



**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ  
ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Εκπαίδευση τεχνητού νευρωνικού δικτύου με χρήση  
εξελικτικών αλγορίθμων**



Του φοιτητή

Γιώργου Νάσκου

Αρ. Μητρώου: 144374

Επιβλέπων καθηγητής

Κωνσταντίνος Γουλιάνας

Θεσσαλονίκη 2020

Τίτλος Δ.Ε: Εκπαίδευση τεχνητού νευρωνικού δικτύου με χρήση γενετικών αλγορίθμων

Κωδικός Δ.Ε: 19037

Όνοματεπώνυμο φοιτητή: Γιώργος Νάσκος

Όνοματεπώνυμο εισηγητή: Κωνσταντίνος Γουλιάνας

Ημερομηνία ανάληψης Δ.Ε: 13-05-2019

Ημερομηνία περάτωσης: 15-06-2020

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε*

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Γιώργου Νάσκου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

## ΠΡΟΛΟΓΟΣ

Όλοι έχουμε σαν σκέψη ότι οι υπολογιστές είναι αυστηρά προγραμματισμένοι να εκτελούν συγκεκριμένες οδηγίες χωρίς να έχουν την δυνατότητα να παρεκκλίνουν από αυτές. Αυτό είναι εν μέρη αλήθεια. Γιατί οι υπολογιστές έχουν την δυνατότητα να επεξεργαστούν δεδομένα, να παράγουν δικά τους συμπεράσματα και με βάση αυτά να πάρουν τις ανάλογες αποφάσεις. Πάντα με την δική μας καθοδήγηση. Αυτός ο κλάδος της πληροφορικής ονομάζεται τεχνητή νοημοσύνη (artificial intelligence). Ένα κομμάτι της μας δείχνει, πως μπορούμε να κάνουμε τον υπολογιστή να μάθει από ένα σύνολο δεδομένων ώστε να επιλύσει από μόνος του ένα πρόβλημα. Το κομμάτι αυτό ονομάζεται μηχανική μάθηση (machine learning). Όπως σε πολλές επιστήμες έτσι και εδώ κρύβονται από πίσω τα μαθηματικά. Πολλοί από τους μαθηματικούς τύπους και τις μεθοδολογίες που χρησιμοποιούνται στην μηχανική μάθηση έχουν υλοποιηθεί πριν ακόμη υπάρξουν οι υπολογιστές που ξέρουμε σήμερα. Ο άνθρωπος έχει την τάση να παρατηρεί την φύση και να μαθαίνει από αυτήν, έτσι έκανε και με την μηχανική μάθηση, όπου η βασικότερη δομή που χρησιμοποιείται είναι τα νευρωνικά δίκτυα, παρμένα από τους νευρώνες του εγκεφάλου μας. Για πολλούς όμως τα μαθηματικά δεν είναι ένας εύκολος τομέας, εδώ ήρθαν να βοηθήσουν οι εξελικτικοί αλγόριθμοι (evolutionary algorithms), με την σειρά τους εμπνευσμένοι από την θεωρία της εξέλιξης, μπορούν χωρίς την χρήση περίπλοκων μαθηματικών να επιλύσουν προβλήματα βελτιστοποίησης. Ένα πρόβλημα βελτιστοποίησης είναι και η εκπαίδευση του νευρωνικού δικτύου, έτσι δημιουργήθηκε ο συνδυασμός των δύο που ονομάζεται νευροεξελικτικοί αλγόριθμοι (neuroevolution algorithms). Σε αυτήν την πτυχιακή εργασία θα δούμε την κατασκευή ενός neuroevolution συστήματος, καθώς και την χρήση του σε προβλήματα χρονοσειρών, συγκρίνοντας το ταυτόχρονα με ένα ήδη υπάρχον σύστημα.

Σελίδες από την διπλωματική εργασία έχουν δημοσιευτεί στο επιστημονικό περιοδικό Journal of Applied Mathematics and Physics:

G. Naskos, G. Konstantinos και M. Athanasios, «Neuroevolution Strategy for Time Series Prediction,» Journal of Applied Mathematics and Physics, vol. 8, no. 6, pp. 1047-1065, 2020.

<https://doi.org/10.4236/jamp.2020.86082>

## ΠΕΡΙΛΗΨΗ

Η βελτιστοποίηση είναι μια έννοια, μια διαδικασία, μια μέθοδος την οποία όλοι οι άνθρωποι χρησιμοποιούν. Από την επιλογή της συντομότερης διαδρομής για τη μετάβαση στον προορισμό μας, τον τρόπο διαχείρισης και διανομής υλικών που έχουμε στην διάθεσή μας, μέχρι την επίλυση πολύπλοκων προβλημάτων. Η πηγή έμπνευσης πάρα πολλών μεθόδων βελτιστοποίησης για πολλούς επιστήμονες υπήρξε η ίδια η φύση καθώς και οι μηχανισμοί που υπάρχουν σε αυτήν. Τα νευρωνικά δίκτυα, εμπνευσμένα από τους νευρώνες του ανθρώπινου εγκεφάλου, έχουν κερδίσει πολύ έδαφος τα τελευταία χρόνια και δίνουν λύση σε καθημερινά προβλήματα. Οι εξελικτικοί αλγόριθμοι φημίζονται για την αποδοτικότητα και την ταχύτητα τους, σε προβλήματα όπου η βέλτιστη λύση βρίσκεται σε ένα τεράστιο πλήθος πιθανών λύσεων καθώς και για την απλότητα τους, γιατί δεν είναι αναγκαία η χρήση περίπλοκων μαθηματικών για την υλοποίηση τους. Ο συνδυασμός αυτών των δύο ονομάζεται neuroevolution. Ο συνδυασμός αυτός δείχνει να δίνει καλύτερες λύσεις στα προβλήματα, καθώς εμπεριέχει τα θετικά και από τους δυο κόσμους. Στη παρούσα πτυχιακή έχει προταθεί μια αρχιτεκτονική και ένας αλγόριθμος για την επίλυση προβλημάτων χρονοσειρών με αρκετά μεγάλη επιτυχία.

## ABSTRACT

Optimization is a concept, a process, a method that all people use. From choosing the shortest route to moving to our destination, how to manage and distribute the materials at our disposal, to solving complex problems. The source of many optimization methods for many scientists has been the nature itself and the mechanisms that exist in it. Neural networks, inspired by the neurons of the human brain, have gained a lot of ground in recent years and provide solutions to everyday problems. Evolutionary algorithms are known for their performance and speed, in problems where the optimal solution lies in a huge number of possible solutions, and for their simplicity, because it is not necessary to use complex mathematics to implement them. The combination of these two is called neuroevolution. This combination seems to provide better solutions to the problems, as it contains the positives of both worlds. In this thesis, an architecture and an algorithm have been proposed to solve time problems with great success.

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Θα ήθελα να ευχαριστήσω θερμά την οικογένεια μου, γιατί και αυτοί με την σειρά τους είναι κομμάτι αυτής της διπλωματικής εργασίας.

Επίσης θα ήθελα να ευχαριστήσω τον Κ. Κωνσταντίνο Γουλιάνο, για την καθοδήγηση και τις συμβουλές του, καθώς και για την αξιοπρέπεια και επαγγελματικότητα του.

## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	3
ΠΕΡΙΛΗΨΗ.....	4
ABSTRACT.....	5
ΕΥΧΑΡΙΣΤΙΕΣ.....	6
ΠΕΡΙΕΧΟΜΕΝΑ.....	7
Ευρετήριο σχημάτων.....	9
Ευρετήριο πινάκων.....	10
Συντομογραφίες.....	11
ΕΙΣΑΓΩΓΗ.....	12
ΚΕΦΑΛΑΙΟ 1 – Μηχανική μάθηση και νευρωνικά δίκτυα.....	13
1.1 Μηχανική μάθηση.....	14
1.2 Τεχνητά νευρωνικά δίκτυα.....	15
1.3 Multilayer perceptron.....	17
1.3 Backpropagation.....	18
ΚΕΦΑΛΑΙΟ 2 – Εξελικτικοί αλγόριθμοι.....	20
2.1 Βασικές ιδιότητες και χαρακτηριστικά.....	20
ΚΕΦΑΛΑΙΟ 3 – Neuroevolution.....	23
3.1 Συστήματα & Αρχιτεκτονικές.....	23
3.1.1 Η μέθοδος Direct.....	24
3.1.2 Η μέθοδος Developmental.....	24
3.1.3 Η μέθοδος Implicit.....	25
3.2 Το σύστημα NEAT.....	25
ΚΕΦΑΛΑΙΟ 4 – Προτεινόμενη αρχιτεκτονική για επίλυση προβλημάτων χρονοσειρών.....	28
4.1 Το σύστημα μας.....	28
4.1.1 Κατασκευή του συστήματος.....	29
4.1.2 Οι μέθοδοι Hybrid και Migration.....	33
4.1.3 Μεταβλητό πλήθος γονιδίων.....	34
4.2 Overtraining.....	35
ΚΕΦΑΛΑΙΟ 5 – Περιβάλλον και εργαλεία ανάπτυξης.....	37
5.1 Επιλογή γλώσσας προγραμματισμού.....	37
5.2 Περιβάλλον ανάπτυξης.....	39
5.3 Γραφικό περιβάλλον.....	41

ΚΕΦΑΛΑΙΟ – 6 Datasets και πειράματα.....	43
6.1 Ανάλυση των datasets.....	43
6.2 Κατάστροση των πειραμάτων.....	44
6.3 Παράμετροι των πειραμάτων.....	46
ΚΕΦΑΛΑΙΟ – 7 Εκτέλεση πειραμάτων και αποτελέσματα.....	49
7.1 Αποτελέσματα.....	49
7.2 Επιπλέον ανάλυση.....	55
7.3 Τελικά συμπεράσματα.....	57
7.4 Προτεινόμενη μελλοντική μελέτη.....	57
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	58

## Ευρετήριο σχημάτων

Σχήμα 1 Μοντέλο τεχνητού νευρώνα.....	16
Σχήμα 2 Μοντέλο νευρωνικού δικτύου multilayer perceptron .....	18
Σχήμα 3 Ψευδοκώδικας Εξελικτικού Αλγορίθμου.....	21
Σχήμα 4 Παράδειγμα αντίστροφων δικτύων. Παρατηρούμε ότι ο γενετικός κώδικας (a) είναι πολύ διαφορετικός ανάμεσα στα δύο δίκτυα, αλλά τα δίκτυα (b) έχουν ίδια απόδοση.....	26
Σχήμα 5 Παράδειγμα κωδικοποίησης νευρωνικού δικτύου σε μονοδιάστατο πίνακα .....	30
Σχήμα 6 Σύστημα με χρήση των μεθόδων migration και hybrid .....	33
Σχήμα 7 Στο διάγραμμα φαίνεται, ότι το validation fitness μειώνεται με το πέρασμα των εποχών. Αυτό δείχνει τη συνεχή βελτίωση του δικτύου, χωρίς να χάνει την γενίκευση του.....	36
Σχήμα 8 Το όνομα της γλώσσας πάρθηκε από το όνομα της καφετέριας που συναντιόνταν οι δημιουργοί της. Εξού και το logo της.....	38
Σχήμα 9 Στιγμιότυπο από το γραφικό περιβάλλον του Apache Netbeans, με το project tree αριστερά και τις βάσεις και άλλα εργαλεία δεξιά .....	40
Σχήμα 10 Το γραφικό περιβάλλον του συστήματος. Ο πίνακας με τα αποτελέσματα και τα διαγράμματα.....	41
Σχήμα 11 Διαγράμματα από την εκτέλεση του συστήματος στο Mackey dataset .....	50
Σχήμα 12 Διαγράμματα από την εκτέλεση του συστήματος στο Lorenz dataset.....	51
Σχήμα 13 Όλα τα τελικά αποτελέσματα από τα test για όλα τα dataset. Παρατηρούμε ότι σε τρία από τα πέντε dataset το test C είναι το καλύτερο. ....	56

## Ευρετήριο πινάκων

Πίνακας 1 Κοινοί παράμετροι, για όλους τους τύπους πειραμάτων .....	47
Πίνακας 2 Μεταβλητές παράμετροι .....	48
Πίνακας 3 Οι παράμετροι του επιπλέον πειράματος D.....	48
Πίνακας 4 Τα τελικά αποτελέσματα από τις προβλέψεις, από όλα τα dataset. Σύγκριση με τις εγγραφές Sequential, Concurrent, MCNE που είναι τα αποτελέσματα της έρευνας [8], για τα αντίστοιχα datasets .....	53

## Συντομογραφίες

NE	Neuroevolution
EA	Εξελικτικοί αλγόριθμοι
ML	Machine learning (Μηχανική μάθηση)
MLP	Multilayer perceptron
BP	Backpropagation
AI	Artificial intelligence (Τεχνητή νοημοσύνη)
IDE	Integrated development environment
NEAT	Neuroevolution of augmenting topologies

## ΕΙΣΑΓΩΓΗ

Οι μεγάλες δυνατότητες των νευρωνικών δικτύων, έχουν αποδειχθεί επανειλημμένα στο παρελθόν από πολλές έρευνες, καθώς βρίσκουν εφαρμογή σε πραγματικά προβλήματα ακόμη και στην καθημερινή μας ζωή [1]. Οι εξελικτικοί αλγόριθμοι (EA), χρησιμοποιούνται σε προβλήματα βελτιστοποίησης. Έχοντας την δυνατότητα να ανταπεξέλθουν καλύτερα στα προβλήματα όπου η βέλτιστη λύση βρίσκεται ανάμεσα σε ένα хаοτικό πλήθος λύσεων [2]. Γενικότερα όσο πιο μεγάλο το εύρος πιθανών λύσεων ενός προβλήματος, τόσο πιο αποδοτικοί είναι οι EA σε αυτό. Από τον συνδυασμό των δύο προκύπτει, το neuroevolution (NE) [3], [4]. Είναι ένα νευρωνικό δίκτυο, το οποίο έχει για αλγόριθμο βελτιστοποίησης τον εξελικτικό. Έχοντας υπόψη, τα πλεονεκτήματα των εξελικτικών, προκύπτει ότι είναι χρήσιμοι στο deep learning [5]. Λόγω του τεράστιου μεγέθους ενός τέτοιου δικτύου, καθιστά ιδανικό για χρήση EA. Έτσι γίνεται να πετύχεις ίδια η και καλύτερα αποτελέσματα, σε λιγότερο χρόνο, από έναν συμβατικό αλγόριθμο βελτιστοποίησης. Επίσης γίνεται να φτιαχτεί ένα σύστημα με EA ώστε να βρει ποια αρχιτεκτονική νευρωνικών δικτύων είναι η κατάλληλη για το πρόβλημα. Στη συνέχεια να εκτελεστεί το προτεινόμενο από τους εξελικτικούς δίκτυο με συμβατική μέθοδο βελτιστοποίησης [5]. Σε αυτή την μελέτη θα ερευνηθεί, η χρήση των NE σε προβλήματα time series [6]. Είναι ένα είδος προβλημάτων τα οποία έχουν αυξημένο βαθμό δυσκολίας, έχοντας ένα επιπλέον παράγοντα, τον χρόνο. Θα χρησιμοποιηθούν πέντε dataset, όπου κάποια από τα δεδομένα τους είναι παρμένα από πραγματικά προβλήματα και κάποια από προσομοιώσεις. Θα γίνουν πέντε διαφορετικά πειράματα σε κάθε ένα από τα dataset και στη συνέχεια θα γίνει σύγκρισή του δικού μας συστήματος, με το σύστημα MCNE [7]. Στα δύο αυτά συστήματα έχουν χρησιμοποιηθεί για τα πειράματα τα ίδια dataset, τα οποία είναι χωρισμένα σε training / validation / testing με τον ίδιο τρόπο [8]. Το σύστημα μας έχει βασιστεί με έναν γενικό βαθμό στο δικό τους MCNE. Παράλληλα θα γίνουν προτάσεις για διαφορετικές μεθόδους και παραμετροποιήσεις στους εξελικτικούς αλγόριθμους, όπου μπορούν να χρησιμοποιήσουν όχι μόνο σε NE αλλά γενικά σε προβλήματα βελτιστοποίησης. Τέλος θα γίνει σύγκριση των αποτελεσμάτων από το σύστημα μας σε σχέση με την αναβαθμισμένη μορφή του δικού τους συστήματος MCNE.

## ΚΕΦΑΛΑΙΟ 1 – Μηχανική μάθηση και νευρωνικά δίκτυα

Η τεχνητή νοημοσύνη είναι μια ορολογία που χρησιμοποιείτε για να περιγράψει ένα τεράστιο σύνολο από επιμέρους μεθοδολογίες και τεχνολογίες που έχουν κάτι κοινό μεταξύ τους, την δυνατότητα να μπορεί ένας υπολογιστής να μιμείται την ανθρώπινη συμπεριφορά, ώστε να δίνει την ψευδαίσθηση ότι έχει ευφυΐα . Ένας μεγάλος κλάδος της τεχνητής νοημοσύνης είναι η μηχανική μάθηση (ML). Παρόλου που είναι ένας πολύ παλιός κλάδος, τα τελευταία χρόνια έχει κερδίσει μεγάλο έδαφος, λόγω της ραγδαίας ανάπτυξης της τεχνολογίας του υλικού του υπολογιστή και την εύκολη αγορά του από τον οποιοδήποτε.

Σήμερα είναι σύνηθες, η χρήση του ML ακόμη και από τα smart phone, κυρίως για την πιο καλή λήψη φωτογραφιών και εκπαίδευση του κινητού ώστε να μάθει την εκάστοτε χρήστη, για μια πιο ολοκληρωμένη εμπειρία. Επίσης βρίσκει μεγάλη χρήση σε συστήματα και συσκευές που σχετίζονται με το IoT τόσο για την συλλογή δεδομένων αλλά όσο και για την μετέπειτα επεξεργασία των δεδομένων, για να παραχθούν τα τελικά συμπεράσματα και στατιστικά αποτελέσματα. Ένας ακόμη κλάδος που χρησιμοποιεί σε μεγάλο βαθμό το ML είναι αυτός της ιατρικής. Υπάρχουν πάρα πολλά νοσοκομεία που έχουν τέτοια προγράμματα, με εκπαιδευμένα συστήματα για να μπορούν να δίνουν στους γιατρούς μια δεύτερη γνώμη για το πόρισμα την ασθένειας του εκάστοτε ασθενή. Και άλλες φορές να μπορούν να σώζουν ζωές βρίσκοντας γρήγορα την ασθένεια, εκεί που ο άνθρωπος θα έκανε λάθος. Όμως τέτοια συστήματα δεν είναι τέλεια και υπάρχει πιθανότητα λάθους και στον τομέα της ιατρικής τα λάθη έχουν τεράστια βαρύτητα. Έτσι σε κάποιες χώρες υπάρχει σε τόσο μεγάλο βαθμό η χρήση τεχνητής νοημοσύνης σε τέτοιου είδους συστήματα που η χρήση τους είναι καθημερινή και τίθενται θέματα ηθικής και ευθύνης όταν ο γιατρός εμπιστεύτηκε το σύστημα και προχώρησε με την αγωγή, αλλά τελικά ήταν λάθος.

Τα νευρωνικά δίκτυα και οι τεχνολογίες τους, είναι ο βασικός πυλώνας στους αλγόριθμους ML, θέτοντας τον βασικό αλγόριθμο για την μάθηση και για τον υπολογισμό των δεδομένων. Υπάρχει μια πληθώρα αλγόριθμων ML και τεράστια παλέτα αρχιτεκτονικής νευρωνικών δικτύων. Ωστόσο, χωρίς κάποιο αλγόριθμο βελτιστοποίησης τα νευρωνικά δίκτυα από μόνα τους δεν μπορούν να παράγουν κάποιο χρήσιμο αποτέλεσμα. Για αυτό έχουν δημιουργηθεί πάρα πολλές μέθοδοι βελτιστοποίησης καθώς και συνδυασμός αυτών, η κάθε μια με τα δικά της θετικά και αρνητικά που την καθιστά ιδανική για συγκεκριμένα προβλήματα.

