



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

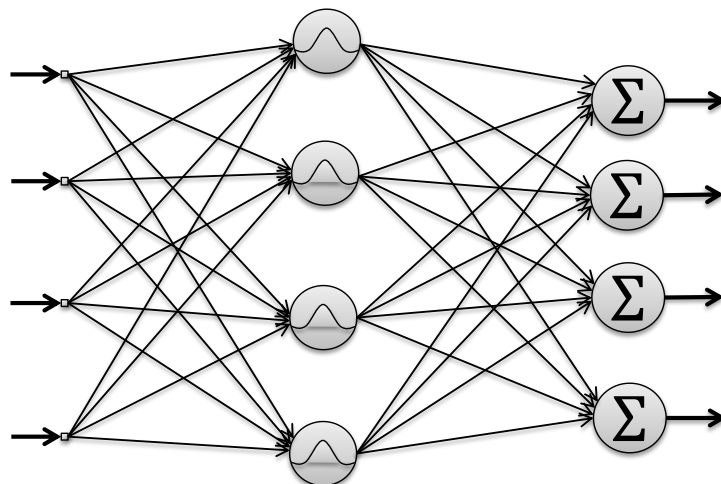
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Πρόγραμμα Μεταπτυχιακών Σπουδών

«ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΠΡΑΓΜΑΤΩΝ ΚΑΙ ΕΥΦΥΗ ΠΕΡΙΒΑΛΛΟΝΤΑ»

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη online μεθόδου εκπαίδευσης νευρωνικών δικτύων
ακτινικής συνάρτησης βάσης



Μεταπτυχιακός Φοιτητής: Εμμανουήλ Δεμενεόπουλος
ΑΜ: 20003

Επιβλέπων: Αλέξανδρος Αλεξανδρίδης, Καθηγητής

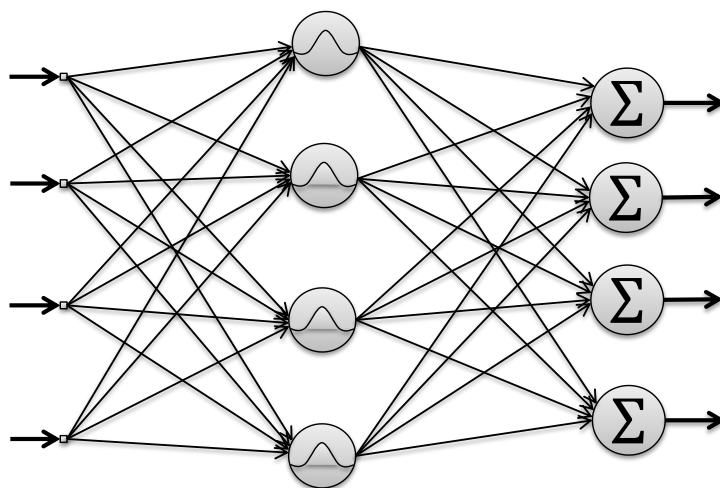
Αιγάλεω - Αθήνα, Ιούλιος, 2023



Master of Science in
«INTERNET OF THINGS AND INTELLIGENT ENVIRONMENTS»

INTERNET OF THINGS AND INTELLIGENT ENVIRONMENTS

Development of online training methods for radial basis function neural networks



MSc Student: Demeneopoulos Emmanouil
AM: 20003

MSc Thesis Supervisor: Alexandros Alexandridis, Professor

Egaleo-Athens, July, 2023

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

Αλεξανδρίδης Αλέξανδρος Καθηγητής (Επιβλέπων)	Ζώης Ηλίας Αναπληρωτής Καθηγητής (Μέλος)	Φαμέλης Ιωάννης Καθηγητής (Μέλος)

Copyright ©Με επιφύλαξη παντός δικαιώματος. All rights reserved.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ και Δεμενέοπουλος Εμμανουήλ, Μάιος 2023
Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.
Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον/την συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΔΙΑΤΡΙΒΗΣ

Ο κάτωθι υπογεγραμμένος ΔΕΜΕΝΕΟΠΟΥΛΟ ΕΜΜΑΝΟΥΗΛ του ΓΕΩΡΓΙΟΥ με αριθμό μητρώου msciot20003, φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ,

δηλώνω υπεύθυνα ότι:

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

Ο Δηλών



Δεμενέοπουλος
Εμμανουήλ

*Στην σύντροφο μου Κατερίνα και στους φίλους μου Νίκο, Μάνο και Σωκράτη που με
στηρίζουν.*

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου Αλέξανδρο Αλεξανδρίδη για την καθοδήγηση του και τις πολύτιμες συμβουλές του κατά την διάρκεια των σπουδών μου. Αλλά και για την σημαντική του συνεισφορά στην διεξαγωγή της εργασίας μου. Στην συνέχεια, θα ήθελα να ευχαριστήσω την Καραμιχαλίδου Δέσποινα που ήταν πάντα διαθέσιμη να βοηθήσει σε ό,τι χρειαζόμουν και την συνολική της συμβολή στην υλοποίηση της πτυχιακής εργασίας. Επίσης, ευχαριστώ την Κατερίνα Διαμαντοπούλου για την συνεισφορά της στον φιλολογικό έλεγχο του κειμένου και την συνολική της στήριξη. Τέλος, σε ένα πιο γενικό επίπεδο οφείλω ένα ευχαριστώ σε όλους τους καθηγητές του τμήματος που με δίδαξαν στο πρόγραμμα του μεταπτυχιακού.

Μάιος 2023

Δεμενεόπουλος Εμμανουήλ

Περίληψη

Με την άφιξη της πρόκλησης των Μεγάλων Δεδομένων (Big Data) οι απαιτήσεις επεξεργασίας διαρκώς αυξάνονται. Τα τεχνητά νευρωνικά δίκτυα RBF αποτελούν λύση σε πολλά προβλήματα που αφορούν στην επεξεργασία δεδομένων του πραγματικού κόσμου. Προβλήματα που έχουν μεταβαλλόμενη φύση και παρέχουν συνεχόμενη ροή δεδομένων, καθιστούν τις διαδικασίες μαζικής εκπαίδευσης ανεπαρκείς για τις απαιτήσεις τους. Για την αντιμετώπιση τέτοιου είδους προβλημάτων αναπτύχθηκε ένα πλήθος από διαδικασίες μηχανικής μάθησης, που έχουν την ικανότητα να εκπαιδεύουν σταδιακά με την άφιξη κάθε νέου δεδομένου, δημιουργώντας μοντέλα που είναι διαρκώς ενημερωμένα για την επικαιρότητα. Η ανάπτυξη νέων αλγορίθμων που θα μειώνουν το υπολογιστικό κόστος, είναι πολύ σημαντική για προβλήματα με απαιτήσεις σε πραγματικό χρόνο. Στα πλαίσια της συγκεκριμένης πτυχιακής εργασίας δημιουργήθηκε μία νέα διαδικασία μάθησης με αφορμή τα μειονεκτήματα που χαρακτηρίζουν τις υπάρχουσες διαδικασίες. Η νέα διαδικασία που δημιουργήθηκε με το όνομα FM-RLS, συνδυάζει πλεονεκτήματα από γνωστές διαδικασίες μάθησης χωρίς να διατηρεί τα μειονεκτήματα τους. Ο αλγόριθμος FM-RLS χωρίζει τη διαδικασία μάθησης σε δύο στάδια. Στο πρώτο στάδιο ο αλγόριθμος ασαφών μέσων αναλαμβάνει την εύρεση των κέντρων, ενώ στο δεύτερο στάδιο ο αλγόριθμος αναδρομικών ελαχίστων τετραγώνων αναβαθμίζει τα συνάπτικα βάρη του δικτύου. Η νέα διαδικασία προσφέρει τη δυνατότητα αναπροσαρμογής της κρυφής στοιβάδας μέσω της πρόσθεσης και αφαίρεσης νευρώνων, ενώ ταυτόχρονα παρέχει τεχνικές αναπροσαρμογής των διαστάσεων των απαιτούμενων πινάκων για να μπορεί να εκπαιδεύει τα βάρη σταδιακά με τη διαδικασία αναδρομικών ελαχίστων τετραγώνων. Η σύγκριση της προτεινόμενης διαδικασίας με άλλους αλγόριθμους σε ένα πλήθος από προβλήματα κατηγοριοποίησης με μεταβαλλόμενη φύση, δείχνει την υπεροχή της σε θέματα απόδοσης, ταχύτητας και δυνατότητας γενίκευσης.

Λέξεις Κλειδιά: Μηχανική μάθηση, Online μάθηση, Σταδιακή μάθηση, Νευρωνικά Δίκτυα RBF, Αλγόριθμος ασαφών μέσων.

Abstract

With the arrival of the Big Data challenge, processing tasks are increasing. RBF artificial neural networks are a solution to many real-world data processing problems. Problems with a changing nature and continuous data streams make batch training procedures inadequate due to their increased requirements. To deal with such problems, a number of machine learning processes have been developed, which have the ability to train incrementally with the arrival of each new data, creating models that are constantly updated. The development of new algorithms that will reduce the computational cost is very important for problems with real-time requirements. In this thesis, a new learning process was developed, in order to overcome the disadvantages that characterize the existing processes. The new procedure created under the name FM-RLS combines advantages of known learning procedures without retaining their disadvantages. FM-RLS divides the learning process into two stages. In the first stage, the fuzzy means algorithm undertakes to find the centers, while in the second stage the recursive least squares algorithm upgrades the synaptic weights of the network. The new procedure offers technical additions and removals of neurons. While at the same time it provides techniques for resizing the required matrices to be able to train incrementally with the recursive least squares process. The proposed procedure is compared with other algorithms in a multitude of classification problems with a changing nature, where it is shown to excel in terms of speed performance and generalization capabilities.

Keywords: Machine learning, Online learning, Incremental learning, RBF Neural Networks, Fuzzy Means.

Πίνακας περιεχομένων

Ευχαριστίες	iii
Περίληψη	v
Abstract	vii
Πίνακας περιεχομένων	x
Πίνακας σχημάτων	xi
Πίνακας πινάκων	xiii
1 Εισαγωγή	1
1.1 Πλαίσιο, σκοπός και στόχοι της πτυχιακής εργασίας	3
1.2 Οργάνωση, κεφαλαίωση, διάρθρωση της εργασίας	4
2 Εισαγωγή στα νευρωνικά δίκτυα	5
2.1 Τι είναι νευρωνικό δίκτυο	5
2.2 Μοντέλα νευρώνων	5
2.3 Συναρτήσεις ενεργοποίησης	7
2.3.1 Συνάρτηση κατωφλίου	7
2.3.2 Σιγμοειδής	8
2.4 Αρχιτεκτονικές δικτύων	9
2.4.1 Δίκτυα Πρόσθιας Τροφοδότησης Ενός Επιπέδου	9
2.4.2 Πολυεπίπεδα Δίκτυα Πρόσθιας Τροφοδότησης MLP	10
2.4.3 Αναδρομικά δίκτυα	10
2.5 Διαδικασίες μάθησης	10
2.5.1 Μάθηση με εκπαιδευτή	11
2.5.2 Μάθηση χωρίς Εκπαιδευτή	12
2.6 Νευρωνικά δίκτυα RBF	13
2.6.1 Αρχιτεκτονική και διαδικασία εκπαίδευσης	13
2.6.2 Ακτινικές συναρτήσεις βάσης	15
2.6.3 Αλγόριθμος ασαφούς λογικής	16
3 Online εκπαίδευση	21

3.1	Εφαρμογές και πλεονεκτήματα	21
3.2	Off-line εκπαίδευση	22
3.3	On-line εκπαίδευση	23
3.4	Χρονικά μεταβαλλόμενα σύνολα δεδομένων	24
3.5	On-line εκπαίδευση νευρωνικών δικτύων RBF	25
3.5.1	Αλγόριθμος εκπαίδευσης με χρήση ασαφών μέσων	25
3.5.2	Οικογένεια αλγορίθμων RAN	28
4	Προτεινόμενη Μεθοδολογία	37
4.1	Προκλήσεις και προβλήματα	37
4.2	Επισκόπηση προτεινόμενης μεθοδολογίας FM-RLS	38
4.3	Υποδομή νευρωνικού δικτύου RBF	39
4.3.1	Αρχικοποίηση παραμέτρων και εισαγωγή πρώτων δεδομένων	41
4.3.2	Αφιξη νέων δεδομένων, έλεγχος και διαγραφή κέντρων	43
4.3.3	Έλεγχος και προσθήκη νέου νεύρωνα	44
4.3.4	Αναβάθμιση παραμέτρων με την μέθοδο αναδρομικών ελαχίστων τετραγώνων	45
5	Πείραμα και αποτελέσματα	47
5.1	Πειραματική διαδικασία	47
5.2	Αποτελέσματα και συζήτηση	50
6	Συμπεράσματα	59
	References	63

Πίνακας σχημάτων

Εικόνα 1.	Μη γραμμικό μοντέλο νευρώνα	6
Εικόνα 2.	Αφινικός μετασχηματισμός λόγω της ύπαρξης πόλωσης	7
Εικόνα 3.	Συνάρτηση κατωφλίου	8
Εικόνα 4.	Σιγμοειδής συνάρτηση	8
Εικόνα 5.	Δίκτυο πρόσθιας τροφοδοσίας ενός επιπέδου (feedforward)	9
Εικόνα 6.	Δίκτυο πρόσθιας τροφοδότησης με ένα κρυφό επίπεδο	10
Εικόνα 7.	Αναδρομικό δίκτυο χωρίς αυτό-ανάδραση	11
Εικόνα 8.	Αναδρομικό δίκτυο με κρυφούς νευρώνες	11
Εικόνα 9.	Σχηματικό διάγραμμα της μάθησης με εκπαιδευτή	12
Εικόνα 10.	Σχηματικό διάγραμμα της ενισχυτικής μάθησης	13
Εικόνα 11.	Σχηματικό διάγραμμα μη επιβλεπόμενης μάθησης	13
Εικόνα 12.	Νευρωνικό δίκτυο RBF	14
Εικόνα 13.	Διαδικασία κατάτμησης σε ασαφή υποσύνολα	17
Εικόνα 14.	Κατηγοριοποίηση δεδομένου εισόδου στα ασαφή υπόχωρο	18
Εικόνα 15.	Μοντέλο Off-line μαζικής εκπαίδευσης	22
Εικόνα 16.	Μοντέλο Off-line σταδιακής εκπαίδευσης	23
Εικόνα 17.	Μοντέλο On-line εκπαίδευσης	23
Εικόνα 18.	Είδη μεταβολής συνόλων δεδομένων.	25
Εικόνα 19.	Διάγραμμα ψευδοκώδικα αλγορίθμου.	26
Εικόνα 20.	Διάγραμμα ψευδοκώδικα αλγορίθμου RAN.	30
Εικόνα 21.	Διάγραμμα ψευδοκώδικα διαδικασίας διαγραφής νευρώνων.	32
Εικόνα 22.	Διάγραμμα ψευδοκώδικα M-RAN.	33
Εικόνα 23.	Διάγραμμα ψευδοκώδικα GAP-RBF.	36
Εικόνα 24.	Διάγραμμα ψευδοκώδικα προτεινόμενης μεθοδολογίας FM-RLS.	39
Εικόνα 25.	Δομή νευρωνικού δικτύου προτεινόμενης διαδικασίας.	41
Εικόνα 26.	Γραφήματα ακρίβειας	55
Εικόνα 27.	Γραφήματα ακρίβειας	56
Εικόνα 28.	Γραφήματα ανάπτυξης νευρώνων	57
Εικόνα 29.	Γραφήματα ανάπτυξης νευρώνων	58

Πίνακας πινάκων

Πίνακας 1.	Χαρακτηριστικά δεδομένων.	49
Πίνακας 2.	Προσαρμοσμένοι μετα-παράμετροι αλγόριθμων για όλα τα σύνολα δεδομένων.	50
Πίνακας 3.	Πίνακας αποτελεσμάτων αλγόριθμων εκπαίδευσης.	51
Πίνακας 4.	Πίνακας αποκλίσεων από την διαδικασία με την βέλτιστη απόδοση.	53

Κεφάλαιο 1

Εισαγωγή

Στις μέρες μας δεδομένα δημιουργούνται, διακινούνται και επεξεργάζονται με ρυθμούς που αυξάνονται διαρκώς. Τεχνολογίες που αφορούν στο διαδίκτυο, όπως για παράδειγμα το Cloud Computing, τα Social Media ή και άλλες ανερχόμενες τεχνολογίες σαν το Internet Of Things (IOT), το οποίο βρίσκει εφαρμογή σε τεχνολογίες διαχείρισης διανομής ηλεκτρικής ενέργειας, σε έξυπνες πόλεις και σπίτια, σε διαχείριση διανομής προϊόντων, σε έξυπνες ηλεκτρικές συσκευές και σε πολλές άλλες εφαρμογές. Ένα απεριόριστο πλήθος από επιστημονικούς, τεχνολογικούς και βιομηχανικούς κλάδους βασίζονται στην καθημερινή δημιουργία, επεξεργασία και χρήση μεγάλων όγκων δεδομένων που λαμβάνονται σε μικρά χρονικά διαστήματα, δημιουργώντας συνεχώς αυξανόμενες απαιτήσεις επεξεργασίας [1], [2]. Η μηχανική μάθηση με χρήση τεχνητών νευρωνικών δικτύων έρχεται να δώσει λύσεις σε προβλήματα διαχείρισης ταξινόμησης και κατηγοριοποίησης σε αυτούς τους όγκους δεδομένων [3]. Η πρόκληση των μεγάλων δεδομένων (Big Data) δεν σταματάει όμως μόνο στο ρυθμό και τον όγκο με τον οποίο λαμβάνονται, αλλά και στην μορφή. Σε προβλήματα που αντιμετωπίζονται στον πραγματικό κόσμο, τα δεδομένα λαμβάνονται συχνά σε δύσκολα επεξεργάσιμες μορφές, οι οποίες περιλαμβάνουν θόρυβο και αλλοιώσεις. Έτσι, μία πρότερη επεξεργασία των δεδομένων με τεχνικές κανονικοποίησης, ολοκληρώσεις, μετασχηματισμούς και ξεκαθάρισμα κατεστραμμένων δεδομένων, είναι απαραίτητη για την προσαρμογή τους στον αλγόριθμο μηχανικής μάθησης που θα τα επεξεργαστεί [4]. Με την άφιξη των μεγάλων δεδομένων, πλέον, δεν μπορούμε να θεωρήσουμε τα προβλήματα σαν μία στατική κατάσταση. Τα περισσότερα προέρχονται από τον πραγματικό κόσμο έχουν μεταβαλλόμενη φύση και ασταθή δομή, και συχνά αποκαλούνται στην βιβλιογραφία με τον όρο "Concept Drift" (Μεταβλητή Φύση) [2], [4], [5]. Σε προβλήματα όπου τα δεδομένα λαμβάνονται διαρκώς το ένα μετά το άλλο ή και σε μορφή πλειάδων, οι κλασικές διαδικασίες Μάζικης Εκπαίδευσης (Batch Learning) αντιμετωπίζουν σοβαρά προβλήματα. Οι χρονοβόρες διαδικασίες μάζικης εκπαίδευσης αδυνατούν να αντεπεξέλθουν στα απαιτούμενα χρονικά περιθώρια συσσωρεύοντας διαρκώς όλο και περισσότερα μη επεξεργασμένα δεδομένα. Λόγω αυτού αυξάνονται οι απαιτήσεις αποθήκευσης, ενώ ταυτόχρονα το μοντέλο δεν είναι επαρκώς ενημερωμένο για την τρέχουσα κατάσταση. Για την αντιμετώπιση αυτών των προβλημάτων έχουν αναπτυχθεί νέες διαδικασίες εκπαίδευσης, οι οποίες λαμβάνουν και εκπαιδεύουν το μοντέλο σταδιακά με την άφιξη κάθε δεδομένου. Οι διαδικασίες Σταδιακής Μάθησης (Incremental learning, Online learning), πλεονεκτούν σημαντικά προσφέροντας την δυνατότητα άμεσης επεξεργασίας των δεδομένων κατά την λήψη τους με ευέλικτες και γρήγορες διαδικασίες και σημαντικά μικρότερες αποθηκευτικές απαιτήσεις [4].

Συχνά διαδικασίες μάζικης μάθησης τροποποιούνται κατάλληλα, έτσι ώστε να μπορούν να λειτουργήσουν σαν διαδικασίες σταδιακής μάθησης και να καλύψουν τις απαιτήσεις προβλημάτων δυναμικής φύσεως [4]. Για παράδειγμα, η γνωστή διαδικασία μάθησης SVM (Support Vector Machine) τροποποιήθηκε σε μία διαδικασία σταδιακής μάθησης (Incremental SVM) και χρησιμοποιήθηκε για την αναγνώριση ανθρώπων από καρέ βίντεο (video frames) με επιτυχία της τάξεως

98.46% με λήψη μόνο του 5% των διαθέσιμων δεδομένων [6]. Επίσης, άλλες υλοποιήσεις δημιούργησαν μία έκδοση σταδιακής μάθησης του διάσημου αλγόριθμου Stochastic Gradient Descent (SGD) [7]. Ένα ακόμη παράδειγμα τροποποιημένου αλγόριθμου για σταδιακή μάθηση είναι μία έκδοση του αλγόριθμου Naive Bays (NB) στον Incremental Naive Bays (INB) με αποτελέσματα κοντά στη διαδικασία NB μαζικής εκπαίδευσης [8]. Επιπρόσθετα, ο αλγόριθμος μαζικής εκπαίδευσης Random Forests (RF) τροποποιήθηκε για να εκπαιδεύει σταδιακά από ροές δεδομένων στον αλγόριθμο Adaptive Random Forests (ADF) [9]. Σε μία επιπλέον μελέτη παρουσιάστηκε μία έκδοση σταδιακού αλγορίθμου Incremental Extreme Learning Machine (IELM) που παρέχει μία τεχνική αφαίρεσης κόμβων από την κρυφή στιβάδα χωρίς αυτό να επηρεάζει την απόδοση του δικτύου [10].

Ένα ακόμα πεδίο που έχει ιδιαίτερο ενδιαφέρον για υλοποιήσεις αλγορίθμων σταδιακής εκπαίδευσης είναι τα νευρωνικά δίκτυα RBF. Τα δίκτυα RBF χαρακτηρίζονται από την απλή αρχιτεκτονική τους, δεδομένου ότι αποτελούνται από μόνο μία κρυφή στιβάδα νευρώνων [11]–[13]. Επίσης, προσφέρουν την δυνατότητα διαχωρισμού του αλγόριθμου εκπαίδευσης σε δύο στάδια. Ένα στάδιο για την τοποθέτηση των κέντρων των ακτινικών συναρτήσεων βάσης και ένα για την εκπαίδευση των συνάψεων του δικτύου. Αυτός ο διαχωρισμός προσφέρει ευελιξία στην διαδικασία εκπαίδευσης, μειώνοντας το υπολογιστικό κόστος κατά την αναβάθμιση των συναπτικών βαρών, ενώ απλοποιεί την πολυπλοκότητα του προβλήματος. Μία από τις πιο συνηθισμένες τακτικές για την επιλογή των κέντρων των ακτινικών συναρτήσεων βάσης είναι η χρήση της γνωστής διαδικασίας με όνομα k-means. Με την βοήθεια του αλγορίθμου k-means τα δεδομένα ταξινομούνται σε ομάδες με βάση την θέση τους στον πολυδιάστατο χώρο, τον οποίο σχηματίζουν οι μεταβλητές εισόδου. Οι συναρτήσεις βάσης τοποθετούνται στο κέντρο των ομάδων που σχηματίστηκαν και στο επόμενο στάδιο υπολογίζονται τα βάρη με γραμμική παλινδρόμηση [12], [14], [15].

Μία νέα τεχνική Ασαφούς Λογικής (Fuzzy means) η οποία διαμερίζει το πεδίο εισόδου σε ασαφή υποσύνολα, αντικαθιστώντας την τεχνική k-means και δημιουργώντας μία καινοτόμος διαδικασία για την εκπαίδευση ενός RBF δικτύου, παρουσιάστηκε στην [16]. Ο αλγόριθμος ασαφούς λογικής έχει στόχο την μείωση του υπολογιστικού φορτίου υπολογίζοντας το απαιτούμενο πλήθος των κέντρων, καθώς και τις θέσεις τους απευθείας. Λόγω αυτού, βρίσκει αποτελεσματική εφαρμογή σε πλήθος διαφορετικών πεδίων [17], [18]. Μεταγενέστερα, υλοποιήθηκε μία αναβαθμισμένη έκδοση που διαμερίζει σε μη συμμετρικά ασαφή υποσύνολα για την αύξηση της ακρίβειας [19].

Σε επόμενη εργασία παρουσιάστηκε μία υβριδική μέθοδος που συνδυάζει τον μη συμμετρικό ασαφή αλγόριθμο με μία εναλλακτική μέθοδο του αλγορίθμου Σμήνους Σωματιδίων (Cooperative Particle Swarm Optimization). Ο αλγόριθμος ασαφούς λογικής εφαρμόζεται σε ένα σμήνος για την εύρεση του πλήθους των απαιτούμενων κέντρων, αλλά και των συντεταγμένων των θέσεων τους. Ενώ ένα δεύτερο σμήνος αναλαμβάνει τον υπολογισμό των συναπτικών βαρών [20].

Όλες οι παραπάνω τεχνικές εκπαίδευσης νευρωνικών δικτύων RBF που έχουμε προαναφέρει, εκπαιδεύουν μαζικά και μετά την λήξη της διαδικασίας η δομή των δικτύων παραμένει σταθερή. Την έλλειψη αυτή ήρθε να καλύψει μία αναβάθμιση του αλγορίθμου, ο οποίος παρουσιάστηκε στην [16] σε μία νέα διαδικασία σταδιακής μάθησης με τεχνικές αφαίρεσης και προσθήκης νευρώνων. Η νέα διαδικασία, την οποία θα αποκαλούμε με το όνομα FUZZY-LS-RLS στα πλαίσια αυτής της εργασίας, έκανε χρήση ενός κυλιόμενου παραθύρου, το οποίο παρέχει δυνατότητα εκπαίδευσης ικανή να αντιμετωπίσει προβλήματα μεταβαλλόμενης φύσεως [21]. Σε πρόσφατη έρευνα παρουσιάστηκε μία παραλλαγή του [21] που επιτρέπει την αυτόματη αλλαγή μεγέθους του δικτύου σύμφωνα με την αποτελεσματικότητα του ασαφούς αλγορίθμου [22]. Μία ακόμη γνωστή οικογένεια σταδιακών αλγορίθμων για εκπαίδευση νευρωνικών δικτύων RBF είναι η οικογένεια των αλγορίθμων RAN. Η οικογένεια RAN είναι ικανή να αντιμετωπίσει προβλήματα κατηγοριοποίησης και προσέγγισης συνάρτησης με μεγάλη απόδοση και ταχύτητα. Ο αλγόριθμος RAN σε αντίθεση με τον αλγόριθμο ασαφούς λογικής της [16], εισάγει ένα νέο κέντρο στο δίκτυο με βάση την

καινοτομία του στον χώρο εισόδου. Το νέο κέντρο θεωρείται καινοτόμο εάν είναι επαρκώς μακριά από το κοντινότερο δεσμευμένο κέντρο και το σφάλμα πρόβλεψης για αυτό είναι μεγάλο. Στην συνέχεια, αναβαθμίζει την συντεταγμένη του κέντρου με την ένταξή τους στις εκπαιδευόμενες μεταβλητές της διαδικασίας [23]. Σε περαιτέρω μελέτη ο αλγόριθμος RAN-EKF αντικατέστησε την διαδικασία LMS (Least Mean Squares) του αλγόριθμου RAN με την διαδικασία EKF (Extended Kalman Filter), δημιουργώντας μία ενισχυμένη έκδοση του RAN με βελτιστοποιημένη απόδοση [24]. Μεταγενέστερα, ο αλγόριθμος M-RAN εισήγαγε μία διαδικασία διαγραφής νευρώνων με στόχο την αφαίρεση νευρώνων με σκοπό την μείωση του υπολογιστικού κόστους. Αυτή η αλλαγή σε συνδυασμό με την προσθήκη ενός ακόμα κριτηρίου απόφασης για την προσθήκη νέου νευρώνα, προσέφερε σημαντικά στην ταχύτητα και στην απόδοση της διαδικασίας εκπαίδευσης [25]. Επιπρόσθετα, παρουσιάστηκε μία πρόταση από τον αλγόριθμο GAP, ο οποίος αφαιρεί νευρώνες αξιολογώντας την σημαντικότητά τους σε σχέση με την κατανομή των δεδομένων εισόδου. Η συγκεκριμένη διαδικασία εισάγει μία τεχνική που αναβαθμίζει μόνο της παραμέτρους του πλησιέστερου κέντρο στο δεδομένο που εισήλθε κατά την διαδικασία αναβάθμισης των συναπτικών βαρών με σκοπό την μείωση του υπολογιστικού κόστους [26]. Σε επακόλουθη έρευνα ο αλγόριθμος SRAN παρουσίασε μία αυτο-προσαρμοζόμενη διαδικασία εκπαίδευσης στην οποία το δίκτυο εκπαιδεύεται με το εισερχόμενο δεδομένο μόνο εάν κρίνεται παραγωγικό, αποφεύγοντας επανεκπαίδευση με δεδομένα που δεν περιλαμβάνουν νέα πληροφορία [27]. Σε επόμενη μελέτη υλοποιήθηκε μία αναβαθμιζόμενη έκδοση του αλγορίθμου RAN με χρήση ενός κυλιόμενου παραθύρου για την κατηγοριοποίηση εγγράφων και, ενώ συγκρίθηκε με άλλες γνωστές διαδικασίες, έδειξε να υπερέχει [28].

Η διαδικασία σταδιακής εκπαίδευσης είναι ένα πεδίο με μεγάλο ενδιαφέρον και πολλά περιθώρια ανάπτυξης. Τα περιθώρια αυτά αποτέλεσαν έμπνευση για την διεξαγωγή της συγκεκριμένης πτυχιακής εργασίας με στόχο την δημιουργία και την μελέτη καινοτόμων τεχνικών, οι οποίες θα προσφέρουν στην οικογένεια των σταδιακών αλγορίθμων. Στην επόμενη ενότητα θα αναφερθούμε αναλυτικότερα στους στόχους της εργασίας.

1.1 Πλαίσιο, σκοπός και στόχοι της πτυχιακής εργασίας

Με την άφιξη του προβλήματος των μεγάλων δεδομένων (Big Data) οι απαιτήσεις για ταχύτερες διαδικασίες εκπαίδευσης που επιτυγχάνουν ακριβείς προβλέψεις διαρκώς αυξάνεται. Αυτό σημαίνει ότι η δημιουργία καινοτόμων διαδικασιών με βελτιστοποιημένες αποδόσεις είναι ιδιαίτερα σημαντική. Η διαδικασία εκπαίδευσης που παρουσιάζεται στην [16] και θα την αποκαλούμε FUZZY-LS-RLS, αποτελεί μία καινοτόμο διαδικασία σταδιακού αλγορίθμου. Η συγκεκριμένη διαδικασία εκπαιδεύει νευρωνικά δίκτυα RBF για αντιμετώπιση προβλημάτων μεταβαλλόμενης φύσης και συνεχόμενης ροής δεδομένων. Παρόλο την καινοτομία που εισήγαγε η ασαφής λογική του αλγορίθμου FUZZY-LS-RLS φαίνεται να χαρακτηρίζεται από ένα σημαντικό μειονέκτημα το οποίο αφορά το γεγονός ότι είναι μία διαδικασία η οποία δεν είναι εξ ολοκλήρου σταδιακή. Ο αλγόριθμος FUZZY-LS-RLS στην περίπτωση που θα χρειαστεί να προσθέσει ή να αφαιρέσει ένα κέντρο στο δίκτυο, εκπαιδεύει εξ ολοκλήρου από την αρχή μαγικά, με τη διαδικασία ελαχίστων τετραγώνων αυξάνοντας το υπολογιστικό κόστος της διαδικασίας. Η εκπαίδευση στην περίπτωση της χρήσης ελαχίστων τετραγώνων γίνεται από ένα πλήθος αποθηκευμένων δεδομένων των οποίων ο αριθμός καθορίζεται από ένα παράθυρο, γεγονός που αυξάνει και τις απαιτήσεις μνήμης τις διαδικασίας εκπαίδευσης. Σε αντίθεση με τον αλγόριθμο FUZZY-LS-RLS οι αλγόριθμοι της οικογένειας RAN είναι εξ ολοκλήρου σταδιακές διαδικασίες [23]–[26]. Παρόλο που πλεονεκτούν στο γεγονός ότι δεν χρειάζεται να εκπαιδεύσουν το δίκτυο κατά την προσθήκη ή την αφαίρεση ενός νέου κέντρου οι διαδικασίες RAN εισάγουν την διαδικασία εύρεσης των κέντρων στη διαδικασία μάθησης σαν μεταβλητές δικτύου, γεγονός που αυξάνει σημαντικά τις υπολογιστικές απαιτήσεις τις διαδικασίες. Σε αντίθεση με τους αλγόριθμους RAN ο αλγόριθμος FUZZY-LS-RLS δεν επι-

βαρύνει τη διαδικασία μάθησης με την εύρεση των συντεταγμένων, δεδομένου ότι η διαδικασία εύρεσης του πλήθους και της θέσης των κέντρων είναι κάτι που αναλαμβάνει αποκλειστικά η διαδικασία ασαφούς λογικής (Fuzzy Means). Αυτός ο διαχωρισμός μειώνει το πλήθος των μεταβλητών του δικτύου και έτσι ελαττώνει το υπολογιστικό κόστος κατά την διαδικασία αναβάθμισης τους, ενώ ταυτόχρονα επιτυγχάνει την απλοποίηση της πολυπλοκότητας του προβλήματος. Στα πλαίσια της πτυχιακής εργασίας στόχος είναι να αναπτυχθεί μία νέα διαδικασία εκπαίδευσης η οποία θα συνδυάζει τα πλεονεκτήματα της διαδικασίας FUZZY-LS-RLS με τα πλεονεκτήματα των αλγορίθμων της οικογένειας RAN. Στην συνέχεια της εργασίας για λόγους ευκολίας θα αναφερόμαστε στην νέα διαδικασία με το όνομα FM-RLS. Πιο αναλυτικά η νέα διαδικασία FM-RLS θα διατηρεί την διαδικασία Fuzzy Means για την εύρεση των κέντρων και σε συνδυασμό με την χρήση της διαδικασίας αναβάθμισης του αντίστροφου πίνακα συνδιακύμανσης που χρησιμοποιεί η διαδικασία RAN-EKF, M-RAN και GAP μπορεί να είναι εξ ολοκλήρου μία σταδιακή διαδικασία, διατηρώντας τα πλεονεκτήματα του FUZZY-LS-RLS χωρίς τα μειονεκτήματά του. Η μείωση της πολυπλοκότητας του προβλήματος που προσφέρει η διαδικασία Fuzzy Means σε συνδυασμό με την εξ ολοκλήρου σταδιακή μάθηση που επιτυγχάνεται από τη μέθοδο αναβάθμισης που χρησιμοποιεί RAN-EKF και M-RAN, αναμένουμε να μας προσφέρει μία νέα διαδικασία εκπαίδευσης με υψηλές αποδόσεις και χαμηλές απαιτήσεις σε υπολογιστικό κόστος αλλά και μνήμη.

1.2 Οργάνωση, κεφαλαίωση, διάρθρωση της εργασίας

Στο κεφάλαιο 1 γίνεται μία βιβλιογραφική αναδρομή σε έρευνες με παρόμοια θεματολογία που υλοποιήθηκαν στο παρελθόν. Στο κεφάλαιο αυτό ο αναγνώστης μπορεί να ενημερωθεί για την ερευνητική πορεία που οδήγησε στην υλοποίηση της συγκεκριμένης εργασίας. Επιπρόσθετα μία πιο αναλυτική περιγραφή της μελέτης που πρόκειται να πραγματοποιηθεί στα πλαίσια της εργασίας και οι λόγοι που οδήγησαν σε αυτή τη υλοποίηση καλύπτονται από το κεφάλαιο αυτό.

Το κεφάλαιο 2 εισάγει των αναγνώστη στις βασικές έννοιες και τις αρχιτεκτονικές των τεχνητών νευρωνικών δικτύων καθώς και των διαδικασιών μάθησης. Στην συνέχεια του κεφαλαίου αυτού θα αναλυθούν οι βασικές αρχές των νευρωνικών δικτύων ακτινικών συναρτήσεων βάση RBF.

Στο κεφάλαιο 3 Εισάγεται η έννοια της Online διαδικασίας μάθησης και της μάθησης από ροές δεδομένων εκπαίδευσης. Επιπρόσθετα το κεφάλαιο εισάγει τον αναγνώστη στην έννοια των προβλημάτων εκπαίδευσης μεταβαλλόμενης φύσης. Στο τέλος του κεφαλαίου περιγράφονται σταδιακές διαδικασίες μάθησης που έχουν υλοποιηθεί στο παρελθόν.

Το Κεφάλαιο 4 αποτελεί μία αναλυτική περιγραφή της υλοποίησης του θέματος της πτυχιακής εργασίας. Το κεφάλαιο αυτό ξεκινάει με την ανάπτυξη των μειονεκτημάτων που εντοπίστηκαν στις ήδη υπάρχουσες διαδικασίες που αποτελούν την αιτία της υλοποίησης της εργασίας. Ενώ συνεχίζει μία αναλυτική περιγραφή της προτεινόμενης διαδικασίας εκπαίδευσης “FM-RLS”.

Στο κεφάλαιο 5 αναπτύσσεται η διαδικασία υλοποίησης της πειραματικής διαδικασίας και οι υποδομές που χρησιμοποιήθηκαν. Στη συνέχεια παρουσιάζονται και συγκρίνονται όλα τα αποτελέσματα των διαδικασιών εκπαίδευσης. Τέλος το κεφάλαιο κλείνει με μία συζήτηση και σχετικά συμπεράσματα για τα αποτελέσματα της πειραματικής διαδικασίας.

