Application of Data Pre-Processing Techniques for Supervised Classification

A dissertation submitted in the partial fulfilment of requirement

for the award of degree of

Master of Engineering

In

Electronics and Communication Engineering

Submitted by:

TEJ SINGH

Roll No: 801261027

Under the guidance of:

Dr. Ravi Kumar

Assistant Professor, ECED

Thapar University, Patiala

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY, Patiala

(Established under the section 3 of UGC Act, 1956)

PATIALA – 147004 (PUNJAB)

DECLARATION

I, **Tej Singh**, hereby declare that the work, which is being presented in the dissertation entitled, **"Application of Data Preprocessing Techniques for Supervised Classification"**, by me in partial fulfillment of the requirements for the award of degree of Master of Engineering in Electronics and Communication from Thapar University (Deemed University), Patiala, is an authentic record of my own work carried out under the supervision of **Dr. Ravi Kumar,** Assistant Professor, Electronics and Communication Engineering Department.

The matter presented in this thesis has not been submitted in any other University/Institute for the award of Master of Engineering.

Date: 10/07/2014

Tej Singh

Tej Singh
Roll No: 801261027

This is to certify that the above statement made by the student is correct to the best of my knowledge and belief.

Date: 10/07/2014

Ravi kumar

Dr. Ravi Kumar
(Assistant Professor)
ECED, Thapar University, Patiala

Countersigned by:

(Dr. Sanjay Sharma)
Professor and Head ECED
ECED, Thapar University, Patiala

(Dr. S. K. Mohapatra)
Dean of Academic Affairs
Thapar University, Patiala

i

ACKNOWLEDGEMENT

I would like to address my special thanks to my advisor, **Dr. Ravi Kumar,** Assistant Professor, ECED at Thapar University Patiala, for his endless and valuable supports, for his great advices and for providing me a nice opportunity to carry out my research in a pleasant working environment at Thapar University, Patiala.

I am thankful to **Dr. Sanjay Sharma**, Prof. & Head, Electronics and Communication Engineering Department, for providing us with adequate infrastructure in carrying the work.

I am also thankful to **Dr. Kulbir Singh,** P.G. Coordinator, Electronics and Communication Engineering Department for the motivation and inspiration that triggered me for the report work.

I am also thankful to all of my friends, colleagues, administrative staff and the technical staff in the institute for their cooperation in the completion of my thesis and also would like to thank all the faculty members of ECED for the full support of my work.

I am deeply grateful to my parents and my elder sister and brother, Dr. Teena Choudhary and Dr. Amit Kumar who constantly encouraged me. I am also thankful to the authors whose work have been consulted and quoted in this work.

Tej Singh

ABSTRACT

Presently, Supervised or Unsupervised clustering algorithms have exploited the benefits of weighting the instances in Effective resource allocation in dynamic environment is a growing area of interest among the researcher. In this context, these algorithms have been applied widely in field such as, image classification, noise suppression in color images, image segmentation, bioinformatics, MANETs and many more signal processing applications.

The goal of any supervised or unsupervised algorithm is to find a function that best suits a set of inputs to its correct output. However, single layer perceptron cannot learn some relatively simple patterns, such as those are not linearly separable. A multi-layered network overcomes such shortcomings called Back-Propagation Neural Network (BPNN) and it can create internal representation and learn different features in each layer.

PCA is a powerful tool for analyzing data. The main advantage of PCA is that once you have found these patterns in the data and you compress the data i.e. by reducing the number of dimensions, without much loss of information.

Domain adaptation allows knowledge from a source domain to be transferred to a different but related target domain. Transfer Cluster Analysis (TCA) tries to learn some transfer components across domains in a reproducing kernel space using maximum mean discrepancy.

Wavelet transform is fast emerging as one of the most potent tools for signal analysis. Wavelet analysis has many advantages over traditional Fourier transform based approaches.
Back-p

TABLE OF CONTENT

DECLARATION i
ACKNOWLEDGEMENT ii
ABSTRACT iii-iv
TABLE OF CONTENT v-viii
LIST OF ABBREVIATIONS ix
LIST OF FIGURES x-xi

CHAPTER 1 INTRODUCTION AND LITERATURE REVIEW 1-19

1.1 Introduction 1
1.2 Pattern Recognition 1
1.3 Pattern Recognition System 1
1.3.1 Sensing 2
1.3.2 Feature Extraction 3
1.3.3 Classification 3
1.3.4 Post Processing 3
1.4 The Pattern Design Cycle 4
1.4.1 Data selection 4
1.4.2 Feature choice 4
1.4.3 Model choice 4
1.4.4 Training 5
1.4.5 Evaluation 5
1.4.6 Computational Complexity 6
1.4.7 Learning and Adaption 6
1.4.7.1 Supervised Learning 6
1.4.7.2 Unsupervised Learning 7
1.4.7.3 Reinforcement Learning 8
1.4.8 Feed Forward Operation and Classification 8
1.4.9 Net activation function 9
1.4.10 Network learning 10
1.4.10.1 Training error 10
1.4.10.2 Learning rate 11
1.4.10.3 Learning curve 11
1.4.11 Clustering 13

1.4.11.1 Clustering and Dimensionality Reduction 14
1.5 Literature Review 14
1.6 Motivation and Objectives 18
1.7 Organization of dissertation 19

CHAPTER 2 DATA DESCRIPTION 20-24
2.1 Data mining and Knowledge Discovery 20
2.2 UCI Machine learning Repository Database 21
2.3 Iris Data set description 22
2.3.1 Scatter Plot of Iris Raw Data Set 22
2.4 User Knowledge Modeling Data Set Description 23
2.4.1 Scatter Plot of User Knowledge Modeling Raw Data Set 24

CHAPTER 3 THE BACK-PROPAGATION ALGORITHM 25-34
3.1 Multilayer Perceptron 25
3.1.1 Function Signal 26
3.1.2 Error Signal 26
3.2 Analytical Description of Back Propagation Algorithm 27
3.2.1 The Passes of Computation in BP Algorithm 30
3.2.2 The Activation Function 31
3.2.2.1 Logistic function 31
3.2.2.2 The Hyperbolic Tangent Function 31
3.2.3 Learning Rate and Momentum Constant 31
3.2.4 Modes of Training in BP Algorithm 32
3.2.4.1 Sequential Mode 32
3.2.4.2 Batch mode 32
3.2.5 Convergence Criterion of BP Algorithm 33
3.3 Error Surface 34

CHAPTER 4 CLUSTER ANALYSIS USING PCA AND TCA 35-41

PREPROCESSING TECHNIQUES

4.1 Component Analysis 35

4.2 Analytical Description of Principal Component Analysis 35

4.2.1 PCA preprocessing on data signal 39

4.3 Transform Cluster Analysis 40

4.3.1 TCA Preprocessing on Data Signal 41

CHAPTER 5 WAVELET TRANSFORM AS A PREPROCESSING TECHNIQUE 42-48

5.1 The Fourier Transform 42

5.2 The Short Time Fourier Transform 43

5.3 The Wavelet 44

5.3.1 The Continuous Wavelet 44

5.3.2 The Continuous Wavelet Synthesis 45

5.4 Wavelet Families 46

5.4.1 Daubechices Wavelets: dbN 47

5.5 Scatter Plots of Wavelet Transform 48

CHAPTER 6 RESULT AND DISCUSSION 49-61

6.0 Description of Box Plot 49

6.1 Classification Performance with of Raw Iris Data 49

6.1.1 Classification Performance of ANN Classifier with Iris Data 50

6.1.2 Classification Performance with Principle Component Analysis (PCA) of Iris Data 51

6.1.3 Classification Performance with Principle Component Analysis (PCA) of Iris Data 52

6.2 Classification Performance with Raw User Knowledge Student Modeling Data 53

6.2.1 Classification Performance of ANN Classifier with User Knowledge Student Modeling Data 53

6.2.2 Classification Performance with Principle Component Analysis (PCA) of User knowledge Student Modeling Data 54
6.2.3 Classification Performance with Transfer Component Analysis (TCA) of User knowledge Student Modeling Data 55
6.3 Classification Performance with Wavelet Transform of Iris Data 56
6.3.1 Classification Performance with Wavelet Transform with PCA of IRIS Data 57
6.3.2 Classification Performance with Wavelet Transform with TCA of IRIS Data 58
6.4 Classification Performance with Wavelet Transform of User Knowledge Student Modeling Data 59
6.4.1 Classification Performance with Wavelet Transform with PCA of User Knowledge Student Modeling Data 60
6.4.2 Classification Performance with Wavelet Transform with TCA of User Knowledge Student Modeling Data 61

CHAPTER 7 CONCLUSION AND FUTURE SCOPE 62
7.1 Conclusion 62
7.2 Future Scope 62
REFERENCE 63

LIST OF ABBREVATIONS

ANN	Artificial Neural Network
BP	Back Propagation Algorithm
BPNN	Back Propagation Neural Network
CWT	Continuous Wavelet Transform
DBMS	Database Management Systems
FT	Fourier Transform
KDD	Knowledge Data Discovery
LR	Learning Rate
MC	Momentum Constant
MLP	Multilayer Perceptron
PCA	Principal Component Analysis
STFT	Short Time Fourier Transform
TCA	Transfer Cluster Analysis

LIST OF FIGURES

Fig. 1.1 Pattern Recognition System 2
Fig. 1.2 The Design Cycle of Pattern Recognition System 5
Fig. 1.3 Simple Model of Machine Learning 6
Fig. 1.4 Block Diagram of Supervised Learning 7
Fig. 1.5 Block Diagram of Unsupervised Learning 7
Fig. 1.6 A Simple Three Layer Neural Network 8
Fig. 1.7 Learning Curve And Its Iteration With Validation And Test Sets 12
Fig. 1.8 Block Diagram of Clustering Procedure 13
Fig. 2.1 Graphical Representation of Data Mining And KDD 20
Fig. 2.2 Scatter Plot of Iris Raw Data 22
Fig. 2.3 Scattered Plot of Raw User Knowledge Modeling Data 24
Fig. 3.1 Architectural Graph of A Multilayer Perceptron With Two Hidden Layers 25
Fig. 4.1 Scatter Plot of Iris Data With PCA 39
Fig. 4.2 Scatter Plot of Student Modeling Data With PCA 39
Fig. 4.3 Scatter Plot of Iris Data With TCA 41
Fig. 4.4 Scatter Plot of Student Modeling Data With TCA 41
Fig. 5.1 Scatter Plot of Wavelet Transform of Iris Data 48
Fig. 5.2 Scatter Plot of Wavelet Transform of Student Modeling Data 48
Fig. 6.1 Performance Box Plot of Iris Raw Data 50
Fig. 6.2 Performance Box Plot of Iris Data With PCA 51
Fig. 6.3 Performance Box Plot of Iris Data With TCA 52
Fig. 6.4 Performance Box Plot of Student Modeling Raw Data 53
Fig. 6.5 Performance Box Plot of Student Modeling Data With PCA 54
Fig. 6.6 Performance Box Plot of Student Modeling Data With TCA 55
Fig. 6.7 Performance Box Plot of Wavelet Transform of Iris Data 56
Fig. 6.8 Performance Box Plot of Wavelet Transform With PCA of Iris Data 57
Fig. 6.9 Performance Box Plot of Wavelet Transform With TCA of Iris Data 58
Fig. 6.10 Performance Box Plot of Wavelet Transform of Student Modeling Data 59
Fig. 6.11 Performance Box Plot of Wavelet Transform With PCA of Student

Modeling Data 60
Fig. 6.12 Performance Box Plot of Wavelet Transform With TCA of Student Modeling Data 61

CHAPTER 1

INTRODUCTION AND LITERATURE REVIEW

1.1 Introduction

Everyday people encounter a large amount of information and store or represent it as data for further analysis and management. One of the vital means in dealing with these data is to classify or group them into a set of categories or clusters. Actually, as one of the most primitive activities of human beings classification plays an important and indispensable role in the long history of human development. In order to learn a new object or understand a new phenomenon, people always try to seek the features that can describe it and further compare it with other known objects or phenomena based on the similarity or dissimilarity, generalized as proximity, according to some certain standards or rules. Basically, classification systems are either supervised or unsupervised depending on whether they assign new inputs to one of a finite number of discrete supervised classes or unsupervised categories, respectively. The goal of clustering is to separate a finite unlabeled data set into a finite and discrete set of natural hidden data structures rather than provide an accurate characterization of unobserved samples generated from the same probability distribution. This can make the task of clustering fall outside of the framework of unsupervised predictive learning problems such as vector quantization, probability density function estimation and entropy maximization. It is noteworthy that clustering differs from multidimensional scaling whose goal is to depict all the evaluated objects in a way that minimizes the topographical distortion while using as few dimensions as possible. [16]

1.2 Pattern Recognition

The ease with which one can recognize a face, understand spoken words, read handwritten characters, identify car keys in pocket by feel and decide whether an apple is ripe by its smell belies the astoundingly complex processes that underlie these acts of pattern recognition. Pattern recognition is the act of taking raw data and

making an action based on the category of the pattern has been crucial for our survival and over the past tens of millions of years it have evolved highly sophisticated neural and cognitive system for such tasks.[30]

It is natural to design and build machine that can recognize patterns. From automated speech recognition, fingerprint identification, optical character recognition, DNA sequence identification and much more. It is clear that reliable, accurate pattern recognition by machine would be immensely useful.[12]

1.3 Pattern Recognition System

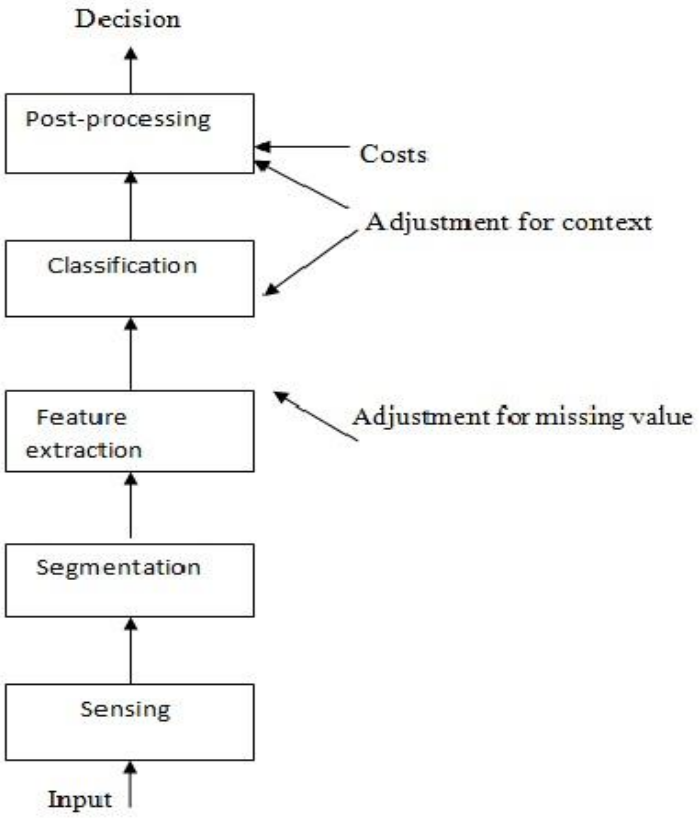

Fig.1.1 Pattern Recognition System

1.3.1 Sensing

The input to pattern recognition system is often some kind of a transducer such as a camera or a microphone array.

1.3.2 Feature Extraction

The traditional goal of the feature extractor is to characterize an object to be recognized by measurement whose values are similar for objects in the same category and very different for objects in different categories. This leads to the idea of seeking distinguishing feature that are invariant to irrelevant transformations of the input.[8]

1.3.3 Classification

The task of the classifier component proper of a full system is to use the feature vector provided by the feature extractor to assign the object to a category. Because perfect classification performance is often impossible, a more general task is to determine the probability for each of the possible categories. The abstraction provided by the feature vector representation of the input data enables the development of a largely domain independent theory of classification. The degree of difficulty of the classification problem depends on the variability in the feature values for object in the same category relative to the difference between feature values for objects in different categories.[30]

1.3.4 Post Processing

The post-processor uses the output of the classifier to decide on the recommended action. The simplest measure of classifier performance is the classification error rate i.e., the percentage of new pattern that are assigned to the wrong category. Thus is required to minimum error rate classification. The post-processor are able to exploit context i.e., input-dependent information other than from the target pattern itself to improve system performance.[3]

1.4 The Pattern Design Cycle

The design of a pattern recognition system usually entails the repetitions of a number of different activities such as data collection, feature choice, model choice, training and evaluation. Fig 1.2 shows typical design cycle for a pattern recognition system.[30][31]

1.4.1 Data Collection

Data collection can account for surprisingly large part of the cost of development of a pattern recognition system. It may be possible to perform a preliminary feasibility study with a small set of examples but much more data will usually be needed to assure good performance in the fielded system. [1]

1.4.2 Feature Choice

The choice of the distinguishing feature is a critical design step and depends on the characteristics of the problem domain. In selecting or designing features, it is obviously would like to find features that are simple to extract, invariant to irrelevant transformations, insensitive to noise and useful for discriminating patterns in different categories.[8]

1.4.3 Model Choice

Sometimes the performance of our classifier might be unsatisfied and thus jumped to an entirely different class of model. When a hypothesized model differs significantly from the true model underlying our patterns and thus a new model is needed. [30]

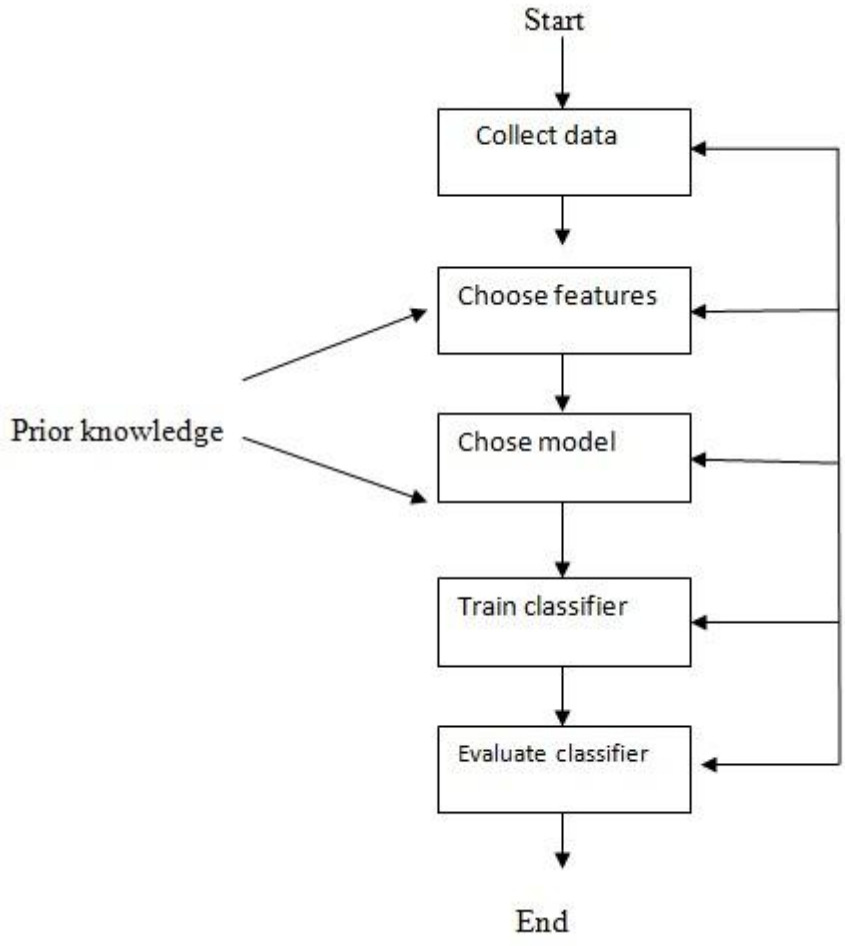

Fig.1.2 The Design Cycle of Pattern Recognition System

1.4.4 Training

In general, the process of using data to determine the classifier is referred to as training the classifier.

