



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΧΡΗΣΗ ΤΕΧΝΗΤΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΓΙΑ ΜΑΘΗΣΗ
ΣΥΝΑΡΤΗΣΕΩΝ

Τζαχρήστας Ευγένιος

Επιβλέπων: Τσούλος Ιωάννης Αναπληρωτής Καθηγητής

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου Ιωάννη Τσούλο για τις συμβουλές και την καθοδήγησή του ώστε να υλοποιηθεί η εργασία. Ακόμη ένα μεγάλο ευχαριστώ στο Τμήμα Πληροφορικής και Τηλεπικοινωνιών με τους εξαιρετικούς καθηγητές που με βοήθησαν στις σπουδές μου. Τέλος την οικογένεια μου που ήταν πάντα δίπλα μου στα φοιτητικά μου χρόνια.

Περίληψη

Η παρούσα εργασία αναφέρεται στη χρήση τεχνητών νευρωνικών δικτύων για μάθηση συναρτήσεων.

Εισάγει τον αναγνώστη στις βασικές έννοιες των τεχνητών νευρωνικών δικτύων και των αλγορίθμων που χρησιμοποιούνται για εκπαίδευσή τους όπως και στις εφαρμογές των τεχνητών νευρωνικών δικτύων.

Αναλύεται η δομή και εξηγείται ο ακριβής τρόπος λειτουργίας στο νευρωνικό δίκτυο MLP από την τροφοδότηση των εισόδων ως και την παραγωγή της εξόδου με τη βοήθεια ενός γενετικού αλγορίθμου. Περιγράφονται κάποιες βασικές λειτουργίες του γενετικού αλγορίθμου που μας βοηθούν να παραχθούν όσο το δυνατόν καλύτερα αποτελέσματα εξόδου.

Επίσης παρουσιάζονται οι συναρτήσεις του κώδικα στην γλώσσα προγραμματισμού C++ και πειραματικά αποτελέσματα σε γνωστά datasets.

Περιεχόμενα

Περίληψη	3
1. Εισαγωγή.....	6
1.1 Μάθηση με επίβλεψη	6
Αναφορές σε βιβλιογραφία	7
1.2 Μάθηση συναρτήσεων	8
1.3 Βελτιστοποίηση συναρτήσεων	11
1.4 Σκοπός της εργασίας	13
2. Τεχνητά νευρωνικά δίκτυα	14
2.1 Τα δίκτυα Perceptron.....	14
2.2 Τα δίκτυα Adaline	21
2.3 Τα δίκτυα MLP	23
2.4 Η μέθοδος Back Propagation.....	25
2.5 Η μέθοδος Gradient Descent.....	29
2.6 Παραδείγματα εφαρμογής νευρωνικών δικτύων	32
Εφαρμογές σε διάφορους τομείς.....	32
3. Γενετικοί αλγόριθμοι	35
3.1 Ιστορική αναδρομή	35
3.2 Μέθοδοι κωδικοποίησης	38
3.3 Γενετικοί τελεστές.....	41
3.4 Παράλληλοι γενετικοί αλγόριθμοι	43
3.5 Εφαρμογές γενετικών αλγορίθμων	45
4. Μέθοδος – αποτελέσματα	46
4.1 Τα dataset που χρησιμοποιήθηκαν	46
4.2 Η προτεινόμενη μέθοδος	51
4.3 Πειραματικά αποτελέσματα.....	57
5. Συμπεράσματα	59

Κατάλογος Εικόνων

Εικόνα 1. Εποπτευόμενη Μάθηση	7
Εικόνα 2 Μη γραμμική συνάρτηση Ενεργοποίησης.....	10
Εικόνα 3. Γενικός τρόπος λειτουργίας αλγορίθμων	11
Εικόνα 4. Νευρώνας ανθρώπινου εγκεφάλου	15
Εικόνα 5. Τεχνητό νευρωνικό δίκτυο Perceptron	16
Εικόνα 6. Είσοδοι του Perceptron και παραγωγή εξόδου.	16
Εικόνα 7. Perceptron με δύο εξόδους	17
Εικόνα 8. Το δίκτυο MLP	23
Εικόνα 9. Gradient Descent.....	30

Κατάλογος Πινάκων

Πίνακας 1. BK δεδομένα.....	57
Πίνακας 2. BL δεδομένα	57
Πίνακας 3. MB δεδομένα.....	57
Πίνακας 4. NT δεδομένα.....	58

1. Εισαγωγή

1.1 Μάθηση με επίβλεψη

Μηχανική μάθηση είναι το υποπεδίο της επιστήμης των υπολογιστών που αναπτύχθηκε από τη μελέτη της αναγνώρισης προτύπων και της υπολογιστικής θεωρίας μάθησης στην τεχνητή νοημοσύνη.

Το 1959, ο Άρθουρ Σάμουελ ορίζει τη μηχανική μάθηση ως «Πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί». Η μηχανική μάθηση διερευνά τη μελέτη και την κατασκευή αλγορίθμων που μπορούν να μαθαίνουν από τα δεδομένα και να κάνουν προβλέψεις σχετικά με αυτά.

Τέτοιοι αλγόριθμοι λειτουργούν κατασκευάζοντας μοντέλα από πειραματικά δεδομένα, προκειμένου να κάνουν προβλέψεις βασιζόμενες στα δεδομένα ή να εξάγουν αποφάσεις που εκφράζονται ως το αποτέλεσμα.

Οι εργασίες μηχανικής μάθησης συνήθως ταξινομούνται σε τρεις μεγάλες κατηγορίες, ανάλογα με τη φύση του εκπαιδευτικού «σήματος» ή την «ανατροφοδότηση» που είναι διαθέσιμα σε ένα σύστημα εκμάθησης. Αυτές είναι:

- **Μάθηση με επίβλεψη (αλλιώς επιτηρούμενη μάθηση) (supervised learnig)**
- **Μη επιβλεπόμενη μάθηση (αλλιώς μη επιτηρούμενη μάθηση) (unsupervised learnig)**
- **Ενισχυτική μάθηση**

Επιβλεπόμενη Μάθηση (Supervised Learning) είναι η διαδικασία όπου ο αλγόριθμος κατασκευάζει μια συνάρτηση που απεικονίζει δεδομένες εισόδους (σύνολο εκπαίδευσης) σε γνωστές επιθυμητές εξόδους, με απώτερο στόχο τη γενίκευση της συνάρτησης αυτής και για εισόδους με άγνωστη έξοδο.

Η Επιβλεπόμενη μάθηση είναι μια κατηγορία μηχανικής μάθησης στόχος της οποίας είναι ο χαρακτηρισμός δεδομένων με βάση κάποια δεδομένα εκπαίδευσης.

Τα δεδομένα εκπαίδευσης αποτελούνται από ένα σύνολο παραδειγμάτων τα οποία χρησιμοποιούνται για εκπαίδευση μοντέλων. Στην επιβλεπόμενη μάθηση, κάθε παράδειγμα αποτελείται από ένα σύνολο εισόδου (συνήθως ένα διάνυσμα από χαρακτηριστικά) και μια επιθυμητή τιμή εξόδου. Οι αλγόριθμοι επιβλεπόμενης μάθησης αναλύουν τα δεδομένα εκπαίδευσης και παράγουν ένα μοντέλο το οποίο μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει νέα παραδείγματα. Το βέλτιστο σενάριο επιτρέπει στον αλγόριθμο να καθορίσει σωστά την ετικέτα της κατηγορίας για άγνωστα μέχρι τώρα παραδείγματα. Για να επιτευχθεί αυτό, απαιτείται

ο αλγόριθμος μάθησης να γενικεύει από τα δεδομένα εκπαίδευσης σε αθέατες καταστάσεις με ένα «λογικό» τρόπο.

Στη μάθηση με επίβλεψη το σύστημα πρέπει να «μάθει» επαγωγικά μια συνάρτηση που ονομάζεται συνάρτηση στόχος (target function) και αποτελεί έκφραση του μοντέλου που περιγράφει τα δεδομένα. Η συνάρτηση στόχος χρησιμοποιείται για την πρόβλεψη τιμής μιας μεταβλητής, που ονομάζεται εξαρτημένη μεταβλητή ή μεταβλητή εξόδου, βάσει των τιμών ενός συνόλου μεταβλητών, που ονομάζονται ανεξάρτητες μεταβλητές ή μεταβλητές εισόδου ή χαρακτηριστικά.

Η επιβλεπόμενη μάθηση χρησιμοποιείται συνήθως σε προβλήματα:

- Ταξινόμησης (Classification)
- Πρόγνωσης (Prediction)
- Διερμηνείας (Interpretation)

Η μάθηση με επίβλεψη χωρίζεται σε δύο κατηγορίες: **στη δομική (structural)** και στη **προσωρινή (temporal)** εκμάθηση. Οι αλγόριθμοι οι οποίοι βρίσκονται στη πρώτη κατηγορία χρησιμοποιούνται για την εύρεση της βέλτιστης σχέσης μεταξύ εισόδων και εξόδων για κάθε ξεχωριστό ζευγάρι προτύπων. Παραδείγματα της δομικής μάθησης αποτελούν η αναγνώριση και η κατηγοροποίηση προτύπων, ενώ παραδείγματα της προσωρινής εκμάθησης η πρόβλεψη και ο έλεγχος.

Παράδειγμα εποπτευόμενης μάθησης

Στο πρώτο βήμα, ένα σύνολο δεδομένων εκπαίδευσης τροφοδοτείται στον αλγόριθμο μηχανικής μάθησης.

Με το σύνολο δεδομένων εκπαίδευσης, ο αλγόριθμος προσαρμόζεται, κάνοντας αλλαγές στις παραμέτρους για τη δημιουργία ενός λογικού μοντέλου.



Εικόνα 1. Εποπτευόμενη Μάθηση

Αναφορές σε βιβλιογραφία

Ως επιστημονικό εγχείρημα, η μηχανική μάθηση αναπτύχθηκε από την αναζήτηση για την τεχνητή νοημοσύνη.

Ήδη από την πρώιμη περίοδο της έρευνας στον τομέα της τεχνητής νοημοσύνης σε ακαδημαϊκό, το ζήτημα της κατασκευής μηχανών που θα μάθαιναν από δεδομένα απασχόλησε τους ερευνητές.

Ένας ορισμός για την μηχανική μάθηση δόθηκε από τον Arthur Samuel το 1959 ως «ένα πεδίο μελέτης που δίνει στους ηλεκτρονικούς υπολογιστές την δυνατότητα να μαθαίνουν χωρίς να έχουν ρητά προγραμματιστεί».

Ο **Tom M. Mitchell** πρότεινε έναν επίσημο ορισμό που χρησιμοποιείται ευρέως:

« Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από εμπειρία E ως προς μια κλίση εργασιών T και ένα μέτρο επίδοσης P , αν η επίδοσή του σε εργασίες κλάσης T , όπως αποτιμάται από το μέτρο P , βελτιώνεται με την εμπειρία E » ακολουθώντας έτσι την πρόταση του Alan Turing στην εργασία του «Υπολογιστικές μηχανές και Νοημοσύνη».

Η ιδέα της επιβλεπόμενης μάθησης διατυπώθηκε από τους **Scudder (1965), Fraclick (1967), Agrawala (1970)** και άλλους.

Ο **Vapnik**^[1] διατύπωσε την ιδέα του μεταγωγικού συμπεράσματος ή μεταγωγής, η οποία θεωρείται ο πρόγονος της μάθησης με ημι-επίβλεψη. Νωρίτερα, ο **Hartley (1968)** είχε εισάγει την ιδέα του μεταγωγικού συμπεράσματος.

Τη δεκαετία του 1970 με την θεώρηση του προβλήματος της εκτίμησης του κανόνα γραμμικότητας της διακρίνουσας Fisher με μη ταξινομημένα δεδομένα, δόθηκε ώθηση στην μάθηση με ημι – επιτήρηση (**Hosmer 1973, McLachlan 1977, O' Neill 1978, McLachlan και Ganesalingam 1982**).

1. Vapnik ant Chervonenkis, 1974, Vapnik ant Sterin,1997

1.2 Μάθηση συναρτήσεων

Ως μάθηση μπορεί να οριστεί η σταδιακή βελτίωση της ικανότητας του δικτύου να επιλύει κάποιο πρόβλημα (π.χ. η σταδιακή προσέγγιση μίας συνάρτησης).

Η μάθηση επιτυγχάνεται μέσω της **εκπαίδευσης**, μίας επαναληπτικής διαδικασίας σταδιακής προσαρμογής των παραμέτρων του δικτύου συνήθως των βαρών και της πόλωσής του σε τιμές κατάλληλες ώστε να επιλύεται με επαρκή επιτυχία το προς εξέταση πρόβλημα. Αφού ένα δίκτυο εκπαιδευτεί, οι παράμετροί του συνήθως «παγώνουν» στις κατάλληλες τιμές και από εκεί κι έπειτα είναι σε λειτουργική κατάσταση. Το ζητούμενο είναι το λειτουργικό δίκτυο να χαρακτηρίζεται από μία

ικανότητα **γενίκευσης**: αυτό σημαίνει πως δίνει ορθές εξόδους για εισόδους καινοφανείς και διαφορετικές από αυτές με τις οποίες εκπαιδεύτηκε.

Επιβλεπόμενη ή επαγωγική μάθηση

- Το σύστημα μαθαίνει μια συνάρτηση f από παραδείγματά της. Η συνάρτηση λέγεται συνάρτηση στόχου.
- Ένα παράδειγμα είναι ένα ζεύγος $x, f(x)$, όπου x είναι η είσοδος και $f(x)$ η έξοδος της συνάρτησης.
- Το πρόβλημα: βρες μια υπόθεση h που προσεγγίζει την f , δοθέντος ενός συνόλου παραδειγμάτων (σύνολο εκπαίδευσης). Μια υπόθεση είναι συνεπής με τα παραδείγματα αν συμφωνεί με την f σε όλα τα παραδείγματα.
- Αυτή είναι μια υπεραπλουστευτική εκδοχή της πραγματικής μάθησης.

Συναρτήσεις ενεργοποίησης

Η συνάρτηση ενεργοποίησης μπορεί να είναι **βηματική** (step transfer function), **γραμμική** (linear transfer function), **μη γραμμική** (non-linear transfer function), **στοχαστική** (stochastic transfer function).

Βηματική συνάρτηση ενεργοποίησης

Η βηματική συνάρτηση ενεργοποίησης μπορεί να είναι:

$$f(x)=1 \text{ όταν } x \geq 0$$

$$f(x)=0 \text{ όταν } x < 0$$

ή οποιαδήποτε άλλη βηματική συνάρτηση.

Γραμμική συνάρτηση ενεργοποίησης

Η γραμμική συνάρτηση ενεργοποίησης μπορεί να είναι:

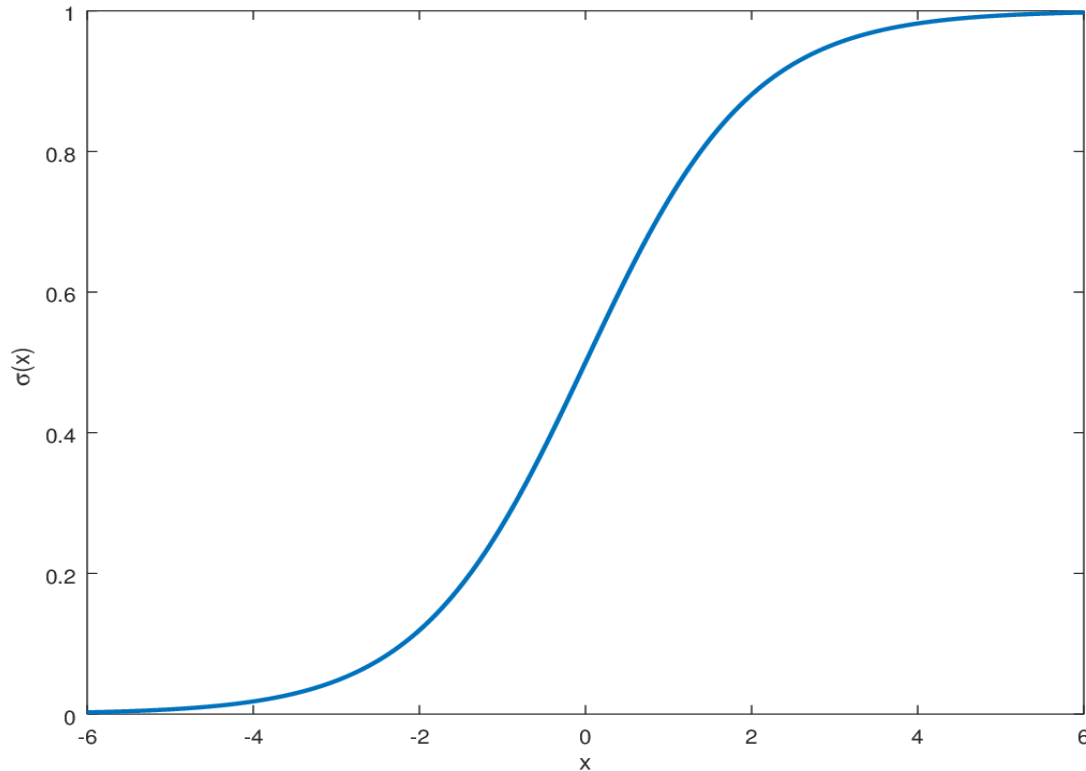
$$f(x)=x$$

ή οποιαδήποτε άλλη γραμμική συνάρτηση.

Μη γραμμική συνάρτηση ενεργοποίησης

Η μη γραμμική συνάρτηση ενεργοποίησης που χρησιμοποιείται συνήθως στα νευρωνικά δίκτυα καλείται σιγμοειδής συνάρτηση. Οι τυπικές σιγμοειδείς είναι δύο:

1. Λογιστική σιγμοειδής: $f(x) = 1/(1+e^{-x})$

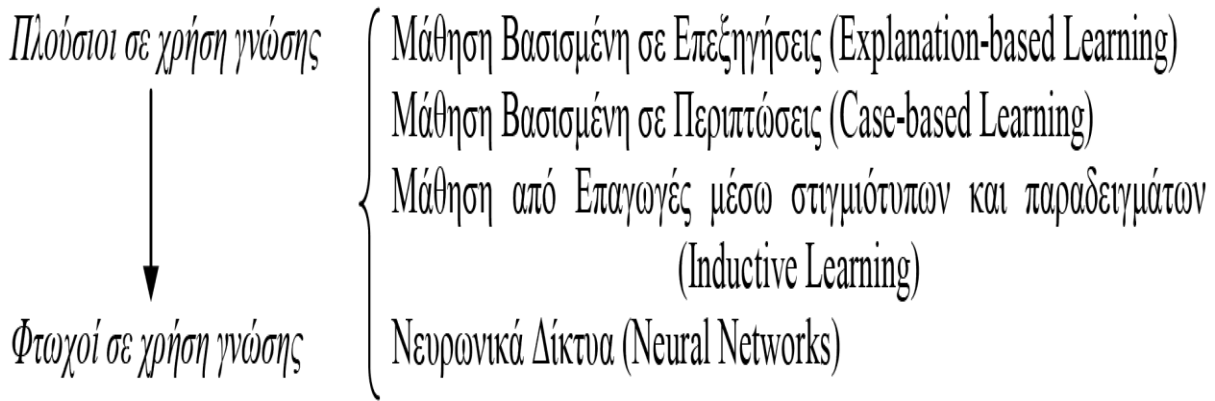


Εικόνα 2 Μη γραμμική συνάρτηση Ενεργοποίησης

2. Υπερβολική εφαπτομένη: $f(x) = \tanh x$

Όλοι οι αλγόριθμοι Μηχανικής Μάθησης διαχειρίζονται τη γνώση αναπαριστώντας την με κάποιον τρόπο.

