

# Neural Networks in Data Mining

Jini E. R.

Msc Computer Science  
Department of Computer Science  
St.Thomas College(Autonomous), Thrissur.

Sunil Sunny

Assistant Professor  
Department of Computer Science  
St .Thomas College(Autonomous),Thrissur

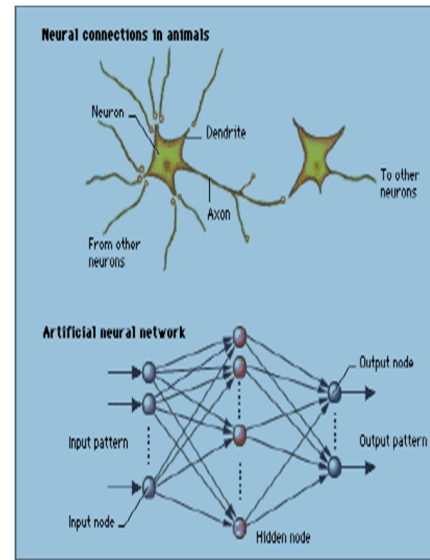
**Abstract** - Data mining is the exploration and analysis of large data sets, in order to discover meaningful pattern and rules. There are many technologies available to data mining practitioners, including Artificial Neural Networks, Genetics, Fuzzy logic and Decision Trees. Many practitioners are wary of Neural Networks due to their black box nature, even though they have proven themselves in many situations. This paper is an overview of artificial neural networks and questions their position as a preferred tool by data mining practitioners.

## I. INTRODUCTION

Data mining is the exploration and analysis of large data sets, in order to discover meaningful pattern and rules. The key idea is to find effective way to combine the computer's power to process the data with the human eye's ability to detect patterns. It is the term used to describe the process of extracting value from a database. A data-warehouse is a location where information is stored. The type of data stored depends largely on the type of industry and the company. Many companies store every piece of data they have collected, while others are more ruthless in what they deem to be "important". Four things are required to 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.

## II. ARTIFICIAL NEURAL NETWORKS

An **artificial neural network** (ANN), often just called a "neural network" (NN), is a mathematical model or computational model based on biological neural networks, in other words, is an emulation of biological neural system. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase [1]. There are two types of NN based on learning technique, they can be supervised where output values are known beforehand and unsupervised where output values are not known [2].



### A. Architecture

Human brain has over 100 billion interconnected neurons. Neurons use this interconnected network to pass information with each other using electric and chemical signals. Although it may seem that neurons are fully connected, two neurons actually do not touch each other. They are separated by tiny gap called synapse. Each neuron process information and then it can "connect" to as many as 50000 other neurons to exchange information. A typical neuron would have four components: Dendrites gather inputs from other neurons and when a certain threshold is reached they generate a nonlinear response. Basic ANN is composed of three layers input, hidden and output layers. Each layer can have number of nodes and nodes from input layer are connected to the nodes from hidden layer. Nodes from hidden layer are connected to the nodes from output layer. The connections represent the weights between nodes [2].

### B. Neural Network Topologies

**Feed forward neural network** : In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network.

**Recurrent network**: Recurrent neural networks that do contain feedback connections. Contrary to feed forward networks, recurrent neural networks (RNs) are models with bi-directional data flow. While a feed forward network propagates data linearly from input to output, RNs also propagate data from later processing stages to earlier stages [1].

### C. Number of Nodes and Layers

Choosing number of nodes for each layer will depend on problem NN is trying to solve, types of data network is dealing with, quality of data and other parameters. Number of input and output nodes depends on training set in hand. Choosing number of nodes in hidden layer is a challenging task. If there are too many nodes in hidden layer, number of possible computations that algorithm has to deal with increases. Picking just few nodes in hidden layer can prevent the algorithm of its learning ability [2].