Κεφάλαιο 2

Εισαγωγή στα νευρωνικά δίκτυα

2.1 Τι είναι νευρωνικό δίκτυο

Σε αντίθεση με τους κλασικούς ηλεκτρονικούς υπολογιστές, ο ανθρώπινος εγκέφαλος εκτελεί υπολογισμούς με εντελώς διαφορετικό τρόπο. Και σε αυτό το γεγονός στηρίχτηκε εξαρχής η ιδέα των τεχνητών νευρωνικών δικτύων. Αναλυτικότερα, ο ανθρώπινος εγκέφαλος αποτελεί ένα μη γραμμικό παράλληλο υπολογιστικό σύστημα, το οποίο αποτελείται από στοιχειώδεις υπολογιστικές μονάδες που ονομάζονται νευρώνες και είναι δομημένες κατάλληλα ώστε να εκτελούν στοχευμένους υπολογισμούς. Υπολογισμοί όπως: Αναγνώριση προτύπων, έλεγχος κίνησης, αντίληψη και οποιαδήποτε άλλη λειτουργία χρήσιμη στον άνθρωπο αντιμετωπίζεται σε πραγματικό χρόνο. Ο ανθρώπινος εγκέφαλος είναι ικανός να εκτελέσει εργασίες με εντυπωσιακές ταχύτητες, όπως η αναγνώριση ενός γνωστού προσώπου μέσα σε 100-200 ms. Μια ιδιαίτερα σημαντική ικανότητα του εγκεφάλου είναι η δυνατότητα να δημιουργεί κανόνες κάτι που συνηθίζουμε να αποκαλούμε εμπειρία. Τα τεχνητά νευρωνικά δίκτυα κατασκευάζονται σε μορφή ολοκληρωμένων ηλεκτρονικών κυκλωμάτων ή εξομοιώνονται σε μορφή λογισμικού με σκοπό να αντιγράψουν τις λειτουργίες του ανθρώπινου εγκεφάλου. Συγκεκριμένα, οι λειτουργίες πραγματοποιούνται με την διασύνδεση ενός μεγάλου αριθμού υπολογιστικών κυττάρων, τα οποία αποκαλούνται «νευρώνες» ή «μονάδες επεξεργασίας». Το νευρωνικό δίκτυο αποτελεί μία προσαρμοστική μηχανή και μπορούμε να το ορίσουμε ως εξής:

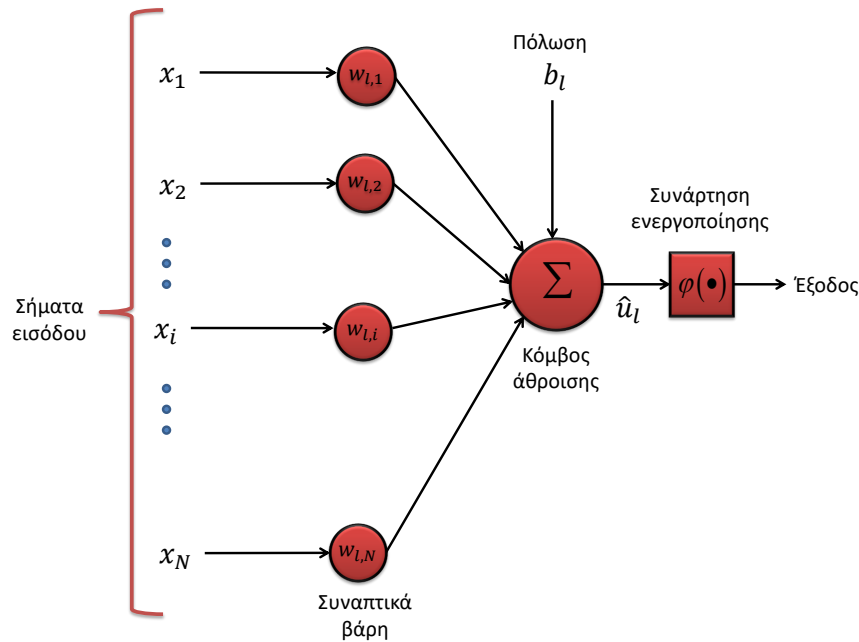
Ένα νευρωνικό δίκτυο είναι ένας τεράστιος παράλληλος επεξεργαστής με καταναμημένη αρχιτεκτονική, ο οποίος αποτελείται από απλές μονάδες επεξεργασίας και έχει από τη φύση του την δυνατότητα να αποθηκεύει εμπειρική γνώση και να την καθιστά διαθέσιμη για χρήση. Μοιάζει με τον ανθρώπινο εγκέφαλο σε δύο σημεία:

- 1 Το δίκτυο προσλαμβάνει την γνώση από το περιβάλλον του μέσω μίας διαδικασίας μάθησης.
- 2 Η ισχύς των συνδέσεων μεταξύ των νευρώνων, που αποκαλούνται συνάπτικα βάρη, χρησιμοποιείται για την αποθήκευση της γνώσης που αποκτάται.

2.2 Μοντέλα νευρώνων

Ένας νευρώνας είναι το βασικό δομικό στοιχείο ενός νευρωνικού δικτύου και στην πραγματικότητα ένα νευρωνικό δίκτυο είναι ένα πλήθος νευρώνων διασυνδεδεμένο μεταξύ τους. Ένα από τα πιο κλασικά μοντέλα νευρώνα που συνηθίζεται να χρησιμοποιείται στα νευρωνικά δίκτυα, παρουσιάζεται στο σχήμα 1 και διακρίνεται από τα εξής χαρακτηριστικά:

1. Ένα σύνολο συνδέσμων ή συνάψεων, κάθε μία εκ των οποίων χαρακτηρίζεται από το δικό της βάρος. Πιο συγκεκριμένα, ένα σήμα στην είσοδο της σύναψης N που συνδέεται με το νευρώνα l πολλαπλασιάζεται με το συνοπτικό βάρος. Σε αντίθεση με τον ανθρώπινο νευρώνα, το συνοπτικό βάρος ενός τεχνητού νευρώνα μπορεί να λαμβάνει και αρνητικές και θετικές τιμές.
2. Έναν αθροιστή για την άθροιση των σημάτων, τα οποία προέρχονται από τα συναπτικά βάρη.
3. Την συνάρτηση ενεργοποίησης ή συνάρτηση περιορισμού που οριοθετεί το σήμα εξόδου σε ένα πεπερασμένο εύρος τιμών, συνήθως κανονικοποιημένο και γράφεται ως μοναδιαίο κλειστό διάστημα με τη μορφή $[0,1]$, ή $[-1,1]$.



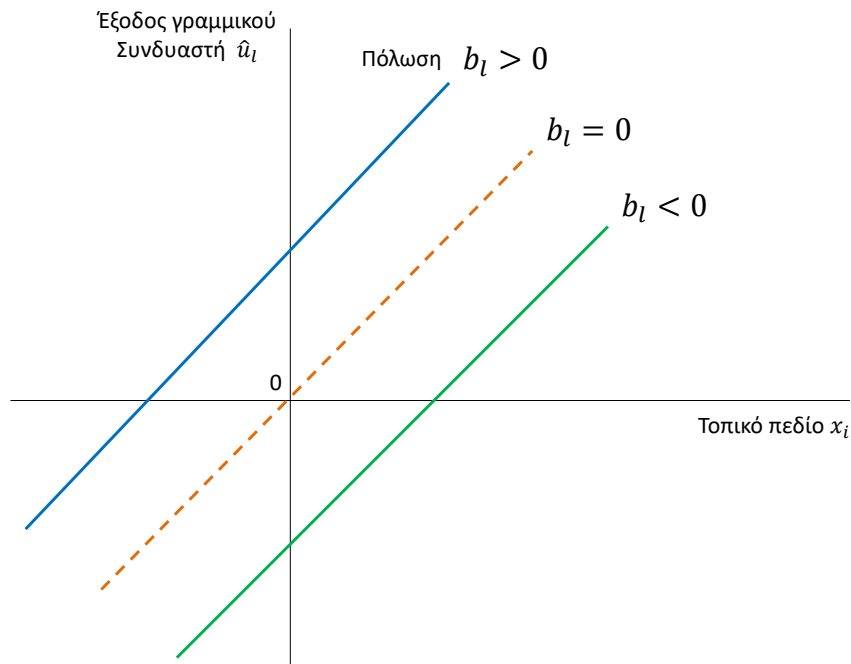
Εικόνα 1. Μη γραμμικό μοντέλο νευρώνα

Η έξοδος \hat{u}_l (πριν τη συνάρτηση ενεργοποίησης) του μοντέλου του σχήματος 1 περιγράφεται μαθηματικά από την σχέση 2.1.

$$\hat{u}_l = \sum_{i=1}^N w_{l,i}x_i + b_l \quad (2.1)$$

Όπου $x_1, x_2, \dots, x_i, \dots, x_N$ είναι τα δεδομένα εισόδου και $w_{l,1}, w_{l,2}, \dots, w_{l,i}, \dots, w_{l,N}$, είναι τα συναπτικά βάρη του νευρώνα l αντίστοιχα. Το \hat{u}_l είναι η έξοδος του αθροιστή και $\varphi(\bullet)$ είναι η συνάρτηση ενεργοποίησης. Το μοντέλο περιλαμβάνει επιπλέον μία εξωτερική διέγερση, η οποία συμβολίζεται με b_l και συχνά αποκαλείται και πόλωση. Η πόλωση b_l μετατοπίζει την συνάρτηση \hat{u}_l σε σχέση με την αρχή των αξόνων και έχει ως αποτέλεσμα την εφαρμογή ενός αφινικού μετασχηματισμού στην έξοδο \hat{u}_l του γραμμικού συνδυαστή (αθροιστή). Ανάλογα με το αν η εξωτερική διέγερση είναι θετική ή αρνητική, η σχέση μεταξύ του δυναμικού ενεργοποίησης (activation potential) του νευρώνα l και της εξόδου του γραμμικού συνδυασμού μετατοπίζεται με τον τρόπο που φαίνεται στο σχήμα 2.2.

Λόγω της πόλωσης, η ευθεία του σχήματος 2 μετατοπίζεται σε σχέση με την αρχή των αξόνων. Τέλος, μπορούμε να αναδομήσουμε την εξίσωση 2.1 στην μορφή της 2.2, έτσι ώστε να συμπεριλαμβάνει τον όρο b_l στα συναπτικά βάρη, θέτοντας όπου $w_{l,0} = b_l$ και $x_0 = 1$.



Εικόνα 2. Αφινικός μετασχηματισμός λόγω της ύπαρξης πόλωσης

$$\hat{u}_l = \sum_{i=0}^N w_{l,i} x_i \quad (2.2)$$

2.3 Συναρτήσεις ενεργοποίησης

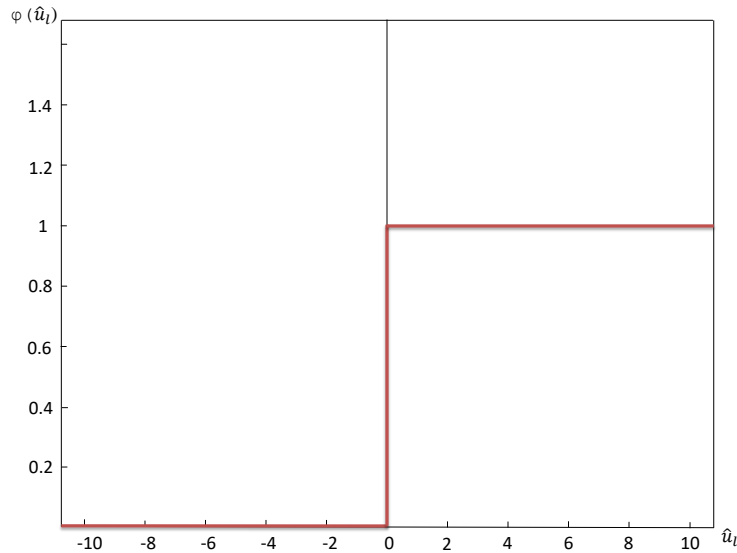
Η συνάρτηση ενεργοποίησης αποτελεί έναν μετασχηματισμό της εξόδου του αθροιστή και καθορίζει την ενεργοποίηση ή όχι της εξόδου στην οποία εφαρμόζεται. Η συνάρτηση ενεργοποίησης μπορεί να είναι πολλών μορφών και στην συγκεκριμένη ενότητα θα αναφερθούμε στα πιο συνηθισμένα είδη.

2.3.1 Συνάρτηση κατωφλίου

Η συνάρτηση κατωφλίου (Threshold Function) παρουσιάζεται στο σχήμα 3 και περιγράφεται από την σχέση 2.3

$$\varphi(\hat{u}_l) = \begin{cases} 1 & \text{εάν } \hat{u}_l \leq 0 \\ 0 & \text{εάν } \hat{u}_l > 0 \end{cases} \quad (2.3)$$

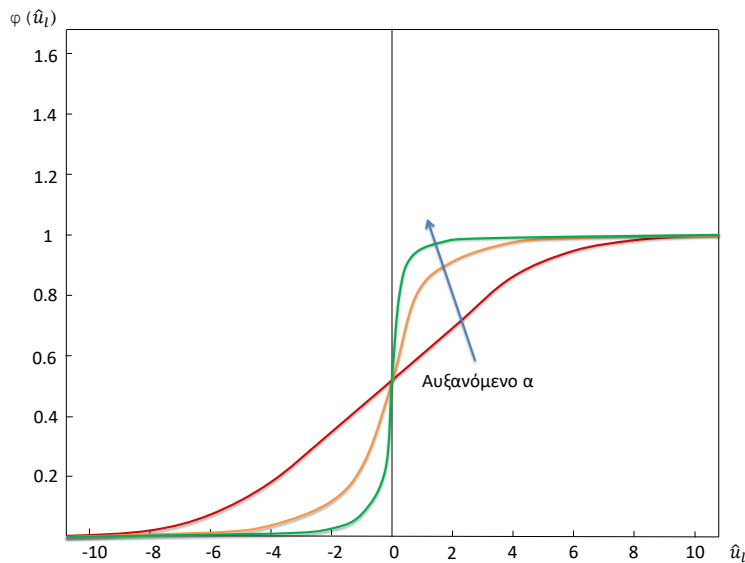
Σύμφωνα με το μοντέλο της συνάρτησης κατωφλίου, η οποία συχνά στην μηχανική αποκαλείται και συνάρτηση Heaviside, η έξοδος του νευρώνα θα δώσει την τιμή 1 αν το τοπικό πεδίο \hat{u}_l είναι μεγαλύτερο ή ίσο του 0, ενώ σε κάθε άλλη περίπτωση θα λάβει την τιμή 0



Εικόνα 3. Συνάρτηση κατωφλίου

2.3.2 Σιγμοειδής

Η πιο συνηθισμένη μορφή συνάρτησης ενεργοποίησης που συναντάμε στα τεχνητά νευρωνικά δίκτυα είναι η σιγμοειδής συνάρτηση η οποία θυμίζει “S” όπως παρατίθεται στο σχήμα 4 Ορίζεται



Εικόνα 4. Σιγμοειδής συνάρτηση

ως αυστηρά αύξουσα συνάρτηση και χαρακτηρίζεται από γραμμική και μη γραμμική συμπεριφορά και περιγράφεται από την συνάρτηση 2.4.

$$\varphi(\hat{u}_i) = \frac{1}{1 + \exp(-a\hat{u}_i)} \quad (2.4)$$

Για τιμές του a που τείνουν στο άπειρο η σιγμοειδής συνάρτηση τείνει να ταυτιστεί με την συνάρτηση κατωφλίου. Σε αντίθεση με την συνάρτηση κατωφλίου που λαμβάνει τιμές 0 ή 1, η σιγμοειδής συνάρτηση μπορεί να λάβει τιμές σε ένα συνεχές πεδίο τιμών στο διάστημα 0 έως 1. Το γεγονός

αυτό την καθιστά διαφορίσιμη συγκριτικά με την συνάρτηση κατωφλίου. Αυτή είναι και η σημαντικότερη διαφορά τους. Σε κάποιες περιπτώσεις είναι απαραίτητο να μετατοπιστεί το πεδίο τιμών της συνάρτησης ενεργοποίησης στο εύρος -1 έως 1 , το οποίο την καθιστά μία περιττή συνάρτηση του τοπικού πεδίου. Για μία τέτοια μεταβολή του πεδίου τιμών η εξίσωση 2.3 μετατρέπεται στην εξίσωση 2.5.

$$\varphi(\hat{u}_l) = \begin{cases} 1 & \text{εάν } \hat{u}_l > 0 \\ 0 & \text{εάν } \hat{u}_l = 0 \\ -1 & \text{εάν } \hat{u}_l < 0 \end{cases} \quad (2.5)$$

Η προαναφερθείσα συνάρτηση συχνά συναντάται και ως συνάρτηση προσήμου. Κατά αναλογία η συνάρτηση της υπερβολικής εφαπτομένης 2.6 μπορεί να χρησιμοποιηθεί σαν αντικατάσταση της σιγμοειδούς συναρτήσης 2.4.

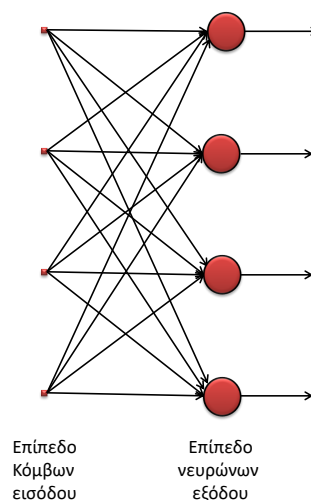
$$\varphi(\hat{u}_l) = \tanh(\hat{u}_l) \quad (2.6)$$

2.4 Αρχιτεκτονικές δικτύων

Η αρχιτεκτονική σχεδίασης ενός νευρωνικού δικτύου είναι καθοριστική για την διαδικασία μάθησης δεδομένου ότι δεν είναι ανεξάρτητες μεταξύ τους. Στις επόμενες ενότητες θα ασχοληθούμε με τις διαδικασίες μάθησης, ενώ σε αυτή την ενότητα θα περιγράψουμε μερικές από τις δομές δικτύων. Μπορούμε να διαχωρίσουμε τις αρχιτεκτονικές δικτύων σε τρία διαφορετικά είδη.

2.4.1 Δίκτυα Πρόσθιας Τροφοδότησης Ενός Επιπέδου

Στο πρώτο είδος αρχιτεκτονικής δικτύων, η ταξινόμηση των νευρώνων στα νευρωνικά δίκτυα γίνεται σε στιβάδες. Η πιο απλοϊκή δομή δικτύου αποτελείται από μία στιβάδα εισόδου, η οποία συνδέεται απευθείας σε μία στιβάδα νευρώνων εξόδου. Τέτοιου είδους δίκτυα αποκαλούνται Δίκτυα Πρόσθιας Τροφοδότησης (Feedforward Networks) και η αρχιτεκτονική τους φαίνεται στο σχήμα 5.

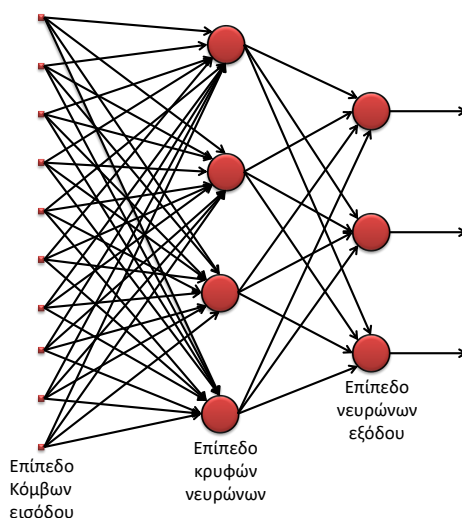


Εικόνα 5. Δίκτυο πρόσθιας τροφοδοσίας ενός επιπέδου (feedforward)

Το νευρωνικό δίκτυο του σχήματος αποτελείται από τέσσερις κόμβους εισόδου και τέσσερις εξόδου, ενώ το αποκαλούμε δίκτυο μιας στιβάδας επειδή έχει μία στιβάδα νευρώνων (στιβάδα των υπολογιστικών κόμβων)

2.4.2 Πολυεπίπεδα Δίκτυα Πρόσθιας Τροφοδότησης MLP

Τα πολυεπίπεδα νευρωνικά δίκτυα MLP (multi layer perceptron) αποτελούνται από μία ή περισσότερες κρυφές στιβάδες νευρώνων. Οι νευρώνες των κρυφών στιβάδων βρίσκονται ανάμεσα στην στοιβάδα εισόδου και την στοιβάδα εξόδου του νευρωνικού δικτύου. Ο αριθμός των στιβάδων του νευρωνικού δικτύου καθορίζει την τάξη των στατιστικών που εξάγει το δίκτυο. Στο σχήμα 6 παρουσιάζεται ένα δίκτυο MLP 10-4-3 μίας κρυφής στιβάδας.



Εικόνα 6. Δίκτυο πρόσθιας τροφοδότησης με ένα κρυφό επίπεδο

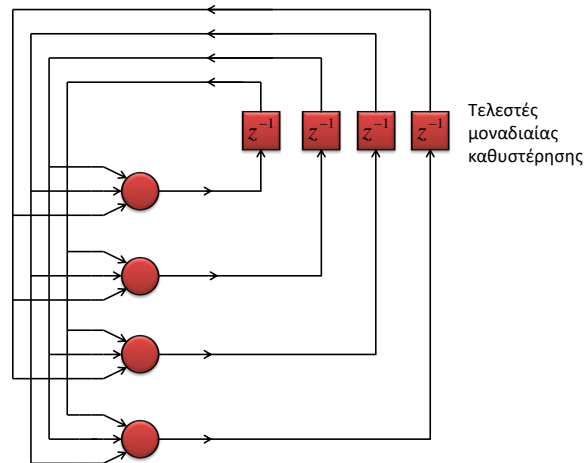
2.4.3 Αναδρομικά δίκτυα

Τα αναδρομικά νευρωνικά δίκτυα χαρακτηρίζονται από το γεγονός ότι αποτελούνται από τουλάχιστον έναν αναδραστικό βρόχο. Όπως φαίνεται στο σχήμα 7, ένα αναδρομικό δίκτυο μπορεί να αποτελείται από ένα πλήθος νευρώνων στο οποίο η έξοδος τους ανατροφοδοτεί την είσοδο όλων των υπολοίπων. Σύμφωνα με το σχήμα 7 η έξοδος των νευρώνων δεν ανατροφοδοτεί τον εαυτό τους. Μία ακόμη αρχιτεκτονική αναδρομικού δικτύου με δύο κρυφές στιβάδες παρατίθεται στο σχήμα 8.

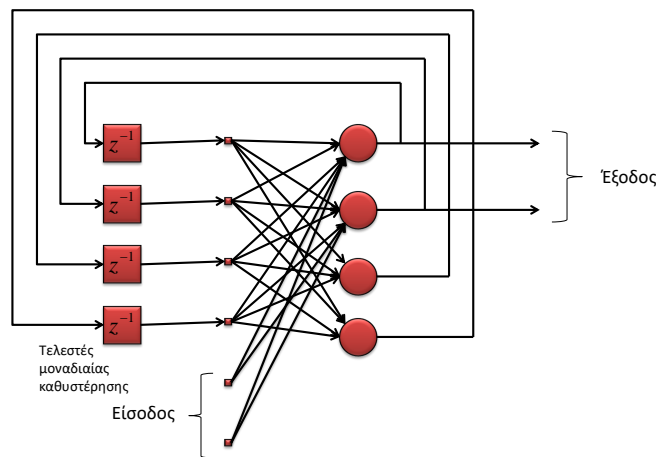
2.5 Διαδικασίες μάθησης

Οι διαδικασίες μάθησης είναι οι τεχνικές με τις οποίες ένα νευρωνικό δίκτυο αφομοιώνει «γνώση» Ορισμός:

Γνώση είναι η αποθηκευμένη πληροφορία ή τα μοντέλα που χρησιμοποιεί ένα άτομο ή μία μηχανή για να ερμηνεύσει, να προβλέψει και να αποκριθεί καταλλήλως στον έξω κόσμο.



Εικόνα 7. Αναδρομικό δίκτυο χωρίς αυτό-ανάδραση



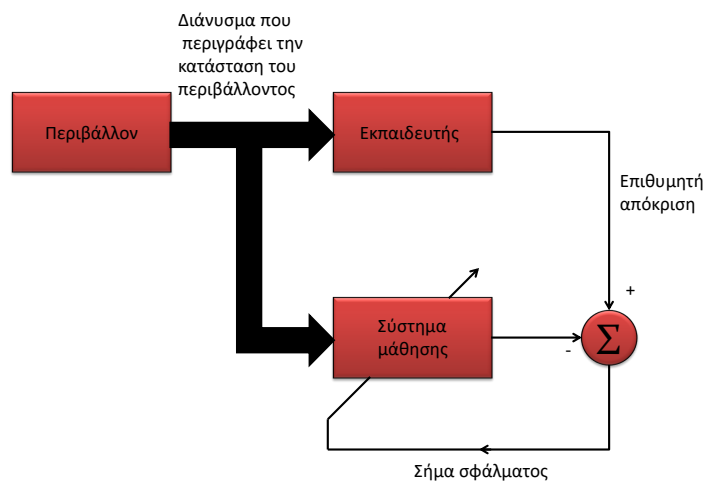
Εικόνα 8. Αναδρομικό δίκτυο με κρυφούς νευρώνες

Οι διαδικασίες μάθησης μπορούν να χωριστούν σε δύο μεγάλες κατηγορίες: μάθηση χωρίς εκπαιδευτή και μάθηση με εκπαιδευτή. Η μάθηση με εκπαιδευτή χωρίζεται περαιτέρω σε δύο υποκατηγορίες την μη επιβλεπόμενη μάθηση και την ενισχυτική μάθηση.

2.5.1 Μάθηση με εκπαιδευτή

Στην διαδικασία μάθησης με εκπαιδευτή, ή αλλιώς επιβλεπόμενη μάθηση, ένα πλήθος δεδομένων εισόδου - εξόδου (διάνυσμα εκπαίδευσης) που διεξάγονται από το περιβάλλον αντιπροσωπεύουν την γνώση του εκπαιδευτή. Όπως παρουσιάζεται στο σχηματικό διάγραμμα του σχήματος 9, ο εκπαιδευτής μέσω του διανύσματος εκπαίδευσης είναι σε θέση να διοχετεύσει την γνώση στο νευρωνικό δίκτυο με δεδομένο ότι γνωρίζει την επιθυμητή απόκριση. Η απόκλιση μεταξύ της πραγματικής εξόδου του δικτύου και της επιθυμητής απόκρισης ορίζουν το σφάλμα. Έτσι, τα συνα-

πτικά βάρη του δικτύου διορθώνονται με στόχο την ελαχιστοποίηση του σφάλματος. Η διαδικασία αυτή επαναλαμβάνεται (για όλα τα δεδομένα εκπαίδευσης) μέχρι το δίκτυο να προσομοιώνει την συμπεριφορά του εκπαιδευτή και στην συνέχεια να απαλλαγεί από αυτόν.



Εικόνα 9. Σχηματικό διάγραμμα της μάθησης με εκπαιδευτή

2.5.2 Μάθηση χωρίς Εκπαιδευτή

Στην μάθηση χωρίς εκπαιδευτή το νευρωνικό δίκτυο δεν χρησιμοποιεί μία συστάδα δεδομένων εισόδου - εξόδου, με τα οποία ένας εκπαιδευτής καλείται να εκπαιδεύσει το δίκτυο. Η μάθηση χωρίς εκπαιδευτή χωρίζεται σε δύο κατηγορίες: Ενισχυτική μάθηση και μη επιβλεπόμενη μάθηση.

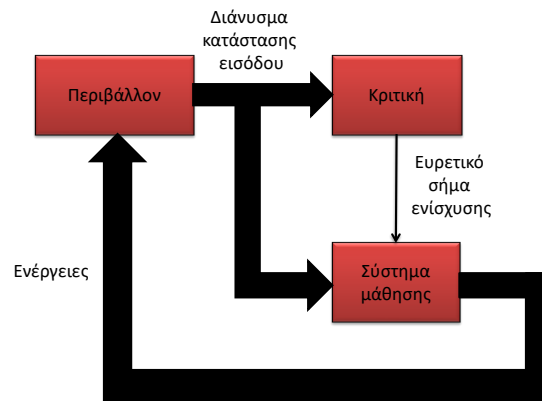
Ενισχυτική Μάθηση

Στην ενισχυτική μάθηση το νευρωνικό δίκτυο βρίσκεται σε συνεχής αλληλεπίδραση με το περιβάλλον. Με αυτόν τον τρόπο το σύστημα μάθησης μπορεί να εξάγει τα δεδομένα εισόδου - εξόδου με τα οποία εκπαιδεύεται το δίκτυο. Με κριτήριο τον βαθμό απόδοσης του δικτύου, το σύστημα μπορεί να προσαρμόσει τα συναπτικά βάρη του. Στην Αρχιτεκτονική που παρουσιάζεται στο σχήμα 10 μπορούμε να δούμε ότι το σύστημα περιλαμβάνει έναν κριτή. Στόχος του κριτή είναι η παραγωγή ενός ευρετικού σήματος ενίσχυσης λαμβάνοντας ένα σήμα από το περιβάλλον.

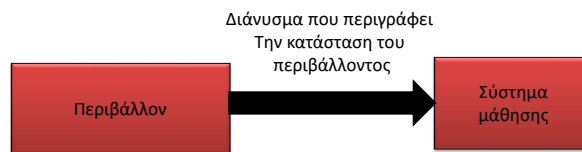
Μη επιβλεπόμενη μάθηση

Σε αντίθεση με τις προηγούμενες μεθόδους αυτή δεν περιλαμβάνει κανέναν εξωτερικό παράγοντα μάθησης όπως τον εκπαιδευτή ή τον κριτή. Αντιθέτως, ένα μέτρο ανεξάρτητο της αναπαράστασης χρησιμοποιείται για την εκπαίδευσή του δικτύου και τον καθορισμό των συναπτικών βαρών. Στο σχήμα 11 παρατίθεται το διάγραμμα της μη επιβλεπόμενης μάθησης.

Συχνά, η μη επιβλεπόμενη μάθηση εφαρμόζεται χρησιμοποιώντας μία μέθοδο ανταγωνιστικής μάθησης κατά την οποία οι νευρώνες ανταγωνίζονται μεταξύ τους σύμφωνα με έναν κανόνα.



Εικόνα 10. Σχηματικό διάγραμμα της ενισχυτικής μάθησης



Εικόνα 11. Σχηματικό διάγραμμα μη επιβλεπόμενης μάθησης

2.6 Νευρωνικά δίκτυα RBF

2.6.1 Αρχιτεκτονική και διαδικασία εκπαίδευσης

Τα νευρωνικά δίκτυα RBF (Radial Basis Function) μπορούν να προσεγγίσουν μη γραμμικά συστήματα και διακρίνονται για την απλότητα της αρχιτεκτονικής τους [11], [13], [29]. Αποτελούνται από μία μοναδική κρυφή στιβάδα, η οποία συνδέεται γραμμικά με την στιβάδα εξόδου. Η λιτή αρχιτεκτονική τους βοηθάει στην εύκολη υλοποίηση γρήγορων, αποτελεσματικών και απλών αλγόριθμων μάθησης.

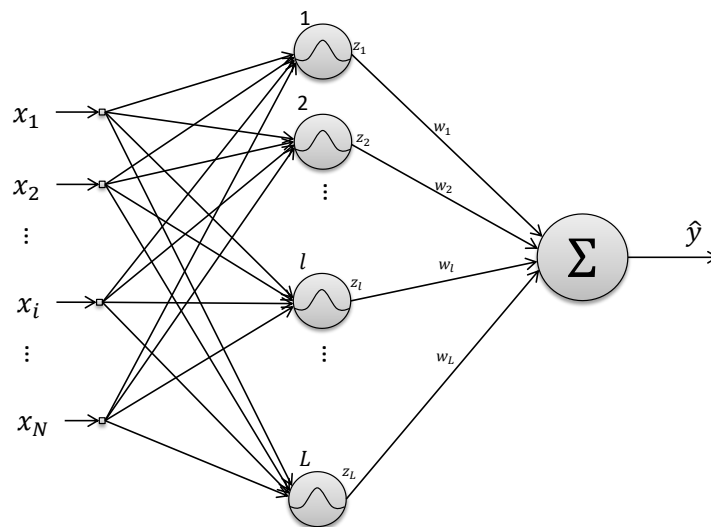
Τροφοδοτώντας ένα δίκτυο RBF με δεδομένα εισόδου - εξόδου, εκπαιδεύεται και προσεγγίζει την πολυεπίπεδη επιφάνεια της συνάρτησης που συνδέει την είσοδο με την έξοδο. Η συνάρτηση δομείται από ένα άθροισμα ακτινικών συναρτήσεων βάσης RBF που τοποθετούνται κατάλληλα

πάνω στην πολύ επίπεδη επιφάνεια, η οποία ορίζεται από τις μεταβλητές εισόδου του δικτύου .

Για να εκπαιδευτεί ένα νευρωνικό δίκτυο RBF χρειάζεται να υπολογιστούν οι παρακάτω παράμετροι :

- Το πλήθος των κέντρων των RBF
- Τα σημεία τοποθέτησης των RBF πάνω στην επιφάνεια που σχηματίζουν οι μεταβλητές εισόδου
- Το πλάτος των RBF (Ο υπολογισμός αυτός προϋποθέτει ότι έχει χρησιμοποιηθεί ακτινική συνάρτηση βάσης που περιέχει πλάτος ή διασπορά, δηλαδή συναρτήσεις τύπου Gaussian)
- Τα συναπτικά βάρη

Η επιλογή του πλήθους των κέντρων των RBF και τα σημεία που θα πρέπει να τοποθετηθούν, είναι μία ιδιαίτερα σημαντική αλλά και πολύπλοκη διαδικασία. Οι αλγόριθμοι που χρησιμοποιούνται για τον υπολογισμό αυτό ποικίλλουν. Μία πρώτη απλή σκέψη θα ήταν να ορίσουμε ένα κέντρο για κάθε δεδομένο εισόδου που λαμβάνει το δίκτυο. Όμως, για μεγάλους όγκους δεδομένων το υπολογιστικό κόστος για μία τέτοια διαδικασία θα ήταν ασύμφορο. Για να αντιμετωπιστεί αυτό το πρόβλημα τα δεδομένα εισόδου ομαδοποιούνται και έτσι ταυτίζεται ένα κέντρο για κάθε ομάδα δεδομένων. Με αυτόν τον τρόπο η πολυπλοκότητα, αλλά και η υπολογιστική απαίτηση του δικτύου μειώνεται χωρίς αυτό απαραίτητα να σημαίνει ότι μειώνεται και η απόδοση του. Η αναγκαιότητα του υπολογισμού των κέντρων είναι ένα από τα σημαντικά μειονεκτήματα των νευρωνικών δικτύων RBF. Μία τυπική δομή ενός νευρωνικού δικτύου RBF παρουσιάζεται στο σχήμα 12.



Εικόνα 12. Νευρωνικό δίκτυο RBF

Η στιβάδα εισόδου αποτελείται από N μεταβλητές και εν συνεχεία κάθε κόμβος της κρυφής στιβάδας συνδέεται με αυτές. Κάθε κόμβος της κρυφής στιβάδας αντιπροσωπεύει μία ακτινική συνάρτηση βάσης, τοποθετημένη σε ένα κέντρο μιας πολυδιάστατης επιφάνειας, που ορίζεται από το πλήθος των μεταβλητών εισόδου. Η κρυφή στιβάδα λαμβάνει τα δεδομένα εισόδου και τα μετασχηματίζει σε έναν χώρο υψηλότερης διάστασης. Προτού υλοποιηθεί ο υπολογισμός της ακτινικής συνάρτησης βάσης, υπολογίζεται πρώτα η ευκλείδεια νόρμα $\mu_l(\mathbf{x}(k))$ του k στοιχείου του διανύσματος εισόδου $\mathbf{x}(k)$ και του κέντρου που αντιστοιχεί ο εκάστοτε κόμβος u_l , βλέπε σχέση 2.7.

$$\mu_l(\mathbf{x}(k)) = \|\mathbf{x}(k) - \mathbf{u}_l\| = \sqrt{\sum_{i=1}^N (x_i(k) - u_{l,i})^2}, \quad k = 1, 2, \dots, K \quad (2.7)$$

Όπου το K υποδεικνύει το πλήθος των δεδομένων εκπαίδευσης, $\mathbf{x}^T(k) = [x_1(k), x_2(k), \dots, x_N(k)]$ είναι το διάνυσμα εισόδου και $\mathbf{u}_l = [u_{l,1}, u_{l,2}, \dots, u_{l,N}]$ οι συντεταγμένες του κέντρου κάθε κόμβου l . Η έξοδος του κάθε κόμβου z_l δίνεται από τη συνάρτηση ενεργοποίησης $g(\mu_l)$ η οποία είναι μία συνάρτηση ακτινικής συμμετρίας RBF. Η συνολική έξοδος των κόμβων όπως φαίνεται στην σχέση 2.8 δίνεται από ένα διάνυσμα $\mathbf{Z}(k)$ το οποίο περιλαμβάνει όλες τις εξόδους από τις RBF.

$$\mathbf{Z}(k) = [g(\mu_1(\mathbf{x}(k))), g(\mu_2(\mathbf{x}(k))), \dots, g(\mu_L(\mathbf{x}(k)))] \quad (2.8)$$

Η οριστική έξοδος του δικτύου υλοποιείται από ένα άθροισμα που αθροίζει όλες τις εξόδους των νευρώνων της κρυφής στιβάδας σύμφωνα με τη σχέση 2.9.

$$\hat{y}(k) = \mathbf{W}^T \times \mathbf{Z}(k) = \sum_{l=1}^L w_l = \sum_{l=1}^L w_l g(\mu_l(\mathbf{x}(k))) \quad (2.9)$$

Όπου $\mathbf{W}^T = [w_1, w_2, \dots, w_L]$ είναι ένα διάνυσμα που περιλαμβάνει τα συναπτικά βάρη του δικτύου. Μετά τον υπολογισμό των θέσεων και του πλήθους των κέντρων, γίνεται ο υπολογισμός των συναπτικών βαρών με γραμμική παλινδρόμηση της εξόδου της κρυφής στιβάδας και των επιθυμητών εξόδων των δεδομένων εκπαίδευσης. Ο υπολογισμός αυτός γίνεται με χρήση γραμμικών ελαχίστων τετραγώνων σύμφωνα με τη σχέση 2.10.

$$\mathbf{W} = (\mathbf{Z}^T \times \mathbf{Z})^{-1} \times \mathbf{Z}^T \times \mathbf{Y} \quad (2.10)$$

Όπου $\mathbf{W}^T = (w_1, w_2, \dots, w_L)$ περιλαμβάνει τις τιμές των συναπτικών βαρών που υπολογίστηκαν, το $\mathbf{Z}^T = [z(1), z(2), \dots, z(K)]$ περιλαμβάνει τις εξόδους της κρυφής στιβάδας για όλα τα δεδομένα εκπαίδευσης και $\mathbf{Y} = [y(1), y(2), \dots, y(K)]$ είναι το διάνυσμα με τις πραγματικές τιμές των εξόδων (Δεδομένα εκπαίδευσης). Ο όρος $(\mathbf{Z}^T \times \mathbf{Z})^{-1}$ είναι ο αντίστροφος του πίνακα συνδιακύμανσης και συμβολίζεται με \mathbf{P} . Η διαδικασία εκπαίδευσης του νευρωνικού δικτύου στην πραγματικότητα είναι ο υπολογισμός των συναπτικών βαρών, αλλά και ο υπολογισμός του πλήθους των RBF και των σημείων τοποθέτησης των κέντρων τους.

2.6.2 Ακτινικές συναρτήσεις βάσης

Οι ακτινικές συναρτήσεις βάσης (RBF) συχνά έχουν ένα ολικό μέγιστο ή ελάχιστο στο κέντρο τους \mathbf{u}_l και είτε αυξάνουν είτε ελαττώνουν όσο απομακρύνονται από αυτό. Η έξοδος ενός νευρώνα της κρυφής στιβάδας υπολογίζεται από μία RBF, της οποίας η είσοδος είναι η ευκλείδεια νόρμα του διανύσματος των δεδομένων εισόδου από τις συντεταγμένες του κέντρου του εκάστοτε κόμβου. Εκτός από το κέντρο πολλές συναρτήσεις βάσης χαρακτηρίζονται από μία άλλη παράμετρο που αποκαλείται πλάτος και συμβολίζεται με σ . Στην περίπτωση που το νευρωνικό δίκτυο χρησιμοποιεί μία RBF με χαρακτηριστικό πλάτος, τότε αυτό σημαίνει ότι θα πρέπει τα πλάτη των RBF

να συμπεριληφθούν στις μεταβλητές υπολογισμού των ελαχίστων τετραγώνων στο διάνυσμα \mathbf{W} της σχέσης 2.10. Στην συνέχεια, παρουσιάζονται ένα πλήθος από συναρτήσεις βάσης μία εκ των οποίων είναι η Gaussian την οποία συναντάμε και πιο συχνά στα δίκτυα RBF.

