



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

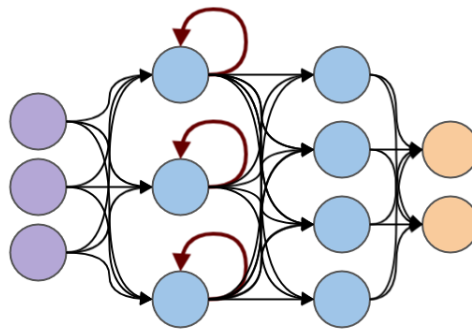
Ανάλυση χρονικών ακολουθιών με βαθιά μάθηση για αναγνώριση ανθρώπινων δραστηριοτήτων

Μελέτη και υλοποίηση

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΓΕΩΡΓΙΑ Ι. ΦΥΤΙΖΑ



Επιβλέπων: Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

Συνεπιβλέπουσα: Παρασκευή Τζούβελη
Εργαστηριακό και Διδακτικό Προσωπικό Ε.Μ.Π.

Αθήνα, Οκτώβριος 2021



Ανάλυση χρονικών ακολουθιών με βαθιά μάθηση για αναγνώριση ανθρώπινων δραστηριοτήτων

Μελέτη και υλοποίηση

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΓΕΩΡΓΙΑ Ι. ΦΥΤΙΖΑ

Επιβλέπων: Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις 18 Οκτωβρίου 2021.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....
Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

.....
Σταφυλοπάτης Ανδρέας-Γεώργιος
Καθηγητής Ε.Μ.Π.

.....
Στάμου Γεώργιος
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2021



(Υπογραφή)

.....
Γεωργία Φύτιζα

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

Copyright © Γεωργία Φύτιζα, 2021.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Ο υπολογισμός με επίκεντρο τον άνθρωπο είναι ένα αναπτυσσόμενο ερευνητικό πεδίο που στοχεύει στην κατανόηση της ανθρώπινης συμπεριφοράς και στην ενσωμάτωση των χρηστών και του κοινωνικού τους πλαισίου με τα συστήματα υπολογιστών. Μία από τις πιο πρόσφατες και ελκυστικές εφαρμογές σε αυτό το πλαίσιο αποτελεί η ανίχνευση της κίνησης του ανθρώπινου σώματος χρησιμοποιώντας κινητά τηλέφωνα ή έξυπνες συσκευές για τη συλλογή πληροφοριών περιβάλλοντος σχετικά με τις ενέργειες των ανθρώπων. Η ικανότητα ενός συστήματος να χρησιμοποιεί όσο το δυνατόν λιγότερους πόρους για την αναγνώριση της δραστηριότητας ενός χρήστη από ακατέργαστα δεδομένα είναι αυτό για το οποίο προσπαθούν πολλοί ερευνητές. Πλέον τέτοιες συσκευές είναι πανταχού παρούσες στην καθημερινότητα μας και οι πληροφορίες που αντλούνται μπορούν να χρησιμοποιηθούν σε αμέτρητες εφαρμογές. Η επιστήμη των Βαθιών Νευρωνικών Δικτύων ασχολείται με τέτοια προβλήματα και προτείνει λύσεις βασισμένες σε σύγχρονες αρχιτεκτονικές Αναδρομικών δικτύων.

Στόχος της διπλωματικής εργασίας είναι η εύρεση αρχιτεκτονικών οι οποίες να καλύπτουν το πρόβλημα της ταξινόμησης των δραστηριοτήτων καθώς επίσης και η ανάλυση και η επεξεργασία ακολουθιακών δεδομένων.

Λέξεις Κλειδιά

Νευρωνικά Δίκτυα, Ακολουθιακά δεδομένα, Χρονοσειρές, Αισθητήρες, HAR, smartphone, RNN, TCN, GRU, LSTM

Abstract

Human-centered computing is an emerging research field that aims to understand human behavior and integrate users and their social context with computer systems. One of the most recent, challenging and appealing applications in this framework consists in sensing human body motion using smartphones to gather context information about people actions. The ability for a system to use as few resources as possible to recognize a user's activity from raw data is what many researchers are striving for. Smart devices are ubiquitous in our daily lives and they provide information which could be used in countless applications. The field of Deep Neural Networks deals with such applications and suggests solutions based on modern recurrent neural networks.

This diploma thesis aims to find architectures that should cover the classification problem of human action recognition as well as the analysis and processing of sequential data.

Keywords

Neural Networks, Sequential data, Time-series, Smartphone, HAR, RNN, GRU, LSTM, TCN

στους γονείς μου

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον καθηγητή κ.Στέφανο Κόλλια για την εμπιστοσύνη που μου έδειξε και για την ευκαιρία που μου έδωσε να εκπονήσω την διπλωματική μου στο εργαστήριο των Νευρωνικών Δικτύων. Επίσης θα ήθελα να ευχαριστώ ιδιαίτερα την Δρ.Παρασκευή Τζούβελη για την πολύτιμη καθοδήγησή της και την εξαιρετική συνεργασία που είχαμε καθ' όλη τη διάρκεια της εκπόνησης της διπλωματικής. Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου για την στήριξη και την ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια.

Αθήνα, Οκτώβριος 2021

Γεωργία Φύτιζα

Περιεχόμενα

Περίληψη	1
Abstract	3
Ευχαριστίες	7
1 Εισαγωγή	17
1.1 Αντικείμενο της διπλωματικής	18
1.2 Οργάνωση του τόμου	19
I Θεωρητικό Μέρος	21
2 Χρονοσειρές	23
2.1 Εισαγωγή στις χρονοσειρές	23
2.1.1 Ανάλυση χρονοσειρών	23
2.2 Ποιοτικά χαρακτηριστικά χρονοσειρών	24
2.2.1 Κυκλικότητα (cycle)	24
2.2.2 Εποχικότητα (seasonality)	25
2.2.3 Ασυνέχειες	25
2.3 Στατιστική ανάλυση χρονοσειρών	27
2.3.1 Μέση τιμή (Average)	27
2.3.2 Τυπική απόκλιση (Standard Deviation)	27
2.3.3 Διακύμανση (Variance)	27
2.3.4 Συνδιακύμανση (Covariance)	27
2.3.5 Συντελεστής αυτοσυσχέτισης (Autocorrelation Coefficient)	28
3 Νευρωνικά Δίκτυα	29
3.1 Αναδρομικά Νευρωνικά Δίκτυα	29
3.1.1 Εισαγωγή στα Αναδρομικά Νευρωνικά Δίκτυα	29
3.1.2 Απλά Αναδρομικά Νευρωνικά Δίκτυα	30
3.1.3 Αλγόριθμος Back Propagation Διαμέσω χρόνου (BPTT)	33
3.1.4 Αδυναμίες Αναδρομικών Νευρωνικών Δικτύων	35
3.2 Επεκτάσεις των Αναδρομικών Νευρωνικών Δικτύων	36
3.2.1 Φραγμένα Αναδρομικά Δίκτυα - GRUs	37
3.2.2 Δίκτυα Μακράς Βραχυπρόθεσμης Μνήμης - LSTM	40
3.3 Υβριδικά Νευρωνικά Δίκτυα	43

3.3.1	Εισαγωγή στα Συνελκτικά Νευρωνικά Δίκτυα	44
3.3.2	Συνελκτικά Δίκτυα Μιας Διάστασης (1D CNNs)	48
3.3.3	Επίπεδο Εγκατάλειψης	50
3.3.4	Ζεύξη Επαναληπτικών και Συνελκτικών Νευρωνικών Δικτύων (CNN-RNNs)	50
3.4	Χρονικά Συνελκτικά Δίκτυα (Temporal Convolutional Network)	52
3.4.1	Αιτώδης συνέλιξη (Causal Convolution)	52
3.4.2	Διαστολή (Dilation)	53
3.4.3	Υπολειμματικά μπλοκ (Residual Blocks)	56
II	Πρακτικό Μέρος	59
4	Περιγραφή και Ανάλυση δεδομένων	61
4.1	Περιγραφή Συνόλου Δεδομένων	61
4.1.1	Μεθοδολογία	61
4.1.2	Επεξεργασία Σημάτων	62
4.1.3	Αντιστοίχιση χαρακτηριστικών	63
4.2	Ανάλυση και Οπτικοποίηση Δεδομένων	64
4.2.1	Ισορροπία Δραστηριοτήτων	64
4.2.2	Χρονική σειρά δεδομένων ανά Υποκείμενο	65
4.2.3	Ιστογράμματα ανά Υποκείμενο	67
4.2.4	Ιστογράμματα ανά δραστηριότητα	69
4.2.5	Διάρκεια δραστηριοτήτων	71
5	Εφαρμογή μεθόδων ταξινόμησης και αποτελέσματα	75
5.1	Αρχιτεκτονική Αναδρομικών Δικτύων	75
5.1.1	Επίπεδο αναδρομικού νευρωνικού δικτύου	75
5.1.2	Επίπεδο εγκατάλειψης	75
5.1.3	Πυκνό Επίπεδο	76
5.1.4	Φόρτωση δεδομένων	76
5.2	Υβριδική Αρχιτεκτονική	77
5.2.1	Φόρτωση δεδομένων	77
5.2.2	Κατανομή Χρόνου	77
5.2.3	Υπόλοιπα επίπεδα	78
5.3	Πειραματικά αποτελέσματα	79
5.3.1	Εφαρμογή RNN νευρωνικού δικτύου	79
5.3.2	Εφαρμογή GRU νευρωνικού δικτύου	79
5.3.3	Εφαρμογή LSTM νευρωνικών δικτύων	81
5.3.4	Εφαρμογή TCN νευρωνικών δικτύων	82
5.3.5	Εφαρμογή CNN + RNN αρχιτεκτονικής	83
5.3.6	Εφαρμογή CNN + GRU αρχιτεκτονικής	84
5.3.7	Εφαρμογή CNN+LSTM αρχιτεκτονικής	86
5.3.8	Εφαρμογή CNN+TCN αρχιτεκτονικής	87

III Επίλογος	89
6 Επίλογος	91
6.1 Σύνοψη και συμπεράσματα	91
6.2 Μελλοντικές Επεκτάσεις	92
Βιβλιογραφία	98

Κατάλογος Σχημάτων

2.1	Γραφική παράσταση χρονοσειράς με ανοδική τάση/κλίση [1]	24
2.2	Οι μηνιαίες πωλήσεις κατοικιών [2]	25
2.3	Η μηνιαία παραγωγή γάλατος ανά αγελάδα [3]	26
2.4	Παραδείγματα χρονοσειρών με ασυνεχή συμπεριφορά [4]	26
3.1	Διαφορά μεταξύ Hidden Layer και Hidden State [5]	30
3.2	Απλό Αναδρομικό Νευρωνικό Δίκτυο με κρυφή κατάσταση [6].	32
3.3	Υπολογιστικό γράφημα που αναδεικνύει τις εξαρτήσεις για ένα μοντέλο RNN σε τρία χρονικά βήματα. Τα πλαίσια αντιπροσωπεύουν μεταβλητές (όχι σκιασμένα) ή παραμέτρους (σκιασμένα) και οι κύκλοι αντιπροσωπεύουν τελεστές πράξεων [7].	34
3.4	Εικονική αναπαράσταση της βραχυπρόθεσμης μνήμης των RNNs [5]	36
3.5	Αναδρομικό Νευρωνικό Δίκτυο με GRUs μονάδες [8]	37
3.6	Αναλυτική σχηματική απεικόνιση μιας GRU μονάδας [8]	38
3.7	Σχηματική απεικόνιση της πύλης ενημέρωσης [8]	38
3.8	Σχηματική απεικόνιση της πύλης επαναφοράς [8]	39
3.9	Σχηματική απεικόνιση για Current Memory Content [8]	39
3.10	Σχηματική απεικόνιση για Hidden State [8]	40
3.11	Η απεικόνιση της sigmoid function και η μαθηματική αναπαράσταση [9]	41
3.12	Σχηματική απεικόνιση για Forget Gate και η μαθηματική αναπαράσταση [10]	41
3.13	Σχηματική απεικόνιση για Input Gate και η μαθηματική αναπαράσταση [10]	42
3.14	Σχηματική απεικόνιση για Cell State και η μαθηματική αναπαράσταση [10]	42
3.15	Σχηματική απεικόνιση για Hidden State και η μαθηματική αναπαράσταση [10]	43
3.16	Σχηματική απεικόνιση του υπολογισμού της συνέλιξης [11]	45
3.17	Σχηματική απεικόνιση του υπολογισμού της υποδειγματοληψίας μεγίστου [12]	46
3.18	Παράδειγμα υπολογισμού συνελκτικού επιπέδου με τεχνική γεμίματος. [13]	47
3.19	Παράδειγμα υπολογισμού συνελκτικού επιπέδου με οριζόντιο βήμα 2 και κάθετο 3. [13]	48
3.20	Υπολογισμός συνέλιξης για 1D-CNN με input channels = output channels = 1 [14]	49
3.21	Υπολογισμός συνέλιξης για 1D-CNN με input channels > 1 [14]	50
3.22	Κρυμένο επίπεδο πριν και μετά το επίπεδο εγκατάλειψης [15]	51
3.23	Δομή CNN-RNNs	52
3.24	Παράδειγμα Causal Convolution [16]	53
3.25	Δίκτυο με διασταλμένα συνελκτικά στρώματα με πλήρη κάλυψη [14]	55

3.26	Δίκτυο με διασταλμένα συνελκτικά στρώματα με 'τύμπες' [14]	56
3.27	Ένα αιτιατό μονοδιάστατο συνελκτικό επίπεδο πριν την αλλαγή [14]	57
3.28	Ένα υπολειμματικό μπλοκ μετά την αλλαγή [14]	57
3.29	Τελικό TCN μοντέλο [14]	58
4.1	Λεπτομέρειες του πρωτοκόλλου δραστηριοτήτων [17]	62
4.2	Σήματα πεδίου χρόνου και συχνότητας που λαμβάνονται από τους αισθητήρες smartphone [17]	63
4.3	Λίστα μετρικών για τον υπολογισμό διανυσμάτων χαρακτηριστικών. [17]	64
4.4	Η ισορροπία των κλάσεων των δραστηριοτήτων του dataset μας	65
4.5	Γραφική παράσταση χρονικής σειράς δεδομένων ενός υποκειμένου	66
4.6	Γραφική παράσταση χρονικής σειράς δεδομένων ενός δεύτερου υποκειμένου	67
4.7	Ιστόγραμμα συνολικής επιτάχυνσης για 10 υποκείμενα	68
4.8	Ιστόγραμμα επιτάχυνσης σώματος για 10 υποκείμενα	69
4.9	Ιστόγραμμα για τα δεδομένα από το γυροσκόπιο για 10 υποκείμενα	69
4.10	Ιστόγραμμα ανά δραστηριότητα για την συνολική επιτάχυνση ενός υποκειμένου	70
4.11	Ιστόγραμμα ανά δραστηριότητα για την επιτάχυνση σώματος ενός υποκειμένου	71
4.12	Ιστόγραμμα ανά δραστηριότητα για την γωνιακή ταχύτητα ενός υποκειμένου	71
4.13	Παράδειγμα boxplot [18].	72
4.14	Boxplot για την διάρκεια των δραστηριοτήτων ανά υποκείμενο στο training set.	73
4.15	Boxplot για την διάρκεια των δραστηριοτήτων ανά υποκείμενο στο test set.	73
5.1	Αρχιτεκτονική με Time Distributed επίπεδα [19]	77

Κατάλογος Πινάκων

5.1	Πίνακας υπερπαραμέτρων RNN αρχιτεκτονικής	79
5.2	Πίνακας σύγχυσης RNN αρχιτεκτονικής	80
5.3	Πίνακας υπερπαραμέτρων GRU αρχιτεκτονικής	80
5.4	Πίνακας σύγχυσης GRU αρχιτεκτονικής	81
5.5	Πίνακας υπερπαραμέτρων LSTM αρχιτεκτονικής	81
5.6	Πίνακας σύγχυσης LSTM αρχιτεκτονικής	82
5.7	Πίνακας υπερπαραμέτρων TCN αρχιτεκτονικής	83
5.8	Πίνακας σύγχυσης TCN αρχιτεκτονικής	83
5.9	Πίνακας υπερπαραμέτρων CNN+RNN αρχιτεκτονικής	84
5.10	Πίνακας σύγχυσης CNN + RNN αρχιτεκτονικής	85
5.11	Πίνακας υπερπαραμέτρων CNN+GRU αρχιτεκτονικής	85
5.12	Πίνακας σύγχυσης CNN + GRU αρχιτεκτονικής	86
5.13	Πίνακας υπερπαραμέτρων CNN+LSTM αρχιτεκτονικής	86
5.14	Πίνακας σύγχυσης CNN + LSTM αρχιτεκτονικής	87
5.15	Πίνακας υπερπαραμέτρων CNN + TCN αρχιτεκτονικής	88
5.16	Πίνακας σύγχυσης CNN + TCN αρχιτεκτονικής	88

Κεφάλαιο 1

Εισαγωγή

Μέχρι πρόσφατα, σχεδόν κάθε πρόγραμμα με το οποίο αλληλεπιδρούσαμε, αποτελούταν από κώδικα, γραμμένο από μηχανικούς λογισμικού και στηριγμένο σε κάποιες πρώτες αρχές. Ας πούμε ότι θέλουμε να δημιουργήσουμε μια εφαρμογή για τη διαχείριση μιας πλατφόρμας ηλεκτρονικού εμπορίου. Αφού κάτσουμε πίσω από έναν πίνακα για μερικές ώρες για να σκεφτούμε το πρόβλημα, θα καταλήξουμε σε μια λύση για την δεδομένη εργασία που να μοιάζει κάπως έτσι: οι χρήστες θα αλληλεπιδρούν με την εφαρμογή μας μέσω μιας διεπαφής που εκτελείται σε ένα πρόγραμμα περιήγησης ιστού ή εφαρμογή για κινητά, η εφαρμογή μας θα αλληλεπιδρά με μια μηχανή βάσης δεδομένων εμπορικής ποιότητας για να παρακολουθεί την κατάσταση κάθε χρήστη και να διατηρεί αρχεία ιστορικών συναλλαγών και τέλος στο επίκεντρο της εφαρμογής μας, θα υπάρχει η επιχειρησιακή λογική (ή ο εγκέφαλος της εφαρμογής μας) που θα αναφέρει μεθοδικά τις κατάλληλες ενέργειες που πρέπει να ακολουθήσει το πρόγραμμά μας σε κάθε πιθανή περίπτωση.

Για να δημιουργήσουμε τον εγκέφαλο της εφαρμογής μας, θα πρέπει να περάσουμε από κάθε πιθανή γωνιακή υπόθεση που αναμένουμε να συναντήσουμε, επινοώντας κατάλληλους κανόνες. Η ικανότητά μας αυτή να σχεδιάζουμε αυτοματοποιημένα συστήματα από τις πρώτες αρχές που οδηγούν σε λειτουργικά προϊόντα και συστήματα, είναι ένα αξιοσημείωτο γνωστικό επίτευγμα και όταν είμαστε σε θέση να επινοήσουμε λύσεις που λειτουργούν 100%, δεν πρέπει να χρησιμοποιούμε μηχανική μάθηση.

Ευτυχώς για την αναπτυσσόμενη κοινότητα των επιστημόνων μηχανικής μάθησης, πολλές εργασίες που θα θέλαμε να αυτοματοποιήσουμε δεν υποκλίνονται τόσο εύκολα στην ανθρώπινη εφευρετικότητα. Ας φανταστούμε να καθόμασταν πάλι πίσω από τον αρχικό πίνακα με τα πιο έξυπνα μυαλά που γνωρίζουμε, αλλά αυτή τη φορά να αντιμετωπίζαμε ένα πρόβλημα της μορφής: "Γράψτε ένα πρόγραμμα που προβλέπει τον αυριανό καιρό με δεδομένα γεωγραφικές πληροφορίες και δορυφορικές εικόνες". Σε αυτή την περίπτωση, ακόμη και η ελίτ των προγραμματιστών δεν μπορούν να κωδικοποιήσουν λύσεις βασισμένοι στις πρώτες αρχές που δίνονται. Αυτό μπορεί να οφείλεται σε πολλούς και διαφόρους λόγους. Μερικές φορές το πρόγραμμα που ψάχνουμε ακολουθεί ένα μοτίβο που αλλάζει με την πάροδο του χρόνου και χρειαζόμαστε τα προγράμματα μας να προσαρμοστούν. Σε άλλες περιπτώσεις, οι σχέσεις μεταξύ των δεδομένων του προβλήματος μας μπορεί να είναι πολύ περίπλοκες, απαιτώντας χιλιάδες ή εκατομμύρια υπολογισμούς που είναι πέρα από τη συνειδητή κατανόησή μας. Τέτοιου είδους προβλήματα επιλύονται με αλγόριθμους και τεχνικές μηχανικής μάθησης.

Η μηχανική μάθηση είναι η μελέτη ισχυρών τεχνικών που μπορούν να μάθουν από την εμπειρία. Καθώς ο αλγόριθμος μηχανικής μάθησης συσσωρεύει περισσότερη εμπειρία, συνήθως με τη μορφή δεδομένων παρατήρησης ή αλληλεπιδράσεων με ένα περιβάλλον, η απόδοσή του βελτιώνεται.

Αντιπαραβάλλοντας τις εφαρμογές που βρίσκει η μηχανική μάθηση με την πλατφόρμα ηλεκτρονικού εμπορίου, η οποία λειτουργεί σύμφωνα με την ίδια επιχειρησιακή λογική, ανεξάρτητα από το πόση εμπειρία συγκεντρώνεται, συνειδητοποιούμε ότι βασισμένοι στις τεχνικές της μηχανικής μάθησης μπορούμε να δημιουργήσουμε χρήσιμες εφαρμογές που θα βελτιώσουν ακόμα περισσότερο τις καθημερινές μας συνήθειες.

1.1 Αντικείμενο της διπλωματικής

Το βασικό ζήτημα που θα μελετήσουμε στα πλαίσια αυτής της διπλωματικής είναι πως μέσω των τεχνικών της βαθιάς μάθησης μπορούμε να αναλύσουμε χρονικές ακολουθίες. Στόχος αυτής της ανάλυσης είναι να μπορέσουμε να κατηγοριοποιήσουμε αυτές τις χρονικές ακολουθίες.

Η μελέτη αυτή χρησιμοποιήθηκε για την ταξινόμηση ανθρώπινων δραστηριοτήτων. Πιο συγκεκριμένα, έχοντας σαν δεδομένα εισόδου τιμές από συγκεκριμένους αισθητήρες από smartphones μέσα σε δεδομένα χρονικά παράθυρα, χρησιμοποιήσαμε διάφορα βαθιά νευρωνικά δίκτυα, προκειμένου να ταξινομηθούν 6 ανθρώπινες δραστηριότητες. Σκοπός δεν είναι μόνο η εύρεση ενός βέλτιστου μοντέλου ταξινόμησης για το παρόν πρόβλημα, αλλά και η κατανόηση των διαχρονικών και διαδομένων βαθιών νευρωνικών δικτύων.

Για να κατανοήσουμε την ανθρώπινη συμπεριφορά και να προβλέψουμε τις ανθρώπινες κινήσεις, η έρευνα για την αναγνώριση ανθρώπινης δραστηριότητας (HAR), χρησιμοποιώντας αισθητήρες σε συσκευές που κρατιούνται στο χέρι και φορητές συσκευές, έχει ενταθεί. Η ικανότητα ενός συστήματος να χρησιμοποιεί όσο το δυνατόν λιγότερους πόρους για να αναγνωρίσει τη δραστηριότητα ενός χρήστη από ακατέργαστα δεδομένα είναι αυτό για το οποίο προσπαθούν πολλοί ερευνητές. Το σύνολο δεδομένων που χρησιμοποιήσαμε σε αυτή την διπλωματική αρχικά μοντελοποιήθηκε από αλγόριθμους μηχανικής μάθησης το 2012 στην μελέτη που έγινε από τους Davide Anguita et al [20]. Αυτή η μέθοδος προσαρμόζει το πρότυπο Διανυσμάτων Υποστήριξης (SVM) και αξιοποιεί την αριθμητική σταθερού σημείου για υπολογιστική μείωση κόστους, με ακρίβεια προβλέψεων ίση με 89%. Σε πιο πρόσφατη δημοσίευση [21], προτάθηκε ένα υβριδικό μοντέλο (CNN + LSTM) για να προσεγγίσει αυτό το πρόβλημα ταξινόμησης. Αυτή η προσέγγιση σημειώνει 92% ακρίβεια στις ταξινομήσεις της.

Η ανάλυση και μερική οπτικοποίηση ακολουθιακών δεδομένων είναι επίσης ένα θέμα που μελετήθηκε στην παρούσα διπλωματική. Η ανάλυση που έγινε βασίστηκε στα χαρακτηριστικά του συνόλου δεδομένων μας και στόχευε στη βαθύτερη κατανόηση του συνόλου καθώς και στην ευκολότερη μοντελοποίηση του.

Τέλος, πραγματοποιήθηκαν αρκετά πειράματα πάνω στο σύνολο δεδομένων προκειμένου να βρεθεί η βέλτιστη αρχιτεκτονική για το παρόν πρόβλημα. Επιπλέον, στόχος των πειραμάτων είναι και η μελέτη διαφόρων αρχιτεκτονικών με σκοπό την σύγκρισή τους και τον εντοπισμό των ποιοτικών τους διαφορών.

1.2 Οργάνωση του τόμου

Η εργασία αυτή είναι οργανωμένη σε πέντε κεφάλαια : Στο Κεφάλαιο 2 δίνεται το θεωρητικό υπόβαθρο των βασικών χαρακτηριστικών που σχετίζονται με τις χρονοσειρές καθώς επίσης και οι στατιστικοί ορισμοί που χρησιμοποιούνται για την ανάλυση των χρονοσειρών. Στο Κεφάλαιο 3 αρχικά περιγράφονται τα απλά αναδρομικά νευρωνικά δίκτυα (RNNs) και οι πιο διαδεδομένες παραλλαγές τους και στη συνέχεια περιγράφονται οι υβριδικές αρχιτεκτονικές και τα δίκτυα TCN. Στο Κεφάλαιο 4 παρουσιάζεται η ανάλυση και η περιγραφή του συνόλου δεδομένων και παρατίθενται σχετικά διαγράμματα. Τέλος, η περιγραφή της υλοποίησης με την ανάλυση των βασικών αρχιτεκτονικών και τις λεπτομέρειες σχετικά με τις πλατφόρμες και τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν δίνονται στο Κεφάλαιο 5.

Μέρος I

Θεωρητικό Μέρος

Κεφάλαιο 2

Χρονοσειρές

Στο κεφάλαιο αυτό παρουσιάζεται η έννοια της χρονοσειράς και τα ποιοτικά χαρακτηριστικά της. Επιπλέον, αναφέρονται κάποιοι ορισμοί χρήσιμοι για την στατιστική ανάλυση των χρονοσειρών.

2.1 Εισαγωγή στις χρονοσειρές

Ως χρονοσειρά ορίζουμε ένα σύνολο διαδοχικών παρατηρήσεων Y_t με τιμές που αντιπροσωπεύουν κάποιο φυσικό μέγεθος, οι οποίες έχουν καταγραφεί σε μια χρονική περίοδο. Αυτές οι παρατηρήσεις καταγράφονται είτε για ένα ολόκληρο χρονικό διάστημα, είτε για ένα τυχαίο κομμάτι του διαστήματος αυτού, είτε για μεμονωμένα χρονικά σημεία. Επιπλέον, οι παρατηρήσεις μπορεί να λαμβάνονται με συνέχεια, όπως η καταγραφή της θερμοκρασίας ή με διακριτό τρόπο όπως ο αριθμός της επισκεψιμότητας ενός μουσείου. Στα πλαίσια της παρούσας εργασίας, θα ασχοληθούμε με διακριτές παρατηρήσεις, οι οποίες έγιναν σε σταθερά χρονικά διαστήματα μήκους 2,56 δευτερολέπτων. Μαθηματικά μπορούμε να ορίσουμε μια χρονοσειρά ως την σειρά Y_t , για μια συνεχή χρονική διάρκεια, δηλαδή $(Y_t)_{t=0, \pm 1, \pm 2, \dots}$, όπου υπάρχουν παρατηρήσεις μόνο για τις χρονικές στιγμές $t = 1, \dots, n$. Έτσι παρατηρούμε τα (Y_1, \dots, Y_n) ή $(Y_t), t \in \mathbb{Z}, (\mathbb{Z} = \dots, 0, 1, \dots)$. Μπορούμε να αναφερθούμε στον δείκτη t του Y_t , ως τον χρόνο και να θεωρήσουμε το Y_t , ως την κατάσταση ή την έξοδο ενός στοχαστικού συστήματος τη χρονική στιγμή t . Η ερμηνεία του δείκτη t ως τον χρόνο που πραγματοποιείται μια παρατήρηση δεν επηρεάζει τη μαθηματική επεξεργασία της χρονοσειράς, η οποία ασχολείται κυρίως με την κοινή κατανομή των μεταβλητών, όμως αυτή η διάταξη των μεταβλητών στο χρόνο είναι συνήθως πολύ σημαντική [22].

2.1.1 Ανάλυση χρονοσειρών

Η ανάλυση χρονοσειρών είναι μια προσέγγιση για την ανάλυση δεδομένων χρονοσειρών με σκοπό την εξαγωγή σημαντικών χαρακτηριστικών από τα δεδομένα τους και τη δημιουργία άλλων χρήσιμων πληροφοριών που χρησιμοποιούνται από διάφορες επιχειρήσεις σε πολλές εφαρμογές.

Η ανάλυση χρονικών ακολουθιών βοηθά στην κατανόηση των χρονικών προτύπων ενός συνόλου μετρικών σημείων δεδομένων, το οποία θεωρούνται κρίσιμης σημασίας για πολλές επιχειρήσεις. Οι τεχνικές πρόβλεψης χρονοσειρών θα μπορούσαν να απαντήσουν σε επι-

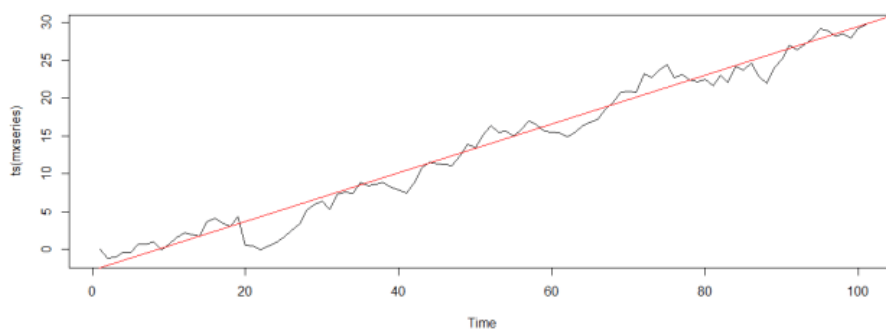
χειρηματικές ερωτήσεις, όπως για παράδειγμα για το πόση επισκεψιμότητα περιμένετε στο ηλεκτρονικό σας κατάστημα ή για το πόσα προϊόντα θα πωληθούν τον επόμενο μήνα. Όλα αυτά αποτελούν σημαντικά προβλήματα χρονικών σειρών για επίλυση. Ο βασικός στόχος της ανάλυσης των χρονικών σειρών είναι συνήθως ο προσδιορισμός ενός μοντέλου το οποίο θα περιγράψει το πρότυπο της χρονικής σειράς και θα μπορούσε να χρησιμοποιηθεί για πρόβλεψη.

2.2 Ποιοτικά χαρακτηριστικά χρονοσειρών

Η ορθή μελέτη μιας χρονοσειράς πραγματοποιείται μέσω της επισκόπησής της στο πεδίο του χρόνου. Για την κατανόηση των βασικών χαρακτηριστικών των χρονοσειρών, μπορούμε να αναλύσουμε τη χρονοσειρά στα βασικά της στοιχεία τα οποία είναι : η τάση, η κυκλικότητα, η εποχικότητα, όπως και οι ασυνέχειες (οι οποίες αποτελούν υποσύνολο των μη κανονικών διακυμάνσεων) [23].

Τάση (trend)

Ως τάση ορίζεται η μακροπρόθεσμη μεταβολή του μέσου επιπέδου των τιμών μιας χρονοσειράς. Εστιάζοντας παραπάνω στον όρο μακροπρόθεσμη μεταβολή, για τον επιτυχή προσδιορισμό της τάσης χρειάζεται να υπάρχει επαρκής αριθμός παρατηρήσεων, καθώς επίσης και να έχει οριστεί κατάλληλα το μήκος της περιόδου μέσα στην οποία θα μελετηθεί η τάση. Η τάση μπορεί να χαρακτηριστεί ως ανοδική, πτωτική ή μηδενική/σταθερή, ενώ σε κάθε περίπτωση αποτελεί μια γενική απεικόνιση της χρονοσειράς. Με άλλα λόγια, παρατηρείται μια τάση όταν υπάρχει μια αυξανόμενη ή μειούμενη κλίση στις χρονοσειρές. Για παράδειγμα, όταν βγαίνει στην αγορά κάποιο νέο κινητό, μετατρέπεται σε τάση για λίγο (ανοδική) και μετά οι πωλήσεις του πέφτουν και σταδιακά εξαφανίζεται (πτωτική). Στο σχήμα 2.1, παρουσιάζεται μια χρονοσειρά, η οποία παρουσιάζει ξεκάθαρη ανοδική τάση.

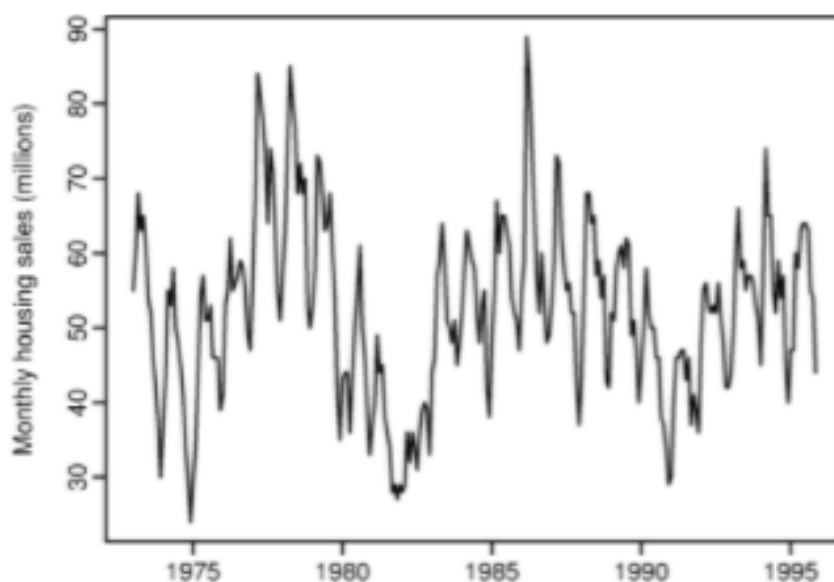


Σχήμα 2.1: Γραφική παράσταση χρονοσειράς με ανοδική τάση/κλίση [1]

2.2.1 Κυκλικότητα (cycle)

Η κυκλικότητα περιγράφεται σαν μια κυματοειδής μεταβολή (ανοδική ή πτωτική), η οποία εμφανίζεται σε περιόδους και συνήθως οφείλεται σε εξωτερικούς παράγοντες. Αυτή η

κυκλική συμπεριφορά, μπορεί να μεταφερθεί με την πάροδο του χρόνου επειδή ο χρόνος μεταξύ των περιόδων δεν είναι ακριβής. Για παράδειγμα, η χρηματιστηριακή αγορά τείνει να κάνει κύκλους μεταξύ περιόδων υψηλών και χαμηλών τιμών, αλλά δεν υπάρχει καθορισμένος χρόνος μεταξύ αυτών των διακυμάνσεων. Στο σχήμα 2.2, απεικονίζονται οι μηνιαίες πωλήσεις κατοικιών την περίοδο 1975 με 1995 όπου μπορούμε να παρατηρήσουμε ένα κυκλικό μοτίβο να επαναλαμβάνεται ανά 5ετία.



