



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΔΙΠΛΩΜΑ ΕΙΔΙΚΕΥΣΗΣ ΣΤΗΝ ΗΛΕΚΤΡΟΝΙΚΗ
ΚΑΙ ΤΗΝ ΡΑΔΙΟΗΛΕΚΤΡΟΛΟΓΙΑ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΦΑΡΜΟΓΗ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΣΤΗΝ ΑΠΟΚΩΔΙΚΟΠΟΙΗΣΗ
ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΗΜΑΤΩΝ

ΑΛΕΞΑΝΔΡΟΣ Γ. ΝΤΟΚΟΣ

Επιβλέπουσα: ANNA ΤΖΑΝΑΚΑΚΗ

ΑΘΗΝΑ
ΦΕΒΡΟΥΑΡΙΟΣ, 2024

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Θέμα:

Εφαρμογή Νευρωνικών Δικτύων στην αποκωδικοποίηση
τηλεπικοινωνιακών σημάτων

Αλέξανδρος Γ. Ντόκος

A.M.: 7101132100116

Επιβλέπουσα

Άννα Τζανακάκη, Αναπληρώτρια Καθηγήτρια

Τριμελής Εξεταστική Επιτροπή

Άννα Τζανακάκη, Αναπληρώτρια Καθηγήτρια

Μάρκος Αναστασόπουλος,, Αναπληρωτής Καθηγητής

Διονύσιος Ρεΐσης, Καθηγητής

ΠΕΡΙΛΗΨΗ

Οι συνελκτικοί κώδικες είναι ένας δημοφιλής τρόπος κωδικοποίησης στα σύγχρονα ψηφιακά συστήματα επικοινωνιών. Για την αποκωδικοποίηση της ληφθείσας ακολουθίας, μπορεί να γίνει χρήση διάφορων στατιστικών μεθόδων. Ένας βέλτιστος και αρκετά διαδεδομένος τρόπος αποκωδικοποίησης συνελκτικα κωδικοποιημένων σημάτων είναι μια μέθοδος μεγίστης πιθανοφάνειας, ο αλγόριθμος Viterbi.

Τα τεχνητά νευρωνικά δίκτυα είναι σύγχρονα μοντέλα μηχανικής μάθησης. Η δομή τους και η βασικές αρχές λειτουργίας τους είναι εμπνευσμένες από τα νευρωνικά δίκτυα των έμβιων οργανισμών. Αποτελούνται από τεχνητούς νευρώνες, οι οποίοι συνδέονται μεταξύ τους. Ο τρόπος σύνδεσης μεταξύ των νευρώνων καθορίζεται από την επιλεγόμενη αρχιτεκτονική. Η λειτουργία τους χωρίζεται σε δύο φάσεις: την φάση εκπαίδευσης ή μάθησης και την φάση ανάκλησης. Κατά την διαδικασία μάθησης, το νευρωνικό δίκτυο μαθαίνει τις κατάλληλες τιμές για τα βάρη των συνδέσεων έτσι ώστε να μπορεί να προσεγγίσει την λειτουργία που θέλει ο σχεδιαστής. Κατά την φάση της ανάκλησης, το νευρωνικό δίκτυο χρησιμοποιείται από τον σχεδιαστή για την τέλεση προβλέψεων βάσει όσων έμαθε κατά την φάση της εκπαίδευσης.

Στα πλαίσια της παρούσης διπλωματικής εργασίας, διερευνήθηκε η ικανότητα των νευρωνικών δικτύων να δρουν ως αποκωδικοποιητές σημάτων, τα οποία έχουν κωδικοποιηθεί με χρήση συνελκτικού κώδικα. Σχεδιάστηκε και εκπαιδεύτηκε νευρωνικό δίκτυο ικανό να αποκωδικοποιήσει με αρκετά μεγάλη επιτυχία μηνύματα, τα οποία είχαν σταλεί από έναν ψηφιακό δέκτη, ο οποίος έκανε χρήση συνελκτικού κώδικα. Για να προσομοιωθούν οι πραγματικές συνθήκες κάτω υπό τις οποίες λειτουργεί ένα τηλεπικοινωνιακό σύστημα, στην αποσταλείσα ακολουθία εισαγόταν και γκαουσιανός θόρυβος.

Δοκιμάστηκαν και αξιολογήθηκαν διάφορες δημοφιλείς αρχιτεκτονικές νευρωνικών δικτύων ως προς τις αποδόσεις τους ως αποκωδικοποιητές. Μέτρο σύγκρισης αποτέλεσε ο δημοφιλής αλγόριθμος Viterbi. Εξετάστηκε τόσο ο ρυθμός σφάλματος ανά ψηφίο (BER) όσο και ο χρόνος αποκωδικοποίησης. Ακόμη, αναλύθηκαν και εξετάστηκαν διάφοροι τρόποι μορφοποίησης της ληφθείσας ακολουθίας, ώστε αυτή να τροφοδοτηθεί στο νευρωνικό δίκτυο.

Λέξεις-Κλειδιά: Ψηφιακές Επικοινωνίες, Συνελκτικοί Κώδικες, Αποκωδικοποίηση σημάτων, Αλγόριθμος Viterbi, Τεχνητή Νοημοσύνη, Μηχανική Μάθηση, Νευρωνικά Δίκτυα.

ABSTRACT

Convolutional codes are a popular method of encoding in modern digital communication systems. For the decoding of the received sequence, various statistical methods can be employed. An optimal and widely used method for decoding convolutionally encoded signals is the maximum likelihood method, specifically the Viterbi algorithm.

Artificial neural networks are contemporary models of machine learning. Their structure and basic principles of operation are inspired by the neural networks of living organisms. They consist of artificial neurons that are interconnected, and the way these neurons are connected is determined by the chosen architecture. Their operation is divided into two phases: the training or learning phase and the recall phase. During the learning process, the neural network learns the appropriate values for the connection weights to approximate the desired functionality. In the recall phase, the neural network is used by the designer to make predictions based on what it learned during the training phase.

In the context of this thesis, the ability of neural networks to act as signal decoders encoded using convolutional codes was explored. A neural network was designed and trained to successfully decode messages sent by a digital transmitter using convolutional codes. To simulate real-world conditions under which a communication system operates, Gaussian noise was introduced into the transmitted sequence.

Various popular architectures of neural networks were tested and evaluated as decoders in terms of their performances. The widely used Viterbi algorithm served as a comparison measure. Both the bit error rate (BER) and decoding time were examined. Additionally, various ways of formatting the received sequence for input to the neural network were analyzed and investigated.

Keywords: Digital Communications, Convolutional Codes, Signal Decoding, Viterbi Algorithm, Artificial Intelligence, Machine Learning, Neural Networks.

Περιεχόμενα

1	Εισαγωγή	7
1.1	Ψηφιακά συστήματα επικοινωνίας	7
1.2	Διόρθωση σφαλμάτων και κωδικοποίηση	11
1.3	Μπλοκ κώδικες	12
1.4	Συνελικτικοί κώδικες	13
1.5	Αποκωδικοποίηση και συνελικτικοί κώδικες	14
1.6	Μηχανική Μάθηση και Νευρωνικά Δίκτυα	16
1.7	Κίνητρα της παρούσης εργασίας	18
2	Συνελικτικοί κώδικες και αποκωδικοποίηση Viterbi	20
2.1	Περιγραφή λειτουργίας συνελικτικού κώδικα	20
2.2	Αλγεβρικές αναπαραστάσεις των συνελικτικών κωδίκων	21
2.3	Οι συνελικτικοί κώδικες ως μηχανή πεπερασμένων καταστάσεων	23
2.4	Γραφικές αναπαραστάσεις των συνελικτικών κωδίκων	25
2.5	Ο αλγόριθμος Viterbi.....	29
2.6	Σκληρές και μαλακές αποφάσεις	31
2.7	Αλγόριθμος Viterbi μαλακών αποφάσεων	34
3	Μηχανική Μάθηση και Νευρωνικά Δίκτυα	36
3.1	Βελτιστοποίηση βασιζόμενη στην κλίση.....	36
3.2	Αναπαράσταση γνώσης	37
3.3	Διαδικασίες μάθησης	40
3.3.1	Μάθηση με εκπαιδευτή.....	40
3.3.2	Ενισχυτική μάθηση	41
3.3.3	Μη επιβλεπόμενη μάθηση	42
3.4	Το πρόβλημα της υπερπροσαρμογής και της υπόπροσαρμογής	43
3.5	Υπερπαράμετροι και σύνολα επαλήθευσης	45
3.6	Νευρωνικά δίκτυα πρόσθιας τροφοδότησης (MLPs)	46
3.7	Αναδρομικά Νευρωνικά Δίκτυα (RNNs).....	48
3.8	Αναδρομικά νευρωνικά δίκτυα διπλής κατεύθυνσης (Bidirectional RNNs)	49
3.9	Προβλήματα εξαφάνισης και έκρηξης κλίσης και δίκτυα LSTM και GRU.....	50
4	Εργαλεία ανάπτυξης λογισμικού	53
4.1	Γλώσσα Python.....	53
4.2	Η βιβλιοθήκη TensorFlow.....	53
4.3	Η πλατφόρμα Google Colaboratory	54
5	Ανάπτυξη νευρωνικού δικτύου για την αποκωδικοποίηση σημάτων συνελικτικού κώδικα	55

5.1	Προσομοίωση τηλεπικοινωνιακού συστήματος.....	55
5.2	Επιλογή διαδικασίας μάθησης.....	58
5.3	Συλλογή και μορφοποίηση δεδομένων εκπαίδευσης, επαλήθευσης και ελέγχου	59
5.4	Αρχιτεκτονικές νευρωνικών δικτύων	61
5.5	Συνάρτηση απωλειών, βελτιστοποιητής και μετρικές αξιολόγησης.....	62
6	Παρουσίαση μετρήσεων και αξιολόγηση αποτελεσμάτων	65
6.1	Διαδικασία εκπαίδευσης νευρωνικών δικτύων	65
6.2	Αξιολόγηση λειτουργίας εκπαιδευμένων νευρωνικών δικτύων ως αποκωδικοποιητές σημάτων	70
6.3	Έλεγχος αλγορίθμου Viterbi και καταλληλότητας του ως εκπαιδευτής.....	72
6.4	Αξιολόγηση μεθόδων μορφοποίησης εισόδου	74
6.5	Μελέτη χρόνων εκτέλεσης.....	76
6.6	Χρήση CPU και χρήση GPU.....	79
7	Συμπεράσματα	81
7.1	Συμπεράσματα από την παρούσα εργασία	81
7.2	Μελλοντικές εργασίες.....	82
8	Βιβλιογραφία.....	84

1 Εισαγωγή

1.1 Ψηφιακά συστήματα επικοινωνίας

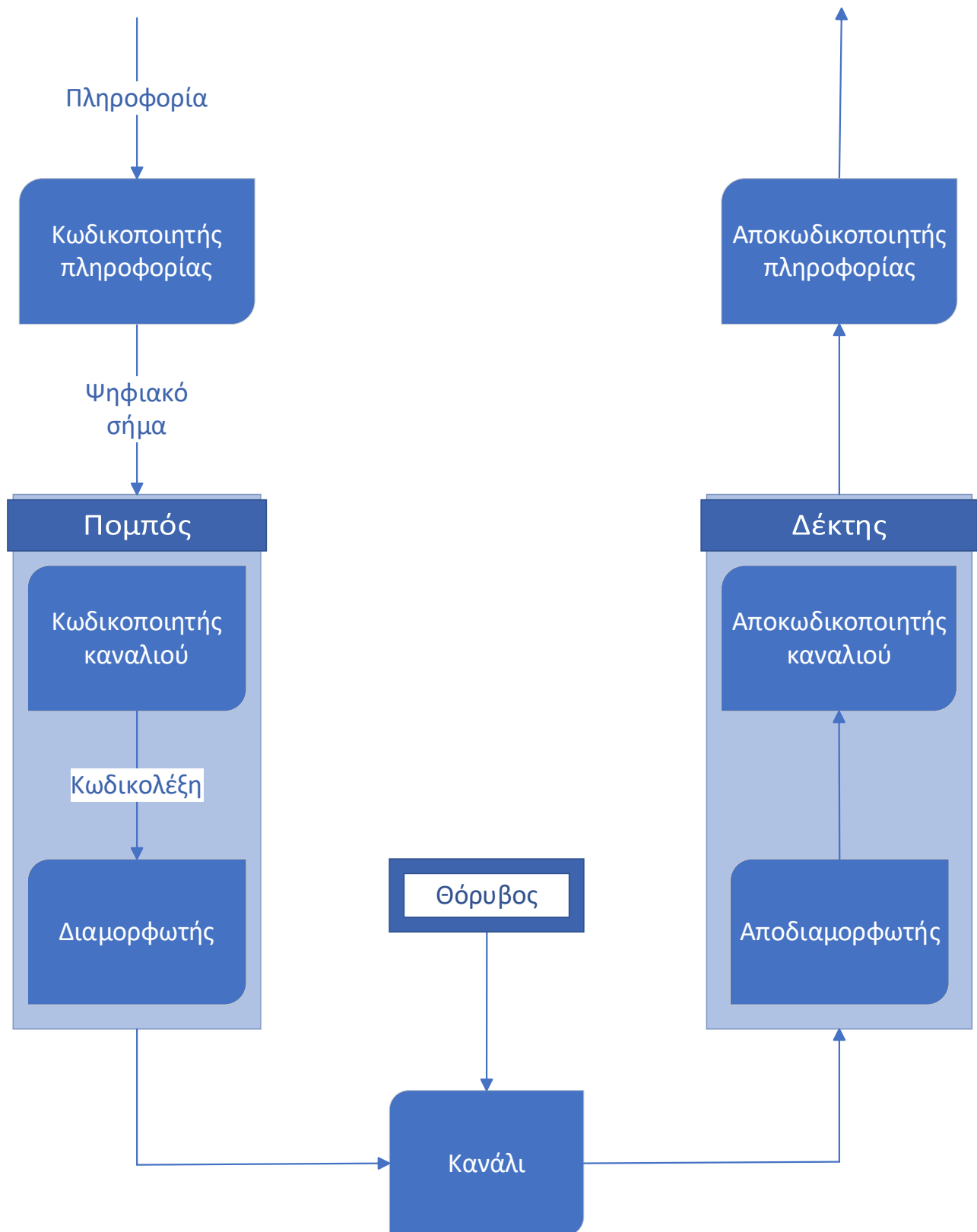
Η επικοινωνία υπήρξε ανέκαθεν ανάγκη του ανθρωπίνου γένους. Υπήρξε κεντρική σε κάθε πολιτισμένη κοινωνία. Η τεχνολογία πάντα έστεκε αρωγός στην προσπάθεια για επικοινωνία. Κάποιος θα μπορούσε να θεωρήσει και τα σήματα καπνού ως μια πρωτόγονη τεχνολογία επικοινωνίας, αλλά η τεχνολογία έγινε σημαντικότερη καθώς πλησιάζουμε προς τις μέρες μας. Πρώτα ο τηλεγράφος, ύστερα το τηλέφωνο και η τηλεόραση και σήμερα τα δίκτυα υπολογιστών και τα δίκτυα κινητής τηλεφωνίας.

Τα συστήματα τηλεπικοινωνίας μέχρι πριν κάποια χρόνια βασίζονταν κυρίως στην μετάδοση πληροφορίας με αναλογικό τρόπο. Πλέον στο προσκήνιο έχουν βρεθεί τα ψηφιακά συστήματα επικοινωνιών. Τα θεωρητικά θεμέλια για τις ψηφιακές επικοινωνίες τέθηκαν από τον Claude Shannon το 1948. Οι τεχνολογικές δυνατότητες της εποχής του καθιστούσαν αδύνατη την ευρεία εφαρμογή αυτών των ιδεών. Η πρόοδος που σημειώθηκε στις τεχνολογίες ημιαγωγών τα επόμενα χρόνια έκανε εφικτή την υλοποίηση τέτοιου είδους συστημάτων.

Τα ψηφιακά συστήματα επικοινωνιών ελκύουν όλο και περισσότερο ενδιαφέρον λόγω των συνεχώς αυξανόμενων αναγκών για μεταφορά δεδομένων και επειδή η ψηφιακή μετάδοση προσφέρει ικανότητες επεξεργασίας σήματος που δεν ήταν εφικτές πριν από μερικά χρόνια. Αυτό που διαφοροποιεί τα ψηφιακά συστήματα από τα αναλογικά είναι πως στα ψηφιακά η προς μετάδοση πληροφορία κωδικοποιείται σε μια ακολουθία δυαδικών ψηφίων. Αυτό περιορίζει το πλήθος των πιθανών κυματομορφών που μπορεί να αποσταλούν από τον πομπό σε έναν περιορισμένο αριθμό. Ο σκοπός στον δέκτη δεν είναι αναπαραχθεί με ακρίβεια η ληφθείσα κυματομορφή, αντιθέτως ο σκοπός είναι να καθοριστεί από μια παραμορφωμένη λόγω θορύβου κυματομορφή ποια από το πεπερασμένο πλήθος κυματομορφών ήταν αυτή που στάλθηκε από τον πομπό.

Το σημαντικότερο πλεονέκτημα ενός ψηφιακού συστήματος είναι η ικανότητα αναγέννησης σήματος. Καθώς οι κυματομορφές διαδίδονται κατά μήκους ενός μέσου μεταφοράς υφίστανται παραμόρφωση λόγω δύο βασικών αιτιών: χαρακτηριστικά του μέσου και ανεπιθύμητος ηλεκτρικός θόρυβος. Αν κατά μήκος της γραμμής μετάδοσης, όσο είναι ακόμα αναγνωρίσιμος ο παλμός, τοποθετηθεί μια ψηφιακή συσκευή (αναγεννητής) ο παλμός μπορεί να «αναγεννηθεί». Αυτό είναι εφικτό μόνο στα ψηφιακά συστήματα τα οποία λειτουργούν σε δύο καταστάσεις (1 και 0). Στα αναλογικά σήματα ακόμη και μια μικρή παραμόρφωση μπορεί να καταστήσει το σήμα

ανεπανόρθωτα παραμορφωμένο. Η παραμόρφωση αυτή είναι αδύνατο να αφαιρεθεί. Με τη χρήση ψηφιακών μεθόδων, πολύ χαμηλότεροι ρυθμοί λαθών μπορούν να επιτευχθούν μέσω εντοπισμού και διόρθωσης λαθών.



Εικόνα 1.1 Ψηφιακό σύστημα επικοινωνίας

Στην Εικόνα 1.1 απεικονίζεται το λειτουργικό διάγραμμα και τα βασικά στοιχεία ενός σύγχρονου ψηφιακού τηλεπικοινωνιακού συστήματος. Η προς μετάδοση πληροφορία μπορεί να είναι είτε ένα αναλογικό σήμα όπως ήχος, εικόνα ή βίντεο είτε ένα ψηφιακό όπως τα δεδομένα υπολογιστή. Αυτή η πληροφορία πρέπει να απεικονιστεί με όσο το δυνατότερο λιγότερο δυαδικά ψηφία. Την χαρτογράφηση της πληροφορίας σε μια ακολουθία δυαδικών ψηφίων (ψηφιακό σήμα) αναλαμβάνει ο κωδικοποιητής πληροφορίας.

Στη συνέχεια το ψηφιακό σήμα οδηγείται στον κωδικοποιητή καναλιού. Ο σκοπός του κωδικοποιητή καναλιού είναι να εισάγει, με ελεγχόμενο τρόπο, επιπλέον ψηφία στην δυαδική πληροφορία τα οποία μπορούν να χρησιμοποιηθούν στον δέκτη για να ξεπεραστούν οι συνέπειες του θορύβου και των παρεμβολών που απαντώνται κατά την μετάδοση του σήματος μέσω του καναλιού. Ως εκ τούτου, η πρόσθετη πλεονάζουσα πληροφορία συνεισφέρει στην αύξηση της αξιοπιστίας των λαμβανόμενων δεδομένων. Επί της ουσίας, τα πλεονάζοντα ψηφία στην πληροφορία βοηθούν τον δέκτη να αποκωδικοποιήσει την επιθυμητή ακολουθία πληροφορίας. Για παράδειγμα, μια τετριμμένη μορφή κωδικοποίησης του ψηφιακού σήματος είναι απλά να επαναληφθεί m φορές κάθε ψηφίο, όπου m ένας θετικός ακέραιος. Μια πιο εκλεπτυσμένη μέθοδος κωδικοποίησης, είναι ο χωρισμός της πληροφορίας σε επιμέρους τμήματα k ψηφίων και η χαρτογράφηση κάθε ακολουθίας k ψηφίων σε μια μοναδική ακολουθία n ψηφίων, η οποία αποκαλείται κωδικολέξη. Η ποσότητα της πλεονάζουσας πληροφορίας που εισάγεται από την κωδικοποίηση των δεδομένων με αυτόν τον τρόπο μετράται από την ποσότητα n/k . Ο αντίστροφος αυτής της ποσότητας ονομάζεται ρυθμός κώδικα.

Η κωδικοποιημένη ακολουθία της εξόδου του κωδικοποιητή καναλιού οδηγείται στον διαμορφωτή, ο οποίος δρα σαν διεπαφή για το κανάλι. Αφού σχεδόν όλα τα τηλεπικοινωνιακά κανάλια που υπάρχουν στην πράξη είναι ικανά να μεταδίδουν ηλεκτρικά σήματα (κυματομορφές), ο κύριος σκοπός αυτού του στοιχείου είναι να χαρτογραφεί την δυαδική πληροφορία σε κυματομορφές. Ας υποθεθεί, πως ένα ψηφίο της κωδικοποιημένης ακολουθίας μεταδίδεται κάθε φορά με έναν συνολικό ρυθμό μετάδοσης R ψηφία/sec. Ο διαμορφωτής τότε πρέπει να χαρτογραφήσει το δυαδικό ψηφίο 0 σε μια κυματομορφή $s_0(t)$ και το δυαδικό ψηφίο 1 σε μια κυματομορφή $s_1(t)$. Με αυτόν τον τρόπο, κάθε ψηφίο της κωδικοποιημένης ακολουθίας μεταδίδεται ξεχωριστά. Αυτό αποκαλείται δυαδική διαμόρφωση. Εναλλακτικά, ο διαμορφωτής μπορεί να μεταδίδει b ψηφία πληροφορίας σε κάθε βήμα χρησιμοποιώντας $M=2^b$ κυματομορφές. Μια τέτοια διαμόρφωση ονομάζεται M -αδική. Να σημειωθεί πως σε αυτή την περίπτωση μια ακολουθία ψηφίων μήκους b εισέρχεται στον διαμορφωτή κάθε b/R δευτερόλεπτα. Έτσι όταν το bitrate του καναλιού είναι

σταθερό, το χρονικό διάστημα που χρειάζεται για να μεταδοθεί μια κυματομορφή της M-αδικής διαμόρφωσης είναι ίσο με το διάστημα που χρειάζεται για να μεταδοθούν b κυματομορφές της δυαδικής διαμόρφωσης.

Το κανάλι επικοινωνίας είναι το φυσικό μέσο που χρησιμοποιείται για να αποσταλεί το σήμα από τον πομπό στον δέκτη. Στις ασύρματες επικοινωνίες, το κανάλι μπορεί να είναι η ατμόσφαιρα. Από την άλλη, ενσύρματες επικοινωνίες, όπως το τηλέφωνο, χρησιμοποιούν μια πληθώρα φυσικών μέσων, όπως τα χάλκινα καλώδια και οι οπτικές ίνες. Οποιοδήποτε και αν είναι το φυσικό μέσο που χρησιμοποιείται για την μετάδοση της πληροφορίας, το μεταδιδόμενο σήμα υφίσταται μια σειρά από τυχαίες παραμορφώσεις λόγω μηχανισμών, όπως ο θερμικός θόρυβος των ηλεκτρικών συσκευών και οι ηλεκτρικές εκκενώσεις (κεραυνοί) κατά τις καταιγίδες.

Στο άκρο του δέκτη ενός ψηφιακού τηλεπικοινωνιακού συστήματος, ο αποδιαμορφωτής επεξεργάζεται τις παραμορφωμένες κυματομορφές που λαμβάνει και μετατρέπει τις κυματομορφές σε μια σειρά από ψηφία τα οποία αντιπροσωπεύουν τις εκτιμήσεις για τα μεταδιδόμενα σήματα. Αυτή η σειρά ψηφίων οδηγείται στο αποκωδικοποιητή καναλιού, ο οποίος αποπειράται να ανακτήσει το αρχικό ψηφιακό σήμα έχοντας γνώση της κωδικοποίησης που χρησιμοποιείται από τον κωδικοποιητή καναλιού και αξιοποιώντας την πλεονάζουσα πληροφορία που έχει εισαχθεί από τον κωδικοποιητή.

Ένα μέτρο της καλής λειτουργίας του αποκωδικοποιητή και του αποδιαμορφωτή είναι η συχνότητα με την οποία συμβαίνουν λάθη στην αποκωδικοποιημένη ακολουθία. Ακριβέστερα, η μέση πιθανότητα λάθους ανά ψηφίο είναι το μέτρο της απόδοσης του συστήματος αποδιαμορφωτή-αποκωδικοποιητή. Γενικά, η πιθανότητα σφάλματος είναι μια συνάρτηση των χαρακτηριστικών του κώδικα, των τύπων των κυματομορφών που χρησιμοποιούνται για την μετάδοση της πληροφορίας, των χαρακτηριστικών του καναλιού και της μεθόδου αποδιαμόρφωσης και αποκωδικοποίησης.

Τέλος, όταν είναι επιθυμητό η έξοδος να είναι αναλογική, ο αποκωδικοποιητής πληροφορίας μετατρέπει την ψηφιακή πληροφορία σε αναλογική και ανακατασκευάζει την αρχική πληροφορία. Λόγω πιθανών σφαλμάτων κατά την αποκωδικοποίηση καναλιού και των παραμορφώσεων που πιθανώς εισάγονται από τον κωδικοποιητή πληροφορίας και τον αποκωδικοποιητή πληροφορίας, το σήμα στην έξοδο εμφανίζει μικρές αποκλίσεις από την αρχική αναλογική πληροφορία.

1.2 Διόρθωση σφαλμάτων και κωδικοποίηση

Η θεμελιώδης ιδέα της θεωρίας πληροφορίας είναι ότι όλες οι επικοινωνίες είναι εν δυνάμει ψηφιακές – δηλαδή μπορούν να αντικατασταθούν από ένα μοντέλο παραγωγής μεταφοράς και λήψης δυαδικών ψηφίων, bits. Όταν αυτά τα bits μεταδίδονται σε ένα κανάλι επικοινωνίας είναι πιθανόν πως κάποια από αυτά θα καταστραφούν από θόρυβο. Στην δημοσίευση-ορόσημο του 1948 «Μια Μαθηματική Θεωρία Πληροφορίας» ο Claude Shannon αναγνώρισε ότι τυχαία επιλεγμένα δυαδικά ψηφία μπορούσαν να χρησιμοποιηθούν για τον έλεγχο της γέννησης, μετάδοσης και λήψης της πληροφορίας. Επίσης, έδειξε πως το πρόβλημα μετάδοσης της πληροφορίας από την πηγή μέσω του καναλιού στον προορισμό μπορεί να χωριστεί στα εξής δύο προβλήματα: την αναπαράσταση της πηγαίας πληροφορίας όσο το δυνατόν καλύτερα σε μια δυαδική ακολουθία (κωδικοποίηση πληροφορίας) και την μετάδοση δυαδικών, τυχαίων ανεξαρτήτων ψηφίων στον κανάλι (κωδικοποίηση καναλιού). Έτσι ο κωδικοποιητής όπως είδαμε και στην Εικόνα 1.1 χωρίζεται σε δυο μέρη με βάση την διαχωριστική αρχή του Shannon, τον κωδικοποιητή πληροφορίας (πηγής) και τον κωδικοποιητή καναλιού. Αυτό απλοποιεί την χρήση του ίδιου καναλιού επικοινωνίας από διαφορετικές πηγές πληροφορίας.

Το θεώρημα του Shannon για την κωδικοποίηση αναφέρει ότι κάθε επικοινωνιακό κανάλι χαρακτηρίζεται από μια μοναδική παράμετρο C_t , τη χωρητικότητα του καναλιού, τέτοια ώστε R_t τυχαία επιλεγμένα bits ανά δευτερόλεπτο μπορούν να μεταδοθούν αξιόπιστα αν και μόνο αν $R_t \leq C_t$. Ο Shannon έδειξε ότι η τιμή του σηματοθορυβικού λόγου δεν είναι σημαντική όσο είναι αρκετά μεγάλη. Αυτό που έχει σημασία είναι ο τρόπος που κωδικοποιούνται τα ψηφία πληροφορίας. Η πληροφορία δε θα έπρεπε να μεταδίδεται με ένα bit πληροφορίας τη φορά, αλλά μεγάλες ακολουθίες πληροφορίας θα έπρεπε να κωδικοποιούνται έτσι ώστε ένα bit πληροφορίας να έχει κάποια επιρροή σε πολλά bit που μεταδίδονται μέσω του καναλιού. Αυτή η δραστικά καινούργια ιδέα γέννησε τη θεωρία κωδικοποίησης.

Η κωδικοποίηση ελέγχου σφαλμάτων πρέπει να προστατεύει την ψηφιακή πληροφορία από τυχόν σφάλματα κατά τη μετάδοση λόγω θορύβου. Κατά τις τελευταίες δεκαετίες έχει παρατηρηθεί μια μεγάλη αύξηση στις τεχνολογικές δυνατότητες που μας επιτρέπει να υιοθετούμε πολύπλοκους αλγορίθμους για τον έλεγχο λαθών.

Οι δύο σημαντικότερες κατηγορίες κωδίκων είναι οι μπλοκ κώδικες και οι συνελκτικοί κώδικες.

1.3 Μπλοκ κώδικες

Ένας μπλοκ κώδικας είναι ένας κώδικας στον οποίον κάθε εισερχόμενη ακολουθία πληροφορίας τέμνεται σε τμήματα κάποιου δοσμένου μήκους k και έπειτα αυτά τα μήκους k μηνύματα χαρτογραφούνται-αντιστοιχίζονται σε κωδικολέξεις. Στο σύνολο υπάρχουν $M=2^k$ πιθανά διαφορετικά μηνύματα. Ο κωδικοποιητής μετατρέπει κάθε μήνυμα σε μια δυαδική ακολουθία, την κωδικολέξη, μήκους n με $n > k$. Ως εκ τούτου, αντίστοιχα με τα 2^k πιθανά διαφορετικά μηνύματα, υπάρχουν και 2^k πιθανές διαφορετικές κωδικολέξεις στην έξοδο του κωδικοποιητή. Το σύνολο αυτών των κωδικολέξεων ονομάζεται (n,k) δυαδικός μπλοκ κώδικας. Ο ρυθμός κωδικοποίησης του κώδικα είναι:

$$R = \frac{\log M}{n} = \frac{k}{n} \quad (1.1)$$

και μετριέται σε bits πληροφορίας ανά χρήση καναλιού. Ο ρυθμός μετάδοσης δεδομένων σε bits πληροφορίας ανά δευτερόλεπτο υπολογίζεται διαιρώντας τον ρυθμό κωδικοποίησης R με την περίοδο αποστολής συμβόλων T (στην περίπτωση τις δυαδικής διαμόρφωσης):

$$R_t = \frac{R}{T} \quad (1.2)$$

Ο ρυθμός $R = k/n$ εκφράζει το τμήμα των ψηφίων στην κωδικολέξη που είναι αναγκαία για την αναπαράσταση της πληροφορίας. Το εναπομείναν τμήμα $1 - R = (n - k)/n$ αντιπροσωπεύει τα πλεονάζοντα ψηφία που μπορούν να χρησιμοποιηθούν για να ανιχνευθούν ή να διορθωθούν σφάλματα. Από τη στιγμή που η μήκους n κωδικολέξη εξαρτάται μόνο από το αντίστοιχο μήνυμα μήκους k στην είσοδο, ο κωδικοποιητής είναι άνευ μνήμης και μπορεί να υλοποιηθεί από ένα συνδυαστικό λογικό κύκλωμα.

Μερικά είδη μπλοκ κωδίκων που χρησιμοποιούνται ευρέως είναι:

- Κώδικες Hamming: Αυτοί οι γραμμικοί μπλοκ κώδικες είναι ικανοί να διορθώσουν σφάλματα ενός ψηφίου να ανιχνεύσουν σφάλματα δύο ψηφίων. Παραδείγματα τέτοιου κώδικα είναι οι Hamming (7,4) και Hamming (15,11).
- Κώδικες Golay: Οι κώδικες Golay είναι μπλοκ κώδικες με δυνατότητα διόρθωσης σφάλματος τριών ψηφίων. Παραδείγματα περιλαμβάνουν τους κώδικες Golay (23,12) και Golay (24,12).
- Κώδικες BCH (Bose-Chaudhuri-Hocquenghem): Οι κώδικες BCH είναι μια κατηγορία κυκλικών μπλοκ κωδίκων γνωστών για τη δυνατότητά τους να διορθώσουν πολλά

σφάλματα. Χρησιμοποιούνται σε διάφορες εφαρμογές επικοινωνίας και αποθήκευσης δεδομένων.

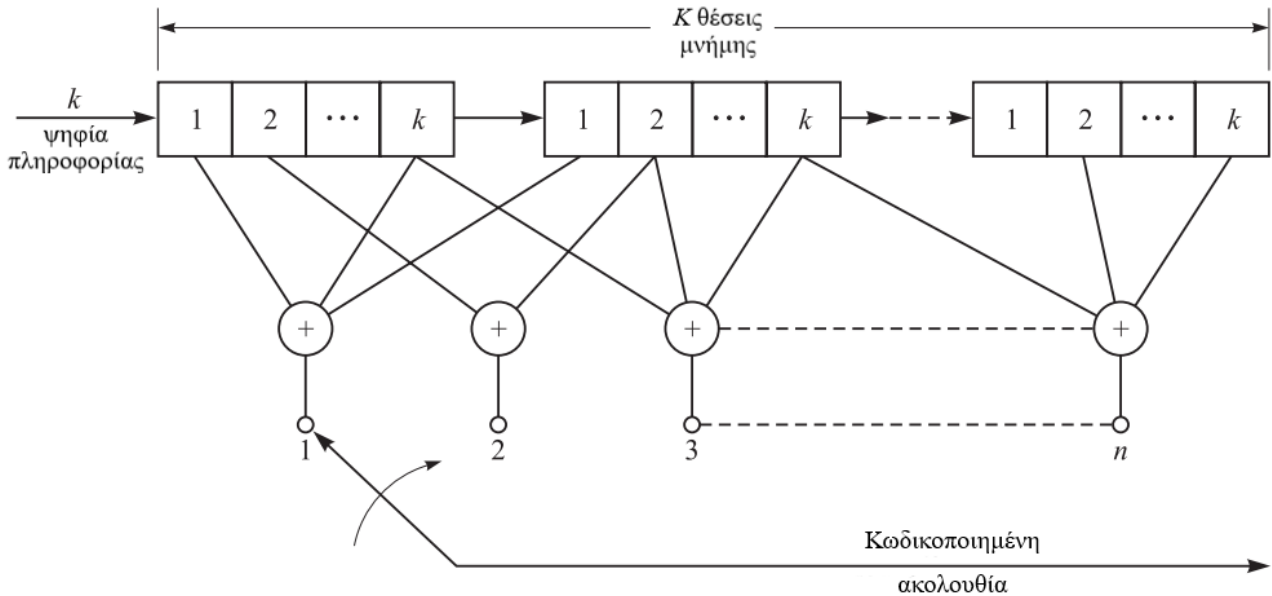
- Κώδικες Reed-Solomon: Αποτελούν υποπερίπτωση των κωδίκων BCH. Οι κώδικες Reed-Solomon είναι ευέλικτοι μπλοκ κώδικες που χρησιμοποιούνται ευρέως σε συστήματα αποθήκευσης και μετάδοσης δεδομένων. Μπορούν να διορθώσουν ένα συγκεκριμένο αριθμό σφαλμάτων και να ανιχνεύσουν ακόμη περισσότερα. Οι κώδικες Reed-Solomon είναι γνωστοί για την ισχυρή ικανότητα διόρθωσης σφαλμάτων και χρησιμοποιούνται συχνά σε εφαρμογές όπως CD, DVD και QR κώδικες.
- Κώδικες Turbo: Οι κώδικες Turbo είναι μια σύγχρονη και ισχυρή κατηγορία κωδικών μπλοκ που χρησιμοποιούν μια συνενωμένη δομή. Χρησιμοποιούνται ευρέως σε συστήματα ασύρματης επικοινωνίας και είναι γνωστοί για την εξαιρετική δυνατότητα διόρθωσης σφαλμάτων.

1.4 Συνελικτικοί κώδικες

Εν αντιθέσει με τους μπλοκ κώδικες, οι συνελικτικοί κώδικες δεν έχουν στην ουσία τμηματική δομή. Ακόμη, διαφέρουν από τους μπλοκ κώδικες στο γεγονός πως ο κωδικοποιητής έχει μνήμη και οι n έξοδοι του αποκωδικοποιητή δεν εξαρτώνται μόνο από τις τρέχουσες k εισόδους αλλά και από τα $K-1$ προηγούμενα μπλοκ εισόδου μήκους k . Ένας τέτοιος κώδικας ονομάζεται (n,k,K) συνελικτικός κώδικας και παράγει n bit εξόδου για k bit εισόδου. Ένας τέτοιος κώδικας παράγεται εισάγοντας την ψηφιακή ακολουθία πληροφορίας μέσα από έναν καταχωρητή ολίσθησης που αποτελείται από K στάδια (των k bit) και n γραμμικούς γεννήτορες (η πρακτική υλοποίηση τους γίνεται συχνά με modulo-2 αθροιστές). Σε κάθε βήμα η ακολουθία εισόδου «ολισθαίνει» κατά k bits εντός του καταχωρητή. Ο ρυθμός κωδικοποίησης ενός (n,k,K) συνελικτικού κώδικα είναι $R = k/n$ ενώ η παράμετρος K αποκαλείται μήκος περιορισμού του κώδικα.

Οι συνελικτικοί κώδικες παρουσιάστηκαν για πρώτη φορά από τον Elias το 1955 σαν μια εναλλακτική στους μπλοκ κώδικες. Λίγο μετά ο Wozencraft πρότεινε την σειριακή αποκωδικοποίηση σαν έναν αποτελεσματικό τρόπο αποκωδικοποίησης και πειραματικές μελέτες άρχισαν να εμφανίζονται. Το 1963, ο Massey πρότεινε μια λιγότερο αποδοτική αλλά ευκολότερη προς υλοποίηση μέθοδο αποκωδικοποίησης, η οποία αποκαλείται αποκωδικοποίηση κατωφλίου. Αυτή η καινοτομία γέννησε μια πληθώρα πρακτικών εφαρμογών των συνελικτικών κωδίκων στην ψηφιακή μετάδοση σε ενσύρματα και ασύρματα μέσα. Έπειτα, το 1967, ο Viterbi πρότεινε έναν αλγόριθμο μέγιστης πιθανοφάνειας για την αποκωδικοποίηση ο οποίος ήταν εύκολο να υλοποιηθεί

για κώδικες με μικρή μήμη. Αυτή η μέθοδος, η αποκαλούμενη αποκωδικοποίηση Viterbi, μαζί με τις βελτιωμένες μεθόδους σειριακής αποκωδικοποίησης οδήγησαν στην χρήση συνελκτικών κωδίκων για διαστημικές και δορυφορικές επικοινωνίες κατά την δεκαετία του 1970.



1.5 Αποκωδικοποίηση και συνελκτικοί κώδικες

Ας υποτεθεί πως ένα κώδικα $\mathbf{m} = m_0 m_1 m_2 \dots m_{K-1}$ αντιστοιχίζεται σε μια κωδικολέξη $\mathbf{u} = u_0 u_1 u_2 \dots u_{N-1}$. Η έξοδος του καναλιού $\mathbf{r} = r_0 r_1 r_2 \dots r_{N-1}$ αποκαλείται ληφθείσα ακολουθία. Ο αποκωδικοποιητής μετατρέπει την ληφθείσα ακολουθία μήκους N, η οποία πιθανόν να έχει παραμορφωθεί από θόρυβο, σε μια ακολουθία μήκους K, την $\tilde{\mathbf{m}}$, η οποία είναι η εκτιμώμενη ακολουθία. Ιδανικά η $\tilde{\mathbf{m}}$ είναι ένα αντίγραφο της αρχικής ακολουθίας \mathbf{m} , όμως ο θόρυβος μπορεί να προκαλέσει μερικά σφάλματα αποκωδικοποίησης. Αν η κωδικολέξη \mathbf{u} έχει αποσταλεί, σφάλμα αποκωδικοποίησης συμβαίνει αν και μόνο αν $\tilde{\mathbf{m}} \neq \mathbf{m}$.

