CHAPTER ONE

INTRODUCTION

1.0 Background to the Study

In the past years, researchers involved in artificial intelligence have developed predictive models for various areas of science including chemistry, engineering, finance and medicine (Laurikkala, 1998). Medical predictive models are designed to facilitate diagnosis of health problems, and routine classification tasks which should otherwise be referred to the specialist in a particular area of medicine. These models are built using existing data from actual cases.

The data can be preprocessed and expressed in a set of rules, such as is often the case in the knowledge-based expert systems, or serve as the training data for statistical and medical learning models (Wei –pinching, and Der- Ming, 2009). These traditional and statistical methods sometimes create uncertainty especially when patterns are generated from random sampling or from overall population. Statistical approach cannot handle large complex data set or perform complex search during pattern extraction from large database, thereby producing unreliable and undependable results (Wei –pinching, and Der- Ming, 2009). These results cannot be used for efficient, timely and effective decision making.

Even though Information technology tools like decision support system and other modeling tools have been considerably useful in decision making, however, enormous effort is required to retrieve relevant knowledge from the databases because of large new Information Technology (IT) tools. This quest for relevant knowledge led to a growing interest in knowledge discovery and Data mining tools which are capable of extracting hidden patterns which ordinary statistical and traditional tools cannot uncover.

Data mining is capable of facilitating the discovery of interesting and useful knowledge from a large database (Myoling and Ingoo, 2003).

Data mining is defined as the exploration and analysis of large quantities of data in order to discover meaningful patterns and rules (Berry et al, 2004).

Classification and prediction are important in decision making. Classification consists of examining the features of a newly presented object and assigning it to one of a predefined set of classes. In computing, the objects to be classified are sometimes presented as records in a data base.

However, since medical domain classification problem is highly non-linear in nature, it is difficult to develop a comprehensive model, taking into account all the independent variables using conventional statistical modeling techniques. Furthermore traditional mixtures of statistical techniques and data management tools are no longer adequate for analyzing vast collection existing of data (Chengic et al 2006). Classification is one of the major tasks in data mining field. There are different techniques in Data mining, but the most popular ones in medicine are logistics regression (LR), decision tree, and the most robust of them all is the Artificial Neural Network (ANN) simply referred to as Neural Network (NN).

Artificial Neural Networks are popular because they have a proven track record in many data mining and decision support applications.

Neural Networks (NNs) are a class of powerful, general-purpose tools readily applied to prediction, classification and clustering. They have been applied across a broad range of industries, from predicting time series in financial world to diagnosing medical conditions, from identifying clusters of valuable customers to identifying fraudulent credit card transactions.

An Artificial Neural Network (ANN) is an artificial representation of the human brain that tries to simulate the learning process. It makes use of a collection of mathematical models that emulate the observed structure and dynamics of the brain in order to simulate its learning process. Artificial Neural Networks have the advantage of representing linear and non-linear relationships and learning this relationship directly from the data being modeled. (Neuro Dimension Inc., (2002). This is what gave neural network its great power over traditional linear models. Traditional linear models are incapable of modeling non-linear relationship among data.

In his work Hornik et al (1989) demonstrated that a neural network can approximate any given functional form to any desired accuracy level, if built with sufficient number of hidden layers.

Neural networks have the ability to learn from examples in much the same way that human experts gain from experience. Neural networks are thus good for prediction problems. A good problem has the following three characteristics:

(i) The inputs are well understood.

(ii) The output is well understood.

(iii) Experience is available.

Neural Network learns from experience derived from the past and not by analyzing complex relationships among data through programming. Learning is the basis of intelligence. It is the ability of a system to improve its performance at a given task without programming.

After training on some set of cases, the network begins to recognize itself and refines its own architecture to fit the data by looking for patterns in the training data or some forms of relationships between the inputs and the result of each record in much the same way as human brain learns.

As good as Neural Network is in classification and predictions, it has some limitations.

Out of the several Neural Network Algorithms, Multilayer Perceptron (MLP) is the most common Artificial Neural Network algorithm because it uses conventional feed forward networks. Back propagation algorithm is widely used for training MLP feed forward Neural Network. But it has been shown that Back propagation algorithm has some shortcomings especially overtraining and inability to find global solution because of low convergence during learning process. All these disadvantages lead to overgeneralization during the learning process of Multilayer Perceptron Neural Network (MLP-NN).

Genetic Algorithm (GA) is a good technique that takes cares of these shortcomings of Back propagation algorithm. Genetic Algorithm is a more recent technique applied to classification problem which are Heuristics search techniques based on the theory of natural selection and evolution (Holland, 1975). GA can be used to find approximate solutions to complicated problem through application of the principles of evolutionary biology. Genetic Algorithms also called evolutionary algorithms have been applied to optimization problem in various industries including complex scheduling problems and classification problems involving complex data types. They have also been used in combination with other data mining algorithms, including determining the best topology for neural networks and can also be used to train and optimize weights in Neural Networks, so that the expected output generated during training of the Neural Network will not be too different from the neural network output (predicted value).

Therefore in this work, Genetic Algorithm techniques will be combined with MLP-NN techniques in order to improve the classification and prediction capability of MLP-NN.

1.2 Statement of Problem

The Multilayer Perceptron Neural Network even though is the most widely used ANN for classification and prediction in medical research has the problems of local minimum and over-fitting during the learning process. Therefore, this research seeks to solve this problem by enhancing the capability of MLP-ANN using Genetic Algorithm (GA). GA is a well-known algorithm for global optimization in classification and prediction problems. GA on its own is not a Data Mining Algorithm that can be used alone in Knowledge Discovery in Database (KDD). The hybrid algorithm of MLP-ANN and GA will therefore enhance KDD process.

1.3 Research Motivations

The MLP back propagation algorithm that is widely used for training multi-layer feed forward networks is characterized by hills and valleys that cause techniques such as Back propagation algorithm (BPN) to become trapped in local minimum. The main disadvantage of Back propagations algorithm (BPN) are:

- There is no assurance of convergence to the right solution

- It could also take forever to converge when small insignificant step size is required to guarantee convergence.
- It could also fail to converge if wrong choice of step size is chosen.

This is known as oscillation. (Bose and Liang, 1996).

The weights generated between interconnections of the neurons during training can cause low convergence. All these shortcomings of the MLP- Back propagation algorithm can also lead to the following major problems:

- Local minima during learning process of Multi-layer perceptron Neural Network (MLP-NN).
- Overgeneralization of classification ability of MLP-NN
- Overtraining, and over fitting during the learning process of the MLP – NN.

Because of the important features of the search ability of the genetic algorithm (GA) as a global search algorithm and also its high functionality of the global optimum, genetic algorithm can accomplish strong searching capability during the learning process which will reduce the problem of low convergence and local optimum during training. Thus, in this work GA is used to overcome the problem of MLP- NN.

GA is well known for improving prediction, classification accuracy and reducing over-training because of its global optimal search capability.

This research is designed to minimize the problem of low accuracy and overtraining by embedding GA into Multilayer perceptron ANN (MLP-ANN). The resulting algorithm is then used to facilitate a better classification and prediction accuracy in medical data base.

1.4 Aims and Objectives of the Study

The aim of this research work is to develop a Data Mining Model that combines Genetic Algorithms with Multilayer perceptron artificial neural network in order to improve the classification/prediction accuracy and also strengthen the learning process of the multilayer perceptron Artificial Neural Network.

The specific objectives are to:

(i) Develop a Neuro-Genetic Data mining tool (NEGEM) for mining data for classification and prediction purposes.

(ii) Create a multi-dimensional HIV/AIDS patients Database containing historical data set using Microsoft SQL server.

(iii) Test the functionality of the NEGEM in mining HIV/AIDS patient Data in order to extract novel patterns that can be used for classification and prediction of HIV/AIDS patients' status.

(iv) Benchmark the performances of NEGEM with that of a well-known off- the shelf Data mining tool - Waikato Engineering Knowledge Analysis (WEKA).

(v) Measure the capability of the classification/prediction accuracy using performance error metrics.

(vi) Simulate the developed model to assist in monitoring the prevalence of HIV/AIDS in any region in Nigeria.

1.5 Research Questions

The research Questions that this thesis addresses are:

RQ1: How can a model that combines the strength of Neural Network with that of Genetic Algorithm be developed?

RQ2: To what extent can the Neuro-Genetic model (NEGEM) be used to mine HIV/AIDS patient Database for classification/prediction of HIV/AIDS patients' status?

RQ3: Does the NEGEM model outperforms the off the shelf Data mining tool (WEKA) in classification/prediction Accuracy when tested on a real life HIV/AIDS patients medical Data?

RQ4: Which of the NEGEM model i.e. (Different hidden layers for the network topology) gave the best solution? Are there any statistical differences among the models?

1.6 Research Methodology

A lot of literature search form the basis of the research work. A review of the theoretical framework on the concept of knowledge discovery system, Data mining, WEKA, and HIV/AIDS in Nigeria was carried out as well as a review of related works on the applications on feed forward Artificial Neural Networks and Genetic Algorithms especially on HIV/AIDS Data management for classification/prediction. The strength and weaknesses of each work were identified to note the gap that existed in literature. A new approach that combines the strengths of works done in MLP-ANN and GA was preferred. Thus a model combining the strengths of Multilayer Perception Neural Network and that of Genetic Algorithm called (NEGEM) was designed. The

MLP Delta learning algorithm was used to implement ANN while GA was used to train in order to avoid low convergence and overtraining of MLP – ANN.

Data of HIV/AIDS patients collected from different hospitals in the southwestern Nigeria was used to test the Model. The data were organized into a two-tier architecture that allows for multi-dimensional analysis. Three different multilayer perceptron hidden layers (one, two and three) network implemented on C# programming language and Microsoft SQL server were used to create the Database under visual studio platform to predict and classify HIV/AIDS data. The Data collected was divided into training, and test data which were imported into the developed model. Mutation and cross over operators with 2000 training epochs in Genetic Algorithms were used as operating parameters to avoid low classification and prediction accuracy and overtraining.

The capability of the model for prediction/classification was compared with that of WEKA, an off- the shelf data mining software to determine NEGEM validity and its contributions. Classification and predictive accuracies were measured using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Recall and precision were used to measure the level of true positive prediction/classification and overtraining.

1.7 Significance of the Study

Artificial Neural Network (ANN) is very popular in classification and prediction problems. Despite its strength, there exists draw backs in its implementation. This research work minimized the effects of some of the drawbacks of MLP-ANN by embedding a Genetic algorithm into it, in order to enhance its functionality in prediction and classification ability. This is very important in medical diagnosis where optimal correct analyses of diseases are beneficial to human existence. The resulting tool can also be used in any hospital that has at least one personal computer as a compliment on ground and also where medical diagnoses are not readily available. This will in no small measure help in facilitating quick diagnosis of HIV/AIDS in rural areas.

The model will assist the government in developing timely intervention policy for the control of HIV/AIDS in Nigeria.

1.8 Organization of the Thesis

Chapter one deals with the background to the study, statement of problems, aims and objectives of the study, research questions, research methodology and significance of the study.

Chapter two discusses the theoretical framework to the study, and related literature on knowledge discovery, statistics, data mining, artificial neural network, Genetic Algorithm, WEKA and the history of HIV/AIDS and visual studio platform.

In addition chapter two also discusses a review of different related work on the Applications of Artificial Neural Networks in classification and prediction modeling, and general data mining applications using Artificial Neural Networks for HIV/AIDs modeling.

In chapter three the Architecture of the developed model, the analysis and design tools, implementation tools, the description of the Neuro-Genetic Algorithm, general discussion of the developed models and the application of in mining HIV/AIDS medical database are described .

Chapter four presents the Result and Discussion of the Neuro-Genetic model and performance measures result of NEGEM and WEKA.

Chapter five highlights the conclusion drawn from the research work, the contributions of the research to knowledge, and recommendations for further research.

CHAPTER TWO

THEORETICAL FRAMEWORK AND REVIEW OF RELATED WORKS

2.1 Introduction to Data Mining

Progress in digital data acquisition and storage technology has resulted in the growth of huge databases. This has occurred in all areas of human endeavor, from the mundane such as supermarket transaction data, credit card usage records, telephone call details, and government statistics to the more exotic such as images of astronomical bodies, molecular databases, and medical records. Thus interest has grown in the possibility of extracting information that might be of value to the owner of the database. The discipline concerned with this task has become known as data mining (David et al, 2001).

Data mining takes advantages of advances in the fields of artificial intelligence and Statistics. Both disciplines are concerned with the problems of pattern recognition and classification. The development of most statistical techniques was, until recently based on elegant theory and analytical methods that worked quite well on the modest amounts of data being analyzed. But a rapid development on Information Technology has resulted in large volume of data being collected and analyzed from different establishments. The increased power of consumption and their lower cost, coupled with the need to analyze enormous datasets with millions of rows, have made traditional statistics practically difficult to use. Therefore, new techniques like Knowledge Discovery in databases and Data mining include relatively recent algorithms like neural nets, decision tree and new approaches such as discriminant analysis emerged to use the virtue of increased computer power on the huge data.

Knowledge Discovery and Data mining are the landmarks of the Information Age because acquiring, storing, and understanding data have posed great challenges and brought a lot of promises. Knowledge Discovery in Databases (KDD) and Data Mining (DM) have emerged as high profile, rapidly evolving, greatly needed and important areas (Witold, 2011).

Fayyad, et al (1996) provided an overview of this emerging field, clarifying how data mining and knowledge discovery in databases are related both to each other and to related fields, such as machine learning, statistics, and databases. Data mining provides automatic pattern recognition and it is used to uncover pattern in data, which are difficult to detect with traditional statistical methods.

Finding useful patterns in data has been given a variety of names, including data mining, knowledge extraction, information discovery, information harvesting, data archaeology, and data pattern processing. The term data mining has mostly been used by statisticians, data analysts, and the management information systems (MIS) communities. It has also gained popularity in the database field. The phrase knowledge discovery in databases was coined at the first KDD workshop in 1989 to emphasize that knowledge is the end product of a data-driven discovery. It has been popularized in the AI and machine-learning fields.

In our view, KDD refers to the overall process of discovering useful knowledge from data, and data mining refers to a particular step in this process. Data mining is the application of specific algorithms for extracting patterns from data. The distinction between the KDD process and the data-mining step (within the process) is depicted in figure 2.1. Data mining is one of the steps in the KDD process. 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 is essential to ensure that useful knowledge is derived from the data.

Knowledge discovery process includes selection, preprocessing, transformation, data mining and interpretation/evaluation.

The data for use in the knowledge discovery process has to be prepared for easy extraction. Data preparation involves selecting the particular data to be targeted from the set of all data and then performing some degree of ‘data cleaning’ after which data adjustments can be made to the data before being stored in a central repository, the data warehouse. (Fayyad et al, 1996 and Bradley et al, 1999). Adjustments to the data are necessary in emphasizing the dimensions under study, whereas cleaning is necessary due to errors that may be inherent in the target data. Erroneous target data may result in the discovery of misleading patterns during data mining.

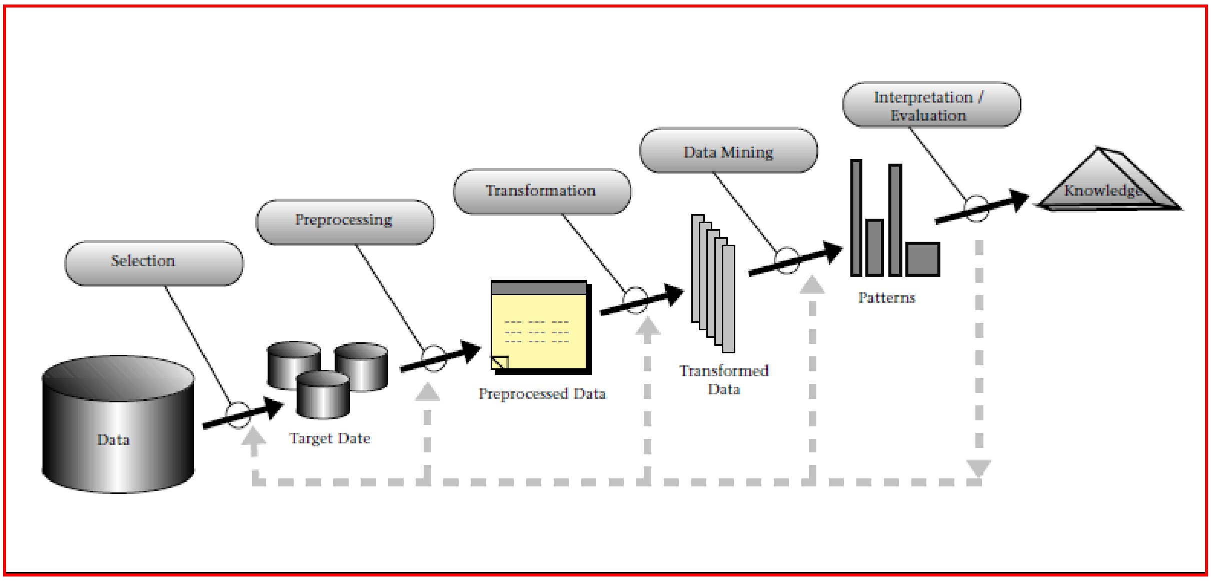


Figure 2.1 Framework of Knowledge discovery process (Fayyad et al, 1996).

Data mining is the analysis of often large observational data sets to find unsuspected relationships within the data in novel ways that are both understandable and useful to the data owner. The definition above refers to “observational data”, as opposed to “experimental data”. Data mining typically deals with data that have already been collected for some purpose other than the data mining analysis. For example, such data might have been collected in order to maintain an up-to-date record of all transactions in an organization. Therefore the objectives of the data mining exercise play no role in the data collection strategy. This is one way in which data mining differs from statistics, in which data are often collected by using efficient strategies to answer specific questions. For this reason, data mining is often referred to as “secondary” data analysis. Data mining is set in the broader context of knowledge discovery in databases, or KDD. This term originated in the Artificial Intelligence (AI) research field. The KDD process involves several stages; selecting the target data, preprocessing the data, transforming them if necessary, performing data mining to extract patterns and relationships, and then interpreting and assessing the discovered structures.

Data Mining has attracted a great deal of attention in the information industry in recent years due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. The information and knowledge gained can be used for applications ranging from business management, production control, and market analysis, to engineering design and science exploration.

An evolutionary path witnessed in the database industry is in the development of the functionalities depicted in Figure 2.2. Early development of data collection and database creation mechanisms served as a prerequisite for later development of effective mechanisms for data storage and retrieval, and query and transaction processing. With numerous database systems offering query and transaction processing as common practice, data analysis and understanding has naturally become the next target.

Data mining is supported by three technologies.

a. Massive data collection.

b. Powerful multi-processor computers.
c. Data mining algorithms

From the user’s point of view, four steps listed in Table 2.1 were revolutionary because they allowed new business questions to be answered accurately and quickly.

The core components of data mining technology have been under development in research areas such as statistics, artificial intelligence, and machine learning. Today, the maturity of these techniques coupled with high-performance relational database, engines and broad data integration efforts make these technologies practical for current data warehouse environments.

There are different data mining algorithm that can be used to accomplish the task below.

1. **Model or pattern structure**: To determine the underlying structure or functional forms that we seek from the data.
2. **Score Function**: To judge the quality of a fitted model.
3. **Optimization and search Method**: To optimize the score function and searching over different model and pattern structure.
4. **Data management strategy**: To handle data access efficiently during the search/optimization.

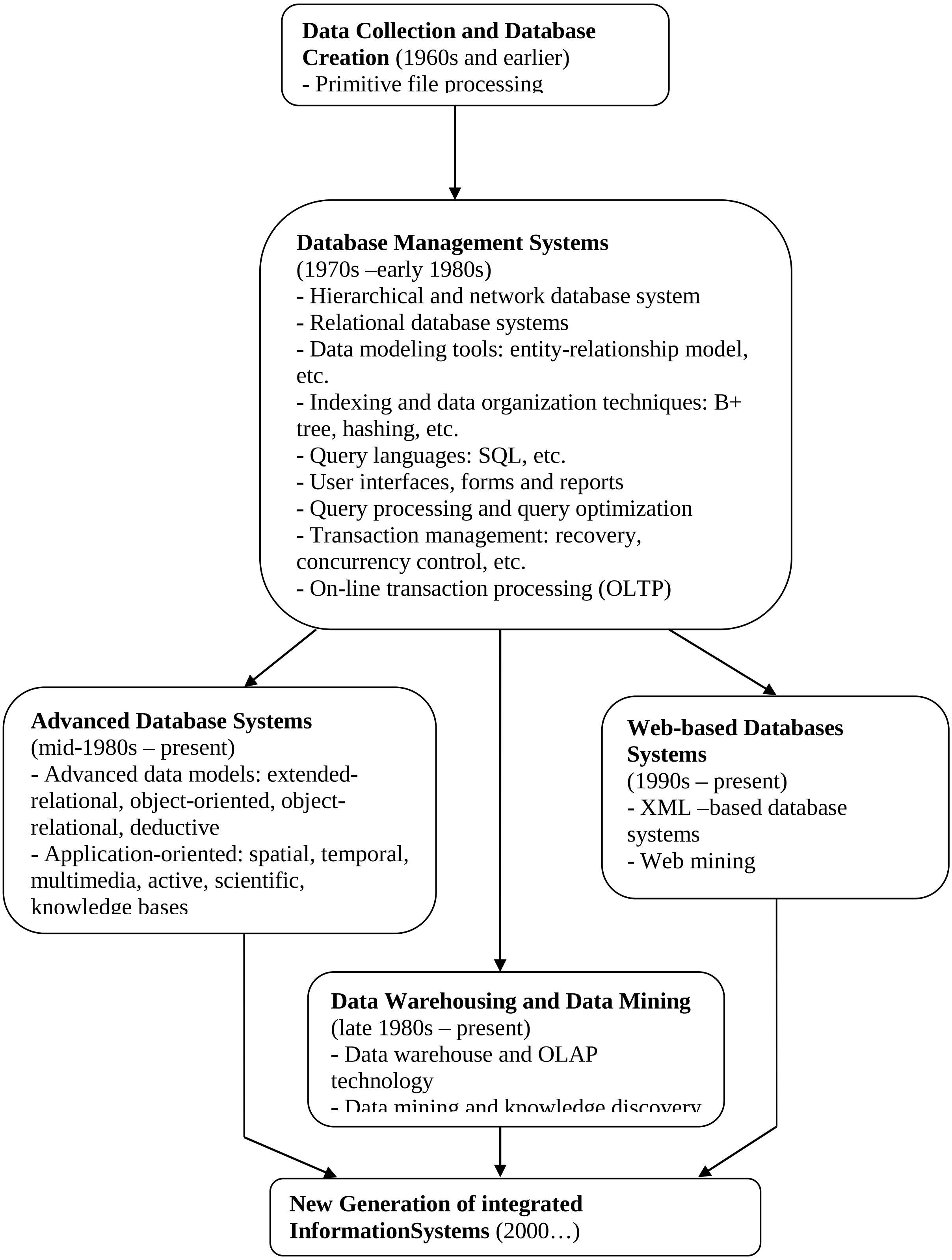

Data Collection and Database Creation (1960s and earlier)
- Primitive file processing
Database Management Systems
(1970s –early 1980s)
- Hierarchical and network database system
- Relational database systems
- Data modeling tools: entity-relationship model, etc.
- Indexing and data organization techniques: B+ tree, hashing, etc.
- Query languages: SQL, etc.
- User interfaces, forms and reports
- Query processing and query optimization
- Transaction management: recovery, concurrency control, etc.
- On-line transaction processing (OLTP)
Advanced Database Systems
(mid-1980s – present)
- Advanced data models: extended-relational, object-oriented, object-relational, deductive
- Application-oriented: spatial, temporal, multimedia, active, scientific, knowledge bases
Web-based Databases Systems
(1990s – present)
- XML –based database systems
- Web mining
Data Warehousing and Data Mining
(late 1980s – present)
- Data warehouse and OLAP technology
- Data mining and knowledge discovery
New Generation of integrated InformationSystems (2000…)

Fig. 2.2 Database Industry Evolutionary Path (http://www.sims.monash.edu.au /subjects/ims5026.nsf/0/ 98d52f2204fed3b6ca256 cda007db6fe/$FILE/Intro.ther.pdf.)

Table 2.1: Steps in the Evolution of Data Mining

Evolutionary step	Business operation	Enabling techniques	Product providers	Characteristics
Data collection (1960s)	What was my total revenue in the last 5 years?	Computers, tapes disks	IBM,CDC	Retrospective, static data delivery
Data access	What were unit sales in store last month?	Relational databases (RDBMS), structured query language (SQL), ODBC	Oracle, Sybase, Informix, IBM, Microsoft	Retrospective, dynamic, data delivery at record level
Data ware housing and decision	What were new sales in the store last month?	On-line analytic processing (OLAP)	Pilot Comshare, Arbor, Cognos	Retrospective, dynamic, data delivery at record level
Data mining	What is likely to happen to Boston unit	Advanced algorithms, multi-processor computer	Pilot, Lochea, IBM, SGL	Retrospective, dynamic, data delivery at record level

Source:www.thearling.com/text/dmwhite.htm May 6th, 2013

2.2 Techniques in Data Mining

The most commonly used techniques in data mining are:

1. Decision Trees
2. Memory-Based Reasoning
3. Market Basket Analysis & Association Rules
4. Link Analysis
5. Automatic Cluster Detection
6. Artificial Neural Networks
7. Genetic Algorithm.

2.2.1 Decision Trees

Decision trees are powerful and popular tools for classification and prediction. The attractiveness of tree-based methods is due in large part to the fact that, in contrast to neural networks, decision trees represents rules. Rules can readily be expressed in English that humans can understand or in a database access language like SQL so that record falling into a particular category may be retrieved.

There are varieties of algorithms for building decision trees which share the describe traits of explicability. Two of the most popular are CART and CHAID which stand respectively for Classification and Regression Trees and Chi-square Automatic Interaction Detection. A new algorithm, C4.5, is gaining popularity and is now available in several software packages.

The strengths of decision-trees are ability to:

- Generate understandable rules
- Perform classification without requiring much computation.

- Handle continuous and categorical variables
- Provide a clear indication of which fields are most important for prediction or classification.

However, decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous variable such as income, blood pressure or interest rate. Decision trees are also problematic for time-series data unless a lot of effort is put into presenting the data in such a way that trends and sequential patterns are made visible.

Decision-tree methods are a good choice when the data-mining task is classification of records or prediction of outcomes. Decision trees are also a natural choice when the goal is to generate rules that can be easily understood, explained, and translated into SQL or a natural language.

2.2.2 Memory Based Reasoning (MBR)

Memory-based reasoning systems support the modeling phase of the data mining process. Their unique feature is that they are relatively machine driven and involve automatic classification of cases. MBR is a highly useful technique that can be applied to text data as well as traditional numeric data domains.

Memory-based reasoning is an empirical classification method. It operates by comparing new unclassified records with known examples and patterns. (Stanfilland Waltz, 1986).

The case that most closely matches the new record is identified, using one of a number of different possible measures. Memory-based reasoning provides best overall classification when compared with the more traditional approaches in classifying jobs with respect to back disorders.(Zurada, et al 2004).

The strength of MBR includes the following:

- It produces results that are readily understandable
- It is applicable to arbitrary data types, even non- relational data.
- It works efficiently on any number of fields
- .It requires minimal amount of effort to maintain the training set..

MBR weaknesses also include the following:

- It is computationally expensive when doing classification and prediction
- It requires a large amount of storage for the training set.
- Its results can be dependent on the choice of distance function, combination function and the number of neighbors.

2.2.3 Market Basket Analysis

Market-basket analysis refers to methodologies studying the composition of a shopping basket of products purchased during a single shopping event.

This technique has been widely applied to grocery store operations as well as other retailing operations. Market basket data in its rawest form would be the transactional list of purchases by customer, indicating only the items purchased together with their prices. This data is challenging because of a number of characteristics that include:

- A very large number of records (often millions of transactions per day)
- Sparseness (each market basket contains only a small portion of items carried)

- Heterogeneity (those with different tastes tend to purchase a specific subset of items).

The aim of market-basket analysis is to identify what products tend to be purchased together. Analyzing transaction-level data can identify purchase patterns, such as which snacks are purchased with chicken during Christmas. This information can be used in determining where to place products in the store, as well as aid inventory management. Product presentations and staffing can be more intelligently planned for specific times of day, days of the week, or holidays. Another commercial application is electronic couponing, tailoring coupon face value and distribution timing using information obtained from market baskets (Russell and Petersen, 2000).

The ultimate goal of market basket analysis is finding the products that customers frequently purchase together. The stores can use this information by putting these products in close proximity of each other and making them more visible and accessible for customers at the time of shopping.

These assortments can affect customer behavior and promote the sales for complement items. The other use of this information is to decide about the layout of catalogs and put the items with strong association together in sales catalogs. The advantage of using sales data for promotions and store layout is that the consumer behavior information determines the items with associations. This information may vary based on the area and the assortments of available items in the store. The point of sale data reflects the real behavior of the group of customers that frequently shop at the same store. Catalogs that are designed based on the market basket analysis are expected to be more effective on consumer behavior and sales promotion.

The strengths of Market Basket Analysis include the following:

- It produces clear and understandable results
- It supports undirected data mining
- It works on variable-length data.
- It uses simple understandable computations.

The weaknesses of Market Basket Analysis are listed below:

- It requires exponentially more computational effort as the problem size grows.
- It has a limited supports for attributes on the data
- It is difficult to determine the right number of items
- It discounts rear items

2.2.4 Link Analysis

Link Analysis is based on a branch of mathematics called graph theory. Link Analysis is not applicable to all types of data nor can it solve all types of problems. However when it can be used, it yields very insightful and actionable results. Areas where it can yield good results include the following:

- Identifying authoritative sources of information on the World Wide Web by analyzing the links between its pages.
- Analyzing telephone call patterns to identify particular market segments such as people working from home.
- Understanding physician referral patterns, a referral is a relationship between two physicians.

2.2.5 Automatic Cluster Detection

Clustering provides a way to learn about the structure of complex data. Once proper clusters have been defined, it is often possible to find simple patterns within each cluster, popular cluster algorithms include K-Means, Gaussian mixture models, agglomerative clustering, divisive clustering and self-organizing maps.

2.2.6 Artificial Neural Network

Artificial neural networks or simply Neural Networks are popular because they have a proven track record in many data mining and decision- support applications. Neural networks are a class of very powerful, general-purpose tools that are readily applied to prediction, classification, and clustering. They have been applied across a broad range of industries, from predicting financial series to diagnosing medical conditions, from identifying clusters of valuable customers to identifying fraudulent credit card transactions, from recognizing numbers written on checks to predicting the failure rates of engines (Berry and Linoff, 1997).

Artificial Neural Networks (ANN) came into being in the late 1950 but it was not until the middle of 1980’s that its algorithm became powerful enough for general application. An artificial neural networks are connectionist architectures, parallel disturbed processing and neuromorphic systems that are artificial representations of the human brain which try to simulate the learning process. This learning process is achieved by using collections of mathematical models that emulate some of the observed properties of biological nervous systems and draw the analogies of adaptive biological learning (Frohlich, 1996).

Principe et al (2000) defined ANN as distributive, adaptive and generally non-linear learning machines built from many different processing elements (PEs). Thus, Artificial Neural Network is a network of interconnected elements, which are made to behave like biological neurons, and thereby enabling the whole network to work in a similar way to the human brain.

It is true that modern computer can easily perform better than human brain in pre-programmable, repetitive computations, but intelligent tasks such as recognizing a face, reading a hand written note that seems simple for human are not trivial even for the most high speed digital computers. In an effort to develop computer that is intelligent as human, software engineers started designing software to mimic the human brain, with its neurons and synaptic connection. Thus the field of artificial neural network was born. Neural Network software analyses data by passing it through several simulated processors instead of using the traditional method of one central processor, to carry out many instructions at a time. The simulated processors are made up of a large number of highly interconnected PEs that are similar to human neurons and tied together with weighted connections that are analogous to synapses. Artificial neural network therefore bears resemblance to the human brain in the following ways:

- It acquires knowledge through learning.
- Its knowledge is stored within inter-neuron connection strength called synaptic weights.

While human beings are good at generalizing from experience, computers usually excel at following explicit instructions over and over. The appeal of neural networks is that they bridge this gap by modeling, on digital computer, the neural connections in human brains.

When used in a well-defined domain, the ability to generalize and learn from data mimics human’s ability to learn from experience. The ability is useful for data mining and it also makes neural networks an exciting area of research promising new and better results in the future (Hornik et al, 1989).

2.2.6.1 The Human Brain

The brain is made up of a large number (over a billion) of neural cells that process information. Each of these neurons is joined to about one hundred thousand, other neurons (Pacific Northwest National Laboratory, 2001; Bose and Liang, 1996). Each cell functions like a simple processor and only the massive interaction between all the cells and their parallel processing makes the brain’s abilities possible. The structure of a neural cell is shown in Fig. 2.3

From the figure, a neuron is made up of a core, dendrites for incoming information and an axon with dendrites for outgoing information sent to linked neurons. Transportation of information between the neurons is in the form of electrical stimulations along dendrites. Incoming information that has reached the neurons dendrite is summed up and then transported along the neurons axon to the dendrites at its end from where the information is transported to other neurons. If the stimulation is beyond a certain threshold, the neuron is said to be activated. On the hand the neuron is said to be inhibited of the stimulation that is coming if it is too low, therefore information will not be transported any further. The connections that exist between the neurons are adaptive, that is, there is dynamic change in the connection structure. It is widely believed that it is this adaptive nature that gives human his learning ability.

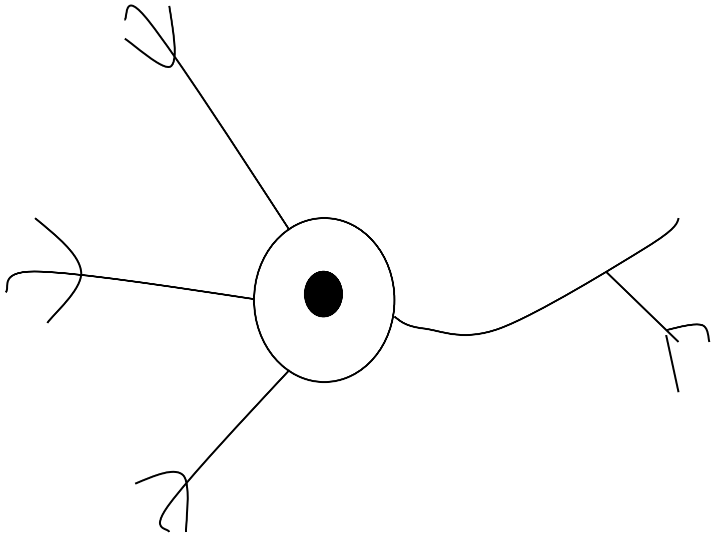


Fig. 2.3: Structure of a neural cell

Similarly, an artificial neural network is also made up of neurons and connection between them. The neurons are transporting incoming information on their outgoing connections to other neurons. These connections are called “weights”.

In order to simulate the electrical information found in the brain, specific values are stored in those weights. The dynamism in the connection structure can also be simulated by changing the weights values.

2.2.6.2 Neural Network Models

Biological neural network is very complex that people still find it difficult to understand it as a whole. Therefore some form of simplification has been made on it, in order to understand its properties and to formulate mathematical models. The under listed are some of the models developed:

- McCulloch –Pitts Model
- Sigmoid Function

(a) McCulloch – Pitts Model

In 1943, Warren McCulloch, a neurophysiologist, and Walter Pits, a logician, postulated a simple model to explain how biological neurons work. While their focus was on understanding the anatomy of the brain, it turned out that this model provided a new approach to solving certain problems outside the realm of neurobiology.

In the 1950s, when digital computers first became available, computer scientists implemented models called perceptron’s based on the work of McCulloch and Pits.

The two researchers, Warrens McCulloch and Walter Pits proposed a model of the neuron as a system for logic computation in any discrete process; be it brain, computer or anywhere else(Bose and Liang, 1996; Principe et al 2000; McCulloch and Pits, 1943). This marks the beginning of effective attempt to mathematically formalize the operation of a neuron (Johnson, 1997).

A McCulloch Pits cell is a very simple two – state machine (Minsky, 1967) from where a single line or wire called the output fiber of the cell emerges from each cell. Output fibers from different cells are not allowed to merge together. Each branch must, in the long run, ends as an input connection to another or perhaps the same cell.

Branch termination can either be in two ways; the one that provides an excitation input or the one that provides an inhibitory input. McCulloch-Pitts model only allows two outputs 0 and 1, which represent either quiet or firing. Further work on the McCulloch-Pitts modeled the Threshold Logic Units (TLUs) with adjustable weights. A TLU is a device with n inputs X_1, X_2 … X_n and an output Y which is a threshold function where

$$Y = \begin{cases} 1 \text{ for } K + 1 \geq 0 \\ 1 \text{ for } K + 1 < 0 \end{cases}$$

There are n + 1 parameters, the weights (W_1, W_2, … W_n) and a threshold θ. Output values is computed by TLU at discrete time indices K = 1, 2 … based on.

$$Y\,(K + 1) = 1$$

$$\begin{cases} \text{If } \sum_{i=1}^{n} W_i X_i(K) \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

If the weights are greater than zero, it means excitatory synapses, but if the weights are less than zero, it means inhibitory synapses. Due to its simplicity, the network of this type can compute any logical functions. McCulloch-Pitts model is very useful in the construction of sequential machines to perform logical operations of any degree of complexity. However, it is only useful for solving linearly separable systems.

(b) Sigmoid Function

Linearly separability as in McCulloch-Pitts model is, it is very impractical for real world problems. A more practical approach is to have a less deterministic function that will allow the use of analogues values so that the result can be differentiated. The solution to this is a sigmoid function. A sigmoid function is a continuous real valued function whose domain is the real number and whose derivation is always positive and whose change is always bounded. A sigmoid function is a smooth and continuous threshold function frequently used as a transfer function in artificial neural networks. It produces a real number for a given input, instead of assigning definite values like (0, 1) as in McCulloch-Pitts models.

Sigmoid functions increase monotonically and provide some form of non-linearity. The non-linearity property of sigmoid function is a necessity of gradient descent. Sigmoid functions can map any input to a finite range of inputs which normally take values between 1 and 0 or -1 and +1.

Training a network on extreme value is of no consequence since sigmoid functions can never reach their maximum or minimum range values. The most common sigmoidal nonlinear functions are the logistic and the hyperbolic tangent functions explained in equations 2.1 and 2.2.

Hyperbolic tangent $F(net) = \tanh(\alpha net)$ (Eq. 2.1)

and

Logistic $F(net) = \frac{1}{1+exp^{-(\propto net)}}$ (Eq. 2.2)

Where

F is a slope parameter and is normally set to 1.

The main different between these two has to do with the range of their output values. The logistic function output values between (0,1), while the hyperbolic tangent output values between [-1, 1].

2.2.6.3 Learning

Due to its nature, derived from the fact that it emulates the brain, artificial neural network exhibit learning properties. Learning is the basis of intelligence. It is a property exhibited when a system's performance is increased at a given task without being reprogrammed.

