



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗΣ

**Πρόβλεψη ωριαίας ζήτησης ενέργειας σε σταθμούς φόρτισης ηλεκτρικών  
οχημάτων με τη χρήση βαθιών νευρωνικών δικτύων**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ευάγγελος Γ. Βισκαδούρος

**Επιβλέπων:** Εμμανουήλ Βαρβαρίγος

Καθηγητής Ε.Μ.Π

Αθήνα, Ιούλιος 2021





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗΣ

**Πρόβλεψη ωριαίας ζήτησης ενέργειας σε σταθμούς φόρτισης ηλεκτρικών  
οχημάτων με τη χρήση βαθιών νευρωνικών δικτύων**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

Ευάγγελος Γ. Βισκαδούρος

**Επιβλέπων:** Εμμανουήλ Βαρβαρίγος

Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 26<sup>η</sup> Ιουλίου 2021.

.....  
Θεοδώρα Βαρβαρίγου

Καθηγήτρια Ε.Μ.Π

.....  
Εμμανουήλ Βαρβαρίγος

Καθηγητής Ε.Μ.Π

.....  
Συμεών Παπαβασιλείου

Καθηγητής Ε.Μ.Π

Αθήνα, Ιούλιος 2021

.....  
Ευάγγελος Γ. Βισκαδούρος

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

© 2021 – All rights reserved



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ

ΠΛΗΡΟΦΟΡΙΚΗΣ

Copyright ©—All rights reserved Ευάγγελος Βισκαδούρος, 2021.

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.



# Περίληψη

Αναλογικά με την καθημερινή αύξηση του πληθυσμού στον κόσμο, αυξάνονται και οι ανάγκες των ανθρώπων. Με τον ίδιο τρόπο αυξάνονται οι απαιτήσεις στο σύστημα ισχύος και εμφανίζεται η ανάγκη για καλύτερη χρήση της ενέργειας. Η εξάντληση των αποθεμάτων ορυκτών καυσίμων, οι επί του παρόντος κυρίαρχοι ενεργειακοί πόροι, δίνει ένα κάλεσμα αφύπνισης για εξεύρεση εναλλακτικών πηγών ενέργειας για αυτούς τους τομείς. Το Smart Grid, μια αναβάθμιση του τρέχοντος συστήματος που είναι πιο αξιόπιστο, αποδοτικό, προσιτό, ασφαλές και φιλικό προς το περιβάλλον, είναι η λύση σε αυτήν την αυξανόμενη ανησυχία. Τα ηλεκτρικά οχήματα είναι πιθανό να εξυπηρετούν το ηλεκτρικό δίκτυο ως ανεξάρτητη κατανεμημένη πηγή ενέργειας. Από ορισμένες μελέτες έχει αποκαλυφθεί ότι τα περισσότερα οχήματα σταθμεύουν σχεδόν το 95% του χρόνου τους. Σε αυτήν την περίπτωση, μπορούν να παραμείνουν συνδεδεμένοι στο δίκτυο και να είναι έτοιμοι να παραδώσουν την ενέργεια που είναι αποθηκευμένη στις μπαταρίες τους. Λαμβάνοντας υπόψη την πρόοδο των οχημάτων με κινητήρα εσωτερικής καύσης, καθιστώντας τα ηλεκτρικά οχήματα αποτελεσματικά και έτσι δημοφιλή είναι αναμφίβολα ένα δύσκολο έργο. Ένας από τους βασικούς παράγοντες για την αποδοχή των EV στην αγορά θα είναι η διαθεσιμότητα σταθμών φόρτισης. Ταυτόχρονα, οι σταθμοί φόρτισης, πρέπει να γίνουν πιο προηγμένοι, να χρησιμοποιούν ενέργεια σε πιο αποτελεσματικό ζήτημα και, συνεπώς, να παρέχουν καλύτερες υπηρεσίες φόρτισης στους ιδιοκτήτες EV, από άποψη τιμής και διαθεσιμότητας ενέργειας. Αυτή η διατριβή διαμορφώνει τη ζήτηση ενέργειας για ιδιοκτήτες Ηλεκτρικών Οχημάτων, φορτίζοντας σε δημόσιους σταθμούς φόρτισης. Αυτός ο αλγόριθμος λαμβάνει ιστορικά δεδομένα των περιόδων φόρτισης ηλεκτρικών οχημάτων και προβλέπει τη μελλοντική ζήτηση με βάση χρονοσειρές, χρησιμοποιώντας ένα LSTM με τη χρήση Keras και Tensorflow στο Python. Ο αλγόριθμος συγκρίνεται με βασικούς αλγόριθμους πρόβλεψης, αποδεικνύοντας την αποτελεσματικότητά του στην πρόβλεψη της ωριαίας ζήτησης ενέργειας σταθμών φόρτισης.

## Λέξεις - κλειδιά

Δίκτυα μακράς και βραχείας μνήμης, Αναδρομικά Νευρωνικά Δίκτυα, Ηλεκτρικά Οχήματα, Ευφυή Δίκτυα, Βαθιά Μηχανική Μάθηση, Βαθιά Νευρωνικά Δίκτυα, Χρονοσειρές, Προβλέψεις, Ηλεκτρικά Οχήματα, Έξυπνη Φόρτιση





# Abstract

In day to day increasing population in the world, the needs of people are increasing. In that same way the demands in the power system are increasing, and there comes the need for better utilisation of energy. Depletion of fossil fuel reserves, the currently dominant energy resources, gives a wakeup call for finding alternative energy sources for these sectors. Smart Grid, an upgrade of the current system which is more reliable, efficient, affordable, secured, and environment friendly, is the solution to this growing concern. Electric Vehicles are potential on serving the electric grid as an independent distributed energy source. It has been revealed by some studies that most vehicles are parked almost 95% of their time. In this case, they can remain connected to the grid and be ready to deliver the energy stored in their batteries. Taking into consideration the advancement of internal combustion engine vehicles, making electric vehicles efficient, and thus popular is undoubtedly a difficult project. One of the key factors for the acceptance of EVs in the marketplace will be the availability of charging stations. At the same time, charging stations, need to become more advanced, use energy in a more efficient matter and thus provide better charging services to the EV owners, in terms of energy price and availability.

This thesis models the demand for energy of Electric Vehicles owners, charging in public charging stations. This algorithm takes historic data of electric vehicle charging sessions, and predicts the future demand based on time-series, using an LSTM with the use of Keras and Tensorflow in Python. The algorithm is compared to baseline forecasting algorithms, proving its efficiency in predicting charging stations' hourly energy demand.

## Keywords

LSTM, RNN, EV, Smart Grid, Deep Learning, Deep Neural Networks, Time Series Data, Forecast, Smart Charging

## Ευχαριστίες

Με την ολοκλήρωση των προπτυχιακών σπουδών μου, θα ήθελα να ευχαριστήσω ειδικά τον επιβλέποντα καθηγητή μου Εμμανουήλ (Μάνο) Βαρβαρίγο για την δυνατότητα που μου έδωσε να εργαστώ πάνω σε ένα τόσο ενδιαφέρον και σύγχρονο θέμα, και τον Διδάκτορα Γεώργιο Τσαούσογλου για την πολύτιμη βοήθεια και καθοδήγησή του κατά την εκπόνηση της παρούσας διπλωματικής εργασίας. Τέλος, θα ήθελα να αφιερώσω την διπλωματική μου στους γονείς μου, και τους φίλους μου και να τους ευχαριστήσω για την απεριόριστη υπομονή και στήριξη τους όλα αυτά τα χρόνια.

## **Acknowledgments**

Upon completion of my undergraduate studies, I would like to thank my supervising professor Emmanuel (Manos) Varvarigos for the opportunity he offered to me, to work on such an interesting and contemporary topic, and Dr. Georgios Tsaousoglou for his valuable help and guidance during the elaboration of the present dissertation. Finally, I would like to dedicate my diploma to my parents and friends and thank them for their unlimited patience and support over those years.

## Κατάλογος Σχημάτων

Σχήμα 2.1: Τα βασικά στοιχεία του συμβατικού (στα αριστερά) και του ευφυούς δικτύου (στα δεξιά)	16
Σχήμα 3.1 Βασικά συστατικά ενός νευρώνα σε απλοποιημένη μορφή.....	20
Σχήμα 3.2 Απλός τεχνητός νευρώνας.....	21
Σχήμα 3.3: Ένας απλός τύπος νευρωνικού δικτύου.....	21
Σχήμα 3.4 Ένα ξεδιπλωμένο αναδρομικό νευρωνικό δίκτυο (RNN).....	22
Σχήμα 3.5: Διαφορετικοί τύποι αναδρομικών νευρωνικών δικτύων (RNN).....	23
Σχήμα 3.6: Ένα απλός τεχνητός νευρώνας στα αριστερά, συγκρινόμενος με ένα RNN στα δεξιά[14]	24
Σχήμα 3.7: Τα κρυφά επίπεδα ενός RNN.....	25
Σχήμα 3.8: Τα επίπεδα του RNN στο παράδειγμα πρόβλεψης χαρακτήρα.....	26
Σχήμα 3.9: Η οπισθοδιάδοση (backpropagation) στο RNN.....	27
Σχήμα 3.10: Το πρόβλημα της εξαφανιζόμενης κλίσης (vanishing gradient).....	28
Σχήμα 3.11: Το πρόβλημα της έκρηξης κλίσης (exploding gradient).....	28
Σχήμα 3.12: Η επαναλαμβανόμενη μονάδα του LSTM και οι πύλες της.....	29
Σχήμα 3.13: Οι πύλες του LSTM.....	30
Σχήμα 3.14: Η πύλη forget του LSTM.....	30
Σχήμα 3.15: Η πύλη εισόδου/input του LSTM.....	31
Σχήμα 3.16: Η πύλη εξόδου/output του LSTM.....	31

# Περιεχόμενα

Περίληψη	7
Abstract.	9
Ευχαριστίες	10
Acknowledgments	11
Κατάλογος Σχημάτων.	12
Περιεχόμενα	13
<b>Κεφάλαιο 1 : Εισαγωγή</b>	<b>14</b>
1.1 Αντικείμενο Διπλωματικής.....	14
1.2 Οργάνωση της Διπλωματικής.....	14
<b>Κεφάλαιο 2: Ηλεκτρικά οχήματα και Ευφυή Δίκτυα</b>	<b>15</b>
2.1 Ευφυή Δίκτυα.....	15
2.2 Ηλεκτρικά οχήματα και η ένταξη τους στα ευφυή δίκτυα.....	18
<b>Κεφάλαιο 3: Θεωρητικό υπόβαθρο</b>	<b>20</b>
3.1 Νευρωνικά Δίκτυα.....	20
3.2 Αναδρομικά Νευρωνικά Δίκτυα (RNN).....	22
3.3 Δίκτυα μακράς και βραχείας μνήμης LSTM.....	29
<b>Κεφάλαιο 4: Δεδομένα</b>	<b>32</b>
4.1 Πηγές Δεδομένων.....	32
4.2 Επεξεργασία Δεδομένων.....	34
<b>Κεφάλαιο 5: Σχεδίαση μοντέλου και αξιολόγηση αποτελεσμάτων</b>	<b>39</b>
5.1 Μοντέλα βασικής απόδοσης.....	39
5.1.1 Persistence .....	40
5.1.2 Linear regression.....	42
5.2 Μοντέλο μακράς και βραχείας μνήμης LSTM.....	47
5.3 Εκπαίδευση μοντέλου LSTM.....	48
5.4 Αποτελέσματα μοντέλου LSTM .....	50
<b>Κεφάλαιο 6: Συμπεράσματα και μελλοντική εργασία</b>	
6.1 Συμπεράσματα και μελλοντική εργασία.....	53
<b>Βιβλιογραφία</b>	<b>54</b>

# Κεφάλαιο 1

## Εισαγωγή

### 1.1 Αντικείμενο Διπλωματικής

Σκοπός της διπλωματικής είναι η εφαρμογή και μελέτη της βαθιάς μηχανικής μάθησης και συγκεκριμένα των αναδρομικών νευρωνικών δικτύων μακράς και βραχείας μνήμης (Lstm), για την πραγματοποίηση βραχυπρόθεσμων προβλέψεων ζήτησης ενέργειας σε σταθμούς φόρτισης αυτοκινήτων. Η βαθιά μηχανική μάθηση χρησιμοποιείται ήδη ευρέως σε ένα μεγάλο φάσμα προβλημάτων. Ως υποκατηγορία τα αναδρομικά νευρωνικά δίκτυα και ειδικότερα τα LSTMs είναι πολύ αποτελεσματικά στη μελέτη χρονοσειρών. Στα πλαίσια αυτής της διπλωματικής, πραγματοποιούνται προβλέψεις ζήτησης ηλεκτρικής ενέργειας σε σταθμούς φόρτισης ηλεκτρικών οχημάτων και εξετάζονται οι παράγοντες που επηρεάζουν την απόδοση των προβλέψεων, με τελικό σκοπό την βελτιστοποίηση των αποτελεσμάτων.

### 1.2 Οργάνωση της διπλωματικής

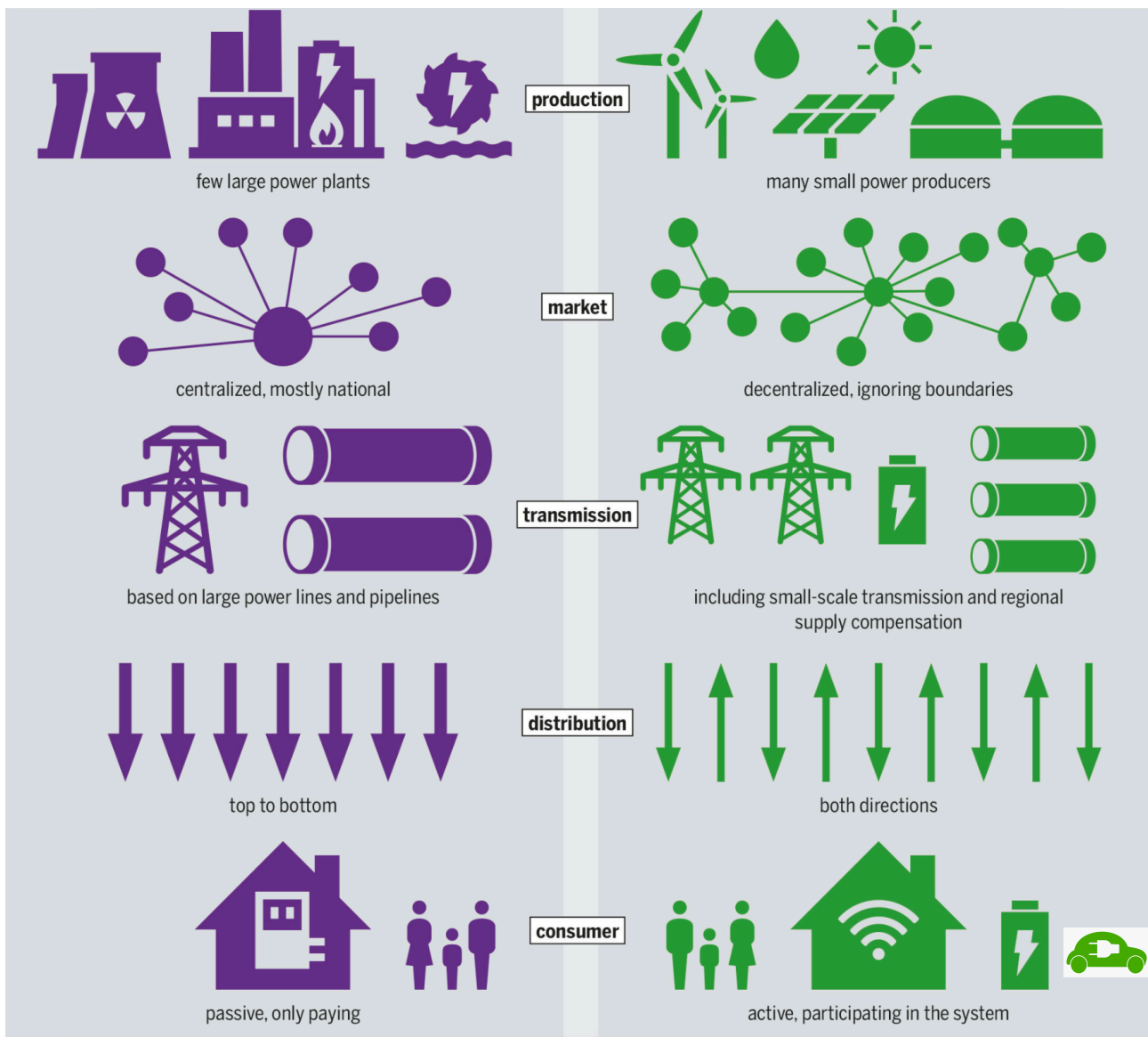
Στο πρώτο κεφάλαιο αναφέρεται το θέμα και το αντικείμενο της διπλωματικής. Στη συνέχεια στο Κεφάλαιο 2 αναλύεται το περιβάλλον κάτω από το οποίο δημιουργήθηκε η ανάγκη για έρευνα πάνω στην πρόβλεψη ζήτησης ενέργειας για σταθμούς φόρτισης ηλεκτρικών οχημάτων. Στο τρίτο κεφάλαιο αναλύονται θεωρητικά οι έννοιες των νευρωνικών και συγκεκριμένα των αναδρομικών νευρωνικών δικτύων και των δικτύων βραχείας και μακράς μνήμης που χρησιμοποιήθηκαν στην ανάπτυξη του μοντέλου, για λύση του ζητούμενου. Στο κεφάλαιο 4, γίνεται μία ανάλυση γύρω από τα διαθέσιμα δεδομένα αυτή τη στιγμή (Ιούλιος 2021) και περιγράφεται αναλυτικά ο τρόπος επεξεργασίας των δεδομένων πριν από την χρήση τους στο νευρωνικό δίκτυο. Στο κεφάλαιο 5, αναλύεται η βασική απόδοση, σε δύο βασικά μοντέλα, το persistence και της γραμμικής παλινδρόμησης, και στη συνέχεια η απόδοση αυτή χρησιμοποιείται ως μέτρο σύγκρισης για να αξιολογηθεί το αναπτυσσόμενο LSTM μοντέλο. Τέλος, στο Κεφάλαιο 6 αναφέρονται κάποιες προσθήκες που θα μπορούσαν να γίνουν μελλοντικά, και με την ύπαρξη περισσότερων δεδομένων.

