ADAPTIVE NEURO-FUZZY INFERENCE SYSTEMS (ANFIS) APPLIED ON MEDICAL DIAGNOSIS

A Dissertation
Presented to
The Academic Faculty

by

Nikolaos D. Mparoutis

In Partial Fulfillment
of the Requirements for the Degree of Diploma of

Electrical and Computer Engineer in

Technical University of Crete

October 2017

ADAPTIVE NEURO-FUZZY INFERENCE SYSTEMS (ANFIS) APPLIED ON MEDICAL DIAGNOSIS

Thesis Committee:

Professor Dr. Michael Zervakis

Advisor

School of Electrical and Computer Engineering

Technical University of Crete

Associate Professor Dr. Georgios Chalkiadakis

Member

School of Electrical and Computer Engineering

Technical University of Crete

Dr. Eleutheria Sergaki

Advisor

School of Electrical and Computer Engineering

Technical University of Crete

Date Approved: October 2017

ACKNOWLEDGEMENTS

Many thanks to my advisors from Technical University of Crete, Dr. Eleutheria Sergaki who cared and supported me through all the process. Her skills, network with the consulting cardiologist and other researchers helped me to clarify the ideas, and build work ethic for a scientific thesis. Without her help this thesis would not be refined.

Also I want to thank Professor Dr. Michael Zervakis and Associate Professor Dr. Georgios Chalkiadakis as members of the commission and my professors. Also thanks to the candidate PhD student Theodoros Kapetanakis and of course the cardiologist from Thriasio hospital, Dr. Eustathios Kartsagoulis whose help for his medical domain was enlightening. Thanks to all my university professors, everyone helped in a unique way.

I am grateful to my family which supported me and offers me unconditional love and also wants for me not only to be myself but to be my best self. Last but not least thanks to all these unique individuals and friends from every place of the earth I met and inspired me.

Personally, this thesis is an aggregation of childhood memories, continuous hard work, much curiosity and future missions.

[*] Contact informations: baroutisnk@gmail.com , +30 6944254859

ABSTRACT

The last thirty years Artificial Intelligence (AI) and Machine Learning (ML) used for computer systems to make fast, inexpensive, non invasive medical predictions and have a crucial importance as supporting tools for the doctors. Since 2013, cardiovascular disease (CVD) is the number one killer factor in the world with 31% of global population and also requires very costly and time consuming hospital treatment. From CVD 42% of the deaths are because of the coronary heart disease (CHD) which we research in this thesis and by using AI and/or ML to build a Computer Aided Diagnosis (CAD) diagnosis system which offers optimal predictability. CHD is the cause of many other CVDs and is incriminated for brain stroke too.

CHD is the stenosis of the main heart arteries caused when a wax substance called plaque builds up inside the coronary arteries. narrowing the coronary arteries and reducing the flow to the heart, leading to serious heart problems or heart failure. The danger of the disease is the silent appearance. The causes are: the age, sex, high cholesterol levels, angina, abnormal blood pressure, the years as smoker, the number of smoking cigarettes per day, family history, high fasting blood sugar, anxiety and the lack of exercise.

In this Thesis we examine the problem of Computer Aided Diagnosis (CAD) of Coronary Heart Disease (CHD), which classifies patients as well as possible with respect to the optimal minimization of the cost of diagnosis, the speed and the less stress and pain for the patients. By using AI and/or ML techniques our goal is to classify the patients in three levels of risk: Absence - Medium high - Very high risk differentiating our research from the previous researches since 1988 where the classification was binary (absence or presence). Then to achieve better results we went deeper into the data science and by using various data preprocessing techniques we aim to construct different datasets of patient's diagnosis data in order to find which dataset offers the best result. Furthermore, based on the above proposed concept we set apart our method even more by proposing a new dataset of patient's diagnosis data which is different than the data of previous researches. To achieve this, we consulted by cardiologist and used data preprocessing techniques.

We used the database from University of Cleveland which includes 298 patient cases, with 13 parameters per patient, used since 1988. Moreover, we used the patient's datasets of University of California Irvine (UCI) machine learning repository, which have 4% missing data of the 15% patient cases. In order to increase the Cleveland's database, we recovered the missing data of UCI's database, using statistical data preprocessing. The result is to increase the Cleveland's dataset by 21%. In collaboration with the cardiologist we constructed and proposed a new diagnosis dataset for each patient, including for each patient a subset of the existing until now parameters, such as: data from the interview answers, the biochemical blood test and from the electrocardiograph (ECG) test, excluding the parameters of stress test and fluoroscopy test.

We applied statistical data preprocessing on data and we processed them with the following AI and ML techniques: A) Adaptive Neuro-fuzzy Inference Systems (ANFIS) based on, i) Subtractive Clustering, ii) Fuzzy C Means, iii) Particle Swamp Optimization, iv) Genetic Algorithm, v) using datasets from PCA with all the above techniques again, B) Artificial Neural Networks (ANN). The mission was to find which strategy will export diagnosis with the optimal accuracy.

After multiply adjustments on the above techniques a multilayer Neural Network was is the best. We created a unique appropriate weight initialization for the feed forward pass and for the scaled conjugate gradient descent algorithm, also adjusted the levels, the nodes and the split ratio. 74% accuracy - mean value for the three classes. Specifically, the class Absence, which is the most important for the patient's safety on the scale of credibility based on ROC performance {Almost excellent, Very Good, Good, Mediocre, Worthless} has Very Good credibility. The classes Medium high and Very high risk have Good credibility. The supporting diagnosis system uses data from basic questions to the patient, simple biochemical examination and ECG, excluding the invasive-expensive-time consuming examinations such as the stress test and the fluoroscopy.

Keywords: Coronary heart disease, biochemical blood test, ECG, stress test, fluoroscopy, medical induction. Data mining: data preprocessing, missing values, statistical data analysis, dimensionality reduction (PCA). Classification: supervised learning. Adaptive Neuro-Fuzzy Inference System, Particle Swamp Optimization, Genetic Algorithm, Artificial Neural Networks. Software: Matlab, IBM SPSS, Xlstat.

Περίληψη

Τα τελευταία τριάντα χρόνια με την είσοδο της μηχανικής μάθησης (ΜΜ) και της τεχνητής νοημοσύνης (ΤΝ) στο κλάδο των ιατρικών επιστημών η έγκαιρη, μη δαπανηρή και μη επεμβατική ιατρική διάγνωση με αυτοματοποιημένα συστήματα αποτελεί σημαντικότατο υποστηρικτικό ιατρικό εργαλείο. Η καρδιαγγειακή πάθηση από το 2003 αποτελεί την πιο θανατηφόρα αιτία με ετήσιο ποσοστό θνησιμότητας 31% του παγκόσμιου πληθυσμού, ενώ επίσης για τους νοσούντες απαιτεί από τις πιο δαπανηρές και χρονοβόρες νοσοκομειακές θεραπείες. Εκ του 31% του αποθανόντος πληθυσμού λόγω καρδιαγγειακών παθήσεων το 42% οφείλεται στην στεφανιαία νόσο την οποία αποσκοπούμε να προβλέψουμε βέλτιστα με υπολογιστικό μοντέλο ΤΝ ή/και ΜΜ στην παρούσα διπλωματική. Αυτή η νόσος είναι και η γενεσιουργός αιτία για πληθώρα άλλων καρδιαγγειακών παθήσεων καθώς και για εγκεφαλικό επεισόδιο.

Ως στεφανιαία νόσος ορίζεται η στένωση των βασικών καρδιακών αρτηριών η οποία προκαλείται από τη συσσώρευση αθηρωματικού υλικού στον αυλό τους και παρεμποδίζει την αιμάτωση του καρδιακού μυ με τελικό αποτέλεσμα την καρδιακή ανεπάρκεια. Η επικινδυνότητα αυτής της νόσου έγκειται στην σιωπηλή εμφάνιση της. Παράμετροι και συμπτώματα που συσχετίζονται με την νόσο αυτή είναι η ηλικία, το φύλο, η υψηλή χοληστερίνη, ο στηθαγχικός πόνος, η αφύσικη αρτηριακή πίεση, η υπέρταση, το οικογενειακό ιστορικό, τα έτη ως καπνιστής, το πλήθος τσιγάρων ημερησίως, το υψηλό σάκχαρο στο αίμα, το άγχος και η έλλειψη άσκησης.

Πιο συγκεκριμένα, στην παρούσα διπλωματική εργασία ερευνούμε την εύρεση του βέλτιστου ελάχιστου σετ ιατρικών δεδομένων για τον ασθενή τα οποία με χρήση αλγόριθμων ΤΝ και ΜΜ πετυχαίνουν διάγνωση βέλτιστης κατηγοριοποίησης της στεφανιαίας νόσου ασθενών για τα τρία στάδια: Απουσία κινδύνου, Μέτρια Υψηλό,

Πολύ Υψηλό κίνδυνο, αντί αποκλειστικά για δύο (Απουσία ή παρουσία της νόσου) όπως συστηματικά από το 1988 μέχρι σήμερα είναι ο στόχος των αντίστοιχων ερευνητικών εργασιών. Στη συνέχεια, για να επιτύχουμε καλύτερα αποτελέσματα, προχωρήσαμε βαθύτερα στην επιστήμη των δεδομένων και χρησιμοποιώντας διάφορες τεχνικές προεπεξεργασίας δεδομένων, στοχεύουμε στην κατασκευή διαφορετικών συνόλων δεδομένων των δεδομένων διάγνωσης του ασθενούς προκειμένου να εντοπίσουμε ποιο σύνολο δεδομένων προσφέρει το καλύτερο αποτέλεσμα. Επιπλέον, με βάση τ ην παραπάνω προτεινόμενη ιδέα διαφοροποιήσαμε ακόμη περισσότερο τη μέθοδο μας, προτείνοντας ένα νέο σύνολο δεδομένων για τη διάγνωση του ασθενούς, το ο ποίο είναι διαφορετικό από τα δεδομένα προηγούμενων ερευνών. Για να το επιτύχ ουμε αυτό, πραγματοποιήσαμε διαβουλεύσεις με καρδιολόγο και χρησιμοποιήσαμε τεχνικές προεπεξεργασίας δεδομένων. Η εν λόγω διάγνωση κατηγοριοποίησης δεν έχει ερευνηθεί μέχρι στιγμής λόγω του μεγάλου πλήθος των παραμέτρων διάγνωσης και της πολυπλοκότητας του συνδυασμού τους. Η κατηγοριοποίηση γίνεται από τους γιατρούς λαμβάνοντας υπόψη προχωρημένες εξετάσεις των ασθενών που είναι χρονοβόρες και δαπανηρές, όπως το τεστ κοπώσεως και το σπινθηρογράφημα του μυοκαρδίου. Όταν είναι διαθέσιμες αυτές οι εξετάσεις, τα υπολογιστικά μοντέλα πρόβλεψης ασθένειας δεν συμβάλουν όμως σημαντικά ή και καθόλου στο ιατρικό συμπέρασμα.

Στους πειραματισμούς μας, χρησιμοποιούμε τη βάση δεδομένων 298 ασθενών του νοσοκομείου του Cleveland, η οποία περιλαμβάνει για κάθε ασθενή 13 τιμές σχετικών με την πάθηση. Αυτές οι παράμετροι χρησιμοποιούνται για τη σχετική διάγνωση από το 1988. Επιπλέον χρησιμοποιήσαμε τις βάσεις ιατρικών δεδομένων από το αποθετήριο του πανεπιστημίου University of California, Irvine (UCI). Αυτό διαθέτει δεδομένα (όχι πλήρη) με απολεσθείσες τιμές δεδομένων της τάξης του 4% για το 15% του πληθυσμού των ασθενών της βάσης. Μέσω προσωπικής παρατήρησης ελέγχτηκαν ένα προς ένα τα δεδομένα και κατόπιν μέσω στατιστικής ανάλυσης δεδομένων συμπληρώσαμε τις κενές τιμές των παραμέτρων. Τα δεδομένα αυτών των ασθενών ενσωματώθηκαν στη βάση δεδομένων ασθενών του Cleveland για να αυξηθεί ο πληθυσμός των ασθενών κατά 21%. Με τη συμβολή του συνεργάτη μας καρδιολόγου δημιουργούμε και προτείνουμε δικό μας υποσύνολο παραμέτρων διάγνωσης για τον κάθε ασθενή, με κριτήριο οι τιμές αυτών να προκύπτουν από απλές και οικονομικές ιατρικές εξετάσεις, όπως του βιοχημικού τεστ αίματος, του ηλεκτροκαρδιογραφήματος και τις απαντήσεις της συνέντευξης του ασθενή προς τον καρδιολόγο. Δεν συμπεριλάβαμε το τεστ κοπώσεως και το σπινθηρογράφημα του μυοκαρδίου.

Αρχικά, προκειμένου να εξεταστεί περεταίρω μείωση των παραμέτρων διάγνωσης, επεξεργαστήκαμε τα δεδομένα μας με μεθόδους στατιστικής προ-επεξεργασίας ανάλυσης δεδομένων με τον αλγόριθμο Κύριων Συνιστωσών και στη συνέχεια τα χρησιμοποιήσαμε για τους παρακάτω ευφυείς αλγόριθμους: A) Νεύρο-Ασαφή συστήματα συμπερασμού (ANFIS) βασισμένα σε i) subtractive ομαδοποίηση, ii) ομαδοποίηση fuzzy c means, iii) αλγόριθμο Βελτιστοποίησης Σμήνους Σωματιδίων για τη βελτίωση του ANFIS iv) Γενετικό Αλγόριθμο σε Νεύρο-Ασαφή συστήματα συμπερασμού για τη βελτίωση του ANFIS και B): εφαρμογή Νευρωνικών Δικτύων πολλαπλών επιπέδων (ANN), με σκοπό να βρεθεί ένα διαγνωστικό σύστημα με τη βέλτιστη ακρίβεια γενίκευσης.

Μετά από πολλαπλές παραμετροποιήσεις όλων των παραπάνω πειραμάτων το Νευρωνικό Δίκτυο πολλαπλών επιπέδων με τη δημιουργία μιας συνδυαστικής τεχνικής για την αρχικοποίηση των βαρών και με συνάρτηση μεταφοράς κλιμακωτών συζυγών κλήσεων ανάστροφης διάδοσης πέτυχε το βέλτιστο αποτέλεσμα.

Για το μειωμένο υποσύνολο δεδομένων ανά ασθενή που προτείνουμε, η βέλτιστη ακρίβεια γενίκευσης είναι 74% μέσος όρος από τις τρεις κατηγορίες κινδύνου. Για την κατηγορία Απουσία κινδύνου που είναι η πιο σημαντική διότι είναι το επίπεδο ασφαλείας για τον ασθενή έχουμε Πολύ Καλή πρόβλεψη κατά ROC το οποίο περιλαμβάνει την αξιολόγηση {Σχεδόν Άριστη, Πολύ Καλή, Καλή, Μέτρια, Άνευ Αξίας}. Για τις κατηγορίες Μέτρια Υψηλό και Πολύ Υψηλό κίνδυνο έχει καλή προβλεψιμότητα το μοντέλο. Το συγκεκριμένο σύστημα ιατρικής υποβοήθησης κάνει χρήση δεδομένων από τις απαντήσεις απλών ερωτήσεων προς τον ασθενή, το βιοχημικό τεστ αίματος και το καρδιογράφημα, εξαιρώντας δεδομένα από επεμβατικές χρονοβόρες και δαπανηρές μεθόδους διάγνωσης.

Λέξεις κλειδιά: Στεφανιαία νόσος, τιμές βιοχημικού τεστ αίματος, ηλεκτροκαρδιογράφημα, τεστ κοπώσεως, σπινθηρογράφημα του μυοκαρδίου, ιατρική συνεπαγωγή. Εξόρυξη δεδομένων: Προ-επεξεργασία δεδομένων, αντικατάσταση χαμένων τιμών, ανάλυση κυρίων συνιστωσών, ιατρική επαγωγή συμπερασμάτων, δεδομένα με λευκό θόρυβο. Κατηγοριοποίηση: Μάθηση υπό επίβλεψη. Νεύρο-ασαφές σύστημα συμπερασμού, Αλγόριθμος Σμήνους Σωματιδίων, Γενετικός Αλγόριθμος, Τεχνητά Νευρωνικά Δίκτυα. Λογισμικά: Matlab, IBM SPSS, Xlstat, MsExcel.

