iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** XII **Month of publication:** December 2017

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 5 Issue XII December 2017- Available at www.ijraset.com

Application of Neural Networks in Forecasting Methods

C.Narayana[1], J. Prabhakara Naik[2], G.Y.Mythili[3], K.Vasu[4], G. Mokesh Rayalu[5]
[1]*Assistant professor, Department of Mathematics, Sriharsha Institute of P. G. Studies, Nellore*
[2] *Lecturer in Statistics, SSBN Degree & PG College(Autonomous), Anantapur*
[3]*Assistant Professor, Department of Mathematics, School of Advanced sciences, VIT, Vellore*
[4]*Assistant Professor, Vidya Educational Institutions, Yadamari , Chittoor*
[5]*Assistant Professor, Department of Mathematics, School of Advanced sciences, VIT, Vellore*

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.

I. INTRODUCTION

A neural network has processing elements called neurons which are used for processing the information. For a simple network, these neurons are arranged in two layers namely input layer and hidden layer. One hidden layer can approximately map any type of non-linear calculation (Weatherford, Lawrence R., 2002). The issue of determining the number of neurons required in the hidden layer is a matter of trial and error process or experimentation.

Each neuron has an activation function which is applied to the inputs provided to the neuron. For propagation of information between the various layers, the weight age connections are used. So ANN also called interconnections network. The weight of these connections depends upon which training algorithm is used to train the network.

The inputs that neurons receive from other neurons are summed up and are provided to the transfer function to get the output from neurons. There are various types of transfer functions used viz. Sigmoid Function, Hyperbolic tangent function, exponential function etc.

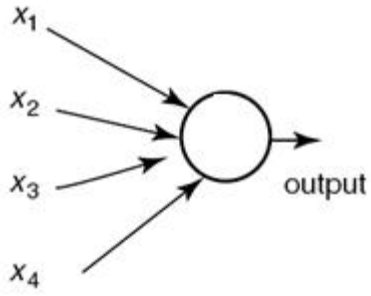
In order for ANN to learn, various training algorithms are used. The selection of an optimal learning algorithm is an open problem (Walczak et al.(1999). Some of them are gradient descent with momentum, gradient descent with adaptive learning rate, quasi-Newton, conjugate gradient, Scaled conjugate gradient and Levenberg-Marquardt.

A. 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.

B. Components of neural networks

1) *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.



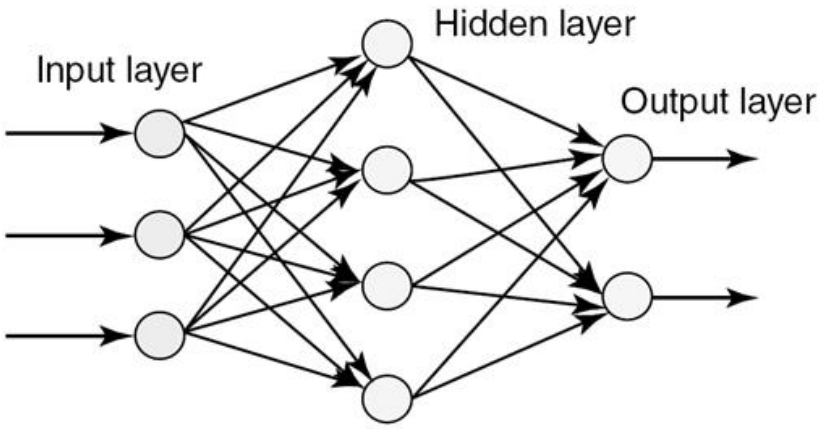
Artificial neuron

2) *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.
3) *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 my contain more that one input node also. Similarly output nodes are final resultants of the forecasting process.

B. 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

1) Processing Elements (Neurons)
2) Connection between the elements
3) Weights of the connection
4) Activation function

C. 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.

1) *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.
2) *Step 2:* Only one hidden layer is needed for a artificial neural network to be a universal function approximator to a continuous function (Hornik et al., 1989) 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 (Cybenko, 1988).
3) *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.
4) *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

II. 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.

A. 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

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

whereois 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).

B. 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 *perceptronlearning 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 wiaccording to:$\delta wi = \eta(d_k - y_k)x_i$; ($\eta$ = 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.

