Stock Price Prediction using Mean Normalization and Neural Networks

Rohit Kriplani[1] Prof. Sankar Ganesh[2]
[1]B.Tech Student [2]Assistant Professor
[1,2]Department of Electronics & Communication Engineering
[1,2]VIT University, Vellore-632014

Abstract— This paper proposes neural networks a non-linear approach to predict future trends of stock market. The model is made up of neural networks whose data are pre-processed using mean normalization technique scaled between -1 to 1. The model is created on MATLAB in the form of simulation work. The method used here is multilayer perceptron taking different number of hidden nodes into account. The Reliance stocks is considered. The sum of squares of differences between target and predicted errors is computed with 4-3-1, 4-5-1 and 4-7-1 neural network architecture.

Key words: Normalization, Backpropagation algorithm, error computation

I. INTRODUCTION

"Artificial intelligence is the study of computation that make it possible to perceive, reason, and act." – Winston [1992]

In recent stock market forecasting trends, neural networks are gaining more attention, may be because of the fact that it learns from the past data and closely fits with any kind of trend. Profit in stock market mostly depends upon prediction. For example, considering a case, if an investor knows open price, high and low price of stocks of a particular day then he/she can predict whether to buy or to sell the stocks (shares) by analyzing past values and then predicting the closing price . Similar learning mechanism is followed by neural networks.

The motivation for research in this field is that it finds large applications in almost all the domains whether it is computer science, economics, data analyses, business and communication field, etc. Recently, big data analysis is one of the major research area that uses neural networks and machine learning algorithms.

A. *Benefits of Neural Networks:*

- Nonlinearity
- Adaptivity
- Fault tolerance
- Neuro biological analogy and many more.

In the proposed system we used five years' stock data of Reliance. The data is trained in three different neural network structures.

The paper is organized as follows: Section I consisted the introduction part. Section II consists of description about Artificial Neural Networks. In Section III, we described data pre-processing technique and methodology. Section IV consists of description about stock dataset and neural network architecture with different hidden nodes. In Section V we have shown the simulated results and observation based on the output. And finally, we concluded in Section VI along with the references.

II. ARTIFICIAL NEURAL NETWORKS

This section consists of comparison between ANN and biological neuron. Artificial neurons are highly correlated to biological neurons in terms of functioning. In biological terms, neurons are the sources to transfer the information from one end to the brain and vice-versa. In biological system receptors convert stimuli from the external environment into electrical pulses that pass information to the neural net (i.e., brain). The effectors convert electrical impulses generated by neural networks into discernible responses as system outputs. The information is transmitted from one end to the other through Synapses which is a complex process.

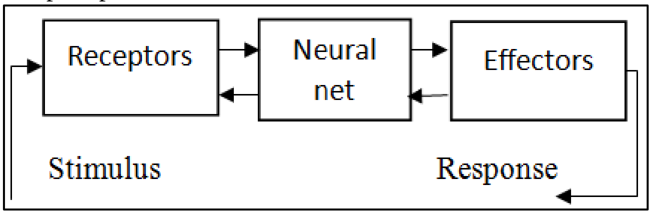

Fig. 1: Block diagram representation of nervous system.

ANN is a similar kind of model which is being made artificially. The model consists of input neurons specifically a signal x_j at the input of synapse j connected to neuron k, is multiplied by synaptic weight w_{kj}. The neural network also consists of externally applied bias.

