



ΤΕΧΝΟΛΟΓΙΚΟ  
ΕΚΠΑΙΔΕΥΤΙΚΟ  
ΙΔΡΥΜΑ  
ΤΕΙ ΗΠΕΙΡΟΥ

**ΤΕΙ ΗΠΕΙΡΟΥ**

**ΣΧΟΛΗ: ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ: ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΠΜΣ: ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΔΙΚΤΥΩΝ**

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΔΙΚΤΥΑΚΗ ΕΦΑΡΜΟΓΗ ΓΙΑ ΚΑΤΑΣΚΕΥΑΖΟΜΕΝΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ**

Γρηγόριος Τσούλος

Επιβλέπων: Ιωάννης Τσούλος, Επίκουρος,

Καθηγητής ΤΕΙ Ηπείρου

Αρτα-Ηπειρος, Σεπτέμβριος, 2018



ΤΕΧΝΟΛΟΓΙΚΟ  
ΕΚΠΑΙΔΕΥΤΙΚΟ  
ΙΔΡΥΜΑ  
—  
ΤΕΙ ΗΠΕΙΡΟΥ  
—  
**ΤΕΙ ΗΠΕΙΡΟΥ**

**ΣΧΟΛΗ: ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**  
**ΤΜΗΜΑ: ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ**  
**ΠΜΣ: ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ**  
**ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΔΙΚΤΥΩΝ**

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΔΙΚΤΥΑΚΗ ΕΦΑΡΜΟΓΗ ΓΙΑ ΚΑΤΑΣΚΕΥΑΖΟΜΕΝΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ**

Γρηγόριος Τσούλος

Επιβλέπων: Ιωάννης Τσούλος, Επίκουρος

Καθηγητής ΤΕΙ Ηπείρου

Αρτα-Ηπειρος, Σεπτέμβριος, 2018

# **A WEB APPLICATION FOR CONSTRUCTED NEURAL NETWORKS**

**Εγκρίθηκε από τριμελή εξεταστική επιτροπή**

Άρτα-Ηπειρος, Ιούνιος, 2018

## **ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ**

1. Επιβλέπων καθηγητής  
Ιωάννης Τσούλος, Επίκουρος  
Καθηγητής, ΤΕΙ Ηπείρου
  
2. Μέλος επιτροπής  
Αλέξανδρος Τζάλλας,  
Καθηγητής Εφαρμογών, ΤΕΙ Ηπείρου
  
3. Μέλος επιτροπής  
Στύλιος Χρυσόστομος,  
Καθηγητής, ΤΕΙ Ηπείρου

Ο διευθυντής του ΠΜΣ  
Στύλιος Χρυσόστομος,  
Καθηγητής, ΤΕΙ Ηπείρου  
Υπογραφή

© Τσούλος, Γρηγόριος, 2018.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

## Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα μεταπτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Τσούλος, Γρηγόριος

Υπογραφή

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Στην οικογένεια και στους φίλους μου για την υποστήριξη, σε όλα τα επίπεδα, στην ολοκλήρωση των ακαδημαϊκών μου στόχων και στον κ. Ιωάννη Τσούλο για τις συμβουλές και την καθοδήγησή του ως άμεσος επιβλέπων κατά τη διάρκεια της διπλωματικής μου εργασίας.

## ΠΕΡΙΛΗΨΗ

Στην παρούσα μεταπτυχιακή εργασία υλοποιήθηκε μια γραφική εφαρμογή σε HTML5 για το λογισμικό NNC που κατασκευάζει τεχνητά νευρωνικά δίκτυα με την χρήση της Γραμματικής Εξέλιξης. Το NNC μπορεί να χρησιμοποιηθεί για κατηγοριοποίηση δεδομένων αλλά και για μάθηση συναρτήσεων από ένα ευρύ σύνολο επιστημονικών προβλημάτων όπως σε προβλήματα φυσικής, χημείας, βιολογίας, οικονομικών κτλ. Το πακέτο NNC έχει εφαρμοστεί με επιτυχία και σε άλλα προβλήματα όπως επίλυση διαφορικών εξισώσεων αλλά και εύρεση αμιδιακού δεσμού 1 σε προβλήματα από τον χώρο της χημείας. Η γραφική εφαρμογή υλοποιήθηκε με την χρήση του πακέτου LARAVEL που είναι διαθέσιμο σε όλους χωρίς πληρωμή. Η γραφική εφαρμογή επιτρέπει στον χρήστη να εισάγει με γραφικό τρόπο τις παραμέτρους του NNC και να παρατηρήσει την πορεία της κατασκευής του βέλτιστου τεχνητού νευρωνικού δικτύου με γραφήματα. Τα τελικά αποτελέσματα από την κατασκευή αποθηκεύονται σε ειδικό χώρο που είναι διακριτός για κάθε χρήστη αλλά μπορούν και να κατέβουν τοπικά στον υπολογιστή του.

**Λέξεις-κλειδιά:** τεχνητά νευρωνικά δίκτυα, γενετικοί αλγόριθμοι, κατασκευαζόμενα τεχνητά νευρωνικά δίκτυα, Γραμματική εξέλιξη



## **ABSTRACT**

In this work a graphical application in HTML5 is presented for the software NNC, then can create artificial neural networks using the well known technique of Grammatical Evolution. The NNC package can be used in classification problems as well as in regression problems from research areas such as physics, chemistry, biology, economics etc. The package NNC has been applied also in other problems such as the solution of differential equations and the location of Amide I bonds in chemical problems. The graphical application has been created using the PHP framework LARAVEL that is freely available from the Internet. The graphical application has some interesting capabilities such as an easy way to provide the parameters of NNC and a graphical view of the progress of the genetic algorithm to locate the best artificial neural network. The final results are stored in a special area for every user the user can download them to his personal computer.

**Keywords:** artificial neural networks, genetic algorithms, constructed artificial neural networks, grammatical evolution

# Πίνακας περιεχομένων

ΕΥΧΑΡΙΣΤΙΕΣ .....	IV
ΠΕΡΙΛΗΨΗ .....	V
ABSTRACT .....	VI
ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ .....	X
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ.....	XI
ΕΙΣΑΓΩΓΗ .....	XII
<b>1.ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ.....</b>	<b>1</b>
1.1 Το πρόβλημα.....	1
1.2 Πρόβληματα κατηγοριοποίησης .....	1
1.2.1 Εκμάθηση .....	2
1.2.2 Κατηγοριοποίηση.....	2
1.2.3 Μέθοδοι κατηγοριοποίησης .....	2
1.2.4 Ο Αλγόριθμος K κοντινότεροι γείτονες .....	4
1.3 Πρόβληματα REGRESSION .....	6
1.4 ΕΙΣΑΓΩΓΗ ΣΤΑ ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ .....	6
1.4.1 Ορισμός.....	6
1.4.2 Εισαγωγή-βασικές έννοιες.....	6
1.4.3 Ιστορική αναδρομή.....	7
1.4.4 Από τα Βιολογικά Νευρωνικά Δίκτυα στα Τεχνητά.....	7
1.4.5 Περιγραφή ενός Νευρώνα .....	10
1.4.6 Τύποι συναρτήσεων ενεργοποίησης.....	12
1.4.7 Αρχιτεκτονική νευρωνικών δικτύων.....	14
1.4.8 Εκπαίδευση νευρωνικών δικτύων.....	20
1.4.9 Εφαρμογές νευρωνικών δικτύων .....	22
1.4.10 Πλεονεκτήματα των νευρωνικών δικτύων .....	23
1.4.11 Μειονεκτήματα των νευρωνικών δικτύων .....	23
1.5 ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ PERCEPTRON .....	24
1.5.1 Εισαγωγή.....	24
1.5.2 Ορισμός.....	24
1.5.3 Perceptron πολλών στρωμάτων .....	28
1.5.4 Μειονεκτήματα των MLP – Εφαρμογές.....	29
<b>2 ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΙ ΕΡΓΑΛΕΙΑ ΥΛΟΠΟΙΗΣΗΣ ΕΦΑΡΜΟΓΗΣ .....</b>	<b>30</b>
2.1 ΔΙΑΔΙΚΤΥΑΚΟΣ ΕΞΥΠΗΡΕΤΗΤΗΣ UBUNTU LINUX SERVER .....	30
2.1.1 Linux .....	30
2.1.2 Ubuntu Server .....	31
2.1.3 Ο διακομιστής Apache HTTP Server .....	32
2.2 Η ΓΛΩΣΣΑ HTML (HYPERTEXT MARKUP LANGUAGE) .....	34
2.2.1 HTML 5.....	34
2.2.2 Η ιστορία της HTML.....	35
2.3 CSS(CASCADING STYLE SHEETS).....	36
2.4 ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ .....	37
2.4.1 Βασικές έννοιες τις βάσεις δεδομένων.....	37

2.4.2 Συστήματα διαχείρισης βάσεων δεδομένων(DBMS).....	38
2.4.3 SQL- MySQL.....	38
2.5 ΡΗΡΜΥADMIN .....	40
2.5.1 Χαρακτηριστικά του <i>phpMyAdmin</i> .....	40
2.5.2 Πλεονεκτήματα του <i>phpMyAdmin</i> .....	41
2.6 Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ PHP .....	41
2.6.1 Εισαγωγή.....	41
2.6.2 Ιστορία της <i>PHP</i> .....	41
2.6.3 Χρήση <i>PHP</i> και <i>HTML</i> .....	42
2.6.4 Πλεονεκτήματα σε σχέση με την <i>HTML</i> .....	43
2.6.5 Γενικά πλεονεκτήματα της <i>PHP</i> γλώσσας .....	43
2.6.6 Γενικά μειονεκτήματα της <i>PHP</i> γλώσσας .....	44
2.7 Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ JAVASCRIPT .....	44
2.7.1 Γενικά στοιχεία της <i>Javascript</i> .....	44
2.7.2 Μοντέλο εκτέλεσης.....	45
2.7.3 <i>Javascript</i> και <i>Java</i> .....	46
2.7.4 Βιβλιοθήκες της <i>Javascript</i> .....	46
2.8 BOOTSTRAP .....	49
2.8.1 Προέλευση.....	49
2.8.2 Χαρακτηριστικά.....	50
2.8.3 Δομή και Λειτουργία.....	50
2.8.4 Χρήση.....	52
2.9 SOFTWARE FRAMEWORKS.....	52
2.9.1 Φιλοσοφία .....	53
2.9.2 Παραδείγματα.....	54
2.9.3 Αρχιτεκτονική.....	55
2.9.4 Πλεονεκτήματα του <i>Framework</i> .....	55
2.9.5 Μειονεκτήματα του <i>Framework</i> .....	56
2.10 LARAVEL FRAMEWORK .....	57
2.10.1 Η Αρχιτεκτονική του <i>framework</i> .....	58
2.10.2 Η δομή της εφαρμογής.....	59
<b>3 ΚΑΤΑΣΚΕΥΑΖΟΜΕΝΑ ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ .....</b>	<b>60</b>
3.1 ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΓΕΝΕΤΙΚΟΥΣ ΑΛΓΟΡΙΘΜΟΥΣ .....	60
3.1.1 Η Θεωρία εξέλιξης των ειδών.....	63
3.1.2 Γενικός τρόπος λειτουργίας των γενετικών αλγορίθμων .....	63
3.1.3 Ορολογία Γενετικών Αλγορίθμων .....	65
3.1.4 Βασικά χαρακτηριστικά γενετικών αλγορίθμων .....	65
3.1.5 Αρχικοποίηση.....	66
3.1.6 Αξιολόγηση καταλληλότητας .....	67
3.1.7 Επιλογή.....	67
3.1.8 Αναπαραγωγή.....	67
3.1.9 Διασταύρωση.....	68
3.1.10 Μετάλλαξη .....	69
3.1.11 Πλεονεκτήματα γενετικών αλγορίθμων .....	70
3.1.12 Μειονεκτήματα Γενετικών Αλγορίθμων .....	70
3.1.13 Εφαρμογές.....	71
3.1.14 Εφαρμογές των ΓΑ για τη βελτιστοποίηση των <i>TNA</i> .....	71
3.2 ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΓΡΑΜΜΑΤΙΚΗ ΕΞΕΛΙΞΗ (GRAMMATICAL EVOLUTION) .....	72
3.2.1 Βασικοί ορισμοί .....	72
3.2.2 Παράδειγμα παραγωγής .....	73

3.2.3 Γενετικοί τελεστές.....	74
<b>3.3 ΚΑΤΑΣΚΕΥΑΖΟΜΕΝΑ ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ.....</b>	<b>75</b>
3.3.1 Εισαγωγή.....	75
3.3.2 Μέθοδος παραγωγής τεχνητών νευρωνικών δικτύων .....	75
<b>3.4 ΧΡΗΣΗ ΚΑΤΑΣΚΕΥΑΖΟΜΕΝΩΝ ΤΕΧΝΗΤΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΣΕ ΕΦΑΡΜΟΓΕΣ.....</b>	<b>78</b>
3.4.1 Επίλυση διαφορικών εξισώσεων.....	78
3.4.2 Υπολογισμός βέλτιστων αποστάσεων σε μόρια .....	78
<b>4. Η ΕΦΑΡΜΟΓΗ ΠΟΥ ΥΛΟΠΟΙΗΘΗΚΕ .....</b>	<b>79</b>
4.1 ΤΟ ΛΟΓΙΣΜΙΚΟ NNC.....	79
4.2 ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	84
4.3 ΠΑΡΑΔΕΙΓΜΑΤΑ ΧΡΗΣΕΩΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ .....	92
4.4. ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ .....	97
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>98</b>

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1	Η λειτουργία του k-NN.....	5
Σχήμα 2	Ο φυσικός νευρώνας σε σχέση με τον στοιχειώδη τεχνητό νευρώνα (Perceptron). ....	8
Σχήμα 3	Μαθηματικό μοντέλο νευρώνα.....	9
Σχήμα 4	Φιλοσοφία λειτουργίας νευρωνικού δικτύου.....	10
Σχήμα 5	Μοντέλο μη γραμμικού νευρώνα.....	11
Σχήμα 6	Επίδραση του κατωφλίου $\theta_k$ στην είσοδο της συνάρτησης ενεργοποίησης. ....	12
Σχήμα 7	Νευρωνικό δίκτυο πολλαπλών στρωμάτων.....	12
Σχήμα 8	Απεικόνιση συνάρτησης κατωφλίου.....	13
Σχήμα 9	Απεικόνιση τμηματικά γραμμικής συνάρτησης.....	13
Σχήμα 10	Σιγμοειδής συναρτήσεις.....	14
Σχήμα 11	Νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης αρχιτεκτονικής 3-3-2. ....	15
Σχήμα 12	Νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης αρχιτεκτονικής 3-3-1. ....	16
Σχήμα 13	Εμπρός-Τροφοδότησης δίκτυο με ένα επίπεδο νευρώνων. ....	17
Σχήμα 14	Πλήρως συνδεδεμένο δίκτυο εμπρός-τροφοδότησης με ένα κρυφό επίπεδο και ένα επίπεδο εξόδου.....	17
Σχήμα 15	Μερικώς συνδεδεμένο δίκτυο εμπρός-τροφοδότησης.....	18
Σχήμα 16	Αναδρομικό δίκτυο χωρίς ανατροφοδότηση και χωρίς κρυφούς νευρώνες. ....	19
Σχήμα 17	Αναδρομικό δίκτυο με κρυφούς νευρώνες. ....	20
Σχήμα 18	Παράδειγμα perceptron.....	24
Σχήμα 19	Η πράξη XOR. ....	25
Σχήμα 20	Ο perceptron για την XOR.....	26
Σχήμα 21	Το πρόβλημα της XOR είναι μη γραμμικά διαχωρίσιμο. ....	27
Σχήμα 22	Ο εξελικτικός κύκλος των Γενετικών Αλγορίθμων. ....	63
Σχήμα 23	Διάγραμμα ροής εκτέλεσης ενός Γ.Α. ....	64
Σχήμα 24	Διασταύρωση ενός σημείου. ....	68
Σχήμα 25	Μετάλλαξη τριών σημείων. ....	69
Σχήμα 26	Γραμματική μια γλώσσας με πράξεις ανάμεσα σε x και y. ....	73
Σχήμα 27	Παράδειγμα παραγωγής έκφρασης. ....	73
Σχήμα 28	Διασταύρωση ενός σημείου σε grammatical evolution. ....	74
Σχήμα 29	Προτεινόμενη γραμματική.....	76
Σχήμα 30	Διεργασίες που δηλώνονται στο BaseController. ....	88
Σχήμα 31	Δημιουργία ενός object για να χρησιμοποιήσει καθόλη την διάρκεια της εκτέλεσης το NNC. ....	88
Σχήμα 32	Εκτέλεση του αλγορίθμου.....	89
Σχήμα 33	Διεργασίες του αλγορίθμου.....	91

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

<b>Εικόνα 1</b> Ubuntu Server. ....	31
<b>Εικόνα 2</b> Λογότυπο Laravel Framework. ....	58
<b>Εικόνα 3</b> Σελίδα καλωσορίσματος στην πλατφόρμα NNC. ....	92
<b>Εικόνα 4</b> Σελίδα register στην πλατφόρμα NNC. ....	92
<b>Εικόνα 5</b> Σελίδα login στην πλατφόρμα NNC. ....	93
<b>Εικόνα 6</b> Σελίδα Algorithm στην πλατφόρμα NNC. ....	93
<b>Εικόνα 7</b> Παράμετροι του NNC. ....	94
<b>Εικόνα 8</b> Σελίδα Graphs στην πλατφόρμα NNC. ....	95
<b>Εικόνα 9</b> Εκτέλεση αλγορίθμου στην σελίδα Graphs. ....	96
<b>Εικόνα 10</b> Σελίδα Profile στην πλατφόρμα NNC. ....	96

## ΕΙΣΑΓΩΓΗ

Στο κείμενο αυτό θα συζητηθούν και θα αναλυθούν ζητήματα, όπως η προσαρμογή δεδομένων, τα τεχνητά νευρωνικά δίκτυα, τα κατασκευαζόμενα τεχνητά νευρωνικά δίκτυα με την χρήση γραμματικής και επίσης η δημιουργία ενός online συστήματος για την εκτέλεση παραδειγμάτων νευρωνικών δικτύων. Με τον όρο προσαρμογή δεδομένων εννοούμε την εύρεση αναλυτικής μορφής μίας συναρτήσεως που να ικανοποιεί μία σειρά από δεδομένα που διαθέτουμε από παρατηρήσεις. Αν βρούμε μία τέτοια μορφή θα μπορούσαμε να προβλέψουμε την τιμή που θα επέστρεφαν οι παρατηρήσεις μας αν γίνονταν σε διαφορετικά σημεία από αυτά των προηγούμενων μετρήσεων, δηλαδή θα περάσουμε από την παρατήρηση στην πρόβλεψη. Η σειρά των μεθόδων που αναπτύσσονται στον τομέα της προσαρμογής δεδομένων μπορούν να μας βοηθήσουν σε πολλές επιστημονικές περιοχές όπως Ιατρική, Χημεία κτλ. Τα τεχνητά νευρωνικά δίκτυα μπορούν να θεωρηθούν σαν μια απεικόνιση των διασυνδέσεων των νευρώνων του ανθρώπινου εγκεφάλου και θεωρούνται σαν μια ικανοποιητική μεθοδολογία στον χώρο της τεχνητής νοημοσύνης, καθώς έχουν την τάση να μαθαίνουν μέσα από παραδείγματα με τον ίδιο περίπου τρόπο που πιστεύεται πως μαθαίνει ο ανθρώπινος εγκέφαλος.

# 1.Τεχνητά νευρωνικά δίκτυα

## 1.1 Το πρόβλημα

Στην εργασία που παρουσιάζεται στην συνέχεια γίνεται μία προσπάθεια να χρησιμοποιηθούν τεχνικές από τους χώρους των τεχνητών νευρωνικών δικτύων για την επίλυση προβλημάτων προσαρμογής δεδομένων. Τα τεχνητά νευρωνικά δίκτυα σήμερα χρησιμοποιούνται με επιτυχία σε δύο μεγάλες κατηγορίες: την διάκριση κατηγοριών [18] και το regression [1], δηλαδή της προσεγγίσεως αγνώστων συναρτήσεων [24]. Η επιτυχία των τεχνητών νευρωνικών δικτύων στην δεύτερη κατηγορία προβλημάτων καθιστά τα τεχνητά νευρωνικά δίκτυα κατάλληλα για την αντιμετώπιση των προβλημάτων που πραγματεύεται η προσαρμογή δεδομένων. Για να μπορέσουν τα τεχνητά νευρωνικά δίκτυα να επιτύχουν καλά αποτελέσματα στην προσαρμογή δεδομένων, δηλαδή για να μάθουν μία συνάρτηση, απαιτούν μία σειρά από δεδομένα εκπαίδευσης και προσπαθούν με διάφορες τεχνικές να μάθουν την συμπεριφορά της άγνωστης συναρτήσεως ακόμα και σε σημεία που δεν υπάρχουν στα δεδομένα εκπαίδευσης. Αυτές οι τεχνικές ξεκινούν από ένα σχετικά μεγάλο σφάλμα προσεγγίσεως της άγνωστης συναρτήσεως και σταδιακά προσπαθούν να το ελαττώσουν [47] .

## 1.2 Προβλήματα κατηγοριοποίησης

Η κατηγοριοποίηση (classification) είναι μία τεχνική της μηχανικής μάθησης, κατά την οποία ένα στοιχείο ανατίθεται σε ένα προκαθορισμένο σύνολο κατηγοριών [24]. Ο όρος κατηγοριοποίηση συναντάται στην βιβλιογραφία και ως ταξινόμηση. Γενικότερα, ο στόχος της διαδικασίας αυτής είναι η ανάπτυξη ενός μοντέλου, το οποίο αργότερα θα μπορεί να χρησιμοποιηθεί για την κατηγοριοποίηση μελλοντικών δεδομένων [19]. Τέτοια παραδείγματα είναι ο διαχωρισμός των emails με βάση την επικεφαλίδα τους ή το περιεχόμενό τους, η πρόβλεψη καρκινικών κυττάρων χαρακτηρίζοντας τα ως καλοήθη ή κακοήθη, η κατηγοριοποίηση πελατών μιας τράπεζας ανάλογα με την πιστωτική τους ικανότητα κ.α [9]. Η κατηγοριοποίηση μπορεί να περιγράψει ως μία διαδικασία δύο βημάτων:



### 1.2.1 Εκμάθηση

Στο πρώτο βήμα της διαδικασίας δημιουργείται το μοντέλο με βάση ένα σύνολο προκατηγοριοποιημένων παραδειγμάτων, που ονομάζεται δεδομένα εκπαίδευσης (training data). Τα δεδομένα εκπαίδευσης αναλύονται από ένα αλγόριθμο κατηγοριοποίησης, προκειμένου να σχηματιστεί το μοντέλο [25]. Λόγω του ότι τα δεδομένα εκπαίδευσης ανήκουν σε μία προκαθορισμένη κατηγορία, η οποία είναι γνωστή, η κατηγοριοποίηση αποτελεί μέθοδο εποπτευομένης μάθησης (supervised learning). Το μοντέλο, που λέγεται και αλλιώς κατηγοριοποιητής (classifier), αναπαρίσταται με τη μορφή κανόνων κατηγοριοποίησης (classification rules), δέντρων απόφασης (decision trees) ή μαθηματικών τύπων.

### 1.2.2 Κατηγοριοποίηση

Μετά την δημιουργία του μοντέλου, το επόμενο βήμα είναι η αξιολόγησή του. Για να επιτευχθεί αυτό, χρησιμοποιούμε τα δεδομένα ελέγχου (test data) και το μοντέλο χρησιμοποιείται για να κατηγοριοποιήσει τα δεδομένα αυτά [6]. Έπειτα, η πρόβλεψη που έγινε με βάση τα δεδομένα ελέγχου συγκρίνεται με την πρόβλεψη που έγινε για τα δεδομένα εκπαίδευσης, τα οποία είναι ανεξάρτητα από αυτά της δοκιμής [12]. Η ακρίβεια του μοντέλου υπολογίζεται από το ποσοστό των δειγμάτων δοκιμής που κατηγοριοποιήθηκαν σωστά από το μοντέλο [41].

Στην περίπτωση που το μοντέλο κριθεί αποδεκτό, τότε μπορεί να χρησιμοποιηθεί για την κατηγοριοποίηση μελλοντικών δειγμάτων δεδομένων, των οποίων η κατηγοριοποίηση είναι άγνωστη [17].

### 1.2.3 Μέθοδοι κατηγοριοποίησης

Σε αυτήν την ενότητα θα δούμε τις πιο σημαντικές κατηγορίες μεθόδων κατηγοριοποίησης που χρησιμοποιούνται σε τέτοιου είδους αναλύσεις και θα μας βοηθήσουν στην επίλυση του παραπάνω προβλήματος [25].

#### **Bayesian**

Η Bayesian τεχνική κατηγοριοποίησης βασίζεται στη στατιστική θεωρία κατηγοριοποίησης του Bayes. Αυτό σημαίνει ότι πραγματοποιείται μια πιθανοτική πρόβλεψη, δηλαδή προβλέπει την πιθανότητα ένα δείγμα  $X$  να ανήκει σε κάποια κατηγορία. [24]

### **Δέντρα Απόφασης στη Μηχανική Μάθηση**

Η μηχανική μάθηση με δέντρα απόφασης χρησιμοποιεί ένα δέντρο απόφασης ως προγνωστικό μοντέλο το οποίο χαρτογραφεί παρατηρήσεις σχετικά με ένα στοιχείο (που αντιπροσωπεύεται στα κλαδιά) σε συμπεράσματα σχετικά με την τιμή στόχο του στοιχείου (που αντιπροσωπεύεται στα φύλλα). Είναι μία από τις προσεγγίσεις προγνωστικής μοντελοποίησης που χρησιμοποιούνται στις στατιστικές, εξόρυξη δεδομένων και μηχανική μάθηση. Τα μοντέλα δέντρων όπου η μεταβλητή στόχος μπορεί να πάρει ένα πεπερασμένο σύνολο τιμών ονομάζονται δέντρα ταξινόμησης. Σε αυτές τις δομές δέντρων, τα φύλλα αντιπροσωπεύουν τις ετικέτες τάξης / κατηγορίας και τα κλαδιά αντιπροσωπεύουν τους συνδέσμους των χαρακτηριστικών που οδηγούν σε αυτές τις ετικέτες κατηγορίας. Τα δέντρα απόφασης, όπου η μεταβλητή στόχος μπορεί να πάρει συνεχείς τιμές (συνήθως πραγματικοί αριθμοί) ονομάζονται δέντρα παλινδρόμησης. Στην ανάλυση αποφάσεων, ένα δέντρο απόφασης μπορεί να χρησιμοποιηθεί για να αντιπροσωπεύσει ρητά τις αποφάσεις και τη λήψη αποφάσεων. Στην εξόρυξη δεδομένων, ένα δέντρο απόφασης περιγράφει τα δεδομένα (αλλά το προκύπτον δέντρο ταξινόμησης μπορεί να είναι μια εισροή για τη λήψη αποφάσεων). Το δέντρο απόφασης στη μηχανική μάθηση είναι μια μέθοδος που χρησιμοποιείται ευρέως στον τομέα της εξόρυξης δεδομένων. Ο στόχος είναι να δημιουργηθεί ένα μοντέλο που προβλέπει την τιμή μιας μεταβλητής στόχου βασισμένο σε διάφορες μεταβλητές εισόδου.

### **Νευρωνικά δίκτυα**

Μια άλλη τεχνική κατηγοριοποίησης που χρησιμοποιείται σε εφαρμογές εξόρυξης γνώσης για πρόβλεψη και κατηγοριοποίηση είναι νευρωνικά δίκτυα με τα ακόλουθα βήματα λειτουργίας τους [47] :

1. Η αναγνώριση των χαρακτηριστικών εισόδου και εξόδου
2. Δημιουργία ενός δικτύου με την κατάλληλη τοπολογία
3. Επιλογή του συνόλου εκπαίδευσης (train data)
4. Επαλήθευση-αξιολόγηση του δικτύου με την χρήση ενός σύνολο ελέγχου (test data).

Τα νευρωνικά δίκτυα και ειδικά τα Κατασκευαζόμενα Νευρωνικά Δίκτυα είναι το κύριο αντικείμενο μελέτης αυτής της διατριβής

### 1.2.4 Ο Αλγόριθμος K κοντινότεροι γείτονες

Ο Αλγόριθμος K κοντινότεροι γείτονες (K Nearest Neighbors - KNN) είναι μία πολύ γνωστή και ευρεία χρησιμοποιούμενη τεχνική κατηγοριοποίησης που στηρίζεται στη χρήση μέτρων βασισμένων στην απόσταση.

Η κεντρική ιδέα είναι πως η τιμή της συνάρτησης-στόχου για ένα νέο στιγμιότυπο βασίζεται αποκλειστικά και μόνο στις αντίστοιχες τιμές των k πιο «κοντινών» στιγμιότυπων εκπαίδευσης, τα οποία αποτελούν τους «γείτονες» του [11]. Δύο ζητήματα πρέπει να αποφασιστούν προκειμένου να καθοριστεί πλήρως ο αλγόριθμος:

1. Ο ορισμός της απόστασης μεταξύ δύο στιγμιότυπων, δηλαδή μμιας τιμής πάνω στο χώρο των στιγμιότυπων, που θα εκφράζει την εγγύτητα, ή αλλιώς την «ομοιότητα» μεταξύ των στιγμιότυπων.
2. Η τιμή του k.

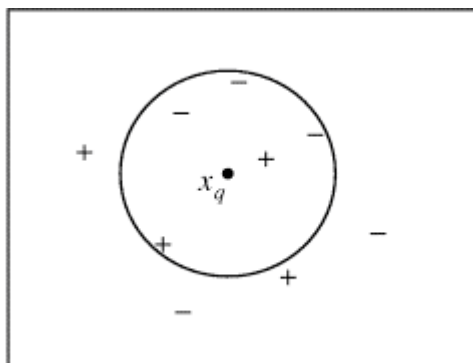
Για το πρώτο ζήτημα, υπάρχουν πολλές εναλλακτικές επιλογές. Η απόφαση εξαρτάται από τα ειδικά χαρακτηριστικά του χώρου στιγμιότυπων του προβλήματος. Ιδιαίτερη σημασία έχει αν στην αναπαράσταση των στιγμιότυπων περιλαμβάνονται αριθμητικά ή συμβολικά χαρακτηριστικά. Στον «παραδοσιακό» k-NN αλγόριθμο, στον οποίο τα στιγμιότυπα θεωρούνται πως ανήκουν στον n-διάστατο χώρο  $R^n$ , μία μέτρηση που υιοθετείται συχνά είναι η γνωστή Ευκλείδεια απόσταση. Πιο συγκεκριμένα, αν τα στιγμιότυπα αναπαρίστανται ως διανύσματα από χαρακτηριστικά που παίρνουν τιμές πραγματικούς αριθμούς, δηλαδή το στιγμιότυπο x αναπαρίσταται από το διάνυσμα:

$$\langle a_1(x), a_2(x), \dots, a_n(x) \rangle$$

Όπου  $a_r(x)$  δηλώνει την τιμή του r-οστού χαρακτηριστικού του x, τότε η απόσταση  $d(x_i, x_j)$  μεταξύ δύο στιγμιότυπων  $x_i$  και  $x_j$  ορίζεται ως:

$$d(x_i, x_j) = \sqrt{\sum_r^n (a_r(x_i) - a_r(x_j))^2}$$

Φυσικά, είναι δυνατόν να επιλεχθούν και άλλα μέτρα ομοιότητας-ανομοιότητας, αντί της Ευκλείδειας.



Σχήμα 1 Η λειτουργία του k-NN.

Στο σχήμα 1 φαίνεται η λειτουργία του k-NN. Τα “+” και τα “-” δείχνουν τα στιγμιότυπα εκπαίδευσης της κάθε κλάσης και το  $x_q$  ένα στιγμιότυπο προς κατάταξη. Φαίνεται πως ο 1-NN κατατάσσει το  $x_q$  ως “+” ενώ ο 7-NN το κατατάσσει ως “-”. Ένα μειονέκτημα που παρουσιάζεται στο προηγούμενο παράδειγμα είναι πως όλα τα χαρακτηριστικά θεωρούνται ισοδύναμα κατά τον υπολογισμό της απόστασης. Αυτό είναι ιδιαίτερα προβληματικό αν δεν είναι όλα τα χαρακτηριστικά σχετικά με τη συγκεκριμένη συνάρτηση-στόχο που επιδιώκεται να προσεγγιστεί, αλλά και γενικότερα, οποτεδήποτε υπάρχουν σημαντικές διαφορές μεταξύ των χαρακτηριστικών ως προς την αξία τους στον προσδιορισμό της συνάρτησης. Σε μία τέτοια περίπτωση, οι παραπάνω μετρήσεις είναι παραπλανητικές, από την άποψη πως μερικά στιγμιότυπα που σχετίζονται πραγματικά μεταξύ τους, είναι δυνατόν να θεωρούνται απομακρυσμένα λόγω των διαφορών τους σε άσχετα ή ασήμαντα χαρακτηριστικά. Μια λύση σε αυτό το πρόβλημα είναι κάθε χαρακτηριστικό να αποτιμάται διαφορετικά στον υπολογισμό της απόστασης, ανάλογα με την αξία του. Η μέθοδος αυτή λέγεται αποτίμηση των χαρακτηριστικών (feature weighting). Με βάση αυτήν, ο τύπος της Ευκλείδειας θα μπορούσε να γίνει:

$$d(x_i, x_j) = \sqrt{\sum_r^n w_r (a_r(x_i) - a_r(x_j))^2}$$

όπου  $w_r$  είναι το βάρος τους χαρακτηριστικού  $a_r$ .

### **1.3 Προβλήματα Regression**

Με τον όρο regression (μάθηση συναρτήσεων) εννοούμε την δημιουργία μοντέλων τα οποία μπορούν να προσεγγίσουν τιμές από μετρήσεις, χρονοσειρές, μαθηματικές συναρτήσεις κτλ. Για παράδειγμα η εκτίμηση της πορείας μιας μετοχής σε ένα χρηματιστήριο είναι πρόβλημα μάθησης. Επίσης η προσέγγιση των καλαθιών (σε κάποιο διάστημα) που μπορεί να επιτύχει ένας παίκτης είναι πρόβλημα μάθησης συναρτήσεων.[1], [10]

### **1.4 Εισαγωγή στα τεχνητά νευρωνικά δίκτυα**

Στην παρούσα διατριβή θα ασχοληθούμε ειδικά με τα τεχνητά νευρωνικά δίκτυα και για αυτό θα παρουσιαστούν στην συνέχεια τα θεμελιώδη συστατικά τους και βασικές εισαγωγικές έννοιες.

#### **1.4.1 Ορισμός**

Νευρωνικό δίκτυο ονομάζεται ένα κύκλωμα διασυνδεδεμένων νευρώνων [20]. Στην περίπτωση των βιολογικών νευρώνων, πρόκειται για ένα τμήμα νευρικού ιστού [16]. Στην περίπτωση τεχνητών νευρώνων, πρόκειται για μία μέθοδο η οποία εμπίπτει στον τομέα της υπολογιστικής νοημοσύνης, όπου στόχος του νευρωνικού δικτύου είναι η επίλυση κάποιου υπολογιστικού προβλήματος [15], [31], [47] .

#### **1.4.2 Εισαγωγή-βασικές έννοιες**

Η όλη ιδέα και η σύλληψη της επιστήμης των τεχνητών νευρωνικών δικτύων βασίζεται στην δομή του ανθρώπινου εγκεφάλου ο οποίος αποτελείται από εκατομμύρια νευρώνες με σκοπό να εκτελούνται όλες οι ανθρώπινες λειτουργίες [10], [46]. Ο εγκέφαλος είναι ένας εξαιρετικά πολύπλοκος μη γραμμικός παράλληλος υπολογιστής. Ένα χαρακτηριστικό παράδειγμα είναι η ανθρώπινη όραση [7], [33]. Ο εγκέφαλος είναι ο κυρίως αποδέκτης των ερεθισμάτων που δέχεται ο άνθρωπος και ανάλογα με το τι βλέπει αντιδρά ανάλογα [20], [40]. Το θέμα εδώ είναι ότι επεξεργάζεται τα όποια ερεθίσματα συναντά με ταχύτητα σχεδόν αμελητέα και μη αντιληπτή για εμάς (περίπου 100 με 200 ns) την ίδια στιγμή που εργασίες πολύ μικρότερου βαθμού πολυπλοκότητας απαιτούν πολύ μεγαλύτερους χρόνους εκτέλεσης

από ένα ισχυρό υπολογιστή [16], [35]. Στην παρούσα εργασία θα επικεντρωθούμε σε μια σημαντική κατηγορία νευρωνικών δικτύων τα οποία εκτελούν χρήσιμους υπολογισμούς αφού εκπαιδευτούν μέσω μιας διαδικασίας μάθησης [9], [37], [45]. Για να πετύχουν καλύτερη απόδοση, τα νευρωνικά δίκτυα χρησιμοποιούν μια σειρά απλών διασυνδεδεμένων μεταξύ τους υπολογιστικών κυττάρων, τα οποία αποκαλούνται «νευρώνες» ή «μονάδες επεξεργασίας» [13], [32].

### **1.4.3 Ιστορική αναδρομή**

Η μοντέρνα περίοδος των νευρωνικών δικτύων λέγεται ότι άρχισε με την πρωτοποριακή δουλειά των McCulloch και Pitts το 1943. Ο πρώτος ήταν ψυχίατρος και ο δεύτερος μεγαλοφυής μαθηματικός [11], [35].

Η επόμενη μεγάλη ανάπτυξη πάνω στα νευρωνικά δίκτυα, ήρθε το 1949 με την έκδοση του βιβλίου του Hebb με τίτλο “The Organization of Behavior”, στο οποίο έγινε για πρώτη φορά μια ιδιαίτερη δήλωση ενός φυσιολογικού κανόνα μάθησης για συνοπτικές τροποποιήσεις. Πιο συγκεκριμένα ο Hebb πρότεινε ότι η συνδετικότητα του εγκεφάλου συνεχώς αλλάζει καθώς ο οργανισμός μαθαίνει διάφορες εργασίες, και ότι οι νευρωνικοί συγκέντρωτες δημιουργούνται από τέτοιες αλλαγές [19], [36].

