International Journal of Scientific & Engineering Research, Volume 5, Issue 3, March-2014 172
ISSN 2229-5518

Prediction of Closing Price of Stock Using Artificial Neural Network

H.B.Kekre, Hitesh Makhija, Pallavi N.Halarnkar

Abstract— This paper analyses the theories used to explain the stock market movements. It uses the Chaos Theory, which essentially states that the stock market is a chaotic system. It then uses Artificial Neural Networks to learn this chaotic system. The Learning algorithm used is the Error Back Propagation Learning Algorithm. The Artificial Neural Network use is the Feed-forward Artificial Neural Network. The Neural Network predicts the next day closing price of a stock.

Index Terms— Chaos, Artifical Neural Network, Feed Forward, Error Back Propogation, Stock Predcition

1 INTRODUCTION

Forecasting has long been in the domain of linear statistics. Linear models can be easily understood and analysed in great detail and they are easy to implement. However, this is not the case when the underlying system is nonlinear as is the case with most of the natural real world systems and stock market is one of them

A computational approach is proposed in [1] for predicting S&P CNX Nifty 50 Index. The proposed approach uses Neural Network model to predict the direction of the movement of the closing value of the index. The method proposed can also be used for predicting the price index value of the stock market. Based on various studies of the network model , an optimal model is proposed for forecasting. The model has been validated across 4 years of the trading days. The highest performance predicted is 89.65% with an average accuracy of 69.72% over a period of 4 years

Even though Artificial Neural networks find a numerous applications and their usage in predicting the stock price, opinions about their contribution are mixed. In [2] eleven guidelines are laid out that can be used in evaluating this literature about neural networks ad their usage for forecasting. 48 studies were located between 1988 to 1994, all of these were tested how they were compared with other alternatives and how well was the technique implemented. Out of these 11 were found to be effectively validated and implemented, another 11 were effectively validated and implemented and produced positive results.

Neural Networks and their methods are very widely used for predicting stock market predictions. NNs have proved to be more effective in solving business problems as compared to other statistical methods and other methods that do not include AI. In [3] a comparative analysis of selected applications is conducted. Based on the study , it is concluded that NN's are mostly implemented for forecasting stock prices, returns and stock modeling and the most frequent technology is back propagation algorithm.

In [4] , a method based on interrelated time series data has been proposed for predicting stock market. Although there are many methods proposed for stock market price but only a few of them consider other time series data for the same. In the proposed method the interrelationship between the predicted stock and various time series data such as other stocks , world stock market indices , foreign exchanges and oil prices are derived. These interrelationships are used for predicting the daily up and down changes in the closing value. The experimental results proved to be good specially in the manufacturing industry.

In [5], a prediction of the market share price using Neural Networks with the given input parameters of the share market is done. Artificial Neural Network can remember data of any number of years , which can be used for training the network and thus predicting the future based on the past data. The proposed method makes use of feed forward architecture for prediction. The network was trained for one year data.

In [6] a method is proposed for predicting the Istanbul Stock Exchange (ISE) market index value using the Artificial Neural Network. The inputs to the system includes previous days index value, previous days TL/US exchange rate, previous day's overnight interest rate and 5 dummy variables each representing the working day of the week. The Network Architecture includes Multi Layer Perceptron and Generalised Feed forward Networks. Training and Testing is performed with these two Network Architectures. Results are compared to moving averages where ANN prove to be better in performance.

In [7] a method is proposed based on, Maximum a Posteriori Hidden Markov Models (HMM's) for forecasting and predicting the stock market values for next day based on the historical data. The proposed approach makes use of fractional change in stock value and the intra-day high and low values of the stock to train the continuous HMM. This HMM is then used to make a Maximum a Posteriori decision over all the possible stock values for the next day.

Selecting stocks for a suitable portfolio is a difficult task. The main aim lies behind earning maximum returns on investment. An improved method of stock picking using self organizing maps is been proposed in [8]. The best of the portfolio constructed by self-organizing maps outperformed the BSE-30 Index by about 16.35% based on one and half month of stock data.

