



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής-Ανάπτυξη Λογισμικού και Τεχνητής Νοημοσύνης»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Διάσχιση λαβυρίνθου με την χρήση τεχνητών νευρωνικών δικτύων. Maze solving with artificial neural networks.
Όνοματεπώνυμο Φοιτητή	Φώτιος Σκουμπουρδής
Πατρώνυμο	Δημοσθένης
Αριθμός Μητρώου	ΜΠΣΠ18022
Επιβλέπων	Διονύσιος Σωτηρόπουλος , Επίκουρος Καθηγητής

Μάρτιος 2021

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Διονύσιος
Σωτηρόπουλος
Επίκουρος
Καθηγητής

Ευάγγελος
Σακκόπουλος
Επίκουρος
Καθηγητής

Γεώργιος
Τσιχριντζής
Καθηγητής

Περιεχόμενα

Περίληψη.....	5	
ΚΕΦΑΛΑΙΟ 1- ΒΙΟΛΟΓΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ		
1.1 Εισαγωγή στα νευρωνικά δίκτυα.....	7	
1.2 Ο ανθρώπινος εγκέφαλος	9	
1.3 Ο βιολογικός νευρώνας	10	
1.4 Η ανατομία και η ιστολογία του νευρώνα του βιολογικού νευρώνα	12	
1.5 Συστατικά και δομή του νευρώνα	14	
1.6 Τύποι των νευρώνων	15	
1.7 Συνδεσιμότητα των νευρώνων	16	
1.8 Η λειτουργία των νευρώνων	18	
ΚΕΦΑΛΑΙΟ 2 -ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ		
2.1 Τι είναι ένα τεχνητό νευρωνικό δίκτυο (ΤΝΔ)	21	
2.2 Ιστορική αναδρομή.....	22	
2.3 Το μοντέλο του τεχνητού νευρώνα	26	
2.4 Νευρωνικά δίκτυα πολλαπλών επιπέδων.....	28	
2.5 Επαναληπτικά νευρωνικά δίκτυα.....	29	
2.6 Λειτουργία του νευρωνικού δικτύου	30	
2.7 Τα Τεχνητά Νευρωνικά Δίκτυα ως κατευθυνόμενοι γράφοι	36	
2.8 Ομοιότητα ΤΝΔ με τον ανθρώπινο εγκέφαλο	40	
2.9 Αντιστοιχία Βιολογικών και Τεχνητών Νευρώνων.....	41	
2.10 Πλεονεκτήματα των νευρωνικών δικτύων.....	42	
2.11 Μειονεκτήματα των νευρωνικών δικτύων.....	45	
ΚΕΦΑΛΑΙΟ 3 – ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ		
3.1 Τι είναι μάθηση.....	47	
3.2 Τι είναι η μηχανική μάθηση.....	48	
3.3 Είδη μηχανικής μάθησης.....	50	
3.4 Μάθηση - Εκπαίδευση νευρωνικών δικτύων.....	52	
ΚΕΦΑΛΑΙΟ 4 – ΕΦΑΡΜΟΓΕΣ ΤΩΝ ΤΕΧΝΗΤΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΣΤΗ ΔΙΑΣΧΥΣΗ ΛΑΒΥΡΙΝΘΩΝ		
4.1 Διάσχιση λαβυρίνθου με την χρήση νευρωνικού δικτύου.....	68	
4.2 Διάσχιση λαβυρίνθου με την χρήση της απόστασης manhatan.....	93	
ΚΕΦΑΛΑΙΟ 5– ΑΛΛΕΣ ΕΦΑΡΜΟΓΕΣ ΤΩΝ ΤΕΧΝΗΤΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ.....		106
Επίλογος.....	111	
Βιβλιογραφία.....	112	
Παράρτημα-Κώδικες Εφαρμογών.....	115	

ΠΕΡΙΛΗΨΗ

Στην παρούσα πτυχιακή εργασία γίνεται μία ευρεία περιγραφή των τεχνητών νευρωνικών δικτύων , έναν επιστημονικό κλάδο που έχει βρεί μεγάλη ανάπτυξη τα τελευταία χρόνια . Η θεματολογία υπάγεται στην περιοχή της Τεχνητής Νοημοσύνης, η οποία μπορεί να επιλύσει πολλά προβλήματα διαφόρων κλάδων όπως είναι η υγεία, η οικονομία , η μετεωτολογία , η αναγνώριση προτύπων και οι συγκοινωνίες .

και αναμένεται να παίξει καθοριστικό ρόλο στην τεχνολογία του μέλλοντος. Τεχνητή Νοημοσύνη είναι η δημιουργία μηχανών οι οποίες να μιμούνται την ανθρώπινη συμπεριφορά. Στην παρούσα εργασία παρουσιάζεται η χρήση των νευρωνικών δικτύων για το πρόβλημα της διάσχισης των λαβυρίθων.

Ειδικότερα :

Το πρώτο κεφάλαιο περιγράφει τους βιολογικούς νευρώνες και τα βιολογικά νευρωνικά δίκτυα. Αυτό είναι σημαντικό γιατί τα τεχνητά νευρωνικά δίκτυα έχουν βιολογική έμπνευση.

Το δεύτερο κεφάλαιο περιλαμβάνει μία εκτενή ανάλυση του τεχνητού νευρώνα, των τεχνητών νευρωνικών δικτύων. Επίσης αναλύει τις ομοιότητες , και τις διαφορές , μεταξύ των τεχνητών νευρωνικών δικτύων και βιολογικών νευρωνικών δικτύων και παρουσιάζει τα πλεονεκτήματα και τα μειονεκτήματα την τεχνητών νευρωνικών δικτύων.

Το τρίτο κεφάλαιο περιγράφει την διαδικασία της εκπαίδευσης των τεχνητών νευρωνικών δικτύων με τον αλγόριθμο της οπισθοδιάδοσης.

Το τέταρτο κεφάλαιο περιγράφει τη διάσχιση ενός λαβυρίνθου με την χρήση τεχνητού νευρωνικού δικτύου. Το πέμπτο κεφάλαιο περιγράφει τομείς στους οποίους τα τεχνητά νευρωνικά δίκτυα έχουν βρεί μεγάλη εφαρμογή.

ABSTRACT

This dissertation provides a broad description of artificial neural networks, a science that has grown significantly in recent years. The topics fall under the area of Artificial Intelligence, which can find solutions to many problems in various fields such as health, economics, meteorology, pattern recognition and transportation. and is expected to play a key role in the technology of the future. Artificial Intelligence is the creation of machines that mimic human behavior. This paper presents the use of neural networks for the problem of labyrinth crossing.

Particularly :

The first chapter describes biological neurons and biological neural networks. This is important because artificial neural networks are biologically inspired.

The second chapter includes an extensive analysis of the artificial neuron, the artificial neural networks. It also analyzes the similarities, and differences, between artificial neural networks and biological neural networks and presents the advantages and disadvantages of artificial neural networks.

The third chapter describes the process of training artificial neural networks with the regression propagation algorithm.

The fourth chapter describes the crossing of a maze using an artificial neural network.

The fifth chapter describes areas in which artificial neural networks have found great application.

ΚΕΦΑΛΑΙΟ 1- ΒΙΟΛΟΓΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

1.1 Εισαγωγή στα νευρωνικά δίκτυα

Τα νευρωνικά δίκτυα (neural networks) αποτελούν μια σχετικά νέα περιοχή στις φυσικές επιστήμες, καθ' όσον έχουν γίνει γνωστά και έχουν αναπτυχθεί μόνο κατά τα τελευταία σαράντα χρόνια περίπου. Η μελέτη τους ξεκίνησε τις τελευταίες δεκαετίες αλλά η μεγάλη ώθηση δόθηκε από το 1980 και μετά. Εντούτοις, η περιοχή αυτή έχει γνωρίσει μια μεγάλη άνθηση, κρίνοντας την μεγάλη ανάπτυξη που έχει παρατηρηθεί, από τον αριθμό των επιστημόνων που ασχολούνται με αυτά τα θέματα, από τον αριθμό των εφαρμογών που έχουν βασιστεί σε αυτά και βέβαια από τα σημαντικά επιτεύγματα, που έχουν συμβάλει στο να γίνουν γνωστά σε ένα ευρύτερο κύκλο. (1. <http://kelifos.physics.auth.gr/> -Α.Π.Θ.) Επομένως αποτελούν ένα τομέα με ιδιαίτερο ενδιαφέρον για τις τεχνολογικές επιστήμες και ιδιαίτερα για την επιστήμη της πληροφορικής.

Το βασικό χαρακτηριστικό τους είναι ότι οι βασικές αρχές και λειτουργίες τους βασίζονται στο νευρικό σύστημα των ζωντανών οργανισμών. Η μελέτη τους και η χρήση τους έχει προχωρήσει πολύ πιο πέρα από τους βιολογικούς οργανισμούς και σήμερα τα νευρωνικά δίκτυα χρησιμοποιούνται για την επίλυση πολλών ειδών προβλημάτων με την χρήση των ηλεκτρονικών υπολογιστών. Η φιλοσοφία τους όμως είναι διαφορετική από τον τρόπο με τον οποίο δουλεύουν οι υπολογιστές.

Ο τρόπος λειτουργίας τους όμως προσπαθεί να συνδυάσει τον τρόπο σκέψης και λειτουργίας του ανθρωπίνου εγκεφάλου με τον αφηρημένο μαθηματικό τρόπο σκέψης. Έτσι στα νευρωνικά δίκτυα χρησιμοποιούμε έννοιες και ιδέες όπως π.χ. κάποιο νευρωνικό δίκτυο μαθαίνει να εκπαιδεύεται, μαθαίνει να θυμάται ή μαθαίνει να ξεχνάει μία αριθμητική τιμή, έννοιες που μέχρι τώρα αποδίδουμε στην ανθρώπινη σκέψη. Βέβαια μπορούν να χρησιμοποιηθούν συναρτήσεις και έννοιες από τα μαθηματικά. (1. <http://kelifos.physics.auth.gr/> -Α.Π.Θ.)

Ένα ιδιαίτερο χαρακτηριστικό είναι ότι οι μελετητές στον τομέα των νευρωνικών δικτύων προέρχονται από αρκετές διαφορετικές επιστήμες όπως π.χ. την ιατρική, τα μαθηματικά την φυσική, την ηλεκτρολογία την πληροφορική κτλ.. Αυτό υποδεικνύει ότι για την ανάπτυξή τους, τις τεχνικές και τις μεθόδους που χρησιμοποιούνται τους απαιτούνται γνώσεις από πολλές διαφορετικές περιοχές και επιστήμες. Έτσι καταλαβαίνουμε ότι τα νευρωνικά δίκτυα αποτελούν μία πρόκληση για τον σύγχρονο επιστημονικό κόσμο και οι γνώσεις που προκύπτουν από την έρευνα και τη μελέτη τους είναι μείζονος σημασίας για τις σύγχρονες εφαρμογές. Γενικότερα είναι από τα λίγα επιστημονικά πεδία που συνδυάζουν γνώσεις από τόσο διαφορετικές επιστημονικές περιοχές με τόσο άμεσο τρόπο.

Ένα πρόβλημα το οποίο μπορεί να επιλυθεί με νευρωνικά δίκτυα είναι η διάσχιση ενός λαβυρίνθου. Στο πρόβλημα αυτό υπάρχει κάποιος πράκτορας ο οποίος ξεκινάει από κάποιο σημείο του λαβυρίνθου και πρέπει να φτάσει στην έξοδό του. Η επίλυση ενός λαβυρίνθου με τη χρήση υπολογιστών είναι μια πολύπλοκη αλλά δελεαστική εργασία, καθώς πρέπει να βρούμε έναν αλγόριθμο που να είναι αποτελεσματικός όσον αφορά το χρόνο και το διάστημα για την εύρεση της πιο σύντομης διαδρομής. Υπάρχει ένας μεγάλος αριθμός αλγορίθμων γραφημάτων για την επίλυση αυτού του προβλήματος, ο καθένας ξεπερνά τον άλλο σε πτυχές απόδοσης. Επίσης έχουν χρησιμοποιηθεί και γεννητικοί αλγόριθμοι για την επίλυση αυτού του προβλήματος δηλαδή αλγόριθμοι οι οποίοι βασίζονται στην διαδικασία της εξέλιξης. Η επίλυση αυτού του θέματος με την χρήση τεχνητών νευρωνικών είναι σημαντική γιατί αποδεικνύει τις αυξημένες δυνατότητες αυτού του κλάδου και μπορεί γενικότερα να βοηθήσει στην ανάπτυξη αντίστοιχων συστημάτων πλοήγησης. Επίσης υπάρχουν πολλές διαφορετικές προσεγγίσεις επίλυσης αυτού του προβλήματος με την βοήθεια των νευρωνικών δικτύων. Η παρούσα εργασία περιέχει δύο τέτοιες προσεγγίσεις.

Γενικότερα είναι από τα λίγα επιστημονικά πεδία που συνδυάζουν γνώσεις από τόσο διαφορετικές επιστημονικές περιοχές με τόσο άμεσο τρόπο.

Τα νευρωνικά δίκτυα βρίσκουν ευρεία εφαρμογή στους ακόλουθους τομείς:

Κατηγοριοποίηση:
Ιατρική, Άμυνα, Γεωργία, Οικονομία/επιχειρήσεις

Αναγνώριση Προτύπων:
Τράπεζες, Πληροφορική και Τηλεπικοινωνίες

Αποτίμηση:
Άμυνα, Ασφάλεια, Μηχανολογία

Πρόβλεψη Καταστάσεων:
Οικονομία/επιχειρήσεις, Γεωργία, Μετεωρολογία

1.2 Ο ανθρώπινος εγκέφαλος

Είναι γνωστό ότι ο ανθρώπινος εγκέφαλος είναι μια από τις πιο περίπλοκες δομές που συναντά κανείς στον φυσικό κόσμο . Επίσης είναι ιδιαίτερα περίπλοκος και ως όργανο που βρίσκεται σε λειτουργία. Μπορούμε να πούμε με σιγουριά ότι ακόμα και σήμερα είναι λίγο μόνο κατανοητός. Αυτό ισχύει διότι γνωρίζουμε και μπορούμε να εξηγήσουμε πολύ λίγα πράγματα σχετικά με τις λειτουργίες του και κάποια από αυτά με διάφορες υποθέσεις που αναγκαζόμαστε να κάνουμε. Προφανώς το πρώτο πράγμα που πρέπει να εξετάσουμε για να γνωρίσουμε και να κατανοήσουμε καλύτερα τη λειτουργία είναι η δομή του.

Από πειραματικά δεδομένα και παρατηρήσεις της Νευρολογίας γνωρίζουμε ορισμένα στοιχεία για τον εγκέφαλό μας π.χ. ότι αποτελείται από δύο συμμετρικά ημισφαίρια και ότι ορισμένες περιοχές του εξειδικεύονται να εκτελούν ορισμένες λειτουργίες. Επίσης έχουμε κατανοήσει και εξηγήσει έστω και εμπειρικά ορισμένες από τις λειτουργίες του, ιδιαίτερα αυτές που σχετίζονται με την επαφή μας και την αλληλεπίδρασή μας με τον εξωτερικό μας κόσμο όπως π.χ. η όραση , η όσφρηση η ακοή κτλ. Με την πρόοδο της τεχνολογίας είμαστε σε θέση να κάνουμε χειρουργικές επεμβάσεις σε τμήματα του εγκεφάλου . Παρ' όλα αυτά οι γενική λειτουργία του είναι ελάχιστα κατανοητή.

Το πιο σημαντικό είναι ότι ο εγκέφαλος είναι το όργανο του σώματος που χρησιμοποιεί την διάνοια και όλα όσα συνεπάγονται με αυτή όπως είναι η ευφυΐα , η αντίληψη , τα συναισθήματα κτλ. Κανένα τεχνητό μοντέλο δεν μπορεί να προβλέψει ανθρώπινα συναισθήματα , εμπειρίες και συμπεριφορές όπως το ανθρώπινο μυαλό. Αυτό συμβαίνει γιατί δεν έχει εξηγηθεί και αναλυθεί επαρκώς η ανατομία , η φυσιολογία και η λειτουργίες του εγκεφάλου μας που δημιουργούν τις γνωστικές εμπειρίες και την αντίληψή μας , μηχανισμοί που είναι εξαιρετικά πολύπλοκοι για να εξηγηθούν. Τέλος υπάρχουν κάποιες λειτουργίες για τις οποίες έχουν δοθεί αρκετές εξηγήσεις όπως είναι η μνήμη , η λειτουργία του ύπνου ,τα όνειρα και άλλα , αποτελούν τομείς για τους οποίους υπάρχει έντονο ερευνητικό ενδιαφέρον αλλά ακόμα είμαστε μακριά από το να έχουμε επαρκή εικόνα και γνώση για τα φαινόμενα αυτά. Η νευρολογία που μελετά το νευρικό σύστημα σε συνεργασία με άλλες επιστήμες όπως είναι η χημεία ,τα μαθηματικά , η επιστήμη των υπολογιστών η βιολογία και η γενετική προσπαθεί να αποκρυπτογραφήσει την λειτουργία του εγκεφάλου και να δώσει απαντήσεις στο πως πραγματοποιούνται οι λειτουργίες που αναφέραμε παραπάνω.

(2.What We Know About the BrainStructure -Karla Batista-García-Ramó1, and Caridad Ivette Fernández-Verdecia – Τι γνωρίζουμε για τον ανθρώπινο εγκέφαλο)

1.3 Ο βιολογικός νευρώνας

Η βασική μονάδα δόμησης του εγκεφάλου και του νευρικού συστήματος γενικότερα είναι ένα εξειδικευμένο κύτταρο που ονομάζεται νευρώνας ή νευρικό κύτταρο. Ο ανθρώπινος εγκέφαλος αποτελείται από έναν πολύ μεγάλο αριθμό νευρώνων, περίπου 10^{10} νευρώνες. Η εκτίμηση αυτή δεν είναι βέβαιη γιατί άλλες μελέτες υπολογίζουν ότι ο αριθμός των νευρώνων είναι περίπου 10^9 κύτταρα ενώ άλλες μελέτες υπολογίζουν περίπου 10^{11} νευρώνες. Οι πιο πρόσφατες μελέτες υπολογίζουν περίπου 86 δισεκατομμύρια νευρικά κύτταρα. Όλοι οι νευρώνες είναι διαφορετικοί μεταξύ τους και δεν υπάρχουν δύο ολόιδιοι νευρώνες μέσα στον ασύλληπτα μεγάλο αυτόν αριθμό.

Κάθε νευρώνας συνδέεται με πολλούς άλλους νευρώνες με εξειδικευμένες επαφές που ονομάζονται συνάψεις. Ο αριθμός των συνάψεων δεν είναι σταθερός, αλλά υπολογίζεται ότι ένας νευρώνας έχει περίπου 10.000 συνάψεις. Κάποιοι νευρώνες έχουν μέχρι και 200.000 συνάψεις περίπου, όπως είναι οι νευρώνες Purkinje οι οποίοι βρίσκονται στην παρεγκεφαλίδα. Πολλές από τις συνδέσεις φαίνεται εκ πρώτης όψεως ότι είναι τυχαίες, ή ότι έχουν στατιστικό χαρακτήρα. Το πιθανότερο είναι ότι έχουν δημιουργηθεί με μεγάλη ακρίβεια τόσο στο επίπεδο από κύτταρο προς κύτταρο τόσο, όσο και σε επίπεδο ολόκληρου του νευρικού συστήματος.

Ένας αριθμός νευρώνων με τις διασυνδέσεις του αποτελούν ένα νευρωνικό δίκτυο (neural network). Το όλο σύστημα όλων των ανθρώπινων νευρωνικών δικτύων που υπάρχουν στον ανθρώπινο οργανισμό ονομάζεται Κεντρικό Νευρικό Σύστημα (Central Neural System). Το σύστημα αυτό εκτείνεται σε όλο το ανθρώπινο σώμα με κεντρικά σημεία την σπονδυλική στήλη και τον εγκέφαλο. Οι νευρώνες βέβαια εκτείνονται μέχρι όλα τα άνω και τα κάτω άκρα. Τα μεγέθη των αριθμών και των συνάψεων του στο νευρικό σύστημα υποδεικνύουν την περιπλοκότητα του εγκεφάλου, αλλά και τις τεράστιες δυνατότητες που παρουσιάζει. Οι νευρώνες ως κύτταρα πιστευόταν ότι δεν πολλαπλασιάζονται και δεν αναπαράγονται όπως συμβαίνει με τα υπόλοιπα κύτταρα του σώματος. Με βάση αυτήν την θεωρία το κεντρικό νευρικό σύστημα στο σύνολό του δημιουργείται στο έμβρυο τις πρώτες ημέρες της κύησής του και είναι τελείως ανεπτυγμένο λίγους μήνες πριν από την γέννηση του οργανισμού και ο εγκέφαλος είναι ένα όργανο χωρίς αναπαραγωγικές

δυνατότητες. Η ανωτέρω θεώρηση ήταν αποδεκτή μέχρι πριν δύο δεκαετίες περίπου γιατί στις μέρες έχει αποδειχθεί ότι τα νευρικά κύτταρα πολλαπλασιάζονται όπως και τα υπόλοιπα κύτταρα του σώματός μας και ότι ο ρυθμός αυτής της αναγέννησης επηρεάζεται πολύ από τις συνθήκες που επικρατούν στο περιβάλλον μας. Αυτά τα ευρήματα μας επιτρέπουν να εξετάσουμε από μια άλλη σκοπιά την πιθανή αντιμετώπιση εκφυλιστικών ασθενειών του νευρικού συστήματος όπως η νόσος του Πάρκινσον και του Αλτςχάιμερ. Ο εγκέφαλος ενός υγιούς ενήλικου ανθρώπου χάνει περίπου 1000 νευρώνες ημερησίως. Έχει διαπιστωθεί ότι μεγαλύτερος αριθμός νευρώνων καταστρέφεται από τι αλκοόλ, τα ναρκωτικά και βέβαια από την προχωρημένη ηλικία.

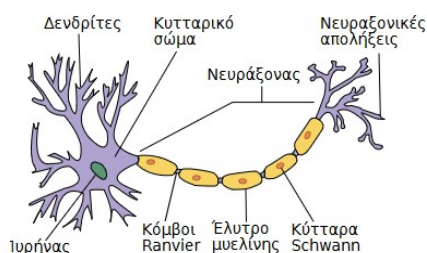
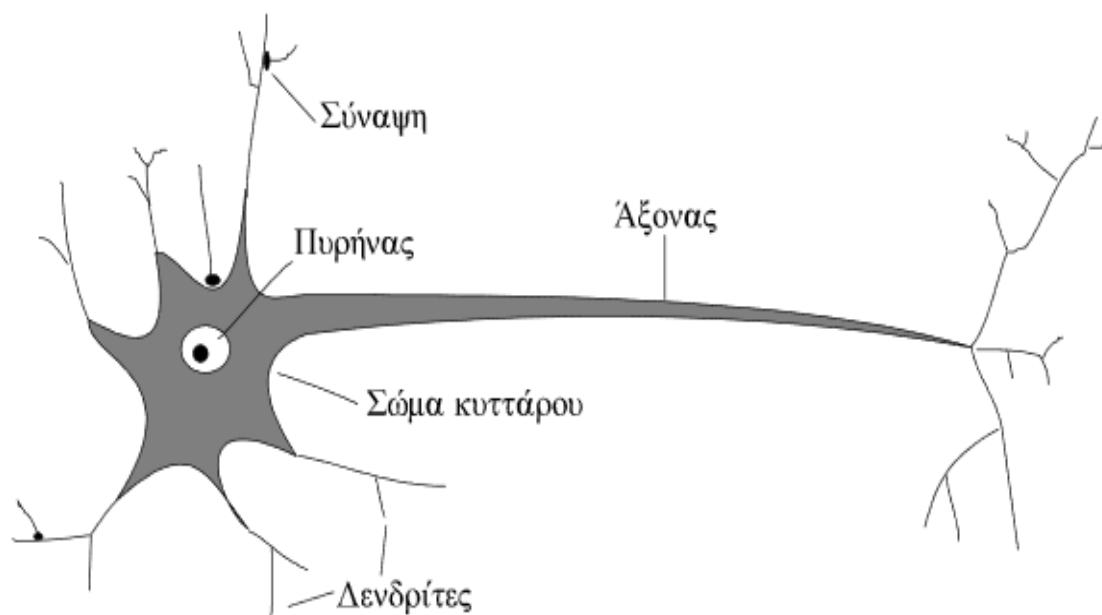
Η παύση της αναπαραγωγής των νευρώνων δεν συνεπάγεται και με παύση της αναπαραγωγής των συνάψεων. Μάλιστα σε όλη την διάρκεια της ζωής ενός οργανισμού οι συνάψεις βρίσκονται σε μία δυναμική ισορροπία δημιουργώντας καινούργιες και καταστρέφοντας παλιές. Η δημιουργία των νέων συνάψεων γίνεται όταν ο εγκέφαλός μας αποκτά περισσότερες εμπειρίες σε σχέση με την αλληλεπίδρασή του από το περιβάλλον, δηλαδή όταν μαθαίνει, αναγνωρίζει, κατανοεί, θυμάται κτλ. Βέβαια οι σοβαρές εκφυλιστικές ασθένειες τις τρίτης ηλικίας προέρχονται περισσότερο από την καταστροφή των συνάψεων στα δίκτυα του κεντρικού νευρικού συστήματος και κυρίως του εγκεφάλου παρά στην καταστροφή των νευρικών κυττάρων.

Ο ρόλος του νευρώνα σε ένα νευρωνικό δίκτυο είναι να λαμβάνει όλα τα σήματα που έρχονται από άλλους νευρώνες, να τα επεξεργάζεται με κατάλληλο τρόπο, και να μεταδίδει το επεξεργασμένο σήμα σε άλλους νευρώνες, έτσι ώστε ένα σήμα να διαδίδεται μέσω ενός τεραστίου αριθμού νευρώνων. Οι νευρώνες έχουν την δυνατότητα να αντιδρούν σε συγκεκριμένες, μεταβολές του περιβάλλοντος όπως είναι η μεταβολή της θερμοκρασίας, η μεταβολή της πίεσης, η μεταβολή της έντασης του φωτός και η μεταβολή του pH. Τα σήματα που επεξεργάζεται ένας νευρώνας είναι ηλεκτρικής μορφής και είναι της τάξης μερικών mV. Αξίζει να αναφέρουμε στις ημέρες μας είναι αποδεκτή η θεώρηση ότι διάφορα τμήματα του εγκεφάλου εξειδικεύονται σε διαφορετικές εγκεφαλικές λειτουργίες όπως είναι η όραση, η αφή, η ακοή, η όσφρηση, η αφή δηλαδή οι αισθήσεις μας. Ο χάρτης του Penfield στο σχήμα 1.1 δίνει παραστατικά τον έλεγχο των διαφόρων τμημάτων από τον εγκέφαλο. (1. <http://kelifos.physics.auth.gr/COURSES-A.Π.Θ.>)

Το σώμα που είναι το μεγαλύτερο κεντρικό τμήμα του κυττάρου μεταξύ των δενδριτών και του νευροάξονα και περιλαμβάνει το πυρήνα του κυττάρου.

Τον νευροάξονα που ονομάζεται και νευρίτης ή απλά άξονας, μια λεπτή ίνα που μπορεί να είναι μέχρι και δεκάδες χιλιάδες φορές μεγαλύτερη σε μήκος από τη διάμετρο του σώματος. Αυτή η δομή μεταφέρει τα νευρικά σήματα από τον νευρώνα. Κάθε νευρώνας διαθέτει μόνο ένα άξονα, αλλά αυτός ο άξονας μπορεί να διακλαδίζεται έντονα, με αποτέλεσμα να επιτυγχάνεται η επικοινωνία με πολλούς άλλους νευρώνες. Εκεί που τελειώνει ο άξονας υπάρχουν τα τελικά κομβία που χρειάζονται για την μεταφορά των πληροφοριών προς άλλους δενδρίτες άλλων νευρώνων. Οι περισσότεροι νευράξονες είναι μονωμένοι με έλυτρα μυελίνης.

Τους δενδρίτες σχετικά κοντές διακλαδιζόμενες κυτταρικές προεξοχές ή αποφυάδες. Αυτά φαίνονται στα παρακάτω σχήματα, στο σχήμα 1.2 και στο σχήμα 1.3.