- Linear:

$$g(\|\mathbf{x}(k) - \mathbf{u}_l\|_2) = \|\mathbf{x}(k) - \mathbf{u}_l\|_2 \quad (2.11)$$

- Cubic:

$$g(\|\mathbf{x}(k) - \mathbf{u}_l\|_2) = \|\mathbf{x}(k) - \mathbf{u}_l\|_2^3 \quad (2.12)$$

- Thin-plate-spline:

$$g(\|\mathbf{x}(k) - \mathbf{u}_l\|_2) = \|\mathbf{x}(k) - \mathbf{u}_l\|_2^2 \ln(\|\mathbf{x}(k) - \mathbf{u}_l\|_2) \quad (2.13)$$

- Multiquadratic:

$$g(\|\mathbf{x}(k) - \mathbf{u}_l\|_2) = \sqrt{\|\mathbf{x}(k) - \mathbf{u}_l\|_2^2 + \sigma_L^2} \quad (2.14)$$

- Inverse multiquadratic:

$$g(\|\mathbf{x}(k) - \mathbf{u}_l\|_2) = \frac{1}{\sqrt{\|\mathbf{x}(k) - \mathbf{u}_l\|_2^2 + \sigma_L^2}} \quad (2.15)$$

- Gaussian:

$$g(\|\mathbf{x}(k) - \mathbf{u}_l\|_2) = \exp(-\|\mathbf{x}(k) - \mathbf{u}_l\|_2^2 / 2\sigma_L^2) \quad (2.16)$$

2.6.3 Αλγόριθμος ασαφούς λογικής

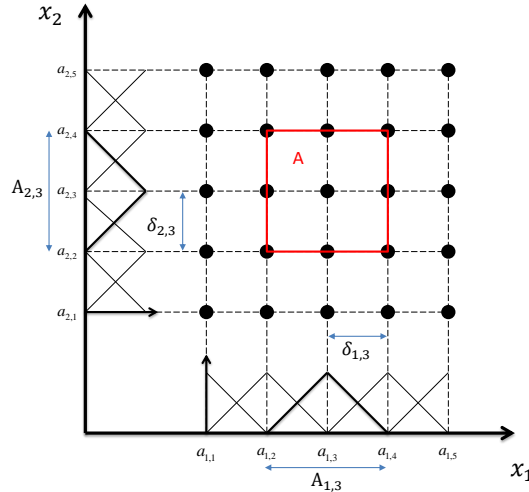
Υπάρχουν διάφοροι τρόποι που μπορεί να γίνει η επιλογή των κέντρων RBF [14], [15]. Ο αλγόριθμος ασαφούς λογικής είναι μία τέτοια διαδικασία και βρίσκει εφαρμογές σε πολλούς τομείς εκτός από την ομαδοποίηση δεδομένων σε νευρωνικά δίκτυα RBF [16]–[19], [30]–[36].

Αν υποθέσουμε ένα νευρωνικό δίκτυο το οποίο αποτελείται από x_i εισόδους όπου $i = 1, 2, \dots, N$. Και ότι οι τιμές που λαμβάνουν οι εισοδοί x_i κυμαίνονται στο διάστημα $x \in [-1, 1]$. Σύμφωνα με την ασαφή λογική [37] το πεδίο ορισμού των μεταβλητών εισόδου διαχωρίζεται σε ένα πλήθος από ασαφή σύνολα. Συγκεκριμένα η διαδικασία διαμερίζει τις μεταβλητές εισόδου x_i σε c_i ασαφή τριγωνικά ισομερή υποσύνολα $\mathbf{A}_{i,j}$ όπου $j = 1, 2, \dots, c_i$. Τα υποσύνολα τα οποία δημιουργούνται για την διάσταση i μπορούν να γραφούν πιο ξεκάθαρα σε μορφή διανύσματος $\mathbf{T}_i = [\mathbf{A}_{i,1}, \mathbf{A}_{i,1}, \dots, \mathbf{A}_{i,c_i}]$ με \mathbf{T}_i να είναι το σύνολο των ασαφών συνόλων για την διάσταση i . Κάθε ασαφές υποσύνολο $\mathbf{A}_{i,j}$ διακρίνεται από τα χαρακτηριστικά $\alpha_{i,j}$ και $\delta_{i,j}$ όπου $\alpha_{i,j}$ είναι το κέντρο του και $\delta_{i,j}$ το μισό του πλάτους του ασαφούς τριγώνου. Κάθε υποσύνολο $\mathbf{A}_{i,j}$ μπορεί να γραφεί στην μορφή $\mathbf{A}_{i,j} = [\alpha_{i,j}, \delta\alpha_{i,j}]$.

Κατά την άφιξη ενός νέου δεδομένου $\mathbf{x}(k)$ ένα ασαφές υποσύνολο $\mathbf{A}_{i,j}$ ενεργοποιείται από το \mathbf{T}_i για κάθε διάσταση i , σχηματίζοντας έναν ασαφή υπόχωρο \mathbf{A}^l όπου l ανήκει στο διάστημα $1 \leq l \leq C$ και C να είναι το πλήθος των ασαφών υπόχωρων. Το πλήθος C δίνεται από το γινόμενο του πλήθους των ασαφών συνόλων ανά μεταβλητή εισόδου όπως περιγράφεται από την σχέση 2.17.

$$C = \prod_i^N c_i \quad (2.17)$$

Στο σχήμα 13 απεικονίζεται ένας ασαφής υπόχωρος \mathbf{A}^l για δύο μεταβλητές εισόδου με $N = 2$ και $c_1 = c_2 = 5$.



Εικόνα 13. Διαδικασία κατάτμησης σε ασαφή υποσύνολα

Κατά την άφιξη ενός δεδομένου $\mathbf{x}(k)$ ενεργοποιείται ένα ασαφές υποσύνολο για κάθε μεταβλητή εισόδου, προκειμένου να προσδιοριστεί ο πλησιέστερος ασαφής υπόχωρος για το δεδομένο $\mathbf{x}(k)$. Η επιλογή του ασαφούς υποσυνόλου που ενεργοποιείται για την κάθε μεταβλητή i δίνεται από την συνάρτηση συμμετοχής $\mu_{\mathbf{A}_{i,j}}(x_i(k))$ όπως περιγράφεται από την σχέση 2.18.

$$\mu_{\mathbf{A}_{i,j}}(x_i(k)) = \begin{cases} 1 - \frac{|x_i(k) - a_{i,j}|}{\delta a_{i,j}} & \text{εάν } x_i(k) \in [a_{i,j} - \delta a_{i,j}, a_{i,j} + \delta a_{i,j}] \\ 0 & \text{Οπουδήποτε αλλού} \end{cases} \quad (2.18)$$

Σύμφωνα με την σχέση 2.18 για κάθε δεδομένο εισόδου $x_i(k)$ που ανήκει στο διάστημα $[a_{i,j} - \delta a_{i,j}, a_{i,j} + \delta a_{i,j}]$ ενεργοποιεί το ασαφές υποσύνολο j ποσοστιαία από τη σχέση $1 - \frac{|x_i(k) - a_{i,j}|}{\delta a_{i,j}}$. Το υποσύνολο j με το μεγαλύτερο ποσοστό ενεργοποίησης είναι και αυτό που τελικά θα χρησιμοποιηθεί για τον προσδιορισμό του υπόχωρου. Για όλα τα άλλα ασαφή υποσύνολα j της συγκεκριμένης μεταβλητής i το ποσοστό ενεργοποίησης θα είναι 0 αφού το δεδομένο $x_i(k)$ δεν βρίσκεται στο διάστημα $[a_{i,j} - \delta a_{i,j}, a_{i,j} + \delta a_{i,j}]$ για αυτά.

Ο υπόχωρος \mathbf{A}^l προκύπτει από τον συνδυασμό N υποσυνόλων $\mathbf{A}_{1,j_1}^l \in \mathbf{T}_1, \mathbf{A}_{2,j_2}^l \in \mathbf{T}_2, \dots, \mathbf{A}_{N,j_N}^l \in \mathbf{T}_N$ όπου j_1, j_2, \dots, j_N ανήκουν στο διάστημα $1 \leq j_1 \leq c_1, 1 \leq j_2 \leq c_2, \dots, 1 \leq j_N \leq c_N$ αντίστοιχα και διατυπώνεται σε μορφή πίνακα όπως φαίνεται στην σχέση 2.19.

$$\mathbf{A}^l = [\mathbf{A}_{1,j_1}^l, \mathbf{A}_{2,j_2}^l, \dots, \mathbf{A}_{N,j_N}^l] = \{[\alpha_{1,j_1}^l, \alpha_{2,j_2}^l, \dots, \alpha_{N,j_N}^l], [\delta a_1, \delta a_2, \dots, \delta a_N]\} \quad (2.19)$$

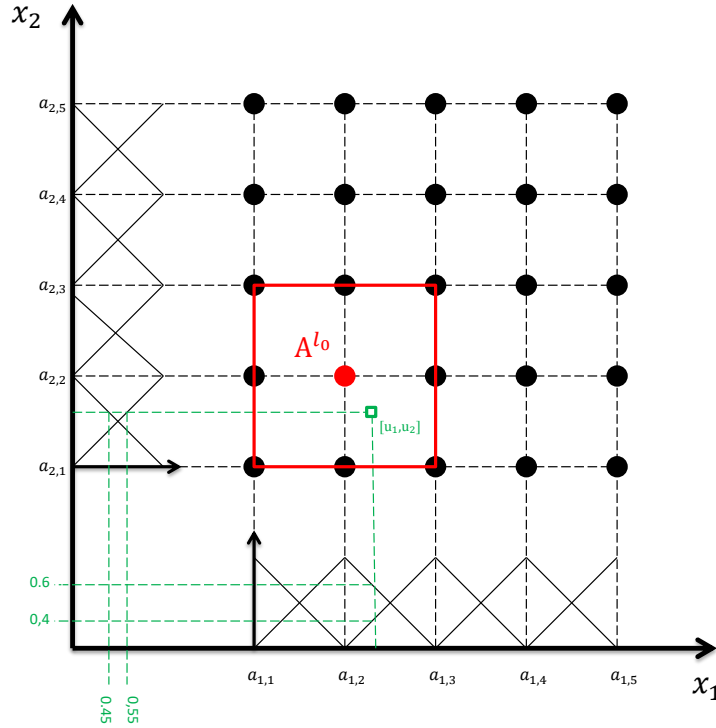
Ή αλλιώς θα μπορούσε να γραφεί πιο απλά στην μορφή $\mathbf{A}^l = [\mathbf{a}^l, \delta \mathbf{a}]$ όπου $\mathbf{a}^l = [\alpha_{1,j_1}^l, \alpha_{2,j_2}^l, \dots, \alpha_{N,j_N}^l]$ είναι το διάνυσμα των κέντρων, ενώ $\delta \mathbf{a} = [\delta a_1, \delta a_2, \dots, \delta a_N]$ είναι το μισό του πλάτους του κάθε ασαφούς υποσυνόλου αντίστοιχα. Δεδομένου ότι τα κέντρα των ασαφών συνόλων έχουν τοποθετηθεί με τέτοιο τρόπο έτσι ώστε να ισαπέχουν, η παράμετρος δa_i είναι ίδια για όλα τα κέντρα $\delta a_1 = \delta a_2 = \delta a_i = \delta a_N$. Με αυτό το δεδομένο μπορούμε να επεκτείνουμε την συνάρτηση συμμετοχής σε πολλές διαστάσεις, εισάγοντας την πολυδιάστατη συνάρτηση $\mu_{\mathbf{A}^l}(\mathbf{x}(k))$ που περιγράφεται από την σχέση 2.20.

$$\mu_{\mathbf{A}^l}(\mathbf{x}(k)) = \begin{cases} 1 - rd^l(\mathbf{x}(k)) & \text{εάν } rd^l(\mathbf{x}(k)) \leq 1 \\ 0 & \text{Οπουδήποτε αλλού} \end{cases} \quad (2.20)$$

Όπου $rd^l(\mathbf{x}(k))$ υποδεικνύει την σχετική ευκλείδεια απόσταση μεταξύ του κέντρου του \mathbf{A}^l (δηλαδή το διάνυσμα \mathbf{a}^l) και του διανύσματος εισόδου $\mathbf{x}(k)$ από τη σχέση 2.21.

$$rd^l(\mathbf{x}(k)) = \frac{\sqrt{[\sum_{i=1}^N (\alpha_{i,j_i}^l - x_i(k))^2]}}{\sqrt{\sum_{i=1}^N (\delta\alpha_i)^2}} \quad (2.21)$$

Με την πολυδιάστατη συνάρτηση συμμετοχής μας δίνεται η δυνατότητα να υπολογίσουμε την σχετική ευκλείδεια απόσταση του διανύσματος εισόδου $\mathbf{x}(k)$ από τα κέντρα των ασαφών υπόχωρων \mathbf{A}^l .



Εικόνα 14. Κατηγοριοποίηση δεδομένου εισόδου στα ασαφή υπόχωρο

Ο υπολογισμός αυτός μπορεί να γίνει για κάθε μεταβλητή του διανύσματος εισόδου $\mathbf{x}^T(k) = [x_1(k), x_2(k), \dots, x_N(k)]$ και με αυτόν τον τρόπο να βρεθεί ο πλησιέστερος υπόχωρος στο διάνυσμα $\mathbf{x}(k)$. Ο πλησιέστερος υπόχωρος \mathbf{A}^{l_0} που σχηματίζεται από τα ασαφή υποσύνολα με τον μεγαλύτερο βαθμό ενεργοποίησης (δηλαδή την μικρότερη απόσταση από το εκάστοτε δεδομένο) δίνεται από το **κριτήριο της ελάχιστης απόστασης** που περιγράφεται σύμφωνα με την σχέση 2.22.

$$j_i = \arg \max_{1 \leq j \leq c_i} (\mu_{\mathbf{A}_{i,j}}(x_i(k))) \quad (2.22)$$

Η σχέση 2.22 υπολογίζει τον δείκτη j_i του πλησιέστερου ασαφούς υποσυνόλου για την διάσταση i του διανύσματος $\mathbf{x}(k) = [x_1(k), x_2(k), \dots, x_i(k)]$. Ο υπολογισμός αυτός μπορεί να επαναληφθεί για κάθε διάσταση i του διανύσματος εισόδου $\mathbf{x}(k)$ και έτσι να βρεθεί ο πλησιέστερος ασαφής υπόχωρος $\mathbf{A}^{l_0} = [\mathbf{A}_{1,j_1}^{l_0}, \mathbf{A}_{2,j_2}^{l_0}, \dots, \mathbf{A}_{N,j_N}^{l_0}]$ ο οποίος αντιστοιχεί στους δείκτες j_i όπου $\mathbf{A}_{i,j_i}^{l_0} \in \mathbf{T}_i$ για $i = 1, 2, \dots, N$. Στο σχήμα 14 μπορούμε να διακρίνουμε τον τρόπο με τον οποίο

ένα δεδομένο εισόδου του διανύσματος $\mathbf{x}^T = [x_1, x_2]$ ενεργοποιεί τα ασαφή τριγωνικά υποσύνολα (με τα αριθμητικά ποσοστά ενεργοποίησης) και να αποδοθεί στον πλησιέστερο ασαφή υπόχωρο. $\mathbf{A}^l = [\mathbf{A}_{1,2}, \mathbf{A}_{2,2}] = \{[a_{1,2}, a_{2,2}], [\delta a_1, \delta a_2]\}$. Με το κριτήριο της ελάχιστης απόστασης μπορούμε να εντοπίσουμε τον υπόχωρο με την ελάχιστη απόσταση ανάμεσα σε ένα C πλήθος υπαρκτών υπόχωρων για κάθε εισερχόμενο δεδομένο k του διανύσματος εισόδου $\mathbf{x}(k)$.

Κεφάλαιο 3

Online εκπαίδευση

3.1 Εφαρμογές και πλεονεκτήματα

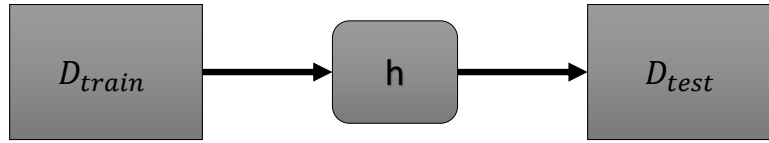
Στις μέρες μας εταιρείες όπως η Google, Amazon, Facebook και πολλές ακόμα διακινούν και διαχειρίζονται μεγάλους όγκους ψηφιοποιημένων δεδομένων κάθε μορφής. Η διακίνηση και η επεξεργασία των δεδομένων είναι απαραίτητη και σε άλλες εφαρμογές όπως είναι τα έξυπνα ρολόγια ή το GPS (Global Positioning System). Επιπρόσθετα διαχείριση και δημιουργία μεγάλων όγκων δεδομένων πραγματοποιείται ακόμα και σε επιστημονικές εφαρμογές, όπως αστροπαρατήρηση, δορυφορικές εφαρμογές αλλά και σε αναρίθμητες εφαρμογές σε διαφορετικούς τομείς. Όλες οι προαναφερθέντες τεχνολογίες λαμβάνουν δεδομένα μεγάλου όγκου γνωστά και ως "Big Data" με τεράστιους ρυθμούς, τα οποία δεδομένα τείνουν διαρκώς να αυξάνονται [1], [2]. Για τα συστήματα μηχανικής μάθησης αποτελεί πρόκληση η επεξεργασία σε πραγματικό χρόνο και η δημιουργία μοντέλων. Τα παραδοσιακά μοντέλα μηχανικής μάθησης στα οποία η επεξεργασία των δεδομένων γίνεται μαζικά (batch learning), δεν αποδίδουν επαρκώς στον απαιτούμενο χρόνο. Και αυτό έχει ως αποτέλεσμα το μοντέλο να μην ενημερώνεται για τις εξελίξεις των δεδομένων στο απαιτούμενο χρονικό διάστημα. Για να ξεπεραστούν τα συγκεκριμένα εμπόδια δημιουργήθηκαν οι on-line αλγόριθμοι μηχανικής μάθησης που συχνά αποκαλούνται και σταδιακοί αλγόριθμοι (incremental learning). Οι on-line αλγόριθμοι είναι σε θέση να εκπαιδεύουν το μοντέλο κατευθείαν για κάθε νέο δεδομένο που είναι διαθέσιμο. Δηλαδή διατηρούν το μοντέλο ενημερωμένο, αλλά ταυτόχρονα εξοικονομούν χώρο από δεδομένα, τα οποία στην περίπτωση της μαζικής εκπαίδευσης θα έπρεπε να αποθηκευτούν. Αυτά τα πλεονεκτήματα έκαναν τις on-line διαδικασίες να κερδίσουν την προσοχή του τομέα των μεγάλων δεδομένων, ανοίγοντας ένα νέο κεφάλαιο στην επιστήμη των νευρωνικών δικτύων.

Ορίζουμε ότι στην online μάθηση για μία σειρά δεδομένων εκπαίδευσης $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_K$ δημιουργούνται h_1, h_2, \dots, h_K μοντέλα όπου $\mathbf{s}_k = (\mathbf{x}(k), y(k)) \in R^n \times \{1, 2, \dots, C\}$ είναι τα δεδομένα εκπαίδευσης και $h_k : R^n \times \{1, 2, \dots, C\}$ είναι το νέο παραγόμενο μοντέλο αποκλειστικά από το μοντέλο h_{k-1} και τα p είναι τα πρόσφατα δεδομένα εκπαίδευσης $\mathbf{s}_k, \dots, \mathbf{s}_{k-p}$ με το p να περιορίζεται αυστηρά. Ένας αλγόριθμος σταδιακής μάθησης θα πρέπει να διακρίνεται από τα εξής:

- Το μοντέλο h_k πρέπει να εκπαιδεύεται σταδιακά κάθε φορά από το προηγούμενο μοντέλο h_{k-1} χωρίς να γίνεται ολοκληρωτική επανεκπαίδευση.
- Να διατηρεί την απαραίτητη προηγούμενη γνώση χωρίς να ξεχνάει σε βαθμό που είναι ζημιογόνο.
- Να αποθηκεύει και να διαχειρίζεται έναν πεπερασμένο αριθμό προηγούμενων δεδομένων p .

3.2 Off-line εκπαίδευση

Στην Off-line εκπαίδευση ο αλγόριθμος δημιουργεί ένα μοντέλο h από έναν όγκο δεδομένων $D_{train} = \{(\mathbf{x}(k), y(k)) | k \in \{1, 2, \dots, K_{train}\}\}$. Όταν η διαδικασία εκπαίδευσης ολοκληρωθεί το μοντέλο δοκιμάζεται από ένα άλλο σύνολο δεδομένων $D_{test} = \{(\mathbf{x}(k), y(k)) | k \in \{1, 2, \dots, K_{test}\}\}$. Το σχήμα 15 παρουσιάζει την διαδικασία μάθησης.



Εικόνα 15. Μοντέλο Off-line μαζικής εκπαίδευσης

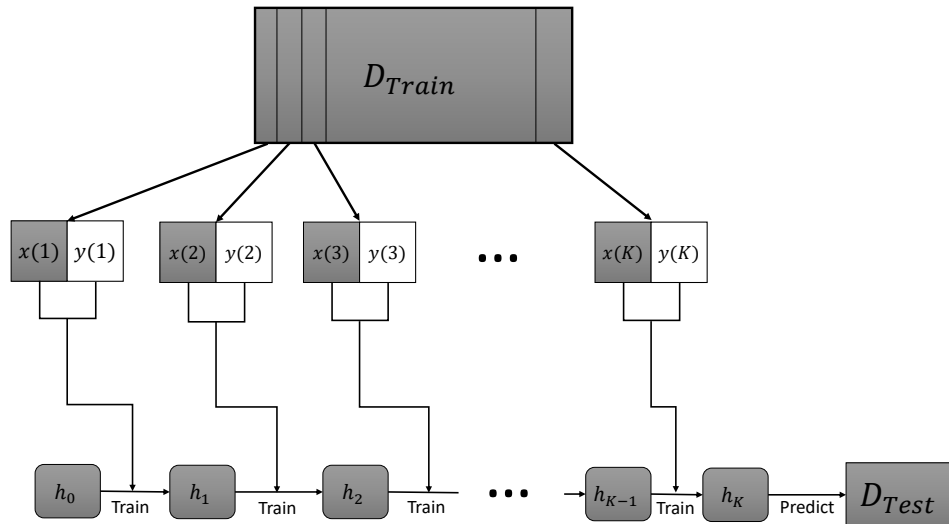
Για κάθε ένα δεδομένο $\mathbf{x}(k) \in D_{test}$ το μοντέλο προβλέπει την έξοδο $\hat{y}(k) = h(\mathbf{x}(k))$ και την απώλεια, βλέπε σχέση 3.1

$$L(\hat{y}(k), y(k)) = \begin{cases} 1 & \text{εάν } \hat{y}(k) \neq y(k) \\ 0 & \text{εάν } \hat{y}(k) = y(k) \end{cases} \quad (3.1)$$

Όπου $\hat{y}(k)$ είναι η πρόβλεψη του μοντέλου $h(\bullet)$ για το δεδομένο $\mathbf{x}(k)$ ενώ $y(k)$ είναι η πραγματική τιμή που αντιστοιχεί στα δεδομένα $\mathbf{x}(k)$. Ο υπολογισμός της ακρίβειας τελικά για το μοντέλο h υπολογίζεται σύμφωνα με την σχέση 3.2.

$$E(D_{Test}) = \frac{1}{K} \sum_{k=1}^K (1 - L(h_k(\mathbf{x}(k)), y(k))) \quad (3.2)$$

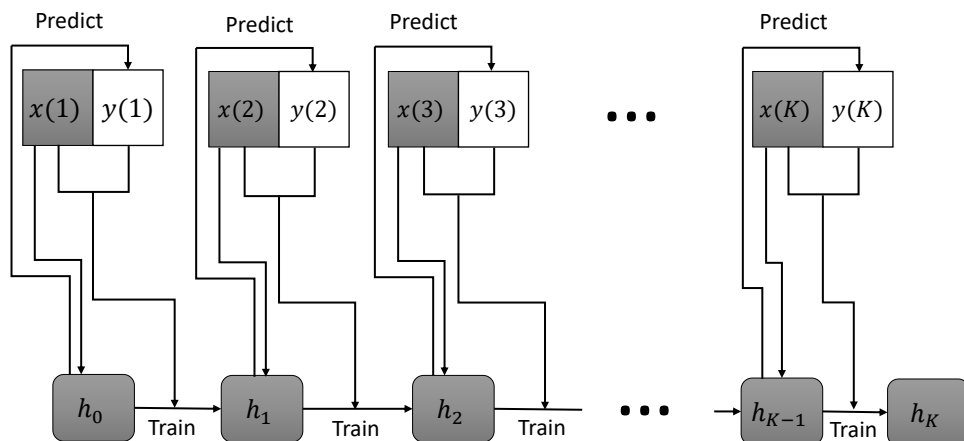
Ο υπολογισμός της ακρίβειας του μοντέλου για το πλήθος των μοντέλων που έχουν δημιουργηθεί μέχρι την χρονική στιγμή K υπολογίζεται από την σχέση 3.2 Εκτός από τους αλγόριθμους μαζικής εκπαίδευσης που προσπελάσουν όλα τα δεδομένα εκπαίδευσης απευθείας, υπάρχουν και οι αλγόριθμοι σταδιακής εκπαίδευσης οι οποίοι προσπελάσουν τα δεδομένα σειριακά το ένα μετά το άλλο. Το μοντέλο του σχήματος 16 δείχνει ότι ο αλγόριθμος για κάθε νέο δεδομένο $(\mathbf{x}(1), y(1)), (\mathbf{x}(2), y(2)), \dots, (\mathbf{x}(K), y(K))$ δημιουργεί το αντίστοιχο μοντέλο h_1, h_2, \dots, h_K , του οποίου το κάθε μοντέλο h_k προκύπτει αποκλειστικά από το προηγούμενο h_{k-1} και από το πλήθος των πεπερασμένων δεδομένων p . Όταν η διαδικασία εκπαίδευσης ολοκληρωθεί μόνο το τελευταίο μοντέλο $h(K)$ χρησιμοποιείται για να πρόβλεψει τα δεδομένα του D_{Test} , έτσι ώστε να καθοριστεί η απόδοση του μοντέλου.



Εικόνα 16. Μοντέλο Off-line σταδιακής εκπαίδευσης

3.3 On-line εκπαίδευση

Το μοντέλο της online εκπαίδευσης συνηθίζεται να χρησιμοποιείται σε δεδομένα που έχουν συνεχόμενη ροή. Όπως φαίνεται και στο σχήμα 17 τα δεδομένα παρέχονται στο σύστημα το ένα μετά το άλλο $S = (s_1, s_2, \dots, s_k, \dots, s_K)$ όπου $s_k = (\mathbf{x}(k), y(k))$, ενώ το k αντιστοιχεί στην τρέχουσα χρονική στιγμή. Το μοντέλο εκπαίδευσης παρέχει την δυνατότητα συνεχόμενης και πιθανών ατελείωτης τροφοδοσίας δεδομένων.



Εικόνα 17. Μοντέλο On-line εκπαίδευσης

Για κάθε μία πλειάδα δεδομένων $\mathbf{s}_k = [\mathbf{x}(k), y(k)]$ το μοντέλο $h_{k-1}(\bullet)$ καλείται να προβλέψει την άγνωστη έξοδο $y(k)$ λαμβάνοντας το $x(k)$. Όταν η πρόβλεψη $\hat{y}(k) = h_{k-1}(x(k))$ ολοκληρωθεί η πραγματική ταυτότητα του $y(k)$ αποκαλύπτεται και η απόδοση του δικτύου υπολογίζεται σύμφωνα με την σχέση 3.1. Στην συνέχεια το μοντέλο εκπαιδεύεται για τα δεδομένα \mathbf{s}_k και προκύπτει το νέο μοντέλο h_k από το προηγούμενο h_{k-1} . Ο υπολογισμός της ακρίβειας του μοντέλου για το πλήθος των μοντέλων που έχουν δημιουργηθεί μέχρι την δεδομένη χρονική στιγμή K υπολογίζεται σύμφωνα με την σχέση 3.3.

$$E(S) = \frac{1}{K} \sum_{k=1}^K (1 - L(h_{k-1}(\mathbf{x}(k)), y(k))) \quad (3.3)$$

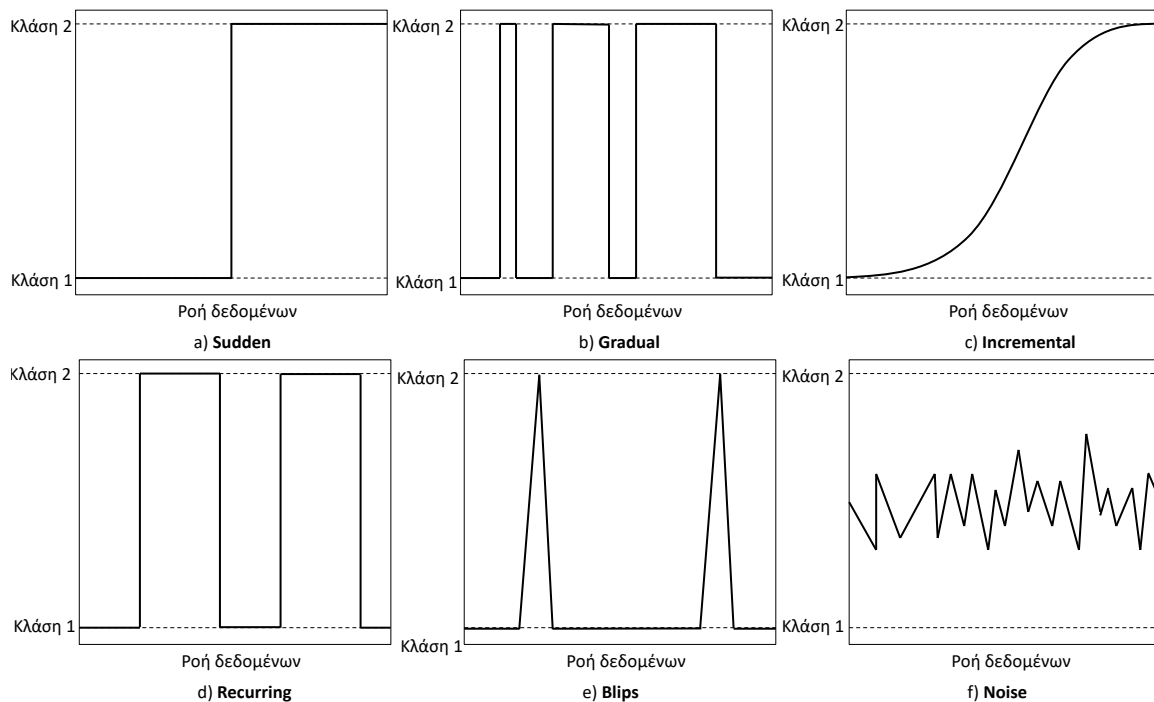
Η βασική διαφορά της εξίσωσης 3.3 από την εξίσωση 3.2 είναι ότι στον υπολογισμό της ακρίβειας προσμετρά και την επιτυχία όλων των προηγούμενων μοντέλων h_{k-1} που έχουν μεσολαβήσει μέχρι την χρονική στιγμή K .

3.4 Χρονικά μεταβαλλόμενα σύνολα δεδομένων

Έχουμε ήδη αναφέρει ότι στο μοντέλο της online εκπαίδευσης τα δεδομένα παρέχονται στο σύστημα σειριακά στην μορφή $S = (s_1, s_2, \dots, s_k, \dots, s_K)$ όπου το s_k έχει δημιουργηθεί από μία κατανομή D_k [4]. Για μία μη χρονικά μεταβαλλόμενη κατανομή στην οποία ισχύει $D_k = D_{k+1}$, θα έχουμε μία μετάβαση $S_k \rightarrow S_{k+1}$. Τα προβλήματα που αντιμετωπίζουμε στον πραγματικό κόσμο είναι εγγενώς μεταβαλλόμενης φύσης. Σε αυτά τα προβλήματα η κατανομή των δεδομένων είναι και αυτή χρονικά μεταβαλλόμενη και σε αυτή την περίπτωση ισχύει ότι $D_k \neq D_{k+1}$. Η μεταβολή ενός προβλήματος κατά την διαδικασία εκπαίδευσης μπορεί να συνδεθεί άμεσα με την πτώση της απόδοσης του δικτύου στην πρόβλεψη των κλάσεων. Για αυτό το λόγο η γνώση της συμπεριφοράς της μεταβολής είναι μία σημαντική πληροφορία για την εκπαιδευτική διαδικασία. Στην συνέχεια αναφερόμαστε αναλυτικότερα στις μορφές μεταβολής κατανομής των δεδομένων, τις οποίες συχνά συναντάμε σε πραγματικά ή και τεχνητά προβλήματα.

- **Sudden:** Στη μεταβολή τύπου Sudden το πρόβλημα χαρακτηρίζεται από μία ροή s_k που ακαριαία μεταβάλλεται σε μία νέα ροή s_{k+1} για την οποία ισχύει ότι $D_k \neq D_{k+1}$.
- **Gradual:** Σε αυτού του τύπου την μεταβολή μπορούμε να θεωρήσουμε ότι η ροή s_{k+1} μπορεί να προκύψει από την κατανομή D_k ή από την D_{k+1} , ενώ ισχύει ότι $D_k \neq D_{k+1}$.
- **Incremental:** Σε αυτού του είδους την μεταβολή η κατανομή D_k μεταβάλλεται στην D_{k+1} σταδιακά χωρίς απότομες μεταβολές.
- **Recurring:** Στη μεταβολή "Recurring" η κατανομή D_{k+1} αναμένουμε να επαναληφθεί είτε με σταθερή περίοδο είτε για μία φορά μετά από συγκεκριμένο αριθμό επαναλήψεων μίας προηγούμενης κατανομής D_{k-j} . Και έτσι προκύπτει ότι $D_{k+1} = D_{k-j}$ όπου j ο αριθμός των επαναλήψεων.
- **Blips:** Πρόκειται για μία μεταβολή εντελώς τυχαία, η οποία θα ήταν ιδανικό να αγνοηθεί κατά την διαδικασία εκπαίδευσης.
- **Noise:** Αυτή η μεταβολή είναι γνωστή και ως "λευκός θόρυβος" που χαρακτηρίζεται από μικρές διακυμάνσεις της κατανομής. Θα έπρεπε ιδανικά, επίσης, να φιλτράρονται κατά τη διαδικασία της εκπαίδευσης.
- **Mixed:** Αυτού του είδους η μεταβολή μπορεί να συνδυάζει δύο ή περισσότερες από τις παραπάνω μορφές μεταβολής.

Στο σχήμα 18 παρατίθενται οι τρόποι με τους οποίους μεταβάλλεται ο κάθε τύπος μεταβολής που αναφέραμε παραπάνω.



Εικόνα 18. Είδη μεταβολής συνόλων δεδομένων.

3.5 On-line εκπαίδευση νευρωνικών δικτύων RBF

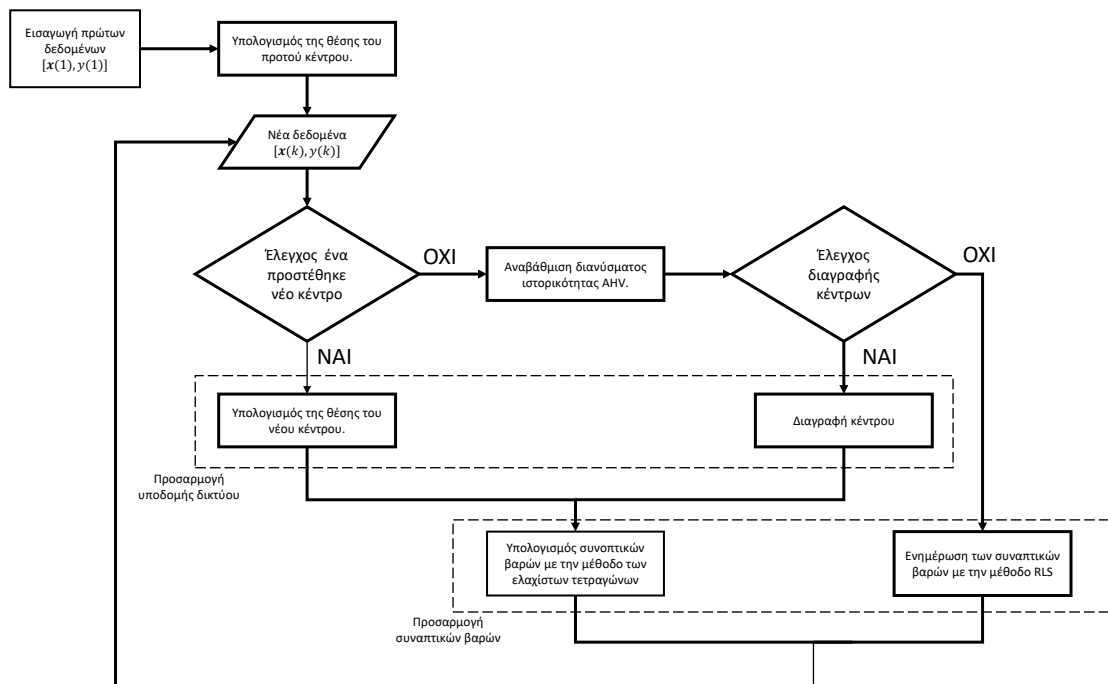
3.5.1 Αλγόριθμος εκπαίδευσης με χρήση ασαφών μέσων

Τα νευρωνικά δίκτυα RBF λόγω της απλότητας της δομής τους, αλλά και της ταχύτητας με την οποία εκπαιδεύονται, είναι μία αρχιτεκτονική με πολλά πλεονεκτήματα για on-line υλοποίησης. Σε αντίθεση με άλλες αρχιτεκτονικές που υπολογίζουν τις παραμέτρους του δικτύου σε ένα στάδιο, στα δίκτυα RBF τα στάδια εκπαίδευσης μπορούν να χωριστούν σε δύο φάσεις. Την φάση του υπολογισμού των κέντρων των ακτινικών συναρτήσεων βάσης και, κατόπιν, τον υπολογισμό των συναπτικών βαρών της κρυφής στιβάδας με την στιβάδα εξόδου, αυξάνοντας έτσι την ταχύτητα της διαδικασίας εκπαίδευσης [21]. Τα δίκτυα RBF μπορούν να χρησιμοποιηθούν για απευθείας εκπαίδευση από έναν όγκο δεδομένων με Μαζική Προσπέλαση (batch learning) ή εκπαίδευση από συνεχόμενη ροή δεδομένων με Σειριακή Προσπέλαση (on-line) για χρονικά μεταβαλλόμενα συστήματα. Σε αυτή την ενότητα θα δούμε μία υλοποίηση κατά την οποία το νευρωνικό δίκτυο εκπαιδεύεται on-line για προσέγγιση χρονικά μεταβαλλόμενων μη γραμμικών συστημάτων και για λόγους ευκολίας θα αποκαλούμε την συγκεκριμένη διαδικασία με το όνομα FUZZY-LS-RLS για το υπόλοιπο του συγγράμματος [21].