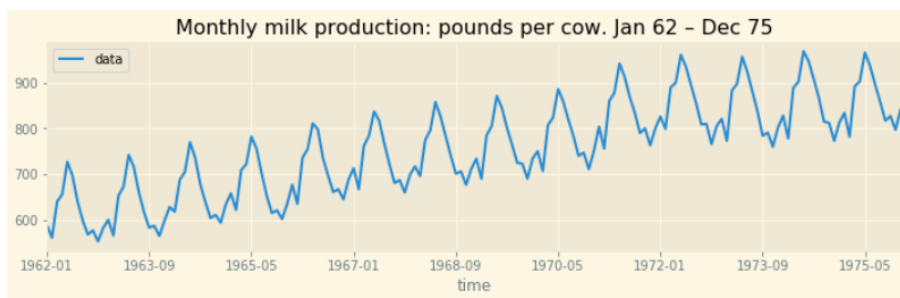
Σχήμα 2.2: Οι μηνιαίες πωλήσεις κατοικιών [2]

2.2.2 Εποχικότητα (seasonality)

Η εποχικότητα μπορεί να οριστεί ως μια περιοδική διακύμανση που έχει σταθερό μήκος μικρότερο του ενός έτους. Η συγκεκριμένη διακύμανση είναι άμεσα κατανοητή και προβλέψιμη, αφού επαναλαμβάνεται με το ίδιο μοτίβο σε σχέση με το χρόνο, γεγονός που την κάνει εύκολα εξηγήσιμη. Η κύρια διαφορά μεταξύ εποχικής και κυκλικής συμπεριφοράς έχει να κάνει με το πόσο τακτική είναι η περίοδος αλλαγής. Μια εποχιακή συμπεριφορά είναι αυστηρά τακτική, πράγμα που σημαίνει ότι υπάρχει ένα ακριβές χρονικό διάστημα μεταξύ των κορυφών και των γωνιών των δεδομένων. Για παράδειγμα, η θερμοκρασία θεωρείται ότι θα έχει εποχιακή συμπεριφορά καθώς είναι πιθανό η πιο κρύα μέρα του έτους και η πιο ζεστή μέρα του έτους να μετακινηθούν χρονικά, αλλά δεν πρόκειται να δούμε ποτέ μια μετατόπιση με την πάροδο του χρόνου τέτοια ώστε τελικά ο χειμώνας να έρχεται τον Ιούνιο στο βόρειο ημισφαίριο. Στο σχήμα 2.3, απεικονίζεται η μηνιαία παραγωγή γάλατος ανά αγελάδα από τον Ιανουάριο του 62 έως τον Δεκέμβρη του 75, όπου γίνεται φανερή η εποχικότητα που χαρακτηρίζει την συγκεκριμένη χρονοσειρά.

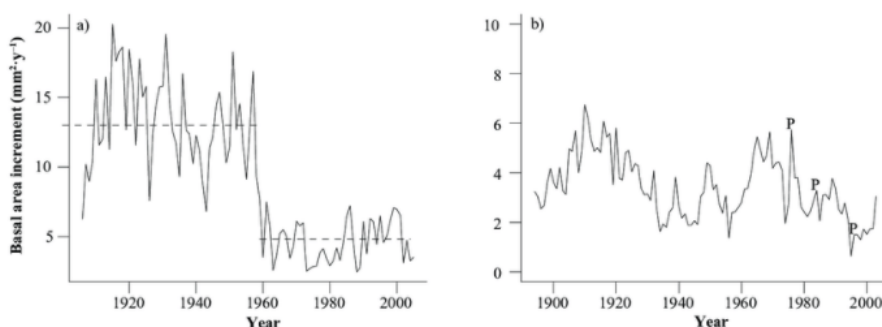
2.2.3 Ασυνέχειες

Οι ασυνέχειες, εμφανίζονται στην γραφική παράσταση μιας χρονοσειράς ως απότομες αλλαγές και είναι πιθανό να έχουν μόνιμο ή και παροδικό χαρακτήρα. Οι παροδικές α-



Σχήμα 2.3: Η μηνιαία παραγωγή γάλατος ανά αγελάδα [3]

συνέχειες, οι οποίες χαρακτηρίζονται και ως outliers, έχουν πάντα μικρή χρονική διάρκεια και θεωρείται ότι αποτελούν μια ασυνήθιστη παρατήρηση, η οποία ταυτίζεται με κάποιο εξαιρετικό γεγονός. Από την άλλη μεριά, οι ασυνέχειες με μόνιμο χαρακτήρα, γνωστές και ως (level-shifts), έχουν την μορφή μιας απότομης αλλαγής στο μέσο επίπεδο των τιμών της χρονοσειράς. Στο σχήμα 2.4, φαίνεται ένα παράδειγμα παροδικής ασυνέχειας (a) και ένα μόνιμης ασυνέχειας (b).



Σχήμα 2.4: Παραδείγματα χρονοσειρών με ασυνεχή συμπεριφορά [4]

Προκειμένου να κατανοήσουμε και να μελετήσουμε καλύτερα μια χρονοσειρά, χρειάζεται να εξάγουμε τα παραπάνω χαρακτηριστικά. Για να το πετύχουμε αυτό χρησιμοποιούμε μαθηματικές σχέσεις για την απομόνωσή τους. Θα μπορούσαμε να διατυπώσουμε μια χρονοσειρά ως μια συνάρτηση των βασικών αυτών χαρακτηριστικών της, η οποία στην γλώσσα των μαθηματικών θα έχει την εξής μορφή :

$$Y_t = f(S_t, T_t, C_t, R_t). \tag{2.1}$$

,όπου :

- Y_t = η παρατήρηση κατά το χρονικό βήμα t
 - S_t = η συνιστώσα της εποχικότητας κατά το χρονικό βήμα t
 - T_t = η συνιστώσα της τάσης κατά το χρονικό βήμα t
 - C_t = η συνιστώσα του κύκλου κατά το χρονικό βήμα t
 - R_t = η συνιστώσα της τυχαιότητας κατά το χρονικό βήμα t
- για $t = 0, 1, \dots$

Υπάρχουν αρκετές συναρτησιακές μορφές για την αντιστοίχιση των χαρακτηριστικών της χρονοσειράς σε μια χρονική περίοδο t . Μια από τις πιο διαδεδομένες μορφές είναι η προ-

σθετική με την οποία η συνάρτηση 2.1 να παίρνει την μορφή :

$$Y_t = S_t + T_t + C_t + R_t. \quad (2.2)$$

και η πολλαπλασιαστική :

$$Y_t = S_t * T_t * C_t * R_t. \quad (2.3)$$

2.3 Στατιστική ανάλυση χρονοσειρών

Η στατιστική ανάλυση των χρονοσειρών παρέχει μεθόδους με τις οποίες μπορούν να υπολογιστούν βασικά στατιστικά χαρακτηριστικά για μια χρονοσειρά Y με μέγεθος n παρατηρήσεων. Αυτό βοηθάει στην επιλογή σωστών και αποτελεσματικών μεθόδων πρόβλεψης για την δεδομένη χρονοσειρά. Τα κυριότερα στατιστικά χαρακτηριστικά μιας χρονοσειράς παρουσιάζονται παρακάτω.

2.3.1 Μέση τιμή (Average)

Ως μέση τιμή ορίζεται ο γραμμικός μέσος όρος των τιμών όλων των παρατηρήσεων της χρονοσειράς και καθορίζει το μέσο επίπεδο που κυμαίνονται οι πραγματικές τιμές της. Η μέση τιμή μπορεί να υπολογιστεί ως :

$$E = \frac{\sum_{i=1}^N Y_i}{N} \quad (2.4)$$

2.3.2 Τυπική απόκλιση (Standard Deviation)

Εμφανίζεται και ως διασπορά καθώς αποτελεί έναν δείκτη του βαθμού διασποράς των παρατηρήσεων γύρω από την μέση τιμή. Η χαμηλή τυπική απόκλιση σημαίνει ότι τα δεδομένα συγκεντρώνονται γύρω από το μέσο όρο και η υψηλή τυπική απόκλιση δείχνει ότι τα δεδομένα είναι πιο εκτεταμένα. Μαθηματικά υπολογίζεται από τον τύπο :

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (Y_i - E)^2}{N}} \quad (2.5)$$

2.3.3 Διακύμανση (Variance)

Διακύμανση (Variance) εκφράζεται ως το τετράγωνο της τυπικής απόκλισης και είναι ένας τρόπος για να δείξουμε πώς τα δεδομένα διασκορπίζονται ή διαδίδονται.

2.3.4 Συνδιακύμανση (Covariance)

Η συνδιακύμανση για δύο διακριτές τυχαίες μεταβλητές X και Y , προσφέρει ένα μέτρο συσχέτισης μεταξύ τους, με αποτέλεσμα να καθορίζει το αν οι μεταβλητές μεταβάλλονται ανάλογα (θετική συνδιακύμανση), αντιστρόφως ανάλογα (αρνητική συνδιακύμανση), ή αν οι

μεταβλητές είναι εντελώς ασυσχέτιστες μεταξύ τους (μηδενική συνδιακύμανση). Η συνδιακύμανση δείχνει επομένως την τάση στη γραμμική σχέση μεταξύ των μεταβλητών. Μπορεί να υπολογιστεί από τον τύπο :

$$\text{Cov}(X, Y) = \frac{\sum_{i=1}^N (X_i - E_x)(Y_i - E_Y)}{N} \quad (2.6)$$

2.3.5 Συντελεστής αυτοσυσχέτισης (Autocorrelation Coefficient)

Ο συντελεστής αυτοσυσχέτισης αποτελεί χαρακτηριστικό ένδειξης της συσχέτισης μεταξύ των παρατηρήσεων της ίδιας μεταβλητής με χρονική υστέρηση k περιόδων. Οι τιμές του συντελεστή αυτοσυσχέτισης κυμαίνονται στο διάστημα μεταξύ $[-1, 1]$. Όσο πιο κοντά στο 1 είναι η τιμή του συντελεστή σημαίνει ότι υπάρχει θετική συσχέτιση μεταξύ των παρατηρήσεων με χρονική υστέρηση k , ενώ όσο πιο κοντά στο -1 είναι σημαίνει ότι η συσχέτιση είναι αρνητική. Η τιμή του συντελεστή αυτοσυσχέτισης αποδεικνύεται εξαιρετικά χρήσιμη στον προσδιορισμό της εποχιακής συμπεριφοράς μιας σειράς παρατηρήσεων. Ο μαθηματικός τύπος υπολογισμού του συντελεστή αυτοσυσχέτισης είναι :

$$\rho_k = \frac{\sum_{i=1+k}^N (X_i - E_x)(Y_i - E_Y)}{\sum_{i=1}^N (Y_i - E_y)^2} \quad (2.7)$$

Κεφάλαιο 3

Νευρωνικά Δίκτυα

Στο κεφάλαιο αυτό αρχικά γίνεται μια εισαγωγή στα Αναδρομικά Νευρωνικά Δίκτυα καθώς και μια αναφορά στα κύρια προβλήματα που αυτά πραγματεύονται. Στην συνέχεια, γίνεται μια αναλυτική περιγραφή των αρχιτεκτονικών στις οποίες είναι βασισμένη η εργασία. Βασικές γνώσεις Νευρωνικών Δικτύων όπως γραμμική/softmax παλινδρόμηση, συνάρτηση απωλειών, αλγόριθμος gradient descent, αλγόριθμος back-propagation, Multi-layer Perceptron και συναρτήσεις ενεργοποίησης θεωρούνται γνωστές.

3.1 Αναδρομικά Νευρωνικά Δίκτυα

3.1.1 Εισαγωγή στα Αναδρομικά Νευρωνικά Δίκτυα

Τα Αναδρομικά Νευρωνικά Δίκτυα (Recurrent Neural Networks), αποτελούν την βάση της εργασίας, καθώς γύρω από αυτά πραγματοποιήθηκαν διάφορα πειράματα. Για να γίνει πιο κατανοητό το ζητούμενο της συγκεκριμένης εργασίας και ο τρόπος με τον οποίο συνδέεται με τις αντίστοιχες επιλογές αρχιτεκτονικών, είναι απαραίτητο να αναφερθούν οι βασικές κατηγορίες προβλημάτων που επιλύονται με Αναδρομικά Νευρωνικά Δίκτυα.

- Ταξινόμηση Ακολουθίας

Η ταξινόμηση ακολουθίας είναι ένα πρόβλημα μοντελοποίησης όπου δέχεται ως είσοδο κάποια ακολουθία δεδομένων και το έργο της είναι να προβλέψει μια κατηγορία για την ακολουθία αυτή.

Γνωστά πεδία: Ταξινόμηση βίντεο, Ταξινόμηση συναισθημάτων.

- Επισήμανση Ακολουθίας

Η επισήμανση ακολουθίας είναι ένα πρόβλημα μοντελοποίησης όπου δέχεται ως είσοδο κάποια ακολουθία δεδομένων και στόχος της είναι να προβλέψει μια κατηγορία για κάθε μία τιμή της ακολουθίας εισόδου.

Γνωστά πεδία: Επισήμανση κομματιών προφορικού/γραφικού λόγου.

- Δημιουργία Ακολουθίας

Η επισήμανση ακολουθίας είναι ένα πρόβλημα μοντελοποίησης όπου δέχεται ως είσοδο κάποια ακολουθία δεδομένων και στόχος της είναι να δημιουργήσει μια ακολουθία εξόδου.

Γνωστά πεδία: Μηχανική μετάφραση και μεταγλώττιση.

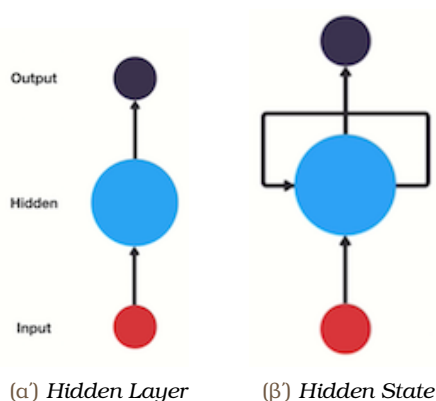
Το πρόβλημα που εξετάζεται στα πλαίσια της παρούσας εργασίας ανήκει στην πρώτη κατηγορία. Συγκεκριμένα, ο στόχος είναι η αναγνώριση της ανθρώπινης δραστηριότητας από ακολουθιακά δεδομένα, αποτελούμενα από μετρήσεις σε 128 διαδοχικές χρονικές στιγμές. Παρόμοιες εργασίες έχουν παραχθεί στο Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης του ΕΜΠ που αφορούν στην ανάλυση ανθρώπινης κίνησης [24], [25], [26], [27], [28].

Παρακάτω θα γίνει μία θεωρητική περιγραφή όλων των αρχιτεκτονικών που χρησιμοποιούνται για κάθε ένα από τα πειράματα.

3.1.2 Απλά Αναδρομικά Νευρωνικά Δίκτυα

Τα Απλά Αναδρομικά Νευρωνικά Δίκτυα (RNNs) είναι μία ειδική κατηγορία νευρωνικών δικτύων που διαχειρίζονται με μεγάλη αποτελεσματικότητα ακολουθιακά δεδομένα, καθώς διαθέτουν μία μεταβλητή κατάσταση που ονομάζεται Κρυφή Κατάσταση (Hidden State) και η οποία τους επιτρέπει να αποθηκεύουν πληροφορία από το παρελθόν (δηλαδή πληροφορίες από δεδομένα που έχουν συναντήσει νωρίτερα). Με αυτό τον τρόπο, μπορούν να εκμεταλλευτούν την πληροφορία του παρελθόντος και μαζί με τις πιο πρόσφατες εισόδους να καθορίσουν την επόμενη έξοδο. Λόγω της σειριακής επεξεργασίας δεδομένων θεωρούνται κατάλληλα για την επεξεργασία χρονοσειρών και άλλων ακολουθιακών δεδομένων [29]. Επιπλέον, έχει αποδειχθεί ότι αποδίδουν σε περιπτώσεις όπως η αναγνώριση γραφής ή η αναγνώριση ομιλίας [30].

Αξίζει να αναφερθεί η διαφορά μεταξύ των Hidden State και Hidden Layer που ήδη γνωρίζουμε από τα MLP. Τα δεύτερα είναι στρώσεις του δικτύου που δεν είναι ορατές από την διαδρομή της είσοδο προς την έξοδο, ενώ τα πρώτα είναι καταστάσεις που αποτελούν οι ίδιες τις εισόδους του δικτύου σε ό,τι κάνουμε σε ένα δεδομένο βήμα και μπορούν να υπολογιστούν μόνο κοιτάζοντας δεδομένα σε προηγούμενα βήματα, χρησιμοποιώντας πληροφορία από το παρελθόν. Παρακάτω φαίνεται σχηματικά η διαφορά τους.



Σχήμα 3.1: Διαφορά μεταξύ Hidden Layer και Hidden State [5]

Για να αναλύσουμε την αρχιτεκτονική των RNNs θα χρησιμοποιήσουμε ένα παράδειγμα γλωσσικού μοντέλου, προκειμένου να την κατανοήσουμε την σπουδαιότητά της. Για ένα τέτοιο μοντέλο αντιστοιχίζουμε τα κειμενικά δεδομένα σε tokens, όπου αυτά τα tokens μπο-

ρούν να θεωρηθούν ως μια ακολουθία διακριτών παρατηρήσεων, όπως λέξεις ή χαρακτήρες. Υποθέτουμε ότι τα tokens σε μια κειμενική ακολουθία μήκους T είναι x_1, x_2, \dots, x_T , στόχος του γλωσσικού μοντέλου είναι η εκτίμηση της από κοινού πιθανότητας της ακολουθίας, η οποία εφαρμόζοντας τους κανόνες της Θεωρίας Πιθανοτήτων υπολογίζεται από την 3.1.

$$P(x_1, x_2, \dots, x_T) = \prod_{t=1}^T P(x_t | x_1, \dots, x_{t-1}). \quad (3.1)$$

Ένα παράδειγμα υπολογισμού της πιθανότητας μιας κειμενικής ακολουθίας με τέσσερις λέξεις φαίνεται παρακάτω στην 3.2.

$$P(\text{deep, learning, is, fun}) = P(\text{deep})P(\text{learning}|\text{deep})P(\text{is}|\text{deep, learning})P(\text{fun}|\text{deep, learning, is}). \quad (3.2)$$

Προκειμένου να υπολογίσουμε το γλωσσικό μοντέλο, πρέπει να υπολογίσουμε την πιθανότητα των λέξεων και την υπό συνθήκη πιθανότητα μιας λέξης δεδομένης των προηγούμενων λέξεων. Αυτές οι πιθανότητες είναι ουσιαστικά οι παράμετροι του γλωσσικού μοντέλου.

Εάν εφαρμόσουμε τα μοντέλα Markov στο γλωσσικό μοντέλο οδηγούμαστε σε έναν αριθμό προσεγγίσεων του μοντέλου. Σημειώνεται ότι μεγαλύτερος βαθμός n στα μοντέλα Markov αντιστοιχεί σε μεγαλύτερες εξαρτήσεις 3.3, 3.4.

$$P(x_1, x_2, x_3, x_4) = P(x_1)P(x_2|x_1)P(x_3|x_2)P(x_4|x_3). \quad (3.3)$$

$$P(x_1, x_2, x_3, x_4) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)P(x_4|x_2, x_3). \quad (3.4)$$

Ωστόσο, με την αύξηση του βαθμού n , έχουμε εκθετική αύξηση των παραμέτρων του μοντέλου. Ως εκ τούτου, θα ήταν προτιμότερο να χρησιμοποιήσουμε ένα ελαφρώς λανθάνον μοντέλο που θα κάνει την εξής προσέγγιση:

$$P(x_t | x_{t-1}, \dots, x_1) \approx P(x_t | h_{t-1}). \quad (3.5)$$

Η h_{t-1} θεωρείται κρυμμένη μεταβλητή και έχει αποθηκευμένη πληροφορία από τα $t - 1$ προηγούμενα βήματα. Αυτή η παράμετρος θα ταυτιστεί με την Κρυφή Κατάσταση που αναφέρθηκε στην αρχή αυτής της ενότητας και για κάθε βήμα t θα υπολογίζεται με βάση την παρούσα είσοδο x_t και την προηγούμενη Κρυφή Κατάσταση h_{t-1} (3.6). Η Κρυφή Κατάσταση h_t είναι αυτή που εμπεριέχει όλη την πληροφορία που έχουμε δει έως τη χρονική στιγμή t .

$$h_t = f(x_t, h_{t-1}). \quad (3.6)$$

Παρατηρούμε ότι για κατάλληλα ορισμένη συνάρτηση f , στο γλωσσικό μοντέλο αναιρείται η προσέγγιση 3.5.

Ας υποθέσουμε μια δέσμη της εισόδου, $X_t \in \mathbb{R}^{n \times d}$ (batch-size) για την χρονική στιγμή t . Πιο αναλυτικά, για μια δέσμη της εισόδου με n δείγματα ακολουθίας, κάθε σειρά του πίνακα X_t αντιστοιχεί σε ένα δείγμα για τη χρονική στιγμή t της ακολουθίας. Κατόπιν, συμβολίζουμε ως $H_t \in \mathbb{R}^{n \times h}$ την κρυφή μεταβλητή για το χρονικό βήμα t .

Σε αντίθεση με τα MLPs στα RNNs η προηγούμενη κρυφή κατάσταση ή η κρυφή κα-

τάσταση του προηγούμενου χρονικού βήματος, H_{t-1} παραμένει αποθηκευμένη και επιπλέον εισάγεται ένας πίνακας βαρών $W_{hh} \in \mathbb{R}^{h \times h}$, ο οποίος περιγράφει πως οι κρυφές μεταβλητές από τα προηγούμενα χρονικά βήματα θα χρησιμοποιηθούν στο παρόν χρονικό βήμα. Πιο συγκεκριμένα, ο υπολογισμός της κρυφής κατάστασης του τρέχοντος χρονικού βήματος καθορίζεται τόσο από την είσοδο του τρέχοντος χρονικού βήματος όσο και από την κρυφή κατάσταση του προηγούμενου χρονικού βήματος ως εξής:

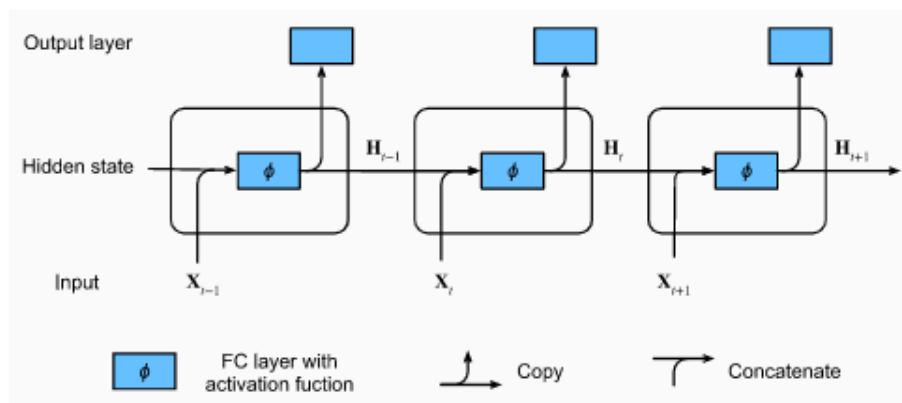
$$H_t = \phi(X_t W_{xh} + H_{t-1} W_{hh} + b_h). \quad (3.7)$$

Συγκριτικά με την εξίσωση ενός Hidden Layer παρατηρούμε ότι υπάρχει ένας πρόσθετος όρος, ο $H_{t-1} W_{hh}$. Ο όρος αυτός συνδέει τις κρυφές καταστάσεις H_t και H_{t-1} μεταξύ γειτονικών χρονικών βημάτων αναδρομικά. Συνεπώς η ιστορικότητα της πληροφορίας των ακολουθιών εισόδου διατηρείται έως το τρέχον βήμα σε ικανοποιητικό βαθμό. Αυτός είναι και ο λόγος που η μεταβλητή H ονομάζεται κρυφή κατάσταση και τα νευρωνικά δίκτυα που με τέτοιες κρυφές καταστάσεις Αναδρομικά Νευρωνικά Δίκτυα.

Γενικά υπάρχουν αρκετοί τρόποι για να κατασκευαστούν Αναδρομικά Νευρωνικά Δίκτυα, κάποιος θα αναλυθούν και στο πλαίσιο της τρέχουσας διπλωματικής εργασίας. Τα δίκτυα όμως που η κρυφή τους κατάσταση ορίζεται από την σχέση 3.7 είναι τα Απλά Νευρωνικά Δίκτυα. Για κάθε χρονικό βήμα t , η έξοδος του RNN υπολογίζεται με παρόμοιο τρόπο με του MLP από την σχέση 3.8.

$$O_t = H_t W_{hq} + b_q. \quad (3.8)$$

Οι παράμετροι ενός Απλού RNN συμπεριλαμβάνουν τα βάρη $W_{xh} \in \mathbb{R}^{d \times h}$, $W_{hh} \in \mathbb{R}^{h \times h}$, και τον όρο bias $b_h \in \mathbb{R}^{1 \times h}$ της κρυφής κατάστασης, καθώς και τα βάρη $W_{hq} \in \mathbb{R}^{h \times q}$, και τον όρο bias $b_q \in \mathbb{R}^{1 \times q}$ του εξωτερικού στρώματος. Αξίζει να σημειωθεί ότι ακόμη και σε διαφορετικά χρονικά βήματα, τα RNNs χρησιμοποιούν πάντα αυτές τις παραμέτρους μοντέλου. Επομένως, το κόστος παραμετροποίησης ενός RNN δεν αυξάνεται καθώς αυξάνεται ο αριθμός των βημάτων χρόνου.



Σχήμα 3.2: Απλό Αναδρομικό Νευρωνικό Δίκτυο με κρυφή κατάσταση [6].

Στο σχήμα 3.2 απεικονίζεται η υπολογιστική λογική ενός Απλού RNN σε τρία διαδοχικά χρονικά βήματα. Σε κάθε χρονικό βήμα, η έξοδος του Hidden State μπορεί να υπολογιστεί ως εξής: 1) συνενώνοντας (concatenation) την είσοδο X_t της τρέχουσας χρονικής στιγμής t

και την κρυφή κατάσταση στο προηγούμενο βήμα H_{t-1} και ι) τροφοδοτώντας το αποτέλεσμα της συνένωσης σε ένα πλήρως συνδεδεμένο επίπεδο (fully-connected layer) με συνάρτηση ενεργοποίησης ϕ . Η έξοδος ενός τέτοιου πλήρους συνδεδεμένου επιπέδου είναι η κρυφή κατάσταση H_t του τρέχοντος χρονικού βήματος t .

Στη συνένωση των X_t και H_{t-1} συμμετείχαν οι μήτρες βαρών W_{xh} και W_{hh} καθώς και ο όρος bias b_h , όπως ορίζει η εξίσωση 3.7. Επιπλέον, η κρυφή κατάσταση του τρέχοντος βήματος t , H_t , θα συμμετάσχει στον υπολογισμό της κρυφής κατάστασης H_{t+1} του επόμενου βήματος $t+1$. Τέλος, θα τροφοδοτηθεί στο πλήρως συνδεδεμένο στρώμα εξόδου για να υπολογιστεί η έξοδος O_t για αυτήν τη χρονική στιγμή t .

3.1.3 Αλγόριθμος Back Propagation Διαμέσω χρόνου (BPTT)

Η εκπαίδευση των Αναδρομικών Νευρωνικών Δικτύων RNNs γίνεται μέσω του αλγόριθμου Back Propagation διαμέσω χρόνου (Back Propagation through time- BPTT [31]), ο οποίος είναι μια παραλλαγή του αλγορίθμου Back Propagation [32] που αποτελεί τον αλγόριθμο εκπαίδευσης για τα MLP δίκτυα.

Για την παρουσίαση του αλγορίθμου εκπαίδευσης των Αναδρομικών Νευρωνικών Δικτύων, για λόγους απλοποίησης θα παραλείψουμε τους όρους bias. Για κάθε χρονικό βήμα t , θεωρούμε ένα διάνυσμα εισόδου και την αντίστοιχη ετικέτα να είναι $x_t \in \mathbb{R}^d$ καθώς και y_t . Επομένως, η κρυφή κατάσταση $h_t \in \mathbb{R}^h$ και η έξοδος $o_t \in \mathbb{R}^q$ υπολογίζονται ως εξής:

$$h_t = W_{hx}x_t + W_{hh}h_{t-1}, \quad (3.9)$$

$$o_t = W_{qh}h_t, \quad (3.10)$$

όπου $W_{hx} \in \mathbb{R}^{h \times d}$, $W_{hh} \in \mathbb{R}^{h \times h}$ και $W_{qh} \in \mathbb{R}^{q \times h}$ είναι οι παράμετροι των βαρών.

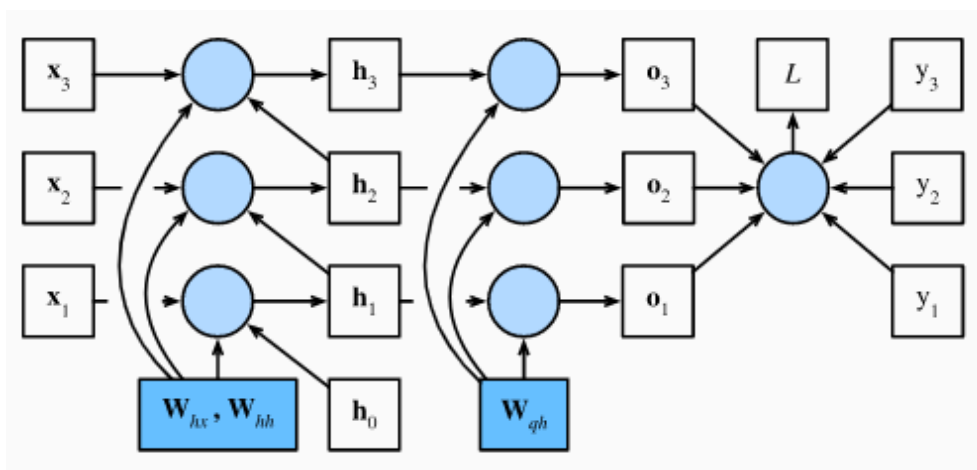
Συμβολίζουμε με $l(o_t, y_t)$ την απώλεια (loss) για ένα συγκεκριμένο χρονικό βήμα t , δηλαδή την διαφορά της εξόδου o_t και της ετικέτας y_t . Επομένως, η συνολική συνάρτηση απωλειών (loss function) για T χρονικά βήματα από την αρχή της ακολουθίας εισόδου είναι:

$$L = \frac{1}{T} \sum_{t=1}^T l(o_t, y_t). \quad (3.11)$$

Προκειμένου να απεικονιστεί η εξάρτηση μεταξύ των μεταβλητών μοντέλου και των παραμέτρων κατά τον υπολογισμό του RNN, μπορούμε να σχεδιάσουμε ένα υπολογιστικό γράφημα για το μοντέλο, όπως φαίνεται στο σχήμα 3.3. Φαίνεται ξεκάθαρα ότι ο υπολογισμός της κρυφής κατάστασης στο τρίτο χρονικό βήμα, h_3 , εξαρτάται από τις παραμέτρους των βαρών W_{hx} , W_{hh} , την κρυφή κατάσταση του προηγούμενου χρονικού βήματος h_2 και την τρέχουσα είσοδο x_3 .

Σημειώνουμε ότι οι παράμετροι του μοντέλου είναι W_{hx} , W_{hh} και W_{qh} . Γενικά, η εκπαίδευση αυτού του μοντέλου απαιτεί τον υπολογισμό της κλίσης της συνάρτησης απωλειών ως προς αυτές τις παραμέτρους, δηλαδή $\partial L / \partial W_{hx}$, $\partial L / \partial W_{hh}$ και $\partial L / \partial W_{qh}$. Οι εξαρτήσεις μεταξύ των μεγεθών μπορούν να βρεθούν διασχίζοντας το γράφημα του σχήματος 3.3 προς την αντίθετη κατεύθυνση των βελών, αποθηκεύοντας τις εκάστοτε παραγώγους.

Λόγω τως διαφορετικών διαστάσεων των πινάκων και των διανυσμάτων, θα χρησιμοποιηθεί



Σχήμα 3.3: Υπολογιστικό γράφημα που αναδεικνύει τις εξαρτήσεις για ένα μοντέλο RNN σε τρία χρονικά βήματα. Τα πλαίσια αντιπροσωπεύουν μεταβλητές (όχι σκιασμένα) ή παραμέτρους (σκιασμένα) και οι κύκλοι αντιπροσωπεύουν τελεστές πράξεων [7].

ο τελεστής $prod$ για τον πολλαπλασιασμό τους, όπου αυτό απαιτείται με βάση τον κανόνα αλυσίδας. Υποθέτουμε ότι έχουμε τις συναρτήσεις $Y = f(X)$ και $Z = g(Y)$, όπου οι εισοδοι και οι εξοδοι X, Y, Z είναι τανυστές αυθαίρετων διαστάσεων. Η χρήση του τελεστή $prod$ εμπερικλείει όλες τις απαραίτητες ενέργειες που θα πρέπει να γίνουν στους τανυστές-ορίσματα X, Y, Z , όπως η αντιστροφή και η εναλλαγή θέσης των εισόδων. Παρακάτω φαίνεται ο κανόνας της αλυσίδας για τις συναρτήσεις που υποθέσαμε.

$$\frac{\partial Z}{\partial X} = prod\left(\frac{\partial Z}{\partial Y}, \frac{\partial Y}{\partial X}\right). \quad (3.12)$$

Επιστρέφοντας στην ανάλυση μας, η διαφορίση της συνάρτησης απωλειών ως προς την έξοδο o_t του μοντέλου για κάποιο δεδομένη χρονική στιγμή t δίνεται από την σχέση 3.13

$$\frac{\partial L}{\partial o_t} = \frac{\partial l(o_t, y_t)}{\partial Y} \in \mathbb{R}^q. \quad (3.13)$$

Βασιζόμενοι στην αρχιτεκτονική του δικτύου 3.3 ξέρουμε ότι η συνάρτηση L εξαρτάται από το W_{gh} μέσω των εξόδων o_1, \dots, o_T . Επομένως, υπολογίσουμε την κλίση της συνάρτησης απωλειών ως προς τα βάρη W_{gh} στο επίπεδο εξόδου, $\partial L / \partial W_{gh} \in \mathbb{R}^{q \times h}$, χρησιμοποιώντας τον κανόνα της αλυσίδας.

$$\frac{\partial L}{\partial W_{gh}} = \sum_{t=1}^T prod\left(\frac{\partial L}{\partial o_t}, \frac{\partial o_t}{\partial W_{gh}}\right) = \sum_{t=1}^T \frac{\partial L}{\partial o_t} h_t^T. \quad (3.14)$$

Στην συνέχεια, από την εικόνα 3.3 φαίνεται πως στο τελευταίο χρονικό βήμα T η συνάρτηση απωλειών εξαρτάται από την κρυφή κατάσταση h_T μόνο μέσω της o_T . Επομένως η κλίση $\partial L / \partial h_T \in \mathbb{R}^h$ μπορεί να υπολογιστεί εύκολα εφαρμόζοντας τον κανόνα της αλυσίδας.

$$\frac{\partial L}{\partial h_T} = prod\left(\frac{\partial L}{\partial o_T}, \frac{\partial o_T}{\partial h_T}\right) = W_{gh}^T \frac{\partial L}{\partial o_T} h_T^T. \quad (3.15)$$

Για κάθε άλλο χρονικό βήμα $t < T$, η συνολική συνάρτηση L εξαρτάται από την κρυφή

κατάσταση h_t μέσω και της επόμενης κρυφής κατάστασης h_{t+1} και της o_t . Εφαρμόζοντας ξανά τον κανόνα της αλυσίδας για τον υπολογισμό της κλίσης $\partial L/\partial h_t \in \mathbb{R}^h$ σε κάθε χρονικό βήμα $t < T$, μπορεί αναδρομικά να υπολογιστεί από τον την σχέση 3.16.

$$\frac{\partial L}{\partial h_t} = \text{prod}\left(\frac{\partial L}{\partial h_{t+1}}, \frac{\partial h_{t+1}}{\partial h_t}\right) + \text{prod}\left(\frac{\partial L}{\partial o_t}, \frac{\partial o_t}{\partial h_t}\right) = W_{hh}^T \frac{\partial L}{\partial h_{t+1}} + W_{qh}^T \frac{\partial L}{\partial o_t}. \quad (3.16)$$

Ξεδιπλώνοντας την προηγούμενη αναδρομική σχέση 3.16 για οποιοδήποτε χρονικό βήμα $1 \leq t \leq T$ μπορούμε να πάρουμε:

$$\frac{\partial L}{\partial h_t} = \sum_{i=t}^T (W_{hh}^T)^{T-i} W_{qh}^T \frac{\partial L}{\partial o_{T+t-i}}. \quad (3.17)$$

Τέλος, στο σχήμα 3.3 φαίνονται η εξάρτηση της συνάρτησης απωλειών από τις παραμέτρους του μοντέλου W_{hx} και W_{hh} στο κρυφό επίπεδο, μέσω των κρυφών καταστάσεων h_1, \dots, h_T . Για τον υπολογισμό της κλίσης της L ως προς αυτές τις παραμέτρους, $\partial L/\partial W_{hx} \in \mathbb{R}^{h \times d}$ και $\partial L/\partial W_{hh} \in \mathbb{R}^{h \times h}$ εφαρμόζουμε για μια ακόμη φορά τον κανόνα της αλυσίδας και προκύπτει:

$$\frac{\partial L}{\partial W_{hx}} = \sum_{t=1}^T \text{prod}\left(\frac{\partial L}{\partial h_t}, \frac{\partial L}{\partial W_{hx}}\right) = \sum_{t=1}^T \frac{\partial L}{\partial h_t} x_t^T. \quad (3.18)$$

$$\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^T \text{prod}\left(\frac{\partial L}{\partial h_t}, \frac{\partial L}{\partial W_{hh}}\right) = \sum_{t=1}^T \frac{\partial L}{\partial h_t} h_{t-1}^T. \quad (3.19)$$

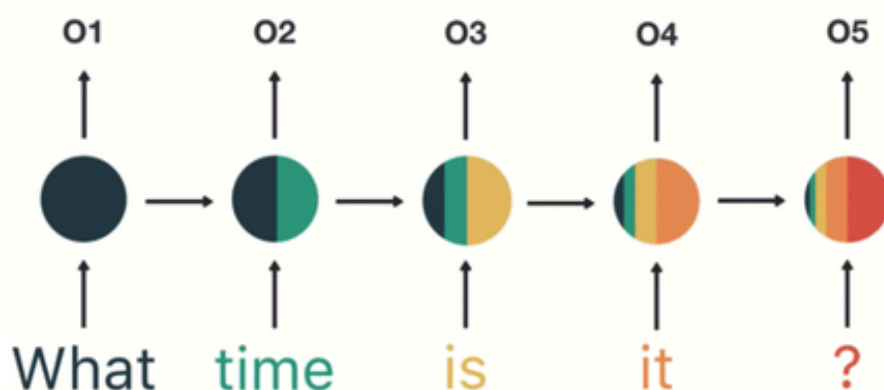
όπου η ποσότητα $\partial L/\partial h_t$ υπολογίζεται επαναληπτικά από τις σχέσεις 3.16 και 3.17. Αυτή η ποσότητα είναι και ο βασικός παράγοντας που επηρεάζει την αριθμητική σταθερότητα.