Ορισμένοι αλγόριθμοι δέχονται ως είσοδο μόνο παρατηρήσεις και άλλοι λαμβάνουν υπόψη τους λίγο ή περισσότερο την προϋπάρχουσα γνώση. Μια προσπάθεια κατάταξης των αλγορίθμων με κριτήριο τον τρόπο μάθησης βασισμένο περισσότερο ή λιγότερο στην υπάρχουσα γνώση δίνεται παρακάτω:



Εικόνα 3. Γενικός τρόπος λειτουργίας αλγορίθμων

Στην εικόνα 3, αποτυπώνεται ο γενικός τρόπος λειτουργίας των αλγορίθμων Μηχανικής Μάθησης. Η βασικότερη φάση κάθε αλγόριθμου είναι η εκπαίδευση, όπου ο αλγόριθμος χρησιμοποιεί ως είσοδο ένα **σύνολο δεδομένων εκπαίδευσης** (training set) προς επίτευξη του σκοπού του, τη δημιουργία νέας γνώσης. Επιπλέον, μπορεί είτε να χρησιμοποιήσει λιγότερο ή περισσότερο την υπάρχουσα γνώση είτε να μην τη χρησιμοποιήσει καθόλου.

Την εκπαίδευση ακολουθεί η φάση της πιστοποίησης της παραγόμενης νέας γνώσης. Συνήθως, η πιστοποίηση πραγματοποιείται καταρχάς από τον ίδιο τον αλγόριθμο μέσω διαδικασιών ανάκλησης (recall) με τη βοήθεια **δεδομένων ελέγχου** (test data) και, στη συνέχεια, μέσω κριτικής που κάνει ο χρήστης βάσει των γνώσεων που διαθέτει για το πρόβλημα που επιχειρεί να λύσει ο αλγόριθμος. Τέλος, η νέα γνώση δίνεται προς χρήση σε εφαρμογές στις οποίες είναι απαραίτητη, για να λυθούν πραγματικά προβλήματα.

1.3 Βελτιστοποίηση συναρτήσεων

Οι τρόποι βελτιστοποίησης συνάρτησης νευρωνικών δικτύων είναι οι μέθοδοι που χρησιμοποιούνται για τη βελτιστοποίηση των παραμέτρων ενός μοντέλου νευρωνικού δικτύου κατά τη διαδικασία εκπαίδευσής του. Ο στόχος είναι να βελτιωθεί η απόδοση του μοντέλου στα δεδομένα εκπαίδευσης και στα δεδομένα επαλήθευσης, και να γενικευτεί σε νέα δεδομένα.

Οι αλγόριθμοι αυτόματης βελτιστοποίησης των υπερπαραμέτρων επιτρέπουν την επιλογή υπερπαραμέτρων χωρίς την ανθρώπινη παρέμβαση. Οι αλγόριθμοι αυτοί έχουν ως κοινό τους το γεγονός πως θεωρούν το μοντέλο, του οποίου τις υπερπαραμέτρους επιλέγουν, ως ένα "μαύρο κουτί", δηλαδή δεν ενδιαφέρονται για την εσωτερική λειτουργία του παρά μόνο για την επιρροή που ασκούν οι υπερπαραμέτροι που επιλέγονται στην αντικειμενική απόδοση του. Θεωρείται

δηλαδή το Νευρωνικό Δίκτυο ως μια αντικειμενική συνάρτηση την οποία προσπαθούμε να βελτιστοποιήσουμε με συστηματικό τρόπο.

Η βελτιστοποίηση αναφέρεται σε κάποια μετρική που σχετίζεται με το πεδίο του προβλήματος το οποίο λύνει το Νευρωνικό Δίκτυο. Μπορεί να είναι το σφάλμα στις προβλέψεις του, που είναι το συνηθέστερο, αλλά μπορεί να είναι και ένας συνδυασμός σφάλματος με υπολογιστικό κόστος ή κάποια άλλη υβριδική μετρική. Οπότε, μπορούμε να θεωρήσουμε την αντικειμενική συνάρτηση ως μια συνάρτηση που δέχεται στην είσοδο της έναν συνδυασμό υπερπαραμέτρων και επιστρέφει στην έξοδο της μια μετρική της απόδοσης του Νευρωνικού Δικτύου για τις συγκεκριμένες τιμές των υπερπαραμέτρων.

Υπάρχουν δύο είδη αλγορίθμων βελτιστοποίησης υπερπαραμέτρων.

Οι αλγόριθμοι αναζήτησης και οι αλγόριθμοι βελτιστοποίησης.

Στους αλγορίθμους αναζήτησης ανήκουν αλγόριθμοι όπως η **τυχαία αναζήτηση (Random Search)** και η **αναζήτηση πλέγματος (Grid Search)**. Το χαρακτηριστικό τους είναι ότι αναζητούν την λύση με "αφελή" τρόπο, δηλαδή χωρίς την χρήση κάποιας ευρετικής μεθόδου που να βελτιώνει την ποιότητα των προτεινόμενων λύσεων σταδιακά. Το χαρακτηριστικό αυτό τους καθιστά μη αποδοτικούς, ιδιαίτερα για μεγάλο αριθμό υπερπαραμέτρων. Στα πλεονεκτήματά τους συμπεριλαμβάνονται η ευκολία με την οποία μπορούν να υλοποιηθούν, καθώς και η εγγενής ευκολία παραλληλοποίησης τους.

Στους αλγορίθμους βελτιστοποίησης ανήκουν αλγόριθμοι όπως η **Bayesian βελτιστοποίηση**, οι **γενετικοί αλγόριθμοι**, η **βελτιστοποίηση σμήνους σωματιδίων (Particle Swarm Optimization)** και **άλλοι**. Κοινό τους χαρακτηριστικό είναι η επιλογή της κάθε προτεινόμενης λύσης με βάση προηγούμενες πληροφορίες που έχει συλλέξει ο αλγόριθμος. Οι αλγόριθμοι αυτοί είναι πιο σύνθετοι και η υλοποίησή τους αντίστοιχα πιο πολύπλοκη.

Μερικοί από τους τρόπους βελτιστοποίησης που μπορούν να χρησιμοποιηθούν είναι:

1. **Στοχαστική Κατάβαση Κλίσης (Stochastic Gradient Descent, SGD)**: Η μέθοδος αυτή αποτελεί τη βασική μέθοδο βελτιστοποίησης και λειτουργεί εντοπίζοντας την κατεύθυνση με τη μεγαλύτερη κλίση και κινούμενοι προς αυτήν την κατεύθυνση. Η στοχαστική κατάβαση κλίσης ανανεώνει τις παραμέτρους του μοντέλου σε κάθε επανάληψη του αλγορίθμου εκπαίδευσης, βασιζόμενη στην κλίση συνάρτησης απώλειας στο παραδειγματολόγιο εκπαίδευσης.
2. **Στοχαστική Κατάβαση Κλίσης με Αδράνεια (Stochastic Gradient Descent with Momentum, SGD+)**

3. **To Gradient descent** είναι ο πιο συνηθισμένος αλγόριθμος για τη βελτιστοποίηση του μοντέλου για την ελαχιστοποίηση του σφάλματος. Προκειμένου να πραγματοποιηθεί η κατάβαση κλίσης, πρέπει να επαναλαμβάνεται το σύνολο δεδομένων εκπαίδευσης ενώ ρυθμίζεται εκ νέου το μοντέλο. Ο στόχος είναι να ελαχιστοποιηθεί η συνάρτηση κόστους για να λαμβάνει το μικρότερο δυνατό σφάλμα και βελτιώνει την ακρίβεια του μοντέλου.
4. Η εξαντλητική αναζήτηση ή αναζήτηση ωμής βίας είναι η διαδικασία αναζήτησης των βέλτιστων υπερπαραμέτρων ελέγχοντας εάν κάθε υποψήφιος ταιριάζει. Η εξαντλητική μέθοδος αναζήτησης είναι απλή. Για παράδειγμα, εάν εργάζεστε με έναν αλγόριθμο k-means, θα αναζητήσετε με μη αυτόματο τρόπο τον σωστό αριθμό συμπλεγμάτων. Ωστόσο, αν υπάρχουν εκατοντάδες και χιλιάδες επιλογές που πρέπει να εξετάσετε, γίνεται αφόρητα βαρύ και αργό. Αυτό καθιστά την αναζήτηση ωμής βίας αναποτελεσματική στην πλειονότητα των πραγματικών

1.4 Σκοπός της εργασίας

Σκοπός της εργασίας είναι η μελέτη των νευρωνικών δικτύων καθώς και η ανάλυση και επεξήγηση των γενετικών αλγορίθμων και τη χρήση τους στην εκπαίδευση των νευρωνικών δικτύων.

Αυτό πραγματοποιείται μέσω ιστορικής αναδρομής από την ανάλογη βιβλιογραφία καθώς και επεξήγηση διάφορων παραλλαγών και ιδιοτήτων τόσο των νευρωνικών δικτύων όσο και των γενετικών αλγορίθμων.

Επίσης παρουσιάζονται τα βήματα στη χρήση των γενετικών αλγορίθμων στην εκπαίδευση των νευρωνικών δικτύων και την εφαρμογή τους στον επιστημονικό τομέα.

Δημιουργήθηκε μια εφαρμογή χρήσης αλγορίθμου για την εκπαίδευση νευρωνικού δικτύου με εισαγωγή δεδομένων και έλεγχο στην αξιοπιστία παραγωγής αποτελεσμάτων.

2. Τεχνητά νευρωνικά δίκτυα

2.1 Τα δίκτυα Perceptron

Τα νευρωνικά δίκτυα αποτελούν σημαντικό μέρος της μηχανικής μάθησης, επομένως είναι απαραίτητο να κατανοήσουμε πώς λειτουργούν. Ένα νευρωνικό δίκτυο είναι ένα σύστημα υπολογιστή που έχει μοντελοποιηθεί με βάση ένα βιολογικό νευρωνικό δίκτυο που περιλαμβάνει νευρώνες συνδεδεμένους μεταξύ τους. Μπορεί να κατασκευαστεί για να επιλύει εργασίες μηχανικής μάθησης, όπως προβλήματα ταξινόμησης και παλινδρόμησης.

Ο αλγόριθμος **perceptron** είναι μια αναπαράσταση του τρόπου λειτουργίας των νευρωνικών δικτύων. Οι τεχνητοί νευρώνες προτάθηκαν για πρώτη φορά από τον **Frank Rosenblatt** το 1957 ως μοντέλα για τον μηχανισμό αντίληψης του ανθρώπινου εγκεφάλου.

Αν και υπήρχαν ήδη αλγόριθμοι μηχανικής μάθησης προηγούμενης γενιάς, όπως οι αλγόριθμοι **Bayes** και **perceptron-style algorithms**, το **Perceptron** θεωρείται ως ο πρώτος αλγόριθμος μηχανικής μάθησης που μπορούσε να «εκπαιδευτεί» να αναγνωρίζει πρότυπα σε δεδομένα εισόδου.

Ο αλγόριθμος **Perceptron** είναι ένας γραμμικός ταξινομητής που μπορεί να εκπαιδευτεί σε ένα σύνολο εισόδων (inputs) και στη συνέχεια να παράγει έναν αντίστοιχο συναφή έξοδο (output). Εάν η έξοδος δεν είναι επιθυμητή, τότε ο αλγόριθμος Perceptron προσαρμόζεται ώστε να βελτιώσει την ακρίβεια του.

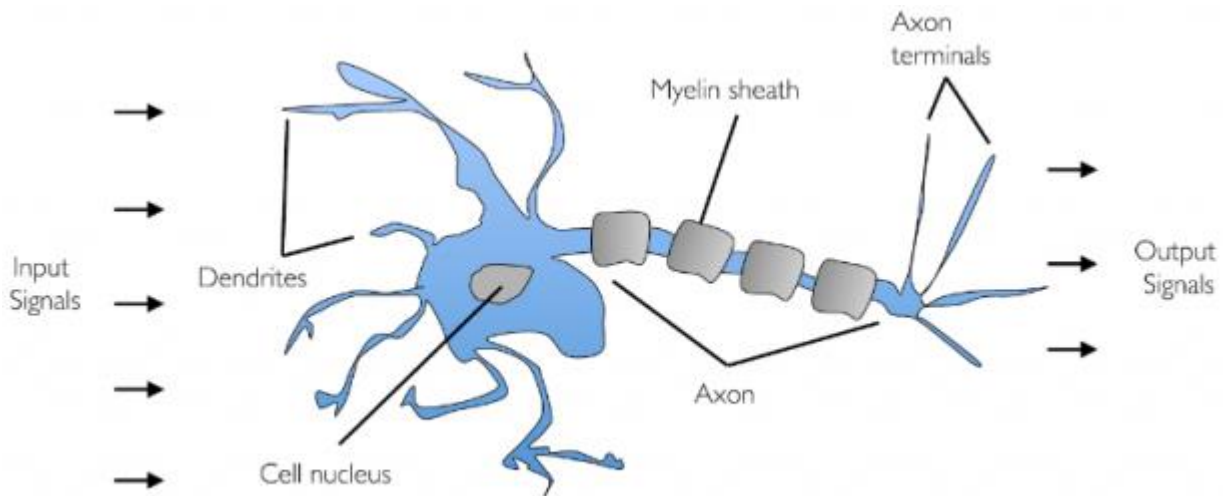
Το Perceptron αποτέλεσε τη βάση για πολλούς αλγορίθμους μηχανικής μάθησης που ακολούθησαν, όπως οι **δικτυακοί παλμοί (neural networks)** και οι **μονάδες επεξεργασίας γραφημάτων (graphical processing units)**. Ως εκ τούτου, το Perceptron θεωρείται ως ένα από τα πιο σημαντικά εργαλεία στην ιστορία της μηχανικής μάθησης και συνεχίζει να εφαρμόζεται.

Το μοντέλο του perceptron αποτελούνταν από τρία κύρια στοιχεία:

Κάθε ένα από αυτά τα τρία κύρια συστήματα θα περιέχει επιπλέον μονάδες που συνδέονται. Αυτές οι συνδέσεις μπορούν να ενεργοποιηθούν ή να απενεργοποιηθούν με βάση το μοτίβο που αναγνωρίζεται. Το αισθητήριο σύστημα θα έπαιρνε το μοτίβο εισόδου. Το σύστημα συσχέτισης θα ενεργοποιούσε ή θα απενεργοποιούσε συγκεκριμένες συνδέσεις και η έξοδος θα παρουσιαζόταν χρησιμοποιώντας το σύστημα απόκρισης.

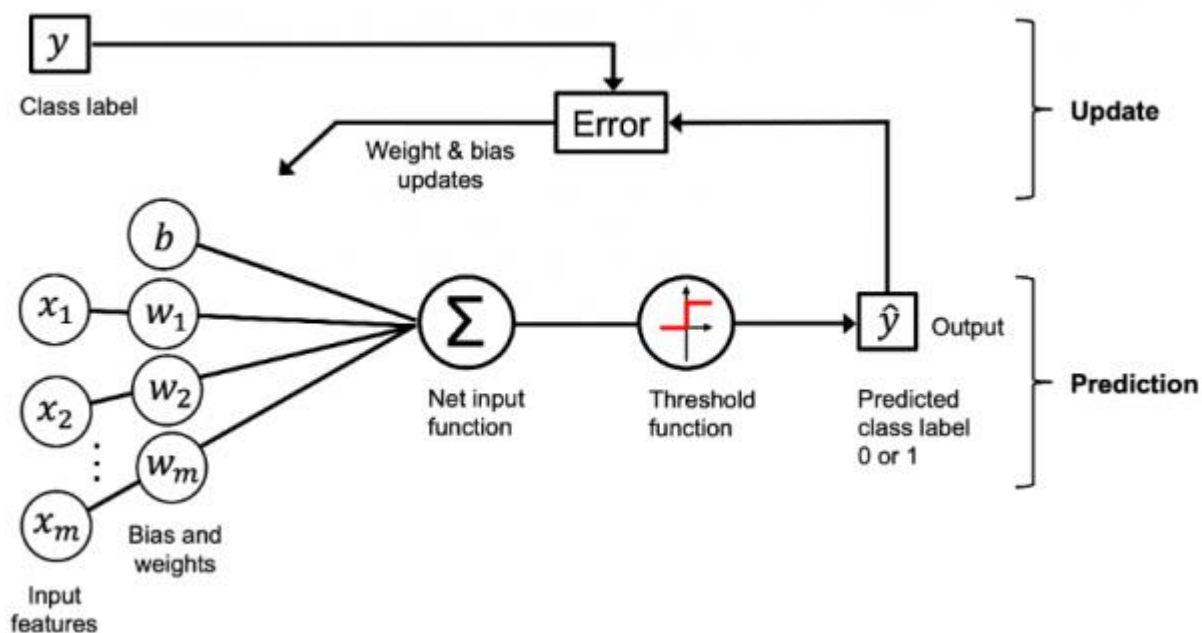
Ένα **perceptron** είναι η πιο θεμελιώδης μονάδα που χρησιμοποιείται για την κατασκευή ενός νευρωνικού δικτύου. Ένα perceptron μοιάζει με έναν νευρώνα στον ανθρώπινο εγκέφαλο. Στην περίπτωση ενός βιολογικού νευρώνα, πολλαπλά σήματα

εισόδου τροφοδοτούνται σε έναν νευρώνα μέσω δενδρίτη και ένα σήμα εξόδου εκπέμπεται κατάλληλα με βάση την ισχύ του σήματος εισόδου και κάποιου άλλου μηχανισμού. Το παρακάτω διάγραμμα αντιπροσωπεύει τον τρόπο με τον οποίο τα σήματα εισόδου περνούν μέσω του βιολογικού νευρώνα.



