



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Εφαρμογή Νευρωνικών Δικτύων στη Διοίκηση
Παραγωγής Βιομηχανικής Μονάδας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΣΠΥΡΟΥ ΦΩΤΟΠΟΥΛΟΥ

Επιβλέπων: Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π.

ΕΡΓΑΣΤΗΡΙΟ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ ΚΑΙ ΔΙΟΙΚΗΣΗΣ
Αθήνα, Οκτώβριος 2018



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Εργαστήριο Συστημάτων Αποφάσεων Και Διοίκησης

Εφαρμογή Νευρωνικών Δικτύων στη Διοίκηση Παραγωγής Βιομηχανικής Μονάδας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΣΠΥΡΟΥ ΦΩΤΟΠΟΥΛΟΥ

Επιβλέπων: Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 18η Οκτωβρίου 2019.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....
Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π.

.....
Ιωάννης Ψαρράς
Καθηγητής Ε.Μ.Π.

.....
Χρυσόστομος Δούκας
Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2018

(Υπογραφή)

.....

ΣΠΥΡΟΥ ΦΩΤΟΠΟΥΛΟΥ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

© 2018 – All rights reserved



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥ-
ΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Εργαστήριο Συστημάτων Αποφάσεων Και Διοίκησης

Copyright ©–All rights reserved Σπύρου Φωτόπουλου, Με επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον καθηγητή κ. Ασκούνη για την επίβλεψη της διπλωματικής μου εργασίας και για την ευκαιρία που μου έδωσε να την εκπονήσω στο εργαστήριο Συστημάτων Αποφάσεων και Διοίκησης. Επίσης ευχαριστώ ιδιαίτερα τον υποψήφιο Δρ. Μιχάλη Αυγουλή για την καθοδήγησή τους και την εξαιρετική συνεργασία που είχαμε. Τέλος θα ήθελα να ευχαριστήσω του φίλους μου Θοδωρή και Κωνσταντίνο και τους υπόλοιπους συμφοιτητές και φίλους μου για την καθοδήγηση και την ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια.

Περίληψη

Η παρούσα διπλωματική εργασία διεξήχθη σε συνεργασία με μια βιομηχανική μονάδα και αφορά την σχεδίαση μίας 'έξυπνης' εφαρμογής ικανή να προβλέπει δείκτες που αφορούν τον τομέα της παραγωγής ενός εργοστασίου. Το εργαλείο που χρησιμοποιήθηκε για την υλοποίηση της πρόβλεψης κατασκευάστηκε χρησιμοποιώντας ένα μοντέλο Νευρωνικού Δικτύου. Συγκεκριμένα το είδος του νευρωνικού δικτύου που χρησιμοποιήθηκε είναι ένα Επαναλαμβανόμενο Νευρωνικό Δίκτυο (RNN – Recurrent Neural Network).

Λέξεις Κλειδιά

Έξυπνη εφαρμογή, Πρόβλεψη, Επαναλαμβανόμενο Νευρωνικό Δίκτυο

Abstract

This diploma thesis is being conducted in collaboration with an industry and involves the design of a "smart" application, which is capable of making predictions, that are useful as far as the productive sector of a plant is concerned. The tool used to implement the forecast was constructed using a Neural Network model. In particular, the type of neuronal network used is a Recurrent Neural Network (RNN).

Keywords

Smart Application, Forecast, Recurrent Neural Network

Περιεχόμενα

Ευχαριστίες	1
Περίληψη	3
Abstract	5
Περιεχόμενα	8
Κατάλογος Πινάκων	11
1 Εισαγωγή	13
1.1 Βιομηχανική Επανάσταση 4.0	13
1.2 Στόχος της Εργασίας	14
1.3 Γραμμή Παραγωγής	15
1.4 Σημασία του A.I (Artificial Intelligence)	15
1.5 Μεγάλα Δεδομένα (Big Data)	16
1.6 Ανάλυση Δεδομένων (Data Analytics)	16
1.7 OEE (Overall Equipment Effectivness)	17
1.7.1 Έξι Βασικές Απώλειες	17
1.8 Κρίσιμα Δείκτες Απόδοσης(KPI - Key Performance Indicators)	19
1.9 Συνεισφορά της Εργασίας	19
2 Μέθοδοι Αναλύσεων και Προβλέψεων στην Βιομηχανία	21
2.1 Ανάλυση Δεδομένων στην Βιομηχανία	21
2.2 Πρόβλεψη	21
2.2.1 Κλασσικές Τεχνικές Προβλέψεων	22
2.3 Μηχανική και Βαθιά Μάθηση (Machine and Deep Learning)	22
2.3.1 Γιατί τεχνητή νοημοσύνη τώρα ;	22
2.3.2 Machine Learning	22
2.3.3 Deep Learning	25
3 Αλγόριθμοι Machine και Deep Learning	29
3.1 Εργαλειοθήκη	29

3.1.1	Python	29
3.1.2	Βιβλιοθήκες	30
3.2	Αλγόριθμοι Machine Learning	31
3.2.1	Linear Regression	32
3.2.2	Decision Tree	36
3.2.3	Random Forest	38
3.2.4	Gradient Boosting Tree	40
3.3	Αλγόριθμοι Deep Learning	41
3.3.1	Artificial Neural Networks	41
3.3.2	Βήμα Βήμα η Δημιουργία του Νευρωνικού	52
3.3.3	Διαγράμματα Ροής	60
3.4	Εκτέλεση του RNN	62
3.4.1	Αποτελέσματα	62
3.4.2	Σχολιασμός των αποτελεσμάτων των Πειραμάτων Α, Β και Γ	67
3.4.3	Αποτελέσματα Τελικής Εφαρμογής για τον δείκτη OEE	68
3.4.4	Αποτέλεσμα Τελικού Λογισμικού για τον αριθμό των σφαλμάτων	74
3.4.5	Σχολιασμός των Αποτελεσμάτων της τελικής εφαρμογής	77
3.4.6	Επεξεργασία του Νευρωνικού	78
4	Αποτελέσματα και Σχολιασμός	79
4.1	Πρόβλεψη με κλασικούς αλγόριθμους Machine Learning	79
4.1.1	Linear Regression	79
4.1.2	Decision Tree	80
4.1.3	Random Forest	81
4.1.4	Gradient Boosting Tree	82
4.2	Πρόβλεψη RNN	83
4.3	Σχολιασμός των αποτελεσμάτων	88
5	Συμπεράσματα και Μελλοντική Εργασία	95
5.1	Τελικές Ρυθμίσεις του RNN	95
5.2	Συμπεράσματα	95
5.3	Μελλοντική Εργασία	97
5.3.1	IBM Watson Studio	97
5.3.2	Αξιοποίηση του λογισμικού σε σύστημα Performance Managment και αξιοποίηση του Cloud	97
5.3.3	Επίλογος	98

Κατάλογος Πινάκων

1.1 Υπολογισμός A, P, Q	19
3.1 Data Set	34
3.2 Shift A, Linear Regression, OEE	35
3.3 Shift B, Linear Regression, OEE	36
3.4 Shift C, Linear Regression, OEE	36
3.5 Shift A, Decision Tree, OEE	37
3.6 Shift B, Decision Tree, OEE	38
3.7 Shift C, Decision Tree, OEE	38
3.8 Shift A, Random Forest, OEE	38
3.9 Shift B, Random Forest, OEE	39
3.10 Shift C, Random Forest, OEE	39
3.11 Shift A, Gradient Boosting Tree, OEE	40
3.12 Shift B, Gradient Boosting Tree, OEE	41
3.13 Shift C, Gradient Boosting Tree, OEE	41
3.14 Data Set	53
3.15 Αποδεκτή μορφή Data set για το RNN	53
3.16 Δεδομένα για Ιανουάριο 2018	54
3.17 Dataset	55
3.18 Τελική Αποδεκτή μορφή Data set	56
3.19 OEE, Χωρίς Προϊόν και χωρίς Βάρδιες	62
3.20 <i>ShiftA</i>	64
3.21 <i>ShiftB</i>	64
3.22 <i>ShiftC</i>	64
3.23 <i>Product Code = 1</i>	65
3.24 <i>Product Code = 2</i>	65
3.25 <i>Product Code = 3</i>	65
3.26 <i>Product Code = 4</i>	66
3.27 <i>Product Code = 5</i>	66
3.28 <i>Product Code = 13</i>	66
3.29 <i>Shift A ProductCode = 1</i>	68
3.30 <i>Shift A ProductCode = 2</i>	68

3.31	<i>Shift A ProductCode = 3</i>	68
3.32	<i>Shift A ProductCode = 4</i>	69
3.33	<i>Shift A ProductCode = 5</i>	69
3.34	<i>Shift A ProductCode = 13</i>	69
3.35	<i>Shift B ProductCode = 1</i>	70
3.36	<i>Shift B ProductCode = 2</i>	70
3.37	<i>Shift B ProductCode = 3</i>	70
3.38	<i>Shift B ProductCode = 4</i>	71
3.39	<i>Shift B ProductCode = 5</i>	71
3.40	<i>Shift B ProductCode = 13</i>	71
3.41	<i>Shift C ProductCode = 1</i>	72
3.42	<i>Shift C ProductCode = 2</i>	72
3.43	<i>Shift C ProductCode = 3</i>	72
3.44	<i>Shift C ProductCode = 4</i>	73
3.45	<i>Shift C ProductCode = 5</i>	73
3.46	<i>Shift C ProductCode = 13</i>	73
3.47	<i>Shift A ProductCode = 1</i>	74
3.48	<i>Shift A ProductCode = 2</i>	74
3.49	<i>Shift A ProductCode = 3</i>	74
3.50	<i>Shift A ProductCode = 4</i>	74
3.51	<i>Shift A ProductCode = 5</i>	74
3.52	<i>Shift A ProductCode = 13</i>	75
3.53	<i>Shift B ProductCode = 1</i>	75
3.54	<i>Shift B ProductCode = 2</i>	75
3.55	<i>Shift B ProductCode = 3</i>	75
3.56	<i>Shift B ProductCode = 4</i>	75
3.57	<i>Shift B ProductCode = 5</i>	75
3.58	<i>Shift B ProductCode = 13</i>	75
3.59	<i>Shift C ProductCode = 1</i>	76
3.60	<i>Shift C ProductCode = 2</i>	76
3.61	<i>Shift C ProductCode = 3</i>	76
3.62	<i>Shift C ProductCode = 4</i>	76
3.63	<i>Shift C ProductCode = 5</i>	76
3.64	<i>Shift C ProductCode = 13</i>	76
3.65	<i>OEE % Shift A (Prediction/Real)</i>	78
3.66	<i>OEE % Shift B (Prediction/Real)</i>	78
3.67	<i>OEE % Shift C (Prediction/Real)</i>	78
4.1	Shift A, Linear Regression, OEE	79
4.2	Shift B, Linear Regression, OEE	79
4.3	Shift C, Linear Regression, OEE	80

4.4	Shift A, Decision Tree, OEE	80
4.5	Shift B, Decision Tree, OEE	80
4.6	Shift C, Decision Tree, OEE	80
4.7	Shift A, Random Forest, OEE	81
4.8	Shift B, Random Forest, OEE	81
4.9	Shift C, Random Forest, OEE	81
4.10	Shift A, Gradient Boosting Tree, OEE	82
4.11	Shift B, Gradient Boosting Tree, OEE	82
4.12	Shift C, Gradient Boosting Tree, OEE	82
4.13	<i>Shift A ProductCode = 1</i>	83
4.14	<i>Shift A ProductCode = 2</i>	83
4.15	<i>Shift A ProductCode = 3</i>	83
4.16	<i>Shift A ProductCode = 4</i>	84
4.17	<i>Shift A ProductCode = 5</i>	84
4.18	<i>Shift A ProductCode = 13</i>	84
4.19	<i>Shift B ProductCode = 1</i>	85
4.20	<i>Shift B ProductCode = 2</i>	85
4.21	<i>Shift B ProductCode = 3</i>	85
4.22	<i>Shift B ProductCode = 4</i>	86
4.23	<i>Shift B ProductCode = 5</i>	86
4.24	<i>Shift B ProductCode = 13</i>	86
4.25	<i>Shift C ProductCode = 1</i>	87
4.26	<i>Shift C ProductCode = 2</i>	87
4.27	<i>Shift C ProductCode = 3</i>	87
4.28	<i>Shift C ProductCode = 4</i>	88
4.29	<i>Shift C ProductCode = 5</i>	88
4.30	<i>Shift C ProductCode = 13</i>	88
4.31	Σφάλματα (%) για Epoch = 10 και Shift = A	90
4.32	Σφάλματα (%) για Epoch = 10 και Shift = B	90
4.33	Σφάλματα (%) για Epoch = 10 και Shift = C	90
4.34	Σφάλματα (%) για Epoch = 50 και Shift = A	91
4.35	Σφάλματα (%) για Epoch = 50 και Shift = B	91
4.36	Σφάλματα (%) για Epoch = 50 και Shift = C	91
4.37	Σφάλματα (%) για Epoch = 100 και Shift = A	92
4.38	Σφάλματα (%) για Epoch = 100 και Shift = B	92
4.39	Σφάλματα (%) για Epoch = 100 και Shift = C	92

Κεφάλαιο 1

Εισαγωγή

1.1 Βιομηχανική Επανάσταση 4.0

Πρώτα ήρθε ο ατμός και οι πρώτες μηχανές που βοήθησαν τους προγόνους μας σε κάποιες από τις εργασίες τους. Μετά ήρθε ο ηλεκτρισμός, οι γραμμές συναρμολόγησης και η γέννηση της μαζικής παραγωγής. Η τρίτη βιομηχανική εποχή ήρθε μαζί με την έλευση των υπολογιστών και την αρχή της αυτοματοποίησης, όταν τα ρομπότ και οι μηχανές άρχισαν να αντικαταστούν τους εργάτες στις γραμμές παραγωγής.

Και τώρα μπαίνουμε στη Βιομηχανία 4ης γενιάς ή όπως είναι στα Γερμανικά ο όρος που αυτοί πρώτοι εισήγαγαν INDUSTRIE 4.0. Στη Βιομηχανία 4ης γενιάς οι υπολογιστές και η αυτοματοποίηση ενώνονται με ένα τελείως νέο τρόπο, με τα ρομπότ να συνδέονται εξ αποστάσεως με υπολογιστικά συστήματα εξοπλισμένα με αλγόριθμους μηχανικής εκμάθησης (machine learning) που μπορούν να μαθαίνουν και να ελέγχουν τα ρομπότ με ελάχιστες εισροές από τους χειριστές.

Μία από τις πρώτες φορές που αναφέρθηκε ο όρος Βιομηχανία 4.0 , ήταν σε ένα ;μέμο; της Γερμανικής κυβέρνησης, που εκδόθηκε το 2013. Η ίδια η καγκελάρια της Γερμανίας, Ανγκέλα Μέρκελ, σε μια άλλη περίπτωση (January 2015 at the World Economic Forum in Davos) αναφέρθηκε με ιδιαίτερη έμφαση στην Βιομηχανία 4.0 σαν τον «τρόπο που θα μπορέσουμε να επιτύχουμε την ;σύντηξη; του διαδικτυακού κόσμου και του κόσμου της βιομηχανικής παραγωγής». Η Βιομηχανία 4ης γενιάς μας εισάγει σε αυτό που αποκαλούσαμε έως τώρα ;έξυπνο εργοστάσιο; στο οποίο κυβερνο-φυσικά (Cyber-Physical / Digital Twins) συστήματα παρακολουθούν τις φυσικές διαδικασίες του εργοστασίου και παίρνουν αποκεντρωμένες αποφάσεις. Τα φυσικά συστήματα γίνονται έτσι το Διαδίκτυο των Πραγμάτων (IoT), επικοινωνώντας και συνεργαζόμενα μεταξύ τους, καθώς και με τους ανθρώπους χειριστές, σε πραγματικό χρόνο δια μέσω του ασύρματου δικτύου.

Για να θεωρηθεί ένα εργοστάσιο ή σύστημα ότι ανήκει στην Βιομηχανία 4ης γενιάς, θα πρέπει να συμπεριλαμβάνει:

1. Διαλειτουργικότητα (Interoperability)- Μηχανές, συσκευές, αισθητήρες και άνθρωποι που συνδέονται και θα επικοινωνούν μεταξύ τους απρόσκοπτα.
2. Διαφάνεια της πληροφορίας- οι μηχανές δημιουργούν ένα εικονικό αντίγραφο του πραγματικού κόσμου μέσω δεδομένων από διάφορους αισθητήρες(sensors) με στόχο να εντάξουν τις χαοτικές πληροφορίες σε ένα στενότερο πλαίσιο που θα είναι επεξεργάσιμο και προσπελάσιμο από μηχανές (Machine Recognizable Big Data, etc).
3. Τεχνική υποστήριξη- που περιλαμβάνει μαζί την ικανότητα των συστημάτων να υποστηρίξουν τους ανθρώπους στην λήψη αποφάσεων, και την ικανότητα να υποβοηθούν τους ανθρώπους με εργασίες που είναι πολύ δύσκολες ή πολύ επικίνδυνες για αυτούς.
4. Αποκεντρωμένη λήψη αποφάσεων- η ικανότητα των cyber-physical συστημάτων να παίρνουν απλές αποφάσεις από μόνα τους και να γίνονται όσο πιο αυτόνομα γίνεται.

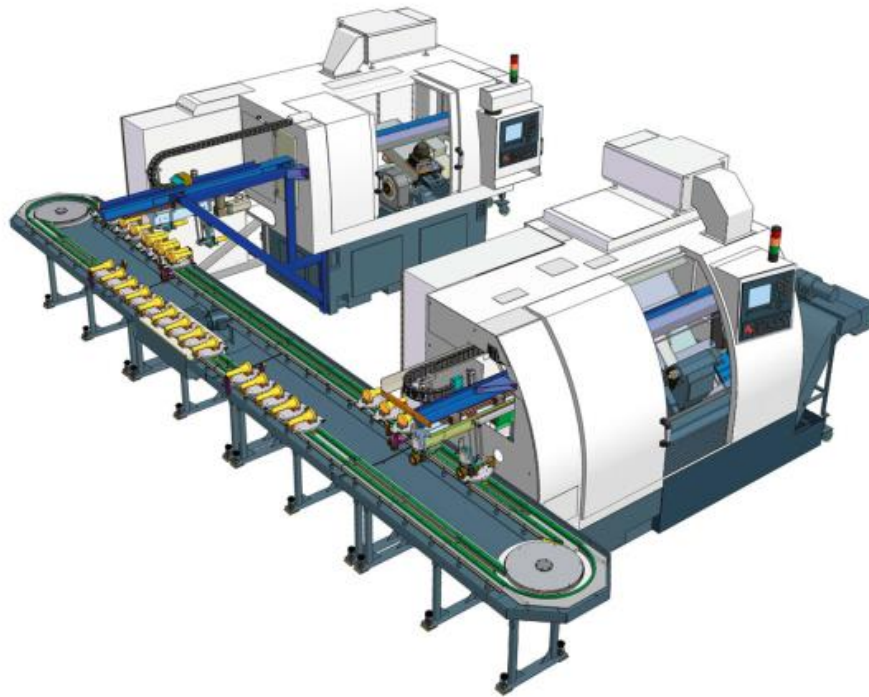
Τα παραπάνω όμως δε σημαίνουν ότι είμαστε ήδη εκεί, υπάρχουν πολλά προβλήματα που πρέπει να επιλυθούν όπως είναι η ασφάλεια των δεδομένων, η ανάγκη για υψηλή αξιοπιστία και σταθερότητα αυτών των cyber-physical συστημάτων, η εξασφάλιση της ακεραιότητας της παραγωγής χωρίς την ανθρώπινη επίβλεψη, η απώλεια εργασιών με υψηλές αποδοχές κ.α. Επιπροσθέτως υπάρχει μεγάλη έλλειψη πεπειραμένου προσωπικού για να εφαρμόσει τέτοια εξειδικευμένα συστήματα παραγωγής, για να μην αναφέρουμε την απροθυμία των ενασχολούμενων με τον τομέα να επενδύσουν σε τέτοιες τεχνολογίες απλόχερα. Το ερώτημα λοιπόν δεν είναι αν θα έρθει η Βιομηχανία 4ης γενιάς αλλά πότε θα έρθει, όπως και με τα μεγάλα δεδομένα οι ειδικοί πιστεύουν ότι όσοι υιοθετήσουν αυτές τις πρακτικές πρώτοι, θα ανταμειφθούν. Το νέο πρόγραμμα εργασίας (WP18-20) για τις Τεχνολογίες Πληροφορικής και Τηλεπικοινωνιών του Ορίζοντα 2020 θα δίνει μεγάλη έμφαση στις τεχνολογίες που θα φέρουν την βιομηχανία 4ης γενιάς ακόμα πιο κοντά μας, στα cyber-physical systems, στην αξιοποίηση των ψηφιακών κόμβων καινοτομίας (Digital Innovation Hubs) ειδικά στην περιοχή των smart factories.

1.2 Στόχος της Εργασίας

Η παρούσα διπλωματική εργασία διεξήχθη για μια πολυεθνική βιομηχανία στην Νοτιοανατολική Ευρώπη. Λόγω εμπιστευτικότητας δεν θα αναφερθεί το όνομά της, για αυτόν τον λόγο θα αναφέρεται ως μια εταιρία καλλυντικών. Στην παρούσα μελέτη, ο στόχος μας ήταν να δημιουργήσουμε μια αξιόπιστη μεθοδολογία για μελλοντική εκτίμηση συγκεκριμένου KPI που ονομάζεται Ολοκληρωμένη Αποτελεσματικότητα Εξοπλισμού (OEE) για μια συγκεκριμένη γραμμή παραγωγής και τύπο προϊόντων, στο εργοστάσιο μιας εταιρίας καλλυντικών. Αυτή η εταιρεία χρησιμοποιεί το δείκτη OEE ως δείκτη απόδοσης των διαδικασιών παραγωγής της. Ο OEE είναι ένας δείκτης που μετρά πόσο αποτελεσματικά μια επιχείρηση αξιοποιεί το παραγωγικό δυναμικό της καταγράφοντας τις απώλειες κατά τη διάρκεια της παραγωγικής διαδικασίας και χρησιμοποιείται συχνά ως δείκτης μέτρησης και βελτίωσης της απόδοσης της παραγωγής σε TPM (Total Productive Maintenance)

1.3 Γραμμή Παραγωγής

Μια γραμμή παραγωγής είναι ένα σύνολο διαδοχικών λειτουργιών εγκατεστημένων σε ένα εργοστάσιο, όπου τα υλικά τίθενται μέσω διεργασίας εξευγενισμού για την παραγωγή ενός τελικού προϊόντος κατάλληλο για περαιτέρω κατανάλωση. ή εξαρτήματα συναρμολογούνται για να φτιάξουν ένα τελικό αντικείμενο. Στην εν λόγω διπλωματική εργασία ασχολούμαστε με μια από τις γραμμές παραγωγής ενός εργοστασίου καλλυντικών. Στο συγκεκριμένο εργοστάσιο υπάρχουν δύο ειδών γραμμές παραγωγής. Το πρώτο είδος γραμμών εκτελεί την διαδικασία αναμίξεων, όπου συντίθεται το μίγμα για τα καλλυντικά και το δεύτερο είδος εκτελεί την διαδικασία της γέμισης του ήδη επεξεργασμένου μίγματος στις κατάλληλες συσκευασίες.

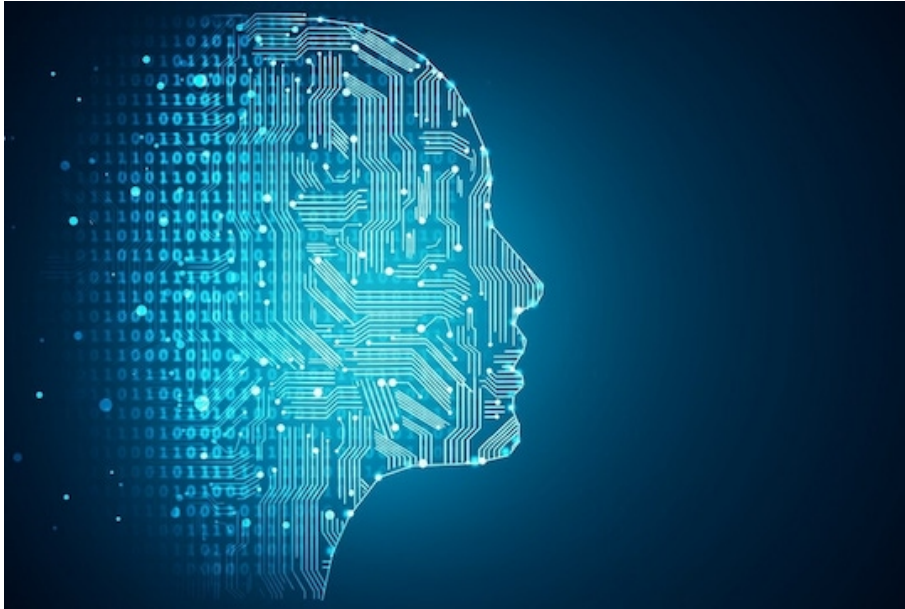


Σχήμα 1.1: Γραμμή Παραγωγής

1.4 Σημασία του A.I (Artificial Intelligence)

Η τεχνητή νοημοσύνη καθιστά τις μηχανές ικανές να μαθαίνουν από την εμπειρία, να προσαρμόζονται σε νέα εισαγόμενα δεδομένα και να εκτελούν ανθρωπομορφικά έργα. Τα περισσότερα παραδείγματα AI τα οποία γνωρίζουμε σήμερα, βασίζονται σε μεγάλο βαθμό στο deep learning και την επεξεργασία φυσικής γλώσσας (ΕΦΓ). Με τη χρήση των τεχνολογιών αυτών, οι υπολογιστές μπορούν να εκπαιδευτούν ώστε να επιτελούν συγκεκριμένα καθήκοντα με επεξεργασία μεγάλων ποσοτήτων δεδομένων και αναγνώριση μορφών στα δεδομένα. Η παρούσα διπλωματική εργασία ‘γεννήθηκε’ μετά από έρευνα των αναγκών του εργοστασίου της εταιρείας καλλυντικών. Το αρχικό ερώτημα που τέθηκε ήταν, αν θα μπορούσαμε να βελτιώσουμε τον τομέα της παραγωγής του εργοστασίου αξιοποιώντας τα μοντέλα της μηχανικής

και βαθιάς μάθησης. Μετά από μελέτη και εκπόνηση πειραμάτων παρατηρήθηκε ότι ένας από του πιο κατάλληλους τρόπους προκειμένου να βελτιώσουμε την παραγωγή θα ήταν προβλέποντας την απόδοση της μηχανής παραγωγής του εργοστασίου. Με αυτόν τον τρόπο μπορεί κανείς να εξάγει πολλά σημαντικά συμπεράσματα για την συμπεριφορά της εκάστοτε μηχανής παραγωγής και να προλάβει πιθανά σφάλματα της.



