ISSN 2230- 9373
Journal of
Engineering Innovation
And Research
Volume: IV, Issue: 3

Artificial Neural network for Data mining –A study

Sareeta Mohanty[1], Dr. Maya Nayak[2]

[1]*Department of Information Technology, Orissa Engineering College, Bhubaneswar, India,*
[2]*Department of Information Technology, Orissa Engineering College, Bhubaneswar, India*
[1]*sareeta_mohanty@gmail.com*
[2]*mayanayak3299@yahoo.com*

Abstract— Data mining is defined as the extraction of hidden predictive information from large databases. It finds its application in real world situations such as business, science, technology, and government .A data mining algorithm constitutes a model, a preference criterion, and a search algorithm. The more common model functions in data mining include classification, clustering, rule generation and knowledge discovery. There are many technologies available to data mining practitioners, including Artificial Neural Networks, Regression, and Decision Trees.

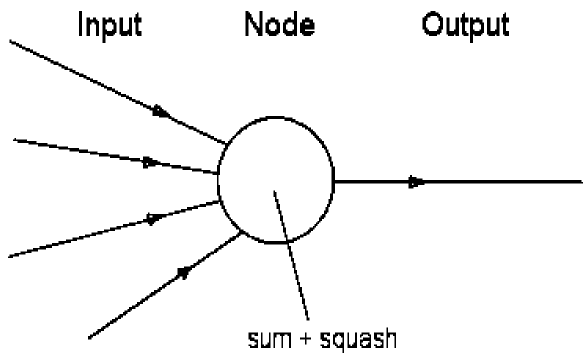
Keywords— **Data mining, Neural network, Neural network training.**

INTRODUCTION

Data mining is defined as the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and systems. overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. In addition to the raw analysis step, it involves database and data management aspects, data processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating. Data mining uses information from past data to analyze the outcome of a particular problem or situation that may arise. Data mining works to analyze data stored in data warehouses that are used to store that data that is being analyzed. That particular data may come from all parts of business, from the production to the management. Data mining interprets its data into real time analysis that can be used to increase sales, promote new product, or delete product that is not value-added to the company. Companies have been collecting data for building massive data warehouses in which to store it. In order to extract actual value from data, data mining is used. Four things are required for data-mine effectively: high-quality data, the "right" data, an adequate sample size and the right tool. There are many tools available to a data mining practitioner. These include decision trees, various types of regression and neural networks.

ARTIFICIAL NEURAL NETWORKS

A neural network is an artificial representation of human brain that tries to simulate its learning process. An artificial neural network is known as Neural network. An artificial



(**Artificial neural network)**

neural network is an efficient information system which resembles in characteristics with biological neural network. ANNs' collective behavior is characterized by their ability to learn , recall and generalized training pattern or data similar to that of human brain. Processing elements are called neurons or artificial neurons in artificial neural network.

ADVANTAGES OF NEURAL NETWORKS

- Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
- Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.
- Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
- Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance.
- However, some network capabilities may be retained even with major network damage.

MCCULLOCHAND PITTS NEURON MODEL

Among numerous neural network models that have been proposed over the years, all share a common building block known as a neuron and a networked interconnection structure. The most widely used neuron model is based on McCulloch and Pitts' work and is illustrated as follows. Each neuron consists of two parts: the net function and the activation function. The net function determines how the network inputs $\{yj\ ;\ 1 \leq j \leq N\}$ are combined inside the neuron. In this figure, a weighted linear combination is adopted:

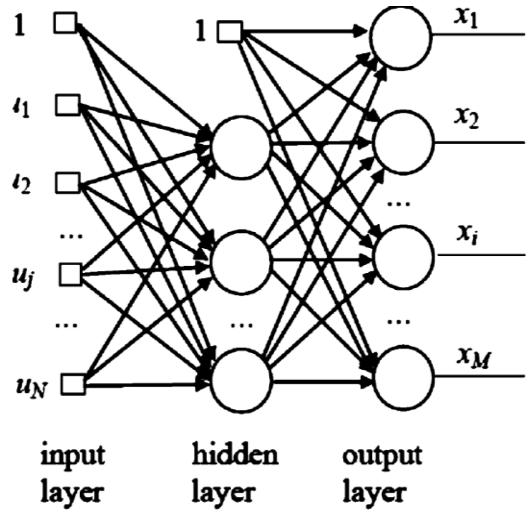
$\{wj\ ;\ 1 \leq j \leq N\}$ are parameters known as synaptic weights. The quantity θ is called the bias

