

# International Journal of Statistics and Applied Mathematics

ISSN: 2456-1452  
Maths 2018; 3(1): 460-464  
© 2018 Stats & Maths  
www.mathsjournal.com  
Received: 18-11-2017  
Accepted: 19-12-2017

## **K Sreenivasulu**

Lecturer in Statistics,  
Sri Sarvodaya College, Nellore,  
Andhra Pradesh, India

## **G Mokesh Rayalu**

Assistant Professor,  
Department of Mathematics,  
School of Advanced sciences,  
VIT, Vellore, Tamil Nadu, India

## Data mining techniques in forecasting methods

**K Sreenivasulu and G Mokesh Rayalu**

### **Abstract**

Over the last few recent years, there has been much research directed at predicting the future and making better decisions. This research has led to many developments in forecasting methods. Most of these methodological advances have been based on statistical techniques. Statistical methods and neural networks are commonly used for time series prediction. Empirical results have shown that Neural Networks outperform linear regression specially in the case of more complex behaviour of dependent variables like nonlinear, dynamic and chaotic behaviours. Neural networks are reliable for modeling nonlinear, dynamic market predictions. Neural Network makes very few assumptions as opposed to normality assumptions commonly found in statistical methods. Neural network can perform prediction after learning the underlying relationship between the input variables and outputs. From a statistician's point of view, neural networks are analogous to nonparametric, nonlinear regression models.

**Keywords:** Data mining techniques, forecasting methods

### **1. Introduction**

Machine learning should be construed as a subfield of computer science i.e., soft computing which is evolved from the learning of pattern recognition and computational learning theory. Machine learning may border on Data Mining where the latter focuses more on exploratory data analysis. It may also be known as unsupervised learning. Machine learning is a field of study that gives to computers the capacity to learn without openly programmed. Machine learning brings out the study and construction of algorithms which could make any one to learn from and.

These algorithms operate by creating a model or inputs in order to make data riven decisions or predictions. Machine learning is closely associated to computational statistics which focuses in prediction making with the use of computers. It has closer ties with mathematical optimization, which yields to methods, theory and application domain. Where programming and designing is not feasible, Machine learning is used.

In the field of data analysis, machine learning is a devise that lend themselves to prediction in commercial use. Hence it is also known as predictive analysis. Such analysis helps researchers, engineers and data analysts to bring out reliable decision and results. Through learning from historical relationships and trends in the data, it brings out hidden insights.

In machine learning, classification is the difficulty in allotting to which of a set of categories a new observation belongs. It is based on the training data set which contains observations whose relationship is observed. Classification is significant in pattern recognition. In machine learning, classification is supervised learning where as clustering is unsupervised. There is a difference between supervised learning and unsupervised learning. In the former, the training data is pairs of input data (vectors) and desired outputs. But in unsupervised learning, there is no priority output. A classifier is one which does classification, particularly in a concrete implementation. It also sometimes refer to mathematical function implemented by a classification algorithm. In machine learning, three terminologies are important. They are instances, features and classes. Instances are observation, features are explanatory variables and classes are the possible categories to be predicted.

### **Correspondence**

#### **G Mokesh Rayalu**

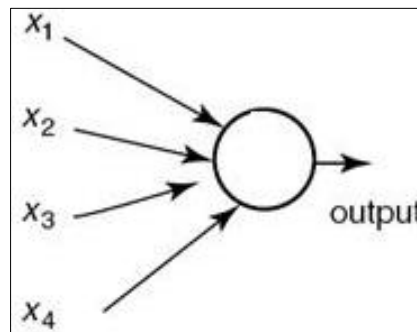
Assistant Professor,  
Department of Mathematics,  
School of Advanced sciences,  
VIT, Vellore, Tamil Nadu, India

### 1.1 Review of neural network methodology

In the literature, we can find different reviews on the neural network methodology based on their own applications. In this section, a brief description was given on components of ANN with its architecture and general steps involved in forecasting methodology.

### 1.2 Components of neural networks

**i. Nodes or neurons:** The nodes can be seen as computational units. They receive inputs, and process them to obtain an output. This processing might be very simple such as summing the inputs, or quite complex like node might contain another network or involves huge iterative calculations.



