ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Τεχνητή νοημοσύνη και blockchain

Χρύσανθος Χ. Παπαδάκης

Επιβλέπουσα καθηγήτρια: Ελένη Αικατερίνη Λελίγκου

ΑΙΓΑΛΕΩ

ΦΕΒΡΟΥΑΡΙΟΣ 2021

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

ELENI AIKATERINI LELIGKOU
Digitally signed by ELENI AIKATERINI LELIGKOU
Date: 2021.03.08 10:37:12 +02'00'

Ελένη Αικατερίνη Λελίγκου

Christos Drosos
Digitally signed by Christos Drosos
Date: 2021.03.07 10:27:00 +02'00'

Χρήστος Δρόσος

Dimitrios Kantzos
Digitally signed by Dimitrios Kantzos
Date: 2021.03.08 09:06:56 +02'00'

Κάντζος Δημήτριος

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/η κάτωθι υπογεγραμμένος/η Παπαδάκης Χρύσαθος - Χρήστος του Κωνσταντίνου, με αριθμό μητρώου 71446636 φοιτητής/τρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής, δηλώνω υπεύθυνα ότι:

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

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο/Η Δηλών/ούσα

Χ. Χ. Παπαδάκης

ΕΥΧΑΡΙΣΤΙΕΣ

Με την παρούσα διπλωματική εργασία θα ήθελα αρχικά να ευχαριστήσω την οικογένεια μου και τους φίλους μου οι οποίοι ήταν πάντα δίπλα μου και δεν έπαψαν ποτέ να πιστεύουν σε εμένα. Στην συνέχεια θα ήθελα να ευχαριστήσω την κυρία Λελίγκου για την συνεχή και άμεση υποστήριξη της καθ' όλη την διάρκεια της εκπόνησης αυτής της διπλωματικής. Τέλος θα ήθελα να ευχαριστήσω τον κύριο Ανδρέα Σορτ ο οποίος ήταν πρόθυμος να μου λύσει οποιαδήποτε απορία και για όλη την βοήθεια που μου προσέφερε σε ό,τι αφορά το πειραματικό κομμάτι της εργασίας.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ

Η έννοια της ομόσπονδης μάθησης προτάθηκε από την Google για πρώτη φορά το 2017. Κύρια ιδέα αυτής της μεθόδου μηχανικής μάθησης, είναι η δημιουργία μοντέλων τεχνητής νοημοσύνης τα οποία βασίζονται σε σύνολα δεδομένων που είναι διαθέσιμα σε πολλαπλές συσκευές ενώ παράλληλα αποφεύγεται η γνωστοποίηση αυτών των δεδομένων σε τρίτους. Πρόσφατες βελτιώσεις εστιάζουν στην υπερκέραση των στατιστικών προκλήσεων καθώς και στην θωράκιση ασφαλείας ενός τέτοιου μοντέλου. Οι απαιτήσεις για την ανάπτυξη ενός συστήματος ομόσπονδης μάθησης είναι τεράστιες, με τη συλλογή δεδομένων απο διαφορετικές συσκευές, το σεβασμό στην ιδιωτικότητα και την προστασία από κακόβουλες ενέργειες να είναι μερικές από τις πιο σημαντικές. Συμπεραίνεται λοιπόν ότι η κατασκευή ενός μοντέλου ομόσπονδης μάθησης, λαμβάνοντας όλα τα παραπάνω υπόψη, καθίσταται ακόμα δυσκολότερη από αυτή που μπορεί να ήταν αρχικά. Για να πληρεί ένα τέτοιο σύστημα τις παραπάνω απαιτήσεις εφαρμόζονται αλγόριθμοι ανάλυσης δεδομένων και σύγχρονα πρωτόκολλα κατανομής δεδομένων (π.χ Blockchain). Με την ραγδαία εξέλιξη της τεχνολογίας τέτοιες μέθοδοι μηχανικής μάθησης βρίσκουν ολοένα και περισσότερο πρακτικές εφαρμογές.

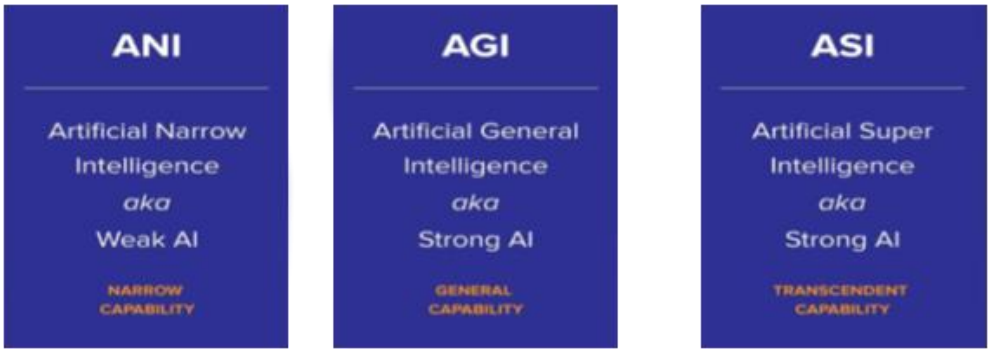
ΕΙΣΑΓΩΓΗ

Η μηχανική μάθηση είναι ένας τεχολογικός τομέας ο οποίος εξελίσσεται συνεχώς. Με τις δυνατότητες που έχει να προσφέρει να είναι πολύ μεγάλες, συστήματα μηχανικής μάθησης χρησιμοποιούνται ευρέως σε βιομηχανίες και εταιρείες για την αυτοματοποίηση πολύπλοκων διεργασιών. Τα αρχικά μοντέλα επεξεργασίας πληροφοριών στην τεχνητή νοημοσύνη περιλαμβάνουν απλές διαδικασίες συλλογής δεδομένων, με τα δεδομένα αυτά στη συνέχεια να ταξινομούνται και να συγχωνεύονται. Στόχος αυτών των διαδικασιών είναι η ανάπτυξη και εκπαίδευση του κύριου μοντέλου το οποίο είναι και το τελικό προϊόν. Ωστόσο με τους περισσότερους τομείς να έχουν πρόσβαση σε περιορισμένο αριθμό δεδομένων ή να μην είναι σε θέση να αποκτήσουν ποιοτικά δεδομένα, η εφαρμογή τεχνολογιών τεχνητής νοημοσύνης καθίσταται πολύ δυσκολότερη. Η ανταλλαγή και χρήση των δεδομένων που έχουν συγκεντρωθεί από διάφορους οργανισμούς θα μπορούσε να δώσει την λύση στο πρόβλημα της πρόσβασης. Αυτή η κλασσική διεργασία όμως, αντιμετωπίζει πολλές προκλήσεις από την θέσπιση νέων νομοθεσιών προστασίας προσωπικών δεδομένων. Μία νέα προσέγγιση για την αντιμετώπιση των παραπάνω προκλήσεων είναι γνωστή ως ομόσπονδη μάθηση. Η ομόσπονδη μάθηση είναι μία νέα προσέγγιση της διανεμημένης μάθησης η οποία επιτρέπει την εκπαίδευση ενός μοντέλου τεχνητής νοημοσύνης έχοντας πρόσβαση σε μία τεράστια συλλογή ποιοτικών δεδομένων τα οποία βρίσκονται σε διάφορες συσκευές χρηστών. Πρόκειται για ένα κομμάτι μιας γενικότερης προσέγγισης η οποία έχει ως στόχο την μεταφορά του συστήματος στο μέρος που βρίσκονται τα δεδομένα έναντι της κλασσικής προσέγγισης που είχε ως στόχο την μεταφορά των δεδομένων στο κύριο σύστημα. Με αυτή την μέθοδο η εταιρία που είναι υπέυθυνη για την παραγωγή του τελικού μοντέλου δεν μπορεί να έχει πρόσβαση στα δεδομένα των χρηστών που χρησιμοποιήθηκαν παρά μόνο στις παραμέτρους που θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου. Παρόλα αυτά οι τεχνικές που εφαρμόζονται για την ανάπτυξη ενός κλασσικού μοντέλου μηχανικής μάθησης είναι απαραίτητες για την δημιουργία ενός συστήματος ομόσπονδης μάθησης.

1. ΤΙ ΕΙΝΑΙ Η ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

Ο όρος τεχνητή νοημοσύνη χρησιμοποιήθηκε για πρώτη φορά το 1956. Περιγράφεται ως μία προσπάθεια μοντελοποίησης της ανθρώπινης νοημοσύνης με στόχο την δημιουργία εξελιγμένων υπολογιστών. Η μοντελοποίηση αυτή πραγματοποιείται με την βοήθεια μαθηματικών αλγορίθμων. Ωστόσο ακόμα και σήμερα που ο τεχνολογικός τομέας της τεχνητής νοημοσύνης βρίσκει πληθώρα εφαρμογών, η επίτευξη του αρχικού στόχου βρίσκεται ακόμα μακριά. Ο λόγος για τον οποίο συστήματα τεχνητής νοημοσύνης χρησιμοποιούνται ευρέως σήμερα είναι η αυτοματόποιηση πολύπλοκων διεργασιών η οποία δεν ήταν δυνατή προηγουμένως με την μέθοδο του κλασσικού προγραμματισμού. Η τεχνητή νοημοσύνη διακρίνεται σε τρεις κατηγορίες: περιορισμένη (Narrow AI), γενική (General AI) και πολύ ισχυρή (Super AI) όπως φαίνεται στη εικόνα 1.1.

Μία περιορισμένη τεχνητή νοημοσύνη είναι σε θέση να πραγματοποιήσει μία πολύπλοκη διεργασία με εξαιρετική ακρίβεια χωρίς όμως να μπορεί να πραγματοποιήσει άλλες απαιτητικές διεργασίες που είναι εκτός του πεδίου της. Αυτή η κατηγορία τεχνητής νοημοσύνης χρησιμοποιείται απο πολλές βιομηχανίες και μεγάλες εταιρείες σε τομείς όπως είναι η επιχειρηματικότητα και η ιατρική. Ένα τέτοιο σύστημα είναι το σύστημα πρότασης προϊόντων από την Amazon, το οποίο χρησιμοποιώντας τις αγορές που μπορεί να έχει κάνει ένας πελάτης προτείνει προϊόντα με βάση τις προτιμήσεις του. Πολλές εταιρείες επενδύουν στην ανάπτυξη τέτοιων συστημάτων ελπίζοντας στο μέλλον να είναι εφικτή η δημιουργία μίας γενικής τεχνητής νοημοσύνης. Μία γενική τεχνητή νοημοσύνη θα είναι σε θέση να αναλάβει πολλές και περίπλοκες διεργασίες τις οποίες θα εκτελεί με πολύ μεγάλη ακρίβεια. Η κύρια διαφορά της σε σχέση με μία περιορισμένη τεχνητή νοημοσύνη είναι ότι συγκλίνει πολύ περισσότερο στον πρωταρχικό στόχο που έχει ο κλάδος της τεχνητής νοημοσύνης που είναι η λύση πολύ σύνθετων προβλημάτων από έναν υπολογιστή χωρίς καμία ανθρώπινη παρέμβαση. Το επόμενο βήμα θα ήταν η δημιουργία μίας πολύ ισχυρής τεχνητής νοημοσύνης που θα ξεπερνούσε σε δυνατότητες ακόμα και τον ίδιο τον άνθρωπο. Ωστόσο η ανάπτυξη της φαίνεται προς το παρόν αδύνατη.

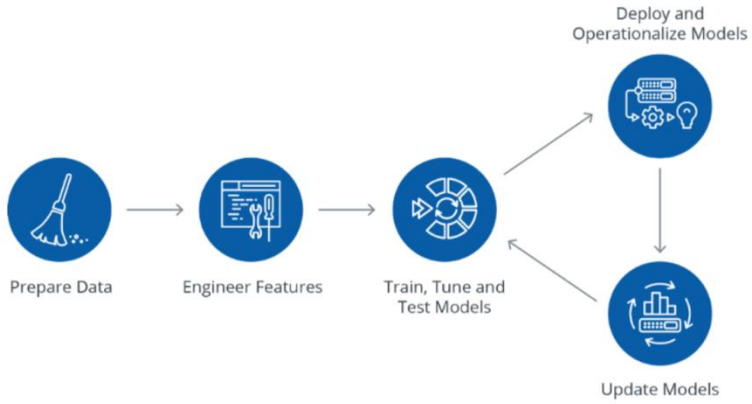


Εικόνα 1.1: Κατηγορίες τεχνητής νοημοσύνης

1.1 Μηχανική μάθηση και βαθιά μάθηση

1.1.1 Μηχανική μάθηση (Machine learning)

Ένα από τα κύρια προβλήματα που είχε αναλάβει να επιλύσει ο τομέας της τεχνητής νοημοσύνης ήταν η δημιουργία μεθόδων οι οποίες θα επιτρέπανε την ανάπτυξη ενός συστήματος υπεύθυνου για την υλοποίηση σύνθετων διεργασιών. Σε αυτό το σημείο εισάχθηκε για πρώτη φορά η ένοια της μηχανικής μάθησης. Η μηχανική μάθηση είναι μία υποκατηγορία της τεχνητής νοημοσύνης η οποία επικεντρώνεται στην εκπαίδευση των υπολογιστών χρησιμοποιώντας ένα πλήθος δεδομένων για την επίλυση συγκεκριμένων προβλημάτων. Πιο συγκεκριμένα η ιδέα πάνω στην οποία βασίζεται η μηχανική μάθηση είναι η δυνατότητα δημιουργίας αλγορίθμων οι οποίοι μπορούν να μαθαίνουν και να κάνουν προβλέψεις βασιζόμενοι σε δεδομένα. Προκειμένου να υλοποιηθεί ένα τέτοιο μοντέλο είναι απαραίτητη η χρήση δειγμάτων καθώς και χαρακτηριστικών τα οποία θα χρησιμοποιήσει με την σειρά του ο αλγόριθμος για να βρει μία σχέση μεταξύ των δεδομένων που θα του τροφοδοτηθούν, δίνοντας έτσι τελικά το επιθυμητό αποτέλεσμα.

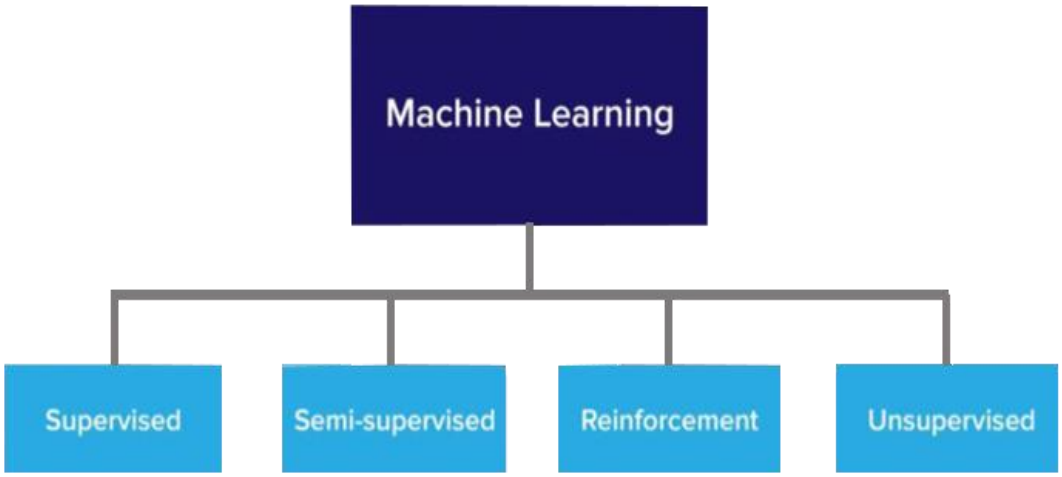


Εικόνα 1.2: Προετοιμασία δεδομένων πριν την τροφοδοσία τους σε ένα μοντέλο

Τα συστήματα μηχανικής μάθησης εκπαιδεύονται από ειδικές συλλογές δειγμάτων που ονομάζονται σύνολα δεδομένων (datasets). Τα δείγματα αυτά μπορούν να αποτελούνται από αριθμούς, εικόνες, κείμενο ή οποιοδήποτε άλλον τύπο δεδομένων. Στη συνέχεια τα δεδομένα επεξεργάζονται ώστε να είναι κατάλληλα για την εκπαίδευση του κύριου μοντέλου (εικόνα 1.2). Κατά την διάρκεια αυτής της επεξεργασίας κάθε δεδομένο περιγράφεται από κάποια χαρακτηριστικά (features). Τα χαρακτηριστικά αυτά είναι πολύ σημαντικά καθώς αποτελούν κύριο παράγοντα που συντελεί στην επίλυση του προβλήματος που έχει ανατεθεί. Ουσιαστικά επιδεικνύουν στον αλγόριθμο που να εστιάσει περισσότερο. Η επιλογή των χαρακτηριστικών ενός συνόλου δεδομένων γίνεται έτσι ώστε ο κύριος αλγόριθμος να είναι σε θέση να μπορεί να βρει μία σχέση μεταξύ των δεδομένων. Σημειώνεται ότι η παραπάνω περίπτωση αναφέρεται στο κομμάτι της μηχανικής μάθησης που ονομάζεται εποπτευόμενη μάθηση

(supervised learning). Αντίθετα στο κομμάτι της μη εποπτευόμενης μάθησης (unsupervised learning) ο αλγόριθμος προσπαθεί να βρει μία σχέση μεταξύ των δεδομένων χωρίς να έχουν δημιουργηθεί κάποια χαρακτηριστικά για το καθένα από αυτά. Πρόκειται για μία προσέγγιση της μηχανικής μάθησης η οποία δεν απαιτεί σε τόσο μεγάλο βαθμό ανθρώπινη παρέμβαση.
Η επίλυση ενός προβλήματος με την βοήθεια της μηχανικής μάθησης, μπορεί να γίνει εφικτή με την βοήθεια διαφόρων μοντέλων. Ανάλογα την επιλογή η ακρίβεια και η ταχύτητα στη λήψη των αποτελεσμάτων μπορεί να διαφέρει. Σε μερικές περιπτώσεις χρησιμοποιείται ένας συνδυασμός αλγορίθμων για να επιτευχθεί ακόμα καλύτερη απόδοση. Οποιοδήποτε λογισμικό μηχανικής μάθησης που πραγματοποιεί συγκεκριμένες διεργασίες είναι περισσότερο ανεξάρτητο από ένα κλασσικό λογισμικό που έχει αναπτυχθεί με την μορφή κλασσικού προγραμματισμού ο οποίος βασίζεται απλά σε ένα σύνολο οδηγιών. Ένα σύστημα μηχανικής μάθησης μαθαίνει να αναγνωρίζει σχέσεις μεταξύ των δεδομένων που του έχουν τροφοδοτηθεί και στη συνέχεια να κάνει διάφορες προβλέψεις. Αν η ποιότητα των δειγμάτων που έχουν χρησιμοποιηθεί είναι υψηλή και τα χαρακτηριστικά που έχουν επιλεχθεί κατάλληλα, τότε η ακρίβεια ενός τέτοιου συστήματος είναι πολύ μεγάλη και σε πολλές περιπτώσεις μπορεί να ξεπεράσει σε δυνατότητες ακόμα και τον άνθρωπο στη συγκεκριμένη διεργασία που του έχει ανατεθεί.

Η μηχανική μάθηση, ανάλογα με τον τρόπο εκπαίδευσης ενός μοντέλου μπορεί να χωριστεί σε τέσσερις βασικές κατηγορίες: εποπτευόμενη μάθηση (supervised learning), μη εποπτευόμενη μάθηση (unsupervised learning), ημιεποπτευόμενη μάθηση (semi-supervised learning) και ενισχυμένη μάθηση (reinforcement learning) όπως φαίνεται στην εικόνα 1.3.



Εικόνα 1.3: Κατηγορίες μηχανικής μάθησης

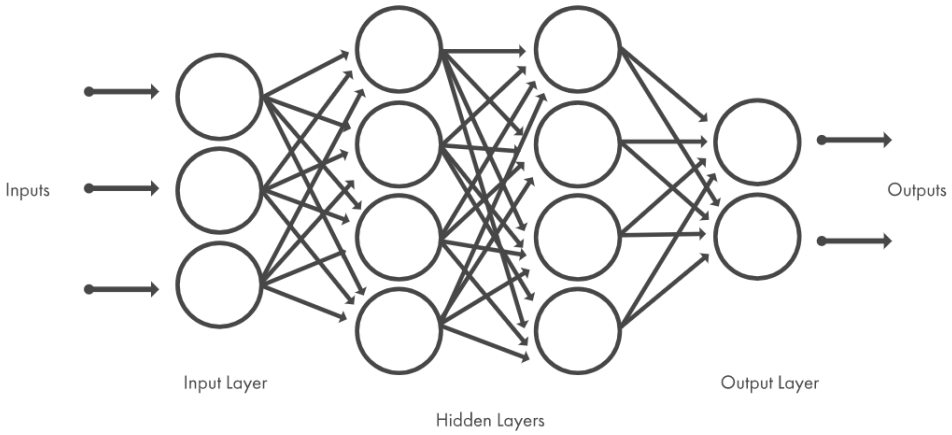
Όπως προαναφέρθηκε η εποπτευόμενη μάθηση στηρίζεται στην εκπαίδευση του μοντέλου με την βοήθεια χαρακτηριστικών που έχουν δοθεί στο τροφοδοτούμενο σύνολο δεδομένων. Ουσιαστικά κάθε δεδομένο έχει και από μία ετικέτα η οποία προσδιορίζεται από τα χαρακτηριστικά του. Αυτού του τύπου δεδομένα ονομάζονται δεδομένα με ετικέτα (labeled data). Για παράδειγμα στην περίπτωση ενός λογισμικού μηχανικής μάθησης το οποίο είναι υπεύθυνο για την αναγνώριση αριθμών, ο κάθε αριθμός έχει και από μία ετικέτα που βοηθά στην κατανόηση των χαρακτηριστικών του. Αρχικά όλα τα δεδομένα εισάγονται στο σύστημα μαζί με τις αντίστοιχες ετικέτες που τους έχουν δοθεί και συνεπώς μπορεί να διακριθούν σε διαφορετικές κατηγορίες. Ο αλγόριθμος μπορεί να αναγνωρίσει τι είναι αυτό που χαρακτηρίζει τον κάθε αριθμό, αρχίζοντας έτσι να συσχετίζει τα δεδομένα μεταξύ τους. Η διαδικασία εκπαίδευσης του μοντέλου συνεχίζεται μέχρι να επιτευχθεί η επιθυμητή ακρίβεια (accuracy). Στη συνέχεια το μοντέλο θα εξεταστεί σε δεδομένα που δεν έχει ξαναδεί. Αυτός ο τύπος μηχανικής μάθησης χρησιμοποιείται για αναγώριση εικόνων, διαχωρισμό και ταξινόμηση δεδομένων. Μερικοί από τους βασικότερους αλγόριθμους που χρησιμοποιεί η εποπτευόμενη μάθηση είναι οι εξής: Logistic regression, Decision tree, K-Nearest neighbors, Support vector machine.

Στην μη εποπτευόμενη μάθηση δεν παρέχονται στα δεδομένα ετικέτες, οπότε ο αλγόριθμος θα πρέπει να βρει σχέσεις μταξύ των δειγμάτων χωρίς κάποια εξωτερική παρέμβαση. Για παράδειγμα ένα μοντέλο μηχανικής μάθησης το οποίο θα πρέπει να αναλύσει τις προτιμήσεις πελατών με βάση το ιστορικό αναζήτησής τους χρησιμοποιεί μη εποπτευόμενη μάθηση. Αυτή η μέθοδος είναι επίσης κατάλληλη για διεισδυτική ανάλυση δεδομένων. Υπάρχουν περιπτώσεις που ο χειριστής ενός τέτοιου λογισμικού δεν είναι ξεκάθαρο τι προσπαθεί να βρει αλλά υπάρχουν μοτίβα και σχέσεις μεταξύ των δεδομένων που το σύστημα μπορεί να ανιχνεύσει. Η μη εποπτευόμενη μάθηση χρησιμοποιείται για την ανάλυση ψεύτικων εικόνων, σε συστήματα συστάσεων, στη συμβούλευση για την διαχείρηση επιχειρήσεων. Μερικά παραδείγματα αλγορίθμων που χρησιμοποιούνται είναι: K-Means clustering, DBSCAN, Mean-shift, Singular value decomposition (SVD).

Η ημιεποπτευόμενη μάθηση είναι μία ανάμειξη των δύο παραπάνω μεθόδων. Τα δεδομένα που θα τροφοδοτηθούν είναι ένας συνδυασμός δειγμάτων με και χωρίς ετικέτα. Σε αυτή την περίπτωση ενώ το επιθυμητό αποτέλεσμα είναι γνωστό, το μοντέλο θα πρέπει να βρει μοτίβα για την δόμηση δεδομένων και να προβλέψει το αποτέλεσμα. Τέλος η μέθοδος της ενισχυμένης μάθησης βασίζεται στη δοκιμή και το σφάλμα. Το μοντέλο ανάλογα με τις ενέργειες που θα πραγματοποιήσει λαμβάνει θετικά ή αρνητικά σήματα τα οποία βοηθούν στην εκπαίδευση του. Με αυτή την μέθοδο δεν είναι απαραίτητη η χρήση στατικών συνόλων δεδομένων, όπως στις παραπάνω μεθόδους, αλλά το μοντέλο είναι σε θέση να εκπαιδευτεί σε δυναμικά περιβάλλοντα. Η ενισχυμένη μάθηση χρησιμοποιείται σε διάφορες περιπτώσεις όπως: στα αυτόνομα αυτοκίνητα, σε ηλεκτρονικά παιχνίδια, στη διαχείριση πόρων.

1.1.2 Βαθιά μάθηση (Deep learning)

Η βαθιά μάθηση είναι μία κατηγορία της μηχανικής μάθησης, η οποία εμπνεύστηκε από την δομή του ανθρώπινου εγκεφάλου. Οι αλγόριθμοι βαθιάς μάθησης χρησιμοποιούν ένα σύμπλεγμα κόμβων οι οποίοι συνδέονται μεταξύ τους δημιουργώντας ένα δίκτυο πολλών επιπέδων, το οποίο ονομάζεται τεχνητό νευρωνικό δίκτυο (εικόνα 1.4). Σε ένα τέτοιο δίκτυο η πληροφορία μεταφέρεται από το ένα επίπεδο κόμβων στο άλλο με την βοήθεια διαδρομών, όπου το τελευταίο επίπεδο είναι υπεύθυνο για την εξαγωγή του τελικού αποτελέσματος. Αν και υπάρχουν και άλλοι αλγόριθμοι που χρησιμοποιούνται στον τομέα της τεχνητής νοημοσύνης, ο αλγόριθμος των νευρωνικών δικτύων φαίνεται να είναι ο πιο αποτελεσματικός εξαιτίας του γεγονότος ότι είναι πολύ ευέλικτος και μπορεί να εφαρμοστεί σε πολλούς τομείς.

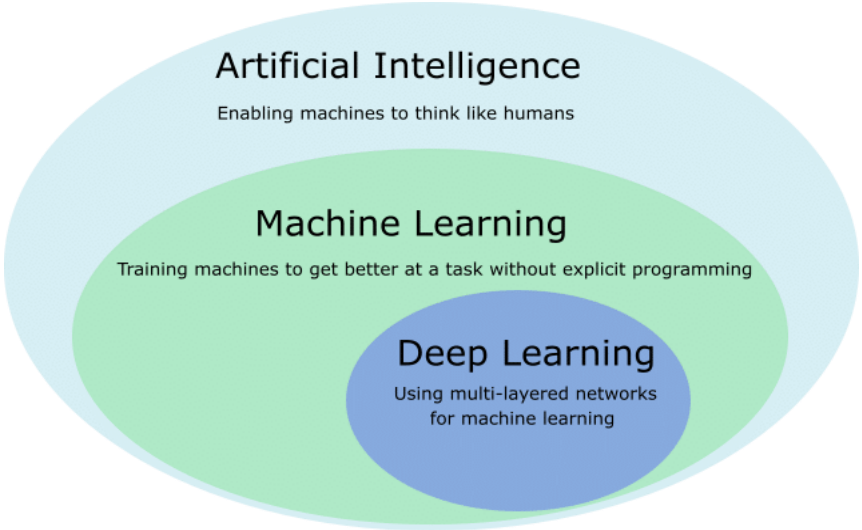


Εικόνα 1.4: Νευρωνικό δίκτυο τεσσάρων επιπέδων

Η βαθιά μάθηση χρησιμοποιείται για την αυτοματοποίηση πολύπλοκων διεργασιών πετυχαίνοντας συνήθως μεγαλύτερη ακρίβεια από αυτή που θα είχε ένα μοντέλο μηχανικής μάθησης αν του είχε ανατεθεί το ίδιο πρόβλημα. Ωστόσο για την εκπαίδευση ενός μοντέλου βαθιάς μάθησης χρειάζεται ένα τεράστιο σύνολο δεδομένων και πολύ μεγάλη υπολογιστική ισχύ. Η βαθιά μάθηση είναι μία μέθοδος που συνεχώς εξελλίσεται και βρίσκει ήδη εφαρμογή σε πολλούς τομείς όπως η ιατρική, η φαρμακευτική, η διοίκηση επιχειρήσεων.

1.2 Διαφορές μεταξύ μηχανικής και βαθιάς μάθησης

Συμπεραίνεται λοιπόν ότι ο κλάδος της τεχνητής νοημοσύνης εμπεριέχει τις μεθόδους της μηχανικής και βαθιάς μάθησης, με την δεύτερη να είναι υποκατηγορία της πρώτης όπως φαίνεται στην εικόνα 1.5.



