



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**

**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ**

## **Διπλωματική Εργασία**

**Ανάπτυξη μεθόδων εκπαίδευσης αναδρομικών νευρωνικών δικτύων**



**Φοιτητής: Διονύσιος Γάκης  
ΑΜ: 18387153**

**Επιβλέπων Καθηγητής**

**Αλέξανδρος Αλεξανδρίδης  
Καθηγητής**

**ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, ΟΚΤΩΒΡΙΟΣ 2023**



**UNIVERSITY OF WEST ATTICA**  
**FACULTY OF ENGINEERING**  
**DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING**

## **Diploma Thesis**

**Development of training methods for recurrent neural networks**



**Student: Dionysios Gakis**  
**Registration Number: 18387153**

**Supervisor**

**Alexandros Alexandridis**  
**Professor**

**ATHENS-EGALEO, OCTOBER 2023**

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

Αλεξανδρίδης Αλέξανδρος, Καθηγητής	Ζέρβας Ευάγγελος, Καθηγητής	Ζώης Ηλίας, Αναπληρωτής Καθηγητής

**Copyright ©** Με επιφύλαξη παντός δικαιώματος. All rights reserved.

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ και Διονύσιος Γάκης,  
Οκτώβριος, 2023**

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 τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον/την συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

**ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ**

Ο κάτωθι υπογεγραμμένος Διονύσιος Γάκης του Νεκταρίου, με αριθμό μητρώου 18387153 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ,

**δηλώνω υπεύθυνα ότι:**

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.

Επιθυμώ την απαγόρευση πρόσβασης στο πλήρες κείμενο της εργασίας μου παρά μόνο έπειτα από αίτησή μου στη Βιβλιοθήκη και έγκριση του επιβλέποντος καθηγητή.»

Ο Δηλών  
Διονύσιος Γάκης





## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα διπλωματική αποτελεί και το τέλος των σπουδών μου στο τμήμα των Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών του Πανεπιστημίου Δυτικής Αττικής. Με την ολοκλήρωσή της, θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες σε όλους όσους συνέβαλλαν στην εκπόνησή της.

Πρώτα απ' όλα θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Αλέξανδρο Αλεξανδρίδη για τη συνεχή καθοδήγηση, τις ουσιώδεις συμβουλές καθώς επίσης και την αδιάκοπη συμπαράσταση και ενθάρρυνση που μου παρείχε σε κάθε στάδιο αυτής της έρευνας. Θα ήθελα επίσης να τον ευχαριστήσω και για τον ρόλο του ως καθηγητή κατά τη διάρκεια των σπουδών μου. Μέσω της άρτιας διδασκαλίας του ανακάλυψα και αγάπησα νέους επιστημονικούς τομείς ενώ στη συνέχεια μου έδωσε την ευκαιρία να τους εξερευνήσω μέσω της συγκεκριμένης διπλωματικής.

Θα ήθελα επίσης να ευχαριστήσω τον υποψήφιο διδάκτορα κ. Ιωάννη Κορδάτο για την τεράστια βοήθεια και καθοδήγηση που μου προσέφερε τόσο στο κομμάτι του προγραμματισμού όσο και στην κατανόηση όλων των σύνθετων εννοιών. Η αφοσίωσή του στην επιτυχία μου αλλά και η πίστη του στις δυνατότητες μου αποτέλεσαν καθοριστικό παράγοντα στην ολοκλήρωση της διπλωματικής.

Ακόμη θα επιθυμούσα να ευχαριστήσω όλα τα υπόλοιπα μέλη του ερευνητικού εργαστηρίου TelSiP για την εξαιρετική ατμόσφαιρα και φιλικότητα που μου προσέφεραν κάθε φορά που βρισκόμουν εκεί.

Ευχαριστώ τους γονείς μου Γεωργία και Νεκτάριο καθώς και τον σκύλο μου Κούπερ που βρίσκονταν πάντα στο πλευρό μου. Τέλος, ευχαριστώ όλους εκείνους τους ορισμένους αγαπητούς και αξιόλογους ανθρώπους που συνάντησα κατά τη διάρκεια των σπουδών μου και έκαναν τις μέρες μου καλύτερες.

## Περίληψη

Αντικείμενο της παρούσας διπλωματικής είναι η ανάπτυξη μεθοδολογιών μηχανικής μάθησης, επικεντρώνοντας σε αρχιτεκτονικές νευρωνικών δικτύων ειδικά προσαρμοσμένες στα μοναδικά χαρακτηριστικά του προβλήματος πρόβλεψης μελλοντικών τιμών μη γραμμικών χρονοσειρών. Ιδιαίτερη έμφαση δίνεται στο σχεδιασμό και την εφαρμογή αναδρομικών νευρωνικών δικτύων λόγω της ικανότητάς τους να διαχειρίζονται ακολουθιακά ιστορικά δεδομένα αλλά και να αντιλαμβάνονται τις σύνθετες χρονικές τους εξαρτήσεις. Η έρευνα ξεκινά με την ανάπτυξη ενός απλού αναδρομικού δικτύου που εκπαιδεύεται μέσω του αλγορίθμου οπισθοδιάδοσης στο χρόνο προκειμένου να πραγματοποιηθεί σύγκριση με τα εξελιγμένα μοντέλα που διατίθενται σε καθιερωμένες βιβλιοθήκες. Στη συνέχεια, αναλύονται οι ειδικές περιπτώσεις των μοντέλων Long Short-Term Memory (LSTM) και Gated Recurrent Unit (GRU) λόγω της ικανότητας αντιμετώπισης συγκεκριμένων προβλημάτων που δημιουργούνται κατά την εκπαίδευση αναδρομικών δικτύων. Τα μοντέλα εφαρμόστηκαν σε δυο απαιτητικές εφαρμογές χρονοσειρών, όπου αναλύθηκε το πρόβλημα πρόβλεψης του επόμενου χρονικού βήματος (Single Step Prediction) αλλά και το πρόβλημα πρόβλεψης μελλοντικής ακολουθίας βημάτων (Multi Step Prediction). Πιο συγκεκριμένα, πρώτα πραγματοποιήθηκε ο έλεγχος της ορθής λειτουργίας όλων των μοντέλων μέσω των συνθετικών δεδομένων της χασοτικής συνάρτησης Mackey – Glass, ενώ στη συνέχεια ακολούθησε εφαρμογή στην πρόβλεψη μελλοντικών τιμών κατανάλωσης ενέργειας ενός δημοσίου κτηρίου. Για λόγους σύγκρισης χρησιμοποιήθηκαν επίσης η γραμμική μέθοδος των ελαχίστων τετραγώνων και τα νευρωνικά δίκτυα τροφοδότησης πολλών επιπέδων (MLP). Σε κάθε περίπτωση, τα αποτελέσματα ανέδειξαν την υπεροχή των αναδρομικών νευρωνικών δικτύων στα προβλήματα πρόβλεψης χρονοσειρών, ενώ το μοντέλο GRU υπερείχε βάσει αποτελεσμάτων έναντι των υπολοίπων τεχνικών.

## Λέξεις – κλειδιά

Αλγόριθμοι Μηχανικής Μάθησης, Αναδρομικά Νευρωνικά δίκτυα, Πρόβλεψη Χρονοσειρών, Ενεργειακή Κατανάλωση Κτιρίων, Μονάδα LSTM, Μονάδα GRU

## **Abstract**

The subject of this thesis is the development of machine learning methodologies, focusing on neural network architectures specifically adapted to the unique characteristics of the problem of predicting future values of non-linear time series. Particular emphasis is placed on the design and implementation of recurrent neural networks due to their ability to handle sequential historical data but also to understand their complex temporal dependencies. The research begins with the development of a simple recurrent network that is trained through the backpropagation through time algorithm in order to perform a comparison with the sophisticated models available in established libraries. The special cases of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models are then analyzed due to their ability to deal with specific problems created when training recurrent networks. The models were applied to two demanding time series applications, where the problem of predicting the next time step (Single Step Prediction) and the problem of predicting a future sequence of steps (Multi Step Prediction) were analyzed. More specifically, the correct operation of all models was checked through the synthetic data of the chaotic Mackey-Glass function, while they were then applied to the prediction of future energy consumption values of a public building problem. Linear least squares and multi-layer neural networks (MLP) were also used for comparison. In any case, the results highlighted the superiority of recurrent neural networks in time series forecasting problems, while the GRU model outperformed the rest of the techniques based on the results.

## **Keywords**

Machine Learning Algorithms, Recurrent Neural Networks, Time Series Forecasting, Building Energy Consumption, LSTM Unit, GRU Unit

## Περιεχόμενα

<b>ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ</b> .....	10
<b>ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ</b> .....	12
<b>ΕΙΣΑΓΩΓΗ</b> .....	13
Αντικείμενο της διπλωματικής.....	13
Σκοπός και στόχοι.....	13
Σχετικές έρευνες .....	14
Καινοτομία .....	16
Δομή.....	16
<b>ΚΕΦΑΛΑΙΟ 1<sup>ο</sup> : ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ</b> .....	18
1.1 Εισαγωγή στην υπολογιστική νοημοσύνη.....	19
1.2 Μηχανική Μάθηση.....	20
1.3 Εισαγωγή στα τεχνητά νευρωνικά δίκτυα .....	23
1.4 Συναρτήσεις ενεργοποίησης.....	26
1.5 Αρχιτεκτονικές τεχνητών νευρωνικών δικτύων.....	30
1.5.1 Δίκτυα πρόσθιας τροφοδότησης ενός επιπέδου.....	30
1.5.2 Δίκτυα πρόσθιας τροφοδότησης πολλών επιπέδων.....	31
1.6 Συναρτήσεις απώλειας.....	32
1.7 Βελτιστοποιητές .....	35
1.7.1 Αλγόριθμος κατάβασης κλίσης .....	35
1.7.2 Αλγόριθμος προσαρμοστατικής εκτίμησης της ροπής (Adam).....	37
1.8 Οπισθοδιάδοση (Backpropagation).....	38
<b>ΚΕΦΑΛΑΙΟ 2<sup>ο</sup> : ΑΝΑΔΡΟΜΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ</b> .....	41
2.1 Απλά αναδρομικά νευρωνικά δίκτυα (Simple RNNs) .....	42
2.2 Οπισθοδιάδοση στο χρόνο .....	43
2.2.1 Οπισθοδιάδοση στο χρόνο με περικοπή .....	47
2.3 Εξαφανιζόμενες και Εκρηκτικές Κλίσεις .....	48
2.4 Περικοπή κλίσης.....	49
2.5 Long Short-Term Memory (LSTM) .....	50
2.6 Gated Recurrent Unit (GRU).....	53
2.7 Σύντομη σύγκριση μεταξύ LSTM και GRU .....	55
<b>ΚΕΦΑΛΑΙΟ 3<sup>ο</sup> : ΥΛΟΠΟΙΗΣΗ ΜΕΘΟΔΩΝ ΕΚΠΑΙΔΕΥΣΗΣ ΑΝΑΔΡΟΜΙΚΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΣΕ PYTHON</b> .....	57
3.1 Η γλώσσα προγραμματισμού Python .....	58
3.2 Ο περιηγητής Anaconda .....	58
3.3 Το περιβάλλον του Jupyter Notebook.....	58
3.4 Η πλατφόρμα της TensorFlow.....	59

3.5 Η Βιβλιοθήκη Keras.....	60
3.5.1 API Κατασκευής Μοντέλων στο Keras.....	60
3.6 Λοιπές απαραίτητες βιβλιοθήκες .....	61
3.7 Το μοντέλο των ελαχίστων τετραγώνων.....	62
3.8 Το Custom μοντέλο RNN .....	65
3.9 Κατασκευή μοντέλων με χρήση του Keras .....	70
3.9.1 Το μοντέλο MLP .....	70
3.9.2 Το μοντέλο απλού RNN .....	71
3.9.3 Το μοντέλο LSTM και GRU .....	71
<b>ΚΕΦΑΛΑΙΟ 4<sup>ο</sup>: CASE STUDY: ΕΦΑΡΜΟΓΕΣ ΑΝΑΔΡΟΜΙΚΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΣΤΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΧΡΟΝΟΣΕΙΡΩΝ.....</b>	<b>73</b>
4.1 Χρονοσειρές .....	73
4.2 Εφαρμογή 1 <sup>η</sup> : Πρόβλεψη τιμών συνάρτησης Mackey – Glass.....	74
4.2.1. Περιγραφή.....	74
4.2.2 Δεδομένα.....	74
4.2.3 Υλοποίηση .....	76
4.2.4 Αποτελέσματα .....	78
4.2.5 Συζήτηση .....	81
4.3 Εφαρμογή 2 <sup>η</sup> : Πρόβλεψη κατανάλωσης ενέργειας δημόσιου κτηρίου.....	81
4.3.1 Περιγραφή.....	81
4.3.2 Δεδομένα.....	82
4.3.3 Υλοποίηση .....	85
4.3.4 Αποτελέσματα .....	86
4.3.5 Συζήτηση.....	90
4.3.6 Πιθανή εφαρμογή προτεινόμενου μοντέλου.....	91
<b>ΣΥΜΠΕΡΑΣΜΑΤΑ.....</b>	<b>93</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ – ΑΝΑΦΟΡΕΣ – ΔΙΑΔΙΚΤΥΑΚΕΣ ΠΗΓΕΣ.....</b>	<b>95</b>

---

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

---

- Σχήμα 1:** Σχεδιάγραμμα επιβλεπόμενης μάθησης
- Σχήμα 2:** Σχεδιάγραμμα μη επιβλεπόμενης μάθησης
- Σχήμα 3:** Σχεδιάγραμμα ενισχυτικής μάθησης
- Σχήμα 4:** Απεικόνιση νευρικού κυττάρου
- Σχήμα 5:** Μη γραμμικό μοντέλο τεχνητού νευρώνα
- Σχήμα 6:** Συνάρτηση κατωφλιού
- Σχήμα 7:** Σιγμοειδής συνάρτηση για μεταβαλλόμενη παράμετρο κλίσης  $a$
- Σχήμα 8:** Συνάρτηση υπερβολικής εφαπτομένης
- Σχήμα 9:** Συνάρτηση διορθωμένης γραμμικής μονάδας
- Σχήμα 10:** Παράδειγμα δικτύου πρόσθιας τροφοδότησης ενός επιπέδου
- Σχήμα 11:** Παράδειγμα δικτύου πρόσθιας τροφοδότησης πολλών επιπέδων
- Σχήμα 12:** Αρχιτεκτονική Simple RNN σε συμπιεσμένη Folded μορφή
- Σχήμα 13:** Αρχιτεκτονική Simple RNN σε αναλυτική Unfolded μορφή
- Σχήμα 14:** Αναδίπλωση αρχιτεκτονικής RNN για κατάλληλη χρήση σε αλγόριθμο BPTT
- Σχήμα 15:** Γραφική αναπαράσταση λειτουργίας αλγορίθμων BPTT και TBPTT.
- Σχήμα 16:** Συμπεριφορά παραγώγου με και χωρίς τη χρήση της μεθόδου περικοπής κλίσης
- Σχήμα 17:** Απεικόνιση πλήρους μοντέλου LSTM
- Σχήμα 18:** Απεικόνιση πλήρους μοντέλου GRU
- Σχήμα 19:** Παραδείγματα μοντέλων κατασκευασμένα με Sequential API και Functional API
- Σχήμα 20:** Αναπαράσταση των δεδομένων της 1<sup>ης</sup> εφαρμογής
- Σχήμα 21:** Γράφημα γραμμικού δικτύου της πρώτης εφαρμογής
- Σχήμα 22:** Γράφημα Keras MLP δικτύου της πρώτης εφαρμογής
- Σχήμα 23:** Γράφημα Custom RNN δικτύου της πρώτης εφαρμογής
- Σχήμα 24:** Γράφημα Keras RNN δικτύου της πρώτης εφαρμογής
- Σχήμα 25:** Γράφημα Keras LSTM δικτύου της πρώτης εφαρμογής

**Σχήμα 26:** Γράφημα Keras GRU δικτύου της πρώτης εφαρμογής

**Σχήμα 27:** Αναπαράσταση των δεδομένων της 2<sup>ης</sup> εφαρμογής

**Σχήμα 28:** Απόσταση τοποθεσίας απόκτησης μετεωρολογικών δεδομένων από το δημαρχείο

**Σχήμα 29:** Αναπαράσταση ωριαίας μέσης θερμοκρασίας σύμφωνα με τα μετρητικά όργανα

**Σχήμα 30:** Αναπαράσταση ωριαίας αίσθησης της θερμοκρασίας (Feels Like Temperature)

**Σχήμα 31:** Αναπαράσταση ωριαίας μέσης υγρασίας

**Σχήμα 32:** Γράφημα γραμμικού δικτύου της δεύτερης εφαρμογής

**Σχήμα 33:** Γράφημα Keras MLP δικτύου της δεύτερης εφαρμογής

**Σχήμα 34:** Γράφημα Custom RNN δικτύου της δεύτερης εφαρμογής

**Σχήμα 35:** Γράφημα Keras RNN δικτύου της δεύτερης εφαρμογής

**Σχήμα 36:** Γράφημα Keras LSTM δικτύου της δεύτερης εφαρμογής

**Σχήμα 37:** Γράφημα Keras GRU δικτύου της δεύτερης εφαρμογής

**Σχήμα 38:** Προτεινόμενο Keras GRU μοντέλο από Multivariate μελέτη

**Σχήμα 39:** Γράφημα διασποράς μεταξύ των πραγματικών και προβλεπόμενων τιμών προτεινόμενου μοντέλου

**Σχήμα 40:** Παραδείγματα καλής λειτουργίας του μοντέλου ανά εποχή

**Σχήμα 41:** Ανίχνευση μη συνηθισμένης συμπεριφοράς στην ενεργειακή κατανάλωση



---

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

---

**Πίνακας 1:** Κατάλληλες συναρτήσεις απώλειας ανάλογα με τις απαιτήσεις του προβλήματος παλινδρόμησης

**Πίνακας 2:** Κατάλληλες συναρτήσεις απώλειας ανάλογα με τον τύπο του προβλήματος ταξινόμησης

**Πίνακας 3:** Μεταβλητές που θα χρησιμοποιηθούν στις εξισώσεις της ενότητας

**Πίνακας 4:** Διαστάσεις παραμέτρων ενός δικτύου LSTM

**Πίνακας 5:** Διαστάσεις παραμέτρων ενός δικτύου GRU

**Πίνακας 6:** Σύγκριση πολυπλοκότητας πράξεων μεταξύ LSTM και GRU

**Πίνακας 7:** Παράμετροι επιλεγμένου μοντέλου MLP της πρώτης εφαρμογής

**Πίνακας 8:** Παράμετροι αναδρομικών μοντέλων της πρώτης εφαρμογής

**Πίνακας 9:** Αποτελέσματα της πρώτης εφαρμογής

**Πίνακας 10:** Παράμετροι αναδρομικών μοντέλων της πρώτης εφαρμογής

**Πίνακας 11:** Παράμετροι προτεινόμενου μοντέλου GRU της δεύτερης εφαρμογής

**Πίνακας 12:** Αποτελέσματα Univariate μελέτης της δεύτερης εφαρμογής

**Πίνακας 13:** Αποτελέσματα Multivariate μελέτης της δεύτερης εφαρμογής

**Πίνακας 14:** Δεδομένα χρονικού διαστήματος παρατήρησης μη συνηθισμένης συμπεριφοράς

---

## ΕΙΣΑΓΩΓΗ

---

### Αντικείμενο της διπλωματικής

Τα νευρωνικά δίκτυα [1][2][3][4][5][6] αποτελούν ισχυρά υπολογιστικά μοντέλα που βασίζονται στη λειτουργία των βιολογικών νευρώνων του ανθρώπινου εγκεφάλου. Παρουσιάζουν την ικανότητα πρόσληψης γνώσης από το περιβάλλον μέσω μιας διαδικασίας μάθησης αλλά και αποθήκευσή της για την πραγματοποίηση μελλοντικών προβλέψεων. Η συγκεκριμένη διπλωματική εργασία βασίζεται στον εκτεταμένο όγκο έρευνας στους τομείς της βαθιάς μάθησης και της πρόβλεψης χρονοσειρών για τη δημιουργία μοντέλων που μπορούν να αποτυπώσουν περίπλοκες χρονικές εξαρτήσεις και μοτίβα ιστορικών δεδομένων. Θα δοθεί ιδιαίτερη έμφαση στα αναδρομικά νευρωνικά δίκτυα και σε ορισμένες παραλλαγές τους. Αυτά αποτελούν μια μεγάλη κατηγορία νευρωνικών δικτύων που ξεχωρίζουν λόγω της ύπαρξης ανάδρασης στην αρχιτεκτονική τους. Για την αξιολόγηση της απόδοσης των μοντέλων θα ερευνηθούν δυο απαιτητικές εφαρμογές. Η πρώτη αφορά τα συνθετικά δεδομένα μιας συνάρτησης Mackey – Glass. Αυτή αποτελεί σημείο αναφοράς στην ανάλυση της συμπεριφοράς χαοτικών συναρτήσεων ενώ η εγγενής μη περιοδική φύση της καθιστά εξαιρετικά δύσκολη τη πρόβλεψη μελλοντικών τιμών της. Η δεύτερη εφαρμογή αφορά πραγματικά δεδομένα κατανάλωσης ενέργειας ενός δημοσίου κτηρίου. Η κατανάλωση ενέργειας κτιρίων αποτελεί μια κρίσιμη πτυχή της βιώσιμης αστικής ανάπτυξης και της διατήρησης του περιβάλλοντος. Συγκεκριμένα ο Διεθνής Οργανισμός Ενέργειας (Δ.Ο.Ε.) έχει προσδιορίσει την βελτίωση της ενεργειακής απόδοσης κτιρίων ως ένα από τα πέντε μέτρα για τη διασφάλιση της μακροπρόθεσμης απαλλαγής από τις ανθρακούχες εκπομπές του ενεργειακού τομέα [7]. Η πρόβλεψη των μελλοντικών τιμών της κατανάλωσης αποτελεί σημαντικό παράγοντα των συστημάτων διαχείρισης ενέργειας κτιρίων (BEMS). Αυτά αποτελούν συστήματα ελέγχου τα οποία είναι σχεδιασμένα για την παρακολούθηση, διαχείριση και βελτιστοποίηση διάφορων υπηρεσιών με σκοπό την όσο το δυνατότερον καλύτερη κατανομή της ηλεκτρικής ενέργειας. Πέρα από τους περιβαλλοντικούς και οικονομικούς παράγοντες, η πρόβλεψη της κατανάλωσης μπορεί να συμβάλλει στη σταθερότητα και αξιοπιστία του γενικότερου ηλεκτρικού δικτύου αφού μπορεί να αποτελέσει ένα δείκτη εξισορρόπησης της απαιτούμενης προσφοράς και ζήτησης του ρεύματος για όλες τις πιθανές περιπτώσεις.

### Σκοπός και στόχοι

Σκοπός της συγκεκριμένης διπλωματικής είναι η ανάπτυξη και η κατάλληλη εφαρμογή μοντέλων αναδρομικών νευρωνικών δικτύων με στόχο τη πρόβλεψη μελλοντικών τιμών

χρονοσειρών. Η ανάλυση αφορά το πρόβλημα πρόβλεψης του αμέσως επόμενου χρονικού βήματος αλλά και μιας αλληλουχίας μελλοντικών χρονικών βημάτων. Περιλαμβάνονται όμως και άλλα στοιχεία που συμβάλλουν στην πραγματοποίηση αυτού το πρωταρχικού σκοπού. Συγκεκριμένα:

- Διεξάγεται μια εκτενής ανασκόπηση της υπάρχουσας βιβλιογραφίας και δημιουργείται μια ολοκληρωμένη θεωρητική βάση για την κατανόηση των αρχών και αλγορίθμων εκπαίδευσης των αναδρομικών νευρωνικών δικτύων.
- Αναφέρονται και επεξηγούνται οι απαραίτητες πλατφόρμες και τα εργαλεία προγραμματισμού για την επιτυχή ανάπτυξη και υλοποίηση όλων των δικτύων.
- Πραγματοποιείται ανάπτυξη ενός απλού αναδρομικού δικτύου που εκπαιδεύεται μέσω του αλγορίθμου οπισθοδιάδοσης στο χρόνο προκειμένου να πραγματοποιηθεί σύγκριση με τα εξελιγμένα μοντέλα που διατίθενται σε καθιερωμένες βιβλιοθήκες. Αυτό συμβάλλει στην ανάδειξη πλεονεκτημάτων και περιορισμών μεταξύ των μεθόδων ανάπτυξης.
- Αναλύονται τεχνικές προεπεξεργασίας δεδομένων για την προετοιμασία εισαγωγής τους στο μοντέλο.
- Διερευνώνται μέθοδοι μηχανικής χαρακτηριστικών (Feature Engineering Methods) με σκοπό την ενίσχυση της καταγραφής πολύπλοκων μοτίβων στα δεδομένα κατανάλωσης ενέργειας των μοντέλων. Εξερευνάται επίσης η ενσωμάτωση εξωγενών παραγόντων, όπως δεδομένα καιρού για τη βελτίωση ακρίβειας της πρόβλεψης.
- Διεξάγονται εκτενείς διαδικασίες αξιολόγησης και σύγκρισης μεταξύ των αναδρομικών μοντέλων ενώ στο τέλος παρέχεται η τελική πρόταση αρχιτεκτονικής. Επιπλέον, παρέχονται επιδόσεις παραδοσιακών γραμμικών μοντέλων αλλά και μοντέλων πρόσθιας τροφοδότησης πολλών επιπέδων.

### Σχετικές έρευνες

Παρόλο που το πλήθος ερευνών πάνω στο τομέα των αναδρομικών νευρωνικών δικτύων έχει αποκτήσει ιδιαίτερη αύξηση τα τελευταία χρόνια, οι ιδέες και αξίες που τα περιγράφουν δεν αποτελούν κάτι καινούριο. Η ικανότητά τους να αποσπούν χρονικές εξαρτήσεις τα έχει κάνει να αποτελούν μια ιδανική επιλογή για προβλήματα πρόβλεψης χρονοσειρών. Σε ένα άρθρο του πανεπιστημίου Monash της Μελβούρνης [\[8\]](#) δημοσιεύτηκε μια εκτενής έρευνα που αφορούσε πολυάριθμες πρακτικές χρήσης των αναδρομικών νευρωνικών δικτύων πάνω σε

δεδομένα χρονοσειρών. Η συγκεκριμένη έρευνα σύγκρινε την αποδοτικότητά τους σε σχέση με υπάρχουσες στατιστικές μεθόδους αξιοποιώντας πολλαπλές βάσεις δεδομένων.

Ωστόσο, η εφαρμογή τους σε προβλήματα πρόβλεψης ενεργειακής κατανάλωσης κτηρίων περιλαμβάνει παραπάνω πτυχές από τον σχεδιασμό ενός μοντέλου. Η πρόβλεψη κατανάλωσης ενός κτηρίου αποτελεί ένα σύνθετο πρόβλημα το οποίο επηρεάζεται από πολυάριθμες μεταβλητές. Η έρευνα έχει δείξει ότι η μηχανική χαρακτηριστικών (Feature engineering) μπορεί να προσφέρει σημαντικά οφέλη βελτιώνοντας την ακρίβεια και παρέχοντας πληροφορίες για τα φαινόμενα που επηρεάζουν την συμπεριφορά των χρονοσειρών. Σε μια μελέτη του πανεπιστημίου Guangzhou στην Κίνα [9], πραγματοποιήθηκε αρχικά μια σύνοψη που αφορούσε το απαραίτητο θεωρητικό υπόβαθρο και τις βασικές πρακτικές μετασχηματισμού εισόδων μέσω του feature engineering, ενώ στη συνέχεια εξετάστηκε το πως αυτές μπορούν να εφαρμοστούν στην πρόβλεψη της κατανάλωσης. Επιπλέον, το ίδιο πανεπιστήμιο δημοσίευσε ακόμη μια έρευνα [10] που συγκέντρωσε δεδομένα από 116 διαφορετικά ερευνητικά άρθρα χρονολογιών από το 2012 έως το 2022 με σκοπό να αναγνωριστούν και να συγκριθούν όσο το δυνατόν περισσότεροι εξωγενείς παράγοντες που μπορούν να επηρεάσουν την απόδοση των μοντέλων. Τα αποτελέσματα έδειξαν ότι η πλειοψηφία των ερευνών βασίστηκε κυρίως σε μετεωρολογικά δεδομένα.

Τα τελευταία χρόνια έχει αρχίσει να δημοσιεύεται ένα μεγάλο πλήθος ερευνών που αφορούν τη μελέτη της απόδοσης των αναδρομικών δικτύων στη συγκεκριμένη εφαρμογή [11] [12] [13]. Το πρώτο άρθρο προβλέπει τη κατανάλωση ενός κτηρίου που ανήκει σε κάποιο πανεπιστήμιο της Μαλαισίας και προτείνει ένα LSTM δίκτυο βασισμένο σε μετεωρολογικά δεδομένα. Το δεύτερο περιλαμβάνει δεδομένα κατανάλωσης ενός χώρου εργασίας στην Ινδία και προτείνει ένα LSTM δίκτυο το οποίο παρουσιάζει καλύτερη απόδοση όταν συγκρίνεται με μηχανές υποστήριξης διανύσματος (SVM, Support Vector Machines [14]). Το τελευταίο άρθρο χρησιμοποίησε δεδομένα κατανάλωσης ενός δημόσιου κτηρίου και παρουσίασε ένα μοντέλο GRU το οποίο απέδιδε καλύτερα σε σύγκριση με άλλες παραλλαγές αναδρομικών δικτύων ενώ χρησιμοποιούσε μια τεχνική dropout [15] προκειμένου να βελτιώσει την ακρίβειά του. Με αυτόν τον τρόπο ενισχύεται η ικανότητα γενίκευσης του μοντέλου καθώς πραγματοποιείται τυχαία απενεργοποίηση συνάψεων για την απομάκρυνση πιθανών πολώσεων. Αυτό που παρατηρείται στη διαθέσιμη βιβλιογραφία είναι ότι η πλειοψηφία των ερευνών αφορά εμπορικά και δημόσια κτήρια. Αυτό είναι λογικό καθώς είναι πιθανότερο να έχει πραγματοποιηθεί η εγκατάσταση του κατάλληλου εξοπλισμού καταγραφής δεδομένων και

διαχείρισης ενέργειας σε σχέση με μια ιδιωτική κατοικία. Επιπλέον, πολλές φορές τα δεδομένα δημοσιεύονται και είναι προβιβάσιμα από τους ενδιαφερόμενους ερευνητές.

Σε πολλές περιπτώσεις ενδέχεται τα αναδρομικά μοντέλα να συνδυαστούν με κάποια άλλη μέθοδο ή κατηγορία δικτύου. Οι έρευνες [16] και [17] προτείνουν ένα υβριδικό δίκτυο που χρησιμοποιεί την ικανότητα των συνελκτικών νευρωνικών δικτύων (CNN, Convolutional Neural Networks [18]) να αναγνωρίζουν χωρικά χαρακτηριστικά μέσω πληροφοριών που δέχεται από ένα ‘έξυπνο’ κτήριο. Με αυτόν τον τρόπο εισάγεται ένα συνελκτικό επίπεδο πριν το αναδρομικό παρέχοντας καλύτερη κατανόηση των ιδιοτήτων των δεδομένων. Η πρώτη έρευνα προτείνει ένα CNN - LSTM δίκτυο ενώ η δεύτερη ένα CNN - GRU δίκτυο. Σε κάθε περίπτωση παρουσιάζουν καλύτερη απόδοση από τις κανονικές εκδοχές τους. Η έρευνα [19] προχωράει αυτή τη σκέψη ένα βήμα παραπέρα και προτείνει έναν συνδυασμό μεθόδων μεταξύ ενός μη επιβλεπόμενου k-means μοντέλου [20] και ενός υβριδικού CNN – LSTM δικτύου. Με αυτόν τον τρόπο τα χωρικά δεδομένα μπορούν να διαιρεθούν σε ομάδες ανάλογα με την ομοιότητά τους και να υπάρξει καλύτερη κατανόηση των παραγόντων που επηρεάζουν την κατανάλωση.

### **Καινοτομία**

Η παρούσα έρευνα αποσκοπεί στην ανάπτυξη μοντέλων αναδρομικών νευρωνικών δικτύων για την εφαρμογή τους στα σύνθετα προβλήματα πρόβλεψης μελλοντικών τιμών μιας χρονοσειράς Mackey – Glass αλλά και μιας χρονοσειράς που περιγράφει την κατανάλωση ενέργειας ενός κτηρίου. Συγκεκριμένα, έχουν επιλεγθεί δεδομένα που αφορούν το δημαρχείο του Αιγάλεω κάνοντας χρήση όλων των εξωγενών παραγόντων που μπορούν να αποκτηθούν. Δίνεται ιδιαίτερη έμφαση στην ερμηνευτικότητα και στην επεξήγηση των μοντέλων ενώ πραγματοποιείται και η ανάπτυξη ενός απλού αναδρομικού δικτύου από το μηδέν για τη σύγκριση της απόδοσής του με τα μοντέλα των καθιερωμένων βιβλιοθηκών. Η διπλωματική θα προστεθεί στη συνολική βιβλιογραφία ενισχύοντας τον τομέα έρευνας των αναδρομικών δικτύων πάνω στις συγκεκριμένες εφαρμογές.

### **Δομή**

Η διπλωματική εργασία αποτελείται από τέσσερα (4) βασικά κεφάλαια καθώς και μία ενότητα όπου παρατίθενται τα συμπεράσματα που εξήχθησαν από το σύνολο της έρευνας.

Στο πρώτο κεφάλαιο πραγματοποιείται μια σύντομη παρουσίαση της υπολογιστικής νοημοσύνης, ενώ στη συνέχεια αναλύεται διεξοδικά ένα κομμάτι της που ονομάζεται τεχνητά νευρωνικά δίκτυα. Αναφέρονται οι αρχές λειτουργίας τους, καθώς και ορισμένες από τις πιο

σημαντικές αρχιτεκτονικές τους. Επιπλέον, δίνεται ιδιαίτερη σημασία στη μέθοδο εκπαίδευσής τους μέσω των αλγορίθμων μηχανικής μάθησης. Το κεφάλαιο περιλαμβάνει επίσης ένα μεγάλο αριθμό αποφάσεων που πρέπει να παρθούν κατά το σχεδιασμό ενός μοντέλου τεχνητών νευρωνικών δικτύων παρέχοντας πολυάριθμες εναλλακτικές.

Στο δεύτερο κεφάλαιο μελετώνται διεξοδικά τα αναδρομικά νευρωνικά δίκτυα, τα οποία αποτελούν και τον κορμό της συγκεκριμένης διπλωματικής εργασίας. Μετά από μια σύντομη εισαγωγή των ικανοτήτων τους, αναλύεται η δομή και η λειτουργία της πιο απλής τους περίπτωσης. Παρουσιάζεται ένας από τους πιο γνωστούς αλγορίθμους εκπαίδευσης τους καθώς και τα προβλήματα που προκύπτουν μέσω της χρήσης του. Στη συνέχεια, παρέχονται κάποιες από τις πιο γνωστές πρακτικές για την αντιμετώπιση αυτών των προβλημάτων και πραγματοποιείται μια σύντομη σύγκριση μεταξύ τους. Αυτές οι πρακτικές μπορεί να έχουν είτε μορφή αλγορίθμου όπως αυτός της περικοπής κλίσης είτε να αποτελούν νέες αρχιτεκτονικές κρυφής μονάδας όπως αυτές των LSTM και GRU.

Το τρίτο κεφάλαιο περιέχει μια σύντομη παρουσίαση των εργαλείων και προγραμμάτων που χρησιμοποιήθηκαν. Επιπλέον, παρέχεται η ανάλυση όλων βασικών αρχών και μεθοδολογιών που αξιοποιούνται από το κάθε μοντέλο κατά την κατασκευή του.

Το τέταρτο και τελευταίο κεφάλαιο αποτελεί το πειραματικό κομμάτι της διπλωματικής. Στην αρχή πραγματοποιείται μια σύντομη ανάλυση των χρονοσειρών και της διάταξης των δεδομένων σε αυτές. Περιλαμβάνονται δύο εφαρμογές. Η πρώτη βασίζεται στη σύγκριση μοντέλων μέσω των συνθετικών δεδομένων μιας συνάρτησης Mackey – Glass ενώ η δεύτερη αφορά τη βασική μελέτη του δημόσιου κτηρίου. Παρατίθενται όλα τα αποτελέσματα των προτεινόμενων μοντέλων καθώς και τα γραφήματα που τα αναπαριστούν. Στο τέλος παρέχεται μια αναλυτική σύγκριση αλλά και ο σχολιασμός της λειτουργίας τους.

---

## ΚΕΦΑΛΑΙΟ 1<sup>ο</sup> : ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

---

Η νοημοσύνη αποτελεί μια από τις αρχαιότερες έννοιες που έχουν απασχολήσει την ανθρωπότητα. Τα βασικότερα ερωτήματα έχουν να κάνουν τόσο με τον τρόπο λειτουργίας της, όσο και με την ικανότητα του ανθρώπου να την αναπτύξει. Η εφεύρεση του ηλεκτρονικού υπολογιστή αποτέλεσε τον σημαντικότερο σταθμό στην εξέλιξη όλων των επιστημών, καθώς περιέγραφε ένα μηχάνημα ικανό να εκτελεί πολυάριθμους υπολογισμούς αλλά και να διαχειρίζεται τεράστιες βάσεις δεδομένων χρησιμοποιώντας συγκεκριμένους αλγορίθμους. Για πρώτη φορά μετά από χρόνια, η δημιουργία ευφών συστημάτων φαινόταν ως κάτι παραπάνω από μια υπόθεση. Σύμφωνα με τον Krishnakumar,

*“ Ένα ευφές σύστημα είναι ένα σύστημα που μιμείται ορισμένες πτυχές της νοημοσύνης που επιδεικνύει η φύση. Αυτά περιλαμβάνουν τη μάθηση, την προσαρμοστικότητα, την ευρωστία σε όλους τους τομείς προβλημάτων, τη βελτίωση της αποτελεσματικότητας (με την πάροδο του χρόνου ή/και του χώρου), τη συμπίεση πληροφοριών (δεδομένα σε γνώση) και την προεκβολή συλλογισμού [21]. ”*

Επιπλέον, σύμφωνα με το λεξικό του Cambridge, η νοημοσύνη ορίζεται ως:

*“ Η ικανότητα να μαθαίνεις, να κατανοείς και να κρίνεις ή να έχεις απόψεις που βασίζονται στη λογική [22]. ”*

Αυτοί οι ορισμοί δεν είναι σε καμία περίπτωση απόλυτοι. Οι έννοιες συνδέονται με την γνωστική λειτουργία του ανθρώπου η οποία ακόμη και σήμερα δεν έχει κατανοηθεί πλήρως. Ο Alan Turing υποστήριζε πως η ερώτηση δεν θα έπρεπε να είναι ‘ Μπορούν τα μηχανήματα να σκεφτούν;’ αλλά ‘ Μπορούν να περάσουν ένα τεστ συμπεριφοράς για ευφυΐα;’. Πίστευε πως η νοημοσύνη ενός συστήματος έχει να κάνει με το αν αυτό είναι ικανό να περάσει γνωστικές εργασίες. Έτσι το 1950, δημιούργησε το Turing test, όπου ένας εξεταστής κάνει ερωτήσεις σε έναν άνθρωπο και μια μηχανή μέσω ενός τερματικού χωρίς να γνωρίζει τι είναι ο καθένας. Αν στο τέλος του τεστ δεν καταφέρει να ξεχωρίσει τον άνθρωπο από την μηχανή, τότε η μηχανή μπορεί να θεωρηθεί ευφυής.

Το 1956 δημιουργήθηκε ένα ερευνητικό εργαστήριο στο πανεπιστήμιο Dartmouth το οποίο αποτελούταν από μόλις 10 μέλη. Εκεί ο John McCarthy δημιούργησε για πρώτη φορά τον όρο Τεχνητή Νοημοσύνη (Artificial Intelligence). Η τεχνητή νοημοσύνη αποτελεί έναν τομέα ο οποίος ασχολείται με τη σχεδίαση ευφών υπολογιστικών συστημάτων τα οποία είναι ικανά να παρουσιάσουν χαρακτηριστικά ανθρώπινης νοημοσύνης [23].



Τα νευρωνικά δίκτυα αποτελούν μια διαδικασία μοντελοποίησης των βιολογικών νευρώνων του εγκεφάλου έτσι ώστε να πραγματοποιηθεί η δημιουργία ενός ευφυούς συστήματος. Στόχος του κεφαλαίου είναι να παρουσιαστούν οι βασικές τους έννοιες καθώς και ορισμένες αρχιτεκτονικές και τρόποι λειτουργίας τους.