TABLE OF CONTENTS

LIST OF ACRONYMS

AI	Artificial Intelligence
ANFIS	Adaptive Neuro-fuzzy Inference System
ANN	Artificial Neural Network
AUC	Area Under the Curve
AWGN	Additive White Gaussian Noise
CAD	Computer Aided Diagnosis
CART	Classification And Regression Tree
CHD	Coronary Heart Disease
CV	Cross Validation
CVD	Cardiovascular Disease
ECG	Electrocardiogram
EVD	Eigen Value Decomposition
FCM	Fuzzy C Means
FIS	Fuzzy Inference System
GA	Genetic Algorithm
IQR	Interquartile Range
KDD	Knowledge Discovery in Databases
KMO	Kaiser-Meyer-Olkin
K-NN	K-Nearest Neighbour
MAR	Missing at Random
MCAR	Missing Completely At Random
MF	Membership Function
ML	Machine Learning
MLP	Multilayer Perceptron
MNAR	Missing Not at Random
NIPALS	Nonlinear Iterative Partial Least Squares
NN	Neural Networks
NP	Non Polynomial
PCA	Principal Component Analysis
PSO	Particle Swamp Optimization
PreLU	Parametric Rectified Linear Unit

ReLU	Rectified Linear Unit
RMSE	Root Mean Square Error
ROC	Received Operating Characteristic
SNR	Signal to Noise Ratio
SVD	Singular Value Decomposition
SVM	Support Vector Machines
TNR	True Negative Rate
TPR	True Positive Rate
UCI	University of California Irvine

CHAPTER ONE

Introduction

1.1 Thesis Objectives

In this Thesis we examine the problem of Computer Aided Diagnosis (CAD) of Coronary Heart Disease (CHD), which classifies patients as well as possible with respect to the optimal minimization of the cost of diagnosis, the speed and the less stress and pain for the patients. Our goal is by using the optimal data and inserting them to AI and/or ML techniques to predict the patient's risk in three levels of risk: Absence - Medium high - Very high risk differentiating our research from the previous researches since 1988 where the classification was binary (absence or presence). Then to achieve better results we went deeper into the data science and by using various data preprocessing techniques we aim to construct different datasets of patient's diagnosis data in order to find which dataset offers the best result. Furthermore, based on the above proposed concept we set apart our method even more by proposing a new dataset of patient's diagnosis data which is different than the data of previous researches. To achieve this, we consulted by cardiologist and used data preprocessing techniques.

Chapter 3, Chapter 7 Chapter 8 contain our work. Chapter 2, Chapter 4, Chapter 5 and Chapter 6 explains briefly the important parts of the theory we applied for our experiments. and Chapter 9 is the conclusion and summarizes the results.

1.2 Thesis Overview

Chapter 1 explains the thesis objectives, and the brief approach of the present work.

Chapter 2 describes how the data mining process take place and the data preprocessing techniques we used.

Chapter 3 shows the application of the data-preprocessing techniques we used to create different scenarios of reduced datasets for our learning models which we used in every system.

Chapter 4 includes the theoretical fundamentals behind Artificial Neural Networks (ANN) which we used to create ANN learning models.

Chapter 5 describes the fundamental theory of Adaptive Neuro-Fuzzy Inference Systems (ANFIS) we used as well the Optimization algorithms which used to test if the ANFIS model can be improved.

In chapter 6 we show previous studies in literature with the use of ANFIS, ANN and various other techniques and their results.

In chapter 7 includes our experiments using ANFIS based on, i) Subtractive Clustering, ii) Fuzzy C Means, iii) Particle Swamp Optimization, iv) Genetic Algorithm, v) using datasets from PCA Particle Swamp Optimization, v) Genetic Algorithm and vi) Artificial Neural Networks (ANN), using different scenarios for each different algorithm and the corresponding experimental results.

Chapter 8 describes our experiments based on ANN for different scenarios and the corresponding experimental results.

Chapter 9 explains the conclusions of the above chapters describing also the optimal result and the goal we achieved which had very good success.

1.3 Software used in the present Thesis

1. IBM SPSS, a predictive analytics commercial software which provides statistical analysis/reporting, in order to replace the missing values on datasets.

2. Xlstat, a predictive analytics commercial software with the use of MSExcel (is easier than in Matlab® for data preprocessing).

3. Matlab® used for data-preprocessing, to find the distribution of the data, to detect the outliers, to do the data scaling, to increase the size of a dataset with by adding AWGN type of noise and to show the results.

4. For models learning used Matlab® to design the ANFIS, the Optimization Algorithms, the ANN and to project their results.

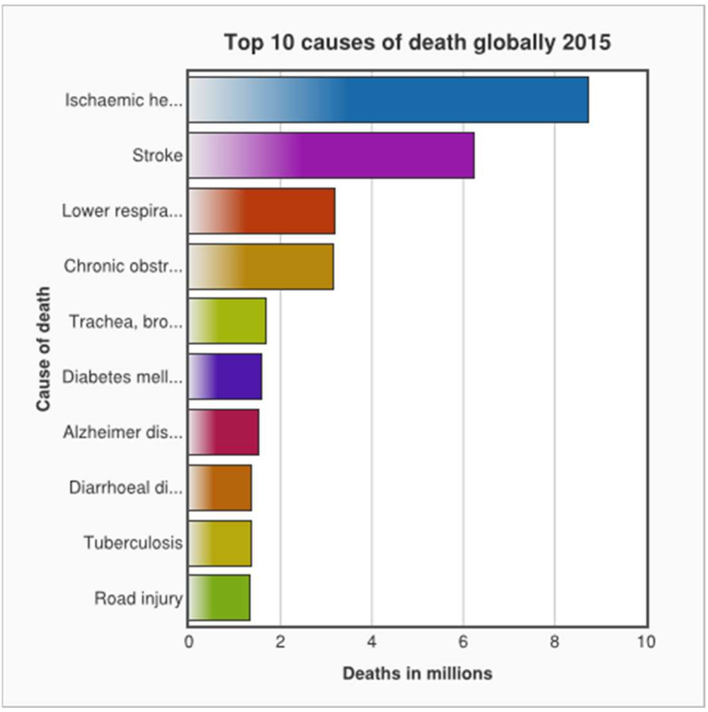
1.4 Thesis research importance

14.1 Introduction to Cardiovascular Heart Disease

Heart disease is a group of conditions affecting the structure and functions of the heart and has many root causes. Heart attack transpires when there is indiscretion in the flow of blood and heart muscle is injured because of inadequate oxygen supply [4]. Risk factors for heart disease include smoking, high blood pressure, cholesterol, high blood sugar, family history, age, sex, anxiety, lack of exercise. The most common symptom is severe chest pain and so called as angina, and fast fatigue. Cardio Vascular Disease (CVD) clinical guidelines spotlight on the management of single risk factors.

Heart disease is the leading cause of death in all over the world in recent years since

2003 [1]. An estimated 17,7 million people died from CVDs in 2015, representing 31% of all global deaths. Of these deaths, an estimated 7,4 million were due to coronary heart disease (CHD) also named ischemic heart disease or atherosclerosis and 6,7 million were due to stroke. The worst scenario is that CHD is related with heart stroke because of cut off blood flow to the brain. Individuals with CHD or those who had a heart attack due to CHD have more than twice the risk of stroke than those who have not, so this highlights the importance of CHD even more [4].



.

Figure 1.1 - World Health Organization.The top 10 causes of death globally in 2015 [1].

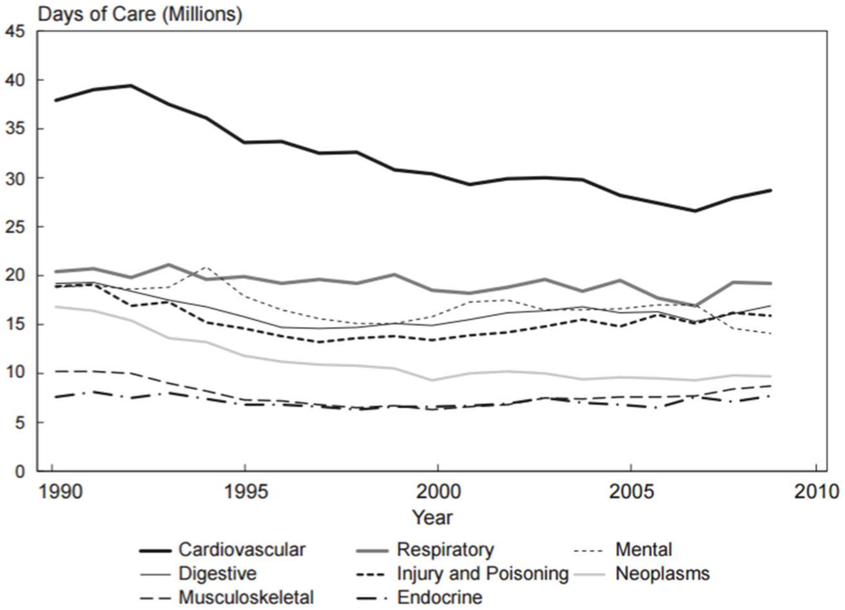


Figure 1.2 – Different diseases and the number of days of hospital care by major diagnosis in USA, 1990–2009. Cardiovascular disease requires much more hospital care than every other disease [3].

The early prediction and prevention of CVD profoundly is crucial for the survival but is very important for other reasons as well. In figure 1.2, CVD from 1990 to 2009 cardiovascular disease ranked first and respiratory disease ranked second in the number of days for which patients received hospital care [3]. Therefore, an early and accurate prediction can reduce the days of care for the patient and the hospital, reducing the emotional, physical, financial cost as well.

1.4.2 The Coronary Heart Disease: Causes and symptoms

From CVD 42% of the kills are because of the coronary heart disease (CHD). CHD is the cause of many other CVDs. CHD is the stenosis of the main heart arteries caused when a wax substance called plaque builds up inside the coronary arteries. narrowing the coronary arteries and reducing the flow to the heart, leading to chest pain, other serious heart problems or heart failure. The danger of the disease is the silent appearance. The causes are: the age, sex, high cholesterol levels, angina, abnormal blood pressure, the years as smoker, the number of smoking cigarettes per day, high fasting blood sugar, anxiety and lack of exercise.

The causes of CHD affect every ethnicity and race the same without a proof for the opposite [44] [45]. Studies say that 50% of black people in U.S.A have CHD when for whites is 33% but the difference is because of unhealthy lifestyle and avoidance of early detection for financial reasons. Other studies found that in U.S.A 41% of black population have high blood pressure, as compared to 27% of whites. The hypothesis is that high rates of high blood pressure in African-Americans may be due to the genetic make-up of people of African descent and this may affects the CHD, however black people in U.S.A are more likely to be overweight than blacks in other countries [47]. In conclusion, it would be a tragic fault to differentiate the prediction of CHD based on the race because the causes are many and affects every ethnicity and race the same.

1.5 Purpose of this study

In this thesis we research for reduced diagnosis data attributes and preprocess medical datasets, design and implement Artificial Intelligence systems and apply Machine Learning algorithms for in order to achieve a CAD diagnosis system which offers the optimal prediction and classification of CHD into three levels of risk: Absence, Medium high, Very high risk, differentiating our research from the previous researches since 1988 where the classification was binary (absence or presence). The classification with three levels of risk has not researched because of the complexity and the significant lack of data. This is the reason why the researchers use data after the patients took advanced heart examinations, such as stress test and fluoroscopy which they are costly, time consuming and stressful and painful (sometimes). Consequently, these predictive systems do not offer significant help to the doctor because he can do the predictions by himself with very good accuracy.

The goal is not to substitute the role of the doctors, instead is to help them and the patients to diagnose the disease faster. So the doctor can warn the patients much sooner and if their life is in high danger to send them for advanced examinations sooner. From such a system more patients can warned for their feature and avoid the delay when the number patients is large.

1.6 Past Research on CHD

Previous works of CHD classification by applying different kinds of classification techniques was for two classes, absence and presence of the disease, which is a very general classification for a disease. Also they used 13 attributes which many of them are very specific and hard to find. To do such a model with so much generality and with the use of very specific data does not offer usefulness to the doctors. More details in chapter 6.

1.7 Brief Approach of the present work

We use diagnosis data of real patients from the database from University of Cleveland which includes 298 patient cases, with 13 parameters per patient, used since 1988, and the similar databases from University of California, Irvine (UCI) machine learning repository. First of all, due that the UCI databases have 4% missing data of the 15% patient cases we recovered the missing data of USI's database, using statistical data preprocessing. The result is to increase the Cleveland's dataset by 21%. After that, in collaboration with the cardiologist, we propose a reduced patient dataset for each patient. Following, we apply and compare statistical data preprocessing techniques, Artificial Intelligence and Machine Learning techniques in order to diagnose the CHD and classify the patients in three levels of risk: Absence - Medium high - Very high risk. We evaluate the performance of our methods with measurements using real different datasets from the above databases. Analytically, the steps we followed are:

1st. Used UCI repository with data for coronary heart disease. The only free repository, other repositories for hospitals forbid to provide such private data without the agreement of legislation.

2nd. Collect and clean the data among different directories and databases from different hospitals inside the UCI repository.

3rd. Cardiologist checked and ranked the measures (attributes) of the UCI repository.

4th. Multiply Imputations algorithm, Linear Interpolation algorithm used to find the missing values.

5th. Used four datasets: Cleveland dataset with [298x14] size, 298 patients and 14 measures for each patient with 14th as the result of diagnosis, Multiply Imputations dataset [364x14] size, Liner Interpolation dataset [364x14] size, AWGN dataset [464x12] (created new dataset by adding noisy data).

6th. Principal Component Analysis to check if we can find interrelationship among attributes and to reduce its number in order to hold the most important.

7th. To create learning models: standard ANFIS with subtractive clustering, standard ANFIS with fuzzy c mean clustering, ANFIS with Particle Swamp Optimization algorithm to test if we can improve the ANFIS, ANFIS with Genetic Algorithm to test if we can improve the ANFIS, multilayer ANN.

8th. Compare the techniques to find the one with the best accuracy of CHD prediction.

CHAPTER TWO

Data Mining

2.1 Knowledge Discovery in Databases Process

Data mining, also popularly referred to as knowledge discovery from data (KDD), is the automated or convenient extraction of patterns representing knowledge implicitly stored or captured in large databases, data warehouses, the Web, other massive information repositories or data streams [38].

There is an urgent need for a new generation of computational theories and tools to assist humans in extracting useful information (knowledge) from the rapidly growing volumes of digital data. These theories and tools are the subject of the emerging field of KDD [5]. At the core of the process is the application of specific data-mining methods for pattern discovery and extraction.

The traditional method of turning data into knowledge relies on manual analysis and interpretation. In the health-care industry, it is common for specialists to periodically analyze current trends and changes in health-care data, say, on a quarterly basis. The specialists then provide a report detailing the analysis to the sponsoring health-care organization; this report becomes the basis for future decision making and planning for health-care management.

The additional steps in the KDD process, such as data preparation, data selection, data cleaning, incorporation of appropriate prior knowledge, and proper interpretation of the results of mining, are essential to ensure that useful knowledge is derived from the data. Blind application of data-mining methods can be a dangerous activity, easily leading to the discovery of meaningless and invalid patterns.The KDD process is interactive and iterative, involving numerous steps with many decisions made by the user. Brachman and Anand (1996) give a practical view of the KDD process, emphasizing the interactive nature of the process. The basic steps are:

1st step is developing an understanding of the application domain and the relevant prior knowledge and identifying the goal of the KDD process from the problem's viewpoint.

2nd step is creating a target data set: selecting a data set, or focusing on a subset of variables or data samples, on which discovery is to be performed.

3nd step is data cleaning and preprocessing. Basic operations include removing noise if appropriate, collecting the necessary information to model or account for noise, deciding on strategies for handling missing data fields, and accounting for time-sequence information and known changes.

4th step is data reduction and projection: finding useful features to represent the data depending on the goal of the task. With dimensionality reduction or transformation methods, the effective number of variables under consideration can be reduced, or invariant representations for the data can be found.