1.4.5 Evaluation

Evaluation is important both to measure the performance of the system and to identify the need for improvements in its components. When an overlay complex system may allow perfect classification of the training samples, it is unlikely perform well on new patterns. This situation is known as over-fitting.[28]

1.4.6 Computational Complexity

In more general term, it might be ask how an algorithm scales as a function of the number of feature dimensions or the number of categories. In some problem, it is possible to design an excellent recognizer but not with within the engineering constraints. Typically, algorithms less concerned with the complexity of learning than with the complexity of making a decision. While computational complexity generally correlates with the complexity of the hypothesized model of the patterns, these two notions are conceptually different. [5]

1.4.7 Learning and Adaptation

In the broad sense any method that incorporates information from training samples in the design of a classifier employs learning. Learning refers to some form of algorithm for reducing the error on a set of training data. A range of gradient descent algorithm that changes a classifier's parameter in order to reduce an error measure now permeate the field of the statistical pattern recognition. [6]

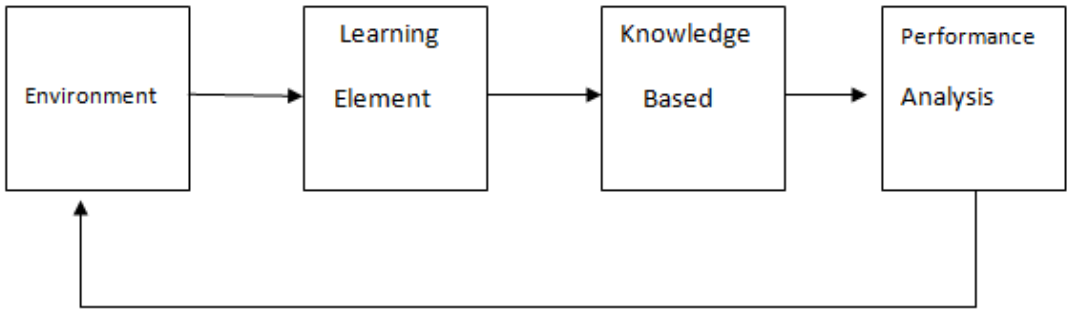

Fig.1.3 Simple Model of Machine Learning

Learning comes in several general forms:

1.4.7.1 Supervised Learning

In supervised learning, a teacher provides a category label or cost for each pattern in a training set and seeks to reduce the sum of the costs for these patterns.[27]

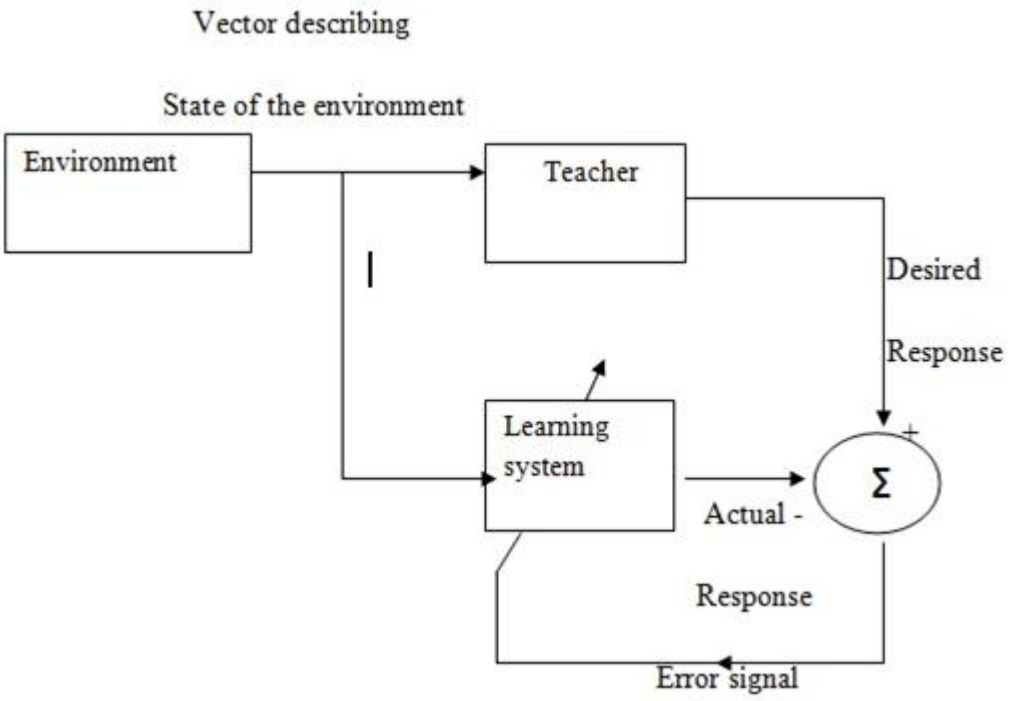

Fig. 1.4 Block Diagram of Supervised Learning

1.4.7.2 Unsupervised Learning

In unsupervised learning or clustering there is no explicit teacher and the system forms clusters or natural grouping of the input patterns. Natural is always defined explicitly or implicitly in the clustering system itself and given a particular set of patterns or cost function, different clustering algorithms lead to different clusters.[28]

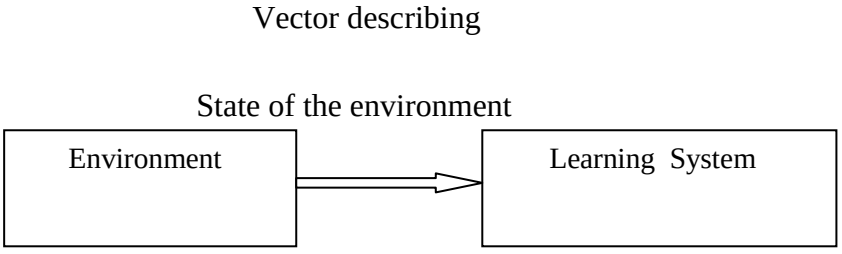

Fig.1.5 Block Diagram of Unsupervised Learning

1.4.7.3 Reinforcement Learning

In reinforcement learning or learning with critic, no desired category signal is given instead the only feedback is that the tentative category is right or wrong. This is analogous to a critic who is merely states that something is right or wrong but does not say specifically how it is wrong. In pattern classification, it is most common that such reinforcement is binary either the tentative decision is correct or it is not. [30]

1.4.8 Feedforward Operation and Classification

Feed forward network consists of an input layer, a hidden layer and an output layer, interconnected by modifiable weight, represented by links between layers. Furthermore, a single bias unit that is connected to each unit other than input units. The function of units is loosely based on properties of biological neurons, and hence called "neurons". Typically, for pattern recognition, where the input units represents the components of a feature vector and where signals emitted by output units will be the values of the discriminant functions used for classification.[30]

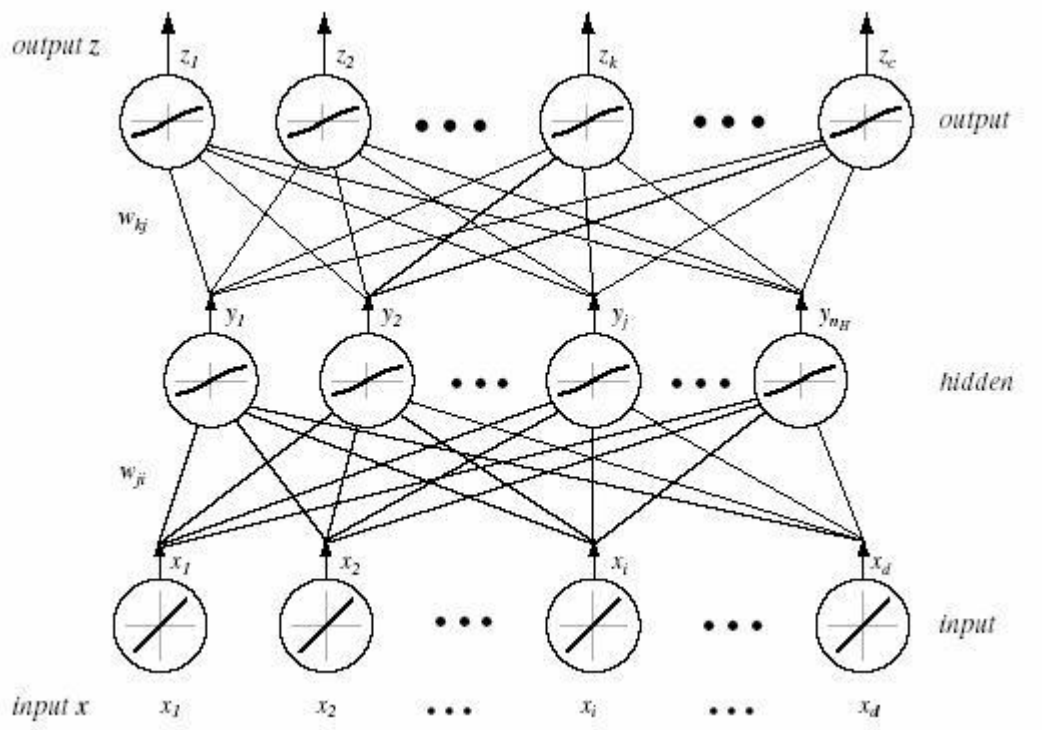

Fig.1.6 A Simple Three Layer Neural Network

1.4.9 Net Activation Function

Each hidden unit computes the weighted sum of its inputs to form its scalar net activation function which we called *net*. It can be expressed as

$$net_j = \sum_{i=1}^{d} x_i w_{ji} + w_{j0} = \sum_{i=0}^{d} x_i w_{ji} \equiv \boldsymbol{w_j^t X} \tag{1.1}$$

where the subscript *i* indexes units in the input layer, *j* in the hidden unit; w_{ji} denote the input-to-hidden layer weights at the hidden unit *j*.

Each hidden unit emits an output that is a nonlinear function of its activation, $f(net_j)$ is given by,

$$y_j = f(net_j) \tag{1.2}$$

A simple threshold or signum function

$$f(net_j) = Sgn(net) = \begin{cases} -1, & if\ net < 0 \\ 1, & if\ net \geq 0 \end{cases} \tag{1.3}$$

This $f(.)$ is sometimes called the activation function or nonlinearity of a unit.

Each output unit similarly computes its net activation based on the hidden unit signals as,

$$net_k = \sum_{j=1}^{n_H} y_j w_{kj} + w_{k0} = \sum_{j=0}^{n_H} y_j w_{kj} \equiv \boldsymbol{w_k^t y} \tag{1.4}$$

where the subscript k indexes units in the output layer and n_H denote the number of hidden units. The bias unit is equal to one of the hidden units whose output is always $y_0 = 1$.

An output unit computes the nonlinear function of its *net*,

$$z_k = f(net_j) \tag{1.5}$$

Clearly, the output z_k , also be as a function of the input feature vector **x**. when there are *c* output units, we can consider the network as computing c discriminant function $z_k = g_k(x)$ and classify the input according to which discrininant function is largest.[30]

1.4.10 Network Learning

The basic approach in learning is to start with an untrained network, present a training pattern to the input layer, pass the signals through the net and determine the output pattern to the output layer. Here these outputs are compares to the target values; any difference corresponding to an error. The error or criterion function is some scalar function of the weights and is minimized when the network outputs match the desired outputs.[7] Thus weights are adjusted to reduce this error. The main problem of learning is that the structures of multidimensional pattern from a set of unlabeled data samples. On viewing geometrically, these samples may form a cloud of points in a d-dimensional space. It is consider that these points came from a single normal distribution and then it could be learn from the data would be contained in the sufficient statistics i.e., the sample mean and the sample variance matrix. In essence these statistics constitute a compact description of the data. The sample mean locate the center of gravity of the cloud and it can be thought of as the single point mean that best represents all of the data in the sense of minimizing the sum of squared distances from mean to the samples. The sample covariance matrix describes the amount the data scatters along various directions around mean. If the data points are actually normally distributed, then the cloud has a simple hyper ellipsoidal shapes and the mean tends to fall in the region where the samples are most densely concentrated. If the samples are not normally distributed these statistics can give a very misleading description of the data. Furthermore, in situations where it have relatively poor knowledge of nature of the data, the assumption of particular parametric forms may goes to poor or meaningless results. Instead of finding the structure in the data we would be imposing structure on it. [33]

1.4.10.1 Training Error

The training error on a pattern to be the sum over output units of the squared differences between the desired output t_k given by a teacher and the actual output z_k,

$$J(w) \equiv \frac{1}{2}\sum_{k=1}^{c}(t_k - z_k)^2 = \frac{1}{2}\|t - z\|^2 \qquad (1.6)$$

where **t** and **z** are the target and the network output vectors of length c and **w** represents all the weights in the network. [30]

1.4.10.2 Learning Rate

The back propagation learning rule is based on gradient descent. The weight are initialized with random values, and then they are changed in a direction that will reduce the error,

$$\Delta w = -\eta \frac{\partial J}{\partial w} \tag{1.7}$$

$$\Delta w_{pq} = -\eta \frac{\partial J}{\partial w_{pq}} \tag{1.8}$$

where η is the learning rate and merely indicates the relative size of the change in weights. It is clear from Eq., (1.8) that the criterion function can never be negative. Furthermore, the learning rule guarantees that learning will stop except in some cases. This iterative algorithm requires taking a weight vector at iteration *n* and updating it as

$$w(n+1) = w(n) + \Delta w(n) \tag{1.9}$$

where *n* index the particular pattern presentation.[28]

1.4.10.3 Learning Curve

Before training has begun, the error on the training set typically set high; through learning the error becomes lower, as shown in a learning curves. The training (par pattern) training error ultimately reaches an asymptotic value which depends upon the Bayes error, the amount of training data, and the expressive power in the network. The higher the Bayes error and fewer the number of such weights, the higher this asymptotic value is likely to be. Because the batch back propagation performs gradient descent in the criterion function, if the learning rate is not high the training error tends to decrease monotonically. The average error on an independent test set is virtually always higher than on the training set and while it generally decreases, it can increase or oscillate.[4]

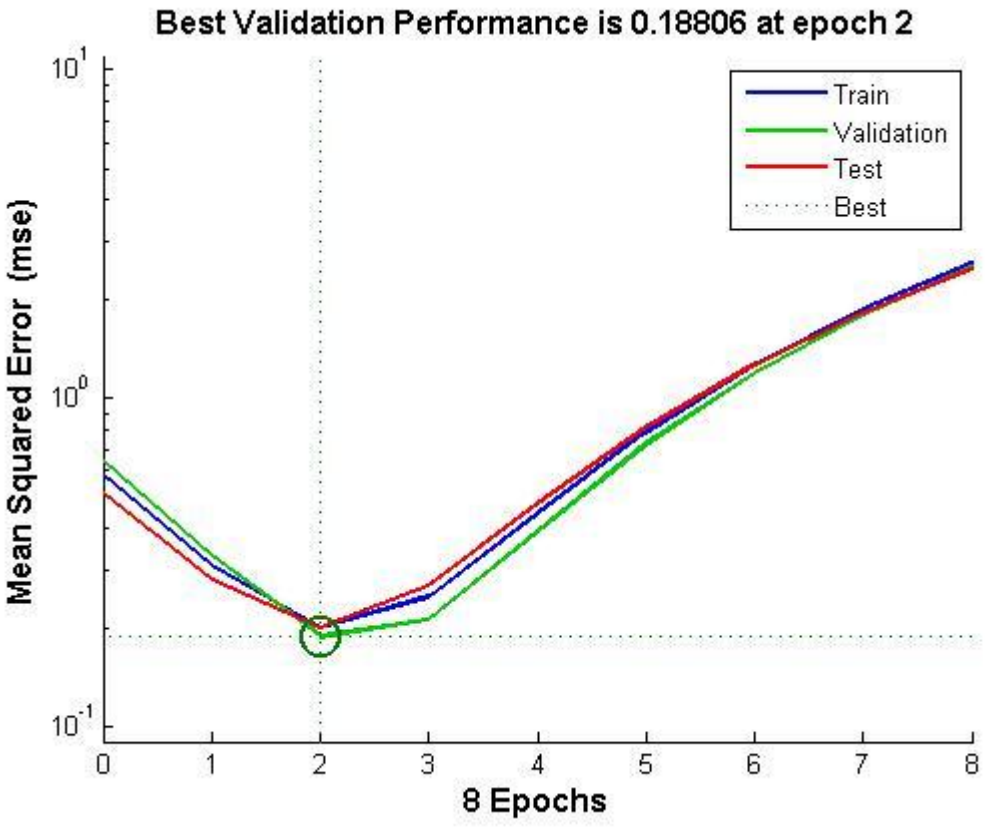

Fig.1.7 Learning Curve and Its Iteration with Validation and Test Sets.