Παρόλο που ο αλγόριθμος Back Propagation διαμέσω χρόνου είναι αρκετά χρονοβόρος και περίπλοκος έχει ένα θετικό σημείο. Αποθηκεύει τις ενδιάμεσες μεταβλητές του αλγορίθμου με τη σειρά υπολογισμού τους για να αποφεύγονται οι αχρείαστοι επιπρόσθετοι υπολογισμοί, όπως για παράδειγμα αποθηκεύει την $\partial L/\partial h_t$ που χρησιμοποιείται στους υπολογισμούς των 3.18, 3.19, επιταχύνοντας σημαντικά τον χρόνο εκτέλεσης του αλγορίθμου.

3.1.4 Αδυναμίες Αναδρομικών Νευρωνικών Δικτύων

Ο αλγόριθμο εκπαίδευσης των Αναδρομικών Νευρωνικών Δικτύων που μελετήσαμε στην προηγούμενη ενότητα έχει κάποια τρωτά σημεία [33]. Από την σχέση 3.18 είναι φανερά κάποια προβλήματα κατά την εκπαίδευση του δικτύου, καθώς όταν οι ακολουθίες εισόδου του μοντέλου έχουν μεγάλο μήκος, τότε οδηγούμαστε δυνητικά σε πολύ μεγάλες δυνάμεις του W_{hh}^T . Οι ιδιότητες αυτής της μήτρας που είναι μικρότερες από 1 μετά από αλληπάλληλες αναδρομές εξαφανίζονται, ενώ οι ιδιότητες μεγαλύτερες του 1 αποκλίνουν. Έτσι, οδηγούμαστε σε αριθμητική αστάθεια, το οποίο εκδηλώνεται είτε υπό τη μορφή εξαφανιζόμενων είτε υπό τη μορφή εκρηγνυόμενων κλίσεων (παραγώγων) καθώς αυξάνονται οι επαναλήψεις. Αυτό σημαίνει, ότι οι κλίσεις (παραγώγοι) κατά την εκτέλεση του αλγορίθμου εκρηγνυονται σε πολύ μεγάλες τιμές ή εξαφανίζονται σε πολύ μικρές.

Αποτέλεσμα αυτού, είναι ότι τα πρώτα στάδια ενός δικτύου αποτελούμενο από n στρώσεις θα έχουν είτε πολύ μικρές ανανεώσεις στα βάρη τους και άρα δεν θα εκπαιδεύονται ικανο-



Σχήμα 3.4: Εικονική αναπαράσταση της βραχυπρόθεσμης μνήμης των RNNs [5]

ποιητικά είτε πολύ μεγάλες, με συνέπεια το δίκτυο να φτάσει σε σημείο κορεσμού και να σταματήσει την εκπαίδευσή του. Το πρόβλημα αυτό είναι γνωστό στο πεδίο των Νευρωνικών Δικτύων με το όνομα η Εξαφάνιση ή Έκρηξη της παραγωγού (Vanishing/Exploding gradient Problem) [34]. Η ύπαρξη αυτού του προβλήματος οφείλεται στην φύση των αλγορίθμων εκπαίδευσης της Κατάβασης Πλαγιάς (Gradient Descent) και του Back Propagation.

Τα αναδρομικά δίκτυα λόγω του Vanishing Gradient Problem χαρακτηρίζονται και ως δίκτυα βραχυπρόθεσμης μνήμης (short-term memory). Θα προσπαθήσουμε να δώσουμε μια σχηματική ερμηνεία του όρου αυτού. Υποθέστε ότι θέλουμε να εκπαιδεύσουμε ένα απλό αναδρομικό νευρωνικό δίκτυο (RNN), προκειμένου να μπορεί να απαντήσει στην ερώτηση "Τι ώρα είναι;" ("What time is it?"), σχήμα 3.4.

Από το σχήμα φαίνεται πως η πληροφορία από προηγούμενες λέξεις/ χρονικές στιγμές τροφοδοτείται κωδικοποιημένη μέσω των κρυφών καταστάσεων στις επόμενες χρονικές στιγμές. Η πληροφορία αυτή απεικονίζεται με διαφορετικό χρώμα για κάθε λέξη. Συνεπώς, φαίνεται στο τελικό βήμα ότι το RNN έχει κωδικοποιήσει πληροφορίες από όλες τις λέξεις των προηγούμενων βημάτων. Παρόλα αυτά, στο παράδειγμα μας 3.4 στο τελικό βήμα, πληροφορία από τις λέξεις "What" και "time" σχεδόν δεν υπάρχει/ έχει ξεχαστεί. Αυτό γίνεται ακόμα πιο έντονο όσο ο αριθμός των βημάτων αυξάνεται καθώς όλο και περισσότερες πληροφορίες "ξεχνιούνται". Για αυτό το λόγο, τα RNNs χαρακτηρίζονται και ως short-term memory δίκτυα.

3.2 Επεκτάσεις των Αναδρομικών Νευρωνικών Δικτύων

Στην προηγούμενη ενότητα εισαγάγαμε τα βασικά των Απλών Αναδρομικών Νευρωνικών Δικτύων τα οποία μπορούν να διαχειριστούν ακολουθιακά δεδομένα. Επιπροσθέτως, έγινε αναφορά και στα τρωτά τους σημεία όπου και φανερώθηκε η ανάγκη επέκτασή τους. Λύση σε αυτό το πρόβλημα προτάθηκε πρώτη φορά από τους Sepp Hochreiter και Jürgen Schmidhuber με την ανάπτυξη ενός επεκταμένου RNN που ονομάστηκε δίκτυο Μακράς Βραχυπρόθεσμης Μνήμης, (Long Short-term Memory-LSTM Neural Network).

Παρακάτω σε αυτή την ενότητα θα εισαγάγουμε δύο από τα πιο ευρέως χρησιμοποιούμενα δίκτυα και επεκτάσεις των απλών RNNs, συγκεκριμένα τις Φραγμένες Αναδρομικές Μο-

νάδες (Gated Recurrent Unit-GRU) και τα Δίκτυα Μακροχρόνιας Βραχυπρόθεσμης Μνήμης (LSTM). Οι συγκεκριμένες επεκτάσεις υιοθετούνται συχνά σε σύγχρονα επαναλαμβανόμενα δίκτυα.

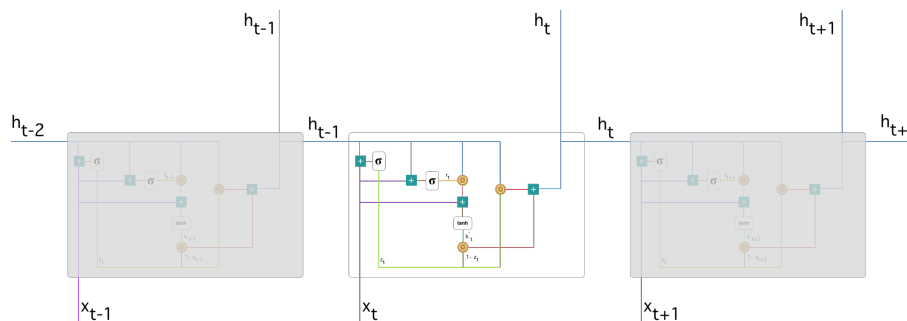
3.2.1 Φραγμένα Αναδρομικά Δίκτυα - GRUs

Η βασική διάκριση μεταξύ των GRUs και των RNNs είναι ότι τα πρώτα υποστηρίζουν έναν πολυπλοκότερο μηχανισμό πύλης στο δικό τους Hidden State. Ένας τέτοιος ειδικός μηχανισμός δίνει στα δίκτυα την δυνατότητα να μαθαίνουν πότε πρέπει να ενημερώνεται (update) μια κρυφή κατάσταση και επίσης πότε πρέπει να γίνεται επαναφορά (reset). Αυτός ο μηχανισμός διευθετεί τις αδυναμίες των απλών RNNs καθώς το δίκτυο θα μπορεί να μάθει εάν το πρώτο στοιχείο της ακολουθίας εισόδου έχει μεγάλη σημασία να μην ενημερώνει την κρυφή του κατάσταση μετά την πρώτη παρατήρηση. Ομοίως, θα μάθει να παραλείπει άσχετες προσωρινές παρατηρήσεις [35].

Στην πράξη, τα GRUs προτάθηκαν πρώτη φορά το 2014 από τους Cho, et al [36]. Παρακάτω θα εξηγηθεί με λεπτομέρεια η χρήση και η συμβολή των δύο πυλών του δικτύου (update και reset) ώστε να γίνει κατανοητή η αποτελεσματικότητα και η υπεροχή των GRUs έναντι των απλών RNNs.

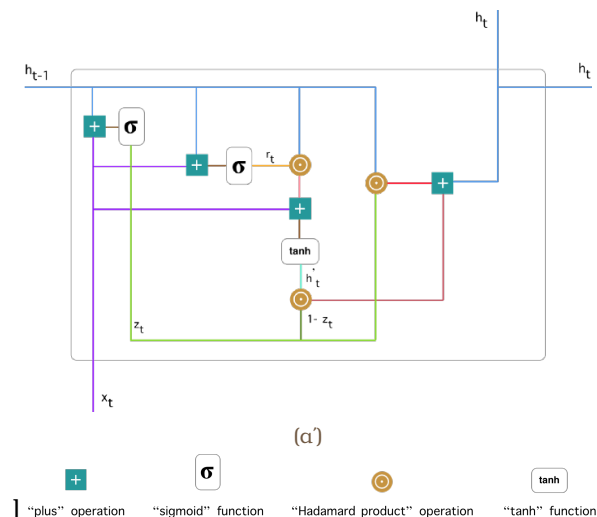
Πρακτικά αυτές οι δύο πύλες αποτελούν δύο διανύσματα τα οποία αποφασίζουν για την πληροφορία που χρειάζεται να περάσει στην έξοδο. Η ιδιότητα αυτών των διανυσμάτων να μπορούν να συμπεριληφθούν στην φάση της εκπαίδευσης κάνει τα GRUs αποτελεσματικά καθώς κρατούν πληροφορίες για αρκετό χρόνο, ενώ ταυτόχρονα αφαιρούν πληροφορίες που θεωρούνται άσχετες με την πρόβλεψη.

Για να εξηγήσουμε τα μαθηματικά πίσω από αυτή την διαδικασία, θα εξετάσουμε μια απλή μονάδα (single unit) του συγκεκριμένου αναδρομικού δικτύου.



Σχήμα 3.5: Αναδρομικό Νευρωνικό Δίκτυο με GRUs μονάδες [8]

Παράκατω απεικονίζεται με πιο πολύ λεπτομέρεια μια GRU μονάδα, πάνω στην οποία θα στηριχτεί η ανάλυσή μας 3.6, 3.5.



Σχήμα 3.6: Αναλυτική σχηματική απεικόνιση μιας GRU μονάδας [8]

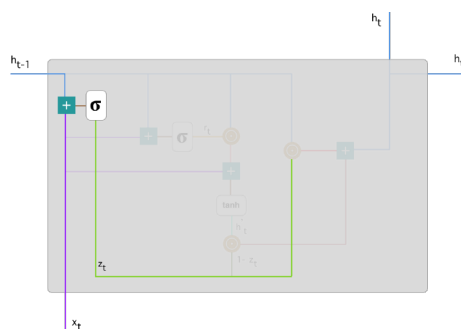
Πύλη Ενημέρωσης (update gate)

Για αρχή θα υπολογίσουμε την έξοδο της πύλης ενημέρωσης z_t για ένα συγκεκριμένο χρονικό βήμα t χρησιμοποιώντας τον ακόλουθο τύπο.

$$z_t = \sigma(W^z X_t + U^z h_{t-1}) \quad (3.20)$$

Από την σχέση 3.20 παρατηρούμε ότι η είσοδος την χρονική στιγμή t με την είσοδο της στην GRU μονάδα πολλαπλασιάζεται με τα δικά της βάρη W^z . Επιπλέον και η κρυφή κατάσταση h_{t-1} που φέρει πληροφορία από τις προηγούμενες $t - 1$ μονάδες GRU πολλαπλασιάζεται με τα δικά της βάρη εκπαίδευσης U^z . Στην συνέχεια, τα δύο αυτά γινόμενα αθροίζονται και στο άθροισμα τους εφαρμόζεται μια sigmoid συνάρτηση ενεργοποίησης (σχήμα 3.11) και το αποτέλεσμα πλέον κινείται στο εύρος από 0 μέχρι 1.

Η πύλη ενημέρωσης update gate βοηθάει το μοντέλο να καθορίσει πόση από την προηγούμενη πληροφορία (από τα προηγούμενα χρονικά βήματα) χρειάζεται να περαστεί για τα μελλοντικά. Αυτό είναι πολύ σημαντικό καθώς το μοντέλο μπορεί να αποφασίσει να κρατήσει όλη την προηγούμενη πληροφορία και να εξαλείψει τον κίνδυνο για vanishing gradient.



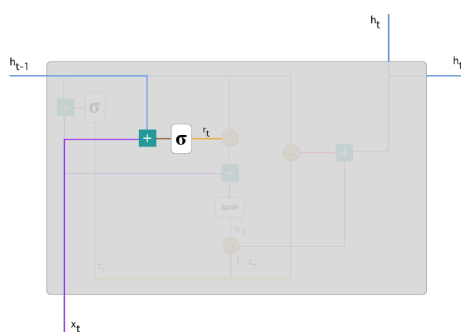
Σχήμα 3.7: Σχηματική απεικόνιση της πύλης ενημέρωσης [8]

Πύλη επαναφοράς (reset gate)

Σε αντίθεση με την πύλη ενημέρωσης η πύλη επαναφοράς καθορίζει πόση από την πληροφορία του παρελθόντος μπορεί να ξεχαστεί από τα μοντέλο.

Ο τύπος υπολογισμού είναι ο ίδιος με την update gate, το μόνο που αλλάζει είναι τα βάρη και η χρησιμότητα της πύλης. Στο σχήμα παρακάτω φαίνεται που βρίσκεται η πύλη επαναφοράς.

$$r_t = \sigma(W^r X_t + U^r h_{t-1}) \quad (3.21)$$



Σχήμα 3.8: Σχηματική απεικόνιση της πύλης επαναφοράς [8]

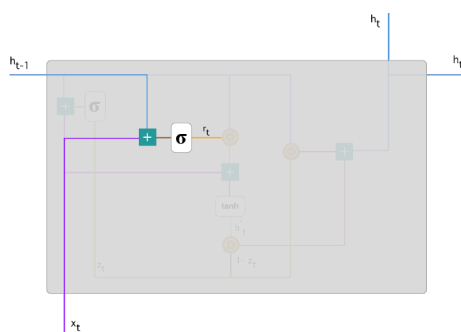
Τρέχον περιεχόμενο μνήμης (Current Memory Content)

Για να ολοκληρωθεί η παρουσίαση των GRUs χρειάζεται να εισαγάγουμε ένα νέο περιεχόμενο μνήμης, το οποίο θα εκμεταλλευτεί το αποτέλεσμα της πύλης επαναφοράς έτσι ώστε να αποθηκεύσει σχετική πληροφορία από το παρελθόν.

Ο τύπος υπολογισμού του τρέχοντος περιεχόμενου μνήμης είναι ο ακόλουθος.

$$h'_t = \tanh(WX_t + r_t \odot Uh_{t-1}) \quad (3.22)$$

Από τον τύπο 3.22 βλέπουμε ότι ο όρος που καθορίζει ποια πληροφορία από το παρελθόν δεν χρειάζεται να διατηρηθεί είναι ο $r_t \odot Uh_{t-1}$. Έτσι γίνεται φανερό η συμβολή της πύλης επαναφοράς στο τελικό αποτέλεσμα.



Σχήμα 3.9: Σχηματική απεικόνιση για Current Memory Content [8]

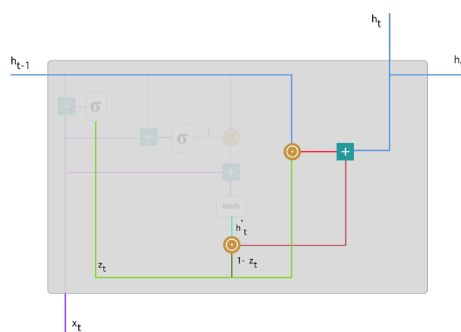
Κρυφή Κατάσταση

Για την ολοκλήρωση της παρουσίασης των GRUs ορίζουμε τον τρόπο με τον οποίο υπολογίζεται η κρυφή κατάσταση για το τρέχον χρονικό βήμα.

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h'_t \quad (3.23)$$

Στο σημείο αυτό η πύλη ενημέρωσης θα καθορίσει τι θα κρατήσει από το current memory context $h'(t)$ και τα προηγούμενα βήματα ($h(t-1)$).

Όταν η πύλη ενημέρωσης z_t είναι κοντά στο 1, διατηρούμε απλώς την παλιά κατάσταση. Σε αυτήν την περίπτωση, οι πληροφορίες από το x_t ουσιαστικά αγνοούνται, παρακάμπτοντας έτσι το βήμα t στην αλυσίδα εξάρτησης της πληροφορίας. Αντιθέτως, όποτε η z_t είναι κοντά στο 0, η νέα κρυφή κατάσταση H_t πλησιάζει πιο πολύ στην h'_t . Κατά αυτόν τον τρόπο, η αρχιτεκτονική της μονάδας GRU μας βοηθά να αντιμετωπίσουμε το πρόβλημα της εξαφάνισης παραγώγου που συναντάται στα απλά RNN, αλλά και να συγκρατήσουμε καλύτερα τις αλληλοεξαρτήσεις ακολουθιών με μεγάλες αποστάσεις χρονικών βημάτων.



Σχήμα 3.10: Σχηματική απεικόνιση για Hidden State [8]

Με προσεκτική εκπαίδευση τα GRUs μπορούν να λύσουν αρκετά πολύπλοκα προβλήματα. Μερικά παραδείγματα είναι της πρόγνωσης πρόβλεψης της εξέλιξης των ασθενειών [37] και των προβλέψεων χρηματιστηρίου [38].

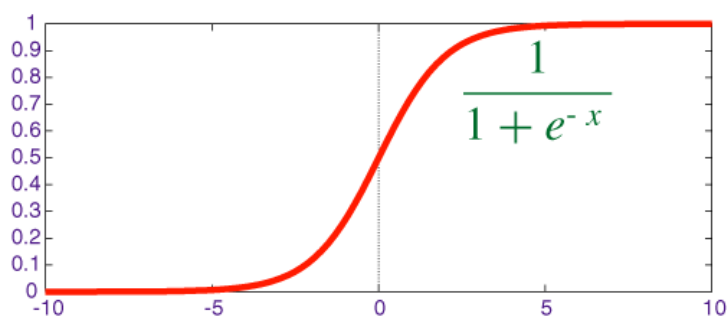
3.2.2 Δίκτυα Μακράς Βραχυπρόθεσμης Μνήμης - LSTM

Η πρόκληση για την αντιμετώπιση της μακροπρόθεσμης διατήρησης πληροφοριών και της βραχυπρόθεσμης παράκαμψης εισόδων υπήρχε για μεγάλο χρονικό διάστημα. Μία από τις πρώτες προσεγγίσεις για να αντιμετωπιστεί αυτό ήταν η μακροπρόθεσμη μνήμη (LSTM), η οποία προτάθηκε από τους Hochreiter και Schmidhuber [39]. Μοιράζεται πολλές από τις ιδιότητες των GRUs. Είναι ενδιαφέρον ότι τα LSTM έχουν ελαφρώς πιο πολύπλοκο σχεδιασμό από τα GRU, αλλά προηγούνται των GRUs σχεδόν δύο δεκαετίες [40].

Η κύρια ιδέα των LSTM είναι το κελί μνήμης (cell state) και οι διάφορες πύλες του. Το πρώτο μπορούμε να το σκεφτούμε σαν τον "αυτοκινητόδρομο" που μεταφέρει σχετικές πληροφορίες μέσα στην διαδοχική αλυσίδα του δικτύου. Μπορούμε να το σκεφτόμαστε και σαν την "μνήμη" του δικτύου. Μέσω αυτού, πληροφορίες από αρχικά χρονικά στάδια μπορούν να φτάσουν σε μετέπειτα χρονικά στάδια, μειώνοντας έτσι τις επιπτώσεις της βραχυπρόθεσμης μνήμης (short-term memory). Καθώς το κελί μνήμης (cell state) διανύει το ταξίδι του στις

χρονικές στιγμές, οι πληροφορίες εισάγονται ή αφαιρούνται από το κελί μνήμης (cell state) μέσω των πυλών. Οι πύλες, και στην περίπτωση των LSTMs, είναι διαφορετικά νευρωνικά δίκτυα που αποφασίζουν σε ποιες πληροφορίες θα επιτρέψουν να μπουν στο κελί μνήμης (cell state) και σε ποιες όχι. Επιπλέον, μπορούν να μάθουν ποιες πληροφορίες είναι σχετικές ώστε να ξεχαστούν ή να διατηρηθούν κατά την διάρκεια της εκπαίδευσης (training).

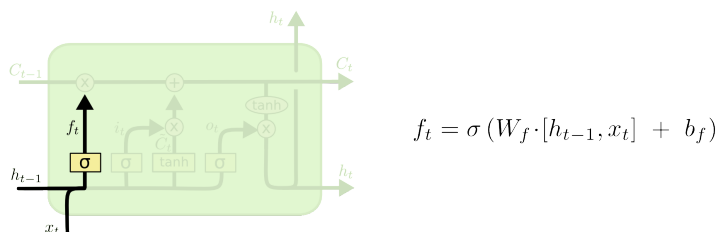
Οι πύλες χρησιμοποιούν για συνάρτηση ενεργοποίησης την sigmoid (σχήμα 3.11). Αυτό είναι βολικό καθώς το δίκτυο μαθαίνει ποια δεδομένα είναι σχετικά όποτε τα κρατάει καθώς πολλαπλασιάζεται με 1 (έξοδος συνάρτησης ενεργοποίησης) ή τα ξεχνάει πολλαπλασιάζοντας με 0 αντίστοιχα. Παρακάτω ακολουθεί μια σύντομη περιγραφή των πυλών από τις οποίες ένα LSTM δίκτυο αποτελείται.



Σχήμα 3.11: Η απεικόνιση της sigmoid function και η μαθηματική αναπαράσταση [9]

Πύλη Λήθης (Forget Gate)

Είναι η πρώτη πύλη που έχουμε. Αυτή η πύλη αποφασίζει ποια πληροφορία πρέπει να "πεταχτεί" και ποια να διατηρηθεί. Πληροφορίες από την προηγούμενη κρυφή κατάσταση (hidden state) και το παρούσα είσοδο περνάνε μέσα από την συνάρτηση ενεργοποίησης sigmoid. Όσο πιο κοντά στο 0 είναι η έξοδος της sigmoid τόσο περισσότερο πρέπει να αγνοήσουμε αυτή την πληροφορία, ενώ όσο πιο κοντά στο 1 πρέπει να την κρατήσουμε. Στο παρακάτω, σχήμα απεικονίζεται σχηματικά και μαθηματικά η συγκεκριμένη πύλη.

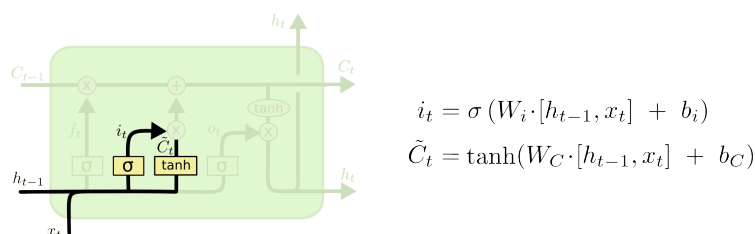


Σχήμα 3.12: Σχηματική απεικόνιση για Forget Gate και η μαθηματική αναπαράσταση [10]

Πύλη Εισόδου (Input Gate)

Στην συνέχεια, θέλουμε να αποφασίσουμε ποια νέα πληροφορία θέλουμε να κρατήσουμε στο κελί μνήμης (cell state). Αυτή η διαδικασία χωρίζεται σε δύο διακριτά μέρη. Αρχικά τον ρόλο αυτόν αναλαμβάνει η πύλη εισόδου, μέσω της οποίας περνάμε την παρούσα είσοδο

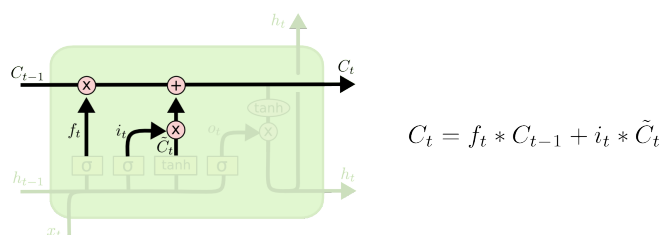
και τη προηγούμενη κρυφή κατάσταση (hidden state) από την συνάρτηση ενεργοποίησης sigmoid, έτσι ώστε η έξοδος να διαχωρίζει τις σημαντικές πληροφορίες για να ενημερωθούν (0 οι πιο ασήμαντες, 1 οι σημαντικές), i_t . Κατόπιν, περνάμε τις ίδιες εισόδους από μια συνάρτηση ενεργοποίησης tanh και δημιουργούμε ένα νέο διάνυσμα \tilde{C}_t με τις υποψήφιες νέες τιμές που μπορούν να προστεθούν στην παρούσα κατάσταση. Η γραφική και μαθηματική απεικόνιση των παραπάνω βρίσκεται στο σχήμα 3.13.



Σχήμα 3.13: Σχηματική απεικόνιση για Input Gate και η μαθηματική αναπαράσταση [10]

Κελί Μνήμης (Cell State)

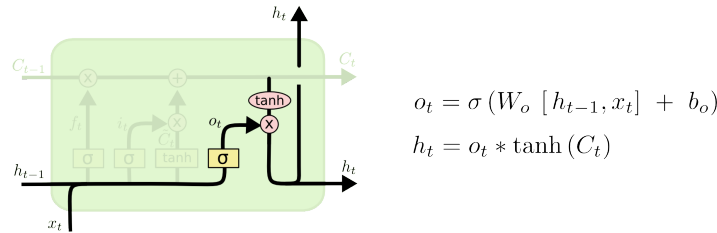
Τέλος, το μόνο που μένει είναι η ενημέρωση του κελιού μνήμης (cell state). Σε αυτό το βήμα απορρίπτουμε μη σχετικές πληροφορίες του παρελθόντος και προσθέτουμε πληροφορίες όπως είχε αποφασιστεί από τα προηγούμενα βήματα. Αυτό επιτυγχάνεται με τον πολλαπλασιασμό του προηγούμενου κελιού μνήμης (cell state) με την έξοδο του forget gate, έτσι αγνοούνται όσες τιμές πολλαπλασιάζονται με τιμές κοντά στο 0. Στην συνέχεια, υπολογίζεται η πρόσθεση με την έξοδο του input gate και έτσι ενημερώνεται το κελί μνήμης (cell state) με τις καινούριες τιμές που είναι σχετικές.



Σχήμα 3.14: Σχηματική απεικόνιση για Cell State και η μαθηματική αναπαράσταση [10]

Πύλη Εξόδου (Output Gate)

Η πύλη εξόδου είναι η τελευταία πύλη και αποφασίζει ποια θα είναι η επόμενη κρυφή κατάσταση (hidden state). Αρχικά, περνάμε την παρούσα είσοδο και την προηγούμενη κρυφή κατάσταση από μια sigmoid συνάρτηση ενεργοποίησης. Στην συνέχεια, περνάμε το καινούργιο κελί μνήμης (cell state) από μια συνάρτηση ενεργοποίησης tanh. Τέλος, πολλαπλασιάζουμε την έξοδο της συνάρτησης ενεργοποίησης tanh με αυτή της sigmoid ώστε να αποφασίσουμε τι πληροφορίες θα μεταφέρει η κρυφή κατάσταση. Τέλος, το καινούριο κελί μνήμης και η κρυφή κατάσταση προωθούνται στο επόμενο χρονικό βήμα.



Σχήμα 3.15: Σχηματική απεικόνιση για Hidden State και η μαθηματική αναπαράσταση [10]

Τα δίκτυα LSTM αποτέλεσαν βασικό κορμό για την δημιουργία διάφορων παραλλαγών, με μια από τις πιο γνωστές να είναι τα δίκτυα GRU που έχουν περιγραφεί σε προηγούμενο κεφάλαιο. Αρκετά διαδεδομένες παραλλαγές είναι τα Depth Gated RNNs [41] και αυτή που προτάθηκε από τους Gers και Schmidhuber το 2000 [42].

Οι Greff, et al [43] σε σχετική έρευνα μελέτησαν και σύγκριναν διαδεδομένες παραλλαγές των LSTM και βρήκαν ότι οι αποδόσεις τους είναι σχεδόν ίδιες. Αντίστοιχη έρευνα πραγματοποιήσαν και οι Jozefowicz, et al. [44] σύγκριναν πάνω από δέκα χιλιάδες αρχιτεκτονικές Αναδρομικών Δικτύων, βρίσκοντας ότι κάποιες από αυτές αποδίδουν καλύτερα σε συγκεκριμένες εργασίες.

3.3 Υβριδικά Νευρωνικά Δίκτυα

Όπως έχουμε αναφέρει, στόχος της παρούσας εργασίας είναι η ανάδειξη μοντέλων που μπορούν να επιλύσουν προβλήματα ταξινόμησης χρονοσειρών. Τα αναδρομικά νευρωνικά δίκτυα αποτελούν την βάση για την επίλυση προβλημάτων τέτοιας φύσεως, εξαιτίας τόσο της προσθήκης της διάστασης του χρόνου κατά την επίλυση του προβλήματος, όσο και της ικανότητας τους να αποθηκεύουν πληροφορία από προηγούμενες χρονικές στιγμές με σκοπό να την αξιοποιήσουν για βελτιωμένη απόδοση. Με απλά λόγια, θα χαρακτηρίζαμε τα RNNs ως τα καταλληλότερα δίκτυα για να ‘κρατούν’ χρονική πληροφορία.

Παρόλα αυτά στα πλαίσια της διπλωματικής μας ανάλυσης θα θέλαμε να προτείνουμε λίγο πιο ιδιαίτερες αρχιτεκτονικές νευρωνικών δικτύων για την αντιμετώπιση προβλημάτων ταξινόμησης χρονοσειρών. Τα μοντέλα αυτά τα ονομάζουμε Υβριδικά Νευρωνικά Δίκτυα (HNNs) [45] και αποτελούνται από τα αναδρομικά δίκτυα και από μία άλλη εξίσου διαδεδομένη κατηγορία νευρωνικών δικτύων, τα Συνελικτικά Νευρωνικά Δίκτυα CNNs [46].

Ωστόσο, σκοπός της παρούσας εργασία δεν είναι η λεπτομερής μελέτη των CNNs. Για αυτό το λόγο θα γίνει μία σύντομη παρουσίαση τους και θα εστιάσουμε στο πως μπορούν να συνδυαστούν με τα RNNs. Σκοπός μας είναι να καταφέρουμε να καταλήξουμε σε αρχιτεκτονικές που να καλύπτουν σε ικανοποιητικό βαθμό το πρόβλημα μας.

Στο Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης του ΕΜΠ έχουν εφαρμοστεί μοντέλα CNN-RNN με μονάδες LSTMs, GRUs, σε πολλές εφαρμογές υπολογιστικής όρασης [47], [48], [49], [50], ιατρικής διάγνωσης [51], [52], [53], [54], ανάλυσης χρονοσειρών για ανίχνευση βλαβών [55] ή πρόβλεψης της παραγωγής [56].

3.3.1 Εισαγωγή στα Συνελικτικά Νευρωνικά Δίκτυα

Σε αυτό το κεφάλαιο γίνεται εισαγωγή στα συνελικτικά νευρωνικά δίκτυα (CNN), μια ευρέως διαδεδομένη κατηγορία νευρωνικών δικτύων. Οι αρχιτεκτονικές που βασίζονται στα CNNs είναι πλέον πανταχού παρούσες στον τομέα της όρασης υπολογιστών και έχουν γίνει τόσο κυρίαρχες που σχεδόν κανείς σήμερα δεν θα μπορούσε να αναπτύξει μια εμπορική εφαρμογή ή να συμμετάσχει σε διαγωνισμό που σχετίζεται με την αναγνώριση εικόνας, την ανίχνευση αντικειμένων ή τη σημασιολογική τμηματοποίηση, χωρίς να βασίζεται σε αυτήν την προσέγγιση.

Τα σύγχρονα CNNs, όπως ονομάζονται στη γλώσσα των Νευρωνικών Δικτύων, οφείλουν το σχεδιασμό τους σε εμπνεύσεις από τη βιολογία, τη θεωρία των ομάδων και σε μια υγιή δόση πειραματικού μαστορέματος. Εκτός από την αποτελεσματικότητα του δείγματος στην επίτευξη μοντέλων υψηλής ακρίβειας προβλέψεων, τα CNNs τείνουν να είναι υπολογιστικά αποδοτικά, τόσο επειδή απαιτούν λιγότερες παραμέτρους από τις πλήρως συνδεδεμένες αρχιτεκτονικές (Fully Connected Layers) όσο και επειδή οι συνελίξεις είναι εύκολο να παραλληλιστούν μεταξύ των πυρήνων GPU. Κατά συνέπεια, οι μελετητές συχνά εφαρμόζουν τα CNNs όποτε είναι δυνατόν, και όλο και περισσότερο εμφανίζονται ως αξιόπιστοι ανταγωνιστές ακόμη και σε εργασίες με μονοδιάστατη δομή ακολουθίας, όπως ανάλυση ήχου, κειμένου και χρονοσειρών, όπου συμβατικά χρησιμοποιούνται επαναλαμβανόμενα νευρωνικά δίκτυα. Ορισμένες έξυπνες προσαρμογές των CNNs τα έφεραν επίσης σε επαφή με δεδομένα δομημένα σε γραφήματα και συστήματα προτάσεων [57].

Σύμφωνα με τις απαιτήσεις της παρούσας διπλωματικής θα ασχοληθούμε με τα μονοδιάστατα CNNs καθώς, όπως αναφέρθηκε είναι αξιόπιστα σε εργασίες χρονοσειρών και ακολουθιών. Παρόλα αυτά χρήσιμη κρίνεται για την κατανόηση των τεχνολογιών που χρησιμοποιήθηκαν, μια γενικότερη περιγραφή της έννοιας και της λειτουργίας των Συνελικτικών Νευρωνικών Δικτύων.