5th step is matching the goals of the KDD process to a particular data-mining method. For example, summarization, classification, regression, clustering, and so on, are described later as well as in Fayyad, Piatetsky-Shapiro, and Smyth (1996).

6th step is exploratory analysis and model and hypothesis selection: choosing the data mining algorithm(s) and selecting method(s) to be used for searching for data patterns. This process includes deciding which models and parameters might be appropriate and matching a particular data-mining method with the overall criteria of the KDD process, for example, the end user might be more interested in understanding the model than its predictive capabilities. In the case of prediction from health decease prediction predictive capabilities of the model is the end goal of KDD process.

7th step is data mining: searching for patterns of interest in a particular representational form or a set of such representations, including classification rules or trees, regression, and clustering. The user can significantly aid the data-mining method by correctly performing the preceding steps.

8th step is interpreting mined patterns, possibly returning to any of steps 1 through 7 for further iteration. This step can also involve visualization of the extracted patterns and models or visualization of the data given the extracted models.

9th step is acting on the discovered knowledge: using the knowledge directly, incorporating the knowledge into another system for further action, or simply documenting it and reporting it to interested parties.

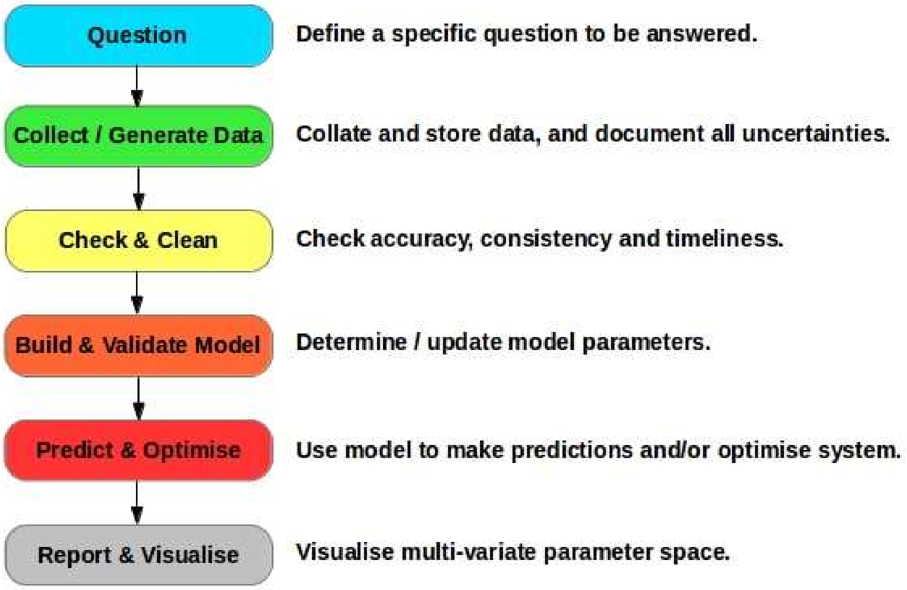

Question
Define a specific question to be answered.
Collect / Generate Data
Collate and store data, and document all uncertainties.
Check & Clean
Check accuracy, consistency and timeliness.
Build & Validate Model
Determine / update model parameters.
Predict & Optimise
Use model to make predictions and/or optimise system.
Report & Visualise
Visualise multi-variate parameter space.

Figure 2.1 – The KDD process.

2.1.1 Data Pre-processing: Techniques for handling missing data

Missing data should be identified by the kind of their missingness. The type of category they belong requires different techniques for handling the missing data [11].

Their categorization is:

Missing Completely at Random (MCAR)

The missing data are unrelated with other observed data or unrelated with the other missing data from the attribute itself.

Missing at Random (MAR)

The missing data are related with the data from other attributes. The probability of missing data can be explained by other attributes. An example is when older responders have more missing data than younger responders. However, the age can ex-plain the missingness.

Missing Not at Random (MNAR)

The probability of missing data for a certain attribute is related to the values on that variable itself. An example is that responders with low income intentionally skip questions about their low income because it violated their privacy. In that case the set is biased. MNAR is a serious problem and requires different techniques.

If the data belong to categories MCAR or MAR different techniques have to be applied in order to find them than NMAR category. The patterns of the data provide the insight where they belong. There are many techniques for handling missing data but the most used are below. Some of them are appropriate and others not, in every case the researcher should experiment with many of them.

Contact the owner for more informations.

2.1.2 Data Pre-processing: Outliers, Data Scaling

Contact the owner for more informations

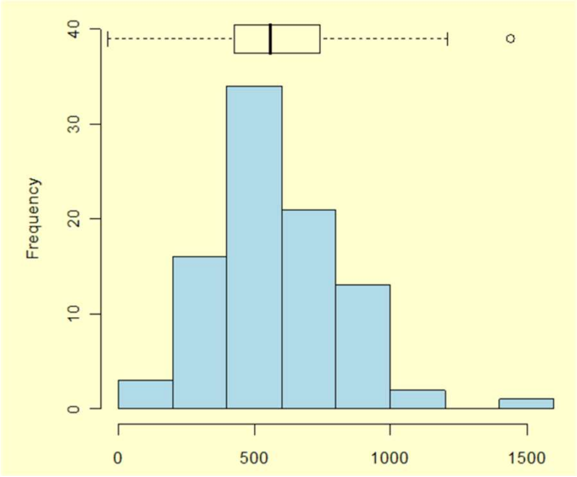


Figure 2.2 – An outlier and the aproximation to normal distribution of the dataset. Outlier data point is the circle.

<u>Data scaling Normalization (or called Min-Max scaling), scaling [min= 0, max= 1]:</u>

Another cases is, if from an attribute1 it's values ∈ R1, for attribute2 ∈ R2 and so on and so forth with respect to R1 and R2 as independent sets the clustering should be under a common range to be right. The solution is the normalization or standardization of the dataset for each column separately. In normalization all the data normalized and $\vec{x}$ ∈ [0,1], or [-1,1] for symmetric S-type functions so they are appropriate for Neural Network's derivative method algorithms. Different ways have found in the literature as Min-max, tanh and median normalization. However, the outliers from the data set will rejected so this is not an appropriate approach for ANN. Normalization for small datasets should be applied on training set and test set separately using the following formula to avoid false scaling [39]. For Normalization [0,1] (scaling to [0, 1] ranghere test data are the data used for testing the model. Testing data provide the "how well the model reacts to new unseen data of the world."

Data scaling: Standardization, [N(μ=0, σ=1)]:

An alternative approach to data scaling normalization is standardization. Standardized values are useful for tracking data that is otherwise incomparable because of different metrics or circumstances. The mean of standardized values will always be zero, and the standard deviation will always be one. The graph of standardized values will have exactly the same shape as the graph of raw data, but it may be a different size and have different coordinates. Standardization is the process which rescales the data to have a mean of zero and unit variance. One method is the **z-scores standardization**. It tells us how far from the mean we are in terms of standard deviations, as follows:

$$zscores = \frac{x_{ij}-\mu_x}{\sigma_X} \text{ , } for\ array\ X \tag{2.3}$$

where σ_x is the standard deviation of *X*, μ_x is the mean value of total population *X*ij is the sample mean.

For a small dataset standardization is different than from a large one. When the distribution of data is organized they are often ordered from smallest to largest or are broken into reasonably sized groups so the most frequent values will affect the conclusions more than the normal way, there formulas are below. Hence, standardization retains the outliers from deletion and makes the data to be bounded inside the small range of a hypersphere. Also, both normalization, or standardization, make gradient descent algorithms to converge faster as a result of symmetric contours.

Often researchers scale all the data and then split into train/test data sets. It is safe only when we are not trying to produce a generalized predictive model. In this case, the testing will validate the model alone, which may be useful if the aim is not to produce a predictive algorithm but to understand the structure of the data (i.e. important variables).

Standardization of training and testing data:

One question is if the data normalization (x-mean(x))/std(x) for the testing data will use the train Mean and Standard Deviation? This is all dependent on size of data sets and whether both train and test are equally representative of the domain we are trying to model. If we have thousands of data points and the test set is fully representative of the training set (hard to prove) then either method will be fine. If using a small but representative test data set then normalizing using the training parameters only is best as sampling errors may negatively bias the predictions.

Standardization formulas for small size datasets are:

Contact the owner for more informations

Also distributions of data with unusual statistical properties non Gaussian, polymodality and heavy tails, Poisson, Binomial, Exponential, Log-normal, Largest-extreme-value, Weibull may result in incorrect conclusions. For example, an attribute with 90% “0” and 10% “1” is not symmetrical, has a heavy tail and is unbalanced

Z-score standardization or Min-Max scaling?

In clustering analyses, standardization may be especially crucial in order to compare similarities between features based on certain distance measures. Another prominent example is the Principal Component Analysis (PCA), where we usually prefer standardization over Min-Max scaling, since we are interested in the components that maximize the variance (depending on the question and if the PCA computes the components via the correlation matrix instead of the covariance matrix.

2.2 Dimensionality reduction with Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is the most famous factor analysis technique that identifies a meaningful interrelationship among the attributes in the data and reduce the dimensions, where for each dimension we consider a feature, without losing useful information [39].

PCA invented in 1901 by Karrl Pearson, it was later independently developed and named by Harold Hotelling in the 1930s. The algorithm is based or on Singular Value Decomposition (SVD) (Golub and Van Loan, 1983) which is faster on computational complexity or on Eigen Value Decomposition (EVD), Matlab® by default uses SVD.

Before PCA, normalization or standardization should take place to bound the data inside the unary hypersphere but not all the times. We use Standard Principal Component Analysis. Standard PCA is for linear–almost linear data distribution, Kernel PCA for data distributed that shaping a curve, circular PCA for data distributed in a cyclic way. PCA may be chosen to have various steps, named principal components, PC-1 for the first principal component, PC-2 for the second component with as many principal components as the number of features.

PCA Concept:

The algorithm searches for one linear vector which is named eigen vector, orthogonal (or not) towards the current axis xx' to project the dataset. This eigen vector is considered the new principal component and is constructed from the eigenvalues which are the weight of significance for each data point. The choice of the right eigenvector based on the finding of the right eigenvector with the maximum magnitude. The maximum magnitude means the maximum amount of variance of the projected data upon this new component. Maximum variance means maximum uniqueness of the information.

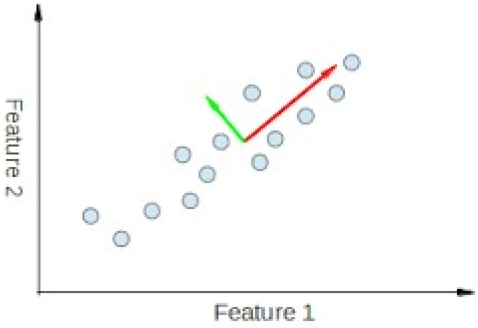


Figure 2.3 – PCA consept: Eigenvectors from a dataset.The red eigen-vector has the most variance and it's extension will be the first principal component.

The direction of the vector is the direction which the projected dataset will point as shows the picture with the red eigenvector in figure 2.3. The red eigenvector will be extended and the data will be projected, as shown in figure 2.4. Its extension will be chosen as the new axon and it is named the first principal component.

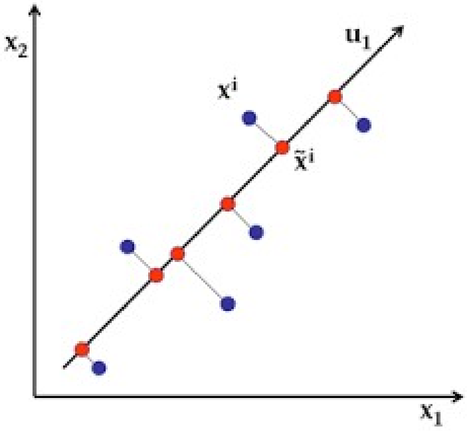


Figure 2.4 – PCA concept: Projections of data-points to the selected extended eigen-vector [39].

After the projection on the new component the same process can create another component orthogonal (or not) to the previous component. Orthogonality between vectors means uncorrelated features, these methods are “varimax”, “quartimax” and “equimax”. There are different rotating methods without orthogonal rotation. When the components should not be orthogonal, it means that there is correlation between the factors and the methods for this are called oblique rotations which consist of the “direct oblimin” and “promax” rotations. In general if the sum of the variances of the first few principal components exceed 80% of the total variance of the original data it gives enough understanding for the driving forces of the original data.

Based on Tabachnick and Fiddell (2007, p.646) [29] as a rule of thumb we look at the factor correlation matrix for correlations around 0,32 and above. If correlations exceed 0,32 then there is 10% (or more) overlap in variance among factors, enough variance to warrant oblique rotation unless there are compelling reasons for orthogonal rotation, which are based on the problem.

2.3 Case Studies in Literature

2.3.1 Case studies with data mining techniques

In table 6.1 there are case studies with an amount of data mining techniques for binary classification (absence or presence) on heart disease based on UCI machine learning repository. Although nobody mentions if the accuracy is on training dataset or on testing, which is significant.

Table 2.1 – Case studies in literature with different data mining techniques and their performance on binary classification.

Reference	Data Mining Techniques	Accuracy Obtained		Number of Attributes Used(without targets labels)	Best Technique
Purusothaman G et al (2015)	Decision Tree	76%		13	Hybrid Model
	Associative Rules	55%			
	K-NN	58%			
	Artificial Neural Networks	85%			
	Support Vector Machine	86%			
	Naïve Bayes	69%			
	Hybrid models	96%			
Srinivas K et al (2010)	Decision Trees (C4.5 algorithm)	82,5%		15	Neural networks (MLP)
	Neural networks (MLP)	89,75%			
	Naïve Bayes	82%			
	Support Vector Machine	82,5%			
Chaitrali S et al (2012)	Decision Trees	96,66%	99,62%	13 and 15	Neural Networks
	Naive Bayes	94,44%	90,74%		
	Neural Networks	99,25%	100%		
John Peter T et al (2012)	Naïve Bayes	83,70%		13	Naïve Bayes
	Decision Tree	76,66%			

	K-NN	75,18%		
	Neural Network	78,485%		
Hlaudi DM et al (2014)	J48 (Weka)	99,0741%	11	J48, REPTREE and SIMPLE CART algorithm
	Bayes Net	98,148%		
	Naive Bayes	97,222%		
	Simple Cart	99,0741%		
	REPTree(Weka)	99,0741%		
Gnanasoundhari SJ et al (2014)	Naive Bayes	52,33%	11	Weighted Associative Classifier
	Neural network	78,43%		
	Weighted Associative Classifier	81,51%		
	Support Vector Machine	60,78%		
Anbarasi M et al (2010)	Naive Bayes	96,55%	6	Decision Tree
	Classification by clustering	88,3%		
	Decision Tree	99,2%		

Another research is from P.Pamela, Gayathri.P and Jaisankar.N which they used PSO for the optimization of the fuzzy membership function. The results obtained from the fuzzy system are interpreted and it was observed that the accuracy was good for binary classification. Cleveland dataset and Switzerland dataset merged and achieved accuracy on training data 92,2 % and accuracy on testing dataset 86%. After applying PSO optimization achieve very good performance of 94,4% on training and 94% on testing [32].

Mohd. A.M. Abushariah, Assal A.M Alquadah, Omar Y.Adwan, Rana M.M. Yousef, designed ANFIS with Cleveland dataset using subtractive clustering with 10 fold cross validation and achieved accuracy 100% on training dataset and accuracy 75,93% on testing dataset without doing any preprocessing upon the data and using binary classification. We criticize their results as poor and this happened because the parameters for subtractive clustering kept fixed with the default values of genfis2 and did not experimented with the options. Also with ANN they improved the results with accuracy 90,74% on training dataset and 87,04% on testing dataset for binary classification [33]. The result just for binary classification is not very promising for the amount of features. We achieved accuracy of 91% with binary classification on testing dataset using the above dataset after applying numerous adjustment on a ANN as it is described in Chapter 8.

Yan, Jiang et al (2006) with a three-layer MLP and 15 nodes using data from the Southwest Hospital and the Dajiang Hospital, both located in P. R. China achieved accuracy for diagnosis (presence or absence) of CHD on training data 91,5 % and on testing data 90,4 %. The dataset consists 352 patients and 40 attributes. For learning used a back propagation algorithm augmented with the momentum term, the adaptive learning rate, the forgetting mechanics, and an optimized algorithm based on the conjugate gradients method [49].