### Artificial neuron

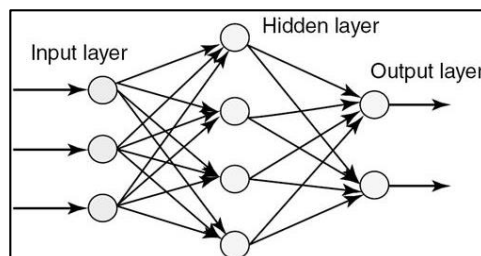
**ii Connectors:** The connections determine the information flow between nodes. They can be unidirectional, when the information flows only in one sense, and bidirectional, when the information flows in either sense.

**iii Input and output nodes:** Input nodes are the starting points of input data in to the system. There are no preceding nodes to this input node. Each ANN may contain more than one input node also. Similarly output nodes are final resultants of the forecasting process.

### 1.3 Architecture of neural networks

The basic architecture consists of three types of neuron layers: input, hidden, and output layers. In the forward networks, the signal flow is from input to output units, which is termed as feed-forward network. There are several other neural network architectures (Elman network, adaptive resonance theory maps, competitive networks, etc.), depending on the properties and requirement of the application.

A simple architecture can be represented diagrammatically as follows.



### Multilayered artificial neural network

Neural network architecture comprises of the following four main elements

- a) Processing Elements (Neurons)
- b) Connection between the elements
- c) Weights of the connection
- d) Activation function

### 1.4 Steps involved in neural network forecasting

We can not find a constant and regular procedure for neural network models forecasting. We can generalize the fitting process in the following steps.

**Step 1:** Usually the modeling of ANN begins by postulating an initial network based on general modeling practices. One regular practice is to add very important variables in the model based on user knowledge on the data and respective scenario.

**Step 2:** Only one hidden layer is needed for a artificial neural network to be a universal function approximator to a continuous function so often only one hidden layer is used; however, more than one hidden layer maybe used since overall fewer neurons will be required. If the function to be approximated is discontinuous, the model will require at most two hidden layers.

**Step 3:** Determine the weights which are usually initialized with random values. The observations are then input to the network and parameters adjusted by several available methods. The adjustment process is repeated until the error converges on a minimum point.

**Step 4:** The results calculated in step 3 are now transformed as output nodes. The number of nodes in the output layer corresponds to the number of variables to be predicted

**2. Neural network learning algorithms**

A neural network has to be configured such that the application of a set of inputs produces 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. The following are some of the important learning algorithms which are applied more regularly in practice.

**2.1 Hebbian learning**

Hebb (1949), proposed a learning which was based on the modification of synaptic connections between neurons. The basic idea is that if two neurons are active simultaneously, their interconnection must be strengthened. If we consider a single layer net, one of the interconnected neurons will be an input unit and one an output unit. If the data are represented in bipolar form, it is easy to express the desired weight update as

$$\omega_i(\text{new}) = \omega_i(\text{old}) + x_i o$$

where  $o$  is the desired output for  $i = 1$  to  $n$ (inputs).

Unfortunately, plain Hebbian learning continually strengthens its weights without bound (unless the input data is properly normalized).

**2.2 Perceptron learning rule**

The perceptron is a single layer neural network whose weights and biases could be trained to produce a correct target vector when presented with the corresponding input vector. The training technique used is called the *perceptron learning rule*. Perceptrons are especially suited for simple problems in pattern classification.

Suppose we have a set of learning samples consisting of an input vector  $x$  and a desired output  $d(k)$ . For a classification task, the  $d(k)$  is usually +1 or -1. The perceptron-learning rule is very simple and can be stated as follows:

1. Start with random weights for the connections.
2. Select an input vector  $x$  from the set of training samples.
3. If output  $y_k \neq d(k)$  (the perceptron gives an incorrect response), modify all connections  $w_i$  according to:  $\delta w_i = \eta(d_k - y_k)x_i$ ; ( $\eta$  = learning rate).
4. Go back to step 2.

Note that the procedure is very similar to the Hebb rule; the only difference is that when the network responds correctly, no connection weights are modified.