Σχήμα 1.2

Σχήμα 1.3

Ο κυρίως κορμός του νευρώνα είναι το σώμα, μέσα στο οποίο βρίσκεται ο πυρήνας του κυττάρου. Ο πυρήνας του κυττάρου είναι ευμεγέθης και σφαιρικός και στον πυρήνα του κυττάρου βρίσκεται όλο το γενετικό υλικό (dna) του οργανισμού. Εδώ πραγματοποιείται η πιο έντονη χημική δράση του κυττάρου που απαιτείται για την σύνθεση των ενζύμων των πρωτεϊνών και άλλων μορίων που είναι απαραίτητα για την ζωή και την λειτουργία του κυττάρου. Ο άξονας είναι μία μεγάλη επέκταση από το σώμα του κυττάρου και εφάπτεται με άλλους νευρώνες. Ο κάθε νευρώνας έχει μόνο έναν άξονα που μεταδίδει σήματα σε άλλους νευρώνες δηλαδή αποστέλλει τα εξερχόμενα σήματα. Οι άξονες σε κάποιους νευρώνες είναι καλυμμένοι με μία ουσία που ονομάζεται μυελίνη, ενώ άλλοι νευρώνες είναι εντελώς ακάλυπτοι. Τέλος υπάρχουν οι λεπτές επεκτάσεις που μοιάζουν με διακλαδώσεις δένδρου και ονομάζονται δενδρίτες. Οι δενδρίτες έρχονται και αυτοί σε επαφή, με άλλους δενδρίτες και δέχονται τα εισερχόμενα σήματα. Ανάμεσα στους νευρώνες κινούνται ηλεκτρικοί παλμοί. Η ταχύτητα των ηλεκτρικών παλμών στους απλούς νευρώνες κυμαίνεται από 10 - 20 m/sec, ενώ σε αυτούς που οι άξονες τους είναι καλυμμένοι με μυελίνη η ταχύτητα φθάνει τα 100 m/sec. Το μήκος το νευρώνων ποικίλει. Μερικοί έχουν μήκος μερικά χιλιοστά, ενώ άλλοι νευρώνες έχουν μήκος που φθάνει περίπου το ένα μέτρο, κυρίως αυτοί που εκτείνονται στα πόδια. Οι νευρώνες που έχουν μεγάλο μήκος έχουν πολύ μικρό πάχος, μερικά χιλιοστά δηλαδή είναι πολύ λεπτοί. Ειδικότερα τόσο ο νευροάξονας όσο και οι δενδρίτες έχουν τυπικό μέγεθος γύρω στο 1 μm , ενώ το Σώμα είναι γύρω στα 25 μm , όχι πολύ μεγαλύτερο από τον πυρήνα που περιέχει. Ο νευροάξονας ενός κινητικού νευρώνα μπορεί να είναι μεγαλύτερος από 1 μέτρο σε μήκος, συνδέοντας, για παράδειγμα, τη βάση της σπονδυλικής στήλης με τα δάχτυλα των ποδιών.

(1.<http://kelifos.physics.auth.gr/COURSES-A.Π.Θ.>)

1.5 Συστατικά και δομή του νευρώνα

Τα συστατικά του νευρών είναι οργανίδια νηματίδια και σωληνοειδή τμήματα που έχουν δημιουργηθεί από περίπου 6×10^9 μόρια πρωτεΐνης, 10^{10} μόρια λιπαρών ουσιών και 6×10^{11} μόρια RNA. Τα 2/3 του βάρους του νευρώνα αποτελούνται από περίπου 1500 μιτοχόνδρια. Τα μιτοχόνδρια είναι τα κέντρα παραγωγής ενέργειας των νευρώνων και των κυττάρων γενικότερα. Πιο συγκεκριμένα μεταφέρουν ταχύτερα γλυκόζη για την παραγωγή ενέργειας. Σε σχέση με τα κύτταρα του οργανισμού ο νευρώνας χρησιμοποιεί και ενέργεια ταχύτερα από κάθε άλλο. Κάθε μιτοχόνδριο χρειάζεται 10^7 άτομα οξυγόνου ανά δευτερόλεπτο. Εάν σταματήσει η παροχή

οξυγόνου για χρονικό διάστημα άνω των 10 δευτερολέπτων ,τότε ο νευρώνας καταστρέφεται. Παρόλο που ο εγκέφαλος αποτελεί περίπου το 2% της μάζας του ανθρωπίνου σώματος , εντούτοις είναι αδηφάγος γιατί καταναλώνει περισσότερο από το 20% του οξυγόνου που λαμβάνει ο οργανισμός. Η κατανάλωση ενέργειας στον εγκέφαλο είναι περίπου 20 Watt, γεγονός που τον καθιστά πολύ αποδοτικό. Αντίστοιχα ένας υπολογιστής χρειάζεται χιλιάδες Watt για να μπορέσει να λειτουργήσει.

1.6 Τύποι των νευρώνων

Οι σημαντικότεροι τύποι νευρώνων είναι οι παρακάτω:

1.Οι *αισθητήριοι νευρώνες* (ή αλλιώς *προσαγωγοί νευρώνες*) : συμμετέχουν στη λήψη ερεθισμάτων από το περιβάλλον, μεταφέροντας τις πληροφορίες από τα αισθητήρια όργανα στο κεντρικό νευρικό σύστημα.

2.Οι *ενδιάμεσοι ή συνδετικοί νευρώνες* : στο εσωτερικό του κεντρικού συστήματος, ενσωματώνουν τις πληροφορίες που παρέχουν οι αισθητήριοι νευρώνες και τις μεταδίδουν στους κινητικούς νευρώνες.

3.Οι *κινητήριοι νευρώνες* (ή αλλιώς *απαγωγοί νευρώνες*): μεταφέρουν τα μηνύματα στα δραστικά κύτταρα. Δραστικά κύτταρα είναι τα κύτταρα τα οποία αποκρίνονται ενεργά σε ερεθίσματα και η απόκρισή τους επιφέρει αποτελέσματα.

Σύμφωνα με τον αριθμό, το μήκος και τις διακλαδώσεις που έχουν, υπάρχουν οι παρακάτω νευρώνες:

1. *Μονόπολοι ή αλλιώς Ψευδομονόπολοι νευρώνες* :περιέχουν μία αποφυάδα που είναι πολύ κοντή και χωρίζεται σε δύο κλάδους από τους οποίους ο ένας από τους δύο θα λειτουργήσει ως δενδρίτης.

2. *Δίπολοι νευρώνες*: νευρώνες με σώμα από το οποίο εκφύονται ένας άξονας και ένας δενδρίτης από αντίθετους πόλους. (τους βρίσκουμε στα δίπολα κύτταρα του αμφιβληστροειδούς χιτώνα)

3. *Πολύπολοι νευρώνες*: με έναν μόνο άξονα και πολλούς δενδρίτες. (οι περισσότεροι νευρώνες) Μια ειδική κατηγορία αυτών των νευρώνων είναι οι *πυραμοειδές νευρώνες*.

1.7 Συνδεσιμότητα των νευρώνων

Οι συνδέσεις μεταξύ των νευρώνων, με τους άξονες και τους δενδρίτες, γίνονται στις επαφές που ονομάζονται *συνάψεις*. Οι νευρώνες επικοινωνούν μεταξύ τους και με άλλους νευρώνες μέσω συνάψεων, όπου η άκρη του νευροάξονα καταλήγει στους δενδρίτες, στο σώμα ή, σπανιότερα στον νευροάξονα άλλων νευρώνων. Η σύναψη έχει περίπλοκη δομή και επιτελεί επίσης περίπλοκες διεργασίες και λειτουργίες κατά την μετάδοση των σημάτων. Ο άξονας όπως αναφέραμε και παραπάνω έχει πάρα πολλές διακλαδώσεις και έτσι μπορεί και στέλνει πολύ μεγάλο αριθμό σε διαφορετικά σημεία. Στα σημεία στα οποία εφάπτονται οι δενδρίτες δημιουργείται μία σύναψη. Η επαφή η οποία δημιουργείται περιέχει ένα κενό το οποίο ονομάζεται *συναπτικό χάσμα* και το οποίο είναι της τάξης του 20-30nm. Η μεμβράνη του πρώτου νευρώνα που στέλνει το σήμα ονομάζεται *προσυναπτική μεμβράνη*, ενώ αυτή του δεύτερου νευρώνα που είναι δέκτης του σήματος ονομάζεται *μετασυναπτική μεμβράνη*. Ο χρόνος μεταφοράς του σήματος είναι 1msec έναντι 0.1 msec στις ηλεκτρικές συνάψεις. Στην άκρη κάθε διακλάδωσης υπάρχει ένα μικρό εξόγκωμα το οποίο εκρέει χημικούς μεταβιβαστές οι οποίοι ονομάζονται *νευροδιαβιβαστές*, οι οποίοι διαπερνούν το συναπτικό χάσμα και έτσι φθάνουν στον άλλο νευρώνα. Πιο συγκεκριμένα οι νευροδιαβιβαστές διαπερνούν το συναπτικό χάσμα και έτσι φθάνουν στον δενδρίτη του άλλου νευρώνα. Τα συναπτικά κυστίδια (*synaptic vesicles*) που βρίσκονται στην άκρη του άξονα ελευθερώνουν τους νευρομεταβιβαστές.

Οι νευροδιαβιβαστές είναι είναι περίπλοκα μόρια, πάνω από πενήντα διαφορετικά είδη τα οποία συμμετέχουν στις αντιδράσεις που λαμβάνουν χώρα μέσα στα κύτταρα. Πιο συγκεκριμένα οι νευροδιαβιβαστές είναι χημικές ουσίες του νευρικού συστήματος, είτε μικρομοριακές ουσίες, όπως τα αμινοξέα και η ακετυλοχολίνη, είτε μεγαλύτερα μόρια, όπως τα πεπτίδια τα οποία συντίθενται στο κυτταρικό σώμα. Αυτές οι χημικές ενώσεις χημικές ενώσεις, χρησιμεύουν στη μεταβίβαση

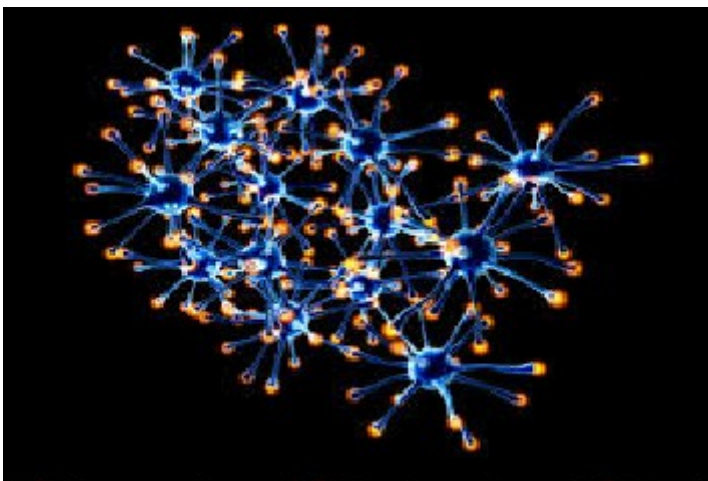
πληροφοριών από ένα νευρώνα στον επόμενο. Σημαντικοί νευροδιαβιβαστές είναι οι παρακάτω:

Σερετονίνη: Ένας από τους πιο γνωστούς νευροδιαβιβαστές που συνδέονται με την κατάθλιψη είναι η σεροτονίνη

Ντοπαμίνη: ρυθμίζει τα κίνητρα, τον μηχανισμό επιβράβευσης, την ευχαρίστηση και την ευφορία. Η έλλειψη της προκαλεί την ασθένεια του Parkinson.

Ακετυλοχολίνη και Σερετονίνη: Σχετίζονται με την μνήμη και μάθηση, και η έλλειψη των οποίων προκαλεί την ασθένεια Alzheimer.

Μερικοί νευροδιαβιβαστές είναι διεγερτικοί, ενώ άλλοι είναι ανασταλτικοί. Π.χ. η γλουταμίνη και η ασπαρτίνη είναι διεγερτικοί, ενώ η γλυκίνη είναι ανασταλτικός νευροδιαβιβαστής. Όταν απελευθερώνονται νευροδιαβιβαστές από ένα νευρώνα και και φθάνουν σε έναν άλλον μέσω των συνάψεων , τότε επηρεάζεται η μεμβράνη του νευρώνα που είναι ο αποδέκτης και αλλάζει ως προς το σήμα που θα στείλει αυτός ο νευρώνας. Παρατηρούμε ότι στις συνάψεις η αιτία της αλλαγής του δυναμικού είναι χημικής και όχι ηλεκτρικής μορφής.



Σχήμα 1.4 Πολλαπλές συνδέσεις μεταξύ νευρώνων

(4. Anatomy of the Brain - Sumit Kumar – Τι γνωρίζουμε για τον ανθρώπινο εγκέφαλο)

(5. Introduction to Neurons - Alla Borisjuk ,Avner FriedmanBard ,Ermentrout, David Terman -Εισαγωγή στους νευρώνες)

(1. <http://kelifos.physics.auth.gr/> -Α.Π.Θ.)

1.8 Η λειτουργία των νευρώνων

Κάθε νευρώνας έχει δύο δυνατές καταστάσεις στις οποίες μπορεί να βρίσκεται: τις ονομάζουμε ενεργό και μη-ενεργό κατάσταση. Όταν ο νευρώνας είναι ενεργός λέγουμε ότι πυροδοτεί, ενώ όταν είναι μη-ενεργός λέμε ότι είναι αδρανής. Ενδιάμεσες καταστάσεις δεν υπάρχουν. Κατά κάποιο τρόπο βλέπουμε ότι ο νευρώνας είναι δυαδικό (binary) στοιχείο, και μοιάζει στο σημείο αυτό με τους ηλεκτρονικούς υπολογιστές. Όταν ο νευρώνας πυροδοτεί, παράγει ένα ηλεκτρικό σήμα (παλμό), το οποίο κάθε φορά έχει τα ίδια χαρακτηριστικά.

Μεγάλο ενδιαφέρον παρουσιάζει ο τρόπος με τον οποίο δημιουργούνται τα ηλεκτρικά σήματα στους νευρώνες. Στην μεμβράνη του κυττάρου του νευρώνα δημιουργείται μια διαφορά δυναμικού μεταξύ της εσωτερικής και εξωτερικής επιφάνειας, όπως ακριβώς συμβαίνει και στους πυκνωτές.

Δυναμικό ενέργειας είναι η ηλεκτρική δραστηριότητα που αναπτύσσεται σε ένα μυικό ή νευρικό κύτταρο κατά τη διάρκεια της δραστηριότητάς του. Δηλαδή, αποτελεί μια παροδική, αναγεννητική, ηλεκτρική ώση, κατά την παραγωγή της οποίας το δυναμικό της μεμβράνης ενός κυττάρου αυξάνει ταχέως σε μια τιμή που είναι περίπου 100mV θετικότερου του φυσιολογικού, αρνητικού δυναμικού ηρεμίας. Αυτό προκαλεί τη διάδοση ενός σήματος κατά μήκος των νευρώνων ή των μυών σε μεγάλες αποστάσεις. Η μετάδοση των δυναμικών ενέργειας επιτρέπει στις πληροφορίες να διαβιβαστούν από τα αισθητήρια όργανα, μέσω των κεντρομόλων νευρώνων, στον εγκέφαλο.

Συνήθως το αρνητικό δυναμικό δημιουργείται στην εσωτερική επιφάνεια. Αυτό γίνεται γιατί υπάρχουν νόρια πρωτεϊνών με αρνητικό φορτίο και τα οποία δεν μπορούν να περάσουν την μεμβράνη και να βγουν έξω από το κύτταρο. Όταν το κύτταρο είναι σε ισορροπία, χωρίς να μεταδίδονται σήματα, τότε το “δυναμικό ηρεμίας” είναι περίπου -70 m volt.

Το δυναμικό αυτό είναι σχετικά πολύ μεγάλο, καθ' ότι το πάχος της μεμβράνης είναι μόλις 1 μ.

Η μεμβράνη έχει πολύ μικρές τρύπες σε όλο το μήκος της και έτσι επιτρέπει άτομα και ιόντα να την διαπερνούν. Τα πιο σημαντικά ιόντα που είναι τα ιόντα χλωρίου, ασβεστίου, καλίου και νατρίου. Κάθε ένα από αυτά έχει τα δικά του κανάλια, τα οποία διαπερνούν την μεμβράνη. Αξιοσημείωτο είναι ότι τα κανάλια αυτά έχουν πύλες οι οποίες ανοίγουν και κλείνουν, έτσι ώστε να επιτρέπουν ή να απαγορεύουν την ροή των ιόντων δια μέσω της μεμβράνης. Η μεμβράνη του κυττάρου έχει κάποια μόρια που είναι πρωτεΐνες, οι οποίες δρουν δρουν ως “αντλία” και μεταφέρουν τα ιόντα δια μέσω της μεμβράνης.

Αναγκάζουν τα ιόντα να κινούνται αντίθετα από την φυσική συγκέντρωση ισορροπίας, πράγμα που για να το κάνουν ξοδεύουν ενέργεια, και για αυτό το λόγο οι νευρώνες χρειάζονται μεγάλα ποσά ενέργειας. Η συγκέντρωση των ιόντων αυτών και ακολούθως η φυσική τους κίνηση κατά μήκος του κυττάρου δημιουργεί ένα ηλεκτρικό ρεύμα το οποίο αποτελεί το ηλεκτρικό σήμα που μεταδίδεται στο κύτταρο. Όπως βλέπουμε στο σχήμα 1.5 (απότομη άνοδος και απότομη κάθοδος) το δυναμικό της μεμβράνης δημιουργεί μία πραγματική εκκένωση. Το εσωτερικό του νευρώνα μπορεί να γίνει στιγμιαία ακόμα και θετικά φορτισμένο, ως προς το υπόλοιπο κύτταρο. Σταδιακά όμως επιστρέφει στην κανονική του κατάσταση, στο δυναμικό ηρεμίας. Είναι ιδιαίτερα αξιοσημείωτο ότι κατά την διάρκεια της αποκατάστασης είναι αδύνατο ο νευρώνας να δεχθεί άλλη διέγερση, έστω και αν πολλαπλά σήματα καταφθάσουν ταυτόχρονα και προσπαθήσουν να τον διεγείρουν.

Η περίοδος κατά την οποία ισχύει αυτό λέγεται περίοδος μεταστροφής (refractory period). Όλα τα σήματα που καταφθάνουν σε ένα νευρώνα σε μια δεδομένη στιγμή αθροίζονται. Αυτό σημαίνει ότι αθροίζονται τα ηλεκτρικά δυναμικά τους. Αν το άθροισμα των σημάτων φθάσει ή ξεπεράσει μια δεδομένη τιμή, (κατώφλι) τότε

θεωρείται ότι ο νευρώνας βρίσκεται σε διεγερμένη κατάσταση και πυροδοτεί. Αυτό έχει ως αποτέλεσμα να στείλει μέσω του άξονα ένα παλμό.

Αν το άθροισμα όμως είναι μικρότερο από την δεδομένη αυτή τιμή, τότε δεν συμβαίνει τίποτα. Στην περίπτωση αυτή ο νευρώνας παραμένει αδρανής. Το δυναμικό αυτό, το οποίο είναι μικρότερο από το κατώφλι χάνετε. Την τιμή του κατωφλίου την ονομάζουμε θ . Το εισερχόμενο σήμα μπορεί να είναι είτε διεγερτικό είτε ανασταλτικό. Όταν το εισερχόμενο σήμα είναι διεγερτικό αυτό σημαίνει ότι το σήμα αυτό είναι θετικό και κάνει το δυναμικό του νευρώνα να πλησιάσει κοντά στο θ . Το τελικό αποτέλεσμα εξαρτάται από την συνάρτηση του κατωφλίου, η οποία συνήθως είναι η συνάρτηση Heaviside η οποία περιγράφεται στο σχήμα 1.5.

Αν είναι ανασταλτικό τότε συμβαίνει το αντίθετο, δηλαδή το σήμα είναι αρνητικό και κάνει το δυναμικό να απομακρύνεται από το θ . Αν το σήμα είναι ίσο ή ξεπερνά το κατώφλι δυναμικού, τότε ο νευρώνας διεγείρεται, και είναι έτοιμος να στείλει έναν παλμό που έχει πάντοτε το ίδιο μέγεθος. Αμέσως μετά τον παλμό, ο νευρώνας επανέρχεται στην αρχική του κατάσταση. Αν οι συνθήκες το επιτρέψουν μπορεί αργότερα να ενεργοποιηθεί πάλι. Το σήμα όμως που μεταδόθηκε συνεχίζει την ίδια διαδικασία σε άλλους νευρώνες του δικτύου χωρίς να ελαττωθεί καθόλου.

(1.<http://kelifos.physics.auth.gr-A.Π.Θ.>)

(6.<https://www.cognifit.com/el/science/didyou-know/neurons>)



Σχήμα 1.5

ΚΕΦΑΛΑΙΟ 2 - Τεχνητά νευρωνικά δίκτυα

2.1 Τι είναι ένα τεχνητό νευρωνικό δίκτυο (ΤΝΔ)

Τα τεχνητά νευρωνικά δίκτυα αποτελούν μία προσπάθεια προσέγγισης και μίμησης της λειτουργίας του εγκεφάλου και της ανθρώπινης σκέψης από τους ηλεκτρονικούς υπολογιστές. Το όνομά τους όπως και άλλοι όροι που θα αναφερθούν παρακάτω προέρχονται από τα βιολογικά νευρωνικά δίκτυα των ανθρωπίνων οργανισμών, τα οποία περιγράφονται στο προηγούμενο κεφάλαιο.

Ένα τεχνητό νευρωνικό δίκτυο (Artificial Neural Network) είναι ένα υπολογιστικό σύστημα υλικού και λογισμικού του οποίου η δομή και η λειτουργία είναι εμπνευσμένη από τον τρόπο λειτουργίας των βιολογικών νευρικών δικτύων, τα οποία αποτελούν δομικά συστατικά των εγκεφάλων των ζώων και των ανθρώπων. Σύμφωνα με τον ορισμό των Alexander και Morton ένα νευρωνικό δίκτυο είναι ένας τεράστιος παράλληλος επεξεργαστής με κατανομημένη αρχιτεκτονική, ο οποίος αποτελείται από απλές μονάδες επεξεργασίας και έχει από τη φύση του τη δυνατότητα να αποθηκεύει εμπειρική γνώση και να την καθιστά διαθέσιμη για χρήση.

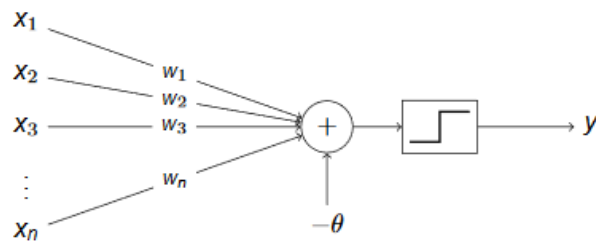
Επίσης υπάρχει και ο ορισμός του Haykin που δόθηκε το 1994 και ορίζει τα τεχνητά νευρωνικά δίκτυα ως εξής:

“Ένα νευρωνικό δίκτυο είναι ένας μεγάλου μεγέθους επεξεργαστής παράλληλου υπολογισμού που έχει μία φυσική ροπή αποθήκευσης πειραματικής γνώσης και μετατροπής της σε μορφή διαθέσιμη για επεξεργασία. Αντίστοιχα με τον ανθρώπινο εγκέφαλο η γνώση αποκτάται μέσω μίας μαθησιακής διεργασίας, τα βάρη μεταξύ των συνάψεων χρησιμοποιούνται για να αποθηκεύσουν τη γνώση .”

2.2 Ιστορική αναδρομή

Τα τεχνητά νευρωνικά δίκτυα , όπως έχουμε αναφέρει στο προηγούμενο κεφάλαιο αποτελούν μία νέα περιοχή και δεν έχουν μεγάλη προιστορία όπως συμβαίνει με άλλους επιστημονικούς τομείς Η μελέτη τους ξεκίνησε σε παγκόσμιο επίπεδο τις τελευταίες δεκαετίες όμως η μεγαλύτερη ανάπτυξη σημειώθηκε από την δεκαετία 1980 και μετά. Σε αυτό συνέβαλε η μεγάλη ανάπτυξη των ηλεκτρονικών υπολογιστών σε υλικό και λογισμικό.

Το πρώτο μοντέλο νευρωνικού δικτύου παρουσιάστηκε το 1943 από τους Warren McCulloch και Walter Pitts. Ο McCulloch ήταν νευροφυσιολόγος και ο Pitts ήταν πρωτοετής φοιτητής μαθηματικών. Σε μία πρώτη τους εργασία αυτοί οι μελετητές παρουσίασαν για πρώτη φορά την ιδέα ότι ένα νευρωνικό δίκτυο αποτελείται από μία συλλογή ενός μεγάλου αριθμού νευρώνων και έδειξαν πως θα μπορούσαν να λειτουργήσουν αυτοί οι νευρώνες με τις συνδέσεις τους. Αυτή θεωρείται ιστορικά η πρώτη εικόνα ενός νευρωνικού δικτύου. Το σχήμα 2.1 αναπαριστά αυτό το πρότυπο. Οι ίδιοι μελετητές το 1947 προχώρησαν σε ένα πιο εξελιγμένο πρότυπο νευρωνικού δικτύου το οποίο μπορούσε να αναγνωρίζει σχήματα. Στο πρότυπο αυτό ο νευρώνας μπορεί να έχει μόνο δύο καταστάσεις και πολλές εισόδους αλλά να έχει μόνο μία έξοδο. Οι δύο καταστάσεις του νευρώνα είναι όταν πυροδοτεί ή όταν βρίσκεται σε ηρεμία. Οι απολήξεις των νευρώνων είναι δύο ειδών: διεγερτικές και ανασταλτικές .Επίσης στο πρότυπο αυτό υπάρχει και η λειτουργία της μνήμης μέσω κλειστών διαδρομών του σήματος μέσα στο δίκτυο. Πιο συγκεκριμένα θεωρούν ότι ένας μηχανισμός είναι η ύπαρξη κλειστών διαδρομών του σήματος μέσα στο δίκτυο. Έτσι όταν μία διαδρομή ενώνει την έξοδο ενός κυττάρου με το σημείο εισόδου μέσα στο ίδιο κύτταρο δημιουργείται ένας μηχανισμός ανάδρασης (feedback).



- x_1, x_2, \dots, x_n : **Είσοδοι** του τεχνητού νευρώνα.
- w_1, w_2, \dots, w_n : **Συναπτικά βάρη** των εισόδων.
- θ : **κατώφλι ενεργοποίησης** του νευρώνα.
- Η **έξοδος** y του νευρώνα είναι:

$$y = \begin{cases} 1, & \sum_{i=1}^n w_i x_i > \theta \quad (\text{πυροδοτεί}) \\ 0, & \text{αλλιώς} \quad (\text{αδρανής}) \end{cases}$$

Σχήμα 2.1

Το μοντέλο τεχνητού νευρώνα McCulloch-Pitts που προτάθηκε το 1943.

Λίγα χρόνια αργότερα ο John Neumann χρησιμοποίησε τις εργασίες των McCulloch-Pitts ως παράδειγμα για υπολογιστικές μηχανές και τότε έγιναν οι πρώτες προσπάθειες άντλησης πληροφοριών από τα βιολογικά νευρωνικά δίκτυα και δημιουργίας των πρώτων τεχνητών νευρωνικών δικτύων. Το 1949 ο Hebb εισάγει το μοντέλο της μάθησης του νευρώνα σύμφωνα με το οποίο το νευρωνικό δίκτυο κάθε φορά που χρησιμοποιεί τις συνδέσεις μεταξύ των νευρώνων, οι συνδέσεις αυτές βελτιώνονται και το δίκτυο προσεγγίζει καλύτερα το πρότυπο το πρότυπο το οποίο παρουσιάζεται. Ειδικότερα όταν ο νευρώνας i διεγείρει επανειλημμένα τον νευρώνα j , τότε αναπτύσσεται μια μεταβολική σύνδεση στον ένα ή και στους δύο νευρώνες, έτσι ώστε η απόδοση του φαινομένου (το i διεγείρει το j) να αυξάνεται. Αν υποθέσουμε ότι w_{ij} είναι το βάρος της σύνδεσης μεταξύ των νευρώνων i και j , x_i η

είσοδος στον νευρώνα j από τον νευρώνα i και x_j η έξοδος του νευρώνα j τότε ισχύει ότι: $w_{ij}(\text{new}) = w_{ij}(\text{old}) + ax_i$.

Εδώ το a είναι μία θετική σταθερά που λέγεται παράμετρος του ρυθμού εκπαίδευσης. Το νέο λοιπόν βάρος w_{ij} θα είναι ίσο με το παλαιό ενισχυμένο κατά όνο για την σύνδεση του νευρώνα i και j και όχι για άλλες συνδέσεις του δικτύου.

Το 1957 ο Frank Rosenblatt προτείνει την στοιχειώδη μονάδα του τεχνητού νευρωνικού δικτύου που ονομάζεται αισθητήρας (perceptron). Ο Rosenblatt κατασκεύασε το πρώτο δίκτυο με υλικό (hardware), το οποίο ήταν ήταν ένα πολύ απλό μοντέλο με δύο επίπεδα, της εισόδου και της εξόδου όπου το σήμα προχωρά μονοδρομικά από την είσοδο προς την έξοδο. Το μοντέλο αυτό στην αρχή είχε σημειώσει πολλές επιτυχίες και δημιούργησε μεγάλο ενθουσιασμό και δημιουργήθηκε η ιδέα ότι τα νευρωνικά δίκτυα πιθανόν είναι η ανώτερη τεχνική που είναι ικανή να επιλύσει όλα τα προβλήματα που μέχρι τότε ήταν άλυτα. Οι πρώτες επιτυχίες και τα πρώτα αποτελέσματα μεγαλοποιήθηκαν αλλά γρήγορα φάνηκε ότι τα μοντέλα αυτά περιείχαν και πολλούς περιορισμούς. Οι Minsky και Papert το 1969 στο βιβλίο "Perceptrons" παρουσίασαν μία συνολική εκτίμηση της χρησιμότητας του perceptron.