## 1.1 Μηχανική μάθηση

Μηχανική μάθηση είναι η δυνατότητα των υπολογιστών να μπορούν να μάθουν, να λύνουν ένα πρόβλημα, χωρίς όμως να προγραμματιστούν για αυτό.

Η πιο συνηθισμένη διαδικασία εκμάθησης ενός τέτοιου αλγορίθμου χωρίζεται σε κάποια στάδια:

1. Αρχικά πρέπει να γίνει η συλλογή των δεδομένων. Τα δεδομένα αυτά θα μπορούσαν να παραχθούν από την παρατήρηση κάποιου φυσικού φαινομένου, αλλά και με τεχνητό τρόπο με την χρήση υπολογιστή και την βοήθεια κάποιας γεννήτριας τυχαίων αριθμών. Οι πηγές δεδομένων θα μπορούσαν να είναι η συλλογή τιμών από έναν η και περισσότερους αισθητήρες, από κάποια ανοικτή στο κοινό βάση δεδομένων ή ακόμη να γίνει καταγραφή δεδομένων από εμάς. Ουσιαστικά δεν υπάρχει κάποιος περιορισμός για την πηγή από την οποία θα γίνει η συλλογή των δεδομένων. Πόσο εύκολη πρόσβαση θα έχουμε στην πηγή της πληροφορίας εξαρτάται από την φύση του προβλήματος. Σε αυτό το στάδιο προσπαθούμε να συλλέξουμε όσο το δυνατόν μεγαλύτερο αριθμό δεδομένων γίνεται.
2. Έπειτα από την συλλογή δεδομένων πρέπει να γίνει στατιστική ανάλυση σε αυτά. Όστε να παραχθούν συμπεράσματα για να μπορέσουμε να αποφασίσουμε άμα θα τα κρατήσουμε όλα ή θα χρειαστεί κάποια παραμετροποίηση σε αυτά. Με αυτή την διαδικασία μπορούμε να καταλάβουμε ποια χαρακτηριστικά από τα δεδομένα είναι πιο σημαντικά και ποια όχι. Για κάποια χαρακτηριστικά μπορεί να βγάλουμε το συμπέρασμα ότι είναι περίσσεια, αυτό σημαίνει ότι θα κάνουν το σύστημα να παρεκκλίνει από τον πραγματικό του στόχο, έτσι θα αφαιρεθούν από τα δεδομένα. Επίσης μπορεί να υπάρχουν δεδομένα με ελλιπή χαρακτηριστικά. Συνηθίζετε τότε ή να διαγράψουμε τελείως αυτές τις εγγραφές, άμα δεν είναι τόσες πολλές, ή να τις αντικαταστήσουμε με τον μέσο της τιμής τους, ώστε να μην τις χάσουμε τελείως αλλά και να μην επηρεάσουμε πολύ τα πραγματικά δεδομένα. Στη συνέχεια περνάμε από κάποια αλγόριθμο κανονικοποίησης ώστε να έχουν κάποια μορφή που είναι εύκολα επεξεργάσιμη από το σύστημα. Είναι πολύ σημαντικό στάδιο γιατί είναι η τροφή του συστήματος μας.
3. Αφότου έχουμε επιλέξει και κανονικοποιήσει τα δεδομένα μας, τότε τα χωρίζουμε σε datasets. Τα δύο βασικά dataset είναι αυτό του training και του testing. Το training είναι το πιο μεγάλο σε μέγεθος και περιέχει δεδομένα τα οποία το σύστημα μας θα τα χρησιμοποιήσει κατά την διάρκεια της εκπαίδευσης. Τα υπόλοιπα αποτελούν τα δεδομένα

του testing. Αφότου ολοκληρωθεί η εκπαίδευση του συστήματος γίνεται χρήση των δεδομένων του testing για να διαπιστωθεί η αποτελεσματικότητα του συστήματος.

4. Το στάδιο της εκπαίδευσης. Όπου επιλέγονται κάποια δεδομένα που θα χρησιμοποιήσει ο αλγόριθμος για να μπορέσει να εκπαιδευτεί. Ανάλογα δηλαδή με τα δεδομένα που θα του δώσουμε και «δείχνοντας» του τότε κάνει λάθος και τότε σωστό, μπορεί από μόνος του να μάθει να λύνει το πρόβλημα. Μια καλή εκπαίδευση είναι αυτή που θα τελειοποιήσει το σύστημα, ώστε να μπορέσει να παράγει σωστά αποτελέσματα για την πληθώρα των δεδομένων.
5. Το στάδιο της εκτίμησης (evaluation) του συστήματος είναι το πιο χαρακτηριστικό γιατί θα μας δείξει, την αποδοτικότητα του συστήματος μας στο πρόβλημα. Δίνοντας στο σύστημα δεδομένα που τα έχουμε διαχωρίσει από αυτά της εκπαίδευσης βλέπουμε τα τελικά αποτελέσματα του συστήματος μας. Αυτό που κάνει αυτούς τους αλγόριθμους να ξεχωρίζουν είναι ότι τολμούν να παράγουν αποτελέσματα ή και ακόμη να προβλέψουν, με βάση δεδομένων που δεν έχουν ξανά συναντήσει. Σε αντίθεση με ένα συμβατικό σύστημα που για να μπορέσει να ξέρει το αποτέλεσμα θα πρέπει να έχει ξανά συναντήσει τα ίδια ακριβώς δεδομένα.
6. Τέλος γίνονται συλλογή αποτελεσμάτων από πολλές εκτελέσεις του συστήματος και από δοκιμές με διαφορετικές παραμέτρους, ώστε να παραχθούν με στατιστική ανάλυση αυτών τα γενικά συμπεράσματα για το τελικό σύστημα.

## 1.2 Τεχνητά νευρωνικά δίκτυα

Ο άνθρωπος παρατηρώντας την φύση, έχει εμπνευστεί πολλές μεθόδους και τεχνικές τις οποίες έχει μεταφέρει στον κλάδο την πληροφορικής. Τα νευρωνικά δίκτυα προσπαθούν να αντιγράψουν τη βιολογική λειτουργία του ανθρώπινου εγκεφάλου. Κάθε νευρώνας λαμβάνει μια διέγερση και έπειτα από επεξεργασία, στέλνει ως έξοδο του την δικιά του διέγερση στους υπόλοιπους νευρώνες με τους οποίους είναι συνδεδεμένος. Έτσι και με τα νευρωνικά δίκτυα, τα οποία είναι ένα σύνολο από νευρώνες που είναι συνδεδεμένοι μεταξύ τους σε μια διάταξη. Δέχεται ως είσοδο κάποια δεδομένα το νευρωνικό δίκτυο, στη συνέχεια αυτά τα δεδομένα, τα επεξεργάζεται και τελικά παράγει ένα αποτέλεσμα. Πιο συγκεκριμένα ένας νευρώνας αποτελείται από βάρη και μια συνάρτηση ενεργοποίησης. Το βάρος είναι ένας αριθμός, που δείχνει, πόσο μεγάλη βαρύτητα έχει

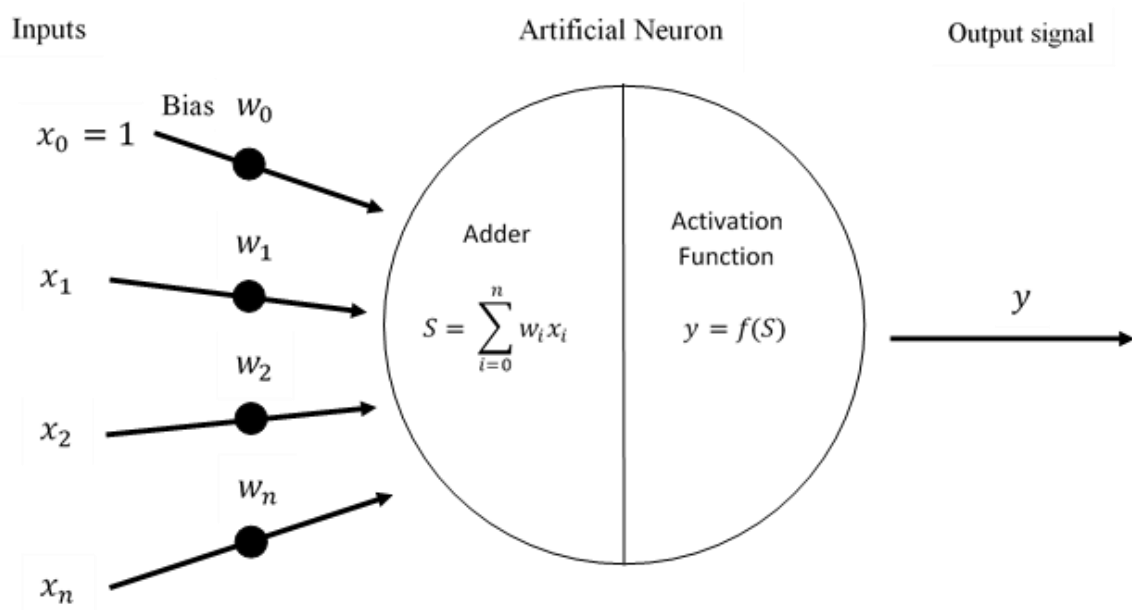
αυτό το χαρακτηριστικό, για τον συγκεκριμένο νευρώνα. Η συνάρτηση ενεργοποίησης είναι μια μαθηματική συνάρτηση, συνηθίζεται να είναι μια μη γραμμική συνάρτηση. Το είδος της εξαρτάται από τον τύπο της πληροφορίας που θέλουμε να εξάγει το δίκτυο για το εκάστοτε πρόβλημα. Σε συχνή χρήση είναι η σιγμοειδής (1.1), υπερβολική εφαιπτόμενη (1.2) και η συνάρτηση του gauss (1.3). Για συναρτήσεις εξόδου στο στρώμα εξόδου, η δυαδική (κωδικοποίηση της τιμής εξόδου σε 0 ή 1), Parametric ReLU (1.4). Η κάθε μια έχει το δικό της χαρακτηριστικό έτσι δεν υπάρχει κάποιος γενικός κανόνας για την επιλογή της καταλληλότερης. Έτσι πρέπει να γίνουν πειράματα με διαφορετικές συναρτήσεις και με βάση τα αποτελέσματα να παρθεί η απόφαση. Το κάθε ένα δεδομένο πολλαπλασιάζεται με το αντίστοιχο του βάρους και στη συνέχεια αθροίζονται. Το αποτέλεσμα αυτό τελικά περνά από μια συνάρτηση ενεργοποίησης, όπου είναι το τελικό αποτέλεσμα του νευρώνα.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (1.1)$$

$$f(x) = \frac{\sinh x}{\cosh x} \quad (1.2)$$

$$f(x) = e^{-x^2} \quad (1.3)$$

$$f(x) = \max(x \cdot a, x) \quad a \in [0, 1] \quad (1.4)$$



Σχήμα 1 Μοντέλο τεχνητού νευρώνα

Τα κυριότερα πλεονεκτήματα που έχουν τα νευρωνικά δίκτυα είναι:

- **Ικανότητα μάθησης**, δηλαδή μέσα από την εκπαίδευσή τους, μπορούν να μάθουν να λύνουν ένα πρόβλημα, λόγω της προσαρμογής στα βάρη τους.
- **Γενίκευση** το πιο δυνατό τους χαρακτηριστικό. Στην ουσία είναι η δυνατότητα τους, να μπορούν να παράγουν αποτελέσματα, για δεδομένα που δεν έχουν δει κατά την εκπαίδευσή τους.
- **Μεγάλο σύνολο** από μεθόδους βελτιστοποίησης με βάση τα νευρωνικά δίκτυα. Που βοηθάνε στην επίλυση προβλημάτων τύπου κατάταξης και τύπου παλινδρόμησης. Μπορούν να χρησιμοποιηθούν και σε προβλήματα που πρέπει να γίνει πρόβλεψη μελλοντικών τιμών. Όπως σε προβλήματα χρονοσειρών, δίνοντας στο σύστημα παρελθοντικές τιμές, μπορεί και προβλέπει τις μελλοντικές.

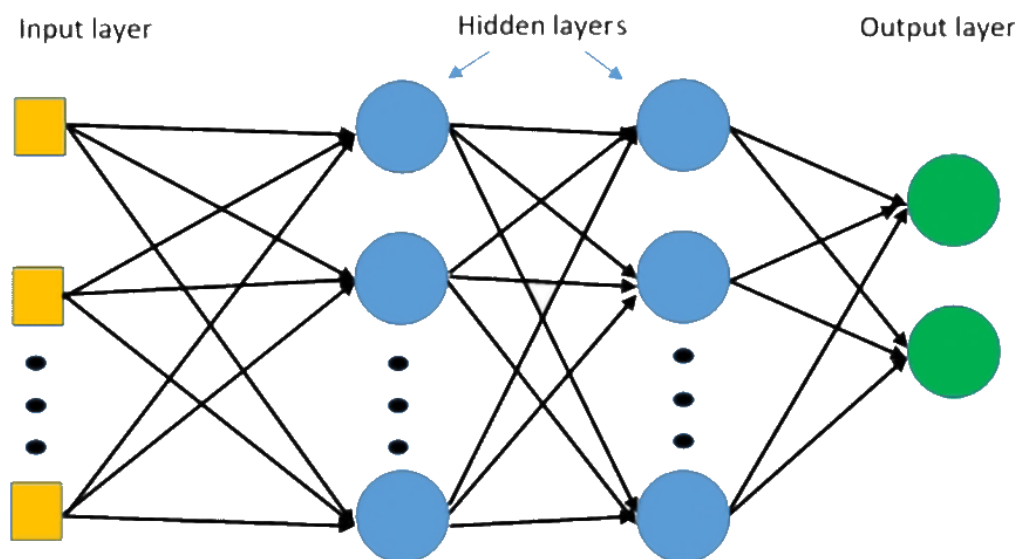
Τα σημαντικότερα μειονεκτήματα τους είναι:

- **Έλλειψη γενικού κανόνα**, δεν υπάρχει κάποιος γενικός κανόνας για την καταλληλότερη χρήση της συνάρτησης ενεργοποίησης, καθώς επίσης, για τον χρυσό αριθμό στρώματων και το πλήθος νευρώνων σε κάθε στρώμα. Έτσι είναι αναγκαίο για να προσδιοριστούν τα παραπάνω, πολλαπλές προσπάθειες και συνεχείς δοκιμές, μέχρις ότου φτάσουμε σε ένα δίκτυο που παράγει ικανοποιητικά αποτελέσματα.
- **Μικρός έλεγχος**, εξαιτίας της φύσης τους, δεν μπορείς να έχεις μεγάλο έλεγχο στο δίκτυο, ώστε να μπορέσεις να το κατευθύνεις, ούτε υπάρχει κάποια χρήσιμη πληροφορία που μπορεί να αντλήσει κάποιος βλέποντας τα βάρη και τις συνδέσεις του δικτύου. Γενικά η πραγματική λειτουργία ενός νευρωνικού δικτύου, είναι ένα μαύρο κουτί, το μόνο που ξέρουμε είναι τα δεδομένα εισόδου και τα αποτελέσματα του.

### 1.3 Multilayer perceptron

Πολύ γρήγορα έγινε αντιληπτό ότι, οι δυνατότητες ενός μόνο στρώματος νευρώνων μπορούσε να λύσει ελάχιστα προβλήματα. Έτσι δημιουργήθηκε το multilayer perceptron (MLP) [9]. Πρακτικά είναι πολλά στρώματα νευρώνων, τα οποία η έξοδος του προηγούμενου είναι η είσοδος του επόμενου, ξεκινώντας από το την αρχική είσοδο των δεδομένων στο δίκτυο. Πλέον σήμερα είναι

η βάση για πάρα πολλά συστήματα καθώς έχει μεγάλη ευελιξία στον σχεδιασμό του χωρίς όριο στο πλήθος στρωμάτων και πλήθος νευρώνων ανά στρώμα.



Σχήμα 2 Μοντέλο νευρωνικού δικτύου multilayer perceptron

### 1.3 Backpropagation

Για να μπορέσει ένα νευρωνικό δίκτυο να παράγει αποτελέσματα πρέπει να λάβουν τιμές τα βάρη του. Αυτό όμως δεν μπορεί να γίνει τυχαία, έτσι γίνεται η χρήση μιας μεθόδου βελτιστοποίησης. Ο ρόλος αυτής της μεθόδου είναι πάρα πολύ σημαντικός για να μπορέσουμε να εκπαιδύσουμε σωστά ένα δίκτυο. Η πιο γνωστή, τουλάχιστον για τα MLP δίκτυα είναι ο αλγόριθμος backpropagation (BP). Κατά την διαδικασία της εκπαίδευσης ορίζεται από τον χρήστη, οι μέγιστες εποχές (επαναλήψεις) του συστήματος. Μία εποχή είναι όταν το σύστημα έχει υπολογίσει για κάθε ένα δεδομένο εισόδου, την τιμή εξόδου του. Οι εποχές ορίζονται για να μπορέσει κάποτε να τερματιστούν αυτές οι επαναλήψεις. Ένας άλλος τρόπος τερματισμού είναι με την χρήση ενός κατώφλι, που όταν το ξεπεράσει το σύστημα τότε θα τερματιστεί. Το κατώφλι μπορεί να είναι μια τιμή σφάλματος που μας ικανοποιεί, έτσι ώστε να μην χρειάζεται η συνέχεια εκτέλεσης του συστήματος. Σε κάθε εποχή υπολογίζεται η έξοδος του δικτύου και συγκρίνεται με τις τιμές των πραγματικών δεδομένων. Στη συνέχεια με μία συνάρτηση υπολογισμού σφάλματος ο αλγόριθμος BP θα κάνει τις κατάλληλες διορθώσεις στα βάρη του δικτύου ώστε την επόμενη φορά να μπορέσει να πετύχει καλύτερα αποτελέσματα. Υπάρχουν πάρα πολλές παραλλαγές και στρατηγικές ML που να έχουν ως βασικό τους αλγόριθμο βελτιστοποίησης τον BP.

Είδαμε τις πιο βασικές ή αλλιώς συμβατικές μεθόδους και τεχνικές μηχανικής μάθησης και νευρωνικών δικτύων. Ενώ ο αλγόριθμος BP είναι πολύ καλώς για την εκπαίδευση νευρωνικών δικτύων για πολλά προβλήματα, είναι όμως άλυτα συνδεδεμένος με μαθηματικές συναρτήσεις για την σωστή του λειτουργία, που τον καθιστά δυσλειτουργικό για αρκετά προβλήματα. Αν θερίσουμε ότι ένα εκπαιδευμένο νευρωνικό δίκτυο είναι μια λύση του προβλήματος, τότε μπορούμε να πούμε ότι ο BP δεν μπορεί να κάνει καλή αναζήτηση στο χώρο των πιθανών λύσεων, λόγω ότι δεν περιέχει μια τυχαία παράμετρο που να λαμβάνει τιμή με τυχαίο τρόπο, για να μπορέσει να κάνει μια πιο γενική αναζήτηση.

## ΚΕΦΑΛΑΙΟ 2 – Εξελικτικοί αλγόριθμοι

Οι εξελικτικοί αλγόριθμοι (EA) δημιουργήθηκαν με την σκέψη ότι η βιολογική διαδικασία της εξέλιξης θα μπορούσε να αποτελέσει τη βάση για την ανάπτυξη αλγορίθμων βελτιστοποίησης [1].

