



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική Εργασία

Εξίσωση μη γραμμικών καναλιών με χρήση τεχνικών Μηχανικής Μάθησης



Φοιτητής: Ανδρέας Καταβάτης
ΑΜ: 50345310

Επιβλέπων Καθηγητής

Ευάγγελος Ζέρβας
Καθηγητής

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, ΟΚΤΩΒΡΙΟΣ 2021



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

Diploma Thesis

Non Linear channel equalization using Machine Learning Techniques



Student: Andreas Katavatis

Supervisor

Evangelos Zervas
Professor

ATHENS-EGALEO, OCTOBER 2021

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

Ευάγγελος Ζέρβας Καθηγητής	Δημήτριος Καλύβας Καθηγητής	Σωτήριος Καραμπέτσος Αναπληρωτής Καθηγητής
(Υπογραφή)	(Υπογραφή)	(Υπογραφή)

Ανδρέας Καταβάτης, Σεπτέμβριος, 2021

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 ΠΕΡΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ ΚΑΙ ΛΟΓΟΚΛΟΠΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπόγραφα ότι η παρούσα εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα αποκλειστικά και ότι είμαι ο αποκλειστικός συγγραφέας του κειμένου της.

Η εργασία μου δεν προσβάλλει οποιασδήποτε μορφής δικαιώματα πνευματικής ιδιοκτησίας, προσωπικότητας ή προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής ή λογοκλοπής.

Κάθε βοήθεια που έλαβα για την ολοκλήρωση της εργασίας είναι αναγνωρισμένη και αναφέρεται λεπτομερώς στο κείμενό της. Ειδικότερα, έχω αναφέρει ευδιάκριτα μέσα στο κείμενο και με την κατάλληλη παραπομπή όλες τις πηγές δεδομένων, κώδικα προγραμματισμού Η/Υ, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών που χρησιμοποιήθηκαν, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης, και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Επιπλέον, όλες οι πηγές που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης κατά τα διεθνή πρότυπα.

Τέλος δηλώνω ενυπόγραφα ότι αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της είναι προϊόν λογοκλοπής.

Ημερομηνία 13/10/2021



(Υπογραφή)

Θα ήθελα να ευχαριστήσω όλους όσους με βοήθησαν με οποιονδήποτε τρόπο για την ολοκλήρωση της παρούσας πτυχιακής εργασίας.

Πρώτα απ' όλα, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Ευάγγελο Ζέρβα για την δυνατότητα που μου έδωσε, να ασχοληθώ με το συγκεκριμένο αντικείμενο και το θέμα που μου ανέθεσε. Επίσης, τον ευχαριστώ για την άψογη συνεργασία καθώς και για τις συμβουλές που μου έδωσε τόσο σε θέματα που αφορούσαν την παρούσα διπλωματική εργασία όσο και σε θέματα σπουδών και γενικής καθοδήγησης. Υπήρξε υποστηρικτής και καθοδηγητής σε όλη τη διάρκεια της διπλωματικής μου και αποτέλεσε πρότυπο τόσο καθηγητή όσο και ανθρώπου.

Ένα μεγάλο ευχαριστώ στον κ. Δημήτρη Καλύβα και στα υπόλοιπα μέλη της εργαστηριακής ομάδα του TelSiP που μου έδωσαν τη δυνατότητα να συμμετέχω στις εβδομαδιαίες συναντήσεις τους, όπου με ενθάρρυναν συνεχώς να εκφράσω τις απορίες μου και με έκαναν να νοιώσω και εγώ ένα ανεπίσημο μέλος αυτής της ομάδας.

Τέλος, ευχαριστώ τους γονείς μου και την αδερφή μου τόσο για την οικονομική αρωγή όλα αυτά τα χρόνια όσο και για την ηθική ενίσχυση και την συνεχή υποστήριξη τους σε κάθε προσπάθειά μου.

Περίληψη

Η έννοια της εξίσωσης καναλιού αναφέρεται σε οποιαδήποτε τεχνική επεξεργασίας σήματος που χρησιμοποιείται στον δέκτη για την αντιμετώπιση των παραμορφώσεων που εισάγονται από το κανάλι επικοινωνίας. Με την άνοδο των νευρωνικών δικτύων τα τελευταία χρόνια έχει δοκιμαστεί και η εφαρμογή τους στον τομέα των τηλεπικοινωνιών. Οι τεχνικές μηχανικής μάθησης έχουν δείξει σημαντικές βελτιώσεις όσον αφορά την εξίσωση καναλιού σε σύγκριση με τις κλασικές μεθόδους. Τα τεχνητά νευρωνικά δίκτυα χωρίζονται σε δίκτυα πρόσθιας τροφοδότησης και σε αναδρομικά νευρωνικά δίκτυα. Και οι δύο αυτές κατηγορίες παρουσιάζουν ομοιότητες στην αρχιτεκτονική τους με τους γραμμικούς εξισωτές αλλά και με τους εξισωτές ανατροφοδότησης απόφασης. Σε αυτή τη διπλωματική εργασία μελετάται η λύση για το πρόβλημα της εξίσωσης μη γραμμικού καναλιού χρησιμοποιώντας τεχνητά νευρωνικά δίκτυα αφού τα νευρωνικά δίκτυα από τη φύση τους λύνουν μη γραμμικά προβλήματα λόγω των συναρτήσεων ενεργοποίησης που χρησιμοποιούν. Αρχικά γίνεται μία επισκόπηση για την ανάγκη εξίσωσης καναλιού, και τις κλασικές τεχνικές που εφαρμόζονται για τη λύση του προβλήματος. Έπειτα αναλύονται τα τεχνητά νευρωνικά δίκτυα ενώ στη συνέχεια παρουσιάζονται και αναλύονται τα αποτελέσματα που έχουν οδηγήσει στους εξισωτές που βασίζονται σε τεχνητά νευρωνικά δίκτυα. Στη συνέχεια πραγματοποιήθηκε η υλοποίηση ενός εξισωτή βασισμένου σε δίκτυο πρόσθιας τροφοδότησης και εκτελέστηκαν τρεις προσομοιώσεις με τελικό στόχο την εξίσωση μη γραμμικού καναλιού.

Λέξεις – κλειδιά

Εξίσωση καναλιού, Τεχνητά νευρωνικά δίκτυα, Τεχνητά νευρωνικά δίκτυα πρόσθιας τροφοδότησης, Αναδρομικά νευρωνικά δίκτυα, Κανάλια επικοινωνίας, Εξισωτές, Διασυμβολική παρεμβολή

Abstract

Channel equalization is a term that is used for any signal processing technique that is used on the receiver. With the recent rise of neural networks in the last couple of years, their application in the field of telecommunications has been tested. Machine learning techniques have shown significant improvements in channel equalization compared to traditional channel equalization techniques. Neural networks can be categorized in feed forward and recurrent. Both of these categories have a similar architecture with linear and decision-feedback equalizers. In this diploma thesis, an equalizer that is based on neural networks is being studied to solve the problem of channel equalization of a nonlinear channel since neural networks specialize in solving nonlinear problems. Firstly, the need for channel equalization, channel equalization, and the traditional techniques used to counter the problem are being overviewed. An overview of artificial neural networks comes after, followed by the review of the research that have led to the neural network-based equalizers. Lastly, a feed forward neural network-based equalizer was created and three simulations were performed with the ultimate goal being the equalization of a nonlinear channel.

Keywords

Channel equalization, Artificial neural networks, Feed forward neural networks, Recurrent neural networks, Communication channels, Equalizers, Intersymbol interference

Περιεχόμενα

Κατάλογος Πινάκων.....	10
Κατάλογος Σχημάτων	10
Αλφαβητικό Ευρετήριο	12
ΕΙΣΑΓΩΓΗ.....	13
ΚΕΦΑΛΑΙΟ 1^ο : Εξίσωση καναλιού.....	15
1.1 Βασικά είδη καναλιών	15
1.1.1 Κανάλι Προσθετικού Λευκού Γκαουσιανού Θορύβου (AWGN)	15
1.1.2 AWGN κανάλι περιορισμένου εύρους ζώνης	16
1.2 Διασυμβολική Παρεμβολή (Intersymbol Interference-ISI)	17
1.2.1 Ορισμός ISI.....	17
1.2.2 Μαθηματική ερμηνεία και αποτροπή ISI.....	18
1.2.3 Κριτήριο Nyquist για μηδενική ISI.....	20
1.3 Σχεδιασμός φίλτρων για γνωστό κανάλι.....	21
1.4 Εξίσωση καναλιού για μη γνωστά κανάλια.....	23
1.4.1 Γραμμικοί εξισωτές (Linear equalizers).....	25
1.4.2 Προσαρμοστικοί εξισωτές.....	28
1.4.3 Εξισωτής ανατροφοδότησης απόφασης.....	30
ΚΕΦΑΛΑΙΟ 2^ο : Τεχνητά Νευρωνικά Δίκτυα	32
2.1 Γενικά	32
2.1.1 Ορισμός	32
2.1.2 Δομή Νευρωνικού Δικτύου	32
2.2 Αρχιτεκτονική Νευρωνικού Δικτύου	33
2.2.1 Πρόσθια τροφοδότηση και ανατροφοδότηση.....	33
2.2.2 Συναρτήσεις ενεργοποίησης	34
2.3 Κυριότερες τοπολογίες ΤΝΔ	38
2.3.1 Perceptron	38
2.3.2 Multilayer Perceptron (MLP).....	39
2.3.3 Radial Basis Function (RBF).....	41
2.3.4 Recurrent Neural Networks (RNN)	41
2.3.5 Convolutional Neural Networks (CNN).....	42
2.4 Εκπαίδευση ΤΝΔ	42
2.4.1 Στάδια μηχανικής μάθησης.....	42
2.4.2 Βασικά είδη εκπαίδευσης	43
2.4.3 Μέθοδος οπισθοδιάδοσης σφάλματος (Error Backpropagation)	43
2.5 Παραδείγματα επίλυσης προβλημάτων με ΤΝΔ.....	45
2.5.1 Ταξινόμηση γραμμικά διαχωρίσιμων εισόδων με Perceptron.....	45
2.5.2 Επίλυση προβλήματος λογικής πύλης XOR με MLP	47
ΚΕΦΑΛΑΙΟ 3^ο : Εξίσωση καναλιού με χρήση ΤΝΔ	51
3.1 Εξίσωση καναλιού χρησιμοποιώντας MLP	51
3.2 RNN εξισωτής	53
3.2.1 Αλγόριθμος Real Time Recurrent Learning (RTRL).....	55
3.2.2 Long short-term memory ΤΝΔ εξισωτής.....	56
3.3 MLP-RBF εξισωτής.....	57
ΚΕΦΑΛΑΙΟ 4^ο : Πειραματικά αποτελέσματα	59

4.1	Πείραμα 1: Εξίσωση AWGN καναλιού περιορισμένου εύρους ζώνης με MLP και RNN	
εξισωτή	59	
4.1.1	Στόχος 1 ^{ου} πειράματος	59
4.1.2	Πειραματικό μέρος 1 ^{ου} πειράματος	59
4.1.3	Αποτελέσματα και συμπεράσματα 1 ^{ου} πειράματος	61
4.2	Πείραμα 2: Εξίσωση μη γραμμικού AWGN καναλιού περιορισμένου εύρους ζώνης με MLP εξισωτή	64
4.2.1	Στόχος 2 ^{ου} πειράματος	64
4.2.2	Πειραματικό μέρος 2 ^{ου} πειράματος	64
4.2.3	Αποτελέσματα και συμπεράσματα 2 ^{ου} πειράματος	66
	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	68
	Βιβλιογραφία – Αναφορές - Διαδικτυακές Πηγές	69
	Παράρτημα Α	74
	Παράρτημα Β.....	82
	Παράρτημα Γ.....	84
	Παράρτημα Δ.....	87
	Παράρτημα Ε.....	95

Κατάλογος Πινάκων

Πίνακας 4. 1: Παράμετροι 2 ^{ης} προσομοίωσης	60
Πίνακας 4. 2: Παράμετροι 3 ^{ης} προσομοίωσης	65

Κατάλογος Σχημάτων

Σχήμα 1. 1: Απεικόνιση λαμβανόμενου σήματος στην έξοδο AWGN καναλιού (Proakis and Salehi 2002).....	15
Σχήμα 1. 2: Κανάλι περιορισμένου εύρους ζώνης, μοντελοποιημένο ως γραμμικό φίλτρο (Proakis and Salehi 2002).....	17
Σχήμα 1. 3: Παραμόρφωση τετραγωνικού παλμού μέσω ενός φίλτρου RC (Ha 2010).	17
Σχήμα 1. 4: Παράδειγμα με είσοδο τετραγωνικού παλμού (Ha 2010).	18
Σχήμα 1. 5: Παράδειγμα με είσοδο παραμορφωμένου τετραγωνικού παλμού (Ha 2010).	18
Σχήμα 1. 6: Μπλόκ διάγραμμα ψηφιακού PAM συστήματος (Proakis and Salehi 2002).	19
Σχήμα 1. 7: Ισοδύναμο φίλτρο καναλιού διακριτού χρόνου (Proakis and Salehi 2002).	24
Σχήμα 1. 8: Εξισωτής επιβολής μηδενισμών όπου δέχεται ένα σήμα που έχει περάσει από ένα κανάλι με AWGN $r(t)$ και η έξοδος του $y(t)$ αποτελεί την είσοδο του φωρατή που παίρνει περιοδικά δείγματα για να λάβει την απόφαση της εξόδου.	25
Σχήμα 1. 9: Γραμμικός FIR εξισωτής (Hur et al. 2005).....	26
Σχήμα 1. 10: Εξισωτής MMSE (Arslan, Evans, and Kiaei 2000).....	28
Σχήμα 1. 11: Μπλοκ διάγραμμα ενός προσαρμοστικού εξισωτή.	29
Σχήμα 1. 12: Βασική δομή ενός DFE εξισωτή.....	30
Σχήμα 1. 13: Επίδοση διαφορετικών μεθόδων εξίσωσης καναλιού. (Atef and Zimmermann 2013).....	31
Σχήμα 2. 1: Νευρωνικό δίκτυο πρόσθιας τροφοδότησης (Quiza and Davim 2011).....	33
Σχήμα 2. 2: Νευρωνικό δίκτυο ανατροφοδότησης (Borah, Sarma, and Talukdar 2015).....	34
Σχήμα 2. 3: Δυναμική βηματική συνάρτηση (Siebenrock [2018] 2021)	35
Σχήμα 2. 4: Γραμμική συνάρτηση (“Understanding Activation Functions in Depth” 2019). .	35
Σχήμα 2. 5: Σιγμοειδής συνάρτηση(Siebenrock [2018] 2021).....	36
Σχήμα 2. 6: Υπερβολική εφαπτομένη συνάρτηση(Siebenrock [2018] 2021).	37
Σχήμα 2. 7: Σύγκριση σιγμοειδής και υπερβολικής εφαπτομένης συνάρτησης (Sharma et al. 2020).....	37

Σχήμα 2. 8: Ανορθωτική γραμμική συνάρτηση (ReLU) (Siebenrock [2018] 2021).	38
Σχήμα 2. 9: Τεχνητός νευρώνας Perceptron (“Perceptron” 2019).	39
Σχήμα 2. 10: Multilayer Perceptron με 3 στρώματα (Hassan et al. 2015).	40
Σχήμα 2. 11: Διαφορά τρόπου ταξινόμησης μεταξύ MLP και RBF (Chandradevan 2017). ...	41
Σχήμα 2. 12: Δημιουργία δύο γραμμικά διαχωρίσιμων ομάδων δεδομένων.	46
Σχήμα 2. 13: Δημιουργία δυαδικού ταξινομητή perceptron	46
Σχήμα 2. 14: Δημιουργία ορίου απόφασης (threshold) για ταξινόμηση στοιχείων.	47
Σχήμα 2. 15: Πίνακας αληθείας λογικής πύλης XOR(Seising 2018)	48
Σχήμα 2. 16: Δημιουργία δύο κλάσεων για δυαδική κατηγοριοποίηση.	48
Σχήμα 2. 17: Δημιουργία MLP με δύο κρυφά στρώματα.	49
Σχήμα 2. 18: Σύγκριση επιθυμητής και πραγματικής απόκρισης TNA.	49
Σχήμα 2. 19: Ταξινόμηση δύο κλάσεων με τη βοήθεια δύο χρωμάτων.....	50
Σχήμα 3. 1: Εξισωμένα σύμβολα διαμόρφωσης 4-QAM με χρήση εξισωτή αλγορίθμου LMS (Baloch, Baloch, and Unar 2012).	52
Σχήμα 3. 2: Εξισωμένα σύμβολα διαμόρφωσης 4-QAM με χρήση MLP TNA (Baloch, Baloch, and Unar 2012).....	52
Σχήμα 3. 3: Εξισωτής RNN (Lavanaia et al. 2015)	53
Σχήμα 3. 4: Σύγκριση απόδοσης RNN και MLP εξισωτών για μη γραμμικό κανάλι (Lavanaia et al. 2015).....	54
Σχήμα 3. 5: Σύγκριση BER μεταξύ RNN εξισωτή, γραμμικού εξισωτή και feedforward NN εξισωτή (Kechriotis, Zervas, and Manolakos 1994).	56
Σχήμα 3. 6: LSTM εξισωτής (X. Lu et al. 2019)	57
Σχήμα 3. 7: Σύγκριση επίδοσης του MLP-RBF εξισωτή με έναν MLP και έναν RBF (B. Lu and Evans 1999)	58
Σχήμα 4. 1: Διάγραμμα BER-SNR για κλασικές τεχνικές εξίσωσης καναλιού.....	61
Σχήμα 4. 2: Διάγραμμα BER-SNR για MLP εξισωτές.	62
Σχήμα 4. 3: Διάγραμμα BER-SNR MLP εξισωτών για υποδεέστερη εκπαίδευση.....	63
Σχήμα 4. 4: Διάγραμμα BER-SNR για RNN εξισωτή.	63
Σχήμα 4. 5: Μη γραμμικό κανάλι (Xu et al. 2018).	64
Σχήμα 4. 6: Διάγραμμα BER-SNR για MLP εξισωτές σε μη γραμμικό κανάλι.	66

Σχήμα 4. 7: Διάγραμμα BER-SNR για RNN εξισωτή σε μη γραμμικό κανάλι.....67

Αλφαβητικό Ευρετήριο

ANN: Artificial Neural Networks

MLP: Multilayer Perceptron

RNN: Recurrent Neural Network

RBF: Radial Basis Function

CNN: Convolutional Neural Network

ZF: Zero-forcing

LMS: Least Mean Square

MSE: Mean Square Error

MMSE: Minimum Mean Square Error

DFE: Decision Feedback Equalizer

TNΔ: Τεχνητά Νευρωνικά Δίκτυα

ISI: Intersymbol Interference

AWGN: Additive White Gaussian Noise

ΕΙΣΑΓΩΓΗ

Η σχεδίαση εξισωτών για εξίσωση καναλιού αποτελεί ένα σημαντικό κομμάτι των τηλεπικοινωνιών και ένα ενεργό αντικείμενο έρευνας και ανάπτυξης μέχρι και σήμερα. Η έννοια εξίσωση καναλιού αναφέρεται σε οποιαδήποτε τεχνική επεξεργασίας σήματος που χρησιμοποιείται στον δέκτη ενός τηλεπικοινωνιακού συστήματος για την αντιμετώπιση της διασυμβολικής παρεμβολής (Intersymbol Interference-ISI) που προκαλείται από ένα κανάλι. Ο στόχος αυτών των εξισωτών είναι να δημιουργήσουν ένα φίλτρο που έχει την αντίστροφη συμπεριφορά από ότι έχει το κανάλι προκειμένου να αντιμετωπίσουν την παραμόρφωση που εισάγεται από το κανάλι. Όταν τα χαρακτηριστικά του καναλιού του τηλεπικοινωνιακού συστήματος είναι γνωστά τότε ο σχεδιασμός φίλτρων εκπομπής και λήψης είναι μία προκαθορισμένη διαδικασία. Ο όρος εξίσωση καναλιού χρησιμοποιείται για κανάλια των οποίων τα χαρακτηριστικά απόκρισης συχνότητας είτε δεν είναι γνωστά είτε μεταβάλλονται με τον χρόνο. Για παράδειγμα, σε ένα τηλεφωνικό δίκτυο η διαδρομή του καναλιού αλλάζει κάθε φορά που ο χρήστης θα καλέσει έναν διαφορετικό αριθμό, αυτό αποτελεί ένα κανάλι του οποίου τα χαρακτηριστικά δεν είναι γνωστά εξ αρχής.

Οι εξισωτές φώρασης ακολουθίας μέγιστης πιθανοφάνειας προσφέρουν την καλύτερη επίδοση σε σχέση με άλλους εξισωτές. Ωστόσο, δεν είναι λίγες οι φορές που δεν επιλέγονται λόγω της μεγάλης υπολογιστικής πολυπλοκότητάς τους. Υποδεέστερες κλασικές λύσεις εξισωτών που κάποιες φορές προτιμώνται είναι ο εξισωτής επιβολής μηδενισμών και ο εξισωτής του ελαχίστου μέσου τετραγωνικού σφάλματος. Πρόκειται για γραμμικούς εξισωτές που παίρνουν το όνομα τους από το κριτήριο το οποίο εφαρμόζουν για τη λειτουργία τους.

Στην περίπτωση που τα χαρακτηριστικά του καναλιού μεταβάλλονται με τον χρόνο χρησιμοποιούνται προσαρμοστικοί εξισωτές οι οποίοι έχουν την ικανότητα να αλλάξουν τη δομή τους κατά της διάρκεια της μετάδοσης για να αντιμετωπίσουν αυτό το πρόβλημα.

Την καλύτερη επίδοση εξισωτή μετά τον MLSE την έχει ο εξισωτής ανατροφοδότησης λόγω της μη γραμμικότητας του ενώ ταυτόχρονα η υπολογιστική πολυπλοκότητα του είναι μικρότερη από αυτή του MLSE.

Με τη ραγδαία εξέλιξη της τεχνητής νοημοσύνης τα τελευταία χρόνια ξεκίνησε η έρευνα για έναν τεχνητό εγκέφαλο με αποτέλεσμα τη δημιουργία των τεχνητών νευρωνικών δικτύων. Τα τεχνητά νευρωνικά δίκτυα αναφέρονται σε μαθηματικά μοντέλα που προσπαθούν να μιμηθούν τη λειτουργία του δικτύου νευρώνων του εγκεφάλου. Πρόκειται για έναν ενεργά ερευνητικό κλάδο όπου μέσω συνεχούς έρευνας γίνονται συνεχώς νέες ανακαλύψεις.

Τα αποτελέσματα αυτής της έρευνας φαίνονται καθημερινά στη ζωή όλων καθώς οι διάφορες εφαρμογές των τεχνητών νευρωνικών δικτύων είναι πλέον μέρος της καθημερινότητάς μας. Οι εφαρμογές τους δεν σταματάνε στα μικρά καθημερινά προβλήματα που είναι ικανά αυτά τα δίκτυα να λύσουν καθώς η συμβολή τους σε άλλους κλάδους όπως η ιατρική μπορεί να χαρακτηριστεί επαναστατική. Μία τέτοια εφαρμογή στην οποία συμβάλουν είναι η εξίσωση καναλιού στον επιστημονικό κλάδο των τηλεπικοινωνιών.

Μεταξύ της δομής των εξισωτών καναλιού και των τεχνητών νευρωνικών δικτύων μπορούν να παρατηρηθούν ομοιότητες μέσα από τις οποίες δεν είναι δύσκολο να κατανοηθεί ο λόγος για τον οποίο τα τεχνητά νευρωνικά δίκτυα χρησιμοποιούνται ως εξισωτές με μεγάλη επιτυχία. Οι κατηγορίες των νευρωνικών σχετικά με την αρχιτεκτονική τους είναι δύο, τα δίκτυα πρόσθιας τροφοδότησης και τα δίκτυα ανατροφοδότησης.

Τα δίκτυα πρόσθιας τροφοδότησης παρουσιάζουν ομοιότητες με τους γραμμικούς εξισωτές ενώ τα δίκτυα ανατροφοδότησης παρουσιάζουν ομοιότητες με τους εξισωτές ανατροφοδότησης.

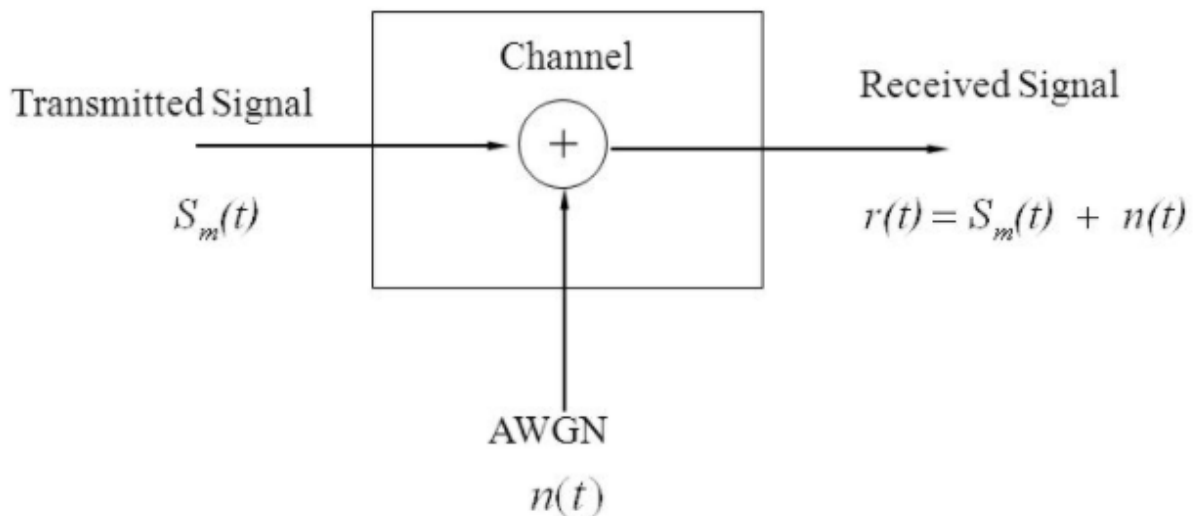
Τα τεχνητά νευρωνικά δίκτυα εξειδικεύονται σε μη γραμμικά προβλήματα λόγω των συναρτήσεων ενεργοποίησης που χρησιμοποιούν. Από την άλλη οι εξισωτές ανατροφοδότησης είναι το μόνο μοντέλο εξισωτή που είναι φτιαγμένο για να αντιμετωπίζει μη γραμμικά κανάλια. Οπότε μπορεί να κατανοήσει κανείς ότι τα τεχνητά νευρωνικά δίκτυα παρουσιάζουν μεγαλύτερες δυνατότητες για την εξίσωση ενός μη γραμμικού καναλιού. Πράγματι, έρευνες έδειξαν ότι ένας εξισωτής που βασίζεται σε ένα νευρωνικό δίκτυο έχει καλύτερη απόδοση από έναν κλασικό εξισωτή.

Η παρούσα διπλωματική εργασία είναι οργανωμένη σε τέσσερα κεφάλαια. Στο πρώτο κεφάλαιο αρχικά περιγράφονται κάποιες εισαγωγικές έννοιες, εξηγεί την έννοια της εξίσωσης καναλιού και περιγράφει κάποιες κλασικές τεχνικές εξίσωσης καναλιού. Το δεύτερο κεφάλαιο αναφέρεται στα τεχνητά νευρωνικά δίκτυα και αναλύει όλες τις βασικές τεχνικές λεπτομέρειες τους. Στο τρίτο κεφάλαιο αναλύονται κάποιες περιπτώσεις εξισωτών που είναι βασισμένοι σε τεχνητά νευρωνικά δίκτυα και συγκρίνεται η επίδοσή τους σε σχέση με τους κλασικούς εξισωτές. Στο τέταρτο και τελευταίο κεφάλαιο παρουσιάζεται το πειραματικό μέρος της διπλωματικής εργασίας στο οποίο εκτελέστηκαν κάποια πειράματα προκειμένου να φτιαχτεί ένας εξισωτής που βασίζεται σε τεχνητά νευρωνικά δίκτυα και καλείται να αντιμετωπίσει διαφορετικού είδους κανάλια, ενώ ταυτόχρονα δοκιμάστηκαν και κάποιες κλασικές μεθόδους εξίσωσης καναλιού. Τέλος, η εργασία ολοκληρώνεται με τα συμπεράσματα και τη βιβλιογραφία. Στα παραρτήματα στο τέλος της εργασίας παρατίθενται οι κώδικες που αναπτύχθηκαν και χρησιμοποιήθηκαν για την υλοποίηση των προσομοιώσεων του πειραματικού μέρους.

1.1 Βασικά είδη καναλιών

1.1.1 Κανάλι Προσθετικού Λευκού Γκαουσιανού Θορύβου (AWGN)

Το κανάλι AWGN είναι ένα από τα απλούστερα μαθηματικά μοντέλα όσον αφορά τη διάδοση ψηφιακής πληροφορίας μέσω ενός καναλιού επικοινωνίας. Αυτά τα κανάλια είναι αναλογικά κανάλια παρόλο που η πληροφορία που δέχονται δύναται να είναι σε ψηφιακή μορφή. Αυτό σημαίνει ότι έχουν την ανάγκη να απεικονίσουν την μεταδιδόμενη ψηφιακής μορφής πληροφορία σε αναλογικές κυματομορφές σήματος πληροφορίας. Ο στόχος λοιπόν για ένα τέτοιο κανάλι είναι να μπορέσει να απεικονίσει σωστά την προς μετάδοση πληροφορία σε αναλογικές κυματομορφές, έτσι ώστε όταν φτάσει η πληροφορία στο δέκτη, να μπορέσει επιτυχώς να τη μετατρέψει σε ψηφιακή μορφή, χωρίς να χάνει μεγάλο μέρος της πληροφορίας. Το ονομά του (AWGN κανάλι) το λαμβάνει λόγω της μαθηματικής συμπεριφοράς του να προσθέτει θόρυβο γκαουσιανής κατανομής στο σήμα εισόδου του. Η λέξη «λευκού» αναφέρεται στην ιδέα του ότι ο θόρυβος έχει ομοιόμορφη ισχύ σε όλο το φάσμα συχνοτήτων, όπως το χρώμα λευκό έχει παρόμοιες εκπομπές σε όλο το ορατό φάσμα (visible spectrum).



Σχήμα 1. 1: Απεικόνιση λαμβανόμενου σήματος στην έξοδο AWGN καναλιού (Proakis and Salehi 2002).

Η μαθηματική του συμπεριφορά περιγράφεται από τη σχέση:

$$r(t) = S_m(t) + n(t) \quad (1.1)$$

όπου $S_m(t)$ είναι το σήμα εισόδου και $n(t)$ είναι ο θόρυβος

1.1.2 AWGN κανάλι περιορισμένου εύρους ζώνης

Τα κανάλια AWGN μπορούν να μοντελοποιηθούν ως ένα γραμμικό φίλτρο με περιορισμό στο εύρος ζώνης του, όσον αφορά τη ψηφιακή μετάδοση σήματος μέσω ενός καναλιού. Παραδείγματα τέτοιων καναλιών που συναντούμε είναι τα τηλεφωνικά κανάλια, τα δορυφορικά κανάλια, τα υπόγεια κανάλια, τα ακουστικά κανάλια και τα ραδιοφωνικά κανάλια μικροκυμάτων οπτικής επαφής (line-of-sight, LOS). Λόγω του περιορισμού εύρους ζώνης που τίθεται από το κανάλι γραμμικού φίλτρου, τα μεταδιδόμενα σήματα πρέπει να έχουν τον κατάλληλο σχεδιασμό για την ικανοποίηση του περιορισμού αυτού. Λόγω του περιορισμού αποκλείεται η χρήση ορθογωνικών παλμών στην έξοδο του διαμορφωτή. Τα κανάλια γραμμικού φίλτρου προκαλούν παραμόρφωση στο μεταδιδόμενο σήμα και αυτή η παραμόρφωση με τη σειρά της καταλήγει σε παρεμβολή μεταξύ των συμβόλων (Intersymbol interference, ISI) στην έξοδο του αποδιαμορφωτή. Αυτό συνεπάγεται με αύξηση της πιθανότητας σφάλματος στον φωρατή. Οι μέθοδοι που αναλαμβάνουν την διόρθωση της παραμόρφωσης που προκαλεί το σφάλμα, ονομάζονται εξισωτές καναλιού (channel equalizer).

Το κανάλι χαρακτηρίζεται από ένα γραμμικό φίλτρο με κρουστική απόκριση $c(t)$ και απόκριση συχνότητας $C(f)$ όπου δίνεται από τη μαθηματική σχέση:

$$c(f) = \int_{-\infty}^{\infty} c(t) e^{-j2\pi ft} dt \quad (1.2)$$

Έστω ότι η είσοδος σε ένα κανάλι περιορισμένου εύρους ζώνης είναι μία κυματομορφή ενός σήματος $g_T(t)$ όπου ο δείκτης T υποδηλώνει την έξοδο του πομπού. Έτσι η απόκριση του καναλιού είναι η συνέλιξη του σήματος $g_T(t)$ με την κρουστική απόκριση του καναλιού $c(t)$, δηλαδή:

$$h(t) = \int_{-\infty}^{\infty} c(t) g_T(t - \tau) dt \quad (1.3)$$

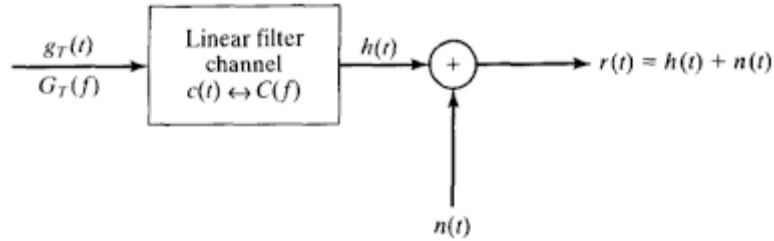
$$h(t) = c(t) * g_T(t) \quad (1.4)$$

Αν η παραπάνω σχέση εκφραστεί στο πεδίο της συχνότητας προκύπτει:

$$H(f) = C(f)G_T(f) \quad (1.5)$$

Όπου $g_T(f)$ είναι το φάσμα του σήματος $g_T(t)$ που προκύπτει μέσω μετασχηματισμού Fourier, $C(f)$ είναι η απόκριση συχνότητας του καναλιού και $H(f)$ είναι το φάσμα της εξόδου $h(t)$. Από της σχέση (1.5) είναι προφανής η παραμόρφωση του σήματος $g_T(t)$.

Σε συνδυασμό με αυτά που αναφέρθηκαν στην ενότητα για ένα απλό AWGN κανάλι, το σήμα που θα λάβει ο αποδιαμορφωτής στην είσοδο του θα έχει μολυνθεί από AWGN. Οπότε το σήμα αυτό θα είναι της μορφής $h(t)+n(t)$ όπου $n(t)$ είναι ο θόρυβος (AWGN). Αυτό φαίνεται στο παρακάτω σχήμα.