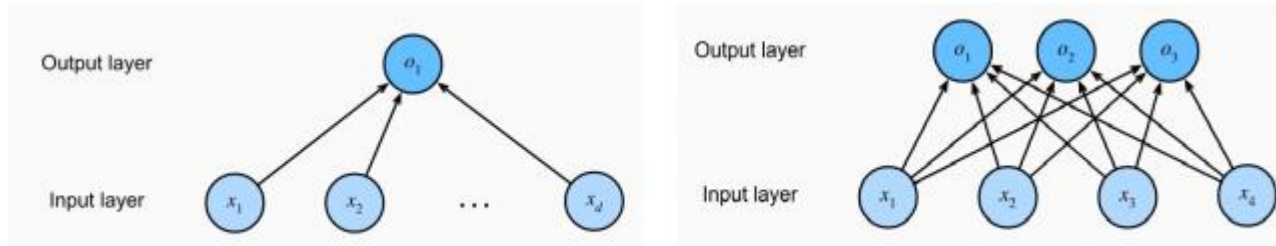
Εικόνα 4. Νευρώνας ανθρώπινου εγκεφάλου

Στην περίπτωση ενός perceptron, τα σήματα εισόδου συνδυάζονται με διαφορετικά βάρη και τροφοδοτούνται στον τεχνητό νευρώνα ή perceptron μαζί με ένα στοιχείο προκατάληψης. Αντιπροσωπευτικά, εντός του perceptron, το καθαρό άθροισμα υπολογίζεται ως το άθροισμα των βαρών και του σήματος εισόδου, και ως στοιχείο πόλωσης, στη συνέχεια, το καθαρό άθροισμα τροφοδοτείται σε μια μη γραμμική συνάρτηση ενεργοποίησης. Με βάση τη λειτουργία ενεργοποίησης, το σήμα εξόδου αποστέλλεται. Το παρακάτω διάγραμμα αντιπροσωπεύει ένα perceptron. Το στοιχείο πόλωσης b και το άθροισμα των βαρών και των σημάτων εισόδου που αντιπροσωπεύονται χρησιμοποιώντας x και w . Η συνάρτηση κατωφλίου αντιπροσωπεύει τη μη γραμμική συνάρτηση ενεργοποίησης.



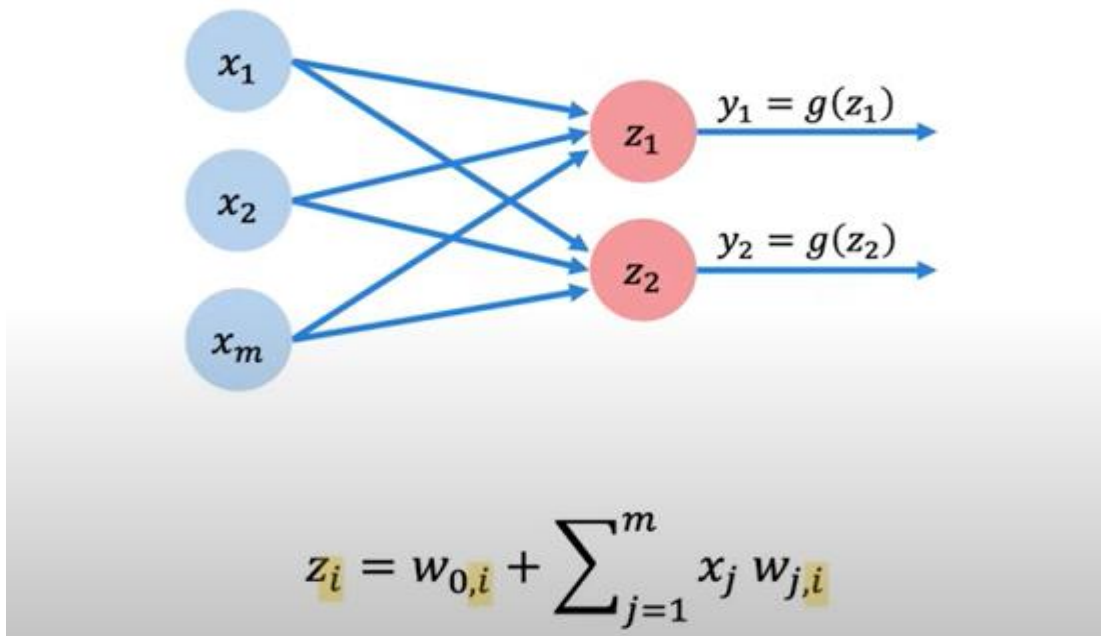
Εικόνα 5. Τεχνητό νευρωνικό δίκτυο Perceptron

Τα **perceptrons** που εκτίθενται με τη μορφή νευρωνικών δικτύων μονής ή πολλαπλών επιπέδων μπορούν να χρησιμοποιηθούν για την εκτέλεση εργασιών παλινδρόμησης και ταξινόμησης. Το παρακάτω διάγραμμα αντιπροσωπεύει το νευρωνικό δίκτυο μονής στρώσης (perceptron), τη γραμμική παλινδρόμηση (αριστερά) και την παλινδρόμηση softmax (κάθε έξοδος y_1, y_2 και y_3)



Εικόνα 6. Είσοδοι του Perceptron και παραγωγή εξόδου.

Ακολουθεί ένα παράδειγμα ενός perceptron πολλαπλών εξόδων.



Εικόνα 7. Perceptron με δύο εξόδους

Η εκπαίδευση ενός perceptron είναι ένα πρόβλημα βελτιστοποίησης που περιλαμβάνει την επαναληπτική ενημέρωση των βαρών με τρόπο που ελαχιστοποιεί τη συνάρτηση σφάλματος. Εξάγαμε τη συνάρτηση σφάλματος και ορίσαμε τι πρέπει να βασίζεται ένα ενημερωμένο βάρος με βάση το τρέχον βάρος και το σφάλμα που υπολογίζεται στην τρέχουσα επανάληψη.

Εκπαίδευση τεχνητού νευρωνικού δικτύου Perceptron

Θεωρούμε το σύνολο εκπαίδευσης $X = \{(x^i, t(x^i))\} \quad i=1, \dots, N$. Τα x^i είναι τα δεδομένα εκπαίδευσης και $x^i = [x^i_1, \dots, x^i_d]$ δηλαδή $x^i \in \mathcal{R}^d$. Το $t(x^i)$ είναι η κατηγορία του δεδομένου x^i . Με την εκπαίδευση επιθυμούμε να βρούμε τις κατάλληλες τιμές για τα βάρη ώστε το υπερεπίπεδο που αντιστοιχεί σε αυτά να

διαχωρίζει ορθά όλα τα δεδομένα του συνόλου εκπαίδευσης, δηλαδή $y(x_i) = t(x_i)$ $\forall X \in \mathcal{X}$

Επαυξημένο διάνυσμα δεδομένων \square Στο διάνυσμα όλων των δεδομένων του συνόλου εκπαίδευσης προσθέτουμε και τη συνιστώσα $x_0 = 1$ που αντιστοιχεί στην είσοδο με τιμή 1 του perceptron \square Το επαυξημένο διάνυσμα για το x_i είναι: $x_i^e = [x_{i0} = 1, x_{i1}, \dots, x_{id}]$ \square Παρέχει ευκολία στη διατύπωση του αλγορίθμου εκπαίδευσης

Βήμα 1-Αρχικοποίηση Αρχικοποίησε τα βάρη του perceptron $w(0) = [w_0(0), w_1(0), \dots, w_d(0)]$ Η αρχικοποίηση γίνεται συνήθως τυχαία π.χ. επιλέγοντας τιμές στο διάστημα $(-1, 1)$

Βήμα 2-Ενημέρωση βαρών

Θέσε ως είσοδο τα πρότυπα x_i του συνόλου εκπαίδευσης (ένα την φορά)

Για κάθε x_i υπολόγισε το $y(x_i)$ και άλλαξε τα βάρη με βάση τον εξής κανόνα (ο κανόνας περιέχει πράξεις με διανύσματα!):

Στη συνέχεια προχώρησε στο επόμενο πρότυπο

Όταν τα πρότυπα εξαντληθούν άρχισε από την αρχή

Κάθε κυκλική επανάληψη όλων των προτύπων

Τερματισμός

Επανάλαβε το βήμα 2 εωσότου όλα τα πρότυπα ταξινομηθούν σωστά, δηλαδή $y(x_i) = t(x_i)$ $\forall X \in \mathcal{X}$

Το παραπάνω είναι ισοδύναμο με το να μη γίνει αλλαγή στα βάρη σε μία εποχή

Ρυθμός μάθησης η

Πρόκειται για σταθερά του αλγορίθμου εκπαίδευσης

Καθορίζει πόσο έντονα μεταβάλλονται τα βάρη σε κάθε ανανέωση

Ισχύει: $\eta \in (0, 1]$ και $\eta > 0$

Η σύγκλιση επηρεάζεται από το ρυθμό μάθησης

Βιβλιογραφία

Το πρώτο μοντέλο νευρωνικού δικτύου το παρουσιάστηκε το 1943 από τους **McCulloch και Pitts**. Σε μία πρώτη εργασία τους παρουσίασαν για πρώτη φορά την ιδέα ότι ένα νευρωνικό δίκτυο αποτελείται από μία συλλογή ενός μεγάλου αριθμού νευρώνων, και έδειξαν πως θα μπορούσαν να λειτουργούν οι νευρώνες με τις διασυνδέσεις τους. Αυτή θεωρείται ιστορικά ότι είναι η πρώτη εικόνα ενός νευρωνικού δικτύου. Μάλιστα οι συγγραφείς θεώρησαν ότι οι νευρώνες και οι συνδέσεις τους είναι ένα πρότυπο, ανάλογο ενός ηλεκτρικού κυκλώματος. Ο **McCulloch** ήταν νευροφυσιολόγος και ο **Pitts** ένας 18χρονος πρωτοετής φοιτητής των Μαθηματικών. Οι ίδιοι συγγραφείς προχώρησαν το 1947 σε πιο εξελιγμένο πρότυπο για την αναγνώριση σχημάτων.

Ένα άλλο έργο της πρώτης αυτής εποχής που αφήνει ακόμα και σήμερα την επιρροή του είναι το βιβλίο του **D. Hebb**, "**The organisation of behavior**" (1949), το οποίο εισάγει τον κανόνα μάθησης του Hebb. Το μοντέλο του **Hebb** έχει ως κεντρική ιδέα τις συνδέσεις μεταξύ μονάδων του συστήματος, δηλαδή τους νευρώνες.

Το μοντέλο του αισθητήρα (perceptron) παρουσιάστηκε για πρώτη φορά το 1957 από τον **F. Rosenblatt**, ο οποίος αρχικά έφτιαξε το πρώτο δίκτυο με hardware, και το οποίο μπορούσε να κάνει πολλές και διάφορες διεργασίες. Το μοντέλο αυτό στην αρχή είχε πολλές επιτυχίες, δημιούργησε μεγάλο ενθουσιασμό, και μάλιστα ήδη αρχίζει να συζητείται η ιδέα ότι πιθανόν τα νευρωνικά δίκτυα να είναι η πιο ανώτερη τεχνική που λύνει όλα τα προβλήματα που μέχρι τότε παρέμεναν άλυτα. Οι πρώτες λοιπόν επιτυχίες μεγαλοποιήθηκαν, αλλά γρήγορα φάνηκε ότι τα μοντέλα αυτά είχαν πολλούς περιορισμούς. Μια συνολική και εμπειριστατωμένη εικόνα του προτύπου αυτού παρουσιάστηκε το 1969 στο βιβλίο "Perceptrons" των **Minsky και Papert**. Στο βιβλίο αυτό γίνεται μία συνολική εκτίμηση της χρησιμότητας του προτύπου του αισθητήρα και όλων των διεργασιών για τα οποία είναι χρήσιμο. Η νέα διαδικασία εκπαίδευσης την μέθοδο της οπισθοδιάδοσης (back-propagation), η οποία κατέληξε να είναι η πιο χρήσιμη σήμερα τεχνική εκπαίδευσης δικτύων. Η μέθοδος αυτή είχε συζητηθεί και από άλλους νωρίτερα, αλλά για πρώτη φορά το 1986 παρουσιάστηκε ολοκληρωμένα και με αυστηρό μαθηματικό τρόπο.

Η ιστορία του perceptron συνεχίστηκε με το έργο που παρουσίασε ο **Geoffrey Hinton** το 1986, όταν παρουσίασε μια νέα διαδικασία εκμάθησης με το όνομα back-propagation που έγινε η ραχοκοκαλιά των σύγχρονων μοντέλων νευρωνικών δικτύων. Αυτή η τεχνική λειτουργεί ελαχιστοποιώντας τη διαφορά μεταξύ των πραγματικών και των επιθυμητών τιμών προσαρμόζοντας τα βάρη του μοντέλου νευρωνικού δικτύου. Έκανε το νευρωνικό δίκτυο να μάθει ή να εξάγει

χαρακτηριστικά και να γενικεύσει ένα μοτίβο ή μια ακολουθία εισόδων για να κάνει αρκετά ακριβείς προβλέψεις σε άορατες αναπαραστάσεις δεδομένων.

Έκτοτε έχει σημειωθεί μεγάλη πρόοδος. Τώρα, έχουμε μοντέλα όπως το **VGGNet**, το **ResNet**, το **Inception** και πολλά άλλα στα χέρια μας που μπορούν να ταξινομήσουν αντικείμενα γρήγορα και με ακρίβεια. Όλα αυτά βασίζονται στο γεγονός ότι προσπαθούμε να μιμηθούμε τον ανθρώπινο εγκέφαλο.

Μετά την πρόοδο σε τόσα πολλά σημεία που παρουσιάστηκε ιδιαίτερα την δεκαετία του 1980, τα τελευταία δέκα χρόνια παρατηρούμε ότι αρχίζουν να εμφανίζονται πολλά σημεία που δείχνουν ότι η περιοχή των νευρωνικών δικτύων έχει πλέον αναπτυχθεί σε ένα ανεξάρτητο πεδίο της επιστήμης με δικά του στοιχεία, δικό του χαρακτήρα σαφώς καθορισμένο, και τέλος με μεγάλο αριθμό επιστημόνων που ασχολούνται αποκλειστικά τώρα με την νέα αυτή περιοχή. Από το 1985 και μετά αρχίζουν τα πρώτα συνέδρια που είναι αφιερωμένα αποκλειστικά σε νευρωνικά δίκτυα, από την American Physical Society, και από την IEEE.

Τα τελευταία χρόνια μετά το 1990 εκδίδονται και άλλα 3- 4 νέα, με συνέπεια να υπάρχουν σήμερα περίπου 10 επιστημονικά περιοδικά αφιερωμένα στα νευρωνικά δίκτυα. Φυσικά, και τα γνωστά περιοδικά της Επιστήμης Υπολογιστών, της Φυσικής, και των Ηλεκτρολόγων Μηχανικών επίσης περιλαμβάνουν πλειάδα άρθρων με νέα αποτελέσματα. Κάθε μήνα πλέον δημοσιεύονται εκατοντάδες εργασίες με αποκλειστικό θέμα κάποια άποψη των νευρωνικών δικτύων.

Μερικά από τα εξειδικευμένα νέα περιοδικά είναι:

- Neural Networks: The Official Journal of the International Neural Network Society (Pergamon Press).
- Network: Computation in Neural Systems (Institute of Physics Publishing).
- International Journal of Neural Systems (World Scientific).
- Neural Computation
- Connection Science: Journal of Neural Computing, Artificial Intelligence and Cognitive Research (Carfax Publishing).

2.2 Τα δίκτυα Adaline

Το **Adaline** είναι ένα νευρωνικό δίκτυο ενός επιπέδου με πολλούς κόμβους όπου κάθε κόμβος δέχεται πολλαπλές εισόδους και δημιουργεί μία έξοδο

Νευρώνας ADALINE: ADAptive LINear Element

- Προτάθηκε από τους Widrow και Hoff το 1962
- Είναι μια ελαφρώς τροποποιημένη έκδοση του νευρώνα Perceptron
 - Δέχεται πραγματικές τιμές ως εισόδους
- Έχει βάρη
 - Έχει ως συνάρτηση ενεργοποίησης τη γραμμική συνάρτηση
- Μπορεί να χρησιμοποιηθεί για προβλήματα ταξινόμησης

Μερικά σημαντικά σημεία σχετικά με την Adaline είναι τα εξής

- Χρησιμοποιεί λειτουργία διπολικής ενεργοποίησης.
- Ο νευρώνας Adaline μπορεί να εκπαιδευτεί χρησιμοποιώντας τον κανόνα Delta ή τον κανόνα Least Mean Square (LMS) ή τον κανόνα widrow-hoff
- Η καθαρή είσοδος συγκρίνεται με την τιμή στόχο για τον υπολογισμό του σήματος σφάλματος.

Με βάση τον προσαρμοστικό αλγόριθμο προπόνησης προσαρμόζονται τα βάρη

Η βασική δομή του **Adaline** είναι παρόμοια με το perceptron που έχει έναν επιπλέον βρόχο ανάδρασης με τη βοήθεια του οποίου η πραγματική έξοδος συγκρίνεται με την επιθυμητή έξοδο/στόχο. Μετά τη σύγκριση με βάση τον αλγόριθμο προπόνησης, τα βάρη και η προκατάληψη θα ενημερωθούν.

Ο κανόνας εκμάθησης βρέθηκε ότι ελαχιστοποιεί το μέσο τετραγωνικό σφάλμα μεταξύ της ενεργοποίησης και των τιμών στόχου. Το Adaline αποτελείται από εκπαιδύσιμα βάρη, συγκρίνει την πραγματική απόδοση με την υπολογιζόμενη απόδοση και με βάση τα σφάλματα εφαρμόζεται αλγόριθμος εκπαίδευσης

Στο Adaline, όλος ο νευρώνας εισόδου συνδέεται απευθείας με τον νευρώνα εξόδου με τη σταθμισμένη συνδεδεμένη διαδρομή. Υπάρχει μια προκατάληψη b της συνάρτησης ενεργοποίησης 1 είναι παρούσα.

Αλγόριθμος:

Βήμα 1: Αρχικοποίηση βάρους δεν χρησιμοποιούνται μηδενικά αλλά μικρές τυχαίες τιμές. Ορισμός ρυθμού εκμάθησης α .

Βήμα 2: Ενώ η συνθήκη διακοπής είναι False, κάντε τα βήματα 3 έως 7.

Βήμα 3: για κάθε σενάριο εκπαίδευσης εκτελέστε τα βήματα 4 έως 6.

Βήμα 4: Ορίστε την ενεργοποίηση της μονάδας εισόδου $x_i = s_i$ για ($i=1$ έως n).

Βήμα 5: Υπολογίστε την καθαρή είσοδο στη μονάδα εξόδου

$$y_{in} = \sum w_i x_i + \beta$$

Εδώ, β είναι η μεροληψία και n είναι ο συνολικός αριθμός των νευρώνων.

Βήμα 6: Ενημερώστε τα βάρη και την προκατάληψη για $i=1$ σε n

$$w_i^{(new)} = w_i^{(παλιό)} + \alpha(t - y_{in})x_i \quad \parallel \quad b^{(new)} = b^{(παλιό)} + \alpha(t - y_{in})$$

και υπολογίστε σφάλμα : $(t - y_{in})^2$

όταν η προβλεπόμενη έξοδος και η πραγματική τιμή είναι ίδιες τότε το βάρος δεν θα αλλάξει.

Βήμα 7: Ελέγξτε την κατάσταση διακοπής. Η κατάσταση διακοπής μπορεί να είναι όταν το βάρος αλλάζει με χαμηλό ρυθμό ή χωρίς αλλαγή.