2.3.2 Case studies with dimensionality reduction methods

The problem with multidimensional matrices in neural networks architectures and data mining is the curse of dimensionality, reducing the classification performance of the model and increasing the computational complexity. The existence of many dimensions leads the data in becoming more sparse from the center and fall out of the n −dimensional hypersphere where should belong. The hypersphere reduces its volume as the n increases based on Euler's formula:

$$V(n) = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)} R^n \qquad (2.6)$$

where R is the radius of the hypershere.

As a result, more data are in the corners and they cannot be caught, therefore the classification fails. Normalization or standardization can reduce this by bounding them to unary – n dimensional space but the data will still be on the corners so the problems remains.

When the number of training data is fixed, 298 samples for Cleveland dataset, then many dimensions has the effect of over fitting. On the other hand, if the number of dimensions is already large and kept fixed, 13 for Cleveland dataset, then the number of training data should grow exponentially one power for each one dimension to avoid overfitting. In medical diseases hundreds of thousands of samples are not available nor a feasible solution in respect to computational efficiency. So reducing the dimensions is a feasible solution if we want to avoid overfitting and/or less computational complexity for classifiers as Neural Network architectures.

Classifiers that tend to model non-linear decision boundaries very accurately do not generalize well and are prone to overfitting. In this category belongs the Neural Networks, K-NN classifiers, Decision trees and therefore the dimensionality should be kept relatively low. On the contrary if a classifier does not classify very accurately but generalizes easily then the number of used features can be higher since the classifier itself is less expressive in computational complexity. In this category belongs naive Bayesian, SVM and other linear classifiers.

In literature there are case studies where suggest that selecting the best attributes and eliminating features with little information is a solution to the problem of dimensionality for the Cleveland dataset. Although searching for an optimal feature selection dataset is a NP-complete problem [31]. The most common algorithms for dimensionality reduction without sacrificing the efficiency factor significantly are: Particle Swamp Optimization (PSO), Genetic Algorithm (GA) Feature Selection, Principle Component Analysis (PCA) and many combinations of classification techniques like Support Vector Machines (SVM) with the above algorithms.

Durairaj, Sivagowry [30] using PSO algorithm on Cleveland dataset tried to select the best features and rejects those with the less influence. From 13 features it reduced to 5 (CP, Exang, Slope, Ca, Thalach) .The performance of classification for healthy and diseased with 5 features with Radial Basis Function Network (RBF) and Multilayer Perceptron (MLP) shown in table 6.2 below. The performance with the reduced dataset is not good because the selected features do not belong to the optimal available feature selection as founded reading the research from Santhanam and Ephzibah which is shown next. Although, the study considers the result as good enough. Although we rejected them as poor considering that it is binary classification, and we did not use these 5 features with ANFIS nor ANN.

Table 2.2 – Performance of ANN, with 5 attributes selected with PSO, for binary classification [30].

	RBF	Multilayer Perceptron Neural Network
Classification Accuracy%	83,49	81,84
Precision	0,83	0,81
Kappa Statistic (good value> 0,7)	0,66	0,63
Relative Mean Squared Error	0,35	0,38

E.Santhanam and E.P. Ephzibah implemented PCA for Cleveland dataset an their proposed method try to optimize the feature selection process and increase the classification accuracy [31]. It is observed that for one of the proposed methods, for PC-1, the prediction accuracy is 92,0% using regression and 95,2% using feed forward neural network classifier which is better than other methods. It is also observed that the accuracy of exp(B) statistic is closer to PC-1, hence concluding that the exp(B) can also be considered for feature selection. Exp(B) statistical formula found the predictive capability for each feature on table 6.3 below. The highest predictive capability is for the sex and the lowest for fasting blood sugar [31]. Their results are aligned with the PCA we followed using orthonormal or oblique rotation.

Table 2.3 - EXP(B) statistic shows the significance of each attribute for Cleveland dataset, [31].

Order	Features	EXP(B)	Order	Features	EXP(B)
1	Sex	3.714	8	Rest ECG	1,278
2	Ca	3,553	9	Trestbps	1,024
3	Exang	2,525	10	Cholesterole	1,005
4	Chest Pain type	1,779	11	Age	0,986
5	Slope	1,738	12	Thalach	0,980
6	Thal	1,410	13	Fasting blood sugar	0,360
7	Old peak	1,281			

CHAPTER THREE

Datasets Used in Thesis

3.1 Thesis Dataset 1: extracted from the Cleveland

We label as Dataset 1 the dataset from the databases of Cleveland University. The Cleveland has been used by researchers since 1988 for thousands of researchers. It has 303 records of patients but 4 has missing data so 298 are the clear records. Also, it has 14 attributes from measures with the 14th as the classification of patient's state with coronary heart disease. Table 3.1 describes the 14 attributes of the Cleveland dataset. The classification is 1,2,3,4 indicating the number of vessel with more than 50% blockage and 0 for absence.

Table 3.1 – CHD patient's diagnosis attributes. Description of the 14 attributes from the Cleveland database [298x14].

	Attribute	**Description**
1	Age	Age is in year.
2	Sex	Value ("0" = male, "1" = female).
3	CP	Chest pain type ("1" = typical angina, "2" = atypical angina, "3" = non-angina pain, "4" = asymptomatic).
4	Trestbps	Resting blood pressure in mm Hg.
5	Chol	Serum cholesterol in mg/dl.
6	Fbs	Indicator of whether fasting blood sugar was > 120 mg/dl ("1" = yes, "0" = no).
7	Restecg	Resting electrocardiographic results ("0" = normal, "1" = ST-T wave abnormality, "2" = probable or definite left ventricular hypertrophy).
8	Thalach	Maximum heart rate achieved.
9	Exang	Indicator of whether the angina is exercise induced ("1" = yes, "0" = no).
10	Oldpeak	ST depression induced by exercise relative to rest.
11	Slope	The slope of the peak exercise ST segment ("1" = up sloping, "2" = flat, "3" = down sloping).
12	Ca	Number of major vessels colored by fluoroscopy.
13	Thal	Summary of heart condition ("3" = normal, "6" = fixed defect, "7" = reversible defect).
14	Num	"The Disease Diagnosis" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4. Here the 0 is denoting no presences of heart disease and 1, 2, 3, and 4 are presenting the presence of heart disease.

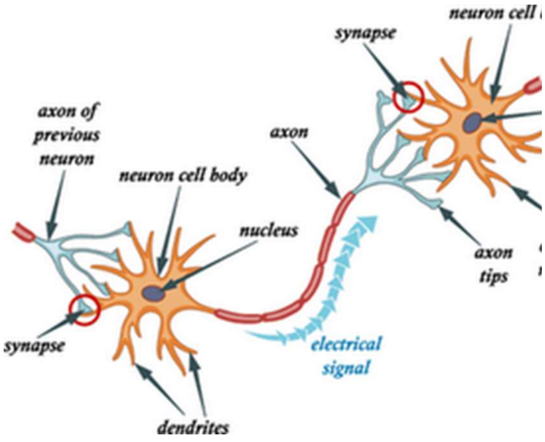
Contact the owner for more informations

CHAPTER FOUR

Artificial Neural Networks: Concepts and Terminology

4.1 From Biological Neurons to Artificial Neurons

The human brain is a source of natural intelligence and a truly remarkable parallel computer. Brain cells function about 10^6 times slower than electronic circuit gates, but human brains process visual, sense of touching and auditory information much faster than any modern computer could do in millions of years. The method behind this speed is the number of the neurons which is around 10^{10} with the number of connection much more.



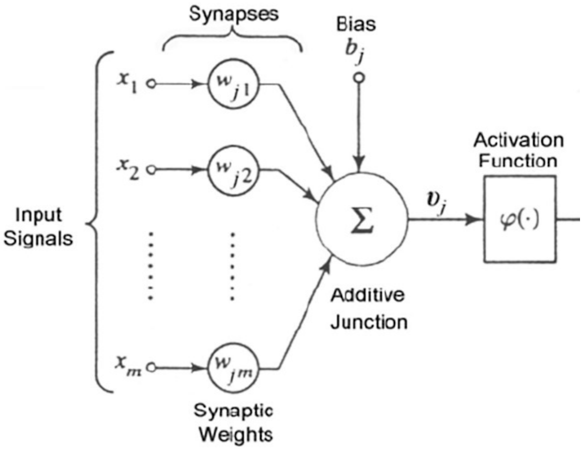


Figure 4.1 – Model of biological neuron

Figure 4.2 –An artificial neuron [26].

A biological neuron as in figure 4.1 works as follow:

1. A neuron receives many electro-chemical signals though dendrites and these signals deliver to nodes which called synapses.
2. Synapses supervise the signals and based on the supervision they set a weight to every signal.
3. The neuron keeps track of the input signals that receives from the synapses for a small time window, after that nothing comes in for a time interval. The total input signal to the cell is the sum of all such synaptic weighted inputs.
4. When the total signal reaches a certain threshold the neuron bursts into activity and generates a spike and through the axons the new signal deliver to other neurons.

Inspired by the biological neural system many researchers applied this methodology of the neural system to artificial neural networks in order to achieve good information processing.

This methodology replaces the ordinary algorithmic approach and does not requires critical decision flows in it's algorithms and based on different connections there are different kinds of *Artificial Neural Networks (ANN).*

4.1.1 Model of an artificial neuron

A neuron is an information processing unit that is fundamental to the operation of a neural network. The figure 4.2 shows a model of an artificial neuron which is identifying by three basic elements:

1. A set of synapses each of which is characterized by a weight or strength. A signal x_i at the input synapse j connected to neuron k is multiplied by a synaptic weight w_{kj}. The weight may take positive or negative values.
2. An adder for summing the input signals. The operations are a linear combination.
3. An activation function for limiting the amplitude of the output of a neuron, as it squashes the amplitude range of the output signal to some finite value.

In figure 4.2 the model also includes an external bias factor b_k. The bias increase or decreases the net input of the activation function by a constant, depending on the positive or negative value. The form of the kth net output is:

$$u_k = \sum_{j=1}^{m} w_{k,j}\, x_j \tag{4.1}$$

$$y_k = \varphi(u_k + b_k) \tag{4.2}$$

where $\{x_1 \dots x_m\}$ are the input signals and $\{w_1 \dots w_{j,m}\}$ are the synaptic weights of neuron, u_k is the linear combiner output, φ(.) is the activation function and y_k is the output.

Typically y_k is normalized to [-1,1] or [0,1]. The bias is different than zero so that the neurons achieve the performance we want and classify the data correctly because in the case that there is not bias and one of the inputs are null or zero the activation function with have poor predictive performance.

As depicts the example in figure 4.3 an output $y_k = \varphi(u_k)$ without bias the neural network is not capable to classify the data with the linear equation causing poor performance. Hence, with a bias the network has a degree of freedom to adjust the position of the line and classify the data better. The result of better data classification is a better predictive model.

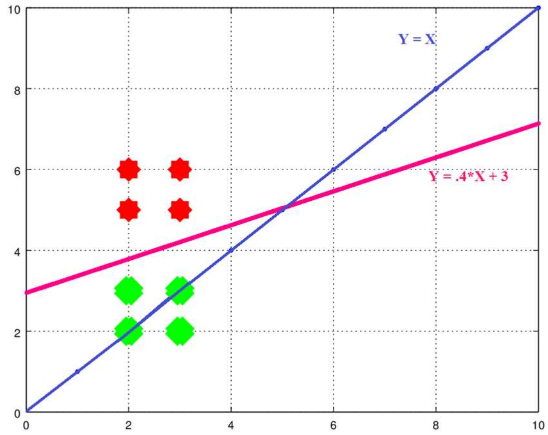


Figure 4.3 - Classification from a neural network with bias, red line and without bias, blue line.

Types of activation functions for hiden nodes :

Activation functions defines the output of the neuron nodes in terms to the induced input. The three basic types are shown in figure 4.4 (a),(b),(c) and the more modern and most used types are shown in figure 4.4 (d),(e),(f),(g).Their mathematical expressions are shown below:

1. **Heaviside function:** is a threshold Function also called described in fig.10a:

$$\varphi(u) = \begin{cases} 1 & if\, u \geq 0 \\ 0 & if\, u < 0 \end{cases} \tag{4.3}$$

2. **Piecewise-Linear Function:** described in fig.10b:

$$\varphi(u) = \begin{cases} 1, & if\, u \geq +\frac{1}{2} \\ u\,, & +\frac{1}{2} > u > -\frac{1}{2} \\ 0, & u \leq\, -\frac{1}{2} \end{cases} \tag{4.4}$$

3. **Sigmoid function:** Sigmoid is the most common form of activation function because it's S-form is ideal for linear and nonlinear behavior and is differentiable, which is very important feature for a neural network. Used as activation function while building neural networks. In mathematical definition way of saying the sigmoid function take any range real number and returns the output value which falls in the range of 0 to 1. Based on the convention we can expect the output value in the range of -1 to 1. The sigmoid function produces the curve which will be in the Shape "S." These curves used in the statistics too. With the cumulative distribution function (The output will range from 0 to 1), described in figure 4.4 c:

$$\varphi(u) = \frac{1}{1+\exp(-a\,u)} \tag{4.5}$$

Where by changing the parameter a we take sigmoid functions with different slope.

4. **Threshold functions:** Also sometimes it is important to have an anti-symmetric activation function so the range from [-1,1] makes a function like that appropriate.

   The threshold functions are now defined as:

$$\varphi(u) = \begin{cases} 1 & if\ u > 0 \\ 0 & if\ u = 0 \\ -1 & if\ u < 0 \end{cases} \tag{4.6}$$

5. **Hyperbolic tangent function:**

$$\varphi(u) = \tanh(u) \tag{4.7}$$

6. **Rectified Linear Unit (ReLU) type of transfer functions:**

   a. **ReLU [51]:**

$$f(x) = max(0, x) \tag{4.8}$$

   b. **Leaky ReLU [52]:**

$$f(x) = \begin{cases} x, & if\ x \geq 0 \\ 0{,}01x, & if\ x \leq 0 \end{cases} \tag{4.9}$$