### 1.1 Εισαγωγή στην υπολογιστική νοημοσύνη

Η υπολογιστική νοημοσύνη αποτελεί μια εναλλακτική προσέγγιση του στόχου της τεχνητής νοημοσύνης. Αποτελεί ένα σύνολο μεθοδολογιών που μπορούν να μιμηθούν μηχανισμούς που παρατηρούνται στη φύση και να εφαρμοστούν μέσω υπολογιστικών συστημάτων. Αυτές μπορούν να προσεγγίσουν την ικανότητα λήψης ορθολογικών αποφάσεων σε ένα αβέβαιο και ανακριβές περιβάλλον. Διαφοροποιείται από την συμβολική τεχνητή νοημοσύνη με 2 τρόπους:

- Το βασικότερο χαρακτηριστικό της είναι η προσαρμογή. Αποτελείται από μεθοδολογίες ικανές να προσεγγίσουν με νέους τρόπους διαφορετικά προβλήματα.
- Επεξεργάζεται αποκλειστικά αριθμητικά δεδομένα. Σε περίπτωση που αυτά είναι μη αριθμητικά, πραγματοποιείται η κατάλληλη αναγωγή. Αντιθέτως, η συμβολική τεχνητή νοημοσύνη επεξεργάζεται σύμβολα που αντιπροσωπεύουν έννοιες και συνδέσεις με αποτέλεσμα να απαιτείται χρήση εξειδικευμένων γλωσσών προγραμματισμού όπως η LISP και η PROLOG.

Η κλασική υπολογιστική νοημοσύνη αποτελείται από 3 μεθοδολογίες ικανές να προσομοιώσουν παραμετρικά κάποιο χαρακτηριστικό της νοημοσύνης που παρατηρείται στη φύση. (α) Τα τεχνητά νευρωνικά δίκτυα είναι ικανά να μιμηθούν την ικανότητα του εγκεφάλου να μαθαίνει μέσω της λήψης και αποστολής δεδομένων με το περιβάλλον του. (β) Τα ασαφή συστήματα μπορούν να προσομοιώσουν την ανθρώπινη εννοιολογική επικοινωνία έτσι ώστε να πραγματοποιείται λήψη αποφάσεων και εξαγωγή συμπερασμάτων σε συνθήκες αβεβαιότητας. (γ) Οι εξελικτικοί αλγόριθμοι ακολουθούν τους κανόνες της φυσικής επιλογής και εξέλιξης προκειμένου να επιλύσουν προβλήματα χρησιμοποιώντας βέλτιστες λύσεις.

Αντίστοιχα υπάρχει και η διευρυμένη (Enhanced) υπολογιστική νοημοσύνη. Αποτελεί ένα συνεργατικό συνδυασμό ή/και επιλεκτική αντικατάσταση κάποιων μεθοδολογιών των αλγορίθμων της κλασικής υπολογιστικής νοημοσύνης. Μερικά από τα πιο γνωστά παραδείγματα είναι τα νευρο-ασαφή συστήματα (Neuro-fuzzy Systems), οι μηχανές διανυσμάτων στήριξης (Support Vector Machines), τα δίκτυα ακτινωτής βάσης (Radial Basis Functions) και οι γνωσιακοί χάρτες (Cognitive Maps).

## 1.2 Μηχανική Μάθηση

Η μηχανική μάθηση περιγράφει έναν τομέα της υπολογιστικής νοημοσύνης που έχει ως στόχο τη δημιουργία αλγορίθμων οι οποίοι είναι ικανοί να συγκεντρώσουν γνώση χρησιμοποιώντας κατάλληλα τα διαθέσιμα δεδομένα. Η γνώση αποθηκεύεται στις ελεύθερες παραμέτρους του μοντέλου που επιλέγεται ενώ οι αλγόριθμοι της μηχανικής μάθησης χρησιμοποιούν τις απαραίτητες διαδικασίες έτσι ώστε αυτή να είναι όσο το δυνατόν αποδοτικότερη.

Ο σχεδιασμός ενός τέτοιου αλγορίθμου μάθησης λαμβάνει υπόψιν του ένα πλήθος χαρακτηριστικών και κανόνων. Σύμφωνα με τον T. M. Mitchell, ένας αλγόριθμος θεωρείται ότι μαθαίνει από την εμπειρία  $E$  σε σχέση με κάποια κατηγορία εργασιών  $T$  και το μέτρο απόδοσης  $P$ , εάν η απόδοσή του σε εργασίες στο  $T$ , όπως μετριέται από το  $P$ , βελτιώνεται με την εμπειρία  $E$ . Ένα μαθησιακό πρόβλημα απαιτεί μια καλά καθορισμένη εργασία, ένα μετρητικό απόδοσης και μια πηγή εκπαιδευτικής εμπειρίας. Η προσέγγιση της μηχανικής μάθησης θα πρέπει να περιγράφεται από σχεδιαστικές αποφάσεις που έχουν να κάνουν με την επιλογή του τύπου της εκπαιδευτικής εμπειρίας, την επιλογή και αναπαράσταση της συνάρτησης στόχου αλλά και την επιλογή ενός αλγορίθμου για την εκμάθηση της συνάρτησης στόχου μέσα από παραδείγματα εκπαίδευσης [24].

Ο τρόπος που ο αλγόριθμος αποκτά τη γνώση, καθορίζει και την κατηγορία μάθησης που τον περιγράφει. Η μηχανική μάθηση περιλαμβάνει 3 βασικές κατηγορίες:

- Επιβλεπόμενη μάθηση (Supervised Learning)
- Μη επιβλεπόμενη μάθηση (Unsupervised Learning)
- Ενισχυμένη μάθηση (Reinforced Learning)

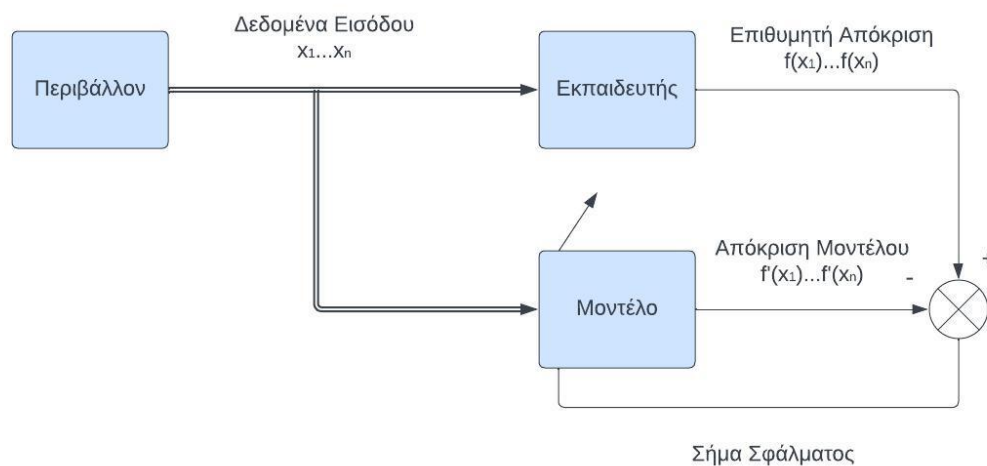
Κάθε μια από αυτές τις κατηγορίες είναι κατάλληλη για την επίλυση συγκεκριμένων τύπων προβλημάτων.

### ΕΠΙΒΛΕΠΟΜΕΝΗ ΜΑΘΗΣΗ

Σε έναν τυπικό αλγόριθμο επιβλεπόμενης μάθησης, στόχος είναι η εκτίμηση μιας συνάρτησης  $f: R^N \rightarrow T$  δοθέντος ενός συνόλου ζευγών  $(x_1, f(x_1), \dots, x_n, f(x_n))$ . Το πεδίο ορισμού  $R$  αντιπροσωπεύει το σύνολο των πραγματικών αριθμών ενώ το πεδίο τιμών  $T$  μπορεί να είναι είτε ένα αριθμητικό σύνολο  $T = R^M$  είτε ένα σύνολο ετικετών (Label). Αποτελεί το ρόλο του εκπαιδευτή και την απαραίτητη γνώση που πρέπει να μεταφερθεί στο μοντέλο. Η διαδικασία υπολογισμού αυτής της βέλτιστης προσέγγισης  $\hat{f}: R^N \rightarrow T$  για  $(x_1, f(x_1), \dots, x_n, f(x_n))$  ονομάζεται μάθηση ή, ισοδύναμα, εκπαίδευση. Η διαδικασία υπολογισμού της τιμής  $\hat{f}(x_0)$  για

$x_0 \in R^N$  με  $x_0 \neq x_i$  και  $i \in \{1, \dots, n\}$ , ονομάζεται γενίκευση. Μια διαδικασία μάθησης στην υπολογιστική νοημοσύνη θα πρέπει να επιτυγχάνει και μια αποδεκτή ικανότητα γενίκευσης [25]. Στο σχήμα 1 περιλαμβάνεται ένα σχεδιάγραμμα που περιγράφει την επιβλεπόμενη μάθηση.

Το πλεονέκτημα της επιβλεπόμενης μάθησης, είναι ότι χρησιμοποιείται σε προβλήματα που παρέχονται οι επιθυμητές τιμές και έτσι δίνεται η δυνατότητα αξιολόγησης της ικανότητας του μοντέλου να τροποποιεί σωστά τις ελεύθερες παραμέτρους του. Όταν οι επιθυμητές τιμές της συνάρτησης είναι διακριτές, τότε περιγράφεται ένα πρόβλημα ταξινόμησης (Classification) των δεδομένων. Στόχος είναι η κατανομή των δεδομένων εισόδου σε κάποια από τις κλάσεις που δημιουργούνται. Αντίστοιχα, αν οι επιθυμητές τιμές της συνάρτησης είναι συνεχείς ή επ' άπειρον διακριτές, τότε περιγράφεται ένα πρόβλημα παλινδρόμησης (Regression) των δεδομένων. Επιδιώκεται η όσο το δυνατόν καλύτερη προσαρμογή των δεδομένων στις επιθυμητές τιμές.

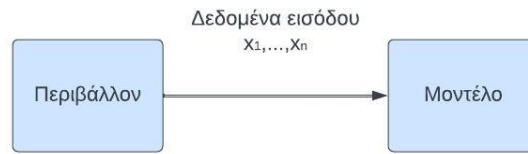


Σχήμα 1: Σχεδιάγραμμα επιβλεπόμενης μάθησης

## ΜΗ ΕΠΙΒΛΕΠΟΜΕΝΗ ΜΑΘΗΣΗ

Στη μη επιβλεπόμενη μάθηση παρουσιάζονται μόνο δεδομένα εισόδου. Η γνώση αποκτάται μέσω της ανάλυσης των δεδομένων και της εξαγωγής τάσεων και μοτίβων που τα περιγράφουν. Χρησιμοποιείται κυρίως σε προβλήματα ομαδοποίησης (Clustering) όπου σκοπός είναι η διαίρεση των δεδομένων σε κλάσεις ανάλογα με την ομοιότητα αυτά έχουν μεταξύ τους. Λόγω της έλλειψης του εκπαιδευτή, δεν δίνεται η ικανότητα αξιολόγησης του

μοντέλου. Η μη επιβλεπόμενη μάθηση αποσκοπεί στην απόκτηση χρήσιμων πληροφοριών από τα δεδομένα. Στο σχήμα 2 περιλαμβάνεται ένα σχεδιάγραμμα που περιγράφει την μη επιβλεπόμενη μάθηση.

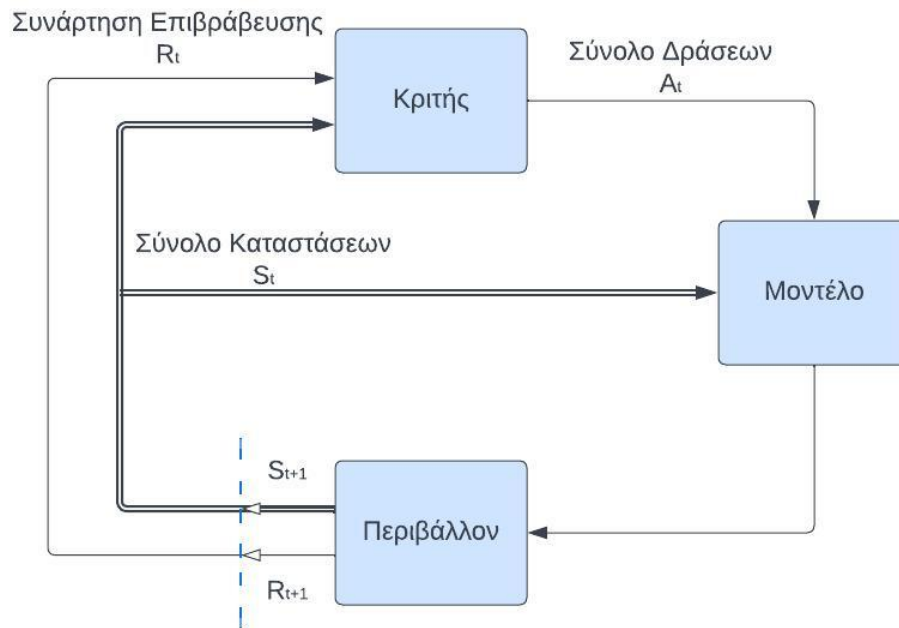


**Σχήμα 2:** Σχεδιάγραμμα μη επιβλεπόμενης μάθησης

### ΕΝΙΣΧΥΤΙΚΗ ΜΑΘΗΣΗ

Αποτελεί την κατηγορία μηχανικής μάθησης που ασχολείται με προβλήματα που απαιτούν τη διαδοχική λήψη αποφάσεων που βασίζονται στην εμπειρία του παρελθόντος. Περιγράφεται από χαρακτηριστικά όπως η παρουσία της αιτίας και του αποτελέσματος, η αβεβαιότητα και ο μη ντετερμινισμός. Η βασική ιδέα έχει να κάνει με την εμφάνιση ενός κριτή (Agent). Αυτός παρουσιάζει την ικανότητα μάθησης μέσω της αλληλεπίδρασης με το περιβάλλον προκειμένου να είναι σε θέση να κρίνει αν μια συμπεριφορά του μοντέλου πρέπει να ανταμειφθεί ή να τιμωρηθεί. Κύριος στόχος του είναι να μάθει μια καλή συμπεριφορά που θα τον οδηγήσει στην επίτευξη των στόχων του. Αυτό σημαίνει ότι τροποποιεί ή αποκτά νέες συμπεριφορές σταδιακά. Επιπλέον, χρησιμοποιεί τεχνικές δοκιμής και σφάλματος χωρίς να έχει κάποια πλήρη γνώση του περιβάλλοντος από πριν [26]. Στο σχήμα 3 περιλαμβάνεται ένα σχεδιάγραμμα που περιγράφει την ενισχυτική μάθηση.

Ο τρόπος λειτουργίας της ενισχυτικής μάθησης οδήγησε τον Thrun [27] να εισάγει το δίλημμα της εξερεύνησης – εκμετάλλευσης (Exploration – Exploitation Dilemma). Η εξερεύνηση αφορά τη λήψη νέων πληροφοριών από το περιβάλλον ενώ η εκμετάλλευση έχει να κάνει με τη μεγιστοποίηση της απόδοσης της τρέχουσας γνώσης και την επιλογή της πιο πολλά υποσχόμενης στρατηγικής. Η κατάλληλη ισορροπία μεταξύ αυτών δεν είναι πάντα εύκολο να βρεθεί. Η συνεχής εξερεύνηση μειώνει την ανταμοιβή του κριτή ενώ η συνεχής εκμετάλλευση μπορεί να αποκρύψει στρατηγικές που δεν έχουν ακόμα ανακαλυφθεί.



**Σχήμα 3:** Σχεδιάγραμμα ενισχυτικής μάθησης

Το σύνολο των καταστάσεων  $S$  περιγράφει ένα σύνολο δεδομένων που περιγράφουν το περιβάλλον που αλληλοεπιδρά ο κριτής. Η συνάρτηση επιβράβευσης  $R$  ορίζει την ανταμοιβή που προσφέρει η συγκεκριμένη κατάσταση. Στόχος του αλγορίθμου είναι η όσο το δυνατόν καλύτερη μεγιστοποίησή της. Αύξησή της σημαίνει επιβράβευση ενώ μείωσή της τιμωρία. Το σύνολο καταστάσεων και η συνάρτηση επιβράβευσης μιας συγκεκριμένης χρονικής στιγμής αναλύονται από τον κριτή ο οποίος καθορίζει ένα σύνολο δράσεων  $A$ . Αυτό το σύνολο ορίζει το πως θα επηρεαστεί το περιβάλλον από τον κριτή. Η δράση μιας ορισμένης χρονικής στιγμής  $t$ , θα επηρεάσει την κατάσταση και την επιβράβευση της επόμενης χρονικής στιγμής  $t+1$ . Το μοντέλο είναι υπεύθυνο για την αποθήκευση της γνώσης και την προσαρμογή του στα νέα δεδομένα.

### 1.3 Εισαγωγή στα τεχνητά νευρωνικά δίκτυα

Οι ηλεκτρονικοί υπολογιστές έχουν εξελιχθεί σε τέτοιο βαθμό όπου είναι ικανοί να εκτελούν αριθμητικούς υπολογισμούς γρηγορότερα και αποδοτικότερα από κάθε άνθρωπο. Επιπλέον, έχουν την ικανότητα αξιόπιστης αποθήκευσης ενός τεράστιου όγκου δεδομένων παρέχοντας την δυνατότητα ανεμπόδιστης πρόσβασής τους από τον χρήστη. Αυτό οφείλεται στο γεγονός του ότι αποτελούνται από πολύπλοκες δομικές μονάδες που συνδέονται μεταξύ τους σειριακά και χωρίζονται σε κατάλληλα τμήματα. Παρ' όλ' αυτά αδυνατούν ακόμα και μέχρι σήμερα να εκτελέσουν διεργασίες που αβίαστα διαχειρίζεται ο ανθρώπινος εγκέφαλος κάθε χρονικής στιγμής της ημέρας όπως η αναγνώριση προτύπων, η αντίληψη και ο έλεγχος της κίνησης.

Ο εγκέφαλος αποτελεί ένα εξαιρετικά πολύπλοκο, μη γραμμικό, παράλληλο σύστημα επεξεργασίας πληροφοριών. Αποτελεί ένα βιολογικό νευρικό σύστημα ικανό να δέχεται ερεθίσματα από το περιβάλλον μέσω υποδοχέων, να τα επεξεργάζεται κατάλληλα και στη συνέχεια να αποδίδει αποκρίσεις μέσω των δραστικών κυττάρων. Τα βασικά δομικά στοιχεία των βιολογικών νευρικών συστημάτων είναι τα νευρικά κύτταρα, που αναφέρονται ως νευρώνες και φαίνονται στο σχήμα 4. Αυτοί παρουσιάζουν ένα εξαιρετικά μεγάλο επίπεδο συνδεσιμότητας και αλληλεπίδρασης μεταξύ τους.



**Σχήμα 4:** Απεικόνιση νευρικού κυττάρου

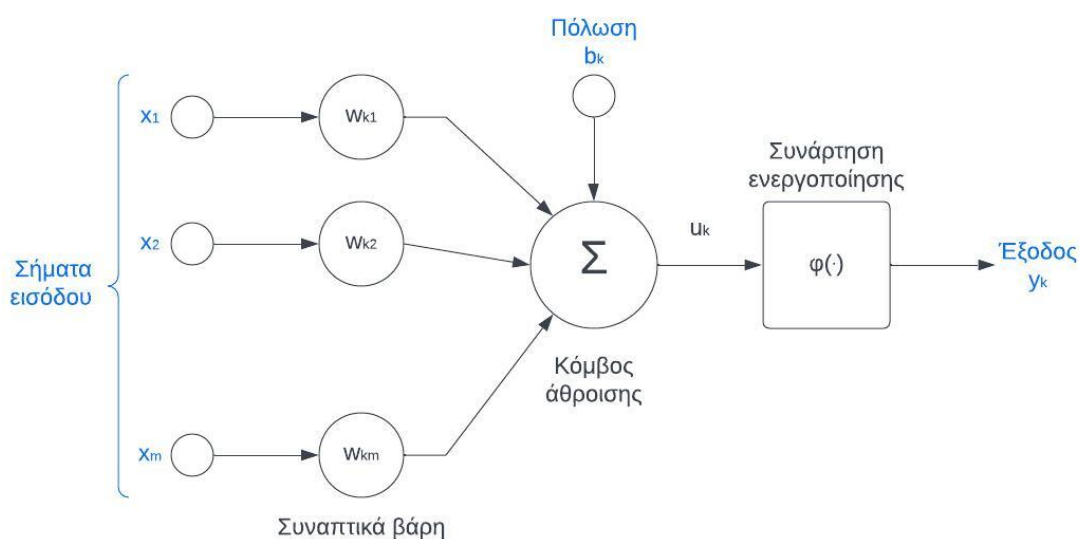
Ένας νευρώνας δομείται από ένα κυτταρικό σώμα, έναν άξονα και τους δενδρίτες. Συνδέεται με τους υπόλοιπους νευρώνες μέσω των συνάψεων που αποτελούν στοιχειώδεις λειτουργικές μονάδες υπεύθυνες για την επιτυχή μεταφορά και μετατροπή ενός ηλεκτροχημικού προσυναπτικού σήματος από τον άξονα ενός νευρώνα σε ένα μετασυναπτικό ηλεκτροχημικό σήμα στον δενδρίτη ενός άλλου. Η μετατροπή απαιτεί μια ισχύ η οποία μπορεί να διαφέρει από σύναψη σε σύναψη και αποτελεί μια μορφή αποθήκευσης της γνώσης. Στη συνέχεια, το ηλεκτροχημικό σήμα περνάει στο κυτταρικό σώμα, όπου αυτό με τη σειρά του συγκεντρώνει και ύστερα το “πυροδοτεί” ως ένα παλμό τάσης σύντομης διάρκειας στον άξονά του όπου θα επαναληφθεί η διαδικασία σύνδεσης με επόμενους νευρώνες. Ένας νευρώνας είναι ικανός είτε να αναστείλει είτε να διεγείρει ένα ηλεκτρικό σήμα. Σύμφωνα με πρόσφατες έρευνες, έχει αποδειχτεί πως ο ανθρώπινος εγκέφαλος περιέχει κατά μέσο όρο 100 δισεκατομμύρια νευρώνες. Κάθε ένας από αυτούς μπορεί να συνδεθεί με έως και 10.000 άλλους νευρώνες, περνώντας σήματα μέσω 1000 τρισεκατομμυρίων συνάψεων [28].

Τα τεχνητά νευρωνικά δίκτυα αποτελούν μια διαδικασία μοντελοποίησης της λειτουργίας των βιολογικών νευρωνικών συστημάτων του εγκεφάλου. Στα πλαίσια της συγκεκριμένης διπλωματικής, ένα τεχνητό νευρωνικό δίκτυο, θα αναφέρεται ως ένα νευρωνικό δίκτυο. Ο Simon Haykin έδωσε στο βιβλίο του [29] τον εξής ορισμό:

“ Ένα νευρωνικό δίκτυο είναι ένας τεράστιος παράλληλος επεξεργαστής με κατανεμημένη αρχιτεκτονική, ο οποίος αποτελείται από απλές μονάδες επεξεργασίας και έχει από τη φύση του τη δυνατότητα να αποθηκεύει εμπειρική γνώση και να την καθιστά διαθέσιμη για χρήση. Μοιάζει με τον ανθρώπινο εγκέφαλο σε δύο σημεία:

1. Το δίκτυο προσλαμβάνει τη γνώση από το περιβάλλον του, μέσω μιας διαδικασίας μάθησης.
2. Η ισχύς των συνδέσεων μεταξύ των νευρώνων, που αποκαλείται συναπτικό βάρος, χρησιμοποιείται για την αποθήκευση γνώσης που αποκτιέται.”

Η μονάδα επεξεργασίας πληροφοριών που περιγράφει τα νευρωνικά δίκτυα, είναι αυτή του τεχνητού μη γραμμικού νευρώνα. Η πρώτη εισήγηση του μοντέλου έγινε το 1943 από τον McCulloch και τον Pitts όπως φαίνεται στο σχήμα 5. Αυτό αποτέλεσε μια ισχυρή βάση έρευνας με αποτέλεσμα κάποια χρόνια αργότερα το 1958, ο Rosenblatt να προτείνει το Perceptron. Το Perceptron αποτελεί το πρώτο νευρωνικό δίκτυο που μπορούσε να περιγραφεί αλγοριθμικά και χρησιμοποιούσε τεχνικές μηχανικής μάθησης με τη βοήθεια εκπαιδευτή προκειμένου να επιλύσει πραγματικά προβλήματα.



Σχήμα 5: Μη γραμμικό μοντέλο τεχνητού νευρώνα



Τα συναπτικά βάρη (Synaptic Weights) αποτελούν τη μονάδα αποθήκευσης της γνώσης του μοντέλου. Η τιμή τους πολλαπλασιάζεται με το κατάλληλο αριθμητικό σήμα εισόδου του συστήματος και έτσι αποκτούν καθοριστικό ρόλο στον υπολογισμό της εξόδου του τεχνητού νευρώνα. Οι αλγόριθμοι μηχανικής μάθησης είναι υπεύθυνοι για την κατάλληλη τροποποίηση και βελτιστοποίηση των τιμών τους προκειμένου να επιτευχθεί η αναμενόμενη έξοδος. Σε αντίθεση με τις συνάψεις του ανθρώπινου εγκεφάλου, μπορούν να πάρουν τόσο αρνητικές όσο και θετικές τιμές.

Ο κόμβος άθροισης λειτουργεί ως γραμμικός συνδυαστής των αριθμητικών σημάτων εισόδου που έχουν σταθμιστεί από τα συναπτικά βάρη του νευρώνα. Συγκεκριμένα για σήματα εισόδου  $[x_1, x_2, \dots, x_m]$  και συναπτικά βάρη νευρώνα  $k$   $[w_{k1}, w_{k2}, \dots, w_{km}]$ , η έξοδος  $u$  του κόμβου άθροισης θα είναι ίση με το αποτέλεσμα της εξίσωσης (1.1):

$$u_k = \sum_{j=1}^m w_{kj}x_j = w_{k1}x_1 + w_{k2}x_2 + \dots + w_{km}x_m \quad (1.1)$$

Το μοντέλο παρουσιάζει επίσης μια εξωτερικά εφαρμοζόμενη πόλωση (Bias)  $b_k$ , η οποία αποτελεί ακόμα μια εκπαιδευόμενη παράμετρο του νευρώνα και καθορίζει τη θετική ή αρνητική μετατόπιση της συνάρτησης ενεργοποίησης. Αυτή προστίθεται γραμμικά στην έξοδο του κόμβου άθροισης.

Η συνάρτηση ενεργοποίησης (Activation Function) αποτελεί ένα μη γραμμικό στοιχείο παραμόρφωσης των τιμών και κανονικοποίησης των πεπερασμένων τιμών της εξόδου σε ένα ορισμένο εύρος πλάτους. Πιο συγκεκριμένα, για συνάρτηση ενεργοποίησης  $\varphi$ , η έξοδος του νευρώνα  $k$  υπολογίζεται ως εξής:

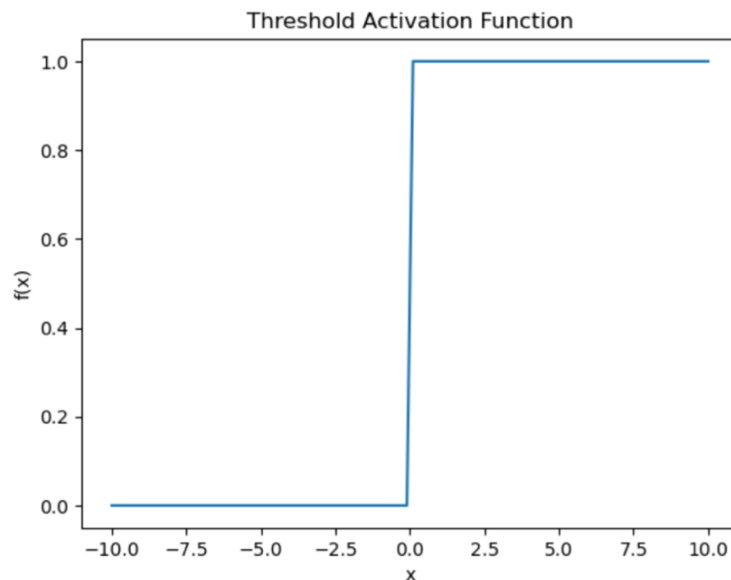
$$y_k = \varphi(u_k + b_k) \quad (1.2)$$

#### 1.4 Συναρτήσεις ενεργοποίησης

Με τη χρήση τους εισάγεται η μη γραμμικότητα στο μοντέλο και εξασφαλίζεται πως το νευρωνικό δίκτυο είναι ικανό να αντιληφθεί σύνθετες σχέσεις μεταξύ των σημάτων εισόδου και των προβλέψεων. Μπορούν να αναφερθούν και ως συναρτήσεις περιορισμού (Squashing Functions), αφού περιορίζουν το επιτρεπτό εύρος πλάτους του σήματος εξόδου σε κάποια πεπερασμένη τιμή. Ανά τα χρόνια, έχει χρησιμοποιηθεί πλήθος διαφορετικών συναρτήσεων ενεργοποίησης κατάλληλες για διαφορετικούς τύπους προβλημάτων. Παρακάτω αναφέρονται κάποιες από τις βασικότερες [30]:

Συνάρτηση Κατωφλιού (Threshold): Η συνάρτηση κατωφλιού αποτελεί μια από τις πρώτες συναρτήσεις ενεργοποίησης που χρησιμοποιήθηκε στη δημιουργία νευρωνικών δικτύων. Η αναπαράστασή της φαίνεται στο σχήμα 6. Είναι μια υπολογιστικά απλή συνάρτηση αλλά ταυτόχρονα περιλαμβάνει ορισμένους περιορισμούς. Είναι ασυνεχής, χωρίς κατάλληλη ικανότητα διαχείρισης συνεχών τιμών και έτσι οδηγεί σε αστάθεια κατά την εκπαίδευση του δικτύου. Επιπλέον, δεν είναι διαφορίσιμη, με αποτέλεσμα να είναι δύσκολο να χρησιμοποιηθεί σε αλγόριθμους που βασίζονται στη κλίση, όπως για παράδειγμα ο αλγόριθμος της οπισθοδιάδοσης (Backpropagation). Περιλαμβάνει τους περιορισμούς της εξίσωσης (1.3):

$$f(x) = \begin{cases} 1 & \text{εάν } x \geq 0 \\ 0 & \text{εάν } x < 0 \end{cases} \quad (1.3)$$

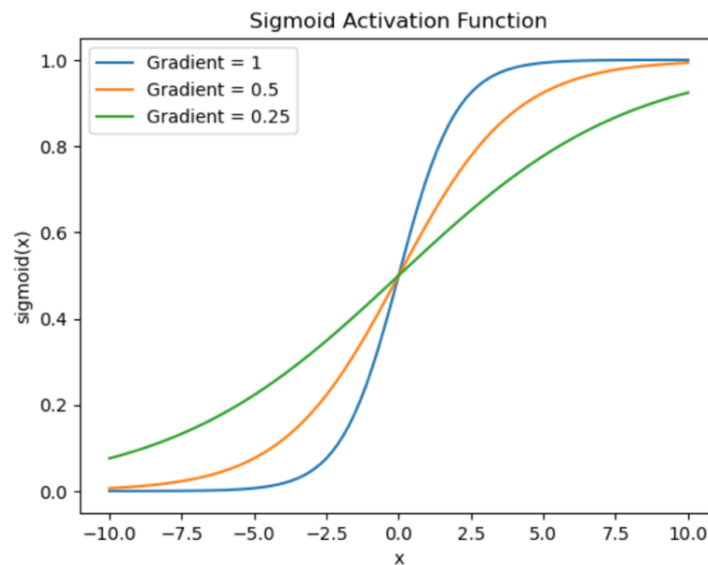


**Σχήμα 6:** Συνάρτηση κατωφλιού

Σιγμοειδής Συνάρτηση (Sigmoid): Η σιγμοειδής συνάρτηση αποτελεί την πλέον κοινή συνάρτηση ενεργοποίησης για την κατασκευή νευρωνικών δικτύων. Η αναπαράστασή της φαίνεται στο σχήμα 7. Ορίζει υπερεπίπεδα στο χώρο των δεδομένων και δημιουργεί τομές απόφασης που αντιστοιχούν σε τομές υπερεπιπέδων. Οι τιμές εισόδου αντιστοιχίζονται σε ένα κλειστό διάστημα τιμών μεταξύ του 0 και του 1. Χρησιμοποιείται συχνά στο επίπεδο εξόδου μοντέλων δυαδικής ταξινόμησης όπου η έξοδος ορίζεται ως η πιθανότητα μιας θετικής κατάστασης. Οι τιμές της υπολογίζονται μέσω της εξίσωσης (1.4):

$$f(x) = \text{sigmoid}(x) = \frac{1}{1 + \exp(-ax)} \quad (1.4)$$

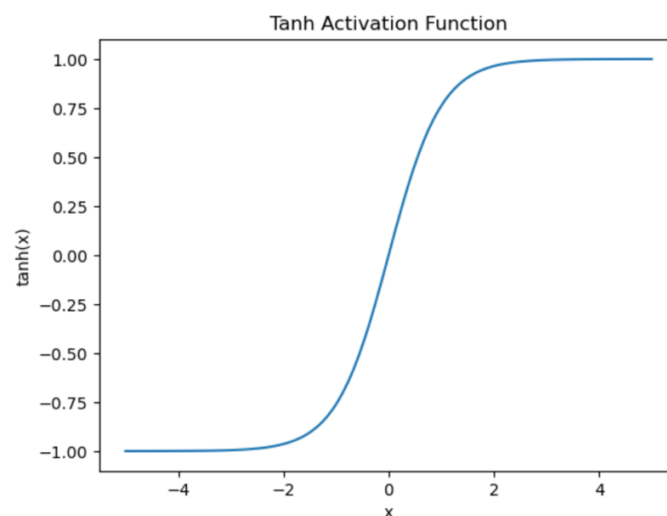
Το  $\alpha$  αποτελεί τη παράμετρο κλίσης της σιγμοειδούς συνάρτησης. Όσο αυτή τείνει προς το άπειρο, τόσο πλησιάζει τη συμπεριφορά της συνάρτησης κατωφλιού ενώ όσο τείνει προς το 0, μετατρέπεται σταδιακά σε μια γραμμική συνάρτηση.



**Σχήμα 7:** Σιγμοειδής συνάρτηση για μεταβαλλόμενη παράμετρο κλίσης  $\alpha$

Συνάρτηση Υπερβολικής εφαπτομένης (Tanh): Η συνάρτηση ενεργοποίησης υπερβολικής εφαπτομένης χρησιμοποιείται κυρίως στα κρυφά επίπεδα του νευρωνικού δικτύου και λειτουργεί παρόμοια με τη σιγμοειδή. Οι τιμές εισόδου αντιστοιχίζονται πλέον σε ένα κλειστό διάστημα τιμών μεταξύ του -1 και του 1. Η αναπαράστασή της φαίνεται στο σχήμα 8 ενώ οι τιμές της υπολογίζονται μέσω της εξίσωσης (1.5):

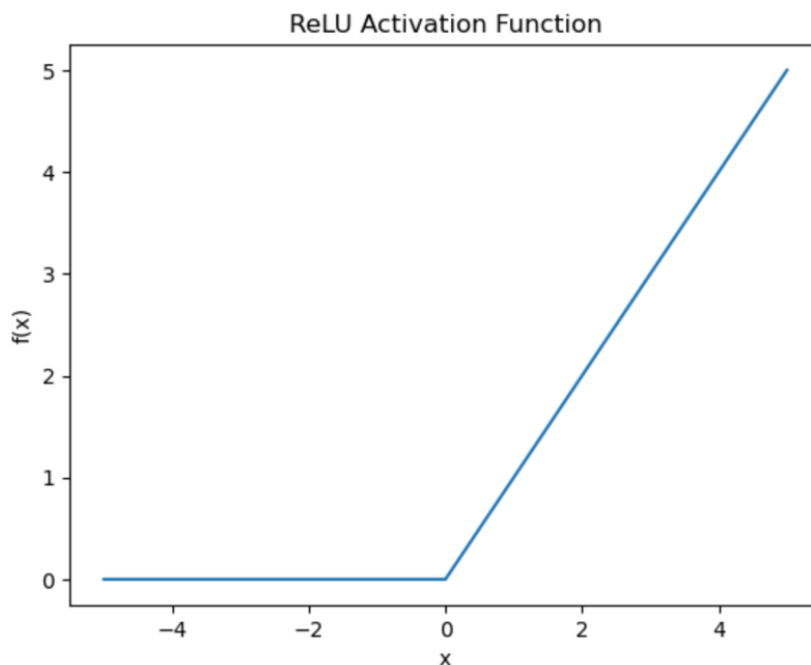
$$f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \quad (1.5)$$



**Σχήμα 8:** Συνάρτηση υπερβολικής εφαπτομένης

Συνάρτηση διορθωμένης γραμμικής μονάδας (ReLU): Η συνάρτηση ενεργοποίησης ReLU (Rectified Linear Unit) αντιστοιχίζει τις αρνητικές τιμές εισόδου στο 0 ενώ τις θετικές τιμές εισόδου στην αρχική τιμή. Με αυτόν τον τρόπο καταφέρνει να μειώσει τη ταχύτητα μάθησης και να κάνει το μοντέλο αποδοτικότερο. Είναι υπολογιστικά απλή και εξαλείφει το πρόβλημα του εξαφανισμού της κλίσης όπου η κλίση έχει υπερβολικά μικρή τιμή για να είναι χρήσιμη κατά τον αλγόριθμο της οπισθοδιάδοσης. Βασικό περιορισμό της συνάρτησης δημιουργούν οι αρνητικές τιμές που μηδενίζονται και έτσι είναι πιθανό ένα μεγάλο κομμάτι του δικτύου να μη συμβάλει στη δημιουργία εξόδου του μοντέλου (Dying ReLU Problem [31]). Η αναπαράστασή της φαίνεται στο σχήμα 9 ενώ οι τιμές της υπολογίζονται μέσω της εξίσωσης (1.6):

$$f(x) = \max(0, x) \quad (1.6)$$



**Σχήμα 9:** Συνάρτηση διορθωμένης γραμμικής μονάδας

Μια υποπερίπτωσή της είναι η Leaky ReLU, όπου η μόνη διαφορά που παρουσιάζει είναι πως οι αρνητικές τιμές εισόδου έχουν πλέον μια μικρή κλίση που παίρνει συνήθως την τιμή 0.01 έτσι ώστε να αντιμετωπίζεται ο περιορισμός που αναφέρθηκε. Είναι πολύ πιο χρήσιμη σε περιπτώσεις νευρωνικών δικτύων πολλών στοιβάδων. Οι τιμές της υπολογίζονται μέσω της εξίσωσης (1.7):

$$f(x) = \max(0.01 \cdot x, x) \quad (1.7)$$

## 1.5 Αρχιτεκτονικές τεχνητών νευρωνικών δικτύων

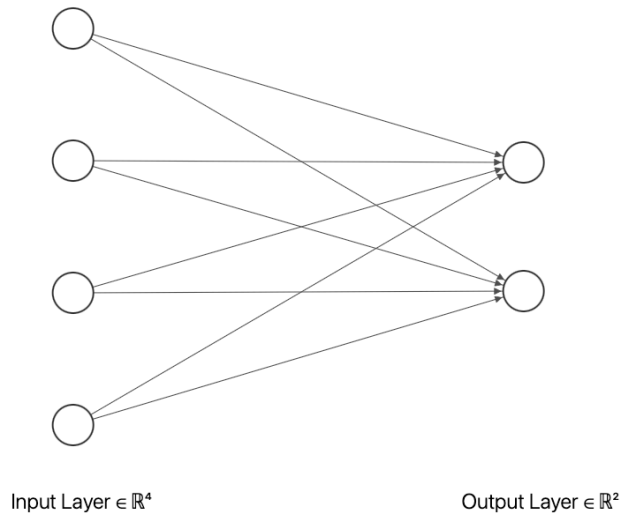
Ένας μεμονωμένος τεχνητός νευρώνας αποτελεί ένα καλά δομημένο μοντέλο, όμως δεν περιλαμβάνει σε καμία περίπτωση τις απαραίτητες ικανότητες για την επίλυση περίπλοκων προβλημάτων υπολογιστικής νοημοσύνης. Όταν όμως συνδυάζονται δύο ή περισσότεροι νευρώνες, δημιουργείται ένα νευρωνικό δίκτυο. Ο τρόπος με τον οποίο συνδέονται, ονομάζεται τοπολογία ή αρχιτεκτονική του νευρωνικού δικτύου. Ανά τα χρόνια, έχει προταθεί και ερευνηθεί ένας μεγάλος αριθμός αρχιτεκτονικών ικανές να επιλύσουν μεμονωμένα προβλήματα ή/και με δυνατότητα γενίκευσης καταλαμβάνοντας ένα μεγαλύτερο εύρος εφαρμογών. Αυτές μπορούν να χωριστούν σε κατηγορίες ανάλογα με ορισμένα χαρακτηριστικά τους και τον τρόπο λειτουργίας τους.