Αναλυτικότερα, ο αλγόριθμος FUZZY-LS-RLS κατακερματίζει τις διαστάσεις εισόδου σε ασαφή υποσύνολα, προκειμένου να υπολογίσει τα κέντρα των ακτινικών συναρτήσεων βάσης, σύμφωνα με την μέθοδο της ασαφούς λογικής, όπως περιγράφηκε στην ενότητα 2.6.3. Σε αντίθεση με την προσέγγιση της ενότητας 2.6.3, στην οποία τα δεδομένα προσπελάσσονται μαζικά και ο υπολογισμός των κέντρων γίνεται για όλο το σύνολο των δεδομένων, σε αυτή την υλοποίηση τα δεδομένα λαμβάνονται σειριακά το ένα μετά το άλλο και ανατίθενται στον πλησιέστερο ασαφή

υπόχωρο κατά την άφιξή τους. Στην περίπτωση που κανένας από τους ήδη δεσμευμένους υπόχωρους δεν είναι αρκετά κοντά σε ένα νέο δεδομένο, ο αλγόριθμος δεσμεύει ένα νέο υπόχωρο στον οποίο ανατίθεται το συγκεκριμένο δεδομένο σαν ένα νέο κέντρο RBF. Προκειμένου να αποφευχθεί η πιθανότητα το νευρωνικό δίκτυο να αυξάνει ασταμάτητα τους νευρώνες του, έχει προστεθεί μία διαδικασία διαγραφής νευρών. Κατά την συγκεκριμένη διαδικασία αν σε έναν δεσμευμένο ασαφή υπόχωρο στον οποίο δεν έχει ανατεθεί κανένα δεδομένο για ένα ορισμένο πλήθος δεδομένων, διαγράφεται το κέντρο που αντιστοιχεί σε αυτόν τον υπόχωρο. Η επιλογή των δεδομένων εκπαίδευσης στην περίπτωση των ελαχίστων τετραγώνων πραγματοποιείται με χρήση ενός κυλιόμενου παραθύρου, του οποίου το μέγεθος αρχικοποιείται στο σύστημα σαν αρχική συνθήκη.

Στην περίπτωση που κατά την άφιξη ενός νέου δεδομένου δεν χρειάστηκε να προστεθεί ή να αφαιρεθεί κάποιο κέντρο, τότε τα συνάπτικα βάρη του δικτύου υπολογίζονται με την μέθοδο των Αναδρομικών Ελαχίστων Τετραγώνων RLS (recursive least squares). Σε αντίθετη περίπτωση ο υπολογισμός των συνάπτικων βαρών υπολογίζεται εκ νέου μαζικά (batch learning) για ένα συγκεκριμένο πλήθος των πιο πρόσφατων δεδομένων με την μέθοδο των ελαχίστων τετραγώνων LS (Least Squares). Η διαδικασία μάθησης ξεκινάει από το πρώτο δεδομένο, ενώ το δίκτυο έχει μηδέν νευρώνες χωρίς να χρειάζεται κάποιο πρότερα αρχικοποιημένο δίκτυο. Το πρώτο κέντρο είναι αυτό του ασαφούς υποσυνόλου, το οποίο ενεργοποιείται από το πρώτο δεδομένο. Είναι λογικό ότι στα πρώτα στάδια μάθησης η επιτυχία του δικτύου είναι μικρή, ενώ σταδιακά αυξάνεται καθώς το δίκτυο τροφοδοτείται με δεδομένα. Η διαδικασία εκπαίδευσης της υλοποιείται σε μορφή συστήματος πολλών εισόδων μίας εξόδου (Multi Input – Single Output MISO). Παρόλα αυτά η μετατροπή αυτού του αλγορίθμου σε σύστημα πολλών εισόδων και πολλών εξόδων είναι σχετικά εύκολη. Στο σχήμα 19 παρουσιάζεται η διαδικασία του αλγορίθμου σε αναλυτικά βήματα.



Εικόνα 19. Διάγραμμα ψευδοκώδικα αλγορίθμου.

Αρχικοποίηση παραμέτρων και υπολογισμός πρώτου δεδομένου

Προτού ο αλγόριθμος ξεκινήσει τη διαδικασία εκπαίδευσης θα πρέπει να οριστεί ένα πλήθος από παραμέτρους αρχικοποίησης.

- Το πλήθος των ασαφών υποσυνόλων c_i ανά διάσταση i , όπου $i = 1, 2, \dots, N$ το πλήθος των διαστάσεων που ορίζεται από της μεταβλητές εισόδου.
- Ο αριθμός N_d που καθορίζει το πλήθος των δεδομένων μετά από το οποίο ένα κέντρο δεν έχει ενεργοποιηθεί θα διαγράφεται.
- Το μέγεθος του παραθύρου N_w που καθορίζει το πλήθος των δεδομένων εκπαίδευσης που αποθηκεύονται για μελλοντική εκπαίδευση με διαδικασία ελαχίστων τετραγώνων.
- Η συντελεστής απόσβεσης λ (forgetting factor) για την μέθοδο RLS.

Όταν ολοκληρωθεί η αρχικοποίηση των συγκεκριμένων παραμέτρων το σύστημα είναι σε θέση να ξεκινήσει την διαδικασία εκπαίδευσης λαμβάνοντας το πρώτο δεδομένο εκπαίδευσης $[\mathbf{x}(1), y(1)]$. Κατόπιν ο πλησιέστερος ασαφής υπόχωρος $\mathbf{A}^1 = [\mathbf{A}_{1,j_1}^1, \mathbf{A}_{2,j_2}^1, \dots, \mathbf{A}_{N,j_N}^1]$ (όπου $j = 1, 2, \dots, c_i$) ενεργοποιείται με βάση το κριτήριο της ελάχιστης απόστασης υπολογίζοντας τους συντελεστές j_i σύμφωνα με την σχέση 2.22 ενότητα 2.6.3. Μετά τον υπολογισμό του πρώτου κέντρου ο αλγόριθμος δεσμεύει δύο πίνακες. Έναν πίνακα \mathbf{u} που θα περιλαμβάνει της συντεταγμένες των κέντρων για κάθε νευρών με διαστάσεις $L \times N$ όπου $\mathbf{u}_l = [u_{l,1}, u_{l,2}, \dots, u_{l,N}]$ (βλέπε ενότητα 2.6.1) είναι η γραμμή l του πίνακα \mathbf{u} . Και ένα διάνυσμα ιστορικού ενεργοποίησης (Activation History Vector AHV) \mathbf{h} διαστάσεων $1 \times L$ που περιλαμβάνει τον αριθμό των δεδομένων εκπαίδευσης που έχουν μεσολαβήσει από την τελευταία φορά που ενεργοποιήθηκε ο κάθε νευρώνας. Δεδομένου ότι έχει προσπελαστεί μόνο ένα δεδομένο εκπαίδευσης έχουμε ότι $\mathbf{u} = [\mathbf{u}_1] = [u_{1,1}, u_{1,2}, \dots, u_{1,N}]^T$, ενώ το διάνυσμα ιστορικού $h = [1]$. Ο αλγόριθμος συνεχίζει με τον υπολογισμό του πίνακα $Z(1)$ που περιλαμβάνει τις εξόδους της κρυφής στιβάδας για το πρώτο δεδομένο, καθώς και του πίνακα W των συναπτικών βαρών. Τέλος υπολογίζεται ο αντίστροφος του πίνακα συνδιακύμανσης $P = (Z^T \times Z)^{-1}$ για το πρώτο δεδομένο εκπαίδευσης. Οι παραπάνω υπολογισμοί υλοποιούνται σύμφωνα με της σχέσεις 3.4, 3.5, 3.6 αντίστοιχα.

$$Z(1) = \|\mathbf{g}(\mu(\mathbf{x}(1)))\| = \|\mathbf{g}(\|\mathbf{x}(1) - \mathbf{u}\|_2)\| = \|\mathbf{g}(\|\mathbf{x}(1) - \mathbf{u}_1\|_2)\| \quad (3.4)$$

$$W(1) = \begin{bmatrix} y(1) \\ Z(1) \end{bmatrix} \quad (3.5)$$

$$P(1) = \begin{bmatrix} 1 \\ (Z(1))^2 \end{bmatrix} \quad (3.6)$$

Όπου $\mu(\cdot)$ είναι η ευκλείδεια νόρμα του κέντρου \mathbf{u} από το δεδομένο $\mathbf{x}(k)$ και $\mathbf{g}(\cdot)$ είναι η ακτινική συνάρτηση βάσης.

Προσθήκη και διαγραφή νευρώνων

Όταν ένα νέο δεδομένο $\mathbf{x}(k)$ εισαχθεί, το σύστημα ελέγχει κατά πόσον τα υπάρχοντα κέντρα είναι επαρκής για το νέο δεδομένο. Για να διαπιστωθεί αυτό ο αλγόριθμος υπολογίζει την απόσταση μεταξύ των υπάρχοντων κέντρων και του δεδομένου $\mathbf{x}(k)$. Εάν η απόσταση $rd^{h_0}(\mathbf{x}(k))$ (βλέπε σχέση 2.21) από το κέντρο του κοντινότερου υπόχωρου l_0 είναι μεγαλύτερη του 1 τότε θεωρείται ότι τα ήδη υπάρχοντα κέντρα είναι ανεπαρκής και θα χρειαστεί να προστεθεί ένα νέο. Σε αντίθετη περίπτωση ο αλγόριθμος διατηρεί τα ήδη υπάρχοντα κέντρα. Στην περίπτωση που θα χρειαστεί να προστεθεί ένα νέο κέντρο ο αλγόριθμος αυξάνει τον αριθμό των νευρώνων L κατά ένα. Ενώ το σημείο τοποθέτησης του κέντρου νέου υπόχωρου \mathbf{A}^L υπολογίζεται με το κριτήριο της ελάχιστης αποστάτης σύμφωνα με την σχέση 2.22. Στην συνέχεια ο πίνακας \mathbf{u} , ενημερώνεται με το κέντρο του καινούργιου νευρώνα. Ενώ η τιμή του διανύσματος \mathbf{h} που αντιστοιχεί, λαμβάνει την τιμή k της

χρονικής στιγμής που δημιουργήθηκε ο νέος νευρώνας. Εάν δεν χρειαστεί να προστεθεί κάποιος νευρώνας στο σύστημα, η τιμή του διανύσματος \mathbf{h} που αντιστοιχεί ενημερώνεται για την τρέχουσα χρονική στιγμή k κατά την οποία ο κοντινότερος υπόχωρος l_0 ενεργοποιήθηκε για το δεδομένο $\mathbf{x}(k)$. Στην περίπτωση που κάποια τιμή του διανύσματος \mathbf{h} δεν έχει ενημερωθεί για μεγαλύτερο διάστημα N_d . Αυτό σημαίνει ότι ο υπόχωρος που αντιστοιχεί σε αυτήν την τιμή του διανύσματος \mathbf{h} δεν έχει ενεργοποιηθεί για μεγαλύτερο χρονικό διάστημα από N_d . Αυτός ο υπόχωρος θεωρείται πλεοναστικός και έτσι διαγράφεται αφαιρώντας την γραμμή που αντιστοιχεί στο διάνυσμα των κέντρων \mathbf{u} , ενώ στην συνέχεια αφαιρείται και το στοιχείο που αντιστοιχεί στο διαμέρισμα \mathbf{h} .

Υπολογισμός συναπτικών βαρών

Αυτός ο αλγόριθμος εκπαίδευσης χρησιμοποιεί δύο μεθόδους για τον υπολογισμό των συναπτικών βαρών του δικτύου. Στην περίπτωση όπου χρειαστεί να προστεθεί ή να αφαιρεθεί νευρώνας. Ο αλγόριθμος υπολογίζει τα συναπτικά βάρη με τη μέθοδο των ελαχίστων τετραγώνων σύμφωνα με τη σχέση 2.10 που περιγράψαμε στην ενότητα 2.6.1. Ο υπολογισμός αυτός γίνεται μαζικά (batch learning) για το πλήθος των τελευταίων δεδομένων που καθορίζεται από το μέγεθος του κυλιόμενου παραθύρου N_s . Στην περίπτωση που δεν χρειαστεί να προστεθεί ή να αφαιρεθεί οποιοδήποτε κέντρο τότε τα συναπτικά βάρη του δικτύου αναβαθμίζονται με τη μέθοδο των αναδρομικών ελαχίστων τετραγώνων με εκθετικό συντελεστή απόσβεσης (exponential forgetting). Ο υπολογισμός που εκτελείται για την αναβάθμιση των συναπτικών βαρών περιγράφεται αναλυτικά από τις εξισώσεις 3.7, 3.8 και 3.9.

$$\mathbf{W}(k) = \mathbf{W}(k-1) + q(k)(y(k) - \mathbf{Z}^T(k)\mathbf{W}(k-1)) \quad (3.7)$$

$$\mathbf{q}(k) = \mathbf{P}(k-1)\mathbf{Z}(k)(\lambda + \mathbf{Z}^T(k)\mathbf{P}(k-1)\mathbf{Z}(k))^{-1} \quad (3.8)$$

$$\mathbf{P}(k) = (\mathbf{I} - q(k)\mathbf{Z}^T(k))\mathbf{P}(k-1)/\lambda \quad (3.9)$$

Όπου λ είναι ο συντελεστής απόσβεσης ο οποίος σταδιακά μειώνει την βαρύτητα στην πληροφορία που προέρχεται από παλαιότερα δεδομένα με έναν πολλαπλασιαστή λ^n . Όπου n είναι τα δεδομένα n παλαιότητας. Ενώ με τον όρο \mathbf{I} συμβολίζεται ο μοναδιαίος διαγώνιος πίνακας.

3.5.2 Οικογένεια αλγορίθμων RAN

Ο αλγόριθμος RAN (resource allocating network) πρωτό δημοσιεύθηκε από των Platt (1991) και χρησιμοποιείται για την εκπαίδευση νευρωνικών δικτύων RBF [23]. Ο RAN ξεκινάει από μηδέν νευρώνες και καθώς λαμβάνει δεδομένα εκπαίδευσης προσθέτει νευρώνες στην κρυφή στιβάδα με βάση την καινοτομία τους. Ο αλγόριθμος RAN χρησιμοποιείται για on-line εκπαίδευση και η βάση του στηρίζεται στην ιδέα ότι το πλήθος των νευρώνων θα πρέπει να σχετίζεται με την πολυπλοκότητα της συνάρτησης που θέλουμε να προσεγγίσουμε και αυτή η πολυπλοκότητα αντικατοπτρίζεται στα δεδομένων εκπαίδευσης. Εάν δεν χρειαστεί να προστεθεί ένας νευρώνας στο δίκτυο κατά την άφιξη ενός δεδομένου που δεν καινοτομεί. Τότε τα συναπτικά βάρη του δικτύου αναβαθμίζονται με την LMS (Least Mean Squares). Ο Niranjan (1993) προτείνει μία ενισχυμένη έκδοση του RAN κατά την οποία ο υπολογισμός των συναπτικών βαρών γίνεται με τον αλγόριθμο Extended Kalman Filter (EKF) αντί των LMS [24]. Η υλοποίηση RAN-EKF έδειξε βελτιωμένες επίδοσης συγκριτικά με την απλή έκδοση RAN. Οι αλγόριθμοι RAN και RAN-EKF έχουν μηχανισμούς μονό προσθήκης νευρώνων στο δίκτυο. Έτσι ένας νευρώνας που θα προστεθεί δεν αφαιρείται ποτέ ακόμα και αν μελλοντικά έχει ελάχιστη συμμετοχή στην απόδοση του δικτύου. Ο αλγόριθμος M-RAN από Lu Yingwei (1997) εισήγαγε μία διαδικασία διαγραφής νευρώνων με βάση την συμμετοχή τους στην απόδοση του δικτύου [25]. Μία νέα πρόταση προσθήκης και αφαίρεσης νευρώνων εισήγαγε ο αλγόριθμος GAP ο οποίος καθόριζε την ανάπτυξη των νευρώνων αξιολογώντας την σημαντικότητα τους σε σχέση με την κατανομή των δεδομένων εισόδου [26].

RAN

Δεδομένου ότι εκπαιδεύουμε ένα νευρωνικό δίκτυο RBF η έξοδος του δικτύου για ένα δεδομένο εισόδου, δίνεται από το γραμμικό συνδυασμό των εξόδων της κρυφής στιβάδας σύμφωνα με την εξίσωση 3.10 όπως αναφέραμε και αναλυτικά στην 2.9 της ενότητας 2.6.1.

$$\hat{y}(k) = w_0 + \sum_{l=1}^L w_l g(\mu_l(\mathbf{x}(k))) \quad (3.10)$$

Στην εξίσωση 3.10 μπορούμε να παρατηρήσουμε τον όρο w_0 που αποκαλείται πόλωση (bias term) και χρησιμοποιείται για τη δυνατότητα της μετατόπισης της συνάρτησης που θέλουμε να προσεγγίσουμε από την αρχή των αξόνων. Η ακτινική συνάρτηση βάσεις $g(\cdot)$ που χρησιμοποιήθηκε στην υλοποίηση του Platt (1991) είναι η γκαουσιανή συνάρτηση τις εξισώσεις 2.16 στην ενότητα 2.6.2.

Όπως προαναφέρθηκε το δίκτυο ξεκινάει με μηδέν νευρώνες στην κρυφή στιβάδα. Κατά την άφιξη των πρώτων δεδομένων $[\mathbf{x}(1), y(1)]$ ο αλγόριθμος αποδίδει την τιμή $y(1)$ στο $w_0 = y(1)$. Κατόπιν για κάθε νέο δεδομένο $[\mathbf{x}(k), y(k)]$ το οποίο εισέρχεται στο δίκτυο ο αλγόριθμος προσθέτει νέους νευρώνες με βάση το κριτήριο της καινοτομίας τους που δίνεται από τις σχέσεις 3.11 και 3.12.

$$\|\mathbf{x}(k) - \mathbf{u}_{nr}\| > \varepsilon_n \quad (3.11)$$

$$e_n = y(k) - \hat{y}(\mathbf{x}(k)) > e_{min} \quad (3.12)$$

Όπου \mathbf{u}_{nr} είναι το κέντρο με την μικρότερη απόσταση από το δεδομένο $\mathbf{x}(k)$. Ενώ τα e_{min} είναι αρχική παράμετρος που επιλέγεται κατάλληλα κατά την αρχικοποίηση του δικτύου και ε_n προκύπτει από μία σχέση από τρεις άλλες αρχικές παραμέτρους. Με βάση το κριτήριο της εξίσωσης 3.11 ένα νέο δεδομένο πρέπει να βρίσκεται επαρκώς μακριά από το πλησιέστερο κέντρο που είναι ήδη δεσμευμένο προκειμένου να θεωρηθεί καινοτόμο. Ενώ σύμφωνα με το κριτήριο της εξίσωσης 3.12 προκειμένου ένα νέο δεδομένο $\mathbf{x}(k)$ να θεωρηθεί καινοτόμο θα πρέπει ταυτόχρονα η πρόβλεψη του δικτύου για αυτό το δεδομένο να έχει σημαντική διαφορά από την πραγματική τιμή, με ένα μεγάλο σφάλμα στην έξοδο του. Η τιμή του συντελεστή e_{min} επιλέγεται να είναι η επιθυμητή απόδοση του δικτύου. Ενώ η απόσταση ε_n αντιπροσωπεύει την ακρίβεια στον χώρο εισόδου και κατά την έναρξη του αλγόριθμου η μεταβλητή παίρνει τη μέγιστη τιμή $\varepsilon_n = \varepsilon_{max}$. Ενώ σταδιακά μειώνεται εκθετικά σύμφωνα με την σχέση 3.13.

$$\varepsilon_n = \max\{\varepsilon_{max}\gamma^k, \varepsilon_{min}\} \quad (3.13)$$

Όπου γ είναι σταθερά απόσβεσης και με βάση την εξίσωση 3.13 ο συντελεστής ε_n θα αποσβένει μέχρι να φτάσει στην τιμή ε_{min} . Με την σχέση 3.13 ο αλγόριθμος ελέγχει τον ρυθμό ανάπτυξης νευρώνων κατά την έναρξη της διαδικασίας εκπαίδευσης καθώς και τον πλήθος των νευρώνων που τελικά ο αλγόριθμος θα σταθεροποιηθεί. Όταν ένας νέος νευρώνας εισαχθεί στο δίκτυο αρχικοποιείται σύμφωνα με τις σχέσεις 3.14, 3.15 και 3.16.

$$w_{L+1} = e_n \quad (3.14)$$

$$u_{L+1} = \mathbf{x}(k) \quad (3.15)$$

$$\sigma_{L+1} = \kappa \|\mathbf{x}(k) - \mathbf{u}_{nr}\|_2 \quad (3.16)$$

Με L να είναι ο αριθμός των νευρώνων ενώ κ είναι ο συντελεστής επικάλυψης όπου καθορίζει την επικάλυψη των ακτινικών συναρτήσεων βάσης στον χώρο εισόδου. Το σ_{L+1} είναι η ευκλείδεια απόσταση του νέου κέντρου από το πλησιέστερο κέντρο σε αυτό, πολλαπλασιασμένη με τον παράγοντα επικάλυψης κ . Στην περίπτωση που αρχικοποιείται για πρώτη φορά ένα κέντρο το σ_1 θα λάβει τιμή ως εξής $\sigma_1 = \kappa \varepsilon_n$. Όταν τα κριτήρια καινοτομίας δεν πληρούνται το σύστημα δεν προσθέτει κάποιο νέο κέντρο στο δίκτυο. Και στη συνέχεια αναβαθμίζει τα συναπτικά βάρη της εξόδου της

κρυφής στιβάδας μην τη στοιβάδα εξόδου, ενώ ταυτόχρονα προσαρμόζει και τα κέντρα των RBF. Οι παράμετροι του συστήματος ταξινομούνται σε μορφή πίνακα $\mathbf{W} = [w_0, w_1, \dots, w_L, \mathbf{u}_1^T, \mathbf{u}_2^T, \dots, \mathbf{u}_L^T]$ και αναβαθμίζονται με την μέθοδο LMS σύμφωνα με την σχέση 3.17.

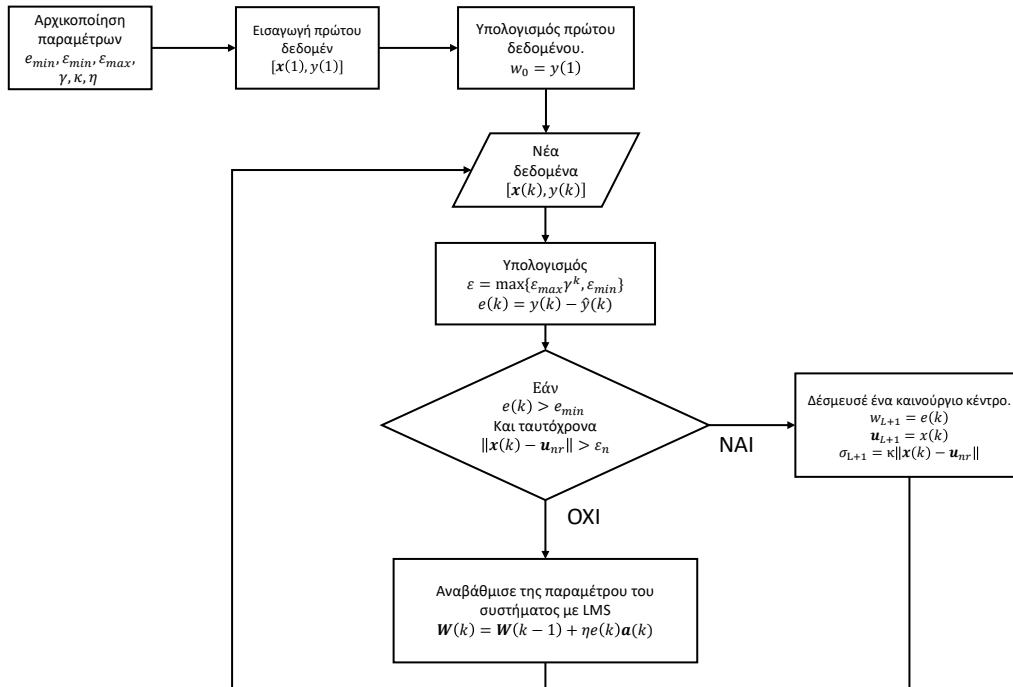
$$\mathbf{W}(k) = \mathbf{W}(k-1) + \eta e_n \mathbf{a}_n \quad (3.17)$$

Όπου η είναι ένας συντελεστής προσαρμογής και $\mathbf{a}_n = \nabla_W \hat{y}(\mathbf{x}(k))$ είναι η παραγωγός της $\hat{y}(\cdot)$ σε σχέση με το διάνυσμα \mathbf{W} υπολογισμένο την στιγμή $\mathbf{W}(k-1)$. Ως εκ τούτου από τον υπολογισμό προκύπτει το αποτέλεσμα της σχέσης 3.18.

3.17.

$$\mathbf{a}_n = [1, g_1(\mathbf{x}(k)), \dots, g_L(\mathbf{x}(k)), g_1(\mathbf{x}(k)) \frac{2w_1}{\sigma_1^2} (\mathbf{x}(k) - \mathbf{u}_1)^T, \dots, g_L(\mathbf{x}(k)) \frac{2w_L}{\sigma_L^2} (\mathbf{x}(k) - \mathbf{u}_L)^T]^T \quad (3.18)$$

Ο ρυθμός αύξησης των νευρώνων του αλγόριθμου ρυθμίζεται από τις παραμέτρους γ , ε_{min} και ε_{max} , ενώ αυτές καθορίζουν την τελική σταθεροποίηση του πλήθους των νευρώνων του δικτύου. Ο αλγόριθμος RAN έχει την ικανότητα να εκπαιδεύει γρήγορα και να επιτυγχάνει μεγάλη ακρίβεια στις πρόβλεψης του. Στο διάγραμμα του σχήματος 20 παρουσιάζεται ο αλγόριθμος 20 σε μορφή ψευδοκώδικα.



Εικόνα 20. Διάγραμμα ψευδοκώδικα αλγορίθμου RAN.

RAN-EKF

Ο αλγόριθμος RAN-EKF παρουσιάζεται σαν μία ενισχυμένη έκδοση του RAN. Η αναβάθμιση που προτείνεται είναι να αντικατασταθεί η διαδικασία LMS που υπολογίζει τις παραμέτρους του δικτύου με την διαδικασία EKF (Extended Kalman Filter). Η διαδικασία EKF αυξάνει την ταχύτητα σύγκλισης του δικτύου ενώ ταυτόχρονα μειώνει την πολυπλοκότητα της υποδομής του, κάνοντας καλύτερη επιλογή νέων κέντρο για της RBF. Η διαδικασία EKF Αναβαθμίζει το διάνυσμα παραμέτρων του δικτύου $\mathbf{W}(k)$ από το διάνυσμα παραμέτρων της προηγούμενης χρονικής

στιγμής $\mathbf{W}(k-1)$ και από τον αντίστροφο του πίνακα συνδιακύμανσης σφάλματος της προηγούμενης χρονικής στιγμής $\mathbf{P}(k-1)$ όπως περιγράφεται από της εξίσωσης 3.19, 3.20 και 3.21 .

$$\mathbf{W}(k) = \mathbf{W}(k-1) + e(k) + \mathbf{q}(k) \quad (3.19)$$

$$\mathbf{q}(k) = [R(k) + \mathbf{a}(k)^T \mathbf{P}(k-1) \mathbf{a}(k)]^{-1} \mathbf{P}(k-1) \mathbf{a}(k) \quad (3.20)$$

$$\mathbf{P}(k) = [\mathbf{I} - \mathbf{q}(k) \mathbf{a}(k)^T] \mathbf{P}(k-1) + Q_0 \mathbf{I} \quad (3.21)$$

Όπου $\mathbf{a}(k)$ είναι το διάνυσμα κλίσης, $R(k)$ είναι η διακύμανση του θορύβου μέτρησης, $\mathbf{P}(k)$ ο αντίστροφος του πίνακα συνδιακύμανσης σφάλματος και \mathbf{I} είναι ο μοναδιαίος διαγώνιος πίνακας. Ο όρος Q_0 είναι ένας παράγοντας τυχαιότητας που επιτρέπει το τυχαίο βήμα στην διεύθυνση του διανύσματος κλίσης. Με αυτό τον τρόπο το σύστημα αποφεύγει την πρόωρη σύγκλιση σε μη επιθυμητά τοπικά ελάχιστα.

Στην περίπτωση όπου ένας καινούριος νευρώνας θα πρέπει να προστεθεί στο δίκτυο οι διαστάσεις του πίνακα \mathbf{P} θα πρέπει να αυξηθούν. Δεδομένου ότι ο P είναι ένας συμμετρικός πίνακας με διαστάσεις $L \times L$ η αναβάθμιση των διαστάσεων του γίνεται όπως περιγράφεται στην σχέση 3.22.

$$\mathbf{P}(k) = \begin{bmatrix} \mathbf{P}(k-1) & 0 \\ 0 & P_0 \mathbf{I} \end{bmatrix} \quad (3.22)$$

Όπου το P_0 είναι η παράμετρος της εκτιμώμενης αβεβαιότητας που εισάγεται στο σύστημα. Οι διαστάσεις του μοναδιαίου διαγώνιου πίνακα \mathbf{I} είναι ανάλογες του πλήθους των μεταβλητών που εισάγουμε στο σύστημα. Ο αλγόριθμος RAN μπορεί εύκολα να μετατραπεί στην ενισχυμένη έκδοση RAN-EKF αντικαθιστώντας απλώς τη διαδικασία LMS με τη διαδικασία EKF που περιγράφεται από τις εξισώσεις 3.19, 3.20, 3.21 και 3.22.

M-RAN

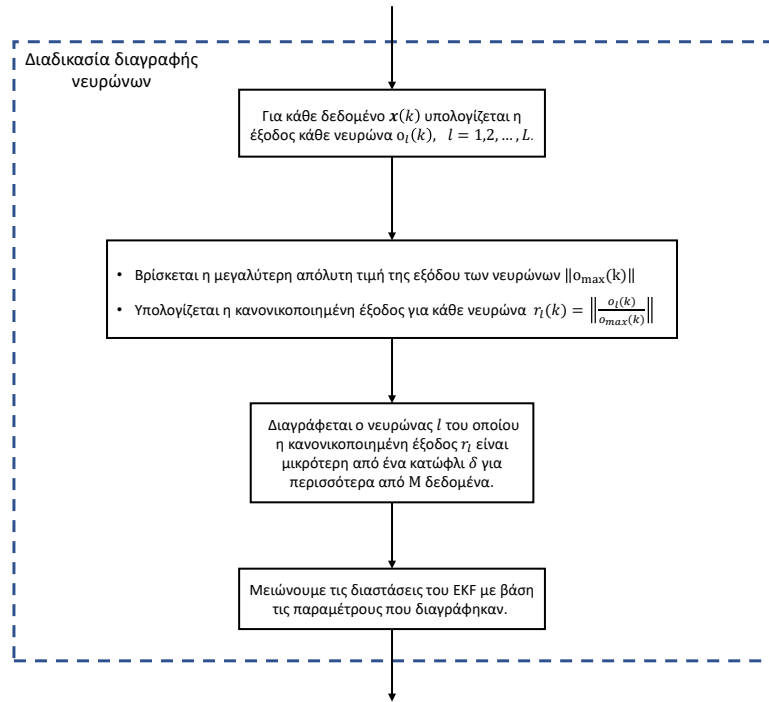
Με μία σύντομη ματιά στους αλγόριθμους RAN και RAN-EKF μπορούμε να διαπιστώσουμε ότι ένας νευρώνας που θα δεσμευτεί δεν θα αφαιρεθεί ποτέ. Σε μία On-line διαδικασία εκπαίδευσης είναι πολύ πιθανόν ένας νευρώνας μετά από αρκετό χρονικό διάστημα να έχει πολύ μικρή συμμετοχή στην τελική έξοδο του δικτύου. Δεδομένου αυτού ο νευρώνας αυτός είναι πλεοναστικός και θα ήταν ιδανικό να μπορεί να αφαιρεθεί. Ο αλγόριθμος M-RAN δημιουργήθηκε με στόχο να αντιμετωπίσει αυτά τα προβλήματα. Υιοθετώντας την βασική ιδέα του RAN-EKF ο M-RAN εισάγει σε αυτόν μία διαδικασία αφαιρέσεις νευρώνων ενώ ταυτόχρονα δημιουργεί ένα επιπλέον κριτήριο για την προσθήκη ενός νέου νευρώνα.

Δεδομένου ότι η έξοδος του νευρώνα l δίνεται από την σχέση 3.23.

$$o_l(k) = w_l \exp\left(-\frac{\|\mathbf{x}(k) - \mathbf{u}_l\|^2}{\sigma_l^2}\right) \quad (3.23)$$

Στην περίπτωση όπου οι τιμές του w_l και του σ_l είναι πολύ μικρές τότε αυτό σημαίνει ότι η έξοδος του συγκεκριμένου νευρώνα τείνει και αυτή να είναι μικρή. Σε αντίθετη περίπτωση εάν ο όρος $\|\mathbf{x}(k) - \mathbf{u}_l\|^2$ είναι πολύ μεγάλος που σημαίνει το δεδομένο είναι πολύ μακριά από το κέντρο της ακτινικής συνάρτησης βάσης τότε πάλι η έξοδος του νευρώνα είναι πολύ μικρή. Κάνοντας έτσι το νευρώνα να έχει πολύ μικρή συμμετοχή στην τελική έξοδο του δικτύου. Αυτό σημαίνει ότι εάν ένας νευρώνας έχει πολύ μικρή συμμετοχή για ένα πλήθος συνεχόμενων δεδομένων εκπαίδευσης, τότε αυτός ο νευρώνας μπορεί να θεωρηθεί πλεονασματικός και να αφαιρεθεί. Ο αλγόριθμος M-RAN προκειμένου να διαπιστώσει ποιοι νευρώνες δεν συμμετέχουν σημαντικά στην έξοδο του δικτύου. Ελέγχει την συμμετοχή από όλες τις εξόδους των νευρώνων για κάθε νέο δεδομένο που λαμβάνει. Εάν η έξοδος ενός νευρώνα είναι διαρκώς μικρότερη από ένα κατώφλι δ για ένα πλήθος συνεχόμενων δεδομένων M τότε αυτός ο νευρώνας διαγράφεται από το δείκτη. Στο κριτήριο

απόφασης της διαγραφής νευρώνων χρησιμοποιείται η κανονικοποιημένη έξοδος των νευρώνων $r_l(k)$ που υπολογίζεται ως εξής $r_l(k) = \|o_l(k)/o_{max}(k)\|$. Δεδομένου ότι οι κανονικές τιμές των εξόδων μπορεί να δημιουργήσουν ασυνέπειες στη διαδικασία διαγραφής. Η διαδικασία διαγραφής των νευρώνων παρουσιάζεται στο σχήμα 21.



Εικόνα 21. Διάγραμμα ψευδοκώδικα διαδικασίας διαγραφής νευρώνων.

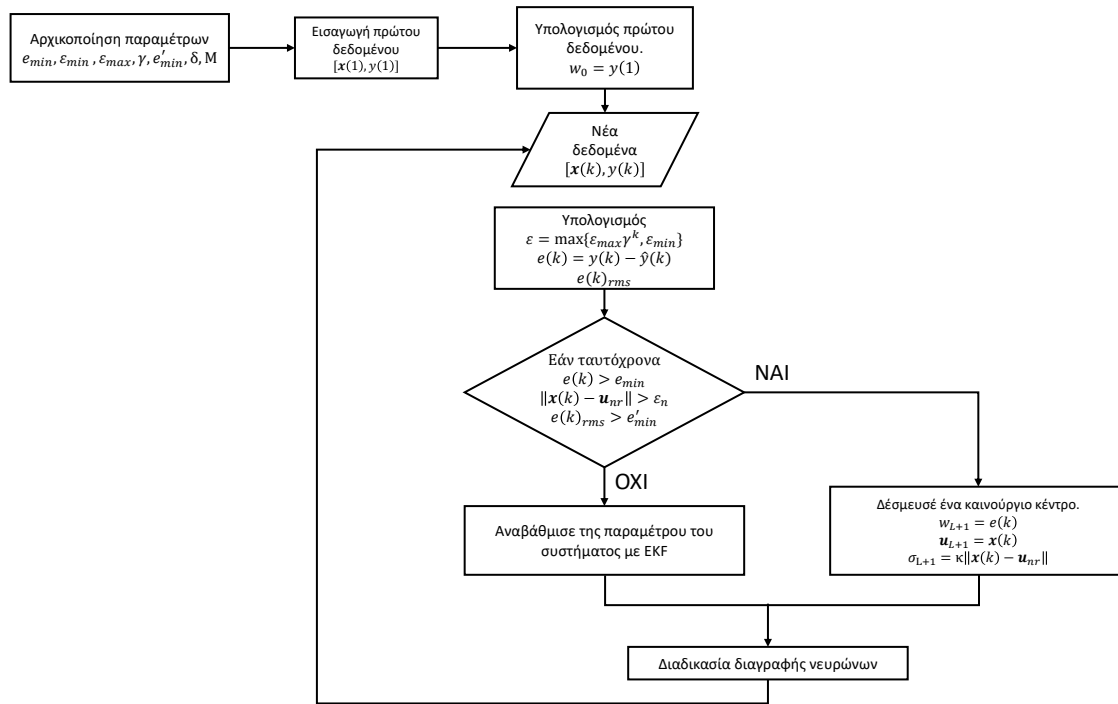
Ο αλγόριθμος M-RAN εκτός από την διαδικασία αφαιρέσεις των περιττών νευρώνων. Προσθέτει ένα επιπλέον κριτήριο για την προσθήκη νέου νευρώνα στο δίκτυο. Σύμφωνα με των M-RAN για να θεωρηθεί ένας νευρώνας καινοτόμος θα πρέπει να πληρούνται ταυτόχρονα το κριτήριο των σχέσεων 3.11, 3.12 και το νέο κριτήριο που εισάγει ο M-RAN στις σχέσεις 3.24.

$$e'_{min} < e_{rms} = \sqrt{\frac{\sum_{i=k-(M-1)}^k [y(k) - \hat{y}(k)]^2}{M}} \quad (3.24)$$

Με αυτό το κριτήριο ένα κέντρο θα δεσμευτεί ως καινοτόμο όταν επιπρόσθετα το μέσο τετραγωνικό σφάλμα για το k δεδομένο μέσα σε ένα κυλιόμενο παράθυρο M (Που ορίσαμε παραπάνω) είναι μεγαλύτερο από ένα κατώφλι e'_{min} . Ο τελικός αλγόριθμος M-RAN παρουσιάζεται σε μορφή ψευδοκώδικα στο διάγραμμα του σχήματος 22.

GAP-RBF

Ο αλγόριθμος GAP-RBF (Growing and Pruning - Radial Basis Function) παρά την καινοτομία του ονόματος του είναι και αυτός ένας ακόμα αλγόριθμος της οικογένειας RAN. Δεδομένου ότι στην πραγματικότητα είναι μία παραλλαγή του αλγόριθμου RAN-EKF με μία νέα πρόταση για την διαδικασία προσθήκης και διαγραφής νευρώνων. Ο αλγόριθμος GAP-RBF συνδέει την επιθυμητή ακρίβεια με τη διαδικασία εκπαίδευσης με βάση την σημαντικότητα ενός νευρώνα. Η διαδικασία υπολογισμού της σημαντικότητας ενός νευρώνα γίνεται με χρήση μιας απλοποιημένης εκδόσεις της γκαουσιανής συνάρτησης προκειμένου να μειωθεί το υπολογιστικό κόστος. Η διαδικασία προσθήκης ή αφαίρεσης ενός νευρώνα καθορίζεται με κριτήριο την σημαντικότητά του. Σε αντίθεση



Εικόνα 22. Διάγραμμα ψευδοκώδικα M-RAN.