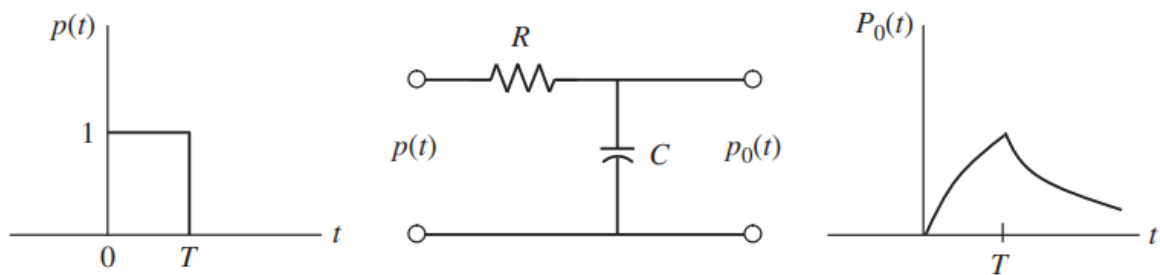


Σχήμα 1. 2: Κανάλι περιορισμένου εύρους ζώνης, μοντελοποιημένο ως γραμμικό φίλτρο (Proakis and Salehi 2002)

1.2 Διασυμβολική Παρεμβολή (Intersymbol Interference-ISI)

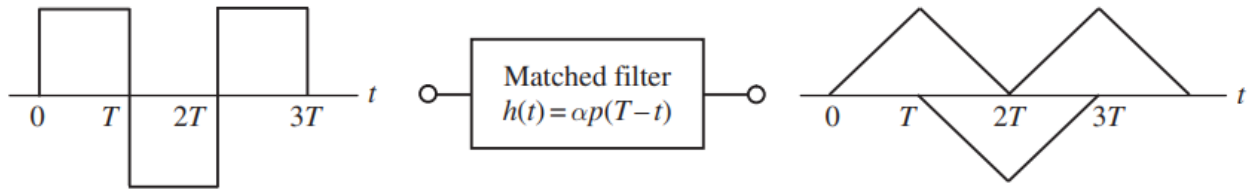
1.2.1 Ορισμός ISI

Στα κανάλια τύπου AWGN με θεωρητικά άπειρο εύρος ζώνης μπορεί να χρησιμοποιηθεί τετραγωνικός παλμός προκειμένου να απεικονισθεί ένα bit ή ένα σύμβολο. Στην πραγματικότητα όμως προκειμένου να διατηρηθεί το σύμβολο στην είσοδο του αποδιαμορφωτή χρειάζεται απλά το εύρος ζώνης του καναλιού να είναι πολύ μεγαλύτερο από αυτό της διαμόρφωσης. Αλλά αυτό δεν είναι κάτι εύκολο καθώς το εύρος ζώνης είναι σπάνιος πόρος οπότε σε πολλές περιπτώσεις τα συστήματα τηλεπικοινωνιών είναι περιορισμένου εύρους ζώνης. Επομένως χρειάζονται φίλτρα στον πομπό και στον δέκτη του συστήματος. Αν το εύρος ζώνης του καναλιού μέσα από το οποίο μεταδίδεται το σήμα είναι μικρότερο από αυτό του σήματος τότε προκύπτει παραμόρφωση των παλμών όπως φαίνεται στο σχήμα 1.3 που απεικονίζει έναν τετραγωνικό παλμό να περνάει μέσα από ένα RC φίλτρο. Σε αυτό το παράδειγμα προκαλείται διασυμβολική παρεμβολή (Intersymbol Interference-ISI) καθώς όπως φαίνεται η παραμορφωμένη άκρη του αρχικά τετραγωνικού παλμού θα επηρεάσει τον επόμενο τετραγωνικό παλμό που θα λάβει ο δέκτης. (Ha 2010)

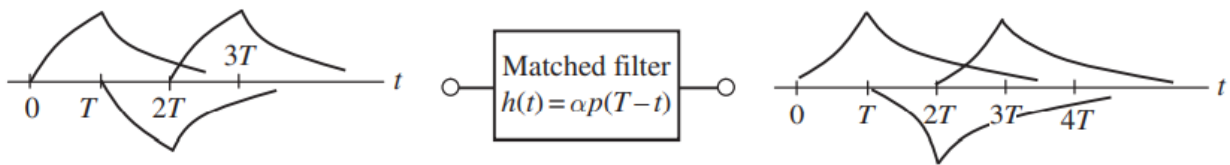


Σχήμα 1. 3: Παραμόρφωση τετραγωνικού παλμού μέσω ενός φίλτρου RC (Ha 2010).

Με την ίδια λογική στο επόμενο παράδειγμα φαίνεται πως διαμορφώνεται η έξοδος ενός προσαρμοσμένου φίλτρου (matched filter) το οποίο έχει ως στόχο την ενίσχυση του σηματοθορυβικού λόγου (Signal to Noise Ratio-SNR) με δύο διαφορετικές εισόδους, με τη μία είσοδο τον τετραγωνικό παλμό πριν περάσει μέσα από το φίλτρο RC και με τη δεύτερη τον ίδιο παλμό αφού περάσει μέσα από το φίλτρο RC. Έστω ότι τα μεταδιδόμενα bits είναι τα 1, 0, 1 με το 1 να αντιστοιχίζεται σε έναν ορθογώνιο παλμό $p(t)$ και το 0 στον αντίθετο παλμό $-p(t)$. Αυτό φαίνεται στο σχήμα 1.4 και 1.5 για τις δύο εισόδους αντίστοιχα.



Σχήμα 1. 4: Παράδειγμα με είσοδο τετραγωνικού παλμού (Ha 2010).

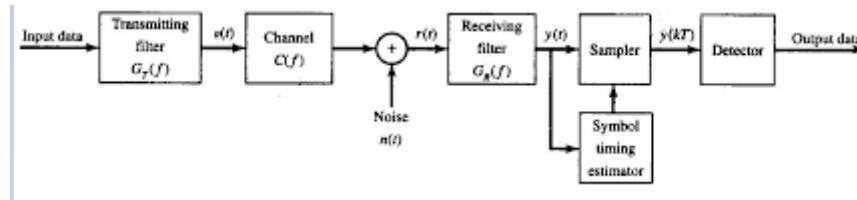


Σχήμα 1. 5: Παράδειγμα με είσοδο παραμορφωμένου τετραγωνικού παλμού (Ha 2010).

Όπως φαίνεται το φαινόμενο της διασυμβολικής παρέμβολης στα δύο παραδείγματα παρατηρείται μόνο στο παράδειγμα του σχήματος 1.5 τις χρονικές στιγμές T , $2T$ και $3T$ καθώς στην έξοδο του φίλτρου του παραδείγματος του σχήματος 1.4 οι τριγωνικοί πλέον παλμοί δεν επηρεάζουν ο ένας τον άλλον για τις ίδιες χρονικές στιγμές καθώς καμία άκρη των παλμών δεν έρχεται σε επαφή με άλλη. Το φιλτράρισμα ενός καναλιού είναι υπεύθυνο για την ISI που συμβαίνει πριν το σήμα περάσει μέσα από το προσαρμοσμένο φίλτρο (Matched filter) ενώ για την ISI που συμβαίνει μετά το φίλτρο όπως στην περίπτωση του σχήματος 1.5 ευθύνεται και το φίλτρο αλλά και το φιλτράρισμα του καναλιού. Όπως γίνεται κατανοητό το φίλτρο που χρησιμοποιείται για την καταπολέμηση της ISI δεν μπορεί να αναστρέψει την ISI που έχει δημιουργηθεί στο σήμα από το κανάλι, όπως φαίνεται στο σχήμα 1.5. Το φιλτράρισμα του καναλιού μετάδοσης είναι πάρα πολύ σημαντικό και στην παρακάτω ενότητα περιγράφεται το κριτήριο Nyquist για μηδενική διασυμβολική παρεμβολή. (Ha 2010)

1.2.2 Μαθηματική ερμηνεία και αποτροπή ISI

Στο σχήμα 1.6 φαίνεται ένα σύστημα επικοινωνίας διαμόρφωσης πλάτους παλμών (PAM). Το $G_T(f)$ απεικονίζει το φίλτρο μετάδοσης με κρουστική απόκριση $g_T(t)$, το $G_R(f)$ το φίλτρο λήψης με κρουστική απόκριση $g_R(t)$ ενώ το κανάλι είναι ένα γραμμικό φίλτρο με AWGN. Έπειτα από το φίλτρο λήψης φαίνεται ένας δειγματολήπτης (sampler) και ένας φωρατής συμβόλων (detector). Επίσης φαίνεται ένας εκτιμητής χρονισμού συμβόλων του οποίου η εργασία είναι να εκτιμάει τις κατάλληλες χρονικές στιγμές για τη σωστή δειγματοληψία που εκτελεί ο δειγματολήπτης στην έξοδο του φίλτρου λήψης (Proakis and Salehi 2002).



Σχήμα 1. 6: Μπλόκ διάγραμμα ψηφιακού PAM συστήματος (Proakis and Salehi 2002).

Έστω ότι στη συγκεκριμένη περίπτωση έχουμε μια μετάδοση M-αδικής PAM. Η συμπεριφορά του φίλτρου εκπομπής εκφράζεται από την παρακάτω μαθηματική σχέση

$$v(t) = \sum_{n=-\infty}^{\infty} a_n g_T(t - nT) \quad (1.6)$$

Όπου $T=k/R_b$, με το k να είναι ο αριθμός των bits που χρειάζονται για να εκφράσουν τα σύμβολα στα οποία χωρίζεται η ακολουθία δεδομένων εισόδου δυαδικής φύσης ενώ R_b είναι ο ρυθμός μετάδοσης με τον οποίο μεταφέρονται τα bits (bit rate). Το a_n καθορίζεται από τα k bits πληροφορίας και διαμορφώνει τον παλμό μετάδοσης $g_T(t)$.

Όπως φαίνεται και από το σχήμα, η έξοδος του καναλιού είναι και η είσοδος του φίλτρου λήψης και εκφράζεται μαθηματικά από την παρακάτω σχέση

$$r(t) = \sum_{n=-\infty}^{\infty} a_n h(t - nT) + n(t) \quad (1.7)$$

όπου $h(t)$ είναι η κρουστική απόκριση του φίλτρου εκπομπής και του καναλιού συνδυαστικά. Αυτό μαθηματικά εκφράζεται ως $h(t)=c(t)*g_T(t)$ με $c(t)$ να υποδηλώνει την κρουστική απόκριση του καναλιού. Η ποσότητα $n(t)$ εκφράζει τον προσθετικό θόρυβο.

Πάντα με οδηγό το σχήμα 1.6, το λαμβανόμενο από το φίλτρο σήμα θα περάσει μέσα από το γραμμικό φίλτρο λήψης με κρουστική απόκριση $g_R(t)$ και απόκριση συχνότητας $G_R(f)$. Εάν η κρουστική απόκριση $g_R(t)$ προσαρμοστεί σωστά στην κρουστική απόκριση $h(t)$ που είδαμε προηγουμένως, τότε ο σηματοθορυβικός λόγος (SNR) στην έξοδο του γίνεται μέγιστος ακριβώς την κατάλληλη στιγμή δειγματοληψίας καθώς η δειγματοληψία είναι το αμέσως επόμενο στάδιο βάση σχήματος.

Η έξοδος του φίλτρου λήψης εκφράζεται μαθηματικά από την παρακάτω σχέση

$$y(t) = \sum_{n=-\infty}^{\infty} a_n x(t - nT) + w(t) \quad (1.8)$$

Όπου $x(t) = h(t)*g_R(t) = g_T(t)*c(t)*g_R(t)$, και $w(t) = n(t)*g_R(t)$ που είναι ο προσθετικός θόρυβος.

Η συμπεριφορά στην έξοδο του δειγματολήπτη εκφράζεται από τη μαθηματική σχέση (1.9)

$$y(mT) = \sum_{n=-\infty}^{\infty} a_n x(mT - nT) + w(mT) \quad (1.9)$$

ή αλλιώς

$$y_m = \sum_{n=-\infty}^{\infty} a_n x_{m-n} + w_m = x_0 a_m + \sum_{n \neq m} a_n x_{m-n} + w_m \quad (1.10)$$

Όπου $x_m = x(mT)$, $w_m = w(mT)$ με $m = 0, \pm 1, \pm 2, \dots$

Στην εξίσωση (1.10) ο όρος $x_0 a_m$ στο δεξιό μέρος της ισότητας, υποδηλώνει το επιθυμητό σύμβολο a_m πολλαπλασιασμένο με την παράμετρο κέρδους x_0 .

Όλα αυτά που προαναφέρθηκαν πάνω στο συγκεκριμένο παράδειγμα του τηλεπικοινωνιακού συστήματος έγιναν με σκοπό να εξηγηθεί μαθηματικά η διασυμβολική παρεμβολή. Ο δεύτερος όρος στην εξίσωση (1.10) αναπαριστά τη διασυμβολική παρεμβολή (ISI) που είναι η επίδραση των άλλων συμβόλων στο επιθυμητό σύμβολο τη χρονική στιγμή δειγματοληψίας $t = mT$. Το w_m αναπαριστά τον προσθετικό θόρυβο.

Εν κατακλείδι η ISI είναι ένα φαινόμενο το οποίο χειροτερεύει την επίδοση του συστήματος και για αυτό είναι απαραίτητη η σωστή σχεδίαση φίλτρων εκπομπής και λήψης προκειμένου ο όρος x_n να μηδενιστεί και να μην υπάρχει ISI. Εάν αυτό επιτευχθεί τότε μπορούν να υπάρχουν σφάλματα μόνο λόγω του θορύβου.

1.2.3 Κριτήριο Nyquist για μηδενική ISI

Βασικό κριτήριο για να αποφευχθεί η παραμόρφωση καναλιού περιορισμένου εύρους ζώνης είναι ο σωστός σχεδιασμός του τηλεπικοινωνιακού συστήματος. Ένας παλμός ικανοποιεί τη συνθήκη μηδενικής διασυμβολικής παρεμβολής (ISI) όταν το φάσμα του τηρεί το κριτήριο Nyquist για μηδενική ISI.

$$x(nT) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \quad (1.11)$$

όπου για να ικανοποιείται η εξίσωση 1.11 πρέπει ο μετασχηματισμός Fourier του σήματος $X(f)$ να ικανοποιεί την εξίσωση 1.12

$$\frac{1}{T} \sum_{m=-\infty}^{m=\infty} X(f + \frac{m}{T}) = 1 \quad (1.12)$$

ή αλλιώς

$$\sum_{m=-\infty}^{m=\infty} X(f + \frac{m}{T}) = T \quad (1.13)$$

όπου $X(f)$ είναι ο μετασχηματισμός Fourier της κρουστικής απόκρισης $x(t)$ και T είναι η περίοδος συμβόλων. (Azurdia-Meza 2013)

1.3 Σχεδιασμός φίλτρων για γνωστό κανάλι

Από το κριτήριο Nyquist για μηδενική ISI βγαίνει το συμπέρασμα ότι για μία μετάδοση χωρίς ISI μέσω ενός καναλιού πρέπει να τηρείται η μαθηματική εξίσωση 1.14 για τα φίλτρα και το κανάλι.

$$G_t(f)C(f)G_R(f) = X_{rc}(f) \quad (1.14)$$

με $X_{rc}(f)$ να συμβολίζει το μετασχηματισμό Fourier του επιθυμητού ανυψωμένου συνημιτόνου με τις παραμέτρους του να εξαρτώνται από το εύρος ζώνης του καναλιού W και την περίοδο σηματοδοσίας T .

Ο σχεδιασμός των φίλτρων εκπομπής και λήψης διαφέρει αναλόγως με το κανάλι που έχουμε να αντιμετωπίσουμε. Σε αυτή την υποενότητα θα συζητηθεί ο σχεδιασμός του δεδομένου ότι το κανάλι έχει γνωστά χαρακτηριστικά, δηλαδή γνωστή απόκριση συχνότητας $C(f)$ και ο στόχος είναι να μεγιστοποιηθεί το SNR προκειμένου να μηδενιστεί η ISI στην έξοδο του φίλτρου λήψης (Proakis and Salehi 2002).

Η συνθήκη που θα πρέπει να ικανοποιηθεί για τη συνιστώσα του σήματος είναι:

$$G_t(f)C(f)G_R(f) = X_{rc}(f)e^{-j2\pi f t_0} \quad (1.15)$$

Όπου $X_{rc}(f)$ είναι το επιθυμητό φάσμα ανυψωμένου συνημιτόνου όπου δίνει μηδενική ISI κατά τις χρονικές στιγμές δειγματοληψίας ενώ t_0 είναι η χρονική καθυστέρηση που είναι απαραίτητη προκειμένου να μπορέσουν να υλοποιηθούν τα φίλτρα λήψης και εκπομπής.

Ο θόρυβος στην έξοδο του φίλτρου λήψης περιγράφεται από τη μαθηματική σχέση 1.16

$$w(t) = \int_{-\infty}^{\infty} n(t - \tau)g_R(\tau)d\tau \quad (1.16)$$

με $n(t)$ τη συνιστώσα θορύβου στην είσοδο του φίλτρου. Από τη στιγμή που ο θόρυβος $n(t)$ είναι γκαουσιανής κατανομής με μηδενική μέση τιμή, αυτό συνεπάγεται ότι $w(t)$ είναι επίσης γκαουσιανής κατανομής με μηδενική μέση τιμή και φασματική πυκνότητα ισχύος:

$$S_w(f) = S_n(f)|G_R(f)|^2 \quad (1.17)$$

Από την (1.17) φαίνεται πως η φασματική πυκνότητα ισχύος του $w(t)$ εξαρτάται από τη φασματική πυκνότητα ισχύος $S_n(f)$ του $n(t)$.

Το φίλτρο εκπομπής που θα σχεδιαστεί θα πρέπει να έχει απόκριση συχνότητας:

$$G_T(f) = \frac{\sqrt{X_{rc}(f)}}{C(f)} e^{-j2\pi f t_0} \quad (1.18)$$

με t_0 να είναι η κατάλληλη καθυστέρηση προκειμένου να είναι αιτιατό το φίλτρο εκπομπής.

Από τη σχέση 1.18 συνεπάγεται

$$C(f)G_T(f) = \sqrt{X_{rc}(f)} e^{-j2\pi f t_0} \quad (1.19)$$

Αναλόγως και το φίλτρο λήψης θα πρέπει να έχει ανάλογη συμπεριφορά. Οπότε η απόκριση συχνότητας του θα είναι όπως φαίνεται στη σχέση 1.20 προκειμένου να μην παραμορφώσει το σήμα που θα φτάσει στον φωρατή

$$G_R(f) = \sqrt{X_{rc}(f)} e^{-j2\pi f t_r} \quad (1.20)$$

με t_r να συμβολίζει την κατάλληλη καθυστέρηση στο φίλτρο λήψης.

Η παραπάνω διαδικασία σχεδίασης φίλτρων εκπομπής και λήψης έχει στόχο την εκμηδένιση της ISI κατά τις χρονικές στιγμές δειγματοληψίας

1.4 Εξίσωση καναλιού για μη γνωστά κανάλια

Όταν τα χαρακτηριστικά απόκρισης συχνότητας ενός καναλιού που καλείται κάποιος να αντιμετωπίσει είναι γνωστά από πριν, τότε ο σχεδιασμός φίλτρων εκπομπής και λήψης είναι μία γνωστή και προκαθορισμένη διαδικασία με στόχο την εκμηδένιση της ISI κατά τις χρονικές στιγμές δειγματοληψίας όπως φαίνεται παραπάνω.

Όταν όμως το κανάλι δεν είναι γνωστό και τα χαρακτηριστικά του μεταβάλλονται με τον χρόνο, η αντιμετώπιση τέτοιων καταστάσεων διαφέρει και στην πραγματική ζωή δεν είναι λίγες οι φορές που κάποιος καλείται να εξισώσει τέτοιου είδους κανάλια, που τα χαρακτηριστικά τους δεν είναι γνωστά ή μεταβάλλονται με τον χρόνο. Ένα παράδειγμα είναι η μετάδοση δεδομένων μέσω ενός τηλεφωνικού δικτύου επιλογής (dial-up telephone network), όπου κάθε φορά που καλούμε έναν αριθμό το κανάλι επικοινωνίας θα είναι διαφορετικό αλλά όταν γίνεται η σύνδεση το κανάλι θα είναι πλέον χρονικά αμετάβλητο για μία εκτεταμένη χρονική περίοδο. Τα κανάλια τα οποία είναι χρονικά μεταβαλλόμενα με τον χρόνο χαρακτηρίζονται από χρονικά μεταβαλλόμενη απόκριση συχνότητας. Παράδειγμα τέτοιων καναλιών είναι τα ραδιοκανάλια (Proakis and Salehi 2002).

Και τα δύο αυτά παραδείγματα καναλιών, είναι περιπτώσεις στις οποίες δεν μπορεί να εφαρμοστεί η ίδια μέθοδος σχεδίασης φίλτρων εκπομπής και λήψης που εφαρμόζεται σε κανάλια με γνωστά χαρακτηριστικά.

Οπότε γνωρίζοντας όλα αυτά το φίλτρο εκπομπής θα σχεδιαστεί με τέτοιο τρόπο ώστε να έχει απόκριση συχνότητας την τετραγωνική ρίζα ανυψωμένου συνημιτόνου όπως φαίνεται στη σχέση 1.21

$$G_T(f) = \begin{cases} \sqrt{X_{rc}(f)} e^{-j2\pi f t_0}, & |f| \leq W \\ 0, & |f| \geq W \end{cases} \quad (1.21)$$

Και αναλόγως το φίλτρο λήψης θα είναι σχεδιασμένο με τέτοιον τρόπο ώστε να είναι προσαρμοσμένο στο φίλτρο εκπομπής όπως φαίνεται στη σχέση 1.22.

$$|G_T(f)||G_R(f)| = X_{rc}(f) \quad (1.22)$$

Έτσι λοιπόν θεωρώντας ότι το σήμα περνάει μέσα από φίλτρο εκπομπής, το κανάλι και το φίλτρο λήψης, το σήμα θα έχει την εξής μορφή στην έξοδο του φίλτρου λήψης

$$y(t) = \sum_{n=-\infty}^{\infty} a_n x(t - nT) + w(t) \quad (1.23)$$

με $x(t) = g_T(t) * c(t) * g_R(t)$

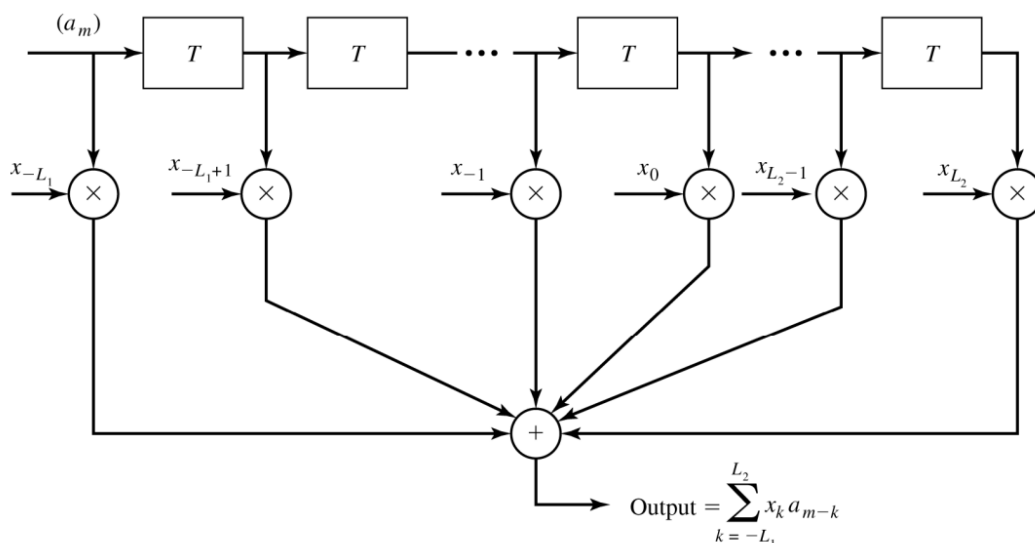
Έπειτα στην έξοδο του φίλτρου λήψης θα γίνει δειγματοληψία με περιοδικό τρόπο και θα παραχθεί η ακολουθία που φαίνεται στη σχέση 1.24

$$y_m = \sum_{n=-\infty}^{\infty} a_n x_{m-n} + w_m = x_0 a_m + \sum_{\substack{n=-\infty \\ n \neq m}}^{\infty} a_n x_{m-n} + w_m \quad (1.24)$$

Έχουμε χρησιμοποιήσει το συμβολισμό $x_n = x(nT)$ με το n να δηλώνει τις χρονικές στιγμές, δηλαδή $n = 0, n = \pm 1$ κτλ ενώ ο μεσαίος όρος στην εξίσωση είναι η ISI

Η ISI είναι ένα φαινόμενο που παρατηρείται στην έξοδο του φίλτρου λήψης και μπορεί να θεωρηθεί ότι συμβαίνει για συγκεκριμένο αριθμό συμβόλων, δηλαδή για έναν πεπερασμένο αριθμό συμβόλων. Με άλλα λόγια η ακολουθία δεδομένων a_m περνάει μέσα από ένα φίλτρο πεπερασμένης κρουστικής απόκρισης (finite impulse response-FIR). Μαθηματικά αυτό εκφράζεται ως $\{x_n, -L_1 \leq n \leq L_2\}$ με L_1 και L_2 να είναι πεπερασμένοι ακέραιοι αριθμοί και να υποδηλώνουν τα όρια για τα οποία υπάρχει ISI ενώ εκτός αυτών των ορίων $x_n = 0$.

Αυτό φαίνεται στο σχήμα 1.7 και ονομάζεται ισοδύναμο φίλτρο καναλιού διακριτού χρόνου. Η έξοδος του φίλτρου απαλλαγμένη από τον γκαουσιανό θόρυβο μπορεί να περιγραφεί από έναν αριθμό M^L όπου $L = L_1 + L_2$ (Proakis and Salehi 2002) καταστάσεων που είναι συνδεδεμένες μεταξύ τους.



Σχήμα 1. 7: Ισοδύναμο φίλτρο καναλιού διακριτού χρόνου (Proakis and Salehi 2002).

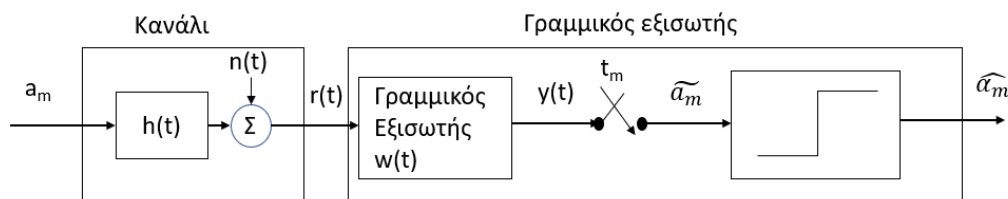
1.4.1 Γραμμικοί εξισωτές (Linear equalizers)

Οι γραμμικοί εξισωτές είναι μία από τις παλαιότερες τεχνικές για την αντιμετώπιση θορύβου και ISI και πρόκειται για απλά γραμμικά φίλτρα που εξαλείφουν την ISI.

Όπως γίνεται κατανοητό ο εξισωτής πρέπει να είναι ένα φίλτρο το οποίο έχει απόκριση συχνότητας η οποία είναι η αντίστροφη του καναλιού προκειμένου να εξαλειφθεί η ISI, αφού τα φίλτρα εκπομπής και λήψης είναι προσαρμοσμένα μεταξύ τους προκειμένου το γινόμενο τους να ισούται με το επιθυμητό φάσμα ανυψωμένου συνημιτόνου που δίνει μηδενική ISI. Η εξίσωση ενός τέτοιου απλού γραμμικού εξισωτή είναι όπως φαίνεται στην εξίσωση 1.25. (Hur et al. 2005)

$$G_E(f) = \frac{1}{C(f)} = \frac{1}{|C(f)|} e^{-j\theta_c(f)} \quad (1.25)$$

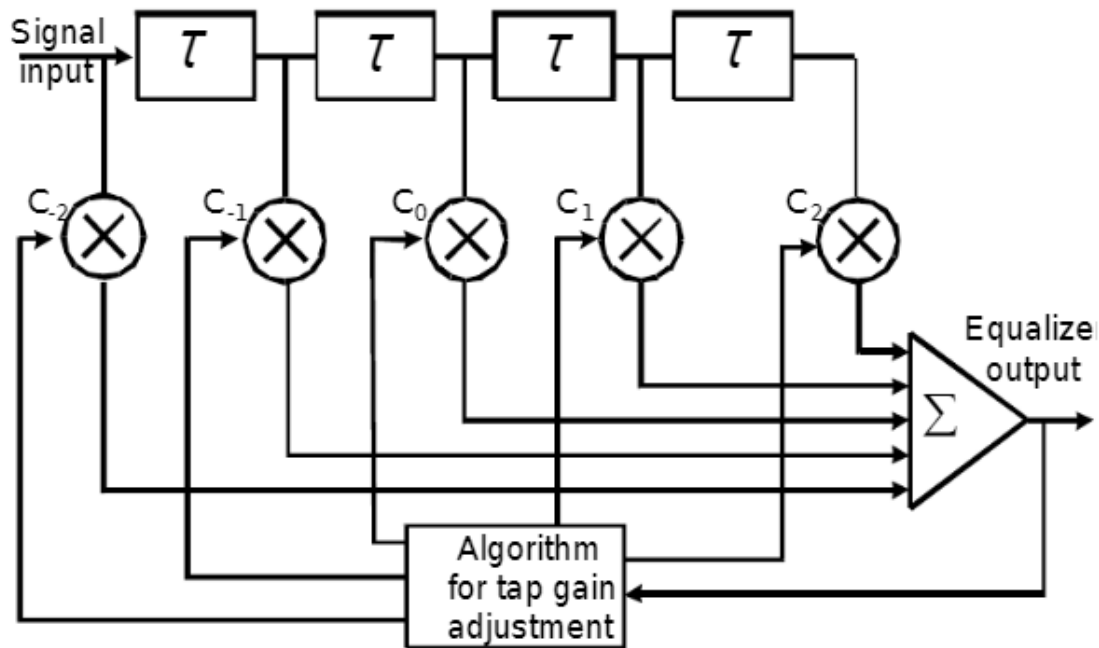
Όταν ο εξισωτής λειτουργεί ως ένα φίλτρο το οποίο έχει αυτή την συμπεριφορά, δηλαδή έχει ως αποτέλεσμα την εξάλειψη της ISI τις χρονικές στιγμές δειγματοληψίας $t=nT$, τότε αυτός ο εξισωτής ονομάζεται εξισωτής επιβολής μηδενισμών (Zero-forcing linear equalizer) και το σχήμα του φαίνεται στο σχήμα 1.8. Αφού λοιπόν η ISI έχει εξαλειφθεί τότε είναι λογικό να θεωρήσουμε ότι η είσοδος του φωρατή έχει τη μορφή $y_m = a_m + w_m$ όπου w_m ο θόρυβος. (Bergmans 1996)



Σχήμα 1. 8: Εξισωτής επιβολής μηδενισμών όπου δέχεται ένα σήμα που έχει περάσει από ένα κανάλι με AWGN $r(t)$ και η έξοδος του $y(t)$ αποτελεί την είσοδο του φωρατή που παίρνει περιοδικά δείγματα για να λάβει την απόφαση της εξόδου.

όπου \tilde{a}_m υποδηλώνει την πληροφορία που φτάνει στον φωρατή μετά την παραμόρφωση του θορύβου, δηλαδή το αντίστοιχο $y_m = a_m + w_m$ που αναφέρθηκε παραπάνω και \hat{a}_m τη πρόβλεψη του φωρατή. Ο διαχωριστής είναι μία βηματική συνάρτηση με την οποία ο φωρατής αποφασίζει αν η έξοδος θα είναι 0 ή 1 δεδομένου ότι η είσοδος είναι δυαδικής φύσης. Η ύπαρξη των φίλτρων που προαναφέρθηκαν δεν επηρεάζει σε κάτι εφόσον ακολουθούν τη συνθήκη που αναφέρθηκε αρχικά.

Από τη στιγμή που γνωρίζουμε ότι η ISI εμφανίζεται μόνο σε έναν πεπερασμένο αριθμό δειγμάτων L , ο εξισωτής μπορεί να χαρακτηριστεί ως ένα FIR φίλτρο, δηλαδή φίλτρο πεπερασμένης κρουστικής απόκρισης. Στο σχήμα 1.9 φαίνεται η δομή ενός τέτοιου εξισωτή. Ο εξισωτής εμπεριέχει καθυστερήσεις «tap» (tap delays), που είναι η καθυστέρηση του σήματος που δίνεται στην είσοδο και μπορεί να πάρουν μέχρι και την τιμή του διαστήματος μεταξύ των συμβόλων αλλά όχι μεγαλύτερη. Η καθυστερημένη μορφή του σήματος εισόδου έχει τη μορφή $x(t - k\tau)$ όπου $k = 1, 2, 3, \dots, n$ και $\tau \leq T$ με το T να δηλώνει το διάστημα μεταξύ των συμβόλων. Όταν $\tau < T$ τότε ο εξισωτής ονομάζεται κλασματικά διαστημοθετημένος εξισωτής (fractional tap-spaced equalizer) και το πλεονέκτημα του είναι ότι είναι ικανός να μειώσει το πρόβλημα της αλλοίωσης σήματος που εμφανίζεται στους εξισωτές που είναι διαστημοθετημένοι με βάση το διάστημα των συμβόλων τους (symbol spaced equalizers), δηλαδή $\tau = T$. (Hur et al. 2005)



Σχήμα 1. 9: Γραμμικός FIR εξισωτής (Hur et al. 2005)

Στο σχήμα 1.9 φαίνονται τα c_n όπου είναι οι συντελεστές των απομαστευτών (tap coefficients) και ο αλγόριθμος ο οποίος χρησιμοποιείται για την προσαρμογή των συντελεστών.

Η κρουστική απόκριση που χαρακτηρίζει τον FIR εξισωτή είναι:

$$g_E(t) = \sum_{n=-N}^N c_n \delta(t - n\tau) \quad (1.26)$$

ενώ η απόκριση συχνότητας

$$G_E(f) = \sum_{n=-N}^N c_n e^{-jw\pi f n\tau} \quad (1.27)$$

Το N έχει επιλεγεί με τέτοιο τρόπο ώστε $2N+1 \geq L$ προκειμένου ο εξισωτής να μπορέσει να καλύψει το μήκος L στο οποίο βρίσκεται η ISI. $2N+1$ είναι ο αριθμός συντελεστών tap c_n του εξισωτή.