Ο βασικός παράγοντας που καθορίζει την κατηγορία τους, είναι ο τρόπος σύνδεσης μεταξύ των νευρώνων. Τα νευρωνικά δίκτυα οργανώνονται σε μορφή επιπέδων. Αν οι νευρώνες ενός επιπέδου συνδέονται απευθείας προς μια μόνο κατεύθυνση με τους νευρώνες του επόμενου, τότε αυτά τα δίκτυα ονομάζονται πρόσθιας τροφοδότησης (Feedforward). Αντίστοιχα, αν παρουσιάζεται έστω και ένας βρόχος ανάδρασης του σήματος, τότε αυτά τα δίκτυα ονομάζονται αναδρομικά (Recurrent). Τα αναδρομικά νευρωνικά δίκτυα θα μελετηθούν διεξοδικά στο επόμενο κεφάλαιο της διπλωματικής.

Κάθε επίπεδο του δικτύου αποτελείται από κόμβους. Αν κάθε κόμβος ενός επιπέδου συνδέεται με κάθε άλλο κόμβο του επόμενου επιπέδου, τότε το δίκτυο χαρακτηρίζεται ως πλήρως συνδεδεμένο. Αντιθέτως, αν δεν παρουσιάζεται κάποια σύνδεση, το δίκτυο χαρακτηρίζεται ως μερικώς συνδεδεμένο. Σημαντικό ρόλο αποτελεί και το είδος του νευρώνα που επιλέγεται. Για παράδειγμα, μια από τις πιο συνηθισμένες επιλογές νευρώνα στα δίκτυα πρόσθιας τροφοδότησης, είναι αυτός του Perceptron, ο οποίος δομείται με τον τρόπο που αναλύθηκε στην ενότητα 1.3 και χρησιμοποιεί τη συνάρτηση ενεργοποίησης κατωφλιού.

### 1.5.1 Δίκτυα πρόσθιας τροφοδότησης ενός επιπέδου

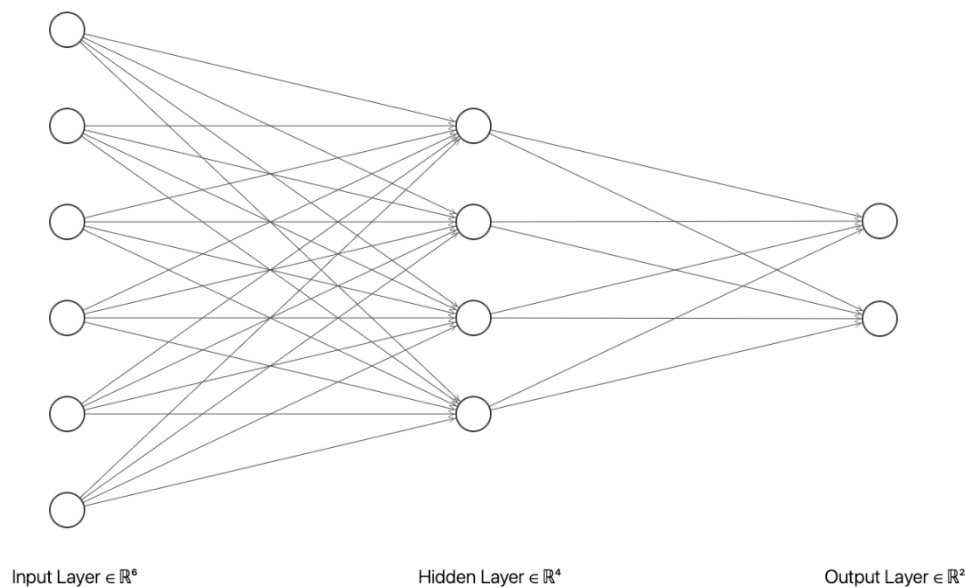
Τα δίκτυα πρόσθιας τροφοδότησης ενός επιπέδου αποτελούν την απλούστερη δυνατή μορφή ενός νευρωνικού δικτύου όπως φαίνεται στο σχήμα 10. Περιλαμβάνει ένα επίπεδο εισόδου, το οποίο αποτελείται από ένα σύνολο κόμβων που περιέχουν τα σήματα εισόδου του δικτύου. Αυτά μπορούν να αποκτηθούν με διάφορους τρόπους, όπως για παράδειγμα ένα σύνολο αισθητήρων. Δεν παρουσιάζει υπολογισμούς, οπότε δεν προσμετρείται στο σύνολο επιπέδων. Αυτό το επίπεδο συνδέεται απευθείας με ένα επίπεδο νευρώνων εξόδου, όπου υπολογίζεται η έξοδος του δικτύου. Η αντίστροφη σύνδεση των επιπέδων, δεν είναι δυνατή.



**Σχήμα 10:** Παράδειγμα δικτύου πρόσθιας τροφοδότησης ενός επιπέδου

### 1.5.2 Δίκτυα πρόσθιας τροφοδότησης πολλών επιπέδων

Τα δίκτυα πρόσθιας τροφοδότησης πολλών επιπέδων περιλαμβάνουν και πάλι το επίπεδο εισόδου και το επίπεδο εξόδου αλλά πλέον προστίθενται και άλλα επίπεδα νευρώνων που εκτελούν υπολογισμούς ανάμεσά τους που ονομάζονται κρυφά επίπεδα όπως φαίνεται στο σχήμα 11. Οι νευρώνες σε κάθε επίπεδο του δικτύου έχουν ως εισόδους τα σήματα εξόδου του προηγούμενου επιπέδου. Προσθέτοντας κρυφά επίπεδα, το δίκτυο έχει την ικανότητα να εξαγάγει στατιστικά υψηλότερης τάξης από την είσοδό του λόγω του επιπλέον συνόλου συνδέσεων και διαστάσεων νευρωνικών αλληλεπιδράσεων [29].



**Σχήμα 11:** Παράδειγμα δικτύου πρόσθιας τροφοδότησης πολλών επιπέδων

Η επιλογή μιας αρχιτεκτονικής αποτελεί μόνο το πρώτο βήμα καθώς είναι απαραίτητη και η κατάλληλη ρύθμιση των παραμέτρων που τη περιγράφουν. Ο αριθμός των επιπέδων  $L$  ονομάζεται βάθος (Depth) του δικτύου ενώ ο αριθμός των νευρώνων  $n_L$  που απαρτίζουν ένα επίπεδο ονομάζεται πλάτος (Width). Για παράδειγμα, η τοπολογία του σχήματος 11, περιγράφεται από βάθος  $L=2$  και πλάτη επιπέδων  $n_0=6$ ,  $n_1=4$  και  $n_2=2$ . Η επιλογή ενός ιδανικού βάρους και πλάτους αποτελεί ένα τετριμμένο εγχείρημα και επί του παρόντος, η μέθοδος επιλογής κατάλληλης διαμόρφωσης για μια συγκεκριμένη εργασία περιλαμβάνει σε μεγάλο επίπεδο τον πειραματισμό. Αν ένα δίκτυο αποτελείται από τουλάχιστον 2 κρυφά επίπεδα, τότε αυτό ονομάζεται βαθύ (Deep Neural Network).

Αυτά είναι σε πολλές περιπτώσεις πιο αποτελεσματικά στο βαθμό ακρίβειας προσέγγισης μιας επιθυμητής συνάρτησης αφού τα παραπάνω επίπεδα επιτρέπουν στο δίκτυο να πραγματοποιήσει εκθετικά περισσότερες συνδέσεις, επιτρέποντας έτσι μια ευρύτερη έξοδο. Κάθε νέο επίπεδο προσθέτει μια συνάρτηση μη γραμμικού επιπέδου στο δίκτυο, διευκολύνοντας έτσι το μοντέλο να πραγματοποιήσει επιθυμητές λειτουργίες που αποτελούν σύνθεση πολλαπλών συναρτήσεων. Απαιτούν όμως μεγάλο υπολογιστικό χρόνο και διαθέσιμους πόρους στη μνήμη του συστήματος.

Σύμφωνα με το καθολικό θεώρημα προσέγγισης (Universal Approximation Theorem) του Kolmogorov [32], οποιαδήποτε διανυσματική, πολλών μεταβλητών, μετρήσιμη (ιδίως, συνεχής) συνάρτηση  $f: \mathbb{R}^{n_0} \rightarrow \mathbb{R}^{n_L}$ , μπορεί να προσεγγιστεί με κάποιο αυθαίρετο μικρό σφάλμα από ένα νευρωνικό δίκτυο πρόσθιας τροφοδότησης με ένα κρυφό στρώμα ( $L=2$ ). Ωστόσο, αυτό δεν μεταφράζεται ιδανικά σε πρακτικό επίπεδο καθώς δεν υπονοεί πως υπάρχει πάντα κάποιος αλγόριθμος εκπαίδευσης που θα φτάσει πραγματικά στην επιθυμητή απόκριση αλλά το ότι υπάρχει κάποιο σύνολο παραμέτρων που την ικανοποιούν. Επιπλέον, δεν λαμβάνεται υπόψιν το αν το δίκτυο είναι βέλτιστο από τη σκοπιά του χρόνου μάθησης και της ευκολίας υλοποίησής του.

Μια από τις πιο γνωστές κατηγορίες δικτύων πρόσθιας τροφοδότησης πολλών επιπέδων είναι αυτή του Perceptron πολλών επιπέδων (Multi-Layer Perceptron ή για συντομία MLP). Η ιδιαιτερότητά τους παρουσιάζεται στο γεγονός του ότι τα επίπεδά του αποτελούνται αποκλειστικά από νευρώνες αρχιτεκτονικής τύπου Perceptron.

## 1.6 Συναρτήσεις απώλειας

Μια συνάρτηση απώλειας (Loss Function) αντιπροσωπεύει μια μαθηματική συνάρτηση, η οποία μετρά τη διαφορά μεταξύ της προβλεπόμενης από ένα μοντέλο μηχανικής μάθησης



εξόδου σε σχέση με την πραγματική έξοδο. Με αυτόν τον τρόπο είναι εφικτό να ποσοτικοποιηθεί ο βαθμός ακρίβειας των προβλέψεων ενός μοντέλου. Οι συναρτήσεις απώλειας είναι εξαιρετικά σημαντικές καθώς αποκτούν βασικό ρόλο στην επιλογή του κατάλληλου μοντέλου. Ένα σύνθετο πρόβλημα όπως για παράδειγμα η πρόβλεψη χρονοσειρών, παρουσιάζει πολυάριθμους παράγοντες που οδηγούν στο συμπέρασμα του ότι δεν είναι δυνατό να οριστεί μια συγκεκριμένη συνάρτηση απώλειας που το περιγράφει. Αυτοί οι παράγοντες μπορεί να είναι πιθανές ακραίες τιμές, ασάφειες στην διανομή των δεδομένων, απαιτήσεις του επιλεγμένου μοντέλου, υπολογιστικές ανάγκες αλλά και απαιτήσεις της απόδοσης. Μια καλή τακτική επιλογής συνάρτησης απώλειας είναι ο προσδιορισμός της από τις ιδιότητες ενός σετ δεδομένων που χρησιμοποιείται κατά την εκπαίδευση, όπως τα όρια και η διανομή [33]. Υπάρχουν πολυάριθμες συναρτήσεις απώλειας ανάλογα με τον τύπο του μοντέλου που χρησιμοποιείται αλλά και το πρόβλημα που επιλύεται. Στην περίπτωση προβλήματος παλινδρόμησης, υπάρχουν δύο εξαιρετικά σημαντικές συναρτήσεις. Αυτές περιλαμβάνονται στις εξισώσεις (1.8) και (1.9). Σε όλες τις παρακάτω περιπτώσεις θα ισχύει ότι  $N$  είναι ο αριθμός των δεδομένων,  $y_i$  η πραγματική έξοδος ενώ  $\hat{y}_i$  η προβλεπόμενη έξοδος.

- Μέσο απόλυτο σφάλμα (Mean Absolute Error – MAE)

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (1.8)$$

Το MAE περιγράφει μια αρκετά απλή συνάρτηση που παρέχει ένα ομοιόμορφο μέτρο απόδοσης του μοντέλου. Επιπλέον, δεν παρουσιάζει μεγάλη ευαισθησία στις ακραίες τιμές. Από την άλλη, η γραμμική προσέγγιση βαθμολόγησης μπορεί να οδηγήσει στο να περαστούν τα ελάχιστα κατά τη διαδικασία της μάθησης και επίσης μπορεί να είναι δύσκολο να υπολογιστούν οι κλίσεις καθώς το 0 δεν είναι διαφοροποιήσιμο.

- Μέσο τετραγωνικό σφάλμα (Mean Squared Error – MSE)

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (1.9)$$

Το MSE μετρά τη μέση τετραγωνική διαφορά μεταξύ των προβλεπόμενων και των πραγματικών τιμών. Αποδίδει μεγαλύτερο βάρος σε ακραίες τιμές και έτσι καθίσταται κατάλληλο σε περιπτώσεις που έχει σημασία το μέγεθος του σφάλματος. Ο τετραγωνισμός των τιμών αυξάνει τον κίνδυνο απότομης αλλαγής των διαβαθμίσεων σε κάποια επανάληψη του αλγορίθμου μάθησης.

Η επιλογή συνάρτησης απώλειας προβλήματος παλινδρόμησης με βάση τον πίνακα 1 είναι ενδεικτική καθώς κύριο παράγοντα θα αποτελεί πάντα το είδος και τα δεδομένα της εκάστοτε εφαρμογής.

**Πίνακας 1:** Κατάλληλες συναρτήσεις απώλειας ανάλογα με τις απαιτήσεις του προβλήματος παλινδρόμησης [33]

Συνάρτηση Απώλειας	Περίπτωση Επιλογής
Mean Absolute Error (MAE)	Όταν χρειάζεται μια μέτρηση σφάλματος που αντιμετωπίζει όλα τα σφάλματα ως ίσα και επιστρέφει μια πιο ερμηνεύσιμη τιμή.
Mean Squared Error (MSE)	Όταν παρουσιάζονται ακραίες περιπτώσεις ακραίων τιμών που πρέπει να ανιχνευτούν.
Mean Bias Error (MBE)	Όταν πρέπει να διατηρηθεί η κατευθυντικότητα των πολώσεων αλλά και να αξιολογηθεί κατά πόσο αυτά εξισορροπούνται.
Relative Absolute Error (RAE)	Όταν χρειάζεται μια απλή μέθοδος απόδοσης μοντέλου προβλέψεων.
Relative Squared Error (RSE)	Όταν χρειάζεται να γίνει σύγκριση μεταξύ μοντέλων διαφορετικής κλίμακας και μονάδων.
Mean Absolute Percentage Error (MAPE)	Όταν απαιτείται κανονικοποίηση λαθών σε μια τυπική κλίμακα λόγω του ποσοστού.
Root Mean Squared Error (RMSE)	Όταν ζητείται μια εύκολα διαφορίσιμη και υπολογιστικά απλή απώλεια. Παρατηρείται συχνά σε τεχνικές βελτιστοποίησης.
Mean Squared Logarithmic Error (MSLE)	Όταν το μοντέλο προβλέπει απευθείας ποσότητες χωρίς κλίμακα και η συμβολή σφάλματος μικρότερων τιμών είναι σημαντικότερη από αυτή των μεγαλύτερων.
Root Mean Squared Logarithmic Error (RMSLE)	Όταν δίνεται έμφαση στην υποεκτίμηση και όχι τόσο στην υπερεκτίμηση.
Normalized Root Mean Squared Error (NRMSE)	Όταν χρειάζεται να εξαλειφθεί η εξάρτηση κλίμακας μεταξύ των μοντέλων που έχουν διαφορετικές εξαρτημένες μεταβλητές και να γίνει σύγκριση μεταξύ τους.
Relative Root Mean Squared Error (RRMSE)	Όταν πρέπει να γίνει σύγκριση διαφορετικών τεχνικών μέτρησης.
Huber Loss	Όταν απαιτείται μεγάλη ανθεκτικότητα σε καθαρά ακραίες τιμές και όχι περιπτώσεις θορύβου. Αποτελεί μια μέση λύση καθώς συμπεριφέρεται σαν MSE όταν η διαφορά εξόδων είναι μικρή και ως MAE όταν είναι μεγάλη.
LogCosh Loss	Ομοίως με την Huber Loss αλλά διπλά διαφορίσιμη παντού.
Quantile Loss	Όταν πρέπει να γίνει πρόβλεψη διαστημάτων και όχι εκτίμηση σημείων.

Τα προβλήματα ταξινόμησης αποτελούν ξεχωριστή κατηγορία μοντέλου με διαφορετικές απαιτήσεις και εκπαίδευση. Σε αντίθεση με τη παλινδρόμηση, ο τύπος του προβλήματος ταξινόμησης περιγράφεται από έναν αριθμό των διαθέσιμων κλάσεων αλλά και από το πόσες από αυτές αντιπροσωπεύουν ένα δεδομένο. Ενδεικτικά περιλαμβάνεται ο πίνακας 2. Εκεί

αναφέρονται κάποιες προτεινόμενες συναρτήσεις απώλειας ανάλογα με τον τύπο του προβλήματος ταξινόμησης.

**Πίνακας 2:** Κατάλληλες συναρτήσεις απώλειας ανάλογα με τον τύπο του προβλήματος ταξινόμησης

Συνάρτηση απώλειας	Τύπος Προβλήματος
Binary Cross Entropy	Binary Classification
Weighted Cross Entropy	
Hinge Loss	
Categorical Cross Entropy	Single Label Multi-Class Classification
Sparse Categorical Cross Entropy	
KullBack Lieber Divergence Loss	
Binary Cross Entropy	Multi-Label Multi-Class Classification

## 1.7 Βελτιστοποιητές

Στη μηχανική μάθηση, οι βελτιστοποιητές (Optimizers) αποτελούν μια σειρά αλγορίθμων που χρησιμοποιούνται για την ελαχιστοποίηση της αντικειμενικής συνάρτησης ενός μοντέλου νευρωνικών δικτύων μέσω της κατάλληλης προσαρμογής των παραμέτρων που το περιγράφουν. Η επιλογή ενός κατάλληλου βελτιστοποιητή θεωρείται ως μια από τις πιο κρίσιμες και δύσκολες επιλογές κατά τον σχεδιασμό καθώς η αυξανόμενη βιβλιογραφία απαριθμεί πλέον εκατοντάδες μεθόδους βελτιστοποίησης. Μια σωστά επιλεγμένη μέθοδος βελτιστοποίησης καθώς και ο σωστός προσδιορισμός των κατάλληλων τιμών των υπερπαραμέτρων που την περιγράφουν, μπορεί να οδηγήσει σε σημαντική μείωση του χρόνου εκπαίδευσης καθώς και αυξημένη απόδοση του εκπαιδευόμενου μοντέλου.

Η βελτιστοποίηση των μοντέλων επιτυγχάνεται μέσω κατάλληλης προσαρμογής του ρυθμού μάθησης (Learning Rate). Ο ρυθμός μάθησης αποτελεί μια υπερπαραμέτρο υπεύθυνη για το μέγεθος του βήματος κάθε επανάληψης του αλγορίθμου βελτιστοποίησης σε κάθε ενημέρωση των παραμέτρων του μοντέλου προς την κατεύθυνση της κλίσης. Η σημασία του θα τονιστεί ιδιαίτερα στα επόμενα κεφάλαια καθώς θα φανεί η συμβολή του στον υπολογισμό των εξισώσεων των αλγορίθμων βελτιστοποίησης.

### 1.7.1 Αλγόριθμος κατάβασης κλίσης

Ο αλγόριθμος κατάβασης κλίσης (Gradient Descent) αποτελεί την πιο απλή περίπτωση ενσωμάτωσης της κλίσης κατά την εκπαίδευση ενός μοντέλου. Υπάρχουν 3 παραλλαγές του αλγορίθμου, οι οποίες διαφέρουν στον όγκο των δεδομένων που χρησιμοποιούνται για τον υπολογισμό της κλίσης της αντικειμενικής συνάρτησης. Αυτές καθορίζουν την ανταλλαγή μεταξύ του χρόνου εκπαίδευσης και της ακρίβειας.

- Κατάβαση κλίσης παρτίδας (Batch Gradient Descent) [34]

Το Batch Gradient Descent αποτελεί μια μέθοδο ελαχιστοποίησης μιας αντικειμενικής συνάρτησης  $J(\theta)$  παραμετροποιούμενη από μεταβλητές  $\theta \in \mathbb{R}^d$  που ενημερώνονται κατάλληλα στην κατεύθυνση αυτής της αντικειμενικής συνάρτησης  $\nabla_{\theta} J(\theta)$ . Σε αυτό υπολογίζεται το πως μεταβάλλεται η απώλεια, ενώ αλλάζει κάθε μεμονωμένη παράμετρος με τις υπόλοιπες να διατηρούνται σταθερές. Ακολουθείται λοιπόν η κατεύθυνση της κλίσης κατηγορικά προκειμένου να οδηγηθεί σε ένα (τοπικό) ελάχιστο. Σημαντικό ρόλο αποτελεί και η τιμή του ρυθμού μάθησης  $\eta$  για το μέγεθος των βημάτων. Μεγάλη τιμή μπορεί να οδηγήσει σε έντονες εναλλαγές και αστάθεια με αποτέλεσμα να μην υπάρξει η ζητούμενη σύγκλιση. Από την άλλη, μικρές τιμές μπορούν να οδηγήσουν σε πολύ μεγάλο χρόνο επίτευξης της σύγκλισης. Η ενημέρωση των παραμέτρων πραγματοποιείται ως εξής:

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta) \quad (1.10)$$

Η συγκεκριμένη μέθοδος απαιτεί τον υπολογισμό των κλίσεων όλου του συνόλου δεδομένων προκειμένου να εκτελεστεί μια ενημέρωση. Είναι αποτελεσματική καθώς παρέχει μια σταθερή σύγκλιση και κλίση σφάλματος. Από την άλλη, είναι αρκετά αργή και απαιτεί να υπάρχει αρκετός χώρος μνήμης όπου θα επεξεργάζεται ολόκληρο το σύνολο δεδομένων. Με αυτή τη μέθοδο δεν είναι εφικτό να υπάρξει online ενημέρωση του μοντέλου. Το batch gradient descent εγγυάται σύγκλιση σε ολικό ελάχιστο για κυρτές επιφάνειες σφάλματος και σύγκλιση σε τοπικό ελάχιστο σε μη κυρτές επιφάνειες σφάλματος.

- Στοχαστική κατάβαση κλίσης (Stochastic Gradient Descent) [35]

Σε αυτή τη περίπτωση ακολουθείται παρόμοια διαδικασία με το batch gradient descent αλλά εκτελείται ενημέρωση των παραμέτρων σε κάθε δεδομένο εκπαίδευσης  $x^{(i)}$  με label  $y^{(i)}$  όπως φαίνεται στην εξίσωση (1.11):

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i)}; y^{(i)}) \quad (1.11)$$

Το Stochastic Gradient Descent είναι πολύ πιο γρήγορο καθώς μειώνονται οι πλεονάζοντες υπολογισμοί και δίνεται η δυνατότητα χρήσης του σε online μάθηση. Εκτελούνται συχνές ενημερώσεις που οδηγούν σε έντονη διακύμανση της συνάρτησης. Αυτό οδηγεί στην ικανότητα μετάβασης σε νέα και πιθανώς δυνητικά καλύτερα τοπικά ελάχιστα αλλά ταυτόχρονα μπορεί να δυσκολεύει τη σύγκλιση στο ολικό ελάχιστο λόγω της συνεχούς

διακύμανσης. Από την άλλη έχει παρατηρηθεί ότι ένας αρκετά μειωμένος ρυθμός μάθησης, μπορεί να οδηγήσει σε παρόμοια συμπεριφορά με τον batch gradient descent [34].

- Κατάβαση κλίσης μικρής παρτίδας (Mini Batch Gradient Descent) [36]

Το Mini Batch Gradient Descent αποτελεί ένα συνδυασμό των μεθόδων που αναφέρθηκαν καθώς εισάγει την έννοια της ενημέρωσης ανά μια παρτίδα δεδομένων εκπαίδευσης  $n$ . Το μέγεθος της παρτίδας αποτελεί μια υπερπαράμετρο που θα πρέπει να ρυθμιστεί κατάλληλα κατά τον συντονισμό της εκπαίδευσης. Από αυτό θα εξαρτηθεί σε μεγάλο βαθμό η απόδοση, η ταχύτητα αλλά και το υπολογιστικό κόστος της εκπαίδευσης. Κοινά μεγέθη παρτίδας μπορεί να είναι μεταξύ 50 και 256 αλλά στην πραγματικότητα θα εξαρτάται πάντα από την εκάστοτε εφαρμογή. Είναι πολύ σημαντικό ο αριθμός παρτίδας να επιλέγεται έχοντας γνώση των διαθέσιμων περιορισμών της μνήμης έτσι ώστε να μπορεί να πραγματοποιηθεί σωστή διαχείριση των πόρων. Η ενημέρωση των παραμέτρων πραγματοποιείται ως εξής:

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i:i+n)}, y^{(i:i+n)}) \quad (1.12)$$

Με την εισαγωγή της παρτίδας επιτυγχάνεται η μείωση της διακύμανσης στις ενημερώσεις των παραμέτρων και έτσι εξασφαλίζεται σταθερότερη σύγκλιση. Επιπλέον, μπορεί εύκολα να προσαρμοστεί σε πιο εξελιγμένους αλγόριθμους βελτιστοποίησης για καλύτερη απόδοση. Ταυτόχρονα όμως προσφέρει κάποιους περιορισμούς και προκλήσεις. Είναι αρκετά δύσκολο να βρεθεί ο βέλτιστος ρυθμός μάθησης και να οριστεί ένα σωστό πρόγραμμα μάθησης (Learning Rate Schedule). Αυτά αποτελούν έννοιες οι οποίες μπορούν να ρυθμιστούν μόνο πριν την έναρξη της εκπαίδευσης και παραμένουν σταθερές καθ' όλη τη διάρκειά της διαδικασίας. Επίσης παρατηρείται δυσκολία ελαχιστοποίησης εξαιρετικά μη κυρτών συναρτήσεων.

### 1.7.2 Αλγόριθμος προσαρμοστικής εκτίμησης της ροπής (Adam)

Ο αλγόριθμος προσαρμοστικής εκτίμησης της ροπής (Adaptive Momentum Estimation) προτάθηκε ως μια μέθοδος στοχαστικής βελτιστοποίησης της απαίτησης των κλίσεων πρώτης τάξης αλλά και της μικρής χρήσης της μνήμης. Ο αλγόριθμος αποτελεί μια μέθοδο υπολογισμού προσαρμοστικού ρυθμού μάθησης για κάθε παράμετρο. Αποθηκεύει έναν εκθετικά αποσυντιθέμενο μέσο όρο παλαιότερων τετραγωνικών κλίσεων  $v_t$  όπως και έναν εκθετικά αποσυντιθέμενο μέσο όρο παλαιότερων κλίσεων  $m_t$ . Έστω ότι:

$$g_t = \nabla_{\theta} J(\theta) \quad (1.13)$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (1.14)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (1.15)$$

Τα  $v_t$  και  $m_t$  αποτελούν προσεγγίσεις της πρώτης ροπής (μέσος όρος) και της δεύτερης ροπής (μη επικεντρωμένη διακύμανση) των κλίσεων. Τα  $\beta_1$  και  $\beta_2$  αποτελούν εκθετικά αποσυντιθέμενους ρυθμούς για τις τρέχουσες εκτιμήσεις και ανήκουν στο διάστημα  $[0,1)$ . Καθώς τα  $v_t$  και  $m_t$  αρχικοποιούνται ως διανύσματα μηδενικών, παρατηρείται bias προς το 0 στα πρώτα χρονικά βήματα και όταν οι ρυθμοί είναι μικροί. Αυτό αντιμετωπίζεται υπολογίζοντας διορθωμένες εκτιμήσεις ως εξής:

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (1.16)$$

$$\widehat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (1.17)$$

Οι παράμετροι ενημερώνονται ως εξής:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\widehat{v}_t + \varepsilon}} \widehat{m}_t \quad (1.18)$$

Οι προτεινόμενες τιμές των δημιουργών είναι  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\varepsilon = 10^{-8}$  και  $\alpha = 0.001$  [37]

Ο αλγόριθμος Adam προσαρμόζει το ρυθμό μάθησης των παραμέτρων και έτσι τη συμμετοχή των κλίσεων στην ενημέρωση των συναπτικών βαρών. Έχει ικανότητα σωστής λειτουργίας σε αραιές κλίσεις και εκτελεί μια μορφή ανόπτησης μεγέθους βήματος. Το μέγεθος των ενημερώσεων είναι αμετάβλητο από την επανακλιμάκωση της διαβάθμισης και δεν απαιτεί σταθερό στόχο.

## 1.8 Οπισθοδιάδοση (Backpropagation)

Η οπισθοδιάδοση αποτελεί έναν από τους πιο γνωστούς αλγορίθμους εκπαίδευσης νευρωνικών δικτύων. Αποτελεί μέθοδο ενημέρωσης των συναπτικών βαρών μεταξύ νευρώνων, έχοντας ως βάση το σφάλμα που υπολογίζεται με κάποια κατάλληλη συνάρτηση απώλειας.

Το πρώτο στάδιο περιλαμβάνει την αρχικοποίηση των συνάψεων και των πολώσεων των νευρώνων σε τυχαίες αρχικές τιμές. Αυτή η διαδικασία μπορεί να δημιουργήσει πιθανότητα παγίδευσης του αλγορίθμου σε κάποιο τοπικό ελάχιστο αλλά και να προκαλέσει μεγάλη διακύμανση των τελικών αποτελεσμάτων. Υπάρχουν διάφοροι τρόποι αντιμετώπισης που εισάγουν την έννοια της ορμής. Είναι απαραίτητο να δοθεί ιδιαίτερη προσοχή κατά την επιλογή της τιμής της καθώς μπορεί να οδηγήσει σε αγνόηση της τοπικής κλίσης.

Ο αλγόριθμος περιλαμβάνει 2 περάσματα υπολογισμών. Ένα προς τα εμπρός (Forward pass) και ένα από εμπρός προς τα πίσω (Backwards pass). Στο forward pass, δεν παρουσιάζεται μεταβολή των συναπτικών βαρών του δικτύου και υπολογίζονται τα λειτουργικά σήματα νευρώνα προς νευρώνα. Υποθέτοντας έναν νευρώνα  $j$  και έναν νευρώνα  $i$ , παρουσιάζεται ο παρακάτω υπολογισμός εξόδου:

$$y_i(n) = \varphi(u_j(n)) \quad (1.19)$$

Με τοπικό πεδίο νευρώνα  $j$  ίσο με:

$$u_j(n) = \sum_{i=0}^m w_{ji}(n)y_i(n) \quad (1.20)$$

Όπου  $m$  ο συνολικός αριθμός εισόδων εξαιρούμενης της πόλωσης σε νευρώνα  $j$ ,  $w_{ij}(n)$  το συναπτικό βάρος σύνδεσης των νευρώνων και  $y_i(n)$  το σήμα εισόδου νευρώνα  $j$ . Αυτή η φάση υπολογισμού ξεκινά από το πρώτο κρυφό επίπεδο μέσω εισαγωγής του διανύσματος εισόδου και τερματίζει στο επίπεδο εξόδου μέσω κατάλληλου υπολογισμού του σήματος σφάλματος κάθε νευρώνα του επιπέδου.

Στο backwards pass το σήμα σφάλματος διαδίδεται προς τα αριστερά επίπεδο προς επίπεδο και υπολογίζεται αναδρομικά η τοπική κλίση  $\delta$  για κάθε νευρώνα ξεχωριστά. Μέσω αυτής μπορεί να χρησιμοποιηθεί κατάλληλα ο κανόνας Δέλτα ή κανόνας Widrow – Hoff [38] σύμφωνα με την εξίσωση (1.21):

$$\Delta w_{ji}(n) = \eta \cdot \delta_j(n) \cdot y_i(n) \quad (1.21)$$

(Διόρθωση βάρους) = (Ρυθμός μάθησης) · (Τοπική κλίση) · (Σήμα εισόδου νευρώνα  $j$ )

Υπάρχουν 2 περιπτώσεις υπολογισμού της τοπικής κλίσης ανάλογα με το επίπεδο που βρίσκεται ο νευρώνας. Αν ο νευρώνας βρίσκεται στο επίπεδο εξόδου, τότε η τοπική κλίση θα

είναι ίση με το γινόμενο της παραγώγου της σχετιζόμενης συνάρτησης ενεργοποίησης  $\varphi'_j(u_j(n))$  και του σήματος σφάλματος  $e_j(n)$  όπως φαίνεται παρακάτω:

$$\delta_j(n) = \varphi'_j(u_j(n)) \cdot e_j(n) \quad (1.22)$$

Αν ο νευρώνας βρίσκεται σε κρυφό επίπεδο, τότε η τοπική κλίση θα είναι ίση με το γινόμενο της παραγώγου της σχετιζόμενης συνάρτησης ενεργοποίησης  $\varphi'_j(u_j(n))$  και του σταθμισμένου αθροίσματος των τοπικών αποκλίσεων που υπολογίζονται στους νευρώνες του επόμενου από αυτό επίπεδο και συνδέονται με τον ζητούμενο νευρώνα σύμφωνα με την εξίσωση (1.23).

$$\delta_j(n) = \varphi'_j(u_j(n)) \sum_k \delta_k(n) \cdot w_{kj}(n) \quad (1.23)$$

Η διαδικασία αυτών των 2 περασμάτων για όλα τα δεδομένα εκπαίδευσης περιγράφεται ως ένα Epoch σε αυτόν τον αλγόριθμο. Στη συνέχεια, ελέγχεται η παρουσία συνθήκης τερματισμού. Αν δεν ικανοποιείται, τότε η διαδικασία των περασμάτων επαναλαμβάνεται. Ως συνθήκη τερματισμού ορίζεται ένας μέγιστος αριθμός epoch, η ικανοποίηση ενός κριτηρίου σφάλματος ή η επιτυχής σύγκλιση των συνάψεων.



---

## ΚΕΦΑΛΑΙΟ 2<sup>ο</sup> : ΑΝΑΔΡΟΜΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

---

Τα αναδρομικά νευρωνικά δίκτυα (Recurrent Neural Networks ή για συντομία RNNs) αποτελούν μια μεγάλη κατηγορία νευρωνικών δικτύων που ξεχωρίζουν λόγω της αρχιτεκτονικής τους. Για να θεωρηθεί ένα νευρωνικό δίκτυο αναδρομικό, θα πρέπει να παρουσιάζει τουλάχιστον ένα βρόχο ανάδρασης. Η ανάδραση (Feedback) θεωρείται πως υπάρχει σε ένα δυναμικό σύστημα, όταν οποιαδήποτε έξοδος ενός στοιχείου του συστήματος επηρεάζει εν μέρει την είσοδο που εφαρμόζεται σε αυτό το συγκεκριμένο στοιχείο, προκαλώντας έτσι την εμφάνιση μιας ή περισσότερων κλειστών «διαδρομών» για τη μετάδοση σημάτων σε όλη την έκταση του συστήματος [29]. Η ανάδραση δεν αποτελεί κάποιο καινούριο φαινόμενο αφού το ανθρώπινο νευρικό σύστημα παρουσιάζει έντονο αναδρομικό χαρακτήρα στη λειτουργία του. Είναι αναμενόμενο να έχουν υπάρξει προσπάθειες μοντελοποίησης αυτής της συμπεριφοράς.

Αυτό που τα διαφοροποιεί, είναι ο τρόπος διαχείρισης των δεδομένων στους κόμβους που τα αποτελούν λόγω της αρχιτεκτονικής τους. Ένα νευρωνικό δίκτυο πρόσθιας τροφοδότησης είναι καθαρά στατικό και δεν μπορεί να επεξεργαστεί χρονικές πληροφορίες. Για να εφαρμοστεί κάποιο χρονικό παράθυρο θα πρέπει κάθε στοιχείο μιας ακολουθίας να μετατραπεί σε διαφορετική παράμετρο εισόδου (Feature). Αυτό δεν είναι ιδανικό καθώς το δίκτυο θα πρέπει να μαθαίνει από την αρχή, σε κάθε βήμα ξεχωριστά, το ότι αναλύει μια ακολουθία, οδηγώντας έτσι σε μεγαλύτερο χρόνο εκπαίδευσης και ανακρίβειες. Αντιθέτως, ένα αναδρομικό νευρωνικό δίκτυο περιέχει κατάλληλες συνδέσεις στη κρυφή στριβάδα του έτσι ώστε οι πληροφορίες όλων των χρονικών βημάτων να είναι συνδεδεμένες μεταξύ τους. Έτσι, κάθε βήμα περιέχει πληροφορίες από όλα τα προηγούμενα βήματα μέσω των τιμών που προσλαμβάνει και με αυτόν τον τρόπο η συσσωρευμένη πληροφορία αποκτάει το ρόλο 'μνήμης' που συμμετέχει στην απόφαση των εξόδων. Με αυτόν τον τρόπο είναι ικανό να αποκτήσει αντίληψη του χρόνου και να χρησιμοποιηθεί αποτελεσματικά σε προβλήματα πρόβλεψης εξόδου χρονικού βήματος  $t+1$  έχοντας δεδομένα βήματος  $t$  ή προηγούμενων βημάτων.

Τα αναδρομικά νευρωνικά δίκτυα αποτελούν μια κατηγορία δυναμικών μοντέλων τα οποία είναι ιδανικά να διαχειρίζονται εφαρμογές ακολουθιακών δεδομένων εισόδου και εξόδου. Αυτό τα καθιστά κατάλληλα σε περιπτώσεις πρόβλεψης μελλοντικών τιμών χρονοσειρών αλλά και προβλήματα που έχουν να κάνουν με χαρακτήρες κειμένων.

## 2.1 Απλά αναδρομικά νευρωνικά δίκτυα (Simple RNNs)

Ένα απλό αναδρομικό νευρωνικό δίκτυο [39] αποτελείται από ένα επίπεδο εισόδου, ένα αναδρομικό κρυφό επίπεδο και ένα επίπεδο εξόδου. Αυτά μοιράζονται τα ίδια σύνολα παραμέτρων σε όλη την διαδικασία λειτουργίας τους. Με αυτόν τον τρόπο επιτυγχάνεται το να μην υπάρχουν περιορισμοί στο μήκος των ακολουθιών και μεταβλητών που επεξεργάζονται.

Το επίπεδο εισόδου αποτελείται από  $N$  μονάδες εισόδου. Οι εισοδοί αποτελούνται από μια ακολουθία διανυσμάτων κατά τη διάρκεια χρόνου  $t$  όπου  $\{\dots, x_{t-1}, x_t, x_{t+1}, \dots\}$  με

$x_t = (x_1, x_2, \dots, x_N)$ . Οι μονάδες εισόδου συνδέονται με τις κρυφές μονάδες του κρυφού επιπέδου μέσω πίνακα βάρους  $W_{IH}$  διαστάσεων  $(N \times M)$ .

