CORE
Metadata, citation and similar papers at core.ac.uk
Provided by The Research Repository @ WVU (West Virginia University)

Graduate Theses, Dissertations, and Problem Reports

2006

Integration of statistical and neural network method for data analysis

Krishna Kumar Chavali
West Virginia University

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Recommended Citation

Chavali, Krishna Kumar, "Integration of statistical and neural network method for data analysis" (2006). *Graduate Theses, Dissertations, and Problem Reports*. 4219.
https://researchrepository.wvu.edu/etd/4219

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Integration of Statistical and Neural Network Method for Data Analysis

By

Krishna Kumar Chavali

Thesis submitted to

The College of Engineering and Mineral Resources

at West Virginia University in partial fulfillment

of the requirements for the degree of

Master of Science in Industrial Engineering

Committee

Dr. Rashpal Ahluwalia (Chair)

Dr. Robert Creese

Dr. Arun Ross

Department of Industrial and Management Systems Engineering

West Virginia University, Morgantown, West Virginia

2006

ABSTRACT

Integration of Statistical and Neural Network Method for Data Analysis

Krishna K Chavali

The issues that are to be considered while selecting data analysis methods are data type and expected solution type. Incorrect selection of either can lead to incomplete solution leading user taking an uninformed decision. The software developed for this thesis (SANE - Statistical And NEural network data analysis) is aimed at enabling a user to use advanced data analysis techniques to handle a given research issue. An effective navigation system is designed for user to select the analysis method by responding to a set of questions. SANE provides web based data analysis using statistical and neural networks methods. The statistical analysis module has methods for finding a relationship between variables, predicting group membership and finding group differences. The neural net module has back propagation and cascade correlation algorithms. Users can apply different methods on same dataset and compare the results. This software is implemented in ASP.NET2.0 with backend in SQL2005.

ACKNOWLEDGEMENTS

First and Foremost, I would like to thank Dr. Rashpal S. Ahluwalia for suggesting this thesis topic and and guiding me for the entire length of my Masters by providing resources, subjects and very essential feedback.

I am also very grateful to my friends who supported me and actually helped me in completion of this work. Without them, the work would have been practically impossible to complete and this document would never been possible.

I would also like to convey regards to my parents, my family and very good friends for being very patient with me and actually pushing me towards the completion of my Masters.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1. Objective

Data Analysis consists of two important components – Problem definition and analysis of solution and these two parts are dependent on one another. Method selection is just as important as the analysis itself because an improper selection will lead user to inconclusive results. The objective of this thesis is to develop an integrated set of Statistical And NEural network (SANE) tools for data analysis and to guide a novice user in selecting appropriate data analysis technique(s). The software will be designed in such a way that the user need not have extensive background in statistics or neural nets. Initially, the statistical methods will be limited to finding a relationship between independent and dependent variables, predicting group membership of a dataset, finding if the dataset is properly grouped, and determining the underlying structure of a dataset. The neural network algorithms will be limited to the back propagation algorithm and the cascade correlation algorithm.

1.2. Background

There are many commercial software tools available for statistical analysis such as SAS [22], Design Expert [7], Minitab [24] and Statistica [30]. These tools require programming skills and can be expensive for small businesses and students. The software tools that do not require much of the programming skills and are relatively user friendly are Microsoft Excel [20] and Lotus 123 [27]. However, these tools have to be installed on a local computer for the analyses to be performed and have limited analytic capability such as finding the degree of the relationship between two variables, checking if there is any significant clustering in the data (analysis of group differences), checking if one particular input data collection falls into one particular group (Group Membership) and analyzing if there is any underlying structure latent from the independent and the dependent variables. These analyses hold well only when the predicting dataset (Independent Dataset) lies with in the range of the dataset that is used for the analysis. There is a need for a method that lets the user forecast the result of a predictor dataset that fall outside the range of the analyzed dataset. For this reason, different forecasting techniques like Moving Average Method [21], where in the

forecast is based on the average response of the latest set of the variables on the time line, Exponential Smoothing [21] where the next period's forecast is made by considering the weighted values of the past outputs and Trend Line method, where a trend line is fitted using the existing dataset using the Least Squares Method [21] are used in forecasting. All forecasting methods need to:

- Decide on the factors that are needed to be used to forecast
- Decide on the function type (network type) that is used to derive the output, and
- Decide on the weights that are required for obtaining the output

Most of the forecasting techniques resemble multivariate analysis and the weights are decided by the users based on their experience. These methods fail to achieve good results unless the user is highly experienced and the data is fairly simple. To solve these problems, a method that involves a more complex network is needed and the network is designed to imitate the behavior of the neural biological systems and therefore these networks are called neural networks. As these parallel processed networks are simulated on a serial processing computer artificially, these systems are also called Artificial Neural Networks (ANN). ANNs have a more complicated structure and have a training strategy that adjusts the weights to suit the output and this leads to more accurate extrapolation of the data.

There are commercially available neural network software tools such as Predict [28] and Neural Net Pro [23]. These tools also require some level of expertise for their application. These tools are available only in the stand alone versions and that makes data interoperability difficult.

More importantly, the software tools for statistical analysis, forecasting and neural network implementation could be prohibitively costly and it is difficult for the user to decide which software should be used for a given research issue. If the incorrect data analysis tool is selected, the results may be confusing and incorrect. There is a need for a software tool that has the following properties:

- The software should have statistical and neural net analysis methods integrated into it

- The software should be application oriented
- The software should provide a navigation system that directs the user precisely to the type of analysis that can be performed for a given research question
- The software should be able to perform different analyses based on the same dataset to compare the results

This thesis presents a software tool called SANE to meet the above objective. SANE is a web based software tool. It can therefore be used on any platform that supports access to the World Wide Web. SANE has the capability to store the data and the results in a database. The user can try different analysis techniques on the same dataset. The following document explains on how different methods work and how they are implemented for the software.

1.3. Existing Tools (Weka Software)

"Weka" [33] is data analysis software which lets the user choose a data analysis method to be performed on his/ her dataset. The GUI (Graphical User Interface) and analysis part of this software is done in Java Virtual Machine and the back end can be a wide array of data sources like .csv (Comma Separated Value) files, Database files or Attribute Relationship File Format (ARFF) type documents. The software essentially provides a navigational system for the user to pick his dataset and a data analysis method and the required parameters all by him/ her self. Also, this software tool is installed locally on the computer so it offers better speed and security than web based systems; Better speed because it is installed locally and therefore does not have to make trips between client system and remote server and reliable because there is no data loss while communicating with a memory resident program.

1.3.1. Back end of Weka

Weka can read data into its system from various inputs like .csv files, Database files and .arff files. An .arff file is as mentioned in the section above is an Attribute Relation File Format document. This format can be considered as a flat file model of a relational database management system. The Class definition and the Attribute scope

is defined within the file itself and the instances of the so defined class are also present in the same file making it an independent entity which translates into easy portability. This document; since it is a plain text file, can be used across different operating systems completely justifying using Java for the development of software.

ARFF File format:

An .arff file is build with two distinct sections; a Header section and a Data Section. The header section contains all definitions, dataset domains and comments. The data section contains rows; can also be referred to as instances of dataset when observed from an Object perspective. Altogether, an .arff file looks like the one in the figure below:

However, the drawback of this system lies in the fact that it still very much depends on users understanding of data analysis and since it is a stand alone system, if there is any new analysis/ approach that is incorporated into the system will not reach all its intended audience. In the case of SANE, the system is capable of navigating the user to appropriate data analysis and data storage is based on SQL Server and data is organized in such a way that each column or each data attribute is stored in a row and this allows in protecting data integrity of the system.

```
% 1. Title: Iris Plants Database
  %
  % 2. Sources:
  %      (a) Creator: R.A. Fisher
  %      (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
  %      (c) Date: July, 1988
  %
  @RELATION iris

  @ATTRIBUTE sepallength  NUMERIC
  @ATTRIBUTE sepalwidth   NUMERIC
  @ATTRIBUTE petallength  NUMERIC
  @ATTRIBUTE petalwidth   NUMERIC
  @ATTRIBUTE class        {Iris-setosa,Iris-versicolor,Iris-virginica}
  @DATA
  5.1,3.5,1.4,0.2,Iris-setosa
  4.9,3.0,1.4,0.2,Iris-setosa
  4.7,3.2,1.3,0.2,Iris-setosa
  4.6,3.1,1.5,0.2,Iris-setosa
  5.0,3.6,1.4,0.2,Iris-setosa
  5.4,3.9,1.7,0.4,Iris-setosa
  4.6,3.4,1.4,0.3,Iris-setosa
  5.0,3.4,1.5,0.2,Iris-setosa
  4.4,2.9,1.4,0.2,Iris-setosa
  4.9,3.1,1.5,0.1,Iris-setosa
```

Figure 1.1 ARFF File snapshot

2. STATISTICAL METHODS

2.1 Introduction

The word “statistics” can be broadly defined as a range of techniques to analyze, interpret, display and make decisions based on data. When the data has to be analyzed, the possible research questions that are of interest are:

1. What is the *degree of relationship* between a set of outputs (dependent variables) and a set of inputs (independent variables)?
2. Can a given dataset be organized into groups? Can *group membership be predicted*?
3. Is there an underlying *group structure* for a given dataset?

For a given dataset, the above three questions will be addressed by the statistical module of SANE. The software tool will also assist the user in identifying the appropriate statistical analysis. Table 2.1 shows the notations used by the statistical module of SANE.

Table 2.1: Notations for Statistical Methods

Variable	Type	Description
NG	Integer	Number of Groups
NIV	Integer	Number of Independent Variables
TIV	Integer	Type of Independent Variables (1 - Discrete, 2 –Continuous, 3 – Mixed)
NDV	Integer	Number of Dependent Variables
NDR	Integer	Number of Data Records
TDV	Integer	Type of Dependent Variables (1 - Discrete, 2 –Continuous, 3 – Mixed)
$X_{n,i}$	Double	The independent variable dataset (n = 1 to NDR, i = 1, NIV)
$Y_{n,o}$	Double	The dependent variable dataset (n = 1 to NDR, o = 1, NDV)
R	Single	The coefficient of correlation
r-sq	Single	The coefficient of determination
SSE	Single	Sum of Squares of Error
SSR_o	Single	Sum of Squares of error due to model
SSTO	Single	Total Sums of Squares of Error
MSE	Single	Mean Square Error
MSR_o	Single	Mean Square Error explained by Model
β_{NIV+1}	Single	Coefficients of the Independent Variables
Df_o	Integer	Degrees of Freedom

2.1 Degree of Relationship

When the objective is to understand the effects of one variable over another variable, this method is used. This method examines 'how much' of the predictor (dependent) variable is explained by predicting (Independent) variable. This is achieved by computing total error within the system and the amount of error that is explained by the independent variable. Once the error explained is taken out of the system, the remaining unexplained error is tested for its randomness. If it satisfies the rules of randomness (Uniformly scattered around the model, no visible trends, error at a data point not based on error at a different data point) are satisfied, then the model is considered to be explained.

2.1.1 Analysis Description

The two main methods of determining the degree of relationship are the Bi-Variate r and Multiple-Regression. The Bi-Variate r method determines the relationship between an independent and a dependent variable. It computes the coefficient of correlation (r). The value of "r" lies between 0 and 1. It represents the portion of the error in the data set that is explained by the regression model. A value of 0 means that the error in the dataset is not at all explained by the model and a value of 1 means that the error in the model is completely explained by the model, that is, all data points are on the regression line. However, a lower value of r does not mean a bad fit. The error values have to be analyzed to check if the normality assumptions for error are satisfied, if they are not, then some of the variables might be missing from the equation. In that case multiple-regression analysis may be appropriate. Multiple-regression is an extension of Bi-Variate regression. It uses several independent variables (X) to predict one dependent variable (Y). In SANE, Bivariate analysis is considered as a special case of the multivariate analysis. The equations for Multivariate Analysis are as follows.

Step1 : Read Data into $X_{NDR,\ NIV}$ and $Y_{NDR,\ NDV}$

Step2 : Calculate values for β_0, β_1 …β_{NIV} using the following equation

$$[\beta]_{NIV} = \left([X]'_{NIV,NDR}[X]_{NIV,NDR}\right)^{-1}[X]'_{NIV,NDR}[Y] \qquad \text{------ (2.1)}$$

Step 3: Once the regression equation is calculated, the error explained from the regression equation is calculated using the following equations:

$$SSTO = \sum (Y_i - \overline{Y})^2 \quad \text{------ (2.2)}$$

$$SSR_o = \sum (E(Y_i) - \overline{Y})^2 \quad \text{------ (2.3)}$$

$$SSE = \sum (Y_i - E(Y_i))^2 \quad \text{------ (2.4)}$$

Where, $E(Y) = \beta_0 + \beta_1 X_1 + ... + \beta_{NIV} X_{NIV}$ ------ (2.5)

$$r = \sqrt{1 - SSE / SSTO} \quad \text{------ (2.6)}$$

$SSE = SSTO - \sum SSR_o$ ------ (2.7)

Step 4: Once the error values are calculated, find the value of *r* and perform Analysis of Variance.

ANOVA

Analysis Of Variance (ANOVA) is required to understand the source of the variance in the model. That is, every dataset will intuitively have some error in it. If a regression model is fitted to the dataset, it explains some of the error in the dataset and some of the error remains unexplained. In this process, the model fitted takes in some degrees of freedom of the data set. ANOVA calculates the amount of error explained per degree of freedom in the model and the error remained unexplained per degree of freedom left in the dataset. The procedure for one way ANOVA is as follows:

Step 1: Compute Mean Square Error (MSE). Dividing each of the *SSR* and the *SSE* with their corresponding degrees of freedom give the MS (Mean Square) values for each treatment.

$$MSR_o = \frac{SSR_o}{Df_o} \quad \text{------ (2.8)}$$

$$MSE = \frac{SSE}{Df - \sum Df_o - 1} \quad \text{------ (2.9)}$$

Step 2: Once the MSR is calculated, the F value is calculated and used to test the null hypothesis, which varies with the problem.