Prediction is a difficult task , specially when the relationship between input and output is non linear, stock price prediction is one such an item. In [9] a method for stock price prediction is proposed, it makes use of ANN and back propagation algorithm. Historical stock prices are used for training the network

ISSN 2229-5518

2 Stock Market

2.1 Definition of a Stock

Stock is a share in the ownership of a company. Stock represents a claim on the company's assets and earnings. Holding a company's stock means that you are one of the many owners (shareholders) of a company, and, as such, you have a claim (albeit usually very small) to everything the company owns. The value of a company is its market capitalization, which is the stock price multiplied by the number of shares outstanding

2.2 Purpose of a Stock Market

The purpose of a stock market is to facilitate the exchange of securities between buyers and sellers, thus reducing the risks of investing. Stock prices change everyday by market forces. This implies that share prices change because of supply and demand. Understanding demand and supply is easy what is difficult is to quantify the impact of the positive as well as negative news on the stock price.

2.3 Concept of a stock price

Financial theorists define stock price as the present value of all future earnings expectations for the company, divided by its number of shares outstanding. In essence the earning capacity of the company is what defines price. Even companies that lose money today can have a high share price because price is based on the future earnings of the company.

3 Speculation : The self-fulfilling prophecy

An increase in demand for a stock, determined by speculation of future value, will cause by itself, the rise in price that is anticipated. A stock price movement therefore by definition is a self-fulfilling prophecy. Consider an investor who buys into a company because he thinks its stock will increase in value. Others, just like him, buy into the stock for the same reasons. The result is an increase in demand, hence an increase in price, the very thing they were anticipating. Note that the speculative investor, and not the company's fundamentals, drove up the value of the stock.

Hence in essence we can assert that

- At the most fundamental level, supply and demand in the market determine stock price.
- Price times the number of shares outstanding (market capitalization) is the value of a company. Comparing just the share price of two companies is meaningless.
- Theoretically earnings are what affect investors' valuation of a company, but there are other indicators that investors use to predict stock price. Remember, it is investors' sentiments, attitudes, and expectations that ultimately affect stock prices.
- There are many theories that try to explain the way stock prices move the way they do. Unfortunately, there is no one theory that can explain everything.

4 Chaos Theory

A relatively new approach to modeling nonlinear dynamic systems like the stock market is chaos theory. [10] Chaos theory analyzes a process under the assumption that part of the process is deterministic and part of the process is random. Chaos is a nonlinear process, which appears to be random. Chaos theory is an attempt to show that order does exist in apparent randomness. In this paper we assume that the stock market exhibits chaos

In essence, a chaotic system is a combination of a deterministic and a random process. The deterministic process can be characterized using regression fitting, while the random process can be characterized by statistical parameters of a distribution function. Thus, using only deterministic or statistical techniques will not fully capture the nature of a chaotic system. A neural networks ability to capture both deterministic and random features makes it ideal for modeling chaotic systems.

5 Using Artificial Neural Networks

Neural networks are used to predict stock market prices because they are able to learn nonlinear mappings between inputs and outputs. Due to the neural networks' ability to learn nonlinear, chaotic systems we propose the use of Artificial Neural Networks to predict the stock market. The most common neural network architecture is the feed foreword back propagation neural network. The term "feed-forward" describes how this neural network processes patterns and recalls patterns. When using a feed-forward neural network neurons are only connected forward. Each layer of the neural network contains connections to the next layer, but there are no connections back. The term "back-propagation" describes how this type of neural network is trained. Back propagation is a form of supervised training. Using the anticipated outputs and the output of the neural network the errors are calculated and the weights of the various layers are adjusted backwards from the output layer all the way back to the input layer. The architecture of this type of Artificial Neural Network will be