Εικόνα 1.5: Έννοιες που εμπεριέχονται στον κλάδο της τεχνητής νοημοσύνης

Η κύρια διαφορά μεταξύ των δύο αυτών μεθόδων είναι ο τρόπος παρουσίασης των δεδομένων. Οι αλγόριθμοι μηχανικής μάθησης σχεδόν πάντα απαιτούν την χρήση δομημένων δεδομένων, δηλαδή δεδομένων που έχουν κατηγοριοποιηθεί προηγουμένως, ενώ αντίθετα τα νευρωνικά δίκτυα στην βαθιά μάθηση στηρίζονται στον αριθμό των στρωμάτων που αυτά αποτελούνται και η επεξεργασία και ταξινόμηση των τροφοδοτούμενων δεδομένων γίνεται αυτόματα. Ωστόσο και στις δύο περιπτώσεις η ποιότητα των δεδομένων παίζει καθοριστικό ρόλο στην ακρίβεια του τελικού αποτελέσματος. Μία άλλη διαφορά είναι ότι επειδή ακριβώς ένα μοντέλο βαθιάς μάθησης είναι σε θέση να εκτελεί πιο απαιτητικές διεργασίες, ο χρόνος εκπαίδευσής του είναι πολύ περισσότερος από τον χρόνο εκπαίδευσης ενός απλού μοντέλου μηχανικής μάθησης. Για την εκπαίδευση ενός νευρωνικού δικτύου χρειάζεται πολύ μεγαλύτερη υπολογιστική ισχύς από αυτή που χρειάζεται για την εκπαίδευση ενός αλγόριθμου μηχανικής μάθησης. Τα νευρωνικά δίκτυα χρησιμοποιούνται για την επίλυση αρκετά απαιτητικών διεργασιών και είναι απαραίτητη η χρήση ενός μεγάλου πλήθους δεδομένων, ενώ στην περίπτωση της μηχανικής μάθησης η εκπαίδευση του μοντέλου μπορεί να γίνει και σε ένα πιο μικρό πλήθος δειγμάτων. Οι διαφορές μεταξύ μηχανικής και βαθιάς μάθησης συνοψίζονται στον παρακάτω πίνακα:

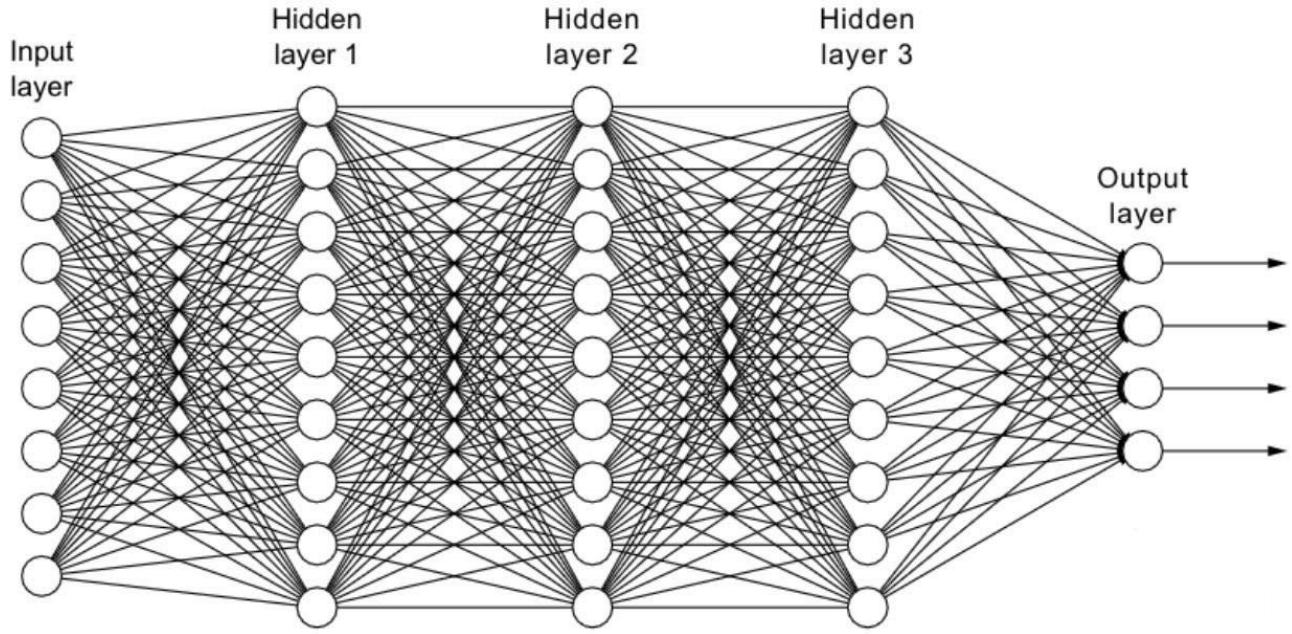
	ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ	**ΒΑΘΙΑ ΜΑΘΗΣΗ**
Διαχωρισμός και συσχέτιση μεταξύ των δεδομένων	Απαιτείται ώστε να είναι δυνατή η κατανόηση των χαρακτηριστικών τους	Δεν είναι απαραίτητο. Η κατανόηση των χαρακτηριστικών γίνεται αυτόματα
Πλήθος δεδομένων	Αποδίδει καλύτερα σε μικρά και μεσαία σύνολα δεδομένων	Αποδίδει καλύτερα σε μεγάλα σύνολα δεδομένων
Υπολογιστική ισχύ	Δεν απαιτείται μεγάλη	Απαιτείται μεγάλη
Χρόνος εκπαίδευσης	Μπορεί να διαφέρει από λεπτά σε ώρες	Μπορεί να διαφέρει από μέρες σε εβδομάδες

Πίνακας 1.1

Και οι δύο μέθοδοι χρησιμοποιούνται ευρέως σε βιομηχανίες και επιχειρήσεις για την αυτοματοποίηση πολύπλοκων και απαιτητικών διεργασιών, με την κάθε μέθοδο να έχει τα πλεονεκτήματα και τα μειονεκτήματα της. Ωστόσο ένας από τους πιο σημαντικούς αλγόριθμους που χρησιμοποιείται και στις δύο περιπτώσεις, με μικρές παραλλαγές, είναι τα τεχνητά νευρωνικά δίκτυα. Ο λόγος είναι, διότι ανεξάρτητα την φύση του προβλήματος, στις περισσότερες περιπτώσεις μπορεί να επιλυθεί με την μέθοδο των νευρωνικών δικτύων και αυτό είναι που τα καθιστά και τόσο ευέλικτα.

2. ΑΝΑΛΥΣΗ ΛΕΙΤΟΥΡΓΙΩΝ ΚΑΙ ΜΕΡΩΝ ΕΝΟΣ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ

Ο όρος νευρωνικά δίκτυα εμφανίστηκε για πρώτη φορά το 1940, αλλά ο τρόπος χρήσης και εκπαίδευσης τους ήταν ακόμα τότε άγνωστος. Δεν ήταν μέχρι και είκοσι χρόνια αργότερα που επιννοήθηκε η έννοια της οπισθοδιάδοσης, παρόλα αυτά τα νευρωνικά δίκτυα άρχισαν να βρίσκουν πρακτική εφαρμογή από το 2010 και ύστερα. Από τότε έχει πραγματοποιηθεί ένα μεγάλο πλήθος ερευνών πάνω στα νευρωνικά δίκτυα και έχουν καταστεί ένας από τους πιο δημοφιλείς αλγόριθμους πάνω στην μηχανική μάθηση εξαιτίας της ικανότητας τους να είναι σε θέση να επιλύουν προβλήματα που φαινόντουσαν πριν αδύνατο να λυθούν. Παραδείγματα περιπτώσεων που εφαρμόζεται η μέθοδος των νευρωνικών δικτύων είναι η αναγνώριση αντικειμένων, τα αυτόνομα αυτοκίνητα, η ανίχνευση καρκινικών κυττάρων στα αρχικά στάδια και ο υπολογισμός κινδύνου για μια επιχείρηση. Η δομή τους είναι εμπνευσμένη από τα βιολογικά νευρωνικά δίκτυα και παρόλο που ο τρόπος εκπαίδευσης τους διαφέρει αρκετά από αυτόν ενός ανθρώπου η βασική αρχή είναι η ίδια. Ένας τεχνητός νευρώνας από μόνος του δεν προσφέρει πολλά, αλλά όταν συνδυαστεί με εκατοντάδες ή και χιλιάδες άλλους νευρώνες, η διασυνδεσιμότητα αυτή φέρνει αποτελέσματα τα οποία συνήθως ξεπερνούν σε απόδοση οποιοδήποτε άλλον αλγόριθμο μηχανικής μάθησης.



Εικόνα 2.1: Νευρωνικό δίκτυο βαθιάς μάθησης

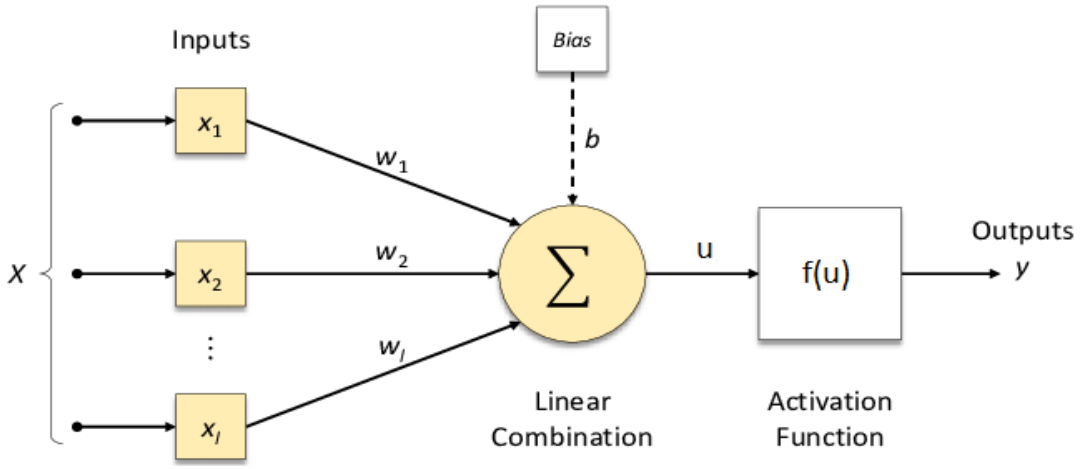
Το πρώτο επίπεδο ενός νευρωνικού δικτύου αποτελεί συνήθως την είσοδο τροφοδοσίας δεδομένων και ονομάζεται στρώμα εισόδου. Στη συνέχεια τα δεδομένα περνάνε και στα επόμενα στρώματα τα οποία ονομάζονται κρυφά στρώματα. Ένα νευρωνικό δίκτυο το

οποίο αποτελείται από δύο ή περισσότερα κρυφά στρώματα ονομάζεται βαθύ νευρωνικό δίκτυο. Τέτοιου είδους νευρωνικά δίκτυα χρησιμοποιούνται στην βαθιά μάθηση, ενώ στην περίπτωση της μηχανικής μάθησης χρησιμοποιούνται νευρωνικά δίκτυα απλούστερης δομής. Τα δεδομένα αφού περάσουν από όλα τα κρυφά στρώματα, καταλήγουν στο τελευταίο επίπεδο το οποίο είναι υπεύθυνο για την εξαγωγή του τελικού αποτελέσματος. Ένα δίκτυο τεχνητών νευρώνων μπορεί να έχει διάφορες μορφές και ανάλογα την φύση του προβλήματος χρησιμοποιείται η κατάλληλη αρχιτεκτονική.

2.1 Αρχιτεκτονικές νευρωνικών δικτύων

2.1.1 Μοντέλο Perceptron

Ένα νευρωνικό δίκτυο στην απλούστερη του μορφή αποτελείται από έναν μόνο νευρώνα. Σε αυτή εδώ την περίπτωση η χρήση της λέξης «δίκτυο» γίνεται καταχρηστικά αφού δεν υπάρχει σύνδεση μεταξύ πολλών νευρώνων. Ένα τέτοιο νευρωνικό δίκτυο ονομάζεται δίκτυο Perceptron και εφευρέθηκε το 1958 από τον Frank Rosenblatt.



Εικόνα 2.2: Αρχιτεκτονική του μοντέλου perceptron

Στο μοντέλο Perceptron οι είσοδοι, από τις οποίες γίνεται η τροφοδοσία των δεδομένων, πριν καταλήξουν στον νευρώνα πολλαπλασιάζονται με κάποιες παραμέτρους οι οποίες ονομάζονται συναπτικά βάρη (weights) και τα οποία συμβολίζονται με w. Ο ρόλος των συναπτικών βαρών είναι να πληροφορούν το δίκτυο πόσο σημαντική είναι η πληροφορία για την εξαγωγή του τελικού αποτελέσματος. Όσο μικρότερη είναι η τιμή ενός συναπτικού βάρους τόσο λιγότερο αντίκτυπο θα έχει η συγκεκριμένη είσοδος στην έξοδο. Αφού πραγματοποιηθεί ο πολλαπλασιασμός της κάθε εισόδου με το αντίστοιχο συναπτικό βάρος ο νευρώνας υπολογίζει το άθροισμα των δεδομένων που εισέρχονται σε αυτόν, συμπεριλαμβάνοντας μία επιπλέον παράμετρο η οποία ονομάζεται πόλωση (bias) και συμβολίζεται με b. Η μαθηματική έκφραση για την συνάρτηση μεταφοράς που υλοποιείται από τον νευρώνα είναι:

$$u = \sum_{i=1}^{n} \text{wi xi} - \text{b}$$

Ο όρος u ονομάζεται διέργεση του νευρώνα και είναι θετικός αν το άθροισμα $\sum_{i=1}^{n} \text{wi xi}$ είναι μεγαλύτερο της τιμής της πόλωσης b, δηλαδή:

$$u > 0, \quad \alpha\nu \ \sum_{i=1}^{n} \text{wi xi} > \text{b}$$

$$\text{u} = 0, \quad \alpha\nu \ \sum_{i=1}^{n} \text{wi xi} = \text{b}$$

$$u < 0, \quad \alpha\nu \ \sum_{i=1}^{n} \text{wi xi} < \text{b}$$

Η διέργεση του νευρώνα u στη συνέχεια οδηγείται σαν είσοδος στην συνάρτηση ενεργοποίησης (activation function) f(u), η οποία δίνει την έξοδο y του νευρώνα.

Η συνάρτηση ενεργοποίησης ειδικά σε ένα μοντέλο perceptron μπορεί να πάρει μία από τις παρακάτω μορφές:

$$f(u) = \begin{cases} 1, & \alpha\nu \ u > 0 \\ 0, & \alpha\nu \ u \leq 0 \end{cases}$$

ή

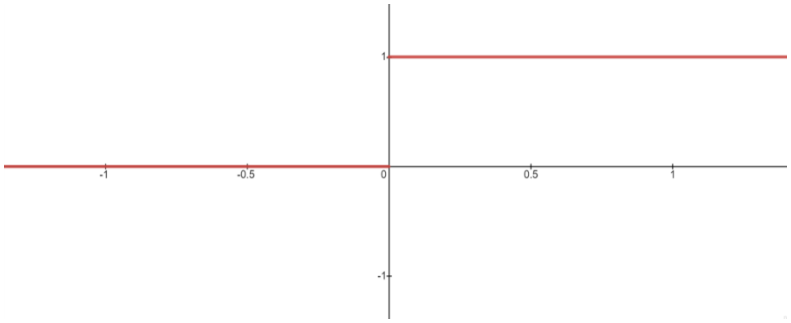
$$f(u) = \begin{cases} 1, & \alpha\nu \ u > 0 \\ -1, & \alpha\nu \ u \leq 0 \end{cases}$$

Επομένως συμπεραίνεται ότι αντίθετα με την διέργεση u του νευρώνα η οποία είναι μία γραμμική σχέση, η συνάρτηση ενεργοποίησης συνήθως είναι μία μη γραμμική συνάρτηση. Στα περισσότερα προβλήματα που προσπαθούν να επιλυθούν με την χρήση των νευρωνικών δικτύων δεν υπάρχει μία γραμμική σχέση μεταξύ δεδομένων και αποτελέσματος, επομένως η χρήση μίας μη γραμμικής μαθηματικής συνάρτησης είναι απαραίτητη. Παρακάτω αναφέρονται μερικές από τις πιο σημαντικές συναρτήσεις ενεργοποίησης που χρησιμοποιούνται σε ένα τεχνητό νευρωνικό δίκτυο:

ΒΗΜΑΤΙΚΗ ΣΥΝΑΡΤΗΣΗ ΕΝΕΡΓΟΠΟΙΗΣΗΣ (STEP ACTIVATION FUNCTION)

Η βηματική συνάρτηση αποτελεί μία από τις πιο απλές μορφές που μπορεί να έχει μία συνάρτηση ενεργοποίησης. Σκοπός αυτής της συνάρτησης είναι να καθορίζει αν ένας νευρώνας θα ενεργοποιηθεί ή θα μείνει ανενεργός. Αν το γινόμενο των εισόδων μαζί με τα αντίστοιχα συναπτικά βάρη τους το οποίο αποτελεί την διέργεση ενός νευρώνα, είναι μεγαλύτερο της πόλωσης b, ή αλλιώς με μαθηματικούς όρους $\sum_{i=1}^{n} \mathrm{wi}\ \mathrm{xi} > b$, τότε ο νευρώνας θα ενεργοποιηθεί και η βηματική συνάρτηση θα δώσει ως αποτέλεσμα 1. Σε διαφορετική περίπτωση το αποτέλεσμα είναι 0.

$$f(x) = \begin{cases} 1, & \alpha\nu\ \ x > 0 \\ 0, & \alpha\nu\ \ x \leq 0 \end{cases}$$

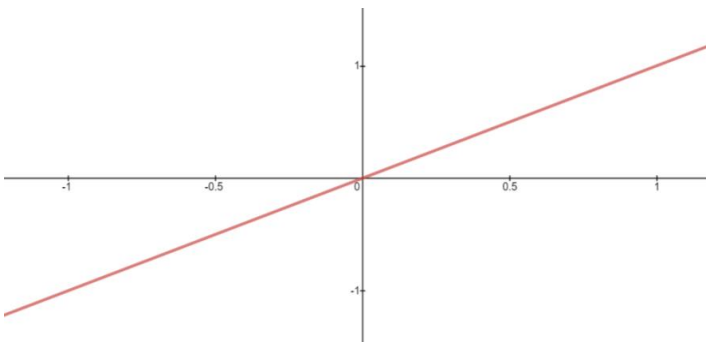


Αυτή η συγκεκριμένη συνάρτηση ενεργοποίησης χρησιμοποιείται στα κρυφά στρώματα ενός νευρωνικού δικτύου, αλλά σήμερα έχει αντικατασταθεί από άλλες συναρτήσεις.

ΓΡΑΜΜΙΚΗ ΣΥΝΑΡΤΗΣΗ ΕΝΕΡΓΟΠΟΙΗΣΗΣ (LINEAR ACTIVATION FUNCTION)

Η συγκεκριμένη αυτή συνάρτηση εκφράζεται μαθηματικά με την εξίσωση μιας ευθείας η οποία διέρχεται από την αρχή των αξόνων. Σε αυτή την περίπτωση οι τιμές τις εξόδου θα είναι ίσες με τις τιμές της εισόδου.

$$f(x) = x$$

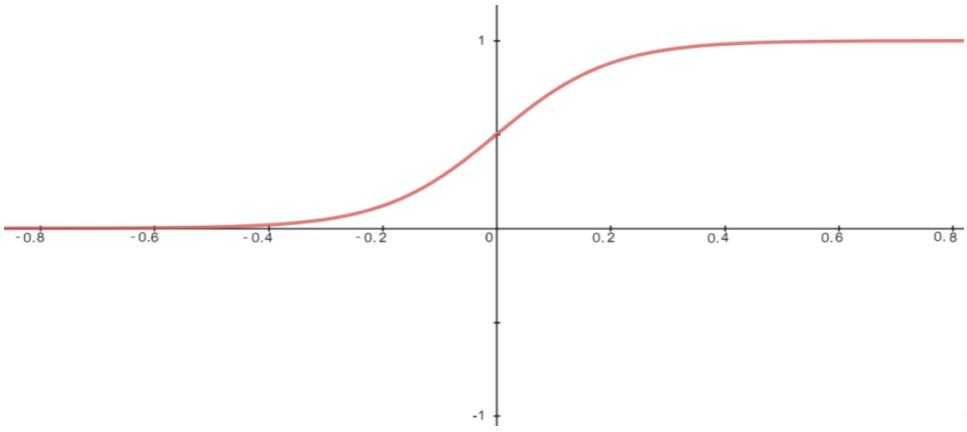


Η γραμμική συνάρτηση ενεργοποίησης συνήθως εφαρμόζεται στο τελευταίο στρώμα εξόδου στην περίπτωση ενός μοντέλου λογιστικής παλινδρόμησης (linear regression).

ΣΙΓΜΟΕΙΔΗΣ ΣΥΝΑΡΤΗΣΗ ΕΝΕΡΓΟΠΟΙΗΣΗΣ (SIGMOID ACTIVATION FUNCTION)

Το πρόβλημα που δημιουργείται όταν γίνεται χρήση της βηματικής συνάρτησης είναι ότι στην περίπτωση εκπαίδευσης ενός νευρωνικού δικτύου με την βοήθεια ενός αλγόριθμου βελτιστοποίησης, δεν είναι αρκετά ξεκάθαρες οι επιρροές που μπορεί να υπάρξουν στην έξοδο των νευρώνων από την ρύθμιση των συναπτικών βαρών. Είναι δύσκολο να προσδιοριστεί η απόσταση που είχε η αρνητική τιμή u της διέργεσης ενός νευρώνα από το μηδέν, διότι ανεξάρτητα από την απόσταση, με την εφαρμογή της βηματικής συνάρτησης στην διέργεση του νευρώνα το αποτέλεσμα θα είναι 0. Είναι προτιμότερο λοιπόν όταν γίνεται εκπαίδευση του νευρωνικού δικτύου να χρησιμοποιούνται συναρτήσεις ενεργοποίησης οι οποίες δίνουν περισσότερες πληροφορίες για την διέργεση του νευρώνα. Μία τέτοια συνάρτηση είναι η σιγμοειδής συνάρτηση ενεργοποίησης η οποία εκφράζεται μαθηματικά και γραφικά ως εξής:

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

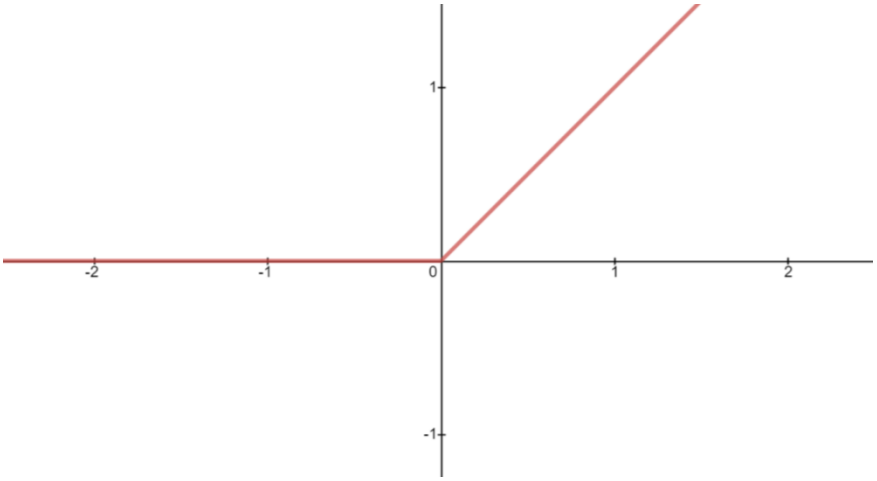


Αυτή η συνάρτηση ενεργοποίησης επιστρέφει την τιμή 0 για πολύ μεγάλες αρνητικές τιμές, την τιμή 0.5 όταν δέχεται σαν είσοδο την τιμή 0 και τέλος επιστρέφει την τιμή 1 για πολύ μεγάλες θετικές τιμές. Με την έξοδο της σιγμοειδής συνάρτησης να παίρνει τιμές από 0 έως 1 την καθιστά κατάλληλη για να χρησιμοποιείται ως συνάρτηση ενεργοποίησης στα βαθιά στρώματα ενός νευρωνικού δικτύου αν και τώρα έχει αντικασταθεί από μία άλλη συνάρτηση ενεργοποίησης που ονομάζεται συνάρτηση ράμπας. Είναι όμως κατάλληλη για προβλήματα ταξινόμησης και εφαρμόζεται ακόμα σε αυτές τις περιπτώσεις στο στρώμα εξόδου ενός νευρωνικού δικτύου.

ΣΥΝΑΡΤΗΣΗ ΡΑΜΠΑΣ (RECTIFIED LINEAR UNIT ACTIVATION FUNCTION ή ReLU)

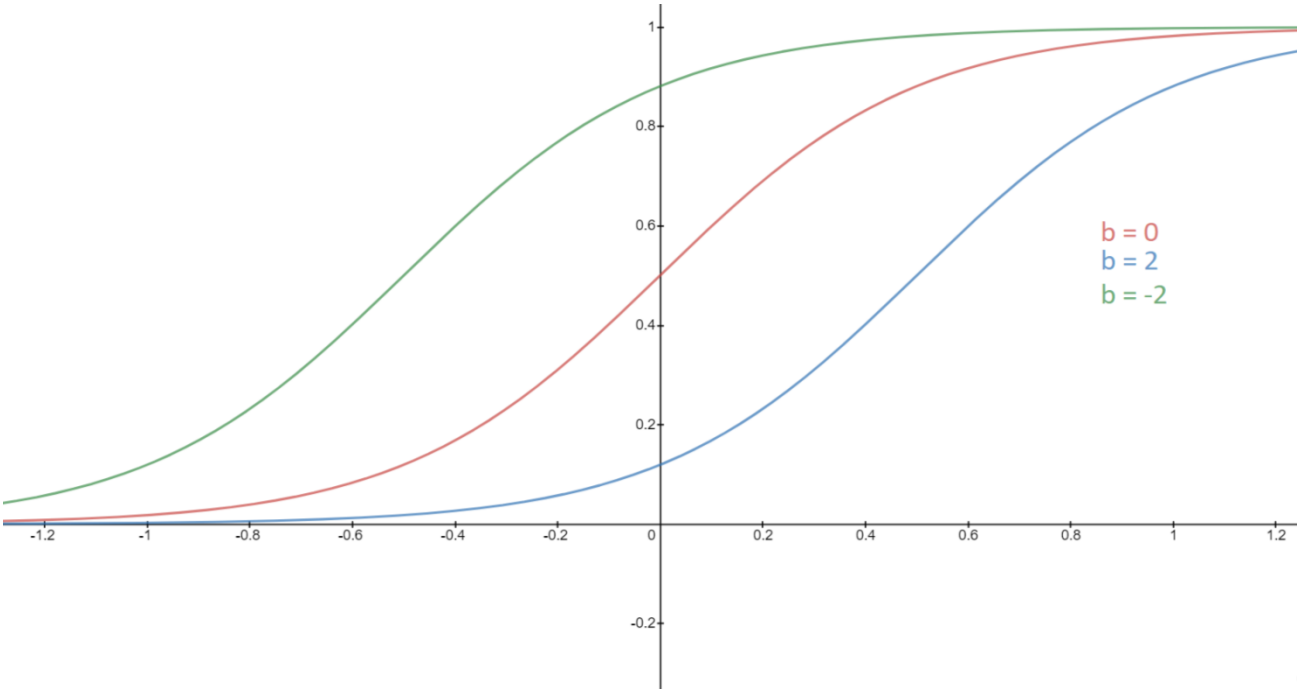
Η συγκεκριμένη συνάρτηση είναι απλούστερη από την σιγμοειδή συνάρτηση. Για τιμές οι οποίες είναι μεγαλύτερες του μηδενός η συνάρτηση ράμπας επιστρέφει σαν έξοδο μία τιμή που είναι ακριβώς ίση με αυτή της εισόδου, διαφορετικά επιστρέφει την τιμή 0.

$$f(x) = \begin{cases} \chi, & \alpha\nu \ x > 0 \\ 0, & \alpha\nu \ x \leq 0 \end{cases}$$



Αυτή η απλή αλλά ταυτόχρονα ισχυρή συνάρτηση ενεργοποίησης χρησιμοποιείται ευρέως στα νευρωνικά δίκτυα για πολλούς λόγους με τους κύριους να είναι η ταχύτητα και η αποτελεσματικότητα που προσφέρει. Τείνει στο να είναι μία γραμμική συνάρτηση ενεργοποίησης παραμένοντας όμως ταυτόχρονα μη γραμμική εξαιτίας της δυνατότητας να έχει ως έξοδο την τιμή 0 για τιμές εισόδου μικρότερες του μηδενός. Αυτός είναι και ο λόγος που είναι τόσο αποτελεσματική.