### D. Setting Weights

The way to control NN is by setting and adjusting weights between nodes. Initial weights are usually set at some random numbers and then they are adjusted during NN training. Logic behind weight update is quite simple. During the NN training weights are updated after iterations. Finding combinations of weights that will help us minimize errors should be the main aim when setting weights [2].

### E. Running and Training NN

Running the network consists of a forward pass and backward pass. In forward pass, outputs are calculated and compared with desired outputs. Error from desired and actual output is calculated. In backward pass, this error is used to alter the weights in the network in order to reduce the size of the error. Forward pass and backward pass are repeated until the error is low enough. When training NN, we are finding network with set of examples that have inputs and desired outputs. If we have a set of 1000 samples, we could use 100 of them to train the network and 900 to test our model [2].

### F. Activation function

Activation functions are needed for hidden layer of the NN to introduce nonlinearity. Activation function can be linear, threshold or sigmoid function. Sigmoid activation function is usually used in hidden layer [2].

## III. NEURAL NETWORKS IN DATA MINING

In more practical terms neural networks are non-linear statistical data modeling tools. They 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 harvesting information from datasets in the process known as data mining. The difference between these data warehouses and ordinary databases is that there is actual manipulation and cross-fertilization of the data helping users makes more informed decisions. Neural networks essentially comprise three pieces: the architecture or model; the learning algorithm; and the activation functions. Neural networks are programmed or "trained" to ". . . store, recognize, and associatively retrieve patterns or database entries; to solve combinatorial optimization problems; to filter noise from measurement data; to control ill-defined problems; in summary, to estimate sampled functions when we do not know the form of the functions." It is precisely these two abilities (pattern recognition and function estimation) which make artificial neural networks (ANN) so prevalent a utility in data mining. As data sets grow to

massive sizes, the need for automated processing becomes clear. With their "model-free" estimators and their dual nature, neural networks serve data mining in a myriad of ways.

Data mining is the business of answering questions that you've not asked yet. Data mining reaches deep into databases. Data mining tasks can be classified into two categories:

- Descriptive
- Predictive

Descriptive data mining provides information to understand what is happening inside the data without a predetermined idea. Predictive data mining allows the user to submit records with unknown field values, and the system will guess the unknown values based on previous patterns discovered from the database [1].

### A. Tasks of data mining

Data mining as a term used for the specific classes of six activities or tasks as follows:

- Classification.
- Estimation
- Prediction
- Affinity grouping or association rules
- Clustering
- Description and visualization

NNs are important data mining tool used for classification and clustering.

1) *Classification*: 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.

2) *Estimation*: Estimation deals with continuously valued outcomes. Given some input data, we use estimation to come up with a value for some unknown continuous variables such as income, height or credit card balance.

3) *Prediction*: Any prediction can be thought of as classification or estimation. The difference is one of emphasis.

4) *Association Rules*: An association rule is a rule which implies certain association relationships among a set of objects (such as "occur together" or "one implies the other") in a database.

5) *Clustering*: Clustering is the task of segmenting a diverse group into a number of similar subgroups or clusters. In clustering, there are no predefined classes.

6) *Description and Visualization*: Data visualization is a powerful form of descriptive data mining. It is not always easy to come up with meaningful visualizations, but the right picture really can be worth a thousand association rules since the human beings are extremely practiced at extracting meaning from visual scenes [3]. In data warehouses, neural networks are just one of the tools used in data mining. ANNs are used to find patterns in the data and to infer rules from them. Neural networks are useful in providing information on associations, classifications, clusters, and forecasting. The back propagation algorithm performs learning on a feed-forward neural network [1].

*B. Feed Forward neural network*

One of the simplest feed forward neural networks (FFNN), such as in Figure 1, consists of three layers: an input layer, hidden layer and output layer. In each layer there are one or more processing elements (PEs). PEs is meant to simulate the neurons in the brain and this is why they are often referred to as neurons or nodes. A PE receives inputs from either the outside world or the previous layer. There are connections between the PEs in each layer that have a weight (parameter) associated with them. This weight is adjusted during training. Information only travels in the forward direction through the network - there are no feedback loops.