It is generally believed that during the leaning process, there are changes in the connection structure among the neurons so that stimulations are only accepted by some neurons. This belief comes from the work carried out by Donald Hebb, a neuroscientist (Hebb, 1949). Hebb modeled the collective behavior of biological neurons and concluded that "when an axon of cell A is near enough to excite a cell B and repeatedly or persistently , some growth process or metabolic changes takes place in one or both cells as one of the cell firing B is increased. What this means is that an active synapse that persistently causes the activation of a post synaptic neuron will have its strength increase while others will experience strength decrease.

Mathematically, it means an updating scheme for the connection weight w_{ij} that links neurons i and j together. If x_i and y_i be the input to neuron j from neuron i, and the output form neuron j respectively, then the update equation 2.3:

$$W_{ij}^{new} = w_{ij}^{old} + \propto X_i y_i \qquad \text{(Eq. 2.3)}$$

is called the Hebbian learning rules(Berry et al, 1997), where $\propto$ is called the learning rate parameter and is a positive number.

Furthermore, it means, there exist firm connection between the neural cells that have learned a specific fact. These connections will enable the speedy collection of information. If some information arrives that is related to what it has learned before, the same, neural cells will be stimulated and adapted by changing their connection structure based on the new information. On the other hand, if specific information is not recollected for a long time, the connection structure between the responsible neural cells becomes weaker. For instance, if someone forgot something he had learned before or can only remember it vaguely.

Similarly, artificial neural network is also made of neurons and dendrites but unlike the biological model, a neural network has a fixed structure, built of a specified number of neurons and a specific number of connections between them called weights, which have certain values that changes during learning process. The problem is how to generate a set of weights that generate internal representations of similar patterns to those presented initially. Many of the solutions proposed for the generation of weight are based on Donald Hebb’s ideas.

After neural network had been trained correctly, there is probability that it will generate the correct output to a given input that had been learned, using the weight values. It is not unlikely that certain error is left after the learning process. Therefore the generated result is only a good approximation to the correct output. Learning can either take two forms: supervised and unsupervised.

(a) Supervised Learning

In supervised learning both the input(s) and the output(s) are provided. When the inputs are fed into the network, the network processes the inputs and generates outputs, which are then compared with the desired outputs. An error value is computed which is then propagated back through the network, causing adjustment in weight that control the network. The process is repeated over and over again with the weights adjusted and error value is computed for each iteration until the error is minimized.

(b) Unsupervised Learning

In this type of network, the inputs are provided only, that is without the output. It cannot be determined what the result of the learning process will be or look like. The weight values of such a neural network are arranged inside a certain range, depending on the given input values .This is referred to as self-organizing.

2.2.6.4 Neural Network Types

The type of neural network shows whether the neurons of one of the network’s layers can be connected among themselves. Thus we have feed forward and feedback/recurrent networks.

(a) Feed forward Networks

Feed forward neural network only allows neuron connections between two different layers. They have the advantage that they are very simple and most times, very fast. A number of neural networks model have been built based on feed forward neural networks.

(i) Perceptron

Frank Rosenblatt in 1958 introduced the perceptron as a representation of a neuron using the biological models as foundation. A perceptron is a supervised, feed forward neural network that has one layer of fixed processing unit and trainable threshold logic unit and is capable of computing any Y □ 1, (Φ) for a given Φ by adjusting the weights and threshold of its trainable TLU (Rosenblatt, 1960). The merger of McCulloch and Pitt'smodel proposed a threshold logic model for neurons and that of Hebb said that learning is realized through synaptic changes between neurons. A typical perceptron network is shown in Fig. 2.4.

Single-layer perceptron networks have limitations and are not computationally complete. Complicated logical operations such as exclusive-OR cannot be solved by a perceptron. What it means is that linearly inseparable problems at the output layer cannot be solved by a perceptron.

Rosenblatt's work also included the perceptron convergence theorem (Principe, et al 2000), an algorithm that helps in the training of perceptron units. If a perceptron can train a set of training patterns, this algorithm is guaranteed to converge to a set of weights that enable correct output of the input data. It is used for pattern classification purposes and simple logical operations.

(ii) . Multi-layer Perceptron

Multi-layer Perceptron was first introduced by M. Minsky and S. Papert in 1969. It is a special case of perceptron whose first –layer units are replaced by trainable threshold logic units in order to allow it to solve non-linear separable problem. Minsky and Papert (1969) called multi-layer perceptron of one trainable hidden layer a Gamba perceptron. The structure is shown in Fig. 2.5.

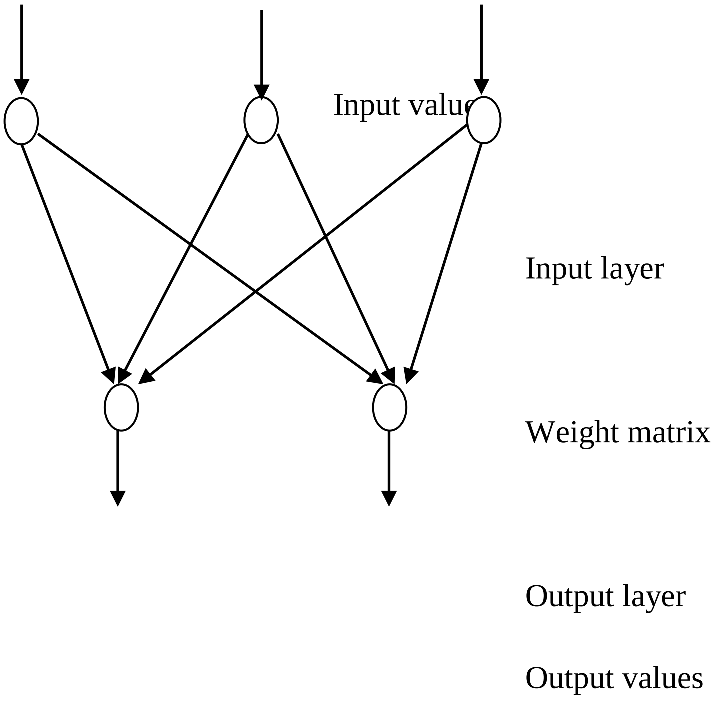


Fig. 2.4: The Structure of a perceptron

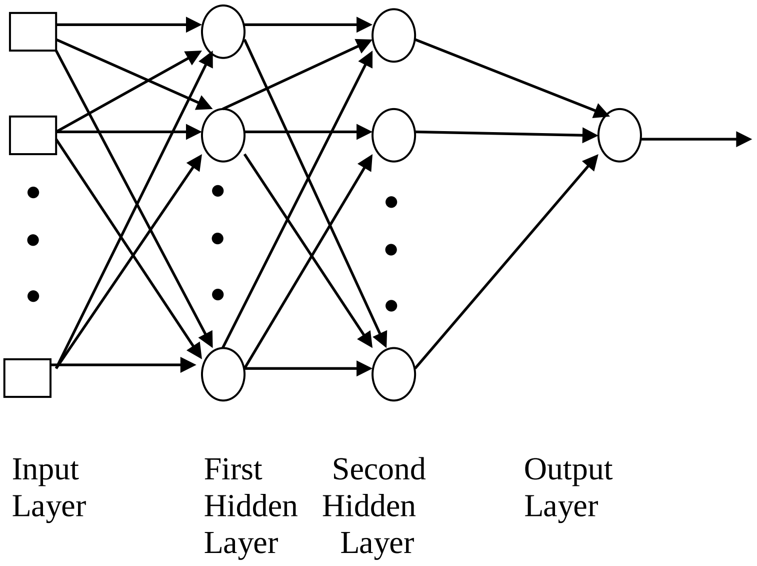


Fig. 2.5: The Structure of Multi-layer Perceptron

Each layer is fully connected to the next one. Depending on the complexity, performance and implementation point of view, the number of hidden layers may be increased or decreased with, corresponding increase or decrease in the number of hidden units and connections.

Both the perceptron and the multi-layer perceptron are trained with error-correction learning. But since perceptron does not have an explicit error available, this stopped further work on multi-layer perceptron around 1970 until a method to train multi-layer perceptron's was later discovered. The method is called back propagation or the generalized delta rule.

With the method, processing is done from the input to the output layer, that is, in the forward direction, following which computed errors are then propagated back in the backward direction, to change the weights to obtain a better result. The algorithm has been rediscovered several times with some variations (Becker and Le Cun, 1989; Jacobs, 1988).

(iii) Back Propagation Net

The Back propagation net was first introduced by Rumelhartet al in 1986 and is one of the most powerful neural net types. It has the same structure as the Multi-layer perceptron and uses the back-propagation algorithm. The back propagation algorithm is general, widely used and not complex, for training multi-layer feed forward networks. It has some disadvantages like low convergence during the learning process.

(b) Feedback or Recurrent Networks

They share the same features with the feed forward network, but with feedback connectionist becomes a non-linear system. As a result of these feedback connections, cycles are present in the network, and training is sometimes iterated for a long time before a result can be generated. A number of networks based on recurrent network have been developed.

(i) Hopfield Network

The Hopfield Net was first introduced by J.J. Hopfield in 1982 and belongs to neural net types, which are called “thermodynamical models”. It consists of a set of neurons where each neuron is connected to each other neuron. There is no difference between input and output neurons. The main application of a Hopfield net is the storage and recognition of pattern, e.g. image files.

Hopfield network do not have any natural applications but many hardware implementations have been reported in (Bose and Liang, 1996; Hopfield, 1982).

(ii) Simulated Annealing

Hopfield network is based on the gradient descent rule which has the disadvantage that once it gets to a local minimum, it is trapped there. Therefore some randomness needs be added to gradient descent rule for the system to jump out of a local minimum. Simulated annealing (Kirkpatrick, et al, 1983) is a method that does this.

It bores analogy to the technique used in metallurgy where a metal is annealed by heating it to a high temperature thereby making the atoms to shake violently. If the metal is cooled suddenly, the randomness would be locked in its state making the metal extremely brittle. If on the other hand, it was cooled slowly, the molecules will become relatively stable for the current temperature.

If this is applied to the neural network state by reducing the amount of noise, it can find its way out of a local minimum. This method is called Boltzmann machine, named after

Ludwig Boltzmann, the Australian physicist that discovered properties similar to those of annealing in gas molecules.

(iii) Kohonen Maps

Finish Professor Teuvo Kohonen in 1982 demonstrated the formation of a topography map by unsupervised self-organization. It is made of a feature map and a neuron layer which arranged themselves according to some input values.

Kohonen identified two key mechanisms for a network to self-organize spatially:

1. Locate the unit that best responds to the given input. This unit is called the winning unit.
2. Modify the connections to the winning unit and connections to units in its neighbourhood.

Kohonen feature map is probably the most useful neural network type. The structure is shown in Fig. 2.6.

2.2.7 Genetic Algorithm

Genetic Algorithms, first introduced by (Holland, 1975), have been applied to a variety of problems and offer intriguing possibilities for general purpose adaptive search algorithms in artificial intelligence, especially for situations where it is difficult or impossible to precisely model the external circumstances faced by the program. Search based on evolutionary models had been tried before Holland. However, these models were based on mutation and natural selection and were not notably successful. The principal difference of Holland’s approach was the incorporation of a ’crossover’ operator to mimic the effect of sexual reproduction Figure 2.7 illustrates the basic idea of GA.

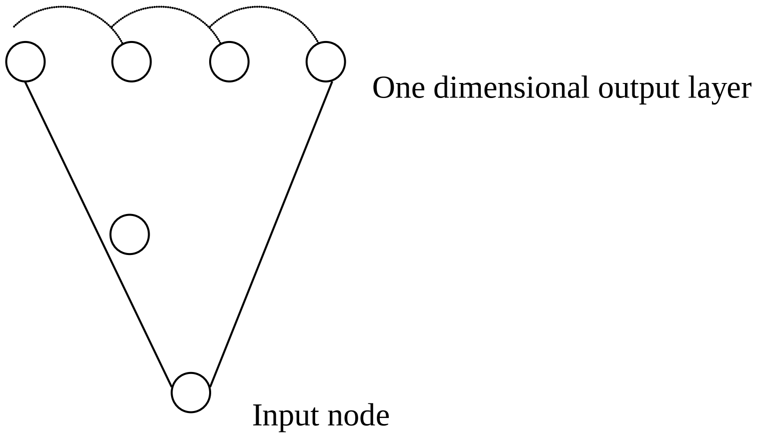


Fig. 2.6: One Dimensional Self Organizing Map

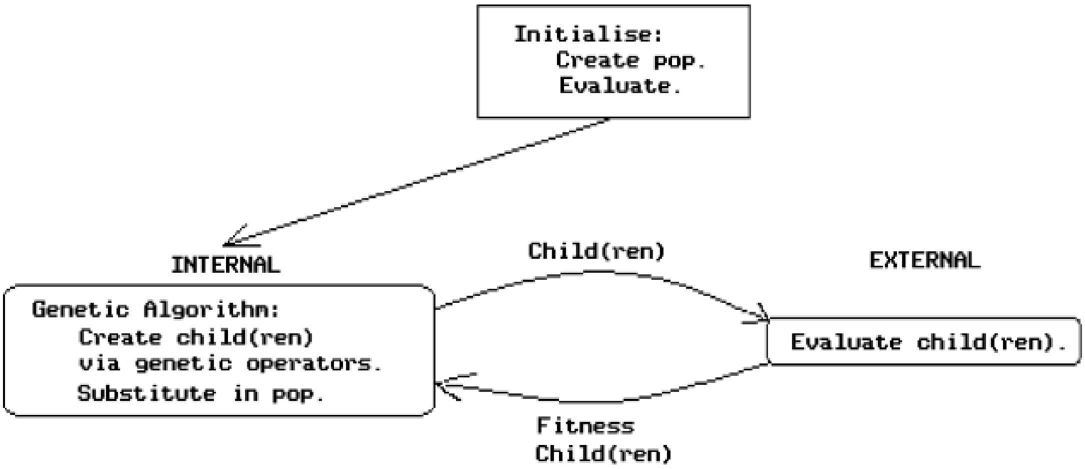


Figure 2.7: Generic Model for Genetic Algorithm

Genetic algorithms are based on an analogy similar to biological processes. Evolution and natural selection have resulted in adaptable, specialized species that are highly suited to their environments. Evolution optimizes the fitness of individuals over succeeding generations by propagating the genetic material in the fittest individuals of one generation to the next generation.

Genetic algorithms, which are also called evolutionary algorithms, have been applied to optimization in large factories, and classification problems involving complex data types.

Optimization problems have three features:

- A set of parameters (which for GAs are called genomes or chromosomes)
- A function that combines the parameters into single value (the fitness function)
- A set of constraints on the parameters (for GAs these are incorporated into the fitness function)

The goal is to find the parameters that maximize or minimize the fitness function, subject to the constraints. Searching through all combinations of parameters that meet the constraints is too cumbersome for even the most advanced computers; even for a small number of parameters, the number of combination is too large to search.

Genetic algorithms are one approach to solving such problems. When the fitness function satisfies some specific mathematical conditions, then differential calculus can be used to find the optimal solution.

In the world of data mining and data analysis, the use of genetic algorithms has not been as widespread as the use of other techniques. Data mining focuses on tasks such as classification and prediction, rather than on optimization.

Genetic algorithms are mathematical procedures that utilize the process of genetic inheritance. They have been usefully applied to a wide variety of analytic problems. Data mining can combine human understanding with automatic analysis of data to detect patterns or key relationships. Given a large database defined over a number of variables, the goal is to efficiently find the most interesting patterns in the database. Genetic algorithms have been applied to identify interesting patterns in some applications. They usually are used in data mining to improve the performance of other algorithms such as decision tree algorithms and association rules.

Genetic algorithms require certain data structure. They operate on a population with characteristics expressed in categorical form. **The analogy with genetics is that the population (genes) consists of characteristics (alleles)**. One way to implement genetic algorithms is to apply operators (reproduction, crossover, selection) with the feature of mutation to enhance generation of potentially better combinations. The genetic algorithm process is thus:

1. **Randomly** select parents.
2. **Reproduce** through crossover. Reproduction is the operator choosing which individual entities will survive. In other words, some objective function or selection characteristic is needed to determine survival. Crossover relates to changes in future generations of entities.
3. **Select survivors** for the next generation through a fitness function.
4. **Mutation** is the operation by which randomly selected attributes of randomly selected entities in subsequent operations are changed.
5. **Iterate** until either a given fitness level is attained, or the preset number of iterations is reached.

Genetic algorithm parameters include population size, crossover rate (the probability that individuals will crossover), and the mutation rate (the probability that a certain entity mutates).

Genetic algorithms are very easy to develop and to validate, which makes them highly attractive. The algorithm is parallel, meaning that it can be applied to large populations efficiently. The algorithm is also efficient in that if it begins with a poor original solution, it can rapidly progress to good solutions. Use of mutation makes the method capable of identifying global optima even in very nonlinear problem domains. The method does not require knowledge about the distribution of the data.

Genetic algorithms require mapping data sets to a form where attributes have discrete values for the genetic algorithm to work with. This is usually possible, but can lose a great deal of detail information when dealing with continuous variables. Coding the data into categorical form can unintentionally lead to biases in the data.

There are also limits to the size of data set that can be analyzed with genetic algorithms. For very large data sets, sampling will be necessary, which leads to different results across different runs over the same data set.

2.2.7.1 Basic Genetic Algorithm Programming steps are:

(1) **Start**: Generate random population of n chromosomes (suitable solutions for the problem)

(2) **Fitness**: Evaluate the fitness f(x) of each chromosome x in the population

(3) **New population**: Create a new population by repeating following steps until the new population is complete

(4) **Selection**: Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)

(5) **Crossover**: With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.

(6) **Mutation**: With a mutation probability mutate new offspring at each locus (position in chromosome).

(7) **Accept**: Place new offspring in a new population

(8) **Replace**: Use new generated population for a further run of algorithm

(9) **Test**: If the end condition is satisfied, stop, and return the best solution in current population

(10) **Loop**: Go to step 2

Genetic algorithms have three basic operators:

- **Selection**
- **Crossover and**
- **Mutation**

(a) Selection

This is the procedure for choosing individuals (parents) on which to perform crossover in order to create new solutions. The idea is that the ‘fitter’ individuals are more prominent in the selection process, with the hope that the offspring they create will be fitter.

The selection step is analogous to the process of natural selection where only the fittest individual in the population survive to pass genetic material on to the next generation.

Unlike nature, though, the size of the population remains constant from one generation to the next, so there is no chance of the population becoming extinct. The chance of a genome surviving to the next generation is proportional to its fitness value—the better the fitness value relative to other genomes, the more copies that survive to the next generation. Survival is based on choosing the genomes in a random way proportional to their fitness. A random number is generated between 0 and 1 and this random number is used to determine whether a copy of a genome survives or not. More random numbers are generated until the next generation has the right number of genomes. Using a random number generator converts the fractional probabilities to whole number approximations, and it also allows some genomes with low fitness to survive.

Two commonly used procedures are ‘roulette wheel’ and ‘tournament’ selection. In roulette wheel, each individual is assigned a slice of a wheel, the size of the slice being proportional to the fitness of the individual. The wheel is then spun and the individual opposite the marker becomes one of the parents. In tournament selection several individuals are chosen at random and the fittest becomes one of the parents.

(b) Cross over

The next operator applied to the surviving genomes is crossover. Crossover, which occurs in nature, creates two new genomes from two existing ones by gluing together pieces of each one. Crossover starts with two genomes and a random position. The first part of one genome swaps places with the first part of the second. The first part of one genome swaps places with the first part of the second. For instance, starting with the two genomes 10110 and 00010 and using a crossover position between the second and third position works as follows:

10 | 110

00 | 010

The result of crossover is (the genes from the second genomes are underlined):

10 | $\underline{010}$

00 | $\underline{110}$

The resulting genomes, called children, each have a piece of their chromosomes inherited form each of their parents. Applying crossover to large population proceeds by selecting pairs of genomes and flipping a coin to determine where they split and swap. This probability is the crossover probability denoted by p. If they do cross over, then a random position is chosen and the children of the original genomes replace them in the next generation. Multiple point crossovers are where this occurs at several points along the string. A crossover probability (Pc) is often given which enables a chance that the parents descend into the next generation unchanged.

A value of 0.5 for the crossover probability (corresponding to a coin toss) generally produces good results. In the example, the two genomes 10110 and 00010 are chosen for crossover, and the position is between the second and third genes. Along with mutation, crossover is the operator that creates new candidate solutions.

(C) Mutation

The final operation is mutation. Mutation rarely occurs in nature and is the result of miscode genetic material being passed from a parent to a child. The resulting change in the gene may represent a significant improvement in fitness over the existing population, although more often than not, the results are harmful. Selection and crossover do a good job of searching the space of possible genomes, but they depend on initial conditions and randomness that might conspire to prevent certain valuable combinations from being considered in succeeding generations.

Mutation provides the additional input. Mutation rate is quite small in nature and is usually kept quite low for genetic algorithms. When a mutation occurs, a bit changes from a 0 to a 1 or from a 1 to a 0.Mutation is a second-order effect that helps avoid premature convergence to a local optimum. When the initial population provides good coverage of the space of possible combinations, succeeding generations move quickly toward the optimal solution by means of selection and crossover. Changes introduced by mutation are likely to be destructive and do not last for more than a generation or two.

2.3 Classification of Data Mining Tasks

Data mining tasks can be categorized based on the objectives of the work at hand. A few of such categorizations are:

- **Exploratory Data Analysis (EDA)**: Here the goal is simply to explore the data without any clear ideas of what is being looked for. Typically, EDA techniques are interactive and visual.
- **Descriptive Modeling**: The goal of a descriptive model is to describe all of the data (or the process generating the data). Examples of such descriptions include models for the overall probability distribution of the data (density estimation), partitioning of the p-dimensional space into groups (cluster analysis and segmentation), and models describing the relationship between variables (dependency modeling).

- **Predictive Modeling classification and Regression**: Here, the aim is to build a model that will permit the value of one variable to be predicted from the known values of other variables. In classification, the variable being predicted is categorical, while in regression the variable is quantitative.
- **Discovering Patterns and Rules**: Three types of tasks listed above are concerned with model building; while discovering of patterns and rules are concerned with pattern detection. For example spotting fraudulent behavior by detecting regions of the space defining the different types of transactions where the data points significantly different from the rest.
- **Retrieval by Content**: Here the user has a pattern of interest and wishes to find similar patterns in the data set. This task is most commonly used for text and image data sets. For text, the pattern may be a set of keywords, and the user may wish to find relevant documents within a large set of possibly relevant documents(e.g. Web pages).

Data mining can accomplish a limited set of tasks and only under limited circumstances. Many problems of intellectual, economic and business interest are also classified in terms of the following six tasks. (Berry et al, 1997).

1. Classification
2. Estimation
3. Prediction
4. Affinity grouping
5. Clustering
6. Description

- **Classification**: This is the most common data mining task. In order to understand and communicate about the world, we constantly classify, categorize and grade. Classification consists of examining the features of a newly presented object and assigning it to one of a predefined set of classes. Examples of classification tasks are assigning key words to articles, and classifying applicants as qualified or not qualified. Decision trees and nearest neighbor, techniques are good for classifying tasks.
- **Estimation**: While classification deals with discrete outcomes, estimation deals with continuous outcomes. Given some input data, estimation comes up with a value from some unknown continuous variables such as height or income. In practice estimation is often used to perform a classification task. Examples of estimation tasks include estimating the number of children in a family and estimating a family's total household income. Regression models and neural networks are well suited to estimate tasks. Survival analysis is good to estimate tasks where the goal is to estimate the time to an event.
- **Prediction**: This task is the same with classification or estimation except that records are classified according to some predicted future behavior or estimated future value. In prediction, the only way to check for accuracy of the classification is to wait and see the outcome. Prediction is treated separately because in predictive modeling there are additional issues regarding the temporal relationship of the input variables or predictors of the target variable. Any method used for classification and estimation can be adapted for use in prediction by using training examples where the value of the variable to be predicted is already known along with historical data for those examples. Virtually all data mining techniques can be used for prediction. The choice of

the technique depends on the nature of the input data, the type of value to be predicted and the importance attached to the explicability of the prediction.

- **Affinity Grouping**: The task of affinity grouping is to determine things that go together. Affinity is used to determine things that go together in a shopping cart at a supermarket. Affinity can also be used to identify cross-selling opportunities and to design attractive packages or groupings of product and services. Affinity grouping is a simple approach to generating rules from data. Association rules and Market Basket Analysis are well suited for affinity grouping tasks.

- **Clustering**: Clustering is the task of segmenting a heterogeneous population into a number of more homogenous subgroups or clusters. What distinguishes clustering from classification is that clustering does not rely on predefined classes. Cluster detection and self-organizing maps are techniques good for clustering tasks.

- **Description and Profiling**: Sometimes the purpose of data mining is simply to describe what is going on in a complicated database in a way that increases our understanding of the people, products, or processes that produced the data in the first place. A good description of a behavior will often suggest an explanation for it as well. Decision trees, Association rules and Cluster detections are good techniques for descriptive and profiling tasks.

The first three tasks classification, estimation and prediction are examples of directed data mining. Affinity grouping and clustering are examples of undirected data mining. Profiling may be either directed or undirected. In directed data mining there is always a target variable, a variable to be classified, estimated or predicted. In undirected data mining, there is no target variable. The task here is to find overall patterns that are not tied to any one variable. Undirected data mining is descriptive in nature, so undirected data are often used for profiling, while directed are used for building profiles. Directed data mining is called supervised learning and undirected is called unsupervised learning. Data mining process is sometimes referred to as knowledge discovery or knowledge discovery in databases.

In knowledge discovery, no prior assumptions are made; the data is allowed to speak for itself.

Knowledge discovery can be either directed or undirected. In direct knowledge discovery, the task is to explain the value of some particular field (income, response, age, credit worthiness etc.) in terms of all the others. We select the target field and direct the computer to tell us how to estimate, classify or predict it. In undirected knowledge discovery there is no target field. The computer is simply asked to identify pattern in the data that may be significant.

Directed knowledge discovery is goal – oriented. There is a specific field whose value is to be predicted, a fixed set of classes to be assigned to each record, or a specific relationship to explore.

In undirected knowledge discovery there is no target field. The data mining tool is simply let loose on the data in the hope that it will discover meaningful structure. One common use for undirected knowledge discovery is market basket analysis that asks what items sell "together". Another application is clustering, where groups of records are assigned to the same cluster if they have something in common.

2.4 The Importance and Uses of Data Mining in Medicine and Public Health

There are several reasons that support the use of data mining in the health sector, covering not just concerns of public health but also the private health sector, some of which are discussed below.

Data overload: There is a wealth of knowledge to be gained from computerized health records. The overwhelming bulk of data stored in these databases makes it extremely difficult, if not impossible, for humans to sift through it and discover knowledge (Cheng, et al 2006).

Some believe that medical breakthroughs have slowed down because of the prohibitive scale and complexity of present-day medical information. Computers and data mining are best-suited for this purpose. (Shillabeer and Roddick, 2007).

Evidence-based medicine and prevention of hospital errors: When medical institutions apply data mining on their existing data, new, useful and potentially life-saving knowledge that otherwise would have remained inert in their databases may be discovered. For instance, an ongoing study on hospitals and safety found that about 87% of hospital deaths in the United States could have been prevented, had hospital staff (including doctors) been more careful in avoiding errors (Health Grades Hospitals Study, 2007).

By mining hospital records, such safety issues could be flagged and addressed by hospital management and government regulators, before the occurrence of such errors.

Policy-making in public health: Lavrac et al. (2007) combined GIS and data mining using among others, WEKA with J48 (free, open source, Java-based data mining tools), to analyze similarities between community health centers in Slovenia. Using data mining, they were able to discover patterns among health centers that led to policy recommendations to their Institute of Public Health. They concluded that “data mining and decision support methods, including novel visualization methods, can lead to better performance in decision making.”

More value for money and cost saving: Data mining allows organizations and institutions to get more out of existing data at minimal extra cost. Data mining have been applied to discover fraud in credit cards and insurance claims (Kou et al. 2004). By extension, these techniques could also be used to detect anomalous patterns in health insurance claims, particularly those operated by the national healthcare insurance system.

Early detection and/or prevention of diseases: Cheng, et al cited the use of classification algorithms to help in the early detection of heart disease, a major public health concern all over the world. Cao et al (2008) described the use of data mining as a tool to aid in monitoring trends in the clinical trials of cancer vaccines. By using data mining and visualization, medical experts could find patterns and anomalies better than by just looking at a set of tabulated data.

Early detection and management of pandemic diseases and public health policy formulation: Health experts have also begun to look at how to apply data mining for early detection and management of pandemics. Researchers have outlined techniques combining spatial modeling, simulation and spatial data mining to find interesting characteristics of disease outbreak.

The analysis that resulted from data mining in the simulated environment could then be used towards more informed policy-making to detect and manage disease outbreaks. Wong et al. (2005) introduced WSARE, an algorithm to detect outbreaks in their early stages. WSARE, which is short for “What’s Strange about Recent Events” is based on association rules and Bayesian networks. Applying WSARE, simulation models have produced relatively accurate predictions of simulated disease outbreaks.

Non-invasive diagnosis and decision support: Some diagnostic and laboratory procedures are invasive, costly and painful to patients. An example of this is conducting a biopsy in women to detect cervical cancer. Thangavel et al (2006) used the K-means clustering algorithm to analyze cervical cancer patients and found that clustering found better predictive results than existing medical opinion. They found a set of interesting attributes that could be used by doctors as additional support on whether or not to recommend a biopsy for a patient suspected of having cervical cancer.

This work therefore used data mining technique to develop a model that is capable of classifying and predicting HIV/AIDS status of patients.

2.5 HIV/AIDS

HIV is a retrovirus that infects cells of the immune system, such as CD4 cells and macrophages, and then destroys or impairs their function. CD4 cells organize the body’s overall immune response to foreign bodies and infections.

Immune deficiency arises from the progressive depletion of the immune system through this infection. Since the immune system is responsible for fighting off infection and cancers, cellular immune deficiency makes individuals more susceptible to opportunistic infections such as pneumocystis carinii pneumonia, toxoplasmosis, systemic and esophageal candidiasis, generalized herpes zoster, Cryptococcimeningitis, and to cancers such as Kaposi sarcoma.

2.5.1 Mode of Operation and Course of Infection

First the virus penetrates the CD4 cells and copies the cell’s DNA to ensure that it cannot be identified and destroys the immune system. The virus replicates many times within the cell, and these new particles destroy the CD4 cell when they emerge. Each of the new viruses infects other cells. During the early stages of infection, many cells are infected and the number of virus particles in the body is high.

The HIV status cannot be detected through tests, as insufficient antibodies have been formed, and this is called the window period. After this primary acute infection a prolonged period without obvious symptoms follows. Later the body experiences severe immunodeficiency resulting in secondary opportunistic infections that are the major causes of death in AIDS patients.

2.5.2 Methods of Transmission

There are three main methods of transmission:

- Sexual contact

- Contaminated blood: sharing of contaminated needles in drug injection; sharing infected blood through blood transfusion

- Mother to child transmission during pregnancy, childbirth and breastfeeding

The predominant mode of transmission of HIV is sexual. According to UNAIDS, three factors influence the biological probability of transmission:

- **Type of sex**: Anal intercourse carries a greater risk than vaginal intercourse, and receptive anal sex is more risky than insertive. The probability of HIV transmission is higher if there are lesions, such as would arise from rape and rough sex. The virus tends to be more easily transmitted from males to females, and the risk of male to female transmission is higher in girls younger than 16, compared to older women before the menopause. This higher biological vulnerability could be due to the immaturity of the genital tract and cervix. Risk of transmission of the virus through oral sex is small, but this is increased if there are abrasions in the mouth.

- **Stage of illness**: HIV infected individuals are more infectious during the earliest phase of infection before antibodies are produced; and at the later phase of the disease when the immune system is unable to combat the virus.
- **Sexually transmitted disease**: A person with an untreated sexually transmitted infection (STI) is, on average, six to 10 times more likely to pass on or acquire HIV during sex. An STI means there is more chance of broken skin or membranes allowing the virus to enter or leave the body.

2.5.3 Diagnosis of HIV and AIDS

Clinical diagnosis of AIDS is difficult, as detection of HIV is limited in the early stages of the infection. Usually, a patient is suspected of AIDS when patients exhibit certain symptoms and suffer from opportunistic infections.

It is possible to detect the HIV antigen (the virus itself) during the period when there are high levels of circulating virus particles, but the period is short, and the level of antigen declines until it is undetectable. Antibodies in the blood are usually detected as evidence of the virus, as they are cheaper and easier to detect than the virus itself.

This can only be done by the current tests at the end of the window period. The two primary blood tests are the enzyme linked immunosorbent assay (ELISA) Test, and the Western blot assay used to confirm a positive ELISA test result. The accuracy of diagnostic tests is measured according to sensitivity and specificity. High sensitivity indicates that the test is able to detect the presence of antibodies, i.e. minimizing false-negative results. On the other hand, a test with high specificity identifies all negatives correctly, producing no false positives. There is a margin of error, so the tests must be selected according to the different purposes, and used in different combinations. There is currently no treatment for eradicating the virus, but the existing treatments are able to control virus replication causing a reduction in HIV virus load in the blood. Such treatments are a combination of different antiretroviral drugs, known as Highly Active Antiretroviral Therapy (HAART).

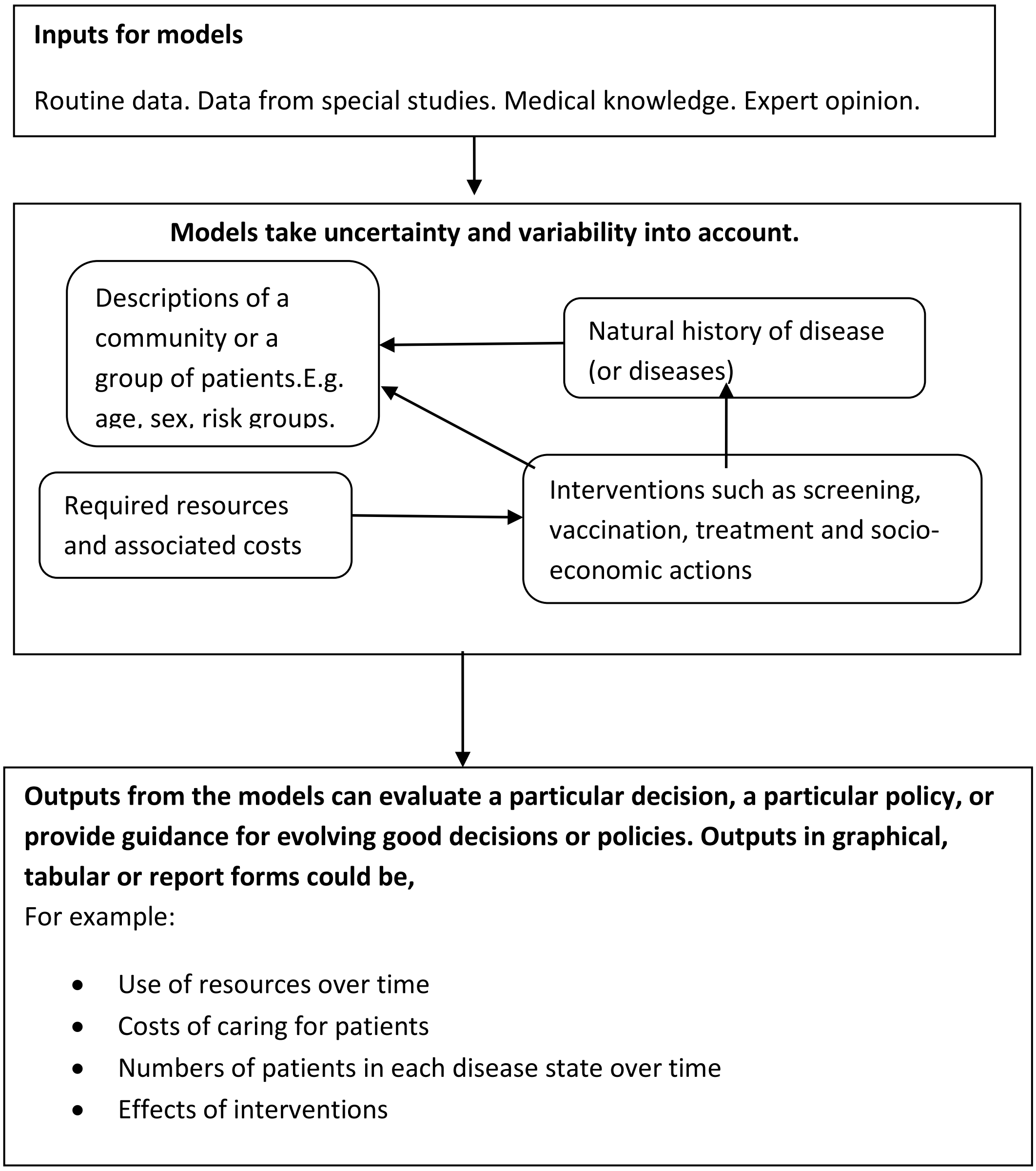
This work seeks to use Data Mining technique to facilitate early detection of the disease and prevention of its spread. Prevention of its spread will go a long way in combating its menace.

2.6 RELATED WORKS

In order to identify gap in the domain of this work, a lot of literature search of previous works were carried out. This section discusses our findings.

Harper and Shahani (2003), made an effort to develop a decision support system for the care of HIV and AIDS patients. They developed a model that was intended for practical use to aid decision making. The model was developed to deal with complexity and uncertainty, variability and inadequate data. The modeling work involved collaboration between operational researchers and medical people to ensure that the result is a practical useful tool that can achieve maximum benefits.

A suitable methodology of operational modeling for disease developed is illustrated in the figure 2.8



Inputs for models

Routine data. Data from special studies. Medical knowledge. Expert opinion.

Models take uncertainty and variability into account.

Descriptions of a community or a group of patients.E.g. age, sex, risk groups.

Natural history of disease (or diseases)

Required resources and associated costs

Interventions such as screening, vaccination, treatment and socio-economic actions

Outputs from the models can evaluate a particular decision, a particular policy, or provide guidance for evolving good decisions or policies. Outputs in graphical, tabular or report forms could be,
For example:

- Use of resources over time
- Costs of caring for patients
- Numbers of patients in each disease state over time
- Effects of interventions

Fig: 2.8: Methodology of Operational Modeling

A framework for operational modeling of diseases was adapted from (Shahani et al., 1994). Within this flexible framework, patient groups, natural history, interventions and resources are linked together as necessary.

The progress of infection overtime was modeled by defining suitable states, which may be experienced by a particular HIV infected patient. Such, states constitute the natural history of the disease.

The following information were to make the model operational:

- The length of stay of a patient in any disease state.
- For each disease state, the set of possible states that a patient can transfer to.

The simplest HIV natural history model is illustrated in figure 2.9.

A patient will move through this model in a stepwise fashion that is, moving from left to right one step at a time. This model becomes operational once patient’s dwelling times in each of the HIV + and AIDS states have been defined and incorporated into the model (Brailsford, 1993).

Interventions such as the provision of life-prolonging drugs, may be modeled through changes to the transition times between states, for example by increasing a patient’s dwelling time in the HIV + state before making the transition to AIDS.

Operational modeling of infectious diseases began in the 1970s. But much of this early modeling was deterministic in nature where no provision for variability and uncertainty was made (Anderson et al., 1986, Islam Q, 1988).