Το σήμα που έχει εξισωθεί και συναντάμε στην έξοδο του εξισωτή είναι

$$q(t) = \sum_{n=-N}^N c_n x(t - n\tau) \quad (1.28)$$

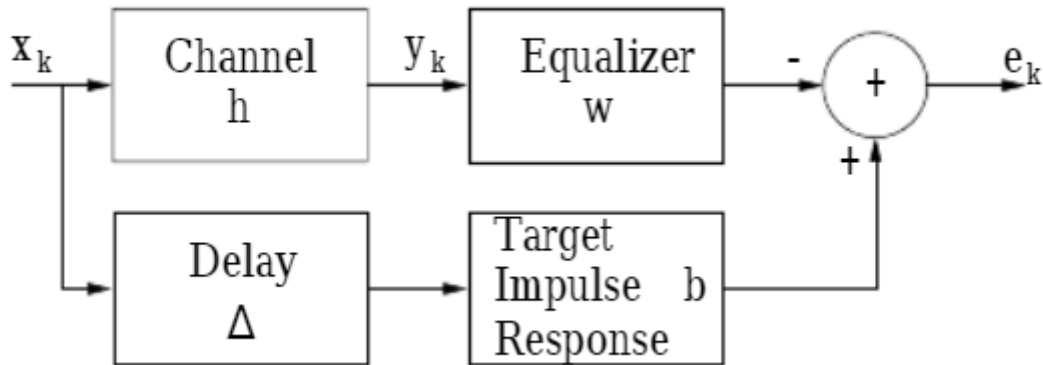
Προκειμένου να εξαλείψουμε την ISI στις χρονικές στιγμές δειγματοληψίας όπως αναφέρθηκε παραπάνω αντικαθιστούμε $t = mT$ (παραπάνω αναφέρθηκε ως $t = nT$).

$$q(mT) = \sum_{n=-N}^N c_n x(mT - n\tau) = \begin{cases} 1, & m = 0 \\ 0, & m = \pm 1, \pm 2, \dots, \pm N \end{cases} \quad (1.29)$$

Έτσι λοιπόν μετά την αντικατάσταση μπορούν να επιβληθούν οι συνθήκες που δίνονται στην εξίσωση 1.29 και ο λόγος για τον οποίο μπορούμε να βάλουμε τιμές για αυτές τις συνθήκες του m , είναι επειδή οι συντελεστές c_n του εξισωτή είναι $2N+1$. (Proakis and Salehi 2002)

Ο εξισωτής επιβολής μηδενισμών είναι ένας εξισωτής που χρησιμοποιείται κυρίως για κανάλια τα οποία έχουν υψηλό SNR και το κυρίως πρόβλημα είναι η ISI. Δεν είναι όμως καλός εξισωτής για κανάλια τα οποία έχουν αρκετό θόρυβο γιατί ο εξισωτής ZF (zero forcing) έχει ως αποτέλεσμα την αύξηση του θορύβου αυτού. Σε αυτές τις περιπτώσεις λοιπόν χρησιμοποιείται ο εξισωτής του ελαχίστου μέσου τετραγωνικού σφάλματος (Minimum mean square error-MMSE). Ο MMSE δεν είναι τόσο αποδοτικός όσο ο ZF σε ότι αφορά τη μείωση της ISI αλλά καταφέρνει να μειώσει το πρόβλημα του θορύβου (Waliullah et al. 2020).

Η λειτουργία του είναι βασισμένη στο σφάλμα μεταξύ της μεταδιδόμενης πληροφορίας και των δειγμάτων στην έξοδο και μέσω αυτού του σφάλματος προσπαθεί να υπολογίσει τις σωστές τιμές για τους συντελεστές του εξισωτή προκειμένου η πραγματική κρουστική απόκριση του εξισωτή να είναι ίση με την επιθυμητή κρουστική απόκριση με κάποια χρονική καθυστέρηση. Η δομή του φαίνεται στο σχήμα 1.10. (Arslan, Evans, and Kiaei 2000)

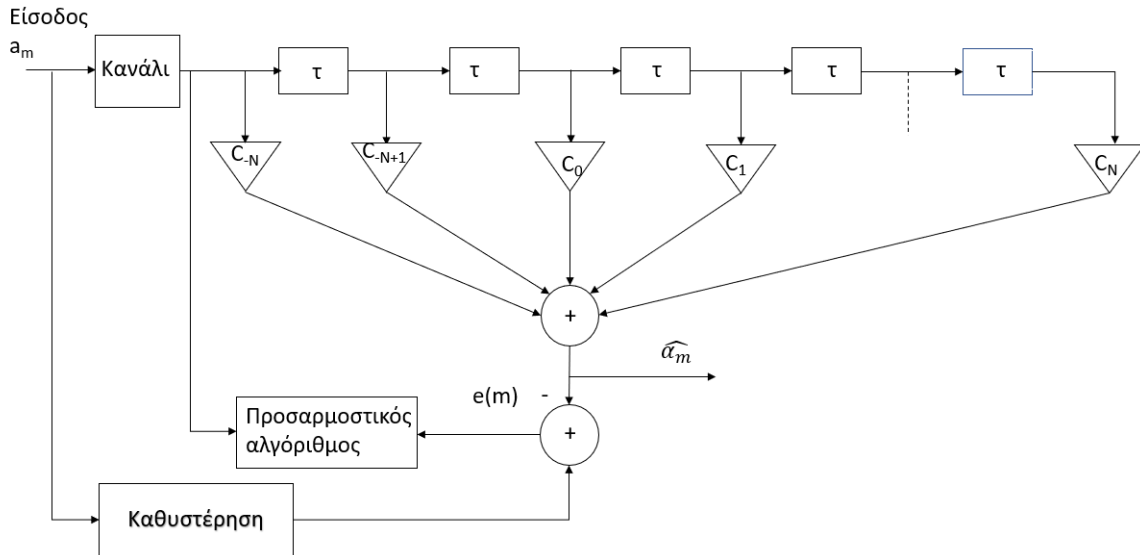


Σχήμα 1. 10: Εξισωτής MMSE (Arslan, Evans, and Kiaei 2000).

1.4.2 Προσαρμοστικοί εξισωτές

Υπάρχουν περιπτώσεις στις οποίες τα χαρακτηριστικά του καναλιού τηλεπικοινωνίας αλλάζουν με τον χρόνο και έτσι απαιτείται προσαρμοστικός εξισωτής (adaptive equalizer) γιατί και η ISI αλλάζει με τον χρόνο αφού αλλάζουν και τα χαρακτηριστικά του καναλιού. Έτσι οι συντελεστές c του εξισωτή θα πρέπει να προσαρμόζονται, να αλλάζουν δηλαδή τις τιμές τους συνεχώς προκειμένου να εξαλείφουν την ISI και για αυτό η διαδικασία ονομάζεται προσαρμοστική. Υπάρχουν προσαρμοστικοί εξισωτές γραμμικής και μη γραμμικής φύσης. Οι γραμμικοί προσαρμοστικοί εξισωτές δεν χρησιμοποιούν ανατροφοδότηση, σε αντίθεση με αυτούς που καλούνται να λύσουν μη γραμμικά προβλήματα εξίσωσης καναλιού. Τέτοιοι μη γραμμικοί εξισωτές χρησιμοποιούνται πολύ στις ασύρματες τηλεπικοινωνίες. Ο πιο γνωστός αλγόριθμος για την εκπαίδευση τέτοιων εξισωτών ονομάζεται αλγόριθμος ελαχίστων τετραγώνων (Least mean square -LMS) και είναι ένας αλγόριθμος στοχαστικής κλίσης. Πρόκειται για έναν απλό αλγόριθμο που δεν χρειάζεται μεγάλη υπολογιστική ισχύ και προσφέρει ικανοποιητικά αποτελέσματα, για αυτό το λόγο και χρησιμοποιείται. Λειτουργεί υπολογίζοντας το σφάλμα μεταξύ επιθυμητής εξόδου και πραγματικής εξόδου και μέσω του σφάλματος αυτού αλλάζει τις τιμές των συντελεστών του εξισωτή. Υπάρχει άλλος ένας αλγόριθμος που χρησιμοποιείται για την εκπαίδευση ενός τέτοιου εξισωτή και αυτός είναι ο αναδρομικός αλγόριθμος ελαχίστων τετραγώνων (Recursive least square -RLS) ο οποίος χρειάζεται μεγαλύτερη υπολογιστική ισχύ. Πρόκειται για έναν αλγόριθμο ντετερμινιστικό (deterministic) και όχι στοχαστικό όπως ο LMS. Οι αλγόριθμοι MSE και ZF που αναφέρθηκαν παραπάνω μπορούν να χρησιμοποιηθούν και εδώ.

Στο σχήμα 1.11 φαίνεται το μπλοκ διάγραμμα ενός προσαρμοστικού εξισωτή όπου $e(m)$ συμβολίζει το σφάλμα, τ την καθυστέρηση και C_N τους συντελεστές του εξισωτή (Atarattu et al. 2012).



Σχήμα 1. 11: Μπλοκ διάγραμμα ενός προσαρμοστικού εξισωτή.

Όπως φαίνεται από το σχήμα 1.11 το σφάλμα είναι αποτέλεσμα της διαφοράς της επιθυμητής και της πραγματικής εξόδου. Μαθηματικά αυτό φαίνεται στη σχέση 1.30

$$e(m) = a_m - \hat{a}_m \quad (1.30)$$

Όπου $e(m)$ το σφάλμα, a_m η επιθυμητή έξοδος, και \hat{a}_m η πραγματική έξοδος του εξισωτή.

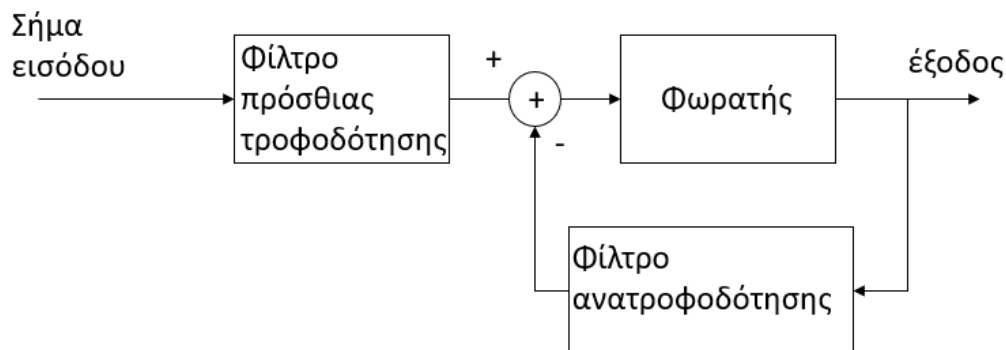
Ο προσαρμοστικός εξισωτής έχει δύο τρόπους με τους οποίους λειτουργεί, αρχικά εκπαιδεύεται με μία ψευδοτυχαία ακολουθία εισόδου και δίνει τιμές στους συντελεστές του για τη συγκεκριμένη ακολουθία, αυτό ονομάζεται λειτουργία εκπαίδευσης (training mode). Έπειτα ο εξισωτής αλλάζει λειτουργία και από αυτή της απόφασης, πηγαίνει σε λειτουργία ιχνηλάτησης (tracking mode) χρησιμοποιώντας έναν προσαρμοστικό αλγόριθμο προκειμένου να αλλάξει τις τιμές των συντελεστών του εξισωτή όπου χρειαστεί. (Atarattu et al. 2012)

1.4.3 Εξισωτής ανατροφοδότησης απόφασης

Πρόκειται για έναν εξισωτή μη γραμμικής φύσης που έχει καλύτερη απόδοση σε σχέση με αυτή ενός γραμμικού εξισωτή ενώ ταυτόχρονα δεν χρειάζεται τόσο μεγάλη υπολογιστική ισχύ και πολυπλοκότητα όπως ο εξισωτής MLSE (Maximum likelihood sequence estimation). Ο εξισωτής MLSE είναι ο καλύτερος που υπάρχει σε θέμα επίδοσης, αλλά λόγω της μεγάλης υπολογιστικής πολυπλοκότητας, δεν είναι πολύ βολική η χρήση του. Οπότε για αυτό το λόγο δημιουργήθηκε ο εξισωτής ανατροφοδότησης απόφασης (decision feedback equalizer -DFE) που προσφέρει περισσότερες δυνατότητες στην απόδοση από έναν γραμμικό εξισωτή και ταυτόχρονα δεν έχει την πολυπλοκότητα του MLSE.

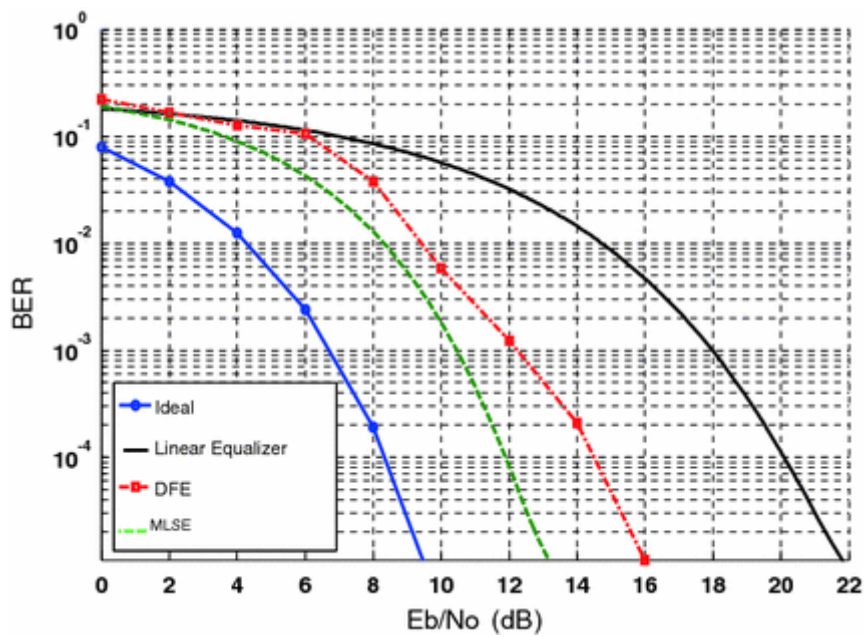
Στο σχήμα 1.12 φαίνεται η δομή ενός DFE εξισωτή ο οποίος αποτελείται από ένα φίλτρο πρόσθιας τροφοδότησης (feed-forward filter -FFF) όπου λαμβάνει το σήμα στην είσοδο του και η έξοδος του δειγματοληπτείται περιοδικά. Περιοδικά λειτουργεί και ο φωρατής ο οποίος στο σχήμα 1.12 έχει την ονομασία «Threshold» γιατί ουσιαστικά ο φωρατής λαμβάνει απόφαση με βάση κάποιο όριο που θέτει για το αν μεταδόθηκε το σύμβολο 0 ή 1.

Πρίν λάβει απόφαση για το ποιο σύμβολο έλαβε, το φίλτρο ανατροφοδότησης (Feedback Filter -FBB) συνδέει γραμμικά τα προηγούμενα σύμβολα για τα οποία πάρθηκαν αποφάσεις με το σύμβολο για το οποίο θα παρθεί απόφαση και μέσω αυτής της διαδικασίας εξαλείφεται η ISI. (Belfiore and Park 1979)



Σχήμα 1. 12: Βασική δομή ενός DFE εξισωτή

Στο σχήμα 1.13 φαίνεται καθαρά η διαφορά επίδοσης του DFE (κόκκινη γραμμή) σε σχέση με έναν γραμμικό εξισωτή (μαύρη γραμμή) καθώς και ότι ο MLSE (πράσινη γραμμή) είναι καλύτερος και από τους δύο και είναι αυτός που πλησιάζει πιο κοντά την ιδανική εξίσωση καναλιού. Όπως επίσης φαίνεται, η διαφορά του MLSE με τον DFE είναι μικρότερη από τη διαφορά του γραμμικού εξισωτή με τον DFE.



Σχήμα 1. 13: Επίδοση διαφορετικών μεθόδων εξίσωσης καναλιού. (Atef and Zimmermann 2013)

ΚΕΦΑΛΑΙΟ 2^ο : Τεχνητά Νευρωνικά Δίκτυα

2.1 Γενικά

2.1.1 Ορισμός

Τα τεχνητά νευρωνικά δίκτυα έχουν βασιστεί πάνω σε μαθηματικά μοντέλα τα οποία βρίσκονται σε πλήρη αντιστοιχία με τα βιολογικά νευρωνικά δίκτυα. Αποτελούνται από ένα σύνολο επεξεργαστικών μονάδων που αποκαλούνται κόμβοι ή αλλιώς νευρώνες, οι οποίοι είναι ομαδοποιημένοι σε στρώματα. Οι συνδέσεις μεταξύ των κόμβων ονομάζονται συνάψεις. Επιπλέον χαρακτηριστικά είναι το «bias» και η συνάρτηση ενεργοποίησης. Οι κόμβοι οι οποίοι δεν συνδέονται με κανέναν προηγούμενο κόμβο ονομάζονται είσοδοι ενώ αυτοί που δεν συνδέονται με κανέναν επόμενο κόμβο ονομάζονται έξοδοι. Τα ΤΝΔ επεξεργάζονται πληροφορίες όταν λάβουν εξωτερικά ερεθίσματα (γνωστά ως είσοδοι). Το ΤΝΔ αποτελείται από ένα σύνολο τεχνητών νευρώνων και ο κάθε νευρώνας του μπορεί να λάβει πολλαπλές εισόδους x_i και να παράγει μία μόνο έξοδο y . (Müller, Reinhardt, and Strickland 1995)

2.1.2 Δομή Νευρωνικού Δικτύου

Τα νευρωνικά δίκτυα αποτελούνται από 3 ειδών στρώματα (layers):

- **Επίπεδο εισόδου (Input layer)** όπου εισάγονται στο ΤΝΔ οι τιμές εισόδων από τον χρήστη.
- **Κρυφά επίπεδα (Hidden Layers)** όπου το ΤΝΔ επεξεργάζεται τα δεδομένα προκειμένου να φτάσει σε μία πρόβλεψη. Τα κρυφά στρώματα μπορούν να είναι πολλαπλά ή και μηδέν.
- **Επίπεδο εξόδου (Output Layer)** όπου το ΤΝΔ δίνει την πρόβλεψη του.

Τα επίπεδα του δικτύου αποτελούνται από έναν αριθμό κόμβων ορισμένο από τον δημιουργό του δικτύου. Αυτοί οι κόμβοι είναι συνδεδεμένοι μεταξύ τους με τέτοιο τρόπο ώστε ο κάθε κόμβος να συνδέεται με πολλαπλούς κόμβους του επόμενου στρώματος ή και με κόμβους του ίδιου στρώματος. Λόγω αυτής της συνδεσιμότητας ο κάθε κόμβος επηρεάζει έναν άλλον κόμβο με τέτοιο τρόπο ώστε να μπορέσει το ΤΝΔ να φτάσει στην επιθυμητή τελική πρόβλεψη. Τα ΤΝΔ είναι γνωστά για την ικανότητα τους να λύσουν μη γραμμικά προβλήματα. Σε κάθε κόμβο λοιπόν εφαρμόζεται η συνάρτηση ενεργοποίησης (activation function) η οποία είναι αυτή που δίνει στο δίκτυο μη γραμμική συμπεριφορά.

Ο δημιουργός του ΤΝΔ πρέπει κατά της δημιουργία του να ορίσει τα παρακάτω στοιχεία:

- Αριθμός κρυφών επιπέδων
- Αριθμός κόμβων (νευρώνων) ανά επίπεδο
- Συνάρτηση ενεργοποίησης (activation function)
- Τρόπο σύνδεσης των κόμβων μεταξύ τους
- Αλγόριθμος για τη διαδικασία εκπαίδευσης

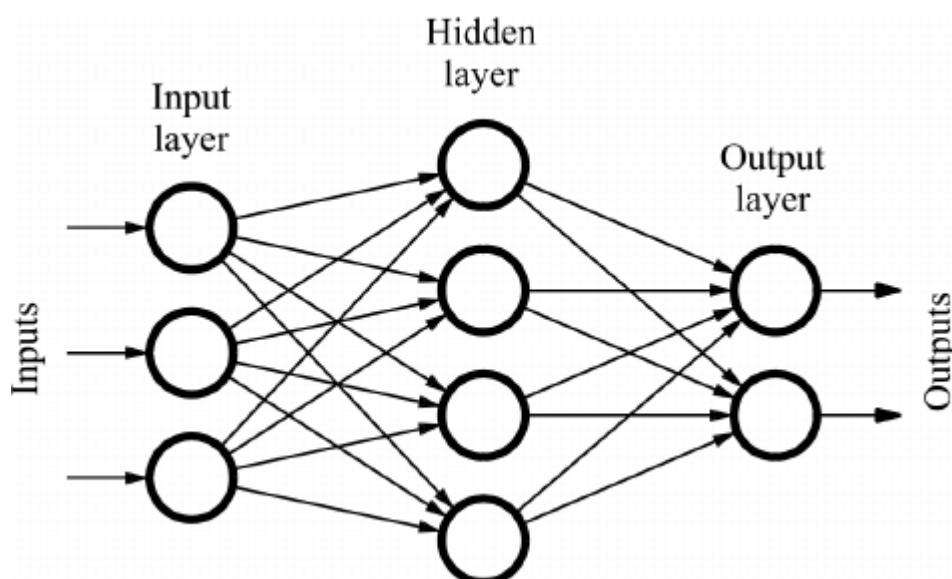
2.2 Αρχιτεκτονική Νευρωνικού Δικτύου

2.2.1 Πρόσθια τροφοδότηση και ανατροφοδότηση

Υπάρχουν δύο βασικοί τύποι ΤΝΔ που αναφέρονται στον τρόπο με τον οποίο είναι συνδεδεμένες οι μονάδες ενός νευρωνικού δικτύου.

- Πρόσθιας τροφοδότησης (feed forward)
- Ανατροφοδότησης (feedback)

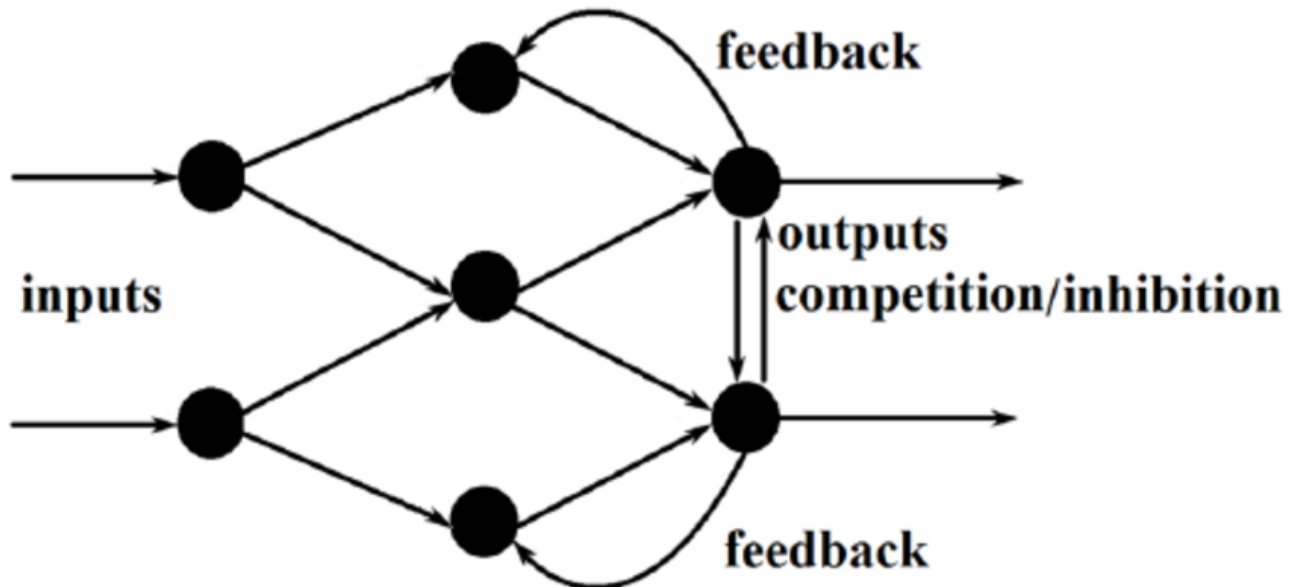
Το feedforward νευρωνικό δίκτυο είναι ο πρώτος τύπος νευρωνικού δικτύου που επινοήθηκε. Το όνομα του «Feedforward» το έχει πάρει για τον λόγο ότι όλες οι πληροφορίες μέσα σε αυτό μπορούν να κινηθούν μόνο προς μία κατεύθυνση, προς τα μπροστά (forward) όπως φαίνεται στο σχήμα 2.1. Οι πληροφορίες προέρχονται από τις εισόδους του νευρωνικού, συνεχίζουν την πορεία τους στα κρυφά στρώματα και από εκεί στην έξοδο του δικτύου, σε αντίθεση με άλλου τύπου νευρωνικά δίκτυα που μπορεί να έχουν μέσα βρόχους εδώ οι πληροφορίες κατευθύνονται μόνο ως προς μία κατεύθυνση (Colombaroni, Fusco, and Isaenko 2021). Δηλαδή δεν υπάρχει έξοδος μονάδας που να τροφοδοτεί μία μονάδα που βρίσκεται στο ίδιο ή σε προηγούμενο επίπεδο. Το feedforward αποτελεί την πιο κοινή αρχιτεκτονική νευρωνικού δικτύου. Ένα παράδειγμα είναι το γνωστό single layer perceptron που είναι ένα δίκτυο το οποίο αποτελείται από είσοδο η οποία τροφοδοτείται κατευθείαν στην έξοδο. Έτσι και το multi layer perceptron (MLP) που βασίζεται στο perceptron και είναι η πιο χρησιμοποιημένη εφαρμογή των feed forward ΤΝΔ κατατάσσεται και αυτό σε αυτήν την κατηγορία. (Tian et al. 2021)



Σχήμα 2. 1: Νευρωνικό δίκτυο πρόσθιας τροφοδότησης (Quiza and Davim 2011)

Τα ΤΝΔ ανατροφοδότησης ή αλλιώς αναδρομικά (Recurrent) ΤΝΔ διαφέρουν με τα feedforward στο ότι έχουν τουλάχιστον έναν νευρώνα ο οποίος μπορεί να στείλει πληροφορίες σε νευρώνα προηγούμενο επιπέδου ή και ίδιου όπως φαίνεται στο σχήμα 2.2 (Stergiou and Siganos 2016). Αυτό όμως δεν σημαίνει ότι κάθε νευρώνας έχει αυτή τη δυνατότητα αλλά ούτε ότι σε κάθε επίπεδο πρέπει να υπάρχει ένας τέτοιος νευρώνας. Ένας τέτοιος νευρώνας ονομάζονται «feedback loop» (Person and Khodakhah 2016). Πρόκειται για δίκτυα τα οποία είναι πολύ ισχυρά και μπορεί να φτάσουν πάρα πολύ υψηλά επίπεδα πολυπλοκότητας.

Τα αναδρομικά ΤΝΔ είναι δυναμικής φύσης αφού η κατάσταση τους αλλάζει συνεχώς έως ότου να φτάσουν στο σημείο ισορροπίας, το οποίο θα αλλάξει πάλι όταν λάβουν νέες εισόδους (Stergiou and Siganos 2016).



Σχήμα 2. 2: Νευρωνικό δίκτυο ανατροφοδότησης (Borah, Sarma, and Talukdar 2015)

2.2.2 Συναρτήσεις ενεργοποίησης

Τα ΤΝΔ έχουν συνήθως μία σταθερή συνάρτηση ενεργοποίησης (activation function) με μη γραμμική συμπεριφορά σε κάθε νευρώνα (Agostinelli et al. 2015). Ένα ΤΝΔ χωρίς συνάρτηση ενεργοποίησης έχει τη συμπεριφορά ενός μοντέλου γραμμικής παλινδρόμησης, όπου δεν έχει την ίδια ισχύ και απόδοση που έχει ένα ΤΝΔ που μπορεί να λύσει μη γραμμικά προβλήματα .

Οι πιο χρησιμοποιημένες συναρτήσεις ενεργοποίησης είναι:

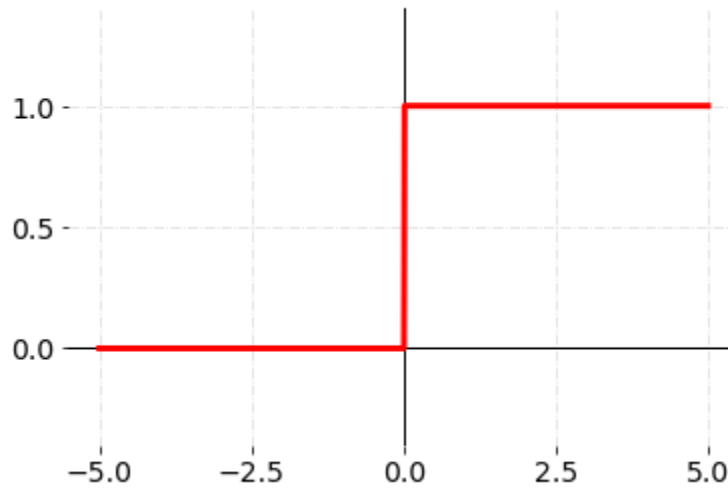
- Δυαδική βηματική συνάρτηση (binary step function)
- Γραμμική συνάρτηση (linear function)
- Σιγμοειδής συνάρτηση (sigmoid function)
- Υπερβολική εφαπτομένη συνάρτηση (tanh function)
- Ανορθωτική γραμμική συνάρτηση (ReLU function)

Δυαδική βηματική συνάρτηση

Μία σημαντική λειτουργία όσον αφορά τις συναρτήσεις ενεργοποίησης είναι ένας ταξινομητής που δρα με βάση ένα κατώφλι. Η συνάρτηση αυτή θα αποφασίσει αν θα ενεργοποιηθεί ο επόμενος νευρώνας ανάλογα με την τιμή που έχει λάβει στην είσοδο. Αν αποφασίσει ότι ο επόμενος νευρώνας δεν θα ενεργοποιηθεί, τότε απλά η έξοδος της δεν θα περάσει στον επόμενο νευρώνα ως την είσοδό του. Πρόκειται για την πιο απλή συνάρτηση ενεργοποίησης που μπορεί να χρησιμοποιηθεί σε ένα ΤΝΔ και μπορεί να χρησιμοποιηθεί μόνο για δυαδική ταξινόμηση και όχι για παραπάνω από δύο κλάσεις (Sharma et al. 2020).

Μαθηματικά ορίζεται ως:

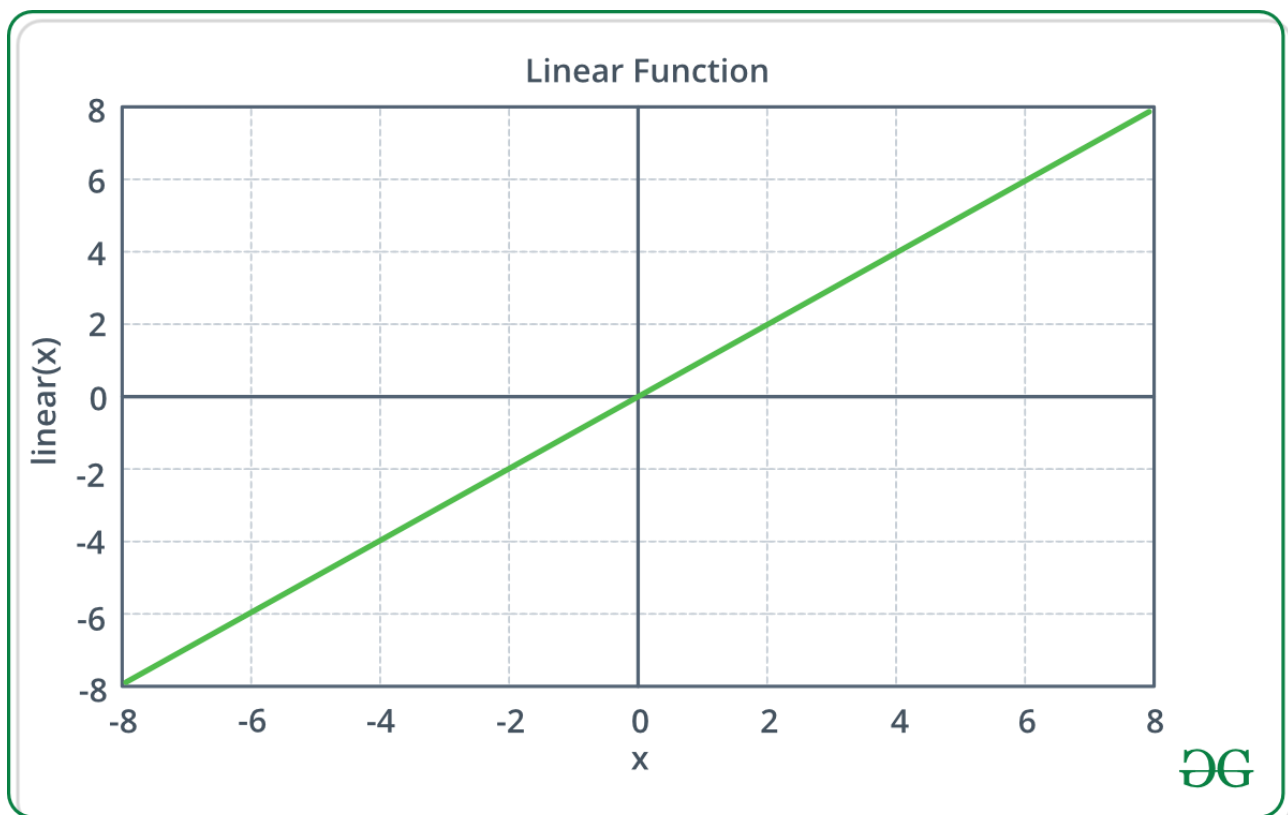
$$f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.1)$$



Σχήμα 2. 3: Δυαδική βηματική συνάρτηση (Siebenrock [2018] 2021)

Γραμμική συνάρτηση

Η γραμμική συνάρτηση δίνει έξοδο ανάλογη της εισόδου, δηλαδή είναι γραμμικά εξαρτημένες όπως φαίνεται στο σχήμα 2.4. Η συνάρτηση είναι η γνωστή $f(x)=ax$ όπου το a ορίζεται από τον χρήστη και είναι μια σταθερή τιμή που δηλώνει την κλίση της γραμμικής συνάρτησης.