————————————————

- *Dr. H.B Kekre is currently a senior Professor at NMIMS University , Mumbai, Inida E-mail: hbkekre@yahoo.com*
- *Hitesh Makhija is an ex student of TSEC, Mumbai ,India*
- *Pallavi Halarnkara is an Assistant Professor at NMIMS university, Mumbai, India E-mail: Pallavi.halarnkar@gmail.com*

ISSN 2229-5518

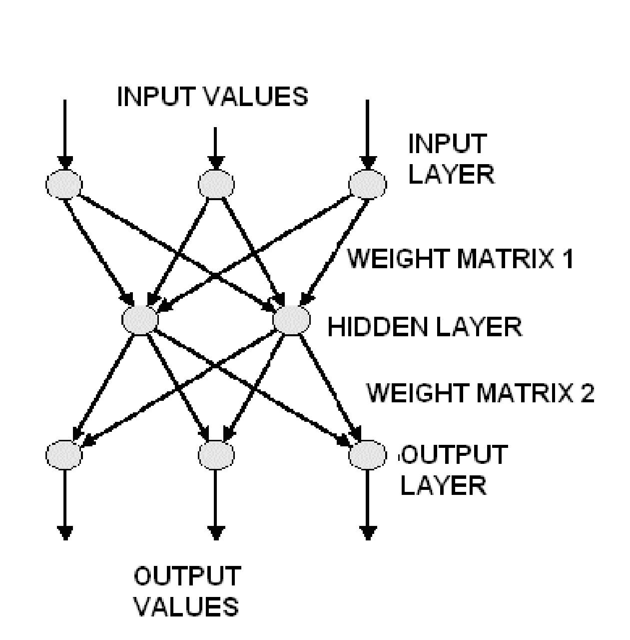


Fig. 1. Artificial Neural Network

6 Inputs

6.1 Value at Risk

While performing risk analysis in the financial markets the most important measure is that of volatility. Formally VAR can be defined as "VaR answers the question: how much can I lose with x% probability over a pre-set horizon" ***source: J. P Morgan, Risk Metrics – Technical Document.*** The method to calculate V.A.R that we propose in this paper is the Variance covariance method. This method assumes that stock returns are normally distributed. Hence we need two statistical measures to calculate the V.A.R, the mean and the standard deviation. Hence this measure takes into account the impact of volatility in the prediction of a stock price. However to maintain the simplicity we will use {(High of the day) – (low of the day)}/(closing price of the day) as the measure of the volatility of the stock

6.2 Volume

This is the measure of the liquidity of the stock. This input will basically be a ratio of the total number of shares of a particular stock traded during a day to the total number of shares traded that day. The higher the liquidity of the stock the more the impact of volatility on the future movement of the stock price.

6.3 Closing price of the stock

The price of a stock is the current estimate of the future earnings of a stock. Hence the stock price implicitly accounts for all the technical and fundamental parameters that are used to analyze the future trend of the stock price. Therefore it is not necessary to feed the neural network with fundamental indicators like the price to earning ratio or other macro economic indicators like the interest rates.

6.4 Closing price of the market

Using this parameter and the closing price of the stock the aim is to make the Artificial Neural Network learn the correlation between the movements of the Sensex (or the Nifty) and the price of the individual stock.

7 Error Back propagation Algorithm

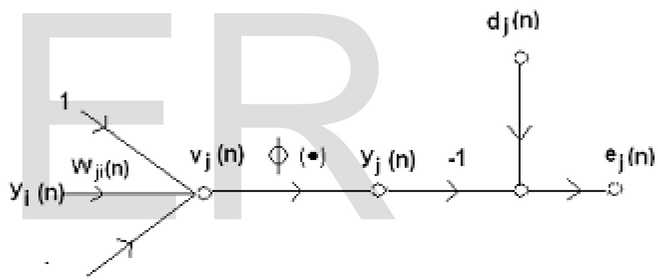
This is a supervised learning algorithm. Back propagation is a form of supervised training. Using the anticipated outputs and the output of the neural network the errors are calculated and the weights of the various layers are adjusted backwards from the output layer all the way back to the input layer. [2]