NETWORK ARCHITECTURES

Neural computing is an alternative to programmed computing which is a mathematical model inspired by biological models. This computing system is made up of a number of artificial neurons and a huge number of interconnections between them. According to the structure of the connections the architectures are as follows :-

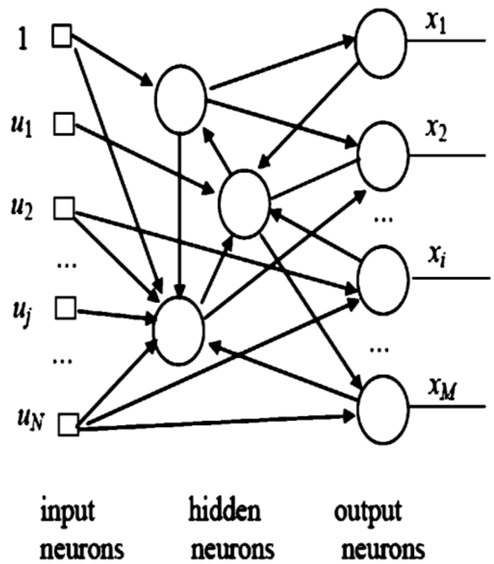
FEED FORWARD NEURAL NETWORK

In feed forward neural networks, the neurons are organized in the form of layers. The neurons in a layer get input from the previous layer and feed their output to the next layer. In this kind of networks connections to the neurons in the same or previous layers are not permitted. The last layer of neurons is called the output layer and the layers between the input and output layers are called the hidden layers. The input layer is made up of special input neurons, transmitting only the applied external input to their outputs .In a network if there is only the layer of input nodes and a single layer of neurons constituting the output layer then they are called single layer network. If there are one or more hidden layers, such networks are called multilayer networks.



RECURRENT NETWORK

The structures, in which connections to the neurons of the same layer or to the previous layers are allowed, are called recurrent networks. For a feed-forward network always exists an assignment of indices to neurons resulting in a triangular weight matrix. Furthermore if the diagonal entries are zero this indicates that there is no self-feedback on the neurons. However in recurrent networks, due to feedback, it is not possible to obtain a triangular weight matrix with any assignment of the indices.



ISSN 2230- 9373

LEARNING METHODS OF ARTIFICIAL NEURAL NETWORK

A **neural network** has to be configured such that the application of a set of inputs produces (either 'direct' or via a relaxation process) the desired set of outputs. Various methods to set the strengths of the connections exist. One way is to set the weights explicitly, using a priori knowledge. Another way is to **'train' the neural network** by feeding it teaching patterns and letting it change its weights according to some learning rule. Various methods of learning are as follows:

Supervised learning or Associative learning in which the network is trained by providing it with input and matching output patterns.

Unsupervised learning or Self-organization in which an (output) unit is trained to respond to clusters of pattern within the input. In this paradigm the system is supposed to discover statistically salient features of the input population. Unlike the supervised learning paradigm, there is no a priori set of categories into which the patterns are to be classified; rather the system must develop its own representation of the input stimuli.

Reinforcement Learning This type of learning may be considered as an intermediate form of the above two types of learning. Here the learning machine does some action on the environment and gets a feedback response from the environment. The learning system grades its action good (rewarding) or bad (punishable) based on the environmental response and accordingly adjusts its parameters. Generally, parameter adjustment is continued until an equilibrium state occurs, following which there will be no more changes in its parameters. The self-organizing neural learning may be categorized under this type of learning.

DIFFERENT TYPES OF NEURAL NETWORK

MULTI LAYER PERCEPTION MODEL

The multilayer perceptron is the most well known and most popular neural network among all the existing neural network paradigms. To introduce the MLP, let us first discuss the perceptron.