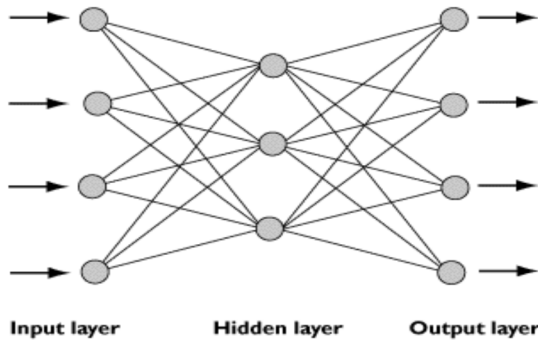


Figure 1

The simplified process for training a FFNN is as follows:

1. Input data is presented to the network and propagated through the network until it reaches the output layer. This forward process produces a predicted output.
2. The predicted output is subtracted from the actual output and an error value for the networks is calculated.
3. The neural network then uses supervised learning, which in most cases is back propagation, to train the network. Back propagation is a learning algorithm for adjusting the weights. It starts with the weights between the output layer PE's and the last hidden layer PE's and works backwards through the network.
4. Once back propagation has finished, the forward process starts again, and this cycle is continued until the error between predicted and actual outputs is minimized [1].

IV. THE BACK PROPAGATION(BP) ALGORITHM

**Back propagation, or propagation of error,** is a common method of teaching artificial neural networks how to perform a given task. The back propagation algorithm is used in layered feed forward ANNs. This means that the artificial neurons are organized in layers, and send their signals "forward", and then the errors are propagated backwards. The back propagation algorithm uses supervised learning, which means that we provide the algorithm with examples of the inputs and outputs we want the network to compute, and then the error (difference between actual and expected results) is calculated. The idea of the back propagation algorithm is to reduce this error, until the ANN *learns* the training data [1].

The algorithm can be decomposed into four steps.

- Feed forward computation
- Back propagation to output layer
- Back propagation to hidden layer
- Weight updates

The algorithm is stopped when the value of the error function has become sufficiently small [2].

**ALGORITHM**

Given: A set of input-output vector pairs.

1. Let A be the number of units in input layer as determined by the training input vectors. Let C be the number of units in output. Now choose B, the number of units in the hidden Layer. We denote the activation levels of the units in the input layer by  $x_j$ , in the hidden layer by  $h_j$  and in the output layer by  $o_j$ .  $W1_{ij}$ = weights from the input layer to the hidden layer, where i indexes the input unit and j indexes the hidden unit.  
 $W2_{ij}$ = weights from the hidden layer to the output layer, where i indexes the hidden unit and j indexes the output unit.
2. Initialize the weights in the network.  
 $W1_{ij} = \text{random}(-0.1, 0.1) \quad \forall \quad i=0 \dots A, j=1 \dots B$   
 $W2_{ij} = \text{random}(-0.1, 0.1) \quad \forall \quad i=0 \dots B, j=1 \dots C$
3. Initialize the activations of the thresholding units should never change.  
 $x_0 = 1.0 \quad h_0 = 1.0$
4. Choose an input-output pair. Suppose the input vector is  $x_i$  and output vector is  $y_i$ . Assign activation levels to the input units.

5. Propagate the activations from the units in the input layer to the hidden layer using the activation function

$$h_j = \frac{1}{1 + e^{-\sum_{i=0}^A W1_{ij}x_i}} \quad \forall \quad j=1 \dots B$$

6. Propagate the activations from the units in the hidden layer to the output layer using the activation function

$$o_j = \frac{1}{1 + e^{-\sum_{i=0}^B W2_{ij}h_i}} \quad \forall \quad j=1 \dots C$$

7. Compute the errors of the units in the output layer denoted  $\delta 2_j$ .

$$\delta 2_j = o_j(1 - o_j)(y_j - o_j) \quad \forall \quad j=1 \dots C \dots \dots \dots (1)$$