## Κεφάλαιο 2

### Ευφυή Δίκτυα και Ηλεκτρικά Οχήματα

#### 2.1 Ευφυή Δίκτυα

Το σύστημα ηλεκτρικής ενέργειας που αναπτύχθηκε πριν από αρκετές δεκαετίες, μετατρέπεται σταδιακά σε “ευφύες”, παγκοσμίως γνωστό ως Smart Grid [1], λόγω της διαθεσιμότητας πληθώρας νέων τεχνολογιών αλλά και της ανάγκης για χρήση ανανεώσιμων πηγών ενέργειας. Το ευφύες δίκτυο παρέχει λύση για σταθερότητα, αξιοπιστία, αποδοτικότητα, χαμηλότερο κόστος και ελάττωση της ατμοσφαιρικής ρύπανσης. Τρεις κυρίαρχοι παράγοντες επηρεάζουν τα μελλοντικά ηλεκτρικά συστήματα του κόσμου: κυβερνητικές πολιτικές, ανάγκες αποδοτικότητας του καταναλωτή και εισαγωγή νέων τεχνολογιών. Παράλληλα, περιβαλλοντικά κίνητρα έχουν ωθήσει κυβερνητικές πολιτικές σε όλο τον κόσμο να επηρεάζουν ολόκληρο το ενεργειακό σύστημα με στόχο τη μεγαλύτερη αποδοτικότητα, τη συντήρηση και την εισαγωγή ανανεώσιμων πηγών ηλεκτρικής ενέργειας. Αυτοί είναι οι κύριοι παράγοντες που επεκτείνουν από τη μία πλευρά τη χρήση όλων των νέων τεχνολογιών ανανεώσιμης ενέργειας και αποθήκευσης, και από την άλλη νέων τεχνικών ενεργειακής απόδοσης και διατήρησης. Οι καταναλωτές γίνονται περισσότερο συμμετοχικοί αφού αποκτούν τη δυνατότητα να συμμετέχουν στις αποφάσεις κατανάλωσης ενέργειας που επηρεάζουν την καθημερινή τους ζωή. Ταυτόχρονα, επεκτείνουν τις ενεργειακές τους ανάγκες. Η συμμετοχή των καταναλωτών στα ευφυή δίκτυα περιλαμβάνει εκτεταμένη χρήση ηλεκτρικών οχημάτων (τόσο ΙΧ οχήματα όσο και μεγαλύτερα οχήματα όπως φορτηγά και λεωφορεία), έλεγχο οικιακών συσκευών με σκοπό της εξοικονόμησης ενέργειας, ιδιοκτησία καταναλωτή παραγωγής ολοένα και περισσότερο από ανανεώσιμες πηγές ενέργειας και διαχείριση αποθήκευσης ηλεκτρικής ενέργειας σε ιδιωτικό επίπεδο (πχ. σε ένα νοικοκυριό) έτσι ώστε να συμπίπτει η παραγωγή με την ανάγκη για κατανάλωση ελαχιστοποιώντας έτσι τη ζήτηση ενέργειας από το δίκτυο. Η διαθεσιμότητα νέων τεχνολογιών, όπως περισσότεροι και ενημερωμένοι αισθητήρες SCADA (σύστημα εποπτείας, ελέγχου και συλλογής πληροφοριών), ασφαλείς αμφίδρομες επικοινωνίες, ολοκληρωμένη διαχείριση δεδομένων και έξυπνοι, αυτόνομοι ελεγκτές έχουν δημιουργήσει ευκαιρίες που δεν υπήρχαν πριν από μια δεκαετία. Το σύστημα ηλεκτρικής ενέργειας του μέλλοντος πρέπει να χρησιμοποιεί όλες αυτές τις τεχνολογίες, για να αντιμετωπίσει όλες αυτές τις ανάγκες δημιουργώντας ένα πιο έξυπνο, πιο αποτελεσματικό και βιώσιμο δίκτυο. Τα τελευταία χρόνια, υπήρξαν πολλά άρθρα και συνέδρια σχετικά με τα ευφυή δίκτυα, αλλά παραμένει μεγάλη η σύγχυση μεταξύ των συμμετεχόντων, σχετικά με το τι συνεπάγεται ο όρος. Αν και έχουν προταθεί πολλοί διαφορετικοί ορισμοί για τα ευφυή δίκτυα, στις περισσότερες περιπτώσεις οι χρήστες έχουν προτείνει ιδιαίτερα συγκεκριμενοποιημένους ορισμούς που σχετίζονται με τις συγκεκριμένες εφαρμογές των εκάστοτε χρηστών και των τοπικών τους αναγκών για εκσυγχρονισμό. Παρακάτω, ορίζουμε το ευφύες δίκτυο (Smart Grid) με τους ευρύτερα αποδεκτούς παγκόσμιους όρους. Αρχίζουμε με μια περιγραφή της σύνθεσης του σημερινού συμβατικού συστήματος ηλεκτρικής ενέργειας και στη συνέχεια θα προσδιορίσουμε τους τομείς που πρέπει να αλλάξουν για να παρέχουν την απαραίτητη νοημοσύνη και έλεγχο για τη μετατροπή σε ασφαλές και αποτελεσματικό ευφύες δίκτυο του μέλλοντος. Οι παρακάτω περιγραφές, περιγράφουν συνοπτικά τις αλλαγές που ενδέχονται να οδηγήσουν στη μετάβαση στο ευφύες δίκτυο. Energy atlas 2018



Σχήμα 2.1: Τα βασικά στοιχεία του συμβατικού (στα αριστερά) και του ευφυούς δικτύου (στα δεξιά)

Μια γενική περιγραφή του σημερινού συμβατικού ηλεκτρικού συστήματος διανομής απεικονίζεται στα αριστερά του Σχήματος 2.1. Παραδοσιακά [2], το σύστημα χωρίζεται σε ως επί το πλείστον απομονωμένα εξαρτήματα: παραγωγή, μετάδοση, υποσταθμούς, διανομή και τον τελικό καταναλωτή. Βασικά χαρακτηριστικά του συμβατικού συστήματος που θα αλλάξουν με σκοπό την μετάβαση σε ένα ευφές δίκτυο είναι τα ακόλουθα χαρακτηριστικά:

1. Οι κεντρικές πηγές παραγωγής ενέργειας
2. Η μονόδρομη ροή ενέργειας από την πηγή στους πελάτες,
3. Η παθητική συμμετοχή των πελατών - καταναλωτών, αφού στο συμβατικό σύστημα η γνώση των καταναλωτών σχετικά με τη χρήση ηλεκτρικής ενέργειας περιορίζεται σε ένα λογαριασμό ενέργειας που λαμβάνεται και εξοφλείται μηνιαία
4. Η παρακολούθηση και ο έλεγχος σε πραγματικό χρόνο, που περιορίζεται κυρίως στην παραγωγή και μετάδοση, και μόνο σε ορισμένες επιχειρήσεις κοινής ωφέλειας, επεκτείνεται στο σύστημα διανομής



5. Το μη ευέλικτο σύστημα που καθιστά δύσκολο το είτε να δέχεται ηλεκτρική ενέργεια από εναλλακτικές πηγές σε οποιοδήποτε σημείο κατά μήκος του δικτύου, είτε να διαχειρίζεται αποτελεσματικά και βιώσιμα νέες υπηρεσίες που επιθυμούν οι χρήστες ηλεκτρικής ενέργειας.

Αυτά τα συμβατικά χαρακτηριστικά εξυπηρετούσαν επαρκώς τις ανάγκες των ηλεκτρικών καταναλωτών και των πελατών στο παρελθόν. Ωστόσο, οι νέες ανάγκες καλύτερης διαχείρισης της ενέργειας, από συνειδητοποιημένους καταναλωτές, σε συνδυασμό με πολιτικές των χωρών που προωθούν τη βιωσιμότητα και την ενεργειακή ανεξαρτησία από ξένες πηγές, τη διαθεσιμότητα πιο έξυπνων τεχνολογιών και τις ολοένα αυξημένες ανάγκες για ενέργεια, απαιτούν ένα εκσυγχρονισμένο σύστημα ηλεκτρικής ενέργειας που να διαφέρει ριζικά και στους 5 τομείς που αναφέρονται παραπάνω.

Ένα γενικό σχήμα του ευφυούς δικτύου ηλεκτρικής ενέργειας, ή Smart Grid, παρουσιάζεται στα δεξιά του Σχήματος 1. Οι βασικές απαιτήσεις αυτού του συστήματος θα χαρακτηρίζονται από τους ακόλουθους μετασχηματισμούς:

- Να καθιστεί δυνατή η ενσωμάτωση ανανεώσιμων πηγών ενέργειας
- Να επιτραπεί η ενεργή συμμετοχή των πελατών με σκοπό την καλύτερη εξοικονόμηση ενέργειας
- Να επιτευχθεί από τα συστήματα επικοινωνιών η αντιμετώπιση της ασφάλειας του συστήματος
- Να βελτιωθεί η χρήση των υπαρχόντων υποδομών για την αντιμετώπιση της μακροπρόθεσμης βιωσιμότητας
- Να επιτραπεί η βελτιστοποιημένη ροή ενέργειας, για να μειωθούν οι απώλειες και το κόστος της ενέργειας
- Να επιτραπεί η ενσωμάτωση ηλεκτρικών οχημάτων για να μειωθεί η εξάρτηση από τα καύσιμα υδρογονανθράκων

## 2.2 Ηλεκτρικά οχήματα και η ένταξη τους στα ευφυή δίκτυα

Τα ηλεκτρικά οχήματα (EVs - συντομογραφία από το Electric Vehicle) [3], με τις δυνατότητές τους για εξοικονόμηση υδρογονανθράκων και μειώσεις εκπομπών διοξειδίου του άνθρακα, εγείρουν σημαντικό καταναλωτικό και πολιτικό ενδιαφέρον. Ωστόσο, δεδομένης της αρκετά μεγάλης ποσότητας ηλεκτρικής ενέργειας που χρειάζονται για τη φόρτιση, προκαλείται μία σύγκυση μεταξύ των επιχειρήσεων κοινής ωφελείας που είναι υπεύθυνες για την παροχή αυτής της ηλεκτρικής ενέργειας, αφού το συμβατικό ηλεκτρικό δίκτυο θα υποφέρει από τον αντίκτυπο της αύξησης των αναγκών λόγω φόρτισης των ηλεκτρικών οχημάτων. Για να ελαχιστοποιηθούν οι αρνητικές συνέπειες, είναι σημαντικό να προετοιμαστεί η ενσωμάτωση των ηλεκτρικών οχημάτων στο δίκτυο. Η πιο σοβαρή ανησυχία για τις επιχειρήσεις κοινής ωφέλειας είναι ο έλεγχος πότε οι σταθμοί φόρτισης ηλεκτρικών οχημάτων (γνωστοί και ως EVSE από τα αρχικά Electric Vehicle Supply Equipment) θα χρειάζονται φορτίο από το δίκτυο. Ένα υψηλό ποσοστό καταναλωτών που θα συνδέει προς φόρτιση ενστικτωδώς τα ηλεκτρικά οχήματα τους όταν φτάνουν στο σπίτι από τη δουλειά, πιθανόν να προκαλέσει σοβαρές επιπτώσεις στην αιχμή της ζήτησης στο δίκτυο. Ένας οικιακός σταθμός φόρτισης τυπικά απαιτεί φορτίο ηλεκτρικής ενέργειας ίσο με 6,6kW (240V 30A) - περίπου ισοδύναμο με το φορτίο ενός ολόκληρου σπιτιού στα 7kW, που σημαίνει ότι η φόρτιση ενός μόνου ηλεκτρικού οχήματος μπορεί να διπλασιάσει το μέγιστο φορτίο ενός σπιτιού έτσι ώστε ακόμη και χαμηλά επίπεδα υιοθέτησης ηλεκτρικών οχημάτων σε μια συγκεκριμένη γειτονιά μπορεί να πιέσουν αρκετά την υπάρχουσα ηλεκτρική υποδομή. Η επικράτηση των ηλεκτρικών οχημάτων στην αγορά (EV) απαιτεί την προετοιμασία του περιβάλλοντος του προϊόντος. Είναι απαραίτητο να αναβαθμιστούν τα δίκτυα ηλεκτρικής ενέργειας και να αναπτυχθούν δημόσιοι σταθμοί φόρτισης.

Οι δημόσιες εγκαταστάσεις φόρτισης είναι απαραίτητες λοιπόν για την επικράτηση των ηλεκτρικών οχημάτων. Παρόλο που πολλές πόλεις σχεδιάζουν την κατασκευή και επέκταση των υποδομών φόρτισης των ηλεκτρικών οχημάτων, είναι πιθανό ότι οι οδηγοί αυτών των οχημάτων θα χρειάζεται να φορτίζουν τα οχήματά τους στο σπίτι τους στο άμεσο μέλλον, μέχρι οι σταθμοί φόρτισης να είναι τόσοι που θα ικανοποιούν τις ανάγκες των οδηγών. Καθώς η μπαταρία απαιτεί εκτεταμένη χρονική περίοδο για επαναφόρτιση, περιορίζεται η χρήση των ηλεκτρικών οχημάτων σε μικρές μόνο αποστάσεις. Οι κατασκευάστριες εταιρείες προσπαθούν να ξεπεράσουν το πρόβλημα της μικρής αυτονομίας με σταθμούς γρήγορης φόρτισης όπου ένα όχημα μπορεί να φορτιστεί κατά ένα μεγάλο ποσοστό σε λίγα μόνο λεπτά. Σε σύγκριση με τα βενζινοκίνητα οχήματα, ένα ηλεκτρικό όχημα απαιτεί περισσότερο χρόνο για ανεφοδιασμό δηλαδή επαναφόρτιση και ο σταθμός γρήγορης φόρτισης κοστίζει περισσότερο για να λειτουργήσει. Αυτά τα εγγενή προβλήματα, σε συνδυασμό με την έλλειψη υποδομής επαναφόρτισης, αναστέλλουν σε μεγάλο βαθμό την ευρεία υιοθέτηση ηλεκτρικών οχημάτων. Το άγχος της εμβέλειας (ένας οδηγός μπορεί να φοβάται ότι το όχημα θα αποφορτιστεί πλήρως πριν φτάσει στον προορισμό) αποτελεί σημαντικό εμπόδιο για τη διείσδυση των ηλεκτρικών οχημάτων στην αγορά.

Το ευφυές δίκτυο είναι το κλειδί για την επίλυση αυτών των προβλημάτων, αφού με εφαρμογή έξυπνης φόρτισης μπορεί να σταθεροποιήσει το φορτίο στο δίκτυο. Παρέχει λοιπόν τις μετρήσεις και τον έλεγχο που απαιτούνται για τον μετριασμό των επιπτώσεων υψηλού φορτίου και την προστασία των εξαρτημάτων του δικτύου διανομής από την υπερφόρτωση λόγω φόρτισης ηλεκτρικών οχημάτων διασφαλίζοντας έτσι την αποτελεσματικότερη παραγωγική ικανότητα παραγωγής ηλεκτρικής ενέργειας. Σε ένα ευφυές δίκτυο, τα βοηθητικά προγράμματα μπορούν να διαχειριστούν το πότε και τον τρόπο με τον οποίο πραγματοποιείται η φόρτιση ενός ηλεκτρικού οχήματος ενώ εξακολουθούν να ακολουθούν και τις προτιμήσεις των πελατών. Η ενσωμάτωση των

ευφυών δικτύων επιτρέπει επίσης στα βοηθητικά προγράμματα να παρέχουν στους καταναλωτές περισσότερες πληροφορίες σχετικά με την εμπειρία τους στην ηλεκτροκίνηση - επιτρέποντας τους να κατανοήσουν καλύτερα το κόστος της τροφοδοσίας, τις θετικές και αρνητικές επιπτώσεις στο περιβάλλον, και τις δυνατότητες γύρω από τον καθορισμό των προτιμήσεων φόρτισης. Το ευφυές δίκτυο επιτρέπει επίσης στις επιχειρήσεις κοινής ωφελείας να συλλέγουν δεδομένα μετρητών ειδικά για τα ηλεκτρικά οχήματα, να προσφέρουν διαφοροποιημένες τιμές για φόρτιση ηλεκτρικών οχημάτων, και να προσελκύουν περισσότερους καταναλωτές προσφέροντας τα κατάλληλα κίνητρα. Λόγω της ισχυρής υποδομής επικοινωνιών που συνεπάγεται το ευφυές δίκτυο, τα βοηθητικά προγράμματα μπορούν να παρακολουθούν εξ αποστάσεως τους σταθμούς φόρτισης και να επιτρέπουν την ολοκληρωμένη διαχείριση της φόρτισης. Μπορούν επίσης να επιλύσουν προβλήματα φόρτισης και απορίες των καταναλωτών χωρίς να χρειάζονται περιττές κλήσεις υπηρεσιών με φυσικούς αντιπροσώπους.

Η ενσωμάτωση των ηλεκτρικών οχημάτων στο ευφυές δίκτυο επιτρέπει τη χρήση των μπαταριών των ηλεκτρικών οχημάτων ως μια μεγάλη καταναλωμένη αποθήκη ενέργειας που χρησιμεύει στην εξισορρόπηση της παραγωγής και της διακύμανσης φορτίου. Οι μπαταρίες μπορούν να φορτιστούν όταν υπάρχουν διαθέσιμες ανανεώσιμες πηγές ενέργειας στο δίκτυο και να προσφέρουν ενέργεια στο δίκτυο όταν οι ανανεώσιμες δεν προσφέρουν. Με αυτό τον τρόπο, τα ηλεκτρικά οχήματα βοηθούν την ενεργειακή βιομηχανία να βελτιστοποιήσει τη διαχείριση φορτίου των δικτύων τους.