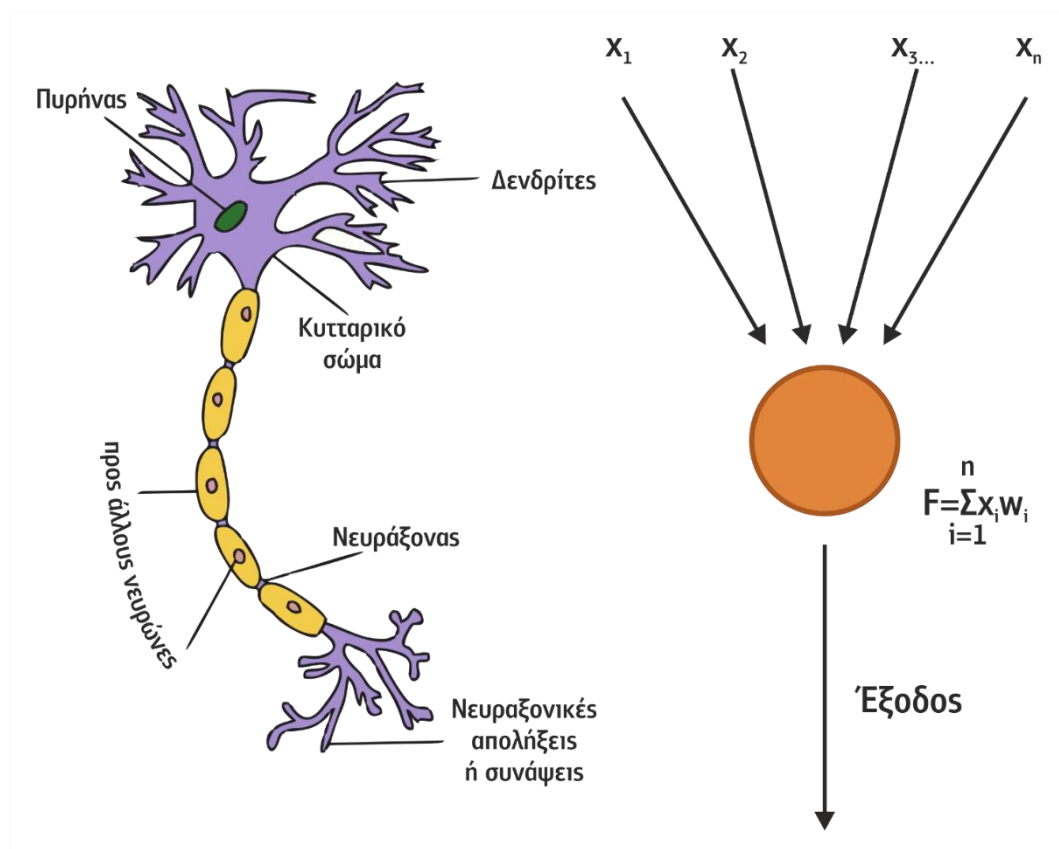
Το 1952 εκδόθηκε το βιβλίο του Ashby με τίτλο “Design for a brain: The Origin of Adaptive Behavior”, το οποίο ασχολήθηκε με την βασική έννοια ότι η προσαρμοζόμενη συμπεριφορά δεν είναι έμφυτη αλλά μαθαίνεται. Ακολούθησαν αρκετές εργασίες και μελέτες που αφορούσαν νευρωνικά δίκτυα την δεκαετία του '80.

Ως αποκορύφωμα το 1989 εκδόθηκε το βιβλίο του Mead με τίτλο “VLSI and Neural Systems”. Αυτό το βιβλίο δίνει μια ασυνήθιστη μίξη περιεχομένων από την νευροβιολογία και την τεχνολογία VLSI. Ίσως περισσότερο από κάθε άλλη έκδοση, η εργασία του Hopfield (1982) και το δίτομο βιβλίο των Rummelhart και McClelland (1986), να ήταν οι πιο σημαίνουσες εκδόσεις υπεύθυνες για την αναζωογόνηση του ενδιαφέροντος για τα νευρωνικά δίκτυα στην δεκαετία του '80.

### **1.4.4 Από τα Βιολογικά Νευρωνικά Δίκτυα στα Τεχνητά**

Ο εγκέφαλος είναι ένας πολύ πολύπλοκος, μη-γραμμικός και παράλληλος υπολογιστής. Έχει τη δυνατότητα να οργανώνει τους νευρώνες έτσι ώστε να εκτελεί συγκεκριμένους

υπολογισμούς πολύ πιο γρήγορα από τους πιο γρήγορους ψηφιακούς υπολογιστές που υπάρχουν. Κατά τη γέννησή του ο εγκέφαλος έχει την ικανότητα να κατασκευάζει τους δικούς του κανόνες, κοινώς “εμπειρία”, η οποία μεγαλώνει με την πάροδο του χρόνου. [3] Κατά τα 2 πρώτα χρόνια ζωής, έχουμε τη μέγιστη ανάπτυξη, όπου περίπου 1 εκατομμύριο συνάψεις (synapses) δημιουργούνται στο δευτερόλεπτο [30], [34]. Οι συνάψεις είναι οι βασικές δομικές και λειτουργικές μονάδες που μεσολαβούν στην ενδοεπικοινωνία των νευρώνων [35], [37].

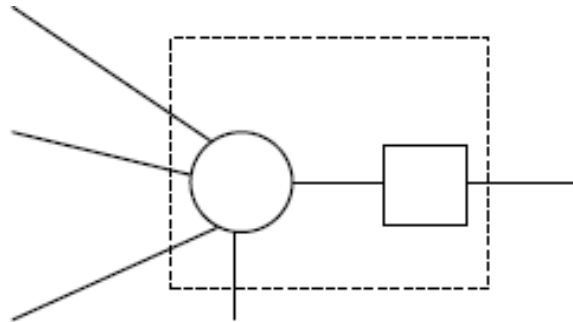


Σχήμα 2 Ο φυσικός νευρώνας σε σχέση με τον στοιχειώδη τεχνητό νευρώνα (Perceptron).

Τα τεχνητά νευρωνικά δίκτυα (Artificial Neural Network, ANN) μιμούνται σε μεγάλο βαθμό τα νευρωνικά δίκτυα του εγκεφάλου. Η διαδικασία για την εκμάθηση ονομάζεται “αλγόριθμος μάθησης”. Τα νευρωνικά δίκτυα αποτελούνται από πολλά προγραμματιζόμενα στοιχεία που ονομάζονται νευρώνες [15], [31]. Τα στοιχεία συνδέονται με συνδεδετικά βάρη, κάτι το οποίο τους επιτρέπει να προσαρμόζονται μέσω μιας διαδικασίας εκμάθησης. Το κυρίαρχο τους πλεονέκτημα σε σχέση με τα στατιστικά μοντέλα έχει να κάνει με το γεγονός ότι μοντελοποιούν ένα πολυδιάστατο πρόβλημα χωρίς να απαιτούν περίπλοκες υποθέσεις ανάμεσα στις μεταβλητές εισόδου [36]. Στο (Σχήμα 3) φαίνεται το μαθηματικό μοντέλο του

νευρώνα. Όπως φαίνεται στο σχήμα ένας νευρώνας μπορεί να δέχεται  $n$  εισόδους ( $x_1, x_2, \dots, x_n$ ) και να παράγει μια έξοδο,  $y$ . Κάθε νευρώνας διαθέτει και μια πόλωση  $b$ , η οποία μπορεί να είναι μη μηδενική ή μηδενική. Σε κάθε μια από τις  $n$  εισόδους  $x_1, x_2, \dots, x_n$  αντιστοιχεί ένα βάρος  $w_1, w_2, \dots, w_n$ , αντίστοιχα. Η συνολική είσοδος  $z$  του νευρώνα υπολογίζεται ως εξής:

$$z = W_1 \cdot X_1 + W_2 \cdot X_2 + \dots + W_n \cdot X_n + b$$



Σχήμα 3 Μαθηματικό μοντέλο νευρώνα.

Ενώ, η έξοδος  $y$  του νευρώνα υπολογίζεται ως εξής:

$$y = f(z) = f(W_1 \cdot X_1 + W_2 \cdot X_2 + \dots + W_n \cdot X_n + b)$$

Η συνάρτηση  $f$  ονομάζεται συνάρτηση ενεργοποίησης ή συνάρτηση μεταφοράς. Δύο τυπικές συναρτήσεις ενεργοποίησης που χρησιμοποιούνται συχνά στα νευρωνικά δίκτυα είναι η σιγμοειδής συνάρτηση:

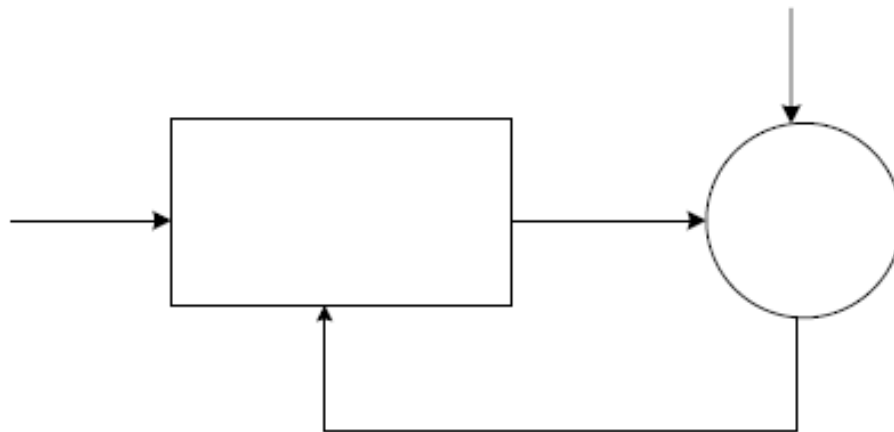
$$f(z) = \text{log sig}(z) = \frac{1}{1 + e^{-z}}$$

και η συνάρτηση υπερβολικής εφαπτομένης:

$$f(z) = \text{tan sig}(z) = \tanh(z) = \frac{2}{1 + e^{-2z}} - 1$$



Η βασική φιλοσοφία των νευρωνικών δικτύων παρουσιάζεται στο Σχήμα 4.

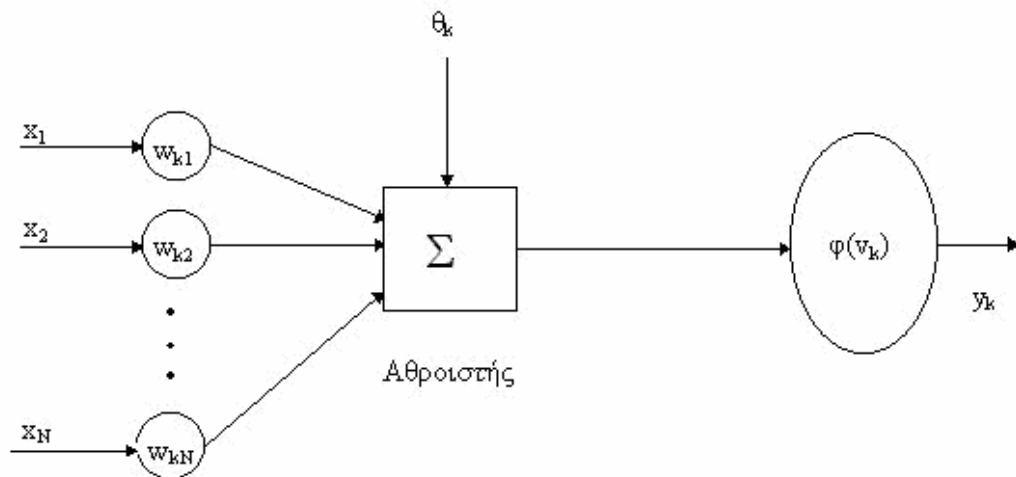


Σχήμα 4 Φιλοσοφία λειτουργίας νευρωνικού δικτύου.

#### 1.4.5 Περιγραφή ενός Νευρώνα

Τα βασικά χαρακτηριστικά ενός νευρώνα είναι τα εξής:

1. Τα συναπτικά βάρη: αν η είσοδος του νευρώνα  $k$  στη σύναψη  $j$  είναι το σήμα  $X_i$ , τότε αυτό πολλαπλασιάζεται με το συναπτικό βάρος  $W_{ki}$ , όπου ο πρώτος δείκτης αναφέρεται στο νευρώνα και ο δεύτερος, στη σύναψη που δέχεται στην είσοδό της το σήμα  $W_{ki}$ .
2. Ο αθροιστής (Adder): ο οποίος στην έξοδό του δίνει το άθροισμα των σταθμισμένων εισόδων [16], [34], [47].
3. Η συνάρτηση ενεργοποίησης (Activation Function): από αυτήν περνά η έξοδος του αθροιστή και δίνει αποτέλεσμα στο διάστημα  $[0, 1]$  ή  $[-1, 1]$ , ανάλογα με τον τύπο της συνάρτησης που επιλέχθηκε [15], [31].
4. Το κατώφλι  $\theta_k$ : πρόκειται για δευτερεύουσα παράμετρο του συστήματος, η οποία συνήθως επιλέγεται με στόχο την καλύτερη ευελιξία του.



Σχήμα 5 Μοντέλο μη γραμμικού νευρώνα.

Στο παραπάνω σχήμα 5 δίνεται το μοντέλο ενός νευρώνα με κατώφλι  $\theta_k$  είσοδο το διάνυσμα  $P = (x_1, x_2, \dots, x_n)$  και βάρη  $W = (w_{11}, w_{12}, \dots, w_{1n})$

Έτσι ισχύει:  $U_k = \sum_j^N w_{kj} \cdot x_j$

όπου προκειμένου  $k=1$ , η έξοδος του νευρώνα είναι:  $y_k = \varphi(u_k - \theta_k)$

όπου  $x_1, x_2, \dots, x_n$ , τα σήματα εισόδου και  $w_1, w_2, \dots, w_n$ , τα συναπτικά βάρη του νευρώνα  $k$ ,  $\theta_k$  το κατώφλι του, και  $\varphi(\cdot)$  η συνάρτηση ενεργοποίησής του.

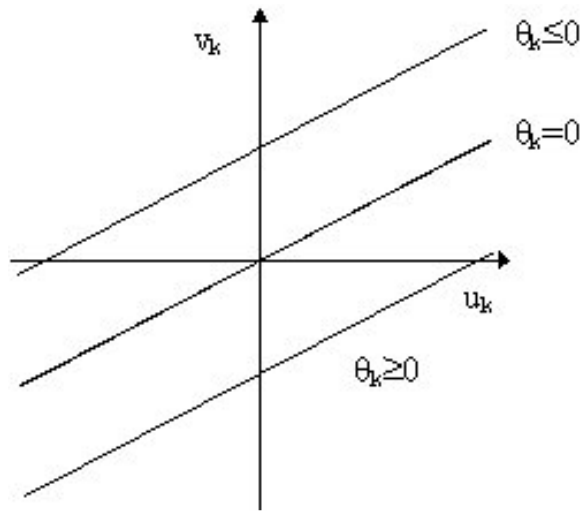
Θεωρώντας, και το κατώφλι ως συναπτικό βάρος  $W_{k0}$  της εισόδου  $x_0 = -1$ , το  $v_k$ , δηλαδή η έξοδος του αθροιστή, δίνεται από τη σχέση:

$$V_k = \sum_{i=0}^N w_{ki} \cdot x_i$$

ενώ η έξοδος του νευρώνα είναι:

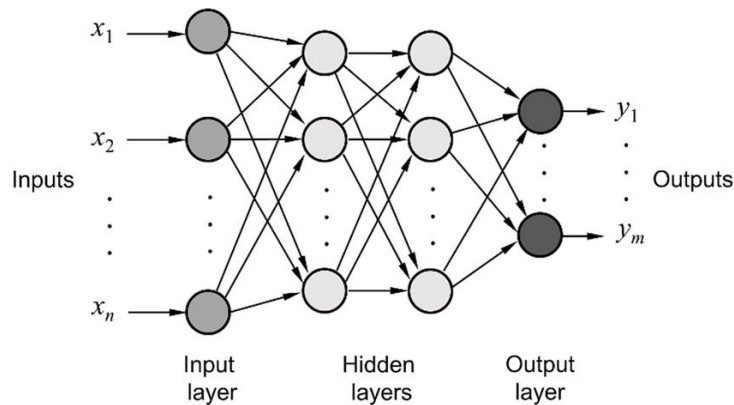
$$y_k = \varphi(v_k)$$

Στο σχήμα 6 που ακολουθεί φαίνεται και ο ρόλος που διαδραματίζει το κατώφλι  $\theta_k$ , το οποίο τελικά ελαττώνει την είσοδο της συνάρτησης ενεργοποίησης όταν το σήμα είναι θετικό και την αυξάνει όταν είναι αρνητικό [30], [49].



Σχήμα 6 Επίδραση του κατωφλίου  $\theta_k$  στην είσοδο της συνάρτησης ενεργοποίησης.

Τα σύγχρονα νευρωνικά δίκτυα αποτελούνται από πολλούς απλούς νευρώνες, όπως αυτόν που περιγράφηκε παραπάνω, οι οποίοι συνθέτουν αρχικά ένα στρώμα και στη συνέχεια ένα σύστημα πολλαπλών στρωμάτων [33].



Σχήμα 7 Νευρωνικό δίκτυο πολλαπλών στρωμάτων.

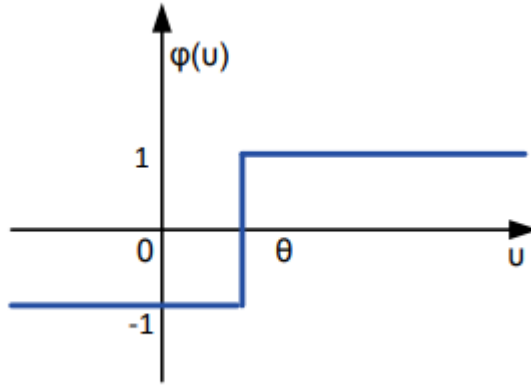
#### 1.4.6 Τύποι συναρτήσεων ενεργοποίησης

Η συνάρτηση ενεργοποίησης  $\varphi(\cdot)$ , ορίζει την έξοδο ενός νευρώνα συναρτήσει του επιπέδου ενεργοποίησης της εισόδου. Έχουμε 3 βασικούς τύπους συναρτήσεων ενεργοποίησης [11].

##### 1. Συνάρτηση Κατωφλιού.

Για αυτόν τον τύπο συνάρτησης ενεργοποίησης, που περιγράφεται στο σχήμα 8, έχουμε:

$$\varphi(u) = \begin{cases} 1, u, & u \geq 0 \\ 0, u, & u < 0 \end{cases}$$



*Σχήμα 8* Απεικόνιση συνάρτησης κατωφλίου.

Επομένως η έξοδος ενός νευρώνα  $k$ , που έχει την παραπάνω συνάρτηση ενεργοποίησης, έχει τη μορφή:

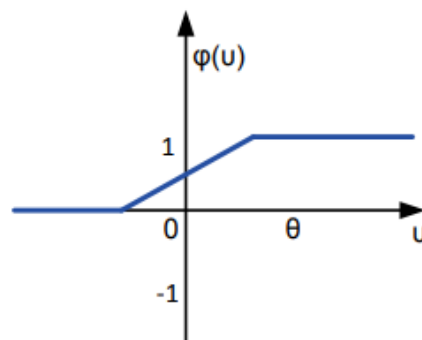
$$y_k = \begin{cases} 1, & u_k \geq \theta \\ 0, & u_k < \theta \end{cases}$$

όπου  $u_k$  είναι το εσωτερικό επίπεδο ενεργοποίησης του νευρώνα και δίνεται από τη σχέση:

## 2. Τμηματικά Γραμμική Συνάρτηση.

Για την τμηματικά γραμμική συνάρτηση, που περιγράφεται στο σχήμα 9, έχουμε:

$$\varphi(u) = \begin{cases} 1, & u \geq \theta \\ u, & -\theta < u < \theta \\ 0, & u \leq -\theta \end{cases}$$



*Σχήμα 9* Απεικόνιση τμηματικά γραμμικής συνάρτησης.

Αυτή η μορφή συνάρτησης ενεργοποίησης μπορεί να θεωρηθεί σαν μια προσέγγιση ενός μη-γραμμικού ενισχυτή [35], [40].

### 3.Σιγμοειδής.

Είναι η πιο συνηθισμένη μορφή συνάρτησης ενεργοποίησης που χρησιμοποιείται στην κατασκευή τεχνητών νευρωνικών δικτύων. Ορίζεται σαν αυστηρά αύξουσα συνάρτηση, η οποία παρουσιάζει εξομάλυνση και ασυμπτωτικές ιδιότητες [32].

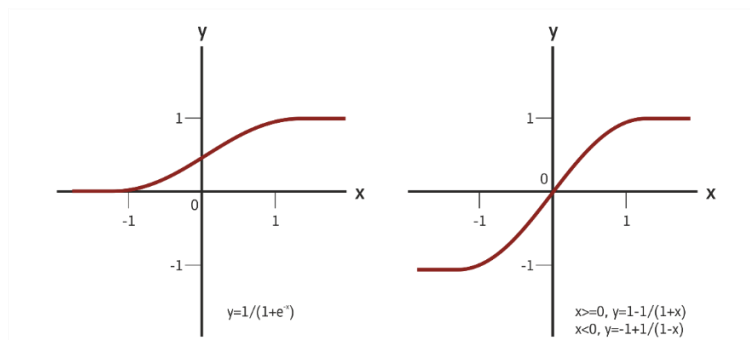
Ένα παράδειγμα είναι η *λογιστική συνάρτηση*, που ορίζεται από τη σχέση:

$$\varphi(v) = \frac{1}{1 + \exp(-av)}$$

Μεταβάλλοντας την παράμετρο  $a$  παίρνουμε διαφορετικές συναρτήσεις, όπως φαίνεται στο σχήμα 10. Οι προηγούμενες συναρτήσεις ενεργοποίησης κυμαίνονται από 0 ως +1. Αν θέλουμε συνάρτηση που να κυμαίνεται από -1 ως +1, επαναπροσδιορίζουμε την συνάρτηση κατωφλιού ως:

$$\varphi(v) \begin{cases} 1, v > 0 \\ 0, v = 0 \\ -1, v < 0 \end{cases}$$

που ονομάζεται συνάρτηση *προσήμου* (*signum*) και συμβολίζεται σαν  $\text{sgn}(\cdot)$ .



Σχήμα 10 Σιγμοειδής συναρτήσεις.

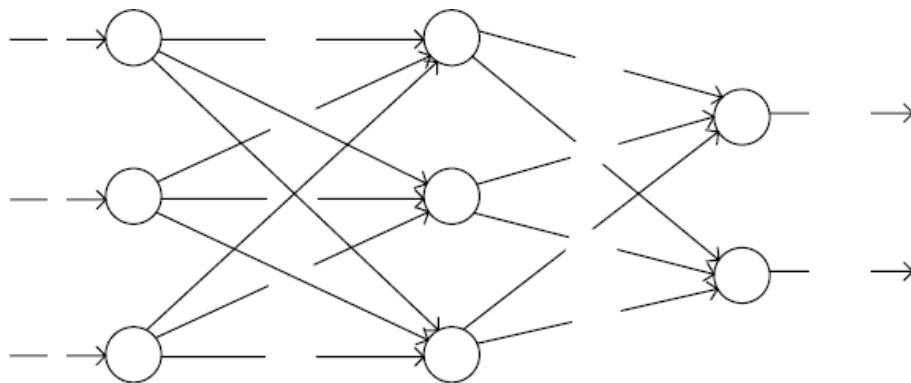
### 1.4.7 Αρχιτεκτονική νευρωνικών δικτύων

Το τεχνητό νευρωνικό δίκτυο μπορεί να θεωρηθεί ως μια συλλογή από νευρώνες αλληλοσυνδεδεμένους μεταξύ τους. Τα πλέον ευρέως χρησιμοποιούμενα νευρωνικά δίκτυα είναι τα νευρωνικά δίκτυα πολλών στρωμάτων, τα οποία είναι νευρωνικά δίκτυα εμπρόσθιας τροφοδότησης (NNET). Τα NNET αποτελούνται από ένα στρώμα εισόδου, ένα

ή περισσότερα κρυφά στρώματα και ένα στρώμα εξόδου. Κάθε στρώμα με διαφορετικά βάρη [16], [39].

Κάθε κρυφός νευρώνας πολλαπλασιάζει κάθε είσοδο με το βάρος της, προσθέτει τα γινόμενα, προσθέτει την πόλωση, και μετά περνάει το άθροισμα μέσω μιας συνάρτησης μεταφοράς (ενεργοποίησης) για να παράγει το αποτέλεσμα [30], [36], [40].

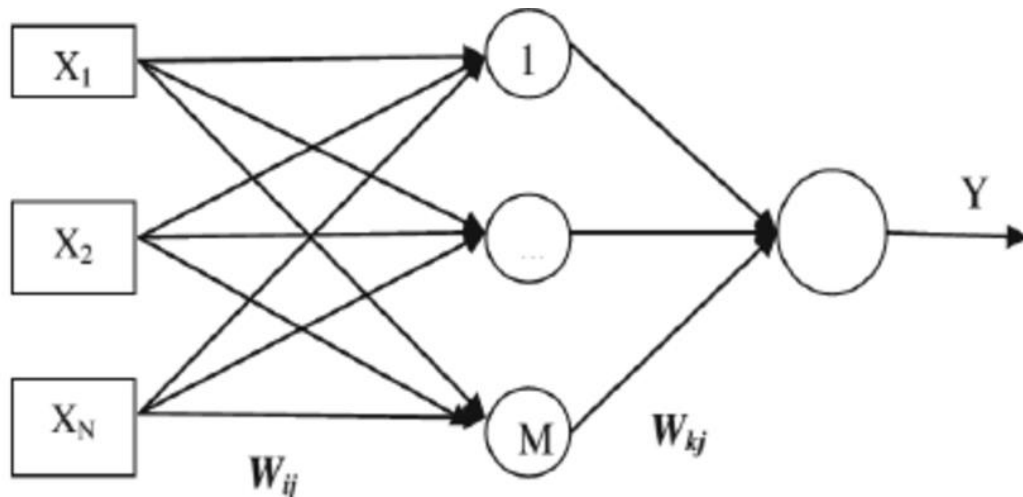
Στο Σχήμα 11 φαίνεται ένα νευρωνικό δίκτυο με 3 νευρώνες στο στρώμα εισόδου, 3 νευρώνες στο κρυφό στρώμα, και 2 νευρώνες στο στρώμα εξόδου. Αυτό το νευρωνικό δίκτυο συμβολίζεται ως δίκτυο αρχιτεκτονικής 3-3-2 (από τους αριθμούς των νευρώνων στο στρώμα εισόδου, το κρυφό στρώμα, και το στρώμα εξόδου αντίστοιχα).



*Σχήμα 11 Νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης αρχιτεκτονικής 3-3-2.*

Το νευρωνικό αυτό δίκτυο έχει  $3 \cdot 3 = 9$  συνδέσεις μεταξύ των 3 νευρώνων εισόδου και των 3 κρυφών νευρώνων, και  $3 \cdot 2 = 6$  συνδέσεις μεταξύ των 3 κρυφών νευρώνων και των 2 νευρώνων εξόδου. Συνολικά το δίκτυο έχει 15 συνδέσεις. Σε κάθε σύνδεση αντιστοιχεί και μια τιμή βάρους  $w_{pj,qk}$  που ενώνει το νευρώνα  $p$  του στρώματος  $j$  με το νευρώνα  $q$  του στρώματος  $k$ . Οι κρυφοί νευρώνες και οι νευρώνες εξόδου του νευρωνικού αυτού δικτύου έχουν μηδενικές πολώσεις. Στους κρυφούς νευρώνες και στους νευρώνες εξόδους χρησιμοποιείται κάποια συνάρτηση ενεργοποίησης (μεταφοράς), για παράδειγμα, η σιγμοειδής συνάρτηση [19].

Στο Σχήμα 1.11 φαίνονται αναλυτικά τα σήματα εισόδου και εξόδου του δεύτερου νευρώνα του δεύτερου στρώματος (κρυφό στρώμα) που αντιστοιχούν στο νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης στο Σχήμα 12. Με  $INPrj$  και  $OUTprj$  συμβολίζονται η συνολική είσοδος και η έξοδος, αντίστοιχα, του νευρώνα  $p$  του στρώματος  $j$ .



Σχήμα 12 Νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης αρχιτεκτονικής 3-3-1.

Η συνολική είσοδος του δεύτερου νευρώνα του δεύτερου στρώματος είναι:

$$INP_{22} = W_{11,22} \cdot X_1 + W_{21,22} \cdot X_2 + W_{31,22} \cdot X_3$$

Επειδή στους κρυφούς νευρώνες και στους νευρώνες εξόδου χρησιμοποιείται η σιγμοειδής συνάρτησης ενεργοποίησης, η έξοδος του δεύτερου νευρώνα του δεύτερου στρώματος είναι:

$$OUT_{22} = \frac{1}{1 + e^{-INP_{22}}}$$

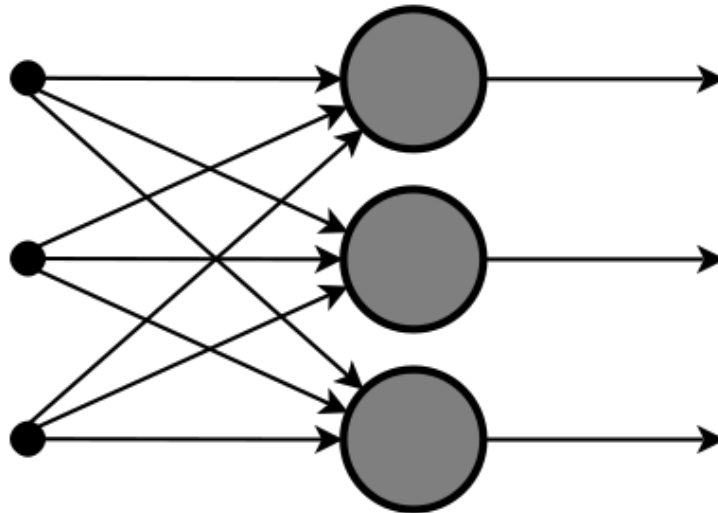
Με αντίστοιχο τρόπο παράγονται τα σήματα εξόδου και των υπόλοιπων νευρώνων του δικτύου εμπρόσθιας τροφοδότησης.

Ο τρόπος με τον οποίο οι νευρώνες ενός νευρωνικού δικτύου δομούνται, είναι στενά συνδεδεμένος με τον αλγόριθμο εκμάθησης που χρησιμοποιείται για την εκπαίδευση του δικτύου. Σαν εμπρός τροφοδότησης, αναφέρονται τα δίκτυα, στα οποία τα σήματα κατευθύνονται από την είσοδο στην έξοδο. Όταν οι εξοδοί κάποιων νευρώνων, γίνονται είσοδοι σε νευρώνες προηγούμενων επιπέδων (προς το μέρος της εισόδου του δικτύου), τότε έχουμε ανάδραση [37].

Μπορούμε να διακρίνουμε 4 διαφορετικές κλάσεις αρχιτεκτονικών δομών

#### 1.Ενός-επιπέδου Εμπρός-Τροφοδότησης Δίκτυα.

Σε ένα τέτοιο δίκτυο, οι νευρώνες είναι οργανωμένοι σε μορφή επιπέδων. Οι νευρώνες του πηγαίου επιπέδου δείχνουν στους νευρώνες του επόμενου επιπέδου αλλά όχι αντίστροφα, όπως στο Σχήμα 13.

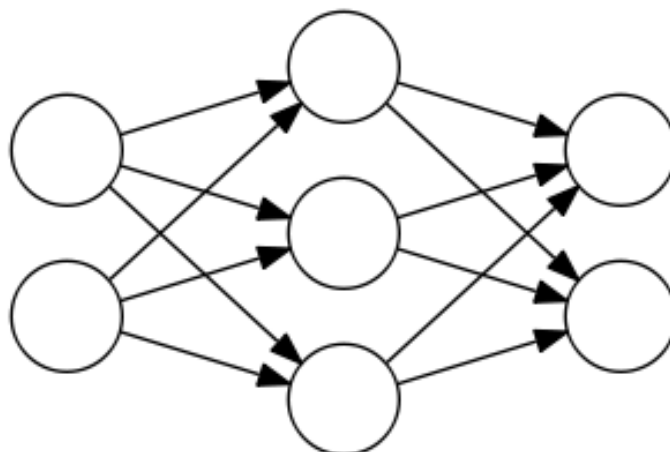


*Σχήμα 13 Εμπρός-Τροφοδότησης δίκτυο με ένα επίπεδο νευρώνων.*

2.Πολλαπλών-Επιπέδων Εμπρός-Τροφοδότησης Δίκτυα.

Εδώ έχουμε περισσότερα του ενός κρυφά επίπεδα, των οποίων οι κόμβοι υπολογισμού ονομάζονται "κρυφοί νευρώνες". Τυπικά, οι νευρώνες σε κάθε επίπεδο έχουν σαν εισόδους τα σήματα εξόδου του προηγούμενου μόνο επιπέδου [15].

Στο σχήμα 14 στην επόμενη σελίδα έχουμε ένα πλήρως συνδεδεμένο νευρωνικό δίκτυο, με την έννοια ότι κάθε κόμβος συνδέεται με όλους τους κόμβους του αμέσως επόμενου επιπέδου.



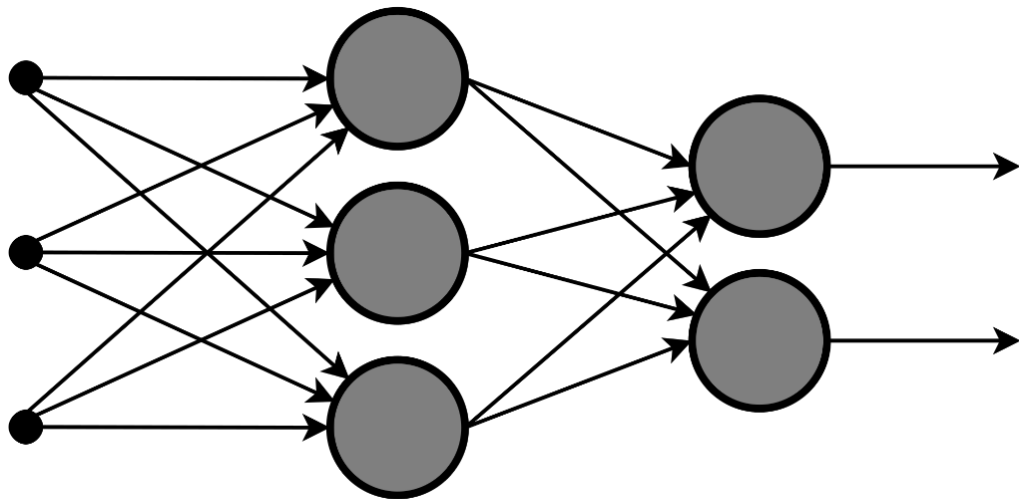
*Σχήμα 14 Πλήρως συνδεδεμένο δίκτυο εμπρός-τροφοδότησης με ένα κρυφό επίπεδο και ένα επίπεδο εξόδου.*

Ένα τέτοιο δίκτυο περιγράφεται συνοπτικά με το συμβολισμό 10-4-2. Αυτός ο συμβολισμός σημαίνει ότι το Ν.Δ. έχει δέκα εισόδους, ένα κρυφό επίπεδο με



τέσσερις νευρώνες (κόμβους) και το επίπεδο εξόδου έχει δύο νευρώνες ή κόμβους. Γενικά, ένα πολυεπίπεδο δίκτυο εμπρός τροφοδότησης με  $p$  εισόδους,  $m$  κρυφά επίπεδα με  $h_j, j=1, \dots, m$  κόμβους ανά επίπεδο και  $n$  κόμβους εξόδου, συμβολίζεται σαν:  $p-h_1, h_2, \dots, h_m-n$ .

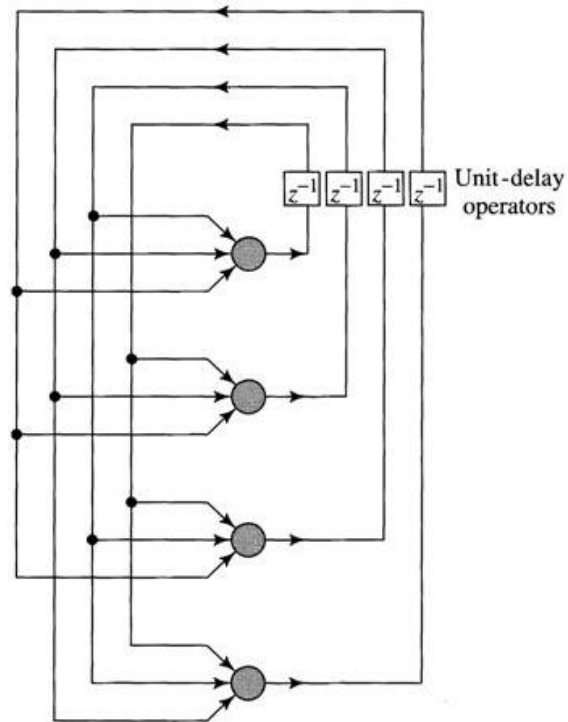
Αντίθετα στο σχήμα 15 έχουμε ένα μερικώς συνδεδεμένο νευρωνικό δίκτυο.



*Σχήμα 15* Μερικώς συνδεδεμένο δίκτυο εμπρός-τροφοδότησης.