Το 1959, την ίδια χρονική περίοδο που έγινε και η ανάπτυξη του μοντέλου του αισθητήρα οι Widrow και Hoff ανέπτυξαν το 1959 δύο νέα μοντέλα, το Adaline και το Madaline, τα οποία ήταν από τα πρώτα μοντέλα που χρησιμοποιήθηκαν επιτυχώς για πρακτικά προβλήματα και συγκεκριμένα χρησιμοποιήθηκαν ως φίλτρα για να εξαλείψουν την ηχώ σε τηλεφωνικές γραμμές.

Την επόμενη είκοσι χρόνια, περίπου μέχρι το 1980 δεν η πρόοδος στον τομέα των νευρωνικών δικτύων ήταν πολύ μικρή γιατί οι περιορισμοί που αναφέρθηκαν παραπάνω αποθάρρυναν τους περισσότερους μελετητές να ασχοληθούν με αυτό το πεδίο. Το 1982 ένας βιολόγος ο John Hopfield σε μία εργασία 5 σελίδων απέδειξε με αυστηρά μαθηματικά ότι ένα νευρωνικό δίκτυο μπορεί να χρησιμοποιηθεί ως

αποθηκευτικός χώρος και πώς μπορεί ένα δίκτυο να επανακτήσει όλη την πληροφορία ενός συστήματος έστω και αν του δοθούν μερικά τμήματα μόνο και όχι ολόκληρο το σύστημα. Αμέσως εκτιμήθηκε η σπουδαιότητα αυτής της ιδιότητας και η εργασία αυτή αποτέλεσε έμπνευση για πολλές άλλες ιδέες που ακολούθησαν.

Ένα επόμενο μεγάλο βήμα ήταν η πρόοδος που σημειώθηκε στην διαδικασία εκπαίδευσης των νευρωνικών δικτύων όταν επινοήθηκε ο κανόνας της διόρθωσης του σφάλματος (error correction learning). Σύμφωνα με αυτόν τον κανόνα κατά την εκπαίδευση του δικτύου σε όποια κατάσταση και αν βρίσκεται αυτό σε μία δεδομένη στιγμή t , σημασία έχει η απόκλιση που δίνει στην έξοδο του το δίκτυο από την αναμενόμενη τιμή ή τον στόχο που έχουμε θέσει. Η διαφορά αυτή δίνει το σφάλμα που παράγει το δίκτυο μας την δεδομένη στιγμή t :

$$\delta(t) = T(t) - O(t) \quad (2.1)$$

όπου όπου « T » είναι ο στόχος, « O » είναι η έξοδος και « δ » είναι το σφάλμα .

Με την χρήση του δ γίνεται ενεργοποίηση ενός μηχανισμού ελέγχου που έχει σαν σκοπό να πραγματοποιήσει ορισμένες διορθωτικές αλλαγές στα βάρη w , γεγονός που θα φέρει τα βάρη πλησιέστερα στην εκπαίδευσή του και στους στόχους του. Αυτό γίνεται με τον υπολογισμό του Δ :

$$\Delta_i = \eta \delta x \quad (2.2)$$

όπου η είναι μία σταθερά, και x_i είναι η τιμή του σήματος εισόδου. Τέλος το $\Delta w(n)$, δηλαδή η διόρθωση του βάρους w στην δεδομένη στιγμή t , δίνεται κατ' ευθείαν από το Δ :

$$\Delta w(n) = \Delta i \quad (2.3)$$

Ο κανόνας αυτός ονομάζεται «κανόνας Δ » ή κανόνας «Widrow–Hoff» και χρησιμοποιείται ευρύτατα σε διάφορες μεθόδους εκπαίδευσης δικτύων. Το 1986 οι McClelland και Rumelhart, δημοσιεύουν το έργο το “Parallel Distributed

Processing” το οποίο άνοιξε καινούργιους δρόμους στην εκπαίδευση των νευρωνικών δικτύων. Στο έργο αυτό παρουσιάστηκε πως η ιδέα πως ένα νευρωνικό δίκτυο μπορεί να θεωρηθεί και να χρησιμοποιηθεί ως παράλληλος επεξεργαστής. Επίσης στο έργο αυτό έκανε ένα σημαντικό βήμα με το να επιτρέπει την ύπαρξη κρυμμένων επιπέδων νευρώνων εκτός από την είσοδο και την έξοδο του νευρωνικού δικτύου. Έτσι έγινε μια μεγάλη αλλαγή στην εσωτερική δομή του δικτύου. Στο ανωτέρω έργο προτάθηκε μία νέα μέθοδος εκπαίδευσης, η μέθοδος της οπισθοδιάδοσης (backpropagation), η οποία χρησιμοποιεί τις εξισώσεις 2.1 και 2.3 και κατέληξε να είναι η πιο χρήσιμη σήμερα τεχνική εκπαίδευσης δικτύων .

2.3 Το μοντέλο του τεχνητού νευρώνα

Όπως στα βιολογικά νευρωνικά δίκτυα το βασικό στοιχείο είναι ο βιολογικός νευρώνας , αντίστοιχα τα τεχνητά νευρωνικά δίκτυα αποτελούνται από τους τεχνητούς νευρώνες. Οι τεχνητοί νευρώνες αναφέρονται και ως μονάδες επεξεργασίας (nodes) ή και κύτταρα (cells) και έχουν συνδέσεις που τους επιτρέπουν να δέχονται και να αποστέλλουν σήματα. Ο τεχνητός νευρώνας είναι ένα χρονικά αναλλοίωτο σύστημα χωρίς μνήμη με πολλές εισόδους και μία έξοδο (σχήμα 2.2). Σε κάθε νευρώνα καταφθάνει ένας αριθμός σημάτων, τα οποία έρχονται ως είσοδος σε αυτόν. Το σύνολο των εισερχομένων σημάτων αθροίζεται και στην περίπτωση που το αποτέλεσμα υπερβαίνει κάποιο όριο ενεργοποίησης υφίσταται μετατροπή μέσω μίας συνάρτησης μεταφοράς. Αυτό το εξερχόμενο σήμα θα είναι είτε το δεδομένο εισόδου σε επόμενο νευρώνα ή το αποτέλεσμα εξόδου του νευρωνικού δικτύου. Το σημείο επικοινωνίας δύο νευρώνων ονομάζεται σύνδεση ή σύναψη (synapse) και η ισχύς της συναπτικό βάρος. Το συναπτικό βάρος το συμβολίζουμε συνήθως με το w .

Κάθε σήμα που μεταδίδεται από ένα νευρώνα σε ένα άλλο μέσα στο νευρωνικό δίκτυο συνδέεται με κάποια τιμή βάρους w και η οποία υποδηλώνει πόσο στενά είναι συνδεδεμένοι οι δύο νευρώνες που συνδέονται με το συγκεκριμένο βάρος. Η τιμή αυτή συνήθως κυμαίνεται σε ένα συγκεκριμένο διάστημα, π.χ. στο διάστημα από -1 ως 1 , αλλά αυτό είναι αυθαίρετο και εξαρτάται από το πρόβλημα που προσπαθούμε να επιλύσουμε. Η σημασία του βάρους είναι κάτι αντίστοιχο όπως και ο χημικός δεσμός ανάμεσα σε δύο άτομα που απαρτίζουν ένα μόριο. Ο δεσμός μας δείχνει πόσο δυνατά είναι συνδεδεμένα τα δύο άτομα του μορίου. Έτσι και ένα βάρος μας λέγει ακριβώς πόσο σημαντική είναι η συνεισφορά του συγκεκριμένου σήματος στην διαμόρφωση της δομής του δικτύου για τους δύο νευρώνες τους οποίους συνδέει.

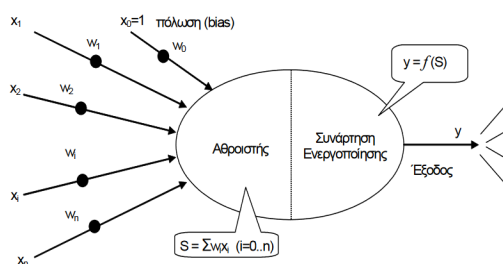
Όταν το w είναι μεγάλο, τότε η συνεισφορά του σήματος είναι μεγάλη, ενώ όταν το w είναι μικρό τότε η συνεισφορά του σήματος είναι μικρή.

Το άλλο βασικό στοιχείο ενός τεχνητού νευρωνικού δικτύου είναι οι στοιβάδες ή αλλιώς τα στρώματα (layers). Κάθε τέτοιο δίκτυο αποτελείται από στοιβάδες που συνδέονται μεταξύ τους και διακρίνονται στην στοιβάδα εισόδου (input layer), την στοιβάδα εξόδου (output layer) και τις κρυμμένες στοιβάδες (hidden layers), όπως φαίνεται στο σχήμα 2.3.

Το τρίτο βασικό στοιχείο είναι οι συνδέσεις. Οι συνδέσεις είναι γραμμές επικοινωνίας από τον νευρώνα που αποστέλλει ένα σήμα (νευρώνα πομπό) στον νευρώνα που λαμβάνει το σήμα (νευρώνα-δέκτη) του σήματος. Υπάρχουν συνδέσεις που προκαλούν πυροδότηση-ενεργοποίηση του νευρώνα που ονομάζονται διεγερτικές (excitatory) και συνδέσεις που την αποτρέπουν και ονομάζονται ανασταλτικές (inhibitory). Επίσης υπάρχουν είδη συνδέσεων που είναι πολύπλοκα όπως η ανάδραση. Οι συνδέσεις μαζί με τα συναπτικά βάρη που την συνοδεύουν καθορίζουν την διάδοση του σήματος μέσα στο δίκτυο.

Τα ανωτέρω συστατικά των νευρωνικών δικτύων, δηλαδή τα βάρη οι συνδέσεις και οι στοιβάδες υπάρχουν στο σχήμα 2.3.

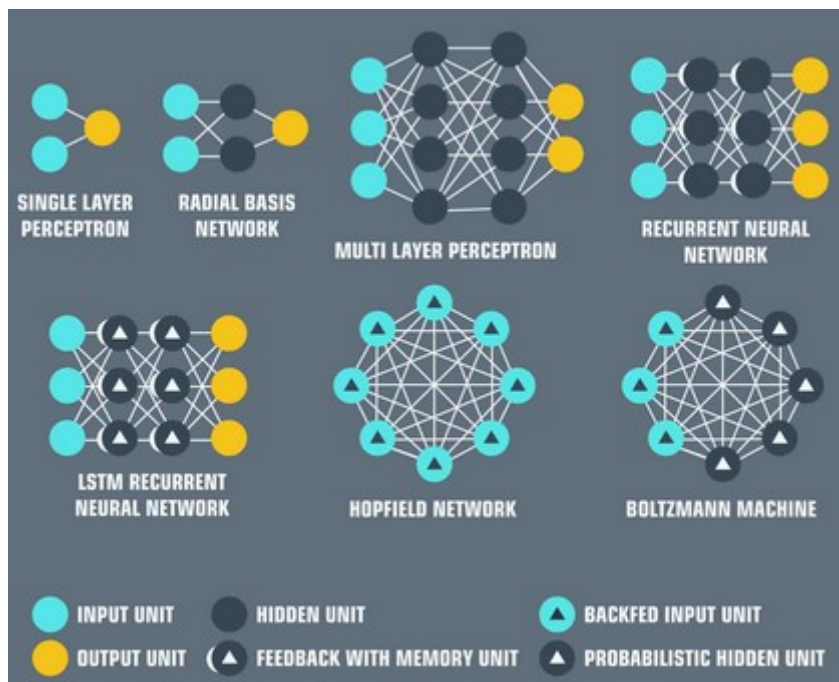
Ένα τεχνητό νευρωνικό δίκτυο αποτελείται από την σειριακή, την παράλληλη και με ανατροφοδότηση σύνδεση νευρώνων. Γενικότερα τα τεχνητά νευρωνικά δίκτυα διακρίνονται σε νευρωνικά δίκτυα πολλαπλών επιπέδων (multilayer neural networks) και επαναληπτικά νευρωνικά δίκτυα (recurrent neural networks).



Σχήμα 2.2 Ένας νευρώνας (ο κύκλος) με πολλές εισόδους, τις $x_1, x_2, x_3 \dots x_n$, τα συναπτικά βάρη $w_1, w_2, w_3, \dots w_n$, και μία έξοδο y .

2.4 Νευρωνικά δίκτυα πολλαπλών επιπέδων

Νευρωνικά δίκτυα πολλαπλών επιπέδων : Σε αυτά τα νευρωνικά δίκτυα οι νευρώνες οργανώνονται σε μια ακολουθία επιπέδων. Τα επίπεδα διακρίνονται σε επίπεδα εισόδου (input layer), σε κρυφά επίπεδα (hidden layers) και σε επίπεδα εξόδου (output layer). Παραδείγματα τεχνητών νευρωνικών δικτύων υπάρχουν στο σχήμα 2.3.



Σχήμα 2.3. Παραδείγματα τεχνητών νευρωνικών δικτύων

(9.On the Origin of Deep Learning ,Haohan ,Bhiksha Raj – Προέλευση της μηχανικής μάθησης)

2.5 Επαναληπτικά νευρωνικά δίκτυα

Τα επαναληπτικά ή αλλιώς αναδρομικά τεχνητά νευρωνικά δίκτυα είναι συστήματα με μνήμη και αναπαρίστανται ως δυναμικά συστήματα. Η απλούστερη μορφή τους είναι αυτή του δικτύου Hopfield διακριτού χρόνου του σχήματος 2. . Στο δίκτυο αυτό , τη χρονική στιγμή $t = 0$, το σήμα εισόδου αρχικοποιεί m επεξεργαστικές μονάδες και κάθε μία από αυτές υλοποιεί την δυναμική εξίσωση

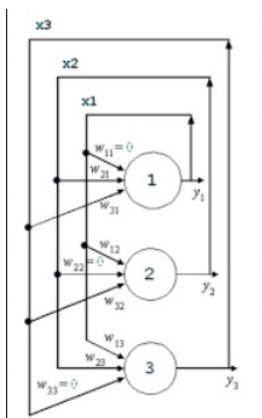
$$a_i(k+1) = f(\sum_{j=1}^m w_{ij}z_j(k)) , \text{ με } k=0,1,2, \dots$$

$$\text{και } z_i(0) = u_i(0)$$

Σε συμπαγή μορφή η παραπάνω εξίσωση γράφεται: $a(k+1) = f(wz(k))$, $k = 1,2,3,\dots$

Ένα τέτοιο νευρωνικό δίκτυο δέχεται το σήμα εισόδου $u(n)$, το οποίο προκαλεί επαναληπτικό υπολογισμό του διανύσματος $a(k + 1)$. Μετά από έναν ορισμένο αριθμό επαναλήψεων , το δίκτυο κατασταλάζει σε μία σταθερή κατάσταση δηλαδή το διάνυσμα a δεν μεταβάλλεται από επανάληψη σε επανάληψη .Η τελική (ευσταθής) κατάσταση του διανύσματος z παρουσιάζεται ως έξοδος του δικτύου .

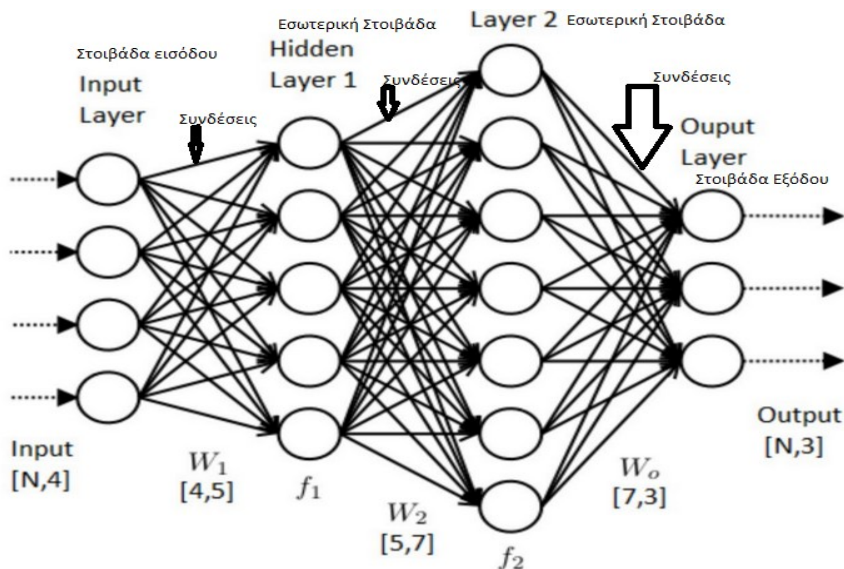
Αν τα βάρη w_{ij} επιλεγούν κατάλληλα , τότε το δίκτυο παρουσιάζει μόνο έναν πεπερασμένο αριθμό ευσταθών καταστάσεων, τα οποία αντιστοιχούν σε “πρότυπα αποθήκευσης του δικτύου” . Κάθε σήμα που είναι είσοδος στο δίκτυο , θεωρείται ως μία θορυβώδης μορφή κάποιου από τα αποθηκευμένα πρότυπα. Το δίκτυο λειτουργεί ως μία συσχετιστική μνήμη και ταυτοποιεί το σήμα εισόδου με κάποιο από τα αποθηκευμένα πρότυπα.



Σχήμα 2.4. Παράδειγμα αναδρομικού νευρωνικού δικτύου.

Σχήμα 2.5 Ένας νευρώνας (ο κύκλος) με πολλές εισόδους, τις $x_1, x_2, x_3 \dots x_n$, τα συναπτικά βάρη $W_1, W_2, W_3, \dots W_n$, και μία έξοδο y .

Τα τεχνητά νευρωνικά δίκτυα προκύπτουν από στην συνδεσμολογία των νευρώνων σε διάφορες τοπολογίες.



(8.Artificial neural networks: fundamentals, computing, design, and application, I.A. Basheer, M. Hajmeer - Τεχνητά νευρωνικά δίκτυα: ανάλυση, σχεδιασμός και εφαρμογές)

2.6 Λειτουργία του νευρωνικού δικτύου

Έχοντας την δομή ενός δικτύου όπως στα σχήματα 2.5, μένει τώρα να δούμε πως και με ποιά διαδικασία μεταδίδεται το σήμα από νευρώνα σε νευρώνα σε όλο το δίκτυο. Ουσιαστικά γίνεται πάντοτε αυτό που αναφέρθηκε παραπάνω, δηλαδή, όλα τα σήματα που φθάνουν σε ένα νευρώνα αθροίζονται και υπόκεινται σε μία διαδικασία και το αποτέλεσμα που παράγεται από αυτή την διαδικασία είναι το σήμα το οποίο μεταδίδεται περαιτέρω στους επόμενους νευρώνες. Η θεώρηση αυτή είναι γενική και ισχύει πάντοτε, αυτό όμως που αλλάζει είναι η διαδικασία η οποία δεν είναι πάντα η ίδια.

Όπως προκύπτει από τα παραπάνω οι νευρώνες εισόδου δεν επιτελούν κανέναν υπολογισμό, μεσολαβούν απλώς ανάμεσα στις εισόδους του δικτύου και στους υπολογιστικούς νευρώνες. Οι νευρώνες εξόδου διοχετεύουν στο περιβάλλον τις τελικές αριθμητικές εξόδους του δικτύου.

Οι υπολογιστικοί νευρώνες πολλαπλασιάζουν κάθε είσοδό τους με το αντίστοιχο συναπτικό βάρος και υπολογίζουν το ολικό άθροισμα των γινομένων. Το άθροισμα αυτό τροφοδοτείται ως όρισμα στη *συνάρτηση ενεργοποίησης*, την οποία υλοποιεί εσωτερικά κάθε κόμβος. Η τιμή που λαμβάνει η συνάρτηση για το εν λόγω όρισμα είναι και η έξοδος του νευρώνα για τις τρέχουσες εισόδους και βάρη.

Εάν x_{ki} είναι η i -οστή είσοδος του k νευρώνα, w_{ki} το i -οστό συναπτικό βάρος του k νευρώνα και $\phi(\cdot)$ η συνάρτηση ενεργοποίησης του νευρωνικού δικτύου, τότε η έξοδος y_k του k νευρώνα δίνεται από την εξίσωση:

$$y_k = \phi \left(\sum_{i=0}^N x_{ki} w_{ki} \right)$$

Στον k -οστό νευρώνα υπάρχει ένα συναπτικό βάρος w_{k0} με ιδιαίτερη σημασία, το οποίο καλείται **πόλωση** ή **κατώφλι** (bias, threshold). Η τιμή της εισόδου του είναι πάντα η μονάδα, $x_{k0} = 1$. Εάν το συνολικό άθροισμα από τις υπόλοιπες εισόδους

του νευρώνα είναι μεγαλύτερο από την τιμή αυτή, τότε ο νευρώνας ενεργοποιείται. Εάν είναι μικρότερο, τότε ο νευρώνας παραμένει ανενεργός. Η ιδέα προέκυψε από τα βιολογικά νευρικά κύτταρα.

Οι τεχνητοί νευρώνες ενός δικτύου αλληλεπιδρούν διαρκώς μεταξύ τους και είναι συνδεδεμένοι με τις λεγόμενες συνάψεις (synapses). Για κάθε ζευγάρι νευρώνων διαφέρει το μέγεθος της αλληλεπίδρασης αλλά προσδιορίζεται από τα συναπτικά βάρη (synaptic weights). Πιο συγκεκριμένα, τα συναπτικά βάρη συνεχώς μεταβάλλονται όσο το δίκτυο στο οποίο ανήκουν αλληλεπιδρά με το εξωτερικό περιβάλλον. (Haykin, 2009) Για το λόγο αυτό, διαρκώς ενδυναμώνεται ή αποδυναμώνεται η ισχύς των δεσμών του δικτύου. Επομένως, η λειτουργία που επιτελούν τα συναπτικά βάρη είναι ιδιαίτερα σημαντική, διότι η γνώση που αποκτά το δίκτυο από το εξωτερικό του περιβάλλον κωδικοποιείται σε αυτά. Έτσι, κάθε νευρωνικό δίκτυο αποκτά την ικανότητα για προσαρμογή στο περιβάλλον και συνεχή εξέλιξη.

Συναρτήσεις Ενεργοποίησης

Η παραγόμενη έξοδος εξαρτάται από την συνάρτηση ενεργοποίησης. Γενικά μπορεί να υπάρχει οποιαδήποτε συνάρτηση παραγωγής της εξόδου, όμως οι πιο συνηθισμένες είναι η βηματική (step transfer function), η γραμμική (linear transfer function), η μη γραμμική (non-linear transfer function), ή στοχαστική (stochastic transfer function) και η σιγμοειδής συνάρτηση (sigmoid function).

Βηματική συνάρτηση ενεργοποίησης

Η βηματική συνάρτηση ενεργοποίησης μπορεί να είναι:

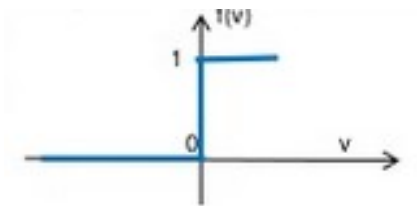
$$\phi(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

ή οποιαδήποτε άλλη βηματική συνάρτηση.

Η γενική μορφή της συνάρτησης είναι η :

$$\varphi(v) = \begin{cases} 1, & \sum_{i=0}^n w_i x_i \geq \theta \\ 0, & \sum_{i=0}^n w_i x_i < \theta \end{cases}$$

και η γραφική της παράσταση είναι η παρακάτω:



Η βηματική συνάρτηση δεν θεωρείται ιδιαίτερα χρήσιμη ως συνάρτηση ενεργοποίησης στα Τεχνητά Νευρωνικά Δίκτυα, καθώς σύμφωνα με τον απειροστικό λογισμό έχει το βασικό μειονέκτημα ότι η παράγωγός της απειρίζεται.

Έτσι προέκυψε η ανάγκη συναρτήσεων ενεργοποίησης που η γραφική τους παράσταση τους να μοιάζει με τη βηματική, αλλά ταυτόχρονα να είναι συνεχείς και παραγωγίσιμες σε όλο το πεδίο ορισμού τους.

Τέτοια συνάρτηση είναι και η σιγμοειδής.

Σιγμοειδής συνάρτηση

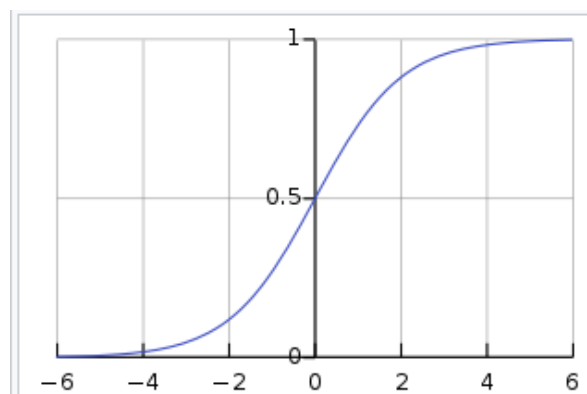
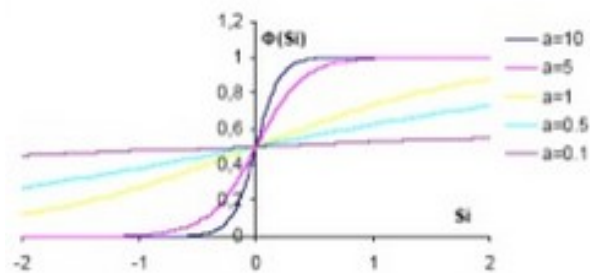
Η σιγμοειδής συνάρτηση είναι μια μαθηματική συνάρτηση η οποία έχει μορφή S και ονομάζεται επίσης και ως σιγμοειδής καμπύλη. Συχνά ως σιγμοειδής συνάρτηση αναφέρεται η ειδική περίπτωση της λογιστικής παλινδρόμησης. Ο παρακάτω μαθηματικός τύπος είναι ο παρακάτω:

$$S(t) = \frac{1}{1 + e^{-t}}$$

και στη γενική μορφή :

$$\varphi(v) = \frac{1}{1 + e^{-av}}$$

Η γραφική παράσταση της συνάρτησης είναι η παρακάτω:



Γραμμική συνάρτηση ενεργοποίησης

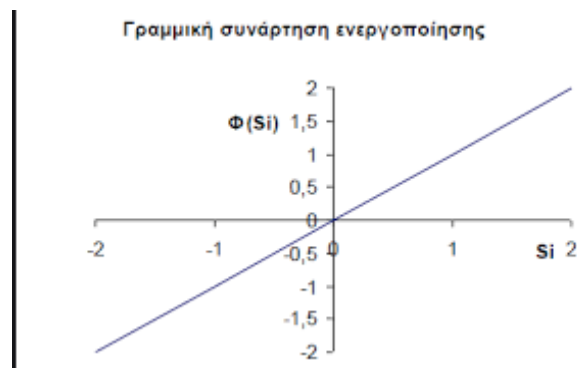
Η γραμμική συνάρτηση ενεργοποίησης μπορεί να είναι:

$$\phi(x) = x$$

και η γραφική της παράσταση είναι η :

$$\phi(x) = x$$

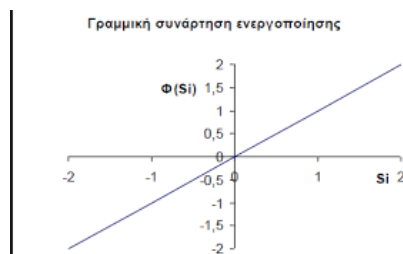
και η γραφική της παράσταση είναι :



Υπερβολική εφαπτομένη

$$\phi(x) = \tanh x$$

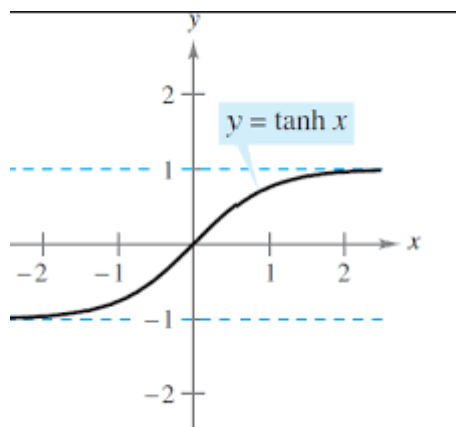
και η γραφική παράσταση είναι η :



Υπερβολική εφαπτομένη

$$\phi(x) = \tanh x$$

και η γραφική παράσταση είναι η :



2.7 Τα Τεχνητά Νευρωνικά Δίκτυα ως κατευθυνόμενοι γράφοι

Κάθε τεχνητό νευρωνικό δίκτυο μπορεί να θεωρηθεί και ως ένας κατευθυνόμενος γράφος. Πρόκειται για έναν γράφο, ο οποίος απαρτίζεται από κόμβους συναπτικών συνδέσεων και ενεργοποίησης. Πιο συγκεκριμένα, ένα νευρωνικό δίκτυο αναπαριστάνεται με ένα πλήθος γραμμικών συναπτικών διασυνδέσεων, μια μη-γραμμική σύνδεση για την ενεργοποίηση του καθώς και με ένα κατώφλι.

