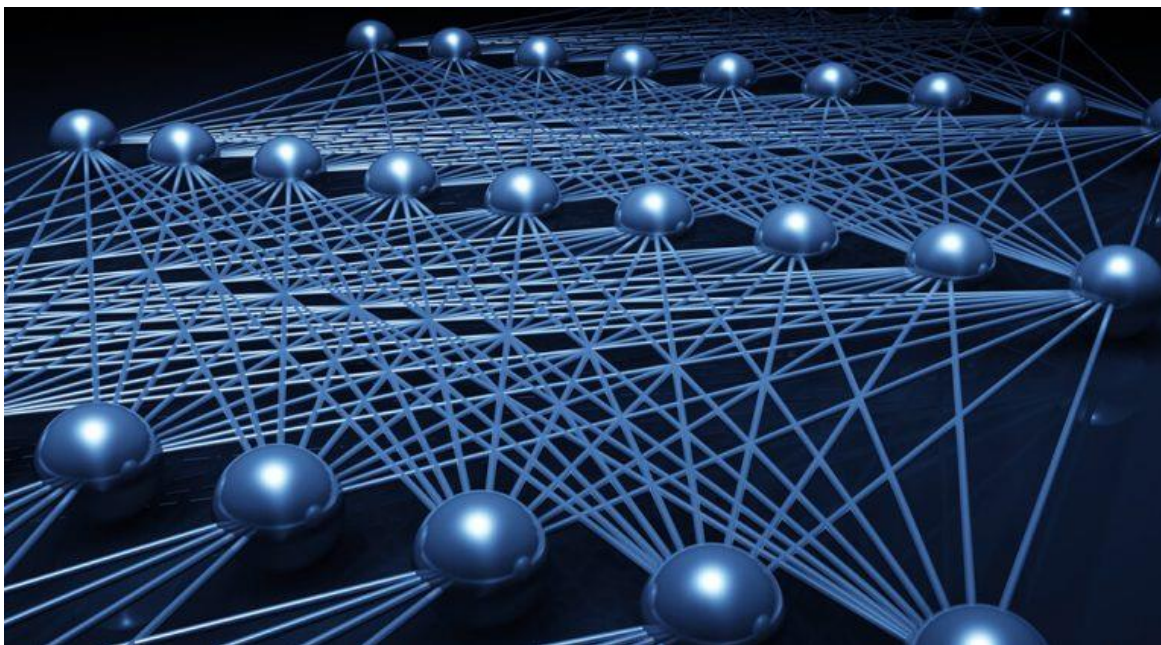


ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ



## 2η Εργασία στο Μεταπτυχιακό μάθημα της Βελτιστοποίησης



### ΤΙΤΛΟΣ ΕΡΓΑΣΙΑΣ

*Εκπαίδευση Πιθανοτικού Νευρωνικού Δικτύου με μεθόδους χωρίς παραγώγους για δεδομένα καρκίνου του μαστού*

### Ομάδα

Βαρδάκας Γεώργιος AM: 432

Σεβεντικίδου Χριστίνα AM: 439

Ντούσια Μαργαρίτα AM: 248

*Υπεύθυνος Καθηγητής: Κ. Παρσόπουλος*

## **ΠΕΡΙΕΧΟΜΕΝΑ**

ΕΙΣΑΓΩΓΗ .....	4
1. Probabilistic neural network.....	5
1.1 Λειτουργία του Πιθανοτικού Νευρωνικού Δικτύου .....	5
2. Μέθοδοι προς υλοποίηση .....	6
2.1 Η μέθοδος Nelder-Mead .....	6
2.2 Η μέθοδος τυχαίας αναζήτησης με χρήση Gaussian κατανομής.....	7
2.3 Η μέθοδος τυχαίου περιπάτου με μεταβλητό βήμα .....	7
2.4 Η μέθοδος Simulated annealing.....	8
3. Μέθοδοι διαχωρισμού .....	9
4. Πειράματα .....	10
4.1 Classification Error Results .....	11
4.1.1 Αποτελέσματα για 50 επαναλήψεις στο trainset .....	12
4.1.2 Αποτελέσματα για 100 επαναλήψεις στο trainset .....	13
4.1.3 Αποτελέσματα για 50 επαναλήψεις στο testset .....	14
4.1.4 Αποτελέσματα για 100 επαναλήψεις στο testset .....	15
4.2 Mean Square Error Results.....	16
4.2.1 Αποτελέσματα για 50 επαναλήψεις στο trainset .....	16
4.2.2 Αποτελέσματα για 100 επαναλήψεις στο trainset .....	17
4.2.3 Αποτελέσματα για 50 επαναλήψεις στο testset .....	19
4.2.4 Αποτελέσματα για 100 επαναλήψεις στο testset .....	20
5. Συγκριτική ανάλυση αποτελεσμάτων και αξιολόγηση .....	21
6. Δομή αναπτυσσόμενου κώδικα .....	22
7. References .....	25

## **ΕΙΚΟΝΕΣ**

Εικόνα 1.1.1: Probabilistic Neural Network.....	6
---	---

## **ΠΙΝΑΚΕΣ**

Πίνακας 2.2 1: Τυπικές τιμές παραμέτρων μεθόδου Nelder-Mead.....	7
Πίνακας 4.1: Τιμές αρχικοποίησης μεθόδου Nelder-Mead .....	10
Πίνακας 4.2: Τιμές αρχικοποίησης μεθόδου Random Search with Gaussian Distribution .....	10
Πίνακας 4.3: Τιμές αρχικοποίησης μεθόδου Random Walk with Variable Step.....	11

Πίνακας 4.4: Τιμές αρχικοποίησης μεθόδου Simulated annealing .....	11
Πίνακας 4.1.1.1: Trainset classification accuracy - Statistics -50 iterations .....	12
Πίνακας 4.1.1.2: T-test p-values at significance level $\alpha=0.05$ on trainset -50 iterations.....	13
Πίνακας 4.1.2.1: Trainset classification accuracy - Statistics -100 iterations .....	13
Πίνακας 4.1.2.2: T-test p-values at significance level $\alpha=0.05$ on trainset -100 iterations.....	14
Πίνακας 4.1.3 1: Testset classification accuracy - Statistics -50 iterations .....	14
Πίνακας 4.1.3.2: T-test p-values at significance level $\alpha=0.05$ on testset -50 iterations .....	15
Πίνακας 4.1.4.1: Testset classification accuracy - Statistics -100 iterations.....	15
Πίνακας 4.1.4.2: T-test p-values at significance level $\alpha=0.05$ on testset -100 iterations .....	16
Πίνακας 4.2.1.1: Trainset classification accuracy - Statistics -50 iterations .....	16
Πίνακας 4.2.1.2: T-test p-values at significance level $\alpha=0.05$ on trainset -50 iterations.....	17
Πίνακας 4.2.2.1: Trainset classification accuracy - Statistics -100 iterations .....	18
Πίνακας 4.2.2.2: T-test p-values at significance level $\alpha=0.05$ on trainset -100 iterations.....	18
Πίνακας 4.2.3.1: Testset Classification accuracy - Statistics -50 iterations .....	19
Πίνακας 4.2.3.2: T-test p-values at significance level $\alpha=0.05$ on testset -50 iterations .....	19
Πίνακας 4.2.4.1: Testset classification accuracy - Statistics -100 iterations.....	20
Πίνακας 4.2.4.2: T-test p-values at significance level $\alpha=0.05$ on testset -100 iterations .....	20
Πίνακας 5.1: Trainset mean classification accuracy .....	21
Πίνακας 5.2: Testset mean classification accuracy.....	21

## **ΔΙΑΓΡΑΜΜΑ**

Διάγραμμα 4.1.1.1: Boxplot of classification accuracy on trainset -50 iterations.....	12
Διάγραμμα 4.1.2.1: Boxplot of classification accuracy on trainset -100 iterations.....	13
Διάγραμμα 4.1.3.1: Boxplot of classification accuracy on testset -50 iterations .....	14
Διάγραμμα 4.1.4.1: Boxplot of classification accuracy on testset -100 iterations .....	15
Διάγραμμα 4.2.1.1: Boxplot of classification accuracy on trainset -50 iterations.....	17
Διάγραμμα 4.2.2.1: Boxplot of classification accuracy on trainset -100 iterations.....	18
Διάγραμμα 4.2.3.1: Boxplot of classification accuracy on testset -50 iterations .....	19
Διάγραμμα 4.2.4.1: Boxplot of classification accuracy on testset -100 iterations .....	20

## ΕΙΣΑΓΩΓΗ

Σκοπός της παρούσας εργασίας είναι η εκπαίδευση του πιθανοτικού νευρωνικού δικτύου (π.ν.δ. - probabilistic neural network), όπως αυτό έχει παρουσιαστεί και αναλυθεί στην παρεχόμενη μελέτη των V. L. GEORGIOU et al. [1], για τις ακόλουθες μεθόδους:

- Nelder-Mead
- Τυχαία αναζήτηση με χρήση Gaussian κατανομής
- Τυχαίος περίπατος με μεταβλητό βήμα
- Simulated annealing

για ένα από τα διαθέσιμα datasets **1) E.coli dataset**, **2) Breast Cancer dataset** και **3) Yeast dataset** χρησιμοποιώντας έναν από τους επακόλουθους τρόπους διαχωρισμού **1) Stratified Random Sampling**, **2) λ-Fold Cross-Validation** ή **3) Train-Validation-Test Partitioning**.

Η παρούσα εργασία δομείται στις παρακάτω ενότητες:

Στην **Ενότητα 1** θα γίνει η περιγραφή του π. ν. δ. σύμφωνα με τη δομή και τη λειτουργία του όπως αυτή παρουσιάζεται και αναπτύσσεται στη δοθείσα έρευνα [1] και θα γίνει η προσαρμογή του στις ανάγκες και στα δεδομένα της ζητούμενης άσκησης.

Στην **Ενότητα 2** θα γίνει μία σύντομη περιγραφή των μεθόδων Nelder-Mead, τυχαίας αναζήτησης με χρήση Gaussian κατανομής, τυχαίου περιπάτου με μεταβλητό βήμα και τέλος της μεθόδου Simulated annealing που χρησιμοποιήθηκαν.

Στην **Ενότητα 3** θα γίνει μια σύντομη περιγραφή των τριών μεθόδων διαχωρισμού 1) Stratified Random Sampling, 2) λ-Fold Cross-Validation και 3) Train-Validation-Test Partitioning, όπως αυτές έχουν αναφερθεί στην εργασία [1].

Στην **Ενότητα 4** θα πραγματοποιηθούν τα διαφορετικά πειράματα για διαφορετικές παραμέτρους των αλγορίθμων, θα καταγραφούν και θα παρουσιαστούν τα αποτελέσματα διαμέσου πινάκων, γραφημάτων, boxplots κ.λ.π.

Στην **Ενότητα 5** θα ακολουθήσει συγκριτική ανάλυση των αποτελεσμάτων για την εύρεση της καλύτερης επιλογής αλγορίθμου συγκριτικά με τους υπόλοιπους καθώς και ενδεικτική σύγκριση με τα δημοσιευμένα αποτελέσματα.

