International Journal of Computing and Business Research (IJCBR)

ISSN (Online) : 2229-6166

Volume 4 Issue 2 May 2013

A PRECISE SCRUTINY ON BACK PROPAGATION NEURAL NETWORK AND ITS ALGORITHMIC APPROACH FOR WEB MINING

Tamanna Jain

M. M. University, Mullana

Haryana, India

ABSTRACT

World Wide Web (WWW) grows up very rapidly in recent years, and it contains an enormous amount of data and information that can be extracted via computer assisted tools, intelligent agents, search engines, and Web mining techniques. Consequently, how to explore useful information and knowledge from WWW is gradually becoming urgent need. However, to search or retrieve information and data from WWW manually is a difficult and time-consuming job because WWW has become a huge database and provided abundant information. Thus, how to effectively search, extract and filter data and information from Internet using intelligent agents and Web mining techniques have become important research issues.Past researches present that machine learning methods and the neural-based prediction or classification methods were extensively used in Web mining techniques. Among used machine learning methods, the gradient descent method is widely used to train various classifiers, such as Back-propagation neural network and linear text classifier. This study highlights the back propagation neural networks that will be used in predicting the query of the search engine.

Keywords: Back propagation Algorithm, Neural Network, Prediction System, Web mining

1. INTRODUCTION

Web mining is the use of data mining techniques to automatically discover and extract information from Web documents and services. Some previous studies have indicated that main challenges in Web mining are in terms of handling high-dimensional data, achieving incremental learning (or incremental mining), scalability, parallel and distributed mining algorithms. However, many traditional data mining methods cannot satisfy these needs for Web mining. In contrast with previous analysis, back propagation neural network can be used to solve the current issue.

Neural networks have emerged as advanced data mining tools in cases where othertechniques may not produce satisfactory predictive models. As the term implies,neural networks have a biologically inspired modelling capability, but are essentiallystatistical modelling tools. In this chapter, we study the basics of neural network modelling, some specific applications, and the process of implementing a neural networkproject.In the sub-field of data classification, neural-network methods have been found to be useful alternatives to statistical techniques such as those which involve regression analysis or probability density estimation (e.g., Holmström et al., 1997). The potential utility of neural networks in the classification of multisource satellite-imagery databases has been recognized for well over a decade, and today neural networks are an established tool in the field of remote sensing.

The most widely applied neural network algorithm in image classification and image processing remains the feedforwardbackpropagation algorithm. Along with these applications, back propagation algorithm can be used in web mining process as a prediction system.

2. NEURAL NETWORK BASICS

Neural networksrepresent a brain metaphor for information processing. These modelsare biologically inspired rather than an exact replica of how the brain actually functions.Neural networks have been shown to be very promising systems in many forecastingapplications and business classification applications due to their ability to "learn" fromthe data, their nonparametric nature (i.e., no rigid assumptions), and their ability to generalize. Neural computing refers to a pattern recognition methodology for machinelearning. The resulting model from neural computing is often called an artificial neuralnetwork (ANN) or a neural network. Neural networks have been used in many businessapplications for pattern recognition, forecasting, prediction, and classification. Neuralnetwork computing is a key component of any data miningtool kit.Applications of neural networks abound in finance, marketing, manufacturing, operations, information systems, and so on. The human brain possesses bewildering capabilities for information processingand problem solving that modern computers cannot compete with in many aspects.It has been postulated that a model or a system that is enlightened and supportedby the results from brain research, with a structure similar to that of biological neuralnetworks, could exhibit similar intelligent functionality. Based on this bottom-up postulation, ANN (also known as connectionist models, parallel distributed processingmodels, neuromorphic systems, or simply neural networks) have been developed asbiologically inspired and plausible models for various tasks.

Biological neural networks are composed of many massively interconnected primitive biological neurons. Each neuron possesses axons and dendrites, finger-like projectionsthat enable the neuron to communicate with its neighboring neurons by transmitting andreceiving electrical and chemical signals. More or less resembling the structure of theircounterparts, ANN are composed of interconnected, simple processing elements calledartificial neurons. In processing information, the processing elements in an ANN operateconcurrently and collectively in a similar fashion to biological neurons.ANN possess somedesirable traits similar to those of biological neural networks, such as the capabilities oflearning, self-organization, and fault tolerance.