Βιβλιογραφία

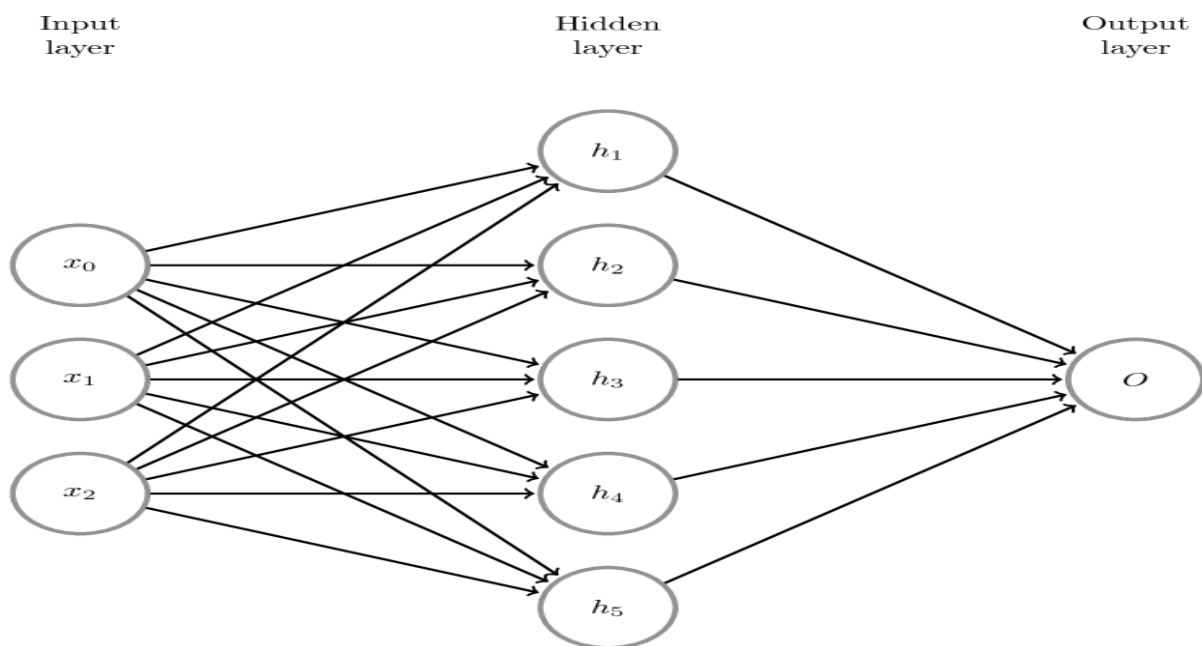
Το **Adaline (Adaptive Linear Neuron)** εισήχθη το 1959, λίγο μετά το perceptron του Rosenblatt, από τον Bernard Widrow και τον Ted Hoff (έναν από τους εφευρέτες του μικροεπεξεργαστή) στο Stanford. Ο Widrow και ο Hoff ήταν ηλεκτρολόγοι μηχανικοί, ωστόσο ο Widrow είχε παρακολουθήσει το διάσημο εργαστήριο Dartmouth για την τεχνητή νοημοσύνη το 1956, μια εμπειρία που τον έκανε να ενδιαφέρεται για την ιδέα της κατασκευής συστημάτων τεχνητής μάθησης που μοιάζουν με τον εγκέφαλο. Ο Widrow και ο Hoff σκέφτηκαν την ιδέα της Adaline κατά τη διάρκεια της πρώτης συνεδρίας που εργαζόνταν μαζί. Εκείνη την εποχή, η εφαρμογή ενός αλγορίθμου σε έναν κεντρικό υπολογιστή ήταν αργή και δαπανηρή, έτσι αποφάσισαν να κατασκευάσουν μια μικρή ηλεκτρονική συσκευή ικανή να

εκπαιδευτεί από τον αλγόριθμο ADALINE για να μάθει να ταξινομεί τα μοτίβα των εισόδων.

Από την άποψη της γνωστικής επιστήμης, η κύρια συμβολή του Adaline ήταν μεθοδολογική και όχι θεωρητική. Αν και το Adaline αρχικά εφαρμόστηκε σε προβλήματα όπως η αναγνώριση ομιλίας και προτύπων (Talbert et al., 1963), η κύρια εφαρμογή του Adaline ήταν στο προσαρμοστικό φιλτράρισμα και στην προσαρμοστική επεξεργασία σήματος. Τεχνολογίες όπως προσαρμοστικές κεραίες, προσαρμοστική ακύρωση θορύβου και προσαρμοστική εξισορρόπηση σε μόντεμ υψηλής ταχύτητας (που κάνει το Wifi να λειτουργεί καλά), αναπτύχθηκαν χρησιμοποιώντας το Adaline (Widrow & Lehr, 1990).

2.3 Τα δίκτυα MLP

Το **MLP** είναι ένα νευρωνικό δίκτυο και αποτελείται από τρεις τύπους στρωμάτων—το επίπεδο εισόδου, το στρώμα εξόδου και το κρυφό στρώμα.



Εικόνα 8. Το δίκτυο MLP

Το στρώμα εισόδου λαμβάνει το σήμα εισόδου που πρόκειται να επεξεργαστεί. Η απαιτούμενη εργασία όπως η πρόβλεψη και η ταξινόμηση εκτελείται από το επίπεδο εξόδου. Ένας αυθαίρετος αριθμός κρυφών επιπέδων που τοποθετούνται μεταξύ του επιπέδου εισόδου και εξόδου είναι η πραγματική υπολογιστική μηχανή του MLP. Παρόμοια με ένα δίκτυο προώθησης τροφοδοσίας σε ένα MLP, τα δεδομένα ρέουν

προς τα εμπρός από το επίπεδο εισόδου στο επίπεδο εξόδου. Οι νευρώνες στο MLP εκπαιδεύονται με τον αλγόριθμο εκμάθησης πίσω διάδοσης.

Τα **MLP** έχουν σχεδιαστεί για να προσεγγίζουν οποιαδήποτε συνεχή συνάρτηση και μπορούν να λύσουν προβλήματα που δεν είναι γραμμικά διαχωρισμένα. Οι κύριες περιπτώσεις χρήσης του MLP είναι η ταξινόμηση προτύπων, η αναγνώριση, η πρόβλεψη και η προσέγγιση.

Το νευρωνικό δίκτυο **MLP** (Multilayer Perceptron) είναι ένα από τα πιο βασικά μοντέλα νευρωνικών δικτύων και έχει μακρά ιστορία στη μηχανική μάθηση και την τεχνητή νοημοσύνη.

Το **MLP** αναπτύχθηκε από τον **Paul Werbos** το 1974 στο πλαίσιο της διατριβής του στο Πανεπιστήμιο του Harvard. Ο Werbos πειραματιζόταν με αλγορίθμους μηχανικής μάθησης για να βελτιώσει την αναγνώριση φωνής και ανακάλυψε τον αλγόριθμο backpropagation που χρησιμοποιείται σήμερα στα MLP.

Οι επιστήμονες της μηχανικής μάθησης **Geoff Hinton, Yann LeCun και Yoshua Bengio**, οι οποίοι κέρδισαν το Νόμπελ Οικονομίας το 2018, ήταν μεταξύ των πρώτων που εκτίμησαν τη δυνατότητα των MLP να εκπαιδευτούν σε βάθος και να χρησιμοποιηθούν για πολλαπλές εργασίες, όπως η αναγνώριση εικόνων και η μετάφραση μηχανής.

Το MLP κέρδισε δημοτικότητα στη δεκαετία του 1980 και του 1990, αλλά υπέστη σημαντική υποβάθμιση στα μέσα της δεκαετίας του 1990

Στη δεκαετία του 1990, η επιτυχία των νευρωνικών δικτύων μειώθηκε λόγω του προβλήματος της υπερεκπαίδευσης (overfitting), όπου το μοντέλο μαθαίνει τα δεδομένα εκπαίδευσης υπερβολικά καλά και δυσκολεύεται να γενικεύσει σε νέα δεδομένα. Ως αποτέλεσμα, οι επιστήμονες στράφηκαν σε άλλες μεθόδους μηχανικής μάθησης, όπως οι μη γραμμικοί ταξινομητές και τα δέντρα απόφασης.

Στην αρχή του 21ου αιώνα, οι νευρωνικοί αλγόριθμοι κέρδισαν ξανά δημοτικότητα λόγω της αύξησης της διαθεσιμότητας μεγάλων ποσοτήτων δεδομένων και των βελτιωμένων υπολογιστικών δυνατοτήτων. Το MLP έγινε ένα από τα βασικά μοντέλα νευρωνικών δικτύων και χρησιμοποιήθηκε ευρέως σε πολλές εφαρμογές, όπως η αναγνώριση φωνής, η εικόνας και η γλώσσας.

Το MLP έχει εξελιχθεί με την πάροδο του χρόνου με πολλές βελτιώσεις και παραλλαγές.

2.4 Η μέθοδος Back Propagation

Μέθοδος Back Propagation

Ας υποθέσουμε ότι δίνεται ένα σύνολο παραδειγμάτων $\mathfrak{S}=\{x(n),d(n)\}$, $n=1,\dots,N$. Το $x(n)$ είναι το διάνυσμα εισόδου της διάστασης m_0 και το $d(n)$ είναι το επιθυμητό διάνυσμα απόκρισης της διάστασης M

Έτσι ένα σήμα σφάλματος, $e_j(n)=d_j(n)-y_j(n)$ μπορεί να οριστεί για τον νευρώνα εξόδου j .

Μπορούμε να εξαγάγουμε έναν αλγόριθμο εκμάθησης για ένα MLP υποθέτοντας μια προσέγγιση βελτιστοποίησης που βασίζεται στην πιο απότομη κατεύθυνση καθόδου, π.χ. $\Delta w(n)=-\eta g(n)$

Όπου $g(n)$ είναι το διάνυσμα κλίσης της συνάρτησης κόστους και η είναι ο ρυθμός εκμάθησης.

Ο αλγόριθμος προέρχεται από την πιο απότομη κατεύθυνση καθόδου ονομάζεται back-propagation

Ας υποθέσουμε ότι ορίζουμε μια συνάρτηση στιγμιαίου κόστους SSE (δηλαδή ανά παράδειγμα) ως εξής:

$$E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n)$$

Όπου C είναι το σύνολο όλων των νευρώνων εξόδου.

Αν υποθέσουμε ότι υπάρχουν N παραδείγματα στο σύνολο \mathfrak{S} τότε το μέσο τετράγωνο σφάλμα είναι:

$$E_{av} = \frac{1}{N} \sum_{n=1}^N E(n)$$

Πρέπει να υπολογίσουμε την κλίση wrt E_{av} ή wrt σε $E(n)$. Στην πρώτη περίπτωση υπολογίζουμε την κλίση ανά εποχή (δηλαδή σε όλα τα μοτίβα N) ενώ στη δεύτερη η κλίση υπολογίζεται ανά μοτίβο.

Στην περίπτωση του E_{av} έχουμε τη λειτουργία Batch του αλγορίθμου. Στην περίπτωση του $E(n)$ έχουμε τον Online ή Στοχαστικό τρόπο του αλγορίθμου.

Ας υποθέσουμε ότι χρησιμοποιούμε την online λειτουργία για τον υπόλοιπο υπολογισμό. Η κλίση ορίζεται ως

$$\bar{g}(n) = \frac{\partial E(n)}{\partial w_{ji}(n)}$$

Χρησιμοποιώντας τον κανόνα της αλυσίδας του λογισμού μπορούμε να γράψουμε:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

Υπολογίζουμε τις διάφορες επιμέρους παραγώγους ως εξής:

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n)$$

Η εξίσωση σχετικά με τις διορθώσεις βάρους μπορεί να γραφεί ως εξής:

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n)$$

Πρέπει να διακρίνουμε δύο περιπτώσεις:

Το j είναι ένας νευρώνας εξόδου

Το j είναι ένας κρυφός νευρώνας

Έτσι, ο αλγόριθμος Back-Propagation είναι ένας αλγόριθμος διόρθωσης σφαλμάτων για εποπτευόμενη μάθηση.

Εάν το j είναι ένας νευρώνας εξόδου, έχουμε ήδη έναν ορισμό του $e_j(n)$, οπότε το $\delta_j(n)$ ορίζεται (μετά την αντικατάσταση) ως:

$$\delta_j(n) = (d_j(n) - y_j(n)) \phi_j'(v_j(n))$$

Εάν το j είναι ένας κρυφός νευρώνας τότε το $\delta_j(n)$ ορίζεται ως:

$$\delta_j(n) = - \frac{\partial E(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = - \frac{\partial E(n)}{\partial y_j(n)} \phi_j'(v_j(n))$$

Για να υπολογίσουμε τη μερική παράγωγο του $E(n)$ wrt σε $y_j(n)$ θυμόμαστε τον ορισμό του $E(n)$ και αλλάζουμε τον δείκτη για τον νευρώνα εξόδου σε k, δηλ.

$$E(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n)$$

Χρησιμοποιούμε ξανά τον κανόνα της αλυσίδας της διαφοροποίησης για να πάρουμε τη μερική παράγωγο του $ek(n)$ wrtyj(n):

Το τοπικό πεδίο $vk(n)$ ορίζεται ως:

$$v_k(n) = \sum_{j=0}^m w_{kj}(n) y_j(n)$$

Όπου m είναι ο αριθμός των νευρώνων (από το προηγούμενο στρώμα) που συνδέονται με τον νευρώνα k .

Ως εκ τούτου:

Συνδυάζοντας όλα μαζί, βρίσκουμε για την τοπική κλίση ενός κρυφού νευρώνα j τον ακόλουθο τύπο:

$$\delta_j(n) = \phi_j'(v_j(n)) \sum_{k \in C} \delta_k(n) w_{kj}(n)$$

Είναι χρήσιμο να θυμόμαστε την ειδική μορφή των παραγώγων για τα λογιστικά και τα υπερβολικά εφαπτομενικά σιγμοειδή:

$\phi_j'(v_j(n)) = y_j(n)[1 - y_j(n)]$ (Logistic)

$\phi_j'(v_j(n)) = [1 - y_j(n)][1 + y_j(n)]$ (Υπ. εφαπτομένη)

Αρχικοποίηση: Υποθέτοντας ότι δεν υπάρχουν διαθέσιμες προηγούμενες πληροφορίες, επιλέξτε τα συναπτικά βάρη και τα κατώφλια από μια ομοιόμορφη κατανομή της οποίας ο μέσος όρος είναι μηδέν και της οποίας η διακύμανση επιλέγεται ώστε το std των τοπικών πεδίων των νευρώνων να βρίσκεται στη μετάβαση μεταξύ του γραμμικού και του κορεσμένου μέρους της σιγμοειδούς συνάρτησης

Μπροστινός Υπολογισμός:

Αφήστε το παράδειγμα εκπαίδευσης στην εποχή να συμβολίζεται με $(x(n), d(n))$, όπου x είναι το διάνυσμα εισόδου και d το επιθυμητό διάνυσμα.

Υπολογίστε τα τοπικά πεδία προχωρώντας προς τα εμπρός μέσα από το επίπεδο του δικτύου. Το τοπικό πεδίο για τον νευρώνα j στο στρώμα l ορίζεται ως:

$$v_j^{(l)}(n) = \sum_{i=0}^m w_{ji}^{(l)}(n) y_i^{(l-1)}(n)$$

όπου m είναι ο αριθμός των νευρώνων που συνδέονται με το j και $y_i^{(l-1)}(n)$ είναι η ενεργοποίηση του νευρώνα i στο στρώμα $(l-1)$. $w_{ji}^{(l)}(n)$ είναι το βάρος

που συνδέει τους νευρώνες j και i .

Για $i=0$, έχουμε $y_0^{(l-1)}(n)=+1$ και $w_{j0}^{(l)}(n)=b_j^{(l)}(n)$ είναι η προκατάληψη του νευρώνα j .

Υποθέτοντας μια σιγμοειδή συνάρτηση, το σήμα εξόδου του νευρώνα j είναι:

$$y_j^{(l)}(n) = \phi_j(v_j^{(l)}(n))$$

Εάν το j βρίσκεται στο επίπεδο εισόδου, απλώς ορίζουμε:

$$y_j^{(0)}(n) = x_j(n)$$

όπου $x_j(n)$ είναι η j η συνιστώσα του διανύσματος εισόδου x .

Αν το j βρίσκεται στο επίπεδο εξόδου έχουμε:

$$e_j(n) = d_j(n) - o_j(n)$$

όπου $o_j(n)$ είναι η j η συνιστώσα του διανύσματος εξόδου o . L είναι ο συνολικός αριθμός των επιπέδων στο δίκτυο.

Υπολογίστε το σήμα σφάλματος:

όπου $d_j(n)$ είναι η επιθυμητή απόκριση για το j ο στοιχείο.

Υπολογισμός προς τα πίσω:

Υπολογίστε τα δ s του δικτύου που ορίζονται από:

$$\delta_j^{(l)}(n) = \begin{cases} e_j^{(L)}(n) \phi_j'(v_j^{(L)}(n)) & \text{for neuron } j \text{ in output layer } L \\ \phi_j'(v_j^{(l)}(n)) \sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) & \text{for neuron } j \text{ in hidden layer } l \end{cases}$$

όπου $\phi_j(\bullet)$ είναι η παράγωγος της συνάρτησης ϕ_j wrt το όρισμα.

Προσαρμόστε τα βάρη χρησιμοποιώντας τον γενικευμένο κανόνα δέλτα:

όπου α είναι η σταθερά της ορμής

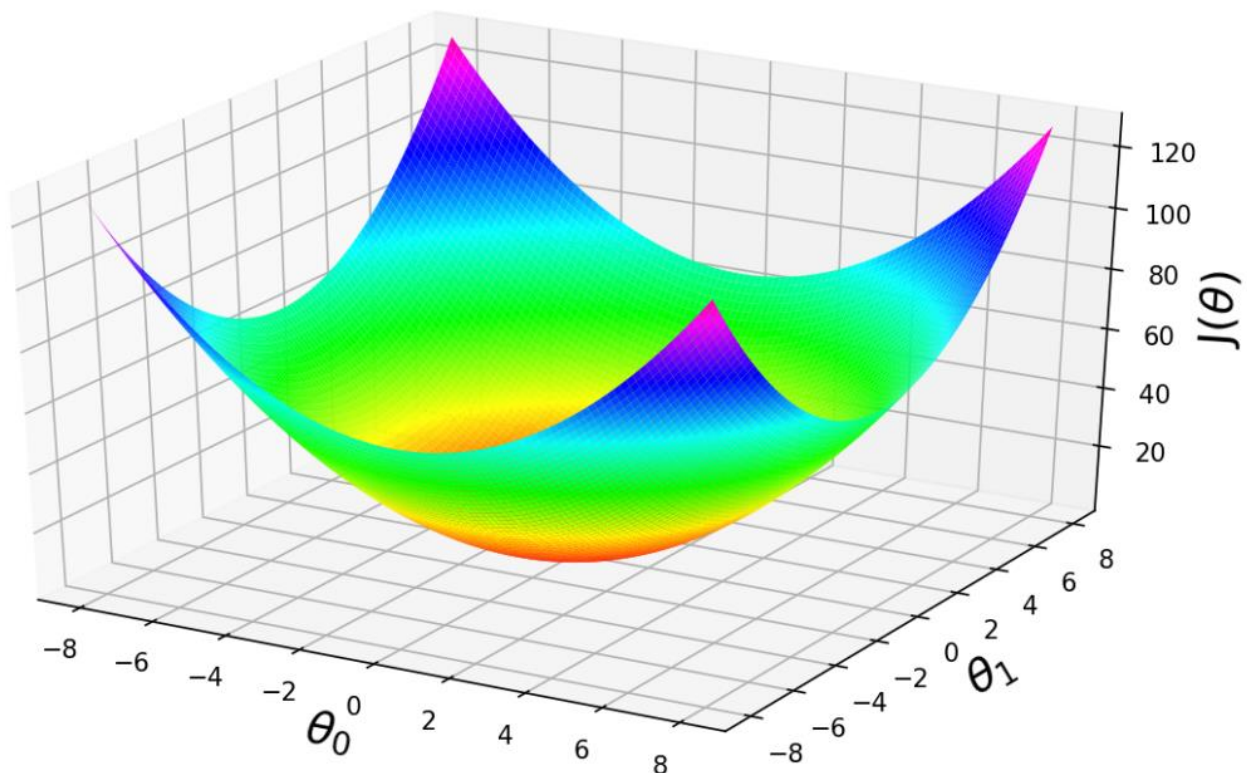
$$\Delta w_{ji}^{(l)}(n) = \alpha \Delta w_{ji}^{(l)}(n-1) + \eta \delta_j^{(l)}(n) y_i^{(l-1)}(n)$$

Ο αλγόριθμος **BP** θεωρείται ότι έχει συγκλίνει όταν ο Ευκλείδειος κανόνας του διανύσματος βαθμίδας φτάσει σε ένα αρκετά μικρό κατώφλι κλίσης.

