδικας για το scaling καθώς και για την δημιουργία των training και test sets σε σωστή μορφή φαίνεται παρακάτω:

```
scaler=StandardScaler()
prediction_days=60
end=math.floor(len(btc)*0.8)

btc_train=btc[prediction_days:end]
btc_test=btc[end-prediction_days+1:]
```

Εικόνα 4.15 Κώδικας Python για την κατάλληλη δημιουργία των training και Test sets

```
scaled_data=scaler.fit_transform(btc_train.values.reshape(-1,1))
scaled_test=scaler.transform(btc_test.values.reshape(-1,1))

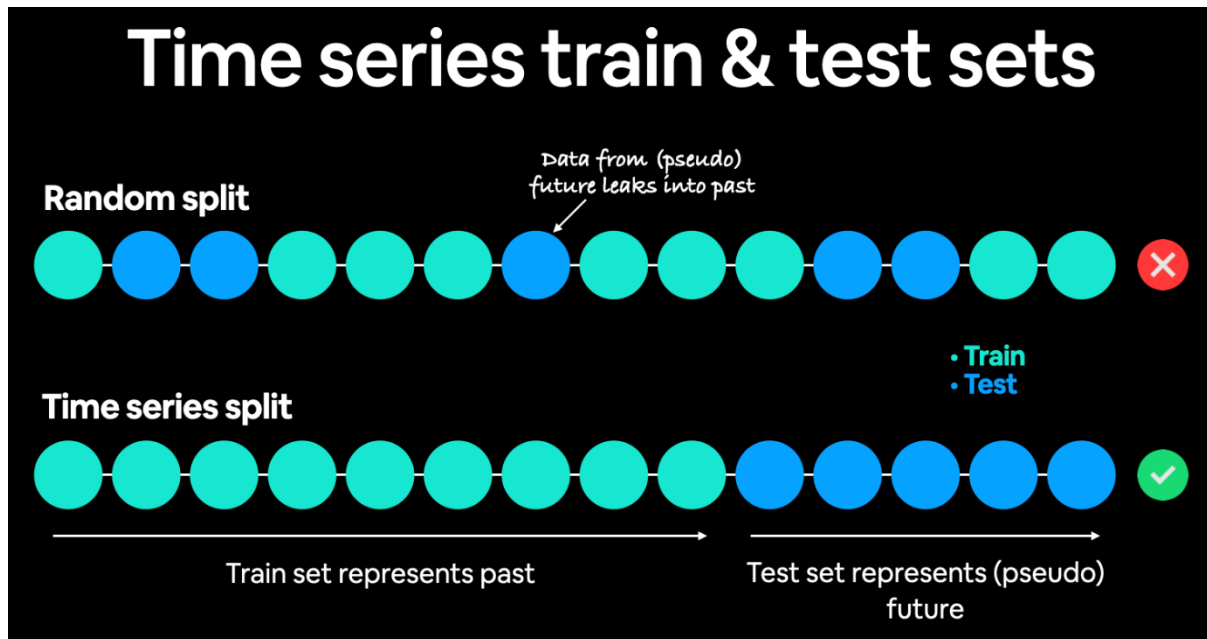
x_train,y_train=[],[]

for x in range(prediction_days,len(scaled_data)):
    x_train.append(scaled_data[x-prediction_days:x,0])
    y_train.append(scaled_data[x,0])
x_train,y_train=np.array(x_train),np.array(y_train)
x_train2=x_train
x_train=np.reshape(x_train,(x_train.shape[0],1,x_train.shape[1]))

x_test=[]

for x in range(prediction_days,len(btc_test)):
    x_test.append(scaled_test[x-prediction_days:x,0])
x_test=np.array(x_test)
x_test2=x_test
x_test=np.reshape(x_test,(x_test.shape[0],1,x_test.shape[1]))
```

Εικόνα 4.16 Κώδικας Python για την κατάλληλη δημιουργία των training και Test sets



Εικόνα 4.17 Σωστή και λάθος διαχώριση δεδομένων χρονοσειράς για εκπαίδευση και δοκιμή

Chapter 5 Πρόβλεψη χρονοσειρών

5.1 Εισαγωγή

Σε αυτό το στάδιο χρησιμοποιήθηκαν κατά κορών οι βιβλιοθήκες sklearn και tensorflow. Η sklearn είναι μια βιβλιοθήκη ανοιχτού κώδικα για ανάλυση δεδομένων καθώς και ο χρυσός κανόνας για μηχανική μάθηση στο οικοσύστημα της rython, καθώς με αυτήν είναι εύκολο να υλοποιηθούν μηχανικής μάθησης μοντέλα καθώς και στατιστικά μοντέλα. Για να εξηγηθεί καλύτερα η βιβλιοθήκη tensorflow όμως πρέπει πρώτα να γίνει καλύτερα κατανοητά το τι είναι ένας tensor από όπου πρέπει και το όνομα της η βιβλιοθήκη εξάλλου,

5.1.1 Tensors

Λοιπόν ένας tensor είναι ένα αντικείμενο το οποίο μπορεί να αναπαρασταθεί ως ένας πολυδιάστατος πίνακας, όμως κάτι τέτοιο είναι δυνατό να αναπαρασταθεί και από λίστα από λίστες από λίστες ποια είναι η διαφορά του? Αυτό μπορεί να γίνει ευκολά ξεκάθαρο με ένα παράδειγμα με τις διάφορες μια λίστας, ενός πίνακα και ενός μητρώου.