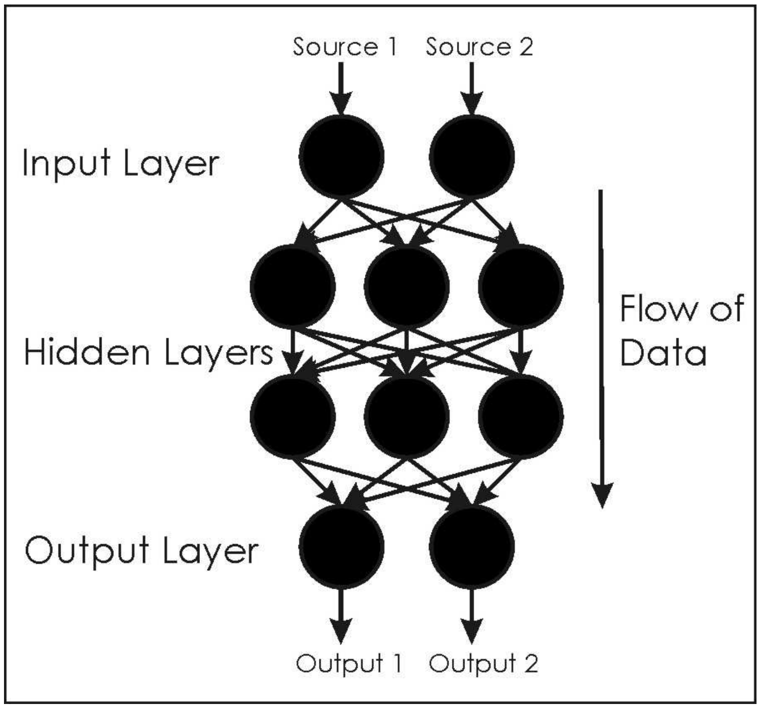

Fig 1: A typical feedforward neural network.

3. LEARNING IN NEURAL NETWORK

An important consideration in an ANN is the use of an appropriate learning algorithm(or training algorithm). Learning algorithms specify the process by which a neural network learns the underlying relationship between input and outputs, or just among theinputs. There are hundreds of them. Learning algorithms in ANN can also be classifiedas supervised learning and unsupervised learning.

Supervised learning uses a set of inputs for which the appropriate (i.e., desired)outputs are known. For example, a historical set of loan applications with the success orfailure of the individual to repay the loan has a set of input parameters and presumedknown outputs. In one type, the difference between the desired and actual outputs isused to calculate corrections to the weights of the neural network. A variation of thisapproach simply acknowledges for each input trial whether the output is correct as thenetwork adjusts weights in an attempt to achieve correct results. Examples of this typeof learning are backpropagation and the Hopfield network.

Apart from the learning modes, the architecture of an artificial neural network can also be categorized in two ways:

- Supervised
  - Recurrent
    - Hopfield
  - Feedforward
    - Non-linear vs linear
    - Backpropagation
    - ML perceptron
    - Boltzman
- Unsupervised
  - Estimators
    - SOFM
  - Extractors
    - ART-1
    - ART-2

4. BACKPROPAGATION

Backpropagation, an abbreviation for "backward propagation of errors", is a common method of training artificial neural networks. From a desired output, the network learns from many inputs, similar to the way a child learns to identify a dog from examples of dogs.It is a supervised learning method, and is a generalization of the delta rule. It requires a dataset of the desired output for many inputs, making up the training set. It is most useful for feed-forward networks (networks that have no feedback, or simply, that have no connections that loop).

Backpropagation requires that the activation function used by the artificial neurons (or "nodes") bedifferentiable.The goal of any supervised learning algorithm is to find a function that best maps a set of inputs to its correct output.An example would be a simple classification task, where the input is an image of an animal, and the correct output would be the name of the animal. Some input and output patterns can be easily learned by single-layer neural networks (i.e. perceptrons). However, these single-layer perceptrons cannot learn some relatively simple patterns, such as those that are not linearly separable.