Τέλος στην **Ενότητα 6** θα παρουσιαστεί και θα αναλυθεί η δομή του αναπτυσσόμενου κώδικα.

**Στην εκπόνηση της παρούσας εργασίας η μέθοδος διαχωρισμού η οποία θα χρησιμοποιηθεί είναι η λ-Fold Cross-Validation και το προς εξέταση dataset είναι το Breast Cancer.**

## 1. Probabilistic neural network

Η αρχιτεκτονική των Τεχνητών Νευρωνικών Δικτύων (ΤΝΔ) βασίζεται σε αυτή των βιολογικών Νευρωνικών Δικτύων. Τα ΤΝΔ μπορούν να εκτελούν μαζικούς υπολογισμούς και αποτελούνται από διάφορα επίπεδα μονάδων επεξεργασίας που ονομάζονται νευρώνες. Σκοπός των ΤΝΔ στην περιοχή της μάθησης υπό επίβλεψη (supervised learning) και συγκεκριμένα στο πρόβλημα της ταξινόμησης (classification) είναι ο διαχωρισμός των δεδομένων σε κατηγορίες. Κατά τη διαδικασία της υπό επίβλεψη μάθησης ξεκινάμε με τυχαίες αρχικοποιήσεις των παραμέτρων και θέτουμε τις τιμές των εισόδων και των στόχων εκμάθησης του δικτύου. Εισάγουμε δηλαδή τα παραδείγματα εκπαίδευσης στο νευρωνικό δίκτυο και στόχος είναι η σωστή ταξινόμησή τους με βάση την κατηγορία τους. Κατά τη διαδικασία εκπαίδευσης το δίκτυο ελαχιστοποιεί την συνάρτηση σφάλματος προσαρμόζοντας κατάλληλα τις παραμέτρους του.

### 1.1 Λειτουργία του Πιθανοτικού Νευρωνικού Δικτύου

Στην δοθείσα εργασία των V. L. GEORGIOU et al. [1], προτείνεται ένα αυτοπροσαρμοζόμενο πιθανοτικό νευρωνικό δίκτυο (Probabilistic Neural Network a.k.a PNN), το οποίο ενσωματώνει αλγόριθμους βελτιστοποίησης χωρίς παραγώγους με σκοπό τον προσδιορισμό των βέλτιστων τιμών των παραμέτρων του. Το PNN δομείται σε 4 επίπεδα **1) input layer, 2) pattern layer, 3) summation layer** και **4) output layer** όπως παρουσιάζεται σχηματικά στην Εικόνα 1.1.1.

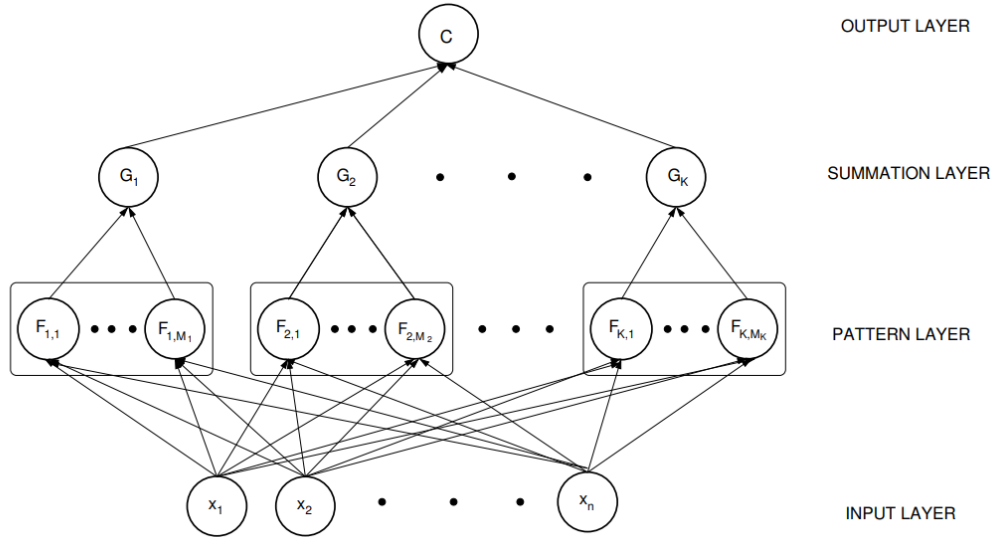
Σε κάθε έναν από τους νευρώνες εισόδου  $x_i$ , με  $i=1,2,...,n$  όπου  $n$  το πλήθος των features, εφαρμόζεται ένα διάνυσμα εισόδου  $X \in R^n$  το οποίο περνάει στο pattern layer. Για κάθε  $X_{i,k}$  να αποτελεί το κέντρο της **Gaussian kernel** συνάρτησης ενεργοποίησης (1), με  $i=1,2,...,M_k$ , όπου  $M_k$  το πλήθος των δεδομένων της κλάσης  $k$  και  $k=1,2,...,K$  το πλήθος των κλάσεων, δημιουργείται ένας νευρώνας στο pattern layer. Στο pattern layer περιέχονται οι υπολογιστικοί νευρώνες του δικτύου και οι έξοδοι των νευρώνων ανά κλάση, από το layer αυτό, αθροίζονται και καταλήγουν στο summation layer. Το summation layer του δικτύου περιέχει τα συνολικά αθροίσματα κάθε κλάσης του pattern layer καθώς και την εκ των προτέρων πιθανότητα (prior probability) κάθε κατηγορίας. Στο summation layer υπολογίζουμε το γινόμενο της prior probability  $w_k$  με το αντίστοιχο άθροισμα  $sum_k$ , ( $G_k = w_k * sum_k$  όπως φαίνεται στη σχέση (2)) για κάθε κατηγορία το οποίο περνάει σαν είσοδο στο output layer. Αυτό με τη σειρά του ταξινομεί το δεδομένο στην κατηγορία με το μεγαλύτερο άθροισμα. **Στην παρούσα εργασία θεωρήσαμε τις πιθανότητες των κατηγοριών ισοπίθανες.**

**Σκοπός της εκπαίδευσης του Πιθανοτικού Νευρωνικού Δικτύου (ΠΝΔ) είναι η βελτιστοποίηση των παραμέτρων της διακύμανσης  $\sigma^2$  μέσω της ελαχιστοποίησης του σφάλματος ταξινόμησης.**

Αναφορικά με τη συνάρτηση ενεργοποίησης (1) **ο πίνακας  $\Sigma$  είναι ένας διαγώνιος πίνακας**, διάστασης  $n \times n$ , με  $n$  το πλήθος των χαρακτηριστικών (features) των δεδομένων, όπου στη διαγώνιο του πίνακα βρίσκονται οι παράμετροι  $\sigma^2$ .

$$f_{i,k}(X) = \frac{1}{(2\pi)^{n/2} \det(\Sigma)^{1/2}} \exp \left( -\frac{1}{2} (X - X_{i,k})^T \Sigma^{-1} (X - X_{i,k}) \right) \quad (1)$$

$$G_k(X) = w_k \sum_{i=1}^{M_k} f_{i,k}(X). \quad (2)$$



Εικόνα 1.1.1: Probabilistic Neural Network

## 2. Μέθοδοι προς υλοποίηση

Στην παρούσα ενότητα θα ακολουθήσει μια σύντομη περιγραφή των προς υλοποίηση μεθόδων βελτιστοποίησης για τις ανάγκες διεκπεραίωσης της παρούσας εργασίας. Προς εξέτασιν μέθοδοι αποτελούν οι **Nelder-Mead**, **τυχαίας αναζήτησης με χρήση Gaussian κατανομής**, **τυχαίου περιπάτου με μεταβλητό βήμα** και **Simulated annealing**.

### 2.1 Η μέθοδος Nelder-Mead

Η μέθοδος Nelder-Mead συγκαταλέγεται στις μεθόδους άμεσης αναζήτησης (direct search). Στη μέθοδο αυτή χρησιμοποιείται ένα σύνολο σημείων  $x_1, x_2, \dots, x_{n+1}$  τα οποία αποτελούν κορυφές σε ένα  $n$ -άπλοκο. Τα σημεία ταξινομούνται ως προς τη συναρτησιακή τους τιμή, από την μικρότερη προς τη μεγαλύτερη,  $f_1 \leq f_2 \leq \dots \leq f_n \leq f_{n+1}$ , και υπολογίζεται το κέντρο βάρους των  $n$  καλύτερων σημείων σύμφωνα με τη σχέση (3).

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (3)$$

Η μέθοδος Nelder-Mead παράγει νέα σημεία με βάση 5 τελεστές  $\rho$  και οι οποίοι είναι οι εξής: **1) Ανάκλαση (reflection)**, **2) Επέκταση (expansion)**, **3) Εξωτερική συστολή (external contraction)**, **4)**

**Εσωτερική συστολή (internal contraction) και 5) Συρρίκνωση (shrink).** Για τις περιπτώσεις 1-4 η σχέση υπολογισμού (4) δίνεται παρακάτω:

$$x = (1 + \rho)\bar{x} - \rho x_{n+1} \quad (4)$$

ενώ η περίπτωση της συρρίκνωσης υπολογίζεται από την εξής σχέση:

$$x_i^{(k+1)} = x_1^{(k)} - \frac{x_i^{(k)} - x_1^{(k)}}{2}, \quad \mu\epsilon \ i = 2, 3, \dots, n+1 \quad (5)$$

Σε κάθε κύκλο του αλγορίθμου θα πραγματοποιηθεί μία από τις διαδικασίες 1 έως 5, και το άπλοκο θα ανανεωθεί αντίστοιχα. Για τις περιπτώσεις 1-4 θα παράγεται ένα νέο σημείο ( $x_{ref}^{(k+1)}, x_{exp}^{(k+1)}, x_{exc}^{(k+1)}, x_{inc}^{(k+1)}$ , αντιστοίχως) το οποίο θα αντικαθιστά το χειρότερο σημείο, δηλ. το τελευταίο στη διάταξη. Στην περίπτωση που δεν πετύχουν αυτά θα εφαρμοστεί η περίπτωση 5, το άπλοκο θα συρρικνωθεί, θα γίνει δηλ. μια ταυτόχρονη συστολή από όλες τις κορυφές του απλόκου προς την καλύτερη όπως αυτό υπολογίζεται από την σχέση (5):

Οι **παράμετροι** της μεθόδου ακολουθούν την διάταξη  $-1 < \rho_{inc} < 0 < \rho_{exc} < \rho_{ref} < \rho_{exp}$  ενώ οι **τυπικές τιμές** οι οποίες θα χρησιμοποιηθούν στην παρούσα εργασία παρουσιάζονται στον Πίνακα 2.2.1 που ακολουθεί:

$\rho_{inc}$	$-\frac{1}{2}$
$\rho_{exc}$	$\frac{1}{2}$
$\rho_{ref}$	1
$\rho_{exp}$	2

Πίνακας 2.2 1: Τυπικές τιμές παραμέτρων μεθόδου Nelder-Mead

## 2.2 Η μέθοδος τυχαίας αναζήτησης με χρήση Gaussian κατανομής

Η μέθοδος εκκινεί με ένα τυχαίο αρχικό διάνυσμα παραμέτρων  $x_k$ , το οποίο αποτελεί το κέντρο της Gaussian κατανομής  $N(x_k, \sigma)$  από την οποία δειγματοληπτούμε σημεία. Για το αρχικό  $x_k$  υπολογίζεται η συναρτησιακή του τιμή  $f(x_k)$  και έπειτα για  $k_{max}$  επαναλήψεις ακολουθεί η εξής διαδικασία: δειγματοληπτείται ένα διάνυσμα παραμέτρων  $x_{k+1}$  από την κατανομή  $N(x_k, \sigma)$  και υπολογίζεται για αυτό η συναρτησιακή του τιμή  $f(x_{k+1})$ . Εάν η συναρτησιακή τιμή έχει μειωθεί, δηλ.  $f(x_{k+1}) < f(x_k)$  τότε στην επόμενη επανάληψη θέτουμε το νέο  $x_k = x_{k+1}$ . Μόλις ολοκληρωθούν οι  $k_{max}$  επαναλήψεις, η μέθοδος επιστρέφει το βέλτιστο διάνυσμα παραμέτρων.