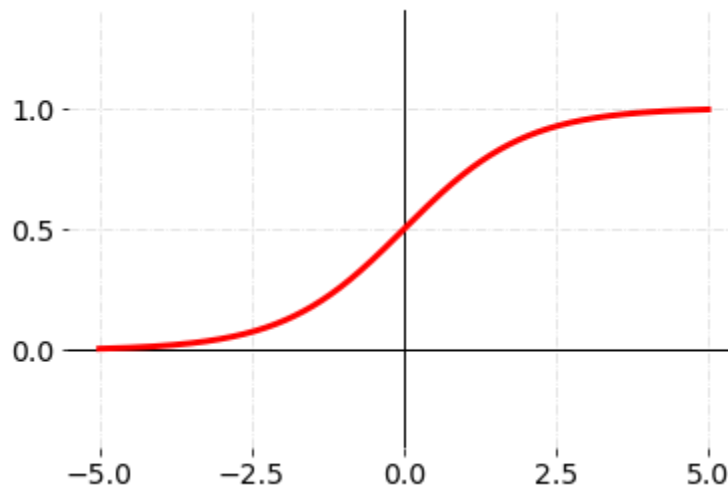
Σχήμα 2. 4: Γραμμική συνάρτηση (“Understanding Activation Functions in Depth” 2019).

Σιγμοειδής συνάρτηση

Πρόκειται για την πιο χρησιμοποιημένη συνάρτηση ενεργοποίησης στα ΤΝΔ καθώς είναι ικανή να δώσει μη γραμμική συμπεριφορά στο ΤΝΔ. Η συγκεκριμένη συνάρτηση μετατρέπει όλες τις τιμές που δέχεται σε τιμές στο διάστημα (0, 1). Όπως φαίνεται στο σχήμα 2.5 οι τιμές του Y αλλάζουν πολύ απότομα για οποιαδήποτε αλλαγή στις τιμές του X στο διάστημα -2 με 1. (Szandała 2021) .

Η μαθηματική σχέση που την περιγράφει είναι:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.2)$$



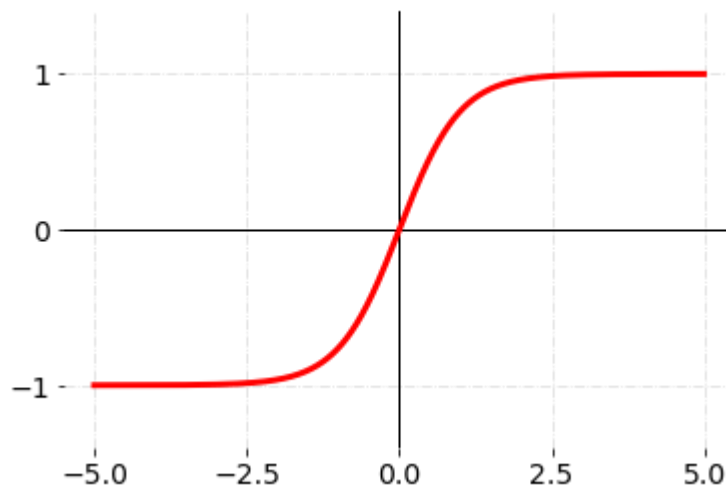
Σχήμα 2. 5: Σιγμοειδής συνάρτηση(Siebenrock [2018] 2021).

Υπερβολική εφαπτομένη συνάρτηση

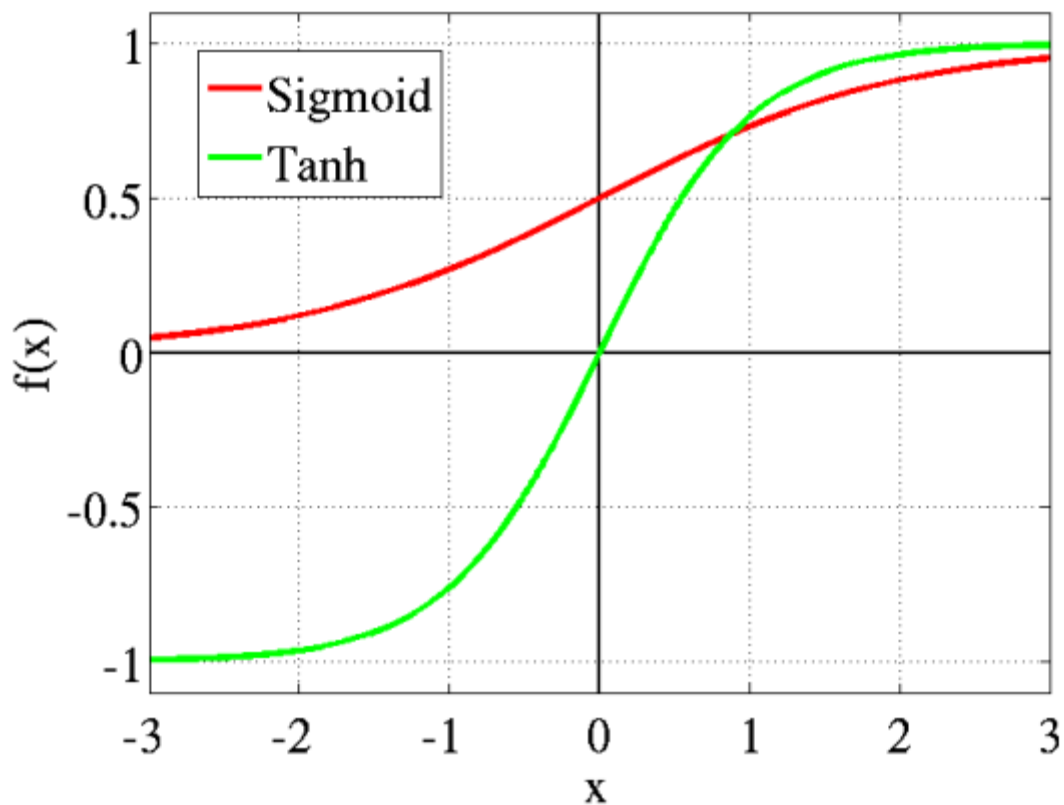
Η υπερβολική εφαπτομένη συνάρτηση, γνωστή και ως Tanh (Hyperbolic Tangent function) είναι μία συνάρτηση συνεχής και παραγωγίσιμη, με παρόμοια συμπεριφορά με τη σιγμοειδή με τη διαφορά ότι το Y μπορεί να πάρει αρνητικές τιμές με όρια από -1 έως 1 και είναι κεντραρισμένη γύρω από το σημείο (0,0) όπως φαίνεται στο σχήμα 2.6. Το μεγάλο πλεονέκτημα της σε σχέση με τη σιγμοειδή είναι ότι οι αρνητικές εισοδοί θα καταλήξουν σε αρνητικές τιμές και οι μηδενικές εισοδοί θα καταλήξουν κοντά στο 0 ενώ στη σιγμοειδή οι αρνητικές εισοδοί καταλήγουν σε θετικές τιμές όπως φαίνεται στο σχήμα 2.7. (Szandała 2021)

Η μαθηματική της σχέση δίνεται ως:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.3)$$



Σχήμα 2. 6: Υπερβολική εφαπτομένη συνάρτηση(Siebenrock [2018] 2021).

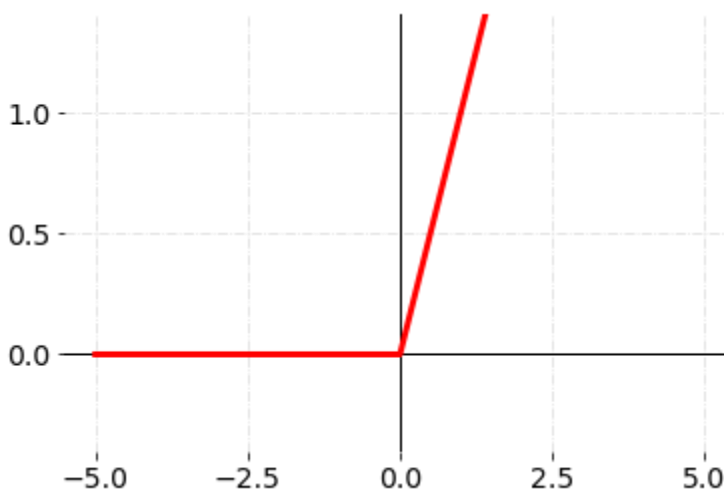


Σχήμα 2. 7: Σύγκριση σιγμοειδής και υπερβολικής εφαπτομένης συνάρτησης (Sharma et al. 2020)

Ανορθωτική γραμμική συνάρτηση

Η ανορθωτική γραμμική συνάρτηση ή αλλιώς ReLU (Rectified Linear Unit) είναι η πιο χρησιμοποιημένη συνάρτηση στον κόσμο καθώς χρησιμοποιείται σχεδόν σε όλα τα συνελκτικά νευρωνικά δίκτυα (CNN), τα οποία έχουν σκοπό την αναγνώριση εικόνας και είναι πλέον πολύ διαδεδομένα. Πρόκειται για μία συνάρτηση που δίνει μηδενική τιμή για τιμές μικρότερες ή ίσες με το μηδέν, ενώ για μεγαλύτερες τιμές από το μηδέν, το Y λαμβάνει ακριβώς την ίδια τιμή όπως φαίνεται στην σχήμα 2.8. Η μαθηματική της συμπεριφορά εκφράζεται από την εξίσωση 2.4. Το κύριο πλεονέκτημα της είναι ότι αντιμετωπίζει το πρόβλημα των εξαφανιζόμενων κλίσεων (vanishing gradient problem) (Apicella et al. 2021) .

$$f(x) = \max(0, x) \quad (2.4)$$



Σχήμα 2. 8: Ανορθωτική γραμμική συνάρτηση (ReLU) (Siebenrock [2018] 2021).

2.3 Κυριότερες τοπολογίες ΤΝΔ

2.3.1 Perceptron

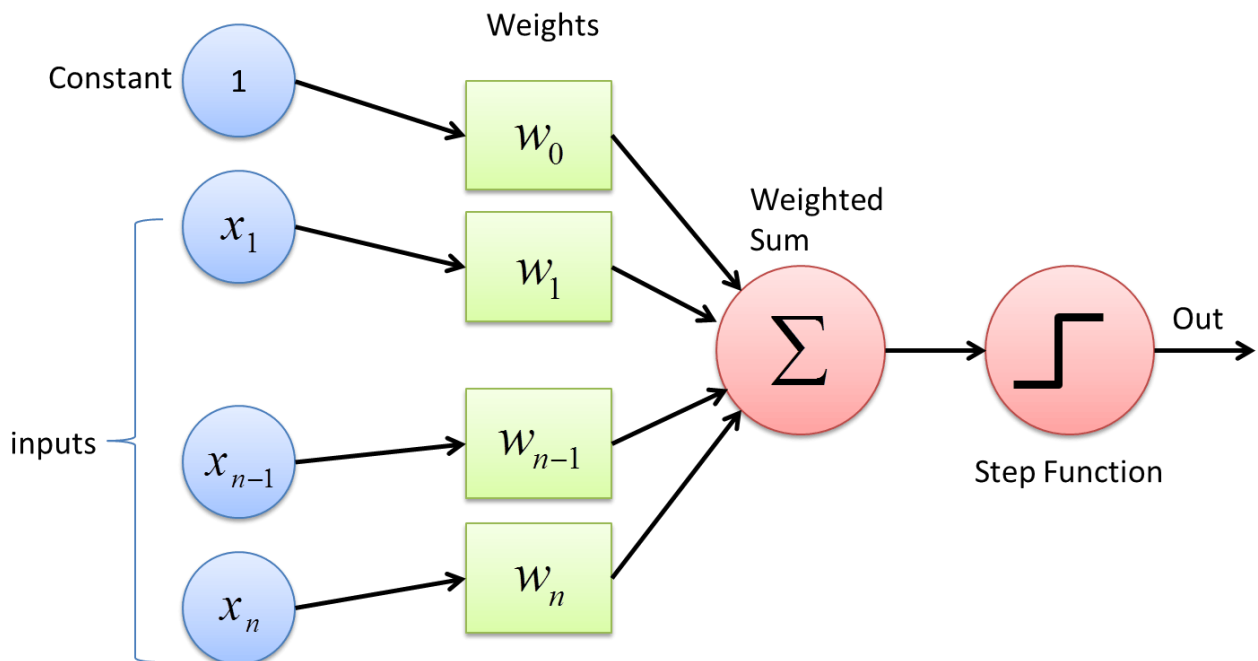
Στα τέλη της δεκαετίας του 50, το 1958, ο Frank Rosenblatt ανέπτυξε το πρώτο ΤΝΔ το οποίο ονομάζεται Perceptron (Rosenblatt 1958) . Πρόκειται για ένα ΤΝΔ πρόσθιας τροφοδότησης πολλών εισόδων και μόνο μιας εξόδου όπως φαίνεται στο σχήμα 2.9. Σήμερα όμως όλα τα ΤΝΔ που δεν περιέχουν στη δομή τους κανένα κρυφό στρώμα συνηθίζεται να αποκαλούνται perceptron. Πρόκειται για τον πιο απλό τύπο ΤΝΔ που καλείται να αντιμετωπίσει προβλήματα δυαδικής ταξινόμησης γραμμικής φύσης. Επειδή τα perceptron δεν ήταν ικανά να λύσουν προβλήματα μη γραμμικής φύσης, δημιουργήθηκαν τα ΤΝΔ με κρυφά επίπεδα. Το perceptron κατά τη διάρκεια εκπαίδευσης αποδίδει τις κατάλληλες αριθμητικές τιμές στα βάρη (weights) από τα οποία αποτελείται, όπου τα βάρη είναι ουσιαστικά υπεύθυνα για να μπορεί το ΤΝΔ να δώσει τη σωστή πρόβλεψη (Rojas 1996).

Σε κάθε τιμή της εισόδου x_i , το perceptron αποδίδει μία τιμή βάρους w_i που καθορίζει τη βαρύτητα της εισόδου για το σωστό αποτέλεσμα και το άθροισμα των γινομένων αυτών καθορίζει την έξοδο του ΤΝΔ όπως φαίνεται στην εξίσωση 2.5

$$f(x) = \sum_i^n x_i w_i + b \quad (2.5)$$

όπου b είναι ένας σταθερός όρος «bias» που δεν εξαρτάται από καμία είσοδο

Έτσι για την πρόβλεψη που θα κάνει το perceptron με βάση την εξίσωση 2.5, έχει ένα κατώφλι (threshold) με το οποίο θα μπορεί να αποφασίσει σε ποιά από τις δύο κατηγορίες θα ταξινομηθεί η πληροφορία που πήρε στην είσοδο, ανάλογα με το αν η τελική τιμή ξεπερνάει το κατώφλι ή όχι. Αυτή την εργασία την αναλαμβάνει η δυαδική βηματική συνάρτηση.

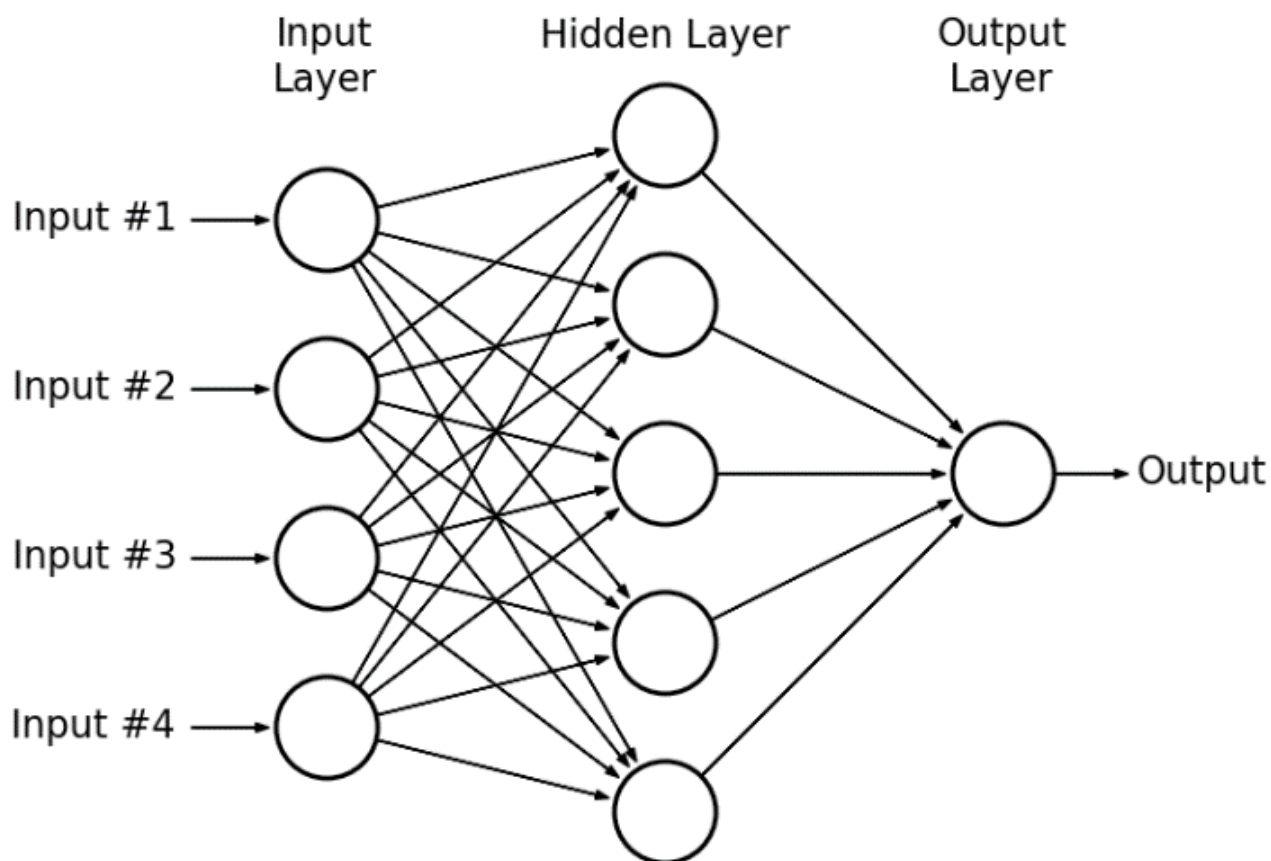


Σχήμα 2. 9: Τεχνητός νευρώνας Perceptron (“Perceptron” 2019).

2.3.2 Multilayer Perceptron (MLP)

Ένα perceptron πολλαπλών στρωμάτων (Multilayer Perceptron) απαρτίζεται από πολλαπλά perceptron συνδεδεμένα μεταξύ τους για την επίλυση πιο σύνθετων προβλημάτων από αυτό της δυαδικής ταξινόμησης. Ένας νευρώνας του συνδέεται με όλους τους νευρώνες του αμέσως επόμενου επιπέδου και η πληροφορία κατευθύνεται μόνο ως προς μία πλευρά, προς τα μπροστά ή αλλιώς προς τα δεξιά και για αυτό πρόκειται για ένα ΤΝΔ πρόσθιας τροφοδότησης όπως φαίνεται στο σχήμα 2.10 και όπως είναι και το απλό perceptron (Wesolowski and Suchacz 2012). Στις περισσότερες περιπτώσεις τα MLP αποτελούνται μόνο από τρία στρώματα (είσοδο, κρυμμένο στρώμα και hidden layer). Αυτά τα ΤΝΔ ονομάζονται «ρηγά» νευρωνικά δίκτυα (Shallow Neural Networks) ή αλλιώς «μη βαθιά» (Non-Deep) (Judd 1988).

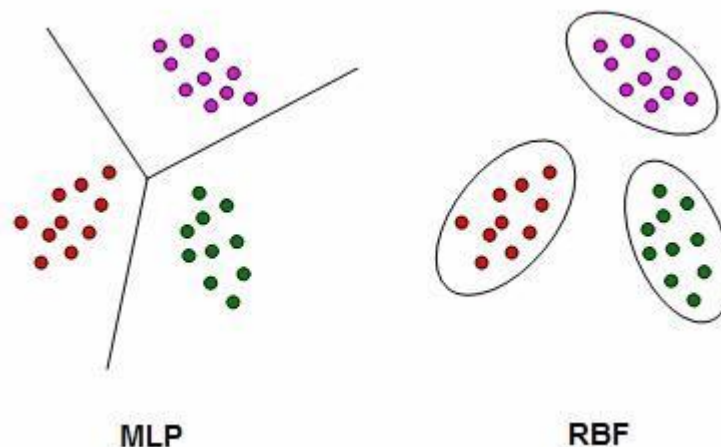
Αυτό όμως δεν συνεπάγεται ότι δεν μπορούν να έχουν παραπάνω από τρία στρώματα για την επίλυση πιο σύνθετων προβλημάτων. Ένα ΤΝΔ που έχει παραπάνω από τρία στρώματα, δηλαδή τουλάχιστον δύο κρυφά στρώματα, ονομάζεται «βαθύ» νευρωνικό δίκτυο (Deep neural network). Πέρα από το πόσα στρώματα θα αποτελείται το MLP, οι υπόλοιπες αποφάσεις που πρέπει να πάρει ο δημιουργός αφορούν τον αριθμό των νευρώνων σε κάθε στρώμα, την επιλογή της συνάρτησης ενεργοποίησης αλλά και τον αλγόριθμο ενεργοποίησης. Ο τρόπος λειτουργίας του δεν διαφέρει από αυτόν του perceptron αφού δέχεται εισόδους, τις πολλαπλασιάζει με τα βάρη και μετά στέλνει την πληροφορία στο επόμενο στρώμα με την ίδια ακριβώς λογική. Παρόλα αυτά επειδή το κάθε στρώμα μπορεί να αποτελείται από μεγάλο αριθμό κόμβων και να υπάρχουν πολλαπλά κρυφά στρώματα, το MLP μπορεί να γίνει πολύ πολύπλοκο και ταυτόχρονα πολύ ισχυρό για την επίλυση δύσκολων προβλημάτων. Επειδή στο MLP μπορεί να χρησιμοποιηθεί συνάρτηση ενεργοποίησης διαφορετική από τη δυαδική βηματική, όπως η πολυχρησιμοποιημένη σιγμοειδής, αυτό του δίνει τη δυνατότητα να δώσει πραγματικές τιμές στην έξοδο, διαφορετικές από αυτές της δυαδικής ταξινόμησης που είναι 0 και 1 ή -1 και 1. Στα προβλήματα ταξινόμησης, αυτές οι τιμές έχουν τη μορφή πιθανότητας που υποδηλώνουν την πιθανότητα των εισόδων να ανήκουν σε κάποια κατηγορία. Το MLP είναι ικανό να λύσει προβλήματα ταξινόμησης εισόδων με πολλαπλές κατηγορίες και όχι μόνο δύο (Ramchoun et al. 2016).



Σχήμα 2. 10: Multilayer Perceptron με 3 στρώματα (Hassan et al. 2015).

2.3.3 Radial Basis Function (RBF)

Τα ΤΝΔ RBF είναι και αυτά ΤΝΔ πρόσθιας τροφοδότησης και έχουν ακριβώς την ίδια δομή με αυτή των MLP. Αποτελείται αυστηρά από τρία στρώματα, είσοδος, κρυμμένο στρώμα και έξοδος. Τα RBF είναι λιγότερο πολύπλοκα από τα MLP για τον εμφανή λόγο ότι δεν μπορούν να αποτελούνται από παραπάνω από τρία στρώματα. Αυτό έχει ως αποτέλεσμα και την πιο εύκολη εκπαίδευση τους σε σχέση με τα MLP. Άλλες διαφορές μεταξύ των δύο είναι ότι ενώ στο MLP η έξοδος αποφασίζεται από όλους τους νευρώνες, στο RBF η έξοδος προσδιορίζεται από συγκεκριμένους νευρώνες σε συγκεκριμένα σημεία. Όσον αφορά την εκπαίδευση τους είναι πολύ σημαντικό στο RBF να δοθούν ακριβείς αρχικές τιμές στις παραμέτρους τους (βάρη κλπ.) ενώ στο MLP δίνονται τυχαίες αρχικές τιμές. Η πιο σημαντική διαφορά και ο λόγος ο οποίος τα RBF παίρνουν το όνομά τους, είναι ότι χρησιμοποιούν συνάρτηση ενεργοποίησης ακτινικής βάσης (radial basis function). Εν κατακλείδι τα RBF μπορούν να χαρακτηριστούν ως απλά ΤΝΔ πρόσθιας τροφοδότησης με συνάρτηση ενεργοποίησης ακτινικής βάσης. Όσον αφορά τις διαφορές τους σε πειραματικά αποτελέσματα, τα RBF ταξινομούν εισόδους με βάση όριο σφαιρών ενώ τα MLP με βάση όριο γραμμών όπως φαίνεται στο σχήμα 2.11. (Yu et al. 2011)



Σχήμα 2. 11: Διαφορά τρόπου ταξινόμησης μεταξύ MLP και RBF (Chandradevan 2017).

2.3.4 Recurrent Neural Networks (RNN)

Όπως αναφέρθηκε στην υποενότητα 2.2.1, πρόκειται για ΤΝΔ που διαφέρουν από τα πολυχρησιμοποιημένα ΤΝΔ πρόσθιας τροφοδότησης στο ότι στη δομή τους έχουν τουλάχιστον έναν κόμβο που ανατροφοδοτεί το σήμα σε έναν κόμβο ίδιου επιπέδου ή προηγούμενου. Χρησιμοποιούνται κυρίως σε περιπτώσεις που χρησιμοποιούν διαδοχικά δεδομένα εισόδου (sequential data) τα οποία δεδομένα αλληλοσχετίζονται μεταξύ τους οπότε η ανατροφοδότηση βοηθάει πολύ στην επίλυση του προβλήματος. Σε αντίθεση με το MLP που χρησιμοποιεί συγκεκριμένο αριθμό εισόδων, το RNN δεν χρειάζεται προκαθορισμένο αριθμό εισόδων και είναι ικανό να λάβει όλες τις διαθέσιμες πληροφορίες που του παρέχονται στην είσοδο έως τη χρονική στιγμή t_c ($x_t, t = 1, 2, \dots, t_c$) προκειμένου να γίνει πρόβλεψη για την έξοδο y την ίδια χρονική στιγμή y_{t_c} . Οι εισοδοί λαμβάνονται με χρονική σειρά και όχι την ίδια χρονική στιγμή όπως στο MLP.

Επίσης τα RNN είναι απόλυτα ικανά και πολλές φορές χρησιμοποιούνται προκειμένου να προβλέψουν τις μελλοντικές τιμές ενός σήματος με βάση την πληροφορία που λαμβάνουν από ένα σήμα εισόδου. Παρόλα αυτά όμως είναι σημαντικό να αναφερθεί ότι η αποδοτικότητα τους μειώνεται αν οι μελλοντικές πληροφορίες που καλούνται να προβλέψουν είναι πολύ μακρινές στο μέλλον. (Schuster and Paliwal 1997)

Είναι εμφανές λοιπόν ότι τα RNN χρησιμοποιούν την έννοια του χρόνου και είναι ιδανικά για αυτό το λόγο για προβλήματα με σήματα ως εισόδους. Επειδή όμως πρόκειται για πολύ ισχυρά ΤΝΔ, αυτό συνεπάγεται και με μεγάλη πολυπλοκότητα και ταυτόχρονα ανάγκη για μεγαλύτερη υπολογιστική ισχύ.

2.3.5 Convolutional Neural Networks (CNN)

Τα συνελκτικά νευρωνικά δίκτυα είναι ένα είδος νευρωνικών δικτύων βαθιάς μηχανικής μάθησης (deep learning). Επινοήθηκαν για την αναγνώριση προτύπων, εικόνων, βίντεο και την κατάταξή τους σε κατηγορίες. Η ReLU που αναφέρθηκε περιληπτικά στην υποενότητα 2.1.2 είναι η πιο χρησιμοποιημένη συνάρτηση ενεργοποίησης στα CNN και η άνοδος της σχετίζεται άμεσα με την άνοδο των CNN τα τελευταία χρόνια (Vedaldi and Lenc 2016). Στην επίλυση των συγκεκριμένων προβλημάτων τα CNN έχουν πολύ καλύτερη επίδοση σε σχέση με άλλες κλασσικές μεθόδους. Αποτελούνται από τριών ειδών στρώματα, συνελκτικά (convolution), «pooling» και τα πλήρως συνδεδεμένα στρώματα (fully-connected). Οι εικόνες είναι ουσιαστικά ένας πίνακας δύο διαστάσεων. Το CNN πραγματοποιεί συνέλιξη του πίνακα αυτού με το γνωστό φίλτρο Kernel προκειμένου να μπορέσει να εξάγει τα χαρακτηριστικά των εικόνων που έλαβε ως εισόδους. Στο pooling layer το CNN προσπαθεί να μειώσει την ανάλυση της εικόνας και να κρατήσει μόνο τις απαραίτητες πληροφορίες που του χρειάζονται για να φτάσει σε πρόβλεψη. Σήμερα υπάρχουν πολλές διαφορετικές αρχιτεκτονικές και αλγόριθμοι εκπαίδευσης CNN και πρόκειται για ΤΝΔ μεγάλης πολυπλοκότητας που συνεχώς εξελίσσονται. (Gu et al. 2017)

2.4 Εκπαίδευση ΤΝΔ

2.4.1 Στάδια μηχανικής μάθησης

Σταδιο Εκπαίδευσης

Όλα τα νευρωνικά δίκτυα δέχονται έναν συγκεκριμένο αριθμό εισόδων και παρέχουν ένα συγκεκριμένο αριθμό εξόδων, ανάλογα με τις ανάγκες του προβλήματος που έχουν να επιλύσουν. Ο τρόπος με τον οποίο θα εκπαιδευτούν δεν είναι συγκεκριμένος, Σε οποιαδήποτε περίπτωση θα πρέπει να δοθούν στο ΤΝΔ κάποια δεδομένα εισόδων τα οποία είναι παρόμοια με αυτά για τα οποία πρέπει να μάθει τι να κάνει. Το ΤΝΔ με βάση τις εισόδους που θα του δοθούν, θα επεξεργαστεί τα πρότυπα των εισόδων και θα προσαρμόσει τη δομή του για να μπορέσει να δώσει την επιθυμητή πρόβλεψη.

Αυτή η διαδικασία μπορεί να διαρκέσει πολύ χρόνο και να γίνει αρκετές φορές μέχρι να υπάρξουν τα επιθυμητά αποτελέσματα με όσο το δυνατόν μικρότερο σφάλμα. Το στάδιο της εκπαίδευσης θεωρείται ότι ολοκληρώθηκε με επιτυχία όταν το σφάλμα μεταξύ επιθυμητής εξόδου και πραγματικής εξόδου έχει την επιθυμητή τιμή.

Σταδιο Ελέγχου

Όταν η διαδικασία της εκπαίδευσης φτάσει στο τέλος της, σειρά παίρνει αυτή του ελέγχου προκειμένου να μπορέσει να δει ο δημιουργός του ΤΝΔ την αποτελεσματικότητα του δικτύου. Έτσι λοιπόν το δίκτυο καλείται να αντιμετωπίσει προβλήματα που δεν έχει ξανασυναντήσει, δηλαδή του δίνονται κάποιες εισοδοί που δεν έχει επεξεργαστεί προηγουμένως και βάση της εκπαίδευσής του θα πρέπει να κάνει μία πρόβλεψη. Προφανώς οι εισοδοί για τις οποίες θα επικαλεσθεί το δίκτυο να κάνει πρόβλεψη θα είναι ανάλογες με αυτές πάνω στις οποίες εκπαιδεύτηκε. Εάν δοθούν εισοδοί διαφορετικής φύσης τότε το δίκτυο δεν θα καταφέρει να δώσει σωστές προβλέψεις.

Όπως γίνεται κατανοητό από τα προαναφερθέντα το στάδιο της εκπαίδευσης είναι η πιο σημαντική λειτουργία ενός ΤΝΔ για να μπορέσει να δουλέψει σωστά και να αποφέρει τα επιθυμητά αποτελέσματα.

2.4.2 Βασικά είδη εκπαίδευσης

Στη μηχανική μάθηση υπάρχουν διαφορετικοί τρόποι εκπαίδευσης με τους δύο πιο βασικούς να είναι οι παρακάτω:

- **Επιβλεπόμενη μάθηση (Supervised learning)**, όπου πρόκειται για μια διαδικασία στην οποία ο δημιουργός παρέχει ένα σετ εισόδων στο ΤΝΔ για το οποίο οι επιθυμητές εξοδοί του είναι γνωστοί προκειμένου να μπορέσει το ΤΝΔ να κατασκευάσει μια γενική σχέση για να χρησιμοποιήσει με επιτυχία σε εισόδους με άγνωστες εξόδους. Η διαδικασία της εκπαίδευσης λειτουργεί με βάση την αριθμητική διαφορά της πραγματικής πρόβλεψης σε σχέση με την επιθυμητή πρόβλεψη που έχει δώσει ο δημιουργός (Almási et al. 2016).
- **Μη επιβλεπόμενη μάθησης (Unsupervised learning)** όπου σε αντίθεση με την επιβλεπόμενη μάθηση, το ΤΝΔ δεν γνωρίζει τις επιθυμητές εξόδους και λαμβάνει μόνο τις εισόδους από τον δημιουργό. Αυτή η τεχνική χρησιμοποιείται κυρίως για προβλήματα ομαδοποίησης δεδομένων και αλυσίδες Markov (Almásí et al. 2016).

2.4.3 Μέθοδος οπισθοδιάδοσης σφάλματος (Error Backpropagation)

Πρόκειται για έναν από τους πιο διαδεδομένους αλγορίθμους εκπαίδευσης ενός ΤΝΔ για εκπαίδευση επιβλεπόμενης μάθησης. Προτού το δίκτυο λάβει τα δεδομένα εισόδων για να εκπαιδευτεί, και αφού έχουν οριστικοποιηθεί όλες οι παράμετροι, δίνει κάποιες αρχικές τιμές στα βάρη του για όλα τα στοιχεία του (νευρώνες). Αυτές οι τιμές μπορεί να είναι μηδενικές ή τυχαίες χωρίς ιδιαίτερη σημασία. Έπειτα το ΤΝΔ είναι έτοιμο να λάβει τις εισόδους και τις επιθυμητές εξόδους από τον χρήστη για τη διαδικασία της εκπαίδευσης.