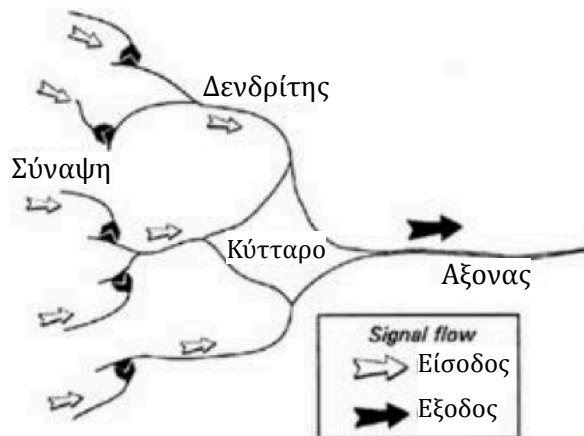
Ωστόσο, η ανάπτυξη σύγχρονων σταθμών φόρτισης και η ένταξη τους στο ευφυές δίκτυο, συνώνυμο της επικράτησης των ηλεκτρικών οχημάτων στην αγορά, απαιτεί ακόμη αρκετή έρευνα για να γίνει δυνατή η μέγιστη εκμετάλλευση της ενέργειας που προσφέρουν οι ανανεώσιμες πηγές ενέργειας, και να ελαχιστοποιηθεί η καύση υδρογονανθράκων. Ένας σταθμός φόρτισης θα πρέπει να έχει τη δυνατότητα να προβλέπει τη μελλοντική ζήτηση σε ενέργεια από τους οδηγούς ηλεκτρικών οχημάτων, έτσι ώστε να προσαρμόζει τις τιμές πώλησης κατάλληλα έτσι ώστε να κερδίζουν και οι δύο πλευρές, σταθμός και οδηγοί, και παράλληλα να εξασφαλίζεται η κάλυψη των αναγκών φόρτισης κάτω από οποιοσδήποτε συνθήκες.

## Κεφάλαιο 3

### Θεωρητικό Υπόβαθρο

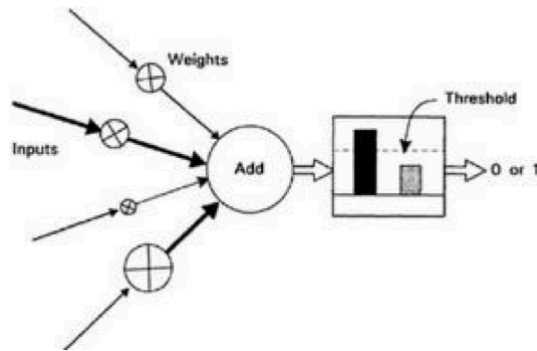
#### 3.1 Νευρωνικά δίκτυα

Με στοιχεία από το [13], ένα νευρωνικό δίκτυο ορίζεται ως ένα διασυνδεδεμένο σύνολο απλών στοιχείων επεξεργασίας, μονάδων ή κόμβων, των οποίων η λειτουργικότητα μοιάζει με τον ανθρώπινο νευρώνα. Η ικανότητα επεξεργασίας ενός τέτοιου δικτύου βρίσκεται στα σημεία σύνδεσης, ή στα βάρη, που λαμβάνουν τιμές μετά από μια διαδικασία προσαρμογής ή μάθησης από ένα σύνολο προτύπων εκπαίδευσης. Ο ανθρώπινος εγκέφαλος αποτελείται από περίπου 100 δισεκατομμύρια νευρικά κύτταρα ή νευρώνες, του οποίου ένα απλοποιημένο παράδειγμα φαίνεται στο Σχήμα 3.1.



Σχήμα 3.1 Βασικά συστατικά ενός νευρώνα σε απλοποιημένη μορφή.

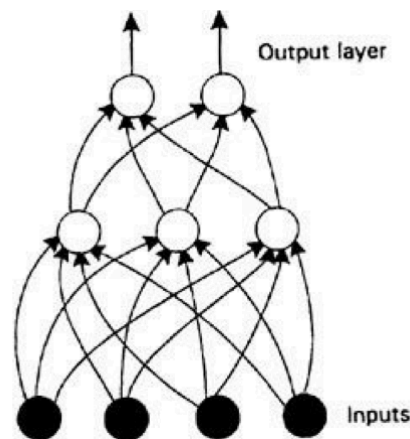
Οι νευρώνες επικοινωνούν μέσω ηλεκτρικών σημάτων, που είναι μικρής διάρκειας και μεγάλου πλάτους ακμής στην καμπύλη της τάσης του κυτταρικού τοιχώματος ή της μεμβράνης. Οι συνδέσεις μεταξύ των νευρώνων δημιουργούνται από ηλεκτροχημικές συνδέσεις-κόμβους που ονομάζονται συνάψεις, οι οποίες βρίσκονται σε κλάδους του κυττάρου που αναφέρονται ως δενδρίτες. Κάθε νευρώνας έχει συνήθως πολλές χιλιάδες συνδέσεις με άλλους νευρώνες και συνεπώς λαμβάνει συνεχώς ένα πλήθος εισερχόμενων σημάτων, τα οποία τελικά φτάνουν στο κύτταρο. Εκεί, ενσωματώνονται ή αθροίζονται, με κάποιο τρόπο και, κατά κανόνα, εάν το προκύπτον σήμα υπερβεί ένα συγκεκριμένο όριο, τότε ο νευρώνας θα "πυροδοτήσει" ή θα δημιουργήσει μια τάση ώθησης ως απάντηση. Αυτό μετά μεταδίδεται σε άλλους νευρώνες μέσω μιας ίνας διακλάδωσης γνωστή ως άξονας. Κατά τον προσδιορισμό του κατά πόσον μια ώθηση πρέπει να παράγεται ή όχι, ορισμένα εισερχόμενα σήματα παράγουν ένα ανασταλτικό αποτέλεσμα και τείνουν να αποτρέπουν την πυροδότηση τάσης ώθησης, ενώ άλλα είναι διεγερτικά και προάγουν την παραγωγή ώσεων. Η διακριτική ικανότητα επεξεργασίας κάθε νευρώνα λοιπόν εξαρτάται από τον τύπο - διεγερτικό ή ανασταλτικό - και τη δύναμη των συναπτικών του συνδέσεων με άλλους νευρώνες.



Σχήμα 3.2 Απλός τεχνητός νευρώνας.

Τα τεχνητά ισοδύναμα των βιολογικών νευρώνων είναι οι κόμβοι ή οι μονάδες όπως φαίνεται στο Σχήμα 3.2. Οι συνάψεις μοντελοποιούνται από έναν μόνο αριθμό ή βάρος, έτσι ώστε κάθε είσοδος να πολλαπλασιάζεται με ένα ακριβώς βάρος πριν αποσταλεί στο ισοδύναμο του κυττάρου. Σε αυτό το παράδειγμα, τα σταθμισμένα σήματα αθροίζονται με απλή πρόσθεση για να ενεργοποιήσουν έναν κόμβο. Στον τύπο του κόμβου που φαίνεται στο Σχήμα 3.2 - που ονομάζεται μονάδα επιλογής κατωφλίου (threshold logic unit ή TLU) - η τιμή της ενεργοποίησης συγκρίνεται με ένα όριο. Εάν η τιμή αυτή υπερβεί το κατώφλι, η μονάδα παράγει έξοδο 1, αλλιώς παράγει 0. Στο σχήμα, το μέγεθος των σημάτων αντιπροσωπεύεται από το πλάτος των αντίστοιχων βελών τους. Τα βάρη εμφανίζονται ως σύμβολα πολλαπλασιασμού ( $X$  μέσα σε κύκλο) και οι τιμές τους είναι ανάλογες με το μέγεθος του συμβόλου. Σε αυτό το παράδειγμα έχουν χρησιμοποιηθεί μόνο θετικά βάρη. Το TLU είναι το απλούστερο μοντέλο ενός τεχνητού νευρώνα.

Ο όρος "δίκτυο" θα χρησιμοποιείται για αναφορά σε οποιοδήποτε σύστημα τεχνητών νευρώνων. Αυτό μπορεί να αφορά κάτι πολύ απλό όπως ένας απλός κόμβος, έως και κάτι πολύ σύνθετο όπως μια μεγάλη συλλογή κόμβων στους οποίους ο καθένας συνδέεται με κάθε άλλο κόμβο στο δίκτυο.



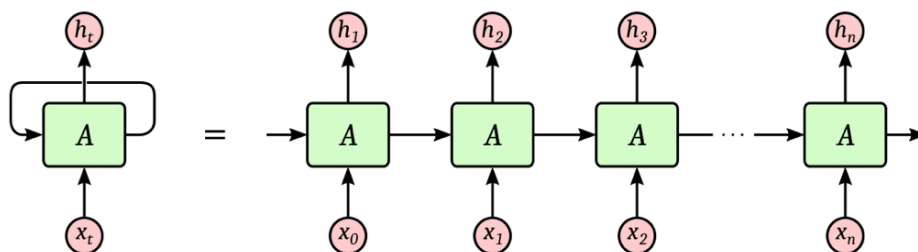
Σχήμα 3.3: Ένας απλός τύπος νευρωνικού δικτύου

Κάθε κόμβος αναπαριστάται τώρα από έναν μόνο κύκλο, αλλά τα βάρη εφαρμόζονται σε όλες τις συνδέσεις. Οι κόμβοι είναι διατεταγμένοι σε μια πολυεπίπεδη δομή στην οποία κάθε σήμα

προέρχεται από μια είσοδο και περνά μέσω δύο κόμβων πριν φτάσει σε μια έξοδο πέρα από την οποία δεν α πλέον. Αυτός ο τύπος εμπροσθοτροφοδοτούμενου δικτύου είναι μόνο ένα παράδειγμα από τα πολλά διαθέσιμα, και χρησιμοποιείται συνήθως για να κατατάξει ένα μοτίβο εισόδου σε μία από πολλές κλάσεις σύμφωνα με το προκύπτον μοτίβο εξόδων. Για παράδειγμα, εάν η είσοδος αποτελείται από μια κωδικοποίηση από εικόνες χειρόγραφων γραμμάτων, το επίπεδο εξόδου (πάνω στην εικόνα) μπορεί να περιέχει 24 κόμβους - έναν για κάθε γράμμα του αλφαβήτου - για να αναγνωρίζεται σε ποιο γράμμα-κλάση ανήκει η κάθε είσοδος. Αυτό θα μπορούσε να γίνει με την κατανομή ενός κόμβου εξόδου ανά κλάση και απαιτώντας ότι μόνο ένας από αυτούς τους κόμβους να ενεργοποιείται κάθε φορά που παρέχεται ένα γράμμα της αντίστοιχης κλάσης στην είσοδο.

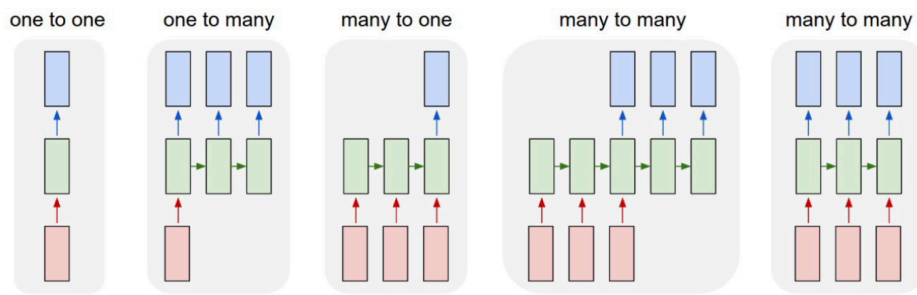
### 3.1 Αναδρομικά Νευρωνικά δίκτυα

Η ανάγκη που δημιούργησε τα Αναδρομικά Νευρωνικά Δίκτυα [9] ή RNNs είναι η ικανότητα για εκμετάλλευση διαδοχικών πληροφοριών. Σε ένα παραδοσιακό νευρωνικό δίκτυο υποθέτουμε ότι όλες οι εισοδοί και οι έξοδοί είναι ανεξάρτητες μεταξύ τους. Υπάρχουν όμως πολλές εφαρμογές στις οποίες κάτι τέτοιο δεν ισχύει για τα δεδομένα. Όταν χρειάζεται να προβλέψουμε την επόμενη λέξη σε μια πρόταση, αυτή η πρόβλεψη εξαρτάται σε μεγάλο βαθμό από τις λέξεις που βρίσκονται στις αμέσως προηγούμενες θέσεις. Τα RNN ονομάζονται αναδρομικά επειδή εκτελούν την ίδια εργασία για κάθε στοιχείο μιας ακολουθίας, με την έξοδο να εξαρτάται από τους προηγούμενους υπολογισμούς, ενώ διαθέτουν μια μορφή «μνήμης» που συγκρατεί πληροφορίες σχετικά με αυτό που έχει υπολογιστεί ως τώρα. Η βασική μορφή φαίνεται στο Σχήμα 3.4



Σχήμα 3.4 Ένα ξεδιπλωμένο αναδρομικό νευρωνικό δίκτυο (RNN)

Ο βασικός λόγος που τα RNNs είναι πιο χρήσιμα για πολλές εφαρμογές είναι ότι επιτρέπουν να χρησιμοποιούμε ακολουθίες διανυσμάτων (vectors), στην είσοδο, την έξοδο ή συνήθως και στις δύο. Υπάρχουν διαφορετικοί τύποι, όπως φαίνεται και στο Σχήμα 3.5.



Σχήμα 3.5: Διαφορετικοί τύποι αναδρομικών νευρωνικών δικτύων (RNN)

Κάθε ένα παραλληλόγραμμο στο παραπάνω σχήμα συμβολίζει διανύσματα (vectors) ενώ τα βέλη συμβολίζουν τις συναρτήσεις. Τα κόκκινα παραλληλόγραμμα αντιπροσωπεύουν την είσοδο και τα μπλε την έξοδο.

Ένα προς ένα: (one to one):

Ονομάζεται επίσης και Plain ή Vanilla Neural network. Χρησιμοποιείται σε προβλήματα όπου τόσο το μέγεθος της εισόδου όσο και την εξόδου είναι σταθερά, και τα δεδομένα δεν εξαρτώνται από προηγούμενα δεδομένα ή έξοδο.

π.χ.: Για ταξινόμηση εικόνων

Ένα προς πολλά (one to many):

Χρησιμοποιείται σε προβλήματα όπου το μέγεθος της εισόδου είναι σταθερό, ενώ η έξοδος είναι μια ακολουθία (sequence)

π.χ.: Image Captioning, δηλαδή μια εφαρμογή που παίρνει την εικόνα ως είσοδο και εξάγει μια σειρά από λέξεις.

Πολλά προς ένα (many to one):

Δέχεται ως είσοδο μια ακολουθία και δίνει έξοδο σταθερού μεγέθους.

π.χ.: Ανάλυση συναισθημάτων όπου μια δεδομένη πρόταση ταξινομείται ως έκφραση θετικού ή αρνητικού συναισθήματος.

Πολλά προς πολλά (many to many):

Δέχεται μια ακολουθία δεδομένων ως είσοδο και δίνει ως έξοδο μια ακολουθία δεδομένων.

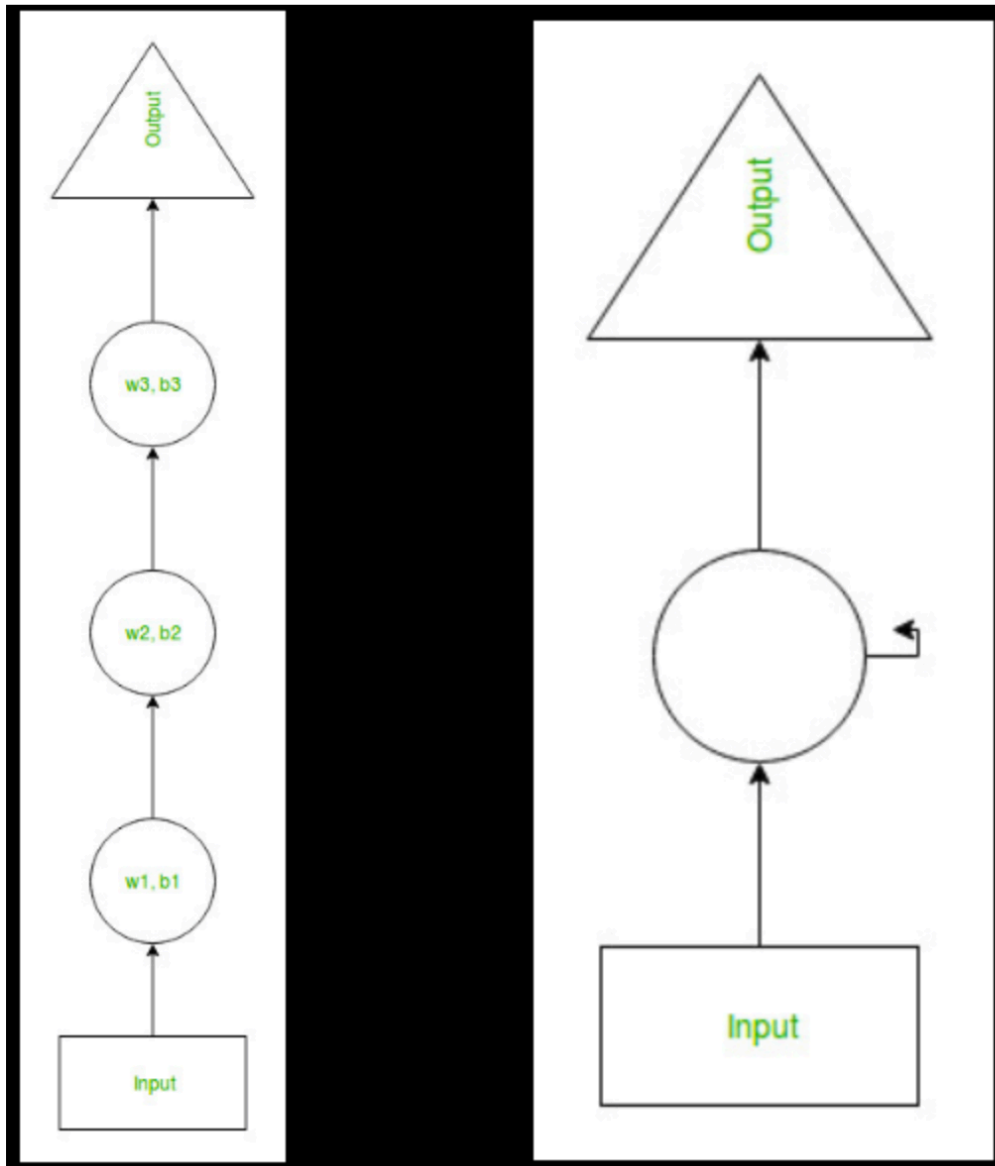
π.χ.: Μηχανική μετάφραση, όπου το RNN διαβάζει μια πρόταση στα Αγγλικά και στη συνέχεια εξάγει μια πρόταση στα ελληνικά.