These models typically consist of a few disease states and sets of differential equations and can be solved exactly by mathematical means or numerical methods. Such models fail to capture the complexities of diseases like HIV that renders them useless in practice.

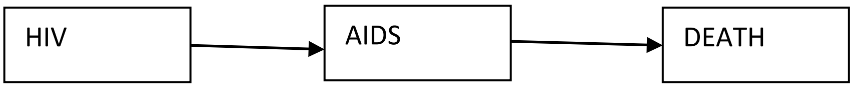


Fig. 2.9: A simple three state for HIV/AIDS progression.

They overcome variability by using average values, which although intuitively attractive when considering large communities, have the effect of significantly biasing results in an environment of non-linearity and variability.

Unlike deterministic models, stochastic model provides a more accurate and realistic model by incorporating uncertainty and variability through the use of probabilities and random variables of HIV/AIDS modeling work, the necessary stochastic descriptions involve discrete states and continuous time. Two stochastic processes have to be considered, namely Markov and Semi-Markov processes.

Under the Markov hypothesis, the probability of a patient’s transition from one state of the HIV infection to the next state depends only on the current state occupied and not on the previous history. From a medical stand point, this Markov assumption is flawed since a patient’s medical history is likely to influence their future disease progression.

In a Semi-Markov process, the transition from one state to a state is governed by the probability transiting and a random duration of occupying the first state before the transition to the next stage. A Semi-Markov process is a better description of the degenerative biological processes that follow HIV infection.

The analytical solution of stochastic models presents a formidable challenge. Unless restricting assumptions are applied, many stochastic models are impossible to solve analytically, numerical methods are needed for solving realistic stochastic models. However, a computer simulation need not make such stringent simplifying assumptions and is an ideal tool for modeling disease.

The aim of the simulation is the solution of a model that mimics the behaviour of a real-life system overtime. The developed simulation model was written using a three-phase approach.

The pioneering approach was developed by Tocher (1963) and it combined simplicity of the activity-based simulation approach with the efficiency of the event-based approach. The operational research group had a three-phase simulation shell called TOCHSIM (Hawkins, 1992) that consisted of skeleton procedures for handling the queues used to hold the calendar of events and entities in each state. Procedures to sample from various probability distributions were also included in their work. This work identified the strength and the weaknesses of the earlier method for management of HIV/AIDS. Such work includes deterministic modeling that can be solved by mathematical means or numerical methods. The advantages of stochastic models over deterministic models as well as the weaknesses of stochastic model were also identified. In addition the work proposed and developed a computer simulation model for a decision support system HIV/AIDS management which overcame the problem generated by deterministic and stochastic modeling. The operational computer model was used to predict the future numbers of patients and associated costs. This model is flexible, easy to use and is a useful tool that might aid decision making by clinicians and managers and thus helping the provisions of effective and efficient care to many HIV/AIDS patients. However the work is limited in that incidence is considered as an input to the system. With limited prevalence data, calculating the likely growth of the epidemic is difficult. The model also cannot extract hidden patterns for future decision making like in prediction of HIV/AIDS status and prediction of drug pattern. Otine (2012), developed a framework for adopting knowledge engineering in information system for monitoring HIV/AIDS patients. An open source approach was adopted due to the resource constrained context of the study to ensure a cost effective and sustainable solution. The work focused on the developments of models for data warehouse and data mining for monitoring HIV/AIDS patients and antiretroviral therapy. The research also involved a situation analysis of HIV in healthcare, different health care information systems; it also covered the development of a knowledge base system, its simulation and testing.

Data mining simulations was done on the data warehouse out of which two learning algorithms (regression and classification) were developed and tested using data from the data warehouse. The algorithms were used to predict viral load from CD4 count test figures and to classify cases of treatment failure.

The work is an open source dimensional model for monitoring antiretroviral therapy that provided an architecture showing the integration of different knowledge engineering components like a data warehouse, data mining platform and user interaction. It also provided a cost effective data mining model for HIV patient monitoring.

Though this work proposed a cost effective data mining model for HIV patient monitoring the under listed were its weaknesses.

- Performance accuracy of the model was not measured.
- The model focused majorly on the monitoring of HIV/AIDS patients and the effectiveness of antiretroviral therapy.

Our work took care measuring of performance accuracy.

Lilly and Balalubramanle, (2009) proposed a Multi-Layer Feed Backward neural network(MLFB) model for medical decision support, implemented with back propagation algorithm in HIV/AIDS Regimens. The work described the regimen specification for the HIV/AIDS patients based on the patient’s unique factors like age, weight, HB, CD4. Using the MLFB back propagation algorithm, the regimens specification for the patient was calculated to approximately predict how long the patient can prolong his/her life with these regimens. This model was trained with 200 patient’s medical information as input and the regimens are the weights of the network. The network was adjusted with the threshold 0.001, so as to minimize error. The mapping and adjustment continued until all mapping example from the training set were learned.

The work contributed immensely to the use of data mining technique for managing HIV/AIDS. However it did not take care of the issue of local minima depicted in figure 2.10 during training time. Thus, there is no assurance of convergence to the right solution and the problem of over-fitting may arise. Our work addressed the issue of local minima while using back propagation algorithm.

Manaswini et al (2011) developed a new method of Multilayer Perceptron (MLP) network to classify HIV/AIDS infected and non-infected status of individuals. In their work, seven features on the basis of patient’s unique factors like age, sex, weight, HB, CD4, CD8 and TB were used as input data. To determine the applicability and best performance of the MLP network, three different training algorithms like Back-propagation, Levenberg-Marquardt and Bayesian Rule algorithms were employed to train the MLP networks. The result also significantly demonstrated the suitability of the MLP network for calculating and specifying the HIV/AIDS positive/negative status of the patient.

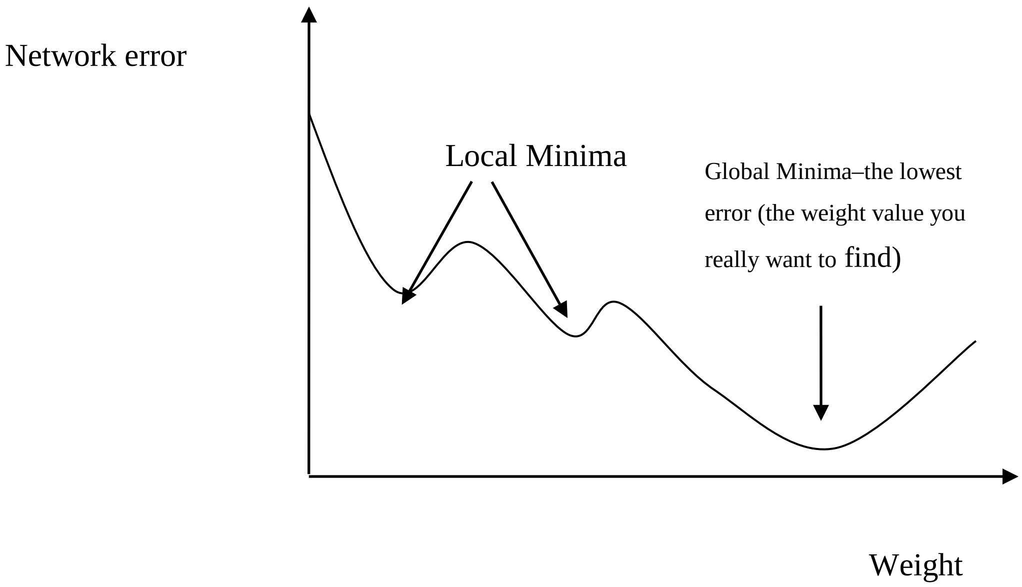

Network error
Local Minima
Global Minima–the lowest error (the weight value you really want to find)
Weight

Fig 2.10: Local Minima

This work clearly showed the strength of MLP network in prediction using three algorithms and concludes with concrete results that the back-propagation algorithm provides the highest accuracy rates and can be used on the context of HIV/AIDS classification/prediction. It however did not take care of low convergence of back-propagation which always lead to over-generalization and can make the results unreliable and undependable for classification/prediction

Sibanda and Pretorius,(2012) reviewed different papers on Applications of Neural Networks in the Modeling of HIV Epidemic. Different work on the application of neural networks to the HIV Modeling was reviewed. It was highlighted that research on HIV modeling falls into four broad areas:

a. Behavioural research
b. Diagnostic research
c. Vaccine research
d. Biomedical research

Most of the research publication reviewed in this work emanated from the four broad HIV research areas and will be presented in three categories namely:

- Prediction
- Classification
- Function approximation

This work highlighted the overview of Artificial Neural Network (ANN), types of neural networks and also shows a clear explanation on training a neural network.

Our work falls within the prediction and classification categories.

Vararuk. et al (2008) proposed a model on Data mining techniques for HIV/AIDS data management. This work investigated, through the use of data mining techniques, patterns in HIV/AIDS patient data. These patterns can be used for better management of the disease and more appropriate targeting of resources.

Design/methodology/approach used consists of a total of 250,000 anonymous records from HIV/AIDS patients in Thailand that were imported into a database. IBM's Intelligent Miner was used for clustering and association rule discovery.

Clustering was used to highlight groups of patients with common characteristics and also errors in data. Association rules identified associations that were not expected in the data. The model also allowed for the identification of symptoms that co-exist or are precursors of other symptoms.

Originality/value – Identification of symptoms that are precursors of other symptoms can allow the targeting of the former so that the later symptoms can be avoided. This study shows that providing a pragmatic and targeted approach to the management of resources available for HIV/AIDS treatment can provide a much better service, while at the same time reducing the expense of that service. This study can also be used as a means of implementing a quality monitoring system to target available resources. This model however cannot be used for classification and prediction of HIV/AIDS.

Larder, et al. (2008) described the application of ANNs for decision support in medicine. The model takes treatment change episodes (TCEs) and HIV drug resistance mutations as basic input variables to train ANN models. In conclusion, the authors of this paper define the reliability of ANN predictions for HIV patients receiving routine clinical care. The paper then compared the utility of ANN models to identify effective treatments for patients failing therapy with that of the rules-based methods currently in use. They confirmed that ANN model gave better predictions and performance than that of rule-based methods.

Dechao et al. (2009) compared three computational methods for the prediction of virological response to combination HIV therapy. HIV treatment failure is commonly associated with drug resistance and the selection of a new regiment is often guided by genotypic resistance testing. The interpretation of complex genotypic data poses a major challenge. In that regard the authors developed artificial neural network (ANN) models that predicted virological response to therapy from HIV genotype and other clinical information. The accuracy of ANN was compared with alternative modelling methodologies such as random forests (RF) and support vector machines (SVM). In conclusion, the researchers noted that RF and SVM models can produce predictions of virological response to HIV treatment that are comparable in accuracy to a committee of ANN models, however combining the predictions of different models further improved their accuracy.

Lamer, et al. (2008) demonstrated the use of ANN trained using evolutionary computation to predict R5, X4, and R5X4 HIV-1 co-receptor usage. The results indicated identification of R5X4 viruses with predictive accuracy of 75.5%.

Cai et al. (1998) studied the application of ANN method for predicting HIV protease cleavage sites in protein. The authors developed Kohonen’s self-organisation model. Kohonen’s self-organization neural network is a two-layer network. Output nodes are arranged regularly on a planar mapping grid. Each input node is connected to every output node via a variable connection weight. The self-organization model is well known for its low-dimensional topology-preserving mapping of high-dimensional patterns and stably evolving properties. The authors concluded that because of the neural network’s superior ability in dealing with nonlinear problems such as predicting HIV protease cleavage sites with 92.06% accuracy in proteins it is quite reliable and thus could be helpful in finding effective inhibitors of HIV protease.

Hatzakis, et al. (2005) studied the application of neural networks to the modelling of mortality and morbidity during loss of HIV T-Cell homeostasis. Multiple factors that may predict mortality of HIV patients include initial response to therapy, viral factors and host immune parameters. Due to the complexity of this problem, the authors developed feed-forward back-propagation neural networks to optimally evaluate outcomes of therapy and predict morbidity and mortality. The neural networks featured 1 input, 1 hidden layer and 1 output layer over a sigmoid transfer function. Adaptive learning was used but not finally adopted since constant learning factor and momentum produced more accurate predictions. The maximum number of epochs was set at 100 000. The model stopped its training process at the epoch where the difference of two consecutive R2 measurements satisfied the convergence criterion of 0.0001. The weights factors were selected accordingly, and were initialized through a random number generator. The accuracy of the neural networks was validated using Cox regression modelling (RM). The results of this research indicated that although neural network and Cox modelling were successful in predicting mortality, the neural network was superior in assessing risk in the studied population. The authors concluded that neural network based modelling can be at least as accurate as regression modelling in predicting morbidity and mortality that occurs in late stages of HIV infection following loss of T-cell homeostasis. This technique may prove useful in deciding when to initiate therapy, evaluate the response to anti-retroviral treatment and eventually in the prediction of morbidity and mortality. This study did not look into the issue of local minima.

Hatzakis and Tsoukas (2001) used neural networks for the assessment of HIV immunopathology. The study was primarily designed to assess the utility of neural networks for surrogate marker-based prediction modelling of HIV disease, evaluating the immune and virologic responses to HAART. A secondary objective of the study was to compare the performance of the NNs with multiple regression analysis-based modelling (MRM). Despite the documented ability of antiretroviral therapies to increase CD4 counts, the use of other surrogate markers in patient management is limited. The authors postulated that additional HIV-related surrogate markers might be used if effective analytical tools were available. The results demonstrated that the neural network was at least as accurate as a multi-regression model.

Hatzakis *et al.* (2005) applied neural networks for a longitudinal assessment of the electronic antiretroviral therapy to determine response to HIV treatment. The relationship between treatment outcomes with disease markers and other contributing factors is complex. The authors developed ANN models based on Jordan-Elman networks to longitudinally follow viral surrogate markers together with demographics, biochemical and laboratory data to describe the drug-virus-host interactions in over 4000 HIV adult patients. Authors concluded that these tools can be used in real-time context of prospective, longitudinal clinical trials of newer antiretroviral drugs.

Herman et al. (1999) compared neural networks with five traditional methods for predicting creatinine clearance in patients with HIV infection. The 16 input variables were age, ideal body weight, actual body weight, body surface area, and the following blood chemistries: sodium, potassium, aspartate aminotransferase, red blood cell count, platelet count, white blood cell count, glucose, serum creatinine, blood urea nitrogen, and albumin. The only output variable was creatinine clearance. The ANN paradigm was a fully connected, three-layer, back-propagation algorithm with input, hidden, and output layers. The hyperbolic tangent function was used as the neuron transfer function. The selected neural architecture was trained for 1000 epochs. The authors concluded that average percentage prediction error, bias, and precision were greatly improved with the ANN over other equations. The approach is easy to use and apply to different kinds of problems and the flexibility of neural networks makes them a promising alternative to established methods.

Betechuoh et al (2008) used neural networks in an inverse configuration for the adaptive control of HIV status of individuals. In this work, a control mechanism to understand how demographic properties affect the risk of being HIV positive is implemented. The research aims to understand whether HIV susceptibility can be controlled by modifying some of the demographic properties such as education. A feed-forward and inverse neural network comprising of 9 inputs and 1 output was constructed. A genetic algorithm was used to choose the optimal number of hidden units. The authors concluded that the proposed method is able to predict the educational level of individuals to an accuracy level of 88% if the HIV status of individuals and other demographic characteristics are known. It is thus possible to understand how the educational level of individuals can be modified to control the susceptibility of individuals to HIV contraction.

This work though similar to ours only dealt with predicting educational level of individuals with HIV status and how educational level can be used to control the process of individuals contacting HIV.

Loannidis et al. (1997) proposed the use of neural networks to model complex immunogenetic associations on the progression of HIV infection. Complex immunogenetic associations of disease involving a large number of gene products are difficult to evaluate with traditional statistical methods and thus may require complex modelling. The authors evaluated the performance of feed-forward back-propagation neural networks in predicting rapid progression to acquire immunodeficiency syndrome (AIDS) for patients with HIV infection on the basis of major histocompatibility complex variables. Network performance was compared with that of logistic regression. This research concluded that neural networks could be trained to recognize genetic patterns in conjunction with associated clinical outcomes. Their performance in modelling these complex associations in a training set was superior to logistic regression models. This was attributed to the ability of neural networks to model very complex data, when trained adequately.

Resino et al. (2011) studied the development of an artificial neural network to predict significant fibrosis in HIV/Hepatitis C (HCV) co-infected patients using clinical data derived from peripheral blood. Patients were randomly divided into an estimation group used to generate the ANN and a test group used to confirm its predictive power. The authors concluded that ANN technique is a helpful tool for guiding therapeutic decisions in the clinical practice of HIV/HCV co-infection.

Pasomsub (2010) researched the application of artificial neural networks for the phenotypic drug resistance prediction. Although phenotypic resistance testing provides more direct measurement of antiretroviral drug resistance than genotypic testing, it is costly and time consuming. However, genotypic resistance testing has the advantage of being simpler and more accessible. The study applied artificial neural network (ANN) system to predict the HIV-1 resistance phenotype from the genotype. The results indicated that by using the ANN, within associated amino acid positions known to influence drug resistance for individual antiretroviral drugs, drug resistance was accurately predicted and generalized for individual HIV-1 subtypes.

Purwanto et al. (2011) studied the application of adaptive neuro-fuzzy inference system for HIV/AIDS time-series prediction. Improving accuracy in time series prediction has always been a challenging task for researchers. Prediction of time series data in healthcare such as HIV/AIDS data has assumed importance in healthcare management. Statistical techniques such as moving average (MA), weighted moving average (WMA) and autoregressive integrated moving average (ARIMA) models have limitations in handling the non-linear relationships among data. In general, for complex healthcare data, it may be difficult to obtain high prediction accuracy rates using the statistical or AI models. In order to solve this problem the authors proposed a hybrid model such as adaptive fuzzy inference system (ANFIS) to predict HIV/AIDS data. The results of this research indicated that the proposed model was superior to statistical and AI models.

Kwak and Lee (1997) studied the application of neural networks to the classification and prediction of the health status of HIV/AIDS patients. Neural network modelling of HIV/AIDS issues involves the interaction of many diverse variables, whose relationships are often unclear and ill-defined. The study utilized the AIDS Cost and Services Utilization Survey (ACSUS), a longitudinal study of persons with HIV-related disease in which a combination of personal interviews and abstraction of medical records was used. The model developed was the three-layer back-propagation algorithm neural network. Input pattern had nine variables: race-white, race-black, race-hispanic, exproute-IDU, exproute-IDU with specific reasons, totipngt (total number of inpatient nights), totamv (total number of ambulatory visits), toterv (total number of emergency room visits), and totobs (total observation days).

Tim and Marwala (2001) used computational intelligence methods for risk assessment of HIV. The design of the study consisted of two parts namely, use of neural networks trained using supervised learning on antenatal survey data to perform binary classification and use of trained neural networks to produce inferred risk probability using Bayesian classification methods to estimate class conditional densities. An auto-associative neural network was trained on complete datasets.

Kim et al. (2010) explored an MLP-based feature subset selection for HIV-1 protease cleavage site analysis. In recent years several machine learning approaches have been applied to model the specificity of the human immunodeficiency virus type 1 (HIV-1) protease cleavage domain. The high dimensional domain dataset contains a small number of samples, which could misguide classification modelling and its interpretation. An appropriate feature selection can alleviate the problem by eliminating irrelevant and redundant features and thus improve prediction performance. In this regard the authors proposed a new feature subset selection method called FS-MLP that selects relevant features using multi-layered perceptron (MLP) learning. The method involved MLP learning with a training dataset and then feature subset selection using de-compositional approach to analyse the trained MLP. The experimental results indicated that the FS-MLP is effective in analysing multi-variate, non-linear and high dimensional datasets such as HIV-1 protease cleavage dataset. The authors concluded that the FS-MLP was a useful method for computational sequence analysis.

This work only gave a general overview of the application of Neural Network to HIV modelling for prediction, classification and function approximation. No optimization Algorithm was used to avoid local minima during learning which can make any of the models to over train. Thus the accuracy or capability of prediction and classification was not evaluated

Nelwamondo et al. (2007) developed a comparison of neural networks and expectation maximization techniques to study missing data using industrial power plant, industrial winding process and HIV prevalence data. The authors compare two approaches to the problem of missing data estimation. The first technique used Maximum Likelihood (ML) and Expectation Maximization (EM) algorithm while the second approach used a system based on auto-associative neural network and genetic algorithm (GA). The authors concluded that EM algorithm was more suitable and performed better in cases where there was little or no interdependency between the input variables, whereas the auto associative neural network and GA combination was suitable when there were inherent nonlinear relationships between some of the given variables.

This work was on comparison of algorithms and did not address the issue of local minima.

Lee and Park (2000) explored the application of neural networks to classify and predict the symptomatic status of HIV/AIDS patients. The purpose of the study was to apply an ANN to provide correct classification of AIDS versus HIV status patients. An ANN model was developed using publicly available HIV/AIDS data in the AIDS Cost and Services utilization survey (ACSUS).The authors concluded that an ANN model can facilitate planning, decision-making, and managerial control by providing hospital administration information. Like others, this work did not take of local minima problem.

Insurge Jung et al. (2007) proposed a Pattern Classification of Back-Propagation Algorithm Using Exclusive Connecting Network. The objective of this work is the design of pattern classification model based on the Back-Propagation (BP) algorithm for decision support system. Standard BP model connects each node in the layers from input to output layers. Therefore, it takes a lot of computing time when doing pattern generation or training the network.

However, this model used exclusive connection in between hidden layer nodes and output nodes. The advantage of this is less number of iteration and better performance compare with standard back-propagation model. Some cases were simulated for classification data and different setting of network factors (for example, number of hidden layers and nodes, number of classification and iteration). During the simulation, it was found that most of simulations cases were satisfied by BP based using exclusive connection network model compared to standard BP. The algorithm developed can be used to identify user's face, analyze and map data. The advantage of this model is less number of iteration and better performance compare with standard back-propagation model. The model is limited because it can converge to local minima.

Lee et al. (2000) proposed an application of neural networks to classify and predict the symptomatic status of HIV/AIDS patients. The purpose of this study was to apply an Artificial Neural Network (ANN) to provide correct classification of AIDS versus HIV status patients. An ANN model was developed using publicly available HIV/AIDS. Input and output data were collected from AIDS Cost and Services Utilization Survey (ACSUS).The proposed model demonstrates which factors will affect classification of AIDS and HIV status. It also reinforces HIV/AIDS patient prevention and care planning and strategies to meet more appropriately health-care policy and regulations. In addition it provides decision-makers and policy-makers with more accurate information to allow them to implement better health-care systems. However the classification ability of the model was not evaluated. It did not make use of any optimization algorithm.

Betechuoh et al. (2006) in their paper introduced a new method of analyzing HIV using a combination of auto encoder networks and genetic algorithms. The proposed method is tested on a set of demographic properties of individuals obtained from the South

African antenatal survey. The area under the Received Operating Characteristic (ROC) curve for the proposed auto encoder network model is 0.86 compared to 0.8 for the conventional feed forward neural network model. The auto-encoder network classifier model proposed yields an accuracy of 92%, when compared to conventional feed-forward neural networks accuracy of 84%. The auto encoder network model for HIV classification, proposed in this paper, thus outperforms the conventional feed-forward neural network models and is a much better classifier.

Betechuoh et al. (2008) compared computational intelligence methods to analyze HIV in order to investigate which network is best suited for HIV classification. The methods analyzed are auto-encoder Multi-Layer Perceptron (MLP), auto-encoder Radial Basis Functions (RBF), Support Vector Machines (SVM) and Neuro-Fuzzy Models (NFM). The auto-encoder MLP yields the highest accuracy of 92% amongst all the models studied. The autoencoder RBF model has the shortest computational time but yields one of the lowest accuracies of 82%. The SVM model yields the worst accuracy of 80%, as well as the worst computational time of 203s. The NFM yields an accuracy of 86%, which is the second highest accuracy. The NFM offers rules, which gives interpretation of the data. The area under the ROC curve for the MLP model is 0.86 compared to an area under the curve of 0.87 for the RBF model, and 0.82 for the neuro-fuzzy model. The auto-encoder MLP network model for HIV classification is thus found to outperform the auto-encoder RBF, SVM and NFM.

The Human Immunodeficiency Virus / Acquired Immunodeficiency syndrome (HIV/AIDS) is only20 years old in India. Within this short period it has emerged as one of the most serious public health problems in the country, which greatly affect the socio-economic growth. The HIV problem is very complex and ill-defined from the modeling point of view. Keeping in the view the complexities of the HIV infection and its transmission, it is difficult to make exact estimates of HIV prevalence. It is more so in the Indian context, with its typical and varied cultural characteristics, and its traditions and values with special reference to sex related risk behaviours.

Therefore (Chaturvedi, 2005) developed a Neuro-Fuzzy dynamic model to estimate HIV on order to facilitate planning of HIV / AIDS prevention and control programs for the population of Agra region. The output generated was reliable.

Otine et al. (2010) focused on dimensional modeling of HIV patient information using open source modeling tools. It aimed to take advantage of the fact that most affected regions by the HIVvirus are also heavily resource constrained (sub-Saharan Africa)but have large quantities of HIV data. Two HIVdata source systems were studied to identify appropriate dimensions and the facts were then modeled using two open source dimensional modeling tools. Use of open source reduced the software costs for dimensional modeling and in turn made data warehousing and data mining more feasible even for those in resource constrained settings but with data available.

Nicole (2006) used the demographic and medical history information obtained from annual South African antenatal surveys to estimate the risk of acquiring HIV. The estimation system consists of a classifier, a neural network trained to perform binary classification, using supervised learning with the survey data. The survey information contains discrete variables such as age, gravidity and parity, as well as the quantitative variables such as race and location, as the inputs to the neural network. HIV status is the output. A multilayer perceptron with a logistic function is trained with a cross entropy error function, providing a probabilistic interpretation of the output.

Predictive and classification performance is measured, and the sensitivity and specificity are illustrated on the Receiver Operating Characteristics. An auto-associative neural network is trained on complete datasets, and when presented with partial data, global optimization methods are used to approximate the missing entries. The effect of the imputed data on the network prediction is investigated.

The results show that all neural network architectures produce similar results, but the neural networks trained with the Bayesian technique have marginally better accuracy and larger areas under the ROC curve. Estimation of the missing data did not affect the probabilistic output of the neural network classifier for single variable estimation, but was unsuccessful for predictions with multiple variable estimations.

The works reviewed confirm the ability of various neural networks models to classify and predict more accurately than the traditional statistical methods. From literature, the researchers were also able to identify the appropriate variables required in modeling HIV/AIDS status. However the works mainly focused on classification and prediction without addressing the issue of local minima, low convergence, overtraining and overfitting. This thesis therefore seeks to bridge this gap in the following ways:

1. Genetic Algorithm is embedded into MLP-ANN to train the MLP-ANN in order to avoid low convergence of Back-propagation of MLP-ANN.

2. Genetic Algorithm is also used to avoid over-fitting and over-generalization of MLP-ANN Back-propagation based.

3. The present study was compared with existing Data mining software (WEKA) to benchmark the classification and prediction performance of the developed model (NEGEM).

4. Performance error metrics like Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Recall and precision was also implemented in the model

to measure, the rate of classification and prediction accuracy, that is, the error rate of the model.

Similar to works of Herman et al., 1999, Betechuoh et al. 2007 and Otine, 2012, the model can also be used for prediction of HIV/AIDS status based on selected input variables or can be used to generate report on the trend of the viral load or CD4 count of each patient.

CHAPTER THREE

METHODOLOGY

3.0 Introduction

This chapter presents data collection, data set design and transformation, as well as the architecture of Neuro-Genetic system.

3.1 Data Collection

The data used in this study were collected from selected tertiary, general hospital, and primary health care centers and non-governmental organizations in Nigeria. The data consists of about 20,000 records of patients who have undergone HIV/AIDS test, out of which about 16,000 tested positive and about 4,000 tested negative. Each patient went through tests from which several values were recorded. The data collection phase was carefully planned to ensure that the data collected are:

- Sufficient
- Real
- Free from mistakes

3.2 The Data Set Design

The data set consists of 14 input variables and one possible output variable HIV positive and HIV negative. The input variables represent the factors that affect the status of HIV/AIDS patients.

The input variables used are:

1. Patient id.
2. State of origin
3. Residential state
4. Living status (dead/alive)
5. Sex
6. Age
7. HIV type (HIV1/HIV2)
1. Profession
2. Mode transmission
3. Symptoms
    i. Symptom 1
    ii. Symptom 2
    iii. Symptom 4
    iv. Symptom 5
    v. Symptom 6
    vi. Symptom 7
    vii. Symptom 8
4. Eliza test
5. CD4 count
6. Viral load
7. Qualification

The output variable is HIV Status (positive/negative)

3.3 Data Transformation

The data were transformed as shown in table 3.2 in a format that is acceptable to the implementation of multi-layer perception Neural Network Data mining Algorithm. Out of the 14 variables, part of the variables was majorly in the design of the input domain table. These were used for data transformation in the prediction of HIV status. Symptoms as a variable were further grouped into sub-symptoms which are illustrated in the input domain table and other relevant variables were included. While output domain table is grouped as either positive or negative.

3.4 Data Collection Techniques

The underlisted techniques were used in collecting the data.

- **Existing Documents**

Existing document containing historical data on HIV/AIDS patient were consulted as well as their laboratory tests. This data were transformed into acceptable format used by the Neuro-Genetic system developed.

- **Interview**

Doctors and the laboratory officers were also interviewed to extract information concerning the disease.

3.5 The Data Set

The data set was divided into three sets as stated below.

1. training set – 70%
2. Verification set – 10%
3. Testing set – 20%

Table 3.1a Input Variable Table

S/N	Input Variable
1	Patient id.
2	State of Origin
3	Residential State
4	Living Status (Dead / Alive)
5	Sex
6	Age
7	HIV type (HIV1/HIV2)
8	Profession
9	Mode of Transmission
10	Symptoms
11	Eliza test
12	CD4 count
13	Viral load
14	Qualification

Table 3.1b: Inputs Domain Table

S/N	Input variables	Domain
1	Symptom 1	0
2	Symptom 2	1
3	Symptom 3	0
4	Symptom 4	1
5	Symptom 5	0
6	Symptom 6	1
7	Symptom 7	0
8	Symptom 8	1
9	Eliza test	0
10	CD4 count	1
11	Viral load	0

Table 3.2: Output’s Domain Table

S/N	Output variable	Domain
1	HIV positive / negative	1 / 0

3.5.1 Training Set

The training set was used to enable the system to observe relationships between input data and the resulting output. This allowed the system to learn and develop a relationship between the input and expected possible output.

During the training phase, the system adjusted its connection weight strength in favour of inputs that were most effective in determining a specific output. The weights were generated with random values for each run.

The quantity of examples used to determine a weight adjustment is called epoch. The genetic algorithm was used to train the Multi-layer perceptron Neural Network system.

3.6 NEGEM Architecture

NEGEM architecture consists of four main units as depicted in figure 3.1

- Data Base Management System
- User Interface
- MLP-ANN GA based
- Data Mining Engine

The DBMS is the component of the NEGEM where knowledge is stored, organized and processed. The algorithm used to develop the data mining system is MLP-ANN- GA based. The GA was used to optimize weight in order to reduce the error generated during the training period which in turn helped to improve the prediction and classification accuracy.

The Data Mining Engine does the classification and prediction while the User Interface is the presentation view the user interacts with. It serves as the controller that shows the result of the prediction and classification.

The Hierarchical Diagram depicted in figure 3.2 shows the major functions of the NEGEM while its context diagram is depicted in figure 3.3.

The context diagram shows the top level function of the NEGEM. It shows the system relationship with outside entities. Whenever the system is to be used, the user supplies data to set up the topology and the system provide the classification/prediction result back to the user. The system also interacts with the database to obtain the test data and stored generated data back into the database.

Figures 3.4 to 3.7 show how data moves from one part of the system to another.

The user initiates the system by providing fresh data.

The user sets the topology of the network by providing values for the number of layers in the network and the number of neurons in each layer. The provided values are stored in the database. This function is executed only when the network is trained.

The user provides the necessary parameters needed for training the network. These parameters include learning rate, momentum, number of époques and number pattern.

The system is trained using patterns that have been saved in the database. This function modifies necessary synaptic weights of the network.