In addition to the use of the training set, there are two conceptual uses of independently selected pattern. One is to state the performance of the fielded network, for this we use the test set. Another is used to decide when to stop training; for this we use validation set.[5]

1.4.11 Clustering

Clustering is an approach that attempts to assess the relationships in a data set by organizing the data patterns into different groups such that data patterns within a group are more similar to one another than those belonging to different groups. Clustering techniques can be classified into supervised and unsupervised methods. The unsupervised clustering method is to detect the underlying structure in the data set for classification, pattern recognition, and model reduction and optimization while the supervised clustering method is usually involved with the human interaction. Clustering is often called an unsupervised learning task as no class values denoting an a priori grouping of the data instances are given, which is the case in supervised learning.[12]

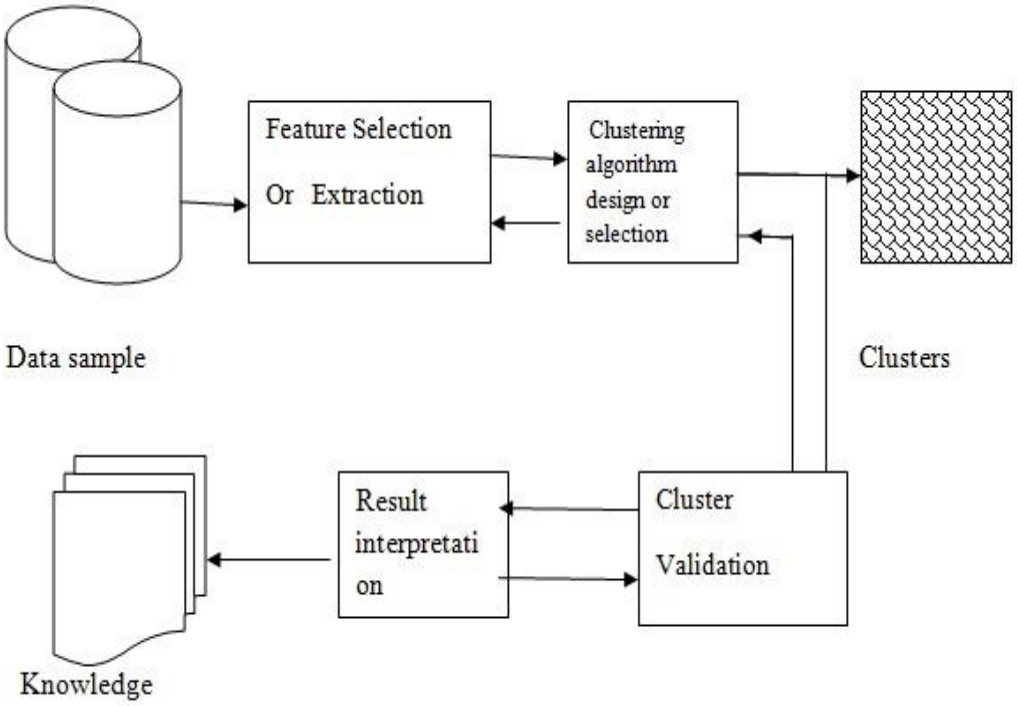

Fig. 1.8 Block Diagram of Clustering Procedure

1.4.11.1 Clustering and Dimensionality Reduction

Due to high dimensionality plagues, a variety of methods for dimensionality reduction have been proposed for pattern recognition problems. These methods provide a functional mapping so that one can determine the image of an arbitrary feature vector. The classical procedures of statistics are principal components and factor analysis. Both of which reduce the dimensionality by forming linear combinations of the features. The object of principal component analysis is to find a lower- dimensional representation that account for the variance of the features. The object of factor analysis is to find a lower dimensional representation that account for the correlations among the features. It can be consider the problem as one of removing or combining highly correlated features then it becomes clear that the techniques of clustering are applicable to this problem.[7][12]

In terms of data matrix, whose *n* rows are the *d*- dimensional samples, ordinary clustering can be thought of as a grouping of the rows, with smaller number of clustering centers being used to represents data, whereas dimensionality reduction can be seen as a grouping of columns, with combined features being used to represent the data. For the purposes of pattern classification, the most serious criticism of all of the approaches to dimensionality reduction that it is overly concerned with faithful representation of the data.[12]

1.5 Literature Review

H. Tolga Kahraman et al., In this paper, a powerful, efficient and simple 'Intuitive Knowledge Classifier' method successfully and proposes to model the domain dependent data of users. A domain independent object model, the user modeling approach and the weight-tuning method were combined with instance-based classification algorithm to implement prove classification performance of well-known the Bayes and the k-nearest neighbor-based methods. The proposed method consist of integration of the domain independent object model, the robust and the effective user modeling approach of AEEC, the instance –based classification algorithm and a novel weight-tuning method.[1]

Jiang R. et al., In this paper, a novel sparse PCA methods to perform anomaly detection and localization for network data stream has been proposed. It can localize anomalies by identifying a sparse low-dimensional space that captures the abnormal event in data streams. To better capture the source of anomalies, the structure information of the network stream data in anomaly localization framework. [2]

Kumar. R et al., In this paper, a neuro-wavelet classifier for identification of odor and gases has been discussed. The proposed approach involves the generation and selection of an optimal set of wavelet coefficients that have better class separability than the raw data. Principal component analysis (PCA) has been used to select the coefficients which contribute most to the variance of the coefficient matrix generated by continuous wavelet transform of the original signal at different scales. The selected coefficients were used as inputs to neural classifiers.[3]

Chayaporn Kaensar, In this paper, the effect of input parameters on BPNN with three different structures including simple back propagation, Back Propagation with momentum and back propagation using conjugate gradient descent methods has been discussed.This paper determined different parameters such as learning rate, momentum constant or even the number of units in the hidden layer that exist in each structure. The simple algorithm obtained high recognition rate but it needed to increase the learning rate while Back Propagation using conjugates gradient descent could provide high result in case of improving hidden nodes.[4]

P. Sinno and Yang.Q, In this paper, several current trends of transfer learning. Transfer learning is classified to three different settings: inductive transfer learning, trans-inductive transfer learning, and unsupervised transfer learning has been discussed. Most previous works focused on the former two setting. It included the instance-transfer approach, the feature-representation-transfer approach, the parameter-transfer approach, and the relational-knowledge transfer approach, respectively. Most of these approaches assume that the selected source domain is related to the target domain. [5]

Pan. J.S et al., In this paper, a novel feature extraction method, TCA, for domain adaptation has been discussed. It learns a set of transfer components such that when projecting domain data onto the latent space spanned by the transfer component, the

distance between domains can be reduced. In order to capture the label dependence in transfer component learning, a semi supervised feature extraction method i.e., SSTCA, which can reduce the distance in data distribution between domains and maximize label dependence in a de noised latent space instead of the original feature space. [6]

Baldi. P and Hornik .K, In this paper, the problem of learning from examples in layered linear feed-forward neural networks using optimization methods, such as back propagation, with respect to the usual quadratic error function E of the connection weights has been discussed. It result shows a complete description of the landscape attached to E in terms of principle component analysis. It also show that E has a unique minimum corresponding to the projection onto the subspace generated by the first principal vectors of a covariance matrix associated with the training patterns.[7]

Pittner.S and Sagar V. Kamarthi, In this paper, a new efficient feature extraction method based on the wavelet transform is presented. This paper specially deals with the assessment of process parameters or states in a given application using the features extracted from the wavelet coefficient of measured process signals. A preprocessing routine that computes robust features correlated to the process parameter of interest is highly desirable. The study investigates a preprocessing method for reducing the size of input signal presented to neural network for increasing the estimation or classification accuracy but retaining most of the intrinsic information content of the measured signals.[8]

C. M. Lee et al., In this paper, the MLP can be divided from the hidden layer into two sub- MLPs and each sub-MLP is optimized by its own BP algorithm. The modified BP algorithm indeed improves the typical BP algorithm especially for an environment with nonlinear distortion, frequency offset and phase and timing error. The computation complexity of the proposed algorithm almost equal that of the conventional BP algorithm.[10]

Bianchi .M et al., In this paper, a novel and efficient algorithm for content-based image retrieval based on Discrete Wavelet Transform and Principal Component Analysis together with inputs drawn from Euclidian operator, a common criterion to measure distance among matrices has been discussed. The former is used to produced

a signature from the query input image, a compressed and codified matrix that holds the key feature of the original data, and the latter is used to obtain the projections of the original data onto particular subspaces. The system's input consists of a query image and its output corresponds to the most similar image found in the data –base, according to the distance criterion adopted.[11]

Phansalkar V.V and Sastry P.S, In this paper, the Back Propagation (BP) with the momentum term is analyzed. It shows that all local minima of the sum of least squares error are stable. Other equilibrium points are unstable. It is also shows that if the momentum term is negative, the speed of convergence goes down. This analysis does not prove that BP with momentum will converges to one of the local minima and BP and BPM have essentially the same behavior over a finite time interval. [11]

Nielsen, F. and Nock, R. In this paper, an iterative unsupervised learning have emphasized a new trend in clustering has been discussed. It basically consists of penalizing solutions via weights on the instance points, somehow making clustering move toward the hardest points to cluster. The motivations come principally from an analogy with powerful supervised classification methods known as boosting algorithms.[12]

Bai, X. *et al.*, In this paper, the problem of feature weight learning for image clustering has been addressed. It normalized all data features between 0 and 1, because it is not possible to determine which features are more important. In this paper, a feature weight learning framework is provides for clustering which can obtain the feature weights and cluster labels simultaneously. An alternative optimization algorithm is adopted to solve this problem. Empirical studies on the toy data and real image data demonstrate that algorithm's effectiveness in improving the clustering performance.[13]

Gu. L. *et al.*, In this paper, they proposed that the semi-supervised clustering can take advantage of some labeled data also called seeds to affect the clustering of unlabeled data A semi supervised clustering method based on a locality-weight fuzzy c-means clustering algorithm. The presented clustering method uses some seeds for the initialization and applies one novel decision rule to reassigning the class label to one data. Experimental results show that proposed method can improve the clustering

performance significantly compared to some unsupervised and semi-supervised clustering algorithms. [14]

Hien. L. *et al.*, In this paper, the feature space structuring methods play a very important role in finding information in large image databases has been discussed. It has organized indexed images in order to facilitate, accelerate and improve the results of further retrieval. It has been presented both formal and experimental comparisons of different unsupervised clustering methods for structuring large image databases and different image databases of increasing sizes to study the scalability of the different approaches. [15]

1.6 Motivation and Objectives

Based on the literature review presented above it is clear that there is a lot of scope to study classification performance of popular classifiers using different preprocessing techniques and several benchmark data sets. This has served as the primary motivation for the author to undertake this work keeping in mind the following objectives:

1. To study the class separability of benchmark data using visualization techniques e.g. scatter plots etc.

2. To subject the data to several preprocessing stages and assess the effect of preprocessing on the class separability.

3. To classify the data using neural classifiers trained with Back Propagation algorithm.

4. To convert the data into frequency domain samples and assess its class separability.

1.7 Organization of Dissertation

This dissertation is organized in the following six chapters.

Chapter 1 : Introduction and Literature View

Chapter 2 : This section gives the overview of the benchmark data used in the course of study. The data are selected in such a manner that the statistical features represents different dimensions and number of samples obtained from real world problem.

Chapter 3 : In this chapter, introduced Back Propagation (BP) algorithm since the BP trains ANN has been used as a primary classifier for every data set used in this study.

Chapter 4 : The next chapter describes in details two highly effective preprocessing technique viz. Principle Component Analysis (PCA) and Transform Cluster Analysis (TCA). The Raw data and the preprocess data has been visualize scatter plot to assess the efficacy of the technique. The other claiming the novelty in the introduction of TCA to the best of his knowledge.

Chapter 5 : The fifth chapter analyze the same data in the frequency domain using wavelet transform as a preprocessing tools. The wavelet coefficient have sub sequential been fed tool and ANN classifier just like in the case of time domain data.

Chapter 6 : Both training and test data results to assess the classification performance.

Chapter 7 : Based upon six chapter the last chapter draws important conclusion and outlines the future scope.

CHPATER 2

DATA DESCRIPTION

2.1 Data Mining And Knowledge Discovery

Data Mining also popularly known as *Knowledge Discovery in Databases* (KDD), refers to the nontrivial extraction of implicit, previously unknown and potentially useful information from data in databases. While data mining and knowledge discovery in databases (KDD) are frequently treated as synonyms, data mining is actually part of the knowledge discovery process.[7][13][28]

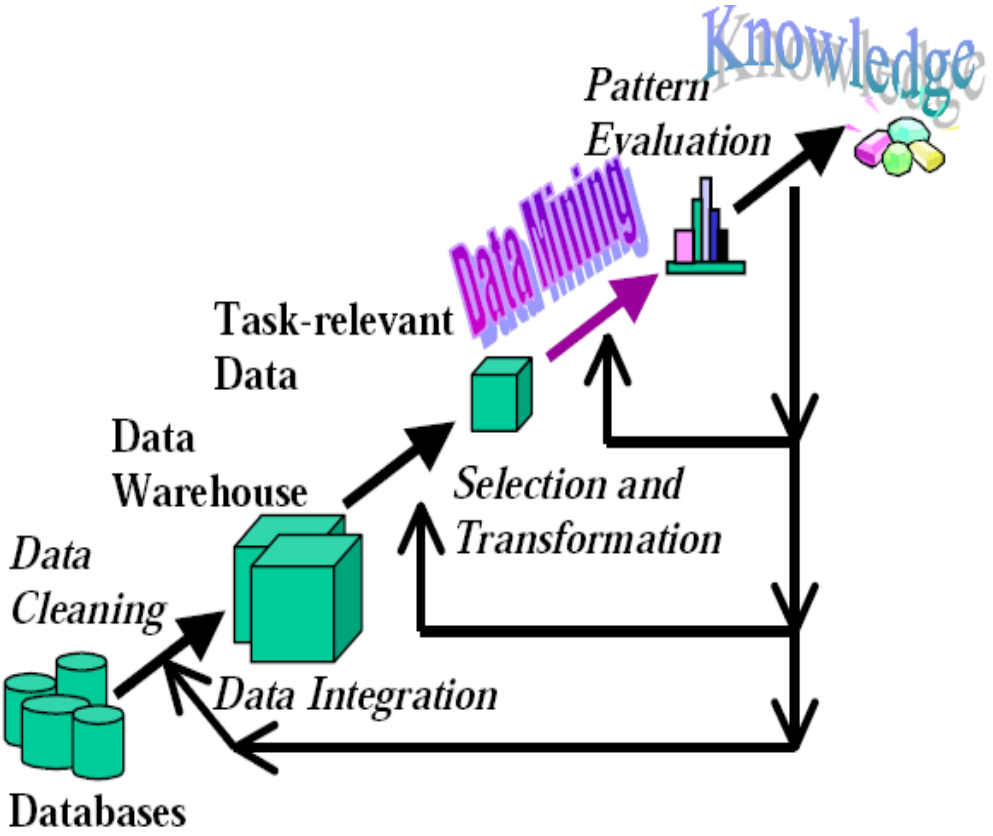

Fig. 2.1 Graphical Representation of Data Mining and KDD

The above figure shows data mining as a step in an iterative knowledge discovery process.

The Knowledge Discovery in Databases process comprises of a few steps leading from raw data collections to some form of new knowledge. The iterative process consists of the following steps,

- **Data cleaning**: also known as data cleansing, it is a phase in which noise data and irrelevant data are removed from the collection.
- **Data integration**: at this stage, multiple data sources, often heterogeneous, may be combined in a common source.
- **Data selection**: at this step, the data relevant to the analysis is decided on and retrieved from the data collection.
- **Data transformation**: also known as data consolidation, it is a phase in which the selected data is transformed into forms appropriate for the mining procedure.
- **Data mining**: it is the crucial step in which clever techniques are applied to extract patterns potentially useful.
- **Pattern evaluation**: In this step, strictly interesting patterns representing knowledge are identified based on given measures.