Το **BP** θεωρείται ότι έχει συγκλίνει όταν η απόλυτη τιμή της μεταβολής στο μέσο τετραγωνικό σφάλμα ανά εποχή είναι αρκετά μικρή.

2.5 Η μέθοδος Gradient Descent

Το **Gradientdescent** είναι ένας αλγόριθμος βελτιστοποίησης που χρησιμοποιείται συνήθως για την εκπαίδευση μοντέλων μηχανικής μάθησης και νευρωνικών δικτύων. Τα δεδομένα εκπαίδευσης βοηθούν αυτά τα μοντέλα να μάθουν με την πάροδο του χρόνου και η συνάρτηση κόστους εντός της βαθμιδωτής κατάβασης λειτουργεί συγκεκριμένα ως βαρόμετρο, μετρώντας την ακρίβειά της με κάθε επανάληψη ενημερώσεων παραμέτρων. Έως ότου η συνάρτηση είναι κοντά ή ίση με το μηδέν, το μοντέλο θα συνεχίσει να προσαρμόζει τις παραμέτρους του για να αποφέρει το μικρότερο δυνατό σφάλμα. Μόλις τα μοντέλα μηχανικής μάθησης βελτιστοποιηθούν για ακρίβεια, μπορούν να αποτελέσουν ισχυρά εργαλεία για εφαρμογές τεχνητής νοημοσύνης (AI) και επιστήμης υπολογιστών.



Εικόνα 9. Gradient Descent

Πώς λειτουργεί η gradientdescent

Ο τύπος για την κλίση μιας ευθείας, που είναι $y = mx + b$, όπου το m αντιπροσωπεύει την κλίση και το b είναι η τομή στον άξονα y .

Θέλουμε να σχεδιάσουμε ένα διάγραμμα διασποράς στα στατιστικά και να βρούμε τη γραμμή της βέλτιστης προσαρμογής, η οποία απαιτούσε τον υπολογισμό του σφάλματος μεταξύ της πραγματικής παραγωγής και της προβλεπόμενης παραγωγής (\hat{y}) χρησιμοποιώντας τον τύπο μέσου τετραγώνου σφάλματος. Ο αλγόριθμος gradientdescent συμπεριφέρεται παρόμοια, αλλά βασίζεται σε μια κυρτή συνάρτηση.

Το σημείο εκκίνησης είναι απλώς ένα αυθαίρετο σημείο για να αξιολογήσουμε την απόδοση. Από αυτό το σημείο εκκίνησης, θα βρούμε την παράγωγο (ή την κλίση), και από εκεί, μπορούμε να χρησιμοποιήσουμε μια εφαπτομένη γραμμή για να παρατηρήσουμε την απότομη κλίση της κλίσης. Η κλίση θα ενημερώσει τις ενημερώσεις των παραμέτρων δηλ. τα βάρη και η προκατάληψη. Η κλίση στο σημείο εκκίνησης θα είναι πιο απότομη, αλλά καθώς δημιουργούνται νέες παράμετροι, η κλίση θα πρέπει σταδιακά να μειωθεί μέχρι να φτάσει στο χαμηλότερο σημείο της καμπύλης, γνωστό ως σημείο σύγκλισης.

Παρόμοια με την εύρεση της γραμμής της καλύτερης προσαρμογής στη γραμμική παλινδρόμηση, ο στόχος της κλίσης καθόδου είναι να ελαχιστοποιήσει τη συνάρτηση κόστους ή το σφάλμα μεταξύ προβλεπόμενου και πραγματικού y . Για να γίνει αυτό, απαιτούνται δύο σημεία δεδομένων μια κατεύθυνση και ένας ρυθμός εκμάθησης. Αυτοί οι παράγοντες καθορίζουν τους μερικούς υπολογισμούς παραγώγων των μελλοντικών επαναλήψεων, επιτρέποντάς του να φτάσει σταδιακά στο τοπικό ή παγκόσμιο ελάχιστο (δηλαδή στο σημείο σύγκλισης).

Ο ρυθμός μάθησης (που αναφέρεται επίσης ως μέγεθος βήματος ή άλφα) είναι το μέγεθος των βημάτων που γίνονται για να επιτευχθεί το ελάχιστο. Αυτή είναι συνήθως μια μικρή τιμή και αξιολογείται και ενημερώνεται με βάση τη συμπεριφορά της συνάρτησης κόστους. Τα υψηλά ποσοστά μάθησης οδηγούν σε μεγαλύτερα βήματα, αλλά κινδυνεύουν να υπερβούν το ελάχιστο. Αντίθετα, ένα χαμηλό ποσοστό μάθησης έχει μικρά μεγέθη βημάτων. Ενώ έχει το πλεονέκτημα της μεγαλύτερης ακρίβειας, ο αριθμός των επαναλήψεων θέτει σε κίνδυνο τη συνολική απόδοση, καθώς αυτό απαιτεί περισσότερο χρόνο και υπολογισμούς για να φτάσει στο ελάχιστο.

Η συνάρτηση κόστους (ή απώλειας) μετρά τη διαφορά ή το σφάλμα μεταξύ του πραγματικού y και του προβλεπόμενου y στην τρέχουσα θέση του. Αυτό βελτιώνει την αποτελεσματικότητα του μοντέλου μηχανικής εκμάθησης παρέχοντας ανατροφοδότηση στο μοντέλο έτσι ώστε να μπορεί να προσαρμόσει τις παραμέτρους για να ελαχιστοποιήσει το σφάλμα και να βρει το τοπικό ή καθολικό ελάχιστο. Επαναλαμβάνεται συνεχώς, κινούμενος κατά την κατεύθυνση της πιο απότομης καθόδου (ή της αρνητικής κλίσης) έως ότου η συνάρτηση κόστους είναι κοντά ή στο μηδέν. Σε αυτό το σημείο, το μοντέλο θα σταματήσει να μαθαίνει. Επιπλέον, ενώ οι όροι, συνάρτηση κόστους και συνάρτηση απώλειας, θεωρούνται συνώνυμοι, υπάρχει μια μικρή διαφορά μεταξύ τους. Αξίζει να σημειωθεί ότι μια συνάρτηση απώλειας αναφέρεται στο σφάλμα ενός παραδείγματος εκπαίδευσης, ενώ μια συνάρτηση κόστους υπολογίζει το μέσο σφάλμα σε ένα ολόκληρο σετ εκπαίδευσης.

Σε βαθύτερα νευρωνικά δίκτυα, συγκεκριμένα επαναλαμβανόμενα νευρωνικά δίκτυα, μπορούμε επίσης να αντιμετωπίσουμε δύο άλλα προβλήματα όταν το μοντέλο εκπαιδεύεται με gradient descent και backpropagation.

2.6 Παραδείγματα εφαρμογής νευρωνικών δικτύων

Οι μεγάλες δυνατότητες των αλγορίθμων νευρωνικών δικτύων, ιδιαίτερα σε προβλήματα ταξινόμησης/κατηγοριοποίησης, δίνουν λύσεις σε ένα τεράστιο εύρος προβλημάτων και έχουν πρακτικές εφαρμογές σχεδόν σε όλους τους τομείς της σύγχρονης έρευνας και ανάπτυξης τεχνολογίας.

Εφαρμογές σε διάφορους τομείς

Μια από τις πρώτες εφαρμογές των Τ.Ν.Δ. που ακόμα παραμένει σημαντική και χρήσιμη σε διάφορους κλάδους είναι η επεξεργασία σήματος. Η εφαρμογή των **Widrow και Hoff** (1960), η οποία αφαιρούσε την ηχώ από μια τηλεφωνική συνομιλία. Η αφαίρεση ανεπιθύμητου θορύβου από ηχητικά σήματα παραμένει μέχρι και σήμερα σημαντική, με διάφορες προσεγγίσεις των αλγορίθμων Τ.Ν.Δ. να δοκιμάζονται όπως δείχνουν οι εργασίες των Badri (2009) και Al-Allaf (2015). Στα πλαίσια της επεξεργασίας σήματος εντάσσεται και η πολύ διαδεδομένη εφαρμογή της αναγνώρισης φωνής, με στόχο είτε την υπαγόρευση κειμένου αντί δακτυλογράφησης, είτε την χρήση συσκευών με προφορικές εντολές. Μια προσέγγιση Τ.Ν.Δ. στο συγκεκριμένο πρόβλημα κάνουν οι **Dede και Sazli** (2010), με πολύ ικανοποιητικά αποτελέσματα.

Η αμέσως επόμενη κατά σειρά ιστορικότητας εφαρμογή είναι η αναγνώριση χειρόγραφων χαρακτήρων

Μια σύγχρονη εφαρμογή deep learning, η οποία εκμεταλλεύεται την ταχύτητα και τις δυνατότητες παράλληλης επεξεργασίας των σύγχρονων καρτών γραφικών (Ciresan et al, 2012) μπορεί πλέον να έχει ακρίβεια 99,77% στην αναγνώριση των χαρακτήρων της βάσης MNIST. Βέβαια, εκτός από την αναγνώριση χειρόγραφων χαρακτήρων, υπάρχουν και άλλες εφαρμογές αναγνώρισης εικόνας, όπως η αναγνώριση προσώπων, η οποία μεταξύ άλλων χρησιμοποιείται σε εφαρμογές βιομετρικής ασφάλειας, η σε εγκληματολογικές αναλύσεις βίντεο (Kasar et al, 2016).

Ένα στάδιο μετά την αναγνώριση εικόνας είναι η αναγνώριση δραστηριοτήτων σε βίντεο. Αυτές οι εφαρμογές αποτελούν την αιχμή της τεχνολογίας Deep Learning και δείχνουν το δρόμο για το τι θα ακολουθήσει. Εντυπωσιακό παράδειγμα είναι η πολύ πρόσφατη εφαρμογή διαβάσματος χειλιών από βίντεο *LipNet*, που χρηματοδοτήθηκε από το παράρτημα τεχνητής νοημοσύνης της Google, το *Google Deepmind*. Το LipNet είναι η πρώτη εφαρμογή αυτού του είδους που επιτυγχάνει μεγαλύτερη απόδοση και ακρίβεια από έναν άνθρωπο εκπαιδευμένο στο διάβασμα των χειλιών, καθώς μπορεί να διαβάσει τα λόγια ομιλητών ακόμα κι αν μιλούν ο

ένας πάνω στον άλλο, σε επίπεδο πρότασης και όχι λέξη προς λέξη, και με ακρίβεια 95,2%! Η εφαρμογή αυτή εκτός των άλλων αποτελεί και έναν οίονο για αυτό που εκτιμάται ότι θα ακολουθήσει σαν τάση στις εφαρμογές νευρωνικών δικτύων: από την αναγνώριση και εξαγωγή χαρακτηριστικών από στατικές φωτογραφίες, περνάμε στην αναγνώριση και εξαγωγή δραστηριοτήτων από βίντεο.

Η Google Deepmind προετοιμάζει ήδη τον διάδοχο της γνωστής υπηρεσίας μετάφρασής της, Google Translate, με τη χρήση ενός αλγορίθμου νευρωνικών δικτύων που επιτυγχάνει έως και 60% καλύτερες μεταφράσεις από την προηγούμενη εφαρμογή (Wu et al., 2016). Επίσης, τα εργαστήρια της NVIDIA βελτιώνουν συνεχώς τους ήδη εφαρμοζόμενους αλγορίθμους αυτόματης οδήγησης, που βρίσκονται σε οχήματα όπως τα ηλεκτροκίνητα Tesla, με τεχνικές που επιτρέπουν την καλύτερη αναγνώριση μικρών δρόμων χωρίς σήμανση (Bojarski et al., 2016).

Ιατρικές εφαρμογές

Πάρα πολλές είναι οι εφαρμογές και στον ιατρικό τομέα. Κυριότερες εξ αυτών, και χαρακτηριστικά παραδείγματα μηχανικής μάθησης γενικότερα, είναι οι εφαρμογές που αναλύουν εικόνες ή ενδείξεις ιατρικών οργάνων με στόχο τον εντοπισμό ανωμαλιών όπως κακοήθεις όγκους. Πρόσφατα παραδείγματα είναι μια νέα μέθοδος ανάλυσης μαστογραφιών για εντοπισμό όγκων που υποδεικνύουν καρκίνο του μαστού, (Addioi et al., 2016), και μια μέθοδος εντοπισμού πιθανών μελανωμάτων στο δέρμα με ανάλυση απλών φωτογραφιών που μπορεί να βγάλει οποιοσδήποτε με το κινητό του (αντί για εικόνες δερμοσκοπίου), επιτρέποντας έτσι την εύκολη και έγκαιρη πρόληψη (Jafari et al.) 2017.

Βιολογικές - γεωλογικές - περιβαλλοντολογικές εφαρμογές

Οι εφαρμογές στον φυσικό κόσμο μπορούν να ξεκινούν από το κυτταρικό επίπεδο, όπως η ταξινόμηση αλυσίδων DNA με σκοπό τον εντοπισμό μοτίβων και ανωμαλιών (Garro et al.2016) και να φτάνουν μέχρι τον εντοπισμό οργανισμών που ζουν στα βάθη των ωκεανών από σκοτεινές φωτογραφίες του βυθού (Hollis et al., 2016), ή την ακριβή πρόβλεψη της στάθμης μιας λίμνης σε καθημερινή βάση (Shiri et al., 2016).

Αρχιτεκτονικές και βιομηχανικές εφαρμογές

Τα νευρωνικά δίκτυα εξυπηρετούν αυτούς τους κλάδους με δυο συνήθως τρόπους: εφαρμογές στατιστικού ποιοτικού ελέγχου, και προσομοιώσεις. Ενδεικτικά αναφέρουμε δύο πολύ πρόσφατες εφαρμογές: εκτίμηση της ταλάντωσης

πολυκατοικιών σε περίπτωση σεισμού (Suryanita et al., 2016), και πρόβλεψη της εξάπλωσης της σκόνης σε έργα γεωτρήσεων - εκσκαφών (Nagesha et al., 2015).

Εφαρμογές στην φωτογραφική τέχνη

Οι αλγόριθμοι τεχνητών νευρωνικών δικτύων αποτελούν ένα πολύ χρήσιμο εργαλείο για την επεξεργασία εικόνων, χάρις στην ικανότητά τους να προσαρμόζονται στις ιδιότητες των δεδομένων εισόδου τους. Πηγαίνοντας ένα βήμα παραπέρα, μια άλλη πρόσφατη εφαρμογή μεταφέρει το στυλ ζωγραφικής ενός πίνακα σε μια φωτογραφία δημιουργώντας εντυπωσιακά καλλιτεχνικά αποτελέσματα (Gatys et al., 2016).

Πωλήσεις και Μάρκετινγκ

Χάρις στην δυνατότητά τους να συνεκτιμούν πολλούς διαφορετικούς παράγοντες όπως η ζήτηση ενός προϊόντος, το μέγεθος και οι οικονομικές δυνατότητες της αγοράς-στόχου, ή η τιμή των συμπληρωματικών προϊόντων, τα νευρωνικά δίκτυα έχουν πολύ καλά αποτελέσματα σε εφαρμογές πρόβλεψης πωλήσεων. Φυσικά, εδώ και δεκαετίες οι επιχειρήσεις χρησιμοποιούν άλλες, παραδοσιακές στατιστικές μεθόδους για την πρόβλεψη των πωλήσεών τους όπως η γραμμική παλινδρόμηση και η εποχική ανάλυση, αλλά μελέτες όπως αυτή των Zhang και Qi (2002) απέδειξαν ότι ένας συνδυασμός αυτών των παραδοσιακών τεχνικών με τεχνητά νευρωνικά δίκτυα αποδίδει πολύ καλύτερα.

Το δε μάρκετινγκ είναι ίσως η επιχειρησιακή λειτουργία με τις περισσότερες ευκαιρίες για χρήση αλγορίθμων Τ.Ν.Δ. Βασικός στόχος του μάρκετινγκ είναι ο εντοπισμός των πελατών που ενδέχεται να αντιδράσουν θετικά σε ένα προϊόν ή μια υπηρεσία, ώστε η διαφήμιση αυτού του προϊόντος ή της υπηρεσίας να είναι πιο εστιασμένη και να έχει μεγαλύτερο ποσοστό επιτυχίας. Μια εφαρμογή Τ.Ν.Δ. κατηγοριοποίησης μπορεί τμηματοποιήσει την αγορά βάσει δημογραφικών, κοινωνικών και οικονομικών παραγόντων, γεωγραφικής θέσης, αγοραστικής συμπεριφοράς κ.α (Badea, 2014). Παράδειγμα τέτοιας εφαρμογής είναι η εργασία των Πετρουλάκη και Μιαουδάκη, πάνω στην τμηματοποίηση της αγοράς κινητών τηλεφώνων (Petroulakis and Miaoudakis, 2007).

Έλεγχος πιστοληπτικής ικανότητας

Ο έλεγχος πιστοληπτικής ικανότητας είναι βασικό μέλημα των τραπεζών πριν δανείσουν ένα φυσικό πρόσωπο ή μια επιχείρηση. Καθώς ουσιαστικά αποτελεί ένα πρόβλημα κατηγοριοποίησης (αιτήσεων) βάσει ανομοιογενών χαρακτηριστικών

(προφίλ πελάτη), χωρίς ιστορικά στοιχεία για τον ίδιο τον πελάτη κάθε φορά, αλλά με ιστορικά στοιχεία για το σύνολο των δανείων που έχει δώσει ο οργανισμός η ή τράπεζα, αποτελεί ένα κλασσικό πεδίοεφαρμογής αλγορίθμων T.N.Δ.