   Parametric ReLU (PReLU) [53]:

$$f(x) = \begin{cases} x, & if\ x \geq 0 \\ ax, & if\ x \leq 0 \end{cases} \tag{4.10}$$

   where $a \leq 1$

   c. **Stochastic ReLU [54]:**

$$f(x) = \begin{cases} x_{ji} & if\ x > 0 \\ a_{ji}\ x_{ji}, & otherwise \end{cases} \tag{4.11}$$

   where $a_{ji} \sim U(l, u), i < u\ \ and\ i, u\ \in [0,1)$

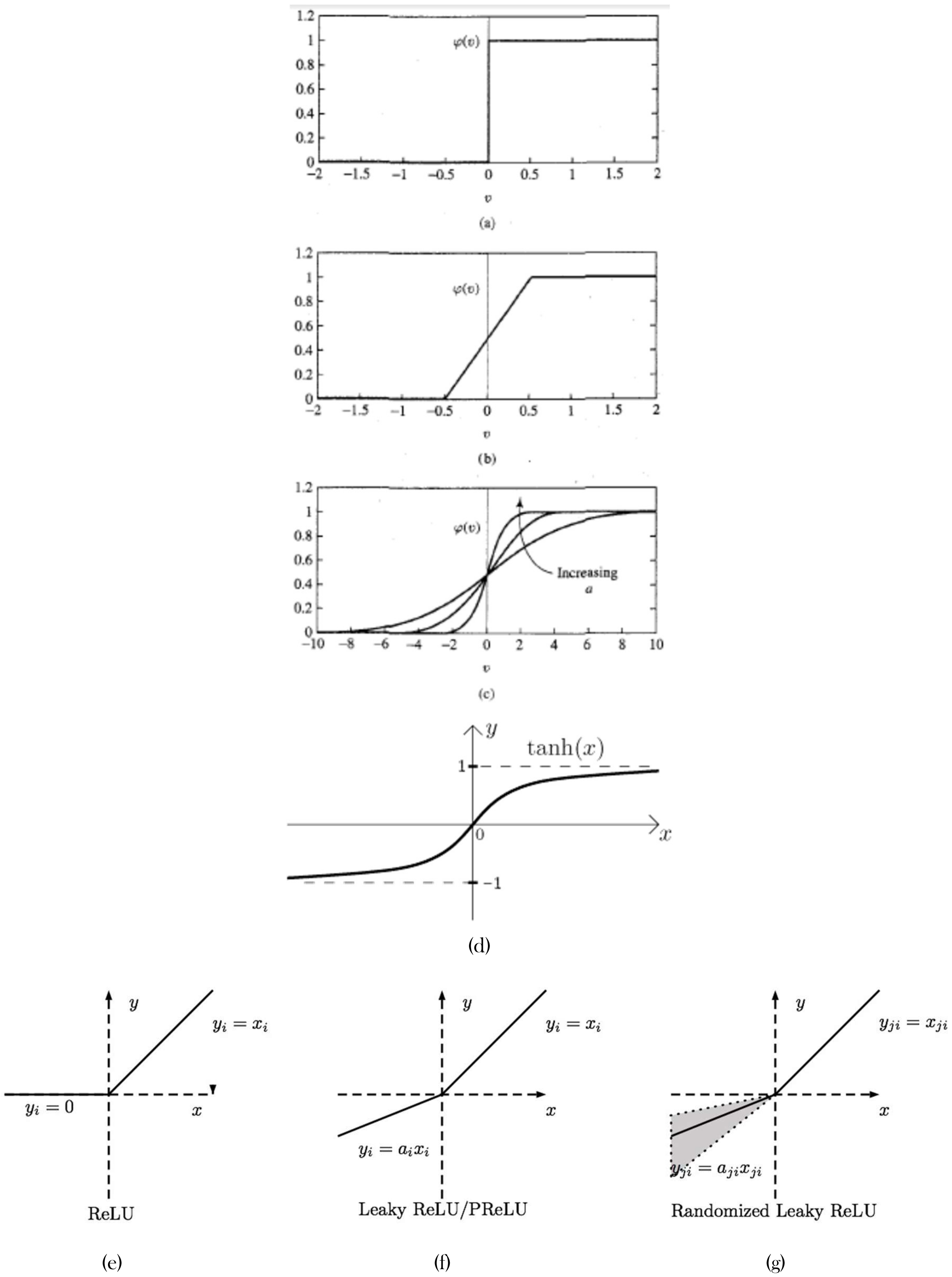


Figure 4.4 – a) Threshold function, b) Piece-wise linear function, c) Sigmoid function for varying slope parameter a, [26], d) Tanh sigmoid function, e) Standard ReLU, f) Leaky ReLU/PreLU, g) Randmized ReLU

4.2 Neural Network Architectures

1. *Single – Layer Feed-forward Networks*

   In a layered neural network the neurons are organized in the form of layers. In the simplest form of a layered network we have an input layer of source nodes that projects onto an output layer of neurons, but not vice versa. Such a network is called single-layer network and is shown in figure 4.5. With the "single - layer" output layer referring to the output layer, without count the input layer because no computation is performed there.

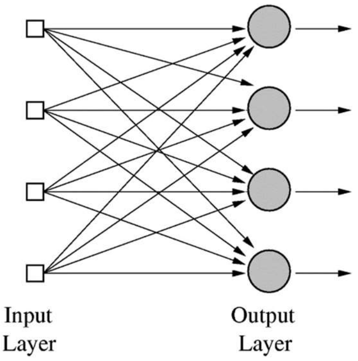


Figure 4.5 - Single layer network [26].

2. *Multilayer Feed-forward Networks*

   This architecture distinguishes itself by placing one or more hidden layers as shows the figure 4.6. By adding one or more hidden layers the network is able to calculate higher-order statistics leading to a better global perspective. Moreover, the hidden layers are valuable when the size of the input layer is large so the data are many. Determining the optimal number of hidden nodes is basically an unsolved problem in neural network research, and so trial and error is generally used.

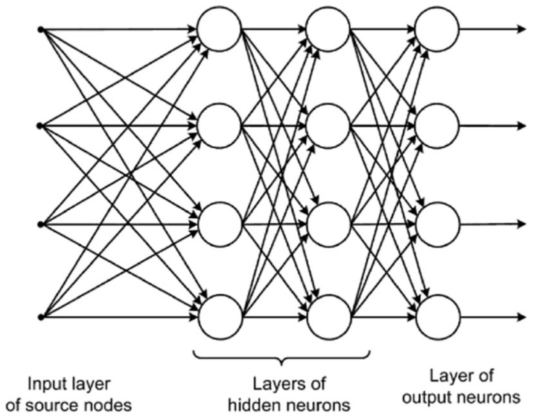


Figure 4.6 – Multilayer network with two hidden layers [26].

3. *Recurrent Network*

   This architecture has the structure of a multilayer network but it has at least one feedback loop. The output of a neuron is fed back to the input of the current hidden layer. This architecture has better learning capability but non linear dynamic behavior because the feedback loop is delayed in order to achieve concurrency between inputs arrivals. Also the Hopfield networks which is a case of recurrent networks are a way to build associative memory holding the most important past information and rejecting the least important.

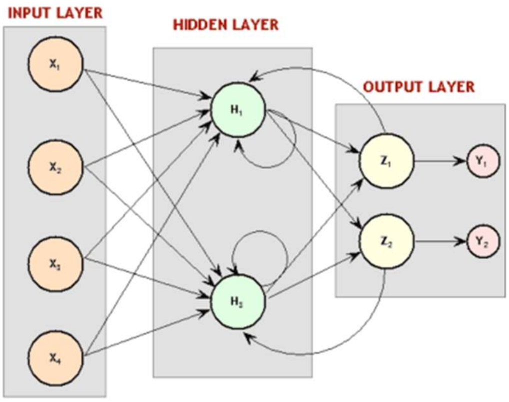


Figure 4.7 – Recurrent neural network with two hidden nodes.

4.3 Neural Networks (NN) implementations

The NN can be used to classify data and make predictions. Behind the scenes, a neural network can be thought of as a complicated mathematical function that has various constants called weights and biases, which must be determined. Training a neural network is the process of finding a set of weights and bias values so that computed outputs closely match the known outputs for a collection of training data items. Once a set of good weights and bias values have been found, the resulting neural network model can make predictions on new data with unknown output values. To optimize a neural network used the technique *back-propagation* method with the used of different types of gradient descend algorithms.The fastest algorithm is the *conjugate gradient descend* because for each iteration it does not use any previous information so the optimization does not stuck in a local minimum. The algorithm is very complex to write it so for more details there are many advanced mathematical books on linear algebra which explain it and the author's publication as well. [55]

Classification types: NN vs Classical Pattern Recognition:

The fundamental difference between a neural network and a classical pattern recognition solution is that in pattern recognition firstly we build a mathematical model observing the training data, validating the model after and then building the final design based on the model. The design of a neural network is based directly on data, the model design is a dynamic process and follows only the trend of the data and no user's observations or other mathematical predefined models. As a consequence the data act upon the model by themselves.

The major task for a neural network is to learn a model of the environment in which it is embedded and to maintain the model sufficiently consistent with the real world so as to achieve the specified goals of the application of interest. Knowledge of the world consists of two kinds of information:

1. The prior information, such as already known information.
2. Observations from sensors, where most of the times are noisy or being subject to errors due to system imperfections. These observations are the examples from which the learning process will take place and are the input signals.

A set of input-output pairs, with each pair consisting of an input signal and the corresponding desired result, is referred to as a set of training data. The observations can be labeled or unlabeled and are part of the inductive learning. Inductive learning is a central task in data mining and neural networks since building descriptive models of a collection of data provides one way of gaining insight into it. Such models can be learned by either supervised or unsupervised methods or semi-supervised depending on the nature of the problem.

In supervised learning the learner is given a set of instances of the form $\langle \vec{x}, y \rangle$, $\vec{x}$ is a vector of values that represent features thought to be relevant to determining the result y. The goal in supervised learning is to induce a general mapping from $\vec{x}$ vectors to y values. The learner must build a model $\hat{y} \; = f(\vec{x})$ of the unknown function f that allows it to predict y values for new unseen examples. Some of the methods which belong in this category are the error correction learning techniques such as ANN, ANFIS etc., stochastic learning techniques as stochastic gradient descend, and hardwired systems.

In unsupervised learning the learner is also given a set of training examples but each instance consists only of the $\vec{x}$ part it does not include the y value. The goal in unsupervised learning is to build a model that accounts for regularities in the training set. Some of the methods which belong in category are the Hebbian algorithm, the differential Hebbian and the Min-Max algorithm.

In semi-supervised learning there are data with label and others without labels because the cost or skills to find the labels is infeasible or they missed. The first goal is to map the correct labels for the unlabeled data from inductions from the labeled data and secondly to create a better model than supervised or unsupervised learning by using the new larger dataset.

4.4 Back-propagation Algorithm

The advantage of Neural Networks from other data mining techniques which do not have a Neural Network Architecture is their learning capacity. By taking the derivative of the cost function (error) with respect to the network parameters and changing them with gradient descent algorithms a neural network achieves the minimum cost value. The most common algorithms for this are the *Back-propagation algorithms* which based on gradient descent to compute the gradients. In *back-propagation*, gradients must be computed in the opposite direction of computing output values. Although there are many good references available that explain the interesting mathematics of back-propagation training, there are very few resources that describe the practical issues involved in implementing back-propagation training. Training with *back-propagation* is an iterative process. The *back-propagation* algorithm is used to search for weights and bias values that generate neural network outputs that most closely match the output values in the training data. The behavior of the *back-propagation* algorithm depends in part on the values of a learning rate. When using the *back-propagation* training algorithm, using the *online approach* is better than using the *batch approach.* Follows a quick view on mathematics of how back-propagation works [26].

For a neural network the weights hold the memory of the model and their values adjust the parameters of the output, so with the bias together, are very important for the learning process. Ideally the network become more knowledgeable about its environment after each iteration, also named epoch, of the learning process.

To adjust the knowledge inside the neural network is, however, complicated. The feed-forward neural network as depicted in figure 4.7 produces an output signal $y_k\,(n)$ at the nth discrete time of the neuron k. This output compared to the desired output or target output or desired response as its named, denoted by $d_k(n)$. Consequently the error denoted by $e_k(n)$ is produced:

$$e_k(n) = d_k(n) - y_k\,(n) \tag{4.12}$$

The error applies the correct adjustment to the synaptic weights of neuron k to make the output signal $y_k\,(n)$ come closer to the desired output $d_k(n)$. The process of adjustment is a step by step manner with the subject goal of minimization of a Cost Function or index of performance, defined as:

$$E(n) = \frac{1}{2} e_k^2(n) \tag{4.13}$$

where $E(n)$ is the instantaneous value of the error energy.

The total error energy for the entire network is obtained by summing all the instantaneous values of error energy for all neurons in the output layer:

$$E(n) = \frac{1}{2} \sum_1^C e_k^2(n) \tag{4.14}$$

where C is the number of all neurons in the output layer.

The average squared error is calculated by summing all the total energy for the entire network for all the N attributes in training dataset and normalizing it dividing by N:

$$E_{avg} = \frac{1}{N} \sum_{n=1}^N E(n) \tag{4.15}$$

The objective of the learning process is to adjust the free parameters (weights and bias) of the network. The E_{avg} is a function of all the free parameters of the network.

As a consequence for a given training set the E_{avg} represents the total cost function as a measure of the learning performance. The final goal is to minimize the E_{avg} over the entire training set.

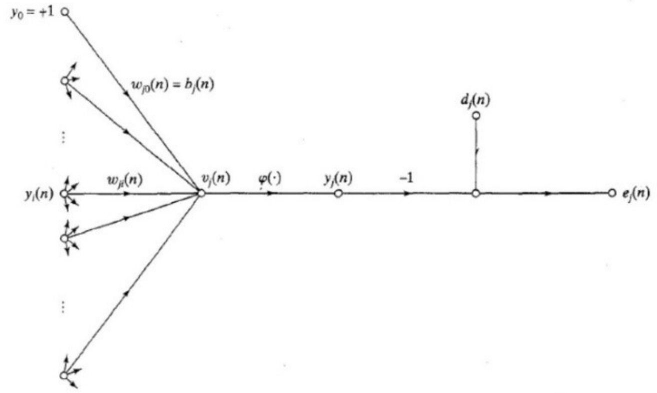


Figure 4.7 - Signal flow graph with output neuron j

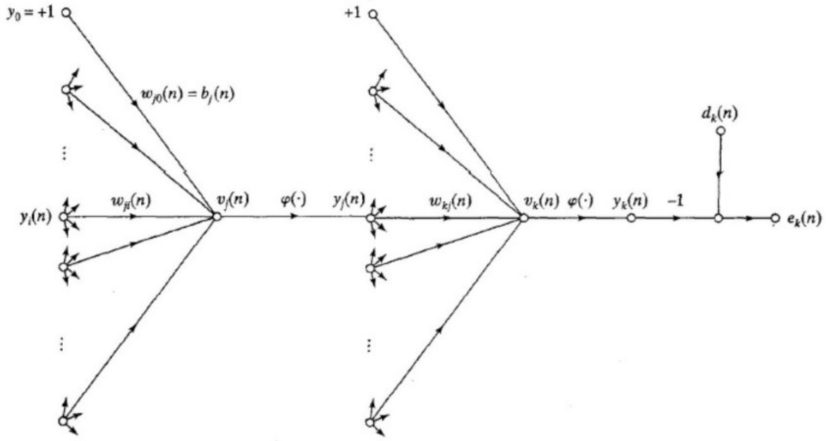


Figure 4.8 – Signal flow from hidden neuron j to output neuron k.

Case neuron j is output:

A neuron is fed by a number of input signals produced by a layer of neurons on its left figure 4.7. The induced local field produced at the input of the activation function associated with neuron j is therefore:

$$u_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_j(n) \qquad (4.16)$$

where *m* is the total number of inputs (excluding the bias) applied to neuron *j*,

w_{ji}(n) is the ith weight that goes as input on jth neuron.

The output from the activation function in:

$$y = \varphi_j(u_j(n)) \tag{4.17}$$

The back-propagation algorithm applies a correction $\Delta w_{ji}\ (n)$ to the weight w_{ji}(n). With use of (4.17) the partial derivative of $\frac{\partial E(n)}{\partial w_{ji}(n)}$

represents the sensitivity factor determining the path of search in weight space for the synaptic weight w_{ji}. According to the chain rule the gradient is:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial u_j(n)} \frac{\partial u_j(n)}{\partial w_{ji}(n)} \tag{4.14}$$

Differentiating the equations: (4.12) with respect to $y_j(n)$, (4.14) with respect to $e_j(n)$, (4.17) with respect to $u_j(n)$, (4.16) with respect to $w_{ij}(n)$ we have sequentially:

$$\frac{de_j(n)}{dy_j(n)} = -1 \tag{4.15}$$

$$\frac{dE(n)}{de_j(n)} = e_j(n) \tag{4.16}$$

$$\frac{dy_j(n)}{du_j(i)} = \varphi_j'(u_j(n)) \tag{4.17}$$

$$\frac{du_j(n)}{dw_{ji}(n)} = y_i(n) \tag{4.18}$$

Equation (4.18) with the use of (4.19) (4.20) (4.21) (4.22) becomes:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -e_j(n)\ \varphi_j'(u_j(n))\ y_j(n) \tag{4.19}$$

The correction weight is: $\Delta w_{ji}(n) = -\eta\ \frac{\partial E(n)}{\partial w_{ji}(n)}$ (4.20)

where η is the step size also named learning-rate parameter.

Finally the local gradient is:

$$\delta_j(n) = -\frac{dE(n)}{du_j(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial u_j(n)} = -e_j(n)\ \varphi_j'(u_j(n)) \tag{4.21}$$

So the type of activation function is very important to the error correction.