Αν η P_E συμβολίζει την πιθανότητα σφάλματος, τότε αυτή δίνεται από τη σχέση

$$P_E = \sum_{\mathbf{r}} P(\tilde{\mathbf{m}} \neq \mathbf{m} | \mathbf{r}) P(\mathbf{r}) \quad (1.3)$$

όπου η πιθανότητα ο δέκτης να λάβει \mathbf{r} , $P(\mathbf{r})$, είναι ανεξάρτητη της μεθόδου αποκωδικοποίησης και $P(\tilde{\mathbf{m}} \neq \mathbf{m} | \mathbf{r})$ είναι η δεσμευμένη πιθανότητα να έχει συμβεί λάθος στην αποκωδικοποίηση δεδομένου ότι ελήφθη η ακολουθία \mathbf{r} . Άρα για να ελαχιστοποιηθεί η πιθανότητα σφάλματος P_E , ο αποκωδικοποιητής πρέπει να σχεδιαστεί έτσι ώστε να επιλέγει ως έξοδο του το $\tilde{\mathbf{m}}$ για το οποίο η $P(\tilde{\mathbf{m}} \neq \mathbf{m} | \mathbf{r})$ ελαχιστοποιείται για ένα δεδομένο \mathbf{r} (ή ισοδύναμα μεγιστοποιείται η

$P(\tilde{\mathbf{m}} = \mathbf{m}|\mathbf{r}) \equiv P(\mathbf{m}|\mathbf{r})$ για ένα δεδομένο \mathbf{r}). Ένας τέτοιος αποκωδικοποιητής ονομάζεται αποκωδικοποιητής MAP (Maximum a posteriori).

Από το θεώρημα του Bayes γνωρίζουμε πως :

$$P(\mathbf{m}|\mathbf{r}) = \frac{P(\mathbf{r}|\mathbf{m})P(\mathbf{m})}{P(\mathbf{r})}$$

Αν όλες οι ακολουθίες πληροφορίας, άρα και οι κωδικολέξεις, είναι ισοπίθανες, τότε η πιθανότητα $P(\mathbf{m})$ είναι ίδια για όλα τα \mathbf{m} , άρα η μεγιστοποίηση της $P(\mathbf{m}|\mathbf{r})$ είναι ισοδύναμη της μεγιστοποίησης της $P(\mathbf{r}|\mathbf{m})$. Η μεγιστοποίηση της $P(\mathbf{r}|\mathbf{m})$ είναι ισοδύναμη της μεγιστοποίησης της $P(\mathbf{r}|\mathbf{u})$ από τη στιγμή που μια και μόνο κωδικολέξη αντιστοιχεί σε κάθε ακολουθία πληροφορίας. Για ένα διακριτό κανάλι άνευ μνήμης (Discrete memoryless channel, DMC) ισχύει πως

$$P(\mathbf{r}|\mathbf{u}) = \prod_i P(r_i|u_i) \quad (1.4)$$

από την στιγμή που για ένα κανάλι χωρίς μνήμη κάθε ληφθέν σύμβολο εξαρτάται μόνο από το αντίστοιχο μεταδιδόμενο. Ένας κωδικοποιητής που έχει σχεδιαστεί έτσι ώστε να μεγιστοποιεί την ποσότητα (1.4) ονομάζεται αποκωδικοποιητής μέγιστης πιθανοφάνειας ή ML (maximum likelihood). Αν οι ακολουθίες πληροφορίας δεν είναι ισοπίθανες, τότε ο αποκωδικοποιητής ML δεν είναι σίγουρα ο βέλτιστος, αφού στις δεσμευμένες πιθανότητες $P(\mathbf{r}|\mathbf{m})$ πρέπει να ανατεθούν βάρη από τις πιθανότητες ακολουθίας πληροφορίας $P(\mathbf{m})$ ώστε να προσδιοριστεί ποια ακολουθία πληροφορίας μεγιστοποιεί την $P(\mathbf{m}|\mathbf{r})$.

Ακόμη, επειδή η λογαριθμική συνάρτηση είναι μονότονη, η μεγιστοποίηση της (1.4) είναι ισοδύναμη της μεγιστοποίησης της:

$$\log P(\mathbf{r}|\mathbf{u}) = \sum_i \log P(r_i|u_i) \quad (1.5)$$

Οι αποκωδικοποιητές ML βρίσκουν πιο συχνά εφαρμογή στους συνελκτικούς κώδικες από ότι στους μπλοκ (αν και θεωρητικά είναι καταλληλότεροι για διόρθωση λαθών). Ο κυριότερος λόγος που συμβαίνει αυτό είναι πως έχουν μεγάλη επεξεργαστική πολυπλοκότητα και ανάγκες σε hardware, η οποία ανεβαίνει κατακόρυφα καθώς αυξάνεται το μήκος των μπλοκ.

Ένας ευρέως διαδεδομένος αποκωδικοποιητής μέγιστης πιθανοφάνειας συνελκτικών κωδίκων είναι ο αποκωδικοποιητής Viterbi, του οποίου η λειτουργία βασίζεται στον ομώνυμο αλγόριθμο.

Πέραν του αλγορίθμου Viterbi, υπάρχουν και άλλοι αλγόριθμοι που δεν βασίζονται σε μεθόδους μέγιστης πιθανοφάνειας. Μερικοί από αυτούς είναι:

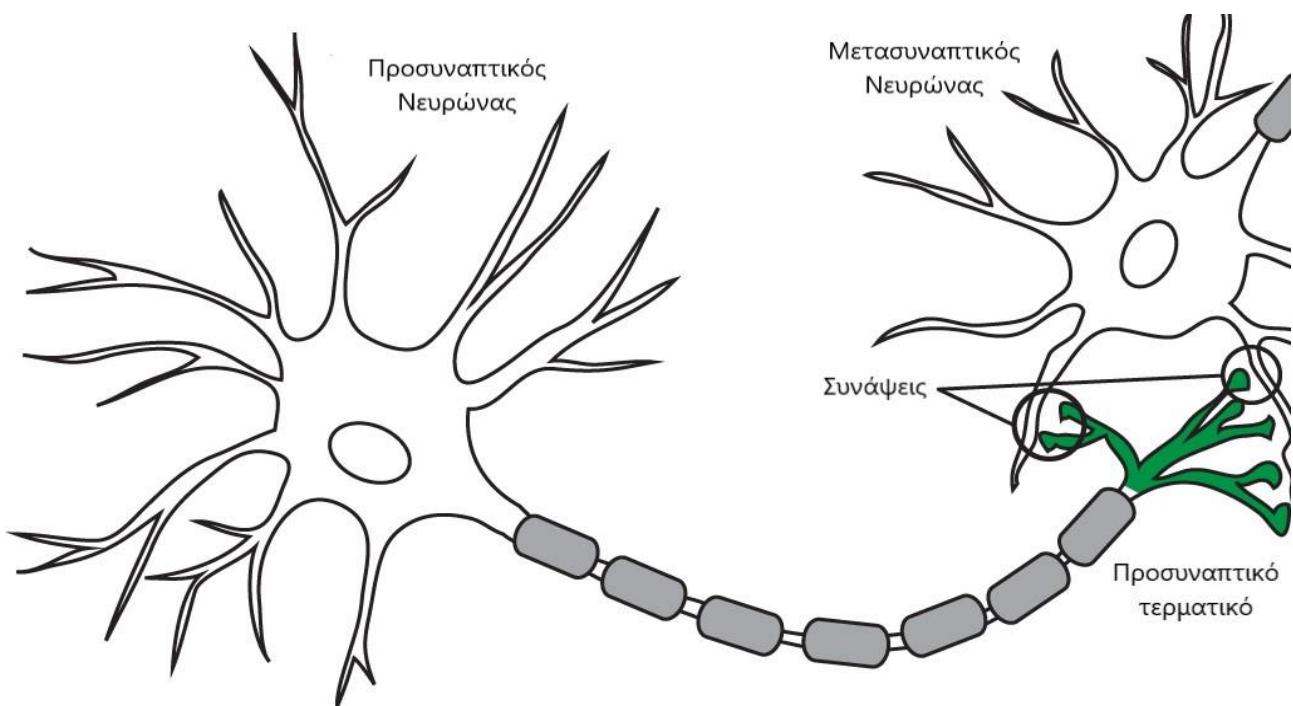
- Αλγόριθμος Fano: Αλγόριθμος σειριακής αποκωδικοποίησης, ο οποίος βασίζεται σε μια δομή δένδρου για να βρει την πιθανότερη αλληλουχία ψηφίων
- Αλγόριθμος Stack: Αλγόριθμος σειριακής αποκωδικοποίησης, ο οποίος βασίζεται σε μια δομή στοίβας για να βρει την πιθανότερη αλληλουχία ψηφίων

Αυτοί οι αλγόριθμοι που προαναφέρθηκαν έχουν μικρότερη υπολογιστική πολυπλοκότητα και μπορούν να χρησιμοποιηθούν όταν το hardware είναι χαμηλών δυνατοτήτων. Όμως ο αλγόριθμος Viterbi έχει βέλτιστες επιδόσεις ως προς την αξιοπιστία της αποκωδικοποίησης.

1.6 Μηχανική Μάθηση και Νευρωνικά Δίκτυα

Οι επιστήμονες και οι εφευρέτες είχαν από τους αρχαίους χρόνους ονειρευτεί την δημιουργία μηχανών με ικανότητα σκέψης. Οι μυθικές φιγούρες του Πυγμαλίωνα, του Δαίδαλου και του Ήφαιστου μπορούν να θεωρηθούν ως μυθικοί εφευρέτες, και της Γαλατείας, του Τάλω και της Πανδώρας ως τεχνητή ζωή.

Όταν συνελήφθη για πρώτη φορά η ιδέα των προγραμματιζόμενων υπολογιστών, οι άνθρωποι αναρωτήθηκαν αν θα μπορούσαν αυτές οι μηχανές να γίνουν τόσο «έξυπνες» (Lovelace, 1842). Σήμερα, η τεχνητή νοημοσύνη (AI) είναι ένα ακμάζον πεδίο με πολλές πρακτικές εφαρμογές και



Εικόνα 1.2 Βιολογικοί νευρώνες και συνάψεις

ενεργά θέματα έρευνας. Τέτοιου είδους λογισμικά αυτοματοποιούν την καθημερινή εργασία, κατανοούν λόγο και εικόνα και υποστηρίζουν την επιστημονική μελέτη.

Τα τεχνητά νευρωνικά δίκτυα είναι δημοφιλείς τεχνικές μηχανικής μάθησης, οι οποίες προσομοιάζουν τους μηχανισμούς μάθησης των βιολογικών οργανισμών. Το ανθρώπινο νευρικό σύστημα περιέχει κύτταρα, τα οποία αναφέρονται ως νευρώνες. Οι νευρώνες συνδέονται μεταξύ τους μέσω των συνάψεων. Η ισχύς των συνάψεων συχνά αλλάζει σε απόκριση στα εξωτερικά ερεθίσματα. Αυτή η απόκριση είναι ο τρόπος μάθησης των ζωντανών οργανισμών.

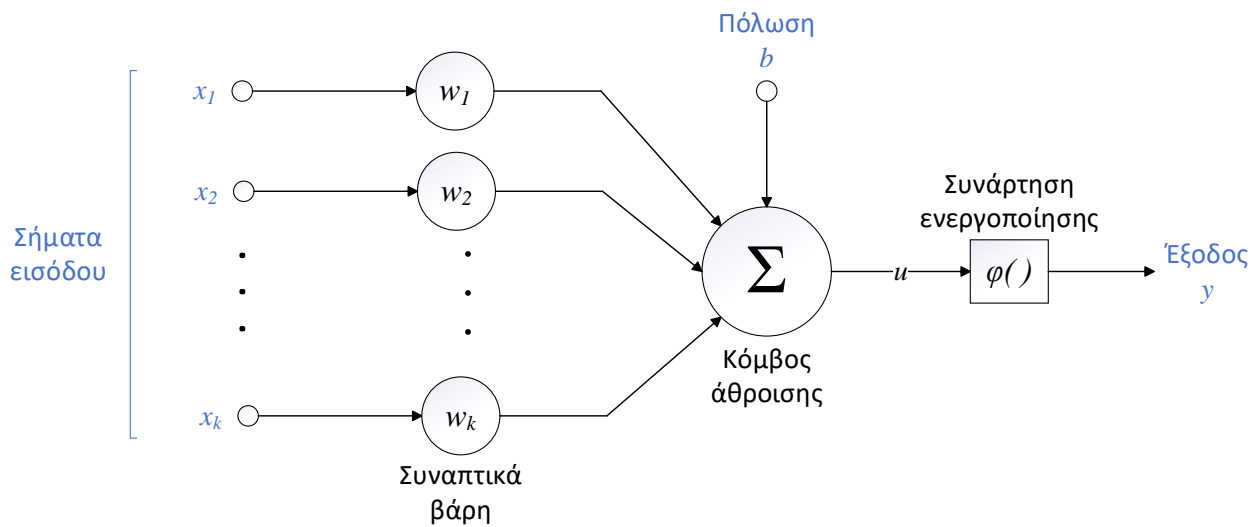
Τα βιολογικά νευρικά συστήματα αποτελούν την έμπνευση για τα τεχνητά, τα οποία περιέχουν υπολογιστικές μονάδες οι οποίοι ονομάζονται νευρώνες. Οι νευρώνες αποτελούν την βάση για την σχεδίαση απλών και πολύπλοκων νευρωνικών δικτύων. Το σχηματικό διάγραμμα της Εικόνα 1.3 παρουσιάζει το μοντέλο ενός νευρώνα. Τα βασικά στοιχεία της μοντέλου λειτουργίας είναι τα εξής:

- Ένα σύνολο συνάψεων, κάθε μία εκ των οποίων χαρακτηρίζεται από το δικό της βάρος w_i . Κάθε σήμα εισόδου x_i που εισέρχεται μέσω της σύναψης i στον νευρώνα, πολλαπλασιάζεται επί το συναπτικό βάρος. Οι τιμές του βάρους μπορεί να είναι και θετικές και αρνητικές. Με την αλλαγή των συναπτικών βαρών των νευρώνων, επιτυγχάνεται η διαδικασία μάθησης.
- Έναν αθροιστή για την άθροιση των σημάτων εισόδου, σταθμισμένα από τα αντίστοιχα βάρη. Ως εδώ η λειτουργία του μοντέλου είναι γραμμική.
- Μια συνάρτηση ενεργοποίησης για τον περιορισμό του πλάτους του σήματος εξόδου. Περιορίζει το επιτρεπτό εύρος του πλάτους της εξόδου σε μια πεπερασμένη τιμή. Αυτό το στοιχείο εισάγει και την έννοια της μη γραμμικότητας στον νευρώνα
- Πολλές φορές στον νευρώνα εφαρμόζεται μια εξωτερική πόλωση b . Η πόλωση έχει ως αποτέλεσμα την αύξηση ή την μείωση της διέγερσης της συνάρτησης ενεργοποίησης, ανάλογα με την τιμή της

Με τη χρήση μαθηματικών, η λειτουργία του νευρώνα μπορεί να περιγραφεί με τις εξής δύο εξισώσεις:

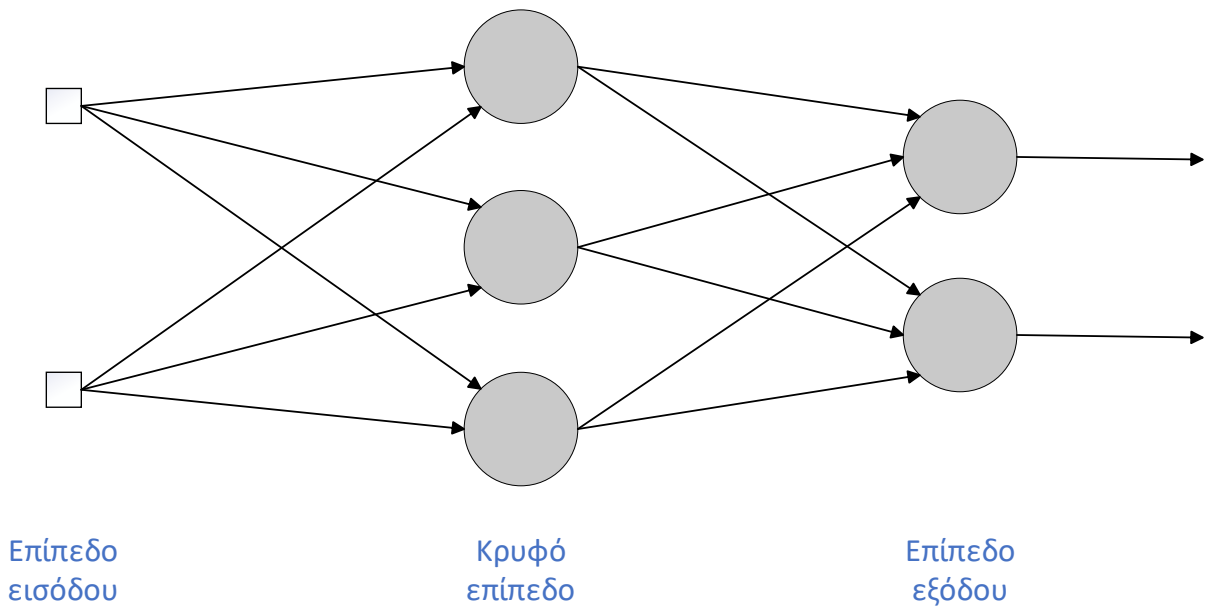
$$\mathbf{u} = \sum_{i=1}^k \mathbf{w}_i x_i + \mathbf{b} \quad (1.6)$$

$$y = \varphi(u) \quad (1.7)$$



Εικόνα 1.3 Το μοντέλο ενός νευρώνα

Πολλοί τέτοιου είδους νευρώνες συνδέονται μεταξύ τους για να δημιουργήσουν ένα νευρωνικό δίκτυο. Ο τρόπος με τον οποίο είναι δομημένοι οι νευρώνες ενός νευρωνικού δικτύου σχετίζεται στενά με τον αλγόριθμο μάθησης που χρησιμοποιείται για την εκπαίδευση του δικτύου. Ένα σχετικά απλό νευρωνικό δίκτυο πρόσθιας τροφοδότησης με ένα κρυφό επίπεδο φαίνεται στην Εικόνα 1.4.



Εικόνα 1.4 Νευρωνικό Δίκτυο

1.7 Κίνητρα της παρούσης εργασίας

Στο δυναμικά εξελισσόμενο πεδίο των ψηφιακών συστημάτων επικοινωνίας, η αναζήτηση πιο αποτελεσματικών και αξιόπιστων τεχνικών διόρθωσης σφαλμάτων είναι μια συνεχής πρόκληση. Η συνελκτική κωδικοποίηση έχει αναδειχθεί ως μια ευρέως υιοθετημένη μέθοδος, επιδεικνύοντας την αποτελεσματικότητά της στην ενίσχυση της αξιοπιστίας των μεταδιδόμενων μηνυμάτων. Ο

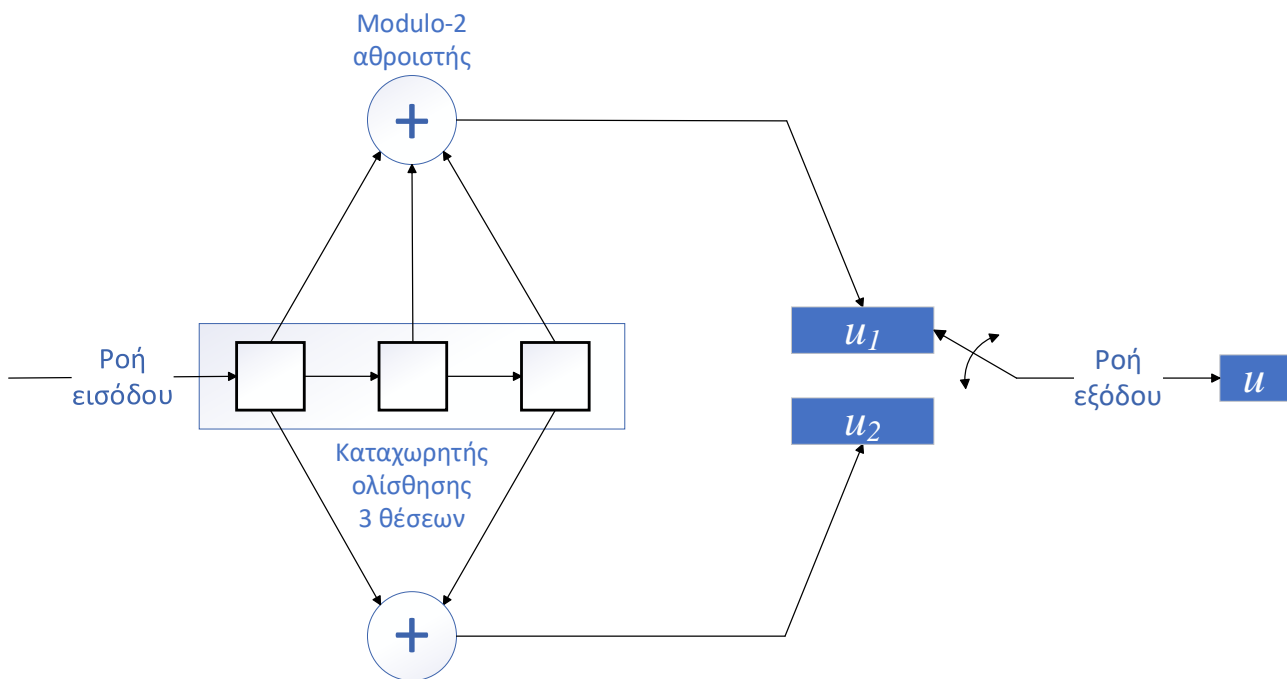
παραδοσιακός αποκωδικοποιητής Viterbi έχει εδώ και καιρό αποτελέσει το θεμέλιο της διόρθωσης σφαλμάτων, προσφέροντας μια σχολαστική προσέγγιση στην αποκωδικοποίηση σε περιβάλλον θορύβου και παρεμβολών.

Ωστόσο, η εμφάνιση των νευρωνικών δικτύων έχει προκαλέσει ένα ενδιαφέρον ερώτημα σχετικά με τη δυνατότητά τους να προσεγγίσουν τις πολύπλοκες λειτουργίες που εκτελούν οι κλασικοί αλγόριθμοι. Η παρούσα εργασία επιδιώκει να εξετάσει αυτό το σημείο τομής μεταξύ της παραδοσιακής επεξεργασίας σήματος και σύγχρονης τεχνητής νοημοσύνης.

Στο επίκεντρο του ενδιαφέροντός μας βρίσκεται η εξερεύνηση της εφικτότητας χρήσης νευρωνικών δικτύων ως εναλλακτικών για τον αποκωδικοποιητή Viterbi. Στόχος μας είναι να κατανοήσουμε εάν τα νευρωνικά δίκτυα μπορούν να ανταγωνιστούν, ή ακόμη και να υπερκεράσουν, την απόδοση του αξιόπιστου αλγορίθμου Viterbi στην αποκωδικοποίηση μηνυμάτων που μεταδίδονται μέσω συνελικτικής κωδικοποίησης.

2 Συνελκτικοί κώδικες και αποκωδικοποίηση Viterbi

2.1 Περιγραφή λειτουργίας συνελκτικού κώδικα



Εικόνα 2.1 (2,1,3) συνελκτικός κωδικοποιητής

Στην Εικόνα 2.1 παρουσιάζεται ένας συνελκτικός κωδικοποιητής. Για να γίνει καλύτερα κατανοητό το πρότυπο λειτουργίας μια τέτοιας συσκευής, ακολουθεί μια αναλυτική περιγραφή των διεργασιών που λαμβάνουν χώρα όταν ο κωδικοποιητής καλείται να επεξεργαστεί την ακολουθία πληροφορίας (1 0 1).

Στην αρχή ο καταχωρητής είναι «καθαρός», που σημαίνει ότι το περιεχόμενό του είναι (0 0 0). Κατά το 1^ο χρονικό βήμα, στον καταχωρητή εισέρχεται το πρώτο ψηφίο της ακολουθίας, το «1». Πλέον, το περιεχόμενό του καταχωρητή είναι (1 0 0). Αυτό σημαίνει πως η τιμή στην ροή u_1 είναι $1 \oplus 0 \oplus 0 = 1^1$. Αντίστοιχα στην u_2 είναι $1 \oplus 0 = 1$. Οι u_1 και u_2 στη συνέχεια διασταυρώνονται και προκύπτει η έξοδος (1 1) και λαμβάνουν τέλος οι διεργασίες του πρώτου βήματος.

Κατά το 2^ο χρονικό βήμα, ο κωδικοποιητής επεξεργάζεται το 2^ο ψηφίο της ακολουθίας, το «0». Αρχικώς, γίνεται εισαγωγή του εντός του καταχωρητή με την διαδικασία της δεξιάς ολίσθησης. Κάθε ψηφίο «ολισθαίνει» κατά μια θέση δεξιότερα. Αυτό έχει ως αποτέλεσμα την είσοδο ενός νέου ψηφίου (καθώς και την απόρριψη του παλαιότερου χρονικά ψηφίου). Πλέον το περιεχόμενό του

¹ Ο τελεστής \oplus συμβολίζει την πράξη της modulo-2 πρόσθεσης. Η modulo-2 άθροιση είναι μια δυαδική πράξη που χρησιμοποιείται στην πληροφορική και τις ψηφιακές επικοινωνίες. Κατά αυτόν τον τύπο δυαδικής άθροισης, τα δυαδικά δεδομένα (συνήθως bits) αθροίζονται χωρίς να ληφθούν υπόψη τα κρατούμενα.

καταχωρητή γίνεται (0 1 0). Με τον τρόπο που προαναφέραμε υπολογίζεται η αντίστοιχη έξοδος και στη συνέχεια η ίδια διαδικασία επαναλαμβάνεται και για το 3^ο και τελευταίο ψηφίο της ακολουθίας.

Σε πολλούς κωδικοποιητές, μετά την επεξεργασία της ακολουθίας, ακολουθεί μια διαδικασία καθαρισμού της μνήμης του καταχωρητή, δηλαδή επαναφοράς της στην αρχική κατάσταση. Στην περίπτωση του καταχωρητή της Εικόνας 2.1 αυτό γίνεται με την εισαγωγή 2 επιπλέον μηδενικών ψηφίων, καθώς ο καταχωρητής έχει 2 θέσεις μνήμης.

Όλη η διαδικασία παρουσιάζεται στον Πίνακα 2.1.

Χρονικό βήμα	Καταχωρητής			u_1	u_2
1	1	0	0	1	1
2	0	1	0	1	0
3	1	0	1	0	0
Καθαρισμός μνήμης (προαιρετικό)					
4	0	1	0	1	0
5	0	0	1	1	1

Πίνακας 2.1 Κωδικοποίηση μηνύματος (1 0 1)

2.2 Αλγεβρικές αναπαραστάσεις των συνελικτικών κωδίκων

Στην ενότητα 1.4 δόθηκε μια γενική περιγραφή των συνελικτικών κωδίκων. Στην πλέον αυστηρή προσέγγιση, οι συνελικτικοί κώδικες μπορούν να θεωρηθούν ως μια ειδική κατηγορία των γραμμικών μπλοκ κωδίκων (με άπειρο μήκος τμήματος). Όμως η συνελικτική τους δομή προσφέρει καλύτερες επιδόσεις και διευκολύνει την αποκωδικοποίηση.

Ένας αλγεβρικός τρόπος να αναπαρασταθεί ένας συνελικτικός κώδικας είναι μέσω ενός πίνακα δημιουργίας όπως και οι μπλοκ κώδικες. Ο πίνακας δημιουργίας G ενός συνελικτικού κώδικα είναι αυτός για τον οποίο ισχύει:

$$\mathbf{u} = \mathbf{mG} \quad (2.1)$$

όπου \mathbf{m} ο πίνακας γραμμή που αναπαριστά την ακολουθία πληροφορίας και \mathbf{u} ο πίνακας γραμμή που αναπαριστά την κωδικοποιημένη ακολουθία. Γενικά, ο πίνακας δημιουργίας για έναν συνελικτικό κώδικα είναι ημίπειρος από τη στιγμή που η ακολουθία πληροφορίας είναι ημίπειρη σε μήκος. Σαν μια πρακτική εναλλακτική στον πίνακα δημιουργίας, μπορεί να χρησιμοποιηθεί μια

ισοδύναμη αναπαράσταση με τη χρήση πολυωνύμων δημιουργίας. Για έναν (n,k,K) συνελκτικό κωδικοποιητή υπάρχουν n πολυώνυμα δημιουργίας, κάθε ένα εκ των οποίων αντιστοιχεί σε κάθε ροή εξόδου του κωδικοποιητή, η οποία παράγεται από έναν modulo-2 αθροιστή. Κάθε πολυώνυμο δημιουργίας είναι επί της ουσίας ένα διάνυσμα με kK διαστάσεις, του οποίου η τιμή της i -οστής θέσης εκφράζει την ύπαρξη ή μη σύνδεσης μεταξύ του αθροιστή και του αντίστοιχου τμήματος του καταχωρητή. Το ψηφίο «1» εκφράζει την ύπαρξη σύνδεσης και το ψηφίο «0» την απουσία αυτής.

Ο $(2,1,3)$ κωδικοποιητής της εικόνα Εικόνα 2.1 έχει έναν καταχωρητή τριών θέσεων, μια ροή εισόδου και δύο ροές εξόδου (οι οποίες διασταυρώνονται και προκύπτει το κωδικοποιημένο σήμα), οι οποίες παράγονται από δύο modulo-2 αθροιστές. Το πολυώνυμο δημιουργίας της ροής u_1 είναι:

$$g_1 = (1 \ 1 \ 1) \quad (2.2)$$

και το πολυώνυμο της ροής u_2 :

$$g_2 = (1 \ 0 \ 1) \quad (2.3)$$

Μπορούν να γραφούν επίσης σε πολυωνυμική μορφή:

$$g_1(x) = 1 + x + x^2 \quad (2.4)$$

$$g_2(x) = 1 + x^2 \quad (2.5)$$

Για τις επιμέρους ροές u_1 και u_2 ισχύει πως:

$$u_1 = m \cdot g_1 \quad (2.6)$$

$$u_2 = m \cdot g_2 \quad (2.7)$$

Ενώ για τη συνολική ροή, η οποία προκύπτει μετά από την διασταύρωση των δύο επιμέρους, ισχύει πως:

$$u = m \cdot G \quad (2.8)$$

όπου ο πίνακας για δημιουργίας G είναι:

$$G = \begin{bmatrix} (g_1)_1 & (g_2)_1 & (g_1)_2 & (g_2)_2 & (g_1)_3 & (g_2)_3 \\ & & (g_1)_1 & (g_2)_1 & (g_1)_2 & (g_2)_2 & (g_1)_3 & (g_2)_3 \\ & & & \ddots & & & & \ddots \end{bmatrix} \\ = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 \\ & & 1 & 1 & 1 & 0 & 1 & 1 \\ & & & & \ddots & & & \ddots \end{bmatrix} \quad (2.9)$$

Μια άλλη προσέγγιση πάνω στην συνελκτική κωδικοποίηση είναι η παλμική απόκριση. Η παλμική απόκριση είναι η απόκριση που θα έχει ο κωδικοποιητής καθώς ένα μοναδικό ψηφίο «1» διέρχεται από αυτόν. Η παλμική απόκριση του κωδικοποιητή της εικόνας 2.1 φαίνεται στον Πίνακας 2.2:

Χρονικό βήμα	Περιεχόμενο καταχωρητή	u_1	u_2
1	1 0 0	1	1
2	0 1 0	1	0
3	0 0 1	1	1

Πίνακας 2.2 Παλμική απόκριση

Η ακολουθία εξόδου (11 10 11) αποτελεί την παλμική απόκριση του κωδικοποιητή. Αποδεικνύεται πως για η έξοδος του κωδικοποιητή για κάθε πιθανή ακολουθία πληροφορίας που καλείται να κωδικοποιήσει, μπορεί να βρεθεί από την υπέρθεση των καταλλήλως χρονικώς μετατοπισμένων αποκρίσεων για όλα τα ψηφία της ακολουθίας. Για να γίνει καλύτερα αντιληπτό: παρατίθεται ένα παράδειγμα: έστω ότι η ακολουθία πληροφορίας είναι (0 1 1 0 1). Η διαδικασία εύρεσης της κωδικοποιημένης ακολουθίας παρουσιάζεται στον Πίνακα 2.3:

ΧΡΟΝΙΚΟ ΕΙΣΑΓΟΜΕΝΟ

ΒΗΜΑ	ΨΗΦΙΟ							
1	0	00	00	00				
2	1		11	10	11			
3	1			11	10	11		
4	0				00	00	00	
5	1					11	10	11
MODULO-2 ΑΘΡΟΙΣΜΑ		00	11	01	01	00	10	11

Πίνακας 2.3 Εύρεση κωδικοποιημένης ακολουθίας μέσω παλμικής απόκρισης

Το modulo-2 άθροισμα είναι η κωδικοποιημένη ακολουθία².

2.3 Οι συνελκτικοί κώδικες ως μηχανή πεπερασμένων καταστάσεων

Η μηχανή πεπερασμένων καταστάσεων (Finite State Machine, FSM) είναι ένα μαθηματικό μοντέλο που χρησιμοποιείται για την περιγραφή της συμπεριφοράς συστημάτων. Η λειτουργία πολλών

² Να σημειωθεί πως αν και φαίνεται πως ο ρυθμός κώδικα είναι διαφορετικός του 1/2, κάτι τέτοιο δεν ισχύει. Το αποτέλεσμα της συγκεκριμένης μεθόδου είναι σωστό αν υποθεθεί πως ο κωδικοποιητής, έπειτα από την είσοδο και την επεξεργασία της ακολουθίας πληροφορίας, προχωράει σε καθαρισμό της μνήμης του καταχωρητή ολίσθησης εισάγοντας «αυτόματα» 2 επιπλέον ψηφία «0». Ο καθαρισμός της μνήμης είναι μια συνήθης πρακτική στους συνελκτικούς κωδικοποιητές.

σύγχρονων συστημάτων μπορεί να αναπαρασταθεί με τέτοια μοντέλα. Οι κρίσιμες παράμετροι αυτού του μοντέλου είναι:

- Καταστάσεις: Κάθε FSM χαρακτηρίζεται από ένα πεπερασμένων σύνολο πιθανών συνθηκών και καταστάσεων στις οποίες μπορεί να βρίσκεται. Αυτές οι καταστάσεις μπορούν να αναπαρασταθούν σαν κόμβοι σε ένα γράφημα
- Μεταβάσεις: Είναι οι σύνδεσμοι μεταξύ των διαφόρων καταστάσεων. Περιγράφουν τα γεγονότα που οδηγούν το σύστημα να μεταβεί από την μια κατάσταση στην άλλη.
- Αρχική κατάσταση: Εκφράζει την κατάσταση στην οποία βρίσκεται το σύστημα, όταν εκκινεί την λειτουργία του.
- Αλφάβητο εισόδου: Είναι το σύνολο των συμβόλων που μπορεί να δεχθεί η μηχανή πεπερασμένων καταστάσεων ως είσοδο. Κάθε μετάβαση σχετίζεται με ένα ή περισσότερα σύμβολα από το αλφάβητο εισόδου.
- Έξοδοι: Σε κάποιες περιπτώσεις μια FSM έχει εξόδους σχετιζόμενες με τις μεταβάσεις. Αυτές οι έξοδοι μπορεί να εκφράζουν γεγονότα που συμβαίνουν όταν μια μετάβαση συμβαίνει.
- Τελικές καταστάσεις: Κάποιες καταστάσεις σε αυτά τα συστήματα μπορεί να έχουν σχεδιαστεί ως τελικές

Ακόμη, οι μηχανές πεπερασμένων καταστάσεων κατηγοριοποιούνται σε:

- Ντετερμινιστικές και μη ντετερμινιστικές: Σε μια ντετερμινιστική FSM, κάθε συνδυασμός εισόδου-κατάστασης οδηγεί σε μια και μόνο προκαθορισμένη μετάβαση. Σε μια μη ντετερμινιστική, ένας συνδυασμός εισόδου-κατάστασης μπορεί να οδηγεί σε πολλαπλές μεταβάσεις. Το ποια συγκεκριμένη μετάβαση θα συμβεί μπορεί να καθορίζεται από εξωτερικούς παράγοντες.
- Μηχανές Mealy και μηχανές Moore: Για μια μηχανή Mealy ισχύει πως η έξοδος που θα παραχθεί, καθορίζεται τόσο από την τρέχουσα κατάσταση της μηχανής, όσο και από την τρέχουσα είσοδο. Εν αντιθέσει, οι έξοδοι των μηχανών Moore, εξαρτώνται αποκλειστικά και μόνο από την τρέχουσα κατάσταση.

Η δομή των συνελκτικών κωδικοποιητών είναι τέτοια που επιτρέπει να περιγραφούν από ένα μοντέλο μηχανής πεπερασμένων καταστάσεων. Ένας (n,k,K) συνελκτικός κωδικοποιητής έχει συνολικά K θέσεις στον καταχωρητή άρα $K-1$ θέσεις μνήμης των k bits. Σε αυτές τις $K-1$ θέσεις μπορεί να βρίσκονται $2^{k(K-1)}$ πιθανοί συνδυασμοί ψηφίων. Αυτοί οι πιθανοί συνδυασμοί στις θέσεις μνήμης αποτελούν και τις καταστάσεις του μοντέλου. Είναι οι πιθανές καταστάσεις στις οποίες μπορεί να βρεθεί ο κωδικοποιητής πριν εισαχθούν σε αυτόν k νέα bits. Η είσοδος αυτής της

δυναμική ακολουθία των k νέων ψηφίων αποτελεί και το γεγονός που προκαλεί τη μετάβαση σε μια νέα κατάσταση και την παραγωγή ακολουθίας εξόδου μήκους n . Οπότε το αλφάβητο εισόδου είναι οι 2^k πιθανές ακολουθίες. Όσον αφορά την αρχική κατάσταση του κωδικοποιητή, είναι αυτή που περιέχει ψηφία «0» σε όλες τις θέσεις μνήμης. Όπως εύκολα γίνεται αντιληπτό από αυτά που αναφέρθηκαν, πρόκειται για μια ντετερμινιστική μηχανή Mealy.

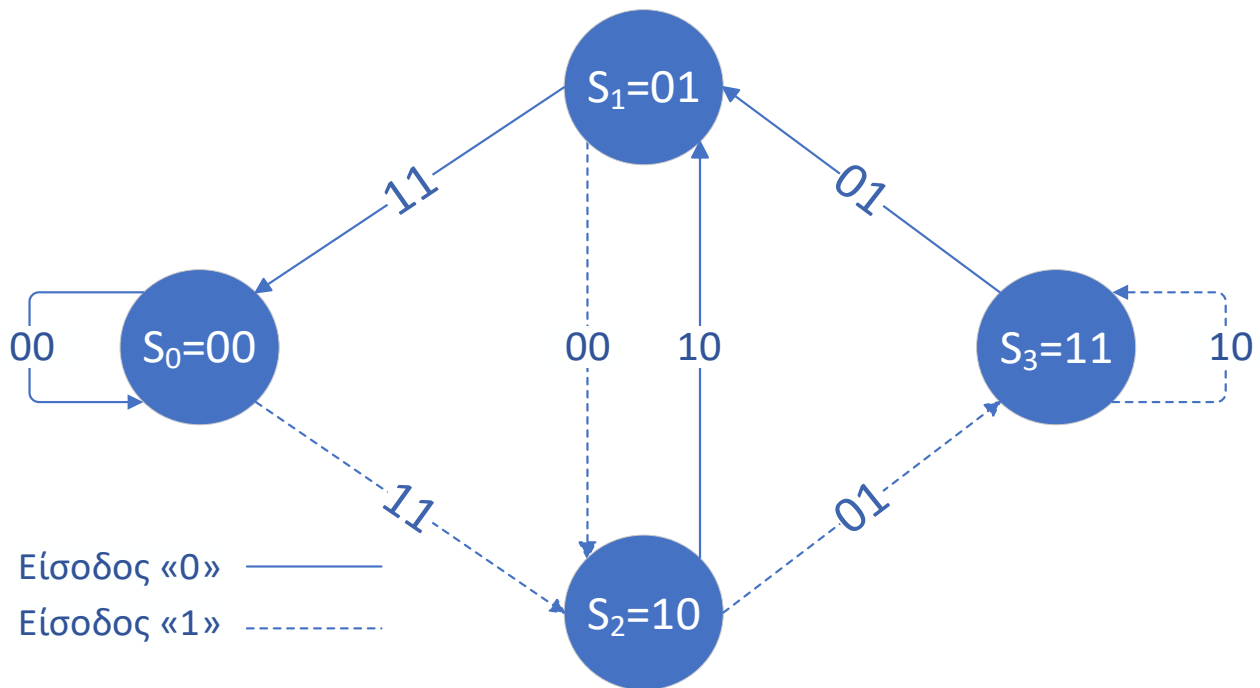
Ο κωδικοποιητής της Εικόνα 2.1, ως συνελκτικός, μπορεί να αναπαρασταθεί με μια μηχανή πεπερασμένων καταστάσεων:

- Ο κωδικοποιητής καθώς έχει 2 θέσεις μνήμης του ενός bit έχει στο σύνολο $2^2=4$ καταστάσεις. Αυτές είναι οι (0 0), (0 1), (1 0) και (1 1).
- Καθώς δέχεται μόνο ένα ψηφίο την φορά ως είσοδο, οι πιθανές εισοδοι του είναι τα ψηφία «0» και «1» για καθένα εκ των οποίων προκύπτει μια μετάβαση από την τρέχουσα του κατάσταση. Αν για παράδειγμα βρίσκεται στην κατάσταση (1 0) και δεχθεί ως είσοδο ένα bit «0», θα μεταβεί στην κατάσταση (0 1), ενώ αν δεχθεί το bit «1» θα μεταβεί στην κατάσταση (1 1).
- Κάθε φορά που υπάρχει μια νέα είσοδος, προκύπτει μια ακολουθία δύο ψηφίων στην έξοδο. Αν για παράδειγμα ο κωδικοποιητής βρίσκεται στην κατάσταση (0 1) και εισέλθει στον καταχωρητή ένα bit «1», τότε από την ροή u_1 προκύπτει το ψηφίο εξόδου $1 \oplus 0 \oplus 1 = 0$ και από την u_2 το ψηφίο $1 \oplus 1 = 0$ άρα η συνολική ακολουθία εξόδου θα είναι η (0 0).
- Αξίζει να σημειωθεί πως στην περίπτωση που ο κωδικοποιητής βρίσκεται στην κατάσταση (0 0) και δεχθεί το ψηφίο «0», παράγει μεν έξοδο αλλά «μεταβαίνει» στην ίδια κατάσταση, την (0 0). Μια τέτοια μετάβαση χαρακτηρίζεται ως αυτό-μετάβαση. Το ίδιο συμβαίνει όταν το σύστημα ευρισκόμενο στην κατάσταση (1 1) δεχθεί ως είσοδο «1».
- Η αρχική κατάσταση του κωδικοποιητή είναι η (0 0).