με των M-RAN πού για την προσθήκη ενός νευρώνα χρησιμοποιεί σαν κριτήριο την διαδικασία RMS (Root Mean Square). Είναι σημαντικό να προσθέσουμε ότι στον συγκεκριμένο αλγόριθμο όταν ένας νευρώνας προστίθεται ή αφαιρείται, ελέγχεται η σημαντικότητα μόνο του κοντινότερου νευρώνα στα εκάστοτε δεδομένα εκπαίδευσης. Επιπρόσθετα στην περίπτωση που δεν προστεθεί η αφαιρεθεί κάποιο κέντρο. Τότε ο αλγόριθμος αναβαθμίζει μόνο των πλησιέστερο νευρώνα στο συγκεκριμένο δεδομένα εκπαίδευσης. Με αυτό τον τρόπο ο αλγόριθμος αυξάνει την ταχύτητα της διαδικασίας εκπαίδευσης.

Θεωρώντας ότι η διαδικασία εκπαίδευσης έχει ξεκινήσει και ένα δίκτυο L νευρώνων έχει σχηματιστεί. Εάν αφαιρέσουμε το νευρώνα j τότε η έξοδος του δικτύου θα πρέπει να δίνεται σύμφωνα με τη σχέση 3.25 (Δδομένου ότι η έξοδος του δικτύου χωρίς να αφαιρέσουμε κάποιον νευρώνα δίνεται από τη σχέση 3.10 χωρίς το «bias term»)

$$\hat{y}_{rmv_j}(k) = \sum_{l=1}^{j-1} w_l g_l(k) + \sum_{j+1}^L w_l g_l(k) \quad (3.25)$$

Με αυτό τον τρόπο το σφάλμα που προκαλείται για το δεδομένο $\mathbf{x}(k)$ στην έξοδο του δικτύου με την αφαίρεση του νευρώνα j μπορεί να δοθεί από την σχέση 3.26.

$$E(j, k) = |\hat{y}(k) - \hat{y}_{rmv_j}(k)| = |w_j| g_j(k) \quad (3.26)$$

Ο υπολογισμός του μέσου σφάλματος $E_{ave}(j)$ για την αφαίρεση του j νευρώνα, μετά από την άφιξη K δεδομένων εκπαίδευσης δίνεται από τη σχέση 3.27.

$$E_{ave}(j) = \frac{\sum_{k=1}^K E(j, k)}{K} = \frac{|w_j|}{K} \sum_{k=1}^K g_j(\mathbf{x}(k)) \quad (3.27)$$

Η εξίσωση 3.27 ορίζει την σημαντικότητα ενός νευρώνα για όλα τα δεδομένα εκπαίδευσης τα οποία έχει λάβει το νευρωνικό δίκτυο. Ένας τέτοιος υπολογισμός για όλα τα δεδομένα εκπαίδευ-

σης έχει μεγάλες απαιτήσεις σε μνήμη και υπολογιστική ισχύ κάτι που δεν εξυπηρετεί για online μοντέλα. Για να ξεπεραστεί αυτό το εμπόδιο ο υπολογισμός και η αποθήκευση γίνεται μόνο για τα δεδομένα τα οποία βρίσκονται επαρκώς κοντά στο κέντρο ενός νευρώνα. Θεωρώντας ένα δεδομένο ότι βρίσκεται κοντά σε ένα κέντρο σε μικρότερη απόσταση από $\|\mathbf{x}(k) - \mathbf{u}_j\| \leq 1.7\sigma_k$ τότε μπορούμε να υπολογίσουμε προσεγγιστικά την σημαντικότητα ενός νευρώνα για όλα τα m δεδομένα εκπαίδευσης τα οποία βρίσκόντουσαν σε αυτό το διάστημα, σύμφωνα με τη σχέση 3.28.

$$E_{ave}(j) \approx \frac{|w_j|}{K} \sum_{k=1}^m g_j(\mathbf{x}(k)) \quad \text{όπου} \quad x(k) \in [u_j - 1.7\sigma_j, u_j + 1.7\sigma_j] \quad (3.28)$$

Τα δεδομένα εκπαίδευσης που βρίσκονται έξω από αυτό το διάστημα $[u_j - 1.7\sigma_j, u_j + 1.7\sigma_j]$ έχουν πολύ μικρή συμμετοχή στον υπολογισμό της σημαντικότητας του νευρώνα και με αυτό τον τρόπο ο προσεγγιστικός υπολογισμός της σχέσης 3.28 δεν έχει σημαντική απόκλιση από την πραγματικότητα.

Προκειμένου να μειωθεί περαιτέρω το υπολογιστικό κόστος του υπολογισμού της γκαουσιανή συνάρτησης στην εξίσωση 3.28, προσεγγίζουμε την γκαουσιανή με ένα πλήθος από γραμμικές συναρτήσεις όπως περιγράφεται από την εξίσωση 3.29.

$$g_j(k) \approx \begin{cases} \frac{0.77}{1.2\sigma_j}(x(k) - u_j + 1.7\sigma_j), & \text{εάν} \quad u_j - 1.7\sigma_j \leq x(k) < u_j - 0.5\sigma_j \\ 0.77 + \frac{0.23}{0.5\sigma_j}(x(k) - u_j + 0.5\sigma_j) & \text{εάν} \quad u_j - 0.5\sigma_j \leq x(k) < u_j \\ 0.77 - \frac{0.23}{0.5\sigma_j}(x(k) - u_j - 0.5\sigma_j) & \text{εάν} \quad u_j \leq x(k) < u_j + 0.5\sigma_j \\ -\frac{0.77}{1.2\sigma_j}(x(k) - u_j - 1.7\sigma_j) & \text{εάν} \quad u_j + 0.5\sigma_j \leq x(k) < u_j + 1.7\sigma_j \\ 0 & \text{Οπουδήποτε αλλού} \end{cases} \quad (3.29)$$

Εάν υποθέσουμε ότι για το νευρώνα j έχουν καταγραφεί m δεδομένα εκπαίδευσης τα οποία βρίσκονται στο διάστημα $[u_j - 1.7\sigma_j, u_j + 1.7\sigma_j]$. Ενώ ταυτόχρονα αυτά τα δεδομένα έχουν προέλθει με ομοιόμορφη κατανομή (δηλαδή έχουν παρθεί με μία διασπορά σε όλο το πεδίο ορισμού του συνόλου δεδομένων \mathbf{X}) από το σύνολο δεδομένων εκπαίδευσης \mathbf{X} . Τότε το ποσοστό των δεδομένων m που θα βρεθούν στο διάστημα $[u_j - 0.5\sigma_j, u_j + 0.5\sigma_j]$ θα είναι $((1)\sigma_j / (3.4)\sigma_j)m$ ενώ το ποσοστό που θα βρεθεί στο διάστημα $[u_j - 1.7\sigma_j, u_j - 0.5\sigma_j] \cup [u_j + 0.5\sigma_j, u_j + 1.7\sigma_j]$ είναι $((2.4)\sigma_j / (3.4)\sigma_j)m$. Από τις εξισώσεις 3.28 και 3.29 προκύπτει η εξίσωση 3.30.

$$E_{ave}(j) \approx \frac{|w_j|}{K} \left(\frac{1}{3.4}m \frac{1 + 0.77}{2} + \frac{2.4}{3.4}m + \frac{0.77}{2} \right) \approx 0.53 \frac{m}{K} |w_j| \quad (3.30)$$

Εάν προεκτείνουμε την εξίσωση 3.30 για τις διαστάσεις N των διανυσμάτων \mathbf{x} και \mathbf{u} τότε έχουμε.

$$E_{ave}(j) \approx 0.53 \frac{N}{n} \frac{m}{K} |w_j| \quad (3.31)$$

Δεδομένου ότι έχουμε θεωρήσει ότι όλα τα δεδομένα k από το σύνολο των δεδομένων εκπαίδευσης \mathbf{X} κατανέμονται ομοιόμορφα στο πεδίο ορισμού του \mathbf{X} . Μπορούμε να θεωρήσουμε ότι $(m/K) = (3.4\sigma_j)^N / S(\mathbf{X})$ με $S(\mathbf{X})$ να είναι το πεδίο τιμών από το οποίο λαμβάνονται τα δεδομένα εκπαίδευσης. Έτσι από την εξίσωση 3.31 και με το παραπάνω δεδομένο μπορούμε να πάρουμε την τελική μορφή τις σχέσεις 3.32 που δίνει την σημαντικότητα για το νευρώνα j .

$$E_{ave}(j) = E_{sig}(j) \approx \left| \frac{(1.8\sigma_j)^N w_j}{S(\mathbf{X})} \right| \quad (3.32)$$

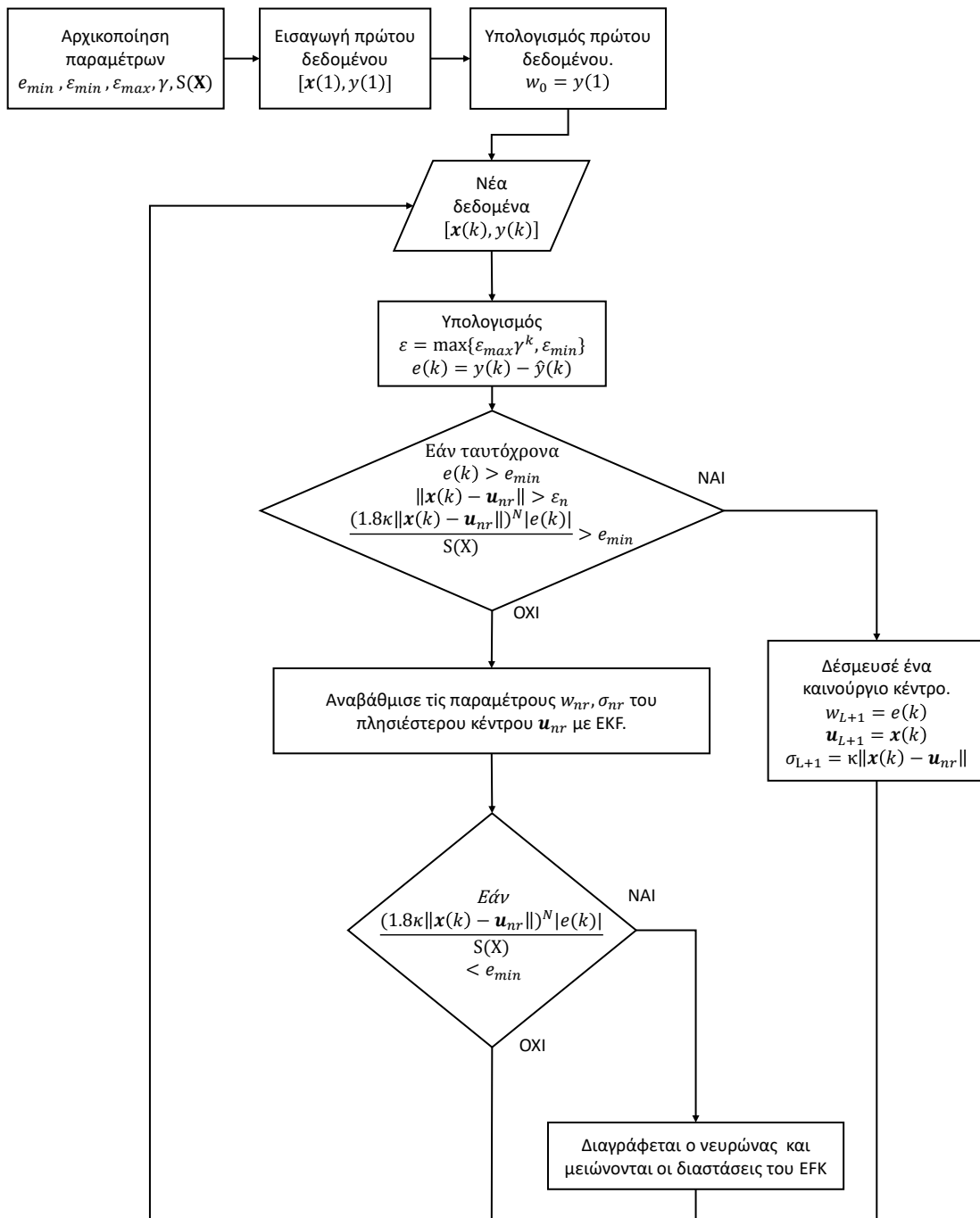
Στον αλγόριθμο GAP-RBF το αν θα προστεθεί ένας νέος νευρώνας καθορίζεται από το εάν ισχύουν ταυτόχρονα οι συνθήκες 3.11 και 3.12 του αλγόριθμου RAN και ταυτόχρονα η σημαντικότητα

του νέο νευρώνα που θα προστεθεί να είναι μεγαλύτεροι από το κατώφλι e_{min} και υπολογίζεται σύμφωνα με την σχέση 3.33.

$$\frac{(1.8\kappa\|x(k) - u_{nr}\|)^N |e(k)|}{S(X)} > e_{min} \quad (3.33)$$

Όπου ο όρος $\kappa\|x(k) - u_{nr}\|$ είναι το σ_{L+1} που θα δοθεί στο νευρώνα εφόσον δεσμευτεί και $|e(k)|$ είναι η τιμή που θα δοθεί στο w_{L+1} , βλέπε σχέσεις 3.15 και 3.16 αντίστοιχα.

Στο στάδιο ελέγχου διαγραφής των νευρώνων ο αλγόριθμος GAP-RBF υπολογίζει την σημαντικότητα του κοντινότερου νευρώνα στο εισερχόμενο δεδομένο $\mathbf{x}(k)$ και στην περίπτωση όπου αυτή βρίσκεται να είναι μικρότερη από e_{min} τότε αυτός ο νευρώνας διαγράφεται. Γνωρίζοντας ότι η πρώτη και η δεύτερη παράγωγος της γκαουσιανής συνάρτησης τείνει εκθετικά στο μηδέν όσο το \mathbf{x} απομακρύνεται από το κέντρο της. Τότε αυτό σημαίνει ότι όλα τα στοιχεία του διανύσματος παραγωγού $\mathbf{a}(k)$ θα τείνουν στο μηδέν εκτός από τις παραμέτρους που αντιστοιχούν στο νευρώνα που βρίσκεται κοντά στο δεδομένο $\mathbf{x}(k)$. Έχοντας αυτή την πληροφορία προκειμένου να μεγιστοποιηθεί η ταχύτητα του αλγόριθμου κατά την άφιξη ενός νέου δεδομένου $\mathbf{x}(k)$. Εάν ένας νέος νευρώνας προστεθεί αυτό σημαίνει ότι μόνο αυτός θα χρειαστεί να αναβαθμιστεί ενώ ταυτόχρονα είναι περιττό να γίνει έλεγχος εκ νέου για διαγραφή νευρώνων λόγω χαμηλής σημαντικότητας. Δεδομένου ότι δεν έχει αλλάξει κάτι από την τελευταία φορά που ελέγχθηκαν. Στην περίπτωση που δεν χρειάζεται να προστεθεί κάποιος νέος νευρώνας τότε αναβαθμίζεται μόνο ο κοντινότερος νευρώνας στο δεδομένο $\mathbf{x}(k)$. Και στη συνέχεια ο ίδιος νευρώνας ελέγχεται για τη σημαντικότητα του και για το εάν θα πρέπει να αφαιρεθεί από το δίκτυο. Στην περίπτωση που δεν προστεθεί η αφαιρεθεί κάποιος νευρώνας. Ο αλγόριθμος αναβαθμίζει τις παραμέτρους που αντιστοιχούν μόνο στον κοντινότερο νευρώνα του εισερχόμενου δεδομένου $\mathbf{x}(k)$. Ο αλγόριθμος GAP-RBF παρουσιάζεται σε μορφή ψευδοκώδικα στο διάγραμμα του σχήματος 23.



Εικόνα 23. Διάγραμμα ψευδοκώδικα GAP-RBF.

Κεφάλαιο 4

Προτεινόμενη Μεθοδολογία

4.1 Προκλήσεις και προβλήματα

Οι αλγόριθμοι της οικογένειας RAN είναι μία ιδιαίτερα διάσημη σειρά online αλγόριθμων για εκπαίδευση νευρωνικών δικτύων RBF. Η αλγόριθμοι RAN προσφέρουν καινοτόμες τεχνικές όπως επιλεκτικής επιλογής των κέντρων καθώς και διαδικασίες διαγραφής και προσθήκης νέων νευρώνων. Παρόλα αυτά οι αλγόριθμοι αυτοί δυσκολεύονται ιδιαίτερα στην προσέγγιση συναρτήσεων που έχουν υψηλές απαιτήσεις σε πλήθος κέντρων RBF. Οι διαδικασίες RAN επιδιώκουν να προσεγγίσουν την βέλτιστη τοποθεσία των κέντρων εισάγοντας της συντεταγμένες στις εκπαιδευόμενες μεταβλητές του δικτύου μαζί με το πλάτος σ και το βάρος w των Gaussian RBF για κάθε κέντρο. Έτσι κάθε νέο κέντρο που προστίθεται αυξάνει σημαντικά την υπολογιστική απαίτηση της διαδικασίας μάθησης. Συγκεκριμένα θα χρειαστούν να προστεθούν στις παραμέτρους υπολογισμού του δικτύου οι μεταβλητές $\sigma, w, c_1, c_2, \dots, c_N$ όπου N το πλήθος των μεταβλητών εισόδου. Αυτό σημαίνει ότι για ένα πρόβλημα που αποτελείται από πολλές μεταβλητές εισόδου x_N για καθένα νευρώνα, θα πρέπει το δίκτυο να προσθέσει μία παράμετρο για κάθε μεταβλητή εισόδου για να μπορεί να προσεγγίσει τις τοποθεσίες των κέντρων. Αυτή η μέθοδος μπορεί πολύ εύκολα να οδηγήσει την διαδικασία εκπαίδευσης σε χρονοβόρους υπολογισμούς ή ακόμα χειρότερα σε αδυναμία επίλυσης προβλημάτων που έχουν υψηλές απαιτήσεις σε πλήθος κέντρων. Ένα ακόμα πρόβλημα το οποίο προκαλείται από την προσθήκη μεγάλου αριθμού μεταβλητών στις παραμέτρους του δικτύου είναι η αύξηση της πολυπλοκότητας της επιφάνειας σφάλματος.

Σε αντίθεση με τους αλγόριθμους της οικογένειας RAN, ο αλγόριθμος FUZZY-LS-RLS που προτείνεται στην μελέτη [16] πλεονεκτεί ιδιαίτερα στο ότι διαχωρίζει την διαδικασία της εκπαίδευσης σε δύο στάδια. Με την τεχνική της ασαφούς λογικής υπολογίζει τα κέντρα των ακτινικών συναρτήσεων βάσης, ελαχιστοποιώντας έτσι τις παραμέτρους που πρέπει να υπολογίσει ο αλγόριθμος εκπαίδευσης. Με αυτόν τον τρόπο η διαδικασία εκπαίδευσης επιταχύνεται, αφού χρησιμοποιεί τις ελάχιστες δυνατές παραμέτρους, ενώ ταυτόχρονα η επιφάνεια σφάλματος είναι απλούστερη, δίνοντας μεγαλύτερη πιθανότητα στον αλγόριθμο εκπαίδευσης να βρεθεί σε ένα ολικό ελάχιστο. Παρόλα αυτά η διαδικασία FUZZY-LS-RLS έχει το σημαντικό μειονέκτημα ότι στην περίπτωση προσθήκης ή αφαίρεσης ενός νευρώνα ο αλγόριθμος εκπαιδεύει το δίκτυο από την αρχή. Χρησιμοποιώντας την μέθοδο των Ελάχιστων Τετραγώνων (Least squares) για ένα ορισμένο παράθυρο δεδομένων εκπαίδευσης. Αυτό σημαίνει ότι αφενός ο αλγόριθμος είναι ιδιαίτερα αργός όταν το δίκτυο προσθέτει ή αφαιρεί ένα κέντρο, αφετέρου υπάρχει ενδεχομένως σημαντική απώλεια στην πληροφορία που κρύβεται σε δεδομένα από παλαιότερες εκπαιδεύσεις από το μέγεθος του παραθύρου, οι οποίες χάνονται κατά την νέα εκπαίδευση. Ένα, επίσης, σημαντικό μειονέκτημα αυτής της τακτικής είναι η απαίτηση επιπλέον μνήμης. Και αυτό διότι ο αλγόριθμος θα πρέπει να κρατάει ένα πλήθος δεδομένων εκπαίδευσης βάσει ενός παραθύρου προκειμένου να επανεκπαιδεύσει το δίκτυο σε περίπτωση ανάγκης προσθήκης ή αφαίρεσης ενός κέντρου.

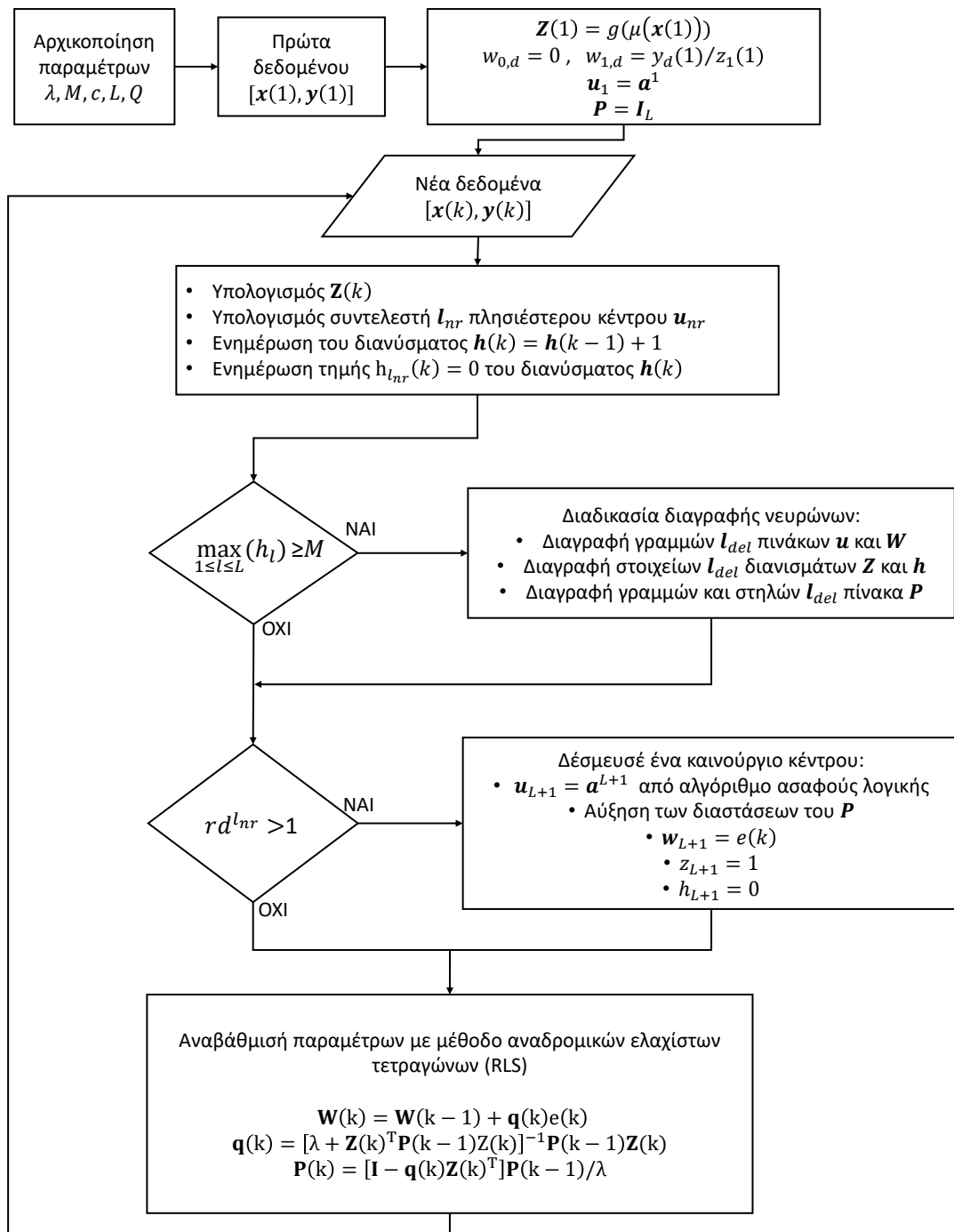
Στο κεφάλαιο αυτό παρουσιάζεται και προτείνεται μία νέα μέθοδος που θα την αποκαλούμε για λόγους ευκολίας FM-RLS, η οποία αποτελεί και το αντικείμενο της πτυχιακής εργασίας. Η διαδικασία FM-RLS στόχο έχει την αντιμετώπιση των μειονεκτημάτων του αλγορίθμου FUZZY-LS-RLS και της οικογένειας RAN διατηρώντας δεξιοτεχνικά τα πλεονεκτήματα και από τους δυο προγόνους της, στοχεύοντας σε μία γρήγορη και πιο οικονομική (σε θέματα μνήμης αλλά υπολογιστικό κόστος) μέθοδο εκπαίδευσης νευρωνικών δικτύων RBF.

4.2 Επισκόπηση προτεινόμενης μεθοδολογίας FM-RLS

Ένα πλεονέκτημα της διαδικασίας εκπαίδευσης των νευρωνικών δικτύων RBF, όπως αναφέρεται και στην [16], είναι ότι μπορεί να διαχωριστεί σε δύο στάδια εκπαίδευσης. Ο αλγόριθμος ασαφούς λογικής, που αναλύεται στην ενότητα 2.6.3, αναλαμβάνει το στάδιο υπολογισμού τοποθέτησης των κέντρων των ακτινικών συναρτήσεων βάσης. Κατά αυτόν τον τρόπο ο αλγόριθμος χρειάζεται να εκπαιδεύσει μόνο τα συνάπτικα βάρη και τα πλάτη των συναρτήσεων βάσης. Ο προτεινόμενος αλγόριθμος διατηρεί την τεχνική της ασαφούς λογικής. Έτσι, με την άφιξη ενός νέου δεδομένου, αυτό ταξινομείται στον πλησιέστερο από τους ήδη δεσμευμένους ασαφείς υπόχωρους σύμφωνα με την σχέση 2.21. Εάν το νέο δεδομένο βρίσκεται αρκετά μακριά από οποιοδήποτε δεσμευμένο ασαφή υπόχωρο, τότε δεσμεύεται ένας νέος ασαφής υπόχωρος με βάση τα ασαφή υποσύνολα που ενεργοποιούνται από αυτό το νέο δεδομένο (κριτήριο ελάχιστης απόστασης σχέση 2.22). Στην διαδικασία εκπαίδευσης του FUZZY-LS-RLS ο έλεγχος για την διαγραφή ενός κέντρου γίνεται εφόσον δεν χρειαστεί να προστεθεί κάποιο νέο κέντρο. Αντιθέτως στην προτεινόμενη διαδικασία FM-RLS ο έλεγχος διαγραφής για τα δεσμευμένα κέντρα, γίνεται σε κάθε άφιξη νέου δεδομένου κατά την έναρξη του κύκλου. Έτσι, με την άφιξη ενός νέου δεδομένου ο αλγόριθμος διαγράφει τα κέντρα για τα οποία ο ασαφής τους υπόχωρος δεν έχει ενεργοποιηθεί για ένα πλήθος M συνεχόμενων αφίξεων. Ύστερα από την λήξη της διαδικασίας ελέγχου διαγραφής ή προσθήκης ενός νέου κέντρου, ο αλγόριθμος αναβαθμίζει τα συνάπτικα βάρη του δικτύου σύμφωνα με την διαδικασία των αναδρομικών ελάχιστων τετραγώνων (RLS) που περιγράφεται στην ενότητα 2.6.1 από τις σχέσεις 3.7, 3.8 και 3.9. Είναι σημαντικό να σημειωθεί ότι κατά την διαδικασία διαγραφής ενός κέντρου ο αλγόριθμος πλέον δεν διαγράφει μόνο το διάνυσμα του κέντρου, αλλά διαγράφει το συναπτικό βάρος και το πλάτος της ακτινικής συνάρτησης βάσης. Στην συνέχεια μειώνει τις διαστάσεις του αντίστροφου πίνακα συνδιακύμανσης, το οποίο συμβαίνει και στην διαδικασία διαγραφής του αλγορίθμου M-RAN (βλέπε ενότητα 3.5.2). Στην περίπτωση που ο αλγόριθμος θα χρειαστεί να προσθέσει ένα νέο κέντρο, η τοποθεσία του νέου κέντρου καθορίζεται από τη διαδικασία ασαφούς λογικής. Ταυτόχρονα αυξάνονται και οι διαστάσεις του διανύσματος των συναπτικών βαρών, των πλατών των ακτινικών συναρτήσεων βάσης και του πίνακα P με την ίδια διαδικασία του αλγορίθμου RAN-EKF (βλέπε σχέση 3.22 ενότητα 3.5.2).

Σε αυτό το σημείο είναι σημαντικό να σημειωθεί ότι η προτεινόμενη διαδικασία FM-RLS, όπως προαναφέρθηκε, στοχεύει στην απλοποίηση της πολυεπίπεδης επιφάνειας σφάλματος. Αυτό επιτυγχάνεται διαχωρίζοντας το πρόβλημα σε δύο στάδια, αφήνοντας τον αλγόριθμο της ασαφούς λογικής να αναλάβει την ταξινόμηση και την τοποθέτηση των κέντρων. Έτσι, ο αλγόριθμος των αναδρομικών ελάχιστων τετραγώνων αναλαμβάνει να προσεγγίσει μία λιγότερο πολύπλοκη συνάρτηση. Προκειμένου να απλοποιήσουμε ακόμα περισσότερο την πολυεπίπεδη επιφάνεια σφάλματος, η ακτινική συνάρτηση βάσης την οποία χρησιμοποιούμε είναι η “Thin Plate Spline”. Η συγκεκριμένη συνάρτηση δεδομένου ότι δεν χρησιμοποιεί την παράμετρο πλάτους “ σ ”, μειώνει τις μεταβλητές εκπαίδευσης και ως εκ τούτου απλοποιεί την πολυεπίπεδη επιφάνεια σφάλματος.

Στο σχήμα 24. παρουσιάζεται αναλυτικά σε μορφή ψευδοκώδικα η προτεινόμενη μεθοδολογία εκπαίδευσης FM-RLS.



Εικόνα 24. Διάγραμμα ψευδοκώδικα προτεινόμενης μεθοδολογίας FM-RLS.

4.3 Υποδομή νευρωνικού δικτύου RBF

Στην online διαδικασία εκπαίδευσης μία ροή δεδομένων $[\mathbf{x}(k), y(k)]$ εκπαίδευσης παρέχεται στο δίκτυο με στόχο την εκπαίδευσή του όπου $\mathbf{x}(k) = [x_1(k), x_2(k), \dots, x_i(k), \dots, x_N(k)]$ να είναι ένα διάνυσμα $1 \times N$ διαστάσεων με $i = 1, 2, \dots, N$. Θεωρώντας ότι το RBF δίκτυο έχει ήδη λάβει ένα πλήθος δεδομένων εκπαίδευσης k και ότι έχει σχηματιστεί ένα δίκτυο που αποτελείται από L

νευρώνες, η πρόβλεψη του δικτύου για το δεδομένο k δίνεται από την σχέση 4.1 με $l = 0, 1, 2, \dots, L$

$$\hat{y}(k) = \sum_{l=0}^L z_l(k)w_l \quad (4.1)$$

Όπου w_l είναι το συναπτικό βάρος και z_l είναι η έξοδος της κρυφής στιβάδας του νευρώνα l . Τα συναπτικά βάρη του δικτύου, καθώς και οι έξοδοι της κρυφής στιβάδας μπορούν να γραφούν σε μορφή διανύσματος $\mathbf{W}^T = [w_0, w_1, w_2, \dots, w_l, \dots, w_L]$ και $\mathbf{Z}^T = [z_0, z_1, z_2, \dots, z_l, \dots, z_L]$ με διαστάσεις $L \times 1$ αντίστοιχα. Είναι σημαντικό να σημειωθεί ότι η προτεινόμενη διαδικασία χρησιμοποιεί πόλωση που σημαίνει ότι το νευρωνικό δίκτυο εκτός από τους νευρώνες της κρυφής στιβάδας που αντιστοιχούν στα δεσμευμένα κέντρα, περιλαμβάνει και ένα επιπλέον σταθερό όρο τον οποίο συμβολίζουμε με z_0 και η έξοδος του είναι ίση με “1”. Αντίστοιχα το συναπτικό βάρος του συγκεκριμένου όρου συμβολίζεται με w_0 .

Το νευρωνικό δίκτυο της διαδικασίας που προτείνουμε στην πραγματικότητα δεν περιλαμβάνει μόνο μία έξοδο. Σε αντίθεση με τις υλοποιήσεις της οικογένειας RAN και της FUZZY-LS-RLS που τα δίκτυα είναι σχεδιασμένα σε μορφή πολλών εισόδων μιας εξόδου (SISO, Single Input Single Outlet), το δίκτυο της προτεινόμενης διαδικασίας είναι υλοποιημένο σε μορφή πολλών εισόδων πολλών εξόδων (MIMO, Multi input multi output). Το οποίο σημαίνει για ένα πρόβλημα κατηγοριοποίησης προτύπων D κλάσεων με $D \in \mathbb{N}^+ = 1, 2, \dots$ ότι το δεδομένο εκπαίδευσης $y(k)$ πρέπει να μετατραπεί σε ένα διάνυσμα $D \times 1$ διαστάσεων όσες είναι και οι κλάσεις του προβλήματος κατηγοριοποίησης. Θεωρώντας ότι $1 \leq y(k) \leq D$ και $y(k) \in \mathbb{N}^+$ τότε οι συντελεστές του διανύσματος $\mathbf{y}(k)^T = [y_1(k), y_2(k), \dots, y_d(k), \dots, y_D(k)]$ προκύπτουν από την σχέση 4.2 με $d = 1, 2, \dots, D$.

$$y_d(k) = \begin{cases} 1 & \text{εάν } d = y(k) \\ 0 & \text{αλλιού} \end{cases} \quad (4.2)$$

Έτσι, για παράδειγμα για ένα πρόβλημα κατηγοριοποίησης πέντε κλάσεων θα έχουμε $D = 5$. Εάν θεωρήσουμε ότι λαμβάνεται ένα δεδομένο εκπαίδευσης $y(k) = 3$, τότε σύμφωνα με την σχέση 4.2 θα πρέπει να μετατραπεί σε ένα διάνυσμα της μορφής $\mathbf{y}(k)^T = [0, 0, 1, 0, 0]$. Εδώ πρέπει να αναφέρουμε ότι σε πραγματικά προβλήματα κατηγοριοποίησης οι κλάσεις δεν είναι απαραίτητο να λαμβάνουν τιμές στο πεδίο $1 \leq y(k) \leq D$ και $y(k) \in \mathbb{N}^+$. Σε αυτήν την περίπτωση θα χρειαστεί σαφώς να γίνει μία αντιστοίχιση των κλάσεων του προβλήματος σε πραγματικούς θετικούς αριθμούς του συνόλου \mathbb{N}^+ .

Αφού δείξαμε τον τρόπο με τον οποίο μετασχηματίζονται τα δεδομένα εκπαίδευσης $y(k)$ για να προσαρμοστούν στην διαδικασία εκπαίδευσης της μεθοδολογίας, μπορούμε πλέον να αναβαθμίσουμε την σχέση 4.1 σε μία σχέση για ένα δίκτυο με πολλές εξόδους. Θεωρώντας ένα δίκτυο με εξόδους $\hat{y}_d(k)$ με $d = 1, 2, \dots, D$ τότε η πρόβλεψη της εξόδου $\hat{y}_d(k)$ θα δίνεται από την σχέση 4.3.

$$\hat{y}_d(k) = \sum_{l=0}^L z_l(k)w_{l,d} \quad (4.3)$$

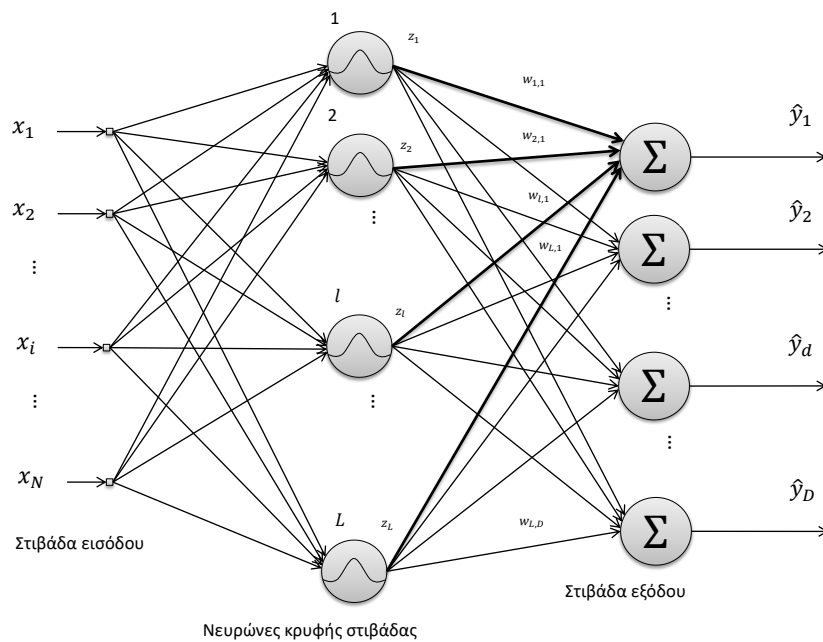
Όπως μπορούμε να δούμε από την σχέση 4.3 τα συναπτικά βάρη $w_{l,d}$ πλέον διακρίνονται από δύο συντελεστές, το οποίο σημαίνει ότι το διάνυσμα \mathbf{W} που περιείχε όλα τα συνοπτικά βάρη για την περίπτωση της μίας εξόδου, στην περίπτωση των πολλών εξόδων θα πρέπει πλέον να είναι ένας πίνακας διαστάσεων $L \times D$, όπως φαίνεται στη σχέση 4.4.

$$\mathbf{W} = \begin{bmatrix} w_{0,1} & w_{0,2} & \dots & w_{0,d} & \dots & w_{0,D} \\ w_{1,1} & w_{1,2} & \dots & w_{1,d} & \dots & w_{1,D} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{l,1} & w_{l,2} & \dots & w_{l,d} & \dots & w_{l,D} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{L,1} & w_{L,2} & \dots & w_{L,d} & \dots & w_{L,D} \end{bmatrix} \quad (4.4)$$

Οι προβλέψεις των εξόδων y_d του νευρωνικού δικτύου μπορούν να γραφούν σε μορφή διανύσματος $\hat{\mathbf{y}}^T(k) = [\hat{y}_1(k), \hat{y}_2(k), \dots, \hat{y}_D(k)]$. Το διάνυσμα $\hat{\mathbf{y}}(k)$ μπορεί να υπολογιστεί με μία πράξη πολλαπλασιασμού πινάκων από την σχέση 4.5.

$$\hat{\mathbf{y}}(k) = \mathbf{W}^T \times \mathbf{Z}(k) \quad (4.5)$$

Οι σχέσεις 4.3 και 4.5 υπολογίζουν τις προβλέψεις εξόδων του νευρωνικού δικτύου. Η μία για μία μεμονωμένη έξοδο d , ενώ η άλλη δίνει απευθείας το διάνυσμα πρόβλεψης $\hat{\mathbf{y}}$ για όλες τις εξόδους. Η δομή του νευρωνικού δικτύου όπως το περιγράψαμε το οποίο θα χρησιμοποιηθεί στην διαδικασία FM-RLS παρουσιάζεται στο σχηματικό διάγραμμα του σχήματος 25.