PERCEPTION MODEL:

An MLP is a variant of the original perceptron model proposed by Rosenblatt in the 1950s. In the perceptron model, a single neuron with a linear weighted net function and a threshold activation function is employed. The input to this neuron $x = (x1, x2, \ldots, xn)$ is a feature vector in an *n*-dimensional feature space. The net function *u(x)* is the weighted sum of the inputs.

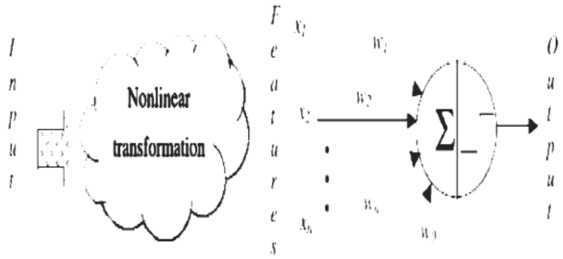
:

$$u(\underline{x}) = w_0 + \sum_{i=1}^{n} w_i x_i$$

the output *y(x)* is obtained from *u(x)* via a threshold activation function.

$$y(\underline{x}) = \begin{cases} 1 & u(\underline{x}) \geq 0 \\ 0 & u(\underline{x}) < 0 \,. \end{cases}$$

$$EI_j = \frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial y_j} \times \frac{dy_j}{dx_j} = EA_j y_j (1 - y_j)$$



The perceptron neuron model can be used for detection and classification

Applications of the Perceptron Neuron Model:

1. The nonlinear transformation that extracts the appropriate feature vector *x* is not specified.
2. The perceptron learning algorithm will not converge for a fixed value of learning rate η if the training feature patterns are not linearly separable.
3. Even though the feature patterns are linearly separable, it is not known how long it takes for the algorithm to converge to a weight vector that corresponds to a hyper plane that separates the feature patterns.

I. LEARNING WITH THE BACKPROPAGATION ALGORITHM

The back propagation algorithm is an involved mathematical tool; however, execution of the training equations is based on iterative processes, and thus is easily implement able

- Weight changes for hidden to output weights.

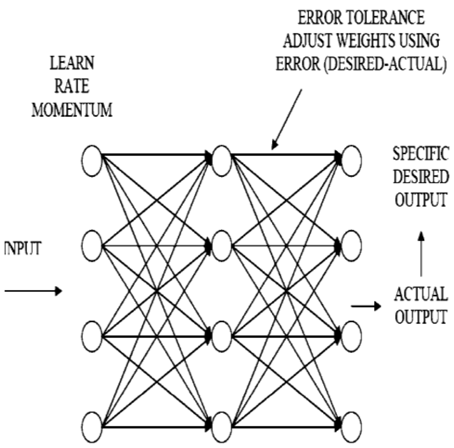
- Weight changes for input to hidden weights error signal is obtained by "back-propagating" error from the output units.

During the training session of the network, a pair of patterns is presented (Xk, Tk), where Xk in the input pattern and Tk is the target or desired pattern. The Xk pattern causes output responses at teach neurons in each layer and, hence, an output Ok at the output layer. At the output layer, the difference between the actual and target outputs yields an error signal. This error signal depends on the values of the weights of the neurons I each layer.This error is minimized, and during this process new values for the weights are obtained. The speed and accuracy of the learning process-that is, the process of updating thee weights-also depends on a factor, known as the learning rate.

Before starting the back propagation learning process, we need the following:

- The set of training patterns, input, and target
- A value for the learning rate
- A criterion that terminates the algorithm
- A methodology for updating weights
- The nonlinearity function (usually the sigmoid)
- Initial weight values (typically small random values) me process then starts by applying the first input Pattern.