Συνελικτικό Επίπεδο (Convolutional Layer)

Η λειτουργία ενός συνελικτικού επιπέδου, όπως φανερώνει το όνομα του, στηρίζεται στην έννοια της συνέλιξης. Η συνέλιξη έχει ταυτιστεί στο πεδίο των Νευρωνικών με την συνέλιξη εικόνων, καθώς εκεί βρίσκουν την κύρια εφαρμογή τους τα CNNs.

Στο πεδίο της όρασης υπολογιστών και πιο συγκεκριμένα στις επεξεργασίας εικόνας, χρησιμοποιούμε τους όρους του πυρήνα (kernel) ή φίλτρου (filter). Ως πυρήνας μπορεί να οριστεί ένα παράθυρο καθορισμένης διάστασης $n \times n$ ίδιου βάθους με την εικόνα (για παράδειγμα 3 σε περίπτωση RGB εικόνας). Οι έννοιες του πυρήνα, του φίλτρου και του βάθους μπορούν να γενικευτούν και σε άλλα δεδομένα εκτός από εικόνες. Παρόλα αυτά επιλέγουμε να εξηγήσουμε την λειτουργία των CNNs πάνω σε δεδομένα -εικόνες για την καλύτερη κατανόησή τους.

Ο μαθηματικός ορισμός της διακριτής συνέλιξης μονοδιάστατων σημάτων δίνεται από την ακόλουθη σχέση:

$$(f * g)[n] = f[n] * g[n] = \sum_{m=-\infty}^{\infty} f[m]g[n - m], \quad (3.24)$$

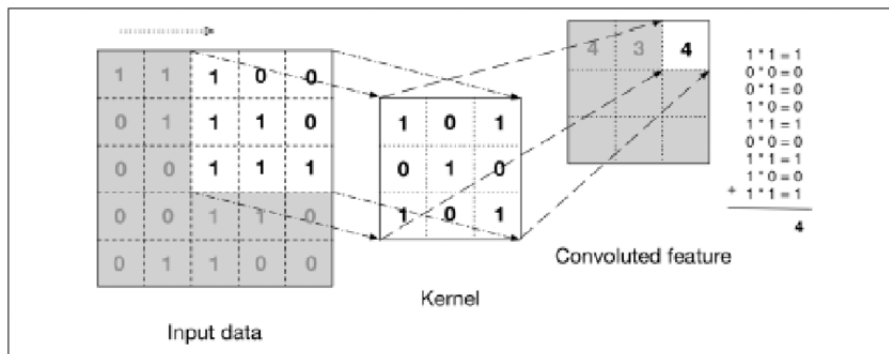
όπου f, g μονοδιάστατα διακριτά σήματα.

Κατά αντίστοιχο τρόπο ορίζεται και η συνέλιξη των δισδιάστατων διακριτών σημάτων μιας εικόνας f με έναν πυρήνα g ως εξής:

$$(f * g)[x, y] = f[x, y] * g[x, y] = \sum_{n1=-\infty}^{\infty} \sum_{n2=-\infty}^{\infty} f[n1, n2]g[x - n1, y - n2] \quad (3.25)$$

Από την σχέση 3.25 καταλαβαίνουμε ότι τα στοιχεία του πυρήνα πολλαπλασιάζονται με τα στοιχεία της εισόδου (pixels) σε κάθε περιοχή και τα αποτελέσματα τους τοποθετούνται στην κατάλληλη θέση στον πίνακα εξόδου. Η έξοδος κάθε τέτοιας πράξης ονομάζεται χάρτης ενεργοποίησης (activation map) ή χάρτης χαρακτηριστικών (feature map), καθώς η τιμή του χάρτη σε κάθε θέση εκφράζει την πιθανότητα με την οποία το επιθυμητό χαρακτηριστικό βρίσκεται σε αυτή την περιοχή της αρχικής εικόνας.

Στο παρακάτω σχήμα απεικονίζεται σχηματικά η πράξη της συνέλιξης έχοντας ένα παράδειγμα δεδομένων εισόδου και πυρήνα, υπολογίζεται η έξοδος, χάρτης χαρακτηριστικών (feature map).



Σχήμα 3.16: Σχηματική απεικόνιση του υπολογισμού της συνέλιξης [11]

Τα βάρη του πυρήνα αποτελούν εκπαιδευσιμες παραμέτρους του δικτύου. Σύμφωνα με τον τυπικό ορισμό της διακριτής συνέλιξης εικόνων, ο πυρήνας ολισθαίνει μόνο ένα χωρικό βήμα σε κάθε μετακίνησή του, στην πράξη όμως το βήμα ολίσθησης που χρησιμοποιείται (stride) μπορεί να είναι και μεγαλύτερο, θα αναφερθούμε πιο αναλυτικά σε αυτό παρακάτω. Ακόμη, συνήθως θέλουμε ως έξοδο πολλούς χάρτες χαρακτηριστικών καθώς θέλουμε να εξάγουμε αρκετά χαρακτηριστικά από μια εικόνα. Για τον λόγο αυτό χρησιμοποιούνται πάνω από ένας πυρήνας ανά συνελκτικό επίπεδο, έχοντας ως αποτέλεσμα η έξοδος του κάθε συνελκτικού επιπέδου να είναι μία τρισδιάστατη "εικόνα" μεγάλου βάθους, το οποίο αποτελείται από διαφορετικούς χάρτες ενεργοποίησης.

Επίπεδο Ενεργοποίησης (Activation Layer)

Τα περισσότερα συστήματα που προσεγγίζει ένα CNN είναι πραγματικά συστήματα, και συνεπώς η συμπεριφορά τους δεν είναι γραμμική. Επομένως χρειάζεται να εισάγουμε μη-γραμμικότητα στο δίκτυο μας. Για αυτόν τον λόγο, ένα επίπεδο ενεργοποίησης ακολουθεί το κάθε συνελκτικό επίπεδο, το οποίο εφαρμόζει στη έξοδό του μία συνάρτηση ενεργοποίησης ϕ . Η πιο διαδεδομένη συνάρτηση ενεργοποίησης είναι η Rectified Linear Unit (ReLU) και

δίνεται από την σχέση :

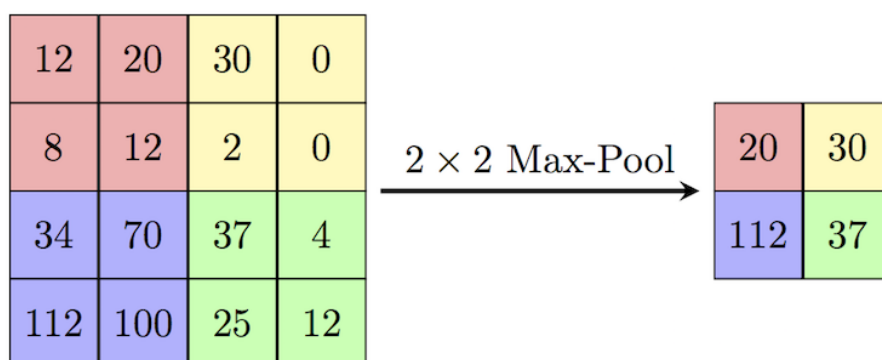
$$\phi(x) = \max(0, x) \quad (3.26)$$

Έχει αποδειχθεί ότι η συγκεκριμένη συνάρτηση επιταχύνει την διαδικασία εκπαίδευσης του δικτύου, απλοποιώντας το backpropagation.

Επίπεδο Υποδειγματοληψίας (Pooling Layer)

Τα επίπεδα υποδειγματοληψίας σκοπεύουν στην μείωση των διαστάσεων των χαρτών ενεργοποίησης που προκύπτουν από τα συνελκτικά επίπεδα. Πρακτικά, χωρίζουν τον χάρτη σε μη επικαλυπτόμενα μεταξύ τους τμήματα και για κάθε ένα από αυτά επιλέγουν μία αντιπροσωπευτική τιμή. Στην πιο συνηθισμένη εφαρμογή τους η τιμή αυτή υπολογίζεται ως το μέγιστο (υποδειγματοληψία μεγίστου - max pooling), αλλά μπορεί να χρησιμοποιηθεί και ο μέσος όρος (υποδειγματοληψία μέσου όρου) ή και τυχαία επιλογή (στοχαστική υποδειγματοληψία).

Πέρα από την μείωση των παραμέτρων και άρα την βελτίωση της ταχύτητας εκπαίδευσης, η ύπαρξη του επιπέδου υποδειγματοληψίας ελαχιστοποιεί και την πιθανότητα υπερεκπαίδευσης του δικτύου. Αντίστοιχη της υποδειγματοληψίας μεγίστου που έχει αποδειχθεί σημαντικά χρήσιμη είναι η υποδειγματοληψία μέσου (average pooling).



Σχήμα 3.17: Σχηματική απεικόνιση του υπολογισμού της υποδειγματοληψίας μεγίστου [12]

Επίπεδο Κανονικοποίησης Παρτίδας (Batch Normalization Layer)

Γενικά, τα Συνελκτικά Νευρωνικά Δίκτυα αποτελούνται από διαδοχικά επίπεδα συνελίξεων, με την έξοδο του κάθε επιπέδου να αποτελεί την είσοδο του επόμενου. Η δομή αυτή, στις περιπτώσεις Συνελκτικών Νευρωνικών Δικτύων με πολλά διαδοχικά επίπεδα, συχνά οδηγεί σε ένα φαινόμενο που ονομάζεται "internal covariate shift", κατά το οποίο η προσαρμογή των παραμέτρων του δικτύου κατά την εκπαίδευση προκαλεί αλλαγή στην κατανομή των ενεργοποιήσεων των διαφόρων επιπέδων (συνήθως των τελευταίων).

Για να αποφευχθεί αυτό το φαινόμενο, συνήθως προσθέτουμε επίπεδα κανονικοποίησης παρτίδας, τα οποία εξασφαλίζουν την κανονικοποίηση των δεδομένων κάθε παρτίδας (batch) σε κάθε επίπεδο. Η κανονικοποίηση γίνεται με χρήση των στατιστικών χαρακτηριστικών του υποσυνόλου, ώστε ο μέσος όρος να ισούται με 0 και η διακύμανση να είναι μοναδιαία.

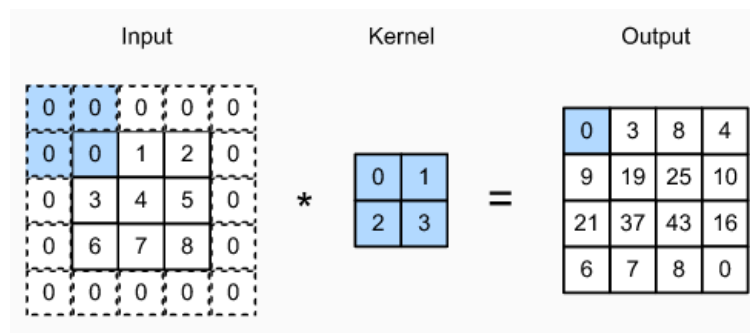
Γέμισμα και Βήμα (Padding και Stride)

Εν γένει η έξοδος ενός συνελκτικού επιπέδου καθορίζεται από τις διαστάσεις της εισόδου και τις διαστάσεις του πυρήνα. Έστω ότι η εικόνα στην είσοδο του συνελκτικού επιπέδου έχει διαστάσεις $(n_h \times n_w)$ και ο πυρήνας $(k_h \times k_w)$, τότε σύμφωνα με όσα περιγράψαμε η έξοδος θα έχει διαστάσεις $(n_h - k_h + 1 \times n_w - k_w + 1)$.

Σε αρκετές περιπτώσεις, ενσωματώνουμε τεχνικές, συμπεριλαμβανομένης των padding και slide, που επηρεάζουν το μέγεθος της εξόδου. Ως κίνητρο, για την χρήση τέτοιων τεχνικών σημειώνεται, ότι δεδομένου ότι οι πυρήνες έχουν γενικά πλάτος και ύψος μεγαλύτερο από 1, μετά την εφαρμογή πολλών διαδοχικών συνελκτικών επιπέδων, τείνουμε να τελειώσουμε με εξόδους πολύ μικρότερες από τις εισόδους μας. Τέτοιου είδους προβλήματα τα διαχειρίζεται πολύ καλά η εφαρμογή padding στις εικόνες εισόδου.

Σε άλλες περιπτώσεις, μπορεί να θέλουμε να μειώσουμε δραστικά τη διάσταση εισόδου, π.χ., αν θεωρήσουμε ότι η αρχική ανάλυση εισόδου είναι δυσκίνητη. Η χρήση slides είναι μια δημοφιλής τεχνική που μπορεί να βοηθήσει σε αυτές τις περιπτώσεις.

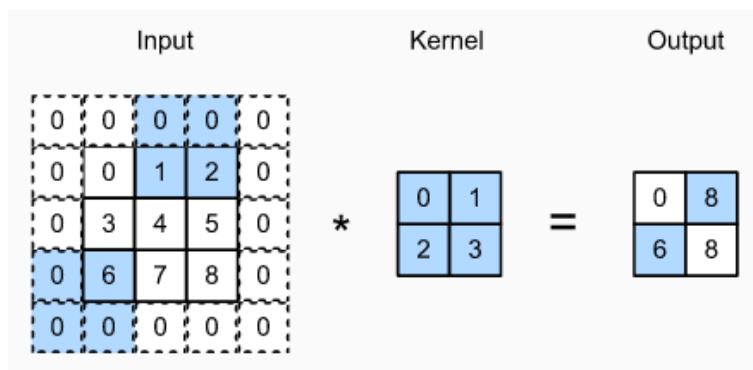
Όπως περιεγράφηκε παραπάνω, ένα δύσκολο θέμα κατά την εφαρμογή των συνελκτικών επιπέδων είναι ότι τείνουμε να χάνουμε pixel στην περίμετρο της εικόνας μας. Δεδομένου ότι χρησιμοποιούμε συνήθως μικρούς πυρήνες, για κάθε δεδομένη συνέλιξη, μπορεί να χάσουμε μόνο μερικά εικονοστοιχεία, αλλά αυτό μπορεί να αθροιστεί καθώς εφαρμόζουμε πολλά διαδοχικά στρώματα. Μια απλή λύση σε αυτό το πρόβλημα είναι να προσθέσουμε επιπλέον εικονοστοιχεία padding γύρω από το όριο της εικόνας εισόδου μας, αυξάνοντας έτσι το πραγματικό μέγεθος της εικόνας. Τυπικά θέτουμε την τιμή των έξτρα εικονοστοιχείων στο 0. Έστω ότι προσθέτουμε ένα σύνολο p_h σειρών γεμίματος (περίπου το μισό από πάνω και το μισό από κάτω) και ένα σύνολο από τις στήλες p_w γεμίματος (περίπου το μισό στα αριστερά και τα μισά στα δεξιά), το σχήμα εξόδου θα είναι $(n_h - k_h + p_h + 1 \times n_w - k_w + p_w + 1)$



Σχήμα 3.18: Παράδειγμα υπολογισμού συνελκτικού επιπέδου με τεχνική γεμίματος. [13]

Κατά τον υπολογισμό της συνέλιξης, ξεκινάμε με το παράθυρο συνέλιξης στην επάνω αριστερή γωνία του τανυστή εισόδου και, στη συνέχεια, το σύρουμε σε όλες τις θέσεις τόσο προς τα κάτω όσο και προς τα δεξιά. Στο επεξηγηματικό παράδειγμα του συνελκτικού επιπέδου, προεπιλέξαμε την ολίσθηση ενός στοιχείου τη φορά. Ωστόσο, μερικές φορές, είτε για υπολογιστική αποδοτικότητα είτε επειδή θέλουμε να κάνουμε δειγματοληψία, μετακινούμε το παράθυρό μας περισσότερα από ένα στοιχεία κάθε φορά, παραλείποντας τις ενδιάμεσες θέσεις.

Αναφερόμαστε στον αριθμό των στοιχείων των γραμμών και στηλών που ολισθαίνει το παράθυρο της συνέλιξης σε κάθε συνελκτικό βήμα, ως βήμα (slide). Μέχρι στιγμής, έχουμε χρησιμοποιήσει βήματα 1, τόσο για ύψος όσο και για πλάτος. Μερικές φορές, μπορεί να θέλουμε να χρησιμοποιήσουμε μεγαλύτερο βήμα. Το 3.24 δείχνει μια δισδιάστατη λειτουργία συνέλιξης με ένα βήμα 3 κάθετα και 2 οριζόντια. Το σχήμα εξόδου θα είναι $((n_h - k_h + p_h + 1)/s_h) \times ((n_w - k_w + p_w + 1)/s_w)$



Σχήμα 3.19: Παράδειγμα υπολογισμού συνέλικτικού επιπέδου με οριζόντιο βήμα 2 και κάθετο 3. [13]

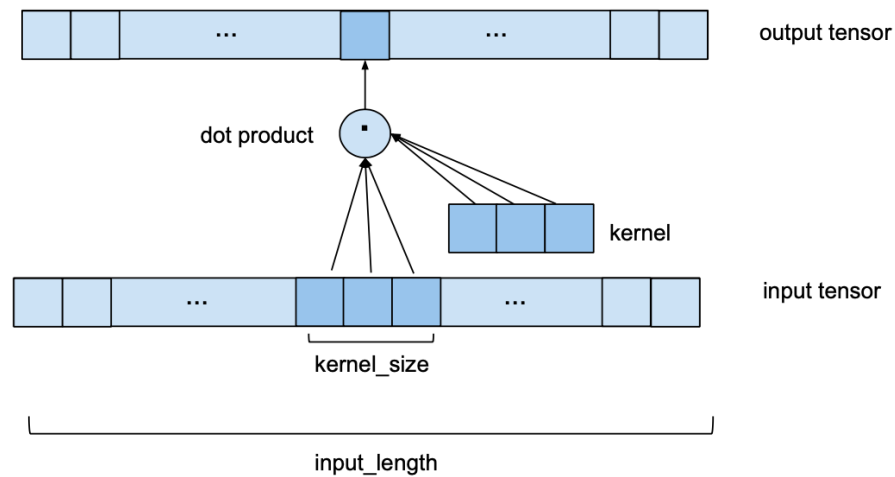
3.3.2 Συνελκτικά Δίκτυα Μιας Διάστασης (1D CNNs)

Παρόλο που τα συνελκτικά δίκτυα αναπτύχθηκαν για προβλήματα ταξινόμησης εικόνων, όπου τα μοντέλα μάθαιναν μια εσωτερική αναπαράσταση μιας δισδιάστατης εισόδου, μπορεί να αξιοποιηθούν και από μια αντίστοιχη διαδικασία για διαδοχικά δεδομένα, όπως έχουμε ήδη αναφέρει.

Σε αυτή την περίπτωση εφαρμόζονται τα μονοδιάστατα συνελκτικά δίκτυα. Στο πρόβλημα που καλούμαστε να επιλύσουμε στα πλαίσια της παρούσας διπλωματικής εργασίας τα δεδομένα από το επιταχυνσιόμετρο (accelerometer) και το γυροσκόπιο (gyroscope) δίνουν είσοδο στο μοντέλο μας αυτής της μορφής. Συνεπώς, το μοντέλο μαθαίνει να εξάγει χαρακτηριστικά από μια αλληλουχία παρατηρήσεων και πώς να τα αντιστοιχεί σε εσωτερικά χαρακτηριστικά για τους διαφόρους τύπους ανθρώπινων δραστηριοτήτων.

Ένα μονοδιάστατο συνελκτικό δίκτυο λαμβάνει ως είσοδο έναν τρισδιάστατο τανυστή και έχει ως έξοδο επίσης ένα τρισδιάστατο τανυστή. Η διαστάσεις της εισόδου έχουν την μορφή (batch size, input length, input channels) ενώ της εξόδου αντίστοιχα (batch size, output length, output channels). Για να κατανοήσουμε τον τρόπο που ένα απλό μονοδιάστατο συνελκτικό στρώμα μετατρέπει τα δεδομένα εισόδου σε εξόδου θα μελετήσουμε ένα στοιχείο από την παρτίδα (batch). Προφανώς η ίδια διαδικασία ακολουθείτε και για κάθε στοιχείο της παρτίδας. Άς ξεκινήσουμε με την πιο απλή περίπτωση που τα κανάλια εισόδου είναι ίδια με αυτά της εξόδου (input channels = output channels) και ίσα με 1. Η εικόνα 3.20 δείχνει πώς υπολογίζεται ένα στοιχείο του τανυστή εξόδου, με βάση ένα στοιχείο από την παρτίδα εισόδου.

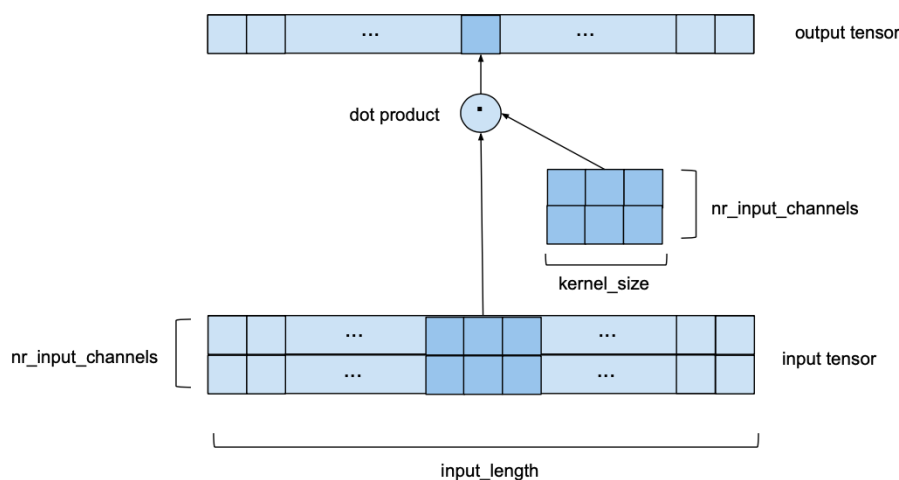
Όπως μπορούμε να δούμε για τον υπολογισμό ενός στοιχείου της εξόδου, εξετάζουμε μια σειρά διαδοχικών στοιχείων εισόδου μήκους ίσου με το μέγεθος πυρήνα (kernel size). Στο συγκεκριμένο παράδειγμα έχουμε επιλέξει μέγεθος πυρήνα ίσο με 3. Για να αποκτήσουμε



Σχήμα 3.20: Υπολογισμός συνέλιξης για 1D-CNN με $input\ channels = output\ channels = 1$ [14]

την έξοδο χρειάζεται να υπολογίσουμε το εσωτερικό γινόμενο της υποακολουθίας (σκιασμένα τετράγωνα) της εισόδου με το διάνυσμα του πυρήνα που περιέχει τα βάρη εκπαίδευσης και έχουν το ίδιο μήκος. Για να υπολογίσουμε το επόμενο στοιχείο της εξόδου εφαρμόζουμε την ίδια διαδικασία με πριν, με την διαφορά ότι μετακινούμε το παράθυρο του πυρήνα μια θέση δεξιότερα ή όσες θέσεις καθορίζονται από το kernel stride. Αξίζει να σημειωθεί ότι το ίδιο σύνολο βαρών εκπαίδευσης χρησιμοποιούνται για τον υπολογισμό κάθε εξόδου στο ίδιο στρώμα του μονοδιάστατου συνελκτικού δικτύου.

Ας επικεντρωθούμε τώρα στην περίπτωση που έχουμε πολλαπλά κανάλια εισόδου ($input\ size > 1$). Σε αυτή την περίπτωση η διαδικασία που περιγράφηκε παραπάνω επαναλαμβάνεται για κάθε κανάλι, αλλά αυτή την φορά με διαφορετικό πυρήνα σε κάθε κανάλι. Συνεπώς, τα βάρη εκπαίδευσης του πυρήνα είναι ίσα με το γινόμενο του μεγέθους του πυρήνα με τον αριθμό των καναλιών εισόδου. Το τελικό στοιχείο εξόδου, σε αυτή την περίπτωση, θα υπολογίζεται ως το άθροισμα των επιμέρους εσωτερικών γινομένων μιας υποακολουθίας εισόδου με τα βάρη του πυρήνα για κάθε κανάλι εισόδου. Στο παρακάτω σχήμα φαίνεται ένα παράδειγμα υπολογισμού ενός στοιχείου της εξόδου, ενώ η είσοδος έχει δύο κανάλια. Επιπλέον, αντί για μονοδιάστατα διανύσματα πυρήνα θεωρούμε ένα πίνακα πυρήνα δύο διαστάσεων (kernel size, input channels).



Σχήμα 3.21: Υπολογισμός συνέλιξης για 1D-CNN με $input\ channels > 1$ [14]

Στην περίπτωση τώρα που τόσο τα κανάλια της εισόδου όσο και της εξόδου είναι περισσότερα από 1, η παραπάνω διαδικασία επαναλαμβάνεται με διαφορετικό πίνακα πυρήνα για κάθε κανάλι εξόδου. Αυτό συμβαίνει γιατί σε κάθε κανάλι εξόδου επιθυμούμε να φιλτράρουμε διαφορετικά χαρακτηριστικά της εισόδου. Συνεπώς, τα βάρη του πυρήνα ή αλλιώς τα βάρη εκπαίδευσης θα είναι ίσα με το γινόμενο του μεγέθους του πυρήνα με τον αριθμό των καναλιών εισόδου και εξόδου ($kernel\ size * input\ channels * output\ channels$).

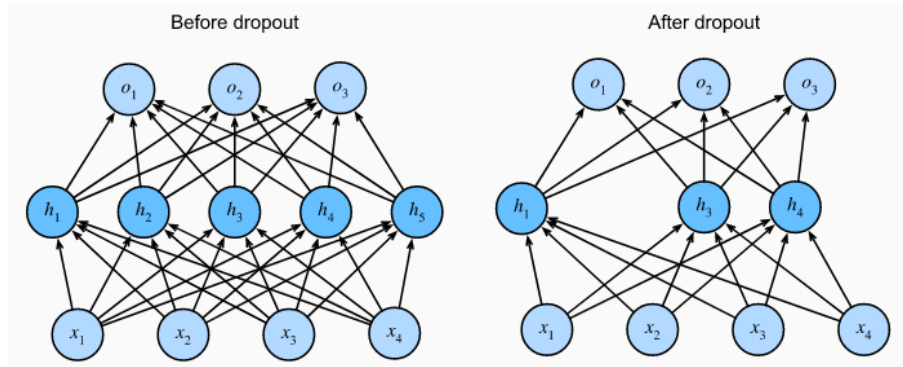
3.3.3 Επίπεδο Εγκατάλειψης

Ένα ακόμη επίπεδο που χρησιμοποιείται συχνά στις σύγχρονες αρχιτεκτονικές είναι το επίπεδο εγκατάλειψης. Το επίπεδο εγκατάλειψης μπορεί να ακολουθεί οποιοδήποτε επίπεδο εκ των Συνελικτικών ή Αναδρομικών νευρωνικών δικτύων.

Πιο συγκεκριμένα όταν εφαρμόζουμε εγκατάλειψη σε ένα κρυφό επίπεδο πρακτικά μηδενίζουμε κάθε κρυφή μονάδα με πιθανότητα p και το αποτέλεσμα μπορεί να θεωρηθεί ως ένα δίκτυο που περιέχει μόνο ένα υποσύνολο των αρχικών νευρώνων. Στο σχήμα 3.22, τα h_2 και h_5 αφαιρούνται. Κατά συνέπεια, ο υπολογισμός των εξόδων δεν εξαρτάται πλέον από το h_2 ή το h_5 και η αντίστοιχη κλίση τους εξαφανίζεται επίσης όταν εκτελείται αντίστροφη αναπαράγωγή. Με αυτόν τον τρόπο, ο υπολογισμός του επιπέδου εξόδου δεν μπορεί να εξαρτάται υπερβολικά από κανένα στοιχείο του h_1, \dots, h_5 και συνεπώς λειτουργεί σαν αποτρεπτικός παράγοντας για την υπερεκπαίδευση του δικτύου μας.

3.3.4 Ζεύξη Επαναληπτικών και Συνελικτικών Νευρωνικών Δικτύων (CNN-RNNs)

Σε αυτή την παράγραφο θα συνδυάσουμε τα συνελικτικά δίκτυα με τα επαναληπτικά των ενότητων 3.2 και 3.3.1. Ο συνδυασμός των CNN με τα RNN ενδείκνυται για την αντιμετώπιση προβλημάτων ταξινόμησης ακολουθιών, όπου τα δεδομένα είναι στη μορφή χωρικών εικόνων και βίντεο. Θα μπορούσαμε να πούμε, ότι συνδυάζοντας αυτές τις δύο αρχιτεκτονικές έχουμε τη δυνατότητα τόσο να εξαγάγουμε τα σημαντικότερα από τα χαρακτηριστικά των δεδομένων



Σχήμα 3.22: Κρυμμένο επίπεδο πριν και μετά το επίπεδο εγκατάλειψης [15]

μας, όσο και να αξιοποιήσουμε αυτά σε μία χρονική σειρά διατηρώντας την όποια πληροφορία χρειαζόμαστε από το παρελθόν. Σε αυτή την εργασία παρουσιάζουμε έναν τρόπο με τον οποίο μπορούμε να ενώσουμε αυτές τις 2 κατηγορίες νευρωνικών δικτύων. Ο τρόπος είναι η απλή τοποθέτηση για αρχή του συνελκτικού δικτύου (για το φιλτράρισμα των σημαντικότερων χαρακτηριστικών των δεδομένων εισόδου) και εν συνεχεία επαναληπτικού δικτύου που θα διαχειριστεί αυτά τα χαρακτηριστικά ως χρονική ακολουθία (και όχι το σύνολο των δεδομένων όπως ένα απλό RNN).

Αυτή η αρχιτεκτονική αναφερόταν αρχικά ως μακροπρόθεσμο επαναλαμβανόμενο δίκτυο μετατροπής ή μοντέλο LRCN, αν και θα χρησιμοποιήσουμε το πιο γενικό όνομα "CNN RNN" για να αναφερθούμε σε RNNs που χρησιμοποιούν ένα CNN ως εμπρόσθιο μέρος [58].

Η αρχιτεκτονική αυτή είναι αρκετά απλή καθώς απλώς τοποθετεί σε σειρά τα δύο πολύ γνωστά μας είδη δικτύων. Το μόνο που αξίζει να προσέξουμε είναι ότι το κομμάτι της εξόδου του πρώτου δικτύου αποτελεί την είσοδο για το δεύτερο. Τα CNN-RNNs αναπτύχθηκαν για προβλήματα πρόβλεψης οπτικών χρονικών σειρών και για την εφαρμογή δημιουργίας περιγραφής κειμένου από ακολουθίες εικόνων (π.χ. βίντεο). Συγκεκριμένα, τα προβλήματα :

- Αναγνώριση Δραστηριότητας
Δημιουργία κειμενικής περιγραφής μιας δραστηριότητας από μια ακολουθία εικόνων.
- Περιγραφή εικόνας
Δημιουργία κειμενικής περιγραφής από μια απλή εικόνα.
- Περιγραφή Βίντεο
Δημιουργία κειμενικής περιγραφής από μια ακολουθία εικόνων.
- Επεξεργασία φυσικής γλώσσας
Αναγνώριση ομιλίας ή πρόβλεψη χρονοσειρών όπου το CNN εξάγει χαρακτηριστικά και το RNN επεξεργάζεται τα δεδομένα [59].

Άλλα προβλήματα που αντιμετωπίζονται επιτυχώς με CNN-RNNs είναι η εκμάθηση της τάσης στις χρονοσειρές [60] και η πρόβλεψη απόδοσης παραγωγής [61].

Η δομή των CNN-RNNs που χρησιμοποιείται στην παρούσα διπλωματική αποτελείται από 3 ανεξάρτητα κομμάτια, το CNN μοντέλο, το RNN μοντέλο, και το πλήρως διασυνδεδεμένο (Dense) στρώμα εξόδου. Σημειώνεται ότι ως επαναληπτικό δίκτυο μπορεί να χρησιμοποιηθεί

οποιαδήποτε από τις αρχιτεκτονικές που έχουμε περιγράψει στις ενότητες 3.1 και 3.2. Το παρακάτω σχήμα δείχνει πως αυτά τα κομμάτια συνδέονται μεταξύ τους.



Σχήμα 3.23: Δομή CNN-RNNs

3.4 Χρονικά Συνελικτικά Δίκτυα (Temporal Convolutional Network)

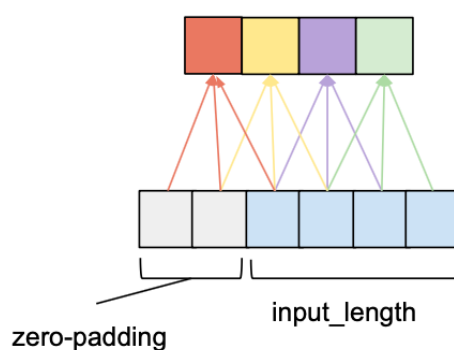
Μέχρι πρόσφατα, το θέμα της μοντελοποίησης ακολουθιακών δεδομένων στα πλαίσια της βαθιάς μάθησης σχετιζόταν σε μεγάλο βαθμό με επαναλαμβανόμενες αρχιτεκτονικές νευρωνικών δικτύων όπως τα LSTMs και τα GRUs. Όμως οι Bai et al. [62] προτείνουν ότι αυτός ο τρόπος σκέψης είναι απαρχαιωμένος και ότι τα συνελικτικά δίκτυα πρέπει να λαμβάνονται υπόψη ως ένας από τους κύριους υποψηφίους κατά τη μοντελοποίηση διαδοχικών δεδομένων. Ακόμη, ήταν σε θέση να δείξουν ότι τα συνελικτικά δίκτυα μπορούν να επιτύχουν καλύτερη απόδοση από τα RNNs σε πολλές εργασίες/προβλήματα, αποφεύγοντας τα κοινά μειονεκτήματα των αναδρομικών μοντέλων, όπως το πρόβλημα της έκρηξης/εξαφάνισης της κλίσης ή η έλλειψη διατήρησης μνήμης. Επιπλέον, η χρήση ενός συνελικτικού δικτύου αντί ενός αναδρομικού μπορεί να οδηγήσει σε βελτιώσεις απόδοσης καθώς επιτρέπει τον παράλληλο υπολογισμό των εξόδων. Η αρχιτεκτονική που προτείνουν ονομάζεται Temporal Convolutional Network (TCN) και θα εξηγηθεί στην παρούσα ενότητα.

Ένα TCN, συντομογραφία του Temporal Convolutional Network δηλαδή Χρονικό Συνελικτικό δίκτυο, αποτελείται από διασταλμένα (dilated), αιτιολογικά (causal) 1D συνελικτικά στρώματα με τα ίδια μήκη εισόδου και εξόδου. Οι παρακάτω υποενότητες αναλύουν λεπτομερώς τι σημαίνουν στην πραγματικότητα αυτοί οι όροι.

3.4.1 Αιτιώδης συνέλιξη (Causal Convolution)

Για να είναι αιτιώδης ή αιτιολογική μια συνελικτική στρώση, θα πρέπει για κάθε i του συνόλου $(0, \dots, \text{μήκος εισόδου} - 1)$ το στοιχείο i της ακολουθίας εξόδου να μπορεί να εξαρτάται μόνο από τα στοιχεία της ακολουθίας εισόδου με δείκτες $(0, \dots, i)$. Με άλλα λόγια, ένα στοιχείο στην ακολουθία εξόδου πρέπει να εξαρτάται μόνο από στοιχεία που έρχονται πριν από αυτό στην ακολουθία εισόδου. Όπως αναφέρθηκε προηγουμένως τα μήκη της εισόδου και της εξόδου θα πρέπει να είναι ίδια. Για να διασφαλίσουμε ότι ένας τανυστής εξόδου έχει το ίδιο μήκος με τον τανυστή εισόδου, πρέπει να εφαρμόσουμε μηδενικό γέμισμα (zero-padding). Εάν εφαρμόσουμε μηδενικό γέμισμα μόνο στην αριστερή πλευρά του τανυστή εισόδου, τότε θα διασφαλιστεί και η αιτιώδης συνέλιξη. Για να το καταλάβουμε αυτό καλύτερα, ας λάβουμε υπόψη μας το δεξιότερο (τελευταίο) στοιχείο της εξόδου. Αν θέσουμε ως δεδομένο ότι δεν υπάρχει γέμισμα (padding) στη δεξιά πλευρά της ακολουθίας εισόδου,

το τελευταίο στοιχείο από το οποίο εξαρτάται είναι το τελευταίο στοιχείο της εισόδου. Τώρα ας εξετάσουμε το δεύτερο από το τελευταίος στοιχείο της ακολουθίας εξόδου. Το παράθυρο του πυρήνα μετατοπίζεται προς τα αριστερά κατά ένα σε σύγκριση με το τελευταίο στοιχείο εξόδου, πράγμα που σημαίνει ότι η τελευταία εξάρτησή του στην ακολουθία εισόδου είναι το δεύτερο από το τελευταίος στοιχείο της ακολουθίας εισόδου. Ακολουθώντας επαγωγικά την ίδια διαδικασία συμπεραίνουμε ότι για κάθε στοιχείο της ακολουθίας εξόδου, η τελευταία εξάρτησή του στην ακολουθία εισόδου έχει τον ίδιο δείκτη με το ίδιο. Στο σχήμα 3.24 φαίνεται ένα παράδειγμα μήκος εισόδου 4 και μέγεθος πυρήνα 3.