**2.3 Conjugate Gradient Learning Algorithm**

Conjugate gradient algorithm is one of the popular search methods to minimize the network output error in conjugate directions. Conjugate gradient method uses orthogonal and linearly independent non zero vector. Two vectors  $d_i$  and  $d_j$  are mutually G-Conjugate if

$$d_i^T G d_j = 0 \quad \text{for } i \neq j \quad \dots (2.3.1)$$

The algorithm firstly developed to minimize the quadratic function of  $n$ -variables

$$F(w) = C - b^T W + 1/2 W^T G W \quad \dots (2.3.2)$$

Where  $W$  is vector with  $n$  elements and  $G$  is  $n \times n$  symmetric and positive definite matrix. The algorithm was then extended to minimization of general linear functions interpretation (2.3.2) as a second order Taylor series expansion of the objective function  $G$ . A starting point  $W$  is selected and the first search direction  $d_1$  is said to be negative gradient  $g_1$  i.e., ( $d_1 = -g_1$ ) conjugate gradient method is to minimize differential function 4.4.2 by generating a sequence of approximation  $W_{k+1}$  iteratively according to

$$W_{k+1} = W_k + \alpha_k d_k \quad \dots (2.3.3)$$

$$d_{k+1} = -g_{k+1} + \beta_k d_k \quad \dots (2.3.4)$$

$\alpha$  and  $\beta$  are momentum terms to avoid oscillations

Let  $\mu = \frac{1}{1 + \beta_x}$  then equation (2.3.4) can be written as

$$d_{k+1} = \frac{1}{\mu_x} [\mu(-g_{k+1}) + (1-\mu)d_k] \dots (2.3.5)$$

The value of  $\alpha_k$  can be determined by line search techniques. Such as golden search and brent algorithm in the way that  $f(W_k + \alpha_k d_k)$  is minimized along the direction  $d_k$ .  $\beta_k$  can be calculated by any of the following formula.

$$\beta_k = \frac{g_{k+1}^T (g_{k+1} - g_k)}{d_k^T (g_{k+1} - g_k)}$$

Polka and Reeves formula is given by

$$\beta_k = \frac{g_{k+1}^T [g_{k+1} - g_k]}{g_k^T g_k}$$

Shanno derives the formula for  $d_{k+1}$  by considering conjugate method as memory less quasi-Newton method.

$$d_{k+1} = g_{k+1} \left[ \left( 1 + \frac{y_k^T y_k}{p_k^T y_k} \right) \left( \frac{p_k^T g_k}{p_k^T y_k} - \frac{y_k^T g_k}{p_k^T y_k} \right) \right] p_k + \frac{p_k^T g_k}{p_k^T y_k} y_k$$

Where  $p_k = \alpha_k d_k$  and  $y_k = g_{k+1} - g_k$

The method performs an approximation line minimization in a descent direction in order to increase numerical stability.

The summary of conjugate gradient algorithm is described below

1. Set  $K=1$ , Initialize  $W$
2. Compute  $g_1 = \nabla f(w_1)$
3. Set  $d_1 = -g_1$
4. Compute  $\alpha_k = \text{avg min } [f(w_k + \alpha_k d_k)]$
5. Update neighbor vector by  $W_{k+1} = W_k + \alpha_k d_k$
6. If network error is less than a pre set minimum value (or) the maximum number of iterations one reached, then stop. Else go to step 7.
7. If  $k+1 > n$  then  $W_1 = W_{k+1}$ ,  $K=1$  and go to step 2.

### 2.4 Multiple Linear Regression Weight Initializations

Back propagation technique has risk of being stopped at local minimum. Multiple linear regression Wight method, neighbor between hidden layer and output layer are obtained by multiple linear regression.

The weight  $W_{ij}$  between input mode  $i$  and output mode  $j$  is initialized by uniform randomization. Once input  $x_i^s$  of sample  $s$  has been fed into input mode and  $w_{ij}^s$  have been assigned values, output value  $R_j^s$  of the hidden layer can be calculated as

$$Y_s = \left( \sum_j V_j R_j^s \right) \tag{2.4.1}$$

Where  $V_j$  is weight between the hidden layer and output layer.