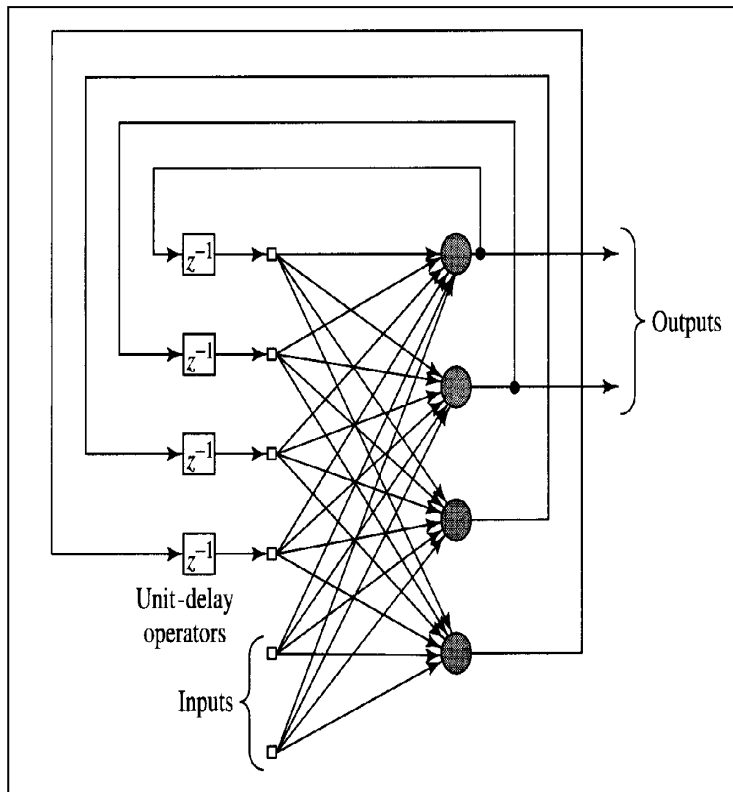
### 3. Αναδρομικά Δίκτυα.

Η διαφορά με τα Δίκτυα Ανατροφοδότησης είναι ότι έχει ένα τουλάχιστον βρόχο ανάδρασης. Στα σχήματα 16 και 17 που ακολουθούν στις επόμενες σελίδες φαίνονται δύο αναδρομικά δίκτυα το πρώτο χωρίς και το δεύτερο με κρυφούς νευρώνες.



*Σχήμα 16 Αναδρομικό δίκτυο χωρίς ανατροφοδότηση και χωρίς κρυφούς νευρώνες.*

Το δίκτυο του σχήματος 16 λέγεται και δίκτυο Hopfield [34], [35]. Είναι μια μη-γραμμική συσχετιστική μνήμη ή μνήμη διευθυνσιοδοτούμενη από τα περιεχόμενα. Η κύρια λειτουργία μιας τέτοιας μνήμης είναι η ανάκτηση ενός προτύπου, που έχει αποθηκευτεί σε αυτήν, σαν απόκριση σε μια ελλιπή ή θορυβώδη έκδοση αυτού του προτύπου.



Σχήμα 17 Αναδρομικό δίκτυο με κρυφούς νευρώνες.

#### 1.4.8 Εκπαίδευση νευρωνικών δικτύων

Στόχος της εκπαίδευσης ενός νευρωνικού δικτύου είναι να υπολογιστούν τα βάρη του (και οι πολώσεις του, εφόσον υπάρχουν) έτσι ώστε να ελαχιστοποιείται το σφάλμα πρόβλεψης στο σύνολο μάθησης [19], [37]. Υπάρχουν πολλές μέθοδοι που μπορούν να χρησιμοποιηθούν για την εκπαίδευση ενός νευρωνικού δικτύου [15], [30], [49]. Στη συνέχεια αναπτύσσεται ο αλγόριθμος εκπαίδευσης με προς τα πίσω διάδοση του σφάλματος ο οποίος δίνει ικανοποιητικά αποτελέσματα για ένα μεγάλο πλήθος προβλημάτων. Κατά την παρουσίαση του αλγόριθμου, γίνεται η θεώρηση ότι οι κρυφοί νευρώνες και οι νευρώνες εξόδου δεν έχουν πολώσεις. Μια τυπική συνάρτηση σφάλματος είναι το άθροισμα τετραγώνων των σφαλμάτων, για  $N$  συνολικό πλήθος προτύπων εκπαίδευσης

$$e = \sum_{i=1}^N (OUT_{forecast,i} - OUT_{actual,i})^2$$

όπου  $OUT_{forecast,i}$  και  $OUT_{actual,i}$  είναι η προβλεπόμενη έξοδος από το νευρωνικό δίκτυο και η επιθυμητή (πραγματική) έξοδος, αντίστοιχα, του  $i$ -οστού προτύπου εκπαίδευσης.

Το σύνολο των  $N$  προτύπων εκπαίδευσης αποτελεί το σύνολο εκπαίδευσης [21], [46]. Οι παρακάτω σχέσεις είναι οι αναδρομικές εξισώσεις εκπαίδευσης, και αποτελούν επίλυση του προβλήματος ελαχιστοποίησης με χρήση του γενικευμένου κανόνα δέλτα:

$$\Delta w_{pj,qk}(n+1) = lr \cdot \delta_{qk} \cdot OUT_{pj} + m \cdot \Delta w_{pj,qk}(n)$$

$$w_{pj,qk}(n+1) = w_{pj,qk}(n) + \Delta w_{pj,qk}(n+1)$$

$N$	ο αύξων αριθμός ανακύκλωσης της διαδικασίας εκπαίδευσης
$w_{pj,qk}(n)$	η τιμή του βάρους που συνδέει το νευρώνα $p$ του στρώματος $j$ με το νευρώνα $q$ του στρώματος $k$ κατά την διάρκεια της ανακύκλωσης $n$
$\Delta w_{pj,qk}(n)$	η μεταβολή του βάρους $w_{pj,qk}(n)$ κατά την διάρκεια της ανακύκλωσης $n$
$\delta_{qk}$	ο συντελεστής μεταβολής $\delta$ του νευρώνα $q$ του στρώματος $k$
$OUT_{pj}$	η έξοδος του νευρώνα $p$ στρώματος $j$
$lr$	ο ρυθμός μάθησης
$m$	ο συντελεστής ορμής

Ο τρόπος υπολογισμού της τιμής του  $\delta$  εξαρτάται από τη θέση του συγκεκριμένου βάρους στο δίκτυο, δηλαδή αν συνδέει νευρώνα του στρώματος εισόδου με κρυφό νευρώνα, ή κρυφό νευρώνα με νευρώνα εξόδου.

$$\delta_{qk} = OUT_{qk} \cdot (1 - OUT_{qk}) \cdot (OUT_{actual,qk} - OUT_{qk})$$

Ο τρόπος υπολογισμού της τιμής του  $\delta$  εξαρτάται από τη θέση του συγκεκριμένου βάρους στο δίκτυο, δηλαδή αν συνδέει νευρώνα του στρώματος εισόδου με κρυφό νευρώνα, ή κρυφό νευρώνα με νευρώνα εξόδου.  $\delta_{qk} = OUT_{qk} \cdot (1 - OUT_{qk}) \cdot (OUT_{actual,qk} - OUT_{qk})$   
 Στη προηγούμενη σχέση το στρώμα  $k$  είναι το στρώμα εξόδου και το  $OUT_{actual,qk}$  είναι η επιθυμητή (πραγματική) τιμή της εξόδου οποιουδήποτε νευρώνα  $q$  του στρώματος  $k$ . Οι τιμές  $OUT_{actual,qk}$  είναι γνωστές από το σύνολο εκπαίδευσης. Η υπολογιζόμενη έξοδος του δικτύου συγκρίνεται με την επιθυμητή και παράγεται ένα σήμα σφάλματος. Το σήμα του σφάλματος διαδίδεται προς τα πίσω μέσω του δικτύου για να ρυθμίσει τα βάρη, όπως φαίνεται από τις προηγούμενες εξισώσεις. Για τους νευρώνες που ανήκουν σε οποιοδήποτε άλλο στρώμα, εκτός από το στρώμα εξόδου, το σφάλμα δεν μπορεί να υπολογιστεί απευθείας επειδή δε δίνεται η επιθυμητή έξοδος για αυτούς τους 'εσωτερικούς' νευρώνες ως τμήμα του συνόλου εκπαίδευσης. Οι τιμές σφάλματος για όλους τους νευρώνες εκτός

των νευρώνων εξόδου υπολογίζονται ως τα βεβαρυσμένα αθροίσματα των σφαλμάτων του στρώματος εξόδου:

$$\delta_{pj} = \text{OUT}_{pj} \cdot (1 - \text{OUT}_{pj}) \cdot \sum_q \delta_{qk} W_{pj,qk}$$

Ο συντελεστής  $1/\sigma$  ονομάζεται ρυθμός μάθησης και ελέγχει το ρυθμό με τον οποίο τα υπολογιζόμενα σφάλματα αλλάζουν τα βάρη του δικτύου. Ο ρυθμός μάθησης συνήθως επιλέγεται μεταξύ 0.01 και 1.0. Ο συντελεστής  $m$  ονομάζεται συντελεστής ορμής και επιτρέπει την αξιοποίηση της πληροφορίας από προηγούμενες τιμές σφάλματος, κατά τη διαδικασία ενημέρωσης των βαρών στη διάρκεια μιας ανακύκλωσης. Ο συντελεστής ορμής βοηθάει στην αποφυγή εγκλωβισμού σε τοπικό ελάχιστο και επιλέγεται μεταξύ 0.01 και 1.0 (αλλά σχεδόν πάντοτε μικρότερος από το ρυθμό μάθησης). Η αναδρομική διαδικασία εκπαίδευσης (τίθεται  $n=n+1$ ) εκτελείται μέχρι ένα συγκεκριμένο αριθμό επαναλήψεων είτε εναλλακτικά μέχρι να επιτευχθεί ικανοποιητική πρόβλεψη των τιμών εξόδου.

#### 1.4.9 Εφαρμογές νευρωνικών δικτύων

Ενδεικτικά παραδείγματα προβλημάτων στα οποία η ανάλυση των νευρωνικών δικτύων έχει εφαρμοστεί με επιτυχία είναι τα εξής [15]:

1. Εφαρμογή αναγνώρισης τραγουδιών Shazam
2. Αναγνώριση ανεπιθύμητων μηνυμάτων (spam detection)
3. Ιατρική διάγνωση: Ένα ευρύ φάσμα ιατρικών εφαρμογών, όπως η καρδιολογικά προβλήματα, τα επίπεδα των διαφόρων ουσιών στο αίμα, αναπνευστικά προβλήματα κτλ
4. Χρηματιστηριακές προβλέψεις: Τα νευρωνικά δίκτυα χρησιμοποιούνται από πολλούς τεχνικούς αναλυτές, ώστε να κάνουν προβλέψεις σχετικά με τις τιμές των μετοχών [39].
5. Παρακολούθηση της κατάστασης των μηχανημάτων: Ένα νευρωνικό δίκτυο, λοιπόν, μπορεί να εκπαιδευτεί με τέτοιο τρόπο, ώστε να διακρίνει από τους ήχους τους οποίους παράγει μια μηχανή είτε αν εκτελεί κανονικά τις λειτουργίες της, είτε βρίσκεται στα πρόθυρα εμφάνισης οποιασδήποτε δυσλειτουργίας [16], [30].
6. Συστήματα διαχείρισης κινητήρα: Τα νευρωνικά δίκτυα έχουν χρησιμοποιηθεί για την ανάλυση των εισροών που δέχονται οι αισθητήρες ενός κινητήρα

#### 1.4.10 Πλεονεκτήματα των νευρωνικών δικτύων

Η χρήση των Νευρωνικών Δικτύων παρέχει τις ακόλουθες χρήσιμες ιδιότητες :

1. **Μη-γραμμικότητα.** Αυτή οφείλεται στο γεγονός ότι ένα Νευρωνικό Δίκτυο δομείται από τη σύνδεση νευρώνων, οι οποίοι είναι μη-γραμμικές συσκευές.
2. **Προσαρμοστικότητα.** Τα Νευρωνικά Δίκτυα έχουν τη δυνατότητα να προσαρμόζουν τα βάρη τους σε οποιαδήποτε αλλαγή [15].
3. **Αντοχή σε σφάλματα.** Ένα νευρωνικό δίκτυο, υλοποιημένο σε υλικό (hardware) έχει τη σημαντική ιδιότητα να είναι “ανεκτικό σε σφάλματα.
4. **Υλοποισιμότητα σε VLSI.** Η παράλληλη φύση του Νευρωνικού Δικτύου, κάνει δυνατή την υλοποίηση του σε VLSI τεχνολογία, έτσι ώστε τα νευρωνικά δίκτυα να μπορούν να χρησιμοποιηθούν σε εφαρμογές πραγματικού χρόνου.
5. **Αναλογία με Νευροβιολογία.** Ο σχεδιασμός νευρωνικών δικτύων γίνεται σε αναλογία με τον εγκέφαλο. Οι νευροβιολόγοι βλέπουν τα νευρωνικά δίκτυα σαν αντικείμενο έρευνας για την εξήγηση νευροβιολογικών φαινομένων. Ομοίως οι μηχανικοί βλέπουν στη νευροβιολογία για νέες ιδέες για την επίλυση πολύπλοκων προβλημάτων.

#### 1.4.11 Μειονεκτήματα των νευρωνικών δικτύων

Στα μειονεκτήματά τους, καταλογίζονται τα εξής:

- Η πιθανότητα απομνημόνευσης των δεδομένων εκπαίδευσης, με αποτέλεσμα να μην αποδώσουν σε νέα δεδομένα το ίδιο καλά (άσχημη γενίκευση).
- Η αυξημένη πιθανότητα να παγιδευτούν σε τοπικό ελάχιστο της επιφάνειας σφάλματος κατά την εκπαίδευση. Αυτή η περίπτωση εξετάζεται αναλυτικά στο επόμενο κεφάλαιο, μαζί με τους κατάλληλους τρόπους για την αντιμετώπισή της [39].

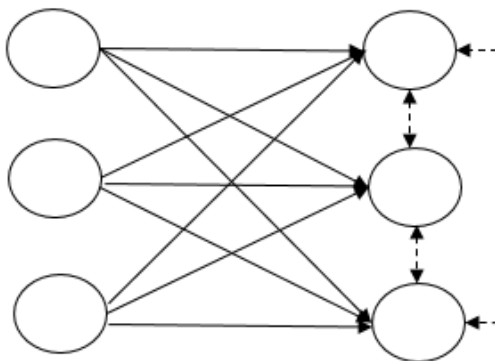
## 1.5 Νευρωνικά δίκτυα Perceptron

### 1.5.1 Εισαγωγή

Το perceptron καταλαμβάνει μια ειδική θέση στην ιστορική εξέλιξη των νευρωνικών δικτύων: ήταν το πρώτο νευρωνικό δίκτυο που μπορούσε να περιγραφεί αλγοριθμικά [15]. Η επινοήση του από τον ψυχολόγο Rosenblatt, ενέπνευσε τους επιστήμονες, ωθώντας τους να αφιερώσουν τις ερευνητικές τους προσπάθειες σε διαφορετικές απόψεις των νευρωνικών δικτύων στις δεκαετίες του 60' και του 70' [16], [30]. Το πιο αξιοσημείωτο, όμως, είναι το γεγονός ότι το perceptron παραμένει τόσο χρήσιμο όσο και ενδιαφέρον όσο ήταν το 1958 όταν πρωτοδημοσιεύτηκε η σχετική εργασία του Rosenblatt.

### 1.5.2 Ορισμός

Ο perceptron είναι ένα δίκτυο με δύο επίπεδα. Το πρώτο επίπεδο απαρτίζεται από τις εισόδους του δικτύου, δεν έχει νευρώνες και επομένως δεν γίνεται καμία επεξεργασία πληροφορίας σε αυτό [16], [30], [33]. Το δεύτερο επίπεδο είναι το επίπεδο εξόδου του δικτύου. Ένα παράδειγμα perceptron δύο επιπέδων, φαίνεται στο Σχήμα 18 που ακολουθεί.



Σχήμα 18 Παράδειγμα perceptron.

Ο στόχος του απλού perceptron είναι να μάθει να λύνει προβλήματα ταξινόμησης, να αντιστοιχεί δηλαδή κάθε σύνολο εισόδων που δέχεται στη σωστή κλάση [32], [34], [45]. Ο perceptron μπορεί να λύσει πολλά τέτοια προβλήματα με επιτυχία. Ένα από τα πλεονεκτήματα του δικτύου αυτού είναι ότι υπάρχει ένας σαφής αλγόριθμος βάσει του οποίου μπορεί να εκπαιδευτεί, ώστε να δίνει σωστά αποτελέσματα [35], [37]. Ο αλγόριθμος αυτός, για την πιο απλή περίπτωση για την οποία τα σύνολα των εισόδων προέρχονται από δύο κλάσεις, έχει ως εξής:

$W_j(n)$  όταν η  $y_j$  είναι σωστή

$$W_j(n+1) = W_j(n) - \eta(n) x_j(n) \text{ όταν } y_j = 1, \text{ ενώ θα έπρεπε να είναι μηδέν 1.6)}$$

$W_j(n) + \eta(n) x_j(n)$  όταν  $y_j = 0$ , ενώ θα έπρεπε να είναι ένα

όπου  $X_j(n)$  το διάνυσμα εισόδου του νευρώνα  $j$ ,  $Y_j(n)$  το διάνυσμα εξόδου και  $W(n)$  το διάνυσμα των βαρών στο βήμα  $n$  του αλγορίθμου,  $W_j(n+1)$  το διάνυσμα των βαρών στο βήμα  $n+1$  και η θετική σταθερά που ονομάζεται παράμετρος ρυθμού εκπαίδευσης (learning-rate parameter).

Το ερώτημα που προκύπτει άμεσα είναι κατά πόσο ο αλγόριθμος αυτός συγκλίνει σε μία σωστή λύση του προβλήματος. Όπως ο ίδιος ο Rosenblatt απέδειξε, αν οι κλάσεις του προβλήματος είναι διαχωρίσιμες από τον perceptron, τότε ο παραπάνω αλγόριθμος συγκλίνει και δίνει σωστά αποτελέσματα και μάλιστα σε πεπερασμένο αριθμό βημάτων. Το θεώρημα αυτό ονομάζεται perceptron convergence theorem. Η ισχύς του θεωρήματος αυτού αποτελεί ένα ακόμα σημαντικό πλεονέκτημα του perceptron.

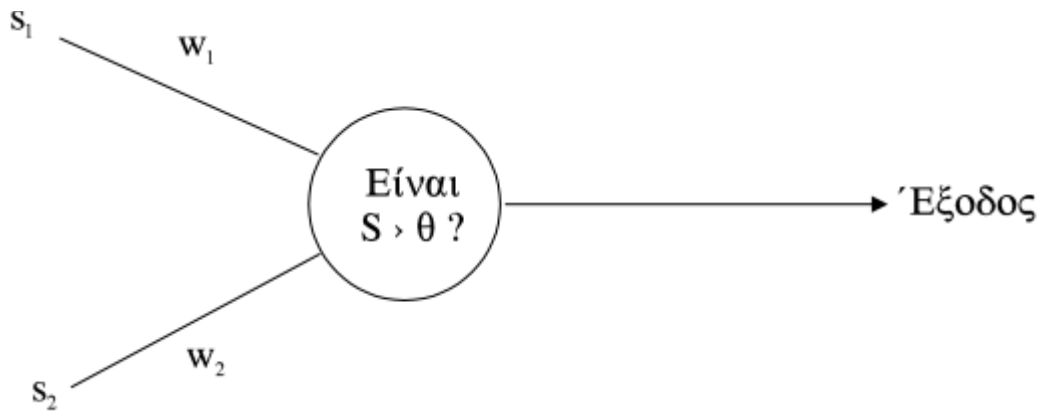
Αργότερα όμως οι Minsky και Papert έδειξαν ότι τα προβλήματα ταξινόμησης που μπορεί να λύσει ο αισθητήρας είναι εκείνα τα οποία είναι γραμμικά διαχωρίσιμα και μόνο. Το συμπέρασμα αυτό γίνεται εύκολα κατανοητό με ένα απλό παράδειγμα. Έστω λοιπόν ότι ο στόχος του αισθητήρα είναι να δίνει σωστά αποτελέσματα σύμφωνα με τη λογική πράξη XOR, η οποία φαίνεται παρακάτω στον σχήμα 19.

Η συνάρτηση X-OR		
Είσοδος 1	Είσοδος 2	Έξοδος
0	0	0
0	1	1
1	0	1
1	1	0

Σχήμα 19 Η πράξη XOR.



Ο κατάλληλος perceptron για το πρόβλημα αυτό είναι εκείνος με δύο εισόδους και έναν νευρώνα εξόδου, όπως φαίνεται στο Σχήμα 20 που ακολουθεί.



Σχήμα 20 Ο perceptron για την XOR.

Η έξοδος του αισθητήρα, όπως είδαμε και παραπάνω, θα δίνεται από τη σχέση:

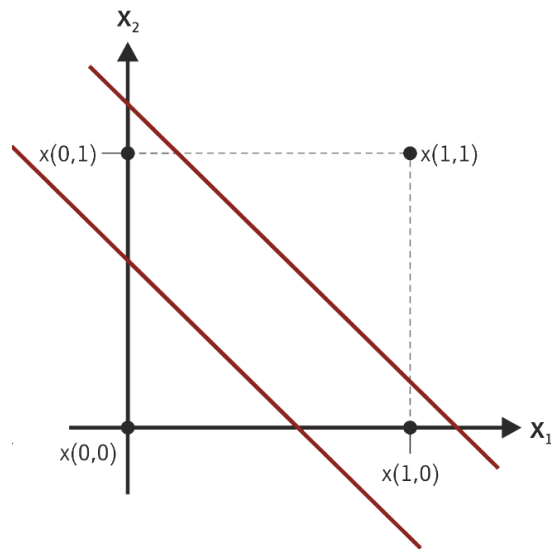
$$0 \text{ αν } w_1 x_1 + w_2 x_2 \leq \theta$$

y =

$$1 \text{ Αλλιώς}$$

όπου  $w_1$  και  $w_2$  τα συναπτικά βάρη και  $\theta$  το κατώφλι. Το ζητούμενο είναι να βρεθούν οι τιμές για τα βάρη, τέτοιες ώστε το αποτέλεσμα των εισόδων να συνάγει με τη λογική πράξη XOR.

Τα τέσσερα σύνολα εισόδων του perceptron αντιστοιχούν σε τέσσερα σημεία στο επίπεδο. Σύμφωνα με το σχήμα 21, τα σημεία (0,0) και (1,1) ανήκουν στην κλάση 0, ενώ τα (0,1) και (1,0) στην κλάση 1. Η εξίσωση 1.5 είναι μία ευθεία που χωρίζει το επίπεδο σε δύο ημιεπίπεδα. Αρκεί λοιπόν να βρεθεί ένα ζεύγος τιμών για τα και τέτοιες ώστε να χωρίζουν τις δύο αυτές κλάσεις. Όπως φαίνεται όμως στο Σχήμα 1.21 παρακάτω, αυτό δεν είναι εφικτό. Δεν υπάρχει καμία ευθεία τέτοια ώστε τα σημεία των δύο κλάσεων να ανήκουν σε διαφορετικά ημιεπίπεδα. Για να το πετύχουμε αυτό χρειαζόμαστε δύο ευθείες. Το πρόβλημα αυτό επομένως δεν είναι γραμμικά διαχωρίσιμο και άρα δεν είναι επιλύσιμο από τον perceptron.



*Σχήμα 21 Το πρόβλημα της XOR είναι μη γραμμικά διαχωρίσιμο.*

Η αδυναμία του perceptron να επιλύσει το πρόβλημα της XOR, δεν είναι αμελητέα. Όπως επεσήμαναν οι Minsky και Papert, πολλά προβλήματα βασίζονται στην XOR. Η αποτυχία αυτή είχε σαν αποτέλεσμα να εγκαταλειφθεί η ιδέα των νευρωνικών δικτύων και να σταματήσει σχεδόν κάθε έρευνα πάνω στο αντικείμενο αυτό [3].

Προκύπτει επομένως το ερώτημα, κατά πόσο θα μπορούσε το αρχικό μοντέλο του αισθητήρα να τροποποιηθεί, ώστε να μπορεί να επιλύει και μη γραμμικά διαχωρίσιμα προβλήματα. Η απάντηση στο ερώτημα αυτό είναι καταφατική. Πράγματι, προσθέτοντας απλά ένα ή περισσότερα επίπεδα νευρώνων μεταξύ του επίπεδου εισόδου και αυτό της εξόδου, ο τροποποιημένος αισθητήρας που προκύπτει μπορεί πλέον να επιλύσει και μη γραμμικά διαχωρίσιμα προβλήματα

### 1.5.3 Perceptron πολλών στρωμάτων

Τα perceptron με ένα στρώμα εισόδου, έχουν περιορισμένες δυνατότητες στην επίλυση πολύπλοκων προβλημάτων εξαιτίας του γραμμικού διαχωρισμού των κατηγοριών του συνόλου εκμάθησης που απαιτείται, ενώ επιπλέον οι χρόνοι σύγκλισης γίνονται πολύ μεγάλοι εάν κάποιο διάνυσμα στην είσοδο διαφέρει πολύ σε μέγεθος από τα υπόλοιπα [15]. Γι' αυτό στις περισσότερες περιπτώσεις χρησιμοποιούνται Perceptrons πολλών στρωμάτων (MLP, Multilayer Perceptrons) [2], [30]. Τα MLP εκπαιδεύονται συνήθως με τον αλγόριθμο της αντίστροφης διάδοσης (Back-propagation). Τα βασικά χαρακτηριστικά των MLP είναι:

1. Οι συναρτήσεις των νευρώνων τους είναι μη γραμμικές και κυρίως συνεχείς και παραγωγίσιμες. Συνήθως χρησιμοποιούνται σιγμοειδείς συναρτήσεις όπως η λογιστική και η συνάρτηση  $\tanh(x)$ . Από αυτές τις δύο, η λογιστική προσομοιώνει καλύτερα τη λειτουργία των φυσικών νευρώνων [32].
2. Το δίκτυο αποτελείται από ένα ή περισσότερα κρυμμένα στρώματα, τα οποία του προσδίδουν χαρακτηριστική ευελιξία.
3. Η έξοδός τους δηλαδή, είναι συνάρτηση αποκλειστικά και μόνο της παρούσας εισόδου.

Η επιλογή μεγέθους του δικτύου είναι εξαιρετικά δύσκολη κατά το σχεδιασμό τους, καθώς μικρό μέγεθος μπορεί να αποδειχθεί ανεπαρκές για την επίλυση του προβλήματος, ενώ ένα πολύ μεγάλο μέγεθος, να δώσει χειρότερα αποτελέσματα επειδή μπορεί απλώς να απομνημονεύσει τις εισόδους [11]. Μέθοδοι οι οποίες έχουν αναπτυχθεί για τον καθορισμό του μεγέθους ενός τεχνητού νευρωνικού δικτύου είναι οι εξής:

- Επιλογή μεγέθους, έπειτα από δοκιμές εκπαίδευσης δικτύων διαφορετικού αριθμού νευρώνων και δομής και σύγκριση της εξόδου τους με την επιθυμητή έξοδο ξεκινώντας από το μικρότερο δυνατό δίκτυο [36], [45].
- Προσπάθεια αύξησης του μεγέθους, προσθέτοντας νευρώνες ανάλογα με τις ανάγκες του προβλήματος κατά τη διάρκεια της εκπαίδευσης.
- Μέθοδος κατά την οποία ξεκινώντας με αφετηρία δίκτυο μεγάλου μεγέθους, προχωρούμε καταστρέφοντας τις συνάψεις, που συνεισφέρουν ελάχιστα ή καθόλου στη διαδικασία εκμάθησης.

#### 1.5.4 Μειονεκτήματα των MLP – Εφαρμογές

Τα κυριότερα μειονεκτήματα των MLP είναι:

- Η δύσκολη θεωρητική τους ανάλυση, εξαιτίας της μη γραμμικότητας, της υψηλής συνεκτικότητας και του έντονου παραλληλισμού που παρουσιάζουν [36].
- Η απουσία μιας άμεσης και ολοκληρωμένης εικόνας της διαδικασίας εκμάθησης, που οφείλεται στην παρουσία κρυμμένων στρωμάτων [16].

Παρ' όλα αυτά, βρίσκουν εφαρμογή σε μια τεράστια ποικιλία προβλημάτων που καλύπτουν σχεδόν όλους τους τομείς του σύγχρονου τρόπου ζωής [30], [45]. Οι σημαντικότερες από αυτές τις εφαρμογές είναι:

- ✓ Αναγνώριση ομιλίας
- ✓ Αναγνώριση χειρόγραφων χαρακτήρων
- ✓ Αναγνώριση προτύπων
- ✓ Αυτόματος έλεγχος
- ✓ Ιατρική διάγνωση, κυρίως όσον αφορά τη διάγνωση καρδιακών επεισοδίων
- ✓ Ανίχνευση σήματος ραντάρ και αναγνώριση στόχου

Επιπλέον, υπάρχει και μια σειρά συνεχώς αναπτυσσόμενων εφαρμογών, σε τομείς όπως:

- ✓ Αεροναυπηγική
- ✓ Αυτοκινητοβιομηχανία
- ✓ Αμυντικά συστήματα
- ✓ Ηλεκτρονική
- ✓ Διασκέδαση-Εικόνα-Ήχος-Κινηματογράφος
- ✓ Τραπεζικό σύστημα και οικονομία
- ✓ Εκμετάλλευση πετρελαίου και παραγώγων του
- ✓ Ρομποτική
- ✓ Τηλεπικοινωνίες
- ✓ Μεταφορές

## 2 Τεχνολογίες και Εργαλεία Υλοποίησης Εφαρμογής

### 2.1 Διαδικτυακός εξυπηρετητής Ubuntu Linux Server

Στην ενότητα αυτή γίνεται περιγραφή του λογισμικού Linux, του Ubuntu Server, του διακομιστή Apache HTTP Server, της γλώσσας προγραμματισμού PHP, καθώς και της σύνδεσης μεταξύ των προαναφερθέντων.

#### 2.1.1 Linux

Το Linux είναι ένα σύγχρονο λειτουργικό σύστημα ανοιχτού κώδικα, που βασίζεται στις βασικές σχεδιαστικές αρχές και την πα-ράδοση του Unix. Αναπτύχθηκε αρχικά από τον Linus Torvalds για τον επεξεργαστή i386 της Intel αλλά έχει μεταφερθεί έως σήμερα σε πολλές διαφορετικές αρχιτεκτονικές ( IA-64, MIPS, Alpha, SPARC, PowerPC κ.α. ). Το Linux είναι ένα μονολιθικό λειτουργικό σύστημα, όπως και τα περισσότερα συστήματα που βασίζονται στο Unix : οι περισσότερες λειτουργίες του υλοποιούνται σε ένα πρόγραμμα, που ονομάζεται πυρήνας του Λ.Σ, το οποίο εκτελείται σε ένα ενιαίο χώρο διευθύνσεων και έχει πλήρη ελεύθερη πρόσβαση στο υλικό . Ο πυρήνας του Linux αναλαμβάνει το σύνολο σχεδόν των λειτουργιών που χαρακτηρίζουν ένα σύγχρονο Λ.Σ.: Διαχείριση CPU, χρονοδρομολόγηση διεργασιών, διαχείριση μνήμης, διαχείριση συσκευών I/O, διαχείριση συστημάτων αρχείων, δια δικτύωση. Στον πυρήνα του Linux βρίσκονται επίσης και οι οδηγοί συσκευών, που επιτρέπουν τον έλεγχο των διαφόρων περιφερειακών συσκευών. Το ιδιαίτερο χαρακτηριστικό του είναι πως διανέμεται σε πολλές και διαφορετικές εκδόσεις αφού ο κάθε ένας μπορεί απλώς να αναδιανείμει τη βασική του έκδοση. Το Ubuntu είναι μία από αυτές.

## Ιστορικά στοιχεία του Ubuntu

Το Ubuntu σχεδιάστηκε το 2004 από τον Mark Shuttleworth, έναν Νοτιοαφρικανό επιχειρηματία και την εταιρεία του Canonical. Με το σύστημα του Debian ως βάση, ο Shuttleworth ξεκίνησε να χτίζει το Ubuntu. Μερικά σημαντικά παραδείγματα είναι πως το 2007 η Dell ξεκίνησε μια συνεργασία με την Canonical για την πώληση υπολογιστών με προ εγκατεστημένο το Ubuntu. Επιπλέον, το 2005, η γαλλική αστυνομία άρχισε να μεταβιβάζει ολόκληρη την υπολογιστική της υποδομή σε μία παραλλαγή του Ubuntu. Το Ubuntu λοιπόν είναι μία ελεύθερη, ανοικτού λογισμικού και δωρεάν διανομή του Linux με αξιοπιστία και ασφάλεια τόσο για σταθερούς υπολογιστές όσο και για τις ανάγκες των εξυπηρετητών.

### 2.1.2 Ubuntu Server

“Χτισμένο πάνω στη σταθερή βάση του Debian - γνωστό για τις στιβαρές του εγκαταστάσεις server - το Ubuntu Server Edition έχει μια ισχυρή κληρονομιά αξιοπιστίας απόδοσης και προβλέψιμης ανάπτυξης. Η έκδοση Ubuntu Server Edition 14.04 LTS, με υποστήριξη μακράς διάρκειας, προσφέρει υποστήριξη για διάφορες συνήθεις διαμορφώσεις, φέρνοντας στις εγκαταστάσεις server την ευκολία χρήσης που έκανε το Ubuntu διάσημο στη χρήση γραφείου. Το νέο Ubuntu Server Edition παρέχει μια ολοκληρωμένη πλατφόρμα για την εγκατάσταση νέων server με υπηρεσίες διαδικτύου, όπως mail, web, DNS, file serving ή database management. Υποστηρίζεται από την Canonical, σαν πλατφόρμα έχει πιστοποιηθεί AWS, Microsoft Azure, Joyent, IBM, HP Cloud κα. υποστηρίζοντας τις ακόλουθες αρχιτεκτονικές: x86, x86-64, ARM v7, Arm64 και Power.



*Εικόνα 1 Ubuntu Server.*

### 2.1.3 Ο διακομιστής Apache HTTP Server

Ο Apache HTTP Server είναι ο πλέον διαδεδομένος εξυπηρετητής του παγκόσμιου ιστού. Ο ρόλος του apache είναι να αναμένει αιτήσεις από διάφορα προγράμματα – χρήστες (clients) όπως είναι ένας ο φυλλομετρητής (browser) ενός χρήστη και στη συνέχεια να εξυπηρετεί αυτές τις αιτήσεις «σερβίροντας» τις σελίδες που ζητούν είτε απευθείας μέσω μιας ηλεκτρονικής διεύθυνσης (URL), είτε μέσω ενός συνδέσμου (link). Ο τρόπος με τον οποίο ο Apache εξυπηρετεί αυτές τις αιτήσεις, είναι σύμφωνα με τα πρότυπα που ορίζει το πρωτόκολλο HTTP (HyperText Transfer Protocol). Ο Apache πρωτοεμφανίστηκε στα μέσα της δεκαετίας του 90' και σταδιακά επικράτησε, αποτελώντας ακρογωνιαίο λίθο στην ανάπτυξη του παγκόσμιου ιστού. Ο Apache είναι έργο της κοινότητας ανοικτού λογισμικού του Apache Software Foundation. Είναι ένα πρόγραμμα ανοικτού κώδικα και διατίθεται ελεύθερα με την άδεια χρήσης Apache License. Τα βασικά χαρακτηριστικά που καθιέρωσαν τον Apache είναι:

- Διατίθεται για εγκατάσταση σε πάρα πολλά λειτουργικά συστήματα, όπως Linux, Unix, Microsoft Windows, Solaris, Mac OS X, Microsoft Windows, NetWare, FreeBSD, OS/2 κ.ά.
- Είναι δομημένο κατά τμήματα (modules). Μόνο οι βασικές λειτουργίες παρέχονται στο κυρίως πρόγραμμα και οι υπόλοιπες μπορούν να προστεθούν δυναμικά, σαν επιπλέον modules.
- Υποστηρίζει αρκετές γλώσσες προγραμματισμού όπως PHP, Perl, Python, MySQL κ.ά.
- Υποστηρίζει τα πρωτόκολλα IPv6 και SSL.
- Έχει πολύ μικρές απαιτήσεις σε πόρους συστήματος και αποκρίνεται ικανοποιητικά ακόμα και σε μεγάλες αυξήσεις φορτίου (συνδέσεις χρηστών).

## Ιστορία Apache

Η πρώτη του έκδοση, γνωστή ως NCSA HTTPd, δημιουργήθηκε από τον Robert McCool και κυκλοφόρησε το 1993. Θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του παγκόσμιου ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε απέναντι στον εξυπηρετητή http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται άλλους εξυπηρετητές βασισμένους στο [Unix](#) σε λειτουργικότητα και απόδοση. Από το 1996 ήταν από τους πιο δημοφιλείς όμως από τον Μάρτιο του 2006 έχει μειωθεί το ποσοστό της εγκατάστασής του κυρίως από τον Microsoft Internet Information Services και την πλατφόρμα .NET. Τον Οκτώβριο του 2007 το μερίδιο του ήταν 47.73% από όλους τους ιστοτόπους. Τον Μάρτιο του 2017, το 49,48% του συνόλου των καταχωρισμένων Ελληνικών τομέων χρησιμοποιούσε το Apache.

Όταν ο Robert McCool έφυγε από το NCSA, το 1994, η ανάπτυξη του HTTPd σταμάτησε. Ο Robert McCool Βοηθήθηκε από διάφορους άλλους προγραμματιστές με τους οποίους δημιούργησε το αρχικό “World Wide Web Group” Brian Behlendorf, Roy T. Fielding, Rob Hartill, David Robinson, Cliff Skolnick, Randy Terbush, Robert S. Thau, Andrew Wilson, Eric Hagberg, Frank Peters και Nicolas Pioch.

Η δεύτερη έκδοση του World Wide Web ήταν ουσιαστικά η γραφή του κώδικα του World Wide Web..., εστιάζοντας περισσότερο στην καλύτερη άρθρωση του και στην ανάπτυξη ενός layer φορητότητας του “World Wide Web portable runtime”. Η έκδοση 2 του πυρήνα του World Wide Web έχει αρκετά σημαντικές βελτιώσεις σε σχέση με την έκδοση 1χ. αυτές συμπεριλαμβάνουν την υποστήριξη σε UNIX καλύτερη υποστήριξη και για άλλες πλατφόρμες εκτός από UNIX (όπως η Microsoft Windows) και υποστήριξη για IPv6. Η έκδοση 2.2 εισήγαγε ένα νέο API εξουσιοδότησης το οποίο επιτρέπει μεγαλύτερη ευελιξία.