Example: The data set used for the multivariate regression analysis consists of two continuous independent variables (Books referred to and Attendance) and one continuous dependent variable (Grade obtained). The SAS output is as follows:

The GLM Procedure

Number of Observations Read 40
Number of Observations Used 40

The SAS System 02:12 Friday, February 24, 2006 4

The GLM Procedure

Dependent Variable: grade

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	3577.67047	1788.83524	9.06	0.0006
Error	37	7306.22953	197.46566		
Corrected Total	39	10883.90000			

R-Square	Coeff Var	Root MSE	grade Mean
0.328712	22.11211	14.05225	63.55000

Source	DF	Type I SS	Mean Square	F Value	Pr > F
attendance	1	2530.547141	2530.547141	12.82	0.0010
booksreferred	1	1047.123329	1047.123329	5.30	0.0270

Source	DF	Type III SS	Mean Square	F Value	Pr > F
attendance	1	944.157970	944.157970	4.78	0.0352
booksreferred	1	1047.123329	1047.123329	5.30	0.0270

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	37.37918520	7.74456446	4.83	<.0001
attendance	1.28347727	0.58696424	2.19	0.0352
booksreferred	4.03689261	1.75304907	2.30	0.0270

Table 2.2: Multi Variate Regression Analysis

This output shows the Analysis of Variance and in this case, the model explains 90% of the variance and the r^2 value is 0.33. F (2, 37, and 0.5) is 6.154 and since the F value obtained for this model is 9.06, this model can be considered true with 99.75% confidence.

2.2 Analysis of Groups

When the problem involves grouping of variables, it is important to know whether a given set of independent variable values fall into a certain group; Logistic Regression (LR) is often used to determine group membership. Or if subjects are randomly assigned to groups, the question usually is the extent to which reliable mean differences on dependent variables are associated with group membership. The methods used to predict group differences are one way ANOVA and Discriminant Function Analysis.

2.2.1 Prediction of Group Membership

Logistic regression is an analysis type in which the results are a probability value. The probability value gives the confidence that a given input row falls into a particular group. The independent variable can be of any type, continuous or discrete. The dependent variable has to be discrete. The number of dependent variables is the number of groups in this analysis for the result would be the probability that a given input row might fall into a given group. The steps for the logistic regression are as follows:

Step 1: Read Data.

Step 2: Build a regression model to fit the following equation using Least Squares Method

$$Y = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}} \qquad \text{------ (2.10)}$$

Step 3: Once the regression equation is built, the model can accept new input data values and predict the group membership of the data set.

2.2.2 Significance of Group Differences

DFA is used for assessing significance of group differences. The independent variable has to be discrete variable and the dependent variable can be either continuous or discrete. DFA has the groups as its independent variable and the factors that determine the group membership as the dependent variables.

Step 1: Group difference in the subjects is based on the different levels of the independent variables. The independent variables should follow a crossed design.

Step 2: Since there are a multiple number of dependent variables, there will be a matrix of response values within each group. These are calculated using equation 2.1

Step 3: Matrices of difference responses are formed by subtracting each response from the appropriate mean. The matrix of differences is squared. (Equations 2.2- 2.5)

Step 4: A Sum of Squares Matrix (SSM) is formed when the squared matrices are summed and the determinants of the various SSMs are found and the ratios between them provide the tests of hypotheses.

The ratio of each *MSR* with the *MSE* will test the null hypothesis for each treatment. The null hypothesis is that all the means are equal and the alternate hypothesis is that at least two means are not equal. If any ratio is greater than F (NG, NDR-1) then the null hypothesis is rejected.

2.3 Analysis of Structure

Analysis of structure is concerned with the latent structure underlying the variables. Factor Analysis and Structural Equation Modeling are performed under this category. Factor Analysis is used when there are hypotheses about the underlying a structure. Structural Equation Modeling combines factor analysis, canonical correlation and multiple-regression. The response variable may be dependant on few underlying factors, or it might contain many independent and dependent variables, the goal of the problem is prediction. Essentially, in analysis of structure, the method that was followed is the user is asked if there are any suspect independent variables affecting the model and if user points any variables, a new model is generated with new variables and the MSE is calculated for new variables to check if there is any significant effect on the model

3 NEURAL NETWORK METHODS

3.1 Introduction

The non-statistical methods are often used when certain assumptions behind the statistical such as the normality assumption and the assumption of the data lying within the analyzed data bounds methods do not hold. In addition, if the confidence interval obtained by the statistical methods is too wide to make any positive inferences, non-statistical analysis approach is preferred. One of the powerful ways of solving problems via non-statistical methods is neural networks.

Neural Networks originated from research aimed at understanding biological information processing system [5]. Unlike traditional computing where a single processor is used to process information and analyze results, the neurons in animal brains contain a small amount of memory and they process the data in parallel to obtain the results. The concept of Artificial Neural Networks (ANN) was born in an attempt to simulate the brain's processing system [4]. Even though there is lack of understanding on how the brain works, models that are simple enough to build produce reasonably good results. Instead of using the central processor to analyze a number of instructions, the neural net software analyzes the data by passing it through several simulated processes that are interconnected.

Once the data that is to be analyzed is collected, the network processes it to learn how the inputs are related to the output. The network considers one data point at a time and trains itself for achieving the output and moves on to the next data point. After the network is trained for all available data points, it then calculates the output for each data point and the squared error value for the complete data. This process of training through all of the data sets is called one epoch or iteration. Iterations like this are continued till the cumulative squared error reaches a specified minimum value. The two main ANN algorithms are the Back Propagation Algorithm (BPA) and the Cascade Correlation Algorithm (CCA). The notation used for these algorithms are described in Table 3.1. This is similar to the naming convention that was used in [17].

3.2 When to Use Neural Networks

Now that the nature of neural networks is understood, the most important question will be when to use Neural network algorithms. Neural network algorithms are thought of as related to AI, Parallel processing, machine learning and other fields. Neural network algorithms are ideally suited when it comes to solving complex problems. A complex problem can be defined as a problem that cannot be clearly solved using traditional statistical methods – For instance, image recognition in the presence of varying background might be a very complex problem when attempted to solve using traditional statistical methods but it could be relatively simpler when tried to solve using a neural network model. So, before considering solving a problem for a dataset, the following questions should be answered:

- A large number of input and output datasets are available but the relationship between them is not clear
- The problem appears to be very complicated but a solution can be easily identified
- It is easy to create a number of examples for correct behavior
- The outputs are fuzzy and non numeric.

3.3 Key Parameters

3.3.1 Learning Rate

When a network is trained, the weights are adjusted in such a way that the overall error is minimized. The rate at which these weights are to be adjusted is decided by the learning rate parameter α. For example, if 10% error in the weights is observed, and if the learning rate is set to 0.1%, the network needs 10 iterations to get adjusted to the error. Higher learning rates need fewer epochs and converge quickly to the minimum error value fast. However, a higher learning rate may lead to the network to memorize the values. That is, a higher learning rate might work for the training set, but not for the testing set. Lower learning rate needs more epochs and converges to the minimum error slowly. This increases the time the network takes to converge to the

minimum value. The decision on the learning rate is dependent on the time that is available for the algorithm to be trained and the generalization that is required for the training algorithm.

3.3.2 Patience Factor

Patience Factor (*PF*) is an integer value that tells the program how long it should wait before it can terminate training when the error stops converging. The inclusion of the patience factor allows the program to wait for a specific time and stop. This factor is necessary because in the training algorithm, the algorithm is set to run as long as the error does not diverge. This condition includes the cases of the error converging or staying at a constant value. This can cause the program not to stop at the constant error value and run forever.

3.3.3 Momentum

Momentum is a factor added to the Back Propagation algorithm to increase the rate of convergence without having to increase the learning rate. The inclusion of the momentum factor reduces the number of iterations by as much as 92%†. The inclusion of *Momentum* in the equation also reduces the probability of the network being caught in local minima. The way the momentum factor affects the training of the network is further discussed in [32].

3.3.4 Maximum Hidden Nodes

Maximum Hidden Nodes (*MHN*) is an essential criterion when it comes to the cascade correlation algorithm. In this algorithm, the network starts with no hidden nodes and it adds hidden nodes one by one till the error reaches a minimum. In this network, the user has to specify the maximum number of the hidden nodes the network can add. This can also be a stopping condition since the training has to be terminated once the maximum number of hidden nodes is reached. The details of the methods to decide on the number of hidden nodes are discussed in [10].

† Data from NeuNet Pro [23], When data was trained without using the momentum factor, the number of iterations it took to converge to an error of 0.35% was 476000 and when momentum is used, the number of iterations it took was 34100, a reduction of 92%

3.3.5 Maximum Allowed Error

The Maximum Allowed Error is defined as the maximum total RMS error that is allowed in the network. The primary objective of the training algorithm is to reduce the total error to a value less than the Maximum Allowed Error value. This is the reason why this condition is also used as the stopping condition for the program. Once the total error is less than this threshold, the algorithm checks for the robustness of the network. The following is a list of notations used in the Neural network algorithms and in development. Most of the notation is borrowed from [25]

Table 3.1 NN Algorithm Variable List

Variable	Description	Variable	Description
NIN	Number of Input Nodes	MinT	Max. val. of the Target Dataset
NON	Number of Output Nodes	Alpha	Learning Rate
NHN	Number of Hidden Nodes	PF	The Patience factor
NDV	Number of Data Values	Bias	Bias Value
TrDV	Training Data Values	MinError	Minimum Error Allowed
TsDV	Testing Data Values	LoopCount	The Iteration Count
MHN	Max Number of Hidden Nodes	Momentum	The Momentum Value
CHN	Current Hidden Node	TErr	Total Error in the current Iteration
$X_{n,i}$	Un-scaled Input Dataset	TPErr	Total Previous Error
$SX_{n,i}$	Scaled Input Dataset	Err	Difference between target and output
$T_{o,n}$	Un-Scaled Target Dataset	$WIH_{i,h}$	Weight from input to hidden layer
$ST_{o,n}$	Un-Scaled Target Dataset	$WHO_{h,o}$	Weight from hidden to output layer
$Y_{o,n}$	Un-Scaled Output Dataset	DWIH	Delta V
$SY_{o,n}$	Scaled Output Dataset	DWHO	Delta W
$cSY_{o,n}$	Computed Output	IHL_h	Net Input to Hidden Layer
MaxX	Max. Val of Input Dataset for a Node	IOL_o	Net Input to Output Layer
MinX	Min Val. of Input Dataset	TotHErr	Total H Error
MaxT	Max. val. of the Target Dataset	Rb	Robustness

3.4 Stopping Condition

The stopping condition decides when the training algorithm should stop. Some of the common stopping conditions are:

Minimum Error value: If the error reaches a minimum value the program terminates. The negative error value with this kind of a stopping condition is the way the error is checked. The error value is calculated in the current iteration is checked against the error value calculated in the previous iteration. So, the training process

continues till a point where the error reaches the minimum value and starts increasing. There is no guarantee that the error reached its global minimum.

1. Fixed Number of Iterations: Whether or not the error reaches its minimum value, the training algorithm runs for a preset number of iterations and stops. It then searches for the global minimum point of the error values and it saves the weights that were used in that particular iteration. The drawback of this method is that the user has to specify the number of epochs the training algorithm should perform. This number of iterations can go to a very high value and there are various techniques available to reduce this number. The effect of different parameters in Cascade Correlation Algorithm on the Iteration Count is discussed in [15]. The different techniques available for Back Propagation Algorithm are:
   - Through magnification of back propagated error [28]
   - Through changing the activation function [31]
   - Through fine tuning of the coefficients of the different parameters [5]
2. Patience Factor: If the error value does not change considerably over a period of time, stop the training process. The training process is monitored over a preset number of iterations and if there is no significant reduction of error, the program is terminated. The probability of this happening is less compared to the other stopping conditions. This approach is used in conjunction with the other stopping conditions. The main purpose of this stopping condition is to bring the training process to a stop from an infinite loop.

In the SANE software the stopping condition is a mix of all of the above stopping conditions. The patience factor is read as an input parameter and once the limit is reached the user is acknowledged of the same. The user is then allowed to choose if he/she wants to end the training or to proceed further. The user is given a choice to whether or not end the program at every local minima and he/she is also asked the number of iterations that the training should continue for before it can stop.

The above approach prevents the training process from ending prematurely since the user decides on the targeted minimum error value. It also avoids getting stuck

when the error does not reduce. The algorithm does not run forever for there is a limitation on the number of iterations.

3.5 Robustness

Robustness of a neural network algorithm is defined as the ability of the network to reach the same target value even if the starting point of the training algorithm is different. This definition, if stated from the dataset perspective, can be stated as the ability of the algorithm to react to the changes in the testing dataset. This perspective is particularly helpful in the testing of the trained neural networks. Typically, testing of neural network algorithms includes following steps

1. Divide the data set into training and testing sets
2. Train the network using the training data and then test it using the testing dataset.
3. Compute the total error in the dataset.
4. Alter the training and testing datasets while keeping their relative proportions constant.
5. Train the network with the final weights in the previous iteration as the initial weights.
6. From the definition of the robustness, if the network is robust, there should not be much change in the weights, implying that there should not be much change in the total error value.
7. If the change in the total error value is greater than a preset threshold value, steps 4 and 5 are repeated.

The robustness of the neural network algorithm is here defined as the reciprocal of the differential error obtained from applying the same weights on two different testing datasets.

$$Rb = \frac{1}{(Err_{Curr.Train} - Err_{Prev.Train})} \qquad \text{------ (3.1)}$$

2.1.1. ROBUSTNESS IN A REMOTE USER ENVIRONMENT

The approach discussed in the previous section would be difficult to implement in a remote user environment for the user might be rather novice. To solve this problem, the robustness calculation is incorporated into the algorithm. The calculation for the robustness is added as an outer loop to the training algorithm. This enables the calculation for robustness to be a stopping condition. The modified algorithm is as follows:

3.6 Back Propagation Algorithm

Step 1: Data Input and Scaling

Data is read from the input table and the variables X and T are populated. The data is the scaled to a range [0.1, 0.9] using the following equation. Scaled data is used by sigmoid activation function. It is a continuously increasing function in this range. Later, this function tends to be rather non- decreasing, than increasing, thereby effecting the sensitivity of the algorithm.

$$SX_{i,n} = 0.1 + \frac{X_{i,n} - MinX_i}{MaxX_i - MinX_i} * 0.8 \quad \text{------} \quad (3.2)$$

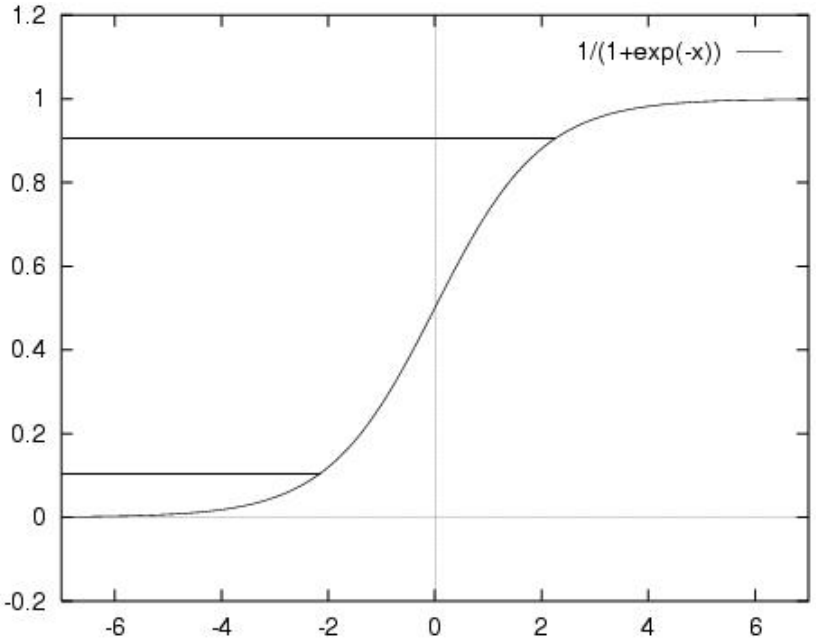

Figure 3.1: Range of the Sigmoid function used [0.1, 0.9]

Step 2: Initialize Network

Split the total dataset into testing and training datasets. The selection of the data points is randomized keeping the proportions constant. This ratio is converted into a data point count in the testing and training sets. The weight related variables, WIH, WHO are initialized to small random values. The total error in the network is calculated using the testing data set for the initial weights using the following equations:

$$IHL_h = \sum TSX_{n,i} * WIH_{i,h} \quad \text{------} \quad (3.3)$$

$$IOL_o = \sum f(IHL_h) * WHO_{n,o} \quad \text{------} \quad (3.4)$$

Where,

$$f(z) = \frac{1}{(1 + e^{-z})} \quad \text{------} \quad (3.5)$$

$$TSY_o = f(IOL_o) \quad \text{------} \quad (3.6)$$

$$ERb_{o,1} = \sum 0.5 * (TST_o - TSY_o)^2 \quad \text{------} \quad (3.7)$$

The subscript 1 in equation (3.7) shows that the error calculated here is the initial error.