Μια λίστα μπορεί να αναπαρασταθεί ως:

```
List= [1,2,3,4]
```

Μια λίστα από λίστες μπορεί να αναπαρασταθεί ως:

```
ListofLists=[[1,2,3,4],[5,6,7,8]]
```

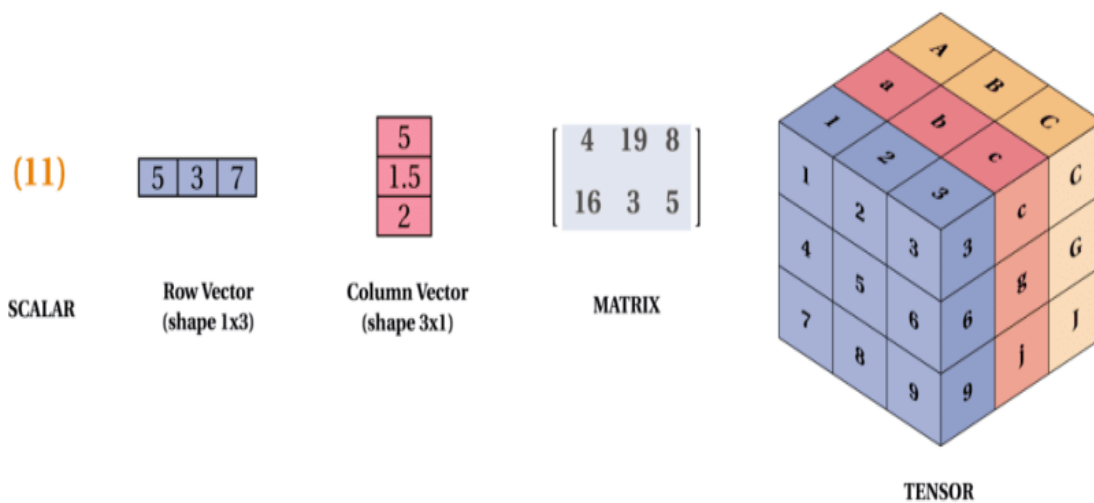
Μια λίστα από λίστες από λίστες μπορεί να αναπαρασταθεί ως:

```
ListofListsofLists=  
[  
[[1,2],[3,4]],  
[[5,6],[7,8]]  
]
```

Κάθε μια από τις παραπάνω λίστες μπορεί να θεωρηθεί ως πίνακας αλλά δεν είναι απαραίτητο να έχει το ίδιο μέγεθος η κάθε λίστα μέσα σε λίστα κάτι το οποίο είναι υποχρεωτικό σε ένα tensor.

Από την άλλη μεριά ένα μητρώο με ένα πίνακα διαφέρουν στο ότι το μητρώο είναι ένας δυσδιάστατος ορθογώνιος πίνακας δηλαδή όλα τα μητρώα είναι πίνακες αλλά όχι οι πίνακες μητρώα.(Magazine & 2014, 2014)

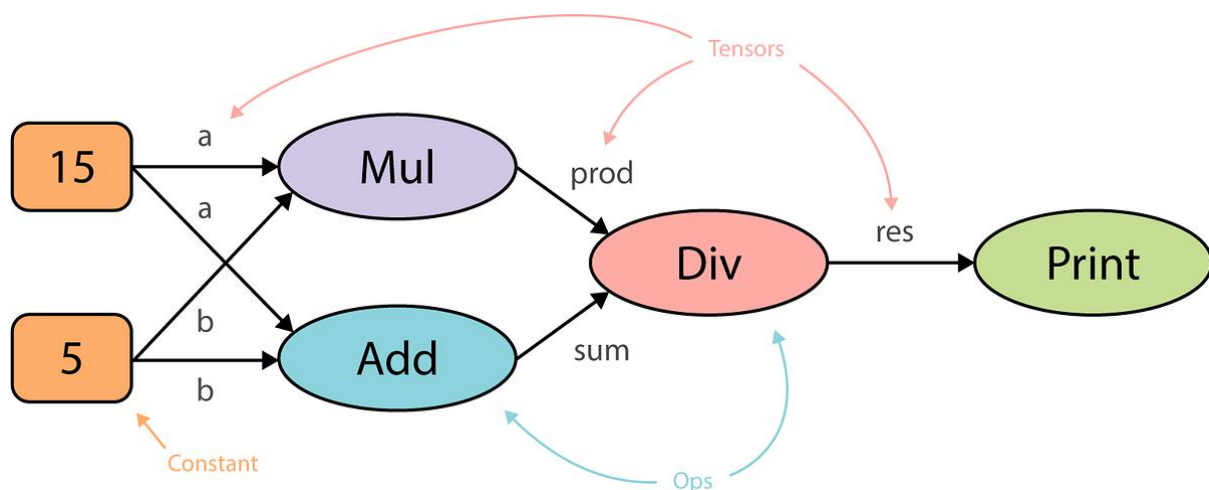
Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου



Εικόνα 5.1 Σύγκριση Tensor με διαφορετικά Data containers

5.1.2 Tensorflow

Η βιβλιοθήκη TensorFlow είναι μια ανοιχτού κώδικα βιβλιοθήκη (open-source) που αναπτύχθηκε από την Google κυρίως για εφαρμογές βαθιάς μάθησης (deep learning applications) και παρέχει υποστήριξη για εφαρμογές μηχανικής μάθησης. Ουσιαστικά κάνει υπολογισμούς χρησιμοποιώντας γραφήματα ροής δεδομένων όπου φέρουν tensors. (Wongsuphasawat et al., 2018)



Εικόνα 5.2 Γράφημα για το πως λειτουργεί η ροή δεδομένων στην βιβλιοθήκη Tensorflow

Η tensorflow αναπτύχθηκε για την απλοποίηση της διαδικασίας εγγραφής μανικής και βαθιάς μάθησης κάτι το οποίο καταφέρνει εκπληκτικά καθώς πλέον χάρις αυτήν τα περισσότερα γίνονται αυτόματα όπως ο προγραμματισμός του νευρώνα και η δημιουργία ενός νευρωνικού δικτύου. Ένα άλλο μεγάλο πλεονέκτημα της tensorflow είναι πως χρησιμοποιεί

Πρόβλεψη της τιμής των κρουπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