The input causes a response to the neurons of the first layer,



which in turn cause a response to the neurons of the next layer, and so on, until a response is obtained at the output layer. That response is then compared with the target response; and the difference (the error signal) is calculated. From the error difference at the Output neurons, the algorithm computes the rate at which the error hanges as the activity level of the neuron changes. So far, the calculations were computed forward (i.e., from the input layer to the output layer). Now, the algorithms steps back one layer before that output layer and recalculate the weights of the output layer (the weights between the last hidden layer and the neurons of the output layer) so that the output error is minimized. The algorithm next calculates the error output at the last hidden layer and computes new values for its weights (the weights between the last and next-to-last hidden layer layers). The algorithm continues calculating the error and Computing new weight values, moving layer by layer backward, toward the input. When the input is reached and the weights do not change, (i.e., when they have reached a steady state), then the algorithm selects the next pair of input-target patterns and repeats the process. Although responses move in a forward direction, weights are calculated by moving backward, hence the name back propagation. The back-propagation algorithm consists of the following steps:

Each Input is then multiplied by a weight that would either inhibit the input . The weighted sum of then inputs in then calculated

First, it computes the total weighted input Xj, using the formula

$$X_j = \sum_i y_i W_{ij}$$

Next, the unit calculates the activity yj using some function of the total weighted input. Typically we use the sigmoid function

$$y_j = \frac{1}{1 + e^{-x_j}}$$

Once the output is calculated, it is compared with the required output and the total Error E is computed. Once the activities of all output units have been determined, the network computes the error E, which is defined by the expression:

$$E = \frac{1}{2}\sum_j (y_j - d_j)^2$$

Where y j is the activity level of the I th unit in the top layer and dj is the desired output of the ith unit. Now the error is propagated backwards

1. Compute how fast the error changes as the activity of an output unit is changed. This error derivative (EA)

is the difference between the actual and the desired activity.

$$EA_j = \frac{\partial E}{\partial y_j} = y_j - d_j$$

2. Compute how fast the error changes as the total input received by an output unit is changed. This quantity (EI) is the answer from step 1 multiplied by the rate at which the output of a unit changes as its total input is changed.

3. Compute how fast the error changes as a weight on the connection into an output unit is changed. This quantity (EW) is the answer from step 2 multiplied by the activity level of the unit from which the connection emanates

$$EW_{ij} = \frac{\partial E}{dW_{ij}} = \frac{\partial E}{\partial x_j} \times \frac{\partial x_j}{dW_{ij}} = EI_j y_i$$

4. Compute how fast the error changes as the activity of a unit in the previous layer is hanged. This crucial step allows back propagation to be applied to multi-layer networks. When the activity of a unit in the previous layer changes, it affects the activities of all the output units to which it is connected. So to compute the overall effect on the error, we add together all these separate effects on output units. But each effect is simple to calculate. It is the answer in step 2 multiplied by the weight on the connection to that output.

$$EA_i = \frac{\partial E}{\partial y_i} = \sum_j \frac{\partial E}{\partial x_j} \times \frac{\partial x_j}{\partial y_i} = \sum_j EI_j W_{ij}$$

By using steps 2 and 4, we can convert the EA's of one layer of units into EA's for the previous layer. This procedure can be repeated to get the EA's for as many previous layers as desired. Once we know the EA of a unit, we can use steps 2 and 3 to compute the EW's on its incoming connection

II. MULTILAYER PERCEPTRON

A multilayer perceptron is used for learning the feature vectors. From experience, 30 neurons in the hidden layer give the best in the hidden layer neurons. The output layer comprises of ten neurons, one neuron for each class. For a particular class of signal, the neuron corresponding to that class should ideally exhibit an output of one while the other neurons exhibit an output of zero. The log-sigmoid transfer function was picked because of its output range (0 to 1) is perfect for learning to output Boolean values. However, in practical cases this is not achievable. To circumvent this problem a post-processing unit is added to select the neuron with the highest excitation as the class of the signal. The

$$E = \sum_{j=1}^{Q} \left(d_j - x_i\right)^2$$

$$\Delta w(k+1) = -h\nabla_w E + a\Delta w(k)$$

network is trained with a back propagation algorithm in which, the error measure E is given as

α is the momentum constant,

η is the learning rate.

The gradient decent algorithm was implemented in batch mode. The performance of a gradient decent algorithm is very dependent on the learning rate. If the learning rate is too large, the training would oscillate back and forth and on the other hand if the learning rate is too small, it would take a long time to reach convergence. To overcome this problem an adaptive learning rate that attempts to keep the step size as large as possible without causing oscillation is used. The learning rate is made responsive to the complexity of the local error surface.