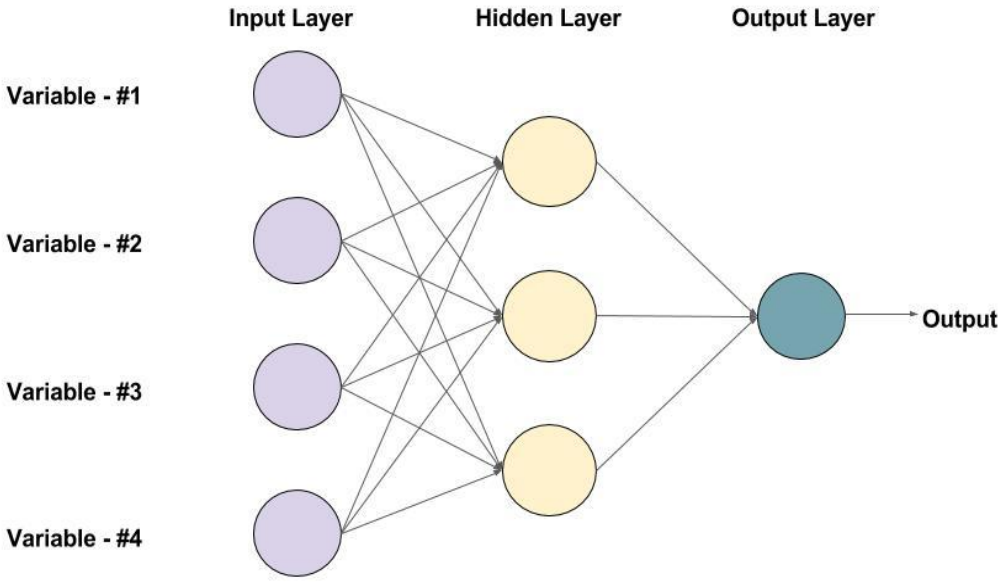
Η μετατόπιση δεξιά ή αριστερά μιας συνάρτησης ενεργοποίησης πραγματοποιείται με τη βοήθεια της παραμέτρου πόλωσης ή αλλιώς κατώφλι ενεργοποίησης b ενός τεχνητού νευρώνα. Αν και αρχικά η ιδιότητα της παραμέτρου αυτής μπορεί να μην φαίνεται σημαντική, είναι όμως ο λόγος που μπορεί να πραγματοποιηθεί σωστά η εκπαίδευση ενός νευρωνικού δικτύου.



Εικόνα 2.3: Μετατόπιση της σιγμοειδούς συνάρτησης με την βοήθεια της παραμέτρου πόλωσης b. Για θετικές τιμές η σιγμοειδής συνάρτηση ενεργοποίησης μετατοπίζεται δεξιά ενώ για αρνητικές αριστερά.

2.1.2 Δίκτυο Perceptron πολλών στρωμάτων (Multilayer Perceptron)

Ένα δίκτυο perceptron πολλών στρωμάτων, συχνά αποκαλείται και δίκτυο πρόσθιας τροφοδοσίας (feedforward neural network), είναι ένα νευρωνικό δίκτυο το οποίο αποτελείται από πολλά στρώματα με το κάθε στρώμα να έχει πάνω από ένα νευρώνα. Τα συναπτικά βάρη των νευρώνων έχουν διαφορετικές τιμές μεταξύ τους. Στόχος ενός τέτοιου δικτύου είναι να προσεγγίσει μία μαθηματική συνάρτηση η οποία θα περιγράφει συνήθως ένα μη γραμμικό πρόβλημα. Για παράδειγμα στην περίπτωση ενός προβλήματος ταξινόμησης τα συναπτικά βάρη του δικτύου θα προσαρμοστούν με την βοήθεια ενός αλγόριθμου με σκοπό την προσέγγιση του επιθυμητού αποτελέσματος. Έτσι η λειτουργία του δικτύου μπορεί να περιγράψει την συνάρτηση του προβλήματος με την μέγιστη δυνατή ακρίβεια (accuracy). Αυτού του είδους νευρωνικά δίκτυα χρησιμοποιούνται κυρίως στην εποπτευόμενη μηχανική μάθηση όπου το επιθυμητό αποτέλεσμα είναι ήδη γνωστό, αλλά βρίσκουν και άλλες εφαρμογές όπως είναι η αναγνώριση εικόνων και η επεξεργασία ομιλίας, με μερικές όμως παραλλαγές στην αρχιτεκτονική τους.



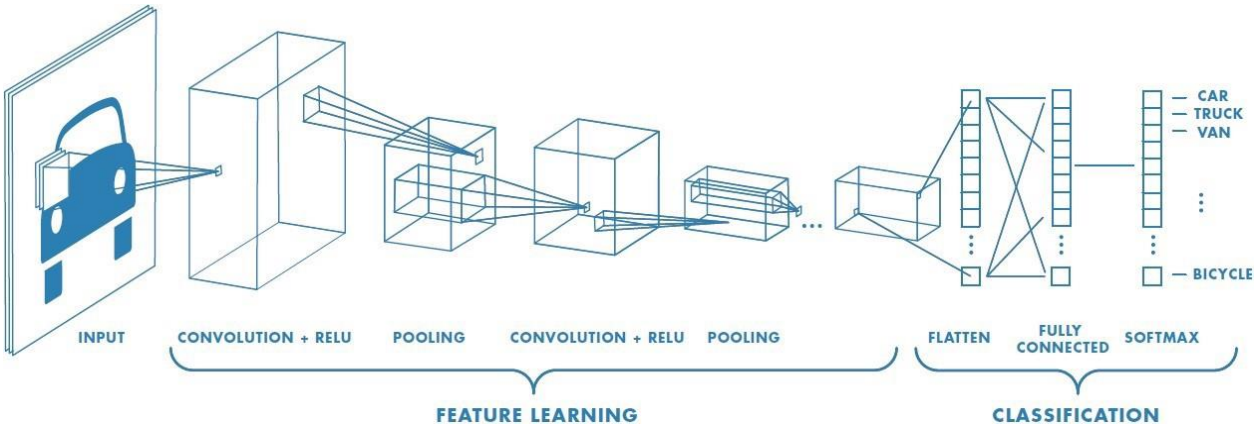
Εικόνα 2.4: Ένα νευρωνικό δίκτυο πρόσθιας τροφοδοσίας με ένα κρυφό στρώμα τριών νευρώνων

Η μορφή ενός δικτύου perceptron πολλών στρωμάτων αποτελείται από έναν συνήθως μεγάλο αριθμό νευρώνων οι οποίοι είναι ταξινομημένοι στα στρώματα του δικτύου. Το πρώτο επίπεδο αποτελεί την είσοδο των δεδομένων στο δίκτυο και ονομάζεται στρώμα εισόδου (input layer). Το τελευταίο επίπεδο αποτελεί την έξοδο του νευρωνικού δικτύου και ονομάζεται στρώμα εξόδου (output layer). Τα στρώματα μεταξύ των επιπέδων εισόδου και εξόδου ονομάζονται κρυφά στρώματα (hidden layers), διότι δεν υπάρχει πρόσβαση στο αποτέλεσμα των διεργασιών τους, και χρησιμοποιούνται για την αύξηση της μη γραμμικότητας του δικτύου. Ένα νευρωνικό δίκτυο μπορεί να περιέχει έναν οποιονδήποτε αριθμό κρυφών στρωμάτων, με το κάθε κρυφό στρώμα να μην περιέχει έναν συγκεκριμένο αριθμό νευρώνων. Αντιθέτως ο αριθμός των νευρώνων στα επίπεδα εισόδου και εξόδου εξαρτάται από τον αριθμό των δεδομένων και το αποτέλεσμα αντίστοιχα. Οι συναρτήσεις ενεργοποίησης που εφαρμόζονται στις εξόδους των νευρώνων είναι διαφορετικές για το κάθε επίπεδο. Σε ένα παράδειγμα ταξινόμησης δεδομένων συνήθως στα κρυφά στρώματα γίνεται χρήση της συνάρτησης ράμπας ενώ στο στρώμα εξόδου εφαρμόζεται η σιγμοειδής συνάρτηση.

Αυτά τα μοντέλα νευρωνικών δικτύων ονομάζονται δίκτυα πρόσθιας τροφοδοσίας διότι τα δεδομένα περνώντας από το επίπεδο εισόδου τροφοδοτούνται στα ενδιάμεσα στρώματα, όπου γίνεται και το σημαντικότερο κομμάτι της επεξεργασίας τους. Στη συνέχεια τα επεξεργασμένα αυτά δεδομένα καταλήγουν στο επίπεδο εξόδου του δικτύου. Ουσιαστικά η διαδρομή που ακολουθούν τα δεδομένα μέσα στο δίκτυο είναι μίας κατεύθυνσης και δεν υπάρχουν κάποιες συνδέσεις ανατροφοδότησης του δικτύου. Η εκπαίδευση αυτών των νευρωνικών δικτύων πραγματοποιείται με την βοήθεια ενός συνδυασμού τεχνικών μάθησης στις οποίες γίνεται αναφορά στη συνέχεια.

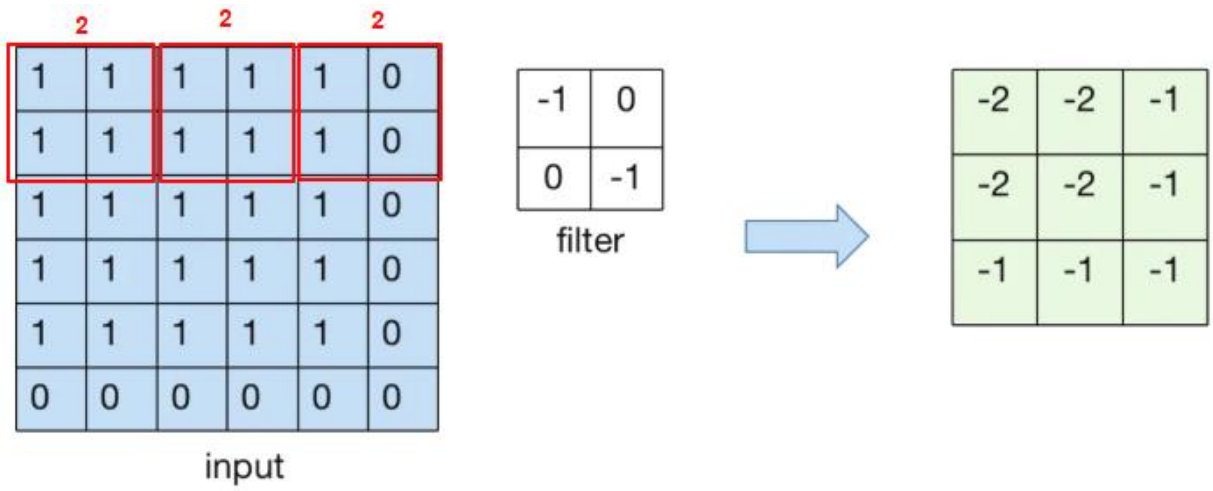
2.1.3 Συνελικτικά νευρωνικά δίκτυα (Convolutional neural networks)

Αυτή η κατηγορία νευρωνικών δικτύων χρησιμοποιείται ευρέως σε εφαρμογές υπολογιστικής όρασης με μερικές από αυτές να η ταξινόμηση εικόνων, ανίχνευση αντικειμένων και αναγνώριση προσώπων. Ένα συνελικτικό νευρωνικό δίκτυο δέχεται ως είσοδο μία εικόνα η οποία παριστάνεται ως μία σειρά εικονοστοιχείων (pixels), της οποίας το μέγεθος εξαρτάται από την ανάλυση της αρχικής εικόνας. Αυτή η σειρά εικονοστοιχείων παίρνει την μορφή ενός πίνακα με διαστάσεις ύψος × πλάτος × d (height × width × dimension), όπου το ύψος και το πλάτος είναι το μέγεθος της εικόνας, και η παράμετρος d αναπαριστά το χρώμα της εικόνας. Αφού το δίκτυο επεξεργαστεί την εικόνα, την ταξινομεί σε συγκεκριμένες κατηγορίες. Κάθε εικόνα που τροφοδοτείται στο δίκτυο περνάει από μία σειρά συνελικτικών στρωμάτων και φίλτρων και στη συνέχεια πραγματοποιείται μία διαδικασία που ονομάζεται συγκέντρωση (pooling). Τέλος τα χαρακτηριστικά που έχουν εξαχθεί οδηγούνται σε ένα νευρωνικό δίκτυο πρόσθιας τροφοδοσίας για ταξινόμηση.



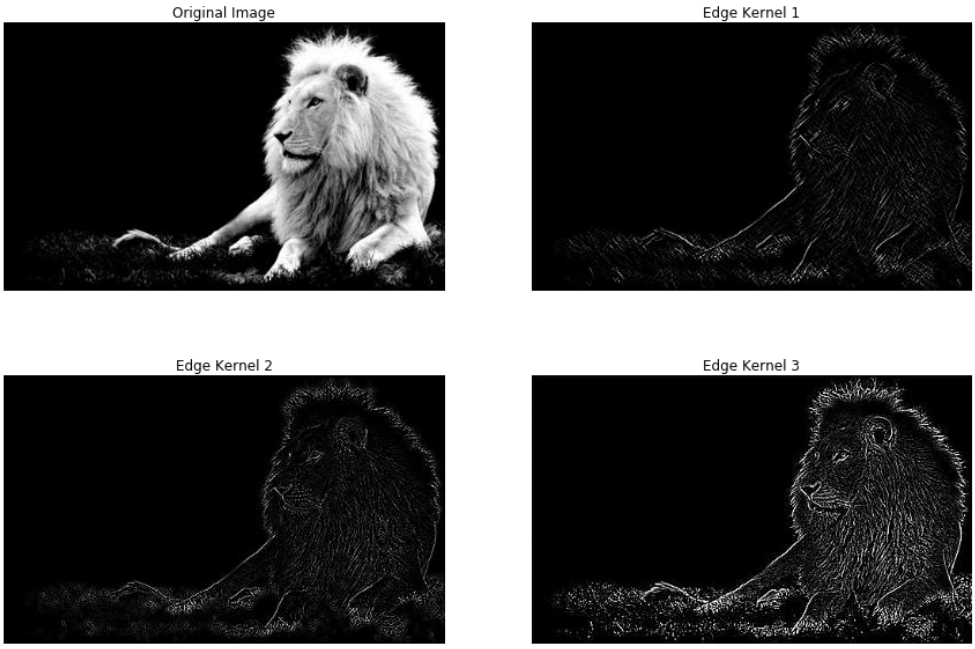
Εικόνα 2.5: Αρχιτεκτονική ενός συνελικτικού δικτύου

Το συνελικτικό στρώμα είναι το πρώτο επίπεδο που εξάγει χαρακτηριστικά από την τροφοδοτούμενη εικόνα. Αυτή η διαδικασία περιγράφεται με μία μαθηματική πράξη που ονομάζεται συνέλιξη η οποία παίρνει σαν είσοδο δύο πίνακες με τον πρώτο να απεικονίζει τα στοιχεία της εικόνας και τον δεύτερο να αποτελεί ένα φίλτρο (kernel), και επιστρέφει έναν νεό πίνακα συγκεκριμένων διαστάσεων. Για παράδειγμα θεωρώντας έναν πίνακα διαστάσεων 6 × 6, ο οποίος περιγράφει όπως προειπώθηκε τα εικονοστοιχεία μιας εικόνας, η εφαρμογή ενός φίλτρου 2 × 2 πάνω στον πίνακα θα γίνεται σταδιακά. Αρχικά θα επιλεχθεί ένα εσωτερικό τμήμα του πίνακα το οποίο θα έχει τις ίδιες διαστάσεις με αυτές του φίλτρου. Στη συνέχεια πραγματοποιείται η πράξη του πολλαπλασιασμού μεταξύ των στοιχείων του τμήματος του πίνακα και του φίλτρου που βρίσκονται στις αντίστοιχες θέσεις και τέλος τα γινόμενα που προκύπτουν θα προστεθούν μεταξύ τους. Η διαδικασία συνεχίζεται με την μετατόπιση του φίλτρου κάποιες θέσεις και ολοκληρώνεται όταν το φίλτρο εφαρμοστεί σε όλα τα τμήματα του πίνακα. Ο αριθμός των θέσεων μετατόπισης του φίλτρου ονομάζεται βήμα (stride).



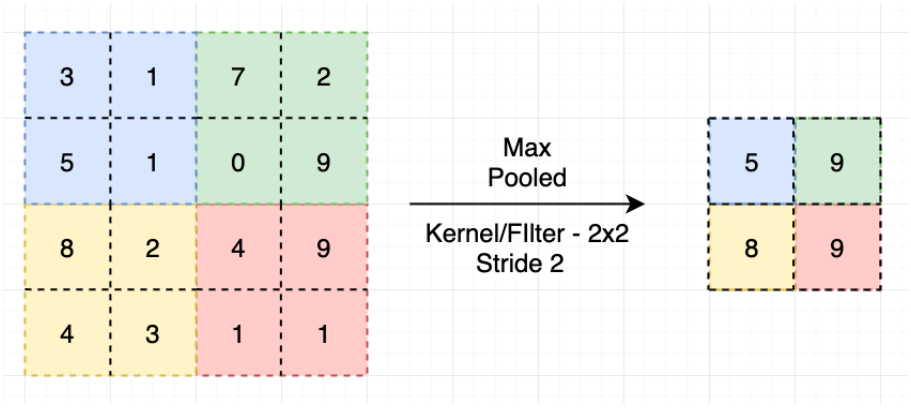
Εικόνα 2.6: Εφαρμογή ενός φίλτρου 2 × 2 σε έναν πίνακα 6 × 6 με βήμα 2

Στην περίπτωση που το φίλτρο δεν προσαρμόζεται κατάλληλα στον πίνακα πραγματοποιείται μία διαδικασία κατά την οποία συμπληρώνεται ο πίνακας με μηδενικά στοιχεία ώστε να επιτευχθεί η κατάλληλη εφαρμογή του φίλτρου. Αυτή η διαδικασία ονομάζεται zero padding. Μία άλλη επιλογή είναι η απόρριψη των τμημάτων του πίνακα στα οποία το φίλτρο δεν εφαρμόστηκε κρατώντας έτσι μόνο τα βασικότερα στοιχεία του πίνακα. Αυτή η μέθοδος ονομάζεται valid padding. Με την εφαρμογή διαφόρων φίλτρων σε μία εικόνα είναι δυνατή η η ανίχνευση βασικών χαρακτηριστικών της.



Εικόνα 2.7: Εξαγωγή χαρακτηριστικών της εικόνας από τα συνελικτικά στρώματα. Αρχικά στο πρώτο στρώμα (Edge Kernel 1) εξάγονται απλά χαρακτηριστικά και στη συνέχεια καθώς η εικόνα τροφοδοτείται στα επόμενα επίπεδα πραγματοποιείται ανίχνευση πιο πολύπλοκων χαρακτηριστικών

Το αποτέλεσμα του συνελικτικού στρώματος θα τροφοδοτηθεί σαν είσοδος στο επόμενο επίπεδο του δικτύου που ονομάζεται στρώμα συγκέντρωσης. Το στρώμα συγκέντρωσης είναι υπεύθυνο για την μείωση των χαρακτηριστικών της εικόνας στην περίπτωση που αυτή είναι μεγάλων διαστάσεων. Η διαδικασία αυτή πραγματοποιείται με την μέθοδο της διατμηματικής συγκέντρωσης (spatial pooling) κατά την οποία με την βοήθεια εφαρμογής ενός άλλου φίλτρου μειώνονται οι διαστάσεις του πίνακα που τροφοδοτήθηκε ως είσοδος, διατηρώντας όμως παράλληλα τις απαραίτητες πληροφορίες της εικόνας. Η διατμηματική συγκέντρωση μπορεί να είναι τριών ειδών: μέγιστη συγκέντρωση (max pooling), μέση συγκέντρωση (average pooling) και αθροιστική συγκέντρωση (sum pooling). Στην μέγιστη συγκέντρωση επιλέγεται το μεγαλύτερο στοιχείο του τμήματος του πίνακα όπου έχει εφαρμοστεί το φίλτρο. Για παράδειγμα στην περίπτωση που έχει εφαρμοστεί σε ένα τμήμα ενός πίνακα 4 × 4 ένα φίλτρο 2 × 2 και τα στοιχεία που υπάρχουν είναι οι αριθμοί [3, 1, 5, 1], με την μέθοδο max pooling θα επιλεχθεί το στοιχείο με την μεγαλύτερη τιμή, δηλαδή ο αριθμός 5.



Εικόνα 2.8: Εφαρμογή της μεθόδου max pooling σε έναν πίνακα 4 × 4 με φίλτρο 2 × 2 και βήμα 2

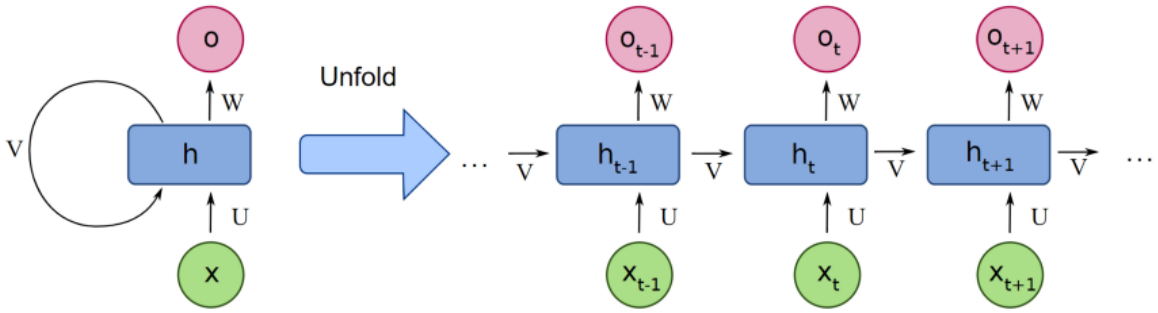
Στην περίπτωση της μέσης συγκέντρωσης υπολογίζεται η μέση τιμή των στοιχείων του τμήματος του πίνακα όπου γίνεται η εφαρμογή του φίλτρου ενώ η μέθοδος της αθροιστικής συγκέντρωσης έχει ως αποτέλεσμα το άθροισμα των στοιχείων του τμήματος.

Αφού γίνει στα πρώτα στρώματα του δικτύου η εξαγωγή βασικών χαρακτηριστικών της εικόνας, για παράδειγμα τέτοια χαρακτηριστικά μπορεί να είναι ευθείες γραμμές, γωνίες κ.τ.λ, τα επόμενα επίπεδα είναι υπεύθυνα για την ανίχνευση πιο πολύπλοκων χαρακτηριστικών. Στη συνέχεια αυτά τα χαρακτηριστικά τροφοδοτούνται ως δεδομένα σε ένα νευρωνικό δίκτυο πρόσθιας τροφοδοσίας το οποίο με την βοήθεια μιας συνάρτησης ενεργοποίησης στο στρώμα εξόδου του πραγματοποιεί ταξινόμηση των δεδομένων.

2.1.4 Αναδρομικά νευρωνικά δίκτυα (Recurrent neural networks)

Ένα αναδρομικό νευρωνικό δίκτυο είναι μία κατηγορία νευρωνικού δικτύου που χρησιμοποιεί διαδοχικά δεδομένα (sequential data). Διαδοχικά δεδομένα καλείται μία ακολουθία εισόδων στην οποία τα δεδομένα εξαρτώνται μεταξύ τους. Για παράδειγμα μία ακολουθία δεδομένων μπορεί να είναι οι προτάσεις ενός κειμένου οι οποίες εξαρτώνται από το περιεχόμενο προηγούμενων προτάσεων. Αυτός ο αλγόριθμος βαθιάς μάθησης χρησιμοποιείται ευρέως σε εφαρμογές όπως μετάφραση κειμένων, επεξεργασία ομιλίας, αναγνώριση ομιλίας, φωνητική αναζήτηση και σε εικονικούς βοηθούς (virtual assistants). Όπως στα συνελικτικά και στα πρόσθιας τροφοδοσίας δίκτυα έτσι και στην περίπτωση των αναδρομικών νευρωνικών δικτύων χρησιμοποιούνται δεδομένα για την εκπαίδευση τους. Κύριο χαρακτηριστικό τους που τα διαχωρίζει από τις άλλες κατηγορίες νευρωνικών δικτύων είναι η «μνήμη» που διαθέτουν καθώς χρησιμοποιούν προηγούμενα δεδομένα που είχαν τροφοδοτηθεί στην είσοδο τους, για την επεξεργασία τωρινών δεδομένων.

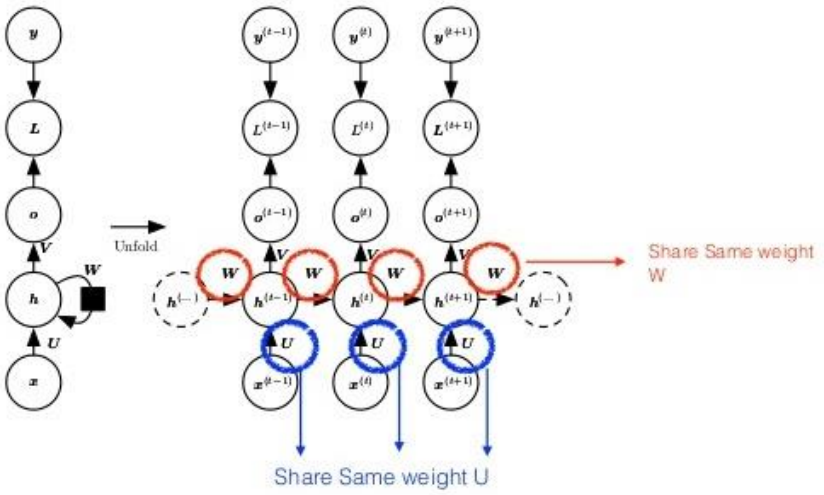
Στις κλασσικές μορφές νευρωνικών δικτύων τα δεδομένα εισόδου και εξόδου είναι τελείως ανεξάρτητα μεταξύ τους ενώ η έξοδος στα αναδρομικά νευρωνικά δίκτυα εξαρτάται από προγενέστερα δεδομένα που υπάρχουν στην ακολουθία εισόδων. Επομένως η έξοδος δεν επηρεάζεται μόνο από τα συναπτικά βάρη του δικτύου αλλά και από την δυνατότητα των κόμβων να είναι σε θέση να ανατροφοδοτούν την είσοδο άλλων νευρώνων ή ακόμα και των ίδιων, επιτρέποντας έτσι και την αναδρομική ροή πληροφορίας μέσα στο δίκτυο. Συνεπώς η ίδια είσοδος μπορεί να παράξει διαφορετικά αποτελέσματα ανάλογα με τα δεδομένα που εισήχθησαν προηγουμένως.



Εικόνα 2.9: Αριστερά απεικονίζεται η δομή ενός αναδρομικού νευρωνικού δικτύου με την έξοδο w να αυτο-ανατροφοδοτείται στον κόμβο h. Δεξιά είναι οι διάφορες καταστάσεις του δικτύου κατά την διαδικασία της αυτο-ανατροφοδότησης του κόμβου h

Ένα ακόμα χαρακτηριστικό των αναδρομικών νευρωνικών δικτύων είναι ότι οι τιμές των συναπτικών βαρών του κάθε νευρώνα είναι κοινές για όλους τους νευρώνες που ανήκουν στο ίδιο στρώμα σε αντίθεση με τα νευρωνικά δίκτυα πρόσθιας τροφοδοσίας που όλες οι παράμετροι έχουν διαφορετικές τιμές μεταξύ τους. Ωστόσο κατά την εκπαίδευση τους όλα τα συναπτικά βάρη των στρωμάτων προσαρμόζονται κατάλληλα για την επίτευξη μεγαλύτερης

ακρίβειας. Με την ανάθεση κοινών συναπτικών βαρών στο κάθε στρώμα η ευελιξία του δικτύου μειώνεται, όμως αποκτάται η δυνατότητα της τροφοδοσίας ακολουθιών δεδομένων των οποίων το μέγεθος δεν είναι απαραίτητο να είναι γνωστό καθώς όλοι οι νευρώνες του κάθε επιπέδου θα λάβουν τις ίδιες παραμέτρους. Αυτό το χαρακτηριστικό καθιστά τα αναδρομικά νευρωνικά δίκτυα κατάλληλα να εξάγουν σχέσεις μεταξύ των δεδομένων μιας ακολουθίας εισόδων.



Εικόνα 2.10: Κοινά συναπτικά βάρη στο κάθε στρώμα ενός αναδρομικού νευρωνικού δικτύου

2.2 Υπολογισμός σφάλματος ενός νευρωνικού δικτύου με την συνάρτηση κόστους

Στόχος ενός νευρωνικού δικτύου είναι να αντιστοιχίσει ένα σύνολο εισόδων σε μία σειρά εξόδων χρησιμοποιώντας τις παραμέτρους μεταξύ των διασυνδέσεων των νευρώνων. Με ένα μοντέλο του οποίου τα συναπτικά βάρη έχουν αρχικοποιηθεί αυθαίρετα ή με πιο σοφιστικές προσεγγίσεις, σκοπός είναι η εκπαίδευση του με την βοήθεια δεδομένων. Για να επιτευχθεί αυτός ο σκοπός προσαρμόζονται κατάλληλα τα συναπτικά βάρη του μοντέλου ώστε να αυξηθεί η ακρίβεια του στις προβλέψεις του. Ο υπολογισμός των τιμών που θα πρέπει να λάβουν τα συναπτικά βάρη για την μεγιστοποίηση της ακρίβειας του δικτύου δεν είναι εφικτός χωρίς τη βοήθεια κάποιου αλγόριθμου, καθώς το σύνολο των παραμέτρων ενός νευρωνικού δικτύου είναι πολύ μεγάλο. Αντί αυτού το πρόβλημα της εκπαίδευσης αντιμετωπίζεται ως ένα πρόβλημα βελτιστοποίησης. Αρχικά υπολογίζεται το σφάλμα του δικτύου με την συνάρτηση κόστους (cost function) ή συνάρτηση ζημίας (loss function) όπως αλλιώς αναφέρεται. Το σφάλμα ενός δικτύου υπολογίζεται συκρίνοντας το αποτέλεσμα του με την επιθυμητή έξοδο. Με άλλα λόγια το αποτέλεσμα της συνάρτησης κόστους (loss) είναι ένας αριθμός ο οποίος δείχνει κατά πόσο το μοντέλο ήταν λάθος στις προβλέψεις του. Στη συνέχεια για να επιτευχθεί η μεγαλύτερη δυνατή ακρίβεια του μοντέλου, η συνάρτηση κόστους θα πρέπει να ελαχιστοποιηθεί, ιδανικά να μηδενιστεί.