Σχήμα 1.2: Τεχνητή Νοημοσύνη

1.5 Μεγάλα Δεδομένα (Big Data)

Τα μεγάλα δεδομένα είναι ένας όρος που περιγράφει τον μεγάλο όγκο δεδομένων – τόσο δομημένων όσο και μη δομημένων – που κατακλύζουν μια επιχείρηση καθημερινά. Αλλά δεν είναι το πλήθος των δεδομένων που είναι σημαντικό, είναι η αξιοποίηση τους. Μεγάλα δεδομένα μπορούν να αναλυθούν για ιδέες που οδηγούν σε καλύτερες αποφάσεις και στρατηγικές επιχειρηματικές κινήσεις.

1.6 Ανάλυση Δεδομένων (Data Analytics)

Η ανάλυση δεδομένων (DA) είναι η διαδικασία εξέτασης συνόλων δεδομένων με τη βοήθεια εξειδικευμένων συστημάτων και λογισμικού προκειμένου να εξαχθούν συμπεράσματα σχετικά με τις πληροφορίες που περιέχουν. Οι τεχνολογίες και οι τεχνικές ανάλυσης δεδομένων χρησιμοποιούνται ευρέως τόσο στις εμπορικές βιομηχανίες για να επιτρέψουν στις οργανώσεις να λαμβάνουν πιο ενημερωμένες επιχειρηματικές αποφάσεις όσο από επιστήμονες και ερευνητές για να επαληθεύσουν ή να διαψεύσουν επιστημονικά μοντέλα, θεωρίες και υποθέσεις.

1.7 OEE (Overall Equipment Effectivness)

Ο OEE (Συνολική Αποτελεσματικότητα Εξοπλισμού) είναι το πρότυπο για τη μέτρηση της απόδοσης της παραγωγικής διαδικασίας. Με απλά λόγια, προσδιορίζει το ποσοστό του χρόνου παραγωγής που είναι πραγματικά παραγωγικό. Μια βαθμολογία OEE 100% σημαίνει ότι τα 'καλά' κομμάτια (Good Parts) κατασκευάζονται, όσο το δυνατόν γρηγορότερα, χωρίς χρόνο διακοπής.

$$OEE = Performance \cdot Availability \cdot Quality \quad (1.1)$$

1.7.1 Έξι Βασικές Απώλειες

Οι παρακάτω έξι μεγαλύτερες πηγές ζημιών είναι οι κύριοι παράγοντες που επηρεάζουν την παραγωγική διαδικασία :

1. Σφάλματα Εξοπλισμού

Αυτές οι απώλειες είναι ο χρόνος που χάνεται όταν η παραγωγική διαδικασία σταματά καθώς ο εξοπλισμός χάνει ξαφνικά την ικανότητα να εκτελεί τις εκχωρημένες λειτουργίες του λόγω λειτουργίας (π.χ. ορισμένα εξαρτήματα σταματούν να λειτουργούν) ή μειωμένη λειτουργία (δηλ. Φυσικές απώλειες όπως ελαττώματα κλπ)

2. Σφάλματα Διαδικασίας

Αυτές οι απώλειες είναι ο χρόνος που χάνεται όταν η μονάδα σταματά να λειτουργεί ως αποτέλεσμα παραγόντων που δεν σχετίζονται με τον εξοπλισμό, όπως χειρισμού σφαλμάτων ή αλλαγών στις φυσικές ή χημικές ιδιότητες των προς επεξεργασία αντικειμένων

3. Αναμενόμενες Απώλειες Παραγωγής

Αυτές οι απώλειες οφείλονται σε μείωση του ποσοστού παραγωγής στις νεοσύστατες επιχειρήσεις, στα διαλείμματα και στις αλλαγές στη μονάδα. Η μείωση της παραγωγής που παρατηρείται σε αυτές τις περιπτώσεις πρέπει να αντιμετωπίζεται ως απώλεια

4. Μη Αναμενόμενες Απώλειες Παραγωγής

Αυτές οι απώλειες οφείλονται στη μείωση της παραγωγής λόγω μη ομαλής λειτουργίας της μονάδας

5. Αδράνεια και Μικρές Διακοπές

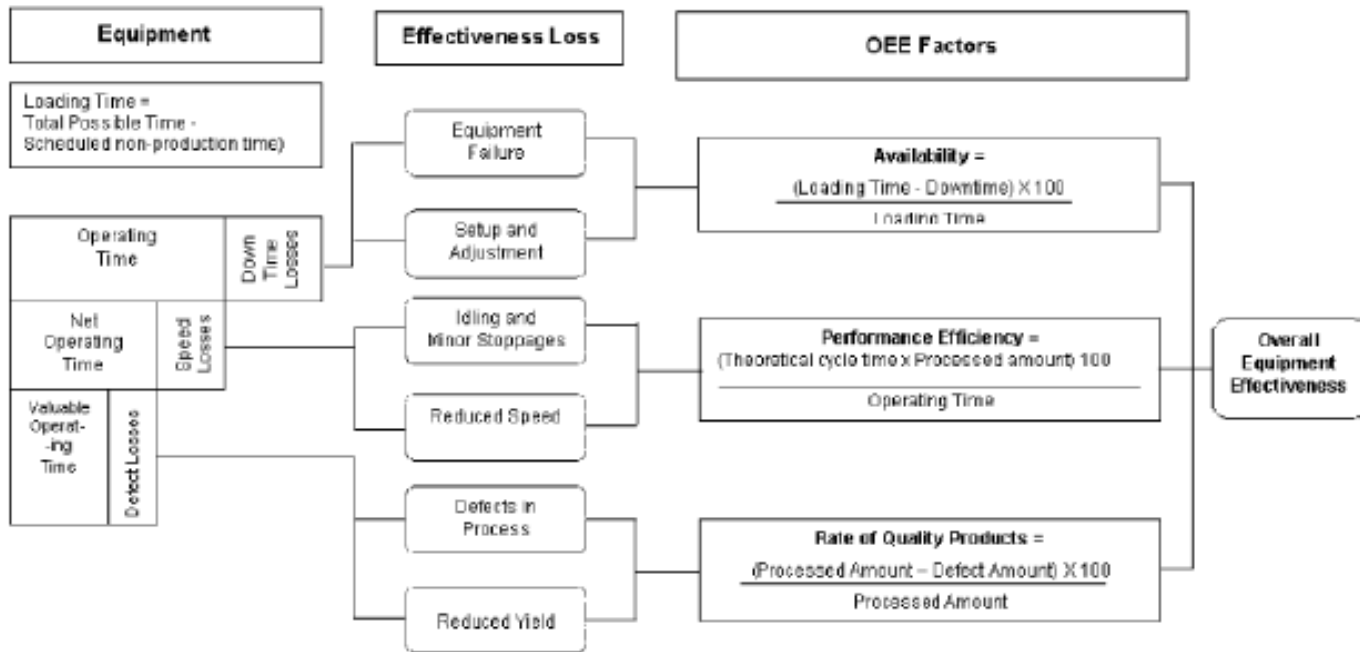
Εμφανίζονται όταν υπάρχει προσωρινή δυσλειτουργία της μηχανής, π.χ. τις απώλειες που μπορεί να προκύψουν κατά την αφαίρεση ελαττωματικών προϊόντων κ.λπ. ή όταν το μηχάνημα βρίσκεται σε κατάσταση αναμονής για την επόμενη εργασία

6. Απώλειες Ποιότητας

Αυτές οι απώλειες περιλαμβάνουν τον χρόνο που καταναλώνεται για την παραγωγή απορριφθέντων προϊόντων, τις φυσικές απώλειες που απορρίπτονται στο υλικό που απορρίπτεται και τις οικονομικές απώλειες λόγω υποβάθμισης του προϊόντος

Τέλος, πρέπει επίσης να ληφθεί υπόψη η διακοπή της παραγωγής. Αυτές οι απώλειες είναι ο χρόνος που χάνεται όταν η παραγωγή σταματά για την προγραμματισμένη ετήσια συντήρηση

ή την περιοδική υπηρεσία. Με βάση τα έξι μοντέλα των πηγών απώλειας που παρουσιάζονται παραπάνω, είναι δυνατόν να κατανοήσουμε τον τρόπο μείωσης του χρόνου φόρτωσης σε αξιόλογο χρόνο λειτουργίας και τον επηρεασμό της αποτελεσματικότητας.



Σχήμα 1.3: Απεικόνιση του τρόπου με τον οποίο μειώνεται ο παραγωγικός χρόνος με βάση τα έξι μοντέλα Απωλειών

Συνοπώς, $OEE = 100\%$ αντιστοιχεί σε, 100% Ποιότητα/Quality (μόνο Καλό Μέρος/Good Parts), 100% Απόδοση/Performane (όσο το δυνατόν γρηγορότερα) και 100% Διαθεσιμότητα/Availability (χωρίς Χρόνο Διακοπής). Μέσω της μέτρησης του OEE και των υποκείμενων ζημιών, αποκτάμε σημαντικές πληροφορίες σχετικά με τον τρόπο συστηματικής βελτίωσης της παραγωγικής διαδικασίας. Ο OEE είναι η καλύτερη μετρική για τον εντοπισμό των απωλειών, τη συγκριτική αξιολόγηση της προόδου και τη βελτίωση της παραγωγικότητας του κατασκευαστικού εξοπλισμού.

Στον παρακάτω πίνακα παρουσιάζονται δύο διαφορετικοί τρόποι υπολογισμού των επιμέρων συνιστωσών του OEE:

Πίνακας 1.1: Υπολογισμός A, P, Q

Factors	Nakajima, 1988	De Groote, 1995
Availability (A)	$\frac{\text{Operating time}}{\text{Loading time}}$	$\frac{\text{Production Time} - \text{Unplanned Down Time}}{\text{planned production time}}$
Performance (P)	$\frac{\text{Net operating time}}{\text{Loading time}}$	$\frac{\text{Actual amount of production}}{\text{planned production}}$
Quality (Q)	$\frac{\text{Valuable operating time}}{\text{Net operating time}}$	$\frac{\text{Actual amount of production} - \text{non-accepted amount}}{\text{Actual amount}}$

1.8 Καίριοι Δείκτες Απόδοσης(KPI - Key Performance Indicators)

Οι Καίριοι Δείκτες Απόδοσης (εφεξής Δείκτες ή KPIs) είναι εργαλεία επιμέτρησης επιδόσεων συγκεκριμένων προσώπων και δραστηριοτήτων στα πλαίσια μιας εταιρείας ή ενός οργανισμού, ή ακόμα και του οργανισμού ως σύνολο αναφορικά με την επίτευξη ποσοτικών, στρατηγικών και επιχειρησιακών στόχων που έχουν τεθεί. Χρησιμοποιούνται κατά κύριο λόγο από τα μεσαία και ανώτερα στελέχη, ώστε να αξιολογηθεί ο βαθμός επιτυχίας των επιμέρους τμημάτων. Μέσα από αυτή την αξιολόγηση επιτυγχάνεται επίσης η ταχύτερη, βαθύτερη και πληρέστερη κατανόηση των βασικών επιχειρησιακών λειτουργιών, καθώς και η διεύρυνση του ελέγχου που ασκείται από τα στελέχη. Συχνά αναφέρονται και ως 'Καίριοι Δείκτες Επιτυχίας' (Key Success Indicators, KSIs). Κατά κύριο λόγο η επιχειρηματική επιτυχία ερμηνεύεται ως η πρόοδος μιας εταιρείας βάσει προκαθορισμένων στρατηγικών οικονομικών στόχων. Όμως, συχνά η επιτυχία έχει και την έννοια του βαθμού επίτευξης, σε περιοδική βάση, λοιπών στόχων που είναι παράλληλοι με τους οικονομικούς και είναι συνυφασμένοι με την επιχειρηματική κουλτούρα, όπως ο βαθμός ικανοποίησης των πελατών και του εργοδοτούμενου προσωπικού.

1.9 Συνεισφορά της Εργασίας

Με την πάροδο του χρόνου, ο τομέας της βιομηχανίας έχει εδραιωθεί ως ο μεγαλύτερος, σε ποσοστά απασχόλησης, σε ολόκληρο τον κόσμο. Συνεπώς είναι επόμενο να αυξάνεται αντίστοιχα και ο ανταγωνισμός. Κατα αυτόν τον τρόπο η εκάστοτε βιομηχανία είτε προσαρμόζεται στα συνεχώς εναλλασσόμενα δεδομένα είτε δεν καταφέρνει να ακολουθήσει αυτούς του ρυθμούς και δημιουργούνται προβλήματα. Ο σκοπός και η συνεισφορά αυτής της εργασίας είναι η αντιμετώπιση των νέων αυτών δεδομένων και η προσαρμογή σε αυτά στον τομέα της παραγωγής ενός εργοστασίου. Μερικά από τα ζητήματα τα οποία μπορεί να αντιμετωπίσει αυτή η εργασία είναι τα παρακάτω :

1. Πώς θα μπορούσαμε να βελτιώσουμε τον τρόπο που προγραμματίζουμε την παραγωγή, προκειμένου να πετύχουμε καλύτερα αποτελέσματα.
2. Πώς θα μπορούσαμε να αποφύγουμε ή να προβλέψουμε πιθανούς κινδύνους.
3. Πώς θα μπορούσαμε να μειώσουμε το ρίσκο σε μια μεγάλης σημασίας απόφαση.

Κεφάλαιο 2

Μέθοδοι Αναλύσεων και Προβλέψεων στην Βιομηχανία

2.1 Ανάλυση Δεδομένων στην Βιομηχανία

Τα σημερινά στελέχη παραγωγής αντιμετωπίζουν ένα νέο τοπίο, το οποίο συμβάλλει στην κερδοφορία. Από τους περιορισμούς της προσφοράς πρώτων υλών στον αυξανόμενο αριθμό και την πολυπλοκότητα των παραγωγικών δραστηριοτήτων που εμπλέκονται στη διαδικασία παραγωγής, οι κατασκευαστές θα μπορούσαν να επωφεληθούν από μια πιο λεπτομερή προσέγγιση στη διάγνωση και τη διόρθωση ζητημάτων διεργασιών. Είναι γνωστό ότι τα προϊόντα είναι τα βασικότερα στοιχεία για μια βιομηχανία, τα δεδομένα και τα αναλυτικά στοιχεία συχνά στοχεύουν στην τελειοποίηση της ανάπτυξης του προϊόντος. Συνεπώς, όλο και περισσότεροι κατασκευαστές εφαρμόζουν διαδικασίες ανάλυσης δεδομένων για να βελτιστοποιήσουν τις αλυσίδες εφοδιασμού (Supply Chain) τους, να βελτιώσουν τον προγραμματισμό των προϊόντων, να προβλέψουν τις πωλήσεις, να μειώσουν το κόστος, να αναπτύξουν νέες προτάσεις και να χρησιμοποιήσουν την αξιοπιστία της μηχανής.

2.2 Πρόβλεψη

Με τον όρο πρόβλεψη αναφερόμαστε στη διαδικασία αξιοποίησης της διαθέσιμης γνώσης για την παραγωγή ισχυρισμών για γεγονότα, των οποίων τα πραγματικά αποτελέσματα συνήθως δεν έχουν ακόμα παρατηρηθεί. Η πρόβλεψη έχει χαρακτήρα στοχαστικό και δημιουργείται από την ανάγκη του ανθρώπου για διορατικότητα στο βάθος του χρόνου. Έτσι και στον τομέα της βιομηχανίας οι προβλέψεις έχουν τεράστια σημασία.

2.2.1 Κλασσικές Τεχνικές Προβλέψεων

Ανάλογα τα δεδομένα που εξετάζουμε, για να έχουμε ως αποτέλεσμα μια αξιόπιστη πρόβλεψη, χρησιμοποιούμε συγκεκριμένες τεχνικές. Τέτοιες τεχνικές είναι :

- i. Κινητοί Μέσοι Όροι.
- ii. Μοντέλο Γραμμικής Τάσης.
- iii. ARIMA κ.α.

2.3 Μηχανική και Βαθιά Μάθηση (Machine and Deep Learning)

Καθώς εξηγήσαμε στην εισαγωγή την έννοια της τεχνητής νοημοσύνης σε αυτό το κεφάλαιο θα μιλήσουμε για το γιατί πλέον όλοι ασχολούνται τόσο έντονα με την τεχνητή νοημοσύνη και το πώς αυτή εφαρμόζεται στην πράξη.

2.3.1 Γιατί τεχνητή νοημοσύνη τώρα ;

Σήμερα εξαιτίας των αλγοριθμικών προόδων, των πολλαπλασιασμό των δεδομένων και τις τεράστιες αυξήσεις στην υπολογιστική ισχύ και χώρου αποθήκευσης η τεχνητή νοημοσύνη ήρθε πολύ έντονα στο προσκήνιο.

Από το 1805 ήδη η εύρεση και βελτιστοποίηση των αλγορίθμων αυτών έχει αρχίσει και συνεχίζεται πολύ έντονα μέχρι και σήμερα. Αρχικά μετά το 1991 άνοιξε ο Παγκόσμιος Ιστός και ήρθε η πρώτη έκρηξη δεδομένων όπου και συνεχίζεται προφανώς μέχρι και σήμερα. Τέλος το 1965 αρχίζει και η εκθετική αύξηση της ισχύος στον κόσμο. Οι τρεις παραπάνω ημερομηνίες ήταν η αφετηρία για έναν τέλειο συνδυασμό για να περάσει η τεχνητή νοημοσύνη από την θεωρία στην πράξη. Πράγματι το 2009 ανακοινώνεται το Spark, ένα εργαλείο το οποίο ήταν ικανό να καθιστά την επεξεργασία των δεδομένων στους δίσκους πολύ ταχύτερη και επανέφερε την δυνατότητα ενημέρωσης μεγάλων δεδομένων και εκτέλεσης αναλυτικών στοιχείων σε πραγματικό χρόνο. Μέχρι και σήμερα η τεχνητή νοημοσύνη γνωρίζει τεράστια ακμή και ακόμα γευτεί πολύ λίγους από τους καρπούς της.

2.3.2 Machine Learning

Οι πιο πρόσφατες εξελίξεις στο AI έχουν επιτευχθεί με την εφαρμογή της εκμάθησης μηχανών σε πολύ μεγάλα σύνολα δεδομένων. Οι αλγόριθμοι μηχανικής μάθησης ανιχνεύουν τα πρότυπα και μαθαίνουν πώς να κάνουν προβλέψεις και συστάσεις επεξεργαζόμενοι δεδομένα και εμπειρίες, παρά να λαμβάνουν ρητές οδηγίες προγραμματισμού. Οι αλγόριθμοι προσαρμόζονται επίσης ως απάντηση σε νέα δεδομένα και εμπειρίες για τη βελτίωση της αποτελεσματικότητας με την πάροδο του χρόνου.

Θα μπορούσαμε να διατηρήσουμε τα είδη των αλγορίθμων μηχανικής μάθησης σε τρεις μεγάλες κατηγορίες.

1. Αλγόριθμοι ικανοί να :
 - (α') Περιγράφουν τι συνέβει
 - (β') Χρησιμοποιούνται από πολλές βιομηχανίες
2. Να Προβλέπουν
 - (α') Ικανότητα να αντιλαμβάνονται τι θα συμβεί
 - (β') Ενασχόληση με την επεξεργασία των δεδομένων
3. Αλγόριθμοι ικανοί να προτείνουν την καλύτερη λύση σε ένα πρόβλημα
 - (α') Κάνουν προτάσεις για την καλύτερη στρατηγική για να πετύχεις τον στόχο σου
 - (β') Επίσης ενασχόληση με την επεξεργασία δεδομένων

Οι δύο τελευταίες κατηγορίες εστιάζουν στο machine learning σε πρακτικό επίπεδο.

Κυρίαρχοι Τύποι Machine Learning

1. Επιβλεπόμενη Μάθηση (Supervised Learning)

(α') **Τι είναι η επιβλεπόμενη μάθηση ;**

Ένας αλγόριθμος χρησιμοποιεί δεδομένα κατάρτισης και ανατροφοδότησης από τον άνθρωπο για να μάθει τη σχέση των δεδομένων εισροών με την έξοδο που έχει οριστεί.

(β') **Πότε τον χρησιμοποιούμε ;**

Όταν γνωρίζουμε πώς να ταξινομήσουμε δεδομένα τα οποία θα τα χρησιμοποιήσουμε σαν είσοδο και τι θέλουμε να προβλέψουμε, αλλά χρειαζόμαστε τον αλγόριθμο και τον κατάλληλο αλγόριθμο για να υπολογίσουμε την έξοδο για νέες εικόνες

(γ') **Πώς δουλεύει ;**

- i. Ορίζουμε αρχικά τα χαρακτηριστικά των δεδομένων εισόδου και στην συνέχεια ορίζουμε την μεταβλητή εξόδου.
- ii. Εκπαιδεύεται ο αλγόριθμος προκειμένου να βρεί την σύνδεση μεταξύ των μεταβλητών εισόδου και εξόδου.
- iii. Όταν τελικά εκπαιδεύεται ο αλγόριθμος και είναι αποδοτικός τότε τον εφαρμόζουμε σε νέα δεδομένα.

(δ') **Αλγόριθμοι επιβλεπόμενης Μάθησης :**

- i. Linear Regression
- ii. Logistic Regression
- iii. Decision Tree
- iv. Naive Bayes
- v. Support Vector Machine

- vi. Random Forest
- vii. Ada Boost
- viii. Gradient Boosting Tree
- ix. Simple Neural Network

2. Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning)

(α') Τι είναι η μη επιβλεπόμενη μάθηση ;

Είναι ένα σύνολο αλγορίθμων οι οποίοι διερευνούμε τα δεδομένα εισόδου χωρίς όμως να υπάρχει συγκεκριμένη μεταβλητή εισόδου

(β') Πότε τον χρησιμοποιούμε ;

Όταν δεν γνωρίζουμε τον τρόπο τον οποίο ταξινομούνται τα δεδομένα και θέλουμε ο αλγόριθμος να κάνει αυτή την διαδικασία για εμάς

(γ') Πώς δουλεύει ;

- i. Ο αλγόριθμος λαμβάνει δεδομένα τα οποία δεν έχουν προκαθορισμένα χαρακτηριστικά
- ii. Συμπεραίνει μια δομή για τα δεδομένα
- iii. Στην συνέχεια ο αλγόριθμος αναγνωρίζει ομάδες δεδομένων που παρουσιάζουν παρόμοια συμπεριφορά και εκτελεί ότι διαδικασία του ορίσουμε.

(δ') Αλγόριθμοι μη επιβλεπόμενης Μάθησης :

- i. K-means Clustering
- ii. Gaussian mixture model
- iii. Hierarchical Clustering
- iv. Recommender System

3. Ενισχυτική Μάθηση (Reinforcement Learning)

(α') Τι είναι η ενισχυτική μάθηση ;

Ένας αλγόριθμος μαθαίνει να εκτελεί ένα έργο απλά προσπαθώντας να μεγιστοποιήσει τις ανταμοιβές που λαμβάνει για τις πράξεις του.

(β') Πότε τον χρησιμοποιούμε ;

Όταν δεν έχουμε πολλά εκπαιδευτικά δεδομένα. δεν μπορούμε να ορίσουμε σαφώς την ιδανική τελική κατάσταση. ή ο μόνος τρόπος για να μάθουμε για το περιβάλλον είναι να αλληλεπιδράσουμε με αυτό.

(γ') Πώς δουλεύει ;

- i. Ο αλγόριθμος αναλαμβάνει δράση για το περιβάλλον.
- ii. Λαμβάνει μια ανταμοιβή αν η ενέργεια φέρνει το μηχάνημα ένα βήμα πιο κοντά στη μεγιστοποίηση των συνολικών ανταμοιβών που είναι διαθέσιμες.
- iii. Ο αλγόριθμος βελτιστοποιείται συνεχώς καιρό με τον καιρό.

(δ') Η χρήση του σε πραγματικά Business Cases

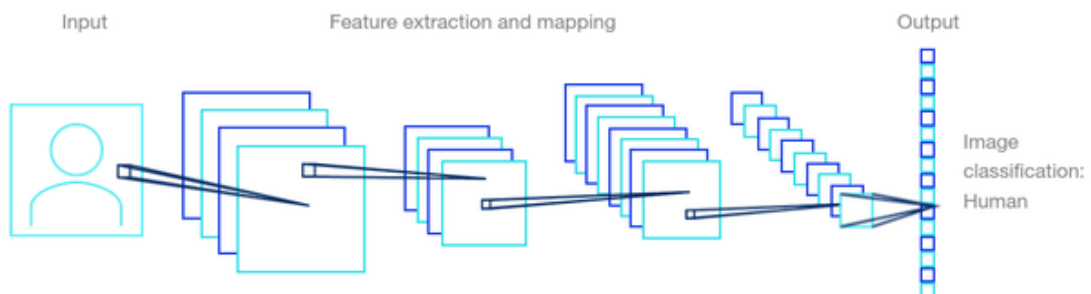
- i. Βελτιστοποιήστε τη στρατηγική συναλλαγών για ένα χαρτοφυλάκιο δικαιωμάτων προαίρεσης.
- ii. Στον τομέα των αποθεμάτων χρησιμοποιώντας ρομπότ.
- iii. Βελτίωση της συμπεριφοράς σε αυτοκίνητα που δεν χρειάζονται οδηγό.
- iv. Εξισορρόπηση του ηλεκτρικού φορτίου ζήτησης στα δίκτυα.