III. PROBABILISTIC NEURAL NETWORK (PNN)

The PNN model is one among the supervised learning networks and has the following features.

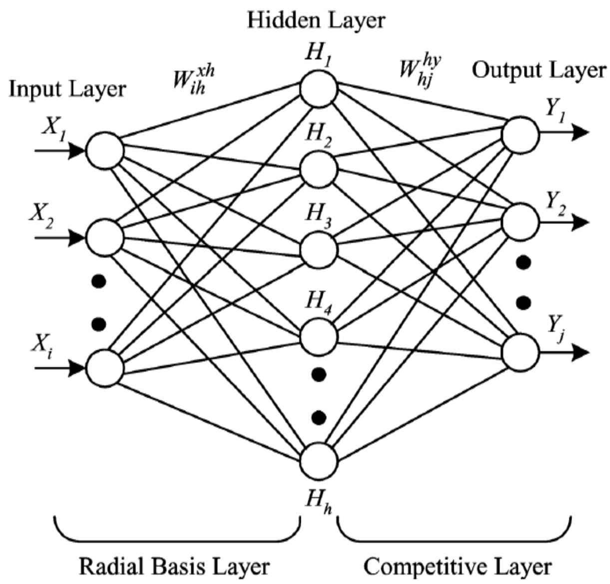
- It is implemented using the probabilistic model, such as Bayesian classifiers.
- A PNN is guaranteed to converge to a Bayesian classifier provided that it is given enough training data.
- No learning processes are required.
- No need to set the initial weights of the network.
- No relationship between learning processes and recalling processes.
- The difference between the inference vector and the target vector are not used to modify the weights of the network.

The learning speed of the PNN model is very fast making it suitable in real time for fault diagnosis and signal classification problems. The figure given below shows the architecture of a PNN model that is composed of the radial basis layer and the competitive layer.

In the signal-classification application, the training examples are classified according to their distribution values of probabilistic density function (pdf), which is the basic principle of the PNN. A simple pdf is as follows:

Modifying and applying to the input vector H of the hidden layer in the PNN is

$$H_{k=\exp(}\frac{-\sum_i(X_i - W_{ih}^{xh}}{2s^2})2$$

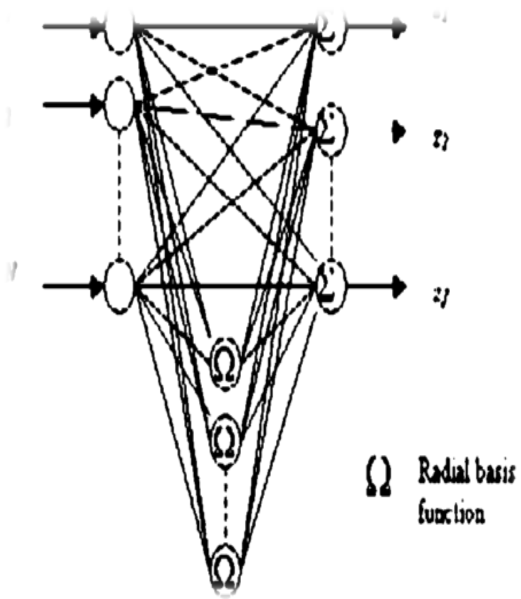


(Probabilistic neural network)

IV. A RADIAL BASIS FUNCTIONAL NEURAL NETWORK (RBFLN)

RADIAL BASIS FUNCTION NETWORK

The Radial Basis Function (RBF) network is a three-layer feed-forward network that uses a linear transfer function for the output units and a nonlinear transfer function (normally the Gaussian) for the hidden layer neurons . Radial basis networks may require more neurons than standard feed-forward back propagation networks, but often they can be designed with lesser time .They perform well when many



training data are available. Much of the inspiration for RBF networks has come from traditional statistical pattern classification techniques. The input layer is simply a fan-out layer and does no processing. The second or hidden layer performs a nonlinear mapping from the input space into a (usually) higher dimensional space whose activation function is selected from a class of functions called basis functions. The final layer performs a simple weighted sum with a linear output. Contrary to BP networks, the weights of the hidden layer basis units (input to hidden layer) are set using some clustering techniques. The idea is that the patterns in the input space form clusters. If the centers of these clusters are known, then the Euclidean distance from the cluster center can be measured. As the input data moves away from the connection weights, the activation value reduces. This distance measure is made nonlinear in such a way that for input data close to a cluster centre gets a value close to 1. Once the hidden layer weights are set, a second phase of training (usually back propagation) is used to adjust the out put.