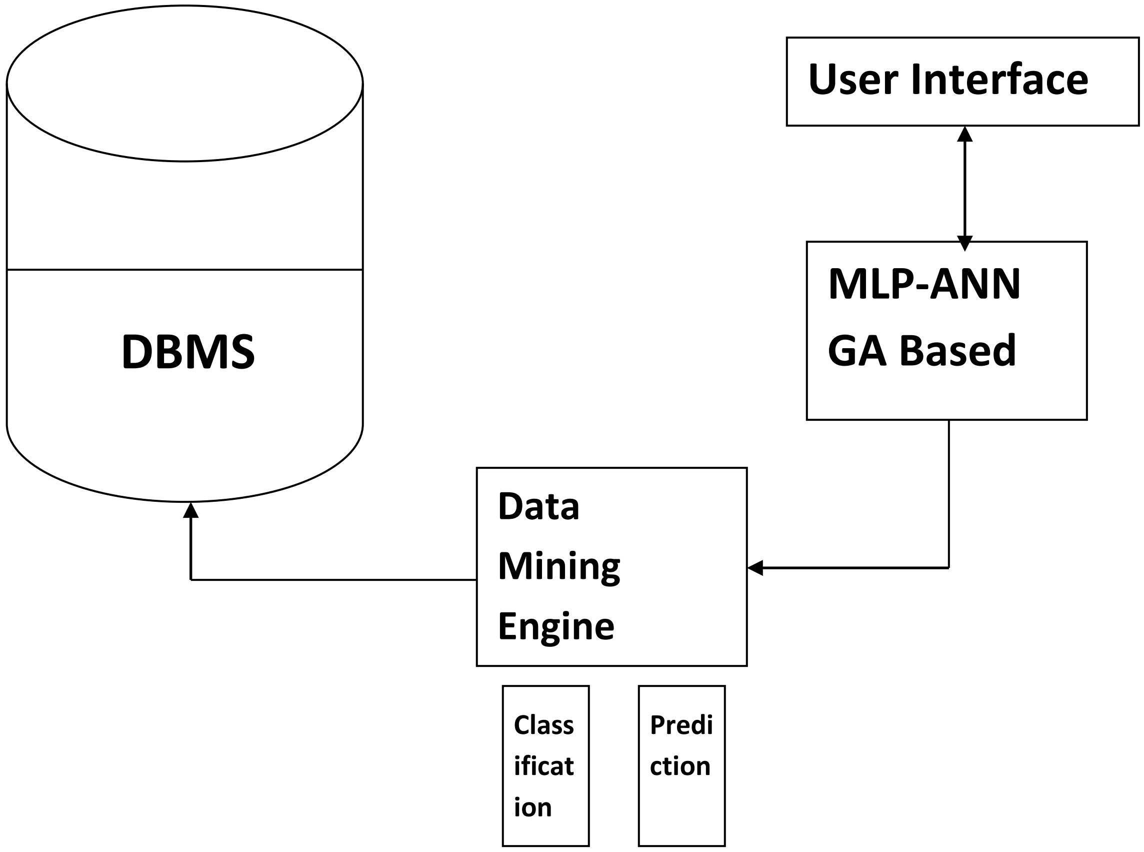


Fig. 3.1: NEGEM Architecture

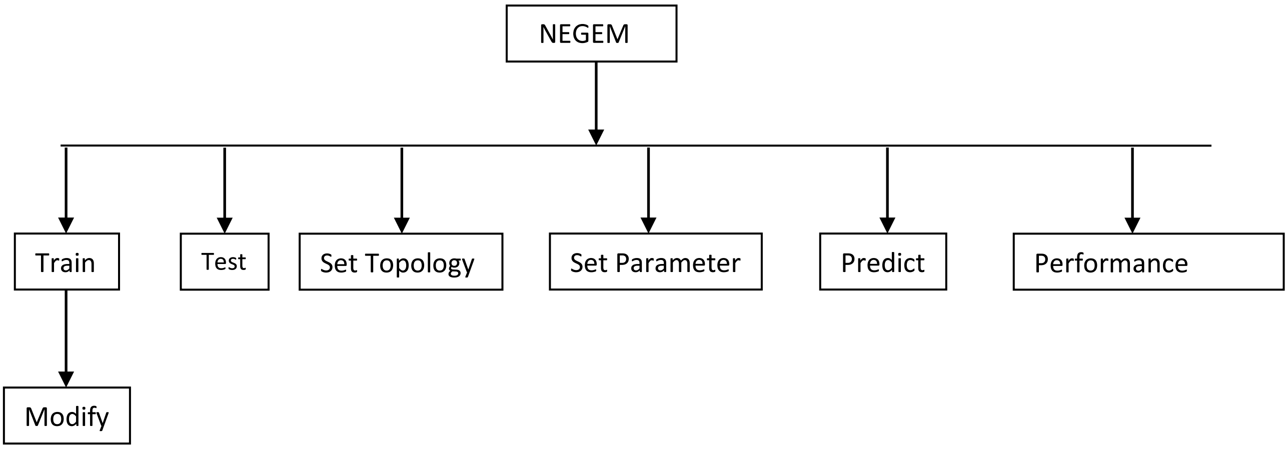


Fig. 3.2: Hierarchical Diagram

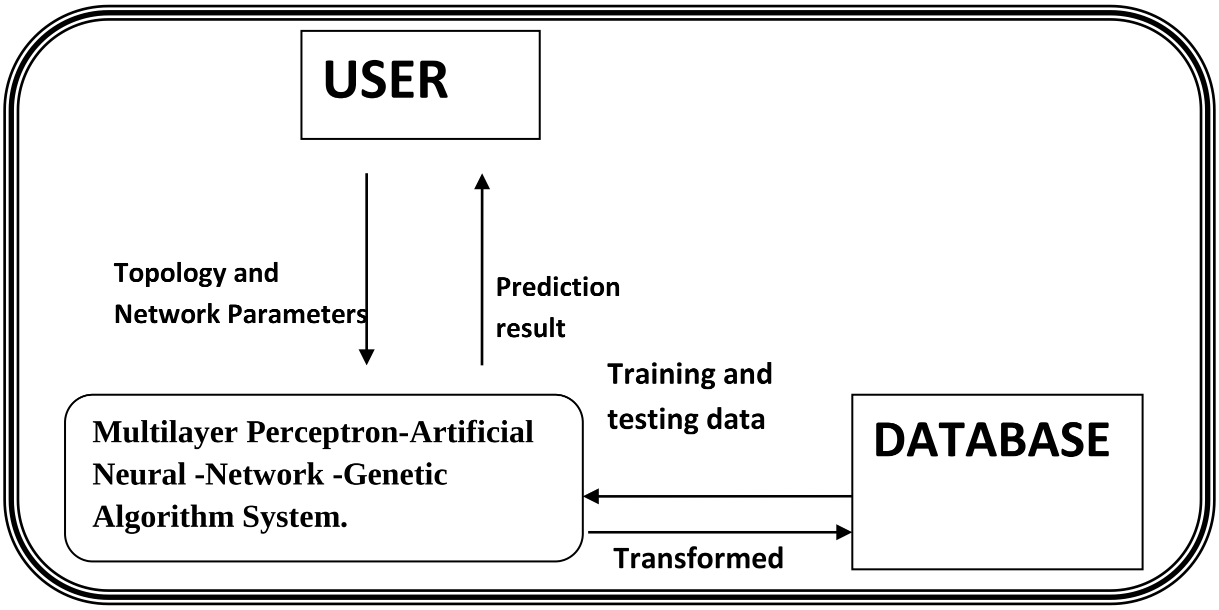


Fig. 3.3: Context Diagram

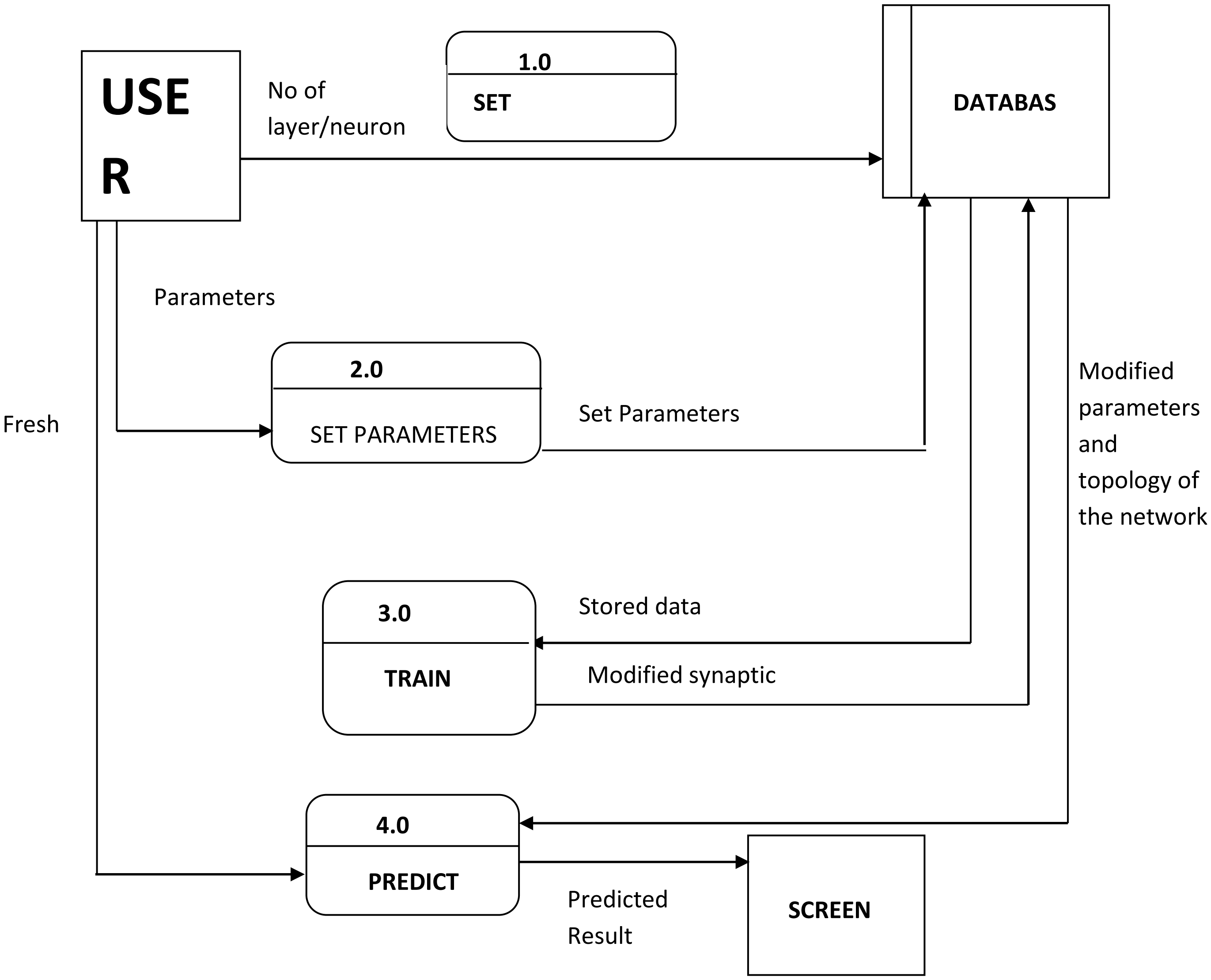


Fig 3.4 Level 0 DFD

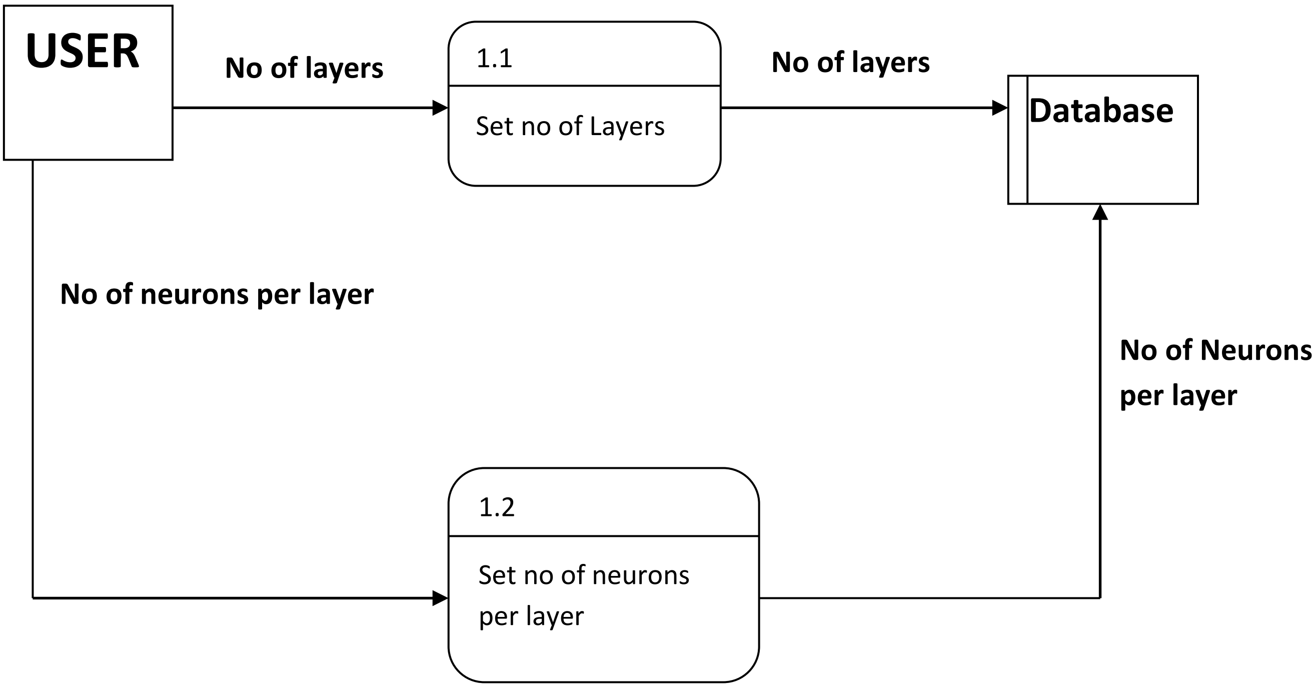


Fig. 3.5: Level 1 DFD Process 1.0

3.7 Design Tools

The designs tools used to develop the model are Entity Relationship and Use Case diagrams depicted in figures 3.8 and 3.9 respectively.

3.7.1 NEGEM Neural Network

There are several different neural network models that differ widely in function and applications. In this thesis, however feed-forward networks called multilayer perceptron neural network back propagation learning was employed. Genetic algorithm was used for training because of the rapid convergence to a local optimum of back propagation learning algorithm.

A feed-forward neural network is a function that takes an input and produces an output. The structure of the neural network is illustrated in figure 3.10.

This network consists of units or neurons or nodes (the circles) and hidden layers connections (the arrows). The number next to each connection called weight (w_i) indicates the strength of the connection. Connections with a positive weight are called excitatory, while the ones with a negative weight are called inhibitory.

The arrangement of neurons and connection is called the architecture of the network, which is also called the topology.

This is a feed-forward network, because the connections are directed in only one way, from top to bottom. There are no loops or circles.

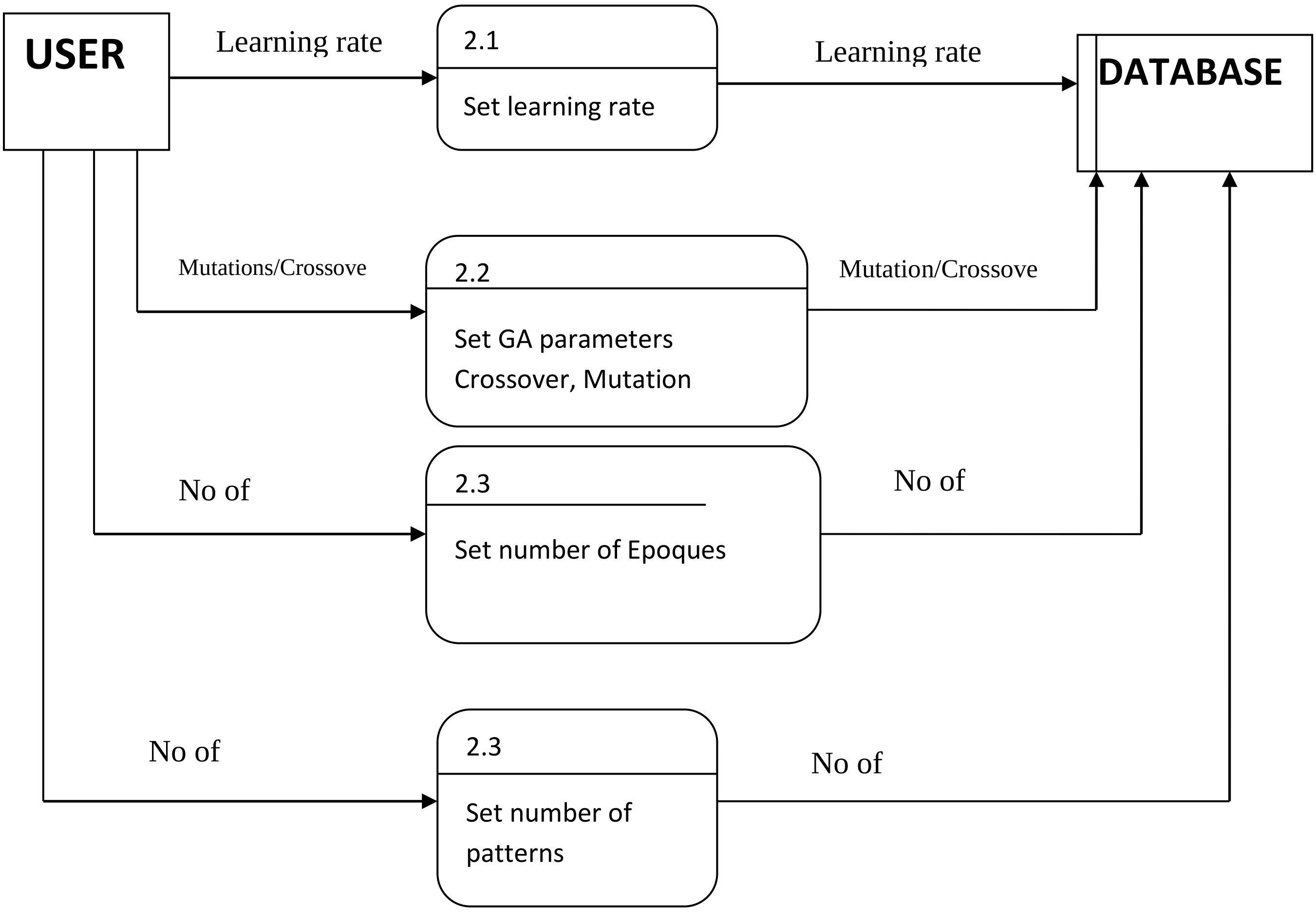


Fig. 3.6: Level 1 DFD Process 2

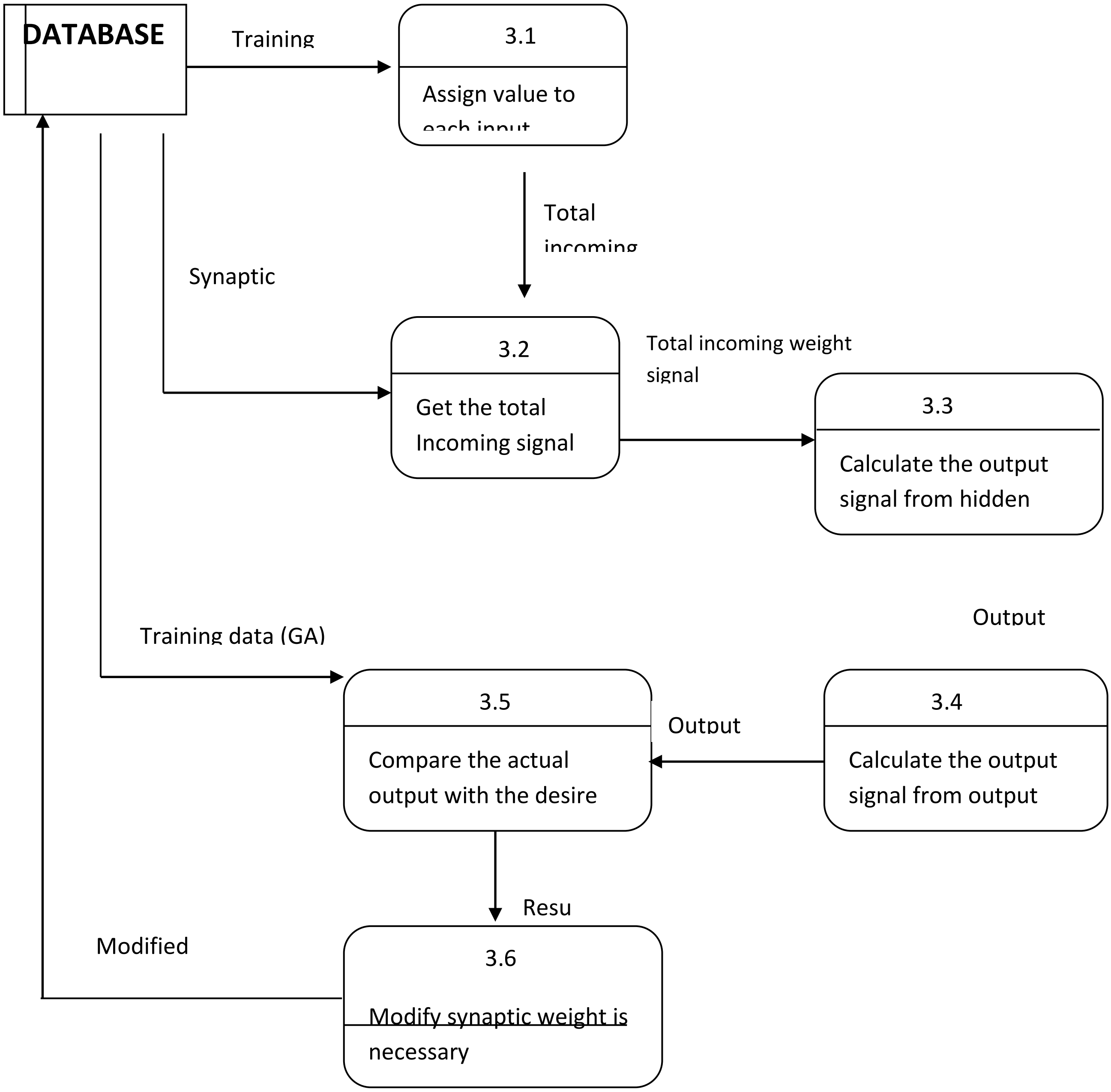

DATABASE
Training
3.1
Assign value to each input
Total incoming
Synaptic
3.2
Get the total Incoming signal
Total incoming weight signal
3.3
Calculate the output signal from hidden
Output
Training data (GA)
3.5
Compare the actual output with the desire
Output
3.4
Calculate the output signal from output
Resu
Modified
3.6
Modify synaptic weight is necessary

Fig. 3.7 Level 1 DFD process 3.0

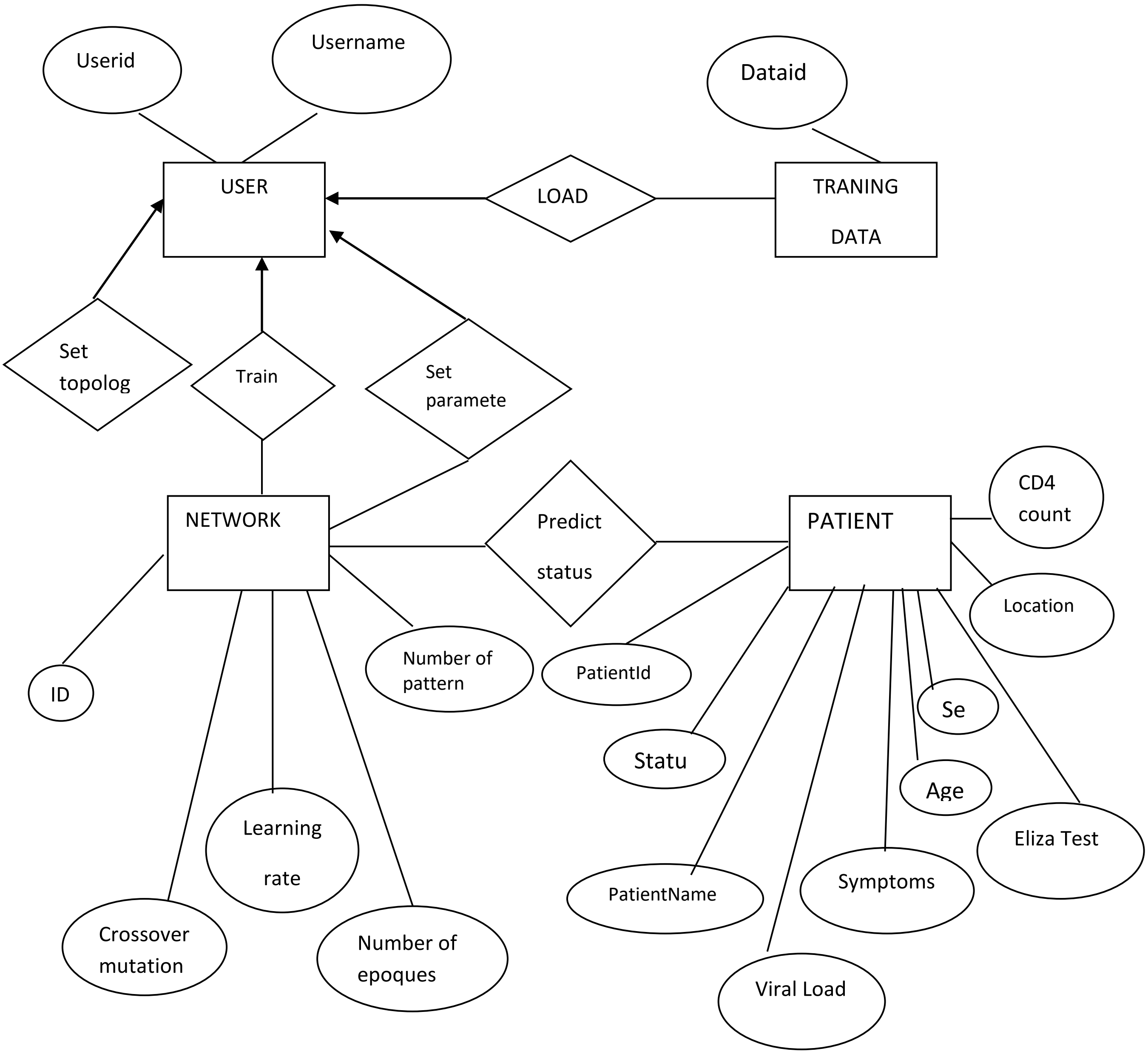


Fig. 3.8: Entity Relationship Diagram

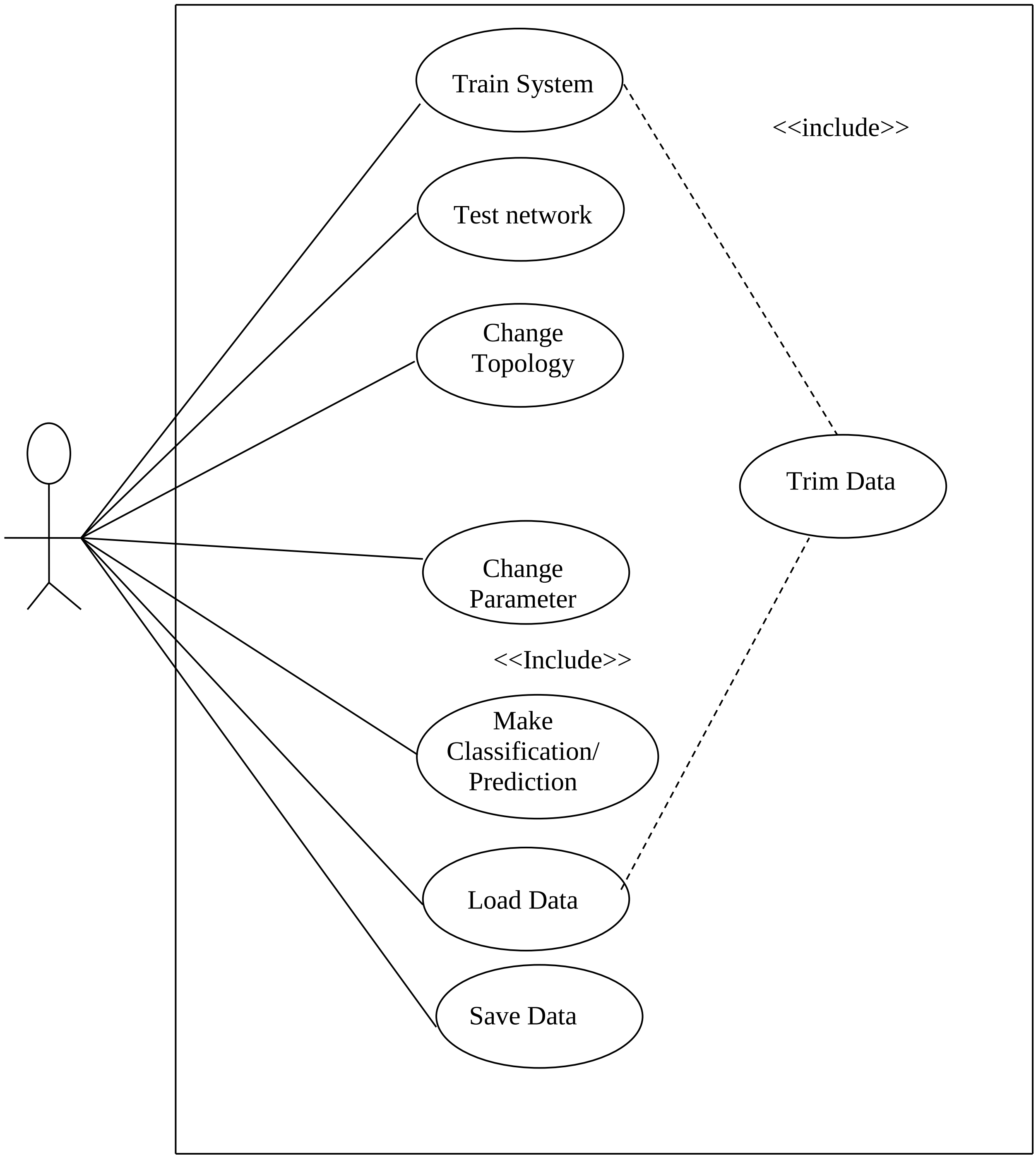


Fig. 3.9: Use Case diagram

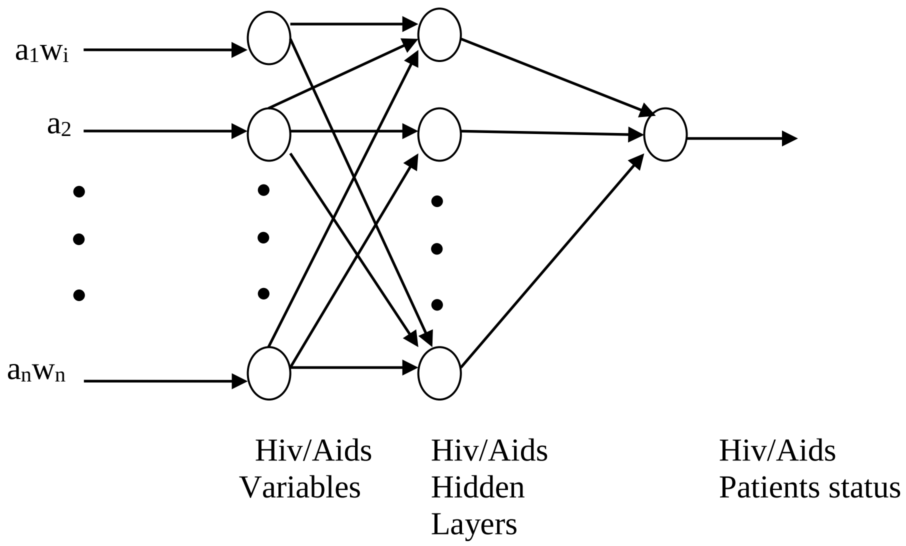


Fig 3.10: A NEGEM neural network for classification/ prediction model

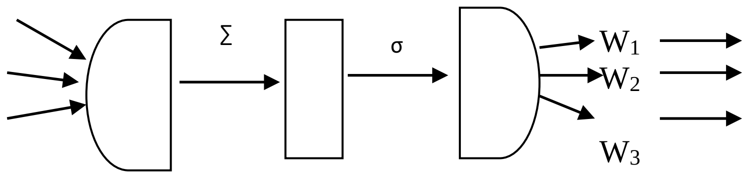


Fig. 3.11: Information Processing in NEGEM Neural Network Unit

NEGEM network topology is a layered one because the nodes of each layer are not interconnected.

Each node receives the weighted activation of other nodes through its incoming connections as depicted in figure 3.11.

These weights are added up (summation) and then the result is passed through an activation function and the outcome is the activation of the node. For each of the outgoing connections, this activation value is multiplied with the specific weight and transferred to the next node.

A few different threshold functions are used. The threshold function is non-linear otherwise the multilayer network becomes a one layer net. The most widely applied threshold function is the logistic sigmoid (Eq. 3.1a).

$$\sigma (x) = \frac{1}{1 + e^{-x}} \qquad \text{(Eq. 3.1a)}$$

Sigmoid is the classical training algorithm for feed-forward neural networks.

3.7.2 Back-Propagation Learning

At the beginning the weights of the network are randomly set or otherwise predefined. The training is performed one pattern at a time. The training of all patterns of a training set is called an epoch. The training set is a representative collection of input-output examples.

The following steps are used to develop the Multilayer Perceptron Neural Network Algorithm

Identify the input and output features (variables).

- Transform the inputs and outputs so they are in a small range (-1 to +1).
- Set up a network with an appropriate topology.
- Train the network on a representative set of training as examples.
- Use the validation set to choose the set of weights that minimizes the error.
- Evaluate the network using the test set to see how well it performs.
- Apply the model generated by the network to predict outcomes from unknown inputs.

3.7.3 Neural Networks algorithm used to developed NEGEM are highlighted below

The standard library routine that we adopted in developing NEGEM is stated below.

The Back-Propagation Learning Algorithm is based on an error correction learning rule and specifically on the minimization of the mean squared error that is a measure of the difference between the actual and the desired output. As all multilayer feed forward networks, the multilayer perceptron's are constructed of at least three layers (one input layer, one or more hidden layers and one output layer), each layer consisting of elementary processing units (artificial neurons), which incorporate a nonlinear activation function, commonly the logistic sigmoid function.

The algorithm calculates the difference between the actual response and the desired output of each neuron of the output layer of the network. Assuming that $y_j(n)$ is the actual output of the j^{th} neuron of the output layer at the iteration n and $d_j(n)$ is the corresponding desired output, the error signal $e_j(n)$ is defined as:

$$e_j(n) = d_j(n) - y_j(n) \qquad \text{(Eq. 3.1)}$$

The instantaneous value of the error energy for the neuron j is defined as e^2_j (n)/2 and correspondingly, the instantaneous total error energy E_{inst}(n) is obtained by summing the neural error energy e^2_j(n)/2 over all neurons in the output layer. Thus,

$$E_{inst}(n) = \frac{1}{2}\sum_{i} e_j^2(n) \qquad \text{(Eq. 3.2)}$$

In the above formula, j runs over all the neurons of the output layer. If N is defined to be the total number of training patterns that consist the training set applied to the neural network during the training process, then the average squared error energy E_{av} is obtained by summing E_{inst}(n) over all the training patterns and then normalizing with respect to the size N of the training set. Thus,

$$E_{av} = \frac{1}{N}\sum_{n=1}^{N} E_{inst}(n) \qquad \text{(Eq. 3.3)}$$

The instantaneous error energy E_{inst} (n), as well as the average squared error energy E_{av}, is a function of all the free parameters of the network. The objective of the learning process is to modify these free parameters of the network in such a way that E_{av} is minimized. To perform this minimization, a simple training algorithm is utilized. The training algorithm updates the synaptic weights on a pattern-by-pattern basis until one epoch, that is, one complete presentation of the entire training set is completed. The correction (modification) w_{ji}(n) that is applied on the synaptic weight w_{ji} (indicating the synaptic strength of the synapse originating from neuron i and directing to neuron j), after the application of the nth training pattern is proportional to the partial derivative $\partial E_{inst}(n)/\partial w_{ji}$. Specifically, the correction applied is given by:

$$\Delta w_{ij} = -\eta \frac{\partial E_{inst}(n)}{\partial w_{ji}(n)} \qquad \text{(Eq. 3.4)}$$

In the above formula, η is the learning-rate parameter of the back-propagation algorithm. The use of the minus sign in (Eq. 3.4) accounts for the gradient-descent in weight-space, reflecting the seek of a direction for weight change that reduces the value of $E_{inst}(n)$. (Eq. 3.4) is the mathematical expression of the Delta Rule. The exact value of the learning rate η is of great importance for the convergence of the algorithm since it modulates the changes in the synaptic weights, from iteration to iteration. The smaller the value of η, the smoother the trajectory in the weight space and the slower the convergence of the algorithm. On the other hand, if the value of η is too large, the resulting large changes in the synaptic weights may result the network to exhibit unstable (oscillatory) behaviour. Thus a modification of the Delta Rule of (Eq. 3.4) was proposed, which includes a momentum term [46]:

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) - \eta \frac{\partial \mathrm{E}_{inst}(n)}{\partial w_{ji}(n)} \qquad \text{(Eq. 3.5)}$$

In the above equation, α is a positive number called the momentum constant. (Eq. 3.5) is called the Generalized Delta Rule and it includes the Delta Rule as a special case (α =0). To derive a formula for $\partial E_{inst}(n)/ \partial w_{ji}$ we apply the chain rule of calculus and we may express this gradient as:

$$\frac{\partial E_{inst}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathrm{E}_{inst}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial \upsilon_j(n)} \frac{\partial \upsilon_j(n)}{\partial w_{ji}(n)} \qquad \text{(Eq. 3.6)}$$

We have to derive formulas for each one of the four right members of (Eq. 3.6). Differentiating both sides of (Eq. 3.2) with respect to $e_j(n)$, we obtain that:

$$\frac{\partial E_{inst}(n)}{\partial e_j(n)} = e_j(n) \qquad \text{(Eq. 3.7)}$$

Differentiating both sides of (Eq. 3.1) with respect to $y_j(n)$, we obtain:

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \qquad \text{(Eq. 3.8)}$$

Recalling that $y_j(n)$, is the output of the neuron j, which is generated by the application of the activation function φ on the total input $\upsilon_j(n)$, received by that neuron we have that:

$$y_j(n) = \phi_j(\upsilon_j(n)) \qquad \text{(Eq. 3.9)}$$

Thus, differentiating both sides of (Eq. 3.9) with respect to $\upsilon_j(n)$, we obtain:

$$\frac{\partial y_j(n)}{\partial \upsilon_j(n)} = \phi'_j(\upsilon_j(n)) \qquad \text{(Eq. 3.10)}$$

In the above equation φ' denotes the first derivative of φ. Finally, to obtain a formula for the last right member of (Eq. 3.6) we recall that υj(n), is the total input received by that neuron j given by:

$$\upsilon_j(n) = \sum_i w_{ji}(n) y_i(n) \qquad \text{(Eq. 3.11)}$$

In the above equation the index *i* runs all over the inputs applied to neuron *j*. By differentiating the previous equation with respect to w_{ji} we obtain that:

$$\frac{\partial \upsilon_j(n)}{\partial w_{ji}(n)} = y_i(n) \qquad \text{(Eq. 3.12)}$$

Replacing (Eqs. 3.7, 3.8, 3.10 and 3.12) in (Eq. 3.6) we obtain that:

$$\frac{\partial E_{inst}(n)}{\partial w_{ji}(n)} = -e_j(n)\phi'_j(\upsilon_j(n))y_i(n) \qquad \text{(Eq. 3.13)}$$

Replacing (Eq. 3.13) in (Eq. 3.5), we get:

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) - \eta \frac{\partial E_{inst}(n)}{\partial w_{ji}(n)} = \alpha \Delta w_{ji}(n-1) + \eta e_j(n)\phi'_j(\upsilon_j(n))y_i(n) \quad \text{(Eq. 3.14)}$$

In the above formula, the local gradient δj(n) can be introduced, given by:

$$\delta_j(n) = -\frac{\partial \mathrm{E}_{inst}(n)}{\partial \upsilon_j(n)} = -\frac{\partial \mathrm{E}_{inst}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial \upsilon_j(n)} = -e_j(n)\phi_j^{'}(\upsilon_j(n)) \qquad \text{(Eq. 3.15)}$$

According to the last equation, the local gradient $\delta_j(n)$ for the output neuron j is equal to the product of the corresponding signal $e_j(n)$ for that neuron and the derivative $\varphi_j'(\upsilon_j(n))$ of the associated activation function. Thus, (Eq. 3.14) can be rewritten in terms of (Eq. 3.15):

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n) \qquad \text{(Eq. 3.16)}$$

In the case that the activation function φ is the logistic sigmoid function, its derivative φ´ is given by:

$$\phi'_j\,(\upsilon(n)) = ay_i(n)[1 - y_i(n)] \qquad \text{(Eq. 3.17)}$$

In the case that neuron *j* is in the output layer, the expression for local gradient is given by:

$$\delta_j(n) = a[d_j(n) - o_j(n)]o_j(n)[1 - o_j(n)] \qquad \text{(Eq. 3.18)}$$

In the above equation $o_j(n)$ is the actual output of the neuron j and $d_j(n)$ is the corresponding desired response. In the case that neuron j is in a hidden layer, the expression for local gradient is given by:

$$\delta_j(n) = ay_j(n)\sum_k \delta_k(n)w_{kj}(n) \qquad \text{(Eq. 3.19)}$$

In the above equation, k represents the neurons that receive via synaptic inputs the output of the neuron j. Using (Eqs. 3.18 and 3.19) we may calculate the local gradient $\delta_j(n)$ without requiring explicit knowledge of the activation function.

The relations for the synaptic weight correction obtained above stand for the case that the training patterns are applied to the network in a sequential mode and the synaptic modifications adaptations given by (Eqs. 3.14 or 3.16) are applied after the application of each particular pattern. The presentation of the entire training set during the learning process is called an epoch. When the learning algorithm is applied on an epoch-by-epoch basis as described above we say that the back-propagation algorithm is applied in sequential mode or pattern-by-pattern mode. In that mode, it is a good practise to randomise the order of presentation of the training examples from one epoch to the next.

A second mode of application of the back-propagation learning algorithm is the batch mode. According to the batch mode synaptic weight modifications are performed after the presentation of all the training examples that constitute the whole total training set.

Following (Eqs. 3.1, 3.2 and 3.3) the average squared error is given by:

$$E_{av} = \frac{1}{2N}\sum_{n=1}^{N}\sum_{j} e_j^2(n) \qquad \text{(Eq. 3.20)}$$

For the case of the application of the back-propagation algorithm in batch mode, the Generalized Delta Rule for the correction of the synaptic weights is modified as follows:

$$\Delta w_{ji}(p) = \alpha\Delta w_{ji}(p-1) - \eta\frac{\partial \mathrm{E}_{av}}{\partial w_{ji}} = \alpha\Delta w_{ji}(p-1) - \frac{\eta}{N}\sum_{n=1}^{N} e_j(n)\frac{\partial e_j(n)}{\partial w_{ji}} \qquad \text{(Eq. 3.21)}$$

In the above equation is presented the synaptic modification from epoch to epoch (p denotes the epoch). According to (Eq. 3.21) in the batch mode the weight adjustment w_{ji} is made only after the entire training set has been presented to the network.

The effectiveness of each one of these two different modes of application of the back-propagation algorithm depends on the nature of the problem under consideration. The sequential (pattern-by-pattern) mode requires less storing space per neural interconnection of the network and may converges faster than the batch mode in pattern classification problems where the patterns are stored in a large database. On the other hand the synaptic modification process in the sequential mode is difficult to be implemented in parallel.

With respect to the convergence rate the **back-propagation** algorithm is relatively slow. This is related to the stochastic nature of the algorithm that provides an instantaneous estimation of the gradient of the error surface in weight space. In the case that the error surface is fairly flat along a weight dimension, the derivative of the error surface with respect to that weight is small in magnitude, therefore the synaptic adjustment applied to the weight is small and consequently many iterations (**époques**) of the algorithms may be required to produce a significant reduction in the error performance of the network.

In the alternative case, that the error surface is highly curved along a weight dimension, the resulting derivative is large in magnitude, so the synaptic modification. This may cause the algorithm to overshoot the minimum of the error surface, or become oscillatory.

To overcome the difficulty that may arise from the slow convergence of the back-propagation algorithm, four heuristics have been proposed, which can be used as guidelines for thinking about how to accelerate algorithm's convergence.

These four heuristics suggest that:

a) Every adjustable network parameter of the error function should have its own individual learning-rate parameter.

b) Every learning rate parameter should be allowed to vary from one iteration to the next.

c) when the derivative of the error function with respect to a synaptic weight has the same algebraic sign for several consecutive iterations of the algorithm, the learning-rate parameter for the particular weight should be increased and

d) When the algebraic sign of the derivative of the error function with respect to a particular synaptic weight alternates for several consecutive iterations of the algorithm, the learning-rate parameter for that weight should be increased.

The use of a different and time-varying learning-rate parameter for each synaptic weight modifies the algorithm in a fundamental way. A modified algorithm, based on these **four heuristics**, performs in a way that the synaptic adjustments are mainly based on the partial derivatives of the error surface with respect to the weights and the estimation of the curvatures of the error surface at the current operating point rather than performing a steepest-descent search.

All the proposed heuristics satisfy the locality constraint that is an inherited characteristic of the back-propagation algorithm. That limits the usefulness of the heuristics in cases that due to the nature of the error surface they do not work.