2.3.3 Deep Learning

Η βαθιά εκμάθηση είναι ένας τύπος μηχανικής μάθησης που μπορεί να επεξεργαστεί ένα ευρύτερο φάσμα πόρων δεδομένων, απαιτεί λιγότερη επεξεργασία δεδομένων από τους ανθρώπους και μπορεί συχνά να παράγει ακριβέστερα αποτελέσματα από τις παραδοσιακές προσεγγίσεις μηχανικής μάθησης (αν και απαιτεί μεγαλύτερο όγκο δεδομένων για να το πράξει). Στη βαθιά εκμάθηση, τα διασυνδεδεμένα στρώματα των υπολογιστών που βασίζονται σε λογισμικό γνωστά ως 'νευρώνες' σχηματίζουν ένα νευρωνικό δίκτυο. Το δίκτυο μπορεί να καταναλώνει τεράστιες ποσότητες δεδομένων εισόδου και να τις επεξεργάζεται μέσω πολλαπλών επιπέδων που μαθαίνουν όλο και πιο πολύπλοκα χαρακτηριστικά των δεδομένων σε κάθε στρώμα. Το δίκτυο μπορεί στη συνέχεια να λάβει μια απόφαση σχετικά με τα δεδομένα, να μάθει αν η απόφασή του είναι σωστή και να χρησιμοποιήσει ό,τι έχει μάθει για να κάνει προσδιορισμούς σχετικά με νέα δεδομένα. Για παράδειγμα, μόλις μάθει τι μοιάζει με ένα αντικείμενο, μπορεί να αναγνωρίσει το αντικείμενο σε μια νέα εικόνα.

Κυρίαρχοι Τύποι Deep Learning**1. Συνελικτικά Νευρωνικά Δίκτυα Convolutional Neural Networks****(α') Τι είναι τα συνελικτικά νευρωνικά δίκτυα ;**

Ένα πολυεπίπεδο νευρωνικό δίκτυο με μια ειδική αρχιτεκτονική σχεδιασμένη να εξαγάγει όλο και πιο πολύπλοκα χαρακτηριστικά των δεδομένων σε κάθε στρώμα για να καθορίσει την έξοδο.



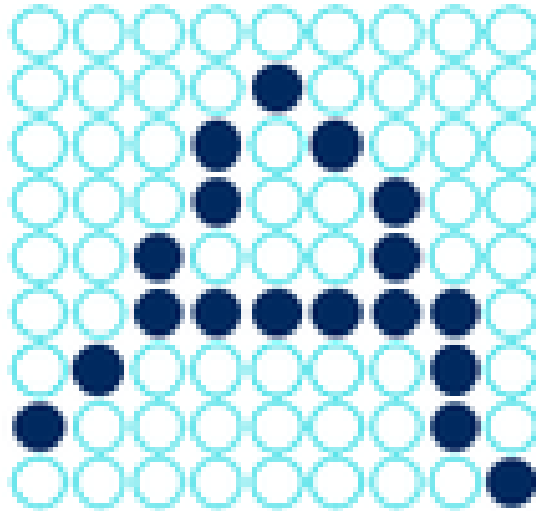
Σχήμα 2.1: Convolutional Neural Network

(β') Πότε το χρησιμοποιούμε ;

Όταν έχω μη δομημένα σετ δεδομένων και πρέπει να εξάγουμε κάποιες πληροφορίες από αυτά.

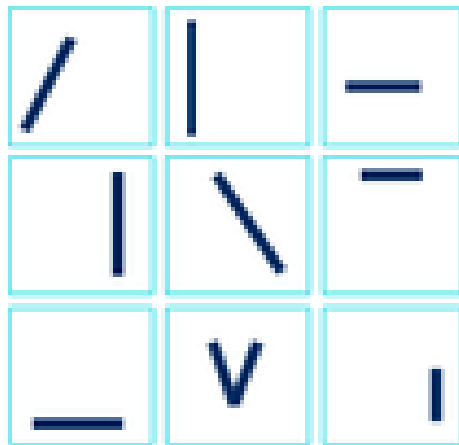
(γ') Πώς δουλεύει ;

- i. Το συνελκτικό νευρωνικό δίκτυο λαμβάνει μία εικόνα για παράδειγμα το γράμμα A , το οποίο είναι μια συγκεκριμένη δομή pixels



Σχήμα 2.2: Γράμμα A

- ii. Στην συνέχεια στα κρυμμένα στρώματα του νευρωνικού εντοπίζονται μοναδικά χαρακτηριστικά στην εικόνα που δέχεται ως είσοδο, για παράδειγμα μεμονωμένες γραμμές που συντελούν το A



Σχήμα 2.3: Γραμμές για την συντέλεση του A

- iii. Πλέον είναι σε θέση να αναγνωρίσει γράμματα και να ταξινομήσει διαφορετικές

εικόνες εάν εντοπίσει σε αυτές μοναδικούς τρόπους όπως είχε εκπαιδευτεί.



Σχήμα 2.4: Τελική έξοδος

(δ') **Πού χρησιμοποιούνται ;**

- i. Διάγνωση προβλημάτων υγείας από εικόνες
- ii. Κατανόηση της καταναλωτικής συμπεριφοράς των ανθρώπων
- iii. Εντοπισμός προβληματικών προϊόντων στην παραγωγή μέσω εικόνων.

2. Επαναλαμβανόμενα Νευρωνικά Δίκτυα (Recurrent Neural Networks)

(α') **Τι είναι τα επαναλαμβανόμενα νευρωνικά δίκτυα ;**

Ένα πολυεπίπεδο νευρωνικό δίκτυο που μπορεί να αποθηκεύσει πληροφορίες σε κόμβους περιβάλλοντος, επιτρέποντάς του να μάθει ακολουθίες δεδομένων και να εξάγει έναν αριθμό ή άλλη ακολουθία

(β') **Πότε το χρησιμοποιούμε ;**

Τα χρησιμοποιούμε όταν δουλεύουμε με χρονικές ακολουθίες και αλληλουχίες δεδομένων

(γ') **Πώς δουλεύει ;**

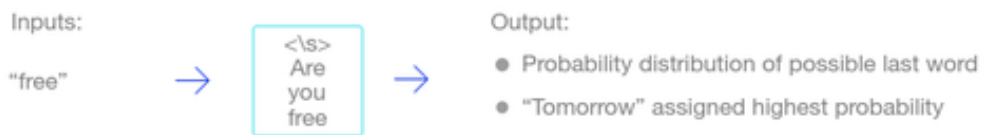
Άλλες αρχιτεκτονικές νευρικών δικτύων υποθέτουν ότι όλες οι εισόδους είναι ανεξάρτητες το ένα από το άλλο. Αλλά αυτή η υπόθεση δεν λειτουργεί καλά για ορισμένα καθήκοντα. Πάρτε, για παράδειγμα, το έργο της πρόβλεψης της επόμενης λέξης σε μια πρόταση - είναι ευκολότερο να προβλέψετε την επόμενη λέξη εάν είναι γνωστές αρκετές λέξεις που ήρθαν πριν.

- i. Σε ένα Επαναλαμβανόμενο Νευρωνικό Δίκτυο, ο νευρώνας λαμβάνει μια εντολή και πυροδοτεί την αφετηρία της πρότασης
- ii. Ο νευρώνας λαμβάνει τη λέξη "Are" και στη συνέχεια εξάγει έναν φορέα αριθμών που τροφοδοτεί πίσω στον νευρώνα για να τον βοηθήσει να "θυμάται"

ότι έλαβε "Are" (και ότι το έλαβε πρώτα). Η ίδια διαδικασία συμβαίνει όταν λαμβάνει "you" και "free", με την κατάσταση του νευρώνα να ενημερώνεται κατά τη λήψη κάθε λέξης.

- iii. μετά τη λήψη του "free", ο νευρώνας εκχωρεί μια πιθανότητα σε κάθε λέξη στο αγγλικό λεξιλόγιο που θα μπορούσε να ολοκληρώσει την πρόταση. Αν εκπαιδευτεί καλά, το δίκτυο θα αναθέσει τη λέξη "tomorrow" μία από τις υψηλότερες πιθανότητες και θα το επιλέξει για να ολοκληρώσει την πρόταση.

Predicting the next word in the sentence "Are you free _____?"



Σχήμα 2.5: Πρόβλεψη της επόμενης λέξης.

(δ') Πού χρησιμοποιούνται ;

- i. Χρήσιμο στην καλύτερη δυνατή μετάφραση ξένων γλωσσών.
- ii. Πρόβλεψη χρηματιστηριακών δεικτών
- iii. Ασφάλεια στις συναλλαγές με πιστωτική κάρτα

Κεφάλαιο 3

Αλγόριθμοι Machine και Deep Learning

Στο κεφάλαιο αυτό θα αναλύσουμε την εργαλεία που χρησιμοποιήθηκαν όπως γλώσσα προγραμματισμού, βιβλιοθήκες και θα γίνει ανάλυση σε βάθος των αλγορίθμων μηχανικής και βαθιάς μάθησης που χρησιμοποιήθηκαν στην εργασία, με σκοπό εκτέλεσης πειραμάτων για καταλήξουμε στο καταλληλότερο αλγόριθμο που θα χρησιμοποιήσουμε στην ανάπτυξη της εφαρμογής.

3.1 Εργαλειοθήκη

3.1.1 Python

Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για να δημιουργηθεί το πρόγραμμα είναι η Python. Η Python είναι μια υψηλού επιπέδου γλώσσα προγραμματισμού η οποία δημιουργήθηκε από τον Ολλανδό Γκίντο βαν Ρόσσουμ (Guido van Rossum) το 1990. Ο κύριος στόχος της είναι η αναγνωσιμότητα του κώδικά της και η ευκολία χρήσης της και το συντακτικό της επιτρέπει στους προγραμματιστές να εκφράσουν έννοιες σε λιγότερες γραμμές κώδικα απ ότι θα ήταν δυνατόν σε γλώσσες όπως η C++ ή η Java. Διακρίνεται λόγω του ότι έχει πολλές βιβλιοθήκες που διευκολύνουν ιδιαίτερα αρκετές συνηθισμένες εργασίες και για την ταχύτητα εκμάθησης της. Οι διερμηνευτές της Python είναι διαθέσιμοι για εγκατάσταση σε πολλά λειτουργικά συστήματα, επιτρέποντας στην Python την εκτέλεση κώδικα σε ευρεία γκάμα συστημάτων. Χρησιμοποιώντας εργαλεία τρίτων, όπως το Py2exe ή το Pyinstaller, ο κώδικας της Python μπορεί να πακεταριστεί σε αυτόνομα εκτελέσιμα προγράμματα για μερικά από τα πιο δημοφιλή λειτουργικά συστήματα, επιτρέποντας τη διανομή του βασισμένου σε Python λογισμικού για χρήση σε αυτά τα περιβάλλοντα χωρίς να απαιτείται εγκατάσταση του διερμηνευτή της Python. Η Python αναπτύσσεται ως ανοιχτό λογισμικό (open source) και η διαχείρισή της γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation. Ο κώδικας διανέμεται με την άδεια Python Software Foundation License η οποία είναι συμβατή με την GPL. Το όνομα της γλώσσας προέρχεται από την ομάδα άγγλων κωμικών Μόντι

Πάιθον.

3.1.2 Βιβλιοθήκες

NumPy

Η NumPy είναι το θεμελιώδες πακέτο για την επιστημονική πληροφορική στη Python. Είναι μια βιβλιοθήκη της Python που παρέχει ένα αντικείμενο πολυδιάστατων πινάκων, διάφορα παράγωγα αντικείμενα (όπως πίνακες με 'μάσκες' και μήτρες) και μια συλλογή ρουτινών για γρήγορες λειτουργίες σε πίνακες, συμπεριλαμβανομένων μαθηματικών, λογικών, χειρισμών σχήματος, διαλογής, επιλογής, εισόδου / εξόδου, διακριτοί μετασχηματισμοί Fourier, βασική γραμμική άλγεβρα, βασικές στατιστικές λειτουργίες, τυχαία προσομοίωση και πολλά άλλα. Στον πυρήνα του πακέτου NumPy, είναι το αντικείμενο ndarray. Αυτό ενσωματώνει n-διάστατες συστοιχίες ομοιογενών τύπων δεδομένων, με πολλές λειτουργίες να εκτελούνται σε compiledκώδικα για απόδοση. Υπάρχουν πολλές σημαντικές διαφορές μεταξύ των πινάκων NumPy και των τυπικών ακολουθιών Python:

1. Οι αριθμητικές πίνακες NumPy έχουν σταθερό μέγεθος στη δημιουργία, σε αντίθεση με τις λίστες της Python (οι οποίες μπορούν να αναπτυχθούν δυναμικά). Η αλλαγή του μεγέθους ενός ndarray θα δημιουργήσει έναν νέο πίνακα και θα διαγράψει το πρωτότυπο.
2. Τα στοιχεία σε ένα πίνακα NumPy πρέπει όλα να είναι του ίδιου τύπου δεδομένων και έτσι θα έχουν το ίδιο μέγεθος στη μνήμη. Η εξαίρεση: μπορούμε να έχουμε πίνακες αντικειμένων (Python, συμπεριλαμβανομένων των NumPy), επιτρέποντας έτσι πίνακες στοιχείων διαφορετικού μεγέθους.
3. Οι αριθμητικές σειρές διευκολύνουν τους προχωρημένους μαθηματικούς και άλλους τύπους επιχειρήσεων σε μεγάλο αριθμό δεδομένων. Τυπικά, τέτοιες λειτουργίες εκτελούνται πιο αποτελεσματικά και με λιγότερους κωδικούς από ό, τι είναι δυνατό χρησιμοποιώντας τις ενσωματωμένες ακολουθίες της Python.
4. Μια αυξανόμενη πληθώρα επιστημονικών και μαθηματικών πακέτων που βασίζονται στην Python χρησιμοποιούν τους αριθμητικούς πίνακες NumPy, αν και αυτά συνήθως υποστηρίζουν είσοδο ακολουθίας Python, μετατρέπουν τέτοιες εισροές σε πίνακες NumPy πριν από την επεξεργασία και συχνά εξάγουν πίνακες NumPy. Με άλλα λόγια, για να χρησιμοποιήσει αποτελεσματικά πολύ (ίσως ακόμη και το μεγαλύτερο μέρος) του επιστημονικού / μαθηματικού λογισμικού που βασίζεται στη Python, είναι απλώς γνωστό πώς να χρησιμοποιήσει τους ενσωματωμένους τύπους αλληλουχιών της Python είναι ανεπαρκής - πρέπει επίσης να γνωρίζουμε πώς να χρησιμοποιούμε τους NumPy πίνακες.

Pandas

Στον προγραμματισμό υπολογιστών, η pandas είναι μια βιβλιοθήκη λογισμικού γραμμένη για τη γλώσσα προγραμματισμού Python για χειρισμό και ανάλυση δεδομένων. Συγκεκριμένα, προσφέρει δομές δεδομένων και λειτουργίες για τον χειρισμό αριθμητικών πινάκων και

χρονοσειρών. Είναι ελεύθερο λογισμικό που εκδίδεται βάσει της άδειας BSD με τρεις ρήτρες. Το όνομα προέρχεται από τον όρο *panel data*, έναν οικονομετρικό όρο για σύνολα δεδομένων που περιλαμβάνουν παρατηρήσεις για πολλαπλές χρονικές περιόδους για τα ίδια άτομα.

Scikit Learn

Η Scikit-learn (πρώην scikits.learn) είναι μια βιβλιοθήκη για machine learning ελεύθερου λογισμικού για τη γλώσσα προγραμματισμού Python. Διαθέτει διάφορους αλγόριθμους ταξινόμησης, παλινδρόμησης και ομαδοποίησης, συμπεριλαμβανομένων των μηχανισμών vector machines, random forest, gradient boosting, k-means και DBSCAN και έχει σχεδιαστεί για να συνεργάζεται με τις αριθμητικές και επιστημονικές βιβλιοθήκες της Python NumPy και SciPy.

Tensorflow

Το TensorFlow είναι μια βιβλιοθήκη λογισμικού ανοιχτού κώδικα για προγραμματισμό ροής δεδομένων σε μια σειρά εργασιών. Είναι μια συμβολική βιβλιοθήκη μαθηματικών και χρησιμοποιείται επίσης για εφαρμογές μηχανικής μάθησης όπως νευρωνικά δίκτυα.

Keras

Η Keras είναι υψηλού επιπέδου νευρωνικών δικτύων βιβλιοθήκη, γραμμένη σε Python και ικανή να τρέχει πάνω από το TensorFlow, το CNTK ή το Theano. Πρόκειται για μια βιβλιοθήκη που είναι χρήσιμη σε εφαρμογές που αφορούν το deep learning. Αναπτύχθηκε με έμφαση στο γρήγορο πειραματισμό. Έχει την ικανότητα να μεταβαίνει από την ιδέα σε αποτέλεσμα με την ελάχιστη δυνατή καθυστέρηση, το οποίο είναι το κλειδί για μια καλή έρευνα.

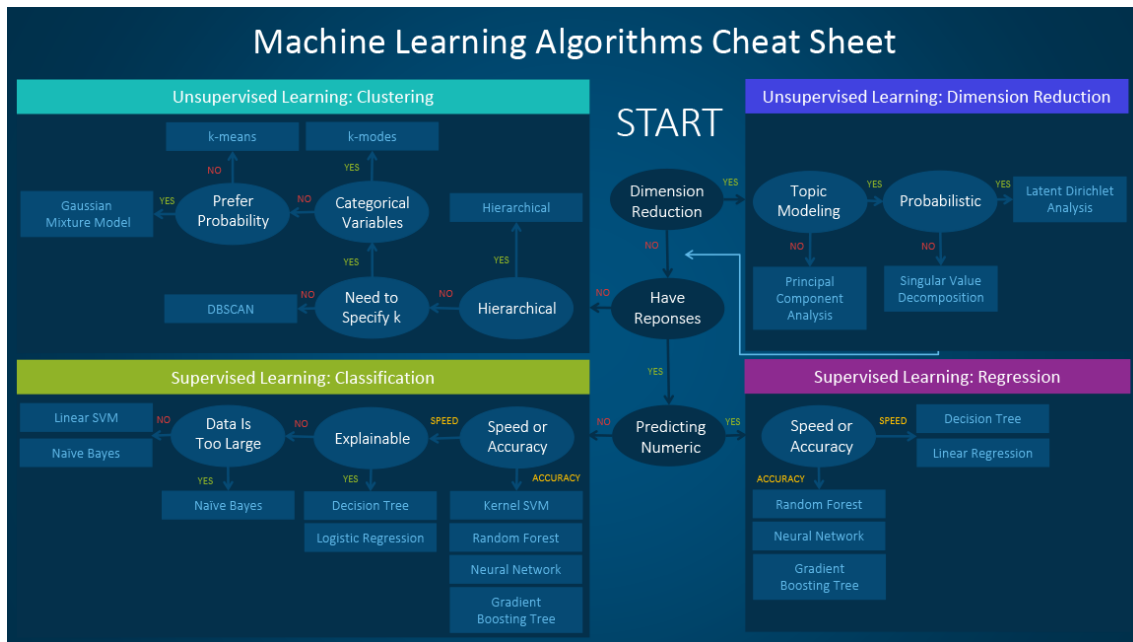
Theano

Το Theano είναι μια βιβλιοθήκη Python και βελτιστοποιεί τον μεταγλωττιστή για τον χειρισμό και την αξιολόγηση των μαθηματικών εκφράσεων, Στο Theano, οι υπολογισμοί εκφράζονται χρησιμοποιώντας συντακτικό NumPy-esque και μεταγλωττίζονται για να λειτουργούν αποτελεσματικά σε αρχιτεκτονικές CPU ή GPU. Το Theano είναι ένα έργο ανοιχτού κώδικα που αναπτύχθηκε κυρίως από το Ινστιτούτο Αλγορίθμων Μάθησης του Μόντρεαλ.

3.2 Αλγόριθμοι Machine Learning

Στο προηγούμενο κεφάλαιο αναλύσαμε τι είναι το machine learning και αναφέραμε κάποιους από τους αλγόριθμους του. Σε αυτό θα εμβαθύνουμε σε εκείνους που επιλέχθηκαν και χρησιμοποιήθηκαν στα πειράματα που έγιναν στην εργασία.

Το παρακάτω σχεδιάγραμμα είναι ένας οδηγός για την επιλογή του καταλληλότερου αλγόριθμου machine learning ανάλογα με την χρήση που θέλουμε να κάνουμε :



Σχήμα 3.1: Χρήσιμο φυλλάδιο για την επιλογή αλγορίθμου μηχανικής μάθησης

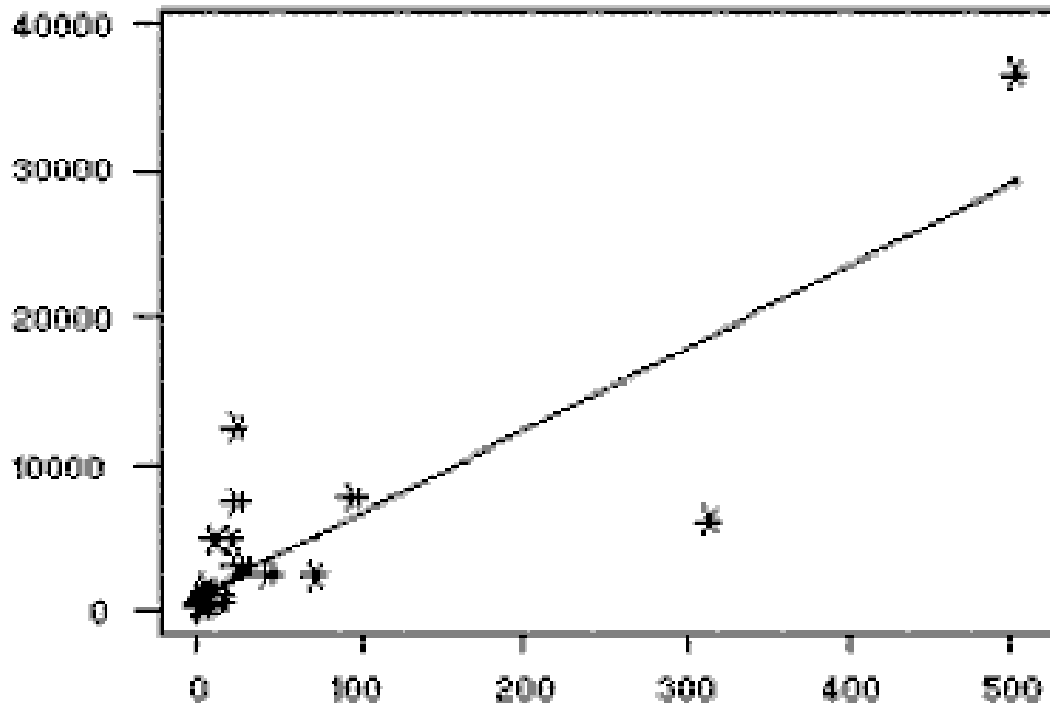
Μετά από την μελέτη αυτού του φυλλαδίου και έχοντας αποσαφηνίσει την δουλειά που θέλουμε να κάνουμε προχωράμε στο πώς θα την κάνουμε.

Καταρχάς το πεδίο το οποίο χτίζεται η εργασία αφορά αλγόριθμους επιβλεπόμενης μάθησης. Ήδη έχουμε αναφέρει κάποιους από τους αλγόριθμους που χρησιμοποιήθηκαν. Αυτοί οι αλγόριθμοι είναι :

1. Linear Regression
2. Decision Tree
3. Random Forest
4. Gradient Boosting Tree
5. Artificial Neural Networks (Multilayer Perceptron)

3.2.1 Linear Regression

Ο πρώτος αλγόριθμος που χρησιμοποιήθηκε είναι η γραμμική παλινδρόμηση. Η γραμμική παλινδρόμηση ή Linear Regression είναι ένας από τους πιο βασικούς και συνήθεις τύπος των Predictive Analytics. Η γραμμική παλινδρόμηση επιχειρεί να μοντελοποιήσει τη σχέση μεταξύ δύο μεταβλητών εφαρμόζοντας μια γραμμική εξίσωση στα παρατηρούμενα δεδομένα. Μια μεταβλητή θεωρείται ερμηνευτική μεταβλητή και η άλλη θεωρείται εξαρτημένη μεταβλητή.



Σχήμα 3.2: Linear Regression

Εκτέλεση του αλγορίθμου

Στο σημείο αυτό θα παρουσιάσουμε το πρώτο πείραμα που έγινε προκειμένου να ελέγξουμε αν ο αλγόριθμος Linear Regression είναι αρκετά αποδοτικός.