Το αναδρομικό κρυφό επίπεδο αποτελείται από  $M$  κρυφές μονάδες  $h_t = (h_1, h_2, \dots, h_M)$ . Το κρυφό επίπεδο έχει το ρόλο του χώρου κατάστασης ή της «Μνήμης» του συστήματος ως εξής:

$$h_t = \mathcal{F}_H(p_t) \quad (2.1)$$

$$p_t = W_{IH} \cdot x_t + W_{HH} \cdot h_{t-1} + b_h \quad (2.2)$$

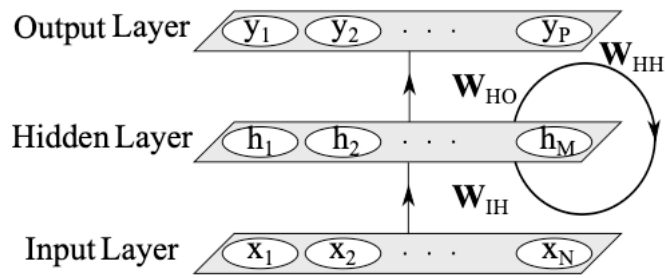
Ως  $\mathcal{F}_H(\cdot)$  ορίζεται κατάλληλη συνάρτηση ενεργοποίησης του κρυφού επιπέδου. Το  $b_h$  αποτελεί το διάνυσμα πόλωσης των κρυφών μονάδων ενώ το  $W_{HH}$  αποτελεί το κατάλληλο πίνακα βάρους μεταξύ των κρυφών καταστάσεων και διαστάσεων  $(M \times M)$ . Οι κρυφές μονάδες συνδέονται με τις μονάδες εξόδου μέσω κατάλληλου πίνακα βάρους  $W_{HO}$  και διαστάσεων  $(M \times P)$ . Το κρυφό επίπεδο αυτό συνοψίζει όλες τις μοναδικές και απαραίτητες καταστάσεις του δικτύου σε πολλά χρονικά βήματα.

Το επίπεδο εξόδου αποτελείται από  $P$  μονάδες εξόδου  $y_t = (y_1, y_2, \dots, y_P)$ . Αυτές υπολογίζονται ως εξής:

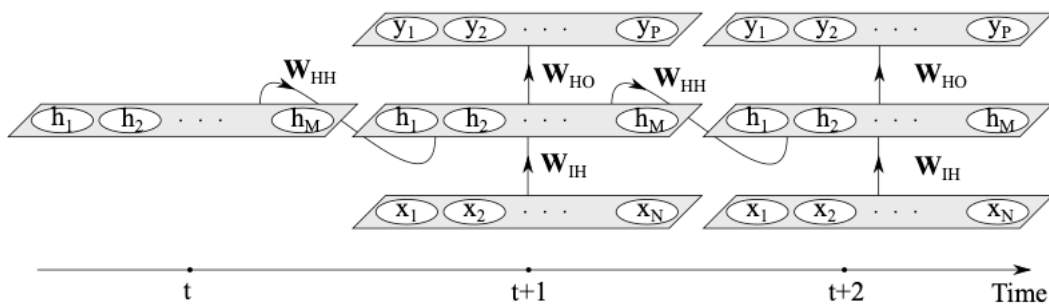
$$y_t = \mathcal{F}_O(o_t) \quad (2.3)$$

$$o_t = W_{HO} \cdot h_t + b_o \quad (2.4)$$

Όπου  $\mathcal{F}_O(\cdot)$  η συνάρτηση ενεργοποίησης του επιπέδου εξόδου και  $b_o$  το διάνυσμα πόλωσης της εξόδου. Από τη στιγμή που υπάρχει διαδοχικότητα των δεδομένων στο χρόνο, τα παραπάνω βήματα θα επαναλαμβάνονται κατά πάροδο χρόνου  $t = (1, \dots, T)$  για  $T$  χρονικές στιγμές. Η αναπαράσταση της αρχιτεκτονικής μπορεί να φανεί στο σχήμα 12 και στο σχήμα 13.



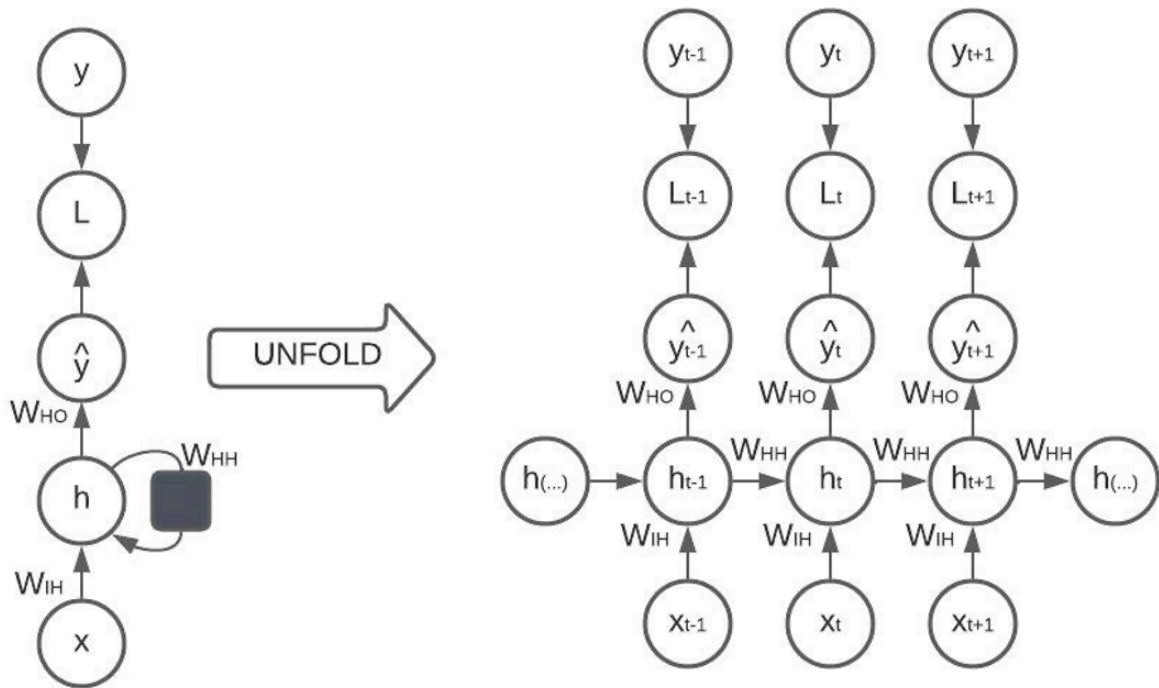
Σχήμα 12: Αρχιτεκτονική Simple RNN σε συμπιεσμένη (Folded) μορφή



Σχήμα 13: Αρχιτεκτονική Simple RNN σε αναλυτική (Unfolded) μορφή

## 2.2 Οπισθοδιάδοση στο χρόνο

Ο αλγόριθμος οπισθοδιάδοσης στο χρόνο (Backpropagation Through Time ή για συντομία BPTT) [40][41], αποτελεί εξέλιξη του απλού αλγορίθμου οπισθοδιάδοσης έτσι ώστε να μπορεί να εφαρμόζεται στα αναδρομικά νευρωνικά δίκτυα. Προκειμένου να οριστεί, πρέπει να επιλεγεί η Unfolded μορφή της αρχιτεκτονικής, όπως φαίνεται στο σχήμα 13, έτσι ώστε να μετατραπεί η χρονική συμπεριφορά του δικτύου σε ένα πολυεπίπεδο δίκτυο πρόσθιας τροφοδότησης. Κύρια διαφορά αποτελεί το ότι οι επιθυμητές αποκρίσεις καθορίζονται για νευρώνες σε πολλά επίπεδα του δικτύου του οποίου η τοπολογία αυξάνεται κατά ένα επίπεδο σε κάθε χρονικό βήμα. Υπάρχει δυνατότητα υλοποίησης του αλγορίθμου τόσο για εκπαίδευση κατά εποχές όσο και για συνεχή εκπαίδευση σε πραγματικό χρόνο.



**Σχήμα 14:** Αναδίπλωση αρχιτεκτονικής RNN για κατάλληλη χρήση σε αλγόριθμο BPTT

Ο πίνακας 3 παρέχεται έτσι ώστε να υπάρξει καλύτερη κατανόηση των εξισώσεων που θα αναλυθούν κατά την εξήγηση του αλγορίθμου μάθησης.

**Πίνακας 3:** Μεταβλητές που θα χρησιμοποιηθούν στις εξισώσεις της ενότητας

Μεταβλητή	Περιγραφή
$x$	Επίπεδο εισόδου
$p$	Κρυφό επίπεδο – Απόκριση πριν τη συνάρτηση ενεργοποίησης του επιπέδου
$h$	Κρυφό επίπεδο – Απόκριση μετά τη συνάρτηση ενεργοποίησης του επιπέδου
$o$	Επίπεδο Εξόδου – Προβλέψεις Μοντέλου πριν τη συνάρτηση ενεργοποίησης του επιπέδου
$\hat{y}$	Επίπεδο Εξόδου – Προβλέψεις Μοντέλου μετά τη συνάρτηση ενεργοποίησης του επιπέδου
$\mathcal{L}$	Συνάρτηση Απώλειας
$y$	Πραγματικές τιμές
$W_{IH}$	Πίνακας Βάρους Εισόδου $\rightarrow$ Κρυφού Επιπέδου
$W_{HH}$	Πίνακας Βάρους Κρυφού Επιπέδου $\rightarrow$ Κρυφού Επιπέδου
$W_{HO}$	Πίνακας Βάρους Κρυφού Επιπέδου $\rightarrow$ Εξόδου
$b_h$	Διάνυσμα πόλωσης κρυφού επιπέδου
$b_o$	Διάνυσμα πόλωσης εξόδου
$\mathcal{F}_H$	Συνάρτηση Ενεργοποίησης κρυφού επιπέδου
$\mathcal{F}_O$	Συνάρτηση Ενεργοποίησης επιπέδου εξόδου

Το πρώτο στάδιο του BPTT περιλαμβάνει τη κατάλληλη αρχικοποίηση των συνάψεων και πολώσεων του μοντέλου. Αυτό εξαρτάται πάντα από τον τύπο του προβλήματος και των δεδομένων. Αυτό είναι εξαιρετικά κρίσιμο καθώς με αυτόν τον τρόπο μπορούν να βελτιωθούν προβλήματα που έχουν να κάνουν με την κλίση.

Παρουσιάζονται 2 περάσματα υπολογισμών. Ένα forward pass και ένα backwards pass. Αντίστοιχα με τον απλό αλγόριθμο της οπισθοδιάδοσης, στο forward pass δεν παρατηρείται μεταβολή των τιμών αλλά γίνεται ο υπολογισμός των λειτουργικών σημάτων του δικτύου νευρώνα προς νευρώνα κατά τη πάροδο του χρόνου  $t$  για χρονικά βήματα  $T$ . Για ένα απλό αναδρομικό νευρωνικό δίκτυο απαιτείται χρήση των εξισώσεων (2.1), (2.2), (2.3) και (2.4).

Στη συνέχεια, οι προβλέψεις του δικτύου εισάγονται σε κατάλληλα επιλεγμένη συνάρτηση απώλειας προκειμένου να πραγματοποιηθεί η σύγκρισή τους με τις πραγματικές τιμές και να αξιολογηθεί η απόδοση του μοντέλου. Από τη στιγμή που παρουσιάζεται πάροδος του χρόνου  $t$ , θα πρέπει να υπολογιστεί ως σύνολο που περιλαμβάνει όλα τα απαραίτητα χρονικά βήματα  $T$ .

$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^T \ell_t(\hat{y}_t, y_t) \quad (2.5)$$

Ο BPTT αποτελεί αλγόριθμο 'διάδοσης' κλίσεων και έτσι προκειμένου να υπάρξει backwards pass, πρέπει να πραγματοποιηθεί υπολογισμός των παραγώγων μέσω της συνάρτησης απώλειας. Αυτό απαιτεί τον υπολογισμό μερικών παραγώγων με χρήση του κανόνα αλυσίδας. Οι απαραίτητες κλίσεις αντιστοιχούν στους πίνακες βάρους και τα διανύσματα πόλωσης. Αυτά παραμένουν κοινά για κάθε χρονικό βήμα αλλά ταυτόχρονα υπολογίζονται ως σύνολα λόγω του αναδρομικού χαρακτήρα του μοντέλου.

Η μερική παράγωγος ως προς το πίνακα βάρους μεταξύ κρυφού επιπέδου και επιπέδου εξόδου υπολογίζεται ως εξής:

$$\frac{\partial \mathcal{L}}{\partial W_{HO}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial W_{HO}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial o_t} \cdot \frac{\partial o_t}{\partial W_{HO}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \hat{y}_t} \cdot \mathcal{F}_o'(o_t) \cdot h_t \quad (2.6)$$

Η μερική παράγωγος ως προς το πίνακα βάρους μεταξύ κρυφών επιπέδων υπολογίζεται ως εξής:

$$\frac{\partial \mathcal{L}}{\partial W_{HH}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial W_{HH}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial o_t} \cdot \frac{\partial o_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial W_{HH}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial o_t} \cdot W_{HO} \cdot \frac{\partial h_t}{\partial W_{HH}} \quad (2.7)$$

Η παράγωγος  $\frac{\partial h_t}{\partial W_{hh}}$  όμως εξαρτάται από τα προηγούμενα χρονικά βήματα και υπολογίζεται αναδρομικά. Για παράδειγμα, αν παρουσιάζεται ένα βήμα πίσω υπολογίζεται ως εξής:

$$\frac{\partial h_t}{\partial W_{HH}} = \frac{\partial h_t}{\partial W_{HH}} + \frac{\partial h_t}{\partial h_{t-1}} \cdot \frac{\partial h_{t-1}}{\partial W_{HH}} \quad (2.8)$$

Αντίστοιχα η παράγωγος  $\frac{\partial h_{t-1}}{\partial W_{hh}}$  αναλύεται με τον ίδιο τρόπο για το προηγούμενο βήμα της εφόσον αυτό υπάρχει. Μπορεί λοιπόν να πραγματοποιηθεί μια γενίκευση αυτής της διαδικασίας και να οριστεί για  $r$  διαθέσιμα προηγούμενα βήματα του μοντέλου:

$$\frac{\partial h_t}{\partial W_{HH}} = \prod_{r=1}^t \frac{\partial h_t}{\partial h_r} \cdot \frac{\partial h_r}{\partial W_{HH}} \quad (2.9)$$

Η παράγωγος  $\frac{\partial h_t}{\partial h_r}$  όμως αντιπροσωπεύει με τη σειρά της ένα σύνολο μιας ακολουθίας χρονικών βημάτων. Επιπλέον, αυτά τα χρονικά βήματα δέχονται επιρροή από τη συνάρτηση ενεργοποίησης των κρυφών επιπέδων και έτσι πρέπει να αναλυθούν και ως προς τις τιμές πριν εισαχθούν σε αυτήν.

$$\frac{\partial h_t}{\partial h_r} = \prod_{i=r+1}^t \frac{\partial h_i}{\partial h_{i-1}} = \prod_{i=r+1}^t \frac{\partial h_i}{\partial p_i} \cdot \frac{\partial p_i}{\partial h_{i-1}} = \prod_{i=r+1}^t \text{diag}(\mathcal{F}_H'(p_i)) \cdot W_{HH}^T \quad (2.10)$$

Έτσι η μερική παράγωγος της εξίσωσης (2.7) μπορεί να γραφτεί με τη βοήθεια της εξίσωσης (2.10) ως εξής:

$$\frac{\partial \mathcal{L}}{\partial W_{HH}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial W_{HH}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial o_t} \cdot W_{HO} \cdot \prod_{r=1}^t \prod_{i=r+1}^t [\text{diag}(\mathcal{F}_H'(p_i)) \cdot W_{HH}^T] \cdot \frac{\partial h_r}{\partial W_{HH}} \quad (2.11)$$

Με τον ίδιο τρόπο μπορεί να υπολογιστεί η μερική παράγωγος μεταξύ εισόδου και κρυφού επιπέδου όπως φαίνεται στην εξίσωση (2.12).

$$\frac{\partial \mathcal{L}}{\partial W_{IH}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial W_{IH}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial o_t} \cdot \frac{\partial o_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial W_{IH}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial o_t} \cdot W_{HO} \cdot \frac{\partial h_t}{\partial W_{IH}} \Rightarrow \quad (2.12)$$

$$\frac{\partial \mathcal{L}}{\partial W_{IH}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial W_{IH}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial o_t} \cdot W_{HO} \cdot \prod_{r=1}^t \prod_{i=r+1}^t [\text{diag}(\mathcal{F}_H'(p_i)) \cdot W_{HH}^T] \cdot \frac{\partial h_r}{\partial W_{IH}}$$

Οι εξισώσεις (2.11) και (2.12) μπορούν να αναλυθούν περαιτέρω ως εξής:

$$\frac{\partial \mathcal{L}}{\partial W_{HH}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial W_{HH}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \hat{y}_t} \cdot \mathcal{F}_O'(o_t) \cdot W_{HO} \cdot \prod_{r=1}^t \prod_{i=r+1}^t [\text{diag}(\mathcal{F}_H'(p_i)) \cdot W_{HH}^T] \cdot h_{r-1} \quad (2.13)$$

$$\frac{\partial \mathcal{L}}{\partial W_{IH}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial W_{IH}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \hat{y}_t} \cdot \mathcal{F}_O'(o_t) \cdot W_{HO} \cdot \prod_{r=1}^t \prod_{i=r+1}^t [\text{diag}(\mathcal{F}_H'(p_i)) \cdot W_{HH}^T] \cdot x_r \quad (2.14)$$

Αντίστοιχα η μερική παράγωγος ως προς το διάνυσμα πόλωσης κρυφού επιπέδου και επιπέδου εξόδου υπολογίζεται ως εξής:

$$\frac{\partial \mathcal{L}}{\partial b_y} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial b_y} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial b_y} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \hat{y}_t} \quad (2.15)$$

$$\frac{\partial \mathcal{L}}{\partial b_h} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial b_h} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial b_h} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial h_t} \quad (2.16)$$

Από τη στιγμή που έχουν υπολογιστεί όλες οι απαραίτητες κλίσεις των πινάκων και των πολώσεων, μπορεί να εφαρμοστεί ο κανόνας δέλτα ή κανόνας Widrow – Hoff προκειμένου να γίνει ο υπολογισμός των νέων τιμών. Υποθέτοντας ότι  $\theta = \{ W_{ho}, W_{hh}, W_{xh}, b_y, b_h \}$  είναι οι τιμές των παραμέτρων που θα μεταβληθούν,  $\eta$  ένας ρυθμός μάθησης και  $\theta'$  τις νέες τιμές αυτών των παραμέτρων. Ο υπολογισμός πραγματοποιείται ως εξής:

$$\theta' = \theta - \eta \cdot \frac{\partial \mathcal{L}}{\partial \theta} \quad (2.17)$$

### 2.2.1 Οπισθοδιάδοση στο χρόνο με περικοπή

Ο αλγόριθμος οπισθοδιάδοσης στο χρόνο με περικοπή (Truncated Backpropagation Through Time ή για συντομία TBPTT) [42], αποτελεί μια μετατροπή του απλού BPTT, όπως φαίνεται στο σχήμα 15. Ο TBPTT έχει ως στόχο τη μείωση του υπολογιστικού κόστους ενημέρωσης των κλίσεων. Αυτό ευθύνεται στο μεγάλο αριθμό επαναλήψεων σε περιπτώσεις μοντέλων όπου αναλύονται για εκτενή χρονικά διαστήματα. Έτσι εφαρμόζεται κατάλληλη περικοπή του συνόλου και καθορίζεται ένα ανώτατο όριο χρονικών βημάτων. Με αυτόν τον τρόπο το μοντέλο χωρίζεται σε υποακολουθίες σταθερού μήκους. Θεωρείται μέθοδος εκτίμησης και όχι υπολογισμού ενώ είναι πιθανό να δημιουργηθούν βραχυπρόθεσμες εξαρτήσεις. Μεγάλη σημασία πρέπει να δοθεί στο μήκος της περικοπής καθώς ένας μικρός αριθμός μπορεί να μην οδηγήσει σε σύγκλιση ενώ ένας μεγάλος αριθμός σε εμφανή καθυστέρηση σύγκλισης.



**Σχήμα 15:** Γραφική αναπαράσταση λειτουργίας αλγορίθμων BPTT και TBPTT. Τα μπλε βέλη αντιπροσωπεύουν διαδόσεις προς τα εμπρός ενώ τα κόκκινα βέλη το αντίστροφο. Οι τελείες αντιπροσωπεύουν είτε την επαναφορά εσωτερικών καταστάσεων είτε την επαναφορά κλίσης.

### 2.3 Εξαφανιζόμενες και Εκρηκτικές Κλίσεις

Το πρόβλημα της εξαφανιζόμενης και εκρηκτικής κλίσης [43] έχει να κάνει με την εκθετική αύξηση ή συρρίκνωση των κλίσεων (Gradients) όταν αυτές διαδίδονται πίσω στο χρόνο. Δεν αποτελεί αποκλειστικό πρόβλημα των αναδρομικών νευρωνικών δικτύων αλλά εμφανίζεται αρκετά συχνά λόγω της πολυπλοκότητάς τους. Αφορά κυρίως μακροπρόθεσμες εξαρτήσεις και είναι πιο πιθανό να εμφανιστεί όταν υπάρχουν δεδομένα αποτελούμενα από μεγάλες αλληλουχίες και κατά επέκταση πολυάριθμα βήματα εκπαίδευσης.

Από τη μια στο πρόβλημα της εξαφανιζόμενης κλίσης, η κλίση μπορεί να γίνει τόσο μικρή έτσι ώστε να μην παρατηρείται ουσιαστική ενημέρωση και εκμάθηση των συνάψεων από κάποια παρελθοντικά βήματα και μετά. Αυτό κάνει το μοντέλο να εκπαιδεύεται έχοντας ως βάση βραχυπρόθεσμα στοιχεία και έτσι να μην είναι ικανό να μάθει περίπλοκες αναπαραστάσεις και μοτίβα. Αντιθέτως, στο πρόβλημα της εκρηκτικής κλίσης παρατηρείται εκθετική αύξηση της κλίσης κατά τη διάδοση στο χρόνο που οδηγεί σε επιθετική ενημέρωση των συνάψεων του δικτύου. Αυτή η μεγαλύτερη επιρροή των μακροπρόθεσμων στοιχείων μπορεί να προκαλέσει αστάθεια και αποκλίσεις.

Προσεγγίζοντας το πρόβλημα με μαθηματικά, αρκεί να αναλυθεί διανυσματικά η εξής μερική παράγωγος:

$$\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\| = \left\| \text{diag}(\mathcal{F}_H'(p_i)) \cdot W_{HH}^T \right\| \leq \left\| \text{diag}(\mathcal{F}_H'(p_i)) \right\| \left\| W_{HH}^T \right\| \quad (2.18)$$

Η τιμή της παραγώγου εξαρτάται από τη συνάρτηση ενεργοποίησης που έχει επιλεγεί.

Αν θεωρηθεί πως αυτή η τιμή είναι ίση με μια μεταβλητή  $\gamma$ , τότε οι τιμές της για 2 από τις πιο συχνά εμφανιζόμενες συναρτήσεις ενεργοποίησης του κρυφού επιπέδου είναι ίσες με:

$$\begin{cases} \mathcal{F}_H'(p_i) \leq \frac{1}{4} = \gamma, \text{ εάν } \mathcal{F}_H \text{ σιγμοειδής} \\ \mathcal{F}_H'(p_i) \leq 1 = \gamma, \text{ εάν } \mathcal{F}_H \text{ tanh} \end{cases} \quad (2.19)$$



Έστω επίσης ότι η νόρμα του πίνακα βάρους κρυφού επιπέδου είναι ίση με μια μεταβλητή  $\lambda$ . Αυτή αντιπροσωπεύει επίσης το μεγαλύτερο μέγεθος των ιδιοτιμών του πίνακα βαρών του κρυφού επιπέδου. Έτσι προσεγγίζοντας το πρόβλημα συνολικά, η εξίσωση (2.10) παίρνει την μορφή:

$$\left\| \frac{\partial h_t}{\partial h_r} \right\| = \left\| \prod_{i=r+1}^t \frac{\partial h_i}{\partial h_{i-1}} \right\| \leq \prod_{i=r+1}^t \gamma \cdot \lambda \leq (\gamma \cdot \lambda)^{(t-r)} \quad (2.20)$$

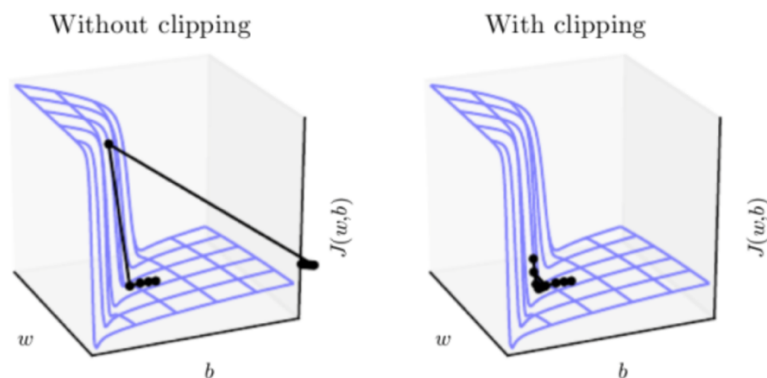
Μέσω της εξίσωσης (2.20) μπορεί να πραγματοποιηθεί η εξαγωγή των παρακάτω συμπερασμάτων:

- Αν  $(\gamma \cdot \lambda) < 1$ , τότε η κλίση θα ‘εξαφανισθεί’ (*Vanishing Gradient Problem*)
- Αν  $(\gamma \cdot \lambda) > 1$ , τότε η κλίση θα ‘εκτοξευθεί’ (*Exploding Gradient Problem*)

Σύντομα μετά τη δημιουργία και την εκπαίδευση μέσω οπισθοδιάδοσης των πρώτων αναδρομικών δικτύων, έγινε εμφανές το πρόβλημα των εξαφανιζόμενων και εκρηκτικών κλίσεων σε δεδομένα αποτελούμενα από μεγάλες αλληλουχίες και κατά επέκταση πολυάριθμα βήματα εκπαίδευσης. Στα επόμενα χρόνια πραγματοποιήθηκαν προσπάθειες αντιμετώπισης αυτών των προβλημάτων εκμάθησης μακροπρόθεσμων εξαρτήσεων.

## 2.4 Περικοπή κλίσης

Η περικοπή κλίσης (*Gradient Clipping*) αποτελεί μια μέθοδο αντιμετώπισης του *Exploding Gradient Problem*. Δημιουργεί ένα όριο (*Threshold*) των πιθανών τιμών που μπορεί να πάρει η παράγωγος σφάλματος και αναπροσαρμόζει την κλίμακα με αποτέλεσμα να βελτιώνεται η συμπεριφορά της κλίσης όπως φαίνεται στο παράδειγμα του σχήματος 16. Χωρίζεται σε 2 διαφορετικές μεθόδους εφαρμογής [44].



**Σχήμα 16:** Συμπεριφορά κλίσης παραγώγου με και χωρίς τη χρήση της μεθόδου περικοπής κλίσης

- Περικοπή μέσω νόρμας

Η νόρμα αποτελεί μία συνάρτηση που ορίζεται πάνω σε ένα διανυσματικό χώρο και η οποία αναθέτει ένα μη αρνητικό πραγματικό αριθμό σε κάθε διάνυσμα. Σε αυτή τη μέθοδο ορίζεται η οριακή τιμή μιας L2 νόρμας και σε περίπτωση που αυτή την υπερβεί, θα υπάρξει κατάλληλη αλλαγή των διανυσματικών τιμών.

$$g \leftarrow \partial C / \partial W$$

if  $\|g\| \geq \text{threshold}$  then

$$g \leftarrow \text{threshold} * g / \|g\|$$

Όπου η  $g$  η τιμή της κλίσης και  $\|g\|$  η νόρμα της κλίσης. Το  $g/\|g\|$  αποτελεί μοναδιαίο διάνυσμα και έτσι η κλίση παραμένει στα όρια με μέγιστη δυνατή τιμή αυτή που έχουμε ορίσει ως `threshold`.

- Περικοπή μέσω τιμής

Στην περικοπή μέσω τιμής ορίζεται μια μέγιστη και μια ελάχιστη οριακή τιμή που παίρνει η κλίση σε περίπτωση που τις υπερβεί.

$$g \leftarrow \partial C / \partial W$$

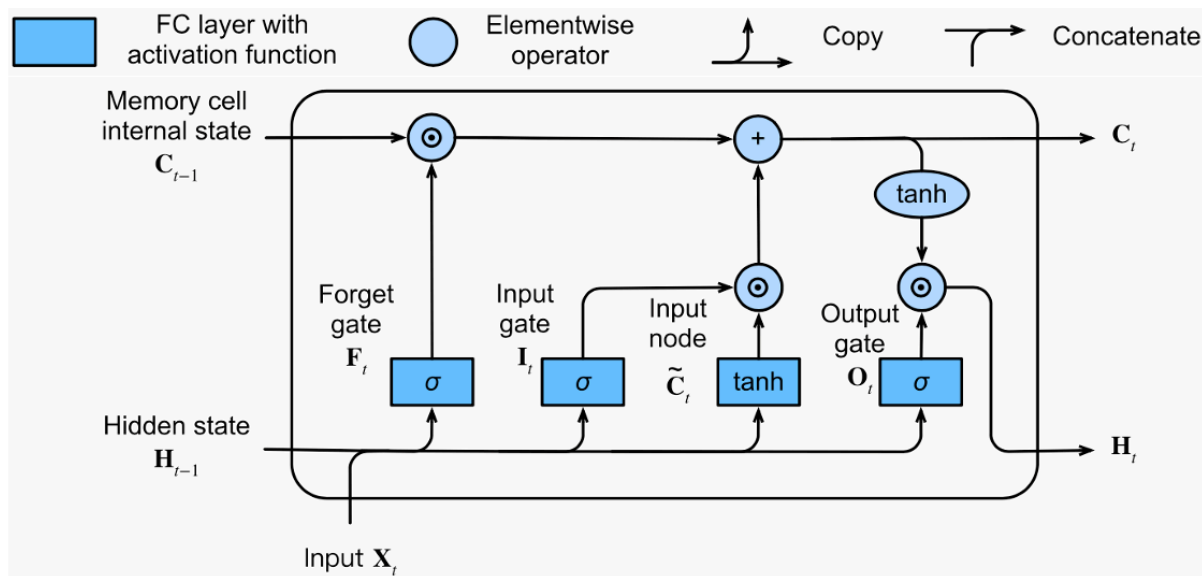
if  $\|g\| \geq \text{max\_threshold}$  or  $\|g\| \leq \text{min\_threshold}$  then

$$g \leftarrow \text{threshold (accordingly)}$$

## 2.5 Long Short-Term Memory (LSTM)

Παρόλο που η περικοπή κλίσης μπορεί να αντιμετωπίσει αποτελεσματικά το πρόβλημα της εκρηκτικής κλίσης, η εξαφανιζόμενη κλίση απαιτεί μια πιο σύνθετη προσέγγιση. Μια από τις πιο αποτελεσματικές λύσεις έρχεται να δώσει ο Hochreiter με τον Schmidhuber το 1997 μέσω του μοντέλου LSTM [45][46] που απεικονίζεται στο σχήμα 17. Αποτελεί μια αρχιτεκτονική αναδρομικών νευρωνικών δικτύων όπου το κρυφό επίπεδο εκμεταλλεύεται πληροφορίες τόσο από την μακροπρόθεσμη μνήμη όσο και από τη βραχυπρόθεσμη μνήμη καθώς και τις μεταξύ τους αλληλεπιδράσεις μέσω κάποιων πυλών. Η βραχυπρόθεσμη μνήμη αναφέρεται στη πληροφορία που δέχεται το δίκτυο από την πιο πρόσφατη είσοδο της ακολουθίας. Αυτή, όπως και στα απλά αναδρομικά νευρωνικά δίκτυα, αποθηκεύεται στο κρυφό επίπεδο του δικτύου και ανανεώνεται σε κάθε χρονικό βήμα. Αντιθέτως η μακροπρόθεσμη μνήμη αναφέρεται στη πληροφορία που διατηρεί το δίκτυο σε μεγαλύτερα χρονικά διαστήματα μέσω των συναπτικών

βαρών. Αυτά ανανεώνονται κατά τη διάρκεια της εκπαίδευσης κωδικοποιώντας γενικότερες γνώσεις για τα δεδομένα. Η βασική διαφορά μεταξύ των απλών αναδρομικών δικτύων και των LSTM διακρίνεται στο ότι τα LSTM υποστηρίζουν gating του κρυφού επιπέδου. Το μοντέλο LSTM αποτελείται από ειδικά διαμορφωμένους μηχανισμούς οι οποίοι είναι ικανοί να αποφασίσουν αν πρέπει να υπάρξει ενημέρωση ή επαναφορά των τιμών ανάλογα με το πόσο σημαντική είναι η πληροφορία μιας ορισμένης χρονικής στιγμής για την εκπαίδευση.



Σχήμα 17: Απεικόνιση πλήρους μοντέλου LSTM [65]

Το μοντέλο LSTM αποτελείται από ένα κρυφό επίπεδο που περιέχει μια κρυφή κατάσταση  $H$  και μια κατάσταση εσωτερικής μνήμης  $C$ . Η κρυφή κατάσταση περιγράφει τη βραχυπρόθεσμη μνήμη και περιέχει τη πληροφορία που περνάει στο επόμενο κρυφό επίπεδο αλλά και εξωτερικά για τον υπολογισμό της εξόδου. Η κατάσταση εσωτερικής μνήμης  $C$  περιγράφει τη μακροπρόθεσμη μνήμη που διαδίδεται αποκλειστικά εσωτερικά στο χρόνο. Αυτές οι καταστάσεις αλληλοεπιδρούν μεταξύ τους μέσω τεσσάρων ορισμένων διαδικασιών. Παρουσιάζει επίσης μια πιο σύνθετη αρχιτεκτονική από τα απλά αναδρομικά δίκτυα και παρέχει μια πολυπλοκότητα πράξεων που απαιτεί ένα μεγαλύτερο μέρος μνήμης και χρόνου εκπαίδευσης. Επιτρέπεται όμως η εισαγωγή μεγαλύτερων ακολουθιών δεδομένων χωρίς εμφάνιση προβλημάτων.

Το LSTM δίκτυο χρησιμοποιεί 4 διαφορετικούς πίνακες βάρους καθώς και 4 διανύσματα πόλωσης που αναλογούν σε κάθε διαδικασία που εφαρμόζεται. Αναλυτικότερα υπάρχει διαφορετικός πίνακας βάρους για το επίπεδο εισόδου και το κρυφό επίπεδο με αποτέλεσμα ο αριθμός τους να αυξάνεται στους 8. Υπάρχει όμως δυνατότητα ένωσης των πινάκων που έχει

ως αποτέλεσμα τη μείωση των απαιτούμενων πράξεων. Στον πίνακα 4 παρουσιάζονται οι διαστάσεις των παραμέτρων ενός LSTM δικτύου.

**Πίνακας 4:** Διαστάσεις παραμέτρων ενός δικτύου LSTM

Έστω $h$ οι κρυφές μονάδες, $n$ το μέγεθος παρτίδας (batch) και $d$ ο αριθμός εισόδων	
ΠΑΡΑΜΕΤΡΟΣ	ΔΙΑΣΤΑΣΕΙΣ
Είσοδος $X_t$	$n \times d$
Κρυφό επίπεδο $H_{t-1}$	$n \times h$
Διανύσματα πόλωσης $b_f, b_i, b_c, b_o$	$1 \times h$
ΑΝΑΛΥΤΙΚΑ	
Πίνακες βάρους επιπέδου εισόδου $W_{xf}, W_{xi}, W_{xc}, W_{xo}$	$d \times h$
Πίνακες βάρους κρυφού επιπέδου $W_{hf}, W_{hi}, W_{hc}, W_{ho}$	$h \times h$
ΜΕ ΕΝΩΣΗ ΠΙΝΑΚΩΝ	
Πίνακες Βάρους $W_f, W_i, W_c, W_o$	$(d + h) \times h$

Κάθε μια από τις 4 διαδικασίες του δικτύου περιλαμβάνει ένα πλήρως συνδεδεμένο επίπεδο που περιέχει μια συνάρτηση ενεργοποίησης. Οι πύλες του δικτύου περιλαμβάνουν τη σιγμοειδή συνάρτηση ενεργοποίησης διαστήματος  $[0,1]$  όπου ουσιαστικά περιγράφει ένα ποσοστό συμμετοχής. Τιμή σιγμοειδούς συνάρτησης ίση με το 0 περιγράφει εγκατάλειψη της πληροφορίας ενώ τιμή ίση με 1 πλήρη διατήρησή της. Αντίστοιχα, οι υπόλοιποι κόμβοι περιέχουν μια συνάρτηση ενεργοποίησης υπερβολικής εφαπτομένης διαστήματος  $[-1,1]$ .

Forget Gate: Συνδέεται άμεσα με το επίπεδο μνήμης και είναι υπεύθυνη για το ποσοστό της μακροπρόθεσμης μνήμης που θα διατηρηθεί στο δίκτυο. Η πληροφορία που θα αφαιρεθεί από την κατάσταση εσωτερικής μνήμης του προηγούμενου βήματος θα εξαρτάται από την τιμή της σιγμοειδούς συνάρτησης. Υπολογίζεται μέσω της εξίσωσης (2.21).

$$F_t = \sigma(H_{t-1}W_{hf} + X_tW_{xf} + b_f) \Leftrightarrow F_t = \sigma(W_f \cdot [H_{t-1}, X_t] + b_f) \quad (2.21)$$

Input Node: Αυτό αποτελεί μια πιθανή μακροπρόθεσμη μνήμη της εισόδου που μπορεί να προστεθεί στη κατάσταση εσωτερικής μνήμης. Ο υπολογισμός πραγματοποιείται μέσω της εξίσωσης (2.22).

$$\tilde{C}_t = \tanh(H_{t-1}W_{hc} + X_tW_{xc} + b_c) \Leftrightarrow \tilde{C}_t = \tanh(W_c \cdot [H_{t-1}, X_t] + b_c) \quad (2.22)$$

Input Gate: Είναι υπεύθυνη για το ποσοστό της πιθανής μακροπρόθεσμης μνήμης της εισόδου  $\tilde{C}_t$  που προστίθεται στη κατάσταση εσωτερικής μνήμης. Υπολογίζεται μέσω της εξίσωσης (2.23).

$$I_t = \sigma(H_{t-1}W_{hi} + X_tW_{xi} + b_i) \Leftrightarrow I_t = \sigma(W_i \cdot [H_{t-1}, X_t] + b_i) \quad (2.23)$$

Memory Cell Internal State: Αποτελεί όλη την πάνω ευθεία του σχήματος 17 και περιγράφει την εσωτερική κατάσταση της μακροπρόθεσμης μνήμης του δικτύου. Ο υπολογισμός πραγματοποιείται μέσω της εξίσωσης (2.24).

$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t \quad (2.24)$$

Το πλεονέκτημα αυτής της κατάστασης είναι ότι δεν περιέχει βάρη και πολώσεις που μπορούν να την επηρεάσουν άμεσα. Ωστόσο, οι πύλη εισόδου και η πύλη λήθης δίνουν την ευελιξία να μάθει πότε πρέπει να παραμένει σταθερή και πότε να μεταβάλλεται κατά το πέρασμα του χρόνου. Αυτό έχει ως αποτέλεσμα την αισθητή ελάττωση του προβλήματος της εξαφανιζόμενης κλίσης. Μέσω της κατάστασης εσωτερικής μνήμης και μιας συνάρτησης υπερβολικής εφαπτομένης, δημιουργείται μια νέα πιθανή βραχυπρόθεσμη μνήμη.