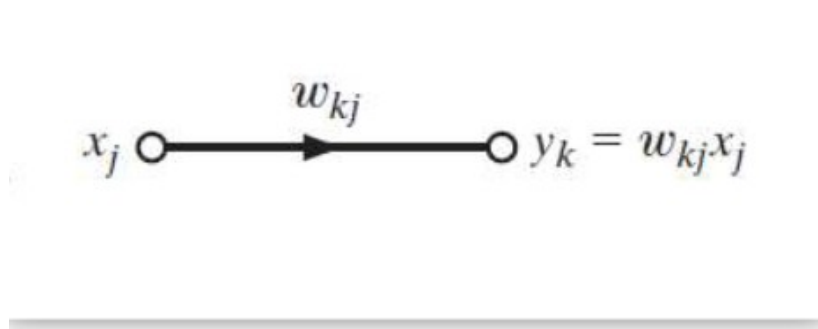
Η θεώρηση αυτή διαθέτει πολλές και χρήσιμες ιδιότητες. Με σκοπό την απλοποίηση όσον αφορά την εμφάνιση ενός τεχνητού νευρώνα, γίνεται χρήση των γράφων ροής

σημάτων. Κάθε σήμα μέσα στο Τεχνητό Νευρωνικό δίκτυο ρέει προς την μοναδική κατεύθυνση που υποδεικνύει το βέλος.

Διακρίνονται δύο διαφορετικοί τύποι συνδέσεων, οι οποίοι είναι οι εξής:

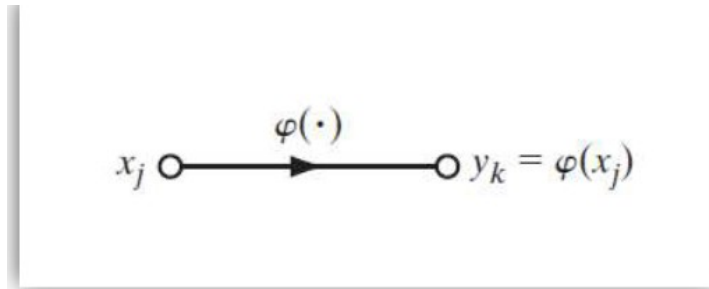
Οι συναπτικές συνδέσεις

Στην περίπτωση αυτή το σήμα για να παραχθεί η έξοδος ή το τελικό σήμα y_k πολλαπλασιάζουμε το ένα σήμα x_j με το βάρος του w_{kj} όπως φαίνεται στο παρακάτω σχήμα :



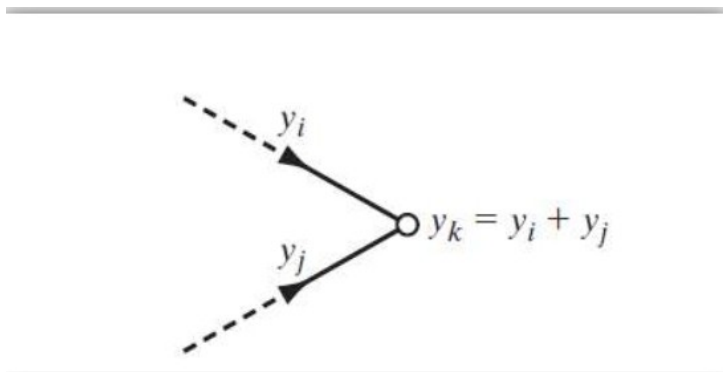
Συνδέσεις ενεργοποίησης

Σε αυτήν την περίπτωση, με τις συνδέσεις ενεργοποίησης αναφερόμαστε σε μια συνάρτηση, η οποία συμβολίζεται με $\Phi(\cdot)$, οποία είναι μη γραμμική όπως φαίνεται στο παρακάτω σχήμα:



Κατευθυνόμενοι γράφοι

Μια ακόμα σημαντική ιδιότητα σε ένα νευρωνικό δίκτυο είναι ότι ένα σήμα κόμβου είναι ίσο με το άθροισμα των σημάτων που καταλήγουν στον συγκεκριμένο κόμβο όπως απεικονίζεται στο παρακάτω σχήμα:



Το σήμα σε έναν κόμβο μεταβιβάζεται σε κάθε εξερχόμενη εξερχόμενη σύνδεση που ξεκινά από αυτόν, με την μεταβίβαση να είναι ανεξάρτητη από τις συναρτήσεις μεταφοράς των εξερχόμενων συνδέσεων όπως φαίνεται στο σχήμα που ακολουθεί:



Από τα παραπάνω προκύπτει ότι ένα τεχνητό νευρωνικό δίκτυο είναι ένας κατευθυνόμενος γράφος, που αποτελείται από κόμβους με συναπτικές και συνδέσεις ενεργοποίησης και έχει τις παρακάτω ιδιότητες :

α) Κάθε νευρώνας, παριστάνεται από ένα σύνολο γραμμικών συναπτικών συνδέσεων, ένα εξωτερικά εφαρμοζόμενο κατώφλι και μια μη-γραμμική σύνδεση ενεργοποίησης .

β) Το κατώφλι παριστάνεται από συναπτικές συνδέσεις με σήμα εισόδου τιμής -1.

γ) Οι συναπτικές συνδέσεις ενός νευρώνα ζυγίζουν τα αντίστοιχα σήματα εισόδου.

δ) Το άθροισμα των βαρών των σημάτων εισόδου καθορίζει συνολικό εσωτερικό επίπεδο ενεργοποίησης του νευρώνα που ζητείται .

ε) Η σύνδεση ενεργοποίησης περιορίζει το εσωτερικό επίπεδο ενεργοποίησης, για την παραγωγή της εξόδου που παριστάνει την κατάσταση του νευρώνα.

2.8 Ομοιότητα ΤΝΔ με τον ανθρώπινο εγκέφαλο

Το ΤΝΔ ομοιάζει με τον ανθρώπινο εγκέφαλο για δύο λόγους:

α) Η γνώση αποκτιέται από το δίκτυο με τη διαδικασία της μάθησης (δοκιμή και σφάλμα). Οι γνώσεις και οι εμπειρίες που αποκτούν τόσο τα νευρωνικά δίκτυα όσο και ο εγκέφαλος των ζωντανών οργανισμών λαμβάνονται από το περιβάλλον τους αντίστοιχα μέσα από μια διαδικασία εκπαίδευσης τους.

β) Τόσο στα Νευρωνικά δίκτυα όσο και στον ανθρώπινο εγκέφαλο υπάρχουν οι αντίστοιχες συνδέσεις . Οι συνδέσεις αυτές γνωστές και ως συνάψεις ή συναπτικά βάρη χρησιμοποιούνται για την αποθήκευση της γνώσης.

Όπως έχουμε αναφέρει παραπάνω ,η έμπνευση των τεχνητών νευρωνικών δικτύων προέρχεται από τα νευρωνικά δίκτυα των ζώντων οργανισμών και ιδιαίτερα του ανθρώπου. Κατά αναλογία με ένα Βιολογικό Νευρωνικό Δίκτυο ένα Τεχνητό αποτελείται από ένα σύνολο τεχνητών νευρώνων που αλληλεπιδρούν, συνδεόμενοι μεταξύ τους με τις συνάψεις .

Στον ανθρώπινο εγκέφαλο οι αριθμοί των μονάδων των νευρώνων και οι συνδέσεις τους είναι πράγματι πολύ μεγάλοι. Περιέχει κατά προσέγγιση 100 δισεκατομμύρια νευρικών κυττάρων, 3.2 εκατομμύρια χιλιόμετρα «καλωδίων» και ένα εκατομμύρια δισεκατομμύρια διασυνδέσεις. Σε αυτό το σημείο τα ΤΝΔ είναι φυσικό να υστερούν σε μεγάλο βαθμό, καθώς ένα ΤΝΔ μπορεί να έχει μερικές εκατοντάδες ή χιλιάδες νευρώνες. Ωστόσο παραμένει εντυπωσιακό το γεγονός ότι ένα ΤΝΔ μπορεί να λύσει

δύσκολα μαθηματικά προβλήματα, τα οποία δεν μπορεί ο ανθρώπινος εγκέφαλος. Μια ακόμα βασική τους διαφορά είναι το ότι στα Βιολογικά Νευρωνικά Δίκτυα οι συνάψεις και η συνδεσμολογία είναι πολύ πιο περίπλοκες σε σχέση με τα αντίστοιχα

των Τεχνητών. Επιπλέον η διαδικασία μάθησης του ανθρώπινου εγκεφάλου είναι γρηγορότερη από εκείνη ενός Τεχνητού Νευρωνικού δικτύου. Αυτό μπορεί να οφείλεται εν μέρει στη μαζική παράλληλη δομή των βιολογικών νευρωνικών δικτύων την οποία προσομοιώνουν και τα τεχνητά νευρωνικά δίκτυα.

2.9 Αντιστοιχία Βιολογικών και Τεχνητών Νευρώνων

Η λειτουργία ενός τεχνητού νευρωνικού δικτύου μιμείται κατά το δυνατό εκείνη του αντίστοιχου βιολογικού νευρωνικού δικτύου, ώστε να εμφανίζονται παρόμοιες ιδιότητες και κατά την επεξεργασία δεδομένων. Εφόσον μιμείται το βιολογικό νευρωνικό δίκτυο υπάρχει μία αντιστοιχία μεταξύ των βιολογικών και τεχνητών νευρώνων. Οπότε τα δομικά στοιχεία

του βιολογικού νευρώνα που περιγράψαμε εκτενώς στο 1ο κεφάλαιο αντιστοιχούν σε δομικά στοιχεία του τεχνητού νευρώνα.

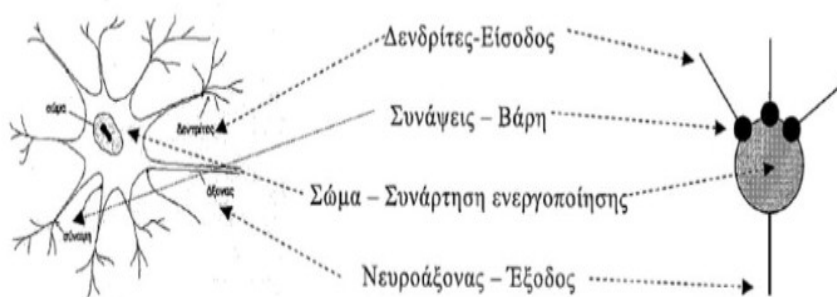
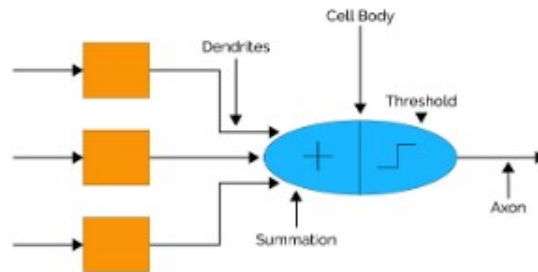
Παραπάνω περιγράψαμε την ομοιότητα των τεχνητών νευρωνικών δικτύων με τα βιολογικά νευρωνικά δίκτυα. Στη συνέχεια παραθέτουμε την αναλογία βιολογικών και τεχνητών νευρώνων.

Οι δενδρίτες αντιστοιχούν σε σταθμισμένες, με συναπτικά βάρη, εισόδους, στον τεχνητό νευρώνα.

Το σώμα του νευρικού κυττάρου αντιστοιχεί στην υπολογιστική μονάδα του τεχνητού νευρώνα, η οποία περιλαμβάνει τη μονάδα άθροισης και τη συνάρτηση ενεργοποίησης(συνάρτηση κατωφλιού) .

Ο άξονας αντιστοιχεί στην έξοδο του τεχνητού νευρώνα.

Τα παραπάνω περιγράφονται στα ακόλουθα σχήματα.



Αναλογία βιολογικού με τεχνητό νευρώνα.

2.10 Πλεονεκτήματα των νευρωνικών δικτύων

Το επιστημονικό ενδιαφέρον για τα Τεχνητά Νευρωνικά Δίκτυα προκύπτει από τη δυνατότητά τους να επιλύουν δύσκολα και ενδιαφέροντα υπολογιστικά προβλήματα του πραγματικού κόσμου. Η χρήση και των τεχνητών νευρωνικών δικτύων έχει τα ακόλουθα χρήσιμα πλεονεκτήματα:

α) Μη-γραμμικότητα

Οι νευρώνες γενικά είναι μη γραμμικές οντότητες, αφού βασίζονται σε μη γραμμικές συναρτήσεις ενεργοποίησης και κατά συνέπεια αφού το ΤΝΔ αποτελείται από τη σύνθεση πολλών νευρώνων, είναι μη γραμμικό. Αυτό είναι

σημαντικό όταν ο μηχανισμός για την παραγωγή σημάτων εισόδου είναι μη γραμμικός.

β) Απομνημόνευση πληροφοριών

Ένα άλλο σημαντικό πλεονέκτημα των νευρωνικών δικτύων είναι η δυνατότητα αποθήκευσης της εμπειρίας και της γνώσης από το περιβάλλον τους. Τις αποθηκευμένες αυτές πληροφορίες μπορεί να τις ανακαλέσει. Επίσης είναι σημαντικό ότι τα συγκεκριμένα δίκτυα έχουν τη δυνατότητα να εξάγουν ορισμένα βασικά στοιχεία ενός συστήματος, ακόμα και αν αυτά βρίσκονται σε θορυβώδη δεδομένα.

γ) Προσαρμοστικότητα

Πλεονέκτημα των τεχνητών νευρωνικών δικτύων είναι και η ικανότητα της προσαρμοστικότητας. Όλα τα δίκτυα έχουν την δυνατότητα να εφαρμόζουν αλλαγές στα βάρη τους όταν παρατηρούνται μεταβολές στο περιβάλλον τους. Ένα δίκτυο, το οποίο ήταν εκπαιδευμένο να εκτελεί λειτουργίες σε ένα συγκεκριμένο περιβάλλον, μπορεί με μεγάλη ευκολία να εκπαιδευτεί εκ νέου προκειμένου να υλοποιήσει μικρές αλλαγές στις λειτουργίες του. (Haykin, 2009). Επιπλέον, ένα νευρωνικό δίκτυο μπορεί να σχεδιαστεί εξ αρχής με σκοπό να μπορεί να μεταβάλλει σε πραγματικό χρόνο τα συναπτόμενα βάρη του. Έτσι είναι δυνατόν να συνεχίσει να εκπαιδεύεται για να αντιμετωπίσει μια μικρή αλλαγή των προτύπων ή ακόμα και μη στατικά προβλήματα.

δ) Απόκριση βασισμένη σε ενδείξεις

Τα εκπαιδευόμενα νευρωνικά δίκτυα μπορούν όχι μόνο να ταξινομήσουν και να τοποθετούν τα πρότυπα εισόδου σε κλάσεις, αλλά επιπρόσθετα δίνουν και τον

βαθμό εμπιστοσύνης αυτής της απόφασης. Έτσι μπορούν να ταξινομήσουν και νέα, άγνωστα κατά τη διάρκεια της εκπαίδευσης πρότυπα. Αυτή η ικανότητα των τεχνητών νευρωνικών δικτύων ονομάζεται γενίκευση (generalization). Επομένως

γενίκευση είναι η ικανότητα ενός νευρωνικού δικτύου να ταξινομεί πρότυπα τα οποία δεν γνώριζε προηγουμένως, ανιχνεύοντας τις ιδιότητες τους.

ε) *Ανοχή στα λάθη*

Στα συγκεκριμένα δίκτυα παρατηρείτε μεγάλη ανοχή στα λάθη. Συγκεκριμένα, κάνοντας σωστούς και κατάλληλους υπολογισμούς, πραγματοποιούνται λειτουργίες χωρίς μεγάλα σφάλματα. Επίσης αυτό σημαίνει ότι η κακή λειτουργία ή η καταστροφή ενός νευρώνα ή κάποιων συνδέσεων δεν είναι ικανή να διαταράξει σημαντικά την λειτουργία τους γιατί η πληροφορία που περιέχουν δεν είναι εντοπισμένη σε συγκεκριμένο σημείο αλλά διάχυτη σε όλο το δίκτυο.

στ) *Ομοιομορφία Ανάλυσης και Σχεδιασμού.*

Όλα τα μοντέλα τεχνητών νευρωνικών δικτύων μοιράζονται κάποιες βασικές αρχές, όπως την έννοια του νευρώνα, των συνδέσεων, της εκπαίδευσης. Αποτέλεσμα αυτού είναι η ευκολότερη διασπορά ιδεών μεταξύ των μελετητών.

ζ) *Αναλογία με την νευρολογία*

Η κατασκευή και η λειτουργία των τεχνητών νευρωνικών δικτύων είναι εμπνευσμένη από τον ανθρώπινο εγκέφαλο. Έτσι οι νευρολόγοι, συχνά μελετούν τα τεχνητά νευρωνικά δίκτυα για να καταλάβουν καλύτερα την λειτουργία του

ανθρώπινου εγκεφάλου και του ανθρώπινου νευρικού συστήματος. Ομοίως οι μηχανικοί τροφοδοτούνται από την νευρολογία με νέες ιδέες για την αντιμετώπιση δύσκολων και πολύπλοκων προβλημάτων

η) Συσχέτιση εισόδου εξόδου

Κατά την εκπαίδευση, παρουσιάζονται στο τεχνητό νευρωνικό δίκτυο δεδομένα εισόδου ή εκπαίδευσης και οι αντίστοιχες επιθυμητές έξοδοι. Σκοπός είναι το νευρωνικό δίκτυο να φτάσει σε μια τέτοια κατάσταση όπου για κάθε πρότυπο εκπαίδευσης η έξοδος του να ταυτίζεται με την επιθυμητή έξοδο. Έτσι δημιουργείται μια συσχέτιση μεταξύ των δεδομένων εισόδου και εξόδου, χωρίς όμως τη χρήση κάποιου προκαθορισμένου στατιστικού ή άλλου μοντέλου .

2.11 Μειονεκτήματα των τεχνητών νευρωνικών δικτύων

Εκτός από τα πλεονεκτήματα τα νευρωνικά δίκτυα έχουν και ορισμένα μειονεκτήματα , τα οποία είναι τα ακόλουθα:

α) Δεν υπάρχουν σαφείς κανόνες για την ανάπτυξη ενός τεχνητού νευρωνικού δικτύου για οποιαδήποτε εφαρμογή

Αυτό σημαίνει ότι δεν υπάρχουν σαφείς κανόνες που να ορίζουν πόσα

κρυφά στρώματα πρέπει να έχει ένα νευρωνικό δίκτυο , πόσους νευρώνες πρέπει να έχει το κάθε στρώμα και ποιές συναρτήσεις ενεργοποίησης πρέπει να χρησιμοποιηθούν.

β) Δεν υπάρχει κάποιος γενικός τρόπος για την ερμηνεία της εσωτερικής λειτουργίας του δικτύου.

Λέγεται πως ακολουθούν την λογική του «μαύρου κουτιού», για αυτό και υπάρχει δυσκολία στην εξήγηση της αιτίας για την οποία ένα δίκτυο φτάνει σε μια συγκεκριμένη λύση και όχι σε κάποια άλλη.

γ) Τα τεχνητά νευρωνικά δίκτυα δεν προτιμούνται σε περιπτώσεις στις οποίες υπάρχουν μαθηματικές ή αλγοριθμικές μέθοδοι επίλυσης.

Αυτό συμβαίνει γιατί ο χρόνος που απαιτείται για τον σχεδιασμό τους είναι αυξημένος. Τέλος, υπάρχουν περιπτώσεις που η εκπαίδευση τους μπορεί να γίνει δύσκολη ή ακόμα και αδύνατη.

ΚΕΦΑΛΑΙΟ 3 -Μηχανική Μάθηση

3.1 Τι είναι μάθηση

Η μάθηση (learning) είναι μία από τις θεμελιώδεις ιδιότητες της νοήμονος συμπεριφοράς του ανθρώπου και είναι ένα σύνθετο εσωτερικό βιολογικό και πνευματικό φαινόμενο που έχει μελετηθεί από διάφορους κλάδους της επιστήμης όπως ψυχολογία, παιδαγωγική, φυσιολογία, ιατρική, βιολογία και άλλοι . Όπως έχει παρατηρηθεί η μάθηση είναι μια πολύπλοκη διαδικασία η οποία δεν έχει ερμηνευθεί και κατανοηθεί πλήρως και κατά τρόπο παραδεκτό από όλους όσους ασχολούνται με αυτή. Στην πραγματικότητα όσα λέγονται και γράφονται για την μάθηση αποτελούν επιστημονικές υποθέσεις οι οποίες προκύπτουν από την από την παρατήρηση και τη μελέτη των αποτελεσμάτων της.

Γενικά υπάρχει μια μεγάλη διάσταση απόψεων μεταξύ των μελετητών για την προσδιορισμό της έννοιας της μάθησης. Ένας ορισμός που περιλαμβάνει αρκετές ιδιότητες της και μπορεί να θεωρηθεί αντιπροσωπευτικός είναι ο παρακάτω :

Μάθηση είναι η διεργασία πρόσληψης και απόκτησης πληροφοριών ή ικανοτήτων συμπεριφοράς και η οποία απαιτεί τη δραστηριότητα συγκεκριμένων εγκεφαλικών περιοχών, εξαρτάται από ορισμένες νευρολογικές δραστηριότητες και κυρίως, συνεπάγεται με τη δημιουργία βιολογικών αλλαγών στα εγκεφαλικά κύτταρα και βιολογικά νευρωνικά δίκτυα. Η μάθηση μπορεί να αφορά την απόκτηση πληροφορίας και δημιουργία γνώσεων με ενσυνείδητο τρόπο, ενώ υπάρχουν μορφές μάθησης που δεν απαιτούν τη λειτουργία της ενσυνείδητης επίγνωσης επί της διεργασίας πρόσληψης των γνώσεων και των πληροφοριών.

3.2 Τι είναι η μηχανική μάθηση

Ο άνθρωπος έχει τη δυνατότητα να οργανώνει και να συσχετίζει τις εμπειρίες και τις παραστάσεις του δημιουργώντας νέες δομές που ονομάζονται πρότυπα (patterns).

Μηχανική μάθηση είναι το πεδίο της επιστήμης των υπολογιστών που έχει ως σκοπό την δημιουργία προτύπων ή μοντέλων από ένα σύνολο δεδομένων από ένα υπολογιστικό σύστημα.

Επίσης η μηχανική μάθηση είναι μέθοδος αυτοματοποίησης της ανάλυσης δεδομένων, όντας τομέας της τεχνητής νοημοσύνης, βασιζόμενο στην ιδέα πως τα συστήματα δύναται να εκπαιδευτούν από δεδομένα, να αναγνωρίσουν μοτίβα και να λάβουν αποφάσεις με την ελάχιστη ανθρώπινη παρέμβαση.

Στο πεδίο της ανάλυσης δεδομένων, η μηχανική μάθηση είναι μια μέθοδος που χρησιμοποιείται για την επινόηση πολύπλοκων μοντέλων και αλγορίθμων που

οδηγούν στην πρόβλεψη. Τα αναλυτικά μοντέλα επιτρέπουν στους ερευνητές, τους επιστήμονες δεδομένων, τους μηχανικούς και τους αναλυτές να παράγουν αξιόπιστες αποφάσεις και αποτελέσματα και να αναδείξουν αλληλοσυσχετίσεις μέσω της μάθησης από ιστορικές σχέσεις και τάσεις στα δεδομένα.

Κατά καιρούς έχουν δοθεί διάφοροι ορισμοί για τη μηχανική μάθηση:

Arthur Samuel(1959), «η μηχανική μάθηση είναι ένα πεδίο έρευνας το οποίο δίνει τη δυνατότητα στους υπολογιστές να μάθουν χωρίς να προγραμματίζονται επακριβώς»

Simon(1983), «η μάθηση σηματοδοτεί προσαρμοστικές αλλαγές σε ένα σύστημα με την έννοια ότι αυτές του επιτρέπουν να κάνει την ίδια εργασία,

ή εργασίες της ίδιας κατηγορίας, πιο αποδοτικά και αποτελεσματικά την επόμενη φορά»

Minsky (1985), «μάθηση είναι να κάνουμε χρήσιμες αλλαγές στο μυαλό μας»

Michalski(1986), «μάθηση είναι η δημιουργία ή η αλλαγή της αναπαράστασης των εμπειριών»

Mitchell, T. (1997), « Ένα πρόγραμμα υπολογιστή μαθαίνει από μια εμπειρία E σε συνάρτηση με κάποιες κλάσεις διεργασιών T και μέτρο απόδοσης P , εάν η απόδοση του στις διεργασίες T , σύμφωνα με το μέτρο P , βελτιώνεται με την εμπειρία E »

3.3 Είδη μηχανικής μάθησης

Με την πάροδο του χρόνου έχουν αναπτυχθεί πολλές τεχνικές μηχανικής μάθησης, οι οποίες ανήκουν σε ένα από τα παρακάτω κύρια είδη:

Μάθηση με επίβλεψη (supervised learning) ή μάθηση με παραδείγματα (learning from examples) ή αλλιώς επιτηρούμενη μάθηση.

Σε αυτό το είδος μάθησης το σύστημα καλείται να "μάθει" μια έννοια ή μια συνάρτηση από ένα σύνολο δεδομένων, η οποία αποτελεί περιγραφή ενός μοντέλου. Ένας τέτοιος αλγόριθμος μάθησης παρουσιάζει πολλά παραδείγματα που έχουν επισημανθεί από τον άνθρωπο. Πιο συγκεκριμένα σύστημα δέχεται τις παραδειγματικές εισόδους καθώς και τα επιθυμητά αποτελέσματα από έναν «δάσκαλο», και ο στόχος είναι να μάθει έναν γενικό κανόνα προκειμένου να αντιστοιχίσει τις εισόδους με τα αποτελέσματα.

Για παράδειγμα, στο πρόβλημα ανίχνευσης αντικειμένων χρησιμοποιούμε εικόνες κατάρτισης όπου οι άνθρωποι έχουν επισημάνει τις θέσεις και τις κατηγορίες σχετικών αντικειμένων. Αφού εκπαιδευτεί από τα παραδείγματα, ο αλγόριθμος είναι σε θέση να προβλέψει τα δεδομένα που δεν είχε δει ποτέ.

Μάθηση χωρίς επίβλεψη (unsupervised learning) ή μάθηση από παρατήρηση (learning from observation) ή μη επιτηρούμενη μάθηση.

Στη μάθηση χωρίς επίβλεψη το σύστημα μας πρέπει μόνο του να ανακαλύψει συσχετίσεις ή ομάδες σε ένα σύνολο δεδομένων, δημιουργώντας πρότυπα, χωρίς να είναι γνωστό αν υπάρχουν, πόσα είναι και ποια είναι.

Σε αυτό το είδος μάθησης, ο αλγόριθμος επιχειρεί να μάθει χρήσιμες ιδιότητες των δεδομένων, χωρίς να αναφέρει ο άνθρωπος-εκπαιδευτής ποια θα είναι η σωστή έξοδος. Εφόσον τα παραδείγματα τα οποία χρησιμοποιούνται δεν είναι χαρακτηρισμένα, δεν υπάρχει σφάλμα για να αξιολογηθούν οι πιθανές λύσεις. Το κλασικό παράδειγμα της μη εποπτευόμενης μάθησης είναι η ομαδοποίηση (clustering).

Η ενισχυτική μάθηση (reinforcement learning)

Η ενισχυτική μάθηση είναι ένα είδος μηχανικής μάθησης όπου το σύστημα μάθησης προσπαθεί να μάθει μέσα από την άμεση αλληλεπίδραση με το περιβάλλον. Το σύστημα έχει ως στόχο τη μεγιστοποίηση της "ανταμοιβής" που λαμβάνει. Η τεχνική αυτή είναι πολύ σημαντική για τα σημερινά παιχνίδια. Ο πράκτορας (agent) είναι η οντότητα που μαθαίνει. Οτιδήποτε άλλο διαφορετικό από αυτό είναι αποτελεί μέρος του περιβάλλοντος. Περιβάλλον και πράκτορας αλληλεπιδρούν συνεχώς. Το πρώτο επιστρέφει "ανταμοιβές" στο δεύτερο και ο δεύτερος επιλέγει ενέργειες. Ο πράκτορας μαθαίνει από τις προηγούμενες εμπειρίες του κάτι το οποίο μακροπρόθεσμα του επιτρέπει να επιλέγει τις βέλτιστες ενέργειες και κινήσεις του.

Αυτό επιτυγχάνεται με τη βοήθεια των "ανταμοιβών" που λαμβάνονται για την κατάσταση που βρίσκεται ο πράκτορας.

Μεταξύ της επιτηρούμενης και της μη επιτηρούμενης μάθησης είναι η ημι-επιτηρούμενη μάθηση, όπου ο δάσκαλος δίνει ένα ελλιπές εκπαιδευτικό σήμα: ένα σύνολο εκπαίδευσης με κάποια (συχνά πολλά) από τα αποτελέσματα στόχους να λείπουν. Η Μεταγωγή είναι μια ειδική περίπτωση της αρχής αυτής, όπου το σύνολο

των καταστάσεων του προβλήματος είναι γνωστό κατά το χρόνο εκμάθησης, όμως ένα μέρος των στόχων λείπουν.

3.4 Μάθηση - Εκπαίδευση νευρωνικών δικτύων

Σημαντική ιδιότητα των τεχνητών νευρωνικών δικτύων είναι η ικανότητά τους να μαθαίνουν από το περιβάλλον τους και να βελτιώνουν την απόδοσή τους μέσω της διαδικασίας μάθησης. Η βελτίωση αυτή γίνεται στην διάρκεια του χρόνου με κάποιο προκαθορισμένο μέτρο. Ένα τεχνητό νευρωνικό δίκτυο μαθαίνει από το περιβάλλον του μέσω μιας επαναληπτικής διαδικασίας προσαρμογών των βαρών στις συνάψεις του. Καθώς το τεχνητό νευρωνικό δίκτυο αλληλεπιδρά με το περιβάλλον του και μαθαίνει από αυτό, τα συναπτικά βάρη μεταβάλλονται, ενδυναμώνοντας ή αποδυναμώνοντας την ισχύ του κάθε δεσμού.