Αφού λάβει τα δεδομένα των εισόδων, θα κάνει μία πρόβλεψη για κάθε δείγμα εισόδου. Εδώ ξεκινάει ολόκληρη η κύρια διαδικασία της μεθόδου οπισθοδιάδοσης. Το δίκτυο θα συγκρίνει την πραγματική έξοδο του, που έχει προκύψει από την μετάβαση των δεδομένων εισόδου σε κάθε στρώμα με τις αρχικές τιμές των βαρών, με την επιθυμητή έξοδο που έχει δώσει ο χρήστης. Όπως προαναφέρθηκε οι αρχικές τιμές των βαρών είναι τυχαίες, οπότε σχεδόν πάντα η επιθυμητή έξοδος με την πραγματική έξοδο θα έχουν ανεπιθύμητο σφάλμα. Από εκεί το ΤΝΔ λαμβάνει το σφάλμα και ξεκινάει να διορθώνει τις τιμές των βαρών των κρυφών επιπέδων προς τα πίσω (από την έξοδο προς την είσοδο). Έτσι το δίκτυο καθορίζει τις απαραίτητες αλλαγές στα βάρη των νευρώνων του προκειμένου να φτάσει όσο πιο κοντά στις επιθυμητές εξόδους με όσο το δυνατόν μικρότερο σφάλμα. Η διαδικασία αυτή μπορεί να χρειαστεί να εκτελεστεί πολλές φορές μέχρι η τιμή του σφάλματος εξόδου να φτάσει την επιθυμητή. (Γεωργούλη 2015)

Το σφάλμα μίας εξόδου o_{kp} ενός νευρώνα k από την επιθυμητή έξοδο a_{kp} , αναφορικά με ένα δείγμα p :

$$E_k = (a_{kp} - o_{kp}) \quad (2.6)$$

Το σφάλμα θα πολλαπλασιαστεί με την παράγωγο της επιλεγμένης συνάρτησης ενεργοποίησης στον νευρώνα k (u_k). Από αυτόν τον υπολογισμό προκύπτει το προσαρμοσμένο σφάλμα νευρώνα:

$$\delta_k = (a_{kp} - o_{kp}) \times g'(u_k) \quad (2.7)$$

Έτσι και το αντίστοιχο σφάλμα για έναν νευρώνα σε ένα κρυφό στρώμα i υπολογίζεται από το σύνολο των προσαρμοσμένων σφαλμάτων k μονάδες του επόμενου επιπέδου όπου ο νευρώνας συνδέεται με βάρη w_{ik} από τον τύπο:

$$\delta_i = g'(u_i) \times \sum_1^k w_{ik} \times \delta_k \quad (2.8)$$

Εάν η παραπάνω διαδικασία εκτελεστεί πολλές φορές και δεν καταφέρει να μειώσει το σφάλμα του ΤΝΔ σε μία επιθυμητή τιμή, υπάρχει η επιλογή να σταματήσει τη διόρθωση σφαλμάτων μετά από έναν ορισμένο αριθμό επαναλήψεων που έχει ορισθεί ή μετά την ολοκλήρωση ενός συγκεκριμένου χρονικού διαστήματος, προκειμένου κάποια στιγμή να σταματήσουν οι κύκλοι διόρθωσης.

Μετά τον υπολογισμό του σφάλματος δ_i για κάθε νευρώνα i , η τιμή στα βάρη εισόδου σε όλους τους νευρώνες καθορίζεται ως εξής:

$$\Delta W_{ji} = -d \times \delta_j \times a_j \quad (2.9)$$

Όπως φαίνεται από την μαθηματική εξίσωση 2.9, η αλλαγή στο βάρος από τον νευρώνα i , στον νευρώνα j (ΔW_{ji}) εξαρτάται από τον ρυθμό μάθησης (d), το σφάλμα του νευρώνα j (δ_j) και την έξοδο του νευρώνα j (a_j). (Γεωργούλη 2015)

Η αλλαγή των βαρών γίνεται μέσω των παραδειγμάτων εκπαίδευσης που δίνονται στο δίκτυο και υπάρχουν δύο διαφορετικοί τρόποι για την αλλαγή τους:

- **Αυξητική εκπαίδευση** (incremental training), όπου οι αλλαγές στις τιμές των βαρών γίνονται άμεσα για κάθε δείγμα που παρουσιάζεται (Γεωργούλη 2015).
- **Μαζική εκπαίδευση** (batch training) , όπου παρουσιάζονται πρώτα όλα τα παραδείγματα και ο μόνος υπολογισμός είναι αυτός των αλλαγών των βαρών για κάθε δείγμα. Στο τέλος αφού έχουν παρουσιαστεί όλα τα παραδείγματα, την ίδια ακριβώς στιγμή γίνεται η αλλαγή των τιμών των βαρών. Πρόκειται για μία αρκετά χρησιμοποιούμενη μέθοδο για την εκπαίδευση των δικτύων. Υπάρχει και η δυνατότητα να ορίσει ο χρήστης το μέγεθος της μαζικής εκπαίδευσης για την οποία γίνεται η εκπαίδευση, δηλαδή να χωρίσει τα παραδείγματα σε πολλαπλές ομάδες αντί για μία μεγάλη ομάδα με όλα τα παραδείγματα (mini-batch size) (Γεωργούλη 2015).

Η παρουσίαση όλων των παραδειγμάτων των δεδομένων εκπαίδευσης αναλογεί σε έναν κύκλο εκπαίδευσης και ορίζεται ως εποχή εκπαίδευσης (training epoch). Συνήθως είναι απαραίτητο να ολοκληρωθούν πολλαπλές εποχές εκπαίδευσης μέχρι το ΤΝΔ να μπορεί να παράγει τα επιθυμητά αποτελέσματα.

2.5 Παραδείγματα επίλυσης προβλημάτων με ΤΝΔ

2.5.1 Ταξινόμηση γραμμικά διαχωρίσιμων εισόδων με Perceptron

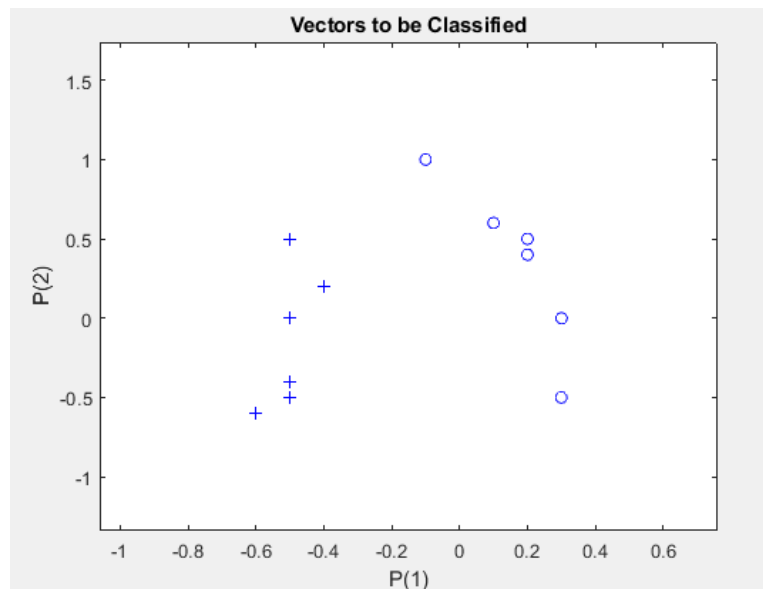
Ένα perceptron εκπαιδεύεται για να μπορεί να ταξινομήσει διάφορες εισόδους δύο διαστάσεων σε δύο κατηγορίες. Οι εισοδοί πρέπει να είναι γραμμικά διαχωρίσιμες προκειμένου το perceptron να μπορεί να τις ταξινομήσει. Έπειτα θα μπορεί να κάνει το ίδιο πράγμα για νέες άγνωστες εισόδους.

Τα βήματα για τη διαδικασία χωρίζονται σε 3 μέρη:

- Ορισμός δεδομένων εισόδου και επιθυμητών προβλέψεων για τη σωστή εκπαίδευση του perceptron
- Δημιουργία και εκπαίδευση perceptron
- Σχεδίαση ορίου απόφασης (threshold)

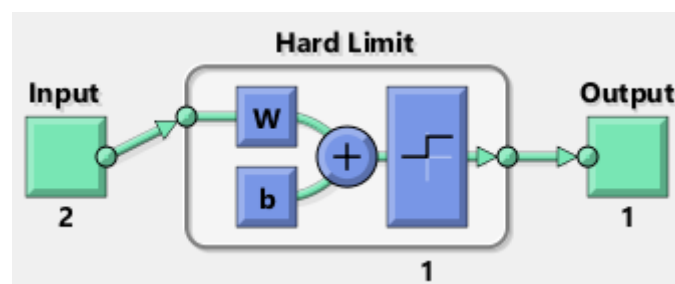
α) Ορισμός δεδομένων

Πρώτα θα δημιουργηθούν γραμμικά διαχωρίσιμα δεδομένα προκειμένου να μπορέσει να εκπαιδευτεί το perceptron.



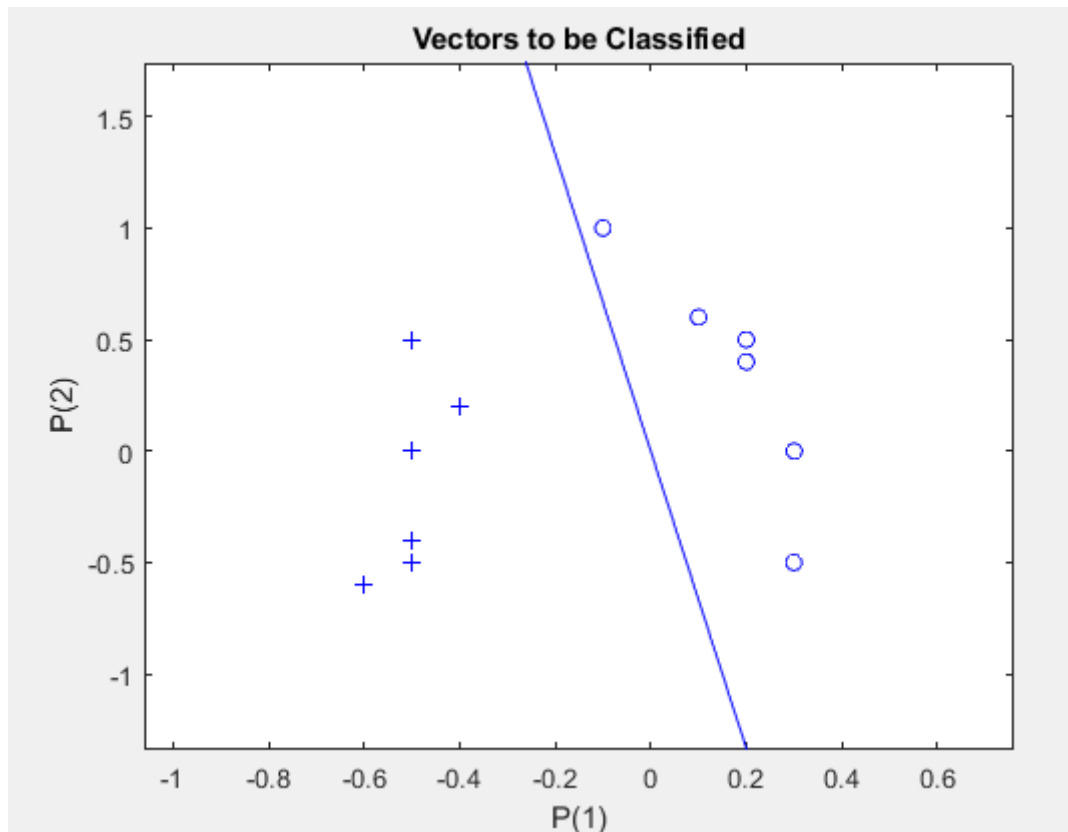
Σχήμα 2. 12: Δημιουργία δύο γραμμικά διαχωρίσιμων ομάδων δεδομένων.

β) Δημιουργία και εκπαίδευση perceptron



Σχήμα 2. 13: Δημιουργία δυαδικού ταξινομητή perceptron

γ) Σχεδίαση ορίου απόφασης



Σχήμα 2. 14: Δημιουργία ορίου απόφασης (threshold) για ταξινόμηση στοιχείων.

Όπως φαίνεται στο σχήμα 2.14, το ΤΝΔ perceptron δημιούργησε μία γραμμή που λειτουργεί σαν όριο απόφασης για να μπορεί να διαχωρίζει τα δεδομένα σε δύο κατηγορίες και να μπορεί να τα ταξινομεί.

Αφού ολοκληρώθηκε η διαδικασία εκπαίδευσης, το ΤΝΔ είναι έτοιμο να ταξινομήσει στοιχεία άγνωστα με αυτά του σταδίου εκπαίδευσης, προκειμένου να δοκιμαστεί η αποδοτικότητα του. Αυτό είναι και το στάδιο ελέγχου.

2.5.2 Επίλυση προβλήματος λογικής πύλης XOR με MLP

Το συγκεκριμένο πρόβλημα δεν μπορεί να λυθεί από ένα απλό ΤΝΔ perceptron με έναν νευρώνα γιατί πρόκειται για μη γραμμικά διαχωρίσιμο πρόβλημα, οπότε θα χρειαστεί ένα MLP.

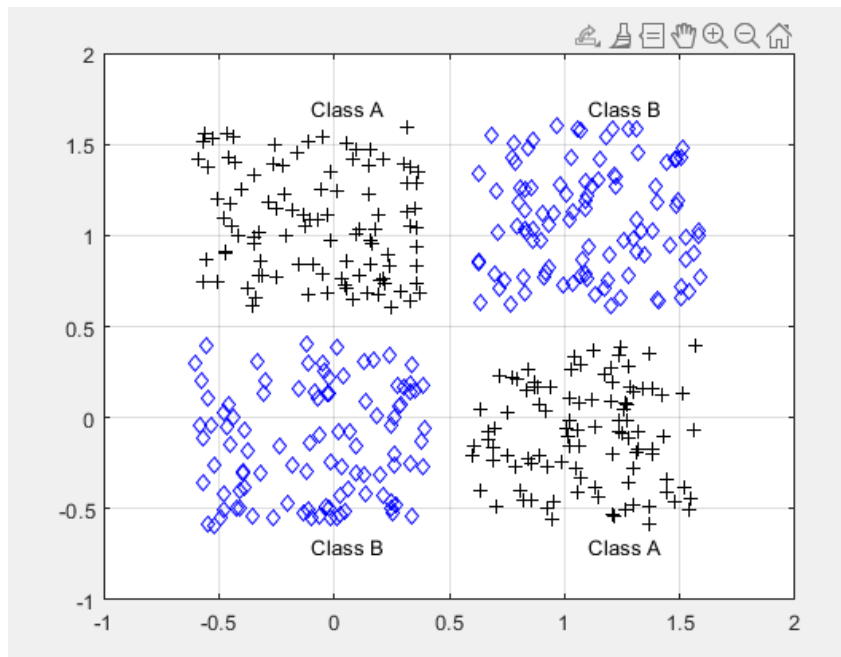
Το πρόβλημα της λογικής πύλης XOR είναι γνωστό. Πρέπει το ΤΝΔ να μπορεί να κάνει πρόβλεψη με βάση τον πίνακα αληθείας όπως φαίνεται στο σχήμα 2.15 για δύο δυαδικές εισόδους. Δηλαδή όταν οι δύο εισοδοί είναι διαφορετικές ή μία από την άλλη (π.χ. 1 και 0) τότε η έξοδος θα πρέπει να είναι 1, ενώ όταν οι εισοδοί είναι ίδιες τότε η έξοδος θα πρέπει να είναι 0.

x_1	x_2	$x_1 \text{ XOR } x_2$
0	0	0
0	1	1
1	0	1
1	1	0

Σχήμα 2. 15: Πίνακας αληθείας λογικής πύλης XOR(Seising 2018)

α) Ορισμός εισόδων προβλήματος

Σε ένα χώρο δύο διαστάσεων, ορίζονται τέσσερις ομάδες (A, B, C, D) δεδομένων εισόδου οι οποίες χωρίζονται σε δύο ζευγάρια (A, C) και (B, D) όπου αντιπροσωπεύουν το πρόβλημα της λογικής πύλης XOR. (“Solving XOR Problem with a Multilayer Perceptron” 2012)



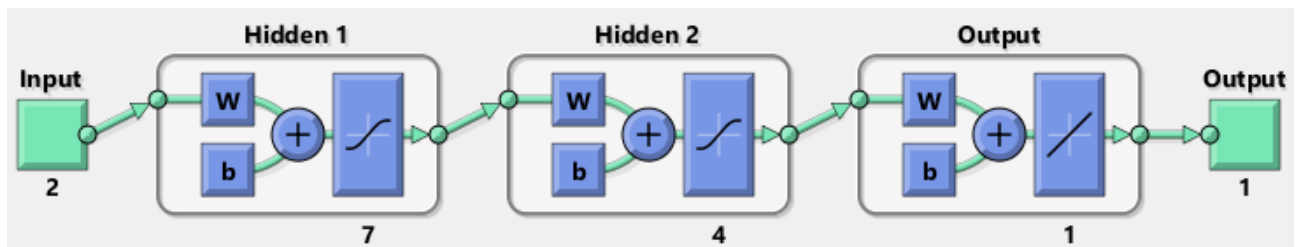
Σχήμα 2. 16: Δημιουργία δύο κλάσεων για δυαδική κατηγοριοποίηση.

β) Προετοιμασία και τροποποίηση εισόδων και εξόδων

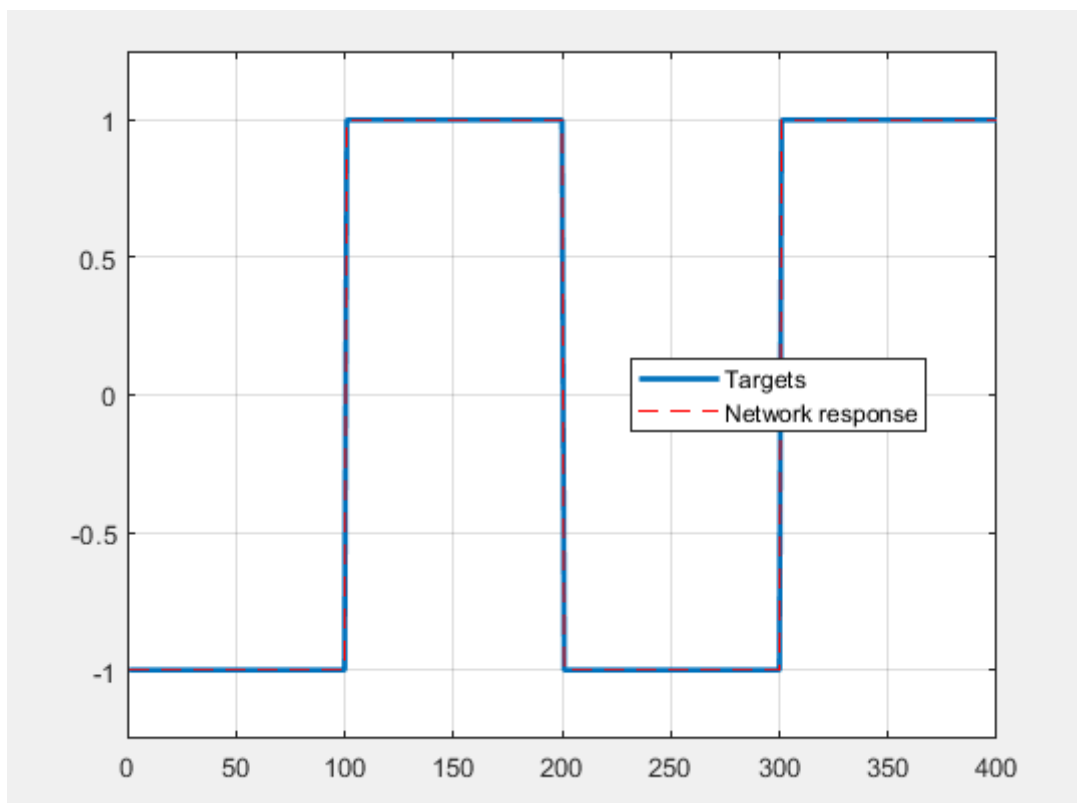
Στον κώδικα του MATLAB όπου γίνεται το πείραμα, οι εισοδοί και οι έξοδοι πρέπει να τηρούν κάποιες συγκεκριμένες προϋποθέσεις προκειμένου να μπορέσει να τις επεξεργαστεί το ΤΝΔ και να εκπαιδευτεί πάνω σε αυτές. (“Solving XOR Problem with a Multilayer Perceptron” 2012)

γ) Δημιουργία και εκπαίδευση MLP

Όπως φαίνεται στο σχήμα 2.17, δημιουργήθηκε ένα MLP με δύο κρυφά στρώματα, με επτά νευρώνες στο πρώτο στρώμα και τέσσερις στο δεύτερο στρώμα. Στο σχήμα 2.18 φαίνεται η απόκριση του ΤΝΔ σε σχέση με την επιθυμητή.



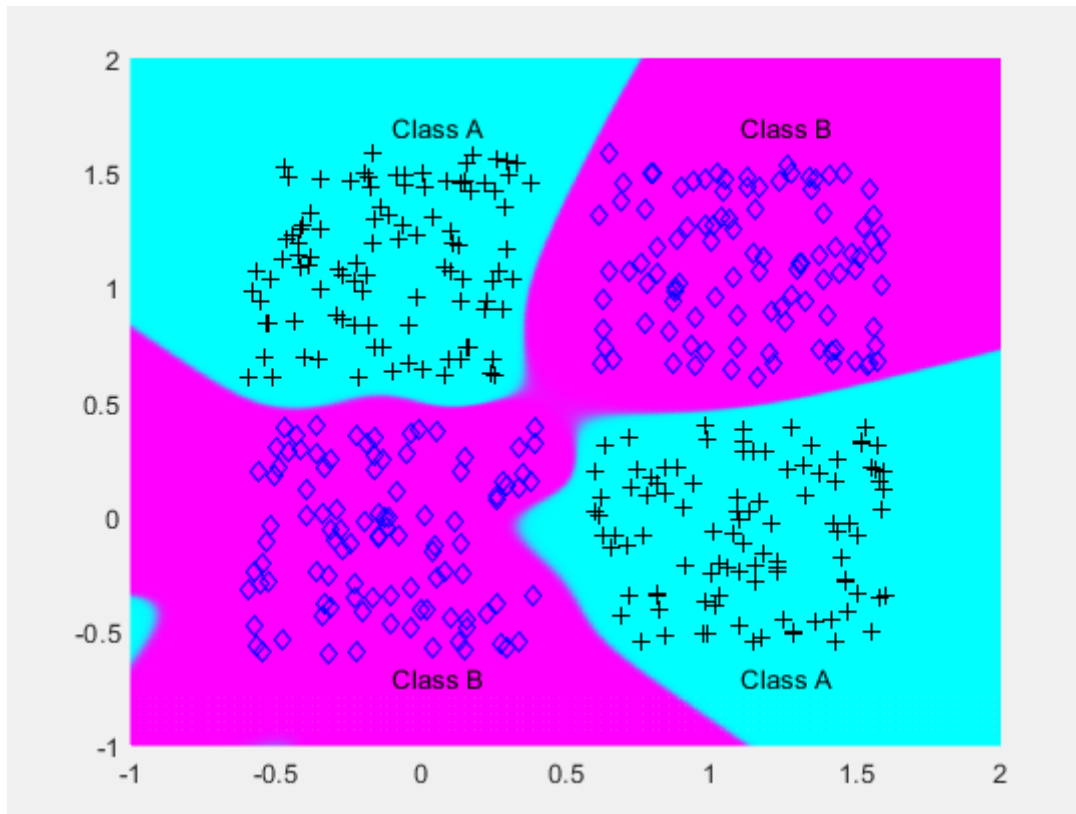
Σχήμα 2. 17: Δημιουργία MLP με δύο κρυφά στρώματα.



Σχήμα 2. 18: Σύγκριση επιθυμητής και πραγματικής απόκρισης ΤΝΔ.

δ) Ταξινόμηση των δύο κλάσεων με διαχωριστικές γραμμές

Όπως φαίνεται στο σχήμα 2.19, το ΤΝΔ χωρίζει τις δύο κλάσεις με τη βοήθεια γραμμών και δύο χρωμάτων. (“Solving XOR Problem with a Multilayer Perceptron” 2012)



Σχήμα 2. 19: Ταξινόμηση δύο κλάσεων με τη βοήθεια δύο χρωμάτων.

ΚΕΦΑΛΑΙΟ 3^ο : Εξίσωση καναλιού με χρήση ΤΝΔ

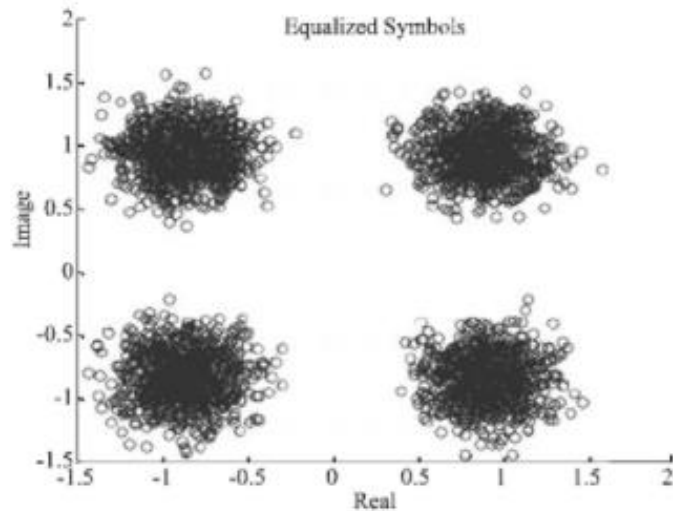
Όπως προαναφέρθηκε η εξίσωση καναλιού είναι μία τεχνική που χρησιμοποιείται για την καταπολέμηση της ISI και του προσθετικού θορύβου αλλά σε πραγματικές πρακτικές συνθήκες δεν είναι λίγες οι φορές που χρειάζεται προσαρμοστικός εξισωτής καθώς τα χαρακτηριστικά του καναλιού αλλάζουν συνεχώς. Όπως ειπώθηκε παραπάνω, ένας προσαρμοστικός εξισωτής πρώτα εκπαιδεύεται και στη συνέχεια αλλάζει τις παραμέτρους του μέσω του αλγορίθμου που χρησιμοποιεί όπως ακριβώς λειτουργούν και τα ΤΝΔ. Το πρόβλημα της εξίσωσης καναλιού γίνεται πιο δύσκολο όταν το κανάλι τηλεπικοινωνίας έχει μη γραμμική συμπεριφορά. Οι προσαρμοστικοί γραμμικοί εξισωτές είναι μία καλή λύση όταν τα κανάλια έχουν γραμμική συμπεριφορά αλλά δεν έχουν καλή απόδοση όταν καλούνται να αντιμετωπίσουν κανάλια μη γραμμικής φύσης. Τα ΤΝΔ όμως έχουν φτιαχτεί κυρίως για να μπορούν να αντιμετωπίζουν δύσκολα προβλήματα μη γραμμικής φύσης λόγω των μη γραμμικών συναρτήσεων ενεργοποίησης που χρησιμοποιούν. Έτσι τα τελευταία χρόνια έχουν γίνει πολλές δοκιμές και τα ΤΝΔ έχουν δείξει ότι είναι ιδανικές λύσεις για εξίσωση καναλιού μη γραμμικής φύσης (Lavanaia et al. 2015).

3.1 Εξίσωση καναλιού χρησιμοποιώντας MLP

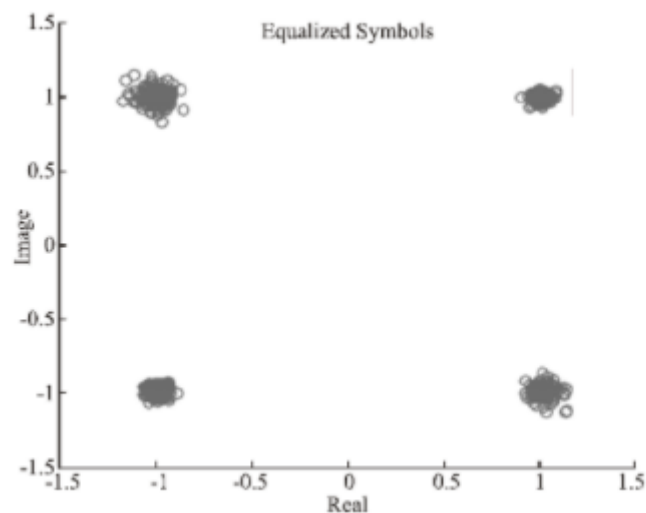
Η εξίσωση καναλιού μπορεί να χαρακτηριστεί ως ένα γεωμετρικό πρόβλημα ταξινόμησης. Αυτό σημαίνει ότι ο στόχος του εξισωτή είναι να μπορέσει να βρει τα όρια αποφάσεων για τα λαμβανόμενα σύμβολα προκειμένου να μπορέσει να προβλέψει σωστά αυτά τα σύμβολα τα οποία λαμβάνει. Το πρόβλημα όμως είναι ότι αυτά τα όρια απόφασης είναι συνήθως μη γραμμικά όρια. Η εξίσωση καναλιού έχει στόχο την αντιστροφή της συμπεριφοράς του καναλιού αλλά αυτό δεν χρειάζεται να επιτευχθεί και μπορεί το πρόβλημα να αντιμετωπιστεί ως ένα απλό πρόβλημα ταξινόμησης με τη μόνη δυσκολία να τεθούν σωστά όρια αποφάσεων. Για αυτό το λόγο τα ΤΝΔ είναι ιδανικά αφού είναι ικανά να λύσουν μη γραμμικά προβλήματα ταξινόμησης. Πέρα από το ότι είναι ικανά να λύσουν μη γραμμικά προβλήματα, όπου είναι και το πιο σημαντικό, έχουν και την ικανότητα να λειτουργούν προσαρμοστικά (K T and Tripathy 2006).

Στην περίπτωση των MLP, όλες του οι είσοδοι θα είναι ολισθημένες εκδοχές του σήματος που έχει περάσει από το κανάλι τηλεπικοινωνίας. Όταν οι έξοδοι του καναλιού περνάνε μέσα από τα κρυμμένα στρώματα του ΤΝΔ, αυτό μέσω των μη γραμμικών συναρτήσεων ενεργοποίησης που έχει καταφέρει να φτιάξει μη γραμμικά όρια απόφασης για την ταξινόμηση που θα επικαλεστεί να κάνει. Συνήθως χρησιμοποιείται ο αλγόριθμος back-propagation και εφόσον πρόκειται για έναν αλγόριθμο επιβλεπόμενης μάθησης, ο MLP εξισωτής θα χρειαστεί να εκπαιδευτεί προτού καλεστεί να κάνει πρόβλεψη (K T and Tripathy 2006).

Οι Baloch, Baloch και Unar πραγματοποίησαν ένα πείραμα προκειμένου να συγκρίνουν τις αποδόσεις κάποιων κλασικών τεχνικών εξίσωσης καναλιού με την απόδοση ενός MLP εξισωτή (Baloch, Baloch, and Unar 2012). Τα αποτελέσματα του πειράματός τους φαίνονται στο σχήμα 3.1 και στο σχήμα 3.2. Πρόκειται για ένα σήμα που περνάει μέσα από ένα AWGN κανάλι με μετάδοση 4-QAM.



Σχήμα 3. 1: Εξισωμένα σύμβολα διαμόρφωσης 4-QAM με χρήση εξισωτή αλγορίθμου LMS (Baloch, Baloch, and Unar 2012).



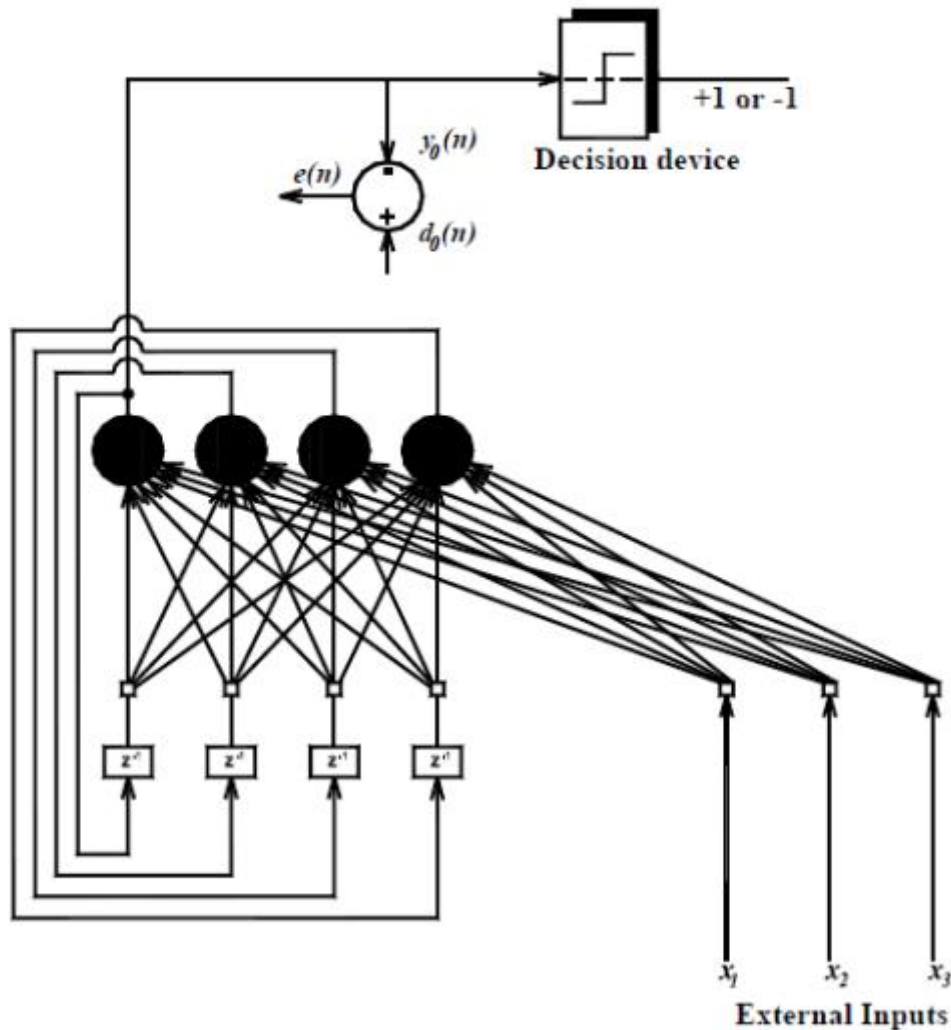
Σχήμα 3. 2: Εξισωμένα σύμβολα διαμόρφωσης 4-QAM με χρήση MLP ΤΝΔ (Baloch, Baloch, and Unar 2012).

Είναι φανερό ότι η παρουσία θορύβου χρησιμοποιώντας το μοντέλο του εξισωτή MLP είναι σημαντικά μικρότερη από αυτό του κλασικού εξισωτή που χρησιμοποιεί αλγόριθμο LMS.

Επίσης από τα διαγράμματα αστερισμών των σχημάτων 3.1 και 3.2 μπορεί να διαπιστώσει κανείς για ποιο λόγο η εξίσωση καναλιού με ΤΝΔ μπορεί να θεωρηθεί ως ένα απλό πρόβλημα ταξινόμησης με τα σύμβολα πληροφορίας εισόδου να μπορούν να εκφραστούν ως κλάσεις και έτσι το ΤΝΔ να κάνει πρόβλεψη για το σε ποια κλάση ανήκει η διαβρωμένη από θόρυβο πληροφορία που λαμβάνει στην είσοδο του.