Σχήμα 3.24: Παράδειγμα Causal Convolution [16]

Αν δεν υπήρχε γέμισμα στα αριστερά θα παρατηρούσαμε από τον υπολογισμό του πρώτου στοιχείου εξόδου ότι χρειάστηκε να λάβουμε υπόψιν μας το δεύτερο και το τρίτο στοιχείο της εισόδου επομένως σε μια τέτοια περίπτωση δεν θα ίσχυε η αιτιολογική συνέλιξη όπως την ορίσαμε στην αρχή της υποενοότητας αυτής. Συνεπώς, μπορούμε να καταλήξουμε στο συμπέρασμα ότι το γέμισμα που απαιτείται στα αριστερά δεν θα πρέπει να έχει μήκος μικρότερο ή ίσο με το μέγεθος του πυρήνα - 1 έτσι ώστε να ικανοποιείτε η αιτιολογική συνέλιξη.

Σημειώστε ότι πρόκειται ουσιαστικά για την ίδια αρχιτεκτονική με το νευρωνικό δίκτυο χρονικής καθυστέρησης που προτάθηκε πριν από σχεδόν 30 χρόνια από τους Waibel et al (1989) [63].

Ένα σημαντικό μειονέκτημα αυτού του βασικού σχεδιασμού είναι ότι για να επιτύχουμε με αποτελεσματικότητα να επεξεργαστούμε χρονοσειρές μεγάλου μήκους, χρειαζόμαστε ένα εξαιρετικά βαθύ δίκτυο ή πολύ μεγάλα φίλτρα/μεγέθη πυρήνα, κανένα από τα οποία δεν ήταν ιδιαίτερα εφικτό κατά την πρώτη εισαγωγή των μεθόδων. Έτσι, στην επόμενη υποενοότητα, περιγράφουμε πώς μπορούν να ενσωματωθούν τεχνικές από σύγχρονες συνελκτικές αρχιτεκτονικές σε ένα TCN για να επιτρέψουν τόσο την ύπαρξη πολύ βαθιών δικτύων όσο και την επεξεργασία χρονοσειρών μεγάλου μήκους.

3.4.2 Διαστολή (Dilation)

Ένα επιθυμητό ποιοτικά χαρακτηριστικό ενός μοντέλου πρόβλεψης είναι η τιμή μιας συγκεκριμένης καταχώρισης στην έξοδο να εξαρτάται από όλες τις προηγούμενες καταχωρήσεις της εισόδου, δηλαδή όλες τις καταχωρήσεις που έχουν έναν δείκτη μικρότερο ή ίσο με τον εαυτό του. Αυτό επιτυγχάνεται όταν το δεκτικό πεδίο, που σημαίνει το σύνολο των

καταχωρήσεων της αρχικής εισόδου που επηρεάζουν μια συγκεκριμένη τιμή της εξόδου, έχει μέγεθος ίσο με το μέγεθος της εισόδου. Αυτό το ονομάζουμε επίσης και «πλήρη κάλυψη ιστορικού». Όπως είδαμε και προηγουμένως ένα απλό μονοδιάστατο συνελκτικό επίπεδο κάνει το κάθε στοιχείο της εξόδου να επηρεάζεται από τα στοιχεία της εισόδου που έχουν δείκτη μικρότερο ή ίσο με το μέγεθος πυρήνα. Για παράδειγμα, εάν έχουμε ένα μέγεθος πυρήνα 3, το 5ο στοιχείο στην έξοδο θα εξαρτηθεί από τα στοιχεία 3, 4 και 5 της εισόδου. Αυτή η προσέγγιση διευρύνεται όταν στοιβάζουμε πολλά επίπεδα το ένα πάνω στο άλλο.

Μπορούμε να γενικεύσουμε τον υπολογισμό του δεκτικού πεδίου (r) για ένα δίκτυο με n μονοδιάστατα συνελκτικά δίκτυα και μέγεθος πυρήνα σε κάθε ένα από αυτά ίσο με k ως:

$$r = 1 + n * (k - 1) \quad (3.27)$$

Για να υπολογίσουμε πόσα επίπεδα χρειάζονται για πλήρη κάλυψη, μπορούμε να θέσουμε το μέγεθος του δεκτικού πεδίου να είναι ίσο με το μέγεθος της εισόδου (l) και να λύσουμε ως προς τον αριθμό των επιπέδων n (πρέπει να στρογγυλοποιήσουμε σε περίπτωση μη ακέραιων τιμών):

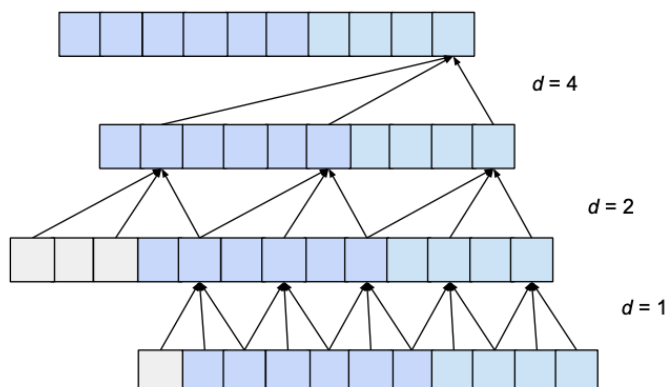
$$n = \frac{l - 1}{k - 1} \quad (3.28)$$

Αυτό σημαίνει ότι, δεδομένου ενός σταθερού μεγέθους πυρήνα, ο αριθμός των επιπέδων που απαιτούνται για πλήρη κάλυψη ιστορικού είναι γραμμικός συνδυασμός του μήκους του τανυστή εισόδου, πράγμα που θα οδηγήσει σε δίκτυα που γίνονται πολύ βαθιά πολύ γρήγορα, οδηγώντας σε μοντέλα με πολύ μεγάλο αριθμό παραμέτρων που χρειάζονται περισσότερο χρόνο για να εκπαιδευτούν. Επιπλέον, ένας μεγάλος αριθμός επιπέδων έχει αποδειχθεί ότι οδηγεί σε προβλήματα που σχετίζονται με την κλίση της συνάρτησης απωλειών όπως έκρηξης κλίσης. Ένας τρόπος αύξησης του μεγέθους του δεκτικού πεδίου διατηρώντας παράλληλα τον αριθμό των επιπέδων σχετικά μικρό είναι η εισαγωγή της διαστολής (dilation) στο συνελκτικό δίκτυο.

Η διαστολή στο πλαίσιο ενός συνελκτικού επιπέδου αναφέρεται στην απόσταση μεταξύ των στοιχείων της ακολουθίας εισόδου που χρησιμοποιούνται για τον υπολογισμό μιας καταχώρισης της ακολουθίας εξόδου. Έτσι, ένα συμβατικό συνελκτικό επίπεδο θα μπορούσε να θεωρηθεί ως 1-διασταλμένο επίπεδο, αφού τα στοιχεία εισόδου που χρησιμοποιούνται για τον υπολογισμό μιας τιμής εξόδου είναι γειτονικά. Σε αυτή την περίπτωση, ένα επίπεδο με διαστολή d και με μέγεθος πυρήνα k έχει ένα δεκτικό πεδίο που εκτείνεται σε μήκος $1 + d * (k - 1)$ για ένα στοιχείο της εξόδου. Εάν το d είναι σταθερό, τότε θα εξακολουθεί να απαιτείται ένας γραμμικός συνδυασμός στο μήκος του τανυστή εισόδου από συνελκτικά επίπεδα n για να επιτευχθεί πλήρης κάλυψη δεκτικού πεδίου. Αν χρησιμοποιήσουμε την διαστολή με έναν τέτοιο τρόπο καταφέρνουμε απλώς να μειώσουμε τον αριθμό n .

Αυτό το πρόβλημα μπορεί να αντιμετωπιστεί αυξάνοντας την τιμή του d εκθετικά καθώς ανεβαίνουμε στα στοιβαγμένα συνελκτικά επίπεδα/στρώματα. Για αυτό, επιλέγουμε έναν ακέραιο σταθερό ως βάση διαστολής (dilation base) που θα μας επιτρέψει να υπολογίσουμε τη διαστολή d ενός συγκεκριμένου επιπέδου σε συνάρτηση με τον αριθμό των επιπέδων κάτω από αυτό, i , ως $d = b^i$. Το παρακάτω σχήμα δείχνει ένα δίκτυο με μήκος εισόδου 10, μέγεθος πυρήνα 3 και βάση διαστολής 2, το οποίο έχει ως αποτέλεσμα 3 διασταλμένα

συνελκτικά επίπεδα για πλήρη κάλυψη.



Σχήμα 3.25: Δίκτυο με διασταθμένα συνελκτικά στρώματα με πλήρη κάλυψη [14]

Στο παραπάνω σχήμα παρουσιάζεται μόνο η συμβολή των στοιχείων της εισόδου που επηρεάζουν την τελευταία τιμή της εξόδου. Ομοίως, εμφανίζονται μόνο τα μηδενικά γεμίσματα (zero-padding) που είναι απαραίτητα για τον υπολογισμό της τελευταίας τιμής εξόδου. Σαφώς, η τελευταία τιμή εξόδου εξαρτάται από όλο το σύνολο της εισόδου. Στην πραγματικότητα, δεδομένων των υπερπαραμέτρων, θα μπορούσε να χρησιμοποιηθεί ένα μήκος εισόδου έως 15, διατηρώντας παράλληλα την πλήρη κάλυψη του δεκτικού πεδίου. Σε γενικές γραμμές, κάθε επιπρόσθετο επίπεδο/στρώμα προσθέτει μια τιμή ίση με $d * (k - 1)$ στο τρέχον πλάτος του δεκτικού πεδίου, όπου η διαστολή d υπολογίζεται ως $d = b^i$, με το i να αντιπροσωπεύει τον αριθμό των επιπέδων κάτω από το νέο μας επίπεδο. Κατά συνέπεια, το πλάτος του δεκτικού πεδίου w ενός TCN με εκθετική διαστολή με βάση b , μέγεθος πυρήνα k και αριθμό στρωμάτων n δίνεται από την σχέση:

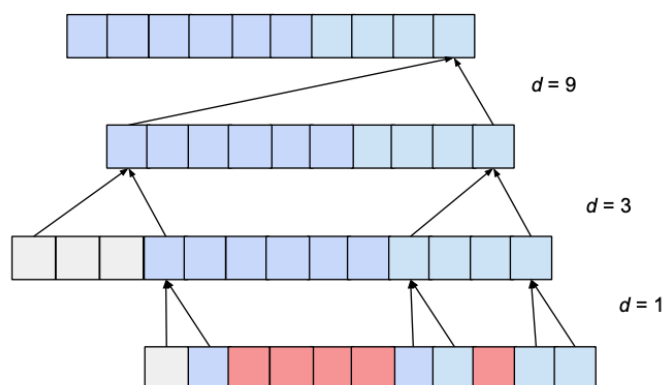
$$w = 1 + \sum_{i=0}^{n-1} (k - 1) * b^i = 1 + (k - 1) \frac{b^n - 1}{b - 1} \quad (3.29)$$

Παρόλα αυτά ιδιαίτερη προσοχή απαιτείται στην επιλογή των υπερπαραμέτρων της βάσης διαστολής (b) και του μεγέθους του πυρήνα (k) καθώς από αυτές εξαρτάται αν το δεκτικό πεδίο θα έχει "τρύπες". Παρόλο που στο παράδειγμα μας το δεκτικό πεδίο μπορεί να καλύψει εύρος μεγαλύτερο από το μήκος εισόδου αν το μέγεθος πυρήνα δεν είναι μεγαλύτερο ή ίσο από την βάση διαστολής τότε υπάρχει ο κίνδυνος ένα σημείο εξόδου να μην εξαρτάται από όλα τα προηγούμενα στοιχεία εισόδου. Ένα τέτοιο παράδειγμα εμφανίζεται στο σχήμα 3.29.

Λαμβάνοντας υπόψη αυτές τις παρατηρήσεις, μπορούμε να υπολογίσουμε πόσα επίπεδα χρειάζεται το δίκτυό μας για πλήρη κάλυψη ιστορικού. Δεδομένου ενός μεγέθους πυρήνα (k), βάσης διαστολής (b), όπου $k \geq b$ και μήκους εισόδου (l), η ακόλουθη ανισότητα πρέπει να ισχύει για πλήρη κάλυψη ιστορικού:

$$1 + (k - 1) \frac{b^n - 1}{b - 1} \geq l \quad (3.30)$$

Λύνοντας πλέον ως προς τον αριθμό των επιπέδων που απαιτούνται για την πλήρη κάλυ-



Σχήμα 3.26: Δίκτυο με διασταθμένα συνελκτικά στρώματα με 'τρύπες' [14]

ψη και επιλέγοντας τον μικρότερο (ανισότητα 3.30) παρατηρούμε ότι πλέον ο αριθμός των επιπέδων που απαιτούνται δεν είναι γραμμικός συνδυασμός του μήκους εισόδου, αλλά λογαριθμικός. Αυτό αποτελεί σημαντική βελτίωση, η οποία μπορεί να επιτευχθεί χωρίς να θυσιάζεται η δεκτική κάλυψη πεδίου.

$$\log_b\left(\frac{(l-1)(b-1)}{(k-1)} + 1\right) \quad (3.31)$$

Τώρα το μόνο που απομένει να καθοριστεί είναι ο αριθμός των μηδενικών στοιχείων γεμίματος που απαιτούνται σε κάθε επίπεδο. Λαμβάνοντας υπόψη μια βάση διαστολής b , ένα μέγεθος πυρήνα k και έναν αριθμό επιπέδων i κάτω από το τρέχον επίπεδο μας, τότε ο αριθμός των καταχωρήσεων μηδενικής γέμισης p που απαιτούνται για το τρέχον επίπεδο υπολογίζονται ως εξής:

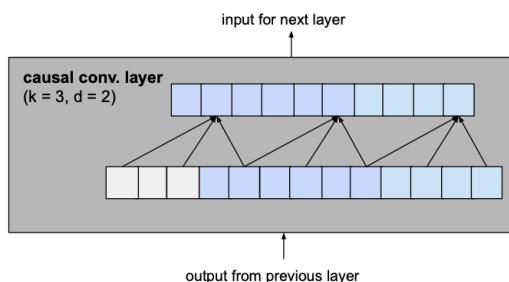
$$p = b^i * (k - 1) \quad (3.32)$$

3.4.3 Υπολειμματικά μπλοκ (Residual Blocks)

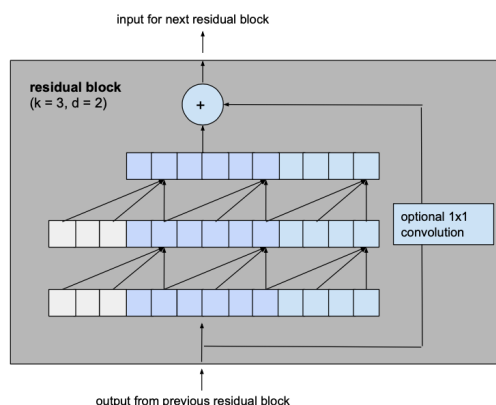
Παραπάνω περιγράφηκαν οι ιδιότητες που αρκεί να έχει η βασική αρχιτεκτονική των Χρονικών Συνελκτικών διτύων (TCN). Οι Bai et al. [62] προτείνουν μερικές προσθήκες σε αυτή την βασική αρχιτεκτονική για βελτιωμένη απόδοση, οι οποίες θα συζητηθούν σε αυτήν την υποενότητα. Οι προσθήκες αυτές είναι οι υπολειμματικές συνδέσεις και οι λειτουργίες κανονικοποίησης και ενεργοποίησης.

Η μεγαλύτερη τροποποίηση που κάνουμε στο προηγούμενο βασικό μοντέλο είναι να αλλάξουμε το θεμελιώδες δομικό στοιχείο του μοντέλου από ένα απλό μονοδιάστατο αιτιακό συνελκτικό επίπεδο σε ένα υπολειμματικό μπλοκ που αποτελείται από 2 επίπεδα με τον ίδιο συντελεστή διαστολής και μια υπολειμματική σύνδεση 3.28.

Η έξοδος των δύο συνελκτικών επιπέδων θα προστεθεί στην είσοδο του υπολειμματικού μπλοκ για να παραχθεί η είσοδος για το επόμενο μπλοκ. Για όλα τα εσωτερικά μπλοκ του δικτύου, δηλαδή για όλα εκτός από το πρώτο και το τελευταίο, τα πλάτη των καναλιών εισόδου και εξόδου είναι τα ίδια, γνωστά και ως αριθμός φίλτρων. Δεδομένου ότι το πρώτο συνελκτικό επίπεδο του πρώτου υπολειμματικού μπλοκ και το δεύτερο συνελκτικό επίπεδο του



Σχήμα 3.27: Ένα αιτιατό μονοδιάστατο συνελκτικό επίπεδο πριν την αλβλαγή [14]



Σχήμα 3.28: Ένα υπολειμματικό μπλοκ μετά την αλβλαγή [14]

τελευταίου υπολειμματικού μπλοκ μπορεί να έχουν διαφορετικά πλάτη καναλιών εισόδου και εξόδου, το πλάτος του υπολειμματικού τανυστή μπορεί να χρειαστεί να ρυθμιστεί, κάτι που γίνεται με τη χρήση ενός συνελκτικού επιπέδου διαστάσεων 1×1 .

Αυτή η αλλαγή επηρεάζει τον υπολογισμό του ελάχιστου αριθμού απαιτούμενων επιπέδων για πλήρη κάλυψη. Τώρα πρέπει να σκεφτούμε πόσα υπολειμματικά μπλοκ είναι απαραίτητα για να επιτευχθεί πλήρης κάλυψη του πεδίου εισόδου. Η προσθήκη ενός υπολειμματικού μπλοκ σε ένα TCN προσθέτει διπλάσιο πλάτος δεκτικού πεδίου από ότι όταν προσθέτουμε ένα βασικό αιτιολογικό επίπεδο, αφού περιλαμβάνει 2 τέτοια επίπεδα. Έτσι, το συνολικό μέγεθος του δεκτικού πεδίου r ενός TCN με βάση διαστολής b , μέγεθος πυρήνα k με $k \geq b$ και αριθμό υπολειπόμενων μπλοκ n μπορεί να υπολογιστεί ως

$$r = 1 + \sum_{i=0}^{n-1} 2(k-1) * b^i = 1 + 2(k-1) \frac{b^n - 1}{b-1} \quad (3.33)$$

όπου οδηγεί αντίστοιχα σε ελάχιστο αριθμό υπολειμματικών μπλοκ n που απαιτούνται για πλήρη κάλυψη ίσο με:

$$n = \log_b \left(\frac{(l-1)(b-1)}{2(k-1)} + 1 \right) \quad (3.34)$$

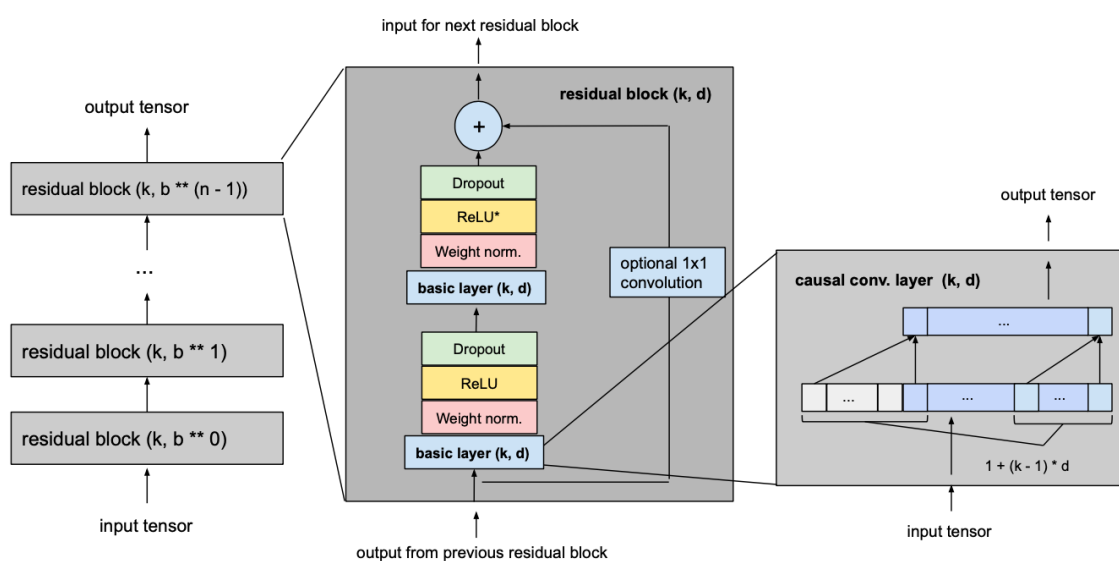
Για να καταστήσουμε το TCN μας κάτι παραπάνω από ένα υπερβολικά περίπλοκο μοντέλο γραμμικής παλινδρόμησης, οι συναρτήσεις ενεργοποίησης πρέπει να προστεθούν πάνω από τα συνελκτικά επίπεδα για να εισαχθούν μη γραμμικότητες. Οι ενεργοποιήσεις με την συνάρτηση ReLU προστίθενται στα υπολειμματικά μπλοκ μετά από και τα δύο συνελκτικά

επίπεδα.

Για την ομαλοποίηση της εισόδου των κρυφών επιπέδων (που αντισταθμίζει το πρόβλημα της εκτόξευσης κλίσης μεταξύ άλλων), η κανονικοποίηση βάρους (weight normalization) εφαρμόζεται σε κάθε συνελκτικό επίπεδο.

Προκειμένου να αποφευχθεί η υπερεκπαίδευση, η εγκατάλειψη εισάγεται μέσω ενός επιπέδου εγκατάλειψης (Dropout Layer) μετά από κάθε συνελκτικό επίπεδο σε κάθε υπολειμματικό μπλοκ.

Η παρακάτω εικόνα δείχνει το τελικό μοντέλο TCN με l ίσο με το μήκος εισόδου, k ίσο με μέγεθος πυρήνα, b ίσο με βάση διαστολής, $k \geq b$ και με ελάχιστο αριθμό υπολειπόμενων μπλοκ για πλήρη κάλυψη ιστορικού n , όπου το n μπορεί να υπολογιστεί από τις άλλες τιμές όπως εξηγείται παραπάνω.



Σχήμα 3.29: Τελικό TCN μοντέλο [14]

Μέρος 

Πρακτικό Μέρος

Κεφάλαιο 4

Περιγραφή και Ανάλυση δεδομένων

Στο παρόν κεφάλαιο, θα ασχοληθούμε εκτενώς με την ανάλυση των δεδομένων που θα χρησιμοποιηθούν για την δημιουργία των μοντέλων μηχανικής μάθησης. Πιο συγκεκριμένα, θα αναφερθούμε στα χαρακτηριστικά των δεδομένων, τον τρόπο συλλογής τους, τις ιδιομορφίες τους αλλά και τον τρόπο επεξεργασίας τους, για την μετατροπή τους σε μορφή κατάλληλη για την εφαρμογή μεθόδων μηχανικής μάθησης. Τα δεδομένα που ελέγχθηκαν είναι δεδομένα από αισθητήρες smartphone, που όπως αναφέραμε στην εισαγωγή είναι καίριας σημασίας για τον προγραμματισμό πολλών εφαρμογών.

Για την επεξεργασία των δεδομένων, χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python και για την οπτικοποίηση τους χρησιμοποιήθηκε η βιβλιοθήκη Matplotlib [64], η οποία είναι βασισμένη σε πίνακες NumPy [65] και σχεδιασμένη να λειτουργεί με την ευρύτερη στοιβά SciPy [66].

4.1 Περιγραφή Συνόλου Δεδομένων

Αδιαμφισβήτητα, ο σημαντικότερος παράγοντας κατά την εκπαίδευση μοντέλων επιβλεπόμενης μάθησης (supervised learning) είναι η συγκέντρωση των κατάλληλων δεδομένων σε συνδυασμό με την σωστή προεπεξεργασία τους. Το πρόβλημα της αναγνώρισης ανθρώπινων δραστηριοτήτων είναι μια ανθρωποκεντρική εφαρμογή. Τα smartphones δημιουργούν νέες ερευνητικές ευκαιρίες για τέτοιου είδους εφαρμογές καθώς ο χρήστης είναι μια πλούσια πηγή πληροφοριών και το τηλέφωνο είναι ένα εξαιρετικό εργαλείο ανίχνευσης.

Για τις ανάγκες της διπλωματικής επιλέχθηκε ένα τυπικό σύνολο δεδομένων αναγνώρισης ανθρώπινης δραστηριότητας, όπως είναι το «Activity Recognition Using Smart Phones Dataset» που διατίθεται από το 2012. Στις επόμενες ενότητες θα παρουσιαστεί η διαδικασία/μεθοδολογία εξαγωγής δεδομένων του συγκεκριμένου dataset, η επεξεργασία των σημάτων και η αντιστοίχιση των χαρακτηριστικών.

4.1.1 Μεθοδολογία

Ένα σύνολο πειραμάτων πραγματοποιήθηκε, ούτως ώστε να αποκτηθεί το σύνολο δεδομένων που χρησιμοποιήθηκε (HAR Dataset). Μια ομάδα εθελοντών, των οποίων οι ηλικίες κυμαίνονταν μεταξύ των 19 και 48 ετών επιλέχθηκαν για αυτή την εργασία. Οι οδηγίες που κλήθηκε ο κάθε εθελοντής να ακολουθήσει ήταν να πραγματοποιήσει ένα πρωτόκολλο

από δραστηριότητες ενώ φόραγε στην μέση του ένα κινητό Samsung Galaxy S II. Οι έξι επιλεγμένες δραστηριότητες καθημερινής ζωής (Activities of Daily Living-ADL) ήταν "Στέκομαι", "Κάθομαι", "Ξαπλώνω", "Περπατάω", "Περπατάω προς τα κάτω" και "Περπατάω προς τα πάνω".

Κάθε εθελοντής/υποκείμενο εκτέλεσε το πρωτόκολλο δραστηριοτήτων δύο φορές, την πρώτη με το κινητό τοποθετημένο στην αριστερή πλευρά της ζώνης και την δεύτερη φορά το τοποθετούσε σε όποια θέση προτιμούσε ή τον βόλευε περισσότερο. Επιπλέον, υπήρχε ένας διαχωρισμός των 5 δευτερολέπτων μεταξύ των δραστηριοτήτων, κατά τα οποία τα υποκείμενα ξεκουράζονταν. Οι δραστηριότητες αυτές πραγματοποιήθηκαν σε συνθήκες εργαστηρίου, παρόλα αυτά ζητήθηκε από τους εθελοντές να εκτελέσουν τις δραστηριότητες με όποια σειρά θέλουν, προκειμένου να παραχθεί ένα "φυσικό" σύνολο δεδομένων. Στον πίνακα του σχήματος 4.1 παρουσιάζονται οι λεπτομέρειες του πρωτοκόλλου δραστηριοτήτων.

Δεδομένου ότι οι εθελοντές ήταν διαφόρων ηλικιών και της τυχαιότητας στην σειρά που εκτελέσαν τις δραστηριότητες μπορούμε θεωρήσουμε ότι το σύνολο δεδομένων μας είναι ικανοποιητικά αντιπροσωπευτικό.

No.	Static	Time (sec)	No.	Dynamic	Time (sec)
0	Start (Standing Pos)	0	7	Walk (1)	15
1	Stand (1)	15	8	Walk (2)	15
2	Sit (1)	15	9	Walk Downstairs (1)	12
3	Stand (2)	15	10	Walk Upstairs (2)	12
4	Lay Down (1)	15	11	Walk Downstairs (1)	12
5	Sit (2)	15	12	Walk Upstairs (2)	12
6	Lay Down (2)	15	13	Walk Downstairs (3)	12
			14	Walk Upstairs (3)	12
			15	Stop	0
				Total	192

Σχήμα 4.1: Λεπτομέρειες του πρωτοκόλλου δραστηριοτήτων [17]

4.1.2 Επεξεργασία Σημάτων

Από την διαδικασία που περιγράφηκε παραπάνω συλλέχθηκαν τριαξονικά σήματα για την γραμμική επιτάχυνση και την γωνιακή ταχύτητα των υποκειμένων από το επιταχυνόμετρο και το γυροσκόπιο του κινητού, με ρυθμό δειγματοληψίας 50Hz. Αυτά τα σήματα έχουν προεπεξεργαστεί ώστε να μειωθεί ο θόρυβος με ένα μέσο φίλτρο και ένα 3ου βαθμού low-pass Butterworth φίλτρο με συχνότητα αποκοπής 20Hz. Ο ρυθμός δειγματοληψίας επαρκούσε για να πιάσει την κίνηση του ανθρώπινου σώματος, καθώς το 99% της ενέργειας του περιέχεται κάτω από τα 15Hz [67].

Το σήμα επιτάχυνσης, το οποίο έχει συστατικά της βαρύτητας και της κίνησης του σώματος, διαχωρίστηκε χρησιμοποιώντας ένα άλλο φίλτρο χαμηλής διέλευσης Butterworth στην επιτάχυνση του σώματος και τη βαρύτητα. Η βαρυτική δύναμη θεωρείται ότι έχει μόνο συστατικά χαμηλής συχνότητας, επομένως διαπιστώθηκε από τα πειράματα ότι τα 0.3 Hz ήταν η βέλτιστη γωνιακή συχνότητα για ένα σταθερό σήμα βαρύτητας.

Επιπρόσθετα χρονικά σήματα ελήφθησαν με υπολογισμό από τα τριαξονικά σήματα του ευκλείδειου μεγέθους και των παραγώγων χρόνου (τράνταγμα da/dt και γωνιακή επιτάχυνση

dw/dt). Στη συνέχεια, τα σήματα χρόνου συλλέχθηκαν σε συρόμενα παράθυρα σταθερού πλάτους ίσο με 2,56 sec και επικαλύπτονται μεταξύ τους κατά 50%, επομένως:

- Η ταχύτητα ενός μέσου ατόμου που περπατά κυμαίνεται σε [90, 130] βήματα/λεπτό [68], δηλαδή τουλάχιστον 1,5 βήματα/δευτερόλεπτο.
- Τουλάχιστον ένας πλήρης κύκλος βάδισης (δύο βήματα) παρουσιάζεται σε κάθε δείγμα παραθύρου.
- Άτομα με βραδύτερο ρυθμό όπως οι ηλικιωμένοι και τα άτομα με ειδικές ανάγκες θα πρέπει επίσης να επωφεληθούν από αυτή τη μέθοδο, καθώς έχει προταθεί μια ελάχιστη ταχύτητα ίση με το 50% του μέσου ανθρώπινου ρυθμού.
- Τα σήματα αντιστοιχίζονται επίσης στον τομέα των συχνοτήτων μέσω ενός Fast Fourier Transform (FFT), βελτιστοποιημένου για ισχύ δύο διανυσμάτων (2,56 δευτερόλεπτα \times 50Hz = 128 κύκλοι)

Έτσι, με τη μέθοδο αυτή λήφθηκαν συνολικά 17 σήματα, τα οποία παρατίθενται στον πίνακα του σχήματος 4.2.

Name	Time	Freq.
Body Acc	1	1
Gravity Acc	1	0
Body Acc Jerk	1	1
Body Angular Speed	1	1
Body Angular Acc	1	0
Body Acc Magnitude	1	1
Gravity Acc Mag	1	0
Body Acc Jerk Mag	1	1
Body Angular Speed Mag	1	1
Body Angular Acc Mag	1	1

Σχήμα 4.2: Σήματα πεδίου χρόνου και συχνότητας που λαμβάνονται από τους αισθητήρες smartphone [17]

4.1.3 Αντιστοίχιση χαρακτηριστικών

Από κάθε παράθυρο δειγματοληψίας που περιγράφηκε παραπάνω, ελήφθη ένα διάνυσμα χαρακτηριστικών. Για τη αντιστοίχιση των χαρακτηριστικών χρησιμοποιήθηκαν τυπικά μέτρα στη βιβλιογραφία HAR [69], όπως ο μέσος όρος, ο συσχετισμός, η περιοχή μεγέθους σήματος (SMA) και οι συντελεστές αυτοανάπτυξης [70]. Ένα νέο σύνολο χαρακτηριστικών χρησιμοποιήθηκε επίσης για τη βελτίωση της μαθησιακής απόδοσης, συμπεριλαμβανομένης της ενέργειας των διαφορετικών ζωνών συχνοτήτων, της κλίσης της συχνότητας και της γωνίας μεταξύ των διανυσμάτων (π.χ. μέση επιτάχυνση σώματος και διάνυσμα y). Ο πίνακας στο σχήμα 4.3 περιέχει τη λίστα με όλα τα μέτρα που εφαρμόζονται στον τομέα χρόνου και συχνότητας σήματα.

Συνολικά εξήχθησαν 561 χαρακτηριστικά για να περιγράψουν κάθε παράθυρο δραστηριότητας. Προκειμένου να διευκολυνθεί η αξιολόγηση της απόδοσης, το σύνολο δεδομένων

έχει επίσης χωριστεί τυχαία σε δύο ανεξάρτητα σύνολα, όπου το 70% των δεδομένων επιλέχθηκαν για εκπαίδευση και το υπόλοιπο 30% για δοκιμές. Το σύνολο δεδομένων αναγνώρισης ανθρώπινης δραστηριότητας έχει διατεθεί για δημόσια χρήση και παρουσιάζεται ως ακατέργαστα σήματα αδρανειακών αισθητήρων και επίσης ως διανύσματα χαρακτηριστικών για κάθε μοτίβο. Έχει υποβληθεί ως Αναγνώριση Ανθρώπινης Δραστηριότητας με τη χρήση συνόλου δεδομένων Smartphones στο αποθετήριο UCI Machine Learning Repository [71].

Function	Description
mean	Mean value
std	Standard deviation
mad	Median absolute value
max	Largest values in array
min	Smallest value in array
sma	Signal magnitude area
energy	Average sum of the squares
iqr	Interquartile range
entropy	Signal Entropy
arCoeff	Autoregression coefficients
correlation	Correlation coefficient
maxFreqInd	Largest frequency component
meanFreq	Frequency signal weighted average
skewness	Frequency signal Skewness
kurtosis	Frequency signal Kurtosis
energyBand	Energy of a frequency interval
angle	Angle between two vectors

Σχήμα 4.3: Λίστα μετρικών για τον υπολογισμό διανυσμάτων χαρακτηριστικών. [17]

4.2 Ανάλυση και Οπτικοποίηση Δεδομένων

Στο παρόν κεφάλαιο θα γίνει ανάλυση και μερική οπτικοποίηση του συνόλου δεδομένων που περιγράφηκε παραπάνω. Στόχος είναι η βαθύτερη κατανόηση των δεδομένων και των αναγκών τους προκειμένου να προταθούν κατάλληλα μοντέλα πρόβλεψης.

Τα ακατέργαστα δεδομένα δεν είναι διαθέσιμα. Αντί αυτών, είναι διαθέσιμη μια προεπεξεργασμένη μορφή τους όπως ακριβώς περιγράφηκε στην υποενότητα "Επεξεργασία σημάτων". Θα εστιάσουμε την προσοχή μας σε αυτά τα σήματα καθώς αυτό είναι πιο ενδιαφέρον για την ανάπτυξη μοντέλων μηχανικής εκμάθησης που μπορούν να μάθουν μια κατάλληλη αναπαράσταση, αντί να χρησιμοποιούν τη συγκεκριμένη περιοχή εστιασμένων χαρακτηριστικών όπως αυτά που προέκυψαν από την αντιστοίχιση χαρακτηριστικών. Μια επιθεώρηση ενός αρχείου δεδομένων φανερώνει ότι οι στήλες διαχωρίζονται από το κενό διάστημα και οι τιμές φαίνεται να κλιμακώνονται στο εύρος [-1, 1].