Στην ιδανική περίπτωση, μετά από κάθε επανάληψη της διαδικασίας μάθησης το τεχνητό νευρωνικό δίκτυο αποκτά περισσότερη γνώση για το περιβάλλον του.

Υπάρχουν πάρα πολλές δραστηριότητες που σχετίζονται με την έννοια της μάθησης. Η έννοια της μάθησης στον άνθρωπο έχει οριστεί στην προηγούμενη ενότητα.

Ορίζουμε την διαδικασία μάθησης στο πλαίσιο των τεχνητών νευρωνικών δικτύων ως :

Μάθηση είναι η διαδικασία με την οποία οι ελεύθερες παράμετροι ενός τεχνητού νευρωνικού δικτύου προσαρμόζονται μέσω μιας συνεχούς διαδικασίας λήψης ερεθισμάτων από το περιβάλλον στο οποίο είναι ενσωματωμένο το τεχνητό νευρωνικό δίκτυο. Ο τύπος της μάθησης καθορίζεται από τον τρόπο με τον οποίο γίνονται οι αλλαγές στις παραμέτρους.

Ο ορισμός αυτός της διαδικασίας μάθησης, συνεπάγεται την παρακάτω σειρά γεγονότων :

- α) Το τεχνητό νευρωνικό δίκτυο διεγείρεται από το περιβάλλον στο οποίο βρίσκεται.
- β) Οι τιμές των βαρών ενός τεχνητού νευρωνικού δικτύου μεταβάλλονται ως αποτέλεσμα των ερεθισμάτων που δέχεται από το περιβάλλον του.
- γ) Το τεχνητό νευρωνικό δίκτυο αντιδρά με διαφορετικό τρόπο στο περιβάλλον, λόγω των αλλαγών που επήλθαν στην εσωτερική του δομή και ειδικότερα στα βάρη των νευρώνων του.
- Δηλαδή η εμπειρική γνώση που λαμβάνει το τεχνητό νευρωνικό δίκτυο από κωδικοποιείται και αποθηκεύεται στα βάρη του.

Για την εκπαίδευση των τεχνητών νευρωνικών δικτύων, δηλαδή για τον τρόπο με τον οποίο θα μεταβάλλονται οι τιμές των βαρών του, έτσι ώστε να επιτυγχάνεται η σωστή έξοδος για το τεχνητό νευρωνικό δίκτυο ανάλογα με τα δεδομένα που εισέρχονται σε αυτό χρησιμοποιούνται οι αλγόριθμοι εκπαίδευσης.

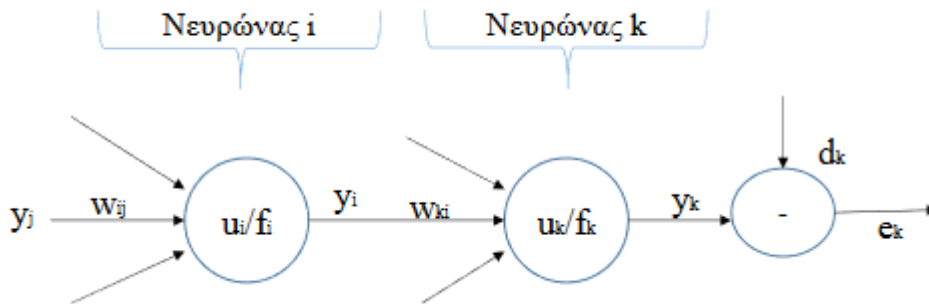
Αλγόριθμος εκπαίδευσης, καλείται ένα προκαθορισμένο σύνολο από κανόνες που χρησιμοποιούνται για την επίλυση ενός προβλήματος εκμάθησης.

Ας θεωρήσουμε ένα πολυστρωματικό νευρωνικό δίκτυο με L κρυφά στρώματα. Το στρώμα εισόδου και κάθε ένα από τα κρυφά στρώματα περιέχουν k_0 και k_i , $i = 1, 2, \dots, L$ νευρώνες αντίστοιχα. Ο νευρώνας I ανήκει

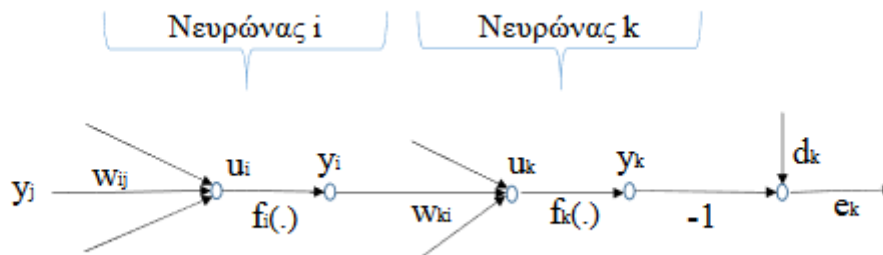
στο κρυφό επίπεδο I και ο νευρώνας k ανήκει στο κρυφό επίπεδο εξόδου L .

Το πλήθος των νευρώνων που υπάρχουν στο κρυφό επίπεδο I είναι m .

Τα παραπάνω φαίνονται στο παρακάτω σχήμα.



Το γράφημα που παρουσιάζει την ροή του σήματος του πολυστρωματικού τεχνητού νευρωνικού δικτύου είναι το παρακάτω:



Το σήμα σφάλματος για ένα κρυφό νευρώνα καθορίζεται αναδρομικά με οπισθοδιάδοση βάσει των σφαλμάτων όλων των νευρώνων επιπέδου $l+1$ με τους οποίους συνδέεται άμεσα ο κρυφός νευρώνας. Η τοπική βαθμίδα σφάλματος δ του κόμβου i υπολογίζεται από τον παρακάτω τύπο:

$$\delta_i(l) = -\frac{\partial J}{\partial u_i} = -\frac{\partial J}{\partial y_i} \cdot \frac{\partial y_i}{\partial u_i} = -\frac{\partial J}{\partial y_i} f'_i(u_i)$$

Η συνάρτηση κόστους J ορίζεται ως εξής:

$$J = \frac{1}{2} \sum_k e_k^2 \rightarrow \frac{\partial J}{\partial y_i} = \sum_k e_k \cdot \frac{\partial e_k}{\partial y_i} = \sum_k e_k \cdot \frac{\partial e_k}{\partial u_k} \cdot \frac{\partial u_k}{\partial y_i}$$

Στη συνέχεια υπολογίζουμε τις μερικές παραγώγους που εμφανίζονται στο άθροισμα μετά την εφαρμογή του κανόνα της αλυσίδας. Οι υπολογισμοί αυτοί περιγράφονται από τις παρακάτω εξισώσεις:

$$\left. \begin{aligned} e_k &= d_k - y_k = d_k - f(u_k) \rightarrow \frac{\partial e_k}{\partial u_k} = -f'_k(u_k) \\ u_k &= \sum_{i=0}^m w_{ki} y_i \rightarrow \frac{\partial u_k}{\partial y_i} = w_{ki} \end{aligned} \right\} \rightarrow$$

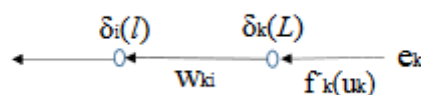
$$\xrightarrow{(*)} \frac{\partial J}{\partial y_i} = - \sum_k e_k \cdot f'_k(u_k) \cdot w_{ki} (**)$$

Η τοπική βαθμίδα του σφάλματος δ ισούται με:

$$\delta_i(l) = - \frac{\partial J}{\partial y_i} \cdot f'_i(u_i) \xrightarrow{(**)} \delta_i(l) = f'_i(u_i) \cdot \sum_k \underbrace{e_k \cdot f'_k(u_k)}_{\delta_k(L)} \cdot w_{ki}$$

$$\delta_i(l) = f'_i(u_i) \cdot \sum_k \delta_k(L) \cdot w_{ki}$$

Στο παρακάτω σχήμα φαίνεται η ροή του σήματος που εκτελεί την οπισθοδιάδοση του σφάλματος



Στη συνέχεια η διόρθωση που εφαρμόζεται στα συναπτικά w βάρη ορίζεται από τον παρακάτω κανόνα:

$$\begin{pmatrix} \text{Διόρθωση} \\ \text{βάρους} \\ \Delta w \end{pmatrix} = \begin{pmatrix} \text{Ρυθμός} \\ \text{μάθησης} \\ \eta \end{pmatrix} \cdot \begin{pmatrix} \text{Τοπική} \\ \text{βαθμίδα} \\ \delta \end{pmatrix} \cdot \begin{pmatrix} \text{Σήμα εισόδου} \\ \text{του νευρώνα} \\ y \text{ ή } x \end{pmatrix}$$

Η τοπική βαθμίδα δ υπολογίζεται από τους παρακάτω τύπους :

α) Εάν ο νευρώνας k είναι νευρώνας εξόδου ισχύει ο τύπος:

$$\delta_k(L) = (d_k - y_k) \cdot f'(u_k) = e_k \cdot f'(u_k)$$

β) Εάν ο νευρώνας i είναι κόμβος του κρυφού στρώματος l :

$$\delta_i(l) = f'(u_i) \cdot \sum_k \delta_k(L) \cdot w_{ki}$$

Τα βάρη ενημερώνονται από τους παρακάτω τύπους:

Για τα συναπτικά βάρη μεταξύ εξόδου και κρυφού επιπέδου:

$$\Delta w_{ki}(L) = n \cdot \delta_k(L) \cdot y_i$$

Για τα συναπτικά βάρη μεταξύ επιπέδου και επιπέδου εισόδου:

$$\Delta w_{ji}(l) = n \cdot \delta_i(l) \cdot x_j$$

όπου j : ο αριθμός εισόδους με $j=1,\dots,n$

i : ο αριθμός του κρυφού κόμβου $i=1,\dots,r$

k : αριθμός κόμβου εξόδου με $k=1,\dots,m$

l : αριθμός στρώματος με $l=1,\dots,L$

Η εξίσωση 3 έδωσε το όνομά αλγόριθμος της οπισθοδιάδοσης (Back-Propagation) σε όλη αυτήν την μέθοδο. Από αυτόν τον τύπο φαίνεται ότι το σφάλμα δ σε οποιονδήποτε του κρυφού στρώματος είναι συνάρτηση των σταθμισμένων τοπικών σφαλμάτων του στρώματος της εξόδου για ένα τεχνητό νευρωνικό δίκτυο με ένα κρυφό στρώμα και τουλάχιστον δύο εξόδους. Το ίδιο ισχύει όταν υπάρχουν περισσότερα κρυφά στρώματα.

Τα σφάλματα πηγάζουν από το τελευταίο στρώμα που είναι το στρώμα εξόδου και προωθούνται προς τα πίσω δηλαδή μέχρι το πρώτο στρώμα. Τα σφάλματα που προωθούνται προς τα πίσω χρησιμοποιούν τα ίδια βάρη του δικτύου που χρησιμοποιούνται και στην ανάκληση.

Κριτήρια τερματισμού του αλγορίθμου

Ένα κριτήριο τερματισμού του αλγορίθμου της οπισθοδιάδοσης που χρησιμοποιείται ευρύτατα είναι η αξιολόγηση του κόστους J . Για να υπολογίσουμε το $J(n)$ κρατάμε τα σταθερά τα βάρη που προέκυψαν από την n εποχή και τροφοδοτούμε το δίκτυο με όλα τα πρότυπα υπολογίζοντας το παρακάτω σφάλμα για κάθε πρότυπο:

$$e(p) = \|\mathbf{d}^{(p)} - \mathbf{y}^{(p)}\|^2$$

Ως πρότυπο εννοούμε την αντιστοιχία τιμής εισόδου και επιθυμητής τιμής εξόδου.

Στη συνέχεια αθροίζουμε όλα αυτά τα σφάλματα και το άθροισμά τους είναι το ζητούμενο κόστος.

$$J(n) = e(1) + \dots + e(p)$$

Το κόστος $J(n)$ είναι ικανοποιητικό εάν για την εποχή εκπαίδευσης n είναι μικρότερο από ένα κατώφλι ε που ορίζει ο χρήστης.

Ένα άλλο εναλλακτικό κριτήριο τερματισμού είναι η αξιοποίηση των συναπτικών βαρών . Στην n -ιοστή εποχή αθροίζουμε τις απόλυτες τιμές των μεταβολών των τιμών των βαρών και τις συγκρίνουμε με ένα κατώφλι ε που έχει ορίζει ο χρήστης.

Δηλαδή :

$$\sum_{i,j} |\Delta w_{ij}| < \varepsilon$$

(10. Understanding Backpropagation Algorithm- [Simeon Kostadinov](#)-Κατανόηση αλγορίθμου οπισθοδιάδοσης)

Στη συνέχεια ακολουθεί ένα παράδειγμα εκπαίδευσης νευρωνικού δικτύου που εκπαιδεύεται να αναγνωρίζει χειρόγραφους αριθμούς από το 0 μέχρι το 9 .

Αρχικά εισάγουμε την βιβλιοθήκη `numpy` της `python` η οποία παρέχει υποστήριξη για δημιουργία απλών πινάκων, πολυδιάστατων πινάκων και συστοιχιών.

Επίσης παρέχει και ένα μεγάλο αριθμό συναρτήσεων για την υποστήριξη και επεξεργασία των ανωτέρω δομών δεδομένων .

Όπως έχουμε αναφέρει η εκπαίδευση των νευρωνικών δικτύων απαιτεί ένα σύνολο δεδομένων εισόδου και ένα σύνολο επιθυμητών εξόδων , όπου κάθε είσοδος θα αντιστοιχεί σε μία επιθυμητή έξοδο . Στην περίπτωση μας αυτό επιτυγχάνεται με το σύνολο δεδομένων `mnist` . Στο παράδειγμά μας το σύνολο δεδομένων εισόδου είναι το `train_images` και το σύνολο επιθυμητών αποτελεσμάτων είναι το `train_labels` . Το σύνολο των δεδομένων το εισάγουμε από το `tensorflow` με την χρήση της παρακάτω εντολής:

Το `tensorflow` είναι μια δωρεάν βιβλιοθήκη λογισμικού ανοιχτού κώδικα για μηχανική μάθηση. Μπορεί να χρησιμοποιηθεί σε ένα ευρύ φάσμα εργασιών αλλά έχει ιδιαίτερη έμφαση στην εκπαίδευση και την διεξαγωγή συμπερασμάτων σε βαθιά νευρωνικά δίκτυα.

Το `Keras` είναι μια βιβλιοθήκη λογισμικού ανοιχτού κώδικα που παρέχει μια διεπαφή `Python` για τεχνητά νευρικά δίκτυα. Το `Keras` λειτουργεί ως διεπαφή για τη βιβλιοθήκη `TensorFlow`.

Στο παράδειγμα που ακολουθεί χρησιμοποιείται το `tensorflow` και το `keras` και για την σύνταξη του κώδικα χρησιμοποιείται το `visual studio code`.

```
from tensorflow.keras.datasets import mnist
```

```
train_images = mnist.train_images()
train_labels = mnist.train_labels()
```

Στην συνέχεια εκτυπώνουμε τα ανωτέρω σύνολα δεδομένων με την χρήση της εντολής print.

```
print(train_images.shape) # (60000, 28, 28)
print(train_labels.shape)
```

Μετά προετοιμάζουμε τα δεδομένα μας. Συγκεκριμένα κανονικοποιούμε τις τιμές των pixel από το διάστημα $[0, 255]$ στο διάστημα $[-0.5, 0.5]$ για να κάνουμε το δίκτυό μας να μπορεί να εκπαιδευτεί πιο εύκολα. Αυτό επιτυγχάνεται διαιρώντας τα σύνολα `train_images` και `test_images` με το 255 και αφαιρώντας το 0.5. Ο κώδικας είναι ο παρακάτω:

```
train_images = (train_images / 255) - 0.5
test_images = (test_images / 255) - 0.5
```

Επίσης μετατρέπουμε τους ανωτέρου πίνακες σε μονοδιάστατους με την χρήση της συνάρτησης `reshape` της βιβλιοθήκης `numpy`. Η συνάρτηση `reshape` δημιουργεί ένα νέο σχήμα σε έναν πίνακα, χωρίς να αλλάζει τις τιμές των δεδομένων του. Πιο συγκεκριμένα οι πίνακες μετασχηματίζονται ώστε κάθε στήλη τους να περιέχει 784 στοιχεία.

```
train_images = train_images.reshape((-1, 784))
```

```
test_images = test_images.reshape((-1, 784))
```

Αφού έχουμε προετοιμάσει τα δεδομένα μας, στη συνέχεια δημιουργούμε το νευρωνικό δίκτυο που θα εκπαιδεύσουμε.

Χρησιμοποιούμε το σειριακό μοντέλο `sequential model`(σειριακό μοντέλο) του `keras`. Το μοντέλο αυτό δηλώνει ότι το νευρωνικό δίκτυο που θα δημιουργήσουμε λειτουργεί σειριακά, δηλαδή ότι η έξοδος από το κάθε στρώμα είναι η είσοδος στο αμέσως επόμενο στρώμα.

Το `model.add` χρησιμοποιείται για να προσθέσουμε ένα στρώμα στο δίκτυό μας. Το `dense` χρησιμοποιείται για να δηλώσουμε ότι τα στρώματα είναι πλήρη συνδεδεμένα μεταξύ τους, δηλαδή ότι κάθε νευρώνας ενός στρώματος θα συνδέεται με συναπτικό βάρος με όλους τους νευρώνες του επόμενου στρώματος του δικτύου.

Ως ορίσματα του `dense` ορίζουμε τον αριθμό των νευρώνων και την συνάρτηση ενεργοποίησης του κάθε στρώματος. Στην περίπτωσή μας γίνεται χρήση της συνάρτησης `relu`(βηματική συναρτηση) και της συνάρτησης `softmax` (κανονικοποιημένη εκθετική συνάρτηση). Οι δύο αυτές συναρτήσεις χρησιμοποιούνται συχνά στα νευρωνικά δίκτυα. Ο κώδικας είναι ο παρακάτω:

```
# Η είσοδος αποτελείται από 784 τιμές εισόδου και
model = Sequential([ # και το πρώτο στρώμα εισόδου περιέχει 64 νευρώνες
    Dense(64, activation='relu', input_shape=(784,)),
    Dense(64, activation='relu'), # Το δεύτερο στρώμα περιέχει 64 νευρώνες
    Dense(10, activation='softmax'), # Το στρώμα εξόδου περιέχει 10 νευρώνες
])
```

Οπότε το νευρωνικό δίκτυο το οποίο κατασκευάζουμε αποτελείται από τέσσερα στρώματα. Τα δύο στρώματα είναι τα κρυφά στρώματα , τα οποία αποτελούνται από 64 νευρώνες το καθένα , και το τελευταίο στρώμα που είναι το στρώμα εξόδου αποτελείται από 10 νευρώνες.

Εκπαίδευση του νευρωνικού δικτύου

Για να εκπαιδύσουμε το δίκτυό μας καλούμε τη συνάρτηση `fit()` και ορίζουμε κάποιες παραμέτρους. Υπάρχουν αρκετές παράμετροι , όμως στη περίπτωση μας χρησιμοποιούμε τις παρακάτω:

Δεδομένα εκπαίδευσης που είναι τα δεδομένα εισόδου και τα δεδομένα εξόδου.

Αριθμός των εποχών. Μία εποχή είναι όταν το σύνολο εισόδου χρησιμοποιήθηκε για ένα ευθύ πέρασμα και ένα ανάποδο πέρασμα για μία μόνο φορά. Στην περίπτωση μας έχουμε ορίσει τέσσερις εποχές.

Αριθμός των δειγμάτων ή batch size: Ορίζει τον αριθμό των δειγμάτων που παίρνουμε από τα δεδομένα εισόδου για να εκπαιδύσουμε το δίκτυό μας.

Ο αριθμός των δειγμάτων είναι ίσος με τον αριθμό των επαναλήψεων για μία εποχή. Για παράδειγμα εάν έχουμε 2000 δεδομένα για την εκπαίδευση του δικτύου μας και ορίσουμε batch 500 δεδομένων , τότε θα υπάρχουν συνολικά

4 batch και κάθε batch θα χρειαστεί τέσσερις επαναλήψεις. Γενικότερα ισχύει ότι αριθμός επαναλήψεων = δεδομένα / μέγεθος batch.

Αυτά υλοποιούνται με τον παρακάτω κώδικα:

```
model.fit(  
    train_images,
```

```

to_categorical(train_labels),
    epochs=5,      # Ορισμός των εποχών
    batch_size=32, # Ορισμός του batch size
)

```

Στη συνέχεια ορίζουμε και το `compile`. Το `compile` περιέχει την συνάρτηση σφάλματος (`loss function`) που υπολογίζει το σφάλμα, δηλαδή εάν για κάποια τιμή εισόδου το νευρωνικό μας δίκτυο δίνει σωστό αποτέλεσμα, τον βελτιστοποιητή (`optimizer`), ο οποίος είναι μια συνάρτηση που διαμορφώνει και διαμορφώνει το δίκτυό σας στην πιο ακριβή δυνατή μορφή του, κάνοντας κάποιες αλλαγές στα βάρη στα βάρη και την μέτρηση `accuracy` δηλαδή πόσο ακριβές είναι το νευρωνικό μας δίκτυο στις προβλέψεις του. Πιο συγκεκριμένα για να επιτευχθεί αυτό χρησιμοποιούμε το νευρωνικό μας δίκτυο σε ένα σύνολο τιμών για το οποίο γνωρίζουμε τις σωστές προβλέψεις. Στη συνέχεια καταμετρούμε τα λάθος αποτελέσματα και βρίσκουμε τι ποσοστό του συνόλου των τιμών είναι. Έτσι υπολογίζουμε πόσο ακριβές είναι το δίκτυό μας.

Η υλοποίηση όλων των παραπάνω γίνεται με τον παρακάτω κώδικα:

```

model.compile(
    optimizer='adam', # χρησιμοποιούμε και τον βελτιστοποιητή adam
    loss='categorical_crossentropy', # Συνάρτηση σφάλματος
    metrics=['accuracy'], # Υπολογισμός ακρίβειας.
)

```

Ο κώδικας ολοκληρωμένος είναι ο παρακάτω:

```

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

```

```
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.datasets import mnist
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()

train_images = (train_images / 255) - 0.5
test_images = (test_images / 255) - 0.5

# Flatten the images.
train_images = train_images.reshape((-1, 784))
test_images = test_images.reshape((-1, 784))

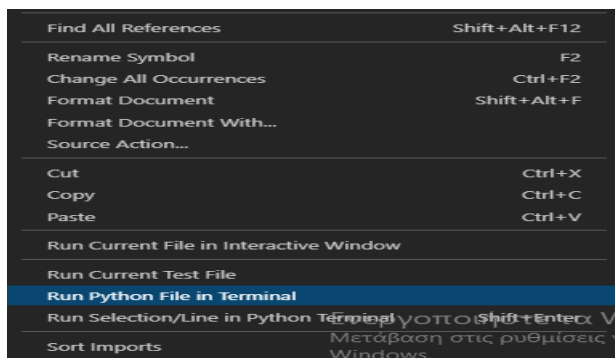
# Build the model.
model = Sequential([
    Dense(64, activation='relu', input_shape=(784,)),
    Dense(64, activation='relu'),
    Dense(10, activation='softmax'),
])

# Compile the model.
model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy'],
)
```



```
# Train the model.  
model.fit(  
    train_images,  
        to_categorical(train_labels),  
    epochs=5,  
    batch_size=32,  
)
```

Για να εκπαιδύσουμε το νευρωνικό μας δίκτυο κάνουμε δεξί κλικ με το ποντίκι ή επάνω στο όνομα του αρχείου ή επάνω στον editor κώδικα και επιλέγουμε το Run Python File in Terminal



```

9  train_images = (train_images / 255) - 0.5
10 test_images = (test_images / 255) - 0.5
11 |
12 # Flatten the images.
13 train_images = train_images.reshape((-1, 784))
14 test_images = test_images.reshape((-1, 784))
15 |
16 # Build the model.
17 model = Sequential([
18     Dense(64, activation='relu', input_shape=(784,)),
19     Dense(64, activation='relu'),
20     Dense(10, activation='softmax'),
21 ])
22 |
23 # Compile the model.
24 model.compile(
25     optimizer='adam',
26     loss='categorical_crossentropy',
27     metrics=['accuracy'],
28 )
29 |
30 # Train the model.
31 model.fit(
32     train_images,
33     to_categorical(train_labels),
34     epochs=5,
35     batch_size=32,
36 )

```

Το αποτέλεσμα είναι το παρακάτω :

```

2020-11-23 22:38:04.161622: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports in
t compiled to use: AVX AVX2
Train on 60000 samples
Epoch 1/5
60000/60000 [=====] - 29s 478us/sample - loss: 0.3485 - accuracy: 0.8957
Epoch 2/5
60000/60000 [=====] - 5s 89us/sample - loss: 0.1804 - accuracy: 0.9443
Epoch 3/5
60000/60000 [=====] - 5s 88us/sample - loss: 0.1403 - accuracy: 0.9569
Epoch 4/5
60000/60000 [=====] - 17s 285us/sample - loss: 0.1207 - accuracy: 0.9630
Epoch 5/5
60000/60000 [=====] - 5s 81us/sample - loss: 0.1052 - accuracy: 0.9669
PS C:\hello_ds>

```

Παρατηρούμε ότι η ακρίβεια του δικτύου βελτιώνεται ως εξής:

1η εποχή: 0.8957

2η εποχή: 0.9443

3η εποχή: 0.9569

4η εποχή: 0.9630

5η εποχή : 0.9669

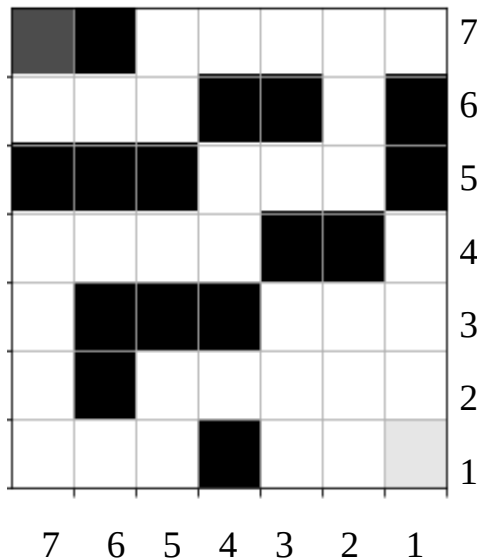
ΚΕΦΑΛΑΙΟ 4 – ΕΦΑΡΜΟΓΕΣ ΤΩΝ ΤΕΝΗΤΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΣΤΗ ΔΙΑΣΧΥΣΗ ΛΑΒΥΡΙΝΘΩΝ

4.1 Διάσχιση λαβυρίνθου με την χρήση νευρωνικού δικτύου.

Στο κεφάλαιο αυτό θα εξετάσουμε την χρήση των νευρωνικών δικτύων στο πρόβλημα του λαβυρίνθου. Το πρόβλημα του λαβυρίνθου αναφέρεται στην διεθνείς αλληλογραφία με τον όρο *rat in maze* (αρουραίος στον λαβύρινθο) ή με τον όρο *maze traversal* (διάσχιση του λαβυρίνθου). Το πρόβλημα αυτό μπορεί να διατυπωθεί ως εξής: Ο πράκτορας γνωρίζει την αρχή του λαβυρίνθου δεν γνωρίζει την δομή του λαβυρίνθου και ψάχνει να την έξοδο του . Κατά μια άλλη παραλλαγή ο πράκτορας γνωρίζει την αρχή και το τέλος του λαβυρίνθου , δεν γνωρίζει την διαδρομή που οδηγεί στο τέλος και προσπαθεί να την βρει.

Για την επίλυση αυτού του προβλήματος έχουν χρησιμοποιηθεί διάφοροι αλγόριθμοι όπως ο αλγόριθμος DFS (Αναζήτηση πρώτα σε Βάθος) , ο αλγόριθμος BFS (Αναζήτηση πρώτα σε πλάτος) και κάποια μαθηματικά συστήματα , όπως είναι τα μοντέλα Markov. Στη συνέχεια θα παρουσιάσουμε δύο προσεγγίσεις επίλυσης του προβλήματος με τη βοήθεια των νευρωνικών δικτύων.

Θεωρούμε ότι κάποιος πράκτορας πρέπει να διασχίσει τον λαβύρινθο που υπάρχει στο σχήμα 4.1 . Αριθμούμε κατακόρυφα και οριζόντια τον λαβύρινθο ώστε να μπορούμε να αναφερόμαστε στα τετράγωνα του όπως αναφερόμαστε στις καρτεσιανές συντεταγμένες π.χ. το τετράγωνο στην τρίτη γραμμή και τέταρτη στήλη ορίζεται ως (3,4). Επίσης ορίζουμε ως αρχή τη θέση (1,1) και σαν τέλος τη θέση (7,7). Ο πράκτορας ξεκινάει από την θέση (1,1) και πρέπει να φτάσει στην θέση (7,7) διασχίζοντας τον λαβύρινθο με την βοήθεια ενός νευρωνικού δικτύου και μπορεί να έχει σαν θέαση το τετράγωνο στο οποίο βρίσκεται. Επίσης μπορεί να έχει θέαση και όλα τα γειτονικά τετράγωνα που βρίσκονται επάνω,κάτω,δεξιά,αριστερά και διαγώνια στο τετράγωνο το οποίο βρίσκεται. Η αρχή του λαβυρίνθου είναι το (1,1) που έχει χρώμα ανοικτό γκρι ενώ το τέλος είναι το (7,7) που έχει χρώμα σκούρο γκρι.

