ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

DEMOKRITOS

ΙΝΣΤ. ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Δι-ιδρυματικό Πρόγραμμα Μεταπτυχιακών Σπουδών στην

«Τεχνητή Νοημοσύνη»

Βαθεία Μηχανική Μάθηση

Στεφόπουλος Ανδρέας

MTN2031

Σταυράκης Κωνσταντίνος

MTN2029

Περιεχόμενα

1. Εισαγωγή 3

2. θεωρητικό υπόβαθρο: Βαθιά Μηχανική Μάθηση 4

2.1 Ταξινόμηση 12

2.2 Αρχικοποίηση βαρών 13

2.3 Στοχαστική βελτιστοποίηση (Stochastic optimization) 14

2.4 Υπερπροσαρμογή (overfitting) 17

2.5 Απόρριψη (Dropout) 20

2.6 Πολυπλοκότητα Γραμμικών μοντέλων 20

2.7 Συνάρτηση ενεργοποίησης RELU 21

2.8 Πολυεπίπεδα Νευρωνικά δίκτυα (Multilayer Neural Networks) 22

2.9 Back Propagation 23

3. Βιβλιογραφία 26

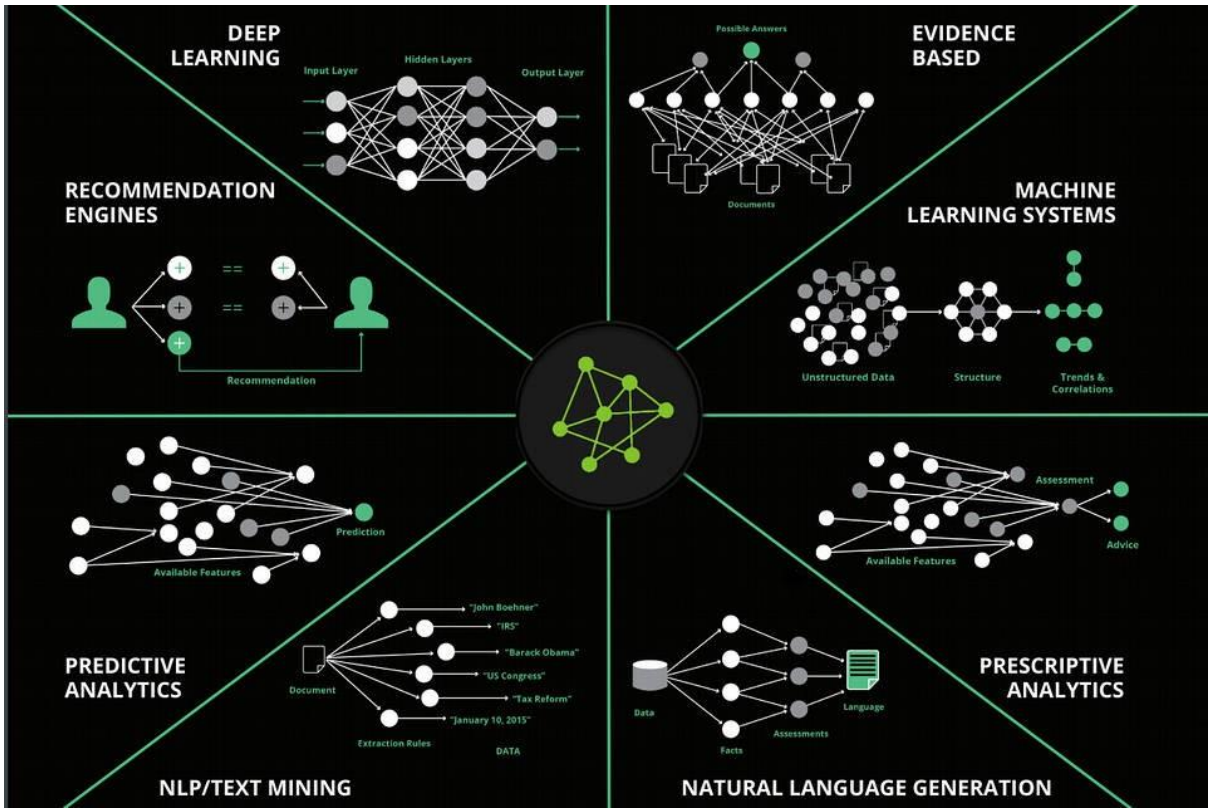
1. Εισαγωγή

Η βαθιά μάθηση αποτελεί μια επαναστατική προσέγγιση μηχανικής μάθησης, που συνιστά μια ειδική κατηγορία τεχνικών μηχανικής μάθησης, κατά την οποία πολλά επίπεδα επεξεργασίας πληροφοριών σε συστήματα ιεραρχικά εποπτευόμενων τεχνικών αξιοποιούνται για τη μη εποπτευόμενη εκμάθηση χαρακτηριστικών, καθώς για ανάλυση προτύπων ή κατηγοριοποίηση. Οι καταβολές και τα κίνητρα της βαθιάς μάθησης εντοπίζονται κυρίως στα τεχνητά νευρωνικά δίκτυα, καθώς και σε άλλους συναφείς επιστημονικούς τομείς, όπως η τεχνητή νοημοσύνη, η γνωσιακή νευροεπιστήμη και η επεξεργασίας σήματος μεταξύ άλλων. Τα τελευταία χρόνια, τα συστήματα που βασίζονται σε τεχνικές και αλγόριθμους βαθιάς μάθησης έχουν γίνει ιδιαιτέρως δημοφιλή τόσο στην ακαδημαϊκή κοινότητα όσο και σε πολλούς κλάδους της βιομηχανίας, λόγω των εξαιρετικών επιδόσεων των μεθόδων αυτών σε πληθώρα προβλημάτων μηχανικής μάθησης. Η παρούσα εργασία διερευνά πως έχουν καθοριστεί οι “βαθιές” θεμελιώδεις ιδέες και πως έχει μετατοπιστεί το ερευνητικό ενδιαφέρον κατά τη διάρκεια του χρόνου. Σε αυτό το πλαίσιο παρουσιάζονται και αναλύονται τα βασικά δομικά στοιχεία για την οικοδόμηση αρχιτεκτονικών βαθιάς μάθησης.

2. θεωρητικό υπόβαθρο: Βαθιά Μηχανική Μάθηση

Η εισαγωγή του κλάδου της μηχανικής μάθησης στην επιστήμη των υπολογιστών, επέτρεψε στους υπολογιστές να μπορούν να αντιμετωπίσουν προβλήματα αντίληψης για τον πραγματικό κόσμο, όσο και να παίρνουν υποκειμενικές αποφάσεις.

Οι αλγόριθμοι ML, επιτρέπουν σε συστήματα AI να προσαρμόζονται εύκολα σε καινούργια προβλήματα, απαιτώντας ελάχιστη επέμβαση από τον άνθρωπο. Για παράδειγμα, ένα Νευρωνικό Δίκτυο που έχει εκπαιδευτεί να αναγνωρίζει γάτες σε εικόνες, δεν απαιτεί να σχεδιαστεί και να εκπαιδευτεί από το μηδέν για να έχει την ικανότητα να αναγνωρίζει και σκύλους.



Εικόνα 1 : Κλάδοι και εφαρμογές της επιστήμης της Τεχνητής Νοημοσύνης

Πολλά προβλήματα που μέχρι πριν μερικά χρόνια λύνονταν με “χειρόγραφη”, προγραμματισμένη από τον άνθρωπο γνώση, σήμερα επιλύονται με χρήση αλγορίθμων ML (σχήμα 1). Κάποια παραδείγματα αφορούν:

- Αναγνώριση ομιλίας - Speech Recognition
- Μηχανική όραση - Computer Vision
  - Αναγνώριση αντικειμένων σε εικόνες - Object Recognition
  - Αναγνώριση και εντοπισμός της θέσης αντικειμένων σε εικόνες - Object Detection

- Αναγνώριση ηλεκτρονικών επιθέσεων στο διαδίκτυο - Cyberattack detection
- Επεξεργασία φυσικής γλώσσας - Natural Language Processing
  - Κατανόηση της φυσικής γλώσσας του ανθρώπου - Natural Language Understanding
  - Μοντελοποίηση και χρήση της φυσικής γλώσσας του ανθρώπου από μηχανές - Natural Language Generation
- Μηχανές αναζήτησης - Search Engines
- Αναπαράσταση γνώσης - Knowledge Representation
- Ρομποτική

Τα προβλήματα Μηχανικής Μάθησης χωρίζονται σε τρεις μεγάλες κατηγορίες:

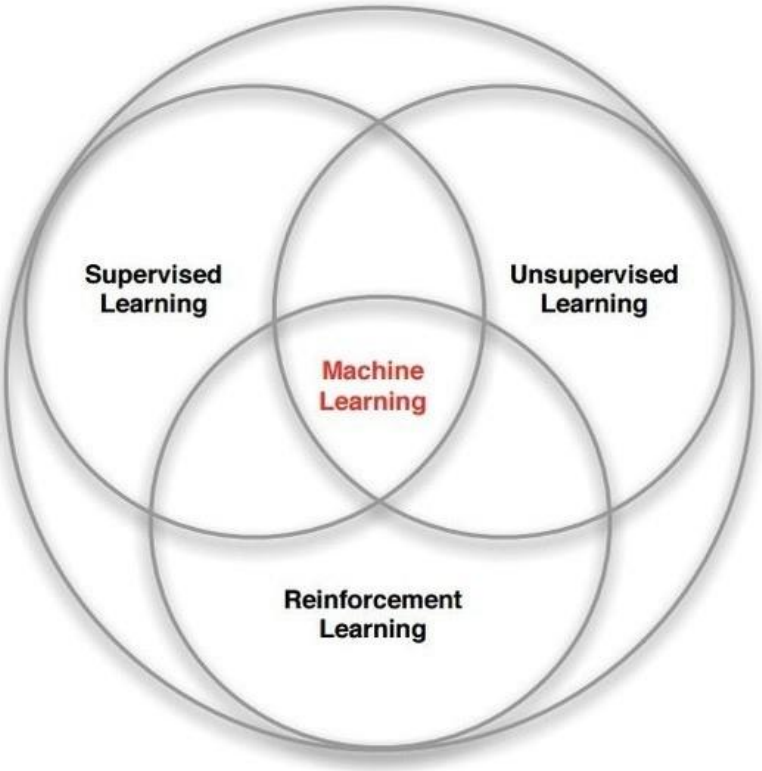
- Υπό επίβλεψη Μάθηση - Supervised Learning: Στο υπολογιστικό σύστημα δίνονται παραδείγματα εισόδου και επιθυμητής εξόδου, δηλαδή στα δεδομένα έχουν προηγουμένως ανατεθεί ετικέτες (labels) και στόχος είναι να εξαχθεί ένας γενικός κανόνας αντιστοίχισης της εισόδου στην επιθυμητή έξοδο. Η αναγνώριση προκαθορισμένων αντικειμένων σε εικόνες είναι ένα πρόβλημα που ανήκει σε αυτή την κατηγορία.
- Χωρίς επίβλεψη Μάθηση - Unsupervised Learning: Τα δεδομένα δεν έχουν ετικέτες (labels) αφήνοντας έτσι τον αλγόριθμο ML να βρει από μόνος του δομές στα δεδομένα εισόδου.
- Εκμάθηση δια ανταμοιβής - Reinforcement Learning: Ο πράκτορας αλληλεπιδρά με ένα δυναμικό περιβάλλον στο οποίο πρέπει να εκτελέσει ένα συγκεκριμένο στόχο, χωρίς την ύπαρξη ενός “δασκάλου” που να ορίζει ρητά αν έχει φθάσει κοντά στον στόχο. Ένα παράδειγμα εφαρμογής είναι η αυτόματη πλοήγηση ενός οχήματος.