C. Conjugate Gradient Learning Algorithm

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 sectors d_i and d_j are neutrally G-Conjugate if

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

The alogorithm firstly defeloped to minise the quaachatic function of n-variables

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

Where W is vector with n elements and G is nxnsymetirc and positive definite matrix. The alogorithm was than extended to minimization of general linear functions interpretation (2.3.2) as a secured 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 g1 i.e., ($d_i = -g_1$) consugate gradient method is to minimse differential function 4.4.2 by generating a sequence of appropriation W_{k+1}interatively according to

$W_{k+1}=W_k+\alpha_k d_k$ … (2.3.3)

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

αandβ are momention terms to avoid oscillations

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

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

The value of αk 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. β_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-nefon 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 appropriation line minimization in a descent direction in order to increase numerical stability.

The summary of conjugate gradient alogarithum is described below

1) Set K=1, Initialize W

2) Compute $g_1 = \nabla f(w_1)$

3) Set $d_i = -g_i$

4) Compute α_k = avg min $[f(w_k+\alpha_k d_k]$

5) Update neigher 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.

D. Multiple Linear Regression Weight Initializations

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

weightW_{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

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

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) $\cong \frac{1}{2}+\frac{x}{4}$ … (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_js.

$$y^s = \frac{1}{2}+\frac{1}{4}\left(\sum_j V_j R_j^s\right) \text{ (or) } 4y^s - 2 = V_1R_1^s + \ldots + V_mR_m^s \quad \ldots (2.4.3)$$

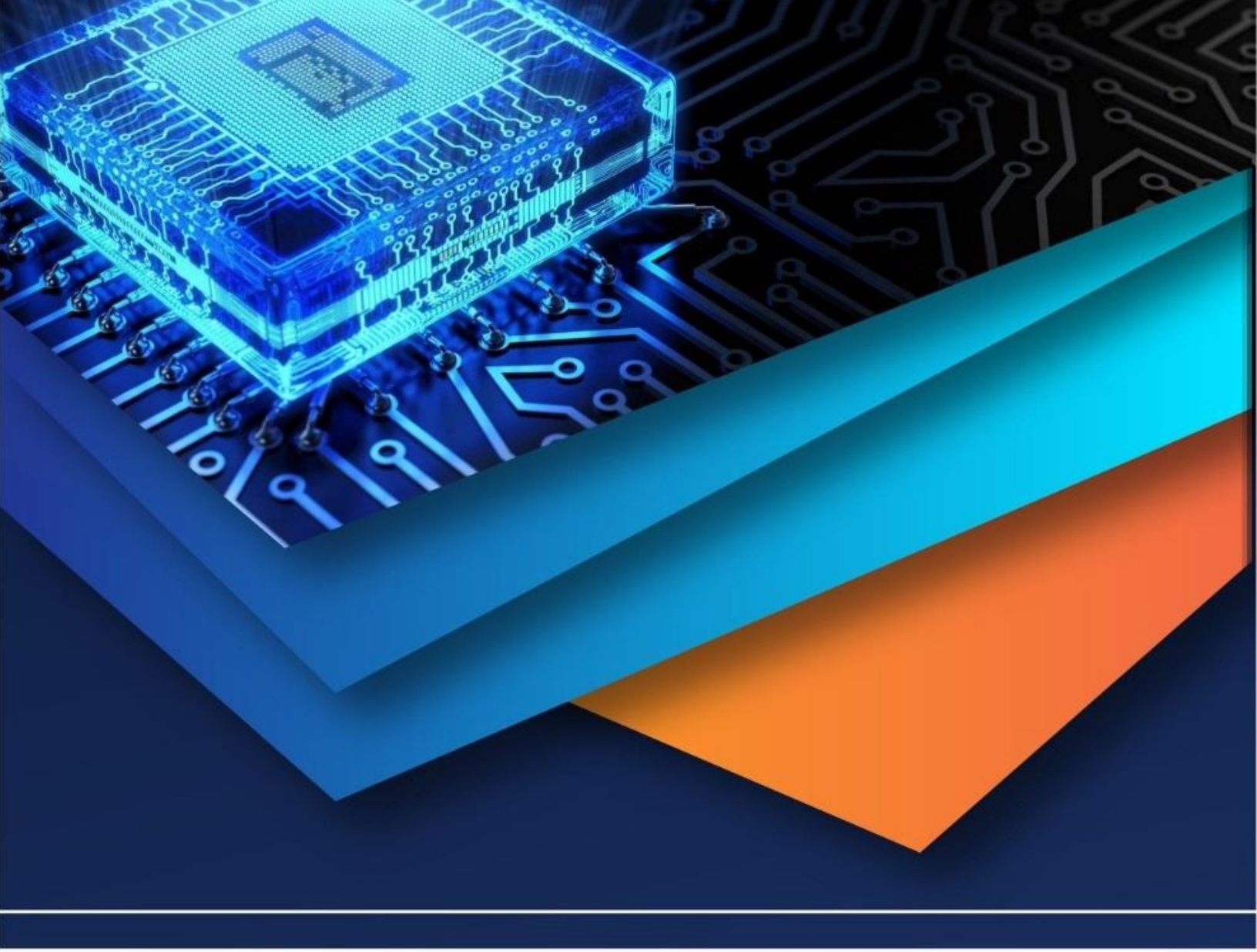
Where m is number of hidden layers. The set of equation in (4.4.8) is multiple linear regression model. R_j^s are considered as repressors, V_j^s can be estimated by standard regression method.