Διπλής κατεύθυνσης πολλά προς πολλά (Bidirectional Many-to-Many):

Συγχρονισμένη ακολουθία εισόδου και εξόδου. Δεν υπάρχουν προκαθορισμένες τιμές που να περιορίζουν το μήκος των ακολουθιών επειδή ο επαναλαμβανόμενος μετασχηματισμός είναι σταθερός και μπορεί να εφαρμοστεί όσες φορές χρειάζεται.

π.χ .: Ταξινόμηση βίντεο όπου χρειάζεται να ταξινομηθεί κάθε καρέ του βίντεο.

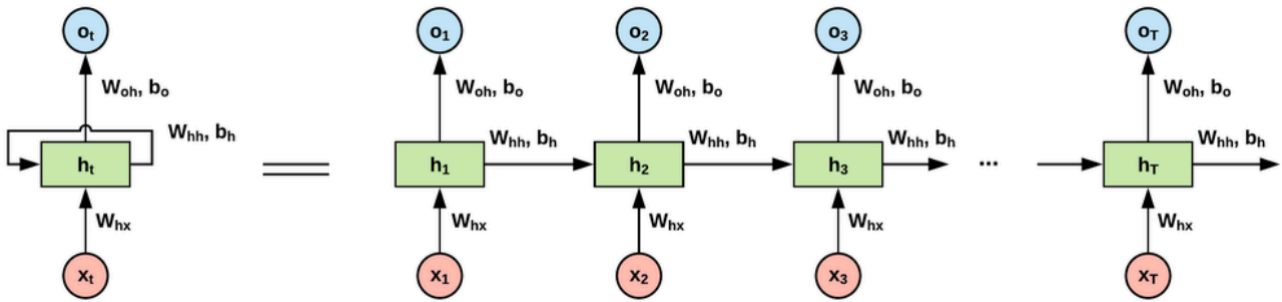
Ένα απλό Νευρωνικό Δίκτυο αποτελείται από τη μονάδα της εισόδου, κάποια κρυφά επίπεδα και τις μονάδες της εξόδου που επεξεργάζονται πληροφορίες ανεξάρτητα, και χωρίς συσχετισμό με προηγούμενες εξόδους. Επίσης, στο νευρωνικό του παρακάτω σχήματος δώσαμε διαφορετικά βάρη και όρους bias στις κρυμμένες μονάδες, αποτρέποντας έτσι οποιαδήποτε απομνημόνευση πληροφορίας.



Σχήμα 3.6: Ένα απλός τεχνητός νευρώνας στα αριστερά, συγκρινόμενος με ένα RNN στα δεξιά[14]

Από την άλλη, τα κρυμμένα επίπεδα των RNN περιλαμβάνουν ίδια βάρη ( $W_{hh}$ ) και ίδιες τιμές bias ( $b_h$ ) επιτρέποντας έτσι να απομνημονεύουν πληροφορίες που έχουν επεξεργαστεί, όπως φαίνεται στο σχήμα 3.7





Σχήμα 3.7: Τα κρυφά επίπεδα ενός RNN

Το  $h_t$  αποτυπώνει την τρέχουσα χρονική στιγμή και ορίζεται από την εξίσωση:

$$h_t = f(h_{t-1}, x_t)$$

όπου  $o_t$  είναι η κατάσταση εξόδου,  $h_t$  η τρέχουσα χρονική στιγμή,  $h_{t-1}$  η προηγούμενη χρονική στιγμή και το  $x_t$  η κατάσταση εισόδου.

Η συνάρτηση ενεργοποίησης ορίζεται ως εξής:

$$h_t = \tanh(W_{hh}h_{t-1} + W_{hx}x_t)$$

Όπου  $W$  είναι το βάρος, το  $h$  είναι το απλό κρυφό διάνυσμα, το  $W_{hh}$  είναι το βάρος στην προηγούμενη κρυφή κατάσταση, και το  $W_{hx}$  είναι το βάρος στην τρέχουσα κατάσταση εισόδου.

Το  $\tanh$  είναι η λειτουργία ενεργοποίησης, που εφαρμόζει μια μη γραμμική σχέση η οποία αναπροσαρμόζει τις τιμές στο εύρος  $[-1, 1]$

Η έξοδος είναι της μορφής:

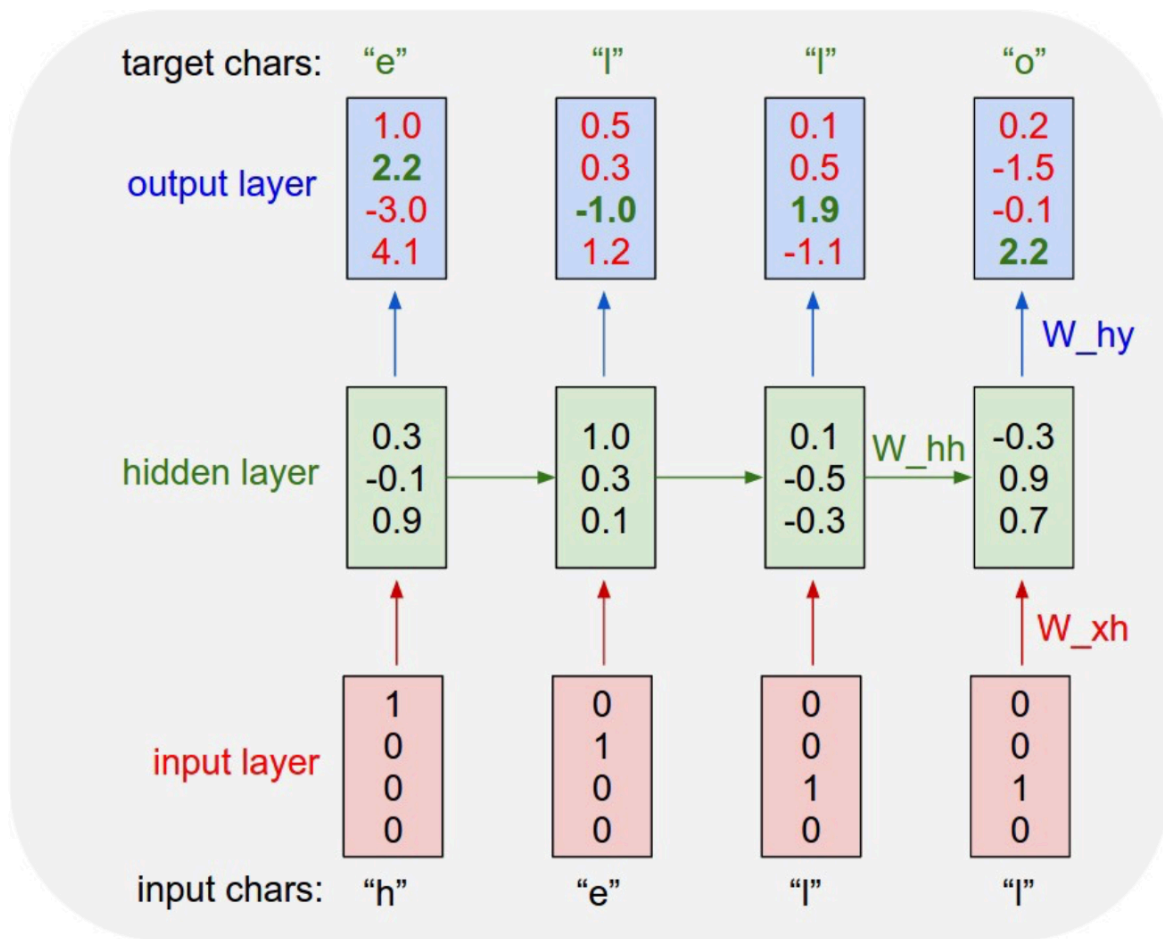
$$y_t = W_{hy}h_t$$

Όπου  $y_t$  είναι η κατάσταση εξόδου και  $W_{hy}$  είναι το βάρος στην κατάσταση εξόδου.

Στη συνέχεια ακολουθεί ένα απλό παράδειγμα, για την πρόβλεψη του επόμενου χαρακτήρα δεδομένης μιας ακολουθίας από χαρακτήρες. Θα δώσουμε στο RNN ένα τεράστιο κομμάτι κειμένου και θα του ζητήσουμε να μοντελοποιήσει την πιθανότητα του επόμενου χαρακτήρα δοσμένης μιας ακολουθίας από προηγούμενους χαρακτήρες.

Ως ένα απλό παράδειγμα, υποθέτουμε ότι έχουμε ένα λεξιλόγιο τεσσάρων πιθανών γραμμάτων "helo" και θέλουμε να εκπαιδεύσουμε το RNN στην ακολουθία "hello". Αυτή η εκπαιδευτική ακολουθία είναι στην πραγματικότητα μια πηγή 4 ξεχωριστών παραδειγμάτων εκπαίδευσης:

1. Το "e" είναι πιθανό να εμφανιστεί μετά από το "h"
2. Το "l" είναι πιθανό να εμφανιστεί μετά από το "he"
3. Το "o" είναι πιθανό να εμφανιστεί μετά από το "hel"
4. Και ότι το "o" είναι πιθανό να εμφανιστεί μετά από το "hell"



Σχήμα 3.8: Τα επίπεδα του RNN στο παράδειγμα πρόβλεψης χαρακτήρα

### Οπισθοδιάδοση

Για να κατανοήσουμε και να οπτικοποιήσουμε την οπισθοδιάδοση χρειάζεται να ξετυλίξουμε το δίκτυο σε όλες τις χρονικές στιγμές για να καταλάβουμε πώς ενημερώνεται το βάρος. Η επιστροφή στην προηγούμενη χρονική στιγμή για ενημέρωση των βαρών ονομάζεται οπισθοδιάδοση ή Backpropagation.

Συνήθως αντιμετωπίζουμε την πλήρη ακολουθία της εισόδου (λέξη) ως ένα παράδειγμα εκπαίδευσης, οπότε το συνολικό σφάλμα είναι το άθροισμα των σφαλμάτων σε κάθε βήμα (χαρακτήρας). Τα βάρη όπως μπορούμε να δούμε είναι τα ίδια σε κάθε βήμα. Αν συνοψίσουμε τα βήματα για την αντίστροφη διάδοση στο χρόνο, τότε το σφάλμα εγκάρσιας εντροπίας υπολογίζεται πρώτα χρησιμοποιώντας την τρέχουσα έξοδο και την πραγματική έξοδο. Εφόσον όμως το δίκτυο είναι ξετυλιγμένο για όλα τα βήματα του χρόνου, για το μη καταγεγραμμένο δίκτυο, η κλίση υπολογίζεται για κάθε βήμα ανάλογα με την παράμετρο βάρους. Αφού το βάρος είναι το ίδιο για όλα τα βήματα, οι κλίσεις μπορούν να συνδυαστούν για όλα τα χρονικά βήματα. Στη συνέχεια, τα βάρη ενημερώνονται τόσο για τον αναδρομικό νευρώνα όσο και για τα πυκνά επίπεδα.

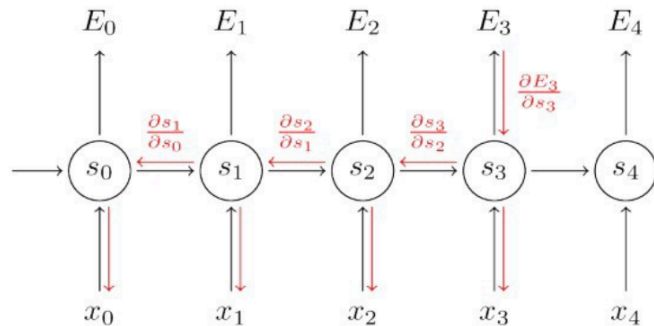
$$\frac{\partial E}{\partial \mathbf{W}} = \sum_t \frac{\partial E_t}{\partial \mathbf{W}}$$

$$\frac{\partial E_3}{\partial \mathbf{W}} = \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \frac{\partial s_3}{\partial \mathbf{W}}$$

But  $s_3 = \tanh(Ux_t + Ws_2)$

**S<sub>3</sub> depends on s<sub>2</sub>, which depends on W and s<sub>1</sub>, and so on.**

$$\frac{\partial E_3}{\partial \mathbf{W}} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \frac{\partial s_3}{\partial s_k} \frac{\partial s_k}{\partial \mathbf{W}}$$



Σχήμα 3.9: Η οπισθοδιάδοση στο RNN

Σημείωση: Η επιστροφή σε κάθε χρονική στιγμή και η ενημέρωση των βαρών της είναι μια πολύ αργή διαδικασία που χρειάζεται τόσο υπολογιστική δύναμη όσο και χρόνο.

Η οπισθοδιάδοση αντιμετωπίζει δύο κύρια προβλήματα:

- Την εξαφανιζόμενη κλίση (vanishing gradient)
- Και την έκρηξη κλίσης (exploding gradient)

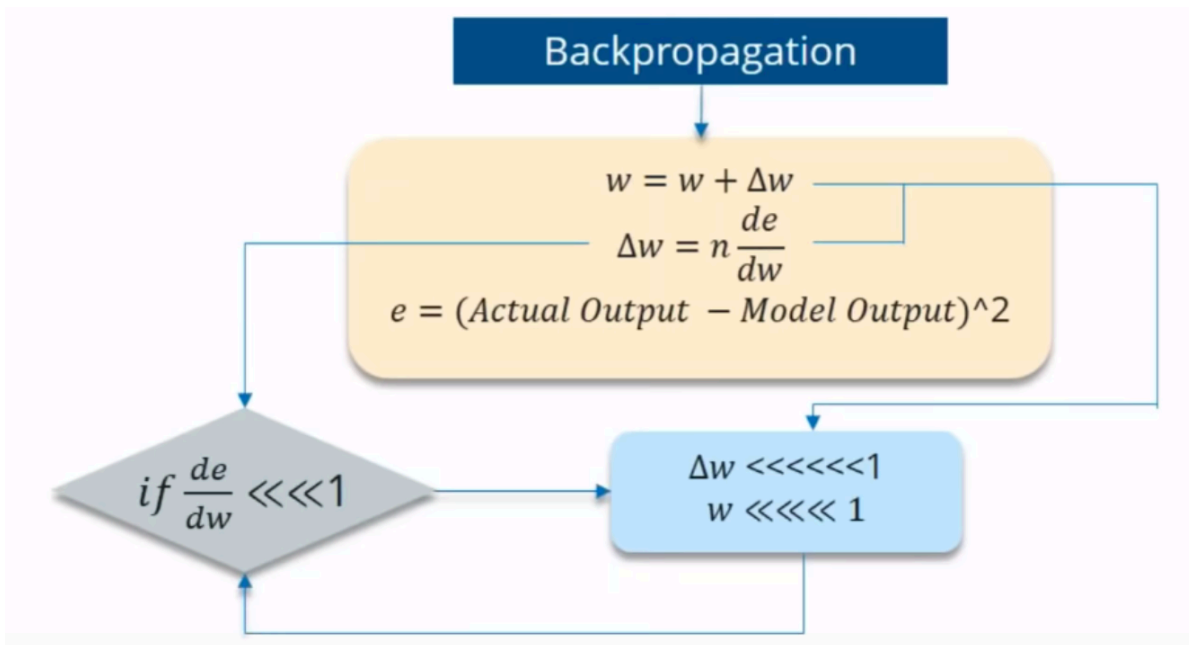
Εξαφανιζόμενη κλίση (vanishing gradient):

Σύμφωνα με το [15], καθένα από τα βάρη του νευρωνικού δικτύου λαμβάνει μια ενημέρωση ανάλογη με τη μερική παράγωγο της συνάρτησης σφάλματος σε σχέση με το τρέχον βάρος σε κάθε επανάληψη της εκπαίδευσης. Το πρόβλημα είναι ότι σε ορισμένες περιπτώσεις, η κλίση θα είναι πολύ μικρή, εμποδίζοντας αποτελεσματικά το βάρος να αλλάξει την τιμή του. Στη χειρότερη περίπτωση, αυτό μπορεί να σταματήσει εντελώς το νευρωνικό δίκτυο από την περαιτέρω εκπαίδευση. Ως ένα παράδειγμα της αιτίας του προβλήματος, οι παραδοσιακές συναρτήσεις ενεργοποίησης, όπως η συνάρτηση υπερβολικής εφαπτομένης (tanh), έχουν κλίσεις στο εύρος (0, 1) και η οπισθοδιάδοση υπολογίζει τις κλίσεις από τον κανόνα της αλυσίδας. Αυτό έχει ως αποτέλεσμα τον πολλαπλασιασμό του n από αυτούς τους μικρούς αριθμούς για τον υπολογισμό των βαθμίδων των πρώτων στρωμάτων σε ένα δίκτυο n επιπέδων, πράγμα που σημαίνει ότι η κλίση μειώνεται εκθετικά με το n ενώ τα αρχικά επίπεδα εκπαιδεύονται πολύ αργά.