Knowledge representation: this is the final phase in which the discovered knowledge is visually represented to the user. This essential step uses visualization techniques to help users understand and interpret the data mining results. [1][8][31]

2.2 UCI Machine Learning Repository Database

The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. The archive was created as an ftp archive in 1987 by David Aha and fellow graduate students at UC Irvine. Since that time, it has been widely used by students, educators, and researchers all over the world as a primary source of machine learning data sets. The current version of the web site was designed in 2007 by Arthur Asuncion and David Newman, and this project is in collaboration with the University of Massachusetts Amherst. Funding support from the National Science Foundation is gratefully acknowledged. [35]

2.3 Iris Data Set Descriptions

This is perhaps the best known database to be found in the pattern recognition literature. Fisher's paper is a classic in the field and is referenced frequently to this day. One class is linearly separable from the other two, the latter are not linearly separable from each other and contained no missing value. The data set contains three classes named as iris setosa, iris versicolour and iris virginica of fifty instances each, where each class refers to a type of iris plant. This data set have multivariate characteristics and it belongs to area to a real life. This data set has four number of attributes sepal length, sepal width, petal length, and petal width. All the attributes length information are measure in centimeters.[35]

2.3.1 Scattered Plot of Iris Raw Data Set

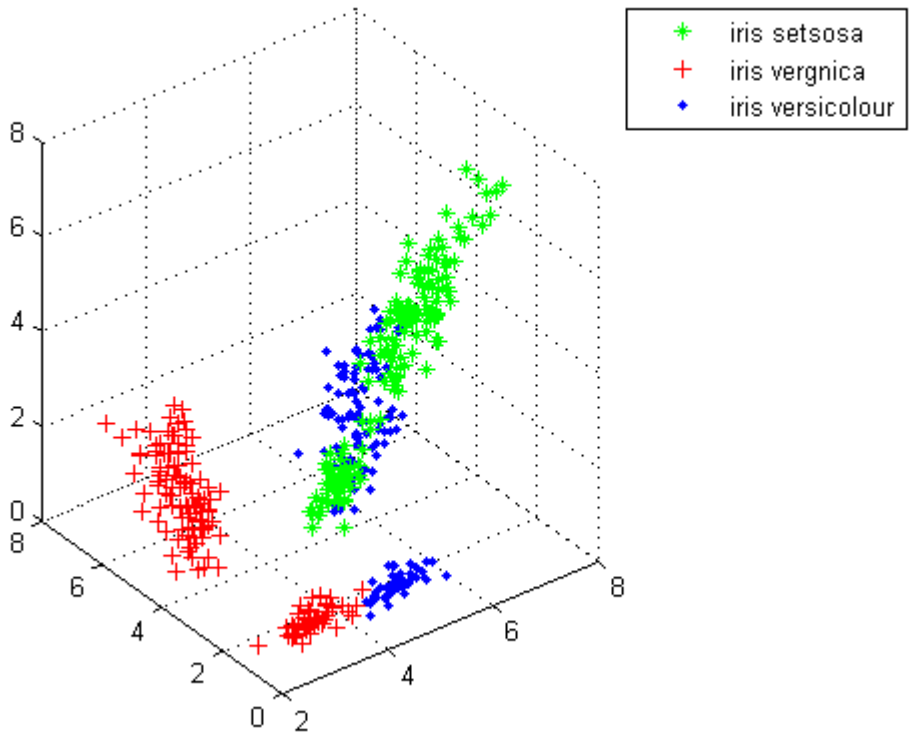

Fig 2.2 Scatter Plot of Iris Raw Data

2.4 User Knowledge Modeling Data Set Description

It is a real data set about the student's knowledge status about the subject of electrical dc machines. The user's knowledge class was classified by the author using intuitive knowledge classifier k-nearest neighbor algorithm. This data set has multivariate characteristics. This data set contains total number data samples 403, which have four type of classes named as Very Low(50), Low(129), Medium(122) and High(130). There is no missing value in the data set and it belongs to area computer . This data set has five attributes given as: [35]

1. STG : The degree of study time for goal object materials.

2. SCG : The degree of repetition number of user for goal object materials.

3. STR : The degree of study time of user for related objects with goal object .

4. LPR : The exam performance of user for related objects with goal object

5. PEG : The exam performance of user for goal objects.

2.4 .1 Scattered Plot of User Knowledge Modeling Raw Data Set

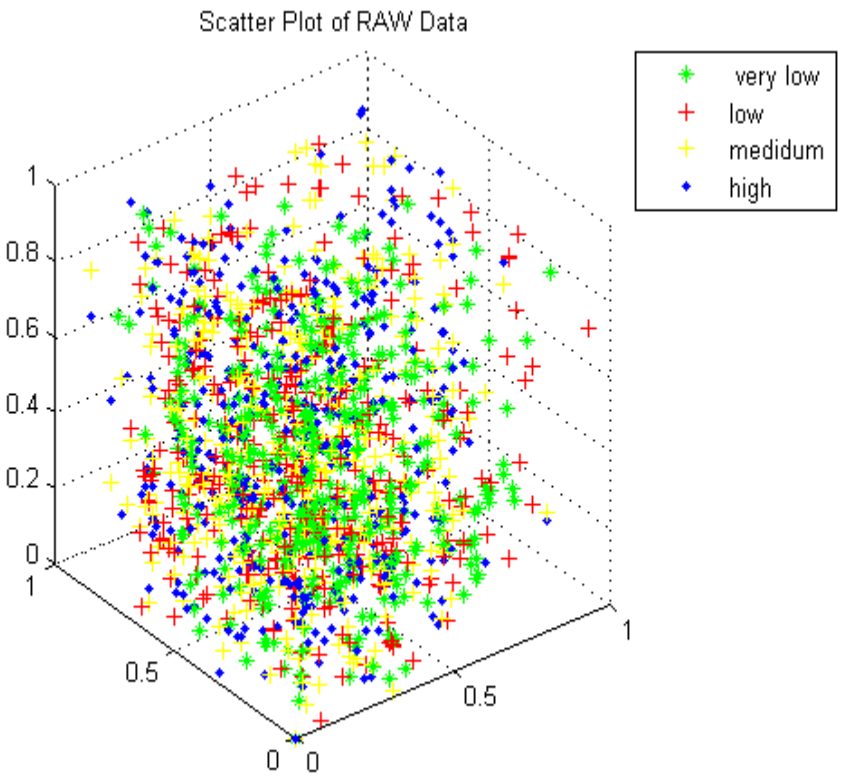

Fig 2.3 Scattered Plot of Raw User Knowledge Modeling Data

CHAPTER 3

THE BACK PROPAGATION ALGORITHM

3.1 Multilayer Perceptron (MLP)

To set the stages for a description of the multilayer perceptron in its general form, the network is connected to all the nodes and neurons in the previous layer. Signal flow through the network progresses in a forward direction, from left to right and on a layer-by-layer basis.[28]

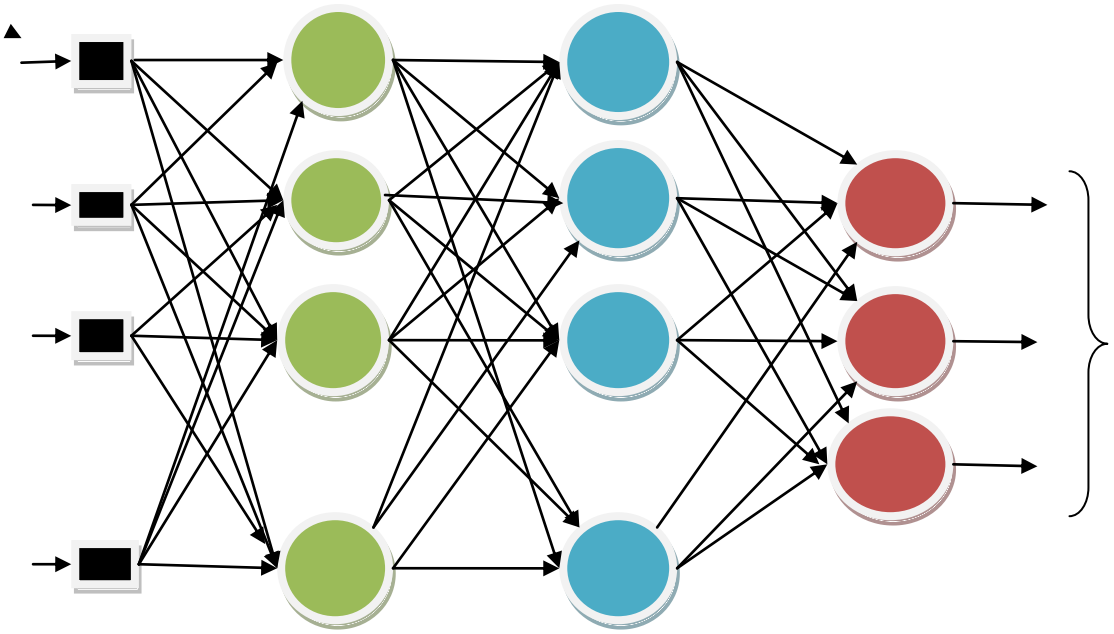

Fig. 3.1Architectural Graph of a Multilayer Perceptron with Two Hidden Layers

Two kinds of signals are identified in this MLP network is given by

3.1.1 Function Signal

A function signal is an input signal that comes in at the input end of the network, propagates forward (neuron by neuron) through the network, and emerges at the output end of the network as an output signal. At each neuron of the network through which a function signal passes, the signal is calculated as a function of the inputs and associated weights applied to that neuron. The function signal is called as the input signal.[28]

3.1.2. Error Signal

An error signal originates at an output neuron of the network, and propagates backward (layer by layer) through the network. It as an error signal because its computation by every neuron of the network involves an error-dependent function in one form or another.

The output neurons (computational nodes) constitute the output layers of the network. The remaining neurons constitute hidden layers of the network. Thus the hidden unit are not of the output or input of the network, hence their designation as hidden. The first hidden layer is fed from input layer made up of sensory units (source node); the resulting output of the first hidden layer are in turn applied to the next hidden layer and so on for the rest of the network. [11]

Each hidden or output neuron of a multilayer perceptron is designed to perform two computations:

1. The computation of the function signal appearing at the output of a neuron, which is expressed as a continuous nonlinear function of the input signal and synaptic weights associated with that neuron.

2. The computation of an estimates of the gradient vector which is needed for the backward pass through the network.

3.2 Analytical Description of Back Propagation Algorithm (BP)

The error signal at output of neuron j at iteration n is defined by

$$e_j(n) = d_j(n) - y_j(n) \tag{3.1}$$

the instantaneous value of the error energy for neuron *j*, is defined as

$$= \frac{1}{2} e_j^2(n) \tag{3.2}$$

Correspondingly the instantaneous value of the total error energy is obtained by summing $\frac{1}{2} e_j^2(n)$ overall all neurons in the output layer; these are the only visible neuron for which error signals can be calculated directly. Thus it is given by

$$\mathrm{E}(n) = \frac{1}{2} \sum_{j \epsilon C} e_j^2(n) \tag{3.3}$$

where the set C includes all the neurons in the output layer of the network. Let N be the total number of patterns contained in the training set. Thus the average square energy with respect to the set size N is given by

$$E_{av} = \frac{1}{N} \sum_{n=1}^{N} E(n) \tag{3.4}$$

The instantaneous error energy $\mathrm{E}(n)$, and therefore the average error energy E_{av}, is a function of all the free parameters of the network. The objective of the learning process is to adjust the free parameters of the network to minimize E_{av}.

Here it is consider a simple method of training in which the weight are updated on a pattern-by-pattern basis until one epoch i.e., one complete presentation of the entire training set has been dealt with and the weights adjustment are made in accordance with the respective error computed for each pattern presented to the network.[27]

The induced local field $v_j(n)$ produced at the input to the activation function associated with neuron *j* is given by

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n) \tag{3.5}$$

where *m* is the total no. of inputs applied to neuron *j*. The synaptic weight w_{j0} ,

(corresponding to the fixed input $y_0 = +1$) equals the bias b_j applied to neuron *j*. Hence the functional signal $y_j(n)$ appearing at the output of neuron *j* at iteration *n* is

$$y_j(n) = \varphi_j(v_j(n)) \tag{3.6}$$

In a similar manner to the LMS algorithm, the BP algorithm applies a correction $\Delta w_{ji}(n)$ to the synaptic weight $w_{ji}(n)$, which is proportional to the partial derivates $\frac{\partial E(n)}{\partial w_{ji}(n)}$,

According to chain rule, the gradient given as:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \tag{3.7}$$

The partial derivative $\frac{\partial E(n)}{\partial w_{ji}(n)}$ represents a sensitivity factor, determine the direction of search in weight space for the synaptic weight w_{ji} .

Differentiating both sides of Eq.(3.7) with respect to $e_j(n)$, gives

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n) \tag{3.8}$$

Now, differentiating Eq.(3.7) with respect to $y_j(n)$, gives

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \tag{3.9}$$

Next, Differentiating Eq.(3.7) with respect to $v_j(n)$, gives

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \varphi_j'(v_j(n)) \tag{3.10}$$

where use of prime signifies differentiating with respect to the argument. Finally, differentiating Eq.(3.7) with respect to $w_{ji}(n)$, gives

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_j(n) \tag{3.11}$$

From Eq. (3.7) to (3.11) gives

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -e_j(n)\,\varphi_j'(v_j(n))y_j(n) \tag{3.12}$$

The correction $\Delta w_{ji}(n)$applied to $w_{ji}(n)$ is defined by the delta rule

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)} \tag{3.13}$$

where η is the learning-rate parameter of the back-propagation algorithm. The use of minus sign in Eq.,(3.13) accounts for gradient descent in weight space (i.e., seeking a direction for weight change that reduce the value of $E(n)$. Accordingly the use of Eq.(3.12) in (3.13) gives

$$\Delta w_{ji}(n) = \eta\delta_j(n)y_i(n) \tag{3.14}$$

where the local gradient $\delta_j(n)$ is defined by

$$\delta_j(n) = -\frac{\partial E(n)}{\partial v_j(n)} \tag{3.15}$$

$$= -\frac{\partial E(n)}{\partial e_j(n)}\frac{\partial e_j(n)}{\partial y_j(n)}\frac{\partial y_j(n)}{\partial v_j(n)} \tag{3.16}$$

$$= e_j(n)\,\varphi_j'(v_j(n)) \tag{3.17}$$

The local gradient points to required change in synaptic weights. According to Eq.(3.17),the local gradient $\delta_j(n)$ for output neuron j is equal to the product of the corresponding error signal $e_j(n)$ for that neuron and the derivatives $\varphi_j'(v_j(n))$ of the associated activation function.

In this context, two distinct cases depending on where in the network neuron j is located.

Case 1 Neuron j is an Output Node

When neuron j is located in the output layer of the network, it is supplied with a desired response of its own.

Case 2 Neuron j is an Hidden Node

When neuron j is located in a hidden layer of the network, there is no specified desired response for that neuron. The error signal for a hidden neuron would have to be determined recursively in terms of the error signals of all the neuron to which that hidden neuron is directly connected; this is where the development of the back-propagation algorithm gets complicated.

Now it summarized the relations that have been derived for the back-propagation algorithm. First, the correction $\Delta w_{ji}\,(n)$ applied to the synaptic weight connecting neuron i to neuron j is defined by the delta rule

$$\begin{pmatrix} weight \\ correction \\ \Delta w_{ji}\,(n) \end{pmatrix} = \begin{pmatrix} learning \\ rate\ parameter \\ \eta \end{pmatrix} . \begin{pmatrix} local \\ gradient \\ \delta_{ji}\,(n) \end{pmatrix} . \begin{pmatrix} input\ signal \\ of\ neuron\ j \\ y_i\,(n) \end{pmatrix} \qquad (3.18)$$

Second, the local gradient $\delta_{ji}\,(n)$ depends on whether neuron j is an output node or a hidden node:

1. If neuron j is an output node, $\delta_{ji}\,(n)$ equals the product of the derivatives $\varphi_j^{'}(v_j(n))$ and the error signal $e_j(n)$, both of which are associated with neuron j.

2. If neuron j is a hidden node, $\delta_{ji}\,(n)$ equals the product of the associated derivative $\varphi_j^{'}(v_j(n))$and the weighted sum of the δ's computed for the neurons in the next hidden or output layer that are connected to neuron j . [28]

3.2.1 The Passes of Computation in BP Algorithm

In the application of the back-propagation algorithm two distinct passes of computation are different. The first pass is referred to as the forward pass and the second pass is referred to as the backward pass.

In the *forward pass* the synaptic weight remain unchanged throughout the network and the function signal of the network are computed on a neuron-by-neuron basis.

On the other hand, in the *backward pass* starts at the output layer by passing the error signals leftward through the network, layer by layer, and recursively computing the

local gradient for each neuron. This recursive process permits the synaptic weights of the network to undergo changes in accordance with the delta rule. [28]

3.2.2 Activation Function

The computation of the δ for each neuron of the multilayer perceptron requires knowledge of the derivatives of the activation function $\varphi(.)$ associated with that neuron. A continuously differently nonlinear activation function commonly used in multilayer perceptrons is sigmoidal nonlinearity; two form are described here below,

3.2.2.1. Logistic Function: This form of sigmoidal nonlinearity in its general form is defined by

$$\varphi_j\left(v_j(n)\right) = \frac{1}{1+\exp(-av_j(n))} \qquad a > 0 \; and \; -\infty < v_j(n) < \infty \tag{3.19}$$

where $v_j(n)$ is the induced local field of neuron j. According to this nonlinearity, the amplitude of the output lies inside the range $0 \leq y_j(n) \leq 1$.

For a sigmoidal activation function the synaptic weight are changed the most for those neurons in the network where the function signal are in the midrange. According to Rumelhart et al., this feature of BP learning that contributed to its stability as a learning algorithm.[29]

3.2.2.2. **Hyperbolic Tangent Function** : The hyperbolic tangent function, which in its most general form is defined by

$$\varphi_j\left(v_j(n)\right) = a\tanh\left(bv_j(n)\right), \qquad (a,b) > 0 \tag{3.20}$$

Where a and b are constants. In reality, the hyperbolic tangent function is just the logistic function rescaled and biased.

3.2.3 Learning Rate (LR) and Momentum Constant (MC)

The BP algorithm provides an approximation to the trajectory in weight space computed by the method of steepest descent. If the learning rate (LR), is smaller than the changes to the synaptic weight in the network will be smaller from one iteration to the next and smoother will be the trajectory in weight space. If it increasing LR, a

slower rate of learning and if LR too large in order to speed up the rate of learning resulting, a form that the network may become unstable. [29]

A simple method of increasing the rate of learning yet avoiding the danger of instability is to modify the delta rule by including a momentum term given by

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n) \qquad (3.21)$$

Where α is usually a positive number called *momentum constant*. It controls the feedback loop acting around $\Delta w_{ji}(n)$, as shown in figure where z^{-1} is the unit-delay operator. Equation (3.21) is called the generalized delta rule. It includes the delta rule as a special case (i.e., α=0). The incorporation of momentum in the BP algorithm represents a minor modification to the weight update and it preventing the learning process from terminating in a shallow local minimum on the error surface.