Nevertheless, modifications of the back-propagation algorithm based on the heuristics are of practical meaning. Thus, two modified learning algorithms, which satisfy the criteria introduced by these four heuristics, have been proposed: the Delta-Delta Learning Algorithm and the Delta-Bar-Delta Learning Algorithm. The **Delta-Delta** Learning Algorithm introduces an individual learning-rate parameter for every free parameter of the network. According to that algorithm the synaptic modification is given by:

$$w_{ji}(n+1) = w_{ji}(n) + \alpha \Delta w_{ji}(n-1) - \eta_{ji}(n+1)\frac{\partial E_{inst}(n)}{\partial w_{ji}(n)} \qquad \text{(Eq. 3.22)}$$

The modification of the individual learning-rate parameter η_{ji} of the synaptic weight w_{ji} is given by:

$$\Delta \eta_{ji}(n+1) = -\gamma \frac{\partial E_{inst}(n)}{\partial \eta_{ji}(n)} = \gamma \frac{\partial E_{inst}(n)}{\partial w_{ji}(n)} \frac{\partial E_{inst}(n-1)}{\partial w_{ji}(n-1)} \qquad \text{(Eq. 3.23)}$$

Thus, the learning-rate parameters are subject to modification form the applic△tion of one pattern to the next. In (Eq. 3.23) γ is a positive constant called control step-size parameter. From (Eq. 3.23) that provides the formula for the modification of the learning-rate parameters we can see that if the algebraic sign of the derivatives of the error function with respect to the synaptic weight remains the same in two consecutive iterations then the learning-rate modification parameter η is positive and the corresponding learning-rate parameter is increased. Otherwise, η is decreased.

Thus, the **Delta-Delta** Learning Algorithm satisfies the four proposed heuristics. The algorithm has the problem that in case that, the partial derivatives of the error function in two consecutive iterations have the same algebraic sign but small magnitudes, the (positive) learning-rate adjustment obtained using (Eq. 3.23) is very small. Additionally, if the partial derivatives of the error function in two consecutive iterations have opposite algebraic sign and large magnitudes then the (negative) learning-rate adjustment is large.

To overcome this problem that is related to the convergence rate and the efficiency of the learning algorithm, it has been proposed the Delta-Bar-Delta Learning Algorithm that introduces a modification on the formula that calculates the learning-rate parameter adjustment.

According to the Delta-Bar-Delta algorithm the synaptic weight adjustment is given by the same formula (Eq. 3.22) that is also used in the case of the Delta-Delta algorithm, but the learning-rate parameter adjustment, is given by the following formula:

In (Eq. 3.25) ξ is a positive constant. S_{ji} is an exponential sum of the current and the previous partial derivatives of the error function with respect to the synaptic weight. If the parameters κ and β in (Eq. 3.24) are set to zero, then the learning rate parameters take a constant value, as in the original back-propagation algorithm. From (Eq. 3.24), providing the learning-rate parameters adjustment, it is obtained that the learning-rate parameters are linearly increased and exponentially decreased. The linear increase does not allow fast increase rates, whereas the exponential decrease denotes that the learning-rate parameters remain positive and are decreased fast.

Because of the low convergence of the back propagation algorithm of the MLP-ANN during training, genetic algorithm was used to train MLP-ANN to improve classification/ prediction ability MLP-ANN.

3.8 Genetic Algorithm Model

Genetic algorithm was used to train MLP- ANN, because of the low convergence of neural network. The genetic algorithm steps used to develop the model are highlighted:

1. Create an initial population of chromosomes.
2. Evaluate the fitness or suitability of each chromosome that makes up the population.

3. Based on this fitness select the chromosomes that will mate or those that have the privilege to mate.
4. Crossover or mate the selected chromosomes and produce offspring.
5. Randomly mutate some of the genes of the chromosome.
6. Repeat steps three through five until a new population is created.
7. The algorithm ends when the best solution has not changed for a preset number of generations.

The basic GA operators are crossover, selection and mutation as explained in section 2.2.7.1 two. Fig. 3.12 illustrates the principle structure of a genetic algorithm. It starts with the random generation of an initial set of individuals, the initial population.

The individuals are evaluated and ranked. Since the number of individuals in each population is kept constant, for each new individual an old one has to be discarded, in general the one with the worst fitness value.

The two basic operators to generate new individual are mutation and crossover. During mutation, a couple of bits of the parameter string are flipped at random. Mutation may be applied to offspring produced by crossover or, as an independent operator, at random to any individual in the population.

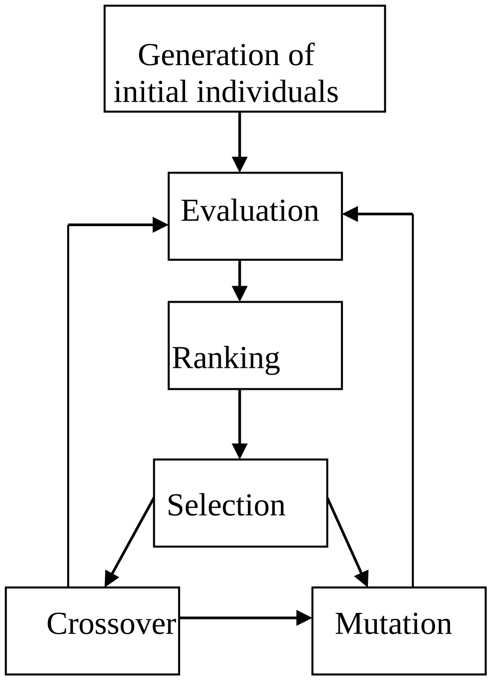


Fig. 3.12: The Principle Structure of a Genetic Algorithm

Crossover simulates the sexual generation of a child, or offspring from two parents. This is performed by taking parts of the bit-string of one of the parents and the other parts from the other parent and combining both in the child. There are three basic kinds of crossover: one-point, two-point and uniform. These are briefly explained below

One-point crossover: Both parent bit-strings are cut at the same point. So, the child can be generated by taken one part from each parent. Notice that the parts are not moved. Also, the randomly selected cutting point is independent from the actual meaning of the bits. One parameter of the bit string may be encoded in more than one bit, and its encoding cut in two pieces during crossover, thus resulting in a new value - different from either parent.

Two-point crossover: Differs from the previous version merely in the point that two random cuts are made, so three pieces have to be put together in order to produce an offspring.

These two are the two original crossover operations. The third one, uniform crossover is suggested in [Syswerda, 1989].

Parent 1: 001010011|0101001010101110

Parent 2: 010101110 |1010101101110101

Child: 001010011 1010101101110101

One-Point Crossover

Parent 1: 001010011 | 01010010 10101110

Parent 2: 010101110 | 10101011 | 01110101

Child: 001010011 10101011 01110101

Two-Point Crossover

During a generation a fixed number of crossovers and mutations are performed.

The selection of individuals for cross-over and mutation is biased towards good individuals.

In the classical fitness-based roulette-wheel, the chance of an individual to be selected is based on its relative fitness in the population.

3.8.1 Detail Description of the Genetic Algorithmic process

The standard tool box for genetic algorithm indicates that the initial random population of GA proceeds to create new members of the population (which progressively replace the old members) using genetic operators, typically mutation, crossover and inversion, modeled on their biological analogs.

Let strings be represented as

a1a2a3...al [ai= 1 or 0].

Using this notation we can describe the operators by which strings are combined to produce new strings. It is the choice of these operators which produces a search strategy that exploits to adapted sets of structural components already discovered. Holland uses three such principal operators Crossover, Mutation and Inversion.

In crossover one or two cut points are selected at random and the operation is used to create two children. A variety of control regimes are possible but we used the simplest, viz. select one of the children at random to go into the next generation. Children tend to be ’like’ their parents so that crossover can be considered as a focusing operator which exploits knowledge already gained, its effects are quite quickly apparent.

Crossing over proceeds in three steps.

a) Two structures a1...al and b1...blare selected at random from the current population.
b) A crossover point x, in the range 1 to l-1 is selected, again at random.
c) Two new structures
    i. a1a2...axbx+1bx+2...bl
    ii. b1b2...bxax+1ax+2...al

are formed.

Crossing over continually introduces new schemata for trial whilst testing extant schemata in new contexts. It can be shown that each crossing over affects a great number of schemata.

The design of the crossover operator is strongly influenced by the nature of the representation.

- Change the representation.
- Modify the crossover operator.
- Or Effect ’genetic repair’ on non-tours which may result.

In mutation an allele is altered at each site with some fixed probability; thus the number of genes altered in a mutation of a long string will be according to a Poisson distribution. Mutation disperses the population throughout the search space, so it might be considered as an information gathering or exploration operator. Search by mutation is a slow process analogous to exhaustive search.

Each structure a1a2...al in the population is operated upon as follows. Position x is modified, with probability independent of the other positions, so that the string is replaced by

a1a2...ax-1 z ax+1...al

Where z is drawn at random from the possible values. If p is the probability of mutation at a single position then the probability of h mutations in a given string is determined by a Poisson distribution with parameter p. Mutation is a ’background’ operator, assuring that the crossover operator has a full range of alleles so that the adaptive plan is not trapped on local optima.

Before inversion we can explain the effects of inversion we have to modify the string representation to be order free. This means that the order of alleles in the string should not have any effect on the genotypical information contained within the string. It turns out that without such an order-free representation, inversion would be nothing more than a brutal mutation.

We can create an order-free representation by redefining alleles as ordered pairs (ai, Pi), in which Pi is an integer,1 _ Pi _ l, and Pi denotes the position of the allele aiin the canonical (i.e. standard) representation. Thus, for example, the string

(a1,2)(a2,4)(a3,1)(a4,3),

in this new representation, maps to the canonical string

a3a1a4a2

in the original representation. For any string, (P1,P2,...,Pl) is a permutation of (1,2,...,l).

Considering the ordered pairs as units, inversion acts as follows. For some randomly selected positions x < y in the string we perform the transformation

(a1, P1)(a2, P2)... (al, Pl)

-->(a1,P1)...(ax, Px)(ay-1,Py-1)(ay-2,Py-2)...(ax+1,Px+1)(ay, Py)...(al, Pl)

Thus the effect of an inversion is to reverse the order of the ordered pair alleles between x+1 and y-1.

This has no effect whatsoever on the genotypical information (i.e. the individual produced by this new string is identical to the individual produced by the original string). Inversion by itself can accomplish nothing. Then what is the point of inversion? Before answering this question we need to describe how crossover operates with the new order-free representation. (Mutation acts just as before - changing the value of ai and having no effect on the associated Pi.)

As previously stated, the ordered pair alleles (ai, Pi) are treated as indivisible units and the cut point(s) for crossover are always chosen between ordered pairs. To perform a crossover on the new representation we rearrange the second parent so that it is homologous to the first parent, literally this means the two strings now have ’the same shape as illustrated below.

Example of an Homologous crossover

Suppose the two strings re

Parent 1: (a1,3)(a2,1)(a3,2)(a4,4)

Parent 2: (b1,2)(b2,3)(b3,4)(b4,1)

We first re-arrange the second string so that the second component in each ordered pair, that is, the ’position indicators line up with those in the first parent.

Parent 1: (a1,3)(a2,1)(a3,2)(a4,4)

Parent 2: (b2,3)(b4,1)(b1,2)(b3,4)

The second string is now homologous to the first. Now suppose the cut-point is taken at the second position. Then a possible child is: Child 1: (a1,3)(a2,1)(b1,2)(b3,4)

3.8.2 Description of the Neuro-Genetic Algorithm (NEGEM)

The Neuro-Genetic Algorithm that was used to develop the model, named NEGEM is illustrated below. In this algorithm, Back propagation algorithm is replaced with genetic algorithm coded in C-Sharp Programming Language for training the model.

Step 1: Generation of initial variables.

Step 2: Generation of MLP-ANN

Step 3: Training

Step 4: Pattern extraction

Step 5: Evaluation

Step 6: Ranking

Step 7: Selection

Step 8: Crossover

Step 9: Mutation

Step 10: Go to step 1 until a reliable result is reached and the error is within the limit specified

This procedure is illustrated in Fig. 3.12

3.9 PERFORMANCE MEASURES

The overall performances of the models were evaluated by some forecasting accuracy measures. The following performance measures were used in the study.

(a) MEAN SQUARE ERROR (MSE)

The mean square error is simply two times the average cost.

The formula is:

$$MSE = \frac{\sum_{j=0}^{P}\sum_{i=0}^{N}(d_{ij}-y_{ij})^2}{Np} \qquad \text{(Eq. 3.24)}$$

Where

P = number of output processing elements.

N = number of exemplars in the data set.

Y_{ij} = Network output for exemplar i at processing element j.

d_{ij} = desired output for exemplar i at processing element j.

(b) ROOT MEAN SQUARE ERROR (RMSE)

The root mean square error (RMSE) is used to measure the differences between values predicted by a model (or an estimation) and the values actually observed. RMSE is a good measure of accuracy. The individual differences are called residuals, and RMSE serves to aggregate them into a single measure of predictive power. RMSE is also the square root of mean square error.

RMSE $(\hat{\theta})$ = $\sqrt{MSE\hat{\theta}}$

$$= \sqrt{E(\bar{\theta}-\theta)^2} \qquad \text{(Eq. 3.25)}$$

RMSE (θ_1, θ_2)= $\sqrt{MSE(\theta_1\theta_2)}$ (Eq. 3.26)

$$= \sqrt{E(\bar{\theta}-\theta)^2}$$

$$= \sqrt{\frac{\sum_{i=1}^{n}=(x_{1i}-x_{2i})^2}{n}}$$

In computational neuro science, the RMSE is used to assess how well a system leans a given model.

(c) MEAN ABSOLUTE ERROR

The mean absolute error (MAE) is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. The mean absolute error is given by

$$MAE = \sum |XE - XP|/N \qquad \text{(Eq. 3.27)}$$

Where

XE= Expected output

XP = prediction or Network output

N = Total number of Records e.g. 12,669

The mean absolute error is a common measure of forecast error in time series analysis, where the terms "mean absolute deviation" is sometimes used in confusion with the more standard definition of mean absolute deviation. Precision is the fraction of retrieved instances that are relevant. This is given as:

$$Precision = TP/(FP + TP) \qquad \text{(Eq. 3.28)}$$

While

Recall is the fraction of relevant instances that are retrieved.

$$Recall = TP/ (FN + TP) \qquad \text{(Eq. 3.29)}$$

PERCENTAGE ACCURACY

Percentage correctly predicted

$$= (Tp + T_N)/ (Fp + Tp + F_N + T_N) \times 100 \qquad \text{(Eq. 3.30)}$$

Percentage correctness PC is:

$$PC = \frac{\text{Number of correct samples}}{\text{Total number of predicted samples}} \times 100 \qquad \text{(Eq. 3.31)}$$

NEGEM was developed by embedding GA into multilayer perceptron algorithm in order to achieve rapid convergence to a global optimum during training of the network so as to increase the classification and prediction ability of the neural network architecture used.

CHAPTER FOUR

IMPLEMENTATION, RESULT AND DISCUSSION

4.1 Implementation Techniques

Hardware & Software Requirement

The following software and hardware were used:

- NEGEM data model
- Microsoft SQL server 2008
- Microsoft Excel
- Microsoft Visual Studio 2010
- Microsoft Windows 7

NEGEM was developed and custom made for HIV classification/prediction. It serves as the front-end of the application in which the user interacts with. It has the capacity to handle millions of HIV related data. NEGEM was built to predict HIV status of the patient using back propagation approach optimized with Genetic Algorithm.

Microsoft SQL server 2008, a relational database management system as a back-end of the application which has the capacity to handle very large data was employed as its data store. Microsoft SQL server 2008 is very efficient and flexible and allows for proper reporting and data analysis. Storage capacity is one of its strength; it can store data of unlimited sizes and also support unlimited concurrent database connections.

Building query with Microsoft SQL server is very easy. So querying the database management system from NEGEM is a function of querying the database with normal Structure Query Language and a little bit of Transact SQL.

Some of the data from the Microsoft SQL server 2008 were exported to Microsoft Excel for further analysis and reporting.

Microsoft Visual Studio 2010 was used for the development environment because it has a very rich Graphical User Interface for building complex applications. Also it has a very strong code intelligence which makes programming easy and more effective. Visual Studio 2010 allows for multiple languages to run on a single platform via .NET framework.

Implementation technique

The neuro-generic knowledge prediction system is implemented using a custom built classification/prediction model called NEGEM.

4.2 Implementation Steps

- Create two connectors: Input-Layer and Hidden-Layer [Sigmoid Layer], Hidden-Layer Input-Layer[Sigmoid-Layer] respectively.
- Initialize the GA with crossover of 50%, mutation rate of 1%, population size of 12000, epoch length of 2, 000 iterations, and the number of weights which is addition of all SynapseCount * 2.
- Make the one that match the exact number of weights to be used in our neural network. Since our network consists of 3 layers (input, hidden, and output) with 14 neurons at the input layer, 1 neuron in the hidden layer, and 1 neuron in the

output layer, a fully connected neural network would require addition of all Synapse Count connections.

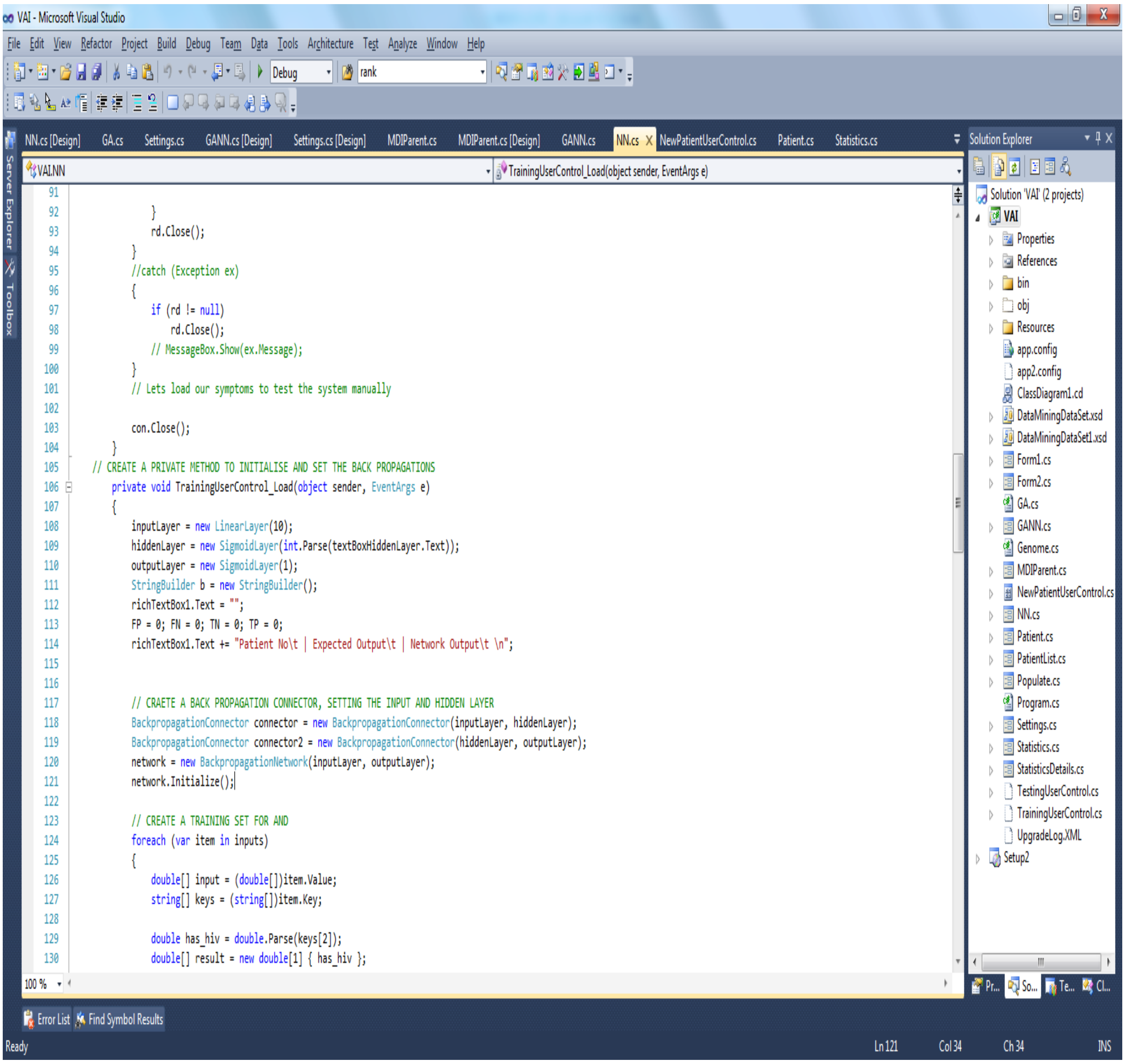
- Double the bias values.
- After the genetic algorithm finishes its evolution epochs, we pick the best result from the final population and assign its weights to a neural network. This gives us the best brain for the AND function. We then run the same test code to try the brain out.

This procedure is depicted in figure 4.1 through figure 4.10

4.3 Model Results and Discussion

In the development of Neuro-Genetic Algorithm Model (NEGEM), the number of hidden layer was varied between the input and output layer in order to select the model that fits multilayer perceptron networks chosen for the study. The multilayer perceptron was combined with Genetic Algorithm using delta learning Algorithms to implement multilayer neural network. Genetic Algorithm was used to train the neural network. Genetic Algorithms was used to optimize weights during training, to avoid overfitting and low convergence speed and over generalization of the network.

Different multilayer networks were simulated with different hidden layers, mutation at 1%, 50% crossover, 2000 epochs in Genetic Algorithm were used to avoid premature convergence to a local optimum. The network output (predicted output) and expected output were determined to check the predictability and classification strength of the model. The learning performance were then measured based on the Mean Square Error (MSE), The Root Mean square Error, Percentage Accuracy of the predicted and classification value of the developed model. The results of the different models are shown in Tables 4.1 to 4.6



```
                }
                rd.Close();
            }
            //catch (Exception ex)
            {
                if (rd != null)
                    rd.Close();
                // MessageBox.Show(ex.Message);
            }
            // Lets load our symptoms to test the system manually

            con.Close();
        }
    // CREATE A PRIVATE METHOD TO INITIALISE AND SET THE BACK PROPAGATIONS
        private void TrainingUserControl_Load(object sender, EventArgs e)
        {
            inputLayer = new LinearLayer(10);
            hiddenLayer = new SigmoidLayer(int.Parse(textBoxHiddenLayer.Text));
            outputLayer = new SigmoidLayer(1);
            StringBuilder b = new StringBuilder();
            richTextBox1.Text = "";
            FP = 0; FN = 0; TN = 0; TP = 0;
            richTextBox1.Text += "Patient No\t | Expected Output\t | Network Output\t \n";


            // CRAETE A BACK PROPAGATION CONNECTOR, SETTING THE INPUT AND HIDDEN LAYER
            BackpropagationConnector connector = new BackpropagationConnector(inputLayer, hiddenLayer);
            BackpropagationConnector connector2 = new BackpropagationConnector(hiddenLayer, outputLayer);
            network = new BackpropagationNetwork(inputLayer, outputLayer);
            network.Initialize();

            // CREATE A TRAINING SET FOR AND
            foreach (var item in inputs)
            {
                double[] input = (double[])item.Value;
                string[] keys = (string[])item.Key;

                double has_hiv = double.Parse(keys[2]);
                double[] result = new double[1] { has_hiv };
```

Fig. 4.1: Visual Studio Implementation screen shot

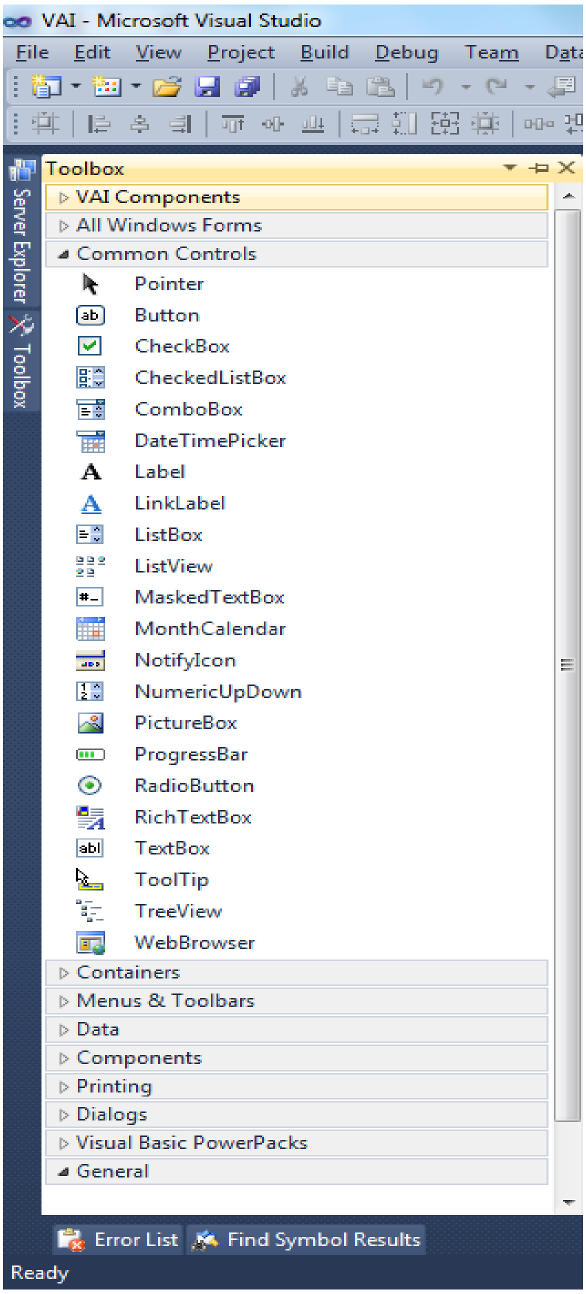
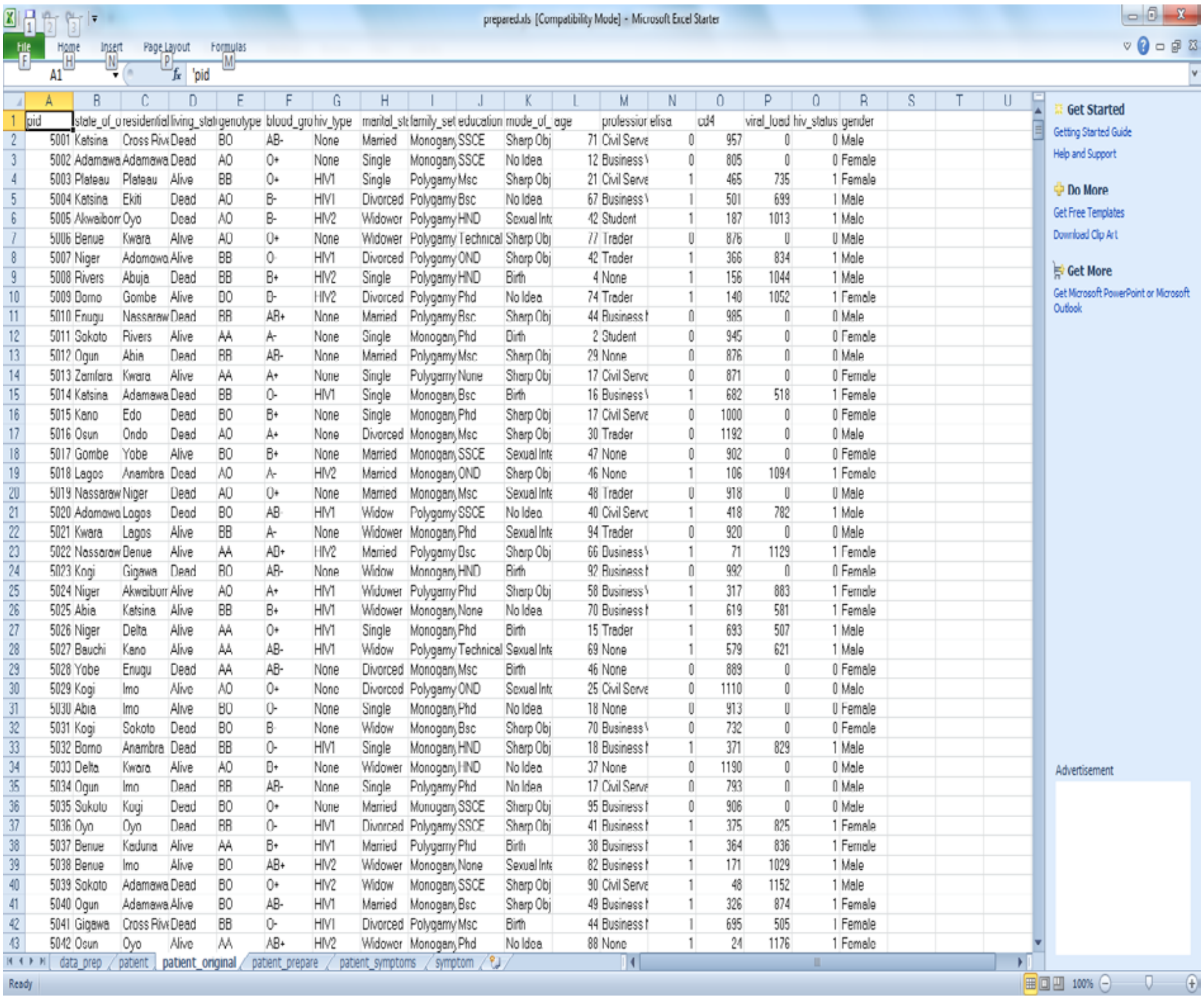
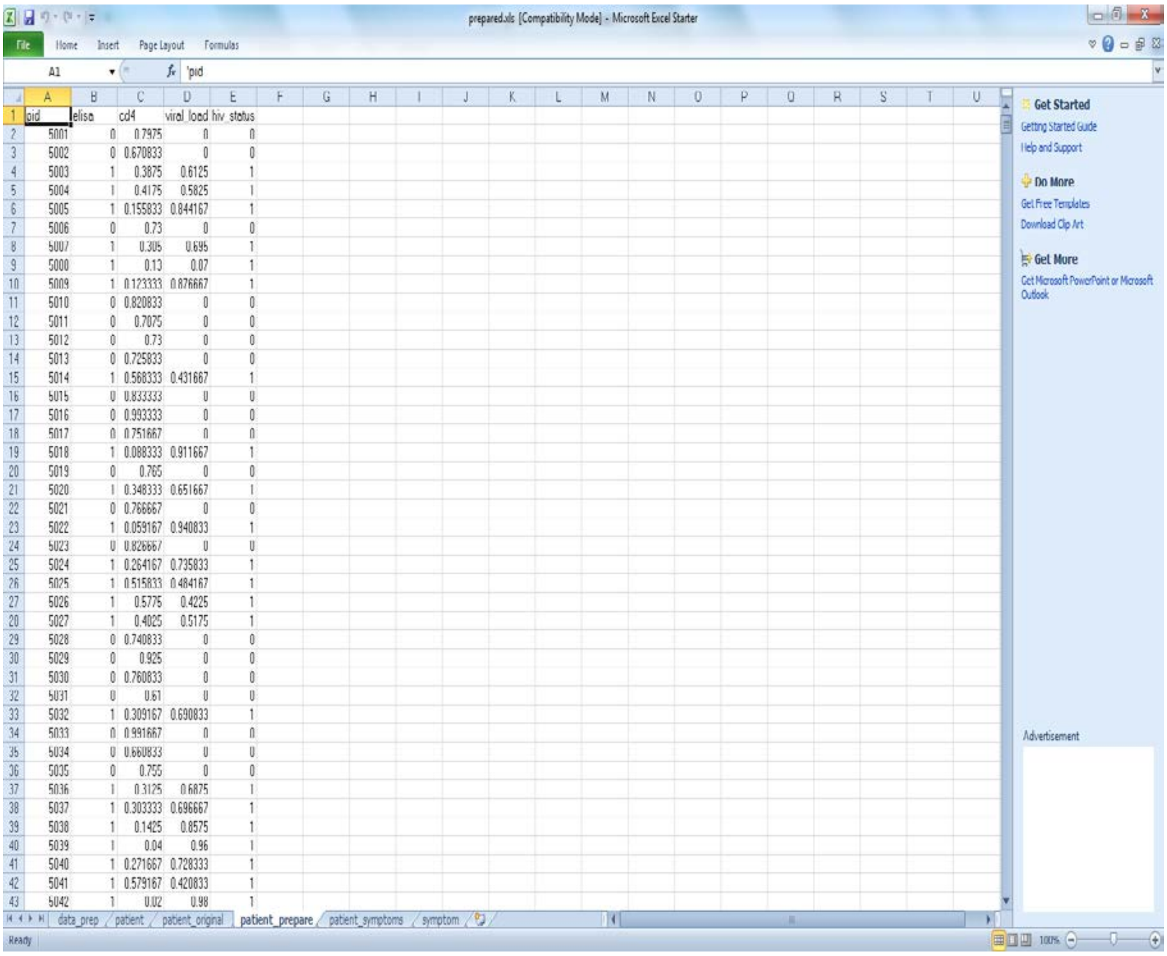


Fig. 4.2: Visual Studio Tool Box



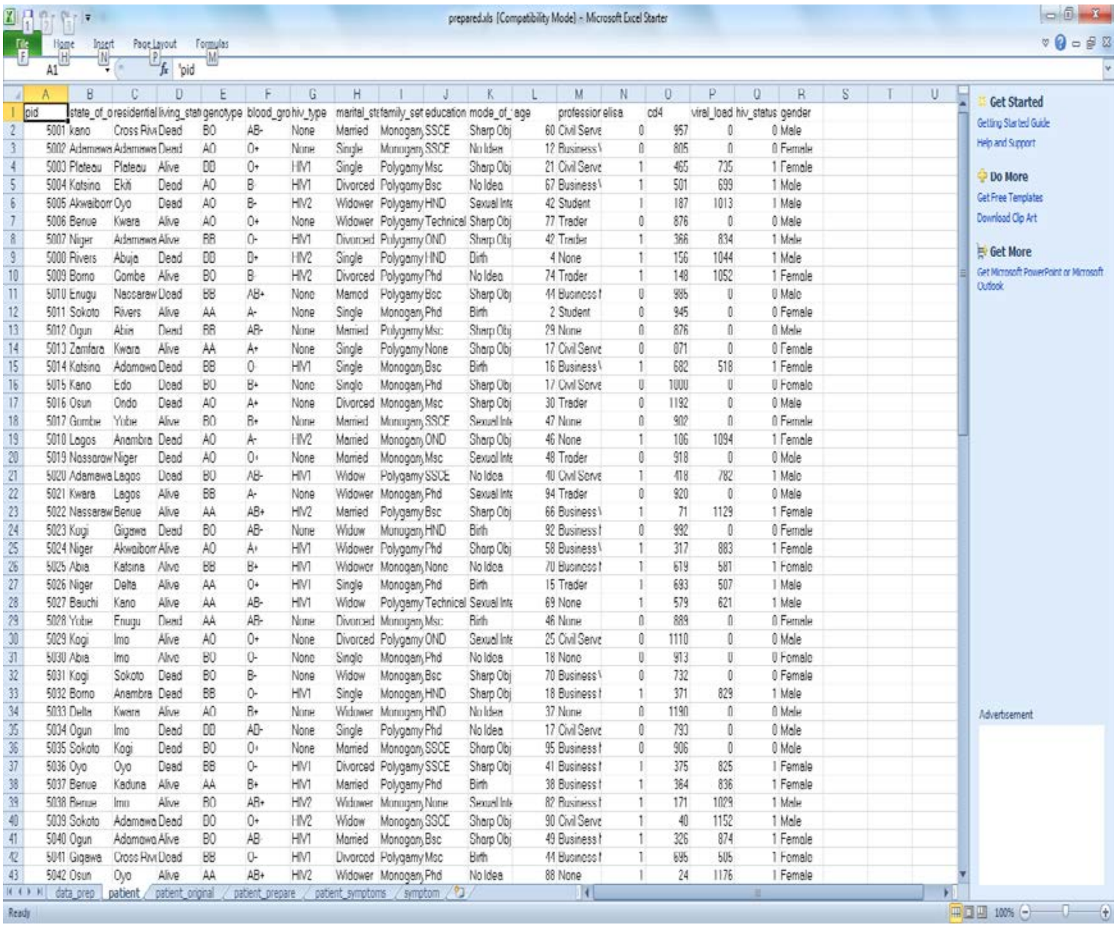
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	pid	state_of_o	residential	living_stat	genotype	blood_gro	hiv_type	marital_st	family_set	education	mode_of_	age	professior	elisa	cd4	viral_load	hiv_status	gender
2	5001	Katsina	Cross Riv	Dead	BO	AB-	None	Married	Monogam	SSCE	Sharp Obj	71	Civil Serve	0	957	0	0	Male
3	5002	Adamawa	Adamawa	Dead	AO	O+	None	Single	Monogam	SSCE	No Idea	12	Business	0	805	0	0	Female
4	5003	Plateau	Plateau	Alive	BB	O+	HIV1	Single	Polygamy	Msc	Sharp Obj	21	Civil Serve	1	465	735	1	Female
5	5004	Katsina	Ekiti	Dead	AO	B-	HIV1	Divorced	Polygamy	Bsc	No Idea	67	Business	1	501	699	1	Male
6	5005	Akwaibom	Oyo	Dead	AO	B-	HIV2	Widower	Polygamy	HND	Sexual Int	42	Student	1	187	1013	1	Male
7	5006	Benue	Kwara	Alive	AO	O+	None	Widower	Polygamy	Technical	Sharp Obj	77	Trader	0	876	0	0	Male
8	5007	Niger	Adamawa	Alive	BB	O-	HIV1	Divorced	Polygamy	OND	Sharp Obj	42	Trader	1	366	834	1	Male
9	5008	Rivers	Abuja	Dead	BB	B+	HIV2	Single	Polygamy	HND	Birth	4	None	1	156	1044	1	Male
10	5009	Borno	Gombe	Alive	BO	B-	HIV2	Divorced	Polygamy	Phd	No Idea	74	Trader	1	140	1052	1	Female
11	5010	Enugu	Nassaraw	Dead	BB	AB+	None	Married	Polygamy	Bsc	Sharp Obj	44	Business	0	985	0	0	Male
12	5011	Sokoto	Rivers	Alive	AA	A-	None	Single	Monogam	Phd	Birth	2	Student	0	945	0	0	Female
13	5012	Ogun	Abia	Dead	BB	AB-	None	Married	Polygamy	Msc	Sharp Obj	29	None	0	876	0	0	Male
14	5013	Zamfara	Kwara	Alive	AA	A+	None	Single	Polygamy	None	Sharp Obj	17	Civil Serve	0	871	0	0	Female
15	5014	Katsina	Adamawa	Dead	BB	O-	HIV1	Single	Monogam	Bsc	Birth	16	Business	1	682	518	1	Female
16	5015	Kano	Edo	Dead	BO	B+	None	Single	Monogam	Phd	Sharp Obj	17	Civil Serve	0	1000	0	0	Female
17	5016	Osun	Ondo	Dead	AO	A+	None	Divorced	Monogam	Msc	Sharp Obj	30	Trader	0	1192	0	0	Male
18	5017	Gombe	Yobe	Alive	BO	B+	None	Married	Monogam	SSCE	Sexual Inte	47	None	0	902	0	0	Female
19	5018	Lagos	Anambra	Dead	AO	A-	HIV2	Married	Monogam	OND	Sharp Obj	46	None	1	106	1094	1	Female
20	5019	Nassaraw	Niger	Dead	AO	O+	None	Married	Monogam	Msc	Sexual Inte	48	Trader	0	918	0	0	Male
21	5020	Adamawa	Lagos	Dead	BO	AB-	HIV1	Widow	Polygamy	SSCE	No Idea	40	Civil Serve	1	418	782	1	Male
22	5021	Kwara	Lagos	Alive	BB	A-	None	Widower	Monogam	Phd	Sexual Inte	94	Trader	0	920	0	0	Male
23	5022	Nassaraw	Benue	Alive	AA	AB+	HIV2	Married	Polygamy	Bsc	Sharp Obj	66	Business	1	71	1129	1	Female
24	5023	Kogi	Gigawa	Dead	BO	AB-	None	Widow	Monogam	HND	Birth	92	Business	0	992	0	0	Female
25	5024	Niger	Akwaibom	Alive	AO	A+	HIV1	Widower	Polygamy	Phd	Sharp Obj	58	Business	1	317	883	1	Female
26	5025	Abia	Katsina	Alive	BB	B+	HIV1	Widower	Monogam	None	No Idea	70	Business	1	619	581	1	Female
27	5026	Niger	Delta	Alive	AA	O+	HIV1	Single	Monogam	Phd	Birth	15	Trader	1	693	507	1	Male
28	5027	Bauchi	Kano	Alive	AA	AB-	HIV1	Widow	Polygamy	Technical	Sexual Inte	69	None	1	579	621	1	Male
29	5028	Yobe	Enugu	Dead	AA	AB-	None	Divorced	Monogam	Msc	Birth	46	None	0	889	0	0	Female
30	5029	Kogi	Imo	Alive	AO	O+	None	Divorced	Polygamy	OND	Sexual Int	25	Civil Serve	0	1110	0	0	Male
31	5030	Abia	Imo	Alive	BO	O-	None	Single	Monogam	Phd	No Idea	18	None	0	913	0	0	Female
32	5031	Kogi	Sokoto	Dead	BO	B-	None	Widow	Monogam	Bsc	Sharp Obj	70	Business	0	732	0	0	Female
33	5032	Borno	Anambra	Dead	BB	O-	HIV1	Single	Monogam	HND	Sharp Obj	18	Business	1	371	829	1	Male
34	5033	Delta	Kwara	Alive	AO	B+	None	Widower	Monogam	HND	No Idea	37	None	0	1190	0	0	Male
35	5034	Ogun	Imo	Dead	BB	AB-	None	Single	Polygamy	Phd	No Idea	17	Civil Serve	0	793	0	0	Male
36	5035	Sokoto	Kogi	Dead	BO	O+	None	Married	Monogam	SSCE	Sharp Obj	95	Business	0	906	0	0	Male
37	5036	Oyo	Oyo	Dead	BB	O-	HIV1	Divorced	Polygamy	SSCE	Sharp Obj	41	Business	1	375	825	1	Female
38	5037	Benue	Kaduna	Alive	AA	B+	HIV1	Married	Polygamy	Phd	Birth	38	Business	1	364	836	1	Female
39	5038	Benue	Imo	Alive	BO	AB+	HIV2	Widower	Monogam	None	Sexual Inte	82	Business	1	171	1029	1	Male
40	5039	Sokoto	Adamawa	Dead	BO	O+	HIV2	Widow	Monogam	SSCE	Sharp Obj	90	Civil Serve	1	48	1152	1	Male
41	5040	Ogun	Adamawa	Alive	BO	AB-	HIV1	Married	Monogam	Bsc	Sharp Obj	49	Business	1	326	874	1	Female
42	5041	Gigawa	Cross Riv	Dead	BB	O-	HIV1	Divorced	Polygamy	Msc	Birth	44	Business	1	695	505	1	Female
43	5042	Osun	Oyo	Alive	AA	AB+	HIV2	Widower	Monogam	Phd	No Idea	88	None	1	24	1176	1	Female