To illustrate its limitation, consider the example mentioned above of ascribing labels to images. A human may classify an image of an animal by recognizing certain features such as the number of limbs, the texture of the skin (whether it is furry, feathered, scaled, etc.), the size of the animal, and the list goes on. A single-layer neural network however, must learn a function that outputs a label solely using the intensity of the pixels in the image. There is no way for it to learn any abstract features of the input since it is limited to having only one layer. A multi-layered network overcomes this limitation as it can create internal representations and learn different features in each layer. The first layer may be responsible for learning the orientations of lines using the inputs from the individual pixels in the image. The second layer may combine the features learned in the first layer and learn to identify simple shapes such as circles. Each higher layer learns more and more abstract features such as those mentioned above that can be used to classify the image. Each layer finds patterns in the layer below it and it is this ability to create internal representations that are independent of outside input that gives multi-layered networks its power. The goal and motivation for developing the backpropagation algorithm is to find a way to train multi-layered neural networks such that it can learn the appropriate internal representations to allow it to learn any arbitrary mapping of input to output.

5. BACKPROPAGATION ALGORITHM

For better understanding, the backpropagation learning algorithm can be divided into two phases: propagation and weight update.

Phase 1: Propagation

Each propagation involves the following steps:

1. Forward propagation of a training pattern's input through the neural network in order to generate the propagation's output activations.
2. Backward propagation of the propagation's output activations through the neural network using the training pattern target in order to generate the deltas of all output and hidden neurons.

Phase 2: Weight update

For each weight-synapse follow the following steps:

1. Multiply its output delta and input activation to get the gradient of the weight.
2. Bring the weight in the opposite direction of the gradient by subtracting a ratio of it from the weight.

This ratio influences the speed and quality of learning; it is called the *learning rate*. The sign of the gradient of a weight indicates where the error is increasing, this is why the weight must be updated in the opposite direction.

Repeat phase 1 and 2 until the performance of the network is satisfactory.

International Journal of Computing and Business Research (IJCBR)

ISSN (Online) : 2229-6166

Volume 4 Issue 2 May 2013

The learning algorithm includes the following procedures:

1. Initialize wei

2. Read in th

3. Compute the

4. Compute the

5. Change the v … s.

Actual algorith

initialize the weights **in** the network (often small random values)

do

for each example e **in** the training **set**

O = neural-net-output(network, e)*// forward pass*

T = teacher output **for** e

compute error (T - O) at the output units

compute delta_wh**for** all weights from hidden layer to output layer *// backward pass*

compute delta_wi**for** all weights from input layer to hidden layer *// backward pass continued*

update the weights **in** the network

until all examples classified correctly or stopping criterion satisfied

return the network

Fig 2: Back Propagation Algorithm

As the algorithm's name implies, the errors propagate backwards from the output nodes to the input nodes. Technically speaking, backpropagation calculates the gradient of the error of the network regarding the network's modifiable weights. This gradient is almost always used in a simple stochastic gradient descent algorithm to find weights that minimize the error. Often the term "backpropagation" is used in a more general sense, to refer to the entire procedure encompassing both the calculation of the gradient and its use in stochastic gradient descent. Backpropagation usually allows quick convergence on satisfactory local minima for error in the kind of networks to which it is suited.

Backpropagation networks are necessarily multilayer perceptrons (usually with one input, one hidden, and one output layer). In order for the hidden layer to serve any useful function, multilayer networks must have non-linear activation functions for the multiple layers: a multilayer network using only linear activation functions is equivalent to some single layer, linear network. Non-linear activation functions that are commonly used include the logistic function, the softmax function, and the gaussian function.

The backpropagation algorithm for calculating a gradient has been rediscovered a number of times, and is a special case of a more general technique called automatic differentiation in the reverse accumulation mode.

6. DEVELOPING NEURAL NETWORK BASED SYSTEMS

Although the development process of ANN is similar to the structured design methodologies of traditional computer-based information systems, some phases are uniqueor have some unique aspects. In the process described here, we assume that the preliminary steps of system development, such as determining information requirements,conducting a feasibility analysis, and gaining a champion in top management for theproject, have been completed successfully. Such steps are generic to any informationsystem.

As shown in Figure 3, the development process for an ANN application includes nine steps.

In step 1, the data to be used for training and testing the networkare collected. Important considerations are that the particular problem is amenable toneural network solution and that adequate data exist and can be obtained.

In step 2,training data must be identified, and a plan must be made for testing the performanceof the network.