## 2.2 Η γλώσσα HTML (HyperText Markup Language)

Η HTML (HyperText Markup Language: Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες. Η HTML δεν είναι μια γλώσσα προγραμματισμού αλλά μια περιγραφική γλώσσα, δηλαδή ένας ειδικός τρόπος γραφής κειμένου και κλήσης άλλων αρχείων ή εφαρμογών βασισμένος σε ετικέτες (tags). Οι ετικέτες αυτές περικλείονται από τα σύμβολα < και > μέσα στο περιεχόμενο της ιστοσελίδας ενώ συνήθως λειτουργούν ανά ζεύγη δηλαδή αποτελούνται από μια ετικέτα αρχής και μια ετικέτα τέλους (για παράδειγμα <html> και </html> αντίστοιχα). Ανάμεσα στις ετικέτες αυτές, οι σχεδιαστές ιστοσελίδων μπορούν να προσθέσουν περιεχόμενο στις ιστοσελίδες όπως κείμενο, πίνακες, εικόνες κλπ. Ο σκοπός ενός φυλλομετρητή σελίδων είναι να διαβάσει τα έγγραφα HTML και να τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο φυλλομετρητής δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας. Αξιοσημείωτο είναι επίσης το γεγονός ότι η HTML περιλαμβάνει ετικέτες όχι μόνο για την προσθήκη περιεχομένου στους ιστότοπους (όπως εικόνες, πίνακες, λίστες, συνδέσμους κ.ά.) αλλά και ετικέτες για την μορφοποίηση αυτών.

### 2.2.1 HTML 5

Η HTML 5 είναι η νέα έκδοση της γλώσσας προγραμματισμού HTML. Το W3C (World Wide Web Consortium) και το WHATWG (Web HyperText Application Technology Working Group) συνεργάστηκαν και δημιούργησαν αυτή την νέα γλώσσα. Είναι ακόμα υπό ανάπτυξη ακόμη και όταν ετοιμαστεί πλήρως θα είναι η επόμενη μεγάλη έκδοση της HTML. Προορίζεται για αντικατάσταση της HTML 4.0.1, της XHTML 1.0 και της DOM 2 HTML. Επίσης προορίζεται για την καλύτερη λειτουργία εντοπισμού λαθών, την πλήρη συμβατότητα ανεξαρτήτως συσκευής και την αντικατάσταση του scripting. Επιπλέον η μείωση της ανάγκης για ιδιότητα plug-in και άλλες διαδικτυακές εφαρμογές (RIA) είναι άλλος ένας βασικός κανόνας που έχει οριστεί για την HTML 5. Βάση για τα νέα χαρακτηριστικά είναι οι HTML, CSS, DOM και JavaScript. Περιέχει το πρότυπο Web Forms 2.0.

Μερικά από τα πλεονεκτήματα χρήσης HTML5 είναι:

- Δυνατότητα σχεδιασμού γραφικών με χρήση JavaScript
- Αναπαραγωγή βίντεο και audio χωρίς να χρειάζεται plugins

- Προσθήκη νέων ετικετών που κάνουν την δημιουργία και την διαχείριση των ιστοσελίδων, ακόμη πιο εύκολη.
- Νέα στοιχεία στις HTML φόρμες (calendar, date, time, search κτλ.)
- Επιτρέπει την δημιουργία mobile web sites αξιοποιώντας στο έπακρο τις τεχνικές δυνατότητες των mobile συσκευών.
- Κάνει χρήση του Geolocation (αισθητήρα GPS) και έτσι ο χρήστης μπορεί να χρησιμοποιήσει διαδικτυακές εφαρμογές πιο γρήγορα από ότι θα έκανε συμπληρώνοντας το αντίστοιχο πεδίο τοποθεσίας.
- Επιτρέπει την αναπαραγωγή βίντεο σε mobile web sites χωρίς τη χρήση Flash.
- Προσφέρει την δυνατότητα χειρισμού δεδομένων ακόμα και όταν δεν υπάρχει σύνδεση με το διαδίκτυο.
- Επιτρέπει την χρήση Vector γραφικών δηλαδή μπορούμε να σχεδιάσουμε απευθείας στο περιηγητή ιστοσελίδων (browser) με την χρήση κώδικα, ενώ προηγουμένως γινόταν με embed αρχεία (εικόνες, γραφήματα κ.α.).

## 2.2.2 Η ιστορία της HTML

Γενικά η HTML διαθέτει ένα πεπερασμένο αριθμό ετικετών που μπορούμε να χρησιμοποιήσουμε. Ωστόσο ο αριθμός αυτός δεν παραμένει σταθερός. Κατά διαστήματα το W3C (World Wide Web Consortium), το οποίο ανέπτυξε και διαχειρίζεται τα πρότυπα της HTML, δημοσιεύει νέα πρότυπα στα οποία προσθέτει καινούργιες ετικέτες που καλύπτουν ή διορθώνουν μια λειτουργία στο προηγούμενο πρότυπο. Η τελευταία αναθεώρηση του HTML προτύπου είναι η HTML5.

Το 1980, ο φυσικός Τιμ Μπέρνερς Λι, ο οποίος δούλευε στο CERN, επινόησε το ENQUIRE, ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN, και δημιούργησε ένα πρωτότυπό του. Αργότερα, το 1989, πρότεινε ένα σύστημα βασισμένο στο διαδίκτυο, το οποίο θα χρησιμοποιούσε υπερκείμενο. Έτσι, έφτιαξε την προδιαγραφή της HTML και έγραψε τον browser και το λογισμικό εξυπηρετητή στα τέλη του 1990.

Η πρώτη δημόσια διαθέσιμη περιγραφή της HTML ήταν ένα έγγραφο και το ονόμασε Ετικέτες HTML, το οποίο πρωτοαναφέρθηκε στο Διαδίκτυο από τον Μπέρνερς Λι στα τέλη του 1991. Περιέγραφε τα 20 στοιχεία τα οποία αποτελούσαν τον αρχικό και απλό σχεδιασμό της HTML. Εκτός από την ετικέτα υπερσυνδέσμου, οι υπόλοιπες ήταν επηρεασμένες από

την SGMLguid, μια μορφή δημιουργίας τεκμηρίωσης, φτιαγμένη στο CERN και βασισμένη στην SGML.

Μετά που τα πρόχειρα HTML και HTML+ τερματιστήκαν, στις αρχές του 1994, το IETF δημιούργησε την Ομάδα Εργασίας για την HTML, και το 1995 ολοκλήρωσε την «HTML 2.0», με την πρόθεση να αποτελέσει την πρώτη προδιαγραφή πάνω στην οποία θα βασίζονταν οι μελλοντικές υλοποιήσεις. Η HTML 2.0 δημοσιεύτηκε ως RFC 1866, και περιείχε ιδέες από τα πρόχειρα HTML και HTML+. Ο σκοπός της αρίθμησης 2.0 ήταν απλά για να ξεχωρίσει την νέα έκδοση από τα πρόχειρα που είχαν προηγηθεί.

Η περαιτέρω ανάπτυξη κάτω από την επιτήρηση του IETF καθυστέρησε λόγω των ενδιαφερόντων. Από το 1996 και μετά, οι προδιαγραφές της HTML τηρούνται, μαζί με ανατροφοδότηση από τους δημιουργούς λογισμικού, από το World Wide Web Consortium (W3C). Ωστόσο, το 2000 η HTML έγινε επίσης παγκόσμιο πρότυπο. Η HTML 4.01 δημοσιεύτηκε από το W3C το 1999, και το 2001 δημοσιεύτηκαν επίσης και τα λάθη και οι παραλείψεις της, το 2008 δημοσιεύτηκε η HTML5 ως πρόχειρο εργασία.

## **2.3 CSS(Cascading Style Sheets)**

Η CSS ή αλληλουχία φύλλων στυλ στα ελληνικά είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Με πιο απλά λόγια χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XML. Ελέγχει δηλαδή την εμφάνιση μιας ιστοσελίδας και γενικότερα ενός ιστότοπου. Είναι η γλώσσα αυτή που κρίνεται απαραίτητη όταν θέλουμε να δημιουργήσουμε μια όμορφη και καλοσχεδιασμένη ιστοσελίδα. Είναι προορισμένη να αναπτύσσει στιλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την HTML.

Εκτός από την ευκολία στην διαχείριση ενός site, ένα άλλο σημαντικό πλεονέκτημα της χρήσης CSS στις σελίδες είναι ο "καθαρότερος" κώδικας, χωρίς πολλές ιδιότητες στις ετικέτες οι οποίες τον κάνουν δυσανάγνωστο. Επιπλέον κάνει την πλοήγηση γρηγορότερη καθώς το αρχείο, μέσα στο οποίο ορίζονται τα στυλ, "διαβάζεται" από τον browser μόνο μια φορά και έπειτα αποθηκεύεται στην cache memory, μειώνοντας έτσι το μέγεθος της πληροφορίας που γίνεται download από τους browsers. Ένα μειονέκτημα των φύλλων CSS είναι το γεγονός ότι κάποιοι παλιοί φυλλομετρητές δεν τα καταλαβαίνουν ή τα

παρερμηνεύουν. Η CSS λειτουργεί μαζί με τον HTML κώδικα και αυτό σημαίνει ότι πρέπει όποιος θέλει να ασχοληθεί με αυτή την γλώσσα πρέπει να γνωρίζει καλά την γλώσσα HTML.

Τα κύρια στοιχεία ενός κανόνα CSS είναι:

Ο επιλογέας (selector): συνήθως είναι μια ετικέτα της HTML που καθορίζει που θα εφαρμοστούν οι κανόνες CSS.

Οι δηλώσεις (declarations): αποτελούνται από τις ιδιότητες που χαρακτηρίζουν τον επιλογέα και τις τιμές που παίρνουν αυτές.

## 2.4 Βάσεις δεδομένων

Με τον όρο βάση δεδομένων εννοείται μία συλλογή από συστηματικά μορφοποιημένα σχετιζόμενα δεδομένα στα οποία είναι δυνατή η ανάκτηση δεδομένων μέσω αναζήτησης κατ' απαίτηση. Ένας άλλος ορισμός είναι ότι μια βάση δεδομένων είναι ένα ολοκληρωμένο σύστημα που αποτελείται από δεδομένα (data) και από το κατάλληλο λογισμικό (software), τα οποία χρησιμοποιώντας το υλικό (hardware) βοηθούν στην ενημέρωση και πληροφόρηση των χρηστών.

### 2.4.1 Βασικές έννοιες τις βάσεις δεδομένων

Αποτελείται από:

- Πεδία (**field**): Μια βάση δεδομένων αποτελείται από μια σειρά στοιχείων με πιο βασικό στοιχείο το πεδίο. Το πεδίο αντιστοιχεί σε ένα δεδομένο και είναι το συστατικό στοιχείο μιας εγγραφής (record). Παραδείγματα πεδίων είναι ένα όνομα πελάτη, μια διεύθυνση κ.α.
- Εγγραφή είναι μια συλλογή από πεδία.
- Πίνακας (**table**): Μια ενότητα από συσχετιζόμενες εγγραφές οι οποίες έχουν τον ίδιο αριθμό πεδίων αποτελούν έναν πίνακα. Στον πίνακα οι εγγραφές αντιστοιχούν στις γραμμές (rows) και τα πεδία στις στήλες του πίνακα (columns).
- Σχεσιακή Βάση Δεδομένων (**relational data base**): Ένα σύνολο συσχετιζόμενων πινάκων αποτελούν μία σχεσιακή βάση δεδομένων η οποία αποτελεί και τον πιο συνήθη τρόπο δημιουργίας βάσεων δεδομένων σήμερα.
- Πρωτεύον Κλειδί (**primary key**): Το πρωτεύον κλειδί είναι μοναδικός αριθμός αναγνώρισης της κάθε εγγραφής ενός πίνακα και δεν είναι δυνατή η ύπαρξη δύο

εγγραφών με το ίδιο κλειδί.

- Δευτερεύον Κλειδί (**foreign key**): Δευτερεύοντα κλειδιά είναι κοινά πεδία μεταξύ των εγγραφών των πινάκων μιας βάσης δεδομένων τα οποία μπορούμε να χρησιμοποιήσουμε προκειμένου να κάνουμε αναζήτηση πληροφοριών στους πίνακες αυτούς. Το δευτερεύον κλειδί είναι πεδίο ενός πίνακα το οποίο προσδιορίζει εγγραφές ενός άλλου πίνακα στον παρόντα πίνακα.
- Ερωτήματα (**queries**): Η αναζήτηση εγγραφών σε μία βάση δεδομένων γίνεται με ερωτήματα.

Για την αναζήτηση εγγραφών σε μία βάση δεδομένων χρησιμοποιείται η γλώσσα Structured Query Language (SQL).

#### 2.4.2 Συστήματα διαχείρισης βάσεων δεδομένων(DBMS)

Το ΣΔΒΔ είναι ένα σύνολο προγραμμάτων και ρουτινών, που σκοπό έχουν το χειρισμό της βάσης, όσον αφορά τη δημιουργία, συντήρηση, επεξεργασία στοιχείων, ελέγχους ασφαλείας κτλ., και την εξυπηρέτηση των χρηστών, όσον αφορά την παροχή στοιχείων και πληροφοριών, χωρίς αυτοί να πρέπει να ασχολούνται με το πώς και το πού τα δεδομένα είναι αποθηκευμένα στη βάση.

#### 2.4.3 SQL- MySQL

SQL ((StructureQueryLanguage) είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων η οποία, αρχικά, βασίστηκε στη σχεσιακή άλγεβρα. Η γλώσσα περιλαμβάνει δυνατότητες ανάκτησης και ενημέρωσης δεδομένων, δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου πρόσβασης στα δεδομένα. Η SQL ήταν μία από τις πρώτες γλώσσες για το σχεσιακό μοντέλο του Edgar F. Codd, στο σημαντικό άρθρο του το 1970, και έγινε η πιο ευρέως χρησιμοποιούμενη γλώσσα για τις σχεσιακές βάσεις δεδομένων. Η SQL, χρησιμοποιεί μία συλλογή από πρότυπα - εντολές οι οποίες δίνουν την δυνατότητα στον χρήστη να διαχειριστεί μια βάση δεδομένων, προσφέροντας λειτουργίες όπως: προσθήκη (INSERT), προβολή (SELECT), ταξινόμηση (ORDER BY), τροποποίηση (UPDATE), διαγραφή (DELETE) των ορισμάτων που ενδεχομένως να υπάρχουν σε κάποιο πίνακα.

## MySQL

Η MySQL είναι βάση δεδομένων μέσα στην οποία μπορούμε να καταχωρούμε, επεξεργαστούμε, αναζητούμε και ταξινομούμε δεδομένα. Είναι σχεσιακό σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων RDBMS (Relational Database Query Language) της οποίας ο πηγαίος κώδικας είναι διαθέσιμος με άδεια χρήσης της GNU (General Public License) και εφευρέθηκε από τον Michael Widenious για τον οποίο λέγεται ότι έδωσε το όνομα της κόρης του 'My' στην εφαρμογή.

Η εφαρμογή ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρεία, τη σουηδική MySQL AB, της οποίας ιδιοκτήτης είναι η Oracle Corporation. Η MySQL είναι βασισμένη στη γλώσσα ερωταποκρίσεων (SQL), οπότε και οι πρότυπα-εντολές της SQL που αναφέρθηκαν αναλυτικώς παραπάνω μπορούν να χρησιμοποιηθούν με την MySQL, οπότε θεωρητικά ένας κώδικας SQL θα πρέπει να τρέχει με τον ίδιο τρόπο και σε μια βάση MySQL.

Η MySQL είναι σήμερα πολύ δημοφιλής βάση δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες και αυτό οφείλεται στο γεγονός ότι η MySQL είναι σχεσιακή, που συνεπάγεται ότι η οργάνωση των δεδομένων γίνεται σε διαφορετικούς πίνακες οι οποίοι σχετίζονται μεταξύ τους με κάποιο σαφώς ορισμένο τρόπο. Χρησιμοποιείται σε κάποιες από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες, όπως Flickr, YouTube, Wikipedia, Google, Facebook και Twitter αλλά και διατίθεται συνήθως ως μέρος ελεύθερων πακέτων λογισμικού πχ XAMPP.

### Πλεονεκτήματα χρήσης MySQL

- Απόδοση: είναι αρκετά γρήγορη.
- Κόστος: διατίθεται δωρεάν για προσωπική χρήση. Η εμπορική άδεια της διατίθεται σε χαμηλό κόστος.
- Ευκολία χρήσης: εύκολη στην εκμάθηση της
- Μεταφερσιμότητα: μπορεί να χρησιμοποιηθεί σε πολλά σύγχρονα λειτουργικά συστήματα
- Πηγαίος Κώδικας: ανήκει στην οικογένεια του λογισμικού ανοικτού κώδικα
- Υποστήριξη: Στην κεντρική της σελίδα [www.mysql.com](http://www.mysql.com) μπορείς να βρεις βοήθεια σε πιθανά προβλήματα.

## 2.5 phpMyAdmin

Το phpMyAdmin είναι μία εφαρμογή ανοικτού κώδικα. Η phpMyAdmin είναι ένα εργαλείο γραμμένο σε PHP που η χρησιμότητα του είναι να χειρίζεται τη MySQL μέσω προγράμματος web. Οι λειτουργίες του είναι η δημιουργία, η τροποποίηση ή διαγραφή δεδομένων, πίνακες, πεδία ή σειρές, εκτέλεση SQL δηλώσεις, ή τη διαχείριση των χρηστών και τα δικαιώματα. Ακόμα υποστηρίζει 78 γλώσσες μεταξύ των οποίων και τα Ελληνικά. Το phpMyAdmin μπορεί να διαχειριστεί ένα ολόκληρο mysql server ή ακόμα και απλές βάσεις δεδομένων όπου ο κάθε χρήστης έχει ένα λογαριασμό και μπορεί να δημιουργήσει και να διαχειριστεί τις δικές του βάσεις δεδομένων (τροποποίηση ή διαγραφή βάσεων δεδομένων, πινάκων, πεδίων ή γραμμών) αλλά και να εκτελέσει SQL ερωτήματα.

### 2.5.1 Χαρακτηριστικά του phpMyAdmin

Χαρακτηριστικά του phpMyAdmin είναι οι εξής:

- Η διασύνδεση Web
- MySQL διαχείρισης βάσεων δεδομένων
- Υποστηρίζει 47 γλώσσες μεταξύ των οποίων είναι και τα Ελληνικά
- Εισαγωγή δεδομένων από CSV και SQL
- Εξαγωγή δεδομένων σε διάφορες μορφές: CSV , SQL , XML , PDF , ISO / IEC 26300 - Κείμενο OpenDocument και φύλλου, το Word, το Excel, LaTeX και άλλοι
- Χορήγηση πολλούς διακομιστές
- Δημιουργία PDF γραφικών της διάταξης βάσης δεδομένων
- Φορτώνει αρχεία κειμένου σε πίνακες
- Εκτελεί ερωτήματα SQL
- Υποστηρίζει MySQLi , μια βελτιωμένη επέκταση του MySQLPhpMy Admin
- Δημιουργία σύνθετων ερωτημάτων χρησιμοποιώντας το Query-by-παραδείγματος (QBE)
- Ψάχνοντας σε παγκόσμιο επίπεδο σε μια βάση δεδομένων ή ένα υποσύνολο αυτό
- Μετασχηματισμός δεδομένων σε οποιαδήποτε μορφή, χρησιμοποιώντας ένα σύνολο από ορισμένες λειτουργίες, όπως η εμφάνιση BLOB -τα δεδομένα ως εικόνα ή download-link
- Ζωντανά διαγράμματα για την παρακολούθηση της δραστηριότητας του διακομιστή MySQL, όπως συνδέσεις, τις διαδικασίες, CPU / χρήση μνήμης, κ.λπ.

- Εργασία με διαφορετικά λειτουργικά συστήματα

## **2.5.2 Πλεονεκτήματα του phpMyAdmin**

- Υψηλή απόδοση
- Διασυνδέσεις με πολλά συστήματα βάσεων δεδομένων
- Ενσωματωμένες βιβλιοθήκες
- Χαμηλό κόστος
- Ευκολία μάθησης και χρήσης
- Μεταφερσιμότητα
- Διαθεσιμότητα

## **2.6 Η γλώσσα προγραμματισμού PHP**

### **2.6.1 Εισαγωγή**

Η PHP είναι μία ευρέως χρησιμοποιούμενη, ανοιχτού κώδικα, scripting γλώσσα προγραμματισμού η οποία είναι ειδικά κατάλληλη για ανάπτυξη εφαρμογών για το Διαδίκτυο και μπορεί να ενσωματωθεί και στην γλώσσα σήμανσης HTML. Είναι ιδανική γλώσσα προγραμματισμού για την δημιουργία ιστοτόπων με δυναμικό περιεχόμενο, καθώς μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό server του Παγκόσμιου Ιστού (όπως έχουμε αναφέρει και παραπάνω ένας τέτοιος είναι ο Apache), ώστε το τελικό περιεχόμενο να παραχθεί σε πραγματικό χρόνο.

### **2.6.2 Ιστορία της PHP**

Η PHP γλώσσα δημιουργήθηκε από τον Rasmus Lerdorf το 1994 και κυκλοφόρησε δημοσίως τον Ιούνιο του 1995. Βασίστηκε στην γλώσσα προγραμματισμού C και δημιούργησε το πρώτο του script με όνομα php.cgi για προσωπική του χρήση. Αφορμή ήταν ότι ήθελε να κρατάει μία λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Το script αυτό πήρε διαστάσεις όταν άρχισε να το δίνει στους φίλους του, οι οποίοι αργότερα του ζητούσαν να το εμπλουτίσει με περισσότερες δυνατότητες. Η γλώσσα τότε είχε το όνομα PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997, δηλαδή μετά από δύο χρόνια, εισήχθη ως PHP / F1 2.0 και έναν χρόνο αργότερα δύο άλλοι



προγραμματιστές, ο Zeev Suraski και ο Andi Gutmans, ξανάγραψαν τη βάση της αρχικής έκδοσης και ξεκίνησαν την PHP 3.40.

Δύο δεκαετίες μετά την έναρξή της, η PHP σημείωσε μια πρωτοφανή ανάπτυξη και εξακολουθεί να είναι ισχυρή. Σήμερα ελέγχει πάνω από το 80% όλων των ιστοσελίδων του πλανήτη. Αυτό περιλαμβάνει μεγάλες εταιρείες μεταξύ άλλων το Facebook, το Wikipedia και το WordPress.

### 2.6.3 Χρήση PHP και HTML

Η PHP και η HTML αλληλοεπιδρούν πολύ μεταξύ τους, διότι η PHP μπορεί να δημιουργήσει ένα HTML αρχείο και η HTML μπορεί να μεταφέρει πληροφορίες στην PHP. Για παράδειγμα, όταν δημιουργείτε μια

πολύπλοκη σελίδα, σε κάποιο σημείο ερχόμαστε αντιμέτωποι με την ανάγκη συνδυασμού της PHP γλώσσας με την HTML για να έχουμε τα αποτελέσματα που χρειαζόμαστε. Σε πρώτη φάση αυτό μπορεί να φαίνεται πολύπλοκο, αφού η PHP και η HTML είναι δύο τελείως ξεχωριστές γλώσσες, αλλά αυτό δεν συμβαίνει. Η PHP έχει σχεδιαστεί έτσι ώστε να μπορεί να αλληλοεπιδρά με την HTML και τα PHP scripts να μπορούν να συμπεριληφθούν σε μια σελίδα HTML χωρίς κανένα πρόβλημα.

Σε μια σελίδα HTML, ο κώδικας PHP περικλείεται μέσα σε ειδικές ετικέτες PHP. Όταν ο επισκέπτης ανοίξει την σελίδα, ο server επεξεργάζεται τον κώδικα PHP και στην συνέχεια αποστέλλει την έξοδο, και όχι τον ίδιο τον κώδικα PHP) στον φυλλομετρητή του χρήστη. Στην πραγματικότητα είναι πολύ απλό να ενσωματώσουμε HTML και PHP. Ένα PHP script μπορεί να αντιμετωπιστεί ως μία HTML σελίδα, με κομμάτια PHP εισαγόμενα από εδώ και από εκεί στον κώδικα. Οτιδήποτε σε ένα PHP script που δεν συμπεριλαμβάνεται σε `<?php..... ?>` ετικέτες αγνοούνται από τον PHP μεταγλωττιστή (compiler) και μεταφέρεται απευθείας στον φυλλομετρητή.

## 2.6.4 Πλεονεκτήματα σε σχέση με την HTML

Μερικά από τα πλεονεκτήματα χρήσης της PHP γλώσσας σε σχέση με την HTML είναι τα εξής:

- Η HTML είναι στατική γλώσσα που σημαίνει ότι τα δεδομένα της δεν μπορούν να αλλάξουν ενώ η PHP, αντίθετα, είναι δυναμική γλώσσα. Επομένως τα δεδομένα του ιστοτόπου μπορούν να ανανεώνονται και να αλλάζουν με εντολή του χρήστη.
- Η HTML δεν είναι ασφαλής για ιστοτόπους που χρειάζονται ασφάλεια στα δεδομένα τους. Πολλοί ιστότοποι όπως είναι και τα e-shop, χρησιμοποιούν προσωπικά δεδομένα και τραπεζικά δεδομένα που χρειάζονται ασφάλεια.
- Η PHP γλώσσα μπορεί να συνδεθεί με μία βάση δεδομένων και να αντλήσει από εκεί τις πληροφορίες της.

## 2.6.5 Γενικά πλεονεκτήματα της PHP γλώσσας

Μερικά από τα πλεονεκτήματα της PHP γλώσσας είναι τα εξής:

- Ανοιχτού κώδικα: Αναπτύσσεται και συντηρείται από μια μεγάλη ομάδα προγραμματιστών της PHP, κάτι που βοηθά στη δημιουργία μιας μεγάλης κοινότητας υποστήριξης και σε μια πλούσια βιβλιοθήκη επέκτασης.
- Ταχύτητα: Είναι σχετικά γρήγορη αφού χρησιμοποιεί πολλούς πόρους συστήματος.
- Εύκολη στη χρήση: Χρησιμοποιεί την γλώσσα προγραμματισμού C σαν σύνταξη, οπότε για εκείνους που είναι εξοικειωμένοι με την C, είναι πολύ εύκολο να δημιουργήσουν εφαρμογές και ιστοσελίδες διαδικτύου.
- Σταθερότητα: Δεδομένου ότι διατηρείται από πολλούς προγραμματιστές, έτσι ώστε όταν εντοπιστούν σφάλματα, μπορούν να διορθωθούν γρήγορα.
- Ισχυρή υποστήριξη βιβλιοθήκης: Μπορείτε εύκολα να βρείτε λειτουργικές μονάδες που χρειάζεστε όπως PDF, Graph κ.α.
- Ενσωματωμένες μονάδες σύνδεσης βάσης δεδομένων: Μπορείτε να συνδεθείτε εύκολα στη βάση δεδομένων χρησιμοποιώντας την PHP, δεδομένου ότι πολλοί ιστότοποι οδηγούν δεδομένα/περιεχόμενο, επομένως θα χρησιμοποιήσουμε συχνά βάση δεδομένων, αυτό θα μειώσει σε μεγάλο βαθμό τον χρόνο ανάπτυξης των εφαρμογών ιστού.
- Μπορεί να εκτελεστεί σε πολλές πλατφόρμες, συμπεριλαμβανομένων των Windows,

Linux και Mac, είναι εύκολο για τους χρήστες να βρουν παρόχους υπηρεσιών φιλοξενίας.

### 2.6.6 Γενικά μειονεκτήματα της PHP γλώσσας

- Ασφάλεια: Δεδομένου ότι είναι ανοιχτό, έτσι ώστε όλοι οι άνθρωποι να μπορούν να δουν τον πηγαίο κώδικα, αν υπάρχουν σφάλματα στον πηγαίο κώδικα, μπορούν να χρησιμοποιηθούν από τους ανθρώπους για να εξερευνήσουν την αδυναμία της PHP.
- Δεν είναι κατάλληλη για μεγάλες εφαρμογές: Είναι δύσκολη στην συντήρηση λόγω του ότι είναι πολυτμηματική γλώσσα
- Αδύνατος τύπος: Η αναγκαστική μετατροπή μπορεί να εκπλήξει απρόσεκτους προγραμματιστές και να οδηγήσει σε απροσδόκητα σφάλματα. Για παράδειγμα, οι συμβολοσειρές "1000" και "1e3" συγκρίνονται ίσες επειδή μετατρέπονται σε αριθμούς κινητής υποδιαστολής.

## 2.7 Η γλώσσα προγραμματισμού JavaScript

### 2.7.1 Γενικά στοιχεία της Javascript

Η JavaScript είναι μια γλώσσα προγραμματισμού ηλεκτρονικών υπολογιστών, η οποία έχει σαν σκοπό την παραγωγή δυναμικού περιεχομένου και την εκτέλεση κώδικα στην πλευρά του πελάτη (client-side) στις ιστοσελίδες. Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης ECMA ονομάζεται ECMAScript.

Η γλώσσα προγραμματισμού JavaScript δημιουργήθηκε αρχικά από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha. Αργότερα, Mocha μετονομάστηκε σε LiveScript, και τελικά σε JavaScript, κυρίως επειδή η ανάπτυξή της επηρεάστηκε περισσότερο από τη γλώσσα προγραμματισμού Java.

LiveScript ήταν το επίσημο όνομα της γλώσσας όταν για πρώτη φορά κυκλοφόρησε στην αγορά σε βήτα (beta) εκδόσεις με το πρόγραμμα περιήγησης στο Web, Netscape Navigator εκδοχή 2.0 τον Σεπτέμβριο του 1995. Αργότερα μετονομάστηκε σε JavaScript σε μια κοινή ανακοίνωση με την εταιρεία Sun Microsystems στις 4 Δεκεμβρίου, 1995, όταν επεκτάθηκε στην έκδοση του προγράμματος περιήγησης στο Web, Netscape εκδοχή 2.0B3.

Η JavaScript απέκτησε μεγάλη επιτυχία ως γλώσσα στην πλευρά του πελάτη (client-side) για εκτέλεση κώδικα σε ιστοσελίδες, και περιλήφθηκε σε διάφορα προγράμματα

περιήγησης στο Web. Κατά συνέπεια, η εταιρεία Microsoft ονόμασε την εφαρμογή της σε JScript για να αποφύγει δύσκολα θέματα εμπορικών σημάτων. JScript πρόσθεσε νέους μεθόδους για να διορθώσει τα Y2K-προβλήματα στην JavaScript, οι οποίοι βασίστηκαν στην java.util.Date τάξη της Java. JScript περιλήφθηκε στο πρόγραμμα Internet Explorer εκδοχή 3.0, το οποίο κυκλοφόρησε τον Αύγουστο του 1996.

Τον Νοέμβριο του 1996, η Netscape ανακοίνωσε ότι είχε υποβάλει τη γλώσσα JavaScript στο Ecma International (μια οργάνωση της τυποποίησης των γλωσσών προγραμματισμού) για εξέταση ως βιομηχανικό πρότυπο, και στη συνέχεια το έργο είχε ως αποτέλεσμα την τυποποιημένη μορφή που ονομάζεται ECMAScript.

Η JavaScript έχει γίνει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στον Παγκόσμιο Ιστό (Web). Αρχικά, όμως, πολλοί επαγγελματίες προγραμματιστές υποτίμησαν τη γλώσσα διότι το κοινό της ήταν ερασιτέχνες συγγραφείς ιστοσελίδων και όχι επαγγελματίες προγραμματιστές (και μεταξύ άλλων λόγων). Με τη χρήση της τεχνολογίας Ajax, η JavaScript γλώσσα επέστρεψε στο προσκήνιο και έφερε πιο επαγγελματική προσοχή προγραμματισμού. Το αποτέλεσμα ήταν ένα καινοτόμο αντίκτυπο στην εξάπλωση των πλαισίων και των βιβλιοθηκών, τη βελτίωση προγραμματισμού με JavaScript, καθώς και αυξημένη χρήση της JavaScript έξω από τα προγράμματα περιήγησης στο Web.

Τον Ιανουάριο του 2009, το έργο CommonJS ιδρύθηκε με στόχο τον καθορισμό ενός κοινού προτύπου βιβλιοθήκης κυρίως για την ανάπτυξη της JavaScript έξω από το πρόγραμμα περιήγησης και μέσα σε άλλες τεχνολογίες (π.χ. Server-side).

### **2.7.2 Μοντέλο εκτέλεσης**

Η αρχική έκδοση της Javascript βασίστηκε στη σύνταξη στη γλώσσα προγραμματισμού C, αν και έχει εξελιχθεί, ενσωματώνοντας πια χαρακτηριστικά από νεότερες γλώσσες. Αρχικά χρησιμοποιήθηκε για προγραμματισμό από την πλευρά του πελάτη (client), που ήταν ο φυλλομετρητής (browser) του χρήστη, και χαρακτηρίστηκε σαν client-side γλώσσα προγραμματισμού. Αυτό σημαίνει ότι η επεξεργασία του κώδικα Javascript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή, αλλά στο πρόγραμμα περιήγησης των επισκεπτών, ενώ μπορεί να ενσωματωθεί σε στατικές σελίδες HTML. Αντίθετα, άλλες γλώσσες όπως η PHP εκτελούνται στο διακομιστή (server-side γλώσσες προγραμματισμού).

Παρά την ευρεία χρήση της Javascript για συγγραφή προγραμμάτων σε περιβάλλον φυλλομετρητή, αξίζει να σημειωθεί ότι από την αρχή χρησιμοποιήθηκε και για τη συγγραφή κώδικα από την πλευρά του διακομιστή, από την ίδια τη Netscape στο προϊόν LiveWire, με μικρή επιτυχία. Η χρήση της Javascript στο διακομιστή εμφανίζεται πάλι σήμερα, με τη διάδοση του Node.js, ενός μοντέλου προγραμματισμού βασισμένο στα γεγονότα (events).

### 2.7.3 Javascript και Java

Η Javascript δεν θα πρέπει να συγχέεται με τη Java, που είναι διαφορετική γλώσσα προγραμματισμού και με διαφορετικές εφαρμογές. Η χρήση της λέξης "Java" στο όνομα της γλώσσας έχει περισσότερη σχέση με το προφίλ του προϊόντος που έπρεπε να έχει και λιγότερο με κάποια πιθανή συμβατότητα ή άλλη στενή σχέση με τη Java.

Ρόλο σε αυτήν τη σύγχυση έπαιξε και ότι η Java και η Javascript έχουν δεχτεί σημαντικές επιρροές από τη γλώσσα C, ειδικά στο συντακτικό, ενώ είναι και οι δύο αντικειμενοστραφείς γλώσσες. Τονίζεται ότι ο σωστός τρόπος γραφής της είναι "Javascript" και όχι 'Java script' σαν δύο λέξεις, όπως λανθασμένα γράφεται ορισμένες φορές.

### 2.7.4 Βιβλιοθήκες της Javascript

Μία βιβλιοθήκη Javascript είναι μία βιβλιοθήκη προ-γραμμένου κώδικα JavaScript που επιτρέπει την ευκολότερη ανάπτυξη εφαρμογών που βασίζονται σε Javascript. Κάποια παραδείγματα τέτοιων βιβλιοθηκών είναι η GUI , DOM, Pure Javascript/ Ajax, Web-application, Template system και άλλες.

#### Vue.js

Το Vue.js είναι ένα front-end framework της JavaScript που δημιουργήθηκε για να οργανώνει και να απλοποιεί την ανάπτυξη ιστού. Το έργο επικεντρώνεται στο να γίνουν πιο προσιτές οι ιδέες για την ανάπτυξη του γραφικού περιβάλλοντος στο διαδίκτυο. Μπορεί να χρησιμοποιηθεί σε υπάρχοντα έργα για να προσθέσει απλή διαδραστικότητα ή για να αντικαταστήσει εξ ολοκλήρου το jQuery και άλλες γλώσσες.

Η βασική λειτουργία μέσα στο σύστημα που εκπληρώνεται με την χρήση του Vue.js είναι η ενσωμάτωση της φόρμας Stripe που είναι υπεύθυνη για τις συναλλαγές. Διευκολύνει τον προγραμματισμό χωρίζοντας την φόρμα σε άλλο αρχείο Επίσης μεταφέρονται δυναμικά

δεδομένα εντός της φόρμας όπως είναι το email του χρήστη ώστε να μην χρειάζεται να το προσθέτει κάθε φορά.

## **jQuery**

Η jQuery είναι μία πολύ-πλατφορμική βιβλιοθήκη JavaScript σχεδιασμένη να απλοποιεί την υλοποίηση σεναρίου HTML από την πλευρά του πελάτη. Η jQuery είναι η πιο δημοφιλής βιβλιοθήκη JavaScript σε χρήση σήμερα, με εγκατάσταση στο 65% των 10 εκατομμυρίων κορυφαίων ιστοσελίδων στον Παγκόσμιο Ιστό με την περισσότερη κίνηση. Η jQuery είναι ένα δωρεάν, ανοιχτού κώδικα λογισμικό με άδεια από την MIT License.

Η σύνταξη της jQuery είναι σχεδιασμένη να κάνει ευκολότερη την πλοήγηση ενός εγγράφου, την επιλογή στοιχείων DOM, τη δημιουργία κινουμένων σχεδίων, τον χειρισμό γεγονότων και τον σχεδιασμό εφαρμογών Ajax. Η jQuery παρέχει επίσης στους προγραμματιστές δυνατότητες για να δημιουργήσουν πρόσθετα εξαρτήματα λογισμικού στην κορυφή της βιβλιοθήκης JavaScript. Αυτό επιτρέπει στους προγραμματιστές να δημιουργήσουν αφαιρέσεις για χαμηλού επιπέδου αλληλεπίδραση και κινούμενα σχέδια, προηγμένα εφέ και υψηλού επιπέδου θεματικά widgets. Η αρθρωτή προσέγγιση της βιβλιοθήκης jQuery επιτρέπει τη δημιουργία ισχυρών δυναμικών ιστοσελίδων και διαδικτυακών εφαρμογών. Η Microsoft και η Nokia δεσμεύουν την jQuery στις πλατφόρμες τους.