2.4 Γραφικές αναπαραστάσεις των συνελκτικών κωδίκων

Οι ιδιότητες των μηχανών πεπερασμένων καταστάσεων (καταστάσεις, μεταβάσεις, εισοδοι και έξοδοι) μπορούν να αποτυπωθούν σχηματικά με ένα διάγραμμα καταστάσεων. Με βάση τις παρατηρήσεις που έγιναν στην ενότητα 2.3, μπορεί να σχεδιαστεί το διάγραμμα καταστάσεων κωδικοποιητή της Εικόνας 2.1. Οι κόμβοι συμβολίζουν τις καταστάσεις, τα βέλη τις μεταβάσεις, ενώ πάνω σε κάθε βέλος αναγράφεται η είσοδος που προκάλεσε τη μετάβαση και η έξοδος που προκύπτει.

Παρότι το διάγραμμα καταστάσεων αποτυπώνει με κομψότητα και σαφήνεια την λειτουργία του κωδικοποιητή, εμφανίζει ένα σημαντικό μειονέκτημα. Αυτού του τύπου τα διαγράμματα δεν είναι ικανά να αποτυπώσουν την λειτουργία του κωδικοποιητή συναρτήσει του χρόνου. Με άλλα λόγια δεν μπορούν να αποτυπώσουν το «μονοπάτι» που ακολουθεί ο κωδικοποιητής ώστε να επεξεργαστεί μια συγκεκριμένη αλληλουχία bit και να παράγει μια κωδικοποιημένη ακολουθία.



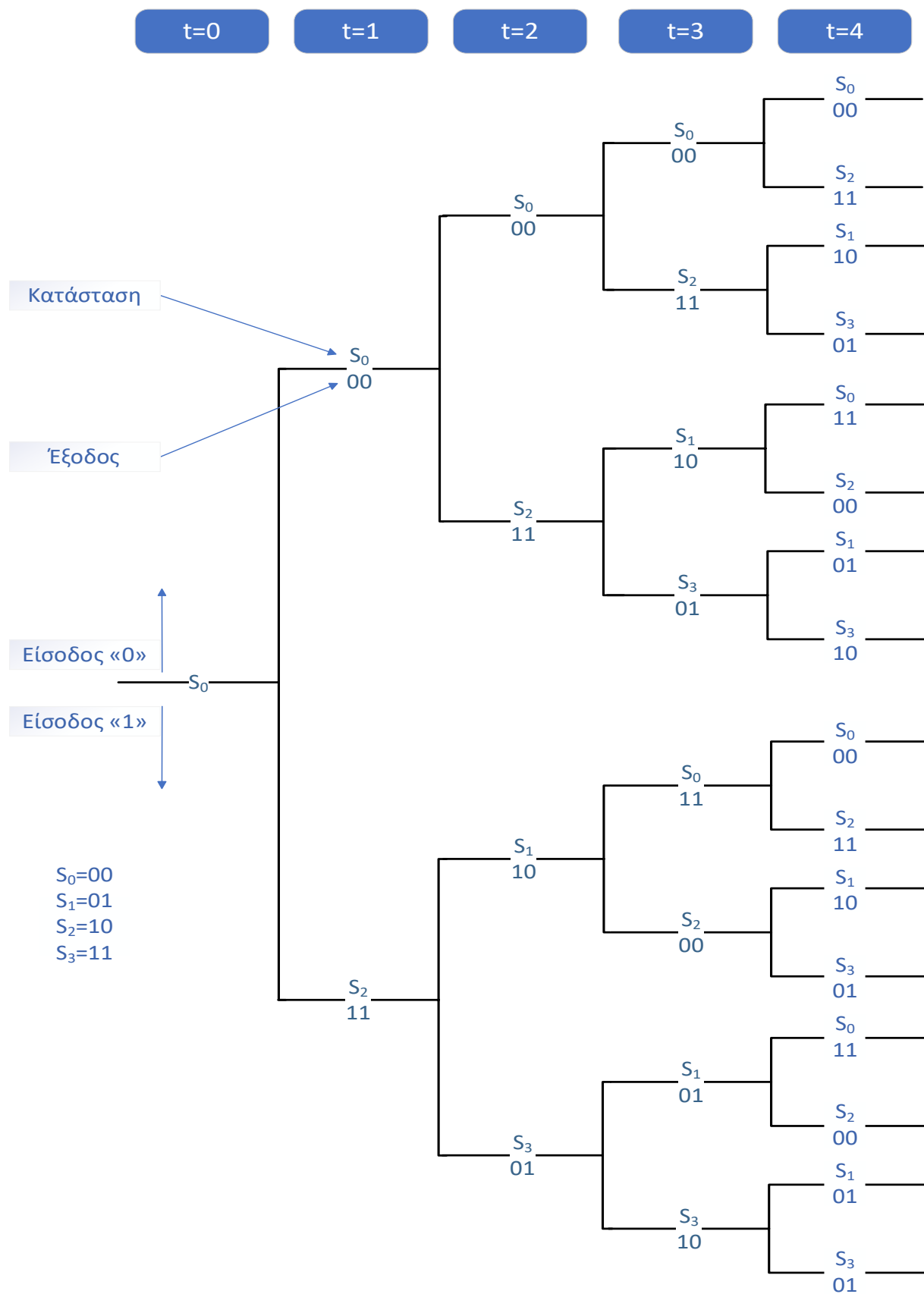
Εικόνα 2.2 Διάγραμμα καταστάσεων του (2,1,3) συνελκτικού κωδικοποιητή

Για τη μελέτη της χρονικής αλληλουχίας των γεγονότων που λαμβάνουν χώρα κατά την κωδικοποίηση μια ακολουθίας πληροφορίας, καταλληλότερα είναι τα δενδρώδη διαγράμματα και τα διαγράμματα Trellis.

Το δενδρώδες διάγραμμα προσθέτει την διάσταση του χρόνου. Εκκινώντας από την αρχική κατάσταση (0 0) παρουσιάζει όλα τα πιθανά «μονοπάτια» που μπορεί να ακολουθήσει ο κωδικοποιητής ανάλογα με την είσοδο που θα λάβει. Για μια συγκεκριμένη ακολουθία εισόδου, μπορούν να προσδιοριστούν η έξοδος και οι καταστάσεις από τις οποίες διέρχεται κατά την επεξεργασία του σήματος ο κωδικοποιητής, «διασχίζοντας» το δενδρώδες διάγραμμα. Στην Εικόνα 2.3 καταγράφεται το δενδρώδες διάγραμμα (για τα 4 πρώτα bit εισόδου) του κωδικοποιητή της Εικόνας 2.1..

Για κάθε νέο χρονικό βήμα, το οποίο εκκινεί με την είσοδο ενός νέου ψηφίου, η διαδικασία κωδικοποίησης μπορεί να περιγραφεί διασχίζοντας το διάγραμμα από τα αριστερά προς τα δεξιά, με κάθε διακλάδωση να περιέχει το αντίστοιχο μερικό τμήμα εξόδου. Ο κανόνας για να βρεθεί μια ακολουθία εξόδου είναι η εξής: αν το bit εισόδου είναι «0», στην επόμενη διακλάδωση κινούμαστε

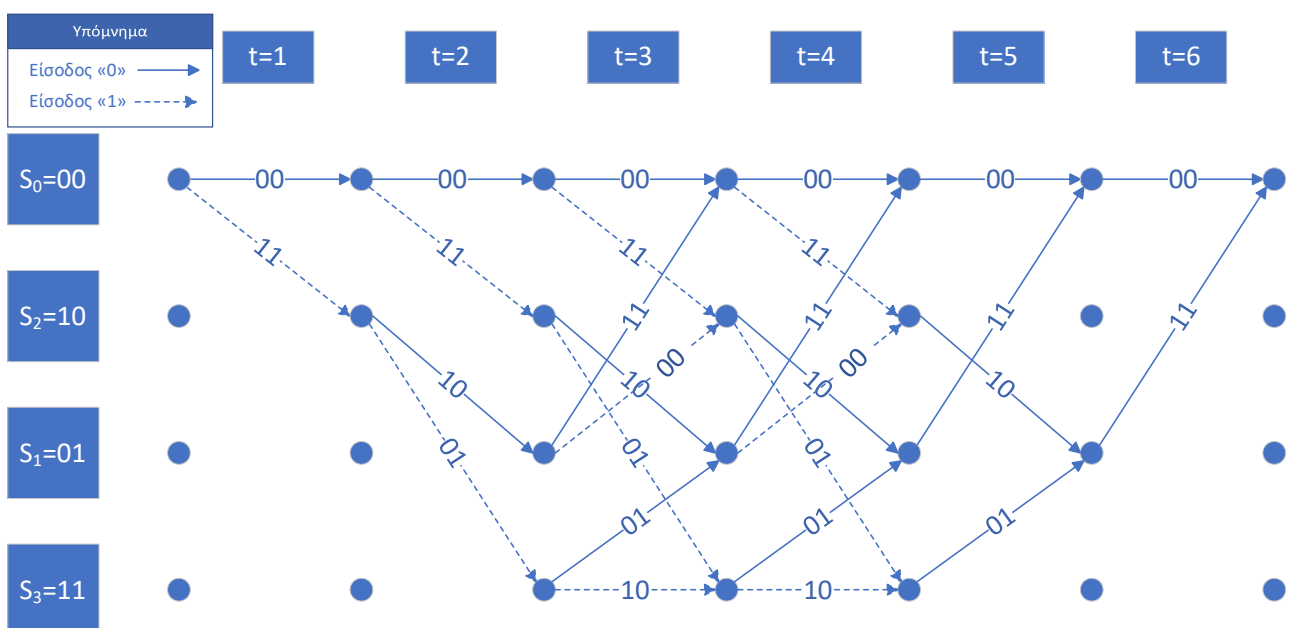
ανοδικά, ενώ αν είναι «1», η κίνηση γίνεται καθοδικά. Για παράδειγμα, για την ακολουθία (1011) προκύπτει πως η έξοδος είναι (11 10 00 01).



Εικόνα 2.3 Διάγραμμα δένδρου για (2,1,3) συνελκτικό κωδικοποιητή

Όπως μπορεί κανείς να παρατηρήσει, η δομή του δενδρώδους διαγράμματος αρχίζει να επαναλαμβάνεται μετά την 3^ο διακλάδωση. Στην γενικότερη περίπτωση ενός (n,k,K) συνελικτικού κωδικοποιητή, αυτή η επανάληψη στη δομή του δενδρώδους διαγράμματος συμβαίνει μετά την K-οστή διακλάδωση. Μετά την 1^η διακλάδωση, παράγονται δύο κόμβοι, ένας σε κατάσταση S_0 και ένας σε κατάσταση S_2 . Μετά την 2^η διακλάδωση, υπάρχουν 4 κόμβοι σε καταστάσεις S_0, S_1, S_2 και S_3 . Μετά την 3^η διακλάδωση, υπάρχουν 2 κόμβοι σε κατάσταση S_0 , 2 κόμβοι σε κατάσταση S_1 , 2 κόμβοι σε κατάσταση S_2 , 2 κόμβοι σε κατάσταση S_3 . Από αυτό το σημείο και έπειτα, το πάνω και το κάτω μισό του δένδρου είναι πανομοιότυπα. Ο λόγος είναι ο εξής: καθώς εισέρχεται το 4^ο bit στον κωδικοποιητή, το 1^ο bit εισόδου αποβάλλεται και δεν επηρεάζει πλέον την έξοδο του κωδικοποιητή. Ως εκ τούτου, μπορεί να ειπωθεί ότι η έξοδος 2 bit για κάθε bit εισόδου εξαρτάται μόνο από το bit εισόδου και από την τρέχουσα κατάσταση (2 προηγούμενα bit εισόδου).

Οποιοιδήποτε δύο κόμβοι που βρίσκονται στην ίδια κατάσταση την ίδια χρονική στιγμή μπορούν να συγχωνευθούν σε έναν από τη στιγμή που όλα τα επόμενα μονοπάτια που εκπορεύονται από αυτούς είναι ίδια. Αυτή την ιδιότητα εκμεταλλεύονται τα διαγράμματα Trellis. Το διάγραμμα Trellis αποτελεί μια πιο συνεπτυγμένη και κομψή αναπαράσταση σε σχέση με το δενδρώδες διάγραμμα. Αποτελεί επί της ουσίας μια επέκταση του διαγράμματος καταστάσεων στο πεδίο του χρόνου. Στην Εικόνα 2.4 φαίνεται ένα διάγραμμα Trellis του κωδικοποιητή της Εικόνας 2.1 που περιγράφει τη λειτουργία του για μια ακολουθία εισόδου 4 bits (περιλαμβάνεται και η εισαγωγή των 2 ψευδών bits «0» μετά την ακολουθία για τον καθαρισμό της μνήμης του καταχωρητή.).



Εικόνα 2.4 Διάγραμμα Trellis (2,1,3) συνελικτικού κώδικα

2.5 Ο αλγόριθμος Viterbi

Καθώς η αποσταλείσα ακολουθία διέρχεται από το κανάλι, υφίσταται παραμόρφωση από το θόρυβο. Έστω ότι για την μετάδοση πληροφορίας μέσω ενός δυαδικού συμμετρικού καναλιού (Binary Symmetric Channel, BSC) χρησιμοποιείται ο συνελκτικός κωδικοποιητής της Εικόνας 2.1.. Σε ένα BSC ο πομπός στέλνει ένα bit και ο δέκτης λαμβάνει ένα bit. Η πιθανότητα να ληφθεί λάθος bit είναι ίση με p ενώ η πιθανότητα να ληφθεί ορθό είναι $1 - p$.

Όταν το σύστημα κωδικοποίησης είναι άνευ μνήμης, ένας αποκωδικοποιητής συμβόλου προς σύμβολο είναι ο βέλτιστος καθώς ελαχιστοποιεί την πιθανότητα λάθους στο ένα σύμβολο. Όταν όμως ο κωδικοποιητής έχει μνήμη, τα σύμβολα της αποσταλείσας ακολουθίας είναι αλληλεξαρτώμενα και ως εκ τούτου η βέλτιστη μέθοδος αποκωδικοποίησης είναι αυτή που λαμβάνει αποφάσεις με κριτήριο την παρατήρηση ολόκληρης της ακολουθίας των λαμβανόμενων συμβόλων και όχι ενός μεμονωμένου συμβόλου. Όπως γίνεται εύκολα αντιληπτό με βάση αυτά που αναφέρθηκαν στην ενότητα 1.5, ένας ML αποκωδικοποιητής επιτελεί τέτοιου είδους λειτουργία.

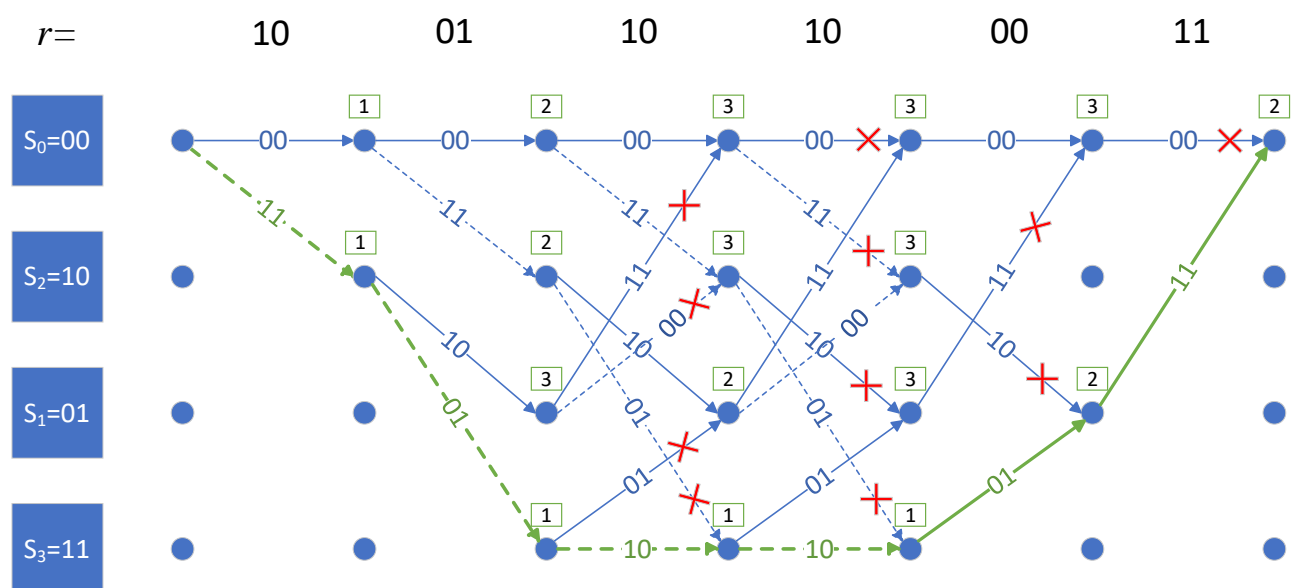
Ένας αποκωδικοποιητής μεγίστης πιθανοφάνειας στην περίπτωση ενός καναλιού άνευ μνήμης λειτουργεί έτσι ώστε να μεγιστοποιήσει την ποσότητα (1.5). Στην ειδική περίπτωση που το κανάλι είναι BSC αποδεικνύεται πως η μεγιστοποίηση της ποσότητας (1.5) είναι ισοδύναμη με την ελαχιστοποίηση της απόστασης Hamming μεταξύ της ληφθείσας ακολουθίας r και της ακολουθίας \tilde{u} που προβλέπει ο αποκωδικοποιητής πως έχει σταλθεί. Η απόσταση Hamming είναι ένας τρόπος μέτρησης της διαφοράς μεταξύ δύο ακολουθιών. Η απόσταση Hamming δύο δυαδικών ακολουθιών είναι το πλήθος των θέσεων στις οποίες η ακολουθίες έχουν διαφορετικό bit.

Τα συστήματα κωδικοποίησης με μνήμη, όπως οι συνελκτικοί κωδικοποιητές, μπορούν, όπως απεδείχθη, να περιγραφούν ως μηχανές πεπερασμένων καταστάσεων οι οποίες αναπαρίστανται σχηματικά από ένα διάγραμμα Trellis. Η αποσταλείσα ακολουθία ψηφίων αντιστοιχεί σε ένα μονοπάτι εντός του διαγράμματος Trellis. Η δομή ενός διαγράμματος Trellis μπορεί να εκμεταλλευθεί έτσι ώστε να πραγματοποιηθεί αποκωδικοποίηση μεγίστης πιθανοφάνειας με έναν πολύ αποδοτικό τρόπο.

Έστω πως ο αποκωδικοποιητής λαμβάνει την ακολουθία $r = (10\ 01\ 10\ 10\ 00\ 11)$. Για να βρεθεί ποια από τις πιθανές κωδικολέξεις βρίσκεται κοντύτερα στην ληφθείσα ακολουθία, ο αποκωδικοποιητής πρέπει να διασχίσει το Trellis από τα αριστερά προς τα δεξιά, απορρίπτοντας όλες τις υποδιαδρομές που δεν θα ήταν δυνατό να είναι τμήμα της πιθανότερης διαδρομής. Στο 2ο χρονικό βήμα του Trellis, υπάρχουν τέσσερις δυνατές υποδιαδρομές, μία για κάθε κατάσταση του

κωδικοποιητή. Στο επόμενο βήμα, όμως υπάρχουν οκτώ υποδιαδρομές, δύο για κάθε κατάσταση του κωδικοποιητή. Για κάθε κατάσταση του κωδικοποιητή σε αυτό το στάδιο, επιβιώνει μόνο μια υποδιαδρομή που οδηγεί σε αυτήν ενώ οι υπόλοιπες απορρίπτονται. Η επιλογή γίνεται με βάση την εγγύτητα στο αντίστοιχο τμήμα της ληφθείσας ακολουθίας, δηλαδή επιλέγεται η υποδιαδρομή που έχει την μικρότερη απόσταση Hamming από την ως εκεί ληφθείσα ακολουθία. Η υποδιαδρομή με τη μεγαλύτερη απόσταση Hamming απορρίπτεται καθώς δεν είναι δυνατό να είναι τμήμα της βέλτιστης διαδρομής. Αυτή η διαδικασία επαναλαμβάνεται μέχρι και το 6^ο στάδιο, όταν και καταλήγουμε στην κατάσταση (00). Σε εκείνο το στάδιο έχει επιβιώσει μόνο μια διαδρομή η οποία αποτελεί και την πιθανότερη. Σε μερικές περιπτώσεις μπορεί δύο υποδιαδρομές να έχουν ακριβώς την ίδια απόσταση Hamming από το αντίστοιχο τμήμα της ληφθείσας ακολουθίας. Για παράδειγμα, στο 4^ο χρονικό στάδιο, οι δύο υποδιαδρομές που καταλήγουν στην κατάσταση $S_1=01$ έχουν απόσταση Hamming ίση με 3. Αυτό σημαίνει πως και οι δύο είναι εξίσου πιθανά ενδεχόμενα. Σε αυτήν την περίπτωση γίνεται μέσω τυχαίας επιλογής μιας εκ των δύο. Εν τέλει, προκύπτει πως η πιθανότερη διαδρομή είναι αυτή που αντιστοιχεί στην κωδικολέξη $u = (11\ 01\ 10\ 10\ 01\ 11)$ και την ακολουθία πληροφορίας $m = (1\ 1\ 1\ 1)$.

Ο αλγόριθμος αποκωδικοποίησης μεγίστης πιθανοφάνειας, ο οποίος περιγράφεται παραπάνω, ονομάζεται αλγόριθμος Viterbi και η απόσταση Hamming αποτελεί την μετρική Viterbi για αυτήν την περίπτωση. Όλη αυτή η διαδικασία καταγράφεται σχηματικά στην Εικόνα 2.5. Σε κάθε υποδιαδρομή που απορρίπτεται σημειώνεται ένα x, ενώ πάνω από κάθε ένα κόμβο σημειώνεται η απόσταση Hamming της επιζούσας υποδιαδρομής που καταλήγει σε αυτήν.



Εικόνα 2.5 Παράδειγμα αποκωδικοποίησης Viterbi

Τα βήματα του αλγορίθμου Viterbi για αποκωδικοποίηση ενός κωδικοποιημένου μηνύματος μήκους L που έχει προέλθει από έναν συνελκτικό κωδικοποιητή με K θέσεις μνήμης είναι στην γενικότερη περίπτωση:

Βήμα 1. Υποθέτουμε ότι ο συνελκτικός κωδικοποιητής είναι αρχικά στην μηδενική κατάσταση, ανατίθεται η τιμή μηδέν στην μετρική Viterbi του αρχικού κόμβου. Τίθεται $t=0$

Βήμα 2. Για κάθε κόμβο στο χρονικό στάδιο $t+1$, βρίσκουμε για κάθε μία από τις πιθανές πρόδρομες διαδρομές στο χρονικό στάδιο t το άθροισμα της μετρικής Viterbi και της μετρικής του συνδεδετικού κλάδου. Καθορισμός του μέγιστου (ή του ελάχιστου, εξαρτάται από την μετρική που χρησιμοποιείται) αυτών των αθροισμάτων και ανάθεση του σε αυτόν τον κόμβο. Επιλογή της αντίστοιχης διαδρομής ως επιζούσα.

Βήμα 3. Αν έχουμε φθάσει στο τέλος του Trellis ($t = L+K$), επιλέγεται ως πρόβλεψη του αποκωδικοποιητή η κωδικολέξη που αντιστοιχεί στην διαδρομή με τη μεγαλύτερη (ή μικρότερη) μετρική Viterbi και ολοκληρώνεται ο αλγόριθμος. Ειδάλλως γίνεται αύξηση του t κατά 1 και μετάβαση ξανά στο Βήμα 2.

2.6 Σκληρές και μαλακές αποφάσεις

Μέχρι τώρα έχει γίνει μελέτη για περιπτώσεις που ο αποδιαμορφωτής του δέκτη παίρνει μόνο «σκληρές» αποφάσεις (Hard decisions), δηλαδή λαμβάνεται βάση του ληφθέντος συμβόλου μια διακριτή απόφαση για το αν αυτό το σύμβολο απεικονίζει ένα bit «0» ή ένα bit «1» χωρίς να καταγράφονται επιπλέον πληροφορίες για το επίπεδο αξιοπιστίας αυτής της απόφασης. Εν αντιθέσει με τις σκληρές αποφάσεις, στην περίπτωση των μαλακών αποφάσεων (soft decisions) ο αποκωδικοποιητής καταγράφει και πληροφορίες για την αξιοπιστία των αποφάσεων που έχει λάβει

Έστω ότι ο διαμορφωτής χρησιμοποιεί μια δυαδική διαμόρφωση μετατόπισης φάσης (Binary Phase-Shift Keying, BPSK). Τα ψηφία «1» και «0» αναπαρίστανται από τις κυματομορφές:

$$s_1(t) = \begin{cases} \sqrt{\frac{2E_s}{T}} \cos \omega t, & 0 \leq t < T \\ 0, & \text{αλλού} \end{cases} \quad (2.10)$$

$$s_0(t) = -s_1(t) \quad (2.11)$$

όπου E_s η ενέργεια της κάθε κυματομορφής και T η χρονική της διάρκεια. Από την σχέση (2.11) γίνεται αντιληπτό πως αυτή διαμόρφωση είναι αντιποδική.

Ας υποθεθεί πως στη συνέχεια πως οι κυματομορφές μεταδίδονται μέσα από ένα κανάλι πρόσθετου λευκού γκαουσιανού θορύβου (Additive White Gaussian Noise, AWGN). Η συνάρτηση

πυκνότητας πιθανότητας του θορύβου που εισάγεται από το κανάλι είναι μια γκαουσιανή κατανομή με μέση τιμή 0 και διακύμανση $N_0/2$.

Έστω ότι στον δέκτη για την αποδιαμόρφωση χρησιμοποιείται ένα προσαρμοσμένο φίλτρο με παλμική απόκριση:

$$h(t) = \begin{cases} \sqrt{\frac{2}{T}} \cos \omega t, & 0 \leq t < T \\ 0, & \text{αλλού} \end{cases} \quad (2.12)$$

Η έξοδος του προσαρμοσμένου φίλτρου Z_i κατά το χρόνο δειγματοληψίας iT αποδεικνύεται πως είναι μία γκαουσιανή τυχαία μεταβλητή με μέση τιμή $\sqrt{E_s}$, αν η αποσταλείσα κυματομορφή ήταν η $s_1(t)$, και $-\sqrt{E_s}$, αν η αποσταλείσα κυματομορφή ήταν η $s_0(t)$. Η διακύμανση είναι και για τις δύο περιπτώσεις είναι ίση με $N_0/2$.

Στην περίπτωση της σκληρής απόφασης, στον αποδιαμορφωτή γίνεται μια κβάντιση δύο επιπέδων του Z_i με κατώφλι το μηδέν. Τότε το κανάλι μας περιγράφεται από το BSC, ένα απλό μοντέλο με δυαδική είσοδο και δυαδική έξοδο. Η πιθανότητα σφάλματος κατά τη μετάδοση σε αυτήν την περίπτωση μπορεί να υπολογιστεί εύκολα και είναι:

$$P_e = P(Z_i < 0 | s_1) = P(Z_i > 0 | s_0) = \text{erfc}(\sqrt{2E_s/N_0}) \quad (2.13)$$

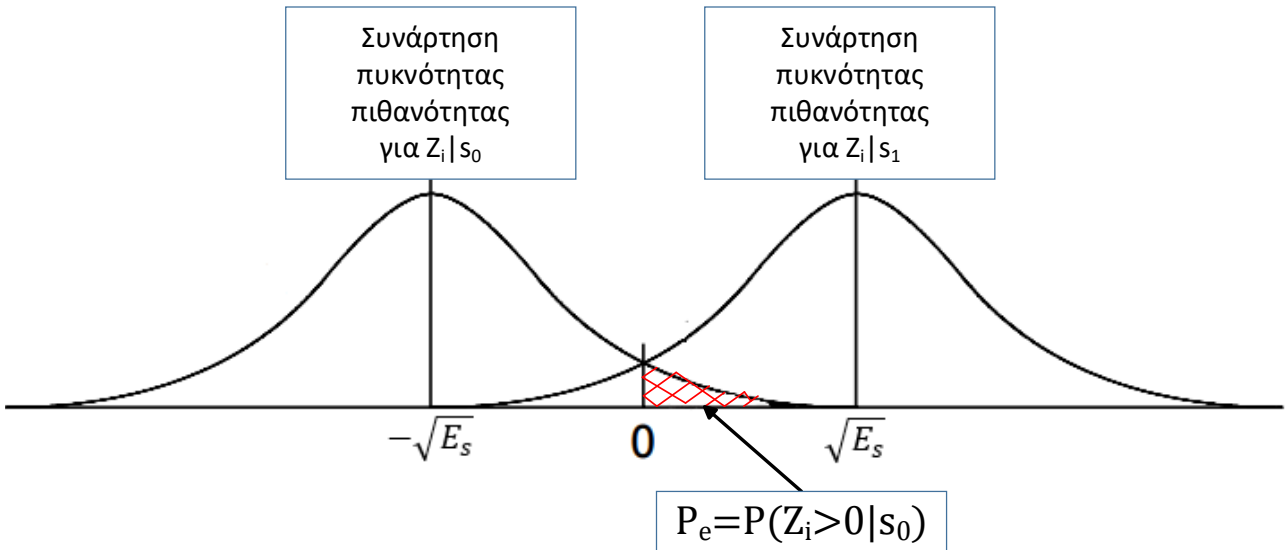
Είναι εμφανές πως η πιθανότητα σφάλματος κατά την μετάδοση είναι συνάρτηση του σηματοθορυβικού λόγου E_s/N_0 . Η συνάρτηση erfc ονομάζεται συμπληρωματική συνάρτηση σφάλματος και είναι ίση με:

$$\text{erfc}(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-y^2/2} dy \quad (2.14)$$

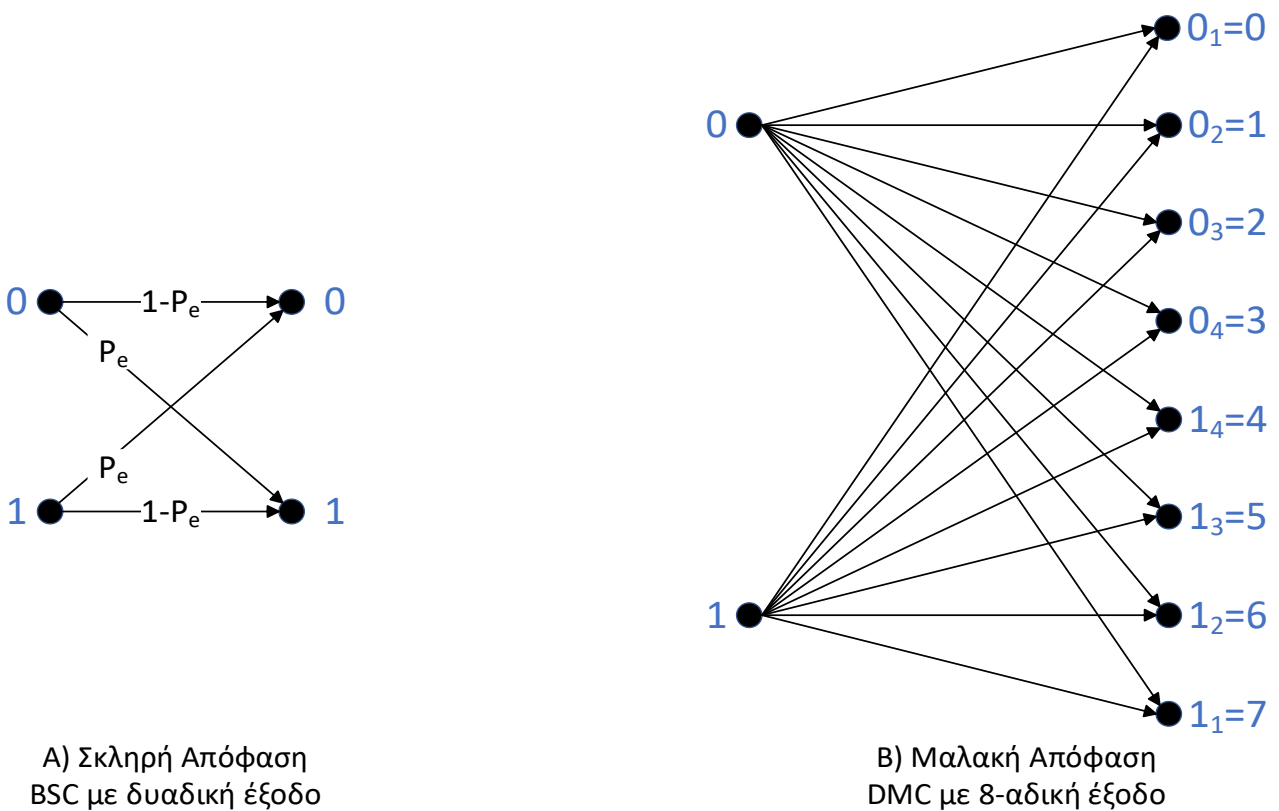
Η συνάρτηση πυκνότητα πιθανότητας του Z_i και η πιθανότητα σφάλματος κατά την μετάδοση αναπαρίστανται γραφικά στην Εικόνα 2.6.

Η χρήση σκληρών αποφάσεων είναι μια σπατάλη χρήσιμης πληροφορίας όταν γίνεται χρήση συνελκτικών κωδικοποιητών και αποκωδικοποιητών μέγιστης πιθανοφάνειας. Αφού η επιρροή κάθε ψηφίου εκτείνεται πέραν αυτού και παίζει ρόλο στον καθορισμό ολόκληρης της ακολουθίας που καλείται να προβλέψει ο αποκωδικοποιητής, θα ήταν βοηθητικό για αυτόν να λαμβάνει μια ένδειξη για την αξιοπιστία του λαμβανόμενου συμβόλου. Ο αποδιαμορφωτής μπορεί να χρησιμοποιήσει την ακριβή τιμή του Z_i ως έξοδο του, αλλά κάτι τέτοιο δεν είναι πρακτικό. Είναι πιο

συχνή μέθοδος να γίνει, για παράδειγμα, μια κβάντιση τριών bit, δηλαδή με 8 επίπεδα κβάντισης. Ο αποκωδικοποιητής λέμε πως λαμβάνει μια μαλακή απόφαση σε αυτήν την περίπτωση. Το κανάλι αυτό είναι ένα DMC με δυαδική είσοδο και 8-αδική έξοδο (Εικόνα 2.7).



Εικόνα 2.6 Συνάρτηση πυκνότητα πιθανότητας Z_i και πιθανότητα σφάλματος μετάδοσης



Εικόνα 2.7 Σκληρές και μαλακές αποφάσεις

Με τη χρήση του θεωρήματος κωδικοποίησης καναλιού του Shannon, προκύπτει πως για ένα AWGN κανάλι που γίνεται χρήση μαλακών αποφάσεων για να υπάρχει αξιόπιστη μετάδοση

χρειάζεται ο σηματοθορυβικός λόγος να είναι μεγαλύτερος του -1,6 dB (όριο Shannon), ενώ με τη χρήση σκληρών αποφάσεων αυτή το όριο μετατοπίζεται στα 0,4 dB. Άρα στην περίπτωση των μαλακών αποφάσεων εμφανίζεται ένα κέρδος της τάξης των 2 dB.

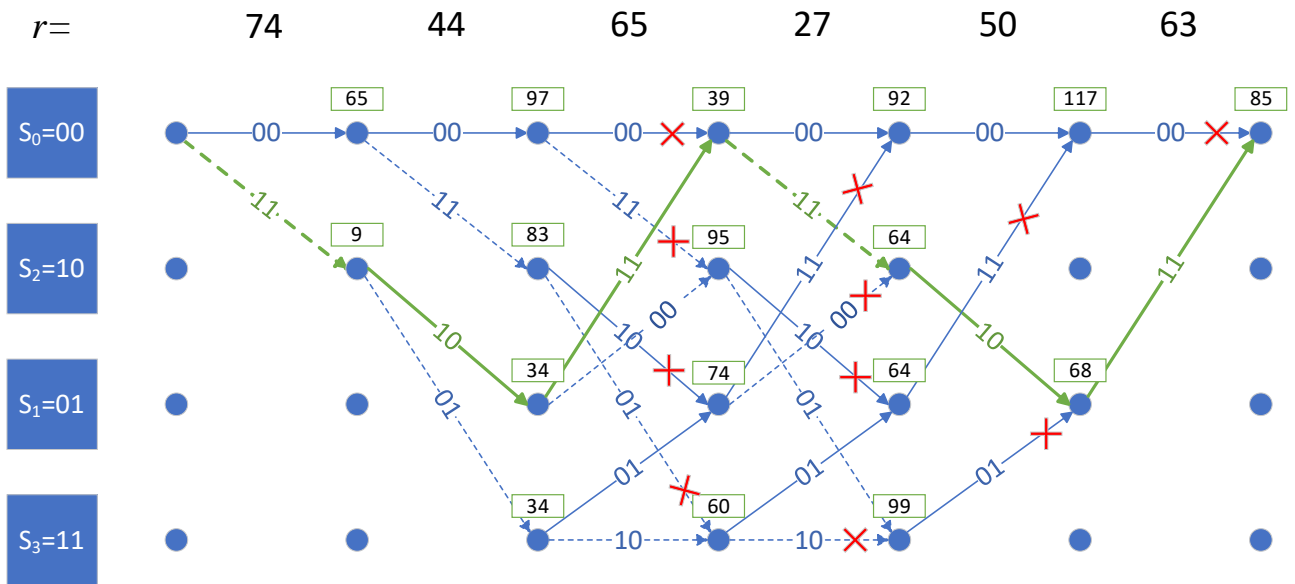
2.7 Αλγόριθμος Viterbi μαλακών αποφάσεων

Ας υποθεθεί πως ο πομπός αποστέλλει μια ακολουθία κωδικοποιημένη από τον συνελκτικό κωδικοποιητή της Εικόνα 2.1. Αυτή τη φορά όμως στον αποδιαμορφωτή του δέκτη η κβάντιση του ληφθέντος σήματος δεν γίνεται σε 2 επίπεδα (σκληρή απόφαση), αλλά σε 8 επίπεδα (μαλακή απόφαση). Πλέον οι πιθανές τιμές της ληφθείσας ακολουθίας κυμαίνονται από 0 μέχρι και 7. Στη συνέχεια αυτές οι τιμές οδηγούνται στον αποκωδικοποιητή.