Assume that sigmoid function  $f(x) = \frac{1}{1 + e^{-x}}$  is used as the transfer function of the network. By Taylor expansion  $f(x) \cong \frac{1}{2} + \frac{x}{4} \dots$  (2.4.2)

Applying linear approximation is (2.4.1) in (2.4.2) we have the following linear approximate relationship between output  $y$  and  $v_j^s$ .

$$y^s = \frac{1}{2} + \frac{1}{4} \left( \sum_j V_j R_j^s \right) \text{ (or) } 4y^s - 2 = V_1 R_1^s + \dots + V_m R_m^s \tag{2.4.3}$$

Where  $m$  is number of hidden layers. The set of equation in (4.4.8) is multiple linear regression model.  $R_j^s$  are considered as repressors,  $V_j^s$  can be estimated by standard regression method.

### 3. Conclusions

The field of neural networks is very dynamic and vast so that the opportunities for future research exist in many aspects, including data preprocessing and representation, architecture selection, and application. In the case of training algorithm for ANN, we need to develop algorithms such that the overall performance should be improved further in terms of time for completion and forecast errors. We need to develop methods and techniques to auto refining of the model parameters when new data being added in to the

system. It saves lot of human efforts and cost involved in the project for forecasting tasks. In this paper, we described the several latest models for forecasting like Artificial Neural networks along with their merits and demerits.

#### 4. References

1. Akaike H. A New Look at the Statistical Model Identification, I.E.E.E. Transactions on Automatic Control, AC. 1974; 19:716-723.
2. Azoff EM. Neural network time series forecasting of financial market. John Wiley & Sons Ltd, 1994.
3. Baum EB, Haussler D. What size net gives valid generalization?, Neural Computation. 1988; 1:151-160.
4. Bishop CM. Neural Networks for Pattern Recognition, Oxford University, 1995.
5. David Skapura M. Building Neural Networks, ACM Press, Addison-Wesley.
6. Eitan Michael Azoff. Reducing Error in Neural Network Time Series Forecasting, Neural Computing & Applications, 1993, 240-247.
7. Emad Saad W, Danil VP, Donald CW. Advanced Neural Network Training Methods for Low False Alarm Stock Trend Prediction, Proc. of World Congress on Neural Networks, Washing D.C, 1996.
8. Fama Eugene F, French Kenneth R. Dividend yields and expected stock returns. Journal of Financial Economics, Elsevier. 1988; 22(1):3-25.
9. Gia-Shuh Jang, Feipei Lai. Intelligent Stock Market Prediction System Using Dual Adaptive-Structure Neural Networks, Proc. Int'l Conference on Artificial Intelligence Applications on Wall Street, 1993.
10. Guido Deboeck J. Trading on the Edge-Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets, Wiley, 1994.
11. Hong Tan, Danil VP, Donald CW. Probabilistic and Time-Delay Neural Network Techniques for Conservative Short-Term Stock Trend Prediction, Proc. of World Congress on Neural Networks, Washington D.C, 1995.
12. Kalyani Dacha. Causal Modeling of Stock Market Prices using Neural Networks and Multiple Regression: A Comparison Report, Finance India, 2007; xxi(3):923-930.
13. Koyck LM. Distributed Lags and Investment Analysis, Amsterdam: North Holland, 1954.
14. Lakonishok *et al.* The Journal of Finance. 1994; 49(5):1541-1578.
15. Leorey Marquez, Tim Hill, Reginald Worthley, William Remus. Neural Network Models as an Alternate to Regression, Proc. of IEEE 24th Annual Hawaii Int'l Conference on System Sciences, 1991; 129-135:VI.
16. Schwarz G. Estimating the Dimension of a Model, Annals of Statistics, 1978; 6:461-464.
17. Sharda R, Patil R. Neural Networks as forecasting experts: an empirical test, Proceedings of the 1990 International Joint Conference on Neural Networks, Washington DC, USA. 1990; I:491-494.
18. Smirlock Michael, Starks Laura T. A Further Examination of Stock Price Changes and Transactions Volume. Journal of Financial Research. 1985; 8:217-225.
19. Walczak S. An empirical analysis of data requirements for financial forecasting with neural networks. Journal of Management Information Systems. 2001; 17(4):203-222.
20. Widrow BSD. Sterns, Adaptive Signal Processing, Englewood Cliffs, NJ: Prentice-Hall, 1985.