Artificial Intelligence Neural Networks Training in the classification model

A Neural Network data model by training the classification model of a breast cancer prediction output in MATLAB.

Michail M, Chaidos I, Manios A

Computer science year 3  ARTIFICIAL INTELLIGENCE



**SCHOOL OF ARCHITECTURE, COMPUTING & ENGINEERING**

*BSc in Computer Science*

**Michail Markou, Chaidos John, Manios Athanasios**

*CN6003 – COMPUTER AND NETWORK SECURITY*

**UEL NUMBER**

*2020732 (MM), 1966133 (CJ), 2020737 (MA)*

***Date***

11/12/2021

Contents

[1. Abstract 1](#_Toc90833818)

[2. Machine Learning Algorithms 1](#_Toc90833819)

[2.1. Description of input Data Set 1](#_Toc90833820)

[2.2. Problem Solving & Data Analysis 3](#_Toc90833821)

[2.2.1. Super Vector Machine (SVM) - fitcsvm 4](#_Toc90833822)

[2.2.2. Artificial Layered Neural Network (AΝΝ) 12](#_Toc90833823)

[2.3. Conclusion (Michail Markou) 33](#_Toc90833824)

[Bibliography 34](#_Toc90833825)

[Appendix 35](#_Toc90833826)

[Glossary 35](#_Toc90833827)

[Figure 1 Word Cloud 1](#_Toc90833790)

[Figure 2 read input dataset self-documenting code in MATLAB 3](#_Toc90833791)

[Figure 3 SVM algorithm type models 4](#_Toc90833792)

[Figure 4 Hyperplane for 3D non-linear separation by scaling (before apply the data to kernel function e.g., cubic or linear) and penalty the data 8](#_Toc90833793)

[Figure 5 Scatter Diagram of B-M Types 10](#_Toc90833794)

[Figure 6 SVM vs Neural Network on various train time, Accuracy (validation), Total Cost, prediction speed, Model Type info 11](#_Toc90833795)

[Figure 7 SVM vs NN confusion Matrix 11](#_Toc90833796)

[Figure 8 SVM vs NN ROC compare shows Top Right Best Results of summary overall confusion matrix for decision making on point (Quadric SVM) 12](#_Toc90833797)

[Figure 9 Layered NN 12](#_Toc90833798)

[Figure 10 steps of Machine Learning (ML) 13](#_Toc90833799)

[Figure 11 training curve 15](#_Toc90833800)

[Figure 12 Local minimum stuck (both lines must) but close to solution function by minimizing the total expected (from data) error 16](#_Toc90833801)

[Figure 13 Example of strong neural network failing (overfitting) on new data classification (there is not a local minimum error cause of train data success) 16](#_Toc90833802)

[Figure 14 Loss function for gradient descent 19](#_Toc90833803)

[Figure 15 GD (gradient descent) CG (conjugate gradient) are slowest in speed but good in memory consumption 19](#_Toc90833804)

[Figure 16 high-level machine learning process for supervised learning scenarios 20](#_Toc90833805)

[Figure 17 view(net) 20](#_Toc90833806)

[Figure 18 Layer transfer function 21](#_Toc90833807)

[Figure 19 biases, learnFcn, LearnParam for traingdx 21](#_Toc90833808)

[Figure 20 Layer Weights function 22](#_Toc90833809)

[Figure 21 Weight & Bias initialization 22](#_Toc90833810)

[Figure 22 Part 1 of script 23](#_Toc90833811)

[Figure 23 Part 2 of script 24](#_Toc90833812)

[Figure 24 net object functions 25](#_Toc90833813)

[Figure 25 Confusion Matrix True/False Positives, False/True Negatives 30](#_Toc90833814)

[Figure 26 ROC (Receiver Operator Characteristic) summarizes all of the confusion matrices that each threshold produced [11] 31](#_Toc90833815)

[Figure 27 ROC makes it easy to identify the best threshold for making a decision 32](#_Toc90833816)

[Figure 28 AUC helps to decide which categorization method is better 32](#_Toc90833817)



Figure 1 Word Cloud

# Abstract

Θα διερευνήσουμε μια μελέτη γύψου από το Wisconsin για δεδομένα καρκίνου του μαστού και θα αναλύσουμε διάφορες μεταβλητές προκειμένου να προσδιορίσουμε εάν ο ιστός του μαστού είναι κακοήθης ή καλοήθης. Μέσω της διαδικασίας ανάλυσης, θα διερευνήσουμε διάφορες τεχνικές, όπως η ταξινόμηση που γίνεται από υπερ-διανυσματικές μηχανές (SVM) και τα νευρωνικά δίκτυα για να προσδιορίσουμε το αποτέλεσμα αυτού του συνόλου δεδομένων ταξινόμησης και ποιες μέθοδοι προσφέρουν τον υψηλότερο βαθμό ακρίβειας στον προσδιορισμό του αρχικού μας δεδομένα εκτελώντας αλγοριθμικές προσεγγίσεις εποπτευόμενης μηχανικής μάθησης όπως αναφέρθηκε προηγουμένως στο MATLAB.

Όπως θα δούμε στην πρώτη εκτέλεση από το Classification Learner που παρέχει και αυτόματη ανάλυση βαθμού βάσης κάθε προσέγγισης αλγορίθμου σε αποδεκτά επίπεδα, το SVM έχει τους καλύτερους αλγόριθμους που καταλαμβάνει την πρώτη θέση για βαθμολογία και συγκεκριμένα το Cubic/Quadratic έχει την υψηλότερη ακρίβεια με 95,2% ενώ στη δεύτερη θέση ακολουθούμενη από τα Νευρωνικά Δίκτυα με την υψηλότερη βαθμολογία 94,4% μιας πρόβλεψης βαθμολογίας με βάση την ταξινόμηση από εκπαιδευμένα δεδομένα για την ταξινόμηση των δεδομένων δοκιμής. Άρα, σαφώς το Τετραγωνικό/Κυβικό SVM με 10 προγνωστικούς παράγοντες είναι οι καλύτεροι αλγόριθμοι για την ταξινόμηση των δεδομένων μας.[[1]](#footnote-1)

Θα βουτήξουμε βαθιά αναλύοντας ολόκληρο τον αγωγό/ροή εργασίας από μη επεξεργασμένα δεδομένα έως το τελικό μοντέλο πρόβλεψης εποπτευόμενης μηχανικής εκμάθησης για έτοιμή παραγωγή και εξαγωγή συμπερασμάτων προβλέψεις και κατηγοριοποιήσεις.

# Machine Learning Algorithms

## Description of input Data Set

Στο coursework τις ανάλυσης δεδομένων για κατάταξη θα χρησιμοποιούν δεδομένα για το καρκίνο ασθενών χωρίζοντας το σε καλοήθη και κακοήθης. Στην επιστήμη δεδομένων και αυτοματοποίηση καθώς υπάρχει κάποιο πάντα λογισμικό με interface (API) που εισάγει τα δεδομένα ώστε να υποστούν μεταγενέστερα κάποια επεξεργασία τείνει να προϋποθέτει την τήρηση κάποιον αυστηρών κριτηρίων ως προς τα ακατέργαστα δεδομένα για να υπάρχει consistency στην ροή και επεξεργασία αυτών των δεδομένων αλλά και αποφυγή σφαλμάτων περί διαφοροποιήσεις τους.

Το Database input μας υπάρχει στο repository URL [kaggle\_wisconsin\_cancer\_data](https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/version/2) τα οποία δεδομένα έρχονται σε μορφή comma separate values (csv) 569 δειγμάτων από ασθενής και 32 στήλες εκ των οποίων οι 30 είναι χαρακτηριστικά ιατρικά δεδομένα του όγκου των ασθενών με numeric scientific notation και ακολουθούν το US/UK decimal and group separation notation (“,” <- thousands, “.” <- decimal full stop/point).

Καθώς πρέπει να υπάρχει ένα καθολικό consistency προ επεξεργασίας αυτών τον δεδομένων γιατί πάντα πρέπει να επικρατεί το σκεπτικό τον μελλοντικών αντίστοιχων εισαγωγών για την διαδικασία αυτοματοποιήσεις τους στα εξής δεδομένα αναλόγως πως θα τα χειριστούμε πρέπει να επιλέξουμε την κατάλληλη κάθαρσή τους. Οι πρώτες 2 στήλες θεωρούνται ξεχωριστές λόγο ως προς το ότι το id των ασθενών δεν το σχετίζουμε άμεσα σε στατιστικές ανάλυσης καθώς θέλουμε μια συνολική εικόνα στην εξαγωγή αποτελεσμάτων αρά στην παρούσα φάση το σβήνουμε η το κρατάμε σε ξεχωριστό αρχείο/μεταβλητή για μεταγενέστερη επεξεργασία.

Το πρόγραμμα μας θα είναι της μορφής script language MATLAB θα διαβάζει επιτόπου το αρχείο δεν θα χρησιμοποιηθεί manual τρόπος ως προς επεξεργασία και μετατροπής csv σε excel η κάποιο έτοιμο import wizard GUI στο MATLAB αρά σε πρώτη φάση διαβάζουμε όλο το αρχείο όπως έχει και διαγράφουμε την πρώτη στήλη κρατώντας την δεύτερη γιατί είναι το response στα predictor μας (είναι το target στα input μας) καθώς όμως το Limitation του script μας στην ανάλυση (natural language process e.g., GPT-3) NLP χαρακτηριστικών και ταξινόμησης γιατί είναι string/text θα το μετατρέψουμε σε αριθμητικά δεδομένα (numeric data) κάτι που μπορεί να «μεταφράσει». Επίσης καλό είναι εξαρχής αλλά μπορεί και μεταγενέστερα να γίνει να κάνουμε το λεγόμενο feature selection δηλαδή από τα Ιατρικά αυτά δεδομένα ποια χρειάζονται να αναλυθούν ώστε να γίνει μια εκτίμηση της ταξινόμησης μελλοντικών αντίστοιχων προβλημάτων μέσα από την εκπαίδευση του neural network δίνοντας μας νέα output σε άγνωστα input, στο παρών πρόβλημα κρατάμε τις 10 πρώτες στήλες από τα λεγόμενα features ή αλλιώς από το original data input unprocessed κρατάμε τις στήλες 3-13 για features (predictors) και την 2 για labeled/target/response (όχι output που πήραμε μετά από εκπαίδευση).



Figure 2 read input dataset self-documenting code in MATLAB 😊

## Problem Solving & Data Analysis

Η επίλυση ενός προβλήματος προϋποθέτει πρώτα την αξιολόγηση ανάλυση σε υψηλό abstract επίπεδο της ιδιαιτερότητες του ώστε να γίνει σωστή προσέγγιση από την επιλογή της πιο efficient ή αποδέκτης λύσης βάση πάντα ανάλογα το κόστος-χρόνος. Δύο πιο συνήθεις μορφές για «applying supervised machine learning» είναι οι αλγόριθμοι/τεχνικές το Super vector machine & Neural Networks οπού ανήκουν στην μεγάλη κατηγορία μοντέλων: regression (when the response is continuous) & classification (when the response belongs to a set of classes) και εφαρμόζονται σε δεδομένα όπου αυτοί οι αλγόριθμοι δημιουργούν μοντέλα που καταγράφουν τις τάσεις στα σύνολα δεδομένων: [machine-learning-models](https://www.mathworks.com/discovery/machine-learning-models.html)

Παρακάτω αναλύουμε αυτές τις 2 κατηγορίες machine learning μοντέλων Που είναι προσιτά σε προβλήματα όπου δημιουργείται ένα μοντέλο που προβλέπει μια ανταπόκριση/απάντηση σε αυτά (predict a response).

### Super Vector Machine (SVM) - fitcsvm

Ένα SVM είναι μια μηχανή μάθησης (που εκπαιδεύεται) που ταξινομεί τα δεδομένα βρίσκοντας το γραμμικό όριο απόφασης (hyperplane) που διαχωρίζει όλα τα σημεία δεδομένων μιας κλάσης από εκείνα της άλλης κλάσης σε 2D είτε 3D δημιουργώντας κύκλο άμα δεν μπορούν να διαχωριστούν σε δισδιάστατο επίπεδο (παρακάτω άμεσα για αυτό). Το καλύτερο υπερεπίπεδο για ένα SVM είναι αυτό με το μεγαλύτερο περιθώριο μεταξύ των δύο κλάσεων (margin), όταν τα δεδομένα είναι γραμμικά διαχωρίσιμα. Εάν τα δεδομένα δεν διαχωρίζονται γραμμικά, χρησιμοποιείται μια συνάρτηση απώλειας για την τιμωρία σημείων στη λάθος πλευρά του υπερεπιπέδου. Τα SVM μερικές φορές χρησιμοποιούν έναν μετασχηματισμό πυρήνα για να μετατρέψουν μη γραμμικά διαχωρίσιμα δεδομένα σε υψηλότερες διαστάσεις όπου μπορεί να βρεθεί ένα γραμμικό όριο απόφασης. Υποστηρίζει out-of-the-box binary classification και με κάποια τροποποίηση η μέσο του binary classification Προσεγγίζεις το multiclass-classification όταν εχει πάνω από 2 διαφορετικά δεδομένα. Χρησιμοποιείται για image/face/text recognition and spam filter et al.

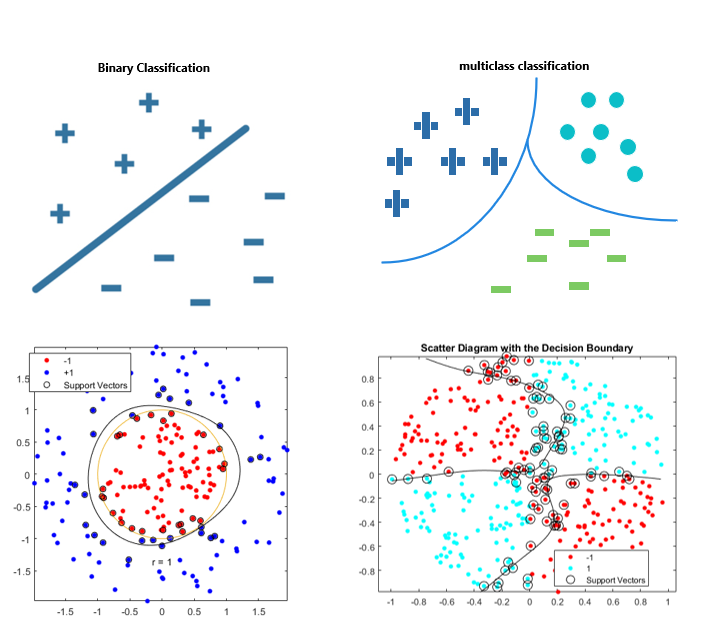


Figure 3 SVM algorithm type models

#### Code









#### Figures - Plots

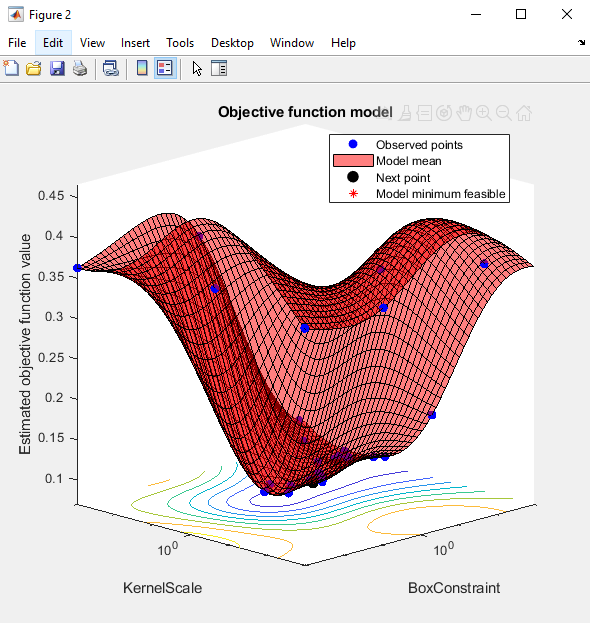
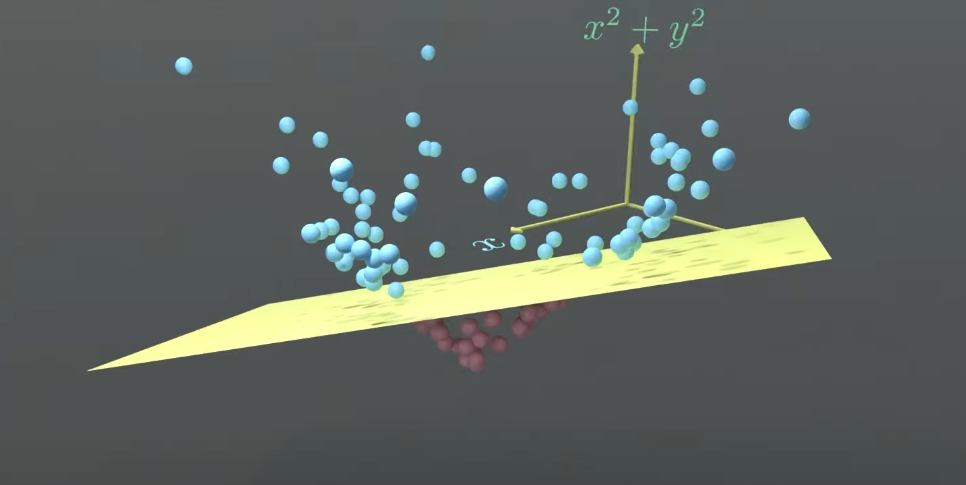
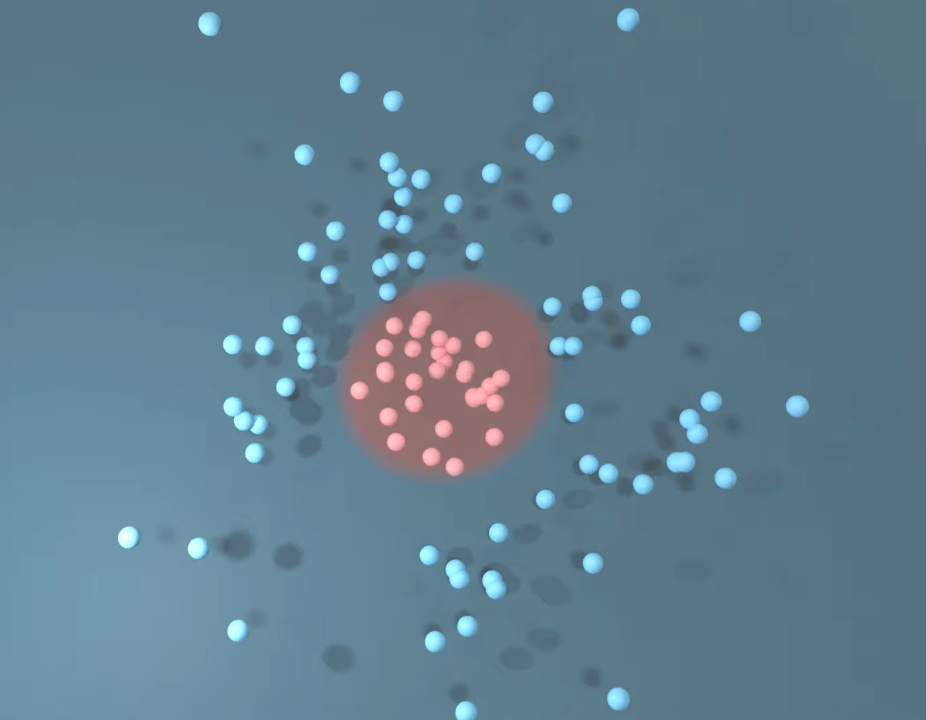
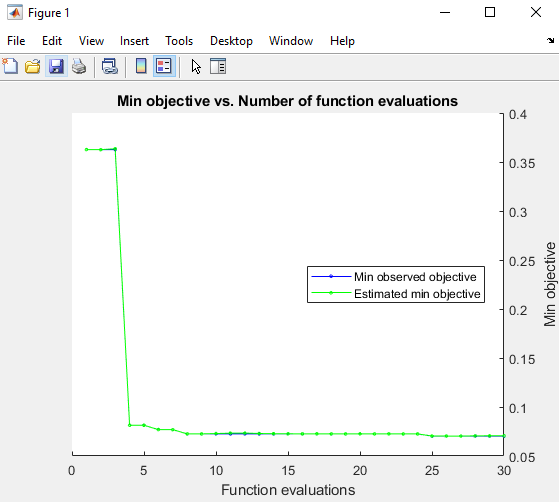


Figure 4 Hyperplane for 3D non-linear separation by scaling (before apply the data to kernel function e.g., cubic or linear) and penalty the data







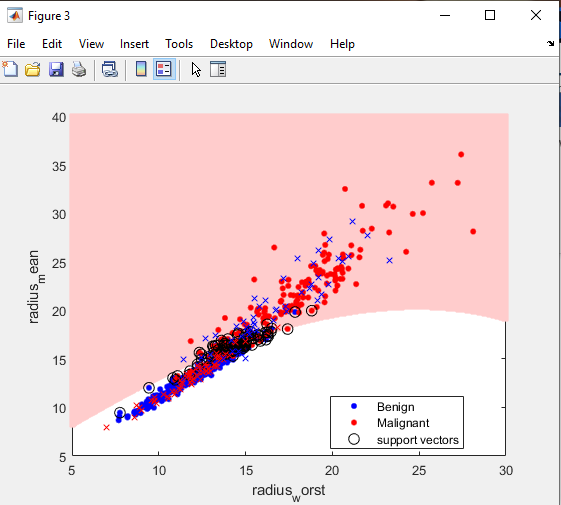


Figure 5 Scatter Diagram of B-M Types

#### Comparison SVM vs NN

Control + scroll mouse in for zoom-in or save image as



Figure 6 SVM vs Neural Network on various train time, Accuracy (validation), Total Cost, prediction speed, Model Type info

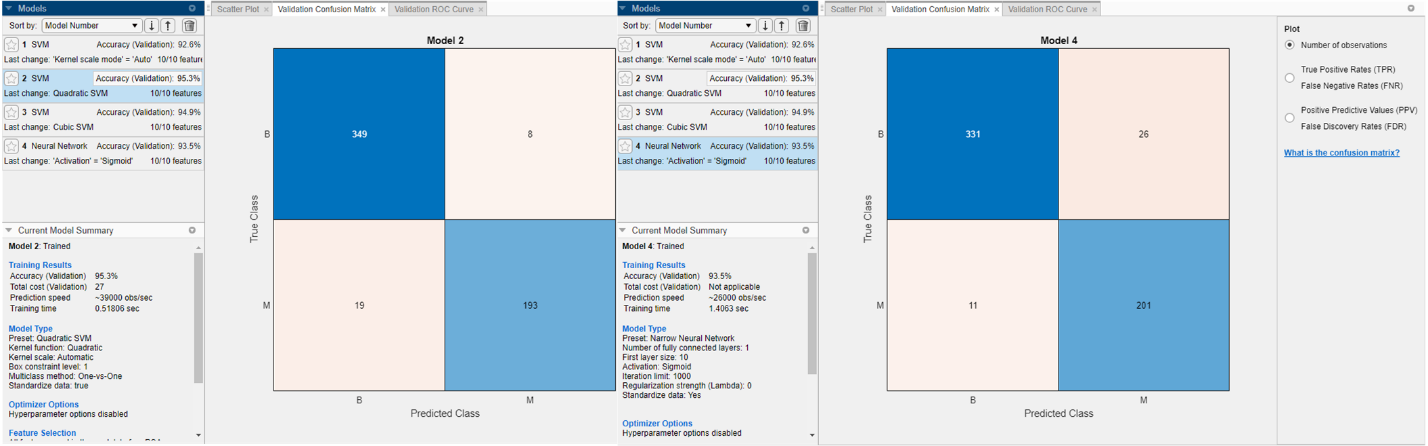


Figure 7 SVM vs NN confusion Matrix



Figure 8 SVM vs NN ROC compare shows Top Right Best Results of summary overall confusion matrix for decision making on point (Quadric SVM)

Quadric SVM as Average is the best compared to NN for this classification problem

### Artificial Layered Neural Network (AΝΝ)

Εμπνευσμένο από τον ανθρώπινο εγκέφαλο, ένα νευρωνικό δίκτυο αποτελείται από πολύ συνδεδεμένα δίκτυα νευρώνων (με πιθανά κρυμμένα ενδιάμεσα μεταξύ input layer + output layer) που συνδέουν τις εισόδους με τις επιθυμητές εξόδους. Το δίκτυο εκπαιδεύεται τροποποιώντας επαναληπτικά (η επανάληψη είναι μήτηρ της μαθήσεις) τα δυνατά σημεία των συνδέσεων, έτσι ώστε οι εισροές εκπαίδευσης να αντιστοιχίζονται στις αποκρίσεις εκπαίδευσης. Με αποτέλεσμα την σωστή κατανόμηση νέων αγνώστων δεδομένων από νέα inputs με βάση το εκπαιδευμένο μοντέλο δικτύου.

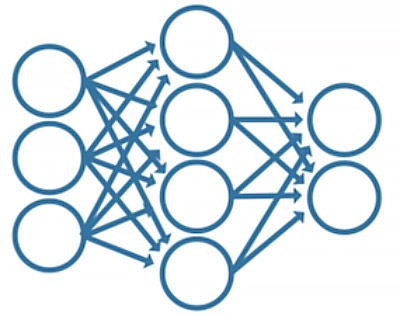


Figure 9 Layered NN

Τα workflow/pipeline βημάτων σχεδιάσεις ενός νευρωνικού δικτύου είναι 4 (μπορούμε να το επαληθεύσουμε από το “*nnstart*” || “*classificationLearner*” steps)

1. Gathering Data
2. prepare data in proper format (predictors & responses) by feature selecting and keeping same matrix dimensions[[2]](#footnote-2)
3. Categorize Data (Train, Validation and Test)[[3]](#footnote-3)
4. Choosing a ML model
   1. Network Architecture by assigning Hidden layers and Neuros per layer
5. Evaluation of Train Network based on input model and visualize or text read the outputs[[4]](#footnote-4)
   1. With/without validation data
   2. Feature selection
6. Finding best hyperparameters by tuning
7. Step 4
8. Re-train and test the model with test set[[5]](#footnote-5)
9. Prediction & visualize final reports in plots and hyperplanes

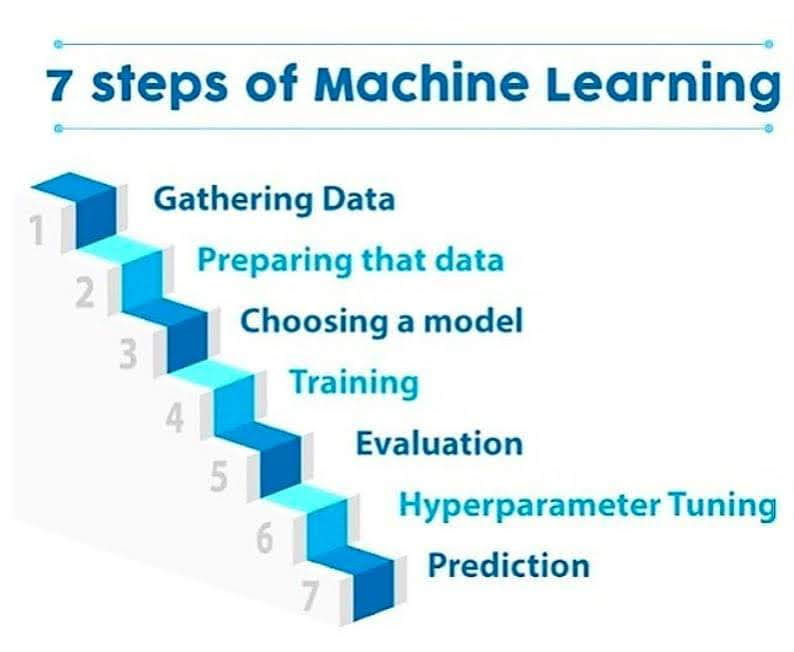


Figure 10 steps of Machine Learning (ML)

#### Training-Test-Validation

Ίσως το ποιο σημαντικό κομμάτι πριν την αρχικοποίησή παραμέτρων νευρωνικού δικτύου διότι θα μας δώσει Insights πως εχει σχεδιαστεί το δίκτυο αλλά και άμα τα data set είναι αρκετά για να προκύψει σωστά η διεκπεραιώσει της εκπαίδευσης ώστε να αξιολογηθούν τα αποτελέσματα και να συγκριθούν με αυτά testing[[6]](#footnote-6).

Η διαδικασία ξεκινά επιλέγοντας τυχαία (για τις 3 κατηγορίες random selection) ένα σύνολο από τα συνολικά δεδομένα χωρίζοντας τα (στο παράδειγμά μας 569 samples) δηλαδή εφαρμόζοντας πάνω σε αυτά ένα σύνολο μαθηματικών συναρτήσεων/αλγόριθμων για τις 3 κατηγορίες των δεδομένων και ξεκινώντας την εκπαίδευση από τυχαία σειρά εκκίνησης στο καρτεσιανό επίπεδο μέσο τον εκάστοτε αλγορίθμων υλοποιήσεις του δικτύου λόγο του λογισμικού/framework χρήσης[[7]](#footnote-7):

##### Train

Τα δεδομένα(τυχαίο υποσύνολο των συνολικών) στα οποία θα «προπονηθεί» το δίκτυο να παράγει αποτελέσματα. Δηλαδή με βάση τα predictors/inputs στα responses/targets πόσο ποσοστού λάθους υπάρχει και εχει καταφέρει να μειώσει ο αλγόριθμος αλλάζοντας κάθε πλήρη επανάληψη των sample για per epoch τα weights και bias (για προσαρμογή ως προς μείωση σφάλματος) ώστε να θεωρηθεί «προπονημένο» να παράγει με νέα δεδομένα σωστή κατηγοριοποίηση με βάση αυτά που εχει ήδη μάθει ξεχωρίζοντας μοτίβα μέσο της επανάληψης της εισόδου των ίδιων δεδομένων ξανά και ξανά και συγκρίνοντας τα με τα targets τι θα πρέπει να πάρει δηλαδή.

Ένα παράδειγμα σωστού εκπαιδευμένου δικτύου είναι να κοιτάξουμε το διάγραμμα/γραφική παράσταση (λεγομένη καμπύλη της εκπαίδευσης, τα σφάλματα (y logarithmic axes) ως προς τον αριθμό επαναλήψεων (x integer numeric axes)) mean square error (mse[[8]](#footnote-8)) για τα train data (μπλε γραμμή πρέπει να μικρύνει Figure 6) με μαθηματική προσέγγιση: mse = (target– output)^2 ανά sample output ως προς τα ποσά epochs (iterations) έχουν τρέξει τα δεδομένα για εκπαίδευση διαρρέοντας το. Στο παράδειγμα μας ανά 569 (προσθέτοντας όλα τα mse σε ένα total\_mse) υπολογισμούς mse διαιρείς με πόσες φορές έτρεξε το νευρωνικό δίκτυο εισάγοντάς ξανά τα ιδιά δεδομένα και συγκρίνοντας με τα output δηλαδή πχ sum\_mse\_of\_input / epochs (e.g., ).

##### Test

Ένα τυχαίο υποσύνολο των συνολικών δεδομένων που χρησιμοποιείται για έλεγχο ποσό καλή είναι η εκπαίδευση του νευρωνικού δικτύου με βάση τα train data set. Συνήθως είναι η πιο κρίσιμη καμπύλη διότι εφαρμόζει αυτή τη σύγκριση/πρόβλεψη με νέα δεδομένα (που δεν εχει δει στην εκπαίδευση) ώστε να αποβεί χρήσιμο το δίκτυο για classification.

Συμβολίζεται με κόκκινη γραμμή (εξ ου και το χρώμα Figure 6) και θα πρέπει να είναι όσο πιο δυνατόν παράλληλη με την γραμμή των train data άμα τυχόν αυξηθεί υπάρχει πιθανότητα δυνατού νευρωνικού δικτύου δηλαδή overfitting χάνοντας το concept του generalization και αυτό εχει επίπτωσή μόνο στα νέα data όχι στα trained ή υψηλής (y) στασιμότητας (local minimum) (χωρίς validation data αυτό γίνεται more pronounced κατά την εκπαίδευση).

1. Δεν επηρεάζει την κρίση του δικτύου ως προς την διασύνδεση (δεν χρησιμοποιεί weight & bias αλλαγές per epoch) των νευρώνων (απλά πυροδότη τους νευρώνες όταν υποστεί κάποιο πέρασμα πάνω από threshold με βάση τα κριτήρια) και νέα εκμάθηση[[9]](#footnote-9) απλά αξιόλογη τα δεδομένα με βάση ως τι έμαθε από τα train dataset με βάση τη μείωση σφάλματος (*y = (t – x)^2* ) ως προς τη προσεγγίσει στο μηδέν.

##### Validation

Υποσύνολο στοχαστικών δεδομένων (πράσινη γραμμή Figure 6) των συνολικών δεδομένων για επαλήθευση με βάση των σφαλμάτων. Δηλαδή ένα τρίτο κομμάτι των δεδομένων κατά ποσό εχει νόημα να συνεχίσω την εκπαίδευση λόγο της στασιμότητας (local minimum) αλλαγής η αύξηση της καμπύλης (overfit) εκπαίδευσης της επικύρωσης μετά από ένα συγκεκριμένο αριθμό σε επανάληψης (epochs) σε τοπικό ελάχιστο η κάποιας υπερεκπαιδευσης overfitting (prevents by identification) συνήθως χαρακτηρίζεται με άμεση αύξηση μαζί με την κόκκινη γραμμή test data (overfit).

Στην αρχή της «χτίσεις» του νευρωνικού δικτύου συνήθως δεν βάζουμε στο διαχωρισμό data και αυτήν για έλεγχο επειδή μας εμποδίζει να δούμε για τυχόν core σφάλματα στην επιλογή τεχνικών/αλγορίθμων των παραμέτρων για την εκπαίδευση του δικτύου. 1 παραπάνω

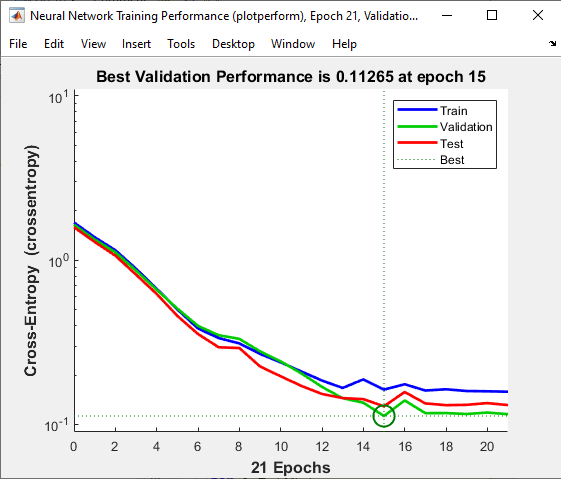


Figure 11 training curve

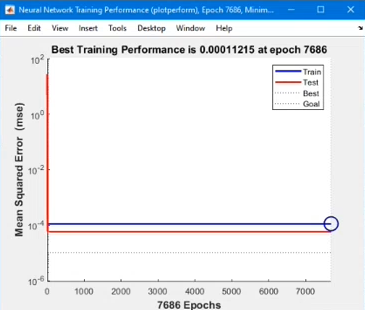


Figure 12 Local minimum stuck (both lines must) but close to solution function by minimizing the total expected (from data) error

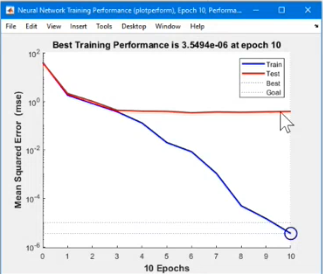


Figure 13 Example of strong neural network failing (overfitting) on new data classification (there is not a local minimum error cause of train data success)

#### Neural Network Architecture Parameters

Τα πιο σημαντικά κριτήρια αρχιτεκτονικής σχεδιάσεις δικτύου σαν ορόσημο είναι τα κρυφά επίπεδα που θα αποτελείται το νευρωνικό δίκτυο αλλά και πόσους νευρώνες αλληλένδετους ανά layer. Θα εχει κατανοηθεί η βαθύτητα του προβλήματος ώστε να χρησιμοποιηθεί μία σωστή διάταξη για την επίλυση του. Δεν υπάρχει κάποιο ιδανικό πάρα μόνο την εκτίμηση του από τον σχεδιαστή του νευρωνικού δικτύου. Για απλά προβλήματα χρειάζονται απλά νευρωνικά δίκτυα με μικρό αριθμό νευρώνων και κρυφών επιπέδων, πιο σύνθετα προβλήματα εκτιμάται η αύξηση των πόρων σε νευρώνες και hidden layers. Όπως προαναφέρθηκε η χρήση λανθασμένων παραμέτρων για τις συζεύξεις των επιπέδων και νευρώνων οδηγεί στην αποτυχία σωστής εκπαίδευσης η ανταπόκρισης σε παρουσία νέων δεδομένων για αξιολόγηση.

Στο συγκεκριμένο πρόβλημά μας εχει χρησιμοποιηθεί ύστερα από παρατήρηση αποτελεσμάτων μετά από training 1 hidden layer με 10 νευρώνες με sigmoid transfer function[[10]](#footnote-10) (το επαληθεύουμε με εντολή *net.layers{1}.transferFcn* Figure 10)[[11]](#footnote-11) [1] (καθώς και 5 θα αρκούσαν οριακά για το συγκεκριμένο παράδειγμα) με 1500 epochs χρησιμοποιώντας μια επαναληπτική on-the-fly πληθώρα επιλογή από συναρτήσεων εκπαίδευσης (trainscg, trainrp, traingdx[[12]](#footnote-12)) Figure 10 Figure 12.

Οι στόχοι είναι η επανάληψη της εκπαίδευσης (re-train) με feed ίδια data λόγο (όπως προαναφέρθηκε) την απόκριση νέων αποτελεσμάτων εξαιτίας της διαφορετικής αρχικοποιήσεις συνθηκών και παραδειγμάτων στο καρτεσιανό επίπεδο αξόνων. Όσο μικρότερο σφάλμα ανά επανάληψη (epoch) σημαδοτή στην καλύτερη κατάταξη των δυαδικών η μη-δεδομένων. Για να επιτευχθεί αυτό αναπροσαρμόζονται στο βήμα της εκπαίδευσης data train (μπλε γραμμής) τα weights & bias[[13]](#footnote-13) ώστε να έρθουν πιο κοντά στο προσιτό αποτέλεσμα και προσδοκώμενο εξ αρχής γνωρίζοντας το (supervised ML).

##### Summary

1. Να αιτιολογηθούν οι επιλογές που θα κάνετε σχετικά με:
   1. Τον χωρισμό των δεδομένων σε training-test-validation.
      1. Πετυχαίνει την σωστή εκπαίδευση και νέα πρόβλεψη/διαχωρισμό του νευρωνικού δικτύου αποφεύγοντας σφάλματα και τοπικά ελάχιστα δημιουργώντας ευαισθησία στις συνάψεις των νευρώνων αναγνωρίζοντας μοτίβα. (Supervised ML)
   2. Τις αρχικές συνθήκες (αρχική επιλογή βαρών και biases).
      1. Χρησιμοποιούνται για την διεκπεραίωση μειώσεις του σφάλματος (loss function) ανά epoch υπολογίζοντας καλύτερη άφιξη στην λύση μέσο παρακολουθήσεις της μειώσεις της συνάρτησης σφάλματος και υφίστανται ανά neuron στη σύναψη (input). Γίνεται adjust κάθε iteration (subset of epoch) μέσο του gradients of the loss landscape δηλαδή η τεχνική backpropagation. Το gradient της ενημέρωσης για ένα συγκεκριμένο σφάλμα υπολογίζεται σε σχέση με το αρχικό σφάλμα και τους νευρώνες μεταξύ του συγκεκριμένου νευρώνα και του σφάλματος. Όταν οι ενημερώσεις βάρους είναι μεγάλες υπάρχουν αρνητικά σενάρια. Αυτό συμβαίνει επειδή όταν τα βάρη αιωρούνται μπρος και πίσω, είναι πιθανό ότι είτε είχαμε μια πολύ ταλαντευόμενη πορεία προς το παγκόσμιο ελάχιστο μας. Επιπλέον, όταν είναι μεγάλα, μπορεί να είναι ότι συνεχώς παρακάμπτετε το βέλτιστο, λαμβάνοντας χειρότερη απόδοση από ό, τι είναι απαραίτητο. [2]
      2. *new\_weight = old\_weight - learning rate \* gradient update*
   3. Τον αριθμό των layers (στρωμάτων) του νευρωνικού δικτύου.
      1. Πετυχαίνουν την ταχύτητα εκπαίδευσης του δικτύου μέσο της γενίκευσης του τύπου κατηγορίας προβλημάτων εφαρμόζοντας non-linear transformations. Με hidden layers ένα perceptron μετατρέπεται σε ένα βαθύ δίκτυο αποθηκεύσεις και επεξεργασίας δεδομένων. Κάθε layer χρησιμεύει για την αναγνωρίσει ενός αντικειμένου e.g., άμα έχουμε ένα αυτοκίνητο σε οπτική εικόνα ένα layer Θα είναι για άμα υπάρχει τζάμι ένα άλλο για άμα υπάρχουν φώτα ένα άλλο άμα υπάρχουν ρόδες et al. [3]
   4. Τον αριθμό των νευρώνων σε κάθε layer.
      1. Καθορίζουν την ταχύτητα εκπαίδευσης και άφιξης σε ένα ελάχιστο σφάλμα καθώς και γενίκευση. Κάθε νευρώνας εχει το δικό του transfer function output δηλαδή το κατώφλι πυροδοτήσεις από τα inputs και μεταφοράς αυτής της πληροφορίας σε ένα mesh network. Συνήθως χωρίς hidden layer ονομάζονται shallow. Ένα layer δεν μπορεί να επεξεργαστεί μεγάλη πληροφορία πολλών διαστάσεων και λύνει 1 πρόβλημα μόνο για το οποίο κατασκευάστηκε [4]
   5. Το είδος των συναρτήσεων μεταφοράς.
      1. Στη μηχανική μάθηση, τα αθροίσματα κάθε κόμβου σταθμίζονται και το άθροισμα διέρχεται μέσω μιας μη γραμμικής συνάρτησης γνωστής ως συνάρτηση ενεργοποίησης ή συνάρτησης μεταφοράς.
      2. Στη μηχανική εκμάθηση, η συνάρτηση ενεργοποίησης χρησιμοποιείται πιο συχνά, ενώ νομίζω ότι η "συνάρτηση μεταφοράς" χρησιμοποιείται πιο συχνά στην επεξεργασία σήματος. Επομένως, όποιος τους χρησιμοποιεί ως δύο διαφορετικούς όρους θα πρέπει να είναι πιο σαφής.
      3. Μια τιμή (ισχύς σήματος) για να επαληθεύσετε εάν ο νευρώνας θα ενεργοποιηθεί και στη συνέχεια να υπολογίσετε μια έξοδο από αυτόν. Έτσι η όλη διαδικασία μπορεί να μεταφέρει ένα σήμα από το ένα στρώμα στο άλλο.
      4. *transfer\_function = activation function + output function* [5]
   6. Το είδος της εκπαίδευσης (αλγόριθμος εκμάθησης) που χρησιμοποιήσατε.
      1. Υπάρχουν αρκετή αλγόριθμοι για εκπαίδευση του νευρωνικού δικτύου στο παράδειγμα μας χρεισημοποιήσαμε τον:
         1. trainscg [6] (scaled conjugate gradient backpropagation)
         2. trainrp [7] (Resilient backpropagation)
         3. Traingdx [8] (Gradient descent with momentum and adaptive learning rate backpropagation)
      2. Οι παραπάνω ανήκουν σε πληθώρα ειδών μεταξύ gradient descent και conjugate gradient οι οποίοι είναι αρκετά efficient άμα υπάρχουν πολλές παράμετροι σώζοντας μνήμη/πόρους (υπολογιστικούς) αλλά είναι αργή, έχει να κάνει με το πώς προσεγγίζει την συνάρτηση απώλειας με μαθηματικό μοντέλο (numerical precision) και την ταχύτητα επεξεργασίας et al [9].

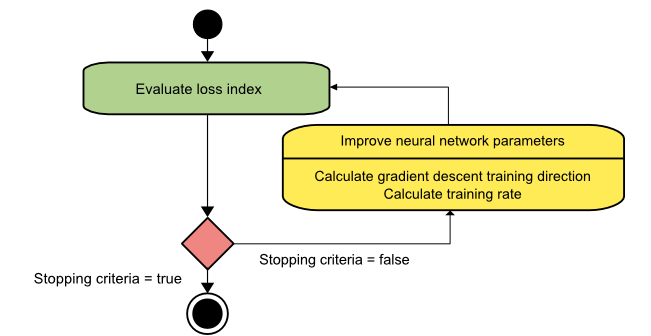


Figure 14 Loss function for gradient descent

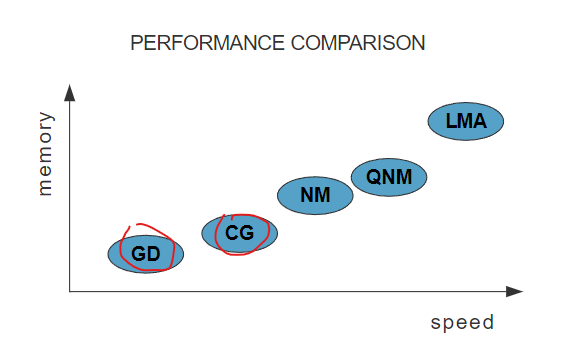


Figure 15 [GD (gradient descent) CG (conjugate gradient) are slowest in speed but good in memory consumption](https://www.neuraldesigner.com/blog/5_algorithms_to_train_a_neural_network#ConjugateGradient)

* 1. Τον αριθμό των epochs.
     1. Αριθμός χρήσιμος για την προσπάθεια στη μείωση του σφάλματος. Ένας πλήρης κύκλος εκπαίδευσης από όλο το dataset. Κάθε epoch κάνουμε feed το δίκτυο με διαφορετικά patterns και αρχίζοντας από διαφορετικό σημείο εκκίνησης βοηθώντας στο generalization. Ευρετικά, ένα κίνητρο είναι ότι (ειδικά για μεγάλα αλλά πεπερασμένα σύνολα κατάρτισης) δίνει στο δίκτυο την ευκαιρία να δει τα προηγούμενα δεδομένα για να αναπροσαρμόσει τις παραμέτρους του μοντέλου έτσι ώστε το μοντέλο να μην είναι προκατειλημμένο (biased) προς τα τελευταία σημεία δεδομένων κατά τη διάρκεια της εκπαίδευσης. Καθώς δεν υπάρχει το ιδανικό σαν νούμερο ποσά epochs θα μας δώσουν καλό αποτέλεσμα θέλει trial and error (iteration != epoch but a subset). [10]
  2. Την τιμή του learning rate και του momentum.
     1. Χρησιμοποιείται για την ελαχιστοποίηση την συνάρτησης σφάλματος αλλά και αποφυγή τοπικού ελάχιστου. Το ορίζουμε πριν την έναρξη εκπαίδευσης και κάνει μικρότερο gradient update. θα μας πάρει περισσότερο χρόνο για να συγκλίνουν (converge), αλλά πιθανότατα δεν έχει υπερχείλιση και ταλάντωση λιγότερο σοβαρά σε epochs. Έτσι βοηθά έμμεσα και τα weights δες 1.e.iv [2]

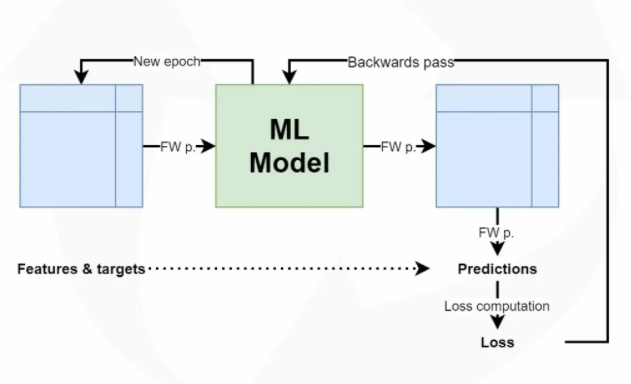


Figure 16 high-level machine learning process for supervised learning scenarios

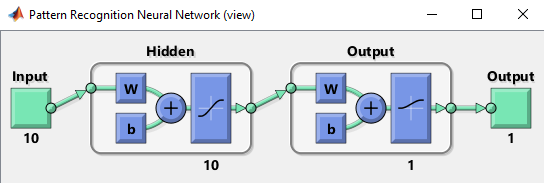


Figure 17 view(net)

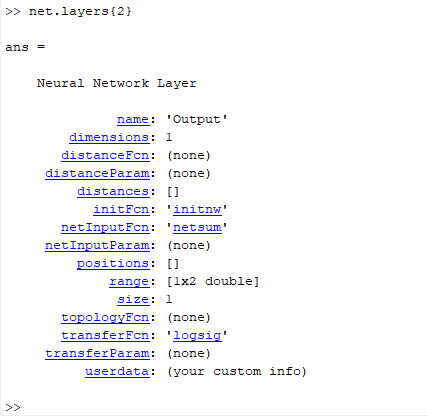


Figure 18 Layer transfer function

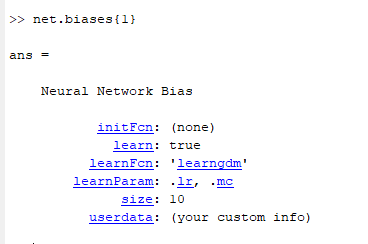


Figure 19 biases, learnFcn, LearnParam for traingdx

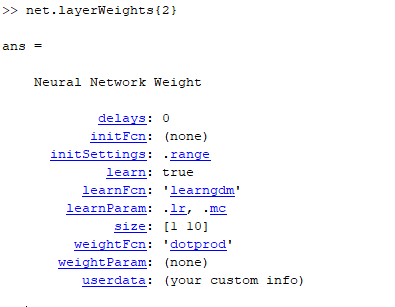


Figure 20 Layer Weights function

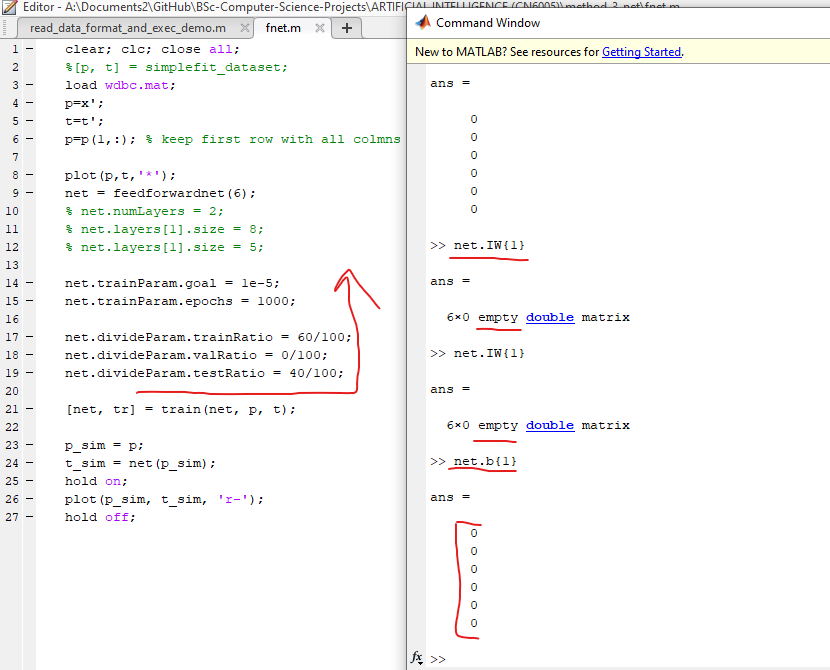


Figure 21 Weight & Bias initialization



Figure 22 Part 1 of script

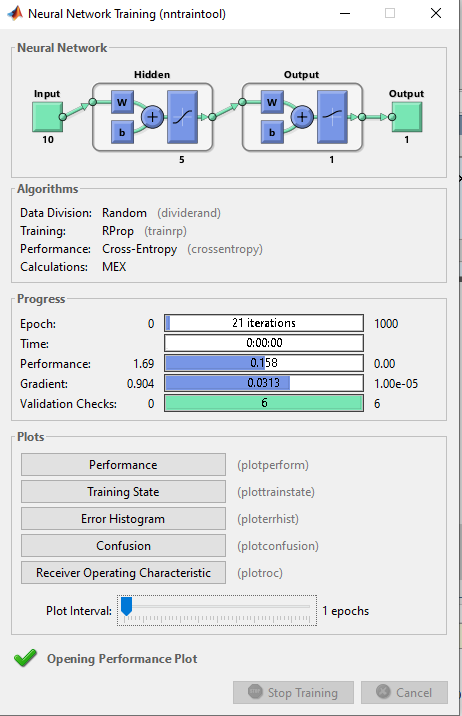


Figure 23 Part 2 of script



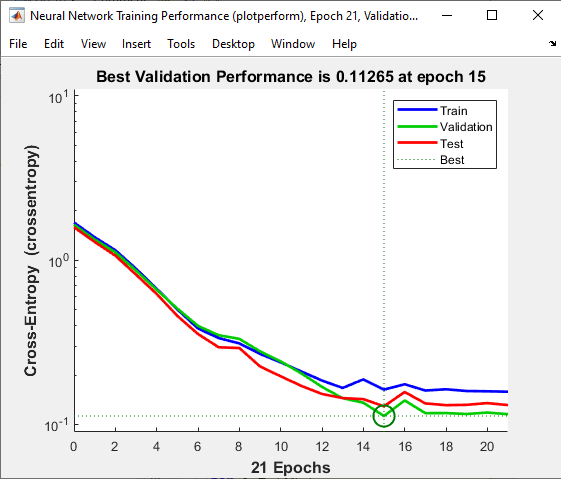
Figure 24 net object functions

#### Plot figures diagrams



Στο παραπάνω figure βλέπουμε ένα δίκτυο με ένα hidden layer από 5 νευρώνες με συνάρτηση μεταφοράς σιγμοειδής και 10 feature/samples/rows input με 1 output (1 or 0, Malignant or Benign) με χωρισμό των δεδομένων τυχαία σε 3 κατηγορίες (train-test-valid) χρήση dividerand.

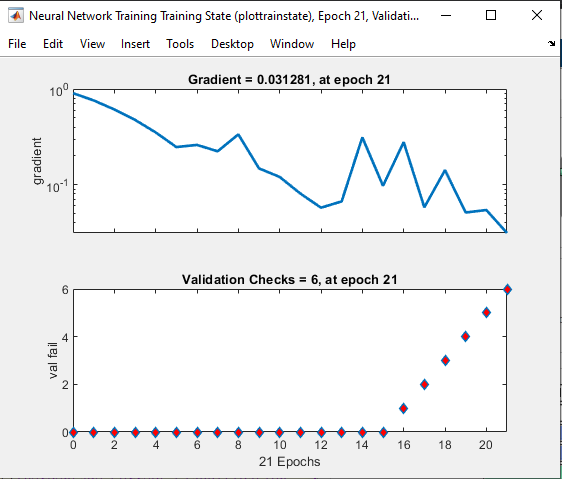
Έχει χρησιμοποιηθεί ο αλγόριθμος trainrp για εκπαίδευση και function loss το cross-entropy (αντί του mse σε αυτό το παράδειγμα λόγο των πολλαπλώς τεστ). Χρειάστηκαν 21 επανάληψης μέχρι το validation check από 6 tries να σταματήσει την εκπαίδευση.



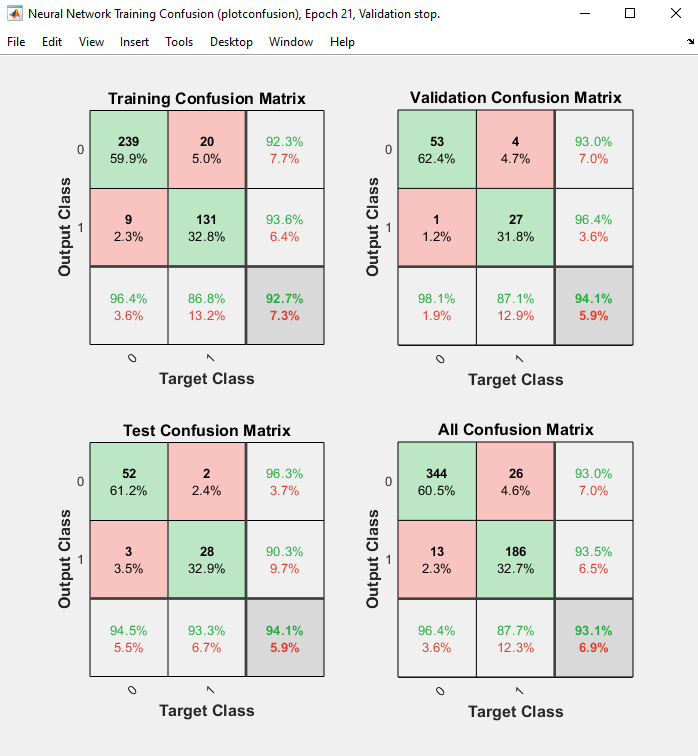
Στο παραπάνω Plot βλέπουμε μια καλή πορεία των 3 κατηγοριών δεδομένων εκπαίδευσης και επαλήθευσης νευρωνικού δικτύου. Γενικώς θα ήταν καλό και οι 3 αυτές καμπύλες να πηγαίνουν κολλητά με χωρίς απότομες αυξομειώσεις αλλά ομαλά (αρμονικά) όσο πιο κοντά γίνεται στο μικρότερο ποσοστού σφάλματος.

Επίσης συμπεραίνουμε ότι το validation υστέρα από 6 προσπάθειές εφόσον βρήκε το ελάχιστο αναμεσά στο 14-16 σταμάτησε την εκπαίδευση (best validation performance) του δικτύου καθώς οι τιμές κολλήσαν σε κάποιο τοπικό ελάχιστο όμως αυτό λόγο του μικρού σφάλματος είναι σχεδόν και το καθολικό ελάχιστο.

Δεν παρατηρούμαι φαινόμενα overfitting (κρατώντας σε θετικό επίπεδο το concept της γενίκευσης) διότι η κόκκινη γραμμή και πράσινη θα έπρεπε να είχαν απότομη ξαφνική ανοδική πορεία.



Βλέπουμε το best validation performance αναμεσά στο 14-16 να προσπαθεί 6 φορές να κρατήσει την εκπαίδευση σε λειτουργεία λόγο του στοχασμού αλλά στη συγκεκριμένη περίπτωση έχουμε το λεγόμενο convergence (σύγκλιση) δηλαδή δεν θα βελτιωθεί περεταίρω το δίκτυο αρά παύει να εκπαιδεύεται.



Το πιο σημαντικό διάγραμμα (plot) ιδίως το κομμάτι του Test confusion matrix.

Μας δείχνει ότι εκ των 2 κατηγοριών (1-malignant, 0-benign) στη κατηγορία 0 από τα 54 data για test πέτυχε σωστά τα 52 και τα 2 τα κατάταξε στην κατηγορία 1 αντί την 0.

Παρόμοια στην κατηγορία 1 3 από αυτά τα κατάταξε στην κατηγορία 0 που είναι λάθος και 28 στην 1 με πιο δεξιά να βλέπουμε ποσό ποσοστό τις % πέτυχε 90.3%

Η Σημαντικότερη τιμή είναι το συνολικό αποτέλεσμα (κυρίως του test data) που μας δίνει βαθμό 94.1% επιτυχίας με 5.9% αποτυχίας.

Συμπέρασμα? Τα πήγε καλά ! καθώς σε κάθε επανάληψη re-train όπως προαναφέρθηκε οι τιμές αυτές αλλάζουν είτε παραπάνω είτε παρακάτω με μεγαλύτερο σκορ από ότι παρατηρήθηκε το 97.2% στο test data και ελάχιστο 89% Αυτό διότι κάθε φορά γίνεται τυχαία η εκκίνηση για λύση του προβλήματος και τα πολλαπλά re-train μα δίνουν το αποτέλεσμα για ένα υψηλότερο σκορ (στην εικόνα είχε τρέξει 1 φορά η εκπαίδευση για χ αριθμό epoch) και να αναπροσαρμόσουν το σύστημα έτσι ώστε να μην είναι πολωμένο/ προκατειλημμένο/biased συγκρίνοντας τα αποτελέσματα με τα προηγούμενα χωρίς να μείνει στάσιμο στα πιο πρόσφατα αποτελέσματα.



Figure 25 [Confusion Matrix True/False Positives, False/True Negatives](https://www.youtube.com/watch?v=4jRBRDbJemM)

True/False Positive = Όταν είναι αληθές (σωστό) στην κατηγορία positive (1) και false κατηγοριοποίηση στη λάθος αλλά από τα positive πάλι.

False/True Negative = Όταν είναι αληθές (σωστό) στην κατηγορία negative (0) και false κατηγοριοποίηση στη λάθος αλλά από τα negatives πάλι.

Δηλαδή μας ενδιαφέρει το χιαστί (διαγώνιος) πόσα TP & TN πέτυχε σωστά τα υπόλοιπα πάνε στα FP & FN τα οποία είναι άκυρα.

Αυτό εχει πολύ απήχηση και όχι μόνο και στα logistic regression όπου έχουμε πολλά confusion matrix άμα πάρουμε πολλά high-low thresholds για να μην συμβεί αυτό ερχόμαστε στο ROC & AUC (Receiver Operator Characteristic & the Area Under the Curve).

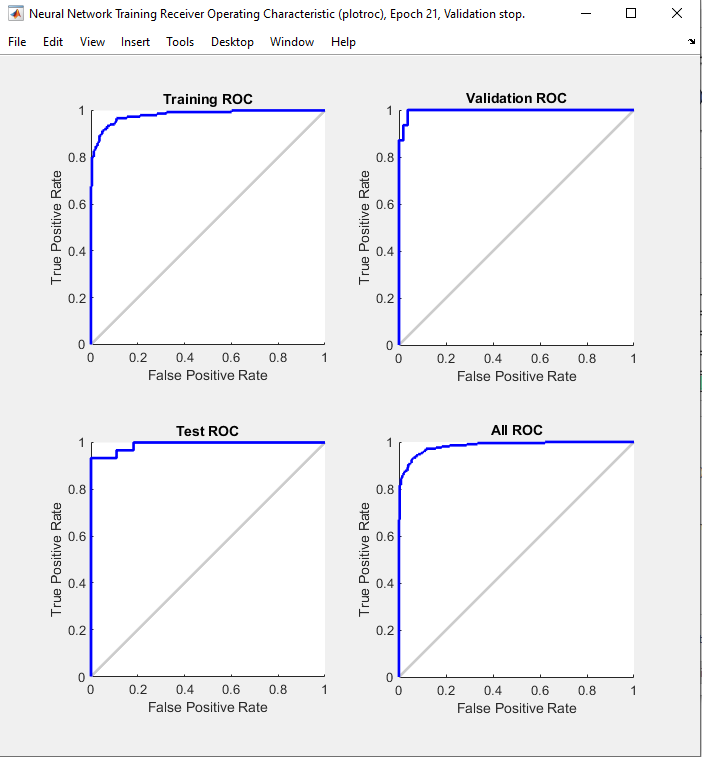


Figure 26 ROC (Receiver Operator Characteristic) summarizes all of the confusion matrices that each threshold produced [11]

H Καμπύλη διαβάζεται από δεξιά προς τα αριστερά δείχνοντας όσο πιο αριστερά υπάρχουν σημεία τόσο πιο σωστά αποτελέσματα έφερε σε εκείνο το threshold (παίρνοντας το 1,1 σημείο στο επίπεδο σαν ίση κατηγοριοποίηση σωστών και λανθασμένων δηλαδή 10 λάθος 10 σωστά και μικραίνοντας τα λάθος όσο πάει αριστερά)

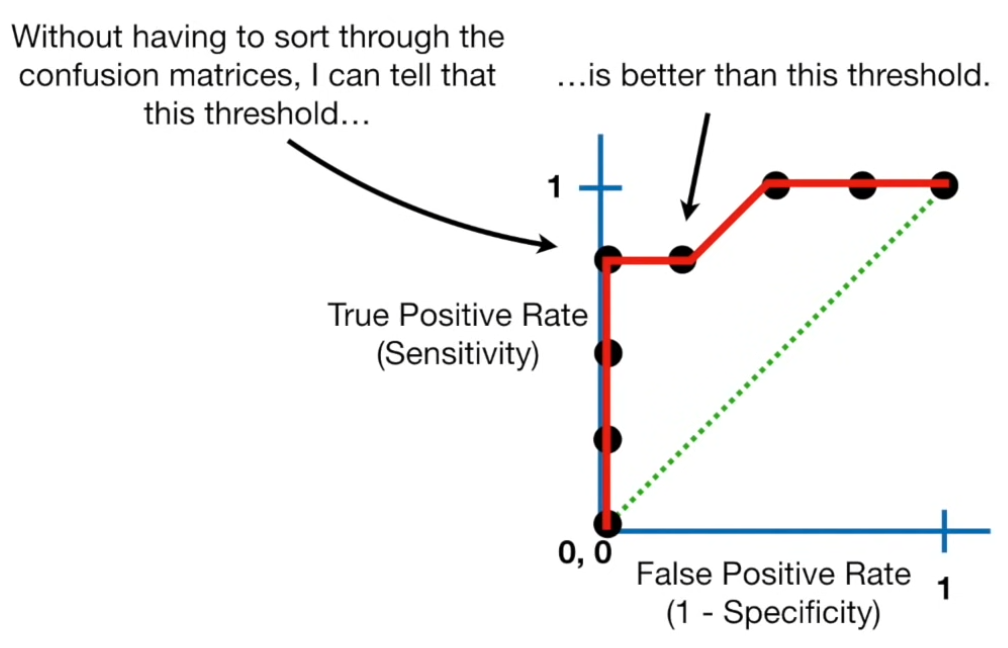


Figure 27 ROC makes it easy to identify the best threshold for making a decision

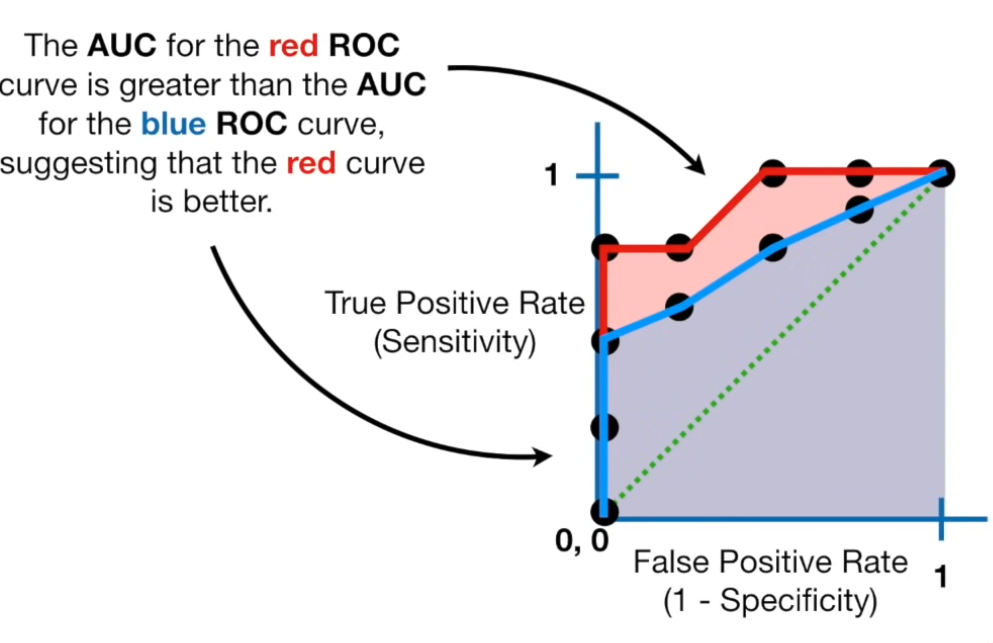
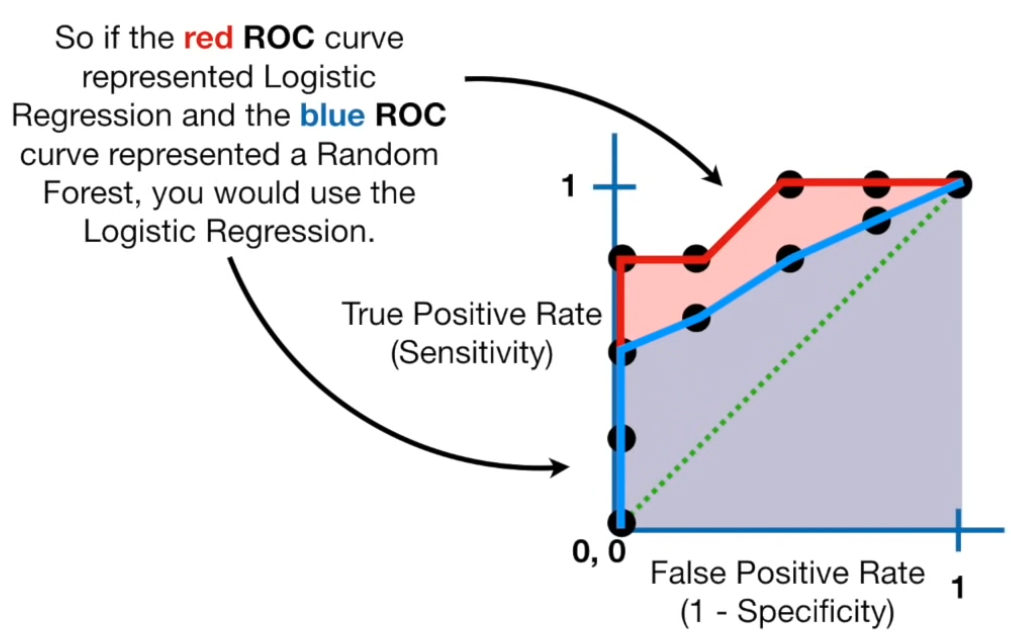
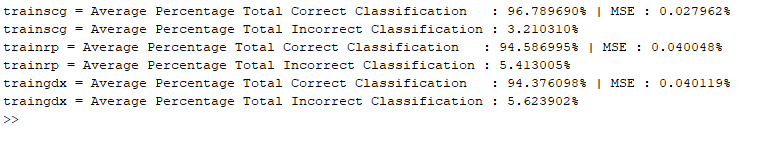


Figure 28 AUC helps to decide which categorization method is better





Στην παραπάνω εικόνα βλέπουμε το μέσο όρο ύστερα από 15 επανάληψης με πληθώρα τεστ (μέσα από το script Figure 17 Part 1 of script Figure 18 Part 2 of script) καθώς και τον υπολογισμό της συνάρτησης απώλειας (Loss function).

##### Generalization

Γενίκευση είναι η ικανότητα του νευρωνικού δικτύου να αποφύγει το λεγόμενο overfitting που συμβαίνει ρυθμίζοντάς λάθος τις αρχικοποιήσημες παραμέτρους του. Δηλαδή το πλαίσιο της εκπαίδευσης του είναι καλύτερο από ποτέ φτάνοντας στο ελάχιστο σημείο σφάλματος άμεσα, όμως χάνει την δυνατότητα να πέτυχει νέα κατηγοριοποίηση/πρόβλεψη των δεδομένων σε νέα άγνωστα δεδομένα. Στο generalization/regularization η καμπύλες κυλούν ομαλά προς το ελάχιστο ποσοστού λάθους (train + test + valid) που αυτό σημαίνει τα hidden layers και neurons per layer έχουν ρυθμιστεί σωστά για το εκάστοτε πρόβλημα.

## Conclusion (Michail Markou)

Για την διεκπεραίωση εκπαίδευσης ενός νευρωνικού δικτύου ακολουθούμε ένα κοινό standard best practice workflow όμως δεν υπάρχουν best practices για την κατασκευή του πάρα μόνο συνέχει προσπάθεια και παρατήρηση του σχεδιαστή για τα αποτελέσματα που φέρνει διότι δεν υπάρχει το ιδανικό με βάση το εκάστοτε πρόβλημα (trial and error). Από την συλλογή ακατέργαστων δεδομένων και ομαδοποιήσεις/επεξεργασίας τους ως των διαχωρισμό τους σε δεδομένα τροφοδοσίας του δικτύου και επαλήθευσης συμβάλει όλη η διαδικασία στη σωστή σχεδίαση ώστε το νευρωνικό δίκτυο να γίνει consumer ready/production ready εννοώντας ότι οι προβλέψεις είναι χρήσιμες και ακολουθούν συγκεκριμένα policies τόσο στο ιδεολογικό κομμάτι όσο και στο τεχνικό. Ένα μοντέλο τεχνίτης νοημοσύνης μας δίνει την δυνατότητα να μελετήσουμε τον κόσμο μέσα από τα big data που εμείς αδυνατούμε να επεξεργαστούμε η να βρούμε μοτίβα στα προβλήματα μέσα από τα πληθώρα παραδείγματα ώστε το μοντέλο σε αντίθεση με εμάς είναι ικανό να ανάλυση αλλά και να εφαρμόσει της γνώσεις του αναγνωρίζοντας τα μελλοντικά προβλήματα από patterns που προκύπτουν από άγνωστα δεδομένα. Μια προσομοίωση του ανθρώπινου εγκεφάλου με στοχαστικές τάσεις.

Στο παράδειγμά μας βλέπουμε ότι εχει προϋπόθει να εφαρμόζεται καθώς χρησιμοποιείται αυτοματοποιημένη διαδικασία εισαγωγής ροής δεδομένων με κοινό structure αναγνωρίζοντας τα χρήσιμα από τα περιττά δεδομένα (data cleansing) στο επίπεδο της προ-επεξεργασίας. Καθώς μετέπειτα ακολουθεί η σχεδίαση του μέσα από επιλογές πληθώρων αλγορίθμων και παραμέτρων (tuning) ώστε να γίνει ένα συνάθροισα και εύρεση μέσου όρου για το ποιος αλγόριθμος/τεχνική έχει το καλύτερο αποτέλεσμα στο παρών πρόβλημα. Λόγο της φύσης της επιβλέπουσας μηχανικής μάθησης ένας «δάσκαλος» «επιβλέπων» «καθηγητής» you might say πρέπει να είναι παρών σε όλη την διαδικασία του workflow/pipeline όχι μόνο σαν παρατηρητής αλλά και σαν συντονιστής για αξιολόγηση και διορθώσει τον αποτελεσμάτων.

Τέλος όλοι η διαδικασία εκπαίδευσης κρίνεται στο ποσό καλά αναγνωρίζει άγνωστα δεδομένα βρίσκοντας τα μοτίβα μέσα από αυτά δηλαδή τα test data είναι στην ουσία το κομμάτι που θα κρίνει ένα production ready machine learning model network.

Pes ta parameters apo diki sou gnwmi kai ti kanoun bale kai kwdika

Pes kai alla gia alla px ama ksekiname me validation etc

Μπορεις να πεις απ το βιντεο του baby to phD sto youtube pws kanis dio simio(x,y) ama apexoun poso metaksi tous? Kanis kapoai afairesi vector apo game programming(to xis fav sto youtube ekinon se c++) gia apostasy semeion thewritika hyperplane den mporis edw ama kanis ploit me maths kapws !? pes oti mporoume polla grafimata na bgalume analoga to math function input put tha ipologisi to graph output plot plane… sigkrine kai me to learner app machine (asto regression?)

# Bibliography

|  |  |
| --- | --- |
| [1] | "how can i changhe the transfer function of output layer of neural network?," mathworks, [Online]. Available: https://www.mathworks.com/matlabcentral/answers/84931-how-can-i-changhe-the-transfer-function-of-output-layer-of-neural-network. [Accessed 12 12 2021]. |
| [2] | "What is a Learning Rate in a Neural Network?," [Online]. Available: https://www.machinecurve.com/index.php/2019/11/06/what-is-a-learning-rate-in-a-neural-network/. [Accessed 12 12 2021]. |
| [3] | "What is a Hidden Layer?," [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/hidden-layer-machine-learning#:~:text=In%20neural%20networks%2C%20a%20hidden%20layer%20is%20located,transformations%20of%20the%20inputs%20entered%20into%20the%20network.. [Accessed 12 12 2021]. |
| [4] | "Neuron," [Online]. Available: http://www.biologyreference.com/Mo-Nu/Neuron.html#:~:text=The%20neuron%20%28nerve%20cell%29%20is%20the%20fundamental%20unit,to%20rapidly%20send%20signals%20across%20physiologically%20long%20distances.. [Accessed 12 12 2021]. |
| [5] | "Neural network: activation function vs transfer function," [Online]. Available: https://intellipaat.com/community/16651/neural-network-activation-function-vs-transfer-function#:~:text=In%20machine%20learning%2C%20the%20sums%20of%20each%20node,function%22%20is%20more%20commonly%20used%20in%20signal%20processing.. [Accessed 12 12 2021]. |
| [6] | "Scaled conjugate gradient backpropagation," [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/trainscg.html. [Accessed 12 12 2021]. |
| [7] | "Resilient backpropagation," [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/trainrp.html?searchHighlight=trainrp&s\_tid=srchtitle\_trainrp\_1. [Accessed 12 12 2021]. |
| [8] | "Gradient descent with momentum and adaptive learning rate backpropagation," [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/traingdx.html?searchHighlight=traingdx&s\_tid=srchtitle\_traingdx\_1. [Accessed 12 12 2021]. |
| [9] | "5 algorithms to train a neural network," neuraldesigner, [Online]. Available: https://www.neuraldesigner.com/blog/5\_algorithms\_to\_train\_a\_neural\_network#ConjugateGradient. [Accessed 12 12 2021]. |
| [10] | "What is an Epoch?," [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/epoch. [Accessed 12 12 2021]. |
| [11] | "ROC and AUC, Clearly Explained! | Logistic Regression," [Online]. Available: https://www.youtube.com/watch?v=4jRBRDbJemM. [Accessed 12 12 2021]. |
| [12] | "What is meaning of mu in artificial neural network (NNTOOL) MATLAB?," researchgate, [Online]. Available: https://www.researchgate.net/post/What-is-meaning-of-mu-in-artificial-neural-network-NNTOOL-MATLAB. [Accessed 12 12 2021]. |

# Appendix

## Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Convergence | Completed the change of calculation in the network that is disturbed by the normal flow (recalculation). A state where the model is aware of loss that has been settled within an error range around the final value and won’t improve further. (a known state of information facts) |

1. Each run produces different accuracy but overall SVM is better than Neural Networks here. [↑](#footnote-ref-1)
2. Usually on columns (depends on implementation) e.g., 10x569 input, 1x569 target/output [↑](#footnote-ref-2)
3. Prepare validation set out of training set (e.g., kfold CV (cross-validation)) that protects against overfitting by partitioning the data set into folds and estimating accuracy of each fold for small data sets or holdout validation for bigger data sets or no validation data to begin with for a NN testing purposes of how its designed in its core before stepping the actual intense tests. [↑](#footnote-ref-3)
4. Usually, a harmony of the visual output line parameters and not far apart should be consistent among plots [↑](#footnote-ref-4)
5. Keep an average score in a holder place for comparing the best effect [↑](#footnote-ref-5)
6. Το train δηλαδή κατηγοριοποιεί με βάση αυτά που εχει δει (pattern recognition classification) ενώ το τεστ κατατάζει με βάση αυτά που εχει εκπαιδευτή χωρίς να τα έχει δει δηλαδή να έχει χτίσει σχέσης μεταξύ των δεδομένων για αναγνωρίσει (δεν κάνει αναγνωρίσει μόνο κατάταξη ενώ train αναγνωρίζει (is influenced από το input με επίδραση αλλαγής neuron), χτίζει σχέσης και κατατάζει). [↑](#footnote-ref-6)
7. Για αυτό αναμένουμε κάθε φορά που τρέχουμε το δίκτυο παίρνουμε διαφορετική καμπύλη εκπαίδευσης φτάνοντας σε διαφορετικό ελάχιστο. [↑](#footnote-ref-7)
8. Το 0 δεν είναι ρεαλιστικό σαν στόχο error. [↑](#footnote-ref-8)
9. Δηλαδή σε κάθε τρέξιμο δεν θα έχει βελτιώσει στα errors του test data [↑](#footnote-ref-9)
10. Στα Hidden layers συστήνεται sigmoid transfer function [↑](#footnote-ref-10)
11. Δηλαδή είναι non-linear νευρώνια (MLP perceptrons) με οποιαδήποτε διαφορίσιμη συνάρτηση e.g., [↑](#footnote-ref-11)
12. Traingdx consists of special extra parameters of learning rate and momentum which they are tend to help in overcoming local minimum threshold errors. Command to display *net.trainParam.lr* or *.mc* [12] Figure 12 Figure 10 [↑](#footnote-ref-12)
13. Weight initialization occurs on the fly during train-data stage process but bias is pre-defined beforehand (with zeros). We can evaluate this by running *net.b{1}* and *net.IW{1}* in the console or script window before train the NN with train-data input and after them Figure 9. Its worth noting that both of them via mathematical functions are being manipulated so manual override is not advised. Further we can examine *net.biases{1} net.layerWeights{2} net.inputWeights{1} net.layerWeights{2}.learnParam* which differs from *net.trainParam.mc net.trainParam.lr* that has been assigned in script code *net.trainParam.lr = 0.1 net.trainParam.mc = 0.4* Figure 14Figure 12 [↑](#footnote-ref-13)