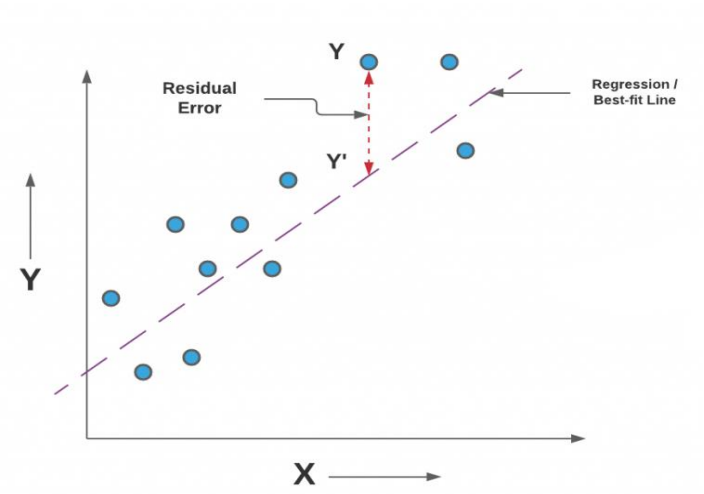
Δεν υπάρχει μία συνάρτηση κόστους η οποία να είναι κατάλληλη για όλες τις περιπτώσεις. Η επιλογή της γίνεται ανάλογα με το είδος του προβλήματος που πρέπει να επιλυθεί και καθορίζεται επίσης και από άλλους παράγοντες. Επιπλέον, εφόσον η συνάρτηση κόστους εξαρτάται από το τελευταίο στρώμα του νευρωνικού δικτύου, η διαμόρφωση του επιπέδου εξόδου πρέπει να είναι ανάλογη αυτής που θα επιλεχθεί. Υπάρχουν δύο βασικές κατηγορίες συναρτήσεων κόστους οι οποίες είναι οι συναρτήσεις κόστους ταξινόμησης και οι συναρτήσεις κόστους παλινδρόμησης, με την πρώτη να διακρίνεται σε δυαδική ταξινόμηση και σε ταξινόμηση πολλαπλών κλάσεων. Στην περίπτωση των συναρτήσεων κόστους ταξινόμησης στόχος είναι η πρόβλεψη του αποτελέσματος από διαφορετικές τιμές ομάδων. Για παράδειγμα εάν γίνεται χρήση ενός συνόλου δεδομένων το οποίο αποτελείται από εικόνες αριθμών οι οποίοι έχουν έυρος 0 έως 9 και πρέπει να γίνει πρόβλεψη του σωστού αριθμού, σε αυτό το πρόβλημα θα γίνει χρήση μιας συνάρτησης κόστους ταξινόμησης. Ενώ εάν το πρόβλημα είναι η πρόβλεψη μιας συνεχούς τιμής, όπως για παράδειγμα η πρόβλεψη των καιρικών συνθηκών της επόμενης μέρας, τότε χρησιμοποιείται μιας συνάρτηση κόστους παλινδρόμησης.

2.2.1 Συναρτήσεις κόστους παλινδρόμησης (Regression loss functions)

Ένα μοντέλο παλινδρόμησης είναι υπεύθυνο για την πρόβλεψη μίας ποσότητας πραγματικών αριθμών. Οι βασικότερες συναρτήσεις κόστους που χρησιμοποιούνται σε αυτά τα μοντέλα είναι:

ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΜΕΣΟΥ ΤΕΤΡΑΓΩΝΙΚΟΥ ΣΦΑΛΜΑΤΟΣ
(MEAN SQUARED ERROR LOSS – MSE)

Η συνάρτηση μέσου τετραγωνικού σφάλματος είναι η πιο κοινή συνάρτηση κόστους που εφαρμόζεται σε προβλήματα παλινδρόμησης. Το μέσο τετραγωνικό σφάλμα υπολογίζεται από των μέσο όρο του τετραγώνου διαφοράς μεταξύ των επιθυμητών τιμών και των τιμών που έχουν προβλεθεί από το μοντέλο. Πιο συγκεκριμένα αν τα δεδομένα που τροφοδοτούνται στο δίκτυο παραστούν ως σημεία στο καρτεσιανό επίπεδο τότε η εφαρμογή της συνάρτησης του μέσου τετραγωνικού σφάλματος θα δώσει μία ευθεία η οποία θα περνάει από όλα τα σημεία έτσι ώστε η διαφορά μεταξύ όλων των σημείων και των σημείων που ανήκουν στην ευθεία να είναι η μικρότερη δυνατή όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 2.11: Ελαχιστοποίηση του σφάλματος σε ένα πρόβλημα λογιστικής παλινδρόμησης με την βοήθεια της συνάρτησης κόστους

Η μαθηματική εξίσωση αυτής της συγκεκριμένης συνάρτησης κόστους έχει την εξής μορφή:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(yi - \hat{y}i)^2,$$

όπου n είναι το σύνολο των προβλέψεων, y είναι τα επιθυμητά αποτελέσματα και $\hat{y}$ είναι τα αποτελέσματα των προβλέψεων. Η τιμές που μπορεί να πάρει η συνάρτηση θα είναι πάντα θετικές. Το τετράγωνο στην συνάρτηση μεταφράζεται ως ότι μεγαλύτερες αποκλίσεις από την επιθυμητή τιμή οδηγούν σε περισσότερο σφάλμα.

ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΜΕΣΟΥ ΤΕΤΡΑΓΩΝΙΚΟΥ ΛΟΓΑΡΙΘΜΙΚΟΥ ΣΦΑΛΜΑΤΟΣ
(MEAN SQUARED LOGARITHMIC ERROR LOSS – MSLE)

Η συνάρτηση μέσου τετραγωνικού λογαριθμικού σφάλματος, όπως υποδηλώνεται από το όνομα της, είναι μία παραλλαγή του μέσου τετραγωνικού σφάλματος. Η χρήση του λογάριθμου σε αυτή την περίπτωση σημαίνει ότι λαμβάνεται υπόψη η σχετική διαφορά μεταξύ της προβλεπόμενης και της πραγματικής τιμής. Πιο συγκεκριμένα η συνάρτηση αυτή θα έχει παρόμοια συμπεριφορά ανεξαρτήτως της τιμής του σφάλματος όπως φαίνεται στην εικόνα 2.12.

True value	Predicted value	MSE loss	MSLE loss
30	20	100	0.02861
30000	20000	100 000 000	0.03100
	Comment	big difference	small difference

Εικόνα 2.12

Επίσης το αποτέλεσμα της συνάρτησης θα είναι μεγαλύτερο όταν η διαφορά μεταξύ της προβλεπόμενης και επιθυμητής τιμής είναι θετική παρά όταν είναι αρνητική, παρουσιάζοντας έτσι μία ασυμμετρία στην καμπύλη του σφάλματος. Αυτή η συνάρτηση κόστους χρησιμοποιείται σε προβλήματα παλινδρόμησης όπου τα μεγάλου μεγέθους σφάλματα δεν πρέπει να λαμβάνονται περισσότερο υπόψη από τα μικρότερα σφάλματα. Η λειτουργία της περιγράφεται με την παρακάτω μαθηματική έκφραση:

$$\text{MSLE} = \frac{1}{n}\sum_{i=0}^{n} \left(\log(yi + 1) - \log(\hat{y}i + 1)\right)^2,$$

όπου N είναι ο αριθμός των προβλέψεων, y είναι τα πραγματικά αποτελέσματα και ŷ είναι οι προβλέψεις. Η αριθμητική τιμή 1 έχει προστεθεί και στα δύο μέλη της διαφοράς, διότι οι μεταβλητές y και ŷ μπορούν να λάβουν την τιμή 0 που σε αυτή τη περίπτωση ο λογάριθμος δεν θα οριζόταν.

ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΑΠΟΛΥΤΟΥ ΜΕΣΟΥ ΣΦΑΛΜΑΤΟΣ
(MEAN ABSOLUTE ERROR LOSS – MAE)

Πρόκειται για μια ακόμη παραλλαγή της συνάρτησης μέσου τετραγωνικού σφάλματος. Αυτή η συνάρτηση κόστους επιστρέφει το αθροίσμα των διαφορών σε απόλυτο μεταξύ των επιθυμητών και των προβλεπόμενων τιμών διαιρεμένο με το πλήθος των προβλέψεων που πραγματοποιήθηκαν από το μοντέλο. Επομένως υπολογίζεται το μέσο μέγεθος του σφάλματος του δικτύου σε ένα σύνολο δεδομένων, χωρίς να λαμβάνεται υπόψη η κατεύθυνση του. Το ευρός των τιμών που μπορεί να λάβει το αποτέλεσμα αυτής της συνάρτησης κυμαίνεται από το 0 μέχρι το ∞.

$$\text{MAE} = \frac{1}{n}\sum_{\iota=0}^{n} |yi - \hat{y}i|$$

2.2.2 Συναρτήσεις κόστους ταξινόμησης (Classification loss functions)

Σε ένα πρόβλημα ταξινόμησης τα δεδομένα που χρησιμοποιούνται ταξινομούνται σε κατηγορίες. Στόχος του μοντέλου που χρησιμοποιείται είναι να προβλέψει μία τιμή, η οποία αντιπροσωπεύει την κατηγορία που ανήκουν τα δεδομένα εισόδου. Τα προβλήματα ταξινόμησης διακρίνονται σε δύο βασικές κατηγορίες: δυαδικής ταξινόμησης και ταξινόμησης πολλαπλών κλάσεων. Στην περίπτωση της δυαδικής ταξινόμησης τα δεδομένα διαχωρίζονται σε δύο κατηγορίες και οι τιμές εξόδου του μοντέλου μπορούν να λάβουν ένα έυρος τιμών από 0 έως 1. Ενώ στην περίπτωση της ταξινόμησης πολλαπλών κλάσεων οι κατηγορίες στις οποίες έχουν ομαδοποιηθεί τα δεδομένα είναι περισσότερες των δύο και τα αποτελέσματα των προβλέψεων του μοντέλου συνήθως θα είναι ακέραιοι αριθμοί.

ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΕΓΚΑΡΣΙΑΣ ΕΝΤΡΟΠΙΑΣ ΠΟΛΛΑΠΛΩΝ ΚΛΑΣΕΩΝ
(MULTICLASS CROSS ENTROPY LOSS)

Ως συναρτήσεις κόστους στα προβλήματα ταξινόμησης χρησιμοποιούνται συναρτήσεις οι οποίες είναι παραλλαγές της εγκάρσιας εντροπίας (cross entropy). Η εγκάρσια εντροπία είναι η διαφορά μεταξύ δύο κατανομών πιθανοτήτων για μία τυχαία μεταβλητή ή για ένα σύνολο δεδομένων. Στα προβλήματα ταξινόμησης πολλαπλών κατηγοριών χρησιμοποιείται η συνάρτηση κόστους εγκάρσιας εντροπίας πολλαπλών κλάσεων, η οποία έχει την παρακάτω μαθηματική έκφραση:

$$\text{Loss} = -\sum_{j} y\text{i, j} \log(\hat{y}i, j),$$

όπου i είναι ο δείκτης του δείγματος στο σύνολο των δεδομένων που χρησιμοποιούνται για την εκπαίδευση του μοντέλου και j είναι ο δείκτης που αντιπροσωπεύει τις τιμές των επιθυμητών αποτελεσμάτων. Οι μεταβλητές yi,j και ŷi,j δεν είναι πραγματικές τιμές αλλά ερμηνεύονται ως πιθανότητες.

ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΔΥΑΔΙΚΗΣ ΕΓΚΑΡΣΙΑΣ ΕΝΤΡΟΠΙΑΣ
(BINARY CROSS ENTROPY LOSS / LOG LOSS)

Πρόκειται για μία ακόμα παραλλαγή της εγκάρσιας εντροπίας. Αυτή η συνάρτηση είναι από τις πιο κοινές συναρτήσεις κόστους που χρησιμοποιείται σε προβλήματα δυαδικής ταξινόμησης. Μαθηματικά γράφεται ως εξής:

$$\text{Loss} = -\frac{1}{n}\sum_{i=1}^{n} yi \log(\hat{y}i) + (1 - yi) \log(1 - \hat{y}i)$$

Η μεταβλητή ŷi είναι η πρόβλεψη του μοντέλου, yi είναι η αντίστοιχη επιθυμητή τιμή και n ο συνολικός αριθμός προβλέψεων του μοντέλου. Η κατάλληλη συνάρτηση ενεργοποίησης που θα πρέπει να εφαρμόζεται στο επίπεδο εξόδου του δικτύου, στην περίπτωση που γίνεται χρήση αυτής της συνάρτησης κόστους, είναι η σιγμοειδής.

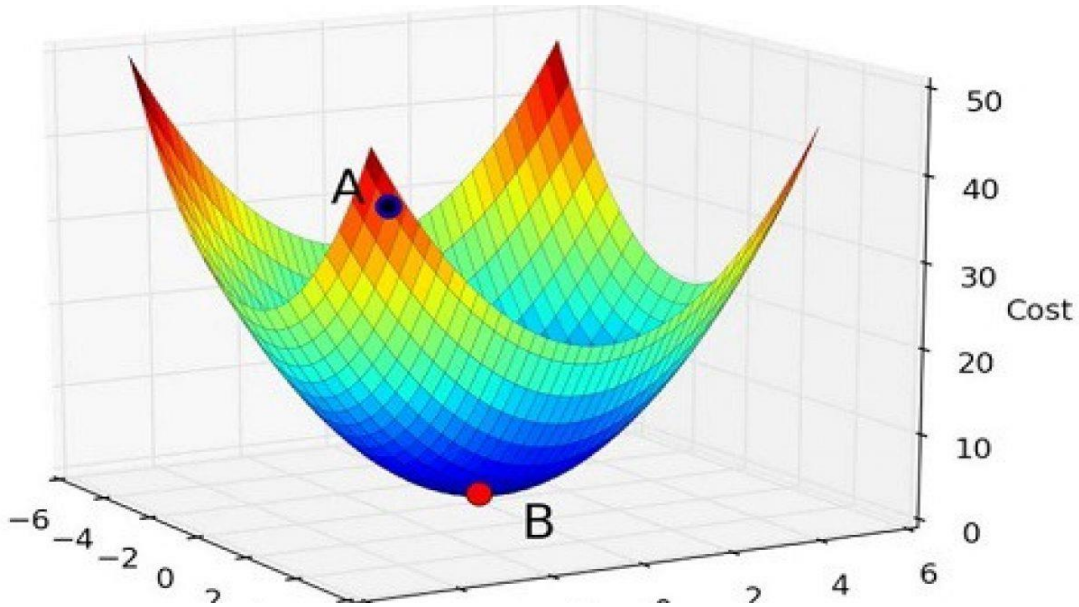
2.3 Αλγόριθμοι βελτιστοποίησης και οπισθοδιάδοση

2.3.1 Αλγόριθμοι βελτιστοποίησης (Optimizers)

Προηγουμένως αναφέρθηκε ότι για να εκπαιδευτεί ένα νευρωνικό δίκτυο βασικός στόχος είναι η ελαχιστοποίηση της συνάρτησης κόστους. Το πρόβλημα εκπαίδευσης του μοντέλου σε αυτή την περίπτωση ερμηνεύεται ως ένα πρόβλημα βελτιστοποίησης στο οποίο γίνεται εύρεση του μέγιστου ή ελαχίστου μίας συνάρτησης. Η συνάρτηση κόστους ελαχιστοποιείται με την βοήθεια ενός αλγόριθμου που ονομάζεται κατάβαση κλίσης (gradient descent). Πρόκειται για έναν αλγόριθμο βελτιστοποίησης με τον οποίο είναι δυνατός ο υπολογισμός ενός τοπικού ελαχίστου μίας διαφορίσιμης συνάρτησης κόστους. Η καλύτερη αναλογία για να γίνει πιο κατανοητή η παραπάνω μέθοδος, είναι αυτή ενός πεζοπόρου ο οποίος βρίσκεται σε μια κοιλάδα με υψώματα και βάθη. Στόχος του πεζοπόρου είναι να κατέβει στο χαμηλότερο σημείο της κοιλάδας (ολικό ελάχιστο της συνάρτησης κόστους). Για να το πετύχει αυτό αρχικά αφού βρίσκεται μακριά από την περιοχή που θέλει να φτάσει, θα αρχίσει να κάνει μεγάλα βήματα προς τον στόχο του. Όσο περισσότερο πλησιάζει στο επιθυμητό σημείο τα βήματά του γίνονται όλο και μικρότερα ώστε να είναι σε θέση να πλησιάσει τον στόχο του με μεγαλύτερη ακρίβεια. Συμπεραίνεται λοιπόν ότι τα βήματα που γίνονται για την προσέγγιση του ελαχίστου σημείου είναι ανάλογα της απόστασης μεταξύ της τωρινής θέσης και του επιθυμητού σημείου. Η παρακάτω εξίσωση περιγράφει τον αλγόριθμο της κατάβασης κλίσης:

$$b = a - \gamma \nabla f(a),$$

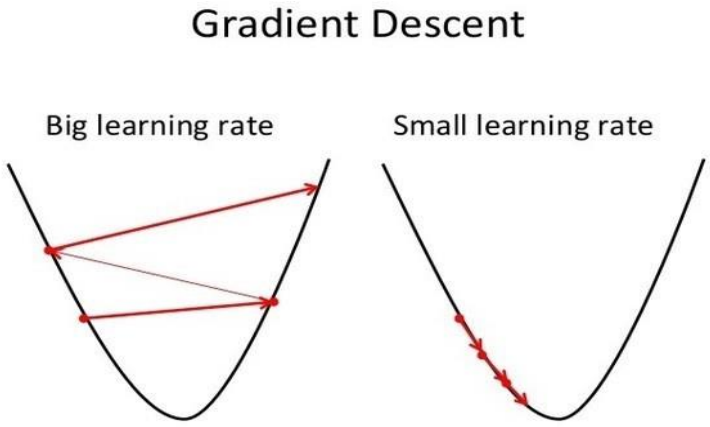
όπου b είναι το επόμενο σημείο της κατάβασης, a είναι η τωρινή θέση, γ είναι το βήμα της κατάβασης ή ρυθμός μάθησης (learning rate) όπως συνήθως αναφέρεται και ο όρος $\nabla f(a)$ είναι η κλίση η οποία υπολογίζεται από την μερική παράγωγο της συνάρτησης κόστους σε σχέση με τα βάρη. Ο όρος «-» εδώ αντιπροσωπεύει την μείωση της κλίσης (κατάβαση). Ένα παράδειγμα εφαρμογής της μεθόδου κατάβασης κλίσης σε ένα πρόβλημα μηχανικής μάθησης είναι το παρακάτω όπου στόχος είναι η ελαχιστοποίηση μιας συνάρτησης κόστους με την προσαρμογή των παραμέτρων της. Στην παρακάτω εικόνα το σημείο Α είναι ένα τυχαίο σημείο που αντιπροσωπεύει την τιμή της συνάρτησης κόστους πριν την εκπαίδευση του νευρωνικού δικτύου ενώ το σημείο Β είναι το ελάχιστο σημείο της συνάρτησης κόστους το οποίο θα προσεγγιστεί εφαρμόζοντας τον αλγόριθμο της κατάβασης κλίσης.



Εικόνα 2.13: Εφαρμογή της μεθόδου κατάβασης κλίσης σε μία συνάρτηση κόστους.

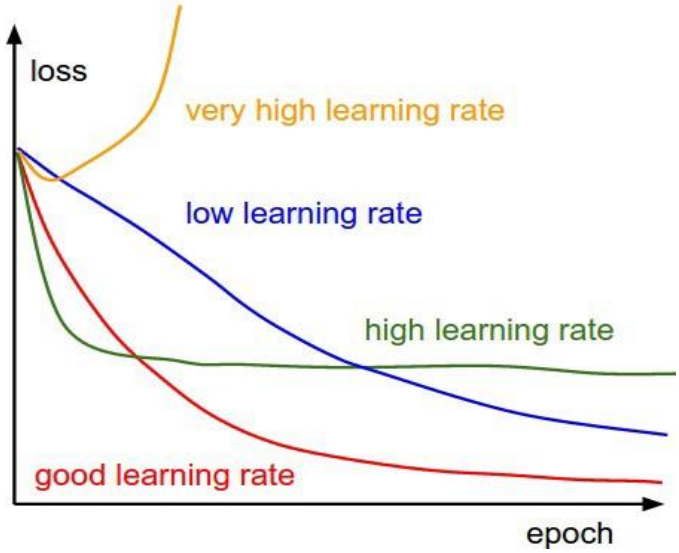
Αρχικά για να βρεθεί το ελάχιστο σημείο της συνάρτησης πρέπει να οριστεί ένα τυχαίο σημείο αρχικοποιώντας αυθαίρετα τις τιμές των συναπτικών βαρών και πολώσεων του νευρωνικού δικτύου. Στη συνέχεια με την βοήθεια του αλγόριθμου της κατάβασης κλίσης βρίσκεται το ολικό ελάχιστο της συνάρτησης κόστους κάνοντας βήματα στην πιο απότομη κλίση. Το μέγεθος του βήματος θα καθορίζεται από την τιμή γ του ρυθμού μάθησης. Εδώ σημειώνεται ότι η εύρεση του ολικού ελαχίστου της συνάρτησης κόστους τις περισσότερες φορές δεν είναι δυνατή εξαιτίας της πολυπλοκότητας της επομένως βρίσκεται ένα τοπικό ελάχιστο.

Για να ελαχιστοποιηθεί σωστά η τιμή της συνάρτησης κόστους, πρέπει να τεθεί μία κατάληλη τιμή στον ρυθμό μάθησης η οποία να μην είναι πολύ μεγάλη αλλά ούτε και πολύ μικρή. Αν τεθεί μία πολύ μεγάλη τιμή τότε το επιθυμητό σημείο θα προσπεραστεί ενώ αν τεθεί μία πολύ μικρή τιμή η ευρέση του τοπικού ελαχίστου θα είναι πολύ χρονοβόρα.



Εικόνα 2.14: Αριστερά η τιμή του ρυθμού μάθησης είναι πολύ μεγάλη ενώ δεξιά είναι πολύ μικρή

Ένας αποτελεσματικός τρόπος παρακολούθησης της συνάρτησης κόστους κατά την διάρκεια της εκπαίδευσης είναι η οπτικοποίηση της με την βοήθεια ενός διαγράμματος. Στην παρακάτω εικόνα φαίνονται οι αλλαγές στο σφάλμα του μοντέλου σε σχέση με τον ρυθμό μάθησης.



Εικόνα 2.15

Όταν η τιμή του ρυθμού μάθησης έχει επιλεχθεί κατάλληλα η τιμή της συνάρτησης κόστους θα πρέπει να μειώνεται. Αν δεν παρατηρείται περαιτέρω μείωση της τιμής της τότε είναι πιθανό να έχει βρεθεί το επιθυμητό ελάχιστο σημείο.

Υπάρχουν αρκετοί αλγόριθμοι βελτιστοποίησης που χρησιμοποιούνται στην μηχανική και βαθιά μάθηση με τους περισσότερους να αποτελούν παραλλαγές της κατάβασης κλίσης. Κύρια διαφορά τους είναι ο αριθμός των δεδομένων που χρησιμοποιούν.

ΚΑΤΑΒΑΣΗ ΚΛΙΣΗΣ ΜΙΑΣ ΠΑΡΤΙΔΑΣ (BATCH GRADIENT DESCENT)

Η κατάβαση κλίσης μίας παρτίδας υπολογίζει το σφάλμα για κάθε παράδειγμα του συνόλου δεδομένων που χρησιμοποιούνται για την εκπαίδευση του μοντέλου με τις παραμέτρους του μοντέλου να προσαρμόζονται μόνο αφού ολοκληρωθεί ένα πέρασμα από όλα τα δεδομένα. Αυτό το πέρασμα στα δεδομένα ονομάζεται μία εποχή (epoch). Τα πλεονεκτήματα της κατάβασης κλίσης μίας παρτίδας είναι ότι προσφέρει αποτελεσματική επεξεργασία, και παράγει σταθερή μείωση του σφάλματος. Ωστόσο τα βασικά μειονεκτήματα της είναι ότι δεν είναι δυνατή η επιπλέον μείωση του σφάλματος από ένα σημείο και έπειτα και ότι χρησιμοποιεί ολόκληρο το σύνολο των δεδομένων για την εκπαίδευση του δικτύου.

ΣΤΟΧΑΣΤΙΚΗ ΚΑΤΑΒΑΣΗ ΚΛΙΣΗΣ (STOCHASTIC GRADIENT DESCENT)

Σε αντίθεση με την προηγούμενη μέθοδο βελτιστοποίησης που περιγράφηκε, η στοχαστική κατάβαση κλίσης ενημερώνει τις παράμετρους του μοντέλου σε κάθε παράδειγμα εκπαίδευσης στα δεδομένα. Ανάλογα το πρόβλημα ο αλγόριθμος αυτός μπορεί να είναι γρηγορότερος στην εύρεση του τοπικού ελαχίστου της συνάρτησης κόστους. Οι συχνές αλλαγές στις παραμέτρους του μοντέλου μπορούν να δώσουν μία πιο λεπτομερής περιγραφή του ρυθμού βελτιστοποίησης. Παρόλα αυτά απαιτείται μεγαλύτερη υπολογιστική ισχύ σε σχέση με την προηγούμενη μέθοδο εξαιτίας της συχνής ενημέρωσης των παραμέτρων του μοντέλου.

ΚΑΤΑΒΑΣΗ ΚΛΙΣΗΣ ΜΙΚΡΩΝ ΠΑΡΤΙΔΩΝ (MINI-BATCH GRADIENT DESCENT)

Είναι η πιο κοινή μέθοδος που χρησιμοποιείται όταν πρόκειται για την εκπαίδευση ενός νευρωνικού δικτύου, καθώς είναι ένας συνδυασμός των δύο παραπάνω αλγορίθμων. Αρχικά το σύνολο των δειγμάτων που θα χρησιμοποιηθούν για την εκπαίδευση του δικτύου θα χωριστεί σε μικρότερα σύνολα τα οποία ονομάζονται παρτίδες (batches) και στη συνέχεια θα πραγματοποιηθεί η διαδικασία ενημέρωσης των παραμέτρων του μοντέλου για κάθε παρτίδα. Ο αριθμός των δειγμάτων σε κάθε παρτίδα μπορεί να διαφέρει συνήθως από 50 έως 256 αν και δεν υπάρχει κάποιος συγκεκριμένος κανόνας ο οποίος εφαρμόζεται για τον υπολογισμό αυτού του αριθμού. Επομένως το σύνολο των δεδομένων σε μία παρτίδα καθορίζεται από την φύση του προβλήματος.

2.3.2 Οπισθοδιάδοση (Backpropagation)

Όπως αναφέρεται παραπάνω προσαρμόζοντας κατάλληλα τα βάρη και τις πολώσεις ενός νευρωνικού δικτύου μπορεί να επιτευχθεί η μέγιστη ακρίβεια του. Για την ενημέρωση των παραμέτρων του δικτύου υπολογίζεται η μερική παράγωγος της συνάρτησης κόστους σε σχέση με τα βάρη και τις πολώσεις του κάθε νευρώνα. Στην περίπτωση όμως ενός νευρωνικού δικτύου πολλών στρωμάτων ο υπολογισμός της μερικής παραγώγου της συνάρτησης κόστους γίνεται πολύ πιο πολύπλοκος και θα πρέπει να επαναλαμβάνεται ακόμα και για την τις πιο μικρές αλλαγές στις παραμέτρους του δικτύου. Ωστόσο η παραπάνω διεργασία είναι δυνατή με την βοήθεια της οπισθοδιάδοσης. Ο αλγόριθμος της οπισθοδιάδοσης αναφέρθηκε για πρώτη φορά το 1970 αλλά η χρησιμότητα του εκτιμήθηκε χάρις ενός άρθρου που δημοσιεύτηκε για πρώτη φορά το 1986 από τους David Rumelhart, Geoffrey Hinton και Ronald Williams. Ουσιαστικά πρόκειται για έναν αλγόριθμο ο οποίος δίνει μία έκφραση της μερικής παραγώγου της συνάρτησης κόστους σε σχέση με το κάθε συναπτικό βάρος του νευρωνικού δικτύου. Το αποτέλεσμα αυτής της έκφρασης δίνει τον τρόπο αλλαγής της συνάρτησης κόστους όταν μεταβάλλονται τα βάρη του δικτύου. Η μέθοδος της οπισθοδιάδοσης χρησιμοποιεί τον κανόνα της αλυσίδας (chain rule) με τον οποίο υπολογίζονται οι παράγωγοι

εμφωλευμένων συναρτήσεων. Για παράδειγμα αν μία συνάρτηση $\mathrm{f(x)}$ περιγράφεται και από άλλη μία συνάρτηση $\mathrm{g(x)}$, τότε η συνάρτηση $\mathrm{f(x)}$ εκφράζεται ως εξής: $f(x) = f(g(x))$. Για τον υπολογισμό της παραγώγου μιας τέτοιας συνάρτησης υπολογίζεται πρώτα η παράγωγος της $\mathrm{f(g(x))}$ ως προς $\mathrm{g(x)}$ και στην συνέχεια το αποτέλεσμα της πολλαπλασιάζεται με την παράγωγο της συνάρτησης $\mathrm{g(x)}$ ως προς χ. Πιο συγκεκριμένα:

$$f(x) = f(g(x)),$$

$$\frac{df(g(x))}{dx} = \frac{df\big(g(x)\big)}{dg(x)} \frac{dg(x)}{dx}$$

Κανόνας αλυσίδας

Η διαδικασία για τον υπολογισμό της μερικής παραγώγου της συνάρτησης κόστους ξεκινάει από το τελευταίο στρώμα του δικτύου και κατευθύνεται προς τα προηγούμενα στρώματα. Αυτός είναι και ο λόγος που αυτή η μέθοδος ονομάζεται οπισθοδιάδοση, γιατί η διαδικασία εκτελείται με την πληροφορία να διαδίδεται προς τα πίσω. Υπάρχουν πολλές παραλλαγές αυτού του αλγόριθμου οι οποίες χρησιμοποιούνται ανάλογα με την αρχιτεκτονική του νευρωνικού δικτύου.