4.2.1 Ισορροπία Δραστηριοτήτων

Ένας καλός πρώτος έλεγχος του συνόλου δεδομένων είναι η ισορροπία των κλάσεων κάθε δραστηριότητας. Έχοντας ως δεδομένο ότι 30 υποκείμενα εκτέλεσαν και τις 6 δραστηριότητες δύο φορές αναμένουμε ότι το σύνολο δεδομένων μας θα είναι ισορροπημένο. Αυτό βοηθάει στην μοντελοποίηση του. Στο σχήμα 4.4 φαίνονται τα ποσοστά κάθε κατηγορίας δραστηριοτήτων. Δεδομένου ότι το σύνολο εκπαίδευσης και το σύνολο επαλήθευσης ήταν ήδη προεπιλεγμένα με αντίστοιχα ποσοστά 70-30% επί του συνόλου των δεδομένων

κρίθηκε σκόπιμο να μελετηθεί η ισορροπία των κλάσεων και στα δύο αυτά σύνολα, καθώς μεγάλη απόκλιση στην διανομή των δειγμάτων από το ένα σύνολο στο άλλο θα μπορούσε να δημιουργήσει πρόβλημα στην εκπαίδευση των μοντέλων μας.

Σημειώνεται ότι η αρίθμηση των δραστηριοτήτων είναι με την σειρά: Περπάτημα(1), Περπάτημα προς τα πάνω(2), Περπάτημα προς τα κάτω(3), Κάθομαι(4), Στέκομαι(5), Ξαπλώνω(6).

```

Train Dataset
Class=1, total=1226, percentage=16.676
Class=2, total=1073, percentage=14.595
Class=3, total=986, percentage=13.411
Class=4, total=1286, percentage=17.492
Class=5, total=1374, percentage=18.689
Class=6, total=1407, percentage=19.138
Test Dataset
Class=1, total=496, percentage=16.831
Class=2, total=471, percentage=15.982
Class=3, total=420, percentage=14.252
Class=4, total=491, percentage=16.661
Class=5, total=532, percentage=18.052
Class=6, total=537, percentage=18.222
Both
Class=1, total=1722, percentage=16.720
Class=2, total=1544, percentage=14.992
Class=3, total=1406, percentage=13.652
Class=4, total=1777, percentage=17.254
Class=5, total=1906, percentage=18.507
Class=6, total=1944, percentage=18.876

```

Σχήμα 4.4: Η ισορροπία των κλάσεων των δραστηριοτήτων του dataset μας

Όπως φαίνεται από το σχήμα τόσο στο σύνολο εκπαίδευσης όσο και επαλήθευσης τα ποσοστά κάθε κλάσης είναι αρκετά κοντινά. Η μέγιστη διαφορά εντοπίζεται στο σύνολο εκπαίδευσης μεταξύ της δραστηριότητας Περπάτημα προς τα κάτω(3) και Ξαπλώνω(6), που είναι της τάξεως του 5,7%. Αυτό όμως δικαιολογείται καθώς η δραστηριότητα 3 κατέχει 3-6% λιγότερα δείγματα και επί του συνόλου των δεδομένων. Επιπλέον, η δραστηριότητα Περπάτημα προς τα πάνω(2) φαίνεται να εμφανίζει λιγότερα δείγματα από τις άλλες, ενώ οι δραστηριότητες Στέκομαι(5) και Ξαπλώνω(6) φαίνεται να έχουν πιο πολλά δείγματα από όλες τις άλλες (περίπου 2%).

Σε κάθε περίπτωση το σύνολο δεδομένων μας κρίνεται ισορροπημένο και δεν απαιτείται καμία περαιτέρω ενέργεια εξισορρόπησης του με σκοπό την καλύτερη μοντελοποίησή του.

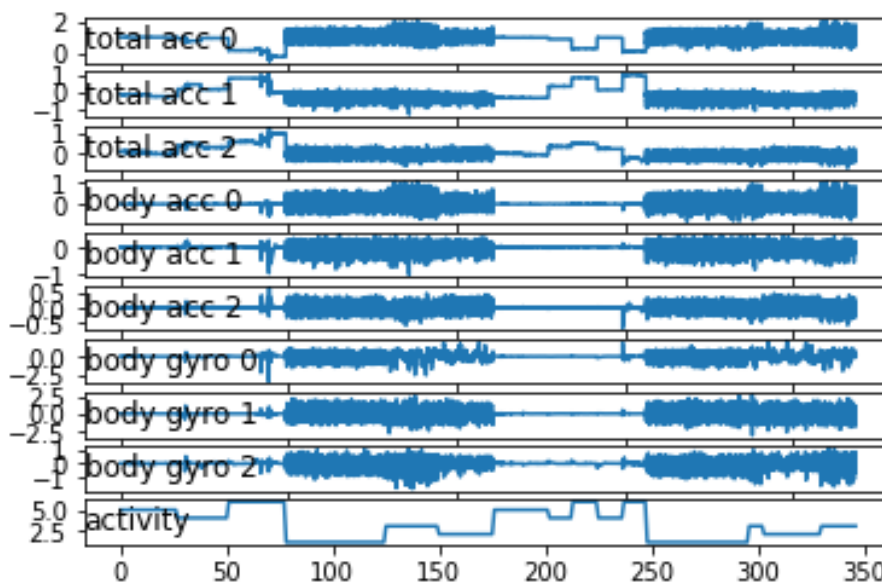
4.2.2 Χρονική σειρά δεδομένων ανά Υποκείμενο

Δουλεύοντας με χρονικά ακολουθιακά δεδομένα, ένας ακόμη έλεγχος που μπορούμε να κάνουμε είναι να δημιουργήσουμε μια γραφική με τα ακατέργαστα δεδομένα. Θυμίζουμε ότι, τα ακατέργαστα δεδομένα αποτελούνται από χρονικά παράθυρα για κάθε χαρακτηριστικό, με αλληλεπικάλυψη 50% μεταξύ τους. Τα χαρακτηριστικά στην περίπτωση μας είναι τα τριαξονικά δεδομένα για την συνολική επιτάχυνση, την επιτάχυνση σώματος και την γωνιακή ταχύτητα, δηλαδή έχουμε συνολικά 9 χαρακτηριστικά.

Για να μπορέσουμε να δημιουργήσουμε τα ακατέργαστα δεδομένα, θα πρέπει πρώτα να αντιστοιχίσουμε τα δεδομένα μας στους εθελοντές/υποκείμενα. Κατόπιν, εύκολα ανακτούμε τα δεδομένα ενός συγκεκριμένου υποκειμένου, στηριζόμενοι στην προηγούμενη αντιστοίχιση. Επειδή, τα δεδομένα αποτελούνται από παράθυρα με επικάλυψη, μπορούμε να γράψουμε μια συνάρτηση για να αφαιρέσουμε αυτήν την επικάλυψη και να ενώσουμε τα παράθυρα που αφορούν έναν συγκεκριμένο εθελοντή, δημιουργώντας μία μεγάλη ακολουθία που μπορεί να σχεδιαστεί απευθείας ως γραφική παράσταση γραμμής.

Μπορούμε να σχεδιάσουμε καθεμία από τις εννέα μεταβλητές για ένα υποκείμενο με τη σειρά και μια τελική γραφική παράσταση για το επίπεδο δραστηριότητας. Κάθε σειρά θα έχει τον ίδιο αριθμό χρονικών βημάτων (μήκος άξονα x), επομένως, μπορεί να είναι χρήσιμο να δημιουργηθεί ένα υποσχέδιο για κάθε μεταβλητή και να ευθυγραμμιστούν όλα τα γραφήματα κάθετα, ώστε να συγκρίνουμε την κίνηση σε κάθε μεταβλητή.

Παρακάτω στο σχήμα 4.5 φαίνεται ένα παράδειγμα γραφικής παράστασης για το υποκείμενο 0.



Σχήμα 4.5: Γραφική παράσταση χρονικής σειράς δεδομένων ενός υποκειμένου

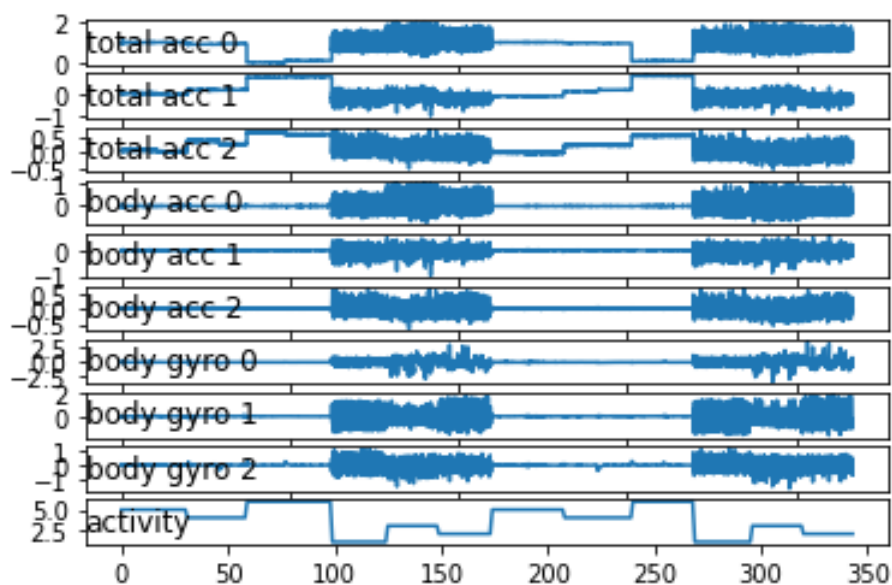
Παρατηρούμε ότι για τις δραστηριότητες που περιέχουν κίνηση (1-3) υπάρχουν έντονες διακυμάνσεις στις τιμές που καταγράφηκαν, ενώ αντίθετα για τις υπόλοιπες δραστηριότητες (4-6) οι γραφικές παρουσιάζουν κάποια στατικότητα. Ακόμη, παρατηρούμε ότι το κάθε υποκείμενο εκτελεί δύο φορές κάθε δραστηριότητα, ενώ κάποιες ακόμα παραπάνω φορές. Επομένως, για έναν εθελοντή δεν μπορούμε να κάνουμε υποθέσεις για το πόσες και ποιες δραστηριότητες έκανε, καθώς και με τι σειρά τις έκανε.

Μπορούμε επίσης να δούμε κάποια σχετικά μεγάλη κίνηση για κάποιες στάσιμες δραστηριότητες, όπως η δραστηριότητα 6. Είναι πιθανό αυτά να είναι ακραία στοιχεία ή να σχετίζονται με μεταβάσεις δραστηριοτήτων. Μπορεί να είναι δυνατόν να εξομαλυνθούν ή να αφαιρεθούν αυτές οι παρατηρήσεις ως ακραίες τιμές.

Παρατηρούμε ακόμη, μεγάλη ομοιότητα στις εννέα μεταβλητές. Αυτό μας οδηγεί στο συμπέρασμα ότι είναι πολύ πιθανό να απαιτείται ένα υποσύνολο αυτών των μεταβλητών για

την ανάπτυξη ενός μοντέλου πρόβλεψης.

Παρόμοια εικόνα έχουμε και για άλλους εθελοντές. Για παράδειγμα για τον εθελοντή 19 παρατηρούμε 4.6 παρατηρούμε ότι τα δεδομένα κατά την εκτέλεση της δραστηριότητας 6 είναι πιο εξομαλυμένα καθώς επίσης παρατηρούμε και μια κανονικότητα στην σειρά που πραγματοποιήθηκαν οι δραστηριότητες από τον συγκεκριμένο εθελοντή.



Σχήμα 4.6: Γραφική παράσταση χρονικής σειράς δεδομένων ενός δεύτερου υποκειμένου

4.2.3 Ιστογράμματα ανά Υποκείμενο

Ένα ιστογράμμα είναι ένα εξαιρετικό εργαλείο για την γρήγορη εκτίμηση μιας κατανομής πιθανότητας που γίνεται εύκολα κατανοητή από σχεδόν οποιοδήποτε κοινό. Η Python προσφέρει πληθώρα διαφορετικών επιλογών για την κατασκευή και τη σχεδίαση ιστογραμμάτων. Οι περισσότεροι άνθρωποι γνωρίζουν ένα ιστογράμμα από τη γραφική του αναπαράσταση, το οποίο είναι παρόμοιο με ένα γράφημα ράβδων [72].

Γενικά χρησιμοποιούμε τα δεδομένα των κινήσεων ενός υποκειμένου για να προβλέψουμε τις πράξεις άλλων. Αυτό σημαίνει ότι πρέπει να υπάρχει κανονικότητα μεταξύ των κινησιακών δεδομένων ενός υποκειμένου με τα υπόλοιπα. Γνωρίζουμε ήδη ότι τα δεδομένα έχουν κλιμακωθεί μεταξύ των τιμών στο $[-1, 1]$, πιθανών ανά υποκείμενο, επομένως αναμένουμε τα πλάτη για τις συγκεκριμένες κινήσεις/δραστηριότητες να είναι παρόμοια.

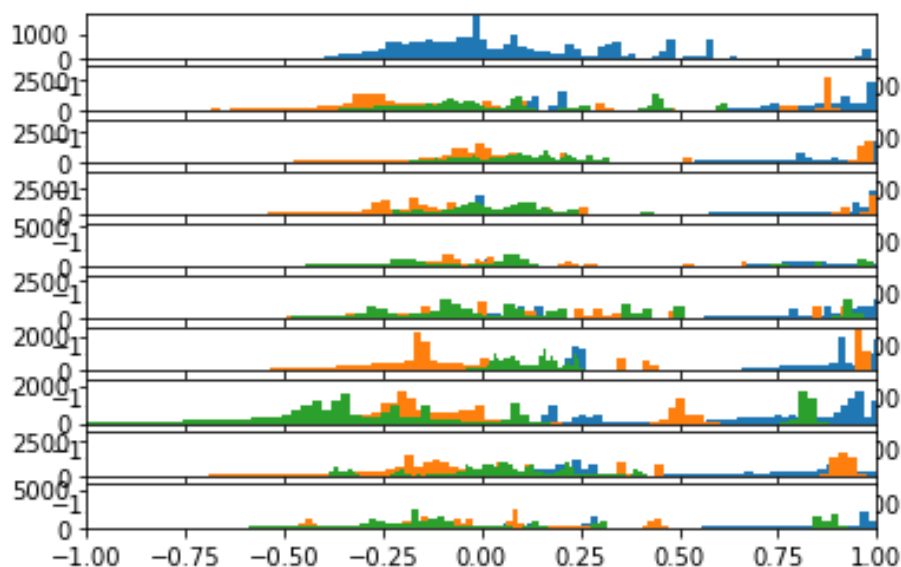
Επιπλέον, αναμένουμε ότι και η κατανομή των κινησιακών δεδομένων θα είναι παρόμοια καθώς τα υποκείμενα εκτελούσαν τις ίδιες ενέργειες/δραστηριότητες.

Μπορούμε να το ελέγξουμε αυτό σχεδιάζοντας και συγκρίνοντας τα ιστογράμματα των δεδομένων κίνησης μεταξύ των υποκειμένων. Μια χρήσιμη προσέγγιση θα ήταν να δημιουργήσουμε ένα σχέδιο ανά υποκείμενο και να σχεδιάσουμε και τους τρεις άξονες (x, y, z, με μπλε, πορτοκαλί και κίτρινο χρώμα αντίστοιχα) ενός αισθητήρα δεδομένων (π.χ. συνολική επιτάχυνση) και στη συνέχεια να το επαναλάβουμε για πολλά υποκείμενα. Τα γραφήματα μπορούν να τροποποιηθούν ώστε να χρησιμοποιούν τον ίδιο άξονα και να ευθυγραμμίζονται οριζόντια, έτσι ώστε να μπορούν να συγκριθούν οι κατανομές για κάθε μεταβλητή στα

υποκείμενα.

Συνολική επιτάχυνση

Στο σχήμα 4.7 φαίνεται το ιστόγραμμα και για τους 3 άξονες, ανά υποκείμενο για τα κινησιακά δεδομένα της συνολικής επιτάχυνσης από το επιταχυνσόμετρο.



Σχήμα 4.7: Ιστόγραμμα συνολικής επιτάχυνσης για 10 υποκείμενα

Μπορούμε να δούμε ότι η κατανομή για έναν δεδομένο άξονα εμφανίζεται να μοιάζει με Gaussian με μεγάλες ξεχωριστές ομάδες δεδομένων.

Επιπλέον, μπορούμε να δούμε μερικές από τις κατανομές να ευθυγραμμίζονται (π.χ. κύριες ομάδες στη μέση περίπου 0,0), υποδηλώνοντας ότι μπορεί να υπάρχει κάποια συνέχεια των δεδομένων κίνησης μεταξύ των υποκειμένων, τουλάχιστον για αυτά τα δεδομένα. Κάτι τέτοιο θα βοηθήσουμε πολύ στην μοντελοποίηση των δεδομένων μας.

Επιτάχυνση σώματος

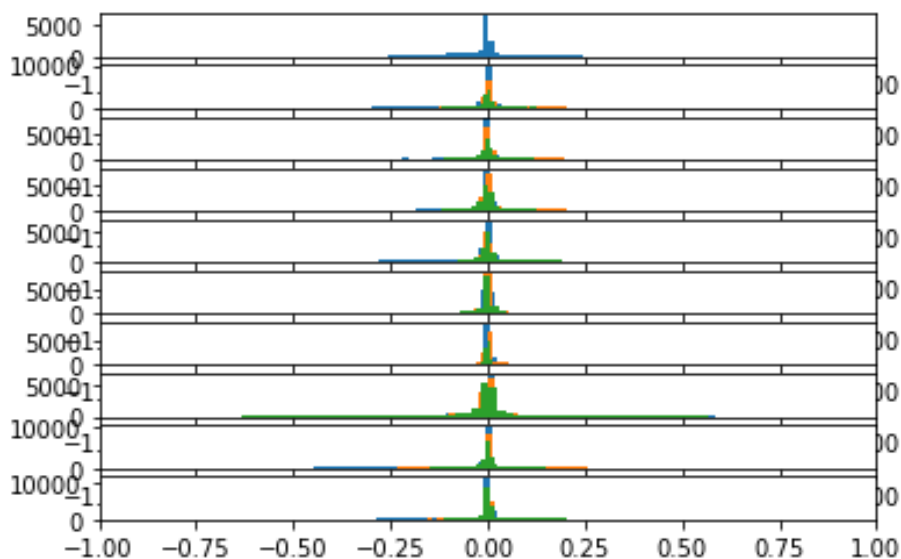
Στο σχήμα 4.8 φαίνεται το ιστόγραμμα και για τους 3 άξονες, ανά υποκείμενο για τα κινησιακά δεδομένα της επιτάχυνσης σώματος.

Εδώ μπορούμε να δούμε όλα τα δεδομένα συγκεντρωμένα γύρω στο 0,0 σε άξονα εντός ενός υποκειμένου και μεταξύ των υποκειμένων. Αυτό υποδηλώνει ότι ίσως τα δεδομένα ήταν κεντραρισμένα (μηδενικό μέσο όρο). Αυτή η ισχυρή συνέπεια μεταξύ των υποκειμένων μπορεί να βοηθήσει στη μοντελοποίηση και μπορεί να υποδηλώνει ότι οι διαφορές μεταξύ των υποκειμένων στα συνολικά δεδομένα επιτάχυνσης μπορεί να μην είναι τόσο χρήσιμες.

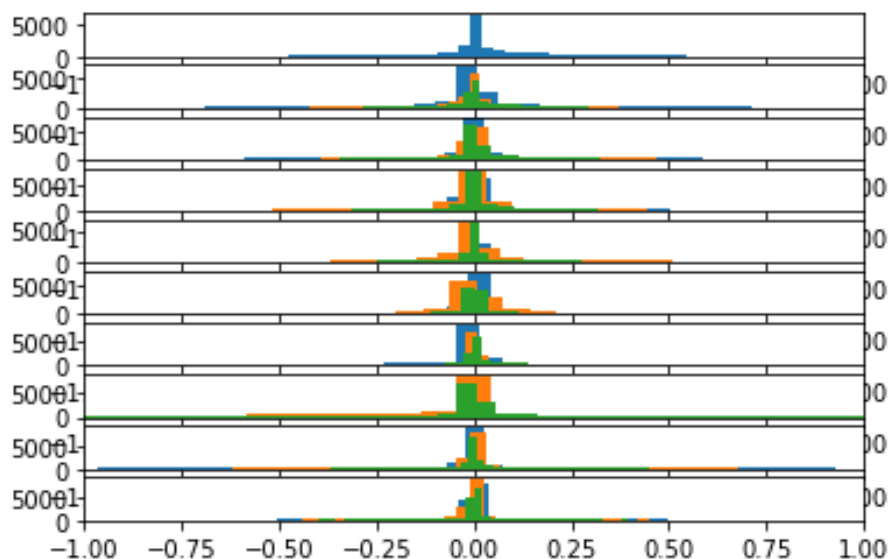
Γωνιακή ταχύτητα

Στο σχήμα 4.9 φαίνεται το ιστόγραμμα και για τους 3 άξονες, ανά υποκείμενο για τα κινησιακά δεδομένα της γωνιακής ταχύτητας από το γυροσκόπιο.

Βλέπουμε μεγάλη πιθανότητα κατανομής Gauss για κάθε άξονα σε κάθε υποκείμενο με επίκεντρο το (0,0). Οι κατανομές είναι λίγο ευρύτερες και δείχνουν να έχουν πιο παχιά



Σχήμα 4.8: Ιστογράμμο επιτάχυνσης σώματος για 10 υποκείμενα



Σχήμα 4.9: Ιστογράμμο για το δεδομένα από το γυροσκόπιο για 10 υποκείμενα

ουρά, αλλά αυτό είναι ένα ενθαρρυντικό εύρημα για μοντελοποίηση δεδομένων κίνησης σε διάφορα υποκείμενα.

4.2.4 Ιστογράμματα ανά δραστηριότητα

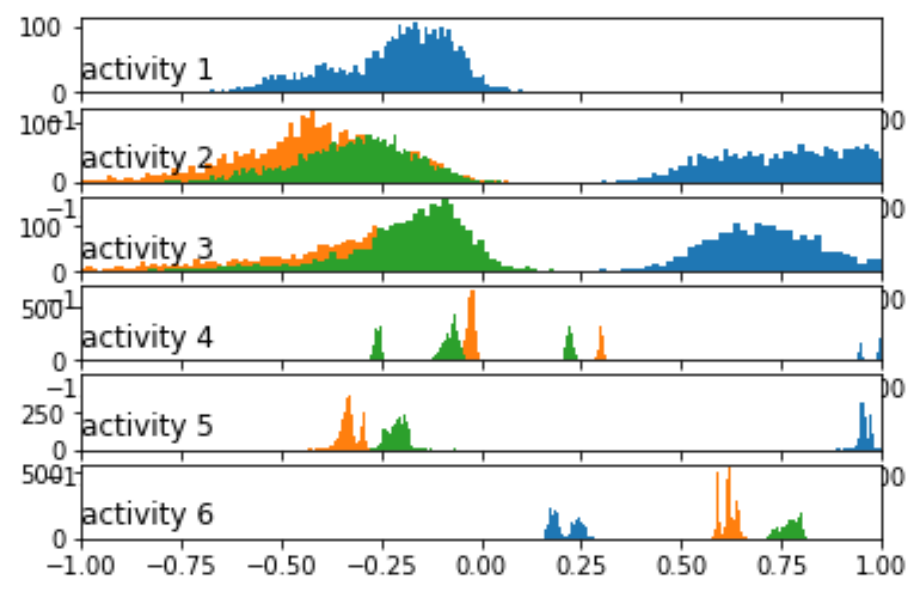
Κάτι ακόμη που μας ενδιαφέρει στα πλαίσια της ανάλυσης των δεδομένων και στην προσπάθεια μοντελοποίησής τους, είναι να κάνουμε διακρίσεις μεταξύ δραστηριοτήτων με βάση τα κινησιακά δεδομένα.

Η απλούστερη περίπτωση για μελέτη της διάκριση μεταξύ δραστηριοτήτων για ένα μόνο υποκείμενο. Μπορούμε να το πετύχουμε αυτό, δημιουργώντας ένα γράφημα ιστογράμματος ανά δραστηριότητα, με τους τρεις άξονες ενός δεδομένου τύπου/αισθητήρα δεδομένων σε κάθε γράφημα. Και πάλι, τα γραφήματα μπορούν να διευθετηθούν οριζόντια για να συ-

γκρίνουν την κατανομή κάθε άξονα δεδομένων ανά δραστηριότητα. Θα περιμέναμε να δούμε διαφορές στις κατανομές σε δραστηριότητες.

Παρακάτω, παρατίθενται τα ιστογράμματα για ένα τυχαίο υποκείμενο (12) για κάθε έναν αισθητήρα δεδομένων.

Συνολική επιτάχυνση



Σχήμα 4.10: Ιστογράμματα ανά δραστηριότητα για την συνολική επιτάχυνση ενός υποκειμένου

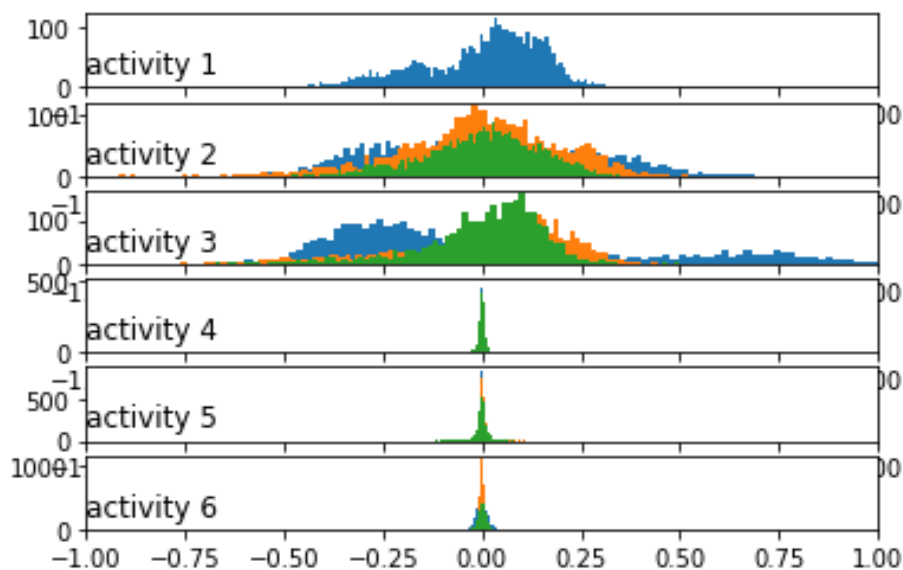
Μπορούμε να δούμε από το σχήμα 4.10 ότι κάθε δραστηριότητα έχει διαφορετική κατανομή δεδομένων, με σημαντική διαφορά μεταξύ των δραστηριοτήτων μεγάλης κίνησης (πρώτες τρεις δραστηριότητες) με τις στάσιμες δραστηριότητες (τρεις τελευταίες δραστηριότητες). Οι κατανομές δεδομένων για τις τρεις πρώτες δραστηριότητες φαίνονται Gaussian με ίσως διαφορετικά μέσα και τυπικές αποκλίσεις. Ενώ οι κατανομές για τις τελευταίες δραστηριότητες φαίνονται πολυτροπικές (δηλαδή με πολλαπλές κορυφές).

Η διαφορά που παρατηρείται μεταξύ των κατανομών των δραστηριοτήτων μεγάλης κίνησης και των στατικών περιμένουμε ότι είναι ένα χαρακτηριστικό κομβικής σημασίας στην ορθή πρόβλεψη των στατικών ή μη δραστηριοτήτων από τα μοντέλα μας.

Επιτάχυνση σώματος

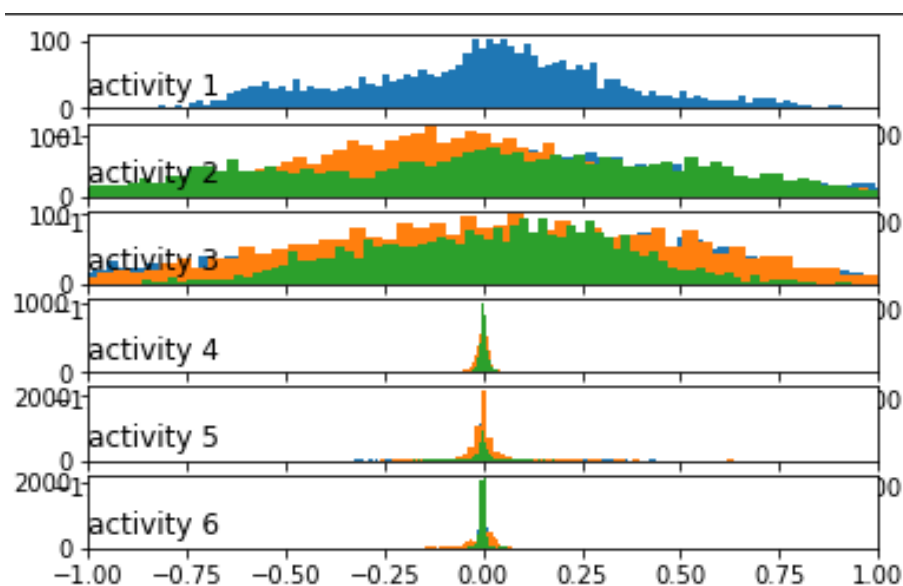
Στο σχήμα 4.11, μπορούμε να δούμε περισσότερες παρόμοιες κατανομές μεταξύ των δραστηριοτήτων μεταξύ των εν κινήσει έναντι των στατικών δραστηριοτήτων. Τα δεδομένα φαίνονται διτροπικά στην περίπτωση των δραστηριοτήτων εν κινήσει και ίσως Gaussian ή εκθετικά στην περίπτωση των στάσιμων δραστηριοτήτων.

Το μοτίβο που βλέπουμε με τις συνολικές κατανομές επιτάχυνσης έναντι των κατανομών επιτάχυνσης σώματος ανά δραστηριότητα αντικατοπτρίζει αυτό που βλέπουμε με τους ίδιους τύπους/αισθητήριων δεδομένων σε υποκείμενα στην προηγούμενη ενότητα. Ίσως τα συνολικά δεδομένα επιτάχυνσης να είναι το κλειδί για τη διάκριση των δραστηριοτήτων.



Σχήμα 4.11: Ιστόγραμμα ανά δραστηριότητα για την επιτάχυνση σώματος ενός υποκειμένου

Γωνιακή ταχύτητα



Σχήμα 4.12: Ιστόγραμμα ανά δραστηριότητα για την γωνιακή ταχύτητα ενός υποκειμένου

Οι γραφικές του σχήματος 4.12 φαίνεται να έχουν παρόμοιο μοτίβο με τα δεδομένα επιτάχυνσης σώματος, παρόλο που δείχνει ίσως κατανομές τύπου Gauss με πιο παχιά ουρά αντί για διτροπικές κατανομές για τις δραστηριότητες εν κινήσει.

4.2.5 Διάρκεια δραστηριοτήτων

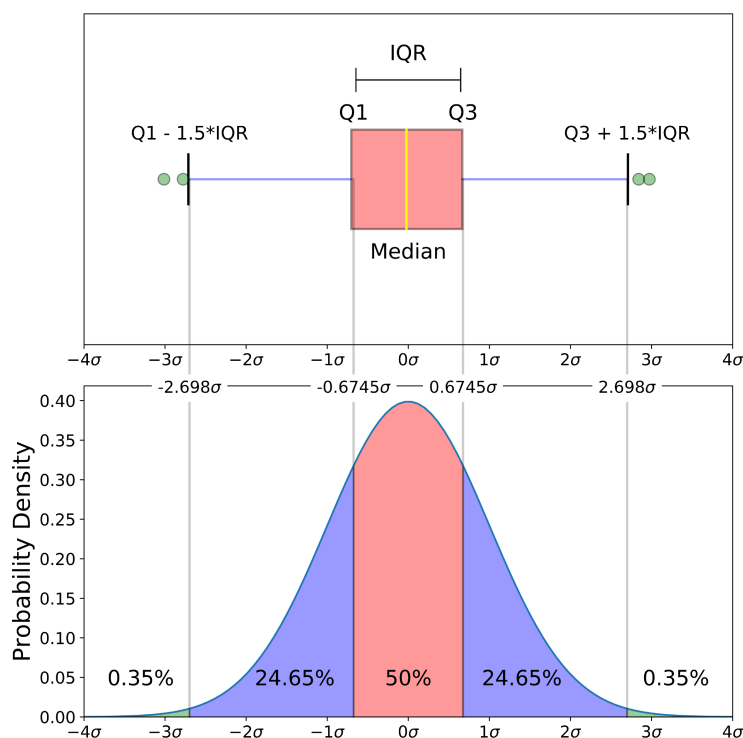
Μια τελευταία περιοχή εξερεύνησης είναι πόσο χρόνο απαιτεί η κάθε δραστηριότητα. Δεδομένου, ότι το σύνολο δεδομένων μας είναι ισορροπημένο περιμένουμε να υπάρχει ισορροπία για ένα υποκείμενο κατά την διάρκεια της καταγραφής των κινήσεων του.

Αυτό το επιβεβαιώνουμε υπολογίζοντας πόσο πολύ (σε δείγματα/χρονικά παράθυρα) ξεδεύει το κάθε υποκείμενο για κάθε δραστηριότητα. Ένας εύχρηστος τρόπος να προβάλλουμε αυτά τα δεδομένα είναι να συνοψίσουμε τις κατανομές ως boxplot.

Τα boxplots είναι ένας τυποποιημένος τρόπος εμφάνισης της κατανομής δεδομένων με βάση μια περίληψη πέντε αριθμών («ελάχιστο», πρώτο τεταρτημόριο (Q1), μέσο, τρίτο τεταρτημόριο (Q3) και «μέγιστο»). Οι πέντε αυτοί αριθμοί ορίζονται ως εξής:

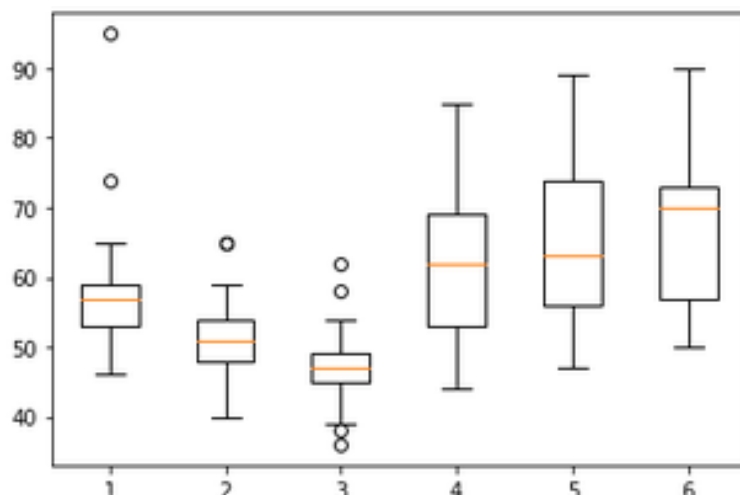
- Μέσο: Η μέση τιμή του συνόλου δεδομένων.
- Πρώτο τεταρτημόριο(Q1): Η μέση τιμή μεταξύ της μικρότερης τιμής και του μέσου.
- Τρίτο τεταρτημόριο(Q3): Η μέση τιμή μεταξύ της μεγαλύτερης τιμής και του μέσου.
- Ελάχιστο: Ορίζεται από την σχέση $Q1 - 1.5 * IQR$, όπου IQR η απόσταση μεταξύ Q3 και Q1.
- Μέγιστο: Ορίζεται από την σχέση $Q3 + 1.5 * IQR$, όπου IQR η απόσταση μεταξύ Q3 και Q1.

Σχηματικά αυτά απεικονίζονται στο σχήμα 4.13



Σχήμα 4.13: Παράδειγμα boxplot [18].

Στην περίπτωση μας που θέλουμε να μελετήσουμε την διάρκεια των δραστηριοτήτων ένας βολικός τρόπος, όπως αναφέραμε, είναι να συνοψίσουμε τις κατανομές ως boxplot που εμφανίζουν τη μέση (γραμμή), το μεσαίο 50% (πλαίσιο), τη γενική έκταση των δεδομένων ως το διατημηματικό εύρος (τα μουστάκια) και τις ακραίες τιμές (ως τελείες).

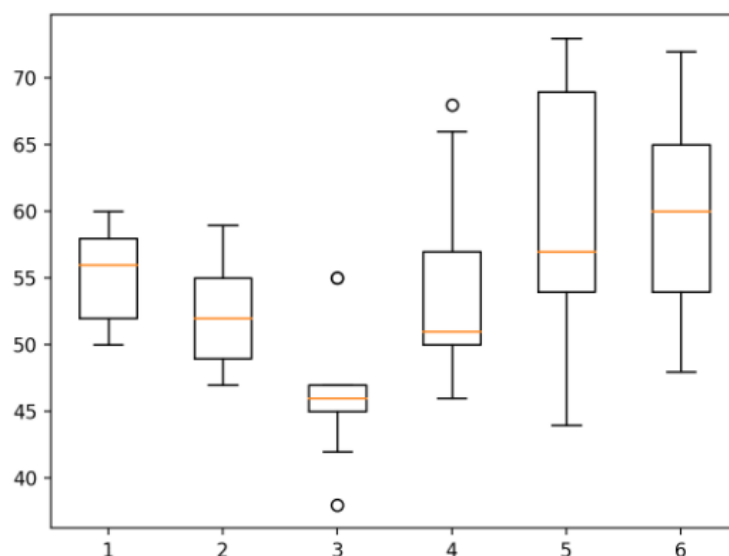


Σχήμα 4.14: *Boxplot για την διάρκεια των δραστηριοτήτων ανά υποκείμενο στο training set.*

Παρατηρούμε στο σχήμα 4.14 ότι οι στατικές δραστηριότητες ξοδεύουν περισσότερο χρόνο συγκριτικά με τις υπόλοιπες. Επίσης, παρατηρούμε ότι η τρίτη κατά σειρά δραστηριότητα κρατάει τον λιγότερο χρόνο.