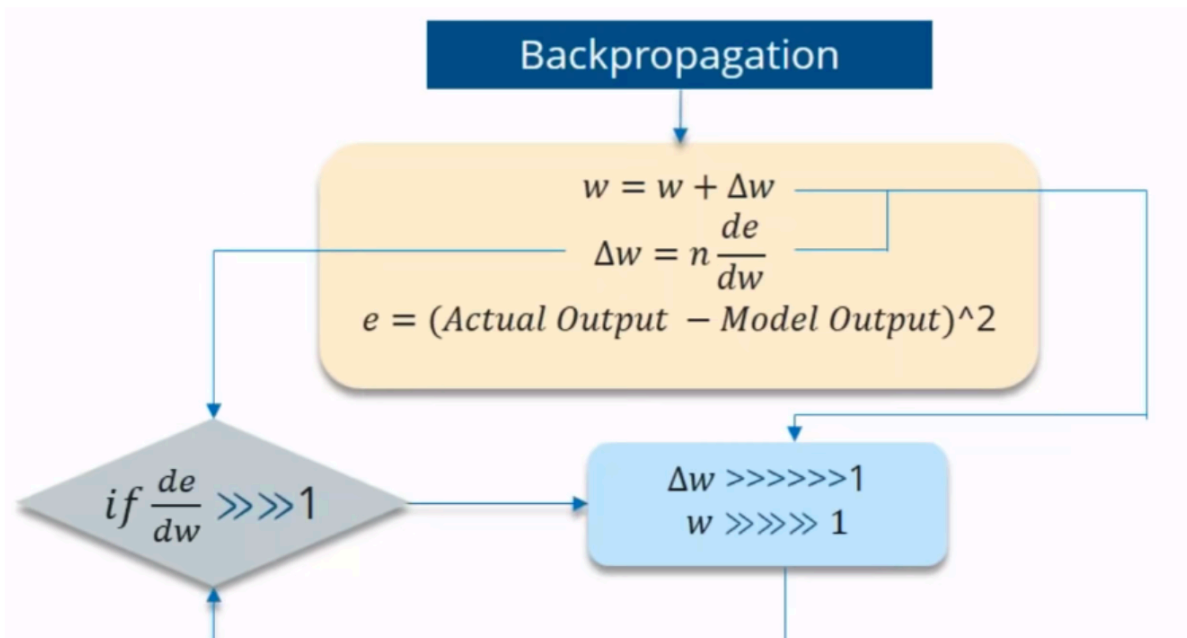
- έκρηξη κλίσης (exploding gradient)

Η έκρηξη κλίσης [16] είναι ένα πρόβλημα που εμφανίζεται όταν συσσωρεύονται μεγάλες κλίσεις σφαλμάτων και οδηγούν σε πολύ μεγάλες ενημερώσεις στα βάρη του νευρωνικού δικτύου κατά τη διάρκεια της εκπαίδευσης. Οι κλίσεις χρησιμοποιούνται κατά τη διάρκεια της εκπαίδευσης για την ενημέρωση των βαρών του δικτύου, αλλά συνήθως αυτή η διαδικασία λειτουργεί καλύτερα όταν αυτές οι ενημερώσεις είναι μικρές και ελεγχόμενες. Όταν συσσωρεύονται τα μεγέθη των κλίσεων,

είναι πιθανό να εμφανιστεί ένα ασταθές δίκτυο, το οποίο μπορεί να προκαλέσει κακά αποτελέσματα δυσκολίας ή ακόμη και ένα μοντέλο που δεν εκπαιδεύεται ποτέ.



Σχήμα 3.10: Το πρόβλημα της εξαφανιζόμενης κλίσης (vanishing gradient)



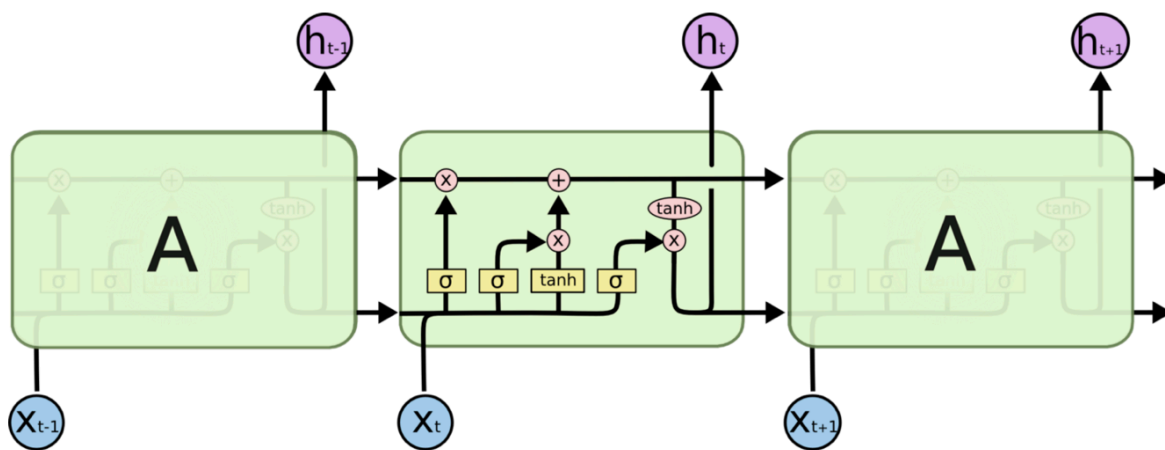
Σχήμα 3.11: Το πρόβλημα της έκρηξης κλίσης (exploding gradient)

Πλεονεκτήματα των αναδρομικών νευρωνικών δικτύων:

Το κύριο πλεονέκτημα του RNN έναντι των άλλων τεχνητών νευρωνικών δικτύων (Artificial Neural Networks ή NN) είναι ότι το RNN μπορεί να μοντελοποιήσει μία ακολουθία δεδομένων (δηλαδή χρονικές σειρές) έτσι ώστε κάθε δείγμα να θεωρείται ότι εξαρτάται από προηγούμενα

Το κύριο μειονέκτημα των αναδρομικών νευρωνικών δικτύων είναι τα προβλήματα εξαφανιζόμενης κλίσης και έκρηξης κλίσης.

### 3.2 Δίκτυα μακράς και βραχείας μνήμης LSTM

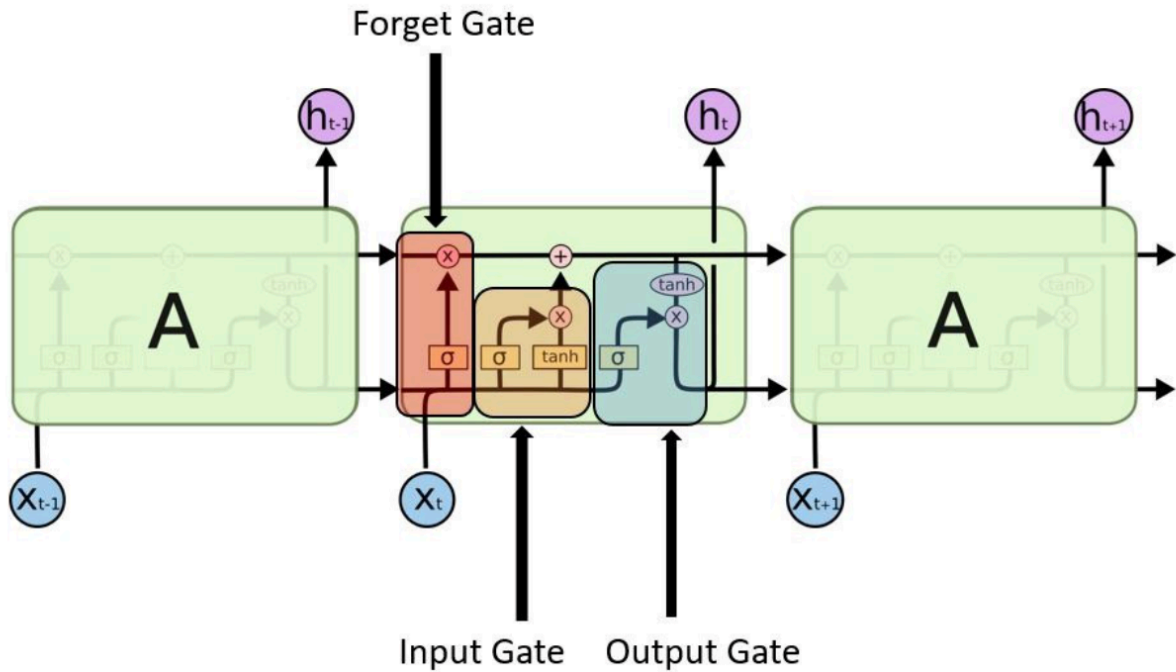


Σχήμα 3.12: Η επαναλαμβανόμενη μονάδα του LSTM και οι πύλες της.

Μια ειδική κατηγορία νευρωνικών δικτύων που ανήκουν στην ευρύτερη κατηγορία των αναδρομικών νευρωνικών δικτύων (RNN) είναι τα δίκτυα μακράς και βραχείας μνήμης (Long short term memory ή LSTM)

Το LSTM υλοποιεί μια διαδικασία τριών βημάτων:

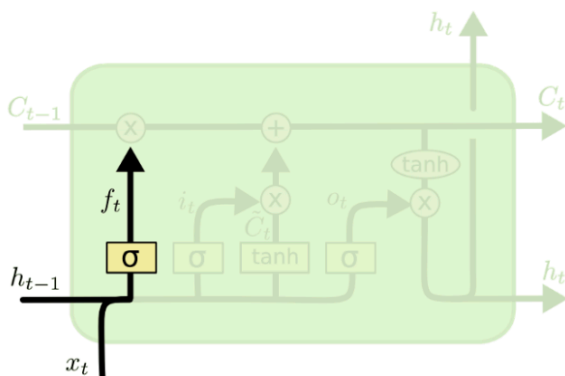
Κάθε μονάδα LSTM έχει 3 πύλες που ονομάζονται: Forget gate, Input gate και Output gate όπως φαίνονται στο Σχήμα 3.13.



Σχήμα 3.13: Οι πύλες του LSTM

Πύλη forget:

Αυτή η πύλη αποφασίζει ποια δεδομένα από το παρελθόν πρέπει να κρατηθούν και ποια να παραλειφθούν τη συγκεκριμένη χρονική στιγμή. Χρησιμοποιεί τη σιγμοειδή συνάρτηση για την απόφαση αυτή. Ελέγχει την προηγούμενη κατάσταση  $h_{t-1}$  και το περιεχόμενο της τωρινής εισόδου  $x_t$  και δίνει ως έξοδο έναν αριθμό μεταξύ του 0 (απέρριψε) και του 1 (κράτα την πληροφορία) για κάθε αριθμό μέσα στο κελί στη κατάσταση  $c_{t-1}$ .

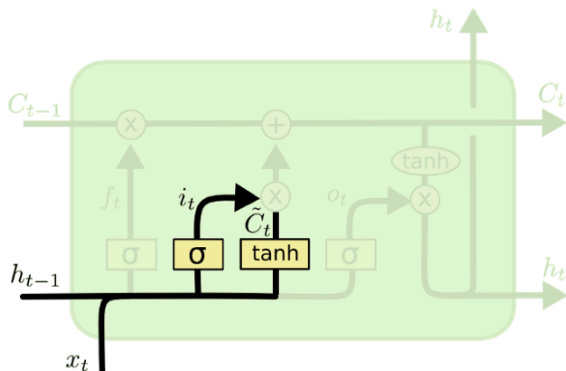


$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Σχήμα 3.14: Η πύλη forget του LSTM

Πύλη εισόδου / input gate:

Αποφασίζει πόσα δεδομένα που βρίσκονται σε αυτή την πύλη θα περάσουν στην τωρινή κατάσταση.



$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

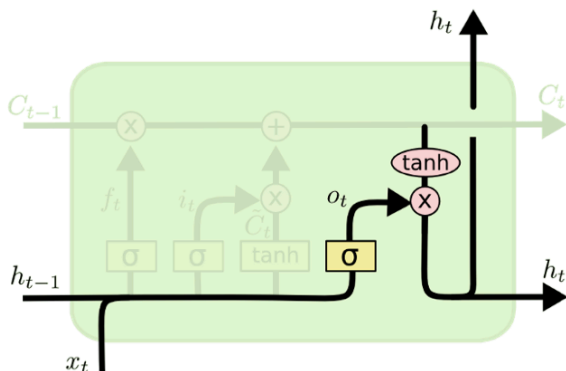
Σχήμα 3.15: Η πύλη εισόδου/input του LSTM

Η σιγμοειδής συνάρτηση αποφασίζει ποιες τιμές επιτρέπονται να περάσουν από το 0 έως το 1. Η συνάρτηση υπερβολικής εφαπτομένης tanh δίνει βάρη στις τιμές που περνούν καθορίζοντας το επίπεδο σημαντικότητάς τους που κυμαίνεται από -1 έως 1.

Πύλη εξόδου:

Αποφασίζει ποιο κομμάτι των δεδομένων που βρίσκονται στην πύλη θα φτάσει έως την έξοδο.

Η σιγμοειδής συνάρτηση αποφασίζει ποιες τιμές επιτρέπονται από το 0 έως το 1, ενώ η συνάρτηση υπερβολικής εφαπτομένης tanh ορίζει βάρη στις τιμές που περνούν αποφασίζοντας το επίπεδο σημαντικότητάς τους που κυμαίνεται από -1 έως 1 και και στη συνέχεια πολλαπλασιάζονται με την έξοδο της σιγμοειδούς συνάντησης.



$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

Σχήμα 3.16: Η πύλη εξόδου/output του LSTM

## Κεφάλαιο 4

### Δεδομένα

#### 4.1 Πηγές Δεδομένων

Ο τομέας της μοντελοποίησης φορτίου φόρτισης ηλεκτρικών οχημάτων έχει αναπτυχθεί ραγδαία την τελευταία δεκαετία. Η ερευνητική δημοσίευση [10] προσφέρει αναλυτικά αποτελέσματα γύρω από τα διαθέσιμα δεδομένα. Επηρεαζόμενοι από τη Συμφωνία του Παρισιού, είναι σημαντικό να συνεχίσουμε να ενθαρρύνουμε καλύτερες τεχνικές μοντελοποίησης για την επιτυχή υιοθέτηση των ηλεκτρικών οχημάτων. Ταυτόχρονα, πολλές επιστημονικές δημοσιεύσεις επισημαίνουν την έλλειψη διαθέσιμων δεδομένων σταθμών φόρτισης για τη δημιουργία μοντέλων που είναι συμβατά με την πραγματικότητα. Η ανοικτή αναζήτηση δεδομένων κάλυψε περισσότερα από 860 αποθετήρια και απέδωσε περίπου 60 σύνολα δεδομένων που σχετίζονται με τη μοντελοποίηση φορτίου φόρτισης ηλεκτρικών οχημάτων. Αυτά τα σύνολα δεδομένων περιλαμβάνουν πληροφορίες σχετικά με τις τοποθεσίες σημείων φόρτισης, ιστορικά και σε πραγματικό χρόνο φόρτιση, μετρήσεις κυκλοφορίας, ταξιδιωτικές έρευνες και ταξινομημένα οχήματα. Τα μοντέλα που εξετάστηκαν κυμαίνονται από στατιστικό χαρακτηρισμό έως στοχαστικές διαδικασίες και μηχανική μάθηση και αξιολογείται το πλαίσιο της εφαρμογής τους. Σύμφωνα με το [9] τα διαθέσιμα ιστορικά δεδομένα από συνεδρίες φορτίσεων ηλεκτρικών οχημάτων ταξινομημένα ανά χώρα έχουν ως εξής:

##### Ολλανδία

Στην Ολλανδία παρατηρήθηκε το 2020, 20,5% εθνικό μερίδιο αγοράς ηλεκτρικών οχημάτων [10] που αντιστοιχεί σε περίπου 71.000 ηλεκτρικά οχήματα.

Συνολικά, 23 σύνολα δεδομένων (datasets) καλύφθηκαν στις Κάτω Χώρες με κάθε τύπο σχετικών δεδομένων που υπήρχαν. Πρώτα απ' όλα, το ElaadNL [11] περιλαμβάνει ιστορικές συνεδρίες φόρτισης που έχουν χρησιμοποιηθεί για μελέτη σε πολλαπλές εργασίες, ωστόσο δεν είναι διαθέσιμα online, αλλά διατίθενται μετά από σχετική αίτηση στον πάροχο. Τα δεδομένα σε πραγματικό χρόνο σχετικά με τη χρήση και την κατανάλωση σε δημόσιους σταθμούς φόρτισης που βρίσκονται στο Ρότερνταμ βρίσκονται στον ιστότοπο EV-BOX [12], που είναι ένας από τους παρόχους σταθμών φόρτισης.

##### Η.Π.Α.

Στις Η.Π.Α. παρατηρήθηκε το 2019, 2,4% εθνικό μερίδιο αγοράς ηλεκτρικών οχημάτων που αντιστοιχεί σε περίπου 130.000 ηλεκτρικά οχήματα.

Η αναζήτηση ανοιχτών δεδομένων για τις ΗΠΑ ήταν εκτεταμένη. Περίπου 370 σύνολα δεδομένων βρέθηκαν συνολικά. Μεταξύ αυτών, βρέθηκαν τρία σύνολα δεδομένων που περιελάμβαναν ιστορικές συνεδρίες φόρτισης. Το πρώτο παρέχει συνεχή σύνολα δεδομένων από συνεδρίες φόρτισης από το 2018 που καταγράφηκαν σε σταθμούς φόρτισης που ανήκουν στην πόλη του Boulder στο Κολοράντο [17]. Το δεύτερο δίνει τις ίδιες πληροφορίες για τις συνεδρίες φόρτισης ηλεκτρικών οχημάτων στην πόλη Palo Alto της Καλιφόρνια από το 2011 έως το 2020. Επιπλέον, ένα ανοιχτό σύνολο δεδομένων συνεδριών φόρτισης είναι διαθέσιμο από την Caltech, το οποίο ενημερώνεται συνεχώς σε συνεργασία με την Power Flex, και είναι διαθέσιμο στο [18].



## Γαλλία

2,1% εθνικό μερίδιο αγοράς ηλεκτρικών οχημάτων που ισοδυναμεί με περίπου 46.000 ηλεκτρικά οχήματα να πωλήθηκαν το 2019 [121].