3. ΟΜΟΣΠΟΝΔΗ ΜΑΘΗΣΗ (FEDERATED LEARNING)

Η εξέλιξη του κλάδου της τεχνητής νοημοσύνης έχει ως κεντρική ιδέα την ιδιωτικότητα των δεδομένων που χρησιμοποιούνται για την εκπαίδευση ενός μοντέλου. Μία νέα προσέγγιση η οποία έχει ως στόχο την προστασία της ιδιωτικότητας προτάθηκε για πρώτη φορά από την Google το 2017. Αυτή η προσέγγιση ονομάζεται ομόσπονδη μάθηση και η κύρια ιδέα πάνω στην οποία βασίζεται είναι η δημιουργία ενός μοντέλου το οποίο θα εκπαιδεύεται με δεδομένα που βρίσκονται σε διάφορες συσκευές. Βασική διαφορά σε σχέση με άλλες μεθόδους μάθησης είναι ότι αντί να μεταφέρονται τα δεδομένα στο κύριο μοντέλο, μεταφέρεται ένα αντίγραφο του μοντέλου σε κάθε συσκευή όπου εκεί γίνεται η εκπαίδευση του. Στη συνέχεια οι παράμετροι των εκπαιδευόμενων μοντέλων κάθε συσκευής μεταφέρονται πάλι πίσω στο κεντρικό μοντέλο το οποίο με την σειρά του θα χρησιμοποιήσει αυτές τις παραμέτρους στην εκπαίδευση του. Έτσι τα δεδομένα παραμένουν εντός της συσκευής προστατεύοντας έτσι την ιδιωτικότητα και το κύριο μοντέλο είναι σε θέση να πετυχαίνει υψηλή ακρίβεια.

3.1 Διανεμημένη μάθηση (Distributed learning)

Η διανεμημένη μάθηση είναι ένας διεπιστημονικός τομέας που αποτελείται από πολλά τμήματα όπως στατιστική, αλγόριθμους, μηχανική μάθηση, βαθιά μάθηση, διανεμημένα συστήματα και συστήματα αποθήκευσης. Πρόκειται για την πιο ευρέως διαδεδομένη και αναπτυσσόμενη μέθοδο μηχανικής μάθησης που χρησιμοποιείται στις βιομηχανίες παραγωγής λόγω της ικανότητας της να χειρίζεται μεγάλο πλήθος δεδομένων. Κύριος στόχος αυτής της μεθόδου είναι η μείωση του χρόνου εκπαίδευσης ενός μοντέλου μηχανικής ή βαθιάς μάθησης αυξάνοντας παράλληλα την απόδοση του. Για την επίτευξη αυτού του στόχου αξιοποιούνται παράλληλες ή διανεμημένες τεχνικές υπολογισμών οι οποίες συμβάλλουν στην γρηγορότερη εκπαίδευση του μοντέλου. Οι τεχνικές αυτές μπορούν να ταξινομηθούν σε δύο βασικές κατηγορίες: παραλληλισμό δεδομένων (data parallelism) και παραλληλισμό μοντέλων (model parallelism).

Ο παραλληλισμός δεδομένων είναι μία τεχνική παραλληλισμού η οποία εφαρμόζεται με τον κατανεμισμό δεδομένων. Αρχικά το σύνολο των δειγμάτων διαιρείται σε έναν αριθμό υποσυνόλων ο οποίος είναι ίσος με τον αριθμό των υπολογιστικών μονάδων που θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου. Κάθε υπολογιστική μονάδα αναλαμβάνει να εκτελέσει υπολογισμούς στο αντίστοιχο υποσύνολο δεδομένων που της έχει ανατεθεί και να παράξει ένα σύνολο παραμέτρων. Στη συνέχεια πραγματοποιείται συγχρονισμός των παραμέτρων όλων των μονάδων μέσω διαδικτυακής επικοινωνίας μέχρι να καταλήξουν σε συμφωνία. Με την ταυτόχρονη χρήση πολλαπλών υπολογιστικών μονάδων πραγματοποιείται επεξεργασία ενός μεγαλύτερου αριθμού δεδομένων σε σύγκριση με κλασσικές μεθόδους οι οποίες χρησιμοποιούν μία υπολογιστική μονάδα, με την προυπόθεση όμως ότι ο συχρονισμός

των παραμέτρων δεν απαιτεί πολύ χρόνο για να ολοκληρωθεί. Η μέθοδος του παραλληλισμού των δεδομένων είναι πολύ αποτελεσματική όταν ο αριθμός των δειγμάτων εκαίδευσης είναι πολύ μεγάλος.

Σε σύγκριση με τον παραλληλισμό δεδομένων, ο παραλληλισμός μοντέλων είναι μία πιο περίπλοκη έννοια. Σε αυτή την μέθοδο πραγματοποιείται κατανεμισμός του ίδιου του κύριου μοντέλου με σκοπό τον διαμοιρασμό του φόρτου εργασίας στις υπολογιστικές μονάδες. Ο παραλληλισμός μοντέλων εφαρμόζεται σε περιπτώσεις που το εκπαιδευόμενο μοντέλο έχει πολύ μεγάλο μέγεθος για να μπορέσει να υλοποιηθεί σε μία υπολογιστική μονάδα. Για αυτό τον λόγο κατανέμεται σε περισσότερες υπολογιστικές μονάδες και έτσι είναι εφικτή η υλοποίηση του.

Ωστόσο εξαιτίας της διεργασίας του συγχρονισμού παραμέτρων, η οποία μπορεί να είναι πολύπλοκη και χρονοβόρα, ο χρόνος που απαιτείται για την ολοκλήρωση μίας επανάληψης εκπαίδευσης σε ένα σύμπλεγμα υπολογιστικών μονάδων μπορεί να είναι περισσότερος σε σύγκριση με πιο κλασσικές προσεγγίσεις. Αυτό συμβαίνει γιατί κάθε μονάδα μπορεί να έχει διαφορετικά χαρακτηριστικά και διαφορετικές δυνατότητες. Επομένως για να πραγματοποιηθεί συγχρονισμός των παραμέτρων θα πρέπει οι υπολογιστικές μονάδες που εκτελούν γρηγορότερους υπολογισμούς να αναμένουν τις υπόλοιπες. Συνεπώς η απόδοση του συστήματος δεσμεύεται από τις πιο αδύναμες υπολογιστικές μονάδες.

3.1.1 Διαφορές ανάμεσα στην ομόσπονδη και την διανεμημένη μάθηση

Παρόλο που η ομόσπονδη και η διανεμημένη μάθηση είναι μέθοδοι αρκετά όμοιες μεταξύ τους καθώς και οι δύο χρησιμοποιούν πολλαπλές συσκευές για την εκπαίδευση ενός κεντρικού μοντέλου, παρουσιάζουν κάποιες βασικές διαφορές. Ο διακομιστής παραμέτρων (parameter server) χρησιμοποιείται ως ένα εργαλείο στην διανεμημένη μάθηση για την μείωση του χρόνου εκπαίδευσης. Είναι υπεύθυνος για την αποθήκευση και τον διαμοιρασμό δεδομένων στις υπολογιστικές μονάδες συμβάλλοντας έτσι στην αποτελεσματικότερη εκπαίδευση του μοντέλου. Στην ομόσπονδη μάθηση οι υπολογιστικές μονάδες αντιπροσωπεύονται από τους ιδιοκτήτες των δεδομένων. Ο κάθε ιδιοκτήτης έχει τον ολοκληρωτικό έλεγχο των δεδομένων του και μπορεί να αποφασίσει εκείνος με ποιό τρόπο και πότε θα συμμετέχει στην διαδικασία μάθησης. Επομένως η προσέγγιση της ομόσπονδης μάθησης αντιμετωπίζει ένα πιο σύνθετο περιβάλλον εκμάθησης. Μία ακόμη διαφορά μεταξύ των δύο αυτών μεθόδων είναι ότι η μέθοδος της ομόσπονδης μάθησης δίνει έμφαση στην προστασία της ιδιωτικότητας των δεδομένων του ιδιοκτήτη κατά την διάρκεια της εκπαίδευσης.

3.2 Κατηγορίες ομόσπονδης μάθησης

Τα συστήματα ομόσπονδης μάθησης μπορούν να κατηγοριοποιηθούν με βάση την δομή των χαρακτηριστικών (features) των δεδομένων που χρησιμοποιούν. Έστω ότι τα δεδομένα της κάθε ομάδας που συμμετέχει σε ένα σύστημα ομόσπονδης μάθησης δηλώνονται με την μορφή πινάκων. Οι σειρές και οι στήλες του κάθε πίνακα αναπαριστούν τα δείγματα και τα χαρακτηριστικά τους αντίστοιχα. Επιπλέον τα δεδομένα που ανήκουν σε μία ομάδα μπορεί να περιέχουν δείγματα με ετικέτες (labeled data). Κάθε δεδομένο έχει και από έναν δείκτη ο οποίος θα συμβολίζεται ως I , ενώ τα χαρακτηριστικά και οι ετικέτες του δεδομένου θα συμβολίζονται με X και Y αντίστοιχα. Οι δείκτες, τα χαρακτηριστικά και οι ετικέτες αποτελούν ένα σύνολο δεδομένων εκπαίδευσης (I, X, Y). Για παράδειγμα τα χαρακτηριστικά ενός συνόλου δειγμάτων σε έναν εμπορικό τομέα μπορεί να είναι οι επιθυμίες των πελατών ή μπορεί σε έναν εκπαιδευτικό τομέα να δηλώνουν την απόδοση ενός σπουδαστή. Ο αριθμός των δεδομένων αλλά και των χαρακτηριστικών τους μπορεί να διαφέρει ανάμεσα στις ομάδες που συμμετέχουν σε ένα σύστημα ομόσπονδης μάθησης. Με βάση την κατανομή των χαρακτηριστικών και του πλήθους των δεδομένων στις διάφορες ομάδες η ομόσπονδη μάθηση μπορεί να ταξινομηθεί σε οριζόντια ομόσπονδη μάθηση, κάθετη ομόσπονδη μάθηση και μεταφερόμενη ομόσπονδη μάθηση.

3.2.1 Οριζόντια ομόσπονδη μάθηση (Horizontal federated learning)

Η οριζόντια ομόσπονδη μάθηση χρησιμοποιείται σε περιπτώσεις που το πλήθος των δειγμάτων στα σύνολα των δεδομένων διαφέρει αλλά ο αριθμός των χαρακτηριστικών τους είναι ίδιος. Για παράδειγμα δύο τοπικές τράπεζες μπορεί να έχουν έναν διαφορετικό αριθμό πελατών της περιοχής τους ωστόσο επειδή και οι δύο σαν επιχειρήσεις ανήκουν στον ίδιο τομέα, τα χαρακτηριστικά των πελατών θα είναι ίδια. Το 2017 η Google πρότεινε ως λύση στην ενημέρωση μοντέλων που βρισκόντουσαν σε κινητές συσκευές την εφαρμογή ενός συστήματος οριζόντιας μάθησης. Σε αυτή την περίπτωση ο χρήστης της συσκευής ενημερώνει τοπικά τις παραμέτρους του μοντέλου οι οποίες στην συνέχεια μεταφέρονται σε ένα cloud, επιτρέποντας έτσι την εκπαίδευση του κεντρικού μοντέλου από πολλούς χρήστες ταυτόχρονα. Στο cloud πραγματοποιείται μία διαδικασία κρυπτογράφησης των δεδομένων συμβάλλοντας στην ασφαλή συσωμμάτωση των ανανεωμένων παραμέτρων. Με το πέρας της εκπαίδευσης το κεντρικό μοντέλο μαζί με τις παραμέτρους του μοιράζεται ξανά σε όλους τους χρήστες του συστήματος. Η δημιουργία ενός τέτοιου μοντέλου διασφαλίζει την προστασία των δεδομένων των χρηστών. Ωστόσο το κόστος επικοινωνίας ενός τέτοιου συστήματος μπορεί να είναι υψηλό και επομένως για την αποφυγή μεγάλων χρόνων εκπαίδευσης είναι απαραίτητη η χρήση σύγχρονων και ισχυρών μονάδων επεξεργασίας. ‘Ενα σύστημα οριζόντιας μάθησης προυποθέτει όλοι οι χρήστες του να μην έχουν δεδομένα που μπορεί να εμποδίσουν την ορθή εκπαίδευση του κεντρικού μοντέλου. Ο κύριος διακομιστής που είναι υπεύθυνος για την

συγκέντρωση των ενημερωμένων παραμέτρων των τοπικών μοντέλων, είναι επίσης υπεύθυνος για την προστασία της ιδιωτικότητας των δεδομένων. Ένα τέτοιο σύστημα μπορεί να γίνει ακόμα πιο ασφαλές με την λήψη μέτρων προστασίας του κεντρικού μοντέλου από κακόβουλους χρήστες.

3.2.2 Κάθετη ομόσπονδη μάθηση (Vertical federated learning)

Η κάθετη ομόσπονδη μάθηση εφαρμόζεται σε περιπτώσεις όπου δύο σύνολα δεδομένων αποτελούνται από την ίδια ποσότητα δειγμάτων αλλά διαφέρουν στον αριθμό των χαρακτηριστικών τους. Για παράδειγμα ας θεωρηθούν δύο διαφορετικές εταιρείες οι οποίες ανήκουν στην ίδια περιοχή, με την πρώτη να είναι μία τραπεζική επιχείρηση και την δεύτερη να είναι μία εταιρεία προώθησης προϊόντων. Και οι δύο επιχειρήσεις έχουν ένα κοινό σύνολο πελατών της περιοχής τους, οπότε τα δεδομένα του κάθε χρήστη που συλλέγει η κάθε εταιρεία θα αναφέρονται στο ίδιο πλήθος χρηστών. Ωστόσο επειδή οι δύο οργανισμοί ανήκουν σε διαφορετικούς τομείς τα χαρακτηριστικά των χρηστών τους θα είναι εντελώς διαφορετικά. Αν υποτεθεί ότι αυτές οι δύο εταιρείες χρειάζονται ένα μοντέλο το οποίο θα προβλέπει την καταναλωτική συμπεριφορά των πελατών τους βασιζόμενο στις προτιμήσεις και τις οικονομικές δυνατότητες τους, τότε η εφαρμογή ενός συστήματος κάθετης ομόσπονδης μάθησης είναι κατάλληλη. Η κάθετη ομόσπονδη μάθηση είναι η διαδικασία κατά την οποία συλλέγονται τα διαφορετικά χαρακτηριστικά των χρηστών του συστήματος, τα οποία θα συνεισφέρουν στον υπολογισμό των κατάλληλων παραμέτρων για την βελτιστοποίηση των προβλέψεων του κεντρικού μοντέλου. Η παραπάνω διαδικασία γίνεται με έναν τρόπο ο οποίος έχει ως βασικό στόχο την διατήρηση της ιδιωτικότητας των χρηστών οι οποίοι συνεισφέρουν με τα δεδομένα τους στην εκπαίδευση του κύριου μοντέλου.

Ένα σύστημα κάθετης ομόσπονδης μάθησης λαμβάνει υπόψη κακόβουλες ενέργειες που μπορεί να συμβούν οι οποίες έχουν ως σκοπό την διαφθορά του κεντρικού μοντέλου. Στο παράδειγμα των δύο εταιρειών μπορεί ένα μέρος των δεδομένων που παρέχει μία από τις δύο επιχειρήσεις να περιλαμβάνει δείγματα τα οποία να αυξάνουν το σφάλμα του κεντρικού μοντέλου στις προβλέψεις του. Σε αυτές τις περιπτώσεις ο χρήστης ο οποίος είναι υπεύθυνος για τις κακόβουλες ενέργειες, μπορεί να έχει πρόσβαση μόνο στα δεδομένα της ομάδας που είναι μέρος. Για την διατήρηση της ασφάλειας των δύο ομάδων του συστήματος, προτείνεται η προσθήκη ενός τρίτου μέλους το οποίο θα είναι ανεξάρτητο από τις άλλες δύο ομάδες. Με το τέλος της εκπαίδευσης του μοντέλου, οι ομάδες του συστήματος μπορούν να έχουν πρόσβαση μόνο στις παραμέτρους οι οποίες σχετίζονται με τα αντίστοιχα χαραρακτηριστικά τους επομένως θα πρέπει να συνεργαστούν για την επίτευξη του επιθυμητού αποτελέσματος.

3.2.3 Μεταφερόμενη ομόσπονδη μάθηση (Federated transfer learning)

Η εφαρμογή της μεταφερόμενης ομόσπονδης μάθησης γίνεται σε περιπτώσεις όπου δύο σύνολα δεδομένων διαφέρουν στο πλήθος των δειγμάτων τους αλλά και στον αριθμό των χαρακτηριστικών τους. Για παράδειγμα ας θεωρηθούν δύο οργανισμοί οι οποίοι βρίσκονται σε διαφορετικές περιοχές, με τον πρώτο να ανήκει στον τραπεζικό τομέα και τον δεύτερο να ανήκει στον τομέα προώθησης προϊόντων. Εξαιτίας των διαφορετικών τοποθεσιών τους οι αντίστοιχες ομάδες των πελατών τους θα είναι διαφορετικές στον μεγαλύτερο βαθμό. Επίσης επειδή οι δύο επιχειρήσεις ανήκουν σε διαφορετικούς τομείς τα περισσότερα χαρακτηριστικά των πελατών τους δεν θα είναι ίδια. Σε αυτή την περίπτωση μπορούν να εφαρμοστούν μεταφερόμενες τεχνικές μάθησης οι οποίες παρέχουν λύσεις στο ζήτημα των διαφορετικών δειγμάτων και χαρακτηριστικών. Πιο συγκεκριμένα εξάγονται συσχετίσεις μεταξύ των χαρακτηριστικών των κοινών δειγμάτων. Κοινά δείγματα μπορεί να είναι ένας μικρός αριθμός πελατών ο οποίος ανήκει και στις δύο εταιρείες. Στη συνέχεια αυτές οι σχέσεις των χαρακτηριστικών που έχουν τα κοινά δείγματα μεταξύ τους, χρησιμοποιούνται σε δείγματα που ανήκουν μόνο σε έναν από τους δύο οργανισμούς. Η μεταφερόμενη ομόσπονδη μάθηση είναι μία σημαντική επέκταση των ήδη υπαρχόντων συστημάτων ομόσπονδης μάθησης, διότι μπορεί να αντιμετωπίσει προβλήματα που είναι εκτός του στόχαστρου των κλασσικών προσεγγίσεων ομόσπονδης μάθησης. Τα μέτρα προστασίας που λαμβάνονται σε ένα σύστημα μεταφερόμενης ομόσπονδης μάθησης είναι όμοια με εκείνα που λαμβάνονται στην κάθετη ομόσπονδη μάθηση.

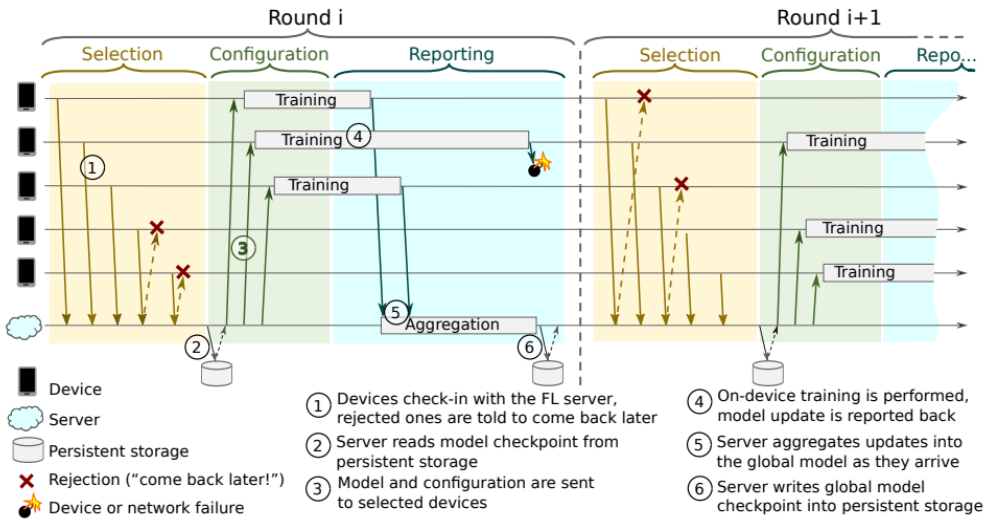
3.3 Αρχιτεκτονική συστημάτων ομόσπονδης μάθησης

Η ομόσπονδη μάθηση πρόκειται για μία νέα έννοια η οποία εξελίσσεται συνεχώς. Επομένως τα συστήματα που δημιουργούνται με βάση αυτή την προσέγγιση μάθησης (FL systems) συνήθως θα διαφέρουν μεταξύ τους καθώς προτείνονται συνεχώς νέες μέθοδοι και αλγόριθμοι οι οποίοι εφαρμόζονται για την υλοποίηση τους. Εδώ γίνεται περιγραφή ενός συστήματος ομόσπονδης μάθησης όπως αυτό προτάθηκε από τον Keith Bonawitz και από άλλους σε ένα επιστημονικό άρθρο με τίτλο «TOWARDS FEDERATED LEARNING AT SCALE: SYSTEM DESIGN», το οποίο δημοσιεύτηκε τον Μάρτιο του 2019. Στόχος αυτού του συστήματος είναι η εκπαίδευση ενός νευρωνικού δικτύου βαθιάς μάθησης χρησιμοποιώντας δεδομένα, τα οποία είναι αποθηκευμένα σε κινητές τηλεφωνικές συσκευές διαφόρων χρηστών, με αυτά να παραμένουν εντός της συσκευής. Με την εκπαίδευση ενός τοπικού μοντέλου στη συσκευή του κάθε χρήστη προκύπτουν νέες παράμετροι οι οποίες συνενώνονται σε ένα cloud μέσω μίας διαδικασίας που ονομάζεται federated averaging, δημιουργώντας έτσι ένα κεντρικό μοντέλο το οποίο προωθείται πάλι πίσω στις συσκευές των χρηστών. Η εκτέλεση μιας διεργασίας που ονομάζεται secure aggregation διασφαλίζει την προστασία της ιδιωτικότητας των δεδομένων που ανήκουν στους χρήστες. Το σύστημα αυτό έχει χρησιμοποιηθεί σε εφαρμογές μεγάλης κλίμακας όπως είναι το εικονικό πληκτρολόγιο SwiftKey που χρησιμοποιείται στα κινητά τηλέφωνα. Παρακάτω γίνεται περιγραφή των τμημάτων αυτού του συστήματος ξεκινώντας από το πρωτόκολλο επικοινωνίας.

3.3.1 Πρωτόκολλο επικοινωνίας

Οι συμμετέχοντες στο πρωτόκολλο είναι στην συγκεκριμένη περίπτωση κινητές τηλεφωνικές συσκευές και ένας διακομιστής FL (FL server), ο οποίος βασίζεται σε μία διανεμημένη υπηρεσία cloud. Οι συσκευές ανακοινώνουν στον διακομιστή ότι είναι έτοιμες να εκτελέσουν μία ομόσπονδη διεργασία (FL task) για έναν συγκεκριμένο αριθμό μελών του συστήματος (FL population). Μία ομόσπονδη διεργασία είναι μία συγκεκριμένη επεξεργασία που πραγματοποιείται από ένα πλήθος μελών του συστήματος. Τέτοιες επεξεργασίες μπορεί να είναι η εκπαίδευση τοπικών μοντέλων στις συσκευές ή η αξιολόγηση αυτών τω μοντέλων με την βοήθεια δεδομένων που ανήκουν στις συσκευές των χρηστών. Καθώς ένας πολύ μεγάλος αριθμός συσκευών ενημερώνει τον διακομιστή FL ότι είναι διαθέσιμες να εκτελέσουν ομόσπονδες διεργασίες, ο διακομιστής θα επιλέξει ένα πλήθος μερικών εκατοντάδων συσκευών για την εκτέλεση της συγκεκριμένης ομόσπονδης εργασίας. Αυτή η επικοινωνία μεταξύ των επιλεγμένων συσκευών και του διακομιστή FL ονομάζεται ένας γύρος (round). Στη συνέχεια ο διακομιστής δίνει συγκεκριμένες οδηγίες στις συσκευές, τις οποίες θα εκτελέσουν για να πραγματοποιηθεί η συγκεκριμένη ομόσποδη διεργασία. Με το πέρας της διεργασίας κάθε συσκευή λαμβάνει τις παραμέτρους του κεντρικού μοντέλου. Στη συνέχεια η κάθε συσκευή πραγματοποιεί μία τοπική επεξεργασία των παραμέτρων που έλαβε

χρησιμοποιώντας τα αντίστοιχα δεδομένα που τους ανήκουν. Οι ανανεωμένες παράμετροι στέλνονται μέσω του διακομιστή πίσω στο κεντρικό μοντέλο και η όλη διαδικασία επαναλαμβάνεται. Η παραπάνω διεργασία εκπαίδευσης του κεντρικού μοντέλου φαίνεται στην *εικόνα 3.1*.



Εικόνα 3.1

Οι συσκευές οι οποίες είναι διαθέσιμες να συμμετέχουν στο σύστημα της ομόσπονδης μάθησης ενημερώνουν ανά τακτά χρονικά διαστήματα τον διακομιστή. Ο διακομιστής FL με την σειρά του επιλέγει ένα υποσύνολο αυτών των συσκευών με βάση τον βέλτιστο αριθμό συσκευών που μπορούν να συμμετέχουν στο σύστημα. Στις συσκευές οι οποίες δεν επιλέχθηκαν ο διακομιστής θα αποστείλει ένα μήνυμα το οποίο θα τις ενημερώνει να επανασυνδεθούν αργότερα. Η διαμόρφωση του διακομιστή γίνεται με βάση την διαδικασία συσσωμάτωσης των παραμέτρων που επιστρέφονται από το κάθε μέλος του συστήματος. Ο διακομιστής αφού λάβει τις ενημερωμένες παραμέτρους από τους χρήστες, τις συσσωματώνει με την διαδικασία federated averaging και στη συνέχεια ενημερώνει τις συσκευές για το πότε να επανασυνδεθούν. Αν το πλήθος των συσκευών είναι επαρκές τότε ο γύρος εκπαίδευσης του κεντρικού μοντέλου θα έχει ολοκληρωθεί με επιτυχία αλλιώς απορρίπτεται. Το πρωτόκολλο επικοινωνίας μεταξύ των συσκευών και του διακομιστή FL έχει μία ανεκτικότητα στις περιπτώσεις που κάποιες συσκευές εγκαταλείψουν την διαδικασία εκπαίδευσης πριν αυτή ολοκληρωθεί. Η επιλογή του πλήθους των συσκευών για την εκπαίδευση του μοντέλου γίνεται με τον διακομιστή να θεωρεί έναν ελάχιστο αριθμό χρηστών με τον οποίο μπορεί να πραγματοποιηθεί η εκπαίδευση. Επιπλέον θεωρείται και ένα χρονικό περιθώριο στο οποίο αν δεν έχει συνδεθεί ο απαραίτητος αριθμός χρηστών που έχει οριστεί ο γύρος εγκαταλείπεται.

Το πρωτόκολλο επικοινωνίας ρυθμίζεται από έναν μηχανισμό ελέγχου ο οποίος ονομάζεται οδηγός ρυθμού (pace steering). Επιτρέπει στον κεντρικό διακομιστή να διαχειρίζεται μικρά και μεγάλα πλήθη συμμετεχόντων. Ο οδηγός του ρυθμού επικοινωνίας

βασίζεται στο χρονικό περιθώριο που δίνει ο διακομιστής στις συσκευές να επανασυνδεθούν. Στην περίπτωση ενός μικρού πλήθους συμμετεχόντων στο σύστημα, ο οδηγός του ρυθμού επικοινωνίας χρησιμοποιείται για να εξασφαλίσει ότι ένας επαρκής αριθμός συσκευών θα συνδεθεί ταυτόχρονα στον διακομιστή. Ωστόσο στην περίπτωση που το σύστημα αποτελείται από πολλές συσκευές ο ρυθμός που αυτές επικοινωνούν με τον διακομιστή θα πρέπει να είναι ο κατάλληλος έτσι ώστε οι συσκευές να μην συνδέονται όλες ταυτόχρονα με αυτόν. Επίσης ο ρυθμός επικοινωνίες θα πρέπει να ρυθμίζεται έτσι ώστε να πραγματοποιούνται και οι απαραίτητες ενέργειες για την ολοκλήρωση των ομόσποδων διεργασιών.