Αρχικά έχουμε στην διάθεση μας ένα dataset το οποίο αφορά μία γραμμή παραγωγής της βιομηχανίας που εξετάζουμε την χρονική διάρκεια από τον Ιούλιο του 2017 μέχρι τον Ιούνιο του 2018. Το dataset έχει την παρακάτω μορφή :

1. Ημερομηνία
2. Βάρδια
3. Κωδικός Προϊόντος
4. Συνολικά κομμάτια που παρήχθησαν
5. Ταχύτητα της μηχανής (π.χ. 70 κομμάτια το λεπτό (70 units per minute))
6. Σφάλματα
7. Δείκτης Απόδοσης
8. Δείκτης Διαθεσιμότητας
9. Δείκτης Ποιότητας
10. OEE

Πίνακας 3.1: Data Set

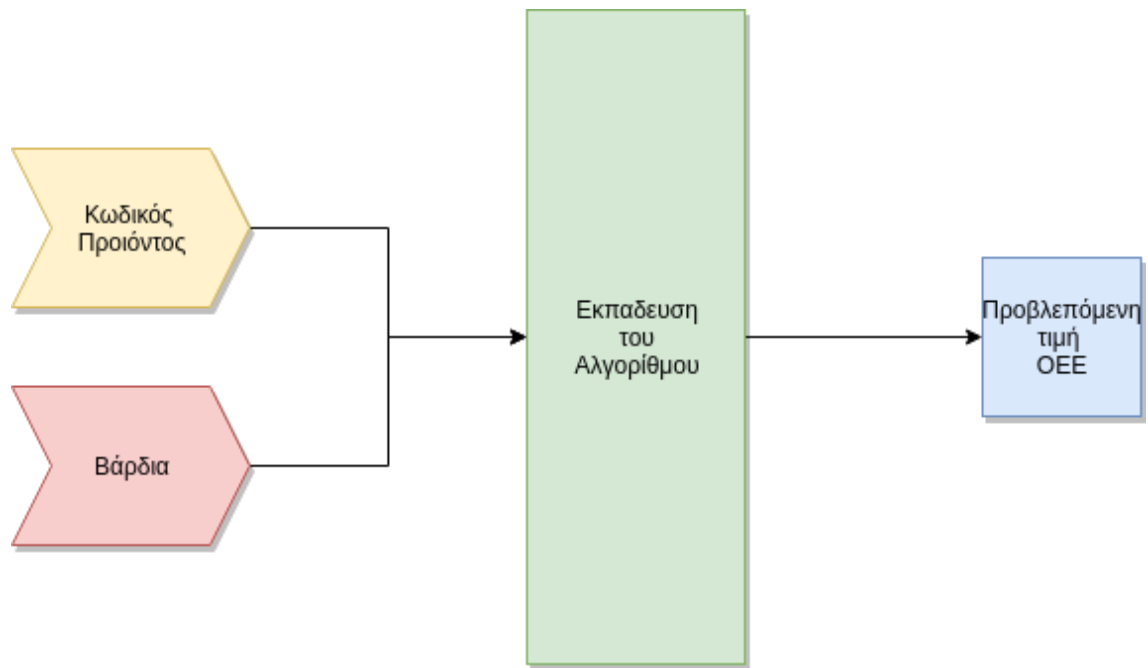
<i>Date</i>	<i>Prod code</i>	<i>Shift</i>	<i>Total parts</i>	<i>Speed</i>	<i>Faults</i>	<i>Performance</i>	<i>Availability</i>	<i>Quality</i>	<i>OEE</i>
2/3/2018	98467	A	1200	70	5	70%	80%	90%	50,4%
3/3/2018	73648	B	3200	50	4	50%	80%	78%	31,2%
4/3/2018	37323	^	690	50	3	69%	80%	80%	44,4%
...

Όλα τα παραπάνω ονομάζονται **χαρακτηριστικά** του dataset. Από όλα αυτά τα δέκα χαρακτηριστικά πρέπει να κάνουμε μερικές επιλογές που αφορούν κυρίως το τι θα μας είναι χρήσιμο προκειμένου ο αλγόριθμος να εκτελεστεί σωστά και να μας δώσει κάποια αποτελέσματα.

Αρχικά πρέπει να ορίσουμε την έξοδο. Πρέπει να ξέρουμε και να έχουμε αποφασίσει τι θα προβλέψουμε. Στην δική μας περίπτωση έχουμε επιλέξει τον δείκτη OEE για πρόβλεψη. Συνεπώς περιμένουμε η μηχανή κατόπιν εκπαίδευσης να μας δώσει μια τιμή του OEE. Στην συνέχεια πρέπει να επιλέξουμε πάνω σε τι δεδομένα θα εκπαιδευτεί η μηχανή προκειμένου να πάρουμε αυτά τα αποτελέσματα. Άρα επι τοις ουσίας ορίζουμε την είσοδο. Για λόγους ομοιότητας του τρόπου εκπαίδευσης όλων των μηχανών με κάθε αλγόριθμο θα ορίσουμε ως είσοδο τα εξής :

1. Βάρδια

2. Κωδικό Προϊόντος



Σχήμα 3.3: Σχεδιάγραμμα εκτέλεσης αλγορίθμου

Άρα έχουμε :

Είσοδος → Εκπαίδευση → Έξοδος

Κατασκευάζουμε λοιπόν τον κώδικα ο οποίος εκτελώντας τον θα μας δώσει τα αποτελέσματα του αλγορίθμου Linear Regression. Η πρόβλεψη που θέλουμε να κάνει ο αλγόριθμος μας είναι η τιμή του ΟΕΕ την επόμενη ώρα για κάθε προϊόν και κάθε βάρδια. Συνεπώς εκπαιδεύουμε τον αλγόριθμο Linear Regression σε όλα τα δεδομένα που έχουμε στην διάθεση μας και εκτελούμε τα παρακάτω πειράματα :

Πίνακας 3.2: Shift A, Linear Regression, OEE

Product Code	Predicted Value%	Real Value%
1	9.28	15.66
2	9.33	1
3	9.56	11
4	18.4	2.75
5	9.83	10
13	11.2	2

Πίνακας 3.3: Shift B, Linear Regression, OEE

Product Code	Predicted Value%	Real Value%
1	9.46	39.6
2	9.6	30.6
3	10.7	35.33
4	12.33	7.66
5	10.01	28.66
13	22	8.16

Πίνακας 3.4: Shift C, Linear Regression, OEE

Product Code	Predicted Value%	Real Value%
1	9.11	26
2	9.21	9
3	9.38	18.99
4	9.52	26.86
5	9.65	20.33
13	10.74	10.05

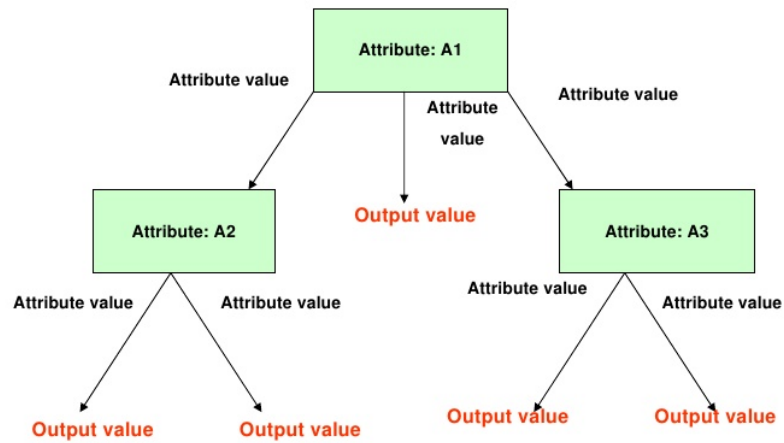
3.2.2 Decision Tree

Η εκμάθηση αυτού του είδους χρησιμοποιεί ένα δέντρο απόφασης (ως μοντέλο πρόβλεψης) για να μεταβεί από τις παρατηρήσεις για ένα στοιχείο (που αντιπροσωπεύεται στους κλάδους) σε συμπεράσματα σχετικά με την τιμή στόχου του αντικειμένου (που αναπαριστάται στα φύλλα). Πρόκειται για μία από τις προγνωστικές μεθόδους μοντελοποίησης που χρησιμοποιούνται στις στατιστικές, στην εξόρυξη δεδομένων και στη μηχανική μάθηση. Τα μοντέλα δένδρων όπου η μεταβλητή-στόχος μπορεί να πάρει ένα διακεκριμένο σύνολο τιμών ονομάζονται δέντρα ταξινόμησης. σε αυτές τις δομές δέντρων, τα φύλλα αντιπροσωπεύουν ετικέτες τάξεων και κλάδους αντιπροσωπεύουν συζεύξεις χαρακτηριστικών που οδηγούν σε αυτές τις ετικέτες κλάσης. Τα δέντρα αποφάσεων όπου η μεταβλητή-στόχος μπορεί να πάρει συνεχείς τιμές (τυπικά πραγματικοί αριθμοί) ονομάζονται δέντρα παλινδρόμησης (regression trees).

Εκτέλεση του αλγορίθμου

Η εκτέλεση του αλγορίθμου γίνεται με τον ίδιο ακριβώς τρόπο, δηλαδή ορίζουμε είσοδο/έξοδο και ακολουθούμε την ίδια διαδικασία χρησιμοποιώντας αυτή την φορά άλλο αλγόριθμο. Τα αποτελέσματα του συγκεκριμένου πειράματος είναι τα παρακάτω :

Building Decision Tree



Σχήμα 3.4: Κατασκευή Δέντρου Αποφάσεων

Πίνακας 3.5: Shift A, Decision Tree, OEE

Product Code	Predicted Value%	Real Value%
1	11.02	15.66
2	11.4	1
3	14.23	11
4	13.2	2.75
5	8.05	10
13	6.45	2

Πίνακας 3.6: Shift B, Decision Tree, OEE

Product Code	Predicted Value%	Real Value%
1	11.85	39.6
2	9.05	30.6
3	14.32	35.33
4	6.7	7.66
5	6.08	28.66
13	4.45	8.16

Πίνακας 3.7: Shift C, Decision Tree, OEE

Product Code	Predicted Value%	Real Value%
1	9.76	26
2	10.8	9
3	13.53	18.99
4	6.66	26.86
5	6.99	20.33
13	12.2	10.05

3.2.3 Random Forest

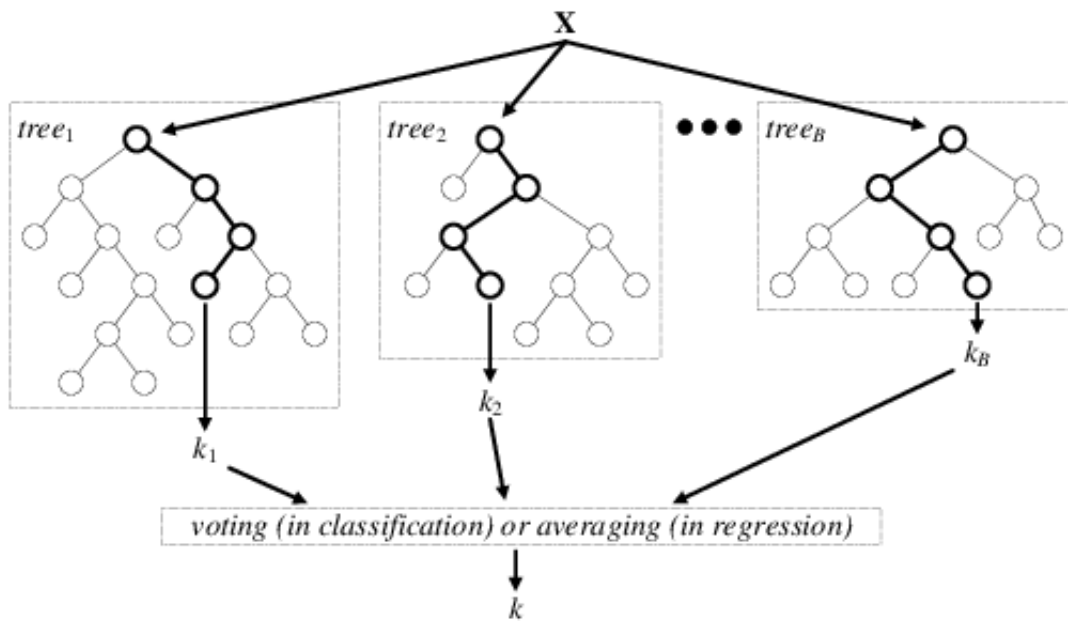
Ένας άλλος αλγόριθμος μηχανικής μάθησης που χρησιμοποιήθηκε είναι ο Random Forest. Ο Random Forest είναι μια μέθοδος μάθησης για την ταξινόμηση, την παλινδρόμηση και άλλα καθήκοντα που λειτουργούν με την κατασκευή ενός πλήθους δέντρων αποφάσεων στο χρόνο εκπαίδευσης. Επίσης διορθώνει τη συνήθεια των δέντρων αποφάσεων να προκαλούν υπερχορεσμό στο εκπαιδευτικό τους σετ (training set).

Εκτέλεση του αλγορίθμου

Ομοίως με τον ίδιο τρόπο όπως και παραπάνω λαμβάνουμε τα παρακάτω αποτελέσματα :

Πίνακας 3.8: Shift A, Random Forest, OEE

Product Code	Predicted Value%	Real Value%
1	10.88	15.66
2	11	1
3	10.92	11
4	8.45	2.75
5	6.56	10
13	7.55	2



Σχήμα 3.5: Random Forest

Πίνακας 3.9: Shift B, Random Forest, OEE

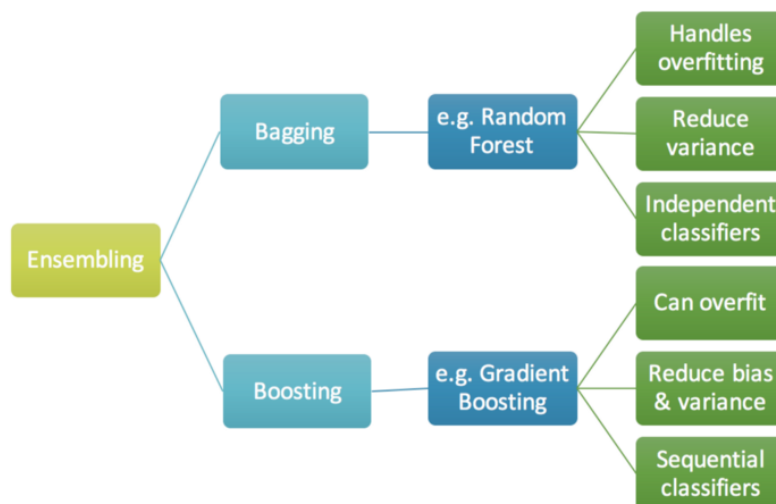
Product Code	Predicted Value%	Real Value%
1	10.88	39.6
2	10.82	30.6
3	10.92	35.33
4	10.99	7.66
5	6.56	28.66
13	6.75	8.16

Πίνακας 3.10: Shift C, Random Forest, OEE

Product Code	Predicted Value%	Real Value%
1	10.88	26
2	11.2	9
3	10.92	18.99
4	6.56	26.86
5	6.56	20.33
13	12.61	10.05

3.2.4 Gradient Boosting Tree

Ο τελευταίος αλγόριθμος μηχανικής μάθησης που χρησιμοποιήθηκε ήταν ο Gradient Boosting Tree. Ο Gradient Boosting είναι μια τεχνική μάθησης μηχανών για προβλήματα παλινδρόμησης και ταξινόμησης, η οποία παράγει ένα μοντέλο πρόβλεψης με τη μορφή ενός συνόλου αδύνατων μοντέλων πρόβλεψης, συνήθως δέντρων αποφάσεων.



Σχήμα 3.6: Gradient Boosting Tree

Εκτέλεση του αλγορίθμου

Τέλος λαμβάνουμε τα αποτελέσματα του συγκεκριμένου αλγορίθμου :

Πίνακας 3.11: Shift A, Gradient Boostion Tree, OEE

Product Code	Predicted Value%	Real Value%
1	36.57	15.66
2	40.3	1
3	35.74	11
4	18.1	2.75
5	27.06	10
13	11.33	2

Πίνακας 3.12: Shift B, Gradient Boosting Tree, OEE

Product Code	Predicted Value%	Real Value%
1	36.68	39.6
2	27.73	30.6
3	35.96	35.33
4	35.22	7.66
5	26.12	28.66
13	22.9	8.16

Πίνακας 3.13: Shift C, Gradient Boosting Tree, OEE

Product Code	Predicted Value%	Real Value%
1	34.01	26
2	32.22	9
3	33.96	18.99
4	25.82	26.86
5	24.76	20.33
13	38.54	10.05

3.3 Αλγόριθμοι Deep Learning

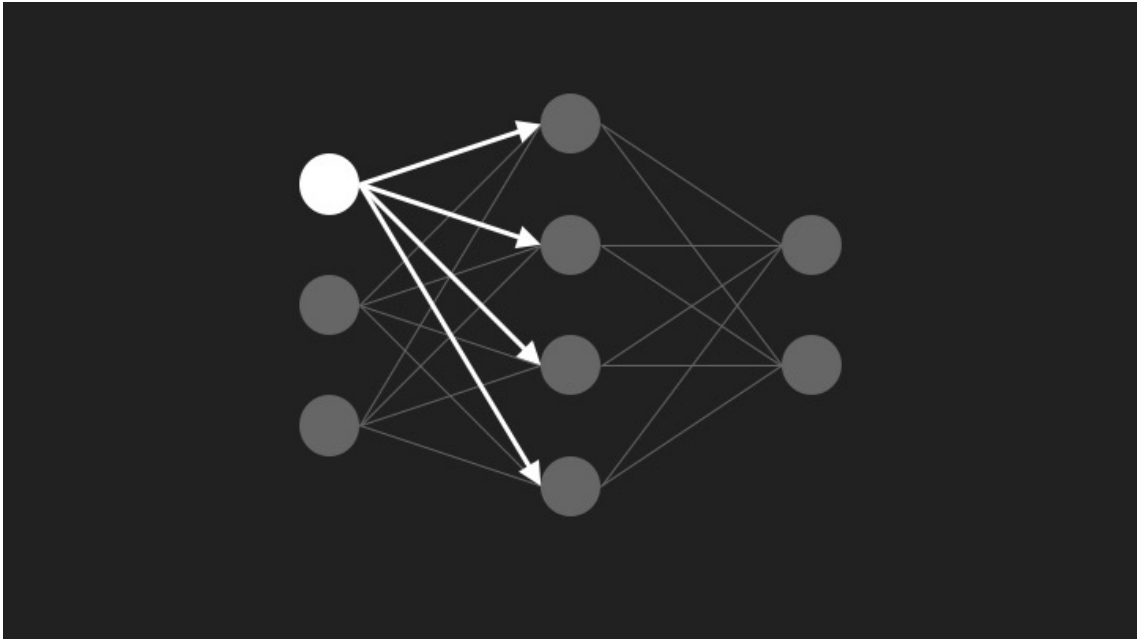
3.3.1 Artificial Neural Networks

Τα τεχνητά νευρικά δίκτυα είναι ένα από τα πιο ισχυρά εργαλεία που χρησιμοποιούνται στη μηχανική μάθηση και την τεχνητή νοημοσύνη. Εμπνευσμένοι από τα βιολογικά νευρωνικά δίκτυα, μπορούν να αντιμετωπίσουν ένα ευρύ φάσμα προβλημάτων που καταστρέφουν το παραδοσιακό AI, συμπεριλαμβανομένης της όρασης του υπολογιστή, της αναγνώρισης ομιλίας, της επεξεργασίας φυσικής γλώσσας και της ρομποτικής. Σε γενικές γραμμές, τα νευρικά δίκτυα μπορούν να χρησιμοποιηθούν για οποιαδήποτε εργασία που περιλαμβάνει αναγνώριση, ταξινόμηση και πρόβλεψη προτύπων. Χρησιμοποιούνται σε μεγάλη κλίμακα από εταιρείες όπως το Google, η Microsoft και το Facebook. Υποστηρίζουν επίσης πρωτοποριακά έργα όπως το AlphaGo, ένα έργο του Google DeepMind, του οποίου το βαθύ νευρωνικό δίκτυο ήταν το πρώτο που χτύπησε έναν ανθρώπινο επαγγελματία Go παίκτη. Παρακάτω θα αναλυθεί η λειτουργία τους, η εκπαίδευσή τους και πώς εφαρμόζονται σε προβλήματα.



Σχήμα 3.7: ANN

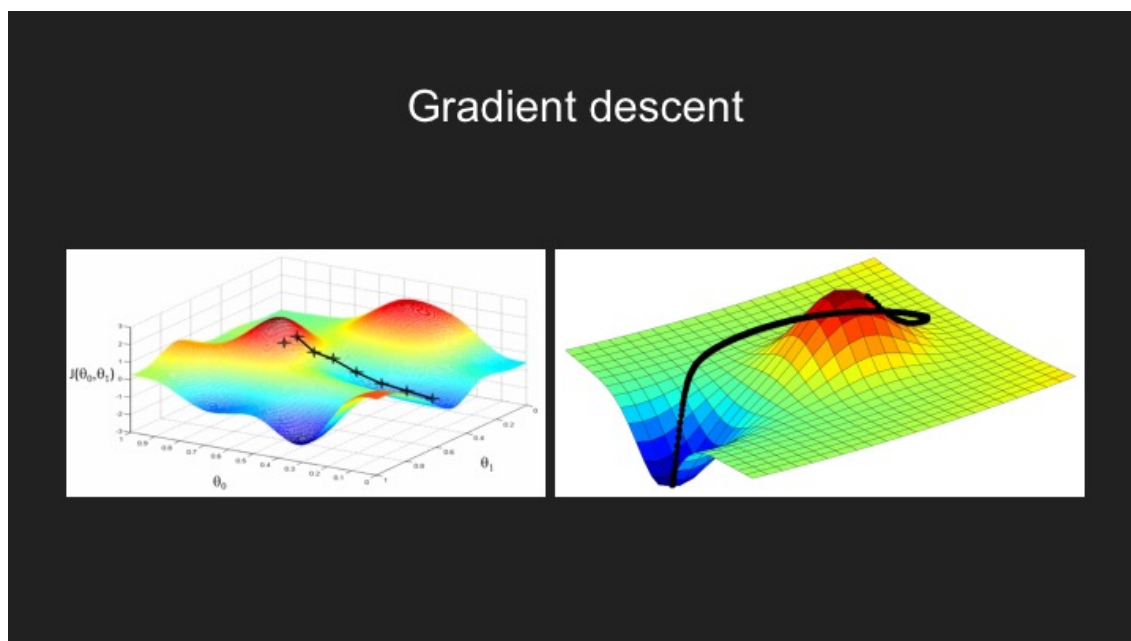
↳ **Τι είναι το Νευρωνικό Δίκτυο ;** Ένα νευρωνικό δίκτυο είναι ένα σύστημα διασυνδεδεμένων νευρώνων που μπορούν να στέλνουν σήματα μεταξύ τους. Τα πλεονεκτήματα των συνδέσεων μεταξύ αυτών των νευρώνων, γνωστά και ως βάρη, καθορίζουν τον τρόπο με τον οποίο συμπεριφέρεται το δίκτυο. Το πιο σημαντικό είναι ότι αυτά τα βάρη μπορούν να συντονιστούν συστηματικά για να κάνουν το νευρικό δίκτυο να συμπεριφέρεται με έναν πολύ συγκεκριμένο, επιθυμητό τρόπο. Για παράδειγμα, εξετάστε το πρόβλημα της δημιουργίας ενός προγράμματος που αναγνωρίζει τα χειρόγραφα ψηφία. Ένα νευρωνικό δίκτυο για αναγνώριση χειρόγραφου θα μπορούσε να αποτελείται από τρία ξεχωριστά στρώματα νευρώνων: ένα στρώμα εισόδου, ένα κρυμμένο στρώμα και ένα στρώμα εξόδου. Κάθε στρώμα νευρώνων συνδέεται πλήρως με την επόμενη. Στο πρόβλημά μας, το στρώμα εισόδου θα είναι ένα σύνολο νευρώνων που αντιπροσωπεύουν κάθε εικονοστοιχείο της εικόνας εισόδου. Είναι το «μάτι» μέσα από το οποίο το νευρικό δίκτυο βλέπει μια εικόνα, παρόμοια με τον τρόπο που κατασκευάζεται ένας ανθρώπινος αμφιβληστροειδής από πολλαπλά κύτταρα φωτουποδοχέων. Οι νευρώνες στο στρώμα εισόδου είναι ενεργοποιημένοι, ανάλογα με το χρώμα του αντίστοιχου εικονοστοιχείου. Ένα λευκό pixel, για παράδειγμα, θα μπορούσε να προκαλέσει την ενεργοποίηση του αντίστοιχου νευρώνα εισόδου. Κάθε ενεργοποιημένος νευρώνας στο στρώμα εισόδου στέλνει ένα σήμα σε όλους τους νευρώνες στο επόμενο στρώμα, το οποίο είναι το κρυμμένο στρώμα. Η ισχύς του σήματος εξαρτάται τόσο από το πόσο ενεργοποιείται ο νευρώνας όσο και από το βάρος της σύνδεσης μεταξύ των δύο νευρώνων (ένα στο στρώμα εισόδου και ένα στο κρυφό στρώμα). Κάθε νευρώνας στο κρυμμένο στρώμα προσθέτει στη συνέχεια όλα τα σήματα που έλαβε από νευρώνες στο στρώμα εισόδου. Στη συνέχεια εφαρμόζει κάτι που ονομάζεται συνάρτηση μεταφοράς ή activation function σε αυτό το άθροισμα. Παραδείγματα activation function περιλαμβάνουν τα σιγμοειδή (τα οποία «σβήνουν» την είσοδο σε κάποιο εύρος), όπως η logistic function και η υπερβολική συνάρτηση εφαπτομένης, καθώς και



Σχήμα 3.8: Νευρώνας

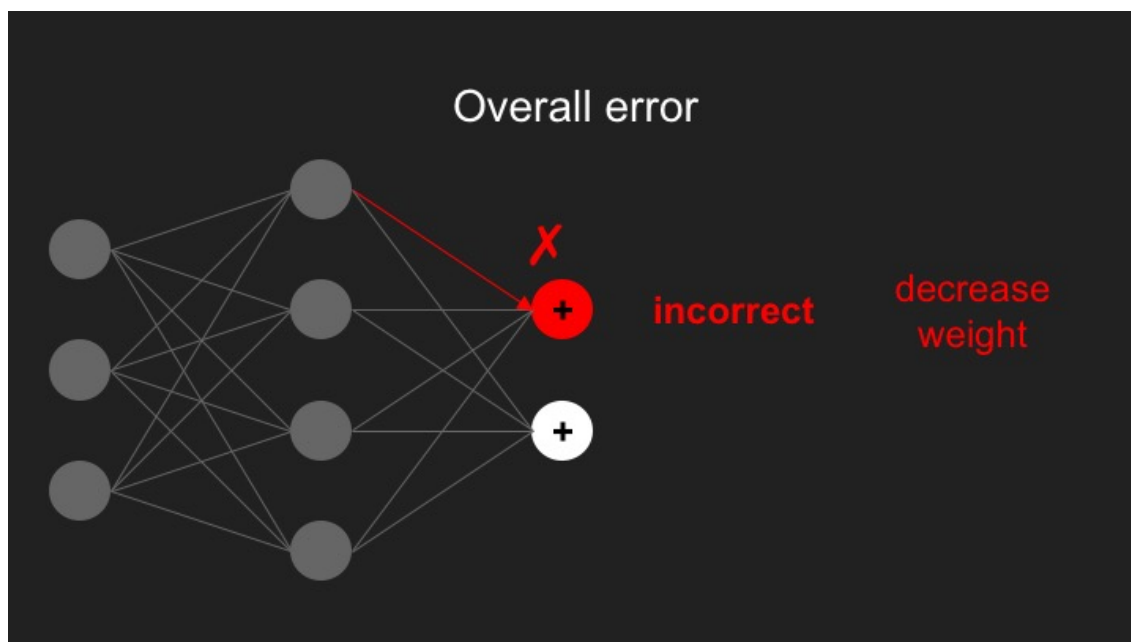
η βηματική. Αυτή η συνάρτηση ενεργοποίησης εισάγει μη γραμμικότητα στο δίκτυο και της επιτρέπει να αντιπροσωπεύει συναρτήσεις που είναι μη γραμμικές. Τα συμμετρικά σιγμοειδή όπως η υπερβολική εφαπτομένη συχνά συγκλίνουν ταχύτερα από την τυπική logistic function. Μετά από κάθε νευρώνα στο κρυφό στρώμα εφαρμόζεται μια λειτουργία ενεργοποίησης στην είσοδό του, με τη σειρά του στέλνει ένα σήμα σε όλους τους νευρώνες στο επόμενο στρώμα (το στρώμα εξόδου). Κάθε νευρώνας στο στρώμα εξόδου αντιστοιχεί σε διαφορετικό πιθανό ψηφίο (δηλ. 0, 1, 2, ..., 8, 9). Ο νευρώνας εξόδου που έχει την υψηλότερη ενεργοποίηση θα αντιπροσωπεύει ποιο ψηφίο το νευρικό δίκτυο σκέφτεται ότι περιέχει η εικόνα.