In steps 3 and 4, a network architecture and a learning method are selected. Theavailability of a particular development tool or the capabilities of the developmentpersonnel may determine the type of neural network to be constructed. Also, certainproblem types have demonstrated high success rates with certain configurations. Important considerations are the exact number of neurons and thenumber of layers. Some packages use genetic algorithms to select the network design.There are parameters for tuning the network to the desired learning-performancelevel.

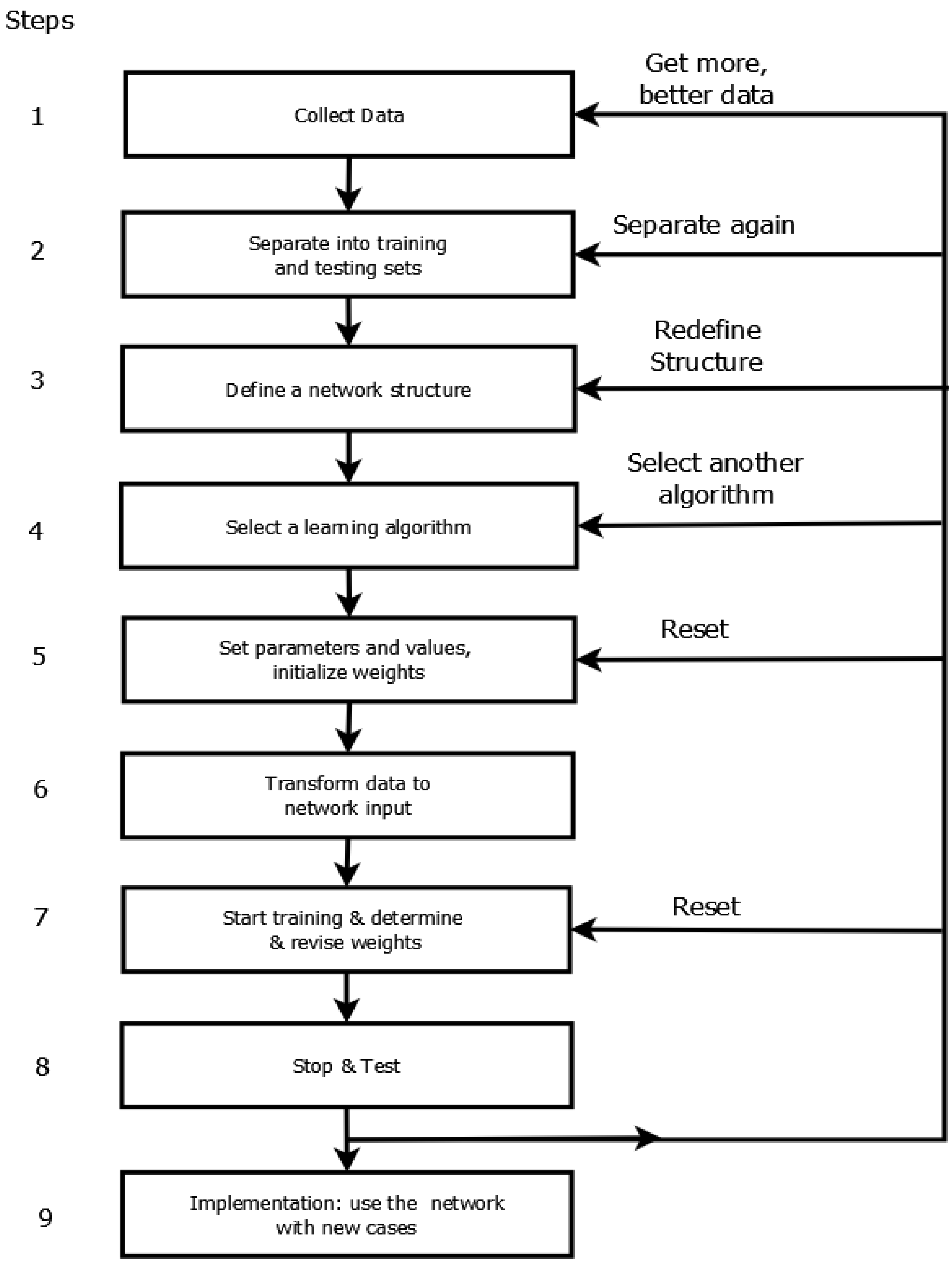

Fig 3: Neural Network System

Part of the process in step 5 is the initialization of the network weights and parameters, followed by the

6.1 DATA COLLECTION AND PREPARATION

The first two steps in the ANN development process involve collecting data and separating them into a training set and a testing set.The training cases are used to adjust theweights, and the testing cases are used for network validation.The data used for trainingand testing must include all the attributes that are useful for solving the problem. Thesystem can only learn as much as the data can tell.Therefore, collection and preparationof data is the most critical step in building a good system.