Στην ενότητα 2.5 έγινε σαφής περιγραφή του αλγορίθμου Viterbi. Το παράδειγμα που δόθηκε αφορούσε όμως αφορούσε έναν Viterbi σκληρών αποφάσεων. Η κύρια διαφορά μεταξύ ενός αλγορίθμου Viterbi σκληρών αποφάσεων και ενός Viterbi μαλακών αποφάσεων είναι η μετρική η οποία θα χρησιμοποιηθεί. Λόγω της χρήσης μαλακών αποφάσεων, δεν είναι δόκιμη η χρήση της απόστασης Hamming ως μετρικής Viterbi. Μια πιο κατάλληλη μετρική Viterbi σε αυτήν την περίπτωση θα ήταν η Ευκλείδεια απόσταση. Η Ευκλείδεια απόσταση υπολογίζεται ως εξής:

- Έστω πως τα ληφθέντα ψηφία σε ένα χρονικό στάδιο είναι (3,6) και πρέπει να υπολογιστεί η μετρική ενός κλάδου με έξοδο (1,0)
- Τα δυαδικά ψηφία εξόδου μετατρέπονται στα αντίστοιχα οκταδικά: (1,0) → (7,0)
- Γίνεται υπολογισμός της απόστασης μεταξύ των διανυσμάτων (7,0) και (3,6):
$$\sqrt{(7-3)^2 + (0-6)^2} = \sqrt{52}$$
- Για λόγους ευκολίας, καθώς η τετραγωνική ρίζα είναι μια γνησίως αύξουσα συνάρτηση για το πεδίο ορισμού της, μπορεί να χρησιμοποιηθεί ως μετρική όχι η Ευκλείδεια απόσταση αλλά το τετράγωνο της απόστασης, δηλαδή στην περίπτωση αυτή το 52

Η γενική δομή του αλγορίθμου παραμένει ίδια. Η μόνη διαφορά από τον Viterbi σκληρών αποφάσεων είναι το κριτήριο που χρησιμοποιείται για τον προσδιορισμό της βέλτιστης διαδρομή, αντί να επιλέγεται η διαδρομή που ελαχιστοποιεί την απόσταση Hamming, επιλέγεται αυτή που ελαχιστοποιεί το άθροισμα των Ευκλειδίων αποστάσεων (ή των τετραγώνων τους). Ένα παράδειγμα αποκωδικοποίησης Viterbi μαλακών αποφάσεων για ένα μήνυμα που κωδικοποιήθηκε από τον κωδικοποιητή της Εικόνα 2.1. παρουσιάζεται στην Εικόνα 2.8.



Εικόνα 2.8 Παράδειγμα αποκωδικοποίησης Viterbi μαλακών αποφάσεων

3 Μηχανική Μάθηση και Νευρωνικά Δίκτυα

3.1 Βελτιστοποίηση βασιζόμενη στην κλίση

Οι περισσότεροι αλγόριθμοι μηχανικής μάθησης, όπως αυτοί που εκπαιδεύουν νευρωνικά δίκτυα, χρησιμοποιούν κάποιο είδος βελτιστοποίησης. Η βελτιστοποίηση είναι η διαδικασία ελαχιστοποίησης ή μεγιστοποίησης κάποιας συνάρτησης $f(x)$ μεταβάλλοντας κατάλληλα το διάνυσμα x .

Η συνάρτηση που χρήζει ελαχιστοποίησης στην περίπτωση των νευρωνικών δικτύων είναι η λεγόμενη συνάρτηση απωλειών ή συνάρτηση σφάλματος ή συνάρτηση κόστους. Η συνάρτηση απωλειών αποτελεί έναν τρόπο ποσοτικοποίησης των διαφορών που εμφανίζει η έξοδος ενός νευρωνικού από τις επιθυμητές αποκρίσεις και είναι συνάρτηση των συναπτικών βαρών του νευρωνικού δικτύου.

Χάριν ευκολίας θα περιγραφεί η διαδικασία της ελαχιστοποίησης για μια συνάρτηση $y = f(x)$ όπου το x είναι ένα βαθμωτό μέγεθος. Η παράγωγος αυτής της συνάρτησης συμβολίζεται ως $f'(x)$ ή $\frac{dy}{dx}$. Η παράγωγος $f'(x)$ συμβολίζει την κλίση της $f(x)$ στο σημείο x . Κατά προσέγγιση ισχύει πως $f(x + \varepsilon) \cong f(x) + \varepsilon f'(x)$.

Το πρόσημο της παραγώγου σε ένα συγκεκριμένο σημείο x μας παρέχει πληροφορίες για το πως μπορούμε να ελαχιστοποιήσουμε την $f(x)$, καθώς μας δείχνει την κατεύθυνση στην οποία πρέπει να μετακινηθεί το σημείο x για να μειωθεί το μέγεθος y . Γνωρίζουμε πως για ένα ε αρκετά μικρό ισχύει πως $f(x - \varepsilon \cdot \text{sign}(f'(x))) < f(x)$. Με διαδοχικές μικρές μετακινήσεις του x σε κατεύθυνση αντίθετη από το πρόσημο της παραγώγου η τιμή της $f(x)$ μειώνεται. Όταν η τιμή του x φτάσει σε ένα σημείο, όπου $f'(x) = 0$, η παράγωγος δεν δίνει πλέον πληροφορίες για την κατεύθυνση μετακίνησης και η διαδικασία τερματίζεται. Ο αλγόριθμος αυτός ονομάζεται κάθοδος κλίσης (gradient descent).

Τα σημεία όπου ισχύει η συνθήκη $f'(x) = 0$ ονομάζονται κρίσιμα σημεία της συνάρτησης f και μπορεί να είναι είτε τοπικά ακρότατα, είτε ολικά ακρότατα, είτε σαγματικά σημεία. Στην περίπτωση της καθόδου κλίσης, πρόκειται είτε για ελάχιστα, είτε για σαγματικά σημεία. Το βέλτιστο αποτέλεσμα είναι ο αλγόριθμος να καταλήξει σε ένα ολικό ελάχιστο της συνάρτησης f . Οι συναρτήσεις απωλειών συχνά εμφανίζουν πολλά τοπικά ελάχιστα και σαγματικά σημεία. Αυτό το γεγονός κάνει την διαδικασία της βελτιστοποίησης δύσκολη, ειδικά όταν το όρισμα δεν είναι βαθμωτό μέγεθος αλλά διανυσματικό μέγεθος. Για αυτό συνήθως γίνεται αποδεκτή και μία λύση η οποία είναι πολύ χαμηλή αλλά όχι απαραίτητα η ελάχιστη.

Για μια συνάρτηση με πολλά ορίσματα, ο κανόνας για την μετακίνηση του x , το οποίο σε αυτή την περίπτωση είναι ένα διανυσματικό μέγεθος, αλλάζει ως εξής:

$$x' = x - \varepsilon \cdot \nabla f(x) \quad (3.1)$$

Το μέγεθος $\nabla f(x)$ είναι ένα διάνυσμα του οποίου τα στοιχεία είναι οι μερικές παράγωγοι της συνάρτησης f και δείχνει προς την κατεύθυνση στην οποία η συνάρτηση παρουσιάζει την μεγαλύτερη μείωση (steepest descent).

Η παράμετρος ε ονομάζεται ρυθμός μάθησης και είναι ένα θετικό βαθμωτό μέγεθος, το οποίο καθορίζει το μέγεθος του βήματος σε κάθε επανάληψη του αλγορίθμου καθόδου κλίσης. Ο ρυθμός μάθησης μπορεί να καθοριστεί με πολλούς τρόπους. Μια δημοφιλής προσέγγιση είναι να τεθεί ίσος με έναν μικρό σταθερό αριθμό.

Σε καθεστώς πολλών μεταβλητών, ο αλγόριθμος καθόδου κλίσης συγκλίνει όταν όλες οι μερικές παράγωγοι της συνάρτησης ισούνται με μηδέν (ή με μια τιμή πολύ κοντά σε αυτό).

Η συντριπτική πλειονότητα των αλγορίθμων μηχανικής μάθησης χρησιμοποιεί την αλγόριθμο της καθόδου κλίσης ή κάποια παραλλαγή της.

3.2 Αναπαράσταση γνώσης

Όπως αναφέρθηκε, κατ' αναλογία με τους βιολογικούς οργανισμούς, τα νευρωνικά δίκτυα αποσκοπούν στην απόκτηση γνώσης μέσω της εκπαίδευσης. Ο όρος γνώση, σύμφωνα με τους Fischler και Firschein (1987), μπορεί να περιγραφεί ως η αποθηκευμένη πληροφορία ή τα μοντέλα που χρησιμοποιεί ένα άτομο ή μια μηχανή για να ερμηνεύσει, να προβλέψει και να αποκριθεί καταλλήλως στον κόσμο γύρω του. Αυτή η γνώση πρέπει βεβαίως να αναπαρασταθεί με τρόπο τέτοιο ώστε να γίνει αντιληπτή από το νευρωνικό δίκτυο.

Ένα νευρωνικό δίκτυο καλείται να μάθει ένα μοντέλο, το οποίο μπορεί να είναι συνεπές με την πραγματικότητα και να φέρνει εις πέρας του στόχους για τους οποίους έχει σχεδιαστεί. Οι παρατηρήσεις (μετρήσεις) επί του κόσμου, μέσω αισθητήρων, είναι η δεξαμενή πληροφοριών από την οποία αντλούνται παραδείγματα για την εκπαίδευση του νευρωνικού.

Τα παραδείγματα μπορεί να είναι χαρακτηρισμένα (labeled) ή μη χαρακτηρισμένα (unlabeled). Στα χαρακτηρισμένα παραδείγματα, κάθε παράδειγμα που αναπαριστά ένα σήμα εισόδου αντιστοιχίζεται με μια επιθυμητή απόκριση στην έξοδο. Από την άλλη τα μη χαρακτηρισμένα αποτελούνται από διαφορετικές υλοποιήσεις του σήματος εισόδου και μόνο. Σε κάθε περίπτωση ένα σύνολο παραδειγμάτων αποτελεί μια πηγή γνώσης για το νευρωνικό την οποία μπορεί να μάθει μέσω μιας διαδικασίας εκπαίδευσης. Τα χαρακτηρισμένα παραδείγματα είναι πιο δύσκολο να

συγκεντρωθούν λόγω της ανάγκης ύπαρξης ενός εκπαιδευτή που να μπορεί να παρέχει μια επιθυμητή απόκριση για κάθε σήμα εισόδου. Εν αντιθέσει, τα μη χαρακτηρισμένα παραδείγματα είναι πιο εύκολα στην συγκέντρωση τους καθώς δεν υπάρχει αυτή η ανάγκη.

Ένα σύνολο ζευγών σημάτων εισόδου και επιθυμητών αποκρίσεων αποτελεί ένα δείγμα εκπαίδευσης. Τα σήματα εισόδου και εξόδου αναπαρίστανται από διανύσματα. Δοθέντος ενός τέτοιου συνόλου, η υλοποίηση ενός νευρωνικού δικτύου μπορεί να διεξαχθεί ως εξής:

- Επιλέγεται μια κατάλληλη αρχιτεκτονική για το νευρωνικό δίκτυο, με ένα επίπεδο εισόδου, με κόμβους εισόδου ίσους στον αριθμό με της διάσταση των διανυσμάτων σημάτων εισόδου, και ένα επίπεδο εξόδου, με αριθμό νευρώνων ίσο με την διάσταση των διανυσμάτων των επιθυμητών εξόδων. Ένα υποσύνολο του δείγματος εκπαίδευσης χρησιμοποιείται για την εκπαίδευση του νευρωνικού δικτύου και την εκμάθηση των συναπτικών βαρών μέσω ενός κατάλληλου αλγορίθμου. Αυτή η φάση της σχεδίασης του δικτύου ονομάζεται «μάθηση» (learning).
- Μετέπειτα ελέγχεται η απόδοση του εκπαιδευμένου δικτύου με δεδομένα τα οποία δεν έχει συναντήσει ποτέ πριν το νευρωνικό. Σε αυτή τη φάση το δίκτυο τροφοδοτείται με ένα σύνολο σημάτων εισόδου χωρίς να του δοθούν στοιχεία για τις επιθυμητές εξόδους. Η απόδοση του δικτύου αξιολογείται μέσω της σύγκρισης της απόκρισης του δικτύου με τις επιθυμητές εξόδους. Αυτή η δεύτερη φάση λειτουργίας αποκαλείται «έλεγχος».

Το παραδείγματα που χρησιμοποιούνται για την εκπαίδευση του δικτύου μπορεί να είναι τόσο «θετικά» όσο και «αρνητικά». Για παράδειγμα, σε ένα πρόβλημα ανίχνευσης μέσω sonar, τα θετικά παραδείγματα είναι τα σήματα εισόδου που αντιστοιχούν στον στόχο που το σύστημα αποσκοπεί να εντοπίσει (λχ. εχθρικό σκάφος). Βέβαια στην περίπτωση ενός sonar η παρουσία θαλάσσιων πλασμάτων δρα ως «θόρυβος» και μπορεί να προκληθεί ψευδής συναγερμός. Για την ελαχιστοποίηση αυτού του προβλήματος, εντός του δείγματος σημάτων εισόδου, περιλαμβάνονται σκόπιμα και «αρνητικά» παραδείγματα που αντιστοιχούν σε παρεμβολές των θαλασσιών πλασμάτων, έτσι ώστε το δίκτυο να εκπαιδευτεί και να μην μπερδεύει τα θαλάσσια πλάσματα με τον πραγματικό του στόχο.

Το ζήτημα της κατάλληλης αναπαράστασης της γνώσης εντός τεχνητού νευρωνικού δικτύου είναι ένα εξαιρετικά δυσεπίλυτο πρόβλημα. Παρόλα αυτά υπάρχουν κάποιοι βασικοί κανόνες που πρέπει να διέπουν τη λύση:

- Οι όμοιες εισόδους που προέρχονται από όμοιες κλάσεις θα πρέπει συνήθως να έχουν όμοιες αναπαραστάσεις μέσα στο δίκτυο και κατά συνέπεια να ταξινομούνται από αυτό σε όμοια κλάση. Μερικά μέτρα για τον καθορισμό της ομοιότητας μεταξύ δύο εισόδων είναι η Ευκλείδεια απόσταση και το εσωτερικό γινόμενο.
- Για τα ανόμοια στοιχεία που πρόκειται να ταξινομηθούν ως διαφορετικές κλάσεις, θα πρέπει να υπάρχουν επαρκώς διαφορετικές αναπαραστάσεις.
- Εάν ένα συγκεκριμένο χαρακτηριστικό είναι σημαντικό, τότε πρέπει να χρησιμοποιηθεί μεγάλος αριθμός νευρώνων για την αναπαράσταση του στο δίκτυο.
- Η πρότερη πληροφορία και τα αναλλοίωτα δεδομένα πρέπει να ενσωματώνονται στην σχεδίαση του νευρωνικού δικτύου. Έτσι απλοποιείται η διαδικασία σχεδίασης καθώς δεν χρειάζεται να τα μάθει. Αυτό έχει ως αποτέλεσμα ένα νευρωνικό με εξειδικευμένη δομή. Ένα νευρωνικό δίκτυο με εξειδικευμένη δομή έχει λιγότερες ελεύθερες παραμέτρους που χρήζουν προσδιορισμού. Ως εκ τούτου, απαιτείται μικρότερο σύνολο δεδομένων για εκπαίδευση και το δίκτυο μαθαίνει γρηγορότερα. Ένα εξειδικευμένο δίκτυο εμφανίζει ακόμη μεγαλύτερο ρυθμό εξυπηρέτησης και μικρότερο κόστος υλοποίησης λόγω του μικρότερου μεγέθους του σε σχέση με ένα πλήρως συνδεδεμένο δίκτυο.
 - Για την ενσωμάτωση της πρότερης πληροφορίας, δεν υπάρχουν καλά ορισμένες διαδικασίες. Αυτή η πρότερη πληροφορία μπορεί να είναι η γνώση πως κάποια συγκεκριμένη είσοδος του νευρωνικού δικτύου δεν επηρεάζει όλες τις εξόδους του. Δύο τεχνικές που δίνουν αποτελέσματα είναι ο περιορισμός της αρχιτεκτονικής του δικτύου μέσω της χρήσης τοπικών συνδέσεων γνωστές και ως δεκτικά πεδία και ο περιορισμός της επιλογής συναπτικών βαρών. Για παράδειγμα, δεν χρειάζεται να υπάρχει σύναψη μεταξύ όλων των κόμβων εισόδου και κάθε νευρώνα του πρώτου κρυφού επιπέδου. Οι κόμβοι εισόδου οι οποίοι έχουν σύναψη με έναν νευρώνα αποτελούν το δεκτικό πεδίο του νευρώνα
 - Για να γίνει πιο σαφής η έννοια των αναλλοίωτων δεδομένων, θα δοθεί ένα παράδειγμα. Έστω πως ένα αντικείμενο περιστρέφεται. Ο παρατηρητής αν και θα παρατηρήσει την οπτική διαφορά, θα κατανοήσει πως πρόκειται για το ίδιο αντικείμενο, καθώς το αντικείμενο είναι αναλλοίωτο στον μετασχηματισμό περιστροφής. Η ενσωμάτωση αυτών των αναλλοίωτων στους μετασχηματισμούς είναι συνήθως δύσκολο να αποτυπωθεί εντός της δομής του νευρωνικού δικτύου καθώς συχνά είναι πολύ μεγάλος ο αριθμός των συνάψεων. Η επιβολή τους μέσω της επιλογής καταλλήλων παραδειγμάτων κατά την διαδικασία εκπαίδευσης (πολλές

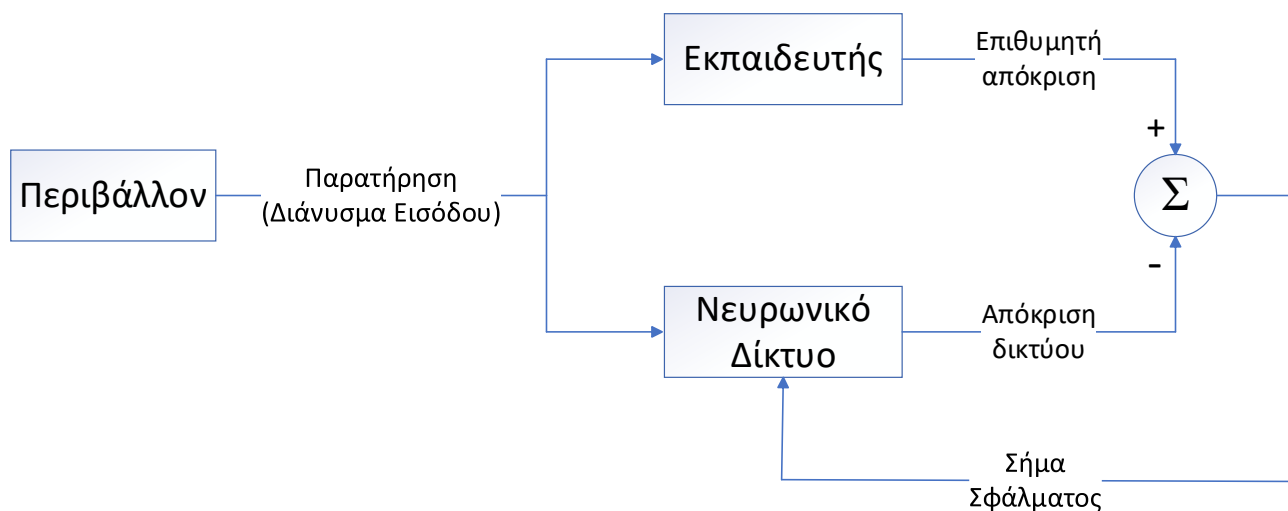
διαφορετικές αναπαραστάσεις του ίδιου αντικειμένου στα σήματα εισόδου) οδηγεί σε πολύ μεγάλο επεξεργαστικό φόρτο. Η χρήση ενός προεπεξεργαστή για την εξαγωγή των αμεταβλήτων χαρακτηριστικών αποτελεί μια καλύτερη λύση καθώς μειώνεται ο αριθμός των χαρακτηριστικών που χρειάζεται να επεξεργαστούν από το δίκτυο και χαλαρώνουν οι προδιαγραφές που επιβάλλονται στη σχεδίαση του δικτύου.

3.3 Διαδικασίες μάθησης

Οι διαδικασίες μάθησης που χρησιμοποιούνται από αλγόριθμους μηχανικής μάθησης μπορούν να κατηγοριοποιηθούν σε δύο γενικές κατηγορίες:

- Μάθηση με εκπαιδευτή
- Μάθηση χωρίς εκπαιδευτή, η οποία μπορεί να χωριστεί σε δύο υποκατηγορίες:
 - Ενισχυτική μάθηση
 - Μη επιβλεπόμενη μάθηση

3.3.1 Μάθηση με εκπαιδευτή



Εικόνα 3.1 Σχηματικό διάγραμμα μάθησης με εκπαιδευτή

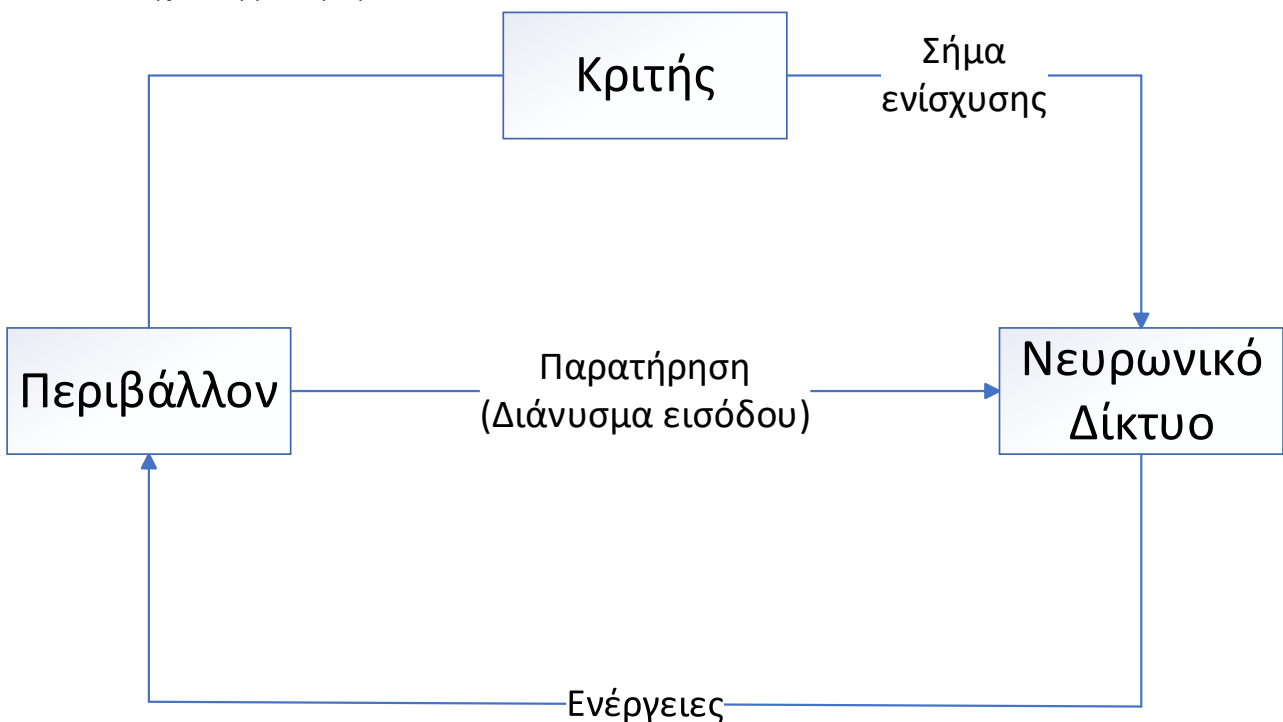
Στην Εικόνα 3.1 απεικονίζεται ένα σχηματικό διάγραμμα της μάθησης με εκπαιδευτή ή επιβλεπόμενης μάθησης. Ας υποθεθεί πως το νευρωνικό δίκτυο και ο εκπαιδευτής λαμβάνουν μια παρατήρηση υπό τη μορφή διανύσματος εισόδου που έχει αντληθεί από το ίδιο περιβάλλον. Ο εκπαιδευτής έχει εγγενή γνώση του περιβάλλοντος και είναι ικανός να παρέχει μια επιθυμητή απόκριση για κάθε πιθανό διάνυσμα εισόδου.

Αυτή η επιθυμητή απόκριση αποτελεί μέτρο σύγκρισης για την απόκριση του δικτύου. Το νευρωνικό δίκτυο τροφοδοτείται με ένα σήμα σφάλματος, το οποίο εκφράζει διαφορά μεταξύ

απόκρισης του δικτύου και απόκρισης του εκπαιδευτή. Αυτό το σήμα σφάλματος λαμβάνεται υπόψιν από το νευρωνικό ώστε να προσαρμόσει τις ελεύθερες παραμέτρους με τέτοιο τρόπο ώστε να προσεγγίσει τη λειτουργία του εκπαιδευτή. Μέσω της επανάληψης αυτής της διαδικασίας για πλήθος διαφορετικών διανυσμάτων εισόδου, το νευρωνικό δίκτυο φθάνει σε μια κατάσταση η οποία προσομοιώνει την λειτουργία του εκπαιδευτή. Κατ' αυτό τον τρόπο, η γνώση του περιβάλλοντος, η οποία βρίσκεται στον εκπαιδευτή, μεταλαμπαδεύεται στο νευρωνικό μέσω εκπαίδευσης και αποθηκεύεται στα συναπτικά βάρη. Όταν επιτευχθεί αυτό, πλέον το νευρωνικό μπορεί να λειτουργήσει αυτόνομα χωρίς την παρουσία εκπαιδευτή.

Η διαδικασία της επιβλεπόμενης μάθησης συνιστά ένα σύστημα ανάδρασης κλειστού βρόχου. Ως μέτρο απόδοσης του συστήματος χρησιμοποιείται μια συνάρτηση απωλειών της οποίας η τιμή εξαρτάται από τις τιμές των συναπτικών βαρών (π.χ. μέσο τετραγωνικό σφάλμα επί του δείγματος εκπαίδευσης). Για να βελτιωθεί η λειτουργία του δικτύου με την πάροδο του χρόνου, χρειάζεται το νευρωνικό δίκτυο να μεταβάλλει τα συναπτικά του βάρη, έτσι ώστε να προσεγγίζεται ένα ελάχιστο της συνάρτησης απωλειών, τοπικό ή γενικό. Ένα σύστημα επιβλεπόμενης μάθησης πετυχαίνει την ελαχιστοποίηση της συνάρτησης απωλειών εκτιμώντας την κλίση (gradient) της συνάρτησης απωλειών και εκτελώντας έναν αλγόριθμο καθόδου κλίσης.

3.3.2 Ενισχυτική μάθηση



Εικόνα 3.2 Σχηματικό διάγραμμα ενισχυτικής μάθησης

Στην Εικόνα 3.2 παρουσιάζεται ένα σχηματικό διάγραμμα της διαδικασίας της ενισχυτικής μάθησης. Σε αυτήν την περίπτωση απουσιάζει ένας εκπαιδευτής και η διαδικασία της εκπαίδευσης

του νευρωνικού δικτύου γίνεται μέσω της αλληλεπίδρασης με το περιβάλλον. Στόχος της εκπαίδευσης και σε αυτήν περίπτωση είναι η ελαχιστοποίηση ενός βαθμωτού δείκτη απόδοσης.

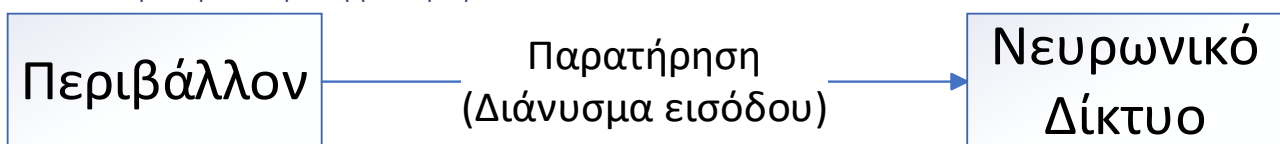
Ας υποθεθεί πως από την παρατήρηση του περιβάλλοντος προκύπτει ένα διάνυσμα που δίνεται στο νευρωνικό ως σήμα εισόδου. Με βάση αυτήν την είσοδο το νευρωνικό θα παράγει ένα σήμα εξόδου βάσει του οποίου θα καθοριστεί κάποια ενέργεια οι οποία θα επηρεάσει την κατάσταση του περιβάλλοντος. Στη συνέχεια μέσω ενός μηχανισμού που λειτουργεί ως κριτής και παρακολουθεί το περιβάλλον, το εκπαιδευόμενο δίκτυο θα τροφοδοτηθεί με ένα σήμα ενίσχυσης (reinforcement signal). Αυτό το σήμα ενίσχυσης μπορεί να είναι είτε «ποινές» είτε «αμοιβές» που λαμβάνει το δίκτυο με βάση τα αποτελέσματα που είχαν οι ενέργειες που αποφάσισε. Η ενίσχυση συνήθως είναι καθυστερούμενη, το οποίο σημαίνει πως το δίκτυο λαμβάνει πλήθος διανυσμάτων εισόδου και γίνεται πληθώρα ενεργειών προτού λάβει ένα σήμα ενίσχυσης. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να εκπαιδευτεί επαρκώς το δίκτυο.

Στόχος της ενισχυτικής μάθησης είναι το νευρωνικό δίκτυο να προσαρμόσει τη λειτουργία του έτσι ώστε να μεγιστοποιεί την συνάρτηση κέρδους, η οποία συγκεντρώνει τις διαδοχικές αμοιβές που λαμβάνει το σύστημα μέσω του σήματος ενίσχυσης (ή να ελαχιστοποιεί την συνάρτηση κόστους).

Η καθυστερούμενη ενισχυτική μάθηση εμφανίζει τις εξής προκλήσεις:

- Η απουσία εκπαιδευτή συνεπάγεται πως δεν παρέχεται επιθυμητή απόκριση ως μέτρο σύγκρισης σε κάθε βήμα της εκπαίδευσης
- Η καθυστέρηση στο σήμα ενίσχυσης συνεπάγεται πως το εκπαιδευόμενο σύστημα πρέπει να είναι σε θέση να καθορίσει από μόνο του το βαθμό επιτυχίας για κάθε ενέργεια της χρονικής αλληλουχίας που οδηγεί στο τελικό αποτέλεσμα. Ο μηχανισμός ενίσχυσης αποτιμά μόνο το τελικό αποτέλεσμα.

3.3.3 Μη επιβλεπόμενη μάθηση



Εικόνα 3.3 Σχηματικό διάγραμμα μη επιβλεπόμενης μάθησης

Στην Εικόνα 3.3 απεικονίζεται ένα σχηματικό διάγραμμα της μη επιβλεπόμενης μάθησης. Σε αυτό το είδος μάθησης δεν υφίσταται η έννοια ενός κριτή που να εποπτεύει το σύστημα. Αντί αυτών των μηχανισμών ελέγχου υπάρχει ένα ανεξάρτητο από την εργασία μέτρο της ποιότητας της λειτουργίας του δικτύου βάσει το οποίου προσαρμόζονται τα συναπτικά βάρη του νευρωνικού

δικτύου. Για ένα ανεξάρτητο από την εργασία μέτρο, αφού το δίκτυο συντονιστεί με τις στατιστικές κατανομές των δεδομένων εισόδου, αναπτύσσει τη δυνατότητα να σχηματίζει εσωτερικές αναπαραστάσεις για την κωδικοποίηση των χαρακτηριστικών εισόδου και μέσω αυτών να δημιουργεί νέες κλάσεις αυτόματα (Becker, 1991).

Για την εκτέλεση αυτής της διαδικασίας μάθησης, μπορεί να χρησιμοποιηθεί ανταγωνιστική μάθηση. Ένα παράδειγμα ανταγωνιστικής μάθησης είναι το εξής: έστω πως έχει αναπτυχθεί ένα νευρωνικό δίκτυο με δύο επίπεδα, ένα επίπεδο εισόδου και ένα ανταγωνιστικό επίπεδο. Οι νευρώνες του ανταγωνιστικού επιπέδου ανταγωνίζονται μεταξύ τους βάσει ενός κανόνα, για να αποκριθούν σε χαρακτηριστικά των δεδομένων εισόδου.

3.4 Το πρόβλημα της υπερπροσαρμογής και της υπόπροσαρμογής

Η καλή απόκριση ενός νευρωνικού δικτύου σε ένα συγκεκριμένο σύνολο δεδομένων εκπαίδευσης δεν εγγυάται ότι το νευρωνικό δίκτυο θα παρέχει καλές αποκρίσεις σε δεδομένα που δεν έχει συναντήσει κατά την εκπαίδευση του. Αυτό το φαινόμενο γίνεται πιο έντονο όταν η δομή του νευρωνικού δικτύου είναι περίπλοκη ή όταν το σύνολο δεδομένων εκπαίδευσης είναι μικρό. Σκοπός του σχεδιαστή είναι να υπερκεράσει αυτό το πρόβλημα και να προσδώσει στο σχεδιαζόμενο νευρωνικό δίκτυο την ικανότητα να ανταποκρίνεται επιτυχώς και σε άγνωστα δεδομένα εισόδου. Αυτή η ικανότητα αποκαλείται συχνά ως γενίκευση (generalization).

Κατά την διαδικασία της εκπαίδευσης ενός νευρωνικού δικτύου, παρέχεται ένα συγκεκριμένο σύνολο δεδομένων εκπαίδευσης. Σκοπός της εκπαίδευσης είναι να ελαχιστοποιήσει το σφάλμα επί του συγκεκριμένου συνόλου εκπαίδευσης, το λεγόμενο σφάλμα εκπαίδευσης, το οποίο εκφράζεται από τη συνάρτηση απωλειών. Αυτό είναι ένα πρόβλημα βελτιστοποίησης. Το νευρωνικό δίκτυο σε αυτήν την περίπτωση μπορεί να προσαρμόσει τα βάρη των συνάψεων του έτσι ώστε να μιμηθεί ένα συγκεκριμένο μοντέλο, το οποίο μπορεί να εξασφαλίζει μια επιθυμητή απόκριση για το σύνολο εκπαίδευσης που του έχει παρασχεθεί. Το σύνολο εκπαίδευσης, όμως, δεν είναι συχνά παρά ένα μικρό υποσύνολο όλων των πιθανών περιπτώσεων και το μοντέλο που θα μάθει το νευρωνικό μπορεί να είναι ανίκανο να ανταπεξέλθει σε δεδομένα που δεν έχει συναντήσει.

Όπως γίνεται αντιληπτό, η σχεδίαση και η εκπαίδευση ενός νευρωνικού δικτύου δεν είναι απλώς ένα πρόβλημα βελτιστοποίησης. Στην μηχανική μάθηση είναι ζητούμενο και η μείωση του σφάλματος γενίκευσης. Το σφάλμα γενίκευσης ή σφάλμα ελέγχου είναι η μέση τιμή του σφάλματος για μια νέα άγνωστη είσοδο. Αυτή η μέση τιμή προκύπτει παρέχοντας ένα σύνολο εισόδων που δεν

έχει συναντήσει κατά την εκπαίδευση του το νευρωνικό και παρατηρώντας την απόκριση του. Αυτό το σύνολο ονομάζεται σύνολο ελέγχου (test set).

Με βάση τα προαναφερθέντα, οι δύο παράγοντες που καθορίζουν την αποτελεσματικότητα ενός αλγορίθμου μηχανικής μάθησης είναι οι εξής:

- Ικανότητα ελαχιστοποίησης του σφάλματος εκπαίδευσης
- Ικανότητα μείωσης του χάσματος μεταξύ σφάλματος εκπαίδευσης και σφάλματος ελέγχου

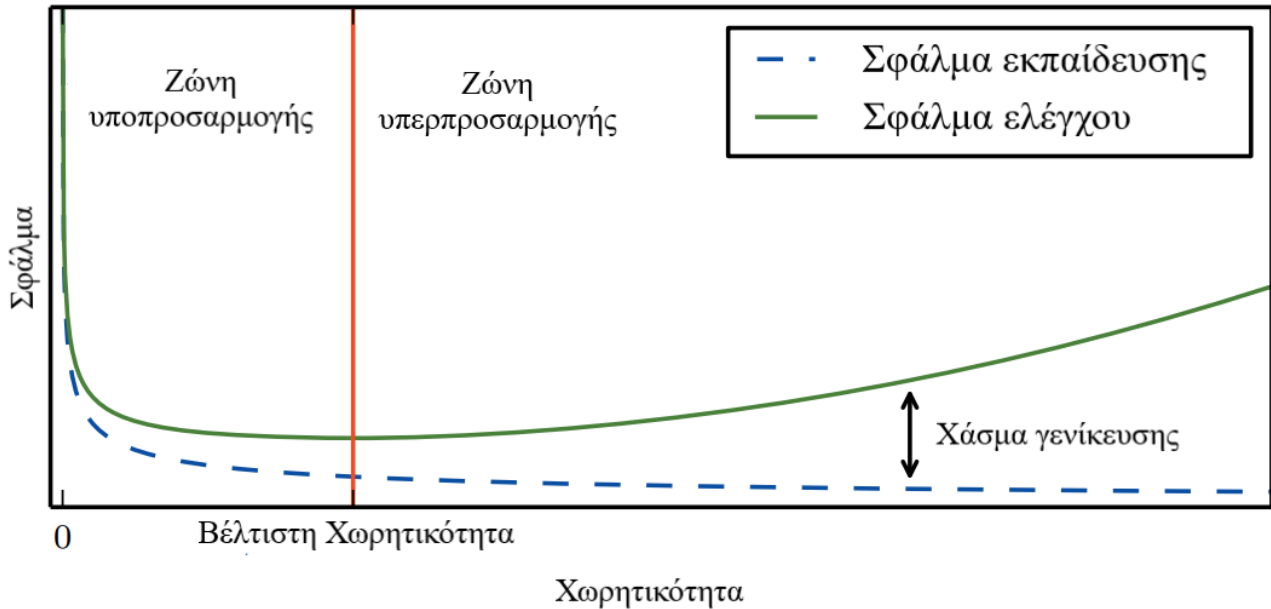
Αυτοί οι δύο παράγοντες αντιστοιχούν στις δύο κύριες προκλήσεις της μηχανικής μάθησης: την υποπροσαρμογή και την υπερπροσαρμογή. Υποπροσαρμογή συμβαίνει όταν ένα σύστημα μηχανικής μάθησης δεν μπορεί να μειώσει αρκετά το σφάλμα εκπαίδευσης. Υπερπροσαρμογή συμβαίνει όταν το χάσμα μεταξύ σφάλματος εκπαίδευσης και σφάλματος ελέγχου είναι πολύ μεγάλο.

Μπορούμε να συμπεράνουμε τότε ένα σύστημα μηχανικής μάθησης είναι πιθανότερο να υπερπροσαρμοστεί ή να υποπροσαρμοστεί ανάλογα με την χωρητικότητα του αλγορίθμου. Η χωρητικότητα ενός συστήματος μηχανικής μάθησης αναφέρεται στην ικανότητα του να μαθαίνει και να αναγνωρίζει μοτίβα πάνω στα δεδομένα. Η χωρητικότητα ενός νευρωνικού δικτύου εξαρτάται άμεσα από την αρχιτεκτονική του. Ένα πιο σύνθετο νευρωνικό δίκτυο έχει υψηλότερη χωρητικότητα καθώς μπορεί να αναγνωρίζει πολυπλοκότερες σχέσεις μεταξύ των δεδομένων. Πολλές φορές, όμως, η υπερβολικά μεγάλη χωρητικότητα μπορεί να οδηγήσει σε υπερπροσαρμογή, καθώς το νευρωνικό δίκτυο μαθαίνει πολύ καλά το σύνολο δεδομένων πάνω στο οποίο εκπαιδεύεται, αλλά δυσκολεύεται να ανταποκριθεί σε δεδομένα που δεν έχει συναντήσει. Από την άλλη ένα νευρωνικό δίκτυο με χαμηλή χωρητικότητα, μπορεί να δυσκολευτεί να αναγνωρίσει περίπλοκα μοτίβα μεταξύ των δεδομένων εισόδου και των επιθυμητών αποκρίσεων και να εμφανίζει το φαινόμενο της υποπροσαρμογής. Για αυτούς τους λόγους, η αρχιτεκτονική ενός δικτύου πρέπει να επιλέγεται βάσει της πολυπλοκότητας του προβλήματος που του έχει ανατεθεί να επιλύσει.

Μια ενέργεια που βελτιώνει την δυνατότητα του νευρωνικού δικτύου να γενικεύει είναι η αύξηση του μεγέθους του συνόλου εκπαίδευσης. Ένας καλός εμπειρικός κανόνας είναι ο αριθμός των παραδειγμάτων που περιέχει ένα σύνολο εκπαίδευσης να είναι 2 με 3 φορές μεγαλύτερος από τις συνολικές ελεύθερες παραμέτρους του συστήματος μηχανικής μάθησης, χωρίς αυτό να είναι δεσμευτικό καθώς πρέπει να ληφθεί υπόψιν και η πολυπλοκότητα της λειτουργίας που ο σχεδιαστής θέλει να επιτελέσει το νευρωνικό. Σε θεωρητικό επίπεδο ένα καταλλήλως σχεδιασμένο

νευρωνικό δίκτυο μπορεί να προσομοιάσει οποιαδήποτε λειτουργία, όμως πολύ συχνά η διαδικασία συγκέντρωσης καταλλήλων δεδομένων είναι πολύ δύσκολη και αργή.