→ **Πώς εκπαιδεύεται το Νευρωνικό Δίκτυο** ; Τα νευρικά δίκτυα "εκπαιδεύονται" προσαρμόζοντας τις δυνάμεις των συνδέσεων ή των βαρών μεταξύ των νευρώνων. Ο backpropagation είναι ένας ιδιαίτερα χρήσιμος και κομψός αλγόριθμος για αυτό το σκοπό. Πρόκειται για μια ειδική περίπτωση της γενικότερης αντίληψης της κλίσης της κλίσης. Η ιδέα πίσω από την καμπύλη κλίσης είναι να βρούμε το ελάχιστο κάποιου τρόπου κάνοντας αργά "κατεβαίνοντας" προς την αντίθετη κατεύθυνση προς την κλίση. Μπορείτε να το σκεφτείτε ως έναν πεζοπόρο που κάνει το δρόμο τους κάτω από ένα βουνό ακολουθώντας την πλαγιά προς την πιο απότομη κλίση. Επί τοις ουσίας, ένα νευρωνικό δίκτυο "μαθαίνει" λίγο πολύ τον ίδιο τρόπο που ένα παιδί μαθαίνει όταν αποκομίζει μια νέα αντίληψη. Ο δάσκαλος ή ο επιβλέπων παρουσιάζει τον μαθητή σε μια σειρά δειγμάτων, ζητώντας από τον εκπαιδευόμενο να τα ταξινομήσει. Αρχικά, ο μαθητής εκτελεί αρκετά εσφαλμένα και κατατάσσει τα δείγματα περισσότερο ή λιγότερο τυχαία. Ωστόσο, καθώς ο εκπαιδευόμενος συναντά όλο και περισσότερα παραδείγματα, κάθε φορά που συγκρίνει την εικασία του με την απάντηση, γίνεται καλύτερο και καλύτερο στον εντοπισμό της σωστής απάντησης. Ο backpropagation είναι απλώς ένας τρόπος ελαχιστοποίησης της συνάρτησης απώλειας ή σφάλματος του δικτύου,



Σχήμα 3.9: Gradient Descent

διάδοσης σφαλμάτων προς τα πίσω μέσω του δικτύου και προσαρμογής των βαρών ανάλογα. Πρώτον, υπολογίζεται η διαφορά μεταξύ της πραγματικής εξόδου και της αναμενόμενης ή επιθυμητής εξόδου στην τελευταία στρώση. Ο αλγόριθμος λειτουργεί έπειτα προς τα πίσω μέσω του δικτύου, υπολογίζοντας πώς η αλλαγή ενός βάρους θα αλλάξει το συνολικό σφάλμα στο στρώμα εξόδου.



Σχήμα 3.10: Συνολικό σφάλμα

Για παράδειγμα, εάν ένα συγκεκριμένο βάρος ή σύνδεση στο δίκτυο συνέβαλε σημαντικά σε ένα μεγάλο σφάλμα, η ισχύς του θα μειωνόταν από τον αλγόριθμο. Στην επόμενη επανάληψη, η έξοδος του νευρικού δικτύου πρέπει να είναι πιο κοντά στην επιθυμητή έξοδο. Το σύνολο δεδομένων που καθορίζει τις επιθυμητές εισόδους και εξόδους ονομάζεται δεδομένα κατάρτισης ή σύνολο εκπαίδευσης. Οι αλγόριθμοι όπως ο backpropagation είναι επαναληπτικοί, πράγμα που σημαίνει ότι συγκλίνουν σε μια καλύτερη διαμόρφωση βάρους κάθε φορά που τρέχουν στο σύνολο δεδομένων.

Feed Forward Αλγόριθμος

Η ενεργοποίηση ενός νευρικού δικτύου ορίζεται διαδοχικά από :

$$y_n = f(x_n)$$

όπου το y_n είναι το διάνυσμα εισόδου στο στρώμα n , το f είναι η συνάρτηση ενεργοποίησης, το x_n είναι ο φορέας εισόδου στο στρώμα n και w_n είναι η μήτρα βάρους μεταξύ των στρωμάτων n και $n + 1$. Το πρώτο στρώμα είναι $n = 1$ και το τελευταίο στρώμα είναι $n = N$.

Ο αλγόριθμος του Backpropagation

Το σφάλμα του δικτύου ορίζεται από :

$$c = \frac{1}{2}(y_N - t)^2$$

Το σφάλμα της κλίσης στο διάνυσμα εισόδου στο στρώμα n ορίζεται ως :

$$\delta_n = \frac{\partial c}{\partial x_n}$$

Το σφάλμα της κλίσης του διανύσματος εισόδου στο τελευταίο στρώμα N είναι:

$$\begin{aligned} \delta_N &= \frac{\partial c}{\partial x_N} \\ &= \frac{\partial \frac{1}{2}(y_N - t)^2}{x_N} \\ &= \left(\frac{\partial \frac{1}{2}(y_N - t)^2}{y_N} \right) \frac{\partial y_N}{x_N} \\ &= (y_N - t) \frac{\partial f(x_N)}{x_N} \\ &= (y_N - t) f'(x_N) \end{aligned}$$

Το σφάλμα της κλίσης του διανύσματος εισόδου στο στρώμα n είναι :

$$\begin{aligned} \delta_n &= \frac{\partial c}{\partial x_n} \\ &= \frac{\partial c}{x_{n+1}} \frac{\partial x_{n+1}}{\partial x_n} \\ &= (\delta_{n+1}) \frac{\partial x_{n+1}}{\partial x_n} \\ &= (\delta_{n+1}) \frac{\partial w_n y_n}{\partial x_n} \\ &= (\delta_{n+1}) \frac{\partial w_n y_n}{y_n} \frac{\partial y_n}{x_n} \\ &= (\delta_{n+1}) \frac{\partial w_n y_n}{\partial y_n} \frac{\partial f(x_n)}{x_n} \\ &= (\delta_{n+1}) w_n f'(x_n) \end{aligned}$$

Επομένως, το σφάλμα της κλίσης του διανύσματος εισόδου σε ένα στρώμα n είναι :

$$\delta_n = f'(x_n) \begin{cases} (y_N - t), & \text{αν } n = N \\ \delta_{n+1} w_n, & \text{αν } n < N \end{cases}$$

Επομένως το σφάλμα της κλίσης του πίνακα του βάρους w_n είναι :

$$\begin{aligned} \frac{\partial c}{w_n} &= \frac{\partial c}{\partial x_{n+1}} \frac{\partial x_{n+1}}{\partial w_n} \\ &= \delta_{n+1} \frac{\partial w_n y_n}{w_n} \\ &= \delta_{n+1} y_n \end{aligned}$$

Συνεπώς η αλλαγή στο βάρος θα είναι :

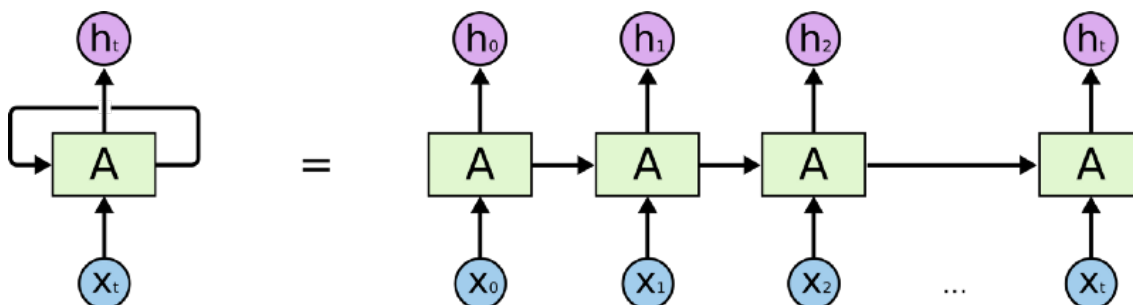
$$\begin{aligned} \Delta_{w_n} &= -a \frac{\partial c}{\partial w_n} \\ &= -a(\delta_{n+1} y_n) \end{aligned}$$

,όπου α ο ρυθμός μάθησης (ο ρυθμός του Gradient Descent). Έτσι, έχουμε δείξει τις απαραίτητες αλλαγές στο βάρος, τις οποίες ακολουθεί ένας αλγόριθμος εκπαίδευσης.

Στην δική μας εργασία κρίθηκε μετά από μελέτη και έρευνα ότι ο κατάλληλος τρόπος για να γίνει σωστή πρόβλεψη του δείκτη ΟΕΕ να γίνει η χρήση του RNN.

RNN

1. Τι είναι το RNN



Σχήμα 3.11: Σκίτσο του RNN

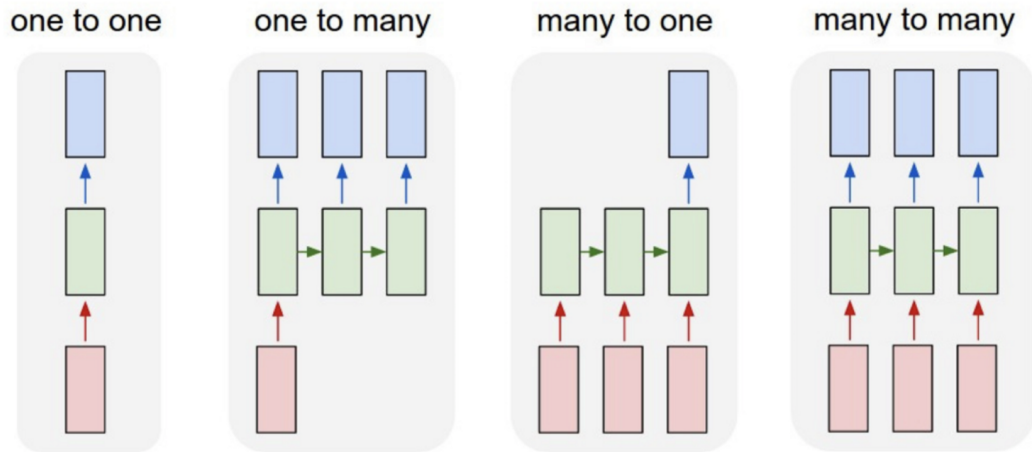
Τα επαναλαμβανόμενα νευρωνικά δίκτυα (RNN) είναι ένας ισχυρός και στιβαρός τύπος νευρωνικών δικτύων και ανήκουν στους πιο ελπιδοφόρους αλγόριθμους αυτή τη στιγμή επειδή είναι οι μόνοι με εσωτερική μνήμη. Τα RNN είναι σχετικά παλιά, όπως πολλοί άλλοι αλγόριθμοι βαθιάς μάθησης. Αρχικά δημιουργήθηκαν στη δεκαετία του '80, αλλά μπορούν να δείξουν τις πραγματικές τους δυνατότητες μετά από μερικά χρόνια, λόγω της αύξησης της διαθέσιμης υπολογιστικής ισχύος, των τεράστιων ποσοτήτων δεδομένων που έχουμε σήμερα και της εφεύρεσης του LSTM στη δεκαετία του 1990. Λόγω της εσωτερικής τους μνήμης, τα RNN είναι σε θέση να θυμούνται σημαντικά πράγματα σχετικά με την εισροή που έλαβαν, γεγονός που τους επιτρέπει να είναι πολύ ακριβή στην πρόβλεψη του τι θα ακολουθήσει.

Αυτός είναι ο λόγος για τον οποίο είναι ο προτιμώμενος αλγόριθμος για διαδοχικά δεδομένα όπως χρονολογικές σειρές, ομιλία, κείμενο, οικονομικά δεδομένα, ήχος, βίντεο, καιρικές συνθήκες και πολλά άλλα, επειδή μπορούν να διαμορφώσουν μια βαθύτερη κατανόηση μιας ακολουθίας και του πλαισίου της, σε σύγκριση με άλλα αλγόριθμους. Τα επαναλαμβανόμενα νευρωνικά δίκτυα παράγουν προγνωστικά αποτελέσματα σε διαδοχικά δεδομένα που άλλοι αλγόριθμοι δεν μπορούν.

2. Πότε το χρησιμοποιούμε

Τα χρησιμοποιούμε κυρίως όταν θέλουμε να προσεγγίσουμε ένα πρόβλημα με διαδοχικά δεδομένα. Είναι σημαντικό να καταλάβουμε ποια είναι αυτά δεδομένα. Είναι δεδομένα, τα οποία βρίσκονται μεταξύ τους σε μια αλληλουχία. Παραδείγματα είναι τα οικονομικά δεδομένα ή η ακολουθία του DNA. Ο πιο δημοφιλής τύπος διαδοχικών δεδομένων είναι ίσως δεδομένα χρονοσειρών, τα οποία είναι απλώς μια σειρά σημείων δεδομένων που παρατίθενται σε χρονολογική σειρά. εικόνα Για να γίνει όμως κατανοητή η λειτουργία του PNN θα πρέπει κανείς πρώτα να καταλάβει πώς δουλεύει ένα παραδοσιακό νευρωνικό δίκτυο (Feed-Forward). Σε ένα νευρωνικό δίκτυο Feed-Forward, οι πληροφορίες μετακινούνται μόνο προς τη μία κατεύθυνση, από το στρώμα εισόδου, μέσω των κρυφών επιπέδων (hidden layers), στο στρώμα εξόδου. Οι πληροφορίες μετακινούνται κατευθείαν μέσω του δικτύου. Εξαιτίας αυτού, οι πληροφορίες δεν αγγίζουν ποτέ έναν κόμβο

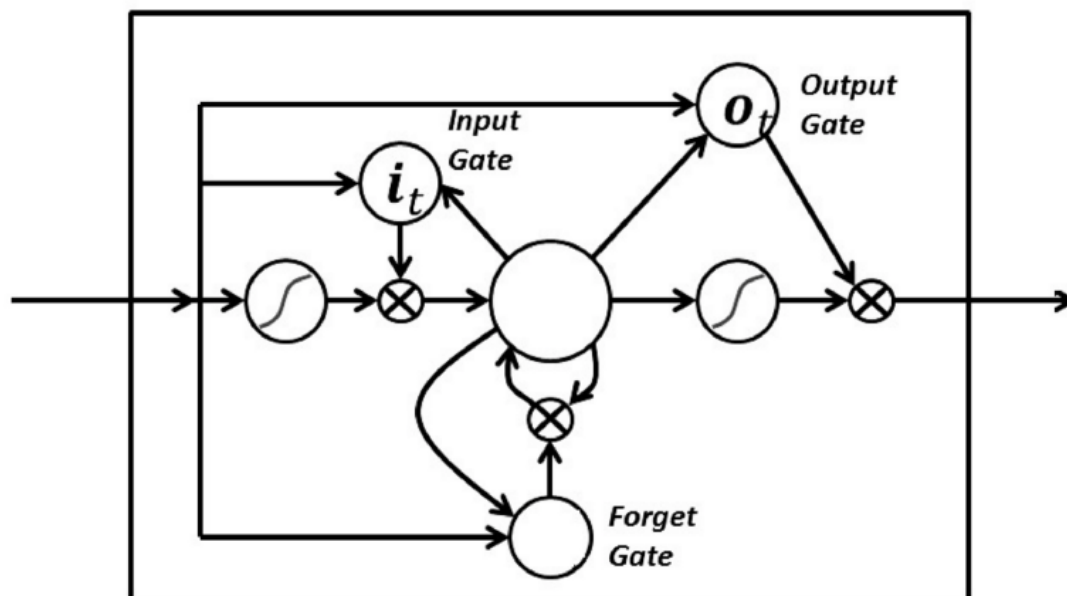
δύο φορές. Τα νευρωνικά δίκτυα προώθησης feed-forward, δεν έχουν μνήμη για την είσοδο που έλαβαν προηγουμένως και είναι επομένως κακά στην πρόβλεψη του τι θα ακολουθήσει. Επειδή ένα δίκτυο feedforward λαμβάνει υπόψη μόνο την τρέχουσα είσοδο, δεν έχει καμία ιδέα της τάξης στο χρόνο. Απλώς δεν μπορούν να θυμούνται τίποτα για το τι συνέβη στο παρελθόν, εκτός από την εκπαίδευσή τους. Σε ένα RNN, ο κύκλος πληροφοριών περνάει μέσα από ένα βρόχο. Όταν λαμβάνει μια απόφαση, λαμβάνει υπόψη την τρέχουσα είσοδο και επίσης τι έχει μάθει από τις εισόδους που έλαβε προηγουμένως. Οι δύο παρακάτω εικόνες απεικονίζουν τη διαφορά στη ροή πληροφοριών μεταξύ ενός RNN και ενός Νευρικού Δικτύου Προώθησης Feed-forward. εικόνα Ένα συνηθισμένο RNN έχει βραχυπρόθεσμη μνήμη. Σε συνδυασμό με τον LSTM έχουν επίσης μια μακροπρόθεσμη μνήμη. Ένας άλλος καλός τρόπος να απεικονιστεί η έννοια της μνήμης του RNN είναι με ένα παράδειγμα: Φανταστείτε ότι έχετε ένα κανονικό νευρωνικό δίκτυο μετάδοσης προς τα εμπρός και να του δώσετε τη λέξη "νευρώνας" ως είσοδο και επεξεργάζεται τη λέξη κατά χαρακτήρα. Τότε φτάνει στον χαρακτήρα "ρ", έχει ήδη ξεχάσει για τα "ν", "ε" και "υ", γεγονός που καθιστά σχεδόν αδύνατο για αυτόν τον τύπο νευρωνικού δικτύου να προβλέψει ποιος χαρακτήρας θα ερχόταν έπειτα. Ένα επαναλαμβανόμενο νευρωνικό δίκτυο είναι σε θέση να θυμάται ακριβώς αυτό, λόγω της εσωτερικής μνήμης του. Παράγει έξοδο, αντιγράφει αυτή την έξοδο και επανακυκλοφορεί στο δίκτυο. Επομένως ένα επαναλαμβανόμενο νευρωνικό δίκτυο έχει δύο εισόδους, το παρόν και το πρόσφατο παρελθόν. Αυτό είναι σημαντικό επειδή η ακολουθία των δεδομένων περιέχει κρίσιμες πληροφορίες για το τι θα ακολουθήσει, γι αυτό και ένα RNN μπορεί να κάνει πράγματα που άλλοι αλγόριθμοι δεν μπορούν. Ένα Νευρωνικό Δίκτυο Προώθησης Τροφοδοσίας εκχωρεί, όπως και σε όλους τους άλλους αλγορίθμους Deep Learning, ένα πίνακα βάρους στις εισόδους του και στη συνέχεια παράγει την έξοδο. Σημειώστε ότι τα RNN εφαρμόζουν βάρη στην τρέχουσα και επίσης στην προηγούμενη είσοδο. Σημειώστε επίσης ότι ενώ τα Neural Networks Feed-Forward χαρτογραφούν μία είσοδο σε μία έξοδο, τα RNNs μπορούν να αντιστοιχίσουν ένα σε πολλά, πολλά σε πολλά (μετάφραση) και πολλά σε ένα (ταξινομώντας μια φωνή).



Σχήμα 3.12: Νευρωνικό Δίκτυο

3. LSTM

Τα δίκτυα μακρινόθεσμης/βραχυπρόθεσμης μνήμης (LSTM) είναι μια επέκταση των επαναλαμβανόμενων νευρωνικών δικτύων, τα οποία επεκτείνουν κατά αυτόν τον τρόπο τη μνήμη τους. Ως εκ τούτου, είναι κατάλληλα για να εκπαιδευτούν από σημαντικές εμπειρίες οι οποίες έχουν μεγάλες χρονικές καθυστερήσεις μεταξύ τους. Οι μονάδες ενός LSTM χρησιμοποιούνται ως κατασκευαστικές μονάδες για τα στρώματα ενός RNN, το οποίο στη συνέχεια αποκαλείται συχνά δίκτυο LSTM. Τα LSTM επιτρέπουν στο RNN να θυμάται τις εισόδους του για μεγάλο χρονικό διάστημα. Αυτό συμβαίνει επειδή τα LSTM περιέχουν τις πληροφορίες τους σε μνήμη, που μοιάζει πολύ με τη μνήμη ενός υπολογιστή, επειδή το LSTM μπορεί να διαβάσει, να γράψει και να διαγράψει πληροφορίες από τη μνήμη του. Αυτή η μνήμη μπορεί να θεωρηθεί ως αποκλεισμένο κελί, όπου ο αποκλεισμός σημαίνει ότι το κελί αποφασίζει εάν πρέπει να αποθηκεύει ή να διαγράψει πληροφορίες (π.χ. εάν ανοίγει τις πύλες ή όχι), με βάση τη σημασία που αποδίδει στις πληροφορίες. Η ανάθεση της σημασίας γίνεται μέσα από τα βάρη, τα οποία επίσης μαθαίνονται από τον αλγόριθμο. Αυτό απλά σημαίνει ότι μαθαίνει με το χρόνο ποιες πληροφορίες είναι σημαντικές και ποιες όχι. Σε ένα LSTM έχετε τρεις πύλες: μια εισόδου, την πύλη για όσα θέλει αν ξεχνάει και μια εξόδου. Αυτές οι πύλες καθορίζουν αν πρέπει να αφήσει ή όχι νέα είσοδο (πύλη εισόδου), να διαγράψει τις πληροφορίες επειδή δεν είναι σημαντικές (πύλη για να "ξεχνάει") ή να την αφήσει να επηρεάσει την έξοδο στο τρέχον βήμα (πύλη εξόδου). Μπορείτε να δείτε μια απεικόνιση ενός RNN με τις τρεις πύλες του παρακάτω:

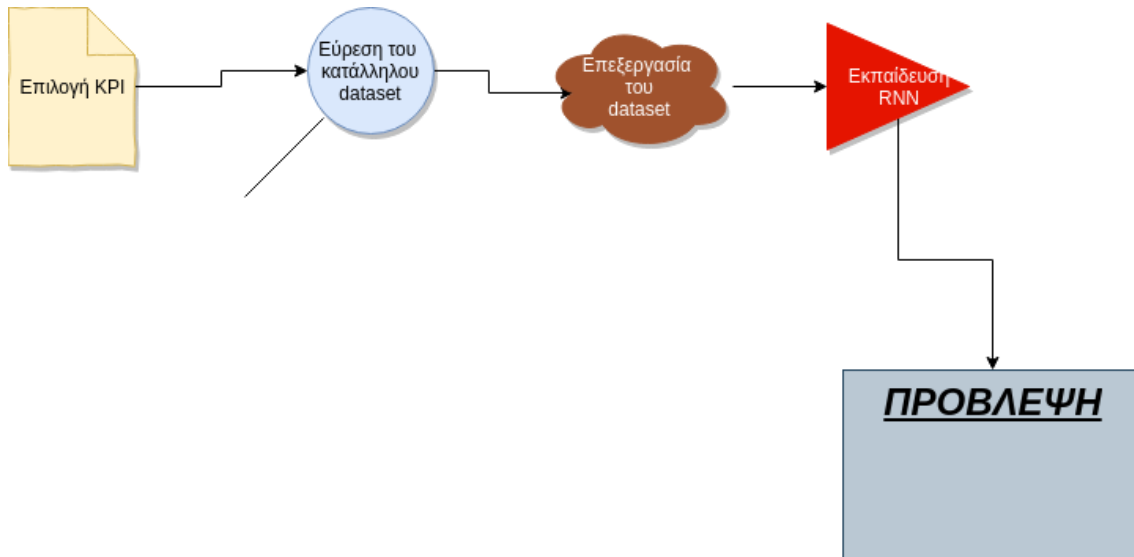


Σχήμα 3.13: Input Gate είναι η Πύλη Εισόδου, Output Gate η πύλη εξόδου και Forget Gate είναι η πύλη που του επιτρέπει να ξεχάσει.

Στο σημείο αυτό θα ασχοληθούμε με τον Επαναλαμβανόμενο Νευρωνικό Δίκτυο και πώς έχει υλοποιηθεί. Θα εμβαθύνουμε παραπάνω φυσικά στον Επαναλαμβανόμενο Νευρωνικό Δίκτυο καθώς είναι και το αντικείμενο της εργασίας.

3.3.2 Βήμα Βήμα η Δημιουργία του Νευρωνικού

Καταρχάς πρέπει στο σημείο αυτό ότι ο τρόπος που θα γίνει το πλάνο επίθεσης μας στο συγκεκριμένο θέμα έχει ως εξής :



Σχήμα 3.14: Πλάνο Επίθεσης

Στην αρχή δηλαδή πρέπει να επιλέξουμε με ποιόν KPI θα δουλέψουμε. Στην δική μας περίπτωση ως ο κατάλληλος τέθηκε ο ΟΕΕ. Στην συνέχεια το πιο βασικό κομμάτι όλων των αλγορίθμων μηχανικής μάθησης είναι η εύρεση του κατάλληλου dataset και η επεξεργασία του. Το dataset ποτέ δεν θα είναι από την αρχή στην ιδανική μορφή για εκπαίδευση. Για αυτόν τον λόγο η επεξεργασία του είναι η αρχή του παντός. Στην συνέχεια όπως και σε κάθε αλγόριθμο μηχανικής μάθησης έτσι και εδώ η μηχανή εκπαιδεύεται και στην συνέχεια μπορούμε να πάρουμε την έξοδό μας, δηλαδή την πρόβλεψη.