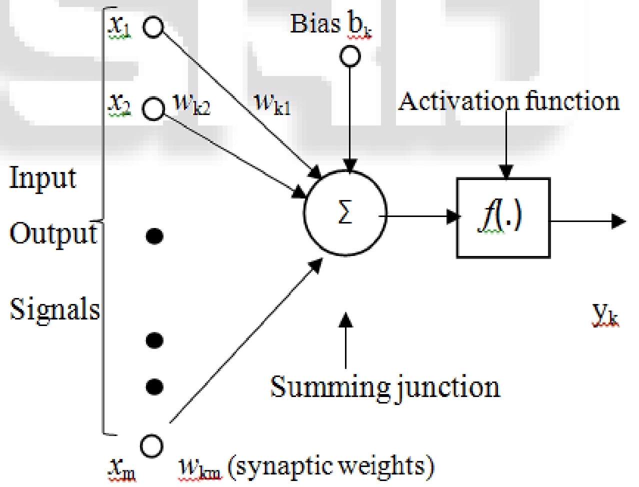

Fig. 2: Artificial Neural network model

In mathematical terms the output is

$$u_k = \sum_{j=1}^{m} Wkj\, Xj \qquad (1.1)$$

and

$$y_k = f\,(u_k + b_k) \qquad (1.2)$$

- where, $x_1, x_2, x_3, \ldots, x_m$ are the input signals,
- $w_{k1}, w_{k2}, w_{k3}, \ldots\ldots, w_{km}$ are respective synaptic weights.
- $f\,()$ represents activation function.
- Use of bias b_k has the effect of applying an affine transformation to the output u_k of the linear combiner in the model given by

$$v_k = u_k + b_k \qquad (1.3)$$

III. Data Pre-processing & Methodology

The input data which is used in this research can be obtained from YAHOO FINANCE. The following procedure is followed for creating various neural network architecture:

1) Data selection and pre-processing.
2) Neural network structure and algorithm.
3) Data training and validation.
4) Comparison with various neural network structure and selection of best network structure.

Elaboration of various steps given above:

A. *Data Selection and Pre-Processing:*

The Reliance data contains opening price, closing price, high and low prices from the previous time step. Out of these, three attributes are considered and one bias is added. The output consists of closing price index. The data is divided into two parts; 70% for training and remaining 30% for validating. While training the data, the minimum sum of squares between target and predicted output is also computed in parallel. In this technique we ignored the days without trading and only considered the data for trading days.

The three input attributes are pre-processed by using following equation:

$$X\_normal = (x-\mu)/(\max(x)-\min(x)) \quad (2)$$

Where, X-normal is normalized value
x is actual data value
μ is mean value of input data set
max(x) is maximum value of input data set
min(x) is minimum value of input data set

It is important to approximate each feature in the range of $-1 \le x_i \le 1$ because this feature scaling technique helps the algorithm to converge faster. Also if normalization is not applied then the value of variable could be too large for the network to process. Normalization also reduces noise and fluctuations present in the data.

B. *Neural Network Structure and Algorithm:*

After preprocessing technique, next step is algorithm selection. In this research, we used Backpropagation Algorithm. It is the algorithm that falls into the general category of gradient descent, machine learning approach. We can use neural network approach for solving two types of problems, first classification and second regression. The Backpropagation algorithm is a supervised learning approach in which target is already known, so here we are using neural networks to solve regression problems, i.e., prediction of closing price index from previous data sets.

The main objective is to minimize the sum of the difference of square of error by running BPN iteratively. The error function is given by the following equation:

$$E = 0.5 * \sum_{i=1}^{N} (target - output)^2 \quad (3)$$

In this algorithm weights are updated after one complete epoch. The weights are adjusted according to the error computed after each iteration. The BPN algorithm is described below:

1) Step 1: Initialize random weights.

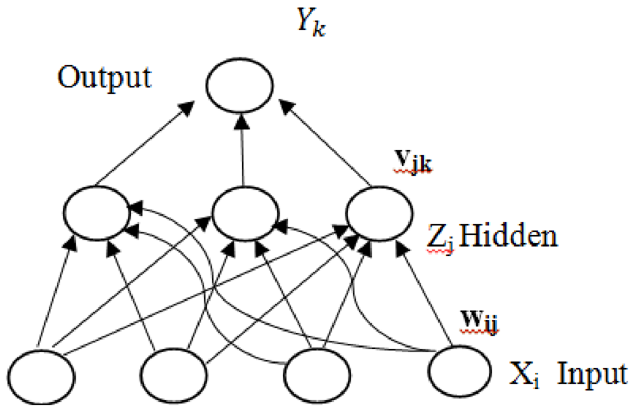

Fig. 3: Neurons model with one hidden layer

Where, w_{ij} are weights between input and hidden layer, v_{jk} are weights between hidden and output layer.