3.2.4 Modes of Training in BP Algorithm

One complete presentation of the entire training set during the learning process is called an *epoch.* The learning process is maintained on an epoch-by-epoch basis until the synaptic weighs and bias levels of the network stabilize and the average squared error over the training set converges to some minimum value. For a given training set, BP learning may thus proceed in one of the two basis ways: [28]

3.2.4.1 Sequential Mode

It is also called as on-line, pattern, or stochastic mode of learning. In this mode of operation weight updating is performed after the presentation of each training example.

3.2.4.2 Batch Mode

In the batch mode of BP learning, weight updating is performed after the presentation of all the training example that constitute an epoch. The use of batch mode of training provides an accurate estimates of the gradient vector and convergence to a local minima. The composition of the batch mode makes it easier to parallelize than the sequential mode.

When the training data are redundant, we find that unlike the batch mode, the sequential mode is able to take advantages of this redundancy because the examples are presented one at a time. This is particularly so when the data set is large and highly redundant. [29]

3.2.5 Convergence Criteria of Back-Propagation Algorithm

In general, the BP algorithm cannot be shown to converge and there are no well-defined criteria for stopping its operation. Rather, there are some reasonable criteria, each with its own practical merit, which may be used to terminate the weight adjustments. Let the weight vector $\boldsymbol{w}^*$ denote a minimum be it local or global minimum of the error surface. A necessary condition for $\boldsymbol{w}^*$ to be a minimum is that the gradient vector **g(w)** of the error surface with respect to the weight vector w be zero at $\boldsymbol{w} = \boldsymbol{w}^*$.

Accordingly, a convergence criteria for BP learning is given by

The back-propagation algorithm is considered to have converged when the Euclidean norm of the gradient vector reaches a sufficiently small gradient threshold.

The drawback of this convergence criterion is that, for successful trials, learning times may be long. Also, it requires the computation of the gradient vector **g(w).**

Another unique property of a minimum that we can use is the fact that the cost function or error measure E_{av} is stationary at the point $\boldsymbol{w} = \boldsymbol{w}^*$. Therefore a different criterion of convergence is given by

The back-propagation algorithm is considered to have converged when the absolute rate of change in the average squared error per epoch is sufficiently small.

The drawback of this criterion may result in a premature termination of the learning process.[28]

3.3 Error Surface

By studying error surfaces function, we can gain knowledge about the back propagation which is based on gradient descent in a criterion function. An error surface depends upon the classification task and there are some general properties of the error surfaces that seems to hold over a broad range of real world pattern recognition problem. One of the main issues that concerned is local minima, if many local minima plague the error landscape, then it unlikely that the network will find the global minimum. The possibility of the presence local minima is one reason that it resort to iterative gradient descent analytic method are highly unlikely to find a single global minima especially in high-dimensional weight spaces.

In computational practice, the network can be caught in a local minima having high training error. It indicated that key features of the problem have not been learned by the network. In such cases it is traditional to reinitialize the weights and train again, possibly also changing other parameters in the net. In many problem convergence to a nonglobal minimum is acceptable, if the error is fairly low. Furthermore, common stopping criteria demand that training terminate even before the minimum is reached. Thus it is not essential that the network be converging toward the global minimum for acceptable performance. [30]

CHAPTER 4

CLUSTER ANALYSIS USING PCA AND TCA PREPROCESSING TECHNIQUES

PCA is a useful statistical technique that has found application in fields such as face recognition and image compression and is a common technique for finding patterns in data of high dimension. It is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in data can be hard to find in data of high dimension where the luxury of graphical representation is not available, PCA is a powerful tool for analyzing data. The other main advantage of PCA is that once you have found these patterns in the data and you compress the data i.e. by reducing the number of dimensions, without much loss of information. [5]

4.1 Component Analysis

Component analysis is an unsupervised approach to finding the right features from the data. Principal Component Analysis (PCA) projects d- dimensional data onto a lower-dimensional subspace in a way that is optimal in a sum-squared error sense.

4.2 Analytical Description of Principal Component Analysis (PCA)

Suppose that x is a vector of *p* random variables and that the variance of the *p* random variables and the structure of the covariance or correlation between the *p* variables are of interest. [33]

Although PCA does not ignore covariance and correlation, it concentrates on variables. The first step is to look for a linear function $\alpha_1^{'} x$ of the elements of x having maximum variance where α_1 is a vector of p constants $\alpha_{11}, \alpha_{12}, \alpha_{13}, \ldots\ldots, \alpha_{1p}$, so that

$$\alpha_1^{'} x = \alpha_{11} x_1 + \alpha_{12} x_2 + \alpha_{13} x_3 +, \ldots\ldots, +\alpha_{1p} x_p = \sum_{j=1}^{p} \alpha_{1j} x_j \qquad (4.1)$$

Next, look for a linear function $\alpha_2' x$, uncorrelated with $\alpha_1' x$ having maximum variance and so on, so that the *k*th stage a linear function $\alpha_k' x$ is found that has maximum variance subject to being uncorrelated with

$$\alpha_1' x, \alpha_2' x, \alpha_3' x, \alpha_4' x , \ldots\ldots, \alpha_{k-1}' x \tag{4.2}$$

The *k*th derived variable, $\alpha_k' x$ is the *k*th PC.

To derive the form of the PCs, Consider first the vector $\alpha_1' x$; the vector α_1 maximize

$$var\left[\alpha_1' x\right] = \alpha_1' \sum \alpha_1 \tag{4.3}$$

It is clear that, the maximum will not be achieved for finite α_1 so a normalization constraint must be imposed. The constraint used in the derivation is $\alpha_1' \alpha_1 = 1$, that is, the sum of square of elements of α_1 equal 1. However, the use of constraints other than $\alpha_1' \alpha_1 = constant$ in the derivation leads to a more difficult optimization problem and it will produce a set of derived variables different from the PCs.

To maximize $\alpha_1' \sum \alpha_1$ subject to $\alpha_1' \alpha_1 = 1$, the standard approach is to use the technique of Lagrange multipliers. Maximize

$$\alpha_1' \sum \alpha_1 - \lambda(\alpha_1' \alpha_1 - 1) \tag{4.4}$$

Where λ is a Lagrange multiplier. Differentiation with respect α_1 to gives

$$\sum \alpha_1 - \lambda \alpha_1 = 0 \tag{4.5}$$

or

$$(\sum - \lambda \mathrm{I}_P)\, \alpha_1 = 0 \tag{4.6}$$

where I_P is the ($p \times p$) identity matrix. Thus, λ is an eigenvalue of Σ and α_1 is the corresponding eigenvector. To decide which of the p eigenvectors gives $\alpha_1' x$ with maximum variance, the quantity to be maximized is

$$\alpha_1' \Sigma \alpha_1 = \alpha_1' \lambda \alpha_1 = \lambda \alpha_1' \alpha_1 = \lambda \tag{4.7}$$

so λ must be as large as possible. Thus α_1 is the eigenvector corresponding to the largest eigenvalue of Σ and $var\left[\alpha_1' x\right] = \alpha_1' \sum \alpha_1 = \lambda_1$, the largest eigenvalue.

In general, the *k*th PC of x is $\alpha_k' x$ and $var(\alpha_1' x) = \lambda_k$

Where λ_k is the kth largest eigenvalue of Σ and α_k is the corresponding eigenvector.

The second PC, $\alpha_2' x$ maximize $\alpha_2' \Sigma \alpha_2$ subject to being uncorrelated with $\alpha_1' x$ or equivalently subject to $cov(\alpha_1' x, \alpha_2' x)$,where *cov (x,y)* denotes the covariance between the random variables x and y. But

$$cov(\alpha_1' x, \alpha_2' x) = \alpha_1' \Sigma \alpha_2 = \alpha_2' \Sigma \alpha_1 = \alpha_2' \lambda \alpha_2 = \lambda_1 \alpha_2' \alpha_1 = \lambda_1 \alpha_1' \alpha_2 \quad (4.8)$$

Thus any of the equations

$$\alpha_1' \Sigma \alpha_2 = 0 \ , \ \alpha_2' \Sigma \alpha_1 = 0 \ ,$$

$$\alpha_1' \alpha_2 = \ , \ \alpha_2' \alpha_1 = 0 \quad (4.9)$$

Could be used to specify zero correlation between $\alpha_1' x$ and $\alpha_2' x$. Choosing the last of these (an arbitrary choice), and nothing that a normalization constraint is again necessary the quantity to be maximized is

$$\alpha_2' \Sigma \alpha_2 - \lambda\left(\alpha_2' \alpha_2 - 1\right) - \phi\, \alpha_2' \alpha_1 \quad (4.10)$$

where, λ and ϕ are Lagrange multipliers. Differentiation with respect to α_2 gives

$$\Sigma \alpha_2 - \lambda(\alpha_2) - \phi \alpha_1 = 0 \quad (4.11)$$

And multiplication of this equation on the left by α_1' gives

$$\alpha_1' \Sigma \alpha_2 - \alpha_1' \lambda(\alpha_2) - \alpha_1' \phi \alpha_1 = 0 \quad (4.12)$$

Since the first two terms are zero and $\alpha_1' \alpha_1 = 1$, reduces to ϕ= 0

Therefore

$$\Sigma \alpha_2 - \lambda(\alpha_2) = 0 \quad (4.13)$$

or

$$(\Sigma - \lambda(I_P))\alpha_2 = 0 \qquad (4.14)$$

So, λ is once more an eigenvalue of Σ and α_2 the corresponding eigenvector.

Again $\lambda = \alpha_2^{'} \Sigma \alpha_2$, so λ is to be large as possible. Assuming that Σ does not have repeated eigenvalue. If it did, it follows that $\alpha_2 = \alpha_1$ violating the constraint $\alpha_1^{'} \alpha_2 = 0$ Hence λ is the second largest eigenvalue of Σ and α_2 is the corresponding eigenvector.

It can be shown that for third, fourth ,….., *pth* PCs, the vectors of the coefficients $\alpha_3 \alpha_4, \ldots\ldots \alpha_p$ are the eigenvectors of Σ corresponding to $\lambda_3 \lambda_4, \ldots\ldots \lambda_p$ the third and fourth largest ,….., and the smallest eigenvalue , respectively. Furthermore

$$var(\alpha_k^{'} x) = \lambda_k \ , \qquad for\ k = 1,2, \ldots\ldots, p \qquad (4.15)$$

This derivation of the PC coefficients and variances as eigenvectors and eigenvalue of a covariance matrix is standard. [33]

4.2.1 PCA Preprocessing on Data Signal

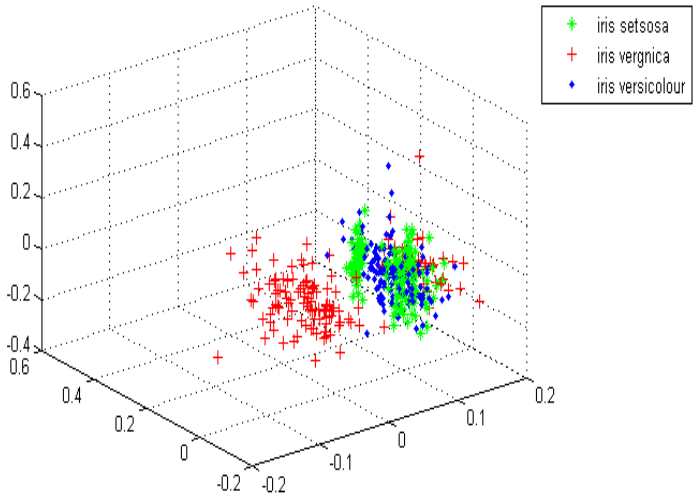

Fig.4.1 Scatter Plot of Iris Data with PCA

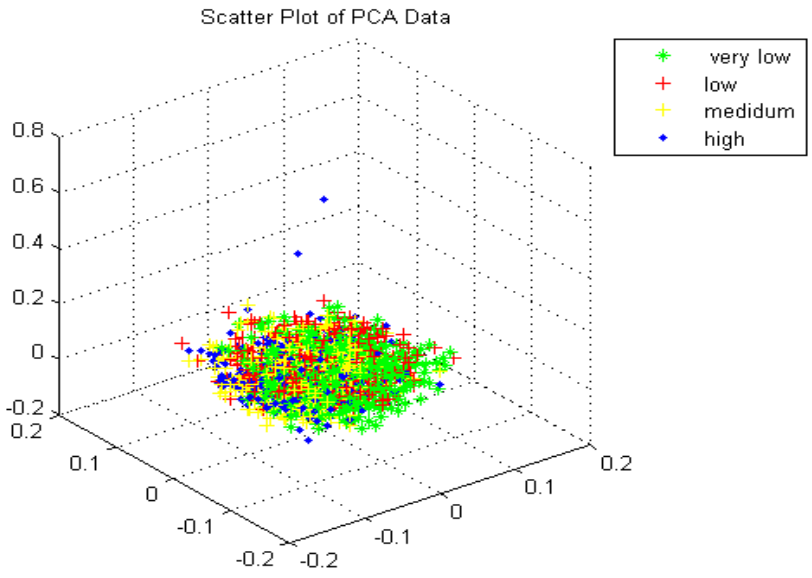

Fig.4.2 Scatter Plot of Student Modeling Data with PCA

4.3 Transform Cluster Analysis (TCA)

Domain adaptation allows knowledge from a source domain to be transferred to a different but related target domain. Transfer Cluster analysis tries to learn some transfer components across domains in a reproducing kernel space using maximum mean discrepancy. In the subspace spanned by these transfer components, data properties are preserved and data distributions in different domains are close to each other. So with the new representations in this subspace, it is easy to apply various standard machine learning algorithms to train classifiers or regression models in the source domain for use in the target domain. [5][8][6]

Transfer Component Analysis preprocessing is carried out in terms of Transfer component Analysis matrix T_{ijk} , given by the following equation

$$T_{ijk} = \frac{X_{ijk} - \overline{X_{\iota J}}}{V_{ij}} + \overline{X_{\iota J}} \qquad if\ V_{ij} \geq 1 \tag{4.16}$$

or $$T_{ijk} = (X_{ijk} - \overline{X_{\iota J}})V_{ij} + \overline{X_{\iota J}} \qquad if\ V_{ij} < 1 \tag{4.17}$$

where X_{ijk}, the original data matrix X ; $\overline{X_{\iota J}}$ mean of X ; V_{ij}, Variance matrix of X

4.3.1 TCA Preprocessing on Data Signal

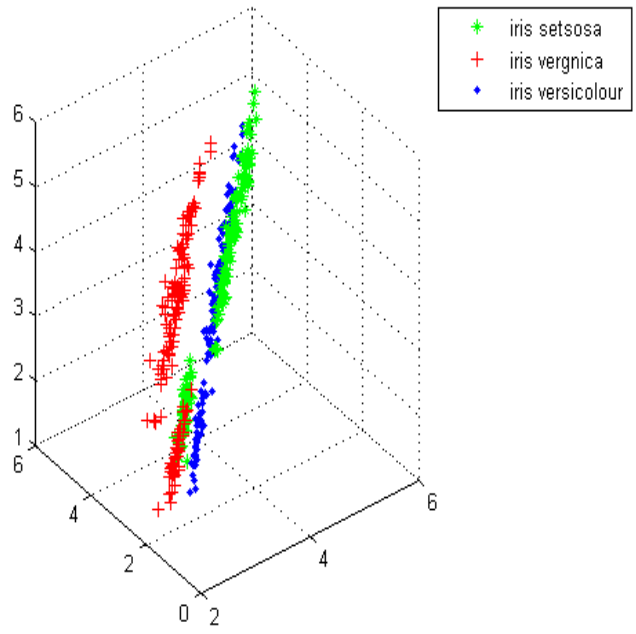

Fig. 4.3 Scatter Plot of Iris Data with TCA

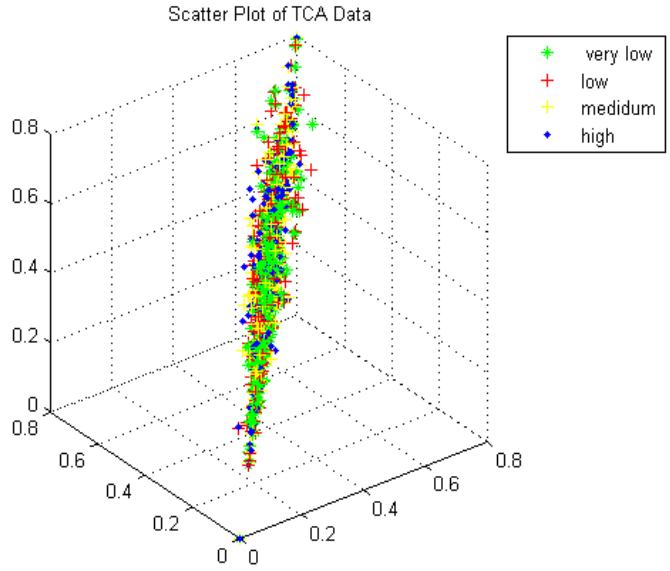

Fig.4.4 Scatter Plot of Student Modeling Data with TCA

CHAPTER 5

WAVELET TRANSFORM AS A PREPROCESSING TECHNIQUE

It is a necessary background to understand how WT works. It has been by far the most important signal processing tool for many years.

5.1 The Fourier Transform (FT)

J. Fourier, showed that any periodic function can be expressed as an infinite sum of periodic complex exponential functions. FT decomposes a signal to complex exponential functions of different frequencies. The way it does this, is defined by the following two equations,

$$X(f) = \int_{-\infty}^{\infty} x(t) * e^{-2j\pi ft} dt \tag{5.1}$$

$$x(t) = \int_{-\infty}^{\infty} X(f) * e^{2j\pi ft}\ df \tag{5.2}$$

In the above equation, t stands for time, f stands for frequency, and x denotes the signal. Note that x denotes the signal in time domain and the X denotes the signal in frequency domain. Equation (5.1) is called the Fourier transform of x(t), and equation (5.2) is called the inverse Fourier transform of X(f), which is x(t). [34]

The Fourier transform tells whether a certain frequency component exists or not. This information is independent of where in time this component appears. It is therefore very important to know whether a signal is stationary or not, prior to processing it with the FT.