Finally the equation (4.24) with the use of (4.23) and (4.25) becomes:

$$\Delta w_{ji}\ (n) = \eta\ \delta_j(n)\ y_i(n) \tag{4.22}$$

which describes the weight adjustment of each layer which optimizes the network.

$$\text{Correction weight} = \begin{pmatrix}\text{learning}\\ \text{parameter}\end{pmatrix} * \begin{pmatrix}\text{local}\\ \text{gradient}\end{pmatrix} * \begin{pmatrix}\text{input signal}\\ \text{of neuron j}\end{pmatrix}$$

$$\eta = \frac{\kappa}{\sqrt{\sum w_{ji}(n)\left(\frac{\partial \mathrm{E(n)}}{\partial w_{ji}(n)}\right)^2}} \qquad (4.23)$$

where k is the step size, which can be changed in order to accelerate the convergence rate in adaptive networks.

Case neuron *j* is hidden:

The case of j as hidden neuron (figure 4.8) is different because there is not a desired signal $d_j(n)$ in the hidden neurons which are behind the final neuron (k) so the error $e_j(n)$ in hidden neuron has to be computed by the equation (4.10). The process has many similarities with the previous and will skip the formulas.

The final result is the *back propagation* formula for the local gradient for output neuron:

$$\delta_j(n) = \varphi_j'\left(u_j(n)\right) \sum_k \delta_\kappa(n) w_{kj}(n) \qquad (4.24)$$

where

j is hidden neuron,

k is an output node .

The correction weight has the same formula format as the equation (4.22):

$$\Delta w_{ji}\,(n) = \eta\ \delta_j(n)\ y_k(n) \qquad (4.25)$$

or:

$$\text{Correction weight} = \left(\begin{matrix}\text{learning}\\ \text{parameter}\end{matrix}\right) * \left(\begin{matrix}\text{local}\\ \text{gradient}\end{matrix}\right) * \left(\begin{matrix}\text{input signal}\\ \text{of neuron j}\end{matrix}\right)$$

The learning rate is the same as in equation (4.23).

4.5 Batch and Online Neural Network Training techniques

There are two different techniques for training a neural network: *batch* and *online*, (*batch* is by far the most common algorithm). In the very early days of neural network, *batch* training was suspected by many researchers to be theoretically superior to *online* training. However, by the mid- to late-1990s, it became quite clear that when using the *back-propagation* algorithm, *online* training leads to a better neural network model in most situations. Understanding their similarities and differences is important in order to be able to create accurate prediction systems. The approaches are similar but can produce very different results.

The general consensus among neural network researchers is that when using the *back-propagation* training algorithm, using the *online* approach is better than using the *batch* approach but more complex.

In *online training (incremental)*, weights and bias values are adjusted for every training item based on the difference between computed outputs and the training data target outputs. In incremental learning, the system learns "incrementally" with an updating algorithm, and the system's knowledge is updated every time. Used in time series where the data are a function of time or the data are too much for the memory to proceed them all at once. The complexity in online is that many nodes proceed the data when others may do not have data yet because of delay, so the first used nodes should not forget its existing knowledge. Online learning also is more complex because the updating should consider the delays between old and new values to nodes so it should give smaller weights (strength) to the older data and bigger weight to new arrivals.

In *batch training* the adjustment *local gradient* values are accumulated over all training items, to give an aggregate set of *deltas*, and then the aggregated local gradients are applied to each weight and bias. In the *Bach* based method, instead of *incremental learning*, it takes less time to process the data that comes in bulk, and it's updated once it is processed. On Batch learning the update takes place after each iteration (epoch) and after all data are inserted, it is faster that online learning which update takes place after each input-output pair is available on the nodes. Online learning also is more complex because the updating should consider the delays between old and new values to nodes so it should give smaller weights (strength) to the older data and bigger weight to new arrivals.

Batch and *online training* can be used with any kind of training algorithm. Depending on the application, each method might be more appropriate.

The next four fields are specific to *batch* training. (i) hold the accumulated delta values (that is, small amounts that will be added) for the input-to-hidden weights, (ii) hold accumulated delta values for the hidden biases, (iii) hold the hidden-to-output weights, and (iv) hold the output biases.

The heart of *batch training* is in method *Train*. The key to *method Train* is the call to helper Compute and Accumulate Deltas. After that call completes, the accumulated delta values (based on all training data) will be stored. Because of deltas are accumulated over all training items the order in which training data is processed doesn't matter, as opposed to online training where it's critically important to visit items in a random order.

Consider α one of the premise parameters among {pi, qi, ri} which works exactly like the weights.

$$\frac{\partial \mathrm{E_p}}{\partial \mathrm{a}_{\mathrm{i,p}}^{\mathrm{k}}} = \sum_{O^* \in S} \frac{\partial \mathrm{E}p}{\partial \mathrm{O}^*} \frac{\partial \mathrm{O}^*}{\partial \mathrm{a}} \tag{4.26}$$

where S is the set of nodes whose outputs depend on w.

The derivative of the overall error with respect to α is

$$\frac{\partial \mathrm{E_p}}{\partial \mathrm{a}_{\mathrm{i,p}}^{\mathrm{k}}} = \sum_{p=1}^{p} \frac{\partial \mathrm{E_p}}{\partial \mathrm{a}} \tag{4.27}$$

Equation (4.27) n is used on Batch learning for hybrid algorithm. On Batch learning the update it is faster that online learning which using the equation (4.26) update takes place after each input-output pair is available on the nodes. The learning rate on equation (4.23) is the same in batch and online learning substituting the weight with the parameter *a.*

4.6 Hybrid-Learning Algorithm: Forward and Backward pass

The hybrid algorithm applies two passes [13], *forward* and *back-word pass*:

1. In *forward pass* signals reach the Level 5 and using sequential least square method calculates the consequent parameters while the kept fixed.
2. In the *backward pass* a back-propagation algorithm propagates the derivative of the error from output layer until layer 2 the then updates the premise parameters by the gradient descend while the consequent parameters kept fixes. Consequent parameters act like the weight in ANN. The researchers suggest that when using the back-propagation training algorithm, using the *Batch approach* is more simple to applied than online learning.

Forward Pass algorithm:

The output from the layer five is:

$$O_i^5 = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2$$

$$= \overline{w}_1 f_1 + \overline{w}_2 f_2$$

$$= (\overline{w}_1 x)p_1 + (\overline{w}_1 y)q_1 + (\overline{w}_1)r_1 +$$

$$(\overline{w}_2 x)p_2 + (\overline{w}_2 y)q_2 + (\overline{w}_2)r_2$$

$$\Rightarrow \; y = f(i,s) \tag{4.28}$$

where {i} is the vector of input variables,

S is the set of parameters.

If there is a function H such that the composite function $H \circ f$ is linear in some of the elements of S, then these elements can be identified by the least-squares method. If the parameter set S can be divided into two sets S1 and S2, which S1 is set of premise parameters and S2 is set of consequent parameters we have $S = S1 \oplus S2$, (where $\oplus$ represents the direct sum) such that $H \circ f$ is linear in the elements of S2, after applying H to Equation (4.28), we have:

$$H \circ O = H \circ f(i, S) \tag{4.29}$$

which is linear in the elements of S2. So the equation (4.29) can be written as the classic linear equation:

$$AX = Y \tag{4.30}$$

where X is an unknown parameter vector whose elements are parameters in S2 and predicted output.

After substituting p training data in equation (4.29) in f then a matrix equation is obtained. Let S2=M, and dimensions of t is M×1 parameter vector, A is p×M matrix and Y is p×1 output vector. This is a standard linear least-squares problem, and the best solution for X, which minimizes $||A\,X - Y||^2$ is the least squares estimator (LSE) X^*.

$$X^* = (A^T A)^{-1} A^T y \tag{4.31}$$

This is computational expensive so Sequential Least Squares Estimator compute the LSE of X. The goal is to approximate the following equation which constructed using the covariance of the data and has the role of slope λ as in a classic linear equation $y=\lambda x$, so the trend of the data defines the slope of the line:

$$X_{i+1} = X_i + S_{i+1} a_{i+1} \left(b_{i+1}^T - a_{i+1}^T\, X_i\right) \tag{4.32}$$

where a^T, b^T are the elements of matrix A and Y.

The Covariance matrix is calculated from:

$$S_{i+1} = S_i - \frac{S_i\, a_{i+1}\, a_{i+1}^T\, S_i}{1 + a_{i+1}^T\, S_i\, a_{i+1}}\ , i = 0,1, \ldots P-1 \tag{4.33}$$

Backward pass algorithm:

In the back propagation the consequents parameters are constant. Premise parameters act like the weight in ANN. Backpropagation discussed in detail in section 4.4.

CHAPTER FIVE

Adaptive Neuro Fuzzy Inference System (ANFIS): Concepts and Terminology

5.1 Fuzzy Logic and Fuzzy Sets

In physical world we do not have precise criteria for the object's membership. The "the class of all real number greater than 1" arises ambiguity. Professor Lotfi A. Zadeh in 1960 inspired by this and created the concept of fuzzy set, a class with a continuum of grades of membership. This framework provides a natural way of dealing with problems that arise imprecision criteria rather than random variables. Later Mamdani and Sugeno created inference systems based on fuzzy set theory.

Fuzzy sets: Let *X* be a space of points, with a generic element of X denoted as by *x*, thus *X = {x}.* A fuzzy set A in X is characterized by a *membership function (M.F.)* which associates each point in X space with a real number in the interval [0,1].The value of $f_A(x)$ representing the grade of membership of *x* in class A, 1 is fully membership 0 no membership. The nearer the value of $f_A(x)$ to unity the higher the grade of membership and belongs more in this class rather than other classes avoiding the overlapping.

The modeling of the systems which have uncertainty is very complicated and many times impossible with the classical mathematics and the differential equations. The use of the Fuzzy inference systems also named as FIS, was that through IF-THEN statements can apply rules which are comprehensive from the humans and apply inference logic which does not require Boolean logic where every other state considered as a false and unstable for the digital system. Although the FIS can handle the intermediate states it cannot adjust itself based on the different inputs so it's parameters stay fixed and the output as well. From the other aspect neural networks have the ability to adapt in the new data and learn by changing the parameters. Although the capability to adapt the multiply layers and makes the process hidden and too complicated for the human to know what is happening and act upon that. The solution to the two problems came from the combination of FIS and Neural Networks which the FIS transform the human knowledge into rules and the NN technique adapts the membership functions and this systems is called Adaptive neuro-fuzzy inference systems following the Takagi-Sugeno (TS) model using supervised learning algorithms.

5.2 ANFIS Architecture

The system [13] consists five layers, for simplicity the description will be based on two inputs *x* and *y* where the FIS produces a single final output *f* and has two IF-THEN rules, as shows figure 5.1.

Rule 1 : IF x is A1 and y is B1 THEN f1 = p1x+q1y+r1
Rule 2 : If x is A2 and y is B2 THEN f2 = p2x+q2y+r2

Where {A1,B1}, {A2,B2} are the fuzzy sets also named membership functions and {p1,q1,r1}, {p2,q2,r2} are linear parameters part of the consequent phase of the Takagi-Sugeno fuzzy inference system. The first and fourth layer have the adaptable parameters so the nodes are adaptive as well while the other two, three and five layer contain fixed nodes.

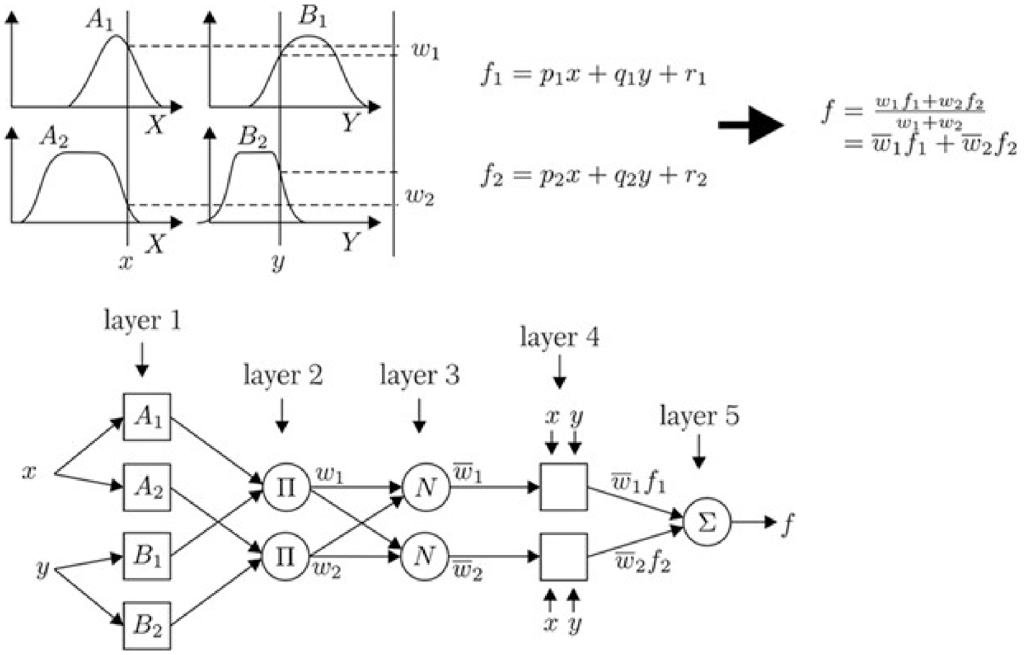


Figure 5.1 – Up is a sugeno type 3 output, down is the Anfis type-3 architecture [13].Sugenos's output rules imported in layer 4 of Anfis.

Layer 1

At the first layer is the fuzzification. Every node-i has a membership function Ai, Bi and an output $O_i^1 \ = \mu_{A_i}(x),\ O_i^1 = \mu_{B_i}(y)\ \ i = 1,2$ where *x* and are inputs, Ai, Bi linguistic labels (low, medium, high etc.) of fuzzy sets characterized by membership functions and $O_{1,i}$ the outputs from the nodes. A MF has the role to calculate with a subjective way the degree of membership of the input x for a fuzzy set A_i. MFs may have different forms but they should be piecewise differentiable, continuous and differentiable, so that the optimization algorithms do not fail. They take values from [0,1].

As a consequence the bell shaped curves like Gaussian are appropriate, also trapezoid or triangular. More often are used:

$$\mu_{A_i}(x) = \frac{1}{1+\left|\frac{x-c_i}{a_i}\right|^{2b_i}} \qquad (5.1)$$

or

$$\mu_{A_i}(x) = \exp(-\left(\frac{x-c_i}{2a_i}\right)^2) \qquad (5.2)$$

where

{ai,bi,ci} are called premise parameters which they can change the MF shapes,

c_i sets the centers of the MF, a_i sets the width of the MF.

Layer 2

Every layer of the network receives the signals applies a T norm operator which is AND or OR and produces the output signal like the synapses in neurons. This is named power or weight of the rule and this goes to the Layer 3.

$$w_i = \mu_{\mathrm{A_i}}(x) x\, \mu_{\mathrm{B_i}}(x) \qquad (5.3)$$

where i=1,2...N inputs

Layer 3

Every node calculates the normalized received power signal and sends it to Level 4. As normalized power of i-th node is the ratio of the power of this node by the sum of the powers from the rules. In this case there are two rules:

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, i=1,2 \qquad (5.4)$$

Layer 4

After receiving the normalized power from level 3 the next step is for every i-th node to produce a function:

$$O_i^4 = \bar{w} f_i = \bar{w}_i (p_i x + q_i y + r_i) \qquad (5.5)$$

where $\{p_i, q_i, r_i\}$ are the consequent parameters.

Layer 5

It has a single node which computes the overall output as the summation of all incoming signals:

$$O_i^5 = \sum_i \bar{w}_i f_i = \frac{\sum \bar{w}_i f_i}{\sum \bar{w}_i} \qquad (5.6)$$

The Adaptive neuro-fuzzy inference system is a type 3 **FIS**, every **MF** is associated with every input with final output a linear combination.