Η Γαλλία διαθέτει επίσης μεγάλο αριθμό ανοιχτών αποθετηρίων δεδομένων. Συνολικά, διερευνήθηκαν 151 σύνολα δεδομένων. Μεταξύ αυτών βρέθηκαν όλα τα είδη σχετικών δεδομένων. Πρώτον, συνεδρίες φόρτισης που καταγράφηκαν από τον Απρίλιο έως τον Μάιο του 2017 στους σταθμούς Belib στο Παρίσι [19]. Όσον αφορά τους ιδιωτικούς σταθμούς φόρτισης, οι συνεδρίες φόρτισης ενός ιδιωτικού στόλου ηλεκτρικών οχημάτων που ανήκουν στην SAP Labs France έχουν καταγραφεί από τον Ιούνιο του 2017 [20]. Αυτό το σύνολο δεδομένων ενημερώνεται κάθε τρεις μήνες. Πέρα από τα δεδομένα συνεδριών φόρτισης, είναι διαθέσιμα δεδομένα για οχήματα σε όλη την επικράτεια, μετρήσεις κυκλοφορίας σε πολλές πόλεις, κίνηση σε πραγματικό χρόνο και δεδομένα εθνικής ταξιδιωτικής έρευνας [100], που πραγματοποιήθηκε προκειμένου να πραγματοποιηθεί χωροχρονική ανάλυση φορτίου EV.

## Ηνωμένο Βασίλειο

2,1% εθνικό μερίδιο αγοράς ηλεκτρικών οχημάτων που ισοδυναμεί με περίπου 50.000 ηλεκτρικά οχήματα να πωλήθηκαν το 2019

Ερευνήθηκαν 72 σύνολα δεδομένων για την ηπειρωτική χώρα του Ηνωμένου Βασιλείου, τα οποία απέδωσαν πολλαπλά σύνολα δεδομένων περιόδων φόρτισης. Δύο από αυτά βρίσκονται στη Σκωτία: η πόλη Ντάντι [21] και το δημοτικό συμβούλιο του Perth and Kinross [22]. Το πρώτο συγκεντρώνει δύο έτη δεδομένων συνεδριών φόρτισης από το 2017 έως το 2018, ενώ το δεύτερο καλύπτει τέσσερα έτη από το 2016 έως το 2019 και με περισσότερη λεπτομέρεια κάθε συνεδρία.

## 4.2 Επεξεργασία των δεδομένων

Μετά από διερεύνηση των δεδομένων, το μεγαλύτερο σύνολο δεδομένων όσον αφορά τις καταχωρήσεις και την περίοδο κάλυψης είναι διαθέσιμο από τη Βιβλιοθήκη Ανοιχτών Δεδομένων της Πόλης του Palo Alto στην Καλιφόρνια και περιέχει δεδομένα από τον Ιούλιο του 2011 και μέχρι τον Δεκέμβριο του 2020, και ενημερώνεται ανά τρίμηνο. Το σύνολο δεδομένων περιέχει σχεδόν 260.000 μεμονωμένες συνεδρίες φόρτισης που χρησιμοποιούνται για την εκπαίδευση και την αξιολόγηση της απόδοσης του μοντέλου μας. Όλες οι συνεδρίες χαρακτηρίζονται από τη χρονική σήμανση άφιξης και αναχώρησης του ηλεκτρικού οχήματος και σε kWh της ενέργειας που καταναλώσε κατά τη φόρτιση. Μαζί με αυτά, κάθε συνεδρία χαρακτηρίζεται επίσης από αριθμό βύσματος (plug number), όνομα σταθμού (station name), διεύθυνση και μερικές ακόμη πληροφορίες που δεν είναι χρήσιμες για τη μοντελοποίηση μας. Δεδομένου ότι τα δεδομένα είναι περιορισμένα και οι σταθμοί φόρτισης που συμμετέχουν στο σύνολο δεδομένων ανέρχονται σε μικρό αριθμό (24), θεωρούμε την πόλη Palo Alto ως μεμονωμένο σταθμό φόρτισης με μεγαλύτερο αριθμό βυσμάτων. Στόχος μας είναι να προβλέψουμε την ωριαία ζήτηση σε kWh ενέργειας για φόρτιση ηλεκτρικών οχημάτων, δεδομένης της ζήτησης καθενός από έναν συγκεκριμένο αριθμό προηγούμενων ωρών στην περίπτωση μας ίση με 24.

Το ακατέργαστο αρχείο CSV που προσφέρεται από το Palo Alto Open Data χρειάζεται κάποια προεπεξεργασία προτού χρησιμοποιηθεί στην εκπαίδευση του νευρωνικού μας δικτύου. Η βιβλιοθήκη ανάλυσης δεδομένων Pandas της Python χρησιμοποιείται για την επεξεργασία των δεδομένων μας. Πρώτα απ' όλα, οι μη χρήσιμες στήλες αφαιρούνται. Διατηρούνται οι στήλες "Start time" και "Energy (kWh)". Η "ώρα έναρξης" είναι μια χρονική σήμανση που υποδεικνύει την ώρα και την ημερομηνία έναρξης της περιόδου φόρτισης και η ενέργεια (kWh) είναι ένας δεκαδικός αριθμός που υποδεικνύει την κατανάλωση ενέργειας σε kWh κάθε περιόδου φόρτισης. Δεν υπάρχουν κενές τιμές (N/A) σε αυτές τις δύο στήλες. Για σκοπούς μοντελοποίησης της ζήτησης, θεωρούμε ότι η διάρκεια κάθε συνεδρίας είναι αμελητέα. Ως αποτέλεσμα, οι συνεδρίες φόρτισης ταξινομούνται κατά ώρα και ημερομηνία έναρξης. Στη συνέχεια, οι συνεδρίες ομαδοποιούνται ανά ώρα και αθροίζεται η συνολική kWh ενέργειας των συνεδριών για κάθε ώρα. Το αποτέλεσμα είναι ένας πίνακας δύο στηλών, με την πρώτη στήλη να είναι ο δείκτης κάθε ώρας στο σύνολο και η δεύτερη να αντιπροσωπεύει τη συνολική ενεργειακή ζήτηση σε kWh για την αντίστοιχη ώρα.

Ο κώδικας φαίνεται παρακάτω:

```
import pandas as pd

## import raw csv file data into dataframe
file = '/content/drive/MyDrive/83palo_alto/CY20Q4.csv'
df = pd.read_csv(file, low_memory=False)

## remove non useful columns
df = df.drop(['MAC Address', 'Station Name', 'Start Time Zone',
             'End Time Zone', 'Org Name',
```

```

        'End Date' , 'Transaction Date (Pacific Time)' ,
'Total Duration (hh:mm:ss)' ,
        'GHG Savings (kg)' , 'Gasoline Savings (gallons)' ,
'Charging Time (hh:mm:ss)' ,
        'Port Type' , 'Port Number' , 'Plug Type' , 'EVSE
ID' , 'Address 1' , 'City' ,
        'State/Province' , 'Postal Code' , 'Country' ,
'Latitude' , 'Longitude' , 'Currency' ,
        'Fee' , 'Ended By' , 'Plug In Event Id' , 'Driver
Postal Code' , 'User ID' , 'County' ,
        'System S/N' , 'Model Number'] , axis = 1)

df.to_csv("/content/drive/MyDrive/83palo_alto/CY20Q4b.csv" ,
index=False)
print('Removed non useful columns and exported to CY20Q4b.csv')

```

Ομαλοποιούμε τα δεδομένα κατά ώρα και κατά ημέρα, για να κάνουμε διαφορετικές δοκιμές, αν και τελικά θα χρησιμοποιήσουμε μόνο τα ωριαία δεδομένα.

```

import pandas as pd
import datetime

file = '/content/drive/MyDrive/83palo_alto/CY20Q4b.csv'
df = pd.read_csv(file, low_memory=False)

# change dtypes from object to datetime
df['Start Date'] = pd.to_datetime(df['Start Date'] , errors
='coerce')

##print(df.isna().sum())          ##count how many NAs
df = df.dropna()    ## remove rows with NA values

#group by Start Date and add energy consumption hourly and daily
hourly_df = df.groupby(pd.Grouper(key='Start
Date',freq='1H')).sum()
daily_df = df.groupby(pd.Grouper(key='Start
Date',freq='1D')).sum()
hourly_df.reset_index( drop=True, inplace=True)

```

```

daily_df.reset_index( drop=True, inplace=True)

hourly_df.index.name = 'Hour'
daily_df.index.name = 'Day'

hourly_df.columns = ['kWh']
daily_df.columns = ['kWh']

print(daily_df.head())
print('')

daily_df.to_csv("/content/drive/MyDrive/83palo_alto/
dailyCY20Q4.csv" , index=False)
hourly_df.to_csv("/content/drive/MyDrive/83palo_alto/
hourlyCY20Q4.csv" , index=False)

print('Grouped by day and exported to dailyCY20Q4.csv')
print('')
print('Grouped by hour and exported to hourlyCY20Q4.csv')

```

Πρέπει να χωρίσουμε τα δεδομένα μας σε δύο μέρη: Τμήμα εκπαίδευσης (training set) και τμήμα δοκιμής (test set). Θα χρησιμοποιήσουμε διαχωρισμό 0,7, έτσι ώστε το 30% των δεδομένων να παραμένει εκτός εκπαίδευσης, για την αξιολόγηση του μοντέλου μας μετά την εκπαίδευση. Δεν υπάρχει ανάγκη για ένα τρίτο τμήμα που θα χρησιμοποιηθεί για την αξιολόγηση του μοντέλου μεταξύ των εποχών, αφού το Keras θα κρατήσει αυτόματα ένα μέρος των δεδομένων του εκπαίδευσης ως δεδομένα επικύρωσης. Το test set χρησιμοποιείται μετά την εκπαίδευση, για πρόβλεψη ζήτησης χρησιμοποιώντας το εκπαιδευμένο μοντέλο και τα αποτελέσματα συγκρίνονται με τις πραγματικές τιμές.

Χρησιμοποιούμε λοιπόν τον παρακάτω κώδικα:

```

import pandas as pd
from sklearn.preprocessing import MinMaxScaler
import numpy as np

file = '/content/drive/MyDrive/83palo_alto/hourlyCY20Q4.csv'
df = pd.read_csv(file, low_memory=False)

## we now need to split our data, into train and test data parts

split = 0.7
train_size=int(len(df)*split)

```

```

test_size=len(df)-train_size
train, test = df.iloc[0:train_size], df.iloc[train_size:len(df)]
test = test.reset_index(drop=True)
train.dropna()
test.dropna(inplace=True)

```

Τέλος, τα δεδομένα κανονικοποιούνται στο εύρος (0,1) για ταχύτερη και ακριβέστερη εκπαίδευση του μοντέλου, καθώς τα LSTM αναμένουν ότι τα δεδομένα θα είναι εντός της κλίμακας της συνάρτησης ενεργοποίησης που χρησιμοποιείται από το δίκτυο, η οποία στην περίπτωση του LSTM είναι η υπερβολική εφραπτομένη (tanh). Για να γίνει το πείραμα δίκαιο, η κανονικοποίηση γίνεται μετά τη διάσπαση των δεδομένων, προκειμένου να αποφευχθεί η διαρροή πληροφοριών από το training set στο test set.

```

#scaling of data using MinMaxScaler of sklearn
scaler_train=MinMaxScaler(feature_range=(0,1))
scaler_test=MinMaxScaler(feature_range=(0,1))

train = scaler_train.fit_transform(np.array(train))
test = scaler_test.fit_transform(np.array(test))

print ('Shape of train is: (%d , %d)' %train.shape)
print('')
print ('Shape of test is: (%d, %d)' %test.shape)

```

Το επόμενο βήμα είναι η δημιουργία ορισμένων ακολουθιών, το μήκος των οποίων είναι προσαρμοσμένο και ορίζεται ίσο με 24. Μετά από πειραματισμούς με διαφορετικές τιμές, το 24 έχει επιλεγεί ως η τιμή που ταιριάζει καλύτερα και μπορεί να περιγραφεί ως η απαίτηση του τα τελευταία 24ωρα μπορούν να περιγράψουν τη ζήτηση της επόμενης ώρας με καλή ακρίβεια.

Υπάρχει επίσης ανάγκη για αναδιαμόρφωση των δεδομένων, καθώς η είσοδος LSTM θα πρέπει να έχει τη μορφή [Δείγματα, Χρόνια βήματα, Χαρακτηριστικά]. Στην περίπτωσή μας έχουμε μόνο ένα χαρακτηριστικό, οπότε η τρίτη διάσταση ισούται με 1.

Αυτά γίνονται με τον παρακάτω κώδικα:

```

##now we need to create sequences of length = time_step
# convert an array of values into a dataset matrix

def create_dataset(dataset, time_step=1):
    dataX, dataY = [], []
    for i in range(len(dataset)-time_step-1):
        a = dataset[i:(i+time_step), 0]   ###i=0, 0,1,2,3-----99
100
        dataX.append(a)
        dataY.append(dataset[i + time_step, 0])
    return np.array(dataX), np.array(dataY)

time_step = 24
Xtrain, ytrain = create_dataset(train, time_step)
Xtest, ytest = create_dataset(test, time_step)

Xtrain =Xtrain.reshape(Xtrain.shape[0],Xtrain.shape[1] , 1)
Xtest = Xtest.reshape(Xtest.shape[0],Xtest.shape[1] , 1)

print ('Shape of Xtrain is: (%d , %d )' %Xtrain.shape)
print('')
print ('Shape of ytrain is: (%d,)' %ytrain.shape)
print('')
print ('Shape of Xtest is: (%d , %d )' %Xtest.shape)
print('')
print ('Shape of ytest is: (%d ,)' %ytest.shape)

```

Και τα αποτελέσματα στην έξοδο που μας δίνουν το σχήμα και το μέγεθος των δεδομένων έχουν ως εξής

```

Shape of Xtrain is: (57816 , 24 , 1)
Shape of ytrain is: (57816,)
Shape of Xtest is: (24765 , 24 , 1)
Shape of ytest is: (24765 ,)

```

## Κεφάλαιο 5

### 5.1 Μοντέλα βασικής απόδοσης (Baseline)

Ένα βασικό σημείο στην απόδοση των προβλέψεων ενός μοντέλου είναι να υπάρχει ένα μέτρο σύγκρισης. Αποτελεί σημείο αναφοράς για όλες τις νέες τεχνικές μοντελοποίησης ενός προβλήματος. Εάν ένα μοντέλο επιτυγχάνει απόδοση ίση ή χειρότερη από το βασικό επίπεδο, η τεχνική πρέπει απορριφθεί.

Η τεχνική που χρησιμοποιείται για τη δημιουργία μιας πρόβλεψης για τον υπολογισμό της βασικής απόδοσης πρέπει να είναι εύκολη στην εφαρμογή και αφελής ως προς τις λεπτομέρειες του συγκεκριμένου προβλήματος.

Προτού καθοριστεί η βάση απόδοσης για ένα πρόβλημα πρόβλεψης, χρειάζεται να αναπτυχθεί μια ζώνη δοκιμής. Αυτή αποτελείται από:

Το σύνολο δεδομένων που θα χρησιμοποιηθούν για να εκπαιδευτούν και να αξιολογηθούν τα μοντέλα.

Την τεχνική δειγματοληψίας που θα χρησιμοποιηθεί για την εκτίμηση της απόδοσης της τεχνικής αυτής.

Το μέτρο απόδοσης που θα χρησιμοποιηθεί για την αξιολόγηση των προβλέψεων (π.χ. μέσο τετραγωνικό σφάλμα).

Το επόμενο βήμα είναι να επιλέγει μια αφελής τεχνική που μπορείτε να χρησιμοποιηθεί για να κάνει μια πρόβλεψη και να υπολογιστεί η απόδοση βάσης.

Η βασική απόδοση σε κάθε πρόβλημα πρόβλεψης χρονοσειρών είναι απαραίτητη προκειμένου να αξιολογηθεί η απόδοση των εκάστοτε αναπτυσσόμενων μοντέλων.

Τρία χαρακτηριστικά είναι που καθιστούν μια τεχνική καλή για τον ορισμό της βασικής απόδοσης:

Να είναι απλή: Μια μέθοδος που απαιτεί λίγη ή καθόλου νοημοσύνη.

Γρήγορη: Μια μέθοδος που εφαρμόζεται γρήγορα και με λίγους υπολογιστικούς πόρους για να κάνει μια πρόβλεψη.

Επαναλήψιμη: Μια μέθοδος που είναι ντετερμινιστική, που σημαίνει ότι παράγει μια αναμενόμενη έξοδο δεδομένης της ίδιας εισόδου.

Ένας κοινός αλγόριθμος που χρησιμοποιείται για τον καθορισμό μιας βασικής απόδοσης είναι ο αλγόριθμος επιμονής.

## 5.1.1 Βασικό μοντέλο Persistence

Ένα βασικό μοντέλο για την πρόβλεψη της μελλοντικής ζήτησης φόρτισης σε kWh μπορεί να αναπτυχθεί χρησιμοποιώντας το λεγόμενο Persistence μοντέλο που χρησιμοποιείται ευρέως σε παρόμοιες εφαρμογές. Αξιολογούμε αυτό το μοντέλο χρησιμοποιώντας μετρήσεις RMSE (μέσο τετραγωνικό σφάλμα) και Normalized RMSE (κανονικοποιημένο μέσο τετραγωνικό σφάλμα). Ο persistence αλγόριθμος χρησιμοποιεί την τιμή του προηγούμενου χρονικού βήματος ( $t-1$ ) για να προβλέψει το αποτέλεσμα στο επόμενο χρονικό βήμα ( $t$ ). Αυτός ο αλγόριθμος είναι αφελής, καθώς δεν υποθέτει τίποτα σχετικά με τις ιδιαιτερότητες του προβλήματος των χρονοσειρών στο οποίο εφαρμόζεται. Όπως αναμενόταν, το αποτέλεσμα δεν είναι ικανοποιητικό με τιμή RMSE ίση με 29,583 και κανονικοποιημένη τιμή RMSE ίση με 0,782.

