A Stock Market Prediction Model using Artificial Neural Network

Kumar Abhishek[1], Anshul Khairwa[2], Tej Pratap[3], Surya Prakash[4]
Department of Computer Science and Engineering
National Institute of Technology Patna, India - 800005
[1]kumar.abhishek@nitp.ac.in, [2]anshul.ahm@gmail.com, [3]tejpratap.nand@gmail.com, [4]suryaprakashsingh7@gmail.com

***Abstract*: The use of Neural networks has found a variegated field of applications in the present world. This has led to the development of various models for financial markets and investment. This paper represents the idea how to predict share market price using Artificial Neural Network with a given input parameters of share market. Artificial Neural Network can remember data of any number of years and it can predict the feature based on the past data. This paper makes use feed forward architecture for prediction. The network was trained using one year data. It shows a good performance for market prediction. The network selected though was not able to predict exact value but it succeeded in prediction the trends of stock market.**

Keywords—*Artificial Neural Network, Feed Forward Network, Stock Prediction*

I. INTRODUCTION

The previous three decades has seen that share market prediction has become a good topic for research. Many researchers have given their idea how to predict share price with more accuracy. There are different methods that have been applied in order to predict Share Market returns. Tang and Fishwick [1]; Wang and Leu [2] provided a general introduction of how a neural network should be developed to model financial and economic time series. During the last decade, Artificial Neural Networks have been used in share market prediction. One of the first such projects was by Kimoto *et al.* [3] who had used ANN for the prediction of Tokyo stock exchange index. Minzuno *et al.* [4] applied ANN again to Tokyo stock exchange to predict buying and selling signals with an overall prediction rate of 63%. Sexton *et al.* [5] theorized that the use of momentum and start of learning at random points may solve the problems that may occur in training process in 1998. Phua *et al.* [6] applied neural network with genetic algorithm to the stock exchange of Singapore and predicted the market direction with an accuracy of 81%.
The share market is dynamic in nature means to predict share price is very complex process by general prediction or computation method. Its main reason is that there is no linear relationship between market parameters and target closing price. Since there is no linear relationship between input patterns and corresponding output patterns, so use of neural network is a choice of interest for share market prediction. Because this network in training phase learns about situations affecting share market price in a given environment. And this learnt knowledge stored in given network is used for predicting future market price.
Here we have used data set of Microsoft Corporation from January 1, 2011 to December 31, 2011 [7] for training and prediction. Considered input parameters for prediction are open, high, low, adj. close and volume. Back propagation algorithm is used for training and for learning we used different functions that are named in later section. Comparison represents the effect on performance while altering number of levels and number of neurons in given layer.

II. ARTIFICIAL NEURAL NETWORK

Artificial Neural Networks (ANNs) are most often chosen for its ability to generalize results from unseen data, especially for dynamic systems on real time basis. ANNs are parallel computational models comprised of densely interconnected adaptive processing units. These networks are fine grained parallel implementations of dynamic systems. ANNs can identify and learn correlated patterns between input data sets and corresponding actual target values. ANNs are networks of highly interconnected neural computing elements that have the ability to respond to input stimuli and to learn to adapt to the environment. ANN includes two working phases, the phase of learning and that of recall. During the learning phase, known data sets are commonly used as a training signal in input and output layers. The recall phase is performed by one pass using the weight obtained in the learning phase.

A) Model used for prediction

The paper here uses Back-Propagation algorithm with multilayer Feed Forward network for prediction.