## **Επισκόπηση jQuery**

Η jQuery, στον πυρήνα της, είναι μία βιβλιοθήκη χειρισμού DOM (Document Object Model). Το DOM είναι μία δομική αναπαράσταση δέντρου όλων των στοιχείων μίας σελίδας Διαδικτύου, και η jQuery απλοποιεί τη σύνταξη για εύρεση, επιλογή και χειρισμό αυτών των στοιχείων DOM. Για παράδειγμα, η jQuery μπορεί να χρησιμοποιηθεί για εύρεση ενός στοιχείου μέσα στο αρχείο με μία συγκεκριμένη ιδιότητα (π.χ. όλα τα στοιχεία με μία h1 ετικέτα), για αλλαγή ενός ή περισσότερων από τα χαρακτηριστικά του (π.χ. χρώμα, ορατότητα), ή για να το καταστήσει ικανό να ανταποκριθεί σε ένα γεγονός (π.χ. ένα κλικ του ποντικιού).

Η jQuery παρέχει επίσης ένα παράδειγμα για χειρισμό γεγονότων που ξεπερνούν τη βασική επιλογή και χειρισμό DOM. Η ανάθεση γεγονότων και ο ορισμός της λειτουργίας callback γίνονται σε ένα μοναδικό βήμα σε μία μοναδική τοποθεσία στον κώδικα. Η jQuery έχει επίσης ως στόχο να ενσωματώσει άλλες λειτουργίες υψηλής χρήσης της JavaScript (π.χ. το

fade in και το fade out κατά το κρύψιμο στοιχείων, κινούμενα σχέδια μέσω της χρήσης ιδιοτήτων CSS).

Τα πλεονεκτήματα της χρήσης jQuery είναι:

- Ενθαρρύνει τον διαχωρισμό μεταξύ της JavaScript και της HTML: Η βιβλιοθήκη jQuery παρέχει απλό συντακτικό για την προσθήκη χειριστών των γεγονότων στο DOM χρησιμοποιώντας την JavaScript, παρά της προσθήκης HTML χαρακτηριστικών των γεγονότων στις JavaScript λειτουργίες. Επομένως, ενθαρρύνονται οι κατασκευαστές να διαχωρίσουν πλήρως τον κώδικα JavaScript από την HTML.
- Συντομία και σαφήνεια: Η jQuery προωθεί τη συντομία και τη σαφήνεια με χαρακτηριστικά όπως είναι οι αλυσιδωτές λειτουργίες και λειτουργία στενογραφίας ονομάτων.
- Εξαλείφει τις ασυμβατότητες μεταξύ των προγραμμάτων περιήγησης: Οι μηχανές JavaScript διαφορετικών προγραμμάτων περιήγησης διαφέρουν ελαφρώς, έτσι ώστε ο κώδικας JavaScript που λειτουργεί για ένα πρόγραμμα περιήγησης ίσως να μην λειτουργεί για ένα άλλο. Όπως και άλλα εργαλεία της JavaScript, η jQuery χειρίζεται όλες αυτές τις αντιφάσεις μεταξύ των προγραμμάτων περιήγησης και παρέχει μία συνεπή διεπαφή που λειτουργεί σε διαφορετικά προγράμματα περιήγησης.
- Επεκτασιμότητα: Νέα γεγονότα, στοιχεία και μέθοδοι μπορούν εύκολα να προστεθούν και μετά να ξαναχρησιμοποιηθούν ως ένα πρόσθετο εξάρτημα λογισμικού.

### **Χαρακτηριστικά jQuery**

Η jQuery περιλαμβάνει τα εξής χαρακτηριστικά:

- DOM element επιλογές χρησιμοποιώντας την ανοιχτού κώδικα μηχανή επιλογής πολλαπλών φυλλομετρητών Sizzle.
- DOM διάσχιση και τροποποίηση
- χειρισμός DOM βασισμένος σε CSS επιλογείς που χρησιμοποιεί τα id και class σαν κριτήρια για να κατασκευάσει επιλογείς.
- Events
- Εφέ και κινητά στοιχεία
- AJAX
- Επεκτασιμότητα μέσω plug-ins
- Εργαλεία όπως πληροφορίες user-agent, ανίχνευση χαρακτηριστικών.
- Μεθόδους συμβατότητας που είναι εγγενώς διαθέσιμα σε σύγχρονα προ-γράμματα περιήγησης.

- Υποστήριξη πολλαπλών φυλλομετρητών.

## **Chart.js**

Η διεπαφή σχεδιασμού Chart.js αναπτύχθηκε αποκλειστικά από την κοινότητα το 2013 και πρόκειται για μια βιβλιοθήκη οπτικοποίησης γραφημάτων 2 διαστάσεων πάνω σε HTML canvas. Η βιβλιοθήκη είναι εύκολη στην χρήση και χρησιμοποιεί αντικείμενα τύπου Chart προκειμένου να εισαχθούν όλα τα απαραίτητα δεδομένα στο γράφημα. Υποστηρίζονται 8 τύποι γραφημάτων: line, bar, radar, doughnut & pie, polar area, bubble, scatter, area και μεικτοί τύποι.

## **2.8 Bootstrap**

Το Bootstrap είναι μία ελεύθερη και ανοιχτού κώδικα συλλογή από εργαλεία για τη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Περιέχει HTML και CSS πρότυπα σχεδίασης για τυπογραφία, φόρμες, κουμπιά πλοήγησης και άλλα στοιχεία διεπαφής, καθώς επίσης και προαιρετικές επεκτάσεις JavaScript. Στόχο έχει να διευκολύνει την ανάπτυξη δυναμικών ιστοσελίδων και διαδικτυακών εφαρμογών. Το Bootstrap είναι ένα front-end πλαίσιο, δηλαδή, μία διεπαφή για τον χρήστη, σε αντίθεση με τον κώδικα από την πλευρά του διακομιστή ο οποίος βρίσκεται στο back-end του διακομιστή. Το Bootstrap είναι το πιο δημοφιλές πρόγραμμα του GitHub και έχει χρησιμοποιηθεί από τη NASA, μεταξύ άλλων.

### **2.8.1 Προέλευση**

Το Bootstrap, που αρχικά ονομάστηκε Twitter Blueprint, αναπτύχθηκε από τον Mark Otto και τον Jacob Thornton στο Twitter ως ένα πλαίσιο για την προώθηση της συνοχής μεταξύ των εσωτερικών εργαλείων. Πριν το Bootstrap, ποικίλες βιβλιοθήκες χρησιμοποιήθηκαν για την ανάπτυξη της διεπαφής, κάτι το οποίο οδήγησε σε ασυνέπειες και υψηλό φόρτο συντήρησης.

Η πρώτη εγκατάσταση υπό πραγματικές συνθήκες συνέβη κατά τη διάρκεια της πρώτης Hackweek του Twitter. Μετονομάστηκε από Twitter Blueprint σε Bootstrap, και κυκλοφόρησε ως ένα λογισμικό ανοιχτού κώδικα στις 19 Αυγούστου του 2011. Έχει συνεχίσει να διατηρείται από τον Mark Otto, τον Jacob Thornton και μια μικρή ομάδα κορυφαίων προγραμματιστών, καθώς επίσης και μία μεγάλη κοινότητα από συνεισφέροντες.



Στις 31 Ιανουαρίου του 2012, ανακοινώθηκε το Bootstrap 2. Αυτή η έκδοση πρόσθεσε το διατακτικό πλέγμα δώδεκα στηλών και ανταποκρίσιμα σχεδιαστικά στοιχεία, καθώς επίσης και αλλαγές σε πολλά από τα υπάρχοντα στοιχεία. Η έκδοση Bootstrap 3 ανακοινώθηκε στις 19 Αυγούστου του 2013, μετακινούμενη προς μία φορητή πρώτη προσέγγιση και χρησιμοποιώντας έναν επίπεδο σχεδιασμό.

Στις 29 Οκτωβρίου του 2014, ο Mark Otto ανακοίνωσε ότι κατασκευάζεται το Bootstrap 4. Η πρώτη alpha έκδοση του Bootstrap 4 δρομολογήθηκε στις 19 Αυγούστου του 2015.

## **2.8.2 Χαρακτηριστικά**

Το Bootstrap είναι συμβατό με τις τελευταίες εκδόσεις των προγραμμάτων περιήγησης του Google Chrome, Firefox, Internet Explorer, Opera, και Safari, αν και μερικά από αυτά τα προγράμματα περιήγησης δεν υποστηρίζονται σε όλες τις πλατφόρμες.

Από την κυκλοφορία της έκδοσης 2.0, υποστηρίζει επίσης τον ανταποκρίσιμο διαδικτυακό σχεδιασμό. Αυτό σημαίνει ότι η διάταξη των ιστοσελίδων προσαρμόζεται δυναμικά, λαμβάνοντας υπόψη τα χαρακτηριστικά της συσκευής που χρησιμοποιείται (επιτραπέζιος υπολογιστής, tablet, κινητό τηλέφωνο). Ξεκινώντας με την έκδοση 3.0, το Bootstrap υιοθέτησε μία πρώτη φιλοσοφία σχεδιασμού για κινητά, δίνοντας έμφαση στον ανταποκρίσιμο σχεδιασμό από προεπιλογή. Η κυκλοφορία της έκδοσης 4.0 alpha πρόσθεσε την υποστήριξη Sass και Flexbox.

Το Bootstrap είναι ανοιχτού κώδικα και διαθέσιμο στο Github. Οι προγραμματιστές ενθαρρύνονται να συμμετέχουν στο project και να κάνουν την δική τους συνεισφορά στη πλατφόρμα.

## **2.8.3 Δομή και Λειτουργία**

Το Bootstrap είναι σπονδυλωτό και αποτελείται ουσιαστικά από μία σειρά στυλ stylesheets φύλλων LESS που εφαρμόζουν τα ποικίλα στοιχεία της εργαλειοθήκης. Ένα στυλ φύλλων που ονομάζεται bootstrap.less περιέχει αυτά τα συστατικά stylesheets. Οι προγραμματιστές μπορούν να προσαρμόσουν το ίδιο το αρχείο Bootstrap, επιλέγοντας τα συστατικά που επιθυμούν να χρησιμοποιήσουν στο project τους.

Οι προσαρμογές είναι δυνατές μέχρι έναν βαθμό, μέσω ενός κεντρικού στυλ διαμόρφωσης των φύλλων. Βαθύτερες αλλαγές είναι εφικτές μέσω των δηλώσεων LESS. Η χρήση

γλώσσας στυλ φύλλων LESS επιτρέπει τη χρήση για μεταβλητές, λειτουργίες και φορείς, ένθετους επιλογείς, γνωστά και ως μείγματα mixins.

Από την έκδοση 2.0, η διαμόρφωση του Bootstrap έχει επίσης μια ειδική επιλογή «Προσαρμογή» στην τεκμηρίωση. Επιπλέον, ο προγραμματιστής επιλέγει μία φόρμα, τα επιθυμητά συστατικά και προσαρμόζει, εάν είναι απαραίτητο, τις τιμές ποικίλων επιλογών σύμφωνα με τις ανάγκες τους. Το πακέτο που δημιουργείται στη συνέχεια περιέχει ήδη το προκατασκευασμένο CSS στυλ φύλλων.

### ***Σύστημα πλέγματος(Grid system) και ο ανταποκρίσιμος σχεδιασμός(responsive design)***

Προσφέρονται με προδιαγραφές ένα διατακτικό πλέγμα των 1170 pixel σε εύρος. Εναλλακτικά, ο προγραμματιστής μπορεί να χρησιμοποιήσει μία μεταβλητού πλάτους διάταξη. Και στις δύο περιπτώσεις, η εργαλειοθήκη έχει τέσσερις παραλλαγές για την χρήση διαφορετικών αναλύσεων και τύπους συσκευών κινητά τηλέφωνα, κατακόρυφη και οριζόντια διάταξη, ταμπλέτες και υπολογιστές με χαμηλή και υψηλή ανάλυση. Κάθε παραλλαγή αλλάζει το πλάτος των στηλών.

### ***Η κατανόηση του CSS στυλ***

Το Bootstrap παρέχει ένα σύνολο στυλ φύλλων που παρέχει βασικούς ορισμούς στυλ για όλα τα βασικά HTML στοιχεία. Αυτά παρέχουν μία ενιαία, σύγχρονη εμφάνιση για μορφοποίηση κειμένου, πίνακες και στοιχεία μίας φόρμας.

### ***Επαναχρησιμοποιήσιμα στοιχεία***

Εκτός από τα βασικά HTML στοιχεία, το Bootstrap περιέχει άλλα στοιχεία περιβάλλοντος που χρησιμοποιούνται συχνά. Αυτά περιλαμβάνουν κουμπιά με προηγμένα χαρακτηριστικά (π.χ. ομαδοποίηση κουμπιών ή κουμπιά με επιλογή drop-down, λίστες πλοήγησης, οριζόντιες και κάθετες καρτέλες, πλοήγηση, πλοήγηση breadcrumb, σελιδοποίηση, κ.τ.λ.), ετικέτες, προηγμένες τυπογραφικές δυνατότητες, εικονίδια, προειδοποιητικά μηνύματα και μία γραμμή προόδου. Τα συστατικά εφαρμόζονται ως κλάσεις CSS, οι οποίες πρέπει να εφαρμοστούν σε συγκεκριμένα στοιχεία HTML μιας σελίδας.

### ***JavaScript στοιχεία***

Το Bootstrap έρχεται με πολλά συστατικά JavaScript σε μία μορφή jQuery plugins. Αυτά παρέχουν επιπλέον στοιχεία διεπαφή χρήστη, όπως είναι τα παράθυρα διαλόγου, οι επεξηγήσεις (tooltips) και τα καρουσέλ (carousels). Επίσης, μπορούν να επεκτείνουν τη

λειτουργικότητα ορισμένων υφιστάμενων στοιχείων της διασύνδεσης, συμπεριλαμβανομένης για παράδειγμα μιας λειτουργίας αυτόματης συμπλήρωσης πεδίων εισαγωγής. Στην έκδοση 2.0, υποστηρίζονται τα ακόλουθα plugins: Modal, Dropdown, Scrollspy, Tab, Tooltip, Popover, Alert, Button, Collapse, Carousel και Typeahead.

#### 2.8.4 Χρήση

Το Bootstrap, για να χρησιμοποιηθεί σε μία σελίδα HTML, ο προγραμματιστής θα πρέπει να κάνει λήψη του στυλ CSS Bootstrap το οποίο περιλαμβάνει μία σύνδεση στο αρχείο HTML. Αν ο προγραμματιστής θέλει να χρησιμοποιήσει τα στοιχεία Javascript, θα πρέπει να αναφέρονται μαζί με τη βιβλιοθήκη jQuery στο HTML έγγραφο.

### 2.9 Software Frameworks

Στον υπολογιστικό προγραμματισμό, ένα software framework είναι μία αφηρημένη έννοια κατά την οποία ένα λογισμικό που παρέχει γενική λειτουργία μπορεί να αλλαχθεί επιλεκτικά από έναν επιπλέον χρήστη ή ένα γραπτό κώδικα, παρέχοντας έτσι μία εφαρμογή ή ένα ειδικό λογισμικό. Ένα πλαίσιο λογισμικού είναι ένα παγκόσμιο, επαναχρησιμοποιούμενο λογισμικό περιβάλλον που παρέχει συγκεκριμένη λειτουργικότητα, ως μέρος μιας μεγαλύτερης πλατφόρμας, για τη διευκόλυνση ανάπτυξης λογισμικών εφαρμογών, προϊόντων και λύσεων. Τα software frameworks μπορεί να περιλαμβάνουν υποστηρικτικά προγράμματα, μεταγλωττιστές, βιβλιοθήκες κώδικα, σετ εργαλείων και διεπαφή προγραμματισμού εφαρμογών (APIs) τα οποία συγκεντρώνουν όλα τα διαφορετικά συστατικά προκειμένου να καταστεί δυνατή η ανάπτυξη ενός project ή μιας λύσης.

Τα frameworks περιέχουν βασικά διακριτικά γνωρίσματα που τα διαχωρίζουν από τις κανονικές βιβλιοθήκες:

- αντιστροφή του ελέγχου: Σε ένα framework, σε αντίθεση με τις βιβλιοθήκες ή τις κανονικές εφαρμογές χρηστών, η συνολική ροή ελέγχου του προγράμματος δεν υπαγορεύεται από τον καλούντα, αλλά από το πλαίσιο.
- προεπιλεγμένη συμπεριφορά: Ένα framework έχει μία προεπιλεγμένη συμπεριφορά. Αυτή η προεπιλεγμένη συμπεριφορά πρέπει να είναι κάποια χρήσιμη συμπεριφορά και όχι μία σειρά από no-ops.
- επεκτασιμότητα: Ένα framework μπορεί να επεκταθεί συνήθως από το χρήστη μέσω επιλεκτικής παράκαμψης ή εξειδικευμένου κώδικα από το χρήστη, για την παροχή

συγκεκριμένης λειτουργικότητας.

- μη-τροποποιήσιμος κώδικας πλαισίου: Ο framework κώδικας ,σε γενικές γραμμές, δεν πρέπει να τροποποιείται, ενώ γίνονται αποδεκτές εφαρμοσμένες επεκτάσεις του χρήστη. Με άλλα λόγια, οι χρήστες μπορούν να επεκτείνουν το framework, αλλά δεν θα πρέπει να τροποποιούν τον κώδικά του.

### 2.9.1 Φιλοσοφία

Οι σχεδιαστές των software frameworks έχουν στόχο να διευκολύνουν την ανάπτυξη λογισμικού, επιτρέποντας στους σχεδιαστές και τους προγραμματιστές να αφιερώσουν το χρόνο τους στην ικανοποίηση των αναγκών του λογισμικού παρά να ασχοληθούν με τις πιο στάνταρ, χαμηλού επιπέδου λεπτομέρειες της παροχής ενός λειτουργικού συστήματος, μειώνοντας έτσι το συνολικό χρόνο ανάπτυξης. Για παράδειγμα, μία ομάδα χρησιμοποιώντας ένα δικτυακής εφαρμογής framework για την ανάπτυξη μίας τραπεζικής ιστοσελίδας, μπορεί να επικεντρωθεί στη δημιουργία ενός ειδικού τραπεζικού κώδικα παρά στους μηχανισμούς χειρισμού των αιτημάτων και της κρατικής διαχείρισης.

Τα frameworks συχνά προσθέτουν στο μέγεθος των προγραμμάτων ένα φαινόμενο που ονομάζεται “φουσκωμένος κώδικας” (“code bloat”). Λόγω των αναγκών των εφαρμογών που καθοδηγούνται από

πελατειακές απαιτήσεις, τόσο τα ανταγωνιστικά όσο και τα συμπληρωματικά frameworks καταλήγουν μερικές φορές σε ένα προϊόν. Επιπλέον, λόγω της πολυπλοκότητάς των APIs τους, η προβλεπόμενη μείωση στο συνολικό χρόνο ανάπτυξης μπορεί να μην επιτευχθεί, εξαιτίας της ανάγκης να δαπανηθεί επιπλέον χρόνος για την εκμάθηση της χρήσης του πλαισίου: αυτή η κριτική σαφώς ισχύει όταν ένα ειδικό ή καινούριο framework αντιμετωπίζεται για πρώτη φορά από το αναπτυξιακό προσωπικό. Εάν ένα τέτοιο framework δεν χρησιμοποιηθεί σε επόμενες ανατεθειμένες εργασίες, ο χρόνος που επενδύθηκε για την εκμάθηση του framework μπορεί να κοστίζει περισσότερο από τον επιδιωκόμενο γραπτό κώδικα, γνωστό στο προσωπικό του project· πολλοί προγραμματιστές κρατάνε αντίγραφα χρήσιμων στερεοτύπων για συνηθισμένες ανάγκες.

Όπως φαίνεται σε βάθος χρόνου, τα πιο αποτελεσματικά frameworks έχουν αποδειχτεί ότι είναι εκείνα που εξελίσσονται από την εκ νέου συντέλεση του κοινού κώδικα της επιχείρησης, αντί τη χρήση ενός γενικού, συγκεκριμένο σε μέγεθος framework που ταιριάζει με όλα και αναπτύχθηκε από τρίτους για γενικούς σκοπούς. Ένα παράδειγμα αυτού θα ήταν

το πώς η διεπαφή χρήστη σε ένα τέτοιο πακέτο εφαρμογών, όπως είναι το office suite, αναπτύσσεται και έχει συνηθισμένη εμφάνιση, αίσθηση, χαρακτηριστικά και μεθόδους διαμοιρασμού δεδομένων, όπως κάποτε οι διάσπαρτες συνοδευτικές εφαρμογές αναπτύσσονται ενωμένες σε μία σύνθεση που είναι μικρότερη· η καινούρια-εξελιγμένη σύνθεση μπορεί να είναι ένα προϊόν που μοιράζεται αναπόσπαστες χρηστικές βιβλιοθήκες και διεπαφές χρήστη.

Αυτή είναι και η αιτία η οποία φέρνει στην επιφάνεια ένα σημαντικό θέμα με τα frameworks. Η δημιουργία ενός framework που είναι κομψό, σε σχέση με ένα που απλά λύνει ένα πρόβλημα, εξακολουθεί να θεωρείται τέχνη και όχι επιστήμη. Η «λογισμική κομψότητα» υπονοεί σαφήνεια, περιεκτικότητα, και λίγα άχρηστα υλικά (πρόσθετη ή ξένη λειτουργικότητα, η περισσότερη από την οποία καθορίζεται από τον χρήστη). Για εκείνα τα frameworks που δημιουργούν κώδικα, για παράδειγμα, η “κομψότητα” θα υπονοούσε τη δημιουργία κώδικα ο οποίος είναι καθαρός και κατανοητός σε έναν αρκετά πεπειραμένο προγραμματιστή (και ο οποίος είναι, συνεπώς, εύκολα τροποποιήσιμος), σε σχέση με εκείνον που απλά δημιουργεί σωστό κώδικα. Το θέμα της κομψότητας είναι ο λόγος που σχετικά λίγα software frameworks έχουν αντέξει στη δοκιμασία του χρόνου: τα καλύτερα frameworks έχουν καταφέρει να εξελιχθούν ομαλά καθώς η υποβόσκουσα τεχνολογία, πάνω στην οποία αναπτύχθηκαν, προχώρησε. Ακόμα και εκεί, έχοντας εξελιχθεί, πολλά τέτοια πακέτα θα διατηρήσουν δυνατότητες κληρονομιάς φουσκώνοντας το τελικό λογισμικό, όπως άλλες αντικατεστημένες μέθοδοι έχουν διατηρηθεί παράλληλα με τις καινούριες μεθόδους.

## 2.9.2 Παραδείγματα

Τα software frameworks συνήθως περιέχουν σημαντικό οικιακό και χρηστικό κώδικα προκειμένου να βοηθήσουν την εκκίνηση εφαρμογών χρηστών, αλλά γενικά επικεντρώνονται σε συγκεκριμένα προβλήματα τομέων όπως:

- Καλλιτεχνικό σχέδιο, μουσική σύνθεση και μηχανική CAD
- Εφαρμογές χρηματοοικονομικών μοντέλων
- Εφαρμογές συστημικών μοντέλων σχετικά με τη Γη
- Συστήματα υποστήριξης αποφάσεων
- Πολυμεσικά framework – Αναπαραγωγή πολυμέσων και συγγραφή
- Ajax framework / JavaScript framework
- Web frameworks

- Ενδιάμεσο λογισμικό (Middleware)
- Cactus framework – Υψηλής απόδοσης επιστημονική πληροφορική
- Application framework – Γενικές εφαρμογές γραφικής διεπαφής χρήστη (GUI-Graphical User Interface)
- Enterprise Architecture framework
- Framework Ανάπτυξης Εφαρμογών Oracle

### 2.9.3 Αρχιτεκτονική

Σύμφωνα με το βιβλίο Pree, τα software frameworks αποτελούνται από παγωμένα σημεία (frozen spots) και καυτά σημεία (hot spots). Τα παγωμένα σημεία καθορίζουν τη συνολική αρχιτεκτονική ενός λογισμικού συστήματος, δηλαδή τα βασικά συστατικά του και τις σχέσεις μεταξύ τους. Αυτά παραμένουν αμετάβλητα (παγωμένα) σε κάθε στιγμιότυπο του application framework. Τα καυτά σημεία αντιπροσωπεύουν εκείνα τα μέρη όπου οι προγραμματιστές χρησιμοποιώντας το πλαίσιο προσθέτουν το δικό τους κώδικα για να προσθέσουν τη λειτουργικότητα που είναι συγκεκριμένη για το δικό τους project.

Σε ένα αντικειμενοστραφή περιβάλλον, ένα πλαίσιο αποτελείται από αφηρημένες και συγκεκριμένες κλάσεις. Η συγκεκριμενοποίηση ενός τέτοιου framework αποτελείται από τη σύνθεση και την υποβάθμιση των υπαρχουσών κλάσεων.

Κατά την ανάπτυξη ενός συγκεκριμένου λογισμικού συστήματος με ένα πλαίσιο λογισμικού, οι κατασκευαστές χρησιμοποιούν τα καυτά σημεία σύμφωνα με τις ιδιαίτερες ανάγκες και απαιτήσεις του συστήματος. Τα Software frameworks βασίζονται στην Αρχή του Hollywood: “Μη μας καλέσετε, θα σας καλέσουμε εμείς.” Αυτό σημαίνει ότι οι, καθορισμένες από το χρήστη, κλάσεις (για παράδειγμα, νέες δευτερεύουσες κλάσεις), λαμβάνουν μηνύματα από τις προκαθορισμένες framework κλάσεις. Οι κατασκευαστές χειρίζονται συνήθως αυτό μέσω της εφαρμογής αφηρημένων υπερκλάσεων.

### 2.9.4 Πλεονεκτήματα του Framework

#### Φορητότητα

Ο τρόπος επιλογής της βάσης δεδομένων και συστήματος για Caching επιτρέπει την εγκατάσταση σε πολλές διαφορετικές εγκαταστάσεις server με διαφορετικές παραμέτρους και χαρακτηριστικά. Επίσης αν η εφαρμογή είναι ανοικτού κώδικα, με την δυνατότητα της φορητότητας είναι πιθανότερο περισσότεροι χρήστες να την χρησιμοποιήσουν.

### **Ταχύτερη ανάπτυξη**

Τα Frameworks έχουν μεγάλες συλλογές από βιβλιοθήκες και συμβάλουν στην γρήγορη υλοποίηση μιας εργασίας. Για παράδειγμα, δεν χρειάζεται να γράψεις κάποια κλάση για να συνδεθείς ή να διαχειριστείς τις βάσεις δεδομένων ή να γράψεις κλάσεις για την διαχείριση των χρηστών.

### **Ασφάλεια της εφαρμογής**

Τα χαρακτηριστικά ασφαλείας όπως είναι η πιστοποίηση χρηστών και η διαχείριση των δικαιωμάτων τους, διαχειρίζονται από το Framework. Οι εγγραφές στη βάση δεδομένων είναι ασφαλείς, χωρίς SQL Injections.

### **Υποστήριξη από την κοινότητα**

Τα Frameworks υποστηρίζονται από κοινότητες και κανάλια IRC. Σε περίπτωση που αντιμετωπίσεις κάποιο πρόβλημα με το Framework, μπορείς να απευθυνθείς σε μια κοινότητα χρηστών και να λάβεις απαντήσεις, όπως επίσης να δεις τα προβλήματα που αντιμετώπισαν άλλοι Developers.

### **Επεκτάσεις και μονάδες**

Κάποια από τα μέλη που συμμετέχουν στις κοινότητες υποστήριξης δημοσιεύουν δωρεάν επεκτάσεις για το framework και μονάδες που μπορείς να κατεβάσεις και να χρησιμοποιήσεις. Τέτοιες επεκτάσεις μπορεί να σας παρέχουν διασύνδεση με άλλες εφαρμογές μέσω κάποιου API.

## **2.9.5 Μειονεκτήματα του Framework**

### **Μαθαίνεις το Framework, όχι τη γλώσσα προγραμματισμού**

Εάν χρησιμοποιείς ένα Framework και γνωρίζεις ελάχιστα για τη γλώσσα πίσω από αυτό, θα μάθεις πως δουλεύει το Framework και όχι την ίδια τη γλώσσα. Ο τρόπος που γράφουμε κώδικα jQuery είναι διαφορετικός από τον τρόπο που γράφουμε Javascript. Δηλαδή, αν γνωρίζουμε jQuery, δεν σημαίνει ότι κατανοούμε την Javascript. Γι' αυτό είναι σημαντικό να είμαστε γνώστες των προγραμματιστικών γλωσσών για την σωστή αξιοποίηση του Framework.

## Περιορισμοί

Η βασική συμπεριφορά του Framework δεν μπορεί να τροποποιηθεί, υποδεικνύοντας ότι όταν χρησιμοποιείτε ένα Framework, πρέπει να σέβεσαι τους περιορισμούς και να εργαστείς με τον τρόπο που απαιτείται.

## Ο κώδικας είναι δημόσιος

Δεδομένου ότι το Framework είναι άμεσα διαθέσιμο σε όλους, προσφέρεται επίσης σε άτομα με κακές προθέσεις. Μπορεί να μελετηθεί και οι ατέλειες του μπορούν να χρησιμοποιηθούν εναντίον μας.

## 2.10 Laravel Framework

Το Laravel είναι ένα σχετικά νέο PHP framework (κυκλοφόρησε το 2011), όπου σύμφωνα με την πρόσφατη online έρευνα του Sitepoint, είναι το πιο δημοφιλές Framework ανάπτυξης μεταξύ των προγραμματιστών. Το Laravel διαθέτει ένα τεράστιο οικοσύστημα με μια άμεση πλατφόρμα φιλοξενίας και ανάπτυξης και το επίσημο website του, προσφέρει πολλά σεμινάρια προβολής με τίτλο Laracasts.

Το Laravel επιχειρεί να βγάλει τον πόνο από την ανάπτυξη διευκολύνοντας τα κοινά καθήκοντα που χρησιμοποιούνται στην πλειοψηφία των έργων ιστού όπως το authentication, routing, sessions, και caching.

Σκοπός του Laravel είναι να καταστήσει την διαδικασία ανάπτυξης μιας εφαρμογής ευχάριστη για τον προγραμματιστή χωρίς να θυσιάζει τη λειτουργικότητα της εφαρμογής. Οι ευτυχείς προγραμματιστές κάνουν τον καλύτερο κώδικα.

Το Laravel αναφέρεται ως Full Stack Framework, διότι χειρίζεται τα πάντα, όπως το web που εξυπηρετεί στη διαχείριση βάσεων δεδομένων και το HTML. Ένα ολοκληρωμένο περιβάλλον ανάπτυξης ιστού μπορεί να προσφέρει μια καλύτερη εμπειρία στον προγραμματιστή.

Ένα από τα ενδιαφέροντα χαρακτηριστικά του Laravel είναι ότι επιβάλλει κάποιους αρκετά σοβαρούς περιορισμούς στον τρόπο δομής των εφαρμογών σας στο διαδίκτυο. Παραδόξως, αυτοί οι περιορισμοί καθιστούν ευκολότερη τη δημιουργία εφαρμογών.





*Εικόνα 2 Λογότυπο Laravel Framework.*

### **2.10.1 Η Αρχιτεκτονική του framework**

Σημείο εκκίνησης για όλα τα requests μιας Laravel εφαρμογής είναι το `public/index.php`. Όλα τα requests κατευθύνονται εκεί από τον web server (Apache / Nginx). Το αρχείο `index.php` δεν περιέχει ιδιαίτερο κώδικα, αλλά αποτελεί το σημείο εκκίνησης του υπόλοιπου framework. Το αρχείο `index.php` φορτώνει τον autoloader που παράγεται από τον Composer και στη συνέχεια ανακτά ένα instance της Laravel εφαρμογής από το `bootstrap/app.php` script. Η πρώτη κίνηση που κάνει η Laravel είναι η δημιουργία ενός instance του application service container.

Έπειτα, το εισερχόμενο request αποστέλλεται είτε στο HTTP kernel είτε στο console kernel ανάλογα με τον τύπο αυτού. Αυτά τα δύο kernels αποτελούν το σημείο από το οποίο περνάνε όλα τα requests.

Το HTTP kernel κάνει extend τη κλάση `Illuminate\Foundation\Http\Kernel` η οποία ορίζει έναν πίνακα από bootstrappers τα οποία θα τρέξουν πριν εκτελεστεί το request. Αυτοί οι bootstrappers τροποποιούν το error handling, το logging, εντοπίζουν το application environment, και πραγματοποιούν μια σειρά από άλλες λειτουργίες που πρέπει να γίνουν πριν γίνει ο χειρισμός του request.

Το HTTP kernel ορίζει επίσης μια λίστα από HTTP middlewares από τα οποία όλα τα requests θα πρέπει να περάσουν προκειμένου να τα χειριστεί στη συνέχεια η εφαρμογή. Αυτά τα middlewares διαχειρίζονται την ανάγνωση και εγγραφή του HTTP session, αποφασίζουν αν η εφαρμογή βρίσκεται σε maintenance mode, πιστοποιούν το CSRF token, και άλλα.

## 2.10.2 Η δομή της εφαρμογής

Το root directory μιας φρέσκιας εφαρμογής σε Laravel περιέχει διάφορους φακέλους:

- Ο app φάκελος περιέχει τον κύριο κώδικα της εφαρμογής.
- Ο bootstrap φάκελος περιέχει μερικά αρχεία τα οποία ξεκινούν το framework και τροποποιούν το autoloading, καθώς και έναν cache φάκελο ο οποίος περιέχει μερικά αρχεία που δημιουργούνται από το framework για βελτιστοποίηση της απόδοσης κατά την εκκίνηση.
- Ο config φάκελος περιέχει όλα τα configuration αρχεία της εφαρμογής.
- Ο database φάκελος περιέχει τα migration και seeds της βάσης.
- Ο public φάκελος περιέχει τα assets (images, JavaScript, CSS, etc).
- Ο resources φάκελος περιέχει τα views, τα raw assets (LESS, SASS, CoffeeScript), και localization αρχεία.
- Ο storage φάκελος περιέχει compiled Blade templates, sessions, caches αρχείων, και άλλα αρχεία που δημιουργούνται από το framework. Αυτός ο φάκελος σπάει στους app, framework και logs φακέλους. Ο app φάκελος μπορεί να χρησιμοποιηθεί για να αποθηκευτούν ότα αρχεία χρησιμοποιεί η εφαρμογή. Ο framework φάκελος χρησιμοποιείται για να αποθηκεύσει αρχεία που παράγονται από το framework και caches. Τέλος, ο logs φάκελος περιέχει τα log αρχεία της εφαρμογής.
- Ο tests φάκελος περιέχει αυτοματοποιημένα tests.
- vendor φάκελος περιέχει τα Composer dependencies.

### **3 Κατασκευαζόμενα τεχνητά νευρωνικά δίκτυα**

Σε αυτό το κεφάλαιο θα γίνει μία περιγραφή των κατασκευαζόμενων τεχνητών νευρωνικών δικτύων. Αρχικά γίνεται μία αναλυτική περιγραφή του γενετικού αλγόριθμου και του τρόπου λειτουργίας του. Αναφέρονται επίσης κάποιες χαρακτηριστικές εφαρμογές του μαζί με τα πλεονεκτήματα και τα μειονεκτήματα του. Στη συνέχεια περιγράφεται η γραμματική εξέλιξη και τέλος παρουσιάζεται η σχέση των δύο αυτών μεθόδων με τα κατασκευαζόμενα τεχνητά νευρωνικά δίκτυα.

#### **3.1 Εισαγωγή στους γενετικούς αλγόριθμους**

Τα τελευταία τριάντα χρόνια, παρατηρείται ένα συνεχώς αυξανόμενο ενδιαφέρον για την ανάπτυξη συστημάτων επίλυσης προβλημάτων βασισμένων στις αρχές της Φυσικής Εξέλιξης [48]. Τα συστήματα αυτού του είδους λειτουργούν διατηρώντας έναν πληθυσμό κωδικοποιημένων πιθανών λύσεων του προβλήματος που προσπαθούμε να επιλύσουμε, και εφαρμόζοντας πάνω σε αυτόν διάφορες διαδικασίες εμπνευσμένες από τη βιολογική εξέλιξη. Έτσι, περνώντας από γενιά σε γενιά, τα συστήματα αυτά δημιουργούν συνεχώς νέους πληθυσμούς πιθανών λύσεων εξελίσσοντάς τους προηγούμενους πληθυσμούς [34].

Οι Γενετικοί Αλγόριθμοι (Genetic Algorithms) είναι ένα παράδειγμα τέτοιου συστήματος που μαζί με τον Εξελικτικό Προγραμματισμό (Evolutionary Programming.), τις Στρατηγικές Εξέλιξης (Evolution Strategies), τα Συστήματα Ταξινόμησης (Classifier Systems) και το Γενετικό Προγραμματισμό (Genetic Programming) αποτελούν μια κατηγορία συστημάτων επίλυσης προβλημάτων που είναι ευρύτερα γνωστή με τον όρο Εξελικτικοί Αλγόριθμοι (Evolutionary Algorithms).

Η πρώτη εμφάνιση των Γενετικών Αλγόριθμων (Γ.Α.) χρονολογείται στις αρχές του 1950, όταν διάφοροι επιστήμονες από το χώρο της βιολογίας αποφάσισαν να χρησιμοποιήσουν υπολογιστές στην προσπάθειά τους να προσομοιώσουν πολύπλοκα βιολογικά συστήματα. Η συστηματική τους ανάπτυξη όμως, που οδήγησε στην μορφή με την οποία είναι γνωστοί σήμερα, πραγματοποιήθηκε στις αρχές του 1970 από τον John Holland και τους συνεργάτες του στο Πανεπιστήμιο του Michigan.

Οι Γ.Α. χρησιμοποιούν ορολογία δανεισμένη από το χώρο της Φυσικής Γενετικής. Αναφέρονται σε άτομα (individuals) ή γενότυπους (genotypes) μέσα σε ένα πληθυσμό. Κάθε άτομο ή γενότυπος αποτελείται από χρωμοσώματα (chromosomes). Στους Γ.Α.

αναφερόμαστε συνήθως σε άτομα με ένα μόνο χρωμόσωμα. Τα χρωμοσώματα αποτελούνται από γονίδια (genes) που είναι διατεταγμένα σε γραμμική ακολουθία. Κάθε γονίδιο επηρεάζει την κληρονομικότητα ενός ή περισσότερων χαρακτηριστικών. Τα γονίδια που επηρεάζουν συγκεκριμένα χαρακτηριστικά γνωρίσματα του ατόμου βρίσκονται και σε συγκεκριμένες θέσεις του χρωματοσώματος. Κάθε χαρακτηριστικό γνώρισμα του ατόμου (όπως για παράδειγμα το χρώμα μαλλιών) έχει τη δυνατότητα να εμφανιστεί με διάφορες μορφές, ανάλογα με την κατάσταση στην οποία βρίσκεται το αντίστοιχο γονίδιο που το επηρεάζει. Οι διαφορετικές αυτές καταστάσεις που μπορεί να πάρει το γονίδιο καλούνται alleles (τιμές χαρακτηριστικού γνωρίσματος).