Παρακάτω φαίνεται ο κώδικας που χρησιμοποιήθηκε για την ανάπτυξη αυτού του βασικού μοντέλου.

```
import pandas as pd
import datetime
import numpy as np
import random
from collections import deque
from sklearn import preprocessing
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, LSTM,
BatchNormalization
from tensorflow.keras.callbacks import TensorBoard
from tensorflow.keras.callbacks import ModelCheckpoint
from matplotlib import pyplot
from sklearn.metrics import mean_squared_error
from math import sqrt

import matplotlib.patches as mpatches
import matplotlib.pyplot as plt

file = '/content/drive/MyDrive/83palo_alto/hourlyCY20Q4.csv'
df = pd.read_csv(file, low_memory=False)

history = df['kWh'].tolist()

predictions = list()
predictions = [0,0]
for i in range(2, (len(history))):
    # make prediction
    predictions.append(history[i-1])
    # observation

rmse = sqrt(mean_squared_error(history, predictions))
print('RMSE: %.3f' % rmse)
obar = sum(history) / len(history)
nrmse = rmse / obar
```

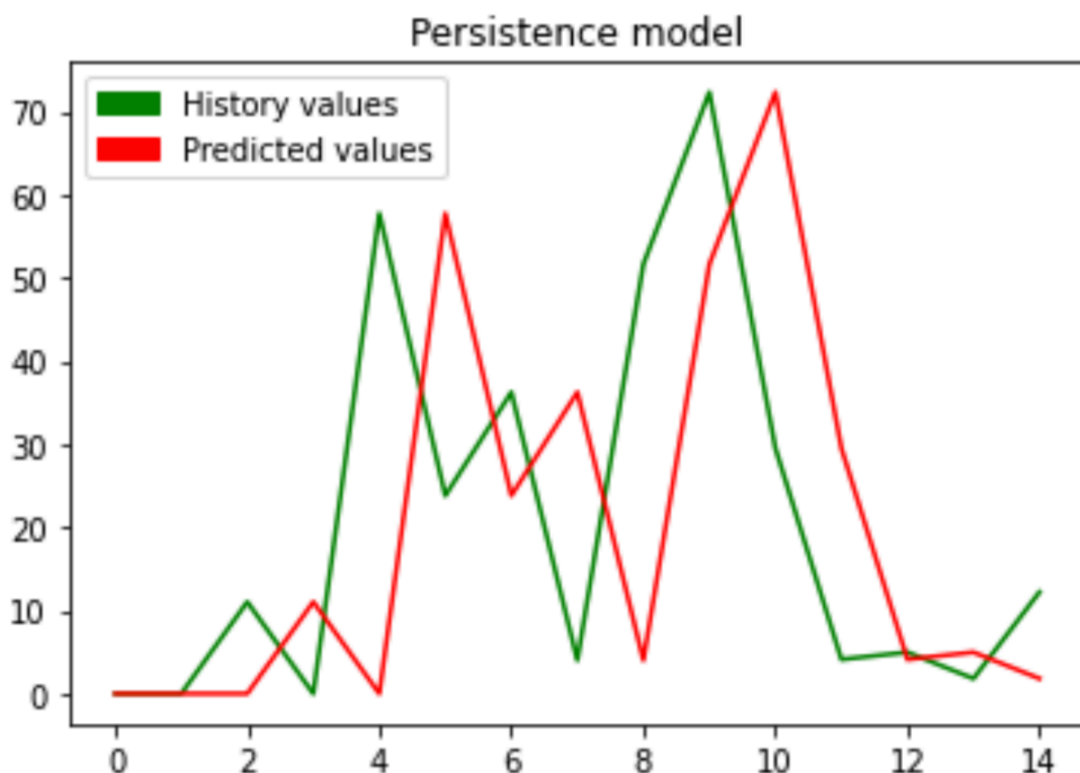


```

print('NRMSE: %.3f' % nrmse)

pyplot.plot(history[81057:81076], color='green')
pyplot.plot(predictions[81057:81076], color='red')
green_patch = mpatches.Patch(color='green', label='History
values')
red_patch = mpatches.Patch(color='red', label='Predicted values')
plt.legend(handles=[green_patch, red_patch])
plt.title("Persistence model")
pyplot.show()

```



Παραπάνω, φαίνονται τα αποτελέσματα των μετρήσεων που χρησιμοποιήθηκαν για την αξιολόγηση και την πρόβλεψη του αλγορίθμου persistence για μια τυχαία σειρά δεδομένων. Ταυτόχρονα προβάλλουμε και τις μετρήσεις από τα RMSE (μέσο τετραγωνικό σφάλμα) και Normalized RMSE (κανονικοποιημένο μέσο τετραγωνικό σφάλμα) .

```

Persistence model
RMSE: 29.583
NRMSE: 0.782

```

## 5.1.2 Βασικό μοντέλο γραμμικής παλινδρόμησης

Για ένα δεύτερο βασικό μοντέλο, ορίζουμε ένα απλό μοντέλο γραμμικής παλινδρόμησης (linear regression) που έχει ένα μόνο πλήρως συνδεδεμένο κρυφό επίπεδο με τον ίδιο αριθμό νευρώνων με το μήκος της εισόδου (24). Το δίκτυο χρησιμοποιεί την ανορθωτική λειτουργία ενεργοποίησης (rectified activation function ή relu) στο κρυφό επίπεδο. Δεν χρησιμοποιούμε καμία συνάρτηση ενεργοποίησης για το επίπεδο εξόδου, καθώς μας ενδιαφέρει η άμεση πρόβλεψη αριθμητικών τιμών. Σε αυτό το μοντέλο, τα αποτελέσματα βελτιώνονται, σε σύγκριση με αυτά του μοντέλου Persistence. Ο κώδικας που χρησιμοποιήθηκε έχει ως εξής:

Δημιουργία του μοντέλου:

```
import tensorflow as tf
from tensorflow import keras
import numpy
import keras.models
import tensorflow
from keras.models import Sequential
from keras.layers import Activation, Dense

model = Sequential()
model.add(Dense(24,
                input_dim = 24,
                kernel_initializer='normal',
                activation='relu'))
model.add(Dense(1, kernel_initializer='normal'))
model.summary()

model.compile(loss='mse', optimizer='adam')
```

Στη συνέχεια η εντολή `model.summary()` μας δίνει το εξής αποτέλεσμα:

☞ Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 24)	600
dense_1 (Dense)	(None, 1)	25
Total params: 625		
Trainable params: 625		
Non-trainable params: 0		

Στη συνέχεια εκπαιδεύουμε το μοντέλο που δημιουργήσαμε με τις εντολές:

```
history = model.fit(
    Xtrain,
    ytrain,
    epochs=25,

    # Calculate validation results on 10% of the training data
    validation_split = 0.1)
```

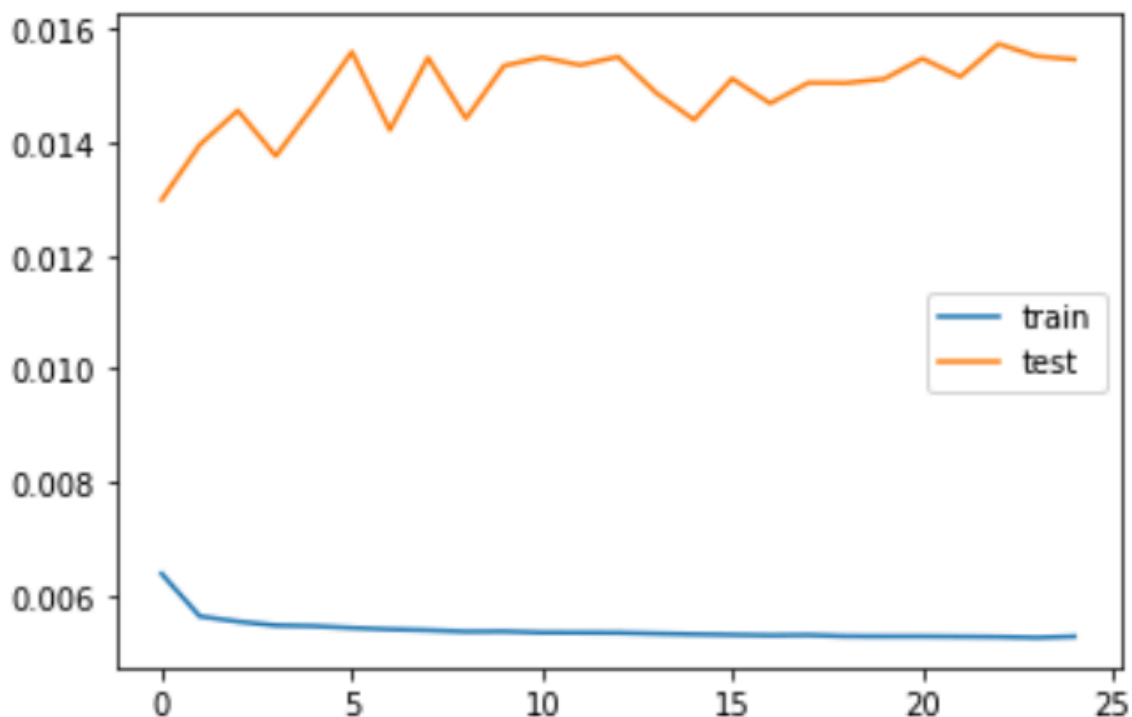
Και το αποτέλεσμα που παίρνουμε από την εκπαίδευση για 25 εποχές έχει ως εξής:

```
Epoch 1/25
1627/1627 [=====] - 77s 44ms/step - loss: 0.0055 - val_loss: 0.0148
Epoch 2/25
1627/1627 [=====] - 72s 44ms/step - loss: 0.0058 - val_loss: 0.0144
Epoch 3/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0055 - val_loss: 0.0148
Epoch 4/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0054 - val_loss: 0.0143
Epoch 5/25
1627/1627 [=====] - 72s 44ms/step - loss: 0.0054 - val_loss: 0.0141
Epoch 6/25
1627/1627 [=====] - 72s 45ms/step - loss: 0.0054 - val_loss: 0.0141
Epoch 7/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0053 - val_loss: 0.0142
Epoch 8/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0059 - val_loss: 0.0142
Epoch 9/25
1627/1627 [=====] - 71s 43ms/step - loss: 0.0053 - val_loss: 0.0143
Epoch 10/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0053 - val_loss: 0.0146
Epoch 11/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0053 - val_loss: 0.0146
Epoch 12/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0053 - val_loss: 0.0148
Epoch 13/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0053 - val_loss: 0.0146
Epoch 14/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0054 - val_loss: 0.0147
Epoch 15/25
1627/1627 [=====] - 70s 43ms/step - loss: 0.0056 - val_loss: 0.0147
Epoch 16/25
1627/1627 [=====] - 69s 42ms/step - loss: 0.0052 - val_loss: 0.0146
Epoch 17/25
1627/1627 [=====] - 67s 41ms/step - loss: 0.0052 - val_loss: 0.0146
Epoch 18/25
1627/1627 [=====] - 69s 42ms/step - loss: 0.0052 - val_loss: 0.0146
Epoch 19/25
1627/1627 [=====] - 68s 42ms/step - loss: 0.0051 - val_loss: 0.0147
Epoch 20/25
1627/1627 [=====] - 70s 43ms/step - loss: 0.0051 - val_loss: 0.0145
Epoch 21/25
1627/1627 [=====] - 72s 44ms/step - loss: 0.0051 - val_loss: 0.0145
Epoch 22/25
1627/1627 [=====] - 72s 44ms/step - loss: 0.0051 - val_loss: 0.0145
Epoch 23/25
1627/1627 [=====] - 73s 45ms/step - loss: 0.0051 - val_loss: 0.0144
Epoch 24/25
1627/1627 [=====] - 72s 45ms/step - loss: 0.0050 - val_loss: 0.0145
Epoch 25/25
1627/1627 [=====] - 72s 44ms/step - loss: 0.0050 - val_loss: 0.0143
```

Παρατηρούμε λοιπόν ότι μετά από την 4η εποχή και μετά, το `val_loss` δηλαδή το σφάλμα που προκύπτει από τα δεδομένα επαλήθευσης σταματάει να μειώνεται, οπότε το μοντέλο έχει εκπαιδευτεί.

Στη συνέχεια με τις παρακάτω εντολές, προβάλλουμε σε γραφική παράσταση την πορεία του `loss` και του `validation_loss`, κατά την εκπαίδευση του μοντέλου και παίρνουμε τα αναμενόμενα αποτελέσματα:

```
import matplotlib.pyplot as plt
plt.plot(history.history['loss'], label='train')
plt.plot(history.history['val_loss'], label='test')
plt.legend();
```



Στη συνέχεια με χρήση των παρακάτω εντολών, χρησιμοποιούμε το εκπαιδευμένο μοντέλο για να κάνουμε πρόβλεψη με τα δεδομένα που είχαμε κρατήσει για έλεγχο του μοντέλου (test set). Μετά την πρόβλεψη, τα δεδομένα που έχουν προκύψει πρέπει να κανονικοποιηθούν αντίστροφα, για να αναπαριστούν πραγματικά μεγέθη, αφού διαφορετικά θα βρισκόταν στο εύρος  $[0,1]$ .

```
#Prediction with the trained linear regression model
```

```
import math
from sklearn.metrics import mean_squared_error
from math import sqrt
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
import numpy as np
```

```
train_predict=model.predict(Xtrain)
```

```

test_predict=model.predict(Xtest)

train_predict = scaler_train.inverse_transform(train_predict)

ytrain = ytrain.reshape(-1,1)
ytrain = scaler_train.inverse_transform(ytrain)

test_predict = scaler_test.inverse_transform(test_predict)
ytest = ytest.reshape(-1,1)

ytest = scaler_test.inverse_transform(ytest)

```

Στο επόμενο στάδιο παίρνουμε ένα τυχαίο κομμάτι από τα δεδομένα της πρόβλεψης, και με τον παρακάτω κώδικα τα συγκρίνουμε γραφικά με τις πραγματικές τιμές. Το αποτέλεσμα είναι καλύτερο από το Persistence model. Ταυτόχρονα προβάλλουμε και τις μετρήσεις από τα RMSE (μέσο τετραγωνικό σφάλμα) και Normalized RMSE (κανονικοποιημένο μέσο τετραγωνικό σφάλμα)

```

from matplotlib import pyplot
from sklearn.metrics import mean_squared_error
import math

from math import sqrt

import matplotlib.patches as mpatches

import matplotlib.pyplot as plt

pyplot.plot(ytest[6200:6300], color='green')

pyplot.plot(test_predict[6200:6300], color='red')

green_patch = mpatches.Patch(color='green', label='actual')
red_patch = mpatches.Patch(color='red', label='predicted')

plt.legend(handles=[green_patch, red_patch])

plt.title("linear model")

pyplot.show()

rmse = math.sqrt(mean_squared_error(ytrain,train_predict))
print('Linear model')

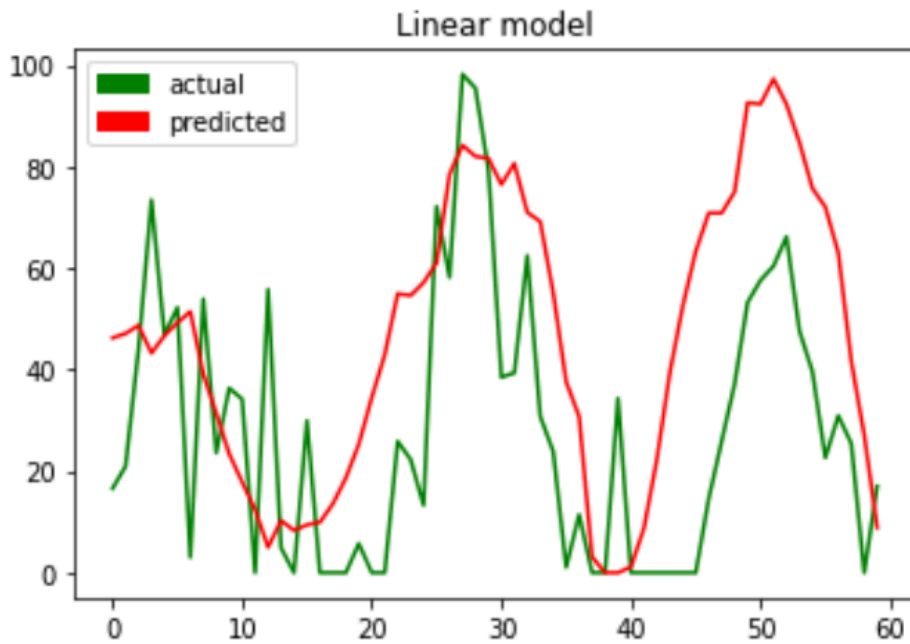
print('RMSE: %.3f' % rmse)
ybar = sum(ytest) / len(ytest)

normse = rmse / ybar

```

```
print('NRMSE: %.3f' % nrmse)
```

Η γραφική παράσταση:



Και οι αριθμητικές τιμές:

```
Linear model  
RMSE: 24.582  
NRMSE: 0.651
```

## 5.2 Μοντέλο LSTM

Η μελλοντική ζήτηση της ενέργειας εξαρτάται σε μεγάλο βαθμό από τις προηγούμενες τιμές της ζήτησης, και ως εκ τούτου η πρόβλεψη της ζήτησης αυτής θεωρείται ως ένα πρόβλημα χρονοσειρών (time-series) . Για την εφαρμογή και την εκπαίδευση του LSTM μοντέλου μας, θα χρησιμοποιήσουμε τη βιβλιοθήκη υψηλού επιπέδου μηχανικής μάθησης Keras σε Python 3.