## 2.3 Η μέθοδος τυχαίου περιπάτου με μεταβλητό βήμα

Στη συγκεκριμένη μέθοδο ξεκινάμε με το αρχικό διάνυσμα παραμέτρων  $x_k$ , μέγεθος βήματος  $s$  και δύο scaling factors  $\alpha, \beta$  με  $\alpha > 1$  και  $\beta \in (0, 1)$ . Σε κάθε επανάληψη του αλγορίθμου επιλέγεται μια τυχαία διεύθυνση  $\rho_k$  ομοιόμορφα στο διάστημα  $[-1, 1]$  η οποία στη συνέχεια κανονικοποιείται. Στη διεύθυνση



αυτή κάνουμε ένα βήμα μεγέθους  $s$  και για το καινούργιο σημείο  $x_t = x_k + s \rho_k$  ελέγχουμε τη συναρτησιακή του τιμή  $f(x_t)$ . Αν  $f(x_t) < f(x_k)$  αυξάνουμε το βήμα  $s$  πολλαπλασιάζοντας με τη σταθερά  $\alpha$  και για το καινούργιο σημείο  $x_l = x_k + s_l \rho_k$  ελέγχουμε τη συναρτησιακή του τιμή  $f(x_l)$ . Αν  $f(x_l) < f(x_t)$  τότε το καινούργιο σημείο μας  $x_{k+1} = x_l$  και  $s_{k+1} = s_l$ . Στην αντίθετη περίπτωση όπου  $f(x_l) > f(x_t)$  το νέο σημείο είναι το  $x_{k+1} = x_t$  και  $s_{k+1} = s_k$ . Αν  $f(x_t) > f(x_k)$  τότε μειώνουμε το βήμα πολλαπλασιάζοντας με το  $\beta$  και για το καινούργιο σημείο  $x_m = x_k + s_m \rho_k$  ελέγχουμε τη συναρτησιακή του τιμή  $f(x_m)$ . Αν  $f(x_m) < f(x_k)$  τότε  $x_{k+1} = x_m$  και  $s_{k+1} = s_m$ . Στο τέλος της διαδικασίας ανανεώνουμε το καλύτερο σημείο  $x_{best}$ .

## 2.4 Η μέθοδος Simulated annealing

Βασικό χαρακτηριστικό της μεθόδου είναι ότι κατά την εξερεύνηση του χώρου λύσεων του προβλήματος, αποδέχεται με κάποια πιθανότητα μεταβάσεις που δεν βελτιώνουν την αντικειμενική συνάρτηση. Με αυτόν τον τρόπο ευελπιστούμε ότι θα αποφύγουμε τον εγκλωβισμό σε περιοχές τοπικών ελαχίστων. Η μέθοδος simulated annealing χαρακτηρίζεται από δύο συναρτήσεις: τη συνάρτηση αποδοχής (Facc) και τη συνάρτηση ψύξης (U).

Σαν συνάρτηση αποδοχής χρησιμοποιούμε τη συνάρτηση Metropolis η οποία φέρει την παρακάτω μορφή:

$$Facc(x_k, y_{k+1}, t_k) = \min\left\{1, e^{\frac{-((f(y_{k+1}) - f(x_k)))}{t_k}}\right\} \quad (6)$$

με  $Facc(x_k, y_{k+1}, t_k)$  να παίρνει τιμές στο  $[0, 1]$ .

Πιο συγκεκριμένα το  $t_k$  είναι η παράμετρος θερμοκρασίας που μειώνεται κατά το πέρασμα των επαναλήψεων σύμφωνα με τη συνάρτηση ψύξης και ρυθμίζει τη συνάρτηση αποδοχής ενώ τα  $f(x_k)$  και  $f(y_{k+1})$  είναι δύο συναρτησιακές τιμές.

Σαν συνάρτηση ψύξης χρησιμοποιούμε την παρακάτω:

$$U(Z_k) = \beta (f(x_k) - f^*)^\alpha \quad (7)$$

Με  $\alpha, \beta > 0$  και  $f^*$  η ελάχιστη εκτιμώμενη συναρτησιακή τιμή.

Η μέθοδος εκκινεί με ένα αρχικό διάνυσμα παραμέτρων  $x_k$ , θερμοκρασία  $t_0$  και παραμέτρους  $\alpha, \beta, \rho$  όπου  $\alpha, \beta$  σταθερές και  $\rho$  το μέγεθος βήματος. Σε κάθε επανάληψη του αλγορίθμου επιλέγεται μια τυχαία διεύθυνση  $s_k$  ομοιόμορφα στο διάστημα  $[-1, 1]$  που στη συνέχεια κανονικοποιείται. Έπειτα υπολογίζεται το σημείο  $y_{k+1} = x_k + \rho s_k$  και η συναρτησιακή του τιμή  $f(y_{k+1})$ . Κάνοντας χρήση της συνάρτησης Metropolis υπολογίζουμε την πιθανότητα αποδοχής του σημείου όπως περιεγράφηκε παραπάνω στη σχέση (6)). Εάν η πιθανότητα αποδοχής είναι μεγαλύτερη από έναν τυχαίο αριθμό που δειγματοληπτούμε ομοιόμορφα στο  $[0, 1]$  ( $rand() \sim U([0, 1])$ ) τότε αποδεχόμαστε το καινούργιο σημείο σύμφωνα με τη σχέση που ακολουθεί:

$$x_{k+1} = \begin{cases} y_{k+1}, & \text{αν } rand() \leq Facc(x_k, y_{k+1}, t_k) \\ x_k, & \text{διαφορετικά} \end{cases} \quad (8)$$



Σημεία τα οποία βελτιώνουν την συναρτησιακή τιμή της  $f(x)$  τα αποδεχόμαστε αμέσως σε αντίθετη περίπτωση τα δεχόμαστε με μια πιθανότητα. Ανανεώνουμε την παράμετρο θερμοκρασίας  $t_{k+1}$  σύμφωνα με τη συνάρτηση ψύξης που περιεγράφηκε και στο τέλος της διαδικασίας ανανεώνουμε το καλύτερο σημείο  $x_{best}$ .

### 3. Μέθοδοι διαχωρισμού

Στη δοθήσα εργασία των V. L. GEORGIU et al. [1] χρησιμοποιούνται 3 μέθοδοι διαχωρισμού οι οποίες είναι οι **1) Stratified Random Sampling**, **2) λ-Fold Cross-Validation** και **3) Train-Validation-Test Partitioning**.

**1). Stratified Random Sampling:** Στη συγκεκριμένη μέθοδο διαχωρισμού έχουμε ένα συνολικό data set μεγέθους  $N$  το οποίο διαιρείται σε  $K$  μη επικαλυπτόμενους υποπληθυσμούς (strata) για το σύνολο των οποίων ισχύει ότι:

$$N = \sum_{k=1}^K N_k \quad (9)$$

όπου  $K$  το συνολικό πλήθος των κλάσεων  $k \in \{1, 2, \dots, K\}$ . Από το κάθε stratum ανεξάρτητα επιλέγεται ένα τυχαίο δείγμα μεγέθους  $\alpha_k$  ενώ το τυχαίο δείγμα το οποίο θα χρησιμοποιηθεί σαν train set ορίζεται ως:

$$\alpha = \sum_{k=1}^K \alpha_k \quad (10)$$

ενώ τα υπόλοιπα δεδομένα αποτελούν το test set.

**2). λ-Fold Cross-Validation:** Στη μέθοδο αυτή διαιρούμε το αρχικό συνολικό data set σε  $\lambda$  ισομεγέθη subsets, κάθε ένα από τα οποία χρησιμοποιείται σαν test set ενώ τα υπόλοιπα σαν train set. Τέλος υπολογίζεται το μέσο σφάλμα σε όλες τις δοκιμές. Για τις ανάγκες υλοποίησης του δεύτερου project θα χρησιμοποιηθεί η παρούσα τεχνική με  $\lambda=10$ , διαμοιρασμός του data set σε 10 ισομεγέθη subsets (10-CV).

**3). Train-Validation-Test Partitioning [2]:** Στη συγκεκριμένη τεχνική τα συνολικά data χωρίζονται σε 3 subsets: **α) train set**, **β) validation set** και **γ) test set** με πιο συνήθεις αναλογίες:

- 70% train, 15% val, 15% test
- 80% train, 10% val, 10% test
- 60% train, 20% val, 20% test

Τα βήματα της τεχνικής αυτής είναι τα εξής:

- Τυχαία αρχικοποίηση του μοντέλου
- Εκπαίδευση κάθε μοντέλου στο train set
- Αξιολόγηση της απόδοσης κάθε train set στο validation set
- Επιλογή του μοντέλου με την καλύτερη απόδοση του validation set
- Αξιολόγηση του επιλεγμένου μοντέλου στο test set

## 4. Πειράματα

Στην τρέχουσα ενότητα παραθέτουμε τα αποτελέσματα των πειραμάτων για το dataset του Breast Cancer για κάθε μια μέθοδο βελτιστοποίησης που χρησιμοποιήθηκε για την εκπαίδευση του Πιθανοτικού Νευρωνικού Δικτύου.

Το Breast Cancer dataset αποκτήθηκε από το Ινστιτούτο Ογκολογίας του Πανεπιστημιακού Ιατρικού Κέντρου της Λουμπλιάνα. Το dataset περιλαμβάνει:

- 277 περιπτώσεις (ύστερα από εκκαθάριση των missing attribute values: 286 (συνολικά)-9 (missing))
- 2 κλάσεις: no-recurrence-events με 196 περιπτώσεις και recurrence-events με 81 περιπτώσεις
- 9 χαρακτηριστικά τα οποία είναι: Age, Menopause, tumor-size, inv-nodes, node-caps, deg-malig, breast, breast-quad και irradiat, τα οποία έπειτα από την εφαρμογή της μεθόδου One Hot Encoding έγιναν 39.