Κάθε γενότυπος αναπαριστά μια πιθανή λύση σε ένα πρόβλημα. Το «αποκωδικοποιημένο» περιεχόμενο ενός συγκεκριμένου χρωμοσώματος καλείται φαινότυπος (phenotype) (π.χ. ένας ζωντανός οργανισμός είναι ο φαινότυπος των 6 χρωμοσωμάτων του). Μια διαδικασία εξέλιξης που εφαρμόζεται πάνω σε ένα πληθυσμό αντιστοιχεί σε ένα εκτενές ψάξιμο στο χώρο των πιθανών λύσεων. Απαραίτητη προϋπόθεση για την επιτυχημένη έκβαση ενός τέτοιου ψαξίματος αποτελεί η εξισορρόπηση δύο διαδικασιών που είναι προφανώς αντικρουόμενες, της εκμετάλλευσης και διατήρησης των καλύτερων λύσεων, και της όσο το δυνατόν καλύτερης εξερεύνησης όλου του διαστήματος.

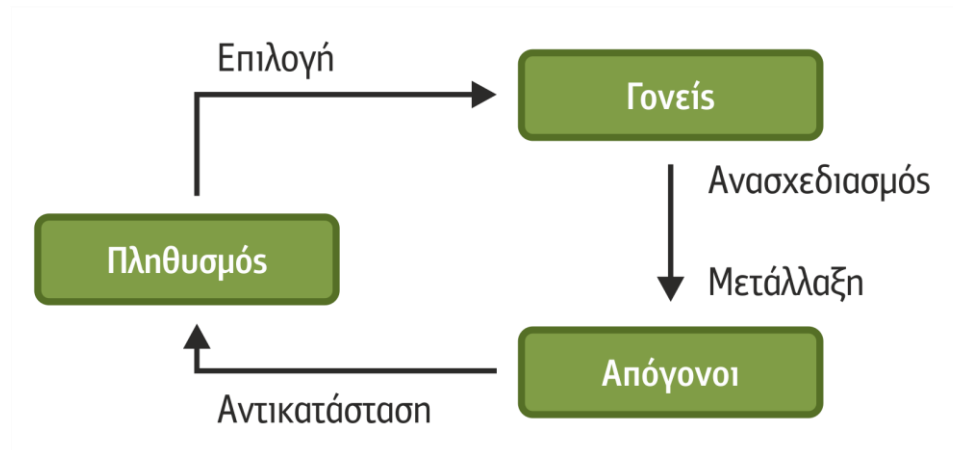
Οι Γ.Α. διατηρούν έναν πληθυσμό πιθανών λύσεων, του προβλήματος που μας ενδιαφέρει, πάνω στον οποίο δουλεύουν, σε αντίθεση με άλλες μεθόδους αναζήτησης που επεξεργάζονται ένα μόνο σημείο του διαστήματος αναζήτησης. Έτσι ένας Γ.Α. πραγματοποιεί αναζήτηση σε πολλές κατευθύνσεις και υποστηρίζει καταγραφή και ανταλλαγή πληροφοριών μεταξύ αυτών των κατευθύνσεων. Ο πληθυσμός υφίσταται μια προσομοιωμένη γενετική εξέλιξη. Σε κάθε γενιά, οι σχετικά "καλές" λύσεις αναπαράγονται, ενώ οι σχετικά "κακές" απομακρύνονται. Ο διαχωρισμός και αποτίμηση των διαφόρων λύσεων γίνεται με την βοήθεια μιας αντικειμενικής συνάρτησης (objective ή fitness function), η οποία παίζει το ρόλο του περιβάλλοντος μέσα στο οποίο εξελίσσεται ο πληθυσμός.

Η δομή ενός απλού Γ.Α. έχει σε γενικές γραμμές ως εξής [Michalewicz, 1996]: Κατά την διάρκεια της γενιάς  $t$ , ο Γ.Α. διατηρεί ένα πληθυσμό  $P(t)$  από  $n$  πιθανές λύσεις (individuals):  $P(t) = \{x_1^t, \dots, x_n^t\}$ . Κάθε χρωμόσωμα  $x_i^t$  αποτιμάται και δίνει ένα μέτρο της καταλληλότητας και ορθότητάς του. Αφού ολοκληρωθεί η αποτίμηση όλων των μελών του πληθυσμού, δημιουργείται ένας νέος πληθυσμός (γενιά  $t+1$ ) που προκύπτει από την επιλογή των πιο κατάλληλων στοιχείων του πληθυσμού της προηγούμενης γενιάς.

Μερικά μέλη από τον καινούργιο αυτό πληθυσμό υφίστανται αλλαγές με την βοήθεια των γενετικών διαδικασιών της διασταύρωσης και της μετάλλαξης σχηματίζοντας νέες πιθανές λύσεις. Η διασταύρωση συνδυάζει τα στοιχεία των χρωμοσωμάτων δύο γονέων για να δημιουργήσει δύο νέους απογόνους ανταλλάσσοντας κομμάτια από τους γονείς. Για παράδειγμα, έστω ότι δύο γονείς αναπαριστώνται με χρωματοσώματα πέντε γονιδίων (a1,b1,c1,d1,e1) και (a2,b2,c2,d2,e2) αντίστοιχα, τότε οι απόγονοι που θα προκύψουν από διασταύρωση με σημείο διασταύρωσης (crossover point) το σημείο 2 είναι οι 7 (a1,b1,c2,d2,e2) και (a2,b2,c1,d1,e1). Διαισθητικά μπορούμε να πούμε ότι η διασταύρωση εξυπηρετεί την ανταλλαγή πληροφοριών μεταξύ διαφορετικών πιθανών λύσεων. Η διαδικασία της μετάλλαξης αλλάζει αυθαίρετα ένα ή περισσότερα γονίδια ενός συγκεκριμένου χρωμοσώματος. Πραγματοποιείται με τυχαία αλλαγή γονιδίων με πιθανότητα ίση με το ρυθμό μετάλλαξης (mutation rate). Για παράδειγμα, έστω ότι ένα χρωμόσωμα αναπαρίσταται με το διάνυσμα πέντε διαστάσεων (a1,b1,c1,d1,e1), τότε το χρωμόσωμα που θα προκύψει με μετάλλαξη στη δεύτερη και στην τέταρτη διάσταση είναι ο(a1,b1 \*,c1,d1 \*,e1). Διαισθητικά μπορούμε να πούμε ότι η μετάλλαξη εξυπηρετεί την εισαγωγή νέων πιθανών λύσεων, διαφορετικών από τις υπάρχουσες, στον ήδη υπάρχοντα πληθυσμό.

Συνοψίζοντας μπορούμε να πούμε ότι ένας Γ.Α. για ένα συγκεκριμένο πρόβλημα πρέπει να αποτελείται από τα παρακάτω πέντε συστατικά:

1. Μια γενετική αναπαράσταση των πιθανών λύσεων του προβλήματος.
2. Έναν τρόπο δημιουργίας ενός αρχικού πληθυσμού από πιθανές λύσεις (αρχικοποίηση).
3. Μια αντικειμενική συνάρτηση αξιολόγησης των μελών του πληθυσμού, που παίζει το ρόλο του περιβάλλοντος.
4. Γενετικούς τελεστές για τη δημιουργία νέων μελών (λύσεων).
5. Τιμές για τις διάφορες παραμέτρους που χρησιμοποιεί ο Γ.Α. (μέγεθος πληθυσμού, πιθανότητες εφαρμογής των γενετικών τελεστών, κ.τ.λ.).



Σχήμα 22 Ο εξελικτικός κύκλος των Γενετικών Αλγορίθμων.

### 3.1.1 Η Θεωρία εξέλιξης των ειδών

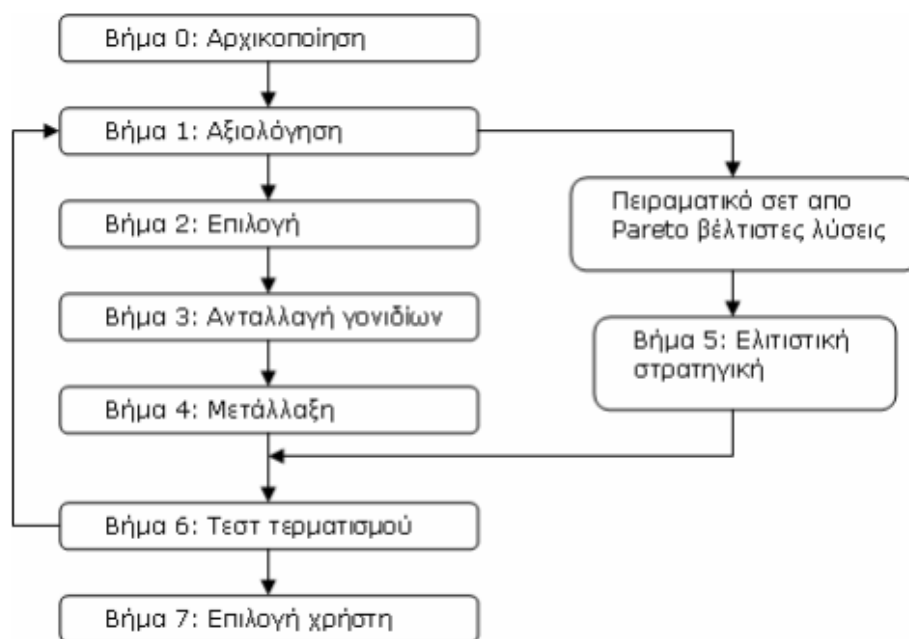
Ο τρόπος λειτουργίας των Γ.Α βασίζεται στους μηχανισμούς εξέλιξης και φυσικής επιλογής. Η φυσική επιλογή καθορίζει ποια μέλη ενός πληθυσμού επιβιώνουν και αναπαράγονται κι εξαρτάται από τις διάφορες συνθήκες που επικρατούν στο περιβάλλον [48]. Η θεωρία της φυσικής εξέλιξης διατυπώθηκε από το Δαρβίνο στα μέσα του 19ου αιώνα και τάραξε τις μέχρι τότε θρησκευτικές αντιλήψεις για την προέλευση της ζωής. Η θεωρία υποστήριζε πως σε ένα συνεχώς μεταβαλλόμενο περιβάλλον με φυσικές δυσκολίες και περιορισμένους πόρους, οι ζωντανοί οργανισμοί είναι αναγκασμένοι να βρουν τρόπο να επιβιώσουν και να αναπαραχθούν.

### 3.1.2 Γενικός τρόπος λειτουργίας των γενετικών αλγορίθμων

Πρώτο βήμα της διαδικασίας είναι η αρχικοποίηση ενός πληθυσμού πιθανών λύσεων με τυχαίο τρόπο. Κάθε χρωμόσωμα αποτελεί μια πιθανή λύση του προβλήματος που έχει τεθεί. Το χρωμόσωμα αυτό είναι ένα διάνυσμα στο οποίο περιέχονται τιμές για τις μεταβλητές του προβλήματος. Στη συνέχεια αξιολογούνται τα άτομα βάσει των τιμών της αντικειμενικής συνάρτησης του προβλήματος που το καθένα δίνει [34]. Η αντικειμενική συνάρτηση παίζει το ρόλο του περιβάλλοντος στη φυσική επιλογή. Από τις τιμές που προκύπτουν ο αλγόριθμος κατατάσσει τα χρωμοσώματα βάσει της πιθανότητας τους να περάσουν στην επόμενη γενιά. Η πιθανότητα υπολογίζεται από την τιμή της αντικειμενικής συνάρτησης που έδωσε το κάθε άτομο προς το άθροισμα των τιμών της αντικειμενικής συνάρτησης όλων

των ατόμων. Το επόμενο και χαρακτηριστικό βήμα του γενετικού αλγόριθμου είναι η δημιουργία τη επόμενης γενιάς πιθανών λύσεων, μέσω της διασταύρωσης και της μετάλλαξης [48]. Τις περισσότερες φορές χρησιμοποιείται και η ελιτίστικη επιλογή για τη δημιουργία του πληθυσμού, η οποία είναι μια διαδικασία που δεν είναι εμπνευσμένη από τη φύση. Στόχος της είναι η εξασφάλιση πως ένα τουλάχιστον από τα καλύτερα μέλη του πληθυσμού θα περάσει στην επόμενη γενιά, βεβαιώνοντας έτσι πως σε περίπτωση που βρεθεί το ολικό ελάχιστο, ο αλγόριθμος θα συγκλίνει σε αυτό. Έτσι, αρχικά επιλέγονται τα καλύτερα άτομα που θα περάσουν ως έχουν στην επόμενη γενιά. Από τα υπόλοιπα κάποια θα επιλεχθούν με πιθανότητα  $p_c$  για να υποστούν διασταύρωση και τα υπόλοιπα με πιθανότητα  $p_m$  (πιθανότητα μετάλλαξης) θα υποστούν μετάλλαξη και θα αλλάξει η δομή τους, δηλαδή η τιμή που έχουν, με τυχαίο τρόπο. Ο νέος πληθυσμός αξιολογείται και ακολουθείται η ίδια διαδικασία μέχρι να ικανοποιηθεί μια συνθήκη τερματισμού.

Η διαδικασία αναπαρίσταται σχηματικά στο παρακάτω διάγραμμα:



Σχήμα 23 Διάγραμμα ροής εκτέλεσης ενός Γ.Α.

### 3.1.3 Ορολογία Γενετικών Αλγορίθμων

1. **Χρωμόσωμα-Γονίδιο:** Το χρωμόσωμα είναι ένα διάνυσμα στο οποίο είναι αποθηκευμένες οι πιθανές τιμές των μεταβλητών του προβλήματος που έχει οριστεί. Κάθε πιθανή τιμή αποτελεί ένα γονίδιο. T
2. **Πληθυσμός-Γενιά:** Ο αριθμός των χρωμοσωμάτων που παράγεται σε κάθε επανάληψη και αποτελούν μια γενιά. Όσο μεγαλύτερος είναι ο πληθυσμός τόσο καλύτερη εξερεύνηση γίνεται στο χώρο των πιθανών λύσεων.
3. **Κωδικοποίηση:** Οι πιθανές τιμές που περιέχουν τα χρωμοσώματα κωδικοποιούνται βάσει κάποιου αλφαβήτου.
4. **Συνάρτηση καταλληλότητας:** Η συνάρτηση δέχεται την αποκωδικοποιημένη τιμή των χρωμοσωμάτων κάθε γενιάς ως είσοδο. Μικρές τιμές υποδηλώνουν ακατάλληλα χρωμοσώματα, ενώ άτομα με μεγάλες τιμές είναι αυτά που έδωσαν την καλύτερη τιμή στην αντικειμενική συνάρτηση. Από την τιμή αυτή εξαρτάται η επιλογή των χρωμοσωμάτων που θα περάσουν στην επόμενη γενιά [5].
5. **Επιλογή:** Αφορά στα άτομα που θα δημιουργήσουν την επόμενη γενιά. Κάποια θα επιλεγθούν να περάσουν αυτούσια στην επόμενη γενιά, άλλα θα επιλεγθούν ως γονείς.
6. **Διασταύρωση:** Ο συνδυασμός δύο χρωμοσωμάτων με σκοπό την αναπαραγωγή τέκνων με χαρακτηριστικά και από τους δύο γονείς.
7. **Μετάλλαξη:** Τυχαίες αλλαγές σε ένα χρωμόσωμα. Βοηθάει στην καλύτερη αναζήτηση στο πεδίο ορισμού του προβλήματος.

### 3.1.4 Βασικά χαρακτηριστικά γενετικών αλγορίθμων

Τα κυριότερα χαρακτηριστικά που διαφοροποιούν τους Γενετικούς Αλγορίθμους από άλλες τεχνικές αλλά και τους δίνουν υπεροχή, είναι τα εξής:

1. Οι Γ.Α. δουλεύουν με μια κωδικοποίηση του συνόλου τιμών που μπορούν να λάβουν οι μεταβλητές και όχι με τις ίδιες τις μεταβλητές του προβλήματος: Για παράδειγμα, αναφέρεται το εξής πρόβλημα βελτιστοποίησης: Έστω ένα μαύρο κουτί με πέντε δυαδικούς διακόπτες (on-off). Για κάθε συνδυασμό  $s$  των διακοπών παράγεται μία έξοδος  $f(s)$ . Ζητείται ο συνδυασμός των διακοπών που μεγιστοποιεί την έξοδο. Σε ένα Γ.Α. όμως, η πρώτη ενέργεια είναι η κωδικοποίηση των διακοπών ως συμβολοσειρές



πεπερασμένου μήκους [8]. Μια απλή κωδικοποίηση θα μπορούσε να γίνει θεωρώντας μια δυαδική συμβολοσειρά μήκους πέντε, όπου η κάθε θέση αναπαριστά ένα διακόπτη. Το 0 αντιστοιχεί στη θέση off και το 1 στη θέση on. Δηλαδή, η συμβολοσειρά 11110 κωδικοποιεί το συνδυασμό κατά τον οποίο οι πρώτοι τέσσερις διακόπτες είναι on και ο τελευταίος off. Η κωδικοποίηση δεν είναι απαραίτητο να είναι πάντα δυαδική.

2. Οι Γ.Α. κάνουν αναζήτηση σε πολλά σημεία ταυτόχρονα και όχι μόνο σε ένα: Οι Γ.Α ενεργούν ταυτόχρονα πάνω σε ένα ευρύ σύνολο σημείων (σύνολο από συμβολοσειρές). Γυρίζοντας στο παράδειγμα με το μαύρο κουτί, οι κλασσικές μέθοδοι θα ξεκινούσαν το ψάξιμο από ένα συνδυασμό των διακοπών και στη συνέχεια, εφαρμόζοντας κάποιο κανόνα μετάβασης, θα δοκίμαζαν τον επόμενο. Αντιθέτως, ένας Γ.Α. αρχίζει το ψάξιμο του από ένα πληθυσμό συνδυασμών συμβολοσειρών και κατόπιν παράγει διαδοχικά καινούριους. Ένας αρχικός πληθυσμός θα μπορούσε να είναι, π.χ. 01101, 11000, 01000 και 10011. Έπειτα ο αλγόριθμος δημιουργεί νέους πληθυσμούς που συγκλίνουν προς την επιθυμητή λύση [48].
3. Οι Γ.Α. χρησιμοποιούν μόνο την αντικειμενική συνάρτηση και καμία επιπρόσθετη πληροφορία: Πολλές μέθοδοι αναζήτησης απαιτούν αρκετές βοηθητικές πληροφορίες για τη συνάρτηση που επεξεργάζονται. Τέτοιου είδους πληροφορίες δεν προ απαιτούνται από τους Γ.Α.

### 3.1.5 Αρχικοποίηση

Σε αυτή τη φάση δημιουργείται ο αρχικός πληθυσμός από αρχικές λύσεις πάνω στον οποίο θα εφαρμοστεί ο γενετικός αλγόριθμος. Ο πληθυσμός αυτός επιλέγεται με τυχαίο τρόπο ανάμεσα σε όλες τις δυνατές τιμές των μεταβλητών του προβλήματος, ενώ το μέγεθός του είναι παράμετρος που καθορίζεται από το χρήστη και παραμένει σταθερό σε όλη τη διάρκεια λειτουργίας του αλγορίθμου. Αφού προκύψει η πρώτη γενιά, ο γενετικός αλγόριθμος μεταβαίνει στο επαναληπτικό του μέρος.

### 3.1.6 Αξιολόγηση καταλληλότητας

Εδώ υπολογίζεται η ικανότητα επιβίωσης κάθε ατόμου του πληθυσμού. Για κάθε συμβολοσειρά του τρέχοντος πληθυσμού υπολογίζεται η απόδοσή της από την αντικειμενική συνάρτηση.

### 3.1.7 Επιλογή

Στη φάση αυτή καθορίζεται ποια άτομα από τον υπάρχον πληθυσμό θα συμμετέχουν στην αναπαραγωγή και θα κληροδοτήσουν στην επόμενη γενιά το σύνολο ή μέρος των χαρακτηριστικών επιτρέποντας την αύξηση των ικανότερων ατόμων. Υπάρχουν διάφοροι τρόποι υλοποίησης της επιλογής [48]. Ο πιο γνωστός είναι η μέθοδος ρουλέτας, στην οποία κάθε συμβολοσειρά ενός πληθυσμού αντιπροσωπεύεται σε ένα μέρος της ρουλέτας, σε αναλογία με την απόδοσή της. Δηλαδή, μέλη με μεγάλη απόδοση κατέχουν μεγαλύτερες σχισμές της ρουλέτας, και κατά συνέπεια έχουν μεγαλύτερη πιθανότητα να επιλεγούν και να περάσουν στην επόμενη γενιά.

Οι πιο γνωστές μέθοδοι με τις οποίες γίνεται η επιλογή είναι οι εξής:

**Επιλογή διαγωνισμού (tournament selection):** Αυτή η μέθοδος επιλογής είναι η πιο απλή. Επιλέγεται τυχαία ένας αριθμός χρωμοσωμάτων (συνήθως δύο) από τον πληθυσμό και το χρωμόσωμα με την ψηλότερη τιμή ικανότητας επιλέγεται.

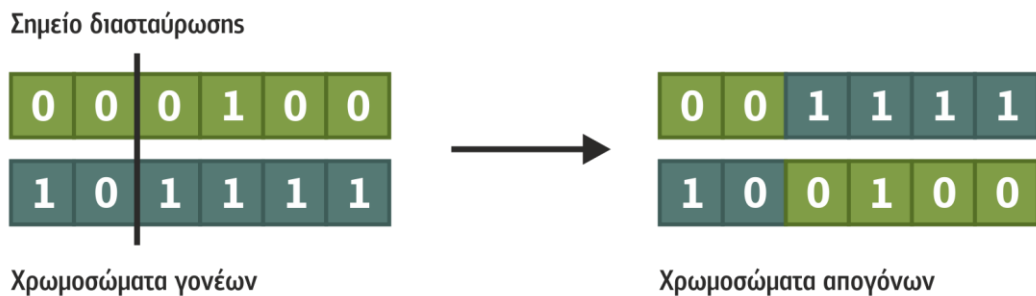
**Επιλογή ρουλέτας (roulette wheel selection):** Στην επιλογή ρουλέτας αρχικά δίνονται οι τιμές καταλληλότητας στα άτομα. Ανάλογα με την καταλληλότητα τους έχουν και την αντίστοιχη πιθανότητα για να επιλεγούν. Σε αυτό το σημείο γίνεται μίμηση της ρουλέτας και σε όποιο χρωμόσωμα σταματήσει η μπίλια τότε αυτό επιλέγεται. Προφανώς τα χρωμοσώματα με μεγαλύτερες τιμές ικανότητας θα επιλεγούν περισσότερες φορές.

### 3.1.8 Αναπαραγωγή

Η λειτουργία της αναπαραγωγής περιλαμβάνει υπολειτουργίες που επιτυγχάνονται με τη χρήση των γενετικών τελεστών της διασταύρωσης (crossover, mating), της μετάλλαξης (mutation) και της αντιμετάθεσης (permutation, swap).

### 3.1.9 Διασταύρωση

Ο πληθυσμός που προέκυψε από τη διαδικασία της επιλογής πρέπει να περάσει από τη διαδικασία της διασταύρωσης. Στόχος είναι η νέα γενιά να περιλαμβάνει άτομα που θα διαφέρουν από τους γονείς τους και θα φέρουν συνδυασμό των καλύτερων χαρακτηριστικών τους. Η ομάδα ατόμων που προέκυψε από την επιλογή, σχηματίζει ζευγάρια με τυχαίο τρόπο. Τα επιλεγμένα ζευγάρια διασταυρώνονται σε ένα τυχαίο σημείο που ονομάζεται σημείο διασταύρωσης. Για παράδειγμα, έστω οι γονείς  $P1$  ( 1 0 1 0 0 1 0 ) και  $P2$  ( 1 0 1 1 0 0 1 ) και το σημείο διασταύρωσης είναι μετά το τρίτο γονίδιο. Οι απόγονοι που θα προκύψουν μετά από τη διασταύρωση αυτή θα είναι οι  $O1$  ( 1 0 1 1 0 0 1 ) και  $O2$  ( 1 0 1 0 0 1 0 ). Τα νέα άτομα περιλαμβάνουν συνδυασμούς χαρακτηριστικών των γονέων τους και με αυτό τον τρόπο μπορούν να προκύψουν επιτυχημένοι συνδυασμοί υψηλής ικανότητας. Φυσικά υπάρχει και το ενδεχόμενο η διασταύρωση να δώσει λιγότερο ικανά παιδιά από τους γονείς, αλλά οι απόγονοι αυτοί δεν θα έχουν μεγάλη πιθανότητα πολλαπλασιασμού στον επόμενο αναπαραγωγικό κύκλο, λόγω μικρής απόδοσης (fitness).



Σχήμα 24 Διασταύρωση ενός σημείου.

Δεν υπάρχει γενικά αποδεκτός τρόπος διασταυρώσεως, ωστόσο ποιο συχνά συναντάμε τους ακόλουθους:

1. **Διασταύρωση ενός σημείου.** Η ανταλλαγή γενετικού υλικού γίνεται με την επιλογή ενός σημείου στα χρωμοσώματα και την ανταλλαγή του γενετικού υλικού που υπάρχει πριν και μετά από αυτό. Στην πράξη πολύ συχνά χρησιμοποιούμε την διασταύρωση ενός σημείου και την διασταύρωση δύο σημείων. Στο σχήμα 3.3 βλέπουμε την διασταύρωση ενός σημείου.
2. **Διασταύρωση δύο σημείων.** Σε αυτήν την μορφή διασταυρώσεως τα χρωμοσώματα θεωρούνται πως είναι κυκλικά με το πρώτο και το τελευταίο στοιχείο τους να είναι

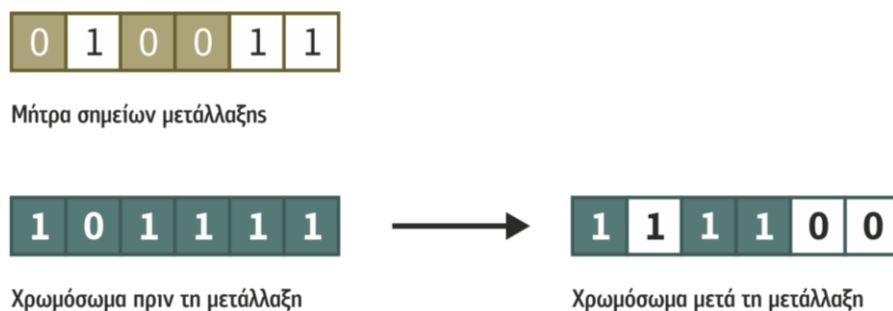
συνδεδεμένα. Επιλέγονται τυχαία 2 σημεία πάνω στα κυκλικά χρωμοσώματα και τα αντίστοιχα τμήματα, ανταλλάσσονται.

- 3. Ομοιόμορφη διασταύρωση.** Σε αυτήν την εκδοχή διασταυρώσεως για κάθε στοιχείο των δύο χρωμοσωμάτων επιλέγεται ένας τυχαίος αριθμός στο διάστημα  $[0,1]$ . Αν αυτός είναι μικρότερος του 0,5 τότε το στοιχείο του πρώτου χρωμοσώματος πηγαίνει στον απόγονο, διαφορετικά το στοιχείο του δεύτερου χρωμοσώματος.

Συνήθως δεν διασταυρώνονται όλα τα μεγέθη ενός πληθυσμού. Το ποσοστό των μελών που θα διασταυρωθούν συνήθως καθορίζεται από ένα συντελεστή επιλογής, ο οποίος είναι στο διάστημα  $[0,1]$ . Όσο μεγαλύτερη είναι η τιμή αυτού του συντελεστή, τόσο περισσότερες διασταυρώσεις θα πραγματοποιηθούν.

### 3.1.10 Μετάλλαξη

Η μετάλλαξη επιλέγει τυχαία και με μικρή πιθανότητα (πιθανότητα μετάλλαξης) γονίδια από τα χρωμοσώματα των ατόμων του πληθυσμού και μεταβάλλει την τιμή τους. Για παράδειγμα έστω τα προηγούμενα χρωμοσώματα  $P1 (1\ 0\ 1\ 0\ 0\ 1\ 0)$  και  $P2 (1\ 0\ 1\ 1\ 0\ 0\ 1)$ . Πρώτα πραγματοποιείται μία μετάλλαξη στο γονίδιο 2 και οι απόγονοι που προκύπτουν είναι οι  $O1 (1\ 1\ 1\ 0\ 0\ 1\ 0)$  και  $O2 (1\ 1\ 1\ 1\ 0\ 0\ 1)$ . Ύστερα πραγματοποιείται μία διασταύρωση στο σημείο 4 και οι τελικοί απόγονοι είναι οι  $O1 (1\ 1\ 1\ 0\ 0\ 0\ 1)$  και  $O2 (1\ 1\ 1\ 1\ 0\ 1\ 0)$ . Η πιθανότητα μετάλλαξης είναι πολύ μικρή, διότι αλλιώς ο γενετικός αλγόριθμος εκφυλίζεται σε τυχαίο ψάξιμο [34].



Σχήμα 25 Μετάλλαξη τριών σημείων.

### **3.1.11 Πλεονεκτήματα γενετικών αλγορίθμων**

Τα πλεονεκτήματα των Γενετικών Αλγορίθμων είναι τα εξής:

1. Οι Γενετικοί Αλγόριθμοι είναι ικανοί να αντιμετωπίσουν προβλήματα των οποίων η λύση δύσκολα προσδιορίζεται, καθώς επίσης βρίσκουν εύκολα ακρότατα σε συναρτήσεις με έντονες διακυμάνσεις, όπου άλλες μέθοδοι αποτυγχάνουν.
2. Είναι εύκολα επεκτάσιμοι και εξελίξιμοι. Μπορούν να υποστούν τροποποιήσεις και αλλαγές για να προσαρμοστούν στις ιδιαιτερότητες του εκάστοτε προβλήματος.
3. Είναι ευέλικτοι και μπορούν να συμμετέχουν σε υβριδικές μορφές με άλλες μεθόδους.
4. Απαιτούν μόνο την αντικειμενική συνάρτηση για να ξεκινήσουν, η οποία δεν χρειάζεται να υπόκειται σε περιορισμούς όπως η συνέχειά της.
5. Επιτυγχάνουν τον συνδυασμό της εξερεύνησης του χώρου αναζήτησης και της εκμετάλλευσης της ήδη επεξεργασμένης πληροφορίας.

### **3.1.12 Μειονεκτήματα Γενετικών Αλγορίθμων**

Τα κυριότερα μειονεκτήματά των γενετικών αλγορίθμων είναι:

1. Πρόβλημα εξοικείωσης με την γενετική. Καθώς η ιδέα των γενετικών αλγορίθμων προέρχεται από την Βιολογία οι όροι που χρησιμοποιούνται για την κατανόηση τους μπορεί να ηχούν παράξενα σε αυτούς που ασχολούνται με την επιστήμη της πληροφορικής.
2. Το πρόβλημα του χρόνου. Υπάρχει δυσπιστία σχετικά με την αποδοτικότητα των γενετικών αλγορίθμων λόγω του ότι βασίζονται σε διαδικασίες που στην φύση απαιτούν μεγάλο υπολογιστικό χρόνο.
3. Παρουσιάζουν αργό χρόνο σύγκλισης
4. Μπορούν να εγκλωβιστούν σε τοπικά ελάχιστα της αντικειμενικής συνάρτησης.

### 3.1.13 Εφαρμογές

Οι γενετικοί αλγόριθμοι έχουν ένα ευρύ φάσμα εφαρμογών όπως φαίνεται παρακάτω:

- Βέλτιστη τοποθέτηση ενεργών στοιχείων για μείωση θορύβου [48]
- Σχεδίαση μηχανών αεροπλάνων
- Αναγνώριση προσώπων
- Προγραμματισμός υπολογιστών
- Συστήματα αναγνώρισης εισβολών
- Σχεδίαση τηλεπικοινωνιακών δικτύων

### 3.1.14 Εφαρμογές των ΓΑ για τη βελτιστοποίηση των ΤΝΔ

Μια εφαρμογή των γενετικών αλγορίθμων, που παρουσιάζει ενδιαφέρον στα πλαίσια της παρούσας πτυχιακής εργασίας, είναι η χρήση τους για τη βελτιστοποίηση προβλημάτων που σχετίζονται με την ανάπτυξη, την εκπαίδευση και τις εφαρμογές των ΤΝΔ [48].

Η εργασία αυτή είναι ιδιαίτερα δύσκολη, γιατί πρέπει να καθοριστεί τόσο η αρχιτεκτονική και η δομή του ΤΝΔ, όσο και οι παράμετροι της εκπαίδευσης μέσα από ένα ευρύτατο σύνολο επιλογών. Το πρόβλημα αυτό αντιμετωπίζεται κυρίως με ευρετικούς αλγόριθμους, που όμως δεν εξυπηρετούν απόλυτα τους στόχους της βελτιστοποίησης, αφενός επειδή ο χώρος αναζήτησης είναι τεράστιος και αφετέρου γιατί η έννοια της καλής αρχιτεκτονικής εξαρτάται άμεσα από την εφαρμογή για την οποία θα χρησιμοποιηθεί το ΤΝΔ.

Κατά την τελευταία περίπου δεκαετία, οι ερευνητές έχουν καταφέρει να αναπτύξουν εξελικτικές μεθόδους ικανές να καλύψουν τις περισσότερες απαιτήσεις ανάπτυξης, εκπαίδευσης και εφαρμογών των ΤΝΔ. Στις εξελικτικές αυτές μεθόδους περιλαμβάνονται μέθοδοι αναζήτησης βέλτιστων τιμών των βαρών, μέθοδοι βελτιστοποίησης της συνολικής αρχιτεκτονικής του ΤΝΔ, των συναρτήσεων ενεργοποίησης και, τέλος, μέθοδοι βελτιστοποίησης των παραμέτρων εκπαίδευσης. Στην παρούσα εργασία, όπως θα περιγραφεί αναλυτικά στη συνέχεια, οι γενετικοί αλγόριθμοι, χρησιμοποιήθηκαν για τη βελτιστοποίηση της αρχιτεκτονικής των νευρωνικών δικτύων.

## 3.2 Εισαγωγή στην Γραμματική Εξέλιξη (Grammatical evolution)

Η τεχνική της Γραμματικής Εξελίξεως είναι μια νέα εξελικτική που αναπτύχθηκε από τους Ο' Neil και Ryan στην εργασία και βασίζεται στην περιγραφή με BNF των γραμματικών για την παραγωγή προγραμμάτων σε οποιαδήποτε γλώσσα προγραμματισμού [44], [48]. Η τεχνική αυτή έχει εφαρμοστεί με επιτυχία σε μία σειρά από προβλήματα όπως έρευνα τριγωνομετρικών ισοτήτων, ρομποτική, αυτόματη παραγωγή αλγορίθμων για cashing, οικονομικά κτλ. Περισσότερα για αυτήν την τεχνική μπορεί να βρει κανείς στα σχετικά άρθρα [4], [8], [26],[27],[28],[29] όπως και στον σχετικό ιστοχώρο <http://www.grammatical-evolution.org>.

### 3.2.1 Βασικοί ορισμοί

Η τεχνική του Grammatical Evolution είναι ένας εξελικτικός αλγόριθμος, ο οποίος μπορεί να παράγει προγράμματα σε οποιαδήποτε γλώσσα προγραμματισμού. Ο αλγόριθμος απαιτεί την γραμματική της αντικειμενικής γλώσσας εκφρασμένη σε BNF μορφή και μια συσχετιζόμενη συνάρτηση καταλληλότητας [8]. Τα χρωμοσώματα στην διαδικασία του Grammatical Evolution δεν εκφράζονται με συντακτικά δένδρα παραγωγής, όπως είδαμε στην προηγούμενη ενότητα, αλλά σαν διανύσματα ακεραίων αριθμών. Κάθε ακέραιος συμβολίζει και έναν κανόνα παραγωγής από την αντίστοιχη BNF γραμματική. Ο αλγόριθμος παραγωγής εκφράσεων ξεκινά από το σύμβολο εκκινήσεως της γραμματικής και βαθμιαία δημιουργεί το αντίστοιχο πρόγραμμα αντικαθιστώντας μη τερματικά σύμβολα με το δεξιό τμήμα του επιλεγμένου κανόνα παραγωγής [14]. Η επιλογή του κανόνα παραγωγής πραγματοποιείται σε δύο βήματα:

- Διαβάζουμε ένα στοιχείο από το χρωμόσωμα, έστω με τιμή  $V$ .
- Διαλέγουμε τον αντίστοιχο κανόνα παραγωγής σύμφωνα με το σχήμα

$$\text{Rule} = V \bmod NR$$

όπου  $NR$  είναι ο αριθμός των κανόνων που διαθέτει το μη - τερματικό σύμβολο το οποίο θα αντικατασταθεί, [26],[27],[28],[29]. Η διαδικασία της αντικαταστάσεως μη τερματικών συμβόλων με το δεξιό τμήμα κανόνων παραγωγής συνεχίζεται μέχρι να συμβεί μία από τις δύο επόμενες καταστάσεις:

1. Να έχει παραχθεί ένα έγκυρο πρόγραμμα στην αντικειμενική γλώσσα προγραμματισμού. Αυτό σημαίνει πως δεν υπάρχουν πλέον μη τερματικά σύμβολα στην έκφραση για να αντικατασταθούν και η διαδικασία τερματίζει με επιτυχία [23], [44].
2. Να έχουν διαβαστεί όλα τα στοιχεία από το χρωμόσωμα και να μην έχει παραχθεί έκφραση που δεν περιέχει μη - τερματικά σύμβολα. Σε αυτήν την περίπτωση είτε το χρωμόσωμα απορρίπτεται και του ανατίθεται μια τιμή καταλληλότητας τέτοια που να μην του επιτρέπει να συμμετάσχει στην διαδικασία αναπαραγωγής ή η ανάγνωση των στοιχείων του αναδίπλωση και συνήθως 1-2 αναδιπλώσεις είναι αρκετές είτε για να παράγουν μια έγκυρη έκφραση είτε για να δείξουν πως το συγκεκριμένο χρωμόσωμα δεν είναι σε θέση να παράγει έγκυρες εκφράσεις [38].