την γραφική μονάδα επεξεργασίας για να επιταχύνει τους υπολογισμούς που χρειάζονται για την εκπαίδευση ενός νευρωνικού δικτύου όπου χωρίς αυτήν θα ήταν μια ιδιαίτερα χρονοβόρα και περίπλοκη διαδικασία. (Silaparasetty, 2020)

5.2 Αρχιτεκτονικές νευρωνικών δικτύων

Η υλοποίηση των νευρωνικών δικτύων για την συγκεκριμένη διπλωματική εργασία έγινε με την χρήση του Google Collaboratory η αλλιώς “Collab”, όπως συνήθως αποκαλείται. Το Collab είναι ένα Cloud based περιβάλλον για Jupiter notebooks, το οποίο σου επιτρέπει να τρέχεις κώδικα στο διαδίκτυο στο browser. Ο λόγος που επιλέχτηκε ήταν για την διευκόλυνση όσον αφορά τις εξαρτήσεις στις διαφανές βιβλιοθήκες που χρησιμοποιήθηκαν στα πλαίσια της διπλωματικής. Το google collab έχει υποστήριξη για τις γλώσσες προγραμματισμού python, R και Julia.

Και τα τρία βαθιά νευρωνικά δίκτυα τα οποία υλοποιήθηκαν έχουν 5 επίπεδα, το πρώτο επίπεδο τους το επίπεδο εισόδου έχει πάντα τόσους νευρώνες όσο είναι το lag τα επόμενα 3 δηλαδή τα κρυφά επίπεδα έχουν 64,32 και 32 αντίστοιχα και τέλος το επίπεδο εξόδου έχει μόνο έναν νευρώνα που μας δίνει την πρόβλεψη. Το mlp χρησιμοποιεί ως συνάρτηση ενεργοποίηση την relu και τα lstm και bilstm χρησιμοποιούν την tanh, όλα χρησιμοποιούν ως optimizer τον adam της βιβλιοθήκης keras με learning rate= 0.001 και υπάρχει ένα dropout=0.2 για όλα τα επίπεδα εκτός από τα επίπεδα εξόδου και ως συνάρτηση σφάλματος χρησιμοποιείται η συνάρτηση μέσου τετραγωνικού σφάλματος (Mean Squared Error,MSE). Όλα τα βαθιά νευρωνικά δίκτυα εκπαιδεύονται για 100 εποχές το πολύ και αυτό διότι χρησιμοποιείται η τεχνική της πρόωρης διακοπής.

5.2.1 MLP

Ο κώδικας για την κατασκευή του MLP παρουσιάζεται παρακάτω καθώς και ένα γράφημα των επίπεδων του:

```
mlp=Sequential()

mlp.add(Dense(prediction_days, input_dim=prediction_days, activation='relu'))
mlp.add(Dropout(0.2))
mlp.add(Dense(64, activation='relu'))
mlp.add(Dropout(0.2))
mlp.add(Dense(32, activation='relu'))
mlp.add(Dropout(0.2))
mlp.add(Dense(32, activation='relu'))
mlp.add(Dropout(0.2))
mlp.add(Dense(1))

mlp.compile(loss='mean_squared_error', optimizer='adam', metrics = ['acc'])
mlp.fit(x_train2,y_train,epochs=100,batch_size=32,callbacks=[es],verbose=2)
```

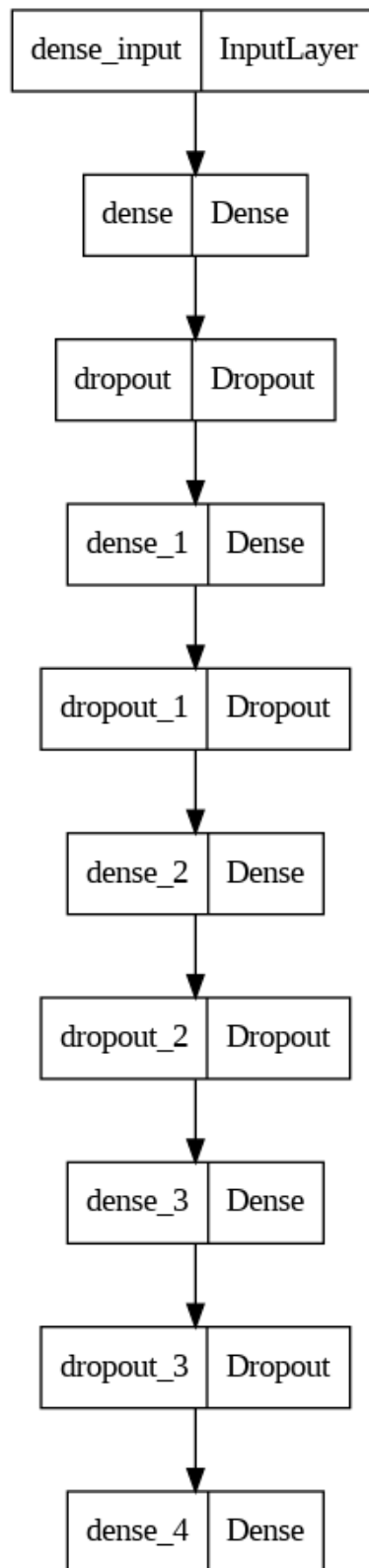
Εικόνα 5.3 Κώδικας Python για υλοποίηση του βαθιού νευρωνικού δικτύου αρχιτεκτονικής MLP

Και ο κώδικας για την δημιουργία του γραφήματος:

```
plot_model(mlp, to_file='model.png')
```

Εικόνα 5.4 Κώδικας Python για την δημιουργία γραφήματος αρχιτεκτονικής βαθιού νευρωνικού δικτύου τύπου MLP

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου



Εικόνα 5.5 Το βαθύ νευρωνικό δίκτυο που υλοποιήθηκε στα πλαίσια υλοποίησης της διπλωματικής αρχιτεκτονικής MLP