3.2 RNN εξισωτής

Τα αναδρομικά ΤΝΔ είναι δίκτυα που χρησιμοποιούνται σε μη γραμμικά προβλήματα λόγω της μη γραμμικής φύσης τους. Θεωρούνται δυναμικά δίκτυα αφού περιέχουν μηχανισμό με τον οποίο ανατροφοδοτούνται. Έχουν αποδειχτεί ως μία πολύ καλή τεχνική εξίσωσης καναλιού. Το μπλοκ διάγραμμα ενός τηλεπικοινωνιακού συστήματος με RNN εξισωτή φαίνεται στο σχήμα 3.3. (Lavanaia et al. 2015)

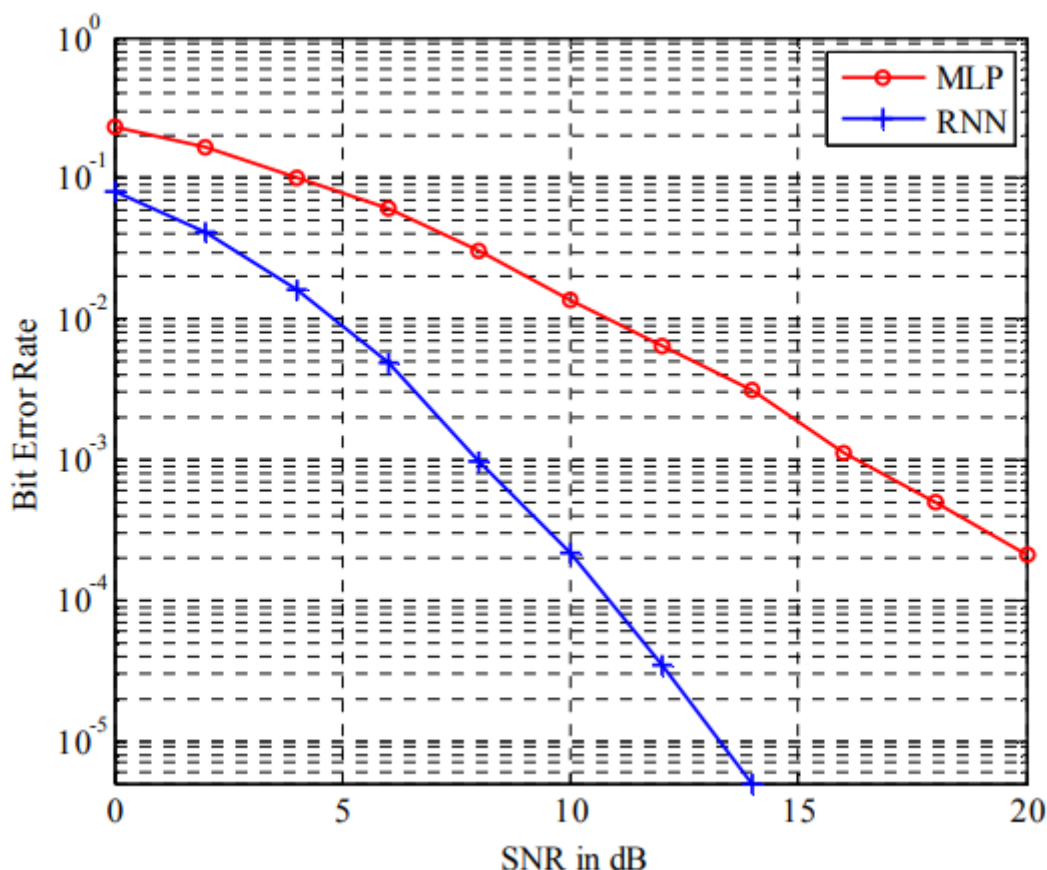


Σχήμα 3. 3: Εξισωτής RNN (Lavanaia et al. 2015)

Ο τρόπος με τον οποίο λειτουργεί είναι ως εξής, ο πομπός στέλνει μία γνωστή ακολουθία εκπαίδευσης (training sequence) στον δέκτη και μέσα από αυτή ο δέκτης αλλάζει τη δομή του προκειμένου να παράγει τα επιθυμητά αποτελέσματα στην έξοδο του. Όταν μιλάμε για τον δέκτη στη συγκεκριμένη περίπτωση, εννοείται ο εξισωτής καναλιού RNN αφού ο εξισωτής είναι υποσύστημα του δέκτη. Με αυτά που προαναφέρθηκαν για τα RNN, γίνεται κατανοητό ότι όσο ο πομπός στέλνει την ακολουθία πάνω στην οποία εκπαιδεύεται ο RNN εξισωτής, ταυτόχρονα ο εξισωτής αλλάζει τις τιμές των βαρών του προκειμένου να μπορέσει να δώσει τις επιθυμητές προβλέψεις. Αυτό γίνεται υπολογίζοντας το σφάλμα μεταξύ της πραγματικής και της επιθυμητής εξόδου. (Kechriotis, Zervas, and Manolakos 1994)

Σε αντίθεση με άλλους εξισωτές που χρησιμοποιούν feed-forward ΤΝΔ, ο RNN εξισωτής συνήθως έχει μικρό μέγεθος, δηλαδή αποτελείται από μικρό αριθμό νευρώνων με καλύτερη επίδοση.

Οι Lavania, Matey, Kumam, Bagadi και Anneru έδειξαν ότι ένας RNN εξισωτής έχει πολύ καλύτερη απόδοση από έναν MLP εξισωτή στην περίπτωση που καλούνται να εξισώσουν ένα μη γραμμικό κανάλι. Η διαφορά στην επίδοσή τους φαίνεται στο σχήμα 3.4 (Lavania et al. 2015).



Σχήμα 3. 4: Σύγκριση απόδοσης RNN και MLP εξισωτών για μη γραμμικό κανάλι (Lavania et al. 2015)

Στο σχήμα 3.4 ο RNN εξισωτής που συμβολίζεται από τη μπλε γραμμή φαίνεται να έχει πολύ καλύτερη απόδοση από τον MLP εξισωτή αφού η απόδοσή του στα 10dB είναι ίδια με αυτή του MLP εξισωτή στα 20dB. Αυτό έρχεται και συμφωνία με αυτά που προαναφέρθηκαν για τον DFE που είναι ένας εξισωτής με ανατροφοδότηση που χρησιμοποιείται για μη γραμμικά κανάλια και έχει καλύτερη απόδοση από τους γραμμικούς εξισωτές. Έτσι και εδώ το ΤΝΔ με ανατροφοδότηση έχει καλύτερη απόδοση από το ΤΝΔ χωρίς ανατροφοδότηση.

3.2.1 Αλγόριθμος Real Time Recurrent Learning (RTRL)

Ο αλγόριθμος RTRL είναι ένας από τους πιο διαδεδομένους αλγόριθμους που χρησιμοποιείται για την εκπαίδευση RNN εξισωτών. Όπως υποδηλώνει και το όνομα του (αναδρομική εκμάθηση σε πραγματικό χρόνο), αλλάζει τις τιμές των βαρών του σε πραγματικό χρόνο. Αυτός ο τρόπος λειτουργίας έχει το πλεονέκτημα ότι δεν χρειάζεται να τεθούν όρια στις εποχές που θα χρειαστεί το ΤΝΔ για να φτάσει το επιθυμητό αποτέλεσμα σφάλματος. Η αλλαγή των βαρών σε πραγματικό χρόνο έχει ως αποτέλεσμα ότι το μοντέλο δεν ακολουθεί πλέον την ακριβή αρνητική κλίση του συνολικού σφάλματος μέσω μίας τροχιάς. Αυτό συνεπάγεται μία πιθανότητα η τροχιά να επηρεαστεί από τις αλλαγές των βαρών σε πραγματικό χρόνο μέσω του αλγορίθμου και αυτό μπορεί να θεωρηθεί ως άλλη μία πηγή αρνητικής ανατροφοδότησης στο σύστημα. Το πρόβλημα μπορεί να αποφευχθεί αν επιλεχθεί μικρός ρυθμός μάθησης του ΤΝΔ το οποίο προϋποθέτει ότι ο χρόνος με τον οποίο γίνονται οι αλλαγές των βαρών πρέπει να είναι πολύ μεγαλύτερος από αυτόν της λειτουργίας του ΤΝΔ. Με άλλα λόγια τα βάρη θα αλλάζουν αρκετά πιο αργά από ότι δουλεύει το ΤΝΔ (Williams and Zipser 1989)

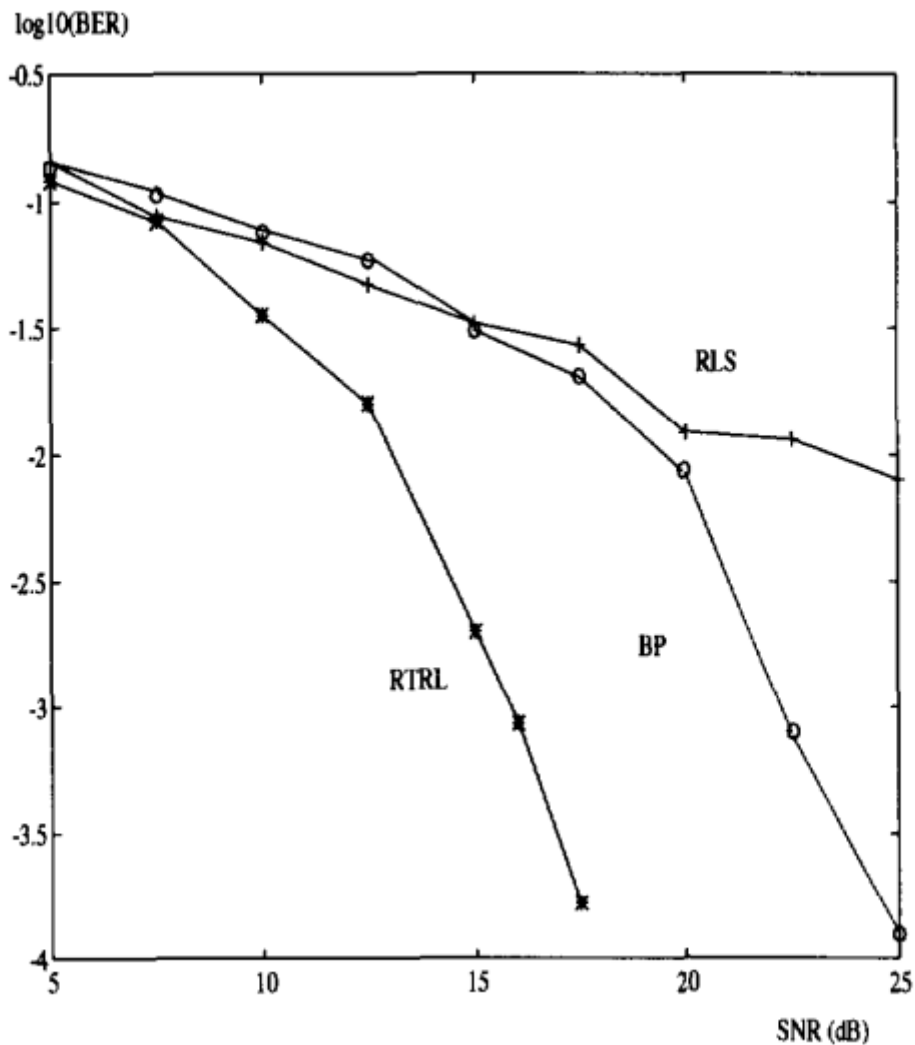
Η εξίσωση με την οποία αλλάζουν τα βάρη δίνεται ως:

$$\Delta w_{ij}(t) = \alpha \sum_{k \in U} e_k(t) p_{ij}^k(t) \quad (3.1)$$

Όπου $e_k(t)$ εκφράζει το σφάλμα μεταξύ πραγματικής και επιθυμητής εξόδου και $p_{ij}^k(t)$ είναι ένα δυναμικό σύστημα με αρχικές συνθήκες $p_{ij}^k(t_0) = 0$. Περιγράφεται αναλυτικά από τους Williams και Zipser στο (Williams and Zipser 1989).

Οι Kechriotis, Zervas, Manolakos έδειξαν ότι ένας RNN εξισωτής που χρησιμοποιεί τον αλγόριθμο RTRL έχει πολύ καλύτερη επίδοση από έναν κλασικό γραμμικό εξισωτή όπως φαίνεται στο σχήμα 3.5 (Kechriotis, Zervas, and Manolakos 1994).

Το σχήμα 3.5 αφορά την εξίσωση ενός μη γραμμικού καναλιού. Η γραμμή «RTRL» αφορά τον RNN εξισωτή, η γραμμή «BP» αφορά έναν feedforward εξισωτή που χρησιμοποιεί τον αλγόριθμο Backpropagation ενώ η «RLS» αφορά έναν γραμμικό εξισωτή που χρησιμοποιεί τον αλγόριθμο «Recursive least squares». Ο ίδιος εξισωτής RNN εμφανίζει καλύτερη επίδοση και σε γραμμικά κανάλια αλλά η πραγματική διαφορά στην επίδοση είναι όταν το κανάλι είναι μη γραμμικό.

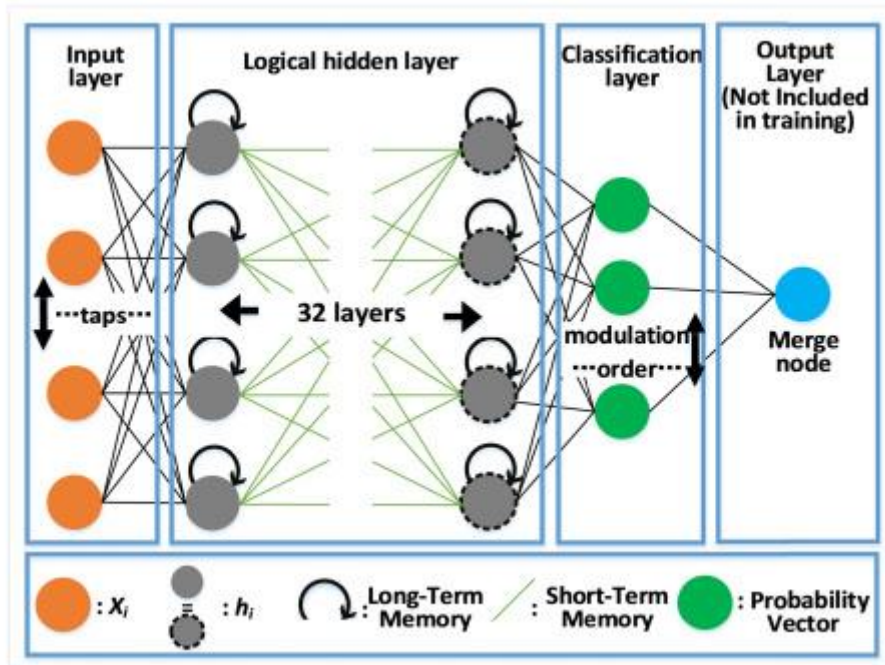


Σχήμα 3. 5: Σύγκριση BER μεταξύ RNN εξισωτή, γραμμικού εξισωτή και feedforward NN εξισωτή (Kechriotis, Zervas, and Manolakos 1994).

3.2.2 Long short-term memory ΤΝΔ εξισωτής

Το LSTM (long short-term memory) ΤΝΔ είναι ένα είδος RNN το οποίο δημιουργήθηκε ειδικά για την επεξεργασία χρονοσειρών και για ακολουθίες δεδομένων. Το LSTM έχει καλύτερη απόδοση από το κλασικό RNN σε περιπτώσεις που το ΤΝΔ χρειάζεται μεγαλύτερης διάρκειας μνήμη. Δηλαδή για την πληροφορία που δέχεται τη χρονική στιγμή t , δεν επαρκεί η ανατροφοδότηση της χρονικής στιγμής $t-1$ και το ΤΝΔ πρέπει να ανατρέξει σε ακόμα παλαιότερες χρονικές στιγμές προκειμένου να εξισώσει σωστά (Wang et al. 2020). Προκειμένου μία μονάδα LSTM να λειτουργήσει με αυτό τον τρόπο χρησιμοποιεί κάποιες μονάδες που ονομάζονται πύλες (gate units) οι οποίες αποφασίζουν για τη ροή της πληροφορίας μέσα στην μονάδα (είσοδος, έξοδος, διαγραφή). (Hochreiter and Schmidhuber 1997).

Στο σχήμα 3.6 φαίνεται ένας ΤΝΔ εξισωτής που χρησιμοποιεί LSTM. Φαίνεται ότι ο εξισωτής αποτελείται από ένα στρώμα εισόδων, κάποια κρυμμένα στρώματα, ένα στρώμα ταξινόμησης και ένα στρώμα εξόδου (X. Lu et al. 2019).



Σχήμα 3. 6: LSTM εξισωτής (X. Lu et al. 2019)

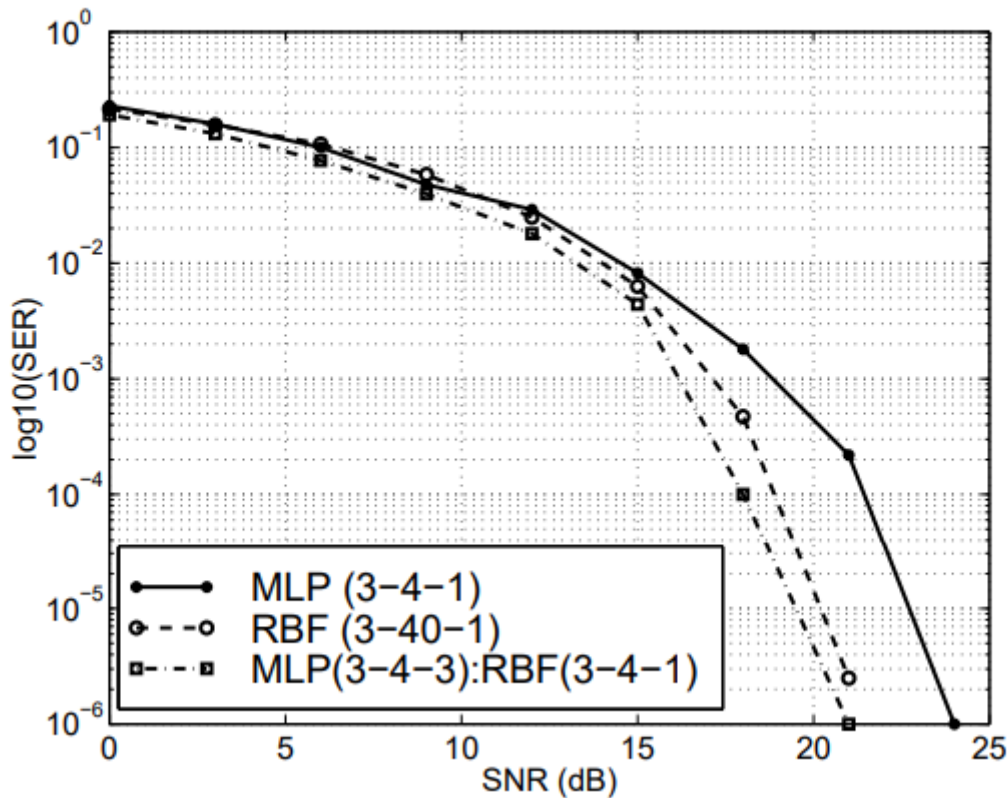
3.3 MLP-RBF εξισωτής

Πέρα από το MLP, τα RBF είναι άλλο ένα ΤΝΔ πρόσθιας τροφοδότησης που μπορεί να χρησιμοποιηθεί για εξίσωση καναλιού. Όπως ο εξισωτής MLP, έτσι και ο RBF εξισωτής ξεπερνάει τον γραμμικό εξισωτή σε απόδοση. Σε αυτούς τους εξισωτές ο αριθμός νευρώνων αυξάνεται εκθετικά με το μήκος του FIR μοντέλου του καναλιού το οποίο καλούνται να εξισώσουν. Όταν πρόκειται για χαμηλές τιμές SNR οι εξισωτές RBF πρέπει να αποτελούνται από αρκετά περισσότερους νευρώνες στο κρυμμένο στρώμα τους από ότι αποτελούνται τα MLP, προκειμένου να έχουν παρόμοια απόδοση με αυτά (B. Lu and Evans 1999).

Σε χαμηλό SNR, τα δεδομένα που έχει στείλει ο πομπός είναι διάσπαρτα λόγω του έντονου προσθετικού θορύβου. Τα δεδομένα τα οποία δεν ανήκουν σε καμία ομάδα από αυτές που έχει δημιουργήσει το RBF προκειμένου να τα κατηγοριοποιήσει θεωρούνται μη σχετικά. Αυτά τα ίδια μη σχετικά δεδομένα όμως έχουν τη δυνατότητα να δημιουργήσουν σοβαρό πρόβλημα στην απόδοση του RBF σε χαμηλό SNR. Το ίδιο όμως δεν συμβαίνει και στα MLP τα οποία μπορούν να παρουσιάσουν προβλήματα σε άλλα σημεία στα οποία τα RBF δεν παρουσιάζουν. Για αυτόν τον λόγο και προκειμένου να βελτιωθεί η απόδοση ενός εξισωτή που χρησιμοποιεί ΤΝΔ πρόσθιας τροφοδότηση, είναι δυνατή η χρησιμοποίηση και των δύο (B. Lu and Evans 1999).

Για να γίνει αυτό, ο αριθμός νευρώνων εισόδων και εξόδων του MLP πρέπει να είναι ίδιος με τον αριθμό των taps του FIR μοντέλου του καναλιού. Το RBF θα συνδέεται σε αλληλουχία με το MLP οπότε θα πρέπει, για να συμβαδίζει με το MLP, να έχει τον ίδιο αριθμό νευρώνων εισόδου με τον αριθμό νευρώνων εξόδου του MLP, ενώ η έξοδος του RBF θα είναι η τελική πρόβλεψη άρα θα είναι μόνο μία. Έτσι το MLP λαμβάνει την πληροφορία και αντιμετωπίζει το πρόβλημα που δημιουργούν τα μη σχετικά δεδομένα προκειμένου το RBF να λάβει τις εξόδους του και να μπορέσει να φτάσει σε τελική πρόβλεψη (B. Lu and Evans 1999).

Οι Lu και Evans πραγματοποίησαν αυτή την υλοποίηση και η απόδοση του MLP-RBF εξισωτή συγκριτικά με έναν MLP και έναν RBF εξισωτή φαίνεται στο σχήμα 3.7.



Σχήμα 3. 7: Σύγκριση επίδοσης του MLP-RBF εξισωτή με έναν MLP και έναν RBF (B. Lu and Evans 1999)

Φαίνεται ότι το σφάλμα πρόβλεψης του εξισωτή που χρησιμοποιεί το MLP διαδοχικά με το RBF είναι καλύτερη σε σχέση με τις μεμονωμένες εκδοχές του. Μπορεί η διαφορά στην απόδοση του MLP-RBF με το RBF να μην είναι τόσο μεγάλη αλλά υπάρχει πολύ σημαντική διαφορά στην πολυπλοκότητα των δύο εξισωτών. Το RBF για να φτάσει τη συγκεκριμένη απόδοση χρειάζεται να χρησιμοποιήσει 40 νευρώνες στο κρυφό στρώμα. Πρόκειται για ένα γραμμικό κανάλι με 3 taps, οπότε οι εισοδοί, εξοδοί του MLP και οι εισοδοί του RBF είναι και αυτές 3 όπως προαναφέρθηκε.

4.1 Πείραμα 1: Εξίσωση AWGN καναλιού περιορισμένου εύρους ζώνης με MLP και RNN εξισωτή

4.1.1 Στόχος 1^ο πειράματος

Σ' αυτό το πείραμα χρησιμοποιούνται διαφορετικές δομές MLP εξισωτών και στη συνέχεια ένας RNN εξισωτής για την εξίσωση του καναλιού. Το κανάλι είναι περιορισμένου εύρους ζώνης και οι εξισωτές καλούνται να αντιμετωπίσουν το πρόβλημα της ISI. Η υλοποίηση γίνεται μέσω του λογισμικού MATLAB και τα δεδομένα εκπαίδευσης και δοκιμής της επίδοσης παράγονται τυχαία. Οι εξισωτές καλούνται να κατηγοριοποιήσουν τα αλλοιωμένα από ISI και AWGN δεδομένα σε δύο κλάσεις. Επίσης θα χρησιμοποιηθούν διαφόρων ειδών κλασικοί γραμμικοί εξισωτές και η επίδοση τους θα συγκριθεί με αυτή των MLP και RNN εξισωτών.

4.1.2 Πειραματικό μέρος 1^ο πειράματος

Το συγκεκριμένο πείραμα εκτελέστηκε ως προσομοίωση στο λογισμικό MATLAB 2021a με την εργαλειοθήκη «Deep learning toolbox» και με την εργαλειοθήκη επικοινωνιών «Communications toolbox». Χρησιμοποιείται BPSK διαμόρφωση. Τα τυχαίως παραγόμενα δεδομένα που είναι 1 ή -1 περνάνε μέσα από γραμμικό AWGN κανάλι με περιορισμένο εύρος ζώνης. Από την έξοδο του καναλιού καταλήγουν στην είσοδο του MLP εξισωτή ως είσοδοι του.

Η συνάρτηση μεταφοράς του καναλιού όπως φαίνεται στην εξίσωση 4.1 αποτελείται από 3 taps. Αυτό σημαίνει ότι ο MLP εξισωτής θα χρειαστεί πολλαπλές εισόδους προκειμένου να εξισώσει το κανάλι και να προβλέψει με επιτυχία το σήμα. Δοκιμάστηκαν πολλές διαφορετικές περιπτώσεις σχετικά με τις εισόδους του MLP εξισωτή αλλά παρατηρήθηκε ότι για πάνω από 3 εισόδους τα αποτελέσματα δεν άλλαξαν. Οι είσοδοι κατανέμονται σε έναν πίνακα για την ομαδοποίηση τους και ολισθαίνουν κατά μία θέση σε κάθε χρονική στιγμή προκειμένου να αντιμετωπισθεί η συνέλιξη.

Για παράδειγμα, αν το διάνυσμα $r = [0.6255 \ 0.3920 \ -0.0119 \ 0.2886]$ απεικονίζει το λαμβανόμενο στο δέκτη σήμα στην έξοδο του αποδιαμορφωτή τότε οι διαδοχικές είσοδοι του MLP είναι:

$$r' = \begin{bmatrix} -0.0119 & 0.2886 \\ 0.3920 & -0.0119 \\ 0.6255 & 0.3920 \end{bmatrix}$$

Με την πρώτη στήλη να είναι η είσοδος του MLP την πρώτη χρονική στιγμή και η δεύτερη στήλη να είναι η είσοδος του MLP τη δεύτερη χρονική στιγμή ενώ η κάθε στήλη αντιστοιχεί και σε ένα σύμβολο, δηλαδή για κάθε στήλη θα υπάρχει και μία πρόβλεψη. Την ίδια λογική ακολουθεί και ο RNN εξισωτής.

Η συνάρτηση μεταφοράς του καναλιού είναι:

$$H(z) = 0.407 + 0.815z^{-1} + 0.407z^{-2} \quad (4.1)$$

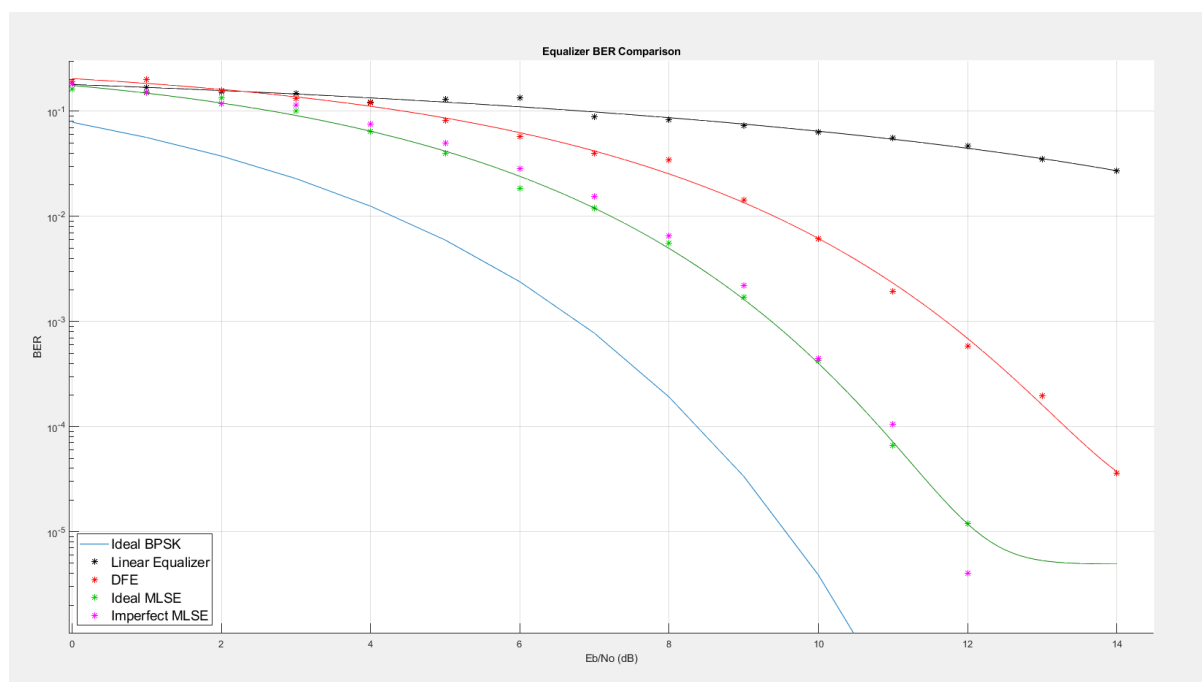
Οι παράμετροι της προσομοίωσης φαίνονται συνοπτικά στην επόμενη σελίδα στον πίνακα 4.2 και οι κώδικες που χρησιμοποιήθηκαν για τους MLP εξισωτές δίνονται στο παράρτημα Α. Ο κώδικας για τις κλασικές μεθόδους εξισωτών στο παράρτημα Β. Ο κώδικας για τον RNN εξισωτή στο παράρτημα Γ.

Πίνακας 4. 1: Παράμετροι 2^{ης} προσομοίωσης

Παράμετροι	Τιμές
Είδος διαμόρφωσης	BPSK
Εύρος SNR	0 dB έως 14 dB
Κανάλι	AWGN περιορισμένου εύρους ζώνης $h(n)=\{0.407, 0.815, 0.407\}$
Δομή MLP 3	{3}
Δομή MLP 5	{5}
Δομή MLP 10	{10}
Δομή MLP 5:10	{5, 10}
Δομή MLP 10:10	{10, 10}
Δομή MLP 15:10	{15 10}
Δομή RNN 10	{10}
Αριθμός δειγμάτων εκπαίδευσης	1000
Αριθμός δειγμάτων δοκιμής	1000000
Συνάρτηση ενεργοποίησης	Σιγμοειδής

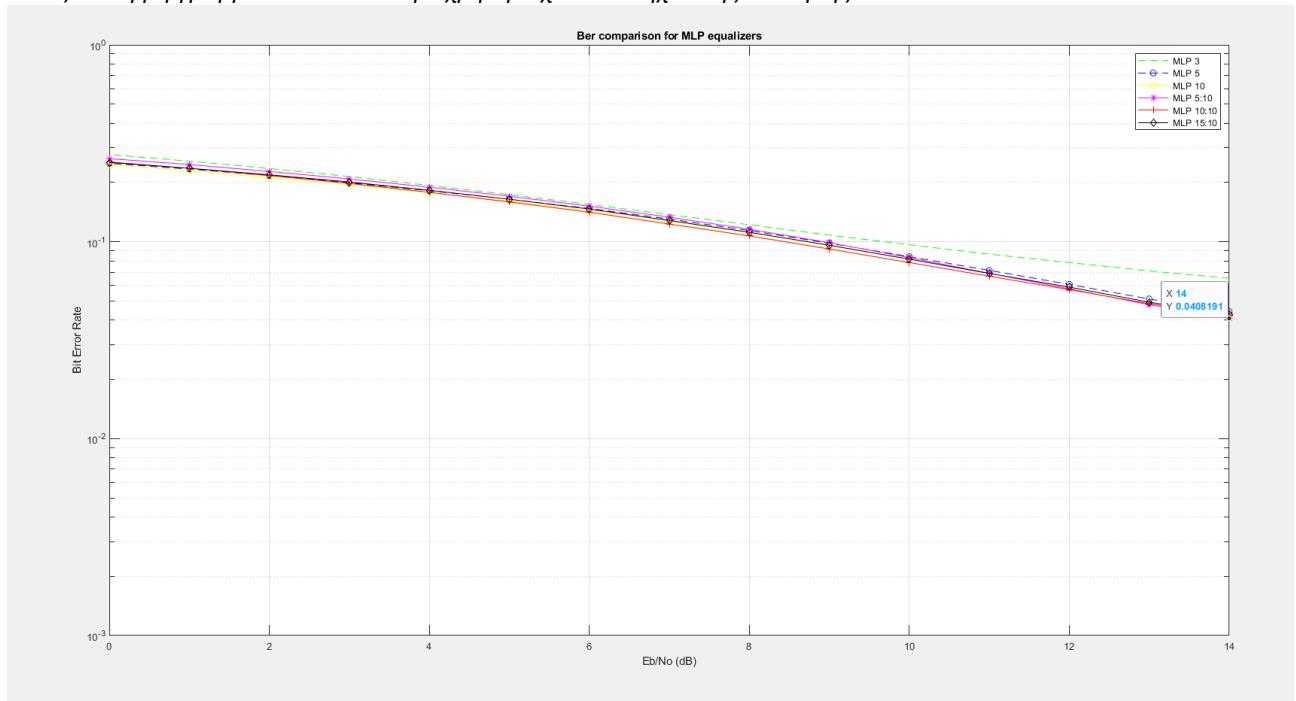
4.1.3 Αποτελέσματα και συμπεράσματα 1^ο πειράματος

Αρχικά στο σχήμα 4.3 φαίνονται οι επιδόσεις κάποιων κλασικών τεχνικών εξίσωσης καναλιού. Η μαύρη γραμμή απεικονίζει την απόδοση ενός γραμμικού εξισωτή ενώ η κόκκινη γραμμή είναι η επίδοση ενός DFE εξισωτή. Έπειτα η πράσινη γραμμή είναι ο ιδανικός εξισωτής MLSE ενώ τα ροζ σημεία είναι η επίδοση ενός μη ιδανικού MLSE εξισωτή. Αυτά τα αποτελέσματα είναι αναμενόμενα και συμβαδίζουν με τη θεωρία καθώς ο γραμμικός εξισωτής έχει τη χειρότερη επίδοση σε σχέση με τους άλλους και ο MLSE την καλύτερη επίδοση. Φαίνεται ότι η διαφορά της επίδοσης των εξισωτών αυξάνεται όσο μεγαλώνει το SNR καθώς στα μικρότερα SNR δεν έχουν τόσο μεγάλη διαφορά όσο στα μεγαλύτερα. Τέλος η συνεχής μπλε γραμμή που φαίνεται να έχει το μικρότερο σφάλμα είναι η ιδανική μετάδοση BPSK. Προφανώς η διαφορά σφάλματος μεταξύ των εξισωτών και της ιδανικής BPSK είναι αρκετά μεγάλη. Αξιοσημείωτο είναι ότι ο μη ιδανικός εξισωτής έχει καλύτερη επίδοση από τον ιδανικό MLSE εξισωτή στα 12 dB. Αυτό το μέρος του πειράματος πραγματοποιήθηκε για μία πρώτη εικόνα της πειραματικής επίδοσης κλασικών εξισωτών αλλά κυρίως προκειμένου να συγκριθούν αυτά τα αποτελέσματα με τα αποτελέσματα ενός MLP και ενός RNN εξισωτή τα οποία φαίνονται στο σχήμα 4.2 και 4.4 αντίστοιχα.