Εικόνα 25. Δομή νευρωνικού δικτύου προτεινόμενης διαδικασίας.

4.3.1 Αρχικοποίηση παραμέτρων και εισαγωγή πρώτων δεδομένων

Κατά την έναρξη της διαδικασίας ο αλγόριθμος αρχικοποιεί τις παραμέτρους λ , M , c , και Q όπου λ είναι ο συντελεστής απόσβεσης του αλγόριθμου αναδρομικών ελαχίστων τετραγώνων, M είναι ένας αριθμός που καθορίζει το πλήθος των συνεχόμενων επαναλήψεων κατά τον οποίο εάν ένας νευρώνας δεν έχει ενεργοποιηθεί τότε θα διαγραφεί, c_i το πλήθος των ασαφών συνόλων ανά μεταβλητή εισόδου i και Q είναι μία σταθερά που χρησιμοποιείται κατά την διαδικασία αύξησης των διαστάσεων του πίνακα \mathbf{P} . Δεν πρέπει να παραλείψουμε ότι η σταθερά Q δεν έχει καμία σύνδεση με τη σταθερά Q_0 του αλγόριθμου RAN-EKF. Επιπρόσθετα, το πλήθος των ασαφών συνόλων της προτεινόμενης διαδικασίας θα είναι ίδιο για όλες τις διαστάσεις που σημαίνει ότι $c_1 = c_2 = c_i = c_N = c$. Κρατώντας ως δεδομένο αυτό, το πλήθος των ασαφών συνόλων για όλες τις διαστάσεις θα συμβολίζεται με c .

Μετά την αρχικοποίηση των σταθερών παραμέτρων ο αλγόριθμος χρησιμοποιεί τα πρώτα δεδομένα $\mathbf{x}(1), \mathbf{y}(1)$ για να υπολογίσει το πρώτο κέντρο. Το διάνυσμα εισόδου $\mathbf{x}(\mathbf{k})$ μπορεί να γραφεί αναλυτικά στην μορφή $\mathbf{x}(\mathbf{k}) = [x_1(k), x_2(k), \dots, x_i(k), \dots, x_N(k)]$ με $i = 1, 2, \dots, N$ όπου N το πλήθος των διαστάσεων του διανύσματος \mathbf{x} και k ο αριθμός της ουράς των δεδομένων

εκπαίδευσης. Η προτεινόμενη διαδικασία δεσμεύει το πρώτο κέντρο με χρήση του αλγόριθμου ασαφούς λογικής. Σύμφωνα με τον συγκεκριμένο αλγόριθμο ο πλησιέστερος ασαφής υπόχωρος $\mathbf{A}^1 = [A_{1,j_1}^1, A_{2,j_2}^1, \dots, A_{N,j_N}^1]$ για το δεδομένο $\mathbf{x}(1)$ υπολογίζεται με βάση το κριτήριο ελάχιστης απόστασης και οι συντελεστές j_i για την κάθε διάσταση i υπολογίζονται από την σχέση 4.6 με $j = 1, 2, \dots, c$ και όπου $k = 1$.

$$j_i = \arg \max_{1 \leq j \leq c} (\mu_{A_{i,j}}(x_i(k))) \quad (4.6)$$

Αφού υπολογιστεί ο πλησιέστερος ασαφής υπόχωρος από το δεδομένο $\mathbf{x}(k)$, εισάγεται ο πίνακας των κέντρων \mathbf{u} ο οποίος θα περιλαμβάνει τις συντεταγμένες των κέντρων για όλους τους νευρώνες της κρυφής στιβάδας. Κατά την προσθήκη του πρώτου κέντρου ο πίνακας θα πρέπει να έχει διαστάσεις $1 \times N$, ενώ στην συνέχεια όταν θα συμπληρωθούν επιπλέον κέντρα οι διαστάσεις του θα είναι $L \times N$ και θα έχει την μορφή της σχέσης 4.7 όπου $l = 1, 2, \dots, L$.

$$\mathbf{u} = \begin{bmatrix} u_{1,1} & u_{1,2} & \dots & u_{1,i} & \dots & u_{1,N} \\ u_{2,1} & u_{2,2} & \dots & u_{2,i} & \dots & u_{2,N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ u_{l,1} & u_{l,2} & \dots & u_{l,i} & \dots & u_{l,N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ u_{L,1} & u_{L,2} & \dots & u_{L,i} & \dots & u_{L,N} \end{bmatrix} \quad (4.7)$$

Οι συντεταγμένες του l κέντρου της κρυφής στιβάδας μπορούν να γραφούν σε μορφή διάνυσματος $\mathbf{u}_l = [u_{l,1}, u_{l,2}, \dots, u_{l,N}]$ με τον συντελεστή l να υποδεικνύει τον l νευρώνα και να αντιστοιχεί στην l γραμμή του πίνακα \mathbf{u} . Άρα, για το πρώτο κέντρο έχουμε $\mathbf{u}_1 = [u_{1,1}, u_{1,2}, \dots, u_{1,N}] = [\alpha_{1,j_1}^1, \alpha_{2,j_2}^1, \dots, \alpha_{N,j_N}^1] = \mathbf{a}^1$ όπου \mathbf{a}^1 είναι το διάνυσμα της θέσης του κέντρου του πλησιέστερου ασαφή υπόχωρου στο δεδομένο $\mathbf{x}(k)$. Αφού εισάγουμε τις συντεταγμένες του πρώτου κέντρου στον πίνακα \mathbf{u} , ο αλγόριθμος δεσμεύει ένα διάνυσμα ιστορικού $\mathbf{h}(k) = [h_1(k), h_2(k), \dots, h_l(k), \dots, h_L(k)]$ με διαστάσεις $1 \times L$. Το διάνυσμα $\mathbf{h}(k)$ απαριθμεί τις συνεχόμενες φορές κατά τις οποίες ένα κέντρο δεν βρίσκεται επαρκώς κοντά στα δεδομένα $\mathbf{x}(k)$. Για την δέσμευση του πρώτου κέντρου το διάνυσμα $h(1)$ θα έχει διαστάσεις 1×1 και θα λάβει την τιμή $h(1) = [h_1(1)] = 0$. Το διάνυσμα ιστορικότητας $\mathbf{h}(k)$ δημιουργείται με σκοπιμότητα να ποσοτικοποιήσει με έναν τρόπο την δραστηριότητα των κέντρων. Σε επόμενες ενότητες θα εξετάσουμε περαιτέρω την διαδικασία ελέγχου διαγραφής των κέντρων και θα αναφερθούμε αναλυτικότερα στο διάνυσμα $h(k)$.

Μετά την ολοκλήρωση της διαδικασίας του υπολογισμού του πρώτου κέντρου ο αλγόριθμος θα χρειαστεί να υπολογίσει τον πίνακα $\mathbf{Z}^T = [z_0, z_1, z_2, \dots, z_l, \dots, z_L]$ που θα περιλαμβάνει την έξοδο των νευρώνων της κρυφής στιβάδας, τον πίνακα των συντελεστών του νευρωνικού δικτύου $\mathbf{W}^T = [w_0, w_1, w_2, \dots, w_l, \dots, w_L]$, καθώς και τον αντίστροφο του πίνακα συνδιακύμανσης \mathbf{P} του αλγόριθμου αναδρομικών ελάχιστων τετραγώνων. Ο υπολογισμός της εξόδου της κρυφής στιβάδας $Z(k)$ δίνεται από την σχέση $g(\mu(k))$ ο οποίος, όπως έχουμε προαναφέρει σε προηγούμενα κεφάλαια, είναι ο υπολογισμός της ακτινικής συνάρτησης $g(\bullet)$ με όρισμα την Ευκλείδεια απόσταση $\mu(k) = \|\mathbf{u} - \mathbf{x}(k)\|_2$ του διανύσματος $x(k)$ από το διάνυσμα των κέντρων \mathbf{u} . Στην διαδικασία FM-RLS η ακτινική συνάρτηση βάση που θα χρησιμοποιηθεί είναι η Thin-Plate-Spline που προκύπτει από την σχέση: $g(\mu(k)) = \mu(k)^2 \ln(\mu(k))$. Με βάση τα παραπάνω η έξοδος της κρυφής στιβάδας δίνεται από την σχέση 4.8.

$$\mathbf{Z}(k) = \|\mathbf{u} - \mathbf{x}(k)\|_2^2 \ln(\|\mathbf{u} - \mathbf{x}(k)\|_2) \quad (4.8)$$

Θέτοντας όπου $k = 1$ για το πρώτο δεδομένο μπορούμε να υπολογίσουμε την έξοδο της κρυφής στιβάδας για αυτό. Στην συνέχεια προκειμένου να μπορέσουμε να υπολογίσουμε εύκολα τα συναπτικά βάρη για το πρώτο δεδομένο θα θέσουμε τα συναπτικά βάρη των πωλήσεων $w_{0,d}$ όλων των εξόδων να είναι 0. Με αυτόν τον τρόπο μπορούμε να υπολογίσουμε τα βάρη $w_{1,d}$ για τα οποία

θα πρέπει να ισχύει ότι $w_{1,d} = y_d(1)/z_1(1)$. Στην συνέχεια η διαδικασία δομεί τον πίνακα $\mathbf{P} = \mathbf{I}_L$ σαν ένα μοναδιαίο διαγώνιο πίνακα διαστάσεων $L \times L$. Προτού ο αλγόριθμος δεχθεί το επόμενο δεδομένο της ροής δεδομένων, υλοποιεί την αντίστοιχη πρόβλεψη του νευρωνικού δικτύου. Ο υπολογισμός της πρόβλεψης $\hat{y}(2)$ για το επόμενο σε σειρά δεδομένο εκπαίδευσης $\mathbf{x}(2)$ υπολογίζεται σύμφωνα με την σχέση 4.5 ως εξής $\hat{y}(2) = \mathbf{W}^T \times \mathbf{Z}(2)$ με το $\mathbf{Z}(2)$ να δίνεται από την σχέση 4.8 για $k = 2$.

4.3.2 Άφιξη νέων δεδομένων, έλεγχος και διαγραφή κέντρων

Μετά την ολοκλήρωση της αρχικοποίησης όλων των απαραίτητων παραμέτρων, καθώς και τον υπολογισμό για τα πρώτα δεδομένα $\mathbf{x}(1), y(1)$, η προτεινόμενη διαδικασία είναι έτοιμη να λάβει το επόμενο δεδομένο k της ροής. Κατά την άφιξη του νέου δεδομένου k το πρώτο πράγμα που κάνει ο αλγόριθμος είναι να υπολογίσει την νέα έξοδο των νευρώνων της κρυφής στιβάδας $\mathbf{Z}(k) = \|\mathbf{u} - \mathbf{x}(k)\|_2^2 \ln(\|\mathbf{u} - \mathbf{x}(k)\|_2)$. Στην συνέχεια υπολογίζεται η σχετική Ευκλείδεια απόσταση $rd^l(x(k))$ του διανύσματος $x(k)$ από όλα τα δεσμευμένα κέντρα \mathbf{u} και διατηρείται η τιμή της απόστασης $rd^{nr}(x(k))$ η οποία αντιστοιχεί στο πλησιέστερο κέντρο $x(k)$ που υπολογίζεται από την σχέση 4.9.

$$rd^{nr}(x(k)) = \min_{1 \leq l \leq L} |rd^l(x(k))| \quad (4.9)$$

Συμβολίζουμε με l_{nr} τον δείκτη του κέντρου l από το οποίο το δεδομένο $x(k)$ θα έχει τη μικρότερη σχετική Ευκλείδεια απόσταση και προκύπτει από την σχέση 4.10.

$$l_{nr} = \mathbf{arg} \min_{1 \leq l \leq L} |rd^l(x(k))| \quad (4.10)$$

Η γνώση του δείκτη l_{nr} του πλησιέστερου κέντρου \mathbf{u}_{nr} στο νέο δεδομένο $x(k)$, καθώς και η σχετική Ευκλείδεια απόσταση $rd^{nr}(x(k))$ θα χρησιμοποιηθεί παρακάτω για να εκτιμηθεί εάν χρειάζεται να αφαιρεθεί ή να προστεθεί ένα κέντρο από το νευρωνικό δίκτυο. Θα επανέλθουμε στην διαδικασία ελέγχου και προσθήκης νέου κέντρου σε επόμενη ενότητα. Στην συνέχεια, ο αλγόριθμος ενημερώνει το διάνυσμα ιστορικότητας $h(k)$. Κατά την άφιξη ενός νέου δεδομένου k το διάνυσμα ιστορικότητας αναβαθμίζεται σύμφωνα με την σχέση 4.11.

$$h(k) = h(k-1) + 1 \quad (4.11)$$

Κατόπιν, ο αλγόριθμος μηδενίζει το l_{nr} στοιχείο του διανύσματος $\mathbf{h}(k)$ που αντιστοιχεί στην ιστορικότητα του πλησιέστερου κέντρου στο δεδομένο k και επομένως το $h_{l_{nr}}(k) = 0$. Με αυτόν τον τρόπο μέσω του διανύσματος ιστορικότητας $h(k)$, ο αλγόριθμος μπορεί να διακρίνει πόσα συνεχόμενα δεδομένα έχουν προηγηθεί χωρίς να έχει ενεργοποιηθεί ο ασαφής υπόχωρος ο οποίος αντιστοιχεί στο εκάστοτε l κέντρο.

Μετά την ολοκλήρωση της διαδικασίας ενημέρωσης του διανύσματος ιστορικότητας, ο αλγόριθμος περνά στην διαδικασία ελέγχου διαγραφής κέντρου. Σε αυτό το στάδιο ο αλγόριθμος ελέγχει το διάνυσμα ιστορικότητας $\mathbf{h}(k)$. Εάν κάποια από τις τιμές του είναι μεγαλύτερες ή και ίσες από την τιμή μίας σταθεράς M , δηλαδή ένα $\max_{1 \leq l \leq L} (h_l(k)) \geq M$, ο αλγόριθμος διαγράφει το κέντρο και τον νευρώνα της κρυφής στιβάδας που αντιστοιχεί στο συγκεκριμένο στοιχείο. Ο συντελεστής l της θέσης καθορίζεται από την σχέση 4.12 και συμβολίζεται με l_{del} .

$$l_{del} = \mathbf{arg} \max_{1 \leq l \leq L} (h_l(k)) \quad (4.12)$$

Σύμφωνα με την σχέση 4.12 ο δείκτης l_{del} υποδεικνύει την θέση των νευρώνων που θα πρέπει να διαγραφούν. Στην περίπτωση στην οποία παραπάνω από ένας νευρώνας έχει λάβει τιμή ίση του M , ο δείκτης l_{del} πρόκειται για ένα διάνυσμα $\mathbf{l}_{del} = [l_{del_1}, l_{del_2}, \dots, l_{del_r}]$ το οποίο θα περιλαμβάνει τις

θέσεις για όλα τα στοιχεία που πρέπει να διαγραφούν. Αφού εντοπιστεί η θέση των κέντρων που έχουν παραμείνει ανενεργά για ένα πλήθος δεδομένων M . Η διαδικασία συνεχίζει με την διαγραφή αυτών των κέντρων από τον πίνακα \mathbf{u} ο οποίος διατηρεί τις συντεταγμένες για όλα τα κέντρα των νευρώνων της κρυφής στιβάδας. Οι γραμμές των κέντρων που πρέπει να διαγραφούν από τον πίνακα \mathbf{u} υποδεικνύονται από τον δείκτη \mathbf{I}_{del} . Σύμφωνα με αυτόν, ο νέος πίνακας $\mathbf{u}(k)$ θα προκύψει μετά την αφαίρεση των γραμμών $\mathbf{u}_{I_{del_1}}, \mathbf{u}_{I_{del_2}}, \dots, \mathbf{u}_{I_{del_r}}$ από τον πίνακα $\mathbf{u}(k-1)$. Ο νέος πίνακας $\mathbf{u}(k)$ που προκύπτει θα πρέπει να έχει διαστάσεις $(L-r) \times N$. Μετά την αναδόμηση του πίνακα $\mathbf{u}(k)$ ο αλγόριθμος συνεχίζει με την αφαίρεση των συναπτικών βαρών από τον πίνακα \mathbf{W} . Με παρόμοιο τρόπο ο νέος πίνακας $\mathbf{W}(k)$ θα προκύψει από την αφαίρεση των γραμμών $\mathbf{w}_{I_{del_1}}, \mathbf{w}_{I_{del_2}}, \dots, \mathbf{w}_{I_{del_r}}$ από τον πίνακα $\mathbf{W}(k-1)$. Στην συνέχεια ο αλγόριθμος προχωρά με προσαρμογή των διανυσμάτων $\mathbf{Z}(k)$ και $\mathbf{h}(k)$, διαγράφοντας τα στοιχεία $z_{I_{del_1}}, z_{I_{del_2}}, \dots, z_{I_{del_r}}$ και $h_{I_{del_1}}, h_{I_{del_2}}, \dots, h_{I_{del_r}}$ των διανυσμάτων $\mathbf{Z}(k-1)$ και $\mathbf{h}(k-1)$ αντίστοιχα. Η διαδικασία διαγραφής νευρώνων θα ολοκληρωθεί με την αναπροσαρμογή του πίνακα \mathbf{P} . Στην περίπτωση των πινάκων \mathbf{u} και \mathbf{W} η διαδικασία διαγράφει μόνο τις γραμμές. Αντίθετα στην περίπτωση του πίνακα \mathbf{P} ο αλγόριθμος θα χρειαστεί να διαγράψει τις ενδεικνύμενες γραμμές, αλλά και τις στήλες που υποδεικνύουν οι συντελεστές του διανύσματος \mathbf{I}_{del} . Ο νέος πίνακας $\mathbf{P}(k)$ θα προκύψει από την διαγραφή των γραμμών και των στηλών $\mathbf{p}_{I_{del_1}}, \mathbf{p}_{I_{del_2}}, \dots, \mathbf{p}_{I_{del_r}}$ από τον πίνακα $\mathbf{P}(k-1)$

4.3.3 Έλεγχος και προσθήκη νέου νευρώνα

Μετά την ολοκλήρωση της διαδικασίας ελέγχου και διαγραφής νευρώνων ο αλγόριθμος περνά αμέσως στην διαδικασία ελέγχου και προσθήκης ενός νέου νευρώνα. Για να πάρει την απόφαση αν χρειάζεται να προστεθεί ένα νέο κέντρο στο νευρωνικό δίκτυο, ο αλγόριθμος εξετάζει πόσο κοντά βρίσκεται το νέο δεδομένο $x(k)$ στα ήδη δεσμευμένα κέντρα. Εάν η σχετική ευκλείδεια απόσταση $rd^{nr}(x(k))$ του πλησιέστερου κέντρου βρεθεί να είναι μεγαλύτερη της μονάδας “1”, τότε ο αλγόριθμος θεωρεί ότι τα υπάρχοντα κέντρα δεν είναι επαρκώς κοντά για να εξομοιωθούν το νέο δεδομένο. Στην περίπτωση αυτή ο αλγόριθμος συνεχίζει την διαδικασία προσθήκης νέου κέντρου με χρήση της ασαφούς λογικής. Με βάση την ασαφή λογική οι συντελεστές j_i του πλησιέστερου ασαφούς υπόχωρου $\mathbf{A}^{L+1} = [A_{1,j_1}^{L+1}, A_{2,j_2}^{L+1}, \dots, A_{N,j_N}^{L+1}]$ στο δεδομένο $x(k)$ υπολογίζονται από την σχέση 4.6. Ολοκληρώνοντας την διαδικασία υπολογισμού του ασαφής υπόχωρου \mathbf{A}^{L+1} , ο αλγόριθμος αποθηκεύει το νέο κέντρο \mathbf{a}^{L+1} του υπόχωρου \mathbf{A}^{L+1} σε μία νέα γραμμή $\mathbf{u}_{L+1} = [u_{L+1,1}, u_{L+1,2}, \dots, u_{L+1,N}] = [\alpha_{1,j_1}^{L+1}, \alpha_{2,j_2}^{L+1}, \dots, \alpha_{N,j_N}^{L+1}] = \mathbf{a}^{L+1}$ του πίνακα \mathbf{u} . Ο νέος πίνακας \mathbf{u}_{new} προκύπτει από τον προηγούμενο πίνακα \mathbf{u}_{old} ως εξής $\mathbf{u}_{new}^T = [\mathbf{u}_{old}^T, \mathbf{u}_{L+1}^T]$. Στην συνέχεια αυξάνει το διάνυσμα ιστορικότητας κατά ένα στοιχείο δίνοντας του την τιμή μηδέν $h_{L+1} = 0$ το οποίο διατηρεί την δραστηριότητα του νέου νευρώνα και, άρα, $\mathbf{h}_{new} = [\mathbf{h}_{old}, h_{L+1}]$. Μετά την προσθήκη του νέου κέντρου στον πίνακα \mathbf{u} ο αλγόριθμος συνεχίζει με την προσθήκη των συναπτικών βαρών του νευρώνα που δημιουργήθηκε, αλλά και με την αύξηση των διαστάσεων των πινάκων \mathbf{Z} και \mathbf{P} . Τα νέα συναπτικά βάρη w_{L+1} θα αρχικοποιηθούν σύμφωνα με το σφάλμα της τελευταίας πρόβλεψης. Οπότε προκύπτει ότι $w_{L+1} = |\mathbf{y}^T(k) - \hat{\mathbf{y}}^T(k)|$ όπου $\mathbf{y}(k)$ να είναι η πρόβλεψη του δικτύου για το δεδομένο k . Ο νέος πίνακας \mathbf{W}_{new} θα προκύψει από τον προηγούμενο πίνακα \mathbf{W}_{old} ως εξής: $\mathbf{W}_{new}^T = [\mathbf{W}_{old}^T, w_{L+1}^T]$. Το διάνυσμα \mathbf{Z} αυξάνεται κατά ένα στοιχείο το οποίο παίρνει την τιμή ένα, δηλαδή, $z_{L+1} = 1$ και το νέο διάνυσμα θα είναι $\mathbf{Z}_{new}^T = [\mathbf{Z}_{old}^T, z_{L+1}^T]$. Τέλος, η διαδικασία ολοκληρώνεται με την αναβάθμιση του πίνακα \mathbf{P} . Οι διαστάσεις του πίνακα \mathbf{P} αυξάνονται κατά ένα και ο νέος πίνακας \mathbf{P}_{new} ο οποίος θα προκύψει θα έχει διάσταση $(L+1) \times (L+1)$ και δίνεται από την σχέση 4.13.

$$\mathbf{P}_{new} = \begin{bmatrix} \mathbf{P}_{old} & 0 \\ 0 & Q \end{bmatrix} \quad (4.13)$$

Σύμφωνα με την σχέση 4.13 ο νέος πίνακας P_{new} που θα προκύψει θα συμπληρωθεί στις γραμμές και τις στήλες του με μηδέν, ενώ το τελευταίο διαγώνιο στοιχείο θα λάβει την τιμή Q . Το Q είναι μία σταθερά που έχει αρχικοποιηθεί κατά την έναρξη του αλγορίθμου με μία τιμή μεγάλης τάξης μεγέθους (συνήθως $Q \gg 100$).

4.3.4 Αναβάθμιση παραμέτρων με την μέθοδο αναδρομικών ελαχίστων τετραγώνων

Μετά την λήξη της διαδικασίας ελέγχου και προσθήκης νέου νευρώνα, ο αλγόριθμος περνά στην διαδικασία αναβάθμισης παραμέτρων με την Μέθοδο Αναδρομικών Ελαχίστων Τετραγώνων (Recursive Least Squares (RLS)). Δεδομένου ότι οι διαστάσεις των πινάκων έχουν προσαρμοστεί καταλλήλως από τα περασμένα βήματα του αλγορίθμου, τα συναπτικά βάρη μπορούν πλέον να αναβαθμιστούν με χρήση του αλγορίθμου RLS για το νέο δεδομένο $x(k)$ σύμφωνα με τις σχέσεις 4.14, 4.15 και 4.16.

$$\mathbf{W}(k) = \mathbf{W}(k-1) + \mathbf{q}(k)(\mathbf{y}(k) - \mathbf{Z}^T(k)\mathbf{W}(k-1)) \quad (4.14)$$

$$\mathbf{q}(k) = \mathbf{P}(k-1)\mathbf{Z}(k)(\lambda + \mathbf{Z}^T(k)\mathbf{P}(k-1)\mathbf{Z}(k))^{-1} \quad (4.15)$$

$$\mathbf{P}(k) = (\mathbf{I} - \mathbf{q}(k)\mathbf{Z}^T(k))\mathbf{P}(k-1)/\lambda \quad (4.16)$$

όπου λ είναι ο συντελεστής απόσβεσης ο οποίος αρχικοποιήθηκε κατά την έναρξη του αλγορίθμου. Τελειώνοντας την διαδικασία αναβάθμισης το νευρωνικό δίκτυο έχει πια εκπαιδευτεί για το δεδομένο $x(k)$. Κατόπιν, ο αλγόριθμος εκτελεί πρόβλεψη για το επόμενο δεδομένο εκπαίδευσης σε σειρά $k+1$ η οποία υπολογίζεται σύμφωνα με την σχέση 4.5 ως εξής: $\hat{\mathbf{y}}(k+1) = \mathbf{W}^T(k) \times \mathbf{Z}(k+1)$

Κεφάλαιο 5

Πείραμα και αποτελέσματα

5.1 Πειραματική διαδικασία

Στα πλαίσια της πειραματικής διαδικασίας υλοποιήθηκε ένα πλήθος από αλγόριθμους εκπαίδευσης με στόχο να συγκριθούν με την διαδικασία FM-RLS που προτείνεται στην συγκεκριμένη πτυχιακή εργασία. Όλοι οι αλγόριθμοι πραγματοποιήθηκαν σε γλώσσα προγραμματισμού MATLAB (R2020a). Ενώ η διαδικασία εκπαίδευσης καθώς και τα αποτελέσματα των δικτύων πραγματοποιήθηκαν σε υπολογιστή με επεξεργαστή (CPU) Intel(R) Core(TM) i5-7400 στα 3.00GHz και μνήμη RAM 8.00GB Single-Channel DDR4 στα 1200MHz. Οι εξομοιώσεις των δικτύων έγιναν σε λειτουργικό σύστημα Windows 10 Education. Για μεγαλύτερη ακρίβεια στους χρόνους των δικτύων και για την αποφυγή μεταβολών, η εφαρμογή MATLAB ήταν ρυθμισμένη να λειτουργεί σε προτεραιότητα "Realtime" (Ειδική λειτουργία των Windows) με οποιαδήποτε άλλη εφαρμογή του υπολογιστή να είναι κλειστή. Είναι σημαντικό να σημειωθεί ότι όλες οι υλοποιήσεις έγιναν με τον βέλτιστο δυνατό τρόπο για την αποφυγή "φλύαρου κώδικα", ο οποίος θα οδηγούσε σε περιττές χρονοκαθυστερήσεις. Ενώ στα σημεία που ήταν δυνατόν οι αλγόριθμοι είχαν παρόμοια διατύπωση έτσι ώστε η σύγκριση να μπορεί να είναι πιο εύστοχη.

Η ακρίβεια της κάθε πρόβλεψης για όλους τους αλγόριθμους υπολογίστηκε σύμφωνα με το μοντέλο της online εκπαίδευσης που περιγράψαμε στην ενότητα 3.3. Συγκεκριμένα, να θυμίσουμε ότι για μία διαδικασία εκπαίδευσης στην οποία τα δεδομένα λαμβάνονται σειριακά στην μορφή $S = (s_1, s_2, \dots, s_k, \dots, s_K)$ με $s_k = (\mathbf{x}(k), y(k))$ και k να είναι η τρέχουσα χρονική στιγμή. Οι online διαδικασίες εκπαίδευσης παράγουν το μοντέλο h_k από το μοντέλο h_{k-1} αφού πρώτα έχει υπολογίσει την ακρίβεια $E(S)$ για το δεδομένο k από το μοντέλο h_{k-1} . Ο υπολογισμός αυτός γίνεται σύμφωνα με την σχέση 3.3 της ενότητας 3.3 την οποία επαναδιατυπώνουμε παρακάτω

$$E(S) = \frac{1}{K} \sum_{k=1}^K (1 - L(h_{k-1}(\mathbf{x}(k)), y(k))) \quad (5.1)$$

Η συνάρτηση της απολαβής $L(\bullet)$ δίνει την τιμή 0 όταν η πρόβλεψη $h_{k-1}(\mathbf{x}(k)) = \hat{y}(k)$ είναι ίση με την πραγματική τιμή $y(k)$ και σε κάθε άλλη περίπτωση δίνει την τιμή 1.

Για την εκπαίδευση των δικτύων χρησιμοποιήθηκε ένα πλήθος από δεδομένα εκπαίδευσης. Κάποια από τα δεδομένα εκπαίδευσης υλοποιήθηκαν με τεχνητό τρόπο από το λογισμικό πρόγραμμα MOA [38], ενώ άλλα αποτελούν προβλήματα του πραγματικού κόσμου από τον επίσημο ιστότοπο του λογισμικού MOA ή και από την βάση δεδομένων UCI [39]. Στην συνέχεια αναγράφεται μία περιληπτική περιγραφή στα προβλήματα, τα οποία αντιπροσωπεύει κάθε δεδομένο εκπαίδευσης και γενικότερες πληροφορίες για αυτά.

- **Electricity** : Πρόκειται για ένα ιδιαίτερα γνωστό σύνολο δεδομένων που αναλύθηκε πρώτα

από τον M. Haggies [40]. Το συγκεκριμένο σύνολο δημιουργήθηκε από δεδομένα της αγοράς ηλεκτρικής ενέργειας της Νέας Νότιας Ουαλίας στην Αυστραλία την περίοδο 1996-1998. Οι τιμές της ηλεκτρικής ενέργειας της συγκεκριμένης πολιτείας μεταβάλλονταν κάθε πέντε λεπτά και καθορίζονταν από τις τρέχουσες απαιτήσεις της αγοράς. Οι κλάσεις του συνόλου περιγράφουν την μεταβολή της τιμής της ηλεκτρικής ενέργειας σχετικά με τις περασμένες 24 ώρες και χαρακτηρίζονται από τις κατηγορίες: πτώση ή αύξηση.

- **Hyperplane** : Αυτό το σύνολο δεδομένων έχει δημιουργηθεί τεχνητά από το λογισμικό MOA. Πρόκειται για την υλοποίηση ενός προβλήματος για την πρόβλεψη της θέσης ενός συνεχούς μεταβαλλόμενου υπερεπίπεδου σε ένα πολυδιάστατο επίπεδο d διαστάσεων. Τα επίπεδα χαρακτηρίζονται με την κλάση "θετικά" ή "αρνητικά" βάσει των συντεταγμένων τους. Η ονοματολογία Faster και Slow αναφέρεται σε δύο σύνολα δεδομένων στα οποία στο ένα επίπεδο μεταβάλλεται ταχύτερα από το άλλο [38], [41].
- **RBF** : Πρόκειται για δυο σύνολα δεδομένων στα οποία δημιουργούνται τυχαία κέντρα Γκαουσιανών RBF με τυχαία βάρη, πλάτη, και κλάση. Οι RBF που δημιουργούνται μεταβάλλονται με σταθερό ρυθμό. Τα σύνολα δεδομένων RBF_GradualRecurrin και RBF_Blips είναι δύο σύνολα με διαφορετικό είδος μεταβολής όπως περιγράψαμε αναλυτικά στην ενότητα 3.4. Το ένα σύνολο δεδομένων μεταβάλλεται με ρυθμό Gradual, ενώ το άλλο με Blips. Τα σύνολα δημιουργήθηκαν τεχνητά με την βοήθεια του λογισμικού MOA [38].
- **SEA** : Πρόκειται για ένα τεχνητό πρόβλημα που αποτελείται από δύο κλάσεις. Για την υλοποίηση του προβλήματος δημιουργείται ένα τρισδιάστατο επίπεδο στο οποίο τοποθετούνται τυχαία σημεία σε συγκεκριμένο διάστημα. Από τις τρεις μεταβλητές που ορίζουν το επίπεδο μόνο οι δύο σχετίζονται μεταξύ τους. Στην συνέχεια τα δεδομένα χωρίζονται σε τέσσερα μέρη και το κάθε σημείο αντιστοιχίζεται στην κλάση 1 αν ισχύει $f_1 + f_2 \leq \theta$, διαφορετικά στην 2 με f_1 και f_2 να είναι οι πρώτες δύο μεταβλητές του επιπέδου και θ να είναι ένα κατώφλι που χωρίζει τις δύο κλάσεις, το οποίο μεταβάλλεται για καθένα από τα τέσσερα μέρη δεδομένων. Τα σύνολα δεδομένων SEASudden και SEASudden_Faster μεταβάλλονται με μεταβολή τύπου sudden διαφορετικής δυναμικής (βλέπε ενότητα 3.4) [42].
- **PokerIsn**: Αυτό το σύνολο δεδομένων αποτελείται από 10 μεταβλητές εισόδου, 2 μεταβλητές για κάθε κάρτα που μπορεί να έχει ένας παίκτης από μία τράπουλα 52 φύλλων σε μία παρτίδα poker. Οι κλάσεις του προβλήματος χωρίζονται σε 10 κατηγορίες οι οποίες περιγράφουν τον πιο πιθανό συνδυασμό του κάθε χεριού με την μεγαλύτερη αξία. Συγκεκριμένα, οι κλάσεις που αντιστοιχούν στα πιθανά χέρια είναι: 0) Nothing in hand, 1) One pair, 2) Two pairs, 3) Three of a kind, 4) Straight, 5) Flush, 6) Full house, 7) Four of a kind, 8) Straight flush και 9) Royal flush [39].
- **Weather**: Στο συγκεκριμένο σύνολο δεδομένων η κάθε καταγραφή περιλαμβάνει 8 διαφορετικές μετρήσεις, οι οποίες χρειάζονται για την πρόγνωση του καιρού. Πιο αναλυτικά, κάθε καταγραφή αποτελείται από καιρικές συνθήκες όπως η θερμοκρασία, το σημείο υγροποίησης, η πίεση από την στάθμη της θάλασσας, η διαύγεια καιρού, η παρατεταμένη μέγιστη ταχύτητα ανέμου, η μέση ταχύτητα ανέμου, η ελάχιστη και μέγιστη θερμοκρασία, αλλά και από την κλάση που αντιστοιχεί στην καταγραφή και προσδιορίζει την κατάσταση του καιρού, βροχή ή όχι. Το συγκεκριμένο σύνολο δεδομένων δημιουργήθηκε από 7000 μετεωρολογικούς σταθμούς σε όλο τον κόσμο στο χρονικό διάστημα 1949-1999 [43], [44].

Στον πίνακα 1 παρουσιάζονται συνοπτικά τα χαρακτηριστικά για όλα τα σύνολα δεδομένων που χρησιμοποιήθηκαν στην πειραματική διαδικασία.

Για κάθε αλγόριθμο που υλοποιήθηκε έγινε μια ρύθμιση και προσαρμογή των μετα-παραμέτρων του εκάστοτε αλγόριθμου για κάθε σύνολο δεδομένων. Η διαδικασία της ρύθμισης

Σύνολο δεδομένων	Πλήθος δεδομένων	Μεταβλητές	Κλάσεις	Τύπος μεταβολής
Hyperplane_Faster	100,000	10	2	gradual
Hyperplane_Slow	100,000	10	2	gradual
RBF_Blips	100,000	10	2	blips
RBF_GradualRecurring	100,000	20	4	gradual recurring
SEASudden_Faster	100,000	3	2	abrupt
SEASudden	100,000	3	2	abrupt
interRBF	201,000	20	15	unknown
elecNormNew	45,312	9	2	unknown
pokerlsn	829,201	10	10	unknown
weather	18,159	8	2	unknown

Πίνακας 1. Χαρακτηριστικά δεδομένων.

των παραμέτρων αποτελεί μία ιδιαίτερα χρονοβόρα διαδικασία, η οποία γίνεται όλο και πιο δύσκολη για αλγόριθμους που είναι σχεδιασμένοι να χρησιμοποιούν πολλές μετα-παραμέτρους. Η ύπαρξη, όμως, επιπλέον μετα-παραμέτρων αυξάνει την προσαρμοστικότητα του αλγόριθμου στα προβλήματα που καλείται να αντιμετωπίσει. Παρόλα αυτά ένας αλγόριθμος με πληθώρα από μετα-παραμέτρους μειονεκτεί δεδομένου ότι η ρύθμιση τους αποτελεί ένα πρόβλημα από μόνο του. Είναι σημαντικό να σημειωθεί πως η αυστηρή ρύθμιση μερικών μετα-παραμέτρων, όπως για παράδειγμα ο συντελεστής γ της οικογένειας RAN, δεν είναι εντελώς απαραίτητη για την λειτουργικότητα του αλγόριθμου. Αυτό σημαίνει ότι χρησιμοποιώντας μία ίδια τιμή για όλα τα δεδομένα εκπαίδευσης δεν θα οδηγήσει σε καταστροφικά αποτελέσματα του αλγόριθμου. Σε αντίθεση η μετα-παραμέτρος ε_{min} της ίδιας οικογένειας έχει καθοριστικό ρόλο για την σωστή απόδοση του αλγόριθμου. Μία λάθος ρύθμιση της συγκεκριμένης παραμέτρου μπορεί να οδηγήσει το δίκτυο σε απόλυτη αποτυχία. Η σωστή προσαρμογή για το κάθε σύνολο δεδομένων είναι εντελώς απαραίτητη. Με βάση τα παραπάνω σχεδιάστηκε ο πίνακας 2 στον οποίο αναφέρονται αναλυτικά όλα τα βάρη που χρησιμοποιήθηκαν για τις παραμέτρους στις οποίες έγινε αυστηρή προσαρμογή σε κάθε σύνολο δεδομένων. Στον πίνακα μπορούμε να δούμε όλες τις τιμές για όλους τους αλγόριθμους που υλοποιήθηκαν για όλα τα σύνολα δεδομένων. Για την πλήρη εικόνα της διαδικασίας εκτέλεσης όλων των αλγόριθμων, αναφερόμαστε στην συνέχεια συνοπτικά για τις αριθμήσεις που δόθηκαν σε κάθε αλγόριθμο και στις μετα-παραμέτρους που είναι κοινές για όλα τα σύνολα δεδομένων.