Συνεπώς με βάση το πλάνο που θα ακολουθήσουμε πρέπει να ορίσουμε τον KPI που θα χρησιμοποιήσουμε. Η επιλογή η δική μας όπως είπαμε και παραπάνω είναι ο δείκτης ΟΕΕ. Στο κεφάλαιο 1 έχουμε αναφέρει ποιός είναι αυτό ο δείκτης. Η επιλογή μας έγινε καταρχάς γιατί για αυτό τον δείκτη είχαμε δεδομένα από το εργοστάσιο και δεύτερον γιατί είναι ένας αρκετά έμπιστος και κατατοπιστικός δείκτης μέτρησης της αποδοτικότητας.

Όσον αφορά την επιλογή του dataset έχουμε επιλέξει να χρησιμοποιήσουμε τα δεδομένα μιας βιομηχανίας τα οποία έχουν την παρακάτω μορφή την οποία την είδαμε και στην εκτέλεση των αλγορίθμων machine learning :

Πίνακας 3.14: Data Set

<i>Date</i>	<i>Prod code</i>	<i>Shift</i>	<i>Total parts</i>	<i>Speed</i>	<i>Faults</i>	<i>Performance</i>	<i>Availability</i>	<i>Quality</i>	<i>OEE</i>
2/3/2018	98467	A	1200	70	5	70%	80%	90%	50,4%
3/3/2018	73648	B	3200	50	4	50%	80%	78%	31,2%
4/3/2018	37323	~	690	50	3	69%	80%	80%	44,4%
...

Στο τελευταίο μέρος αυτού του κεφαλαίου θα αναφερθεί ο τρόπος με τον οποίο έχουμε κατασκευάσει το RNN. Όπως είπαμε το RNN είναι ένα είδος νευρωνικού δικτύου το οποίο έχει την δυνατότητα να θυμάται τα στοιχεία πάνω στα οποία εκπαιδεύεται, το οποίο γίνεται από την φύση του και την βοήθεια του LSTM. Αφού αναλύσαμε τον τρόπο λειτουργίας αυτού του είδους νευρωνικού δικτύου, επόμενο βήμα είναι η κατανόηση του τρόπου δημιουργίας του και τέλος η εκτέλεση του και η λήψη των αποτελεσμάτων.

Επεξεργασία του Dataset

Για την εκπαίδευση κάθε αλγορίθμου machine learning χρειαζόμαστε ένα dataset, το οποίο πληροί συγκεκριμένες προδιαγραφές για κάθε αλγόριθμο. Συνεπώς και το RNN προκειμένου να εκπαιδευτεί χρειάζεται ένα ειδικά διαμορφωμένο dataset. Είναι σημαντικό στο σημείο αυτό να υπενθυμίσουμε ότι πρόκειται για ένα αλγόριθμο, ο οποίος εκπαιδεύεται με δεδομένα μια χρονική ακολουθία και με ανατροφοδότηση. Με άλλα λόγια το dataset στο οποίο εκπαιδεύεται είναι μόνο μια χρονική ακολουθία της μορφής :

Πίνακας 3.15: Αποδεκτή μορφή Data set για το RNN

<i>OEE (%)</i>
OEE (day 1)
OEE (day 2)
OEE (day 3)
OEE (day 4)
etc.

Όλοι αυτοί οι αριθμοί που βρίσκονται στα παραπάνω κελιά αναφέρονται σε τιμές μιας χρονικής ακολουθίας. Άρα αφού ένα RNN δέχεται δεδομένα της παραπάνω μορφής καλούμαστε να δημιουργήσουμε ένα τέτοιου είδους dataset. Έχουμε στην διάθεση μας δεδομένα της γραμμής παραγωγής τα οποία έχουν την μορφή του παραπάνω πίνακα. Παρακάτω βρίσκεται ένα μικρό κομμάτι από τα πραγματικά δεδομένα της γραμμής για τον μήνα Ιανουάριο του 2018 :

Πίνακας 3.16: Δεδομένα για Ιανουάριο 2018

Product Code	Hour	Start	Total Parts	Good Parts	Bad Parts	Runtime Minutes	Downtime Minutes	Faults	OEE	A	P	Q
5	19	02/01/18 (7 P.M)	1064	756	308	50.1	9.9	67	10.5	83	17.7	71.1
5	20	02/01/18	1487	1260	177	60	0	74	17.5	100	20	87.7
5	21	02/01/18	777	936	0	37.6	22.4	32	13	62	17	100
5	22	02/01/18	1431	1344	87	53.1	6.9	57	18.7	88.5	22.5	93.9
5	23	02/01/18	1453	1716	0	60	0	27	23.8	100	20.2	100
5	0	03/01/18	1393	1512	0	60	0	40	21	100	19.3	100
5	1	03/01/18	1661	1332	329	60	0	52	18.5	100	23.1	80.2
5	2	03/01/18	1326	1008	318	48.6	11.4	44	14	81.1	22.7	76
etc.												

Η πρώτη στήλη αφορά το προϊόν. Στο ολικό dataset (Ιούλιος 2017-Ιούνιος 2018) τα προϊόντα έχουν ομαδοποιηθεί βάσει κάποιων κοινών τους χαρακτηριστικών. Για παράδειγμα το προϊόν x και το προϊόν y ανήκουν στην κατηγορία 5 του παραπάνω dataset. Αυτή η ομαδοποίηση έχει πραγματοποιηθεί επειδή το πλήθος των προϊόντων που θα προέκυπτε θα ήταν αρκετά μεγάλο με αποτέλεσμα το dataset να μειωνόταν στην συνέχεια και αυτό θα είχε ως αποτέλεσμα να μην επιτυγχάνονται καλές προβλέψεις. Η επόμενη στήλη αφορά τις ώρες. Για παράδειγμα η ώρα 0 αντιστοιχεί στις 12:00 π.μ. Οι υπόλοιπες στήλες είναι τα συνολικά κομμάτια προϊόντων που παρήχθησαν, τα καλά και στην συνέχεια τα κακά κομμάτια (ελαττωματικά), το Runtime και Downtime της συγκεκριμένης ώρας, τα σφάλματα και τέλος οι δείκτες OEE, διαθεσιμότητας, απόδοσης και ποιότητας. Είναι εμφανές ότι σε ένα τέτοιο dataset δεν μπορεί εκπαιδευτεί το RNN συνεπώς καθιστάται απαραίτητη η επεξεργασία του προκειμένου να έρθει στην μορφή του πίνακα 4.3. Για να γίνει κάτι τέτοιο αρχικά πρέπει να ορίσουμε τι θέλουμε να προβλέψουμε το οποίο είναι προφανώς ο δείκτης OEE. Συνεπώς για τη

βασική εφαρμογή (η βασική εφαρμογή προβλέπει μόνο το OEE θα ακολουθήσουν πειράματα στο κεφάλαιο 5 τα οποία θα προβλέπουν και τους υπόλοιπους δείκτες) δεν χρειάζεται να εμπλακούν οι δείκτες A, P, Q και τα σφάλματα (faults). Με αυτόν τον τρόπο το dataset παίρνει την παρακάτω μορφή :

Πίνακας 3.17: Dataset

Product Code	Hour	OEE (%)
5	0	40
5	1	56
5	2	34.5
5	3	23.3
5	4	60.8
5	5	28.3
<i>etc.</i>		

Όπως έχουμε ήδη αναφέρει ο παραπάνω πίνακας είναι ένα απόσπασμα από τον Ιανουάριο του 2018. Στην διάθεση μας έχουμε δεδομένα από Ιούλιο του 2017 ως Ιούνιο του 2018. Στην συνέχεια για να εκπαιδευτεί το νευρωνικό χρειάζεται να οριστούν πόσες προηγούμενες παρατηρήσεις θα πρέπει να λαμβάνονται υπόψη προκειμένου να επιτυγχάνεται μια αξιόπιστη πρόβλεψη. Στο συγκεκριμένο RNN ο αριθμός των προηγούμενων παρατηρήσεων ή timesteps είναι 80 ή 60 ανάλογα με το πλήθος των παρατηρήσεων που έχει το dataset. Ο λόγος ο οποίος διαφέρει ο αριθμός των time steps οφείλεται στο γεγονός ότι δουλεύουμε επί της ουσίας με διαφορετικά dataset κάθε φορά. Αυτό συμβαίνει γιατί όπως έχουμε αναφέρει και παραπάνω ένα RNN για να εκπαιδευτεί χρειάζεται dataset της παρακάτω μορφής. Συνεπώς εφόσον μας επιτρέπεται για εκπαίδευση ένας πίνακας, οποίο θα είναι μία στήλη, πρέπει να βρούμε έναν τρόπο για να έχουμε dataset της επιτρεπόμενης μορφής στην διάθεση μας. Ο τρόπος που γίνεται αυτός ο χωρισμός είναι με βάση τις έξι διαφορετικές ομάδες προϊόντων που έχουμε : 1, 2, 3, 4, 5, 13

Επίσης έχουμε την στήλη *hour* η οποία δείχνει τις ώρες της παρατήρησης. Η στήλη *hour* μπορεί να πάρει τιμές : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23. Δεν μπορούμε όμως να αφήσουμε τις παρατηρήσεις με την μορφή αυτή. Επειδή ασχολούμαστε με εργοστάσιο πρέπει τις ώρες να τις χωρίσουμε σε βάρδιες. Άρα τις ώρες [8, 9, 10, 11, 12, 13, 14, 15] τις ονομαδοποιούμε ως *ShiftA*. Προσοχή όταν αναφερόμαστε στην ώρα 8 για παράδειγμα εννοούμε από τις 8:00 μέχρι τις 8:59. Αντίστοιχα χωρίζουμε σε *ShiftB* και *ShiftC*. Επομένως προκειμένου να είναι επιτρεπόμενο το dataset χρειάζεται να κάνουμε μια ομαδοποίηση. Αυτή η ομαδοποίηση γίνεται ως εξής ανά προϊόν και ανά ημερομηνία. Για παράδειγμα έχουμε τον παρακάτω πίνακα :

Πίνακας 3.18: Τελική Αποδεκτή μορφή Data set

<i>OEE (Product Code = 1 and Shift A) (%)</i>
60.5
38.6
36.4
17.4
19.5
80.4
etc.

Αυτή λοιπόν θα είναι και η τελική μορφή των dataset που θα χρησιμοποιήσουμε.

Στην συνέχεια θα ασχοληθούμε με την εκπαίδευση του RNN αφού πρώτα το επεξεργαστούμε.

Κατασκευή του RNN

Κάθε Νευρωνικό Δίκτυο κατασκευάζεται με έναν συγκεκριμένο τρόπο. Αρχικά πρέπει πρώτα να οριστούν τα στρώματα ή layers του νευρωνικού. Αυτά είναι :

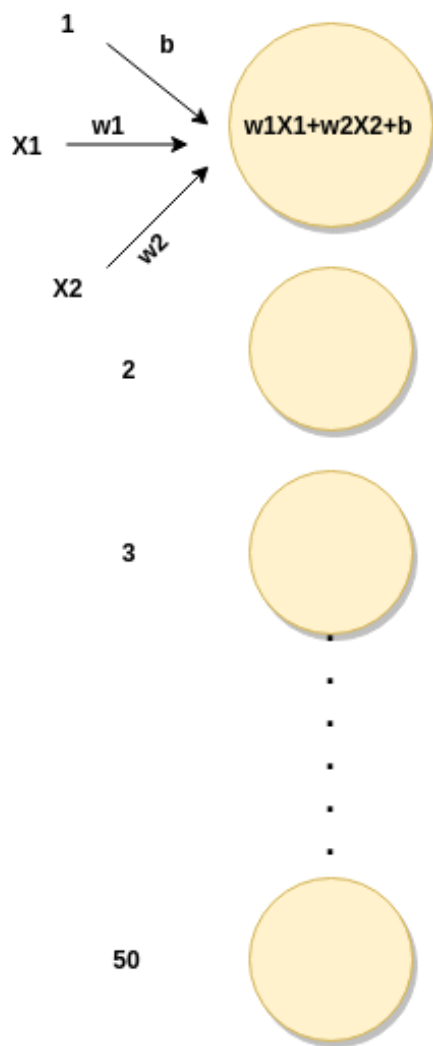
1. Input Layer
2. Hidden Layer
3. Hidden Layer
4. ...
5. Output Layer

Συνεπώς πρέπει να ορίσουμε πόσα hidden layers θα χρειαστεί να εισάγουμε. Δεν υπάρχει κάποιος κανόνας σχετικά με τον αριθμό τους και αυτό οφείλεται στο γεγονός ότι ένα τεχνητό νευρωνικό δίκτυο δεν κατασκευάζεται βάση κάποιων κανόνων αλλά με την εμπειρία. Πρόκειται λοιπόν για μια διαδικασία 'καλλιτεχνική' και 'αφηρημένη'. Το RNN που χρησιμοποιείται στην περίπτωση του εν λόγω λογισμικού αποτελείται από το input layer, τρία hidden layers και φυσικά το output layer. Όπως αναφέραμε και προηγουμένως η επιλογή πέντε στρωμάτων/layers βασίζεται κυρίως σε πειράματα που έχουν πραγματοποιηθεί. Το παρακάτω σχήμα απεικονίζει τα layers του νευρωνικού : Στο παρακάτω κεφάλαιο θα γίνουν πειράματα πρόσθε-



Σχήμα 3.15: Διάγραμμα ροής για τα στρώματα του RNN

σης ή αφαίρεσης ενός ή παραπάνω κρυφών στρωμάτων. Στην συνέχεια πρέπει να οριστεί ο αριθμός των νευρώνων που χρησιμοποιείται στο RNN. Οι νευρώνες συχνά καλούνται ή units ή nodes. Η διαδικασία επιλογής του κατάλληλου αριθμού νευρώνων είναι ακριβώς ίδια με την διαδικασία επιλογής αριθμού των layers. Στην κατασκευή του εν λόγω RNN έχουν επιλεγεί 50 νευρώνες όπως απεικονίζεται στο παρακάτω σχήμα για λόγους κατανόησης :



Σχήμα 3.16: Διάγραμμα για τους νευρώνες

Στα περισσότερα δίκτυα μάθησης, το σφάλμα υπολογίζεται ως η διαφορά μεταξύ της πραγματικής τιμής και της προβλεπόμενης. Η συνάρτηση που χρησιμοποιείται για τον υπολογισμό αυτού του σφάλματος είναι γνωστή ως Loss Function. Διαφορετικές συναρτήσεις απώλειας δίνουν διαφορετικά σφάλματα για την ίδια πρόβλεψη και έτσι θα έχουν σημαντική επίδραση στην απόδοση του μοντέλου. Μία από τις πιο ευρέως χρησιμοποιούμενες συναρτήσεις απώλειας είναι το μέσο τετραγωνικό σφάλμα, το οποίο υπολογίζει το τετράγωνο της διαφοράς μεταξύ της πραγματικής τιμής και της προβλεπόμενης τιμής. Κάθε loss function χρησιμοποιούνται για την αντιμετώπιση διαφορετικών τύπων εργασιών. Η συνάρτηση που χρησιμοποιούμε στο RNN είναι η mean squared error (μέσο τετραγωνικό σφάλμα).

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \check{y}_i)^2$$

Για ακριβείς προβλέψεις, πρέπει να ελαχιστοποιήσετε το υπολογισμένο σφάλμα. Σε ένα νευρωνικό δίκτυο, αυτό γίνεται με τη χρήση του Back Propagation. Το τρέχον σφάλμα τυπικά

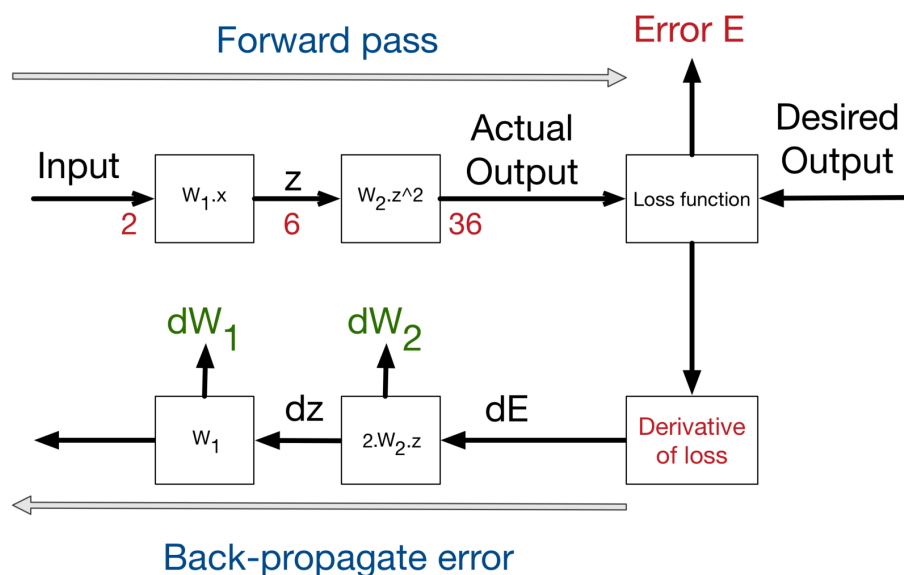
μεταδίδεται προς τα πίσω σε ένα προηγούμενο στρώμα, όπου χρησιμοποιείται για την τροποποίηση των βαρών και της απόκλισης με τέτοιο τρόπο ώστε το σφάλμα να ελαχιστοποιείται. Τα βάρη τροποποιούνται χρησιμοποιώντας μια λειτουργία που ονομάζεται Συνάρτηση βελτιστοποίησης (optimization function). Στην συνέχεια ορίζουμε ως optimizer function στο τελευταίο layer του νευρωνικού την *adam*. Ο *Adam* είναι ένας αλγόριθμος βελτιστοποίησης που μπορεί να χρησιμοποιηθεί αντί του *Stochastic Gradient Descent* για την ενημέρωση των βαρών του δικτύου με βάση τα δεδομένα εκπαίδευσης.

Εποχές (Epochs)

Μία ακόμη σημαντική παράμετρος για τον RNN είναι αυτή της εποχής ή epoch. Δεν υπάρχει κάποιος ακριβής ορισμός για το epoch. Για να γίνει όμως κατανοητό θα το ορίσουμε ως :

Ένα Epoch ορίζεται ως μία εκτέλεση του for-propagation και μία εκτέλεση του back-propagation σε όλα τα δεδομένα που έχουμε στην διάθεση μας για την εκπαίδευση του Νευρωνικού.

Για να βοηθήσουμε στην κατανόηση θα απεικονίσουμε όσο καλύτερα μπορούμε το ένα epoch :



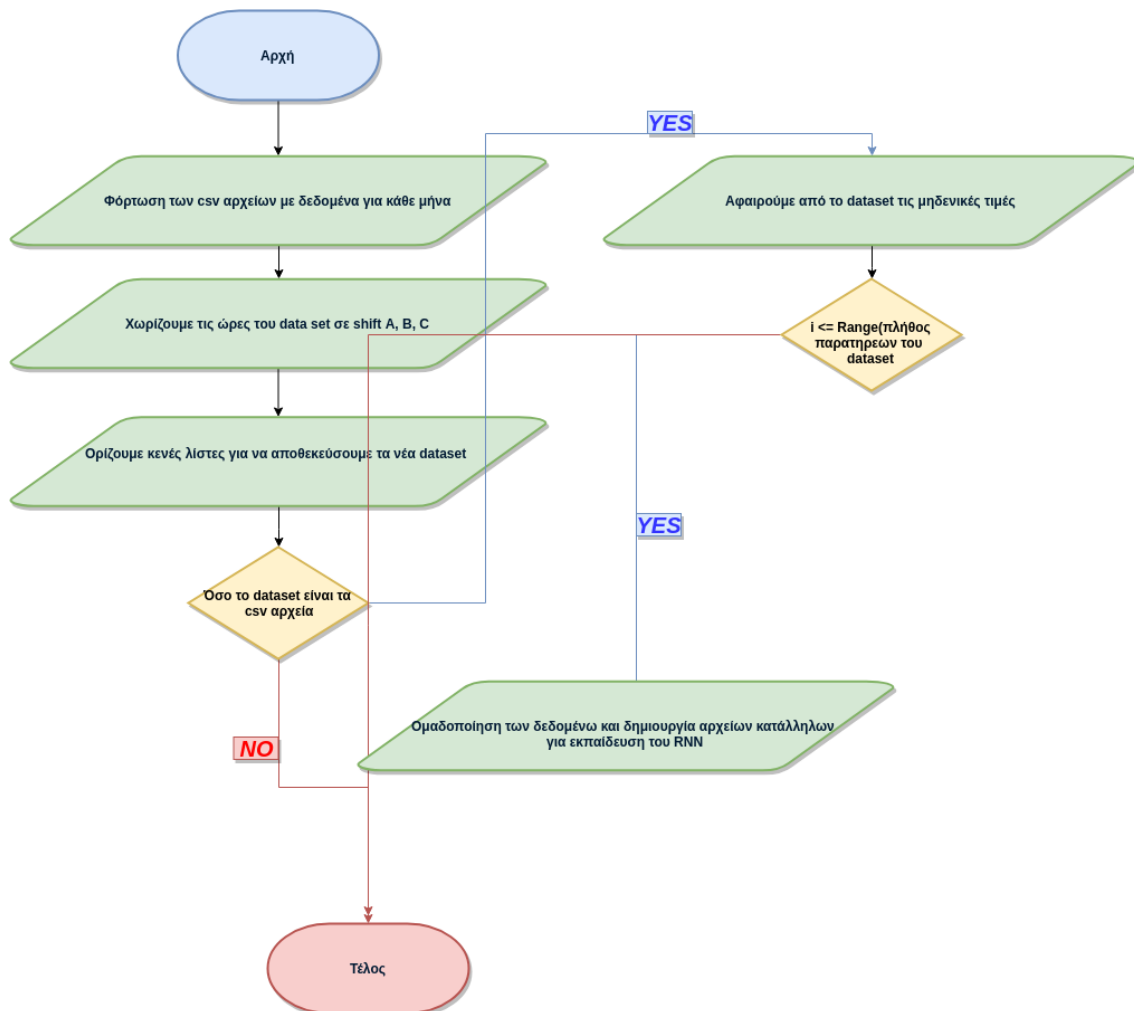
Σχήμα 3.17

Γνωρίζοντας πως λειτουργούν τα for-back περάσματα σε ένα νευρωνικό δίκτυο, ένα epoch στην ουσία είναι η εκτέλεση του σχήματος μία φορά.

Αρχικά θα παρουσιάσουμε τα διαγράμματα ροής που δημιουργήθηκαν προκειμένου να γίνει ακόμα πιο αντιληπτό ο τρόπος με τον οποίο θα εκτελέσουμε όλη την διαδικασία από την επεξεργασία των δεδομένων μέχρι και την κατασκευή του RNN και την εξαγωγή των προβλέψεων.