Output Gate: Είναι υπεύθυνη για το ποσοστό συμμετοχής της πιθανής βραχυπρόθεσμης μνήμης στη νέα τιμή της κρυφής κατάστασης. Υπολογίζεται μέσω της εξίσωσης (2.25).

$$O_t = \sigma(H_{t-1}W_{ho} + X_tW_{xo} + b_o) \Leftrightarrow O_t = \sigma(W_o \cdot [H_{t-1}, X_t] + b_o) \quad (2.25)$$

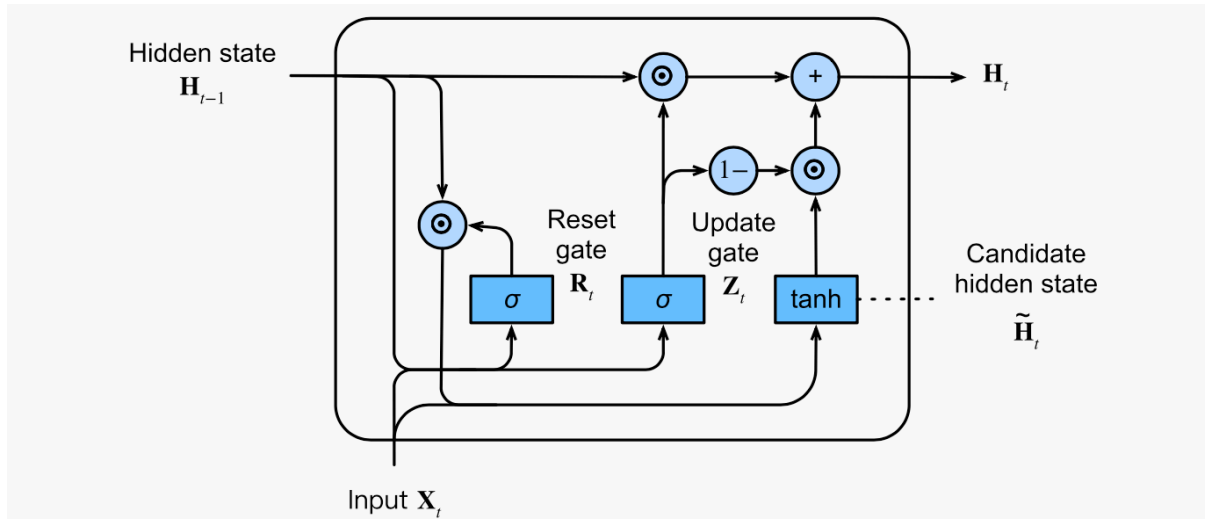
Hidden State: Στο τέλος, υπολογίζεται μέσω της εξίσωσης (2.26) η νέα τιμή της κρυφής κατάστασης  $H$  όπου περνάει στο επόμενο κρυφό επίπεδο και αλλά και στο επίπεδο εξόδου.

$$H_t = O_t \odot \tanh(C_t) \quad (2.26)$$

## 2.6 Gated Recurrent Unit (GRU)

Ύστερα από τη δημιουργία των LSTM, πραγματοποιήθηκαν πολλές έρευνες που είχαν ως σκοπό τόσο την εξέλιξη όσο και την απλοποίησή τους. Στόχος ήταν η διατήρηση μιας αρχιτεκτονικής που ενσωματώνει μια κατάσταση εσωτερικής μνήμης καθώς και των πολλαπλασιαστικών μηχανισμών πυλών που επιτρέπουν την δημιουργία διαδικασιών

αλληλεπίδρασης με την κρυφή κατάσταση του δικτύου. Μια από τις πιο αποδοτικές προτάσεις έγινε από τον Cho το 2014 [47] όπου εισήγαγε την έννοια των φραγμένων αναδρομικών μονάδων. Τα GRU αποτελούν μια απλούστερη αρχιτεκτονική η οποία σε πολλές περιπτώσεις έχει συγκρίσιμη απόδοση με τα LSTM ενώ απαιτεί μικρότερο χρόνο εκτέλεσης. Η δομή του περιγράφεται στο σχήμα 18.



**Σχήμα 18:** Απεικόνιση πλήρους μοντέλου GRU [65]

Στο συγκεκριμένο μοντέλο οι πύλες που χρησιμοποιούνται έχουν μειωθεί σε 2 καθώς η πύλη λήθης και η πύλη εισόδου έχουν ενωθεί σε μια πύλη ενημέρωσης. Επιπλέον, έχει ενωθεί και η κρυφή κατάσταση με την εσωτερική κατάσταση μνήμης αλλά έχουν παραμείνει κατάλληλες διαδικασίες διαχείρισης μακροπρόθεσμης και βραχυπρόθεσμης μνήμης. Αντίστοιχα αποτελείται από πίνακες βάρους για κάθε διαδικασία. Αυτοί με τη σειρά τους μπορούν να αναλυθούν για επίπεδο εισόδου και κρυφό επίπεδο. Στον πίνακα 5 παρουσιάζονται οι διαστάσεις των παραμέτρων ενός GRU δικτύου.

**Πίνακας 5:** Διαστάσεις παραμέτρων ενός δικτύου GRU

Έστω $\mathbf{h}$ οι κρυφές μονάδες, $\mathbf{n}$ το μέγεθος παρτίδας (batch) και $\mathbf{d}$ ο αριθμός εισόδων	
ΠΑΡΑΜΕΤΡΟΣ	ΔΙΑΣΤΑΣΕΙΣ
Είσοδος $X_t$	$n \times d$
Κρυφό επίπεδο $H_{t-1}$	$n \times h$
Διανύσματα πόλωσης $b_r, b_z, b_h$	$1 \times h$
ΑΝΑΛΥΤΙΚΑ	
Πίνακες βάρους επίπεδου εισόδου $W_{xr}, W_{xz}, W_{xh}$	$d \times h$
Πίνακες βάρους κρυφού επίπεδου $W_{hr}, W_{hz}, W_{hh}$	$h \times h$
ΜΕ ΕΝΩΣΗ ΠΙΝΑΚΩΝ	
Πίνακες Βάρους $W_r, W_z, W_h$	$(d + h) \times h$

Reset Gate: Η πύλη επαναφοράς είναι υπεύθυνη για το ποσοστό συμμετοχής της κρυφής κατάστασης του προηγούμενου βήματος. Υπολογίζεται μέσω της εξίσωσης (2.27).

$$R_t = \sigma(H_{t-1}W_{hr} + X_tW_{xr} + b_r) \Leftrightarrow R_t = \sigma(W_r \cdot [H_{t-1}, X_t] + b_r) \quad (2.27)$$

Candidate Hidden State: Αποτελεί μια νέα πιθανή μνήμη της κρυφής κατάστασης του δικτύου για μια συγκεκριμένη τιμή εισόδου. Ο υπολογισμός της πραγματοποιείται μέσω της εξίσωσης (2.28).

$$\tilde{H}_t = \tanh((R_t \odot H_{t-1})W_{hh} + X_tW_{xh} + b_h) \Leftrightarrow \tilde{H}_t = \tanh(W_h \cdot [R_t \odot H_{t-1}, X_t] + b_h) \quad (2.28)$$

Update Gate: Η πύλη ενημέρωσης αφορά τις μακροπρόθεσμες εξαρτήσεις. Είναι υπεύθυνη για το ποσοστό συμμετοχής της προηγούμενης κρυφής κατάστασης και ταυτόχρονα αποτελεί μέτρο σύγκρισής με τη νέα πιθανή μνήμη. Υπολογίζεται μέσω της εξίσωσης (2.29).

$$Z_t = \sigma(H_{t-1}W_{hz} + X_tW_{xz} + b_z) \Leftrightarrow Z_t = \sigma(W_z \cdot [H_{t-1}, X_t] + b_z) \quad (2.29)$$

Hidden State: Αποτελεί την τιμή της κρυφής κατάστασης η οποία περνάει στο επόμενο επίπεδο αλλά και στο επίπεδο εξόδου. Ο υπολογισμός της πραγματοποιείται μέσω της εξίσωσης (2.30).

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t \quad (2.30)$$

Παρατηρείται πως αν η πύλη ενημέρωσης πάρει μια τιμή κοντά στο 1, παραμένει η παλιά κατάσταση ενώ αν πάρει μια τιμή κοντά στο 0, η κρυφή κατάσταση πλησιάζει τη πιθανή τιμή που την αντικαθιστά.

## 2.7 Σύντομη σύγκριση μεταξύ LSTM και GRU

Το πιο σημαντικό χαρακτηριστικό που μοιράζονται είναι το πρόσθετο στοιχείο στην ενημέρωσή τους από χρονική στιγμή  $t$  σε  $t+1$ . Τα απλά αναδρομικά δίκτυα αντικαθιστούν πάντα τη πληροφορία με μια νέα τιμή που υπολογίζεται από τη τρέχουσα είσοδο και τη προηγούμενη κρυφή κατάσταση. Αντιθέτως, τα LSTM και GRU διατηρούν το υπάρχον περιεχόμενο και προσθέτουν πάνω σε αυτό. Αυτή η προσθετική φύση περιλαμβάνει 2 βασικά πλεονεκτήματα. Πρώτον, είναι εύκολο για κάθε μονάδα να θυμάται την ύπαρξη μιας συγκεκριμένης πληροφορίας στη ροή εισόδου για μια μεγάλη σειρά βημάτων. Οποιοδήποτε

σημαντική πληροφορία, που αποφασίζεται είτε από την forget gate του δικτύου LSTM είτε από την πύλη update gate του δικτύου GRU, δεν θα αντικατασταθεί αλλά θα διατηρηθεί ως έχει. Δεύτερον, και ίσως πιο σημαντικό, αυτή η προσθήκη δημιουργεί αποτελεσματικά μονοπάτια συντόμευσης που παρακάμπτουν πολλαπλά χρονικά βήματα. Αυτές οι συντομεύσεις επιτρέπουν στο σφάλμα να επαναδιαδίδεται εύκολα χωρίς να εξαφανίζεται πολύ γρήγορα (εάν η μονάδα πύλης είναι σχεδόν κορεσμένη στο 1) ως αποτέλεσμα της διέλευσης από πολλαπλές, οριοθετημένες μη γραμμικότητες, μειώνοντας έτσι τη δυσκολία λόγω των κλίσεων εξαφάνισης [48].

Ωστόσο, αυτές οι δύο μονάδες παρουσιάζουν αρκετές διαφορές. Ένα χαρακτηριστικό του δικτύου LSTM που λείπει από το GRU, είναι η ελεγχόμενη έκθεση του περιεχομένου της μνήμης. Στη μονάδα LSTM, η ποσότητα του περιεχομένου μνήμης που εμφανίζεται ή χρησιμοποιείται από άλλες μονάδες στο δίκτυο ελέγχεται από την πύλη εξόδου. Από την άλλη πλευρά, η GRU εκθέτει το πλήρες περιεχόμενό της χωρίς κανέναν έλεγχο. Μια άλλη βασική διαφορά έχει να κάνει με την πολυπλοκότητα των πράξεων αλλά και τον αριθμό των παραμέτρων. Πιο συγκεκριμένα, για  $m$  ως μέγεθος διανύσματος εισόδου  $X_t$  και  $n$  αριθμό κρυφών μονάδων στο επίπεδο  $h_t$  παρουσιάζεται η πολυπλοκότητα των πράξεων μεταξύ των επιπέδων στον πίνακα 6 και η πολυπλοκότητα των παραμέτρων στον πίνακα 7:

**Πίνακας 6:** Σύγκριση πολυπλοκότητας πράξεων μεταξύ LSTM και GRU [49]

RNN Επίπεδο	Αριθμός Υπολογισμών		
	Πολλαπλασιασμοί	Προσθέσεις	Μη γραμμικοί
LSTM	$4n^2 + 4nm + 3n$	$4n^2 + 4nm + 5n$	$5n$
	$= LSTM_{mul}$	$= LSTM_{add}$	$= LSTM_{nonlinear}$
GRU	$3n^2 + 3nm + 3n$	$3n^2 + 3nm + 2n$	$3n$
	$= 0.75 LSTM_{mul}$	$= 0.75 LSTM_{add}$	$= 0.6 LSTM_{nonlinear}$

**Πίνακας 7:** Σύγκριση πολυπλοκότητας παραμέτρων μεταξύ LSTM και GRU [49]

RNN Επίπεδο	Αριθμός Παραμέτρων	
	Βάρη	Πολώσεις
LSTM	$4n^2 + 4nm$	$4n$
	$= LSTM_{weights}$	$= LSTM_{biases}$
GRU	$3n^2 + 3nm$	$3n$
	$= 0.75 LSTM_{weights}$	$= 0.75 LSTM_{biases}$

Κάθε μια από τις 2 αρχιτεκτονικές παρουσιάζει πλεονεκτήματα και αδυναμίες. Η απόδοσή τους εξαρτάται επίσης και από τα δεδομένα και επομένως μια συνηθισμένη πρακτική είναι να εξετάζονται και ύστερα να επιλέγεται η καλύτερη αρχιτεκτονική για τη συγκεκριμένη περίπτωση.



---

### ΚΕΦΑΛΑΙΟ 3<sup>ο</sup> : ΥΛΟΠΟΙΗΣΗ ΜΕΘΟΔΩΝ ΕΚΠΑΙΔΕΥΣΗΣ ΑΝΑΔΡΟΜΙΚΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΣΕ ΡΥΤΗΘΝ

---

Μία από τις βασικότερες ιδέες και κίνητρα πίσω από τον πολύπλευρο τομέα του προγραμματισμού ηλεκτρονικών υπολογιστών είναι η αυτοματοποίηση και η ενίσχυση επαναλαμβανόμενων διαδικασιών. Στον κλασσικό προγραμματισμό, ο χρήστης δημιουργεί ένα σετ κανόνων που αποτελούν το πρόγραμμα. Αυτό επεξεργάζεται ένα σύνολο δεδομένων εισόδου και στη συνέχεια υπολογίζει εξόδους. Αυτή η διαδικασία αποτελεί πρότυπο της συμβολικής τεχνητής νοημοσύνης. Ωστόσο, η ανάπτυξη ενός συνόλου κανόνων που μπορούν να ενσωματωθούν σε ένα πρόγραμμα υπολογιστή, είναι πολλές φορές κουραστική και εξαιρετικά προκλητική. Η υπολογιστική νοημοσύνη και οι πρακτικές της μηχανικής μάθησης προσπαθούν να ενσωματώσουν λύσεις σε αυτά τα προβλήματα ακολουθώντας μια διαφορετική πρακτική ανάπτυξης. Στόχος είναι η κατασκευή ενός μοντέλου που δέχεται τα δεδομένα εισόδου και τις εξόδους με σκοπό τη δημιουργία του σετ κανόνων. Αυτό με τη σειρά του θα πρέπει να είναι ικανό να εφαρμοστεί σε νέα δεδομένα εισόδου και να παράγει πρωτότυπα συμπεράσματα.

Με το πέρασμα των χρόνων, έχει χρησιμοποιηθεί ένα ευρύ φάσμα διαφορετικών γλωσσών προγραμματισμού και περιβαλλόντων που επιτρέπουν την έρευνα και την ανάπτυξη εφαρμογών μηχανικής μάθησης. Ωστόσο, καθώς η γενικής χρήσης γλώσσα Python γνώρισε τεράστια αύξηση δημοτικότητας στην επιστημονική κοινότητα την τελευταία δεκαετία, οι πιο πρόσφατες βιβλιοθήκες μηχανικής μάθησης και βαθιάς μάθησης βασίζονται πλέον σε αυτή [50]. Στόχος αυτού του κεφαλαίου είναι η παρουσίαση αυτών των βιβλιοθηκών και εργαλείων, προκειμένου να κατασκευαστούν διάφορα μοντέλα νευρωνικών δικτύων με ιδιαίτερη έμφαση στα αναδρομικά νευρωνικά δίκτυα.

Αρχικά, θα υλοποιηθεί το γραμμικό μοντέλο των ελαχίστων τετραγώνων και το μοντέλο πρόσθιας τροφοδότησης πολλών επιπέδων (MLP) με στόχο τη σύγκριση της απόδοσής τους σε σχέση με τα αναδρομικά μοντέλα. Στη συνέχεια, θα πραγματοποιηθεί ανάπτυξη ενός απλού αναδρομικού νευρωνικού δικτύου που εκπαιδεύεται με τη μέθοδο οπισθοδιάδοσης στο χρόνο για την ανάδειξη πλεονεκτημάτων και περιορισμών σε σχέση με το αντίστοιχο μοντέλο που παρέχεται από την καθιερωμένη βιβλιοθήκη του Keras. Η συγκεκριμένη βιβλιοθήκη θα χρησιμοποιηθεί ακόμη και για τη κατασκευή αναδρομικών μοντέλων αυξημένης πολυπλοκότητας που στοχεύουν στην αντιμετώπιση του προβλήματος εμφάνισης της εξαφανιζόμενης κλίσης, όπως είναι το μοντέλο LSTM και το μοντέλο GRU.

### 3.1 Η γλώσσα προγραμματισμού Python

Η Python αποτελεί μια εξαιρετικά ισχυρή γλώσσα προγραμματισμού η οποία συνδυάζει τη δύναμη των γλωσσών προγραμματισμού γενικού σκοπού (General Purpose) με την ευκολία χρήσης γλωσσών προγραμματισμού συγκεκριμένης περιοχής (Domain-Specific) όπως είναι το Matlab ή η R. Αποτελεί μια υψηλού επιπέδου ερμηνευμένη γλώσσα προγραμματισμού η οποία βασίζεται στην αναγνωσιμότητα και στην ευκολία εκμάθησης από τον χρήστη. Η Python διαθέτει βιβλιοθήκες για φόρτωση δεδομένων, οπτικοποίηση, στατιστικά στοιχεία, επεξεργασία φυσικής γλώσσας, επεξεργασία εικόνας και πολλές άλλες εφαρμογές. Παρέχει εργαλεία που βασίζονται στην άμεση αλληλεπίδραση με τον κώδικα και μπορεί να γίνει χρήση της μέσω κάποιου τερματικού. Ως γλώσσα προγραμματισμού γενικής χρήσης, η Python επιτρέπει επίσης τη δημιουργία σύνθετων γραφικών διεπαφών χρήστη (GUIs) και διαδικτυακών υπηρεσιών, καθώς και την ενσωμάτωση σε υπάρχοντα συστήματα. Πέρα από τα πλεονεκτήματά της, υποστηρίζεται από μια τεράστια κοινότητα που εκμεταλλεύεται τον ανοιχτού κώδικα σχεδιασμό της ενώ δέχεται συνεχώς επίσημες αναβαθμίσεις βελτιώνοντας έτσι τις δυνατότητες και τις αποδόσεις της.

### 3.2 Ο περιηγητής Anaconda

Πολλές φορές προκειμένου να εκτελεστούν τα πακέτα βιβλιοθηκών, εξαρτώνται από συγκεκριμένες εκδόσεις άλλων πακέτων. Για την εκτέλεση ενός προγράμματος μπορεί ακόμη να απαιτηθεί και η χρήση πολυάριθμων εκδόσεων που διαχωρίζονται από περιβάλλοντα. Το πρόγραμμα διεπαφής γραμμής εντολών (CLI) conda λειτουργεί ως διαχειριστής αυτών των πακέτων και των περιβαλλόντων. Με αυτόν τον τρόπο εξασφαλίζεται ότι κάθε έκδοση κάθε πακέτου περιλαμβάνει όλες τις κατάλληλες εξαρτήσεις για την σωστή λειτουργία της. Ο περιηγητής Anaconda αποτελεί μια γραφική διεπαφή χρήστη (GUI) που επιτρέπει την πραγματοποίηση αυτών των λειτουργιών χωρίς να χρειάζεται να πληκτρολογηθούν οι εντολές σε κάποιο παράθυρο τερματικού. Το Navigator μπορεί να αναζητήσει πακέτα στο Anaconda Cloud ή σε ένα τοπικό αποθετήριο Anaconda. Στη συνέχεια μπορεί να τα εγκαταστήσει σε κάποιο περιβάλλον και να τα ενημερώσει όταν αυτό είναι απαραίτητο. Ο περιηγητής είναι διαθέσιμος για Windows, macOS και Linux.

### 3.3 Το περιβάλλον του Jupyter Notebook

Τα ολοκληρωμένα περιβάλλοντα ανάπτυξης (IDE) έχουν ως στόχο την αύξηση της παραγωγικότητας του χρήστη συνδυάζοντας πολλαπλά βοηθητικά προγράμματα σε μια ενιαία διεπαφή. Με τη βοήθειά τους, κομμάτια κώδικα και εφαρμογές λογισμικού μπορούν να γραφτούν, να δοκιμαστούν, να εξεταστούν για σφάλματα καθώς και να επεξεργαστούν μέσω

διαφόρων λειτουργιών που προσφέρονται. Το Jupyter Notebook δημιουργήθηκε το 2014 και αποτελεί ένα από αυτά τα IDE. Αποτελεί μια εφαρμογή διακομιστή-πελάτη (Server-Client Application) που επιτρέπει την επεξεργασία και εκτέλεση προγραμμάτων μέσω ενός περιηγητή ιστού (Web Browser). Στην αρχή είχε σχεδιαστεί να υποστηρίζει τις γλώσσες προγραμματισμού Python, R και Julia, όμως πλέον υποστηρίζει πάνω από 40 διαφορετικές γλώσσες προγραμματισμού. Αποτελείται από δύο βασικά στοιχεία:

- **Kernel:** Είναι ένα πρόγραμμα που τρέχει και εξετάζει τον κώδικα του χρήστη. Από προεπιλογή διαθέτει Kernel σε γλώσσα Python που μπορεί να αλλάξει.
- **Dashboard:** Αποτελείται από μια ακολουθία κελιών τα οποία μπορούν να περιλαμβάνουν στοιχεία όπως κώδικες, κείμενα, εικόνες, απεικονίσεις και εξισώσεις. Τα κελιά μπορούν να αναλυθούν και να εκτελεστούν αυτόνομα. Επιπλέον, διαθέτει και μια γραμμή εργαλείων που βοηθούν στη διαμόρφωση της ροής εργασιών και στην επεξεργασία των αρχείων.

### 3.4 Η πλατφόρμα της TensorFlow

Η TensorFlow αποτελεί μια βιβλιοθήκη ανοιχτού κώδικα κατάλληλη για αριθμητικούς υπολογισμούς και εφαρμογές μηχανικής μάθησης μεγάλης κλίμακας. Δημιουργήθηκε από ομάδες έρευνας της Google και έγινε διαθέσιμη στο κοινό το 2015. Η TensorFlow χρησιμοποιεί ένα ενοποιημένο γράφημα ροής δεδομένων για να αναπαραστήσει τόσο τον υπολογισμό σε έναν αλγόριθμο όσο και την κατάσταση στην οποία λειτουργεί ο αλγόριθμος. Με αυτόν τον τρόπο επιτρέπει τον πειραματισμό διαφορετικών παράλληλων σχεδιασμών που μπορούν να εκφορτώσουν υπολογισμούς σε διακομιστές που διατηρούν κοινή κατάσταση προκειμένου να μειώσουν την κυκλοφορία του δικτύου. Κάθε κόμβος στο γράφημα αντιπροσωπεύει μια μαθηματική πράξη και κάθε σύνδεση ή ακμή μεταξύ των κόμβων είναι ένας πολυδιάστατος πίνακας δεδομένων ή tensor. Όλοι οι tensors είναι πυκνοί (Dense). Αυτή η απόφαση διασφαλίζει ότι τα χαμηλότερα επίπεδα του συστήματος μπορούν να έχουν απλές υλοποιήσεις για την εκχώρηση μνήμης και τη σειριοποίηση. Αντιστοιχίζει τους κόμβους ενός γραφήματος ροής δεδομένων σε πολλά μηχανήματα σε ένα σύμπλεγμα και μέσα σε ένα μηχανήμα σε πολλαπλές υπολογιστικές συσκευές, συμπεριλαμβανομένων πολυπύρηνων CPU, GPU γενικής χρήσης και προσαρμοσμένων σχεδιασμένων ASIC γνωστών ως Tensor Processing Units (TPU) που μπορούν να βρεθούν μέσω του Cloud της Google και προσφέρουν μεγαλύτερη επιτάχυνση υπολογισμών. Τον Οκτώβριο του 2019 κυκλοφόρησε μια νέα έκδοση της βιβλιοθήκης με το όνομα 'TensorFlow 2.0'. Δόθηκε ιδιαίτερη βάση στη βελτίωση της απόδοσης και στην ευκολία χρήσης [51].

### 3.5 Η Βιβλιοθήκη Keras

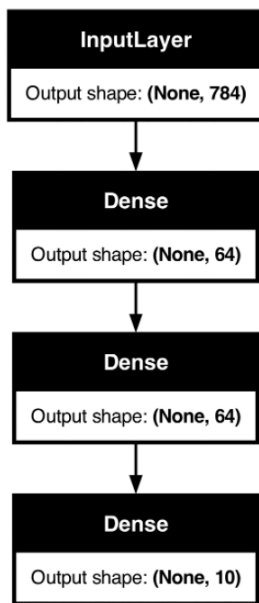
Το Keras αποτελεί μια επιπέδου-μοντέλου (Model-Level) βιβλιοθήκη, η οποία παρέχει δομικά στοιχεία υψηλού επιπέδου για την ανάπτυξη μοντέλων βαθιάς μάθησης. Δεν χειρίζεται λειτουργίες χαμηλού επιπέδου, όπως η διαχείριση tensor. Αντιθέτως, διαθέτει μια εξειδικευμένη και καλά βελτιστοποιημένη βιβλιοθήκη tensor που λειτουργεί ως μηχανή υποστήριξης. Αυτή δεν στηρίζεται σε μια συγκεκριμένη βιβλιοθήκη εκτέλεσης, όπως για παράδειγμα η TensorFlow, και έτσι οποιοδήποτε κομμάτι κώδικα γράφεται μέσω του Keras μπορεί να λειτουργήσει και να εκτελεστεί μέσω διαφορετικών βιβλιοθηκών. Αυτό είναι εξαιρετικά χρήσιμο καθώς μπορεί να πραγματοποιηθεί πολύ εύκολα η εναλλαγή μεταξύ των βιβλιοθηκών εκτέλεσης προκειμένου να βρεθεί εύκολα η πιο αποδοτική και γρήγορη για μια συγκεκριμένη εργασία. Η πιο ευρέως διαθέσιμη και επεκτάσιμη από αυτές είναι η TensorFlow αλλά υπάρχουν και άλλες όπως για παράδειγμα η Theano και το Microsoft Cognitive Toolkit. Το Keras μπορεί να εκτελείται τόσο με χρήση της CPU όσο και με της GPU. Στη CPU η TensorFlow χρησιμοποιεί μια βιβλιοθήκη χαμηλού επιπέδου που ονομάζεται Eigen ενώ όταν χρησιμοποιεί τη GPU χρησιμοποιεί μια βιβλιοθήκη καλά βελτιστοποιημένων λειτουργιών βαθιάς μάθησης που ονομάζεται NVIDIA CUDA deep neural network (cuDNN) [52].

#### 3.5.1 API Κατασκευής Μοντέλων στο Keras

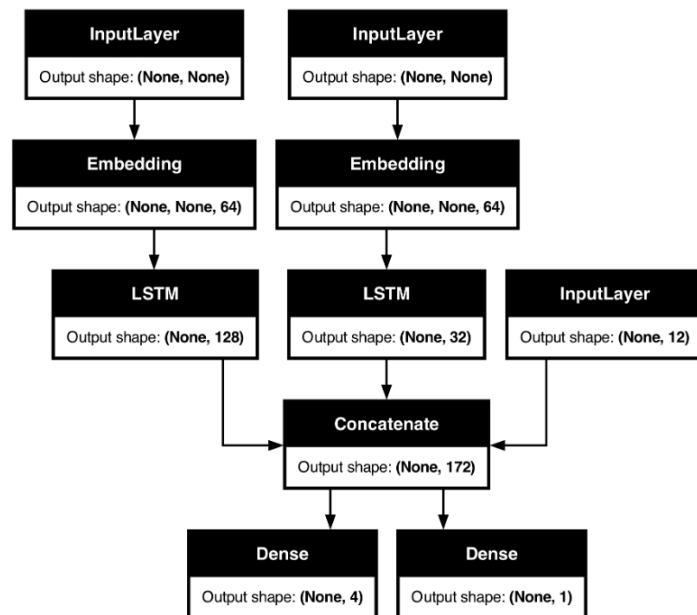
Το Keras προσφέρει τρεις διαφορετικές επιλογές κατασκευής μοντέλων μηχανικής μάθησης.

- Sequential API: Αποτελεί μια απλή γραμμική λίστα επιπέδων. Περιορίζεται σε στοιβές επιπέδων μιας εισόδου και εξόδου
- Functional API: Με αυτόν τον τρόπο μπορούν να δημιουργηθούν μοντέλα πολλών εισόδων και εξόδων αλλά και πολύπλοκης εσωτερικής τοπολογίας όπως φαίνεται στο παράδειγμα του σχήματος 19. Τα νευρωνικά δίκτυα επιτρέπεται να είναι αυθαίρετα κατευθυνόμενα άκυκλα γραφήματα επιπέδων. Οι συνδέσεις μεταξύ των επιπέδων σχηματίζουν ένα κατευθυνόμενο γράφημα όπου τα δεδομένα ρέουν προς μία κατεύθυνση, από την είσοδο στην έξοδο. Η μόνη περίπτωση που υπάρχει ανάδραση είναι όταν χρησιμοποιείται κάποιο επίπεδο αναδρομικού νευρωνικού δικτύου.
- Model Subclassing: Όταν όλες οι λειτουργίες και επίπεδα δημιουργούνται από την αρχή από τον ίδιο το χρήστη. Αυτή η επιλογή έχει νόημα σε περιπτώσεις έρευνας και πειραματισμού. Είναι πολύ αποδοτική αλλά απαιτεί καλή γνώση αντικειμενοστραφή προγραμματισμού.

### SEQUENTIAL API



### FUNCTIONAL API



Σχήμα 19: Παραδείγματα μοντέλων κατασκευασμένα με Sequential API και Functional API

### 3.6 Λοιπές απαραίτητες βιβλιοθήκες

Στη συγκεκριμένη διπλωματική εργασία χρησιμοποιήθηκαν και κάποιες άλλες βιβλιοθήκες πέρα από αυτή της TensorFlow και του Keras:

- Sci-kit learn: Αυτό αποτελεί μια βιβλιοθήκη μηχανικής μάθησης ανοιχτού κώδικα που υποστηρίζει εποπτευόμενη και μη εποπτευόμενη μάθηση. Παρέχει επίσης διάφορα εργαλεία κατάλληλα για την προεπεξεργασία δεδομένων, την επιλογή, προσαρμογή και αξιολόγηση ενός μοντέλου αλλά και πολλά άλλα βοηθητικά προγράμματα. Στη διπλωματική θα χρησιμοποιηθούν συναρτήσεις κανονικοποίησης δεδομένων εισόδου και υπολογισμού στατιστικών δεικτών της απόδοσης του μοντέλου.
- NumPy: Αυτό αποτελεί ένα από τα θεμελιώδη πακέτα για επιστημονικούς υπολογισμούς στη Python. Παρέχει ένα πολυδιάστατο αντικείμενο πίνακα, διάφορα παράγωγα αντικείμενα όπως και μια ποικιλία από ρουτίνες για γρήγορους υπολογισμούς. Μπορεί επίσης να υποστηρίζει πλήθος μαθηματικών συναρτήσεων υψηλού επιπέδου μεταξύ των αντικειμένων του όπως για παράδειγμα πράξεις γραμμικής άλγεβρας, ο μετασχηματισμός Fourier αλλά και γεννήτριες ψευδοτυχαίων αριθμών.
- Pandas: Αυτό αποτελεί μια βιβλιοθήκη ανοιχτού κώδικα που κατασκευάστηκε με στόχο τον χειρισμό και την ανάλυση δεδομένων. Αποτελείται από πίνακες που ονομάζονται

dataframes και μοιάζουν να έχουν μορφή υπολογιστικών φύλλων. Σε αντίθεση με τους πίνακες του NumPy, τα dataframes μπορούν να υποστηρίξουν ετερογενείς τύπους δεδομένων στις σειρές και τις στήλες τους. Επιπλέον, έχει την ικανότητα να δέχεται δεδομένα από μια μεγάλη ποικιλία μορφών αρχείων όπως αρχεία SQL, Excel και αρχεία τιμών διαχωρισμένων με κόμματα (CSV).

- Matplotlib: Αυτό αποτελεί την κύρια επιστημονική βιβλιοθήκη οπτικοποίησης δεδομένων και σχεδίασης γραφικών παραστάσεων. Αναπτύχθηκε για να παρέχει ένα ευέλικτο και ισχυρό εργαλείο δημιουργίας υψηλής ποιότητας δισδιάστατων και τρισδιάστατων γραφημάτων και απεικονίσεων στην Python. Αυτά μπορούν να είναι στατικά, κινούμενα αλλά ακόμη και διαδραστικά.
- Pydot: Αυτή είναι μια βιβλιοθήκη που αποτελεί διεπαφή με το λογισμικό οπτικοποίησης γραφημάτων ανοιχτού κώδικα Graphviz. Με αυτόν τον τρόπο θα πραγματοποιηθεί η απεικόνιση των μοντέλων που κατασκευάστηκαν μέσω των API του Keras.

### 3.7 Το μοντέλο των ελαχίστων τετραγώνων

Η μέθοδος των ελαχίστων τετραγώνων αποτελεί μια διαδικασία που βασίζεται σε απλά μαθηματικά και εξισώσεις γραμμικής άλγεβρας προκειμένου να πραγματοποιηθεί η εύρεση μιας βέλτιστης ευθείας η οποία προσαρμόζεται στα πειραματικά δεδομένα ελαχιστοποιώντας όσο το δυνατόν περισσότερο την απόστασή της από αυτά.

Αυτή η μέθοδος είναι προσεγγιστική καθώς είναι πολύ δύσκολο να παρατηρηθούν γραμμικές σχέσεις μεταξύ μεταβλητών στη φύση. Επιπλέον, μπορούν να δημιουργηθούν πειραματικά σφάλματα κατά την ανάκτηση των μετρήσεων. Αυτά μπορεί να οφείλονται σε διάφορους λόγους όπως για παράδειγμα ο θόρυβος, η αβεβαιότητα του οργάνου, διάφορες εξωτερικές διαταραχές, το ανθρώπινο σφάλμα, το σφάλμα βαθμονόμησης οργάνου, το σφάλμα μηδενισμού, το σφάλμα ανάγνωσης και το σφάλμα στρογγυλοποίησης. Όσο πιο πολύπλοκη γίνεται μια συνάρτηση, τόσο πιο δύσκολο είναι να υπολογιστεί με ακρίβεια η σωστή συνάρτηση εξόδου.

Για τον υπολογισμό της ευθείας γίνεται χρήση του τετραγώνου της απόστασης καθώς αυτό είναι διαφορίσιμο αλλά και πιο εύκολο στο χειρισμό. Έστω ότι πρέπει να προσεγγιστεί γραμμικό σύστημα μιας εισόδου και εξόδου. Αυτό παρουσιάζει μαθηματικοποιημένο μοντέλο που προϋποθέτει την εύρεση μεταβλητών βάσει συγκεκριμένων εξισώσεων.

Ως  $P$  ορίζεται το πλήθος των δεδομένων ενώ ως  $\varepsilon$  το σφάλμα, δηλαδή η ελάχιστη απόσταση των πειραματικών δεδομένων από την βέλτιστη ευθεία. Πιο συγκεκριμένα, ο υπολογισμός της βέλτιστης ευθείας υπολογίζεται ως εξής:

$$g(a, b) = \varepsilon_1^2 + \varepsilon_2^2 + \dots + \varepsilon_P^2 = \sum_{j=1}^P \varepsilon_j^2 = \sum_{j=1}^P (y_j - \hat{y}_j)^2 \quad (3.1)$$

$$\hat{y}_j = ax_j + b \quad (3.2)$$

Προκειμένου να βρεθεί το ελάχιστο μιας συνάρτησης, πρέπει η παράγωγός της να είναι ίση με το μηδέν. Στη συγκεκριμένη περίπτωση, το ελάχιστο προκύπτει όταν μηδενίζονται ταυτόχρονα οι δυο μερικές παράγωγοι της συνάρτησης ως προς τις παραμέτρους  $a$  και  $b$  όπως φαίνεται στην εξίσωση (3.3).

$$\frac{\partial g}{\partial a} = 0 \quad \text{KAI} \quad \frac{\partial g}{\partial b} = 0 \quad (3.3)$$

Με χρήση κατάλληλων υπολογισμών, προκύπτει ένα σύστημα δυο εξισώσεων και δυο αγνώστων. Η λύση αυτού του συστήματος παρέχει τις κατάλληλες τιμές των συντελεστών. Αναλυτικότερα:

$$a = \frac{SS_{xy}}{SS_{xx}} \quad (3.4)$$

$$SS_{xy} = \sum_{j=1}^P x_j y_j - \frac{\sum_{j=1}^P x_j \sum_{j=1}^P y_j}{P} \quad (3.5)$$

$$SS_{xx} = \sum_{j=1}^P (x_j)^2 - \frac{(\sum_{j=1}^P x_j)^2}{P} \quad (3.6)$$

$$b = \bar{y} - a\bar{x} \quad (3.7)$$

$$\bar{y} = \frac{\sum_{j=1}^P y_j}{P} \quad (3.8)$$

$$\bar{x} = \frac{\sum_{j=1}^P x_j}{P} \quad (3.9)$$

Μετά των υπολογισμό των συντελεστών των εξισώσεων (3.4) και (3.7), αυτοί μπορούν να χρησιμοποιηθούν για την εύρεση της βέλτιστης ευθείας στην εξίσωση (3.2). Σε πολλές περιπτώσεις η συνάρτηση περιέχει παραπάνω από μια ανεξάρτητες μεταβλητές με αποτέλεσμα

να περιγράψει ένα γραμμικό σύστημα  $N$  εισόδων και μιας εξόδου. Ο αυξημένος αριθμός μεταβλητών δυσχεραίνει τους υπολογισμούς των εξισώσεων. Ένας τρόπος αντιμετώπισης αυτού του προβλήματος είναι η υλοποίηση του αλγορίθμου των ελαχίστων τετραγώνων σε μορφή πινάκων. Ο πρώτος πίνακας που δημιουργείται είναι αυτός που περιγράφει τα δεδομένα εισόδου.

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1N} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{P1} & x_{P2} & \dots & x_{PN} & 1 \end{bmatrix} \quad (3.10)$$

Ο αριθμός των στηλών του πίνακα αντιπροσωπεύει τις μεταβλητές εισόδου  $N$  του συστήματος ενώ ο αριθμός των σειρών, το πλήθος των δεδομένων  $P$ . Σε περίπτωση που πρέπει να υπάρξει και ένας σταθερός όρος που περιγράφει τη πόλωση, προστίθεται ακόμα μια στήλη στο τέλος του πίνακα η οποία αποτελείται από άσσους. Στη συνέχεια, πρέπει να φτιαχτεί ένας δεύτερος πίνακας που να περιλαμβάνει τις πραγματικές τιμές εξόδου της συνάρτησης.

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_P \end{bmatrix} \quad (3.11)$$

Ο τρίτος και τελευταίος πίνακας είναι αυτός που περιλαμβάνει τις απαραίτητες παραμέτρους επίλυσης του συστήματος.

$$A = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_P \\ b \end{bmatrix} \quad (3.12)$$

Όταν το σύστημα περιλαμβάνει παραπάνω από μια εισόδους, η συνάρτηση που το περιγράφει έχει τη μορφή:

$$\hat{y} = \sum_{j=1}^P \alpha_j x_j + b \quad (3.13)$$

Είναι λογικό να διαπιστωθεί πως το πηλίκο των πινάκων  $X$  και  $A$ , περιγράφει έναν πίνακα  $\hat{Y}$  που περιλαμβάνει τις προβλέψεις του συστήματος για κάθε δεδομένο. Αυτός ο πίνακας περιγράφεται στην εξίσωση (3.14).

$$X \cdot A = \begin{bmatrix} x_{11}\alpha_1 & x_{12}\alpha_2 & \dots & x_{1N}\alpha_P & b \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{P1}\alpha_1 & x_{P2}\alpha_2 & \dots & x_{PN}\alpha_P & b \end{bmatrix} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_P \end{bmatrix} = \hat{Y} \quad (3.14)$$

Στόχος πρέπει να είναι οι προβλέψεις του συστήματος  $\hat{Y}$  να ταυτίζονται όσο το δυνατόν περισσότερο με τις πραγματικές τιμές  $Y$ . Για να πραγματοποιηθεί αυτό πρέπει να βρεθεί πίνακας παραμέτρων  $A$  ικανός να επιλύσει την εξίσωση (3.15):



$$Y = X \cdot A \quad (3.15)$$

Ύστερα από κάποιες απαραίτητες διαδικασίες προκειμένου να απομονωθεί ο πίνακας  $A$ , η εξίσωση (3.15) παίρνει τη μορφή:

$$A = (X^T \cdot X)^{-1} \cdot X^T \cdot Y \quad (3.16)$$

Ο πίνακας  $X^T \cdot X$  ονομάζεται πίνακας συνδιασποράς και παρέχει σημαντικές πληροφορίες για τον τρόπο που είναι κατανομημένες οι εισόδου του συστήματος. Είναι πολύ σημαντικό να είναι αντιστρέψιμος καθώς σε περίπτωση που δεν είναι, δεν υπάρχει μοναδικός τρόπος για να υπολογιστούν οι βέλτιστες τιμές των παραμέτρων του πίνακα  $A$ .

Σε προβλήματα πρόβλεψης μελλοντικών τιμών χρονοσειρών, ο αριθμός μεταβλητών εισόδου  $N$ , θα αντιπροσωπεύει μια επιλεγμένη αλληλουχία παρελθοντικών τιμών.