3.3.2 Συσκευές

Όλες οι συσκευές που συμμετέχουν στο σύστημα θα πρέπει να έχουν συγκεντρώσει και αποθηκεύσει δεδομένα για την εκπαίδευση και αξιολόγηση του κεντρικού μοντέλου. Εφαρμογές οι οποίες βρίσκονται σε κάθε συσκευή είναι υπεύθυνες για την διαθεσιμότητα των τοπικών δεδομένων στο περιβάλλον λειτουργίας ομόσπονδης μάθησης (FL runtime) με την μορφή ενός χώρου αποθήκευσης παραδειγμάτων. Για παράδειγμα ο χώρος αποθήκευσης παραδειγμάτων μίας εφαρμογής μπορεί να είναι μία βάση SQL η οποία καταγράφει συγκεκριμένες ενέργειες ενός χρήστη, αφού όμως πρώτα ο χρήστης ενημερωθεί και συναινέσει για την παραπάνω ενέργεια. Ο χώρος αποθήκευσης παραδειγμάτων των εφαρμογών συνήθως είναι περιορισμένος, συνεπώς δεδομένα τα οποία είχαν συλλεχθεί και δεν είναι πλέον απαραίτητα διαγράφονται. Τα τοπικά δεδομένα τα οποία είναι αποθηκευμένα στις συσκευές των χρηστών είναι ευάλωτα απέναντι σε κακόβουλες ενέργειες, για αυτόν τον λόγο οι εφαρμογές οι οποίες είναι υπεύθυνες για την συλλογή τους θα πρέπει να διασφαλίζουν ότι τα δεδομένα είναι κρυπτογραφημένα.

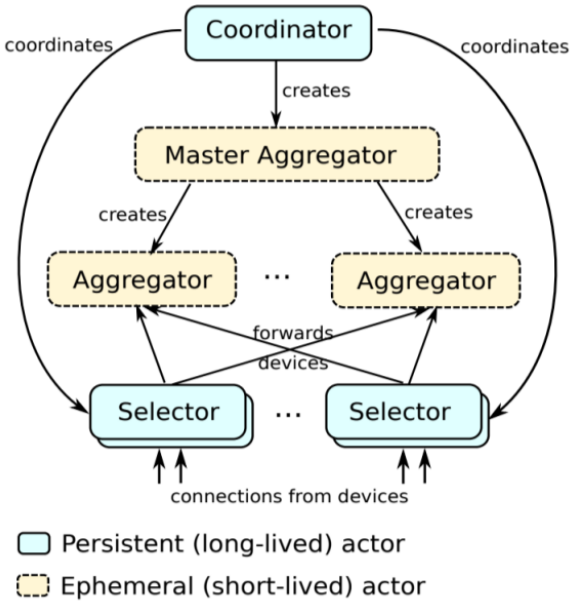
Μία εφαρμογή διαμορφώνει το περιβάλλον λειτουργίας ομόσπονδης μάθησης παρέχοντας ένα πλήθος συσκευών καθώς και τους αποθηκευτικούς χώρους των παραδειγμάτων τους. Το περιβάλλον λειτουργίας οργανώνεται από έναν προγραμματιστή διεργασιών (Android’s JobScheduler). Για να μην επηρεάζεται αρνητικά η απόδοση των συσκευών όταν αυτές χρησιμοποιούνται από τους χρήστες τους, το περιβάλλον λειτουργίας ομόσπονδης μάθησης πραγματοποιείται σε συσκευές οι οποίες είναι αδρανείς, φορτίζονται και είναι συνδεδεμένες σε ασφαλή δίκτυα. Εάν αυτές οι συνθήκες δεν πληρούνται το περιβάλλον λειτουργίας τερματίζεται. Στη περίπτωση που το περιβάλλον ομόσπονδης λειτουργίας ξεκινήσει, ενημερώνει τον διακομιστή FL για το πλήθος των συσκευών οι οποίες είναι διαθέσιμες να επιτελέσουν ομόσπονδες διεργασίες. Εάν επιλεχθούν συσκευές από τον διακομιστή, το περιβάλλον λειτουργίας θα λάβει ένα πλάνο και με βάση αυτό θα επιτελέσει κάποιους υπολογισμούς για την εκπαίδευση ή αξιολόγηση του κεντρικού μοντέλου χρησιμοποιώντας τα δεδομένα τω συσκευών. Αφού ολοκληρωθεί η εκτέλεση του πλάνου οι ανανεωμένοι παράμετροι ή τα στοιχεία αξιολόγησης επιστρέφονται πίσω στον κεντρικό

διακομιστή. Αν δεν υπάρχουν διεργασίες που πρέπει να εκτελεστούν άμεσα, ο διακομιστής θα προτείνει στις συσκευές να επανασυνδεθούν αργότερα.

Οι συσκευές οι οποίες είναι μέλη του συστήματος συμμετέχουν ανώνυμα. Επομένως δεν είναι δυνατό να πιστοποιηθούν μέσω της ταυτότητας των αντίστοιχων χρηστών τους. Για την προστασία του συστήματος από κακόβουλες ενέργειες οι οποίες θα επηρεάσουν αρνητικά το κεντρικό μοντέλο, χρησιμοποιείται μία μέθοδος (Android's remote attestation mechanism) η οποία πιστοποεί όλες τις συσκευές και εφαρμογές που συμετέχουν στο σύστημα.

3.3.3 Διακομιστής

Ο σχεδιασμός του διακομιστή FL διαμορφώνεται έτσι ώστε να διαχειρίζεται έναν μεγάλο αριθμό συσκευών και δεδομένων. Ο διακομιστής FL σχεδιάζεται με βάση ενός μοντέλου προγραμματιστικών φορέων (Actor Programming Model) το οποίο δημοσιεύτηκε το 1973 από τον Carl Hewitt και άλλους. Ο όρος «φορέας» εδώ έχει την ένοια ενός εικονικού επεξεργαστή ο οποίος είναι υπεύθυνος μαζί με άλλους φορείς για την ροή τω δεδομένων στο σύστημα. Ο κάθε φορέας διαχειρίζεται ένα πλήθος ακολουθιών μηνυμάτων. Όταν ένας φορέας λάβει ένα μήνυμα μπορεί να πάρει ανάλογες αποφάσεις, να επικοινωνήσει με άλλους φορείς ή να δημιουργήσει νέους με έναν δυναμικό τρόπο. Ανάλογα με τις απαιτήσεις του συστήματος οι φορείς μπορούν να τοποθετηθούν μαζί σε μία επεξεργαστική μονάδα ή να διανεμηθούν σε πολλαπλά κέντρα επεξεργασίας δεδομένων που βρίσκονται σε διάφορες περιοχές. Οι κύριοι φορείς που συμμετέχουν σε ένα σύστημα ομόσπονδης μάθησης φαίνονται στην *εικόνα 3.2*.



Εικόνα 3.2

Οι συντονιστές (coordinators) είναι οι ανώτεροι φορείς του συστήματος και είναι υπεύθυνοι για την συνεργασία και τον συγχρονισμό μεταξύ άλλων ομάδων φορέων. Σε ένα σύστημα ομόσπονδης μάθησης υπάρχουν πολλαπλοί συντονιστές με τον καθένα να αναλαμβάνει ένα πλήθος συσκευών που είναι μέρος του συστήματος. Ο κάθε συντονιστής καταθέτει την διεύθυνση του και το πλήθος των συσκευών που έχει αναλάβει σε έναν κοινό διακομιστή. Συνεπώς υπάρχει πάντοτε ένας φορέας συντονισμού ο οποίος είναι υπεύθυνος για ένα πλήθος συσκευών και είναι προσιτός και σε άλλους φορείς του συστήματος όπως είναι οι φορείς διαλογών. Ο συντονιστής λαμβάνει πληροφορίες για τον αριθμό των συσκευών που είναι συνδεδεμένες στο κάθε φορέα διαλογής και τους δίνει οδηγίες για το πόσες συσκευές είναι απαραίτητο να συνδεθούν στο σύστημα, για την διεκπεραίωση της ομόσπονδης διεργασίας που έχει ανατεθεί από τον διακομιστή FL. Επίσης οι φορείς συντονισμού μπορούν να δημιουργήσουν νέους φορείς οι οποίοι είναι υπεύθυνοι για την συσσωμάτωση των παραμέτρων που επιστρέφονται μετά από κάθε γύρο εκπαίδευσης των τοπικών μοντέλων.

Οι φορείς διαλογών (selectors) είναι υπεύθυνοι να δέχονται και να προωθούν συνδεδεμένες συσκευές στο σύστημα. Λαμβάνουν περιοδικά πληροφορίες από τον συντονιστή για τον αριθμό των συσκευών που χρειάζεται κάθε φορά και με βάση αυτές τις πληροφορίες αποφασίζουν για το αν θα διαλέξουν ή όχι την κάθε συσκευή. Με την δημιουργία των φορέων συσσωμάτωσης (aggregators), ο φορέας συντονισμού δίνει την εντολή στους διαλογιστές να προωθήσουν ένα υποσύνολο από το πλήθος των συσκευών τους στους φορείς συσσωμάτωσης. Με αυτό τον τρόπο ο συντονιστής είναι σε θέση να αναθέσει διεργασίες ομόσπονδης μάθησης στις συσκευές πιο αποτελεσματικά ανεξάρτητα από τον αριθμό των συσκευών που είναι διαθέσιμος. Οι κύριοι φορείς συσσωμάτωσης (master aggregators) είναι υπεύθυνοι για τους γύρους της κάθε ομόσπονδης επεξεργασίας.

Σε περιπτώσεις αποτυχίας το σύστημα θα συνεχίσει την διαδικασία εκπαίδευσης ολοκληρώνοντας τον τωρινό γύρο ή θα ξεκινήσει από τα αποτελέσματα του προηγούμενου γύρου που ολοκληρώθηκε. Σε πολλές περιπτώσεις η κατάρρευση ενός φορέα δεν θα εμποδίσει την επιτυχή ολοκλήρωση του γύρου. Για παράδειγμα εάν ένας φορέας συσσωμάτωσης ή διαλογής καταρρέυσει οι συσκευές τους που είναι συνδεδεμένες θα είναι οι μόνες που θα αποσυνδεθούν από το σύστημα. Αν ο κύριος φορέας συσσωμάτωσης καταρρεύσει τότε ο γύρος που είχε αναλάβει θα σταματήσει και θα πραγματοποιηθεί η επανεκκίνηση του από τον συντονιστή. Τέλος αν γίνει κατάρρευση του συντονιστή οι φορείς διαλογών θα δημιουργήσουν έναν νέο φορέα συντονισμού.

3.3.4 Αναλυτικά στοιχεία

Ένα σύστημα ομόσπονδης μάθησης έχει ως κύριο στόχο την διασφάλιση της ιδιωτικότητας των χρηστών που συμμετέχουν σε αυτό. Επομένως δεν είναι δυνατή η πρόσβαση στις συσκευές των χρηστών του συστήματος. Για τον λόγο αυτό η παρακολούθηση της λειτουργίας των συσκευών πραγματοποιείται με την βοήθεια αναλυτικών στοιχείων που συλλέγονται από αυτές. Επειδή κάθε συσκευή που συμμετέχει στο σύστημα συνεισφέρει στην εκπαίδευση του κεντρικού μοντέλου, οι επεξεργασίες που πραγματοποιούνται σε αυτές θα πρέπει να μην έχουν αρνητικές επιδράσεις στην απόδοση και την λειτουργία τους. Για να επιτευχθεί αυτό καταγράφεται η δραστηριότητα της κάθε συσκευής σε ένα cloud. Για παράδειγμα αυτές οι καταγραφές μπορεί να αφορούν την κατάσταση στην οποία ήταν η συσκευή όταν αυτή άρχισε να πραγματοποιεί εκπαίδευση σε ένα τοπικό μοντέλο, το ποσό της μνήμης της που χρησιμοποιήθηκε, τα σφάλματα που ανιχνεύθηκαν, την έκδοση του περιβάλλοντος ομόσπονδης λειτουργίας που πραγματοποιήθηκε κ.ο.κ. Αυτές οι καταχωρήσεις δεν περιέχουν πληροφορίες οι οποίες να αφορούν ιδιωτικά στοιχεία και δεδομένα του χρήστη. Στη συνέχεια οι καταγραφές αυτές συγκεντρώνονται σε πίνακες και χρησιμοποιούνται ως αναλυτικά στοιχεία.

Όσον αφορά τον διακομιστή FL συγκεντρώνονται επίσης στοιχεία και πληροφορίες που αφορούν την λειτουργία του. Τέτοια στοιχεία μπορεί να είναι το πλήθος των συσκευών το οποίο επιλέχθηκε για έναν γύρο εκπαίδευσης του κεντρικού μοντέλου, η χρονική διάρκεια των διαφόρων φάσεων του γύρου, σφάλματα που μπορεί να προέκυψαν κ.ο.κ.

3.3.5 Ασφαλής συσσωμάτωση παραμέτρων

Η ασφαλής συσσωμάτωση παραμέτρων (secure aggregation) είναι ένα υπολογιστικό πρωτόκολλο ασφαλείας το οποίο κρυπτογραφεί τις παραμέτρους που θα αποστείλουν οι συσκευές του συστήματος μετά από την εκπαίδευση των αντίστοιχων τοπικών μοντέλων τους. Αφού συγκεντρωθεί ένας επαρκής αριθμός παραμέτρων, το άθροισμα τους θα μεταφερθεί στον διακομιστή FL. Αυτό το υπολογιστικό πρωτόκολλο ασφαλείας μπορεί να εφαρμοσθεί ως ένα ακόμα στρώμα προστασίας των δεδομένων που βρίσκονται στις συσκευές των χρηστών, από κακόβουλες ενέργειες. Η ασφαλής συσσωμάτωση παραμέτρων είναι ένα πρωτόκολλο που αποτελείται από τέσσερις γύρους. Κατά την διάρκεια όλων των γύρων ο διακομιστής συγκεντρώνει πληροφορίες από όλες τις συσκευές που συμμετέχουν στον γύρο εκπαίδευσης του κεντρικού μοντέλου και στη συνέχεια στέλνει μία απάντηση σε κάθε συσκευή ξεχωριστά. Οι δύο πρώτοι γύροι του πρωτοκόλλου συγκροτούν μία φάση προετοιμασίας κατά την οποία συγκεντρώνονται οι συσκευές που θα συμμετάσχουν στην εκπαίδευση του κεντρικού μοντέλου. Σε περίπτωση που κάποια συσκευή αποχωρήσει από την διαδικασία εκπαίδευσης οι παράμετροι της δεν θα χρησιμοποιηθούν. Στον τρίτο γύρω του πρωτοκόλλου οι συσκευές θα προωθήσουν τις παραμέτρους που επεξεργάστηκαν κρυπτογραφημένες στον διακομιστή και

εκείνος με την σειρά του θα υπολογίσει και θα αποθηκεύσει το άθροισμα τους. Τα τοπικά μοντέλα όλων των συσκευών που θα ολοκληρώσουν με επιτυχία αυτό τον γύρο θα ενημερωθούν. Τέλος ο τέταρτος γύρος του πρωτοκόλλου συνιστά μία φάση οριστικοποίησης κατά την οποία οι συσκευές αποκαλύπτουν ένα επαρκές μέρος των παραμέτρων που έχουν υπολογίσει στον διακομιστή ώστε εκείνος να μπορεί να αποκρυπτογραφήσει τις παραμέτρους που έχουν συσσωματωθεί. Δεν είναι απαραίτητο όλες οι συσκευές να ολοκληρώσουν αυτό τον γύρο, παρά μόνο ένας επαρκής αριθμός ώστε το πρωτόκολλο να ολοκληρωθεί με επιτυχία.

3.4 Προκλήσεις που μπορεί να υπάρξουν στην δημιουργία ενός συστήματος fl

Η μέθοδος της ομόσπονδης μάθησης βασίζεται σε δεδομένα τα οποία συλλέγονται από διάφορες συσκευές χρηστών (π.χ κινητά τηλέφωνα) ή και από κέντρα δεδομένων επιχειρήσεων (π.χ τράπεζες). Για την διασφάλιση της ιδιωτικότητας των χρηστών πραγματοποιείται η εκπαίδευση ενός τοπικού μοντέλου στις συσκευές και οι ανανεωμένες παράμετροι που θα προκύψουν επιστρέφονται στο κεντρικό μοντέλο. Με αυτό τον τρόπο τα δεδομένα όλων των μελών που συμμετέχουν σε ένα σύστημα FL παραμένουν στην κατοχή τους και δεν γνωστοποιούνται σε τρίτους. Ωστόσο αυτή η προσέγγιση μάθησης που πραγματοποιείται σε ετερογενείς συσκευές και δίκτυα μεγάλης κλίμακας παρουσιάζει νέες προκλήσεις με τις βασικότερες από αυτές να περιγράφονται παρακάτω.

3.4.1 Υψηλό κόστος επικοινωνίας

Η διασφάλιση των δεδομένων των μελών ενός δικτύου ομόσπονδης μάθησης μπορεί να αυξήσει σημαντικά το κόστος επικοινωνίας στο δίκτυο. Τα συστήματα ομόσπονδης μάθησης μπορεί να αποτελούνται από έναν τεράστιο αριθμό συσκευών με αποτέλεσμα η επικοινωνία μεταξύ των μερών του δικτύου να επιβραδύνει την διαδικασία εκπαίδευσης του κεντρικού μοντέλου. Προκειμένου ο χρόνος εκπαίδευσης του κεντρικού μοντέλου να μην επιβαρύνεται από το κόστος επικοινωνίας του συστήματος χρησιμοποιούνται νέες πιο αποτελεσματικές μέθοδοι επικοινωνίας, οι οποίες αποστέλλουν μικρότερο μέγεθος μηνυμάτων και μικρότερα σύνολα παραμέτρων στον διακομιστή. Οι βασικές λύσεις για την περαιτέρω μείωση του κόστους επικοινωνίας σε ένα τέτοιο σύστημα είναι η μείωση του συνολικού αριθμού των γύρων επικοινωνίας ή η μείωση του μεγέθους των μηνυμάτων που αποστέλλονται σε κάθε γύρο.

3.4.2 Ανομειογένεια συσκευών

Ο χώρος αποθήκευσης, ο τρόπος επεξεργασίας δεδομένων και οι δυνατότητες επικοινωνίας της κάθε συσκευής σε ένα δίκτυο ομόσπονδης μάθησης μπορεί να διαφέρουν εξαιτίας των διαφόρων τεχνικών χαρακτηριστικών που παρουσιάζουν. Για παράδειγμα στην περίπτωση που οι συσκευές ενός συστήματος ομόσπονδης μάθησης είναι κινητά τηλέφωνα, οι διαφορές που θα παρουσιάζουν στα χαρακτηριστικά μπορεί να αφορούν την επεξεργαστική ισχύ τους (CPU, memory), την συνδεσιμότητα τους στο διαδίκτυο (3G, 4G, wifi), το επίπεδο μπαταρίας. Επίσης λόγω του ότι η απόδοση των συσκευών δεν θα πρέπει να επηρεάζεται αρνητικά από την διαδικασία εκπαίδευσης του τοπικού μοντέλου και την αποστολή των ανανεωμένων παραμέτρων, τίθενται κάποιοι περιορισμοί όσον αφορά την συμμετοχή των συσκευών στην διαδικασία της εκπαίδευσης. Για παράδειγμα μία συσκευή μπορεί να συμμετέχει στην διαδικασία της ομόσπονδης μάθησης όταν δεν χρησιμοποιείται από τον

χρήστη της, όταν είναι συνδεδεμένη σε μία πηγή φόρτισης κ.ο.κ. Επιπλέον μπορεί να μην είναι όλες οι συσκευές το ίδιο αξιόπιστες και αρκετά συχνό φαινόμενο είναι η αποσύνδεση ενός πλήθους συσκευών κατά την διάρκεια της εκπαίδευσης. Συνεπώς νέες μέθοδοι ομόσπονδης μάθησης που αναπτύσσονται θα πρέπει να βασίζονται σε ένα σχετικά μικρό πλήθος συμμετεχόντων, να παρουσιάζουν ανεκτικότητα στην ανομοιογένεια των συσκευών που συμμετέχουν στο σύστημα και ανθεκτικότητα στην αποσύνδεση των συσκευών κατά την διάρκεια της εκπαίδευσης.

3.4.3 Προστασία ιδιωτικότητας

Η προστασία της ιδιωτικότητας είναι ζήτημα μείζονος σημασίας στις εφαρμογές της ομόσπονδης μάθησης. Η διασφάλιση των δεδομένων των μελών ενός συστήματος ομόσπονδης μάθησης πραγματοποιείται με τον διαμοιρασμό των παραμέτρων που έχουν προκύψει από την εκπαίδευση των τοπικών μοντέλων σε κάθε συσκευή, αντί να αποστέλλονται τα ίδια τα δεδομένα. Ωστόσο ακόμα και σε αυτή την περίπτωση οι παράμετροι που δημιουργούνται από κάθε συσκευή μπορούν να αποκαλύψουν ευαίσθητα δεδομένα σε τρίτους ή στον κεντρικό διακομιστή FL. Νέες μέθοδοι έχουν ως στόχο την ενίσχυση της ιδιωτικότητας των συστημάτων ομόσπονδης μάθησης χρησιμοποιώντας διαδικασίες ασφαλούς επεξεργασίας δεδομένων για ένα μεγάλο πλήθος συμμετεχόντων. Αυτές οι προσεγγίσεις παρέχουν προστασία της ιδιωτικότητας με κόστος όμως την μείωση της απόδοσης του κεντρικού μοντέλου και της αποτελεσματικότητας του συστήματος. Η κατανόηση και εξισορρόπηση τω εννοιών της ιδωτικότητας και της απόδοσης αποτελεί μία από τις μεγαλύτερες προκλήσεις όσον αφορά στην δημιουργία ενός συστήματος ομόσπονδης μάθησης.

3.5 Προστασία ενός συστήματος fl από κακόβουλους χρήστες

Οι κλασσικές προσεγγίσεις μηχανικής μάθησης βασίζονται πάνω σε έναν αλγόριθμο ο οποίος εκπαιδεύεται με την βοήθεια μίας κεντρικής πηγής δεδομένων. Η διανεμημένη μάθηση προτείνει την κατανομή των δεδομένων και τα στοιχεία ενός μοντέλου σε πολλαπλές υπολογιστικές μονάδες ως μία λύση για την μείωση του χρόνου εκπαίδευσης του μοντέλου. Ωστόσο η μέθοδος της διανεμημένης μάθησης δεν επιλύει το πρόβλημα της προστασίας ιδιωτικότητας και δεν μπορεί να εφαρμοστεί σε περιπτώσεις στις οποίες το πλήθος των υπολογιστικών μονάδων είναι πολύ μεγάλο. Για την αντιμετώπιση των παραπάνω προβλημάτων δημιουργήθηκε η μέθοδος της ομόσπονδης μάθησης. Η ομόσπονδη μάθηση προτείνει μία νέα προσέγγιση εκπαίδευσης στην οποία ένα κεντρικό μοντέλο εκπαιδεύεται με την βοήθεια δεδομένων τα οποία βρίσκονται σε διάφορες συσκευές. Αρχικά πραγματοποιείται η εκπαίδευση ενός τοπικού μοντέλου σε κάθε συσκευή και στη συνέχεια οι παράμετροι που θα προκύψουν θα συσσωματωθούν στο κεντρικό μοντέλο. Τέλος πραγματοποιείται ενημέρωση όλων των τοπικών μοντέλων που βρίσκονται στις συσκευές από το κεντρικό μοντέλο και η διαδικασία εκπαίδευσης επαναλαμβάνεται. Με αυτό τον τρόπο τα δεδομένα που βρίσκονται σε κάθε συσκευή παραμένουν εντός αυτής διασφαλίζοντας έτσι την ιδωτικότητα των δεδομένων. Ωστόσο τα συστήματα ομόσπονδης μάθησης είναι ευάλωτα σε επιθέσεις κατά των εκπαιδευόμενων μοντέλων εξαιτίας της διανεμημένης δομής τους σε διάφορες συσκευές.

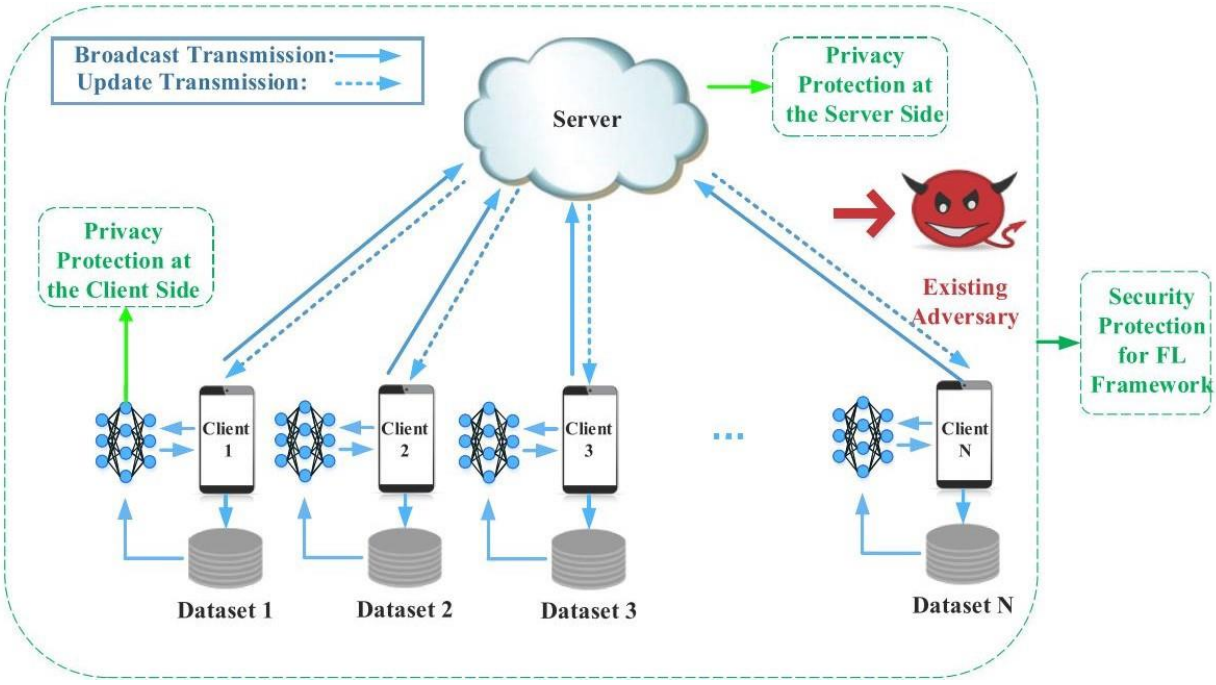
Οι χρήστες των συσκευών και ο διακομιστής ενός συστήματος FL έχουν πρόσβαση στις παραμέτρους που δημιουργούνται κατά την διάρκεια της εκπαίδευσης. Επομένως μπορούν να πραγματοποιηθούν επιθέσεις από τους χρήστες και τον διακομιστή ή από τρίτους οι οποίοι έχουν πρόσβαση στις παραμέτρους που μοιράζονται κατά την διάρκεια επικοινωνίας μεταξύ των συσκευών και του διακομιστή. Υπάρχουν τρία βασικά είδη επιθέσεων: επιθέσεις αλλοίωσης των παραμέτρων του μοντέλου, επθέσεις αλλοίωσης των δεδομένων και επιθέσεις υπεκφυγής.

Οι επιθέσεις αλλοίωσης των παραμέτρων ενός μοντέλου (model update poisoning attacks) έχουν ως στόχο την αλλοίωση των παραμέτρων τοπικών μοντέλων που βρίσκονται στις συσκευές των χρηστών. Σε αυτή την περίπτωση οι κακόβουλοι χρήστες (adversaries) ελέγχουν έναν αριθμό συσκευών (εικόνα 3.3) και είναι σε θέση να μεταβάλλουν τις παραμέτρους που δημιουργούνται από αυτές, κατευθύνοντας έτσι το αποτέλεσμα του κεντρικού μοντέλου με στόχο την εξυπηρέτηση δικών τους σκοπών. Αυτού τους είδους οι επιθέσεις ταξινομούνται σε στοχευμένες επιθέσεις και μη στοχευμένες επιθέσεις. Οι μη στοχευμένες επιθέσεις έχουν ως στόχο την αλλοίωση των αποτελεσμάτων του κεντρικού μοντέλου. Αντίθετα οι στοχευμένες επιθέσεις εστιάζουν στην αλλοίωση ενός τοπικού μοντέλου μιας συσκευής. Η αντιμετώπιση αυτού του είδους επιθέσεων αποτελεί μία μεγάλη πρόκληση καθώς τα τοπικά μοντέλα που επηρεάζονται από αυτές τις επιθέσεις δεν παρουσιάζουν, στις περισσότερες περιπτώσεις,

αρκετά μεγάλες διαφορές σε σχέση με τα υπόλοιπα τοπικά μοντέλα. Για την αντιμετώπιση αυτού του είδους επιθέσεων είναι απαραίτητη η πρόσβαση στα δεδομένα των χρηστών, πράγμα το οποίο δεν είναι δυνατό στα περιβάλλοντα ομόσπονδης μάθησης.