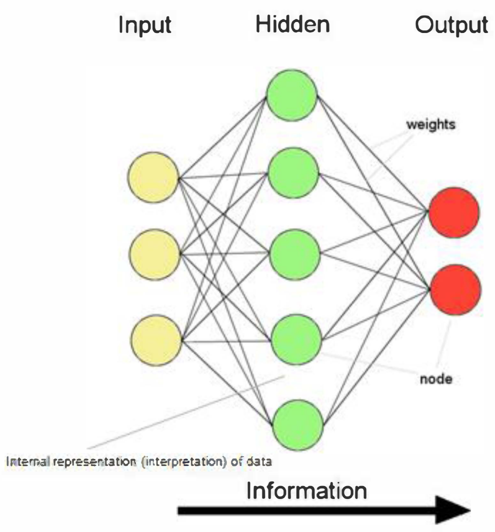
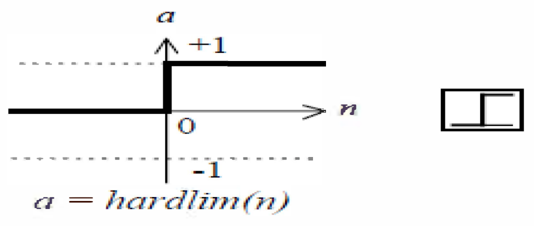
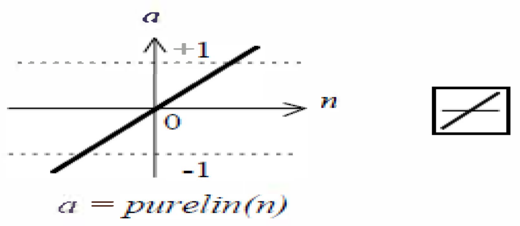


Fig.1—Feed Forward architecture

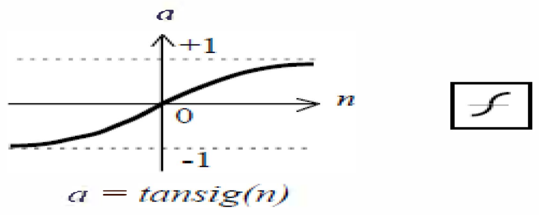
Above figure have three layers of neurons, where one input layer, one output layer and one hidden layer is present. Every neuron employs activation function that fires when total input is more than a given threshold, that can lie in one of the category (symbol corresponding to transfer function is given in right side):



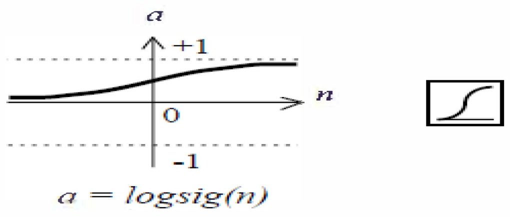
(a)



(b)



(c)



(d)

Fig.2—Transfer function (a) Hardlim (b) Purelin (c) Tansig (d) Logsig

Since there is no linear relationship between input and output patterns in share market prediction so here learning functions hardlim and purelin are rarely used.

III. METHODOLOGY

The work was divided into following units:

1. Training process
2. Experimental Results
3. Comparison and conclusion

A) Training process

It is the process where network is in learning phase during which network stores knowledge. This knowledge is used by the system to predict future output based on input parameters. Here weights and biases related to each neuron are changed iteratively to converge to suitable value.
It can be performed by two ways -

- Incremental mode: Here after applying every training sample weights and biases are updated.
- Batch mode: Here weights and biases are updated after applying a set of training samples. It is more time efficient than incremental mode. We have used this mode for training.

There are many learning algorithms to perform training. Learning is of two types -

- Unsupervised learning: Here target class or values are not known, network makes cluster on the basis of similarity of training items. Here concept of clustering is used to trace target class.
- Supervised learning: Here target class or values are known. On the basis of input parameters using network, output is determined and comparing it by target, error is calculated. Then network property is changed to minimize error by updating weights and biases. This concept is used here.

Genetic algorithm used here for learning is Back Propagation. Standard Back Propagation is gradient descent algorithm, it is as following:
Let
P = input matrix (1-D) (D=dimension)
O = output matrix (1-D)
T = target matrix (1-D)
B = bias matrix (1-D)
W = weight matrix (2-D)
e(k) = T(k)-O(k) (here k=1, 2, 3… corresponds to kth row in given matrix)
Then we calculate error matrix (1-D) as

1. $E(k) = 1/2(T(k)-O(k))^2 = 1/2e(k)^2$
2. $dE(k)/dW_{1,j} = e(k)*de(k)/dW_{1,j}$
   (for j=1, 2, 3….)
   here $W_{i,j}$ = weight of link from node j to i.
3. $dE(k)/dB = e(k)*de(k)/dB$

if a be the learning factor then weights and biases are updated as following:

$W(k_{new}) <\!- W(k)+a*e(k)*P^T(k)$
And $B(k_{new}) <\!- B(k)+a*e(k)$

There are many variations of this algorithm. Newton's method is one of them.
In our experiment we have used Levenberg-Marquardt algorithm that is extension of quasi-Newton method based on newton's method. In our given Matlab tool there is a training function 'trainlm' that employees this method in training process. We have used this function in our experiment. Its brief overview is as following:
It was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training Feed-Forward networks), then the Hessian matrix can be approximated as
$\mathbf{H} = \mathbf{J}^T\mathbf{J}$
and the gradient can be computed as
$\mathbf{g} = \mathbf{J}^T\mathbf{e}$
where **J** is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and **e** is a vector of network errors. The Jacobian matrix can be computed through a standard Back-Propagation technique that is much less complex than computing the Hessian matrix.
The Levenberg-Marquardt algorithm [8] uses this approximation to the Hessian matrix in the following Newton-like update:
$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T\mathbf{J} + \mu\mathbf{I}]^{-1}\mathbf{J}^T\mathbf{e}$

Parameters related to trainlm are:

1. mu: it is a scalar if it is 0, it is just the newton method and if large then it is gradient descent with small step size.
2. Mu_inc: if performance function is increased in any step then it is multiplied by mu for new value of mu.
3. Mu_dec: if performance function is decreased then it is multiplied by mu to get new value of mu.
4. Mu_max: if value of mu becomes larger than mu_max in any step then training stops.
5. Mu_reduc: it is used to control the amount of memory used by algorithm.
6. Epochs: it specifies in maximum how many iterations convergence is found.
7. Show: if it is true training status is displayed at each epoch. If it is set to nan it means false.
8. Goal: it is scale of minimum error after getting it further iteration is not required.
9. Time: specifies in what maximum time goal is reached.

B) *Experimental results*

The dataset of Microsoft Corporation has been used for experiment [7]. There are 6 parameters in dataset open, high, low, volume, adj. close and close. We have taken 5 parameters open, high, low, volume and adj. close as input parameters and close as output parameters for predicting stock price.
The training functions are trainlm and traingdx in our experiment. Results are shown below in tables.

TABLE 1:
OBSERVATION FOR TRAINLM

No. of layers	Neurons in input and hidden layers	Neurons in output layer	MSE (Mean Squared Error)
2	10	1	0.00650
2	15	1	0.0068
2	25	1	0.00516
3	10	1	0.00347
3	15	1	0.00495
3	20	1	0.00356

4	10	1	0.00599
4	25	1	0.00356
5	5	1	0.0084
5	10	1	0.0058
5	30	1	0.00438

TABLE 2:
OBSERVATION FOR TRAINGDX

No. of layers	Neurons in input and hidden layers	Neurons in output layer	MSE (Mean Squared Error)
2	5	1	0.0699
2	10	1	0.0430
3	1	1	0.0969
3	15	1	0.0954
4	1	1	0.144
4	15	1	0.240
4	20	1	0.215
5	1	1	0.0313
5	15	1	0.00399
5	25	1	0.00364

This observation is taken when other parameters are: epochs = 1000, time = inf, goal = 0, min_grad = 1e-005, max_fail = 6, lr = 0.01, lr_inc = 1.05, lr_dec = 0.7, max_perf_inc = 1.04, mc = 0.9
From above experiment we conclude that trainlm has better performance than traingdx for share market prediction.

Performance and Regression graph for trainlm:

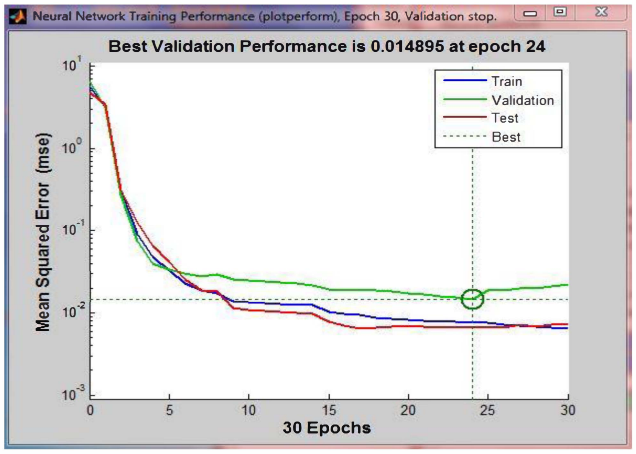


Fig.3—Performance curve for trainlm

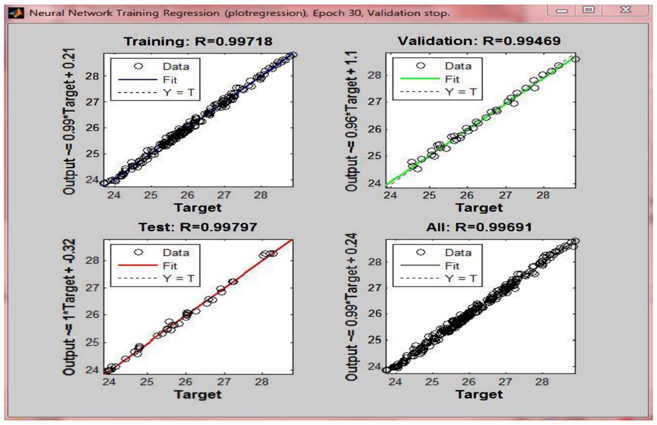


Fig.4—Regression plot for trainlm

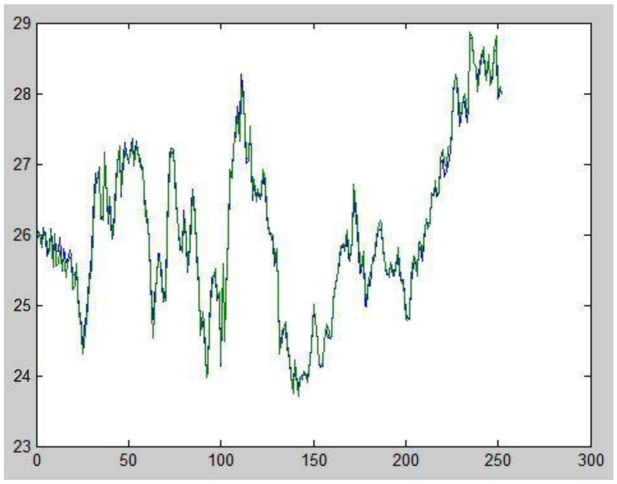


Fig.5—Plot between output and target value for trainlm

Performance and Regression graph for traingdx:

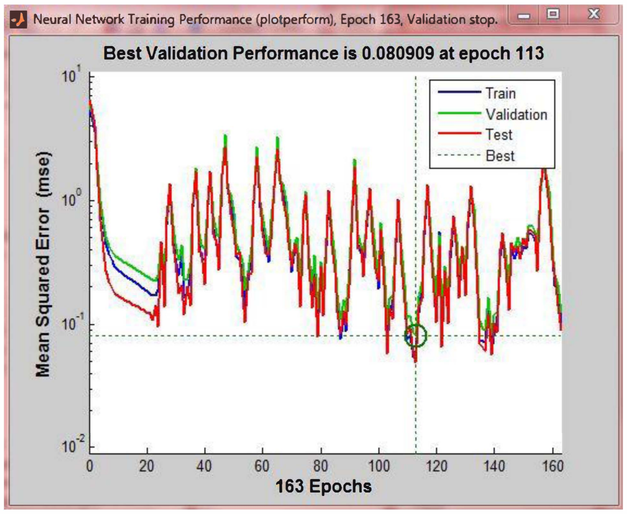


Fig.6— Performance curve for traingdx

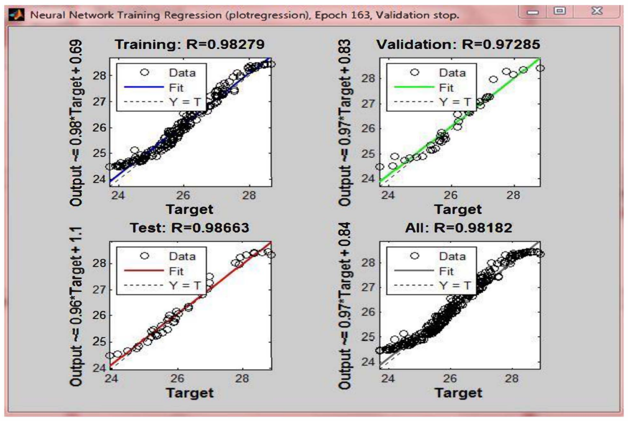


Fig.7— Regression plot for traingdx

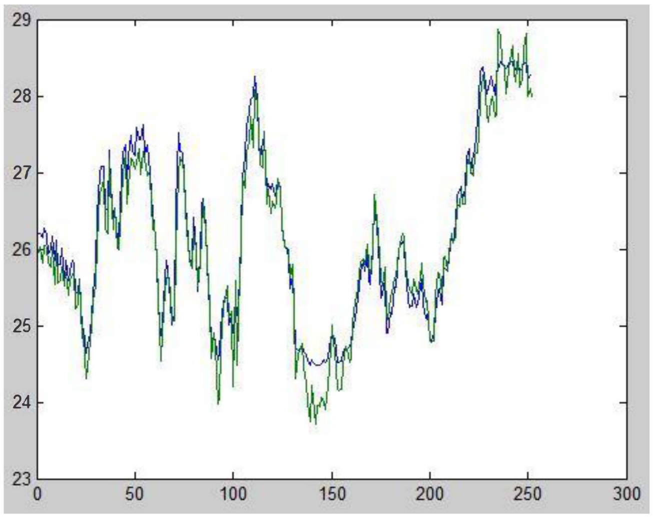


Fig.8— Plot between output and target value for traingdx

IV. CONCLUSION

The experiment conducted in this paper uses simple and efficient approach to stock prediction using Back-Propagation with Feed Forward Network.

The Accuracy of the network recorded was 99% in case of trainlm (with two layer, 10 neurons in input layer and 1 in output layer) and MSE = 0.00650.

The Accuracy of the network recorded was 98% in case of traingdx (with two layers, 10 neurons in input layer and 1 in output layer) MSE = 0.0430.

The Future scope of the work will concentrate towards trying different ANN architecture and identifying which suits well for prediction along with different training and learning functions.

ACKNOWLEDGMENT

This paper is supported by Ankur Khairwa. This research would not have been possible without his technical assistance and his guidance about various finance related terms. We express our gratitude to Ankur Khairwa.

REFERENCES

[1] Z.Tang and P.A.Fishwick, "Backpropagation neural nets as models for time series forcasting," ORSA journal on computing, vol.5, No. 4, pp. 374-384, 1993.

[2] J.H.wang and J.Y.Leu, "stock market trend prediction using ARIMA-based neural network,"Proc. Of IEEE conference on neural networks, vol.4, pp2160-2165, 1996

[3] Kimoto,T., asakawa, K., Yoda , M,. andTakeoka, M. ,Stock market prediction system with modular neural network, in proceedings of the International Joint Conference on Neural Network, 1-6(1990).

[4] Mizuno, H., Kosaka , M., Yajima , H. and Komoda N. , Application of Neural Network to Technical Analysis of Stock Market Prediction, Studies in Information and Control , vol.7, 1998, no.3, pp.111-120.

[5] Sexton, R. S., R. E. Dorsey and J.D.Dohnson, Toward global optimization of neural networks: A comparison of the genetic algorithm and backpropagation, Decision Support systems 22(1998), 171- 185.

[6] Phua, P.K.H. Ming, D., Lin, W., Neural network With Genetic Algorithms for Stocks Prediction, Fifth Conference of the Association of Asian-Pacific Operations Research Societies, 5th – 7th July (2000), Singapore.

[7]http://ichart.finance.yahoo.com/table.csv?s=MSFT&a=00&b=1&c=2011&d=11&e=31&f=2011&g=d&ignore=.csv Sunday

[8] Neural Network Toolbox for Use with MATLAB® *Howard DemuthMark Beale*