Fig. 4.3: Raw data in excel/ Microsoft Excel 2007 Application Window



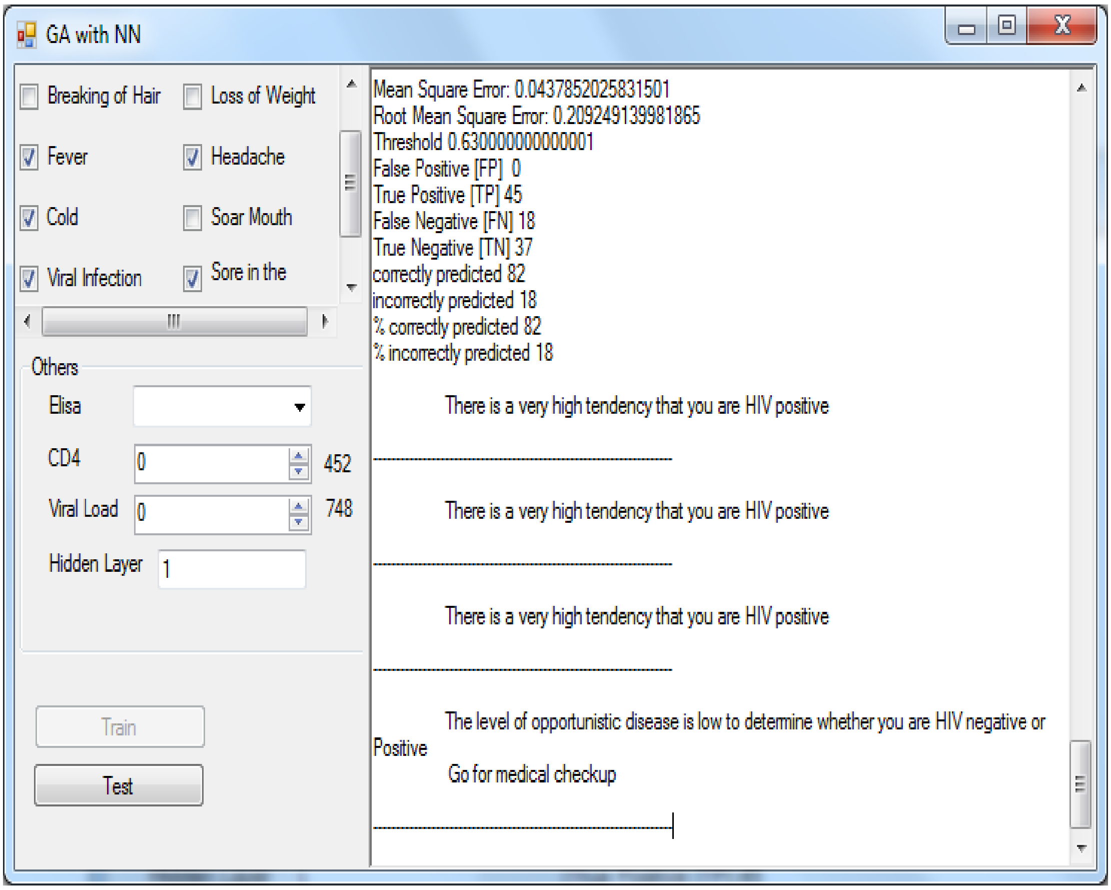
	A	B	C	D	E
1	pid	elisa	cd4	viral_load	hiv_status
2	5001	0	0.7975	0	0
3	5002	0	0.670833	0	0
4	5003	1	0.3875	0.6125	1
5	5004	1	0.4175	0.5825	1
6	5005	1	0.155833	0.844167	1
7	5006	0	0.73	0	0
8	5007	1	0.305	0.695	1
9	5008	1	0.13	0.87	1
10	5009	1	0.123333	0.876667	1
11	5010	0	0.820833	0	0
12	5011	0	0.7075	0	0
13	5012	0	0.73	0	0
14	5013	0	0.725833	0	0
15	5014	1	0.568333	0.431667	1
16	5015	0	0.833333	0	0
17	5016	0	0.993333	0	0
18	5017	0	0.751667	0	0
19	5018	1	0.088333	0.911667	1
20	5019	0	0.765	0	0
21	5020	1	0.348333	0.651667	1
22	5021	0	0.766667	0	0
23	5022	1	0.059167	0.940833	1
24	5023	0	0.826667	0	0
25	5024	1	0.264167	0.735833	1
26	5025	1	0.515833	0.484167	1
27	5026	1	0.5775	0.4225	1
28	5027	1	0.4825	0.5175	1
29	5028	0	0.740833	0	0
30	5029	0	0.925	0	0
31	5030	0	0.760833	0	0
32	5031	0	0.61	0	0
33	5032	1	0.309167	0.690833	1
34	5033	0	0.991667	0	0
35	5034	0	0.660833	0	0
36	5035	0	0.755	0	0
37	5036	1	0.3125	0.6875	1
38	5037	1	0.303333	0.696667	1
39	5038	1	0.1425	0.8575	1
40	5039	1	0.04	0.96	1
41	5040	1	0.271667	0.728333	1
42	5041	1	0.579167	0.420833	1
43	5042	1	0.02	0.98	1

Fig. 4.4: Prepared data in excel ready for training the model /Sample dataset ready for training



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	pid	state_of_o	residential	living_stat	genotype	blood_gro	hiv_type	marital_st	family_set	education	mode_of_	age	professior	elisa	cd4	viral_load	hiv_status	gender
2	5001	kano	Cross Rive	Dead	BO	AB-	None	Married	Monogam	SSCE	Sharp Obj	60	Civil Serve	0	957	0	0	Male
3	5002	Adamawa	Adamawa	Dead	AO	O+	None	Single	Monogam	SSCE	No Idea	12	Business\	0	805	0	0	Female
4	5003	Plateau	Plateau	Alive	DD	O+	HIV1	Single	Polygamy	Msc	Sharp Obj	21	Civil Serve	1	465	735	1	Female
5	5004	Katsina	Ekiti	Dead	AO	B	HIV1	Divorced	Polygamy	Bsc	No Idea	67	Business\	1	501	699	1	Male
6	5005	Akwaibom	Oyo	Dead	AO	B-	HIV2	Widower	Polygamy	HND	Sexual Inte	42	Student	1	187	1013	1	Male
7	5006	Benue	Kwara	Alive	AO	O+	None	Widower	Polygamy	Technical	Sharp Obj	77	Trader	0	876	0	0	Male
8	5007	Niger	Adamawa	Alive	BB	O-	HIV1	Divorced	Polygamy	OND	Sharp Obj	42	Trader	1	366	834	1	Male
9	5008	Rivers	Abuja	Dead	DD	D+	HIV2	Single	Polygamy	HND	Birth	4	None	1	156	1044	1	Male
10	5009	Borno	Gombe	Alive	BO	B	HIV2	Divorced	Polygamy	Phd	No Idea	74	Trader	1	148	1052	1	Female
11	5010	Enugu	Nassaraw	Dead	BB	AB+	None	Married	Polygamy	Bsc	Sharp Obj	44	Business f	0	985	0	0	Male
12	5011	Sokoto	Rivers	Alive	AA	A-	None	Single	Monogam	Phd	Birth	2	Student	0	945	0	0	Female
13	5012	Ogun	Abia	Dead	BB	AB-	None	Married	Polygamy	Msc	Sharp Obj	29	None	0	876	0	0	Male
14	5013	Zamfara	Kwara	Alive	AA	A+	None	Single	Polygamy	None	Sharp Obj	17	Civil Serve	0	871	0	0	Female
15	5014	Katsina	Adamawa	Dead	BB	O	HIV1	Single	Monogam	Bsc	Birth	16	Business\	1	682	518	1	Female
16	5015	Kano	Edo	Dead	BO	B+	None	Single	Monogam	Phd	Sharp Obj	17	Civil Serve	0	1000	0	0	Female
17	5016	Osun	Ondo	Dead	AO	A+	None	Divorced	Monogam	Msc	Sharp Obj	30	Trader	0	1192	0	0	Male
18	5017	Gombe	Yobe	Alive	BO	B+	None	Married	Monogam	SSCE	Sexual Inte	47	None	0	902	0	0	Female
19	5018	Lagos	Anambra	Dead	AO	A-	HIV2	Married	Monogam	OND	Sharp Obj	46	None	1	106	1094	1	Female
20	5019	Nassaraw	Niger	Dead	AO	O+	None	Married	Monogam	Msc	Sexual Inte	48	Trader	0	918	0	0	Male
21	5020	Adamawa	Lagos	Dead	BO	AB-	HIV1	Widow	Polygamy	SSCE	No Idea	40	Civil Serve	1	418	782	1	Male
22	5021	Kwara	Lagos	Alive	BB	A-	None	Widower	Monogam	Phd	Sexual Inte	94	Trader	0	920	0	0	Male
23	5022	Nassaraw	Benue	Alive	AA	AB+	HIV2	Married	Polygamy	Bsc	Sharp Obj	66	Business\	1	71	1129	1	Female
24	5023	Kogi	Gigawa	Dead	BO	AB-	None	Widow	Monogam	HND	Birth	92	Business f	0	992	0	0	Female
25	5024	Niger	Akwaibom	Alive	AO	A+	HIV1	Widower	Polygamy	Phd	Sharp Obj	58	Business\	1	317	883	1	Female
26	5025	Abia	Katsina	Alive	BB	B+	HIV1	Widower	Monogam	None	No Idea	70	Business f	1	619	581	1	Female
27	5026	Niger	Delta	Alive	AA	O+	HIV1	Single	Monogam	Phd	Birth	15	Trader	1	693	507	1	Male
28	5027	Bauchi	Kano	Alive	AA	AB-	HIV1	Widow	Polygamy	Technical	Sexual Inte	69	None	1	579	621	1	Male
29	5028	Yobe	Enugu	Dead	AA	AB-	None	Divorced	Monogam	Msc	Birth	46	None	0	889	0	0	Female
30	5029	Kogi	Imo	Alive	AO	O+	None	Divorced	Polygamy	OND	Sexual Inte	25	Civil Serve	0	1110	0	0	Male
31	5030	Abia	Imo	Alive	BO	O-	None	Single	Monogam	Phd	No Idea	18	None	0	913	0	0	Female
32	5031	Kogi	Sokoto	Dead	BO	B-	None	Widow	Monogam	Bsc	Sharp Obj	70	Business\	0	732	0	0	Female
33	5032	Borno	Anambra	Dead	BB	O-	HIV1	Single	Monogam	HND	Sharp Obj	18	Business f	1	371	829	1	Male
34	5033	Delta	Kwara	Alive	AO	B+	None	Widower	Monogam	HND	No Idea	37	None	0	1190	0	0	Male
35	5034	Ogun	Imo	Dead	DD	AD-	None	Single	Polygamy	Phd	No Idea	17	Civil Serve	0	793	0	0	Male
36	5035	Sokoto	Kogi	Dead	BO	O+	None	Married	Monogam	SSCE	Sharp Obj	95	Business f	0	906	0	0	Male
37	5036	Oyo	Oyo	Dead	BB	O-	HIV1	Divorced	Polygamy	SSCE	Sharp Obj	41	Business f	1	375	825	1	Female
38	5037	Benue	Kaduna	Alive	AA	B+	HIV1	Married	Polygamy	Phd	Birth	38	Business f	1	364	836	1	Female
39	5038	Benue	Imo	Alive	BO	AB+	HIV2	Widower	Monogam	None	Sexual Inte	82	Business f	1	171	1029	1	Male
40	5039	Sokoto	Adamawa	Dead	DO	O+	HIV2	Widow	Monogam	SSCE	Sharp Obj	90	Civil Serve	1	40	1152	1	Male
41	5040	Ogun	Adamawa	Alive	BO	AB-	HIV1	Married	Monogam	Bsc	Sharp Obj	49	Business f	1	326	874	1	Female
42	5041	Gigawa	Cross Rive	Dead	BB	O-	HIV1	Divorced	Polygamy	Msc	Birth	44	Business f	1	695	505	1	Female
43	5042	Osun	Oyo	Alive	AA	AB+	HIV2	Widower	Monogam	Phd	No Idea	88	None	1	24	1176	1	Female

Fig. 4.5: Sample patient data



□ Breaking of Hair □ Loss of Weight

☑ Fever ☑ Headache

☑ Cold □ Soar Mouth

☑ Viral Infection ☑ Sore in the

Elisa:

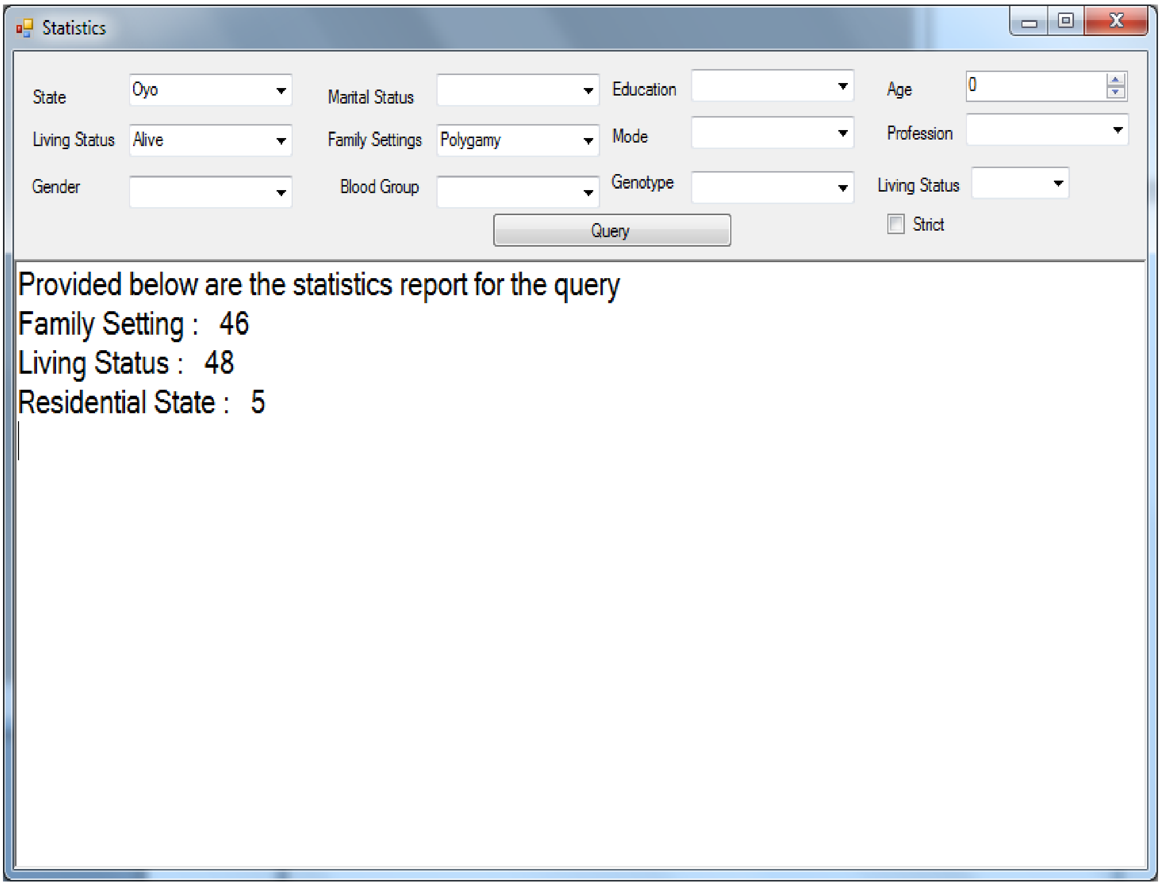
CD4: 0 452

Viral Load: 0 748

Hidden Layer: 1

```
Mean Square Error: 0.0437852025831501
Root Mean Square Error: 0.209249139981865
Threshold 0.630000000000001
False Positive [FP]  0
True Positive [TP] 45
False Negative [FN] 18
True Negative [TN] 37
correctly predicted 82
incorrectly predicted 18
% correctly predicted 82
% incorrectly predicted 18

           There is a very high tendency that you are HIV positive

--------------------------------------------------------------

           There is a very high tendency that you are HIV positive

--------------------------------------------------------------

           There is a very high tendency that you are HIV positive

--------------------------------------------------------------

           The level of opportunistic disease is low to determine whether you are HIV negative or
Positive
           Go for medical checkup

--------------------------------------------------------------
```

Fig. 4.6: Classification/Prediction window



State: Oyo

Living Status: Alive

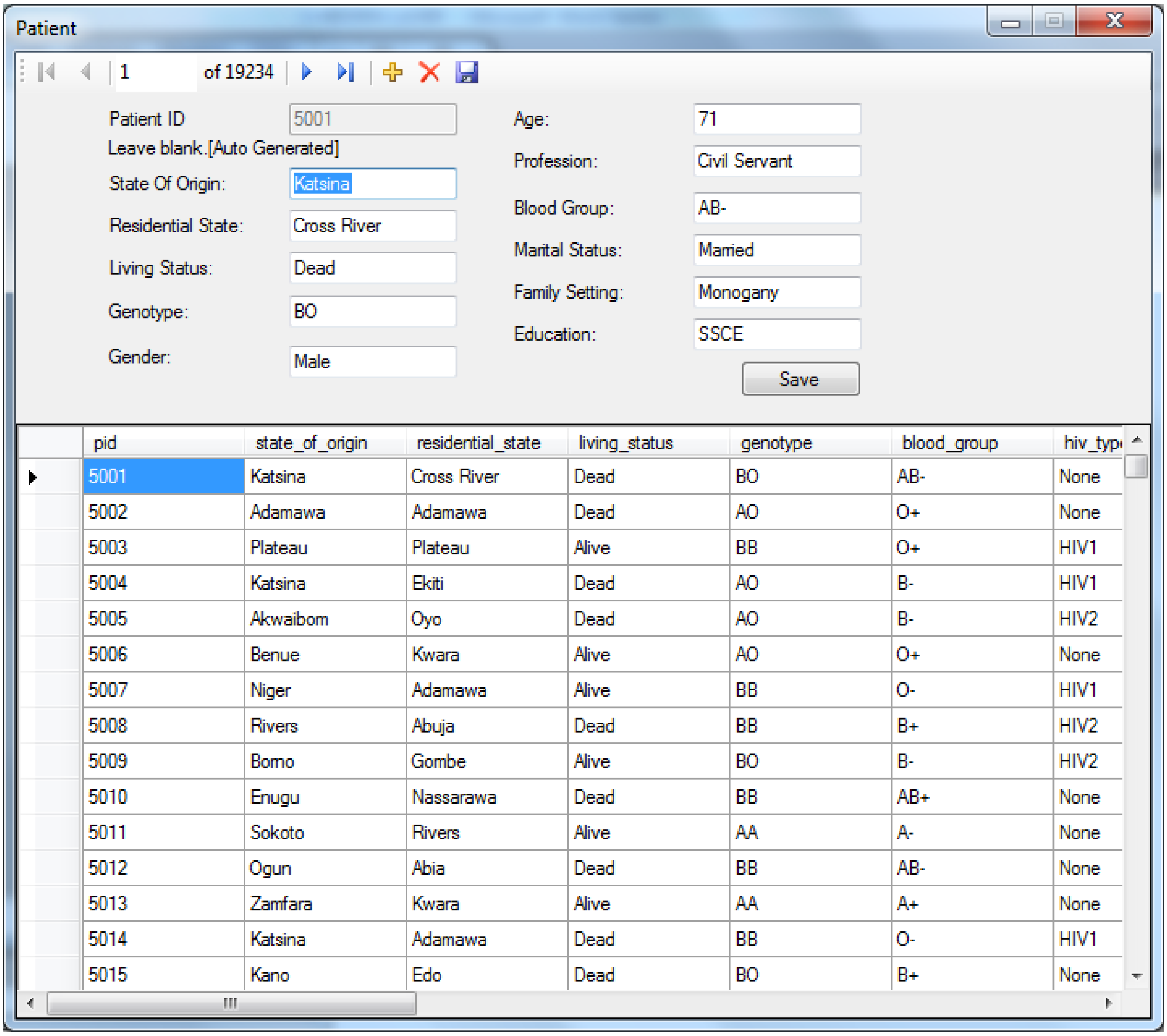
Family Settings: Polygamy

Age: 0

```
Provided below are the statistics report for the query
Family Setting :   46
Living Status :   48
Residential State :   5
```

Fig. 4.7: Demographic/Statistical Window

WEKA Interfaces

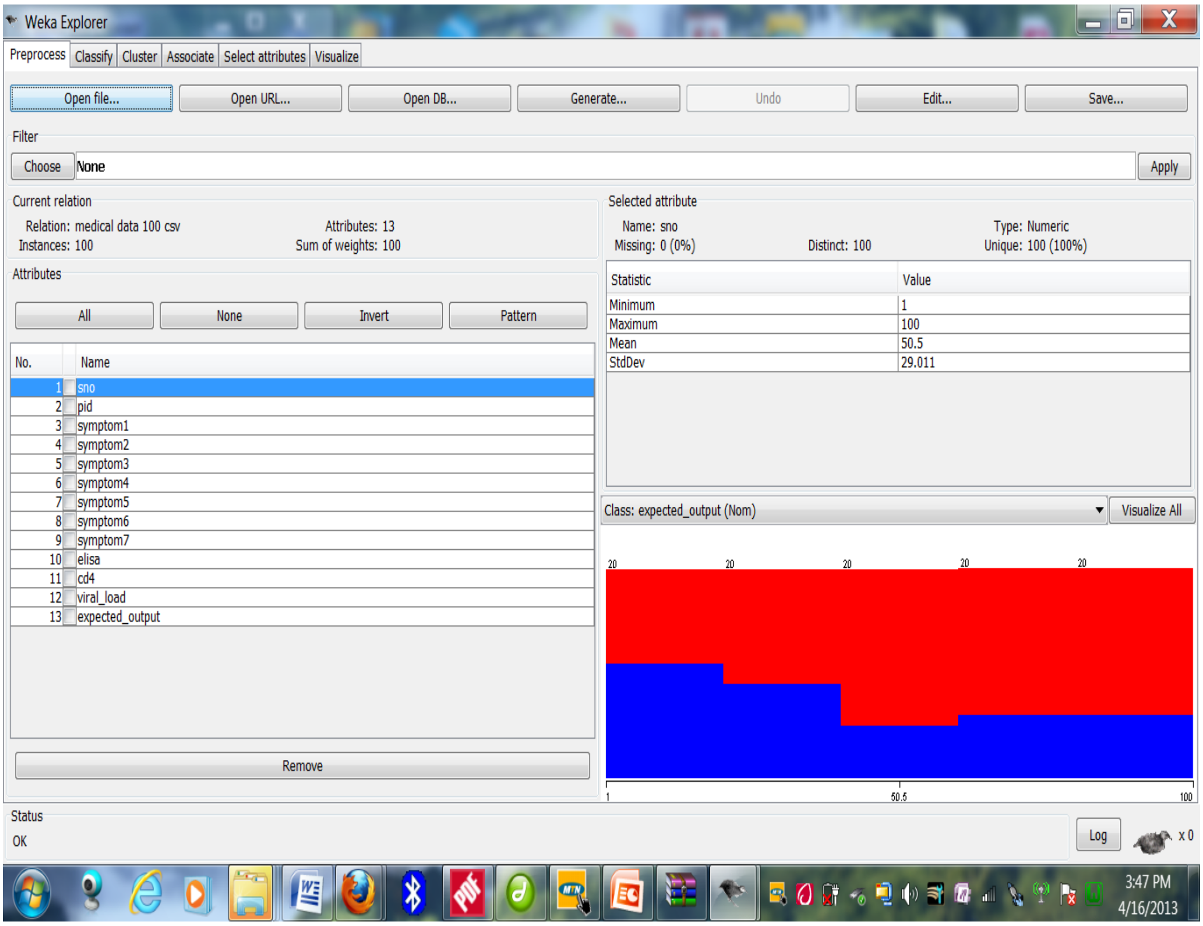


Patient ID 5001
Leave blank.[Auto Generated]
State Of Origin: Katsina
Residential State: Cross River
Living Status: Dead
Genotype: BO
Gender: Male

Age: 71
Profession: Civil Servant
Blood Group: AB-
Marital Status: Married
Family Setting: Monogany
Education: SSCE

pid	state_of_origin	residential_state	living_status	genotype	blood_group	hiv_typ
5001	Katsina	Cross River	Dead	BO	AB-	None
5002	Adamawa	Adamawa	Dead	AO	O+	None
5003	Plateau	Plateau	Alive	BB	O+	HIV1
5004	Katsina	Ekiti	Dead	AO	B-	HIV1
5005	Akwaibom	Oyo	Dead	AO	B-	HIV2
5006	Benue	Kwara	Alive	AO	O+	None
5007	Niger	Adamawa	Alive	BB	O-	HIV1
5008	Rivers	Abuja	Dead	BB	B+	HIV2
5009	Borno	Gombe	Alive	BO	B-	HIV2
5010	Enugu	Nassarawa	Dead	BB	AB+	None
5011	Sokoto	Rivers	Alive	AA	A-	None
5012	Ogun	Abia	Dead	BB	AB-	None
5013	Zamfara	Kwara	Alive	AA	A+	None
5014	Katsina	Adamawa	Dead	BB	O-	HIV1
5015	Kano	Edo	Dead	BO	B+	None

Fig. 4.8: HIV/Aids Patient Window



Filter

None

Current relation

Relation: medical data 100 csv   Attributes: 13
Instances: 100   Sum of weights: 100

Selected attribute

Name: sno   Type: Numeric
Missing: 0 (0%)   Distinct: 100   Unique: 100 (100%)

Statistic	Value
Minimum	1
Maximum	100
Mean	50.5
StdDev	29.011

Attributes

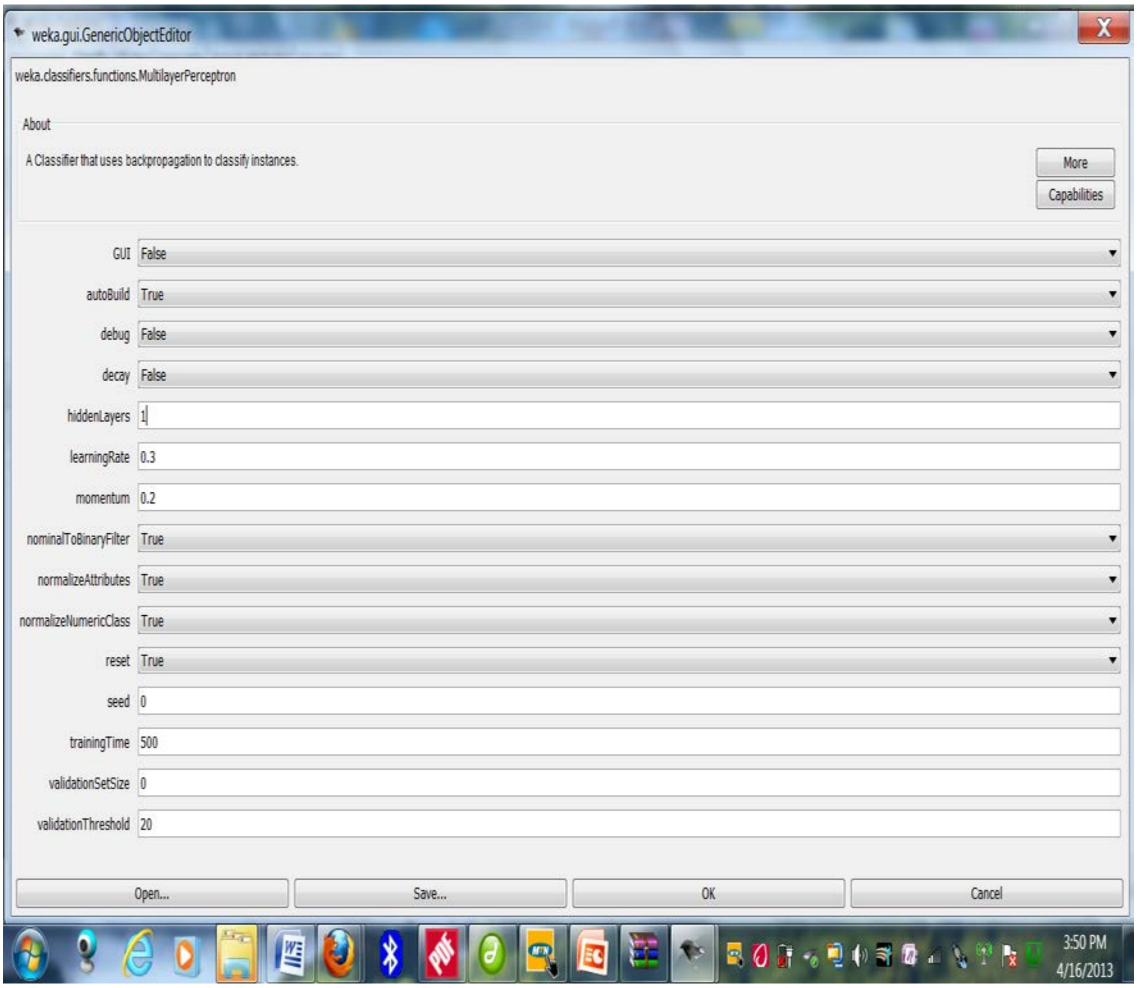
No.	Name
1	sno
2	pid
3	symptom1
4	symptom2
5	symptom3
6	symptom4
7	symptom5
8	symptom6
9	symptom7
10	elisa
11	cd4
12	viral_load
13	expected_output

Class: expected_output (Nom)

Status

OK

Figure 4.9: Dataset Loading using WEKA Explorer



weka.gui.GenericObjectEditor

weka.classifiers.functions.MultilayerPerceptron

About

A Classifier that uses backpropagation to classify instances.

GUI	False
autoBuild	True
debug	False
decay	False
hiddenLayers	1
learningRate	0.3
momentum	0.2
nominalToBinaryFilter	True
normalizeAttributes	True
normalizeNumericClass	True
reset	True
seed	0
trainingTime	500
validationSetSize	0
validationThreshold	20

Figure 4.10: Parameter Settings for Function Classifier with Multilayer Perceptron (Hidden Layer 1,2,3)

4.3.1 Model Performance

Table 4.1 Settings of system parameters.

Parameters	**value**
Population size test data	100
Mutation	1%
Crossover	50%
Epochs	2000 iterations
Network layer	3 layers - 8 Input layer - 3 Hidden layer - 1 Output layer
Number of weights	12 variable weights

Table 4.2: Model performance with 1 hidden layer on the transformed data

Parameters	Value
Hidden layer	One
Mean square error	2.01 x10^{-9}
Root Mean Square Error	2.9 x 10-^{16}
Percentage correctness of the predicted value	98.0%
Percentage incorrectly predicted value	2%

Table 4.2 shows that the model performance with one hidden is a strong network for the prediction of HIV patient's status because it generated a lower Root mean Square error and the percentage Accuracy of 98% shows a better and efficient predicted value.

The model performance with 2 hidden layer shows also a better performance with low Root Mean Square Error and high percentage but not as 1 hidden layer, this still reflects that the model with 2 hidden layer still predict accurately but not as accurate as 1 hidden layer.

The model performance with 3 hidden layers shows a higher root mean square error than the first and two hidden layers.

4.3.2 Model Performance with 3 Hidden Layer on the Transformed Data

The model performance with 3 hidden shows a higher Root Mean Square Error and less percentage accuracy than the one, two, and three hidden layers.

4.3.3 Root Mean Square Error (RMSE) performance.

The root mean square error (RMSE) is the one of the best error performance measure metrics. Table 4.2, Table 4.3, and Table 4.4give the root mean square error (MSE) as a function of the number of the hidden layer 1, 2, 3 and respectively for the network models.

The goal of changing the hidden layers is to arrive at the least Root Mean Square Error as possible and it can be observed from the Table 4.2 Table 4.3 Table 4.4 and table 4.2 for 1 hidden layer was the best model with 98% correctly predicted output and lowest ROOT MEAN SQUARE ERROR (RMSE) of 2.9 x 10-^{16}.

Table 4.3: Model performance with two hidden layer on the transformed data.

Parameters	Value
Hidden layer	Two
Mean square error	0.004
Root Mean Square Error	9.87 x 10-^{15}
Percentage correctness of the predicted value	97%
Percentage incorrectly predicted value	3%

Table 4.4: Model Performance with 3 Hidden Layer on the Transformed Data

Parameters	Value.
Hidden layer	Three
Mean square error	0.005
Root Mean Square Error	7.0 x 10-^{8}
Percentage correctness of the predicted value	90.0%
Percentage incorrectly predicted value	10%

Table 4.5 Overall Neuro-Genetic Model Prediction and Classification Result.

Number hidden layer	Training RMSE	% correctly predicted output	% incorrectly predicted output
1	2.9 x 10-^{16}	98%	2%
2	9.87 x 10-^{15}	97%	3%
3	7.0 x 10-^{8}	90%	10%

The prediction/ classification results using Neuro-Genetic Algorithm model for HIV /AIDS patients generated different percentage accuracies and different forecast error rate respectively by varying the hidden layers as shown from table 4.5 summarized above, it can be observed that 6 percentage correct prediction and classification for 1 hidden layer is 98% was the best with the lowest root mean square error of 2.9 x 10-^{16}.

It is also observed from table 4.6 that 3 hidden layer network recorded the highest mean MSE when compared with other hidden layers. This could be a good indication that the network did not fully learn the problem with only 3 hidden layers and it could also indicate that the problem being solve is not linearly separable. Varying the number of hidden layer resulted in significant improvement in the minimum mean (MSE) as can be seen from table 4.5 and table 4.6 and can also be that the mean MSE for 1 and hidden layers has the lowest mean square error (MSE).

Table 4.6 Neuro Genetic Mean Square Error Rate Performance

Hidden layer	MSE
1	0.0000009
2	0.04
3	0.005

Even from the mean square error table, it is also observed that 1 hidden layer network model recorded the least forecast error rate, which still shows that network model with 1 hidden layer shows a better performance in its prediction ability.

4.3.4 Mean Square Error Performance

Thus the models with hidden layer 1 have the least mean square error (MSE) of 0.000009 and root mean square error rate of 2.9 x 10^{-16} (RMSE) recorded by the model. From table 4.2, to 4.6 it can be seen that the different hidden layers were used and percentage accuracies were also varied up till 3 hidden layers, and root mean square error also varies. The minimum root mean square error (RMSE) for 1, 2, and 3 hidden layers networks are 2.9 x 10^{-16}, 9.87 x 10^{-15}, 7.0 x 10^{-8} and respectively.

Combining these results, the model of 1 hidden layer has the least root mean square error of 2.9 x 10^{-16} followed by RMSE of 9.87 x 10^{-15} respectively as shown in table 4.5

High root mean square error (RMSE) were recorded by the models with 3 hidden layer while the model with 1 hidden layer recorded the least error of 9.87 x 10^{-15} of RMSE and 0.00000091 (MSE), therefore it has the best generalization capability, so multilayer perception neural network genetic based with 1 hidden layer was the best model selected for the study.

4.3.5 Learning Curves

The plot of RMSE across hidden layers is shown below to illustrate the performance of the model.