Κάποια προβλήματα είναι υβριδικά, δηλαδή συνδυασμός των πιο πάνω. Στο σχήμα 2 απεικονίζεται το διάγραμμα Venn των διαφόρων αλγοριθμικών κατηγοριών ML.

Επιπλέον, οι Supervised Learning αλγόριθμοι χωρίζονται σε 2 κατηγορίες, ανάλογα με την επιθυμητή μορφή της εξόδου του αλγόριθμου ML:

- Ταξινόμησης - Classification: Όταν η έξοδος παίρνει διακριτές τιμές (discrete).
- Regression: Όταν η έξοδος παίρνει συνεχείς τιμές.

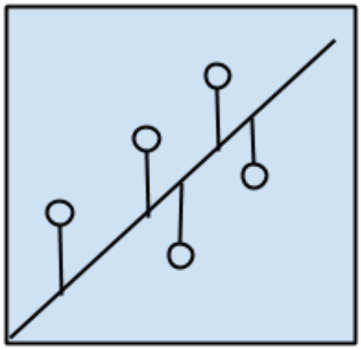
Γενικότερα, οι αλγόριθμοι ML ομαδοποιούνται και ανάλογα με την ομοιότητα τους σε σχέση με την λειτουργία που εκτελούν. Πιο κάτω αναφέρονται οι πιο δημοφιλείς αλγόριθμοι ML ομαδοποιημένοι με βάση την λειτουργία τους:



Εικόνα 2 : Διάγραμμα Venn των διαφόρων κατηγοριών μηχανικής μάθησης

Regression

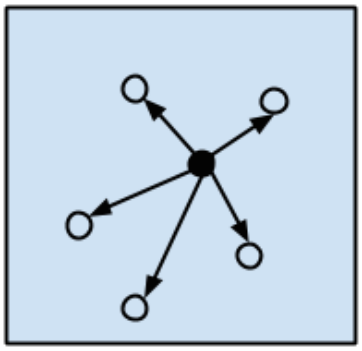
Ασχολείται με τη μοντελοποίηση της σχέσης μεταξύ των μεταβλητών που ανανεώνονται επαναληπτικά χρησιμοποιώντας ένα μέτρο σφάλματος για τις προβλέψεις που γίνονται από το μοντέλο



- Ordinary Least SquaresRegression (OLSR)
- Linear Regression
- Logistic Regression
- Stepwise Regression
- Multivariate Adaptive Splines (MARS) Regression
- Locally Estimated Smoothing (LOESS) Scatterplot

Instance-based

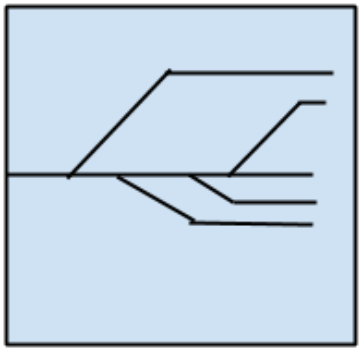
Αυτές οι μέθοδοι δημιουργούν μία βάση με παραδείγματα δεδομένων και συγκρίνουν τις νέες εισόδους με αυτές που έχουν καταχωρηθεί στην βάση δεδομένων χρησιμοποιώντας ένα μέτρο ομοιότητας, για την πιθανοτική εύρεση της καλύτερης αντιστοιχίας.



- k-Nearest Neighbour (kNN)
- Learning Vector Quantization (LVQ)
- Self-Organizing Map (SOM)
- Locally Weighted Learning (LWL)

Regularization

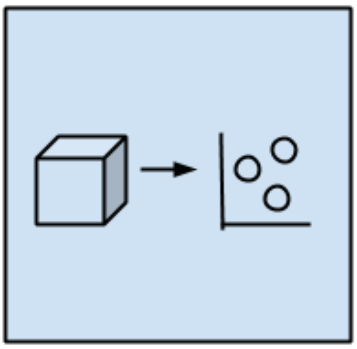
Χρησιμοποιούνται σαν επεκτάσεις άλλων μεθόδων και "τιμωρούν" πολύπλοκα μοντέλα, ευνοώντας έτσι απλούστερα μοντέλα τα οποίο είναι συνήθως καλύτερα στην γενίκευση της επίλυσης του εκάστοτε προβλήματος.



- Ridge Regression
- Least Absolute Shrinkage and Selection Operator (LASSO)
- Least-Angle Regression (LARS)
- Elastic Net

Dimensionality Reduction

Χρησιμοποιούνται για την αφαίρεση ασήμαντης πληροφορίας από τα δεδομένα. Πολλές από τις μεθόδους αυτές χρησιμοποιούνται σαν επεκτάσεις σε μοντέλα επίλυσης προβλημάτων regression και classification

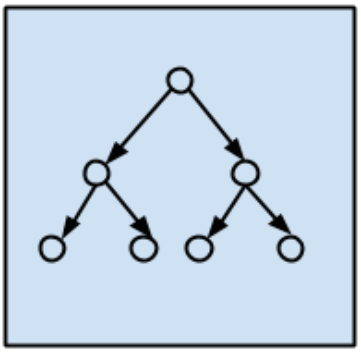


- Principal Component Analysis (PCA)
- Discriminant Analysis:
  - Linear (LDA)
  - Mixture (MDA)
  - Quadratic (QDA)
  - Flexible (FDA)
- Principal Component Regression (PCR)
- Multidimensional Scaling (MDS)

Decision Trees

Χρησιμοποιούνται για την κατασκευή μοντέλων λήψης αποφάσεων, τα οποία χρησιμοποιούν τις πραγματικές τιμές των χαρακτηριστικών των δεδομένων.

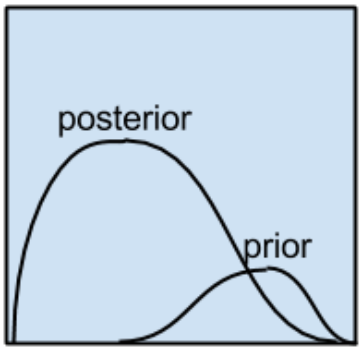
- Classification and Regression Tree (CART)
- Conditional Decision Trees
- M5



Bayesian

Εφαρμόζουν το θεώρημα του Bayes για την επίλυση τόσο προβλημάτων regression, αλλά και classification

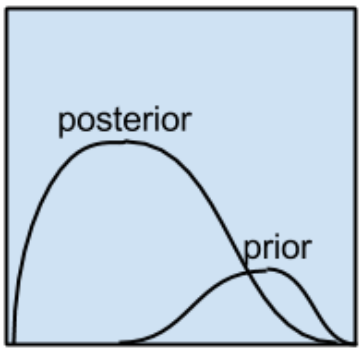
- Naive Bayes
- Gaussian Naive Bayes
- Bayesian Network (BN)
- Bayesian Belief Network (BBN)



Clustering

Περιγράφουν τις κλάσεις του προβλήματος

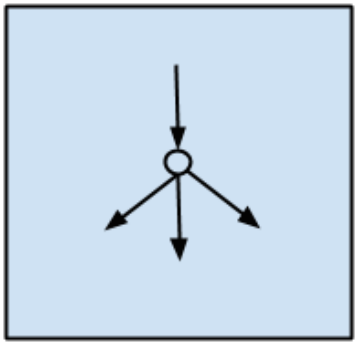
- k-Means
- k-Medians
- Expectation Maximisation (EM)
- Hierarchical Clustering



Artificial Neural Networks (ANN)

Μοντέλα εμπνευσμένα από τη δομή ή/και την λειτουργία των βιολογικών νευρωνικών δικτύων. Χρησιμοποιούνται στην επίλυση προβλημάτων classification ή/και regression.

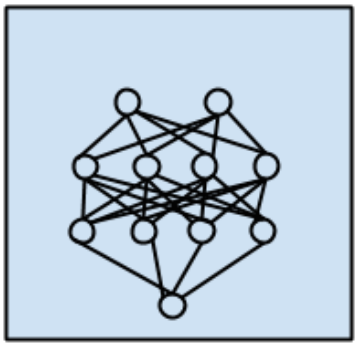
- Perceptron
- Back-Propagation
- Radial Basis Function Network (RBFN)



Deep Learning (DL)

Οι αλγόριθμοι DL είναι η σύγχρονη επέκταση των ANN, τα οποία εκμεταλλεύονται την αφθονία επεξεργαστικής ισχύος των σύγχρονων υπολογιστικών συστημάτων.