Η γενικότερη σχέση μεταξύ σφαλμάτων (εκπαίδευσης και ελέγχου) και χωρητικότητας φαίνεται στην παρακάτω Εικόνα 3.4.



Εικόνα 3.4 Γενικότερη σχέση χωρητικότητας και σφαλμάτων

3.5 Υπερπαραμέτροι και σύνολα επαλήθευσης

Οι περισσότεροι αλγόριθμοι μηχανικής μάθησης έχουν κάποιες υπερπαραμέτρους. Οι υπερπαραμέτροι είναι τιμές που μπορούν να χρησιμοποιηθούν από τον σχεδιαστή ενός αλγορίθμου μηχανικής μάθησης έτσι ώστε να καθοριστεί η ακριβής συμπεριφορά του αλγορίθμου. Οι τιμές αυτές δεν μπορούν να προσαρμοστούν από τον ίδιο τον αλγόριθμο. Τέτοιες τιμές μπορεί να είναι ο ρυθμός μάθησης και ο αριθμός των νευρώνων σε κάθε επίπεδο ενός νευρωνικού δικτύου.

Πολλές φορές όμως η τιμή των υπερπαραμέτρων, η οποία καθορίζεται αρχικώς από τον σχεδιαστή, δεν είναι η καλύτερη δυνατή για το σύστημα. Για να επιλυθεί αυτό το ζήτημα, χρειάζεται ένας μηχανισμός ώστε να γίνεται προσαρμογή των υπερπαραμέτρων κατά την διάρκεια εκτέλεσης του αλγορίθμου μάθησης ενός νευρωνικού δικτύου.

Για την επίλυση αυτού του προβλήματος, χρειάζεται η δημιουργία ενός νέου συνόλου δεδομένων, του συνόλου επαλήθευσης. Αυτό το σύνολο πρέπει να είναι διαφορετικό από το σύνολο ελέγχου. Το σύνολο ελέγχου πρέπει να μην έχει χρησιμοποιηθεί για τη ρύθμιση καμίας παραμέτρου του

μοντέλου μηχανικής μάθησης, συμπεριλαμβανομένων και των υπερπαραμέτρων. Συνήθως ως σύνολο επαλήθευσης χρησιμοποιείται ένα μέρος του συνόλου εκπαίδευσης. Οπότε το σύνολο εκπαίδευσης χωρίζεται σε 2 τμήματα: ένα που αφορά την εκμάθηση των συναπτικών βαρών και συνεχίζει να αποκαλείται σύνολο εκπαίδευσης, και ένα άλλο που χρησιμοποιείται για την προσαρμογή των υπερπαραμέτρων σε τακτά διαστήματα, το οποίο ονομάζεται σύνολο επαλήθευσης. Συνηθίζεται ένα ποσοστό της τάξης του 80% να χρησιμοποιείται για εκμάθηση και ένα ποσοστό της τάξης του 20% για το σύνολο επαλήθευσης.

Πέραν της προσαρμογής των υπερπαραμέτρων, το σύνολο επαλήθευσης χρησιμοποιείται και για την εκτίμηση του σφάλματος ελέγχου. Καθώς το σύνολο επαλήθευσης χρησιμοποιείται για την προσαρμογή των υπερπαραμέτρων, συνήθως το σφάλμα του απέχει από το σφάλμα ελέγχου αλλά σε μικρότερο βαθμό από ότι το σφάλμα εκπαίδευσης. Αυτήν την εκτίμηση του σφάλματος ελέγχου, χρησιμοποιεί και μια τεχνική που ονομάζεται *early stopping*. Αυτή η τεχνική σταματά την εκτέλεση του αλγορίθμου καθόδου κλίσης πρόωρα. Αυτός ο τερματισμός οφείλεται στην αύξηση του χάσματος μεταξύ σφάλματος εκπαίδευσης και σφάλματος επαλήθευσης.

3.6 Νευρωνικά δίκτυα πρόσθιας τροφοδότησης (MLPs)

Τα νευρωνικά δίκτυα πρόσθιας τροφοδότησης ή αλλιώς *Perceptron* πολλών επιπέδων (*Multilayer Perceptron*, MLP) είναι ένας από τους βασικούς και δημοφιλείς τύπους νευρωνικών δικτύων που χρησιμοποιούνται σήμερα. Σκοπός τους είναι να προσεγγίσουν κάποια λειτουργία f . Για παράδειγμα ένας ταξινομητής, $y = f(x)$ ταξινομεί μια είσοδο x σε μία κατηγορία y . Ένα δίκτυο πρόσθιας τροφοδότησης θα προσπαθήσει να επιτελέσει μια λειτουργία $y = \tilde{f}(x; \kappa)$, η οποία θα προσεγγίζει την λειτουργία f , αφότου το δίκτυο μάθει τις κατάλληλες τιμές για τις παραμέτρους κ .

Καλούνται δίκτυα πρόσθιας τροφοδότησης, γιατί η πληροφορία ρέει από το επίπεδο εισόδου, όπου εισέρχεται η είσοδος x , στα ενδιάμεσα κρυφά επίπεδα του νευρωνικού και τέλος από αυτά στο επίπεδο εξόδου που παράγει μια έξοδο y . Δεν υπάρχει καμία σύνδεση ανατροφοδότησης, η οποία να τροφοδοτεί κάποια τιμή της εξόδου πίσω στο νευρωνικό.

Αποκαλούνται «δίκτυα», επειδή γενικά η λειτουργία τους μπορεί να εκφραστεί μαθηματικά ως μια σύνθεση συναρτήσεων. Για παράδειγμα, ένα τέτοιο δίκτυο μπορεί να αναπαρασταθεί με μια αλυσιδωτή σχέση όπως αυτή: $f(x) = f^{(3)}\left(f^{(2)}\left(f^{(1)}(x)\right)\right)$. Η συναρτήσεις $f^{(1)}$, $f^{(2)}$ και $f^{(3)}$ εκφράζουν την λειτουργία του πρώτου, του δεύτερου και του τρίτου επιπέδου αντίστοιχα. Το συνολικό μήκος της αλυσίδας προσδιορίζει το βάθος του μοντέλου. Το τελευταίο επίπεδο ονομάζεται επίπεδο εξόδου, καθώς η έξοδος του είναι και η έξοδος του νευρωνικού δικτύου. Τα

ενδιάμεσα επίπεδα ονομάζονται κρυφά επίπεδα καθώς η έξοδος τους δεν παρουσιάζεται στο τελικό αποτέλεσμα αλλά βοηθάει στην αναγνώριση σχέσεων μεταξύ εισόδου και εξόδου.

Όπως αναφέρθηκε και στις προηγούμενες ενότητες του κεφαλαίου αυτού, για να αξιολογηθεί η απόκριση του νευρωνικού δικτύου χρησιμοποιείται μια συνάρτηση απωλειών και σκοπός του αλγορίθμου μηχανικής μάθησης που εκπαιδεύει το δίκτυο είναι να την ελαχιστοποιήσει. Στην περίπτωση των MLP συνήθως γίνεται χρήση του αλγορίθμου οπισθοδιάδοσης (backpropagation, BP), ο οποίος είναι μια παραλλαγή του αλγορίθμου καθόδου κλίσης (Ενότητα 3.1). Στη συνέχεια παρατίθεται μια βήμα προς βήμα εξήγηση του αλγορίθμου:

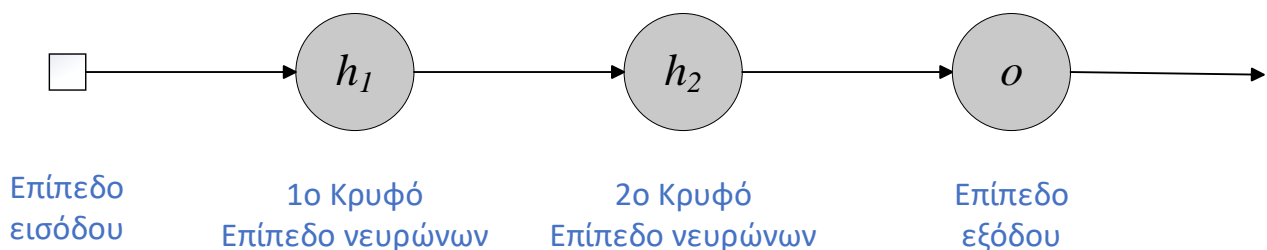
Βήμα 1. Το δίκτυο τροφοδοτείται με δεδομένα του συνόλου εκπαίδευσης στην είσοδο και αυτά «περνούν» μέσα από τα επίπεδα του μέχρι να παραχθεί έξοδος. Η έξοδος συγκρίνεται με την επιθυμητή απόκριση και το σφάλμα υπολογίζεται με τη χρήση μια συνάρτησης απωλειών. (Πρόσθια Διάδοση)

Βήμα 2. Υπολογίζεται η κλίση (∇) της συνάρτησης απωλειών συναρτήσει των συναπτικών βαρών χρησιμοποιώντας τον κανόνα της αλυσίδας του Leibniz. Ξεκινώντας από το επίπεδο εξόδου και κινούμενοι προς τα πίσω μέσα στο δίκτυο, οι μερικές παράγωγοι των συναπτικών βαρών υπολογίζονται σε κάθε επίπεδο. Τα βάρη ενημερώνονται με βάση τη σχέση (3.1) (Ενότητα 3.1) και χρησιμοποιούνται στην επόμενη επανάληψη. (Οπισθοδιάδοση)

Βήμα 3. Επανάληψη των βημάτων μέχρι ο αλγόριθμος να συγκλίνει (μηδενισμός μερικών παραγώγων).

Ο αλγόριθμος οπισθοδιάδοσης (για ένα διάνυσμα εισόδου), όπως αντιλαμβάνεται κανείς, έχει πολυπλοκότητα $O(N \cdot D)$, όπου το D συμβολίζει το βάθος του μοντέλου και το N τον αριθμό των νευρώνων σε κάθε επίπεδο. Καθώς αυξάνεται ο αριθμός των επιπέδων, αυξάνεται και το «μήκος διάδοσης» της πληροφορίας (πρόσθιας και οπίσθιας).

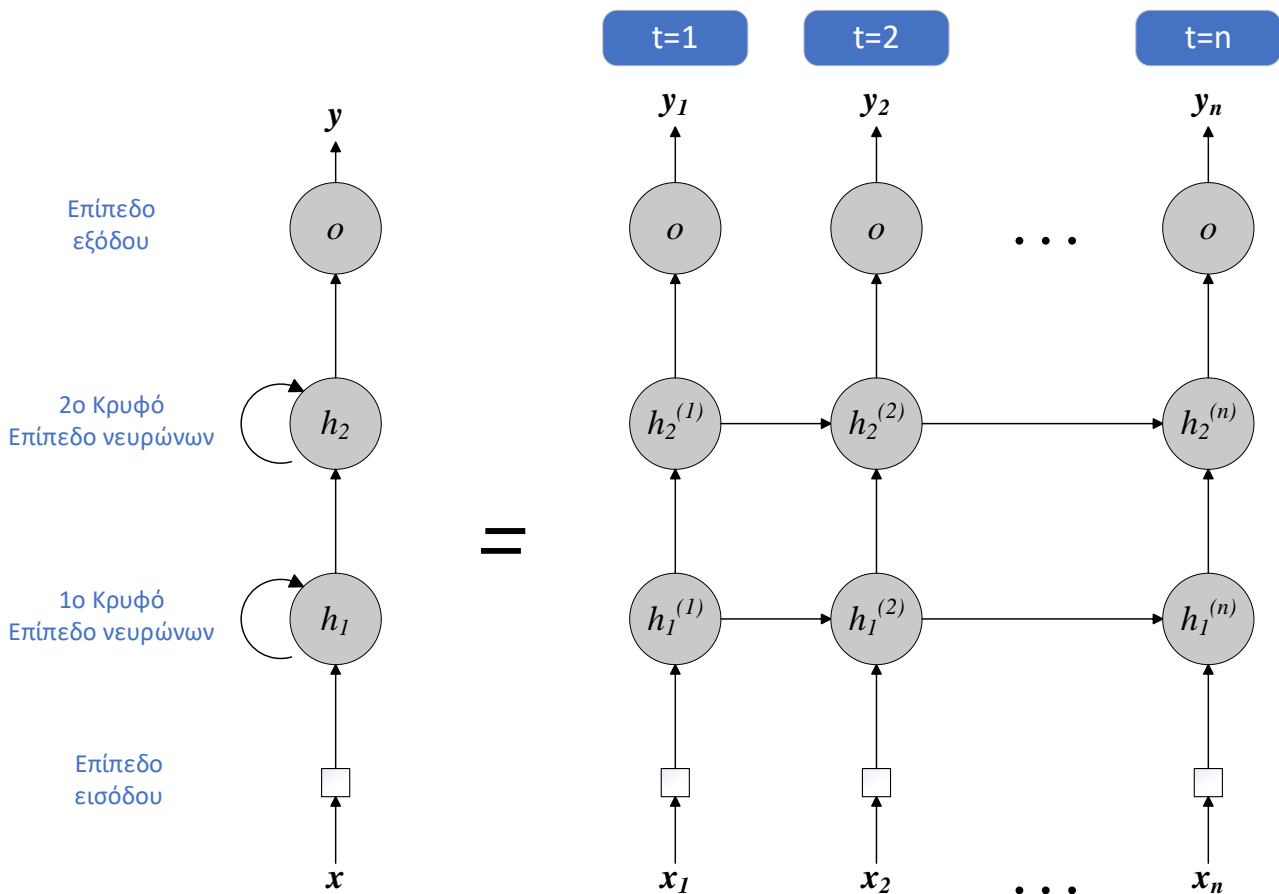
Η γενική δομή ενός δικτύου πρόσθιας τροφοδότησης παρουσιάζεται στην Εικόνα 3.5



Εικόνα 3.5 Δίκτυο πρόσθιας τροφοδότησης (MLP)

3.7 Αναδρομικά Νευρωνικά Δίκτυα (RNNs)

Εν αντιθέσει με τα MLPs, τα αναδρομικά νευρωνικά δίκτυα (Recurrent Neural Networks, RNN) περιέχουν στην δομή τους κάποιου είδους σύνδεση ανατροφοδότησης. Η ανάγκη τέτοιων δικτύων προκύπτει από το γεγονός ότι τα MLPs δεν είναι ικανά να επεξεργαστούν αποδοτικά σειριακά/ακολουθιακά δεδομένα λόγω της δομής τους. Στην περίπτωση των σειριακών δεδομένων, για να υπολογιστεί μια έξοδος χρειάζεται να υπάρχει γνώση και προηγούμενων εισόδων πέραν της τρέχουσας. Στα MLP κάθε είσοδος είναι ανεξάρτητη από την προηγούμενη. Παράδειγμα σειριακών δεδομένων είναι μια πρόταση. Για παράδειγμα, αν ένα νευρωνικό θέλει να κάνει μια πρόβλεψη για την τελευταία λέξη μιας πρότασης χρειάζεται να «θυμάται» όλες τις προηγούμενες λέξεις. Η γενική δομή ενός RNN (με δύο κρυφά επίπεδα) φαίνεται στην Εικόνα 3.6.



Εικόνα 3.6 Αναδρομικό νευρωνικό δίκτυο (RNN)

Όπως μπορεί εύκολα να διακρίνει κανείς, η διαφορά μεταξύ MLP και RNN είναι η χρήση βρόχων ανάδρασης στο RNN. Αυτό οδηγεί την πληροφορία να μην διαδίδεται μόνο «πρόσθια» κατά μήκος των επιπέδων του νευρωνικού αλλά και κατά «μήκος του χρόνου» μέσω των βρόχων ανάδρασης.

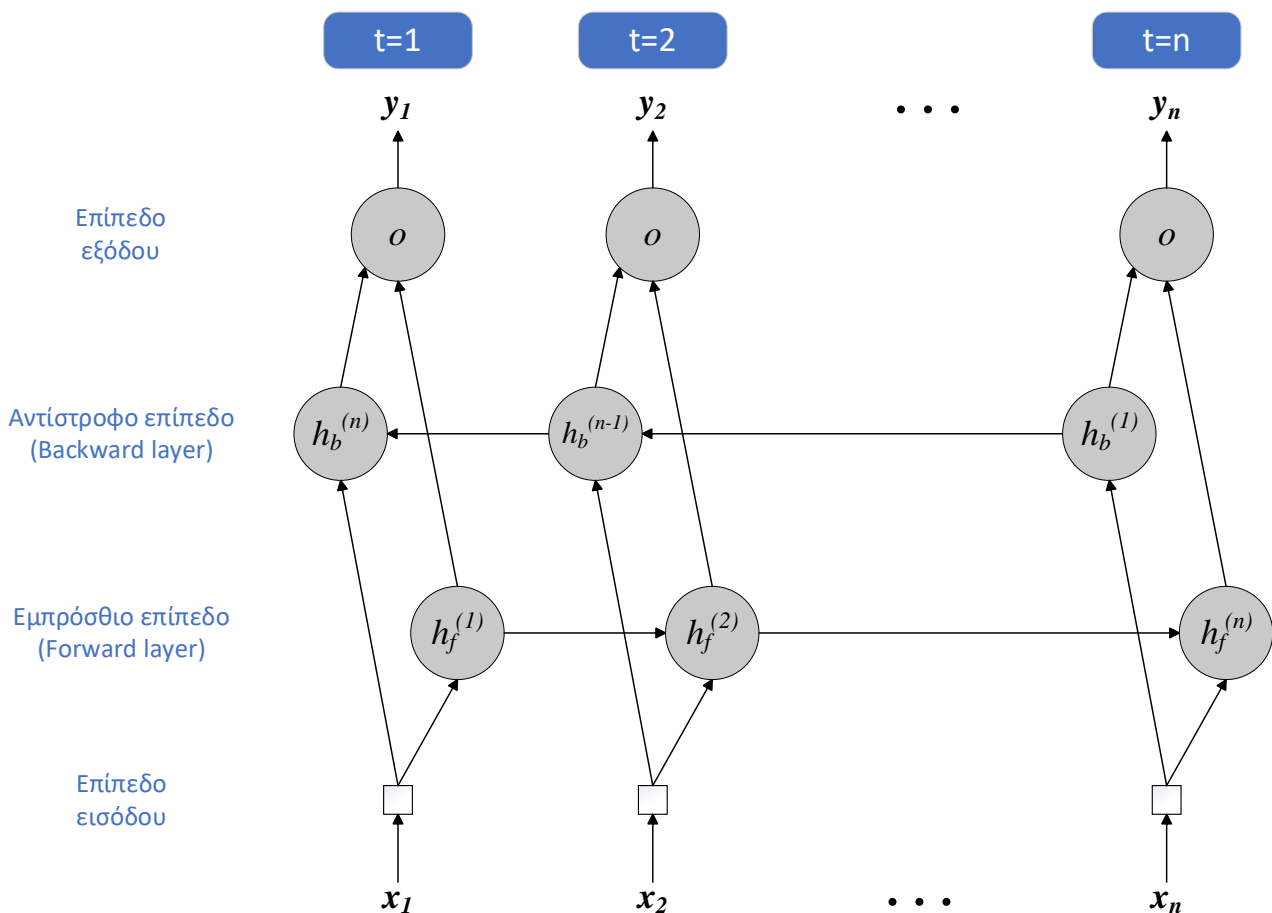
Ο δημοφιλέστερος αλγόριθμος μηχανικής μάθησης, ο οποίος χρησιμοποιείται στα RNN, είναι ο αλγόριθμος οπισθοδιάδοσης μέσω χρόνου (Backpropagation through time, BPTT) και είναι επί της

ουσίας μια επέκταση του αλγορίθμου οπισθοδιάδοσης, ο οποίος χρησιμοποιείται στα MLP. Η υπολογιστική πολυπλοκότητα του αυξάνεται λόγω της εισαγωγής και της χρονικής διάστασης.

3.8 Αναδρομικά νευρωνικά δίκτυα διπλής κατεύθυνσης (Bidirectional RNNs)

Ένα μειονέκτημα των RNN είναι πως ανά πάσα στιγμή υπάρχει γνώση μόνο για τις πρότερες εισόδους μιας ακολουθίας, αλλά καμία γνώση για τις μελλοντικές. Σε αρκετές εφαρμογές, τα αποτελέσματα βελτιώνονται σημαντικά αν υπάρχει γνώση τόσο του παρελθόντος όσο και τους μέλλοντος. Σε αυτές τις εφαρμογές, η πρόβλεψη ενός χρονικού βήματος μπορεί να εξαρτάται από ολόκληρη της ακολουθία.

Τα αναδρομικά νευρωνικά δίκτυα διπλής κατεύθυνσης (Bidirectional RNN) σχεδιάστηκαν για να καλύψουν αυτήν την ανάγκη. Εμφανίζουν αξιοσημείωτο βαθμό επιτυχίας σε εφαρμογές οπτικής αναγνώρισης χαρακτήρων (optical character recognition, OCR) και αναγνώρισης φωνής. Όπως φανερώνει και το όνομα, τα RNN διπλής κατεύθυνσης συνδυάζουν ένα RNN το οποίο «κινείται» μπροστά στο χρόνο ξεκινώντας από την αρχή της ακολουθίας, και ένα άλλο RNN το οποίο «κινείται» πίσω στο χρόνο ξεκινώντας από το τέλος της ακολουθίας. Αυτό το μοντέλο παρουσιάζεται σχηματικά στην Εικόνα 3.7

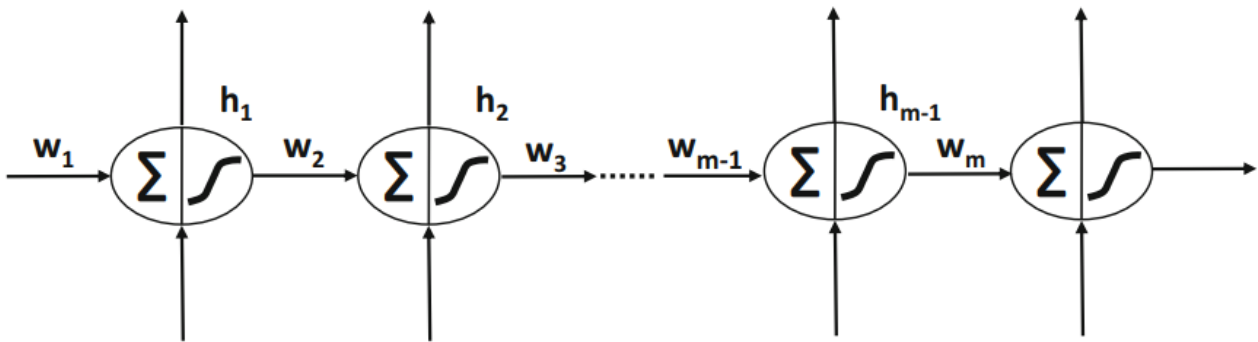


Εικόνα 3.7 Αναδρομικό νευρωνικό δίκτυο διπλής κατεύθυνσης (Bidirectional RNN)

3.9 Προβλήματα εξαφάνισης και έκρηξης κλίσης και δίκτυα LSTM και GRU

Τα αναδρομικά νευρωνικά δίκτυα είναι δύσκολο να εκπαιδευτούν εξαιτίας του γεγονότος ότι η είσοδος της διάστασης του χρόνου στην αρχιτεκτονική του δικτύου, οδηγεί σε πολύ «βαθιά» νευρωνικά δίκτυα (δηλαδή δίκτυα με πολλές ενδιάμεσες συνάψεις μεταξύ εισόδου και εξόδου), ειδικά αν η ακολουθία εισόδου έχει πολύ μεγάλο μήκος. Αυτό οδηγεί σε έξαρση των προβλημάτων της έκρηξης κλίσης (exploding gradient) και εξαφάνισης κλίσης (vanishing gradient). Αυτά τα προβλήματα εμφανίζονται και στα δίκτυα πρόσθιας τροφοδότησης αλλά είναι πολύ πιο έντονα στα αναδρομικά δίκτυα.

Έστω ότι υπάρχει ένα αναδρομικό δίκτυο με ένα μόνο επίπεδο και με έναν μόνο νευρώνα και ένα μήνυμα εισόδου με T χρονικά βήματα. Η συνάρτηση ενεργοποίησης, που χρησιμοποιείται στους νευρώνες, είναι η $\varphi(x) = \tanh x$. Η αρχιτεκτονική του φαίνεται στην Εικόνα 3.8



Εικόνα 3.8 Αρχιτεκτονική αναδρομικού δικτύου

Έστω πως h_1, \dots, h_T οι κρυφές τιμές (κατάστασεις) στα διάφορα επίπεδα και L η συνάρτηση απωλειών, η οποία χρησιμοποιείται για την εκπαίδευση του μοντέλου. Με χρήση του κανόνα της αλυσίδας του Leibniz, του οποίου γίνεται εκμετάλλευση στην τεχνική της οπισθοδιάδοσης, προκύπτει πως:

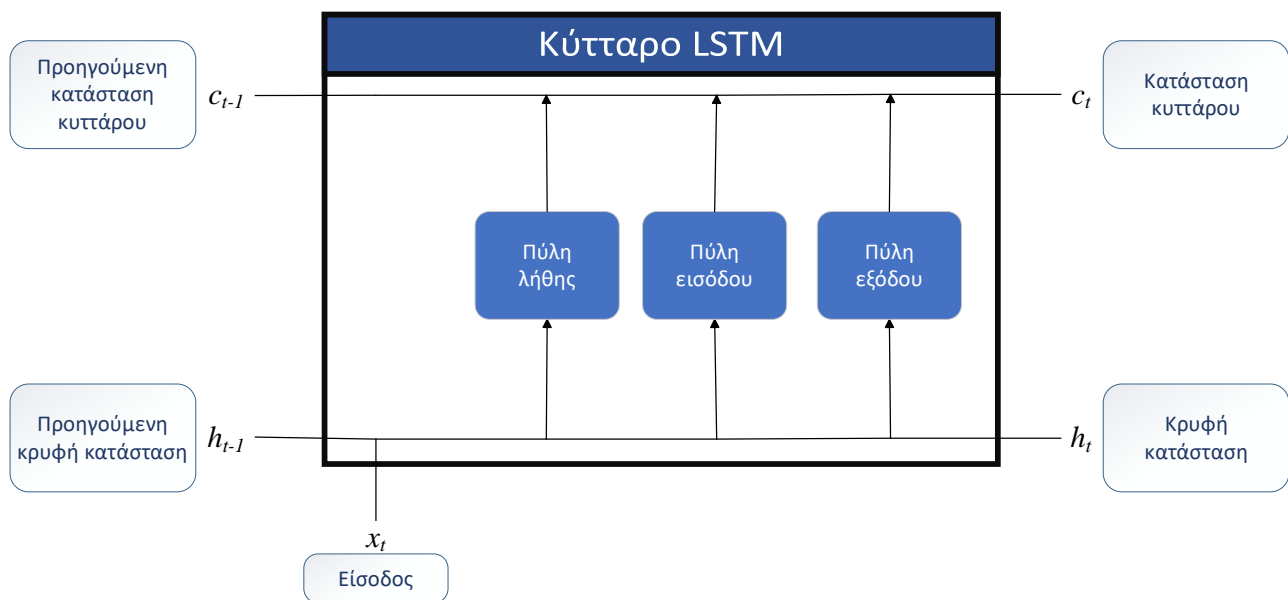
$$\frac{\partial L}{\partial h_t} = \varphi'(h_{t+1}) \cdot w_{t+1} \frac{\partial L}{\partial h_{t+1}} \quad (3.2)$$

Από την στιγμή που τα βάρη μεταξύ διαφορετικών χρονικά επιπέδων είναι ίδια, ισχύει πως $w_{t+1} = w$. Όταν ισχύει πως $w > 1$, αυτός ο πολλαπλασιασμός θα έχει μια τάση συνεχούς αύξησης και θα ισχύει πως $\frac{\partial L}{\partial h_t} \rightarrow \infty$ καθώς εκτελείται ο αλγόριθμος της οπισθοδιάδοσης, ενώ όταν $w < 1$, θα έχει μια τάση συνεχούς μείωσης και θα ισχύει πως $\frac{\partial L}{\partial h_t} \rightarrow 0$. Η δεύτερη περίπτωση εμφανίζεται πιο συχνά στα RNN και προκαλεί το πρόβλημα της εξαφάνισης κλίσης.

Το πρόβλημα της εξαφάνισης κλίσης οδηγεί στην απώλεια πληροφορίας για μεγάλες ακολουθίες εισόδου, καθώς η πληροφορία τους αποθηκεύεται σε αυτές τις κρυφές καταστάσεις h_t .

Για την αντιμετώπιση αυτών των προβλημάτων, έχουν αναπτυχθεί πιο περίπλοκοι τύποι αναδρομικών νευρωνικών δικτύων όπως το κύτταρο LSTM (Long Short-Term Memory) και το κύτταρο GRU (Gated Recurrent Unit). Ανήκουν σε μια γενικότερη κατηγορία αναδρομικών δικτύων που ονομάζονται RNNs με πύλες (gated RNNs). Χρησιμοποιούν ειδικές δομές, οι οποίες ονομάζονται πύλες, οι οποίες συγκρατούν την πληροφορία στην μνήμη για όσο χρονικό διάστημα χρειαστεί.

Το σχηματικό διάγραμμα ενός τυπικού κυττάρου LSTM απεικονίζεται στην Εικόνα 3.9.



Εικόνα 3.9 Κύτταρο LSTM

Σε ένα απλό κύτταρο RNN, η κρυφή κατάσταση χρησιμοποιείται για δύο σκοπούς: την αποθήκευση πληροφορίας και την τέλεση προβλέψεων. Σε ένα κύτταρο LSTM, αυτές οι δύο λειτουργίες διαχωρίζονται :

- Η κατάσταση κυττάρου ή εσωτερική μνήμη είναι η τιμή στην οποία αποθηκεύονται όλες οι απαραίτητες πληροφορίες
- Η κρυφή κατάσταση χρησιμοποιείται για τον υπολογισμό της εξόδου, δηλαδή για την τέλεση προβλέψεων

Στην συνέχεια περιγράφεται συνοπτικά η λειτουργία κάθε πύλης:

- Η πύλη λήθης είναι υπεύθυνη για το ποια πληροφορία θεωρείται ασήμαντη και πρέπει να αφαιρεθεί από τη κατάσταση κυττάρου (μνήμη).
- Η πύλη εισόδου είναι υπεύθυνη για το ποια πληροφορία πρέπει να αποθηκευτεί στη κατάσταση κυττάρου.
- Η πύλη εξόδου αποφασίζει για το ποια πληροφορία πρέπει να παρθεί από την κατάσταση κυττάρου και να δοθεί ως έξοδος.

Τα κύτταρα LSTM καταπολεμούν αρκετά αποτελεσματικά το πρόβλημα της εξαφάνισης κλίσης. Όμως αυτή βελτίωση στις επιδόσεις έρχεται με ένα κόστος στην υπολογιστική πολυπλοκότητα του δικτύου. Το κύτταρο LSTM έχει πάρα πολλές ελεύθερες παραμέτρους λόγω της ύπαρξης πολλών πυλών και καταστάσεων. Αυτό οδηγεί σε αύξηση του χρόνου εκπαίδευσης του νευρωνικού δικτύου.

Το GRU είναι πρακτικά μια απλοποιημένη μορφή του κυττάρου LSTM. Η απλούστευση αυτή επιτυγχάνεται με τη χρήση μιας πύλης για την ταυτόχρονη τέλεση των λειτουργιών των πυλών λήθης και εξόδου του κυττάρου LSTM. Ακόμη, το κύτταρο GRU δεν χρησιμοποιεί καταστάσεις κυττάρου. Παρόλα αυτά, η βασική ιδέα στο GRU είναι παρόμοια με αυτή στο LSTM όσον αφορά την ανανέωση των κρυφών καταστάσεων h_t .

4 Εργαλεία ανάπτυξης λογισμικού

Σε αυτό το μικρό κεφάλαιο αναφέρονται μερικές πληροφορίες για τα εργαλεία που χρησιμοποιήθηκαν, για την υλοποίηση νευρωνικών δικτύων που χρησιμοποιούνται στη αποκωδικοποίηση σημάτων που έχουν κωδικοποιηθεί με χρήση συνελκτικού κώδικα.

4.1 Γλώσσα Python

Η Python είναι μια διερμηνευόμενη γενικού σκοπού και πολύ υψηλού επιπέδου γλώσσα προγραμματισμού. Υποστηρίζει τόσο τον διαδικαστικό όσο και το αντικειμενοστραφή προγραμματισμό.

Δημιουργήθηκε από τον Guido Van Rossum το 1989 και κυκλοφόρησε για πρώτη φορά το 1991. Κύριος στόχος της Python είναι η εύκολη χρήση της καθώς και η αναγνωσιμότητα του κώδικα της. Το συντακτικό της επιτρέπει στον προγραμματιστή να εκφράσει έννοιες σε λιγότερες γραμμές κώδικα από ότι σε γλώσσες όπως η C++ και η Java. Περιέχει πολύ μεγάλο βιβλιοθηκών που διευκολύνουν την εκτέλεση εργασιών ρουτίνας. Μειονεκτεί στο γεγονός ότι είναι πιο αργή κατά την εκτέλεση της από γλώσσες όπως η C και η C++.

Η Python αναπτύσσεται ως λογισμικό ανοιχτού κώδικα (open source) και η διαχείριση της γίνεται από το Python Software Foundation.

4.2 Η βιβλιοθήκη TensorFlow

Η TensorFlow είναι μια ελεύθερη ανοιχτού κώδικα βιβλιοθήκη για μηχανική μάθηση και τεχνητή νοημοσύνη. Μπορεί να χρησιμοποιηθεί για πληθώρα εργασιών, αλλά ειδικεύεται στην δημιουργία, εκπαίδευση και αξιολόγηση νευρωνικών δικτύων.

Η βιβλιοθήκη TensorFlow αναπτύχθηκε από την ομάδα Google Brain και η πρώτη της έκδοση κυκλοφόρησε το 2015. Η Google κυκλοφόρησε μια ενημερωμένη έκδοση της TensorFlow, την TensorFlow 2.0, το 2019. Μπορεί να χρησιμοποιηθεί με μια πληθώρα γλωσσών προγραμματισμού, όπως η Python, η Java και η C++.

Η TensorFlow παρέχει πληθώρα εργαλείων στον χρήστη για την σχεδίαση νευρωνικών δικτύων. Μερικά από αυτά είναι:

- Συναρτήσεις απωλειών (losses): Η βιβλιοθήκη παρέχει πληθώρα έτοιμων προς χρήση συναρτήσεων απωλειών για την εκπαίδευση και την αξιολόγηση μοντέλων νευρωνικών δικτύων.

- Μετρικές (metrics): Για την αξιολόγηση των επιδόσεων των δομών μηχανικής μάθησης, η βιβλιοθήκη παρέχει διάφορες έτοιμες μετρικές.
- Σχεδίαση αρχιτεκτονικής νευρωνικών δικτύων: Η βιβλιοθήκη παρέχει εργαλεία για εύκολη σχεδίαση και εκπαίδευση πολλών δημοφιλών αρχιτεκτονικών νευρωνικών δικτύων. Ακόμη, παρέχει έτοιμες πολλές δημοφιλείς συναρτήσεις ενεργοποίησης (activation functions) (βλέπε ενότητα 1.6).
- Βελτιστοποιητές (Optimizers): Η TensorFlow παρέχει ένα σύνολο βελτιστοποιητών για την εκπαίδευση των νευρωνικών δικτύων. Οι βελτιστοποιητές είναι οι αλγόριθμοι μηχανικής μάθησης που ελαχιστοποιούν την συνάρτηση κόστους.

4.3 Η πλατφόρμα Google Colaboratory

Η πλατφόρμα Google Colaboratory είναι μια online πλατφόρμα βασισμένη σε υπηρεσίες νέφους, η οποία επιτρέπει στον χρήστη να γράψει και να εκτελέσει κώδικα Python μέσα από το φυλλομετρητή του. Μερικά βασικά χαρακτηριστικά του είναι τα εξής:

- Το Colab είναι φτιαγμένο με βάση τον κώδικα του Project Jupyter και φιλοξενεί Jupyter Notebooks χωρίς να απαιτείται καμία εγκατάσταση λογισμικού στην συσκευή.
- Παρέχει δωρεάν πρόσβαση σε επεξεργαστικούς πόρους. Πιο συγκεκριμένα, παρέχει δωρεάν πρόσβαση σε μονάδες CPU, GPU και TPU, οι οποίες μπορούν να χρησιμοποιηθούν για την εκτέλεση του κώδικα.
- Έχει πολλές προεγκατεστημένες βιβλιοθήκες, οι οποίες χρησιμοποιούνται ευρέως στους τομείς της μηχανικής μάθησης και της ανάλυσης δεδομένων, όπως οι TensorFlow, PyTorch και Keras.

```

x_array=tokenize_seq(x_array, level)
start_time=time()
model.predict(x_array)
end_time=time()
nn_time[i]=end-start
if save:
    np.save("nn_time.npy", nn_time)
plt.rcParams['figure.figsize'] = [15, 8]
plt.rcParams['font.size'] = 14
plt.clf()
plt.plot(batch_size, nn_time, label=model.name, linewidth=1.5)
plt.legend(loc='lower right')
plt.xlabel('Batch Size')
plt.ylabel('Time (sec)')
plt.yscale('log')
plt.show()
return batch_size,nn_time

batch,mlp_time_cpu=NeuralTime(mlp,seq_length,0,3,level=0,save=False)
np.save("mlp_time.npy", nn_time)
batch,nn_time=NeuralTime(rnn,seq_length,0,3,level=0,save=False)
np.save("rnn_time.npy", nn_time)
batch,nn_time=NeuralTime(BiRNN,seq_length,0,3,level=0,save=False)
np.save("BiRNN_time.npy", nn_time)

Batch size is 1
1/1 [=====] - 0s 15ms/step
Batch size is 2
1/1 [=====] - 0s 14ms/step
Batch size is 3
1/1 [=====] - 0s 16ms/step
Batch size is 4
1/1 [=====] - 0s 20ms/step

```

Εικόνα 4.1 Η πλατφόρμα Google Colaboratory

5 Ανάπτυξη νευρωνικού δικτύου για την αποκωδικοποίηση σημάτων συνελκτικού κώδικα

5.1 Προσομοίωση τηλεπικοινωνιακού συστήματος

Το πρώτο βήμα, το οποίο πρέπει να γίνει, προτού προχωρήσουμε στην διαδικασία σχεδίασης του νευρωνικού δικτύου, είναι η συγγραφή κώδικα που θα προσομοιώνει την λειτουργία ενός πλήρους ψηφιακού τηλεπικοινωνιακού συστήματος, το οποίο κάνει χρήση συνελκτικού κώδικα. Ακολουθεί μια σύντομη περιγραφή των συναρτήσεων του κώδικα που αναπτύχθηκε.

Την προσομοίωση της λειτουργίας του ψηφιακού πομπού αναλαμβάνουν οι εξής συναρτήσεις:

- **encoder**: προσομοιώνει την λειτουργία του κωδικοποιητή της Εικόνας 2.1. Η συνάρτηση αυτή δέχεται ως όρισμα μια ακολουθία bit και επιστρέφει την κωδικοποιημένη ακολουθία. Υπάρχει η επιλογή για καθαρισμό ή μη της μνήμης του κωδικοποιητή. Χρησιμοποιεί την κλάση `conv_encoder`, η οποία προσομοιώνει τον συνελκτικό κωδικοποιητή της Εικόνας 2.1.
- **modulator**: προσομοιώνει την λειτουργία ενός διαμορφωτή που επιτελεί αντιποδική διαμόρφωση. Δέχεται ως όρισμα μια ακολουθία bit και επιστρέφει μια ακολουθία. Η επιστρεφόμενη ακολουθία έχει την τιμή «1» στις θέσεις που η ακολουθία εισόδου είχε ψηφίο «1» και την τιμή «-1» στις θέσεις που η ακολουθία εισόδου είχε την τιμή «0».

Την προσομοίωση της λειτουργία του καναλιού αναλαμβάνει η παρακάτω συνάρτηση:

- **awgn_channel**: προσομοιώνει την λειτουργία ενός καναλιού που εισάγει λευκό γκαουσιανό θόρυβο στην αποστελλόμενη κυματομορφή. Δίνει την δυνατότητα στον χρήστη να επιλέγει το SNR που θέλει να έχει στο ληφθέν σήμα

Την προσομοίωση του ψηφιακού δέκτη επιτελούν οι εξής συναρτήσεις:

- **demodulator**: προσομοιώνει την λειτουργία του αποδιαμορφωτή. Δίνεται η δυνατότητα στον χρήστη να επιλέγει τα επίπεδα κβάντισης του αποδιαμορφωτή.
- **viterbi**: προσομοιώνει την λειτουργία ενός αποκωδικοποιητή που κάνει χρήση του αλγορίθμου Viterbi (Ενότητα 2.5) για τον (2,1,3) συνελκτικό κωδικοποιητή της Εικόνας 2.1. Βεβαίως, ενημερώνεται μέσω ορίσματος για τα επίπεδα κβάντισης που χρησιμοποιεί ο demodulator για να μπορεί να επιτελεί σωστά την αποκωδικοποίηση Viterbi.