We will analyze the back propogation algorithm by considering two cases

1. Neuron j is a Output layer Neuron

2. Neuron j is a hidden layer neuron.

Case 1. Consider Output Neuron j
Signal Flow graph



Now the error of the output neuron j is given by

$$e_j(n) = d_j(n) - y_j(n)$$

$d_j(n)$ — Expected output of neuron j

$y_j(n)$ — Actual output of neuron j

The mean square Error function E(n) is given by

$$E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n)$$

C is the set of output neurons.

The local induced field of neuron j is given by

ISSN 2229-5518

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n)\, y_j(n)$$

m – The number of neurons in the hidden layer.

$w_{ji}(n)$ - Weight Matrix of the weights from the hidden neuron i to the output neuron j

Hence the output of neuron j will be the output of the activation function of neuron j when an input to the activation function is the local induced field of neuron j.

$$y_j(n) = \phi_j(v_j(n))$$

ϕ_j - activation function of neuron j

Hence our basic aim is to minimize the error function E(n) by modifying the weight matrix W. Hence we have to consider the partial derivative of the error function with respect to the weight matrix. By the chain rule of partial derivatives we have.

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \times \frac{\partial e_j(n)}{\partial y_j(n)} \times \frac{\partial y_j(n)}{\partial v_j(n)} \times \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

Now,

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n)$$

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1$$

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \phi_j'(v_j(n))$$

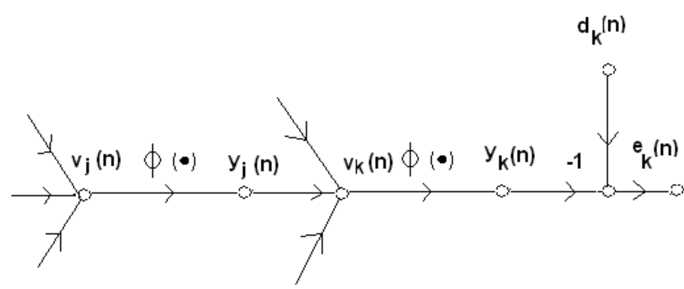
$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_j(n)$$

Hence the change in the weights will be in the direction opposite to the direction of the partial derivative of the error function with respect to the weight matrix and will be scaled by a factor known as learning rate r. Hence we have

$$\triangle w_{ji}(n) = -r \frac{\partial E(n)}{\partial w_{ji}(n)}$$

$$\triangle w_{ji}(n) = r\, e_j(n)\, \phi_j'(v_j(n))\, y_j(n)$$

Case 2: Consider hidden neuron j



Now the above diagram is the signal flow graph of the output neuron k connected to the hidden neuron j.

Hence the error function will be given by

$$E(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n)$$

Where,

$$e_k(n) = d_k(n) - y_k(n)$$

ISSN 2229-5518

$$y_k(n) = \phi_k(v_k(n))$$

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -e_j(n)\ \phi_j'(v_j(n))\ y_j(n)$$

$\phi_k(v_k(n))$ **Is the local induced feild of output neuron k**

Hence we have

$$e_k(n) = d_k(n) - \phi_k(v_k(n))$$

And,

$$v_k(n) = \sum_{j=0}^{m} w_{kj}(n)\ y_j(n)$$

Again we have to minimize the error function E(n) by adjusting the synaptic weights.