Fourier transform is not suitable if the signal has time varying frequency, i.e., the signal is non-stationary. If only, the signal has the frequency component *f* at all times then the result obtained by the Fourier transform makes sense. [34]

5.2 Short Time Fourier Transform (STFT)

There is only a minor difference between STFT and FT. In STFT, the signal is divided into small enough segments where these segments of the signal can be assumed to be stationary. For this purpose, a window function *w* is chosen. The width of this window must be equal to the segment of the signal where its stationarity is valid.[8][34]

This window function is first located to the very beginning of the signal i.e., the window function is located at t=0. It is assumed that the width of the window is T sec. At this time instant, the window function will overlap with the first T/2 seconds. The window function and the signal are then multiplied. By doing this, only the first T/2 seconds of the signal is being chosen, with the appropriate weighting of the window. Then this product is assumed to be just another signal, whose FT is to be taken. In other words, FT of this product is taken, just as taking the FT of any signal.

The result of this transformation is the FT of the first T/2 seconds of the signal. If this portion of the signal is stationary, as it is assumed, then there will be no problem and the obtained result will be a true frequency representation of the first T/2 seconds of the signal.

The next step, would be shifting this window (for some t1 seconds) to a new location, multiplying with the signal, and taking the FT of the product. This procedure is followed, until the end of the signal is reached by shifting the window with "t1" seconds intervals.

Mathematically the expression for STFT given by

$$STFT(t,f) = \int_t [x(t) * \omega^*(t-t')] * e^{-j2\pi ft}\, dt \qquad (5.3)$$

x(t) is the signal itself, w(t) is the window function, and * is the complex conjugate. As seen from the above equation (5.3), the STFT of the signal is nothing but the FT of the signal multiplied by a window function. For every t' and f a new STFT coefficient is computed.

5.3 Wavelet

Since STFT gives the Time-Frequency representation of the signal, why it is need for the wavelet transform. The problem with the STFT has something to do with the width of the window function that is used. This width of the window function is known as the support of the window. If the window function is narrow, than it is known as compactly supported. This terminology is more often used in the wavelet world. [3][7]

5.3.1 The Continuous Wavelet Transform (CWT)

The continuous wavelet transform was developed as an alternative approach to the Short Time Fourier transform (STFT) to overcome the resolution problem. The wavelet analysis is done in a similar way to the STFT analysis, in the sense that the signal is multiplied with a function, it is the wavelet, similar to the window function in the STFT and the transform is computed separately for different segments of the time-domain signal. However, there are two main differences between the STFT and the CWT: [34]

1. The Fourier transforms of the windowed signals are not taken, and therefore single peak will be seen corresponding to a sinusoid i.e. negative frequencies are not computed.

2. The width of the window is changed as the transform is computed for every single spectral component, which is probably the most significant characteristic of the wavelet transform.

The continuous wavelet transform is defined as follows

$$CWT_x^{\psi}(\tau, s) = \Psi_x^{\psi}(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t)\psi^*\left(\frac{t-\tau}{s}\right) dt \qquad (5.4)$$

As seen in the above equation, the transformed signal is a function of two variables, **τ** and s, the translation and scale parameters respectively. **ψ** (t) is the transforming function and it is called the mother wavelet.

The term wavelet means a small wave. The smallness refers to the condition that this window function is of finite length compactly supported. The wave refers to the

condition that this function is oscillatory. The term mother implies that the functions with different region of support that are used in the transformation process are derived from one main function, or the mother wavelet. In other words, the mother wavelet is a prototype for generating the other window functions.

The term translation is used in the same sense as it was used in the STFT. It is related to the location of the window as the window is shifted through the signal. This term corresponds to time information in the transform domain. [8]

5.3.2 The CWT Wavelet Synthesis

The continuous wavelet transform is a reversible transform, provided that Equation (5.6) is satisfied. Fortunately, this is a very non-restrictive requirement. The continuous wavelet transform is reversible if Equation (5.6) is satisfied, even though the basis functions are in general may not be orthonormal. The reconstruction is possible by using the following reconstruction formula:[3][7]

$$x(t) = \frac{1}{c_\psi^2} \int_s \int_\tau \Psi_x^\psi(\tau, s) \frac{1}{s^2} \psi\left(\frac{t-\tau}{s}\right) d\tau ds \tag{5.5}$$

where ψ is a constant that depends on the wavelet used. The success of the reconstruction depends on this constant called, the admissibility constant, to satisfy the following admissibility condition:

$$c_\psi = \left\{ 2\pi \int_{-\infty}^{\infty} \frac{\left|\hat{\psi}(\xi)\right|^2}{|\xi|} d\xi \right\}^{\frac{1}{2}} < \infty \tag{5.6}$$

where $\hat{\psi}(\xi)$ is the FT of ψ(t). Equation (5.6) implies that $\hat{\psi}(0) = 0$, which is

$$\int \psi(t) dt = 0 \tag{5.7}$$

As stated above, Equation (5.7) is not a very restrictive requirement since many wavelet functions can be found whose integral is zero. For Equation (5.7) to be satisfied, the wavelet must be oscillatory.

5.4 Wavelet Families

The choice of wavelet is dictated by the signal or image characteristics or nature of applications. If we know the properties of analysis and synthesis analysis, we can choose the wavelet that optimized our application. Wavelet families are varying in terms of several important properties.[34]

- Support of wavelet in time and frequency and rate of decay.
- Symmertry or antisymmertry of wavelet i.e., the accompanying perfect reconstruction filter have linear phase.
- Number of vanishing movement i.e., wavelet with increasing no. of vanishing movement result in sparse representation for a large class of signals and images.
- Regularity of wavelet i.e., smother wavelet provides sharper frequency resolution. Additionally, iterative algorithms for wavelet construction converge faster.
- Existance of scaling function, φ

Wavelet Families(short name)	Wavelet Family Name
'haar'	Haar Wavelet
'db'	Daubechices wavelet
'Sym'	Symlet wavelet
'coif'	Coiflet wavelet
'bior'	Biorthogonal wavelet
'rbio'	Reverse biorthogonal
'meyr'	Meyer wavelet

Table: 1 Wavelet Families

5.4.1 Daubechices Wavelets: dbN

The dbN wavelet are the Daubechices extremal phase wavelets. N refers to the number of vanishing movements. db1 wavelet is also known as ‘Haar’ wavelet. The Haar wavelet is the only wavelet which is orthogonal with linear phase.

5.7 Scatter Plots of Wavelet Transform

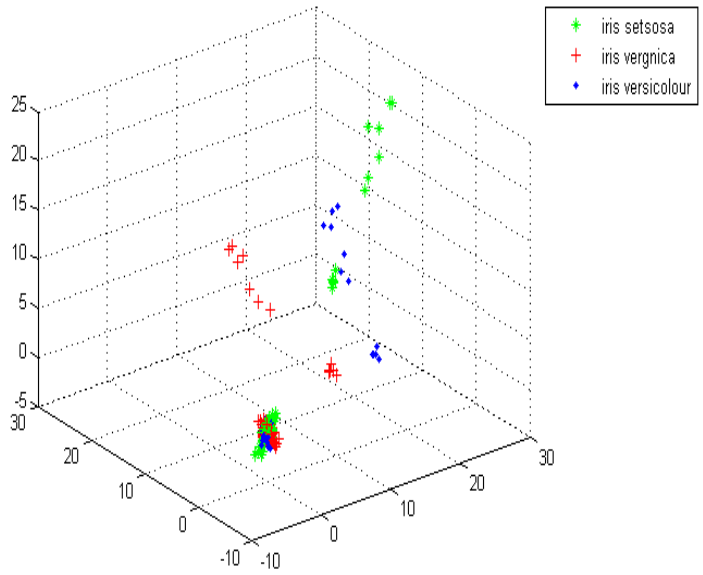

Fig. 5.1 Scatter Plot of Wavelet Transform of Iris Data

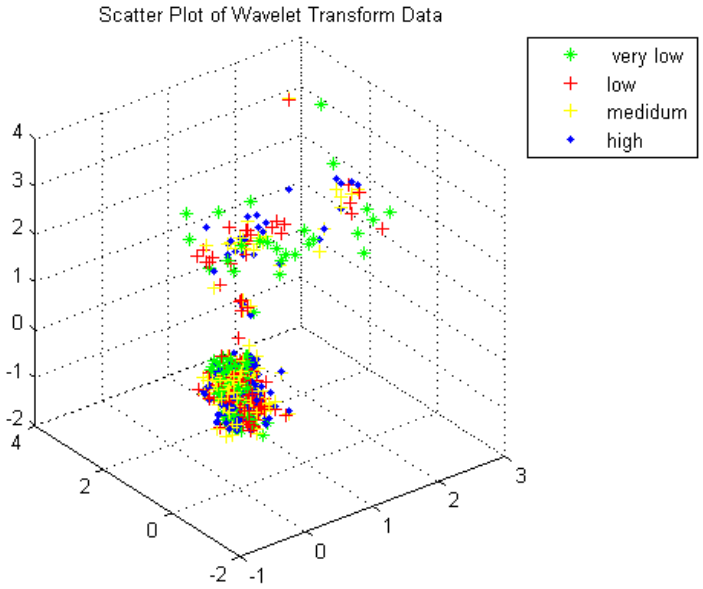

Fig. 5.2 Scatter Plot of Wavelet Transform of Student Modeling Data

CHAPTER 6

RESULT AND DISCUSSION

6.0 Box Plot Descriptions

The Box Plot is a graphical representation of data that shows a data set's lowest value, highest value, median value and the size of the first and third quartile. The box plot is a good alternatives or compliment to a histogram and is usually better for showing several simultaneous comparisons. Box Plot produces a box plot of the data in X. If X is a matrix, there is one box per column; if X is a vector, there is just one box. On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles. In our Box Plot structures, the horizontal axis of Box Plot shows that number of hidden layer neuron in the network while the vertical axes shows corresponding mean square error (MSE). The learning rate of the network varies from 0.1 to 1 per single column. [32]

6.1 Classification Performance with of Raw Iris Data

The scatter plot of Iris Raw Data has been shown in section 2.3.1. It is evident from the plot that the data is highly jumbled and its very different to draw linear decision boundary with three classes. This data was fed to the Back Propagation (BP) trained neural classifier to train to 1000 epoch with a no. of neuron in the hidden layer varies experimentally. 50 Percentage of data was kept set in the training and the rest was designated test data. The experimental training was done for different combination of momentum constant and learning rate coefficient and their extensive result have been depicted in the form of Box Plot as shown in figure.

6.1.1 Classification Performance of ANN Classifier with Iris Data

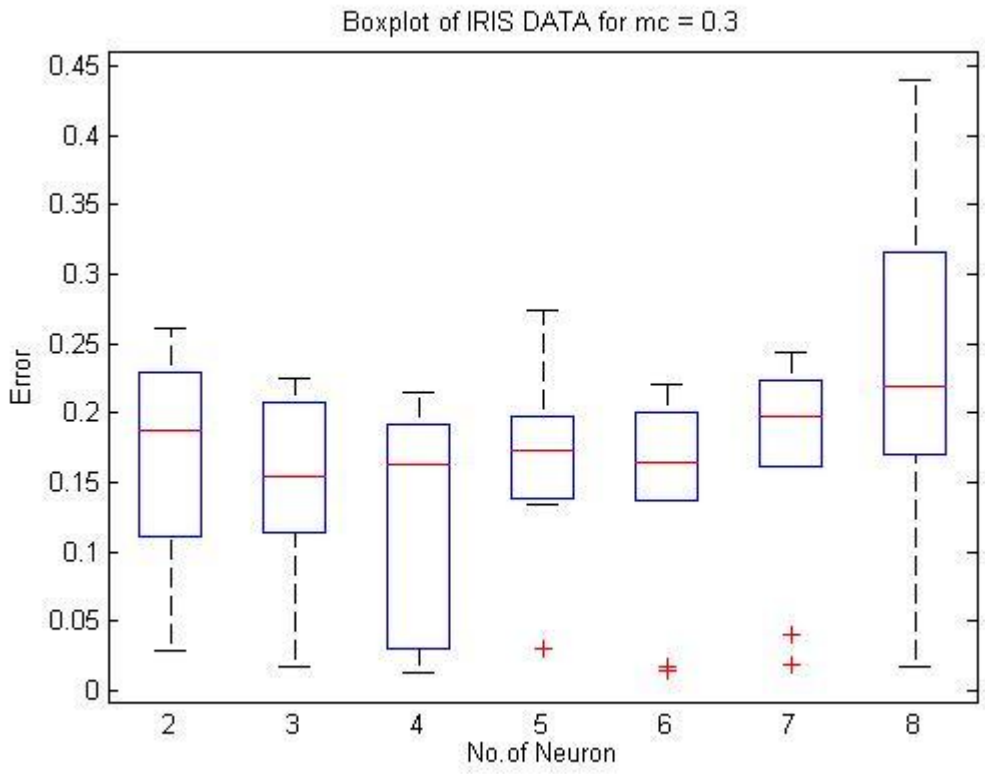

Fig. 6.1 Performance Box Plot of Iris Raw Data

As evident from above Box Plot that minimum mean error was obtained when the number of neurons **3** and corresponding momentum constant, **mc=0.3**. The learning rate of the network varies from its value from **lr=0.1 to 1,** which on this optimizing network architectures. The test data was applied and this percent classify was obtain.

6.1.2 Classification Performance with Principle Component Analysis (PCA) of Iris Data

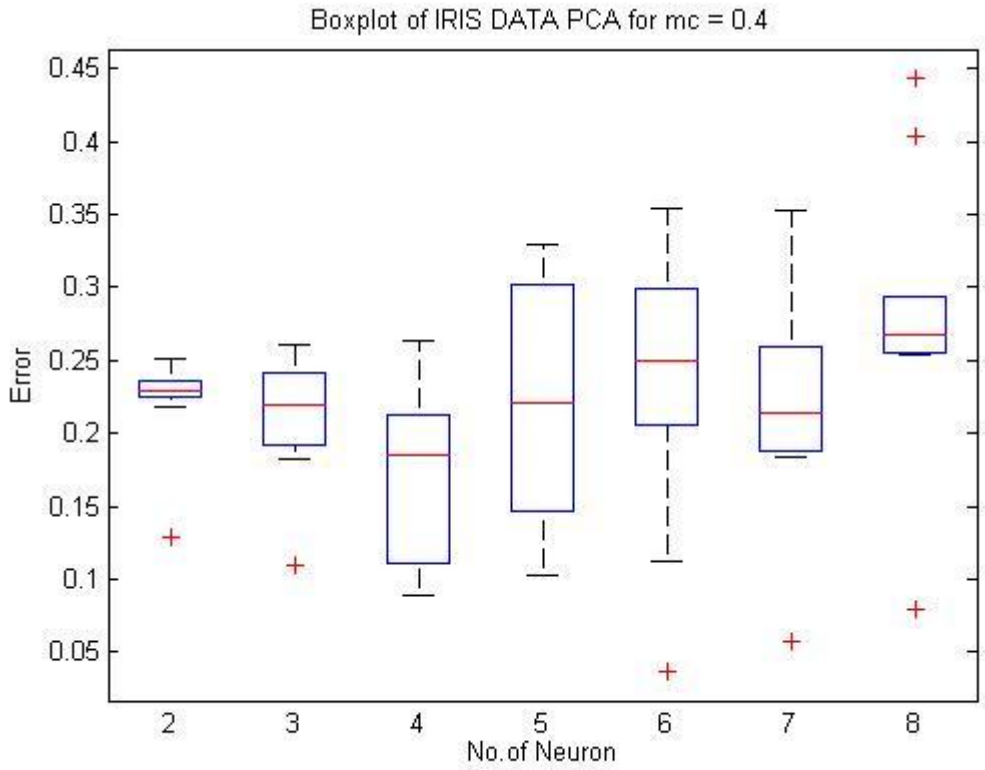

Fig.6.2 Performance Box Plot of Iris Data with PCA

As evident from above Box Plot that minimum mean error was obtained when the number of neurons **4** and corresponding momentum constant, **mc=0.4**. The learning rate of the network varies from its value from **lr=0.1 to 1,** which on this optimizing network architectures. The test data was applied and this percent classify was obtain.

6.1.3 Classification Performance with Transfer Component Analysis (TCA) of Iris Data

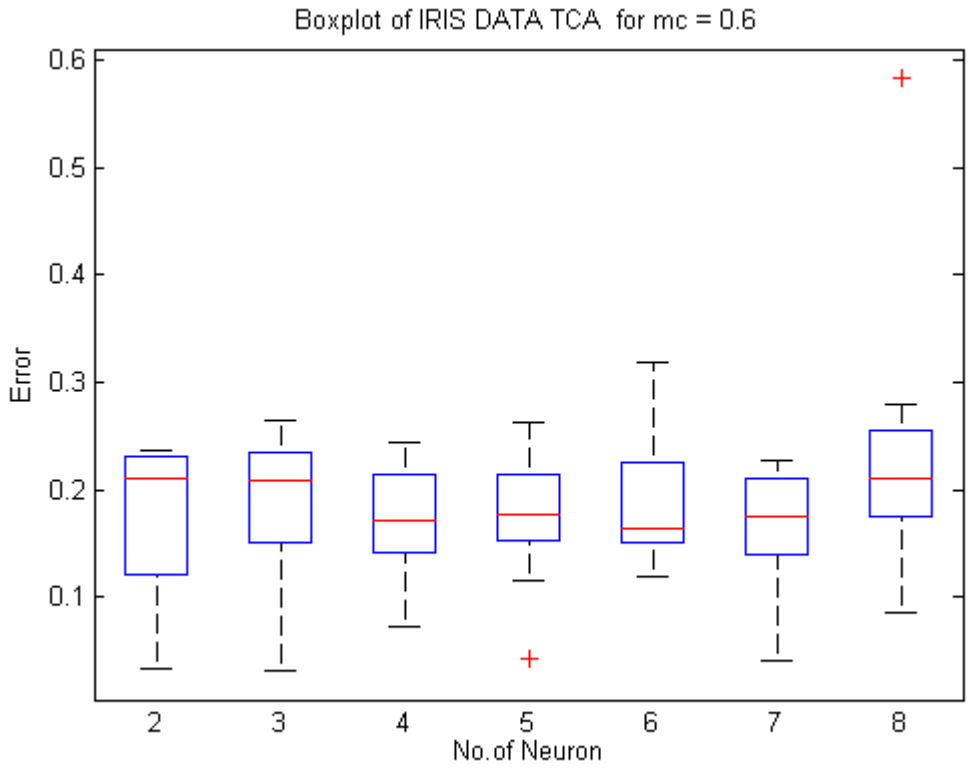

Fig. 6.3 Performance Box Plot of Iris Data with TCA

As evident from above Box Plot that minimum mean error was obtained when the number of neurons **6** and corresponding momentum constant, **mc=0.6**. The learning rate of the network varies from its value from **lr=0.1 to 1,** which on this optimizing network architectures. The test data was applied and this percent classify was obtain.