- **FUZZY-LS-RLS**: Ο Αλγόριθμος LS-RLS περιλαμβάνει την παράμετρο N_d για την διαγραφή των ανενεργών κέντρων η οποία είναι ρυθμισμένη να έχει την τιμή 2000 για όλα τα δεδομένα εκπαίδευσης. Το πλήθος των ασαφών συνόλων ανά διάσταση c_i έχει ρυθμιστεί να είναι κοινό για όλες τις διαστάσεις, όποτε ισχύει ότι $c_1 = c_2 = \dots = c_N = c$ και οι τιμές του c δίνονται αναλυτικά στον πίνακα 2.
- **FM-RLS**: Όπως και στον LS-RLS, έτσι και στην προτεινόμενη διαδικασία ισχύει ότι $c_1 = c_2 = \dots = c_N = c$. Όλες οι μετα-παραμέτροι περιλαμβάνονται στον πίνακα 2.
- **RAN**: Οι παράμετροι ε_{max} και γ λαμβάνουν τιμές 2 και 0.99 αντίστοιχα για όλα τα σύνολα δεδομένων. Οι τιμές αυτών των παραμέτρων συνδυαστικά με την τιμή της ε_{min} , που δίνεται από τον πίνακα 2, καθορίζουν τον ρυθμό ανάπτυξης νευρώνων αλλά και την τελική σταθεροποίηση των νευρώνων στο δίκτυο. Επιπρόσθετα, οι παράμετροι ε_{min} , κ και η λαμβάνουν τις τιμές 0.1, 0.4 και 0.01 αντίστοιχα για όλα τα σύνολα. Η τιμή ε_{min} υποδεικνύει το επιτρεπόμενο σφάλμα στο δίκτυο, το κ είναι ένας συντελεστής που ρυθμίζει την επικάλυψη των ακτινικών συναρτήσεων βάσης, ενώ η παράμετρος η ρυθμίζει τον διορθωτικό παράγοντα στην διαδικασία αναβάθμισης LMS.

- **RAN-EKF**: Παρομοίως οι παράμετροι έχουν τις τιμές: $\varepsilon_{max} = 4$, $\gamma = 0.99$, $\varepsilon_{min} = 0.1$ και $\kappa = 0.4$ για όλα τα σύνολα. Οι νέες παράμετροι που εισάγει ο RAN-EKF για όλα τα σύνολα παίρνουν τιμές $Q_0 = 0.000005$, $R_n = 1$ και $P_0 = 1$ όπου Q_0 είναι ένας παράγοντας τυχαιότητας της διαδικασίας EKF, R_n είναι η διακύμανση του θορύβου μέτρησης και P_0 είναι παράμετρος της εκτιμώμενης αβεβαιότητας.
- **M-RAN**: Οι παράμετροι που χρησιμοποιούνται και στον RAN-EKF έχουν τιμές: $\varepsilon_{max} = 4$, $\gamma = 0.99$, $\varepsilon_{min} = 0.1$, $\kappa = 0.4$, $Q_0 = 0.000005$, $R_n = 1$ και $P_0 = 1$ ίδιες, δηλαδή, με τον RAN-EKF. Η διαδικασία M-RAN εισάγει τις παραμέτρους δ και M για την διαδικασία διαγραφής νευρώνων με τιμές 0.08 και 2000 για όλα τα σύνολα. Η παράμετρος ε'_{min} για το νέο κριτήριο απόφασης προσθήκης νευρώνων λαμβάνει την τιμή 0.02.
- **GAP**: Ο αλγόριθμος GAP χρησιμοποιεί τις περισσότερες παραμέτρους του M-RAN εκτός από τις ε'_{min} , δ και M . Επίσης, εισάγει την παράμετρο $S(X)$ η οποία είναι ο όγκος που ορίζεται από το πεδίο τιμών του σχηματιζόμενου χώρου που από τις μεταβλητές εισόδου. Για τα σύνολα δεδομένων τα οποία προκύπτουν τεχνητά θεωρείται ότι έχουν ομοιόμορφη κατανομή στο πεδίο τιμών X . Κρατώντας αυτό ως δεδομένο και γνωρίζοντας ότι τα δεδομένα κανονικοποιούνται στο διάστημα $[-1, 1]$, μπορούμε να θεωρήσουμε το $S(X) = 2^N$ όπου N είναι το πλήθος των μεταβλητών εισόδου του εκάστοτε συνόλου δεδομένων. Δυστυχώς για τα δεδομένα που προκύπτουν από τον πραγματικό κόσμο δεν ισχύει η ομοιόμορφη κατανομή. Για αυτά τα δεδομένα οι τιμές του $S(X)$ επιλέχθηκαν κατόπιν διαδικασίας ρύθμισης της παραμέτρου. Ενώ για τις υπόλοιπες παραμέτρους του δικτύου δόθηκαν οι τιμές $\varepsilon_{max} = 4$, $\gamma = 0.99$, $\varepsilon_{min} = 0.1$, $\kappa = 0.4$ $Q_0 = 0.000005$, $R_n = 1$ και $P_0 = 1$

Σύνολο δεδομένων	FUZZY-LS-RLS			FM-RLS			RAN	RAN-EKF	M-RAN	GAP	
Μεταπάρμετροι	N_w	c	λ	N_d	c	λ	ε_{min}	ε_{min}	ε_{min}	ε_{min}	$S(X)$
HyperplaneFaste	2000	4	0.9990	1000	4	0.9990	2.2	3.1	2.5	2.5	$S(X)$
HyperplaneSlow	2000	4	0.9998	1000	4	0.9998	2.4	3.2	2.8	2.8	$S(X)$
RBFBlips	2000	12	0.9977	2000	12	0.9977	0.8	2.0	1.9	1.9	$S(X)$
RBFGradualRec	2000	13	0.9978	2000	13	0.9978	0.8	1.7	1.8	1.8	$S(X)$
SEASuddenFaster	1000	10	0.9991	2000	10	0.9991	0.5	2.3	0.9	0.9	$S(X)$
SEASudden	1000	10	0.9995	2000	10	0.9995	0.5	2.0	1.0	1.0	$S(X)$
elecNormNew	1000	20	0.9956	2000	20	0.9956	0.6	2.6	0.5	0.5	0.1
interRBF	2000	18	0.9925	1000	18	0.9925	0.5	2.1	1.6	1.6	10^5
pokerlsn	2000	9	0.9934	10000	9	0.9977	0.9	2.6	1.6	1.6	10^4
weather	2000	12	0.9982	10000	12	0.9982	0.5	2.5	1.1	1.1	10^{-3}

Πίνακας 2. Προσαρμοσμένοι μετα-παράμετροι αλγορίθμων για όλα τα σύνολα δεδομένων.

5.2 Αποτελέσματα και συζήτηση

Προκειμένου να μπορέσουμε να κάνουμε μία πρώτη σύγκριση των αλγορίθμων που υλοποιήθηκαν στα πλαίσια της πειραματικής διαδικασίας, δημιουργήσαμε τον πίνακα 3. Ο πίνακας αυτός περιλαμβάνει την ακρίβεια, τον χρόνο εκπαίδευσης, το πλήθος των κέντρων μαζί με το πλήθος των μεταβλητών εκπαίδευσης, καθώς και μία διαδικασία αξιολόγησης με βαθμούς για κάθε διαδικασία και για όλα τα σύνολα δεδομένων. Είναι σημαντικό να σημειωθεί ότι στους αλγόριθμους εκπαίδευσης FUZZY-LS-RLS και FM-RLS η διαδικασία RLS καλύτερα να αναβαθμίσει για κάθε δεδομένο εκπαίδευσης $x(k)$ που λαμβάνετε μόνο μία παράμετρο για κάθε δεσμευμένο κέντρο του δικτύου και για κάθε έξοδο συν την Bias σύναψη. Επομένως, το πλήθος των παραμέτρων που πρέπει να αναβαθμιστούν από τον RLS ανά κέντρο δίνεται από την σχέση $(N+1)*D$ όπου N το πλήθος των δεσμευμένων νευρώνων και D το πλήθος των εξόδων. Αντιθέτως, στους αλγόριθμους

της οικογένειας RAN οι συντεταγμένες τον κέντρο εισάγονται στην διαδικασία LMS ή την EKF. Δεδομένου ότι η συνάρτηση βάσης η οποία χρησιμοποιείται είναι Gaussian, τελικά θα χρειαστεί να αναβαθμιστούν οι παράμετροι σ του πλάτους οι παράμετροι u_1, u_2, \dots, u_L των συντεταγμένων των κέντρων και μία επιπλέον για την Bias σύναψη και άρα τελικά το πλήθος των παραμέτρων που πρέπει να αναβαθμίζονται με την άφιξη κάθε νέου δεδομένου $x(k)$ από την διαδικασία LMS ή την EKF ανά κέντρο δίνεται από την σχέση $(L*(N+1)) + 1$ όπου L το πλήθος των μεταβλητών εισόδου. Μία ακόμα πληροφορία που φαίνεται στον πίνακα 3 είναι το πλήθος των παραμέτρων που αναβαθμίζονται με την άφιξη κάθε νέου δεδομένου από τις διαδικασίες LS,RLS,LMS και EKF αντίστοιχα για τον κάθε αλγόριθμο. Στον πίνακα μπορούμε να δούμε τις βέλτιστες τιμές για κάθε διαδικασία με υπογράμμιση.

Σύνολο δεδομένων	Χαρακτηριστικό	FUZZY-LS-RLS	FM-RLS	RAN	RAN-EKF	M-RAN	GAP
HyperplaneFaste	Ακρίβεια(Βαθμός)	0.8790 (2)	0.8844 (4)	0.8579 (1)	<u>0.9067</u> (6)	0.9009 (5)	0.8798 (3)
	Χρόνος(ms)(Βαθμός)	28.95 (3)	18.55 (5)	<u>15.70</u> (6)	20.72 (4)	36.57 (2)	53.58 (1)
	Κέντρα/Παράμετροι	103/208	77/156	73/877	<u>4/49</u>	14/169	24/289
	Βαθμολογία	5	9	7	10	7	4
HyperplaneSlow	Ακρίβεια(Βαθμός)	0.8908 (2)	0.9036 (4)	0.8736 (1)	<u>0.9193</u> (6)	0.9179 (5)	0.9003 (3)
	Χρόνος(ms)(Βαθμός)	28.82 (1)	18.48 (4)	<u>14.54</u> (6)	20.73 (3)	16.91 (5)	26.59 (2)
	Κέντρα/Παράμετροι	103/208	77/156	45/541	4/49	<u>2/25</u>	12/145
	Βαθμολογία	3	8	7	9	10	5
RBFBlips	Ακρίβεια(Βαθμός)	<u>0.9879</u> (6)	0.9647 (5)	0.9166 (4)	0.9134 (3)	0.8905 (2)	0.5079 (1)
	Χρόνος(ms)(Βαθμός)	810.17 (2)	<u>50.46</u> (6)	127.22 (5)	37315.20 (1)	464.20 (3)	195.10 (4)
	Κέντρα/Παράμετροι	189/760	189/760	4759/104699	178/3917	<u>21/463</u>	17/375
	Βαθμολογία	8	11	9	4	5	5
RBFGradualRec	Ακρίβεια(Βαθμός)	<u>0.9836</u> (6)	0.9516 (5)	0.9076 (4)	0.9037 (3)	0.8929 (2)	0.4537 (1)
	Χρόνος(ms)(Βαθμός)	1178.90 (2)	<u>66.95</u> (6)	120.36 (5)	19466.67 (1)	912.94 (3)	226.80 (4)
	Κέντρα/Παράμετροι	282/1132	282/1132	5552/122145	159/3499	30/661	<u>23/507</u>
	Βαθμολογία	8	11	9	4	5	5
SEASuddenFaster	Ακρίβεια(Βαθμός)	0.8730 (4)	0.8745 (5)	0.8718 (3)	0.8703 (2)	<u>0.8755</u> (6)	0.8658 (1)
	Χρόνος(ms)(Βαθμός)	30.09 (1)	25.60 (3)	15.65 (5)	17.87 (4)	29.26 (2)	<u>15.08</u> (6)
	Κέντρα/Παράμετροι	132/266	132/266	78/391	<u>2/11</u>	28/141	10/51
	Βαθμολογία	5	8	8	6	8	7
SEASudden	Ακρίβεια(Βαθμός)	0.8782 (5)	<u>0.8806</u> (6)	0.8772 (4)	0.8699 (3)	<u>0.8806</u> (6)	0.8527 (2)
	Χρόνος(ms)(Βαθμός)	30.42 (1)	25.53 (2)	<u>15.49</u> (6)	17.74 (4)	21.44 (3)	16.58 (5)
	Κέντρα/Παράμετροι	130/262	130/262	79/396	<u>2/11</u>	19/96	14/71
	Βαθμολογία	6	8	10	7	9	7
elecNormNew	Ακρίβεια(Βαθμός)	0.8394 (4)	<u>0.8691</u> (6)	0.7604 (2)	0.7541 (1)	0.8608 (5)	0.7922 (3)
	Χρόνος(ms)(Βαθμός)	135.79 (3)	22.01 (4)	7.00 (5)	<u>5.03</u> (6)	637.68 (1)	532.77 (2)
	Κέντρα/Παράμετροι	244/490	244/490	410/4101	<u>3/31</u>	71/711	107/1071
	Βαθμολογία	7	10	7	7	6	5
interRBF	Ακρίβεια(Βαθμός)	0.7659 (2)	<u>0.9654</u> (6)	0.8350 (5)	0.8223 (4)	0.7922 (3)	0.5619 (1)
	Χρόνος(ms)(Βαθμός)	1689.05 (3)	<u>126.50</u> (6)	317.48 (5)	34386.27 (1)	1645.88 (4)	5503.35 (2)
	Κέντρα/Παράμετροι	305/4590	194/2925	4864/107009	140/3081	<u>22/485</u>	61/1343
	Βαθμολογία	5	12	10	5	7	3
pokerlsn	Ακρίβεια(Βαθμός)	0.8063 (2)	0.8506 (5)	0.8305 (3)	<u>0.8951</u> (6)	0.8456 (4)	0.7203 (1)
	Χρόνος(ms)(Βαθμός)	3812.72 (2)	<u>2256.23</u> (6)	2283.61 (5)	3920.46 (1)	2733.37 (4)	2792.19 (3)
	Κέντρα, Μεταβλητές	224/2250	400/4010	6293/75517	36/433	<u>25/301</u>	46/553
	Βαθμολογία	4	11	8	7	8	4
weather	Ακρίβεια (Βαθμός)	<u>0.8074</u> (6)	0.8004 (4)	0.7865 (2)	0.7837 (1)	0.8024 (5)	0.7970 (3)
	Χρόνος(ms)(Βαθμός)	5.59 (1)	2.36 (4)	1.75 (5)	<u>1.31</u> (6)	3.80 (2)	3.70 (3)
	Κέντρα, Μεταβλητές	79/160	122/246	129/1291	<u>3/31</u>	16/161	16/161
	Βαθμολογία	7	8	7	7	7	6
-	Συνολική βαθμολογία ακρίβειας	39	50	29	35	43	19
-	Συνολική βαθμολογία χρόνου	19	46	53	31	29	32
-	Συνολική βαθμολογία	58	96	82	66	72	51

Πίνακας 3. Πίνακας αποτελεσμάτων αλγόριθμων εκπαίδευσης.

Για να μπορέσουμε να σχηματίσουμε μία ξεκάθαρη εικόνα για την συνολική απόδοση των διαδικασιών εκπαίδευσης προσθέσαμε μία απλή μέθοδο βαθμολόγησης στον πίνακα 3. Σύμφωνα με την μέθοδο αυτή η κάθε διαδικασία λαμβάνει από 6 έως 1 βαθμό με το μεγαλύτερο να τον λαμβάνει ο αλγόριθμος με τα καλύτερα αποτελέσματα, ενώ τον μικρότερο αυτός με τα χειρότερα. Στον πίνακα 3 βαθμολογήθηκαν όλοι οι αλγόριθμοι εκπαίδευσης για την ακρίβεια τους, αλλά και την ταχύτητα της διαδικασίας εκπαίδευσης. Στην τελευταία γραμμή κάθε διαδικασίας αθροίζονται οι δύο βαθμοί και, έτσι, ο αλγόριθμος λαμβάνει μία συνολική βαθμολογία για την συνολική του απόδοση. Τέλος, στην τελευταία γραμμή του πίνακα γίνεται ένα άθροισμα από τις συνολικές

βαθμολογίες για όλα τα δεδομένα εκπαίδευσης κάθε διαδικασίας. Βάσει των αποτελεσμάτων του πίνακα 3 ξεκινάμε συγκρίνοντας την αποδοτικότητα του αλγόριθμου FUZZY-LS-RLS με τον FM-RLS. Ο αλγόριθμος FUZZY-LS-RLS είναι μία υβριδική διαδικασία που συνδυάζει την τεχνική RLS για την σταδιακή αναβάθμιση των παραμέτρων. Στην περίπτωση προσθήκης ή διαγραφής κάποιου νευρώνα επανεκπαιδεύει μαζικά ολόκληρο το δίκτυο από την αρχή με την διαδικασία LS, χρησιμοποιώντας ένα κυλιόμενο παράθυρο. Δεδομένο αυτού θα περιμέναμε να επιτυγχάνει μεγαλύτερες ακρίβειες από διαδικασίες εξ ολοκλήρου σταδιακής εκπαίδευσης. Παρόλα αυτά παρατηρώντας τον πίνακα 3 μπορούμε να διακρίνουμε ότι η διαδικασία FM-RLS μπόρεσε να ξεπεράσει σε ακρίβεια τον FUZZY-LS-RLS σε 7 από τα 10 σύνολα δεδομένων. Είναι σημαντικό να σημειωθεί ότι μία αύξηση του κυλιόμενου παραθύρου, το οποίο χρησιμοποιείται στην διαδικασία FUZZY-LS-RLS, έχει μεγάλο υπολογιστικό κόστος αναγκάζοντας την διαδικασία να περιοριστεί σε μικρότερα πλαίσια. Με αποτέλεσμα κατά την επανεκπαίδευση του δικτύου με την διαδικασία LS χάνεται ένα μέρος σημαντικής πληροφορίας που πιθανόν υπήρχε από προηγούμενα δεδομένα. Αντιθέτως η διαδικασία FM-RLS χρησιμοποιεί το παράθυρο μόνο για τη διαδικασία διαγραφής νευρώνα χωρίς να το χρησιμοποιεί για την διαδικασία αναβάθμισης των συνάψεων. Στην διαδικασία FM-RLS μία αύξηση του παραθύρου κάνει τον αλγόριθμο λιγότερο αυστηρό με την διαγραφή των κέντρων. Αυτό μπορεί να σημαίνει ότι νευρώνες που δεν προσφέρουν στο δίκτυο παραμένουν στη διαδικασία εκπαίδευσης επιβαρύνοντας υπολογιστικά. Παρόλο αυτό μία αύξηση του κυλιόμενου παραθύρου δεν συνεπάγεται άμεσα αύξηση υπολογιστικού κόστους. Από τον πίνακα μπορούμε να διακρίνουμε ότι η διαδικασία FM-RLS μπόρεσε να είναι ταχύτερη από την FUZZY-LS-RLS σε όλα τα σύνολα δεδομένων εκπαίδευσης. Σύμφωνα με τα αποτελέσματα του πίνακα μπορούμε να ισχυριστούμε ότι η διαδικασία FM-RLS έχει καλύτερη απόδοση, γεγονός που επιβεβαιώνεται και από την συνολική βαθμολογία συμπληρώνοντας 96 βαθμούς έναντι της διαδικασίας FUZZY-LS-RLS με μόλις 58 βαθμούς.

Στην συνέχεια θα αγνοήσουμε τον αλγόριθμο FUZZY-LS-RLS και θα συνεχίσουμε με την σύγκριση της διαδικασίας FM-RLS με τους υπόλοιπους αλγόριθμους της οικογένειας RAN. Κοιτάζοντας τον πίνακα 3 διακρίνουμε ότι η διαδικασία FM-RLS μπόρεσε να πετύχει καλύτερη ακρίβεια σε πέντε 5 από τα 10 σύνολα δεδομένα και μεγαλύτερη ταχύτητα σε 4 από τα 10 σύνολα. Με μία πρώτη ματιά διαπιστώνουμε ότι ο αλγόριθμος FM-RLS αποτελεί μία ανταγωνιστική διαδικασία έναντι των υπολοίπων, κάτι που επιβεβαιώνεται και από το σύνολο της βαθμολογίας που μπόρεσε να συμπληρώσει. Ένας αλγόριθμος σταδιακής εκπαίδευσης μας ενδιαφέρει να μπορεί να προσφέρει καλές ακρίβειες αλλά και μικρούς χρόνους εκτέλεσης. Άρα, για έναν αλγόριθμο με υψηλή ακρίβεια, αλλά με μεγάλες απαιτήσεις χρόνου για μία διαδικασία σταδιακής εκπαίδευσης, μπορεί να θεωρηθεί σημαντικό μειονέκτημα. Παρατηρώντας τον πίνακα 3 βλέπουμε παραδείγματα στα οποία άλλες διαδικασίες έχουν καλύτερες ακρίβειες όμως με μεγάλο κόστος χρόνου. Παραδείγματος χάρι ο αλγόριθμος RAN-EKF μπόρεσε να πετύχει ποσοστό επιτυχίας 0.8951 στο σύνολο δεδομένων Poker1sn σε αντίθεση με τον FM-RLS που κατάφερε 0.8506. Παρόλα αυτά, όμως, αυτή η διαφορά κόστισε στην διαδικασία RAN-EKF: 1,664.23 ms περισσότερο χρόνο (3920.46ms - 2256.23ms = 1,664.23 ms), καθιστώντας την πιο αργή διαδικασία από όλες τις άλλες. Ένα άλλο χαρακτηριστικό παράδειγμα για την ακρίβεια σε σχέση με τον χρόνο αφορά στο σύνολο δεδομένων RBFBlips. Στο οποίο η διαδικασία RAN-EKF χρειάστηκε 37315.20ms σε αντίθεση με την FM-RLS που μπόρεσε να πετύχει καλύτερη ακρίβεια σε μόλις 50.46ms. Μία πιο στοχευμένη σύγκριση που αξίζει να γίνει είναι με τον αλγόριθμο RAN, ο οποίος κατάφερε να είναι δεύτερος σε σειρά βαθμολογίας με 82 βαθμούς. Μπορούμε να συμπεράνουμε ότι το μεγαλύτερο ποσοστό της βαθμολογίας του, το έλαβε από το γεγονός ότι είναι η πιο γρήγορη διαδικασία από όλες, συμπληρώνοντας 53 βαθμούς στην συνολική βαθμολογία χρόνου. Οι ακρίβειες, όμως, που επιτυγχάνει συνήθως είναι χειρότερες από τις άλλες διαδικασίες εκπαίδευσης καθιστώντας τον πέμπτο σε σειρά συνολικής βαθμολογίας ακρίβειας με 29 βαθμούς. Θα μπορούσε κανείς να θεωρήσει τον RAN μία πολύ καλή επιλογή για προβλήματα που έχουν υψηλές απαιτήσεις σε οικονομία υπολογιστικού κόστους. Σε αντίθεση ο αλγόριθμος FM-RLS μπόρεσε να διεκδικήσει την μεγαλύτερη βαθμολογία

ακρίβειας από όλες τις άλλες διαδικασίες εκπαίδευσης και ταυτόχρονα διατήρησε υψηλές ταχύτητες που τον θέτουν δεύτερο σε σειρά πιο γρήγορο με 46 βαθμούς στην συνολική βαθμολογία χρόνου. Μία ακόμα ενδιαφέρουσα σύγκριση είναι ο αλγόριθμος M-RAN ο οποίος διεκδίκησε την τρίτη θέση συμπληρώνοντας 72 συνολικά βαθμούς. Μολονότι, ο αλγόριθμος M-RAN φαίνεται να επιτυγχάνει καλές ακρίβειες στα περισσότερα προβλήματα, συγκρίσιμες με αυτές του FM-RLS, μπορούμε να διακρίνουμε περιπτώσεις όπως στα σύνολα δεδομένων RBFBlips και interRBF στα οποία οι χρόνοι εκτέλεσης ήταν σημαντικά μεγαλύτεροι σε σύγκριση με αυτούς του FM-RLS, χωρίς απαραίτητα η ακρίβεια να είναι σημαντικά μεγαλύτερη. Παρόλο αυτό σε κάποια προβλήματα, συγκεκριμένα στα SEASudden και HyperplaneSlow, ο M-RAN μπόρεσε να διατηρήσει τις ίδιες ή και καλύτερες ακρίβειες από τον FM-RLS με ελαφρώς καλύτερο χρόνο.

Μία ακόμα σημαντική διαπίστωση που αφορά στην ικανότητα γενίκευσης του FM-RLS προκύπτει από τα αποτελέσματα του πίνακα 3. Από τον πίνακα 3 μπορούμε να υπολογίσουμε την απόκλιση ακρίβειας για κάθε διαδικασία από την διαδικασία που πέτυχε την μέγιστη ακρίβεια σε κάθε σύνολο δεδομένων. Το ίδιο αντίστοιχα μπορούμε να κάνουμε για τους χρόνους εκτέλεσης. Με αυτό τον τρόπο προκύπτει ο πίνακας 4 ο οποίος περιλαμβάνει τη διαφορά αποδόσεις του εκάστοτε αλγόριθμου από τον αλγόριθμο με την βέλτιστη απόδοση για το κάθε σύνολο εκπαίδευσης. Από τον πί-

Σύνολο δεδομένων	Χαρακτηριστικό	FUZZY-LS-RLS	FM-RLS	RAN	RAN-EKF	M-RAN	GAP
HyperplaneFaste	Απόκλιση ακρίβειας	0.0277	0.0223	0.0488	0	0.0058	0.0269
	Απόκλιση χρόνου(ms)	13.25	2.85	0	5.02	20.87	37.88
HyperplaneSlow	Απόκλιση ακρίβειας	0.0285	0.0157	0.0457	0	0.0014	0.019
	Απόκλιση χρόνου(ms)	14.28	3.94	0	6.19	2.37	12.05
RBFBlips	Απόκλιση ακρίβειας	0	0.0232	0.0713	0.0745	0.0974	0.48
	Απόκλιση χρόνου(ms)	759.71	0	76.76	37264.74	413.74	144.64
RBFGradualRec	Απόκλιση ακρίβειας	0	0.032	0.0760	0.0799	0.0907	0.5299
	Απόκλιση χρόνου(ms)	1111.95	0	53.41	19399.72	845.99	159.85
SEASuddenFaster	Απόκλιση ακρίβειας	0.0025	0.0010	0.0037	0.0052	0	0.0097
	Απόκλιση χρόνου(ms)	15.01	10.52	0.57	2.79	14.18	0
SEASudden	Απόκλιση ακρίβειας	0.0024	0	0.0034	0.0107	0	0.0279
	Απόκλιση χρόνου(ms)	14.93	10.04	0	2.25	5.95	1.09
elecNormNew	Απόκλιση ακρίβειας	0.0297	0	0.1087	0.115	0.0083	0.0769
	Απόκλιση χρόνου(ms)	130.76	16.98	1.97	0	632.65	527.74
interRBF	Απόκλιση ακρίβειας	0.1995	0	0.1304	0.1431	0.1732	0.4035
	Απόκλιση χρόνου(ms)	1562.55	0	190.98	34259.77	1519.38	5376.85
pokerlsln	Απόκλιση ακρίβειας	0.0888	0.0445	0.0646	0	0.0495	0.1748
	Απόκλιση χρόνου(ms)	1556.49	0	27.3800	1664.23	477.14	535.96
weather	Απόκλιση ακρίβειας	0	0.0070	0.0209	0.0237	0.005	0.0104
	Απόκλιση χρόνου(ms)	4.28	1.05	0.44	0	2.49	2.39
-	Μέγιστη απόκλιση ακρίβειας	0.1995	0.0445	0.1304	0.1431	0.1732	0.5299
-	Μέγιστη Απόκλιση χρόνου	1562.55	16.98	190.98	37264.74	1519.38	5376.85

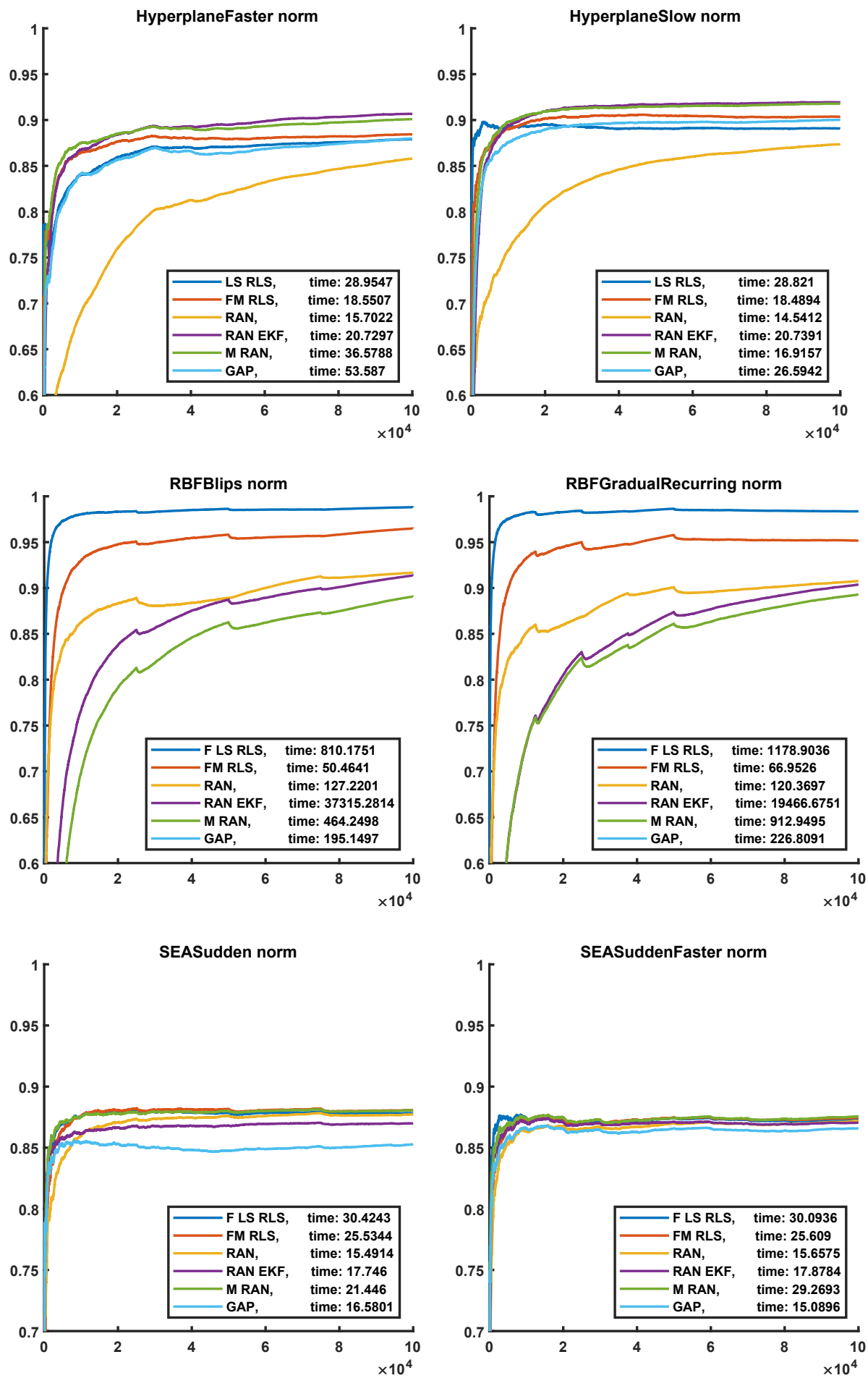
Πίνακας 4. Πίνακας αποκλίσεων από την διαδικασία με την βέλτιστη απόδοση.

νακα 4 μπορούμε να διακρίνουμε ότι η μέγιστη απόκλιση ακρίβειας του FM-RLS από την μέγιστη ακρίβεια, για κάθε διαδικασία και σύνολο δεδομένων, ήταν της τάξεως $0.8951 - 0.8506 = 0.0445$ για το σύνολο Pokerlsln έναντι του αλγόριθμου RAN-EKF (Βλέπε μέγιστη απόκλιση ακρίβειας πίνακα 4). Αντιθέτως, διαδικασίες όπως ο GAP στα σύνολα δεδομένων RBFBlips, RBFGradualRec, interRBF και οι αλγόριθμοι RAN-EKF και RAN στο elecNormNew και, επίσης, ο M-RAN στο σύνολο interRBF έχουν σημαντικές αποκλίσεις στην ακρίβεια που απέχουν από 0.1 έως και 0.5 από τις μέγιστες ακρίβειες για το κάθε σύνολο. Ομοίως εξετάζοντας τους χρόνους εκτέλεσης του FM-RLS, βλέπουμε ότι η μέγιστη απόκλιση από άλλες ταχύτερες διαδικασίες παρουσιάζεται στο σύνολο δεδομένων elecNormNew και είναι της τάξεως των $22.01ms - 5.03ms = 16.98m$ έναντι του αλγόριθμου RAN-EKF. Αντίστοιχα, η διαδικασία FUZZY-LS-RLS χρειάστηκε $1562.55ms$ για το σύνολο interRBF. Η RAN $190.98ms$ για το σύνολο interRBF. Η RAN-EKF χρειάστηκε $37264.74ms$ για το RBFBlips και 1519.38 για το σύνολο interRBF. Τέλος, ο GAP είχε $5376.85ms$ στο σύνολο interRBF. Να σημειωθεί ότι οι μεγάλες αποκλίσεις για τις άλλες διαδικασίες δεν ήταν

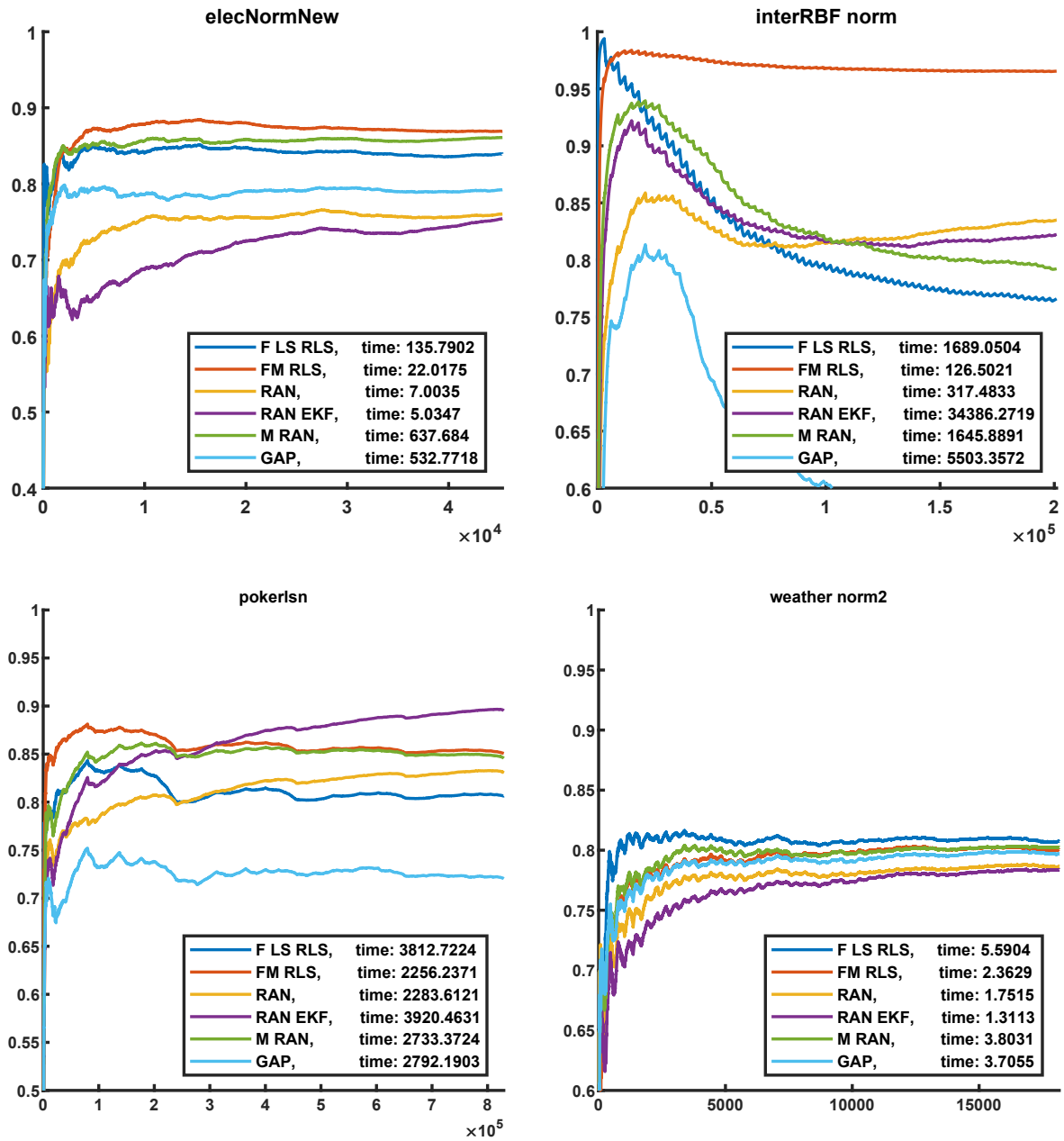
φαινόμενο που αφορά αποκλειστικά στο σύνολο interRBF, αλλά παρατηρείται και σε άλλα σύνολα δεδομένων όπως το RBFBlips, το RBFGradualRec, το ElecNormNew και το PokerIsn.

Ένα ακόμα σημαντικό σημείο που πρέπει να μελετηθεί είναι η σύγκλιση των διαδικασιών εκπαίδευσης. Για να μπορέσουμε να εξετάσουμε τις ικανότητες σύγκλισης των αλγορίθμων παρουσιάζονται στο σχήμα 26 τα γραφήματα ακρίβειας για όλες τις διαδικασίες και για όλα τα σύνολα δεδομένων εκπαίδευσης. Με μία πρώτη ματιά στα αποτελέσματα μπορούμε να διακρίνουμε ότι η διαδικασία FUZZY-LS-RLS συγκλίνει ταχύτερα σε 6 από τα 10 σύνολα σε σχέση με όλες τις άλλες διαδικασίες. Αυτό οφείλεται στο γεγονός ότι η διαδικασία κάνει χρήση ελάχιστων τετραγώνων (LS) κάθε φορά που δεσμεύει ένα νέο κέντρο, δηλαδή, γίνεται πολύ συχνά κατά την έναρξη του αλγόριθμου. Η διαδικασία (LS) αποτελεί μία διαδικασία μαζικής μάθησης η οποία παρέχει τα βέλτιστα αποτελέσματα, γεγονός που βοηθάει στην σύγκλιση της διαδικασίας. Αγνοώντας τον αλγόριθμο FUZZY-LS-RLS η διαδικασία που φαίνεται να έχει τα καλύτερα αποτελέσματα σύγκλισης είναι ο αλγόριθμος M-RAN. Αναλυτικά, ο αλγόριθμος M-RAN φαίνεται να συγκλίνει ταχύτερα σε 5 από τα 10 σύνολα το οποίο επιτυγχάνεται από την επιλεκτική διαδικασία αναπτύξης νευρώνων. Η διαδικασία ανάπτυξης και διαγραφής νευρώνων του M-RAN δίνει το πλεονέκτημα να αναπτύσσει ταχύρρυθμα νευρώνες χωρίς αυτοί να δεσμεύονται πλεοναστικά, σε αντίθεση με τις διαδικασίες RAN-EKF και RAN οι οποίες συγκλίνουν βραδύτερα. Η διαδικασία FM-RLS φαίνεται να διεκδικεί την τρίτη θέση καταφέροντας να είναι ταχύτερη σε 4 από τα 5 σύνολα δεδομένων. Τελειώνοντας, επισημαίνουμε ότι η κακή απόδοση του αλγόριθμου GAP τον καθιστά τελευταίο από όλες τις διαδικασίες. Στο σχήμα 27 παρατίθεται ο ρυθμός ανάπτυξης των νευρώνων για όλες τις διαδικασίες μάθησης και για όλα τα σύνολα δεδομένων.