- Autoncoder
- Multilayer Perseptron (MLP)
- Deep Boltzmann Machine (DBM)
- Deep Belief Networks (DBN)
- Convolutional Neural Network (CNN)
- Stacked Auto-Encoders
- Recurrent Neural Networks (RNN)

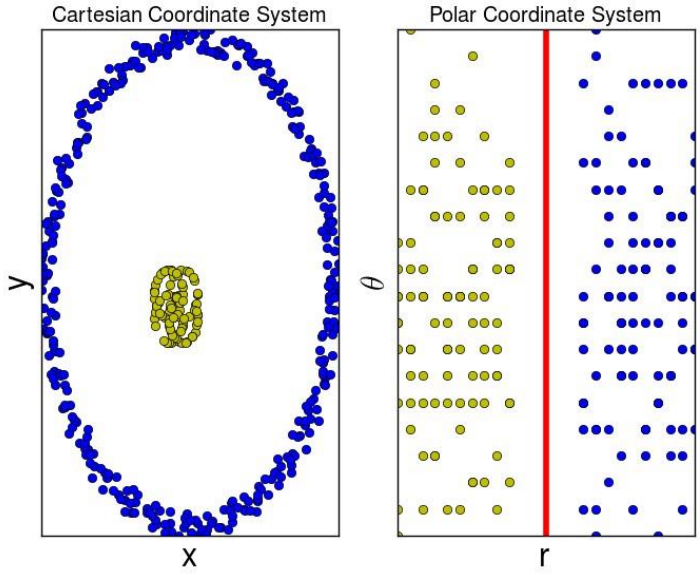


Η μορφή της αναπαράστασης των δεδομένων αποτελεί σημαντικό παράγοντα στην απόδοση των αλγορίθμων ML. Μία αναπαράσταση αποτελείται από χαρακτηριστικά (features). Για παράδειγμα ένα χρήσιμο χαρακτηριστικό στην ταυτοποίηση ομιλητή από δεδομένα ήχου, είναι η εκτίμηση του μεγέθους της φωνητικής έκτασης του ομιλητή. Έτσι, πολλά προβλήματα τεχνητής νοημοσύνης μπορούν να λυθούν με κατάλληλη σχεδίαση και επιλογή των χαρακτηριστικών για το συγκεκριμένο πρόβλημα. Το σύνολο των χαρακτηριστικών αυτών αποτελεί την αναπαράσταση των δεδομένων σε ένα πιο υψηλό και αφαιρετικό επίπεδο αντίληψης για τους υπολογιστές η οποία στην συνέχεια δίνεται σαν είσοδος σε έναν απλό ML αλγόριθμο, ο οποίος έχει μάθει να αντιστοιχεί την αναπαράσταση των δεδομένων στην επιθυμητή έξοδο.

Ένα απλό και κατανοητό παράδειγμα το οποίο δείχνει την εξάρτηση της επίδοσης των αλγορίθμων ML από την μορφή της αναπαράστασης που του δίνεται, φαίνεται στο σχήμα 2.3. Έστω ότι θέλουμε να διαχωρίσουμε τα δεδομένα μας σε δύο κλάσεις, χαράζοντας

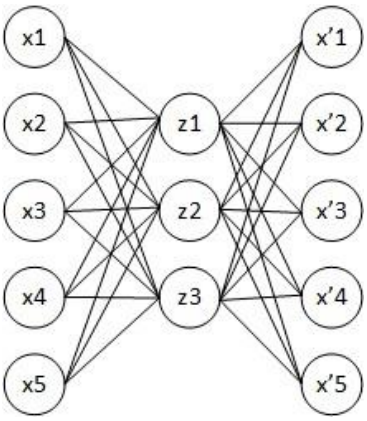
μία ευθεία μεταξύ τους. Αν αναπαραστήσουμε τα δεδομένα στο Καρτεσιανό σύστημα συντεταγμένων (αριστερό διάγραμμα), τότε η επίλυση του προβλήματος είναι αδύνατη αφού δεν υπάρχει καμία ευθεία που να διαχωρίζει τις δύο κλάσεις. Ωστόσο, αν αναπαραστήσουμε τα δεδομένα στο πολικό σύστημα συντεταγμένων (δεξί διάγραμμα), τότε το πρόβλημα λύνεται εύκολα χαράζοντας μία κάθετη ευθεία, με $r = a, a \in [r1, r2]$.

Στα περισσότερα προβλήματα τεχνητής νοημοσύνης, η επιλογή κατάλληλων χαρακτηριστικών είναι δύσκολο και χρονοβόρο έργο. Έστω ότι θέλουμε να αναγνωρίσουμε πρόσωπα σε εικόνες. Ένα χαρακτηριστικό θα μπορούσε να είναι τα μάτια. Δυστυχώς όμως, η αναγνώριση ματιών είναι ένα δύσκολο πρόβλημα αφού δεν μπορεί να περιγραφεί πάντα επακριβώς έχοντας σαν δεδομένα τις τιμές των pixel της εικόνας. Η γεωμετρική για παράδειγμα μορφή των ματιών σε μία εικόνα λήψης εξαρτάται από την γωνία λήψης της εικόνας, τον φωτισμό, τις ανακλάσεις του φωτισμού, την απόσταση από την οποία γίνεται η λήψη, την ανάλυση της κάμερας,



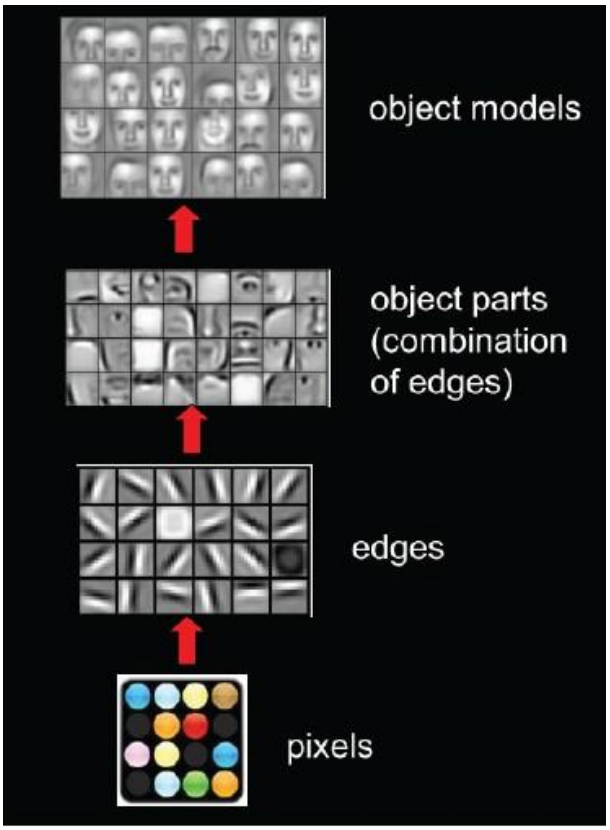
Εικόνα 3 : Παράδειγμα διαφορετικών αναπαραστάσεων των δεδομένων κτλ.

Το πρόβλημα αυτό, της επιλογής κατάλληλης αναπαράστασης, μπορεί να λυθεί χρησιμοποιώντας τεχνικές μηχανικής μάθησης για την εκμάθηση της ίδιας της αναπαράστασης. Αυτή η προσέγγιση είναι γνωστή ως *Εκμάθηση Αναπαραστάσεων (Representation Learning)*. Οι αλγόριθμοι εκμάθησης αναπαραστάσεων είναι ικανοί να "μάθουν" ένα καλό σετ χαρακτηριστικών (features). Ένα απλό παράδειγμα αλγορίθμου εκμάθησης αναπαραστάσεων είναι αυτό του Autoencoder [3] που φαίνεται στην εικόνα 4.



Εικόνα 4 : Απλό μοντέλο Autoncoder με ένα κρυφό επίπεδο.

Ο Autoencoder, στην πιο απλή του μορφή, είναι ο συνδυασμός ενός κωδικοποιητή (encoder) ο οποίος μετατρέπει τα δεδομένα εισόδου σε μία διαφορετική αναπαράσταση, και ενός αποκωδικοποιητή ο οποίος επαναφέρει την αναπαράσταση αυτή στην αρχική μορφή της αναπαράστασης των δεδομένων εισόδου. Οι Autoencoders ανήκουν στην κατηγορία των Νευρωνικών Δικτύων και είναι Unsupervised ML αλγόριθμοι. Ένα συχνά εμφανιζόμενο πρόβλημα σε εφαρμογές τεχνητής νοημοσύνης είναι η εύρεση και εξαγωγή χαρακτηριστικών *υψηλού επιπεδου* από τα δεδομένα. Τα μοντέλα *Βαθιάς Μηχανικής Μάθησης* δίνουν λύσεις σε αυτό το πρόβλημα εκμάθησης αναπαραστάσεων με την εισαγωγή χαρακτηριστικών τα οποία εκφράζονται με βάση άλλες, απλούστερες αναπαραστάσεις. Αυτή η προσέγγιση δίνει την δυνατότητα στους υπολογιστές να κατασκευάζουν σύνθετες έννοιες χρησιμοποιώντας απλούστερες. Για παράδειγμα, η αναγνώριση αντικειμένων μπορεί να εκφραστεί με έννοιες όπως το γεωμετρικό σχήμα των αντικειμένων, το οποίο με την σειρά του ορίζεται από γωνίες και περιγράμματα. Επίσης, οι γωνίες και τα περιγράμματα ορίζονται από ακμές. Στο σχήμα 5, παρουσιάζονται τα φίλτρα που έμαθε ένα μοντέλο CNN για αναγνώριση προσώπων σε εικόνες. Το συγκεκριμένο μοντέλο CNN έχει 3 κρυφά επίπεδα (hidden layers); το πρώτο κρυφό επίπεδο εξάγει από τα δεδομένα εισόδου (τιμές των πίξελ) πληροφορία σχετικά με τις ακμές, το δεύτερο, έχοντας σαν είσοδο την πληροφορία παρουσίας ακμών, εξάγει πληροφορία σχετικά με τις γωνίες και τα περιγράμματα και το τρίτο παίρνει σαν είσοδο την πληροφορία αυτή και κατασκευάζει μοντέλα αντικειμένων, δηλαδή εξάγει πληροφορία σχετικά με το γεωμετρικό σχήμα των αντικειμένων.



Εικόνα 5 : Παράδειγμα απεικόνισης των φίλτρων ενός μοντέλου CNN για αναγνώριση προσώπου

2.1 Ταξινόμηση

Στην επιβλεπόμενη μάθηση η ταξινόμηση υλοποιείται δίδοντας στα παραδείγματα μια ετικέτα (Label) ενός χαρακτηριστικού που τα διαχωρίζει από τα υπόλοιπα. Στα προβλήματα ταξινόμησης ως επί το πλείστο δίδονται δύο ετικέτες, αφού συνήθως γίνεται διαχωρισμός σε δυο κατηγορίες. Αυτές μπορεί να είναι για τον διαχωρισμό αντικειμένων σε μεγάλα ή μικρά, ψηλά ή κοντά, ή όπως στην εργασία θετικά ή αρνητικά. Έτσι δημιουργείται ένα σετ εκπαίδευσης (training set). Το σετ εκπαίδευσης είναι αυτό που θα διαχωρίζει τα παραδείγματα σε σχετικά η μη.

Ο λογιστικός ταξινομητής αποτελεί την κύρια γραμμική εξίσωση εκπαίδευσης και είναι ένας γραμμικός ταξινομητής που χαρακτηρίζεται από την παρακάτω εξίσωση :

$$Wx + b = y \quad (20)$$

όπου με x συμβολίζεται η είσοδος των δεδομένων, η οποία μπορεί να είναι ένα bit μιας εικόνας, μίας λέξεις, ενός ήχου ή γενικά ψηφιακού αρχείου, το W εκφράζει την τιμή του βάρους (weight) και το b την τάση πόλωσης (bias). Η διαδικασία εύρεσης του σωστού συνδυασμού βαρών και τάσεων ονομάζεται και εκπαίδευση του Μοντέλου. Η έξοδος y της εξίσωσης ονομάζεται πρόβλεψη (prediction), η τιμή της οποίας όσο μεγαλύτερη είναι τόσο μεγαλύτερη είναι η πιθανότητα το αποτέλεσμα να είναι σωστό. Οι τιμές του y στη λογιστική παλινδρόμηση λέγονται και logits.

Η πρόβλεψη y που δίδεται από την εξίσωση (20) μπορεί να πάρει διάφορες τιμές και το πεδίο ορισμού έχει μεγάλο εύρος. Η μετατροπή του σε τιμές εντός πεδίου ορισμού [0,1] υλοποιείται με την συνάρτηση Softmax (21) (Bridle, 1990) όπου τα αποτελέσματα περιορίζονται και εκφράζονται πλέον σε πιθανότητες. Οι μεγάλες τιμές της εξίσωσης (20) γίνονται πιθανότητες που τείνουν 1 και οι μικρές κοντά στο 0. Η εξίσωση περιγράφεται παρακάτω:

$$S(y_i) = \frac{e^{y_i}}{\sum_i e^{y_i}}$$

Η συνάρτηση S (Softmax) των τιμών y_i που εξάγονται από τον γραμμικό ταξινομητή της εξίσωσης 20, ισούται με το κλάσμα του e υψωμένα στο y_i προς το συνολικό άθροισμα των αποτελεσμάτων e υψωμένα στο y_i.

Οι ετικέτες ή labels οι οποίες ταξινομούνται στη σωστή κλάση τείνουν στο 1 ενώ οι υπόλοιπες στο 0. Κάθε ετικέτα αντιπροσωπεύεται από ένα διάνυσμα και έχει την τιμή 1 για τη σωστή κλάση και 0 για τις υπόλοιπες. Δημιουργούνται πίνακες με πολλά 0 και ελάχιστα 1. Αυτό το πρόβλημα λύνεται με τη συνάρτηση cross entropy Hopfield (1987) and Bishop (1995) που περιγράφεται από τη σχέση:

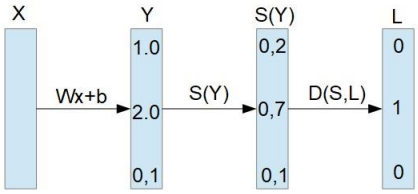
$$D(S, L) = -\sum_i L_i \log(S_i)$$

Η Συνάρτηση μετατρέπει τις πιθανότητες σε τιμές 0 και 1. Αν η πιθανότητα είναι μεγάλη τότε δίδει τιμή 1 αν είναι μικρή 0. Οι Τιμές S είναι τα αποτελέσματα της συνάρτησης Softmax ενώ το L (από το Logits) εκφράζει τις τιμές 0 ή 1.

Από όλες τις συναρτήσεις μαζί συνεπάγεται ότι :

$$D(S(Wx+b), L) = -\sum_{j} L_j \log\left(\frac{e^{Wx+b}}{\sum_j e^{Wx+b}}\right)$$

και σχηματικά η διαδικασία που περιγράφηκε παρουσιάζεται στην εικόνα 18



Εικόνα 6 : Διαδικασια τιμών εισοδου χ σε logits

Η ανωτέρω συνάρτηση (23) ονομάζεται multinomial logistic classification ή πολυεστιακή λογιστική ταξινόμηση. Στόχος είναι η μείωση της cross entropy και αυτό γίνεται υπολογίζοντας την απώλεια εκπαίδευσης (Loss) ή Average cross entropy η οποία ισούται με:

$$Loss = \frac{1}{N}\sum_{j} D(S(Wx_j + b), L_j)$$

Αυτό που επιδιώκεται είναι η ελαχιστοποίηση της τιμής της συνάρτησης στη διάρκεια της εκπαίδευσης. Όσο μειώνεται η τιμή της loss το μοντέλο εκπαιδεύεται. Μαθηματικά όπως φαίνεται από την εξίσωση 24, ισούται με το σύνολο των τιμών που προκύπτουν από την διαδικασία της πολυεστιακής λογιστικής ταξινόμησης, προς τα παραδείγματα N.

2.2 Αρχικοποίηση βαρών

Από την συνάρτηση

$$D(S(Wx+b), L) = -\sum_{i} L_i \log\left(\frac{e^{Wx+b}}{\sum_i e^{Wx+b}}\right)$$

παρατηρείται ότι οι μοναδικές μεταβλητές είναι τα βάρη W και η τάση b . Η σωστή εύρεσή των βαρών και των τάσεων είναι αυτή που θα εκπαιδεύσει το δίκτυο. Οι αρχικές τιμές που θα δοθούν (αρχικοποίηση βαρών) ως w_0 και b_0 και οι τιμές που θα επακολουθήσουν ώστε να φτάσει το δίκτυο στις σωστές ή σωστότερες τιμές ώστε να εκπαιδευτεί. Λύνεται με αλγόριθμους στοχαστικής βελτιστοποίησης. Η βελτιστοποίηση είναι μια επαναλαμβανόμενη ακολουθία εύρεσης βαρών και τάσεων όπως φαίνεται στις παρακάτω εξισώσεις:

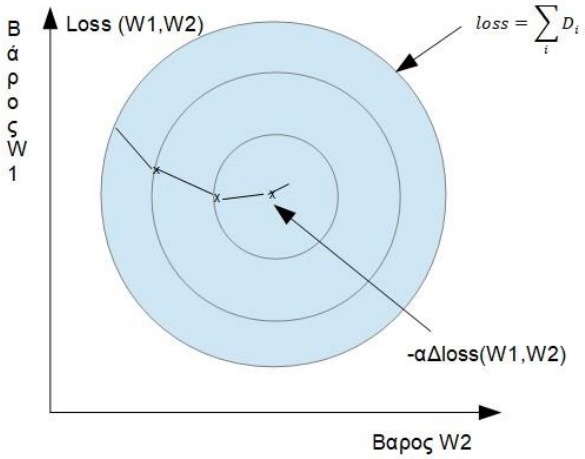
$w \leftarrow w - \Delta_w Loss$

και

$b \leftarrow b - \Delta_b Loss$

όπου w και b είναι τα βάρη και οι τάσεις αντίστοιχά και $\Delta_w Loss$ $\Delta_b Loss$ οι παράγωγοι συνάρτησης απώλειας βαρών και τάσεων.

Η μείωση της συνάρτησης κόστους γίνεται με αλγορίθμους βελτιστοποίησης. Ο gradient descent είναι ένας γνωστός αλγόριθμος βελτιστοποίησης. Ο gradient descent βρίσκει την παραγωγό (διαφορά) των δυο βαρών w_1 και w_2 και επαναλαμβάνεται όσο αυτή μειώνεται μέχρι την επιθυμητή λύση όπως περιγράφεται στην εικόνα 20. Είναι ένας κάλος αλγόριθμος βελτιστοποίησης αλλά έχει ένα μεγάλο μειονέκτημα. Σε μεγάλα σετ δεδομένων έχει αποδειχθεί ότι είναι μια χρονοβόρα και κοστοβόρα σε πόρους διαδικασία καθώς απαιτείται ο υπολογισμός της παραγώγου όλων των δεδομένων σε απαγορευτικά πολλές επαναλήψεις. Το πρόβλημα αυτό λύνεται με τους στοχαστικούς αλγορίθμους βελτιστοποίησης.



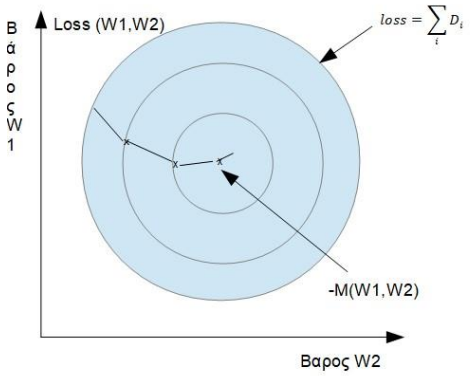
Εικόνα 7 : Μειωση του κόστους με αλγοριθμο βελτιστοποίησης Gradient Descend

2.3 Στοχαστική βελτιστοποίηση (Stochastic optimization)

Οι μέθοδοι στοχαστικής βελτιστοποίησης είναι μέθοδοι που παράγουν και χρησιμοποιούν τυχαίες μεταβλητές για την επίλυση του προβλήματος. Περιλαμβάνουν επίσης μεθόδους με τυχαίες επαναλήψεις. (Spall, J. C.2003). Οι μέθοδοι στοχαστικής βελτιστοποίησης γενικεύουν τις ντετερμινιστικές μεθόδους για τα προσδιοριστικά προβλήματα. Ο λόγος για τον οποίο χρησιμοποιείται αυτή η μέθοδος είναι διότι αν χρησιμοποιηθεί η μέθοδος gradient descent θα πρέπει να χρησιμοποιηθούν όλα τα δεδομένα κάτι που κάνει τον υπολογισμό πολύπλοκο.

Στους στοχαστικούς αλγόριθμους βελτιστοποίησης χρησιμοποιείται ένας εκτιμητής (estimator). Ο εκτιμητής υπολογίζει μια μικρή ποσότητα της συνάρτησης κόστους των εκπαιδευμένων δεδομένων τυχαία. Έτσι, υπολογίζει την απώλεια για το δείγμα της παραγώγου ως προς το δείγμα. Ακόμα και αν κατά την έναρξη της εκπαίδευσης δεν παρουσιάζονται τα αναμενόμενα αποτελέσματα κατά την εξέλιξή της αυτό επιτυγχάνεται. Η

διαδικασία αυτή λέγεται stochastic gradient descent. Αυτό που την κάνει και πετυχαίνει είναι το Momentum ή ορμή. Η λειτουργία της είναι απλή και αποδοτική. Κατά τη διάρκεια της εκπαίδευσης το δείγμα παραδειγμάτων τείνει στη σωστή λύση και σε κάποιες περιπτώσεις απομακρύνεται από αυτή. Το momentum κρατά μόνο τις περιπτώσεις που τα τυχαία παραδείγματα προσδίδουν βελτίωση.

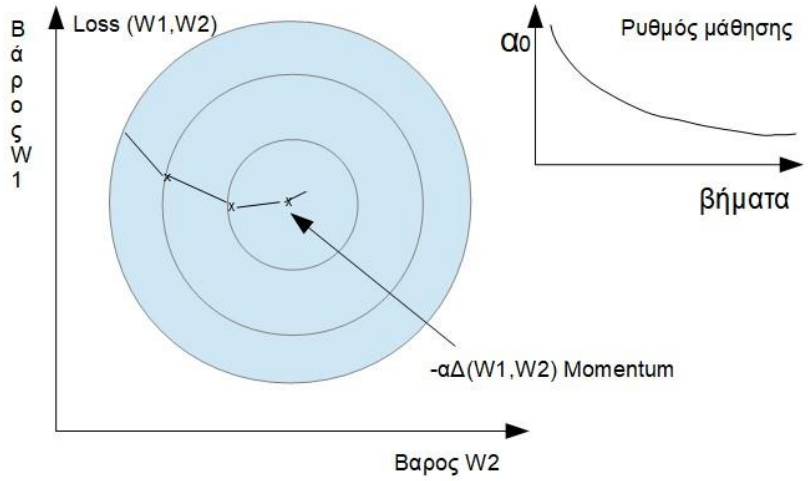


Εικόνα 8 : Βελτιστοποίηση με ορμή

Η ορμή ακολουθεί την παρακάτω σχέση:

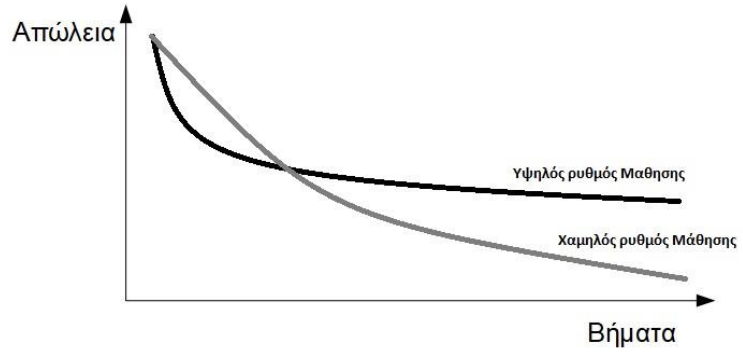
$$M \leftarrow 0.9M + \Delta loss$$

Ένα άλλο χαρακτηριστικό είναι η απόσβεση ποσοστού μάθησης (Learning rate decay), η οποία συμβολίζεται με α. Κατά τη διάρκεια της εκπαίδευσης όσο υπολογίζεται το momentum παρουσιάζεται η παράμετρος αυτή, η οποία κατά την διάρκεια της εκπαίδευσης πρέπει να μειώνεται .



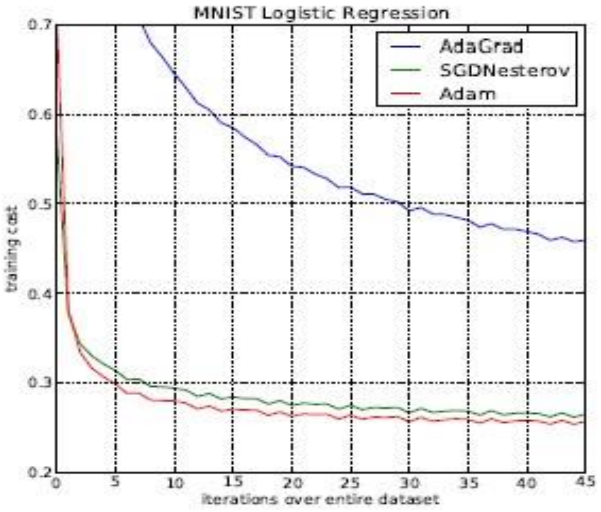
Εικόνα 9 : learning rate decay

Η επιλογή του σωστού Ρυθμού μάθησης (Learning rate) είναι καθοριστικής σημασίας για την εκπαίδευση του νευρωνικού δικτύου. Κατά την εκπαίδευση του ρυθμού μάθησης, η χρήση μίας υψηλότερης τιμής μάθησης δεν σημαίνει ότι το μοντέλο εκπαίδευσης μαθαίνει απαραίτητα γρηγορότερα. Έχει αποδειχθεί ότι χρησιμοποιώντας μικρούς ρυθμούς μάθησης το μοντέλο αποδίδει καλύτερα. Παρατηρώντας την εικόνα 22 φαίνεται, ότι σε υψηλό ρυθμό μάθησης η μάθηση ξεκινά γρηγορότερα σε σχέση με έναν χαμηλότερο ρυθμό μάθησης , αλλά στη συνέχεια αυτός που αποδίδει καλύτερα είναι ο χαμηλός που τείνει στο μηδέν της γραφικής παράστασης σε σχέση με τον υψηλό.



Εικόνα 10 : Σύγκριση Ρυθμών Μαθησης

Ο Αλγόριθμος ADAGRAD (ADAptive GRAdient Descent) είναι μια βελτιστοποίηση του SGD (Stochastic Gradient Descent) ο οποίος υπολογίζει μόνος του την Αρχικοποίηση ρυθμού μάθησης (learning rate), Διάσπαση ρυθμού μάθησης (learning rate decay), Ορμή (momentum) ,μέγεθος παρτίδας παραδειγμάτων (batch size), και αρχικοποίηση των βαρών (weight initialization). ο ADAGRAD υλοποιεί τα ανωτέρω εκτός του batch size και αρχικοποίησης των βαρών. Αφήνει τον ρυθμό μάθησης να προσαρμοστεί σε σχέση με τις παραμέτρους. όταν οι παράμετροι είναι μη συχνοί κάνει μεγάλες αλλαγές ενώ όταν είναι συχνοί μικρές. ο καλύτερος optimizer όπως φαίνεται στην εικόνα 22 είναι ο ADAM optimizer (ADAptive Momentum) ο οποίος είναι βελτίωση των προηγουμένων βελτιστοποιητών υπολογίζει όσα και ο ADAGRAD και επίσης υπολογίζει για κάθε παράμετρο το ρυθμό μάθησης και τις αλλαγές της ορμής (Momentum) (Diederik P. Kingma and Jimmy Lei Ba 2015)

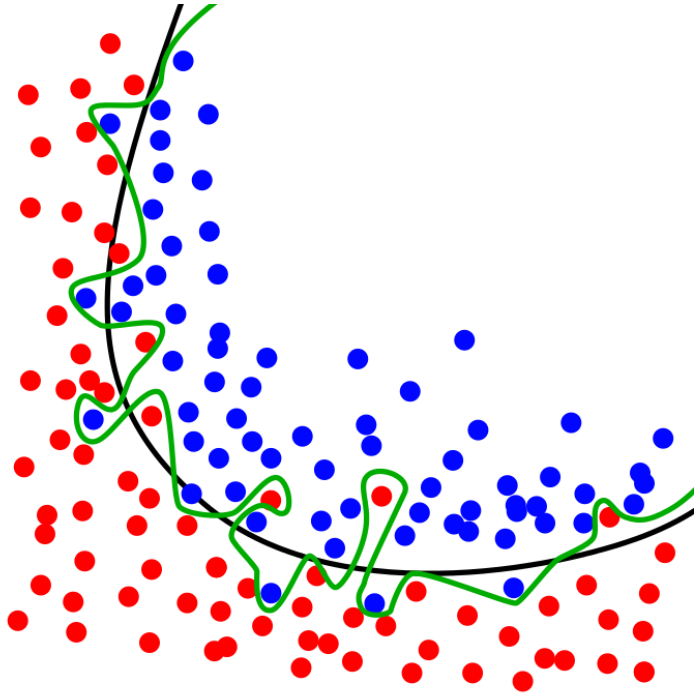


Εικόνα 11 : Απόδοση αλγοριθμων βελτιστοποίησης (πηγή: Diederik P. Kingma and Jimmy Lei Ba 2015)

2.4 Υπερπροσαρμογή (overfitting)

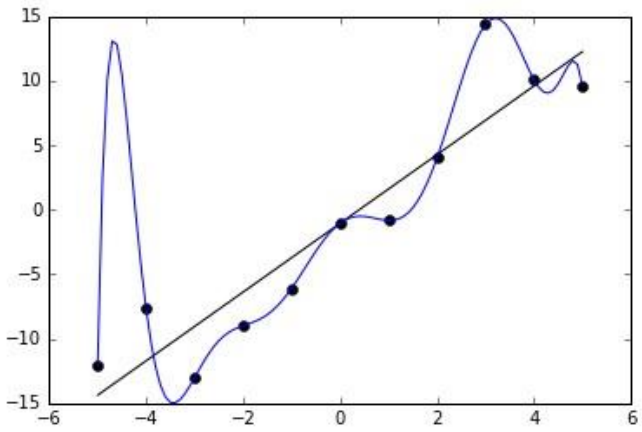
Υπερπροσαρμογή είναι η παραγωγή μιας ανάλυσης που αντιστοιχεί πάρα πολύ ή με ακρίβεια σε ένα συγκεκριμένο σύνολο δεδομένων και συνεπώς μπορεί να μην ικανοποιήσει πρόσθετα δεδομένα ή να προβλέψει αξιόπιστες μελλοντικές παρατηρήσεις (Oxford Dictionaries Online).

Ένα υπερπροσαρμοστικό μοντέλο είναι ένα στατιστικό μοντέλο που περιέχει περισσότερες παραμέτρους από αυτές που δικαιολογούνται από τα δεδομένα (Everitt And Skrondal , 2010). Στην εικόνα 12 παρουσιάζεται ένα πρόβλημα ταξινόμησης στο οποίο πρέπει να διαχωρίσουμε τις μπλε και κόκκινες κουκίδες. Οι κουκίδες μπορεί να είναι λέξεις με θετικό συναίσθημα ή και εικόνες που να περιγράφουν ένα σκύλο ή μία γάτα. Τα νευρωνικά δίκτυα είναι τόσο πολύπλοκά λόγω της αρχιτεκτονικής τους (Πολλοί κόμβοι, πολλά κρυφά επίπεδα) που κατά την επίλυση του προβλήματος του διαχωρισμού μπορούν να διαχωρίσουν όλες τις κουκκίδες δημιουργώντας πολυώνυμα μεγάλου βαθμού. Έτσι υλοποιείται μια λύση όπως την πράσινη γραμμή της εικόνας. Στην πραγματικότητα αυτό που είναι αποδοτικότερο για το δίκτυο είναι η μαύρη καμπύλη όπου τα παραδείγματα διαχωρίζονται με μικρότερη ακρίβεια αφήνοντας κάποια παραδείγματα εκτός αλλά δημιουργώντας μία ποιοτικότερη λύση.



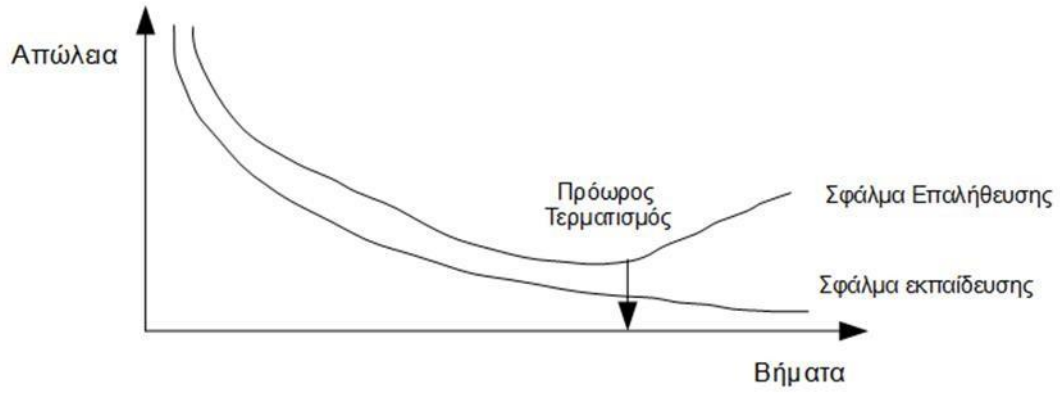
Εικόνα 12 : Υπερπροσαρμογή παραδειγμάτων (Πηγή: Wikipedia)

Για να γίνει πιο κατανοητό το πρόβλημα της υπερπροσαρμογής θα αναλυθεί η εικόνα 13. Για την εύρεση της σωστής κλίσης βάση των παραδειγμάτων που στην περίπτωσή της εικόνας είναι οι μαύρες κουκκίδες πρέπει να βρεθεί μια ευθεία με κλίση η οποία μπορεί να μην περνάει από όλες τις κουκίδες αλλά ποιοτικά τις ακολουθά. Ένα όμως δίκτυο πολλών επιπέδων δεν θα αρκεστεί σε αυτό αλλά θα προσπαθήσει να λύσει το πρόβλημα με απόλυτο και υπερβολικό τρόπο. Η λύση δίδεται από ένα πολυώνυμο μεγάλου βαθμού όπως η μπλε γραμμή την εικόνας που αναφερόμαστε.



Εικόνα 13 : Υπερπροσαρμογή (Πηγή: Wikipedia)

Το πρόβλημα που αντιμετωπίζουμε είναι ότι λύνεται απολυτά ο διαχωρισμός των δεδομένων εκπαίδευσης αλλά το δίκτυο παρουσιάζει απόκλιση όταν δίδουμε τα δεδομένα επαλήθευσης. Ένας τρόπος αντιμετώπισης του Overfitting είναι ο πρόωρος τερματισμός της εκπαίδευσης στο σημείο που ξεκινάει η απόκλιση των σφαλμάτων. Αυτό όμως περιορίζει την εκπαίδευση του δικτύου. Τα παραπάνω φαίνονται διαγραματικά στην εικόνα 14.



Εικόνα 14 : Υπερπροσαρμογή σην συναρτηση κόστους

Ένας άλλος τρόπος για να αποφευχθεί η υπερπροσαρμογή (overfitting) είναι η ομαλοποίηση (Regularization) του δικτύου. Κατά την εκπαίδευση του όταν αρχίζει να υπερπροσαρμόζεται τα βάρη των νευρώνων αρχίζουν να παίρνουν μεγάλες τιμές. Η ομαλοποίηση λειτουργεί με την επιβολή μιας παραμέτρου α στη συνάρτηση κόστους όπου δεν επιτρέπει να δίδονται υψηλές τιμές τα βάρη. Υπάρχουν δυο μέθοδοι ομαλοποίησης η L1 που ονομάζεται και LASSO και η L2 η οποία αποτελεί εξέλιξη της L1 γνωστή ως Ridge Regression.

Η μέθοδος ομαλοποίησης L1 κατά τον υπολογισμό της συνάρτησης απώλειας προσθέτει μια ποινή βάρους α στο απολυτό σύνολο των βαρών.

$$loss \leftarrow loss + weight\ penalty$$

$$loss \leftarrow \sum_i (y_i - \hat{y}_i)^2 + a \sum_i |w_i|$$

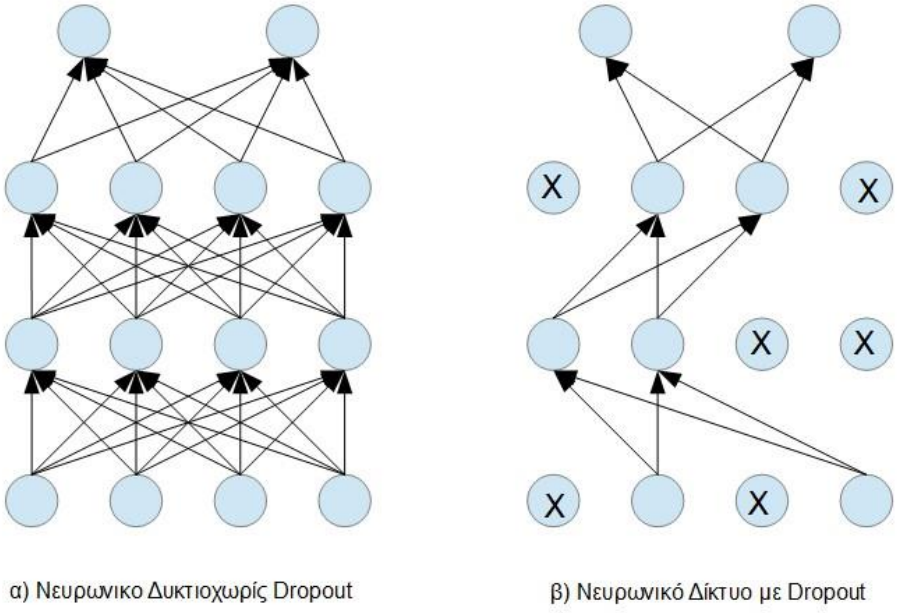
Στην Ridge Regression η L2 προστίθεται ποινή βάρους το τετράγωνο των συνολικών βαρών.

$$loss \leftarrow loss + weight\ penalty$$

$$loss \leftarrow \sum_i (y_i - \hat{y}_i)^2 + a \sum_i w_i^2$$

2.5 Απόρριψη (Dropout)

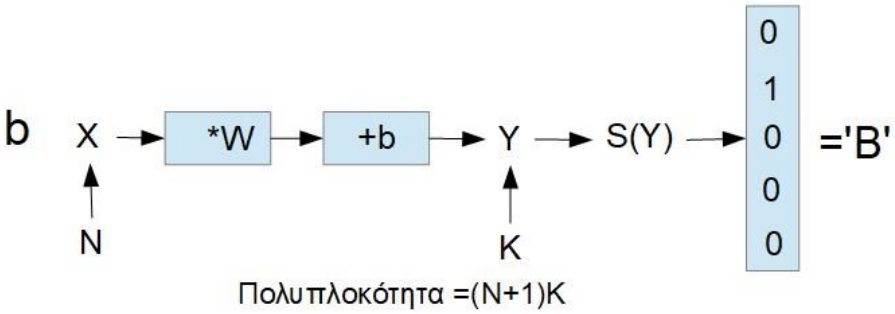
Η απόρριψη (Dropout) (Nitish Srivastava,Georey Hinton, Alex Krizhevsky,Ilya Sutskever, Ruslan Salakhutdinov, 2014) είναι μια τεχνική αποτροπής της υπερφόρτωσης (Overfitting) και παρέχει έναν τρόπο να συνδυάζει αποτελεσματικά όλες σχεδόν τις αρχιτεκτονικές νευρωνικών δικτύων αποτελεσματικά. Ο όρος "dropout" αναφέρεται στην απομάκρυνση κόμβων (κρυφές και ορατές) σε ένα νευρωνικό δίκτυο. Με την απομάκρυνση κόμβου, εννοούμε προσωρινά την απομάκρυνσή του από το δίκτυο, μαζί με όλες τις εισερχόμενες και εξερχόμενες συνδέσεις, όπως φαίνεται στην εικόνα 15 . Η επιλογή των μονάδων που θα απομακρυνθούν είναι τυχαία. Στην απλούστερη περίπτωση κάθε μονάδα διατηρείται με μια σταθερή πιθανότητα p ανεξάρτητα από άλλες μονάδες, όπου το p μπορεί να επιλεγεί χρησιμοποιώντας ένα σύνολο επικύρωσης ή μπορεί απλά να ρυθμιστεί στο 0,5, το οποίο φαίνεται να είναι κοντά στη βέλτιστη, στις περισσότερες αρχιτεκτονικές δικτύων . Η απόρριψη υλοποιείται μόνο για το σετ δεδομένων εκπαίδευσης. Για το σετ επικύρωσης τα δεδομένα περνάνε από όλους τους κόμβους.



Εικόνα 15 : Dropout με απόρριψη 0.5

2.6 Πολυπλοκότητα Γραμμικών μοντέλων

Σημαντικός παράγοντας για την ανάλυση των νευρωνικών δικτύων είναι ο υπολογισμός της πολυπλοκότητας του μοντέλου. Όπως παρουσιάζεται στην εικόνα 28 για τον υπολογισμό της πολυπλοκότητας του του γραμμικού μοντέλου που παρουσιάζεται αρκεί να πολλαπλασιάσουμε το σύνολο των εισόδων χ και να το πολλαπλασιάσουμε με τις πιθανές εξόδους που εξάγει το μοντέλο. Αν για παράδειγμα εισάγουμε μια εικόνα με 100 εικονοστοιχεία και από την έξοδο αναμένουμε 5 πιθανά αποτελέσματα τότε η πολυπλοκότητα είναι (100+1)*5 που ισούται με 505.

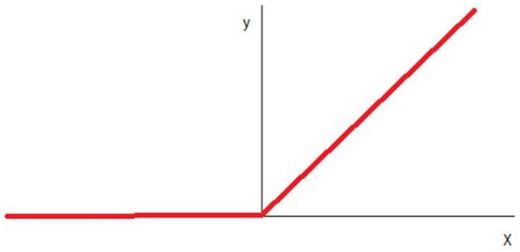


Εικόνα 16 : Υπολογισμός πολυπλοκότητα μοντελου Xw+b

Η εκτέλεση της λειτουργίας της εικόνας 16 μαθηματικά είναι ουσιαστικά ένας πολλαπλασιασμός πινάκων.

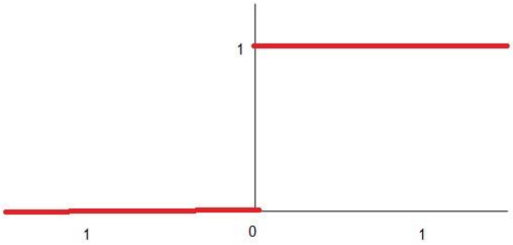
2.7 Συνάρτηση ενεργοποίησης RELU

Επικρατέστερη συνάρτηση ενεργοποίησης για τα βαθιά νευρωνικά δίκτυα έχει επικρατήσει η συνάρτηση RELU (Abien Fred M. Agarap 2018).Η συνάρτηση ενεργοποίησης RELU (REctified Linear Unit),είναι μία απλή μη γραμμική συνάρτηση για την οποία όπου x>0 y=x και x<0 y=0. Η γραφική της παράσταση περιγράφεται στην εικόνα 17.



Εικόνα 17 : Συνάρτηση Relu

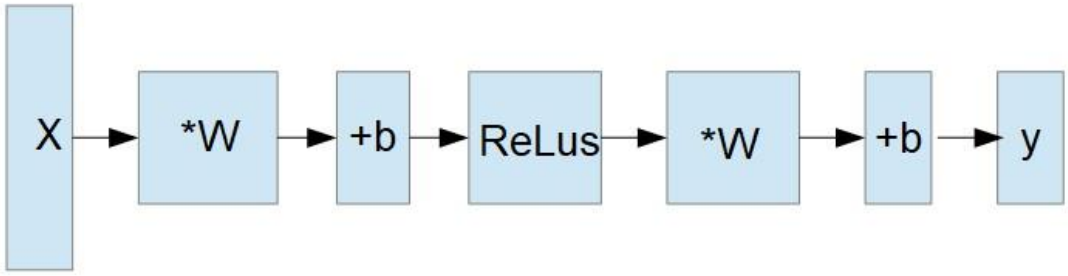
Η συνάρτηση RELU είναι αποδοτικότερη από τις κλασικές συναρτήσεις ενεργοποίησης όπως η sigmoid και η tanx. Αυτό συμβαίνει διότι η παράγωγος της RELU παίρνει δύο διακριτές τιμές [0,1]. Για χ <0 η παράγωγος παίρνει την τιμή 0 και για χ>0 την τιμή 1.



Εικόνα 18 : Παράγωγος Relu

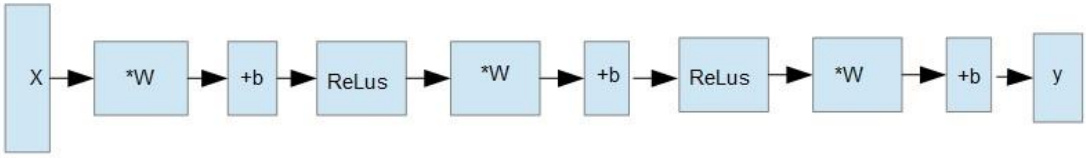
2.8 Πολυεπίπεδα Νευρωνικά δίκτυα (Multilayer Neural Networks)

Η προσθήκη ενός κρυμμένου στρώματος σε ένα δίκτυο του επιτρέπει να διαμορφώνει πιο σύνθετες λειτουργίες. Επίσης, με τη χρήση μιας μη γραμμικής συνάρτησης ενεργοποίησης στο κρυφό στρώμα του επιτρέπει να μοντελοποιεί μη γραμμικές λειτουργίες. Στην εικόνα 19 αναπαρίσταται παραπάνω ένα νευρωνικό δίκτυο δυο επιπέδων, ενός κρυφού και ενός επιπέδου εξόδου με συνάρτηση ενεργοποίησης RELU.



Εικόνα 19 : Νευρωνικό δίκτυο 2 επιπέδων

Στην εικόνα 20 απεικονίζεται ένα πολυεπίπεδο νευρωνικό δίκτυο δυο κρυφών επιπέδων και ενός επιπέδου εξόδου.

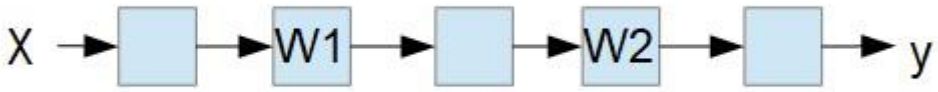


Εικόνα 20 : Νευρωνικό δίκτυο 3 επιπέδων

Το πρώτο επίπεδο αποτελείται ουσιαστικά από το σύνολο των βαρών και των τάσεων που εφαρμόζεται στην είσοδο X και ενεργοποιείται μέσω της συνάρτησης ενεργοποίησης Relu στην έξοδό του. Η έξοδος αυτού του στρώματος τροφοδοτεί το επόμενο επίπεδο, αλλά δεν είναι παρατηρήσιμη εκτός του δικτύου, αφού πλέον αποτελεί κρυφό επίπεδο. Όμοια το δεύτερο επίπεδο αποτελείται από το σύνολο των βαρών και των τάσεων που εφαρμόζεται στη είσοδο του X και ενεργοποιείται μέσω της συνάρτησης ενεργοποίησης Relu. Τέλος το επίπεδο εξόδου αποτελείται από τα βάρη και τις τάσεις, και μέσω της συνάρτηση Softmax δημιουργεί τις πιθανότητες και τα αποτελέσματα.

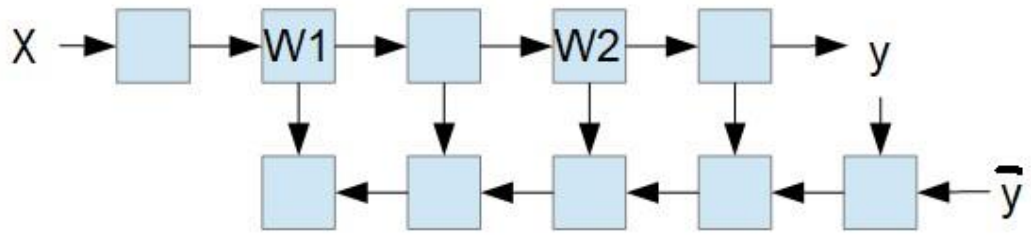
2.9 Back Propagation

Ο αλγόριθμός Back Propagation, για τον οποίο έχει γίνει αναφορά σε προηγούμενες ενότητες, εφαρμόζεται και στην βαθιά μηχανική μάθηση. Ο Αλγόριθμος Back Propagation είναι μια γενίκευση του κανόνα δέλτα σε πολυεπίπεδα δίκτυα πρόσθιας τροφοδότησης, που χρησιμοποιεί τον κανόνα της αλυσίδας (chain rule) για να υπολογίσει διαδοχικά τις κλίσεις για κάθε στρώμα. Με τον κανόνα της αλυσίδας γίνεται η εύρεση της παραγώγου του μοντέλου του δικτύου με εύκολο και γρήγορο τρόπο. Για να γίνει κατανοητός ο κανόνας της αλυσίδας απλοποιούμε το μοντέλο σε απλές λειτουργίες όπως απεικονίζεται στην εικόνα 21:



Εικόνα 21 : Απλοποιημένο μοντελο 2 επιπεδων

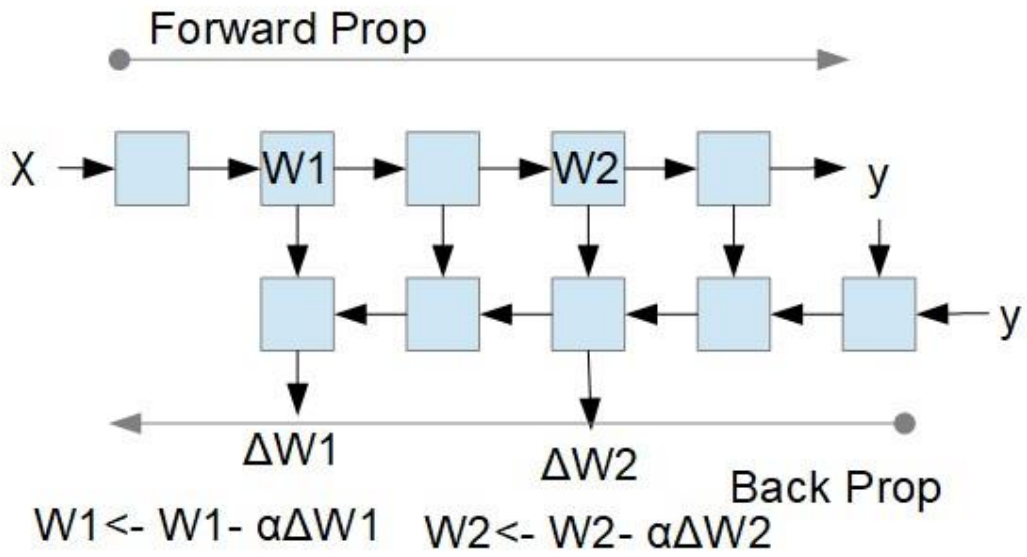
Το δίκτυο είναι μια σειρά από απλές πράξεις ή λειτουργίες όπως γραμμικοί μετασχηματισμοί και συναρτήσεις ενεργοποίησης. Από την είσοδο X εισέρχονται τα δεδομένα τα οποία περνούν τις λειτουργίες που μπορεί να είναι γραμμικοί μετασχηματισμοί ή συναρτήσεις ενεργοποίησης και βγαίνουν στην έξοδο y. Για τον υπολογισμό της παραγώγου υλοποιείται άλλος ένας γράφος όπως το σχήμα:



Εικόνα 22 : Σχηματική παρουσιαση παραγώγου $y-\hat{y}$

Ο backpropagation για την υλοποίησή όπως φαίνεται στην εικόνα 23 χρησιμοποιεί ένα στοχαστικό αλγόριθμο βελτιστοποίησης κλίσης και μία συνάρτηση απώλειας. Λειτουργεί σε δύο φάσεις την φάση διάδοσης (forward prop) και την φάση οπισδοδιάδοσης σφάλματος (Back Prop). Κατά τη φάση της διάδοσης με τον στοχαστικό αλγόριθμο βελτιστοποίησης ρυθμίζεται το βάρος των νευρώνων υπολογίζοντας την κλίση μέσω της συνάρτησης απώλειας

(loss function). Έτσι όταν το διάνυσμα εισόδου παρουσιάζεται στο δίκτυο, μεταδίδεται προς τα εμπρός μέσω του δικτύου, επίπεδο-επίπεδο, μέχρι να φτάσει στο στρώμα εξόδου. Στη συνέχεια, η έξοδος του δικτύου συγκρίνεται με την επιθυμητή έξοδο μέσω της loss function. Η τιμή σφάλματος που προκύπτει υπολογίζεται για καθένα από τους νευρώνες στο στρώμα εξόδου. Οι τιμές σφάλματος στη συνέχεια μεταδίδονται από την έξοδο πίσω μέσω του δικτύου, έως ότου κάθε νευρώνας έχει μια σχετική τιμή σφάλματος που αντανακλά τη συμβολή του στην αρχική έξοδο. Το Backpropagation χρησιμοποιεί αυτές τις τιμές σφάλματος για να υπολογίσει την κλίση της συνάρτησης απώλειας loss function. Κατά τη φάση της οπισθοδιάδοσης (back prop), αυτή η κλίση τροφοδοτείται στη μέθοδο βελτιστοποίησης, η οποία με τη σειρά της ενημερώνει τα βάρη, σε μια προσπάθεια να ελαχιστοποιηθεί η loss function .

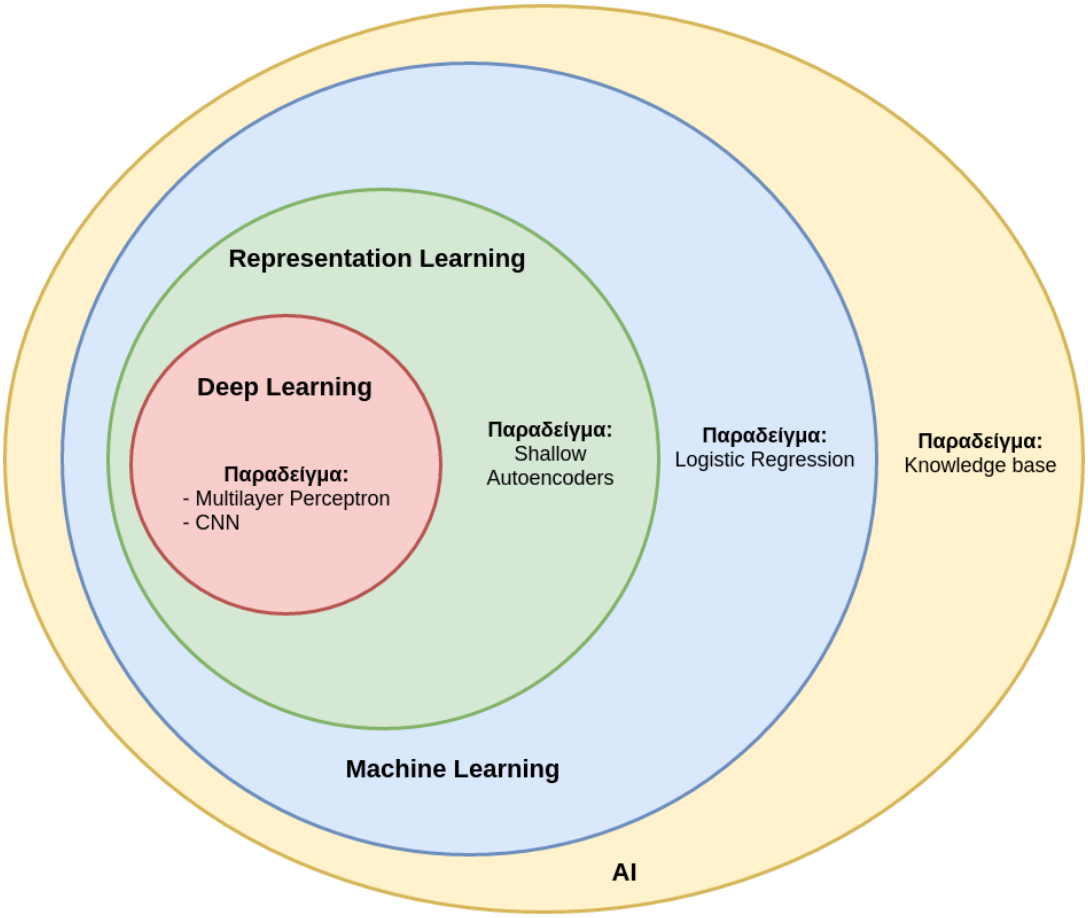


Εικόνα 23 : Σχηματική απεικόνιση Back Propagation

Η ανωτέρω διαδικασία επαναλαμβάνεται πολλές φορές στα αρχικά σας ενημερώνοντας τα βάρη και τις τάσεις έως ότου βελτιστοποιηθεί ολόκληρο το μοντέλο.

Συνοψίζοντας, η βαθιά μηχανική μάθηση, είναι μία υποκατηγορία του ML και ένας χαρακτηριστικός αντιπρόσωπος της σύγχρονης τεχνητής νοημοσύνης. Πιο συγκεκριμένα, είναι ένα είδος μηχανικής μάθησης, η οποία προσδίδει σε υπολογιστικά συστήματα ευφυΐα, δηλαδή την ικανότητα εκμάθησης με την χρήση εμπειρίας και δεδομένων. Σύμφωνα με τους *Ian Goodfellow, Yoshua Bengio και Aaron Courville*, η μηχανική μάθηση είναι η μόνη βιώσιμη προσέγγιση στην κατασκευή συστημάτων AI τα οποία μπορούν να αντεπεξέλθουν σε πολύπλοκα περιβάλλοντα και προβλήματα [8]. Η βαθιά μηχανική μάθηση καταφέρνει να μαθαίνει να αναπαριστά τον κόσμο ως μία ένθετη ιεραρχία εννοιών όπου η κάθε έννοια ορίζεται σε σχέση με άλλες πιο απλές έννοιες, και πιο αφηρημένες μορφές αναπαραστάσεων σε σχέση με λιγότερο αφηρημένες. Από το διάγραμμα Venn που βλέπουμε στο εικόνα 24

παρατηρούμε ότι η βαθιά μηχανική μάθηση ανήκει στην κατηγορία της εκμάθησης αναπαραστάσεων, η οποία με την σειρά της είναι ένα είδος μηχανικής μάθησης που χρησιμοποιείται για την κατασκευή νοούμενων συστημάτων.



Εικόνα 24 : Διάγραμμα Venn που δείχνει πως η επιστήμη της βαθιάς μάθησης ανήκει στην κατηγορία τεχνικών εκμάθησης αναπαραστάσεων, οι οποίες τεχνικές ανήκουν με την σειρά τους στην ευρύτερη επιστήμη της μηχανικής μάθησης

3. Βιβλιογραφία

[1] Arel, I., Rose, D. C., & Coop, R. (2009). DeSTIN: A Scalable Deep Learning Architecture with Application to High-Dimensional Robust Pattern Recognition. In Proceedings of AAAI Fall Symposium: Biologically Inspired Cognitive Architectures.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. *"ImageNet Classification with Deep Convolutional Neural Networks*". In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *"Advances in Neural Information Processing Systems 25*", pages 1097–1105. Curran Associates, Inc., 2012.

[3] Pierre Baldi. *"Autoencoders, unsupervised learning, and deep architectures.*". ICML unsupervised and transfer learning, 27(37-50):1, 2012.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *"Deep Learning*". Book in preparation for MIT Press, 2016. URL *http://www.deeplearningbook.org*.