Hence again we have to consider the partial derivative of the error function with respect to the weight matrix. By the chain rule of partial derivatives we have

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial y_j(n)} \times \frac{\partial y_j(n)}{\partial v_j(n)} \times \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

Now,

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_{k \in C} e_k(n) \frac{\partial e_k(n)}{\partial y_j(n)}$$

Now by chain rule

$$\frac{\partial e_k(n)}{\partial y_j(n)} = \frac{\partial e_k(n)}{\partial v_k(n)}\ \frac{\partial v_k(n)}{\partial y_j(n)}$$

Now,

$$\frac{\partial e_k(n)}{\partial v_k(n)} = -\phi_k'(v_k(n))$$

And,

$$\frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n)$$

Hence

$$\frac{\partial E(n)}{\partial y_j(n)} = -\sum_{k \in C} e_k(n)\ \phi_k'(v_k(n))\ w_{kj}(n)$$

Now,

$$y_j(n) = \phi_j(v_j(n))$$

Hence

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \phi_j'(v_j(n))$$

Also,

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n)$$

Hence,

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -y_i(n)\phi_j'(v_j(n)) \sum_k e_k(n).\phi_k'(v_k(n))\ w_{kj}(n)$$

Hence the change in the weights will be in the direction opposite to the direction of the partial derivative of the error function with respect to the weight matrix and will be scaled by a factor known as learning rate r. Hence we have

$$\triangle w_{ji}(n) = r \ \mathbf{x}$$

$$y_i(n)\phi_j'(v_j(n)) \sum_k e_k(n).\phi_k'(v_k(n))\ w_{kj}(n)$$

Momentum

To have the flexibility to increase the learning rate and at the same time ensure that the neural network does not become

ISSN 2229-5518

unstable, i.e oscillatory we use the concept of momentum. We use m as a positive number. This is called the momentum constant.
Hence now the rule for weight change becomes

$$\triangle_m W(n) = m\, W(n-1) + \triangle W(n)$$

where,

$\triangle_m W(n)$ – **Change of weights when momentum is used**

Activation Function

The activation function used is hyperbolic tangent. The activation function is given by

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

Batch updation of weights.

The weight updation takes place after the presentation of all the training samples that constitute the epoch.

8. Experimental Results

Configuration 1

The network parameters for this configuration were as follows:

Number of Hidden Neurons = 15
Learning Rate = 0.1
Momentum = 0.4
Window Size = 10
Training Sets = 30
The results of this configuration are as follows.

Actual Output	Predicted Output
2843.00	2823.10
2913.50	2891.95
2935.20	2969.55
2935.45	2972.29
2921.80	2992.21

In this configuration the learning rate was kept at 0.1 which increased the learning time significantly but yielded good accuracy in terms of predicted output. Hence in the next configuration under consideration we increased the learning rate.

Configuration 2

The network parameters for this configuration were as follows:

Number of Hidden Neurons = 15
Learning Rate = 0.4
Momentum = 0.7
Window Size = 10
Training Sets = 35

The results of this configuration were as follows:

Actual Output	Predicted Output
2877.95	2871.26
2966.90	2952.56
2959.75	2942.13
2920.00	2911.10
2890.45	2876.61

In this configuration the training time reduced due to an increase in the learning rate was offset by the fact that the number training sets were increased. It also caused sharp spikes in the learning curves and stabilized at a global minimum larger than the minimum mean square error specified. Due to a large momentum we observed high frequency of oscillations around the global minimum before the network finally stabilized.

Configuration 3

The network parameters for this configuration were as follows:

Number of Hidden Neurons = 20
Learning Rate = 0.4
Momentum = 0.8
Window Size = 15
Training Sets = 40

The results of this configuration were as follows:

ISSN 2229-5518

Actual Output	Predicted Output
2896.35	2885.25
2897.30	2891.35
2960.00	2971.55
3036.55	3012.29

In this configuration the increase in training time due to the increase in the number of neurons in the hidden layer was offset by a high learning rate and high momentum combined with an increase in the window size.

Due to a large momentum we obtained a highly linear learning curve, which was interrupted by sharp spikes due to a high learning rate. The results obtained were therefore not satisfactory. Hence in the next configuration we reduced the learning rate, momentum and the number of neurons in the hidden layer.