Δημιουργούμε το μοντέλο μας χρησιμοποιώντας το Sequential module του Keras. Το μοντέλο μας περιλαμβάνει ένα αμφίδρομο LSTM επίπεδο, δύο πυκνά (Dense) επίπεδα και δύο επίπεδα Dropout. Τα αμφίδρομα LSTM είναι μια επέκταση των παραδοσιακών LSTM που μπορούν να βελτιώσουν την απόδοση του μοντέλου σε προβλήματα ταξινόμησης ακολουθιών. Έτσι, αυτό το επίπεδο εκπαιδεύει δύο αντί για ένα LSTM στην ακολουθία εισόδου. Αυτό μπορεί να οδηγήσει σε γρηγορότερη και ακόμη και πληρέστερη εκμάθηση του προβλήματος. Μετά από αυτό, προσθέτουμε δύο επίπεδα Dropout και δύο πυκνά συνδεδεμένα επίπεδα νευρωνικών δικτύων, το ένα μετά το άλλο. Κατά τον ορισμό των επιπέδων Dropout, καθορίζουμε τιμή ίση με 0,2, που σημαίνει ότι το 20% των επιπέδων θα πέσει (drop). Το τελευταίο Dense layer αποτελείται από έναν μόνο νευρώνα έτσι ώστε να ταιριάζει με την επιθυμητή έξοδο. Αφού καθορίσουμε τα επίπεδα του μοντέλου μας, το μεταγλωττίζουμε (compile) χρησιμοποιώντας το δημοφιλές adam optimiser και ορίζουμε τη λειτουργία απώλειας (loss function) ως το μέσο τετραγωνικό σφάλμα (PMSE

Παρακάτω φαίνεται ο κώδικας που χρησιμοποιήθηκε για τον ορισμό των επιπέδων του μοντέλου:

```
from tensorflow import keras
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, LSTM,
BatchNormalization
from tensorflow.keras.callbacks import TensorBoard
from tensorflow.keras.callbacks import ModelCheckpoint
model = keras.Sequential()

model.add(
    keras.layers.Bidirectional(
        keras.layers.LSTM(
            units=128,
            input_shape=(Xtrain.shape[1], Xtrain.shape[2])
        )
    )
)

model.add(Dense(25,
                kernel_initializer='normal',
                activation='relu'))

model.add(Dense(1, kernel_initializer='normal', activation='relu'))

model.compile(loss='mean_squared_error', optimizer='adam')
```

## 5.3 Εκπαίδευση μοντέλου LSTM

Με τη χρήση του 0,1 ως ποσοστό δεδομένων εισόδου για επαλήθευση του μοντέλου, ορίζουμε ότι το μοντέλο χρησιμοποιεί το 10% των δεδομένων της εισόδου για επαλήθευση κατά τη διάρκεια της εκπαίδευσης, ανάμεσα στις εποχές. Μπορούμε να παρατηρήσουμε την πρόοδο και σταματάμε την προπόνηση μετά από 7 εποχές όταν το validation loss σταματά να μειώνεται.

Χρησιμοποιούμε τις παρακάτω εντολές για την εκπαίδευση του μοντέλου:

```
history = model.fit(
    Xtrain, ytrain,
    validation_split=0.1,
    epochs=25,
    batch_size=32,
    shuffle=False
)
```

Και παίρνουμε το εξής αποτέλεσμα:

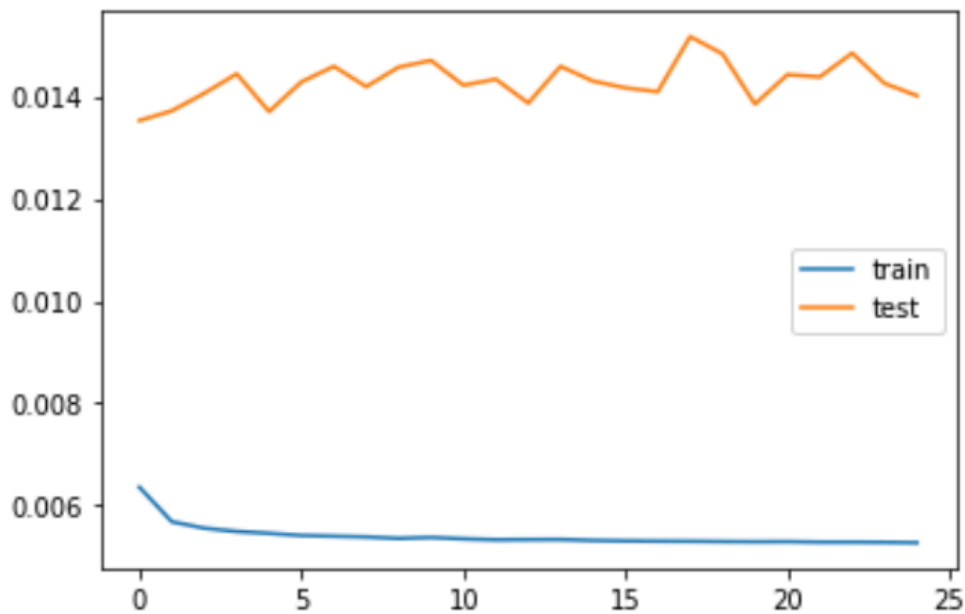
```
Epoch 1/25
1627/1627 [=====] - 77s 44ms/step - loss: 0.0055 - val_loss: 0.0148
Epoch 2/25
1627/1627 [=====] - 72s 44ms/step - loss: 0.0058 - val_loss: 0.0144
Epoch 3/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0055 - val_loss: 0.0148
Epoch 4/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0054 - val_loss: 0.0143
Epoch 5/25
1627/1627 [=====] - 72s 44ms/step - loss: 0.0054 - val_loss: 0.0141
Epoch 6/25
1627/1627 [=====] - 72s 45ms/step - loss: 0.0054 - val_loss: 0.0141
Epoch 7/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0053 - val_loss: 0.0142
Epoch 8/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0059 - val_loss: 0.0142
Epoch 9/25
1627/1627 [=====] - 71s 43ms/step - loss: 0.0053 - val_loss: 0.0143
Epoch 10/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0053 - val_loss: 0.0146
Epoch 11/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0053 - val_loss: 0.0146
Epoch 12/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0053 - val_loss: 0.0148
Epoch 13/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0053 - val_loss: 0.0146
Epoch 14/25
1627/1627 [=====] - 71s 44ms/step - loss: 0.0054 - val_loss: 0.0147
Epoch 15/25
1627/1627 [=====] - 70s 43ms/step - loss: 0.0056 - val_loss: 0.0147
Epoch 16/25
1627/1627 [=====] - 69s 42ms/step - loss: 0.0052 - val_loss: 0.0146
Epoch 17/25
1627/1627 [=====] - 67s 41ms/step - loss: 0.0052 - val_loss: 0.0146
Epoch 18/25
1627/1627 [=====] - 69s 42ms/step - loss: 0.0052 - val_loss: 0.0146
Epoch 19/25
1627/1627 [=====] - 68s 42ms/step - loss: 0.0051 - val_loss: 0.0147
Epoch 20/25
1627/1627 [=====] - 70s 43ms/step - loss: 0.0051 - val_loss: 0.0145
Epoch 21/25
1627/1627 [=====] - 72s 44ms/step - loss: 0.0051 - val_loss: 0.0145
Epoch 22/25
1627/1627 [=====] - 72s 44ms/step - loss: 0.0051 - val_loss: 0.0145
Epoch 23/25
1627/1627 [=====] - 73s 45ms/step - loss: 0.0051 - val_loss: 0.0144
Epoch 24/25
1627/1627 [=====] - 72s 45ms/step - loss: 0.0050 - val_loss: 0.0145
Epoch 25/25
1627/1627 [=====] - 72s 44ms/step - loss: 0.0050 - val_loss: 0.0143
```



Με τη χρήση των ακόλουθων εντολών προβάλλουμε γραφικά την πορεία του loss και του validation\_loss κατά τη διάρκεια της εκπαίδευσης.

```
import matplotlib.pyplot as plt
plt.plot(history.history['loss'], label='train')
plt.plot(history.history['val_loss'], label='test')
plt.legend();
```

Και η γραφική παράσταση έχει προφανώς ως εξής:



## 5.4 Αποτελέσματα μοντέλου LSTM

Μετά την επιτυχή εκπαίδευση του μοντέλου μας, χρησιμοποιούμε το σετ δοκιμών για να προβλέψουμε την ωριαία ζήτηση και να τη συγκρίνουμε με τις πραγματικές τιμές. Μετά την πρόβλεψη, οι τιμές κλιμακώνονται αντίστροφα έτσι ώστε να δείχνουν τις πραγματικές τιμές. Αυτό το κάνουμε με τις παρακάτω εντολές:

```
#Prediction with the trained LSTM model and inverse scaling of the data
```

```
import math

from sklearn.metrics import mean_squared_error
from math import sqrt
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
import numpy as np

train_predict=model.predict(Xtrain)
test_predict=model.predict(Xtest)

train_predict = scaler_train.inverse_transform(train_predict)

ytrain = ytrain.reshape(-1,1)
ytrain = scaler_train.inverse_transform(ytrain)

test_predict = scaler_test.inverse_transform(test_predict)
ytest = ytest.reshape(-1,1)

ytest = scaler_test.inverse_transform(ytest)
```

Στο επόμενο στάδιο παίρνουμε ένα τυχαίο κομμάτι από τα δεδομένα της πρόβλεψης, και με τον παρακάτω κώδικα τα συγκρίνουμε γραφικά με τις πραγματικές τιμές. Το αποτέλεσμα είναι καλύτερο από το Persistence model. Ταυτόχρονα προβάλλουμε και τις μετρήσεις από τα RMSE (ρίζα μέσης τετραγωνικής απόκλισης) και Normalized RMSE (κανονικοποιημένη ρίζα μέσης τετραγωνικής απόκλισης).

```
from matplotlib import pyplot
from sklearn.metrics import mean_squared_error
from math import sqrt

import matplotlib.patches as mpatches
import matplotlib.pyplot as plt
```

```

test_predict = np.squeeze(test_predict, -1)
print(test_predict.shape)

print(ytest.shape)

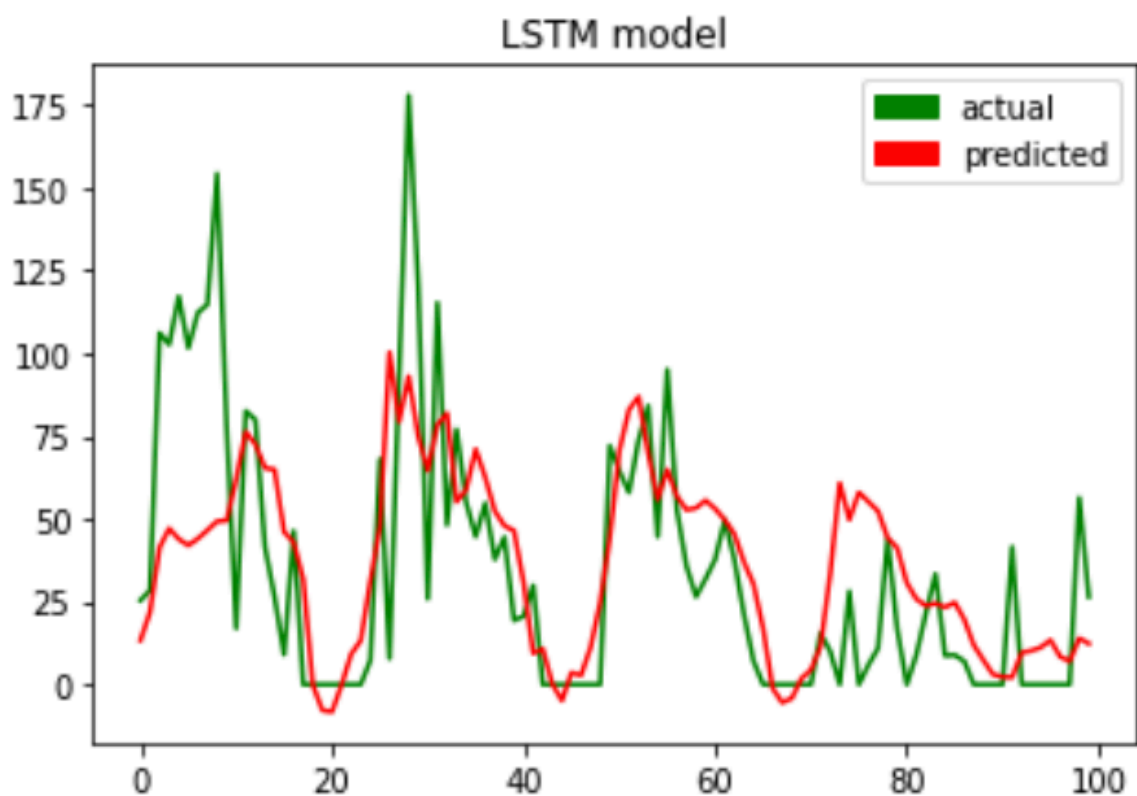
pyplot.plot(ytest[6300:6360], color='green')
pyplot.plot(test_predict[6300:6360], color='red')
green_patch = mpatches.Patch(color='green', label='actual')
red_patch = mpatches.Patch(color='red', label='predicted')
plt.legend(handles=[green_patch, red_patch])
plt.title("LSTM model")
pyplot.show()

rmse = math.sqrt(mean_squared_error(ytrain,train_predict))

print(rmse)

```

Και παίρνουμε τη γραφική παράσταση:



Τα αποτελέσματα είναι πολλά υποσχόμενα και μετά την εκπαίδευση του μοντέλου 10 συνεχόμενες φορές έχουμε τις ακόλουθες τιμές:

```
LSTM model  
Min RMSE: 16.235 NRMSE: 0.429  
Max RMSE: 18.024 NRMSE: 0.477  
Mean RMSE: 17.310 NRMSE: 0.452
```

## Κεφάλαιο 6

### Συμπεράσματα και μελλοντική εργασία

#### 6.1 Συμπεράσματα και μελλοντική εργασία

Τα αποτελέσματα συγκριτικά με τη βασική απόδοση δείχνουν ξεκάθαρα πόσο ισχυρά είναι τα LSTM για την ανάλυση χρονοσειρών και διαδοχικών δεδομένων.

Το μοντέλο έχει μεγάλες δυνατότητες βελτίωσης, με την εισαγωγή περισσότερων δεδομένων στην εκπαίδευση, πράγμα που σημαίνει ότι με τη χρήση πρόσθετων χαρακτηριστικών (features) και αύξηση των διαστάσεων του μοντέλου, η πρόβλεψη μπορεί να γίνει ακόμη πιο ακριβής.

## Βιβλιογραφία

- [1] Smart Grid The Future of the Electric Energy System. Available from: [https://www.researchgate.net/publication/325742525\\_Smart\\_Grid\\_The\\_Future\\_of\\_the\\_Electric\\_Energy\\_System](https://www.researchgate.net/publication/325742525_Smart_Grid_The_Future_of_the_Electric_Energy_System)
- [2] Smart Grid The Future of the Electric Energy System. Available from: [https://www.researchgate.net/publication/325742525\\_Smart\\_Grid\\_The\\_Future\\_of\\_the\\_Electric\\_Energy\\_System](https://www.researchgate.net/publication/325742525_Smart_Grid_The_Future_of_the_Electric_Energy_System)
- [3] Electric vehicles: A review of network modelling and future research needs, [https://www.researchgate.net/publication/292344412\\_Electric\\_vehicles\\_A\\_review\\_of\\_network\\_modelling\\_and\\_future\\_research\\_needs](https://www.researchgate.net/publication/292344412_Electric_vehicles_A_review_of_network_modelling_and_future_research_needs)
- [4] (PDF) Electric vehicles: A review of network modelling and future research needs. Available from: [https://www.researchgate.net/publication/292344412\\_Electric\\_vehicles\\_A\\_review\\_of\\_network\\_modelling\\_and\\_future\\_research\\_needs](https://www.researchgate.net/publication/292344412_Electric_vehicles_A_review_of_network_modelling_and_future_research_needs)
- [5] Smart Grid The Future of the Electric Energy System [https://www.researchgate.net/publication/325742525\\_Smart\\_Grid\\_The\\_Future\\_of\\_the\\_Electric\\_Energy\\_System](https://www.researchgate.net/publication/325742525_Smart_Grid_The_Future_of_the_Electric_Energy_System)
- [6] Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network - Alex Sherstinsky <https://arxiv.org/abs/1808.03314>
- [7] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term memory. Neural computation, 1997.
- [8] Electric Vehicles Statistics in the netherlands <https://www.rvo.nl/sites/default/files/2021/03/Statistics%20Electric%20Vehicles%20and%20Charging%20in%20The%20Netherlands%20up%20to%20and%20including%20January%202021.pdf>
- [9] Recurrent Neural Networks and LSTM explained <https://purnasaigudikandula.medium.com/recurrent-neural-networks-and-lstm-explained-7f51c7f6bbb9>
- [10] *Energies* 2021, 14, 2233. <https://doi.org/10.3390/en14082233> <https://www.mdpi.com/journal/energies>
- [11] Elaad NL, Data Analytics. Library Catalog. Available online: [www.elaad.nl/research/data-analytics](http://www.elaad.nl/research/data-analytics)
- [12] RotterdamOpenData. Charging Point Locations and Usage in Rotterdam. Library Catalog. Available online: <http://rotterdamopendata.nl>
- [13] Kevin Gurney University of Sheffield, An introduction to neural networks [https://www.inf.ed.ac.uk/teaching/courses/nlu/assets/reading/Gurney\\_et\\_al.pdf](https://www.inf.ed.ac.uk/teaching/courses/nlu/assets/reading/Gurney_et_al.pdf)
- [14] Introduction to Recurrent Neural Networks <https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/>
- [15] Vanishing gradient problem Wikipedia- [https://en.wikipedia.org/wiki/Vanishing\\_gradient\\_problem](https://en.wikipedia.org/wiki/Vanishing_gradient_problem)

[16] Exploding Gradient Problem - Deep AI <https://deepai.org/machine-learning-glossary-and-terms/exploding-gradient-problem>

[17] Open Data Boulder Colorado <https://open-data.bouldercolorado.gov>

[18] Caltech open data - <https://ev.caltech.edu/dataset>

[19] ParisData. Belib' Availability in Real-Time. Library Catalog. Available online: [opendata.paris.fr](http://opendata.paris.fr)

[20] SAP Labs. Electric Vehicle Charging Transactions of SAP Labs France Company Fleet. <https://news.sap.com>

[21] TransportTeam. Electric Vehicle Charging Sessions Dundee. Library Catalog. Available online: [data.dundee.gov.uk](http://data.dundee.gov.uk)

[22] OpenDataTeam. Electric Vehicle Charging Station Usage in Perth and Kinross. Library Catalog. Available online: [data.pkc.gov.uk](http://data.pkc.gov.uk)