Η διαφορά στις κατανομές των δραστηριοτήτων δεν είναι μεγάλη, γεγονός που υποδηλώνει μικρή ανάγκη περικοπής των δραστηριοτήτων μεγαλύτερης διάρκειας ή υπερβολικής δειγματοληψίας των δραστηριοτήτων κίνησης.

Στο σχήμα 4.15 παρουσιάζεται το αντίστοιχο boxplot για την διάρκεια των δραστηριοτήτων ανά υποκείμενο στο σύνολο επαλήθευσης. Τα αποτελέσματα είναι ενθαρρυντικά καθώς υποδηλώνουν ότι πράγματι το σύνολο επαλήθευσης και εκπαίδευσης είναι λογικά αντιπροσωπευτικά του συνόλου του συνόλου δεδομένων.



Σχήμα 4.15: *Boxplot για την διάρκεια των δραστηριοτήτων ανά υποκείμενο στο test set.*

Κεφάλαιο 5

Εφαρμογή μεθόδων ταξινόμησης και αποτελέσματα

Σε αυτό το κεφάλαιο θα παρουσιαστούν οι αρχιτεκτονικές και τα αποτελέσματα των αλγορίθμων μηχανικής μάθησης για την ταξινόμηση των ανθρώπινων δραστηριοτήτων του συνόλου δεδομένων που περιγράφηκε ήδη. Πιο συγκεκριμένα, θα χρησιμοποιηθούν τα 3 είδη των Αναδρομικών νευρωνικών δικτύων που περιγράφηκαν σε προηγούμενη ενότητα [3.1](#) RNNs, LSTMs, GRUs, η υβριδική αρχιτεκτονική CNN + RNN και τα TCN νευρωνικά δίκτυα.

5.1 Αρχιτεκτονική Αναδρομικών Δικτύων

Στην ενότητα αυτή παρουσιάζονται οι βασικές αρχιτεκτονικές που στηρίζονται στα αναδρομικά νευρωνικά δίκτυα.

5.1.1 Επίπεδο αναδρομικού νευρωνικού δικτύου

Το πρώτο επίπεδο της αρχιτεκτονικής μας είναι το αναδρομικό νευρωνικό δίκτυο. Πραγματοποιήθηκαν 3 πειράματα με την συγκεκριμένη αρχιτεκτονική με διαφορετικό αναδρομικό νευρωνικό δίκτυο στο πρώτο επίπεδο κάθε φορά (RNNs, LSTMs, GRUs). Σημειώνουμε, ότι πραγματοποιήθηκε πείραμα και με το νευρωνικό δίκτυο TCN σε αυτή τη θέση.

Οι βασικές παράμετροι των αναδρομικών νευρωνικών δικτύων είναι ο αριθμός των διαφορετικών αναδρομικών μονάδων που αποτελούνται καθώς και το διάνυσμα εισόδου που στην περίπτωση μας είναι ίσο με [αριθμός χρονικών βημάτων, αριθμός χαρακτηριστικών].

5.1.2 Επίπεδο εγκατάλειψης

Το επόμενο επίπεδο που ακολουθεί στην συγκεκριμένη αρχιτεκτονική είναι το επίπεδο εγκατάλειψης. Το συγκεκριμένο επίπεδο περιγράφεται αναλυτικά στην ενότητα [3.3.3](#).

Ως παράμετρο το παρόν επίπεδο δέχεται ένα αριθμό, ο οποίος αναπαριστά την πιθανότητα p με την οποία μηδενίζεται κάθε κρυφή μονάδα.

5.1.3 Πυκνό Επίπεδο

Στην περίπτωση των δικτύων ταξινόμησης, η έξοδος των προηγούμενων επιπέδων αναδιατάσσεται σε διάνυσμα (flatten) (αν η έξοδος των προηγούμενων επιπέδων δεν είναι μονοδιάστατη) και δίνεται ως είσοδος σε ένα σύνολο πλήρως συνδεδεμένων επιπέδων, το οποίο αναλαμβάνει την τελική ταξινόμηση των αρχικών χαρακτηριστικών. Τα επίπεδα αυτά συνήθως χρησιμοποιούν σαν συνάρτηση ενεργοποίησης τη ReLU, με εξαίρεση το τελευταίο το οποίο χρησιμοποιεί την SoftMax:

$$\sigma(x)_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \quad (5.1)$$

Για την συγκεκριμένη αρχιτεκτονική δεν χρειαστήκαμε επίπεδο flatten ενώ ταυτόχρονα προσθέτουμε δύο πυκνά (fully connected) επίπεδα, το πρώτο με ReLU ως συνάρτηση ενεργοποίησης και το δεύτερο με SoftMax για την τελική ταξινόμηση του δικτύου με τις σχετικές παραμέτρους.

5.1.4 Φόρτωση δεδομένων

Όπως έχει ήδη περιγραφεί υπάρχουν τρεις κύριοι τύποι σημάτων στα ακατέργαστα δεδομένα: ολική επιτάχυνση, επιτάχυνση σώματος και γυροσκόπιο σώματος. Το καθένα έχει 3 άξονες δεδομένων. Αυτό σημαίνει ότι υπάρχουν συνολικά εννέα μεταβλητές για κάθε χρονικό βήμα. Επιπλέον, κάθε σειρά δεδομένων έχει χωριστεί σε αλληλεπικαλυπτόμενα παράθυρα δεδομένων 2,56 δευτερολέπτων, ή 128 χρονικά βήματα. Αυτά τα παράθυρα δεδομένων αντιστοιχούν στα παράθυρα των σχεδιασμένων χαρακτηριστικών (σειρών) στην προηγούμενη ενότητα. Αυτό σημαίνει ότι μια σειρά δεδομένων έχει $(128 * 9)$ ή 1.152 στοιχεία.

Τα σήματα είναι αποθηκευμένα σε έναν αρχικό φάκελο του συνόλου δεδομένων μας, μέσα στους υποφακέλους εκπαίδευσης και δοκιμών. Κάθε άξονας, κάθε σήματος αποθηκεύεται σε ξεχωριστό αρχείο, πράγμα που σημαίνει ότι καθένα από τα σύνολα δεδομένων εκπαίδευσης και δοκιμών έχει εννέα αρχεία εισόδου για φόρτωση και ένα αρχείο εξόδου για φόρτωση. Μπορούμε να ομαδοποιήσουμε τη φόρτωση αυτών των αρχείων σε ομάδες, δεδομένης της συνεπούς δομής του αρχικού φακέλου του συνόλου δεδομένων μας και των συμβάσεων ονοματοδοσίας αρχείων.

Τα δεδομένα εισόδου είναι σε μορφή CSV όπου οι στήλες χωρίζονται με κενό χώρο. Μπορούμε να φορτώσουμε όλα τα δεδομένα για μια δεδομένη ομάδα (εκπαίδευσης ή δοκιμής) σε έναν τριδιάστατο πίνακα NumPy, όπου οι διαστάσεις του πίνακα είναι [δείγματα, χρονικά βήματα, χαρακτηριστικά]. Για να γίνει πιο ξεκάθαρο, υπάρχουν 128 χρονικά βήματα και εννέα χαρακτηριστικά, όπου ο αριθμός των δειγμάτων είναι ο αριθμός των γραμμών σε κάθε αρχείο δεδομένων ακατέργαστου σήματος.

Τα δεδομένα εξόδου ορίζονται ως ακέραιοι με τον αριθμό κλάσης. Πρέπει να μετατρέψουμε αυτούς τους ακέραιους κλάσης σε τέτοια μορφή έτσι ώστε τα δεδομένα μας να είναι κατάλληλα για την προσαρμογή ενός μοντέλου ταξινόμησης πολλαπλών κατηγοριών νευρωνικών δικτύων. Η μορφή αυτή είναι τα one-hot διανύσματα τα οποία έχουν μήκος όσο ο μέγιστος ακέραιος κλάσης και έχουν 1 μόνο στην θέση που υποδεικνύει αυτός ο ακέραιος και 0 στις υπόλοιπες θέσεις.

Εφόσον φορτώσουμε τα δεδομένα μας, είμαστε έτοιμοι να τα περάσουμε ως είσοδο στο πρώτο επίπεδο της αρχιτεκτονικής μας.

5.2 Υβριδική Αρχιτεκτονική

Στην παρούσα ενότητα παρουσιάζονται τα επίπεδα της αρχιτεκτονικής που επιλέχθηκε ώστε να υλοποιηθεί η θεωρητική υβριδική αρχιτεκτονική, όπως αυτή περιγράφηκε στην ενότητα 3.3, έτσι ώστε να καλύπτονται οι ανάγκες του προβλήματος μας.

5.2.1 Φόρτωση δεδομένων

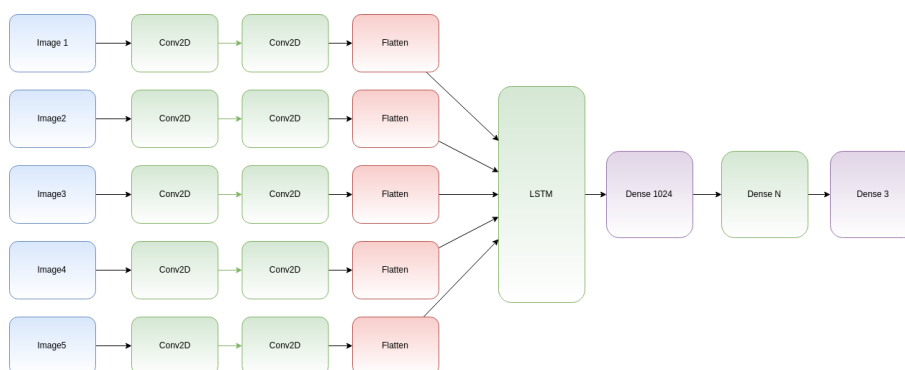
Δεδομένου ότι σε αυτό το μοντέλο στα πρώτα επίπεδα θα υπάρχουν συνελκτικά στρώματα η προσέγγιση που κάναμε για τα δεδομένα εισόδου είναι να χωρίσουμε κάθε παράθυρο 128 βημάτων σε δευτερεύοντα βήματα για την επεξεργασία τους μέσω των μοντέλων CNNs. Για παράδειγμα, τα 128 χρονικά βήματα σε κάθε παράθυρο μπορούν να χωριστούν σε τέσσερις υποδιαίρέσεις των 32 χρονικών βημάτων.

5.2.2 Κατανομή Χρόνου

Πριν ξεκινήσουμε να παραθέτουμε τα διάφορα επίπεδα της αρχιτεκτονικής μας σκόπιμη κρίνεται η παρουσίαση των επιπέδων με κατανομή χρόνου.

Αυτά αποτελούν μια τεχνική που επιτρέπει την εφαρμογή ενός επιπέδου σε κάθε χρονικό κομμάτι μιας εισόδου.

Κάθε είσοδος θα πρέπει να είναι τουλάχιστον τρισδιάστατη και η πρώτη διάσταση της πρώτης εισόδου θα θεωρείται ότι είναι η χρονική διάσταση. Για παράδειγμα, αν εξετάζαμε μια παρτίδα 32 δειγμάτων βίντεο, όπου κάθε δείγμα είναι μια εικόνα RGB 128x128 pixels σε 10 χρονικά βήματα τότε το σχήμα εισόδου παρτίδας είναι (32, 10, 128, 128, 3). Στην συνέχεια, μπορούμε να χρησιμοποιήσουμε το επίπεδο με κατανομή χρόνου/Time Distributed για να εφαρμόσουμε το ίδιο επίπεδο Conv2D σε καθένα από τα 10 χρονικά βήματα, ανεξάρτητα. Στο σχήμα 5.1 φαίνεται ένα παράδειγμα αρχιτεκτονικής που δέχεται ως είσοδο κάποια δείγματα βίντεο τα οποία επεξεργάζονται με συνελκτικά δίκτυα, μέσω επιπέδων με κατανομή χρόνου πριν μπουν ως είσοδο σε κάποιο αναδρομικό νευρωνικό δίκτυο (LSTM).



Σχήμα 5.1: Αρχιτεκτονική με Time Distributed επίπεδα [19]

Στην περίπτωση μας οι διαστάσεις του επιπέδου με κατανομή χρόνου/Time Distributed θα είναι (αριθμός δειγμάτων, 4, 32, 9) και αντί για Conv2D χρησιμοποιούνται μονοδιάστατα συνελκτικά δύκτια, όπως έχουν ήδη περιγραφθεί.

5.2.3 Υπόλοιπα επίπεδα

Σε αυτή την ενότητα θα γίνει μια σύντομη περιγραφή για τα επίπεδα της αρχιτεκτονικής στα οποία εφαρμόζεται το επίπεδο κατανομής χρόνο καθώς και για τα επίπεδα που τα ακολουθούν.

Μονοδιάστατα συνελκτικά

Όπως ήδη αναφέρθηκε τα πρώτα χρονικά κατανεμημένα επίπεδα είναι μονοδιάστατα συνελκτικά επίπεδα προκειμένου να εξαγάγουμε τα χαρακτηριστικά κάθε χρονικού βήματος. Πιο συγκεκριμένα, χρησιμοποιήθηκαν δύο τέτοια επίπεδα.

Επίπεδο Εγκατάλειψης

Τα δύο συνελκτικά επίπεδα ακολουθεί ένα επίπεδο εγκατάλειψης προκειμένου να αποφευχθεί η υπερεκπαίδευση του δικτύου καθώς και να απαλειφθούν πιθανώς περιττές πληροφορίες.

Μονοδιάστατο επίπεδο μέγιστης δειγματοληψίας

Το συγκεκριμένο επίπεδο υποδειγμαληπτεί την έξοδο των προηγούμενων επιπέδων παίρνοντας τη μέγιστη τιμή σε ένα χωρικό παράθυρο συγκεκριμένου μεγέθους που εμείς επιλέγουμε. Για περισσότερες λεπτομέρειες για αυτό το επίπεδο μπορείτε να δείτε στο [3.3.1](#)

Επίπεδο Αναδιάταξης

Αυτό το επίπεδο αναδιατάζει την είσοδο σε κατάλληλη μορφή ώστε να μπορεί να αποτελέσει είσοδο σε ένα αναδρομικό νευρωνικό δίκτυο. Η είσοδος που δέχεται αυτό το επίπεδο είναι της μορφής (αριθμός δειγμάτων, χρονικά βήματα, χαρακτηριστικά, αριθμός καναλιών εξόδου), ενώ η έξοδος που δίνει είναι της μορφής (αριθμός δειγμάτων, χρονικά βήματα, χαρακτηριστικά * αριθμός καναλιών εξόδου).

Αναδρομικό επίπεδο και επίπεδα ταξινόμησης

Τα επίπεδα που ακολουθούν τα παραπάνω δεν είναι πλέον χρονικά κατανεμημένα. Παρουσιάζουν την ίδια ακριβώς αρχιτεκτονική με την αρχιτεκτονική αναδρομικών νευρωνικών δικτύων που περιγράφηκε προηγουμένως στην ίδια ενότητα.

Πιο συγκεκριμένα, χρησιμοποιήθηκε ένα αναδρομικό νευρωνικό δίκτυο (RNN, GRU, LSTM) ή ένα δίκτυο TCN. Κατόπιν, ένα επίπεδο εγκατάλειψης και τέλος δύο πυκνά επίπεδα προκειμένου να επιτευχθεί η ταξινόμηση του παρόντος δείγματος.

5.3 Πειραματικά αποτελέσματα

5.3.1 Εφαρμογή RNN νευρωνικού δικτύου

Η πρώτη μέθοδος ταξινόμησης που εφαρμόστηκε ήταν μέσω των νευρωνικών δικτύων RNN. Για την εκπαίδευσή τους χρησιμοποιήθηκαν τα δεδομένα εκπαίδευσης (training data) και τα δεδομένα ελέγχου (test data) όπως έχουν χωριστεί από το σύνολο δεδομένων μας.

Το νευρωνικό δίκτυο RNN, έχει αρκετές παραμέτρους που πρέπει να προσδιοριστούν πριν γίνει η εκπαίδευση του. Κάποιες από αυτές είναι, το πόσα επίπεδα RNN θα χρησιμοποιήσουμε, πόσες μονάδες θα έχει το κάθε επίπεδο, η συνάρτηση ενεργοποίησης του, αλλά και το αν το νευρωνικό δίκτυο θα επιστρέφει μια τιμή για κάθε χρονική στιγμή (return sequence = True) ή αν θα επιστρέφει μόνο την τελική τιμή της τελευταίας χρονικής στιγμής (return sequence = False). Ακόμα, για την εκπαίδευση του μοντέλου θα πρέπει να οριστεί ο αλγόριθμος βελτιστοποίησης αλλά και το batch size, δηλαδή μετά από κάθε πόσα πακέτα δεδομένων μορφής (batch size, timesteps, features) θα ανανεώνονται οι κλίσεις (gradients). Για τον καθορισμό των υπερπαραμέτρων του μοντέλου έγιναν δοκιμές σε ένα τμήμα των δεδομένων για την επιλογή των βέλτιστων και χρησιμοποιήθηκε η τεχνική της διασταυρωμένη επικύρωση (cross-validation). Οι παράμετροι που ελέγχθηκαν φαίνονται στον 5.1.

Υπερπαραμέτροι	Εναλλακτικές	Βέλτιστες
batch size	16, 32, 64	64
epochs	25, 50	25
optimizer	'Adam', 'SGD'	'Adam'
loss	'binary crossentropy', 'categorical crossentropy', 'poisson'	'categorical crossentropy'
activation	'ReLU', 'tanh'	'ReLU'
dropout rate	0.2, 0.5, 0.8	0.5
rnn units	64, 128, 256	128

Πίνακας 5.1: Πίνακας υπερπαραμέτρων RNN αρχιτεκτονικής

Οι βέλτιστες υπερπαραμέτροι οδήγησαν την συγκεκριμένη αρχιτεκτονική σε σκορ ακρίβειας (accuracy) 80.83%. Πιο αναλυτικά στον πίνακα 5.2 φαίνεται ο πίνακας σύγχυσης (confusion matrix) ο οποίος δείχνει αναλυτικά τις ταξινομήσεις που έγιναν καθώς μπορούν να φανούν και οι αστοχίες του δικτύου.

Οι περισσότερες αστοχίες του δικτύου μας παρατηρούνται μεταξύ των δραστηριοτήτων 1 και 2 καθώς το 37% των δραστηριοτήτων της κατηγορίας 1 ταξινομήθηκαν στην κατηγορία 2. Μια ακόμη αδυναμία του δικτύου εντοπίζεται στην ταξινόμηση των δραστηριοτήτων 4 και 5 όπου 176 εγγραφές της δραστηριότητας 4 ταξινομήθηκαν ως δραστηριότητα 5.

5.3.2 Εφαρμογή GRU νευρωνικού δικτύου

Η δεύτερη μέθοδος ταξινόμησης που εφαρμόστηκε ήταν μέσω των νευρωνικών δικτύων GRU. Για την εκπαίδευσή τους χρησιμοποιήθηκαν τα δεδομένα εκπαίδευσης (training data) και τα δεδομένα ελέγχου (test data) όπως ακριβώς και προηγουμένως.

Προκειμένου να πετύχουμε τα βέλτιστα αποτελέσματα για την εκπαίδευση ενός GRU

1	2	3	4	5	6
313	30	0	12	10	0
176	364	14	1	0	2
7	72	399	2	0	24
0	0	0	300	26	1
0	0	7	176	496	0
0	5	0	0	0	510

Πίνακας 5.2: Πίνακας σύγκρισης RNN αρχιτεκτονικής

δικτύου και κατ' επέκταση όλης της αρχιτεκτονικής που υλοποιήσαμε χρειάστηκε να μεταβάλουμε αρκετές παραμέτρους. Οι βασικές παράμετροι που σχετίζονται με την διαδικασία εκπαίδευσης που μεταβάλαμε είναι ο αριθμός των εποχών (epochs), το μέγεθος παρτίδας (batch size), η συνάρτηση σφάλματος που χρησιμοποιείται κατά τον αλγόριθμο back propagation καθώς και ο βελτιστοποιητής (optimizer) που χρησιμοποιήθηκε. Επιπλέον, πειραματιστήκαμε και με τον ρυθμό εγκατάλειψης (dropout rate) στο επίπεδο εγκατάλειψης καθώς και τις συναρτήσεις ενεργοποίησης τόσο στο επίπεδο του αναδρομικού δικτύου (GRU) όσο και στα πυκνά επίπεδα (Dense layers). Τέλος, δοκιμάστηκαν κάποιες τιμές και για τον αριθμό των μονάδων του GRU δικτύου ώστε να επιλεχθεί ο βέλτιστος αριθμός. Στο πίνακα 5.3 παρουσιάζονται αναλυτικά οι τιμές που δοκιμάστηκαν καθώς και αυτές που οδήγησαν σε καλύτερα αποτελέσματα σύμφωνα με την μετρική ακρίβειας (accuracy).

Υπερπαραμέτροι	Εναλλακτικές	Βέλτιστες
batch size	16, 32, 64	64
epochs	25, 50	25
optimizer	'Adam', 'SGD'	'Adam'
loss	'binary crossentropy', 'categorical crossentropy', 'poisson'	'categorical crossentropy'
activation	'ReLU', 'tanh'	'ReLU'
dropout rate	0.2, 0.5, 0.8	0.5
gru units	64, 128, 256	128

Πίνακας 5.3: Πίνακας υπερπαραμέτρων GRU αρχιτεκτονικής

Παρατηρούμε ότι για τις βέλτιστες παραμέτρους σημειώνεται ακρίβεια 92,8% στην ταξινόμηση των δραστηριοτήτων. Αυτή η αύξηση θα μπορούσε να δικαιολογηθεί από την αδυναμία μακροπρόθεσμης μνήμης των απλών αναδρομικών δικτύων (RNNs) όπως έχει εξηγηθεί αναλυτικά στο κεφάλαιο της θεωρίας. Τα χρονικά βήματα που δέχεται σαν είσοδο το αναδρομικό μας δίκτυο είναι 128 συνεπώς φαίνεται το GRU να μπορεί να ανταποκριθεί καλύτερα στις απαιτήσεις των δεδομένων. Ακόμη, παρατηρούμε ότι το δίκτυο GRU απαιτεί περισσότερες εποχές από το απλό αναδρομικό νευρωνικό δίκτυο για να εκπαιδευτεί και να μπορέσει να δώσει μεγαλύτερη ακρίβεια.

Στον πίνακα σύγκρισης 5.4 φαίνονται αναλυτικά οι ταξινομήσεις που έκανε το μοντέλο μας. Παρατηρούμε ότι το μοντέλο μας βελτιώνει δραματικά τις σωστές ταξινομήσεις για την δραστηριότητα 2 φτάνοντας σε ακρίβεια προβλέψεων σχεδόν 100% η οποία παρουσιάζει α-

ύξηση 27% από το προηγούμενο μοντέλο. Αντίστοιχη βελτίωση στην ακρίβεια προβλέψεων παρουσιάζεται και για την δραστηριότητα 1 όπου είναι 29% βελτιωμένη συγκριτικά με την προηγούμενη αρχιτεκτονική με τις όποιες αστοχίες να οφείλονται στην πρόβλεψη των δραστηριότητας 2 και 3. Η δραστηριότητα 3 παρουσιάζει και αυτή ακρίβεια προβλέψεων σχεδόν 100% όπως ακριβώς και η δραστηριότητα 2.

Για τις στατικές δραστηριότητες (4-6) πετυχαίνουμε μεγαλύτερη ακρίβεια συγκριτικά με την προηγούμενη αρχιτεκτονική, παρόλα αυτά φαίνεται να υπάρχουν ακόμη περιθώρια βελτίωσης στην ταξινόμηση μεταξύ των δραστηριοτήτων 4 και 5. Πιο συγκεκριμένα, το 24% της δραστηριότητας 4 ταξινομείται στην 5. Επιπλέον, παρατηρούμε ότι η δραστηριότητα 6 προβλέπεται με απόλυτη επιτυχία καθώς κανένα δείγμα του συνόλου δεδομένων μας ταξινομήθηκε λανθασμένα σε άλλη δραστηριότητα και μόλις 1 δείγμα ταξινομήθηκε λανθασμένα σε αυτή την δραστηριότητα. Τέλος, αξίζει να σημειώσουμε ότι εάν το πρόβλημα που μας ενδιέφερε ήταν ο διαχωρισμός των δραστηριοτήτων σε στατικές ή μη δραστηριότητες το μοντέλο μας θα ήταν ιδανικό με βάση τα αποτελέσματα του πίνακα 5.4.

1	2	3	4	5	6
457	4	6	0	0	0
29	464	5	8	1	0
10	3	409	0	0	0
0	0	0	389	50	0
0	0	0	93	481	0
0	0	0	1	0	537

Πίνακας 5.4: Πίνακας σύγχυσης GRU αρχιτεκτονικής

5.3.3 Εφαρμογή LSTM νευρωνικών δικτύων

Η επόμενη μέθοδος που εφαρμόστηκε για την ταξινόμηση των δραστηριοτήτων ήταν μέσω αναδρομικών δικτύων LSTM.

Οι παράμετροι που προσδιορίστηκαν για την παρούσα αρχιτεκτονική είναι ίδιες με την προηγούμενη και αναγράφονται στον πίνακα 5.5 καθώς τα δίκτυα GRU και LSTM χρησιμοποιούνται ακριβώς με τον ίδιο τρόπο.

Υπερπαράμετροι	Εναλλακτικές	Βέλτιστες
batch size	16, 32, 64	64
epochs	25, 50	25
optimizer	'Adam', 'SGD'	'Adam'
loss	'binary crossentropy', 'categorical crossentropy', 'poisson'	'categorical crossentropy'
activation	'ReLU', 'tanh'	'ReLU'
dropout rate	0.2, 0.5, 0.8	0.5
units	64, 128, 256	128

Πίνακας 5.5: Πίνακας υπερπαραμέτρων LSTM αρχιτεκτονικής

Παρατηρούμε ότι για τις βέλτιστες παραμέτρους σημειώνεται ακρίβεια 92.6%. Η επίδοση

της παρούσας αρχιτεκτονικής είναι παρόμοια με της αρχιτεκτονικής με GRU αναδρομικό νευρωνικό δίκτυο. Αυτό είναι κάτι που αναμέναμε, σύμφωνα με την θεωρία, να δούμε καθώς τα GRUs αποτελούν απλοποιημένη παραλλαγή των LSTMs.

Στον πίνακα σύγχυσης 5.6 φαίνονται αναλυτικά οι ταξινομήσεις που έκανε το μοντέλο μας. Συγκριτικά με τις ταξινομήσεις που έκανε το μοντέλο με το RNN ισχύουν αντίστοιχες παρατηρήσεις όπως με το μοντέλο με το GRU. Συγκριτικά με τον πίνακα όμως 5.4 παρατηρούμε ότι το παρόν μοντέλο εμφανίζεται με ακόμα πιο βελτιωμένες ταξινομήσεις στις δραστηριότητες 1-3 καθώς η ακρίβεια για κάθε μια από αυτές είναι για την πρώτη 95%, την δεύτερη 99,7% και για την τρίτη 99,5%. Παρόλα αυτά στις στατικές δραστηριότητες υπάρχουν πιο πολλές αστοχίες τόσο για την 4 όσο και για την 5.

1	2	3	4	5	6
471	0	2	1	1	0
16	470	0	1	1	0
9	1	418	0	0	0
0	0	0	372	68	0
0	0	0	117	462	0
0	0	0	0	0	537

Πίνακας 5.6: Πίνακας σύγχυσης LSTM αρχιτεκτονικής

5.3.4 Εφαρμογή TCN νευρωνικών δικτύων

Η τέταρτη μέθοδος ταξινόμησης που εφαρμόστηκε ήταν μέσω των νευρωνικών δικτύων TCN. Για την εκπαίδευσή τους χρησιμοποιήθηκαν τα δεδομένα εκπαίδευσης (training data) και τα δεδομένα ελέγχου (test data) όπως ακριβώς και προηγουμένως.

Προκειμένου να πετύχουμε τα βέλτιστα αποτελέσματα για την εκπαίδευση της αρχιτεκτονικής μας χρειάστηκε να ελέγξουμε αρκετές υπερπαραμέτρους, τόσο καθοριστικές για την εκπαίδευση του μοντέλου μας όσο και για καθένα στρώμα. Οι βασικές υπερπαραμέτροι που σχετίζονται με την διαδικασία εκπαίδευσης είναι οι ίδιες και με τα προηγούμενα πειράματα. Για το TCN επίπεδο μελετήσαμε την παράμετρο use skip connectios, η οποία καθορίζει αν θα έχουμε σύνδεση μεταξύ της εισόδου ενός υπολειμματικού μπλοκ με την έξοδο του και το μέγεθος πυρήνα που χρησιμοποιούν τα επίπεδα των μονοδιάστατων CNNs. Ακόμη, ελέγχθηκε ο αριθμός των φίλτρων (nb size) που χρησιμοποιήθηκε από τα επίπεδα των CNNs (η αντίστοιχη παράμετρος στα LSTMs είναι η unit). Άλλη μια παράμετρος είναι η παράμετρος του γεμίσματος (padding) που καθορίζει αν θα είναι το γέμισμα αιτιώδης όπως στην θεωρητική υλοποίηση ή όχι. Τέλος, πειραματιστήκαμε και με τον ρυθμό εγκατάλειψης (dropout rate) στο επίπεδο εγκατάλειψης καθώς και τις συναρτήσεις ενεργοποίησης τόσο στο επίπεδο του TCN όσο και στα πυκνά επίπεδα (Dense layers). Στο πίνακα 5.7 παρουσιάζονται αναλυτικά οι τιμές που δοκιμάστηκαν καθώς και αυτές που οδήγησαν σε καλύτερα αποτελέσματα σύμφωνα με την μετρική ακρίβειας (accuracy).

Για την παρούσα αρχιτεκτονική με τις βέλτιστες παραμέτρους παρατηρούμε ακρίβεια 93,35%, μεγαλύτερη από όλες τις προηγούμενες. Παρόλο που η εσωτερική δομή ενός TCN διαφέρει εντελώς από αυτή των GRU, LSTM παρατηρούμε ότι οι επιδόσεις είναι πολύ

Υπερπαράμετροι	Εναλλακτικές	Βέλτιστες
batch size	16, 32, 64	64
epochs	25, 50	25
optimizer	'Adam', 'SGD'	'Adam'
loss	'binary crossentropy', 'categorical crossentropy', 'poisson'	'categorical crossentropy'
activation	'ReLU', 'tanh'	'tanh'
kernel size	3, 5	3
use skip connections	True, False	True
padding	causal, same	causal
nb size	64, 128	64
dropout rate	0.2, 0.5, 0.8	0.5

Πίνακας 5.7: Πίνακας υπερπαραμέτρων TCN αρχιτεκτονικής

κοντινές. Αυτό δείχνει ότι για το παρόν πρόβλημα και τα δύο είδη νευρωνικών δικτύων επεξεργάζονται εξίσου ικανοποιητικά τα δεδομένα των χρονικών ακολουθιών μας.

Στον πίνακα σύγκρισης 5.8 φαίνονται αναλυτικά οι ταξινομήσεις που έκανε το μοντέλο μας. Για τις εν κινήσει δραστηριότητες παρατηρούμε παρόμοια αποτελέσματα με τα προηγούμενα δύο μοντέλα. Το μόνο που θα μπορούσαμε να σημειώσουμε σαν 'παρατήρηση είναι ότι όλες οι άστοχες ταξινόμησης τόσο των δραστηριοτήτων 1 και 2 είναι γιατί προβλέπονται ως 3. Για τις στατικές δραστηριότητες παρατηρούμε ότι παρόλο που η ακρίβεια ταξινόμησης για την δραστηριότητα 4 είναι μεγαλύτερη συγκριτικά με τον 5.6, διαισθητικά οι αστοχίες που συμβαίνουν είναι μεγαλύτερες καθώς ταξινομούνται δραστηριότητες της κατηγορίας 'Κάθομαι' ως 'Περπάτημα προς τα επάνω'. Για της δραστηριότητες 5 και 6 ισχύουν τα ίδια που σχολιάστηκαν για τα προηγούμενα μοντέλα.

1	2	3	4	5	6
479	1	0	0	1	0
2	454	0	22	0	0
15	16	420	0	0	0
0	0	0	391	61	0
0	0	0	73	470	0
0	0	0	5	0	537

Πίνακας 5.8: Πίνακας σύγκρισης TCN αρχιτεκτονικής

5.3.5 Εφαρμογή CNN + RNN αρχιτεκτονικής

Στις παρακάτω υποενότητες παρουσιάζονται τα πειραματικά αποτελέσματα για τις υβριδικές αρχιτεκτονικές. Η πρώτη μέθοδος ταξινόμησης που εφαρμόστηκε για τα υβριδικά μοντέλα ήταν μέσω των νευρωνικών δικτύων CNN(s) + RNN(s). Για την εκπαίδευσή τους χρησιμοποιήθηκαν τα δεδομένα εκπαίδευσης (training data) και τα δεδομένα ελέγχου (test data) όπως ακριβώς και προηγουμένως.

Με στόχο την εύρεση των βέλτιστων παραμέτρων για την επίτευξη της βέλτιστης λει-

τουργίας του μοντέλου μας χρειάστηκε να ελέγξουμε αρκετές παραμέτρους, οι οποίες είναι τόσο καθοριστικές για την εκπαίδευση του μοντέλου μας όσο και για την λειτουργία των δικτύων των CNNs και των RNNs που χρησιμοποιήθηκαν. Οι βασικές υπερπαραμέτροι που σχετίζονται με την διαδικασία εκπαίδευσης είναι οι ίδιες και με τα προηγούμενα πειράματα. Οι υπερπαραμέτροι που σχετίζονται με την λειτουργία των συνελκτικών δικτύων και δοκιμάστηκαν στα πλαίσια των πειραμάτων είναι ο αριθμός των φίλτρων των συνελκτικών επιπέδων (filters) και το μέγεθος του πυρήνα (kernel size). Για τα απλά αναδρομικά νευρωνικά δίκτυα ελέγχθηκαν οι ίδιες υπερπαραμέτροι με την απλή RNN αρχιτεκτονική. Τέλος, πειραματιστήκαμε και με τον ρυθμό εγκατάλειψης (dropout rate) στο επίπεδο εγκατάλειψης καθώς και τις συναρτήσεις ενεργοποίησης σε όλα τα επίπεδα της αρχιτεκτονικής μας. Αναλυτικά οι τιμές που δοκιμαστήκαν καθώς και οι βέλτιστες τιμές φαίνονται στον πίνακα 5.9.

Υπερπαραμέτροι	Εναλλακτικές	Βέλτιστες
batch size	16, 32, 64	64
epochs	25, 50	50
optimizer	'Adam', 'SGD'	'Adam'
loss	'binary crossentropy', 'categorical crossentropy', 'poisson'	'categorical crossentropy'
activation	'ReLU', 'tanh'	'ReLU'
kernel size	3, 5	3, 5
filters	32, 64, 128	32
dropout rate	0.2, 0.5, 0.8	0.5
rnn units	64, 128, 256	128

Πίνακας 5.9: Πίνακας υπερπαραμέτρων CNN+RNN αρχιτεκτονικής

Για τις βέλτιστες τιμές των υπερπαραμέτρων σημειώθηκε ακρίβεια στις ταξινομήσεις ίση με 92,2%. Παρατηρούμε ότι η υβριδική αρχιτεκτονική υπερισχύει σε επιδόσεις της απλής RNN αρχιτεκτονικής η οποία είχε σκόρ ακρίβειας μόλις 80,83%. Προκειμένου να χρησιμοποιήσουμε την υβριδική αρχιτεκτονική, χρειάστηκε να τροποποιήσουμε τα δεδομένα εισόδου μας ανά δείγμα, από 128 χρονικά βήματα σε 4 υποακολουθίες των 32 χρονικών βημάτων. Αυτό φαίνεται να έπαιξε καθοριστικό ρόλο στην επίδοση του μοντέλου, καθώς πλέον το RNN δεν υπέφερε από βραχυπρόθεσμη μνήμη αφού τα χρονικά βήματα που χρειάζονταν να επεξεργαστεί ήταν μειωμένα.