2) Step 2: During feedforward, output at each layer is computed using following equations:
   – output_sum = bias + $\sum_{i=1}^{n}(input \textbf{ x } weight)$
   – output = f(output_sum), i.e., using activation function.
3) Step 3: Backpropagation of error and updating weights.

Following Activation functions can be used:

1) Unipolar Sigmoid:

$$f(x) = \frac{1}{1+\exp(-\lambda x)}$$

2) Bipolar Sigmoid:

$$f(x) = \frac{2}{1+\exp(-\lambda x)} - 1$$

And many more functions can be used as per requirement.

C. *Hidden Nodes:*

Neural networks can have arbitrary number of hidden nodes. In this research we compared three structures, one with 3 hidden nodes, another with 5 hidden nodes and last one with 7 hidden nodes.

D. *Network Description, Data Training and Validation:*

Neural network is connected with simple processing units called neurons. It resembles the biological neurons and works in parallel. The neuron relates the set of input variables, say X_i (i=1, 2, 3….), to output variables Y_j (j=1, 2, 3,…..) which is similar to like any other model whether it is linear or polynomial approximation. The input variables are mapped on output variable through activation function. This function can be linear or non-linear.

Each neuron receives input signal which transmits through connection which have some cost value W, i.e., weights. A bias is also added.

In this paper, we compared three structures with random weights and bias. We trained neural network to reduce square error to 0.04. The network has *i* inputs, *j* hidden layers, so number of weighted connections are *i* x *j*. The output layer has *j* inputs and *k* outputs, so *j* x *k* weighted connections. The relation between input and output neuron is made using activation function described in section II.

E. *Data Training:*

The input data is pre-proceeded and arranged in the form of a matrix of dimension n x m, where n is total input training sets and m is total input attributes.

– Inputs (open, high, low, bias)

- Output (closing value)
- Percentage training set (70%)
- Percentage of validating data set (30%)
- Learning rate (0.1)

1) Error Calculation:

While training neural network with BPN, we simultaneously computed mean square after each epoch.

2) Cross Validation:

The remaining 30% of data is validated by using the updated weights which was obtained after training phase. The graphs are plotted and compared with original values as shown.

IV. SIMULATION RESULT AND OBSERVATION

- Input (Open price, high price, low price, bias)
- Output (Closing price)

From simulation results, it has been observed that neural network with 3 hidden nodes requires more iterations to converge to local minima. Among 3, 5 & 7 hidden nodes neural architectures, we got best result with 5 hidden nodes. The convergence time with 5 hidden nodes is lesser than other two neural structures.