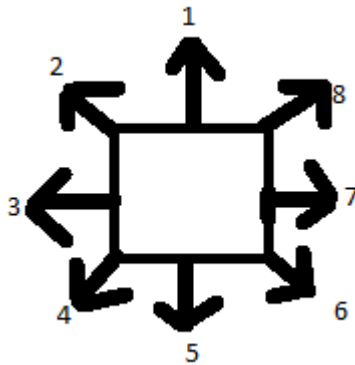


Σχήμα 4.1

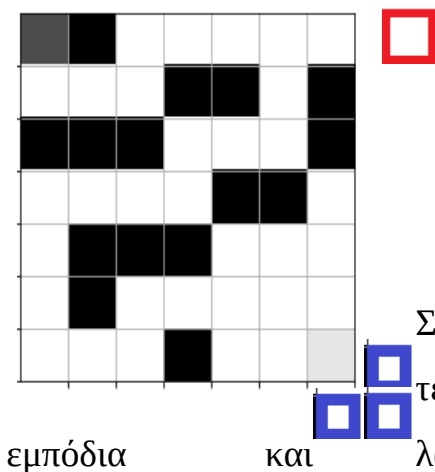
Πρέπει να εκπαιδεύσουμε ένα νευρωνικό δίκτυο το οποίο θα χρησιμοποιεί ο πράκτορας για να βρεί την έξοδο του λαβυρίνθου η οποία βρίσκεται στο (7,7). Όπως έχουμε ήδη αναφέρει η δημιουργία και η εκπαίδευση νευρωνικού δικτύου προϋποθέτει ένα σύνολο δεδομένων (data) και ένα σύνολο αποτελεσμάτων όπου κάθε τιμή του συνόλου δεδομένων θα αντιστοιχεί σε μία τιμή του συνόλου αποτελεσμάτων. Στην περίπτωσή μας το σύνολο των δεδομένων είναι τα οκτώ γειτονικά τετράγωνα για κάθε ελεύθερο τετράγωνο δηλαδή το επάνω , το κάτω , το δεξί , το αριστερό τετράγωνο και τα τέσσερα διαγώνια τετράγωνα. Το σύνολο αποτελεσμάτων είναι το πλησιέστερο από τα οκτώ γειτονικά τετράγωνα προς το (7,7) τετράγωνο. Οπότε κάθε είσοδος

του συνόλου δεδομένων αποτελείται από ένα διάνυσμα οχτώ τιμών που αντιστοιχεί σε μία έξοδο του συνόλου αποτελεσμάτων, η οποία αποτελείται από ένα διάνυσμα που περιέχει μία τιμή στο σύνολο των αποτελεσμάτων. Τα διανύσματα του συνόλου

δεδομένων περιέχουν τις τιμές 0 και 1 όπου το 0 δηλώνει ότι υπάρχει εμπόδιο και ο πράκτορας δεν μπορεί να μεταβεί στο τετράγωνο αυτό και το 1 δηλώνει ότι το τετράγωνο είναι ελεύθερο και ο πράκτορας μπορεί να μεταβεί σε αυτό. Επίσης ως εμπόδιο θεωρούμε και τα εξωτερικά ‘νοητά τετράγωνα’ τα οποία είναι εκτός λαβυρίνθου , όπως π.χ.το (1,1) συνορεύει με τρία τέτοια ‘νοητά τετράγωνα’, ένα κάτω , ένα δεξιά και ένα διαγώνια και το (1,7) έχει ένα τέτοιο τετράγωνο (σχήμα 4.3). Οπότε τα συμβολίζουμε με 0 . Τα διανύσματα του συνόλου αποτελεσμάτων περιέχουν μία τιμή από το 1 έως το 8 που δηλώνει προς πιο τετράγωνο που πρέπει να κινηθεί ο πράκτορας. Η αρίθμηση για τα διανύσματα του συνόλου αποτελεσμάτων γίνεται αντίστροφα από τους δείκτες του ρολογιού , δηλαδή από τα αριστερά προς τα δεξιά όπως φαίνεται στο σχήμα 4.2.



Σχήμα 4.2



εμπόδια και

Νοητά εξωτερικά τετράγωνα.

Σχήμα 4.3(Τα μπλε τετράγωνα στο (1,1) και το κόκκινο τετράγωνο στο (7,1). Τα τετράγωνα αυτά θεωρούνται ως λαμβάνουν την τιμή 0. Κατ’ ουσίαν

θεωρούμε ότι βρίσκεται εκτός του λαβυρίνθου ως εμπόδιο αφού ο πράκτορας δεν μπορεί να μεταβεί εκεί.).

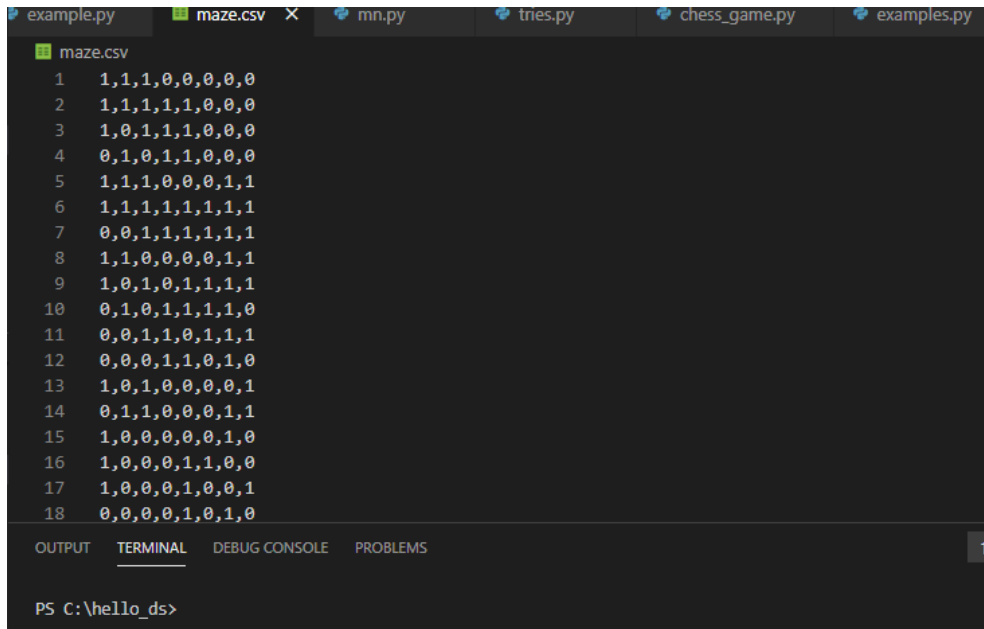
Με βάση όσα έχουμε αναφέρει παραπάνω θα δημιουργήσουμε το σύνολο δεδομένων και το σύνολο αποτελεσμάτων. Για την δημιουργία του συνόλου δεδομένων βρίσκουμε για κάθε ελεύθερο τετράγωνο τα οκτώ γειτονικά του και ελέγχουμε εάν είναι ελεύθερα ή εάν έχουν εμπόδια. Έτσι δημιουργούμε το διάνυσμα εισόδου που αποτελείται από οκτώ τιμές. Στη συνέχεια για να βρούμε την τιμή αποτελέσματος που αντιστοιχεί για την τιμή εισόδου του συγκεκριμένου τετραγώνου ελέγχουμε πιο από τα ανωτέρω γειτονικά τετράγωνα βρίσκεται πιο κοντά στο τέλος του λαβυρίνθου που είναι το τετράγωνο (7,7) και του δίνουμε μία τιμή από το 1 μέχρι το 8 σύμφωνα με το σχήμα 4.2. Παράδειγμα για το (1,2) ισχύει ότι (1,1,1,1,1,0,0,0) και η τιμή του αποτελέσματος είναι 1. Αξίζει να αναφέρουμε ότι για κάποιο διάνυσμα του συνόλου δεδομένων ενδέχεται να αντιστοιχούν δύο τιμές του συνόλου αποτελεσμάτων, όπως συμβαίνει με το (1,1) όπου μπορούμε να θεωρήσουμε την τιμή 1 ως τιμή εξόδου αλλά και την τιμή 2. Στις περιπτώσεις αυτές αυτό το διάνυσμα εμφανίζεται στο σύνολο δεδομένων κάθε φορά με την εκάστοτε έξοδό του, για παράδειγμα το (1,2) θα εμφανιστεί μία φορά με έξοδο το 1 και μία δεύτερη φορά με έξοδο το 3. Η ανωτέρω διαδικασία γίνεται για όλα τα ελεύθερα τετράγωνα του λαβυρίνθου μας. Η αρίθμηση στο σύνολο προτύπων μας, δηλαδή στο σύνολο δεδομένων και στο σύνολο αποτελεσμάτων αριθμείται με φορά αντίστροφα από τους δείκτες του ρολογιού, δηλαδή από τα δεξιά προς τα αριστερά και κάθε θέση του διανύσματος τιμών αντιστοιχεί σε ένα συγκεκριμένο γειτονικό τετράγωνο. Ειδικότερα το διάνυσμα εισόδου έχει την μορφή $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8)$ όπου το α_1 αντιστοιχεί στην θέση του επάνω γειτονικού τετραγώνου, το α_2 στην θέση του αριστερού διαγώνιου γειτονικού τετραγώνου, το α_3 στην θέση του αριστερού γειτονικού τετραγώνου το α_4 στην θέση του κάτω

διαγώνιου αριστερού γειτονικού τετραγώνου , το α5 στη θέση του κάτω γειτονικού γειτονικού τετραγώνου, το α6 στη θέση του κάτω διαγώνιου δεξιού γειτονικού τετραγώνου το α7 στη θέση του δεξιού γειτονικού τετραγώνου και το α8 στη θέση του επάνω διαγώνιου γειτονικού τετραγώνου όπως φαίνεται στο παρακάτω σχήμα 4.4.

α2	α1	α8
α3		α7
α4	α5	α6

Σχήμα 4.4

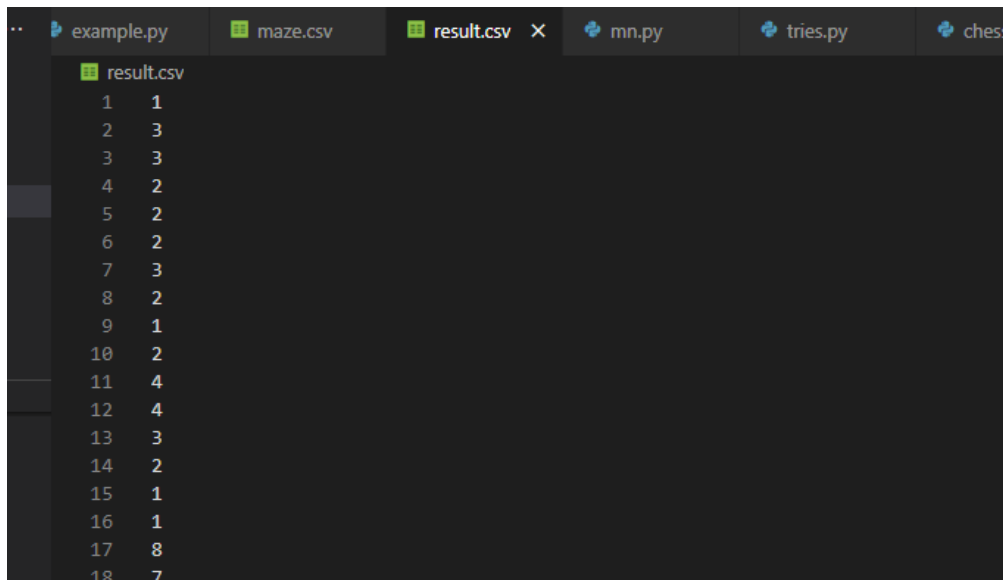
Παρατηρούμε ότι ο λαβύρινθος μας περιέχει 33 ελεύθερα τετράγωνα. Οπότε θα έχουμε τουλάχιστον 33 διανύσματα εισόδου με τα αντίστοιχα διανύσματα εξόδου. Με βάση όσα έχουμε αναφέρει παραπάνω δημιουργούμε στο visual code το csv αρχείο maze(Σχήμα 4.5) που περιέχει 35 διανύσματα του συνόλου δεδομένων και το csv αρχείο results(Σχήμα 4.6) που περιέχει τα αντίστοιχα αποτελέσματα.



```
example.py  maze.csv  X  mn.py  tries.py  chess_game.py  examples.py
maze.csv
1  1,1,1,0,0,0,0,0
2  1,1,1,1,1,0,0,0
3  1,0,1,1,1,0,0,0
4  0,1,0,1,1,0,0,0
5  1,1,1,0,0,0,1,1
6  1,1,1,1,1,1,1,1
7  0,0,1,1,1,1,1,1
8  1,1,0,0,0,0,1,1
9  1,0,1,0,1,1,1,1
10 0,1,0,1,1,1,1,0
11 0,0,1,1,0,1,1,1
12 0,0,0,1,1,0,1,0
13 1,0,1,0,0,0,0,1
14 0,1,1,0,0,0,1,1
15 1,0,0,0,0,0,1,0
16 1,0,0,0,1,1,0,0
17 1,0,0,0,1,0,0,1
18 0,0,0,0,1,0,1,0

OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS
PS C:\hello_ds>
```

Σχήμα 4.5 Το αρχείο maze.



```
example.py  maze.csv  result.csv  X  mn.py  tries.py  chess_
result.csv
1  1
2  3
3  3
4  2
5  2
6  2
7  3
8  2
9  1
10 2
11 4
12 4
13 3
14 2
15 1
16 1
17 8
18 7
```

Σχήμα 4.6 Το αρχείο results.

Στη συνέχεια δημιουργούμε το νευρωνικός μας δίκτυο το οποίο αποτελείται από το στρώμα εισόδου τρία κρυφά στρώματα νευρώνων και το στρώμα εξόδου. Το στρώμα εισόδου αποτελείται από οχτώ νευρώνες , όσα το μέγεθος του διανύσματος εισόδου τα τρία κρυφά στρώματα που περιέχουν τρεις νευρώνες και το στρώμα εξόδου που περιέχει έναν νευρώνα , όσο και το μέγεθος του διανύσματος εξόδου. Στο πρώτο στρώμα χρησιμοποιείται η συνάρτηση ενεργοποίησης relu(ανορθωτής) , στο δεύτερο στρώμα η συνάρτηση ενεργοποίησης sigmoid(σιγμοειδής) και στο τρίτο στρώμα η συνάρτηση ενεργοποίησης softmax(κανονικοποιημένη εκθετική συνάρτηση).

Δημιουργούμε το αρχείο traverse.py στο visual code και ο κώδικας του νευρωνικού δικτύου σε keras και tensorflow είναι ο παρακάτω:

```
import tensorflow as tf

from tensorflow import keras

import numpy as np

from numpy import loadtxt

from keras.models import Sequential

from keras.layers import Dense

c= loadtxt('maze.csv', delimiter=',')

d = loadtxt('result.csv', delimiter=',')

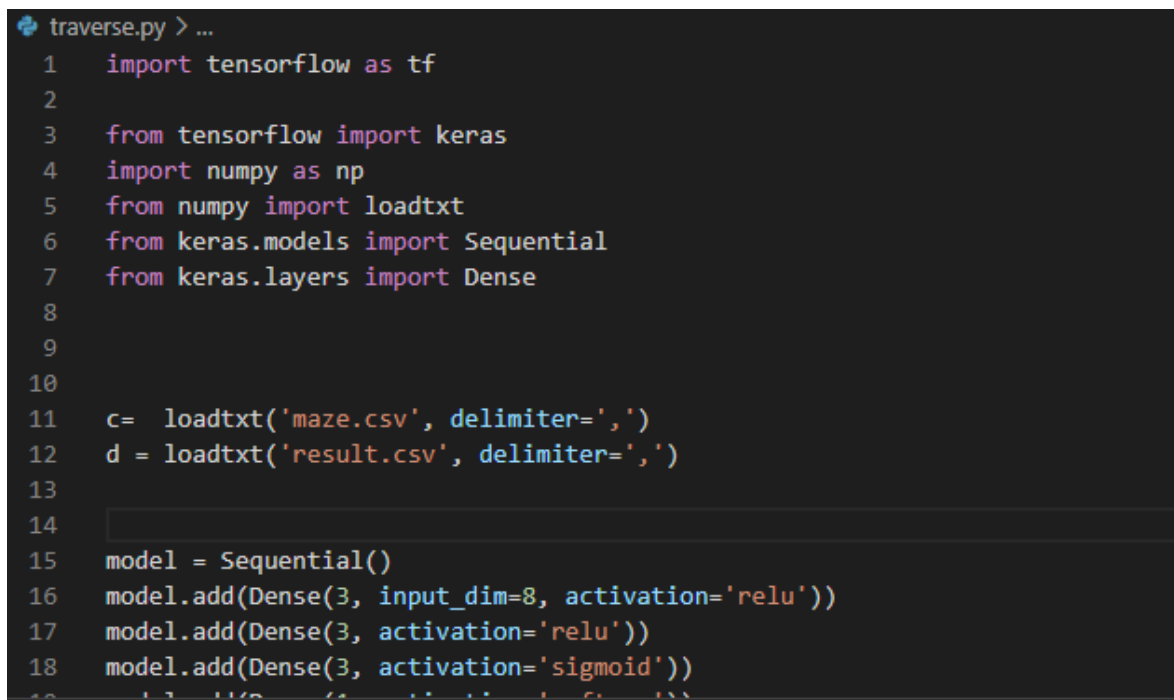
model = Sequential()

model.add(Dense(3, input_dim=8, activation='relu'))

model.add(Dense(3, activation='relu'))

model.add(Dense(3, activation='sigmoid'))
```

```
model.add(Dense(1, activation='softmax'))  
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])  
model.fit(c, d, epochs=150, batch_size=5)  
  
_, accuracy = model.evaluate(c, d)  
print('Accuracy: %.2f' % (accuracy*100))  
  
predictions = model.predict_classes(c)  
  
for i in range(33):  
    print('%s => %d (expected %d)' % (c[i].tolist(), predictions[i], d[i]))
```



```
traverse.py > ...  
1  import tensorflow as tf  
2  
3  from tensorflow import keras  
4  import numpy as np  
5  from numpy import loadtxt  
6  from keras.models import Sequential  
7  from keras.layers import Dense  
8  
9  
10  
11  c= loadtxt('maze.csv', delimiter=',')  
12  d = loadtxt('result.csv', delimiter=',')  
13  
14  
15  model = Sequential()  
16  model.add(Dense(3, input_dim=8, activation='relu'))  
17  model.add(Dense(3, activation='relu'))  
18  model.add(Dense(3, activation='sigmoid'))
```

Σε αυτόν τον κώδικα έχει γίνει χρήση της εντολής `loadtext` που μετατρέπει τα αρχεία στους πίνακες `c` και `d` αντίστοιχα και της εντολής `model.evaluate` που υπολογίζει τις σωστές προβλέψεις του νευρωνικού δικτύου, αφού το έχουμε εκπαιδεύσει. Γενικότερα για να ελέγξουμε την αποδοτικότητα ενός νευρωνικού δικτύου χρησιμοποιούμε ένα ορισμένο ποσοστό από το σύνολο προτύπων, συνήθως από 10% έως 25%, το οποίο δεν χρησιμοποιείται στην εκπαίδευση. Στην περίπτωσή μας, λόγω ότι έχουμε λίγες τιμές (35 πρότυπα)

δεν έχουμε την δυνατότητα για ένα τέτοιο ποσοστό οπότε χρησιμοποιούμε για εκπαίδευση και έλεγχο απόδοσης όλο το σύνολο των προτύπων. Ο έλεγχος της αποδοτικότητας πραγματοποιείται με την εντολή `accuracy = model.evaluate(c, d)`. Επίσης για να αποθηκεύσουμε τις τιμές των βαρών μας μετά την εκπαίδευση χρησιμοποιούμε την εντολή `model.save_weights('model.h5')`.

Στη συνέχεια εκπαιδεύουμε το δίκτυό μας.

```
14 |
15 | model = Sequential()
16 | model.add(Dense(3, input_dim=8, activation='relu'))
17 | model.add(Dense(3, activation='relu'))
18 | model.add(Dense(3, activation='sigmoid'))
19 | model.add(Dense(1, activation='softmax'))
20 | model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
21 | model.fit(c, d, epochs=150, batch_size=5)
22 |
23 | _, accuracy = model.evaluate(c, d)
24 | print('Accuracy: %.2f' % (accuracy*100))
25 |
26 |
27 | predictions = model.predict_classes(c)
28 |
29 | for i in range(33):
30 |     print('%s => %d (expected %d)' % (c[i].tolist(), predictions[i], d[i]))
```

```

OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS
Epoch 6/150
35/35 [=====] - 0s 685us/step - loss: -28.4760 - accuracy: 0.2286
Epoch 7/150
35/35 [=====] - 0s 1ms/step - loss: -28.4760 - accuracy: 0.2286
Epoch 8/150
35/35 [=====] - 0s 714us/step - loss: -28.4760 - accuracy: 0.2286
Epoch 9/150
35/35 [=====] - 0s 543us/step - loss: -28.4760 - accuracy: 0.2286
Epoch 10/150
35/35 [=====] - 0s 571us/step - loss: -28.4760 - accuracy: 0.2286
Epoch 11/150
35/35 [=====] - 0s 371us/step - loss: -28.4760 - accuracy: 0.2286
Epoch 12/150
35/35 [=====] - 0s 343us/step - loss: -28.4760 - accuracy: 0.2286
Epoch 13/150
35/35 [=====] - 0s 428us/step - loss: -28.4760 - accuracy: 0.2286
Epoch 14/150
35/35 [=====] - 0s 343us/step - loss: -28.4760 - accuracy: 0.2286
Epoch 15/150
35/35 [=====] - 0s 371us/step - loss: -28.4760 - accuracy: 0.2286
Epoch 16/150
35/35 [=====] - 0s 371us/step - loss: -28.4760 - accuracy: 0.2286
Epoch 17/150
35/35 [=====] - 0s 428us/step - loss: -28.4760 - accuracy: 0.2286

```

Αφού εκπαιδεύσουμε το δίκτυό μας , συντάσσουμε τον κώδικα που θα χρησιμοποιήσει τα βάρη , ώστε να πορέσουμε να κάνουμε προβλέψεις με το δίκτυό μας. Ο κώδικας είναι στο αρχείο cross.py είναι ο παρακάτω:

```

import numpy as np

from keras.models import Sequential

from keras.layers.core import Dense

model = Sequential()

model.add(Dense(3, input_dim=8, activation='relu'))

model.add(Dense(3, activation='relu'))

model.add(Dense(3, activation='sigmoid'))

model.add(Dense(1, activation='softmax'))

model.load_weights('model.h5')

```

```
a = input("What is x?")
b = input("What is x?")
c = input("What is x?")
d = input("What is x?")
e = input("What is x?")
f = input("What is x?")
g = input("What is x?")
h = input("What is x?")
#print("Nice to meet you " + a + "!")
t_data= np.array([[a,b,c,d,e,f,g,h]], "float32")

def my_function(a):
    w=model.predict(a).round()
    return w

s = my_function(t_data)
print(s)
```

```
cross.py > ...
1 import numpy as np
2 from keras.models import Sequential
3 from keras.layers.core import Dense
4
5 model = Sequential()
6 model.add(Dense(3, input_dim=8, activation='relu'))
7 model.add(Dense(3, activation='relu'))
8 model.add(Dense(3, activation='sigmoid'))
9 model.add(Dense(1, activation='softmax'))
10 model.load_weights('model.h5')
11
12 a = input("What is x?")
13 b = input("What is x?")
14 c = input("What is x?")
15 d = input("What is x?")
16 e = input("What is x?")
17 f = input("What is x?")
18 g = input("What is x?")
19 h = input("What is x?")
20 #print("Nice to meet you " + a + "!")
21
22
23
24 t_data= np.array([[a,b,c,d,e,f,g,h]], "float32")
25
26
27 def my_function(a):
28     w=model.predict(a).round()
29     return w
30
31 s = my_function(t_data)
32 print(s)
```

Όπως έχουμε αναφέρει και παραπάνω τα νευρωνικά δίκτυα ενδέχεται να δώσουν και λάθος προβλέψεις. Εμείς για να αντιμετωπίσουμε αυτό το πρόβλημα χρησιμοποιούμε την ευκλείδεια απόσταση. Ειδικότερα αφού ο πράκτορας κάνει χρήση του νευρωνικού δικτύου και αντιληφθεί ότι η πρόβλεψη είναι λάθος(αυτό το αντιλαμβάνεται , γιατί έχει θέαση των οκτώ γειτονικών τετραγώνων οπότε γνωρίζει ποιά είναι ελεύθερα και ποιά περιέχουν

εμπόδια) τότε κάνει χρήση της ευκλείδειας απόστασης μεταξύ των ελευθέρων τετραγώνων $\alpha_1, \alpha_2, \alpha_3, \alpha_7, \alpha_8$, και το τετραγώνου (7,7) χρησιμοποιώντας τις συντεταγμένες. Ο πράκτορας μεταβαίνει στο τετράγωνο που απέχει τη μικρότερη απόσταση. Ο κώδικας για την ευκλείδεια απόσταση δημιουργείται στο αρχείο `euclidean.py` και είναι ο παρακάτω:

```
import numpy as np
import math

valA=int(input("What is x coordinate?"))
valB =int(input("What is y coordinate?"))

a=((7-valA)**2)
b= ((7-valB)**2)
p1 =[a,b]
p2=[7,7]

distance = math.sqrt(a+b)
print(distance)
```

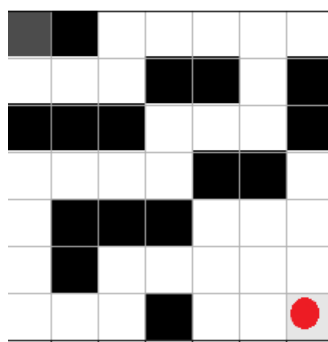


```

euclidean.py X example.py mn.py tri
euclidean.py > ...
1  import numpy as np
2  import math
3
4  valA = int(input("What is x coordinate?"))
5  valB =int( input("What is y coordinate?"))
6
7  a=((7-valA)**2)
8  b= ((7-valB)**2)
9  p1 = [a,b]
10 p2=[7,7]
11
12
13 distance = math.sqrt(a+b)
14 print(distance)
15
16
17

```

Θεωρούμε ότι ο πράκτορας έχει μαζί του έναν υπολογιστή όπου μπορεί και τρέχει το νευρωνικό δίκτυο που έχουμε δημιουργήσει και τα ανωτέρω προγράμματα. Όπως έχουμε αναφέρει παραπάνω ο πράκτορας γνωρίζει την αρχή του λαβυρίνθου και γνωρίζει ότι το τέλος του λαβυρίνθου είναι το (7,7). Επίσης γνωρίζει τα οκτώ γειτονικά τετράγωνα σε κάθε θέση που βρίσκεται και κάνει χρήση του νευρωνικού δικτύου για να διασχίσει τον λαβύρινθο και να φτάσει στο (7,7).



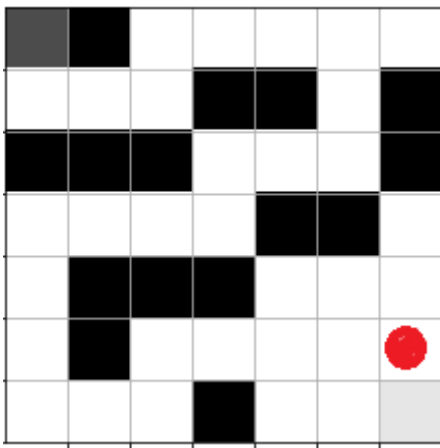
Ο πράκτορας βρίσκεται στην θέση (1,1). Εισάγει τις τιμές των οκτώ γειτονικών τετραγώνων δηλαδή το (1,1,1,0,0,0,0,0) στο κώδικα του αρχείου cross παίρνει την

τιμή 1 που δηλώνει ότι πρέπει να κινηθεί ευθεία. Δεν υπάρχει εμπόδιο οπότε ο πράκτορας μεταβαίνει στο (2,1).

```

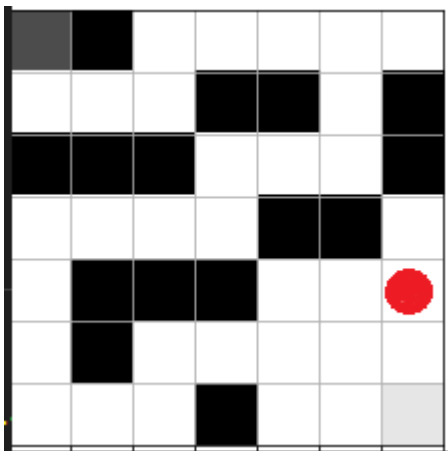
2021-01-11 22:32:39.948196: I tensorflow/core/platform/c
iled to use: AVX AVX2
What is the first square?1
What is the second square?1
What is the third square?1
What is the fourth square?0
What is the fifth square?0
What is the six square?0
What is the seventh square?0
What is the eighth square?0
[[1.]]
PS C:\hello ds>

```



Στη συνέχεια εισάγει στο κώδικα του αρχείου cross το (1,1,1,1,1,0,0,0) και παίρνει την τιμή 1 οπότε μεταβαίνει στο (3,1), αφού είναι ελεύθερο.