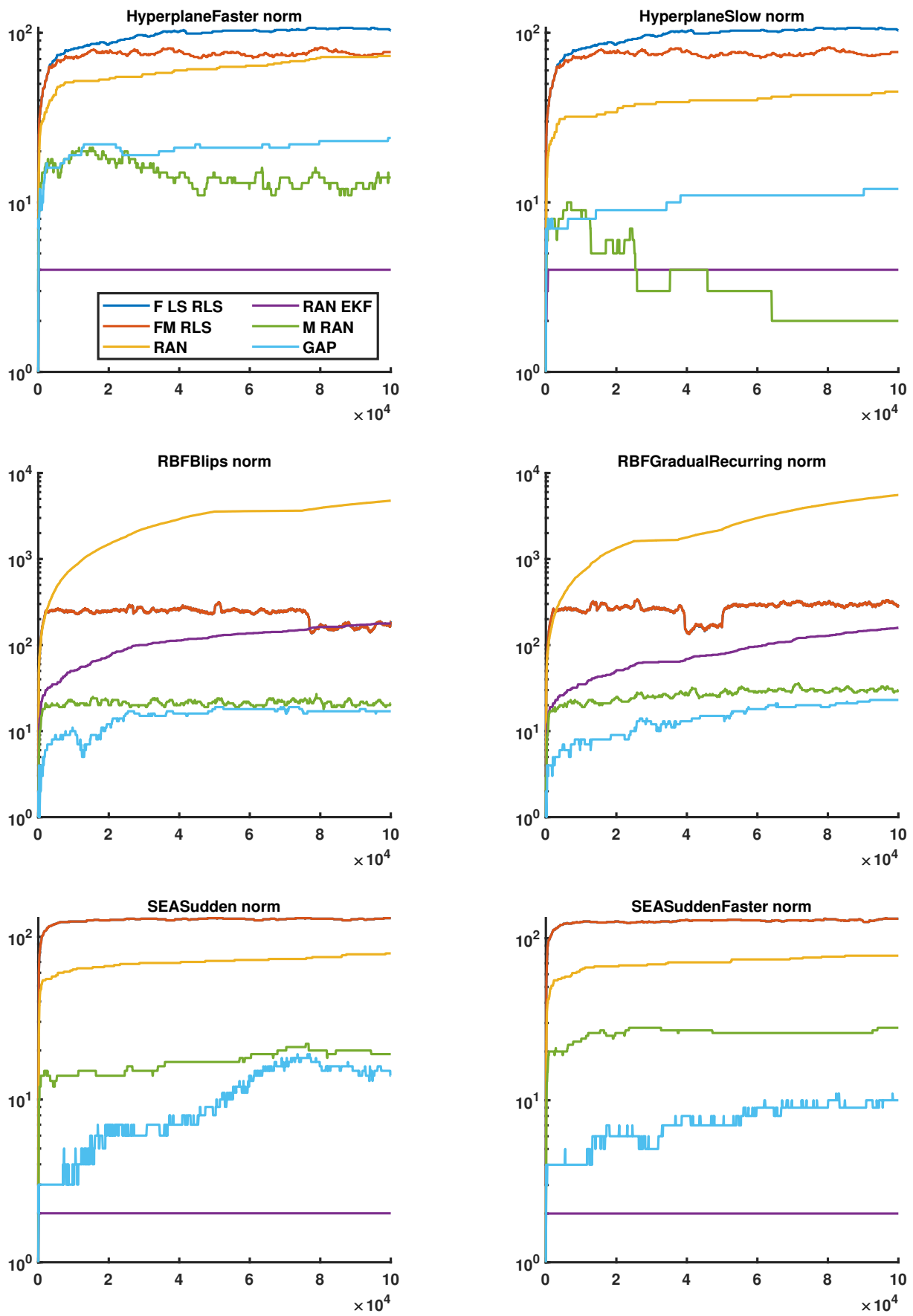
Συνοψίζοντας σύμφωνα με τα αποτελέσματα η διαδικασία FM-RLS κατάφερε να είναι η δεύτερη ταχύτερη διαδικασία με τα καλύτερα αποτελέσματα ακρίβειας σύμφωνα με τη διαδικασία βαθμολόγησης που αναπτύξαμε. Επίσης, η σταθερή απόδοση σε χρόνους και ακρίβεια σε όλα τα σύνολα δείχνει την δυναμική του σε θέματα γενικοποίησης. Ο αλγόριθμος FM-RLS βγήκε τρίτος σε ζητήματα σύγκλισης, κάτι το οποίο αφήνει περιθώρια βελτίωσης. Επιπρόσθετα μπορούμε να συμπεράνουμε ότι ο προτεινόμενος αλγόριθμος δείχνει να είναι μία δυναμική εξέλιξη των προγόνων του FUZZY-LS-RLS και RAN-EKF, αφού κατάφερε να βελτιώσει ταχύτητα και ακρίβεια ταυτόχρονα. Εν κατακλείδι, η διαδικασία FM-RLS φαίνεται να έχει καλύτερη ικανότητα γενίκευσης με πιο σταθερά αποτελέσματα, επιτυγχάνοντας καλύτερες ακρίβειες και σε λιγότερο χρόνο εκτέλεσης έναντι των διαδικασιών με τις οποίες συγκρίθηκε.



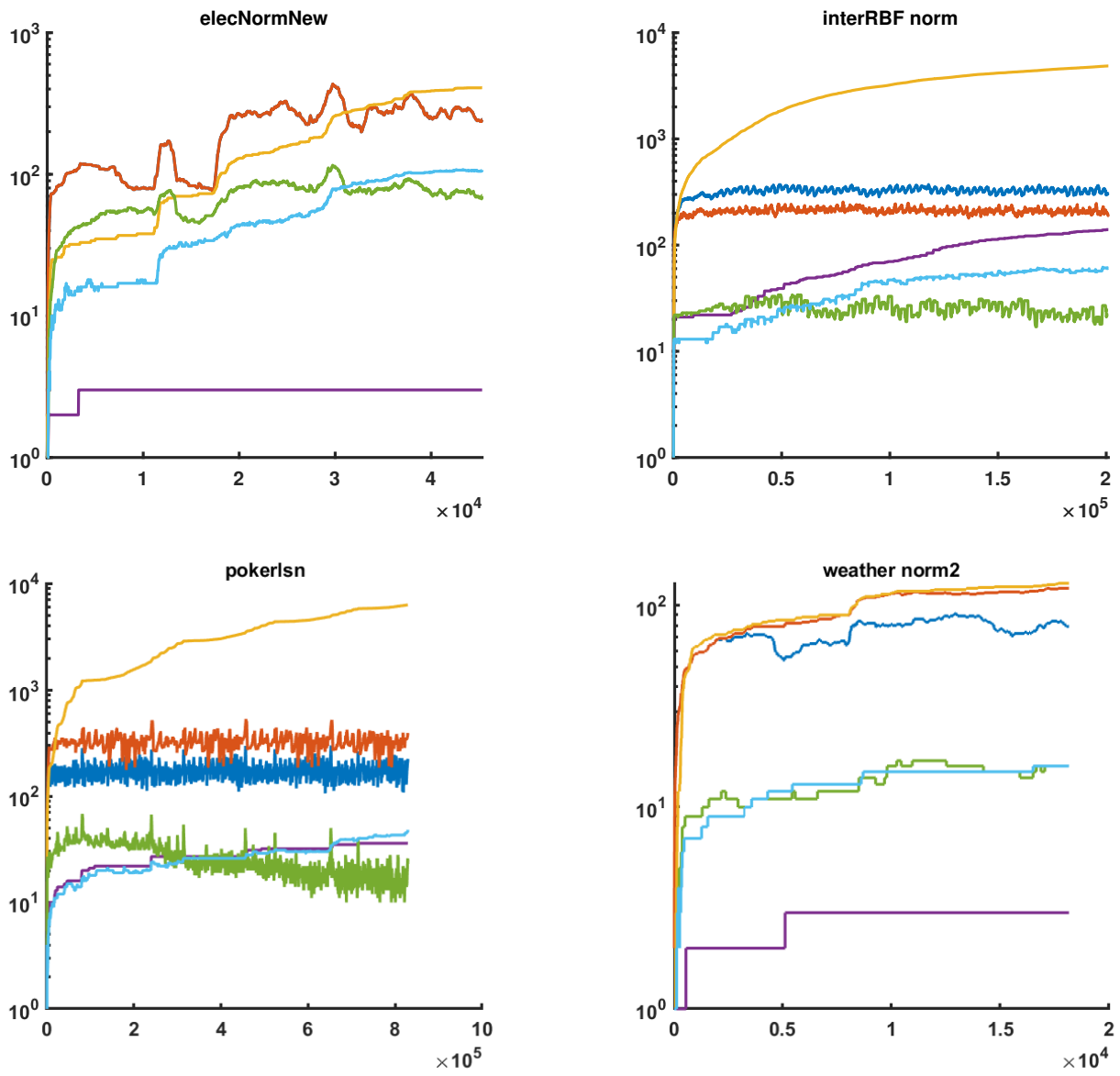
Εικόνα 26. Γραφήματα ακρίβειας



Εικόνα 27. Γραφήματα ακρίβειας



Εικόνα 28. Γραφήματα ανάπτυξης νευρώνων



Εικόνα 29. Γραφήματα ανάπτυξης νευρώνων

Κεφάλαιο 6

Συμπεράσματα

Στις μέρες μας ο όγκος των δεδομένων που παράγεται και χρησιμοποιείται σε καθημερινή βάση από σύγχρονες και ανερχόμενες τεχνολογίες συνεχώς αυξάνεται. Η διαχείριση και η επεξεργασία του όγκου αυτών των δεδομένων στα απαιτούμενα χρονικά περιθώρια συνεχώς δυσκολεύει, δημιουργώντας ένα πρόβλημα γνωστό ως πρόβλημα των μεγάλων δεδομένων (Big Data Problem). Τα τεχνητά νευρωνικά δίκτυα υλοποιώντας μοντέλα πρόβλεψης και ταξινόμησης προτύπων αποτελούν μία λύση για την διαχείριση αυτού του όγκου δεδομένων. Παρόλο αυτό, η διαδικασία μάθησης που χρησιμοποιείται για την εκπαίδευση του δικτύου είναι καθοριστική για την απόδοση του μοντέλου. Τα προβλήματα που συναντάμε στον πραγματικό κόσμο συνήθως έχουν δυναμική φύση και είναι διαρκώς μεταβαλλόμενα, ενώ λαμβάνουν δεδομένα ασταμάτητα σε πραγματικό χρόνο. Οι τεχνικές μάθησης οι οποίες εκπαιδεύουν το δίκτυο μαζί με ένα πλήθος δεδομένων έχουν υψηλές υπολογιστικές απαιτήσεις. Με αποτέλεσμα να δημιουργούν σημαντικές δυσκολίες στην αναβάθμιση των μοντέλων στα νέα επίκαιρα δεδομένα, ή ακόμα χειρότερα σε μοντέλα που δεν αντιστοιχούν στα δεδομένα της επικαιρότητας. Λόγω αυτού έχουν αναπτυχθεί νέες τεχνικές μηχανικής μάθησης, οι οποίες αναβαθμίζουν το δίκτυο με την γνώση κάθε δεδομένου την ώρα που είναι διαθέσιμο με στόχο την αντιμετώπιση προβλημάτων μεταβαλλόμενης φύσης από ροές δεδομένων με μεγάλους ρυθμούς.

Ο αλγόριθμος RAN που παρουσιάστηκε στην [23] αποτελεί μία διαδικασία μηχανικής μάθησης για τεχνητά νευρωνικά δίκτυα RBF και παρέχει την δυνατότητα εκπαίδευσης του δικτύου από ροές σε πραγματικό χρόνο. Ο συγκεκριμένος αλγόριθμος μπορεί να προσθέτει νευρώνες στο δίκτυο με βάση έναν κανόνα καινοτομίας, χωρίς να χρειάζεται να επανεκπαιδεύσει όλες τις παραμέτρους από την αρχή. Ο αλγόριθμος RAN, επιπλέον, επιτυγχάνει την συνεχόμενη σταδιακή εκπαίδευση χωρίς να επανεκπαιδεύσει το δίκτυο με χρήση τεχνικών αναπροσαρμογής των διαστάσεων των απαραίτητων πινάκων. Η διαδικασία RAN αναβαθμίζει τις συντεταγμένες των κέντρων και τα συνάπτικά βάρη σαν παραμέτρους προς εκπαίδευση του δικτύου. Η καινοτομία του αλγορίθμου RAN αποτέλεσε πηγή έμπνευσης για την δημιουργία μιας οικογένειας αλγορίθμων που βασίστηκαν σε αυτόν όπως οι RAN-EKF, M-RAN και GAP. Οι μεταγενέστερες εναλλακτικές εκδόσεις του RAN προσέφεραν τεχνικές διαγραφής νευρώνων, διαφοροποιημένες μεθόδους επιλογής νέων κέντρων και εναλλακτικές μεθόδους αναβάθμισης των συνάψεων [23]–[26].

Μία ακόμα μέθοδος μηχανικής μάθησης νευρωνικών δικτύων RBF αναπτύχθηκε στην [21] με στόχο την αντιμετώπιση προβλημάτων δυναμικής φύσεως, την οποία αναφέρουμε με τον τίτλο FUZZY-LS-RLS. Η συγκεκριμένη διαδικασία μάθησης χωρίζει την διαδικασία εκπαίδευσης σε δύο στάδια. Στο πρώτο στάδιο τα κέντρα στα οποία θα τοποθετηθούν οι ακτινικές συναρτήσεις βάσεις καθορίζονται με μια διαδικασία ασαφούς λογικής. Ο υπολογισμός της τοποθεσίας των κέντρων αποτελεί ανεξάρτητη διαδικασία από την διαδικασία υπολογισμού των συναπτικών βαρών και τον αναλαμβάνει αποκλειστικά η διαδικασία ασαφούς λογικής. Αυτός ο διαχωρισμός της διαδικασίας μάθησης σε δύο στάδια προσφέρει σημαντικά πλεονεκτήματα στην διαδικασία σε θέματα

ταχύτητας και αποδόσεις.

Οι παραπάνω αλγόριθμοι έχουν περιθώρια βελτίωσης δεδομένου ότι διακρίνονται από κάποια μειονεκτήματα. Οι αλγόριθμοι της οικογένειας RAN δεν εκμεταλλεύονται το πλεονέκτημα του διαχωρισμού της διαδικασίας μάθησης σε δύο στάδια. Αυτό σημαίνει ότι για κάθε νέο κέντρο που προστίθεται στο δίκτυο, επιβαρύνεται η διαδικασία αναβαθμίσεις με μία μεταβλητή για κάθε συντεταγμένη του νέου κέντρου. Κατά αυτόν τον τρόπο μπορεί εύκολα να αυξηθεί η πολυπλοκότητα του προβλήματος, αλλά και οι υπολογιστικές απαιτήσεις του. Αντίθετα με τους RAN, ο αλγόριθμος FUZZY-LS-RLS με χρήση ασαφούς λογικής παρέχει το πλεονέκτημα του διαχωρισμού της διαδικασίας μάθησης. Παρά αυτό το πλεονέκτημα, ο FUZZY-LS-RLS κάθε φορά που χρειάζεται να προσθέσει ή να αφαιρέσει ένα κέντρο, εκπαιδεύει το δίκτυο από την αρχή από ένα πλήθος δεδομένων που αποθηκεύονται σύμφωνα με το μέγεθος ενός κυλιόμενου παραθύρου. Το οποίο παράθυρο, όμως, δημιουργεί σημαντικά μειονεκτήματα στην διαδικασία μάθησης όπως η αύξηση του υπολογιστικού κόστους κάθε φορά που χρειάζεται να προστεθεί ή να αφαιρεθεί ένα κέντρο και οι μεγαλύτερες απαιτήσεις μνήμης για την αποθήκευση των δεδομένων που ορίζει το παράθυρο. Επιπρόσθετα, επειδή το μέγεθος του παραθύρου σε κάποια προβλήματα έχει περιορισμούς λόγω υπολογιστικού κόστους, σημαντική πληροφορία μπορεί να χαθεί από παλαιότερα δεδομένα που είχαν ήδη εκπαιδεύσει το δίκτυο με αποτέλεσμα την μείωση της απόδοσης.

Η δημιουργία ενός αλγόριθμου ο οποίος θα μειώνει το υπολογιστικό κόστος κρατώντας την ακρίβεια σε υψηλά επίπεδα έχει μεγάλη αξία σε προβλήματα με μεταβαλλόμενη φύση και συνεχόμενη ροή δεδομένων. Η ταχύτητα και η απόδοση είναι και τα δύο πολύ σημαντικά χαρακτηριστικά, έτσι ώστε η διαδικασία να διατηρεί ενημερωμένο το μοντέλο με τα νέα δεδομένα χωρίς ανεπιθύμητες εκκρεμότητες λόγω υπολογιστικού φορτίου. Τα μειονεκτήματα που διακρίνουν τις διαδικασίες εκπαίδευσης της οικογένειας RAN, αλλά και της διαδικασίας FUZZY-LS-RLS, ενέπνευσαν την δημιουργία αυτής της πτυχιακής εργασίας. Σκοπός της είναι η δημιουργία μιας νέας διαδικασίας μάθησης, η οποία θα διατηρεί τα πλεονεκτήματα των παραπάνω αλγορίθμων χωρίς, όμως, τα μειονεκτήματα. Αυτό το εγχείρημα οδήγησε σε μία νέα πρόταση που διατηρεί τον διαχωρισμό της διαδικασίας μάθησης σε δύο στάδια με την επιλογή των κέντρων να γίνεται αποκλειστικά από την διαδικασία ασαφούς λογικής. Ταυτόχρονα, η προτεινόμενη διαδικασία διατηρεί τις τεχνικές αναπροσαρμογής των διαστάσεων των απαραίτητων πινάκων των αλγορίθμων RAN. Κατά αυτόν τον τρόπο πετύχει την συνεχόμενη σταδιακή εκπαίδευση χωρίς να εκπαιδεύει το δίκτυο από την αρχή κάθε φορά που αλλάζει η υποδομή του. Ο συνδυασμός αυτός οδήγησε στην δημιουργία μιας νέας διαδικασίας την οποία αποκαλούμε με το όνομα FM-RLS. Αποφεύγοντας την χρήση επανεκπαίδευσης όχι μόνο η διαδικασία μπορεί να είναι ταχύτερη, αλλά και με μικρότερες απαιτήσεις μνήμης, αφού πλέον δεν χρειάζεται η αποθήκευση δεδομένων με χρήση κυλιόμενου παραθύρου. Μία ακόμα σημαντική προσθήκη της προτεινόμενης διαδικασίας είναι ότι διατηρεί όλα τα πλεονεκτήματα του αλγορίθμου του διαχωρισμού της εκπαίδευσης σε δύο στάδια. Αυτό σημαίνει ότι σε αντίθεση με τις διαδικασίες RAN που εισάγουν την τοποθεσία των κέντρων στην διαδικασία μάθησης, ο αλγόριθμος FM-RLS χωρίζει την επιλογή και την τοποθέτηση των κέντρων από την αναβάθμιση των μεταβλητών εκπαίδευσης του δικτύου. Με τον διαχωρισμό αυτό σε συνδυασμό με την χρήση της ακτινικής συνάρτησης βάσης Thin-Plate-Spline στην οποία δεν απαιτείται ο υπολογισμός της μεταβλητής διασποράς “σ” της συνάρτησης Gauss, επιτυγχάνεται η μείωση του υπολογιστικού κόστους, άλλα και η μείωση της πολυπλοκότητας του προβλήματος. Επιπρόσθετα, η υποδομή του νευρωνικού δικτύου που εκπαιδεύτηκε ήταν της μορφής MIMO (Multi Input Multi output) με στόχο την ισχυροποίηση των προβλημάτων με πολλές κλάσεις απασχολώντας μία έξοδο για κάθε κλάση. Το οποίο αυξάνει τις υπολογιστικές απαιτήσεις της διαδικασίας με επιθυμητό τρόπο δεδομένου ότι προβλήματα με πολλές κλάσεις έχουν συχνά μεγαλύτερη πολυπλοκότητα.

Στα πλαίσια της πειραματικής διαδικασίας ο νέος αλγόριθμος FM-RLS συγκρίθηκε σε ένα πλήθος πραγματικών προβλημάτων κατηγοριοποίησης με τις διαδικασίες μάθησης FUZZY-LS-RLS, RAN, RAN-EKF, M-RAN και GAP. Οι διαδικασίες συγκρίθηκαν σε επίπεδο ταχύτητας

και ακριβείας αποτελεσμάτων, ταυτόχρονα, και στην συνέχεια αξιολογήθηκαν με ένα σύστημα βαθμολόγησης προκειμένου να διαπιστωθεί η απόδοσή τους συνδυαστικά. Τα αποτελέσματα της πειραματικής διαδικασίας έδειξαν ότι η διαδικασία FM-RLS αποτελεί μία αποδοτικότερη, αλλά και γρήγορη διαδικασία με καλύτερες ικανότητες γενίκευσης σε αναλογία με τις διαδικασίες που συγκρίθηκε. Σύμφωνα με τα αποτελέσματα ο αλγόριθμος FM-RLS αποτελεί μία νέα δυναμική πρόταση στο πεδίο των Online αλγορίθμων εκπαίδευσης νευρωνικών δικτύων RBF για προβλήματα κατηγοριοποίησης δεδομένων μεταβαλλόμενης φύσεως.

Σχέδια που αφορούν μελλοντική ερευνητική δραστηριότητα έχουν να κάνουν με την ενσωμάτωση της διαδικασίας μάθησης FM-RLS σε Μοντέλα Προβλεπτικού Ελέγχου (Model Predictive Control, MPC) [29], [45]–[49] για τον έλεγχο μη γραμμικών συστημάτων.

References

- [1] M. Chen, S. Mao, and Y. Liu, «Big data: A survey,» *Mob. Networks Appl.*, vol. 19, no. 2, pp. 171–209, 2014, ISSN: 1383469X. DOI: [10.1007/s11036-013-0489-0](https://doi.org/10.1007/s11036-013-0489-0).
- [2] V. Losing, B. Hammer, and H. Wersing, «Incremental on-line learning: A review and comparison of state of the art algorithms,» *Neurocomputing*, vol. 275, pp. 1261–1274, 2018, ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2017.06.084>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231217315928>.
- [3] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd. Upper Saddle River, NJ: Prentice Hall, 1999.
- [4] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, and F. Herrera, «A survey on data preprocessing for data stream mining: Current status and future directions,» *Neurocomputing*, vol. 239, pp. 39–57, 2017, ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2017.01.078>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231217302631>.
- [5] D. Brzezinski, «Encyclopedia of Machine Learning and Data Science,» *Encycl. Mach. Learn. Data Sci.*, no. November, 2020. DOI: [10.1007/978-1-4899-7502-7](https://doi.org/10.1007/978-1-4899-7502-7).
- [6] Y. Lu, K. Boukharouba, J. Boonaert, A. Fleury, and S. Lecoeuche, «Application of an Incremental SVM Algorithm for On-line Human Recognition from Video Surveillance Using Texture and Color Features,» *Neurocomputing*, vol. 126, pp. 132–140, Feb. 2014. DOI: [10.1016/j.neucom.2012.08.071](https://doi.org/10.1016/j.neucom.2012.08.071). [Online]. Available: <https://hal.science/hal-00988202>.
- [7] E. Jothimurugesan, A. Tahmasbi, P. Gibbons, and S. Tirthapura, «Variance-reduced stochastic gradient descent on streaming data,» in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/ceb648f9146a6345d604ab093b02c73-Paper.pdf.
- [8] C. Salperwyck, V. Lemaire, and C. Hue, «Incremental weighted naive bays classifiers for data stream,» in *Data Science, Learning by Latent Structures, and Knowledge Discovery*, B. Lausen, S. Krolak-Schwerdt, and M. Böhmer, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 179–190, ISBN: 978-3-662-44983-7. DOI: [10.1007/978-3-662-44983-7_16](https://doi.org/10.1007/978-3-662-44983-7_16).
- [9] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfharinger, G. Holmes, and T. Abdessalem, «Adaptive random forests for evolving data stream classification,» *Machine Learning*, vol. 106, no. 9, pp. 1469–1495, Oct. 2017, ISSN: 1573-0565. DOI: [10.1007/s10994-017-5642-8](https://doi.org/10.1007/s10994-017-5642-8). [Online]. Available: <https://doi.org/10.1007/s10994-017-5642-8>.

- [10] S. Xu and J. Wang, «A fast incremental extreme learning machine algorithm for data streams classification,» *Expert Systems with Applications*, vol. 65, pp. 332–344, 2016, ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2016.08.052>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417416304511>.
- [11] J. Moody and C. J. Darken, «Fast Learning in Networks of Locally-Tuned Processing Units,» *Neural Computation*, vol. 1, no. 2, pp. 281–294, Jun. 1989, ISSN: 0899-7667. DOI: [10.1162/neco.1989.1.2.281](https://doi.org/10.1162/neco.1989.1.2.281). eprint: <https://direct.mit.edu/neco/article-pdf/1/2/281/811853/neco.1989.1.2.281.pdf>. [Online]. Available: <https://doi.org/10.1162/neco.1989.1.2.281>.
- [12] J. Leonard and M. Kramer, «Radial basis function networks for classifying process faults,» *IEEE Control Systems Magazine*, vol. 11, no. 3, pp. 31–38, 1991. DOI: [10.1109/37.75576](https://doi.org/10.1109/37.75576).
- [13] A. Leonardis and H. Bischof, «An efficient mdl-based construction of rbf networks,» *Neural Networks*, vol. 11, no. 5, pp. 963–973, 1998, ISSN: 0893-6080. DOI: [https://doi.org/10.1016/S0893-6080\(98\)00051-3](https://doi.org/10.1016/S0893-6080(98)00051-3). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608098000513>.
- [14] J. MacQueen *et al.*, «Some methods for classification and analysis of multivariate observations,» in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Oakland, CA, USA, vol. 1, 1967, pp. 281–297.
- [15] C. Darken and J. Moody, «Fast adaptive k-means clustering: Some empirical results,» in *1990 IJCNN International Joint Conference on Neural Networks*, 1990, 233–238 vol.2. DOI: [10.1109/IJCNN.1990.137720](https://doi.org/10.1109/IJCNN.1990.137720).
- [16] H. Sarimveis, A. Alexandridis, G. Tsekouras, and G. Bafas, «A fast and efficient algorithm for training radial basis function neural networks based on a fuzzy partition of the input space,» *Industrial & Engineering Chemistry Research*, vol. 41, no. 4, pp. 751–759, 2002. DOI: [10.1021/ie010263h](https://doi.org/10.1021/ie010263h). eprint: <https://doi.org/10.1021/ie010263h>. [Online]. Available: <https://doi.org/10.1021/ie010263h>.
- [17] D. Karamichailidou, V. Kaloutsas, and A. Alexandridis, «Wind turbine power curve modeling using radial basis function neural networks and tabu search,» *Renewable Energy*, vol. 163, pp. 2137–2152, 2021, ISSN: 0960-1481. DOI: <https://doi.org/10.1016/j.renene.2020.10.020>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0960148120315871>.
- [18] D. Karamichailidou, A. Alexandridis, G. Anagnostopoulos, G. Syriopoulos, and O. Sekkas, «Modeling biogas production from anaerobic wastewater treatment plants using radial basis function networks and differential evolution,» *Computers Chemical Engineering*, vol. 157, p. 107629, 2022, ISSN: 0098-1354. DOI: <https://doi.org/10.1016/j.compchemeng.2021.107629>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0098135421004075>.
- [19] A. Alexandridis, H. Sarimveis, and K. Ninos, «A radial basis function network training algorithm using a non-symmetric partition of the input space – application to a model predictive control configuration,» *Advances in Engineering Software*, vol. 42, no. 10, pp. 830–837, 2011, ISSN: 0965-9978. DOI: <https://doi.org/10.1016/j.advengsoft.2011.05.026>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0965997811001335>.
- [20] A. Alexandridis, E. Chondrodima, and H. Sarimveis, «Radial basis function network training using a nonsymmetric partition of the input space and particle swarm optimization,» *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 2, pp. 219–230, Feb. 2013, ISSN: 2162-2388. DOI: [10.1109/TNNLS.2012.2227794](https://doi.org/10.1109/TNNLS.2012.2227794).

- [21] A. Alexandridis, H. Sarimveis, and G. Bafas, «A new algorithm for online structure and parameter adaptation of rbf networks,» *Neural Networks*, vol. 16, no. 7, pp. 1003–1017, 2003, ISSN: 0893-6080. DOI: [https://doi.org/10.1016/S0893-6080\(03\)00052-2](https://doi.org/10.1016/S0893-6080(03)00052-2). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608003000522>.
- [22] D. Karamichailidou, S. Koletsios, and A. Alexandridis, «An rbf online learning scheme for non-stationary environments based on fuzzy means and givens rotations,» *Neurocomputing*, vol. 501, pp. 370–386, 2022, ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2022.06.016>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231222007305>.
- [23] J. Platt, «A Resource-Allocating Network for Function Interpolation,» *Neural Computation*, vol. 3, no. 2, pp. 213–225, Jun. 1991, ISSN: 0899-7667. DOI: [10.1162/neco.1991.3.2.213](https://doi.org/10.1162/neco.1991.3.2.213). eprint: <https://direct.mit.edu/neco/article-pdf/3/2/213/812123/neco.1991.3.2.213.pdf>. [Online]. Available: <https://doi.org/10.1162/neco.1991.3.2.213>.
- [24] V. Kadirkamanathan and M. Niranjan, «A Function Estimation Approach to Sequential Learning with Neural Networks,» *Neural Computation*, vol. 5, no. 6, pp. 954–975, Nov. 1993, ISSN: 0899-7667. DOI: [10.1162/neco.1993.5.6.954](https://doi.org/10.1162/neco.1993.5.6.954). eprint: <https://direct.mit.edu/neco/article-pdf/5/6/954/812670/neco.1993.5.6.954.pdf>. [Online]. Available: <https://doi.org/10.1162/neco.1993.5.6.954>.
- [25] L. Yingwei, N. Sundararajan, and P. Saratchandran, «A Sequential Learning Scheme for Function Approximation Using Minimal Radial Basis Function Neural Networks,» *Neural Computation*, vol. 9, no. 2, pp. 461–478, Feb. 1997, ISSN: 0899-7667. DOI: [10.1162/neco.1997.9.2.461](https://doi.org/10.1162/neco.1997.9.2.461). eprint: <https://direct.mit.edu/neco/article-pdf/9/2/461/813602/neco.1997.9.2.461.pdf>. [Online]. Available: <https://doi.org/10.1162/neco.1997.9.2.461>.
- [26] G.-B. Huang, P. Saratchandran, and N. Sundararajan, «An efficient sequential learning algorithm for growing and pruning rbf (gap-rbf) networks,» *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 6, pp. 2284–2292, Dec. 2004, ISSN: 1941-0492. DOI: [10.1109/TSMCB.2004.834428](https://doi.org/10.1109/TSMCB.2004.834428).
- [27] S. Suresh, K. Dong, and H. Kim, «A sequential learning algorithm for self-adaptive resource allocation network classifier,» *Neurocomputing*, vol. 73, no. 16, pp. 3012–3019, 2010, 10th Brazilian Symposium on Neural Networks (SBRN2008), ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2010.07.003>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231210003243>.
- [28] W. Song, J. Z. Liang, X. L. He, and P. Chen, «Taking advantage of improved resource allocating network and latent semantic feature selection approach for automated text categorization,» *Applied Soft Computing*, vol. 21, pp. 210–220, 2014, ISSN: 1568-4946. DOI: <https://doi.org/10.1016/j.asoc.2014.03.002>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1568494614001082>.
- [29] A. Alexandridis, E. Chondrodima, N. Giannopoulos, and H. Sarimveis, «A fast and efficient method for training categorical radial basis function networks,» *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 11, pp. 2831–2836, Nov. 2017, ISSN: 2162-2388. DOI: [10.1109/TNNLS.2016.2598722](https://doi.org/10.1109/TNNLS.2016.2598722).

- [30] A. Alexandridis, M. Stogiannos, A. Kyriou, and H. Sarimveis, «An offset-free neural controller based on a non-extrapolating scheme for approximating the inverse process dynamics,» *Journal of Process Control*, vol. 23, no. 7, pp. 968–979, 2013, ISSN: 0959-1524. DOI: <https://doi.org/10.1016/j.jprocont.2013.04.008>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0959152413001078>.
- [31] A. Alexandridis, E. Chondrodima, K. Moutzouris, and D. Triantis, «A neural network approach for the prediction of the refractive index based on experimental data,» *Journal of Materials Science*, vol. 47, no. 2, pp. 883–891, Jan. 2012, ISSN: 1573-4803. DOI: [10.1007/s10853-011-5868-y](https://doi.org/10.1007/s10853-011-5868-y). [Online]. Available: <https://doi.org/10.1007/s10853-011-5868-y>.
- [32] N. Giamarelos, M. Papadimitrakis, M. Stogiannos, E. N. Zois, N.-A. I. Livanos, and A. Alexandridis, «A machine learning model ensemble for mixed power load forecasting across multiple time horizons,» *Sensors*, vol. 23, no. 12, 2023, ISSN: 1424-8220. DOI: [10.3390/s23125436](https://doi.org/10.3390/s23125436). [Online]. Available: <https://www.mdpi.com/1424-8220/23/12/5436>.
- [33] D. Kagkas, D. Karamichailidou, and A. Alexandridis, «Chess position evaluation using radial basis function neural networks,» *Complexity*, vol. 2023, p. 7143943, Apr. 2023, ISSN: 1076-2787. DOI: [10.1155/2023/7143943](https://doi.org/10.1155/2023/7143943). [Online]. Available: <https://doi.org/10.1155/2023/7143943>.
- [34] A. Alexandridis, M. Stogiannos, N. Papaioannou, E. Zois, and H. Sarimveis, «An inverse neural controller based on the applicability domain of rbf network models,» *Sensors*, vol. 18, no. 1, 2018, ISSN: 1424-8220. DOI: [10.3390/s18010315](https://doi.org/10.3390/s18010315). [Online]. Available: <https://www.mdpi.com/1424-8220/18/1/315>.
- [35] A. Alexandridis and E. Chondrodima, «A medical diagnostic tool based on radial basis function classifiers and evolutionary simulated annealing,» *Journal of Biomedical Informatics*, vol. 49, pp. 61–72, 2014, ISSN: 1532-0464. DOI: <https://doi.org/10.1016/j.jbi.2014.03.008>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1532046414000653>.
- [36] A. ALEXANDRIDIS, «Evolving rbf neural networks for adaptive soft-sensor design,» *International Journal of Neural Systems*, vol. 23, no. 06, p. 1350029, 2013, PMID: 24156672. DOI: [10.1142/S0129065713500299](https://doi.org/10.1142/S0129065713500299). eprint: <https://doi.org/10.1142/S0129065713500299>. [Online]. Available: <https://doi.org/10.1142/S0129065713500299>.
- [37] A. Alexandridis, C. Siettos, H. Sarimveis, A. Boudouvis, and G. Bafas, «Modelling of nonlinear process dynamics using kohonen’s neural networks, fuzzy systems and chebyshev series,» *Computers Chemical Engineering*, vol. 26, no. 4, pp. 479–486, 2002, ISSN: 0098-1354. DOI: [https://doi.org/10.1016/S0098-1354\(01\)00785-2](https://doi.org/10.1016/S0098-1354(01)00785-2). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0098135401007852>.
- [38] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, «MOA: massive online analysis,» 2010, pp. 1601–1604. DOI: [10.5555/1756006.1859903](https://doi.org/10.5555/1756006.1859903). [Online]. Available: <https://dl.acm.org/doi/10.5555/1756006.1859903>.
- [39] M. Kelly, R. Longjohn, and K. Nottingham, *Uci machine learning repository*, 2010. [Online]. Available: <https://archive.ics.uci.edu>.
- [40] M. Harries, N. S. Wales, *et al.*, «Splice-2 comparative evaluation: Electricity pricing,» 1999.

- [41] G. Hulten, L. Spencer, and P. Domingos, «Mining time-changing data streams,» in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '01, San Francisco, California: Association for Computing Machinery, 2001, pp. 97–106, ISBN: 158113391X. DOI: [10.1145/502512.502529](https://doi.org/10.1145/502512.502529). [Online]. Available: <https://doi.org/10.1145/502512.502529>.
- [42] W. N. Street and Y. Kim, «A streaming ensemble algorithm (sea) for large-scale classification,» in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '01, San Francisco, California: Association for Computing Machinery, 2001, pp. 377–382, ISBN: 158113391X. DOI: [10.1145/502512.502568](https://doi.org/10.1145/502512.502568). [Online]. Available: <https://doi.org/10.1145/502512.502568>.
- [43] *U.S. National Oceanic and Atmospheric Administration (NOAA) Weather data*, 2010. [Online]. Available: <https://users.rowan.edu/~polikar/nse.html>.
- [44] G. Ditzler and R. Polikar, «Incremental learning of concept drift from streaming imbalanced data,» *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 10, pp. 2283–2301, 2013. DOI: [10.1109/TKDE.2012.136](https://doi.org/10.1109/TKDE.2012.136).
- [45] I. Kalogeropoulos, A. Alexandridis, and H. Sarimveis, «Economic oriented dynamic matrix control of wastewater treatment plants,» *Journal of Process Control*, vol. 118, pp. 202–217, 2022, ISSN: 0959-1524. DOI: <https://doi.org/10.1016/j.jprocont.2022.08.006>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0959152422001482>.
- [46] A. Kapnopoulos and A. Alexandridis, «A cooperative particle swarm optimization approach for tuning an mpc-based quadrotor trajectory tracking scheme,» *Aerospace Science and Technology*, vol. 127, p. 107725, 2022, ISSN: 1270-9638. DOI: <https://doi.org/10.1016/j.ast.2022.107725>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1270963822003996>.
- [47] M. Papadimitrakis and A. Alexandridis, «Active vehicle suspension control using road preview model predictive control and radial basis function networks,» *Applied Soft Computing*, vol. 120, p. 108646, 2022, ISSN: 1568-4946. DOI: <https://doi.org/10.1016/j.asoc.2022.108646>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1568494622001296>.
- [48] M. Papadimitrakis, M. Stogiannos, H. Sarimveis, and A. Alexandridis, «Multi-ship control and collision avoidance using mpc and rbf-based trajectory predictions,» *Sensors*, vol. 21, no. 21, 2021, ISSN: 1424-8220. DOI: [10.3390/s21216959](https://doi.org/10.3390/s21216959). [Online]. Available: <https://www.mdpi.com/1424-8220/21/21/6959>.
- [49] M. Stogiannos, A. Alexandridis, and H. Sarimveis, «Model predictive control for systems with fast dynamics using inverse neural models,» *ISA Transactions*, vol. 72, pp. 161–177, 2018, ISSN: 0019-0578. DOI: <https://doi.org/10.1016/j.isatra.2017.09.016>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0019057817305657>.