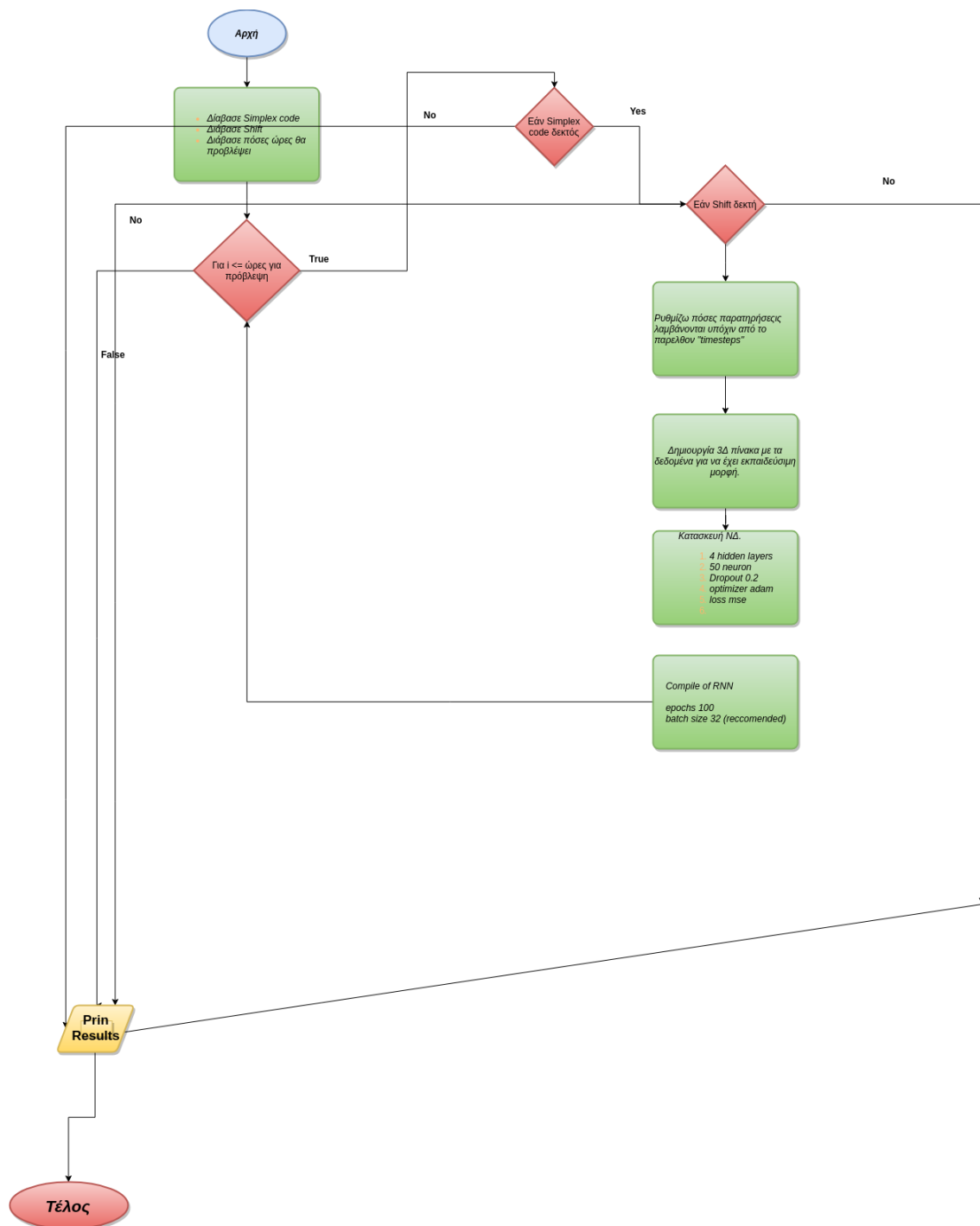
3.3.3 Διαγράμματα Ροής

Παρακάτω παρατίθεται το πλάνο 'επίθεσης' που ακολουθήθηκε προκειμένου να καταφέρουμε να δημιουργήσουμε την εφαρμογή και να λάβουμε τα αποτελέσματα. Πριν από αυτό όμως πρέπει να πούμε μερικά πράγματα για την τελική εφαρμογή. Ο κώδικας όπως αναφέραμε και παραπάνω γράφτηκε σε γλώσσα προγραμματισμού Python και συμπεριλαμβάνει δύο μεγάλα ξεχωριστά αρχεία ένα για την επεξεργασία των δεδομένων και ένα για την δημιουργία και εκτέλεση του RNN και τέλος ένα με την εκτέλεση πειραμάτων με κλασικούς αλγόριθμους Machine Learning. Αρχικά παρουσιάζεται το διάγραμμα ροής της επεξεργασίας και ομαδοποίησης των δεδομένων :



Σχήμα 3.18: Επεξεργασία Δεδομένων

Στη συνέχεια παρουσιάζεται το διάγραμμα ροής την δημιουργίας του νευρωνικού δικτύου :



Σχήμα 3.19: Κώδικας του RNN

3.4 Εκτέλεση του RNN

Αφού εξηγήθηκε περιληπτικά ο τρόπος με τον οποίο έχει γραφτεί ο κώδικας ήρθε η ώρα να περάσουμε στα αποτελέσματα των πειραμάτων. Αρχικά η εφαρμογή ήταν ικανή να κάνει προβλέψεις για την επόμενη ώρα. Ο λόγος που μπορεί να κάνει προβλέψεις για την επόμενη ώρα είναι γιατί το νευρωνικό έχει εκπαιδευτεί πάνω σε data set το οποίο αναφέρεται στον δείκτη OEE κάθε ώρα, συνεπώς δεν θα μπορούσε για παράδειγμα να προβλέψει την επόμενη μέρα ή βάρδια. Ωστόσο επειδή η μακροχρόνια πρόβλεψη είναι προφανώς πολύ χρήσιμη μέσω κάποιων αλλαγών που αφορούν κυρίως τον κώδικα κατορθώθηκε, όχι τόσο επιτυχημένα, να γίνει πρόβλεψη που αφορά μια βάρδια. Παρ' όλο που δεν ήταν και πολύ ακριβείς οι προβλέψεις στην πρώτη μας προσπάθεια, γίνεται φανερό ότι κάτι τέτοιο δεν είναι ακατόρθωτο και μπορεί στο μέλλον να γίνει και με μεγάλη επιτυχία.

3.4.1 Αποτελέσματα

Πείραμα A

Πρόβλεψη OEE για ένα και μοναδικό προϊόν και χωρίς διαχωρισμό του σε βάρδιες

Αρχικά θα παραθέσουμε τα αποτελέσματα του πειράματος χωρίς να χωρίσουμε το dataset σε κωδικούς προϊόντων (SKU code) και σε βάρδιες (Shifts). Αυτό το πείραμα πρακτικά δεν μπορεί να μας δώσει αποτελέσματα τα οποία θα θέλαμε, αν σκεφτεί κανείς ότι μια γραμμή παραγωγής δεν παράγει μόνο ένα προϊόν αλλά στην δική μας περίπτωση έξι. Επίσης η κάθε γραμμή δουλεύει σε βάρδιες και αυτές αναφέρονται κυρίως στο άτομο που χειρίζεται την μηχανή εκείνη την βάρδια. Αυτό που έχουμε κάνει προκειμένου να διεξαχθεί αυτό το πείραμα είναι να δημιουργήσουμε ένα ολικό dataset το οποίο συμπεριλαμβάνει και τον SKU code (1, 2, 3, 4, 5, 13) και τις βάρδιες (A, B, C). Στην συνέχεια εκπαιδεύεται το RNN σε όλους του OEE δείκτες από τον Ιούλιο το 17 μέχρι και τον Ιούνιο του 18. Ο λόγος για τον οποίο εκτελούμε το συγκεκριμένο πείραμα είναι για να αποδείξουμε ότι όσο περισσότερα δεδομένα έχουμε στην διάθεσή μας για να εκπαιδεύσουμε το νευρωνικό δίκτυο τόσο πιο ακριβείς θα είναι οι προβλέψεις που θα πάρουμε. Συνεπώς αυτή είναι και η ουσία εκτέλεσης ενός τέτοιου πειράματος. Παρακάτω παρουσιάζεται ο πίνακας με την πρόβλεψη για 10, 50 και 100 εποχές.

Πίνακας 3.19: OEE, Χωρίς Προϊόν και χωρίς Βάρδιες

Epochs	Predicted Value	Real Value
10	26.76	19.23
50	23.43	19.23
100	20.65	19.23

Το RNN εκπαιδεύτηκε να δίνει περισσότερο βάρος στις 120 προηγούμενες παρατηρήσεις πριν κάνει την πρόβλεψη. Παρατηρούμε ότι οι προβλέψεις είναι αρκετά κοντά στις πραγματικές τιμές και είναι λογικό με βάση τα παραπάνω που αναφέραμε.

Πείραμα B

Πρόβλεψη OEE για ένα προϊόν και τρεις βάρδιες

Στο συγκεκριμένο πείραμα εξάγουμε αποτελέσματα τα οποία αφορούν ένα προϊόν και την συμπεριφορά του στις τρεις βάρδιες. Οι ρυθμίσεις που χρειάζονται για να εκτελεστεί αυτό το πείραμα είναι οι εξής : Επεξεργαστήστε τα δεδομένα διαφορετικά, δηλαδή δεν ομαδοποιούμε το data set σε *OEE of Product Code 1 and Shift A* αλλά σε *OEE Shift A*. Επίσης ρυθμίζουμε 120 time steps back, ίδιες activation, optimization functions και loss function. Επίσης το πείραμα έγινε για δύο τιμές του Epoch 10 και 50 και 100. Τελικά έχουμε για την προβλεπόμενη τιμή και την πραγματική:

Πίνακας 3.20: *ShiftA*

Epoch	Real OEE %	Predicted OEE %
10	9.65	19.94
50	9.65	11.23
100	9.65	2.03

Πίνακας 3.21: *ShiftB*

Epoch	Real OEE %	Predicted OEE %
10	2.4	8.03
50	2.4	11.89
100	2.4	7.67

Πίνακας 3.22: *ShiftC*

Epoch	Real OEE %	Predicted OEE %
10	20.3	19.7
50	20.3	20.34
100	20.3	20.32

Πείραμα Γ

Πρόβλεψη OEE για όλα τα προϊόντα και σε 'μία' βάρδια

Στο συγκεκριμένο πείραμα εξάγουμε αποτελέσματα τα οποία αφορούν όλα τα προϊόντα σε μια μόνο βάρδια. Οι ρυθμίσεις που χρειάζονται για να εκτελεστεί αυτό το πείραμα είναι οι εξής : Επεξεργαζόμαστε τα δεδομένα διαφορετικά, δηλαδή δεν ομαδοποιούμε το dataset σε π.χ **OEE of Product Code 1 and Shift A** αλλά σε **OEE Simplex 1**. Επίσης ρυθμίζουμε 120 time steps back, ίδιες activation και optimization functions και loss function. Με την εκτέλεση του πειράματος για εποχές 10, 50, 100 τελικά έχουμε :

Πίνακας 3.23: *Product Code = 1*

Epoch	Real OEE %	Predicted OEE %
10	15.66	27.02
50	15.66	26.42
100	15.66	18.34

Πίνακας 3.24: *Product Code = 2*

Epoch	Real OEE %	Predicted OEE %
10	8.74	22.2
50	8.74	22.3
100	8.74	11.3

Πίνακας 3.25: *Product Code = 3*

Epoch	Real OEE %	Predicted OEE %
1	11	28.7
50	11	27.68
100	11	24.55

Πίνακας 3.26: *Product Code = 4*

Epoch	Real OEE %	Predicted OEE %
10	4.14	11.3
50	4.14	10.87
100	4.14	8.9

Πίνακας 3.27: *Product Code = 5*

Epoch	Real OEE %	Predicted OEE %
10	20.33	16.14
50	20.33	19.6
100	20.33	21.65

Πίνακας 3.28: *Product Code = 13*

Epoch	Real OEE %	Predicted OEE %
10	1.15	6.53
50	1.15	8.76
100	1.15	7.43

Ο λόγος για τον οποίο εκτελούνται τα **Πειράματα Β και Γ** στοχεύουν στο να αποδείξουν ότι ένα Επαναλαμβανόμενο Νευρωνικό Δίκτυο εκτός από έναν ικανό όγκο δεδομένων για εκπαίδευση χρειάζεται και τα δεδομένα αυτά να είναι κατά κάποιον τρόπο συνδεδεμένα το ένα με το άλλο. Σε αυτά τα πειράματα λοιπόν αποδεικνύεται ότι για να εξάγουμε καλές προβλέψεις αφενός θα πρέπει να έχουμε ποσότητα δεδομένων αφετέρου και ποιότητα.

3.4.2 Σχολιασμός των αποτελεσμάτων των Πειραμάτων Α, Β και Γ

Από τα παραπάνω πειράματα είναι προφανές ότι δεν μπορούμε να εξάγουμε αποτελέσματα τα οποία είναι κατάλληλα για την χρήση που θέλουμε να κάνουμε στη εφαρμογή. Ωστόσο μπορούμε να εξάγουμε κάποια πολύ σημαντικά συμπεράσματα όπως :

1. Τα Νευρωνικά Δίκτυα προκειμένου να δώσουν πετυχημένες προβλέψεις χρειάζονται πληθώρα δεδομένων.
2. Ένα Νευρωνικό Δίκτυο είναι ένα εργαλείο του οποίου η κατασκευή είναι τελείως υποκειμενική και αφηρημένη, με αποτέλεσμα να είναι εξ ολοκλήρου στο χέρι του προγραμματιστή να βρει την χρυσή τομή, στην κατασκευή, ούτως ώστε να εξάγει τα καλύτερα δυνατά αποτελέσματα.

3.4.3 Αποτελέσματα Τελικής Εφαρμογής για τον δείκτη OEE

Σε αυτό το σημείο θα παρουσιαστούν τα αποτελέσματα του τελικού λογισμικού. Το πρώτο αρχείο όπως έχουμε ήδη αναφέρει αφορά τον κώδικα που έχουμε δημιουργήσει προκειμένου να γίνεται η επεξεργασία και η ομαδοποίηση των δεδομένων μας. Αρχικά δεν έχουμε κρατήσεις τις παρατηρήσεις οι οποίες είναι διάφορες του μηδενός, γιατί αυτό σημαίνει ότι η μηχανή δεν λειτουργούσε οπότε δεν έχει νόημα να ληφθεί υπόψη στην εκπαίδευση. Στη συνέχεια χωρίσαμε το dataset όπως έχουμε αναφέρει παραπάνω. Τέλος ο τρόπος που έχει κατασκευαστεί το RNN αναφέρεται στο τέταρτο κεφάλαιο. Ο κώδικας εκτελείται για *Epoch* 10, 50, 100 και λαμβάνουμε τα παρακάτω αποτελέσματα :

Πίνακας 3.29: *Shift A ProductCode = 1*

Epoch	Real OEE%	Predicted OEE%
10	15.66	37.6568
50	15.66	28.83
100	15.66	23.54

Πίνακας 3.30: *Shift A ProductCode = 2*

Epoch	Real OEE%	Predicted OEE%
10	1	11.3
50	1	22.2
100	1	12.76

Πίνακας 3.31: *Shift A ProductCode = 3*

Epoch	Real OEE%	Predicted OEE%
10	11	6.01
50	11	24.73
100	11	21.9297

Πίνακας 3.32: *Shift A ProductCode = 4*

Epoch	Real OEE%	Predicted OEE%
10	2.75	23.34
50	2.75	17.3
100	2.75	12.3

Πίνακας 3.33: *Shift A ProductCode = 5*

Epoch	Real OEE%	Predicted OEE%
10	10	3.6
50	10	19.53
100	10	18.04

Πίνακας 3.34: *Shift A ProductCode = 13*

Epoch	Real OEE%	Predicted OEE%
10	2	23.3
50	2	18.3
100	2	8.43

Πίνακας 3.35: *Shift B ProductCode = 1*

Epoch	Real OEE%	Predicted OEE%
10	39.6	24.34
50	39.6	27.4
100	39.6	34,8

Πίνακας 3.36: *Shift B ProductCode = 2*

Epoch	Real OEE%	Predicted OEE%
1	30.66	3.01
50	30.66	20.82
100	30.66	20.99

Πίνακας 3.37: *Shift B ProductCode = 3*

Epoch	Real OEE%	Predicted OEE%
10	35.33	3.51
50	35.33	27.56
100	35.33	28.82

Πίνακας 3.38: *Shift B ProductCode = 4*

Epoch	Real OEE%	Predicted OEE%
10	7.16	15.4
50	7.16	18.6
100	7.16	16.3

Πίνακας 3.39: *Shift B ProductCode = 5*

Epoch	Real OEE%	Predicted OEE%
10	28.66	2.8
50	28.66	21.8
100	28.66	17.38

Πίνακας 3.40: *Shift B ProductCode = 13*

Epoch	Real OEE%	Predicted OEE%
10	8.66	12.3
50	8.86	11.45
100	8.86	8,9

Πίνακας 3.41: *Shift C ProductCode = 1*

Epoch	Real OEE%	Predicted OEE%
1	26	15.33
50	26	27.05
100	26	24.6

Πίνακας 3.42: *Shift C ProductCode = 2*

Epoch	Real OEE%	Predicted OEE%
10	9	1.33
50	9	2.06
100	9	9.67

Πίνακας 3.43: *Shift C ProductCode = 3*

Epoch	Real OEE%	Predicted OEE%
10	18.99	4.02
50	18.99	30.67
100	18.99	28.8

Πίνακας 3.44: *Shift C ProductCode = 4*

Epoch	Real OEE%	Predicted OEE%
10	28.66	6.78
50	28.66	21.81
100	28.66	21.89

Πίνακας 3.45: *Shift C ProductCode = 5*

Epoch	Real OEE%	Predicted OEE%
10	20.33	1.82
50	20.33	20.49
100	20.33	18.25

Πίνακας 3.46: *Shift C ProductCode = 13*

Epoch	Real OEE%	Predicted OEE%
10	10.5	35.5
50	10.5	30.34
100	10.5	29.44

3.4.4 Αποτέλεσμα Τελικού Λογισμικού για τον αριθμό των σφαλμάτων

Το συγκεκριμένο πείραμα γίνεται για να διαπιστώσουμε αν μπορούμε να προβλέψουμε σωστά τον αριθμό των σφαλμάτων της μηχανής. Κάτι τέτοιο μπορεί να μας βοηθήσει αρκετά για να αντιμετωπίσουμε προβλήματα που δημιουργούνται στην μηχανή. Το συγκεκριμένο πείραμα εκτελέστηκε για epoch = 10, 4 Hidden Layers, 50 νευρώνες και τις γνωστές συναρτήσεις απώλειας και ενεργοποίησης. Λαμβάνουμε τα παρακάτω αποτελέσματα :

Πίνακας 3.47: *Shift A ProductCode = 1*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
0	2

Πίνακας 3.48: *Shift A ProductCode = 2*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
18	21

Πίνακας 3.49: *Shift A ProductCode = 3*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
50	17

Πίνακας 3.50: *Shift A ProductCode = 4*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
0	23

Πίνακας 3.51: *Shift A ProductCode = 5*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
86	17

Πίνακας 3.52: *Shift A ProductCode = 13*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
1	7

Πίνακας 3.53: *Shift B ProductCode = 1*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
0	21

Πίνακας 3.54: *Shift B ProductCode = 2*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
0	15

Πίνακας 3.55: *Shift B ProductCode = 3*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
5	14

Πίνακας 3.56: *Shift B ProductCode = 4*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
15	19

Πίνακας 3.57: *Shift B ProductCode = 5*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
32	8

Πίνακας 3.58: *Shift B ProductCode = 13*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
15	11

Πίνακας 3.59: *Shift C ProductCode = 1*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
8	12

Πίνακας 3.60: *Shift C ProductCode = 2*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
0	8

Πίνακας 3.61: *Shift C ProductCode = 3*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
0	1

Πίνακας 3.62: *Shift C ProductCode = 4*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
47	30

Πίνακας 3.63: *Shift C ProductCode = 5*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
9	13

Πίνακας 3.64: *Shift C ProductCode = 13*

<i>Real Quantity of Faults</i>	<i>Predicted Quantity of Faults</i>
0	7

Παρατηρούμε ότι οι προβλέψεις στον αριθμό των σφαλμάτων είχαν κάποια απόκλιση από τις πραγματικές. Ωστόσο πρέπει να επισημάνουμε ότι οι παρατηρήσεις για τις οποίες δεν είχαμε κανένα σφάλμα ίσως επηρεάζουν την εκπαίδευση του νευρωνικού. Επίσης το ίδιο το νευρωνικό θα μπορούσε να δημιουργηθεί διαφορετικά για ένα τέτοιο πείραμα. Ένα άλλο

βασικό στοιχείο είναι ότι εκπαιδεύσαμε το νευρωνικό για σχετικά μικρό αριθμό epochs αφού έχουμε παρατηρήσει και στα προηγούμενα πειράματα ότι όσο αυξάνεται ο αριθμός των epochs τόσο πιο ακριβής είναι η πρόβλεψή μας. Τέλος στην περίπτωση που θα θέλαμε να αναπτύξουμε περισσότερο ένα τέτοιο πείραμα θα ήταν σωστό να μην πειραματιστούμε μόνο με ένα νευρωνικό δίκτυο αλλά και με κλασικούς αλγορίθμους μηχανικής μάθησης τύπου Regression.

3.4.5 Σχολιασμός των Αποτελεσμάτων της τελικής εφαρμογής

Αξίζει να σημειωθεί το γεγονός ότι επειδή το ολικό dataset ομαδοποιήθηκε σε πολλά μικρά διαφορετικά dataset για να γίνει η ορθή εκπαίδευση του RNN κάποια από τα αποτελέσματα δεν πλησίαζαν τόσο πολύ στις πραγματικές τιμές. Αυτό συμβαίνει γιατί π.χ για την Shift B στο προϊόν 13 δεν υπήρχαν αρκετές παρατηρήσεις για να εκπαιδευτεί καλύτερα.

Στο αρχικό dataset το οποίο εργαζόμαστε έχουμε και μια στήλη με τον αριθμό των σφαλμάτων της κάθε ομάδας προϊόντος για κάθε βάρδια. Αυτό σημαίνει ότι μπορούμε να προβλέψουμε και τον αριθμό των σφαλμάτων με τον ίδιο τρόπο αλλάζοντας απλά την στήλη των δεδομένων που εκπαιδεύεται από OEE σε Faults.

Συνοψίζοντας θα λέγαμε ότι οι προβλέψεις/αποτελέσματα που εξάγαμε είναι αρκετά θετικά παρατηρώντας ότι σε πολύ λίγες περιπτώσεις οι αποκλίσεις ήταν πολύ μεγάλες. Υπάρχει φυσικά η δυνατότητα να προσπαθήσουμε να διορθώσουμε αυτές τις τιμές δοκιμάζοντας να επεξεργαστούμε το ίδιο το RNN και να πειραματιστούμε με παραμέτρους όπως είναι οι Νευρώνες, τα επίπεδα (Layers) και τα timesteps που εκπαιδεύεται. Ακόμα μπορούμε να αλλάξουμε την loss και optimization function.

3.4.6 Επεξεργασία του Νευρωνικού

Σαν τελευταίο πείραμα επιχειρήσαμε να επεξεργαστούμε το Νευρωνικό Δίκτυο ως προς τους νευρώνες του δικτύου. Το πείραμα αυτό έγινε αλλάζοντας τους Νευρώνες από 50 σε 10 και 100 και κρατώντας τα κρυμμένα στρώματά του νευρωνικού ως έχει. Ο λόγος που γίνεται αυτό το πείραμα είναι για να παρατηρήσουμε αν μπορεί να είναι αποδοτικότερο το Νευρωνικό με λιγότερους ή περισσότερους Νευρώνες. :

Πείραμα Δ

Αλλαγή του αριθμού των Νευρώνων

Πίνακας 3.65: *OEE % Shift A* (Prediction/Real)

<i>Product Code</i>	<i>10 Neurons</i>	<i>100 Neurons</i>
1	26.98/15.66	25.19/15.66
2	13.6/1	13.66/1
3	23.95/11	22.23/11
4	17.79/2.75	22.4/2.75
5	17.52/10	18.34/10
13	19.49/2	6.02/2

Πίνακας 3.66: *OEE % Shift B* (Prediction/Real)

<i>Product Code</i>	<i>10 Neurons</i>	<i>100 Neurons</i>
1	26.31/39.6	23.28/39.6
2	5.7/30.66	28.3/30.66
3	28.73/46.9	28.68/46.9
4	23.8/7.16	15.2/7.16
5	22.11/28.33	22.5/28.33
13	23.92/8.66	21.92/8.66

Πίνακας 3.67: *OEE % Shift C* (Prediction/Real)

<i>Product Code</i>	<i>10 Neurons</i>	<i>100 Neurons</i>
1	26.079/33.66	27.17/33.66
2	19.05/9	17.7/9
3	29.22/6.33	30.23/6.33
4	18.2/28.66	20.3/28.66
5	19.18/11.66	19.92/11.66
13	29.89/10.05	31.36/10.05

Κεφάλαιο 4

Αποτελέσματα και Σχολιασμός

Στο κεφάλαιο αυτό θα παραθέσουμε συγκεντρωτικά όλα τα αποτελέσματα τα οποία λήφθηκαν από την εκτέλεση των πειραμάτων στο τρίτο κεφάλαιο και θα ακολουθήσει ένας σχολιασμός των αποτελεσμάτων αυτών. Αυτή η διαδικασία γίνεται προκειμένου να αποφανθεί τελικά ποιός είναι ο κατάλληλος αλγόριθμος για μια τέτοια εργασία.