---

#### **Αλγόριθμος 1:** Μέθοδος Ελαχίστων Τετραγώνων με χρήση πινάκων

---

**Είσοδοι:**      Αριθμός μεταβλητών εισόδου  $N$   
                  Πλήθος δεδομένων προς πρόβλεψη  $P$

**Έξοδος:**      Προβλέψεις  $\hat{Y}$

1: Δημιουργία πίνακα  $X$  σε μορφή της εξίσωσης (3.10)

2: Δημιουργία πίνακα  $Y$  σε μορφή της εξίσωσης (3.11)

3: Πίνακας  $A \leftarrow$  Υπολογισμός μέσω της εξίσωσης (3.16)

4: Προβλέψεις  $\hat{Y} \leftarrow$  Υπολογισμός μέσω της εξίσωσης (3.14)

**ΤΕΛΟΣ ΑΛΓΟΡΙΘΜΟΥ**

---

### **3.8 Το Custom μοντέλο RNN**

Για το custom μοντέλο κατασκευάστηκαν οι απαραίτητες συναρτήσεις κώδικα που παρέχουν τη δομή και τη λειτουργία ενός απλού RNN δικτύου. Χρησιμοποιήθηκε ο αλγόριθμος της οπισθοδιάδοσης στο χρόνο (BPTT) για την εκμάθηση των παραμέτρων ενώ πραγματοποιήθηκαν οι παρακάτω σχεδιαστικές επιλογές:

- Η συνάρτηση απώλειας για την αξιολόγηση της απόδοσης του μοντέλου είναι το μέσο τετραγωνικό σφάλμα (MSE). Η εξίσωση (2.5) παίρνει τη μορφή:

$$MSE(\hat{y}, y) = \sum_{t=1}^T (y_t - \hat{y}_t)^2 \quad (3.17)$$

- Κατά τη διάρκεια των υπολογισμών των κλίσεων με χρήση του κανόνα αλυσίδας, απαιτείται η παράγωγος της συνάρτησης απώλειας σε σχέση με τις προβλεπόμενες τιμές. Αυτό σημαίνει πως πραγματοποιείται αλλαγή πρόσημων ως προς τις πραγματικές και προβλεπόμενες τιμές. Επίσης χρησιμοποιείται η σύμβαση όπου η κλίση ορίζεται χωρίς τον συντελεστή 2 μπροστά. Αυτό απλοποιεί τον υπολογισμό χωρίς να αλλάξει το αποτέλεσμα της βελτιστοποίησης. Έτσι διασφαλίζεται η συνέπεια με άλλες συναρτήσεις απώλειας και τεχνικές βελτιστοποίησης. Η εξίσωση είναι ίση με:

$$\frac{\partial \ell_t}{\partial \hat{y}_t} = (\hat{y}_t - y_t) \quad (3.18)$$

- Το μοντέλο αποτελείται από συνάρτηση ενεργοποίησης υπερβολικής εφαπτομένης στο κρυφό επίπεδο ενώ δεν περιλαμβάνει συνάρτηση ενεργοποίησης στο επίπεδο εξόδου. Η παράγωγος της υπερβολικής εφαπτομένης ορίζεται ως:

$$\frac{\partial \tanh(x)}{\partial x} = 1 - \tanh(x)^2 \quad (3.19)$$

Ενώ μέσω της εξίσωσης (3.19), η παράγωγος ως προς το κρυφό επίπεδο είναι ίση με:

$$\frac{\partial h_i}{\partial p_i} = 1 - \tanh(p_i)^2 \quad (3.20)$$

- Οι αρχικοποιήσεις στους πίνακες βάρους και στα διανύσματα πολώσεως πραγματοποιούνται μέσω της μεθόδου Xavier [53] για ένα ορισμένο διάστημα κατανομής. Με αυτόν τον τρόπο μπορεί να διατηρείται η διακύμανση ανά το πέρασμα του κάθε επιπέδου σε ορισμένα όρια έτσι ώστε να είναι δυνατή η πλήρης αξιοποίηση των λειτουργιών της συνάρτησης ενεργοποίησης. Ο τύπος που χρησιμοποιήθηκε είναι ο εξής:

$$x = \frac{1}{\sqrt{\text{Αριθμός μονάδων κρυφού επιπέδου}}} \text{ για } x \in [-1,1] \quad (3.21)$$

- Χρησιμοποιείται stochastic gradient descent και έτσι εκτελείται ενημέρωση παραμέτρων για κάθε δεδομένο εκπαίδευσης.
- Η είσοδος αποτελείται από μια ακολουθία μήκους κάποιων χρονικών βημάτων μιας χρονοσειράς. Επομένως, περιγράφει ένα δεδομένο εισόδου το οποίο θα αναλύεται για T χρονικές στιγμές.

- Περιέχει μοναδικό κρυφό επίπεδο απλού αναδρομικού δικτύου.
- Αποτελείται από μια μεταβλητή εξόδου που περιέχει τιμές για όλες τις χρονικές στιγμές αλλά μόνο η τελευταία είναι απαραίτητη και αποτελεί τη πρόβλεψη του μοντέλου.

Έχοντας ως βάση αυτές τις σχεδιαστικές επιλογές, μπορεί να πραγματοποιηθεί η κατάλληλη μετατροπή των εξισώσεων του αλγορίθμου οπισθοδιάδοσης στο χρόνο του κεφαλαίου 2 της διπλωματικής. Πιο συγκεκριμένα για το forward pass και τον υπολογισμό των λειτουργικών σημάτων του συστήματος:

- Η εξίσωση (2.1) έχει τη μορφή:

$$h_t = \tanh(W_{IH} \cdot x_t + W_{HH} \cdot h_{t-1} + b_h) \quad (3.22)$$

- Ενώ η εξίσωση (2.3) έχει τη μορφή:

$$y_t = W_{HO} \cdot h_t + b_o \quad (3.23)$$

Αντίστοιχα για το backwards pass και τον υπολογισμό των κλίσεων στους απαραίτητους πίνακες βάρους και πολώσεων:

- Η μερική παράγωγος της εξίσωσης (2.6) υπολογίζεται ως εξής:

$$\frac{\partial \mathcal{L}}{\partial W_{HO}} = \sum_{t=1}^T (\hat{y}_t - y_t) \cdot h_t \quad (3.24)$$

- Η μερική παράγωγος της εξίσωσης (2.15) υπολογίζεται ως εξής:

$$\frac{\partial \mathcal{L}}{\partial b_y} = \sum_{t=1}^T (\hat{y}_t - y_t) \quad (3.25)$$

- Η μερική παράγωγος της εξίσωσης (2.13) υπολογίζεται ως εξής:

$$\frac{\partial \mathcal{L}}{\partial W_{HH}} = \sum_{t=1}^T (\hat{y}_t - y_t) \cdot W_{HO} \cdot \prod_{r=1}^t \prod_{i=r+1}^t [(1 - \tanh(p_i)^2) \cdot W_{HH}^T] \cdot h_{r-1} \quad (3.26)$$

- Η μερική παράγωγος της εξίσωσης (2.16) υπολογίζεται ως εξής:

$$\frac{\partial \mathcal{L}}{\partial b_h} = \sum_{t=1}^T (\hat{y}_t - y_t) \cdot W_{HO} \cdot \prod_{r=1}^t \prod_{i=r+1}^t [(1 - \tanh(p_i)^2) \cdot W_{HH}^T] \quad (3.27)$$

- Η μερική παράγωγος της εξίσωσης (2.14) υπολογίζεται ως εξής:

$$\frac{\partial \mathcal{L}}{\partial W_{IH}} = \sum_{t=1}^T (\hat{y}_t - y_t) \cdot W_{HO} \cdot \prod_{r=1}^t \prod_{i=r+1}^t [(1 - \tanh(p_i)^2) \cdot W_{HH}^T] \cdot x_r \quad (3.28)$$

Στη συνέχεια περιέχονται κάποιοι αλγόριθμοι που περιγράφουν σε μια απλοποιημένη μορφή τη σειρά των διαδικασιών που εκτελέστηκαν. Η μεγαλύτερη βάση έχει δοθεί στην εκπαίδευση του μοντέλου.

---

### Αλγόριθμος 2: Πρόσθια τροφοδότηση RNN (Forward Pass)

---

**Είσοδοι:** Δεδομένο εισόδου  $x$  που θα αναλυθεί για  $T$  χρονικές στιγμές

Πίνακες βάρους  $W_{IH}$   $W_{HH}$   $W_{HO}$

Διανύσματα πόλωσης  $b_h$   $b_o$

**Έξοδοι:** Πίνακας τιμών κρυφών μονάδων  $h$

Πίνακας τιμών εξόδων  $y$

- 1: Μηδενισμός πινάκων τιμών κρυφών μονάδων και εξόδων
- 2: **ΓΙΑ** τιμή μέσα στο εύρος του μήκους του δεδομένου εισόδου:
- 3: Εύρεση κρυφής κατάστασης χρονικής στιγμής μέσω της εξίσωσης (3.22)
- 4: Κατάλληλη αποθήκευση σε πίνακα τιμών κρυφών μονάδων
- 5: Εύρεση εξόδου χρονικής στιγμής μέσω της εξίσωσης (3.23)
- 6: Κατάλληλη αποθήκευση σε πίνακα τιμών εξόδου

7: **ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ ΑΛΓΟΡΙΘΜΟΥ**

---

### Αλγόριθμος 3: Τροφοδότηση RNN από εμπρός προς τα πίσω (Backwards Pass)

---

**Είσοδοι:** Δεδομένο εισόδου  $x$  που θα αναλυθεί για  $T$  χρονικές στιγμές

Πίνακες βάρους  $W_{IH}$   $W_{HH}$   $W_{HO}$

Διανύσματα πόλωσης  $b_h$   $b_o$

Πίνακας τιμών κρυφών μονάδων  $h$  για το δεδομένο εισόδου

Κλίση συνάρτησης απώλειας  $\mathcal{L}$

Ρυθμός μάθησης  $lr$

**Έξοδοι:** Νέες τιμές πινάκων βαρών και πολώσεων σύμφωνα με τον κανόνα Widrow - Hoff

- 1: Δημιουργία και αρχικοποίηση πινάκων κλίσης
- 2: **ΓΙΑ** τιμή μέσα στο εύρος του μήκους του δεδομένου εισόδου ξεκινώντας από το τέλος:
- 3: Υπολογισμός κλίσης μέσω της εξίσωσης (3.24)
- 4: Υπολογισμός κλίσης μέσω της εξίσωσης (3.25)
- 5: Διάδοση της κλίσης προς τη κρυφή μονάδα
- 6: **ΣΥΝΘΗΚΗ** που ενεργοποιείται όταν δεν αναλύεται η τελευταία κρυφή μονάδα:
- 7: Κατάλληλες πράξεις έτσι ώστε να προστεθεί η κλίση από την επόμενη κρυφή μονάδα
- 8: **ΤΕΛΟΣ ΣΥΝΘΗΚΗΣ**
- 9: Υπολογισμός παραγώγου tanh τιμής κρυφής μονάδας μέσω της εξίσωσης (3.20)
- 10: Πολλαπλασιασμός παραγώγου με κλίση κρυφής μονάδας για μη γραμμικότητα
- 11: Αποθήκευση για χρήση σε υπολογισμούς επόμενων χρονικών στιγμών
- 12: **ΣΥΝΘΗΚΗ** που ενεργοποιείται όταν δεν αναλύεται η πρώτη κρυφή μονάδα:
- 13: Υπολογισμός κλίσης μέσω της εξίσωσης (3.26)
- 14: Υπολογισμός κλίσης μέσω της εξίσωσης (3.27)
- 15: **ΤΕΛΟΣ ΣΥΝΘΗΚΗΣ**
- 16: Υπολογισμός κλίσης μέσω της εξίσωσης (3.28)
- 17: **ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ**
- 18: Υπολογισμός νέων τιμών πινάκων βάρους και πολώσεων μέσω του κανόνα Widrow – Hoff από την εξίσωση (2.17)

#### **ΤΕΛΟΣ ΑΛΓΟΡΙΘΜΟΥ**

---

#### **Αλγόριθμος 4: Custom Μοντέλο RNN**

---

**Είσοδοι:**       Σύνολο δεδομένων εισόδου  
                  Αριθμός Epoch

**Έξοδος:**        Προβλέψεις  $\hat{Y}$  του μοντέλου

- 1: Απαραίτητες διαδικασίες προ-επεξεργασίας των δεδομένων
- 2: Απαραίτητες αρχικοποιήσεις πινάκων βάρους και πολώσεων
- 3: **ΓΙΑ** τιμή μέσα στο εύρος των Epoch:
- 4: **ΓΙΑ** τιμή μέσα στο εύρος του συνόλου δεδομένων εκπαίδευσης:
- 5: Απομόνωση δεδομένου εισόδου που θα αναλυθεί για  $T$  χρονικές στιγμές
- 6: Απομόνωση τιμής εξόδου προς πρόβλεψη
- 7: **ΚΑΛΕΣΜΑ ΑΛΓΟΡΙΘΜΟΥ 2**
- 8: Εύρεση κλίσης μέσω συνάρτησης απώλειας από εξίσωση (3.18)
- 9: **ΚΑΛΕΣΜΑ ΑΛΓΟΡΙΘΜΟΥ 3**
- 10: Υπολογισμός απώλειας κλίσης δεδομένων εκπαίδευσης από εξίσωση (3.17)
- 11: Υπολογισμός απώλειας κλίσης δεδομένων αξιολόγησης μέσω απαραίτητων διαδικασιών

12: ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ

13: ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ

14: Υπολογισμός προβλέψεων  $\hat{Y}$  μέσω απαραίτητων διαδικασιών και **ΑΛΓΟΡΙΘΜΟΥ 2**

15: Απαραίτητες διαδικασίες υπολογισμού δεικτών απόδοσης

16: Απαραίτητες διαδικασίες απεικόνισης των δεδομένων

**ΤΕΛΟΣ ΑΛΓΟΡΙΘΜΟΥ**

---

### 3.9 Κατασκευή μοντέλων με χρήση του Keras

Όπως έχει αναλυθεί στο πρώτο κεφάλαιο της διπλωματικής, τα νευρωνικά δίκτυα περιγράφονται από ένα σημαντικό μεγάλο αριθμό παραγόντων που μπορούν να επηρεάσουν τη διαδικασία μάθησης και την τελική απόδοση. Το Keras παρέχει δομικά στοιχεία υψηλού επιπέδου τα οποία που μπορούν να παραμετροποιηθούν προκειμένου να επιλεγθούν τα οι κατάλληλες μέθοδοι και εργαλεία για κάθε περίπτωση. Επιπλέον, το μοντέλο μπορεί να δημιουργηθεί με απλό και κατανοητό τρόπο μόνο με λίγες γραμμές κώδικα.

#### 3.9.1 Το μοντέλο MLP

Το μοντέλο περιλαμβάνει πλήρως συνδεδεμένα (Dense) επίπεδα και χρησιμοποιεί τη συνάρτηση `tf.keras.layers.Dense`. Μεγάλη σημασία πρέπει να δοθεί στη μορφή που έχει η είσοδος και η έξοδος του συστήματος. Όταν πρέπει να πραγματοποιηθεί η πρόβλεψη μιας μελλοντικής τιμής μιας χρονοσειράς, η είσοδος του μοντέλου αποτελείται από έναν πίνακα δυο διαστάσεων [*timesteps*, *features*]. Ως *timesteps* αναφέρεται το μήκος της αλληλουχίας των προηγούμενων τιμών της χρονοσειράς που χρησιμοποιούνται ενώ ως *features* αναφέρονται τα διαφορετικά χαρακτηριστικά που μπορούν να περιγράψουν μια συγκεκριμένη χρονική στιγμή. Αντίστοιχες διαστάσεις έχει και η έξοδος του συστήματος αλλά σε αυτήν την περίπτωση ως *timesteps* αναφέρεται το μήκος της αλληλουχίας μελλοντικών τιμών της χρονοσειράς που προβλέπονται.

Η χρήση νευρωνικών δικτύων αυξάνει ριζικά και τον αριθμό των παραμέτρων που μπορούν να εκπαιδευτούν σε σύγκριση με το μοντέλο των ελαχίστων τετραγώνων. Συγκεκριμένα για ένα μοντέλο MLP που περιλαμβάνει  $n$  dense συνολικά επίπεδα, οι παράμετροι υπολογίζονται ως εξής:

$$\text{παραμέτροι} = \left[ \sum_{i=1}^n \text{Αριθμός εισόδων επιπέδου } i \cdot \text{Αριθμός εξόδων επιπέδου } i \right] + \text{αριθμός πολώσεων συνάψεων} \quad (3.29)$$

Η πολυπλοκότητα του μοντέλου αυξάνει ακόμη και τον αριθμό των απαραίτητων βημάτων που απαιτούνται για την εκπαίδευση αλλά και τον χρόνο εκτέλεσης. Συγκεκριμένα:

$$\text{Αριθμός βημάτων εκπαίδευσης} = \frac{\text{Αριθμός δειγμάτων εκπαίδευσης}}{\text{Μέγεθος παρτίδας}} \cdot \text{Epochs} \quad (3.30)$$

### 3.9.2 Το μοντέλο απλού RNN

Σε αυτήν την περίπτωση το μοντέλο χρησιμοποιεί ένα αναδρομικό κρυφό επίπεδο μέσω της συνάρτησης `tf.keras.layers.SimpleRNN`. Αυτό συνδέεται και πάλι με ένα `dense` επίπεδο εξόδου. Η είσοδος της συνάρτησης αποτελείται από έναν `tensor` τριών διαστάσεων με μορφή `[batch, timesteps, features]`. Τα `timesteps` και τα `features` κρατούν την ίδια σημασία με τα MLP, ενώ η διάσταση `batch` αναφέρεται σε έναν αριθμό παρτίδας δεδομένων εισόδου. Αντίστοιχα, η διάσταση της εξόδου μπορεί να περιγράφεται από δυο διαφορετικές εκδοχές και εξαρτάται από την `return_sequences` παράμετρό της. Αν αυτή είναι `TRUE`, περιγράφεται από έναν τριών διαστάσεων `tensor [batch, timesteps, features]` που περιλαμβάνει όλες τις τιμές εξόδου των κρυφών μονάδων. Αντιθέτως, αν η παράμετρος είναι `FALSE`, περιγράφεται από έναν πίνακα δυο διαστάσεων `[batch, features]` που περιλαμβάνει μόνο την έξοδο της τελευταίας κρυφής μονάδας.

Ο αριθμός των παραμέτρων του κρυφού επιπέδου που μπορούν να εκπαιδευτούν υπολογίζεται ως εξής:

$$\text{παραμέτροι} = (\text{αριθμός κρυφων μονάδων} + \text{αριθμός feature εισόδου}) \cdot \text{αριθμός κρυφων μονάδων} + \text{αριθμός κρυφων μονάδων} \quad (3.31)$$

### 3.9.3 Το μοντέλο LSTM και GRU

Οι είσοδοι και οι έξοδοι των επιπέδων τους διατηρούν τους ίδιους κανόνες με τα RNN ενώ οι συναρτήσεις που τα περιγράφουν είναι οι `tf.keras.layers.LSTM` και `tf.keras.layers.GRU`. Η διαφορά παρατηρείται στον αριθμό και στην οργάνωση των παραμέτρων που μπορούν να εκπαιδευτούν. Αποτελούνται από 3 διαφορετικούς τένσορες. Ο πρώτος είναι ο `Input Tensor` και περιλαμβάνει του πίνακες βάρους των διαδικασιών που συνδέονται με το επίπεδο εισόδου. Τα LSTM αποτελούνται από 4 τέτοιους πίνακες διάστασης `[ διάσταση εισόδου, αριθμός κρυφών μονάδων ]`. Ο `input tensor` όμως αποτελείται από την ένωση αυτών και έτσι έχει διάσταση `[ διάσταση εισόδου, αριθμός κρυφών μονάδων * 4 ]`. Επιπλέον υπάρχει συγκεκριμένη σειρά που ακολουθεί η ένωση.

*INPUT GATE → FORGET GATE → CELL STATE → OUTPUT GATE*

Με τον ίδιο τρόπο δημιουργείται και ο input tensor στα GRU αλλά τώρα αποτελείται από την ένωση 3 τέτοιων πινάκων και έτσι έχει διαστάσεις [ διάσταση εισόδου, αριθμός κρυφών μονάδων \* 3]. Η αντίστοιχη σειρά ένωσης είναι η παρακάτω:

*RESET GATE → UPDATE GATE → CANDIDATE HIDDEN STATE*

Ο δεύτερος είναι ο Recurrent Tensor και περιλαμβάνει του πίνακες βάρους των διαδικασιών που συνδέονται με το κρυφό επίπεδο. Ο αριθμός των πινάκων αλλά και η μεθοδολογία ένωσης παραμένει ίδια αλλά πλέον οι διαστάσεις των πινάκων είναι [αριθμός κρυφών μονάδων, αριθμός κρυφών μονάδων] ενώ ο τένσορας έχει διάσταση [αριθμός κρυφών μονάδων, αριθμός κρυφών μονάδων \* 4] για τα LSTM, ενώ έχει διάσταση [αριθμός κρυφών μονάδων, αριθμός κρυφών μονάδων \* 3] για τα GRU. Ο τελευταίος είναι ο Bias Tensor που θα ακολουθήσει την ίδια διαδικασία για τα διανύσματα πόλωσης του μοντέλου. Οι διαστάσεις τους είναι [μονάδες πόλωσης,] ενώ ο τελικός τένσορας έχει διαστάσεις [μονάδες πόλωσης \* 4,] για τα LSTM ενώ [μονάδες πόλωσης \* 3,] για τα GRU.



---

## ΚΕΦΑΛΑΙΟ 4<sup>ο</sup>: CASE STUDY: ΕΦΑΡΜΟΓΕΣ ΑΝΑΔΡΟΜΙΚΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΣΤΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΧΡΟΝΟΣΕΙΡΩΝ

---

Αυτό το κεφάλαιο περιγράφει το βασικό ερευνητικό κομμάτι της διπλωματικής. Πιο συγκεκριμένα, στα πλαίσια του κεφαλαίου περιγράφεται η εφαρμογή αλλά και η κατάλληλη ρύθμιση των αρχιτεκτονικών που αναλύθηκαν στο κεφάλαιο 3 για την πρόβλεψη μελλοντικών τιμών χρονοσειρών. Περιλαμβάνονται δυο σύνθετες και απαιτητικές εφαρμογές. Η πρώτη αφορά τα συνθετικά δεδομένα της χαοτικής συνάρτησης Mackey – Glass, ενώ η δεύτερη τα πραγματικά δεδομένα κατανάλωσης ηλεκτρικής ενέργειας ενός δημόσιου κτηρίου. Η αξιολόγηση των προτεινόμενων μοντέλων θα αναδείξει την απόδοση όλων των δικτύων καθώς και την ικανότητά τους να διαχειρίζονται ακολουθιακά ιστορικά δεδομένα αλλά και να αντιλαμβάνονται σύνθετες χρονικές εξαρτήσεις.

### 4.1 Χρονοσειρές

Μια χρονοσειρά [54][55] αποτελεί μια διατεταγμένη ακολουθία δεδομένων τα οποία καταγράφονται σε ίσα χρονικά διαστήματα και εκφράζουν την εξέλιξη των τιμών μια μεταβλητής στο χρόνο. Αν μια ορισμένη χρονική στιγμή μιας χρονοσειράς περιγράφεται από μοναδική μεταβλητή, τότε αυτή χαρακτηρίζεται ως Univariate. Αντιθέτως, αν ληφθούν υπόψιν μεταβλητές που περιγράφουν διαφορετικά χαρακτηριστικά για μια ορισμένη χρονική στιγμή, τότε η χρονοσειρά χαρακτηρίζεται ως Multivariate.

Η ανάλυση των χρονοσειρών μπορεί να χωριστεί σε δυο βασικά μέρη. Το πρώτο μέρος έχει να κάνει με την κατανόηση και λήψη της δομής και της ιστορικής συμπεριφοράς των παρατηρούμενων δεδομένων. Το δεύτερο μέρος αφορά την προσαρμογή ενός μοντέλου για την πραγματοποίηση μελλοντικών προβλέψεων. Αν η πρόβλεψη αφορά μόνο μία επόμενη τιμή, τότε ονομάζεται μοναδικού βήματος πρόβλεψη (Single-Step Prediction). Αντιθέτως, αν αφορά μια αλληλουχία επόμενων τιμών, τότε ονομάζεται πρόβλεψη πολλών βημάτων (Multi-Step Prediction). Η ανάλυση χρονοσειρών αποτελεί ένα βασικό εργαλείο που βρίσκει εφαρμογή σε πληθώρα εφαρμογών που περιγράφουν γνωστούς τομείς έρευνας. Για παράδειγμα μπορεί να εφαρμοστεί στη λήψη αποφάσεων εταιρειών που βασίζονται σε οικονομικά μεγέθη, στην αναγνώριση παθημάτων μέσω της ηλεκτρικής δραστηριότητας σωματικών οργάνων αλλά και στη πρόβλεψη φυσικών φαινομένων όπως η θερμοκρασία ή η σεισμική δόνηση.

Μια χρονοσειρά μπορεί επίσης να είναι συνεχής ή διακριτή. Σε μια συνεχή χρονοσειρά, οι παρατηρήσεις καταγράφονται σε κάθε χρονική στιγμή, ενώ μια διακριτή χρονοσειρά περιέχει παρατηρήσεις που καταγράφονται σε διακριτά χρονικά σημεία τα οποία συνήθως

παρεμβάλλονται από ίσα χρονικά διαστήματα. Για παράδειγμα, η μέτρηση της θερμοκρασίας ή η μέτρηση της συγκέντρωσης ενός διαλύματος μέσω ενός κατάλληλου αισθητήρα μπορούν να θεωρηθούν ως συνεχείς χρονοσειρές. Αντίστοιχα ο ετήσιος πληθυσμός μιας χώρας ή η ημερήσια παραγωγή μιας εταιρείας μπορούν να αντιπροσωπεύουν διακριτές χρονοσειρές. Η μεταβλητή που παρατηρείται σε μια διακριτή χρονική σειρά θεωρείται ότι καταγράφεται ως συνεχής μεταβλητή χρησιμοποιώντας την κλίμακα των πραγματικών αριθμών. Επιπλέον, μια συνεχής χρονοσειρά μπορεί εύκολα να μετατραπεί σε μια διακριτή μέσω της συγχώνευσης των δεδομένων σε ένα καθορισμένο χρονικό διάστημα.

## 4.2 Εφαρμογή 1<sup>η</sup> : Πρόβλεψη τιμών συνάρτησης Mackey – Glass

### 4.2.1. Περιγραφή

Η συνάρτηση Mackey - Glass, που εισάχθηκε από τους Michael Mackey και Leon Glass το 1977, αποτελεί μια μαθηματική διαφορική εξίσωση που χρησιμοποιείται για τη μοντελοποίηση δυναμικών συστημάτων. Αποτελεί θεμελιώδες πλαίσιο για τη διερεύνηση και την κατανόηση πολύπλοκων δυναμικών, ιδιαίτερα του χάους, σε συστήματα με χρονικές καθυστερήσεις. Αρχικά χρησιμοποιήθηκε σε ιατρικές έρευνες αφού παρουσίαζε την ικανότητα να μιμείται τόσο την υγιή όσο και την παθολογική συμπεριφορά ορισμένων βιολογικών σεναρίων. Πιο συγκεκριμένα, η πρώτη της εφαρμογή είχε να κάνει με τη μοντελοποίηση της διακύμανσης στη σχετική ποσότητα ώριμων κυττάρων του αίματος. Τα τελευταία χρόνια, η χρονοσειρά αυτή έχει χρησιμοποιηθεί εκτενώς για την αξιολόγηση μοντέλων νευρωνικών δικτύων [56]. Αυτό οφείλεται στο ότι περιγράφεται από μια γνωστή εξίσωση χωρίς να επηρεάζεται από στοχαστικούς παράγοντες, προσφέροντας ένα ελεγχόμενο περιβάλλον για την αξιολόγηση. Στην παρούσα μελέτη, η εξίσωση χρησιμοποιείται ως μια διαδικασία επαλήθευσης, καθώς επιτρέπει την παρακολούθηση της συμπεριφοράς των μοντέλων που αναλύθηκαν και δημιουργήθηκαν στο τρίτο κεφάλαιο της διπλωματικής, έτσι ώστε να επιβεβαιωθεί ότι ακολουθούν τα αναμενόμενα πρότυπα. Πραγματοποιήθηκε η κατάλληλη επεξεργασία των δεδομένων και επιχειρήθηκε η παραμετροποίηση των μοντέλων με στόχο την όσο το δυνατόν καλύτερη πρόβλεψη μελλοντικών τιμών της.

### 4.2.2 Δεδομένα

Το σύνολο των δεδομένων (Dataset) της εφαρμογής θα κατασκευαστεί σύμφωνα με την παρακάτω εξίσωση διαφορών:

$$y(t+1) = (1-b)y(t) + a \frac{y(t-\tau)}{(1+y^e(t-\tau))} \quad (4.3)$$

Η παράμετρος  $\tau$  θα είναι ίση με το 30 και θα καθορίζει πόσο πίσω στο χρόνο θα εξαρτάται το σύστημα από τις προηγούμενες τιμές του. Το  $b$  θα ισούται με 0.1 και θα ελέγχει την εκθετική αποσύνθεση του συστήματος. Το  $a$  θα είναι ίσο με 0.2 και αποτελεί μια παράμετρο που είναι ικανή να κλιμακώνει την επίδραση της καθυστερημένης τιμής ενώ το  $e$  αποτελεί έναν εκθέτη που ρυθμίζει τη μη γραμμικότητα του συστήματος και θα είναι ίσο με 10. Έχοντας ως δεδομένα αυτές τις τιμές των παραμέτρων, η εξίσωση (4.3) θα πάρει τη μορφή:

$$y(t+1) = 0.9y(t) + 0.2 \frac{y(t-30)}{(1 + y^{10}(t-30))} \quad (4.4)$$

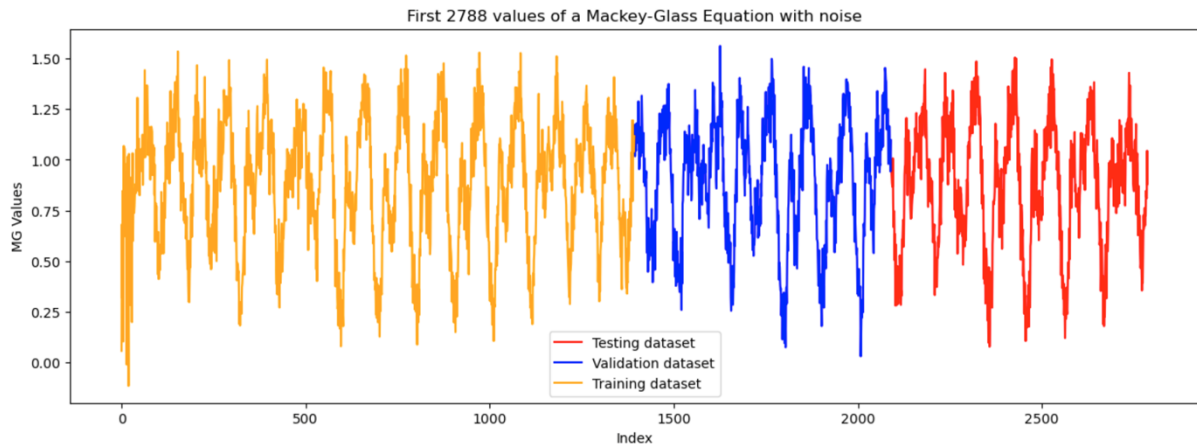
Γίνεται εμφανές πως για να παραχθούν δεδομένα από τη συγκεκριμένη εξίσωση διαφορών, θα πρέπει να είναι ήδη διαθέσιμες οι πρώτες 30 τιμές της συνάρτησης. Χρειάζεται επίσης να περιέχονται στο διάστημα τιμών (0,1). Χρησιμοποιήθηκε η ομοιόμορφη (Uniform) κατανομή για την κατασκευή τους. Με αυτόν τον τρόπο οποιαδήποτε τιμή μέσα στο καθορισμένο εύρος είχε την ίδια πιθανότητα εμφάνισης και δεν προστέθηκε κάποιο bias.

Επιλέχθηκε να δημιουργηθούν και να αναλυθούν οι πρώτες 2788 τιμές της συνάρτησης Mackey – Glass. Στη συνέχεια τους προστέθηκε θόρυβος (*Gaussian Noise*  $\sim N(0,0.1)$ ) έτσι ώστε να συμπεριληφθεί ένα κομμάτι στοχαστικότητας στην εφαρμογή.

Το σετ δεδομένων που παράχθηκε μέσω της εξίσωσης (4.4) χωρίστηκε σε τρία επιμέρους σύνολα δεδομένων με αναλογία (50% - 25% - 25%). Συγκεκριμένα αυτά είναι:

- Σύνολο δεδομένων εκπαίδευσης (Training dataset): Αποτελείται από τα πρώτα 1394 δεδομένα. Μέσω αυτών θα πραγματοποιηθεί η εκπαίδευση των μοντέλων.
- Σύνολο δεδομένων αξιολόγησης (Validation dataset): Αποτελείται από τα επόμενα 697 δεδομένα. Μέσω αυτών θα πραγματοποιηθεί η επιλογή του καλύτερου μοντέλου βάσει κάποιων συνθηκών.
- Σύνολο δεδομένων ελέγχου (Testing dataset): Αποτελείται από τα τελευταία 697 δεδομένα. Αποτελεί ένα τελείως ανεξάρτητο σύνολο μέσω του οποίου θα πραγματοποιηθεί ο έλεγχος της απόδοσης του επιλεγμένου μοντέλου.

Στο σχήμα 20 παρουσιάζεται το σύνολο των δεδομένων καθώς και ο διαχωρισμός τους σε υποσύνολα. Η μόνη εξαίρεση παρουσιάζεται στο γραμμικό μοντέλο των ελαχίστων τετραγώνων όπου τα δεδομένα χωρίζονται σε δυο υποσύνολα με αναλογία (50% - 50%). Τα πρώτα 1394 δεδομένα αποτελούν το Training set ενώ τα υπόλοιπα αποτελούν το Testing set.



**Σχήμα 20:** Αναπαράσταση των δεδομένων της 1<sup>ης</sup> εφαρμογής

Πρέπει ακόμη να πραγματοποιηθεί κανονικοποίηση των δεδομένων με κάποια από τις υπάρχουσες τεχνικές. Αυτό δεν θα αλλάξει τις σχετικές αποστάσεις μεταξύ των δεδομένων, αλλά θα βοηθήσει στον έλεγχο της ευαισθησίας των διάφορων αλγορίθμων που χρησιμοποιούνται κατά την εκπαίδευση και επιλογή του μοντέλου. Στη συγκεκριμένη μελέτη επιλέχθηκε να γίνει κανονικοποίηση μέσω κλιμάκωσης ελαχίστου – μεγίστου (Min – Max scaling) για ένα διάστημα  $[-1, 1]$ . Σε αυτήν την περίπτωση δίνεται μεγάλη βαρύτητα στη μέγιστη  $x_{max}$  και ελάχιστη  $x_{min}$  τιμή του dataset. Η εξίσωση υπολογισμού των σταθμισμένων τιμών  $x_{scaled}$  είναι η εξής:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (4.5)$$

### 4.2.3 Υλοποίηση

Για την υλοποίηση της εφαρμογής θα πραγματοποιηθεί χρήση των προγραμματιστικών εργαλείων και μοντέλων δικτύων που αναφέρθηκαν στο τρίτο κεφάλαιο της διπλωματικής. Τα δεδομένα είναι οργανωμένα και περιγράφουν μια Univariate χρονοσειρά. Η μεταβλητή της χρονοσειράς αποτελεί την τιμή εξόδου της συνάρτησης Mackey – Glass για ένα διακριτό εύρος βημάτων. Στόχος όλων των δοκιμών θα είναι να πραγματοποιηθεί Multi-Step Prediction για τις επόμενες έξι τιμές της χρονοσειράς κάνοντας χρήση ενός αριθμού ιστορικών δεδομένων.

Σε πρώτο στάδιο, κατασκευάστηκαν κάποια μοντέλα με στόχο τη σύγκριση της απόδοσής τους σε σχέση με τα αναδρομικά νευρωνικά δίκτυα. Το πρώτο από αυτά ήταν το γραμμικό μοντέλο των ελαχίστων τετραγώνων. Επιλέχθηκαν 6 ιστορικά δεδομένα ενώ χρησιμοποιήθηκε κώδικας βασισμένος στον αλγόριθμο 1 και όχι κάποια από τις διαθέσιμες συναρτήσεις βιβλιοθηκών. Το δεύτερο αποτελεί ένα MLP δίκτυο κατασκευασμένο με το functional API του Keras.

Χρησιμοποιεί και αυτό 6 ιστορικά δεδομένα ενώ οι παράμετροί του ρυθμίστηκαν βάσει του πίνακα 7:

**Πίνακας 7:** Παράμετροι επιλεγμένου μοντέλου MLP της πρώτης εφαρμογής

ΠΑΡΑΜΕΤΡΟΣ	ΕΠΙΛΟΓΗ
Κρυφές στοιβάδες	2
Αριθμός νευρώνων	10 στη κάθε στοιβάδα
Συνάρτηση ενεργοποίησης	ReLU σε κάθε στοιβάδα
Συνάρτηση απώλειας	MSE
Optimizer	Mini Batch Gradient Descent
Ρυθμός μάθησης	$10^{-3}$
Batch	4
Epochs	100

Το δεύτερο στάδιο της υλοποίησης περιλαμβάνει τα αναδρομικά νευρωνικά δίκτυα. Αρχικά, κατασκευάστηκε ένα Custom RNN από την αρχή κάνοντας χρήση των αλγορίθμων 2, 3 και 4. Στη συνέχεια, πραγματοποιήθηκε κατάλληλη χρήση τεχνικών προγραμματισμού έτσι ώστε να κατασκευαστεί το ίδιο RNN χρησιμοποιώντας το Keras. Με αυτόν τον τρόπο δόθηκε η ικανότητα σύγκρισής τους αλλά και επαληθεύτηκε η σωστή λειτουργία του Custom μοντέλου. Στο τρίτο στάδιο πραγματοποιήθηκε η κατασκευή μοντέλων που αποτελούν πιο σύνθετες παραλλαγές των απλών αναδρομικών δικτύων. Αυτά έκαναν χρήση επιπέδων LSTM και GRU. Πραγματοποιήθηκε κάποια αλλαγή ορισμένων παραμέτρων προκειμένου να εξυπηρετήσουν αυτήν την αύξηση της πολυπλοκότητας. Στον πίνακα 8 παρουσιάζεται η ρύθμιση των παραμέτρων όλων των αναδρομικών μοντέλων:

**Πίνακας 8:** Παράμετροι αναδρομικών μοντέλων της πρώτης εφαρμογής

ΠΑΡΑΜΕΤΡΟΣ	CUSTOM RNN	KERAS RNN	KERAS LSTM	KERAS GRU
Αριθμός ιστορικών δεδομένων	2	2	6	6
Αρχικοποίηση	Custom Xavier	Custom Xavier	Custom Xavier	Custom Xavier
Κρυφές στοιβάδες	1	1	1	1
Αριθμός μονάδων	40	40	40	40
Συνάρτηση ενεργοποίησης	tanh	tanh	tanh	tanh
Συνάρτηση απώλειας	MSE	MSE	MSE	MSE
Optimizer	SGD	SGD	SGD	SGD
Ρυθμός μάθησης	$10^{-3}$	$10^{-3}$	$10^{-3}$	$10^{-3}$
Batch	1	1	1	1
Epochs	10	10	20	20

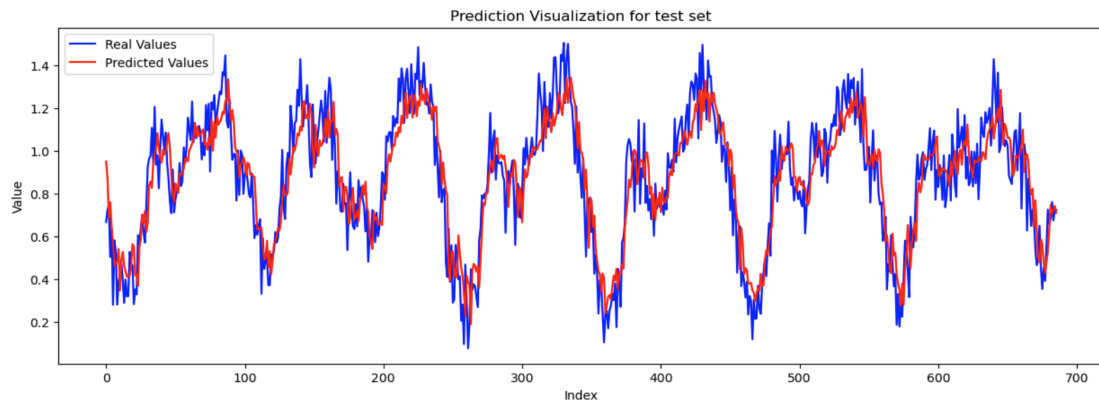
#### 4.2.4 Αποτελέσματα

Όλα τα αποτελέσματα περιγράφουν τον μέσο όρο 10 διαφορετικών επαναλήψεων. Έγινε χρήση συγκεκριμένων seed για την εισαγωγή της στοχαστικότητας έτσι ώστε να μπορούν να γίνουν οι κατάλληλες συγκρίσεις. Σε κάθε επανάληψη επιλέγονταν οι παράμετροι του Epoch που παρουσίαζε την ελάχιστη τιμή στο validation loss. Περιλαμβάνονται στον πίνακα 9 όπου πραγματοποιείται και η επιλογή του καλύτερου μοντέλου.