Configuration 4

The network parameters for this configuration were as follows:

Number of Hidden Neurons = 15
Learning Rate = 0.2
Momentum = 0.7
Window Size = 12
Training Sets = 40

The results of this configuration were as follows:

Actual Output	Predicted Output
2896.35	2891.11
2897.30	2892.39
2960.00	2965.52
3036.55	3022.69

Due to a relatively small learning rate and moderate momentum combined with 15 neurons in the hidden layer the network yielded excellent generalization.

However due to low learning rate there was a problem of local minima which was alleviated to some extent by the relatively high momentum which in turn caused high oscillations. However, when the network stabilized, the generalization was excellent.

Configuration 5.

The network parameters for this configuration were as follows:

Number of Hidden Neurons = 10
Learning Rate = 0.2
Momentum = 0.5
Window Size = 12
Training Sets = 30

The results of this configuration were as follows:

Actual Output	Predicted Output
2843.00	2810.61
2913.50	2888.31
2935.20	2896.10
2935.45	2891.45
2921.80	2888.11

In this configuration we tried to reduce the oscillations by reducing the momentum. But this resulted in the network getting stuck in the local minima, thereby yielding inaccurate results.

Configuration 6

The network parameters for this configuration were as follows:

Number of Hidden Neurons = 15
Learning Rate = 0.2
Momentum = 0.8
Window Size = 15
Training Sets = 40

The results of this configuration were as follows:

Actual Output	Predicted Output
2896.35	2889.65
2897.30	2891.23
2960.00	2989.42
3036.55	3001.43

Due to the combination of a low learning rate combined with a relatively high momentum yielded a smooth curve. This configuration did not yield excellent generalization because of the relatively large window size, where the network tends to memorize the patterns.

ISSN 2229-5518

In addition to the network parameters taken into consideration, the performance of the network is also influenced by the nature of the initialization of weights which was done randomly.

9. Conclusion

After a comparative analysis of the performance of the network in the six configurations, we concluded that configuration 4 is optimum for the designed neural network, since it is fair trade-off between learning time, accuracy and generalization.

References

[1] M. Majumdar, MD. A. Hussian. "Forecasting of Indian Stock Market Index Using Artificial Neural Network". www .nseindia.com/content/research/FinalPaper206.pdf.

[2] Monica Adya and Fred Collopy, "How Effective are Neural Networks at Forecasting and Prediction? A Review and Evaluation", Journal of Forecasting J. Forecast. 17, 1998. pp. 481-495

[3] Zekic, Marijana. "Neural network applications in stock market predictions-a methodology analysis." In proceedings of the 9th International Conference on Information and Intelligent Systems, vol. 98, pp. 255-263. 1998.

[4] Ryota, K., and N. Tomoharu. "Stock market prediction based on interrelated time series data." In Computers & Informatics (ISCI), 2012 IEEE Symposium on, pp. 17-21. IEEE, 2012.

[5] Abhishek, Kumar, Anshul Khairwa, Tej Pratap, and Surya Prakash. "A stock market prediction model using Artificial Neural Network." In Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on, pp. 1-5. IEEE, 2012.

[6] Birgul Egeli, Asst. "Stock Market Prediction Using Artificial Neural Networks."Decision Support Systems 22: 171-185.

[7] Gupta, Aditya, and Bhuwan Dhingra. "Stock market prediction using hidden markov models." In Engineering and Systems (SCES), 2012 Students Conference on, pp. 1-4. IEEE, 2012.

[8] Khan, A., T. K. Bandopadhyaya, and Sudhir Sharma. "Classification of stocks using self organizing map." International Journal of Soft Computing Applications4 (2009): 19-24.

[9] Ramani, Prakash, and P. D. Murarka. "Stock Market Prediction Using Artificial Neural Network." International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue no. 4 (2013) pp.873-877.

[10] Ramon Lawrence "Using Neural Networks to Forecast Stock Market Prices",Course Project, University of Manitoba Dec. 12, 1997.