**Η μέθοδος One Hot Encoding** χρησιμοποιείται σε κατηγορηματικές μεταβλητές και τις μετασχηματίζει σε διανύσματα μήκους όσο και το πλήθος των τιμών της μεταβλητής. Συγκεκριμένα, δίνεται η τιμή 1 στη θέση της αντίστοιχης κατηγορίας ενώ στις υπόλοιπες η τιμή 0.

Η μέθοδος One Hot Encoding με παράδειγμα: Έστω μια τυχαία μεταβλητή class με τιμές recurrence-events και no-recurrence-events. Αυτή θα μετατραπεί σε ένα vector της μορφής:

*[class recurrence-events, class no-recurrence-events]*

με πιθανές τιμές [0,1] ή [1,0] ανάλογα με την κατηγορία του δεδομένου.

Για τις ανάγκες υλοποίησης της παρούσας εργασίας χρησιμοποιήθηκε η Cross Validation μέθοδος διαχωρισμού και τα δεδομένα διαιρέθηκαν σε 10 ισομεγέθεις κατηγορίες.

Αρχικά, στον Πίνακα 4.1.1 – 4.1.4 παραθέτουμε τις αρχικοποιήσεις κάποιων βασικών παραμέτρων για τις 4 μεθόδους που υλοποιήθηκαν: **Nelder-Mead (NM)**, **Random Search with Gaussian Distribution (RSGD)**, **Random Walk with Variable Step (RWVS)** και **Simulated annealing (SA)**.

Κάθε μέθοδος εκκινεί με ένα τυχαίο διάνυσμα παραμέτρων  $\sigma$  το οποίο κατά την είσοδό του στο νευρωνικό δίκτυο μετατρέπεται σε  $\sigma^2$ . Τα πειράματα έτρεξαν για 50 και 100 επαναλήψεις.

	$\sigma$ εκκίνησης	pinc	pexc	pref	pexp	kmax
NM	$[0, \sqrt{5}]$	- 1/2	1/2	1	2	50 & 100

Πίνακας 4.1: Τιμές αρχικοποίησης μεθόδου **Nelder-Mead**

	$\sigma$ εκκίνησης	$\sigma$	kmax
RSGD	$[0, \sqrt{5}]$	0.5	50 & 100

Πίνακας 4.2: Τιμές αρχικοποίησης μεθόδου **Random Search with Gaussian Distribution**

	σ εκκίνησης	α	β	s	kmax
<b>RWVS</b>	$[0, \sqrt{5}]$	1.5	0.5	1	50 & 100

Πίνακας 4.3: Τιμές αρχικοποίησης μεθόδου **Random Walk with Variable Step**

	σ εκκίνησης	t	α	β	ρ	f*	kmax
<b>SA</b>	$[0, \sqrt{5}]$	1	1	1	0,5	0	50 & 100

Πίνακας 4.4: Τιμές αρχικοποίησης μεθόδου **Simulated annealing**

Δοκιμάστηκαν δύο συναρτήσεις σφάλματος (αντικειμενική συνάρτηση που θέλουμε να ελαχιστοποιήσουμε) οι οποίες είναι οι: **mean squared error** και **classification error**.

- Η συνάρτηση **Mean Square Error** δίνεται από τον παρακάτω τύπο:

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (11)$$

με *MSE* να είναι το mean squared error, *n* ο αριθμός των data points,  $Y_i$  οι παρατηρούμενες τιμές και  $\hat{Y}_i$  οι προβλεπόμενες τιμές.

**Στην παρούσα εργασία λόγω του ότι τα  $Y_i, \hat{Y}_i$  είναι διανύσματα χρησιμοποιήθηκε η νόρμα 1.**

Το νευρωνικό δίκτυο για τη συγκεκριμένη συνάρτηση δίνει σαν έξοδο μια κατανομή πιθανότητας π.χ εάν η έξοδός μας είναι ένα διάνυσμα  $[0.8, 0.2]$  τότε το νευρωνικό θεωρεί ότι με πιθανότητα 0.8 το δεδομένο ανήκει στην κατηγορία A και με πιθανότητα 0.2 στην κατηγορία B.

- Η συνάρτηση **Classification Error** υπολογίζει το σφάλμα ταξινόμησης, δηλ. το πλήθος των λάθος ταξινομημένων δεδομένων στο train set.

## 4.1 Classification Error Results

Στη συγκεκριμένη ενότητα δίνεται αναλυτικά η παρουσίαση και αναπαράσταση των αποτελεσμάτων, ο στατιστικός έλεγχος υποθέσεων για κάθε πείραμα που έγινε και η σύγκριση μεταξύ των αλγορίθμων. Η αξιολόγησή μας βασίζεται στην μετρική **classification accuracy**.

Για να ελέγξουμε εάν οι διαφορές που παρατηρήσαμε είναι στατιστικά σημαντικές, κάνουμε στατιστικό έλεγχο υποθέσεων. Χρησιμοποιήσαμε παραμετρικό έλεγχο **t-test** ο οποίος υποθέτει ότι τα δύο δείγματα ακολουθούν κανονική κατανομή με ίσες διακυμάνσεις και ελέγχει τις μέσες τιμές. Με το τεστ αυτό πραγματοποιήθηκε η σύγκριση των **mean accuracies** των μεθόδων ανά δύο, με μηδενική υπόθεση ότι οι κατανομές των **mean accuracies** δεν διαφέρουν μεταξύ τους στατιστικά σημαντικά (ίσες μέσες τιμές) και εναλλακτική υπόθεση ότι οι κατανομές διαφέρουν, δηλαδή έχουν διαφορετικές μέσες τιμές. Το επίπεδο σημαντικότητας που χρησιμοποιήθηκε είναι **α=0.05**.

#### 4.1.1 Αποτελέσματα για 50 επαναλήψεις στο trainset

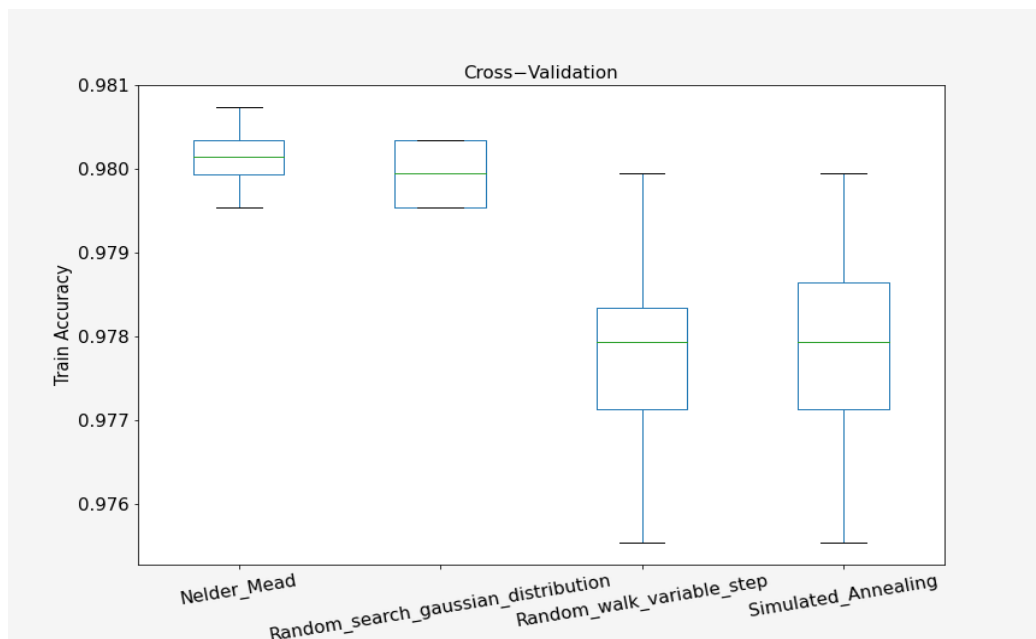
Στη συγκεκριμένη υποενότητα παρουσιάζονται τα αποτελέσματα των πειραμάτων για 50 επαναλήψεις στο trainset.

Όπως παρατηρείται στον Πίνακα 4.1.1.1 η μέθοδος **NM** επικρατεί όλων των υπόλοιπων μεθόδων για όλα τα στατιστικά μεγέθη (**Mean, Median, St.Dev., Min, Max**). Δεδομένου ότι τα στατιστικά μεγέθη αφορούν την ακρίβεια ταξινόμησης μίας μεθόδου, ιδανικά θα ισχύει ότι:

- τα **Mean, Median, Min** και **Max** λαμβάνουν όσο το δυνατόν μεγαλύτερες τιμές και
- η **St.Dev.** λαμβάνει όσο το δυνατόν χαμηλότερες τιμές

	Mean	Median	St.Dev.	Min	Max
NM	0.980091	0.980144	0.000324	0.979539	0.980744
RSGD	0.979997	0.979945	0.000345	0.979539	0.980349
RWVS	0.977871	0.97794	0.001054	0.975531	0.979945
SA	0.977818	0.977937	0.00109	0.975531	0.979945

Πίνακας 4.1.1.1: Trainset classification accuracy - Statistics -50 iterations



Διάγραμμα 4.1.1.1: Boxplot of classification accuracy on trainset -50 iterations

Η **NM** είναι η καλύτερη μέθοδος όπως φαίνεται παραπάνω, ωστόσο δε διαφέρει στατιστικά σημαντικά η μέση τιμή της από την **RSGD**, η οποία είναι η δεύτερη καλύτερη σε απόδοση. Τα αποτελέσματα αυτά επιβεβαιώνονται στατιστικά μέσω των p-values όπως προέκυψαν από το t-test. (Πίνακας 4.1.1.2).

	NM	RSGD	RWVS	SA
NM	-	<b>0.283764</b>	7.34E-16	9.81E-16
RSGD	-	-	4.91E-15	6.17E-15
RWVS	-	-	-	<b>0.847745</b>

Πίνακας 4.1.1.2: T-test p-values at significance level  $\alpha=0.05$  on trainset -50 iterations

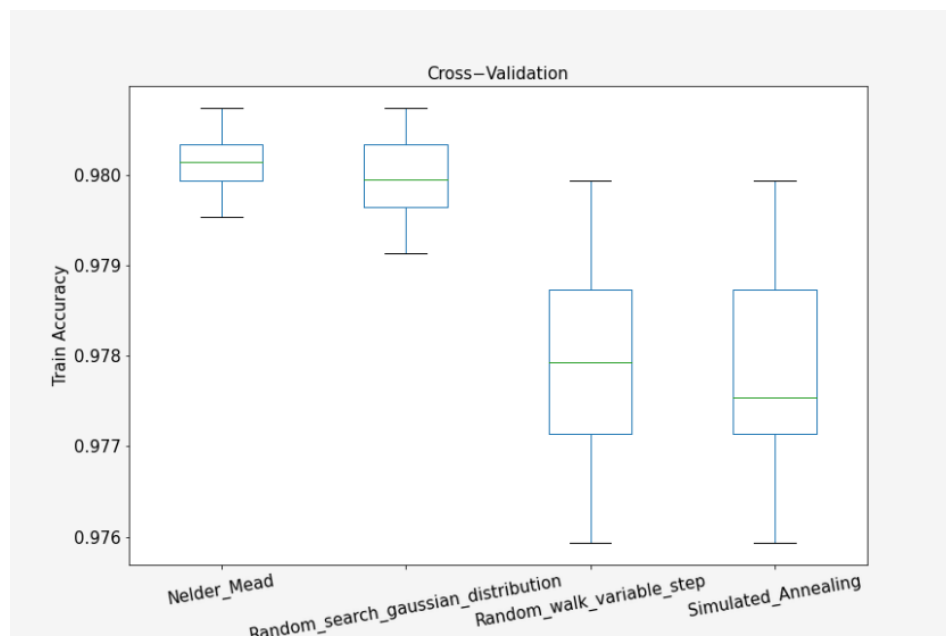
#### 4.1.2 Αποτελέσματα για 100 επαναλήψεις στο trainset

Στη συγκεκριμένη υποενότητα παρουσιάζονται τα αποτελέσματα των πειραμάτων για 100 επαναλήψεις στο trainset.