The details of results of the model that emerge the best in each hidden layers (hidden layers 1,2, 3) including their learning curves

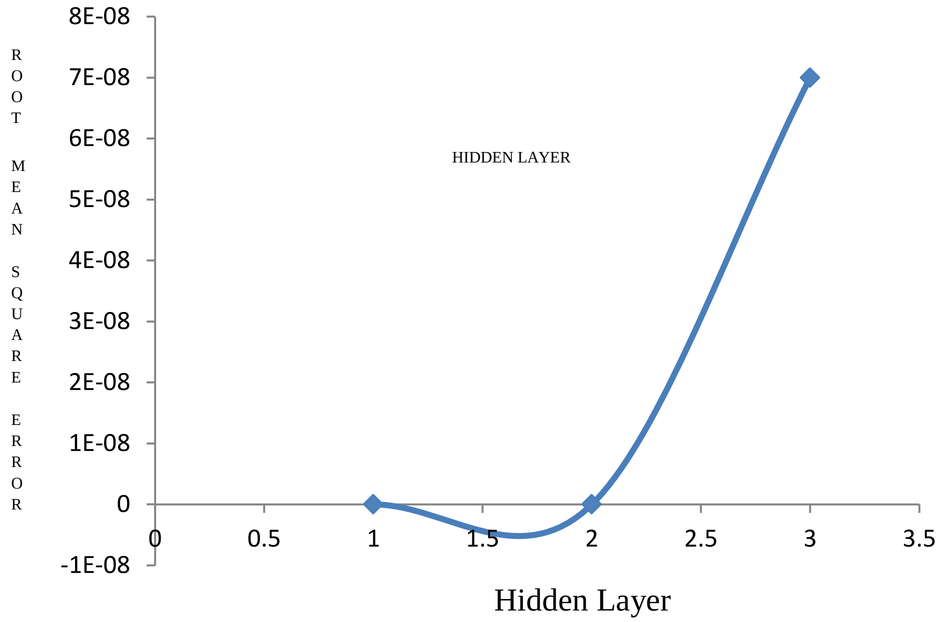


Fig. 4.11 Graph of NEGEM Model

From the graph above it shows that one hidden layer has the least root mean square error compared to the error rate generate by the other hidden layers which shows that Neuro Genetic Model with one hidden layer was the best model selected for the study

Fig. 4.12 shows a graph of hidden layer versus percentage of correctly predicted output which shows that the percentage accuracy of NEGEM varies as the number of hidden layer changes. It also shows that one hidden layer has the highest percentage accuracy, so NEGEM with one hidden layer is the best model.

Figure 4.13 shows a graph of hidden layer versus percentage of incorrectly predicted output. This shows that the percentage of incorrectly predicted output of NEGEM also varies as the number hidden layer changes. From above it is revealed that NEGEM with one hidden layer has the least incorrectly predicted output. So, NEGEM with one hidden layer is the best model for the study.

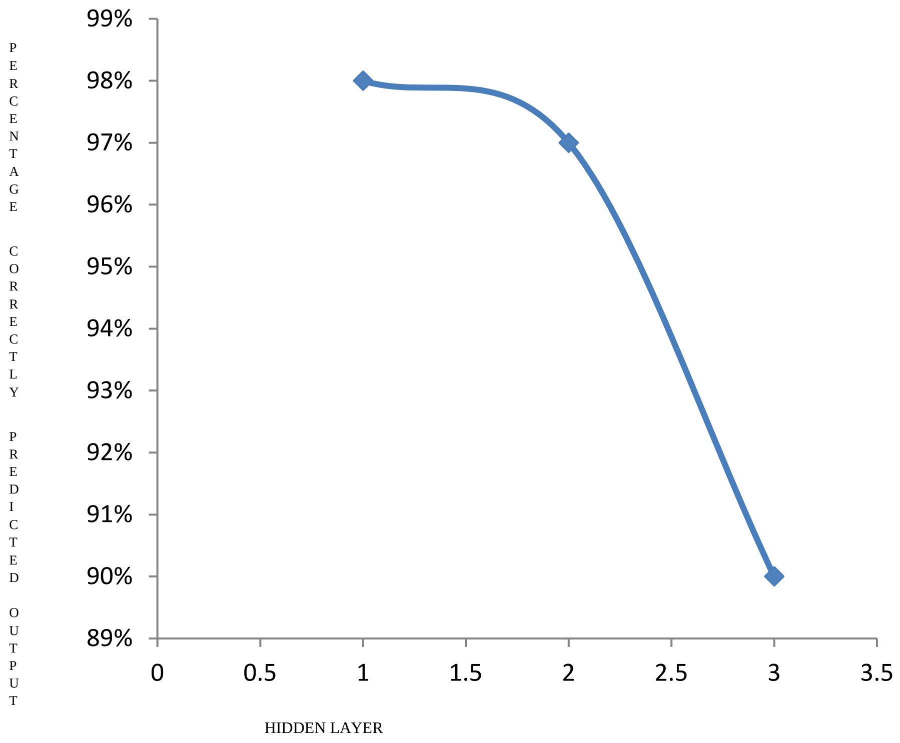


Fig. 4.12 NEGEM graph of hidden layer versus percentage correctly Predicted output

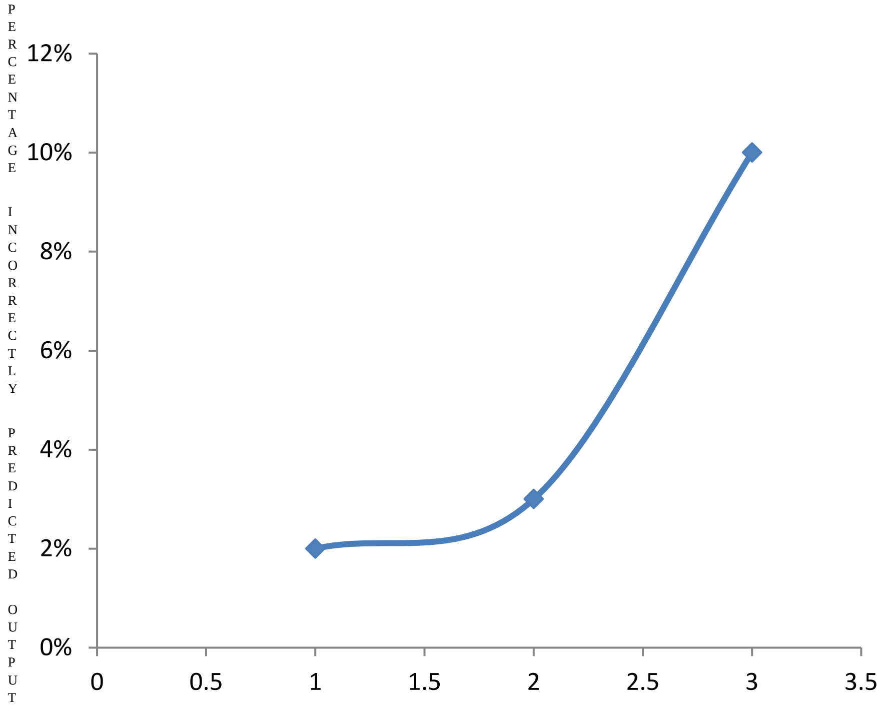


Fig. 4.13 Graph of percentage incorrectly predicted output versus hidden layers

4.4 Model Selection

The following conclusions can be drawn from the results analyzed. The performance of the 1 hidden layer is very impressive, because it generated a lower forecast error in its prediction value, with high percentage correctly predicted value, in the training set. It shows that it has a good generalization capability and it does not overfit as it is observed in the values of RMSE and MSE, generated by the model.

Neural network researchers have shown that too many weights lead to poor generalization. This is also confirmed from the increase in the value of RMSE's as the number of hidden layers (number of weight) increases. The good performances recorded in the training set by 1- hidden layer and 2 hidden layer models are a result of good training. This is clearly indicated by small error of 2.9 x 10^{-16} that was achieved with 1 hidden layer followed by 9.8 x 10^{-15} and 7.0 x 10^{-8} with 2 hidden layers models respectively.

4.5 Results of Prediction, Classification and Accuracy of Result of NEGEM

When the performance of the network was gauged using the test data sample. The result of table 4.7 was obtained.

Table 4.7: The predicted output and expected output for NEGEM

Expected output	Actual network Output
0	0
1	1
1	1
0	0
0	0
0	0
1	1
1	1
1	1
0	0
0	0
0	0
0	0
1	1
1	0
0	0
1	1
1	1
0	0
0	0
0	0
0	0
1	1
1	1
1	1

1	1
1	1
0	0
0	0
0	0
1	1
1	1
1	1
1	1
1	1
0	0
0	0
1	1
1	1
1	1
1	1
1	1
0	0
0	0
0	0

The percentage correctness is:

$$PC = \frac{\text{number of correct samples}}{\text{Total number of predicted sample}} \quad x\ 100$$

$$= \quad 44/45 \text{ x } 100$$

$$= \quad 97.777$$

From the result of the study, the percentage accuracy for correctly predicted output is about 98.0% and percentage incorrectly predicted output is about 2% for 1 hidden layer. This shows that the Neuro-genetic algorithm (NEGEM) model performs excellently in the prediction of HIV/AIDS patient status with one hidden layer.

Table 4.8Percentage Correctness for Prediction Output

Hidden layer	%correctness	% incorrectness
1	98.0%	2%
2	97%	3%
3	90%	10%

Table 4.8 shows that the percentage of correctly predicted output for all the hidden layers is much higher than incorrectly predicted output. This shows a good prediction capability.

4.6 Analysis of results of Multilayer Perception Using WEKA Data Mining Tool

WEKA Data mining tools were used for the same training data used in NEGEM model. Tables 4.9to 4.10 and figure 4.12show the analysis of WEKA's results

Table 4.9 Performance of the MLP- WEKA Model

Hidden layer	Root mean squared error(RMSE)
1	0.003
2	0.0008
3	0.137

Performance of MLP-WEKA Model for One Hidden Layer.

Perimeters	Value
Mean absolute error	0.0022
Root mean squared error	0.003
Relative absolute error	0.47

Table 4.10 Performance of the MLP-WEKA Model for 2 Hidden Layer

Parameters	Value
Mean absolute error	0.0008
Root mean square error	0.0008
Relative absolute error	0.1582

The mean absolute error is the same as root mean square error which is the same value recorded in table 4.10.

Fig. 4.14 shows the MLP-WEKA graph of hidden layer versus RMSE which also shows that the RMSE varies as the hidden layer changes. It shows that WEKA recorded a higher Root mean square root compared with NEGEM

4.6.1 Percentage accuracy of MLP-WEKA

The percentage predicted accuracy of WEKA classifier recorded a constant value in one, two and three hidden layers which shows a serious over-fitting in its predicted value, this is because WEKA did not implement genetic algorithm (GA) an Optimization algorithm. GA has a good fitness function that avoids low convergence speed and over-fitting. GA also has global optimization strength during training which helps in minimizing weights generated that cause low convergence speed common with ordinary multilayer perception neural network.

The percentage accuracy of NEGEM is 98% which indicates a strong accuracy in prediction and classification value of NEGEM compared to WEKA that over-fits in its percentage accuracy of the predicted output.

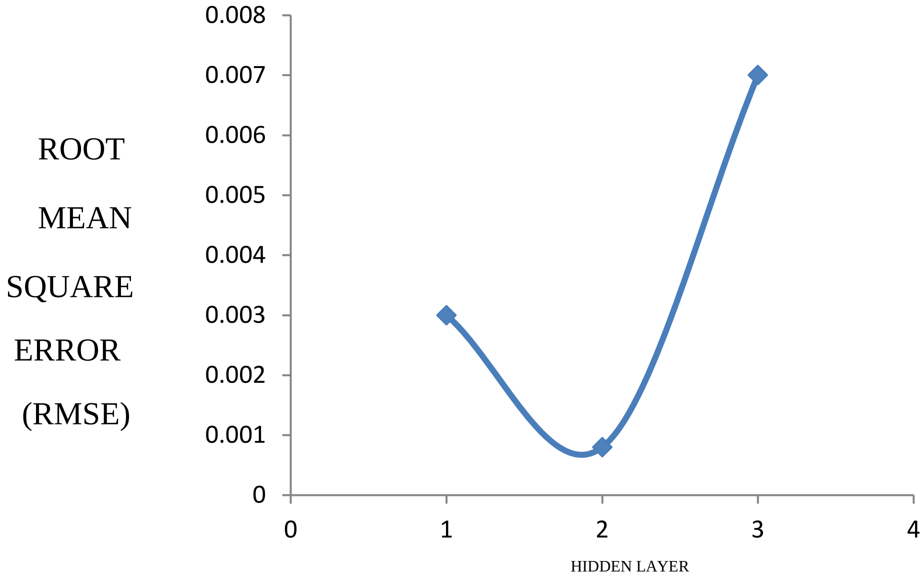


Fig. 4.14 Graph of MLP-WEKA Model

Table 4.11: The expected value and the predicted network value of the MLP- WEKA

Expected value	Predicted Network Value
1	0
1	0
0	0
1	1
1	1
0	0
1	1
1	1
0	0
0	0
0	0.
0	0.
1	0
1	1
0	1
0	0
1	1

4.7 Comparative Analysis of NEGEM and WEKA Results

The Waikato Environment for Knowledge Analysis (WEKA) Data mining software containing a number of the state – of – the art learning algorithms was used to benchmark the capability of NEGEM. It was found out that NEGEM outperforms WEKA. The results are recorded in Tables 4.12 to 4.16

From the table 4.12, NEGEM records low Mean Absolute Error in all the hidden layers than WEKA which shows that it's predictive / classification accuracy is higher than WEKA which records higher MAE.

From table 4.13, the recall value of NEGEM for all the one, two and three hidden layers records positive range for prediction accuracy which is between 0 and less than 1 while WEKA over-trains in it recall value.

Table 4.14 shows that the precision value of NEGEM gives a better performance in prediction/classification based on the accuracy acceptable range of precision performance which is the range of 0 and less than 1 than WEKA.

From table 4.15, NEGEM shows a better performance in percentage in percentage accuracies than WEKA in all the 3 hidden layers. This indicates that NEGEM outperform WEKA.

Table 4.12 Mean Absolute Error Performance Measure

Hidden layer NEGEM	NEGEM	WEKA
1	0.0009	0.001
2	0.00	0.001
3	0.00030	0.0007

Table 4.13 Recall Performance Measure

Hidden Layer	NEGEM	WEKA
1	0.98	1.0
2	0.001	1.0
3	0.98	1.0

Table 4.14Precision Performance Measure

Hidden layer	NEGEM	WEKA
1	0.96	1.0
2	0.96	1.0
3	0.98	1.0

Table 4.15 Mean Class Range Percentage Performance of WEKA

Hidden Layer	NEGEM	WEKA
1	98%	50%
2	97%	50%
3	90%	50%

Table 4.16: Comparative Performance of Neuro-Genetic Model (NEGEM) and MLP WEKA

Hidden layers	NEGEM (RMSE)	WEKA (RMSE)
1	2.9 x 10-^{16}	0.001
2	9.87 x 10^{-15}	0.0008
3	7.0 x 10^{-8}	0.137

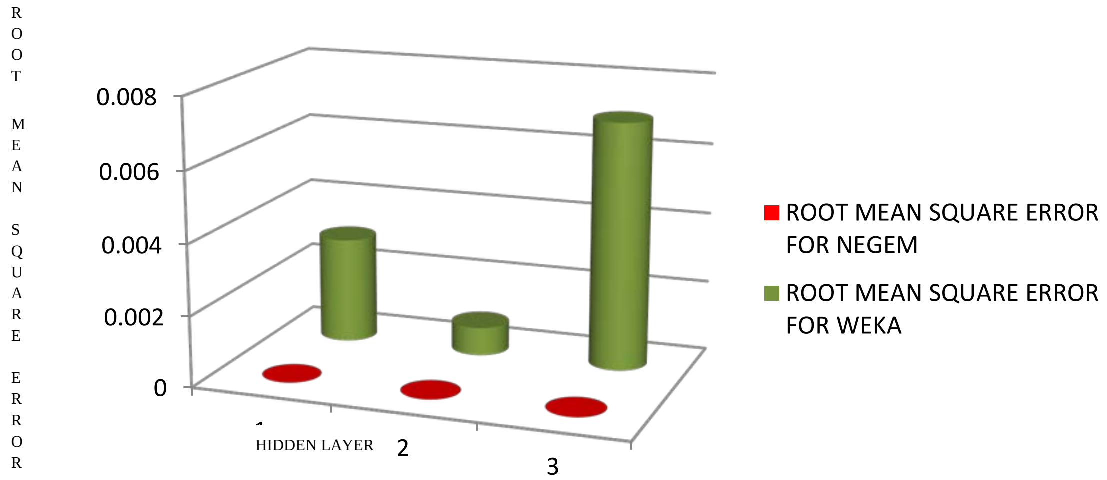


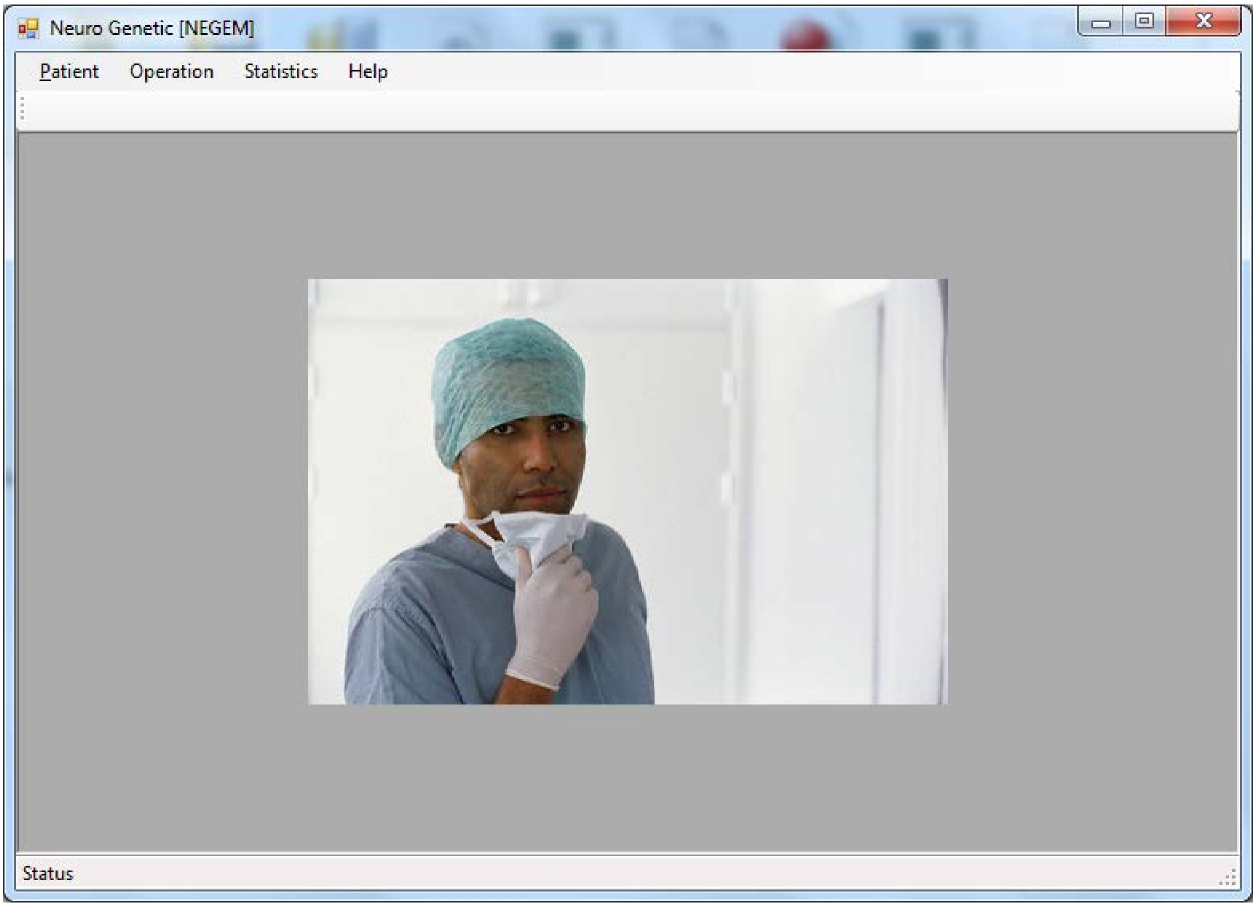
Fig. 4.15 Comparative Graph Of Error Performance Rate Of Negem Versus Mlp-Weka

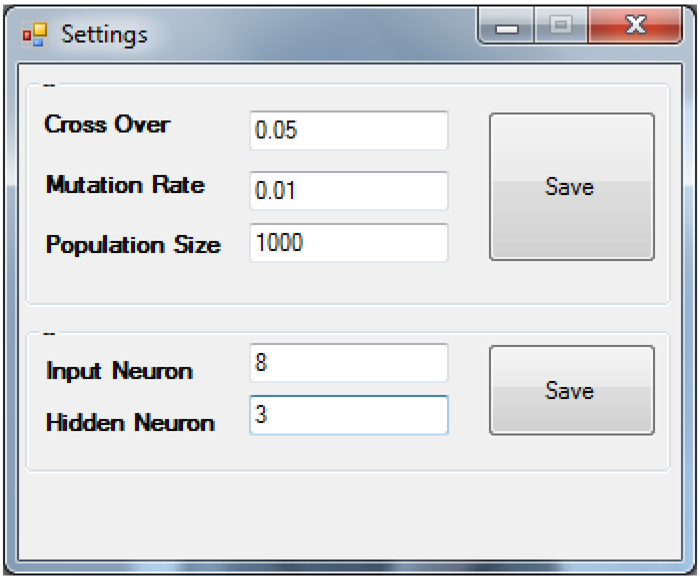
From the Fig.4.15 and Table 4.16, NEGEM model outperforms MLP- WEKA data mining tool in that NEGEM for ONE hidden layer recorded a lower root mean square error (RMSE) of 2.9 x 10^{-16} while ordinary multi-layer perception implemented in WEKA recorded RMSE of 0.001 for one hidden layer. From table 4.16 and the figure 4.13 RMSE of 1, 2, 3 hidden layers of NEGEM recorded very low predicted error rate compared to the RMSE of WEKA which recorded a higher RMSE in all the 1,2, and 3 hidden layers.

4.8 Implementation of Neuro-Genetic Model

When the application starts it creates main window named Neuro-Genetic [NEGEM] as shown Screen Shot Topology below. The window contains the main menu of the application. To use the Neuro-Genetic model, the user has to open the operation menu by clicking on it. The user is expected to train the network before it can predict. Training the network has to do with setting up the topology and parameters.

After the user has filled all expected values for the algorithm, automatically, a screen shot comes up to inform the user about the success of the training and if he want to test the system. The test button becomes enabled. The screen shot shows the result of the training that has been carried out. It shows the expected output versus network output [predicted output] and it also shows that the status of the patient





Cross Over 0.05

Mutation Rate 0.01

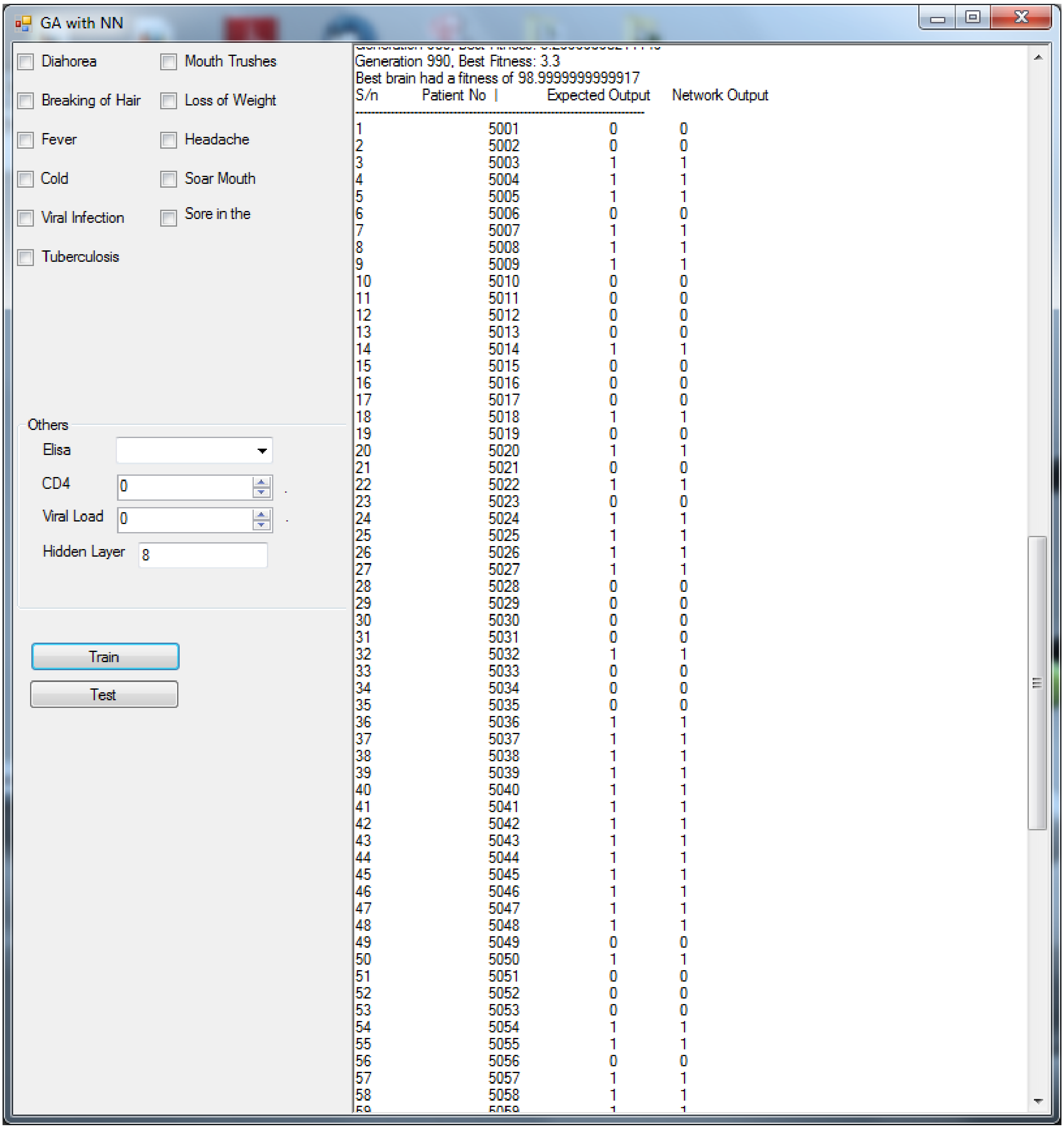
Population Size 1000

Input Neuron 8

Hidden Neuron 3

Screen shot Topology

TESTING RESULT



Elisa

CD4 0

Viral Load 0

Hidden Layer 8

```
Generation 990, Best Fitness: 3.3
Best brain had a fitness of 98.9999999999917
S/n        Patient No |        Expected Output    Network Output
-----------------------------------------------------------------
1                  5001               0            0
2                  5002               0            0
3                  5003               1            1
4                  5004               1            1
5                  5005               1            1
6                  5006               0            0
7                  5007               1            1
8                  5008               1            1
9                  5009               1            1
10                 5010               0            0
11                 5011               0            0
12                 5012               0            0
13                 5013               0            0
14                 5014               1            1
15                 5015               0            0
16                 5016               0            0
17                 5017               0            0
18                 5018               1            1
19                 5019               0            0
20                 5020               1            1
21                 5021               0            0
22                 5022               1            1
23                 5023               0            0
24                 5024               1            1
25                 5025               1            1
26                 5026               1            1
27                 5027               1            1
28                 5028               0            0
29                 5029               0            0
30                 5030               0            0
31                 5031               0            0
32                 5032               1            1
33                 5033               0            0
34                 5034               0            0
35                 5035               0            0
36                 5036               1            1
37                 5037               1            1
38                 5038               1            1
39                 5039               1            1
40                 5040               1            1
41                 5041               1            1
42                 5042               1            1
43                 5043               1            1
44                 5044               1            1
45                 5045               1            1
46                 5046               1            1
47                 5047               1            1
48                 5048               1            1
49                 5049               0            0
50                 5050               1            1
51                 5051               0            0
52                 5052               0            0
53                 5053               0            0
54                 5054               1            1
55                 5055               1            1
56                 5056               0            0
57                 5057               1            1
58                 5058               1            1
```

Screen Shot Output

CHAPTER FIVE

SUMMARY, CONCLUSION AND FURTHER WORK

5.1 Summary

This research work developed a Neuro-Genetic Algorithm model (NEGEM) by embedding genetic algorithm into multilayer perception artificial neural network for classification and prediction using HIV/AIDS medical data.

The data set consists of 14 input variables and 1 output variable that has only two possible outcomes. The input data represent the factors that affect HIV/AIDS classification status while the output variable is the HIV/AIDS status (HIV/AIDS positive and HIV/AIDS negative).

Demographic and treatment data on HIV/AIDS patients from 2000-2011 were collected from selected tertiary and general hospitals, primary health care and non-governmental organizations in southwestern Nigeria. The data were saved in a created two-tier medical database which allows for multidimensional analysis. Three different MLP-hidden layers (one, two and three) networks implemented in C# programming language and Microsoft Structured Query Language (SQL) server were used to develop the Neural Network topology to predict and classify HIV/AIDS data.

A model combining multi-layer perceptron neural network and genetic algorithm was used to train and optimize weight in order to avoid low convergence speed of artificial neural network and also reduce over-fitting.

The sample data was divided into three -training set, verification and test data in ratio 70:10:20 respectively. Genetic algorithms parameters of mutation rate of 1%; crossover of 50%; population size of 100; epoch length of 2,000 iterations were used to develop the model in order to avoid premature convergence to a local optimum and over-fitting.

The number of hidden layers was varied to give different network topological models of different sizes. These models were then evaluated based on Mean Square Error (MSE) l, Root Mean Square (RMSE), Recall & Precision matrix as well as percentage accuracy of the predicted value to determine best model for prediction. Neuro- Genetic model with one hidden layer that gave the best result was then chosen as the prediction model. The verification data set was then applied to the model. The model gave 98% accuracy without over fitting and premature convergence.

The ability of the selected NEGEM model to classify and predict was compared with Waikato Engineering Knowledge Analysis (WEKA), an existing MLP software. Classification and predictive accuracies were measured using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Recall and precision were used to measure the level of true positive prediction/classification and over-training. NEGEM model outperformed WEKA model.

5.2 Conclusion

The research work shows the effectiveness of Neuro-Genetic model in mining database for prediction and classification. The study indicated the strength of genetic algorithm embedded in Artificial Neural Network in reducing over-fitting and avoiding low convergence speed during training in order to obtain a better accuracy in the prediction and classification value. The results indicated that artificial neural network genetic algorithm based yielded better results especially with one hidden layer which has a lower root mean square error in its prediction value.

Contributions of the Research to Knowledge

NEGEM model improved the predictive and classification performance of the Multilayer Perceptron (MLP) artificial neural network (ANN) by using the global search capability of Genetic algorithm to:

1. Minimize the problem of low convergence speed during learning process of MLP-ANN
2. Avoid over fitting and overgeneralization

5.3 Further Works

Further works will involve the Comparison of Neuro- Genetic (NEGEM) algorithm with other data mining algorithms like decision tree and logistic regression using the same medical database. Further work will also be extended to mine patterns that will be used for prediction of drug pattern for HIV/ AIDS patient’s data. Model can be extended to evaluate an established approach for identifying new drug safety signals. NEGEM model can also be used to mine other Database application for prediction/classification.

REFERENCES

Adeyemo. O. O. and Osofisan, A. O. 2012. V-Diagnostic: A Data Mining System For Human Immuno-Defeciency Virus Diagnosis, *International Journal of Computer Science and Information Security*, vol 10,77-81.

Adeyemo. O. O. and Osofisan, A.O 2012. Empirical Study of Statistical Based Method and Artificial Neural Networks for credit rating in Nigeria banks. *International journal of Research and Reviews in Computer Science,* vol 3, no. 3, 731-1737.

Anderson, R.M, Medley, G.F., May, R.M., and Johnson, A.M. A preliminary study of the transmission dynamics of the human immunodeficiency virus (HIV), the causative agents of AIDS IMA, *Journal of Mathematics Applied in Medicine and Biology,* 229–263

Andes, A. Widow, B., Lehr, M. and Wan, E. 1990. MRIII: A Robust algorithm for training analog neural networks. *In Proc. Intl. Joint Conf. On Neural Networks, Washington, DC,* vol 1,533-536.

Antonia J.J. 2002. Genetic algorithms and their applications to the design of neural networks. Neural Computing & Applications.

Apte C., Liu B., E.P.D. Pednault, and P. Smyth 2002. Business applications of data mining, Communications of the ACM.

Bailey-Kellog, C. Ramakrishnan, N. and Marathe, M. 2006. Spatial Data Mining to Support Pandemic Preparedness. SIGKDD Explorations (8) 1, 80-82.

Becker, S. and Le Cun, Y. 1989. Improving the convergence of the back-propagation learning with second-order methods. *In Hinton, G. Touretzky, D. and Sejnowski, T., editors, Proc 198 Connectionist Models Summer School, Pittsburgh*. 29-37.

Benc, R., 1997. Application of Artificial Neural Networks to power systems, *http://members.tripod.com/ann.2ps/toc.html*. 2nd July, 2003.

Berry, Michael J.A. and Linoff, Gordon, S. 2004. Data Mining Techniques for Marketing Sales and Customer Support.

Berry, Michael, J.A. and Linoff, Gordon, S. 1997. Data mining Techniques for Marketing Sales and Customers Support.

Betechuoh B.L., Marwala T. and Tettey T. 2006. Autoencoder networks for HIV classification, current science, vol. 91, no. 11.

Betechuoh, B.L. Marwala T. and Manana, J.V. 2008. Computational Intelligence for HIV Modelling. *International Conference on Intelligent Engineering Systems INES*: 127 – 132

Bioinfo Optimization Theory & Applications Volume 1,01-05 *http://www.bioinfo.in/contents.php?id=292*

Bock, R. K. 1998. Sigmoid Function. *http://rkb.home.cern.ch/rkb/ AN16pp/mode259.html*. 20th August, 2003.

Bose, N.K. and Liang, P. 1996. Neural Network Fundamentals with Graphs, Algorithms and Applications. Mc.Graw-Hill Inc.

Brailsford, S.C., 1993. Modelling for HIV Infection and AIDS, *Ph.D. Thesis, University of Southampton.*

Cai C.H., Fu, W.C. Cheng, C.H. and Kwong, W.W. 1998. Mining association rules with weighted items, *Proceedings of 1998 International Database Engineering and Applications Symposium, Cardiff, Wales*.

Cao, X., Maloney, K.B. and Brusic, V. 2008. Data mining of cancer vaccine trials: a bird's-eye view. *Immuno Research, 4:7. DOI: 10.1186/1745-7580-4-*7

Chaturvedi D.K. 2005. Dynamic Model of HIV/AIDS Population of Agra Region.

Chengic, T.H., Wei, C.P., and Tseng, V.S. 2006. Feature Selection for Medical Data Mining: Comparisons of Expert Judgment and Automatic Approaches. *Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems* (CBMS'06).

Dan C. 2011. Beginning C# Object-Oriented Programming. Apress**.**

David H., Heikki M., and Padhraic S. 2001. Principles of Data Mining, ISBN 0-262-08290.

Dechao W., Brendan L. , Andrew R., Julio M., Richard H., Frank D’ W., Joep L., Scott W., Lidia Ruiz , Marı´a Je´sus Pe´rez-Elı ´as, Sean Emery, Jose Drakos, N. 2009, Compare three computational methods for the prediction of virological response to combination HIV therapy. *http://cbl.leeds.ac.uk /nikos/pail/intml/subsection 3.11.4.html*

Dubois D., E. Hullermeier, H. P. 2006. A systematic approach to the assessment of fuzzy association rules, Data Mining and Knowledge Discovery.

Exarchos T.P., Papaloukas, C. Fotiadis, D.I., Michalis, L.K. 2006. An association rule mining-based methodology for automated detection of ischemic ECG beats, *IEEE Transactions on Biomedical Engineering.*

Fayaad Gregory Piatetsky-shapiro and Padhraic Smyth, 2006 Framework for Knowledge Discovery in Databases and Data Mining.

Fibak J., Pawlak Z., Sowinski K., Sowinski R., 1986. Rough sets based decision algorithm for treatment of duodenal ulcer by HSV, Biological Sciences 34:

Frawley W.J., Piatetsky-Shapiro G., Matheus C.J., 1991. Knowledge discovery in databases: *an overview, The AAAI Workshop on Knowledge Discovery in Databases.*

Frohlich, J. 1996. What are Neural Networks? *http:iirfhs8012.fh.regensburg.de/v saj39122/jfro ehl/diplom/e-11-text.html.*

Gatell, A. D. M., Carlo T., Maurizio Z., Clifford L. 2009. A comparison of three computational modelling methods for the prediction of virological response to the combination of HIV therapy, *Artificial Intelligence in Medicine* 47,63—74, *http://www.intl.elsevierhealth.com/journals/aim*

Goodarzi and Freitas 2010. The Application of Genetic Function Approximation (GFA) and artificial neural networks. *European Journal of Medicinal Chemistry,* Vol 45, no 4,, ISBN, 1352-1358

Gyenesei A. 2000. Mining weighted association rules for fuzzy quantitative items, *Proceedings of PKDD Conference Lyon, France*, 416–423.

Harper, P.R. and Shahani, A.K. 2002. A decision support system for the care of HIV and AIDS patients in India. *European Journal of Operational Reasearch.*

Hassanien A.E. 2003. Intelligent data analysis of breast cancer based on rough set theory, *International Journal on Artificial Intelligence Tools* 12:4, 479–493.

Hatzakis and Tsoukas, 2001. Neural Networks for the assessment of HIV immunopathology. Immune Deficiency Treatment Center, McGill University.*US National Library of MedicineNational Institutes of Health.*

Hatzakis, G. E. and Tsoukas, C. M. 2002 Neural networks morbidity and mortality modeling during loss of HIV T-cell homeostasis. *McGill University Health Centre, Montreal General Hospital, Montreal, Quebec, Canada. Proceedings / AMIA*

Hatzakis, G.E. Mathur, M. Gilbert, L. Maniar, L.K. Panos, G. Patel, A. Wanchu, A. and Tsoukas, C.M. 2005 Neural Network-longitudinal assessment of the Electronic Anti-Retroviral Therapy, *annual symposium, proceedings archive*.

Hebb, D.O. 1949. Organization of behaviour, Wiley, New York.

Herman R.A, Noormohamed S, Hirankarn S, Shelton MJ, Huang E, Morse GD, Hewitt RG, Stapleton JT, 1999. Comparism of Neural Networks with five traditional methods for predicting creatinine clearance in patients with HIV infection. *College of Pharmacy, University of Iowa, Iowa City, USA.*

Holland, J.H., 1975. Adaption in natural and Artificial Systems. The University of Michigan press, Ann Arbor.

Hopfield, J.J. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Science. USA*, 79: 2554-2558.

Hopfield, J.J. and Tank, D.W. 1986. Computing with neural circuits: a model science, 233: 625-633.

Hornik, K. Stinchcombe, M and White, H., 1989. Multilayer feed forward networks are universal approximators. *Neural Network 2,* 359-366.