Step 3: Feed Forward

During the feed forward stage, each input node receives an input signal and broadcasts the signal to each of the hidden nodes. Each hidden node then computes its activation and then sends it to each output node. Each output node computes its activation and the response is calculated for the given input pattern using the following equations. Equation (3.8) computes the net input to the hidden layer. The weighted sum of IHL is calculated using (3.9) and that is the Input to the Outer Layer (IOL). This calculated value of the output node is found using (9).

$$IHL_h = \sum WIH_{i,h} * SX_{n,i} \quad \text{------ (3.8)}$$

$$IOL_o = \sum WHO_{h,o} * f(IHL_h) \quad \text{------ (3.9)}$$

$$SY_{o,n} = f(IOL_{h,o}) \quad \text{------ (3.10)}$$

Step 4: Back Propagation

Each output so calculated is compared against the target value and the error value is calculated. This value is used to distribute the error back to the hidden layer

$$Err_{o,n} = ST_{o,n} - SY_{o,n} \quad \text{------ (3.11)}$$

$$DWHO_{0,o} = \alpha * Err_{o,n} * df(IOL_{h,o}) \quad \text{------ (3.12)}$$

$$DWHO_{n,o} = DWHO_{0,o} * df(IOL_{h,o}) \quad \text{------ (3.13)}$$

$$df(z) = f(z) * (1 - f(z)) \quad \text{------ (3.14)}$$

$$DWIH_{0,h} = \alpha * Err_{o,n} * df(IOL_{h,o}) * WHO_{h,o} * df(IHL_{i,h}) \quad \text{------ (3.15)}$$

$$WIH_{i,h} = DWIH_{0,h} * SX_{i,n} \quad \text{------ (3.16)}$$

Step 5: Adjust Weights

The error so transmitted is used to update the values of the weights between the hidden layer and output layer. These are also used to update the weights between the input layer and the hidden layer.

$$WIH_{i,h} = (1 + Momentum) * DWIH_{i,h} + WIH_{i,h} \quad \text{------ (3.17)}$$

$$WHO_{h,o} = (1 + Momentum) * DWHO_{h,o} + WHO_{h,o} \quad \text{------ (3.18)}$$

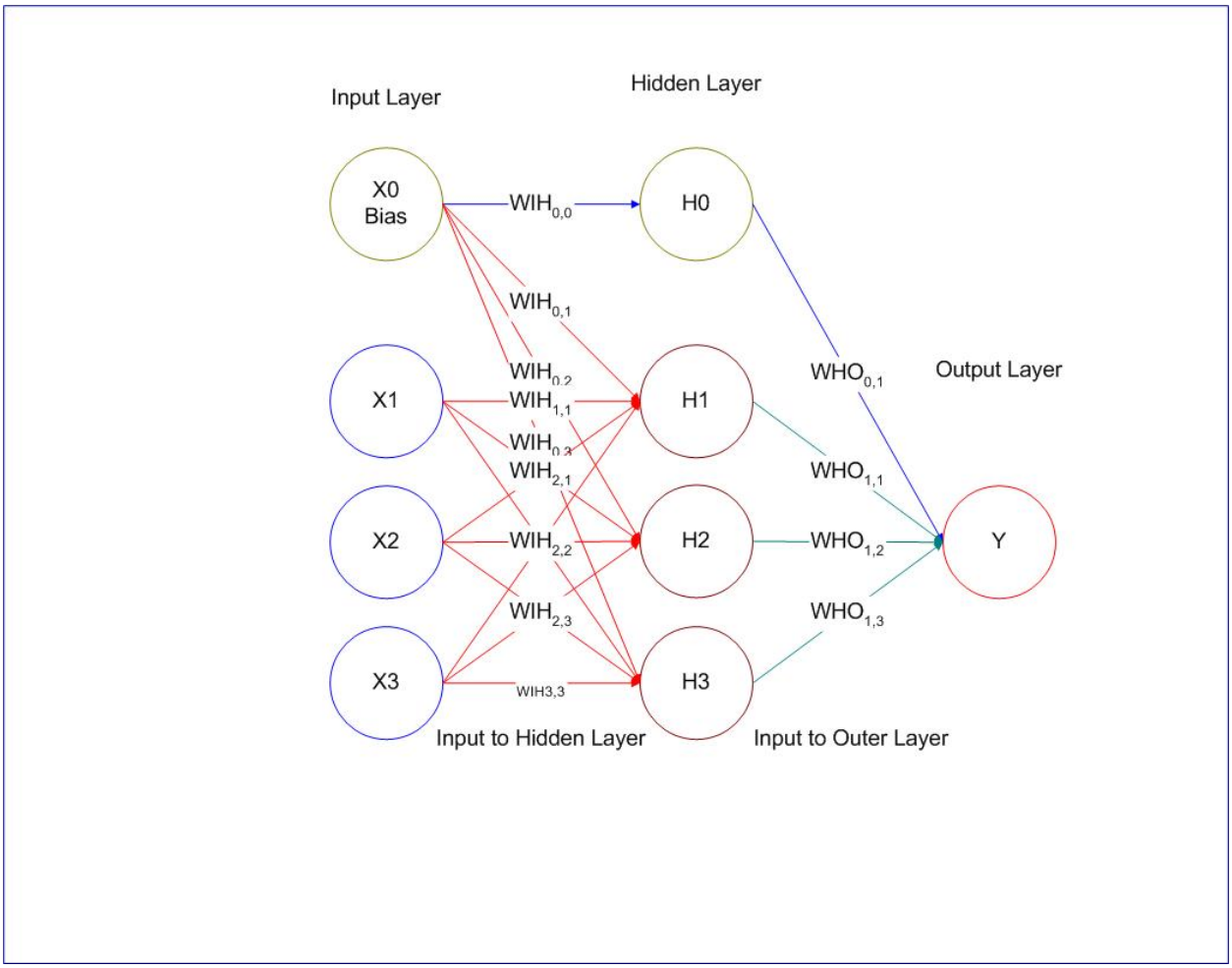

Figure 3.2: Architecture of BPA implemented using a Feed Forward Network

3.7 Cascade Correlation Algorithm

In the Cascade Correlation Algorithm (CCA), hidden nodes are added "automatically". Initially, the training starts off with no hidden nodes, as shown in Figure 3.3 and the minimal network is trained for minimal error. Upon completion of training, a

new hidden node is added. Each hidden node receives a connection from each of the networks input nodes and also from the already existing hidden nodes. This hidden node is trained till it has best correlation with already existing network and current error than from the previous stage and then the added hidden node is frozen. This creates multiple hidden layer architecture and makes the algorithm very powerful. The sample final architecture of CCA is shown in Figure 3.4.

Since the algorithm trains each hidden node and the training objective in this case is to maximize the correlation of the hidden nodes with respect to the total residual error from the rest of the network instead of just reducing the total error. This causes the algorithm to freeze the existing network once the hidden nodes are added and it just takes care of the one hidden node that is being trained. So, in this case, robustness assumes a different definition. Here in this case, where it is not possible to train the network once it is done with training, it is not possible to adjust weights to suit the different training datasets. So, in this case, the definition is changed to train the network with the best dataset that represents the complete data. Since the objective of the training algorithm is to find the best dataset, the robustness becomes a qualitative variable and its definition would be 'If the mean of the squared error observed in the training and testing datasets are not significantly different, the training algorithm is robust'. The confidence levels can be set according to the results obtained from the dataset. The training algorithm is as follows:

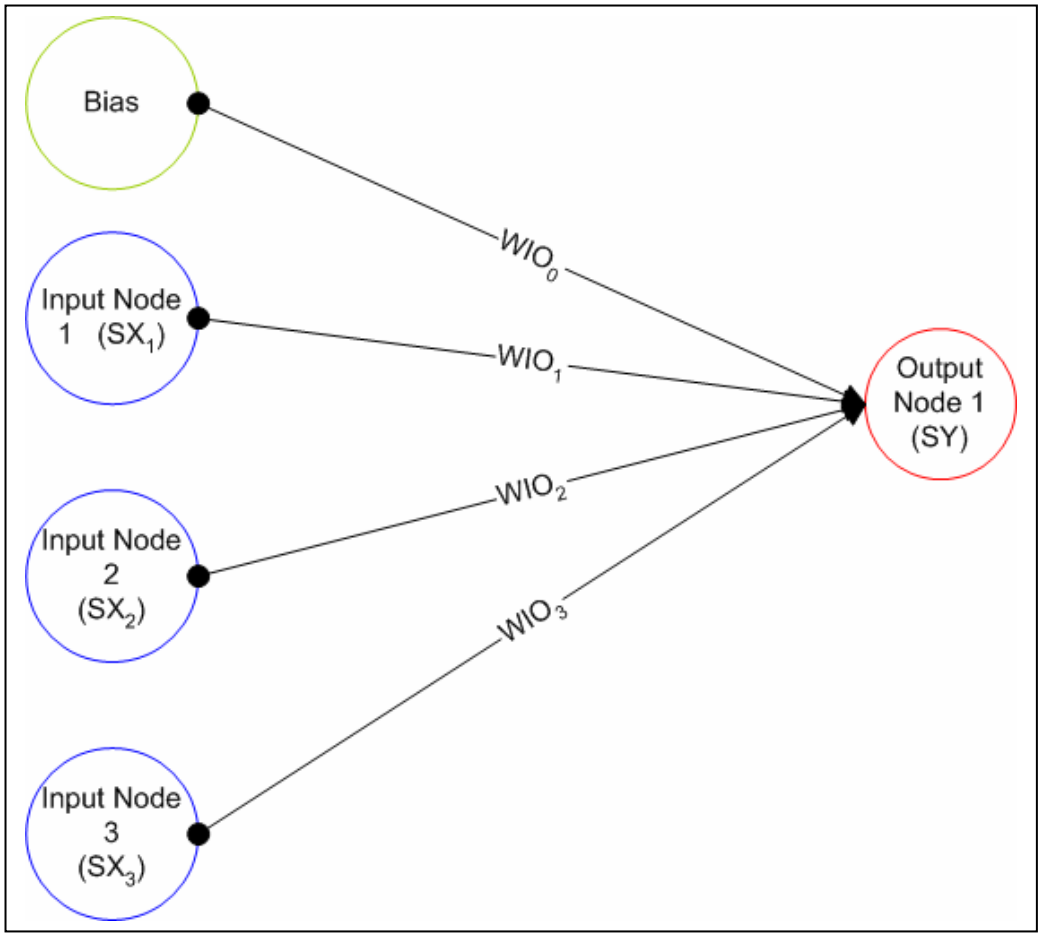

Figure 3.3: CCA Initial Architecture

The cascade correlation algorithm is as follows:

Step1: Data Input

In this step data is read into *X* and *T* (vectors). The values for *NIN, NON, NDV* are updated. The read *X* and *T* values are then scaled using equation 3.2 and the variables *SX* and *ST* are populated.

Step 2: Initialize Network

The weight related variables, WIH, WHO, WHH[ϒ], WIO are initialized to small random values.