Οι επιθέσεις αλλοίωσης δεδομένων (data poisoning attacks) χαρακτηρίζονται ως πιο περιορισμένες επιθέσεις καθώς σε αυτές τις περιπτώσεις ένας κακόβουλος χρήστης μπορεί να μεταβάλλει μόνο τα τοπικά δεδομένα μίας συσκευής. Για να το πετύχει αυτό αλλοιώνει τα χαρακτηριστικά και τις ετικέτες των δεδομένων που βρίσκονται στην συσκευή. Αλλοίωση των δεδομένων μπορεί να πραγματοποιηθεί και στην περίπτωση μίας επίθεσης υπεκφυγής, με βασική διαφορά όμως ότι τα δεδομένα που μεταβάλλονται σε αυτή την περίπτωση είναι δείγματα τα οποία τροφοδοτούνται σε ένα τοπικό μοντέλο κατά την διάρκεια της εκπαίδευσης. Ωστόσο στις επιθέσεις αλλοίωσης δεδομένων τα δεδομένα που μεταβάλλονται μπορεί να είναι δείγματα που θα χρησιμοποιηθούν για την αξιολόγηση του μοντέλου.

Μέθοδοι οι οποίες έχουν ως στόχο την διασφάλιση της ιδιωτικότητας των χρηστών προστατεύουν τις πληροφορίες που μοιράζονται κατά την διάρκεια επικοινωνίας μεταξύ του διακομιστή και των συσκευών. Επομένως οι μηχανισμοί προστασίας των συστημάτων FL που αναπτύσσονται θα πρέπει να επικεντρώνονται στην αντιμετώπιση των επιθέσεων που έχουν ως στόχο τους χρήστες. Παρακάτω γίνεται περιγραφή μίας μεθόδου αντιμετώπισης κακόβουλων χρηστών, η οποία χρησιμοποιείται σε ένα δυναμικό μοντέλο ομόσπονδης μάθησης όπως αυτό προτάθηκε από την Nuria Rodriguez-Barroso και από άλλους σε ένα επιστημονικό άρθρο με τίτλο «Dynamic Federated Learning Model for Identifying Adversarial Clients» που δημοσιεύθηκε τον Ιούλιο του 2020.



Εικόνα 3.3

3.5.1 Δυναμικό σύστημα fl για την αναγνώριση κακόβουλων χρηστών

Τα συστήματα ομόσπονδης μάθησης δεν είναι σε θέση να έχουν πρόσβαση στα δεδομένα εκπαίδευσης των τοπικών μοντέλων που βρίσκονται στις συσκευές των χρηστών. Επομένως οι επιθέσεις που πραγματοποιούνται που έχουν ως σκοπό την μεταβολή των δεδομένων που είναι αποθηκευμένα σε τοπικές συσκευές, και πιο συκεκριμένα την αλλοίωση των ετικετών των δειγμάτων εκπαίδευσης, μπορούν να διαφθείρουν το κεντρικό μοντέλο FL χωρίς να είναι δυνατή η ανίχνευση τους. Οι επιθέσεις που έχουν ως στόχο την αλλοίωση των ετικετών ενός συνόλου δεδομένων εκπαίδευσης (dirty-label poisoning adversarial attack), μπορούν να προσομοιωθούν αναθέτοντας αυθαίρετα ετικέτες στα δείγματα εκπαίδευσης ενός υποσυνόλου συμμετεχόντων σε ένα σύστημα FL. Για παράδειγμα έστω ένας συμμετέχοντας σε ένα περιβάλλον ομόσπονδης μάθησης ο οποίος συμβολίζεται με C_i και τα τοπικά δεδομένα που του ανήκουν συμβολίζονται με D_i. Τα δεδομένα D_i αποτελούνται από τα δείγματα εκπαίδευσης και τις αντίστοιχες ετικέτες τους, επομένως μπορούν να εκφραστούν και ως $D_i = \{x_i, y_i\}$, όπου x_i είναι τα δείγματα εκπαίδευσης και y_i οι ετικέτες των δειγμάτων. Ο συμμετέχοντας C_i θα αποτελεί έναν κακόβουλο χρήστη (adversarial client) στην περίπτωση που χρησιμοποιεί ένα αλλοιωμένο σύνολο δειγμάτων $\dot{D}_i$ το οποίο εκφράζεται ως $\dot{D}_i = \{x_i, \acute{y}_i\}$, όπου $\acute{y}_i$ είναι ένα σύνολο ετικετών το οποίο έχει επιλεχθεί αυθαίρετα από τον κακόβουλο χρήστη.

Όσον αφορά την αδυναμία ενός συστήματος FL να έχει πρόσβαση στα δεδομένα των συμμετεχόντων για να είναι σε θέση να διακρίνει κακόβουλους χρήστες, προτείνεται ένα δυναμικό μοντέλο FL το οποίο επιλέγει τους χρήστες που θα συμμετέχουν στην διαδικασία εκπαίδευσης και αποκλείει χρήστες οι οποίοι είναι πιθανόν να έχουν ως στόχο την αλλοίωση του κεντρικού μοντέλου. Το δυναμικό αυτό μοντέλο FL χρησιμοποιεί έναν χειριστή ομόσπονδης συσσωμάτωσης (federated aggregation operator) ο οποίος βασίζεται σε έναν χειριστή IOWA (Induced Ordered Weighted Averaging). Οι χειριστές IOWA είναι συναρτήσεις οι οποίες σταθμίζουν την συνεισφορά ενός συνόλου χρηστών κατά την διάρκεια της διαδικασίας συσσωμάτωσης παραμέτρων. Αποτελούνται από μία συνάρτηση ταξινόμησης (induced ordering function) η οποία αξιολογώντας την απόδοση των συμμετεχόντων, τους ταξινομεί με βάση την αξιοπιστία τους και από έναν ποσοτικοποιητή (linguistic quantifier) ο οποίος είναι υπεύθυνος για το ποσοστό της συνεισφοράς που θα έχει ο κάθε χρήστης στην εκπαίδευση του κεντρικού μοντέλου. Η αξιολόγηση της απόδοσης των συμμετεχόντων ενός συστήματος ομόσπονδης μάθησης πραγματοποιείται με την βοήθεια μίας συνάρτησης τοπικής ακρίβειας (Local Accuracy Function), η οποία υπολογίζει την απόδοση ενός τοπικού μοντέλου χρησιμοποιώντας ένα σχετικά μικρό σύνολο δειγμάτων αξιολόγησης (validation set). Στη συνέχεια με βάση την απόδοση των συμμετεχόντων ο ποσοτικοποιητής θα ορίσει το ποσοστό της συνεισφοράς του κάθε χρήστη στην διαδικασία της εκπαίδευσης. Με αυτό τον τρόπο τοπικά μοντέλα τα οποία έχουν δεχθεί αλλοίωση στα δεδομένα τους και επομένως η απόδοση τους είναι χαμηλή, θα έχουν μία ανεπαίσθητη επιρροή στο κεντρικό μοντέλο FL.

3.6 Σύστημα ομόσπονδης μάθησης με αρχιτεκτονική blockchain (BlockFL)

Όπως προαναφέρθηκε η εκπαίδευση ενός κεντρικού μοντέλου σε ένα σύστημα FL πραγματοποιείται αξιοποιώντας τα δεδομένα που βρίσκονται σε ένα μεγάλο πλήθος συσκευών, ανεξαρτήτως της τοποθεσίας τους, και χρησιμοποιώντας έναν διακομιστή ο οποίος είναι υπεύθυνος για τη συσσωμάτωση των παραμέτρων που προκύπτουν από την κάθε συσκευή. Ωστόσο η διαδικασία συσσωμάτωσης των παραμέτρων μπορεί να είναι πολύ χρονοβόρα και ένα σύστημα FL είναι ευάλωτο σε περιπτώσεις δυσλειτουργίας του διακομιστή. Επίσης μία συσκευή μπορεί να συνεισφέρει περισσότερο από άλλες στην εκπαίδευση του κεντρικού μοντέλου χωρίς όμως να της προσφέρεται μία επιπλέον επιβράβευση. Επομένως ο χρήστης αυτής της συσκευής είναι πιθανό να μην είναι πρόθυμος πλέον να συμμετέχει στην διαδικασία της εκπαίδευσης. Προκειμένου να επιλυθούν τα παραπάνω ζητήματα προτάθηκε ένα σύστημα FL με αρχιτεκτονική blockchain (BlockFL) όπου ένα δίκτυο blockchain είναι σε θέση να διαμοιράζει τις παραμέτρους των τοπικών μοντέλων μεταξύ τους, επιβραβεύοντας παράλληλα τους συμμετέχοντες ανάλογα με την συνεισφορά τους στην εκπαίδευση του κεντρικού μοντέλου. Με την ανάλογη επιβράβευση των συμμετεχόντων προωθείται η συμμετοχή ενός μεγαλύτερου πλήθους συσκευών οι οποίες έχουν περισσότερα δεδομένα. Επιπλέον ένα σύστημα BlockFL δεν είναι τόσο ευάλωτο σε περιπτώσεις δυσλειτουργιών και είναι σε θέση να αντιμετωπίσει κακόβουλους χρήστες, αφού οι παράμετροι του κάθε τοπικού μοντέλου αξιολογούνται μέσω μίας διαδικασίας επικύρωσης. Για την βαθύτερη κατανόηση ενός συστήματος BlockFL περιγράφεται αρχικά ο τρόπος λειτουργίας του blockchain.

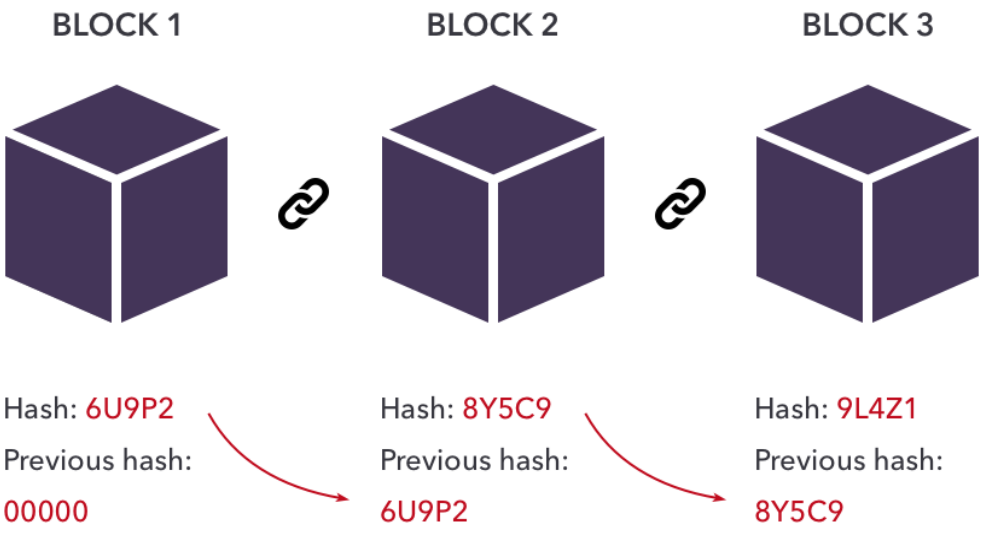
3.6.1 Τι είναι το blockchain

Το blockchain πρόκειται ουσιαστικά για ένα είδος βάσης δεδομένων. Μία βάση δεδομένων είναι μία συλλογή πληροφοριών η οποία αποθηκεύεται σε ένα ηλεκτρονικό σύστημα. Οι πληροφορίες που είναι αποθηκευμένες στις βάσεις δεδομένων ταξινομούνται με την δομή ενός πίνακα έτσι ώστε να είναι δυνατή η αναζήτηση και η επεξεργασία τους. Όταν πρόκειται για μεγάλου μεγέθους βάσεις δεδομένων οι παραπάνω διεργασίες επιτυγχάνονται με την αποθήκευση των πληροφοριών σε διακομιστές οι οποίοι αποτελούνται από πολλαπλές υπολογιστικές μονάδες. Βασική διαφορά μεταξύ μιας τυπικής βάσης δεδομένων και του blockchain είναι ο τρόπος δομής των δεδομένων. Το blockchain συλλέγει πληροφορίες με την μορφή συνόλων τα οποία ονομάζονται blocks. Κάθε σύνολο έχει περιορισμένο χώρο αποθήκευσης και συνδέεται με το προηγούμενο δημιουργώντας έτσι μία αλυσίδα δεδομένων. Με την είσοδο ενός block στην αλυσίδα του δίνεται μία χρονική σήμανση, δημιουργώντας έτσι μία χρονοσειρά δεδομένων η οποία είναι αμετάκλητη.

Βασικός στόχος του blockchain είναι η αποθήκευση και η διανομή ψηφιακών πληροφοριών χωρίς όμως να είναι δυνατή η επεξεργασία τους. Αυτή η μέθοδος εφαρμόστηκε για πρώτη φορά σε ένα σύστημα κρυπτονομισμάτων το οποίο χρησιμοποιούσε το blockchain

για την αποθήκευση όλων των συναλλαγών που πραγματοποιόντουσαν. Κάθε block στην αλυσίδα κατείχε πληροφορίες για την κάθε συναλλαγή και στην περίπτωση σφάλματος στα δεδομένα του, μπορούσε να χρησιμοποιήσει τα υπόλοιπα σύνολα δεδομένων που αποτελούσαν μέρη της αλυσίδας ως σημεία αναφοράς για να διορθώσει το σφάλμα. Επομένως οι πληροφορίες που είναι αποθηκευμένες στο blockchain δεν μπορούν να υποστούν κάποια επεξεργασία εκτός αν η πλειοψηφία του δικτύου συμφωνήσει στην εφαρμογή κάποιων αλλαγών στο σύστημα. Κάθε τμήμα του blockchain περιέχει ένα αντίγραφο της αλυσίδας το οποίο ενημερώνεται κάθε φορά που ένα νέο block προστίθεται. Επομένως οι χρήστες του συστήματος μπορούν να παρακολουθούν οποιαδήποτε συναλλαγή συμβαίνει στο σύστημα.

Η τεχνολογία του blockchain διασφαλίζει την προστασία των δεδομένων που είναι αποθηκευμένα στο σύστημα με πολλούς τρόπους. Οποιοδήποτε σύνολο δεδομένων αποθηκευτεί στο σύστημα προστίθεται στο τέλος της αλυσίδας, επομένως ο τρόπος αποθήκευσης πληροφοριών είναι γραμμικός. Αφού πραγματοποιηθεί η πρόσθεση ενός block στην αλυσίδα, είναι πολύ δύσκολο να μεταβληθούν τα περιεχόμενα του εκτός αν τα περισσότερα μέλη του συστήματος καταλήξουν σε συμφωνία. Κάθε τμήμα του blockchain περιέχει μία μοναδική ετικέτα (hash) μαζί με την ετικέτα του προηγούμενου block καθώς και μία χρονική σήμανση, όπως φαίνεται στην εικόνα 3.4.



Εικόνα 3.4

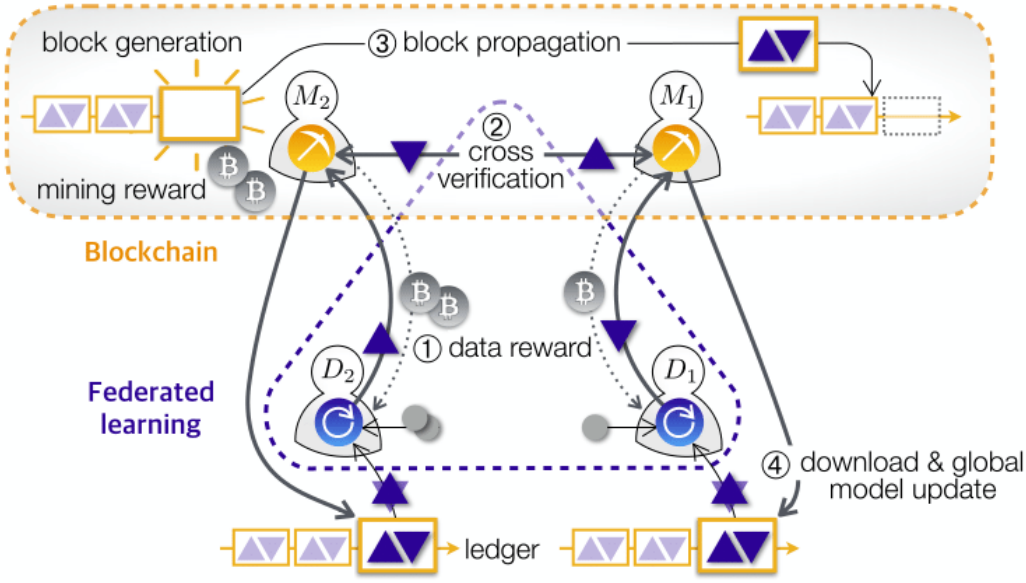
Οι ετικέτες αυτές δημιουργούνται με την βοήθεια μίας συνάρτησης η οποία μετατρέπει ψηφιακές πληροφορίες σε μία σειρά χαρακτήρων. Αν για κάποιο λόγο οι πληροφορίες σε ένα σύνολο δεδομένων του blockchain μεταβληθούν, θα αλλάξει και η ετικέτα αυτού του συνόλου. Αυτή η ιδιότητα του blockchain έχει πολύ σημαντικό ρόλο στην ασφάλεια του συστήματος. Για παράδειγμα αν ένας χρήστης του συστήματος ήθελε να αλλοιώσει τις πληροφορίες του

συνόλου δεδομένων που του άνηκε, για την εξυπηρέτηση δικών του σκοπών, τότε οι ετικέτες των τμημάτων του blockchain θα μεταβαλλόντουσαν και οι υπόλοιποι χρήστες του συστήματος θα ενημερωνόντουσαν για αυτή την αλλαγή. Στη συνέχεια το block του κακόβουλου χρήστη θα αφαιρούνταν από το σύστημα χωρίς να έχουν αλλοιωθεί καθόλου τα δεδομένα των υπολοίπων χρηστών. Ακόμα και αν ένας χρήστης κατάφερνε να έχει στην κατοχή του το μεγαλύτερο τμήμα του blockchain, αυτό θα απαιτούσαι την σπατάλη ενός μεγάλου αριθμού πόρων και επομένως θα καθιστούσε την οποιαδήποτε επίθεση στο σύστημα άσκοπη.

Το blockchain χρησιμοποιήθηκε για πρώτη φορά σε συστήματα συναλλαγών κρυπτονομισμάτων αλλά από τότε έχει βρει εφαρμογή και σε πολλούς άλλους τομείς όπως είναι η βιομηχανία τροφίμων, οι τραπεζικές συναλλαγές, τα κέντρα υγείας κ.ο.κ. Ένα παράδειγμα αποτελεί η χρήση του blockchain από την εταιρεία IBM η οποία δημιούργησε ένα σύστημα το οποίο βασίζεται στο blockchain για τον εντοπισμό της διαδρομής που έκαναν τα προϊόντα τροφίμων να φτάσουν στον προορισμό τους. Επομένως σε περίπτωση που κάποιο τρόφιμο βρεθεί μολυσμένο η εταιρεία έχει την δυνατότητα να ακολουθήσει τη διαδρομή που έκανε το προϊόν και επομένως να εντοπίσει το πρόβλημα που προέκυψε σε πολύ μικρότερο χρονικό διάστημα και με αποτελεσματικότερο τρόπο. Ένα άλλο παράδειγμα είναι η χρήση συστημάτων blockchain από τις τραπεζικές επιχειρήσεις. Οι πελάτες των τραπεζικών επιχειρήσεων δεν έχουν την δυνατότητα να πραγματοποιούν οποιαδήποτε στιγμή χρηματικές συναλλαγές, διότι η τράπεζα θα πρέπει πρώτα να τις επεξεργαστεί. Με την ενσωμάτωση της τεχνολογίας blockchain στα τραπεζικά συστήματα οι πελάτες είναι σε θέση να πραγματοποιούν τις συναλλαγές που επιθυμούν οποιαδήποτε στιγμή σε πολύ λιγότερο χρόνο και με περισσότερη ασφάλεια. Το blockchain μπορεί να εφαρμοστεί και σε άλλου είδους συστήματα στα οποία βασική προτεραιότητα είναι η διασφάλιση της προστασίας των δεδομένων των χρηστών όπως και συμβαίνει στην περίπτωση ενός συστήματος BlockFL.

3.6.2 Περιγραφή ενός συστήματος BlockFL

Η δομή ενός συστήματος BlockFL αποτελείται από δύο ομάδες συσκευών. Η πρώτη ομάδα μπορεί να αποτελείται από οποιοδήποτε είδος συσκευών, με κάθε συσκευή να πραγματοποιεί μία διαδικασία εκπαίδευσης στο αντίστοιχο τοπικό μοντέλο της. Η δεύτερη ομάδα συσκευών απαρτίζεται από ένα πλήθος υπολογιστικών μονάδων, οι οποίες έχουν επιλεχθεί τυχαία, και είναι υπεύθυνη για την διανομή και επαλήθευση των παραμέτρων που έχουν προκύψει από την εκπαίδευση των τοπικών μοντέλων που βρίσκονται στις συσκευές της πρώτης ομάδας. Οι υπολογιστικές μονάδες που πραγματοποιούν αυτή την διαδικασία ονομάζονται miners και το σύνολο των διεργασιών που εκτελούν ονομάζεται mining. Η λειτουργία ενός συστήματος BlockFL ξεκινάει με τις συσκευές να υπολογίζουν τις νέες παραμέτρους των τοπικών μοντέλων τους και να τις προωθούν στους αντίστοιχους miners που βρίσκονται στο δίκτυο blockchain. Στην συνέχεια οι miners ανταλλάσουν και επικυρώνουν όλες τις παραμέτρους των τοπικών μοντέλων ενώ παράλληλα εκτελούν μία διεργασία η οποία ονομάζεται Proof of Work (PoW). Κατά την διάρκεια αυτής της διεργασίας ένας miner εκτελεί πολύπλοκες υπολογιστικές πράξεις οι οποίες όταν ολοκληρωθούν εκείνος θα λάβει μία ανταμοιβή, συνήθως με την μορφή κρυπτονομίσματος, και έπειτα δημιουργείται ένα νέο block στο οποίο έχουν αποθηκευτεί οι επαληθευμένες παράμετροι των τοπικών μοντέλων. Το block που δημιουργήθηκε προστίθεται στη αλυσίδα του blockchain και στη συνέχεια λαμβάνεται από τις συσκευές. Τέλος αφού κάθε συσκευή έχει πρόσβαση σε όλες τις παραμέτρους των τοπικών μοντέλων, μπορεί να υπολογίσει τις ενημερώσεις που θα εφαρμοστούν στο κεντρικό μοντέλο. Η λειτουργία ενός συστήματος BlockFL που περιγράφηκε παραπάνω φαίνεται στην εικόνα 3.5.



Εικόνα 3.5

Σε ένα σύστημα BlockFL τα blocks και οι επαλήθευση τους που πραγματοποιείται από τους miners σχεδιάζονται έτσι ώστε να ανταλλάσονται οι παράμετροι των τοπικών μοντέλων μέσω μίας διανεμημένης αλυσίδας πληροφοριών. Κάθε block στην αλυσίδα αποτελείται από το κυρίως σώμα του και τα μέρη της κεφαλής του. Το σώμα ενός block αποθηκεύει τις παραμέτρους των τοπικών μοντέλων των συσκευών καθώς και τον χρόνο που χρειάστηκε για τον υπολογισμό τους. Η κεφαλή του κάθε block περιέχει πληροφορίες για την ετικέτα του προηγούμενου block, για τον ρυθμό δημιουργίας του καθώς και το αποτέλεσμα της διεργασίας PoW. Το μέγεθος ενός block καθορίζεται από την συνάρτηση $size = h + \delta mND$, όπου h είναι η κεφαλή του block και δ_m είναι το μέγεθος των παραμέτρων του τοπικού μοντέλου της συσκευής N_D. Κάθε miner έχει από ένα υποψήφιο block το οποίο κατέχει πληροφορίες για τις παραμέτρους των τοπικών μοντέλων των αντίστοιχων συσκευών και πληροφορίες από τους άλλους miners. Η διαδικασία αποθήκευσης πληροφοριών στο block συνεχίζεται μέχρι το πλήθος των δεδομένων να γίνει ίσο με το μέγεθος του block ή η διάρκεια της διαδικασίας να ξεπεράσει έναν χρόνο αναμονής. Στη συνέχεια ο miner εκτελώντας την διεργασία PoW δημιουργεί τυχαία μία ετικέτα και επαναλαμβάνει αυτή την διαδικασία μέχρι η ετικέτα που θα δημιουργήσει να μην ξεπερνάει μία επιθυμητή τιμή. Το υποψήφιο block του miner, που θα ολοκληρώσει πρώτος την διαδικασία, θα προστεθεί στην αλυσίδα του blockchain και θα διανεμηθεί στους υπόλοιπους miners. Όλοι οι miners που έλαβαν το block είναι αναγκασμένοι να σταματήσουν τις διεργασίες που εκτελούσαν και να το προσθέσουν στην τοπική τους αλυσίδα πληροφοριών. Ο ρυθμός δημιουργίας ενός block εξαρτάται από τον βαθμό δυσκολίας της διεργασίας PoW. Όσο μικρότερη είναι η επιθυμητή τιμή που θα πρέπει να πάρει η ετικέτα, τόσο δυσκολότερη καθίσταται η εκτέλεση της διεργασίας PoW και ο ρυθμός δημιουργίας των blocks μειώνεται.

Το δίκτυο blockchain σε ένα σύστημα BlockFL παρέχει επίσης ανταμοιβές στις συσκευές, για τα σύνολα των δεδομένων που προσφέρουν, και στους miners για την διαδικασία επικύρωσης των δεδομένων που εκτελούν. Η ανταμοιβή κάθε συσκευής λαμβάνεται από τον αντίστοιχο miner και είναι ανάλογη του μεγέθους του συνόλου των δεδομένων της. Όταν ένας miner ολοκληρώσει πρώτος την διεργασία PoW, τότε θα λάβει την ανταμοιβή του από το δίκτυο blockchain η οποία είναι ανάλογη του μεγέθους των παραμέτρων που έχει συγκεντρώσει. Εδώ σημειώνεται ότι ένα σύστημα BlockFL μπορεί να βελτιωθεί περαιτέρω με την χρήση ενός μηχανισμού επιβράβευσης, ο οποίος πέρα από το μέγεθος του συνόλου των δεδομένων θα λαμβάνει υπόψη και την ποιότητα των δειγμάτων που χρησιμοποιούνται για την εκπαίδευση του κεντρικού μοντέλου η οποία επηρεάζει άμεσα την ακρίβεια του.

Αφού ολοκληρωθεί η διαδικασία πρόσθεσης του block στην αλυσίδα πληροφοριών από τους miners, η κάθε συσκευή θα λάβει αυτό το block από τον αντίστοιχο miner. Στη συνέχεια θα υπολογίσει τις νέες παραμέτρους του κεντρικού μοντέλου χρησιμοποιώντας όλες τις τοπικές παραμέτρους που έχουν συγκεντρωθεί και αποθηκευτεί στο block. Με αυτό τον

τρόπο το κεντρικό μοντέλο FL είναι πολύ ανθεκτικό σε περιπτώσεις δυσλειτουργιών ενός τμήματος του συστήματος ή κάποιας επίθεσης από κακόβουλους χρήστες. Ωστόσο ένα σύστημα BlockFL εξαιτίας της πολυπλοκότητας του, έχει μεγαλύτερο κόστος επικοινωνίας και περισσότερο χρόνο καθυστέρησης (latency) σε σχέση με ένα κλασσικό σύστημα FL. Για την ελαχιστοποίηση του χρόνου καθυστέρησης μειώνεται ο βαθμός δυσκολίας της διεργασίας PoW αυξάνοντας έτσι τον ρυθμό δημιουργίας των blocks.

3.7 Εφαρμογές χρήσης συστημάτων FL

Η ομόσπονδη μάθηση είναι μία καινοτόμος μέθοδος μάθησης η οποία παρέχει την δυνατότητα εκπαίδευσης ενός κεντρικού μοντέλου βαθιάς μάθησης, χρησιμοποιώντας δεδομένα τα οποία είναι αποθηκευμένα σε πολλαπλές συσκευές και διασφαλίζοντας παράλληλα την ιδωτικότητα αυτών των δεδομένων. Η παραπάνω μέθοδος μάθησης εφαρμόζεται ήδη σε τομείς, όπως είναι η προώθηση προϊόντων και η χρηματοοικονομία, στους οποίους η συλλογή δεδομένων είναι μία διαδικασία η οποία είναι δύσκολο να πραγματοποιηθεί εξαιτίας της προστασίας της ιδωτικότητας και της ασφάλειας των δεδομένων. Ένα παράδειγμα εφαρμογής ενός συστήματος FL αποτελεί η έξυπνη λιανική πώληση προϊόντων. Σκοπός της είναι η χρήση μεθόδων μηχανικής μάθησης για την παροχή εξατομικευμένων προϊόντων και υπηρεσιών σε υποψήφιους πελάτες. Τα χαρακτηριστικά των δεδομένων που θα χρησιμοποιηθούν σε αυτή την περίπτωση μπορούν να αφορούν την οικονομική δυνατότητα του πελάτη, τις προσωπικές προτιμήσεις του και τα χαρακτηριστικά που επιθυμεί να έχει ένα προϊόν. Ωστόσο το πιθανότερο είναι τα δεδομένα αυτά να μην βρίσκονται όλα σε μία επιχείρηση αλλά να ανήκουν σε διάφορες εταιρείες. Πιο συγκεκριμένα τα δεδομένα που αφορούν την οικονομική δυνατότητα του πελάτη μπορούν να εξαχθούν από μία τραπεζική επιχείρηση ενώ τα δεδομένα που έχουν σχέση με τις προσωπικές προτιμήσεις του και με τα χαρακτηριστικά του προϊόντος που επιθυμεί, μπορούν να βρίσκονται σε εφαρμογές κοινωνικής δικτύωσης και σε ιστότοπους ηλεκτρονικών αγορών αντίστοιχα, τα οποία είχε επισκεφτεί ο πελάτης στο παρελθόν. Επομένως η συγκέντρωση των δεδομένων σε έναν μεμονωμένο χώρο αποθήκευσης για την εκπαίδευση ενός μοντέλου είναι πολύ δύσκολο να πραγματοποιηθεί. Το παραπάνω πρόβλημα μπορεί να επιλυθεί εφαρμόζοντας την μέθοδο της ομόσπονδης μάθησης για την δημιουργία ενός κεντρικού μοντέλου το οποίο θα εκπαιδεύεται χρησιμοποιώντας τα δεδομένα, με αυτά όμως να παραμένουν εντός των επιχειρήσεων. Με αυτό τον τρόπο διασφαλίζεται η ιδιωτικότητα των δεδομένων αλλά παράλληλα παρέχονται στους πελάτες εξατομικευμένες υπηρεσίες, εξυπηρετώντας έτσι τα συμφέροντα και των δύο πλευρών.