In general, the more data used, the better. Larger data sets increase processing timeduring training but improve the accuracy of the training and often lead to faster convergence to a good set of weights. For a moderately sized data set, typically 80 percent ofthe data are randomly selected for training and 20 percent are selected for testing; forsmall data sets, typically all the data are used for training and testing; and for large datasets, a sufficiently large sample is taken and treated like a moderately sized data set.

For example, say a bank wants to build a neural network–based system in order touse clients' financial data to determine whether they may go bankrupt.The bank needsto first identify what financial data may be used as inputs and how to obtain them. Fiveattributes may be useful inputs:

(1) working capital/total assets
(2) retained earnings/total assets
(3) earnings before interest and taxes /total assets
(4) market value of equity/total debt
(5) sales/total sales.

The output is a binary variable:bankruptcy or not.

6.2 SELECTION OF NETWORK STRUCTURE

After the training and testing data sets are identified, the next step is to design the structure of the neural networks.This includes the selection of a topology and determinationof

(1) input nodes
(2) output nodes
(3) number of hidden layers
(4) number of hidden nodes.

The multilayer feedforward topology is often used in business applications,although other network models are beginning to find some business use as well.The design of input nodes must be based on the attributes of the data set. In theexample of predicting bankruptcy, for example, the bank might choose a three-layerstructure that includes one input layer, one output layer, and one hidden layer. Theinput layer contains five nodes, each of which is a variable, and the output layer contains a node with 0 for bankrupt and 1 for safe. Determining the number of hiddennodes is tricky. A few heuristics have been proposed, but none of them is unquestionably the best. A typical approach is to choose the average number of input and outputnodes. In the previous case, the hidden node may be set to(5+1)/2=3.

6.3 LEARNING ALGORITHM SELECTION

After the network structure is chosen, we need to find a learning algorithm to identifya set of connection weights that best cover the training data and have the best predictive accuracy. For the feedforward topology we chose for the bankruptcy-predictionproblem, a typical approach is to use the backpropagation algorithm. Because manycommercial packages are available on the market, there is no need to implement thelearning algorithm by ourselves. Instead, we can choose a suitable commercial packageto analyze the data.

6.4 NETWORK TRAINING

Training of ANN is an iterative process that starts from a random set of weights andgradually enhances the fitness of the network model and the known data set. The iteration continues until the error sum is converged to below a preset acceptable level. In thebackpropagation algorithm, two parameters, learning rate and momentum, can beadjusted to control the speed of reaching a solution.These determine the ratio of the difference between the calculated value and the actual value of the training cases. Somesoftware packages may have their own parameters in their learning heuristics to speedup the learning process. It is important to read carefully when using this type of software.

Some data conversion may be necessary in the training process. This includes

(1) changing the data format to meet the requirements of the software

(2) normalizingthe data scale to make the data more comparable

(3) removing problematic data.

When the training data set is ready, it is loaded into the package, and the learning procedure is executed. Depending on the number of nodes and the size of the trainingdata set, reaching a solution may take from a few thousand to millions of iterations.

6.5 TESTING

The availabledata are divided into training and testing data sets. When the training has been completed, it is necessary to test the network.Testing (step 8) examines the performance ofthe derived network model by measuring its ability to classify the testing data correctly.

Black-box testing (i.e., comparing test results to historical results) is the primaryapproach for verifying that inputs produce the appropriate outputs. Error terms can beused to compare results against known benchmark methods.