### 2.1 Βασικές ιδιότητες και χαρακτηριστικά

Η θεωρία στηρίζεται στην φυσική επιλογή και την γενετική μεταβολή, έτσι έχοντας ένα πληθυσμό από άτομα, τα οποία ανταγωνίζονται μεταξύ τους, ώστε να επιλεχθούν οι καλύτεροι. Ανασυνδυάζοντας τους καλύτερους μεταξύ τους θα παραχθούν καλύτεροι απόγονοι. Άμα συνεχιστεί ή διαδικασία σε μια επανάληψη, τότε μετά από καιρό θα φτάσει ο πληθυσμός να περιέχει άτομα που είναι πολύ καλύτερα από τα αρχικά. Ουσιαστικά τα άτομα του πληθυσμού σε ένα πραγματικό πρόβλημα είναι μια υποψήφια λύση του προβλήματος.

Οι EA χωρίζονται σε κάποια στάδια:

1. Αρχικά, δημιουργείται ένας πληθυσμός (population) από άτομα (individuals) (Σχήμα 3 – Step 1). Κάθε άτομο περιέχει χρωμοσώματα και τα χρωμοσώματα περιέχουν γονίδια. Τα γονίδια είναι οι τιμές που έχουν τα χαρακτηριστικά του ατόμου, που είναι και αυτά που θα καθορίσουν πόσο καλή λύση του προβλήματος είναι. Ο αριθμός αυτός είναι η τιμή ικανότητας (fitness) του ατόμου.
2. Αφού βρεθεί το fitness για το κάθε άτομο (Σχήμα 3 – Step 2) τότε, φτιάχνετε ένα νέο σύνολο από άτομα, τα οποία επιλέχθηκαν βάση μιας μεθόδου, με βάση το fitness τους (Σχήμα 3 – Step 3). Γενικά επιλέγονται αυτοί που είναι πιο καλοί, αλλά δίνεται και η ευκαιρία να επιλεχθούν και οι πιο κακοί, γιατί μπορεί να περιέχουν γενετικό κώδικα που είναι χρήσιμος για το επόμενο στάδιο.
3. Αυτό το στάδιο είναι ο ανασυνδυασμός, του συνόλου από το προηγούμενο στάδιο. Έτσι αυτά τα άτομα θα γονιμοποιήσουν την επόμενη γενιά (Σχήμα 3 – Step 4). Ο ανασυνδυασμός είναι ο βασικότερος τρόπος ανάπτυξης του πληθυσμού. Συνήθως επιλέγονται δύο άτομα από το σύνολο που παράχθηκε μέσω της μεθόδου επιλογής. Έτσι ανασυνδυάζοντας τα δύο αυτά άτομα, πιθανότατα θα παραχθούν δύο ακόμη καλύτεροι

απόγονοι. Ο ανασυνδυασμός είναι αυτός που κάνει τους εξελικτικούς, έναν αλγόριθμο στοχευμένης αναζήτησης και όχι απλά μια τυχαία αναζήτηση.

4. Οι νέοι απόγονοι θα περάσουν από μια μέθοδο τυχαίας μετάλλαξης των γονιδίων τους. Αυτό γίνεται για να μπορέσει ο πληθυσμός να ξεπεράσει τυχόν τοπικά βέλτιστα, κατά την πορεία ανάπτυξης του (Σχήμα 3 – Step 5).
5. Τέλος οι απόγονοι ενσωματώνονται στον πληθυσμό γίνεται ο υπολογισμός του fitness για κάθε νέο άτομο του πληθυσμού (Σχήμα 3 – Step 6) και η διαδικασία ξεκινά από την αρχή

```
1 BEGIN
2   Step 1: INITIALISE population with random candidate solutions;
3   Step 2: EVALUATE each candidate;
4   REPEAT UNTIL (TERMINATION CONDITION is satisfied) DO
5     Step 3: SELECT parents;
6     Step 4: RECOMBINE pairs of parents;
7     Step 5: MUTATE the resulting offspring;
8     Step 6: EVALUATE new candidates;
9   DONE
10 END
```

Σχήμα 3 Ψευδοκώδικας Εξελικτικού Αλγορίθμου

Τα πλεονέκτημα που κερδίζει κάποιος από τους EA, είναι ότι μπορούν να λύσουν δύσκολα προβλήματα και γρήγορα. Όσο μεγαλύτερο είναι το πλήθος των πιθανών βέλτιστων λύσεων τόσο πιο αποδοτικοί είναι και από άποψη ταχύτητας αλλά και αποτελέσματος.

Τα κυριότερα πλεονεκτήματα τους είναι:

- Οι περισσότερες συμβατικές μέθοδοι είναι δύσκαμπτες ή και ακατάλληλες κάποιες φορές, για αρκετά προβλήματα, λόγω της ανάγκης τους για χρήση περίπλοκων μαθηματικών. Η χρήση τέτοιου είδους μαθηματικών είναι αδιάφορη για τους EA, που αυτό τους καθιστά κατάλληλους για ένα μεγάλο φάσμα προβλημάτων.
- Είναι από τις λίγες μεθόδους, που κάνει ταυτόχρονα εξερεύνηση του χώρου αναζήτησης και εκμεταλλεύεται την ήδη επεξεργασμένη πληροφορία. Έτσι με το τυχαίο ψάξιμο γίνεται καλή εξερεύνηση του χώρου και με την αναζήτηση γίνεται καλή εκμετάλλευση της πληροφορίας.

## ΚΕΦΑΛΑΙΟ 2

- Ένα ακόμη μεγάλο πλεονέκτημα τους είναι ο εύκολος παραλληλισμός, μια δυνατότητα που σπανίζει στις υπόλοιπες ανταγωνιστικές μεθόδους. Έχει ως αποτέλεσμα να μπορούν να πετύχουν καλύτερα αποτελέσματα, αλλά και σε μικρότερο χρόνο.
- Τέλος μπορούν να συνδυαστούν και με άλλες μεθόδους. Αν και η ισχύς των EA είναι μεγάλη, υπάρχουν μερικές περιπτώσεις προβλημάτων στα οποία άλλες μέθοδοι παράγουν πολύ καλά αποτελέσματα, λόγω της εξειδίκευσης τους στο πρόβλημα. Αυτό είναι αποτέλεσμα της μεγάλης ευελιξίας των EA

Όπως όμως με κάθε μέθοδο υπάρχουν και μειονεκτήματα:

- Το πρώτο που πρέπει να κάνει κάποιος που θέλει να λύσει ένα πρόβλημα με τους EA είναι να κωδικοποιήσει το πρόβλημα με τέτοιο τρόπο ώστε να μπορούν να το επεξεργαστούν οι EA. Η οποία διαδικασία μπορεί μερικές φορές να είναι δύσκολη ανάλογα με την φύση του προβλήματος. Η κωδικοποίηση αυτή είναι μια διαδικασία που θα χρειαστεί χρόνο, για να μπορέσει κάποιος να κατανοήσει και να νοιώσει άνετα με αυτήν.
- Το επόμενο βήμα είναι να μπορέσει κάποιος να ορίσει και να υλοποιήσει ένα στόχο για EA. Αυτός ο στόχος είναι που θα καθοδηγήσει τον πληθυσμό του αλγορίθμου στην βέλτιστη λύση. Επομένως το στάδιο αυτό, επηρεάζει κατά μεγάλο βαθμό στην εύρεση της βέλτιστης λύσης. Το οποίο και αυτό με την σειρά του ζητά χρόνο και εξοικείωση μαζί του.

Οι EA είναι μια μεθοδολογία πολύ παλιά η οποία δεν υπήρχε σε καθημερινή χρήση. Σήμερα έχει ξανά γεννηθεί μέσα από την μηχανική μάθηση, έχοντας τον ρόλο τις μεθόδου βελτιστοποίησης του αλγορίθμου.

## ΚΕΦΑΛΑΙΟ 3 – Neuroevolution

Λόγω της μεγάλης ευελιξίας που κατέχουν οι ΕΑ, δίνουν την δυνατότητα να μπορούν να συνδυαστούν με άλλες μεθόδους. Για να μπορέσουν όμως να λύσουν προβλήματα μηχανικής μάθησης χρειάζονται τα νευρωνικά δίκτυα. Έτσι προκύπτει ο συνδυασμός των ΕΑ μαζί με τα νευρωνικά δίκτυα. Τα νευρωνικά δίκτυα από μόνα τους δεν μπορούν να εκπαιδευτούν, έχουν την ανάγκη χρήσης αλγόριθμου βελτιστοποίησης. Ένας από τους πιο γνωστούς είναι ο backpropagation (BP). Στο NE [3], [4], την θέση του αλγορίθμου βελτιστοποίησης καταλαμβάνουν οι ΕΑ.

### 3.1 Συστήματα & Αρχιτεκτονικές

Τα θετικά που προκύπτουν, είναι ότι πλέον δεν είναι αναγκαία η χρήση περίπλοκων μαθηματικών για την εκπαίδευση και βελτιστοποίηση του δικτύου. Επίσης δίνονται πολλοί τρόποι για παραμετροποίηση και έλεγχο κατά την εκπαίδευση, που μπορούν να συνδυαστούν με πάρα πολλές μεθόδους. Καθώς μειώνονται σε σημαντικό βαθμό, οι προσπάθειες και τα πειράματα που χρειάζονται για να μπορέσουμε να βρούμε το κατάλληλο δίκτυο. Και σε πολύ μεγάλα δίκτυα, για παράδειγμα στο deep learning, μειώνονται κατά πολύ οι χρόνοι εκπαίδευσης, παράγοντας όμως σταθερά καλά αποτελέσματα.

Για την κατασκευή ενός συστήματος NE είναι απαραίτητη η κατάλληλη επιλογή συνάρτησης για τον υπολογισμό του fitness και εισαγωγή των κατάλληλων επιλογών στις παραμέτρους του συστήματος. Αλλά το σημαντικότερο είναι η μέθοδος κωδικοποίησης του προβλήματος, ώστε να μπορούν να εφαρμοστούν οι μέθοδοι των ΕΑ σε αυτό. Γιατί με αυτήν την επιλογή μας θα κριθεί το εύρος των υπόλοιπων μεθόδων που θα μπορούμε να χρησιμοποιήσουμε κατά την κατασκευή του συστήματος. Υπάρχουν τρεις γενικές οικογένειες κωδικοποίησης: Direct, Developmental και Implicit. Οι μέθοδοι αυτοί μπορούν να χρησιμοποιηθούν τόσο για την κατασκευή και περιγραφή της τοπολογίας του νευρωνικού δικτύου όσο και τις παραμέτρους του. Οι μέθοδοι Developmental και Direct χρησιμοποιούνται πιο πολύ για την κατασκευή τοπολογίες του δικτύου, λόγω τις μεγάλης τους ευελιξίας δηλαδή την δυνατότητα να μπορούν να παράγουν διαφορετικού μεγέθους και τοπολογίας δίκτυο, καθώς και να κάνουν μερική χρήση των συνδέσεων μεταξύ των νευρώνων όπου αυτό θεωρείτε αναγκαίο από το σύστημα. Ενώ η μέθοδος Direct βρίσκετε συνήθως όταν έχουμε να ανατήξουμε δίκτυα σταθερού μεγέθους.

### 3.1.1 Η μέθοδος Direct

Σε αυτήν την μέθοδο υπάρχει απευθείας συσχέτιση ένα προς ένα ανάμεσα στις τιμές των νευρώνων (βάρη, bias) με την κωδικοποιημένη μορφή του δικτύου. Η κατασκευή του είναι η πιο απλή σε σχέση με τις άλλες δύο μεθόδους, δίχως έχοντας κάποια επιπλέον περίπλοκη λογική για την κωδικοποίηση του δικτύου. Υπάρχουν αρκετοί τρόποι που μπορεί κάποιος να υλοποιήσει αυτή την μέθοδο, είτε με δεκαδικές τιμές είτε με χαρακτήρες που στην συνέχεια θα πάρουν την τελική τους αριθμητική μορφή.

Αυτή είναι η μέθοδος που επιλέξαμε για την κατασκευή του συστήματος. Υπάρχει όμως μια σημαντική διαφορά. Το συστήματα και οι αρχιτεκτονικές που περιγράφουν αυτήν την μέθοδο, διαχειρίζονται τις μεθόδους των ΕΑ πρώτα πάνω στην λογική του δικτύου. Δηλαδή μπορεί να αντληθεί την έννοια του νευρώνα και του νευρωνικού δικτύου, έτσι μπορεί για παράδειγμα στο τελεστή της μετάλλαξης να προσθέσει, να αφαιρέσει η και να αλλάζει θέσεις τους νευρώνες του δικτύου. Έπειτα το δίκτυο κωδικοποιείται ώστε να μπορεί να γίνει ο ανασυνδυασμός στην επόμενη εποχή. Ενώ στο σύστημα μας δεν υπάρχει η λογική του δικτύου κατά την εφαρμογή των μεθόδων ΕΑ, έως ότου φτάσει η στιγμή για την χρήση του δικτύου, είτε για training είτε για evaluation. Άρα το σύστημα μας όσο αφορά το κομμάτι των ΕΑ «βλέπει» μόνο δεδομένα, χωρίς να ξέρει την πραγματική τους φύση, που στην προκειμένη περίπτωση είναι ένα νευρωνικό δίκτυο.

Ένας πίνακας από δεκαδικούς αριθμούς περιγράφει το κωδικοποιημένο δίκτυο έχοντας τα βάρη και τα bias του κάθε νευρώνα. Αυτό καθιστά πολύ εύκολη την υλοποίηση απλών μεθόδων για ανασυνδιασμό και μετάλλαξη, χωρίς να υπάρχουν επιπλοκές κατά την χρήση τους. Η κωδικοποίηση αυτή έχει τα θετικά της ταχύτητας λόγω της απευθείας ανάθεσης και ανάγνωσης τιμών και ελάχιστη κατανάλωση σε πόρους της μνήμης.

### 3.1.2 Η μέθοδος Developmental

Η χρήση της προηγούμενης μεθόδου Direct συνηθίζεται σε δίκτυα με σχετικά μικρό μέγεθος. Σε αντίθεση με την μέθοδο Developmental όπου είναι κατάλληλη για κατασκευή μεγάλων και περίπλοκων δικτύων. Σε αυτήν την μέθοδο η κωδικοποίηση γίνεται με τέτοιο τρόπο ώστε να μπορεί να αλλάζει το μέγεθος και το σχήμα του δικτύου και με το πέρασμα των εποχών να μεγαλώνει αλλά πάντα με την καθοδήγηση των ΕΑ. Έτσι στο αποτέλεσμα της κωδικοποίησης μπορεί να υπάρχουν δεδομένα που περιγράφουν νευρώνες οι οποίοι όμως δεν έχουν ακόμη συνδεθεί με το υπόλοιπο δίκτυο.

Η μέθοδος αυτή δίνει την δυνατότητα μάθησης, βελτίωσης και κατασκευής του δικτύου σε πραγματικό χρόνο, ενώ δηλαδή το πρόγραμμα βρίσκεται στην ζωντανή χρήση του. Αυτό είναι κάτι που ξεχωρίζει από τα συνηθισμένα συστήματα, όπου η τοπολογία του δικτύου καθώς και τα βάρη του είναι ήδη προκαθορισμένα κατά την διάρκεια εκτέλεσης του προγράμματος στην ζωντανή εφαρμογή.

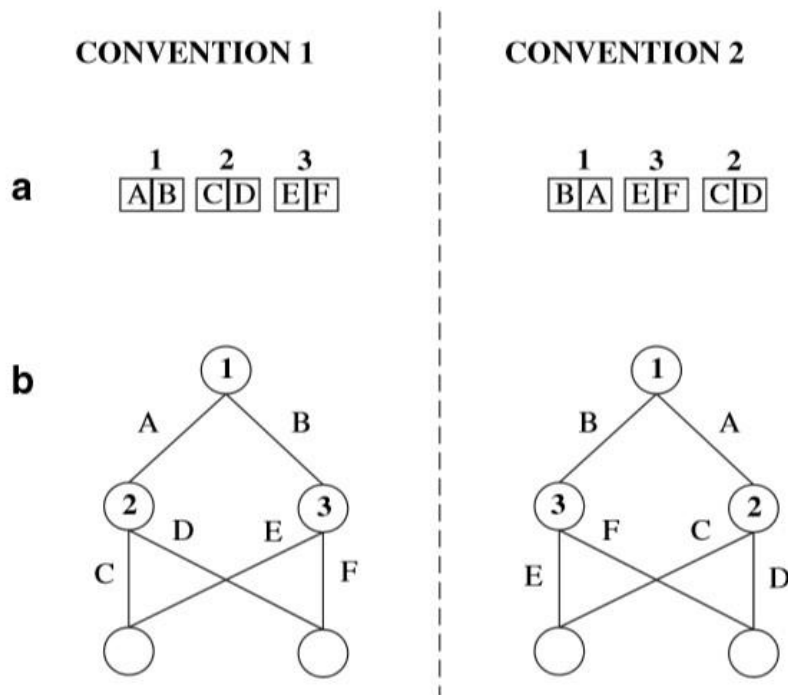
### 3.1.3 Η μέθοδος Implicit

Είναι εμπνευσμένη από μια βιολογική διαδικασία, στην οποία τα γονίδια δεν κωδικοποιούνται απευθείας αλλά ελέγχονται από το φυσικό τους περιβάλλον και τα χημικά που υπάρχουν σε αυτό. Έτσι υπάρχει μια ενδιάμεση διαδικασία πριν μπορέσει η πληροφορία από το νευρωνικό δίκτυο να κωδικοποιηθεί. Είναι αρκετά πιο περίπλοκη από τις δύο προηγούμενες έχοντας περισσότερα στάδια για την ολοκλήρωση της κωδικοποίησης.

Σε αυτήν την μέθοδο συνηθίζεται να αναπαριστάται το γενετικό υλικό ως μια αλληλουχία χαρακτήρων που χωρίζεται σε κομμάτια. Το κάθε κομμάτι είναι η αναπαράσταση ενός νευρώνα και το κάθε κομμάτι συνδέεται με τα άλλα με ένα προκαθορισμένο αλφαριθμητικό.

## 3.2 Το σύστημα NEAT

Neuroevolution of augmenting topologies (NEAT) είναι ένα σύστημα που κωδικοποιεί το γενετικό υλικό και εξελίσσει νευρωνικά δίκτυα. Έχει την δυνατότητα να δημιουργεί δυναμικά νέα γονίδια και να τα εισάγει στο γενετικό κώδικα καθώς και να τα αφαιρεί. Έτσι παράγει γενετικός κώδικας μεταβλητού μεγέθους. Ένα συνηθισμένο πρόβλημα στα NE συστήματα είναι ότι γίνεται να κατασκευαστούν δίκτυα όπου τα σχήματα είναι τους είναι αντίστροφα μεταξύ τους. Δηλαδή είναι δύο δίκτυα με μόνη διαφορά την θέση των νευρώνων, αλλά όλα τα άλλα (βάρη και bias) είναι με ίδιες τιμές και στα δύο. Αυτό έχει ως αποτέλεσμα να παράγουν τα ίδια αποτελέσματα και να έχουν την ίδια απόδοση αλλά για τους EA είναι διαφορετικά, λόγω ότι έχουν διαφορετικό γενετικό κώδικα. Το χειρότερο είναι ότι σε αυτήν την περίπτωση μπορεί κατά την διάρκεια του crossover να παραχθούν απόγονοι με διπλότυπο γενετικό κώδικα που συνεπάγεται σε πολύ χαμηλό fitness.