Οι περισσότερες τέτοιες εφαρμογές κατατάσσουν τις αιτήσεις, ανάλογα με τα διαθέσιμα χαρακτηριστικά και το ιστορικό της εταιρίας, σε ασφαλείς, αβέβαιες, ή επισφαλείς, επιταχύνοντας έτσι την διαδικασία λήψης αυτής της απόφασης, και επιτρέποντας ενδεχομένως και σε κατώτερα στελέχη τη δυνατότητα να αναλάβουν αυτήν την απόφαση με ένα καλό επίπεδο σιγουριάς (De Nittis et al., 1998). Σε ένα παρόμοιο άρθρο παρατηρείται ότι μια τέτοια διαδικασία θα μπορούσε να γίνει σε δύο στάδια: σε πρώτο στάδιο να γίνονται αποδεκτές ή να απορρίπτονται αυτόματα οι ασφαλείς η οι επισφαλείς αιτήσεις αντίστοιχα, και στη συνέχεια να γίνεται αξιολόγηση εκ νέου για τις περιπτώσεις δεν μπορούν να ταξινομηθούν με σιγουριά, ώστε ο λήπτης της απόφασης να έχει μια ιδέα για την πλευρά στην οποία κλίνουν (West and Muchineuta, 2002).

Εκτίμηση αξίας

Η εκτίμηση της αξίας ενός ακινήτου ή άλλου περιουσιακού στοιχείου είναι σημαντική για τις τράπεζες, ιδιαίτερα σε περιπτώσεις υποθηκών η πλειστηριασμών. Ιδιαίτερα στην περίπτωση των ακινήτων, μέθοδοι T.N.Δ. που λαμβάνουν υπόψη παράγοντες όπως οι καιρικές και περιβαλλοντικές συνθήκες, ή η πρόσβαση σε μέσα μαζικής μεταφοράς κάνουν πολύ καλύτερες προβλέψεις από κλασσικά μοντέλα πρόβλεψης χρονοσειρών (Chiarazzo et al., 2014).

3. Γενετικοί αλγόριθμοι

3.1 Ιστορική αναδρομή

Ο γενετικός αλγόριθμος είναι μια μέθοδος για την επίλυση προβλημάτων βελτιστοποίησης τόσο περιορισμένων όσο και μη περιορισμένων που βασίζεται στη φυσική επιλογή, τη διαδικασία που οδηγεί τη βιολογική εξέλιξη.

Ο γενετικός αλγόριθμος τροποποιεί επανειλημμένα έναν πληθυσμό μεμονωμένων λύσεων. Σε κάθε βήμα, ο γενετικός αλγόριθμος επιλέγει άτομα από τον τρέχοντα πληθυσμό ως γονείς και τα χρησιμοποιεί για να παράγει παιδιά για την επόμενη γενιά

Κατά τη διάρκεια των διαδοχικών γενεών, ο πληθυσμός «εξελιίσσεται» προς μια βέλτιστη λύση.

Σε έναν γενετικό αλγόριθμο, ένας πληθυσμός υποψήφιων λύσεων (που ονομάζονται άτομα, πλάσματα, οργανισμοί ή φαινότυποι) σε ένα πρόβλημα βελτιστοποίησης εξελίσσεται προς καλύτερες λύσεις. Κάθε υποψήφιο διάλυμα έχει ένα σύνολο ιδιοτήτων (τα χρωμοσώματα ή τον γονότυπο του) που μπορούν να μεταλλαχθούν και να αλλοιωθούν. Παραδοσιακά, οι λύσεις αναπαρίστανται δυαδικά ως συμβολοσειρές των 0 και 1, αλλά είναι επίσης δυνατές και άλλες κωδικοποιήσεις.

Ένας τυπικός γενετικός αλγόριθμος απαιτεί:

- μια γενετική αναπαράσταση του πεδίου λύσης,
- μια συνάρτηση καταλληλότητας για την αξιολόγηση του τομέα λύσης.

Το μέγεθος του πληθυσμού εξαρτάται από τη φύση του προβλήματος, αλλά συνήθως περιέχει αρκετές εκατοντάδες ή χιλιάδες πιθανές λύσεις. Συχνά, ο αρχικός πληθυσμός δημιουργείται τυχαία, επιτρέποντας όλο το φάσμα των πιθανών λύσεων (ο χώρος αναζήτησης). Περιστασιακά, οι λύσεις μπορεί να «σπείρονται» σε περιοχές όπου είναι πιθανό να βρεθούν οι βέλτιστες λύσεις.

Πλεονεκτήματα

- Γρήγοροι και αξιόπιστοι
- Συνεργάζονται εύκολα με άλλα συστήματα
- Επεκτείνονται εύκολα
- Εφαρμόζονται σχεδόν σε κάθε πεδίο επιστήμης
- Είναι ανθεκτικοί σε σφάλματα
- Μπορούν να παραλληλοποιηθούν εύκολα

Βασικές αρχές

Η αναπαράσταση είναι ο θεμέλιος λίθος των γενετικών αλγορίθμων. Θα πρέπει να μην χάνεται πληροφορία κατά την αναπαράσταση δεδομένων (π.χ. από μετατροπή δεκαδικών τιμών με υποδιαστολή σε δυαδική αναπαράσταση). Θα πρέπει να είναι φανερή η διάκριση ανάμεσα σε καλά και κακά χρωμοσώματα.

ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

Η ιστορική αναδρομή ξεκινά από τις πρώτες προσπάθειες θεμελίωσης της θεωρίας της Εξέλιξης και φτάνει μέχρι τα τελευταία χρόνια, όπου πλέον η εφαρμογή των Γ.Α. είναι καθημερινή πρακτική.

Charles Darwin, 1809-1882. Κυρίαρχη μορφή στην επιστήμη της Βιολογίας. Ανακάλυψε και διατύπωσε τη θεωρία για την εξέλιξη μέσω της Φυσικής Επιλογής.

Νεοδαρβινισμός, 1930-σήμερα. Είναι η σύνθεση της Δαρβινικής Εξέλιξης και των σύγχρονων αντιλήψεων για την γενετική δομή. Στη βάση αυτής της θεωρίας βρίσκεται η πεποίθηση ότι η μετάλλαξη συμβαίνει τυχαία και επιφέρει ποικιλία στο γενετικό υλικό.

Ο John Holland αρχίζει την ερευνά του στην προσαρμογή των προγραμμάτων υπολογιστών, 1960. Θεωρείται ο "πατέρας" των Γ.Α. παρότι δεν τους βάφτισε ο ίδιος. Σειρά διαφόρων εργασιών που εκδόθηκαν το 1962 πάνω στη θεωρία των προσαρμοστικών συστημάτων έβαλαν τα θεμέλια για την ανάπτυξη του χώρου.

Ο J.D. Bagley βαφτίζει τους Γ.Α. , 1967. Η διδακτορική διατριβή του Bagley περιέχει την πρώτη δημοσιευμένη εφαρμογή των Γ.Α. που πρώτη φορά παρουσιάζονται με το όνομα αυτό.

R.S. Rosenberg, 1967. Δημοσιεύει εργασία, στην οποία γίνεται λόγος για προσομοίωση πληθυσμών μονοκύτταρων οργανισμών σε υπολογιστικό περιβάλλον.

Προσαρμογή στα Φυσικά και Τεχνητά συστήματα, 1975. Τίτλος του βιβλίου που εκδίδει ο Holland το 1975, στο οποίο αναπτύσσει τις ιδέες και την θεωρία των Γ.Α. Το βιβλίο θεωρείται πλέον κλασικό για τον χώρο. Θίγονται θέματα όπως η θεωρία των σχημάτων, η βέλτιστη κατανομή των ευκαιριών, σχέδια αναπαραγωγής, γενετικές λειτουργίες, η ευρωστία των Γ.Α. και πλήθος άλλα.

K,A. De Jong. 1975. Με την εργασία που εκδίδει βοηθά την πειραματική αξιολόγηση των Γ.Α. Σύμφωνα με αυτήν, προτείνονται λειτουργίες που ελέγχουν έναν Γ.Α. και την ικανότητά του να αντιμετωπίζει δύσκολα προβλήματα.

Ο J.J. Grefenstette δημιουργεί το GENESIS. 1980. Το GENESIS είναι ένα σύστημα ανάπτυξης Γ.Α. υλοποιημένο στη γλώσσα προγραμματισμού C, που έχει βοηθήσει σημαντικά στη διάδοση του γενετικού προγραμματισμού καθώς έγινε διαθέσιμο στο ευρύ κοινό.

1^ο Διεθνές Συνέδριο των Γ.Α. και των εφαρμογών τους, 1985. Ο χώρος αποκτά ένα μεγάλο συνέδριο που πλέον λαμβάνει χώρα κάθε δύο χρόνια και αντικατοπτρίζει το μεγάλο οργασμό που παρατηρείται σε επίπεδο τόσο θεωρίας, όσο και εφαρμογών.

Πολυάριθμες εκδόσεις Βιβλίων για Γ.Α., 1989-1999. Άλλη μια ένδειξη της τεράστιας ανάπτυξης του χώρου και της αποδοχής της νέας τεχνολογίας.

Ανάπτυξη πακέτων λογισμικού για Γ.Α.. 1990-1999. Πολλές εταιρίες δημιουργούν εμπορικά πακέτα που επιτρέπουν σε χρήστες να ενσωματώσουν στις εφαρμογές τους στοιχεία Γενετικού Προγραμματισμού (Genetic Programming). Ένα τέτοιο πακέτο είναι το EOS (Evolutionary Object System). Βασίζεται στη δημοφιλή γλώσσα

αντικειμενοστραφούς προγραμματισμού C++ και παρέχει μεγάλες δυνατότητες προσαρμογών και επεκτάσεων.

Ορισμένες από τις αρχικές εφαρμογές των γενετικών αλγορίθμων περιλαμβάνουν

- Βελτιστοποίηση συναρτήσεων
- Σχεδιασμός κυκλωμάτων
- Επιλογή χαρακτηριστικών σε μεγάλα σύνολα δεδομένων
- Εύρεση μέγιστης τιμής αριθμητικών συναρτήσεων.
- Επεξεργασία εικόνων
- Αναγνώριση προτύπων, όπως ακμές, επιφάνειες, ακόμη και αντικείμενα, σε ψηφιοποιημένες εικόνες.
- Συνδυαστική βελτιστοποίηση.

Το κλασικό πρόβλημα κατανομής πόρων σε δραστηριότητες, με σκοπό τη μεγιστοποίηση του οφέλους ή την ελάττωση του κόστους.

Στα συστήματα μηχανικής μάθησης οι γενετικοί αλγόριθμοι μπορεί να χρησιμοποιηθούν για την προσέγγιση συναρτήσεων.

Η πιο γνωστή εφαρμογή είναι αυτή των συστημάτων ταξινόμησης (classifier systems), ωστόσο οι γενετικοί αλγόριθμοι έχουν χρησιμοποιηθεί και σε παιχνίδια, επίλυση λαβυρίθμων, καθώς και για πολιτικές και οικονομικές αναλύσεις.

3.2 Μέθοδοι κωδικοποίησης

Δυαδική κωδικοποίηση

Σε έναν γενετικό αλγόριθμο, η δυαδική κωδικοποίηση είναι μια δημοφιλής τεχνική για την αναπαράσταση των υποψήφιων λύσεων ή ατόμων στον πληθυσμό. Η δυαδική κωδικοποίηση είναι ένας τρόπος αναπαράστασης δεδομένων χρησιμοποιώντας μόνο δύο σύμβολα, συνήθως 0 και 1. Στην περίπτωση γενετικών αλγορίθμων, η δυαδική κωδικοποίηση χρησιμοποιείται για την αναπαράσταση των υποψήφιων λύσεων ως συμβολοσειρά 0 και 1, όπου κάθε bit αντιπροσωπεύει ένα συγκεκριμένο χαρακτηριστικό ή παράμετρος της λύσης.

Για παράδειγμα, ας υποθέσουμε ότι θέλουμε να βελτιστοποιήσουμε μια συνάρτηση που λαμβάνει τρεις παραμέτρους εισόδου: x , y και z . Μπορούμε να αναπαραστήσουμε κάθε παράμετρο χρησιμοποιώντας μια δυαδική συμβολοσειρά,

όπου κάθε bit αντιπροσωπεύει μια τιμή είτε 0 είτε 1. Ας υποθέσουμε ότι αποφασίζουμε να χρησιμοποιήσουμε 4 bit για κάθε παράμετρο, έτσι το x αντιπροσωπεύεται από τα πρώτα 4 bit, το y αντιπροσωπεύεται από το επόμενο 4 bit και το z αντιπροσωπεύεται από τα τελευταία 4 bit. Τότε μια υποψήφια λύση με $x=5$, $y=2$ και $z=9$ θα αντιπροσωπευόταν από τη δυαδική συμβολοσειρά 0101 0010 1001.

Gray κωδικοποίηση

Ο κώδικας Gray είναι δυαδικός κώδικας χωρίς βάρη που χρησιμοποιείται για την κωδικοποίηση των δεκαδικών αριθμών.

Ο κώδικας Gray ονομάζεται κατοπτρικός κώδικας, λόγω του τρόπου κατασκευής του. Ο κώδικας Gray έχει το εξής σημαντικό χαρακτηριστικό: στον κώδικα Gray αλλάζει ένα μόνο bit μεταξύ δύο διαδοχικών αριθμών. Για παράδειγμα, οι διαδοχικοί αριθμοί 5 και 6 του κώδικα Gray είναι 0111 και 0101, αντίστοιχα, δηλαδή αλλάζει μόνο το δεύτερο bit από δεξιά. Επίσης, οι διαδοχικοί αριθμοί 7 και 8 του κώδικα Gray είναι 0100 και 1100, αντίστοιχα, δηλαδή αλλάζει μόνο το τέταρτο bit από αριστερά. Αυτό δεν συμβαίνει στο δυαδικό σύστημα. Οι διαδοχικοί αριθμοί 5 και 6 στο δυαδικό σύστημα είναι 0101 και 0110, αντίστοιχα, δηλαδή αλλάζουν τα 2 bits από δεξιά. Επίσης, οι διαδοχικοί αριθμοί 7 και 8 στο δυαδικό σύστημα είναι 0111 και 1000, αντίστοιχα, δηλαδή αλλάζουν και τα 4 bits.

Αν χρησιμοποιούνται δυαδικοί αριθμοί για τη μετάβαση από έναν αριθμό στον επόμενο, τότε υπάρχει η πιθανότητα σφάλματος: η μετάβαση από το 0111 (7) στο 1000 (8) μπορεί να οδηγήσει (για μικρό χρονικό διάστημα) στο 0110 (4) αν το LSB αλλάζει γρηγορότερα από τα άλλα bits, με αποτέλεσμα να γίνει λάθος στη μετατροπή. Αν χρησιμοποιείται ο κώδικας Gray για τη μετάβαση από έναν αριθμό στον επόμενο, τότε η πιθανότητα σφάλματος εξαλείφεται: η μετάβαση από το 0100 (7) στο 1100 (8) επιτυγχάνεται με την αλλαγή ενός (1) μόνο bit.

Ακέραια κωδικοποίηση

Η ακέραια κωδικοποίηση είναι μια μέθοδος κωδικοποίησης που χρησιμοποιείται στους γενετικούς αλγορίθμους για την αναπαράσταση των λύσεων προβλημάτων. Στην ακέραια κωδικοποίηση, κάθε λύση αναπαρίσταται από έναν ακέραιο αριθμό.

Οι γενετικοί αλγόριθμοι εκτελούν μια διαδικασία επιλογής, διασταύρωσης και μετάλλαξης στους κωδικοποιημένους αριθμούς για τη βελτιστοποίηση μιας συνάρτησης. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να επιτευχθεί μια αποδεκτή λύση στο πρόβλημα.

Η ακέραια κωδικοποίηση είναι πολύ απλή στην υλοποίηση και στην εφαρμογή και είναι ιδιαίτερα χρήσιμη σε προβλήματα βελτιστοποίησης που απαιτούν ακέραιες τιμές ως λύσεις. Ωστόσο, η ακέραια κωδικοποίηση μπορεί να οδηγήσει σε μη βέλτιστες λύσεις σε προβλήματα που απαιτούν συνεχείς τιμές, καθώς οι ακέραιοι αριθμοί δεν μπορούν να αντιπροσωπεύσουν όλες τις δυνατές τιμές της συνάρτησης.

Δεκαδική κωδικοποίηση

Ο γενετικός αλγόριθμος κωδικοποίησης δεκαδικών είναι ένας τύπος γενετικού αλγόριθμου που κωδικοποιεί λύσεις σε ένα πρόβλημα ως δεκαδικούς αριθμούς. Σε έναν δεκαδικό γενετικό αλγόριθμο κωδικοποίησης, κάθε άτομο στον πληθυσμό αναπαρίσταται ως δεκαδικός αριθμός. Ο δεκαδικός αριθμός αντιπροσωπεύει μια πιθανή λύση στο πρόβλημα που επιλύεται.

Η βασική ιδέα πίσω από έναν γενετικό αλγόριθμο κωδικοποίησης δεκαδικών είναι η χρήση γενετικών τελεστών όπως η επιλογή, η διασταύρωση και η μετάλλαξη για την εξέλιξη του πληθυσμού των δεκαδικών αριθμών προς καλύτερες λύσεις. Κατά τη διαδικασία επιλογής, άτομα με υψηλότερες τιμές φυσικής κατάστασης είναι πιο πιθανό να επιλεγούν για αναπαραγωγή. Στη διαδικασία της διασταύρωσης, δύο γονικά άτομα συνδυάζονται για να δημιουργήσουν ένα νέο παιδί με χαρακτηριστικά και των δύο γονέων. Η μετάλλαξη περιλαμβάνει τυχαίες αλλαγές στον δεκαδικό αριθμό για την εισαγωγή νέου γενετικού υλικού στον πληθυσμό.

Ο γενετικός αλγόριθμος κωδικοποίησης δεκαδικών έχει χρησιμοποιηθεί για την επίλυση ενός ευρέος φάσματος προβλημάτων βελτιστοποίησης, όπως ο προγραμματισμός, η διαχείριση αποθέματος και η μηχανική εκμάθηση. Ένα πλεονέκτημα του δεκαδικού γενετικού αλγορίθμου κωδικοποίησης είναι η απλότητα και η ευκολία εφαρμογής του. Επιπλέον, η δεκαδική κωδικοποίηση επιτρέπει τη χρήση παραδοσιακών αριθμητικών τελεστών όπως πρόσθεση, αφαίρεση, πολλαπλασιασμός και διαίρεση, που μπορούν να απλοποιήσουν τη διαδικασία αξιολόγησης της καταλληλότητας. Ωστόσο, ο γενετικός αλγόριθμος κωδικοποίησης δεκαδικών μπορεί να μην είναι κατάλληλος για όλα τα προβλήματα και μπορεί να απαιτεί προσεκτική επιλογή παραμέτρων όπως το μέγεθος του πληθυσμού, ο ρυθμός διασταύρωσης και ο ρυθμός μετάλλαξης για την επίτευξη βέλτιστων αποτελεσμάτων.