6.2 Classification Performance with Raw User Knowledge Student Modeling Data

The scatter plot of User knowledge Student Modeling Raw Data has been shown in section 2.4.1. It is evident from the plot that the data is highly jumbled and it is very different to draw linear decision boundary with four classes. This data was fed to the Back Propagation (BP) trained neural classifier to train to 1000 epoch with a no. of neuron in the hidden layer varies experimentally. 50 Percentage of data was kept set in the training and the rest was designated test data. The experimental training was done for different combination of momentum constant and learning rate coefficient and their extensive result have been depicted in the form of Box Plot as shown in figure.

6.2.1 Classification Performance of ANN Classifier with User Knowledge Student Modeling Data

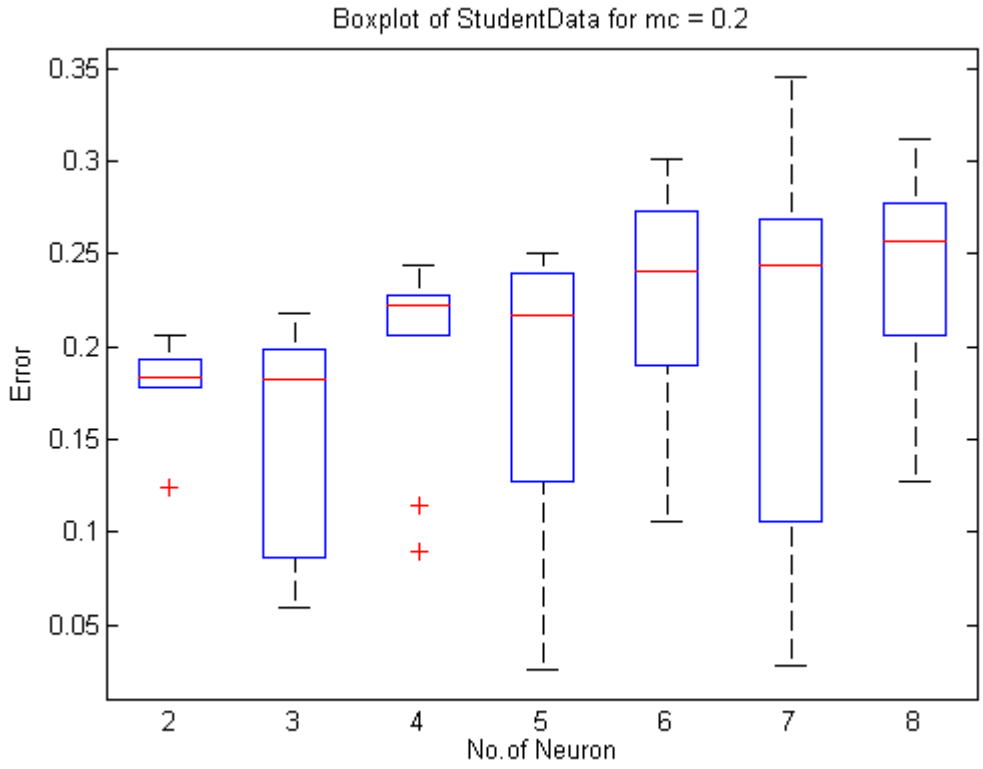

Fig. 6.4 Performance Box Plot of Student Modeling Raw Data

As evident from above Box Plot that minimum mean error was obtained when the number of neurons **3** and corresponding momentum constant, **mc=0.2**. The learning rate of the network varies from its value from **lr=0.1 to 1,** which on this optimizing network architectures. The test data was applied and this percent classify was obtain.

6.2.2 Classification Performance with Principle Component Analysis (PCA) of User knowledge Student Modeling Data

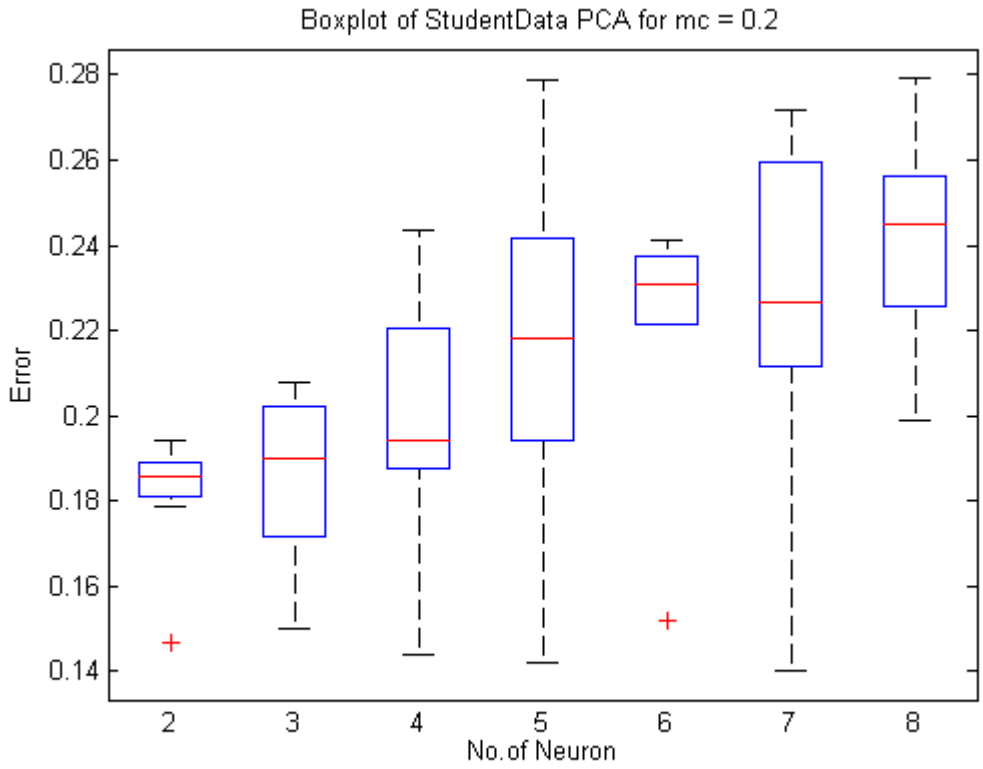

Fig. 6.5 Performance Box Plot of Student Modeling Data with PCA

As evident from above Box Plot that minimum mean error was obtained when the number of neurons **2** and corresponding momentum constant, **mc=0.2**. The learning rate of the network varies from its value from **lr=0.1 to 1,** which on this optimizing network architectures. The test data was applied and this percent classify was obtain.

6.2.3 Classification Performance with Transfer Component Analysis (TCA) of User knowledge Student Modeling Data

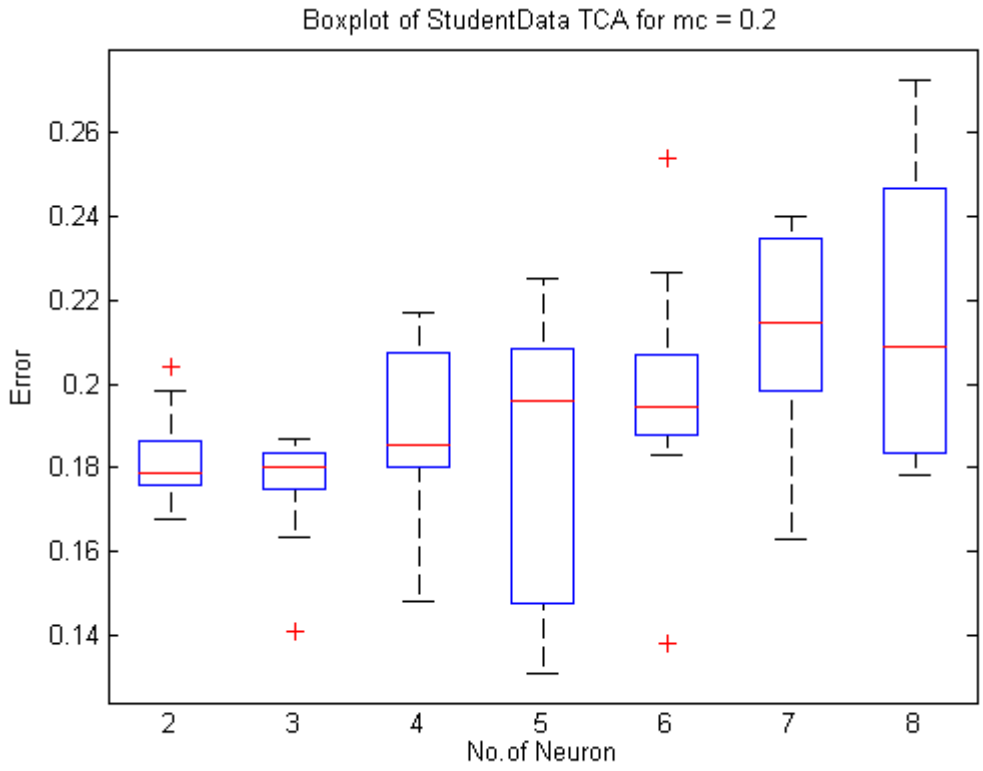

Fig. 6.6 Performance Box Plot of Student Modeling Data with TCA

As evident from above Box Plot that minimum mean error was obtained when the number of neurons **2** and corresponding momentum constant, **mc=0.2**. The learning rate of the network varies from its value from **lr=0.1 to 1,** which on this optimizing network architectures. The test data was applied and this percent classify was obtain.

6.3 Classification Performance with Wavelet Transform of Iris Data

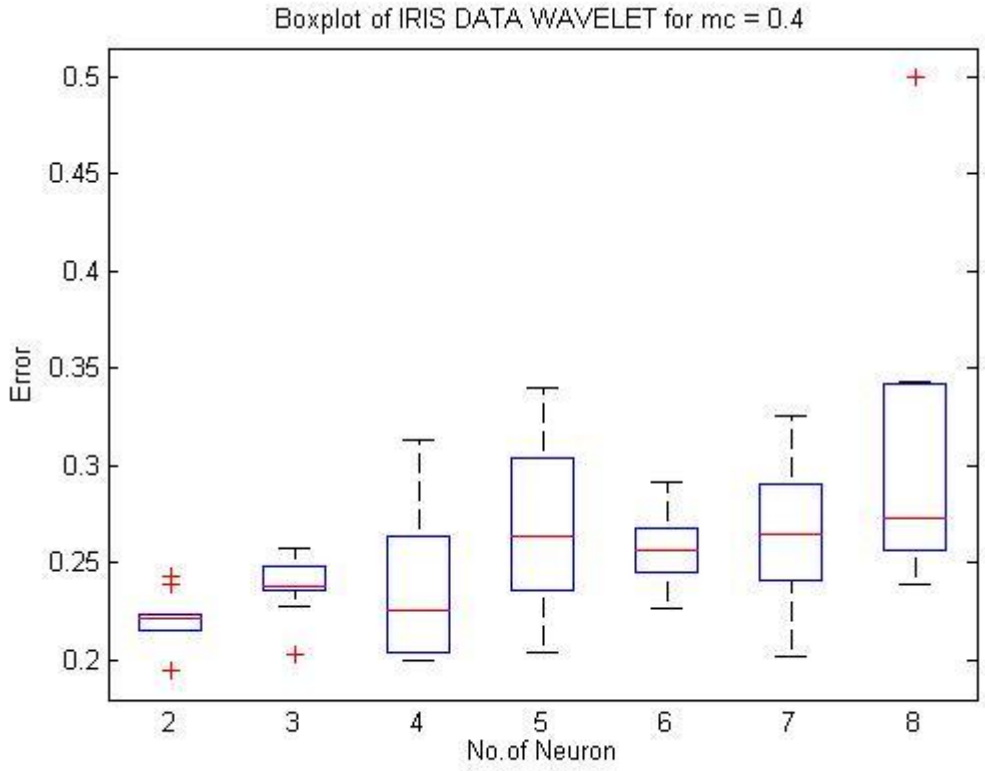

Fig. 6.7 Performance Box Plot of Wavelet Transform of Iris Data

As evident from above Box Plot that minimum mean error was obtained when the number of neurons **2** and corresponding momentum constant, **mc=0.4**. The learning rate of the network varies from its value from **lr=0.1 to 1,** which on this optimizing network architectures. The test data was applied and this percent classify was obtain.

6.3.1 Classification Performance with Wavelet Transform with PCA of IRIS Data

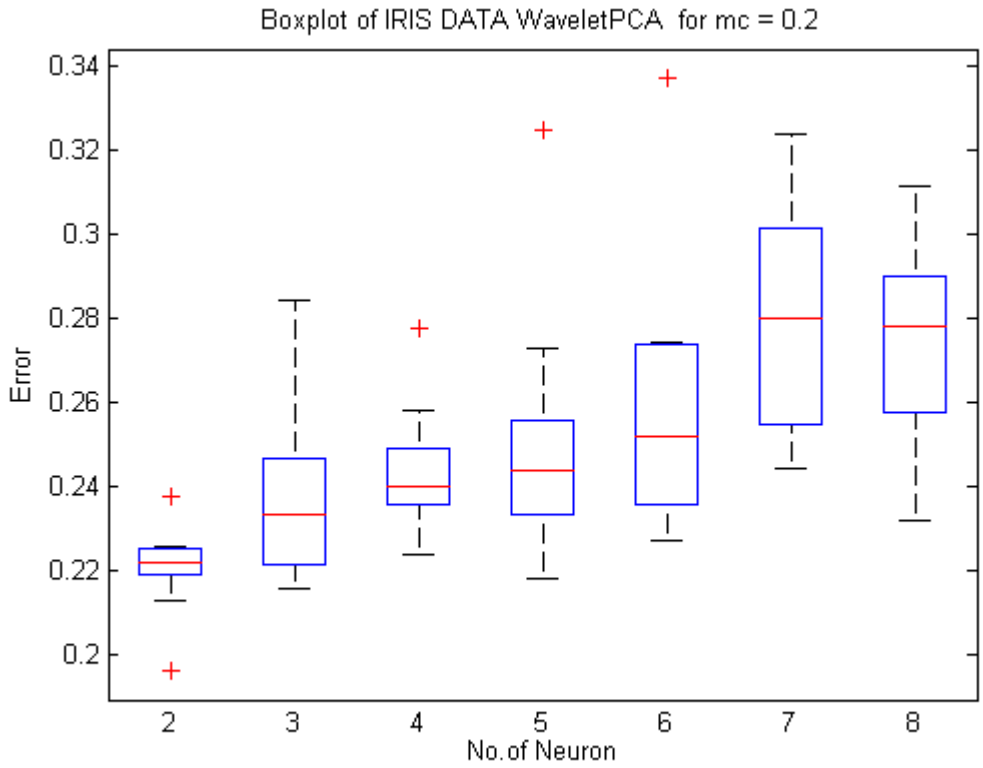

Fig. 6.8 Performance Box Plot of Wavelet Transform with PCA of Iris Data

As evident from above Box Plot that minimum mean error was obtained when the number of neurons **2** and corresponding momentum constant, **mc=0.2**. The learning rate of the network varies from its value from **lr=0.1 to 1,** which on this optimizing network architectures. The test data was applied and this percent classify was obtain.

6.3.2 Classification Performance with Wavelet Transform with TCA of IRIS Data

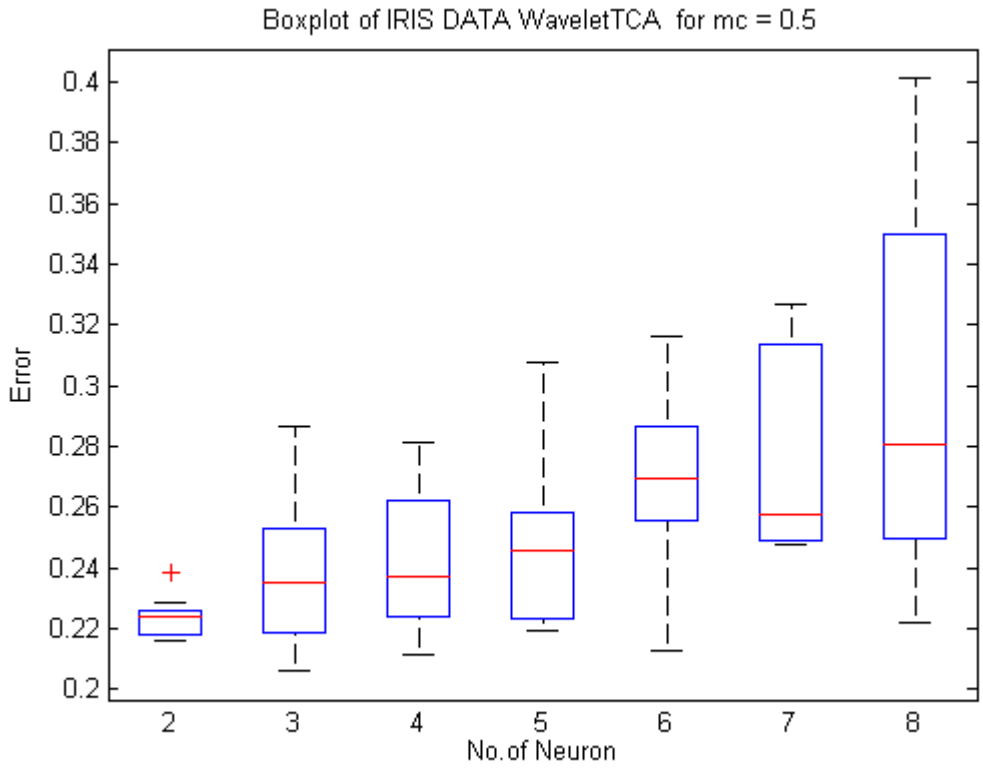

Fig. 6.9 Performance Box Plot of Wavelet Transform with TCA of Iris Data

As evident from above Box Plot that minimum mean error was obtained when the number of neurons **2** and corresponding momentum constant, **mc=0.5** . The learning rate of the network varies from its value from **lr=0.1 to 1,** which on this optimizing network architectures. The test data was applied and this percent classify was obtain.