CHAPTER SIX

Experimental Studies with ANFIS

6.1 ANFIS Process

ANFIS in provides three ways to create the rules that models the data behavior: genfis1, genfis2, genfis3. The more the rules the betters the result of cost function unless there is overfitting. Overfitting occurs if there are many MF and there are a few samples so the free parameters are memorizing the previous values. Hence, different methods, parameters and datasets should be applied to find the minimum RMSE for the testing dataset. The flowchart for adjusting the ANFIS is in figure 6.1:

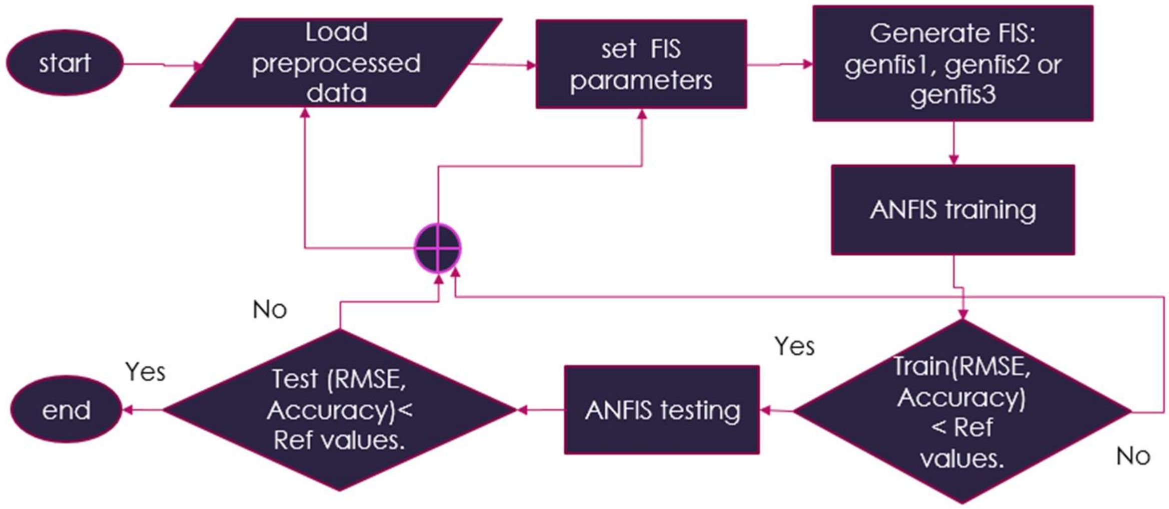

start
Load preprocessed data
set FIS parameters
Generate FIS: genfis1, genfis2 or genfis3
ANFIS training
Train(RMSE, Accuracy) < Ref values.
Yes
No
ANFIS testing
Test (RMSE, Accuracy)< Ref values.
No
Yes
end

Figure 6.1 - ANFIS flowchart on how to tune the ANFIS.

6.1.1 ANFIS as Classifier

We use 3 classes to define 3 levels of risk. In UCI database 1,2,3,4 is the number of major main arteries in the heart with more than 50% blockage and 0 the absence. Previous researches ignore the fact that 1, 2, 3, 4 is the number of major main arteries because there is not description in the database. As a result, the danger can be classified better, binary classification is too general but 5 classes are too many to be used with so a few data samples. The table 6.1 shows the grouping of the classes.

Table 6.1 – Thesis proposed grouping for the classes of Cleveland dataset for the CHD predicton.

Grouping Classes	{0}	{1,2}	{3,4}
Risk levels	Class Absence: 0	Class Medium High Risk: 1	Class Very High Risk: 2

6.1.2 Measurements of Performance

After developing a classifier, the performance measuring follows. Performance is measured calculating the accuracy on the training data but more important on the accuracy how well the model generalizes which is the accuracy from the testing data. Overfitting or underfitting are common cases for machine learning models where in both cases the performance drops because of overextension of learning parameters in the first case and on few learning parameters in the second case. The ANFIS algorithm generates an output FIS for training, validating, and testing data. Training and checking (also called validation) RMSE is calculated for each data sample. The RMSE compares the predicted output of the FIS, $\boldsymbol{y_{predicted}}$ and the actual output value $\boldsymbol{y_{actual}}$. N represent the number of samples. The RMSE for training can be expressed as:

$$RMSE = \sqrt{\frac{\sum_{K=1}^{N}\left(y_{predicted,k} - y_{actual,k}\right)^2}{N}} \qquad (6.1)$$

RMSE is not a panacea to deduce how well the model fits the data, it is used because it penalizes the large errors more than the small errors. A small value of RMSE should be considered good with respect to the range of the value of $y_{classifier,k}$, also named as dependent variable in literature, RMSE also used as the cost function in machine learning algorithms.

Validation data prevents the learning from overfitting: When the training RMSE drops and the validation RMSE start to increase, significant, then overfitting begins to happen so the training stops. For under fitting we need more data or/and to optimize the adjustments of the system. For overfitting we need to “loose” the optimization of the adjustments of the system so that to avoid overextension of it’s normal capabilities.

The RMSE gives a more accurate value of the error between a model (output of FIS) and observed data (training/validation data output value). There are statistical properties such as variance and standard deviation that makes RMSE a desirable measurement. It is desirable to have a RMSE decrease or converge as the number of iterations increase.

The RMSE of the checking data is used to prevent overfitting. Overfitting occurs when the RMSE of the checking data increases. It is a result of fitting the fuzzy system to the training data so well that it no longer fits the testing data effectively and this leads to a loss of generality. The ANFIS algorithm chooses model parameters associated with the minimum checking error prior to overfitting (if it exists). Once RMSE of the training data is shown to decrease as the number of iterations increases and overfitting is eliminated, evaluation of the ANFIS is made. Generally the accuracy of a classifier can be expressed as:

$$Accuracy = \frac{tp+tn}{tp+tn+fp+f} \qquad (6.2)$$

tp = true positive, tn = true negative, fp = false positive, fn = false negative

The confusion matrix calculates the accuracy for two classes or many classes. Table 6.2 is the confusion matrix for a two class classifier, table 6.3 is the confusion matrix for many classes

Table 6.2 - Confusion matrix for a two class classifier.

Predicted Class Actual Class	Positive	Negative
Positive	tp	tp
Negative	fn	tn

The term accuracy is the ratio of sum of instances that were correctly classified to total number of instances present:

Table 6.3 - Confusion matrix for many classes

Predicted Classes Actual classes	Class 1	Class 2	Class N	Total
Class 1	Accuracy 1	false	false	
Classe2	false	Accuracy 2	false	
Class N	false	false	Accuracy N	
				$\sum_{1}^{N} Accuracy_i$

For the ANFIS we applied the formula of accuracy:

$$Accuracy = \frac{tp+tn}{tp+tn+fp+f} = \frac{round(y_{predicted})=y_{actual}}{N} \qquad (6.3)$$

Where $y_{predicted}$ rounded to the nearest integer as y_{actual} is an integer number.

For classifiers there are also the precision, fp-rate, tp-rate also named as sensitivity or recall, f-measure, kappa-statistic [44]

The precision is the ratio of the predicted positive instances that were correct:

$$precision = \frac{tp}{tp+f} \qquad (6.4)$$

The recall or tp-rate or sensitivity or True Positive Rate (TPR) is the ratio of positive instances that were correctly classified as positive:

$$recall = tp_{recall} = \frac{tp}{tp+fn} \qquad (6.5)$$

The fp-rate or specificity or True Negative Rate (TNR) is the ratio of negative instances that were incorrectly classified as positive:

$$fp_{rate} = \frac{fp}{fp+t} \qquad (6.6)$$

In some scenarios high precision may be more important, while in other scenarios high recall may be more significant. In most types we try to improve both values. The combined form of these values is called the f-measure or f-score or F1 score:

$$f_{measure} = \frac{2*precision*recall}{precision+reca} \qquad (6.7)$$

Kappa statistic measures the inter-rater agreement for categorical data. Is considered to be a measure of reliability among different raters or judges.

$$k = \frac{prob_o-prob_e}{1-pro\ _e} \qquad (6.8)$$

where $prob_o$ is the probability of observed agreements among raters and $prob_e$ is the expected probability of agreements by chance. If $k = 1$ the raters have completely agreed with each other's decision. If $k = 0$ then the judges or raters are not agreed. A good measurement is k>0.7.

With ANFIS we use only the accuracy because in literature it is done that way for the same type of problems.

6.1.3 Cross Validation Methods

K-fold cross validation method divides the whole dataset into k-folds and takes the k-1 folds as training set and the rest one as test set. The process repeats until every fold is used as testing set and the rest *k-1* as training. With k fold cross validation some data may be processed only one time, also it explores a few ways which the data could have been partitioned but it is a legit method for model validation. Mostly 10 fold or 5 folds are used, depending on the executions time. The final result is the mean value from the k repetitions.

In Cross Validation (CV) with Holdout the dataset splits between training and testing sets randomly by a percentage of 80%/20% usually. Every simulation produces uncertain results so Monte Carlo simulation can be applied in order to find the mean value. Monte Carlo is hundreds or thousands simulations on random samples. Holdout CV with Monte Carlo is also named Bootstrap validation, it is the most accurate method but requires much more time to execute than k-fold CV.

We applied 5-fold CV in ANFIS, while ANFIS with optimization on its parameters and the large number of many dimensional inputs make the process of Monte Carlo with Holdout CV extremely slow for use.

6.2 Methods of building FIS Structures for ANFIS

Fuzzy Inference System should be created in order to provide the membership functions, the rules and the five levels of architecture. There are three ways for building a FIS structure of ANFIS.

1. **ANFIS with Grid Partitioning to create the FIS**

   In Matlab a FIS structure from partitioning is created by:
   fismat = genfis1(data, numMFs, inmftype, outmftype)

Genfis1 implements grid partitioning where the dataset partitioned and the rules inducted from the partitions. From experiments with the dataset found that this approach is good for small dimensional datasets, less than 7 columns and with a few number of MFs otherwise the execution cannot be done with a normal computer because it takes huge amount of time. As a result of the above experiments genfis1 rejected.

2. ANFIS with Subtractive Clustering to create the FIS

In Matlab a FIS structure from partitioning is created by:

fismat = genfis2(Xin,Xout,radii,xBounds,options)

Genfis2 uses subtractive clustering to calculate the centers and the number of the centers and as a result induces the rules which models the data behavior. The input membership function type is 'gaussmf', and the output membership function type is 'linear'. This approach is more efficient than genfis1 for many dimensional datasets and through adjustments to the algorithm different clusters and number of clusters can be found. As a result, the RMSE differs based on the adjustments of the options and radius.

Subtractive clustering algorithm

Subtractive clustering algorithm [25] is the improvement of the idea of mountain clustering. From the data it deduces the cluster centers. The algorithm is fast because the complexity is not based on the number of dimensions but in the number of data in contrast of the mountain clustering. The steps are:

Step 1: Selects the first center from the normalized dataset. Every data point is a candidate center so in every point assigned to Gaussian function

$$D_i = \sum_{j=1}^{n} \exp\left(\frac{- x_i - x_j{}^2}{(r_a / 2)^2} \right) \qquad (6.8)$$

where x_i is the candidate center, x_j is the *j* the data-point and r_a is the radius of the center from the neighboring points. A data-point with many neighbors will have large D so after D_i calculations the maximum D is chosen as the initial center. For the case of equal maximum D randomly selects one of them.

After that the power of every point which belongs next to the cluster center is reduced by subtracting a value:

$$D_i \cdot \exp\left(\frac{- x_i - x_{c_1}{}^2}{(r_b / 2)^2} \right) \qquad (6.9)$$

So the Gaussian function of the i-th neighbor points which are inside the radius r_b gets destroyed with the formula:

$$D_i = D_i - D_{c_1} \exp\left(\frac{- x_i - x_{c_1}{}^{2}}{\left(r_b / 2\right)^2} \right) \tag{6.10}$$

Step 2: The next cluster center should be found and for that reason the previous Gaussian function of the center should be destroyed in order to cancel its influence. The reductions are based on the distance from the previous center.

$$D_i = D_i - Dc_1 \exp\left(\frac{- x_i - x_{c_i}{}^{2}}{\left(r_b / 2\right)^2} \right) \tag{6.11}$$

$where\ r_b = \eta \cdot r_a,$

η is the squash factor r_b is the radius of the first center from a neighbor data point in order to avoid overfitting between clusters. Because $r_b > r_a$ the value of squash factor determines the overfitting .The default value is η =1,5.

Step 3: The same process is repeated for other clusters and in the end the maximum D. The algorithm returns the number of clusters based on the specified parameters such as radius, squash factor, accept ration reject ratio, which are adjusted with *radi* and *options*. The r_a parameter strongly affects the number of clusters that will be generated. A large value of generally results in fewer clusters that lead to a coarse model, while, a small value of can produce an excessive number of rules that may result in an over defined system. The accept ratio sets the potential, as a fraction of the potential of the first cluster center, above which another data point will be accepted as a cluster center. But reject ratio sets the potential, as a fraction of the potential of the first cluster center, below which a data point will be rejected as a cluster center.

3. **ANFIS using Fuzzy C Means clustering for creating FIS**

In Matlab a FIS structure from FCM is created by:
fismat = genfis3(Xin, Xout, FIStype, cluster_n, fcmOptions)

Genfis3 uses the fuzzy c means algorithm to deduce the rules that models the data behavior given the number of cluster centers. The method uses the FCM algorithm to determine the number of the rules and membership functions. Generates a Sugeno-type FIS structure (fismat) given input data Xin and output data Xout. The matrices Xin and Xout have one column per

FIS input and output respectively. By default, the output membership function type is sugeno (output membership function type is linear), however if with *type* can specify type as 'mamdani' (then the output membership function type is gaussmf). Number of clusters can be specified in *cluster_n*, they determine the number of rules and MFs, *fcmOptions* adjusts the specified options of FCM algorithm. FcmOptions are the amount of fuzzy overlap between clusters range (default: 2), the maximum number of iterations (default: 100),

minimum improvement in objective function between two consecutive iterations (default: 1e-005)

Fuzzy c means algorithm (FCM)

Fuzzy clustering [17] by contrast to other clustering techniques, which data belong to only one group among the other groups, data can belong to more than one group. The resulting partition is therefore a fuzzy partition. Each cluster is associated with a MF that expresses the degree to which individual data point belongs to the cluster. Given the number of clusters separates the dataset into c fuzzy clusters by minimizing the within group sum of squared error objective function:

$$JJ_m(U,CL) = \sum_{k=1}^{n}\sum_{i=1}^{c}(U_{i,k})^m \ \left||x_k - cl_i|\right|^2 \qquad (6.12)$$

where $1 \leq m \leq \ \infty$

U is the MF matrix, CL is the set of cluster centers. The squared error is used as a performance index that measures the weighted sum of distances between cluster centers and elements in the corresponding fuzzy clusters. The number *m* governs the influence of membership grades in the performance index. The partition becomes fuzzier with increasing *m* and it has been shown that the FCM algorithm converges for any *m* $\in (1,\infty)$.The necessary conditions for the equation (4.12) to reach its minimum is:

$$U_{ik} = \frac{1}{\left(\sum_1^c \frac{\left||x_k - cl_i|\right|^2}{||x_k - cl_i||}\right)^{\frac{2}{(m-1)}}} \quad , \forall i, \forall k \qquad (6.13)$$

Contact the owner for more informations

6.5 ANFIS with Particle Swamp Optimization

Particle Swamp Optimization (PSO) algorithm [15], [30], inspired form bird's swamp move into the air to find a best position. Each particle gets information from each other and based on the knowledge obtained then moves with some speed to a local best position It takes also into consideration its previous local best position and its neighbors temporary best position to adjust the current particle to its updated local best position and generally moves the swarm to the global best. PSO used to optimize the particles (membership function parameters) of genfis3 so that to minimize the cost function which is RMSE. The steps of PSO are:

1. Initialize the swarm, of particles such that the position of each particle is random within the hyperspace.
2. Evaluated the performance of each particle, using its current position.
3. Compare the performance of each individual to its best performance.
4. Change the velocity vector for each particle.
5. Move each particle to a new position
6. Go to 2 and repeat until converges

Parameters are initialized randomly in step1, then being updated using PSO In each iteration, in first iteration {ai} are updated then in second iteration {bi} are updated, then {ci} are updated and then after updating all parameters again {ai} updated and the process repeated [15].