3.3 ΓΕΝΕΤΙΚΟΙ ΤΕΛΕΣΤΕΣ

Η συνάρτηση επιλογής επιλέγει τους γονείς για την επόμενη γενιά με βάση τις κλιμακούμενες τιμές τους από τη συνάρτηση κλιμάκωσης φυσικής κατάστασης.

Οι κλιμακούμενες τιμές καταλληλότητας ονομάζονται τιμές προσδοκίας. Ένα άτομο μπορεί να επιλεγεί περισσότερες από μία φορές ως γονέας, οπότε συνεισφέρει τα γονίδιά του σε περισσότερα από ένα παιδιά.

Η προεπιλεγμένη επιλογή ορίζει μια γραμμή στην οποία κάθε γονέας αντιστοιχεί σε ένα τμήμα της γραμμής μήκους ανάλογο με την κλιμακούμενη τιμή του. Ο αλγόριθμος κινείται κατά μήκος της γραμμής σε βήματα ίσου μεγέθους. Σε κάθε βήμα, ο αλγόριθμος εκχωρεί έναν γονέα από την ενότητα στην οποία προσγειώνεται.

Ο χειριστής διασταύρωσης είναι ανάλογος με την αναπαραγωγή και τη βιολογική διασταύρωση. Σε αυτό επιλέγονται περισσότεροι από ένας γονείς και παράγονται ένας ή περισσότεροι απόγονοι χρησιμοποιώντας το γενετικό υλικό των γονέων. Το crossover εφαρμόζεται συνήθως σε GA με μεγάλη πιθανότητα – pc .

Μερικοί από τους πιο ευρέως χρησιμοποιούμενους τελεστές crossover:

One Point Crossover

Σε αυτό το crossover ενός σημείου, επιλέγεται ένα τυχαίο σημείο διασταύρωσης και οι ουρές των δύο γονέων του ανταλλάσσονται για να αποκτήσουν νέα ελατήρια.

Crossover πολλαπλών σημείων

Η διασταύρωση πολλαπλών σημείων είναι μια γενίκευση της διασταύρωσης ενός σημείου όπου τα εναλλασσόμενα τμήματα ανταλλάσσονται για να ληφθούν νέα ελατήρια.

Ομοιόμορφο Crossover

Σε μια ομοιόμορφη διασταύρωση, δεν διαιρούμε το χρωμόσωμα σε τμήματα, αλλά αντιμετωπίζουμε κάθε γονίδιο ξεχωριστά. Σε αυτό, ουσιαστικά γυρνάμε ένα νόμισμα για κάθε χρωμόσωμα για να αποφασίσουμε αν θα συμπεριληφθεί ή όχι στο απόγονο. Μπορούμε επίσης να προκαταλάβουμε το νόμισμα σε έναν γονέα, για να έχουμε περισσότερο γενετικό υλικό στο παιδί από αυτόν τον γονιό.

Η μετάλλαξη είναι το τμήμα του GA που σχετίζεται με την «εξερεύνηση» του χώρου αναζήτησης. Έχει παρατηρηθεί ότι η μετάλλαξη είναι απαραίτητη για τη σύγκλιση του GA ενώ η διασταύρωση δεν είναι.

Με απλά λόγια, η μετάλλαξη μπορεί να οριστεί ως μια μικρή τυχαία αλλαγή στο χρωμόσωμα, για να ληφθεί μια νέα λύση. Χρησιμοποιείται για τη διατήρηση και εισαγωγή της ποικιλομορφίας στον γενετικό πληθυσμό και συνήθως εφαρμόζεται με μικρή πιθανότητα – μ.μ. Εάν η πιθανότητα είναι πολύ υψηλή, το GA μειώνεται σε μια τυχαία αναζήτηση.

Μεταλλάξεις

Μερικοί από τους πιο συχνά χρησιμοποιούμενους τελεστές μετάλλαξης. Όπως και οι τελεστές crossover, αυτή δεν είναι μια εξαντλητική λίστα και ο σχεδιαστής GA μπορεί να βρει πιο χρήσιμο έναν συνδυασμό αυτών των προσεγγίσεων ή έναν τελεστή μετάλλαξης για συγκεκριμένο πρόβλημα.

Bit Flip Mutation

Σε αυτή τη μετάλλαξη αναστροφής bit, επιλέγουμε ένα ή περισσότερα τυχαία bit και τα αναστρέφουμε. Αυτό χρησιμοποιείται για δυαδικά κωδικοποιημένα GA.

Τυχαία επαναφορά

Η τυχαία επαναφορά είναι μια επέκταση της αναστροφής bit για την αναπαράσταση ακέραιου αριθμού. Σε αυτό, μια τυχαία τιμή από το σύνολο των επιτρεπόμενων τιμών εκχωρείται σε ένα τυχαία επιλεγμένο γονίδιο.

Ανταλλαγή μετάλλαξης

Στη μετάλλαξη swap, επιλέγουμε τυχαία δύο θέσεις στο χρωμόσωμα και ανταλλάσσουμε τις τιμές. Αυτό είναι σύνηθες σε κωδικοποιήσεις που βασίζονται σε μετάθεση.

Scramble Mutation

Η μετάλλαξη Scramble είναι επίσης δημοφιλής με τις αναπαραστάσεις μετάθεσης. Σε αυτό, από ολόκληρο το χρωμόσωμα, επιλέγεται ένα υποσύνολο γονιδίων και οι τιμές τους ανακατεύονται ή ανακατεύονται τυχαία.

Μετάλλαξη Αναστροφής

Στη μετάλλαξη αναστροφής, επιλέγουμε ένα υποσύνολο γονιδίων όπως στη μετάλλαξη κρέμπλες, αλλά αντί να ανακατεύουμε το υποσύνολο, απλώς αντιστρέφουμε ολόκληρη τη συμβολοσειρά στο υποσύνολο.

3.4 Παράλληλοι γενετικοί αλγόριθμοι

Master-Slave

Πιθανώς ο ευκολότερος τρόπος για την εφαρμογή GA σε παράλληλους υπολογιστές είναι να διανείμει την αξιολόγηση της καταλληλότητας μεταξύ πολλών υποτελών επεξεργαστών ενώ ένα master εκτελεί τις λειτουργίες GA (επιλογή, διασταύρωση και μετάλλαξη).

Αυτοί οι αλγόριθμοι είναι σημαντικοί για διάφορους λόγους:

1. Εξερευνούν την αναζήτηση χώρο με τον ίδιο ακριβώς τρόπο όπως τα σειριακά GA, και επομένως οι υπάρχουσες κατευθυντήριες γραμμές σχεδιασμού για απλά GA ισχύουν άμεσα.
2. Είναι πολύ εύκολο στην εφαρμογή, γεγονός που τα κάνει δημοφιλή στους επαγγελματίες.
3. Σε πολλές περιπτώσεις τα κύρια-υπότελα GA οδηγούν σε σημαντικές βελτιώσεις στην απόδοση.

Ο χρόνος εκτέλεσης των master-slaveGA έχει δύο βασικά στοιχεία:

1. Ο χρόνος που χρησιμοποιείται στους υπολογισμούς και ο χρόνος που χρησιμοποιείται για την επικοινωνία πληροφοριών μεταξύ των επεξεργαστών.
2. Ο χρόνος υπολογισμού καθορίζεται σε μεγάλο βαθμό από το μέγεθος του πληθυσμού, οπότε μπορεί κανείς να μπει στον πειρασμό να μειώσει το πληθυσμού για να γίνει η ΓΣ πιο γρήγορη.

Η επικοινωνία πραγματοποιείται σε κάθε γενιά όταν ο κύριος στέλνει άτομα στους σκλάβους και όταν οι σκλάβοι επιστρέφουν τις αξιολογήσεις φυσικής κατάστασης στον κύριο.

Αυτή η ανταλλαγή χρησιμοποιείται για να βρεθεί ο αριθμός των σκλάβων που ελαχιστοποιεί το χρόνο εκτέλεσης του master-slaveGA.

Fine-Grained

Ο παράλληλος γενετικός αλγόριθμος Fine-Grained είναι μια τεχνική βελτιστοποίησης που χρησιμοποιεί παραλληλισμό στην εκτέλεση γενετικού αλγορίθμου. Η βασική ιδέα είναι να διαμεριστεί ο πληθυσμός σε μικρότερα σύνολα ,

τα οποία αλληλεπιδρούν μεταξύ τους για την επίλυση του προβλήματος βελτιστοποίησης.

Κατά την εκτέλεση τα υποσύνολα του πληθυσμού εξελίσσονται ταυτόχρονα σε διαφορετικούς επεξεργαστές με σκοπό την ταχύτερη σύγκλιση σε μια βέλτιστη λύση.

Το κλειδί για την επιτυχία του αλγορίθμου Fine-Grained είναι η κατανομή των υποσυνόλων του πληθυσμού η οποία προετοιμάζει την απόδοση του αλγορίθμου.

Η τεχνική αυτή χρησιμοποιείται σε προβλήματα βελτιστοποίησης που απαιτούν υπολογιστική ισχύ, όπως η αναζήτηση βέλτιστων παραμέτρων σε μηχανική μάθηση και βελτιστοποίηση λογισμικών εφαρμογών.

Coarse-Grained

Ο παράλληλος γενετικός αλγόριθμος Coarse-Grained Genetic Algorithm είναι ένας αλγόριθμος βελτιστοποίησης που βασίζεται στην μέθοδο των γενετικών αλγορίθμων και σχεδιάστηκε για να αντιμετωπίσει τα προβλήματα που σχετίζονται με την αποτελεσματική εκτέλεση του αλγορίθμου σε παράλληλα υπολογιστικά συστήματα.

Ο συγκεκριμένος αλγόριθμος (Coarse-Grained) χρησιμοποιεί μια σταδιακή προσέγγιση για την επίλυση του προβλήματος βελτιστοποίησης, διαιρώντας το πρόβλημα σε μικρότερα υποπροβλήματα και εκτελώντας τον αλγόριθμο σε κάθε υποπρόβλημα παράλληλα σε διάφορα νήματα.

Ένα από τα κύρια χαρακτηριστικά του είναι η χρήση ενός τελεστή διασταύρωσης, ο οποίος επιλέγει μόνο μία μικρή μερίδα των γονιδίων για διασταύρωση κάθε φορά. Αυτό βοηθάει στη διατήρηση της ποιότητας των λύσεων κατά τη διασταύρωση και μειώνει τον κίνδυνο ανεπιθύμητων απωλειών πληροφορίας.

ΙΣΤΟΡΙΚΟ

Υπάρχουν πολλά παραδείγματα master-slaveGA στη βιβλιογραφία. Πλέον από τις διαθέσιμες δημοσιεύσεις είναι από επαγγελματίες που πρέπει να βρουν λύσεις στις εφαρμογές τους πιο γρήγορες, αλλά υπάρχουν και μερικές σημαντικές θεωρητικές μελέτες.

Ο Bethke (1976) ήταν ο πρώτος που περιέγραψε παράλληλες υλοποιήσεις του α συμβατικό GA και ενός GA με χάσμα γενεών (ένας αλγόριθμος όπου μόνο ένα κλάσμα του πληθυσμού αντικαθίσταται σε κάθε γενιά).

Έκανε μια λεπτομερή ανάλυση της αποτελεσματικότητας της χρήσης της ικανότητας επεξεργασίας και κατέληξε στο συμπέρασμα ότι μπορεί να είναι κοντά στο 100% εάν

οι αξιολογήσεις φυσικής κατάστασης είναι ακριβείς σε σχέση με τις λειτουργίες GA, που συνήθως συμβαίνει.

Μια άλλη πρόμη μελέτη των παράλληλων GAmaster-slave έγινε από τον **Grefenstette** (1981). Πρότεινε τέσσερα πρωτότυπα για παράλληλους GA. Ο πρώτος έχει τρία πρωτότυπα τα οποία ήταν παραλλαγές ενός σχεδίου master-slave και το τέταρτο ήταν μια πολλαπλή παράλληλη GA.

3.5 Εφαρμογές γενετικών αλγορίθμων

Οι γενετικοί αλγόριθμοι είναι μια μέθοδος βελτιστοποίησης που επιτρέπει στους υπολογιστές να μιμηθούν τη φυσική εξέλιξη για να βρουν τις βέλτιστες λύσεις σε προβλήματα. Οι γενετικοί αλγόριθμοι χρησιμοποιούνται σε πολλούς τομείς, όπως η μηχανική, η βιολογία, η οικονομική και η φυσική επιστήμη. Παρακάτω παρουσιάζονται μερικά παραδείγματα εφαρμογής γενετικών αλγορίθμων:

1. Βελτιστοποίηση λειτουργίας κυκλωμάτων: Οι γενετικοί αλγόριθμοι χρησιμοποιούνται για τη βελτιστοποίηση της λειτουργίας κυκλωμάτων και τη βελτίωση της απόδοσής τους. Συγκεκριμένα, οι γενετικοί αλγόριθμοι μπορούν να βελτιστοποιήσουν τις παραμέτρους του κυκλώματος όπως η τάση τροφοδοσίας, η αντίσταση, η χωρητικότητα και άλλες.

2. Προγραμματισμός αυτόνομων οχημάτων: Οι γενετικοί αλγόριθμοι μπορούν να βελτιστοποιήσουν την απόδοση των αυτόνομων οχημάτων. Για παράδειγμα, μπορούν να βρουν τις βέλτιστες παραμέτρους για τον έλεγχο της ταχύτητας, της κατεύθυνσης, της ενέργειας και άλλων στοιχείων που επηρεάζουν την απόδοση των οχημάτων.

3. Εύρεση βέλτιστων εμπορικών στρατηγικών: Οι γενετικοί αλγόριθμοι μπορούν να βοηθήσουν τους επενδυτές στην εύρεση των βέλτιστων εμπορικών στρατηγικών. Μπορούν να αναλύουν τις αγορές, να προβλέπουν τις τάσεις και να βρίσκουν τις βέλτιστες επενδυτικές ευκαιρίες.

4. Βελτιστοποίηση των εργασιών στις παραγωγικές διαδικασίες: Οι γενετικοί αλγόριθμοι μπορούν να βελτιστοποιήσουν την απόδοση στις παραγωγικές διαδικασίες. Μπορούν να βελτιστοποιήσουν τον σχεδιασμό των μηχανών, των εργαλείων και των αλγορίθμων που χρησιμοποιούνται στην παραγωγή.

5. Επιλογή κατάλληλου συνδυασμού φαρμάκων: Οι γενετικοί αλγόριθμοι μπορούν να βοηθήσουν τους γιατρούς στην επιλογή του βέλτιστου συνδυασμού φαρμάκων για τη θεραπεία μιας ασθένειας.

Βιβλιογραφία

Li, X., Xu, J., Huang, H., & Li, K. (2019). Multi-objective optimization of an industrial polymerization reactor using a hybrid genetic algorithm. *Chemical Engineering Research and Design*, 148, 103-113.

Wang, Z., Zhang, Y., & Hu, Y. (2018). Design of intelligent controller for a quadrotor based on genetic algorithm optimized fuzzy neural network. *Aerospace Science and Technology*, 78, 1-10.

Dhir, R., & Kalia, H. (2019). Performance analysis of genetic algorithm-based trading strategy for Indian stock market. *International Journal of Business Forecasting and Marketing Intelligence*, 5(2), 204-222.

Huang, J., & Chang, C. H. (2018). Multi-objective optimization of machining parameters for the turning process using the grey relational analysis and genetic algorithm. *Journal of Manufacturing Systems*, 48, 122-133.

Bahrami, H., Khatami, S., & Ghasem-Aghaee, N. (2020). A multi-objective genetic algorithm for personalized drug combination therapy based on pharmacokinetic and pharmacodynamic parameters. *Journal of Biomedical Informatics*, 109, 103527.

4. Μέθοδος – αποτελέσματα

4.1 Τα dataset που χρησιμοποιήθηκαν

Τα δεδομένα που χρησιμοποιήθηκαν είναι το σύνολο δεδομένων ΒΚ το οποίο αποτελείται από τέσσερις εισόδους και μία έξοδο. Η κάθε είσοδος περιέχει 48 διαφορετικά δεδομένα. Το άλλο σύνολο δεδομένων το οποίο είναι το ΒL διαφέρει επειδή αποτελείται από 7 εισόδους και μια έξοδο με 20 διαφορετικά δεδομένα στην κάθε είσοδο. Επίσης χρησιμοποιήσαμε το σύνολο δεδομένων ΜΒ το οποίο περιέχει 2 εισόδους και μια έξοδο, και 31 διαφορετικά δεδομένα στην κάθε είσοδο. Τέλος χρησιμοποιήσαμε το σύνολο δεδομένων ΝΤ το οποίο και αυτό περιέχει 2 εισόδους και 1 έξοδο. Μπορούμε να πούμε πως οι εισοδοί αναφέρονται και ως διάσταση του προβλήματος και τα δεδομένα σε κάθε είσοδο καταγραφές ή περιπτώσεις.

Μπορούμε να δούμε τα δεδομένα στις παρακάτω εικόνες.