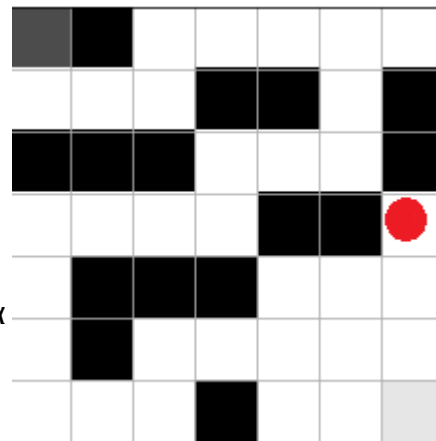
```
2021-01-11 22:29:18.749965: I tensorflow/core/platform/cpu_tf
What is the first square?1
What is the second square?1
What is the third square?1
What is the fourth square?1
What is the fifth square?1
What is the six square?0
What is the seventh square?0
What is the eighth square?0
[[1.]]
PS C:\hello ds> & C:/Users/fotios/.conda/envs/tensorflow/pyth
```



Στη συνέχεια εισάγει την τιμή (1,0,1,1,1,0,0,0) στον κώδικα του αρχείου cross και παίρνει την τιμή 1 και το επάνω τετράγωνο το (4,1) είναι ελεύθερο οπότε μεταβαίνει σε αυτό.

```

2021-01-11 22:34:33.278154: I tensorflow/core/
iled to use: AVX AVX2
What is the first square?1
What is the second square?0
What is the third square?1
What is the fourth square?1
What is the fifth square?1
What is the six square?0
What is the seventh square?0
What is the eighth square?0
[[1.]]
    
```



Μετά
θέτει
την

τιμή (0,1,0,1,1,0,0,0) στον κώδικα του αρχείου cross και παίρνει την τιμή 1.

```

using tensorflow backend.
2021-01-11 22:36:15.164553: I tensorflow/core/
iled to use: AVX AVX2
What is the first square?0
What is the second square?1
What is the third square?0
What is the fourth square?1
What is the fifth square?1
What is the six square?0
What is the seventh square?0
What is the eighth square?0
[[1.]]
    
```

Ο πράκτορας παρατηρεί , αφού έχει θέαση σε στα οκτώ γειτονικά τετράγωνα ότι η τιμή 1 που υποδικνύει το τετράγωνο (5,1) είναι λάθος , γιατί το τετράγωνο αυτό περιέχει εμπόδιο. Τότε θα κάνει χρήση του κώδικα της pythop που υπάρχει στο αρχείο euclidean.py και θα υπολογίσει την ευκλείδεια απόσταση για τα ελεύθερα γειτονικά τετράγωνα τα οποία είναι το (3,1),(3,2) και (5,2).

Για το (3,1):

```

PS C:\hello_ds> & C:/Users/fotios/.conda
What is x coordinate?3
What is y coordinate?1
7.211102550927978
PS C:\hello_ds>
    
```

Η απόσταση είναι περίπου 7,211.

Για το (3,2):

```
PS C:\hello_ds> & C:/Users/fotios/.cond
What is x coordinate?3
What is y coordinate?2
6.4031242374328485
PS C:\hello_ds> |
```

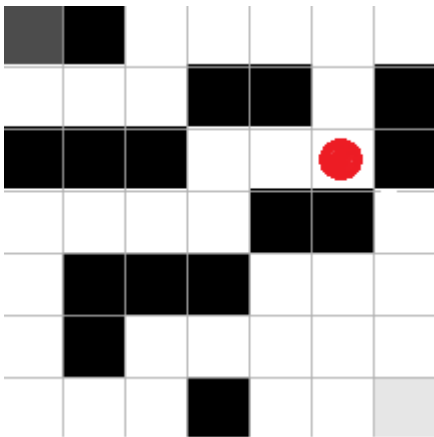
Η απόσταση είναι περίπου 6,4.

Για το (5,2):

```
PS C:\hello_ds> & C:/Users/fotios/.cond
What is x coordinate?5
What is y coordinate?2
5.385164807134504
PS C:\hello ds> |
```

Η απόσταση είναι περίπου 5,38.

Παρατηρεί ότι η μικρότερη απόσταση είναι από το τετράγωνο (5,2) οπότε μεταβαίνει σε αυτό.

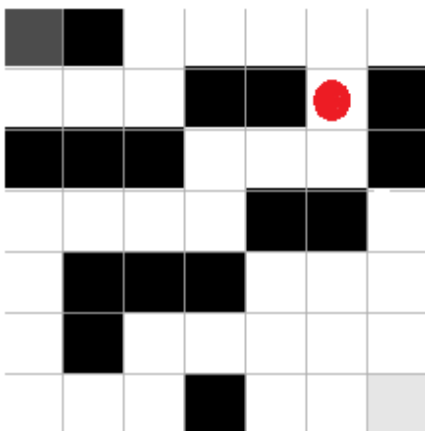


Στη συνέχεια θέτει στο κώδικα cross.py τα οκτώ γειτονικά τετράγωνα το (5,2) που είναι το διάνυσμα (1,0,1,0,0,1,0,0) και παίρνει την τιμή 1.

```
PS C:\hello_ds> & C:/Users/fotios/.conda/envs/t
Using TensorFlow backend.
2020-12-04 15:27:06.348664: I tensorflow/core/p
ed to use: AVX AVX2
What is the first square?1
What is the second square?0
What is the third square?1
What is the fourth square?0
What is the fifth square?0
What is the six square?1
What is the seventh square?0
What is the eighth square?0
[[1.]]
PS C:\hello ds>
```

Παρατηρεί ότι η τιμή αυτή που του δίνει το νευρωνικό δίκτυο οδηγεί στο

τετράγωνο (6,2) , οπότε μεταβαίνει σε αυτό.

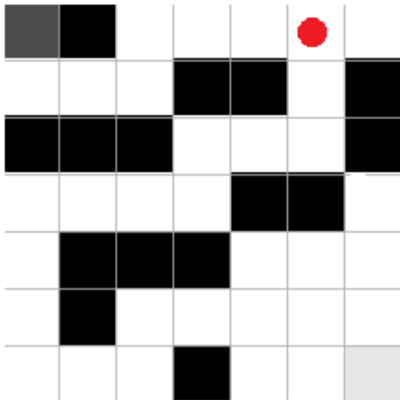


Μετά από αυτήν την κίνηση χρησιμοποιεί ξανά τον κώδικα για το (6,2) και τα γειτονικά του τετράγωνα , δηλαδή δίνει ως είσοδο το διάνυσμα

(1,1,0,1,1,0,0,1) και παίρνει την τιμή 1.

```
PS C:\hello_ds> & C:/Users/fotios/.conda/envs/tensorf
Using TensorFlow backend.
2020-12-04 15:35:30.939713: I tensorflow/core/platfor
ed to use: AVX AVX2
What is the first square?1
What is the second square?1
What is the third square?0
What is the fourth square?1
What is the fifth square?1
What is the six square?0
What is the seventh square?0
What is the eighth square?1
[[1.]]
```

Ομοίως με το προηγούμενο τετράγωνο ο πράκτορας μας παρατηρεί ότι το νευρωνικό δίκτυο του δείχνει ότι την θέση (7,2) που είναι ελεύθερη οπότε μεταβαίνει σε αυτή.



Ομοίως θέτει το διάνυσμα που προκύπτει από το (7,2) στο κώδικα του cross.py που είναι το (0,0,1,0,1,0,1,0) και παίρνει την τιμή 1.

```
Using TensorFlow backend.
2020-12-04 15:47:05.499314: I tensorflow/core/pl
ed to use: AVX AVX2
What is the first square?0
What is the second square?0
What is the third square?1
What is the fourth square?0
What is the fifth square?1
What is the six square?0
What is the seventh square?1
What is the eighth square?0
[[1.]]
```

Η τιμή αυτή υποδεικνύει μετάβαση εκτός του λαβυρίνθου , άρα είναι λάθος εκτίμηση. Ο πράκτορας θα κάνει χρήση του αρχείου euclidean.py για τα ελεύθερα γειτονικά τετράγωνα που είναι το (6,2) το (7,1) και το (7,3)

Για το (7,1):

```
PS C:\hello_ds> & C:/Users/fotios/
What is x coordinate?7
What is y coordinate?1
6.0
PS C:\hello_ds>
```

Η απόσταση είναι 6.

Για το (6,2):

```
PS C:\hello_ds> conda activate tensorflow
PS C:\hello_ds> & C:/Users/fotios/.conda/envs/ten
What is x coordinate?6
What is y coordinate?2
5.0990195135927845
PS C:\hello_ds>
```

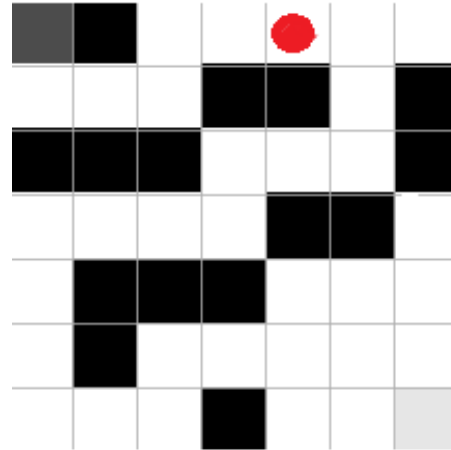
Η απόσταση είναι περίπου 5,09.

Για το (7,3):

```
PS C:\hello_ds> & C:/Users/fotios/.conda/envs/te
What is x coordinate?7
What is y coordinate?3
4.0
PS C:\hello_ds>
```

Η απόσταση είναι 4.

Η μικρότερη απόσταση είναι του (7,3) και είναι 4 , οπότε μεταβαίνει σε αυτό.



Στη συνέχεια κάνει χρήση του κώδικα στο αρχείο cross.py για το διάνυσμα της θέσης (7,3) που είναι το (0,0,1,0,0,1,0,0) και παίρνει την τιμή 1.

```
PS C:\hello_ds> & C:/Users/fotios/.conda/en
Using TensorFlow backend.
2020-12-05 23:24:03.684347: I tensorflow/co
ed to use: AVX AVX2
What is the first square?0
What is the second square?0
What is the third square?1
What is the fourth square?0
What is the fifth square?0
What is the sixth square?1
What is the seventh square?0
What is the eighth square?0
[[1.]]
PS C:\hello_ds>
```

Ο πράκτορας βλέπει ότι η τιμή αυτή τον οδηγεί εκτός λαβυρίνθου άρα είναι λάθος, Οπότε όπως και στην προηγούμενη περίπτωση θα χρησιμοποιήσει την ευκλείδεια απόσταση με τον κώδικα euclidean.py για τα ελεύθερα γειτονικά σημεία , δηλαδή για τα (7,2) και (7,4).

Για το (7,2) (αν και έχει υπολογιστεί από πριν) :


```
PS C:\hello_ds> & C:/Users/fotios/.conda/
What is x coordinate?7
What is y coordinate?2
5.0
```

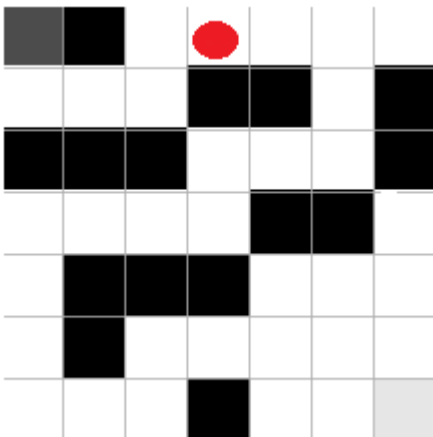
Η απόσταση είναι 5.

Για το (7,4):

```
PS C:\hello_ds> & C:/Users/fotios/.conda/
What is x coordinate?7
What is y coordinate?4
3.0
```

Η απόσταση είναι 3.

Η μικρότερη απόσταση είναι του (7,4) οπότε μεταβαίνει σε αυτό το σημείο.



Στη συνέχεια κάνει χρήση του κώδικα στο αρχείο cross.py για το διάνυσμα της θέσης (7,4) που είναι το (0,0,1,1,0,0,1,0) και παίρνει την τιμή 1.

```
2020-12-05 23:43:15.467109: I tensorflow/core/platf
ed to use: AVX AVX2
What is the first square?0
What is the second square?0
What is the third square?1
What is the fourth square?1
What is the fifth square?0
What is the six square?0
What is the seventh square?1
What is the eighth square?0
[[1.]]
```

Ο πράκτορας βλέπει ότι η τιμή αυτή τον οδηγεί εκτός λαβυρίνθου άρα είναι λάθος, Οπότε όπως και στην προηγούμενη κίνηση χρησιμοποιεί την ευκλείδεια απόσταση με τον κώδικα euclidean.py για τα ελεύθερα γειτονικά τετράγωνα που είναι τα (7,3) , (7,5) και το (6,2).

Για το (7,3):

```
PS C:\hello_ds> & C:/Users/fotios/.
What is x coordinate?7
What is y coordinate?3
4.0
```

Η απόσταση είναι 4.

Για το (7,5):

```
PS C:\hello_ds> & C:/Users/fotios/.
What is x coordinate?7
What is y coordinate?5
2.0
```

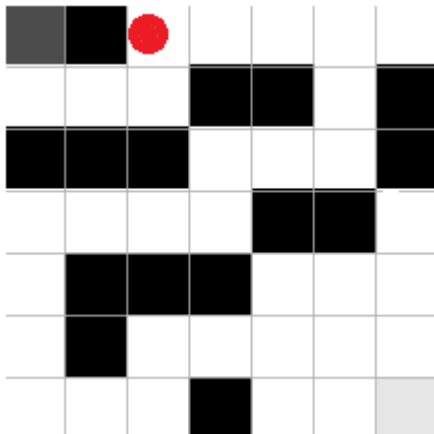
Η απόσταση είναι 2.

Για το (6,2):

```
PS C:\hello_ds> & C:/Users/fotios/.
What is x coordinate?6
What is y coordinate?2
5.0990195135927845
```

Η απόσταση είναι περίπου 5,09.

Εγώ η απόσταση που είναι μικρότερη από όλες είναι η 2 που αντιστοιχεί στο (7,5), άρα ο πράκτορας μεταβαίνει εκεί.



Μετά κάνει χρήση του νευρωνικού δικτύου το διάνυσμα της θέσης (7,5) που είναι το (0,0,0,1,1,0,0,1) και παίρνει την τιμή 1.

```
PS C:\hello_ds> & C:/Users/fotios/.
Using TensorFlow backend.
2020-12-06 00:01:16.547498: I tensorflow
ed to use: AVX AVX2
What is the first square?0
What is the second square?0
What is the third square?0
What is the fourth square?1
What is the fifth square?1
What is the sixth square?0
What is the seventh square?0
What is the eighth square?1
[[1.]]
```

Όπως και σε προηγούμενες περιπτώσεις πράκτορας βλέπει ότι η τιμή αυτή τον οδηγεί εκτός λαβυρίνθου άρα είναι λάθος, Οπότε ευκλείδεια απόσταση με τον κώδικα euclidean.py για τα ελεύθερα γειτονικά τετράγωνα που είναι τα (7,4) , (6,5) και το (6,6).

Για το (7,4):

```
PS C:\hello_ds> & C:/Users/fotios/
What is x coordinate?7
What is y coordinate?4
3.0
```

Η απόσταση είναι 3.

Για το (6,5):

```
PS C:\hello_ds> & C:/Users/
What is x coordinate?6
What is y coordinate?5
2.23606797749979
```

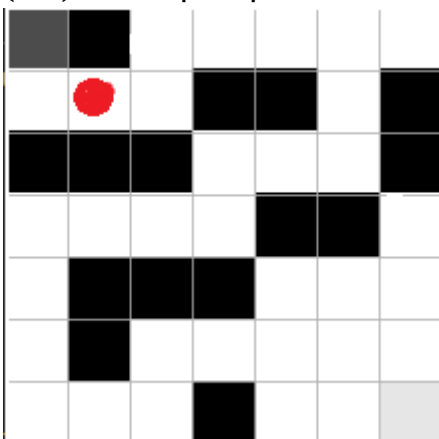
Η απόσταση είναι 2,23 περίπου.

Για το (6,6):

```
PS C:\hello_ds> & C:/Users/
What is x coordinate?6
What is y coordinate?6
1.4142135623730951
```

Η απόσταση είναι 1,414 περίπου.

Παρατηρεί ότι η μικρότερη απόσταση που είναι η 1,424 αντιστοιχεί στο σημείο (6,6) οπότε μεταβαίνει σε αυτό.



Μετά κάνει χρήση του νευρωνικού δικτύου για το (6,6) , δηλαδή χρησιμοποιεί το διάνυσμα (0,1,1,0,0,0,1,1) και παίρνει την τιμή 1.

```
Using TensorFlow backend.
2020-12-06 00:15:54.922437: I tensorflow/core/common_runtime/gpu/gpu_device.cc:170:
ed to use: AVX AVX2
What is the first square?0
What is the second square?1
What is the third square?1
What is the fourth square?0
What is the fifth square?0
What is the six square?0
What is the seventh square?1
What is the eighth square?1
[[1.]]
```

Το αποτέλεσμα αυτό τον οδηγεί στο (7,6) που περιέχει εμπόδιο άρα θα κάνει χρήση της ευκλείδειας απόστασης για τα ελεύθερα γειτονικά σημεία που είναι τα (7,5) , (6,5),(6,7) και (7,7).

Για το (7,5):

```
[[4.]]
PS C:\hello_ds> & C:/Users/fo
What is x coordinate?7
What is y coordinate?5
2.0
```

Η απόσταση είναι 2.

Για το (6,5):

```
PS C:\hello_ds> & C:/Use
What is x coordinate?6
What is y coordinate?5
2.23606797749979
```

Η απόσταση είναι 2,23 περίπου.

Για το (6,7):

```
PS C:\hello_ds> & C:/Users
What is x coordinate?6
What is y coordinate?7
1.0
```

Η απόσταση είναι 1.

Για το (7,7):

```
PS C:\hello_ds> & C:/Users/fo
What is x coordinate?7
What is y coordinate?7
0.0
```

Η απόσταση είναι 0.

Παρατηρεί ότι η μικρότερη απόσταση είναι το 0 που αντιστοιχεί στο τετράγωνο (7,7). Άρα ο πράκτορας έχει φτάσει στη έξοδο.

Η ακρίβεια είναι μια μέτρηση για την αξιολόγηση των μοντέλων ταξινόμησης. Άτυπα, η ακρίβεια είναι το κλάσμα των προβλέψεων που το μοντέλο μας πήρε σωστά. Επίσημα, η ακρίβεια έχει τον ακόλουθο ορισμό: Ακρίβεια = Αριθμός σωστών προβλέψεων / Συνολικός αριθμός προβλέψεων

Παρατηρούμε ότι σύμφωνα με την σελίδα 73 το νευρωνικό δίκτυο που έχουμε δημιουργήσει έχει ακρίβεια(accuracy) 0.22 ή 22%. Στην συγκεκριμένη περίπτωση έχουμε μικρό ποσοστό ακρίβειας γιατί έχουμε μικρό αριθμό προτύπων για την εκπαίδευση του δικτύου μας. Όταν εκπαιδεύσουμε το δίκτυό μας με μικρό αριθμό προτύπων έχουμε μικρό ποσοστό ακρίβειας. Για να επιτύχουμε μεγάλο ποσοστό ακρίβειας δηλαδή από το 70% και επάνω σύμφωνα με πολύ γενικές εκτιμήσεις πρέπει να έχουμε τουλάχιστον 700 πρότυπα για την εκπαίδευση.

4.2 Διάσχιση λαβυρίνθου με την χρήση της απόστασης manhattan.

Μία άλλη υλοποίηση για την εύρεση της εξόδου του λαβυρίνθου είναι με την χρήση της απόστασης Μανχάταν(Manhattan distance). Η απόσταση Μανχάταν ορίζεται ως εξής:

Αν $A(x_a, y_a)$ και $B(x_b, y_b)$ είναι οι συντεταγμένες δύο σημείων στο καρτεσιανό επίπεδο τότε η απόσταση Μανχάταν είναι ίση με $AB = |x_a - x_b| + |y_a - y_b|$ (α).

Στην δική μας περίπτωση ο πράκτορας κάνει χρήση της απόστασης Μανχάταν θέτοντας ως συντεταγμένες τις συντεταγμένες των τετραγώνων του λαβυρίνθου που αναφέραμε στην προηγούμενη υλοποίηση. Ειδικότερα υπολογίζεται η απόσταση Μανχάταν μεταξύ του τετραγώνου (7,7) και των οκτώ γειτονικών τετραγώνων του.

Όποιο τετράγωνο δώσει την μικρότερη απόσταση , σημαίνει ότι απέχει λιγότερο από την έξοδο άρα ο πράκτορας μεταβαίνει σε αυτό. Αφού το τετράγωνο εξόδου είναι το (7,7) η εξίσωση (α) μπορεί να γραφεί ως εξής:

$$AB = |7 - x_b| + |7 - y_b| \quad (\beta)$$

Οι προϋποθέσεις είναι ίδιες με αυτές της προηγούμενης υλοποίησης , δηλαδή η αρχή είναι το (1,1) το τέλος είναι το (7,7) και ο πράκτορας έχει θέαση τα οκτώ γειτονικά τετράγωνα. Επίσης γνωρίζει ότι η αρχή είναι το (1,1) και το τέλος το (7,7). Θα δημιουργήσουμε ένα τεχνητό νευρωνικό δίκτυο που να υπολογίζει την απόσταση Μανχάταν μεταξύ των τετραγώνων. Την απόσταση (β) δηλαδή θα μας την δίνει το νευρωνικό δίκτυο.

Το νευρωνικό δίκτυο θα έχει σαν είσοδο ένα διάνυσμα δύο τιμών που είναι οι συντεταγμένες κάποιου τετραγώνου , και θα μας δίνει ως έξοδο ένα διάνυσμα μία τιμής που είναι η απόσταση μεταξύ αυτού του τετραγώνου και του τετραγώνου (7,7).

Αρχικά πρέπει να δημιουργήσουμε το σύνολο δεδομένων και το σύνολο εξόδου. Το σύνολο δεδομένων είναι ένα διάνυσμα δύο τιμών όπου και οι δύο τιμές λαμβάνονται από το διάστημα 1 έως 7, όσες και οι συντεταγμένες του λαβυρίνθου μας. Οπότε για την συντεταγμένη x δημιουργούμε μία λίστα που περιέχει εννιακόσιες ενενήντα τρεις τιμές πραγματικών αριθμών και μία δεύτερη λίστα που περιέχει επτά τιμές πραγματικών αριθμών. Στη συνέχεια συνενώνουμε τις λίστες και δημιουργούμε μία λίστα χιλίων πραγματικών αριθμών την οποία και ταξινομούμε κατά αύξουσα σειρά. Ακολουθούμε όμοια διαδικασία για την συντεταγμένη y. Δημιουργούμε και τα αρχεία x_desc και y_desc όπου περιέχουν τις τιμές των x και y κατά φθίνουσα σειρά. Συνενώνοντας τα x_desc και y_desc και τα x και y κατά σειρά δημιουργούμε ένα σύνολο δεδομένων τριών χιλιάδων τιμών.

Ο κώδικας είναι ο παρακάτω:

```

from random import randint
import numpy as np
import random
import tensorflow as tf
from tensorflow import keras
from numpy import loadtxt
from keras.models import Sequential
from keras.layers import Dense
import csv

a = np.random.uniform(1, 7, 993) // Δημιουργία των τυχαίων 993 τιμών από το 1 έως το 7.
b = [1,2,3,4,5,6,7] // Δημιουργία των τιμών 1,2,3,4,5,6,7

x = np.concatenate([a,b]) // Συννένωση των δύο λιστών
y = np.concatenate([a,b])
y.sort() // Ταξινόμηση των δύο λιστών κατά αύξουσα σειρά.
x.sort()
x_desc = -np.sort(-x) // Ταξινόμηση των δύο λιστών κατά φθίνουσα σειρά
y_desc = -np.sort(-y)
np.savetxt("x.csv",x, delimiter = ",") // Αποθήκευση των αρχείων που περιέχουν
np.savetxt("y.csv",y, delimiter = ",") // τις ταξινομημένες λίστες.
np.savetxt("x_desc.csv",x_desc, delimiter = ",")
np.savetxt("y_desc.csv",x_desc, delimiter = ",")

x_f = np.concatenate([x,x,x_desc]) // Διάταξη των λιστών που δημιουργήσαμε

```

```

y_f = np.concatenate([y,y_desc]) // παραπάνω για την δημιουργία του συνόλου
                                   // δεδομένων των 3000 τιμών.

np.savetxt("x_f.csv",x_f, delimiter = ",")
np.savetxt("y_f.csv",y_f, delimiter = ",")

i=0

list = []

for entry in x_f: // Τοποθέτηση των τιμών των λιστών σε μορφή x,y και
    list.append([x_f[i],y_f[i]]) // αποθήκευση τους στο αρχείο values.csv.
    i = i+1

my_array = np.asarray(list)
print(my_array)
np.savetxt("values.csv", my_array, delimiter = ",")

```

```
X = x_f*(-1)+7
```

```
Y = y_f*(-1)+7 // Πολλαπλασιάζουμε όλα τα x και όλα τα y με το -1, προσθέτουμε
```

```
res = X+Y // το 7 και μετά προσθέτουμε αυτά τα δύο αθροίσματα για την
```

```
np.savetxt("outs.csv",res, delimiter = ",") // δημιουργία του συνόλου αποτελεσμάτων.
```

Για την δημιουργία του συνόλου τιμών πολλαπλασιάζουμε όλα τα x και όλα τα y με το -1, προσθέτουμε το 7 και μετά προσθέτουμε αυτά τα δύο αθροίσματα για την δημιουργία του συνόλου αποτελεσμάτων. Το σύνολο αυτό το αποθηκεύουμε στο αρχείο με το όνομα outs.csv.

Μετά από την δημιουργία των προτύπων δημιουργούμε το νευρωνικό μας δίκτυο το οποίο αποτελείται από τέσσερα στρώματα , το πρώτο περιέχει δύο νευρώνες , τα δύο κρυφά περιέχουν οκτώ νευρώνες και το στρώμα εξόδου περιέχει έναν νευρώνα. Τα δύο εσωτερικά στρώματα περιέχουν για συνάρτηση ενεργοποίησης την relu(τον ανορθωτή).

Ο κώδικας είναι ο παρακάτω και αποθηκεύεται στο αρχείο manhattan.py.


```
import numpy as np

import keras

from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam
from keras import regularizers
import matplotlib.pyplot as plt
import tensorflow as tf

from tensorflow import keras
import numpy as np
from numpy import loadtxt
from keras.models import Sequential
from keras.layers import Dense

model = Sequential()
model.add(Dense(8, activation='relu', kernel_regularizer=regularizers.l2(0.001),
input_shape = (2,)))
model.add(Dense(8, activation='relu', kernel_regularizer=regularizers.l2(0.001)))
model.add(Dense(1))

model.compile(optimizer=Adam(),loss='mse')

# generate 10,000 random numbers in [-50, 50], along with their squares
```

```
x= loadtxt('values.csv', delimiter=',')
y= loadtxt('outs.csv', delimiter=',')

# fit the model, keeping 2,000 samples as validation set
hist = model.fit(x,y,validation_split=0.2,
                 epochs= 500,
                 batch_size=1000)
accuracy = model.evaluate(x,y)
print('Accuracy: %.2f' % (accuracy*100))
# check some predictions:
#print(model.predict([4,5]))
model.save_weights('model.h7')
```

Ύστερα δημιουργούμε των κώδικα της python που θα εισάγουμε τιμές και θα τρέχουμε το εκπαιδευμένο νευρωνικό μας δίκτυο. Ο κώδικας αποθηκεύεται στο αρχείο `model.csv` και είναι ο παρακάτω.

```
import numpy as np
import keras
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam
from keras import regularizers
import matplotlib.pyplot as plt
```

```
import tensorflow as tf

from tensorflow import keras

import numpy as np

from numpy import loadtxt

from keras.models import Sequential

from keras.layers import Dense

model = Sequential()

model.add(Dense(8, activation='relu', kernel_regularizer=regularizers.l2(0.001),
input_shape = (2,)))

model.add(Dense(8, activation='relu', kernel_regularizer=regularizers.l2(0.001)))

model.add(Dense(1))

model.compile(optimizer=Adam(),loss='mse')

model.load_weights('model.h7')

a = input("What is the first square?")

b = input("What is the second square?")

t_data= np.array([[a,b]], "float32")

def my_function(a):

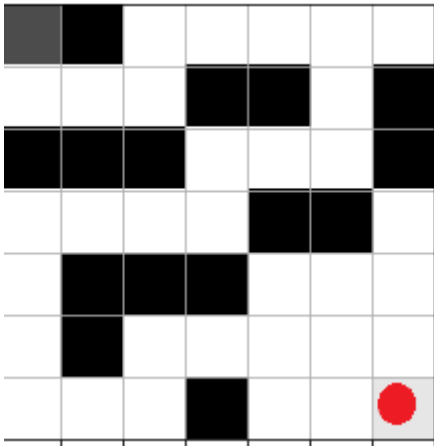
    w=model.predict(a)

    return w

s = my_function(t_data)

print(s)
```

Ο πράκτορας που είναι ο κόκκινος κύκλος βρίσκεται αρχικά στη θέση (1,1) και χρησιμοποιεί την απόσταση manhattan για τα τρία γειτονικά τετράγωνα , δηλαδή για τα (1,2) ,(2,2) και (2,1).