Σχήμα 4 Παράδειγμα αντίστροφων δικτύων. Παρατηρούμε ότι ο γενετικός κώδικας (α) είναι πολύ διαφορετικός ανάμεσα στα δύο δίκτυα, αλλά τα δίκτυα (β) έχουν ίδια απόδοση.

Το NEAT είναι σχεδιασμένο έτσι ώστε να λύνει το παραπάνω πρόβλημα, έχοντας την δυνατότητα ευελιξίας για την αλλαγή του σχήματος στα δίκτυα. Μπορεί να εφαρμόσει crossover πάνω σε δύο άτομα με διαφορετικό μέγεθος γενετικού υλικού, καθώς ξεκινά από ένα απλό μικρό δίκτυο και με το πέρας των εποχών φτάνει να είναι μεγάλο και περίπλοκο. Για να το καταφέρει αυτό καταγράφει με έναν μοναδικό στίγμα (αριθμό) το κάθε γονίδιο. Αυτός το στίγμα προσδιορίζει την χρονική στιγμή δημιουργίας του γονιδίου, κατά την ανάπτυξη του πληθυσμού. Έπειτα στο crossover μπορεί να ξέρει πόσο όμοια είναι δύο άτομα, ελέγχοντας πόσο κοντά είναι τα στίγμα τους. Ο στόχος του είναι να μην μπλέκονται σε μεγάλο βαθμό άτομα που είναι παλιά άρα θεωρητικά έχουν καλό fitness με νέα άτομα που δεν έχουν καλό fitness. Επίσης τα στίγματα βοηθάνε να βρίσκει εύκολα τα σημεία που μπορούν να συνδεθούν δύο κομμάτια από δύο διαφορετικά δίκτυα με άνισο μέγεθος γενετικού κώδικα.

Κάποιες μελέτες έδειξαν, η απλή μέθοδος κωδικοποίησης Direct μπορεί να φέρει πολύ καλά αποτελέσματα στην περίπτωση που γνωρίζουμε στο περίπου το κατάλληλο σχήμα για το δίκτυο [αναφορά2]. Σε αυτήν την περίπτωση συνήθως το δίκτυο είναι μικρού μεγέθους. Αντιθέτως στις

## Neuroevolution

περιπτώσεις που οι απαιτήσεις του προβλήματος δείχνουν την ανάγκη για χρήση πιο μεγάλου και περίπλοκου δικτύου τότε καλύτερη επιλογή είναι οι μέθοδοι Developmental και Implicit.

Το NEAT είναι ένα πολύ γνωστό σύστημα, που υπάρχει ως βιβλιοθήκη για την γλώσσα προγραμματισμού Python. Αρκετά παραμετροποιήσιμη και με μεγάλο εύρος μεθόδων.

## ΚΕΦΑΛΑΙΟ 4 – Προτεινόμενη αρχιτεκτονική για επίλυση προβλημάτων χρονοσειρών

Υπάρχουν αρκετές μελέτες στον κλάδο του NE. Μία από τις πιο πρόσφατες ασχολείται με προβλήματα τύπου time series [6]. Είναι ένα ιδιαίτερο είδος προβλημάτων, γιατί φέρει το στοιχείο του χρόνου, ως μια επιπλέον παράμετρο. Η μελέτη αυτή [7], προτείνει ένα αναβαθμισμένο σύστημα, για την επίλυση τέτοιων προβλημάτων. Πρακτικά, συνδυάζει τους EA με συμβατικούς αλγορίθμους, με στόχο να πετύχει καλύτερα αποτελέσματα. Σε δεύτερη μελέτη τους έχουν το ίδιο σύστημα, αλλά πειραματίζονται με διαφορετικές στρατηγικές [8], πετυχαίνοντας έτσι πολύ καλά αποτελέσματα. Εμείς έχοντας υπόψη μας τα καλύτερα αποτελέσματα από την δεύτερη τους μελέτη, σε σύγκριση με την δική μας και στη συνέχεια βασισμένοι ως μια γενική εικόνα στη παλιά τους στρατηγική MCNE [7] καταφέραμε να παράγουμε καλύτερα αποτελέσματα, με περίπου το ίδιο κόστος. Επίσης, έχουμε χρησιμοποιήσει τα ίδια datasets, από την πρόσφατη δημοσίευση τους [8], [10], [11], [12], [13], [14] ώστε να μπορεί να υπάρξει κάποια ουσιαστική σύγκρισή των συστημάτων.

### 4.1 Το σύστημα μας

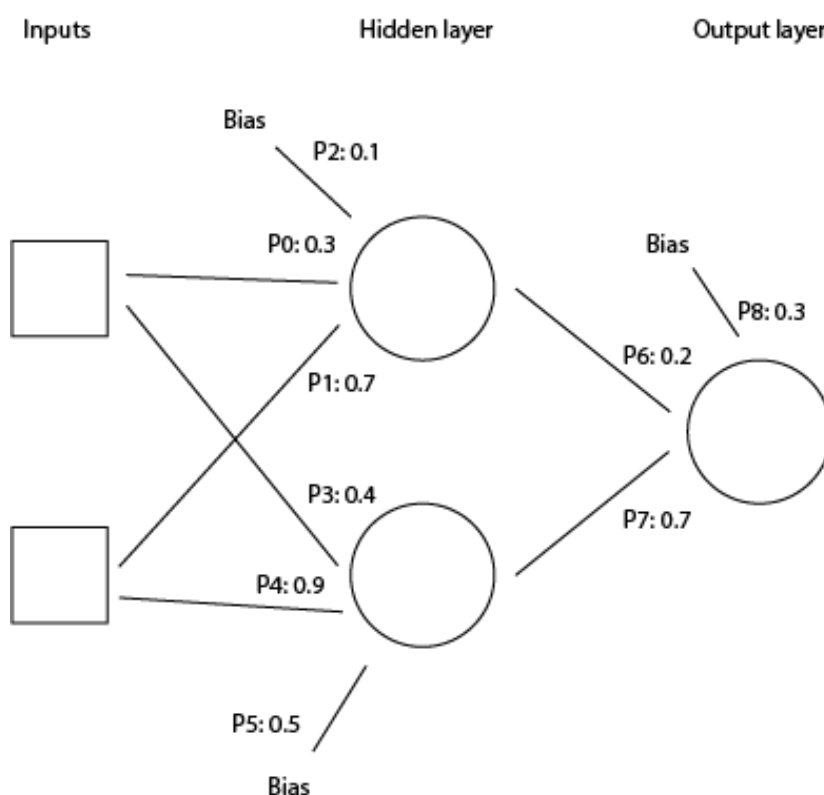
Για την εκπαίδευση νευρωνικού δικτύου από τους EA, είναι απαραίτητη η υλοποίηση μια μεθόδου κωδικοποίησης του δικτύου, έτσι ώστε να είναι προσβάσιμο από τους EA. Η κωδικοποίηση του νευρωνικού δικτύου διαφέρει στην δική μας μέθοδο, σε σχέση με την υπάρχουσα μελέτη [7]. Καθώς αυτό είναι ένα από τα σημαντικότερα κομμάτια των EA, έχει μεγάλη επίπτωση στα καλά αποτελέσματα του αλγορίθμου. Για το σύστημα μας ένα σύνολο από νευρωνικά δίκτυα είναι ένας υποπληθυσμός, αυτό συνεπάγεται ότι τα άτομα του υποπληθυσμού, είναι νευρωνικά δίκτυα. Ενώ στο σύστημα τους [7], ο κάθε νευρώνας είναι και ένας υποπληθυσμός και τα άτομα του είναι νευρώνες με διαφορετικά βάρη. Αυτό έχει ως αποτέλεσμα κατά την ανάπτυξη του υποπληθυσμού και όταν γίνεται αντικατάσταση των χειρότερων ατόμων του από τα καλύτερα, να αλλάζουν μόνο οι νευρώνες. Ενώ στη δική μας μέθοδο αλλάζει όλο το νευρωνικό δίκτυο. Για να υπάρχει η δυνατότητα σύγκρισης των αποτελεσμάτων ανάμεσα στα δυο συστήματα, ο υπολογισμός του fitness και στα δύο συστήματα γίνεται με τον ίδιο τρόπο, καθώς και το γενικό σκεπτικό των συστημάτων είναι το ίδιο. Από τις τρεις μεθόδους που παραθέτουν στην πρόσφατη έρευνα τους [8], η MCNE [7] είναι πιο κοντά στο δικό μας σύστημα και με αυτή θα γίνει η σύγκριση.

#### 4.1.1 Κατασκευή του συστήματος

Η κατασκευή του δικού μας συστήματος δεν βασίστηκε πάνω σε κάποια βιβλιοθήκη. Όλα έχουν δημιουργηθεί από την αρχή. Αυτή η απόφαση πάρθηκε με το σκεπτικό, ότι για να γίνει σωστή έρευνα πρέπει να υπάρχει πλήρης έλεγχος πάνω στο σύστημα, κάτι που δεν μπορεί να σου προσφέρει μια έτοιμη βιβλιοθήκη. Βέβαια ακολουθήθηκαν οι γνωστές τεχνικές και για τους EA αλλά και για τα νευρωνικά δίκτυα, με κάποιες παραμετροποιήσεις. Το πλεονέκτημα που κερδίζει κάποιος με αυτήν την υλοποίηση είναι, ότι έχει πολλές παραμέτρους που μπορεί να ελέγξει με πολύ εύκολο τρόπο, ώστε να κάνει το σύστημα να λύσει το πρόβλημα με τον καλύτερο δυνατό τρόπο. Το σύστημα αυτό καταφέρνει να φέρει καλύτερα αποτελέσματα σε όλα τα dataset [10], [11], [12], [13], [14].

Το πρώτο και πολύ σημαντικό είναι η κωδικοποίηση του προβλήματος, ώστε να μπορεί να γίνει επεξεργασία από τους EA. Ο στόχος της κωδικοποίησης είναι η αναπαράσταση του προβλήματος ως μια «επίπεδη» δομή. Όπου τα στοιχεία της δομής έχουν ίση ποιοτική αξία. Στο συγκεκριμένο πρόβλημα, πρέπει να γίνει κωδικοποίηση του νευρωνικού δικτύου. Επιλέξαμε τον πίνακα ως δομή κωδικοποίησης. Γιατί είναι πολύ βολική λόγω τις εύκολης πρόσβασης στα στοιχεία του πίνακα από τους EA. Έτσι ένα μονοδιάστατος πίνακας περιγράφει όλο το νευρωνικό δίκτυο. Έχει δηλαδή όλη την πληροφορία που χρειαζόμαστε για την κατασκευή και χρήση του νευρωνικού δικτύου. Λόγω ότι τα δίκτυα που χρησιμοποιούμε είναι πλήρη συνδεδεμένα, δηλαδή δεν γίνεται να λείπουν συνδέσεις μεταξύ νευρώνων, άρα η μόνη πληροφορία που πρέπει να κρατάει ο πίνακας είναι οι τιμές από τα βάρη των νευρώνων καθώς και την τιμή bias του κάθε νευρώνα. Έτσι οι EA έχοντας έναν τέτοιο πίνακα μπορούν να εφαρμόσουν πάνω του όλες τις μεθόδους τους χωρίς να υπάρχει κάποιο πρόβλημα και χωρίς να παράγεται μη έγκυρο δίκτυο.

**Αρχικό δίκτυο**



**Κωδικοποιημένο δίκτυο**

Θέση	P0	P1	P2	P3	P4	P5	P6	P7	P8
Τιμή	0.3	0.7	0.1	0.4	0.9	0.5	0.2	0.7	0.3

Σχήμα 5 Παράδειγμα κωδικοποίησης νευρωνικού δικτύου σε μονοδιάστατο πίνακα

Αναλυτικά για τους ΕΑ, με βάση τον (Αλγόριθμο 1), έχουν χρησιμοποιηθεί οι παρακάτω μέθοδοι:

- Για την επιλογή, η χρήση της μεθόδου tournament, έχει δείξει πολύ καλύτερα αποτελέσματα, σε σχέση με την roulette. Φαίνεται να βοηθά στην πιο γρήγορη ανάπτυξη του πληθυσμού, ταυτόχρονα να μπορεί να κρατήσει σε ισορροπία τον πληθυσμό σε περίπτωση μεγάλης πιθανότητας mutation. Στη roulette ορίζετε ένα ποσοστό επιλογής για το κάθε ένα άτομο με βάση το fitness του. Έτσι με την παραγωγή ενός τυχαίου αριθμού επιλέγετε το άτομο που θα μπει στο σύνολο των επιλεγμένων. Η διαδικασία αυτή

επαναλαμβάνετε έως ότου φτάσουμε στον επιθυμητό πλήθος ατόμων. Με αυτή την μέθοδο δίνετε πολύ μεγάλη πιθανότητα επιλογής στα καλά άτομα αφήνοντας τα χειρότερα με ελάχιστες πιθανότητες επιλογής. Ο λόγος που θέλουμε να μπορούν να επιλεχθούν και τα χειρότερα είναι γιατί μπορεί να περιέχουν γενετικό υλικό που θα είναι πολύ χρήσιμο για την ανάπτυξη του συστήματος.

Η tournament έχει μόνο μια παράμετρο, που ελέγχει πόσο θα είναι το πλήθος των ατόμων που θα ανταγωνιστούν μεταξύ τους. Με αυτόν τον τρόπο μπορείς να ελέγξεις, πόσο μεγάλη ή μικρή πιθανότητα έχουν να επιλεχθούν, τα όχι και τόσο καλά άτομα του πληθυσμού. Υπάρχει περίπτωση σε κάποια προβλήματα, να μην αρκεί το fitness για να μπορέσει να κριθεί ποιος πρέπει να επιλεχθεί, λόγω της γενικότητας του fitness. Μία λύση είναι η δημιουργία μιας επιπλέον τιμής για κάθε άτομο. Έτσι δεν χρειάζεται να πειράξουμε το πραγματικό fitness, ο υπολογισμός της γίνεται σε κάθε εποχή αμέσως μετά τον υπολογισμό του fitness. Την τιμή αυτή την ονομάσαμε percent of fitness γιατί είναι μια τιμή σε ποσοστό με βάση το fitness. Αυτό δίνει την δυνατότητα δημιουργίας πολλών μεθόδων, υπολογισμού της. Η μέθοδος επιλογής μπορεί να βλέπει σε αυτή την τιμή αντί του fitness, καθώς έτσι παραμένει σταθερή η λογική του υπολογισμού του fitness, αλλά παράλληλα μπορούμε να επιλέγουμε άτομα με διαφορετική στρατηγική.

Προτείνουμε τρεις μεθόδους, υπολογισμού αυτής της τιμής:

1. Την πρώτη μέθοδο την ονομάσαμε rank. Γιατί ουσιαστικά, το κάθε άτομο θα πάρει την τιμή του με βάση τον χειρότερο και τον καλύτερο του τρέχον πληθυσμού. Θεωρούμε τον χειρότερο ότι έχει τιμή 0% και τον καλύτερο 100%. Ο τύπος που το περιγράφει είναι:

$$Percent = 100 \cdot \frac{c - min}{max - min} \quad (3.1)$$

- **c**: Το fitness του ατόμου
- **min**: Το μικρότερο fitness ατόμου μέσα στο πληθυσμό
- **max**: Το μεγαλύτερο fitness ατόμου μέσα στο πληθυσμό

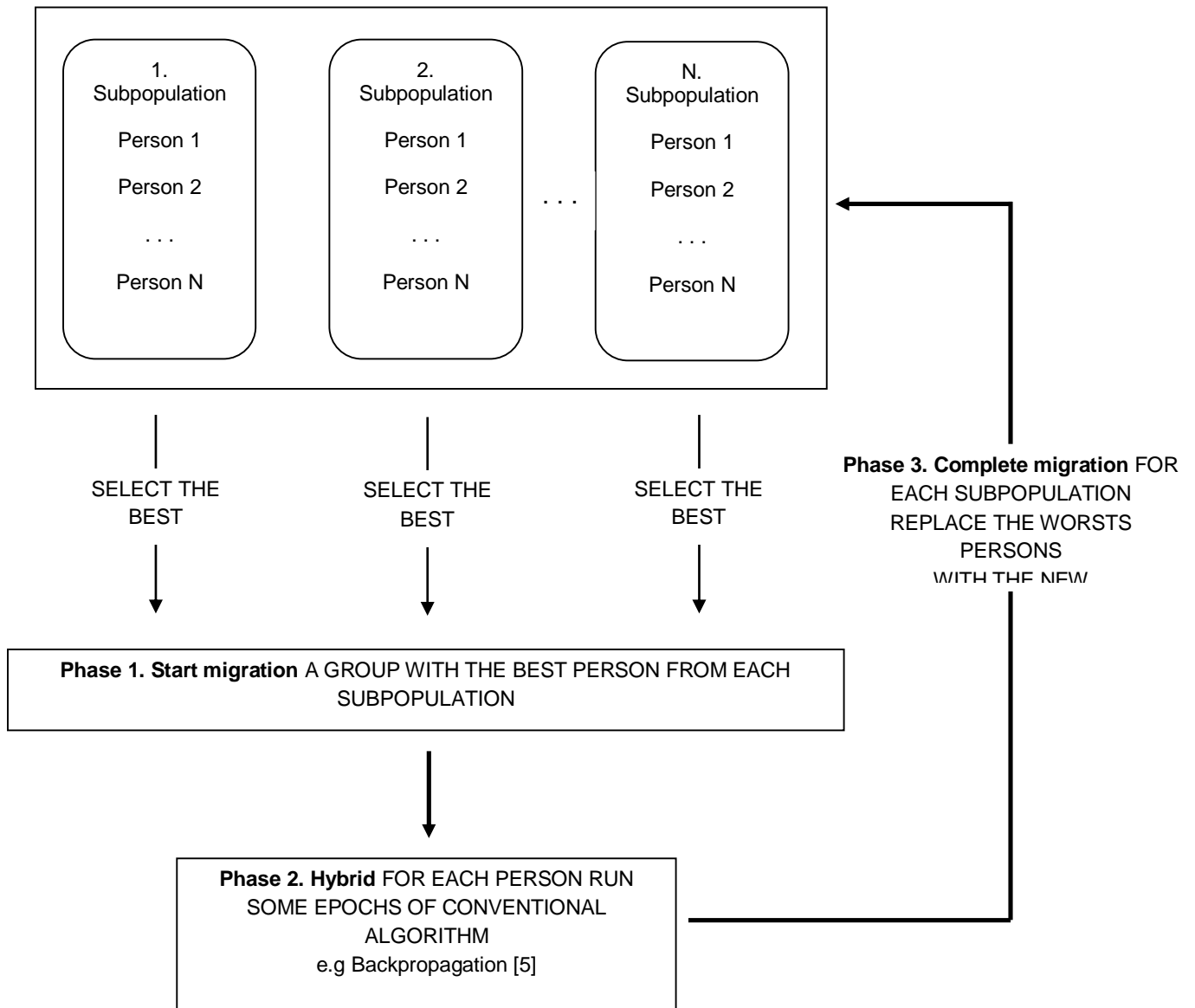
2. Η current best, υπολογίζει το percent, με βάση το fitness του καλύτερου ατόμου του τρέχον πληθυσμού (current, best person fitness). Έτσι, το εύρος του ποσοστού είναι [0, current, best person fitness]. Άρα κάποιος για να πάρει τιμή 0% θα πρέπει

να έχει και 0 fitness. Είναι λιγότερη αυστηρή με τα όχι και τόσο καλά άτομα του πληθυσμού, σε σχέση με την μέθοδο rank.