Αυτό είναι το σύνολο δεδομένων του ΒΚ.

```
4
48
0.088800 201.000000 36.020000 28.000000 0.640788
0.098300 191.000000 40.710000 30.000000 0.623918
0.127600 196.000000 38.400000 28.000000 0.613616
0.167100 201.000000 34.100000 31.000000 0.633323
0.049400 193.000000 32.380000 32.000000 0.596745
0.110700 196.000000 35.220000 25.000000 0.478650
0.182800 191.000000 29.540000 28.000000 0.419230
0.162700 196.000000 31.350000 28.000000 0.595402
0.156300 193.000000 34.560000 32.000000 0.548522
0.268100 183.000000 39.530000 27.000000 0.574201
0.130000 188.000000 30.770000 26.000000 0.359062
0.089600 198.000000 25.670000 30.000000 0.407883
0.207100 178.000000 36.220000 30.000000 0.372201
0.224400 185.000000 36.550000 23.000000 0.452523
0.105800 191.000000 28.350000 28.000000 0.494177
0.157700 193.000000 31.070000 25.000000 0.410869
0.125600 196.000000 27.870000 29.000000 0.694386
0.107000 198.000000 24.310000 34.000000 0.358017
0.134300 193.000000 31.260000 28.000000 0.421170
0.238300 185.000000 35.250000 26.000000 0.329651
0.216400 193.000000 24.490000 32.000000 0.237683
0.227000 191.000000 31.720000 27.000000 0.406987
0.118800 191.000000 22.740000 24.000000 0.372947
0.194000 193.000000 20.620000 27.000000 0.297850
0.249500 185.000000 30.460000 25.000000 0.467901
0.206900 170.000000 33.840000 30.000000 0.401911
0.087700 193.000000 21.670000 26.000000 0.623470
0.101000 193.000000 21.790000 24.000000 0.474769
0.094200 201.000000 20.170000 26.000000 0.435802
0.107100 196.000000 24.280000 24.000000 0.223350
0.213000 188.000000 21.590000 30.000000 0.377426
0.104300 196.000000 16.300000 23.000000 0.256793
0.113000 191.000000 23.010000 25.000000 0.255151
0.147700 196.000000 20.310000 31.000000 0.460436
0.212700 188.000000 14.570000 37.000000 0.131084
0.089800 196.000000 13.370000 34.000000 0.342938
0.214600 188.000000 20.510000 24.000000 0.526575
0.152800 191.000000 16.360000 33.000000 0.364586
0.156000 191.000000 16.030000 23.000000 0.162735
```

Εδώ βλέπουμε το σύνολο δεδομένων του BL

```
7
20
30.000000 2.000000 10.000000 1.500000 2.000000 6.000000 11.280000 0.185808
2.000000 2.000000 10.000000 1.500000 2.000000 10.000000 8.390000 0.115338
16.000000 2.000000 10.000000 2.500000 2.000000 6.000000 9.220000 0.135577
22.000000 2.000000 10.000000 2.500000 2.000000 10.000000 3.940000 0.066828
33.000000 2.000000 30.000000 1.500000 0.000000 6.000000 27.020000 0.130334
17.000000 2.000000 30.000000 1.500000 0.000000 10.000000 19.460000 0.068886
28.000000 2.000000 30.000000 1.500000 2.000000 6.000000 18.540000 0.061448
27.000000 2.000000 30.000000 1.500000 2.000000 10.000000 25.700000 0.119605
14.000000 2.000000 30.000000 2.500000 0.000000 6.000000 19.020000 0.065350
21.000000 2.000000 30.000000 2.500000 2.000000 10.000000 30.120000 0.155572
23.000000 6.000000 10.000000 1.500000 0.000000 6.000000 13.420000 0.237991
35.000000 6.000000 10.000000 1.500000 0.000000 10.000000 34.260000 0.746159
34.000000 6.000000 10.000000 1.500000 2.000000 10.000000 10.600000 0.169227
31.000000 6.000000 10.000000 2.500000 0.000000 6.000000 28.890000 0.615216
9.000000 6.000000 10.000000 2.500000 0.000000 10.000000 35.610000 0.779078
15.000000 6.000000 10.000000 2.500000 2.000000 10.000000 6.000000 0.057059
26.000000 6.000000 30.000000 1.500000 2.000000 6.000000 111.660000 0.818337
11.000000 6.000000 30.000000 1.500000 2.000000 0.000000 109.100000 0.797488
3.000000 4.000000 20.000000 2.000000 1.000000 8.000000 9.390000 0.025238
7.000000 4.000000 20.000000 2.000000 1.000000 8.000000 13.940000 0.080712
```


Αυτό είναι το σύνολο δεδομένων του MB

```
2
30
0.000000 590.000000 0.524052
0.000000 520.000000 0.437318
1.000000 500.000000 0.083819
1.000000 600.000000 0.405977
0.000000 440.000000 0.489796
0.000000 680.000000 0.978863
0.000000 600.000000 0.317055
0.000000 590.000000 0.576531
0.000000 640.000000 0.892128
0.000000 540.000000 0.690962
0.000000 540.000000 0.767493
0.000000 570.000000 0.655977
0.000000 710.000000 0.787172
0.000000 650.000000 0.080175
0.000000 580.000000 0.730321
1.000000 630.000000 0.526968
0.000000 580.000000 0.504373
0.000000 610.000000 0.616618
0.000000 620.000000 0.626822
0.000000 530.000000 0.448251
0.000000 570.000000 1.000000
0.000000 570.000000 0.215743
1.000000 640.000000 0.586735
0.000000 600.000000 0.580175
0.000000 720.000000 0.687318
0.000000 570.000000 0.208455
0.000000 530.000000 0.322886
0.000000 580.000000 0.638484
0.000000 610.000000 0.220117
0.000000 470.000000 1.000000
```

Αυτό είναι το σύνολο δεδομένων του NT

```
2
65
96.700000 1.000000 0.797753
97.100000 1.000000 0.820225
97.100000 1.000000 0.921348
97.400000 1.000000 0.808989
97.400000 1.000000 0.876404
97.500000 1.000000 0.842697
97.600000 1.000000 0.831461
97.800000 1.000000 0.730337
97.800000 1.000000 0.831461
97.900000 1.000000 0.853933
98.000000 1.000000 0.797753
98.000000 1.000000 0.831461
98.000000 1.000000 0.719101
98.000000 1.000000 0.876404
98.200000 1.000000 0.797753
98.200000 1.000000 0.808989
98.300000 1.000000 0.966292
98.300000 1.000000 0.808989
98.400000 1.000000 0.943820
98.500000 1.000000 0.764045
98.600000 1.000000 0.865169
98.600000 1.000000 0.876404
98.600000 1.000000 0.932584
98.600000 1.000000 0.786517
98.700000 1.000000 0.820225
98.700000 1.000000 0.876404
98.800000 1.000000 0.876404
98.800000 1.000000 0.910112
98.900000 1.000000 0.898876
99.000000 1.000000 0.842697
99.000000 1.000000 0.887640
99.000000 1.000000 0.910112
99.200000 1.000000 0.932584
99.300000 1.000000 0.707865
96.400000 2.000000 0.775281
96.700000 2.000000 0.696629
97.200000 2.000000 0.764045
97.600000 2.000000 0.685393
97.700000 2.000000 0.943820
```

4.2 Η προτεινόμενη μέθοδος

Η συγκεκριμένη συνάρτηση `isPointInt` μας βοηθάει να ελέγχουμε αν τα βάρη που έχουν δοθεί είναι σωστά.

```
int isPointInt(Data x, Data l, Data r) {  
  
    for (int i = 0; i < x.size(); i++)  
  
        if (x[i] < l[i] || x[i] > r[i]) return 0;  
        return 1;  
  
}
```

Αρχικοποιούμε ένα πίνακα με τα βάρη με την βοήθεια της συνάρτησης `makeChromosome` εφόσον είναι σωστά δοσμένα. Δηλαδή τα χρωμοσώματα είναι τα βάρη για το νευρωνικό μας δίκτυο.

```
void makeChromosome(Data &x, Data l, Data r){ // Δημιουργία χρωμοσωμάτων  
  
    for(int i=0 ;i<x.size(); i++){  
        x[i]=l[i]+(r[i]-l[i])*rand();  
    }  
  
}
```

Με την βοήθεια της συνάρτησης `mysigmoid` δημιουργούμε την συνάρτηση ενεργοποίησης για το νευρωνικό μας δίκτυο.

```
double mysigmoid(double x){ //Υπολογισμός Σιγμοειδούς  
  
    return (1/1+exp(-x));  
  
}
```

Η ProblemFitness μας βοηθάει να βρούμε την καταληλότητα η οποία επιστρέφει το αποτέλεσμα της διαφοράς των τετραγώνων, και του αποτελέσματος που επιθούμε να βρούμε και την καλούμε στην συνάρτηση CalculateFitness.

```
double problemFitness(Data x, Data l, Data r, double N_output, double y_output) // Τετραγωνικό σφάλμα
{
    if(!isPointInt(x, l, r))
    {
        return 1e+100;
    }

    return pow(x: N_output - y_output, y: 2);
}
```

Στη μέθοδο CalculateFitness υπολογίζουμε την συνάρτηση με την οποία συντάσσεται το νευρωνικό μας δίκτυο MLP.

```

Data calculateFitness(int d,vector<Data> genome, vector<double> &fitness ,vector<Data> data, Data l, Data r, int h, int g, int p) {
// Υπολογισμός Καταλληλότητας
for (int i = 0; i < g; i++) { // Για τα γενόμενα
    fitness[i] = 0;
    for (int j = 0; j < p; j++) { // Για τα πρότυπα
        double sum_funtction=0;

        for(int c=0; c<h; c++) { // Για τον συνολικό αριθμό μονάδων επεξεργασίας
            double p1=genome[i][(d+2)*c-(d+1)-1];
            double p2=0;
            for(int t=0; t<d; t++){ //Για την διάσταση του προβλήματος

                p2=p2+ genome[i][(d+2)*c-(d+1)+t-1]*data[j][t-1];
            }
            p2=p2+genome[i][(d+2)*c-1];

            double sigmoid_function=mysigmoid( x: p2);
            sum_funtction=sum_funtction+p1*sigmoid_function;
        }

        double y_output=sum_funtction;
        double N_output=data[j][d];
        fitness[i] = fitness[i] + problemFitness( x: genome[i], l, r, N_output,
                                                y_output);
    }
}
return fitness;
}

```

Με την βοήθεια της συνάρτησης sortChromosomes ταξινομούμε τα χρωμοσώματα κατά φθίνουσα σειρά.

```

void sortChromosomes(vector<Data> &genome,vector<double> &fitness) //Ταξιινόμηση
{
    Data temp;
    temp.resize( new_size: genome[0].size());
    for(int i=0;i<genome.size();i++)
    {
        for(int j=0;j<genome.size()-1;j++)
        {
            if(fitness[j+1]<fitness[j])
            {
                double dtemp=fitness[j];
                fitness[j]=fitness[j+1];
                fitness[j+1]=dtemp;
                temp=genome[j];
                genome[j]=genome[j+1];
                genome[j+1]=temp;
            }
        }
    }
}

```

Στην συνέχεια χρησιμοποιούμε την συνάρτηση makeChild στην οποία εκτελούμε την μέθοδο crossover ή αλλιώς διασταύρωση με την επιλογή 2 γονέων.

```
void makechild(Data &child,Data parent1,Data parent2) { //Διασταύρωση
    child.resize( new_size: parent1.size()); //Είναι για το xi
    child.resize( new_size: parent2.size()); // Είναι για το yi
    double a=(rand()*(-0.5))+1.5;
    for(int i=0; i<child.size(); i++){

        parent1[i]=a*parent1[i]+(1-a)*parent2[i];
        parent2[i]=a*parent2[i]+(1-a)*parent1[i];

    }
}
```

Η συνάρτηση CrossoverChromosomes παίρνει σαν εισόδους τα δεδομένα του πίνακα των χρωμοσωμάτων που είναι τα βάρη του. Με την τιμή $pc=0.99$ δημιουργεί ένα πίνακα με τα παιδιά που έχουν δημιουργηθεί, και τέλος αντικαθιστούμε το χειρότερο χρωμόσωμα.

```
void crossoverChromosomes(vector<Data> &genome,vector<double> fitness) // Επιλογή tournament
{
    const double pc=0.99;
    vector<Data> children;
    children.resize( new_size: (int)(genome.size()*pc));
    const int tournament_size=4;
    int count_children=0;
    do
    {
        int parent[2];
        for(int i=0;i<2;i++)
        {
            int min_pos=0;
            double min_fitness=1e+100;
            for(int j=0;j<tournament_size;j++)
            {
                int r=rand() % genome.size();
                if(fitness[r]<min_fitness)
                {
                    min_fitness=fitness[r];
                    min_pos=r;
                }
            }
            parent[i]=min_pos;
        }

        makechild( & children[count_children], parent1: genome[parent[0]], parent2: genome[parent[1]]);
        count_children++;
        if(count_children==children.size()) break;
        makechild( & children[count_children], parent1: genome[parent[1]], parent2: genome[parent[0]]);
        count_children++;
    }while(count_children<children.size());
}
```

Αντικατάσταση του χειρότερου χρωμοσώματος.

```
//Αντικαθιστούμε το χειρότερο χρωμόσωμα
for(int i=0;i<count_children;i++)
{
    genome[genome.size()-i-1]=children[i];
}
}
```

Τέλος είναι η μετάλλαξη χρωμοσωμάτων η οποία ορίζεται με την συνάρτηση `mutateChromosomes` και λειτουργεί ως εξής: για κάθε χρωμόσωμα αν ένας τυχαίος αριθμός είναι μικρότερος από το χρωμόσωμα τότε υπολογίζουμε ξανά με τον ίδιο τρόπο.

```
void mutateChromosomes(vector<Data> &genome, Data l, Data r) // Μετάλλαξη χρωμοσωμάτων
{
    double pm=1.0/genome.size();
    for(int i=1;i<genome.size();i++)
    {
        for(int j=0;j<genome[i].size();j++)
        {
            double rt=rand();
            if(rt<pm)
            {
                genome[i][j]=l[j]+(r[j]-l[j])*rand();
            }
        }
    }
}
```


4.3 Πειραματικά αποτελέσματα

Για τα BK dataset έχουμε Nodes=4 και αλλάζουμε τον αριθμό των χρωμοσωμάτων και παίρνουμε τα εξής αποτελέσματα.

Chromosomes	Train Error	Test Error
10	3.312621	3.323556
20	5.672159	5.271587
30	6.602402	6.955916
40	7.644566	8.041150
50	8.375080	8.707662

Πίνακας 1. BK Δεδομένα

Για τα BL δεδομένα χρησιμοποιούμε αριθμό κόμβων ίσο με 7 Nodes=7.

Chromosomes	Train Error	Test Error
5	0.083242	0.268361
10	0.134542	2.703590
15	1.762026	5.474470
20	3.078096	5.489339

Πίνακας 2. BL δεδομένα

Για τα MB δεδομένα έχουμε αριθμό κόμβων ίσο με 2 Nodes=2.

Chromosomes	Train Error	Test Error
5	0.877621	1.653251
10	3.542026	2.844048
15	5.720814	5.023419
20	7.226954	6.385627
25	9.426766	8.734797
30	11.030588	10.255975

Πίνακας 3. MB δεδομένα.

Για τα NT δεδομένα έχουμε Nodes=2.

Chromosomes	Train Error	Test Error
5	3.580608	3.345159
10	6.935995	6.368263
15	10.185332	9.541475
20	13.902540	12.665701
25	17.580232	15.870473
30	21.462820	19.292389
35	25.050875	22.421288
40	27.950255	25.887516
45	31.704075	29.700668
50	34.510286	33.511299
55	38.863901	36.485417
60	42.405625	40.458021
65	46.011609	44.094810

Πίνακας 4. NT δεδομένα.

5. Συμπεράσματα

Στα πλαίσια της συγκεκριμένης πτυχιακής εργασίας πραγματοποιήθηκε η μελέτη και η περιγραφή του κλάδου των τεχνητών νευρωνικών δικτύων.

Έγινε ιστορική αναφορά στα νευρωνικά δίκτυα καθώς και των εφαρμογών που χρησιμοποιούν τεχνητά νευρωνικά δίκτυα.

Τα τεχνητά νευρωνικά δίκτυα αποτελούν έναν καινούριο επιστημονικό κλάδο, ο οποίος αναπτύχθηκε περαιτέρω τα τελευταία χρόνια. Πρόκειται για έναν πολλά υποσχόμενο κλάδο των επιστημών, ικανός να επιλύσει προβλήματα από πολλούς και διαφορετικούς επιστημονικούς κλάδους, όπως την ιατρική, την οικονομία, και την επιστήμη των υπολογιστών.

Στην εργασία αυτή εκπαιδεύσαμε ένα τεχνητό νευρωνικό δίκτυο MLP (Multi Layer Perceptron) με την χρήση γενετικού αλγορίθμου.

Για την εκπαίδευση ενός δικτύου MLP χρησιμοποιήθηκε η χρήση του γενετικού αλγορίθμου στο δυαδικό σύστημα. Συγκεκριμένα όμως ο αλγόριθμος Error Back Propagation στοχεύει στην επίλυση του σφάλματος σύμφωνα με τον κανόνα εκπαίδευσης διόρθωσης λάθους.

Χρησιμοποιήθηκε η συνάρτηση του νευρωνικού δικτύου MLP λαμβάνοντας αριθμητικά δεδομένα και προσπαθήσαμε να εξάγουμε όσο το δυνατόν χαμηλότερο αποτέλεσμα στο Test Error γιατί με αυτή την τιμή δείχνουμε πόσο εκπαιδευμένο είναι το νευρωνικό δίκτυο.

Η συγκεκριμένη μέθοδος θα μπορούσε να βελτιωθεί στο μέλλον με τη χρήση ενός πιο εξελιγμένου γενετικού αλγορίθμου.

Οι επιστήμονες στοχεύουν στην κατασκευή πιο εξελιγμένων δικτύων με τη χρήση πιο ρεαλιστικών αλγορίθμων για την μάθηση των δικτύων αυτών καθώς και πιο ισχυρών τρόπων όσον αφορά την σύνδεση των νευρώνων που υπάρχουν στο νευρωνικό δίκτυο.

Η εκπαίδευση των νευρωνικών δικτύων με την συγκεκριμένη μέθοδο πρέπει στο μέλλον να εξελιχθεί προκειμένου να χρησιμοποιηθεί για πιο πολύπλοκες και απαιτητικές εφαρμογές.

Αυτό μπορεί να επιτευχθεί με την συνεχή βελτίωση της ισχύος που διαθέτουν οι ηλεκτρονικοί υπολογιστές όσον αφορά την υπολογιστική τους λειτουργία.

Βιβλιογραφία

Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, 65(6), 386-408.

Minsky, M. L., & Papert, S. A. (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT Press.

Widrow, B., & Lehr, M. A. (1990). 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9), 1415-1442.

Haykin, S. (2009). *Neural Networks and Learning Machines* (3rd ed.). Pearson.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

<https://towardsdatascience.com/machine-learning-evolution-the-story-of-perceptron-b1de3f180ed6>)

<https://levelup.gitconnected.com/training-a-single-perceptron-405026d61f4b>

<https://en.wikipedia.org/wiki/ADALINE#:~:text=3%20MADALINE,Definition,is%20the%20weight%20vector>)

<https://towardsdatascience.com/adaline-neural-networks-the-origin-of-gradient-descent-783ed05d7c18>)

<https://studyglance.in/nn/display.php?tno=5&topic=Adaptive-Linear-Neuron>)

<https://www.geeksforgeeks.org/adaline-and-madaline-network>

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.

Haykin, S. (1999). *Neural networks: a comprehensive foundation*. Prentice Hall.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Introduction to Evolutionary Computing" by A.E. Eiben και J.E. Smith

"Genetic Algorithms in Search, Optimization, and Machine Learning" by D.E. Goldberg

"Evolutionary Computation: Toward a New Philosophy of Machine Intelligence" by D.B. Fogel

"An Introduction to Genetic Algorithms" by M. Mitchell

E. Cantú-Paz, Αποτελεσματικοί και Ακριβείς Παράλληλοι Γενετικοί Αλγόριθμοι
KluwerAcademicPublishers 2001

[Genetic Algorithms and Evolutionary Computation](#)

https://link.springer.com/chapter/10.1007/978-1-4615-4369-5_3

Chen, Z., Zou, L., & Zhang, B. (2016). A novel coarse-grained parallel genetic algorithm based on domain decomposition. *Applied Soft Computing*, 40, 222-233.

Liu, L., Xu, X., Wang, J., & Yu, Z. (2017). A new coarse-grained parallel genetic algorithm for global numerical optimization. *Applied Soft Computing*, 51, 123-136.

Wang, Y., Qin, X., Zhang, W., & Zhou, Y. (2019). A coarse-grained parallel genetic algorithm based on decomposition and reconstruction. *Journal of Computational Science*, 31, 121-132.

Chen, C., Chen, J., Huang, S., & Zhang, W. (2019). A parallel coarse-grained genetic algorithm for high-dimensional numerical optimization. *Soft Computing*, 23(13), 5087-5099.

Liu, S., Liu, H., Wang, L., & Chen, X. (2020). A coarse-grained parallel genetic algorithm based on hyperplane partitioning. *IEEE Transactions on Cybernetics*, 50(4), 1637-1649