Όπως παρουσιάζεται στον παρακάτω πίνακα η μέθοδος **NM** επικρατεί όλων των υπόλοιπων μεθόδων για όλα τα στατιστικά μεγέθη. Επίσης παρατηρείται ότι στο συγκεκριμένο πείραμα η μέθοδος **RSGD** έχει δώσει παραπλήσια αποτελέσματα όσον αφορά τα στατιστικά μεγέθη **Mean**, **St.Dev.**, **Min** ενώ έχει την ίδια τιμή με την **RSGD** για **Max** (Πίνακας 4.1.2.1). Τα συμπεράσματα αυτά γίνονται αντιληπτά και στο αντίστοιχο boxplot (Διάγραμμα 4.1.2.1)

	Mean	Median	St.Dev.	Min	Max
NM	<b>0.980091</b>	<b>0.980144</b>	<b>0.000324</b>	<b>0.979539</b>	<b>0.980744</b>
RSGD	<b>0.980024</b>	0.979947	<b>0.000386</b>	<b>0.979139</b>	<b>0.980744</b>
RWVS	0.977911	0.977936	0.001026	0.975936	0.979942
SA	0.977858	0.977541	0.001045	0.975931	0.979942

Πίνακας 4.1.2.1: Trainset classification accuracy - Statistics -100 iterations



Διάγραμμα 4.1.2.1: Boxplot of classification accuracy on trainset -100 iterations

Στον Πίνακα 4.1.2.2 παρουσιάζονται τα αποτελέσματα του στατιστικού ελέγχου υποθέσεων t-test από τα οποία παρατηρούμε ότι οι μέθοδοι **NM** και **RSGD** δεν έχουν στατιστικά σημαντική διαφορά μεταξύ

τους, και ομοίως οι **SA** με την **RWVS**. Η καλύτερη μέθοδος σύμφωνα με τα παραπάνω είναι η **NM** και ακολουθεί **RSGD**.

	NM	RSGD	RWVS	SA
NM	-	<b>0.470627</b>	5.74E-16	4.21E-16
RSGD	-	-	3.99E-15	2.81E-15
RWVS	-	-	-	<b>0.842231</b>

Πίνακας 4.1.2.2: T-test p-values at significance level  $\alpha=0.05$  on trainset -100 iterations

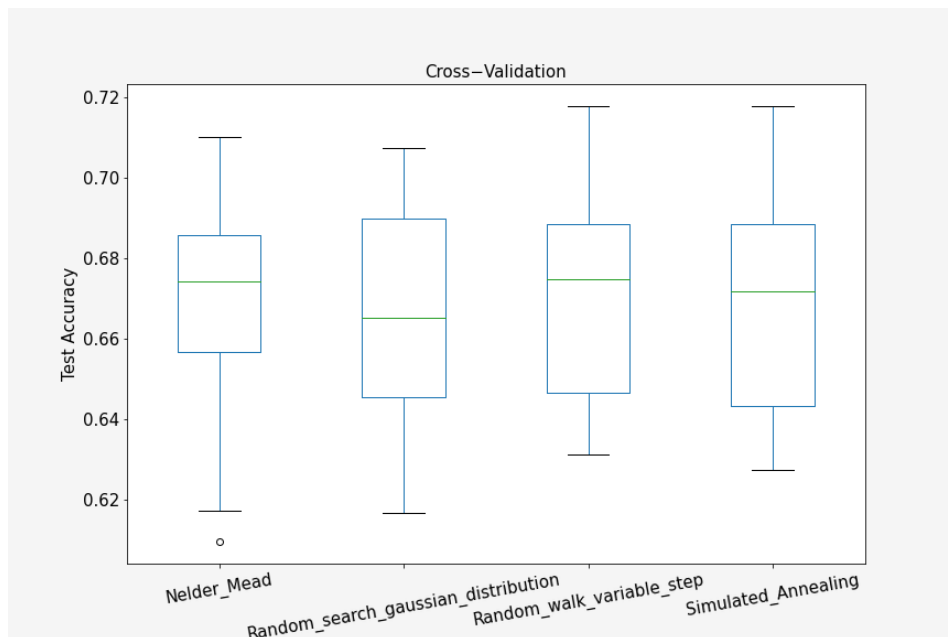
#### 4.1.3 Αποτελέσματα για 50 επαναλήψεις στο testset

Στη συγκεκριμένη υποενότητα παρουσιάζονται τα αποτελέσματα των πειραμάτων για 50 επαναλήψεις στο testset.

Όπως παρατηρούμε στον Πίνακα 4.1.3.1 η μέθοδος **RWVS** επικράτησε των υπόλοιπων αλγορίθμων για το σύνολο των στατιστικών μεγεθών. Αρκετά κοντά στα αποτελέσματα της **RWVS** βρίσκεται και η **NM** για τα μεγέθη **Mean**, **Median** και **St.Dev.**, ενώ η **SA** παρουσιάζει την ίδια **Max** τιμή με την **RWVS**.

	Mean	Median	St.Dev.	Min	Max
NM	<b>0.669303</b>	<b>0.674206</b>	<b>0.024669</b>	0.609656	0.710185
RSGD	0.664303	0.66541	0.027901	0.616931	0.70754
RWVS	<b>0.669599</b>	<b>0.674934</b>	<b>0.024401</b>	<b>0.631217</b>	<b>0.717857</b>
SA	0.668884	0.671825	0.024992	0.627646	<b>0.717857</b>

Πίνακας 4.1.3 1: Testset classification accuracy - Statistics -50 iterations



Διάγραμμα 4.1.3.1: Boxplot of classification accuracy on testset -50 iterations

	NM	RSGD	RWVS	SA
NM	-	<b>0.465099</b>	0.962967	0.948134
RSGD	-	-	<b>0.437104</b>	0.505597
RWVS	-	-	-	0.911204

Πίνακας 4.1.3.1: T-test p-values at significance level  $\alpha=0.05$  on testset -50 iterations

Όπως βλέπουμε και από τον στατιστικό έλεγχο υποθέσεων που πραγματοποιήθηκε και παρουσιάζεται στον Πίνακα 4.1.3.2 οι μέθοδοι δεν διαφέρουν στατιστικά σημαντικά μεταξύ τους. Η καλύτερη μέθοδος στην συγκεκριμένη περίπτωση είναι η **RWVS**.

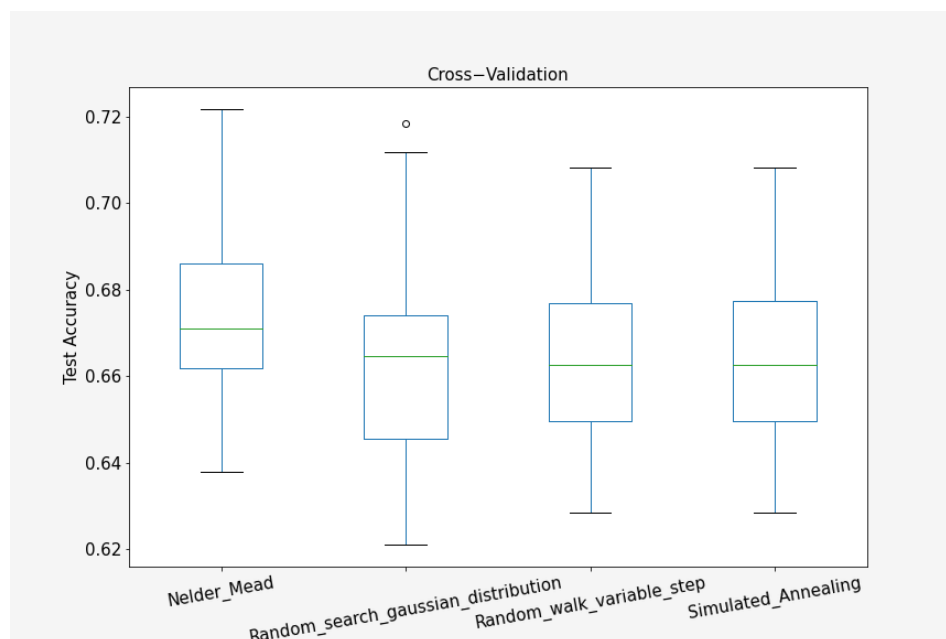
#### 4.1.4 Αποτελέσματα για 100 επαναλήψεις στο testset

Στη συγκεκριμένη υποενότητα παρουσιάζονται τα αποτελέσματα των πειραμάτων για 100 επαναλήψεις στο testset.

Όπως παρατηρούμε στον Πίνακα 4.1.4.1 η μέθοδος **NM** επικρατεί όλων των υπόλοιπων μεθόδων για όλα τα στατιστικά μεγέθη. Αρκετά κοντά στην τιμή για **St.Dev.** της **NM** είναι και η αντίστοιχη της **RWVS**.

	Mean	Median	St.Dev.	Min	Max
NM	<b>0.674497</b>	<b>0.671032</b>	<b>0.019859</b>	<b>0.637963</b>	<b>0.721693</b>
RSGD	0.664158	0.664749	0.023805	0.621032	0.718386
RWVS	0.663505	0.662632	<b>0.019877</b>	0.628439	0.708069
SA	0.663386	0.662632	0.020023	0.628439	0.708069

Πίνακας 4.1.4.1: Testset classification accuracy - Statistics -100 iterations



Διάγραμμα 4.1.4.1: Boxplot of classification accuracy on testset -100 iterations



Για τις 100 επαναλήψεις, παρατηρώντας τα στατιστικά μεγέθη καθώς και το boxplot είναι πλέον εμφανές ότι η μέθοδος **NM** είναι καλύτερη από τις υπόλοιπες. Προχωράμε σε στατιστικό έλεγχο υποθέσεων τα αποτελέσματα το οποίου καταγράφονται στον Πίνακα 4.1.4.2.