(WHH is only initialized after the addition of the second hidden node

Step 3: Train Network

Compute the Scaled Output Values

$$cSY_{o,n} = WIO_{0,O} + \sum_{i=1}^{NIN} SX_{i,n} WIO_{i,o} \quad \text{------ (3.19)}$$

$$SY_{o,n} = f(cSY_{o,n}) \quad \text{------ (3.20)}$$

Find the error value by comparing the scaled output value with the target value

$$Err_o = (ST_{o,n} - SY_{o,n}) * df(cSTY_{o,n}) \quad \text{------ (3.21)}$$

Calculate the weight adjustments based on the error value and the learning rate

$$DWIO_{0,o} = \alpha * Err_{o,k} \quad \text{------ (3.22)}$$

$$DWIO_i = \alpha * Err_{o,k} * SX_{i,n} \quad \text{------ (3.23)}$$

$$WIO_{i,o} = WIO_{i,o} + DWIO_{i,o} \quad \text{------ (3.24)}$$

$$TErr = \sum Err^2{}_{o,k} \quad \text{------ (3.25)}$$

Step 4:

Repeat steps 5 and 6 while the current number of hidden nodes is less than the maximum allowed hidden nodes and the total error stops reducing (*H<=MHN* and *TErr>TPErr*)

Step 5: Add Hidden Node

$$IHL_{i,CHN} = WIH_{0,h} + \sum SX_{i,n} + \sum WHH_{h-1} IHL_{i,CHN} \quad \text{------ (3.26)}$$

$$IHL_{CHN} = f(IHL_{CHN}) \quad \text{------ (3.27)}$$

$$totalHErr = \sum_{n=1}^{NDV} IHL_{i,h} \quad \text{------ (3.28)}$$

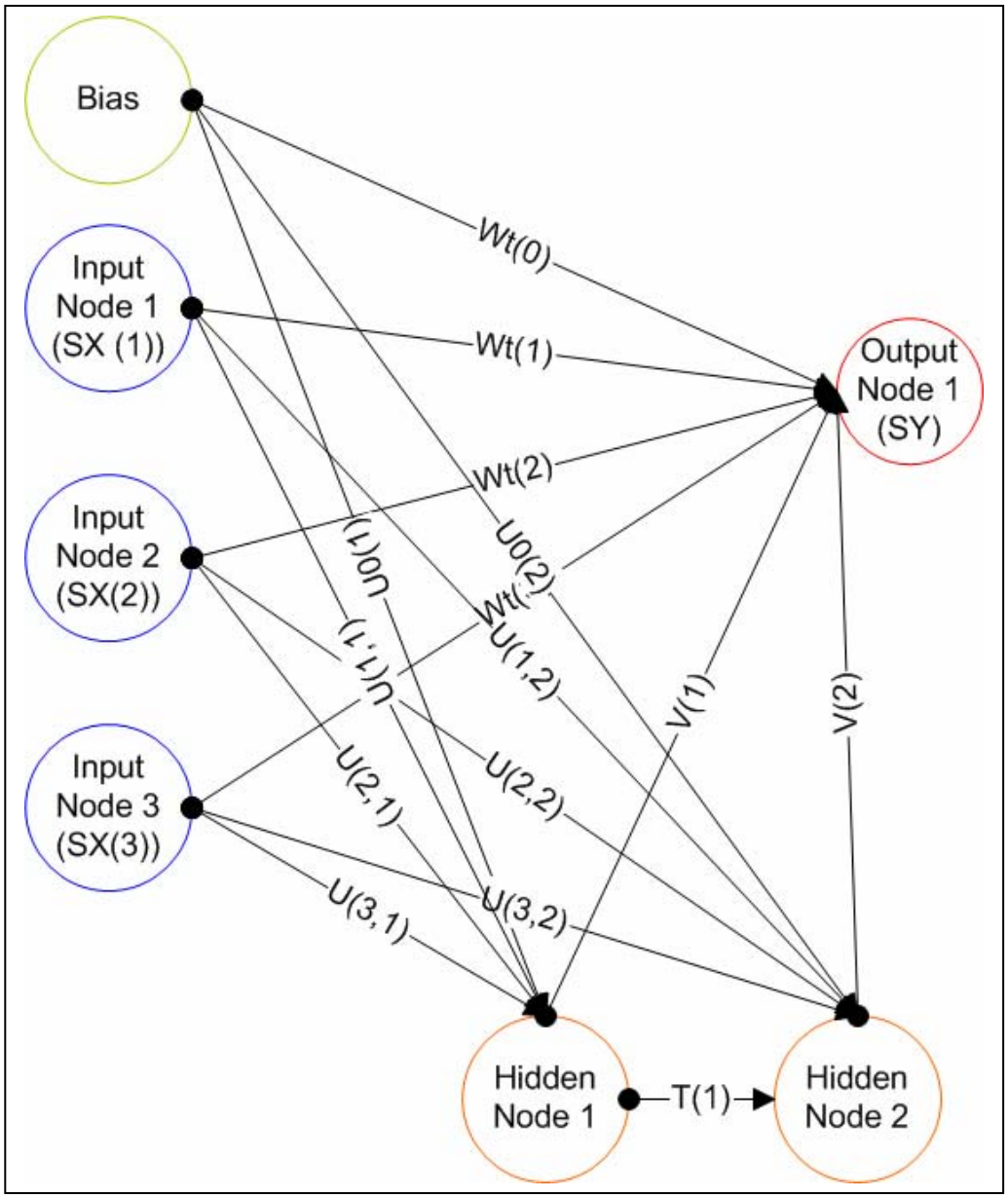

Figure 3.4: CCA Final Architecture

Step 6: Train Network till the Correlation is maximized

$$CR = \sum_{o=1}^{NON} \left| \sum_{n=1}^{NDV} (IHL_{i,CHN} - totalHErr / NDV) * (Err_{o,n} - TErr / NDV) \right| \text{------ (3.29)}$$

$$DCR_i = \sum_{o=1}^{NON} SCR_o \sum_{n=1}^{NDV} IHL_{i,CHN} SX_{i,n} (Err_{o,n} - TErr / NDV) \quad \text{------ (3.30)}$$

Where SCR is the sign of

$$\sum_{n=1}^{NDV} (IHL_{i,CHN} - totHErr / NDV) * (Err_{o,n} - TErr / NDV) \quad \text{------ (3.31)}$$

$$CSY_{o,n} = WIO_{0,o} + \sum_{i=1}^{NIN} SX_{i,n} WIO_{i,o} + \sum_{h=1}^{CHN} IOL_{h,o} WHO_{h,o} \quad \text{------ (3.32)}$$

$$SY_{o,n} = f(CSY_{o,n}) \quad \text{------ (3.33)}$$

Error is calculated using equation 3.21 and the weights are updated using equations 3.22 – 3.28.

Example of Cascade Correlation Algorithm: Patient Satisfaction Data is used for training of this algorithm. The data set has 10 data points, 3 input nodes and 1 output node. The learning rate is set to 0.2 and the maximum number of hidden nodes is set to 2. The network took 32 iterations to be trained, table 3.3 shows the results.

Table 3.3 CCA Output

Achieved Values	Desired Values
0.315	0.251
0.378	0.316
0.208	0.1
0.429	0.34
0.317	0.25
0.316	0.179
0.322	0.181
0.554	0.274
0.886	0.9
0.868	0.86

The following is the plot that shows how the training algorithm followed the actual dataset.

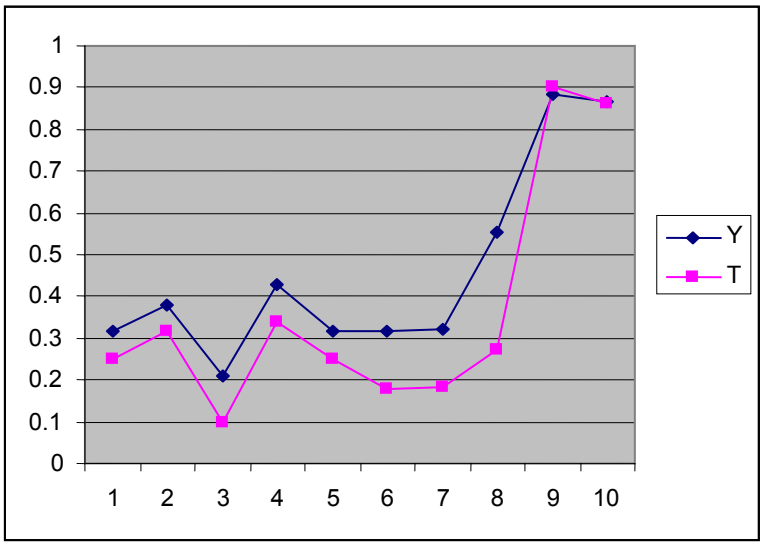

Figure 3.4 CCA Training Network Output (Y – Achieved and T – Target)

4 SOFTWARE DESIGN AND IMPLEMENTATION

4.1 The SANE System

The SANE software is a web based system. Its front end is in ASP.Net/Jscript and the back end in Microsoft SQL Server 2000. The proto type of the design was implemented in VB.Net as a stand alone model. It will be modified and implemented in ASP.Net. The updated version of the software can be found at the URL http://dev.rsa.wvu.edu/SANE/. The design is broadly classified into Statistical and Non Statistical Methods and the user selects the method he/she wants to use in order to solve the problem.

4.2 Software Design Process

Requirement documents were developed in order to quantify the objectives and compare the implemented methods with quantified objectives; asses if any gaps were present in the analysis and thereby progress towards the complete fulfillment of the objectives. The requirement documents were created and are divided into two major classes.

a. The Design Specification documents and

b. The Functional Specification documents.

The design specification document basically outlines the design of the system as to explaining “what needs to be done” and the functional specification document explains how the system works as to explaining “what was done”. A comparison of these documents was referred to as “Gap Analysis” and that describes the gaps left over, if there are any. In the case of SANE, the documents were created to organize, design and develop the software along the design guidelines. The document format is customized to suit design needs and to incorporate into this document. The Objective (requirement) and the implementation part are combined as two parts of the same document and are presented as two parts in the same document. These documents are presented in a rather high level design perspective. The requirements themselves are present in Appendix B.

4.3 Software Design

The software was implemented in ASP.NET with SQL Server 2005 as its backend. For convenience in programming, the whole project is divided into modules. The modules are organized in such a way that all functions go into one module, all variable declaration goes into one module, all interface programming goes into one module and all backend operations are accessed from one module. So, for any analysis type, the program will access three different modules.

Function Modules:

Function modules contain all functions related to different algorithms. Examples are:

- Statistics Module (myStatlib.vb): This module consists of different statistical operations. These operations include Descriptive Statistical functions, functions for permutations and combinations, discrete and continuous distributions.
- Matrix Library (mdlMatrix.vb): This Module consists of different matrix related functions. These include functions for Matrix algebra.
- ANOVA Library (mdlANOVA.vb): This module consists of different procedures to perform ANOVA in different scenarios.
- Back Propagation Algorithm Module (mdlBPA.vb): This module contains different sub routines for BPA training.
- Cascade Correlation Algorithm Module (mdlCCA.vb): This module contains different sub routines for CCA training.

Variable declaration module:

To make development more uniform, all variable references are directed to an outside declaration module. These modules contain all public variable declarations and some most general function definitions (Ex: Sigmoid function). Examples of such modules are:

- Statistical variable Module (mdlStatVar.vb): This module contains the variable list used for statistical analysis

- Neural net variable Module (mdlNNVar.vb): This module contains the variable list used for Neural net analysis.

And, there are two program modules that relate to user management and dataset entry. These are the only modules that access the database with read write permissions. The modules are:

- User Management Module (*mdlUser.vb*): This module consists of the basic user related operations. These operations include User creation, deletion and user account modifications.
- Data Entry Module (*mdlData.vb*): This module contains the data entry related routines to enter data sets into database.

4.4 Front End

The Home Page: The home page consists of the login options and the basic Neural and Statistical analysis branches. However, the buttons choosing the basic analysis types are disabled until the login requirements (proper user id and password) are met. The page consists of four major areas:

1. Header is the top frame and contains the user information.

2. Navigation bar is the frame that is on the left of the screen and it contains the list of the problem types/algorithms the user can select.

3. The Work Area occupies the maximum area and this is the area where the actual problem solution takes place. The user inputs the data here and performs the appropriate analysis.

4. The Footer is the frame in the lower area where the help is displayed. This frame contains brief help regarding the analysis type being used.

4.5 Navigation of the Website

Statistical Analysis: If the selected method of analysis is statistical, the user will be asked for the kind of analysis he/she is interested in performing. Once the user chooses the major research question, he/she will be directed to the appropriate screen.

The dataset will be read from the user and the appropriate analysis is performed. The screen that are associated with the statistical analysis are as follows:

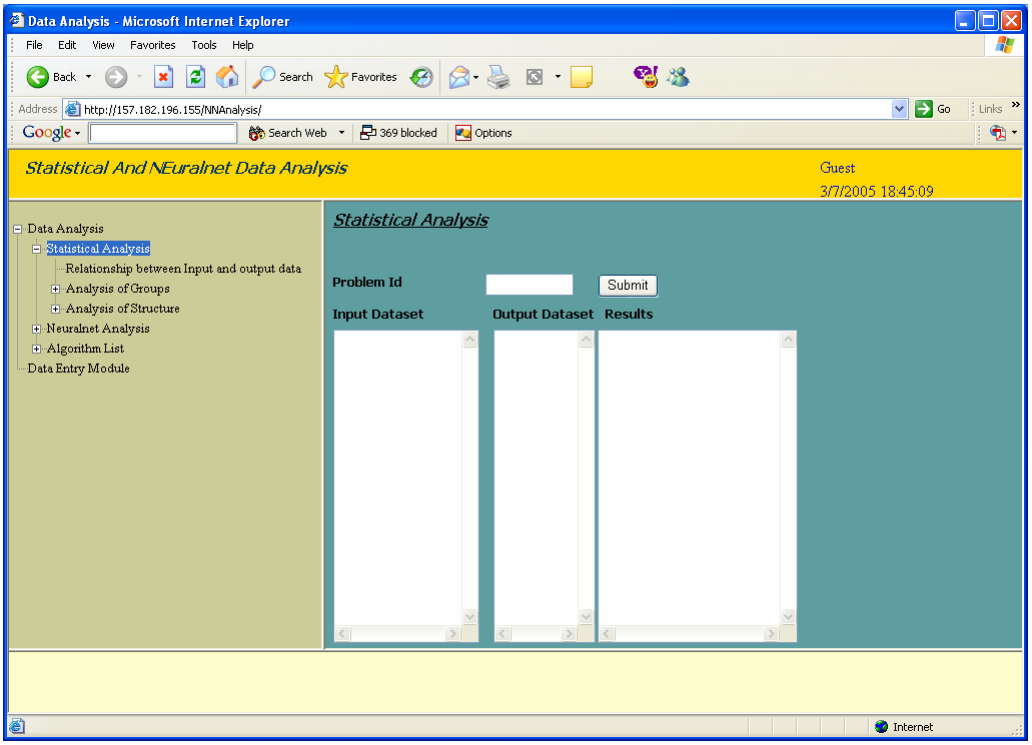

Figure 4. 1 Statistical Data Analysis

General Navigation: For the statistical methods, the user will provide the problem Id. Once the problem Id is entered, the program selects the input and output data sets.