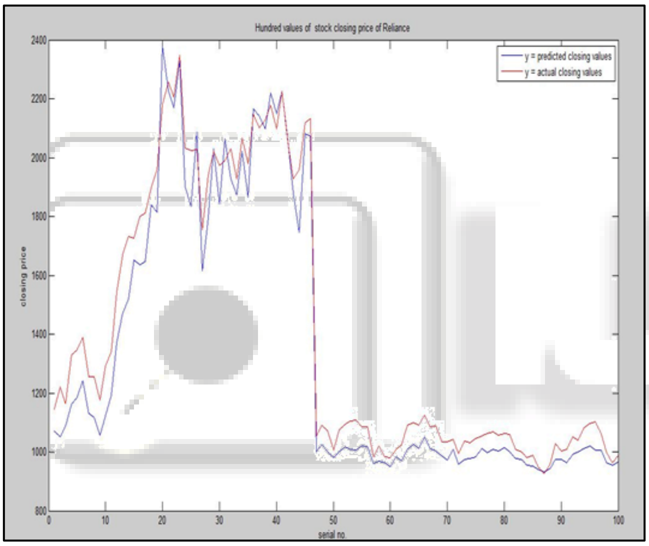

Fig. 4.1: Actual and Predicted closing Price with 3 hidden nodes

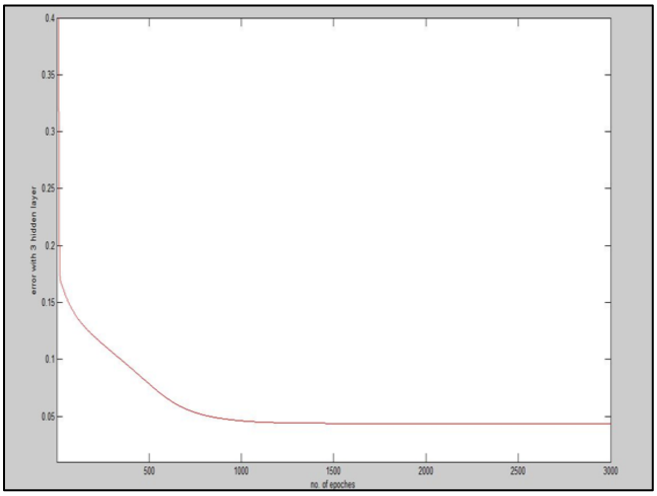

Fig. 4.2: Error computation, with 3 hidden nodes, during training.