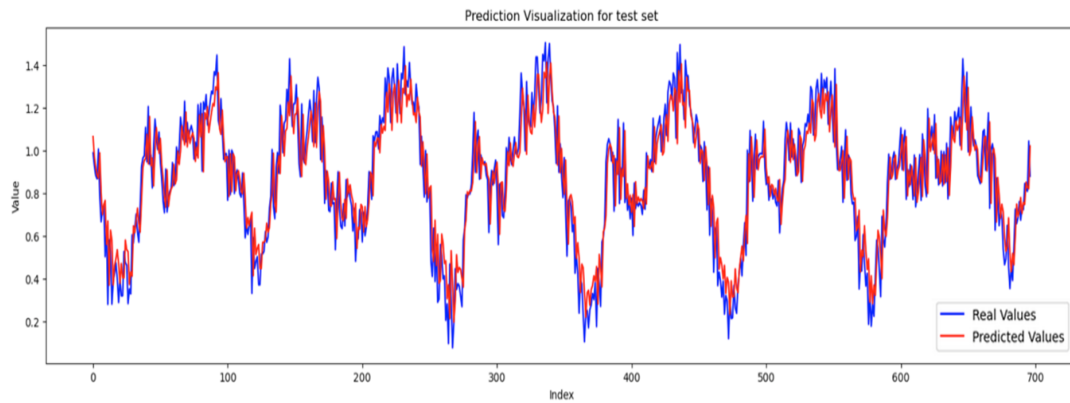
**Πίνακας 9:** Αποτελέσματα της πρώτης εφαρμογής

	LINEAR			KERAS MLP		
	MAE	RMSE	R2	MAE	RMSE	R2
1 STEP	0,1237	0,1538	0,7492	0,1109	0,1382	0,7949
2 STEPS	0,1350	0,1672	0,7039	0,1215	0,1513	0,7543
3 STEPS	0,1424	0,1773	0,6672	0,1284	0,1605	0,7237
4 STEPS	0,1520	0,1881	0,6263	0,1426	0,1770	0,6648
5 STEPS	0,1696	0,2066	0,5485	0,1585	0,1948	0,5944
6 STEPS	0,1801	0,2192	0,4901	0,1734	0,2127	0,5156
	CUSTOM RNN			KERAS RNN		
	MAE	RMSE	R2	MAE	RMSE	R2
1 STEP	0,1086	0,1366	0,8011	0,1239	0,1544	0,7536
2 STEPS	0,1191	0,1474	0,7686	0,1377	0,1703	0,7002
3 STEPS	0,1270	0,1585	0,7330	0,1446	0,1801	0,6623
4 STEPS	0,1407	0,1762	0,6706	0,1486	0,1850	0,6463
5 STEPS	0,1578	0,1943	0,5998	0,1689	0,2080	0,5536
6 STEPS	0,1699	0,2087	0,5385	0,1800	0,2204	0,4984
	KERAS LSTM			KERAS GRU		
	MAE	RMSE	R2	MAE	RMSE	R2
1 STEP	0,1135	0,1410	0,7906	0,1057	0,1325	0,8150
2 STEPS	0,1279	0,1578	0,7373	0,1191	0,1471	0,7719
3 STEPS	0,1416	0,1747	0,6784	0,1317	0,1632	0,7192
4 STEPS	0,1594	0,1943	0,6018	0,1474	0,1818	0,6515
5 STEPS	0,1743	0,2113	0,5285	0,1640	0,1999	0,5777
6 STEPS	0,1871	0,2261	0,4594	0,1776	0,2164	0,5052

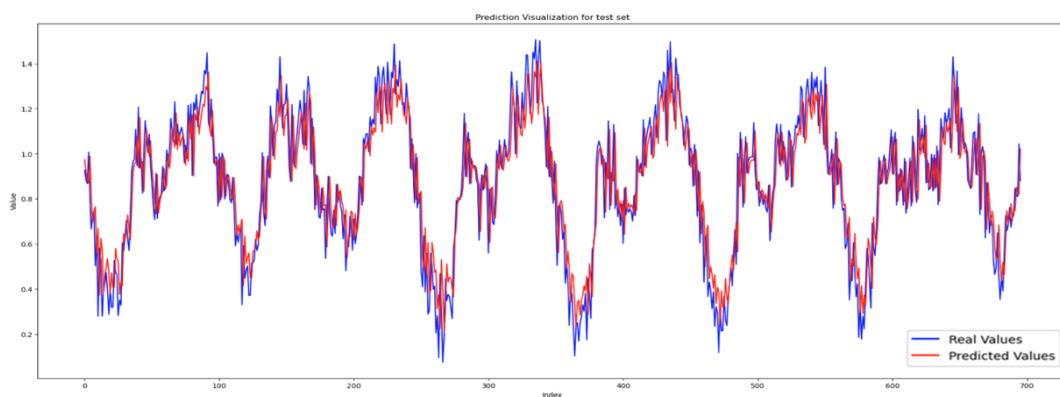
Στη συνέχεια θα παρουσιαστούν τα γραφήματα των επιλεγμένων μοντέλων για την πρόβλεψη του πρώτου βήματος. Το σχήμα 21 περιλαμβάνει τα αποτελέσματα των ελαχίστων τετραγώνων. Το σχήμα 22 περιλαμβάνει τα αποτελέσματα του Keras MLP δικτύου. Το σχήμα 23 περιλαμβάνει τα αποτελέσματα του Custom RNN δικτύου. Το σχήμα 24 περιλαμβάνει τα αποτελέσματα του Keras RNN δικτύου. Το σχήμα 25 περιλαμβάνει τα αποτελέσματα του Keras LSTM δικτύου ενώ το σχήμα 26 περιλαμβάνει τα αποτελέσματα του Keras GRU δικτύου.



Σχήμα 21: Γράφημα γραμμικού δικτύου της πρώτης εφαρμογής

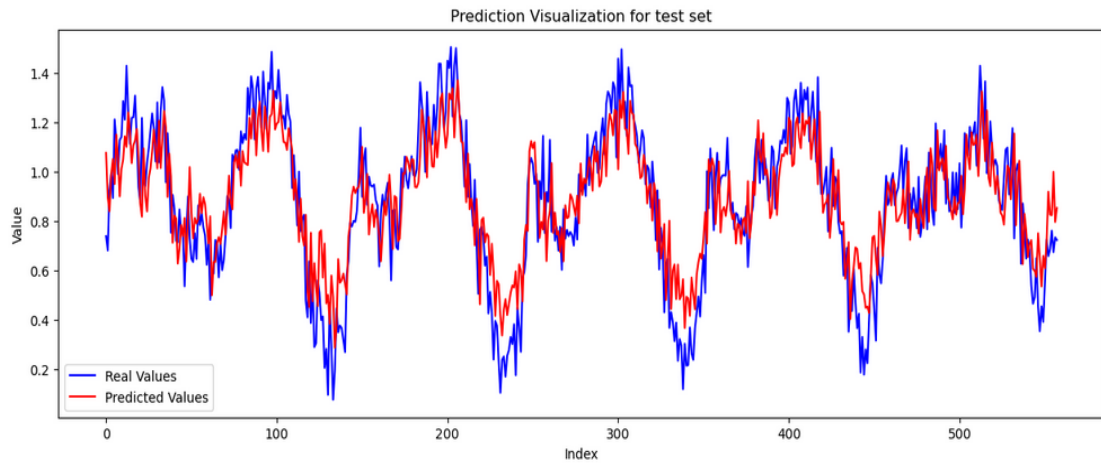


Σχήμα 22: Γράφημα Keras MLP δικτύου της πρώτης εφαρμογής

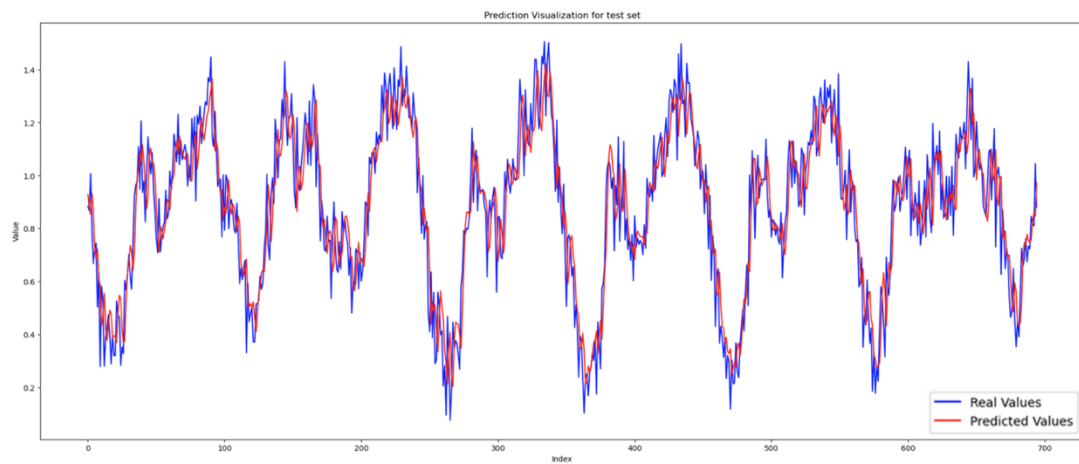


Σχήμα 23: Γράφημα Custom RNN δικτύου της πρώτης εφαρμογής

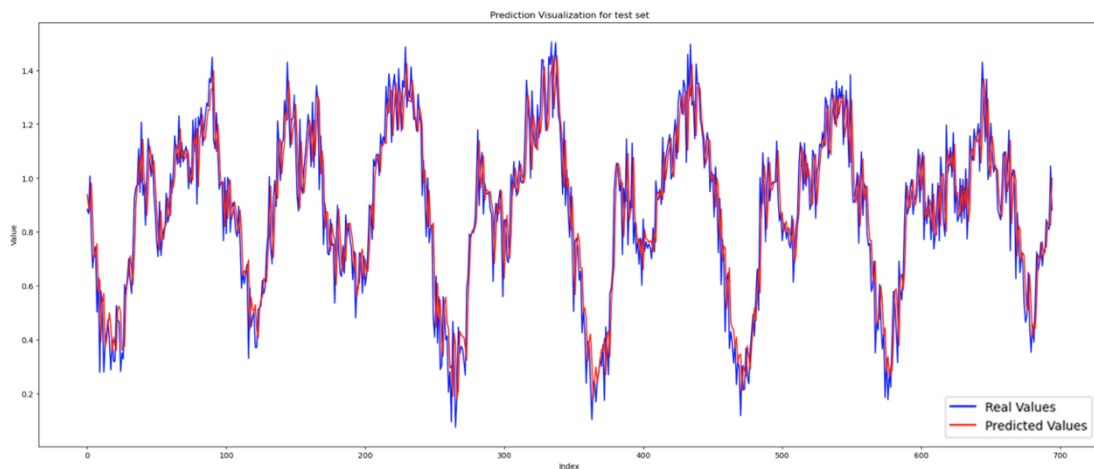




Σχήμα 24: Γράφημα Keras RNN δικτύου της πρώτης εφαρμογής



Σχήμα 25: Γράφημα Keras LSTM δικτύου της πρώτης εφαρμογής



Σχήμα 26: Γράφημα Keras GRU δικτύου της πρώτης εφαρμογής



#### 4.2.5 Συζήτηση

Η απόδοση όλων των προτεινόμενων μοντέλων φαίνεται να εξασθενεί όσο πιο μπροστά στο μέλλον απαιτείται να πραγματοποιηθεί η πρόβλεψη. Αυτό είναι λογικό καθώς δημιουργείται το πρόβλημα συσσώρευσης του σφάλματος. Αυτό σημαίνει ότι ακόμη και μικρές ανακρίβειες που δημιουργούνται στο παρελθόν, μπορούν να οδηγήσουν σε σημαντικές αποκλίσεις σε μεταγενέστερες προβλέψεις. Όλα τα αποτελέσματα είναι επηρεασμένα από το θόρυβο που έχει προστεθεί στα δεδομένα και παρουσιάζουν αρκετά χαμηλότερη απόδοση απότι θα είχαν χωρίς αυτόν. Φαίνεται πως το γραμμικό μοντέλο λειτουργεί λίγο χειρότερα από το MLP δίκτυο. Αυτό είναι αναμενόμενο αφού η λειτουργία των νευρωνικών δικτύων καθορίζεται από ένα μεγάλο αριθμό παραγόντων και παραμέτρων που μπορούν να επηρεάσουν τη μάθηση και την απόδοση ενός μοντέλου. Επιπλέον, το γραμμικό μοντέλο παρουσιάζει διαφορετική κατανομή των δεδομένων σε υποσύνολα χωρίς λειτουργίες που παρέχουν την ικανότητα επιλογής μοντέλου μέσω ενός validation set. Τα αναδρομικά νευρωνικά δίκτυα δείχνουν την ικανότητα αντίληψης χρονικών εξαρτήσεων και λειτουργούν καλύτερα στο συγκεκριμένο πρόβλημα. Το μοντέλο RNN του Keras αλλά και το LSTM μοντέλο είχαν χειρότερη απόδοση από το Custom RNN αποδεικνύοντας πως ένας ατομικά κατασκευασμένος αλγόριθμος όπου παρέχει τη δυνατότητα προσαρμογής των πράξεων και των διαδικασιών είναι προτιμότερος, αφού σε ορισμένες περιπτώσεις θα είναι ικανός να ξεπεράσει αναβαθμισμένες και σύνθετες συναρτήσεις βιβλιοθηκών. Επιπλέον, το μοντέλο RNN του Keras παρουσίασε χειρότερη απόδοση και από το MLP μοντέλο. Αυτό πιθανώς να ευθύνεται στο γεγονός του ότι οι λειτουργίες της βιβλιοθήκης δεν συνεργάζονται καλά με τις custom αρχικοποιήσεις που επιλέχθηκαν. Φαίνεται πως η καλύτερη αρχιτεκτονική ήταν αυτή που κάνει χρήση των μονάδων GRU αποδεικνύοντας πως αποτελούν την κατάλληλη αρχιτεκτονική για τα ορισμένα δεδομένα και ανάγκες. Παρατηρείται επίσης η σταδιακή αύξηση των παραμέτρων προς εκπαίδευση. Αυτή οδήγησε σε πιο πολύπλοκες αρχιτεκτονικές αλλά και σε εμφανή αύξηση του χρόνου εκπαίδευσης των μοντέλων.

### 4.3 Εφαρμογή 2<sup>η</sup> : Πρόβλεψη κατανάλωσης ενέργειας δημόσιου κτηρίου

#### 4.3.1 Περιγραφή

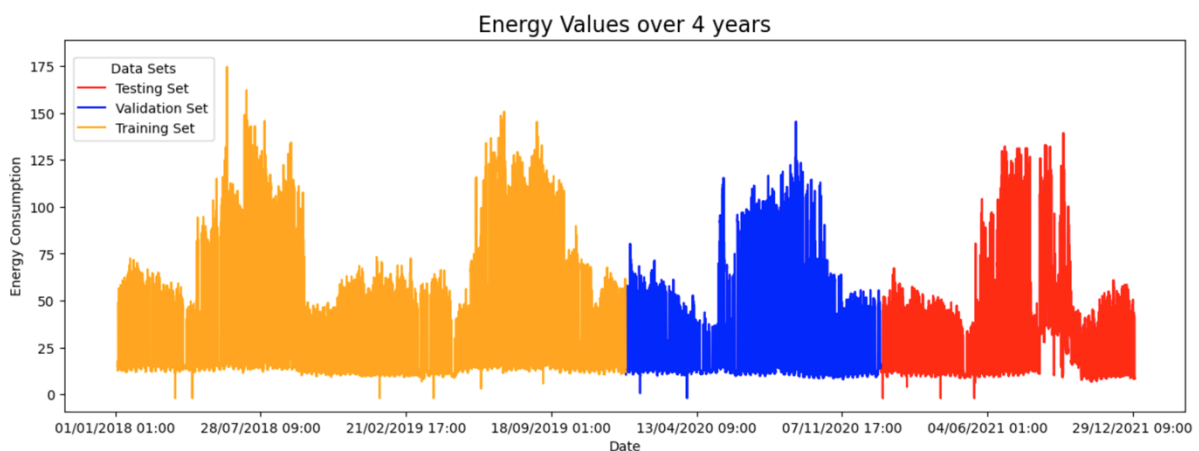
Η συγκεκριμένη εφαρμογή αποσκοπεί στην ανάπτυξη αλγορίθμων αναδρομικών νευρωνικών δικτύων με στόχο την πρόβλεψη της μελλοντικής κατά μερικών ωρών κατανάλωσης ενέργειας ενός δημοσίου κτηρίου. Συγκεκριμένα το κτήριο που ερευνάται είναι το δημαρχείο του δήμου Αιγάλεω. Η έρευνα θα χωριστεί σε δύο βασικά μέρη. Το πρώτο θα χρησιμοποιήσει μόνο μια μεταβλητή που θα περιγράφει την κατανάλωση της ενέργειας ενώ το δεύτερο θα προσθέσει

μεταβλητές από εξωτερικούς παράγοντες που είναι πιθανό να επηρεάσουν την απόδοση της των μοντέλων.

### 4.3.2 Δεδομένα

Οι τιμές της ενεργειακής κατανάλωσης απομονώθηκαν κατάλληλα από ένα μεγαλύτερο dataset και στη συνέχεια επεξεργάστηκαν για την είσοδό τους στα μοντέλα. Επιλέχθηκε να πραγματοποιηθεί ανάλυση δεδομένων που περιγράφουν την ωριαία κατανάλωση του κτηρίου για ένα διάστημα τεσσάρων χρόνων (35064 τιμές – Από 01/01/2018 01:00 μέχρι 01/01/2022 00:00). Επιλέχθηκε τα πρώτα 2 χρόνια (17520 τιμές – Από 01/01/2018 01:00 μέχρι 01/01/2020 00:00) να αποτελούν το Training dataset. Ο επόμενος χρόνος (8784 τιμές – Από 01/01/2020 01:00 μέχρι 01/01/2021 00:00) επιλέχθηκε ως Validation dataset. Ο τελευταίος χρόνος (8760 τιμές – Από 01/01/2021 01:00 μέχρι 01/01/2022 00:00) επιλέχθηκε ως Testing dataset.

Σκοπός ήταν να γίνει κατανομή των δεδομένων σε υποσύνολα (50% – 25% – 25%). Αυτό μεταφράζεται σωστά στον αριθμό των χρόνων (2 χρόνια – 1 χρόνος – 1 χρόνος) όμως παρουσιάζει μια ιδιαιτερότητα στο αριθμό των δεδομένων. Αυτό οφείλεται στο ότι το εύρος τιμών περιλαμβάνει το δίσεκτο έτος 2020. Έτσι το validation set περιέχει 24 παραπάνω τιμές από το testing set έτσι ώστε να περιλαμβάνει την έξτρα μέρα του Φεβρουαρίου και τα τελικά υποσύνολα να περιγράφουν ολοκληρωμένα έτη. Ο διαχωρισμός τους μπορεί να φανεί στο σχήμα 27. Μόνη εξαίρεση αποτελεί και πάλι το γραμμικό μοντέλο των ελαχίστων τετραγώνων όπου τα δεδομένα χωρίζονται σε δυο υποσύνολα με αναλογία (50% - 50%). Τα πρώτα 17520 δεδομένα αποτελούν το Training set ενώ τα υπόλοιπα αποτελούν το Testing set. Επιπλέον, επιλέχθηκε να γίνει κανονικοποίηση των δεδομένων μέσω Min – Max scaling και κάνοντας χρήση της εξίσωσης (4.5) για ένα διάστημα [-1, 1].

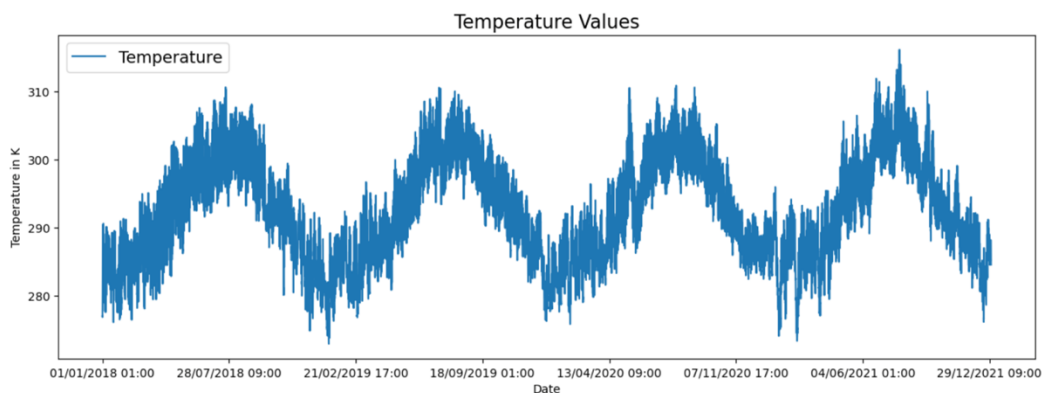


Σχήμα 27: Αναπαράσταση των δεδομένων της 2<sup>ης</sup> εφαρμογής

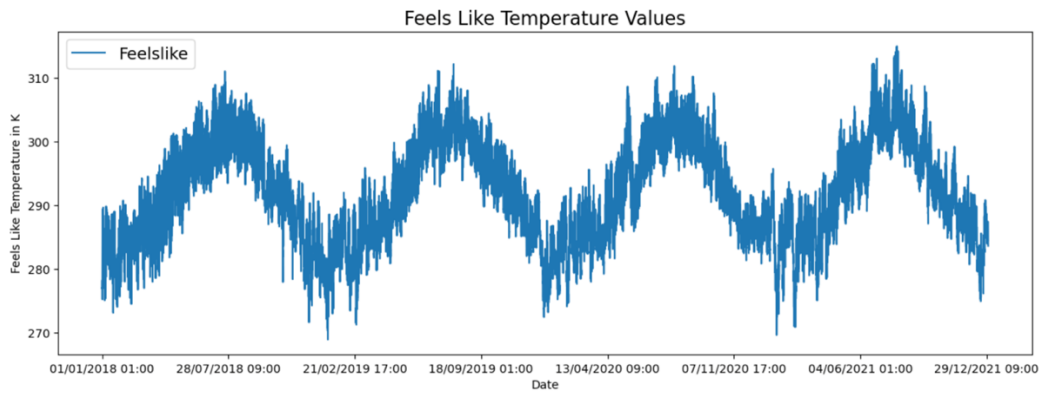
Για το Multivariate κομμάτι της έρευνας αποκτήθηκαν δεδομένα και από ένα ακόμη dataset που περιείχε διάφορα μετεωρολογικά δεδομένα. Θα πρέπει όμως να αναφερθεί, ότι το μέρος που αποκτήθηκαν βρίσκεται περίπου 1050 μέτρα μακριά από τη τοποθεσία του δημαρχείου όπως φαίνεται στο σχήμα 28.



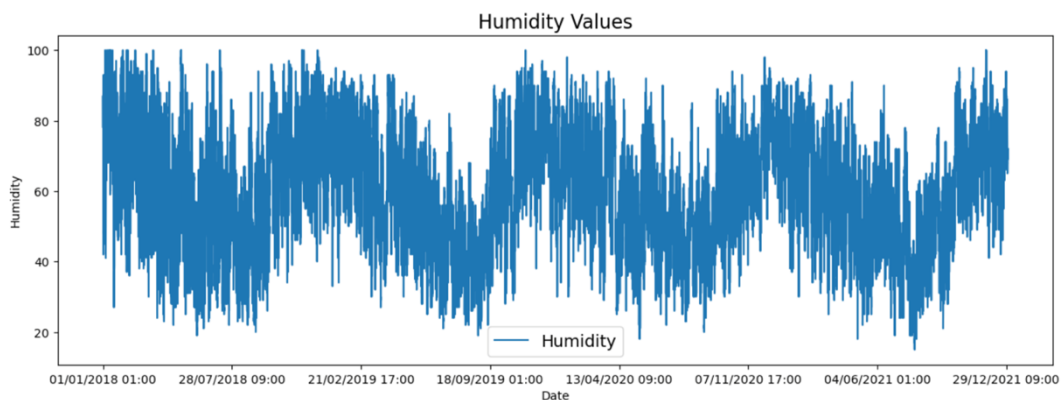
**Σχήμα 28:** Απόσταση τοποθεσίας απόκτησης μετεωρολογικών δεδομένων από το δημαρχείο. Αυτό μείωσε αρκετά την ποιότητα αλλά και την ποσότητα των δεδομένων που μπορούσαν να χρησιμοποιηθούν στη μελέτη. Μεταβλητές όπως αυτές που περιγράφουν την συμπεριφορά του ανέμου δεν παρείχαν αρκετή πληροφορία ενώ άλλες εξαιρετικά σημαντικές μεταβλητές δεν παρουσίαζαν τιμές για όλες τις απαιτούμενες χρονικές περιόδους. Ύστερα από κάποιες δοκιμές, αποφασίστηκε να παρθούν 3 διαφορετικές μεταβλητές οι οποίες περιγράφουν την ωριαία μέση θερμοκρασία σύμφωνα με τα μετρητικά όργανα, την ωριαία αίσθηση της θερμοκρασίας (Feels Like Temperature) αλλά και την ωριαία μέση υγρασία. Αυτές ήταν το λιγότερο πιθανό να έχουν επηρεαστεί σε μεγάλο βαθμό από την απόσταση. Οι τιμές των μεταβλητών περιλαμβάνονται στα σχήματα 29, 30 και 31.



**Σχήμα 29:** Αναπαράσταση ωριαίας μέσης θερμοκρασίας σύμφωνα με τα μετρητικά όργανα



**Σχήμα 30:** Αναπαράσταση ωριαίας αίσθησης της θερμοκρασίας (Feels Like Temperature)



**Σχήμα 31:** Αναπαράσταση ωριαίας μέσης υγρασίας

Όλα τα dataset περιλάμβαναν τις απαραίτητες ημερομηνίες και ώρες παρέχοντας την ικανότητα δημιουργίας κατηγορικών δεδομένων (Categorical data) τα οποία είναι ικανά να βοηθήσουν στην καλύτερη κατανόηση και περιγραφή του προβλήματος. Μέσω του feature engineering, ένα εύρος ημερομηνιών και ωρών μπορεί να διαιρεθεί και να πραγματοποιηθεί η δημιουργία διακριτών κατηγοριών και διαστημάτων. Με αυτόν τον τρόπο μη αριθμητικά δεδομένα μπορούν να ποσοτικοποιηθούν και να χρησιμοποιηθούν κατάλληλα ως μεταβλητές εισόδου των μοντέλων. Δημιουργήθηκαν αριθμητικές μεταβλητές ακέραιων αριθμών. Αυτές περιγράφουν την ώρα της ημέρας (0 οι 00:00 μέχρι 23 οι 23:00), την ημέρα της εβδομάδας (0 η Δευτέρα μέχρι 6 η Κυριακή) αλλά και το μήνα του έτους (0 ο Ιανουάριος μέχρι 11 ο Δεκέμβριος). Δημιουργήθηκαν επίσης δυαδικές μεταβλητές καταστάσεων. Η μία αφορούσε το αν κάποια συγκεκριμένη ημερομηνία αποτελούσε επίσημη αργία (0 αν όχι – 1 αν ναι) για τους εργαζομένους του δημοσίου ενώ η δεύτερη περιέγραφε αν μια συγκεκριμένη ώρα αποτελούσε μέρα ή νύχτα (1 για μέρα ενώ 0 για νύχτα). Η αλλαγή της ώρας κατά τη διάρκεια του έτους κατέστησε λίγο δύσκολη αυτή τη περιγραφή αλλά μετά από ορισμένες δοκιμές φάνηκε πως το μοντέλο απέδιδε καλύτερα όταν η μέρα περιγράφονταν από διάστημα μεταξύ των 07:00 και 20:00.

Όλες οι παραπάνω μεταβλητές της Multivariate μελέτης δέχτηκαν παρόμοια επεξεργασία με τη μεταβλητή της ενέργειας. Τέλος επιλέχτηκε και μια ακόμα μεταβλητή η οποία παρείχε την κατανάλωση της ενέργειας ακριβώς μια μέρα πριν.

### 4.3.3 Υλοποίηση

Αρχικά θα πραγματοποιηθεί η Univariate μελέτη χρησιμοποιώντας ιστορικά δεδομένα της κατανάλωσης ενέργειας. Το μοντέλο των ελαχίστων τετραγώνων θα χρησιμοποιήσει έξι προηγούμενες τιμές ενώ το Keras MLP θα ρυθμιστεί βάσει των παραμέτρων του πίνακα 7. Η μόνη διαφορά που θα παρουσιάζει είναι ο αριθμός παρτίδας του που πλέον θα είναι ίσος με 16. Στον πίνακα 10 παρουσιάζεται η ρύθμιση των παραμέτρων όλων των αναδρομικών μοντέλων:

**Πίνακας 10:** Παράμετροι αναδρομικών μοντέλων της πρώτης εφαρμογής

ΠΑΡΑΜΕΤΡΟΣ	CUSTOM RNN	KERAS RNN	KERAS LSTM	KERAS GRU
Αριθμός ιστορικών δεδομένων	2	2	3	3
Αρχικοποίηση	Custom Xavier	Custom Xavier	Keras Default	Keras Default
Κρυφές στοιβάδες	1	1	1	1
Αριθμός μονάδων	25	25	40	40
Συνάρτηση ενεργοποίησης	tanh	tanh	tanh	tanh
Συνάρτηση απώλειας	MSE	MSE	MSE	MSE
Optimizer	SGD	SGD	SGD	SGD
Ρυθμός μάθησης	$10^{-3}$	$10^{-3}$	$10^{-3}$	$10^{-3}$
Batch	1	1	1	1
Epochs	10	10	20	20

Για την Multivariate μελέτη έχει επιλεγθεί να γίνει χρήση του Keras GRU δικτύου λόγω της καλής ανταπόκρισης των δεδομένων σε αυτό. Ο πίνακας 11 παρουσιάζει τις ρυθμίσεις του προτεινόμενου μοντέλου. Οι ίδιοι παράμετροι θα χρησιμοποιηθούν σε ένα Keras LSTM και ένα Keras RNN δίκτυο προκειμένου πραγματοποιηθεί σύγκριση των αποδόσεών τους.

**Πίνακας 11:** Παράμετροι προτεινόμενου μοντέλου GRU της δεύτερης εφαρμογής

ΠΑΡΑΜΕΤΡΟΣ	ΕΠΙΛΟΓΗ
Αριθμός ιστορικών δεδομένων	3
Αρχικοποίηση	Keras Default
Κρυφές στοιβάδες	1
Αριθμός νευρώνων	100
Συνάρτηση ενεργοποίησης	tanh
Συνάρτηση απώλειας	MSE
Optimizer	SGD
Ρυθμός μάθησης	$10^{-3}$
Batch	1
Epochs	25

### 4.3.4 Αποτελέσματα

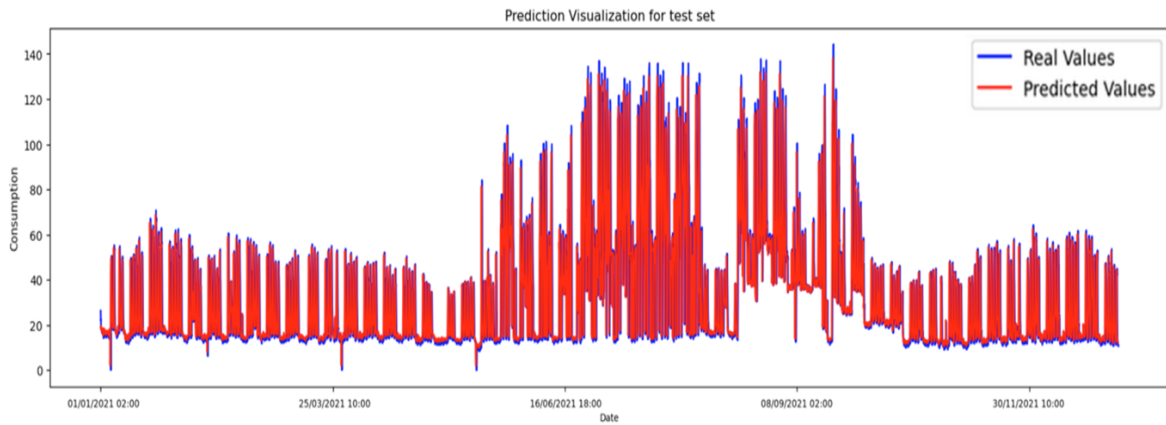
Όλα τα αποτελέσματα περιγράφουν τον μέσο όρο 10 διαφορετικών επαναλήψεων. Έγινε χρήση συγκεκριμένων seed για την εισαγωγή της στοχαστικότητας έτσι ώστε να μπορούν να γίνουν οι κατάλληλες συγκρίσεις. Σε κάθε επανάληψη επιλέγονταν οι παράμετροι του Epoch που παρουσίαζε την ελάχιστη τιμή στο validation loss. Περιλαμβάνονται στον πίνακα 12 όπου πραγματοποιείται και η επιλογή του καλύτερου μοντέλου.

**Πίνακας 12:** Αποτελέσματα Univariate μελέτης της δεύτερης εφαρμογής

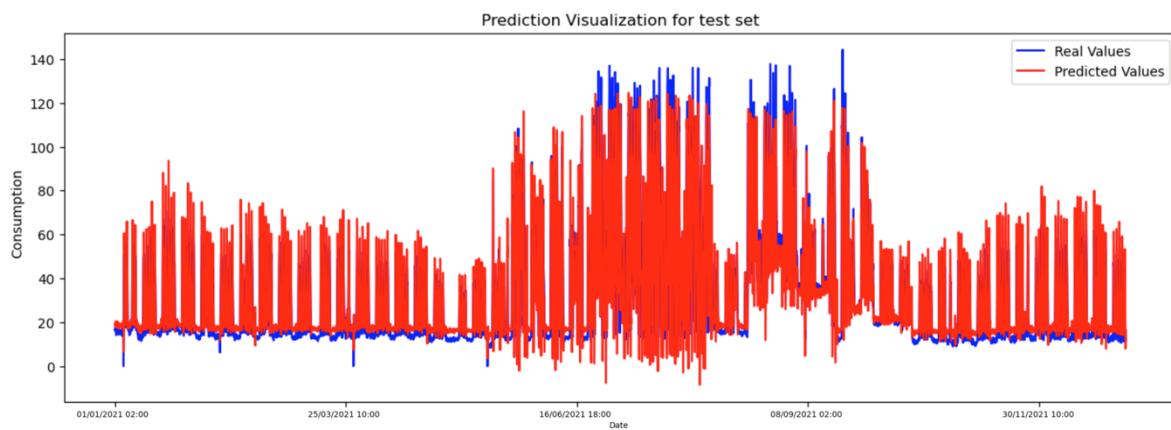
	LINEAR			KERAS MLP		
	MAE	RMSE	R2	MAE	RMSE	R2
1 STEP	4,2239	8,1812	0,9084	6,4216	9,2174	0,8725
2 STEPS	7,2958	13,5243	0,7497	9,4976	13,3062	0,7349
3 STEPS	10,2591	17,9207	0,5606	11,8466	15,9638	0,6211
4 STEPS	13,1119	21,7198	0,3546	14,2795	18,9202	0,4679
5 STEPS	15,8189	25,0637	0,1406	16,3501	21,2791	0,3265
6 STEPS	18,3199	28,0374	-0,0753	18,1453	23,2130	0,1991
	CUSTOM RNN			KERAS RNN		
	MAE	RMSE	R2	MAE	RMSE	R2
1 STEP	3,9039	7,3359	0,9148	5,1429	8,3626	0,8976
2 STEPS	7,1616	12,1010	0,7844	7,9676	12,0779	0,7849
3 STEPS	10,0470	15,9287	0,6264	11,0010	15,8326	0,6298
4 STEPS	12,5466	19,0771	0,4642	13,5697	18,5471	0,4927
5 STEPS	14,7526	21,6729	0,3084	15,8734	21,1518	0,3405
6 STEPS	16,6771	23,8322	0,1639	17,5332	22,7914	0,2347
	KERAS LSTM			KERAS GRU		
	MAE	RMSE	R2	MAE	RMSE	R2
1 STEP	5,9548	9,1126	0,8766	4,7341	7,2550	0,9223
2 STEPS	9,0902	12,7170	0,7611	7,5987	11,2312	0,8142
3 STEPS	11,7566	15,8986	0,6276	10,8273	14,9075	0,6728
4 STEPS	14,3596	18,8467	0,4764	13,3733	17,8851	0,5290
5 STEPS	16,5344	21,2968	0,3317	15,5877	20,3736	0,3888
6 STEPS	18,2458	23,3022	0,2001	17,2472	22,3291	0,2660

Στη συνέχεια θα παρουσιαστούν τα γραφήματα των επιλεγμένων μοντέλων για την πρόβλεψη του πρώτου βήματος. Το σχήμα 32 περιλαμβάνει τα αποτελέσματα των ελαχίστων τετραγώνων. Το σχήμα 33 περιλαμβάνει τα αποτελέσματα του Keras MLP δικτύου. Το σχήμα 34 περιλαμβάνει τα αποτελέσματα του Custom RNN δικτύου. Το σχήμα 35 περιλαμβάνει τα αποτελέσματα του Keras RNN δικτύου. Το σχήμα 36 περιλαμβάνει τα αποτελέσματα του Keras LSTM δικτύου ενώ το σχήμα 37 περιλαμβάνει τα αποτελέσματα του Keras GRU δικτύου.