Στον πίνακα 5.10 φαίνονται αναλυτικά οι ταξινομήσεις που έκανε το μοντέλο μας. Για όλες τις δραστηριότητες παρατηρούμε αντίστοιχα αποτελέσματα όπως και με τις 3 προηγούμενες αρχιτεκτονικές. Το μόνο που αξίζει να σημειώσουμε είναι ότι το μοντέλο μας συγκριτικά με τα προηγούμενα σημειώνει περισσότερες αστοχίες στην ταξινόμηση των στατικών δραστηριοτήτων καθώς τις ταξινομεί ως 'Περπάτημα προς τα επάνω'. Μια αντίστοιχη αδυναμία είχε φανερώσει και η αρχιτεκτονική TCN.

5.3.6 Εφαρμογή CNN + GRU αρχιτεκτονικής

Η δεύτερη μέθοδος ταξινόμησης που εφαρμόστηκε για τα υβριδικά μοντέλα ήταν μέσω των νευρωνικών δικτύων CNN(s) + GRU(s). Για την εκπαίδευσή τους χρησιμοποιήθηκαν τα

1	2	3	4	5	6
468	2	0	0	0	0
2	445	5	9	1	27
26	23	417	0	0	0
0	0	0	417	93	0
0	0	0	72	439	0
0	1	0	0	0	537

Πίνακας 5.10: Πίνακας σύγκρισης CNN + RNN αρχιτεκτονικής

δεδομένα εκπαίδευσης (training data) και τα δεδομένα ελέγχου (test data) όπως ακριβώς και προηγουμένως.

Προκειμένου να πετύχουμε τα βέλτιστα αποτελέσματα για την εκπαίδευση της αρχιτεκτονικής μας χρειάστηκε να ελέγξουμε αρκετές υπερπαραμέτρους, τόσο καθοριστικές για την εκπαίδευση του μοντέλου μας όσο και για καθένα επίπεδο. Οι βασικές υπερπαραμέτρους που σχετίζονται με την διαδικασία εκπαίδευσης είναι οι ίδιες και με τα προηγούμενα πειράματα. Για το GRU επίπεδο μελετήσαμε τις ίδιες παραμέτρους με την απλή GRU αρχιτεκτονική, δηλαδή τον αριθμό των μονάδων GRU (units). Ακόμη, ελέγχθηκε ο αριθμός των φίλτρων (filters) που χρησιμοποιήθηκε από τα επίπεδα των CNNs καθώς και το μέγεθός του πυρήνα (kernel size). Τέλος, πειραματιστήκαμε και με τον ρυθμό εγκατάλειψης (dropout rate) στο επίπεδο εγκατάλειψης καθώς και τις συναρτήσεις ενεργοποίησης σε όλα τα επίπεδα της αρχιτεκτονικής μας. Αναλυτικά οι τιμές που δοκιμαστήκαν καθώς και οι βέλτιστες τιμές φαίνονται στον πίνακα 5.11.

Υπερπαραμέτροι	Εναλλακτικές	Βέλτιστες
batch size	16, 32, 64	64
epochs	25, 50	50
optimizer	'Adam', 'SGD'	'Adam'
loss	'binary crossentropy', 'categorical crossentropy', 'poisson'	'categorical crossentropy'
activation	'ReLU', 'tanh'	'ReLU'
kernel size	3, 5	3, 5
filters	32, 64, 128	32
dropout rate	0.2, 0.5, 0.8	0.2
gru units	64, 128, 256	128

Πίνακας 5.11: Πίνακας υπερπαραμέτρων CNN+GRU αρχιτεκτονικής

Με τις βέλτιστες παραμέτρους το μοντέλο μας σημείωσε σκορ ακρίβειας 93,28%. Το σκόρ που σημειώθηκε είναι κοντινό με αυτό του προηγούμενου υβριδικού μοντέλου. Από αυτή την παρατήρηση θα μπορούσαμε να συμπεράνουμε ότι η αλλαγή στον τρόπο που δίνουμε τα δεδομένα στο μοντέλο μας και η μείωση πρακτικά των χρονικών βημάτων που επεξεργάζεται το εκάστοτε αναδρομικό νευρωνικό δίκτυο, οδηγεί σε υψηλή ακρίβεια ταξινόμησης, ανεξαρτήτου αν το αναδρομικό δίκτυο θα είναι RNN ή όχι.

Στον πίνακα σύγκρισης 5.12 παρουσιάζονται αναλυτικά οι ταξινομήσεις που έγιναν από

το μοντέλο μας. Συγκριτικά με την αρχιτεκτονική με το απλό GRU δίκτυο 5.4 παρατηρούμε αντίστοιχα αποτελέσματα, με κάποιες δραστηριότητες να ταξινομούνται καλύτερα από το υβριδικό μοντέλο, όπως η 1, και με άλλες καλύτερα από το απλό, όπως η 2. Σε κάθε περίπτωση τα συνολικά αποτελέσματα είναι παρόμοια.

1	2	3	4	5	6
478	13	2	0	1	0
0	435	0	3	0	0
18	23	418	1	0	0
0	0	0	383	33	0
0	0	0	99	498	0
0	0	0	5	0	537

Πίνακας 5.12: Πίνακας σύγκρισης CNN + GRU αρχιτεκτονικής

5.3.7 Εφαρμογή CNN+LSTM αρχιτεκτονικής

Η τρίτη μέθοδος ταξινόμησης που εφαρμόστηκε για τα υβριδικά μοντέλα ήταν μέσω των νευρωνικών δικτύων CNN(s) + LSTM(s). Για την εκπαίδευσή τους χρησιμοποιήθηκαν τα δεδομένα εκπαίδευσης (training data) και τα δεδομένα ελέγχου (test data) όπως ακριβώς και προηγουμένως.

Για την βέλτιστη εφαρμογή της παρούσας αρχιτεκτονικής θα χρειαστεί πάλι να ελέγξουμε ορισμένες υπερπαραμέτρους. Οι υπερπαραμέτροι που σχετίζονται με την διαδικασία της εκπαίδευσης είναι οι ίδιες με όλα τα προηγούμενα πειράματα. Οι παράμετροι που σχετίζονται με το LSTM επίπεδο είναι οι ίδιες, δηλαδή ο αριθμός των αναδρομικών μονάδων που χρησιμοποιούνται (units). Επιπλέον, ελέγχθηκε ο αριθμός των φίλτρων(filters) που χρησιμοποιήθηκαν από τα επίπεδα των CNNs καθώς και το μέγεθος του πυρήνα (kernel size). Τέλος, πειραματιστήκαμε και με τον ρυθμό εγκατάλειψης (dropout rate) στο επίπεδο εγκατάλειψης καθώς και τις συναρτήσεις ενεργοποίησης σε όλα τα επίπεδα της αρχιτεκτονικής μας. Αναλυτικά οι τιμές που δοκιμαστήκαν καθώς και οι βέλτιστες τιμές φαίνονται στον πίνακα 5.13

Υπερπαραμέτροι	Εναλλακτικές	Βέλτιστες
batch size	16, 32, 64	64
epochs	25, 50	50
optimizer	'Adam', 'SGD'	'Adam'
loss	'binary crossentropy', 'categorical crossentropy', 'poisson'	'poisson'
activatin	'ReLU', 'tanh'	'ReLU'
kernel size	3, 5	3
filters	32, 64, 128	128
dropout rate	0.2, 0.5, 0.8	0.2
units	64, 128, 256	128

Πίνακας 5.13: Πίνακας υπερπαραμέτρων CNN+LSTM αρχιτεκτονικής

Χρησιμοποιώντας τις βέλτιστες τιμές των υπερπαραμέτρων το μοντέλο μας σημειώνει σκορ ακρίβειας ίσο με 92,97%. Και για αυτό το μοντέλο η συγκεκριμένη μετρική κυμαίνεται στα ίδια επίπεδα. Σημειώνουμε ότι τόσο στο παρόν μοντέλο, όσο και στο προηγούμενο (CNN + GRU) ο ρυθμός εγκατάλειψης είναι 0.2, ενώ στα περισσότερα πειράματα ήταν 0.5. Αυτό δηλώνει την ανάγκη του υβριδικού δικτύου να μεταφέρει μεταξύ των στρωμάτων ακόμα περισσότερες πληροφορίες, συγκριτικά με τα απλά μοντέλα.

Στον πίνακα σύγχυσης 5.14 φαίνονται αναλυτικά οι ταξινομήσεις που έγιναν από το μοντέλο μας. Οι περισσότερες αστοχίες του εντοπίζονται στα ίδια σημεία με όλα τα μοντέλα, δηλαδή μικρή αδυναμία σωστής ταξινόμησης των δραστηριοτήτων 1 και 2 καθώς ταξινομούνται ως δραστηριότητα 3, καθώς και μια σύγχυση μεταξύ των δραστηριοτήτων 4 και 5.

1	2	3	4	5	6
469	6	3	0	0	0
2	441	0	3	0	0
25	24	417	0	0	0
0	0	0	388	44	0
0	0	0	94	488	0
0	0	0	6	0	537

Πίνακας 5.14: Πίνακας σύγχυσης CNN + LSTM αρχιτεκτονικής

5.3.8 Εφαρμογή CNN+TCN αρχιτεκτονικής

Η τελευταία μέθοδος ταξινόμησης που εφαρμόστηκε στα πλαίσια της παρούσας διπλωματικής εργασίας ήταν μέσω των νευρωνικών δικτύων CNN(s) + TCN. Για την εκπαίδευσή τους χρησιμοποιήθηκαν τα δεδομένα εκπαίδευσης (training data) και τα δεδομένα ελέγχου (test data) όπως ακριβώς και προηγουμένως.

Προκειμένου να εκπαιδύσουμε το μοντέλο μας να ταξινομή τις 6 δραστηριότητες, χρειάστηκε να εξετάσουμε αρκετές από τις υπερπαραμέτρους τόσο των νευρωνικών επιπέδων της αρχιτεκτονικής μας, όσο και της διαδικασίας εκπαίδευσης. Για το δεύτερο οι υπερπαραμέτροι που μελετήθηκαν είναι ο αριθμός των εποχών (epochs), το μέγεθος παρτίδας (batch size), η συνάρτηση σφάλματος (loss) και ο βελτιστοποιητής (optimizer). Για το επίπεδο εγκατάλειψης μελετήθηκε η υπερπαραμέτρος του ρυθμού εγκατάλειψης (dropout rate). Επιπλέον, μελετήθηκαν δύο συναρτήσεις ενεργοποίησης σε όλα τα επίπεδα της αρχιτεκτονικής μας, όπου απαιτούταν. Ακόμη, ελέγχθηκε ο αριθμός των φίλτρων (filters) που χρησιμοποιήθηκε από τα επίπεδα των CNNs καθώς και το μέγεθος του πυρήνα (kernel size). Τέλος, για το TCN επίπεδο μελετήσαμε την παράμετρο use skip connectios, η οποία καθορίζει αν θα έχουμε σύνδεση μεταξύ της εισόδου ενός υπολειμματικού μπλοκ με την έξοδο του και το μέγεθος πυρήνα που χρησιμοποιούν τα επίπεδα των μονοδιάστατων CNNs. Ακόμη, ελέγχθηκε ο αριθμός των φίλτρων (nb size) που χρησιμοποιήθηκε από τα επίπεδα των CNNs (η αντίστοιχη παράμετρος στα LSTMs είναι η unit). Άλλη μια παράμετρος είναι η παράμετρος του γεμίσιματος (padding) που καθορίζει αν θα είναι το γέμισμα αιτιώδης όπως στην θεωρητική υλοποίηση ή όχι. Αναλυτικά οι τιμές που δοκιμαστήκαν καθώς και οι βέλτιστες τιμές φαίνονται στον πίνακα 5.15.

Υπερπαράμετροι	Εναλλακτικές	Βέλτιστες
batch size	16, 32, 64	64
epochs	25, 50	25
optimizer	'Adam', 'SGD'	'Adam'
loss	'binary crossentropy', 'categorical crossentropy', 'poisson'	'categorical crossentropy'
activation	'ReLU', 'tanh'	'tanh'
kernel size (TCN)	3, 5	3
use skip connections	True, False	True
padding	causal, same	same
nb size	64, 128	64
dropout rate	0.2, 0.5, 0.8	0.5
kernel size (CNN)	3, 5	3
filters (CNN)	32, 64, 128	64

Πίνακας 5.15: Πίνακας υπερπαραμέτρων CNN + TCN αρχιτεκτονικής

Θέτοντας τις τιμές των παραμέτρων ίσες με τις βέλτιστες τιμές που προέκυψαν, πετυχαίνουμε ακρίβεια ταξινομήσεων ίση με 92.3%. Αυτή η επίδοση είναι κοντινή με την αντίστοιχη του απλού TCN νευρωνικού δικτύου αλλά υπολείπεται ελαφρώς αυτής. Η εφαρμογή των μονοδιάστατων CNNs για ανάλυση χρονοσειρών αποτελεί την βάση και για την εσωτερική αρχιτεκτονική των TCN νευρωνικών δικτύων, επομένως είναι λογικό να μην υπάρχουν μεγάλες αποκλίσεις στην συμπεριφορά αυτών των δικτύων. Επιπλέον, σημειώνουμε ότι και οι 4 υβριδικές αρχιτεκτονικές πετυχαίνουν πολύ κοντινές επιδόσεις.

Στον πίνακα 5.16 καταγράφονται αναλυτικά όλες οι ταξινομήσεις που έγιναν, για όλες τις δραστηριότητες. Για μια ακόμη φορά, η εικόνα των ταξινομήσεων είναι αρκετά παρόμοια. Για τις εν κινήσει δραστηριότητες παρατηρούμε ότι τόσο η δραστηριότητα 1 (περπάτημα), όσο και η 2 (περπάτημα προς τα επάνω), ταξινομούνται ορισμένες φορές ως 3 (περπάτημα προς τα κάτω), πράγμα το οποίο έχει παρατηρηθεί σε όλα τα μοντέλα μας. Επίσης, σημειώνεται μια μικρή αδυναμία στον καθορισμό μεταξύ των δραστηριοτήτων 4 και 5. Τέλος, σημειώνουμε ότι το υβριδικό μοντέλο με TCN πετυχαίνει καλύτερη ταξινόμηση μεταξύ των στατικών και μη δραστηριοτήτων, σε σύγκριση με το απλό μοντέλο.

1	2	3	4	5	6
466	2	0	0	0	0
2	442	0	1	1	0
28	27	418	0	0	0
0	0	1	417	91	0
0	0	0	67	440	0
0	0	0	6	0	537

Πίνακας 5.16: Πίνακας σύγχυσης CNN + TCN αρχιτεκτονικής

Μέρος **III**

Επίλογος

Επίλογος

6.1 Σύνοψη και συμπεράσματα

Στόχο της παρούσας διπλωματικής εργασίας αποτελεί η ανάλυση χρονικών ακολουθιών για την ταξινόμηση των ανθρώπινων δραστηριοτήτων.

Για την επίλυση αυτού του προβλήματος χρησιμοποιήθηκε το UCI HAR σύνολο δεδομένων το οποίο αποτελείται από 7352 δείγματα στο σύνολο εκπαίδευσης του και από 2947 στο σύνολο επαλήθευσης. Αρχικά ελέγχθηκε η ισορροπία και των δύο συνόλων, η οποία λόγω του τρόπου δημιουργίας των δεδομένων υπήρχε. Κατόπιν, από τα χρονικά παράθυρα των δειγμάτων σχεδιάσαμε γραφικές παραστάσεις για κάθε εθελοντή που εκτέλεσε τις δραστηριότητες. Με αυτό τον τρόπο εντοπίστηκαν ορισμένα ποιοτικά χαρακτηριστικά των δραστηριοτήτων καθώς και ορισμένες ομοιότητες και διαφορές μεταξύ τους. Στην συνέχεια της ανάλυσης μας δημιουργήσαμε ιστογράμματα για κάθε αισθητήρα με σκοπό να εντοπίσουμε τις κατανομές των δεδομένων για όλες τις δραστηριότητες ανά εθελοντή, προκειμένου να ελέγξουμε την ομοιομορφία και την συνέπεια του συνόλου δεδομένων μας. Ιστογράμματα έγιναν και ανά δραστηριότητα ανά εθελοντή καθώς το πρόβλημα που ασχοληθήκαμε είναι πρόβλημα ταξινόμησης και για την μοντελοποίηση του είναι απαραίτητο να υπάρχει διάκριση μεταξύ των δραστηριοτήτων. Τέλος, ελέγχθηκε η διάρκεια της κάθε δραστηριότητας μέσω Boxplot. Τα συμπεράσματα της ανάλυσης του συνόλου δεδομένων ήταν αρκετά ενθαρρυντικά για την μοντελοποίηση του καθώς δεν φάνηκαν να υπήρχε ούτε ανισορροπία μεταξύ των δραστηριοτήτων, ούτε ανομοιομορφία μεταξύ των κατανομών στα ιστογράμματα. Επιπλέον, φάνηκε ότι κάθε χαρακτηριστικό κάθε αισθητήρα προσδιόριζε διαφορετικά χαρακτηριστικά τα οποία συμβάλουν στο να καθοριστεί μοναδικά κάθε δραστηριότητα. Επομένως, το σύνολο δεδομένων μας φάνηκε αρκετά πλήρες χωρίς περιττές πληροφορίες ή ακραίες τιμές.

Μετά την ανάλυση δεδομένων, ακολούθησε το πειραματικό κομμάτι της διπλωματικής. Αρχικά περιγράφηκαν οι βασικές αρχιτεκτονικές που χρησιμοποιήθηκαν για την ταξινόμηση των δραστηριοτήτων, η αρχιτεκτονική των αναδρομικών δικτύων και η υβριδική αρχιτεκτονική. Κατόπιν, δοκιμάστηκαν τα νευρωνικά δίκτυα RNN, GRU, LSTM μέσω της πρώτης αρχιτεκτονικής καθώς και τα χρονικά συνελκτικά νευρωνικά δίκτυα (TCN). Μέσω της δεύτερης αρχιτεκτονικής έγινε πάντρεμα όλων των παραπάνω νευρωνικών με μονοδιάστατα CNNs. Όλα τους είχαν την δυνατότητα ταξινόμησης υψηλής ακρίβειας με το πιο αποδοτικό να είναι το νευρωνικό δίκτυο TCN, ωστόσο χωρίς να διαφοροποιείται σημαντικά από τα υπόλοιπα μοντέλα. Ευδιάκριτη διαφοροποίηση παρατηρήθηκε στην ακρίβεια των ταξινομήσεων μόνο

με το RNN νευρωνικό δίκτυο που φάνηκε να υπολείπεται των υπολοίπων. Αυτό οφειλόταν στην αδυναμία των RNNs να αποθηκεύουν στην μνήμη τους πληροφορίες από μεγάλες σε μήκος χρονοσειρές.

6.2 Μελλοντικές Επεκτάσεις

Το σύστημα που αναπτύχθηκε στο πλαίσιο αυτής της διπλωματικής εργασίας θα μπορούσε να βελτιωθεί και να επεκταθεί περαιτέρω, τουλάχιστον ως προς δύο κατευθύνσεις. Συγκεκριμένα, αναφέρονται τα ακόλουθα :

- Επιπλέον προετοιμασίας των δεδομένων. Μερικά παραδείγματα δοκιμών που πιθανόν να βοηθήσουν στην μοντελοποίηση του συνόλου δεδομένων μπορεί να είναι: κανονικοποίηση (normalization) και προτυποποίηση (standardization) των δεδομένων ανά και κατά εθελοντή. Ακόμη, μπορεί να δοκιμαστεί και κάποιος αλγόριθμός επιλογής χαρακτηριστικών
- Εξερεύνηση άλλων νευρωνικών δικτύων που επεξεργάζονται χρονοσειρές. Τέτοια δίκτυα μπορεί να είναι το WaveNet και το Prophet.

Τέλος, η εργασία της ταξινόμησης των ανθρώπινων δραστηριοτήτων μπορεί να βρει εφαρμογή σε πολλές χρήσιμες εφαρμογές. Ένα ενδιαφέρον παράδειγμα μπορεί να είναι σε οίκους φροντίδας ηλικιωμένων όπου σε περίπτωση πτώσης ηλικιωμένου να μπορεί να σταλεί σήμα βοήθειας. Ακόμη, η ταξινόμηση θα μπορούσε να είναι και σε στατικές και μη δραστηριότητες καθώς και σε δραστηριότητες που χρειάζονται να είσαι όρθιος ή καθιστός.

Αξίζει να γίνει αναφορά ότι τα αποτελέσματα της ταξινόμησης των ανθρώπινων δραστηριοτήτων μπορούν να παίξουν σημαντικό ρόλο από την δημιουργία στατιστικών ανά ηλικιακή ομάδα ή γεωγραφική περιοχή που να δείχνουν πόσο έντονοι είναι οι ρυθμοί ζωής της καθημερινότητας, μέχρι και στην εξιχνίαση εγκλημάτων, ελέγχοντας αν η περιγραφή του δράστη είναι σύμφωνη με τις δραστηριότητες που έχει καταγραφεί να κάνει από κάποια smart συσκευή που είχε πάνω του.

Βιβλιογραφία

- [1] *Trend in times series analysis* - ORAYLIS. = <https://www.oraylis.de/blog/2015/trend-in-times-series-analysis>. Ημερομηνία πρόσβασης: 2021-09-18.
- [2] *time series - What is the difference between period cycle and seasonality? - Cross Validated*. <https://stats.stackexchange.com/questions/234492/what-is-the-difference-between-period-cycle-and-seasonality/234601>. Ημερομηνία πρόσβασης: 2021-09-18.
- [3] *Time Series in Python – Part 2: Dealing with seasonal data* | by Benjamin Etienne | Towards Data Science. <https://towardsdatascience.com/time-series-in-python-part-2-dealing-with-seasonal-data-397a65b74051>. Ημερομηνία πρόσβασης: 2021-09-18.
- [4] *Example of the interventions detected in the individual BAI time-series... | Download Scientific Diagram*. https://www.researchgate.net/figure/Example-of-the-interventions-detected-in-the-individual-BAI-time-series-and-accounted-for_{_}fig2_{_}283310582. Ημερομηνία πρόσβασης: 2021-09-18.
- [5] *Illustrated Guide to Recurrent Neural Networks* | by Michael Phi | Towards Data Science. <https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9>. Ημερομηνία πρόσβασης: 2021-06-27.
- [6] *8.3. Language Models and the Dataset – Dive into Deep Learning 0.16.6 documentation*. https://d2l.ai/chapter_{_}recurrent-neural-networks/language-models-and-dataset.html. Ημερομηνία πρόσβασης: 2021-06-27.
- [7] *8.7. Backpropagation Through Time – Dive into Deep Learning 0.16.6 documentation*. https://d2l.ai/chapter_{_}recurrent-neural-networks/bptt.html. Ημερομηνία πρόσβασης: 2021-06-28.
- [8] *Understanding GRU Networks. In this article, I will try to give a . . .* | by Simeon Kostadinov | Towards Data Science. <https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be>. Ημερομηνία πρόσβασης: 2021-06-15.
- [9] *What is the sigmoid function, and what is its use in machine learning’s neural networks?* <https://www.luisotsm.com/single-post/what-is-the-sigmoid-function-and-what-is-its-use-in-machine-learning-s-neural-networks>. Ημερομηνία πρόσβασης: 2021-09-28.
- [10] *Understanding LSTM Networks - colah’s blog*. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Ημερομηνία πρόσβασης: 2021-06-03.

- [11] 6. Convolutional Neural Networks – Dive into Deep Learning 0.17.0 documentation. <https://d2l.ai/chapter{ }convolutional-neural-networks/index.html>. Ημερομηνία πρόσβασης: 2021-08-09.
- [12] Max-pooling / Pooling - Computer Science Wiki. <https://computersciencewiki.org/index.php/Max-pooling{ }/{ }Pooling>. Ημερομηνία πρόσβασης: 2021-08-11.
- [13] 6.3. Padding and Stride – Dive into Deep Learning 0.17.0 documentation. <https://d2l.ai/chapter{ }convolutional-neural-networks/padding-and-strides.html>. Ημερομηνία πρόσβασης: 2021-08-19.
- [14] Temporal Convolutional Networks and Forecasting | Unit8 Blog. <https://unit8.co/resources/temporal-convolutional-networks-and-forecasting/>. Ημερομηνία πρόσβασης: 2021-08-11.
- [15] 4.6. Dropout – Dive into Deep Learning 0.17.0 documentation. <https://d2l.ai/chapter{ }multilayer-perceptrons/dropout.html{#}dropout-in-practice>. Ημερομηνία πρόσβασης: 2021-09-01.
- [16] Temporal Convolutional Networks and Forecasting | Unit8 Blog. <https://unit8.co/resources/temporal-convolutional-networks-and-forecasting/>. Ημερομηνία πρόσβασης: 2021-08-21.
- [17] SciPy.org – SciPy.org. <https://numpy.org/>. Ημερομηνία πρόσβασης: 2021-08-25.
- [18] Understanding Boxplots. The image above is a boxplot. A boxplot... | by Michael Galarnyk | Towards Data Science. <https://towardsdatascience.com/understanding-boxplots-5e2df7bcdb51>. Ημερομηνία πρόσβασης: 2021-08-29.
- [19] 4.6. Dropout – Dive into Deep Learning 0.17.0 documentation. <https://d2l.ai/chapter{ }multilayer-perceptrons/dropout.html{#}dropout-in-practice>. Ημερομηνία πρόσβασης: 2021-09-01.
- [20] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra και Jorge L. Reyes-Ortiz. *Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7657 LNCS:216-223, 2012.
- [21] Ronald Mutegeki και Dong Seog Han. *A CNN-LSTM Approach to Human Activity Recognition. 2020 International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2020*, σελίδες 362-366, 2020.
- [22] Van Der Vaart: Time Series | PDF | Stationary Process | Time Series. <https://www.scribd.com/document/190544157/van-der-Vaart-Time-Series>. Ημερομηνία πρόσβασης: 2021-08-25.

- [23] ΠΣΣ ΠΛΨΣ / ΠΠΥΛΣ ΦΩΣ , ΣΠΥΛΣ ΣΛΣ. <https://www.politeianet.gr/books/9789602663332-petropoulos-fotios-summetria-epicheirisiakes-problepseis-222491>. Ημερομηνία πρόσβασης: 2021-09-12.
- [24] Anargyros Chatzitofis, Leonidas Saroglou, Prodromos Boutis, Petros Drakoulis, Nikolaos Zioulis, Shishir Subramanyam, Bart Kevelham, Caecilia Charbonnier, Pablo Cesar, Dimitrios Zarpalas και others. *HUMAN4D: A human-centric multimodal dataset for motions and immersive media*. *IEEE Access*, 8:176241–176262, 2020.
- [25] Georgios Tsatiris, Kostas Karpouzis και Stefanos Kollias. *A compact sequence encoding scheme for online human activity recognition in HRI applications*. *International Conference on Engineering Applications of Neural Networks*, σελίδες 3–14. Springer, 2020.
- [26] Anargyros Chatzitofis, Dimitrios Zarpalas, Stefanos Kollias και Petros Daras. *DeepMoCap: Deep optical motion capture using multiple depth sensors and retro-reflectors*. *Sensors*, 19(2):282, 2019.
- [27] Konstantinos Rapantzikos, Yannis Avrithis και Stefanos Kollias. *Spatiotemporal saliency for event detection and representation in the 3D wavelet domain: potential in human action recognition*. *Proceedings of the 6th ACM international conference on Image and video retrieval*, σελίδες 294–301, 2007.
- [28] Konstantinos Rapantzikos, Nicolas Tsapatsoulis, Yannis Avrithis και Stefanos Kollias. *Spatiotemporal saliency for video classification*. *Signal Processing: Image Communication*, 24(7):557–571, 2009.
- [29] Changkyu Choi. *Time Series Forecasting with Recurrent Neural Networks in Presence of Missing Data*. Τεχνική Αναφορά με αριθμό, RNN2, 2018.
- [30] Alex Graves, Abdel Rahman Mohamed και Geoffrey Hinton. *SPEECH RECOGNITION WITH DEEP RECURRENT NEURAL NETWORKS*. Τεχνική Αναφορά με αριθμό.
- [31] Paul J. Werbos. *Backpropagation Through Time: What It Does and How to Do It*. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.
- [32] Robert Hecht-Nielsen. *Theory of the Backpropagation Neural Network*. Τεχνική Αναφορά με αριθμό.
- [33] Razvan Pascanu, Tomas Mikolov και Yoshua Bengio. *On the difficulty of training recurrent neural networks*. Τεχνική Αναφορά με αριθμό, 2013.
- [34] *The Vanishing/Exploding Gradient Problem in Deep Neural Networks* | by Kurtis Pykes | *Towards Data Science*. <https://towardsdatascience.com/the-vanishing-exploding-gradient-problem-in-deep-neural-networks-191358470c11>. Ημερομηνία πρόσβασης: 2021-06-29.

- [35] 9.1. *Gated Recurrent Units (GRU) – Dive into Deep Learning 0.16.7 documentation*. https://d2l.ai/chapter_{_}recurrent-modern/gru.html. Ημερομηνία πρόσβασης: 2021-07-24.
- [36] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk και Yoshua Bengio. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*.
- [37] M. Pavithra, K. Saruladha και K. Sathyabama. *GRU based deep learning model for prognosis prediction of disease progression. Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019*, σελίδες 840–844, 2019.
- [38] Khaled A. Althelaya, El Sayed M. El-Alfy και Salahadin Mohammed. *Stock Market Forecast Using Multivariate Analysis with Bidirectional and Stacked (LSTM, GRU). 21st Saudi Computer Society National Computer Conference, NCC 2018*, 2018.
- [39] Sepp Hochreiter και Jürgen Schmidhuber. *Long Short-Term Memory. Neural Computation*, 9(8):1735–1780, 1997.
- [40] 9.2. *Long Short-Term Memory (LSTM) – Dive into Deep Learning 0.17.0 documentation*. https://d2l.ai/chapter_{_}recurrent-modern/lstm.html. Ημερομηνία πρόσβασης: 2021-08-02.
- [41] Kaisheng Yao, Trevor Cohn, Katerina Vylomova, Kevin Duh και Chris Dyer. *Depth-Gated Recurrent Neural Networks*.
- [42] Felix A. Gers, Jürgen Schmidhuber και Fred Cummins. *Learning to Forget: Continual Prediction with LSTM. Neural Computation*, 12(10):2451–2471, 2000.
- [43] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink και Jürgen Schmidhuber. *TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS 1 LSTM: A Search Space Odyssey*.
- [44] Rafal Jozefowicz και Wojciech Zaremba. *An Empirical Exploration of Recurrent Network Architectures*.
- [45] Tao Lin, Tian Guo και Karl Aberer. *Hybrid neural networks for learning the trend in time series. IJCAI International Joint Conference on Artificial Intelligence*, 0:2273–2279, 2017.
- [46] *A Comprehensive Guide to Convolutional Neural Networks – the ELI5 way | by Sumit Saha | Towards Data Science*. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. Ημερομηνία πρόσβασης: 2021-08-08.
- [47] Miao Yu, Dimitrios Kollias, James Wingate, Niro Siriwardena και Stefanos Kollias. *Machine learning for predictive modelling of ambulance calls. Electronics*, 10(4):482, 2021.

- [48] Dimitrios Kollias, Anastasios Arsenos, Levon Soukissian και Stefanos Kollias. *MIA-COV19D: COVID-19 Detection through 3-D Chest CT Image Analysis*. *arXiv preprint arXiv:2106.07524*, 2021.
- [49] Dimitrios Kollias και Stefanos P Zafeiriou. *Exploiting multi-cnn features in cnn-rnn based dimensional emotion recognition on the omg in-the-wild dataset*. *IEEE Transactions on Affective Computing*, 2020.
- [50] Fabio De Sousa Ribeiro, Francesco Calivá, Mark Swainson, Kjartan Gudmundsson, Georgios Leontidis και Stefanos Kollias. *Deep bayesian self-training*. *Neural Computing and Applications*, σελίδες 1–17, 2019.
- [51] Dimitrios Kollias, Athanasios Tagaris, Andreas Stafylopatis, Stefanos Kollias και Georgios Tagaris. *Deep neural architectures for prediction in healthcare*. *Complex & Intelligent Systems*, 4(2):119–131, 2018.
- [52] James Wingate, Ilianna Kollia, Luc Bidaut και Stefanos Kollias. *Unified deep learning approach for prediction of Parkinson’s disease*. *IET Image Processing*, 14(10):1980–1989, 2020.
- [53] Dimitris Kollias, Y Vlaxos, M Seferis, Ilianna Kollia, Levon Sukissian, James Wingate και Stefanos D Kollias. *Transparent Adaptation in Deep Medical Image Diagnosis*. *TAILOR*, σελίδες 251–267, 2020.
- [54] Dimitrios Kollias, N Bouas, Y Vlaxos, V Brillakis, M Seferis, Ilianna Kollia, Levon Sukissian, James Wingate και S Kollias. *Deep transparent prediction through latent representation analysis*. *arXiv preprint arXiv:2009.07044*, 2020.
- [55] Aiden Durrant, Georgios Leontidis και Stefanos Kollias. *3D convolutional and recurrent neural networks for reactor perturbation unfolding and anomaly detection*. *EPJ Nuclear Sciences & Technologies*, 2019.
- [56] Bashar Alhnaity, Stefanos Kollias, Georgios Leontidis, Shouyong Jiang, Bert Champ και Simon Pearson. *An autoencoder wavelet based deep neural network with attention mechanism for multi-step prediction of plant growth*. *Information Sciences*, 560:35–50, 2021.
- [57] 6. *Convolutional Neural Networks – Dive into Deep Learning 0.17.0 documentation*. <https://d2l.ai/chapter{-}convolutional-neural-networks/index.html>. Ημερομηνία πρόσβασης: 2021-08-09.
- [58] *CNN Long Short-Term Memory Networks*. <https://machinelearningmastery.com/cnn-long-short-term-memory-networks/>. Ημερομηνία πρόσβασης: 2021-08-19.
- [59] *CNN Long Short-Term Memory Networks*. <https://machinelearningmastery.com/cnn-long-short-term-memory-networks/>. Ημερομηνία πρόσβασης: 2021-08-19.

- [60] Tao Lin, Tian Guo και Karl Aberer. *Hybrid neural networks for learning the trend in time series*. *IJCAI International Joint Conference on Artificial Intelligence*, 0:2273–2279, 2017.
- [61] Saeed Khaki, Lizhi Wang και Sotirios V. Archontoulis. *A CNN-RNN Framework for Crop Yield Prediction*. *Frontiers in Plant Science*, 0:1750, 2020.
- [62] Shaojie Bai, J Zico Kolter και Vladlen Koltun. *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*.
- [63] Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano και Kevin J. Lang. *Phoneme Recognition Using Time-Delay Neural Networks*. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):328–339, 1989.
- [64] *Matplotlib: Python plotting – Matplotlib 3.4.3 documentation*. <https://matplotlib.org/>. Ημερομηνία πρόσβασης: 2021-08-25.
- [65] *NumPy*. <https://numpy.org/>. Ημερομηνία πρόσβασης: 2021-08-25.
- [66] *SciPy.org – SciPy.org*. <https://www.scipy.org/>. Ημερομηνία πρόσβασης: 2021-08-25.
- [67] Dean M. Karantonis, Michael R. Narayanan, Merryn Mathie, Nigel H. Lovell και Branko G. Celler. *Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring*. *IEEE Transactions on Information Technology in Biomedicine*, 10(1):156–167, 2006.
- [68] Chiraz BenAbdelkader, Ross Cutler και Larry Davis. *Stride and cadence as a biometric in automatic person identification and verification*. *Proceedings - 5th IEEE International Conference on Automatic Face Gesture Recognition, FGR 2002*, σελίδες 372–377, 2002.
- [69] Jhun Ying Yang, Jeen Shing Wang και Yen Ping Chen. *Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers*. *Pattern Recognition Letters*, 29(16):2213–2220, 2008.
- [70] A M Khan, Y. K Lee, S Y Lee και T. S Kim. *Human Activity Recognition via An Accelerometer-Enabled-Smartphone Using Kernel Discriminant Analysis*.
- [71] *UCI Machine Learning Repository: Human Activity Recognition Using Smartphones Data Set*. <https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones>. Ημερομηνία πρόσβασης: 2021-08-28.
- [72] *Python Histogram Plotting: NumPy, Matplotlib, Pandas & Seaborn - Real Python*. <https://realpython.com/python-histograms/>. Ημερομηνία πρόσβασης: 2021-08-29.