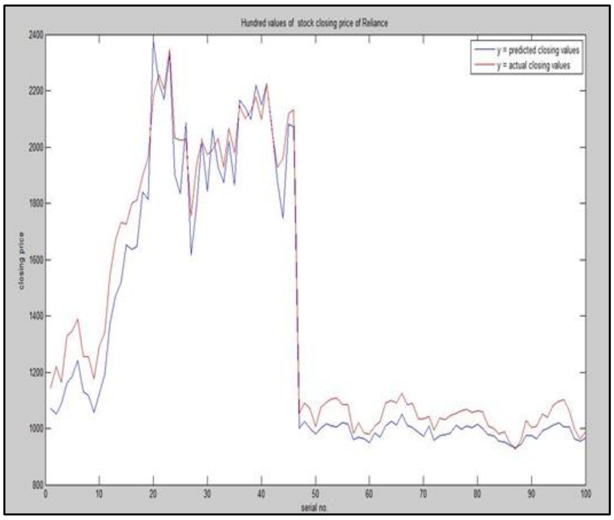

Fig. 5.1: Actual and predicted closing price with 5 hidden nodes

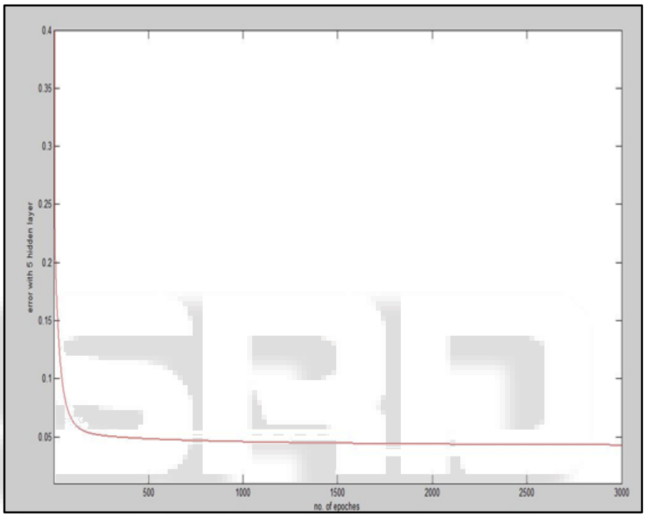

Fig. 5.2: Error computation, with 5 hidden nodes, during training

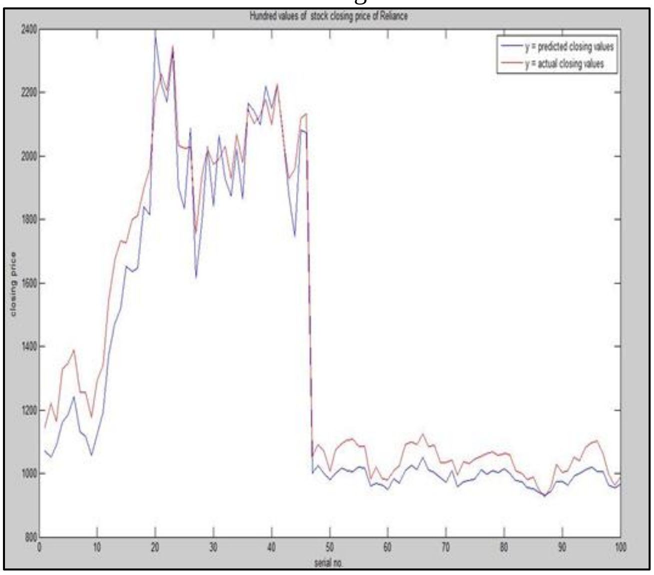

Fig. 6.1: Actual and Predicted closing Price with 7 hidden nodes

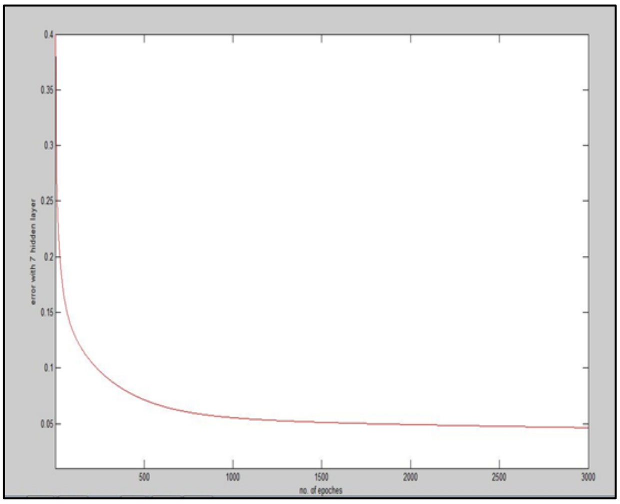

Fig. 6.2: Error computation, with 7 hidden nodes, during training

V. CONCLUSION

In our research, we described applications of neural networks in Stock Market Predictions. The system we developed and tested predicts the closing price of Reliance stock market. In this methodology, we varied hidden nodes within neural network structure. The prediction obtained was fairly accurate as shown in figure 4.1, 5.1 & 6.1. The error versus number of epochs graphs were plotted with different hidden layers as shown in figures 4.2, 5.2 & 6.2. It has been found that the neural network with 5 hidden nodes converges faster as compared to other neural network structures, i.e., with 3 and 7 hidden nodes. Furthermore, it has been observed that accuracy can be improved by modifying pre-processing techniques, changing learning rate, varying hidden nodes. So it has been concluded that neural network is an efficient tool to work upon stock market prediction.

REFERENCES

[1] Yahoo Finance, Last access on: 20 February [2015].Available: http://finance.yahoo.in.
[2] Manna Majumder, MD Anwar Hussain, “Forecasting Of Indian Stock Market Index Using Artificial Neural Network”.
[3] Abhishek Kar, “Stock Prediction using Artificial Neural Networks”, Dept. of Computer Science and Engineering, IIT Kanpur.
[4] Jason E. Kutsurelis, “Forecasting financial markets using neural networks: An analysis of methods and accuracy”, Naval Post Graduate School Monterey, California, September 1998.
[5] P. B. Patel, “Forecasting closing price indices using neural networks”, IEEE Conference on Systems, Man, and Cybernetics, Taipei, Taiwan October 8-11, 2006
[6] Neural Networks: A comprehensive foundation by, Simon Haykins, 2nd edition
[7] Fundamentals of Neural Networks, Laurene Frausett