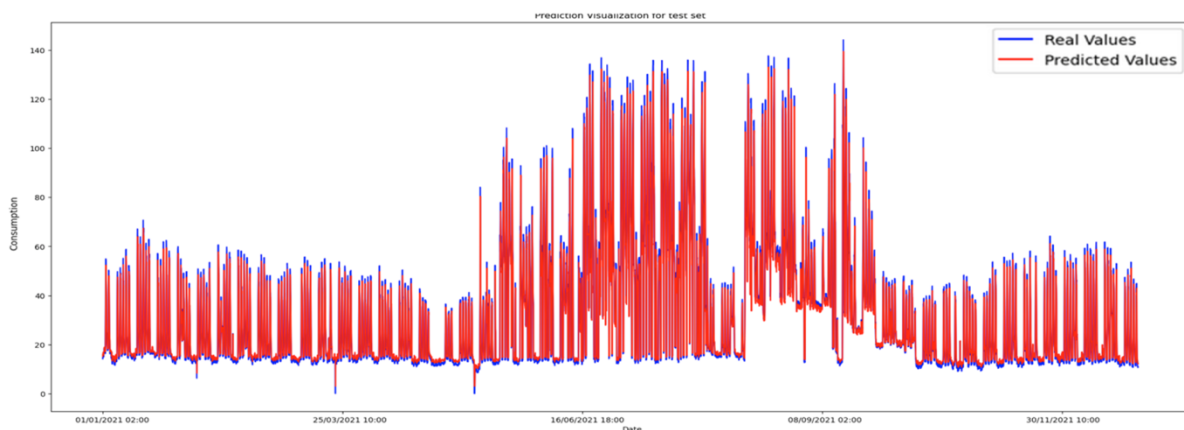




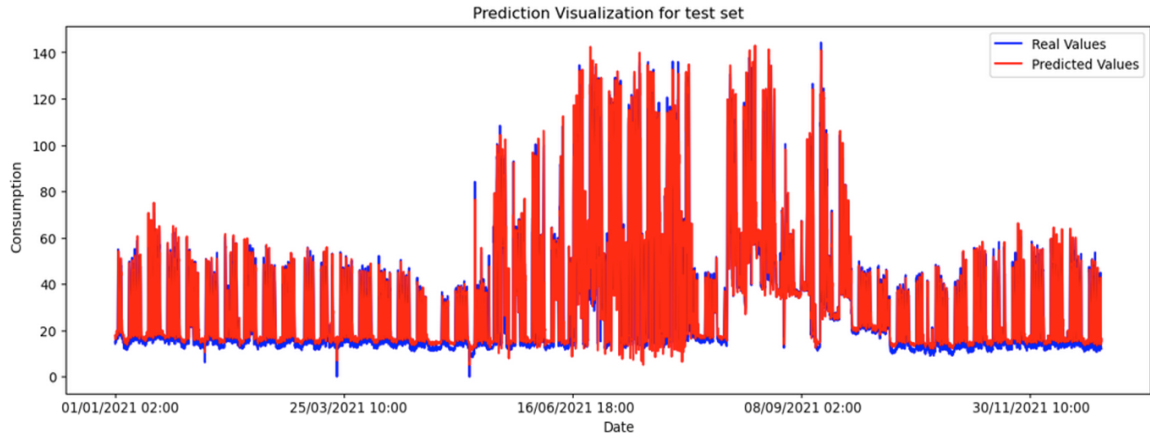
Σχήμα 32: Γράφημα γραμμικού δικτύου της δεύτερης εφαρμογής



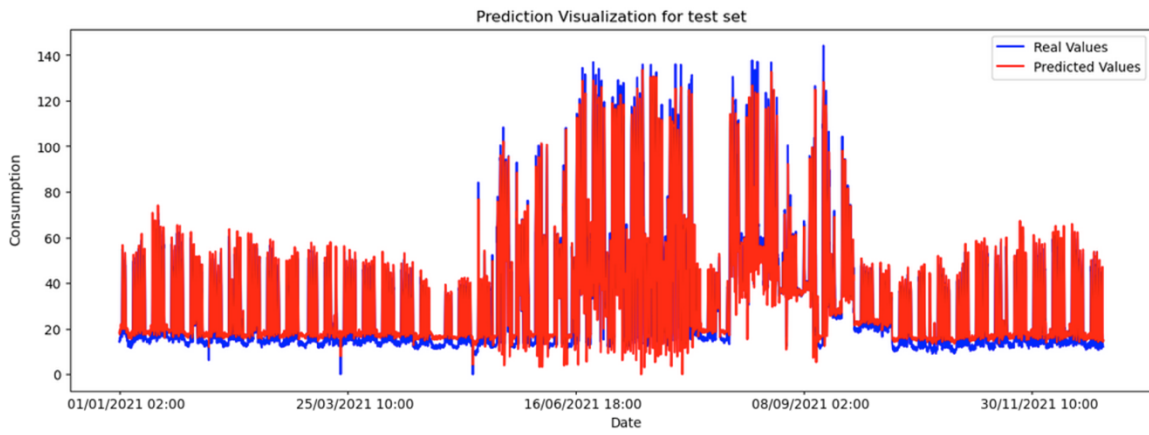
Σχήμα 33: Γράφημα Keras MLP δικτύου της δεύτερης εφαρμογής



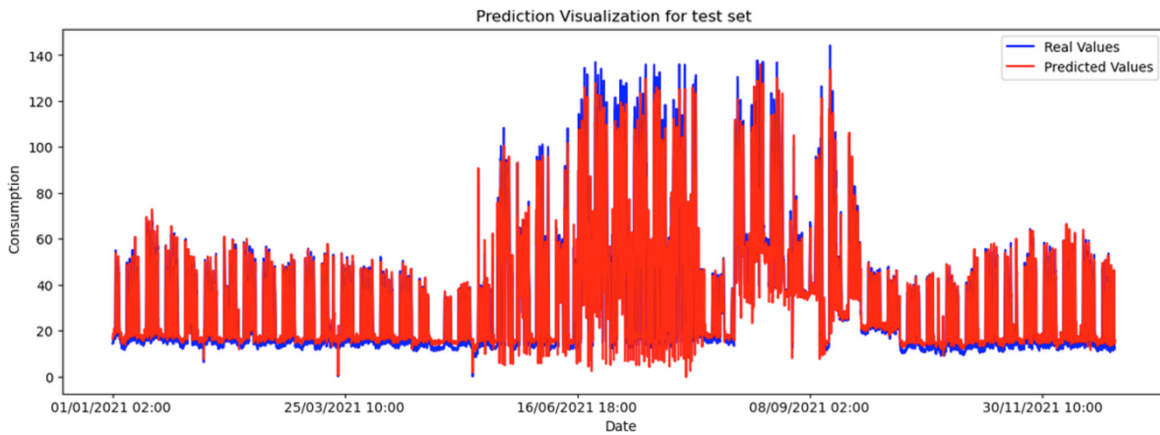
Σχήμα 34: Γράφημα Custom RNN δικτύου της δεύτερης εφαρμογής



Σχήμα 35: Γράφημα Keras RNN δικτύου της δεύτερης εφαρμογής



Σχήμα 36: Γράφημα Keras LSTM δικτύου της δεύτερης εφαρμογής



Σχήμα 37: Γράφημα Keras GRU δικτύου της δεύτερης εφαρμογής

Στον πίνακα 13 περιλαμβάνονται τα αποτελέσματα του επιλεγμένου Keras GRU μοντέλου τα οποία συγκρίνονται με το αντίστοιχο Keras LSTM και Keras RNN.



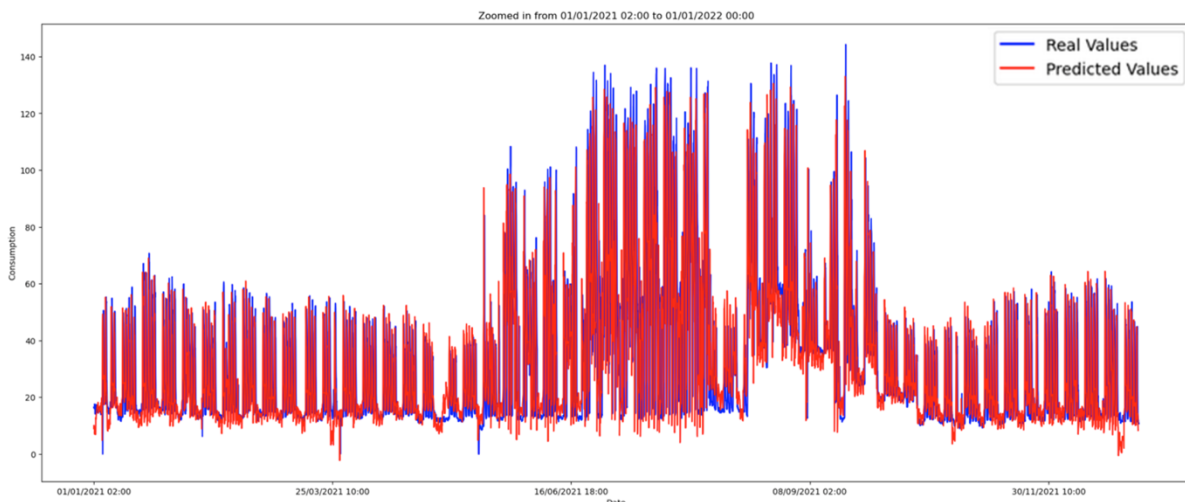
**Πίνακας 13:** Αποτελέσματα Multivariate μελέτης της δεύτερης εφαρμογής

	KERAS GRU		
	MAE	RMSE	R2
1 STEP	5,1820	4,3525	0,9310
2 STEPS	7,9852	11,0971	0,8270
3 STEPS	10,5002	14,1778	0,7013
4 STEPS	12,6190	16,7376	0,5835
5 STEPS	14,2478	18,8042	0,4747
6 STEPS	14,8668	19,7091	0,4226

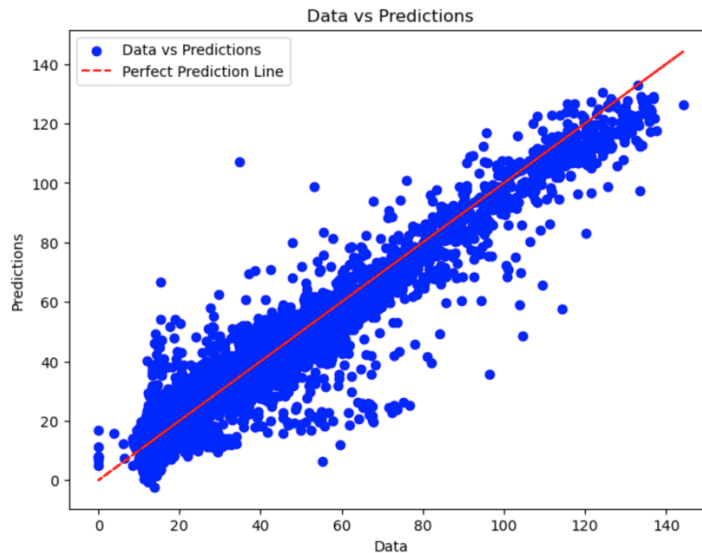
  

	KERAS RNN			KERAS LSTM		
	MAE	RMSE	R2	MAE	RMSE	R2
1 STEP	5,7088	8,2821	0,8979	5,6699	8,0711	0,9033
2 STEPS	8,3912	11,9184	0,7887	8,8273	12,0264	0,7851
3 STEPS	10,5556	14,2971	0,6960	10,6760	14,6803	0,6798
4 STEPS	11,8892	16,4596	0,5971	12,9634	17,1371	0,5636
5 STEPS	12,9666	17,9348	0,5220	14,7611	19,6030	0,4291
6 STEPS	13,8765	19,2255	0,4503	16,5029	21,9598	0,2836

Στο σχήμα 38 περιλαμβάνεται το γράφημα που απεικονίζει τις προβλέψεις του πρώτου βήματος του προτεινόμενου Keras GRU μοντέλου. Επιπλέον, στο σχήμα 39 παρουσιάζεται ένα γράφημα διασποράς μεταξύ των πραγματικών και προβλεπόμενων τιμών. Σε αυτό φαίνεται πόσο αποκλίνουν από την ιδανική ευθεία των 45 μοιρών αλλά και μια πρώτη εικόνα της κατανομής των δεδομένων που υπερεκτιμούνται (σημεία πάνω από τη βέλτιστη ευθεία) και υποεκτιμούνται (σημεία κάτω από τη βέλτιστη ευθεία).



**Σχήμα 38:** Προτεινόμενο Keras GRU μοντέλο από Multivariate μελέτη



**Σχήμα 39:** Γράφημα διασποράς μεταξύ των πραγματικών και προβλεπόμενων τιμών προτεινόμενου μοντέλου

#### 4.3.5 Συζήτηση

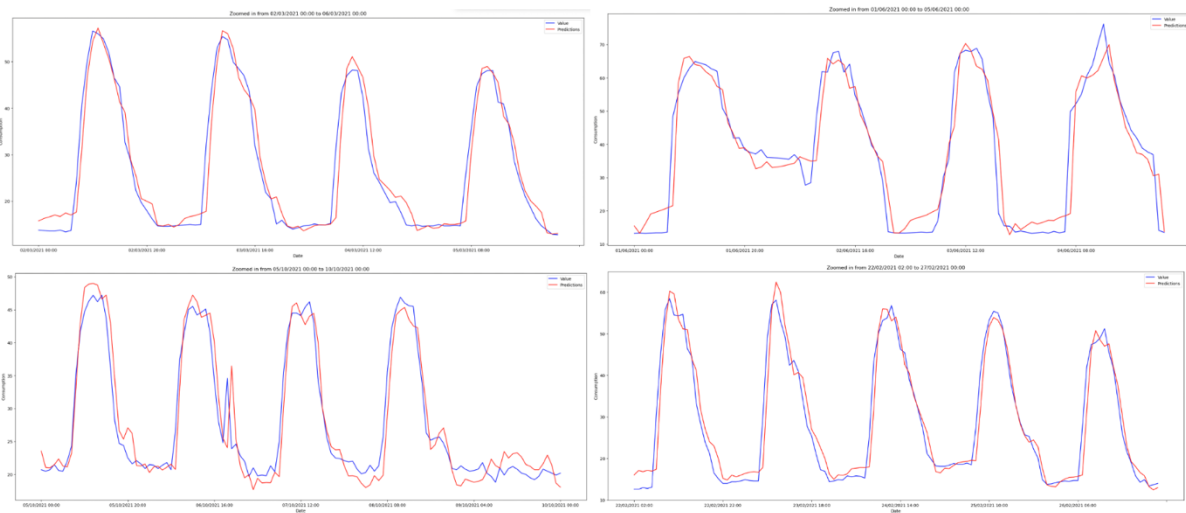
Είναι εμφανές πως η απόδοση των μοντέλων εξασθενεί σε μεγάλο βαθμό όταν επιχειρείται να πραγματοποιηθεί πρόβλεψη τιμών που βρίσκονται σε αργότερα χρονικά βήματα. Αυτό το φαινόμενο είναι πιο έντονο ακόμα και από την πρώτη εφαρμογή όμως αυτό είναι αναμενόμενο αφού πλέον ερευνάται ένα πραγματικό πρόβλημα το οποίο είναι πολύ πιο περίπλοκο από μια ελεγχόμενη συνθετική χρονοσειρά. Εξαιρεση αποτελεί το Multivariate μοντέλο. Φαίνεται πως οι επιπρόσθετες μεταβλητές παρείχαν απαραίτητες πληροφορίες που το βοήθησαν να αναγνωρίσει ορισμένα μοτίβα και χρονικές εξαρτήσεις.

##### - Univariate μελέτη

Το μοντέλο των ελαχίστων τετραγώνων προσφέρει αρχικά μια ικανοποιητική απόδοση δεδομένου του προβλήματος αλλά φαίνεται πως είναι το λιγότερο ικανό να προβλέψει καταναλώσεις μεταγενέστερων χρονικών στιγμών. Το Keras MLP αδυνατεί να παρέχει απόδοση καλύτερη από αυτήν των αναδρομικών δικτύων. Για άλλη μια φορά το custom RNN υπερτερεί σε σχέση με το Keras RNN αλλά και το Keras LSTM δείχνοντας την ικανότητα του αλγορίθμου και των μεθόδων εκπαίδευσης που περιέχει. Το Keras GRU παρείχε την καλύτερη απόδοση από όλα τα μοντέλα αποδεικνύοντας πόσο σημαντική είναι η ικανότητα διατήρησης και ενσωμάτωσης της μακροπρόθεσμης μνήμης. Το μοντέλο αυτό, ανταποκρίθηκε καλύτερα στα δεδομένα και απαίτησε μικρότερο χρόνο εκπαίδευσης σε σχέση με το Keras LSTM.

- Multivariate μελέτη

Το προτεινόμενο Keras GRU μοντέλο είχε καλύτερη απόδοση από τα αντίστοιχα Keras LSTM και Keras RNN μοντέλα φανερώνοντας για ακόμη μια φορά πως συνεργάζεται καλά με τα δεδομένα του προβλήματος. Η εισαγωγή των μεταβλητών απέφερε πολύ θετικά αποτελέσματα καθώς μειώθηκε η εξασθένηση απόδοσης για μεταγενέστερες μελλοντικές στιγμές και επιπλέον, το μοντέλο μπορούσε να προβλέψει χαμηλότερες τιμές απόδοσης που εμφανίζονταν τις μέρες που το δημαρχείο παρέμενε κλειστό. Με την Univariate έρευνα αυτό δεν ήταν εφικτό καθώς παρατηρήθηκε πως παρόλο που παρουσιαζόταν η συμπεριφορά των κατώτερων τιμών, η πρόβλεψη έδειχνε σταθερά ψηλότερα καθιστώντας το μοντέλο μη ιδανικό. Το προτεινόμενο μοντέλο παρουσίασε μια καλή συμπεριφορά, όπως φαίνεται και από τις περιόδους πρόβλεψης που απεικονίζονται στο σχήμα 40.

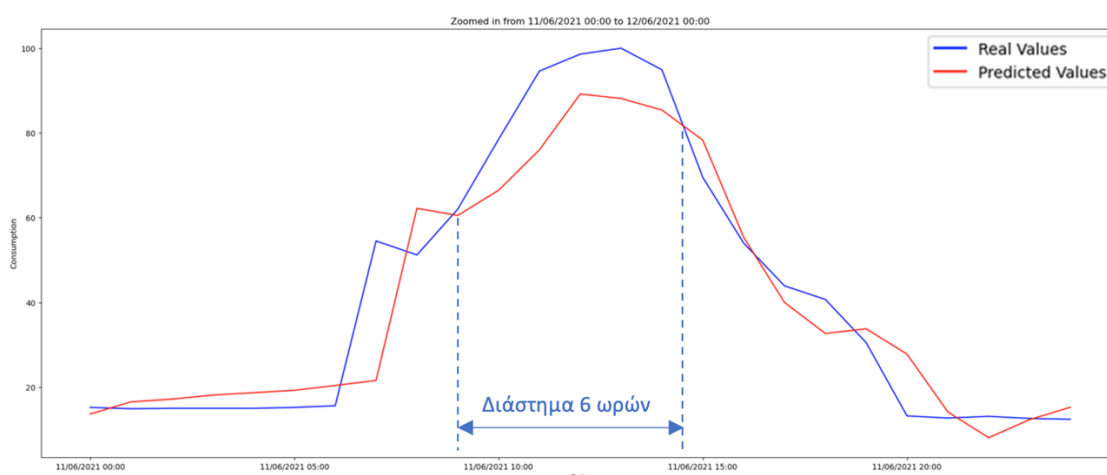


Σχήμα 40: Παραδείγματα καλής λειτουργίας του μοντέλου ανά εποχή

#### 4.3.6 Πιθανή εφαρμογή προτεινόμενου μοντέλου

Η ικανοποιητική ακρίβειά πρόβλεψης του μοντέλου δίνει την δυνατότητα πρακτικών εφαρμογών στο χώρο του δημαρχείου. Μια πιθανή εφαρμογή του θα μπορούσε να είναι η εμφάνιση κάποιας προειδοποίησης σε περιπτώσεις όπου προβλέπεται πως η κατανάλωση θα είναι σταθερά χαμηλότερη για ένα χρονικό εύρος κάποιων ωρών. Αυτό θα μπορούσε να ανιχνεύσει μια συμπεριφορά πέρα από το συνηθισμένο στο δίκτυο και πολλές φορές θα μπορούσε να φανερώσει προβλήματα που σχετίζονται με το ηλεκτρικό δίκτυο και την κατανομή ενέργειας του κτηρίου. Με αυτόν τον τρόπο θα ήταν πιθανό ο διαχειριστής του κτηρίου να αξιοποιήσει αυτή τη προειδοποίηση έτσι ώστε να ερευνήσει πιθανές αιτίες εμφάνισης του φαινομένου. Με τον τρόπο αυτό θα μπορούσε να επιτευχθεί ο περιορισμός του προβλήματος και εξοικονόμηση της καταναλισκόμενης ενέργειας στο κτήριο. Ένα τέτοιο

παράδειγμα εμφανίζεται στο σχήμα 41 όπου παρουσιάζεται μια πρόβλεψη χαμηλότερης κατανάλωσης από το μοντέλο για ένα χρονικό εύρος 6 συνεχόμενων ωρών (Από τις 11/06/2021 στις 09:00 μέχρι τις 11/06/2021 στις 14:00). Ο πίνακας 14 περιλαμβάνει τον υπολογισμό του μέσου απόλυτου σφάλματος για το χρονικό εύρος αυτών των 6 ωρών. Σημειώνεται ότι τα δεδομένα αυτά αντιστοιχούν σε μια Παρασκευή που δεν αναφέρεται ως κάποια ειδική περίπτωση αργίας, ενώ οι ώρες παρατήρησης του φαινομένου αποτελούν ενεργές ώρες γραφείου. Με βάση την ανωτέρω, κατά το συγκεκριμένο διάστημα είναι πολύ πιθανό να υπήρξε κάποιο πρόβλημα, ή λάθος στη χρήση της ενέργειας, το οποίο προκάλεσε την αυξημένη κατανάλωση.



**Σχήμα 41:** Ανίχνευση μη συνηθισμένης συμπεριφοράς στην πρόβλεψη της κατανάλωσης

**Πίνακας 14:** Δεδομένα χρονικού διαστήματος παρατήρησης μη συνηθισμένης συμπεριφοράς

Datetime	Consumption	Prediction	Absolute Error
11/06/2021 09:00	62,0	60,5233	1,4767
11/06/2021 10:00	78,5	66,4565	12,0435
11/06/2021 11:00	94,6	76,0106	18,5894
11/06/2021 12:00	98,6	89,1889	9,4111
11/06/2021 13:00	100,0	88,1371	11,8629
11/06/2021 14:00	94,9	85,3973	9,5027
Region MAE			10,48105
Model MAE			5,1820

---

## ΣΥΜΠΕΡΑΣΜΑΤΑ

---

Βασικό αντικείμενο μελέτης της παρούσας διπλωματικής εργασίας αποτέλεσε η αρχιτεκτονική των αναδρομικών νευρωνικών δικτύων και η χρήση της για τη μοντελοποίηση μη γραμμικών χρονοσειρών. Στο πλαίσιο των στόχων που τέθηκαν, η διπλωματική ανέδειξε σημαντικά αποτελέσματα που επικυρώνουν το θεωρητικό υπόβαθρο και ενισχύουν την κατανόηση του τρόπου λειτουργίας των αναδρομικών νευρωνικών δικτύων σε συνθετικά και πραγματικά δεδομένα. Επιβεβαιώθηκε η ικανότητά τους να διαχειρίζονται ακολουθιακά ιστορικά δεδομένα αλλά και να αντιλαμβάνονται τις σύνθετες χρονικές τους εξαρτήσεις. Σημαντικό στοιχείο μελέτης αποτέλεσε και η σύγκρισή των αναδρομικών νευρωνικών δικτύων με διαφορετικά μοντέλα πρόβλεψης, αναδεικνύοντας την υπεροχή τους στις προδιαγραφές των προβλημάτων που περιγράφονται από δεδομένα χρονοσειρών.

Πιο συγκεκριμένα, αναλύθηκαν συνολικά έξι διαφορετικές αρχιτεκτονικές. Αρχικά, υλοποιήθηκε το γραμμικό μοντέλο των ελαχίστων τετραγώνων και το μοντέλο πρόσθιας τροφοδότησης πολλών επιπέδων (MLP) με στόχο τη σύγκριση της απόδοσής τους σε σχέση με τα αναδρομικά μοντέλα. Στη συνέχεια, πραγματοποιήθηκε ανάπτυξη ενός βασικού μοντέλου αναδρομικού νευρωνικού δικτύου που εκπαιδεύεται με τη μέθοδο οπισθοδιάδοσης στο χρόνο για την ανάδειξη πλεονεκτημάτων και περιορισμών σε σχέση με το αντίστοιχο μοντέλο που παρέχεται από την καθιερωμένη βιβλιοθήκη του Keras. Η συγκεκριμένη βιβλιοθήκη χρησιμοποιήθηκε ακόμη και για τη κατασκευή αναδρομικών μοντέλων αυξημένης πολυπλοκότητας που στοχεύουν στην αντιμετώπιση του προβλήματος εμφάνισης της εξαφανιζόμενης κλίσης. Αυτά είναι το μοντέλο LSTM και το μοντέλο GRU.

Μέσω των αρχιτεκτονικών ερευνήθηκαν δυο απαιτητικές εφαρμογές, στα πλαίσια των οποίων αναλύθηκε η περίπτωση πρόβλεψης του επόμενου χρονικού βήματος χρονοσειράς (Single-step Prediction), αλλά και η πρόβλεψη μιας αλληλουχίας μελλοντικών τιμών (Multi-step Prediction). Η πρώτη εφαρμογή αφορούσε τα συνθετικά δεδομένα που παράχθηκαν από μια χαοτική συνάρτηση Mackey – Glass. Μέσω αυτής πραγματοποιήθηκε μια πρώτη αξιολόγηση των αρχιτεκτονικών αλλά και ο έλεγχος της ορθής λειτουργίας τους. Όλα τα αναδρομικά μοντέλα παρουσίασαν ικανοποιητική απόδοση, όμως το GRU μοντέλο αναδείχτηκε ως η καλύτερη επιλογή για τις προδιαγραφές του συγκεκριμένου προβλήματος.

Η δεύτερη εφαρμογή περιλάμβανε πραγματικά δεδομένα κατανάλωσης ενέργειας ενός δημόσιου κτηρίου. Επιδιώχθηκε η βελτίωση της ακρίβειας και της απόδοσης των προβλέψεων κατανάλωσης, με στόχο την αντιμετώπιση της αυξανόμενης ανάγκης εύρεσης βιώσιμων

λύσεων του ενεργειακού προβλήματος. Ύστερα από εκτενείς πειραματικές διαδικασίες, προτάθηκε ένα αναδρομικό νευρωνικό δίκτυο αποτελούμενου από μονάδες GRU. Η επιλογή του μοντέλου δεν αποτέλεσε εύκολο ζήτημα καθώς ανέδειξε προκλήσεις που έχουν να κάνουν με την εύρεση βέλτιστων ρυθμίσεων των παραμέτρων αλλά και διαχείρισης του όγκου των δεδομένων. Χρησιμοποιήθηκε ένα εύρος μεταβλητών που περιέγραφαν εξωτερικούς μετεωρολογικούς παράγοντες επιρροής του μοντέλου αλλά και κατηγορικών δεδομένων από την χρονική στιγμή καταγραφής της κατανάλωσης ενέργειας. Το προτεινόμενο μοντέλο παρείχε ικανοποιητική ακρίβεια και ανέδειξε την ικανότητα ανίχνευσης μη συνηθισμένης συμπεριφοράς στην πρόβλεψη της κατανάλωσης. Τέλος προτάθηκε να χρησιμοποιηθεί κάποιο είδος προειδοποίησης σε περιπτώσεις που η πρόβλεψη του μοντέλου είναι σταθερά χαμηλότερη για ένα χρονικό εύρος κάποιων ωρών. Περαιτέρω έρευνα από κάποιον υπεύθυνο του κτηρίου θα μπορούσε να φανερώσει προβλήματα που σχετίζονται με το ηλεκτρικό δίκτυο και την κατανομή της ενέργειας.

Η παρούσα μελέτη παρουσιάζει πολλές προοπτικές προς μελλοντική έρευνα. Αρχικά, θα μπορούσαν να πραγματοποιηθούν διαδικασίες βελτιστοποίησης που έχουν να κάνουν με τη κατάλληλη ρύθμιση των παραμέτρων και της αρχιτεκτονικής του δικτύου. Επιπλέον, επιπρόσθετες εξωτερικές μεταβλητές αλλά και αύξηση του αριθμού των διαθέσιμων δεδομένων προς επεξεργασία, θα είχαν τη δυνατότητα μείωσης του θορύβου και καλύτερης επεξήγησης των μη γραμμικών ιδιοτήτων του προβλήματος. Στην έρευνα αξιοποιήθηκαν μόνο μετεωρολογικά δεδομένα ενώ θα ήταν εξίσου αποτελεσματικό ένα σύνολο μεταβλητών που αναλύουν τα δομικά στοιχεία του κτιρίου αλλά και εσωτερικές καταστάσεις, όπως για παράδειγμα την πληρότητα των χώρων του δημαρχείου. Τέλος, θα μπορούσε να πραγματοποιηθεί η ανάπτυξη κάποιας μεθόδου που εκμεταλλεύεται τις μαθηματικές ιδιότητες και την ικανότητα μοντελοποίησης πολύπλοκων σχέσεων των δικτύων ακτινικής συνάρτησης βάσης (RBF, Radial Basis Function) [57] [58] [59] [60] [61] [62] [63] [64], σε συνδυασμό με την ικανότητα των δικτύων RNN στην προσέγγιση χρονοσειρών, δημιουργώντας έτσι υβριδικά δίκτυα RBF – RNN.

---

**ΒΙΒΛΙΟΓΡΑΦΙΑ – ΑΝΑΦΟΡΕΣ – ΔΙΑΔΙΚΤΥΑΚΕΣ ΠΗΓΕΣ**

---

- [1] Alexandridis, A.P., C.I. Siettos, H.K. Sarimveis, A.G. Boudouvis, G.V. Bafas, “Modeling of nonlinear process dynamics using Kohonen’s Neural Networks, Fuzzy Systems and Chebyshev Series”, *Computers and Chemical Engineering*, 26 (2002), pp. 479–486.
- [2] Alexandridis, A., M. Stogiannos, A. Kyriou, H. Sarimveis, “An offset-free neural controller based on approximating the inverse process dynamics” *Journal of Process Control*, 23(7) (2013), pp. 968–979.
- [3] Stogiannos, M., A. Alexandridis, H. Sarimveis, “Model predictive control for systems with fast dynamics using inverse neural models”, *ISA Transactions*, 72 (2018), pp. 161-177
- [4] Giamarellos, N., M. Papadimitrakis, M. Stogiannos, E.N. Zois, N.A. Livanos, A. Alexandridis, “A Machine Learning Model Ensemble for Mixed Power Load Forecasting across Multiple Time Horizons”, *Sensors*, 23(12) (2023) 5436.
- [5] Kagkas D., D. Karamichailidou, A. Alexandridis, “Chess position evaluation using radial basis function neural networks”, *Complexity*, (2023) 7143943.
- [6] Karamichailidou D., V. Kaloutsas, A. Alexandridis, “Wind Turbine Power Curve Modeling Using Radial Basis Function Neural Networks and Tabu Search”, *Renewable Energy*, 163 (2021), 2137-2152.
- [7] "Climate change," *Nature*, vol. 479, no. 7373, pp. 267-268, 2011. [Online]. Available: <https://doi.org/10.1038/479267b>.
- [8] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent Neural Networks for Time Series Forecasting: Current status and future directions," in *International Journal of Forecasting*, vol. 37, no. 1, pp. 388-427, 2021, ISSN 0169-2070, doi: 10.1016/j.ijforecast.2020.06.008.
- [9] Z. Wang, L. Xia, H. Yuan, R. S. Srinivasan, and X. Song, "Principles, research status, and prospects of feature engineering for data-driven building energy prediction: A comprehensive review," in *Journal of Building Engineering*, vol. 58, p. 105028, 2022, ISSN 2352-7102, doi: 10.1016/j.job.2022.105028.
- [10] H. Liu, J. Liang, Y. Liu, and H. Wu, “A Review of Data-Driven Building Energy Prediction,” *Buildings*, vol. 13, no. 2, p. 532, Feb. 2023, doi: 10.3390/buildings13020532.
- [11] M. Faiq, K. G. Tan, C. P. Liew, F. Hossain, C.-P. Tso, L. L. Lim, A. Y. K. Wong, and Z. M. Shah, "Prediction of energy consumption in campus buildings using long short-term

memory," in *Alexandria Engineering Journal*, vol. 67, pp. 65-76, 2023, ISSN 1110-0168, doi: 10.1016/j.aej.2022.12.015.

[12] W. Li and M. Wang, "Office Building Energy Consumption Prediction Based on LSTM and SVR," *2020 IEEE 3rd International Conference of Safe Production and Informatization (IICSPI)*, Chongqing City, China, 2020, pp. 171-176, doi: 10.1109/IICSPI51290.2020.9332384.

[13] S. Mahjoub, L. Chrifi-Alaoui, B. Marhic, and L. Delahoche, "Predicting Energy Consumption Using LSTM, Multi-Layer GRU and Drop-GRU Neural Networks," *Sensors*, vol. 22, no. 11, p. 4062, May 2022, doi: 10.3390/s22114062.

[14] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," in *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18-28, July-Aug. 1998, doi: 10.1109/5254.708428.

[15] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014, [Online]. Available: <https://jmlr.csail.mit.edu/papers/volume15/srivastava14a/srivastava14a.pdf>

[16] L. Cascone, S. Sadiq, S. Ullah, S. Mirjalili, H. U. R. Siddiqui, and M. Umer, "Predicting Household Electric Power Consumption Using Multi-step Time Series with Convolutional LSTM," in *Big Data Research*, vol. 31, p. 100360, 2023, ISSN 2214-5796, doi: 10.1016/j.bdr.2022.100360. [Online]. Available: <https://doi.org/10.1016/j.bdr.2022.100360>.

[17] N. Bhoj and R. S. Bhadoria, "Time-series based prediction for energy consumption of smart home data using hybrid convolution-recurrent neural network," in *Telematics and Informatics*, vol. 75, p. 101907, 2022, ISSN 0736-5853, doi: 10.1016/j.tele.2022.101907. [Online]. Available: <https://doi.org/10.1016/j.tele.2022.101907>.

[18] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," arXiv:1511.08458 [cs.NE], 2015.

[19] N. Somu, G. R. M. Raman, and K. Ramamritham, "A deep learning framework for building energy consumption forecast," in *Renewable and Sustainable Energy Reviews*, vol. 137, p. 110591, 2021, ISSN 1364-0321, doi: 10.1016/j.rser.2020.110591. [Online]. Available: <https://doi.org/10.1016/j.rser.2020.110591>.



- [20] X. Jin and J. Han, "K-Means Clustering," in Encyclopedia of Machine Learning, C. Sammut and G.I. Webb, Eds. Springer, Boston, MA, 2011, doi: 10.1007/978-0-387-30164-8\_425.
- [21] K. Krishnakumar, "Intelligent systems for aerospace engineering – an overview," NASA Technical Report, Document ID: 20030105746, 2003.
- [22] "intelligence," Aug. 16, 2023.  
<https://dictionary.cambridge.org/dictionary/english/intelligence>
- [23] M. Negnevitsky, Artificial Intelligence. Pearson Education, 2005.
- [24] J.T. M. Mitchell, Machine Learning. McGraw-Hill Science/Engineering/Math, 1997.
- [25] Β. Καμπουρλάζος, Γ. Παπακώστας, Εισαγωγή στην υπολογιστική νοημοσύνη: Μια ολιστική προσέγγιση. Κάλλιπος, Ανοιχτές Ακαδημαϊκές Εκδόσεις, 2015.
- [26] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An Introduction to Deep Reinforcement Learning," Foundations and Trends® in Machine Learning, vol. 11, no. 3–4, pp. 219–354, 2018, doi: 10.1561/22000000071.
- [27] S. Thrun, "Efficient Exploration In Reinforcement Learning," Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-CS-92-102, Jan. 1992.
- [28] J. Zhang, "Basic Neural Units of the Brain: Neurons, Synapses and Action Potential," arXiv:1906.01703 [q-bio.NC], Jun. 2019
- [29] S. S. Haykin, Neural Networks and Learning Machines. 2016.
- [30] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," Neurocomputing, vol. 503, pp. 92–108, Sep. 2022, doi: 10.1016/j.neucom.2022.06.111.
- [31] Lu, L., Yeonjong, S., Yanhui, Y., & Em Karniadakis, G. (2020). "Dying ReLU and Initialization: Theory and Numerical Examples." Communications in Computational Physics, 28(5), 1671-1706.
- [32] V. Kůrková, "Kolmogorov's theorem and multilayer neural networks," Neural Networks, vol. 5, no. 3, pp. 501-506, 1992.
- [33] A. Jadon, A. Patil, and S. Jadon, "A Comprehensive Survey of Regression Based Loss Functions for Time Series Forecasting," arXiv:2211.02989 [cs.LG], 2022.

- [34] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv:1609.04747 [cs.LG], 2017.
- [35] Y. Tian, Y. Zhang, and H. Zhang, "Recent Advances in Stochastic Gradient Descent in Deep Learning," *Mathematics*, vol. 11, no. 3, p. 682, Jan. 2023, doi: 10.3390/math11030682.
- [36] S. Khirirat, H. R. Feyzmahdavian and M. Johansson, "Mini-batch gradient descent: Faster convergence under data sparsity," *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Melbourne, VIC, Australia, 2017, pp. 2880-2887, doi: 10.1109/CDC.2017.8264077.
- [37] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv:1412.6980 [cs.LG], 2017.
- [38] S.W. Ellacott, "An Analysis of the Delta Rule," in *International Neural Network Conference*, 1990, pp. 145. [Online]. Available: [https://doi.org/10.1007/978-94-009-0643-3\\_145](https://doi.org/10.1007/978-94-009-0643-3_145)
- [39] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent Advances in Recurrent Neural Networks," arXiv:1801.01078 [cs.NE], 2018.
- [40] P. J. Werbos, "Backpropagation through time: what it does and how to do it," in *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550-1560, Oct. 1990, doi: 10.1109/5.58337.
- [41] R. M. Schmidt, "Recurrent Neural Networks (RNNs): A Gentle Introduction and Overview," arXiv:1912.05911 [cs.LG], 2019.
- [42] C. Tallec and Y. Ollivier, "Unbiasing Truncated Backpropagation Through Time," arXiv:1705.08209 [cs.NE], 2017.
- [43] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training Recurrent Neural Networks," arXiv preprint arXiv:1211.5063, 2013.
- [44] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie, "Why gradient clipping accelerates training: A theoretical justification for adaptivity," 2020, arXiv:1905.11881 [math.OC].
- [45] R. C. Staudemeyer and E. Rothstein Morris, "Understanding LSTM - A Tutorial into Long Short-Term Memory Recurrent Neural Networks," arXiv:1909.09586 [cs.NE], 2019.
- [46] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

- [47] K. Cho et al., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," arXiv:1406.1078 [cs.CL], 2014.
- [48] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," arXiv:1412.3555 [cs.NE], 2014.
- [49] N. M. Rezk, M. Purnaprajna, T. Nordström and Z. Ul-Abdin, "Recurrent Neural Networks: An Embedded Computing Perspective," in *IEEE Access*, vol. 8, pp. 57967-57996, 2020, doi: 10.1109/ACCESS.2020.2982416.
- [50] S. Raschka, J. Patterson, and C. Nolet, "Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence," *Information*, vol. 11, no. 4, p. 193, Apr. 2020, doi: 10.3390/info11040193.
- [51] M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," arXiv:1605.08695 [cs.DC], 2016.
- [52] F. Chollet, *Deep Learning with Python, Second Edition*. Simon and Schuster, 2021.
- [53] Siddharth Krishna Kumar, "On weight initialization in deep neural networks," 2017, arXiv:1704.08863 [cs.LG].
- [54] R. Adhikari and R. K. Agrawal, "An Introductory Study on Time Series Modeling and Forecasting," arXiv:1302.6613 [cs.LG], 2013.
- [55] J. Jose, "Introduction to Time Series Analysis and Its Applications," Aug. 2022.
- [56] A. Alexandridis, E. Chondrodima and H. Sarimveis, "Radial Basis Function Network Training Using a Nonsymmetric Partition of the Input Space and Particle Swarm Optimization," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 2, pp. 219-230, Feb. 2013, doi: 10.1109/TNNLS.2012.2227794.
- [57] Alexandridis, A., H. Sarimveis, K. Ninos, "RBF network training using a non-symmetric partition of the input space – Application to an MPC configuration", *Advances in Engineering Software*, 42(10) (2011), pp. 830-837.
- [58] Alexandridis, A., E. Chondrodima, H. Sarimveis, "Cooperative learning for radial basis function networks using particle swarm optimization", *Applied Soft Computing*, 49 (2016), pp. 485-497.

- [59] Alexandridis, A., E. Chondrodima, N. Giannopoulos, H. Sarimveis, “A Fast and Efficient Method for Training Categorical Radial Basis Function Networks”, *IEEE Transactions on Neural Networks and Learning Systems*, 28(11) (2017), pp. 2831 - 2836.
- [60] Alexandridis, A., M. Stogiannos, N. Papaioannou, E. Zois, H. Sarimveis, “An Inverse Neural Controller Based on the Applicability Domain of RBF Network Models”, *Sensors*, 18(1) (2018), 315.
- [61] Papadimitrakis, M., A. Alexandridis, “Active vehicle suspension control using road preview model predictive control and radial basis function networks”, *Applied Soft Computing*, 120C (2022), 108646.
- [62] Papadimitrakis, M. M. Stogiannos, H. Sarimveis, A. Alexandridis, “Multi-ship control and collision avoidance using MPC and RBF-based trajectory predictions”, *Sensors*, 21(21) (2021) 6959.
- [63] Karamichailidou, D., S. Koletsios, A. Alexandridis, “An RBF online learning scheme for non-stationary environments based on fuzzy means and Givens rotations”, ***Neurocomputing***, 501C (2022) 370-386.
- [64] Karamichailidou, D., A. Alexandridis, G. Anagnostopoulos, G. Syriopoulos, O. Sekkas, “Modeling biogas production from anaerobic wastewater treatment plants using radial basis function networks and differential evolution”, ***Computers and Chemical Engineering***, 157C (2022), 107629.
- [65] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into Deep Learning," Cambridge University Press, 2023. [Online]. Available: [\url{https://D2L.ai}](https://D2L.ai)