5.2.2 LSTM

Ο κώδικας για την κατασκευή του LSTM παρουσιάζεται παρακάτω καθώς και ένα γράφημα των επίπεδων του:

```
lstm=Sequential()  
lstm.add(LSTM(units=prediction_days,return_sequences=True,input_shape=(1,prediction_days)))  
lstm.add(Dropout(0.2))  
lstm.add(LSTM(units=64,return_sequences=True))  
lstm.add(Dropout(0.2))  
lstm.add(LSTM(units=32,return_sequences=True))  
lstm.add(Dropout(0.2))  
lstm.add(LSTM(units=32,return_sequences=False))  
lstm.add(Dropout(0.2))  
lstm.add(Dense(1))  
  
lstm.compile('adam','mean_squared_error')  
lstm.fit(x_train,y_train,epochs=100,batch_size=32,callbacks=[es],verbose=2)
```

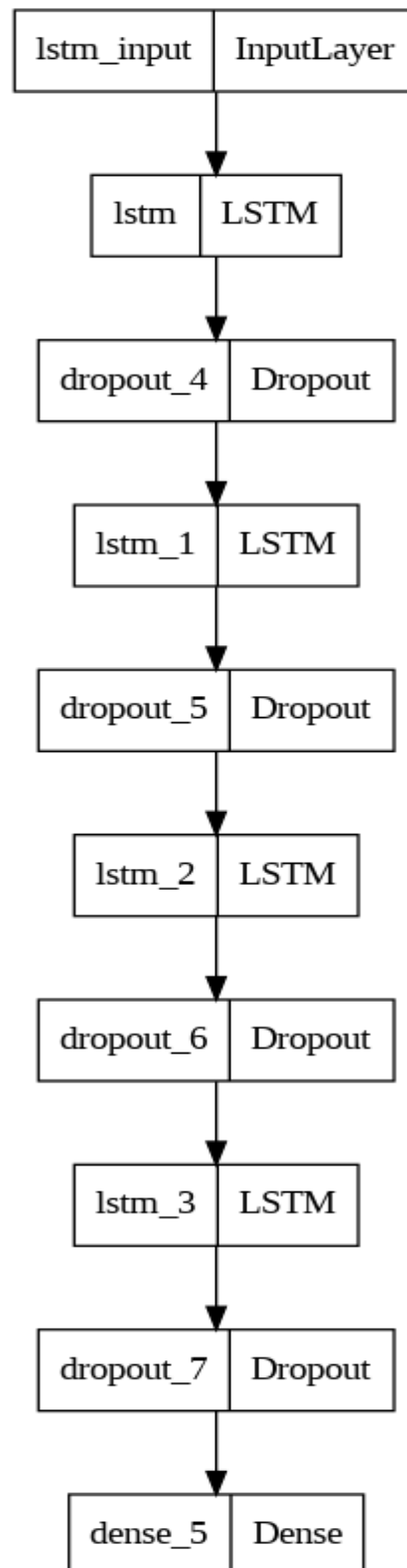
Εικόνα 5.6 Κώδικας Python για υλοποίηση του βαθιού νευρωνικού δικτύου αρχιτεκτονικής LSTM

Και ο κώδικας για την δημιουργία του γραφήματος:

```
] plot_model(lstm, to_file='model.png')
```

Εικόνα 5.7 Κώδικας Python για την δημιουργία γραφήματος αρχιτεκτονικής βαθιού νευρωνικού δικτύου τύπου LSTM

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου



Εικόνα 5.8 Το βαθύ νευρωνικό δίκτυο που υλοποιήθηκε στα πλαίσια υλοποίησης της διπλωματικής αρχιτεκτονικής LSTM

5.2.3 Bilstm

Ο κώδικας για την κατασκευή του BI-LSTM παρουσιάζεται παρακάτω καθώς και ένα γράφημα των επιπέδων του:

```
bilstm=Sequential()  
bilstm.add(Bidirectional(LSTM(units=prediction_days,return_sequences=True,input_shape=(1,prediction_days))))  
bilstm.add(Dropout(0.2))  
bilstm.add(Bidirectional(LSTM(units=64,return_sequences=True)))  
bilstm.add(Dropout(0.2))  
bilstm.add(Bidirectional(LSTM(units=32,return_sequences=True)))  
bilstm.add(Dropout(0.2))  
bilstm.add(Bidirectional(LSTM(units=32,return_sequences=False)))  
bilstm.add(Dropout(0.2))  
bilstm.add(Dense(1))  
bilstm.compile('adam','mean_squared_error')  
bilstm.fit(x_train,y_train,epochs=100,batch_size=32,callbacks=[es],verbose=2)
```