3. Η τελευταία για να μπορεί να λειτουργήσει, χρειάζεται να ξέρουμε την μέγιστη τιμή που μπορεί να πάρει το fitness (total best fitness), γενικά στο πρόβλημα μας. Το οποίο δεν είναι μια πληροφορία που μπορούμε να την έχουμε πάντα. Το εύρος του ποσοστού είναι  $[0, \text{total best fitness}]$ , η χρήση της είναι στα προβλήματα που δεν μπορούμε ή δεν έχουμε σκεφτεί κάποιο πολύ καλό τρόπο για τον υπολογισμό του fitness, έτσι πρέπει να είμαστε πολύ φιλικοί με τα άτομα που δεν έχουν τόσο καλό fitness.
- Για την μέθοδο του ανασυνδυασμού, χρησιμοποιείτε το crossover, με δύο μόνο γονείς, που παράγουν δύο απογόνους. Η παράμετρος σε αυτή την μέθοδο, ορίζει ανά πόσα γονίδια θα γίνεται το crossover. Όσο μικρότερος είναι ο αριθμός της παραμέτρου, τόσο πιο καλή πλέξη θα γίνει στα γονίδια. Η παράμετρος βοηθά, στον καλύτερο έλεγχο της ανάπτυξης.
  - Στη μέθοδο του mutation, ορίζετε πόσες πιθανότητες έχει το κάθε γονίδιο να μεταλλαχθεί. Εδώ κάναμε δύο παραμετροποιήσεις. Όταν είναι να γίνει το mutation, δεν παίρνει απλά μια τυχαία τιμή μέσα από ένα εύρος, αλλά προστίθεται η αφαιρείται μια τιμή από την παλιά. Αυτό κάνει την ανάπτυξη, πιο ομαλή, χωρίς να τυχαίνει να σκοτώνονται τα καλά άτομα λόγω μεγάλης αλλαγής τιμής. Επίσης όσο περνάνε οι εποχές τόσο αυξάνετε η πιθανότητα του mutation. Έτσι αφήνουμε στην αρχή τον πληθυσμό να αναπτυχθεί από μόνος του και στη συνέχεια που θα “κολλά” σε τοπικά βέλτιστα, τον βοηθάμε να προχωρήσει με την υψηλή πιθανότητα mutation. Η αλλαγή της πιθανότητας με βάση τις εποχές γίνεται με την γραμμική συνάρτηση. Σε αυτό το κομμάτι, μπορεί να χρησιμοποιήσει κάποιος όποια συνάρτηση θέλει. Αλλά από πειράματα που κάναμε δεν υπήρχε μεγάλη αλλαγή στα αποτελέσματα, με πιο περίπλοκες συναρτήσεις από την γραμμική.
  - Μία ακόμη σημαντική μέθοδος για την ανάπτυξη καθώς και για τα τελικά αποτελέσματα είναι η παράλληλη ανάπτυξη με υποπληθυσμού. Αυτό το πετύχαμε ξεκινώντας διαφορετικά threads για κάθε ένα πληθυσμό. Το θετικό είναι ότι έτσι μπορούμε να υλοποιήσουμε μια μέθοδο που λέγεται migration. Ανά κάποιες εποχές, δημιουργείτε ένα σύνολο, από το καλύτερα άτομα, από κάθε υποπληθυσμό. Αυτό το σύνολο είναι που θα αντικαταστήσει, τα χειρότερα άτομα, από κάθε υποπληθυσμό. Έτσι η συνολική απόδοση και ανάπτυξη αυξάνετε κατά πολύ. Γιατί μπορεί κάποιος υποπληθυσμός να μην έχει αρκετά καλά άτομα, αλλά με το migration αυτό διορθώνεται. Το αρνητικό με αυτή τη μέθοδο είναι όταν δοθεί στο σύστημα να δημιουργήσει πολλούς υποπληθυσμούς, επομένως και πολλά threads. Όταν ο επεξεργαστής του υπολογιστή δεν μπορεί να

## Προτεινόμενη αρχιτεκτονική για επίλυση προβλημάτων χρονοσειρών

υποστηρίζει τόσα πολλά threads, αυτό θα οδηγήσει να ολοκληρωθεί σε περισσότερο χρόνο η εκτέλεση, παρόλο που είναι παράλληλη εκτέλεση λόγω των threads.



Σχήμα 6 Σύστημα με χρήση των μεθόδων migration και hybrid

### 4.1.2 Οι μέθοδοι Hybrid και Migration

Και από το δικό μας αλλά και από το σύστημα τους [8] χρησιμοποιούνται hybrid EA. Όταν ανά κάποιες εποχές, εκτελείται συμβατικός αλγόριθμος, για την καλύτερη ανάπτυξη τότε οι EA λέγονται hybrid [15]. Στο NE, ο συμβατικός αλγόριθμος, είναι μια μέθοδος βελτιστοποίησης για νευρωνικά δίκτυα. Στην περίπτωση μας είναι το BP [5]. Ανά κάποιο συγκεκριμένο πλήθος εποχών,

τρέχουν κάποιες εποχές BP. Στο σύστημα μας αυτό γίνεται αμέσως πριν ολοκληρωθεί το migration. Συγκεκριμένα αφού φτιαχτεί το σύνολο που προορίζεται για migration, αυτά τα άτομα του συνόλου, τρέχουν κάποιες εποχές BP. Χωρίς το hybrid το σύστημα μας έβγαζε καλά αποτελέσματα, αλλά δεν μπορούσε να ξεπεράσει τα αποτελέσματα από κάποια προβλήματα της υπάρχουσας μελέτης [8]. Αλλά με την χρήση του hybrid μπόρεσε και σε αυτά τα προβλήματα, να ξεπεράσει τα καλύτερα αποτελέσματα του συστήματος τους [8].

Συγκεκριμένα, για το BP, έγινε η χρήση της απλής μορφής του [5]. Η διαφορά είναι, στη παράμετρο του BP, το learning rate. Στο σύστημα μας είναι μεταβλητό σε σχέση με την εποχή. Έτσι ξεκινά αρχικά με ένα σχετικά μεγάλο, learning rate και έπειτα, καθώς περνάνε οι εποχές μικραίνει. Αυτό γίνεται γιατί, στις αρχικές εποχές, δεν μας πειράζει να γίνονται μεγάλες αλλαγές, «βήματα» στα βάρη των νευρωνικών δικτύων. Αντιθέτως αυτό βοηθά στη πιο γρήγορη ανάπτυξη. Καθώς περνάνε οι εποχές και ταυτόχρονα πλησιάζουμε στη βέλτιστη λύση, θέλουμε να γίνονται μικρές αλλαγές, ώστε να μπορεί το σύστημα να πλησιάσει όσο πιο πολύ γίνεται στην βέλτιστη λύση.

#### 4.1.3 Μεταβλητό πλήθος γονιδίων

Ένα ακόμη χαρακτηριστικό του συστήματος μας, είναι η δυνατότητα ανάπτυξης των EA, με μεταβλητό νευρωνικό δίκτυο. Δεδομένου δυο παραμέτρων, όπου η μία ελέγχει το εύρος του πλήθους των στρωμάτων και η άλλη το εύρος του πλήθους των νευρώνων, ανά στρώμα. Οι EA στη συνέχεια, στη φάση δημιουργίας, του πρώτου πληθυσμού, φτιάχνει διαφορετικά τυχαία νευρωνικά δίκτυα, με βάση τις δύο προηγούμενες παραμέτρους. Αυτό δημιουργεί άτομα με μεταβλητό πλήθος γονιδίων, το οποίο συνεπάγεται ότι χρειάζεται μια διαφορετική μέθοδος για τον ανασυνδυασμό και για την μετάλλαξη, που απλά να υποστηρίζει άτομα με μεταβλητό πλήθος γονιδίων. Υπάρχουν πολλές υλοποιήσεις, για EA με μεταβλητό πλήθος γονιδίων. Τα προβλήματα που αντιμετωπίσαμε είναι ότι αυτές οι υλοποιήσεις προϋποθέτουν, να μπορεί στο άτομο να προστεθούν ή να αφαιρεθούν, τυχαίο N πλήθος γονιδίων, καθώς και άλλων πράξεων όπως η μεταφορά ενός πλήθους γονιδίων από μία θέση σε μία άλλη, στη φάση της μετάλλαξης του πληθυσμού [16]. Αυτό είναι αδύνατο για το συγκεκριμένο σύστημα, δεδομένου ότι ένα άτομο είναι ένα νευρωνικό δίκτυο και τα γονίδια του είναι όλα τα βάρη του δικτύου, τότε το μόνο που μπορούμε να κάνουμε είναι να προσθέτουμε ή να αφαιρούμε νευρώνες, μόνο στα κρυφά στρώματα του δικτύου. Άρα η παραπάνω μέθοδος μετάλλαξης μπορεί να εφαρμοστεί στο σύστημα μας αλλά με κάποιους περιορισμούς. Επίσης, η μεταφορά N γονιδίων και ως θεωρήσουμε ότι τα N γονίδια είναι ένας νευρώνας, δεν μπορούμε να μεταφέρουμε αυτόν τον νευρώνα σε άλλο στρώμα με ασφάλεια. Καθώς οι νευρώνες του στρώματος που προορίζεται να πάει, μπορεί να μην

έχουν το ίδιο πλήθος εισόδων, με αυτόν που μεταφέρουμε, ώστε και μετά την μεταφορά να παραμείνει ένα έγκυρο δίκτυο.

Ο ανασυνδυασμός που προτείνουμε και έχουμε υλοποιήσει στο σύστημα, αποτελείται από ένα τύπο, ο οποίος είναι ένας χάρτης ανάμεσα σε δύο μονοδιάστατους πίνακες A και B, που έχουν διαφορετικό μέγεθος, ο ένας από τον άλλον. Ο κάθε πίνακας είναι ένα άτομο και το κάθε στοιχείο του πίνακα είναι ένα γονίδιο. Ο χάρτης αυτός δείχνει, δεδομένου μίας θέσης στον A, ποια θα πρέπει να είναι η αντίστοιχη θέση στον πίνακα B. Βέβαια, δεν μπορεί να υπάρξει πλήρης αντιστοίχιση, ανάμεσα σε όλες τις θέσεις και τον δύο πινάκων. Αυτός ο τύπος μπορεί να συνδυαστεί εύκολα με τις προηγούμενες σταθερού μεγέθους μεθόδους ανασυνδυασμού.

Αν θέλουμε να πάρουμε από μίας θέση του A την θέση του B τότε, η συνάρτηση είναι:

$$P_B = \frac{P_A \cdot L_B}{L_A} \quad (3.2)$$

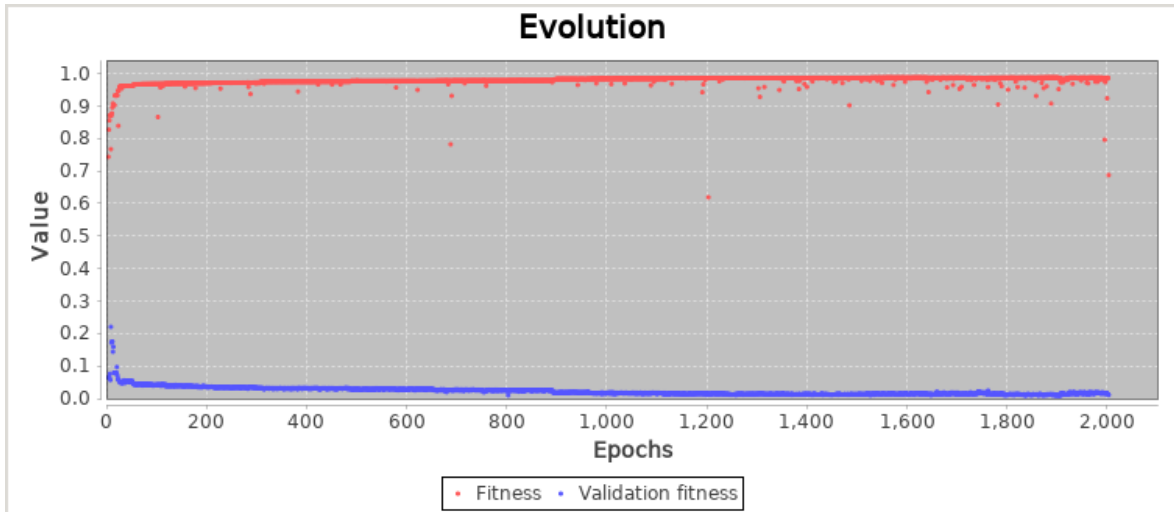
Όπου:

- $P_B$ : η θέση του B που ψάχνουμε
- $P_A$ : η θέση του A που ξέρουμε
- $L_B$ : το μέγεθος του πίνακα B
- $L_A$ : το μέγεθος του πίνακα A

## 4.2 Overtraining

Ένα από τα σημαντικότερα προβλήματα με τα συμβατικά νευρωνικά δίκτυα, είναι το overtraining. Αυτό σημαίνει, ότι χάνει το δίκτυο την δυνατότητα της γενίκευσης και ειδικεύεται μόνο σε κάποιες από τις περιπτώσεις. Είναι μια καταστροφική περίπτωση εκπαίδευσης, που λύνεται συνήθως ορίζοντας ένα σύνολο, από τα δεδομένα, που ονομάζεται validation set. Το σύνολο αυτό, χρησιμοποιείται από το σύστημα κατά την διάρκεια της εκπαίδευσης, όχι για την εκπαίδευση του δικτύου, αλλά για να μπορεί να παρακολουθεί άμα χάνει την γενίκευση του. Το σύστημα μας το υποστηρίζει αυτό, καθώς παρέχει την δυνατότητα να διακοπεί η εκπαίδευση ή να τερματίσει κανονικά ολοκληρώνοντας όλες τις εποχές του και να επιλεγθεί το καλύτερο άτομο από όλες τις εποχές με βάση το validation set. Τα πειράματα που κάναμε δείχνουν ότι γενικά το σύστημα πετυχαίνει το καλύτερο validation error πολύ κοντά στο τέλος των εποχών. Αυτό δείχνει ότι το σύστημα δεν χάνει εύκολα τη γενίκευση και η χρήση του validation set, είναι αναγκαία σε λίγα

μόνο από τα προβλήματα.



Σχήμα 7 Στο διάγραμμα φαίνεται, ότι το validation fitness μειώνεται με το πέρασμα των εποχών. Αυτό δείχνει τη συνεχή βελτίωση του δικτύου, χωρίς να χάνει την γενίκευση του.

Η κατάρθρωση του συστήματος και η επιλογή των μεθόδων είναι μια πολύ σημαντική αλλά και χρονοβόρα διαδικασία, καθώς χρειάζονται επανειλημμένες δοκιμές με την κάθε μια μέθοδο ξεχωριστά και με διαφορετικές παραμέτρους. Όστε να επιλεγθούν αυτές που παράγουν τα καλύτερα αποτελέσματα.

## ΚΕΦΑΛΑΙΟ 5 – Περιβάλλον και εργαλεία ανάπτυξης

Σε αυτό το κεφάλαιο θα μιλήσουμε για την γλώσσα προγραμματισμού και βιβλιοθήκες καθώς και το περιβάλλον ανάπτυξης που χρησιμοποιήθηκαν για την κατασκευή του συστήματος, καθώς επίσης και τους λόγους που επιλέχθηκαν, μαζί με τις εναλλακτικές τους.

### 5.1 Επιλογή γλώσσας προγραμματισμού

Για κάθε πρόβλημα που θέλουμε να επιλύσουμε ή project να υλοποιήσουμε, πρέπει να γίνει μια σκέψη για την επιλογή της γλώσσας προγραμματισμού. Είναι μια πολύ σημαντική απόφαση που δεν πρέπει να την αφήσουμε στην τύχη.

Διότι δεν υπάρχει η καλύτερη γλώσσα προγραμματισμού, άμα υπήρχε δεν θα είχαμε μια τεράστια παλέτα από γλώσσες. Η καλύτερη γλώσσα είναι αυτή που θα μας βοηθήσει να επιλύσουμε το πρόβλημα μας, με τον καλύτερο και πιο εύκολο τρόπο. Για αυτόν τον λόγο πρέπει να γίνει ανάλυση των απαιτήσεων του προβλήματος πριν προβούμε στην επιλογή της.

Η γλώσσα προγραμματισμού που επιλέχθηκε είναι η Java. Η Java είναι μια γλώσσα γενικού σκοπού, δηλαδή δεν ειδικεύεται σε κάποιο συγκεκριμένο είδος προβλημάτων. Έτσι υπάρχει ένας τεράστιος όγκος από βιβλιοθήκες που μπορεί κάποιος να χρησιμοποιήσει για να επιλύσει το πρόβλημα του με τον πιο αποδοτικό τρόπο.

Τα μεγαλύτερα πλεονεκτήματα αυτής της γλώσσας σε σχέση με τις υπόλοιπες του ίδιου είδους είναι:

- **Αυστηρό συντακτικό**, που αναγκάζει τον χρήστη της, να δομήσει το πρόβλημα και σε επίπεδο του κώδικα αλλά και στο μυαλό του ώστε να έχει μια πιο πλήρη και σωστή εικόνα της υλοποίησής του.
- **Αυστηρά δομημένα**, λόγω της πλήρους αντικειμενοστρέφειας της είναι τρομερά δομημένα, στον βαθμό που κάποιος θέλει να την αξιοποιήσει σωστά, τον αναγκάζει να γράψει δομημένα. Η δόμηση βοηθά στην ολοκληρωτική εικόνα του προβλήματος, στην πιο εύκολη οργάνωση και κατανόηση του κώδικα και στην πιο εύκολη μελλοντική συντήρηση κώδικα του.
- **Μεταφερσιμότητα**, είναι η έννοια που περιγράφει την δυνατότητα μια γλώσσας να εκτελείται χωρίς καμία επέμβαση από τον προγραμματιστή και επιπλέον υλοποίηση διαφορετικού προγράμματος, ώστε να μπορεί να εκτελεστεί σωστά σε οποιοδήποτε υλικό

και λειτουργικό σύστημα.

- **Υψηλού επιπέδου** γλώσσα προγραμματισμού. Που σημαίνει ότι διαχειρίζεται σε μεγάλο βαθμό διάφορες προγραμματιστικές ρουτίνες που σε αντίθετη περίπτωση θα έπρεπε να υλοποιηθούν. Κάποια από αυτά είναι ο garbage collector, μιας και είναι αντικειμενοστραφής γλώσσα, χρειάζεται κάποιο μηχανισμό για να αποδεσμεύονται τα πλέον αχρείαστα αντικείμενα από την κύρια μνήμη, αυτόματος έλεγχος για τα όρια των πινάκων και άλλα πολλά. Έτσι αφήνει τον προγραμματιστή να συγκεντρωθεί περισσότερο στην οργάνωση και δομή και κατανόηση του προβλήματος.
- **Πολλές βιβλιοθήκες Artificial intelligence (AI)**, όπως για νευρωνικά δίκτυα Neuroph, Deeplearning4j, επεξεργασίας φυσικής γλώσσας, Apache OpenNLP, Stanford CoreNLP, μηχανική μάθησης Java-ML, RapidMiner, Weka, Deep Java Library, για γενετικούς αλγορίθμους Jenetics, Watchmaker Framework, ECJ 23, Java Genetic Algorithms Package (JGAP) και πάρα πολλά ακόμη.



Σχήμα 8 Το όνομα της γλώσσας πάρθηκε από το όνομα της καφετέριας που συναντιόνταν οι δημιουργοί της. Εξού και το logo της.

Για το συγκεκριμένο πρόβλημα έγινε αυτή η επιλογή γιατί θέλαμε να γίνει έρευνα πάνω σε κάποιο νέο σύστημα και όχι σε κάποιο έτοιμο. Καθώς και να φτιαχτεί ένα mini framework για τους εξελικτικούς αλγορίθμους ώστε να είναι επαναχρησιμοποιήσιμο και για άλλα project και προβλήματα. Όπως είναι κατανοητό τα θετικά της Java βοηθάνε πάρα πολύ για στη δημιουργία ενός framework ή library.