Σχήμα 4. 1: Διάγραμμα BER-SNR για κλασικές τεχνικές εξίσωσης καναλιού.

Προκειμένου να βρεθεί ο ιδανικός MLP εξισωτής δοκιμάστηκαν διάφορες δομές όπως αναφέρθηκαν στον πίνακα 4.1 και τα αποτελέσματα τους φαίνονται στο σχήμα 4.2.

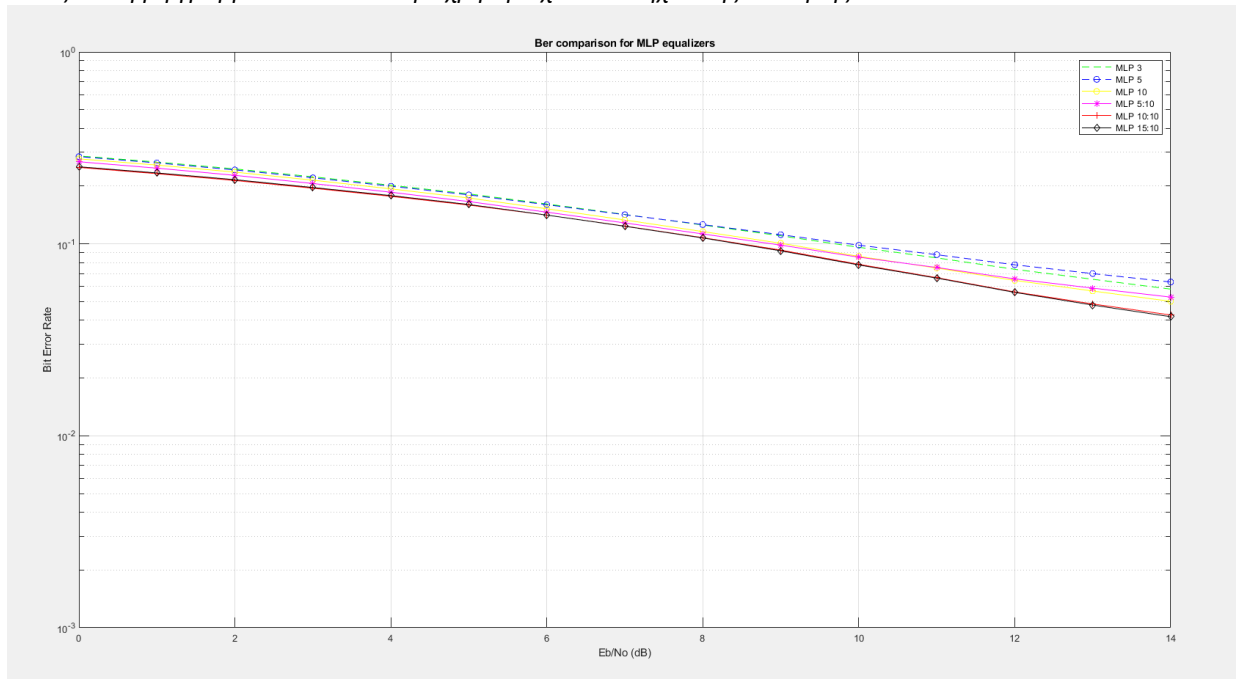


Σχήμα 4. 2: Διάγραμμα BER-SNR για MLP εξισωτές.

Από το σχήμα 4.4 φαίνεται ότι όλοι οι εξισωτές έχουν σχεδόν την ίδια επίδοση με εξαίρεση τον MLP εξισωτή που χρησιμοποιεί μόνο 3 νευρώνες ο οποίος έχει μία μικρή διαφορά με τους άλλους. Στο στάδιο της εκπαίδευσης των νευρωνικών έγιναν πολλές δοκιμές και παρατηρήθηκε ότι η δομή των MLP εξισωτών δεν έχει μεγάλη σημασία καθώς όλες οι δομές μπορούν να αποφέρουν την ίδια επίδοση εφόσον εκπαιδευτούν σωστά. Παρόλα αυτά τα πιο πολύπλοκα και πιο βαθιά ΤΝΔ έδειξαν καλύτερη επίδοση στις περισσότερες δοκιμές. Δηλαδή ενώ οι πιο ρηχοί MLP εξισωτές είναι ικανοί να έχουν ακριβώς την ίδια επίδοση, αν δεν εφαρμοστούν ορισμένες συνθήκες στην εκπαίδευση τους, φαίνεται να είναι ελαφρώς κατώτεροι. Η εξαντλητική και με συγκεκριμένες συνθήκες εκπαίδευση απαιτεί μεγαλύτερη υπολογιστική ισχύ καθώς έχει μεγαλύτερη υπολογιστική πολυπλοκότητα.

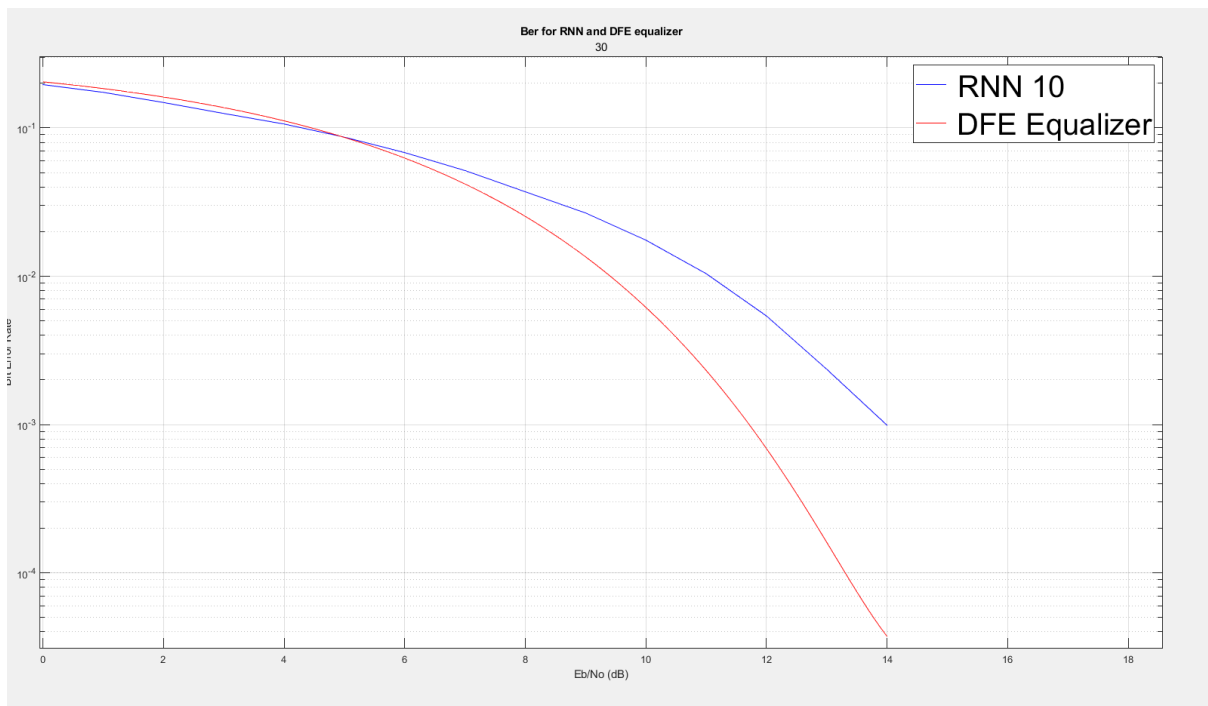
Ενώ τα αποτελέσματα του συγκεκριμένου πειράματος είναι αρκετά ενθαρρυντικά, οι MLP εξισωτές δεν καταφέρνουν να φτάσουν την επίδοση του γραμμικού εξισωτή με τη διαφορά μεταξύ των BER να είναι όμως πολύ μικρή όπως φαίνεται από τη σύγκριση των τιμών BER στα 14 dB. Η διαφορά ανέρχεται στην τιμή του $0.0408 - 0.0271 = 0.0137$ dB. Η διαφορά τους είναι σχεδόν ασήμαντη αλλά βάσει της θεωρίας ο MLP εξισωτής θα έπρεπε να έχει καλύτερη απόδοση από τον γραμμικό εξισωτή. Άμα υπήρχε η δυνατότητα μεγαλύτερης υπολογιστικής ισχύς τότε ο κάθε MLP εξισωτής θα είχε εκπαιδευτεί για κάθε SNR ξεχωριστά και σίγουρα θα εμφάνιζε μία καλύτερη συνολική συμπεριφορά. Παρόλα αυτά όμως δεν σημαίνει ότι οι συγκεκριμένοι MLP εξισωτές δεν μπορούν να χρησιμοποιηθούν για εξίσωση καναλιού καθώς το σφάλμα τους είναι αρκετά μικρό και αρκετά κοντά στον κλασικό γραμμικό εξισωτή.

Στο σχήμα 4.5 φαίνεται μια υποβέλτιστη περίπτωση εκπαίδευσης των ίδιων MLP εξισωτών. Συγκρίνοντας τα δύο διαγράμματα μπορεί κανείς να φτάσει στο συμπέρασμα ότι το πρόβλημα στην επίδοση των MLP εξισωτών σε σχέση με τον γραμμικό εξισωτή έχει μεγάλη πιθανότητα να βρίσκεται στον τρόπο εκπαίδευσης.



Σχήμα 4. 3: Διάγραμμα BER-SNR MLP εξισωτών για υποδεέστερη εκπαίδευση.

Στο σχήμα 4.4 φαίνεται ότι η επίδοση του RNN εξισωτή είναι εμφανώς καλύτερη από αυτή του MLP εξισωτή και του γραμμικού εξισωτή. Σε σχέση με τον DFE εξισωτή παρατηρείται καλύτερη επίδοση στα χαμηλά dB αλλά εμφανή διαφορά στα υψηλά dB καθώς ο DFE έχει μικρότερο σφάλμα, περίπου 10^{-1} στα 14 dB. Πρόκειται για έναν RNN εξισωτή που χρησιμοποιεί ένα κρυφό στρώμα με 10 κόμβους. Δοκιμάστηκαν και άλλες δομές RNN εξισωτή αλλά δεν είχαν διαφορετικά αποτελέσματα.



Σχήμα 4. 4: Διάγραμμα BER-SNR για RNN εξισωτή.

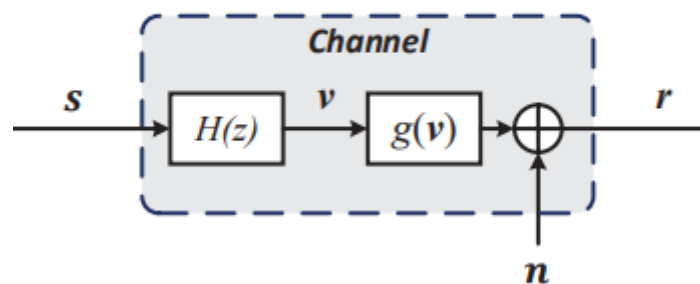
4.2 Πείραμα 2: Εξίσωση μη γραμμικού AWGN καναλιού περιορισμένου εύρους ζώνης με MLP εξισωτή

4.2.1 Στόχος 2^ο πειράματος

Το πείραμα ακολουθεί την ίδια λογική με το προηγούμενο, δηλαδή δοκιμάζονται και πάλι διάφορες δομές MLP εξισωτών και ένας RNN εξισωτής για την εξίσωση καναλιού. Τα ΤΝΔ είναι δίκτυα με τα οποία δεν είναι πάντα πολύ απλό να προσεγγισθούν μαθηματικά προβλήματα όπως στο προηγούμενο πείραμα. Η ισχύς τους κρύβεται στην επίλυση μη γραμμικών προβλημάτων λόγω των συναρτήσεων ενεργοποίησης που χρησιμοποιούν. Για αυτό σε αυτό το πείραμα οι εξισωτές θα καλεστούν να εξισώσουν ένα μη γραμμικό AWGN κανάλι περιορισμένου εύρους ζώνης. Σαφής στόχος ο σχολιασμός της επίδοσης τους για το πως αντιμετωπίζουν τον θόρυβο και την ISI με τη μη γραμμικότητα. Το προηγούμενο πείραμα μπορεί να θεωρηθεί προστάδιο του συγκεκριμένου πειράματος.

4.2.2 Πειραματικό μέρος 2^ο πειράματος

Όπως στην προηγούμενη περίπτωση έτσι και σ' αυτήν την προσομοίωση η υλοποίηση έγινε μέσω του MATLAB 2021a με την εργαλειοθήκη «Deep learning toolbox». Η λογική είναι ακριβώς η ίδια, δηλαδή το σήμα που παράγεται τυχαία σε 1 ή -1 (BPSK διαμόρφωση) περνάει μέσα από το κανάλι και έπειτα η έξοδος του καναλιού γίνεται η είσοδος του MLP εξισωτή. Η μεθοδολογία για τη δημιουργία εισόδων για τον εξισωτή είναι ακριβώς η ίδια όπως παρουσιάστηκε στην υποενότητα 4.1.2. Στην εξίσωση 4.2 δίνεται η σχέση που κάνει το κανάλι μη γραμμικό και στο σχήμα 4.5 φαίνεται η αρχιτεκτονική του καναλιού. Στο παράρτημα Δ και Ε βρίσκονται οι κώδικες για τον MLP και RNN εξισωτή αντίστοιχα.



Σχήμα 4. 5: Μη γραμμικό κανάλι (Xu et al. 2018).

$$g(v) = v - 0.9v^3 \quad (4.2)$$

Το v συμβολίζει το σύμβολο στην έξοδο του γραμμικού τμήματος του καναλιού. Αφού υλοποιηθεί η μη γραμμική σχέση της εξίσωσης 4.2 τότε προστίθεται στο $g(v)$ προσθετικός θόρυβος και η έξοδος του καναλιού r κατηγοριοποιείται έτσι όπως αναφέρθηκε στην υποενότητα 4.1.2 για να μπορέσει ο MLP εξισωτής να λύσει το πρόβλημα ταξινόμησης. Η κρουστική απόκριση του γραμμικού τμήματος του καναλιού φαίνεται στην εξίσωση 4.1. Στον πίνακα 4.2 φαίνονται συνοπτικά οι παράμετροι της 2^{ης} προσομοίωσης.

Πίνακας 4. 2: Παράμετροι 3^{ης} προσομοίωσης

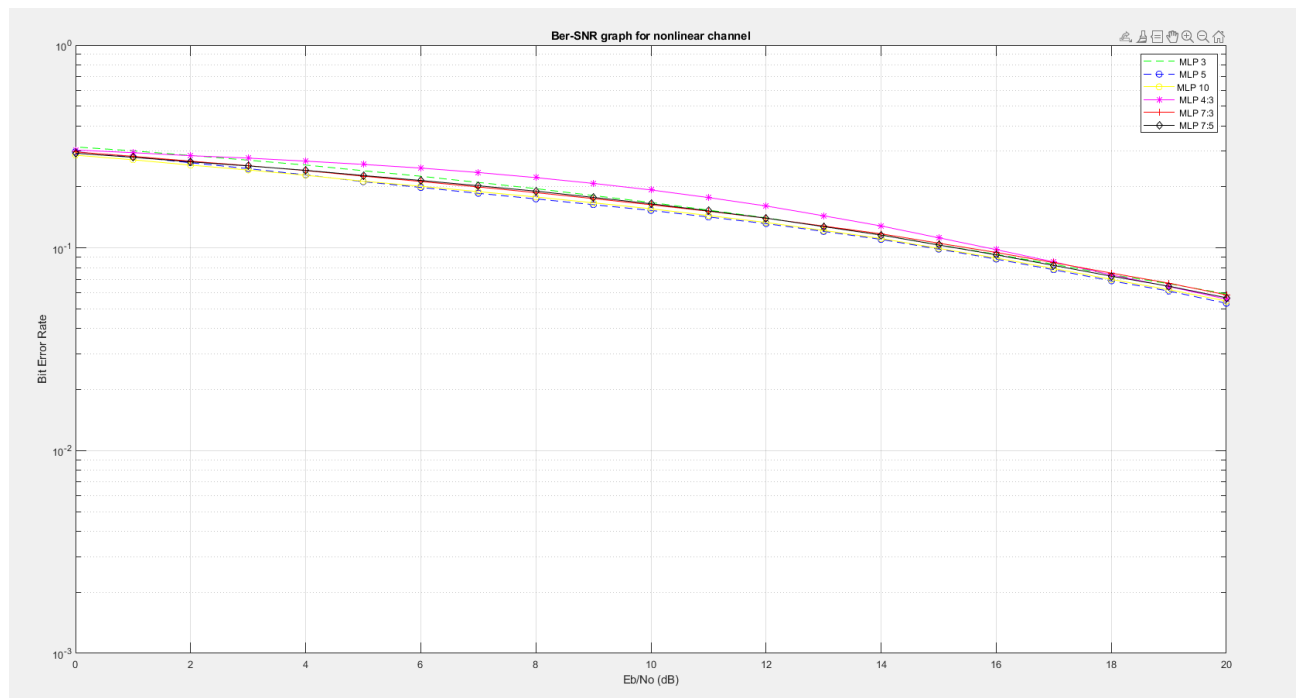
Παράμετροι	Τιμές
Είδος διαμόρφωσης	BPSK
Εύρος SNR	0 dB έως 20 dB
Κανάλι	Μη γραμμικό AWGN περιορισμένου εύρους ζώνης
Δομή MLP 3	{3}
Δομή MLP 5	{5}
Δομή MLP 10	{10}
Δομή MLP 4:3	{4, 3}
Δομή MLP 7:3	{7, 3}
Δομή RNN 10	{10}
Δομή MLP 7:5	{7, 5}
Αριθμός δειγμάτων εκπαίδευσης	1000
Αριθμός δειγμάτων δοκιμής	1000000
Συνάρτηση ενεργοποίησης	Σιγμοειδής

4.2.3 Αποτελέσματα και συμπεράσματα 2^ο πειράματος

Και σε αυτό το πείραμα φαίνεται ότι η δομή των MLP εξισωτών δεν έχει σημασία στην επίδοση τους και πιο μεγάλο ρόλο παίζει η σωστή διαδικασία εκπαίδευσης καθώς όλοι οι εξισωτές είναι ικανοί να δώσουν την ίδια επίδοση όπως φαίνεται στο σχήμα 4.6. Στο συγκεκριμένο πείραμα το εύρος SNR είναι 0 dB – 20 dB. Η επίδοση έχει πέσει αισθητά συγκριτικά με το προηγούμενο πείραμα αλλά αυτό είναι κάτι αναμενόμενο καθώς το κανάλι πλέον είναι πιο δύσκολο να εξισωθεί αφού προστέθηκε η μη γραμμικότητα. Παρόλα αυτά τα αποτελέσματα είναι αρκετά φυσιολογικά και κοντά σε αυτά του προηγούμενου πειράματος με μία μικρή πτώση στην επίδοση λόγω της μη γραμμικότητας. Κατά το στάδιο της εκπαίδευσης παρατηρήθηκε ότι οι MLP εξισωτές εκπαιδευόντουσαν με μεγαλύτερη ευκολία και καλύτερη επίδοση για υψηλές τιμές SNR. Στη συγκεκριμένη προσομοίωση τα ΤΝΔ εκπαιδεύτηκαν στα 30 dB.

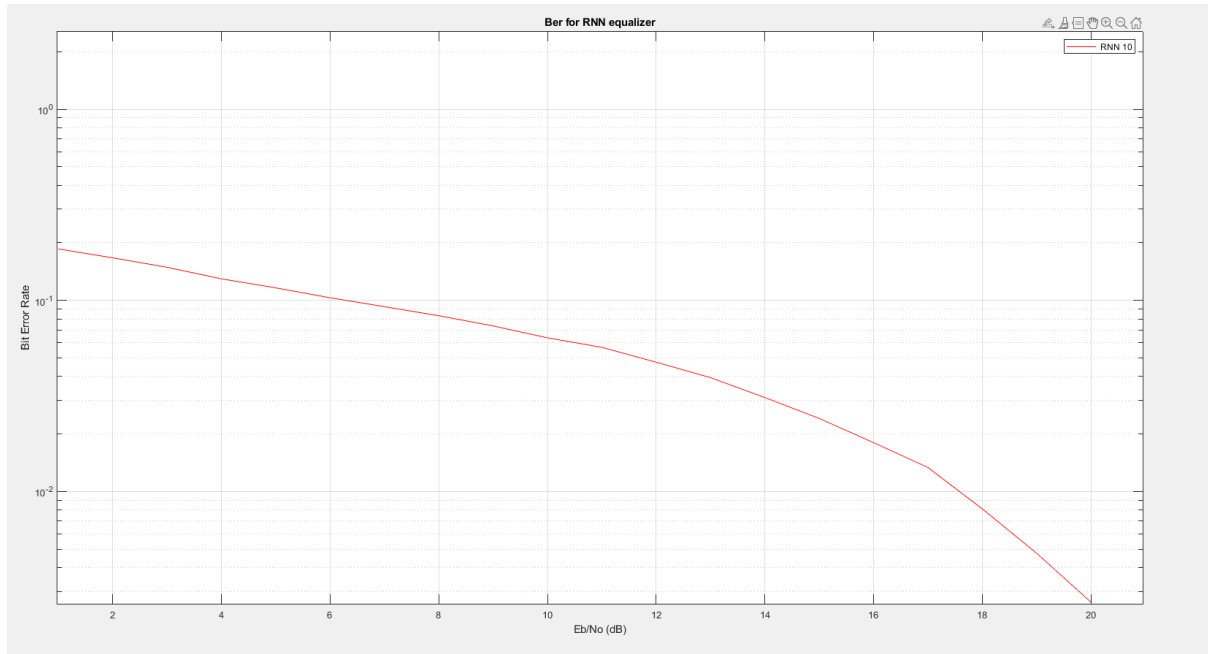
Παρόλα αυτά στη συγκεκριμένη προσομοίωση ο εξισωτής MLP 5 φαίνεται να έχει την καλύτερη επίδοση με τον εξισωτή MLP 10 να έρχεται δεύτερος. Ο εξισωτής MLP 4:3 φαίνεται να είναι λίγο χειρότερος καθώς μέχρι να φτάσει στα 15 dB έχει μια εμφανώς χειρότερη επίδοση αλλά μετά τα 15 dB η επίδοση του βελτιστοποιείται αρκετά και στα 20 dB φαίνεται να είναι υποδεέστερος μόνο του MLP 5.

Γενικότερα είναι σαφές ότι ένα απλό ρηχό ΤΝΔ μπορεί να εκτελέσει την εξίσωση καναλιού και τα πολλαπλά στρώματα δεν κάνουν καμία διαφορά. Οπότε τα ρηχά ΤΝΔ θα πρέπει να προτιμηθούν καθώς δεν υπάρχει ανάγκη αφού το πρόβλημα θεωρείται χαμηλής υπολογιστικής πολυπλοκότητας και δεν υπάρχει λόγος να χρησιμοποιηθούν πιο πολύπλοκα ΤΝΔ.



Σχήμα 4. 6: Διάγραμμα BER-SNR για MLP εξισωτές σε μη γραμμικό κανάλι.

Ο RNN εξισωτής έχει πάλι αρκετά καλύτερη επίδοση από τον MLP εξισωτή όπως φαίνεται στο σχήμα 4.7. Η διαφορά τους φαίνεται σε όλες τις τιμές των dB. Όπως στο προηγούμενο πείραμα, έτσι και σε αυτό ο RNN εξισωτής αποτελείται από ένα κρυφό στρώμα με 10 κόμβους. Άξιο αναφοράς ότι σε σχέση με τον MLP εξισωτή, ο RNN εξισωτής δείχνει αρκετά καλύτερη βελτίωση σφάλματος όσο μεγαλώνουν οι τιμές του SNR.



Σχήμα 4. 7: Διάγραμμα BER-SNR για RNN εξισωτή σε μη γραμμικό κανάλι

ΣΥΜΠΕΡΑΣΜΑΤΑ

Σε αυτή τη διπλωματική εργασία παρουσιάστηκαν διάφορες τεχνικές για τη λύση του προβλήματος της εξίσωσης καναλιού. Με την άνοδο των ΤΝΔ τα τελευταία χρόνια παρατηρήθηκε ότι αυτά μπορούν να συμβάλλουν σημαντικά στις τηλεπικοινωνίες μέσω της χρησιμοποίησης τους στους δέκτες τηλεπικοινωνιακών συστημάτων ως εξισωτές καναλιού. Δεν είναι δύσκολο να καταλάβει κανείς το λόγο που άρχισαν να γίνονται δοκιμές με ΤΝΔ εξισωτές καθώς ένα ΤΝΔ και ένας κλασικός εξισωτής μοιάζουν πολύ στη δομή τους. Το μόνο αρνητικό σχετικά με τη χρήση ΤΝΔ ως εξισωτές είναι ότι χρειάζονται μεγαλύτερη υπολογιστική ισχύ καθώς πρόκειται για μοντέλα μεγαλύτερης πολυπλοκότητας που χρειάζονται εκτενή εκπαίδευση.

Ο MLP εξισωτής είναι η πιο εύκολα υλοποιήσιμη περίπτωση εξισωτή βασισμένου σε ΤΝΔ, ο οποίος όμως κάποιες φορές χρειάζεται εκτενή εκπαίδευση. Τα ΤΝΔ πρόσθιας τροφοδότησης ταιριάζουν με τους γραμμικούς εξισωτές καθώς έχουν παρόμοια αρχιτεκτονική. Για αυτό εξισωτές όπως ο MLP ή ο RBF εξισωτής είναι ικανοί να προσφέρουν καλύτερη επίδοση από τους κλασικούς γραμμικούς εξισωτές ενώ είναι δυνατός ο συνδυασμός και των δύο αυτών ΤΝΔ για την υλοποίηση ενός MLP-RBF εξισωτή.

Οι εξισωτές που βασίζονται σε RNN ΤΝΔ έχουν σημαντικά καλύτερη επίδοση από αυτούς της πρόσθιας τροφοδότησης και η δομή της ταιριάζει με αυτή του DFE εξισωτή ο οποίος και αυτός έχει καλύτερη επίδοση από έναν γραμμικό εξισωτή. Είναι λοιπόν αναμενόμενο και ο RNN εξισωτής να υπερτερεί του MLP, RBF κλπ.. Πρόκειται για τη βέλτιστη περίπτωση εξίσωσης δύσκολων καναλιών, όπως μη γραμμικά κανάλια και κανάλια με διαλείψεις χωρίς αυτό να σημαίνει ότι δεν έχουν καλύτερη επίδοση και σε γραμμικά κανάλια. Γίνεται κατανοητό ότι οι RNN εξισωτές χρειάζονται μεγαλύτερη υπολογιστική ισχύ λόγω της πολυπλοκότητας που χρειάζονται για να εκπαιδευτούν σωστά.

Η διαφορά μεταξύ γραμμικού και DFE εξισωτή φάνηκε στις προσομοιώσεις που έγιναν. Παρόμοια διαφορά φάνηκε και στην επίδοση μεταξύ του MLP και RNN εξισωτή. Πρόκειται για ένα αναμενόμενο αποτέλεσμα αφού και τα δύο ΤΝΔ έχουν παρόμοια αρχιτεκτονική με τις αντίστοιχες κλασικές μεθόδους εξίσωσης καναλιού.

Η μελλοντική εργασία και έρευνα θα πρέπει να επικεντρωθεί γύρω από την επίδοση του DFE εξισωτή σε μη γραμμικό κανάλι και πως συγκρίνεται με αυτή του RNN εξισωτή. Στα πειράματα που υλοποιήθηκαν χρησιμοποιήθηκε διαμόρφωση BPSK. Μεγάλο ενδιαφέρον έχει η χρησιμοποίηση διαφορετικού είδους διαμόρφωσης, όπως QPSK ή QAM καθώς και η χρήση συνελκτικών νευρωνικών δικτύων για την ταύτοχρονη εξίσωση και αποκωδικοποίηση καναλιού.

Agostinelli, Forest, Matthew Hoffman, Peter Sadowski, and Pierre Baldi. 2015. “Learning Activation Functions to Improve Deep Neural Networks.” *ArXiv:1412.6830 [Cs, Stat]*, April. <http://arxiv.org/abs/1412.6830>.

Almási, Adela-Diana, Stanisław Woźniak, Valentin Cristea, Yusuf Leblebici, and Ton Engbersen. 2016. “Review of Advances in Neural Networks: Neural Design Technology Stack.” *Neurocomputing* 174 (January): 31–41. doi:10.1016/j.neucom.2015.02.092.

AL-Musawi, Raed, Ali Ali, Haider Alhattab, and Kadhum Al-Majdi. 2016. “Adaptive Channel Equalization Using Tuning-Free Non-Parametric NLMS with Variable Step Size,” June.

Apicella, Andrea, Francesco Donnarumma, Francesco Isgrò, and Roberto Prevete. 2021. “A Survey on Modern Trainable Activation Functions.” *Neural Networks* 138 (June): 14–32. doi:10.1016/j.neunet.2021.01.026.

Arslan, Guner, Brian Evans, and Sayfe Kiaei. 2000. “Optimum Channel Shortening for Discrete Multitone Transceivers.” In , 5:2965–68 vol.5. doi:10.1109/ICASSP.2000.861156.

Atapattu, Lakmali, Gayan Arachchige, K. Ziri-Castro, Hajime Suzuki, and Dhammika Jayalath. 2012. “Linear Adaptive Channel Equalization for Multiuser MIMO-OFDM Systems.” In , 1–5. doi:10.1109/atnac.2012.6398080.

Atef, Mohamed, and Horst Zimmermann. 2013. “Equalization Techniques.” In *Optical Communication over Plastic Optical Fibers: Integrated Optical Receiver Technology*, edited by Mohamed Atef and Horst Zimmermann, 23–40. Springer Series in Optical Sciences. Berlin, Heidelberg: Springer. doi:10.1007/978-3-642-30388-3_3.

Azurdia-Meza, Cesar. 2013. “Analysis, Design, and Implementation of Parametric Nyquist ISI-Free Pulses in Wireless Communications Systems.”

Baloch, Saba, Javed Baloch, and Mukhtiar Unar. 2012. “Channel Equalization Using Multilayer Perceptron Networks.” *Mehran University Research Journal of Engineering and Technology* 31 (July): 469–74.

Belfiore, C.A., and J.H. Park. 1979. “Decision Feedback Equalization.” *Proceedings of the IEEE* 67 (8): 1143–56. doi:10.1109/PROC.1979.11409.

Bergmans, Jan W. M. 1996. “Linear Equalization.” In *Digital Baseband Transmission and Recording*, edited by Jan W. M. Bergmans, 231–64. Boston, MA: Springer US. doi:10.1007/978-1-4757-2471-4_5.

Borah, Tripti, Kandarpa Sarma, and Pranhari Talukdar. 2015. “Biometric Identification System Using Neuro and Fuzzy Computational Approaches.” In , 335–68. doi:10.4018/978-1-4666-8654-0.ch016.

Chandradevan, Ramraj. 2017. “Radial Basis Functions Neural Networks — All We Need to Know.” *Medium*. August 18. <https://towardsdatascience.com/radial-basis-functions-neural-networks-all-we-need-to-know-9a88cc053448>.

Colombaroni, Chiara, Gaetano Fusco, and Natalia Isaenko. 2021. “Modeling Car Following with Feed-Forward and Long-Short Term Memory Neural Networks.” *Transportation Research Procedia*, 23rd EURO Working Group on Transportation Meeting, EWGT 2020, 16-18 September 2020, Paphos, Cyprus, 52 (January): 195–202. doi:10.1016/j.trpro.2021.01.022.

Gu, Jiuxiang, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, et al. 2017. “Recent Advances in Convolutional Neural Networks.” *ArXiv:1512.07108 [Cs]*, October. <http://arxiv.org/abs/1512.07108>.

Ha, Tri T., ed. 2010. “Intersymbol Interference and Equalization.” In *Theory and Design of Digital Communication Systems*, 473–521. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511778681.010.

Hassan, Hassan, Abdelazim Negm, Mohamed Zahran, and Oliver Saavedra. 2015. “ASSESSMENT OF ARTIFICIAL NEURAL NETWORK FOR BATHYMETRY ESTIMATION USING HIGH RESOLUTION SATELLITE IMAGERY IN SHALLOW LAKES: CASE STUDY EL BURULLUS LAKE.” *International Water Technology Journal* 5 (December).

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. “Long Short-Term Memory.” *Neural Computation* 9 (December): 1735–80. doi:10.1162/neco.1997.9.8.1735.

Hur, Youngsik, Minjae Maeng, Cheolwoo Chun, Franklin Bien, Hyungsoo Kim, Soumya Chandramouli, Edward Gebara, and Joy Laskar. 2005. “Equalization and Near-End Crosstalk (NEXT) Noise Cancellation for 20-Gb/s 4-PAM Backplane Serial I/O Interconnections.” *Microwave Theory and Techniques, IEEE Transactions On* 53 (February): 246–55. doi:10.1109/TMTT.2004.839311.

Judd, Stephen. 1988. “On the Complexity of Loading Shallow Neural Networks.” *Journal of Complexity* 4 (3): 177–92. doi:10.1016/0885-064X(88)90019-2.

K T, Raghavendra, and Amiya Tripathy. 2006. “An Efficient Channel Equalizer Using Artificial Neural Networks.” *Neural Network World* 16 (January).

Kechriotis, G., E. Zervas, and E.S. Manolakos. 1994. "Using Recurrent Neural Networks for Adaptive Communication Channel Equalization." *IEEE Transactions on Neural Networks* 5 (2): 267–78. doi:10.1109/72.279190.

Lavanaia, Shubham, Brando Kumam, Palash Sushil Matey, Visalakshi Annepu, and Kalapraveen Bagadi. 2015. "Adaptive Channel Equalization Using Recurrent Neural Network under SUI Channel Model." In *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 1–6. doi:10.1109/ICIIECS.2015.7193035.

Lu, B., and B.L. Evans. 1999. "Channel Equalization by Feedforward Neural Networks." In *1999 IEEE International Symposium on Circuits and Systems (ISCAS)*, 5:587–90 vol.5. doi:10.1109/ISCAS.1999.777640.