8. Compute the errors of the units in the hidden layer denoted  $\delta 1_j$ .

$$\delta 1_j = h_j(1 - h_j) \sum_{i=1}^c \delta 2_i \cdot W2_{ji}$$

$$\forall j=1 \dots B$$

9. Adjust the weights between the hidden layer and output layer. The learning rate is denoted  $\eta$ . A reasonable value of  $\eta$  is 0.35.

$$\Delta W2_{ij} = \eta \cdot \delta 2_j \cdot h_i \quad \forall \quad i=0 \dots B, j=1 \dots C. \dots \dots (2)$$

10. Adjust the weights between the input layer and hidden layer.

$$\Delta W1_{ij} = \eta \cdot \delta 1_j \cdot x_i \quad \forall \quad i=0 \dots A, j=1 \dots B. \dots \dots (3)$$

11. Go to step 4 and repeat. When all the input-output pairs have been to the network, one epoch has been completed. Repeat step 4 to 10 for as many epochs as desired [4].

The speed of learning can be increased by modifying the weight modification steps 9 and 10 to include a momentum term  $\alpha$ . The weight update formulas become:

$$\Delta W_{2ij}(t + 1) = \eta \cdot f'_{2j} \cdot h_i + \alpha \Delta W_{2ij}(t) \dots (4)$$

$$\Delta W_{1ij}(t + 1) = \eta \cdot f'_{1j} \cdot h_i + \alpha \Delta W_{1ij}(t) \dots (5)$$

A. Example

NN on figure 2 has two nodes (N00, N01) in input layer, two nodes (N10, N11) in hidden layer and one node (N20) in output. Input layer nodes are connected to hidden layer nodes with weights (W01-W04). Hidden layer nodes are connected to output layer nodes with weights (W10 and W11). The values that were given to weights are taken randomly and will be changed during BP iterations. Table with input node values and desired output are given in figure 3. Sigmoid function formula is [2]

$$f(x) = \frac{1}{1+e^{-x}}$$

1) *Feed Forward Computation:* Feedforward computation is a two step process. First part is getting the values of the hidden layer nodes and second part is to compute values of output layer.

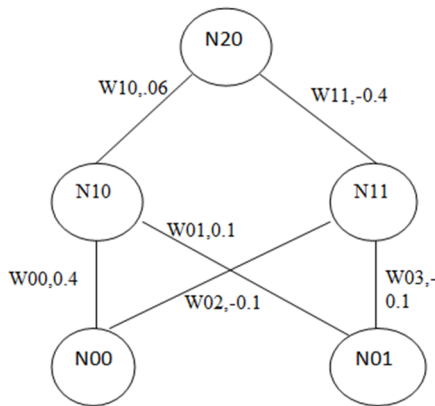


Figure 2

n00	n01	output n20
1	1	1
1	0	0
0	1	0
0	0	0

Figure 3

$\eta = 0.45, \alpha = 0.9$

Using sigmoid function compute

$$N10 = f(x1) = f(w00 * n00 + w01 * n01) = f(.4 + .1) = f(.5) = 0.622549$$

$$N11 = f(x2) = f(W02 * n00 + W03 * n01) = f(-.1 - .1) = f(-.2) = 0.450166$$

When hidden layer values are calculated, network propagates values into output layer(N20).

$$N20 = f(x3) = f(W10 * n10 + W11 * n11) = f(.06 * 0.622549 + (-.4 * 0.450166)) = f(-0.1427188) = 0.464381$$

Forward pass is completed.