Η πιο συνηθισμένη επιλογή για μηχανική μάθηση και γενικότερα για προβλήματα AI, είναι η script γλώσσα Python.

Και αυτή με την σειρά της είναι μια γλώσσα προγραμματισμού γενικού σκοπού και με τεράστιο όγκο από βιβλιοθήκες για πάρα πολλά διαφορετικά προβλήματα, όπως για μαθηματικά,

διαγράμματα, γραφικά περιβάλλοντα, backend server και φυσικά μηχανική μάθηση, χρήση Cuda για επιτάχυνση εκπαίδευσης αλγορίθμων μηχανικής μάθησης, εξελικτικούς αλγορίθμους και πάρα πολλά ακόμη. Όντως όμως script γλώσσα υστερεί πολύ στην δομή και «αυστηρότητα» στον προγραμματισμό, που την καθιστά ακατάλληλη για πολύ μεγάλα project. Όχι ότι δεν μπορεί κάποιος να την χρησιμοποιήσει κατά αυτόν τον τρόπο, απλά δεν θα έχει κάποιο όφελος από το συντακτικό της ή την γενική νοοτροπία τις γλώσσας.

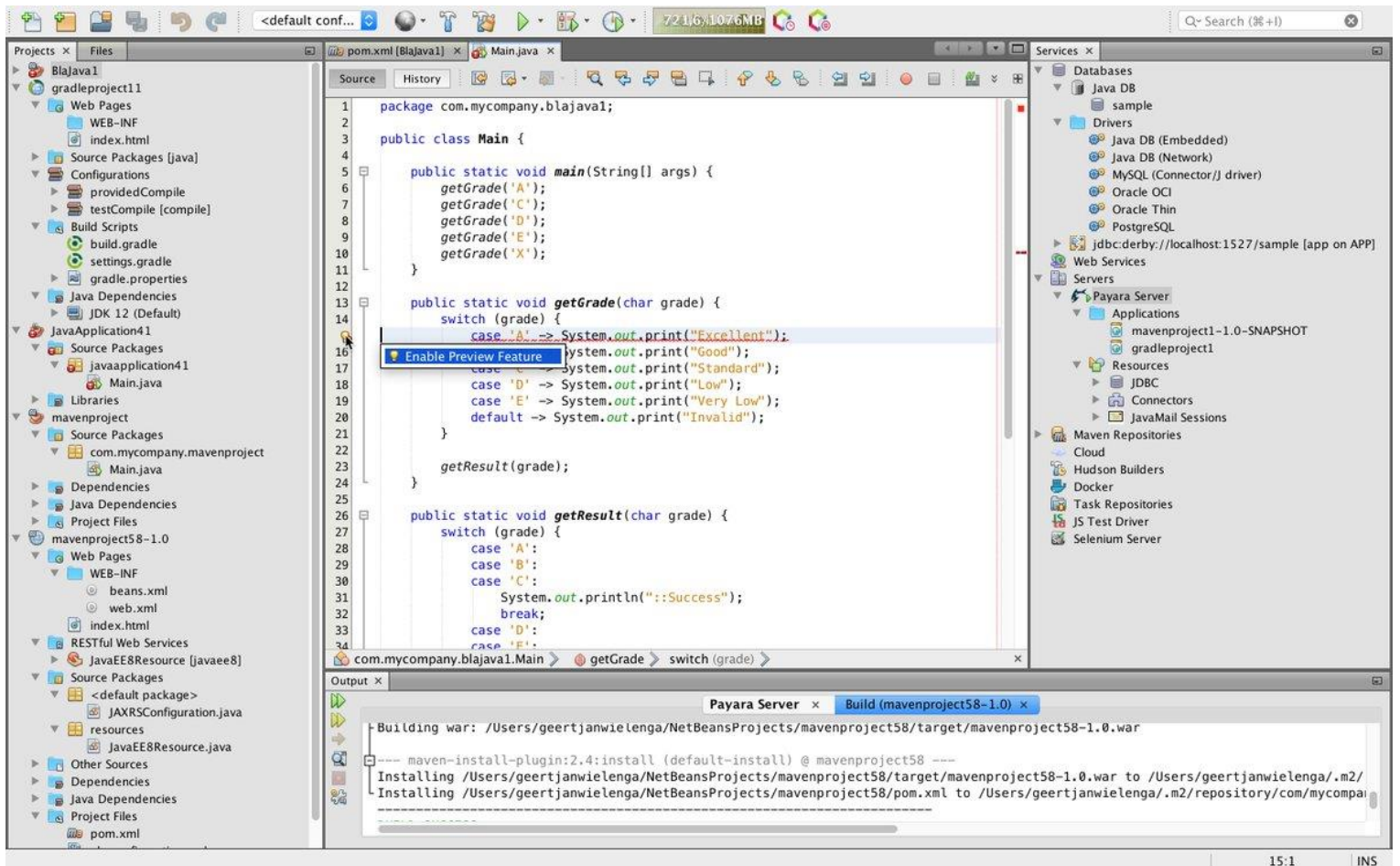
Ο λόγος που έχει κερδίσει τόσο πολύ έδαφος είναι για την βιβλιοθήκη μηχανικής μάθησης Tensorflow που την υποστηρίζει ενεργά η εταιρεία Google, καθώς και η βιβλιοθήκη Keras που και αυτή με την σειρά της έχει πολύ μεγάλη γκάμα από εργαλεία. Για το κομμάτι του NE υπάρχει η βιβλιοθήκη NEAT που είναι έχει αποδείξει έμπρακτα την αποτελεσματικότητά της.

## 5.2 Περιβάλλον ανάπτυξης

Όπως και με την γλώσσα προγραμματισμού είναι πολύ σημαντικό και η επιλογή ενός καλού περιβάλλοντος ανάπτυξης. Για την πιο εύκολη και γρήγορη υλοποίηση του εκάστοτε project.

Στο συγκεκριμένο έγινε η χρήση του Apache Netbeans, είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης η αλλιώς Integrated Development Environment IDE. Είναι αναπτυγμένο στην γλώσσα Java και υποστηρίζει μια πληθώρα από γλώσσες προγραμματισμού για πάρα πολλά διαφορετικά project. Όπως για Web εφαρμογές, desktop, scripts και άλλα πολλά. Είναι πλήρως δωρεάν και με όλες τις λειτουργίες και δυνατότητες του και open source με άδεια Apache License 2.0. Έχει την δυνατότητα εύκολης εγκατάστασης επεκτάσεων ώστε να μπορεί ο χρήστης του να αυξήσει ακόμη περισσότερο τις δυνατότητες του IDE και να το προσαρμόσει στις ανάγκες του. Όσο αφορά την Java, διαθέτει γραφικό περιβάλλον για την ανάπτυξη, γραφικού περιβάλλοντος χρήση, Graphical User Interface (GUI), που υλοποιήσει τον κώδικα αυτόματα και αφήνει τον προγραμματιστή να συγκεντρωθεί στην σωστή κατασκευή του γραφικού παρά στην λεπτομέρεια και επαναληπτικότητα του κώδικα. Επιπλέον βρίσκει όλα τα συντακτικά λάθη αλλά και τα πιο συνηθισμένα λογικά λάθη και προτείνει λύσεις για την επίλυση τους. Επίσης έχει γραφικό περιβάλλον για version control εργαλεία όπως το GIT και άλλα γνωστά.

## ΚΕΦΑΛΑΙΟ 5



Σχήμα 9 Στιγμιότυπο από το γραφικό περιβάλλον του Apache Netbeans, με το project tree αριστερά και τις βάσεις και άλλα εργαλεία δεξιά

Ένα ακόμη πολύ δημοφιλές IDE είναι της εταιρείας JetBrains, η οποία έχει κατασκευάσει μια τεράστια σουίτα από IDE, σχεδόν για όλες τις γλώσσες προγραμματισμού αλλά για πιο συγκεκριμένα project. Όπως για web frontend, backend, desktop, Android και άλλα πολλά. Έχει πάρα πολλές δυνατότητες, από αυτόματη συμπλήρωση και βοήθεια στο συντακτικό της εκάστοτε γλώσσας μέχρι και διόρθωση ορθογραφικών. Έχει γραφικό περιβάλλον για version control εργαλεία όπως το GIT και άλλα γνωστά, με επιπλέον λειτουργίες από τις προκαθορισμένες.

Το αρνητικό είναι ότι δωρεάν κάτω από Apache 2 License, είναι μια πολύ απλή και βασική έκδοση του IDE, με αρκετά λιγότερες δυνατότητες και χωρίς να ειδικεύεται όπως οι υπόλοιπες εκδόσεις. Όλα τα υπόλοιπα είναι με συνδρομή που είναι αρκετά ακριβή. Το Android Studio είναι δωρεάν.

### 5.3 Γραφικό περιβάλλον

Για την πιο καλή αντίληψη και προβολή των αποτελεσμάτων καθώς και απόδοσης του συστήματος κατασκευάστηκε ένα γραφικό περιβάλλον στη Java Swing.



Σχήμα 10 Το γραφικό περιβάλλον του συστήματος. Ο πίνακας με τα αποτελέσματα και τα διαγράμματα.

Οι πιο χρήσιμες πληροφορίες για τις εκτελέσεις φαίνονται στο γραφικό περιβάλλον. Όπως είναι ο πίνακας με τα αποτελέσματα, όπου περιγράφει τους ακριβείς αριθμούς, evaluation error, validation error, training error και το σχήμα του νευρωνικού δικτύου. Από κάτω του υπάρχουν δυο κουμπιά για την περιήγηση ανάμεσα στα πολλαπλά αποτελέσματα για το κάθε ένα thread. Στο διάγραμμα του prediction που βρίσκετε πάνω δεξιά, προβάλλονται τα πραγματικά δεδομένα μαζί με αυτά της πρόβλεψης, ώστε και να παρατηρηθεί η φύση των δεδομένων του dataset, καθώς και σε ποια σημεία το σύστημα τα καταφέρει καλύτερα και σε ποια όχι. Από κάτω του ακριβώς είναι το διάγραμμα του σφάλματος, για κάθε ένα feature που υπάρχει στο prediction διάγραμμα.

Έτσι με μεγαλύτερη ακρίβεια μπορεί να γίνει η παρακολούθηση του σφάλματος και η διακύμανση του με βάση τα δεδομένα. Κάτω αριστερά είναι το διάγραμμα της εξέλιξης. Αφορά αποκλειστικά του EA και την ανάπτυξη τους. Στον άξονα του X στο διάγραμμα είναι οι εποχές και στο Y με κόκκινο

είναι το fitness. Από εδώ μπορούμε να παρακολουθήσουμε την ανάπτυξη των πληθυσμών και να βγάλουμε συμπεράσματα για το πόσο καλή είναι. Με βάση την καμπύλη του διαγράμματος, άμα είναι απότομη η όχι και άμα έχουμε μεγάλη πιθανότητα μετάλλαξης. Επειδή το διάγραμμα αυτό φτιάχνετε από το fitness του καλύτερου ατόμου αλλά για την κάθε εποχή ξεχωριστά, μπορούμε να διακρίνουμε άμα έχουμε πολλές απότομες πτώσεις των καλύτερων ατόμων, βλέποντας τις κουκίδες που ξεφεύγουν πολύ απότομα προς τα κάτω. Άμα αυτές οι κουκίδες είναι πάρα πολλές τότε η πιθανότητα μετάλλαξης είναι πολύ μεγάλη. Επιπροσθέτως στο διάγραμμα αυτό με μπλε χρώμα είναι το validation error. Πιο αναλυτικά, είναι το fitness του καλύτερου ατόμου στην τρέχον εποχή, υπολογισμένο όμως με βάση το validation dataset. Από το validation error, μπορούμε να διακρίνουμε άμα το σύστημα μας έχει πάθει overtraining και αυτό είναι ορατό όταν το validation error ενώ μειωνόταν ξεκινάει και αυξάνεται. Τέλος υπάρχει ένα βοηθητικό progress bar για να μπορούμε να καταλάβουμε σε ποια στιγμή την ανάπτυξης βρίσκεται το σύστημα, μαζί με έναν μετρητή που δείχνει σε ποια εκτέλεση βρισκόμαστε.

Για την κατασκευή των διαγραμμάτων, έγινε χρήση της βιβλιοθήκης JFreeChart . Είναι open source και δωρεάν για την χρήση της, κάτω από την άδεια GNU Lesser General Public License. Έχει πολλές δυνατότητες για κατασκευή όλων των γνωστών διαγραμμάτων καθώς και πιο σύνθετων, με μεγάλη δυνατότητα παραμετροποίησης.

Οι παραπάνω προτεινόμενοι τρόποι επιλογής εργαλείων μπορεί να διαφέρουν στην πραγματικότητα, γιατί υπάρχουν παράμετροι που είναι διαφορετικοί ανά άνθρωπο και project. Όπως ο χρόνος οι γνώσεις η εμπειρία και τα ιδιαίτερα χαρακτηριστικά του εκάστοτε project.

## ΚΕΦΑΛΑΙΟ – 6 Datasets και πειράματα

Με βάση το δικό μας προτεινόμενο σύστημα του κεφαλαίου 3, έγινε σύγκριση απόδοσης με αυτό της ήδη υπάρχουσας έρευνας. Σε αυτήν που είναι η πιο πρόσφατη τους έρευνα, χρησιμοποίησαν την μέθοδο MCNE που έχουν μιλήσει για αυτήν πιο αναλυτικά στην προηγούμενη του έρευνα. Το σύστημα μας σε μια γενική αρχιτεκτονική εικόνα του, έχει κάποια κοινά στοιχεία με το δικό τους MCNE. Στην πρόσφατη του έρευνα προτείνουν δύο ακόμη στρατηγικές, βελτιωμένες εκδόσεις της MCNE. Παρόλα αυτά το σύστημα μας κατάφερε να παράγει καλύτερα αποτελέσματα από όλες τις στρατηγικές τους.

### 6.1 Ανάλυση των datasets

Για την πιο σωστή σύγκριση του δικού μας συστήματος με του δικού τους, έγινε η χρήση των ίδιων dataset, καθώς και η μέθοδος υπολογισμού του σφάλματος είναι η ίδια.

Η επιλογή των datasets είναι πολύ σημαντική για τον σχηματισμό της τελική εικόνας του συστήματος. Έτσι στα πειράματα μας έγινε η χρήση πέντε dataset, που το κάθε ένα είναι από διαφορετική πηγή και με τελείως διαφορετικά δεδομένα. Αυτό δίνει την δυνατότητα πιο σωστού ελέγχου της αποτελεσματικότητας.

Τα dataset πιο συγκεκριμένα είναι τα εξής:

1. Sunspot [11]. Το μαγνητικό πεδίο του Ήλιου κάνει μια περιοδική κίνηση και κάθε περίοδος του διαρκεί περίπου 11 χρόνια. Στο τέλος κάθε περιόδου του, έχουν αλλάξει θέση οι πόλοι του μεταξύ τους. Έχει παρατηρηθεί ότι σχηματίζονται στην επιφάνεια του κηλίδες, σκοτεινά σημεία, που σχετίζονται με αυτή του την περιοδικότητά, λόγω της αλλαγής της πολικότητας τους. Το φαινόμενο αυτό επηρεάζει σημαντικά το ηλιακό μας σύστημα καθώς και το κλίμα στη Γη. Το σύστημα μας τίθεται να προβλέψει το πλήθος των κηλίδων κατά της περιόδους του. Έχει παρατηρηθεί ότι είναι δύσκολη η πρόβλεψη αυτή, καθώς υπάρχουν απρόσμενες αλλαγές, που επηρεάζουν το πλήθος των κηλίδων.
2. Mackey-Glass [13]. Αναφέρεται, σε συναρτήσεις που παράγουν ποικιλόμορφες κυματομορφές, που προφανώς έχουν απεριόριστες ή “χαοτικές” λύσεις. Τα αποτελέσματα αυτά συσχετίζονται με την δυναμική αναπνοή και τις αιματοποιητικές ασθένειες.

Ουσιαστικά, έχουν παρατηρηθεί κάποιες χρόνιες ή και οξείες ασθένειες που κάποια από τα συμπτώματα τους εμφανίζουν μια περιοδικότητα και προσπαθούν στη μελέτη αυτή να συσχετίσουν, μαθηματικές συναρτήσεις, έτσι ώστε να γίνει πρόβλεψη της περιοδικότητας των συμπτωμάτων.

3. Lazer [14]. Είναι δεδομένα που έχουν καταγραφεί με την χρήση ενός απομακρυσμένου υπέρυθρου laser, που παράχθηκε για χρήση benchmark, το οποίο έχει χρησιμοποιηθεί σε πολλούς διαγωνισμούς. Τα δεδομένα του είναι, χαμηλής διάστασης, μη γραμμικά και σταθερά time series.
4. Lorenz [12]. Έχει φτιαχτεί, με δεδομένα από παρατηρήσεις διάφορων, υδροδυναμικών ροών. Και αυτά με την σειρά τους, δείχνουν γενική περιοδικότητα, αλλά σε βάθος χρόνου φαίνεται η αστάθεια στα δεδομένα του. Είναι μια ακόμη φυσική παρατήρηση, άρα εμπεριέχει το στοιχείο του απρόβλεπτου.
5. Taiwan Trading Index Exchange (TWI Exchange) [10]. Απλή καταγραφή της συναλλαγματικής ισοτιμίας της Ταϊβάν μέσα σε ένα συγκεκριμένο χρονικό διάστημα. Το συγκεκριμένο dataset διαφέρει αρκετά από τα υπόλοιπα, καθώς οι τιμές τους κυμαίνονται σε πολύ μικρότερο εύρος.

Χρησιμοποιούμε την ήδη χωρισμένη μορφή των dataset, με 60/20/20 % δεδομένα, training/validation/testing αντίστοιχα, όπου είναι παρμένα από το δικό τους σύστημα [αναφορά link για τα dataset]. Τα dataset παρόλο που είναι πολύ διαφορετικά μεταξύ τους, έχουν κοινό χαρακτηριστικό ότι περιγράφουν μια περιοδικότητα. Αυτό το χαρακτηριστικό είναι που τα καθιστά προβλέψιμα από ένα νευρωνικό δίκτυο τύπου MLP. Σε αντίθετη περίπτωση θα γινόταν χρήση κάποιου δικτύου που λαμβάνει υπόψη του καλύτερα τις προηγούμενες τιμές ώστε να γίνει η πρόβλεψη. Η περιοδικότητα τους δεν είναι σταθερή πάντα και οι ενδιάμεσες καταστάσεις τους είναι απρόβλεπτες, για αυτό δεν αρκεί μια μαθηματική συνάρτηση για την περιγραφή και πρόβλεψη τις περιοδικότητας τους.

## 6.2 Κατάστρωση των πειραμάτων

Για να μπορέσει να υπάρξει σωστή σύγκριση των δύο συστημάτων, έπρεπε για την προσομοίωση χρησιμοποιηθούν τα δεδομένα από την έρευνα [8], [10], [11], [12], [13], [14], έτσι ακριβώς όπως είναι χωρισμένα στα υποσύνολα τους. Καθώς και η μέτρηση του σφάλματος γίνεται με την ίδια μέθοδο root mean square error (RMSE). Η ίδια μέθοδος είναι αυτή που καθοδηγεί, την ανάπτυξη

των υποπληθυσμών, έχοντας την θέση του fitness. Η επιλογή του καλύτερου ατόμου, από κάθε εκτέλεση, γίνεται με βάση το fitness του ατόμου, στο validation set.

Το κόστος  $C$  του κάθε πειράματος υπολογίζεται ως εξής:

$$C = \left( \sum^{n+1} m \cdot t \right) + \left( \sum^r z \right) \quad (5.1)$$

Όπου:

$n$ : οι εποχές του εξελικτικού αλγορίθμου

$m$ : το πλήθος των ατόμων του υποπληθυσμού

$t$ : το πλήθος των υποπληθυσμών

$r$ : πόσες φορές θα εκτελεστεί ο αλγόριθμος BP

$z$ : το πλήθος των εποχών που θα εκτελέσει κάθε φορά ο αλγόριθμος BP

Όπου το  $z$  υπολογίζεται:

$$z = p \cdot n \quad (5.2)$$

Και το  $r$ :

$$r = \frac{n}{z} \quad (5.3)$$

Το  $N$  είναι αυξημένο κατά ένα, για να ληφθεί υπόψη και ο υπολογισμός του fitness, κατά την αρχικοποίηση του συστήματος.

Έχει οριστεί μια παράμετρος που την ονομάζουμε “ποσοστό”  $p$ , η οποία ορίζει δύο πράγματα:

Κάθε πόσες εποχές, του εξελικτικού αλγορίθμου, θα εκτελείται ο αλγόριθμος BP.

Πόσες θα είναι οι εποχές του BP

Και το 1 και το 2 υπολογίζονται με τον ίδιο τρόπο:

$$p \cdot n \quad (5.4)$$

Ουσιαστικά το 2, είναι η μεταβλητή  $z$ .

Η απόφαση αυτή, για την μεταβλητή  $p$ , πάρθηκε ώστε όταν μεταβάλλεται η τιμή της, να παραμένει σε ισορροπία μεταξύ, του πόσο συχνά και για πόσες εποχές θα εκτελείται ο BP. Για παράδειγμα, άμα αυξηθεί η τιμή της τότε θα εκτελεστεί για περισσότερες εποχές ο BP, αλλά για λιγότερες φορές.

### 6.3 Παράμετροι των πειραμάτων

Τα πειράματα χωρίστηκαν σε τέσσερις τύπους. Τύποι έχουν κάποιες κοινές παράμετροι και διαφέρουν σε κάποιες συγκεκριμένες, όπου με βάση αυτές θα παραχθούν τα τελικά συμπεράσματα για την γενική εικόνα του συστήματος καθώς και για την αποδοτικότητα του. Και οι τέσσερις τύποι εφαρμόστηκαν, σε όλα τα dataset. Για κάθε ένα τύπο έγιναν τριάντα εκτελέσεις του συστήματος. Αυτό γίνεται γιατί από την φύση τους οι εξελικτικοί αλγόριθμοι, έχουν το στοιχείο του τυχαίου σε πολλά σημεία. Τα συμπεράσματα παράγονται, μέσα από τα στατιστικά από τις πολλές εκτελέσεις

Πίνακας 1 Κοινοί παράμετροι, για όλους τους τύπους πειραμάτων

Παράμετρος	Τιμή
Εποχές	2000
Μέγεθος υποπληθυσμού	15
Μέθοδος επιλογής	Tournament K=3 (μέγεθος του συνόλου tournament)
Μέθοδος ανασυνδυασμού	Random K=5 (μέγεθος του crossover break size)
Μέθοδος μετάλλαξης	Simple random (πρόσθεση/αφαίρεση ενός τυχαίου αριθμού)
Εύρος τιμών, για τον μεταβλητό, ρυθμό μάθησης <i>learning rate</i>	[0.3 , 0.00001]
Εύρος τιμών, για την πιθανότητα μετάλλαξης	[200, 80]
Εύρος τιμών, για την τιμή της μετάλλαξης	[-1, 1]
Μέγεθος νευρωνικού δικτύου	Ένα κρυφό στρώμα με 9 νευρώνες Ένα νευρώνα στο στρώμα εξόδου
Συνάρτηση ενεργοποίησης, για τους νευρώνες του κρυφού στρώματος	Logistic sigmoid [16]
Συνάρτηση εξόδου του δικτύου	Ground relu [17] (εκτός του Lorenz dataset [12], όπου έχει την γραμμική)

Πίνακας 2 Μεταβλητές παράμετροι

		Παράμετροι	
		ποσοστό ( <i>P</i> )	Threads
Τύποι πειράματος	A	0.1	5
	B	0.3	5
	C	0.1	10

Φτιάχτηκε και ένας ακόμη τύπος πειράματος με όνομα D. Με αυτό τον τύπο, μπορούμε να βγάλουμε συμπεράσματα, για την αποτελεσματικότητα της μεταβλητής εξέλιξης. Έχει τις ίδιες κοινές παραμέτρους, με τους προηγούμενους τύπους.

Οι διαφορετικοί παράμετροι είναι οι εξής:

Πίνακας 3 Οι παράμετροι του επιπλέον πειράματος D

Παράμετρος	Τιμή
Ποσοστό ( <i>P</i> )	0.1
Threads	5
Μέθοδος ανασυνδυασμού	Μεταβλητού μεγέθους
Μέθοδος, ποσοστό του fitness	Current best ranked
Μεταβλητό μέγεθος νευρωνικού δικτύου	Κρυφό στρώμα: [13], [8] νευρώνες Ένα νευρώνα στο στρώμα εξόδου

Για να μπορέσουμε να αντιληφθούμε την αποτελεσματικότητα της χρήσης μεθόδου ποσοστού, φτιάχτηκε ο τύπος test E. Είναι ουσιαστικά ακριβώς όπως το test D, απλά χωρίς την χρήση μεθόδου ποσοστού. Τελικά έχουμε πέντε τύπους προβλημάτων A, B, C, D, E

## ΚΕΦΑΛΑΙΟ – 7 Εκτέλεση πειραμάτων και αποτελέσματα

Όλα τα πειράματα του τελικού σταδίου έγιναν στον ίδιο υπολογιστή, με το μόνο ενεργό πρόγραμμα χρήση στο λειτουργικό σύστημα να είναι αυτό του test. Δεν υπάρχει λόγος καταγραφής χρόνου εκτέλεσης γιατί δεν μπορεί να γίνει σύγκριση των χρόνων με άλλον υπολογιστή, λόγω τις διαφοράς που θα έχουν τόσο στο υλικό μέρος αλλά και στο λειτουργικό. Για αυτό έχει κατασκευαστεί μια συνάρτηση κόστους που υπολογίζεται με βάση το πλήθος των υπολογισμών που χρειάζεται το σύστημα για κάθε test.

### 7.1 Αποτελέσματα

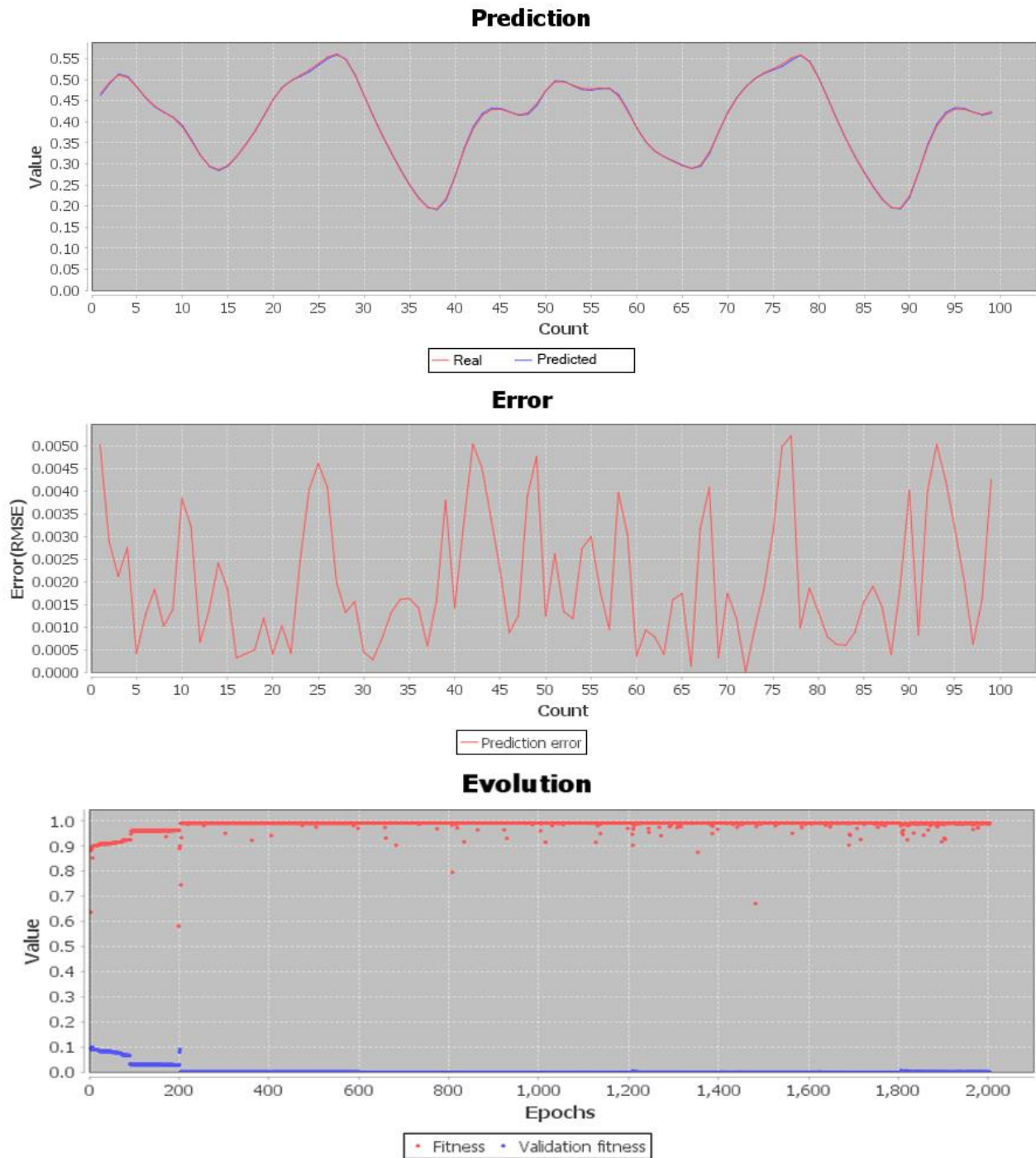
Αφού έγιναν όλες οι εκτελέσεις, έπειτα συλλέχθηκαν όλα τα δεδομένα από την έξοδο του συστήματος και παράχθηκαν οι παρακάτω πίνακες. Στους πίνακες αυτούς, γίνεται η σύγκριση, των αποτελεσμάτων του συστήματος μας (A, B, C, D, E) σε σχέση με τα αποτελέσματα της ήδη υπάρχουσας μελέτης (Sequential, Concurrent, MCNE) [8]. Όπως έχει προαναφερθεί, οι εξελικτικοί έχουν το στοιχείο του τυχαίου σε μεγάλο βαθμό, λόγω αυτού, έχει προστεθεί μία επιπλέον στήλη η Mean error bias (MEB). Οι τιμές της στήλης ανήκουν στο διάστημα [-100, 100], δείχνοντας κατά πόσο % το μέσο σφάλμα (στήλη Mean error), βρίσκεται πιο κοντά στο καλύτερο σφάλμα, (στήλη Best error). Έχοντας την τιμή 100% σημαίνει ότι το μέσο σφάλμα είναι ίσο με το καλύτερο σφάλμα, αντιθέτως έχοντας την τιμή -100% σημαίνει ότι είναι ίσο με το χειρότερο σφάλμα. Είναι αρκετά σημαντική στήλη, καθώς μπορούμε να διαπιστώσουμε, άμα η τιμή που πετύχαμε στο καλύτερο σφάλμα, είναι μια αρκετά πιθανή τιμή, ή απλά σταθήκαμε τυχεροί. Παρόλα αυτά, δεν μπορεί να γίνει σύγκριση του MEB από το ένα test στο άλλο. Προφανώς, γιατί έχουν διαφορετικές τιμές στο μέγιστο και ελάχιστο σφάλμα.

Ο τύπος που περιγράφει το MEB είναι:

$$MEB = - \left( 2 \cdot \frac{\mu_{\varepsilon} - m_{\varepsilon}}{M_{\varepsilon} - m_{\varepsilon}} - 1 \right) \cdot 100 \quad (6.1)$$

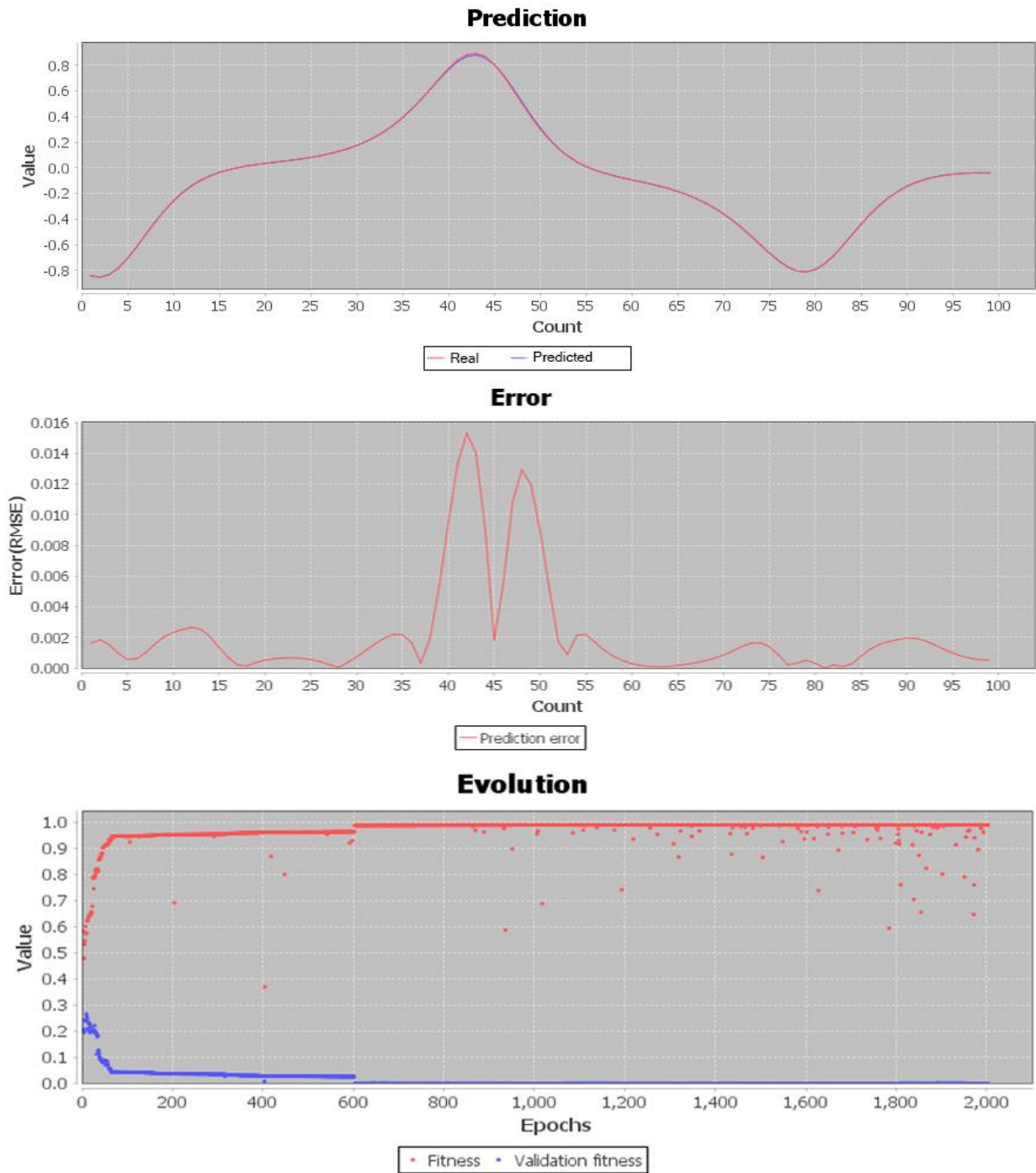
Όπου:

- $\mu_{\varepsilon}$ : το μέσο σφάλμα
- $m_{\varepsilon}$ : το ελάχιστο σφάλμα
- $M_{\varepsilon}$ : το μέγιστο σφάλμα



Σχήμα 11 Διαγράμματα από την εκτέλεση του συστήματος στο Mackey dataset

## Εκτέλεση πειραμάτων και αποτελέσματα



Σχήμα 12 Διαγράμματα από την εκτέλεση του συστήματος στο Lorenz dataset

Η στήλη Cost υπολογίζεται με τον τύπου του κόστους που έχει προαναφερθεί. Επίσης, προστέθηκε η στήλη Cost per thread, η οποία δείχνει το κόστος ανά thread. Γιατί έτσι όπως υπολογίζεται το αρχικό κόστος, είναι σαν να υπονοείται ότι η εκτέλεση του προγράμματος είναι σειριακή, ενώ στη πραγματικότητα είναι παράλληλη λόγω των thread. Η στήλη Mean (best validation error epoch) (MBVE), είναι ο μέσος όρος των εποχών, των οποίων προέκυψε το καλύτερο validation error. Έτσι προκύπτει το συμπέρασμα, ότι όσο μεγαλύτερο MBVE, τόσο λιγότερο το σύστημα εξαρτάται από το validation error, για την εύρεση του βέλτιστου δικτύου. Όλα τα παραπάνω εξηγούν κατά πόσο είναι χρήσιμο ως προς το σύστημα το validation dataset και κατά προέκταση το validation error.

Από το Σχήμα 9, μπορούμε να δούμε την αποτελεσματικότητα του συστήματος από το πρώτο διάγραμμα. Καθώς και η κόκκινη γραμμή που είναι τα πραγματικά δεδομένα και η μπλε που είναι η πρόβλεψη, εφάπτονται σχεδόν σε όλο το μήκος τους. Πιο αναλυτικά μπορούμε να δούμε στο δεύτερο διάγραμμα ότι το σύστημα έχει αρκετά αυξημένη αστοχία στην κορυφή που σχηματίζουν τα δεδομένα. Στο τρίτο διάγραμμα η εξέλιξη φαίνεται πολύ καλή καθώς δεν υπάρχει overtraining, και σχετικά από νωρίς το σύστημα έφτασε σε μια πολύ καλή λύση.

Στο Σχήμα 11, παρατηρούμε την τεράστια διαφορά του Mackey dataset σε σχέση με του Σχήματος 12 Lorenz. Και τα δύο όμως περιγράφουν μια περιοδικότητα. Και σε αυτό το dataset παρατηρούμε πολύ καλά αποτελέσματα από το σύστημα, με ακατάστατο σφάλμα που γενικά παρατηρείται αύξηση σε απότομες κορυφές η κοιλάδες που σχηματίζουν τα δεδομένα. Από την εξέλιξη βλέπουμε να υπάρχει ένα μικρό overtraining από την εποχή 1.800 περίπου και έπειτα.

Εκτέλεση πειραμάτων και αποτελέσματα

Πίνακας 4 Τα τελικά αποτελέσματα από τις προβλέψεις, από όλα τα dataset. Σύγκριση με τις εγγραφές Sequential, Concurrent, MCNE που είναι τα αποτελέσματα της έρευνας [8], για τα αντίστοιχα datasets

Santa Fe Laser [14]							
Test type	Best error	Mean error	Mean error bias	Worst error	Cost	Cost per thread	Mean (best validation error epoch)
A	0.0122094	0.0369875	5.63%	0.0647200	200075	40015	1619
B	0.0151571	0.0430133	-17.93%	0.0624001	195075	39015	1742
C	0.0138906	0.0323462	11.01%	0.0553674	500150	50015	1715
D	0.0186784	0.0606052	38.98%	0.1560946	200075	40015	1509
E	0.0203611	0.0656812	54.02%	0.2174987	200075	40015	1269
Sequential	0.0571243	0.0695330	-65.25%	0.0721420	269421	-	-
Concurrent	0.0634781	0.0768557	-68.66%	0.0793412	121200	-	-
MCNE	0.1471420	0.1949820	-33.44%	0.2188464	100000	-	-

Lorenz [12]							
Test type	Best error	Mean error	Mean error bias	Worst error	Cost	Cost per thread	Mean validation error epoch
A	0.0027403	0.0052443	24.78%	0.0093983	200075	40015	1897
B	0.0022897	0.0096347	33.82%	0.0244856	195075	39015	1813
C	0.0019629	0.0038730	28.18%	0.0072823	500150	50015	1949
D	0.0022847	0.0144763	32.90%	0.0386242	200075	40015	1767
E	0.0037872	0.0146563	62.75%	0.0621459	200075	40015	1809
Sequential	0.0713540	0.0731450	48.59%	0.0783210	260668	-	-
Concurrent	0.3214887	0.3445700	22.62%	0.3811421	121200	-	-
MCNE	0.0747062	0.0753210	73.42%	0.0793321	100000	-	-

<b>Mackey Glass [13]</b>							
Test type	Best error	Mean error	Mean error bias	Worst error	Cost	Cost per thread	Mean validation error epoch
A	0.0027723	0.0037160	45.77%	0.0062529	200075	40015	1880
B	0.0027492	0.0037760	14.47%	0.0051503	195075	39015	1864
C	0.0023960	0.0033838	11.41%	0.0046261	500150	50015	1966
D	0.0010170	0.0040382	6.24%	0.0074617	200075	40015	1814
E	0.0020943	0.0043915	36.02%	0.0092757	200075	40015	1873
Sequential	0.0019264	0.0045463	-37.10%	0.0057482	271031	-	-
Concurrent	0.0032004	0.0059527	-56.75%	0.0067121	121200	-	-
MCNE	0.0123215	0.0252556	-16.57%	0.0345122	100000	-	-

<b>Sunspot [11]</b>							
Test type	Best error	Mean error	Mean error bias	Worst error	Cost	Cost per thread	Mean validation error epoch
A	0.0077448	0.0100750	13.07%	0.0131057	200075	40015	1909
B	0.0083803	0.0109812	13.68%	0.0144062	195075	39015	1817
C	0.0063976	0.0077735	30.12%	0.0103354	500150	50015	1877
D	0.0078447	0.0128080	27.05%	0.0214516	200075	40015	1578
E	0.0077889	0.0134720	38.76%	0.0263481	200075	40015	1485
Sequential	0.0107341	0.0127696	53.78%	0.0195412	205039	-	-
Concurrent	0.0146470	0.0193530	10.14%	0.0251210	121200	-	-
MCNE	0.0246412	0.0478444	-9.27%	0.0671124	100000	-	-

TWI Exchange [10]							
Test type	Best error	Mean error	Mean error bias	Worst error	Cost	Cost per thread	Mean validation error epoch
A	0.0118479	0.0127522	48.92%	0.0153886	200075	40015	1658
B	0.0120645	0.0130102	42.49%	0.0153533	195075	39015	1576
C	0.0115597	0.0123720	46.37%	0.0145887	500150	50015	1701
D	0.0121878	0.0134124	53.20%	0.0174216	200075	40015	1465
E	0.0113766	0.0133122	40.42%	0.0178735	200075	40015	1604
Sequential	0.0354120	0.0394227	-38.23%	0.0412148	272318	-	-
Concurrent	0.0363142	0.0397674	0.59%	0.0432614	121200	-	-
MCNE	0.0745214	0.0852743	-28.59%	0.0912457	100000	-	-

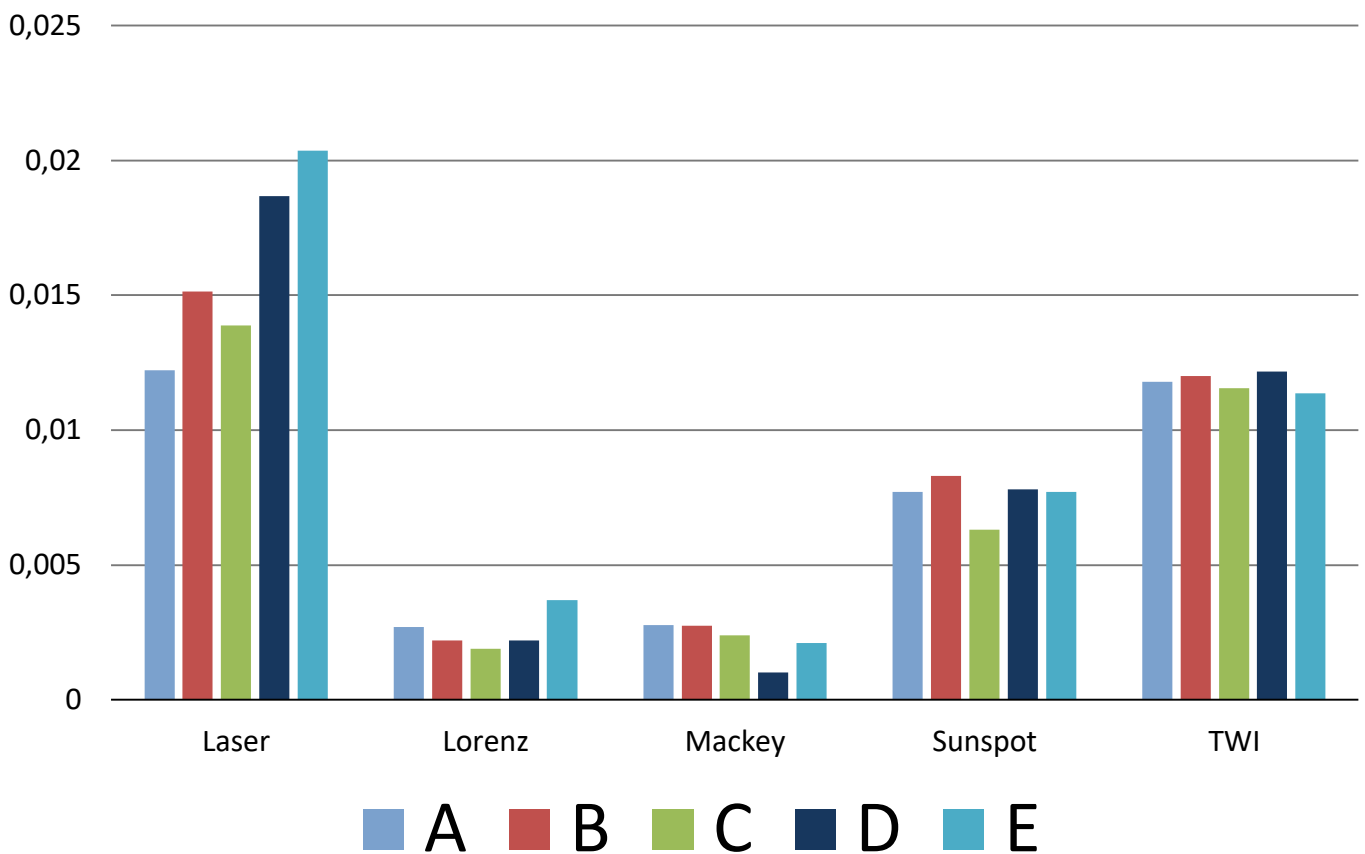
Στην πιο πρόσφατη δημοσίευση τους [8] έχουν δείξει πως το σύστημα τους βγάζει τα καλύτερα αποτελέσματα με την Sequential στρατηγική τους, σε σχέση με τις άλλες δυο στρατηγικές (Concurrent, MCNE), βγάζοντας χειρότερη την MCNE. Παρόλο που το δικό μας σύστημα βασίζεται με ένα γενικό τρόπο στη MCNE, καταφέρνει και παράγει καλύτερα αποτελέσματα από όλες τις στρατηγικές τους, τόσο στο μέσο σφάλμα όσο και στο μικρότερο σφάλμα, με περίπου το ίδιο κόστος. Κυρίαρχα test είναι το C και το A.

## 7.2 Επιπλέον ανάλυση

Έγινε μια επιπλέον στατιστική ανάλυση για να διαπιστωθεί η αποτελεσματικότητα των δικών μας test μεταξύ τους. Για την στατιστική ανάλυση έγινε χρήση του one-way ANOVA [18], με την χρήση του γραφικού προγράμματος PSPP [19]. Για να μπορέσει να δουλέψει η μέθοδος ANOVA χρειάζεται τουλάχιστον δύο παραμέτρους, ένα παράγοντα και μια εξαρτημένη μεταβλητή. Στο πρόβλημα μας ο παράγοντας είναι τα test που είναι ποιοτική μεταβλητή και εξαρτημένη μεταβλητή το evaluation error που είναι ποσοτική μεταβλητή. Το πρώτο συμπέρασμα από την στατιστική ανάλυση είναι, το test με τα καλύτερα αποτελέσματα σε όλα τα dataset είναι το C. Έχοντας το καλύτερο μέσο όρο απόδοσης, παράλληλα κατέχει και την μικρότερη διασπορά, που σημαίνει ότι παράγει συχνά και σταθερά, καλύτερα αποτελέσματα. Δεύτερο έρχεται το test A, που

στατιστικά δεν διαφέρει και πολύ σε απόδοση από το C. Τρίτο είναι το B, τέταρτο είναι το D και τελευταίο είναι το E. Από την απόδοση του D σε σχέση με το E, μπορούμε να συμπεράνουμε ότι, βοηθά η μέθοδος Percent of fitness, αλλά όχι σε πολύ μεγάλο βαθμό, γιατί στατιστικά η απόδοση των δύο test δεν διαφέρει και πολύ. Ένα από τα βασικότερα προβλήματα απόφασης, που αναλύονται στη έρευνα τους [8], είναι πόσο συχνά και από πόσες φορές πρέπει να εκτελείται ο αλγόριθμος του BP. Από την χαμηλή απόδοση του B, βλέπουμε ότι οι πολλές συνεχόμενες εποχές BP με μικρή όμως συχνότητα, δεν βοηθάνε το σύστημα, αντιθέτως οι μικρές σε πλήθος εποχές BP αλλά με μεγάλη συχνότητα σε εκτέλεση, αυξάνει την απόδοση του συστήματος.

### Αποτελέσματα από όλα τα dataset σε όλα τα tests



Σχήμα 13 Όλα τα τελικά αποτελέσματα από τα test για όλα τα dataset. Παρατηρούμε ότι σε τρία από τα πέντε dataset το test C είναι το καλύτερο.

### 7.3 Τελικά συμπεράσματα

Το γενικό συμπέρασμα που καταλήγουμε, όσο μεγαλύτερο πλήθος υποπληθυσμών έχουμε τόσο καλύτερα, τόσο πιο αποδοτικό σύστημα έχουμε. Διότι αυτό συνεπάγεται, σε μεγαλύτερο δείγμα πληθυσμού κατά το migration, που βοηθά του πιο αδύναμους πληθυσμούς, που με την σειρά του σημαίνει περισσότερα άτομα για ανάπτυξη, από το BP. Βέβαια αυτό έρχεται σε μεγάλο κόστος, στη περίπτωση που δε γίνει χρήση πολλαπλών thread και ταυτόχρονα δεν υπάρχει τέτοια υποστήριξη από το υλικό του υπολογιστή, συγκεκριμένα από τον επεξεργαστή του. Στην περίπτωση που πληρούνται οι προϋποθέσεις, τότε οι υποπληθυσμοί θα αναπτυχθούν παράλληλα σε πραγματικό χρόνο, ο κάθε ένας στο δικό του thread.

### 7.4 Προτεινόμενη μελλοντική μελέτη

Ένα κομμάτι που είναι εύλογο να αναπτυχθεί καλύτερα είναι αυτό του μεταβλητού μεγέθους χρωμοσωμάτων των ατόμων. Δηλαδή να γίνει μια πιο σωστή υλοποίηση μεθόδων στο κομμάτι του ανασυνδυασμού καθώς και στην μετάλλαξη, που να είναι βασιζόμενη πάνω σε ήδη υπάρχουσες τεχνικές [20], αλλά να παραμετροποιηθεί για να μπορεί να δουλέψει αρμονικά με τα νευρωνικά δίκτυα. Πολύ χρήσιμος θα είναι ο συνδυασμός των EA, με διαφορετικά δίκτυα όπως τα RNN [21] ή τα LSTM [22], ώστε να ανοιχτεί ακόμη περισσότερο το σύστημα, στα είδη των προβλημάτων που μπορεί να επιλύσει.

Επίσης θα ήταν σοφή η αναβάθμιση του BP, στο κομμάτι του δυναμικού learning rate, με την χρήση κάποιας άλλης μεθόδου από μια απλή συνάρτηση. Για παράδειγμα μια πολύ αποτελεσματική τέτοια μέθοδος είναι η Adam [23], η οποία έχει αποδείξει την αποτελεσματικότητά της, μέσα από την πληθώρα βιβλιοθηκών που την χρησιμοποιούν, γενικά στη μηχανική μάθηση. Ένας ακόμη μεγάλος κλάδος για έρευνα είναι το deep learning [13], [24], καθώς φαίνεται να χρησιμοποιείται σε ένα μεγάλο εύρος πραγματικών προβλημάτων, με μεγάλη επιτυχία.

## ΣΥΜΠΕΡΑΣΜΑΤΑ

Η τεχνική νοημοσύνη και πιο συγκεκριμένα ο κλάδος της μηχανικής μάθησης έχει πραγματοποιήσει πολύ μεγάλη ανάπτυξη τα τελευταία χρόνια, καθώς μπορεί να χρησιμοποιηθεί με επιτυχία σε ένα τεράστιο εύρος καθημερινών προβλημάτων. Με την ταυτόχρονη ραγδαία ανάπτυξη των έξυπνων συσκευών, διευρύνετε ακόμη περισσότερο ο ορίζοντας για την χρήση μηχανικής μάθησης, κάνοντας τον κλάδο αυτό, ένα αχανές περιβάλλον για έρευνα και ανάπτυξη νέων τεχνικών και μεθόδων, για την επίλυση κάθε προβλήματος

- [1] M. R. Islam, H. H. Lu, M. J. Hossain and L. Li, "A Comparison of Performance of GA, PSO and Differential Evolution Algorithms for Dynamic Phase Reconfiguration Technology of a Smart Grid," in *2019 IEEE Congress on Evolutionary Computation*, Wellington, New Zealand, 2019.
- [2] H. Mühlenbein, M. Gorges-Schleuter and O. Krämer, "Evolution algorithms in combinatorial optimization," *Parallel Computing*, vol. 7, no. 1, pp. 65-85, 1988.
- [3] L. Rodzina, O. Rodzina and S. Rodzin, "Neuroevolution: Problems, algorithms, and experiments," in *IEEE 10th International Conference on Application of Information and Communication Technologies*, Azerbaijan, Baku, 2016.
- [4] K. O. Stanley, J. Clune, J. Lehman and R. Miikkulainen, "Designing neural networks through neuroevolution," *Nature Machine Intelligence*, no. 1, pp. 24-35, 2019.
- [5] H.-N. Robert, "Theory of the Backpropagation Neural Network," *Neural Networks for Perception*, vol. III, pp. 65-93, 1992.
- [6] A. Palmer, J. J. Montaña and A. Sesé, "Designing an artificial neural network for forecasting tourism time series," *Tourism Management*, vol. 27, no. 5, pp. 781-790, 2006.
- [7] G. Wong, R. Chandra and A. Sharma, "Memetic Cooperative Neuro-Evolution for Chaotic Time Series Prediction," in *Neural Information Processing, A. Hirose, S. Ozawa, K. Doya, K. Ikeda, M. Lee, and D. Liu*, Springer, Cham, 2016, p. 299–308.
- [8] G. Wong, R. Chandra and A. Sharma, "Information Collection Strategies In Memetic Cooperative Neuroevolution For Time Series Prediction," in *International Joint Conference on Neural Networks*, Rio, Brazil, 2018.
- [9] M. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric Environment*, vol. 32, no. 14-15, pp. 2627-2636, 1998.
- [10] Admin, "Exchange rate (twi). may 1970 aug 1995," [Online]. Available: <https://datamarket.com/data/set/22tb>. [Accessed Feb 2014].

- [11] S. W. D. Center, "The International Sunspot Number (1843-2001), International Sunspot Number Monthly Bulletin and Online Catalogue," Royal Observatory of Belgium, Avenue Circulaire 3, 1180 Brussels, Belgium, [Online]. Available: <http://www.sidc.be/silso>. [Accessed 02 02 2015].
- [12] E. Lorenz, "Deterministic non-periodic flows," *Journal of Atmospheric Science*, pp. 267-285, 1963.
- [13] M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287-289, 1997.
- [14] A. S. Weigend and N. A. Gershenfeld, "Laser problem dataset: The santa fe time series competition data," 1994. [Online]. Available: <http://www-psych.stanford.edu/~andreas/Time-Series/SantaFe.html>.
- [15] C. Grosan and A. Abraham, "Hybrid Evolutionary Algorithms: Methodologies, Architectures, and Reviews," in *Hybrid Evolutionary Algorithms*, Berlin, Springer, 2007, pp. 1-17.
- [16] B. Karlik and A. V. Olgac, "Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks," *Bekir Karlik and A. Vehbi Olgac International Journal of Artificial Intelligence And Expert Systems*, vol. 1, no. 4, pp. 111-122, 2011.
- [17] Y. Li and Y. Yuan, "Convergence Analysis of Two-layer Neural Networks with ReLU Activation," in *Neural Information Processing Systems Conference*, Long Beach Convention & Entertainment Center, CA, USA, 2017.
- [18] A. Cuevas, M. Febrero and R. Fraiman, "An anova test for functional data," *Computational Statistics & Data Analysis*, vol. 47, no. 1, pp. 111-122, 2004.
- [19] "GNU PSPP," GNU Free Software Foundation, 29 6 2019. [Online]. Available: <https://www.gnu.org/software/pspp/>. [Accessed 10 1 2020].
- [20] B. Hutt and K. Warwick, "Synapsing Variable-Length Crossover: Meaningful Crossover for Variable-Length Genomes," *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, vol. 11, no. 1, pp. 118-131, 2007.
- [21] T. Mikolov, M. Karafiát, L. Burget, J. Černocký and S. Khudanpur, "Recurrent Neural Network Based Language Model," in *11th Annual Conference of the International Speech Communication Association*, Makuhari, Chiba, Japan, 2010.

- [22] F. Gers, J. Schmidhuber and F. Cummins, "Learning to forget: continual prediction with LSTM," in *9th International Conference on Artificial Neural Networks*, Edinburgh, UK, 1999.
- [23] D. P. Kingma and B. Jimmy, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations 2015*, San Diego, CA, USA, 2015.
- [24] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley and J. Clune, "Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning," 18 Dec 2017. [Online]. Available: <https://arxiv.org/abs/1712.06567>. [Accessed 10 1 2020].
- [25] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, no. 521, pp. 436-444, 2015.
- [26] D. Floreano and P. Dürr, "Neuroevolution: from architectures to learning," *Evolutionary Intelligence*, vol. 1, no. 1, pp. 47-62, 2008.
- [27] J. Reisinger and R. Miikkulainen, "Acquiring evolvability through adaptive representations," in *GECCO '07 Proceedings of the 9th annual conference on Genetic and evolutionary computation*, London England, 2007.
- [28] C. Igel, "Neuroevolution for reinforcement learning using evolution strategies," *The 2003 Congress on Evolutionary Computation, 2003. CEC '03*, vol. 4, pp. 2588-2595, 2003.