Contact the owner for more informations

6.6 ANFIS with Genetic Algorithm

Genetic algorithm based on the concept of biological evolutionary processes. GA encodes each point in a solution space into a binary bit string called a chromosome, and each chromosome is evaluated by a fitness function, which corresponds to the objective function of the original problem. Usually, GA keeps a pool of chromosomes at the same time, and these chromosomes can evolve with the operations of selection, crossover, and mutation. After a number of generations, the population will contain, hopefully, chromosomes with better fitness values. Even under the best conditions, only local optimal solution can be expected. GA used to test if it can optimize the membership functions of ANFIS. The steps of GA:

1. Initialization: Create an initial population. The population number is created by a fixed value.
2. Evaluation: The cost function is the minimization of the RMSE and MSE. In an array it holds the best and worst values.
3. Selection: Roulette Wheel Selection algorithm does the selection for the next best generations with the best fitness value.
4. Crossover: From the selected sets it searches for similarities between the sets and keep them for the next generation.
5. Mutation: A vector takes random values and creates next generations. It avoids to fall into a local minimum. Evaluates the mutant vector merges the population sorts them, finds the worst and the best fitness among them stores the best and iterates.
6. Termination takes place when all the iterations finished.

When compared with GA, PSO requires shorter time and memory to obtain better results. Number of clusters, number of population, crossover percentage, mutation percentage, maximum number of iterations, number of offspings, selection pressure are parameters that changes the cost function and the optimization but after experimentation with the above the efficiency had negligible drop for training and for testing data. Table 7.8 summarizes the experiments. From the experiments the number of iterations, the population size and the number of clusters contribute the most to minimizing the cost but the execution time increases a lot.

Contact the owner for more informations

6.7 ANFIS with datasets from PCA

- The new datasets with reduced dimension applying PCA as described in Chapter 3 are:
  Cleveland_PCA[298x3]
  Cleveland_PCA[282x3]
  Cleveland_PCA[282x6]
  Cleveland_PCA[282x11]
  Cleveland_MImputations[375x6]
  Cleveland _MImputations_Scaling[375x6]

- ANFIS tested with methods genfis2 and genfis3 trained with every dataset from above and the accuracy on testing was less than 50% – 55%. Consequently, PCA rejected as method to reduce the dimensions on datasets. Although this lead to the idea to judge the attributes' significance value based on the vectors and their angles between them. So communicating with the cardiologist describing the system, the attributes, the limitations and the simplicity we want to achieve he concluded to an order which weights every attribute's significance. Doctor's order came out based on the examination process considering the ease of the examination, the time and the cost. As a result, constructed the dataset [282x11] with 282 records as it described in Chapter 3.

CHAPTER SEVEN

Experimental Studies with Artificial Neural Networks (ANN)

7.1 Performance factors of ANN

In literature there is not a standard formula of best selection and only with experimentation we can choose the best values for the factors. The most important factors are described below. Dataset size, data range, number of hidden layers, number of the nodes, transfer function type also named activation function, epochs, train/test/validation ratio, learning rate, error goal, performance function, all the above affects the performance of the ANN.

A network with a few hidden layers and nodes cannot fully identify the signals, hence this leads to under fitting. However, a network with many hidden layers and many nodes has less samples in relation to its free parameters so memorizes data points rather than learn the general patterns and this leads to overfitting. So if the hidden layer is too large, it might cause the problem to be under-characterized and the network must optimize more parameters than there are data vectors to constrain these parameters and if there are not many data then convergence will fail (NaN).

7.1.1 Proposed Weight Initialization Technique

Contact the owner for more informations.

Table 7.1 - Best experiments for Multilayer Perceptron Neural Network with every dataset, for three classes, the adjusting parameters and the system's performance.

Dataset	Accuracy%	
	Training	Testing
Cleveland[282x21]	67,2	71,4
AWGN[464x12]	83,5	74
M.I[375x14]	65,3	77,3
L.I[375x14]	44,1	55,3

Graphic Results

Contact the owner for more informations

CHAPTER EIGHT

Conclusion

Coronary Heart Disease is the most dangerous killer factor among the world. Unhealthy food habits no physical activity, high stress, family history, age, sex, are common factors to increase gradually the coronary blockage with severe consequences on blood flow to the heart. The simple medical examinations can give a general idea about the severity although the predictions are not specific on early stages and only with stress test and fluoroscopy can give more clear results about the blockage. The patients do not take these advanced examinations on early stages of the disease or when they do it they are on profound danger of disease. Other times doctors are very busy and cannot examine large amount of patients so a patient's examination gets delayed because the stress test ad fluoroscopy requires much more time than simple examinations as the biochemical blood examination and the ECG.

Artificial Intelligence and Machine Learning using a dataset can create a learning model which will do the prediction with some fuzziness and the goal is to increase it as much as possible to achieve better accuracy.

More specifically, in this thesis used medical datasets, AI systems, ML algorithms to create supervised learning for the optimal prediction of CHD classifying it into three levels of risk: *Absence*, *Medium high*, *Very high risk*. With this classification and the datasets, we constructed differentiated this research from the previous researches since 1988 where the classification was binary (absence or presence) and the system used after advanced examination stages. The classification with three levels of risk has not researched because of the complexity and the significant lack of data. This is the reason why the researchers use data after the patients took advanced heart examinations, such as stress test and fluoroscopy which they are costly, time consuming and painful (sometimes). Consequently, these predictive systems give a very general prediction (absence or present) after advanced examinations and do not offer significant help to the doctor because he can do the predictions by himself with very good accuracy using the medical induction.

Firstly, we used the dataset from University of Cleveland which is used since 1988. Then from the UCI machine learning repository we observed other datasets with missing data and attached those patients to the Cleveland dataset. The missing values was 4% of the total data then using data preprocessing replaced these missing values and the Cleveland dataset increased by 21%. Moreover, with research on the UCI repository and the consulting cardiologist we constructed a new dataset with eleven attributes only from data based on simple questions, biochemical examination and electrocardiograph (ECG) excluding the measures of stress test and fluoroscopy.

Totally we applied statistical data preprocessing on data and we processed them with the following AI and ML techniques: A) Adaptive Neuro-fuzzy Inference Systems (ANFIS) based on, i) Subtractive Clustering, ii) Fuzzy C Means, iii) Particle Swamp Optimization, iv) Genetic Algorithm, v) using datasets from PCA with all the above techniques again, B) Artificial Neural Networks (ANN). The mission was to find which strategy will export diagnosis with the optimal accuracy.

Using the dataset with the eleven attributes an ANFIS with the use of subtractive clustering achieved 72,8% test accuracy for the three classes. Also tested ANFIS with two optimizations algorithms upon its membership functions and both achieved poor predictability accuracy: With Particle Swamp Optimization achieved 60% test accuracy and ANFIS with Genetic Algorithm 56% test accuracy.

After multiply calibrations on the above techniques (i) to (iv), a multilayer Neural Network with the appropriate weight initialization, three layers and sigmoid transfer function gave the best test accuracy: 74% mean value from the three classes, by using eleven attributes such as: age, sex, location of chest pain, number smoking cigarettes per year, the number of years as smoker, if there is family history, the level of cholesterol, the level of fasting blood sugar, the blood pressure, if there is hypertension, and the electrocardiogram. Specifically, the class *Absence* can be used with very good credibility based on ROC plot categorization {Almost excellent, Very Good, Good, Mediocre, Worthless}. The classes *Medium high* and *Very high* risk have with good credibility. The supporting system helps the doctors to make predictions on CHD much faster reducing the financial cost and the stress for the patients by warning them soon about their condition.

Also another ANN achieve 77.3% test accuracy by using the augmented dataset (21% more data that the typical used dataset as is described above) but it requires thirteen attributes and many of them are result of stress test and fluoroscopy. This result which is very close to 74% highlights much more the importance of the previous method with the ANN and the eleven simple attributes.

REFERENCES

[1] World Health Organization, Cardiovascular diseases (CVDs) http://www.who.int/mediacentre/factsheets/fs317/en/

[2] World Health Organization, Cardiovascular diseases (CVDs), The top 10 causes of death. http://www.who.int/mediacentre/factsheets/fs310/en/

[3] Morbidity & Mortality: 2012 Chart Book on Cardiovascular, Lung, and Blood Diseases https://www.nhlbi.nih.gov/files/docs/research/2012_ChartBook.pdf,page 8.

[4] How Cardiovascular & Stroke Risks Relate http://www.strokeassociation.org/STROKEORG/LifeAfterStroke/HealthyLivingAfterStroke/UnderstandingRiskyConditions/How-Cardiovascular-Stroke-Risks Relate_UCM_310369_Article.jsp#.WSGzU2jyguE

[4] Chen A.H., "HDPS: Heart Disease Prediction System", Computing in Cardiology, ISSN 0276-6574, pp 557-560,2011.

[5] Fayyad,G.Piatetsky,P.Smyth "From Data Mining to Knowledge Discovery in Databases", AI Magazine Volume 17 Number 3 1996.

[6] V.Gujiri, A.Joshi, A.Chavan, "ECG Signal Analysis for Abnormality Detection in the Heart beat", GRD Journals- Global Research and Development Journal for Engineering, Volume 1, Issue 10, September 2016 ISSN: 2455-5703.

[7] Dr. Ahmed, "Heart Blockage Explained with pictures", https://myheart.net/articles/heart-blockage-explained-with-pictures/

[8] Angioplasty and stent placement, https://medlineplus.gov/ency/article/007473.htm

[9] "How much blockage in the coronary arteries is needed to cause angina". http://scarysymptoms.com/2012/02/how-much-coronary-artery-blockage-can/

[10] Angioplasty.org, Discussion Forum: Experiences with stress tests. http://www.ptca.org/forumtopics/topic_stress_tests.html

[11] H.Kang, "The prevention and handling of the missing data".

[12] Matlab PCA https://www.mathworks.com/help/stats/princomp.html

[13] R. Jang, “ANFIS: Adaptive-Network-Based Fuzzy Inference System”, IEEE Transactions on systems, man, and cybernetics, vol 23, No. 3, May/June 1993.

[14] Adriano Oliveira Cruz ,“ANFIS: Adaptive Neuro-Fuzzy Inference Systems”.

[15] S.Sivagowry 1, M. Durairaj “A Study on the accessible techniques to classify and predict the risk of Cardio Vascular Disease”, International Journal of Computer Trends and Technology (IJCTT) – Volume 32 Number 1 - February 2016.

[15] V.Sydi, Mahdi Shoorehdeli “Training ANFIS with modified PSO”, Conference Paper July 2007 DOI: 10.1109/MED.2007.4433927.

[16] Mollaiy Berneti, “Design of Fuzzy Subtractive Clustering Model using Particle Swarm Optimization for the Permeability Prediction of the Reservoir”, “International Journal of Computer Applications (0975 – 8887) Volume 29– No.11, September 2011”.

[17] L. Chiu, “Fuzzy Model Identification Based on Cluster Estimation”, “Journal of Intelligent and Fuzzy Systems · January 1994 DOI: 10.3233/IFS-1994-2306”.

[18] Takagi, Sugeno “Fuzzy Identification of systems and its applications to modeling and control”, IEEE transactions on systems, man, and cybernetics, vol, smc 15, No1 ,January/February 1985.

[19] Raul Rojas, “Neural Networks A Systematic Introduction”, Springer.

[20] Matlab, “Fuzzy Logic Toolbox, A user’s guide”.

[21] Matlab, “Neural Network Toolbox, Getting started guide”.

[22] W.Craven, W. Shavlik, “Using Neurall Networks for Data Mining”.

[23] J. Shlens, “A Tutorial on Principal Component Analysis, Derivation Discussion and Singular Value Decomposition”.

[24] J.Vijayashree, N.Ch.Sriman, Narayana, Iyengar, “ Heart Disease Prediction System Using Data Mining and Hybrid Intelligent Techniques: A Review”, International Journal of Bio-Science and Bio-Technology Vol.8, No.4 (2016), pp. 139-148.

[25] Priyono, Ridwan2, Alias1, Atiq, o. k. Rahmat3, Hassan2 & Mohd. Alauddin Mohd. Ali, “Generation of Fuzzy Rules with Subtractive Clustering”, Jurnal Teknologi, 43(D) Dis.2005: 143–153.

[26] Simon Haykin, “Neural Networks: A Comprehensive Foundation”, Second edition, Prentice-Hall, Upper Saddle River, NJ, 1999. ISBN 0-13-273350-1.

[27] IBM Knowledge Center (SPSS Statistics v 23.0.0). Prop. of Var. Explained. Retrieved May 28th 2017 from: https://www.ibm.com/support/

[28] Elbedwedhy, M.Zawbaa, Ghali, Hassanien, “ Detection of Heart Disease using Binary Particle Swarm Optimization “, Proceedings of Federated Conference on Computer Science and Information System, pp 177-182, © IEEE,2012.

[29] Tabachnick and Fidell (2007, p. 646).

[30] Durairaj. M, Sivagowry. S, “Feature Diminution by Using Particle Swarm Optimization for Envisaging the Heart Syndrome”, International Journal of Information Technology and Computer Science (IJITCS) 2015.02.05.

[31] T.Sathanam, E.P. Ephzibah “Heart Disease Classification Using PCA and Feed Forward Network”.

[32] Liu,Frazier, Kumar, “Comparative assessment of the measures of thematic classification accuracy”, Remote Sensing of Environment 107 (2007) 606–616.

[33] Persi Pamela, Gayathri.P, Jaisankar.N, “ A Fuzzy Optimization technique for the prediction of Coronary Heart Disease using Decision Tree”, International Journal of Engineering and Technology, Vol 5(3), pp 2506-2514, June- July 2013.

[34] Mohd. A.M. Abushariah, Assal A.M Alquadah, Y.Adwan, Yousef, “Automatic Heart Disease Diagnosis system based on Artificial Neural Network and ANFIS approach”, Journal of Software Engineering and Application, pp 1055-1064, 2014.

[35] M.N. Norazian, A. Mohd Mustafa Al Bakri, Y. Ahmad Shukri, R. Nor Azam, “Estimating Missing Data Using Interpolation Technique: Effects on Data Distributions”.

[36] Glorot, Xavier, and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." International conference on artificial intelligence and statistics. 2010.

[37] "UCI Coronary Heart Disease Repository" http://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/

[38] Jiawei Han and Micheline Kamber, "Data Mining: Concepts and Techniques"

[39] Christopher Bishop "Pattern Recognition and Machine Learning".

[40] Boston University, "Interquartile Range", http://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_summarizingdata/bs704_summarizingdata7.html

[41] National Heart, Lung and Blood Institute, "What to expect during stress testing" https://www.nhlbi.nih.gov/health/health-topics/topics/stress/during

[42] Dr. Eric Bricker, "5 types of Cardiac Stress Test" http://www.compassphs.com/blog/price-transparency/5-types-of-cardiac-stress-tests-they-are-not-all-the-same-2/

[43] Matlab, "Receiver Operating Characteristic", https://www.mathworks.com/help/stats/perfcurve.html

[44] Kumar, Indrayan, "Receiver Operating Characteristic (ROC) Curve for Medical Researchers".

[45] Harvard Medical School, "Race and Ethnicity: Clues to your heart disease Rink" https://www.health.harvard.edu/heart-health/race-and-ethnicity-clues-to-your-heart-disease-risk

[46] Colantonio et al, "Black-White Differences In Incident Fatal, Non Fatal, And Total Coronary Heart Disease."

[47] "High Blood Pressure in African Americans", https://www.webmd.com/hypertension-high-blood-pressure/guide/hypertension-in-african-americans#1

[48] Yaser Mustafa, Ismail, "Learning From Data". (Edition 2012, Chapter 7, pg. 26).

[49] Yan, H., Jiang, Y., Zheng, J., Peng, C., Li, Q. (2006). “A multilayer perceptron-based medical decision support system for heart disease diagnosis.” Expert Systems with Applications, 30(2), 272-281.

[50] Xavier Glorot, Antoine Bordes, Yoshua Bengio , “Deep Sparse Rectifier Neural Networks”

[51] Alex Krizhevsky, Sutskever, Hinton “ImageNet Classification with Deep Convolutional Neural Networks”

[52] Maas, Andrew L.; Hannun, Awni Y.; Ng, Andrew Y. (June 2013). "Rectifier nonlinearities improve neural network acoustic models".

[53] He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian (2015-02-06)."Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification"

[54] Xu, Bing; Wang, Naiyan; Chen, Tianqi; Li, Mu (2015-05-04). “Empirical Evaluation of Rectified Activations in Convolutional Network.”

[55] Hestenes, Magnus R.;Stiefel, Eduard, “Methods of Conjugate Gradients for Solving Linear Systems”, Journal of Research of the National Bureau of Standards.