### 3.2.2 Παράδειγμα παραγωγής

Για να γίνει περισσότερο κατανοητή η διαδικασία παραγωγής προγραμμάτων έστω πως διατίθεται η γραμματική του σχήματος 26 που παριστά πράξεις προσθέσεως και αφαιρέσεως ανάμεσα στο  $x$  και το  $y$ . Το σύμβολο  $S$  είναι το αρχικό σύμβολο της γραμματικής και στους υλοποιημένους αλγορίθμους αυτού του κειμένου γίνεται η υπόθεση πως το αρχικό σύμβολο θα έχει πάντα έναν κανόνα παραγωγής [23], [38], [42], [43], [44]. Οι αριθμοί στο τέλος των κανόνων παραγωγής διακρίτοποιούν τους κανόνες μεταξύ τους. [4] Έστω το χρωμόσωμα [8,2,5,3,3]. Σταδιακά θα γίνουν οι επόμενες παραγωγές του σχήματος 27.

$$\begin{aligned}
 S &::= \langle \text{expr} \rangle \\
 \langle \text{expr} \rangle &::= \langle \text{expr} \rangle + \langle \text{expr} \rangle \quad (0) \\
 | \langle \text{expr} \rangle - \langle \text{expr} \rangle \quad (1) \\
 | x \quad (2) \\
 | y \quad (3)
 \end{aligned}$$

*Σχήμα 26* Γραμματική μια γλώσσας με πράξεις ανάμεσα σε  $x$  και  $y$ .

$$\begin{aligned}
 S &::= \langle \text{expr} \rangle \\
 \langle \text{expr} \rangle &+ \langle \text{expr} \rangle \\
 x &+ \langle \text{expr} \rangle \\
 x &+ \langle \text{expr} \rangle - \langle \text{expr} \rangle \\
 x &+ y - \langle \text{expr} \rangle \\
 x &+ y - y
 \end{aligned}$$

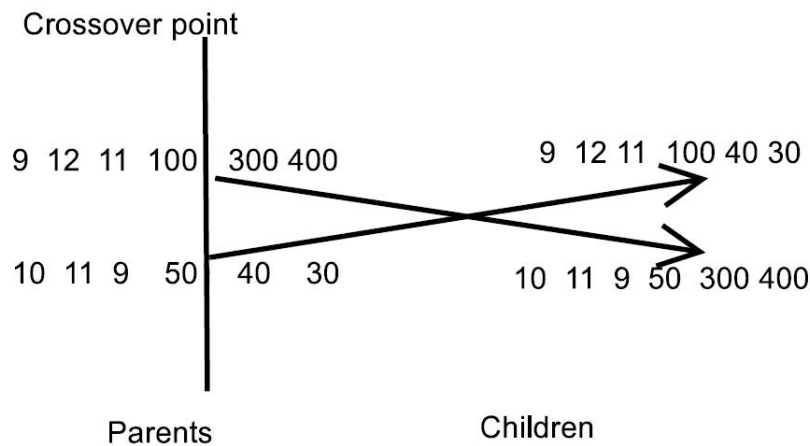
*Σχήμα 27* Παράδειγμα παραγωγής έκφρασης.



σύμβολο της γραμματικής αντικαθίσταται από το σύμβολο  $\langle \text{expr} \rangle$ , αφού ο μόνος γραμματικός κανόνας που υπάρχει από το αρχικό σύμβολο είναι αυτός που οδηγεί στο  $\langle \text{expr} \rangle$ . Στην συνέχεια επειδή το  $\langle \text{expr} \rangle$  είναι μη τερματικό σύμβολο διαβάζεται ένα στοιχείο από το χρωμόσωμα προκειμένου να παραχθεί συμβολοσειρά με τερματικά σύμβολα (που είναι και ο σκοπός κάθε γραμματικής) [8]. Το επόμενο σύμβολο στο χρωμόσωμα είναι το 8. Το πλήθος των κανόνων του συμβόλου  $\langle \text{expr} \rangle$  είναι 4 και  $8 \% 4 = 0$ . Αυτό έχει σαν αποτέλεσμα να επιλεγεί ο πρώτος κανόνας που θα δώσει  $\langle \text{expr} \rangle + \langle \text{expr} \rangle$ . Στην συνέχεια καθένα από τα μη τερματικά  $\langle \text{expr} \rangle$  που παρήχθησαν πρέπει να αναχθεί σε σειρά τερματικών συμβόλων, [26],[27],[28],[29]. Με την ανάγνωση ακόμα ενός συμβόλου από το χρωμόσωμα λαμβάνεται ο κανόνας 2 που θα μετατρέψει το πρώτο  $\langle \text{expr} \rangle$  σε x. Στην συνέχεια με την ίδια διαδικασία το δεύτερο  $\langle \text{expr} \rangle$  θα γίνει  $\langle \text{expr} \rangle - \langle \text{expr} \rangle$  και τελικά y-y.

### 3.2.3 Γενετικοί τελεστές

Η διασταύρωση και η μετάλλαξη στην περίπτωση του Grammatical Evolution γίνεται με την χρήση των απλών τελεστών διασταυρώσεως και μεταλλάξεως που παρουσιάστηκαν στην περίπτωση των γενετικών αλγορίθμων, αφού ουσιαστικά τα χρωμοσώματα είναι ακέραιοι αριθμοί και δεν απαιτείται κάποια ειδική μεταχείριση πέραν αυτής που εμφανίζεται στους γενετικούς αλγόριθμους [23], [26],[27],[28],[29]. Ένα παράδειγμα από την εφαρμογή της διαδικασίας της διασταυρώσεως ενός σημείου σε grammatical evolution παρουσιάζεται στο σχήμα 28. Όμοια η διαδικασία της μεταλλάξεως στο Grammatical Evolution δεν είναι τίποτα παραπάνω από την τυχαία αλλαγή κάποιου στοιχείου ενός χρωμοσώματος σε άλλη τιμή [38].



Σχήμα 28 Διασταύρωση ενός σημείου σε grammatical evolution.

### 3.3 Κατασκευαζόμενα τεχνητά νευρωνικά δίκτυα

#### 3.3.1 Εισαγωγή

Παρόλο που η αρχιτεκτονική του νευρωνικού δικτύου καθώς και ο αλγόριθμος εκπαίδευσης διαδραματίζουν τον σημαντικότερο ρόλο στην προσέγγιση και οι μικρές τροποποιήσεις σε καθένα από αυτά μπορούν να οδηγήσουν σε σημαντικές αλλαγές στην αποδοτικότητα του δικτύου. Στις περισσότερες περιπτώσεις, οι επιστήμονες πρέπει να πειραματιστούν με την αρχιτεκτονική του νευρωνικού δικτύου προσθέτοντας ή κόβοντας κόμβους υπολογισμού από το δίκτυο, προκειμένου να επιτύχουν τη μέγιστη απόδοση [42]. Παρακάτω θα δούμε μια μέθοδο που μπορεί να παράγει τεχνητά νευρωνικά δίκτυα χρησιμοποιώντας μια διαδικασία χαρτογράφησης που εφαρμόζεται από μια γραμματική που εκφράζεται σε μορφή Backus Naur. Η προτεινόμενη μέθοδος στοχεύει στην εξαγωγή της βέλτιστης αρχιτεκτονικής για ένα συγκεκριμένο πρόβλημα που συνοδεύεται από ένα βέλτιστο σύνολο βαρών [43], [48].

#### 3.3.2 Μέθοδος παραγωγής τεχνητών νευρωνικών δικτύων

Κάθε χρωμόσωμα, μέσω μιας διαδικασίας χαρτογράφησης που περιγράφεται στο [8] που εφαρμόζεται από τη γραμματική BNF του σχήματος 3.8, χαρτογραφείται σε ένα τεχνητό νευρωνικό δίκτυο με ένα κρυφό επίπεδο και ένα επίπεδο εξόδου. Η έξοδος του κατασκευασμένου νευρωνικού δικτύου είναι ένα άθροισμα διαφορετικών σιγμοειδών μονάδων και μπορεί να διαμορφωθεί ως

$$N(x, p) = \sum_{i=1}^H p(d+2)i - (d+1) \text{sig}(o_i(x) + p(d+2)i),$$

$$\text{όπου } o_i(x) = \sum_{j=1}^d p(d+2)i - (d+1) + j^x$$

και  $x \in R^d$ ,  $H = \text{nodes}/d+2$  είναι ο αριθμός των κρυφών κόμβων και το  $p$  υποδηλώνει τις παραμέτρους (βάρη) του νευρωνικού δικτύου [43], [48]. Η συνάρτηση  $\text{sig}(x)$  είναι η σιγμοειδής συνάρτηση  $\text{sig}(x) = \frac{1}{1+\exp(-x)}$ . Οι σταθεροί κόμβοι αντιπροσωπεύουν τον συνολικό αριθμό παραμέτρων.

Τα βασικά βήματα του προτεινόμενου αλγορίθμου είναι τα εξής:

- 1) Αρχικοποίηση του πληθυσμού, όπου κάθε στοιχείο του κάθε χρωμοσώματος αρχικοποιείται τυχαία στην περιοχή [0..255].

```
<S>::=<sigexpr>
<sigexpr>::=<Node>
      |<Node> <sigexpr>
<Node>::=<weight>*sig(<sum>+<bias>)
<sum>::=<weight>*<xxlist>
      |<sum>+<sum>
<xxlist>::=x1 | x2 | ... | xD
<weight>::=<number>
<bias>::=<number>
<number>::=(<digitlist>.<digitlist>)
      |(-<digitlist>.<digitlist>)
<digitlist>::=<digit>
      |<digit><digitlist>
<digit>::=0 | 1 | 2 | 3 | 4
      | 5 | 6 | 7 | 8 | 9
```

*Σχήμα 29 Προτεινόμενη γραμματική.*

- 2) Συνάρτηση αξιολόγησης. Για ένα συγκεκριμένο χρωμόσωμα  $g$ , τα βήματα για της συνάρτησης αξιολόγησης είναι τα εξής:
- Εφαρμόστε τη διαδικασία χαρτογράφησης της γραμματικής εξέλιξης, δημιουργώντας μια συνάρτηση  $f(x)$ .
  - Ορίστε  $v = 0$ .
  - Για κάθε σημείο  $(x_i, y_i)$  στο σύνολο εκπαίδευσης  $v = v + (f(x_i) - y_i)^2$ . Οι τιμές της συνάρτησης αξιολόγησης ορίζονται ως  $-v$ .
- 3) Εφαρμογή των γενετικών διαδικασιών επιλογής, διασταύρωσης και μετάλλαξης. Η διαδικασία επιλογής γίνεται με επιλογή τουρνουά με μέγεθος τουρνουά 8. Η διαδικασία επιλογής δημιουργεί ομάδα από υποψήφιους γονείς που θα χρησιμοποιηθεί στη διασταύρωση για τη δημιουργία νέων χρωμοσωμάτων, τα οποία θα αντικαταστήσουν τα χειρότερα χρωμοσώματα στον πληθυσμό [42]. Στη διαδικασία μετάλλαξης κάθε στοιχείο κάθε χρωμοσώματος μπορεί να αλλάξει με πιθανότητα  $pm \in [0, 1]$ .
- 4) Τερματισμός είτε εάν επιτευχθεί ένας μέγιστος αριθμός γενεών (τυπική τιμή 2000) είτε το καλύτερο χρωμόσωμα του πληθυσμού έχει τιμή αξιολόγησης κάτω από ένα προκαθορισμένο κατώφλι  $e$  (τυπική τιμή  $e = 10^{-7}$ ). Διαφορετικά μεταβείτε στο βήμα 2.

## **3.4 Χρήση κατασκευαζόμενων τεχνητών νευρωνικών δικτύων σε εφαρμογές**

### **3.4.1 Επίλυση διαφορικών εξισώσεων**

Με την χρήση κατασκευαζόμενων τεχνητών νευρωνικών δικτύων υλοποιήθηκε μια μέθοδος για επίλυση συνηθισμένων και μερικών διαφορικών εξισώσεων [43], [48]. Η μέθοδος δημιουργεί δοκιμαστικές λύσεις σε μορφή νευρωνικού δικτύου χρησιμοποιώντας Γραμματική Εξέλιξη. Οι δοκιμαστικές λύσεις βελτιώνονται περιοδικά χρησιμοποιώντας μια τοπική διαδικασία βελτιστοποίησης. Επίσης αυτή η μέθοδος μπορεί εύκολα να παραλληλιστεί. Τα κύρια πλεονεκτήματα της προτεινόμενης μεθόδου είναι τα εξής:

1. Ο χρήστης απαιτείται μόνο να κάνει δειγματοληψία των διαφορικών εξισώσεων στο πεδίο ορισμού για να δημιουργήσει τα αρχεία εκπαίδευσης / ελέγχου.
2. Η μέθοδος είναι αρκετά γενική για να εφαρμοστεί σε ODEs, SODEs και PDEs.
3. Η τελική λύση εκφράζεται σε κλειστή αναλυτική μορφή (μορφή νευρωνικού δικτύου) η οποία είναι μια διαφοροποιήσιμη μορφή.
4. Η τελική λύση παρέχει καλές ικανότητες γενίκευσης, ακόμη και εκτός της περιοχής της διαφορικής εξίσωσης.
5. Η προτεινόμενη μέθοδος βασίζεται σε γενετικούς αλγόριθμους, πράγμα που σημαίνει ότι μπορεί εύκολα να παραλληλιστεί.

### **3.4.2 Υπολογισμός βέλτιστων αποστάσεων σε μόρια**

Η φασματοσκοπία υπερύθρου είναι ένα ισχυρό εργαλείο για την αναγνώριση των χαρακτηριστικών ομάδων που υπάρχουν στο υπό μελέτη μόριο (McMurry, 1996). Η περιοχή IR του φάσματος περιλαμβάνει αποσυντεθειμένες πληροφορίες του εξεταζόμενου μοριακού συστήματος που επέκτειναν τη χρήση της φασματοσκοπίας IR. εκτός από τον προσδιορισμό πρωτογενούς δομής ενός μικρού μορίου, η φασματοσκοπία IR μπορεί να αποτελέσει πολύτιμο εργαλείο για τον προσδιορισμό είτε της δευτερογενούς δομής μιας πρωτεΐνης είτε

για πολλούς τρέχοντες ερευνητικούς τομείς που σχετίζονται με βιολογικά φαινόμενα (Barth, 2007). Η μέθοδος της κατασκευής των νευρωνικών δικτύων εφαρμόστηκε με μεγάλη επιτυχία στο γενικό πρόβλημα της τοποθεσίας και τον χαρακτηρισμό διακριτών ταλαντώσεων μιας ομάδας σωμάτων (άτομα) με μια ποικιλία διαφορετικών συνόλων ταχυτήτων [22]. Η μέθοδος είναι ικανή να κατασκευάσει και να εκπαιδεύσει μικρά και μεγάλα παιδιά συμπαγή νευρωνικά δίκτυα με καλές ικανότητες γενίκευσης.

## 4. Η εφαρμογή που υλοποιήθηκε

Στην συνέχεια θα αναφερθούμε στην γραφική εφαρμογή που υλοποιήθηκε με την χρήση του πακέτου Laravel. Βλέποντας αρχικά βήμα βήμα την εγκατάσταση και την υλοποίηση των σημαντικότερων τμημάτων της εφαρμογής. Έπειτα θα παρουσιαστούν παραδείγματα εφαρμογής της πλατφόρμας του NNC.

### 4.1 Το λογισμικό NNC

Το λογισμικό NNC κατασκευάζει τεχνητά νευρωνικά δίκτυα με την χρήση της Γραμματικής Εξέλιξης. Το NNC μπορεί να χρησιμοποιηθεί για κατηγοριοποίηση δεδομένων αλλά και για μάθηση συναρτήσεων από ένα ευρύ σύνολο επιστημονικών προβλημάτων όπως σε προβλήματα φυσικής, χημείας, βιολογίας, οικονομικών κτλ. Το πακέτο NNC έχει εφαρμοστεί με επιτυχία και σε άλλα προβλήματα όπως επίλυση διαφορικών εξισώσεων αλλά και εύρεση αμιδιακού δεσμού 1 σε προβλήματα από τον χώρο της χημείας. Στην ενότητα αυτή θα αναφερθούμε στο λογισμικό NNC το οποίο κατασκευάζει τεχνητά νευρωνικά δίκτυα με την χρήση της Γραμματικής Εξέλιξης. Στην συνέχεια θα αναφερθούμε στην γραφική εφαρμογή υλοποιήθηκε με την χρήση του πακέτου Laravel. Βλέποντας αρχικά βήμα βήμα την εγκατάσταση και την υλοποίηση των σημαντικότερων τμημάτων της εφαρμογής. Έπειτα θα παρουσιαστούν παραδείγματα εφαρμογής της πλατφόρμας του NNC.

### Γενική περιγραφή

Προκειμένου να κατασκευαστεί το νευρωνικό δίκτυο χρησιμοποιώντας τη γραμματική εξέλιξη, το ίδιο το δίκτυο πρέπει να εκφραστεί χρησιμοποιώντας τη γραμματική BNF. Ένα δίκτυο δύο επιπέδων μπορεί να εκφραστεί ως

$$N(\vec{x}, \vec{w}) = \sum_{i=1}^H w_{(d+2)i-(d+1)} \sigma \left( \sum_{3=l}^d I_3 w_{(d+2)i-(d+1)+3+w_{(d+2)i}} \right)$$

όπου  $H$  είναι ο συνολικός αριθμός μονάδων επεξεργασίας (κρυμμένοι κόμβοι) του νευρωνικού δικτύου και  $d$  είναι η διάσταση του αντικειμενικού προβλήματος που είναι ο αριθμός των στοιχείων του φορέα  $x$ . Επίσης, ορίζουμε ως  $w$  το διάνυσμα βάρους του νευρωνικού δικτύου. Η σιγμοειδής συνάρτηση  $\sigma(x)$  ορίζεται ως:  $\sigma(x) = 1 / (1 + \exp(-x))$

### Περιγραφή αλγορίθμου

Τα βασικά βήματα του αλγορίθμου είναι:

#### 1. Βήμα αρχικοποίησης.

1. Ορισμός  $g$  ως ο μέγιστος αριθμός γενεών και  $c$  ως ο αριθμός των χρωμοσωμάτων στον πληθυσμό.
2. Θέτουμε την τιμή  $P_s$  ως ρυθμό επιλογής και  $P_m$  ως ρυθμό μετάλλαξης.
3. Θέτουμε  $L_i$  τον αριθμό των γενεών που θα πρέπει να περάσουν πριν την εφαρμογή της τοπικής διαδικασίας αναζήτησης.
4. Θέτουμε  $L_c$  τον αριθμό των χρωμοσωμάτων που θα συμμετάσχουν στην τοπική διαδικασία αναζήτησης
5. Θέτουμε  $iter = 0$
6. Αρχικοποιούμε τα χρωμοσώματα ως τυχαίους φορείς των ακέραιων αριθμών.

#### 2. Γενετικό βήμα

##### 1. Για $i = 1, \dots, g$ do

- a) Δημιουργούμε για κάθε χρωμόσωμα ένα νευρωνικό δίκτυο  $C$  με γραμματική εξέλιξη.
- b) Υπολογίζουμε την καταλληλότητα  $f$  για κάθε χρωμόσωμα χρησιμοποιώντας τη διαδικασία που περιγράφεται στην ενότητα 3.2.
- c) Εφαρμόζουμε τις γενετικές διαδικασίες της διασταύρωσης και της μετάλλαξης: Τα χρωμοσώματα ταξινομούνται κατά φθίνουσα σειρά ανάλογα με την καταλληλότητα. Τα πρώτα  $(1 - P_s) \cdot c$  χρωμοσώματα αντιγράφονται άθικτα στις επόμενες γενιές. Τα υπόλοιπα χρωμοσώματα παράγονται χρησιμοποιώντας επιλογή τουρνουά, ενός σημείου διασταύρωσης και μετάλλαξη. Κατά τη διάρκεια της μετάλλαξης για κάθε στοιχείο σε κάθε χρωμόσωμα παράγεται ένας τυχαίος αριθμός  $r$  στην περιοχή  $[0,1]$ . Εάν  $r \leq P_m$  τότε το αντίστοιχο στοιχείο αλλάζει τυχαία.

##### 2. EndFor

### 3. Τοπικό βήμα αναζήτησης

1. Εάν  $\text{iters} \bmod L_i = 0$  τότε

- I. Επιλέγουμε τυχαία χρωμοσώματα  $L_c$  από τον γενετικό πληθυσμό και δημιουργούμε το σύνολο LS από αυτά τα χρωμοσώματα
- II. Για κάθε  $X(i)$  στο LS
  - a. Επιλέγουμε τυχαία ένα  $Y$  από τον πληθυσμό
  - b. Δημιουργούμε έναν απόγονο  $Z$  του  $X(i)$  και το  $Y$  χρησιμοποιώντας ένα σημείο διασταύρωσης.
  - c. Αν το  $f(z) < f(x(i))$  τότε  $X(i) = Z$ ,  $f(x(i)) = f(Z)$
  - d. ορίζουμε  $\text{iter} = \text{iter} + 1$ . Εάν το  $\text{iter} > \text{itermax}$  τερματίζει αλλιώς πάμε στο γενετικό βήμα.

### 4. Βήμα αξιολόγησης

(α) Δημιουργούμε ένα νευρωνικό δίκτυο για το καλύτερο χρωμόσωμα.

(β) Αξιολογούμε το νευρωνικό δίκτυο είτε σε κάποιο σύνολο δεδομένων κατηγοριοποίησης / προσαρμογής και αναφέρουμε τα αποτελέσματα

### **Εγκατάσταση**

Το λογισμικό είναι γραμμένο εξ ολοκλήρου σε ANSI C ++ χρησιμοποιώντας τη βιβλιοθήκη προγραμματισμού QT. Ως αποτέλεσμα, το λογισμικό μπορεί να εγκατασταθεί στα περισσότερα λειτουργικά συστήματα ακόμα και σε κινητές συσκευές (Android, Ios, κ.λπ.). Το πρόγραμμα είναι ελεύθερα διαθέσιμο από τη διεύθυνση <https://github.com/itsoulos/NNC> και ο χρήστης θα πρέπει να εκτελέσει τις ακόλουθες εντολές στα περισσότερα συστήματα UNIX για την μεταγλώττιση του έργου:

1. Download NNC-master.zip from the above url
2. `gunzip NNC-master.zip`
3. `cd NNC`
4. `qmake .`
5. `make`



## Σύνολο παραμέτρων γραμμής εντολών

Οι διαθέσιμες επιλογές της γραμμής εντολών του εκτελέσιμου NNC είναι:

1.-**h**. Τυπώνει μια οθόνη βοήθειας πληροφοριών και το πρόγραμμα τερματίζεται.

2.-**p filename**. Αυτή η παράμετρος καθορίζει τη διαδρομή του αντικειμενικού προβλήματος. Εάν η τιμή του τύπου παραμέτρου είναι νευρωνική, τότε αυτή η παράμετρος καθορίζει το αρχείο εκπαίδευσης για κατηγοριοποίηση ή προσαρμογή δεδομένων. Διαφορετικά, αυτή η παράμετρος καθορίζει την κοινόχρηστη βιβλιοθήκη με τη διαφορική εξίσωση προς επίλυση.

3.-**t filename**. Η παράμετρος αυτή filename (όνομα αρχείου) καθορίζει το αρχείο ελέγχου για το νευρωνικό δίκτυο. Αυτή η παράμετρος είναι έγκυρη μόνο εάν το είδος έχει την τιμή νευρικής.

4.-**c count**. Ο αριθμός ακέραιων αριθμών προσδιορίζει την ποσότητα των χρωμοσωμάτων που χρησιμοποιούνται στον γενετικό πληθυσμό. Η προεπιλεγμένη τιμή για αυτήν την παράμετρο είναι 500.

5.-**l length**. Αυτή η παράμετρος ακεραίου προσδιορίζει το μέγεθος κάθε χρωμοσώματος. Η προεπιλεγμένη τιμή είναι 200.

6.-**s rate**. Αυτή η παράμετρος ακεραίου καθορίζει τον ρυθμό επιλογής που χρησιμοποιείται στον γενετικό αλγόριθμο. Η προεπιλεγμένη τιμή είναι 0,1 (10%)

7.-**m rate**. Αυτή η παράμετρος ακεραίου αντιπροσωπεύει τον ρυθμό μετάλλαξης στον γενετικό αλγόριθμο. Η προεπιλεγμένη τιμή είναι 0,05 (5%)

8.-**r seed**. Αυτή η παράμετρος ακεραίου προσδιορίζει τον αριθμό που χρησιμοποιείται ως τυχαίος σπόρος σε τυχαίες γεννήτριες. Η προεπιλεγμένη τιμή είναι 1

9.-**k kind**. Αυτή η παράμετρος συμβολοσειράς καθορίζει τη χρήση του κατασκευασμένου νευρωνικού δικτύου. Οι αποδεκτές τιμές είναι

(α) neural, για προβλήματα προσαρμογής ή κατηγοριοποίησης.

(β) ode, για την επίλυση συνήθων διαφορικών εξισώσεων.

(c) pde, για την επίλυση ελλειπτικών μερικών διαφορικών εξισώσεων.

(d) sode, για την επίλυση συστημάτων συνήθων διαφορικών εξισώσεων.

**10.-o filename.** Αυτή η παράμετρος συμβολοσειράς ορίζει το αρχείο εξόδου της εκτέλεσης. Η μορφή του αρχείου εξόδου εξαρτάται από το είδος της παραμέτρου.

**11.-n gens.** Η γενική παράμετρος gens προσδιορίζει τον μέγιστο επιτρεπόμενο αριθμό γενεών. Η προεπιλεγμένη τιμή είναι 500.

**12.-g gens.** Ο γενικός παράμετρος gens καθορίζει το ποσό των γενεών που θα εκτελεστούν πριν από το στάδιο τοπικής αναζήτησης του γενετικού αλγορίθμου. Η προεπιλεγμένη τιμή για αυτήν την παράμετρο είναι 50.

**13.-d count.** Ο αριθμός ακέραιων παραμέτρων καθορίζει την ποσότητα των χρωμοσωμάτων που θα λάβουν μέρος στο στάδιο τοπικής αναζήτησης του γενετικού αλγορίθμου. Η προεπιλεγμένη τιμή για αυτήν την παράμετρο είναι 20.

### Classification or Regression file

Τα προβλήματα κατηγοριοποίησης και προσαρμογής θα πρέπει να μορφοποιηθούν σε αρχεία σύμφωνα με τη μορφή εμφάνισης στο παρακάτω σχήμα. Ο ακέραιος αριθμός  $D$  υποδηλώνει τον αριθμό των χαρακτηριστικών του συνόλου δεδομένων και το  $M$  αντιπροσωπεύει τον αριθμό των μοτίβων. Σε κάθε επόμενη γραμμή του αρχείου θα πρέπει να το πρότυπο εισόδου και η τελική στήλη είναι η πραγματική έξοδος (κατηγορία) για το αντίστοιχο μοτίβο. Ο χρήστης μπορεί να τρέξει ένα παράδειγμα σαν αυτό χρησιμοποιώντας τη γραμμή εντολών `NNC -p train -t test` όπου `train file` είναι το αρχείο εκπαίδευσης για το NNC και το `test file` είναι το αρχείο ελέγχου όπου το νευρωνικό δίκτυο θα εφαρμοστεί στο τέλος.

$D$					
$M$					
	$x_{11}$	$x_{12}$	...	$x_{1D}$	$y_1$
	$x_{21}$	$x_{22}$	...	$x_{2D}$	$y_2$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$x_{M1}$	$x_{M2}$	...	$x_{MD}$	$y_M$

## 4.2 Εγκατάσταση και υλοποίηση της εφαρμογής

Η πλατφόρμα, επιτρέπει στους χρήστες να επιλύσουν το πρόβλημά τους, δεδομένου ότι αναφέρονται οι κατάλληλες διαστάσεις και πως ο χρήστης είναι σε θέση να επιλύσει μια εξίσωση και ένα γράφημα της εκπαίδευσης των δικτύων. Η πλατφόρμα έχει σχεδιαστεί έτσι ώστε να παρέχει στον χρήστη την καταλληλότερη λύση, χρησιμοποιώντας τον αλγόριθμο NNC. Το σύστημα παίρνει τα αρχεία και τα ανεβάζει σε έναν απομακρυσμένο server που έχει διαμορφωθεί για να τρέξει το NNC.

Το σύστημα παίρνει ένα αρχείο εκπαίδευσης και ένα αρχείο ελέγχου ως μέσο. Τα αρχεία εισόδου επικυρώνονται από την πλευρά του χρήστη με τη χρήση dropzone και jQuery βιβλιοθηκών. Τα αρχεία συγκρίνονται ως προς την διάσταση του προβλήματος, αν έχουν την ίδια διάσταση προβλήματος τα αρχεία θα γίνονται αποδεκτά και αποθηκεύονται προσωρινά στον εξυπηρετητή μαζί με τις συγκεκριμένες διαδρομές τους. Τα αρχεία γίνονται αποδεκτά με τις ακόλουθες μορφές: "json, text, csv". Η πρώτη γραμμή κάθε αρχείου πρέπει να περιέχει την διάσταση του προβλήματος, και τα δύο αρχεία πρέπει να έχουν την ίδια διάσταση προβλήματος. Π.χ. αν το test.json έχει διάσταση προβλήματος 45 το train.text πρέπει επίσης να έχει διάσταση προβλήματος 45. Τα αρχεία δεν χρειάζεται να είναι του ίδιου τύπου, αλλά να έχουν την ίδια διάσταση προβλήματος.

Στην συνέχεια, τα αρχεία αποθηκεύονται μαζί με προκαθορισμένες τιμές στις γενιές, τα χρωμοσώματα, τον ρυθμό επιλογής, το ρυθμό μετάλλαξης και το μήκος του χρωμοσώματος. Στη συνέχεια ο χρήστης μπορεί να εκτελέσει τον αλγόριθμο. Καθώς το τμήμα αυτό απαιτεί πλέον υπολογιστική ισχύ, ο χρήστης μπορεί να τρέξει μόνο μία φορά τον αλγόριθμο κάθε φορά. Ο ελεγκτής που είναι υπεύθυνος για την εκτέλεση του NNC εξασφαλίζει πως κάθε χρήστης μπορεί να τρέξει μόνο μία φορά τον αλγόριθμο κάθε φορά. Μπορούμε επίσης να περιορίσουμε και να βάλουμε σε αναμονή τις αιτήσεις εκτέλεσης, αν χρειαστεί, για να εξοικονομήσουμε υπολογιστική ισχύ του διακομιστή. Ο διακομιστής αποθηκεύει και στέλνει δεδομένα πραγματικού χρόνου στον χρήστη. Ωστόσο, μπορεί να παρατηρηθεί μικρή καθυστέρηση μεταξύ της ανάκτηση δεδομένων από το διακομιστή και την εκτέλεσή τους. Αυτή η καθυστέρηση προστίθεται με σκοπό να αποφευχθεί η έλλειψη οποιασδήποτε γενιάς, διότι τόσο ο χρόνος του διακομιστή όσο και ο χρόνος στον υπολογιστή του χρήστη, μπορεί να διαφέρουν ανάλογα με τον υπολογιστή στον οποίο τρέχει το λογισμικό, τα αρχεία ελέγχου και τα αρχεία εκπαίδευσης, την εκτέλεση του αλγορίθμου από τον διακομιστή,

επομένως για να ξεπεραστεί αυτό το πρόβλημα προστέθηκε μια μικρή καθυστέρηση στην ανταπόκριση του αλγορίθμου.

Μόλις ο αλγόριθμος ξεκινήσει να εκτελείται, όλοι οι σύνδεσμοι και τα κουμπιά απενεργοποιούνται για να αποφευχθεί οποιοδήποτε πρόβλημα κατά την εκτέλεση του NNC. Ωστόσο, ο χρήστης μπορεί να σταματήσει τον αλγόριθμο οποιαδήποτε στιγμή θελήσει και χρειαστεί, με την χρήση του κουμπιού "τερματισμός".

Μόλις ολοκληρωθεί η εκτέλεση, το γράφημα και ο πίνακας θα λάβουν μια τιμή «t» αντί για γενιά, η οποία θα ενεργοποιήσει το σύστημα να σταματήσει να στέλνει αιτήματα για περισσότερες λύσεις, να συμπληρώσει το σφάλμα εκπαίδευσης, το σφάλμα ελέγχου, το σφάλμα κατηγοριοποίησης, τη λύση και επίσης να αποθηκεύσει τον παραγόμενο πίνακα έτσι ώστε ο χρήστης να μπορεί αργότερα να το δει ή να κάνει λήψη αν χρειαστεί. Μια επιλογή λήψης είναι επίσης διαθέσιμη για το χρήστη για να κατεβάσει τον πίνακα ανά πάσα στιγμή, ακόμα και κατά τη διάρκεια την οποία η παραγωγή εκτελείται.

Ένα πλήρες ιστορικό εκτελέσεων από ένα χρήστη αποθηκεύεται παράλληλα με τις γενεές, τα χρωμοσώματα, τον ρυθμό επιλογής, τον ρυθμό μετάλλαξης και το μήκος του χρωμοσώματος, ωστόσο, αυτές οι τιμές δεν χρησιμεύουν και δεν εμφανίζονται στον χρήστη. Εάν ο διακομιστής επεκταθεί για εμπορικούς σκοπούς, οι τιμές εκτέλεσης, οι δοκιμαστικές και οι τρέχουσες τιμές μπορούν επίσης να αποθηκευτούν και να χρησιμοποιηθούν αργότερα, αυτή τη στιγμή δεν είναι εφικτό να γίνει κάτι τέτοιο σε διακομιστές Linux μικρής κλίμακας.

Το σύστημα προστατεύεται επίσης με χρήση επαλήθευσης ηλεκτρονικού ταχυδρομείου. Που σημαίνει ότι το λογισμικό μπορεί να το χρησιμοποιήσει ένας χρήστης, εφόσον κάνει πρώτα έναν λογαριασμό και έπειτα ο διαχειριστής του δώσει πρόσβαση στο σύστημα. Ο χρήστης πρέπει να δημιουργήσει έναν λογαριασμό χρησιμοποιώντας έναν ορθό λογαριασμό ηλεκτρονικού ταχυδρομείου, στη συνέχεια αποστέλλεται μήνυμα επαλήθευσης μέσω του ηλεκτρονικού ταχυδρομείου, έτσι ώστε να γίνει επικύρωση του χρήστη από τον διαχειριστή. Επίσης ο χρήστης μπορεί να κάνει ανάκτηση κωδικού πρόσβασης σε περίπτωση που ξεχάσει τον κωδικό πρόσβασής του. Το ηλεκτρονικό ταχυδρομείο χρησιμοποιεί αυτήν τη στιγμή το διακομιστή smtp mailtrap για την αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου. Αυτό μπορεί να αλλάξει σε ένα ενεργό διακομιστή εργασίας στο μέλλον.

Το σύστημα είναι κατασκευασμένο με το framework Laravel. Το πρώτο βήμα είναι να συνδεθεί η βάση δεδομένων με το framework. Ο Laravel δίνει ένα αρχείο .env που

αποθηκεύει όλες τις ρυθμίσεις που απαιτούνται για ένα έργο. Αφού δημιουργήσουμε μια βάση δεδομένων χρησιμοποιώντας την ακόλουθη εντολή.

- `mysql -u username -p password`
- `create database genetic_neural_network;`
- `exit;`

Το δεύτερο βήμα ήταν να δημιουργήσουμε ένα σύστημα ελέγχου ταυτότητας. Το Laravel διαθέτει ένα βασικό σύστημα ασφαλείας που παρέχει απλή εγγραφή και σύνδεση. Αυτό μπορεί να ενεργοποιηθεί χρησιμοποιώντας την ακόλουθη εντολή.

- `'php artisan make:auth'`

Αυτό δημιουργεί μια κλάση / μοντέλο χρήστη που έχει επεκταθεί με το μοντέλο Eloquent, αρχεία βάσης δεδομένων για το χρήστη και κωδικό πρόσβασης, ελεγκτή ελέγχου ταυτότητας και αρχεία προβολής auth στον κατάλογο προβολών / auth. Στη συνέχεια, η βάση δεδομένων τροποποιείται ανάλογα με τις ανάγκες του έργου, στην περίπτωση μας έπρεπε να προσθέσουμε μια ένδειξη επαλήθευσης, αλλά αντί να αλλάξουμε την προεπιλεγμένη λειτουργικότητα, δημιουργήσαμε έναν διαφορετικό ελεγκτή και κλάση για την επαλήθευση - `verifyUser` χρησιμοποιώντας την ακόλουθη εντολή.

- `php artisan make:model verifyuser -cm`
- `-cm` are used to generate a controller and a model file.

Στη συνέχεια, δημιουργήσαμε μια σχέση επαλήθευσης του χρήστη με την κλάση του χρήστη και στη συνέχεια μεταφέραμε την βάση δεδομένων χρησιμοποιώντας την ακόλουθη εντολή

- `php artisan migrate`

Αυτό έδωσε ένα βασικό σημείο εκκίνησης, δουλέυαμε σε τοπικό διακομιστή εκείνη την χρονική στιγμή και δεν μπορούσαμε να στείλουμε μήνυμα ηλεκτρονικού ταχυδρομείου για επαλήθευση. Δημιουργήσαμε λοιπόν έναν λογαριασμό `mailtrap.io` και χρησιμοποιήσαμε μια υπηρεσία ταχυδρομείου δοκιμής για να στείλουμε e-mail επαλήθευσης. Μετά από αρκετές δοκιμές, τελικά τελειοποιήσαμε την επαλήθευση μέσω ηλεκτρονικού ταχυδρομείου. Αυτό σήμαινε επίσης ότι έπρεπε να αλλάξουμε τις προεπιλεγμένες λειτουργίες ελεγκτή auth για να το προσαρμόσουμε στις ανάγκες μας.

Το επόμενο βήμα ήταν να δημιουργηθεί μια κλάση και ένας ελεγκτής για τις εκτελέσεις του `nnc`. Αρχικά ξεκινήσαμε με πολύ λίγες οντότητες στον πίνακα, οι τιμές του `nnc` αποθηκεύονται σε έναν πίνακα και στη συνέχεια όταν ζητείται η λειτουργία αλγόριθμου εκτέλεσης, η συνάρτηση παίρνει εισόδους από το αντικείμενο `nnc`. Το `id` του αντικειμένου

npc που είναι αποθηκευμένο στη βάση δεδομένων μεταβιβάζεται στη σελίδα γραφικών παραστάσεων, το οποίο χρησιμοποιείται για την εκτέλεση του σωστού αντικειμένου. Τα αρχεία για το npc αποθηκεύονται σε ένα φάκελο, ο φάκελος είναι μοναδικός για κάθε χρήστη βάση του αναγνωριστικού ID τους.

Το μοντέλο, η κλάση και ο ελεγκτής δημιουργήθηκαν με τη χρήση του "rhp artisan make: model NNC -cm". Αργότερα, για να βελτιώσουμε το έργο μας, προσθέσαμε περισσότερα χαρακτηριστικά στην κλάση και ανανεώσαμε τις μεταφορές μας.

Είχαμε δημιουργήσει ήδη το απαιτούμενο front end γραφικό περιβάλλον χρησιμοποιώντας bootstrap, charts.js και datatables.js, πράγμα που μας διευκόλυνε να καταλάβουμε τη ροή του έργου και να βελτιώσουμε την αποδοτικότητα της χρήσης του.

Το επόμενο πρόβλημα που συναντήσαμε ήταν να το να εκτελέσουμε τον αλγόριθμο ασύγχρονα και να δείξουμε αποτελέσματα στον χρήστη. Εάν επιστρέφαμε το αποτέλεσμα απευθείας, ο αλγόριθμος θα κολλούσε για λίγο, πριν εκτελέσει τελείως τα αποτελέσματα, η πλευρά του χρήστη θα πάγωνε και το σύστημα θα κολλούσε ολοκληρωτικά. Ως εκ τούτου, έπρεπε να κάνουμε έναν νέο πίνακα που θα αποθηκεύει ενδιάμεσα το αποτέλεσμα του αλγορίθμου, έπρεπε να αλλάξουμε το γραφικό περιβάλλον των γραφημάτων και των πινάκων του προγράμματος jQuery σε vue.js το οποίο ήταν πιο εφικτό να χρησιμοποιηθεί όταν γίνεται χειρισμός και αλλαγή των δεδομένων σε πραγματικό χρόνο. Ο πίνακας λύσεων θα αποθηκεύει όλες τις τιμές που δημιουργήθηκαν από τον αλγόριθμο, ενώ στην πλευρά του χρήστη το πρόγραμμα περιήγησης θα συνεχίσει να υποβάλλει αιτήματα για νέα δεδομένα από πίνακες λύσεων για τον αλγόριθμο που εκτελούσε αυτή τη στιγμή. Αυτό έγινε εφικτό με τη χρήση της συνάρτησης rhp proc\_open (), αυτή η λειτουργία μας επέτρεψε να λαμβάνουμε δεδομένα από τον αλγόριθμο εκτέλεσης σε πραγματικό χρόνο. Στη συνέχεια, έπρεπε να διατηρήσουμε μια στατική σειρά ελεγκτών που αποθήκευαν τη διαδικασία και την ταυτότητα χρήστη. Αν ο χρήστης είχε ήδη τρέξει μία διεργασία, δεν θα του επέτρεπε να τρέξει περισσότερες. Αυτή η προσέγγιση μπορεί επίσης να χρησιμοποιηθεί αργότερα κάνοντας μια στατική μεταβλητή στην κλάση npc και αποθηκεύοντας στατικά το id του χρήστη, το npc id και το id της διεργασίας, αυτή η σειρά μπορεί στη συνέχεια να χρησιμοποιηθεί για να περιορίσει τον αριθμό των διεργασιών που εκτελούν τον αλγόριθμο στο διακομιστή, έτσι ώστε να βελτιώσουν τις επιδόσεις και άλλους σκοπούς, στην περίπτωση που το έργο χρησιμοποιηθεί εμπορικά.

```

1. //Διεργασίες που δηλώνονται στο BaseController
2.
3. public function __construct() {
4.     $this->middleware( 'auth' );
5.     if ( ! isset( self::$processes ) ) {
6.         self::$processes = array();
7.     }
8. }

```

*Σχήμα 30 Διεργασίες που δηλώνονται στο BaseController.*

```

1. // Δημιουργία ενός object για να χρησιμοποιήσει καθόλη την διάρκεια της
2. εκτέλεσης το NNC
3. public function create( Request $request ) {
4.     $nnc = new NNC();
5.     $nnc->user_id = Auth::user()->id;
6.     $nnc->chromosomes = $request->chromosomes;
7.     $nnc->generation = $request->generations;
8.     $nnc->selection = $request->selection;
9.     $nnc->mutation = $request->mutation;
10.    $nnc->genome_length = $request->genome_length;
11.    $nnc->save();
12.    $test_file = $this->test_file = $request->test_file;
13.    $train_file = $this->train_file = $request->train_file;
14.    $this->current_nnc = $nnc;
15.    $generations = $nnc->generations;
16.
17.    return view( 'pages.graphs', compact( 'nnc', 'test_file', 'train_file' ) );
18. }

```

*Σχήμα 31 Δημιουργία ενός object για να χρησιμοποιήσει καθόλη την διάρκεια της εκτέλεσης το NNC.*

Μόλις ολοκληρωθεί η εκτέλεση, η λειτουργία θα αποθηκεύσει το σφάλμα εκπαίδευσης, το σφάλμα ελέγχου, το σφάλμα κλάσης και τη λύση του νευρωνικού δικτύου μαζί με ένα γράφημα στη βάση δεδομένων και την εικόνα του γραφήματος στο διακομιστή σε έναν κατάλογο ο οποίος είναι συγκεκριμένος για κάθε χρήστη. Αυτό σήμαινε επίσης ότι ο πίνακας λύσεων (ο οποίος αποθηκεύει δεδομένα για να γίνεται επικοινωνία μεταξύ της εκτέλεσης σε πραγματικό χρόνο και του προγράμματος περιήγησης του χρήστη) θα

συμπληρωθεί με δεδομένα. Αυτό το πρόβλημα, λύνεται με την εκτέλεση μιας εργασίας μετά από κάθε ώρα η οποία διαγράφει όλα τα αρχεία που είναι παλαιότερα από 1 ώρα.

```
1. /*
2.     * Εκτέλεση του αλγορίθμου
3.     * */
4.     public function run( Request $request ) {
5.         if ( $this->check_process() ) {
6.
7.             //Λήψη δεδομένων
8.             $nnc          = NNC::where( 'id', '=', $request->nnc_id )-
>first();
9.             $test_file    = $request->test_file;
10.            $train_file   = $request->train_file;
11.            $chromosomes  = $nnc->chromosomes;
12.            $generations  = $nnc->generation;
13.            $selection_rate = $nnc->selection;
14.            $mutation_rate = $nnc->mutation;
15.            $genome_length = $nnc->genome_length;
16.            $command      = ( "./NNC -p $train_file -t $test_file -
n $generations -s $selection_rate -m $mutation_rate -l $genome_length -
d $chromosomes" );
17.            $cmd          = $command;
```

*Σχήμα 32 Εκτέλεση του αλγορίθμου.*



```

1. //      dd( $cmd ); // ελέγχει την σύνταξη του αλγορίθμου
2.          $descriptorspec = array(
3.              0 => array( "pipe", "r" ), // το stdin είναι pipe από το οποίο
το child διαβάζει δεδομένα
4.              1 => array( "pipe", "w" ), // το stdout είναι pipe στο οποίο το
child θα γράφει δεδομένα
5.              2 => array( "pipe", "w" ) // το stderr είναι pipe στο οποίο το
child θα γράφει δεδομένα
6.          ); //χωρίζει την διεργασία σε pipes
7.          flush(); // αδειάζει αχρείαστες τιμές έτσι ώστε να εκτελείται πιο γρή
γορα η διεργασία
8.          $process = proc_open( $cmd, $descriptorspec, $pipes, realpath( './' )
, array() ); //εκτέλεση της εντολής
9.
10. /* Διατήρηση μιας λίστας διεργασιών όλων των χρηστών και αποθήκευση των
11.   δεδομένων σε μια στατική μεταβλητή με το όνομα $processes*/
12.          $status = proc_get_status( $process );
13.          $this->process_list( [
14.              'user_id'    => Auth::user()->id,
15.              'process_id' => $status['pid'],
16.              'process'    => $process
17.          ] );
18. /* τα threads εκτελούνται, μετατρέπονται τα δεδομένα και αποθηκεύονται οι
19.   λύσεις στο πίνακα λύσεων για επεξεργασία σε πραγματικό χρόνο*/
20.
21.          if ( is_resource( $process ) ) {
22.              while ( $s = fgets( $pipes[1] ) ) {
23.                  $s = substr( $s, 0, strpos( $s, "\n" ) );
24.                  $obj = explode( "\t", $s );
25.                  if ( sizeof( $obj ) > 1 ) {
26.
27.                      $sol = new solution();
28.                      $sol->generation = $obj[0];
29.                      $sol->minimum = - 1 * $obj[1];
30.                      $sol->neural_network = $obj[2];
31.                      $sol->user_id = Auth::user()->id;
32.                      $sol->nnc_id = $nnc->id;
33.
34.                      $sol->save();
35.                  } else {
36.                      if ( str_contains( $obj[0], 'TRAIN ' ) ) {
37.                          $obj = explode( "=", $obj[0] );
38.                          $nnc->train_error = $obj[1];
39.                          $nnc->save();

```

```

40.         } else if ( str_contains( $obj[0], 'TEST ' ) ) {
41.             $obj          = explode( "=", $obj[0] );
42.             $nnc->test_error = $obj[1];
43.             $nnc->save();
44.         } else if ( str_contains( $obj[0], 'CLASS ' ) ) {
45.             $obj          = explode( "=", $obj[0] );
46.             $nnc->class_error = $obj[1];
47.             $nnc->save();
48.         } else if ( str_contains( $obj[0], 'SOLUTION' ) ) {
49.             $obj          = explode( "=", $obj[0] );
50.             $nnc->solution = $obj[1];
51.             $nnc->save();
52.         }
53.     }
54.     flush();
55. }
56. }
57. }
58. }

```

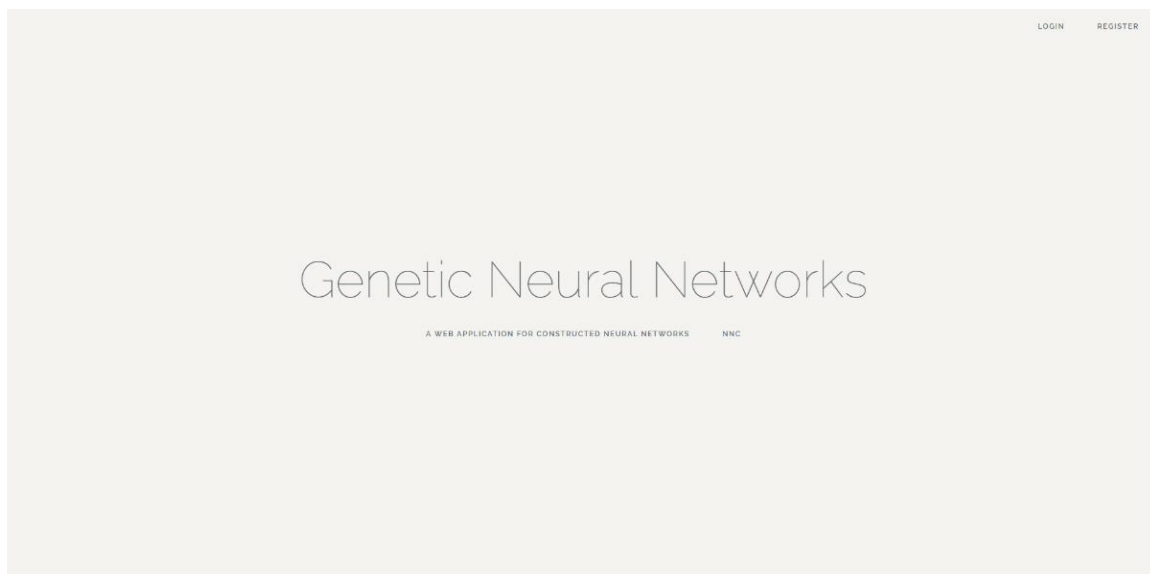
*Σχήμα 33 Διεργασίες του αλγορίθμου.*

Επίσης έπρεπε να φιλτράρουμε και να ταιριάζουμε τα αποτελέσματα του αλγορίθμου για να τα αποθηκεύσουμε στη σωστή βάση δεδομένων. Το άμεσο αποτέλεσμα του τρέχοντος αλγόριθμου εμφανίζεται στη σελίδα γραφημάτων ενώ τα προηγούμενα αρχεία καταγραφής εμφανίζονται στη σελίδα προφίλ.

### 4.3 Παραδείγματα χρήσεως της εφαρμογής

Παρακάτω μπορούμε να δούμε παραδείγματα χρήσεως της εφαρμογής. Στην online εφαρμογή για την χρήση του αλγορίθμου NNC μπορεί κάποιος να εισέλθει με το παρακάτω url: <http://itsoulos.teiep.gr/nnc/public/>

Η πρώτη αρχική σελίδα και σελίδα καλωσορίσματος φαίνεται στην εικόνα 3 παρακάτω.

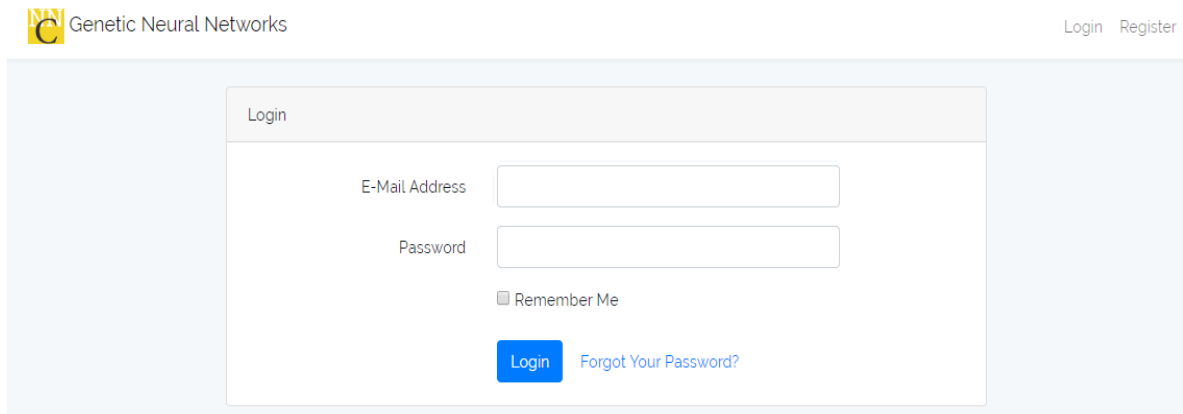


*Εικόνα 3 Σελίδα καλωσορίσματος στην πλατφόρμα NNC.*

Στην αρχική οθόνη καλωσορίσματος μπορούμε να πατήσουμε πάνω δεξιά Register και να κάνουμε λογαριασμό στην πλατφόρμα του NNC εφόσον δεν έχουμε ακόμα κάποιο δημιουργημένο λογαριασμό όπως φαίνεται και στην εικόνα 4.

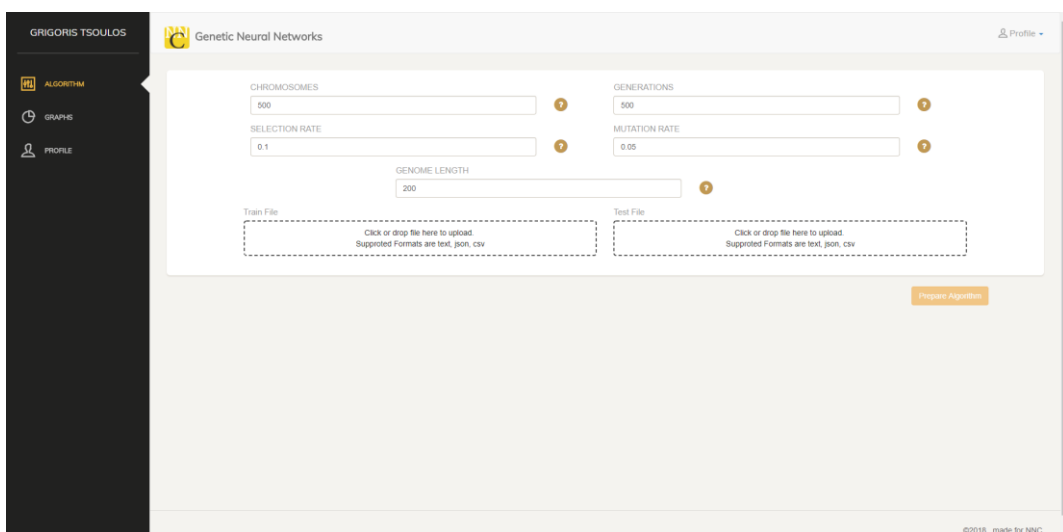
*Εικόνα 4 Σελίδα register στην πλατφόρμα NNC.*

Μόλις έχουμε δημιουργήσει ένα λογαριασμό, μπορούμε να κάνουμε Login στην πλατφόρμα του NNC Algorithm και να τρέξουμε τον αλγόριθμο όπως φαίνεται και στην εικόνα 5. Για κάθε νέο λογαριασμό που δημιουργείται ο νέος χρήστης για κάνει την πρώτη είσοδο στο online system του NNC θα πρέπει να του έχει δώσει πρόσβαση ο διαχειριστής του συστήματος.



*Εικόνα 5 Σελίδα login στην πλατφόρμα NNC.*

Με την είσοδο μας στο σύστημα αυτά που μπορούμε να διακρίνουμε το όνομα χρήστη πάνω αριστερά, που έχει δώσει ο χρήστης κατά την εγγραφή του. Στην αριστερή πλευρά βλέπουμε το μενού το οποίο και είναι: ALGORITHM, GRAPHS, PROFILE. Πάνω δεξιά στην οθόνη μπορούμε να δούμε το εικονίδιο profile, το οποίο πατώντας το θα εμφανιστεί ένα αναδυόμενο μενού όπου μπορούμε είτε να μπούμε στην σελίδα του profile του χρήστη είτε να κάνουμε logout από την εφαρμογή. Τα παραπάνω φαίνονται στην εικόνα 6 που ακολουθεί.



*Εικόνα 6 Σελίδα Algorithm στην πλατφόρμα NNC*

Η σελίδα algorithm η οποία και φαίνεται στην εικόνα 7, είναι και η πρώτη σελίδα που βλέπουμε όταν κάνουμε login στο σύστημα. Στην σελίδα αυτή δίνουμε τις παραμέτρους και τα αρχεία test και train για να τρέξουμε τον αλγόριθμο. Στην σελίδα αυτή δηλαδή θα δώσουμε:

Chromosomes

Generations

Selection Rate

Mutation Rate

Genome Length

Train file

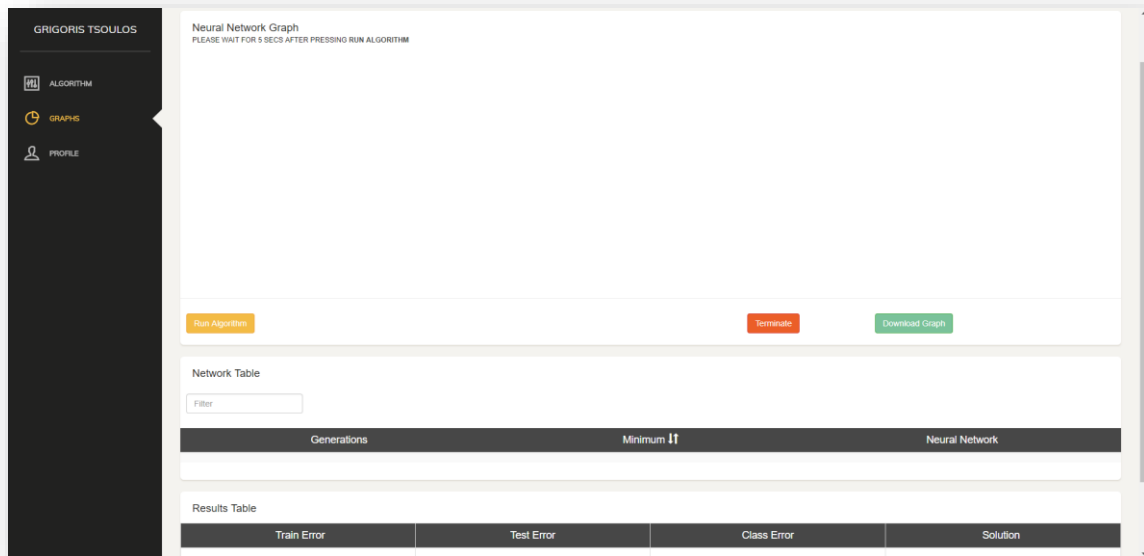
Test file

Οι παράμετροι αυτοί έχουν προκαθορισμένες τιμές τις οποίες και μπορούμε να αλλάξουμε εφόσον το επιθυμούμε. Επίσης τα αρχεία test και train θα πρέπει για να γίνουν δεκτά από το σύστημα να είναι της μορφής text,json,csv και η διάσταση του προβλήματος να είναι ίδια αν κάτι από αυτά δεν είναι σωστά τότε ο αλγόριθμος δεν δίνει την έγκριση για να τρέξει.

The image shows a web interface for configuring an algorithm. It features several input fields and two file upload areas. The fields are: CHROMOSOMES (dropdown menu with value 500), GENERATIONS (text input with value 500), SELECTION RATE (text input with value 0,1), MUTATION RATE (text input with value 0,05), and GENOME LENGTH (text input with value 200). Each field has a question mark icon to its right. Below these are two dashed boxes for 'Train File' and 'Test File', each containing the text 'Click or drop file here to upload. Supported Formats are text, json, csv'. At the bottom right, there is an orange button labeled 'Prepare Algorithm'.

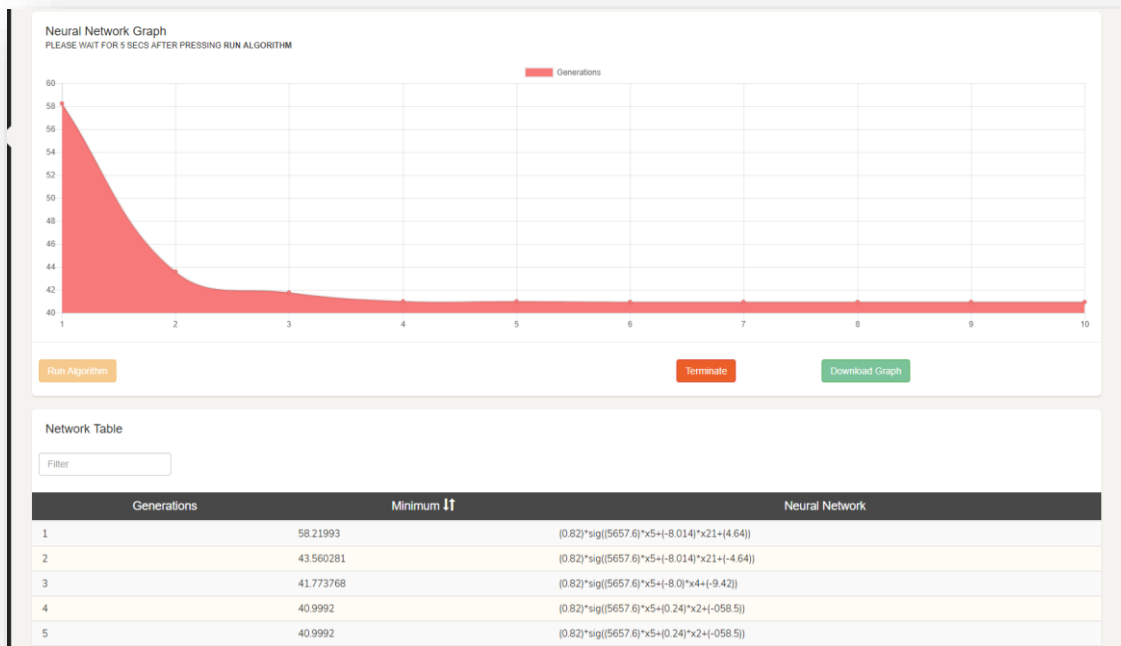
*Εικόνα 7 Παράμετροι του NNC.*

Εφόσον τα στοιχεία που δώσουμε για την εφαρμογή του αλγορίθμου είναι σωστά τότε θα εμφανιστεί ένα μήνυμα ότι μπορούμε να τρέξουμε τον αλγόριθμο και θα ενεργοποιηθεί το κουμπί 'Prepare Algorithm' το οποίο μόλις πατήσουμε θα μεταφερθούμε στην σελίδα Graphs. Στην σελίδα αυτή μπορούμε να τρέξουμε τον αλγόριθμο και να δούμε τα αποτελέσματα της εκτέλεσης του. Στην σελίδα Graphs μπορούμε να τρέξουμε τον αλγόριθμο πατώντας το κουμπί 'Run Algorithm', όπως επίσης έχουμε την δυνατότητα και να τον σταματήσουμε πατώντας το κουμπί 'terminate'.



*Εικόνα 8 Σελίδα Graphs στην πλατφόρμα NNC.*

Επίσης υπάρχει το κουμπί 'Download Graph' με το οποίο μπορούμε να κατεβάσουμε το γράφημα του αλγορίθμου είτε κατά την διάρκεια που τρέχει είτε στο τέλος όταν τελειώσει. Το γράφημα το οποίο αναπαριστά τις γενιές με το minimum του αλγορίθμου. Στην σελίδα αυτή επίσης μπορούμε να δούμε κατά την διάρκεια που τρέχει ο αλγόριθμος τις γενιές, το minimum και το νευρωνικό δίκτυο που σχηματίζεται. Στο τέλος του αλγορίθμου σε διαφορετικό πίνακα θα δούμε τα τελικά αποτελέσματα του. Όπως φαίνονται στις εικόνες 8 και 9.



Εικόνα 9 Εκτέλεση αλγορίθμου στην σελίδα Graphs.

Τέλος στην σελίδα Profile την οποία βλέπουμε στην εικόνα 10, μπορούμε να κάνουμε αλλαγή του κωδικού μας για την είσοδο στην πλατφόρμα. Το πιο σημαντικό όμως είναι ότι μπορούμε να δούμε τα αποτελέσματα από κάθε εκτέλεση που έχουμε κάνει με τον NNC και να κατεβάσουμε το γράφημα από κάθε μία από τις εκτελέσεις αυτές καθώς και να κατεβάσουμε το txt αρχείο με τα αποτελέσματα εξόδου του νευρωνικού δικτύου και τις εισόδους που είχαμε δώσει για να τρέξει.

ID	Train Error	Test Error	Class Error	Solution	Graph	Text	Date
1	22.3160787216	2.8967226971	11.42%	$(-22.71645) \cdot \text{sig}((0.00995) \cdot x_5 + (91.16327) \cdot x_{27} + (-100.00287)) + (0.85470) \cdot \text{sig}((10.17988) \cdot x_1 + (12.84201) \cdot x_{54} + (-12.8))$	Graph	text	2 months ago
2	22.3160785443	2.8966886209	11.42%	$(-38.11952) \cdot \text{sig}((2.14251) \cdot x_5 + (93.74889) \cdot x_{27} + (-100.00005)) + (0.86419) \cdot \text{sig}((9.28627) \cdot x_1 + (10.70562) \cdot x_{54} + (-10.3182))$	Graph	text	2 months ago
3	22.3160787216	2.8967226971	11.42%	$(-22.71645) \cdot \text{sig}((0.00995) \cdot x_5 + (91.16327) \cdot x_{27} + (-100.00287)) + (0.85470) \cdot \text{sig}((10.17988) \cdot x_1 + (12.84201) \cdot x_{54} + (-12.8))$	Graph	text	2 months ago
4	22.2640608810	2.9037338591	11.42%	$(-38.11952) \cdot \text{sig}((2.14251) \cdot x_5 + (93.74889) \cdot x_{27} + (-100.00005)) + (0.86419) \cdot \text{sig}((9.28627) \cdot x_1 + (10.70562) \cdot x_{54} + (-10.3182))$	Graph	text	2 months ago
5	22.0668812587	3.0049622585	11.42%	$(-38.79838) \cdot \text{sig}((2.02954) \cdot x_5 + (92.84817) \cdot x_{27} + (-100.00004)) + (0.85654) \cdot \text{sig}((7.46290) \cdot x_1 + (9.90215) \cdot x_{54} + (0.81992) \cdot x_{29} + (-8.52890))$	Graph	text	2 months ago
6	22.0668812587	3.0049622585	11.42%	$(-38.79838) \cdot \text{sig}((2.02954) \cdot x_5 + (92.84817) \cdot x_{27} + (-100.00004)) + (0.85654) \cdot \text{sig}((7.46290) \cdot x_1 + (9.90215) \cdot x_{54} + (0.81992) \cdot x_{29} + (-8.52890))$	Graph	text	2 months ago
7	22.0668812587	3.0049622585	11.42%	$(-38.79838) \cdot \text{sig}((2.02954) \cdot x_5 + (92.84817) \cdot x_{27} + (-100.00004)) + (0.85654) \cdot \text{sig}((7.46290) \cdot x_1 + (9.90215) \cdot x_{54} + (0.81992) \cdot x_{29} + (-8.52890))$	Graph	text	2 months ago

Εικόνα 10 Σελίδα Profile στην πλατφόρμα NNC.

#### **4.4. Συμπεράσματα – Μελλοντική εργασία**

Η εφαρμογή αποτελεί ένα συνδυασμό γραφικής εφαρμογής σε HTML/ LARAVEL και του λογισμικού NNC που μπορεί να χρησιμοποιηθεί για την επίλυση προβλημάτων κατηγοριοποίησης. Ο χρήστης μέσω μιας γραφικής εφαρμογής μπορεί να εισάγει τις παραμέτρους του προβλήματος προς επίλυση και να παρακολουθήσει την πορεία της αλλά και να κατεβάσει τα τελικά αποτελέσματα στον υπολογιστή του.

Το λογισμικό μπορεί να επεκταθεί και για άλλου είδους προβλήματα όπως επίλυση διαφορικών εξισώσεων με την πιθανή προσθήκη νέας περιοχής στην γραφική εφαρμογή. Ακόμα υπάρχει η δυνατότητα βελτίωσης των παραγόμενων αποτελεσμάτων με την εφαρμογή μεθόδου τοπικής βελτιστοποίησης πάνω στο παραγόμενο τεχνητό νευρωνικό δίκτυο για την περαιτέρω βελτίωση των κόμβων επεξεργασίας. Ακόμα η μέθοδος στην παρούσα μορφή τερματίζει με ένα όριο γενιών στον γενετικό αλγόριθμο. Αυτό μπορεί να είναι μη αποδοτικό σε μεγάλα προβλήματα και για αυτό μπορούν να προστεθούν και πιο “έξυπνες” τεχνικές τερματισμού του γενετικού αλγορίθμου. Επίσης μπορεί να αναπτυχθεί και εφαρμογή βελτιστοποιημένη για κινητά τηλέφωνα και tablet για την καλύτερη διευκόλυνση των χρηστών που θα ασχοληθούν στο μέλλον.



## Βιβλιογραφία

- [1] D. F. Specht, «A General Regression Neural Network,» *IEEE TRANSACTIONS ON NEURAL NETWORKS*, τόμ. 2, 1991.
- [2] Z. Byoung-Tak και H. Mulhenbein, «Evolving Optimal Neural Networks Using Genetic Algorithms with Occam's Razor,» *Complex Systems*, τόμ. 7, 1993.
- [3] A.C.C. Coolen, *A Beginner's Guide to the Mathematics of Neural Networks*.
- [4] I. G. Tsoulos, D. Gavrilis και E. Glavas, «Neural Network Construction using Grammatical Evolution,» 2005.
- [5] Md. M. Rahman and T. Akter Setu, «An Implementation for Combining Neural Networks and Genetic Algorithms,» *International Journal of Ambient Systems and Applications*, τόμ. 6, αρ. 3, 2015.
- [6] Saravanan K. and S. Sasithra, «REVIEW ON CLASSIFICATION BASED ON ARTIFICIAL NEURAL NETWORKS,» *International Journal of Ambient Systems and Applications*, τόμ. 2, αρ. 4, 2014.
- [7] Ilya Sutskever O. Vinyals and Q. V. Le, *Sequence to Sequence Learning with Neural Networks*.
- [8] Fardin Ahmadizar, K. Soltanian, F. AkhlaghianTab and I. Tsoulos, *Artificial neural network development by means of a novel combination of grammatical evolution and genetic algorithm*.
- [9] S. Sootla, *Artificial neural network for image classification*, Tartu , 2015.
- [10] M. Bocco, E. Willington and Mónica Arias, «COMPARISON OF REGRESSION AND NEURAL NETWORKS MODELS TO ESTIMATE SOLAR RADIATION,» *Chilean Journal of Agricultural Research*, τόμ. 70, αρ. 3, 2010.
- [11] M. Chiarandini, *Machine Learning: Linear Regression and Neural Networks*.
- [12] G. Ou and Y. L. Murphey, «Multi-class pattern classification using neural networks,» *Science Direct*, τόμ. 40, 2007.
- [13] L. A. Gatys, A. S. Ecker and M. Bethge, *Image Style Transfer Using Convolutional Neural Networks*.
- [14] F. Noorian, A. M. de Silva and P. H.W. Leong, *Grammatical Evolution: A Tutorial using gramEvol*, 2016.
- [15] K. Gurney, *An introduction to neural networks*, London and New York, 1997.
- [16] S. Haykin, *Neural Networks and Learning Machines*, 2008.
- [17] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella and J. Schmidhuber, *Flexible, High Performance Convolutional Neural Networks for Image Classification*.
- [18] I. Sutskever, *TRAINING RECURRENT NEURAL NETWORKS*, Toronto, 2013.
- [19] J. Larsen, *Introduction to Artificial Neural Networks*, 1999.
- [20] D. E. Rumelhart, B. Widrow and M. A. Lehr, «The Basic Ideas in Neural Networks,» *Communications of the ACM*, τόμ. 37, αρ. 3, 1994.
- [21] G. Hinton, N. Srivastava and Kevin Swersky, «Neural Networks for Machine Learning».
- [22] IG.V. Papamokos, I.G. Tsoulos, I.N. Demetropoulos and E. Glavas, «Location of amide I mode of vibration in computed data utilizing constructed neural networks,» *Expert Systems with Applications*, τόμ. 36, 2009.

- [23] I. G. Tsoulos and I. E. Lagaris, «Grammar inference with grammatical evolution».
- [24] A. W. Moore, *Regression and Classification with Neural Networks*, 2001.
- [25] M. Rocha, P. Cortez and J. Neves, *Evolution of Neural Networks for Classification and Regression*, 2004.
- [26] I. Tsoulos, D. Gavrilis and E. Glavas, *Neural Network Construction and Training using Grammatical Evolution*.
- [27] I. Tsoulos, D. Gavrilis and E. Glavas, «Neural network construction and training using grammatical evolution,» *Neurocomputing*, τόμ. 72, 2008.
- [28] I. Tsoulos, D. Gavrilis and E. Glavas, «Neural Network Construction using Grammatical Evolution,» 2005.
- [29] I. Tsoulos, D. Gavrilis and E. Glavas, *Neural Network Construction using Grammatical Evolution*.
- [30] C. M. Bishop, *Neural Networks: A pattern Recognition Respective*, 1996.
- [31] C. Gershenson, *Artificial Neural Networks for Beginners*.
- [32] V. Cheung and Kevin Cannons, *An Introduction to Neural Networks*.
- [33] B. Krose and P. van der Smagt, *An introduction to Neural Networks*, 1996.
- [34] R. Rojas, *Neural Networks. A Systematic Introduction*, Berlin: Springer, 1996.
- [35] D. Kriesel, *A Brief Introduction to Neural Networks*, 2005.
- [36] A. K. Jain and J. M. K.M. Mohiuddin, *Artificial Neural Networks: A tutorial*, 1996.
- [37] M. T. Hagan, H. B. Demuth, M. H. Beale and O. De Jesús, *Neural Network Design*.
- [38] G. Georgoulas, D. Gavrilis, I. G. Tsoulos, C. Stylios, J. Bernardes and P. P. Groumpos, «Novel approach for fetal heart rate classification introducing grammatical evolution,» *Biomedical Signal Processing and Control*, τόμ. 2, 2007.
- [39] A. R. Omondi and J. C. Rajapakse, *FPGA Implementations of Neural Networks*, Netherlands: Springer, 2006.
- [40] P. Mehtani, *Pattern Classification using Artificial Neural Networks*, Orissa.
- [41] D. Reby, S. Lek, I. Dimopoulos, J. Joachim, J. Lauga and S. Aulagnier, «Artificial neural networks as a classification method in the behavioural sciences,» *Behavioural Processes*, τόμ. 40, 1997.
- [42] D. Gavrilis, I. G. Tsoulos and E. Dermatas, «Selecting and constructing features using grammatical evolution,» *Pattern Recognition Letters*, 2008.
- [43] I. G. Tsoulos, D. Gavrilis and E. Glavas, «Solving differential equations with constructed neural networks,» *Neurocomputing*, τόμ. 72, 2009.
- [44] I. G. Tsoulos and I. E. Lagaris, *Solving differential equations with genetic programming*.
- [45] IBM, *IBM SPSS Neural Networks 22*.
- [46] K. O. Stanley and R. Miikkulainen, «Evolving Neural Networks through Augmenting Topologies,» *The MIT Press Journals Evolutionary Computation*, τόμ. 10, αρ. 2.
- [47] Ι. Τσούλος, *Τμηματικά Νευρωνικά Δίκτυα για Προσαρμογή Δεδομένων*, 2001.
- [48] Ι. Τσούλος, *Καθολική Βελτιστοποίηση: Μέθοδοι, Λογισμικό και Εφαρμογές*, 2005.
- [49] I. G. Tsoulos, A. Tzallas and D. Tsalikakis, *Evolutionary based weight decaying method for neural network training*.