	NM	RSGD	RWVS	SA
NM	-	<b>0.07288</b>	<b>0.036341</b>	<b>0.035077</b>
RSGD	-	-	0.908646	0.892399
RWVS	-	-	-	0.981641

Πίνακας 4.1.4.2: T-test p-values at significance level  $\alpha=0.05$  on testset -100 iterations

Παρατηρείται ότι η **NM** διαφέρει από τις μεθόδους **RWVS** και **SA** στατιστικά σημαντικά. Οι υπόλοιπες τρεις μέθοδοι μεταξύ τους φαίνεται να μην έχουν μεγάλη διαφορά μεταξύ τους και θεωρούνται στατιστικά ισοδύναμες.

## 4.2 Mean Square Error Results

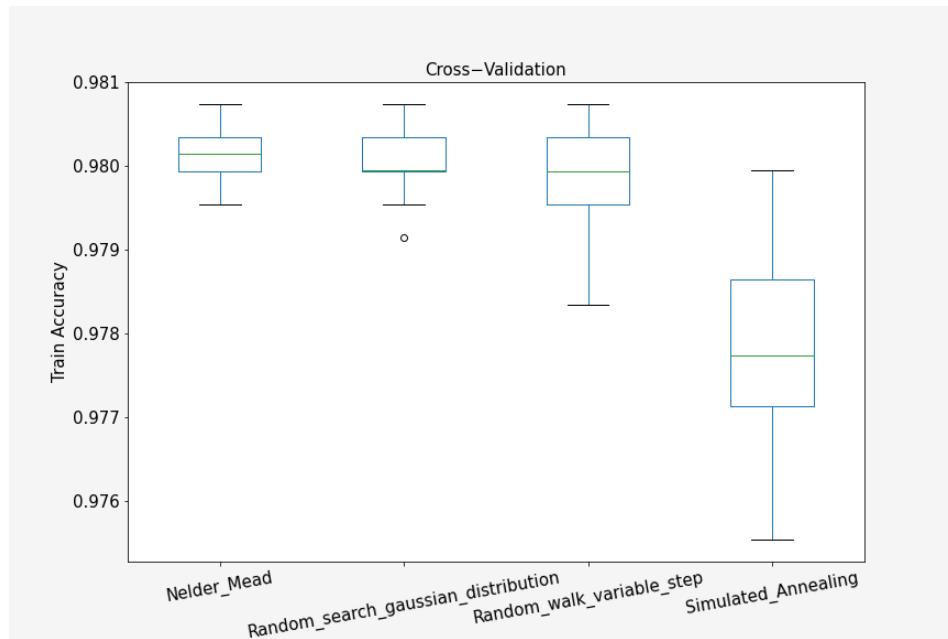
### 4.2.1 Αποτελέσματα για 50 επαναλήψεις στο trainset

Στη συγκεκριμένη υποενότητα παρουσιάζονται τα αποτελέσματα των πειραμάτων για 50 επαναλήψεις στο trainset.

Όπως παρατηρείται στον παρακάτω πίνακα (Πίνακας 4.2.1.1) η μέθοδος **NM** είναι η επικρατέστερη των υπολοίπων για τα στατιστικά μεγέθη **Mean**, **Median**, **St.Dev.** και **Min**, ενώ παρουσιάζει την ίδια **Max** τιμή με τις μεθόδους **RSGD** και **RWVS**. Επίσης στο συγκεκριμένο πείραμα η μέθοδος **RSGD** έχει δώσει παραπλήσια αποτελέσματα όσον αφορά τα στατιστικά μεγέθη **Mean**, **St.Dev.** και **Min**. Τα συμπεράσματα αυτά γίνονται αντιληπτά και στο αντίστοιχο boxplot (Διάγραμμα 4.2.1.1)

	Mean	Median	St.Dev.	Min	Max
NM	<b>0.980091</b>	<b>0.980144</b>	<b>0.000324</b>	<b>0.979539</b>	<b>0.980744</b>
RSGD	<b>0.980037</b>	0.979946	<b>0.000344</b>	0.979142	<b>0.980744</b>
RWVS	0.979837	0.979944	0.000537	0.978341	<b>0.980744</b>
SA	0.977777	0.977737	0.001103	0.975531	0.979945

Πίνακας 4.2.1.1: Trainset classification accuracy - Statistics -50 iterations



Διάγραμμα 4.2.1.1: Boxplot of classification accuracy on trainset -50 iterations

Όπως βλέπουμε από τον παρακάτω στατιστικό έλεγχο υποθέσεων οι **NM** και **RSGD** δεν διαφέρουν στατιστικά σημαντικά μεταξύ τους, ομοίως και οι **RSGD** με την **RWVS** αν και είναι αρκετά κοντά στο επίπεδο σημαντικότητας. Το υπόλοιπο των μεθόδων παρουσιάζει στατιστικά σημαντική διαφορά μεταξύ τους. Οι δύο καλύτερες μέθοδοι είναι οι **NM** και **RSGD**.

	NM	RSGD	RWVS	SA
NM	-	<b>0.538857</b>	<b>0.030342</b>	7.41E-16
RSGD	-	-	<b>0.089846</b>	2.25E-15
RWVS	-	-	-	6.27E-13

Πίνακας 4.2.1.2: T-test p-values at significance level  $\alpha=0.05$  on trainset -50 iterations

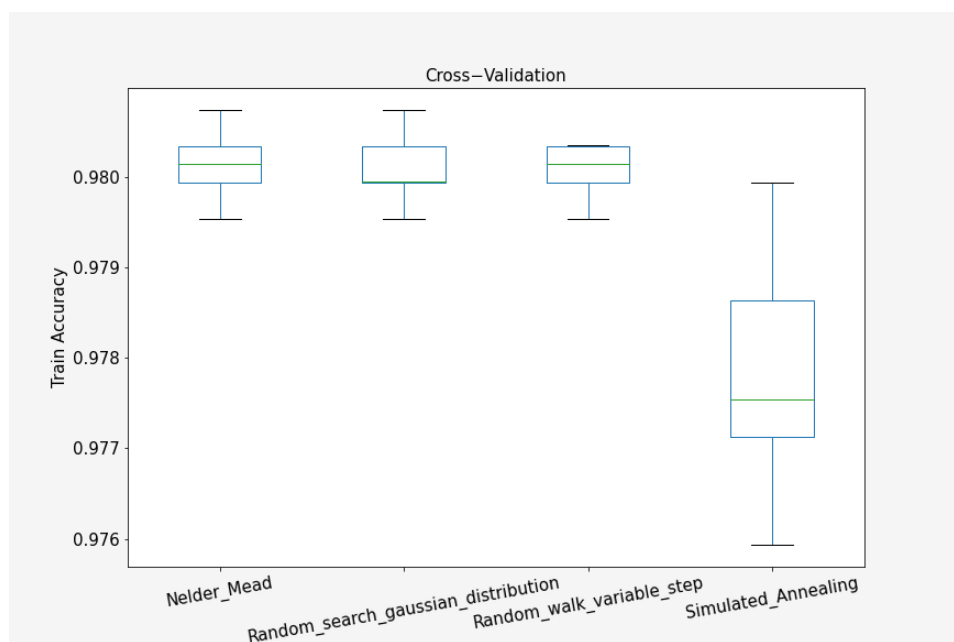
#### 4.2.2 Αποτελέσματα για 100 επαναλήψεις στο trainset

Στη συγκεκριμένη υποενότητα παρουσιάζονται τα αποτελέσματα των πειραμάτων για 100 επαναλήψεις στο trainset.

Όπως παρατηρείται στον Πίνακα 4.2.2.1 η μέθοδος η οποία δίνει τα καλύτερα συνολικά αποτελέσματα είναι η **NM** για το σύνολο των στατιστικών μεγεθών. Η **NM** φέρει τις ίδιες τιμές για τα στατιστικά μεγέθη **Median** και **Min** με την **RWVS** και τις ίδιες για **Min** και **Max** με την μέθοδο **RSGD**. Γενικά παρατηρείται ότι η **RWVS** ως επί τον πλείστον πλησιάζει οριακά την **NM**.

	Mean	Median	St.Dev.	Min	Max
NM	<b>0.980091</b>	<b>0.980144</b>	<b>0.000324</b>	<b>0.979539</b>	<b>0.980744</b>
RSGD	0.980077	0.979947	<b>0.000322</b>	<b>0.979539</b>	<b>0.980744</b>
RWVS	0.980064	<b>0.980144</b>	<b>0.000319</b>	<b>0.979539</b>	<b>0.980349</b>
SA	0.977764	0.977541	0.001088	0.975931	0.979942

Πίνακας 4.2.2.1: Trainset classification accuracy - Statistics -100 iterations



Διάγραμμα 4.2.2.1: Boxplot of classification accuracy on trainset -100 iterations

Όπως βλέπουμε από τον παρακάτω στατιστικό έλεγχο υποθέσεων οι **NM** με την **RSGD** και την **RWVS** δεν διαφέρουν στατιστικά σημαντικά μεταξύ τους, ομοίως και οι **RSGD** με την **RWVS**. Το υπόλοιπο των μεθόδων παρουσιάζει στατιστικά σημαντική διαφορά. Η καλύτερη μέθοδος είναι η **NM** και ακολουθούν οι δύο μέθοδοι **RSGD** και **RWVS**.

	NM	RSGD	RWVS	SA
NM	-	<b>0.873079</b>	<b>0.748791</b>	3.58E-16
RSGD	-	-	<b>0.872458</b>	4.39E-16
RWVS	-	-	-	5.37E-16

Πίνακας 4.2.2.2: T-test p-values at significance level  $\alpha=0.05$  on trainset -100 iterations

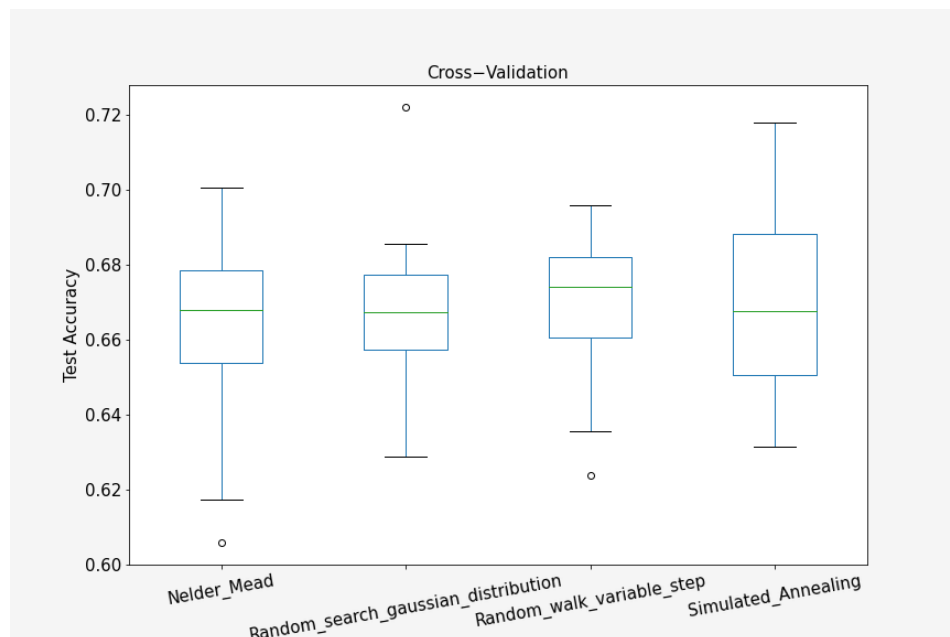
### 4.2.3 Αποτελέσματα για 50 επαναλήψεις στο testset

Στη συγκεκριμένη υποενότητα παρουσιάζονται τα αποτελέσματα των πειραμάτων για 50 επαναλήψεις στο testset.