Για την συνάρτηση “viterbi” θα δοθούν περισσότερες λεπτομέρειες, καθώς είναι σαφώς το πιο πολύπλοκο τμήμα και βεβαίως το σημείο ενδιαφέροντος, καθώς αντικείμενο μελέτης αυτής της εργασίας είναι η αποκωδικοποίηση σημάτων:

- Ορίζεται η κλάση `node`, η οποία αντιπροσωπεύει κάθε κόμβο κατάστασης εντός του διαγράμματος Trellis. Γνωρίσματα (attributes) αυτής της κλάσης είναι:
 - `tim`: το χρονικό βήμα
 - `distance`: το τετράγωνο της Ευκλείδειας απόστασης από το ληφθέν σήμα
 - `m1`: το πρώτο ψηφίο της κατάστασης
 - `m2`: το δεύτερο ψηφίο της κατάστασης
 - `path`: η διαδρομή που έχει ακολουθηθεί ώστε να φθάσουμε σε αυτόν τον κόμβο
- Μέθοδοι (methods) της κλάσης `node` είναι οι εξής:
 - `__init__`: ο κατασκευαστής (constructor). Καλείται κάθε φορά που δημιουργείται ένα αντικείμενο αυτής της κλάσης και αρχικοποιεί τα γνωρίσματα του νέου αντικειμένου
 - `__lt__`: καθορίζει τη λειτουργία του τελεστή «<» όταν χρησιμοποιείται για την σύγκριση δύο αντικειμένων αυτής της κλάσης. Με βάση των κώδικα μας, επιστρέφει «True» ή «False» ανάλογα με το αποτέλεσμα της σύγκρισης των γνωρισμάτων `distance` των δύο αντικειμένων κλάσης `node`
 - `type`: όταν καλείται τυπώνει τα γνωρίσματα του αντικειμένου
 - `ins`: μέσω αυτής της μεθόδου υλοποιείται μια διαδικασία μετάβασης εντός του Trellis. Παίρνει ως ορίσματα το `bit` που προκαλεί τη μετάβαση καθώς και το αντίστοιχο τμήμα της ληφθείσας ακολουθίας. Κάνει χρήση και αντικειμένων της κλάσης `conv_encoder` για να υπολογίσει τα ψηφία εξόδου του κλάδου (αυτό σημαίνει πως ο αποκωδικοποιητής έχει **εγγενή γνώση** της δομής του κωδικοποιητή). Επιστρέφει ένα νέο αντικείμενο της κλάσης `node`, το οποίο έχει ενημερώσει τα γνωρίσματα του βάσει της μετάβασης που συνέβη.
- Η συνάρτηση δέχεται δύο ορίσματα: την ληφθείσα ακολουθία και τα επίπεδα κβάντισης του αποδιαμορφωτή
- Ορίζει δύο μονοδιάστατους πίνακες αντικειμένων κλάσης `node` τεσσάρων θέσεων, τον `old_states` και τον `new_states`. Οι τέσσερις θέσεις του κάθε πίνακα αντιστοιχούν στις τέσσερις πιθανές καταστάσεις. Ο πίνακας `old_states` έχει αποθηκευμένους τους κόμβους του τρέχοντος χρονικού βήματος, ενώ ο σκοπός του πίνακα `new_states` είναι να αποθηκεύσει του κόμβους του επόμενου χρονικού βήματος αφότου καθοριστούν. Στη θέση

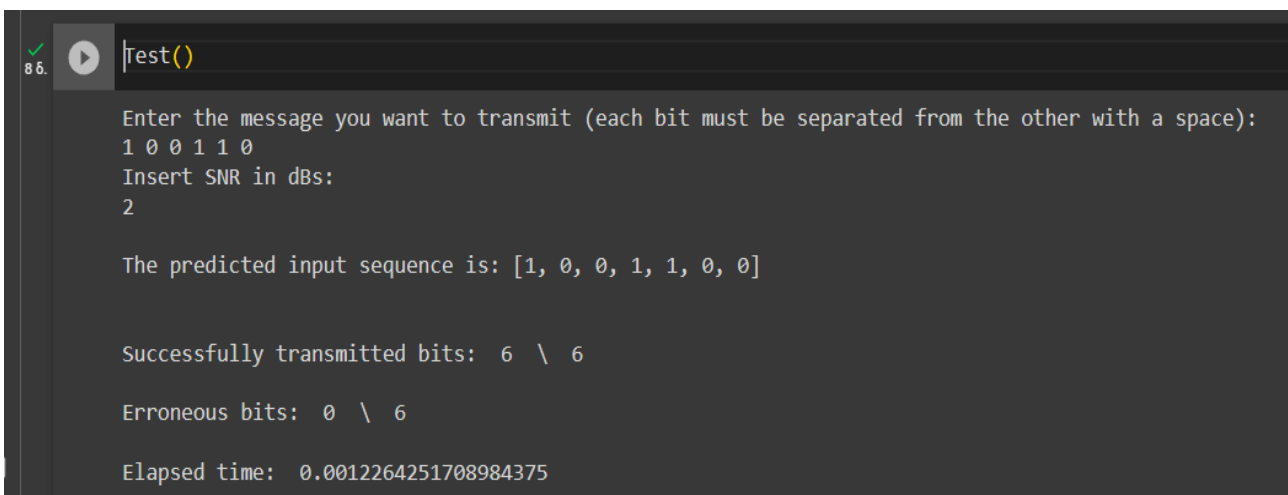
‘0’ του πίνακα `old_states` τοποθετείται το αντικείμενο ‘`initial_state`’, το οποίο αποτελεί τον αρχικό κόμβο (all zero node) του διαγράμματος Trellis. Οι υπόλοιπες θέσεις των πινάκων είναι επί της ουσίας κενές (χρησιμοποιείται ως προσωρινή τιμή το ‘`false_state`’ μέχρι να αντικατασταθεί από κάποια έγκυρη τιμή)

- Εκτελείται ένας βρόχος επανάληψης `for`. Ο αριθμός των επαναλήψεων είναι ίσος με το μισό του πλήθους των ψηφίων/συμβόλων στο ληφθέν σήμα και σε κάθε επανάληψη γίνεται επεξεργασία 2 ψηφίων της ληφθείσας ακολουθίας. Αυτό συμβαίνει καθώς είναι γνωστό πως ο συνελκτικός κωδικοποιητής που χρησιμοποιείται εξάγει δύο κωδικοποιημένα bit για κάθε bit πληροφορίας που επεξεργάζεται.
 - Μέσα σε αυτόν τον βρόχο `for` υπάρχει ένας εμφωλευμένος βρόχος `for`, ο οποίος επαναλαμβάνεται για κάθε κόμβο κατάστασης στην στον πίνακα `old_states`.
 - Εντός και αυτού του βρόχου, βρίσκεται ένας εμφωλευμένος βρόχος `for` ο οποίος καλεί για κάθε κόμβο του `old_states` την μέθοδο `ins` για τα δύο πιθανά bit εισόδου, το ‘0’ και το ‘1’. Για κάθε νέο κόμβο, ο οποίος προκύπτει ως επιστροφή της μεθόδου `ins`, λαμβάνεται απόφαση για το αν ή όχι θα αποθηκευτεί αυτός ο στον πίνακα `new_states`. Η απόφαση λαμβάνεται με το εξής κριτήριο: αν η θέση του πίνακα `new_states`, στην οποία μπορεί να τοποθετηθεί ο κόμβος, είναι κενή ή αν καταλαμβάνεται από έναν κόμβο με μεγαλύτερη Ευκλείδεια απόσταση (`distance`), τότε ο κόμβος αποθηκεύεται.
 - Αφού εκτελεστούν οι δύο εμφωλευμένοι βρόχοι, ανανεώνονται οι πίνακες `new_states` και `old_states`. Στον πίνακα `old_states` εκχωρούνται οι τιμές του πίνακα `new_states` και ο πίνακας `new_states` «αδειάζεται». Έτσι, γίνεται η προετοιμασία για την επόμενη επανάληψη του εξωτερικού βρόχου `for` (επόμενο χρονικό βήμα).
- Αφότου εκτελεστεί και ο εξωτερικός βρόχος `for`, επιλέγεται ως λύση ο κόμβος του πίνακα `old_states` με την μικρότερη Ευκλείδεια απόσταση. Από την συνάρτηση επιστρέφεται, η διαδρομή (`path`) του επιλεχθέντος κόμβου.
- Όπως αναφέρθηκε και στην Ενότητα 2.5, όταν προκύπτουν δύο ή περισσότερες υποδιαδρομές με ίδια μετρική Viterbi, είναι ισοπίθανες. Όποια και αν επιλεχθεί, είναι εξίσου σωστή. Παρά ταύτα, με βάση τον αλγόριθμο που έχει υλοποιηθεί, δίνεται μια σειρά προτεραιότητας ανάλογα με την κατάσταση από την οποία έχει προέλθει η υποδιαδρομή. Η σειρά προτεραιότητας είναι: (00), (01), (10), (11).

Η γενική υπολογιστική πολυπλοκότητα ενός αλγορίθμου Viterbi είναι ίση με $O(N \cdot M^2)$, όπου N το μήκος της ληφθείσας ακολουθίας και M ο αριθμός των καταστάσεων. Σε αυτήν την υλοποίηση λόγω

γνώσης του συστήματος κωδικοποίησης, ο αριθμός των καταστάσεων είναι σταθερός και ίσος με 4 και η επεξεργασία της ληφθείσας ακολουθίας γίνεται ανά δυάδα ψηφίων, οπότε η πολυπλοκότητα γίνεται ίση με $O(N/2)$.

«Συνενώνοντας» όλες αυτές τις λειτουργίες, ο χρήστης έχει στην διάθεση του ένα πλήρες τηλεπικοινωνιακό σύστημα. Στην παρακάτω εικόνα φαίνεται η έξοδος ενός προγράμματος, το οποίο κάνει χρήση αυτών των συναρτήσεων και αφού ζητήσει από τον χρήστη να εισάγει μια ακολουθία bit και το επιθυμητό SNR, προσομοιώνει την αποστολή ενός τέτοιου μηνύματος μέσα από το σύστημα που περιεγράφηκε. Για τον demodulator και τον viterbi έχουν επιλεγεί 8 επίπεδα κβάντισης (μαλακός αποκωδικοποιητής Viterbi).



```
86. |test()
Enter the message you want to transmit (each bit must be separated from the other with a space):
1 0 0 1 1 0
Insert SNR in dBs:
2

The predicted input sequence is: [1, 0, 0, 1, 1, 0, 0]

Successfully transmitted bits: 6 \ 6

Erroneous bits: 0 \ 6

Elapsed time: 0.0012264251708984375
```

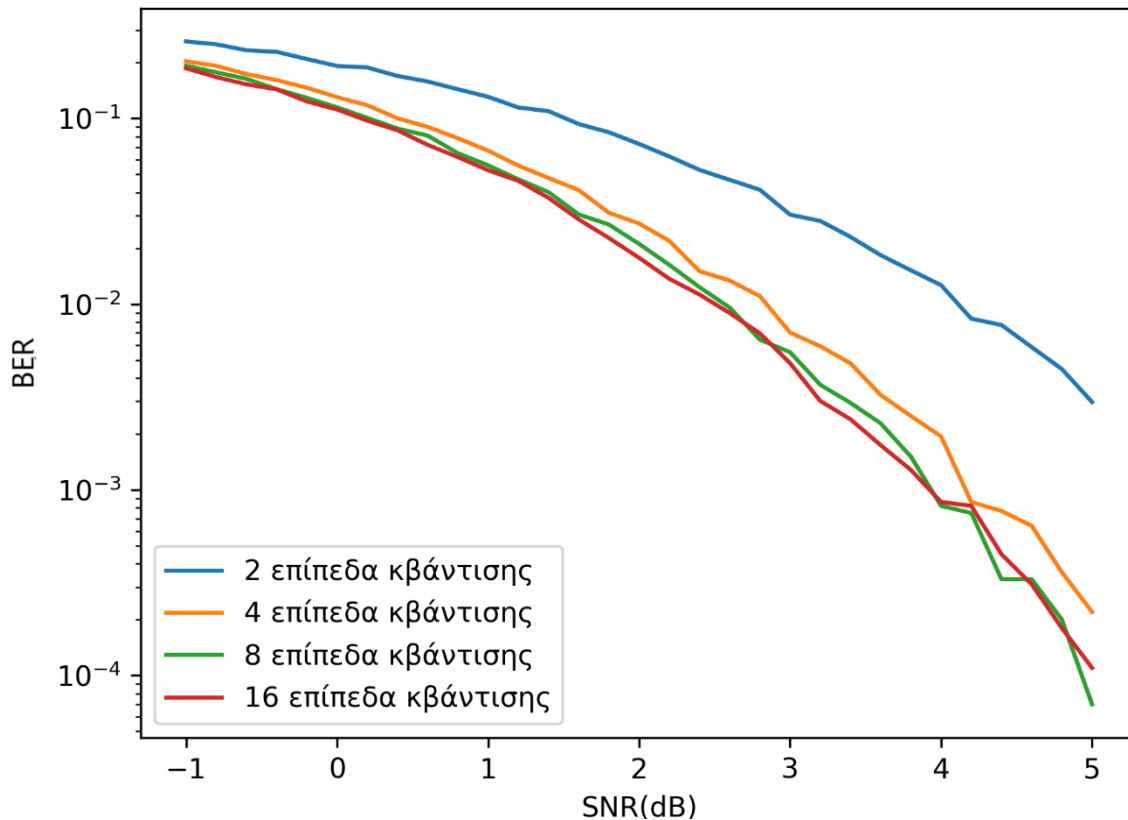
Εικόνα 5.1 Έξοδος προγράμματος προσομοίωσης ψηφιακού τηλεπικοινωνιακού συστήματος

5.2 Επιλογή διαδικασίας μάθησης

Όσον αφορά την διαδικασία μάθησης, μια δόκιμη επιλογή φαίνεται να είναι η **επιβλεπόμενη μάθηση** (Ενότητα 3.3.1). Ως εκπαιδευτής θα χρησιμοποιηθεί ο αποκωδικοποιητής Viterbi, ο οποίος υλοποιείται μέσω της συνάρτησης «viterbi». Μια τέτοια επιλογή θα ήταν ιδανική, γιατί ο αλγόριθμος Viterbi είναι αποτελεσματικός στην αποκωδικοποίηση συνελκτικών κωδίκων. Ακόμη η υλοποίηση, που έχει πραγματοποιηθεί, έχει εγγενή γνώση του συστήματος κωδικοποίησης που χρησιμοποιείται, όπως αναφέρθηκε στην προηγούμενη ενότητα. Με την χρήση της συνάρτησης 'viterbi' θα παραχθούν τα απαραίτητα σύνολα δεδομένων που χρειάζονται για την εκπαίδευση και την αξιολόγηση των επιδόσεων των νευρωνικών δικτύων που θα υλοποιηθούν.

Ένα ερώτημα, το οποίο μένει να απαντηθεί, είναι ο αριθμός των επιπέδων κβάντισης του αποκωδικοποιητή Viterbi, ο οποίος θα χρησιμοποιηθεί ως εκπαιδευτής. Στην Εικόνα 5.2 φαίνεται το γράφημα του ρυθμού σφάλματος ανά ψηφίο (Bit Error Rate, BER) συναρτήσει του σηματοθορυβικού λόγου SNR για διάφορα επίπεδα κβάντισης. Τα δεδομένα πάρθηκαν από

πολλαπλές εκτελέσεις του κώδικα, που σχεδιάστηκε, για διάφορες τιμές του SNR. Το γράφημα υλοποιήθηκε με την βοήθεια της βιβλιοθήκης matplotlib της Python.



Εικόνα 5.2 BER για διάφορα επίπεδα κβάντισης του διαμορφωτή και του Viterbi

Όπως ήταν αναμενόμενο, ο Viterbi σκληρών αποφάσεων (2 επίπεδα κβάντισης) εμφανίζει χειρότερες επιδόσεις από τον Viterbi μαλακών αποφάσεων. Μάλιστα, όσο αυξάνεται ο αριθμός των επιπέδων κβάντισης, τόσο καλύτερες οι επιδόσεις ως προς το BER. Βέβαια, από ένα σημείο και έπειτα φαίνεται πως η απόσταση μεταξύ των επιδόσεων μειώνεται. Μεταξύ 8 και 16 επιπέδων κβάντισης, υπάρχει πολύ μικρή διαφορά όσον αφορά τις επιδόσεις. Όμως υπάρχει και μια σταδιακή αύξηση του υπολογιστικού φόρτου.

Για να υπάρχει μια ισορροπία μεταξύ υπολογιστικού φόρτου και ακρίβειας, ο **Viterbi μαλακών αποφάσεων με 8 επίπεδα κβάντισης** θα χρησιμοποιηθεί ως εκπαιδευτής, την λειτουργία του οποίου θα προσπαθούν να «μιμηθούν» τα εκπαιδευόμενα νευρωνικά δίκτυα.

5.3 Συλλογή και μορφοποίηση δεδομένων εκπαίδευσης, επαλήθευσης και ελέγχου

Μια απαραίτητη διαδικασία για την εκπαίδευση και την αξιολόγηση της λειτουργίας ενός νευρωνικού δικτύου είναι η συγκέντρωση δεδομένων και η σωστή αναπαράστασή τους.

Η συγκέντρωση δεδομένων θα γίνει με τη βοήθεια του κώδικα που αναπτύχθηκε. Η ακριβής διαδικασία είναι η εξής:

- Δημιουργία τυχαίων διακριτών ακολουθιών bits συγκεκριμένου μήκους.
- Εισαγωγή των τυχαίων ακολουθιών στην προσομοίωση τηλεπικοινωνιακού συστήματος που έχει υλοποιηθεί και επεξεργασία τους από της εξής μονάδες κατά σειρά: encoder, modulator, αγωγή_channel, demodulator. Οι ακολουθίες, οι οποίες θα προκύψουν, θα έχουν διπλάσιο μήκος από τις αρχικές, θα έχουν επηρεαστεί από τον θόρυβο που εισάγει το κανάλι και θα χρησιμοποιηθούν για να αντληθούν τα διανύσματα εισόδου.
- Αυτές οι ακολουθίες στη συνέχεια θα εισαχθούν στον Viterbi αποκωδικοποιητή. Οι αποκρίσεις του Viterbi αποκωδικοποιητή θα αποτελέσουν τις επιθυμητές αποκρίσεις με την βοήθεια των οποίων θα υπολογίζεται το σφάλμα της εξόδου του νευρωνικού.
- Τα σύνολο των δεδομένων που έχουν συλλεχθεί, θα τεμαχίζεται σε τρία τμήματα
 - 60% σύνολο εκπαίδευσης (train set)
 - 20% σύνολο επαλήθευσης (validation set)
 - 20% σύνολο ελέγχου (test set)

Αφότου γίνει συλλογή των δεδομένων, ένα σημαντικό θέμα είναι το πως θα μορφοποιηθούν τα δεδομένα εισόδου, τα οποία θα εισαχθούν στο νευρωνικό δίκτυο. Σκοπός είναι να μορφοποιηθούν με τέτοιο τρόπο, ο οποίος να βοηθά το νευρωνικό στην διαδικασία μάθησης.

Έστω πως οι τυχαίες ακολουθίες είναι μήκους N και ο διαμορφωτής έχει 8 επίπεδα κβάντισης. Τρεις δημοφιλείς τρόποι μορφοποίησης της ακολουθίας εισόδου, οι οποίοι θα εξεταστούν στα πλαίσια της συγκεκριμένης εργασίας, είναι οι εξής

- **Tokenizing και One hot encoding:** Σε κάθε χρονικό βήμα στην ληφθείσα ακολουθία προστίθενται 2 ψηφία που κυμαίνονται από 0 έως 7. Αυτό οδηγεί σε $8^2=64$ πιθανούς διαφορετικούς συνδυασμούς. Η πληροφορία εισόδου καταγράφεται σε έναν πίνακα, ο οποίος περιέχει N διανύσματα 64 διαστάσεων. Αυτά τα διανύσματα έχουν 63 μηδενικά και 1 μονάδα. Η θέση της μονάδας εξαρτάται από το ποιος ήταν ο συνδυασμός των ψηφίων της ληφθείσας ακολουθίας. Αυτή η μέθοδος αναπαράστασης δεδομένων είναι συνηθισμένη σε μοντέλα επεξεργασίας φυσικής γλώσσας (NLP)
- **Δυαδική αναπαράσταση (Binary Representation):** Τα δύο ψηφία μετατρέπονται στην δυαδική τους μορφή και συνενώνονται. Για παράδειγμα, η δυάδα ψηφίων (3,4) μετατρέπεται στην δυαδική ακολουθία (011100). Αυτό σημαίνει πως το «διάνυσμα εισόδου» σε αυτήν περίπτωση είναι ένα πίνακας διαστάσεων $N \times 6$

- **Κανονικοποίηση (Normalization):** Τα δύο ψηφία κανονικοποιούνται έτσι ώστε η μέγιστη τιμή που εμφανίζεται στο σύνολο δεδομένων να είναι 1. Στην συγκεκριμένη περίπτωση, διαιρούνται με το 7. Το διάνυσμα εισόδου είναι ένας πίνακας διαστάσεων $N \times 2$

Όσον αφορά το διάνυσμα εξόδου, αυτό θα είναι ένα διάνυσμα $N \times 1$, το οποίο εκφράζει η πρόβλεψη του νευρωνικού δικτύου για το ποια ήταν η ακολουθία πληροφορίας που στάλθηκε.

5.4 Αρχιτεκτονικές νευρωνικών δικτύων

Στα πλαίσια της παρούσης εργασίας, θα εξεταστεί η αποδοτικότητα διαφόρων αρχιτεκτονικών νευρωνικών δικτύων για την επίλυση του προβλήματος της αποκωδικοποίησης σημάτων, τα οποία έχουν κωδικοποιηθεί από συνελκτικικούς κωδικοποιητές.

Πιο συγκεκριμένα θα εκπαιδευτούν και θα αξιολογηθούν τα εξής δίκτυα:

- **Δίκτυο πρόσθιας τροφοδότησης (MLP)**
 - Επίπεδο εισόδου
 - 2 κρυφά επίπεδα με 128 και 64 νευρώνες αντίστοιχα
 - Μετά από κάθε κρυφό επίπεδο, τοποθετείται ένα επίπεδο Dropout. Αυτά τα επίπεδα θέτουν τυχαία κάποιες από τις τιμές εισόδου τους ίσες με το μηδέν με κάποια προκαθορισμένη πιθανότητα. Χρησιμοποιούνται συχνά στα MLP για την αποφυγή του φαινομένου της υπερπροσαρμογής.
 - Επίπεδο εξόδου
- **Αναδρομικό νευρωνικό δίκτυο (RNN)**
 - Επίπεδο εισόδου
 - 2 κρυφά επίπεδα GRU με 64 νευρώνες το κάθε ένα
 - Επίπεδο εξόδου
- **Αναδρομικό νευρωνικό δίκτυο διπλής κατεύθυνσης (Bidirectional RNN)**
 - Επίπεδο εισόδου
 - 2 αμφίδρομα κρυφά επίπεδα (στο σύνολο 4 δηλαδή) GRU με 64 νευρώνες το κάθε ένα
 - Επίπεδο εξόδου

Αξίζει να σημειωθεί πως στους νευρώνες των επιπέδων εξόδου όλων των νευρωνικών χρησιμοποιείται η **σιγμοειδής συνάρτηση** ως συνάρτηση ενεργοποίησης. Χρησιμοποιείται συχνά στο επίπεδο εξόδου των μοντέλων δυαδικής ταξινόμησης. Το πρόβλημα που επιλύουμε μπορεί να

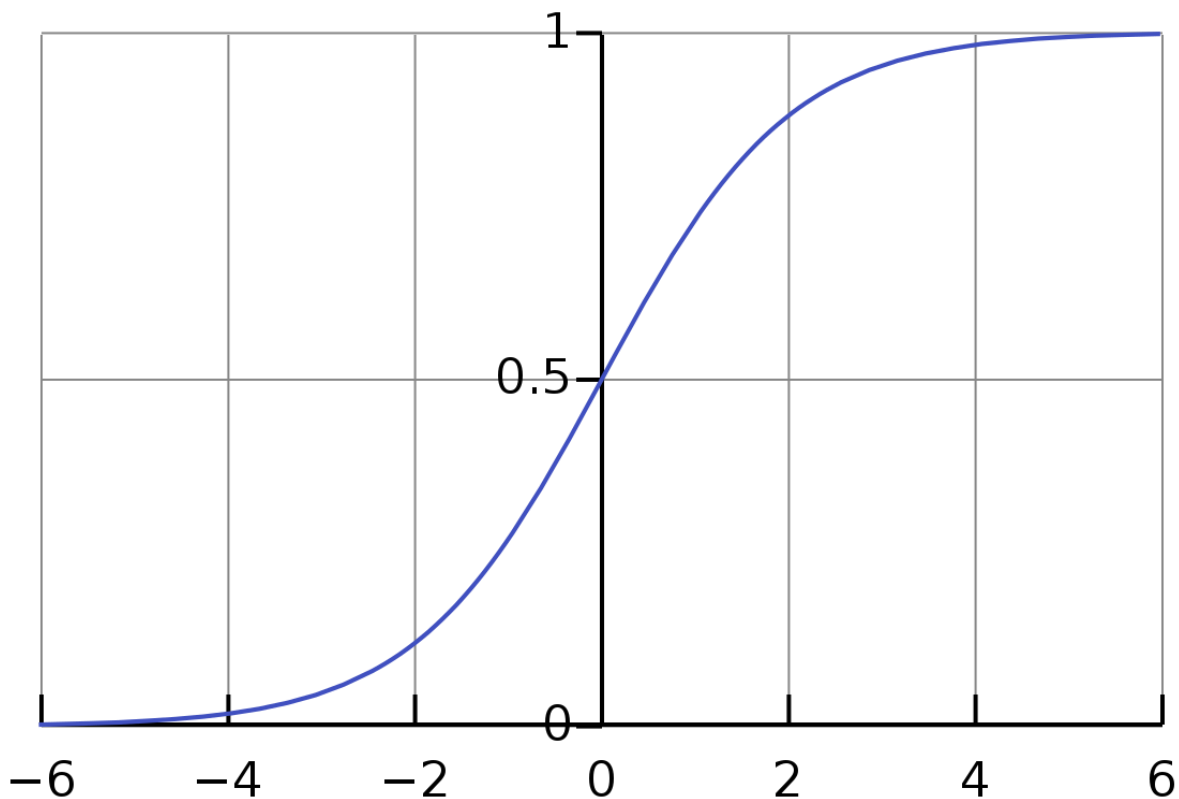
ερμηνευθεί ως ένα πρόβλημα δυαδικής ταξινόμησης καθώς η ακολουθία πληροφορίας αποτελείται από μια αλληλουχία ψηφίων, τα οποία μπορούν να είναι είτε '0', είτε '1'.

Η σιγμοειδής δίνεται από την σχέση:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (5.1)$$

και συνθλίβει την είσοδο της στο διάστημα $[0,1]$, όπως φαίνεται από το γράφημα της Εικόνας 5.3.

Η έξοδος της μπορεί να ερμηνευτεί ως η πιθανότητα η είσοδος να ανήκει στη θετική κλάση, το '1'.



Εικόνα 5.3 Η σιγμοειδής συνάρτηση

5.5 Συνάρτηση απωλειών, βελτιστοποιητής και μετρικές αξιολόγησης

Η συνάρτηση απωλειών που θα χρησιμοποιηθεί για την ποσοτικοποίηση των διαφορών μεταξύ των προβλέψεων του μοντέλου και των επιθυμητών αποκρίσεων είναι η **Binary Cross-Entropy (BCE)**. Χρησιμοποιείται συχνά για προβλήματα δυαδικής ταξινόμησης. Μετρά την απόδοση ενός μοντέλου ταξινόμησης, του οποίου η έξοδος είναι μια τιμή πιθανότητας μεταξύ 0 και 1.

Η σχέση με την οποία υπολογίζεται για ένα πλήθος N παρατηρήσεων είναι:

$$BCE = \frac{1}{N} \sum_{i=1}^N -(y_i \cdot \ln(p_i) + (1 - y_i) \cdot \ln(1 - p_i)) \quad (5.2)$$

όπου:

- y_i : είναι η επιθυμητή απόκριση
- p_i : είναι η προβλεφθείσα πιθανότητα το παράδειγμα να ανήκει στην κλάση '1'

Επί της ουσίας, αυτή σχέση τιμωρεί το μοντέλο πρόβλεψης όταν η προβλεφθείσα πιθανότητα αποκλίνει από την επιθυμητή απόκριση. Η βιβλιοθήκη TensorFlow περιλαμβάνει έτοιμη υλοποίηση της συνάρτησης BCE.

Ο αλγόριθμος βελτιστοποίησης, ο οποίος θα χρησιμοποιηθεί, είναι ο αλγόριθμος **Adaptive Moment Estimation (Adam)**. Είναι ένας δημοφιλής αλγόριθμος βελτιστοποίησης, ο οποίος χρησιμοποιείται για την εκπαίδευση πολλών νευρωνικών δικτύων. Βασίζεται σε μια παραλλαγή της μεθόδου καθόδου κλίσης, την στοχαστική κάθοδο κλίσης (stochastic gradient descent, SGD).

Μερικά πλεονεκτήματα του Adam είναι τα εξής:

- Προσαρμοζόμενοι ρυθμοί μάθησης: Έχει την ικανότητα να προσαρμόζει δυναμικά τους ρυθμούς μάθησης για κάθε παράμετρο του δικτύου.
- Αποτελεσματικότητα: Ο αλγόριθμος Adam συγκλίνει ταχύτερα από τους παραδοσιακούς αλγορίθμους μηχανικής μάθησης, ιδίως για μεγάλα σύνολα δεδομένων εκπαίδευσης και πολύπλοκα μοντέλα.
- Χαμηλή κατανάλωση μνήμης: Αυτό είναι ένα σημαντικό πλεονέκτημα, όταν τα μοντέλα μηχανικής μάθησης είναι πολύπλοκα και έχουν πολύ μεγάλο αριθμό ελεύθερων παραμέτρων.
- Ανεξαρτησία από την αρχικοποίηση υπερπαραμέτρων: Είναι λιγότερο ευαίσθητος στην αρχική τιμή που δίνει ο σχεδιαστής σε υπερπαραμέτρους, όπως ο ρυθμός μάθησης, συγκριτικά με άλλους αλγορίθμους βελτιστοποίησης
- Δημοφιλία και ευελιξία: Ο βελτιστοποιητής Adam είναι αρκετά διαδεδομένος στην προγραμματιστική κοινότητα καθώς είναι αποδοτικός για πληθώρα εργασιών και συνόλων δεδομένων.

Η βιβλιοθήκη TensorFlow παρέχει έτοιμη υλοποίηση του βελτιστοποιητή Adam.

Όσον αφορά την μετρική αξιολόγησης της λειτουργίας του δικτύου, θα χρησιμοποιηθεί η μετρική **accuracy**, σε πρώτο στάδιο, την οποία παρέχει έτοιμη η βιβλιοθήκη TensorFlow. Η μετρική accuracy ορίζεται ως εξής:

$$Accuracy = \frac{\text{Αριθμός σωστών προβλέψεων}}{\text{Αριθμός συνολικών προβλέψεων}} \quad (5.3)$$

Με άλλα λόγια, αυτή η μετρική εκφράζει το ποσοστό στο οποίο συμφωνούν οι προβλέψεις του νευρωνικού δικτύου με τις προβλέψεις του αποκωδικοποιητή Viterbi. Αξίζει να σημειωθεί πως ο αποκωδικοποιητής Viterbi δεν κάνει πάντα σωστή αποκωδικοποίηση παρότι βελτιστοποιεί την πιθανότητα εύρεσης της ακολουθίας, η οποία στάλθηκε. Ακόμη, κάποιες φορές υπάρχουν, παραπάνω από ένα βέλτιστα μονοπάτια εντός του διαγράμματος Trellis.

Η μετρική accuracy δεν είναι πάντα εντελώς αξιόπιστη. Ιδιαίτερα σε περιπτώσεις, όπου το σύνολο δεδομένων δεν είναι «ισορροπημένο», μπορεί να είναι παραπλανητική. Έστω πως σε ένα σύνολο δεδομένων το 90% των δεδομένων ανήκει στην κλάση '1' και μόλις το 10% ανήκει στην κλάση '0'. Αν ένα νευρωνικό δίκτυο κατατάσσει το 100% των περιπτώσεων που καλείται να μελετήσει στην κλάση '1', τότε θα εμφανίζει 0,9 ή 90% accuracy. Όπως καταλαβαίνει κανείς, όμως, το δίκτυο δεν θα έχει εκπαιδευθεί σωστά καθώς δεν θα αξιοποιεί σωστά τα δεδομένα της εισόδου. Αντιθέτως θα βασίζει την απόφαση του στην πιθανότητα εμφάνισης των κλάσεων στην έξοδο. Στην περίπτωση που εξετάζεται, κάτι τέτοιο δεν συμβαίνει, καθώς τα ενδεχόμενα αποστολής ψηφίου «1» ή ψηφίου «0» είναι ισοπίθανα.

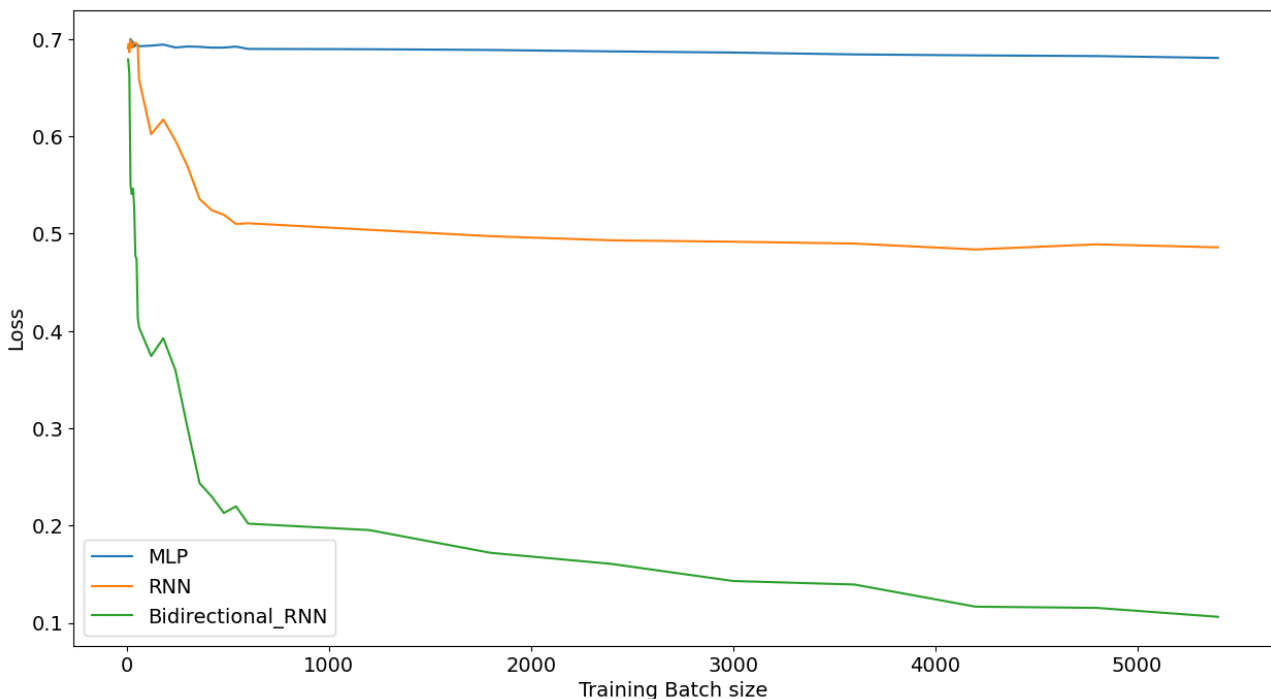
Μια ακόμα πιο σημαντική και αξιόπιστη μετρική αξιολόγησης, η οποία θα χρησιμοποιηθεί, είναι ο **ρυθμός σφάλματος ανά ψηφίο (BER)**, ο οποίος εμφανίζει το νευρωνικό δίκτυο κατά την λειτουργία του ως αποκωδικοποιητής αφότου εκπαιδευτεί.

6 Παρουσίαση μετρήσεων και αξιολόγηση αποτελεσμάτων

6.1 Διαδικασία εκπαίδευσης νευρωνικών δικτύων

Σε αυτήν την ενότητα, θα αξιολογηθεί η διαδικασία εκπαίδευσης των νευρωνικών δικτύων που αναφέρθηκαν στο προηγούμενο κεφάλαιο. Να σημειωθεί πως για όλες τις μετρήσεις σε αυτήν την ενότητα, έχει χρησιμοποιηθεί η μέθοδος μορφοποίησης One Hot Encoding, ενώ ως μονάδα επεξεργασίας η GPU.

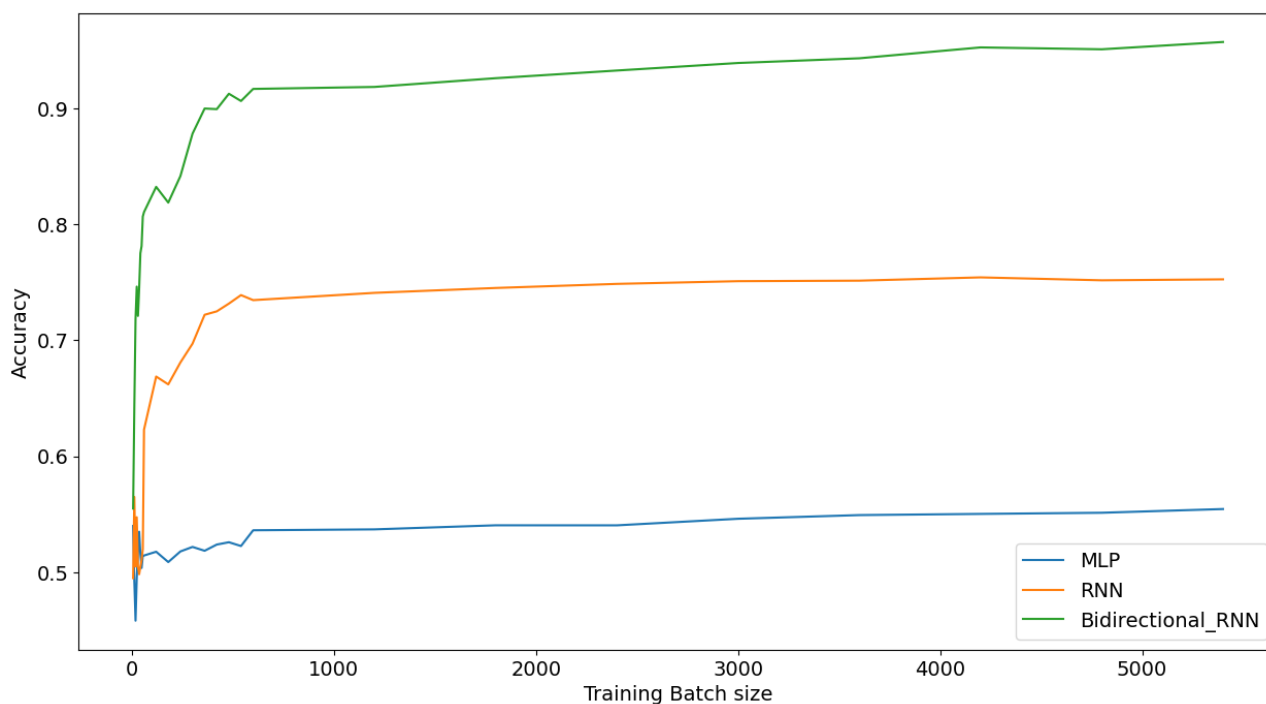
Πραγματοποιείται εκπαίδευση των νευρωνικών δικτύων στην αποκωδικοποίηση μηνυμάτων μήκους 200 ψηφίων. Η πρόβλεψη των νευρωνικών δικτύων και του αποκωδικοποιητή Viterbi θα έχει μήκος 100 bit. Αρχικά επιχειρείται διαδικασία εκπαίδευσης για διάφορα μεγέθη της δέσμης συνόλου εκπαίδευσης (Training Batch Size). Η τιμή του ρυθμού μάθησης που επιλέχθηκε ήταν ίση με $5 \cdot 10^{-3}$ και για τον περιορισμό του φαινομένου της υπερπροσαρμογής γίνεται χρήση της τεχνικής Early Stopping (Ενότητα 3.5). Το σύνολο εκπαίδευσης, επαλήθευσης και αξιολόγησης, θα περιέχουν μηνύματα με θόρυβο (SNR=-1), ώστε τα δίκτυα να εκπαιδευτούν και να αξιολογηθούν σε μια κατάσταση που θα προσομοιάζει την πραγματικότητα. Αφότου ολοκληρωθεί η διαδικασία εκπαίδευσης, τα δίκτυα αξιολογούνται με τη βοήθεια του συνόλου ελέγχου (test set). Στην παρακάτω εικόνα, απεικονίζεται το γράφημα $Test Set Loss = f(Training Batch Size)$ (Η μεταβλητή Loss είναι η τιμή της συνάρτησης απωλειών επί του συνόλου ελέγχου).