Ο τομέας της υγείας είναι ένας ακόμα τομέας ο οποίος μπορεί να επωφεληθεί σε μεγάλο βαθμό από την εφαρμογή μεθόδων ομόσπονδης μάθησης. Τα ιατρικά δεδομένα τα οποία μπορεί να αφορούν συμπτώματα ασθενών, ιατρικές αναφορές κ.ο.κ είναι απόρρητα και φυλλάσονται σε ιατρικά ιδρύματα και νοσοκομεία, επομένως η συλλογή τους καθίσταται πολύ δύσκολη. Η ανεπάρκεια δεδομένων στον τομέα της υγείας έχει οδηγήσει στην αδυναμία ανάπτυξης μοντέλων τεχνητής νοημοσύνης υψηλών αποδόσεων. Για την αντιμετώπιση αυτού του προβλήματος μπορούν να χρησιμοποιηθούν τεχνικές ομόσπονδης μάθησης οι οποίες θα επιτρέπουν την ανάπτυξη ενός μοντέλου υψηλής αποτελεσματικότητας προστατεύοντας παράλληλα την ιδιωτικότητα των δεδομένων.

Η συλλογή μεγάλων συνόλων δεδομένων καθώς και η προστασία της ιδιωτικότητας αποτελούν μερικές από τις μεγαλύτερες προκλήσεις για το τομέα της τεχνητής νοημοσύνης. Νέες προσεγγίσεις μάθησης δημιουργήθηκαν για την αντιμετώπιση των παραπάνω προκλήσεων με την μέθοδο της ομόσπονδης μάθησης να φαίνεται ότι είναι η αποτελεσματικότερη. Όταν το σύνολο των δεδομένων μίας επιχείρησης δεν είναι σε θέση να παράξει ένα ιδανικό μοντέλο, τότε με την χρήση τεχνικών FL είναι δυνατή η δημιουργία ενός κεντρικού μοντέλου το οποίο διανέμεται σε ένα πλήθος επιχειρήσεων. Οι επιχειρήσεις μοιράζοντας μεταξύ τους τις παραμέτρους που έχουν προκύψει από την εκπαίδευση των τοπικών μοντέλων τους, είναι σε σε θέση να παράξουν ένα κεντρικό μοντέλο FL με μεγαλύτερη ακρίβεια χωρίς να αποκαλύπτουν τα δεδομένα τους. Ένας τρόπος ο οποίος θα προσέλκυε περισσότερους οργανισμούς να πάρουν μέρος σε ένα σύστημα ομόσπονδης μάθησης είναι η διανομή ανταμοιβών στους συμμετέχοντες εξυπηρετώντας έτσι τα συμφέροντα όλων των μερών του συστήματος. Με την συνεχή βελτίωση των συστημάτων ομόσπονδης μάθησης ο τομέας της τεχνητής νοημοσύνης εξελίσσεται και είναι πολύ πιθανό στο κοντινό μέλλον η συλλογή και η ιδωτικότητα των δεδομένων να μην αποτελεί πια πρόκληση στην δημιουργία αυτών των συστημάτων.

4. ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ: ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ ΣΥΣΤΗΜΑΤΟΣ FL

Σε αυτό το κεφάλαιο πραγματοποιείται η δημιουργία ενός συστήματος FL χρησιμοποιώντας την βιβλιοθήκη TensorFlow, η οποία είναι μία βιβλιοθήκη ανοικτού κώδικα που δημιουργήθηκε από την Google και χρησιμοποιείται σε εφαρμογές βαθιάς και μηχανικής μάθησης. Η δημιουργία του κεντρικού μοντέλου FL πραγματοποιείται κάνοντας χρήση της βιβλιοθήκης Keras που πρόκειται για ένα API βαθιάς μάθησης το οποίο έχει κτιστεί πάνω στο TensorFlow. Για την αξιολόγηση και εκπαίδευση του κεντρικού μοντέλου χρησιμοποιήθηκε το σύνολο δεδομένων της MNIST το οποίο αποτελείται συνολικά από 70000 δείγματα ασπρόμαυρων εικόνων των πρώτων 10 ψηφίων (0 έως 9), με τα 60000 από αυτά να αποτελούν τα δείγματα εκπαίδευσης και τα υπόλοιπα 10000 τα δείγματα αξιολόγησης.

Το σύστημα FL θα αποτελείται από ένα νευρωνικό δίκτυο βαθιάς μάθησης, το οποίο είναι το κεντρικό μοντέλο, και από ένα πλήθος συμμετεχόντων (10 συνολικά). Οι συμμετέχοντες του συστήματος (clients) θα λάβουν ένα αντίγραφο του κεντρικού μοντέλου FL και στη συνέχεια θα πραγματοποιήσουν εκπαίδευση στα τοπικά μοντέλα τους. Για την δημιουργία του συστήματος αρχικά πραγματοποιείται η εισαγωγή των δεδομένων εκπαίδευσης και αξιολόγησης και στην συνέχεια δημιουργείται το κεντρικό μοντέλο. Το κεντρικό μοντέλο FL αποτελείται από τρία στρώματα νευρώνων με το επίπεδο εισόδου να διαθέτει 784 νευρώνες, το πλήθος των οποίων είναι ίσο με το σύνολο των εικονοστοιχείων των δειγμάτων. Ο αριθμός των νευρώνων του δεύτερου στρώματος καθορίστηκε αυθαίρετα και είναι ίσος με 128 νευρώνες, ενώ το πλήθος των νευρώνων του επιπέδου εξόδου είναι ίσο με το σύνολο των κλάσεων στις οποίες θα ταξινομηθούν τα δεδομένα από το δίκτυο. Οι διεργασίες που προαναφέρθηκαν πραγματοποιούνται εκτελώντας το παρακάτω τμήμα κώδικα:

```
from keras.datasets import mnist
import random

(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
class_names = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
train_images = train_images / 255.0
test_images = test_images / 255.0

train_images_split = np.array_split(np.array(train_images),10)
train_labels_split = np.array_split(np.array(train_labels),10)
train_labels_adversary_split = np.array(train_labels_adversary, 10)
```

```
average_model = keras.Sequential([
        keras.layers.Flatten(input_shape=(28, 28)),
        keras.layers.Dense(128, activation='relu'),
        keras.layers.Dense(10)])

flmodel = list()
for x in range(10):
      flmodel.append(keras.Sequential([
        keras.layers.Flatten(input_shape=(28, 28)),
        keras.layers.Dense(128, activation='relu'),
        keras.layers.Dense(10)
        ]))
```

4.1 Μετρήσεις και αποτελέσματα

Για την σύνταξη του κεντρικού μοντέλου FL εξετάστηκαν διάφοροι συνδυασμοί αλγορίθμων βελτιστοποίησης (optimizers) και συναρτήσεων ενεργοποίησης. Όλοι οι συνδυασμοί που εξετάστηκαν καθώς και η ακρίβεια που είχε το μοντέλο για τρεις εποχές εκπαίδευσης σε κάθε συνδυασμό φαίνονται στον πίνακα 4.1.

Optimizer	Activation function	Epoch1		Epoch 2		Epoch 3		Batch size
		Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	
SGD	sigmoid	1.8291	0.5442	0.7813	0.8358	0.5559	0.8659	32
	relu	1.0362	0.7385	0.3505	0.9032	0.2971	0.9162	32
RMSprop	sigmoid	0.6143	0.8483	0.2091	0.9400	0.1544	0.9537	32
	relu	0.4199	0.8804	0.1300	0.9619	0.0857	0.9743	32
Adam	sigmoid	0.6914	0.8292	0.2098	0.9396	0.1488	0.9564	32
	relu	0.4339	0.8799	0.1261	0.9629	0.0806	**0.9760**	32

Πίνακας 4.1

Όπως παρατηρείται από τον πίνακα 4.1 η συνάρτηση ενεργοποίησης relu πετυχαίνει μεγαλύτερη ακρίβεια από την σιγμοειδή συνάρτηση για κάθε αλγόριθμο βελτιστοποίησης που εξετάστηκε. Αυτό συμβαίνει διότι η σιγμοειδής συνάρτηση ενεργοποίησης για πολύ μεγάλες τιμές στην έξοδο του κάθε νευρώνα δεν θα εμφανίσει μεγάλες διαφορές στο αποτέλεσμα της καθώς επιστρέφει ένα έυρος τιμών από 0 έως 1. Αντίθετα η συνάρτηση relu θα επιστρέψει στην έξοδο της μία τιμή η οποία θα είναι ίση με την διέργεση του νευρώνα, όταν αυτή είναι θετική, δίνοντας έτσι την δυνατότητα στα συναπτικά βάρη του δικτύου να επηρεάζουν αποτελεσματικότερα την έξοδο του. Επομένως η συνάρτηση ενεργοποίησης που θα εφαρμοστεί στο δίκτυο είναι η relu. Ένα ακόμα συμπέρασμα το οποίο μπορεί να εξαχθεί από το πίνακα 4.1 είναι ότι το δίκτυο, για σταθερή τιμή του batch size, έχει μεγαλύτερη ακρίβεια όταν εφαρμόζονται οι αλγόριθμοι βελτιστοποίησης RMSprop και adam, σε σχέση με τον αλγόριθμο SGD. Οι optimizers RMSprop και adam μπορούν να πετυχαίνουν μεγαλύτερη ακρίβεια καθώς έχουν την δυνατότητα να προσαρμόζουν τον ρυθμό μάθησης κατά την διάρκεια της εκπαίδευσης, σε αντίθεση με τον αλγόριθμο SGD ο οποίος χρησιμοποιεί ένα σταθερό ρυθμό μάθησης σε όλη την διάρκεια της εκπαίδευσης. Ωστόσο με την εφαρμογή του αλγόριθμου βελτιστοποίησης adam επιτυγχάνεται λίγο μεγαλύτερη ακρίβεια σε σχέση με αυτή του αλγόριθμου RMSprop, επομένως ο optimizer που επιλέγεται για την εκπαίδευση του νευρωνικού δικτύου είναι ο adam. Η σύνταξη του κεντρικού μοντέλου FL πραγματοποιείται με την εκτέλεση του παρακάτω τμήματος κώδικα:

```
average_model.compile(optimizer='adam',
loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
metrics=['accuracy'])

for model in flmodel:
      model.compile(optimizer='adam',
      loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=
      True), metrics=['accuracy'])
```

Μία ακόμα παράμετρος η οποία εξετάστηκε κατά την διάρκεια της εκπαίδευσης είναι η παράμετρος batch size. Η τιμή της παραμέτρου batch size καθορίζει το μέγεθος του συνόλου των δεδομένων που θα χρησιμοποιηθεί από το συνολικό πλήθος των δειγμάτων εκπαίδευσης. Όπως φαίνεται στον πίνακα 4.2 για διάφορες τιμές του batch size επηρεάζεται ο χρόνος εκπαίδευσης του δικτύου σε κάθε εποχή καθώς και η ακρίβεια του.

Batch size	Epoch 1			Epoch 2			Epoch 3		
	Time (s)	Loss	Accuracy	Time (s)	Loss	Accuracy	Time (s)	Loss	Accuracy
2 (30000 samples)	44	0.3070	0.9045	44	0.1031	0.9663	43	0.0784	0.9763
32 (1875 samples)	4	0.4207	0.8815	3	0.1215	0.9642	3	0.0793	**0.9769**
100 (600 samples)	2	0.5725	0.8410	2	0.1615	0.9544	2	0.1163	0.9670
1000 (60 samples)	1	1.2739	0.6415	1	0.3344	0.9063	1	0.2576	0.9278

Πίνακας 4.2

Για μικρές τιμές του batch size, δηλαδή για μεγάλα σύνολα δειγμάτων σε κάθε εποχή εκπαίδευσης, ο χρόνος εκπαίδευσης είναι μεγαλύτερος σε σχέση με αυτόν που απαιτείται για υψηλότερες τιμές του batch size. Επίσης όταν η παράμετρος batch size λάβει πολύ μεγάλες τιμές, η εκπαίδευση του δικτύου πραγματοποιείται με πολύ λιγότερα δείγματα και επομένως μειώνεται η ακρίβεια του. Η μέγιστη τιμή στην ακρίβεια του δικτύου επιτυγχάνεται όταν το σύνολο των δεδομένων εκπαίδευσης είναι ίσο με 1875 δείγματα.

Τέλος, όσον αφορά την σύνταξη του κεντρικού μοντέλου, για την εύρεση ενός ρυθμού μάθησης ο οποίος θα είναι κατάλληλος για την εκπαίδευση του δικτύου εξετάστηκαν διάφορες τιμές αυτής της παραμέτρου. Στον πίνακα 4.3 φαίνεται πώς επηρεάζεται η ακρίβεια του δικτύου για διάφορες τιμές του ρυθμού μάθησης.

Learning rate	Loss	Accuracy
1	2.4022	0.1034
0.1	1.4114	0.4933
0.001	0.0453	**0.9863**
0.0001	0.1800	0.9496

Πίνακας 4.3

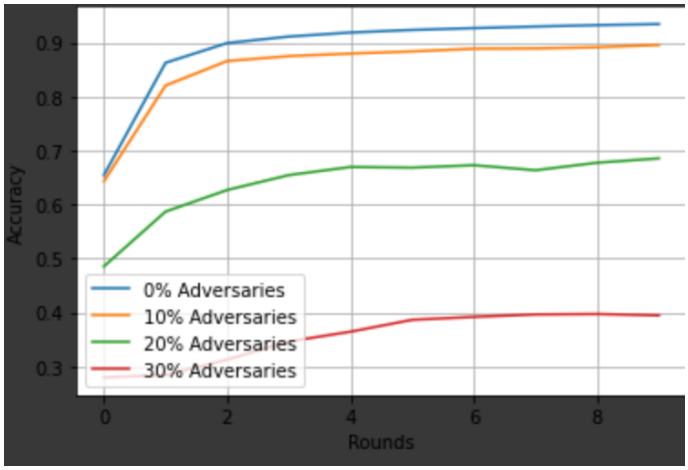
Για σχετικά μεγάλες τιμές του ρυθμού μάθησης η ακρίβεια του δικτύου είναι πολύ χαμηλή. Όπως προαναφέρθηκε στο κεφάλαιο 2 μεγάλες τιμές στον ρυθμό μάθησης ενός νευρωνικού δικτύου έχουν ως αποτέλεσμα μείωση στην ακρίβεια του, διότι δεν μπορεί να προσεγγιστεί σωστά ένα ελάχιστο σημείο της συνάρτησης κόστους. Αντίθετα, όπως φαίνεται και στον πίνακα 4.3, για πολύ μικρές τιμές του ρυθμού μάθησης η προσέγγιση του ελάχιστου σημείου της συνάρτησης κόστους απαιτεί περισσότερο χρόνο εκπαίδευσης του δικτύου, επομένως η ακρίβεια του εμφανίζεται μειωμένη. Η μεγαλύτερη ακρίβεια επιτυγχάνεται όταν η τιμή του ρυθμού μάθησης είναι ίση με 0.001.

Στην συνέχεια με την ολοκλήρωση της σύνταξης του κεντρικού μοντέλου FL, οι συμμετέχοντες του συστήματος θα εκπαιδεύσουν τα τοπικά μοντέλα τους και οι παράμετροι που θα προκύψουν θα συσσωματωθούν για την ανανέωση των παραμέτρων του κεντρικού μοντέλου. Παράλληλα εξετάζεται η περίπτωση στην οποία ένα ποσοστό των μελών του συστήματος αποτελείται από κακόβουλους χρήστες οι οποίοι έχουν ως στόχο την αλλοίωση του κεντρικού μοντέλου FL μεταβάλλοντας ένα μέρος των ετικετών των δειγμάτων εκπαίδευσης. Η αξιολόγηση των τοπικών παραμέτρων των συμμετεχόντων του συστήματος καθώς και η εκπαίδευση των τοπικών μοντέλων πραγματοποιείται από τις συναρτήσεις `evaluatemodel()` και `newround()` αντίστοιχα.

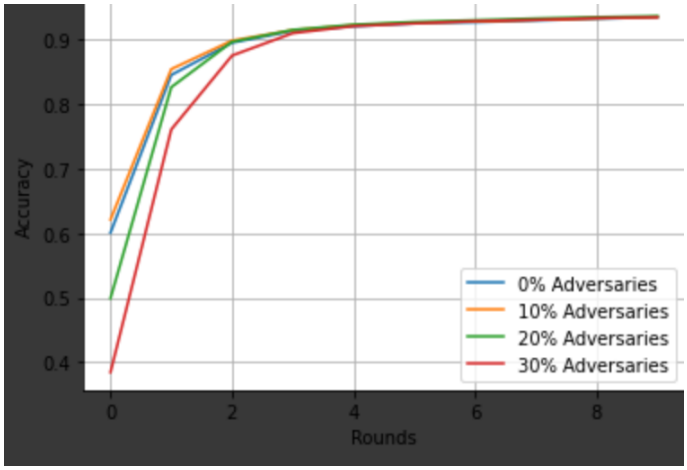
```
def newround (global_model_weights, adversaries):
    for adversary in range(1, adversaries):
        print ("adversary: ", adversary)
        flmodel[adversary].set_weights(global_model_weights)
        flmodel[adversary].fit(train_images_split[adversary],
train_labels_adversary_split[adversary], epochs=1)

    for model in range(adversaries, len(flmodel)):
        print ("honest: ", model)
        flmodel[model].set_weights(global_model_weights)
        flmodel[model].fit(train_images_split[model],
        train_labels_split[model], epochs=1)


def evaluate_model (global_model_weights, model_update):
    global_model.save_weights("pre.h5")

    average_model.set_weights(global_model_weights)
    test_loss,test_acc=global_model.evaluate(test_images,test_labels)

    new_weights2 = [0.8*x + 0.2*y for x, y in
    zip(global_model_weights, model_update)]
    average_model.set_weights(new_weights2)
```

```
test_loss_candidate, test_acc_candidate =
average_model.evaluate(test_images, test_labels)

average_model.load_weights('pre.h5')
if test_acc_candidate>test_acc:
    print ('Model Update is Healthy')
    return True
else:
    print ('Model Update should be discarded')
    return False
```

Αξιολογώντας τις τοπικές παραμέτρους των τοπικών μοντέλων του συστήματος το κεντρικό μοντέλο FL είναι ανθεκτικότερο σε επικείμενες επιθέσεις κακόβουλων χρηστών, καθώς με την εφαρμογή της συνάρτησης `evaluatemodel()` οι παράμετροι οι οποίες μειώνουν την ακρίβεια του μοντέλου απορρίπτονται. Στα παρακάτω διαγράμματα αριστερά φαίνεται η απόδοση του συστήματος όταν αυτό είναι ευάλωτο σε πιθανές επιθέσεις οι οποίες έχουν ως στόχο την αλλοίωση των δειγμάτων εκπαίδευσης, ενώ δεξιά φαίνεται η απόδοση του συστήματος όταν εφαρμόζεται η συνάρτηση αξιολόγησης των παραμέτρων των τοπικών μοντέλων.



Διάγραμμα 5.1: Απόδοση του συστήματος FL χωρίς προστασία



Διάγραμμα 5.2: Απόδοση του συστήματος FL με προστασία

Πίνακας 4.4

4.2 Συμπεράσματα

Η ακρίβεια στην περίπτωση του κλασσικού συστήματος FL μειώθηκε σε ένα πολύ μεγάλο ποσοστό, επειδή δεν εφαρμόστηκε κάποιο είδος προστασίας από επιθέσεις κακόβουλων χρηστών. Αντίθετα το σύστημα FL που δημιουργήθηκε, με την εφαρμογή της προστασίας, ήταν σε θέση να ανταποκριθεί σε ένα αρκετά καλό ποσοστό κακόβουλων χρηστών (έως 30% adversaries), πετυχαίνοντας σχετικά υψηλές τιμές ακρίβειας. Σε αυτό το σημείο σημειώνεται ότι για την προστασία του συστήματος θα μπορούσαν να χρησιμοποιηθούν και άλλες μέθοδοι, σαν αυτές που αναφέρθηκαν στο προηγούμενο κεφάλαιο, οι οποίες θα βελτίωναν περαιτέρω την ασφάλεια του συστήματος.

ΑΝΑΦΟΡΕΣ

[1] Serokel, “Artificial Intelligence vs. Machine Learning vs. Deep Learning: What’s the Difference”, source: https://medium.com/ai-in-plain-english/artificial-intelligence-vs-machine-learning-vs-deep-learning-whats-the-difference-dccce18efe7f, 2020

[2] Ajay Kapoor, “Deep Learning vs Machine Learning: A Simple Explanation”, source: https://hackernoon.com/deep-learning-vs-machine-learning-a-simple-explanation-47405b3eef08, 2020

[3] James Le, “The 8 Neural Network Architectures Machine Learning Researchers Need to Learn”, source: https://www.kdnuggets.com/2018/02/8-neural-network-architectures-machine-learning-researchers-need-learn.html, 2018

[4] “Neural Networks from Scratch”, by Harrison Kinsley & Daniel Kukieła, 2020

[5] “Τεχνητά νευρωνικά δίκτυα”, Κωνσταντίνος Διαμαντάρας, (ΕΚΔΟΣΕΙΣ ΚΛΕΙΔΑΡΙΘΜΟΣ 2007)

[6] Tushar Gupta, “Deep Learning: Feedforward Neural Network”, source: https://towardsdatascience.com/deep-learning-feedforward-neural-network-26a6705dbdc7, 2017

[7] Yash Upadhyay, “Introduction to FeedForward Neural Networks”, source: https://towardsdatascience.com/feed-forward-neural-networks-c503faa46620, 2019

[8] Sumit Saha, “A Comprehensive Guide to Convolutional Neural Networks – the ELI5 way”, source: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53, 2018

[9] Prabhu, “Understanding of Convolutional Neural Networks (CNN) – Deep Learning”, source: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148, 2018

[10] IBM Cloud Education, “Recurrent Neural Networks”, source: https://www.ibm.com/cloud/learn/recurrent-neural-networks, 2020

[11] Mahendran Venkatachalam, “Recurrent Neural Networks Remembering what’s important”, source: https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce, 2019

[12] Pedro Torres Perez, “Deep Learning: Recurrent Neural Networks”, source: https://medium.com/deeplearningbrasilia/deep-learning-recurrent-neural-networks-f9482a24d010, 2018

[13] Jason Brownlee, “Loss and Loss Functions for Training Deep Learning Neural Networks”, source: https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/, 2019

[14] Rohit Dwivedi, “What Are Different Loss Functions Used as Optimizers in Neural Networks?” source: https://www.analyticssteps.com/blogs/what-are-different-loss-functions-used-optimizers-neural-networks, 2020

[15] Jason Brownlee, “How to Choose Loss Functions When Training Deep Learning Neural Networks” source: https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/, 2020

[16] Prince Grover, “5 Regression Loss Functions All Machine Learners Should Know”, source: https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0#:~:text=Mean%20Absolute%20Error%20(MAE)%20is,predictions%2C%20without%20considering%20their%20directions, 2018

[17] PELTARION, “Mean squared logarithmic error (MSLE)”, source: https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/mean-squared-logarithmic-error-(msle), 2020

[18] PELTARION, “Binary Crossentropy”, source: https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/binary-crossentropy, 2020

[19] Niklas Donges, “GRADIENT DESCENT: AN INTRODUCTION TO 1 OF MACHINE LEARNING’S MOST POPULAR ALGORITHMS”, source: https://builtin.com/data-science/gradient-descent, 2019

[20] Abhijit Roy, “An Introduction to Gradient Descent and Backpropagation”, source: https://towardsdatascience.com/an-introduction-to-gradient-descent-and-backpropagation-81648bdb19b2, 2020

[21] Michel Kana, “Do You Understand Gradient Descent and Backpropagation? Most Don’t.”, source: https://medium.com/towards-artificial-intelligence/do-you-understand-gradient-descent-and-backpropagation-most-dont-929d65f57a6c, 2020

[22] “Neural Networks and Deep Learning”, by Michael Nielsen, source: http://neuralnetworksanddeeplearning.com/faq.html, 2019

[23] Hao Zhang, “Intro to Distributed Deep Learning Systems”, source: https://medium.com/@Petuum/intro-to-distributed-deep-learning-systems-a2e45c6b8e7, 2018

[24] Luke Conway, “Blockchain Explained”, source: https://www.investopedia.com/terms/b/blockchain.asp, 2020

[25] QIANG YANG, YANG LIU, TIANJIAN CHEN and YONGXIN TONG, “Federated Machine Learning: Concept and Applications”, arXiv:1902.04885v1 [cs.AI] 13 Feb 2019

[26] Keith Bonawitz et al., “TOWARDS FEDERATED LEARNING AT SCALE: SYSTEM DESIGN”, arXiv:1902.01046v2 [cs.LG] 22 Mar 2019

[27] Hewitt C., Bishop P. B., and Steiger R. “A universal modular ACTOR formalism for artificial intelligence”, In Proceedings of the 3rd International Joint Conference on Artificial Intelligence, Stanford, CA, USA, August 20-23, 1973, pp. 235–245, 1973

[28] Tian Li, Anit Kumar Sahu, Ameet Talwalkar and Virginia Smith, “Federated Learning: Challenges, Methods, and Future Directions”, arXiv:1908.07873v1 [cs.LG] 21 Aug 2019

[29] Nuria Rodr´ıguez-Barroso et al., “Dynamic Federated Learning Model for Identifying Adversarial Clients”, arXiv:2007.15030v1 [cs.LG] 29 Jul 2020

[30] Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim, “Blockchained On-Device Federated Learning”, arXiv:1808.03949v2 [cs.IT] 1 Jul 2019

ΠΗΓΕΣ ΕΙΚΟΝΩΝ

Εικόνα 1.1: https://medium.com/ai-in-plain-english/artificial-intelligence-vs-machine-learning-vs-deep-learning-whats-the-difference-dccce18efe7f

Εικόνα 1.2: https://www.slideshare.net/caiomsouza/pentaho-world-2018-handson-training-tackling-data-science-challenges-with-pdi-friday-october-27th-2017

Εικόνα 1.3: https://medium.com/ai-in-plain-english/artificial-intelligence-vs-machine-learning-vs-deep-learning-whats-the-difference-dccce18efe7f

Εικόνα 1.4: https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html

Εικόνα 1.5: https://www.7wdata.be/big-data/ai-vs-machine-learning-vs-deep-learning/

Εικόνα 2.1: https://freecontent.manning.com/neural-network-architectures/

Εικόνα 2.2: https://www.researchgate.net/figure/Structure-of-Perceptron_fig2_330742498

Εικόνα 2.4: https://learnopencv.com/understanding-feedforward-neural-networks/

Εικόνα 2.5: https://www.mdpi.com/2076-3417/11/1/158

Εικόνα 2.6: https://www.tutorialexample.com/understand-tf-nn-conv2d-compute-a-2-d-convolution-in-tensorflow-tensorflow-tutorial/

Εικόνα 2.7: https://beckernick.github.io/convolutions/

Εικόνα 2.8: https://colab.research.google.com/drive/1FokTzVN4e08pUKXqN2jarnW5Gba-H4PJ?usp=sharing

Εικόνα 2.9: https://en.wikipedia.org/wiki/Recurrent_neural_network#/%20media%20/%20File:Recurrent_neural_network_unfold.svg

Εικόνα 2.10: https://www.slideshare.net/LarryGuo2/chapter-10-170505-l

Εικόνα 2.11: https://vitalflux.com/mean-square-error-r-squared-which-one-to-use/

Εικόνα 2.12: https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/mean-squared-logarithmic-error-(msle)

Εικόνα 2.13: https://www.techjuice.pk/deep-dive-gradient-descent-in-machine-learning/

Εικόνα 2.14: https://medium.com/@iceberg12/overcome-accuracy-limits-in-training-deep-learning-with-cyclical-learning-rate-and-snapshot-f5abce6e9e91

Εικόνα 2.15: https://www.programmersought.com/article/74024489636/

Εικόνα 3.1, εικόνα 3.2: Keith Bonawitz et al., “TOWARDS FEDERATED LEARNING AT SCALE: SYSTEM DESIGN”, arXiv:1902.01046v2 [cs.LG] 22 Mar 2019

Εικόνα 3.3: Chuan Ma et al., “On Safeguarding Privacy and Security in the Framework of Federated Learning”, arXiv:1909.06512v2 [cs.NI] 22 Feb 2020

Εικόνα 3.4: https://www.ig.com/en/trading-strategies/what-is-blockchain-technology--200710

Εικόνα 3.5: Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim, “Blockchained On-Device Federated Learning”, arXiv:1808.03949v2 [cs.IT] 1 Jul 2019

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