Όπως παρατηρείται στον Πίνακα 4.2.3.1 τα αποτελέσματα είναι αμφιλεγόμενα καθότι καμία από τις μεθόδους δεν έχει απόλυτη κυριαρχία. Οι τιμές των **Mean** και **Median** είναι οι καλύτερες για την **RWVS** μέθοδο, ενώ για τα στατιστικά μεγέθη **St.Dev.** και **Max** καλύτερη εμφανίζεται η **RSGD** και τέλος η **SA** έχει καλύτερη **Min** τιμή.

	Mean	Median	St.Dev.	Min	Max
NM	0.66489	0.667857	0.021002	0.605688	0.700661
RSGD	0.667319	0.667262	<b>0.016729</b>	0.628704	<b>0.722222</b>
RWVS	<b>0.669506</b>	<b>0.67414</b>	0.018793	0.62381	0.696032
SA	0.668779	0.667593	0.023854	<b>0.631349</b>	0.717989

Πίνακας 4.2.3.1: Testset Classification accuracy - Statistics -50 iterations



Διάγραμμα 4.2.3.1: Boxplot of classification accuracy on testset -50 iterations

Παρατηρώντας τα παρακάτω αποτελέσματα, βλέπουμε ότι οι μέθοδοι δε διαφέρουν στατιστικά σημαντικά μεταξύ τους. Η μέθοδος με την καλύτερη απόδοση είναι η **RWVS** καθώς κατάφερε την καλύτερη τιμή σε **Mean** και **Median**.

	NM	RSGD	RWVS	SA
NM	-	0.622061	<b>0.373322</b>	0.50539
RSGD	-	-	0.635803	0.784784
RWVS	-	-	-	0.896057

Πίνακας 4.2.3.2: T-test p-values at significance level  $\alpha=0.05$  on testset -50 iterations

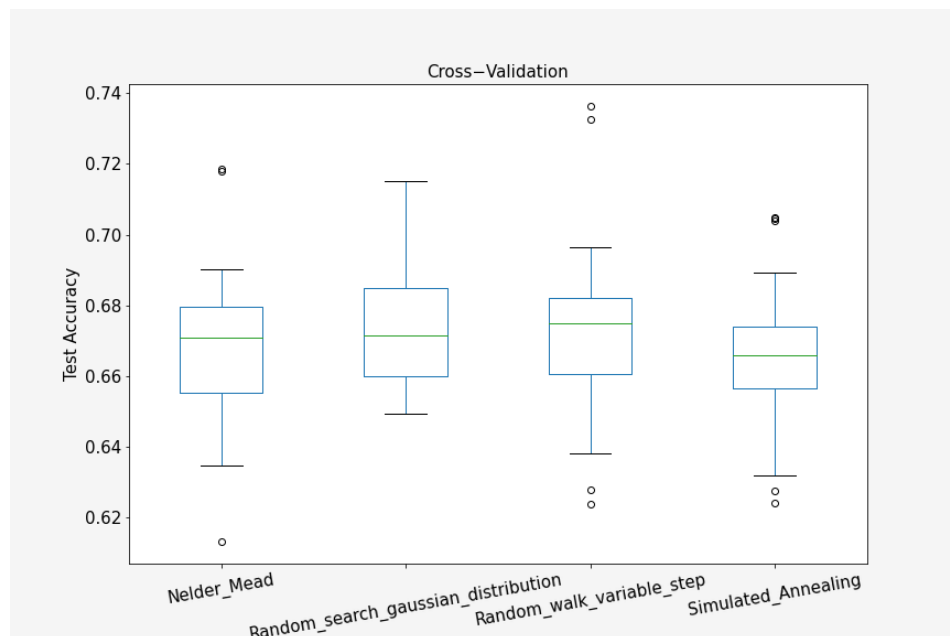
#### 4.2.4 Αποτελέσματα για 100 επαναλήψεις στο testset

Στη συγκεκριμένη υποενότητα παρουσιάζονται τα αποτελέσματα των πειραμάτων για 100 επαναλήψεις στο testset.

Για το συγκεκριμένο πείραμα, όπως χαρακτηριστικά φαίνεται στον Πίνακα 4.2.4.1, καμία μέθοδος δεν παρουσιάζει απόλυτη κυριαρχία έναντι κάποιας άλλης αλλά οι βέλτιστες τιμές είναι διαμοιραζόμενες ανάμεσα στις **RSGD** και **RWVS** μεθόδους.

	Mean	Median	St.Dev.	Min	Max
NM	0.667857	0.671032	0.022734	0.613228	0.718519
RSGD	<b>0.673929</b>	0.671429	<b>0.01763</b>	<b>0.649339</b>	0.715212
RWVS	0.673034	<b>0.675000</b>	0.02456	0.623942	<b>0.736376</b>
SA	0.665675	0.665939	0.019735	0.624339	0.704894

Πίνακας 4.2.4.1: Testset classification accuracy - Statistics -100 iterations



Διάγραμμα 4.2.4.1: Boxplot of classification accuracy on testset -100 iterations

Όπως παρατηρούμε και στο τελευταίο σκετ πειραμάτων, τα αποτελέσματα του οποίου αναγράφονται στον Πίνακα 4.2.3.2, σύμφωνα με τον στατιστικό έλεγχο υποθέσεων οι μέθοδοι δεν διαφέρουν στατιστικά σημαντικά μεταξύ τους. Οι δύο μέθοδοι με τις καλύτερες αποδόσεις είναι οι **RSGD** και **RWVS** καθώς η πρώτη κατάφερε καλύτερο **Mean** και η δεύτερη καλύτερο **Median**.

	NM	RSGD	RWVS	SA
NM	-	0.252454	0.400384	0.692758
RSGD	-	-	0.871746	<b>0.09291</b>
RWVS	-	-	-	0.205885

Πίνακας 4.2.4.2: T-test p-values at significance level  $\alpha=0.05$  on testset -100 iterations

## 5. Συγκριτική ανάλυση αποτελεσμάτων και αξιολόγηση

Σκοπός της παρούσας εργασίας είναι η εκπαίδευση ενός Πιθανοτικού Νευρωνικού Δικτύου όπως αυτό έχει παρουσιαστεί στην εργασία των V. L. GEORGIU et al. [1] για ένα dataset το οποίο περιέχει καταγεγραμμένα **δεδομένα καρκίνου του μαστού**. Για τους σκοπούς διεκπεραίωσης της παρούσας εργασίας χρησιμοποιήθηκε η **10-Fold Cross Validation μέθοδος διαχωρισμού** των δεδομένων και υλοποιήθηκαν 4 αλγόριθμοι βελτιστοποίησης. Ο 1<sup>ος</sup> αλγόριθμος (**Nelder - Mead**) αποτελεί έναν αλγόριθμο άμεσης αναζήτησης ενώ οι υπόλοιποι 3 (**Random Search with Gaussian Distribution, Random Walk with Variable Step, Simulated annealing**) είναι στοχαστικοί αλγόριθμοι. Και οι τέσσερις μεθοδολογίες ανήκουν στις μεθόδους βελτιστοποίησης χωρίς παραγώγους.

Δοκιμάστηκαν δύο συναρτήσεις σφάλματος οι οποίες είναι οι: **mean squared error** και **classification error**.

Όπως παρατηρείται στον Πίνακα 5.1 ο οποίος παρουσιάζει τα συγκεντρωτικά αποτελέσματα των mean classification accuracies των τεσσάρων προς εξέταση μεθόδων για το trainset, οι δύο καλύτερες μέθοδοι οι οποίες ελαχιστοποιούν το classification error φαίνεται να είναι η **NM** και ακολουθεί η **RSGD**. Το ίδιο ισχύει για το Mean Squared Error.

CLASSIFICATION ERROR			MEAN SQUARED ERROR	
	Mean Value		Mean Value	
	$k_{\max} = 50$	$k_{\max} = 100$	$k_{\max} = 50$	$k_{\max} = 100$
NM	0.980091	0.980091	0.980091	0.980091
RSGD	0.979997	0.980024	0.980037	0.980077
RWVS	0.977871	0.977911	0.979837	0.980064
SA	0.977818	0.977858	0.977777	0.977764

Πίνακας 5.1: Trainset mean classification accuracy

Από τον Πίνακα 5.2 ο οποίος παρουσιάζει τα συγκεντρωτικά αποτελέσματα των mean classification accuracies των τεσσάρων προς εξέταση μεθόδων για το testset. Η καλύτερη μέθοδος η οποία ελαχιστοποιεί το classification error φαίνεται να είναι η **NM** και η καλύτερη μέθοδος που ελαχιστοποιεί το mean squared error είναι η **RWVS**.

CLASSIFICATION ERROR			MEAN SQUARED ERROR	
	Mean Value		Mean Value	
	$k_{\max} = 50$	$k_{\max} = 100$	$k_{\max} = 50$	$k_{\max} = 100$
NM	0.669303	0.674497	0.664890	0.667857
RSGD	0.664303	0.664158	0.667319	0.673929
RWVS	0.669599	0.663505	0.669506	0.673034
SA	0.668884	0.663386	0.668779	0.665675

Πίνακας 5.2: Testset mean classification accuracy

## ΓΕΝΙΚΗ ΠΑΡΑΤΗΡΗΣΗ

Ενώ οι μέθοδοι **NM** και **RS GD** στις περισσότερες περιπτώσεις βελτιώνουν το training accuracy έναντι των υπολοίπων, παρατηρούμε ότι στο test accuracy δεν έχουν απαραίτητα τα καλύτερα αποτελέσματα. Αυτό συμβαίνει καθώς οι δύο μέθοδοι μειώνουν υπερβολικά πολύ το σφάλμα εκπαίδευσης (σχεδόν στο μηδέν ή και ακριβώς μηδέν) και προκύπτει το φαινόμενο της **υπερ-εκπαίδευσης του νευρωνικού δικτύου (overfitting)**. Αυτό έχει ως αποτέλεσμα το νευρωνικό δίκτυο να χάνει την ικανότητα γενίκευσης του.

## **Η ΕΝΔΕΙΚΤΙΚΗ ΣΥΓΚΡΙΣΗ ΜΕ ΤΑ ΔΗΜΟΣΙΕΥΜΕΝΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΕΝ ΠΡΑΓΜΑΤΟΠΟΙΗΘΗΚΕ ΚΑΘΟΤΙ ΑΥΤΟ ΔΕΝ ΕΙΝΑΙ ΕΦΙΚΤΟ ΛΟΓΩ ΔΙΑΦΟΡΕΤΙΚΟΥ ΔΟΣΜΕΝΟΥ DATASET.**

## 6. Δομή αναπτυσσόμενου κώδικα

Στην παρούσα ενότητα περιγράφεται η λειτουργία του κώδικα που αναπτύχθηκε για την τρέχουσα εργασία. Ακολουθεί μια σύντομη περιγραφή των κλάσεων, μεθόδων και συναρτήσεων που χρησιμοποιήθηκαν.