Εικόνα 6.1 Test Set Loss = $f(Training Batch Size)$ (Αποκωδικοποίηση μηνυμάτων 200 ψηφίων)

Όπως αναμενόταν, η αύξηση του μεγέθους του συνόλου εκπαίδευσης οδηγεί σε μείωση της τιμής της συνάρτησης απωλειών. Βέβαια από ένα σημείο και έπειτα, η ρυθμός αυτής της μείωσης μετριάζεται. Από ένα σημείο και έπειτα, φαίνεται μάλιστα η μείωση της τιμής της συνάρτησης απωλειών να παύει εντελώς. Αυτό πιθανώς οφείλεται στο γεγονός, πως αφότου το μέγεθος του συνόλου εκπαίδευσης φτάσει μια συγκεκριμένη τιμή, ο αλγόριθμος βελτιστοποίησης θα τείνει να βρίσκει τις βέλτιστες τιμές για τα συναπτικά βάρη, άρα η περαιτέρω αύξηση του θα προξενεί μηδαμινές οι αμελητέες αλλαγές. Τις καλύτερες επιδόσεις για το εύρος μεγεθών, το οποίο έχει χρησιμοποιηθεί, φαίνεται να επιτυγχάνει το RNN διπλής κατεύθυνσης ($\sim 0,1$), ακολουθούμενο από το RNN ($\sim 0,5$) και το MLP ($\sim 0,7$).

Στη συνέχεια παρατίθεται το γράφημα $Test Set Accuracy = f(Training Batch Size)$ (Εικόνα 6.2).



Εικόνα 6.2 $Test Set Accuracy = f(Training Batch Size)$ (Αποκωδικοποίηση μηνυμάτων 200 ψηφίων)

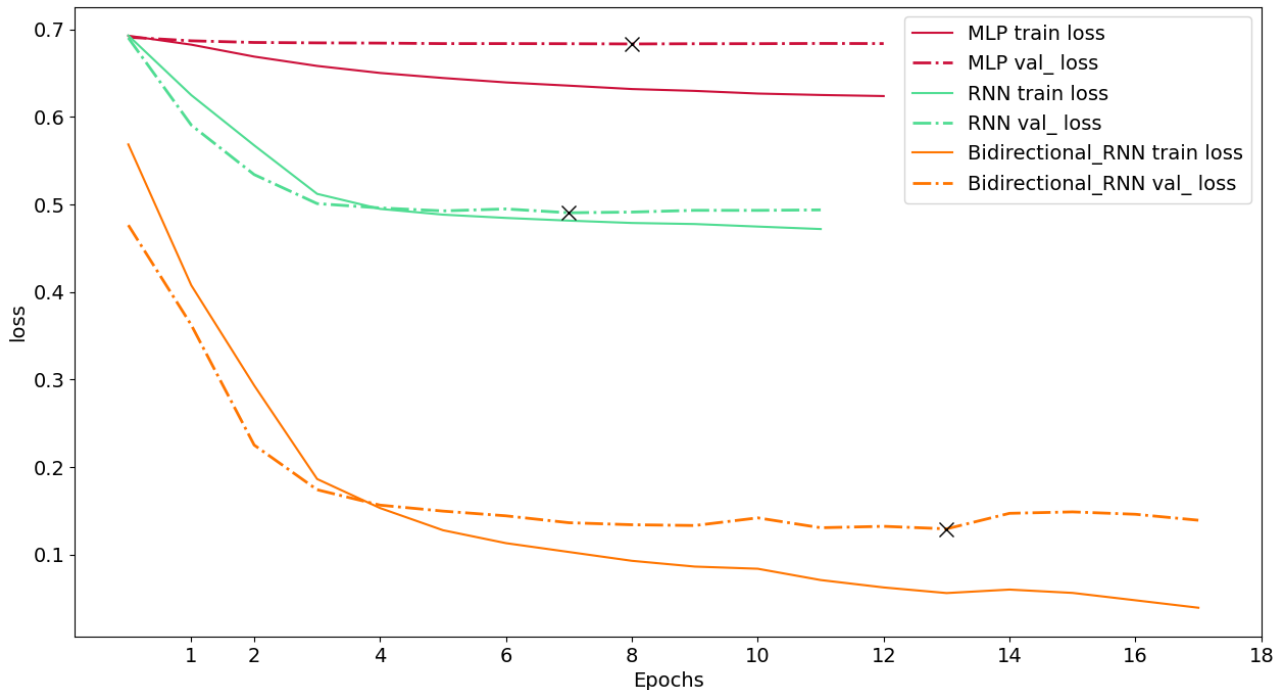
Όπως αναμενόταν, η μετρική αξιολόγησης accuracy ακολουθεί αντιστρόφως ανάλογη πορεία από αυτής του σφάλματος. Καθώς αυξάνεται το μέγεθος του συνόλου εκπαίδευσης, τόσο μεγαλύτερη η τιμή της Accuracy. Η αύξηση της μετρικής Accuracy, σημαίνει την αύξηση του ποσοστού των προβλέψεων του νευρωνικού δικτύου που συμπίπτει με τις προβλέψεις του αποκωδικοποιητή Viterbi. Αυτό σηματοδοτεί μια καλύτερη προσέγγιση της λειτουργίας του αποκωδικοποιητή Viterbi. Τις καλύτερες επιδόσεις εμφανίζει το RNN διπλής κατεύθυνσης (επιτυγχάνει ποσοστά ακρίβειας

της τάξης του 90-95%). Μεταξύ των άλλων δύο, το RNN είναι το καλύτερο (70-75%) και το χειρότερο είναι το MLP (50-55%).

Από την παρατήρηση των γραφημάτων μπορεί εύκολα να πει κανείς, πως το MLP δεν δείχνει να βελτιώνει σχεδόν καθόλου τη λειτουργία του, όσο και αν αυξηθεί το μέγεθος του συνόλου εκπαίδευσης. Αυτό σημαίνει πως μάλλον αυτού του είδους οι αρχιτεκτονικές είναι ακατάλληλες για να μιμηθούν μια διεργασία όπως η αποκωδικοποίηση Viterbi. Αυτό δεν είναι απρόσμενο, αν αναλογιστεί κανείς πως τα MLP δεν περιέχουν κάποιον μηχανισμό μνήμης εν αντιθέσει με τα δίκτυα RNN. Αυτό τα καθιστά εν γένει χειρότερα στην επεξεργασία σειριακών/ακολουθιακών δεδομένων, όπως είναι αυτά τα οποία επεξεργαζόμαστε.

Επίσης, γίνεται αντιληπτό πως τα RNN διπλής κατεύθυνσης είναι ικανότερα στην μίμηση της λειτουργίας του αποκωδικοποιητή Viterbi από ένα απλό RNN. Αυτό συμβαίνει γιατί ο αποκωδικοποιητής Viterbi είναι ένας αποκωδικοποιητής μεγίστης πιθανοφάνειας. Ένας τέτοιου είδους αποκωδικοποιητής δεν αποκωδικοποιεί ένα ψηφίο τη φορά. Αντιθέτως, αφότου λάβει και επεξεργαστεί ολόκληρη την ακολουθία, προβαίνει στην πρόβλεψη του πιθανότερου μονοπατιού. Η δομή ενός απλού RNN (μονής κατεύθυνσης) είναι τέτοια που να επιτρέπει τη συμβολή της τρέχουσας και όλων των προηγούμενων εισόδων στην διαμόρφωση της τιμής της τρέχουσας πρόβλεψης, αλλά καμίας μελλοντικής. Το RNN διπλής κατεύθυνσης λαμβάνει υπόψιν του για την διαμόρφωση της τρέχουσας πρόβλεψης, τόσο τις προηγούμενες όσο και τις επόμενες εισόδους. Με άλλα λόγια, λαμβάνει υπόψιν του όλη την κωδικοποιημένη ακολουθία εισόδου για την πρόβλεψη του κάθε ψηφίου της αποκωδικοποιημένης ακολουθίας εξόδου. Άρα είναι απολύτως λογικό να μπορεί να μιμηθεί καλύτερα έναν αποκωδικοποιητή μεγίστης πιθανοφάνειας, όπως ο αποκωδικοποιητής Viterbi. Για να γίνει ακόμα πιο σαφές στον αναγνώστη, θα δοθεί ένα κατάλληλο παράδειγμα: έστω πως στο χρονικό στάδιο $t=6$ μια από τις πιθανές υποδιαδρομές έχει Ευκλείδεια απόσταση από τη ληφθείσα ακολουθία ίση με 57 (υποδιαδρομή 1) ενώ μια άλλη υποδιαδρομή έχει απόσταση ίση με 58 (υποδιαδρομή 2). Στο επόμενο χρονικό βήμα, και από τις δύο προκύπτουν διαδρομές που καταλήγουν στην ίδια κατάσταση. Με τις ανανεωμένες Ευκλείδειες αποστάσεις, η διαδρομή που φύεται από την υποδιαδρομή 1 έχει απόσταση ίση με 65, ενώ η διαδρομή που φύεται από την υποδιαδρομή 2 έχει απόσταση ίση με 61. Ο αποκωδικοποιητής Viterbi επιλέγει ως πιθανότερη την διαδρομή που φύεται από την υποδιαδρομή 2. Ένα επαρκώς εκπαιδευμένο RNN (μονής κατεύθυνσης) δεν είναι ικανό να αναπαράγει ασφαλώς μια τέτοια απόφαση, καθώς δεν έχει γνώση του μέλλοντος, άρα θα συνεχίσει μέχρι να θεωρεί ως πιθανότερη την υποδιαδρομή '1' (μέχρι το χρονικό στάδιο $t=6$).

Στη συνέχεια εξετάζεται πιο προσεκτικά η διαδικασία εκπαίδευσης. Το μέγεθος της δέσμης του συνόλου εκπαίδευσης θα είναι 6000 (6000 ζεύγη εισόδου-επιθυμητής απόκρισης). Στην Εικόνα 6.3 απεικονίζεται το σφάλμα εκπαίδευσης (train_loss) και το σφάλμα επαλήθευσης (val_loss) συναρτήσει των εποχών³.



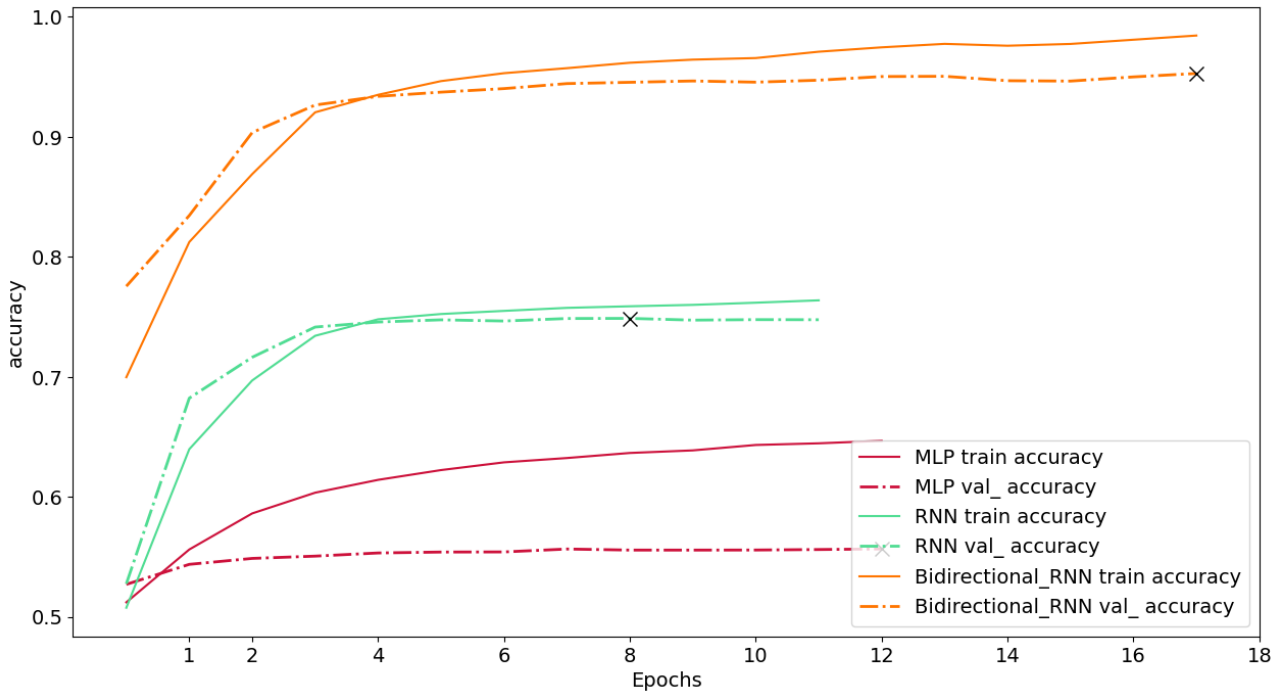
Εικόνα 6.3 $Loss = f(Epochs)$ (Αποκωδικοποίηση μηνυμάτων 200 ψηφίων)

Καθώς αυξάνεται ο αριθμός των εποχών, η τιμή της συνάρτησης απωλειών μειώνεται. Αρχικά μειώνεται τόσο για το σύνολο εκπαίδευσης όσο και για το σύνολο επαλήθευσης. Από μια εποχή και έπειτα, όμως, αρχίζει να μειώνεται μόνο το σφάλμα του συνόλου εκπαίδευσης ενώ το σφάλμα του συνόλου επαλήθευσης παραμένει σταθερό ή και εμφανίζει ελαφρώς αυξητικές τάσεις. Αυτό είναι μια ένδειξη ότι το εκπαιδευόμενο δίκτυο αρχίζει να υπερπροσαρμόζεται. Για αυτό γίνεται χρήση της τεχνικής Early Stopping. Η τεχνική αυτή τερματίζει πρόωρα τον αλγόριθμο μηχανικής μάθησης όταν για ένα συγκεκριμένο αριθμό εποχών δεν υπάρχει βελτίωση στο σφάλμα του συνόλου επαλήθευσης (val_loss). Στην υλοποίηση αυτή έχει επιλεγεί αυτός ο αριθμός να είναι ίσος με 4. Αφότου ολοκληρωθεί η διαδικασία της εκπαίδευσης, οι παράμετροι από την εποχή που επιτεύχθηκε η καλύτερη επίδοση επί του συνόλου επαλήθευσης, επιλέγονται ως παράμετροι του

³ Εποχή (Epoch) ονομάζεται μια πλήρης διέλευση από το ολόκληρο σύνολο εκπαίδευσης κατά την εκπαίδευση του νευρωνικού δικτύου. Κατά την διάρκεια μιας εποχής ο αλγόριθμος μηχανικής μάθησης επεξεργάζεται όλο το σύνολο δεδομένων εκπαίδευσης. Ο αριθμός των εποχών είναι μια υπερπαράμετρος, η οποία ορίζεται από τον σχεδιαστή προτού ξεκινήσει η διαδικασία εκπαίδευσης.

τελικού μοντέλου. Αυτή η εποχή συμβολίζεται με ένα x στο γράφημα. Όπως είδαμε και προηγουμένως, το RNN διπλής κατεύθυνσης εμφανίζει τις καλύτερες επιδόσεις.

Στην Εικόνα 6.4 απεικονίζεται η τιμή της μετρικής accuracy για το σύνολο εκπαίδευσης (train_accuracy) και το σύνολο επαλήθευσης (val_accuracy) συναρτήσει των εποχών.



Εικόνα 6.4 Accuracy = f (Epochs) (Αποκωδικοποίηση μηνυμάτων 200 ψηφίων)

Η εικόνα του γραφήματος είναι αναμενόμενη βάσει της εικόνας και του γραφήματος της εικόνας 6.4. Η πορεία που ακολουθούν οι καμπύλες της μετρικής accuracy είναι αντιστροφή από την πορεία των καμπύλων της συνάρτησης απωλειών.

Μετά την διαδικασία της εκπαίδευσης, λαμβάνει χώρα η διαδικασία ελέγχου και αξιολόγησης της διαδικασίας εκπαίδευσης. Αυτή γίνεται με τη βοήθεια του συνόλου ελέγχου (2000 ζεύγη διανυσμάτων εισόδου και επιθυμητών αποκρίσεων που αντλήθηκαν από τον αποκωδικοποιητή Viterbi). Τα αποτελέσματα, τα οποία προκύπτουν από την διαδικασία αξιολόγησης, καταγράφονται στον παρακάτω πίνακα.

Τύπος Νευρωνικού	Test Loss	Test Accuracy
MLP	0,6842	0,5533
RNN	0,4911	0,7491
Bidirectional RNN	0,1318	0,9515

Πίνακας 6.1 Δεδομένα αξιολόγησης νευρωνικών δικτύων έπειτα από την διαδικασία εκπαίδευσης

Αν παρατηρήσει κανείς τα δεδομένα του Πίνακα 6.1 καθώς και τις Εικόνες 6.3 και 6.4, μπορεί εύκολα κανείς να παρατηρήσει ότι το φαινόμενο της υπερπροσαρμογής δεν είναι έντονο. Το χάσμα μεταξύ των μετρήσεων, οι οποίες αντιστοιχούν στο σύνολο εκπαίδευσης και στο σύνολο ελέγχου, δεν είναι πολύ μεγάλο. Σε αυτό βοήθησε και επιστράτευση της τεχνικής Early Stopping.

Στις παρακάτω εικόνες φαίνονται μερικά στιγμιότυπα από την διαδικασία εκπαίδευσης και αξιολόγησης των νευρωνικών στο προγραμματιστικό περιβάλλον.

```
Epoch 1/50
47/47 [=====] - 6s 33ms/step - loss: 0.6922 - accuracy: 0.5118 - val_loss: 0.6820 - val_accuracy: 0.5545
Epoch 2/50
47/47 [=====] - 1s 15ms/step - loss: 0.6133 - accuracy: 0.6581 - val_loss: 0.5870 - val_accuracy: 0.6840
Epoch 3/50
47/47 [=====] - 1s 15ms/step - loss: 0.5653 - accuracy: 0.6972 - val_loss: 0.5339 - val_accuracy: 0.7173
Epoch 4/50
47/47 [=====] - 1s 16ms/step - loss: 0.5139 - accuracy: 0.7314 - val_loss: 0.5036 - val_accuracy: 0.7390
Epoch 5/50
47/47 [=====] - 1s 20ms/step - loss: 0.4960 - accuracy: 0.7460 - val_loss: 0.4990 - val_accuracy: 0.7432
Epoch 6/50
47/47 [=====] - 1s 19ms/step - loss: 0.4879 - accuracy: 0.7519 - val_loss: 0.4917 - val_accuracy: 0.7485
Epoch 7/50
47/47 [=====] - 1s 20ms/step - loss: 0.4841 - accuracy: 0.7544 - val_loss: 0.4947 - val_accuracy: 0.7465
Epoch 8/50
47/47 [=====] - 1s 20ms/step - loss: 0.4815 - accuracy: 0.7566 - val_loss: 0.4928 - val_accuracy: 0.7481
Epoch 9/50
47/47 [=====] - 1s 16ms/step - loss: 0.4776 - accuracy: 0.7593 - val_loss: 0.4951 - val_accuracy: 0.7461
Epoch 10/50
45/47 [=====>...] - ETA: 0s - loss: 0.4758 - accuracy: 0.7604Restoring model weights from the end of the best epoch.
47/47 [=====] - 1s 15ms/step - loss: 0.4753 - accuracy: 0.7606 - val_loss: 0.4959 - val_accuracy: 0.7469
Epoch 00010: early stopping
```

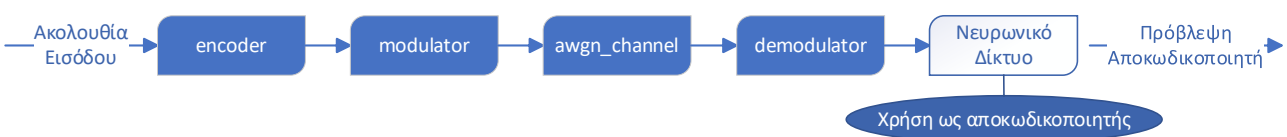
Εικόνα 6.5 Εκπαίδευση ενός νευρωνικού δικτύου

```
MLP:
63/63 [=====] - 0s 3ms/step - loss: 0.6842 - accuracy: 0.5533
RNN:
63/63 [=====] - 0s 8ms/step - loss: 0.4911 - accuracy: 0.7491
Bidirectional RNN:
63/63 [=====] - 1s 9ms/step - loss: 0.1318 - accuracy: 0.9515
[0.13176439702510834, 0.9514999985694885]
```

Εικόνα 6.6 Αξιολόγηση εκπαιδευμένων νευρωνικών δικτύων

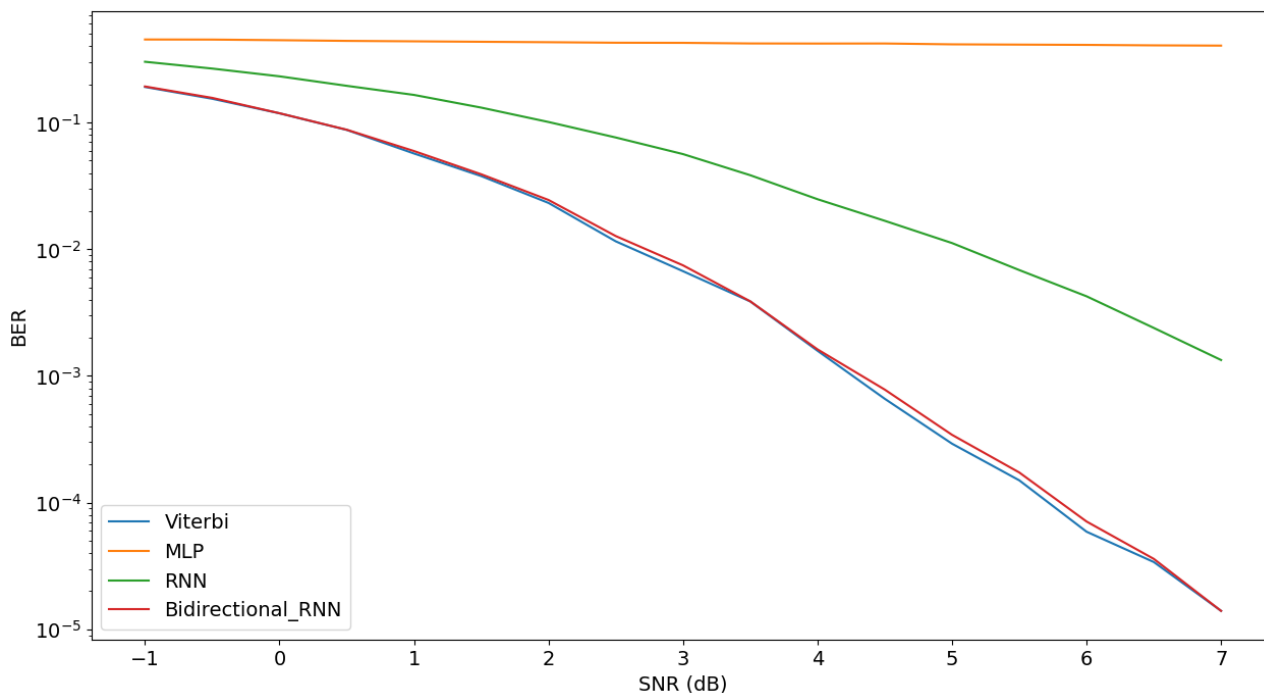
6.2 Αξιολόγηση λειτουργίας εκπαιδευμένων νευρωνικών δικτύων ως αποκωδικοποιητές σημάτων

Σε αυτήν την ενότητα τα δίκτυα, τα οποία εκπαιδεύτηκαν στην προηγούμενη παράγραφο (6000 ζεύγη εισόδου-επιθυμητής εξόδου) για να μιμούνται την λειτουργία του αποκωδικοποιητή Viterbi, δοκιμάζονται και αξιολογούνται ως προς την αποκωδικοποίηση σημάτων, τα οποία έχουν κωδικοποιηθεί με τη χρήση του συνελκτικού κωδικοποιητή της Εικόνας 2.1.



Εικόνα 6.7 Χρήση εκπαιδευμένου νευρωνικού δικτύου ως αποκωδικοποιητής σημάτων

Στην Εικόνα 6.7 απεικονίζεται ένα γράφημα του ρυθμού σφάλματος ανά ψηφίο της αποκωδικοποιημένης ακολουθίας (BER) συναρτήσεως του σηματοθορυβικού λόγου. Φαίνονται οι επιδόσεις των τριών νευρωνικών δικτύων, που εκπαιδεύτηκαν προηγουμένως, καθώς και του παραδοσιακού αλγορίθμου Viterbi. Οι τιμές του SNR, οι οποίες εξετάστηκαν ήταν στο εύρος $[-1,7]$, ενώ για κάθε τιμή του SNR εξετάζονταν μια δέσμη με 10.000 μηνύματα μήκους 100 bits. Αυτό σημαίνει πως για κάθε τιμή του SNR εξετάζονται 10^6 bits, άρα μπορούν να εντοπιστούν με σχετική ακρίβεια BER της τάξης του 10^{-5} .



Εικόνα 6.8 BER = f(SNR)

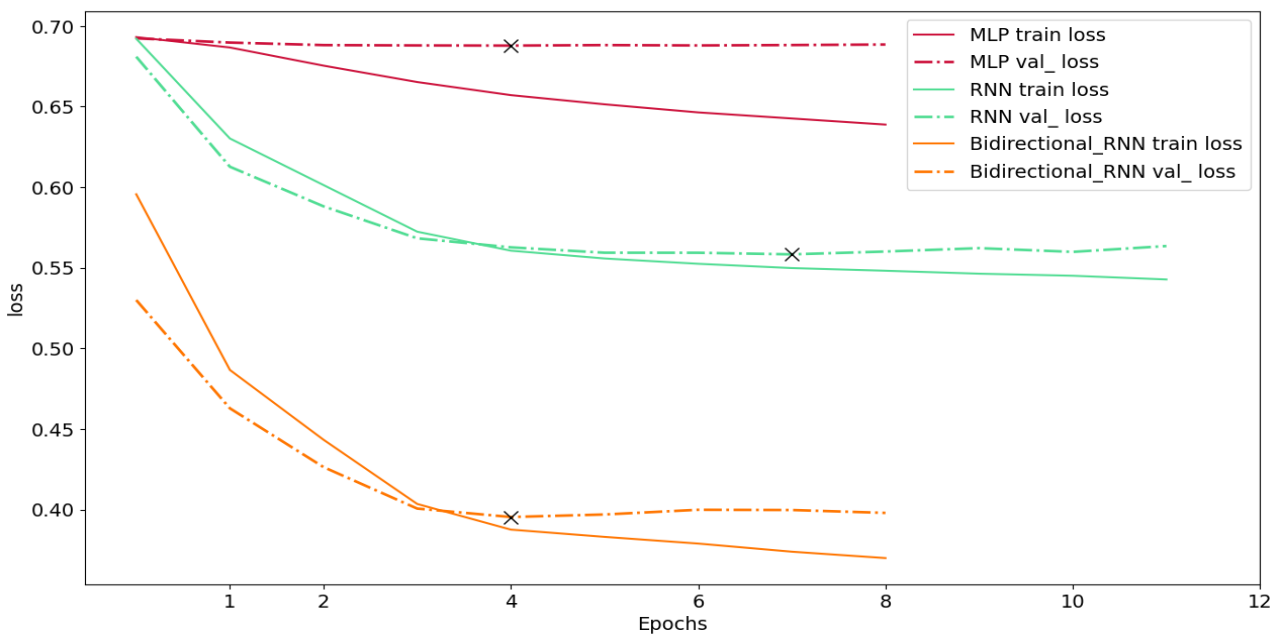
Την καλύτερη επίδοση στην διόρθωση σημάτων μεταξύ των νευρωνικών δικτύων έχει το RNN διπλής κατεύθυνσης, ακολουθούμενο από το RNN, ενώ τις χειρότερες επιδόσεις εμφανίζει το MLP. Με άλλα λόγια, το νευρωνικό δίκτυο, το οποίο προσέγγιζε καλύτερα την λειτουργία του αποκωδικοποιητή Viterbi, εμφανίζει τις καλύτερες επιδόσεις στην αποκωδικοποίηση σημάτων. Αυτό είναι λογικό επειδή ο αλγόριθμος Viterbi είναι η πιο αξιόπιστη μέθοδος αποκωδικοποίησης σημάτων, που έχουν κωδικοποιηθεί από συνελκτικό κωδικοποιητή.

Το RNN διπλής κατεύθυνσης έχει επιδόσεις ανάλογες με τον παραδοσιακό αποκωδικοποιητή Viterbi. Οι καμπύλες τους σχεδόν συμπίπτουν. Αν και φαίνεται μια πολύ ελαφριά υπεροχή στις επιδόσεις του αποκωδικοποιητή Viterbi, ο ρυθμός σφαλμάτων είναι της ίδια τάξης μεγέθους και μάλιστα σχεδόν ίδιος.

6.3 Έλεγχος αλγορίθμου Viterbi και καταλληλότητας του ως εκπαιδευτής

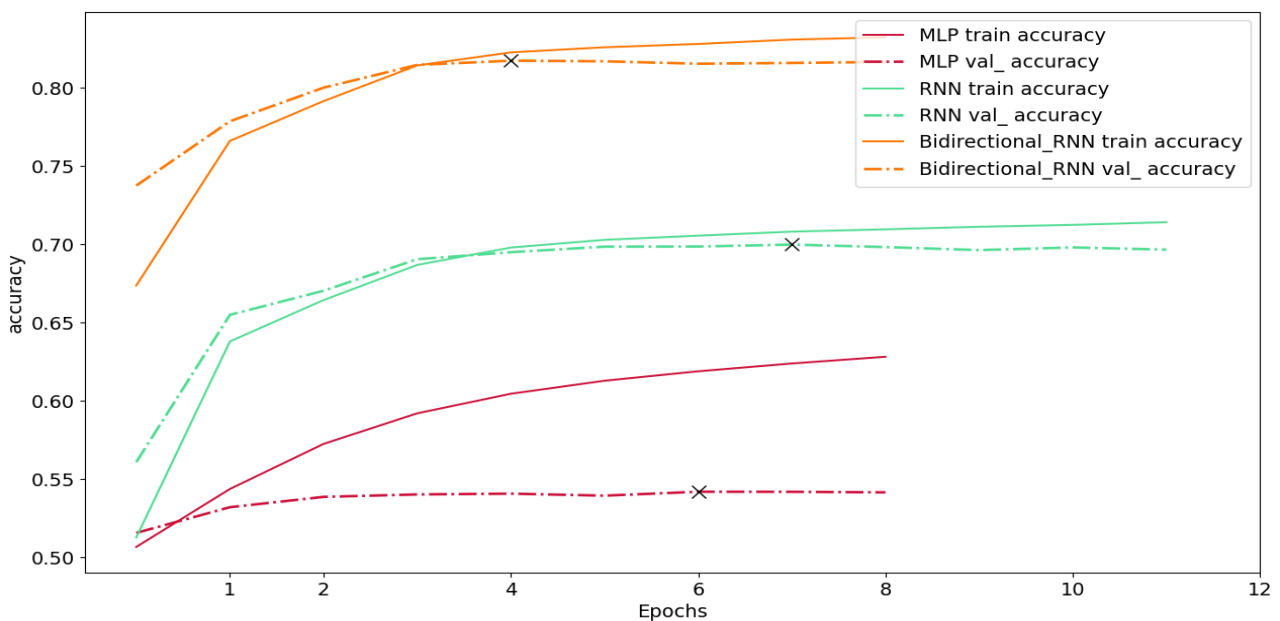
Ακόμη, έγινε και μια άλλη δοκιμή. Έγινε εκπαίδευση των δικτύων χρησιμοποιώντας ως επιθυμητές αποκρίσεις τις τυχαίες ακολουθίες που στάλθηκαν και όχι τις προβλέψεις του αποκωδικοποιητή Viterbi. Όλες οι υπόλοιπες παράμετροι κρατήθηκαν ίδιες (μήκος τυχαίας ακολουθίας = 100, μέγεθος δέσμης συνόλου εκπαίδευσης = 6000, SNR = -1, ρυθμός μάθησης = $5 \cdot 10^{-3}$).

Στα γραφήματα των παρακάτω εικόνων φαίνεται η τιμή της συνάρτησης απωλειών και της μετρικής accuracy συναρτήσει των εποχών.



Εικόνα 6.9

,



Εικόνα 6.10

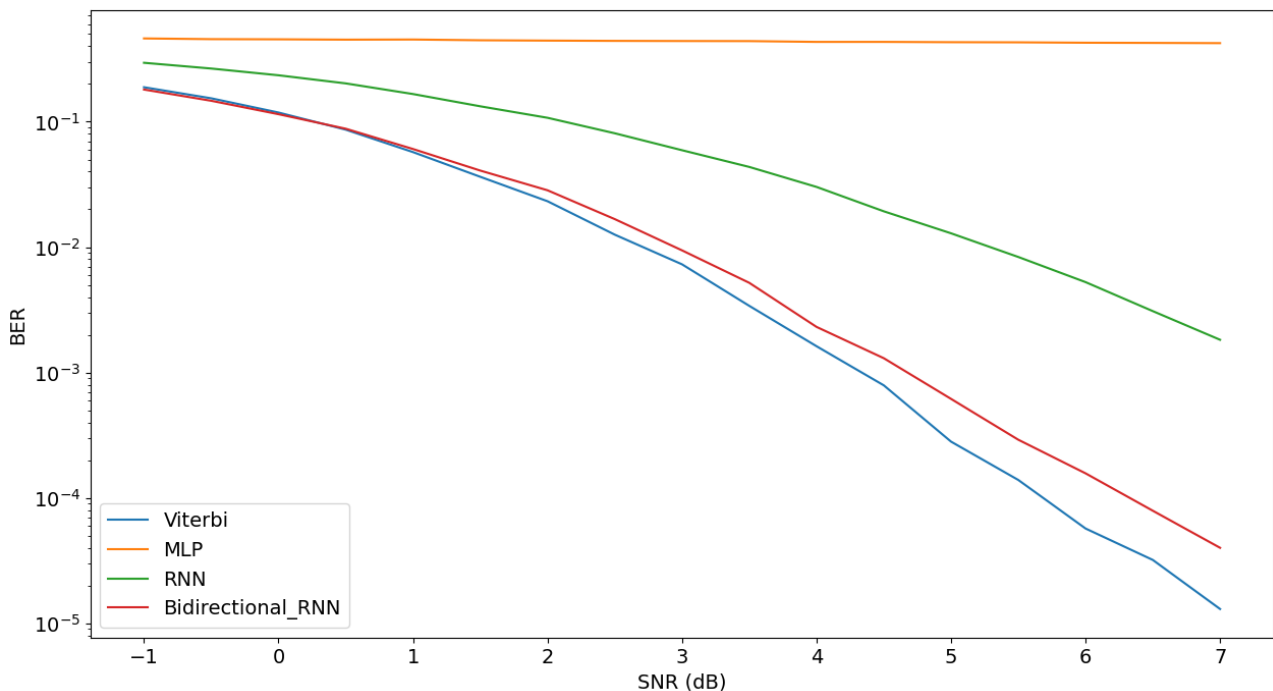
Μετά την εκπαίδευση, γίνεται μια πρώτη αξιολόγηση της διαδικασίας εκπαίδευσης με το σύνολο ελέγχου. Τα δεδομένα αξιολόγησης συγκεντρώνονται στον Πίνακα 6.2.

Τύπος Νευρωνικού	<i>Test Loss</i>	<i>Test Accuracy</i>
MLP	0,6873	0,5422
RNN	0,5573	0,7005
Bidirectional RNN	0,3919	0,8211

Πίνακας 6.2

Και σε αυτή την περίπτωση, το RNN διπλής κατεύθυνσης εμφανίζει τα καλύτερα αποτελέσματα. Αυτό είναι μια περαιτέρω απόδειξη ότι η δομή του είναι κατάλληλη για την αποκωδικοποίηση μηνυμάτων κωδικοποιημένων με χρήση συνελκτικού κώδικα. Τα ποσοστά accuracy του RNN σε αυτήν την περίπτωση είναι παρόμοια με τα ποσοστά που θα είχε ένας αποκωδικοποιητής Viterbi.

Στην Εικόνα 6.11 απεικονίζεται το γράφημα του BER των νευρωνικών δικτύων, τα οποία εκπαιδεύτηκαν έχοντας ως επιθυμητές αποκρίσεις τις τυχαίες ακολουθίες που στάλθηκαν και όχι τις προβλέψεις του Viterbi.



Εικόνα 6.11. $BER = f(SNR)$ για νευρωνικά δίκτυα που εκπαιδεύονται με τη χρήση των αρχικών τυχαίων ακολουθιών πληροφορίας ως επιθυμητές αποκρίσεις

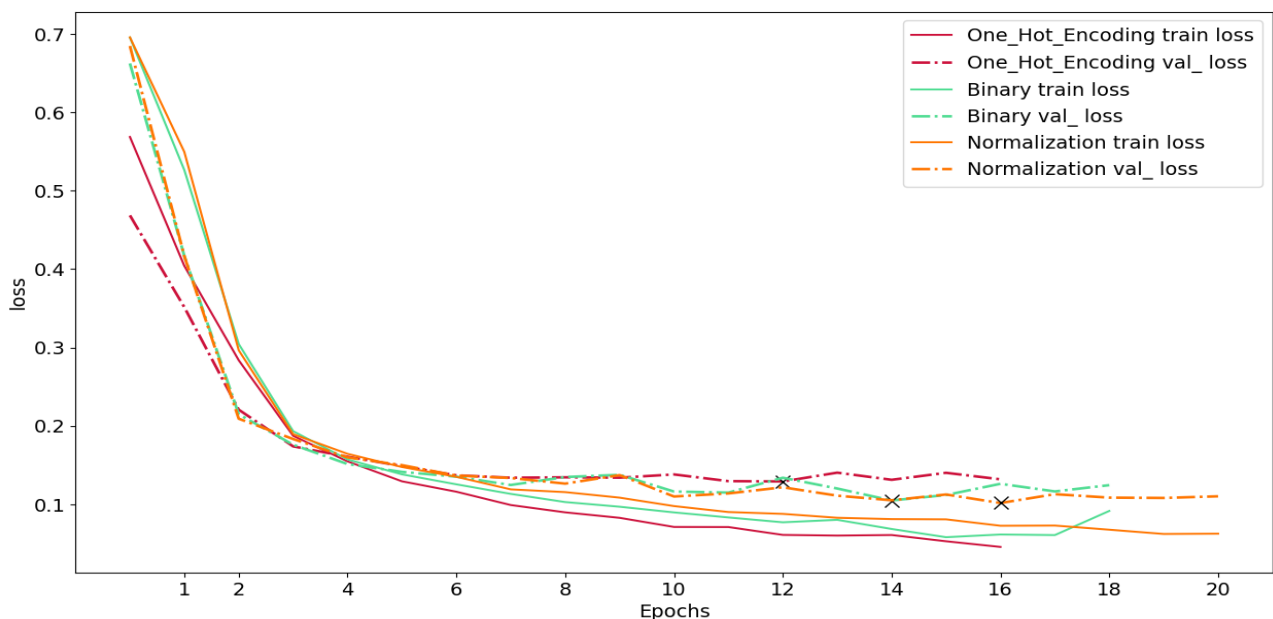
Η μορφή των γραφικών αναπαραστάσεων μοιάζει με αυτή της Εικόνας 6.8. Το RNN διπλής κατεύθυνσης τείνει να πλησιάσει τις επιδόσεις το κλασικού αλγορίθμου Viterbi, χωρίς όμως να τις

ξεπερνά. Αυτό είναι μία απόδειξη ότι ο αλγόριθμος Viterbi είναι ο βέλτιστος αλγόριθμος για την αποκωδικοποίηση σημάτων συνελκτικού κώδικα. Επίσης η σύγκλιση μεταξύ της καμπύλης του Bidirectional RNN και του Viterbi όταν χρησιμοποιήθηκε ως εκπαιδευτής ο αποκωδικοποιητής Viterbi ήταν ελαφρώς μεγαλύτερη. Αυτό οφείλεται στο γεγονός, πως ακόμη και όταν η πρόβλεψη του Viterbi είναι εσφαλμένη, αυτό οφείλεται στην μεγάλη παραμόρφωση της ληφθείσας ακολουθίας από τον θόρυβο και όχι σε κάποια εγγενή αδυναμία του Viterbi. Ο αλγόριθμος Viterbi πάντα βρίσκει την πιθανότερη ακολουθία πληροφορίας βάσει αυτού του μηνύματος που έλαβε. Όλα αυτά τα δεδομένα επιβεβαιώνουν πως η χρήση του αποκωδικοποιητή Viterbi σαν εκπαιδευτής των νευρωνικών δικτύων ήταν ορθή.