Για το (1,2):

```
2020-12-14 00:47:40.497557: I
iled to use: AVX AVX2
What is the first square?1
What is the second square?2
[[11.499397]]
```

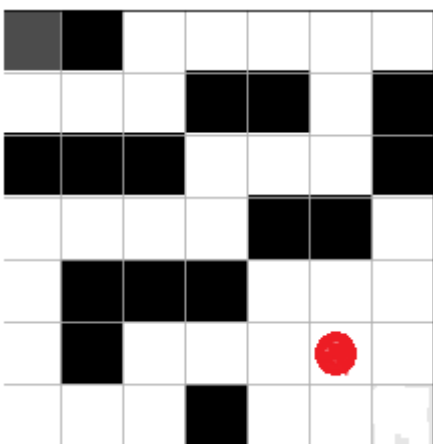
Για το (2,2):

```
iled to use: AVX AVX2
What is the first square?2
What is the second square?2
[[10.02544]]
PS C:\hello> cd . & C:\Users\fotios/.
Using TensorFlow backend.
```

Για το (2,1):

```
What is the first square?2
What is the second square?1
[[10.559155]]
PS C:\hello_ds> & C:/Users/fotios/.
Using TensorFlow backend.
```

Παρατηρεί ότι η μικρότερη απόσταση manhattan είναι του τετραγώνου (2,2) οπότε μεταβαίνει σε αυτό.



Ομοίως υπολογίζουμε τις αποστάσεις για τα ελεύθερα γειτονικά τετράγωνα:

Για το (2,3):

ευρωπαϊκών δικτύων.

Για το (3,3) :

```
What is the first square?2
What is the second square?3
[[9.491725]]
```

```
What is the first square?3
What is the second square?3
[[8.017768]]
```

Για το (3,2):

```
What is the first square?3
What is the second square?2
[[8.551483]]
```

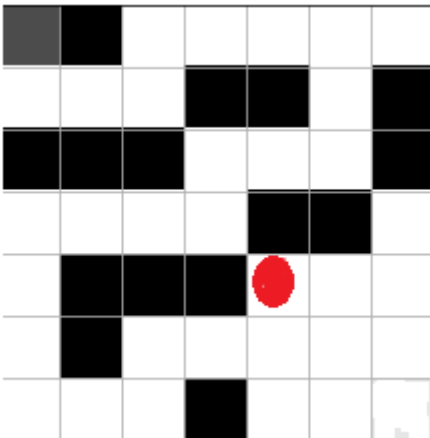
Για το (2,1):

```
What is the first square?2
What is the second square?1
[[10.559155]]
```

Για το (3,1):

```
What is the first square?3
What is the second square?1
[[9.085197]]
```

Παρατηρούμε ότι η μικρότερη απόσταση manhattan είναι του τετραγώνου (3,3) οπότε ο πράκτορας μεταβαίνει σε αυτό.



Για τετράγωνα (2,1) και (1,2) έχει υπολογιστεί η απόσταση στο προηγούμενο βήμα , οπότε δεν χρειάζεται να υπολογιστεί ξανά.

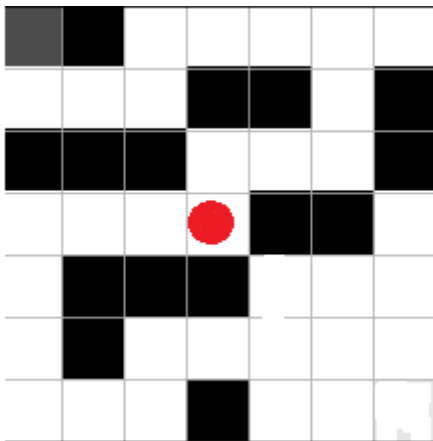
Συνεχίζει με τα ελεύθερα τετράγωνα (3,2) , (2,2) ,(2,3) και (4,4). Τα τρία πρώτα έχουν υπολογιστεί στο προηγούμενο βήμα οπότε υπολογίζεται το (4,4).

Για το (4,4):

```
What is the first square?4
What is the second square?4
[[6.0100956]]
```

Παρατηρεί ότι η μικρότερη απόσταση είναι του τετραγώνου (4,4) , που είναι ίση με 6.0100956 , οπότε

μεταβαίνει σε αυτό.



Μετά υπολογίζει τις αποστάσεις για τα ελεύθερα τετράγωνα (3,3) ,(4,5) ,(5,3) και (5,4).

Για το (3,3) η απόσταση υπολογίστηκε στο προηγούμενο βήμα οπότε για τα τα άλλα τετράγωνα έχουμε:

Για το (4,5):

```
What is the first square?4
What is the second square?5
[[5.476382]]
```

Για το (5,3):

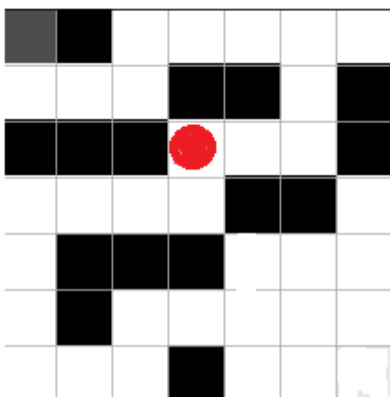
```
What is the first square?5
What is the second square?3
[[5.069851]]
```

Για το (5,4):

```
What is the first square?5
What is the second square?4
[[4.536137]]
```

Η μικρότερη απόσταση manhattan είναι του τετραγώνου (5,4) , που είναι ίση με 4.536137.

Οπότε ο πράκτορας μας μεταβαίνει σε αυτό.

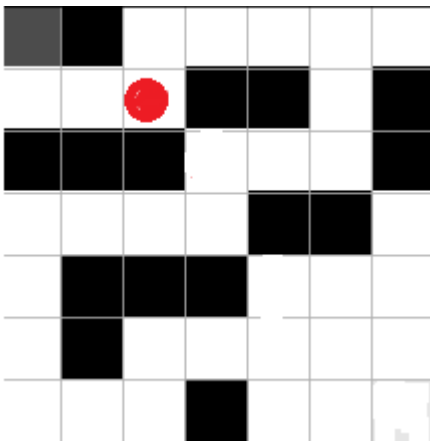


Τα ελεύθερα γειτονικά τετράγωνα είναι τα (5,3),(4,4) και (6,5). Τα δύο πρώτα έχουν υπολογιστεί στην προηγούμενη κίνηση οπότε υπολογίζει μόνο το (6,5).

Για το (6,5):

```
What is the first square?6
What is the second square?5
[[2.528465]]
```

Η μικρότερη απόσταση manhattan είναι του τετραγώνου (6,5) και είναι ίση με 2.528465. Οπότε ο πράκτορας μετακινείται σε αυτό το τετράγωνο.



Τα ελεύθερα τετράγωνα είναι τα (5,4) , (7,5) και (6,6). Το πρώτο έχει υπολογιστεί στο προηγούμενο βήμα οπότε υπολογίζει με την βοήθεια το νευρωνικού δικτύου τα δύο τελευταία τετράγωνα.

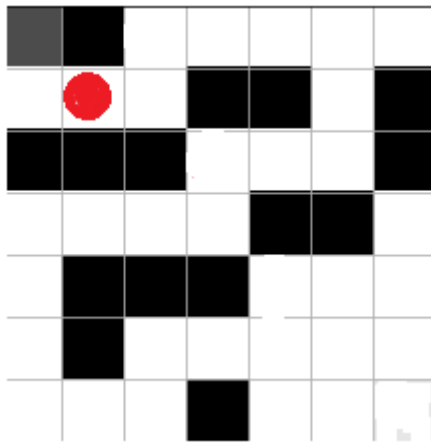
Για το (7,5):

```
What is the first square?7
What is the second square?5
[[8.01303]]
```

Για το (6,6):

```
What is the first square?6
What is the second square?6
[[1.9947474]]
PS C:\hello_ds>
```

Παρατηρεί ότι η μικρότερη απόσταση manhattan είναι του τετραγώνου (6,6) που είναι ίση με 1.054056 , οπότε μεταβαίνει σε αυτό.



Τα ελεύθερα γειτονικά τετράγωνα είναι τα (7,5), (6,5),(6,7) και (7,7). Το πρώτο έχει υπολογιστεί στην προηγούμενη κίνηση οπότε υπολογίζει το (7,4), (6,5) και το (7,7).

Για το (7,4):

```

illed to use: AVX AVX2
What is the first square?7
What is the second square?4
[[12.01915]]

```

Για το (6,5):

```

What is the first square?6
What is the second square?5
[[6.0060463]]

```

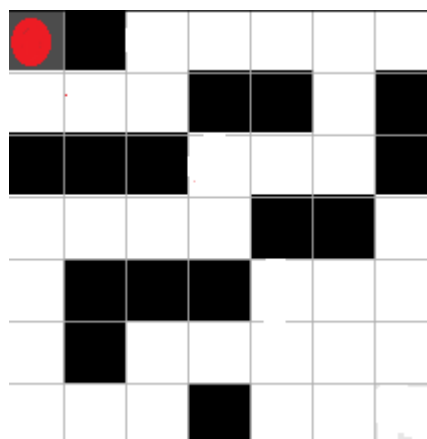
Για το (7,7):

```

2020-12-14 02:48:12.478755: I tensorflow
illed to use: AVX AVX2
What is the first square?7
What is the second square?7
[[0.00078031]]

```

Παρατηρεί ότι η μικρότερη απόσταση manhattan είναι του τετραγώνου (7,7) που είναι ίση με 0.00078031. Οπότε μεταβαίνει στο τετράγωνο (8,8) που είναι και το τέλος του λαβυρίνθου μας.



Στην υλοποίηση αυτή δεν ασχοληθήκαμε με τον υπολογισμό της ακρίβειας (accuracy). Το νευρωνικό μας δίκτυο υπολογίζει τιμές συναρτήσεων με ακρίβεια δεκαδικών αριθμών και εάν υπολογίσουμε την ακρίβεια του δικτύου πολλές τιμές που είναι έξοδοι του νευρωνικού μας δικτύου διαφέρουν ελάχιστα από την αντίστοιχη τιμή της συνάρτησης manhattan . Αυτό θα έχει σαν αποτέλεσμα να ληφθούν ως λανθασμένες αρκετές έξοδοι του νευρωνικού μας δικτύου. Από την χρήση του όμως παρατηρούμε ότι τα αποτελέσματα είναι ικανοποιητικά.

ΚΕΦΑΛΑΙΟ 5 – ΑΛΛΕΣ ΕΦΑΡΜΟΓΕΣ ΤΩΝ ΤΕΧΝΗΤΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ

Όπως έχουμε αναφέρει και στο κεφάλαιο 1 τα νευρωνικά δίκτυα είναι ένας επιστημονικός κλάδος που βρίσκει εφαρμογή στην αντιμετώπιση αρκετών προβλημάτων. Αυτό συμβαίνει γιατί διευκολύνουν την επίλυση σύνθετων προβλημάτων που άμα προσπαθήσουμε να τα επιλύσουμε με αλγοριθμικές μεθόδους , οδηγούμαστε στην δημιουργία πολύπλοκων αλγορίθμων και πολύπλοκων και μεγάλων κωδίκων. Γενικότερα τα τεχνητά νευρωνικά δίκτυα μας βοηθούν στην επίλυση των προβλημάτων χωρίς εκτεταμένο προγραμματισμό με συγκεκριμένους κανόνες και προϋποθέσεις. Είναι απλουστευμένα μοντέλα που χρησιμοποιούνται για πολλά είδη προβλημάτων με το μεγαλύτερο μέρος των πολύπλοκων μαθηματικών υπολογισμών να λαμβάνει μέρος πίσω από τα παρασκήνια. Επίσης είναι πολύ πιο γρήγορα στην σχεδίαση,εκπαίδευση και πρόβλεψη σε σχέση με τα συμβατικά προγράμματα που υλοποιούν τους αλγορίθμους.

Ορισμένες εφαρμογές στις οποίες χρησιμοποιούνται ευρέως είναι οι παρακάτω:

Εφαρμογές αναγνώρισης εικόνας: Αναγνώριση ή ταξινόμηση εικόνας είναι ο διαχωρισμός των εικόνων σε ορισμένες προκαθορισμένες διακριτές κατηγορίες. Οι μελετητές χρησιμοποιούν την μηχανική μάθηση και ιδιαίτερα τις αρχιτεκτονικές των τεχνητών νευρωνικών δικτύων για να την αναγνώριση εικόνας.

Ειδικότερα υπάρχουν οι παρακάτω περιπτώσεις :

Κατηγοριοποίηση εικόνων: Διαχωρισμός των εικόνων ανάλογα με τα αντικείμενα τα οποία περιέχουν π.χ. Αν υποθέσουμε ότι έχουμε εικόνες που περιέχουν το 0 και το 1 και πρέπει να τις διαχωρίσουμε σε δύο ομάδες , όπου στην πρώτη ομάδα

τοποθετούνται αυτές που περιέχουν το 1 και στην άλλη ομάδα τοποθετούνται αυτές που περιέχουν το 0.

Ανίχνευση αντικειμένων: Μία εικόνα μπορεί να περιέχει πολλά διαφορετικά αντικείμενα. Σε αυτή την περίπτωση η αναγνώριση εικόνας έχει ως σκοπό την ανίχνευση της θέσης κάποιου ή κάποιων συγκεκριμένων αντικειμένων που περιέχονται στην εικόνα. Ο εντοπισμός αυτών των αντικειμένων γίνεται συνήθως με την εμφάνιση πλαισίων στο συγκεκριμένο αντικείμενο.

Έτσι, το λογισμικό αναγνώρισης εικόνων και οι εφαρμογές μπορούν να ορίσουν τι απεικονίζεται σε μια εικόνα και να διακρίνουν το ένα αντικείμενο από το άλλο.

Αναγνώριση ομιλίας και λόγου: Το σύστημα αναγνώρισης ομιλίας μετατρέπει τα σήματα ομιλίας, τα λόγια μας δηλαδή και τα αποκωδικοποιεί σε κείμενο ή σε κάποια μορφή νοήματος. Εκτός από την Δυναμική στρέβλωση χρόνου (Dynamic time warping) και τα κρυμμένα μοντέλα Μαρκόβ (Hidden Markov models) χρησιμοποιούνται ευρέως τα τεχνητά νευρωνικά δίκτυα. Οι συγκρίσεις των διαφόρων μεθόδων αναγνώρισης ομιλίας έχουν δείξει ότι τα τεχνητά νευρωνικά δίκτυα παρουσιάζουν μεγάλη απόδοση σε αυτόν τον τομέα.

Ταξινόμηση κειμένου: Η ταξινόμηση κειμένου είναι ουσιαστικό μέρος της αναζήτησης και του φιλτραρίσματος εγγράφων, των διαδικτυακών αναζητήσεων μας στον παγκόσμιο ιστό και της αναγνώρισης γλώσσας και ανάλυσης συναισθημάτων. Τα νευρωνικά δίκτυα χρησιμοποιούνται ευρέως για τέτοιου είδους εργασίες. Η αναγνώριση οντοτήτων και τα τμήματα της προσθήκης ετικετών είναι μερικές από τις εφαρμογές που εμπίπτουν στον τομέα της επεξεργασίας φυσικής γλώσσας (NLP).

Ιατρική: Τα τεχνητά νευρωνικά δίκτυα χρησιμοποιούνται στην ιατρική για διαγνώσεις και πιστεύεται ότι θα λάβουν ευρεία εφαρμογή τα επόμενα χρόνια. Στην σημερινή εποχή, χρησιμοποιούνται για ανιχνεύσεις ασθενειών σε διάφορους κλάδους της ιατρικής, ειδικά στην καρδιολογία και την ακτινολογία. Τέτοιες περιπτώσεις είναι η ανίχνευση ασθενειών σε ακτινογραφίες, σε καρδιογραφήματα και σε υπερήχους. Τα νευρωνικά δίκτυα εκπαιδεύονται και μαθαίνουν από τα παραδείγματα και τα πρότυπα εκπαίδευσης οπότε δεν απαιτούνται λεπτομέρειες για την αναγνώριση κάποιας συγκεκριμένης ασθένειας. Το μόνο που απαιτείται είναι να επιλεγθούν σωστά τα παραδείγματα και τα πρότυπα ώστε να δημιουργηθεί ένα αποτελεσματικό νευρωνικό δίκτυο.

(11. Neural Networks in Medicine-[Jayesh Babu Ahire](#) _Νευρωνικά δίκτυα στην ιατρική)

Οι εφαρμογές των νευρωνικών δικτύων έχουν αυξηθεί και στην εξόρυξη ιατρικών δεδομένων. (<https://pubmed.ncbi.nlm.nih.gov/18690868/>)

(1. <http://kelifos.physics.auth.gr/COURSES/neural/K3.pdf>)

Μετεωρολογία: Η πρόγνωση του καιρού είναι μία διαδικασία όπου ένας τεράστιος αριθμός δεδομένων που προέρχονται από διαφορετικούς τύπους αισθητήρων πρέπει να επεξεργαστεί μέσω πολύπλοκων, χρονοβόρων αλγορίθμων. Αυτή η διεργασία μπορεί να επιτευχθεί από μια νέα προσέγγιση που είναι η μηχανική μάθηση και τα νευρωνικά δίκτυα. Ειδικότερα ένα αισθητήριο σύστημα χρησιμοποιείται για τη "λήψη" μετεωρολογικών δεδομένων σε έναν μετεωρολογικό σταθμό. Τα δεδομένα αυτά θα χρησιμοποιηθούν ως πρότυπα για την εκπαίδευση του τεχνητού νευρωνικού δικτύου.

Μετά την εκπαίδευση του νευρωνικού δικτύου το μετεωρολογικό σύστημα είναι σε θέση να προβλέψει την εξέλιξη αυτών των παραμέτρων σε ένα ορισμένο χρονικό

διάστημα, δίνοντας πολύτιμες ενδείξεις για την πιθανότητα καύσωνα , βροχής , ομίχλης , χαλαζιού ή χιονιού στο κοντινό μέλλον.

(12.Use of ANN models in the prediction of meteorological data - P. Rajendra · K. V. N. Murthy · A. Subbarao Rahul Boadh – Χρήση των νευρωνικών δικτύων στα μοντέλα πρόβλεψης καιρού.)

Οικονομία: Τα νευρωνικά δίκτυα έχουν μεγάλη εφαρμογή και στα οικονομικά. Βοηθούν τις εταιρείες χρηματοοικονομικών υπηρεσιών να αυτοματοποιήσουν ολοένα και πιο περίπλοκες διαδικασίες και να πάρουν αποφάσεις σχετικές με τα το κόστος, την επικινδυνότητα μια ενέργειας και την πιθανότητα να υπάρξει ζημιά ή κέρδος σε κάποια οικονομική δραστηριότητα. Αυτά δίνουν στις επιχειρήσεις ένα ανταγωνιστικό πλεονέκτημα.

Τραπεζικό Σύστημα: Στις τραπεζικές εργασίες είναι δύσκολο να υπολογιστεί ο παράγοντας επικινδυνότητας για τις αιτήσεις των δανείων , ιδιαίτερα των στεγαστικών. Με βάση τα στοιχεία που μπορεί να περιέχει κάποια αίτηση για δάνειο , η τράπεζα θέλει να γνωρίζει τι πιθανότητα υπάρχει να πληρώσει ο πελάτης το δάνειο. Συλλέγοντας χιλιάδες αιτήσεις όπου οι μισές απορρίφθηκαν και οι μισές εγκρίθηκαν , έχουν σχεδιαστεί νευρωνικά δίκτυα για τον υπολογισμό της παραπάνω πιθανότητας.

Εντοπισμός απάτης για οικονομικές συναλλαγές.

(13.Detecting Fraudulent Bank Account Based on Convolutional Neural Network with Heterogeneous Data-Roberto Caldelli- Εντοπισμός απάτης τραπεζικού λογαριασμού με συμβατικό νευρωνικό δίκτυο ετερογενή Δεδομένα)

Προβλέψεις υπερανάλιψης βάσει του ιστορικού συναλλαγών του πελάτη.

Κεφαλαιαγορές: Τα νευρωνικά δίκτυα βοηθούν τους εμπόρους και τις εταιρίες να αποφασίσουν τι τιμή να αναφέρουν όταν αγοράζουν ή πωλούν ομόλογα ή προϊόντα για τους πελάτες τους βάσει ιστορικών και πραγματικών δεδομένων αγοράς. Τα ιστορικά και πραγματικά δεδομένα θα χρησιμοποιηθούν ως πρότυπα για την εκπαίδευση του νευρωνικού δικτύου.

Εξαγωγή πληροφοριών σχετικά με τα κέρδη ή τις ζημιές από χρηματοοικονομικές εκθέσεις για να βοηθήσουν στη λήψη αποφάσεων για επενδύσεις Εξαγωγή πληροφοριών και σύνοψη νομικών εγγράφων.

(14.A Neural Network Approach to Financial Forecasting - P.Enyindah ,OnwuachuUzochukwu C. -Ένα νευρωνικό δίκτυο για την προσέγγιση οικονομικών προβλέψεων).

Επίλογος

Σκοπός της παρούσας εργασίας ήταν να παρουσιαστεί η δομή, η λειτουργία και η εκπαίδευση των τεχνητών νευρωνικών δικτύων και οι τομείς στους οποίους το ανωτέρω πεδίο της τεχνητής νοημοσύνης έχει βρεί μεγάλη εφαρμογή. Δόθηκε ιδιαίτερη έμφαση στην ανάπτυξη νευρωνικών δικτύων για την διάσχιση λαβυρίνθων και αποδείχθηκε ότι και στο συγκεκριμένο θέμα τα δίκτυα αυτά μπορούν να πετύχουν ικανοποιητικά αποτελέσματα. Γενικά τα τεχνητά νευρωνικά δίκτυα μπορούν να επιλύσουν ικανοποιητικά έναν μεγάλο αριθμό προβλημάτων στα οποία η αλγοριθμική επίλυση είναι δυσχερής.

Βιβλιογραφία

Δημοσιεύσεις:

1. <http://kelifos.physics.auth.gr/> -Α.Π.Θ.
2. What We Know About the Brain Structure -Karla Batista-García-Ramó1, and Caridad Ivette Fernández-Verdecia – Τι γνωρίζουμε για τον ανθρώπινο εγκέφαλο
3. Wilder Penfield redrew the map of the brain by opening the heads of living patients by Brian Resnick www.vox.com/science-and-health
4. Anatomy of the Brain - Sumit Kumar – Τι γνωρίζουμε για τον ανθρώπινο εγκέφαλο)
5. Introduction to Neurons - Alla Borisjuk ,Avner FriedmanBard ,Ermentrout, David Terman -Εισαγωγή στους νευρώνες
6. <https://www.cognifit.com/el/science/didyou-know/neurons>
8. Artificial neural networks: fundamentals, computing, design, and applicationa, bI.A. Basheer, M. Hajmeer
9. On the Origin of Deep Learning ,Haohan ,Bhiksha Raj – Προέλευση της μηχανικής μάθησης

10. Understanding Backpropagation Algorithm- Simeon Kostadinov-Κατανόηση αλγορίθμου οπισθοδιάδοσης

11. Neural Networks in Medicine-[Jayesh Babu Ahire](#) _Νευρωνικά δίκτυα στην ιατρική

12. Use of ANN models in the prediction of meteorological data - P. Rajendra · K. V. N. Murthy · A. Subbarao Rahul Boadh – Χρήση των νευρωνικών δικτύων στα μοντέλα πρόβλεψης καιρού.

13. Detecting Fraudulent Bank Account Based on Convolutional Neural Network with Heterogeneous Data-Roberto Caldelli- Εντοπισμός απάτης τραπεζικού λογαριασμού με συμβατικό νευρωνικό δίκτυο ετερογενή Δεδομένα.

14. A Neural Network Approach to Financial Forecasting - P.Enyindah ,OnwuachuUzochukwu C. -Ένα νευρωνικό δίκτυο για την προσέγγιση οικονομικών προβλέψεων

Βιβλία

15. I. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας, Η. Σακελλαρίου. *Τεχνητή Νοημοσύνη- Γ' Έκδοση*,[Εκδόσεις Πανεπιστημίου Μακεδονίας](#), 2011

16. *ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ -ΜΙΑ ΣΥΓΧΡΟΝΗ ΠΡΟΣΕΓΓΙΣΗ*

Εκδόσεις Κλειδάριθμος

[RUSSELL STUART, NORVIG PETER](#)

17. Τεχνητά νευρωνικά δίκτυα – Διαμαντάρας Κωμσταντίνος-Εκδόσεις Κλειδάριθμος

18. Νευρωνικά δίκτυα και μηχανική μάθηση- Haykin Simon – Εκδόσεις Κλειδάριθμος

19. Τεχνητά Νευρωνικά Δίκτυα Θεωρία & Εφαρμογές-Ρίζος Γεώργιος – Εκδόσεις Νέων Τεχνολογιών

Ιστοσελίδες

20. <https://www.samyzaf.com/ML/rl/qmaze.html>

21. <https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/>

22. <https://towardsdatascience.com/building-our-first-neural-network-in-keras-bdc8abb17f5>

23. <https://victorzhou.com/blog/keras-neural-network-tutorial/>

24. <https://www.tensorflow.org/tutorials/keras/classification>

25. https://medium.com/@sandy_lee/how-to-train-neural-networks-for-image-classification-part-1-21327fe1cc1

Παράρτημα – Κώδικες εφαρμογών

```
cross.py > ...
1  import numpy as np
2  from keras.models import Sequential
3  from keras.layers.core import Dense
4
5  model = Sequential()
6  model.add(Dense(3, input_dim=8, activation='relu'))
7  model.add(Dense(3, activation='relu'))
8  model.add(Dense(3, activation='sigmoid'))
9  model.add(Dense(1, activation='softmax'))
10 model.load_weights('model.h5')
11
12 a = input("What is the first square?")
13 b = input("What is the second square?")
14 c = input("What is the third square?")
15 d = input("What is the fourth square?")
16 e = input("What is the fifth square?")
17 f = input("What is the sixth square?")
18 g = input("What is the seventh square?")
19 h = input("What is the eighth square?")
20
```

```
t_data= np.array([[a,b,c,d,e,f,g,h]], "float32")

def my_function(a):
    w=model.predict(a).round()
    return w

s = my_function(t_data)
print(s)
```

Κώδικας για το αρχείο cross.

```
traverse.py > ...
1  import tensorflow as tf
2
3  from tensorflow import keras
4  import numpy as np
5  from numpy import loadtxt
6  from keras.models import Sequential
7  from keras.layers import Dense
8
9
10
11  c= loadtxt('maze.csv', delimiter=',')
12  d = loadtxt('result.csv', delimiter=',')
13
14
15  model = Sequential()
16  model.add(Dense(3, input_dim=8, activation='relu'))
17  model.add(Dense(3, activation='relu'))
18  model.add(Dense(3, activation='sigmoid'))
19  model.add(Dense(1, activation='softmax'))
20  model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
21  model.fit(c, d, epochs=150, batch_size=5)
```

```
traverse.py > ...
22
23 _, accuracy = model.evaluate(c, d)
24 print('Accuracy: %.2f' % (accuracy*100))
25
26
27 predictions = model.predict_classes(c)
28
29 for i in range(33):
30     print('%s => %d (expected %d)' % (c[i].tolist(), predictions[i], d[i]))
31
32
33 model.save_weights('model.h5')
```

Κώδικας για το τεχνητό νευρωνικό δίκτυο του αρχείου traverse.py.

```
euclidean.py > ...
1 import numpy as np
2 import math
3
4 valA = int(input("What is x coordinate?"))
5 valB = int(input("What is y coordinate?"))
6
7 a=((7-valA)**2)
8 b= ((7-valB)**2)
9 p1 = [a,b]
10 p2=[7,7]
11
12
13 distance = math.sqrt(a+b)
14 print(distance)
15
16
17
```

Κώδικας της ευκλίδειας απόστασης του αρχείου euclidean.py.

```
manh.py > ...
1  import tensorflow as tf
2
3  from tensorflow import keras
4  import numpy as np
5  from numpy import loadtxt
6  from keras.models import Sequential
7  from keras.layers import Dense
8
9
10
11  c = loadtxt('values.csv', delimiter=',')
12  d = loadtxt('outs.csv', delimiter=',')
13
14
15  model = Sequential()
16  model.add(Dense(3, input_dim=2, activation='relu'))
17  model.add(Dense(5, activation='relu'))
18  model.add(Dense(5, activation='sigmoid'))
19  model.add(Dense(1, activation='softmax'))
20  model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
21  model.fit(c, d, epochs=150, batch_size=5)
22
```

```
manh.py > ...
23
24
25  predictions = model.predict_classes(c)
26
27  for i in range(33):
28  | print('%s => %d (expected %d)' % (c[i].tolist(), predictions[i], d[i]))
29
30  model.save_weights('model.h7')
```

Κώδικας του αρχείου manh.py που υπολογίζει την απόσταση μανχάταν μέσω νευρωνικού δικτύου.