Lu, Xingyu, Chao Lu, Weixiang Yu, Liang Qiao, Shangyu Liang, Alan Pak Tao Lau, and Nan Chi. 2019. "Memory-Controlled Deep LSTM Neural Network Post-Equalizer Used in High-Speed PAM VLC System." *Optics Express* 27 (5). Optical Society of America: 7822–33. doi:10.1364/OE.27.007822.

Müller, Berndt, Joachim Reinhardt, and Michael T. Strickland. 1995. *Neural Networks: An Introduction*. Springer Science & Business Media.

"Perceptron." 2019. *DeepAI*. May 17. <https://deepai.org/machine-learning-glossary-and-terms/perceptron>.

Person, Abigail L., and Kamran Khodakhah. 2016. "Recurrent Feedback Loops in Associative Learning." *Neuron* 89 (3): 427–30. doi:10.1016/j.neuron.2016.01.037.

Proakis, John G., and Masoud Salehi. 2002. *Communication Systems Engineering*. 2nd ed. Upper Saddle River, N.J: Prentice Hall.

Quiza, Ramon, and J. Davim. 2011. "Computational Methods and Optimization." In *Machining of Hard Materials*, 177–208. doi:10.1007/978-1-84996-450-0.

Ramchoun, Hassan, Mohammed Amine, Mohammed Amine Janati Idrissi, Youssef Ghanou, and Mohamed Ettaouil. 2016. "Multilayer Perceptron: Architecture Optimization and Training." *International Journal of Interactive Multimedia and Artificial Intelligence* 4 (January): 26–30. doi:10.9781/ijimai.2016.415.

Rojas, Raúl. 1996. "Weighted Networks—The Perceptron." In *Neural Networks*, by Raúl Rojas, 55–76. Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-61068-4_3.

Rosenblatt, F. 1958. “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.” *Psychological Review* 65 (6). US: American Psychological Association: 386–408. doi:10.1037/h0042519.

Schuster, M., and K.K. Paliwal. 1997. “Bidirectional Recurrent Neural Networks.” *IEEE Transactions on Signal Processing* 45 (11): 2673–81. doi:10.1109/78.650093.

Seising, Rudolf. 2018. “The Emergence of Fuzzy Sets in the Decade of the Perceptron—Lotfi A. Zadeh’s and Frank Rosenblatt’s Research Work on Pattern Classification.” *Mathematics* 6 (June): 110. doi:10.3390/math6070110.

Sharma, Siddharth, Simone Sharma, UG Scholar, and Anidhya Athaiya. 2020. “□ACTIVATION FUNCTIONS IN NEURAL NETWORKS” 4 (12): 7.

Siebenrock, Kai. (2018) 2021. *Activation Functions*. Jupyter Notebook. <https://github.com/siebenrock/activation-functions>.

“Solving XOR Problem with a Multilayer Perceptron.” 2012. *Nn04_mlp_xor.m*. October 9. http://lab.fs.uni-lj.si/lasin/wp/IMIT_files/neural/nn04_mlp_xor/#1.

Szandała, Tomasz. 2021. “Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks.” In *Bio-Inspired Neurocomputing*, edited by Akash Kumar Bhoi, Pradeep Kumar Mallick, Chuan-Ming Liu, and Valentina E. Balas, 903:203–24. Studies in Computational Intelligence. Singapore: Springer Singapore. doi:10.1007/978-981-15-5495-7_11.

“Understanding Activation Functions in Depth.” 2019. *GeeksforGeeks*. April 10. <https://www.geeksforgeeks.org/understanding-activation-functions-in-depth/>.

Vedaldi, Andrea, and Karel Lenc. 2016. “MatConvNet - Convolutional Neural Networks for MATLAB.” *ArXiv:1412.4564 [Cs]*, May. <http://arxiv.org/abs/1412.4564>.

Waliullah, G. M., Diponkor Bala, Mst Hena, Md Abdullah, and Mohammad Hossain. 2020. “Performance Analysis of Zero Forcing and MMSE Equalizer on MIMO System in Wireless Channel” 8 (August): 19–25.

Wang, Zihao, Zhifei Xu, Jiayi He, Chulsoon Hwang, Jun Fan, and Hervé Delingette. 2020. “Long Short-Term Memory Neuron Equalizer.” *ArXiv:2010.14009 [Eess]*, October. <http://arxiv.org/abs/2010.14009>.

Wesolowski, Marek, and Bogdan Suchacz. 2012. “Artificial Neural Networks: Theoretical Background and Pharmaceutical Applications: A Review.” *Journal of AOAC International* 95 (3): 652–68. doi:10.5740/jaoacint.sge_wesolowski_ann.

Williams, Ronald J., and David Zipser. 1989. "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks." *Neural Computation* 1 (2): 270–80. doi:10.1162/neco.1989.1.2.270.

Xu, Weihong, Zhiwei Zhong, Yair Be'ery, Xiaohu You, and Chuan Zhang. 2018. "Joint Neural Network Equalizer and Decoder." In *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, 1–5. doi:10.1109/ISWCS.2018.8491056.

Yu, Hao, Tiantian Xie, Stanisław Paszczynski, and Bogdan M. Wilamowski. 2011. "Advantages of Radial Basis Function Networks for Dynamic System Design." *IEEE Transactions on Industrial Electronics* 58 (12): 5438–50. doi:10.1109/TIE.2011.2164773.

Γεωργούλη, Κατερίνα. 2015. *Τεχνητή Νοημοσύνη*. www.kallipos.gr

Παράρτημα Α

```
clear;
clc;
percentErrors=1;
percentErrors2=1;
percentErrors3=1;
percentErrors4=1;
percentErrors5=1;
percentErrors6=1;
load ('s1');

%s=sign(rand(1,10000)-0.5);
s = s1;
h=[0.407 0.815 0.407];
sum(h.^2);
v=conv(s,h);

sigma=sqrt(1/10^(14/10));
r=v+sigma*randn(1,length(v));

j=1;
i=1;
k=1; %counter of size(s)
M=3;
A=zeros;

for k=(M:length(s))
    if s(k-2)==-1
        kata(1,j)=0;
        kata(2,j)=1;
    else
        kata(1,j)=1;
        kata(2,j)=0;
    end

    A(1:M,j)=[r(k) r(k-1) r(k-2)]';
    j=j+1;
    k=k+1;
end

% Solve a Pattern Recognition Problem with a Neural Network
% Script generated by Neural Pattern Recognition app
% Created 15-Apr-2021 01:12:10
%
% This script assumes these variables are defined:
%
% A - input data.
% kata - target data.
%%perm=randperm(width(A));
x = A;
t = kata;

% Choose a Training Function
% For a list of all training functions type: help nntain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
```

Εξίσωση μη γραμμικών καναλιών με χρήση τεχνικών Μηχανικής Μάθησης

```
% 'trainscg' uses less memory. Suitable in low memory situations.  
trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.
```

```
% Create a Pattern Recognition Network  
hiddenLayerSize = [3];  
net = patternnet(hiddenLayerSize, trainFcn);
```

```
% Setup Division of Data for Training, Validation, Testing  
net.divideParam.trainRatio = 70/100;  
net.divideParam.valRatio = 15/100;  
net.divideParam.testRatio = 15/100;
```

```
%%while percentErrors>0.02  
% Train the Network  
[net,tr] = train(net,x,t);
```

```
% Test the Network  
y = net(x);  
e = gsubtract(t,y);  
performance = perform(net,t,y)  
tind = vec2ind(t);  
yind = vec2ind(y);  
percentErrors = sum(tind ~= yind)/numel(tind);  
%%end  
% View the Network
```

```
% Plots  
% Uncomment these lines to enable various plots.  
%figure, plotperform(tr)  
%figure, plottrainstate(tr)  
%figure, ploterrhist(e)  
%figure, plotconfusion(t,y)  
%figure, plotroc(t,y)
```

```
net_3=net  
save net_3
```

```
%Second ANN  
trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.
```

```
% Create a Pattern Recognition Network  
hiddenLayerSize = [5];  
net = patternnet(hiddenLayerSize, trainFcn);
```

```
% Setup Division of Data for Training, Validation, Testing  
net.divideParam.trainRatio = 70/100;  
net.divideParam.valRatio = 15/100;  
net.divideParam.testRatio = 15/100;
```

```
% Train the Network  
[net,tr] = train(net,x,t);
```

```
%%while percentErrors2>0.02  
% Test the Network
```

Εξίσωση μη γραμμικών καναλιών με χρήση τεχνικών Μηχανικής Μάθησης

```
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors2 = sum(tind ~= yind)/numel(tind);
%%end

net_5=net
save net_5

% Third ANN

trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.

% Create a Pattern Recognition Network
hiddenLayerSize = [10];
net = patternnet(hiddenLayerSize, trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
%%while percentErrors3>0.02
% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors3 = sum(tind ~= yind)/numel(tind);
%%end

net_10=net
save net_10

%4th ANN

trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.

% Create a Pattern Recognition Network
hiddenLayerSize = [5 10];
net = patternnet(hiddenLayerSize, trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
```

Εξίσωση μη γραμμικών καναλιών με χρήση τεχνικών Μηχανικής Μάθησης
%%while percentErrors4>0.02

```
% Train the Network  
[net,tr] = train(net,x,t);
```

```
% Test the Network  
y = net(x);  
e = gsubtract(t,y);  
performance = perform(net,t,y)  
tind = vec2ind(t);  
yind = vec2ind(y);  
percentErrors4 = sum(tind ~= yind)/numel(tind);  
%%end
```

```
net_5_10=net  
save net_5_10
```

%5th ANN

```
trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.
```

```
% Create a Pattern Recognition Network  
hiddenLayerSize = [10 10];  
net = patternnet(hiddenLayerSize, trainFcn);
```

```
% Setup Division of Data for Training, Validation, Testing  
net.divideParam.trainRatio = 70/100;  
net.divideParam.valRatio = 15/100;  
net.divideParam.testRatio = 15/100;
```

```
%%while percentErrors5>0.02
```

```
% Train the Network  
[net,tr] = train(net,x,t);
```

```
% Test the Network  
y = net(x);  
e = gsubtract(t,y);  
performance = perform(net,t,y)  
tind = vec2ind(t);  
yind = vec2ind(y);  
percentErrors5 = sum(tind ~= yind)/numel(tind);  
%%end
```

```
net_10_10=net  
save net_10_10
```

%6th ANN

```
trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.
```

```
% Create a Pattern Recognition Network  
hiddenLayerSize = [15 10];  
net = patternnet(hiddenLayerSize, trainFcn);
```

Εξίσωση μη γραμμικών καναλιών με χρήση τεχνικών Μηχανικής Μάθησης

```
% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

%%while percentErrors6>0.02
% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors6 = sum(tind ~= yind)/numel(tind);
%%end

net_15_10=net
save net_15_10

%Testing the networks

times=1;
timesdB=1;
PE=zeros(times);
PE2=zeros(times);
PE3=zeros(times);
PE4=zeros(times);
PE5=zeros;
PE6=zeros;
dBArray=zeros(timesdB);
Testarisma=1;
s=sign(rand(1,1000000)-0.5); % Production of Bitstream
h=[0.407 0.815 0.407]; % Channel
sum(h.^2);
v=conv(s,h) ;
M=3;
for dB=(14:-1:0)
sigma=sqrt(1/10^(dB/10));
r=v+sigma*randn(1,length(v));% Signal with distortion and noise

j=1;
i=1;
k=1; %counter of size(s)
A=zeros(i,j); % Input of ANN
```

Εξίσωση μη γραμμικών καναλιών με χρήση τεχνικών Μηχανικής Μάθησης

```
for k=(M:length(s))
    if s(k-2)==-1
        katal(1,j)=0;
        katal(2,j)=1;
    else
        katal(1,j)=1;
        katal(2,j)=0;
    end

    A(1:M,j)=[r(k) r(k-1) r(k-2)]';
    j=j+1;
    k=k+1;

end

% Solve a Pattern Recognition Problem with a Neural net_8_16_8_1work
% Script generated by Neural Pattern Recognition app
% Created 25-Apr-2021 01:12:10
%
% This script assumes these variables are defined:
%
% A - input data.
% katal - target data.
% Already Trained

x = A;
t = katal;

% Test the net_3
y = net_3(x);
e = gsubtract(t,y);
performance = perform(net_3,t,y);
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);

% Test the net_5
y2 = net_5(x);
e2 = gsubtract(t,y2);
performance2 = perform(net_5,t,y2);
tind2 = vec2ind(t);
yind2 = vec2ind(y2);
percentErrors2 = sum(tind2 ~= yind2)/numel(tind2);

% Test the net_10
y3 = net_10(x);
e3 = gsubtract(t,y3);
performance3 = perform(net_10,t,y3);
tind3 = vec2ind(t);
yind3 = vec2ind(y3);
percentErrors3 = sum(tind3 ~= yind3)/numel(tind3);

%Test the net net_5_10

y4 = net_5_10(x);
e4 = gsubtract(t,y4);
performance4 = perform(net_5_10,t,y4);
tind4 = vec2ind(t);
```

```
yind4 = vec2ind(y4);
percentErrors4 = sum(tind4 ~= yind4)/numel(tind4);
% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotconfusion(t,y)
%figure, plotroc(t,y)

%Test the net net_10_10

y5 = net_10_10(x);
e5 = gsubtract(t,y5);
performance5 = perform(net_10_10,t,y5);
tind5 = vec2ind(t);
yind5 = vec2ind(y5);
percentErrors5 = sum(tind5 ~= yind5)/numel(tind5);
% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotconfusion(t,y)
%figure, plotroc(t,y)

%Test the net net_15_10

y6 = net_15_10(x);
e6 = gsubtract(t,y6);
performance6 = perform(net_15_10,t,y6);
tind6 = vec2ind(t);
yind6 = vec2ind(y6);
percentErrors6 = sum(tind6 ~= yind6)/numel(tind6);
% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotconfusion(t,y)
%figure, plotroc(t,y)

PE(times)=percentErrors;
PE2(times)=percentErrors2;% Matrix for BER
PE3(times)=percentErrors3;
PE4(times)=percentErrors4;
PE5(times)=percentErrors5;
PE6(times)=percentErrors6;
times=times+1 ; % Counter for BER
dBArray(timesdB)=dB ; % dB of SNR
timesdB=timesdB+1 ; % Counter for dB
end
```



```
semilogy(dBArray, PE, 'g--', dBArray, PE2, 'b--o', dBArray, PE3, 'y-  
o', dBArray, PE4, 'm-*', dBArray, PE5, 'r-+', dBArray, PE6, 'k-d')  
hold on  
grid  
legend(' MLP {3} ', ' MLP {5} ', ' MLP {10} ', ' MLP {5:10}', ' MLP {10:10}', '  
MLP {15:10}', 'Location', 'northeast')  
title('Ber comparison for MLP equalizers')  
xlabel('Eb/No (dB)')  
ylabel('Bit Error Rate')  
ylim([10^(-3) 10^0])
```

Παράρτημα Β

```

% System simulation parameters
Fs = 1;           % sampling frequency (notional)
nBits = 2048;    % number of BPSK symbols per vector
maxErrs = 200;   % target number of errors at each Eb/No
maxBits = 1e6;   % maximum number of symbols at each Eb/No

% Modulated signal parameters
M = 2;           % order of modulation
Rs = Fs;         % symbol rate
nSamp = Fs/Rs;   % samples per symbol
Rb = Rs*log2(M); % bit rate

% Channel parameters
chnl = [0.407 0.815 0.407]'; % channel impulse response
chnlLen = length(chnl);      % channel length, in samples
EbNo = 0:14;                 % in dB
BER = zeros(size(EbNo));     % initialize values

% Create BPSK modulator
bpskMod = comm.BPSKModulator;

% Specify a seed for the random number generators to ensure repeatability.
rng(12345)

% Linear equalizer parameters
nWts = 31;           % number of weights
algType = 'RLS';     % RLS algorithm
forgetFactor = 0.999999; % parameter of RLS algorithm

% DFE parameters - use same update algorithms as linear equalizer
nFwdWts = 15;       % number of feedforward weights
nFbkWts = 15;       % number of feedback weights

% MLSE equalizer parameters
tbLen = 30;         % MLSE equalizer traceback length
numStates = M^(chnlLen-1); % number of trellis states
[mlseMetric,mlseStates,mlseInputs] = deal([]);
const = constellation(bpskMod); % signal constellation
mlseType = 'ideal'; % perfect channel estimates at first
mlseMode = 'cont'; % no MLSE resets

% Channel estimation parameters
chnlEst = chnl;     % perfect estimation initially
prefixLen = 2*chnlLen; % cyclic prefix length
excessEst = 1;      % length of estimated channel impulse response
% beyond the true length

% Initialize the graphics for the simulation. Plot the unequalized channel
% frequency response, and the BER of an ideal BPSK system.
idealBER = berawgn(EbNo, 'psk', M, 'nondiff');

[hBER, hLegend, legendString, hLinSpec, hDfeSpec, hErrs, hText1, hText2, ...
 hFit, hEstPlot, hFig, hLinFig, hDfeFig] = eqber_graphics('init', ...
 chnl, EbNo, idealBER, nBits);

```

```
linEq = comm.LinearEqualizer('Algorithm',algType, ...
    'ForgettingFactor',forgetFactor, ...
    'NumTaps',nWts, ...
    'Constellation',const, ...
    'ReferenceTap',round(nWts/2), ...
    'TrainingFlagInputPort',true);

dfeEq = comm.DecisionFeedbackEqualizer('Algorithm',algType, ...
    'ForgettingFactor',forgetFactor, ...
    'NumForwardTaps',nFwdWts, ...
    'NumFeedbackTaps',nFbkWts, ...
    'Constellation',const, ...
    'ReferenceTap',round(nFwdWts/2), ...
    'TrainingFlagInputPort',true);

firstRun = true; % flag to ensure known initial states for noise and data
eqType = 'linear';
eqber_adaptive;

close(hFig(ishghandle(hFig)));

eqType = 'dfe';
eqber_adaptive;

close(hLinFig(ishghandle(hLinFig)),hDfeFig(ishghandle(hDfeFig)));

eqType = 'mlse';
mlseType = 'ideal';
eqber_mlse;

mlseType = 'imperfect';
eqber_mlse;
```

Παράρτημα Γ

```

clear;
clc;
s=sign(rand(1,100000)-0.5);
h=[0.407 0.815 0.407];
sum(h.^2);
v=conv(s,h);
for dB=(1:14)
sigma=sqrt(1/10^(dB/10));
r=v+sigma*randn(1,length(v));

j=1;
i=1;
k=1; %counter of size(s)
M=3;
A=zeros;

for k=(M:length(s))
    if s(k-2)==-1
        kata(1,j)=0;

    else
        kata(1,j)=1;

    end

    A(1:M,j)=[r(k) r(k-1) r(k-2)]';
    j=j+1;
    k=k+1;

end

% Solve an Autoregression Problem with External Input with a NARX Neural
Network
% Script generated by Neural Time Series app
% Created 08-Oct-2021 20:14:26
%
% This script assumes these variables are defined:
%
% A - input time series.
% kata - feedback time series.

X = tonndata(A,true,false);
T = tonndata(kata,true,false);

% Choose a Training Function
% For a list of all training functions type: help ntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.

% Create a Nonlinear Autoregressive Network with External Input
inputDelays = 1:2;
feedbackDelays = 1:2;
hiddenLayerSize = 10;
net = narxnet(inputDelays,feedbackDelays,hiddenLayerSize,'open',trainFcn);

```

Εξίσωση μη γραμμικών καναλιών με χρήση τεχνικών Μηχανικής Μάθησης

```
% Prepare the Data for Training and Simulation
% The function PREPARETS prepares timeseries data for a particular network,
% shifting time by the minimum amount to fill input states and layer
% states. Using PREPARETS allows you to keep your original time series data
% unchanged, while easily customizing it for networks with differing
% numbers of delays, with open loop or closed loop feedback modes.
[x,xi,ai,t] = preparets(net,X,{},T);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Train the Network
[net,tr] = train(net,x,t,xi,ai);

% Test the Network
y = net(x,xi,ai);
e = gsubtract(t,y);
performance = perform(net,t,y)

% View the Network
%%% view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotresponse(t,y)
%figure, ploterrcorr(e)
%figure, plotinerrcorr(x,e)

% Closed Loop Network
% Use this network to do multi-step prediction.
% The function CLOSELOOP replaces the feedback input with a direct
% connection from the output layer.
%%%netc = closeloop(net);
%%%netc.name = [net.name ' - Closed Loop'];
%%%view(netc)
%%%[xc,xic,aic,tc] = preparets(netc,X,{},T);
%%%yc = netc(xc,xic,aic);
%%%closedLoopPerformance = perform(net,tc,yc)

% Step-Ahead Prediction Network
% For some applications it helps to get the prediction a timestep early.
% The original network returns predicted y(t+1) at the same time it is
% given y(t+1). For some applications such as decision making, it would
% help to have predicted y(t+1) once y(t) is available, but before the
% actual y(t+1) occurs. The network can be made to return its output a
% timestep early by removing one delay so that its minimal tap delay is now
% 0 instead of 1. The new network returns the same outputs as the original
% network, but outputs are shifted left one timestep.
%%%nets = removedelay(net);
%%%nets.name = [net.name ' - Predict One Step Ahead'];
%%%view(nets)
%%%[xs,xis,ais,ts] = preparets(nets,X,{},T);
%%%ys = nets(xs,xis,ais);
%%%stepAheadPerformance = perform(nets,ts,ys)
y2=cell2mat(y);
arithmos=0;
```

Εξίσωση μη γραμμικών καναλιών με χρήση τεχνικών Μηχανικής Μάθησης

```
for i=(1:length(y2))
    if y2(i)>0.5
        y2(i)=1;
    else
        y2(i)=0;
    end
end
for i=(1:length(y2))
    if y2(i)~=kata(i+2)
        arithmos=arithmos+1;
    end
end
PE(dB)=arithmos/length(y2);
end
dBArray=[ 1 2 3 4 5 6 7 8 9 10 11 12 13 14];

semilogy(dBArray,PE,'r')
hold on
grid
legend(' RNN {10} ', 'Location','northeast')
title('Ber for RNN equalizer')
xlabel('Eb/No (dB)')
ylabel('Bit Error Rate')
ylim([10^(-3) 10^0])
```

Παράρτημα Δ

```

percentErrors=1;
percentErrors2=1;
percentErrors3=1;
percentErrors4=1;
percentErrors5=1;
percentErrors6=1;
load ('s1');

%s=sign(rand(1,10000)-0.5);
s = s1;
h=[0.407 0.815 0.407];
sum(h.^2);
v=conv(s,h);
g = v -0.9*(v.^3);

sigma=sqrt(1/10^(30/10));
r=g+sigma*randn(1,length(v));

j=1;
i=1;
k=1; %counter of size(s)
M=3;
A=zeros;

for k=(M:length(s))
    if s(k-2)==-1
        kata(1,j)=0;
        kata(2,j)=1;
    else
        kata(1,j)=1;
        kata(2,j)=0;
    end

    A(1:M,j)=[r(k) r(k-1) r(k-2) ]';
    j=j+1;
    k=k+1;

end

% Solve a Pattern Recognition Problem with a Neural Network
% Script generated by Neural Pattern Recognition app
% Created 15-Apr-2021 01:12:10
%
% This script assumes these variables are defined:
%
% A - input data.
% kata - target data.
%%perm=randperm(width(A));
x = A;
t = kata;

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.

```

Εξίσωση μη γραμμικών καναλιών με χρήση τεχνικών Μηχανικής Μάθησης
trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.

```
% Create a Pattern Recognition Network
hiddenLayerSize = [3];
net = patternnet(hiddenLayerSize, trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
```

```
%%while percentErrors>0.02
% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);
%%end
% View the Network
```

```
% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotconfusion(t,y)
%figure, plotroc(t,y)
```

```
net_3=net
save net_3
```

```
%Second ANN
trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.
```

```
% Create a Pattern Recognition Network
hiddenLayerSize = [5];
net = patternnet(hiddenLayerSize, trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
```

```
% Train the Network
[net,tr] = train(net,x,t);
```

```
%%while percentErrors2>0.02
% Test the Network
y = net(x);
```


Εξίσωση μη γραμμικών καναλιών με χρήση τεχνικών Μηχανικής Μάθησης

```
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors2 = sum(tind ~= yind)/numel(tind);
%%end

net_5=net
save net_5

% Third ANN

trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.

% Create a Pattern Recognition Network
hiddenLayerSize = [10];
net = patternnet(hiddenLayerSize, trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
%%while percentErrors3>0.02
% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors3 = sum(tind ~= yind)/numel(tind);
%%end

net_10=net
save net_10

%4th ANN

trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.

% Create a Pattern Recognition Network
hiddenLayerSize = [4 3];
net = patternnet(hiddenLayerSize, trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

%%while percentErrors4>0.02
```

```
% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors4 = sum(tind ~= yind)/numel(tind);
%%end

net_4_3=net
save net_4_3

%5th ANN

trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.

% Create a Pattern Recognition Network
hiddenLayerSize = [7 3];
net = patternnet(hiddenLayerSize, trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

%%while percentErrors5>0.02
% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors5 = sum(tind ~= yind)/numel(tind);
%%end

net_7_3=net
save net_7_3

%6th ANN

trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.

% Create a Pattern Recognition Network
hiddenLayerSize = [7 5];
net = patternnet(hiddenLayerSize, trainFcn);
```

Εξίσωση μη γραμμικών καναλιών με χρήση τεχνικών Μηχανικής Μάθησης

```
% Setup Division of Data for Training, Validation, Testing
```

```
net.divideParam.trainRatio = 70/100;
```

```
net.divideParam.valRatio = 15/100;
```

```
net.divideParam.testRatio = 15/100;
```

```
%%while percentErrors6>0.02
```

```
% Train the Network
```

```
[net,tr] = train(net,x,t);
```

```
% Test the Network
```

```
y = net(x);
```

```
e = gsubtract(t,y);
```

```
performance = perform(net,t,y)
```

```
tind = vec2ind(t);
```

```
yind = vec2ind(y);
```

```
percentErrors6 = sum(tind ~= yind)/numel(tind);
```

```
%%end
```

```
net_7_5=net
```

```
save net_7_5
```

```
%Kwdikas gia testarisma
```

```
times=1;
```

```
timesdB=1;
```

```
PE=zeros(times);
```

```
PE2=zeros(times);
```

```
PE3=zeros(times);
```

```
PE4=zeros(times);
```

```
PE5=zeros;
```

```
PE6=zeros;
```

```
dBArray=zeros(timesdB);
```

```
Testarisma=1;
```

```
s=sign(rand(1,1000000)-0.5); % Production of Bitstream
```

```
h=[0.407 0.815 0.407]; % Channel
```

```
sum(h.^2);
```

```
v=conv(s,h) ;
```

```
g = v -0.9*(v.^3);
```

```
M=3;
```

```
for dB=(20:-1:0)
```

```
sigma=sqrt(1/10^(dB/10));
```

```
r=g+sigma*randn(1,length(v));% Signal with distortion and noise
```

```
j=1;
```

```
i=1;
```

```
k=1; %counter of size(s)
```

Εξίσωση μη γραμμικών καναλιών με χρήση τεχνικών Μηχανικής Μάθησης
A=zeros(i,j); % Input of ANN

```
for k=(M:length(s))
    if s(k-2)==-1
        katal(1,j)=0;
        katal(2,j)=1;
    else
        katal(1,j)=1;
        katal(2,j)=0;
    end

    A(1:M,j)=[r(k) r(k-1) r(k-2)]';
    j=j+1;
    k=k+1;

end

% Solve a Pattern Recognition Problem with a Neural net_8_16_8_1work
% Script generated by Neural Pattern Recognition app
% Created 25-Apr-2021 01:12:10
%
% This script assumes these variables are defined:
%
% A - input data.
% katal - target data.
% Already Trained

x = A;
t = katal;

% Test the net_3
y = net_3(x);
e = gsubtract(t,y);
performance = perform(net_3,t,y);
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);

% Test the net_5
y2 = net_5(x);
e2 = gsubtract(t,y2);
performance2 = perform(net_5,t,y2);
tind2 = vec2ind(t);
yind2 = vec2ind(y2);
percentErrors2 = sum(tind2 ~= yind2)/numel(tind2);

% Test the net_10
y3 = net_10(x);
e3 = gsubtract(t,y3);
performance3 = perform(net_10,t,y3);
tind3 = vec2ind(t);
yind3 = vec2ind(y3);
percentErrors3 = sum(tind3 ~= yind3)/numel(tind3);

%Test the net net_4_3
```

```

y4 = net_4_3(x);
e4 = gsubtract(t,y4);
performance4 = perform(net_4_3,t,y4);
tind4 = vec2ind(t);
yind4 = vec2ind(y4);
percentErrors4 = sum(tind4 ~= yind4)/numel(tind4);
% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotconfusion(t,y)
%figure, plotroc(t,y)

```

```
%Test the net net_7_3
```

```

y5 = net_7_3(x);
e5 = gsubtract(t,y5);
performance5 = perform(net_7_3,t,y5);
tind5 = vec2ind(t);
yind5 = vec2ind(y5);
percentErrors5 = sum(tind5 ~= yind5)/numel(tind5);
% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotconfusion(t,y)
%figure, plotroc(t,y)

```

```
%Test the net net_7_5
```

```

y6 = net_7_5(x);
e6 = gsubtract(t,y6);
performance6 = perform(net_7_5,t,y6);
tind6 = vec2ind(t);
yind6 = vec2ind(y6);
percentErrors6 = sum(tind6 ~= yind6)/numel(tind6);
% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotconfusion(t,y)
%figure, plotroc(t,y)

```

```

PE(times)=percentErrors;
PE2(times)=percentErrors2;% Matrix for BER
PE3(times)=percentErrors3;
PE4(times)=percentErrors4;
PE5(times)=percentErrors5;
PE6(times)=percentErrors6;
times=times+1 ; % Counter for BER
dBArray(timesdB)=dB ; % dB of SNR

```

```
Εξίσωση μη γραμμικών καναλιών με χρήση τεχνικών Μηχανικής Μάθησης
timesdB=timesdB+1 ; % Counter for dB
end
```

```
semilogy(dBArray, PE, 'g--', dBArray, PE2, 'b--o', dBArray, PE3, 'y-
o', dBArray, PE4, 'm-*', dBArray, PE5, 'r+', dBArray, PE6, 'k-d')
hold on
grid
title ('Ber-SNR graph for nonlinear channel')
legend(' MLP 3 ', ' MLP 5 ', 'MLP 10 ', ' MLP 4:3', ' MLP 7:3', ' MLP 7:5',
'Location', 'northeast')
xlabel('Eb/No (dB)')
ylabel('Bit Error Rate')
ylim([10^(-3) 10^0])
```

Παράρτημα Ε

```
clear;
clc;
s=sign(rand(1,100000)-0.5);
h=[0.407 0.815 0.407];
sum(h.^2);
v=conv(s,h);
for dB=(1:14)
sigma=sqrt(1/10^(dB/10));
r=v+sigma*randn(1,length(v));

j=1;
i=1;
k=1; %counter of size(s)
M=3;
A=zeros;

for k=(M:length(s))
    if s(k-2)==-1
        kata(1,j)=0;

    else
        kata(1,j)=1;

    end

    A(1:M,j)=[r(k) r(k-1) r(k-2)]';
    j=j+1;
    k=k+1;

end

% Solve an Autoregression Problem with External Input with a NARX Neural
Network
% Script generated by Neural Time Series app
% Created 08-Oct-2021 20:14:26
%
% This script assumes these variables are defined:
%
% A - input time series.
% kata - feedback time series.

X = tonndata(A,true,false);
T = tonndata(kata,true,false);

% Choose a Training Function
% For a list of all training functions type: help ntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.

% Create a Nonlinear Autoregressive Network with External Input
inputDelays = 1:2;
feedbackDelays = 1:2;
hiddenLayerSize = 10;
net = narxnet(inputDelays,feedbackDelays,hiddenLayerSize,'open',trainFcn);
```

Εξίσωση μη γραμμικών καναλιών με χρήση τεχνικών Μηχανικής Μάθησης

```
% Prepare the Data for Training and Simulation
% The function PREPARETS prepares timeseries data for a particular network,
% shifting time by the minimum amount to fill input states and layer
% states. Using PREPARETS allows you to keep your original time series data
% unchanged, while easily customizing it for networks with differing
% numbers of delays, with open loop or closed loop feedback modes.
[x,xi,ai,t] = preparets(net,X,{},T);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Train the Network
[net,tr] = train(net,x,t,xi,ai);

% Test the Network
y = net(x,xi,ai);
e = gsubtract(t,y);
performance = perform(net,t,y)

% View the Network
%%% view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotresponse(t,y)
%figure, ploterrcorr(e)
%figure, plotinerrcorr(x,e)

% Closed Loop Network
% Use this network to do multi-step prediction.
% The function CLOSELOOP replaces the feedback input with a direct
% connection from the output layer.
%%%netc = closeloop(net);
%%%netc.name = [net.name ' - Closed Loop'];
%%%view(netc)
%%%[xc,xic,aic,tc] = preparets(netc,X,{},T);
%%%yc = netc(xc,xic,aic);
%%%closedLoopPerformance = perform(net,tc,yc)

% Step-Ahead Prediction Network
% For some applications it helps to get the prediction a timestep early.
% The original network returns predicted y(t+1) at the same time it is
% given y(t+1). For some applications such as decision making, it would
% help to have predicted y(t+1) once y(t) is available, but before the
% actual y(t+1) occurs. The network can be made to return its output a
% timestep early by removing one delay so that its minimal tap delay is now
% 0 instead of 1. The new network returns the same outputs as the original
% network, but outputs are shifted left one timestep.
%%%nets = removedelay(net);
%%%nets.name = [net.name ' - Predict One Step Ahead'];
%%%view(nets)
%%%[xs,xis,ais,ts] = preparets(nets,X,{},T);
%%%ys = nets(xs,xis,ais);
%%%stepAheadPerformance = perform(nets,ts,ys)
y2=cell2mat(y);
arithmos=0;
```


Εξίσωση μη γραμμικών καναλιών με χρήση τεχνικών Μηχανικής Μάθησης

```
for i=(1:length(y2))
    if y2(i)>0.5
        y2(i)=1;
    else
        y2(i)=0;
    end
end
for i=(1:length(y2))
    if y2(i)~=kata(i+2)
        arithmos=arithmos+1;
    end
end
PE(dB)=arithmos/length(y2);
end
dBArray=[ 1 2 3 4 5 6 7 8 9 10 11 12 13 14];

semilogy(dBArray,PE,'r')
hold on
grid
legend(' RNN {10} ', 'Location','northeast')
title('Ber for RNN equalizer')
xlabel('Eb/No (dB)')
ylabel('Bit Error Rate')
ylim([10^(-3) 10^0])
```