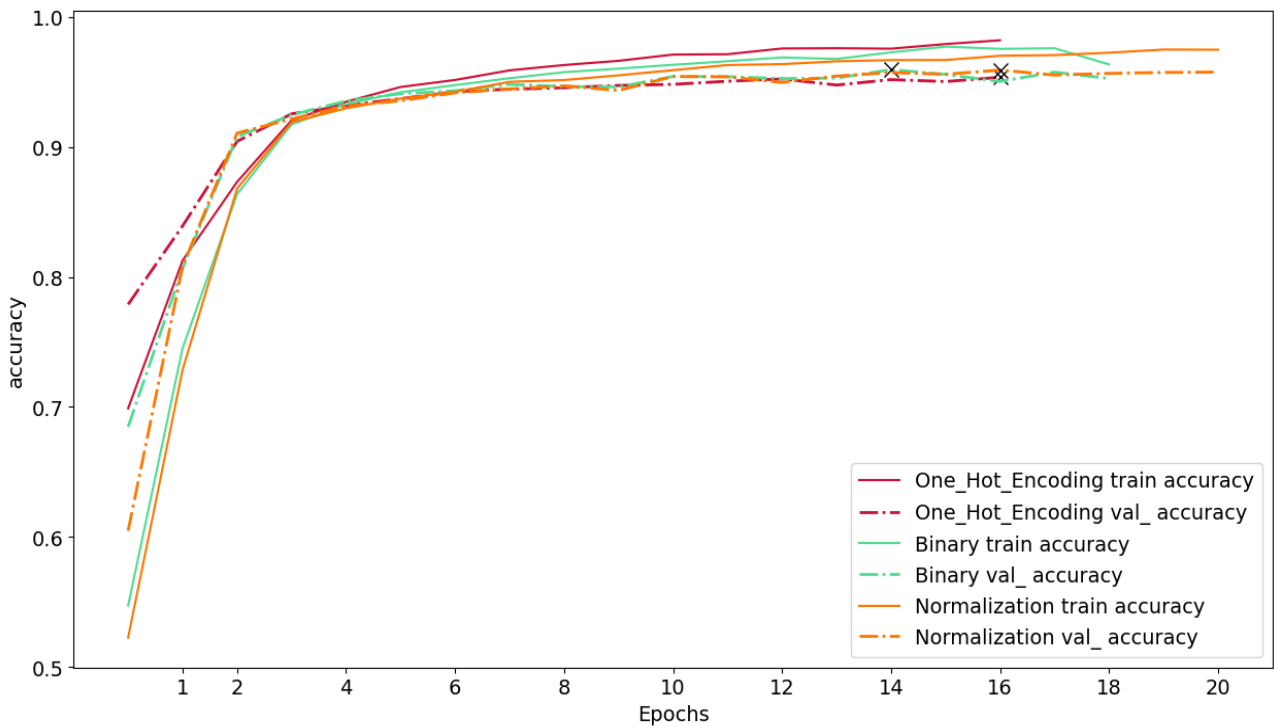
Πρέπει να σημειωθεί βέβαια, πως η εκτέλεση του αλγορίθμου Viterbi, εισάγει επιπλέον υπολογιστικό φόρτο και κάνει την διαδικασία συλλογής δεδομένων πιο δύσκολη και αργή.

6.4 Αξιολόγηση μεθόδων μορφοποίησης εισόδου

Σε όλα τα παραδείγματα μέχρι στιγμής έχει χρησιμοποιηθεί η μέθοδος One Hot Encoding για την αναπαράσταση της ληφθείσας ακολουθίας, η οποία τροφοδοτείται στα νευρωνικά δίκτυα. Σε αυτήν την ενότητα θα συγκριθούν και οι τρεις μέθοδοι μορφοποίησης, οι οποίες αναφέρθηκαν στην ενότητα 5.3. Το ίδιο νευρωνικό δίκτυο (RNN διπλής κατεύθυνσης) θα εκπαιδευτεί, με τη χρήση των 3 διαφορετικών μορφοποιήσεων. Οι παράμετροι που καθορίζονται από τον χρήστη, θα παραμείνουν σταθερές και για τις 3 περιπτώσεις (μήκος τυχαίας ακολουθίας = 100, μέγεθος δέσμης συνόλου εκπαίδευσης = 6000, SNR = -1, ρυθμός μάθησης = $5 \cdot 10^{-3}$). Αρχικά, παρατίθενται τα γραφήματα Loss = f (Epochs) και Accuracy = f (Epochs) για το training set και το validation set.



Εικόνα 6.12 Loss = f (Epochs) για διαφορετικές μορφοποιήσεις δεδομένων εισόδου



Εικόνα 6.13 Accuracy = f (Epochs) για διαφορετικές μορφοποιήσεις δεδομένων εισόδου

Η διαδικασία εκπαίδευσης και για τις 3 τεχνικές μορφοποίησης δεδομένων εμφανίζει παρόμοια εξέλιξη. Μια παρατήρηση, η οποία μπορεί να γίνει, είναι πως με τη χρήση της τεχνικής One Hot Encoding φαίνεται να εμφανίζεται ένα μεγαλύτερο χάσμα μεταξύ τις τιμές του σφάλματος (και του accuracy) για το σύνολο επαλήθευσης και το σύνολο εκπαίδευσης. Αυτό μπορεί να δείχνει μια ελαφρώς μεγαλύτερη τάση για υπερπροσαρμογή, όταν γίνεται χρήση One Hot Encoding για τα δεδομένα εισόδου.

Όσον αφορά τις επιδόσεις που εμφάνισαν τα εκπαιδευμένα νευρωνικά δίκτυα, όταν κλήθηκαν να επεξεργαστούν ακολουθίες εισόδου που δεν είχαν ξανασυναντήσει, αυτές καταγράφονται στον Πίνακα 6.3.

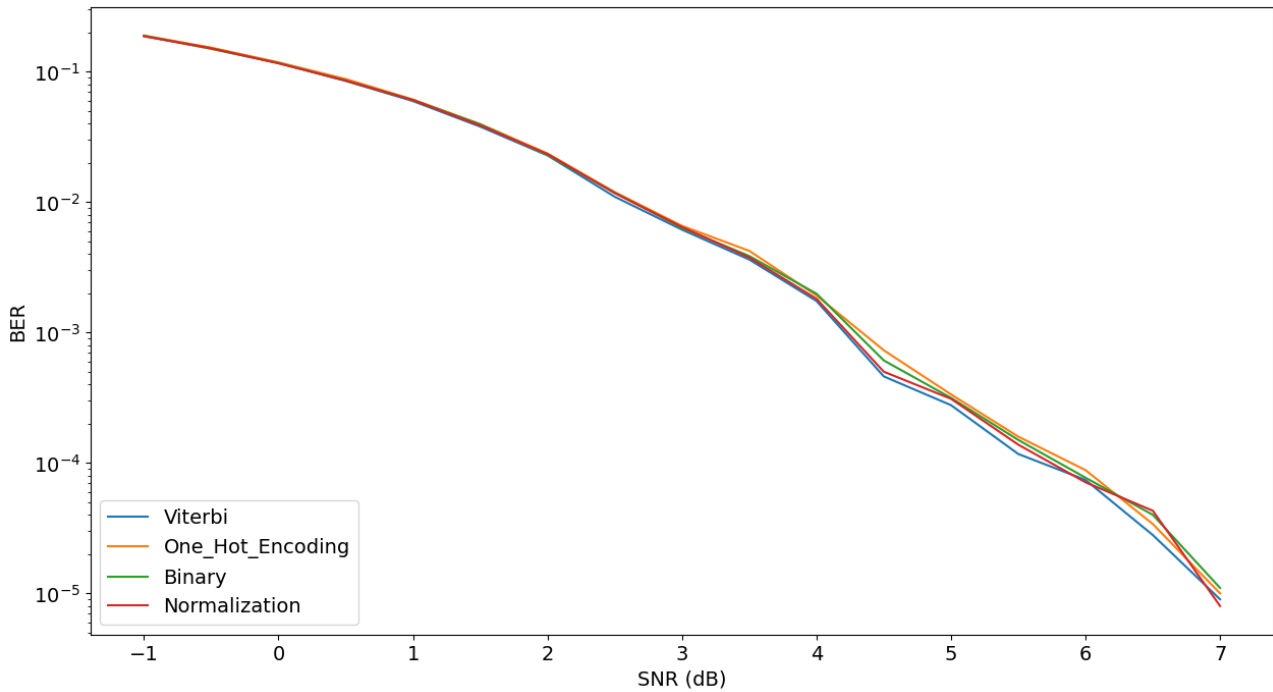
Τύπος Μορφοποίησης	<i>Test Loss</i>	<i>Test Accuracy</i>
One Hot Encoding	0,1351	0,9496
Binary Form	0,1097	0,9585
Normalization	0,1072	0,9575

Πίνακας 6.3

Αυτό που μπορεί να παρατηρηθεί στις τιμές του παραπάνω πίνακα είναι πως η τιμή της συνάρτησης απωλειών είναι ελαφρώς μεγαλύτερη για την περίπτωση του One Hot Encoding από

τις αντίστοιχες τιμές για το Binary Representation και το Normalization. Ακόμη εμφανίζει ελαφρώς μικρότερο Accuracy (διαφορά της τάξης του 0,5%)

Στην Εικόνα 6.14, με μετρήσεις οι οποίες πάρθηκαν με τη χρήση των εκπαιδευμένων μοντέλων, παρατίθεται το γράφημα BER=f(SNR).



Εικόνα 6.14 BER = f(SNR) για διάφορες μορφοποιήσεις των δεδομένων εισόδου

Από αυτό το γράφημα φαίνεται πως και με τις 3 διαμορφώσεις επιτυγχάνονται ανάλογα αποτελέσματα. Οι 3 καμπύλες σχεδόν συμπίπτουν μεταξύ τους και με αυτήν του αλγορίθμου Viterbi. Οι διαφορές είναι πάρα πολύ μικρές.

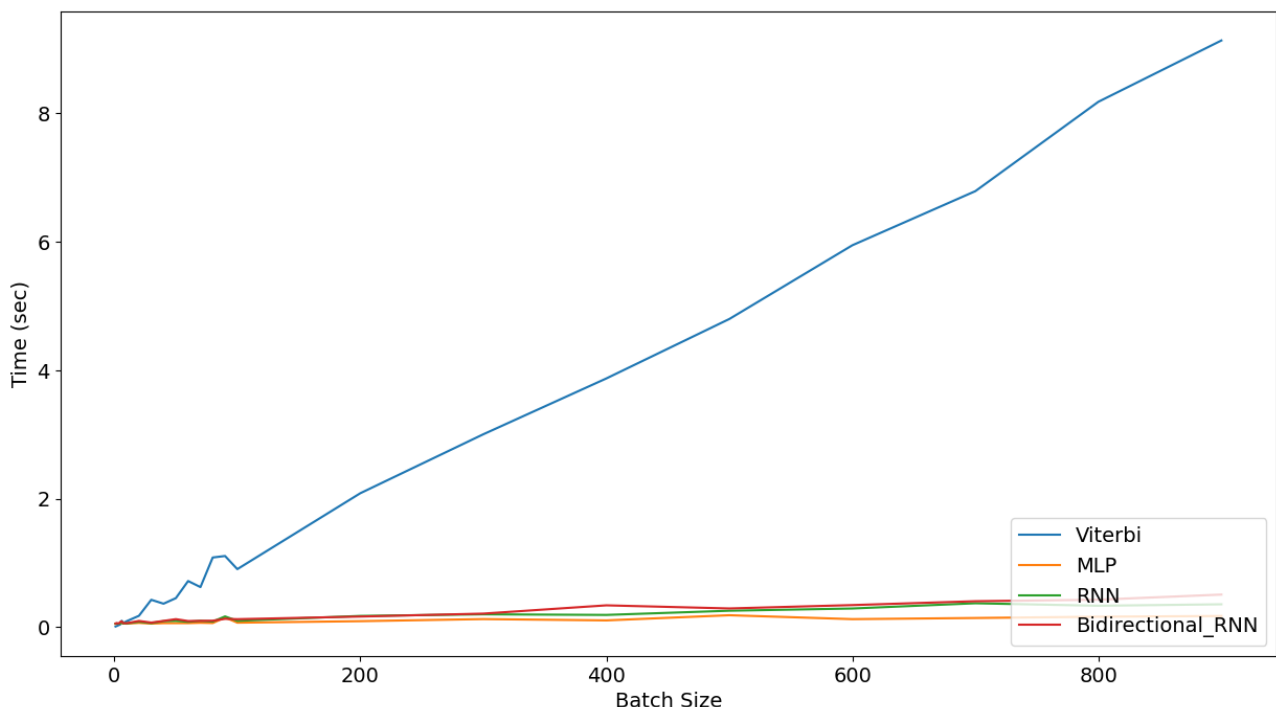
Αξίζει να σημειωθεί, πως η τεχνική One Hot Encoding έχει πολύ μεγαλύτερες απαιτήσεις σε μνήμη από τις άλλες δύο, ιδιαίτερα όταν ο αριθμός των κβαντίσεων στον αποδιαμορφωτή είναι πολύ μεγάλος. Στην περίπτωση που εξετάζεται εδώ (οκτώ επίπεδα κβάντισης), χρειάζεται ένας πίνακας 100 X 64 για να αποτυπωθεί μια ληφθείσα ακολουθία με τη μέθοδο One Hot Encoding, ενώ για την μέθοδο Normalization χρειάζεται ένας πίνακας 100 X 2 και για τη μέθοδο Binary Form ένας πίνακας 100 X 6.

6.5 Μελέτη χρόνων εκτέλεσης

Ο χρόνος, ο οποίος χρειάζεται ένας αποκωδικοποιητής να επεξεργαστεί και να αποκωδικοποιήσει τις ληφθείσες ακολουθίες, είναι ένας από τους παράγοντες που επηρεάζουν την καθυστέρηση (latency) στην μεταγωγή πακέτων σε ένα τηλεπικοινωνιακό σύστημα. Αυτός ο χρόνος είναι

συνήθως ανάλογος της υπολογιστικής πολυπλοκότητας του μοντέλου που χρησιμοποιείται στον αποκωδικοποιητή.

Για αυτόν το λόγο, μελετήθηκαν οι χρόνοι, που χρειάστηκαν τα εκπαιδευμένα νευρωνικά δίκτυα και ο αποκωδικοποιητής Viterbi, για να αποκωδικοποιήσουν τα μηνύματα. Συγκεκριμένα τροφοδοτήθηκαν δέσμες μηνυμάτων μήκους 200 ψηφίων στα εξεταζόμενα μοντέλα και έγινε μέτρηση του χρόνου που χρειάστηκε για την επεξεργασία της δέσμης και την παραγωγή προβλέψεων (οι προβλέψεις έχουν μήκος 100 bits). Αυτή διαδικασία έγινε για ένα εύρος μεγεθών της δέσμης. Το γράφημα του χρόνου εκτέλεσης της αποκωδικοποίησης συναρτήσει του μεγέθους της δέσμης των μηνυμάτων για τα διαφορετικά μοντέλα αποκωδικοποίησης φαίνονται στην Εικόνα 6.15.

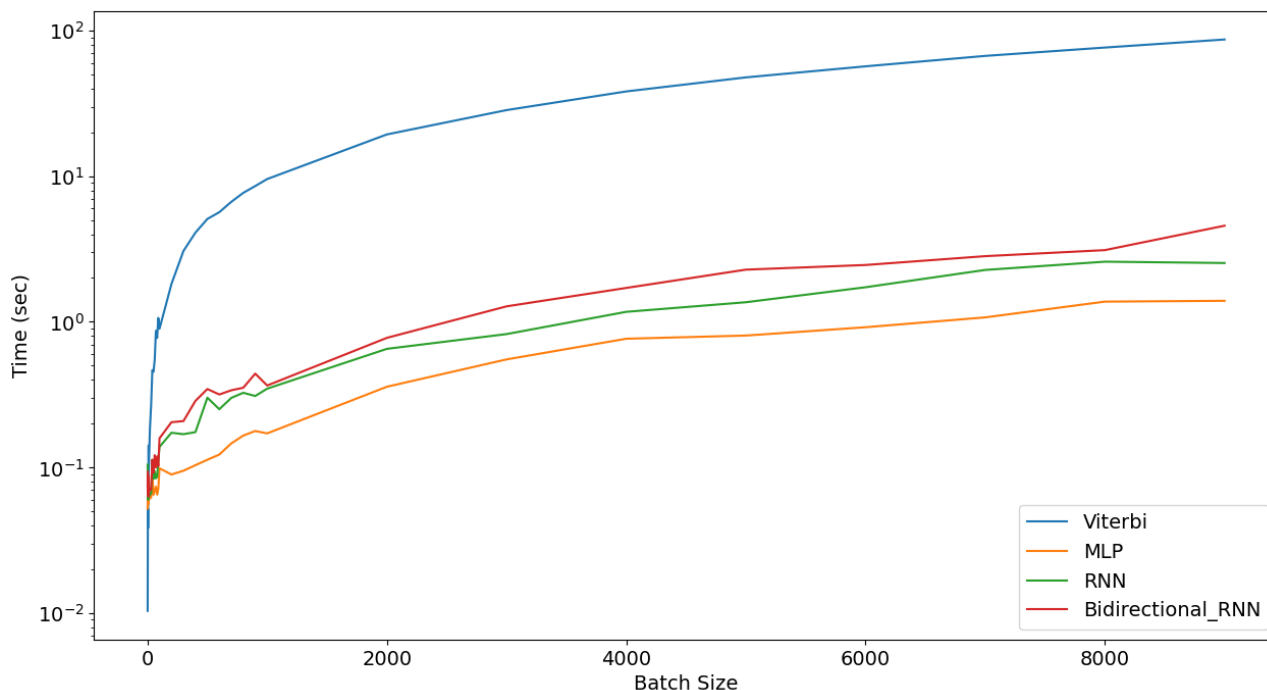


Εικόνα 6.15 Χρόνοι εκτέλεσης αποκωδικοποίησης για διάφορες αρχιτεκτονικές αποκωδικοποιητή (Χρήση GPU)

Όπως φαίνεται από την παραπάνω εικόνα, ο χρόνος εκτέλεσης της διαδικασίας αποκωδικοποίησης για τον αποκωδικοποιητή Viterbi ανεβαίνει δραματικά σε σχέση με τον αντίστοιχους χρόνους για τα νευρωνικά δίκτυα καθώς ανεβαίνει το μέγεθος των δεσμών.

Ο λόγος για τον οποίο παρατηρείται αυτή η διαφορά είναι επειδή δομή των νευρωνικών δικτύων είναι τέτοια που να ευνοεί την παράλληλη επεξεργασία. Αντιθέτως, ο αλγόριθμος Viterbi έχει κατ' ανάγκην σειριακό χαρακτήρα, οπότε δεν επιτρέπει την παραλληλοποίηση των διεργασιών, και ο χρόνος εκτέλεσης του για ένα μήνυμα εξαρτάται από το μήκος της ακολουθίας που έχει να επεξεργαστεί. Αυτή η διαφορά είναι που προξενεί αυτό το χάσμα μεταξύ των χρόνων εκτέλεσης.

Στην Εικόνα 6.16 απεικονίζεται και πάλι ο χρόνος εκτέλεσης συναρτήσεων του μεγέθους της δέσμης των μηνυμάτων. Η διαφορά σε αυτό το γράφημα είναι πως έχει αυξηθεί το εύρος των μεγεθών και πως ο άξονας των χρόνων είναι λογαριθμικός.



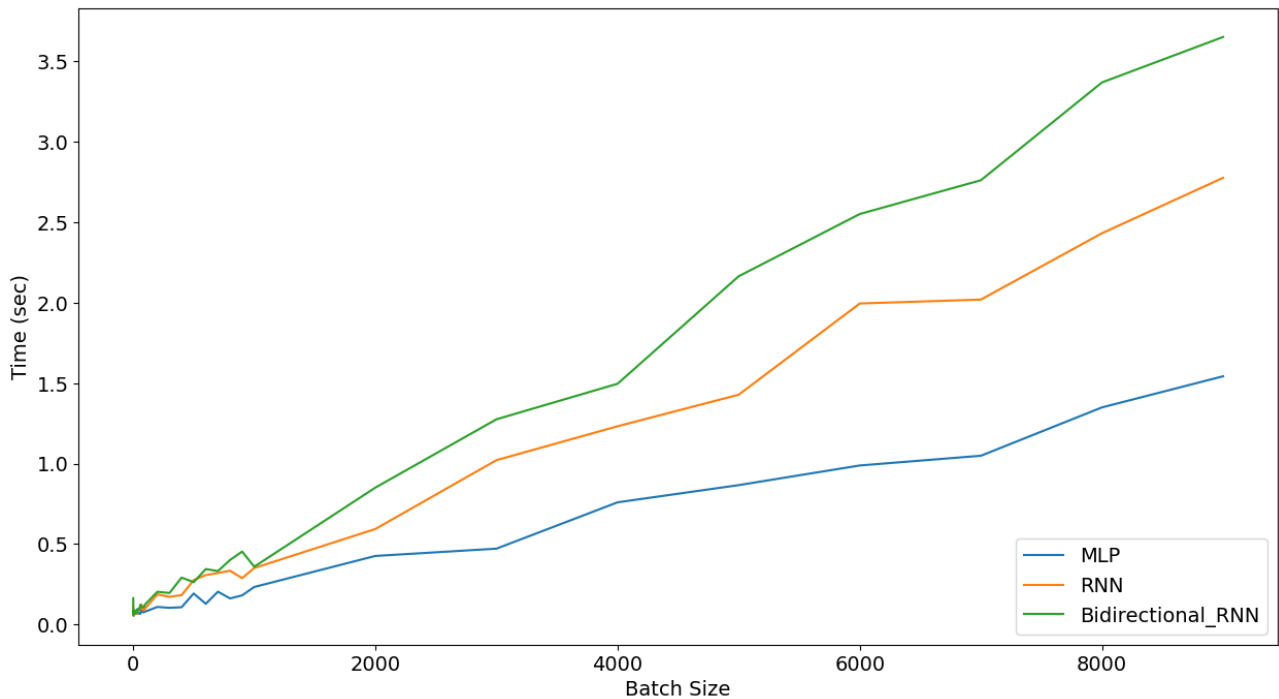
Εικόνα 6.16 Χρόνοι εκτέλεσης αποκωδικοποίησης για διάφορες αρχιτεκτονικές αποκωδικοποιητή (Λογαριθμικός άξονας χρόνων) (Χρήση GPU)

Για πιο μεγάλες δέσμες μεγεθών, το χάσμα μεταξύ του χρόνου αποκωδικοποίησης των νευρωνικών και του Viterbi, είναι περίπου ίσο με 2 τάξεις μεγέθους. Επίσης, όσον αφορά τα νευρωνικά, το πιο αργό φαίνεται να είναι το RNN διπλής κατεύθυνσης. Το πιο γρήγορο είναι το MLP, ενώ η καμπύλη που αντιστοιχεί στο RNN βρίσκεται μεταξύ των δύο.

Ο λόγος που τα MLP είναι πιο γρήγορα από τα RNN είναι η διαφορά στην δομή τους. Τα MLP μπορούν να επεξεργαστούν πολλαπλές εισόδους παράλληλα. Αυτό τα κάνει πιο κατάλληλα για παραλληλοποίηση διεργασιών. Από την άλλη, τα RNN, ειδικά τα παραδοσιακά, είναι σειριακά από την φύση τους, πράγμα το οποίο περιορίζει την δυνατότητα παραλληλοποίησης. Βέβαια η χρήση ειδικών τύπων RNN, όπως το LSTM και το GRU, διευκολύνει την παραλληλοποίηση. Αυτές οι αρχιτεκτονικές εισάγουν τη χρήση πυλών, οι οποίες βοηθούν στην παραλληλοποίηση.

Ακόμη, το αμφίδρομο RNN είναι λίγο πιο αργό από το RNN μονής κατεύθυνσης. Αυτό μπορεί να οφείλεται στο γεγονός, ότι έχουν περισσότερες συνάψεις από τα μονοκατευθυντήρια. Όπως και να έχει, παραμένουν αρκετά γρηγορότερα από τον αλγόριθμο Viterbi στην επεξεργασία μεγάλων δεσμών μηνυμάτων.

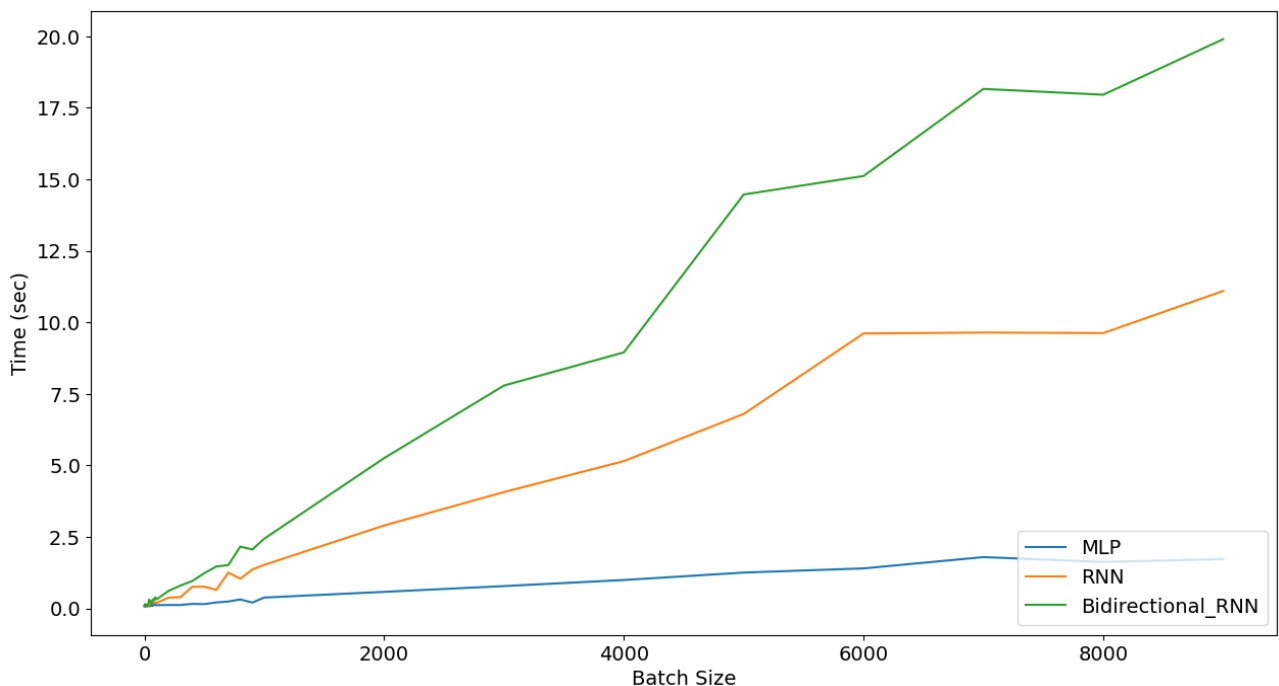
Για να φανεί πιο καθαρά η διαφορά στους χρόνους εκτέλεσης μεταξύ των νευρωνικών δικτύων, στο γράφημα της Εικόνας 6.17 παρουσιάζονται μόνο οι καμπύλες που αντιστοιχούν νευρωνικά.



Εικόνα 6.17 Χρόνοι εκτέλεσης αποκωδικοποίησης για διάφορες αρχιτεκτονικές νευρωνικών δικτύων (Χρήση GPU)

Να σημειωθεί πως αυτές οι αιχμές στα γραφήματα μπορεί να οφείλονται σε πολλούς λόγους (ανταγωνισμός για πόρους, ελευθέρωση μνήμης GPU, κλπ.). Για να περιοριστεί αυτό το φαινόμενο, έγινε εκτέλεση πολλές φορές και πάρθηκε ο μέσος όρος των χρόνων για κάθε μέγεθος δέσμης.

6.6 Χρήση CPU και χρήση GPU



Εικόνα 6.18 Χρόνοι εκτέλεσης αποκωδικοποίησης για διάφορες αρχιτεκτονικές νευρωνικών δικτύων (Χρήση CPU)

Για τις μετρήσεις την ενότητας 6.5, χρησιμοποιήθηκε μια μονάδα GPU, παρεχόμενη από το Google Colaboratory, σαν μονάδα επεξεργασίας. Στην Εικόνα 6.18 απεικονίζεται πάλι μια γραφική αναπαράσταση του χρόνου, που χρειάστηκαν τα νευρωνικά δίκτυα για να επιτελέσουν την διαδικασία αποκωδικοποίησης, συναρτήσει του μεγέθους της δέσμης των μηνυμάτων, αλλά αυτή τη φορά με τη χρήση CPU σαν μονάδα επεξεργασίας. Οι χρόνοι εκτέλεσης της διαδικασίας αποκωδικοποίησης διαφέρουν. Η μονάδα CPU αποδεικνύεται πιο αργή από την μονάδα GPU στην επεξεργασία μηνυμάτων με τη χρήση νευρωνικών δικτύων. Αν και δεν παρατίθεται στο γράφημα, ο αλγόριθμος Viterbi παρουσιάζει περίπου ίδιους χρόνους εκτέλεσης και με τις δύο μονάδες επεξεργασίας. Η μονάδα CPU εμφανίζει καθυστέρηση σε σχέση με την μονάδα GPU και στην διαδικασία εκπαίδευσης των νευρωνικών δικτύων.

Αυτό το χρονικό χάσμα στην παραγωγή προβλέψεων με τη χρήση νευρωνικών δικτύων οφείλεται σε δομικά χαρακτηριστικά των δύο μονάδων επεξεργασίας. Μια μονάδα GPU είναι φτιαγμένη με τέτοιο τρόπο ώστε να εξειδικεύεται στην παραλληλοποίηση διεργασιών. Για παράδειγμα μια μονάδα CPU έχει λίγους πανίσχυρους και ευέλικτους πυρήνες, ενώ μια μονάδα GPU έχει πολύ περισσότερους αλλά απλούστερους πυρήνες. Ακόμη, μια μονάδα GPU είναι εξοπλισμένη με μνήμη ικανή για υψηλές ταχύτητες μεταφοράς δεδομένων, το οποίο βοηθά στην επεξεργασία μεγάλων ποσοτήτων δεδομένων. Είναι σύνηθες κατά την εκπαίδευση και την τέλεση προβλέψεων με χρήση νευρωνικών δικτύων να γίνεται επεξεργασία μεγάλης ποσότητας δεδομένων. Δημοφιλείς βιβλιοθήκες βαθιάς μάθησης (deep learning libraries), όπως η TensorFlow και η PyTorch, εκμεταλλεύονται τις δυνατότητες παράλληλης επεξεργασίας, που παρέχουν οι μονάδες GPU, για να διευκολύνουν τους προγραμματιστές να αξιοποιήσουν πλήρως την επεξεργαστική τους ισχύ.

7 Συμπεράσματα

7.1 Συμπεράσματα από την παρούσα εργασία

Στην παρούσα εργασία εξετάστηκε η ικανότητα τριών θεμελιωδών αρχιτεκτονικών νευρωνικών δικτύων να αποκωδικοποιούν μηνύματα, τα οποία έχουν κωδικοποιηθεί με τη χρήση συνελκτικών κωδίκων. Οι αρχιτεκτονικές, οι οποίες εξετάστηκαν είναι τα νευρωνικά δίκτυα πρόσθιας τροφοδότησης ή αλλιώς Perceptron πολλών επιπέδων (MLPs), τα αναδρομικά νευρωνικά δίκτυα (RNNs) και τα αναδρομικά νευρωνικά δίκτυα διπλής κατεύθυνσης (Bidirectional RNNs). Η σχεδίαση και η εκπαίδευση έγινε με την βοήθεια της βιβλιοθήκης μηχανικής μάθησης TensorFlow.

Ως διαδικασία μάθησης επιλέχθηκε η επιβλεπόμενη μάθηση. Κατά την επιβλεπόμενη μάθηση, η προσαρμογή των συναπτικών βαρών γίνεται με σκοπό την μίμηση της λειτουργίας ενός εκπαιδευτή. Ως εκπαιδευτής επιλέχθηκε ένας δημοφιλής αποκωδικοποιητής συνελκτικών κωδίκων, ο αποκωδικοποιητής Viterbi.

Αναπτύχθηκε κώδικας, ο οποίος προσομοιώνει την λειτουργία ενός πλήρους ψηφιακού τηλεπικοινωνιακού συστήματος. Ο κώδικας αυτός χρησιμοποιήθηκε για την συλλογή των δεδομένων εκπαίδευσης καθώς και για την συλλογή μετρήσεων, οι οποίες αφορούσαν την λειτουργία των νευρωνικών δικτύων ως αποκωδικοποιητής συνελκτικών σημάτων. Από την μελέτη και την αξιολόγηση των μετρήσεων, οι οποίες έγιναν με τη βοήθεια του κώδικα που αναπτύχθηκε στα πλαίσια αυτής της διπλωματικής εργασίας, προέκυψαν τα εξής συμπεράσματα:

- Η επιβλεπόμενη μάθηση φαίνεται να έχει ικανοποιητικά αποτελέσματα.
- Επαληθεύτηκε πως η χρήση ενός αποκωδικοποιητή, ο οποίος χρησιμοποιεί τον αλγόριθμο Viterbi, ως εκπαιδευτής αυτών των δικτύων είναι μια ενδεδειγμένη επιλογή, καθώς είναι ο βέλτιστος τρόπος αποκωδικοποίησης συνελκτικά κωδικοποιημένων σημάτων.
- Η αρχιτεκτονική, η οποία προσεγγίζει τις επιδόσεις του αλγορίθμου Viterbi ως προς το ρυθμό σφαλμάτων (BER) είναι τα αναδρομικά δίκτυα διπλής κατεύθυνσης. Αυτό συμβαίνει λόγω της δομής τους, η οποία τους επιτρέπει για κάθε ψηφίο της πρόβλεψής τους, να λαμβάνουν υπόψιν ολόκληρη την ακολουθία εισόδου.
- Τα απλά δίκτυα πρόσθιας τροφοδότησης, αν και γρήγορα, εύκολα υλοποιήσιμα και αρκετά δημοφιλή, δεν φαίνεται να είναι κατάλληλα για μια διεργασία όπως η αποκωδικοποίηση συνελκτικά κωδικοποιημένων σημάτων. Αυτό συμβαίνει γιατί η αρχιτεκτονική τους δεν περιέχει κάποιον εσωτερικό μηχανισμό μνήμης.

- Για να επιτευχθεί η διαδικασία εκπαίδευσης των νευρωνικών δικτύων, πρέπει πρώτα να συλλεγεί ένα σύνολο δεδομένων εκπαίδευσης. Το σύνολο δεδομένων πρέπει να έχει ένα επαρκές μέγεθος, έτσι ώστε το εκπαιδευόμενο νευρωνικό δίκτυο να μπορέσει να προσαρμόσει κατάλληλα τα συναπτικά του βάρη. Στην περίπτωση που το μέγεθος του συνόλου δεδομένων εκπαίδευσης είναι μικρό, το μοντέλο μηχανικής μάθησης θα πάσχει από το φαινόμενο της υποπροσαρμογής και δε θα προσεγγίζει επαρκώς τη λειτουργία του αποκωδικοποιητή Viterbi. Η συλλογή του συνόλου δεδομένων μπορεί να είναι πολλές φορές πιο «επίπονη» υπολογιστικά από την διαδικασία εκπαίδευσης και πρόβλεψης.
- Υπάρχουν πολλοί τρόποι μορφοποίησης των δεδομένων εισόδου του νευρωνικού δικτύου. Στα πλαίσια αυτής της εργασίας δοκιμάστηκαν οι μέθοδοι One Hot Encoding, Binary Representation και Normalization. Δεν παρατηρήθηκε σημαντική διαφορά στις επιδόσεις του εκπαιδευόμενου νευρωνικού δικτύου. Ίσως να υπάρχει μια πολύ μικρή υπεροχή των μεθόδων Binary Representation και Normalization. Βέβαια, είναι σίγουρο πως η μέθοδος One Hot Encoding δεσμεύει περισσότερο χώρο στη μνήμη από τις άλλες δύο.
- Για την αποκωδικοποίηση μεγάλων δεσμών ληφθέντων μηνυμάτων, τα νευρωνικά δίκτυα είναι πολύ πιο γρήγορα από τον παραδοσιακό αλγόριθμο Viterbi. Αυτή η διαφορά στον χρόνο επεξεργασίας έγκειται στο γεγονός ότι τα νευρωνικά δίκτυα παρέχουν την δυνατότητα παραλληλοποίησης διεργασιών, εν αντιθέσει με τον αλγόριθμο Viterbi, ο οποίος είναι από την φύση του σειριακός.
- Οι μονάδες επεξεργασίας, οι οποίες εξειδικεύονται στην παραλληλοποίηση διεργασιών, όπως η GPU, εμφανίζουν καλύτερες επιδόσεις σε διεργασίες σχετικές με την μηχανική μάθηση, όπως η εκπαίδευση νευρωνικών δικτύων και η χρήση εκπαιδευμένων μοντέλων για την τέλεση προβλέψεων.

7.2 Μελλοντικές εργασίες

Αν και η λειτουργία του αναδρομικού νευρωνικού δικτύου διπλής κατεύθυνσης ως αποκωδικοποιητής συνελκτικών σημάτων ήταν παραπάνω από ικανοποιητική, υπάρχουν μερικά ζητήματα που θα μπορούσαν να μελετηθούν από μελλοντικές εργασίες:

- Στην τρέχουσα εργασία, σε πειραματικό επίπεδο, μελετάται η αποκωδικοποίηση μηνυμάτων, τα οποία έχουν σταλεί από έναν (2,1,3) συνελκτικό κωδικοποιητή. Ενδεχομένως, θα μπορούσε να γίνει μια γενίκευση, η οποία θα επέτρεπε στον σχεδιαστή/χρήστη να καθορίζει ο ίδιος τον τύπο του συνελκτικού κωδικοποιητή.

- Η δομή των νευρωνικών δικτύων, τα οποία χρησιμοποιήθηκαν, και η διαδικασία εκπαίδευσης τους αποσκοπούν στο να εκπαιδεύσουν το νευρωνικό δίκτυο να αποκωδικοποιεί μηνύματα συγκεκριμένου μήκους. Ενδεχομένως, θα μπορούσε να σχεδιαστεί και να εκπαιδευτεί νευρωνικό δίκτυο το οποίο να είναι ικανό να επεξεργάζεται μηνύματα μεταβλητού μήκους.

8 Βιβλιογραφία

- [1] R. G. Gallager, Principles of Digital Communication, Cambridge Univesity Press, 2008.
- [2] A. J. Viterbi και J. K. Omura, Principles of Digital Communication and Coding, Dover Publications, 2009.
- [3] R. Johannesson και K. S. Zizangirov, Fundamentals of Convolutional Coding, Wiley, 2015.
- [4] B. Sklar και P. K. Ray, Digital Communications, Pearson Education, 2014.
- [5] J. G. Proakis και M. Salehi, Digital Communications, McGraw-Hill, 2008.
- [6] S. Lin και D. J. Costello, Error Control Coding, Pearson Education, 2004.
- [7] N. Shlezinger, Y. C. Eldar, N. Farsad και A. J. Goldsmith, ViterbiNet: Symbol Detection Using a Deep Learning Basic Algorithm.
- [8] R. H. Kuehne, A. Iqbal και J. Gall, NeuralNetwork-Viterbi: A Framework for Weakly Supervised Video Learning, 2018.
- [9] I. Goodfellow, Y. Bengio και A. Courville, Deep Learning, The MIT Press, 2016.
- [10] C. C. Aggarwal, Neural Networks and Deep Learning, Springer International Publishing, 2018.
- [11] S. Ravichandiran, Hands-On Deep Learning Algorithms with Python, Packt Publishing, 2019.
- [12] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, 1999.
- [13] A. B. Downey, Think Python: How to Think Like a Computer Scientist, O' Reilly, 2015.
- [14] «Επίσημη ιστοσελίδα βιβλιοθήκης TensorFlow,» [Ηλεκτρονικό]. Available: https://www.tensorflow.org/api_docs/python/tf.
- [15] B. Caulfield, «What's the Difference Between a CPU and a GPU?,» 16 12 2009. [Ηλεκτρονικό]. Available: <https://blogs.nvidia.com/blog/whats-the-difference-between-a-cpu-and-a-gpu/>.