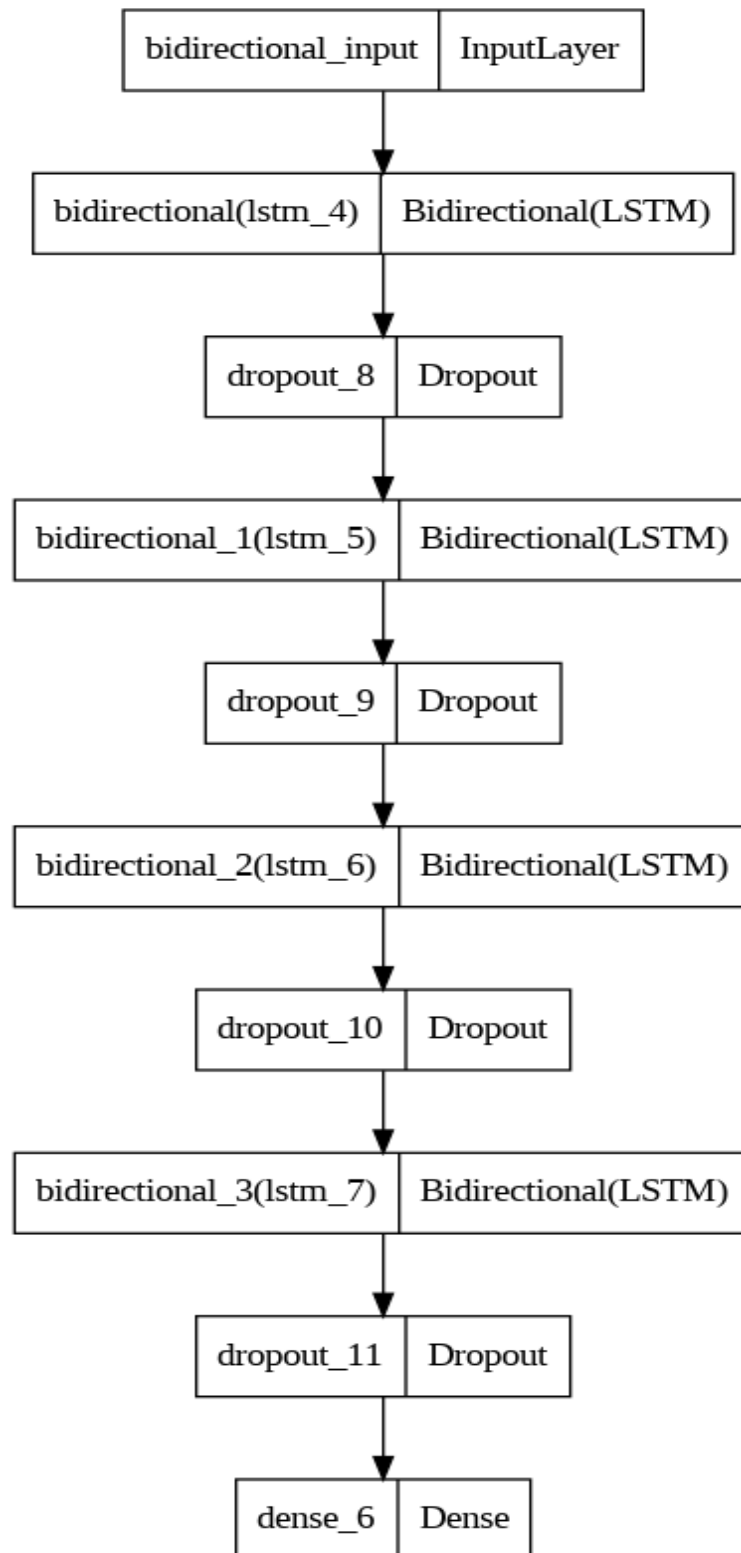
Εικόνα 5.9 Κώδικας Python για υλοποίηση του βαθιού νευρωνικού δικτύου αρχιτεκτονικής BI-LSTM

Και ο κώδικας για την δημιουργία του γραφήματος:

```
plot_model(bilstm, to_file='model.png')
```

Εικόνα 5.10 Κώδικας Python για την δημιουργία γραφήματος αρχιτεκτονικής βαθιού νευρωνικού δικτύου τύπου BI-LSTM

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου



Εικόνα 5.11 Το βαθύ νευρωνικό δίκτυο που υλοποιήθηκε στα πλαίσια υλοποίησης της διπλωματικής αρχιτεκτονικής Bilstm

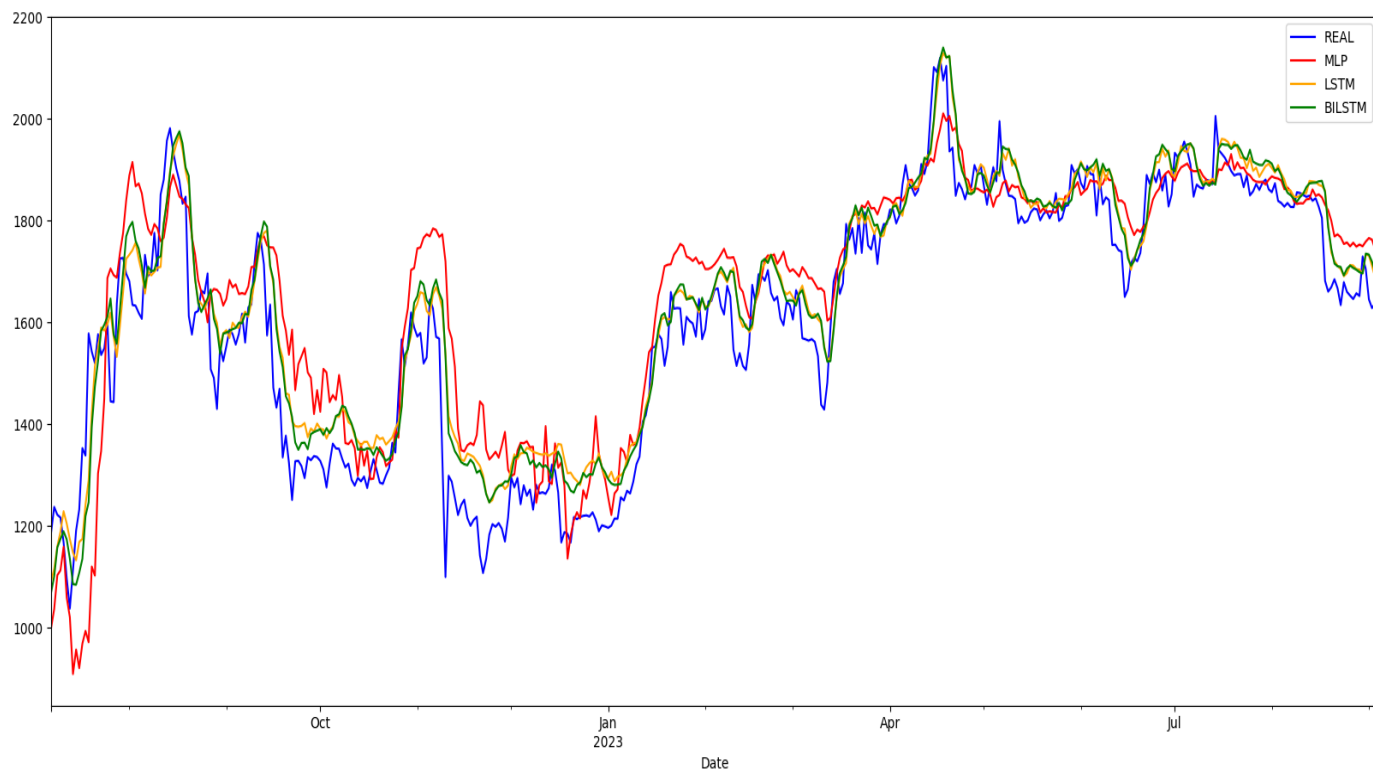
Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