2) *Backpropagation to the Output Layer:*

Next step is to calculate the error of N20 node using(1)  
 $N20 \text{ error} = 0.464381(1 - 0.464381)(1 - 0.464381) = 0.133225$

Error is propagated from output layer to hidden layer first. Before weights can be updated, rate of change needs to be found. Using (2) and (3)

$$\Delta W10 = \eta * N20 \text{ error} * N10$$

$$\Delta W10 = .45 * .133225 * .622459 = 0.37317$$

Now new weight for W10 can be calculated using

$$W10_{new} = W10_{old} + \Delta W10 + (\alpha * \Delta(t-1)) =$$

$$.06 + .37317 + .9 * 0 = 0.097139.$$

$$\Delta W11 = \eta * N20 \text{ error} * N11$$

$$\Delta W11 = .45 * .133225 * .450166 = 0.026988$$

$$W11_{new} = W11_{old} + \Delta W11 + (\alpha * \Delta(t-1)) = -0.4 + 0.026988 = -0.373012$$

The value of  $\Delta(t-1)$  is previous delta change of the weight.

3) *Backpropagation to the Hidden Layer:* Now errors has to be propagated from hidden layer down to the input layer, then propagating error from output to hidden layer. Output of nodes N10 and N11 was unknown.

$$N10 \text{ error} = N20 \text{ error} * W10_{new} = 0.133225 * 0.097317 = 0.012965$$

$$N11 \text{ error} = N20 \text{ error} * W11_{new} = 0.133225 * (-0.373012) = -0.049706$$

Once error for hidden layer nodes is known, weights between input and hidden layer can be updated

$$\Delta W00 = \eta * N10 \text{ error} * N00 = 0.45 * 0.012965 = 0.005834$$

$$\Delta W01 = \eta * N10 \text{ error} * N01 = 0.45 * 0.012965 * 1 = 0.005834$$

$$\Delta W02 = \eta * N11 \text{ error} * N00 = 0.45 * -0.049706 * 1 = -0.022368$$

$$\Delta W03 = \eta * N11 \text{ error} * N01 = 0.45 * -0.049706 * 1 = -0.022368$$

Then the new weights between input and hidden layer becomes

$$W00_{new} = W00_{old} + \Delta W00 + (\alpha * \Delta(t-1)) = 0.4 + 0.005834 + 0.9 * 0 = 0.405834$$

$$W01_{new} = W01_{old} + \Delta W01 + (\alpha * \Delta(t-1)) = 0.1 + 0.005834 + 0.9 * 0 = 0.105384$$

$$W02_{new} = W02_{old} + \Delta W02 + (\alpha * \Delta(t-1)) = -0.1 + -0.022368 + 0.9 * 0 = -0.122368$$

$$W03_{new} = W03_{old} + \Delta W03 + (\alpha * \Delta(t-1)) = -0.1 + -0.022368 + 0.9 * 0 = -0.122368$$

4) *Weight Updates:* Important thing is not to update any weights until all errors have been calculated. It is easy to forget this and if new weights were used while calculating errors, results would not be valid. Here is quick second pass using new weights to see if error has decreased.

$$N10 = f(x1) = f(W00 * N00 + W01 * N01) = f(.506) = 0.62386831$$

$$N11 = f(x2) = f(W02 * N00 + W03 * N01) = f(-0.244) = 0.4393008$$

$$N20 = f(x3) = f(W10 * N10 + W11 * N11) = f(-.103343991) = 0.474186972$$

$$N20 = f(x3) = f(W10 * N10 + W11 * N11) = f(-.103343991) = 0.474186972$$

Forward pass is completed.



Next step is to calculate the error of N20 node using(1).  
From the table in figure b output should be 1.

$$\begin{aligned} N20 \text{ error} &= 0.474186972(1 - 0.474186972)(1 - \\ &0.474186972) \\ &= 0.131102901 \end{aligned}$$

So after iteration, calculated error was 0.133225 and new calculated error is 0.131102901. This algorithm has improved not by much but this should give good idea on how BP algorithm works. The algorithm is stopped when the value of the error function has become sufficiently small [2].