RADIAL BASIS FUNCTIONAL LINK NEURAL NETWORK

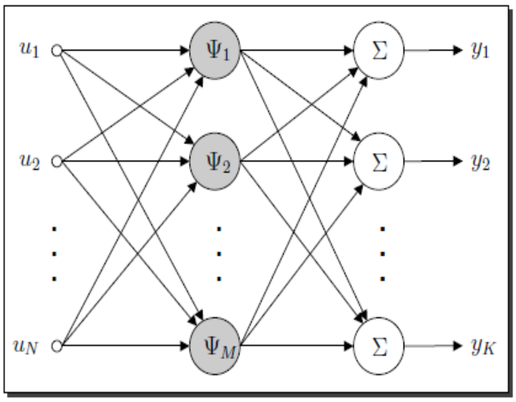
The radial basis function neural networks (RBFNNs) train rapidly, are robust and based on elegant concepts .The radial functions (RBFs) originated in 1964 as potential functions, but were first used for nonlinear regression. The architecture and training algorithms for RBFNNs are simple and they train more quickly than do multiple perceptron (MLP) networks. Unlike MLPs, they allow somewhat for explanation when interpreted as fuzzy rule-based systems. We use RBFs with the random vector functional link nets (RVFLNs) to obtain the powerful radial basis functional link nets (RBFLNs). As RBFNN represents a nonlinear model while the RBFLN includes that nonlinear model as well as a linear model (the direct lines from the input to output nodes) so that the linear parts of a mapping do not need to be approximated by the nonlinear model. Thus the RBFLN is a more complete model of a general nonlinear mapping. Both MLPs and RBFNNs are universal approximators, and thus RBFLNs are also universal approximators because they are more general and include RBFNNs. The radial basis functional link net is a variant of the functional link net or the radial basis function neural network. It is more general than the RBF neural network because it consists of both nonlinear and linear links. In the following section a new algorithm has been presented to train the radial basis RBFLN for pattern recognition of non-stationary power signal database.

MODULAR NEURAL NETWORK

A Modular Neural Network (MNN) is a Neural Network (NN) that consists of several modules, each module carrying out one sub-task of the NN's global task, and all modules functionally integrated. A module can be a sub-structure or a learning sub procedure of the whole network. The network's global task can be any neural network application, e.g., mapping, function approximation, clustering or associative memory application.

WAVELET NEURAL NETWORK

Wavelet neural networks are the networks that combine the theory of wavelets and neural networks into one. A wavelet neural network generally consists of a feed-forward neural network, with one hidden layer, whose activation functions are drawn from an ortho-normal wavelet family. One applications of wavelet neural networks is that of function estimation. If a series of observed values of a function is given, a wavelet network can be trained to learn the composition of that function, and hence calculate an expected value for a given input.



(A wavelet neural network)

ONE DIMENSSIONAL WAVELET NEURAL NETWORK

The simplest form of wavelet neural network is one with a single input and a single output. The hidden layer of neurons consist of wavelons, whose input parameters (possibly fixed) include the wavelet dilation and translation coefficients. These wavelons produce a non-zero output when the input lies within a small area of the input domain. The output of a wavelet neural network is a linear weighted combination of the wavelet activation functions.

MULTIDIMENSIONAL WAVELET NEURAL NETWORK

The input in this case is a multidimensional vector and the wavelons consist of multidimensional wavelet activation functions. They will produce a non-zero output when the input vector lies within a small area of the multidimensional input space. The output of the wavelet neural network is one or more linear combinations of these multidimensional wavelets.