Insurge J. 2007. A Pattern Classification of Back-Propagation Algorithm Using Exclusive Connecting Network.

IslamQ, 1988. A framework of general factors for infectious disease emergence. NCBI.

Jacobs, R.A. 1988. Increased rate of convergence through learning rate adaptations Neural Networks, 1:295-307.

Jeff H. 2005. Programming Neural Networks in java Artificial Intelligence Article: *Understanding Neural Network.*

Jiawei H. and Micheline K. 2001. Data Mining Concepts and Techniques, 1-10.

John W. 2005. Encyclopedia of Data warehousing and Mining, Vol 2, *Montclair State University, U.S.A*, 1382

John, P. A. Philip, G. M, James J. G.,Richard A. K., 1997. Use of Neural Networks to model complex immunogenetic associations on the progression of HIV infection. *European Journal of Clinical Investigation.*

Johnson, S. 1997. Using Neural Networks in Finance: A Practical Approach. *Unpublished B.Sc. Thesis. Dept. of Computer Science, South Bank University.*

Kim G, Kim Y, Lim H and Kim H 2013. Their work is a reiteration of the work by Thompson et al. *An MLP-based feature subset*.

Kirkpatrick, S., Gelatt, C.D. Jr and Vecchi, M.P. 1983. Optimization by simulated annealing. Science, 220: 671-680.

Kohonen, T. 1982. Self-Organized formation of topologically correct feature maps. *Biological cybernetics*. 43:59-69.

Kou, Y., Lu, C.-T., Sirwongwattana, S., and Huang, Y.-P. 2004. Survey of fraud detection Techniques. In Networking, Sensing and Control, *IEEE International Conference on Networking, Sensing and Control*. 749-754.

Kuok C. and Fu, H. Wong 1998. Mining fuzzy association rules in databases, *ACM SIGMOD Record* 27, 41–46; T.-P. Hong, C.-S. Kuo, S.-C. Chi (2001).

Kurt T, 1995. An Introduction to Data Mining. Retrieved on May 6th, 2013 *http://thearling.com/text/dmwhite/dnwhite.htm*

Kwak N,K, and Lee C 1997. A neural network application to classification of health status of HIV/AIDS patients :87-97.

Ladner R., F.E. Petry, M.A. Cobb 2003. Fuzzy set approaches to spatial data mining of association rules, Transactions in GIS 7, 123–138.

Lamers, S. L., 2008. Demonstrate the use of artificial neural networks trained via evolutionary computation to predict : 291–300.

Larder B, Wang D, Revell A. 2008. Application of artificial neural networks for decision support in medicine, *Methods Mol*. Biol. 458: 123-36

Laurikkalaj, Joshua M., 1998. A genetic-based machine learning system to discover the diagnostic rules for female urinary incontinence, no 3, 217-228.

Lavrac et al. 2007. Data Mining and Visualization for decision support, *Journal of Biomedical Informatics* 438–447.

Lee C.W. and Park J.A. 2000. The Application of Neural Networks to classify and predict the symptomatic status of HIV/AIDS patients. *Information & Management* 38, 23238

Lee C.W. and Park J.A. 2000. Assessment of HIV/AIDS-related health performance using an artificial neural network. *Information & Management*, vol. 38, 231-23

Lee W.J., S.-J. Lee 2004. Discovery of fuzzy temporal association rules, *IEEE Transactions on Systems, Man, and Cybernetics;*

Leke Betechuoh B., Marwala, T. and Tettey, T. 2007. Neural networks in an inverse configuration for the adaptive control of HIV status, *International Journal of Computational Intelligence individuals*. ISSN 0973-1873 Vol.3, No. 1.. 11-15.

Lilly M. F., Balalubramanle, P. 2009. : Support Implementation of Back-propagation Algorithm in HIV/AIDS REGIMEN.International *Journal of Reviews in computing*,

Loannidis JP, Mokrysz C, Nosek BA, Flint J, Robinson ES, Munafo`, 1997. Explores the application of neural networks in the study of. HIV/AIDS.

Lu S.F., H. Hu, F. Li 2001. Mining weighted association rules, Intelligent Data Analysis, 5, 211–225.

Manaswin, P., Ranjit K. S. 2011. *International Journal of Computer Applications in Engineering Sciences (IJCAES),* Vol. 1, Issue 1.

Marko, D. A, Andrej K. E. 2007. Data mining and visualization for decision support and modeling of public health-care resources, *Journal of Biomedical Informatics 40* 438–447 Pages 438-447

McCulloch, W.S. and Pitts, W.A. 1943. A logical calculus of the ideas immanent in neural nets. Bull. Math. Biophysics, 5: 115-133.

Michael J.A. Berry and Gordon S. Linoff. 2004. Data Mining Techniques, Second Edition.*Wiley Computer* Publishing, *Hoboken, NJ.*

Michalowski W., S. Rubin, R. Slowinski, S. Wilk 2003. Mobile clinical support system for pediatric emergencies, Decision Support Systems 36:2, 161–176.

Microsoft Visual C# ,2008 Step by Step.

Minsky, M. 1967. Computation: Finite and Infinite Machines. Prentice Hall, Englewood Cliffs, N.J.

Minsky. M and Papert S. 1969. Perceptron: Neural Networks: A Systematic introduction, Englewood Cliffs, N.J, 312

Minsky M., 1967. Computation: Finite and Infinite Machines, *Prentice Hall, Englewood Cliffs, N.J.*

Mohaghegh, S., 2003. Essential Components of an Integrated Data Mining Tool for the Oil and Gas Industry,With an Example Application in the DJ Basin. Paper SPE 84441, Proceedings, *SPE Annual Conference and Exhibition*

Myoung K., Ingoo H., 2003. The Discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms, *Expert systems with applications*, 631-646.

Nada, L. A., Marko, B. A., Aleksander, P. C., Bojan, C.,Harper, P.R., Shahani, A.K. 2003. A decision support system for the care of HIV and AIDS patients in India, *European Journal of Operational Research* 147 187–197.

Nelwamondo J., Msiza, F. V. and T. Marwalapared, 2007. Worked on a comparison of neural network with basic neural network-genetic algorithm estimation method.

Neuro Dimension, Inc (25th December, 2002), Neuro Solutions: What is a neural network? *http://www.nd.com /neurosolutions /products/us/what is NN.html.*

Nicole M. Dutch, 2006. A survey to estimate the risk of acquiring HIV using demographic and medical history information.

Osofisan, A.O.,Adeyemo. O. O., Sawyer, B. A., and Oluwafemi E. 2011. Prediction of Kidney Failure Using Artificial Neural Network, *European Journal of Scientific Research,* Vol 61,No 4, 487-492, 2011.

Osofisan, A. O. 2011. Data Dust to Data Gold, *Inaugural Lecture, University of Ibadan, Department of Computer Science.*

Otine C. D., Samuel B. K., and Lena T. 2010. Dimensional Modeling of HIV Data Using Open Source. *World Academy of Science, Engineering and Technology* (63)2010.

Otine D.O. 2012. HIV Patient Monitoring Framework Through Knowledge Engineering, Blekinge Institute of technology, Doctoral Dissertation, Series, No.2012.12.

Pacific Northwest National Laboratory (PNNL), 2001. Neural Networks: What are Neural Networks, http://www.emsl.pnl.gov:2080/proj/neuron/neural/what.html.4th November.

Pasomsub 2010. Application of artificial neural networks for the phenotypic drug resistance predictionJpn J Infect Dis 2010 Mar; 63(2):87-94.*Patients Analysis Using Clustering Technique. Asian Journal of Information Technology* (5) 4, 413-417.

Pasomsub E, Sukasem C, Sungkanuparph S, Kijsirikul B, Chantratita W, 2010. The application of artificial neural networks for phenotypic drug resistance prediction: evaluation and comparison with other interpretation systems. Mar; 63(2):87-94.

Paul J. Deitel and Harvey M. Deitel 2010. C# 2010 for Programmers (4th Edition) (Deitel Developer Series).

Pawlak Z. 1982. Rough set, *International Journal of Computer and Information Sciences*.

Pawlak Z., R. Sowinski 1994. Decision analysis using rough sets, *International Transactions of Operational Research* 1:1, 107–114.

Pearl J. 1988. Probabilistic reasoning in intelligent systems, *Networks of Plausible Inference San Mateo, CA: Morgan Kaufmann.*

Philipp K. 1994. Combining Genetic Algorithmsand Neural Networks: *The Encoding Problem A Thesis the University of Tennessee, Knoxville.*

Principe, J.C. Eliano, N.R. and Lefebore, W.C. 2000. Neural Networks and adaptive systems: fundamentals through Simulation, John Wiley and Sons, Inc. *Proceedings of National System Theory and Regional Analysis Conference, Baotou (In Chinese); K.-G. Zhao.*

Purwanto, C., Eswaran, and Logeswaran, R. 2011. The Application of Adaptive Neuro-fuzzy inference system for HIV/AIDS time-series prediction. *Communications in Computer and Information Science* Vol. 253, 1-13.

Resino S, Seoane JA, Bellón JM, Dorado J, Martin-Sanchez F, Alvarez E, Cosín J, López JC, Lopéz G, Miralles P, Berenguer. 2011. Development of an artificial neural network to predict significant fibrosis in HIV/Hepatitis C (HCV) *Journal of Infection*, Vol 62, issue 1,77-86.

Rodriguez W., M. Last, A. Kandel, H. Bunke 2001. Geometric approach to data mining, International Journal of Image & Graphics 2.

Rosenblatt, F. 1960. On the Convergence of reinforcement procedures in simple perceptron’s. Technical report VG1196-GA Cornel Aeronautical Laboratory.

Ruben D. Canlas Jr., 2005. Data Mining in Health care Current Applications and Issues. Applications: *M.Sc. Thesis. Information technology, Carnegie Mellon University, Australia.*

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. 1986. Learning internal representations by back-propagation errors. Nature, 323: 533-536.

Russell G.J., and Petersen, A. 2000. Analysis of cross category dependence in market basket selection, Journal of Retailing.

Sardari S, Sardari D. 2002. Applications of artificial neural network in AIDS research and therapy. Curr Pharm Des., 8(8):659-70.

Shahani N, Gaudio, A. C., Korolkovas, A. & Takahata, Y, 1994. *A framework for operational modelling for diseases. European Journal of Operational Research.*

Shillabeer, A. and Roddick, J.F. 2007. Establishing a Lineage for Medical Knowledge Discovery. In Proc. 6th Australasian Data Mining Conference

Sibanda Wilbert and Philip Pretorius2012. The Differential Effects of Demographic Characteristics on HIV Prevalence in South Africa. 2012-06-26

Stanfill C., D.L. and Waltz 1986. Toward memory-based reasoning, Communications of the ACM 29:12, Dec, 1213–1223; D. Aha, D. Kibler, M. Albert (1991). *Instance-based learning algorithms, Machine*

Stephen G.H., 1994. Applied Economic Forecasting Techniques. Harvester Wheatsheaf.Susanna L. L., Marco S., Michael S. McGrath, and Gary B. Fogel. (2008), Prediction of R5, X4, and R5X4 HIV-1 Coreceptor Usage with Evolved Neural Networks, *IEEE/ACM Trans Computer Biol Bioinform. 2008 Apr-Jun; 5(2): 291–300. doi: 10.1109/TCBB.2007.1074*.

Taryn Nicole Ho, T.M. Johan 2006. Predicting HIV Status using Neural Networks and Demographic factors.

Thangavel, K., Jaganathan, P.P. and Easmi, P.O 2006. Data Mining Approach to Cervical Cancer

Tim and Marwala 2001. Used computational intelligence methods for risk assessment of HIV. *www.researchgate.net/.../226325492*.

Tocher, K.D., 1963. The Art of Simulation. English Universities Press.

Tsoi, A.C. S. Zhang, M. Hagenbuchner 2005. Pattern discovery on Australian medical claims data – A systematic approach, *IEEE Transactions on Knowledge & Data Engineering*.

Tsumoto S. 2004. Mining diagnostic rules from clinical databases using rough sets and medical diagnostic model, Information Sciences 162:2, 65–80.

UNAIDS questions and answers basic facts about HIV/AIDS epidemic and its impact,*http:www.unaids.org/resources.asp*

Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth.1996. *Data Mining to Knowledge Discovery in Data Base.* http://www.aaai.org/

Vararuk, I. Petrounias, V. Kodogiannis, 2008. "Data mining techniques for HIV/AIDS data management in Thailand", *Journal of Enterprise Information Management*, Vol. 21: 1,.52 – 70.

Wei-Pinchang, Der-MingLiou, 2009. Comparison of Three Data Mining Techniques in the Analysis of Breast Cancer Data.

Weiss S.M., B.F. White, C.V. Apte, F.J. Damerau 2000. Lightweight document matching for help-desk applications, *IEEE Expert Intelligent Systems &their Applications*.

Wilbert S. and Philip Artificial Neural Networks- A Review of Applications of Neural Networks in the Modeling of HIV Epidemic. *International Journal of Computer Applications* 44(16):1-4

Wilson A., Thabane L., Holbrook A 2003."Application of data mining techniques in pharmacovigilance". *British Journal of Clinical Pharmacology*. (57) 2, 127-134.

Witold P. 2011. Wiley Interdisciplinary John Wiley & sons, Vol. 1, Issue1, 1.

Wong, W.K., Moore, A., Cooper, G. and Wagner, M. 2005. What's Strange About Recent Events (WSARE): An Algorithm for the Early Detection of Disease Outbreaks? *Journal of Machine Learning Research. 6,*

Wright, 1998. Data Mining Techniques and Applications in health care.

Yan, X., S. Zhang, C. Zhangt 2007. Data structures for association rule discovery, *Applied Artificial Intelligence.*

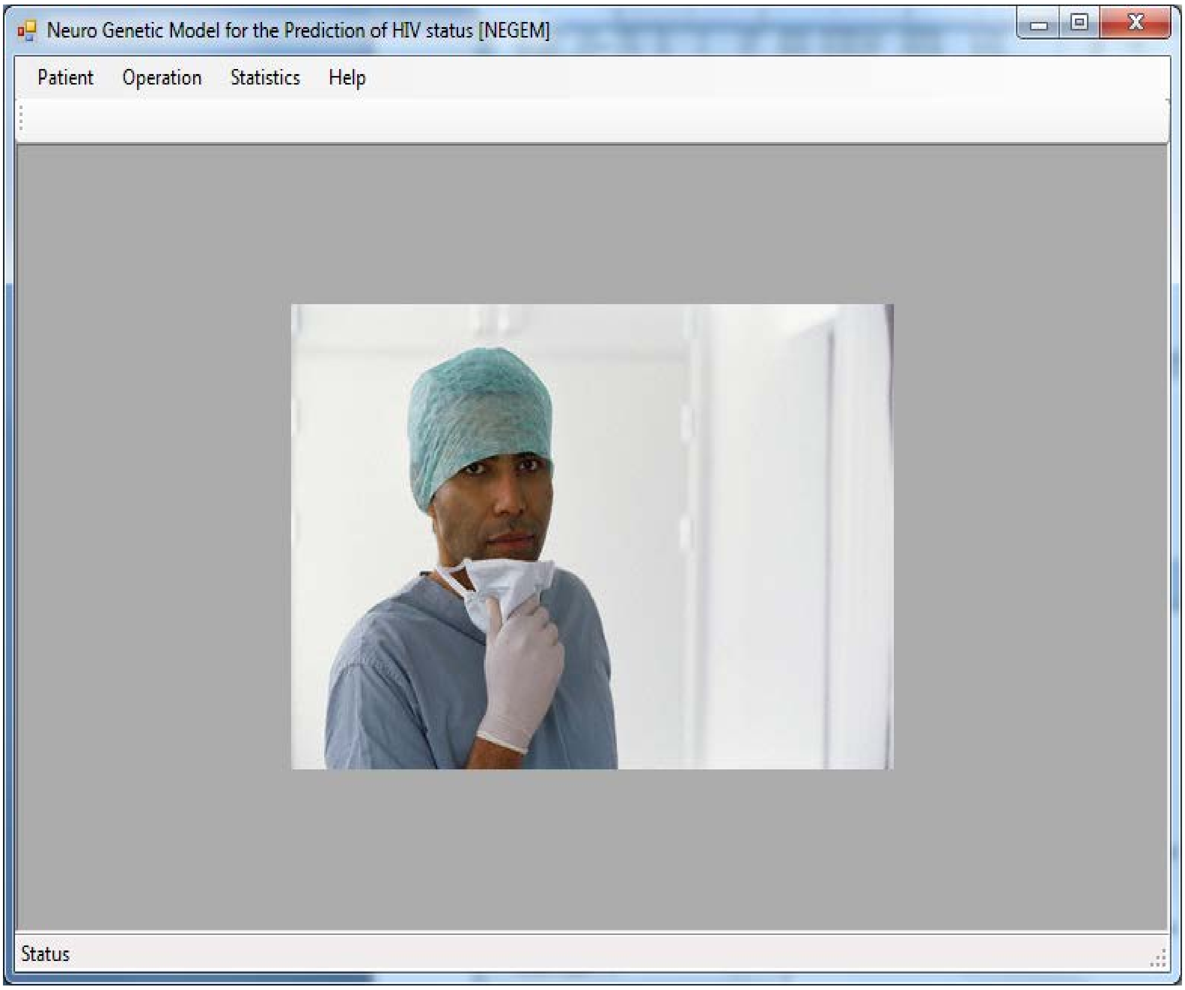
Yu-Dong Cai, Xiao-Jun Liu, Xue-Biao Xu, Kuo-Chen Chou, 1998. Artificial neural network method for predicting HIV protease cleavage sites in protein. *Journal of Protein Chemistry - J PROTEIN CHEM*, vol. 17, no. 7. 607-615, 1998.

Zadeh L.A. 1965. Fuzzy sets, Information and Control 8, 338–356.

Zhao K. G. 1989. Set pair analysis – a new concept and new systematic approach.

Zurada J., W. Karwowski, W. Marras 2004. Classification of jobs with risk of low back disorders by applying data mining techniques, Occupational Ergonomics.

APPENDIX


Neuro Genetic Model for the Prediction of HIV status [NEGEM]
Patient
Operation
Statistics
Help
Status

The transformed, Randomized Exemplars

PID	Symptom 1	Symptom 2	Symptom 3	Symptom 4	Symptom 5	Symptom 6	Symptom 7	Output
5015	0	0	1	0	0	0	0	0
5016	0	0	0	0	0	0	0	0
5017	0	0	0	0	0	0	0	0
5018	1	1	1	1	0	0	0	1
5019	0	0	0	1	0	0	0	0
5020	1	0	1	0	1	0	0	1
5021	0	1	0	0	0	0	0	0
5022	1	0	1	1	1	0	0	1
5023	0	0	0	0	0	0	0	0
5024	1	1	1	0	0	0	0	1
5025	1	0	0	1	1	1	0	1
5026	1	0	1	1	0	0	0	1
5027	0	1	1	1	0	0	0	1
5028	0	0	0	1	0	0	0	0
5029	0	0	1	0	0	0	0	0
5030	0	0	0	0	0	0	0	0
5031	0	0	1	0	0	0	0	0
5032	1	0	1	1	0	1	0	1
5033	0	0	0	0	0	0	0	0
5034	0	1	0	0	0	0	0	0
5035	0	0	0	0	0	0	0	0
5036	1	1	1	0	0	0	0	1
5037	1	1	0	1	0	0	0	1
5038	1	1	1	0	0	1	0	1
5039	1	0	0	1	0	1	0	1
5040	1	1	0	1	0	1	0	1
5041	1	1	1	0	1	0	0	1
5042	0	1	1	1	1	0	0	1
5043	0	0	1	1	1	0	0	1
5044	0	1	1	0	1	1	0	1
5045	0	0	1	1	1	0	0	1
5046	1	0	1	1	0	1	0	1
5047	1	0	0	1	1	1	0	1
5048	1	0	1	1	1	0	0	1
5049	0	0	0	0	1	0	0	0
5050	1	1	0	1	1	0	0	1
5051	0	1	0	0	0	0	0	0
5052	0	0	0	0	0	0	0	0
5053	1	0	0	0	0	0	0	0
5054	1	1	0	1	0	0	0	1
5055	0	0	1	1	0	1	0	1

5056	0	0	0	1	0	0	0	0
5057	1	0	0	1	1	1	0	1
5058	0	0	1	1	1	0	0	1
5059	0	0	1	1	1	0	0	1
5060	1	1	0	0	1	1	0	1
5061	1	1	0	1	1	0	0	1
5062	1	0	1	1	1	0	0	1
5063	1	1	1	0	0	0	0	1
5064	0	0	0	0	0	0	0	0
5065	1	1	0	1	0	0	0	1
5066	0	1	1	1	0	0	0	1
5067	1	0	1	0	1	0	0	1
5068	1	1	1	1	0	0	0	1
5069	1	1	0	0	1	0	0	1
5070	1	1	1	0	1	0	0	1
5071	1	1	1	0	1	0	0	1
5072	1	0	0	0	0	0	0	0
5073	0	1	1	1	0	0	0	1
5074	0	0	0	0	0	0	0	0
5075	1	1	0	1	1	0	0	1
5076	0	1	0	0	0	0	0	0
5077	1	0	0	1	0	1	0	1
5078	0	0	0	0	0	0	0	0
5079	1	0	0	1	1	1	0	1
5080	0	0	1	0	0	0	0	0
5081	0	0	0	0	1	0	0	0
5082	1	1	0	0	1	0	0	1
5083	1	1	1	0	1	0	0	1
5084	0	1	1	1	1	0	0	1
5085	0	0	0	0	0	0	0	0
5086	0	1	0	0	0	0	0	0
5087	0	0	0	0	0	0	0	0
5088	0	1	1	1	1	0	0	1
5089	0	1	1	1	0	0	0	1
5090	1	0	1	0	1	0	0	1
5091	1	1	0	1	0	0	0	1
5092	0	0	0	0	0	0	0	0
5093	0	1	1	0	1	0	0	1
5094	1	0	1	1	1	0	0	1
5095	1	0	1	1	0	0	0	1
5096	0	0	0	1	0	0	0	0
5097	1	1	0	1	1	0	0	1
5098	1	1	1	0	0	0	0	1
5099	0	1	1	1	0	0	0	1

5579	1	1	1	1	0	0	0	1
5580	0	1	0	1	1	1	0	1
5581	0	0	0	0	0	0	0	0
5582	1	1	0	1	0	0	0	1
5583	0	0	0	0	0	0	0	0
5584	0	0	0	0	0	0	0	0
5585	0	0	1	1	1	0	0	1
5586	1	1	0	1	0	0	0	1
5587	0	0	0	0	0	0	0	0
5588	0	0	0	1	0	0	0	0
5589	0	0	0	0	0	0	0	0
5590	1	1	1	0	0	0	0	1
5591	0	0	0	0	0	0	0	0
5592	1	0	0	0	0	0	0	0
5593	0	1	0	0	0	0	0	0
5594	1	1	1	0	0	0	0	1
5595	0	0	1	1	0	1	0	1
5596	1	1	0	0	1	0	0	1
5597	1	0	1	1	1	0	0	1
5598	0	1	1	1	0	0	0	1
5599	1	1	1	1	0	0	0	1
5600	1	0	1	1	0	0	0	1
5601	0	0	0	0	0	0	0	0
5602	1	0	0	0	0	0	0	0
5603	0	0	0	0	0	0	0	0
5604	1	0	0	0	0	0	0	0
5605	0	0	0	0	0	0	0	0
5606	1	0	1	1	0	1	0	1
5607	0	1	1	1	1	0	0	1
5608	1	1	0	1	1	0	0	1
5609	0	0	0	0	0	0	0	0
5610	0	0	0	0	0	1	0	0
5611	0	1	1	1	0	0	0	1
5612	1	0	1	1	1	0	0	1
5613	0	0	0	0	0	0	0	0
5614	0	1	1	1	1	0	0	1
5615	1	1	1	0	1	0	0	1
5616	1	1	0	1	1	0	0	1
5617	1	0	0	0	0	0	0	0
5618	1	0	1	1	0	0	0	1
5619	1	1	1	0	1	0	0	1
5620	1	1	1	1	0	0	0	1
5621	1	1	0	1	1	0	0	1
5622	1	1	0	1	0	0	0	1

5623	0	0	0	0	0	0	0	0
5624	1	0	0	0	0	0	0	0
5625	0	0	0	0	0	0	0	0
5626	0	0	0	0	0	0	0	0
5627	1	1	1	0	1	0	0	1
5628	0	0	0	0	1	0	0	0
5629	1	0	0	0	0	0	0	0
5630	1	1	0	0	1	0	0	1
5631	1	0	1	1	1	0	0	1
5632	1	0	0	1	0	1	0	1
5633	1	0	1	0	1	0	0	1
5634	0	1	1	0	1	0	0	1
5635	1	0	0	0	0	0	0	0
5636	1	0	0	0	0	0	0	0
5637	0	1	1	1	0	0	0	1
5638	0	0	0	0	0	0	0	0
5639	1	0	0	0	0	0	0	0
5640	1	0	1	1	1	0	0	1
5641	1	1	0	1	0	0	0	1
5642	1	0	0	0	0	0	0	0
5643	0	1	1	1	0	0	0	1
5644	1	0	1	1	1	0	0	1
5645	0	1	1	0	1	1	0	1
5646	0	0	0	0	0	0	0	0
5647	1	1	0	1	1	0	0	1
5648	0	1	1	1	1	0	0	1
5649	1	0	0	0	0	0	0	0
5650	0	1	1	0	1	1	0	1
5651	0	1	0	0	0	0	0	0
5652	0	0	0	0	0	0	0	0
5653	1	1	1	0	1	0	0	1
5654	0	0	0	0	0	0	0	0
5655	0	0	0	0	0	0	0	0
5656	1	1	1	0	1	0	0	1
5657	1	1	1	1	0	0	0	1
5658	1	1	0	1	0	0	0	1
5659	1	0	1	1	1	0	0	1
5660	1	1	1	0	1	0	0	1
5661	0	1	1	1	0	0	0	1
5662	1	0	1	1	0	0	0	1
5663	1	1	1	1	0	0	0	1
5664	0	1	1	0	1	1	0	1
5665	1	1	1	1	0	0	0	1
5666	1	1	0	0	1	0	0	1

5667	0	0	1	0	0	0	0	0
5668	1	1	1	1	0	0	0	1
5669	0	0	0	0	0	0	0	0
5670	1	0	0	0	0	0	0	0
5671	0	0	0	0	0	0	0	0
5672	0	1	0	0	0	0	0	0
5673	1	1	0	1	1	0	0	1
5674	0	0	0	1	0	0	0	0
5675	0	1	1	1	0	0	0	1
5676	0	1	0	0	0	0	0	0
5677	0	1	1	0	1	1	0	1
5678	0	0	1	1	1	0	0	1
5679	0	0	0	0	0	0	0	0
5680	0	0	0	1	0	0	0	0
5681	0	1	1	1	1	0	0	1
5682	1	0	0	0	0	0	0	0
5683	1	1	0	1	1	0	0	1
5684	0	1	1	1	1	0	0	1
5685	0	0	0	0	0	0	0	0
5686	0	0	1	0	0	0	0	0
5687	0	0	1	1	0	1	0	1
5688	0	0	0	0	1	0	0	0
5689	0	1	1	0	1	0	0	1
5690	1	0	1	1	0	1	0	1
5691	0	0	0	0	0	0	0	0
5692	0	1	0	1	1	1	0	1
5693	0	1	1	1	1	0	0	1
5694	0	1	0	0	0	0	0	0
5695	1	1	1	0	1	0	0	1
5696	1	1	1	0	1	0	0	1
5697	1	0	0	0	0	0	0	0
5698	1	0	0	1	1	1	0	1
5699	1	0	1	1	1	0	0	1
5700	0	0	0	0	0	0	0	0
5701	0	0	1	0	1	1	0	1
5702	0	0	0	1	0	0	0	0
5703	0	0	0	0	0	0	0	0
5704	1	0	0	0	0	0	0	0
5705	0	1	1	1	1	0	0	1
5706	0	1	1	1	0	0	0	1
5707	1	0	1	0	1	1	0	1
5708	1	1	1	0	1	0	0	1
5709	0	0	0	0	0	0	0	0
5710	1	0	0	1	0	1	0	1

5711	1	1	0	0	1	1	0	1
5712	1	1	0	1	0	0	0	1
5713	1	0	1	1	1	0	0	1
5714	1	1	0	1	1	0	0	1
5715	0	0	0	0	0	0	0	0
5716	0	0	0	0	0	0	0	0
5717	0	1	0	1	1	1	0	1
5718	0	0	0	0	0	0	0	0
5719	0	0	0	0	0	0	0	0
5720	0	0	0	0	0	0	0	0
5721	1	1	0	0	1	0	0	1
5722	1	1	1	1	0	0	0	1
5723	1	0	1	1	1	0	0	1
5724	0	1	1	0	1	1	0	1
5725	0	1	0	0	1	1	0	1
5726	0	0	1	1	0	1	0	1
5727	1	1	0	0	1	1	0	1
5728	1	0	1	0	1	0	0	1
5729	0	0	1	0	0	0	0	0
5730	1	0	0	0	0	0	0	0
5731	1	1	0	1	1	0	0	1
5732	0	1	1	1	0	1	0	1
5733	0	0	1	0	0	0	0	0
5734	0	1	0	0	0	0	0	0
5735	0	1	0	0	0	0	0	0
5736	0	0	0	0	0	0	0	0
5737	1	1	1	0	0	0	0	1
5738	0	0	0	0	0	0	0	0
5739	1	0	0	0	0	0	0	0
5740	0	0	0	0	0	0	0	0
5741	0	1	1	1	0	0	0	1
5742	0	0	1	0	0	0	0	0
5743	1	0	0	0	0	0	0	0
5744	1	1	1	0	0	1	0	1
5745	1	1	1	0	0	0	0	1
5746	1	1	0	1	0	0	0	1
5747	0	0	0	0	0	0	0	0
5748	1	1	1	1	0	0	0	1
5749	0	1	1	1	0	0	0	1
5750	0	0	1	1	1	0	0	1
5751	0	1	1	1	0	1	0	1
5752	0	1	0	0	1	1	0	1
5753	1	1	1	1	0	0	0	1
5754	0	1	1	1	0	0	0	1

5755	0	0	0	0	0	0	0	0
5756	0	0	1	0	0	0	0	0
5757	0	1	0	0	1	1	0	1
5758	0	1	1	0	1	1	0	1
5759	1	0	1	1	0	0	0	1
5760	0	0	1	0	0	0	0	0
5761	0	0	0	1	0	0	0	0
5762	0	1	0	1	1	1	0	1
5763	0	0	0	0	0	0	0	0
5764	1	1	0	1	1	0	0	1
5765	0	0	0	0	1	0	0	0
5766	1	0	0	0	0	0	0	0
5767	0	1	0	0	0	0	0	0
5768	0	0	0	0	0	0	0	0
5769	1	1	0	0	1	0	0	1
5770	0	1	1	0	1	1	0	1
5771	1	1	0	1	0	0	0	1
5772	0	0	1	1	1	1	0	1
5773	1	0	0	1	1	1	0	1
5774	0	0	0	1	0	0	0	0
5775	1	1	1	0	1	0	0	1
5776	0	0	0	0	1	0	0	0
5777	1	1	1	0	1	0	0	1
5778	1	0	1	1	0	1	0	1
5779	0	0	0	0	0	0	0	0
5780	1	0	0	0	0	0	0	0
5781	0	0	0	0	0	0	0	0
5782	1	1	0	1	1	0	0	1
5783	0	1	1	1	1	0	0	1
5784	0	1	1	1	1	0	0	1
5785	1	0	0	1	1	1	0	1
5786	0	0	0	0	0	0	0	0
5787	0	1	1	1	1	0	0	1
5788	1	1	0	1	0	0	0	1
5789	0	0	0	0	0	0	0	0
5790	0	0	0	0	0	0	0	0
5791	0	1	1	1	1	0	0	1
5792	0	0	0	0	0	0	0	0
5793	1	0	1	1	1	0	0	1
5794	1	0	1	1	0	0	0	1
5795	1	1	0	1	0	0	0	1
5796	0	0	0	0	1	0	0	0
5797	1	0	1	1	1	0	0	1
5798	0	0	0	0	0	0	0	0

5799	0	0	0	0	0	0	0	0
5800	1	0	0	0	0	0	0	0
5801	1	1	1	0	1	0	0	1
5802	1	1	1	1	0	0	0	1
5803	0	0	0	0	0	0	0	0
5804	1	1	0	1	0	1	0	1
5805	1	1	1	0	0	0	0	1
5806	0	0	1	0	0	0	0	0
5807	1	0	1	1	0	0	0	1
5808	0	1	1	1	0	0	0	1
5809	0	0	1	0	0	0	0	0
5810	1	0	0	0	0	0	0	0
5811	0	0	0	0	0	0	0	0
5812	0	0	0	1	0	0	0	0
5813	1	1	0	1	0	0	0	1
5814	0	0	0	0	0	0	0	0
5815	0	0	0	0	0	0	0	0
5816	0	1	0	0	0	0	0	0
5817	1	1	1	0	0	1	0	1
5818	1	0	1	0	1	0	0	1
5819	1	1	0	0	1	0	0	1
5820	1	1	0	0	1	0	0	1
5821	0	0	0	0	0	0	0	0
5822	1	0	1	1	0	0	0	1
5823	0	1	1	0	1	1	0	1
5824	0	0	1	1	1	0	0	1
5825	0	1	1	1	0	0	0	1
5826	0	1	1	1	1	0	0	1
5827	0	0	0	0	0	1	0	0
5828	0	0	0	0	0	0	0	0
5829	0	1	0	0	0	0	0	0
5830	1	0	1	0	1	1	0	1
5831	1	1	1	1	0	0	0	1
5832	1	1	0	1	0	0	0	1
5833	1	0	0	1	1	0	0	1
5834	0	0	0	0	0	0	0	0
5835	0	0	0	0	0	0	0	0
5836	0	0	1	0	0	0	0	0
5837	1	0	0	0	0	0	0	0
5838	1	0	1	1	0	1	0	1
5839	1	0	1	0	1	0	0	1
5840	0	0	1	0	0	0	0	0
5841	0	0	0	0	0	0	0	0
5842	1	1	0	0	1	0	0	1

5843	1	0	1	0	1	0	0	1
5844	0	0	0	0	0	0	0	0
5845	1	0	0	1	1	0	0	1
5846	0	1	0	0	0	0	0	0
5847	1	0	0	0	0	0	0	0
5848	0	0	0	0	0	0	0	0
5849	0	0	0	0	0	0	0	0
5850	0	0	0	0	0	0	0	0
5851	1	1	0	0	0	1	0	1
5852	1	0	1	1	1	0	0	1
5853	1	0	1	1	0	0	0	1
5854	0	1	0	0	0	0	0	0
5855	0	0	1	1	1	0	0	1
5856	0	1	1	0	1	0	0	1
5857	1	1	1	0	1	0	0	1
5858	0	0	1	0	0	0	0	0
5859	0	0	1	1	1	0	0	1
5860	1	1	1	1	0	0	0	1
5861	0	1	1	0	1	0	0	1
5862	1	0	0	0	0	0	0	0
5863	0	1	0	1	0	1	0	1
5864	0	0	0	0	0	0	0	0
5865	0	0	0	0	0	0	0	0
5866	1	0	0	1	1	1	0	1
5867	0	1	0	1	1	0	0	1
5868	0	0	0	0	0	0	0	0
5869	0	1	0	0	0	0	0	0
5870	0	1	0	1	1	0	0	1
5871	1	1	1	0	1	0	0	1
5872	0	0	1	0	0	0	0	0
5873	0	1	0	1	1	1	0	1
5874	1	0	0	0	0	0	0	0
5875	1	0	1	0	1	0	0	1
5876	1	0	0	0	0	0	0	0
5877	0	0	1	0	0	0	0	0
5878	1	0	0	0	0	0	0	0
5879	1	0	1	0	1	0	0	1
5880	0	1	0	0	0	0	0	0
5881	0	1	1	1	1	0	0	1
5882	0	1	0	0	0	0	0	0
5883	0	0	1	0	0	0	0	0
5884	0	0	0	0	0	0	0	0
5885	0	1	0	0	0	0	0	0
5886	0	1	0	0	0	0	0	0

5887	0	0	1	0	0	0	0	0
5888	0	0	0	0	0	0	0	0
5889	0	0	0	1	0	0	0	0
5890	0	0	0	0	0	0	0	0
5891	0	0	0	0	0	0	0	0
5892	1	0	0	0	0	0	0	0
5893	1	0	1	0	1	0	0	1
5894	1	0	1	1	0	0	0	1
5895	0	0	0	0	0	0	0	0
5896	0	0	0	0	0	0	0	0
5897	0	1	1	0	0	1	0	1

NEGEM Pseudo Code

```
void function BestFit_Load
    {
       create a connection object con
       set the connection string to the connection stored in our config file
       open the connection if not opened
       create a command object to cammand our database
       set the command text to an sql to load all symptoms from the database
       Now if our command returns recordset. Iterate through each recordset and create a list of
          checkboxes that will be used to test the system
       while (rd.Read())
       {
         create a checkbox object
         set the chekbox text to the symptom
         set the value to the value of the symptom in our databse
       }

      close reader
                    close connection
    }

                   function void TestingUserControl_Load
    {
       Create object of input layer
                              Create object of hidden layer
                              Create object of output layer from sigmoid layer

                              Create a back propagation object and set the input and its hidden layer
                                     respectively
                              Create a back propagation network and pass in the input and hidden layer
                                    respectively
                              initialize our network object
                              set number of weight to 0 initially
                              foreach( connectiors in our network)
                              {
                                        numweights+=totalSynapseCount
                              }
                              Call function GetData to get our data from the database.
                              Set our threshold
                              Initialise our GA class with crossover rate, mutation rate, the weights,
                                      population size, generation size and the num of weights
                              Create a fitness function
                              Display the result of our ga function go in the textbox
                              Create an array of weights and fitness variable respectively
                              Pass in the weight and the fitness variable to the GetBest function of our GA
                              Display the result of this function
    }
                   function fitnessFunction
    {
       fitness = 0
       counter++
       set predicted_value=0, expected_value=0
                     SET NETWORK OUTPUT PASSING IN THE NETWORK AND THE WEIGHT
       call setNetworkWeights passing in weight and network
       create a temporary variable temp

          Get our input from the array item
          predicted_value = network.Run(input)[0]
          // The closest the output is to zero, the more fit it is.
    temp = 1 - predicted_value;
    fitness += 1 - predicted_value;
                    GET THE HIV OUTPUT FROM THE DICTONARY ENTRY CLASS
        expected_value=double.Parse(keys[2])
        print our expected and our predicted

        IF PREDICTED VALUE ABOVE threshold predicted value=1 else predicted value=0

        WHERE EXPECTED VALUE=1, dX=|Xe-Xp| i.e expected value-predicted value [False
                 Positive{FP} - False Negative {FN}]
         if (expected_value == 0 && predicted_value == 0)
             increase TN by 1;
         else if (expected_value == 0 && predicted_value == 1)
             increase FP by 1;
         else if (expected_value == 1 && predicted_value == 1)
             increase TP by 1
         else if (expected_value == 1 && predicted_value == 0)
             increase FN by 1
return our fitness;
```