**main.py:** Το πρόγραμμα εκκινεί μέσα από τη συνάρτηση `main`. Στο αρχείο `seeds.csv` υπάρχουν 30 αριθμοί (seeds) που τα χρησιμοποιούμε για την αρχικοποίηση της γεννήτριας τυχαίων αριθμών. Για κάθε πείραμα χρησιμοποιήσαμε 10 fold cross validation για τον διαχωρισμό του dataset σε train set και test set. Για κάθε fold καλούμε τις 4 μεθόδους βελτιστοποίησης μέσω της κλάσης `Optimizer`, όπου η κάθε μέθοδος εκτελείται για πλήθος επαναλήψεων `max_iterations` ( $k_{max}$ ) και επιστρέφει τις βέλτιστες παραμέτρους. Για τις βέλτιστες παραμέτρους αυτές ελέγχουμε το classification accuracy στο test set και train set, για την κάθε μια από τις 4 μεθόδους. Αφού ολοκληρωθούν τα 10 folds, για κάθε μέθοδο έχει αποθηκευτεί η μέση τιμή του accuracy για test και train set. Τέλος, τα αποτελέσματα αυτά αποθηκεύονται σε excel αρχεία.

**class Optimizer:** Στην κλάση `Optimizer` υλοποιούνται οι 4 μέθοδοι βελτιστοποίησης, οι οποίες είναι οι εξής:

- `random_search_gaussian_distribution(self, model_parameters):` Η συνάρτηση εκτελεί τη μέθοδο τυχαίας αναζήτησης με κανονική κατανομή και δέχεται σαν `model_parameters` τις παραμέτρους του μοντέλου (ένα αρχικό τυχαίο διάνυσμα). Η συνάρτηση επιστρέφει τις `optimized` παραμέτρους για το μοντέλο καθώς και την αντίστοιχη συναρτησιακή τιμή.
- `random_walk_variable_step(self, model_parameters, s, a, beta):` Η συνάρτηση εκτελεί τη μέθοδο τυχαίας αναζήτησης με μεταβλητό βήμα και δέχεται σαν `model_parameters` τις παραμέτρους του μοντέλου (ένα αρχικό τυχαίο διάνυσμα) και επιπλέον τις παραμέτρους `s, a, beta` όπου `s` το μέγεθος του βήματος, `a` ο συντελεστής αύξησης του βήματος και `beta` ο συντελεστής μείωσης του βήματος. Η συνάρτηση επιστρέφει τις `optimized` παραμέτρους για το μοντέλο καθώς και την αντίστοιχη συναρτησιακή τιμή.



- *simulated\_annealing(self, model\_parameters, t, a, b, r)*: Η συνάρτηση εκτελεί τη μέθοδο προσομοιωμένης ανόπτησης και δέχεται σαν *model\_parameters* τις παραμέτρους του μοντέλου (ένα αρχικό τυχαίο διάνυσμα) και επιπλέον τις παραμέτρους *t, a, b, r*, όπου *t* παράμετρος θερμοκρασίας, *a* και *b* σταθερές απαραίτητες για την συνάρτηση ψύξης που χρησιμοποιήθηκε και *r* μέγεθος βήματος. Η συνάρτηση επιστρέφει τις *optimized* παραμέτρους για το μοντέλο καθώς και την αντίστοιχη συναρτησιακή τιμή.
- *Nelder\_Mead(self, model\_parameters)*: Η συνάρτηση εκτελεί τη μέθοδο άμεσης αναζήτησης Nelder Mead και δέχεται σαν *model\_parameters* τις παραμέτρους του μοντέλου (ένα αρχικό τυχαίο διάνυσμα). Η συνάρτηση επιστρέφει τις *optimized* παραμέτρους για το μοντέλο καθώς και την αντίστοιχη συναρτησιακή τιμή.

**Κλάση Neuron**: Στην κλάση Neuron υπάρχουν δύο συναρτήσεις. Η συνάρτηση *\_\_init\_\_(self, x)* η οποία αρχικοποιεί το κέντρο της Gaussian ενεργοποίησης του νευρώνα και η *gaussian\_activation*.

- *gaussian\_activation(self, x, det\_spr\_par\_mat, inv\_spr\_par\_mat)*: Η συνάρτηση εκτελεί τη Gaussian ενεργοποίηση του νευρώνα και δέχεται σαν παραμέτρους τα εξής: *x* (το δεδομένο), *det\_spr\_par\_mat* (η ορίζουσα του πίνακα  $\Sigma$ ) και *inv\_model\_parameters* (ο αντίστροφος του πίνακα  $\Sigma$  -matrix of spread-smoothing parameters-). Η συνάρτηση επιστρέφει την Gaussian ενεργοποίηση για το συγκεκριμένο δεδομένο.

**Κλάση Neural\_Network**: Στην κλάση Neural\_Network υπάρχουν 4 συναρτήσεις. Οι συναρτήσεις που υπάρχουν εκεί είναι:

- *\_\_init\_\_(self, data, labels)* η οποία δέχεται τα δεδομένα με τις αντίστοιχες κατηγορίες και αρχικοποιεί δύο λίστες νευρώνων, μία για την κλάση 1 και για την κλάση 2.
- *\_\_init\_neurons(self)* στην οποία αρχικοποιούμε τους νευρώνες και τα κέντρα τους και τα τοποθετούμε στην αντίστοιχη λίστα νευρώνων ανάλογα με την κατηγορία στην οποία ανήκουν (1 ή 2).
- *\_\_forward\_pass(self, model\_parameters, data)*: Η συνάρτηση δέχεται τις παραμέτρους του δικτύου και ένα σύνολο δεδομένων, υπολογίζει την απόκριση του δικτύου για κάθε ένα δεδομένο και επιστρέφει ένα διάνυσμα με τις αποκρίσεις του δικτύου. Σε περίπτωση που χρησιμοποιούμε το classification error επιστρέφει απευθείας την κατηγορία του δεδομένου ενώ στην περίπτωση του mean square error επιστρέφει κατανομή πιθανότητας.
- *error\_function(self, model\_parameters)*: Η συνάρτηση δέχεται τις παραμέτρους του μοντέλου, υπολογίζει την τιμή της συνάρτησης σφάλματος και την επιστρέφει (mean squared error ή classification error ανάλογα με το ποιο από τα δύο αρχεία εκτελείται)
- *predict(self, model\_parameters, data)*: Η συνάρτηση δέχεται τα δεδομένα και τις παραμέτρους του μοντέλου και επιστρέφει ένα διάνυσμα με τις προβλέψεις του νευρωνικού δικτύου.

**Επιπροσθέτως για την προεπεξεργασία των δεδομένων υλοποιήθηκε η παρακάτω συνάρτηση:**

*preprocess\_data(data\_path)*: Η συνάρτηση δέχεται το path που βρίσκονται τα data και κάνει όλη την προεπεξεργασία των δεδομένων. Συγκεκριμένα, διαβάζει τα δεδομένα, πετάει εκτός του dataset τις εγγραφές που έχουν missing values, χωρίζει τα δεδομένα από τις κατηγορίες τους και τέλος τα μετασχηματίζει αφαιρώντας από το κάθε feature τη μέση τιμή του και διαιρώντας με την τυπική

απόκλιση του, ώστε τα features των δεδομένων να γίνουν ίδιας κλίμακας. Η συνάρτηση επιστρέφει τα δεδομένα και τις κατηγορίες τους.

### **ΟΔΗΓΙΕΣ ΓΙΑ ΤΗΝ ΟΡΘΗ ΕΚΤΕΛΕΣΗ ΤΟΥ ΚΩΔΙΚΑ**

Η υλοποίηση των μεθόδων έγινε σε γλώσσα Python 3.8.

Οι βιβλιοθήκες που έχουν χρησιμοποιηθεί είναι οι numpy, pandas, matplotlib, math, sys, time, sklearn και scipy.

Για τους μαθηματικούς υπολογισμούς χρησιμοποιήθηκαν οι math και numpy.

Για το διάβασμα και προεπεξεργασία των δεδομένων χρησιμοποιήθηκε η βιβλιοθήκη pandas.

Επίσης, τα modules που χρησιμοποιήθηκαν είναι τα time για χρονομέτρηση και sys για να διαβάζουμε παραμέτρους από το terminal.

Η βιβλιοθήκη scipy χρησιμοποιήθηκε για την πραγματοποίηση των t-tests.

Με την sklearn έγινε ο μετασχηματισμός δεδομένων, το cross validation και ο υπολογισμός του accuracy των μεθόδων.

Υπάρχουν δύο εκτελέσιμα αρχεία (1) optimization\_classification\_error.py & (2) optimization\_mse.py

Η γενική μορφή της εντολής εκτέλεσης του προγράμματος είναι η εξής:

`python3 εκτελέσιμο_αρχείο seed_index_low seed_index_high`

Π.χ. για να τρέξουν τα πρώτα 3 πειράματα χρησιμοποιώντας το optimization\_classification\_error.py γράφω:

**`python3 optimization_classification_error.py 0 3`**

- Μέσα στη συνάρτηση Main η μεταβλητή **max\_iterations** ορίζει το πλήθος των συναρτησιακών υπολογισμών ( $k_{\max}$ ).

### **ΠΑΡΑΤΗΡΗΣΗ**

Δύο ιδιότητες γραμμικής άλγεβρας βοήθησαν στην βελτίωση της ταχύτητας του κώδικά μας (24 φορές πιο γρήγορη απόδοση) και οι οποίες είναι:

- i) Η ορίζουσα ενός διαγώνιου πίνακα ισούται με το γινόμενο των στοιχείων της διαγώνιου, συνεπώς δεν υπολογίσαμε αναλυτικά την ορίζουσα του  $\Sigma$ .
- ii) Ο αντίστροφος ενός διαγώνιου πίνακα είναι ο ίδιος ο πίνακας με αντεστραμμένα τα στοιχεία της διαγώνιου, συνεπώς δεν υπολογίσαμε τον αντίστροφο  $\Sigma$  αναλυτικά.

## 7. References

- [1]. V. L. Georgiou, N. G. Pavlidis, K. E. Parsopoulos, PH. D. Alevizos and M. N. Vrahatis, New self-adaptive probabilistic neural networks in bioinformatic and medical tasks, International Journal on Artificial Intelligence Tools Vol. 15, No. 3 (2006) 371–396.
- [2]. Rachel Lea Ballantyne Draelos , GLASS BOX, Machine Learning and Medicine, Best Use of Train/Val/Test Splits, with Tips for Medical Data, <https://glassboxmedicine.com/2019/09/15/best-use-of-train-val-test-splits-with-tips-for-medical-data/>