V. NEURAL NETWORK APPLICATION IN DATA MINING

Neural networks can be used to model complex relationships between inputs and outputs or to find patterns in data. Using neural networks as a tool, data warehousing firms are extracting information from datasets in the process known as data mining. Prediction, Clustering, Association Rules. Classification and prediction is a predictive model, but clustering and association rules are descriptive models. The most common action in data mining is classification. It recognizes patterns that describe the group to which an item belongs. It does this by examining existing items that already have been classified and inferring a set of rules. Similar to classification is clustering. The major difference being that no groups have been predefined. Prediction is the construction and use of a model to assess the class of an unlabeled object or to assess the value or value ranges of a given object is likely to have. The next application is forecasting. This is different from predictions because it estimates the future value of continuous variables based on patterns within the data. Neural networks, depending on the architecture, provide associations, classifications, clusters, prediction and forecasting to the data mining industry. Financial forecasting is of considerable practical interest. Due to neural networks can mine valuable information from a mass of history information and be efficiently used in financial areas, so the applications of neural networks to financial forecasting have been very popular over the last few years. Some researches show that neural networks performed better than conventional statistical approaches in financial forecasting and information on associations, classifications, clusters, and forecasting. The back propagation algorithm performs learning on a feed-forward neural network.

XII.CONCLUSION

Neural network is considered as a promising tool for data mining. The neural networks are becoming very popular with data mining practitioners, particularly in medical research, finance and marketing .This is because they have proven their predictive power through comparison with other statistical techniques using real data sets.

REFERENCE

Ajith Abraham ―Artificial Neural Networks, *Oklahoma State University, Stillwater, OK, USA*

Neural Networks, Volume 39, March 2013, Ozyildirim, B.M.; Avci, M. In Elseiver

Neural Networks, Volume 44, August 2013, Hosseini, A.; Wang, J.; Hosseini, S.M. in,Elseiver

Jacek M. Zurada, "Introduction to Artificial Neural Systems", pp 32-36, 2006.

NEURAL NETWORKS IN DATA MINING.**in JATIT 2009**

Data Mining Techniques: Arjun K. Pujari

Artificial Neural Networks: Galgotia Publication

IEEE Transactions on Neural Networks; "DataMining in a Soft Computing Framework: ASurvey", Authors: Sushmita Mitra, Sankar K. Pal and Pabitra Mitra. (January 2002, Vol. 13, No. 1)

Neural Networks based Data Mining and Knowledge Discovery in Inventory Applications: Kanti Bansal, Sanjeev Vadhavkar, Amar Gupta

Data Mining Techniques: Electronic textbook, Statsoft: http:// www.statsoftinc.com/ textbook/ stdatmin.html#neural

Adriaans P, Zantinge D. Data mining [M].Addision_Wesley Longman,1996.

Elements of Artificial Neural Networks: Kishan Mehrotra, Chilukuri K. Mohan, Sanjay Ranka

[1] Neural Networks based Data Mining and Knowledge Discovery in Inventory Applications: Kanti Bansal, Sanjeev, Vadhavkar, Amar Gupta

[2] Guan Li, Liang Hongjun. Data warehouse and data mining. Microcomputer Applications. 1999

Adriaans P, Zantinge D. Data mining [M]. Addision_Wesley Longman,1996.

Sareeta Mohanty received her B.E degree in Computer Science and Engineering from Utkal University, Bhubaneswar, Odisha, INDIA in 2004. She received M.Tech in Computer Science from Utkal University, Bhubaneswar, Odisha, INDIA in 2012. She is a Asst. Professor with Department of Information Technology, Orissa Engineering College, Bhubaneswar since 2006. Her research interests include digital signal processing, soft computing, design and analysis of algorithm, theory of computation, artificial intelligence.

Dr. Maya Nayak received her B.E. degree in Computer Science and Engineering from Bangalore University, Karnataka, INDIA in 1993. She received her M.E. degree in Communication System Engg. From UCE ,Burla , Odisha in 1997. She achieved the Ph.D. in Computer Science and Engg. from BPUT , Odisha , in 2009. Backed with an experience of 19 years in the field of teaching, research and administration, she is working as Professor and chairing the position of addi. Principal, and HOD , Dept. of Information Technology and P.G. (Computer Science & Engineering) in Orissa Engineering College, Bhubaneswar. She has published a number of papers in National and International Journals, conferences and delivered invited talks in the field of Soft Computing & Data Mining. Her research interests include Data mining and Soft computing.