#### V. ADVANTAGES OF NEURAL NETWORKS

- High Accuracy: Neural networks are able to Approximate complex non-linear mappings
- Noise Tolerance: Neural networks are very Flexible with respect to incomplete, missing And noisy data.
- Independence from prior assumptions: Neural Networks do not make a priori assumptions About the distribution of the data, or the form Of interactions between factors.
- Ease of maintenance: Neural networks can be Updated with fresh data, making them useful For dynamic environments.
- Neural networks can be implemented in Parallel hardware
- When an element of the neural network fails, it can continue without any problem by their parallel nature [1].

#### VI. APPLICATIONS OF NEURAL NETWORKS

Fraud detection, telecommunications, medicine, marketing, bankruptcy prediction, insurance, the list goes on. The following are examples of where neural networks have been used.

##### Accounting

Identifying tax fraud

Enhancing auditing by finding irregularities

##### Finance

Signature and bank note verification

Risk Management

Foreign exchange rate forecasting

Bankruptcy prediction

Customer credit scoring

Credit card approval and fraud detection

Forecasting economic turning points

Bond rating and trading

Loan approvals

Economic and financial forecasting

##### Marketing

Classification of consumer spending pattern

New product analysis

Identification of customer characteristics

Sale forecasts

##### Human resources

Predicting employee's performance and behavior

Determining personnel resource requirements [1]

#### VII. DESIGN PROBLEMS

- There are no general methods to determine the optimal number of neurones necessary for solving any problem.
- It is difficult to select a training data set which fully describes the problem to be solved [1].

#### VIII. SOLUTIONS TO IMPROVE ANN PERFORMANCE

- Designing neural networks using genetic algorithms
- Neuro-Fuzzy systems

#### IX. CONCLUSION

NNs are important data mining tool used for classification and clustering. It is an attempt to build machine that will mimic brain activities and be able to learn.

NN is an interconnected network that resembles human brain. The most important characteristic of NN is its ability to learn. When presented with training set where input and values are known, NN model could be created to help with classifying new data. Results that are achieved by using NN are encouraging especially in some fields like pattern recognition and function estimation. BP algorithm is the most popular algorithm used in NN.

Artificial Neural Networks offer qualitative methods for business and economic systems that traditional quantitative tools in statistics and econometrics cannot quantify due to the complexity in translating the systems into precise mathematical functions. Hence, the use of neural networks in data mining is a promising field of research especially given the ready availability of large mass of data sets and the reported ability of neural networks to detect and assimilate relationships between a large numbers of variables.

#### X. REFERENCES

- [1] Dr. Yashpal Singh ,Alok Singh Chauhan "Neural In Networks Data Mining" , India , (Journal of Theoretical and Applied Information Technology)2005.
- [2] Mirza Cilimkovic "Neural Networks and Back Propagation Algorithm" ,Ireland.
- [3] Anand V. Saurkar, Vaibhav Bhujade, Priti Bhagat, Amit Khaparde " A Review Paper on Various Data Mining Techniques" ,India,(International Journal of Research in Computer Science and Software Engineering).
- [4] E.Rich, K.Knight and S.B.Nair , Artificial Intelligence, 3<sup>rd</sup> Edn. TMGH, New Delhi, 2009.
- [5] Vidushi Sharma, Sachin Rai, Anurag Dev "A Comprehensive Study of Artificial Neural Networks", India,(International Journal of Research in Computer Science and Software Engineering).
- [6] Sumit Garg, Arvind K. Sharma "Comparative Analysis of Data Mining Techniques on Educational Dataset", India, (International Journal of Computer Applications).
- [7] O.S. Eluyode and Dipo Theophilus Akomolafe, "Comparative study of biological and artificial neural networks", Nigeria,( European Journal of Applied Engineering and Scientific Research).
- [8] Gaurab Tewary "EFFECTIVE DATA MINING FOR PROPER MINING CLASSIFICATION USING NEURAL NETWORKS", India, (International Journal of Data Mining & Knowledge Management Process).