4.1 Πρόβλεψη με κλασικούς αλγόριθμους Machine Learning

4.1.1 Linear Regression

Πίνακας 4.1: Shift A, Linear Regression, OEE

Product Code	Predicted Value%	Real Value%
1	9.28	15.66
2	9.33	1
3	9.56	11
4	18.4	2.75
5	9.83	10
13	11.2	2

Πίνακας 4.2: Shift B, Linear Regression, OEE

Product Code	Predicted Value%	Real Value%
1	9.46	39.6
2	9.6	30.6
3	10.7	35.33
4	12.33	7.66
5	10.01	28.66
13	22	8.16

Πίνακας 4.3: Shift C, Linear Regression, OEE

Product Code	Predicted Value%	Real Value%
1	9.11	26
2	9.21	9
3	9.38	18.99
4	9.52	26.86
5	9.65	20.33
13	10.74	10.05

4.1.2 Decision Tree

Πίνακας 4.4: Shift A, Decision Tree, OEE

Product Code	Predicted Value%	Real Value%
1	11.02	15.66
2	11.4	1
3	14.23	11
4	13.2	2.75
5	8.05	10
13	6.45	2

Πίνακας 4.5: Shift B, Decision Tree, OEE

Product Code	Predicted Value%	Real Value%
1	11.85	39.6
2	9.05	30.6
3	14.32	35.33
4	6.7	7.66
5	6.08	28.66
13	4.45	8.16

Πίνακας 4.6: Shift C, Decision Tree, OEE

Product Code	Predicted Value%	Real Value%
1	9.76	26
2	10.8	9
3	13.53	18.99
4	6.66	26.86
5	6.99	20.33
13	12.2	10.05

4.1.3 Random Forest

Πίνακας 4.7: Shift A, Random Forest, OEE

Product Code	Predicted Value%	Real Value%
1	10.88	15.66
2	11	1
3	10.92	11
4	8.45	2.75
5	6.56	10
13	7.55	2

Πίνακας 4.8: Shift B, Random Forest, OEE

Product Code	Predicted Value%	Real Value%
1	10.88	39.6
2	10.82	30.6
3	10.92	35.33
4	10.99	7.66
5	6.56	28.66
13	6.75	8.16

Πίνακας 4.9: Shift C, Random Forest, OEE

Product Code	Predicted Value%	Real Value%
1	10.88	26
2	11.2	9
3	10.92	18.99
4	6.56	26.86
5	6.56	20.33
13	12.61	10.05

4.1.4 Gradient Boosting Tree

Πίνακας 4.10: Shift A, Gradient Boosting Tree, OEE

Product Code	Predicted Value%	Real Value%
1	36.57	15.66
2	40.3	1
3	35.74	11
4	18.1	2.75
5	27.06	10
13	11.33	2

Πίνακας 4.11: Shift B, Gradient Boosting Tree, OEE

Product Code	Predicted Value%	Real Value%
1	36.68	39.6
2	27.73	30.6
3	35.96	35.33
4	35.22	7.66
5	26.12	28.66
13	22.9	8.16

Πίνακας 4.12: Shift C, Gradient Boosting Tree, OEE

Product Code	Predicted Value%	Real Value%
1	34.01	26
2	32.22	9
3	33.96	18.99
4	25.82	26.86
5	24.76	20.33
13	38.54	10.05

4.2 Πρόβλεψη RNN

Πίνακας 4.13: *Shift A ProductCode = 1*

Epoch	Real OEE%	Predicted OEE%
10	15.66	37.6568
50	15.66	28.83
100	15.66	23.54

Πίνακας 4.14: *Shift A ProductCode = 2*

Epoch	Real OEE%	Predicted OEE%
10	1	11.3
50	1	22.2
100	1	12.76

Πίνακας 4.15: *Shift A ProductCode = 3*

Epoch	Real OEE%	Predicted OEE%
10	11	6.01
50	11	24.73
100	11	21.9297

Πίνακας 4.16: *Shift A ProductCode = 4*

Epoch	Real OEE%	Predicted OEE%
10	2.75	23.34
50	2.75	17.3
100	2.75	12.3

Πίνακας 4.17: *Shift A ProductCode = 5*

Epoch	Real OEE%	Predicted OEE%
10	10	3.6
50	10	19.53
100	10	18.04

Πίνακας 4.18: *Shift A ProductCode = 13*

Epoch	Real OEE%	Predicted OEE%
10	2	23.3
50	2	18.3
100	2	8.43

Πίνακας 4.19: *Shift B ProductCode = 1*

Epoch	Real OEE%	Predicted OEE%
10	39.6	24.34
50	39.6	27.4
100	39.6	34,8

Πίνακας 4.20: *Shift B ProductCode = 2*

Epoch	Real OEE%	Predicted OEE%
1	30.66	3.01
50	30.66	20.82
100	30.66	20.99

Πίνακας 4.21: *Shift B ProductCode = 3*

Epoch	Real OEE%	Predicted OEE%
10	35.33	3.51
50	35.33	27.56
100	35.33	28.82

Πίνακας 4.22: *Shift B ProductCode = 4*

Epoch	Real OEE%	Predicted OEE%
10	7.16	15.4
50	7.16	18.6
100	7.16	16.3

Πίνακας 4.23: *Shift B ProductCode = 5*

Epoch	Real OEE%	Predicted OEE%
10	28.66	2.8
50	28.66	21.8
100	28.66	17.38

Πίνακας 4.24: *Shift B ProductCode = 13*

Epoch	Real OEE%	Predicted OEE%
10	8.66	12.3
50	8.86	11.45
100	8.86	8,9

Πίνακας 4.25: *Shift C ProductCode = 1*

Epoch	Real OEE%	Predicted OEE%
1	26	15.33
50	26	27.05
100	26	24.6

Πίνακας 4.26: *Shift C ProductCode = 2*

Epoch	Real OEE%	Predicted OEE%
10	9	1.33
50	9	2.06
100	9	9.67

Πίνακας 4.27: *Shift C ProductCode = 3*

Epoch	Real OEE%	Predicted OEE%
10	18.99	4.02
50	18.99	30.67
100	18.99	28.8

Πίνακας 4.28: *Shift C ProductCode = 4*

Epoch	Real OEE%	Predicted OEE%
10	28.66	6.78
50	28.66	21.81
100	28.66	21.89

Πίνακας 4.29: *Shift C ProductCode = 5*

Epoch	Real OEE%	Predicted OEE%
10	20.33	1.82
50	20.33	20.49
100	20.33	18.25

Πίνακας 4.30: *Shift C ProductCode = 13*

Epoch	Real OEE%	Predicted OEE%
10	10.5	35.5
50	10.5	30.34
100	10.5	29.44

4.3 Σχολιασμός των αποτελεσμάτων

Συνοψίζοντας αξίζει να σημειωθεί ότι στο συγκεκριμένο κεφάλαιο παραθέσαμε αποτελέσματα που αφορούν την πρόβλεψη του OEE κάποια ώρα. Αυτή η ώρα δεν είναι άλλη από την επόμενη από αυτή που έχει εκπαιδευτεί το Νευρωνικό. Δηλαδή αν το Νευρωνικό είχε

εκπαιδευτεί μέχρι την 13η ώρα θα προβλέψει την 14η. Άρα επί τοις ουσίας το Νευρωνικό εκπαιδευεται σε όλο το dataset εκτός από την τελευταία παρατήρηση.

Στην συνέχεια, πρέπει να επισημάνουμε ότι στους κλασικούς αλγορίθμους Machine Learning δεν υπάρχει η παράμετρος epoch, είναι μια μοναδική παράμετρος που έχουν μόνο τα Νευρωνικά Δίκτυα και ρυθμίζεται κατά την δημιουργία και εκτέλεση του Νευρωνικού.

Έχοντας αναφέρει τα παραπάνω θα σχολιάσουμε τα αποτελέσματα. Καταρχήν από όλα τα πειράματα που έγιναν με RNN αναφέρονται μόνο αυτά που έγιναν με την τελική εφαρμογή. Αυτό γίνεται γιατί μόνο σε αυτή την περίπτωση μελετάμε όμοια πράγματα. Παρατηρούμε ότι τα αποτελέσματα που πήραμε από την εκτέλεση του RNN είναι πολύ καλύτερα από τα αποτελέσματα που λάβαμε εκτελώντας τα ίδια με κλασικούς αλγορίθμους machine learning.

Παρακάτω παρατίθεται ο πίνακας με τα σφάλματα που προέκυψαν στην εκτέλεση της εφαρμογής χρησιμοποιώντας το RNN και τους κλασικούς αλγορίθμους Machine Learning προκειμένου να διαπιστώσουμε ότι όντως το Νευρωνικό είναι πιο αποδοτικό από τους κλασικούς αλγορίθμους Machine Learning. Ο τύπος του σφάλματος που χρησιμοποιήθηκε είναι η εκατοστιαία διαφορά, δηλαδή :

$$\frac{|X_{predicted} - X_{real}|}{|X_{real}|} \times 100\%$$

Πίνακας 4.31: Σφάλματα (%) για Epoch = 10 και Shift = A

Product Code	RNN	Linear Regrssion	Decision Tree	Random Forest	Gradient Boosting Tree
1	140,42	40,74	29,62	30,52	133,52
2	1030	833	1040	1000	3930
3	45,36	13,1	29,36	0,72	224,91
4	748,72	569	380	207,27	558,18
5	64	17	19,5	34,4	170,6
13	1065	460	222,5	277,5	465

Πίνακας 4.32: Σφάλματα (%) για Epoch = 10 και Shift = B

Product Code	RNN	Linear Regrssion	Decision Tree	Random Forest	Gradient Boosting Tree
1	38,53	76,11	70,07	72,52	7,37
2	90,16	68,62	70,42	64,64	9,37
3	90,06	69,71	59,46	69,09	1,78
4	413,33	60,96	12,53	43,47	359,79
5	90,23	65,07	78,78	77,11	8,86
13	50,73	169,60	45,46	17,15	180,63

Πίνακας 4.33: Σφάλματα (%) για Epoch = 10 και Shift = C

Product Code	RNN	Linear Regrssion	Decision Tree	Random Forest	Gradient Boosting Tree
1	41,03	64,96	62,46	58,15	30,80
2	85,22	2,33	20	24,44	258
3	78,83	50,60	28,75	42,49	78,83
4	74,75	64,55	75,20	75,57	3,87
5	91,04	52,53	65,61	67,73	21,79
13	253,23	6,86	21,39	25,47	283,48

Πίνακας 4.34: Σφάλματα (%) για Epoch = 50 και Shift = A

Product Code	RNN	Linear Regrssion	Decision Tree	Random Forest	Gradient Boosting Tree
1	140,42	40,74	29,62	30,52	133,52
2	1030	833	1040	1000	3930
3	45,36	13,1	29,36	0,72	224,99
4	748,72	569	380	207,27	558,18
5	64	17	19,5	34,4	170,6
13	1065	460	222,5	277,5	465

Πίνακας 4.35: Σφάλματα (%) για Epoch = 50 και Shift = B

Product Code	RNN	Linear Regrssion	Decision Tree	Random Forest	Gradient Boosting Tree
1	30,80	76,11	70,07	72,52	7,37
2	31,96	68,62	70,42	64,64	9,37
3	21,99	69,71	59,46	69,09	1,78
4	520	60,96	12,53	43,47	359,79
5	23,93	65,07	78,78	77,11	8,86
13	40,31	169,60	45,46	17,15	180,63

Πίνακας 4.36: Σφάλματα (%) για Epoch = 50 και Shift = C

Product Code	RNN	Linear Regrssion	Decision Tree	Random Forest	Gradient Boosting Tree
1	4,03	64,96	62,46	58,15	30,84
2	77,11	2,33	20	24,44	258
3	61,52	50,60	28,75	42,49	78,83
4	18,80	64,55	75,21	75,57	3,87
5	0,78	52,53	65,61	67,73	21,79
13	201,89	6,86	21,39	25,47	283,48

Πίνακας 4.37: Σφάλματα (%) για Epoch = 100 και Shift = A

Product Code	RNN	Linear Regrssion	Decision Tree	Random Forest	Gradient Boosting Tree
1	50,31	40,74	29,62	30,52	133,52
2	1176	833	1040	1000	3930
3	99,09	13,1	29,36	0,72	224,91
4	347,27	569	380	207,27	558,18
5	80,4	17	19,5	34,4	170,6
13	321,5	460	222,5	277,5	465

Πίνακας 4.38: Σφάλματα (%) για Epoch = 100 και Shift = B

Product Code	RNN	Linear Regrssion	Decision Tree	Random Forest	Gradient Boosting Tree
1	12,12	76,11	70,07	72,52	7,37
2	31,42	68,62	70,42	64,64	9,37
3	18,42	69,71	59,46	69,09	1,78
4	443,33	60,96	12,53	43,47	359,79
5	39,42	65,07	78,7	77,11	8,86
13	9,06	169,61	45,46	17,15	180,63

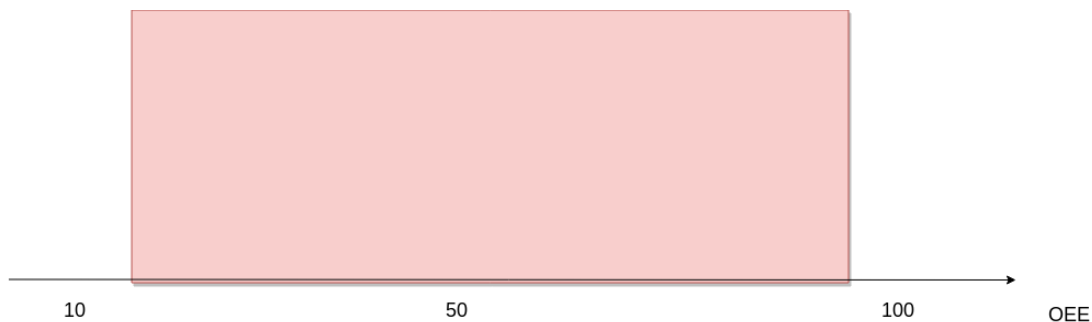
Πίνακας 4.39: Σφάλματα (%) για Epoch = 100 και Shift = C

Product Code	RNN	Linear Regrssion	Decision Tree	Random Forest	Gradient Boosting Tree
1	5,38	64,96	62,46	58,15	30,81
2	7,44	2,33	20	24,44	258
3	51,65	50,61	28,75	42,49	78,83
4	18,51	64,55	75,21	75,57	3,87
5	10,23	52,53	65,61	67,73	21,79
13	192,93	6,86	21,39	25,47	283,48

Παρατηρώντας τους παραπάνω πίνακες διαπιστώνουμε πως το Νευρωνικό Δίκτυο έχει καλύτερα αποτελέσματα σε σχέση με τους κλασικούς αλγορίθμους του Machine Learning. Ωστόσο κάποια από τα σφάλματα είναι αρκετά μεγάλα και ξεπερνάνε και το 100%. Αυτό όμως δεν σημαίνει ότι δίνει εσφαλμένα αποτελέσματα γιατί οι προβλεπόμενες τιμές δεν είναι ποσοτικές αλλά ποιοτικές.

Όπως έχουμε αναφέρει και στο προηγούμενο κεφάλαιο σε τέτοιου είδους προβλήματα, δηλαδή εκείνα τα οποία μελετάμε μια χρονοσειρά, το κυρίαρχο και πιο ικανό εργαλείο το οποίο έχουμε στην διάθεση μας είναι ένα Επαναλαμβανόμενο Νευρωνικό Δίκτυο. Αυτό φαίνεται καθώς οι τιμές που παίρνουμε από την εκτέλεση του είναι διαφορετικές μεταξύ τους και προσεγγίζουν πολύ τις πραγματικές σε αντίθεση με αυτές που παίρνουμε εκτελώντας κλασικούς αλγορίθμους που είναι σχεδόν ίδιες μεταξύ τους. Ο λόγος για τον οποίο είναι όμως το RNN το καλύτερο εργαλείο (έχει αναφερθεί αλλά θα τον θυμήσουμε πάλι), είναι ότι διαθέτει εσωτερική μνήμη. Αυτό φαίνεται καθώς οι τιμές που παίρνουμε από την εκτέλεση του είναι διαφορετικές μεταξύ τους και προσεγγίζουν πολύ τις πραγματικές σε σχέση με αυτές που παίρνουμε εκτελώντας κλασικούς αλγορίθμους που είναι σχεδόν ίδιες.

Συνεπώς από την παρατήρηση των πινάκων, όσων αφορά τις προβλέψεις, σε ένα εύρος με ποσοστά OEE από 10-80 % το RNN είναι η καλύτερη μέθοδος πρόβλεψης.



Σχήμα 4.1: Εύρος τιμών RNN

Να σημειωθεί ότι το dataset με το οποίο δουλέψαμε είχε εύρος τιμών του OEE από 0 ως 100 %. Επίσης το γεγονός ότι το εύρος των πραγματικών τιμών του OEE που προσπαθήσαμε να προβλέψουμε ήταν από 10 ως το πολύ 40 % δεν σημαίνει ότι το νευρωνικό εκπαιδεύτηκε μόνο σε αυτές.

Κεφάλαιο 5

Συμπεράσματα και Μελλοντική Εργασία

5.1 Τελικές Ρυθμίσεις του RNN

Το RNN το οποίο κατασκευάσαμε είχε κάποιες συγκεκριμένες ρυθμίσεις. Αυτές οι ρυθμίσεις αφορούν τον αριθμό των νευρώνων, τον αριθμό των κρυμμένων στρωμάτων του νευρωνικού, τις συναρτησεις βελτιστοποίησης, απωλειών και οι εποχές.

Αυτές οι ρυθμίσεις του Νευρωνικού κρίθηκαν οι κατάλληλες :

1. Νευρώνες = 50 ή 100

2. Στρώματα (Layers) = 6

InputLayer + 4 HiddenLayers + OutputLayer

3. Συνάρτηση :

→ *Loss Function* = MSE

→ *Optimizer Function* = ADAM

4. Epochs = 100

5.2 Συμπεράσματα

Για ακόμα μια φορά αποδεικνύεται ότι , ο τρόπος με τον οποίο επιλέγουμε να κατασκευάσουμε το Νευρωνικό Δίκτυο είναι μια διαδικασία εμπειρική, αφηρημένη, υποκειμενική και καλλιτεχνική. Δεν υπάρχουν βασικές αρχές στην κατασκευή ενός νευρωνικού είτε αυτό ονομάζεται **Επαναλαμβανόμενο**, είτε **Τεχνητό**, είτε **Συνελικτικό**. Συνεπώς είναι στην ευχέρεια του εκάστοτε προγραμματιστή που κατασκευάζει ένα τέτοιο δίκτυο να πειραματιστεί πάνω στο νευρωνικό και να επιλέξει τον τρόπο που θέλει εκείνος να το κατασκευάσει. Αυτό είναι και η απόδειξη ότι τα νευρωνικά δίκτυα

και η τεχνητή νοημοσύνη έχουν υψηλό ερευνητικό ενδιαφέρον. Αυτή είναι άλλωστε και η ' **γοητεία** ' της Τεχνητής Νοημοσύνης.

Στο εισαγωγικό κεφάλαιο είχαν τεθεί κάποια ερωτήματα σχετικά με την συνεισφορά της εργασίας, τα οποία τελικά δεν μείναν αναπάντητα. Ξεκινήσαμε λοιπόν, μιλώντας για την συνεισφορά την εργασίας και τα ζητήματα τα οποία θα μπορεί να επιλύσει. Αυτά ήταν :

1. **Πώς θα μπορούσαμε να βελτιώσουμε τον τρόπο που προγραμματίζουμε την παραγωγή, προκειμένου να πετύχουμε καλύτερα αποτελέσματα.**

Η πρόβλεψη της τιμή της του ΟΕΕ είναι ένα πολύτιμο εργαλείο για τον υπεύθυνο της παραγωγής. Είναι ανάγκη να γίνονται μια σειρά από προβλέψεις προκειμένου να γνωρίζει κανείς, με κάποιες αποκλίσεις, τι να περιμένει όσον αφορά την απόδοση της μηχανής.

2. **Πώς θα μπορούσαμε να αποφύγουμε ή να προβλέψουμε πιθανούς κινδύνους.**

Έχοντας εκτελέσει το πείραμα το οποίο προβλέπει τον αριθμό των σφαλμάτων μιας μηχανής, μπορούμε να αναγνωρίσουμε κινδύνους και να τους αντιμετωπίσουμε πριν συμβούν. Κάτι τέτοιο είναι πολύ σημαντικό και σε μια πιθανή προέκταση της εργασίας θα μπορούσε να συνδιαστεί με ένα ANN που εκτός από την πρόβλεψη του αριθμού των σφαλμάτων θα έχει την δυνατότητα της ταξινόμησης των σφαλμάτων αυτών σε κατηγορίες όπως για παράδειγμα διακοπή ρεύματος, υπερχείληση της γεμιστικής μηχανής κλπ.

3. **Πώς θα μπορούσαμε να μειώσουμε το ρίσκο σε μια μεγάλης σημασίας απόφαση.**

Η απάντηση σε αυτό το ερώτημα είναι ο συνδιασμός των δύο παραπάνω. Δηλαδή έχοντας στην διάθεση του ο υπεύθυνος γνώση για το την απόδοση μια μηχανής και γνώση για τα σφάλματα, έχει αυτομάτως στην διάθεση του ένα πολύ ισχυρό εργαλείο.

Συνεπώς έχοντας καλές προβλέψεις στην διάθεσή του κάποιος μπορεί να εκτιμήσει καλύτερα καταστάσεις, να οργανώσει και να προγραμματίσει καλύτερα την παραγωγή.

5.3 Μελλοντική Εργασία

Το τελευταίο κομμάτι της παρούσας εργασίας αφορά την μελλοντική εργασία που θα γίνει όσον αφορά το λογισμικό. Αρχικά το λογισμικό βρίσκεται σε μια πολύ πρώιμη μορφή χωρίς αυτό να σημαίνει ότι δεν μπορεί να βελτιωθεί και να γίνει ένα πάρα πολύ χρήσιμο εργαλείο για ένα εργοστάσιο. Αυτό απαιτεί παραπάνω μελέτη και πειράματα στα Νευρωνικά Δίκτυα προκειμένου να βελτιωθεί το λογισμικό. Στην συνέχεια θα πρέπει να μετατραπεί η μορφή του λογισμικού από κώδικα σε μορφή εφαρμογής (ανάπτυξη του interface).

5.3.1 IBM Watson Studio

Το Watson Studio είναι μια πολύ χρήσιμη πλατφόρμα που προσφέρει η **IBM** όπου ο καθένας μπορεί να γράψει κώδικα, να σχεδιάσει μοντέλα μηχανικής και βαθειάς μάθησης, να κάνει αναλύσεις δεδομένων και έχει και πολλές ακόμα δυνατότητες. Στο κομμάτι του interface λοιπόν το Watson Studio μπορεί να φανεί πάρα πολύ χρήσιμο αφού υπάρχει η δυνατότητα να δημιουργήσεις ένα πολύ φιλικό προς τον χρήστη περιβάλλον με μεγάλη ευκολία. Άρα όσον αναφορά την μελλοντική εργασία, υπάρχει πιθανότητα για συνεργασία μας με την IBM προκειμένου να δημιουργηθεί μια **beta** μορφή του λογισμικού με το κατάλληλο interface προκειμένου να ξεκινήσει η εφαρμογή του σε μια γραμμή παραγωγής ενός εργοστασίου.

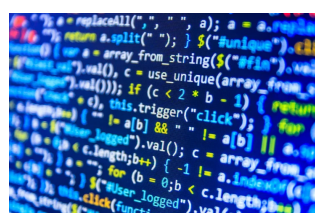
5.3.2 Αξιοποίηση του λογισμικού σε σύστημα Performance Management και αξιοποίηση του Cloud

Προφανώς μια τέτοια εφαρμογή για να έχει νόημα να λειτουργήσει και να προσφέρει αποτελέσματα θα πρέπει να υπάρχουν δεδομένα. Πώς όμως θα γίνει κάτι τέτοιο ; Η απάντηση είναι με το ' πάντρεμα ' της εφαρμογής με ένα σύστημα Performance Management, ένα σύστημα δηλαδή το οποίο συλλέγει πληροφορίες και δεδομένα, στην δική μας περίπτωση δεδομένα για μία ή περισσότερες γραμμές παραγωγής. Συνεπώς με τον συνδυασμό ενός τέτοιου συστήματος, συνεχούς συλλογής πληροφοριών η οποία θα μας εξασφαλίσει πληθώρα δεδομένων, δημιουργείται εύπορη γη όπου μπορεί εύκολα μια τέτοιου είδους εφαρμογή να λειτουργήσει πολύ αποδοτικά.

Όμως για να λειτουργήσει αποδοτικά αυτός ο συνδυασμός χρειάζεται ο κατάλληλος cloud server. Αυτό είναι πολύ λογικό αφού όταν μιλάμε για συλλογή δεδομένων πρέπει ήδη να έχουμε εξασφαλίσει ότι όλα αυτά τα δεδομένα θα έχουν αρκετό χώρο στην βάση, αφού κάθε πληροφορία είναι απαραίτητη.

5.3.3 Επίλογος

Τελειώνοντας λοιπόν αυτή την διπλωματική εργασία καταλήγουμε στο συμπέρασμα ότι η τεχνητή νοημοσύνη μπορεί να συμβάλει σημαντικά στην βελτίωση και εκσυγχρονισμό της βιομηχανίας και συγκεκριμένα του τομέα της παραγωγής. Συνεπώς ανακεφαλαιώνοντας όσα είπαμε για την μελλοντική εργασία παρατίθεται το τελικό σχεδιάγραμμα που αφορά το πώς αυτό το λογισμικό θα μπορούσε να αποκτήσει υπόσταση και να τεθεί σε λειτουργία σε κάποια γραμμή παραγωγής.



(α') Κώδικας

+



(β') Interface

+



(γ') Cloud Service



(δ') Τελική Εφαρμογή

Σχήμα 5.1

Βιβλιογραφία

- [1] Βασίλης Ασημακόπουλος, Φώτιος Πετρόπουλος, 2011, Επιχειρησιακές προβλέψεις , Αθήνα, Εκδόσεις Συμμετρίας
- [2] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28 (ICML'13), Sanjoy Dasgupta and David McAllester (Eds.), Vol. 28. JMLR.org III-1310-III-1318.
- [3] Burnwal, S. and Deb, S. Int J Adv Manuf Technol (2013) 64: 951. <https://doi.org/10.1007/s00170-012-4061-z>
- [4] Mukund Subramaniyan, Production Data Analytics ? To identify productivity potentials, Master Thesis
- [5] Marco Garetti and Marco Taisch (1999) Neural networks in production planning and control, Production Planning and Control, 10:4, 324-339,
- [6] Zwietering, P. J., Kraaij, van, M. J. A. L., Aarts, E. H. L., and Wessels, J. (1991). Neural networks and production planning. (Memorandum COSOR; Vol. 9115). Eindhoven: Technische Universiteit Eindhoven.
- [7] Mangal, Ankita and Kumar, Nishant. (2016). Using Big Data to Enhance the Bosch Production Line Performance: A Kaggle Challenge. 10.1109/BigData.2016.7840826.
- [8] Mehrjoo, S. and Bashiri, M. J Ind Eng Int (2013) 9: 14. <https://doi.org/10.1186/2251-712X-9-14>
- [9] ?Production scheduling using genetic algorithms, Control Eng. Pract., vol. 1, no. 1, p. 207, 1993.
- [10] Paolo Priore, David de la Fuente, Javier Puente, Jos? Parre?o, A comparison of machine-learning algorithms for dynamic scheduling of flexible manufacturing systems, Engineering Applications of Artificial Intelligence, Volume 19, Issue 3, 2006, Pages 247-255, ISSN 0952-1976,
- [11] Yeou-Ren Shiue, Ken-Chuan Lee, Chao-Ton Su, Real-time scheduling for a smart factory using a reinforcement learning approach, Computers and Industrial Engineering, 2018,ISSN 0360-8352

- [12] X. Fan, X. Zhu, K. C. Kuo, C. Lu and J. Wu, "Big data analytics to improve photomask manufacturing productivity," 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, 2017, pp. 2341-2345.
- [13] Heger, Jens and Bani, Hatem and Scholz-Reiter, Bernd. (2012). Improving Production Scheduling with Machine Learning.
- [14] <https://bigtheta.io/2016/02/24/intro-to-neural-networks.html>
- [15] <https://bigtheta.io/2016/02/27/the-math-behind-backpropagation.html>
- [16] <https://www.statisticshowto.datasciencecentral.com/moving-average/>
- [17] <https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5>
- [18] <http://digilib.teiemt.gr/jspui/bitstream/123456789/2527/1/022005x03x077.pdf>
- [19] <https://en.wikipedia.org/wiki/Backpropagation>
- [20] <http://www.cs.uoi.gr/~arly/courses/nm/slides/K5.pdf>
- [21] <https://machinelearningmastery.com/what-is-deep-learning/>
- [22] <https://www.pwc.co.uk/issues/data-analytics/industries/data-and-analytics-in-the-manufacturing-sector.html>
- [23] https://repository.kallipos.gr/bitstream/11419/3382/1/02_chapter_04.pdf
- [24] <https://pergamos.lib.uoa.gr/uoa/dl/frontend/file/lib/default/data/1713761/theFile>
- [25] <https://searchdatamanagement.techtarget.com/definition/data-analytics>
- [26] http://ec.europa.eu/eurostat/statistics-explained/index.php?title=Employment_statistics/el
- [27] <https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d>
- [28] <https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/an-executives-guide-to-ai?cid=other-eml-ttn-mip-mck-oth-1806&hlkid=4b5adbd962424349b47af01ff87b9dac&hctky=9845522&hdpid=f3a30356-6857-47f4-a1a5-0d0e5ff12a0a>