5.3 Αποτελέσματα και συγκρίσεις

Ο κώδικας που χρησιμοποιήθηκε για να κατασκευαστούν τα διαγράμματα των προβλέψεων σε σχέση με τα πραγματικές τιμές καθώς και τα ίδια τα διαγράμματα παρουσιάζονται παρακάτω:

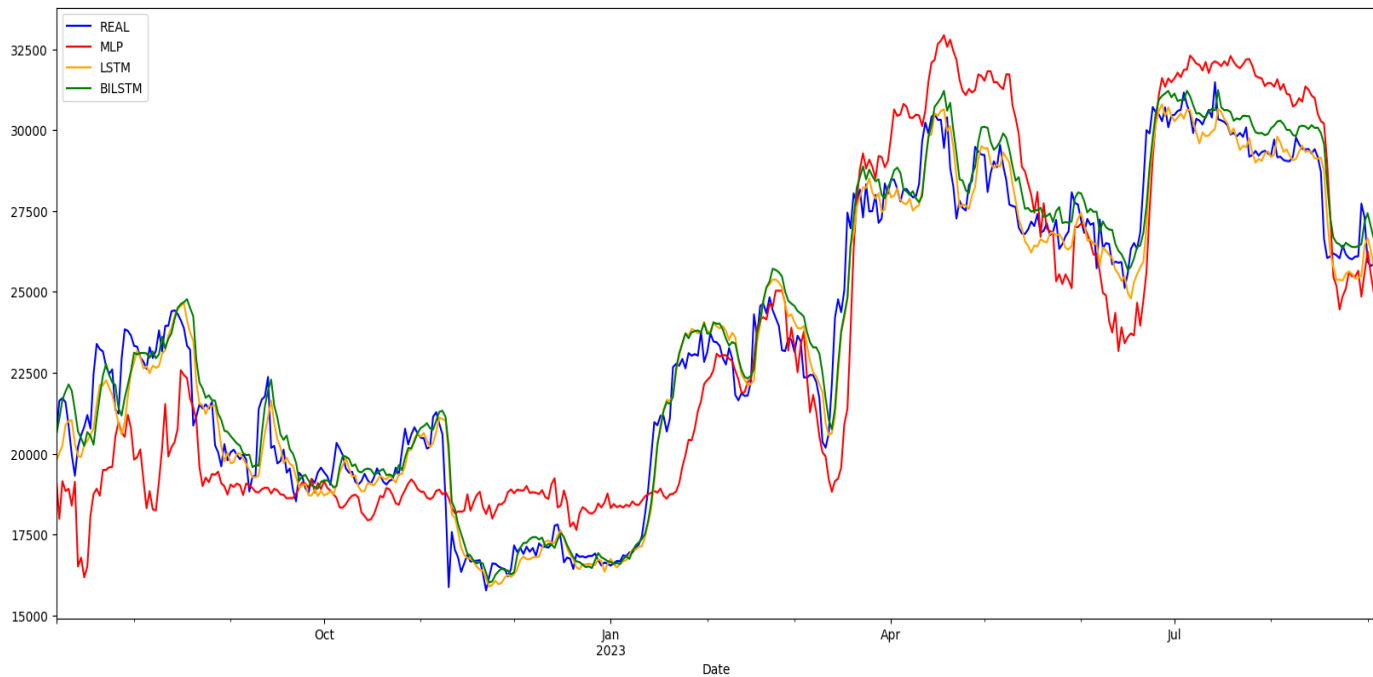
```
results['Open'].plot(figsize=(20,8), color='blue')
results['predictionsmlp'].plot(color='red')
results['predictionslstm'].plot(color='orange')
results['predictionsbilstm'].plot(color='green')
plt.legend(['REAL', 'MLP', 'LSTM', 'BILSTM'])
plt.show()
```

Εικόνα 5.12 Κώδικας Python για την οπτικοποίηση των προβλέψεων σε σχέση με τις πραγματικές τιμές των κρυπτονομισμάτων



Εικόνα 5.13 Σύγκριση των προβλέψεων με την πραγματική τιμή του Ethereum

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου



Εικόνα 5.14 Σύγκριση των προβλέψεων με την πραγματική τιμή του Bitcoin

Παρατηρώντας τόσο την πρόβλεψη στο Ethereum όσο και στο bitcoin είναι εύκολο κάνεις να καταλάβει ότι τα νευρωνικά είναι ένας σχετικά καλός τρόπος για να κάνει κάποιος πρόβλεψη χρονοσειρών καθώς σε όλες τις περιπτώσεις οι προβλέψεις είναι αρκετά καλές, βέβαια δεν είναι όλα τα νευρωνικά εξίσου καλά όπως περιμέναμε το καλύτερο και στις δυο περιπτώσεις ήταν το lstm με το bilstm να είναι λίγο πιο πίσω και το mlp να είναι αισθητά χειρότερο από τα δυο προηγούμενα.

Chapter 6 Συμπεράσματα & προτάσεις

Η πρόβλεψη των χρονοσειρών είναι ένα πολύ δύσκολο πρόβλημα και πιο συγκεκριμένα η πρόβλεψη των κρυπτονομισμάτων λόγω της φύσης τους random walk χρονοσειρές αλλά και των αποτόμων αλλαγών τους μετατρέπουν ένα είδη δύσκολο πρόβλημα σε ακόμα δυσκολότερο. Στα πλαίσια αυτής της διπλωματικής υλοποιήθηκαν βαθιά νευρωνικά δίκτυα με ιδιαίτερα εύκολο τρόπο χάρις το tensorflow τα οποία πάρα την δυσκολία του προβλήματος κατάφεραν να μας δώσουν μια αξιολογή πρόβλεψη για τις χρονοσειρές μας, παρόλο που δεν προσφέρουν αρκετά καλές προβλέψεις ώστε να μπορείς να έχεις κέρδος με αυτές μας παρέχουν μια ιδέα για την πιθανή τάση που θα ακολουθήσει η χρονοσειρά στο κοντινό μέλλον. Όπως ακριβώς περιμέναμε η αρχιτεκτονική lstm ήταν η καλύτερη στην πρόβλεψη πράγμα το οποίο είναι απολύτως λογικό καθώς η πρόβλεψη χρονοσειρών ταιριάζει τελειά στα χαρακτηριστικά του, το bilstm όχι μόνον χρειάζεται περισσότερο χρόνο και υπολογισμούς αλλά δεν είναι και εξίσου καλό με το lstm αλλά συνεχίζει να είναι με μεγάλη διαφορά καλύτερο από το μεν απλό αλλά όχι και τόσο καλό όσον αφορά την πρόβλεψη χρονοσειρών mlp.

Στα πλαίσια αυτή της διπλωματικής χρησιμοποιήθηκε μια πληθώρα τεχνικών για την αποσαφήνιση των χαρακτηριστικών των χρονοσειρών μας ώστε να επιλεγθούν με τον καταλληλότερο τρόπο οι υπερπαραμέτροι των νευρωνικών μας δικτύων, υλοποιήθηκαν τρία διαφορετικά βαθιά νευρωνικά δίκτυα και συγκρίθηκαν ως προς το πιο από αυτά είναι το καταλληλότερο για την πρόβλεψη κρυπτονομισμάτων.

Στις μελλοντικές κατευθύνσεις προτείνουμε:

- Η χρήση πιο εξελιγμένων μεθόδων για την χαρακτηριστικών των νευρωνικών μοντέλων τα οποία ονομάζονται autoencoders θα μπορούσε να μας οδηγήσει σε καλύτερη επιλογή χαρακτηριστικών και μείωση των διαστάσεων της πολυπλοκότητας, καθώς είναι ικανά να εντοπίζουν τα πιο σημαντικά χαρακτηριστικά πραγματοποιώντας μη γραμμικούς μετασχηματισμούς χωρίς να χάνουν πολύ πληροφορία.
- Η χρήση οικονομικών γεγονότων ως χαρακτηριστικά για την πρόβλεψη θα μπορούσε να μας παρέχει μιας εντελώς νέα οπτική γωνιά για την πρόβλεψη και εν τελεί να οδηγήσει σε ένα πολύ καλύτερο μοντέλο.
- Τεχνική ensembling δηλαδή ουσιαστικά συνδυασμός μοντέλων και τεχνικών πρόβλεψης για την δημιουργία ενός ακόμα καλύτερου μοντέλου.
- Σύγκριση των μοντέλων με διαφορετικές τεχνικές συγκρίσεις και σε διαφορετικά διαστήματα, δηλαδή κάποια μοντέλα, τεχνικές μπορεί να ταιριάζουν για πρόβλεψη στο κοντινό μέλλον καλύτερα και κάποια αλλά για μακροχρόνια πρόβλεψη

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

- Θα μπορούσε να χρησιμοποιηθεί ενισχυτική μάθηση για δεδομένα με χρονικό βήμα του ενός δευτερολέπτου για να δημιουργηθεί bot που να “αγοράζει” και να “πουλάει” κρυπτονομισματα αναλόγως της πρόβλεψης του στο αμέσως επόμενο χρονικό βήμα.

Βιβλιογραφία

- and, C. D.-I. J. of T. P., & 2017, undefined. (n.d.). A naïve approach for comparing a forecast model. *Researchgate.Net*, 5(3). Retrieved September 8, 2023, from https://www.researchgate.net/profile/Chuda-Dhawal/publication/326972994_A_Naive_Approach_for_Comparing_a_Forecast_Model/links/5d109f1ea6fdcc2462a03da6/A-Naive-Approach-for-Comparing-a-Forecast-Model.pdf
- Ariyo, A., Adewumi, A., 16th, C. A.-2014 Uks.-A., & 2014, undefined. (2014). Stock price prediction using the ARIMA model. *Ieeexplore.Ieee.Org*. <https://doi.org/10.1109/UKSim.2014.67>
- Bry, G., procedures, C. B.-C. analysis of time series: S., & 1971, undefined. (1971). Front matter to "Cyclical Analysis of Time Series: Selected Procedures and Computer Programs". *Nber.Org*. <https://www.nber.org/system/files/chapters/c2145/c2145.pdf>
- Catania, L., Forecasting, S. G.-I. J. of, & 2022, undefined. (n.d.). Forecasting cryptocurrency volatility. *Elsevier*. Retrieved September 8, 2023, from <https://www.sciencedirect.com/science/article/pii/S0169207021001059>
- Chowdhury, R., Rahman, M., ... M. R.-P. A. S., & 2020, undefined. (n.d.). An approach to predict and forecast the price of constituents and index of cryptocurrency using machine learning. *Elsevier*. Retrieved September 8, 2023, from <https://www.sciencedirect.com/science/article/pii/S0378437120302703>
- Dayhoff, J. (1990). *Neural network architectures: an introduction*. <https://dl.acm.org/doi/abs/10.5555/104012>
- de Livera, A. M., Hyndman, R. J., & Snyder, R. D. (2011). Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, 106(496), 1513–1527. <https://doi.org/10.1198/JASA.2011.TM09771>
- Engel, P. M., Pinto, R. C., Henrique, J., Flores, F., Martins Engel, P., & Pinto, R. C. (2012). Autocorrelation and partial autocorrelation functions to improve neural networks models on univariate time series forecasting. *Ieeexplore.Ieee.Org*. <https://doi.org/10.1109/IJCNN.2012.6252470>
- Huang, Z., Xu, W., & Yu, K. (2015). *Bidirectional LSTM-CRF Models for Sequence Tagging*. <http://arxiv.org/abs/1508.01991>
- Hylleberg, S., Jørgensen, C., & Sørensen, N. K. (1993). Seasonality in macroeconomic time series. *Empirical Economics*, 18(2), 321–335. <https://doi.org/10.1007/BF01205406>
- Ji, Z., Li, J., Information, M. T.-A. in N., & 2021, undefined. (n.d.). Early-stopped neural networks are consistent. *Proceedings.Neurips.Cc*. Retrieved September 8, 2023, from <https://proceedings.neurips.cc/paper/2021/hash/0e1ebad68af7f0ae4830b7ac92bc3c6f-Abstract.html>

- Journal, A. T.-F. C. and I., & 2018, undefined. (n.d.). Time series forecasting using artificial neural networks methodologies: A systematic review. *Elsevier*. Retrieved September 8, 2023, from <https://www.sciencedirect.com/science/article/pii/S2314728817300715>
- Luo, R., Tian, F., Qin, T., ... E. C.-A. in neural, & 2018, undefined. (n.d.). Neural architecture optimization. *Proceedings.Neurips.Cc*. Retrieved September 8, 2023, from <https://proceedings.neurips.cc/paper/2018/hash/933670f1ac8ba969f32989c312faba75-Abstract.html>
- Magazine, P. C.-I. S. P., & 2014, undefined. (2014). Tensors: a brief introduction. *leexplore.Ieee.Org*, 31(3), 44–53. <https://doi.org/10.1109/MSP.2014.2298533i>
- Manuca, R., Phenomena, R. S.-P. D. N., & 1996, undefined. (n.d.). Stationarity and nonstationarity in time series analysis. *Elsevier*. Retrieved September 8, 2023, from <https://www.sciencedirect.com/science/article/pii/S016727899600139X>
- Newbold, P. (1983). ARIMA model building and the time series analysis approach to forecasting. *Journal of Forecasting*, 2(1), 23–35. <https://doi.org/10.1002/FOR.3980020104>
- Nwogu, E. C., Celestine, K., Dozie, N., Ifeyinwa, M. H., Nwogu, E. C., Iwueze, I. S., Dozie, K. C. N., & Mbachu, H. I. (2019). Choice between mixed and multiplicative models in time series decomposition. *Researchgate.Net*, 2019(5), 153–159. <https://doi.org/10.5923/j.statistics.20190905.04>
- Phenomena, A. S.-P. D. N., & 2020, undefined. (n.d.). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Elsevier*. Retrieved September 8, 2023, from <https://www.sciencedirect.com/science/article/pii/S0167278919305974>
- reviews, M. M.-E., & 2019, undefined. (n.d.). Trend analysis of climate time series: A review of methods. *Elsevier*. Retrieved September 8, 2023, from <https://www.sciencedirect.com/science/article/pii/S0012825218303726>
- Roy, S., Nanjiba, S., International, A. C.-2018 21st, & 2018, undefined. (2018). Bitcoin price forecasting using time series analysis. *leexplore.Ieee.Org*, 21–23. <https://doi.org/10.1109/ICCITECHN.2018.8631923>
- Sciences, C. C.-I., & 1995, undefined. (n.d.). Time series segmentation: A sliding window approach. *Elsevier*. Retrieved September 8, 2023, from <https://www.sciencedirect.com/science/article/pii/S002002559500021G>
- Silaparasetty, N. (2020). The Tensorflow Machine Learning Library. *Machine Learning Concepts with Python and the Jupyter Notebook Environment*, 149–171. https://doi.org/10.1007/978-1-4842-5967-2_8
- Srivastava, N., Hinton, G., ... A. K.-T. journal of machine, & 2014, undefined. (n.d.). Dropout: a simple way to prevent neural networks from overfitting. *Jmlr.Org*. Retrieved September 8, 2023, from https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm_content=buffer79b43&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer,

Πρόβλεψη της τιμής των κρυπτονομισμάτων με τεχνικές βαθιάς μάθησης σε δεδομένα μεγάλου όγκου

- Staudemeyer, R. C., & Morris, E. R. (2019). *Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks*. <http://arxiv.org/abs/1909.09586>
- Tarasova, T., Usatenko, O., Makurin, A., Ivanenko, V., & Cherchata, A. (2020). Accounting and features of mathematical modeling of the system to forecast cryptocurrency exchange rate. *GrowingScience.Com*. <https://doi.org/10.5267/j.ac.2020.1.003>
- Taud, H., & Mas, J. F. (2018). *Multilayer Perceptron (MLP)*. 451–455. https://doi.org/10.1007/978-3-319-60801-3_27
- Toronto, N. S.-U. of, & 2013, undefined. (2013). Improving neural networks with dropout. *Cs.Toronto.Edu*. http://www.cs.toronto.edu/~nitish/msc_thesis.pdf
- Wongsuphasawat, K., Smilkov, D., Wexler, J., Wilson, J., Mané, M., Fritz, D., Krishnan, D., Ví, F. B., & Wattenberg, M. (2018). Visualizing dataflow graphs of deep learning models in tensorflow. *Ieeexplore.Ieee.Org*, 24(1). <https://doi.org/10.1109/TVCG.2017.2744878>
- Yu, Y., Si, X., Hu, C., computation, J. Z.-N., & 2019, undefined. (n.d.). A review of recurrent neural networks: LSTM cells and network architectures. *Direct.Mit.Edu*. Retrieved September 8, 2023, from <https://direct.mit.edu/neco/article-abstract/31/7/1235/8500>