6.4 Classification Performance with Wavelet Transform of User Knowledge Student Modeling Data

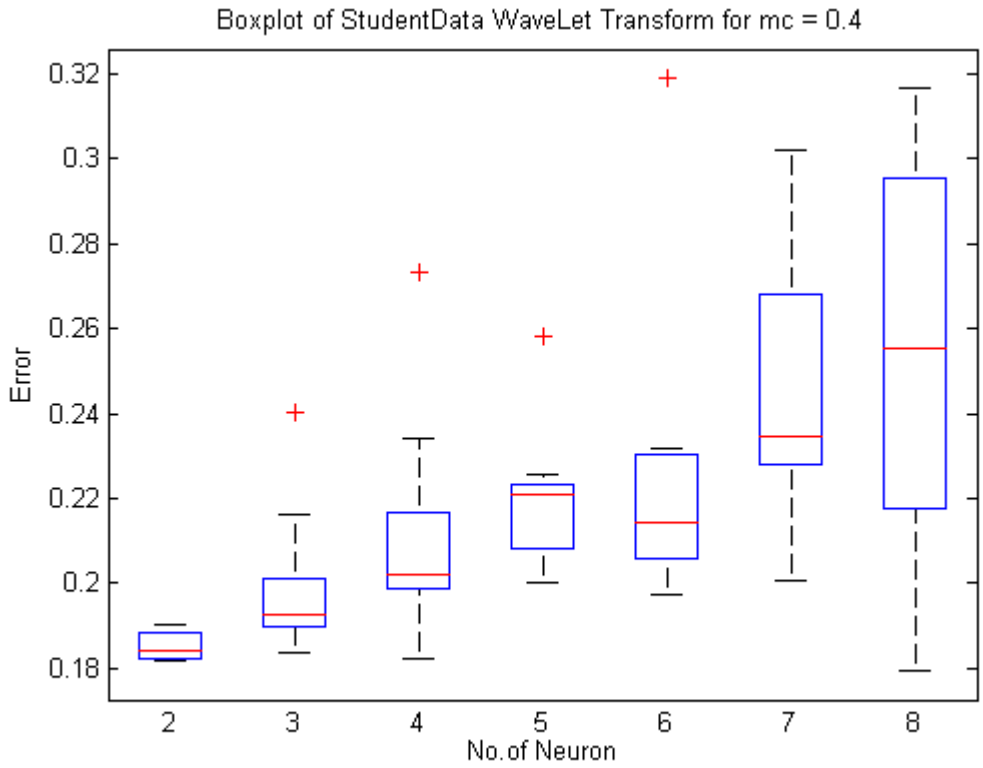

Fig.6.10 Performance Box Plot of Wavelet Transform of Student Modeling Data

As evident from above Box Plot that minimum mean error was obtained when the number of neurons **2** and corresponding momentum constant, **mc=0.2** . The learning rate of the network varies from its value from **lr=0.1 to 1,** which on this optimizing network architectures. The test data was applied and this percent classify was obtain.

6.4.1 Classification Performance with Wavelet Transform with PCA of User Knowledge Student Modeling Data

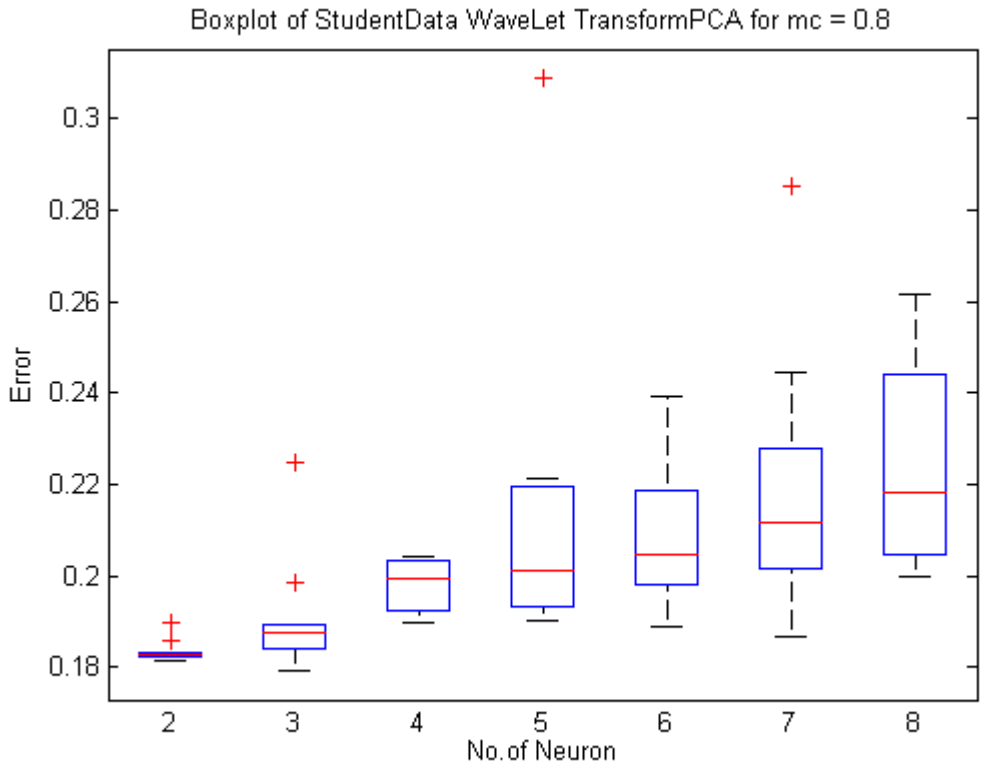

Fig. 6.11 Performance Box Plot of Wavelet Transform with PCA of Student Modeling Data

As evident from above Box Plot that minimum mean error was obtained when the number of neurons **2** and corresponding momentum constant, **mc=0.8** . The learning rate of the network varies from its value from **lr=0.1 to 1,** which on this optimizing network architectures. The test data was applied and this percent classify was obtain.

6.4.2 Classification Performance with Wavelet Transform with TCA of User Knowledge Student Modeling Data

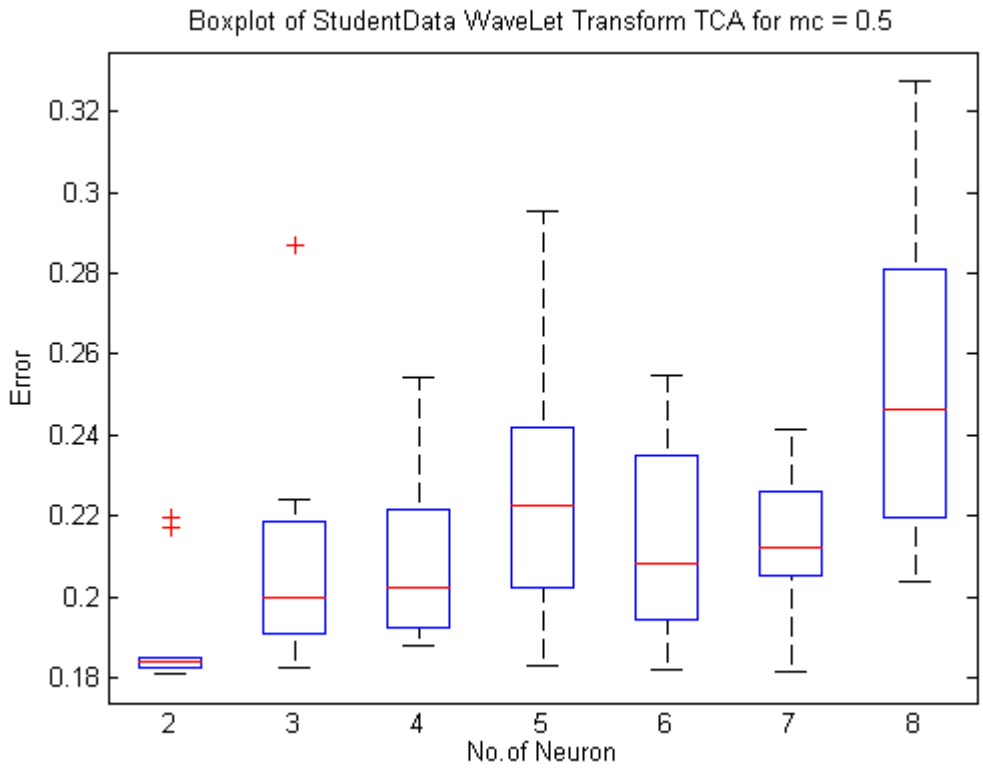

Fig. 6.12 Performance Box Plot of Wavelet Transform with TCA of Student Modeling Data

As evident from above Box Plot that minimum mean error was obtained when the number of neurons **2** and corresponding momentum constant, **mc=0.5** . The learning rate of the network varies from its value from **lr=0.1 to 1,** which on this optimizing network architectures. The test data was applied and this percent classify was obtain.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

Based upon the results presented in the previous chapter it can be concluded that data preprocessing has a profound effect on classification performance of the final classifier. In this work it has also come out as a new observation that if the data set is subjected to wavelet transform then the final architecture of the neural classifier becomes too simple and optimal performance can be obtained with only two neurons in the hidden layer. This holds a lot of promise from hardware implementation point of view since a simpler hardware can be designed with less silicon resources for realizing a wavelet coefficient trained neural classifier.

7.2 Future Scope

The Above mentioned techniques and algorithms will be applied to high dimensional unlabeled data in both time and frequency domain which could serve as starting point for the development of clustering technique for digital signal processing technique (DSP) applications. Furthermore, the effect of different distance measures can be evaluated on benchmark data.

REFRENCE

[1] H. Tolga Kahraman, Seref Sagiroglu, and Ilhami Colak, "The Development Of Intuitive Knowledge Classifier And The Modeling Of Domain Dependent Data", *Elsevier Knowledge Based System,* pp. 283-295*,* Vol 37, 2013.

[2] Ruoyi Jiang, Hongliang Fei, and Jun Huan, "A Family of Joint Sparse PCA Algorithms for Anomaly Localization in Network Data Streams", *IEEE Transactions On Knowledge And Data Engineering,* Vol. 25, No. 11, November 2013.

[3] Ravi Kumar, R. R. Das, V. N. Mishra, and R. Dwivedi, "Wavelet Coefficient Trained Neural Network Classifier for Improvement in Qualitative Classification Performance of Oxygen-Plasma Treated Thick Film Tin Oxide Sensor Array Exposed to Different Odors/Gases", *IEEE Sensors Journal,* Vol. 11, No. 4, April 2011.

[4] Chayaporn Kaensar, "Analysis On The Parameter Of Back Propagation Algorithm With Three Weight Adjustment Structure For Hand Written Digit Recognition", *IEEE Confrerence on Pattern Analysis and Machine Intelligence,* Vol. 28, No. 8, August 2013.

[5] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang, *Fellow, IEEE,"* Domain Adaptation via Transfer Component Analysis*", IEEE Transactions On Neural Networks,* Vol. 22, No. 2, February 2011.

[6] Sinno Jialin Pan and Qiang Yang", A Survey on Transfer Learning", *IEEE Transactions On Knowledge And Data Engineering,* Vol. 22, No. 10, October 2010.

[7] Pierre Baldi And Kurt Hornik, "Neural Networks and Principal Component Analysis, *Neural Networks,* Vol. 2, pp. 53-58, 1989.

[8] Stefan Pittner and Sagar V. Kamarthi, "Feature Extraction From Wavelet Coefficients for Pattern Recognition Tasks " *IEEE Transactions On Pattern Analysis And Machine Intelligence,* Vol. 21, No. 1, January 1999.

[9] C.-M. Lee, S.-S. Yang and C.-L. Ho, "Modified Back-Propagation Algorithm Applied To Decision-Feedback Equalization", *IEEE Proc.-Vis. Image Signal Process.,* Vol. 153, No. 6, December 2006.

[10] Marcelo Franceschi de Bianchi, Rodrigo Capobianco Guido, "A Wavelet-PCA Approach For Content-Based Image Retrieval", *Proceedings of the 38th Southeastern Symposium on System Theory* , Tennessee Technological University Cookeville, TN, USA, March 2006.

[11] V. V. Phansalkar and P. S. Sastry, "Analysis of the Back-Propagation Algorithm with Momentum", *IEEE Transactions On Neural Networks.* Vol 5. NO. 3. MAY l994.

[12] Richard Nock and Frank Nielsen, "On Weighting Clustering*", IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 8, August 2006.

[13] Liang Bai, Jiye Liang, Chuangyin Dang, and Fuyuan Cao, "The Impact of Cluster Representatives on the Convergence of the K-Modes Type Clustering". *IEEE Transactions on Pattern Analysis and Machine intelligence*, Vol. 35, No. 6, June 2013.

[14] Xinxin Bai Gang Chen ,Zhonglin Li, Wenjun Yin, JinDong "An Unsupervised Feature Weight Learning " , *IBM China Research Laboratory*, Beijing, China, 2011.

[15] Vladimir N. Vapnik, "An Overview of Statistical Learning Theory", *IEEE Transactions on Neural Networks,* Vol. 10, No. 5, September 1999.

[16] Rui Xu, and Donald Wunsch, "Survey of Clustering Algorithms*", IEEE Transactions On Neural Networks,* Vol. 16, No. 3, May 2005.

[17] Yixin Chen, James Z. Wang, and Robert Krovetz, "Cluster-Based Retrieval of Images by Unsupervised Learning ", *IEEE Transactions On Image Processing,* Vol. 14, No. 8, August 2005.

[18] Lei Gu, Xianling Lu, "Semi-supervised Locality-weight Fuzzy C-Means Clustering Based on Seeds and One Novel Decision Rule", *3rd International*

Conference on System Science, Engineering Design and Manufacturing Informatization, USA, August 2012.

[19] Chih-Cheng Hung, Sameer Kulkarni, and Bor-Chen Kuo, "A New Weighted Fuzzy C-Means Clustering Algorithm for Remotely Sensed Image Classification", *IEEE Journal of Selected topics in Signal Processing,* Vol. 5, No. 3, June 2011.

[20] Chang Li, Yafeng Wang, Fan Huang and Dacheng Yang, "A Novel Enhanced Weighted Clustering Algorithm for Mobile Networks", *Wireless Theories and Technologies Lab (WT&T)* 2012.

[21] A.Topchy, B.Minaei-Bidgoli, A.K. Jain, and W. F. Punch, "Adaptive Clustering Ensembles*," Proc. 17th Int'l Conf. Pattern Recognition,* pp. 272-275, 2004.

[22] B. Zhang, M. Hsu, and U. Dayal , "k-Harmonic Means—A Spatial Clustering Algorithm with Boosting," *Temporal , Spatial, and Spatio -Temporal Data Mining*, pp. 31-45, 2000.

[23] H. Attias, "A Variational Bayesian Framework for Graphical Models," *Advances in Neural Information Processing Systems 12*, pp. 209-215,1999.

[24] N. R. Pal And J. Biswas "Cluster Validation Using Graph Theoretic Concepts" *Pergamon Pattern Recognition,* Vol. 30, No. 6, pp. 847-857,1997.

[25] M.J. Kearns, Y. Mansour, and A.Y. Ng, "An Information-Theoretic Analysis of Hard and Soft Assignment Methods for Clustering,"*Proc. 13th Int'l Conf. Uncertainty in Artificial Intelligence*, pp. 282-293, 1997.

[26] J. C. Bezdek , "Pattern Recognition with Fuzzy Objective Function Algorithms''. *Plenum Press*, 1981.

[27] A.P. Dempster , N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc. B,* Vol. 39, pp. 1-38, 1977.

[28] Haykin, S., *Neural Network, A Comprensive Foundation 3rd ,Edition*, Prentice-Hall (2010).

[29] Kumar, S., *Neural Network , A Classroom Approach ,8th Edition* ,The McGraw-Hill Companies 2009.

[30] Richard O. Duda, Peter E. Hart, *"Pattern Classification",* 2th Edition, John Wiley & Sons 2012.

[31] C.M. Bishop, *Neural Networks for Pattern Recognition.* Oxford: Clarendon Press, 1995.

[32] McGill, R., J. W. Tukey, and W. A. Larsen. *"Variations of Boxplots",* The American Statistician. Vol. 32, No. 1, 1978, pp. 12–16.

[33] I.T. Jolliffe, *"Principal Component Analysis", 2nd Edition*, Springer, New York, 2002.

[34]Robi Polikar, *Fundamental Concepts and Overview of Wavelet Theory*, 2nd Edition ,Ames, Iova, 1996.

[35] Https://archive.ics.uci.edu