The network is generally not expected to perform perfectly (zero error is difficult,if not impossible, to attain), and only a certain level of accuracy is really required. Forexample, if 1 means nonbankrupt and 0 means bankrupt, then any output between 0.1 and 1 might indicate a certain likelihood of nonbankrupty. The neural networkapplication is usually an alternative to another method that can be used as a benchmark against which to compare accuracy. For example, a statistical technique such as
multiple regression or another quantitative method may be known to classify inputscorrectly 50 percent

of the time.
The neural network implementation often improves on this. For example, Liang(1992) reported that ANN performance was superior to the performance of multiple discriminant analysis and rule induction. Ainscough and Aronson (1999) investigated theapplication of neural network models in predicting retail sales, given a set of several inputs(e.g., regular price, various promotions). They compared their results to those of multipleregression and improved the adjusted R^2(correlation coefficient) from .5 to .7. If the neural network is replacing manual operations, performance levels and speed of humanprocessing can be the standard for deciding whether the testing phase is successful.
The test plan should include routine cases as well as potentially problematic situations. If the testing reveals large deviations, the training set must be re-examined, andthe training process may have to be repeated (some “bad” data may have to be omittedfrom the input set).
Note that we cannot equate neural network results exactly with those found usingstatistical methods. For example, in stepwise linear regression, input variables aresometimes determined to be insignificant, but because of the nature of neural computing, a neural network uses them to attain higher levels of accuracy.When they are omitted from a neural network model, its performance typically suffers.

6.6 IMPLEMENTATION OF ANN

Implementation of an ANN (step 9) often requires interfaces with other computerbased information systems and user training. Ongoing monitoring and feedback to thedevelopers are recommended for system improvements and long-term success. It is also important to gain the confidence of users and management early in the deployment to ensure that the system is accepted and used properly.

7. SUMMARY OF BACK PROPAGATION ALGORITHM

a) The network consists of three layers: input layer, output layer and the intermediate layer i.e. thehidden layer.
b) These layers comprises of the neurons which are connected to form the entire network.
c) Weights are assigned on the connections which marks the signal strength. The weight values are computed based on the input signal and the error function back propagated to the input layer.
d) The role of hidden layer is to update the weights on the connections based on the input signal and error signal.
e) The algorithm operates in two phases: Initially, the training phase wherein the training data samples are provided at the input layer in order to train the network with predefined set of data classes. Eventually, during the testing phase, the input layer is provided with the test data for prediction of the applied patterns.
f) The desired output is already known to the network. Therefore, if in case the computed output does not match the desired output, the difference in the result is backpropagated to the input layer so that the connection weights of the perceptrons are adjusted in order to reduce the error. The

process is continued until the error is reduced to a negligible amount.

g) This algorithm works in either of the 2 modes: Incremental mode in which each propagation is followed immediately by the weight update or batch mode in which the weight updations take place after many propagations. Usually batch mode is followed due to less time consumption and less no. of propagative iterations.
h) The advantage of using this algorithm is that it is simple to use and well suited to provide a solution to all the complex patterns.
i) Moreover, the implementation of this algorithm is faster and efficient depending upon the amount of input-output data available in the layers.

8. CONCLUSION& FUTURE WORK

In this manuscript, the concept of backpropagation neural network algorithm is studied and a novel approach is highlighted to develop an artificial neural network system. Many machine learning techniques can be used to construct a prediction system but the approach highlighted in this paper is better due to the algorithm complexity and to attain more precise results. The neural network approach is used in various real life applications of data mining, web security, medical diagnosis etc. This approach works as a sequential learning machine taking the input patterns one by one but as a future prospect, work will be carried to generate high level neural networks for prediction system

REFERENCES

1. http://www.webpages.ttu.edu/dleverin/neural_network/neural_networks.html
2. http://www70.homepage.villanova.edu/matthew.liberatore/Mgt2206/turban_online_ch06.pdf
3. http://www.ijser.org/researchpaper%5CBackpropagation-Algorithm-An-Artificial-Neural-Network-Approach-for-Pattern-Recognition.pdf
4. http://en.wikipedia.org/wiki/Backpropagation
5. http://itee.uq.edu.au/~mikael/papers/rn_dallas
6. https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&cad=rja&ved=0CD8QFjAD&url=http%3A%2F%2Fwww.cs.gsu.edu%2F~cscyqz%2Fcourses%2Fai%2FAI-Sample-Paper.doc&ei=YZKKUcPsI8GIrQeThYDIAQ&usg=AFQjCNG3KcFEWmzcMQGI9ajHnf3Y5_v6Pg&bvm=bv.46226182,d.bmk