Neural Net Model: In neural net model, the user is required to make a selection among the Back Propagation Algorithm and the Cascade Correlation Algorithm. Once the user selects the noise level in the data, the user will be directed to the appropriate algorithm interface and he/she will be provided with the default parameters for each algorithm. The various screens that are associated with the neural net model are

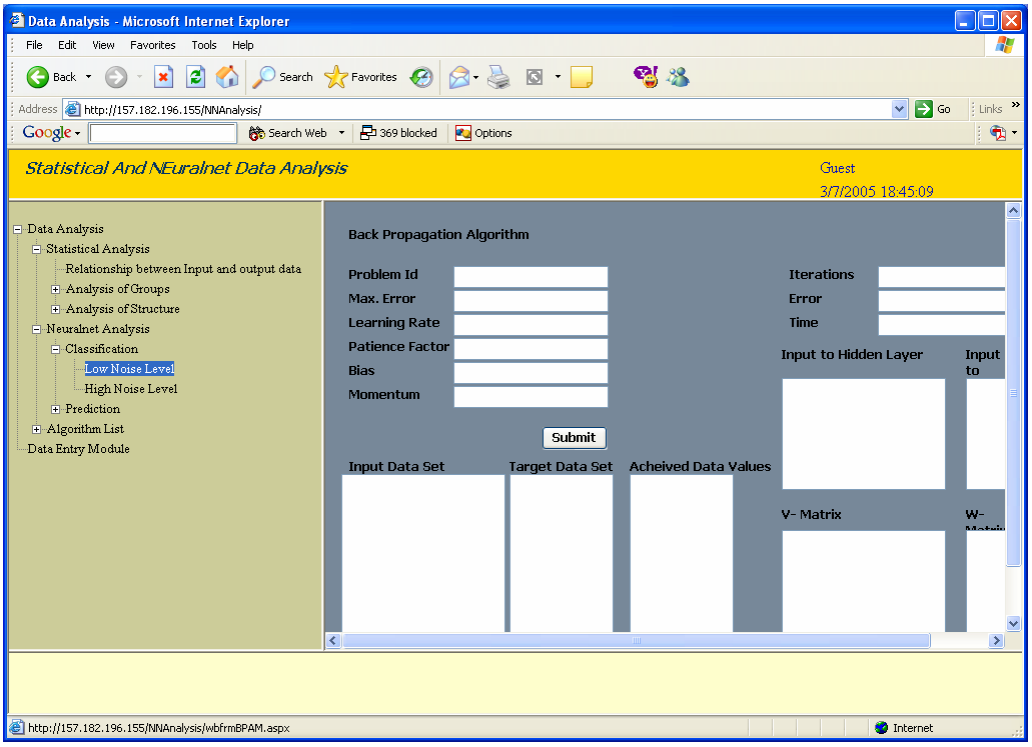

Figure 4.2: Back Propagation Algorithm

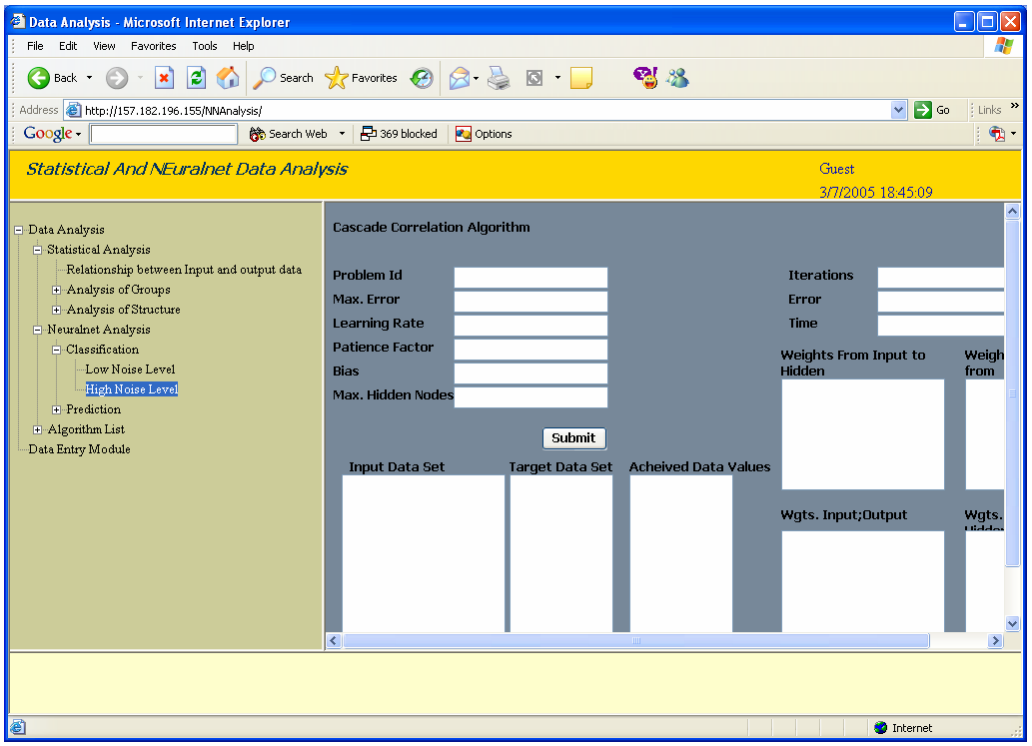

Figure 4.3: Cascade Correlation Algorithm

General Navigation: The user inputs for this segment are the problemId and the maximum error. Once the user enters the problem Id and the maximum error, an estimated learning rate, patience factor, momentum for BPA and maximum hidden nodes for cascade correlation algorithm are displayed. The user is free to change any of these factors and once the factors are finalized, the user clicks the train button. The results are displayed once the training is done. Testing data is entered in the input data set and the results can be obtained by clicking the test button. The output is displayed in the 'Achieved Data Values' box.

Data Entry Module: It is in this module the user creates and modifies the different problem data sets. The way the datasets are stored was discussed in Chapter 5.

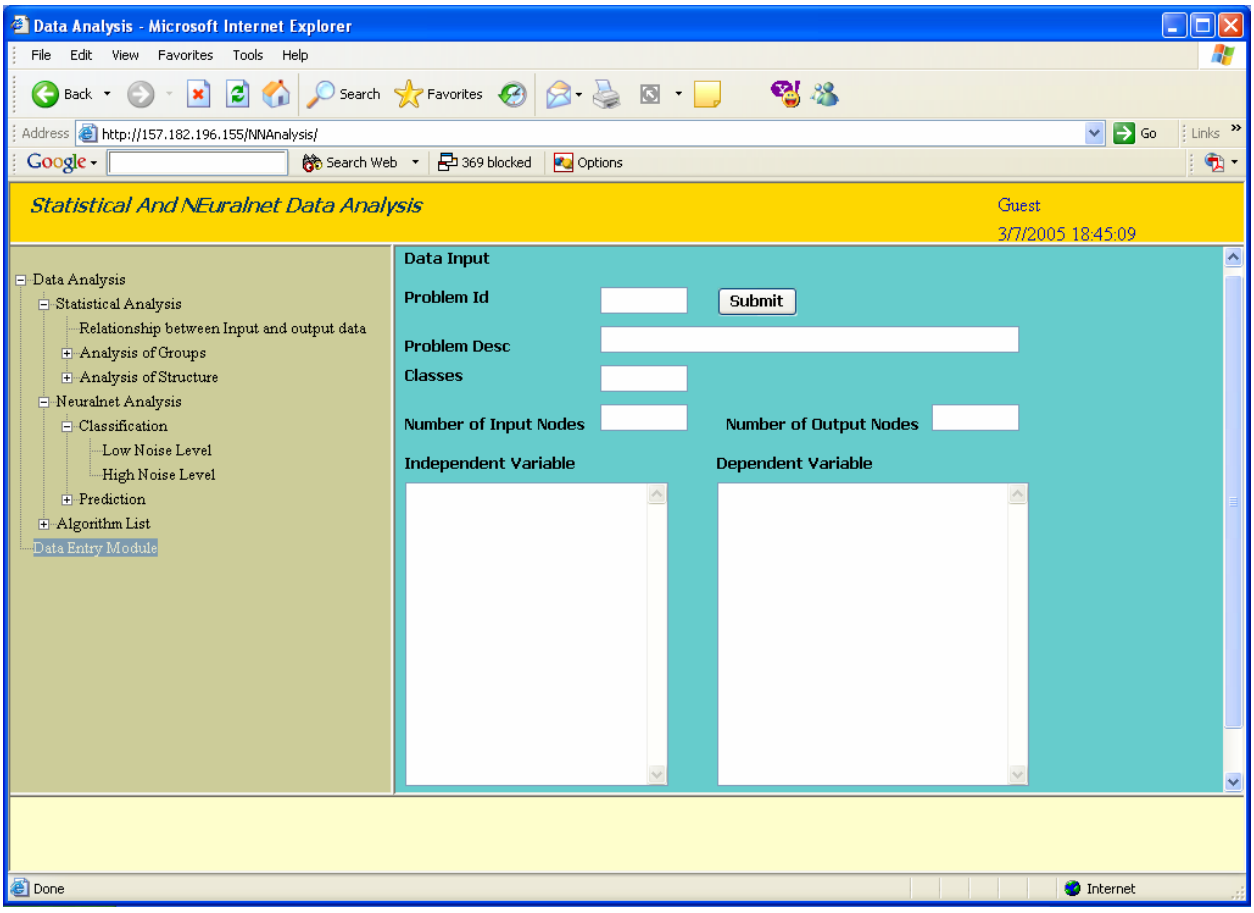

Figure 4.4: Data Entry Module

Navigation: Once the user enters the Problem Id, the program will search for a similar problem Id in the data base and if the problem already exists, it will be displayed on the screen. The user can modify the problem data. The modified records are updated in the database. If the problem Id is a new value, then a new entry for that Id in the data base and the data is stored.

5 DATABASE DESIGN AND IMPLEMENTATION

5.1 Introduction

The database of SANE is designed in such a way that data is accessible to any data analysis user intends to perform. The database is designed with following issues in mind:

The database should allow user to store and retrieve data without losing its accuracy or integrity

- The database should allow user any data analysis the user intends to perform
- The database should have a versioning capability so user could perform multiple data analyses on dataset and compare them

The database is designed to fulfill the requirements listed above and the database contains four tables

5.2 List of Tables

The Analysis Table (*tblAnalysis*)

This table stores information regarding different analyses performed on a given dataset. Its fields are shown in Table 5.1.

Table 5. 1 Analysis Table

Field Name	Data Type	Length	Description
AnalysisType	Varchar	5	The type of analysis performed
AnalysisDesc	Varchar	50	Analyisis Description

If the analyses to be performed are Bi-Variate Analysis, Multiple Regression and Back Propagation Algorithm, the table values would be:

Table 5.2 Analysis Table Snapshot	
AnalysisType	**AnalysisDesc**
BIVAR	Bi-Variate Analysis
MUVAR	Multi-Variate Analysis
BPA	Back Propagation Algorithm

The Problem Table (*tblProblem*)

This table is required for storing the problem description. One record is required for each problem. The table contains Problem Identification Number (*ProblemId*), Problem Description (*ProblemDesc*), Number of Input Nodes (*NIN*), Number of output Nodes (*NON*) and the number of classes (*Classes*) of the dependent variables available(this is a non zero value only if the dependent variables are discrete). Table 5.2 shows the structure of the table.

Table 5.3 Problem Table			
Field Name	**Data Type**	**Length**	**Description**
ProblemId	Varchar	10	Unique Problem Identification,
ProblemDesc	Varchar	50	Problem Description
InputNodes	Integer	2	Number of Input Nodes
OutputNodes	Integer	2	Number of Output Nodes
Classes	Integer	2	Number of DV Classes (Only for Discrete DV)

If for example we had two problems, one was being the Iris Data set [25] and the other being Patient Satisfaction, the table values would be:

Table 5.4 Problem Table Snapshot				
Problem Id	**Problem Description**	**Input Nodes**	**Output Nodes**	**Classes**
00001	Iris Dataset	4	1	3
00002	Patient Satisfaction Data	3	1	0

The Dataset Table (*tblDataSet*)

This table is required for storing the Datasets for each problem. This table contains the actual data sets. There is no primary key in the table but it is indexed on the *ProblemId* and the *NodeType*. The *ProblemId* in this table relates to the *ProblemId* in the Problem Table and the *Nodetype* is the type of the Node that the particular record relates to. Input Nodes are given a Node type of 1 and output Nodes are given a Node type of 2. A node Id of 0 means that that particular record is used for storing the problem description. The actual data is stored in a memo field and the values are stored in a Comma separated format. The structure of the table is as shown below

Table 5. 5 The Dataset Table

Field Name	Data Type	Length	Description
ProblemId	Varchar	10	Problem Identification
NodeType	Integer	2	Input/ Output Node
NodeDesc	Varchar	50	Node Description
Data	Varchar	0	Comma separated data values pertaining to that node

For the ProblemTable shown above, the Dataset Table would be:

Table 5.6 The Dataset Table Snapshot

Problem Id	Node Type	Node Desc	Data
00001	1(Means Input Node)	Sepal Length	5.1, 5.4……
00001	1	Sepal Width	0.5, 0.8……
…	…	…	…
00001	2 (Means Output Node)	Iris Type	1, 1……
00002	……	……	……
……	……	……	……

The Solution Table (*tblSolution*)

This table is required for storing the solution to each problem. This table is organized in a similar way to the Dataset table. It contains a non-unique *ProblemId* and a non-unique *AnalysisType*. However, the concatenated string of *ProblemId* and *AnalysisType* is a unique field. This allows the user to perform multiple analyses on the same problem. The field *ProblemId* is related to the problemId in the Problem table and the *AnalysisType* is related to the field *AnalysisType* in the Analysis table. The actual solution of the problem is stored as weights and the *Weights* field is a memo field with all weights stored in a pre determined order in a comma separated format. The structure of the table is as shown below:

Table 5. 7 The Solution Table

Field Name	Data Type	Length	Description
ProblemId	Varchar	5	Non unique Problem Identification
AnalysisType	Varchar	10	Non Unique Analysis type
Version	Number	2	Version Number
Weights	Varchar	0	Comma separated value to store weights

If the Iris data set is analyzed using the Multi-Variate Analysis and Back propagation, the following rows will be created in the solution table:

Table 5.8 The Solution table Snapshot

ProblemId	AnalysisType	Version	Weights
00001	MUVAR	1	5, 1.2, -3.5, 4.2, 1.677
00001	BPA	1	1, 2.11, 3.32, 4.87, -0.58 ……

5.3 The Relationship Diagram

The relationship diagram looks as follows for SANE database

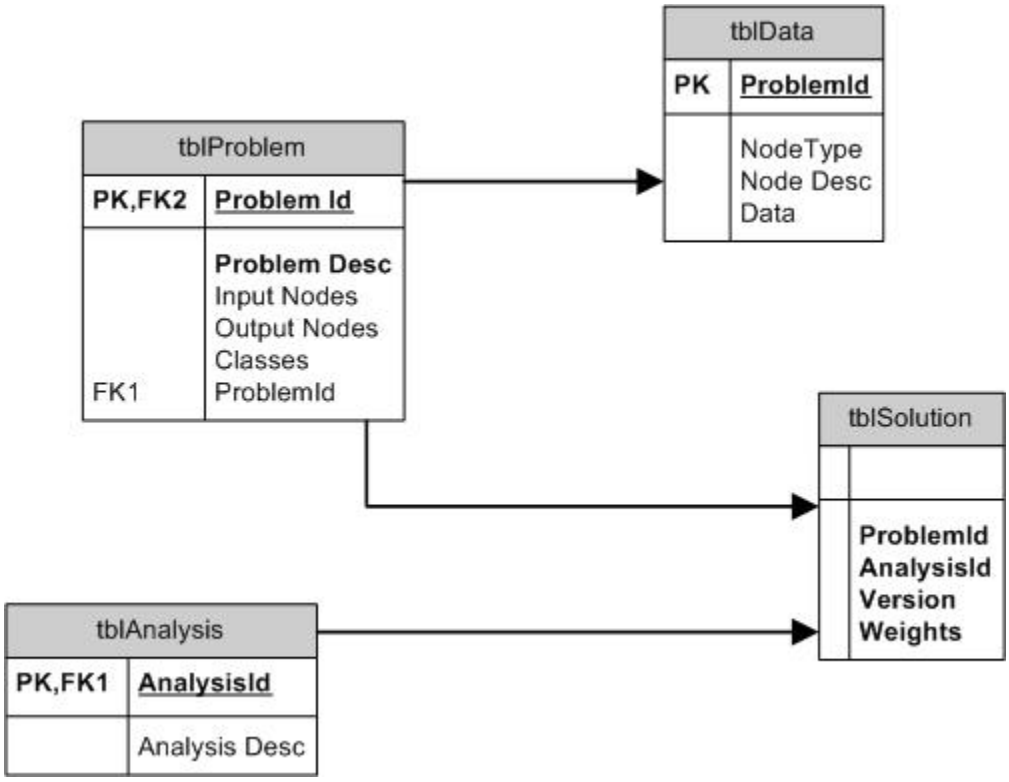

Figure 5.1 Relationship Diagram

6 APPLICATION STUDY

6.1 Introduction

This section deals about the selection of analysis along with an example. Selection of the data analysis is based on figure 6.1. The user is required to determine if data training is required. If the problem does not require training, the user is lead to the statistical analysis module. In the statistical analysis module, the user will have three different options to choose. They are

1. Degree of relationship: If the research question is to find the degree of relationship between two variables, the user has to select this option.
2. Analysis of groups: If the research question deals with groups, the user should select this option. This option leads to two different options and if the research question is to find the group membership, logistic regression is selected and if the research question is to find if there is any clustering in the data, discriminant function analysis is used.
3. Analysis of Structure: In this is the option is selected, the user will be asked if there is any hypothesis about the structure. Depending on his answer, the user is directed to factor analysis or structural equation modeling.

If the data set requires training, the user is asked about the noise level in the dataset. If the noise level is low, BPA is selected and if the noise level is higher, CCA is selected for CCA is more effective when it comes to data analysis with greater noise.

6.2 Datasets Used

To demonstrate the selection of the data analysis techniques, two different data sets were considered and analyzed with a hypothetical research question. The sample problems that were implemented were

1. Car Mileage Data and
2. Iris Data
3. Patient Satisfaction Dataset

The data set selected for the application study deals with car manufacturing countries and the mileage of the cars that are produced by those countries. The problem to solve is to find if any one particular country is making more fuel efficient cars than the others. The research question in this issue is to find the significance of the group differences. This problem needs no neural network training for it is just used to find if there is any significance in the group means. The data set for the problem is as follows:

The analysis that has to be selected is found using the following data analysis flow sheet.

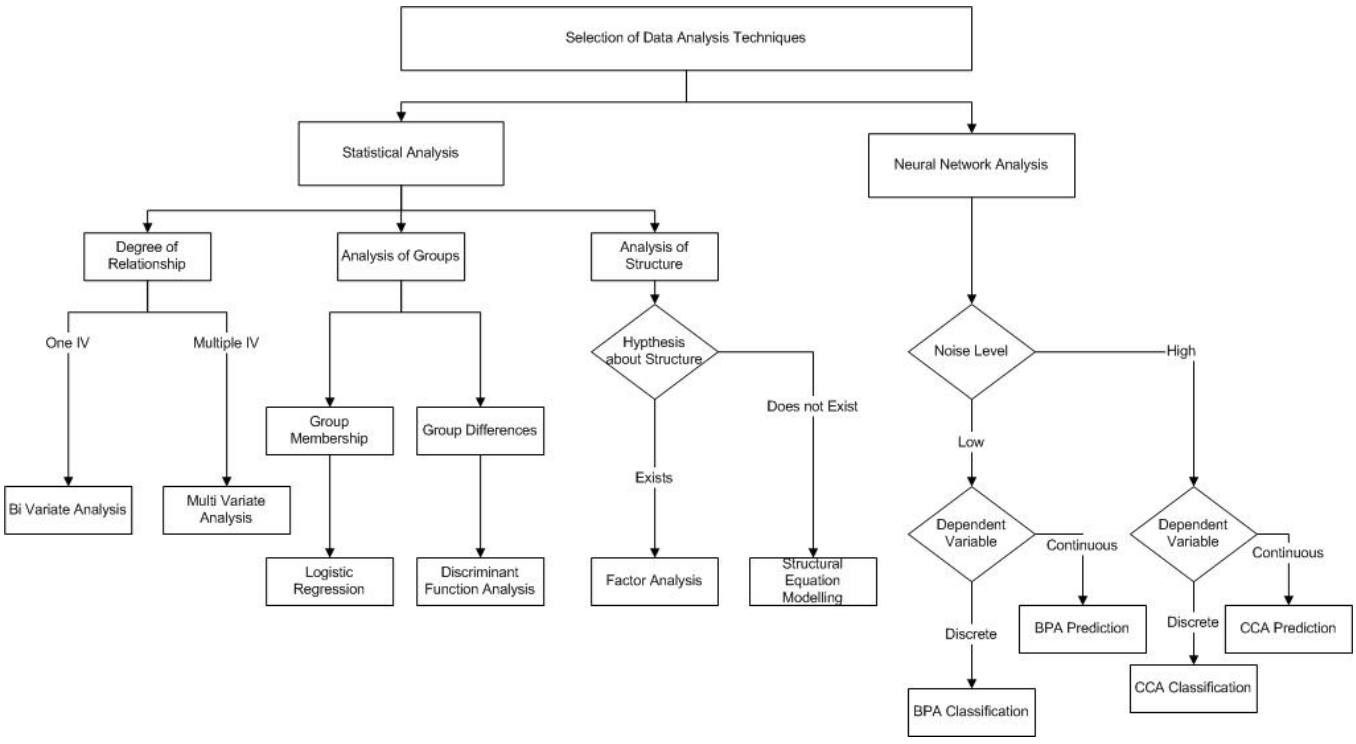

Figure 6.1 Data Analysis Table

6.3 Statistical Example

The above data set was used to determine if there is any difference in the group means and the following ANOVA is made from the data.

Table 6.1: ANOVA Car Mileage Statistics					
Treatment	**SS**	**df**	**MS**	**F**	**F***
Germany	131.552	4	32.888	0.308336	7.39
Japan	123.28	6	20.54667	0.192632	6.98
Sweden	10.58	1	10.58	0.099191	10
US	769.7295	21	36.65379	0.343642	6.22
SSE	319.9885	3	106.6628		
SSTO	1355.13	35			

The null Hypothesis in this case is that there is no difference in group means and the alternative hypothesis is that there is at least one group that has a different mean. The null hypothesis is confirmed from the above shown ANOVA table. The screenshot of the above problem is shown in figure 6.2.

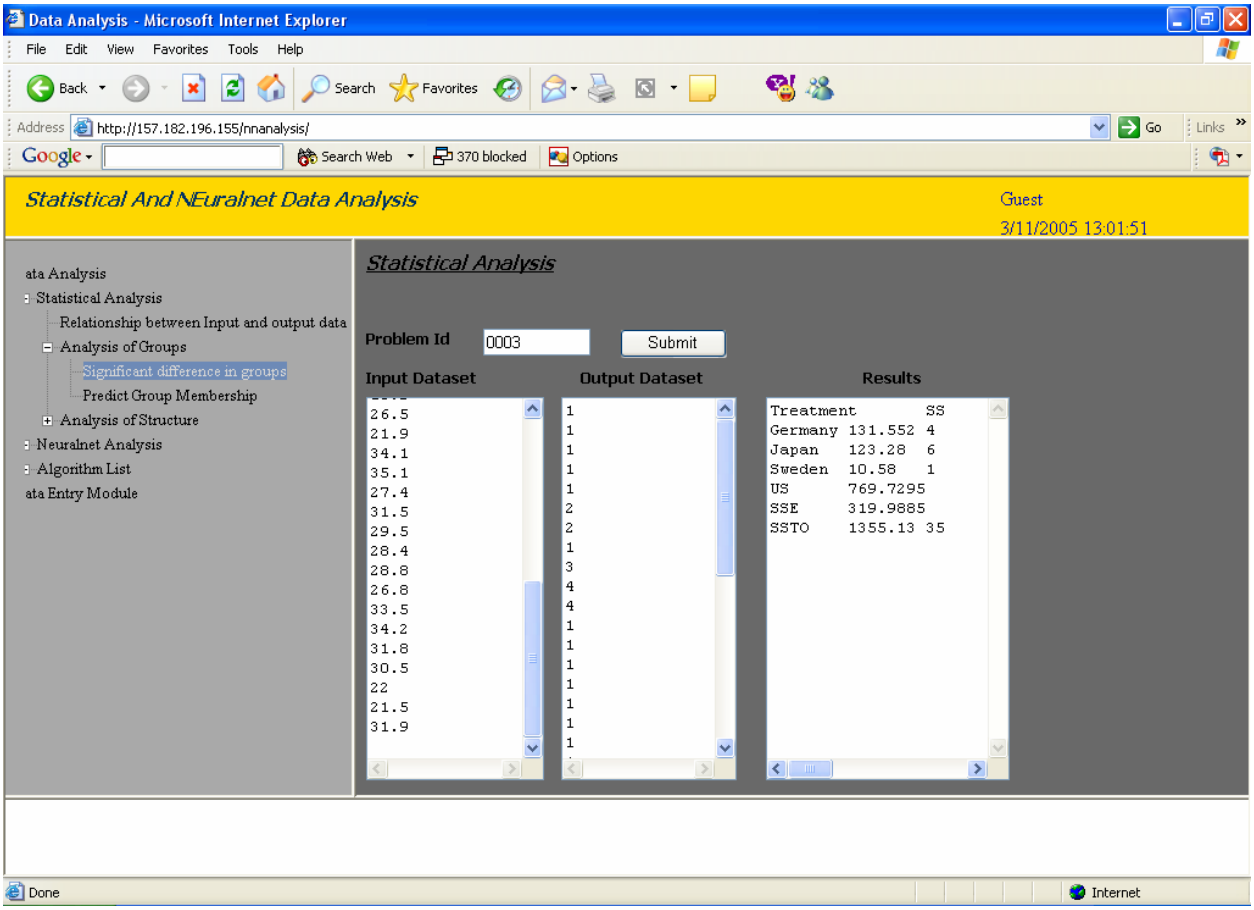

Figure 6.2: Screenshot- Significant Difference in Groups

6.4 Non Statistical Example

The data set selected for this problem is Iris Dataset. The objective of this problem is to determine the Iris type based on four different parameters (Sepal Length, Sepal Width, Petal Length and Petal Width). This data has four continuous independent variables and one discrete dependent variable. 150 Observations are used for training the Cascade Correlation network and 92.67% of the observations were classified correctly. The classification pattern is displayed on figure 6.4

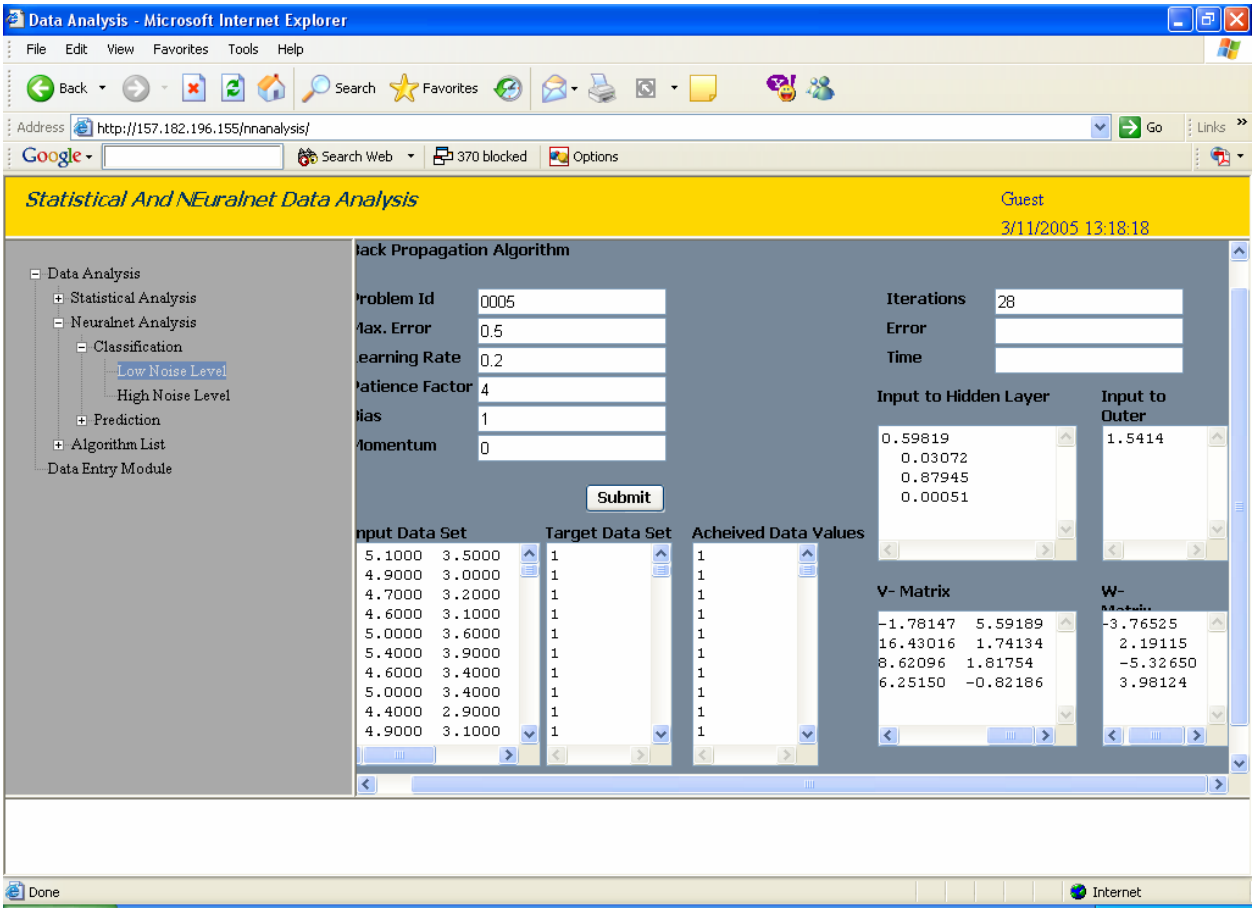

Figure 6.3 NN Analysis, BPA Classification Implementation

Figure 6.4 Iris Classification

SANE includes a module that would convert the dataset stored in database and convert it into a .arff format document. Iris dataset was thus converted from SQL Server database version and was analyzed using Weka software and was used to compare the accuracy of this implementation with SANE.

```
=== Run information ===

Scheme:       weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation:     iris
Instances:    150
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Sigmoid Node 0
    Inputs    Weights
```

```
   Threshold    -3.5015971588434005
   Node 3    -1.0058110853859954
   Node 4    9.07503844669134
   Node 5    -4.107780453339232
Sigmoid Node 1
   Inputs    Weights
   Threshold    1.0692845992273172
   Node 3    3.898873687789399
   Node 4    -9.768910360340262
   Node 5    -8.59913449315134
Sigmoid Node 2
   Inputs    Weights
   Threshold    -1.0071762383436442
   Node 3    -4.21840613382704
   Node 4    -3.6260596863211187
   Node 5    8.805122981737846
Sigmoid Node 3
   Inputs    Weights
   Threshold    3.3824855566856806
   Attrib sepallength    0.9099827458022274
   Attrib sepalwidth    1.5675138827531336
   Attrib petallength    -5.037338107319896
   Attrib petalwidth    -4.915469682506095
Sigmoid Node 4
   Inputs    Weights
   Threshold    -3.3305735922918323
   Attrib sepallength    -1.1116750023770103
   Attrib sepalwidth    3.1250096866676533
   Attrib petallength    -4.133137022912303
   Attrib petalwidth    -4.079589727871455
Sigmoid Node 5
   Inputs    Weights
   Threshold    -7.496091023618096
   Attrib sepallength    -1.2158878822058805
   Attrib sepalwidth    -3.533282131753492
   Attrib petallength    8.401834252274107
   Attrib petalwidth    9.460215580472838
Class Iris-setosa
   Input
   Node 0
Class Iris-versicolor
   Input
   Node 1
Class Iris-virginica
   Input
   Node 2


Time taken to build model: 0.73 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         146               97.3333 %
Incorrectly Classified Instances         4                2.6667 %
Kappa statistic                          0.96
```

```
Mean absolute error                  0.0327
Root mean squared error              0.1291
Relative absolute error              7.3555 %
Root relative squared error         27.3796 %
Total Number of Instances           150

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  1         0          1         1         1        Iris-setosa
  0.96      0.02       0.96      0.96      0.96     Iris-versicolor
  0.96      0.02       0.96      0.96      0.96     Iris-virginica

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 48  2 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

Figure 6.5 Weka Output for Iris Classification Dataset

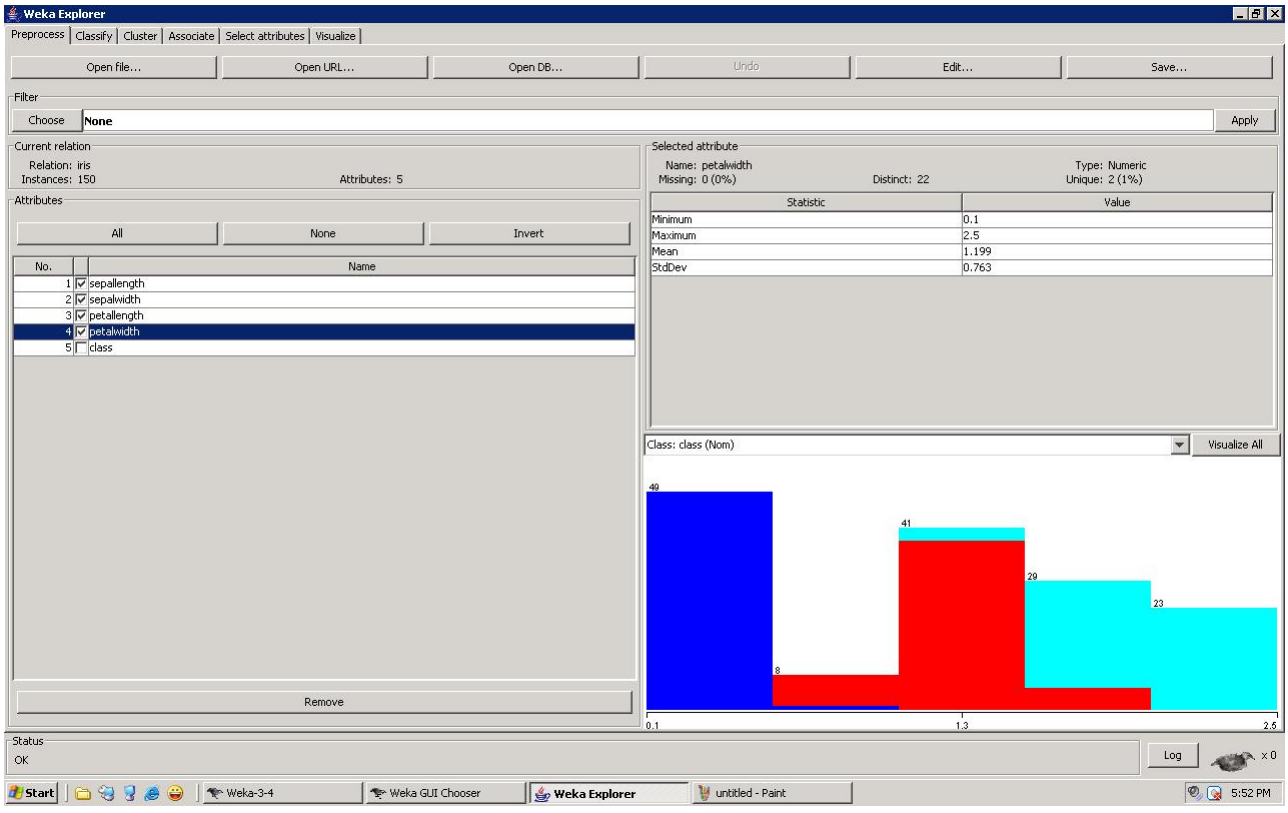

Figure 6.6 Weka Ouput for Iris Classification

The results from Weka infer that the application was 97% accurate when compared to algorithm used in SANE which was 93% accurate. The visualization from

the rule based classification of screen displayed. A comparison of this image can be made with the image in 6.4 and can be seen that all the deviation occurred in Iris-Virginica class (Class 2).

7 CONCLUSIONS AND FUTURE WORK

The primary reason for development of SANE system is to let the user access a variety of data analysis techniques without intricate knowledge in areas of Statistical analysis or neural network analysis. As far as the scope of SANE system was defined, it provides basic statistical and neural network based analysis to well defined datasets. The possible future work also depends on a similar set of norms set for the current version, namely, Automatic data analysis selection, robust data management and platform independence. Based on these norms, more statistical analyses that allow user to do statistical analysis on non Gaussian data and neural network analyses can be incorporated to cover the breadth of data analysis systems and various techniques can be incorporated into the system to allow for a rather thorough analysis of the SANE system. From this point forward, the system might also include algorithms like clustering algorithms, decision tree algorithms in it and considering system independence side, the software can be implemented completely on the client side and the data the software depends will be on might be XML kind of database system.

REFERENCES

1. Barbara G. Tabachnick, Linda S. Fidell , “Using Multivariate Statistics”, pp332-339.
2. Blonda P, Pasquariello G, Smid J, “Comparison of Back Propagation, Cascade Correlation and Kohonen Algorithms for Cloud retrieval” Proceedings of the International Conference on Neural Networks, 1993, Vol. V2, pages 1231-1234.
3. D Alpsan, M Towsey, O Ozdamar, A Tsoi, DN Ghista, “Are modified back propagation algorithms worth the effort?”, 1994. IEEE World Congress on Computational Intelligence.
4. Daniel T. Larose, “Discovering Knowledge in Data, An Introduction to Data Mining”, 2004
5. Design Expert, http://www.statease.com/
6. Hagiwara Masafumi, Sato Akira, “Analysis of momentum term in back propagation”, IEICE Transactions on Information and Systems, August-1995, V E78-D, pages 1080-1086.
7. Henderson, H. V. and Velleman, P. F. "Building Regression Models Interactively." Biometrics, 1981, 37, 391-411
8. “How many hidden units should I use”, http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-10.html
9. “Comparison between conventional computers and neural networks”, http://www-cse.stanford.edu/classes/sophomore-college/projects-00/neural-networks/Comparison/comparison.html
10. IBM Lotus 123, http:// www306.ibm.com/ software/lotus/support/ smartsuite/support.html
11. Jay Heizer, Barry Render, “Operations Management”, 6th edition, Publisher, Year, chapter 4
12. Jeremy Miles, “Data sets: From Applying Regression and Correlation”, http://www.jeremymiles.co.uk/regressionbook/data/
13. Jihoon Yang, Vasant Honavar, “Experiments with Cascade Correlation Algorithm”, Department of Computer Science, Iowa State University, 1991, Technical Report # 91-16
14. John, George H, “Derivation of a more numerically stable update rule”, IEEE International conference on neural networks, 1995, pages 1126-1129
15. Laurene Fausett, “Fundamentals of Neural Networks”, Publisher, Year
16. Liang Manjun, Shi Zhu, “Improved back propagation algorithm for neural networks”, Publisher, Year
17. Maulawka JJ, Verma BK, “Improving training time of the back propagation algorithm”, ISSN 0820-0750, 1994, pages 85-88.

18. Microsoft Excel, http://office.microsoft.com/en-us/assistance/CH010422641033.aspx
19. Minitab, “http://minitab.com/
20. Neter, Kunter et. Al., “Applied Linear Regression Models”, 3rd Edition, Publisher, Year
21. Neunet Pro, http://www.cormactech.com/neunet/index.html
22. Predict, http://www.neuralware.com/
23. R A Fisher (1936), “The use of multiple measurements in taxonomic problems”, Annals of Eugenics 7, 179-188
24. Rashpal S. Ahluwalia, Sundar Chidambaram, “An Artificial Neural Network Approach to Forecasting”, Proceedings (403), Artificial Intelligence and applications 2003
25. SAS, “http://www.sas.com/
26. SC Ng, S H Leung and A Luk, “A Generalized Back-Propagation Algorithm for Faster Convergence”, Proceedings of the 1996 IEEE International Conference on Neural Networks, ICNN Part 1(of 4), pages 409- 413.
27. SC Ng, S H Leung and A Luk, “Convergence of the Generalized Back Propagation Algorithm with constant learning rates”, Proceedings of the 1998 IEEE International Joint, Conference on Neural Networks. Part 2(of 3), pages 1090-1094.
28. Scott E Falham, Christian Lebiere, “The Cascade Correlation Learning Architecture”, August 29, 1991 CMU-CS-90-100
29. Statistica, http://www.statsoft.com/
30. Tai-Hoon Cho, Richard W Conners, Philip A Araman, “Fast back propagation learning using steep activation functions and automatic weight reinitialization”,1990
31. V V Phansalkar, P S Sastry, “Analysis of back propagation algorithm with momentum”, IEEE, Piscataway, NJ, USA, 1994, Pages 505-506
32. Weka Software - Ian H. Witten and Eibe Frank (2005) "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005

APPENDIX A – DATASET SOURCE

Analysis Type	Data Set	Reference
Bivariate/ Multiple Regression	Patient Satisfaction Data	[25]
Logistic Regression	Iris Data set	[26]
Discriminant Function Analysis	Car Mileage Data	
BPA Prediction	Patient Satisfaction Data	[25]
BPA Classification	Iris Data set	[26]
Cascade Correlation Prediction	Patient Satisfaction Data	[25]
Cascade Correlation Classification	Iris Data set	[26]

Patient Satisfaction Dataset

Factor1	Factor2	Factor3	Satisfaction Index
0.254	0.189	0.243	0.251
0.304	0.197	0.344	0.316
0.3	0.25	0.36	0.1
0.127	0.255	0.165	0.34
0.132	0.279	0.134	0.25
0.173	0.197	0.249	0.179
0.449	0.298	0.494	0.181
0.643	0.9	0.88	0.274
0.644	0.79	0.76	0.9

Iris Dataset

Sepal Length	Speal Width	Petal Length	Petal Width	Iris Type
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa
5	3.4	1.5	0.2	Iris-setosa
4.4	2.9	1.4	0.2	Iris-setosa
4.9	3.1	1.5	0.1	Iris-setosa
5.4	3.7	1.5	0.2	Iris-setosa
4.8	3.4	1.6	0.2	Iris-setosa
4.8	3	1.4	0.1	Iris-setosa
4.3	3	1.1	0.1	Iris-setosa
5.8	4	1.2	0.2	Iris-setosa
5.7	4.4	1.5	0.4	Iris-setosa
5.4	3.9	1.3	0.4	Iris-setosa
5.1	3.5	1.4	0.3	Iris-setosa
5.7	3.8	1.7	0.3	Iris-setosa
5.1	3.8	1.5	0.3	Iris-setosa
5.4	3.4	1.7	0.2	Iris-setosa
5.1	3.7	1.5	0.4	Iris-setosa
4.6	3.6	1	0.2	Iris-setosa
5.1	3.3	1.7	0.5	Iris-setosa
4.8	3.4	1.9	0.2	Iris-setosa
5	3	1.6	0.2	Iris-setosa
5	3.4	1.6	0.4	Iris-setosa
5.2	3.5	1.5	0.2	Iris-setosa
5.2	3.4	1.4	0.2	Iris-setosa
4.7	3.2	1.6	0.2	Iris-setosa
4.8	3.1	1.6	0.2	Iris-setosa
5.4	3.4	1.5	0.4	Iris-setosa
5.2	4.1	1.5	0.1	Iris-setosa
5.5	4.2	1.4	0.2	Iris-setosa
4.9	3.1	1.5	0.1	Iris-setosa
5	3.2	1.2	0.2	Iris-setosa
5.5	3.5	1.3	0.2	Iris-setosa
4.9	3.1	1.5	0.1	Iris-setosa
4.4	3	1.3	0.2	Iris-setosa
5.1	3.4	1.5	0.2	Iris-setosa
5	3.5	1.3	0.3	Iris-setosa
4.5	2.3	1.3	0.3	Iris-setosa
4.4	3.2	1.3	0.2	Iris-setosa
5	3.5	1.6	0.6	Iris-setosa

5.1	3.8	1.9	0.4	Iris-setosa
4.8	3	1.4	0.3	Iris-setosa
5.1	3.8	1.6	0.2	Iris-setosa
4.6	3.2	1.4	0.2	Iris-setosa
5.3	3.7	1.5	0.2	Iris-setosa
5	3.3	1.4	0.2	Iris-setosa
7	3.2	4.7	1.4	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
6.9	3.1	4.9	1.5	Iris-versicolor
5.5	2.3	4	1.3	Iris-versicolor
6.5	2.8	4.6	1.5	Iris-versicolor
5.7	2.8	4.5	1.3	Iris-versicolor
6.3	3.3	4.7	1.6	Iris-versicolor
4.9	2.4	3.3	1	Iris-versicolor
6.6	2.9	4.6	1.3	Iris-versicolor
5.2	2.7	3.9	1.4	Iris-versicolor
5	2	3.5	1	Iris-versicolor
5.9	3	4.2	1.5	Iris-versicolor
6	2.2	4	1	Iris-versicolor
6.1	2.9	4.7	1.4	Iris-versicolor
5.6	2.9	3.6	1.3	Iris-versicolor
6.7	3.1	4.4	1.4	Iris-versicolor
5.6	3	4.5	1.5	Iris-versicolor
5.8	2.7	4.1	1	Iris-versicolor
6.2	2.2	4.5	1.5	Iris-versicolor
5.6	2.5	3.9	1.1	Iris-versicolor
5.9	3.2	4.8	1.8	Iris-versicolor
6.1	2.8	4	1.3	Iris-versicolor
6.3	2.5	4.9	1.5	Iris-versicolor
6.1	2.8	4.7	1.2	Iris-versicolor
6.4	2.9	4.3	1.3	Iris-versicolor
6.6	3	4.4	1.4	Iris-versicolor
6.8	2.8	4.8	1.4	Iris-versicolor
6.7	3	5	1.7	Iris-versicolor
6	2.9	4.5	1.5	Iris-versicolor
5.7	2.6	3.5	1	Iris-versicolor
5.5	2.4	3.8	1.1	Iris-versicolor
5.5	2.4	3.7	1	Iris-versicolor
5.8	2.7	3.9	1.2	Iris-versicolor
6	2.7	5.1	1.6	Iris-versicolor
5.4	3	4.5	1.5	Iris-versicolor
6	3.4	4.5	1.6	Iris-versicolor
6.7	3.1	4.7	1.5	Iris-versicolor
6.3	2.3	4.4	1.3	Iris-versicolor
5.6	3	4.1	1.3	Iris-versicolor
5.5	2.5	4	1.3	Iris-versicolor
5.5	2.6	4.4	1.2	Iris-versicolor
6.1	3	4.6	1.4	Iris-versicolor

5.8	2.6	4	1.2	Iris-versicolor
5	2.3	3.3	1	Iris-versicolor
5.6	2.7	4.2	1.3	Iris-versicolor
5.7	3	4.2	1.2	Iris-versicolor
5.7	2.9	4.2	1.3	Iris-versicolor
6.2	2.9	4.3	1.3	Iris-versicolor
5.1	2.5	3	1.1	Iris-versicolor
5.7	2.8	4.1	1.3	Iris-versicolor
6.3	3.3	6	2.5	Iris-virginica
5.8	2.7	5.1	1.9	Iris-virginica
7.1	3	5.9	2.1	Iris-virginica
6.3	2.9	5.6	1.8	Iris-virginica
6.5	3	5.8	2.2	Iris-virginica
7.6	3	6.6	2.1	Iris-virginica
4.9	2.5	4.5	1.7	Iris-virginica
7.3	2.9	6.3	1.8	Iris-virginica
6.7	2.5	5.8	1.8	Iris-virginica
7.2	3.6	6.1	2.5	Iris-virginica
6.5	3.2	5.1	2	Iris-virginica
6.4	2.7	5.3	1.9	Iris-virginica
6.8	3	5.5	2.1	Iris-virginica
5.7	2.5	5	2	Iris-virginica
5.8	2.8	5.1	2.4	Iris-virginica
6.4	3.2	5.3	2.3	Iris-virginica
6.5	3	5.5	1.8	Iris-virginica
7.7	3.8	6.7	2.2	Iris-virginica
7.7	2.6	6.9	2.3	Iris-virginica
6	2.2	5	1.5	Iris-virginica
6.9	3.2	5.7	2.3	Iris-virginica
5.6	2.8	4.9	2	Iris-virginica
7.7	2.8	6.7	2	Iris-virginica
6.3	2.7	4.9	1.8	Iris-virginica
6.7	3.3	5.7	2.1	Iris-virginica
7.2	3.2	6	1.8	Iris-virginica
6.2	2.8	4.8	1.8	Iris-virginica
6.1	3	4.9	1.8	Iris-virginica
6.4	2.8	5.6	2.1	Iris-virginica
7.2	3	5.8	1.6	Iris-virginica
7.4	2.8	6.1	1.9	Iris-virginica
7.9	3.8	6.4	2	Iris-virginica
6.4	2.8	5.6	2.2	Iris-virginica
6.3	2.8	5.1	1.5	Iris-virginica
6.1	2.6	5.6	1.4	Iris-virginica
7.7	3	6.1	2.3	Iris-virginica
6.3	3.4	5.6	2.4	Iris-virginica
6.4	3.1	5.5	1.8	Iris-virginica
6	3	4.8	1.8	Iris-virginica
6.9	3.1	5.4	2.1	Iris-virginica

6.7	3.1	5.6	2.4	Iris-virginica
6.9	3.1	5.1	2.3	Iris-virginica
5.8	2.7	5.1	1.9	Iris-virginica
6.8	3.2	5.9	2.3	Iris-virginica
6.7	3.3	5.7	2.5	Iris-virginica
6.7	3	5.2	2.3	Iris-virginica
6.3	2.5	5	1.9	Iris-virginica
6.5	3	5.2	2	Iris-virginica
6.2	3.4	5.4	2.3	Iris-virginica
5.9	3	5.1	1.8	Iris-virginica

Car Mileage Dataset

Country	Car	MPG
U.S.	Buick Estate Wagon	16.9
U.S.	Ford Country Squire Wagon	15.5
U.S.	Chevy Malibu Wagon	19.2
U.S.	Chrysler LeBaron Wagon	18.5
U.S.	Chevette	30
Japan	Toyota Corona	27.5
Japan	Datsun 510	27.2
U.S.	Dodge Omni	30.9
Germany	Audi 5000	20.3
Sweden	Volvo 240 GL	17
Sweden	Saab 99 GLE	21.6
U.S.	Buick Century Special	20.6
U.S.	Mercury Zephyr	20.8
U.S.	Dodge Aspen	18.6
U.S.	AMC Concord D/L	18.1
U.S.	Chevy Caprice Classic	17
U.S.	Ford LTD	17.6
U.S.	Mercury Grand Marquis	16.5
U.S.	Dodge St Regis	18.2
U.S.	Ford Mustang 4	26.5
U.S.	Ford Mustang Ghia	21.9
Japan	Mazda GLC	34.1
Japan	Dodge Colt	35.1
U.S.	AMC Spirit	27.4
Germany	VW Scirocco	31.5
Japan	Honda Accord LX	29.5
U.S.	Buick Skylark	28.4
U.S.	Chevy Citation	28.8
U.S.	Olds Omega	26.8
U.S.	Pontiac Phoenix	33.5
U.S.	Plymouth Horizon	34.2
Japan	Datsun 210	31.8
Germany	VW Dasher	30.5
Japan	Datsun 810	22
Germany	BMW 320i	21.5
Germany	VW Rabbit	31.9

APPENDIX B – REQUIREMENT INDEX

System scope

System Purpose

The SANE System is designed to allow the user to navigate to a proper data analysis technique from the data possessed by the user along with the results he/she might wish to achieve

In scope

- Allow the user to select the appropriate data analysis technique
- Provide an environment where the user is able to execute the appropriate algorithm
- Provide storing, versioning and result comparing capabilities

Out of scope

N/A

System end-users

Any user with an access to the system

Assumptions

The user should has the data prepared for input of the system

Constraints

All algorithm will be not be applicable to very data sets. The user is required to know the kind of analysis desired.

Requirements

ID	Requirement
1.0	The system should have the appropriate navigational structure
2.0	The system should have a data base system capable of addressing the issues mentioned in the purpose
3.0	The system should have the appropriate interface to let the user obtain the results.

Proposed Functionality

The software will be designed in such a way that the user need not have extensive background in statistics or neural nets. Initially, the statistical methods will be limited to finding a relationship between independent and dependent variables, predicting group membership of a dataset, finding if the dataset is properly grouped, and determining the underlying structure of a dataset. The neural network algorithms will be limited to the back propagation algorithm and the cascade correlation algorithm.

System Architecture

Input(s) to Navigational System

Input System	Information Provided
User Interface	Input to Database and optional parameters required for data analysis
SQL Server Database	Provides SANE the data in the required format;

Output(s) from Navigational System

System	Information Provided	Output Type
SANE	Data Analysis results	Database Output and output to GUI

Testing Considerations

- Is the Navigation system capable of directing the user to appropriate algorithm?
- Is the database capable of storing and retrieving the required data in the required format?
- Are the results provided by SANE accurate within an acceptable limit?

APPENDIX C – NAVIGATIONAL REQUIREMENTS

Navigation System scope

Navigation System Purpose

- SANE should allow users to navigate to a proper data analysis technique based on the data provided by the user and the kind of data analysis required.

In scope for Navigational System

- SANE, with or without the knowledge of user, should be able to take user to appropriate data analysis technique. It should be able to direct user to an appropriate data analysis technique if he/ she is not sure of which analysis has to be performed and at the same time, SANE should allow the user to choose if he needs to have a particular data analysis to be performed.

Out of scope for Navigational System

N/A

System end-users for Navigational System

- Any user with an access to the system

Navigational System Assumptions

- The user should have the data prepared for input of the system

Constraints on Navigation System

- Not all the algorithms will be applicable to all the different data sets. The user should know what he/ she wants the result to be like

Navigational Requirements

ID	Requirement
1.1	SANE, based on problem Id, should be able to direct the user to appropriate screen for data analysis
1.2	If user chooses to apply a particular data analysis to be performed on his data set, SANE should allow the user doing that.
1.3	SANE should have an interface that allows the user visualize results from different analysis

Proposed Functionality for Navigation System

- SANE should be able to decide if the data set needs any training. This is decided by measuring the deviation of dataset from normal distribution. If the problem does not require training, the user is lead to the statistical analysis module. In the statistical analysis module, the user will have three options to choose. They are:

1. Degree of relationship: If the research question is to find the degree of relationship between two variables, the user has to select this option.
2. Analysis of groups: If the research question deals with groups, the user should select this option. This option leads to two different options and if the research question is to find the group membership, logistic regression is selected and if the research question is to find if there is any clustering in the data, discriminant function analysis is used.
3. Analysis of Structure: In this is the option is selected, the user will be asked if there is any hypothesis about the structure; meaning, if there are any suspect input variables that might be effecting the model. ANOVA is done on the data set with suspect input variables and if no satisfactory model could be concluded the user should consider neural network training.

- If the data set requires training, the user is asked about the noise level in the dataset. If the noise level is low, BPA is selected and if the noise level is higher, CCA is selected.

Navigation System - Testing Considerations

- Is the navigation system capable of directing the user to the appropriate algorithm?

Implementation of Navigation System

- This classification provides user both selection based on required result type and the analysis list. This lets user not only perform not only data analysis based on problem type, but can also compare the results with a different analysis type.

- Algorithm List
    - MultiVariate Regression
    - Logistic Regression
    - BPA Classification
    - BPA Prediction
    - CCA Classification
    - CCA Prediction
    - Data Entry Module
- Data Analysis
    - Statistical Analysis
        - Relationship between Input and output data
        - Analysis of Groups
            - Significant difference in groups
            - Predict Group Membership
        - Analysis of Structure
            - Is there a Hypothesis about Structure
                - Yes
                - No

- Neural Net Analysis
    - Classification
        - Low Noise Level
        - High Noise Level
    - Prediction
        - Low Noise Level
        - High Noise Level

APPENDIX D – INTERFACE REQUIREMENTS

Interface Module - System scope

Interface Purpose

- SANE should have an interface that is generic to all algorithms.

Interface scope

- To create an interface that allows user to manage and analyze data sets
- To create a list of Algorithms to allow the actual process of data analysis

System end-users of SANE Interface

- Any user with an access to the system

Interface Assumptions

- The user should have the data prepared for the proper input of the system

Interface Constraints

- Not all the algorithms will be applicable to all the different data sets. The user should know what he/ she wants the result to be like

Interface Requirements

ID	Requirement
3.1	SANE should have a generic user interface that needs minimal user intervention
3.2	SANE should have a data entry module incorporated into it which allows datasets to be entered without referring to the analysis type
3.3	SANE should have different statistical analysis types and Neural network analysis algorithms to address a wide range of data analysis techniques

Proposed Functionality for SANE Interface

- The navigational part of SANE should be designed in such a way that the user need not have extensive background in statistics or neural nets to find an analysis type that suits his/ her problem. The user should be guided through a set of hyperlinks that land user at an appropriate data analysis type.

Design of Interface Output(s):

Output(s)

System	Information Provided	Output Type
SANE	Data Analysis results	Database Output and output to GUI

Interface Testing Considerations

The user interface allows user to read the results in required format

The data entry module will not have any analysis type related questions built into it

The statistical analysis focuses on Degree of Relationship, Significance Group differences, Group Membership, Underlying structure of a data set

The Neural network algorithms considered are BPA and CCA

APPENDIX E – DATABASE REQUIREMENTS

Database scope

Database Purpose

- SANE should have a database support capability that lets the software have the functionality described in Requirements Index document

Database - In scope

- SANE, with or without the knowledge of user, should be able to take user to appropriate data analysis technique. It should be able to direct user to an appropriate data analysis technique if he/ she is not sure of which analysis has to be performed and at the same time, SANE should allow the user to choose if he needs to have a particular data analysis to be performed.

Database - Out of scope

N/A

System end-users for Database

- Any user with an access to the system

Database Assumptions

- The user should have the data prepared for the proper input of the system

Constraints on Database System

- Not all the algorithms will be applicable to all the different data sets. The user should know what he/ she wants the result to be like

Database Requirements

ID	Requirement
2.1	The database should let the user analyze same dataset using different analyses
2.2	The database should have a versioning capability that allows the user to play with the same data set but with different optional parameters applied at different magnitudes to it.
2.3	SANE should have an interface that allows the user visualize results from different analysis

Proposed Functionality for Database System

- SANE database should allow the user to store problem datasets, analysis type data sets along with solution files. It should also have the capability to store user data.

Database Testing Considerations

- Is a data set once stored in the database retrievable without any loss of data?
- Is the database capable of storing different solution types without any loss in the result accuracy?