III. CONCLUSIONS

The field of neural networks is very dynamic and vast so that the opportunities for future research exist in many aspects, including data pre-processing 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.

REFERENCES

[1] Akaike, H. (1974). "A New Look at the Statistical Model Identification," I.E.E.E. Transactions on Automatic Control, AC 19, 716-723.
[2] Azoff. E. M. (1994) “Neural network time series forecasting of financial market.” John Wiley & Sons Ltd.
[3] Baum E.B., Haussler D. (1988) What size net gives valid generalization?, Neural Computation 1, pp.151- 160.
[4] Bishop, C.M. (1995) “Neural Networks for Pattern Recognition,” Oxford University.
[5] David M. Skapura. Building Neural Networks, ACM Press, Addison-Wesley.
[6] Eitan Michael Azoff. (1993) Reducing Error in Neural Network Time Series Forecasting, Neural Computing & Applications, pp.240-247.
[7] Emad W. Saad, Danil V.P., Donald C.W. (1996) Advanced Neural Network Training Methods for Low False Alarm Stock Trend Prediction, Proc. of World Congress on Neural Networks, Washing D.C.
[8] Fama, F. Eugene & French, R. Kenneth, (1988) “Dividend yields and expected stock returns,” Journal of Financial Economics, Elsevier, vol. 22(1), pp. 3-25.
[9] Gia-Shuh Jang, Feipei Lai. (1993) Intelligent Stock Market Prediction System Using Dual Adaptive-Structure Neural Networks, Proc. Int’l Conference on Artificial Intelligence Applications on Wall Street.
[10] Guido J. Deboeck. (1994) Trading on the Edge - Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets, Wiley.
[11] Hong Tan, Danil V. P., Donald C. W. (1995) Probabilistic and Time-Delay Neural Network Techniques for Conservative Short-Term Stock Trend Prediction, Proc. of World Congress on Neural Networks, Washington D.C.
[12] Kalyani Dacha (2007) “Causal Modeling of Stock Market Prices using Neural Networks and Multiple Regression: A Comparison Report,” Finance India,Vol. xxi , No.3, pp. 923-930.
[13] Koyck, L.M. (1954), Distributed Lags and Investment Analysis, Amsterdam: North Holland.
[14] Lakonishok et al. (1994) “The Journal of Finance,” Volume 49, Issue 5, pp. 1541-1578. Dec.
[15] Leorey Marquez, Tim Hill, Reginald Worthley, William Remus. (1991) Neural Network Models as an Alternate to Regression, Proc. of IEEE 24th Annual Hawaii Int’l Conference on System Sciences, pp.129-135, Vol VI.
[16] Schwarz, G. (1978). "Estimating the Dimension of a Model," Annals of Statistics, 6, 461-464.
[17] Sharda, R. and R. Patil, (1990) “Neural Networks as forecasting experts: an empirical test,” Proceedings of the 1990 International Joint Conference on Neural Networks, Vol-I, pp. 491-494, Washington DC, USA.
[18] Smirlock Michael and Starks, T. Laura, (1985) “A Further Examination of Stock Price Changes and Transactions Volume,” Journal of Financial Research 8, pp. 217-225.
[19] Walczak, S. (2001). An empirical analysis of data requirements for financial forecasting with neural networks. Journal of Management Information Systems, 17 (4), 203-222.
[20] Widrow, B. and S.D (1985). Sterns, Adaptive Signal Processing, Englewood Cliffs, NJ: Prentice-Hall.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY