Indian Journal of Science and Technology, *Vol 8(35), DOI: 10.17485/ijst/2015/v8i35/71306, December 2015*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Stock Market Prediction using Neuro-Genetic Model

K. V. Prema*, N. Manish Agarwal, Murali Krishna and Visakha Agarwal

Department of Computer Science and Engineering, MIT, Manipal University, Manipal - 576104, Karnataka, India; drprema.mits@gmail.com, nmanishagarwal@gmail.com, murali.sn@gmail.com, visakhaagarwal1993@gmail.com

Abstract

Background/Objectives: To design a stock market prediction system using neuro-genetic approach. To predict BSE Sensex closing price using an artificial neural network. To optimize the synaptic weight values using genetic algorithm. **Methods/Statistical Analysis:** In this research work input parameters related to the BSE Sensex are fed as input dataset to the Multi Layer Perceptron neural network and the future day Sensex closing value is predicted as the output. Various training algorithms are implemented and the results are compared. The best neural network model is further is subjected to synaptic weight optimization using Genetic Algorithm. The various models are then subjected to testing over a period of 15 days, to obtain the most accurate model. **Findings:** The proposed system applies variants of Back Propagation (BP) learning algorithm on a Multi Layer Perceptron network (MLP) which is trained using four years' BSE Sensex data. The performance of the network is measured by Normalized Mean Squared Error (NMSE). It is observed that resilient back propagation algorithm with log sigmoid activation function gives the lowest NMSE of 0.003745. The research work also uses Genetic Algorithm (GA) for weight optimization. BP suffers from the danger of getting stuck in local minima. This is avoided by using GA to select the best synaptic weights and node thresholds initially and then proceeding with the training of MLP using BP. It is observed that this hybrid model gives optimized results. In order to substantiate the model proposed, experiments are first conducted without using GA. The results of this general BP MLP model are then compared with that of GA-BP MLP model and analyzed. NMSE for the GA-BP MLP model is 0.003092121. Artificial Neural Network has evolved out to be a better technique in capturing the structural relationship between a stock's performance and its determinant factors more accurately than many other statistical techniques. Although neural network models are used for stock prediction very little work is done on BSE SENSEX data. The proposed work is unique as experiments are done using different variants of back propagation learning algorithm and different activation functions. Therefore these experimental results add value to the existing work. The proposed work also demonstrates that performance improvement can be achieved by using genetic algorithm. **Application/Improvements:** The proposed model can also be used for forecasting index returns of markets like New York Stock Exchange, Hang Seng Stock Exchange, Korea Stock Exchange, Taiwan Stock Exchange etc., by using appropriate data set.

Keywords: Artificial Neural Network, Closing Value, Genetic Algorithm, Optimization, Sensex Prediction, Synaptic Weight

1. Introduction

Stock market prediction is an important task for stock traders, applied researchers and stock investors. Various methods have been devised for the same. They are fundamental analysis, technical analysis, traditional time series forecasting, and Artificial Neural Networks. Technical analysis uses historical data such as stock prices and volume information for prediction. It is highly subjective in nature and has been shown to be statistically invalid. Nevertheless it is preferred by many stock traders. Traditional time series techniques require a large amount of high-quality data and are suited for short-term forecasting only[1].

There are some distinguishing features that make Artificial Neural Network (ANN) a preferred technique for prediction as compared to other models of forecasting. ANN gaining increasing acceptance in the business area due to its ability to learn and detect relationships among nonlinear variables. Also, it allows deeper analysis of large sets of data, especially those that have the tendency to fluctuate within a short of period of time[2].

*Author for correspondence

A neural network is a massively parallel distributed processor made up of simple processing units which has a natural propensity for storing experiential knowledge and making it available for use[3]. ANNs are capable of handling non-linear systems effectively. Stock market is a chaos system[4]. Chaos is a non-linear deterministic system which only appears random because of its irregular fluctuations. These systems are dynamic, a periodic, complicated and difficult to deal with normal analytical methods. The neural networks are effective in learning such non-linear chaotic systems because they make very few assumptions about the functional form of the underlying dynamic dependencies and their initial conditions. But most of the non-linear statistical techniques require that the non-linear model be specified before the estimation of the parameters and generally it happens that the pre-specified non-linear models may fail to observe the critical features of the complex system under study. ANN has evolved out to be a better technique in capturing the structural relationship between a stock's performance and its determinant factors more accurately than many other statistical techniques[5–8].

Genetic algorithms, based on Darwinian survival of the fittest, are self adaptive globally optimistic search procedures[9–11]. Genetic Algorithm (GA) is an iterative algorithm that is parallel and global. It is also a soft computing method. GAs are good at taking larger, potentially huge search spaces and navigating them looking for optimal combinations of things and solutions which we might not find in a life time[11].

The MLP is trained by providing the stock market parameters as inputs. The network is trained using four years' BSE Sensex data. The research work also uses Genetic algorithm for weight optimization. BP suffers from the danger of getting stuck in local minima. In order to substantiate the model proposed, experiments are first conducted without using GA. The results of this general BP MLP model are then compared with that of GA-BP MLP model and analyzed. It is observed that with the help of available data upto nth day, the prediction for the (n+1)th day can be made.

2. Background

Prediction of stock prices is a classic problem[12]. A number of researchers are exploring the different neural network architectures for stock market related applications. In[13] explains the use of neural network to model financial and economic time series. In[14] used ANN for stock market trend prediction. In[15] used genetic algorithm with back propagation for achieving optimization. In[16] applied neural network with genetic algorithm to the stock exchange of Singapore to predict the market direction. In[17] used feed forward neural network architecture for stock market trend prediction. A comprehensive review of neural network concepts and principals can be found in[18,19]. An overview of financial and investment applications of neural networks can be found in[20,21].

3. Artificial Neural Network

An Artificial Neural Network is an information processing paradigm that is inspired by the way biological nervous systems process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements, called the neurons, working in union to solve specific problems. An Artificial Neural Network, like people, learns by example. It is configured for a specific application, such as pattern recognition or classification, through a learning process. Learning in Neural Networks, like in biological systems, involves adjustments to the synaptic connections[22].

Artificial Neural Network is a soft computing technique, which is used to solve problems that cannot be solved by the conventional rule-based programming algorithms. An important aspect of this method is training the neurons to enrich it with knowledge so as to guide it in its analysis, predictions or classifications.

An artificial neuron is characterized by:

- Architecture (organization of neurons).
- Training or learning (determining weights on the connections).
- Activation function.

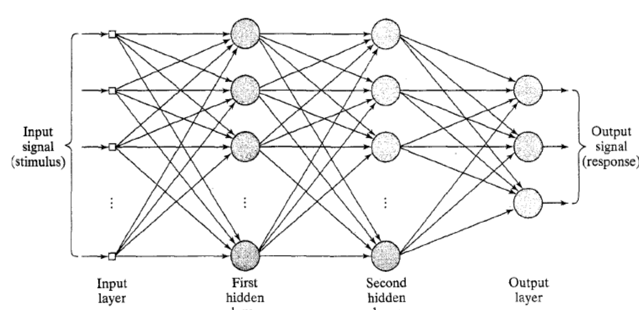

Figure 1. Neural network architecture.

Input neurons are those that act as sensors, and receive the input information from the environment, or user. Hidden neurons are those that process the input information received, depending on their weights. Output neurons are those that act as actuators and receive the processed value from the hidden neurons and produce output for the user. There can be more than one hidden layer.

4. Genetic Algorithm

Genetic Algorithm is the mimicry of the natural selection process of biological evolution. The algorithm repeatedly modifies a population of individual solutions. The input parameters are considered as population, from which the Genetic Algorithm randomly selects data, which act as parents. The parents are modified to produce children for the next generation. Over successive generations, the population evolves towards optimized solutions. Genetic Algorithm can be used to solve both constrained and unconstrained problems of optimization. Thus, discontinuous, non-differentiable, stochastic or highly non-linear problems functions can also be optimized using Genetic Algorithm.

By using Artificial Neural Network for stock prediction, the Neural Network is trained to establish all the patterns within the input dataset and hence predict future values, considering past uncertainties as well. Furthermore, the Neural Network architecture is improvised using Genetic Algorithm to obtain optimized weights, for a faster and more accurate training of the Neural Network.

BSE Sensex historical dataset for 4 years is collected from the official website of BSE SENSEX. Input parameters are collected from 24th February 2011 to 7th April 2015. With the help of this dataset, the future day closing price is determined as the output. The dataset comprises of:

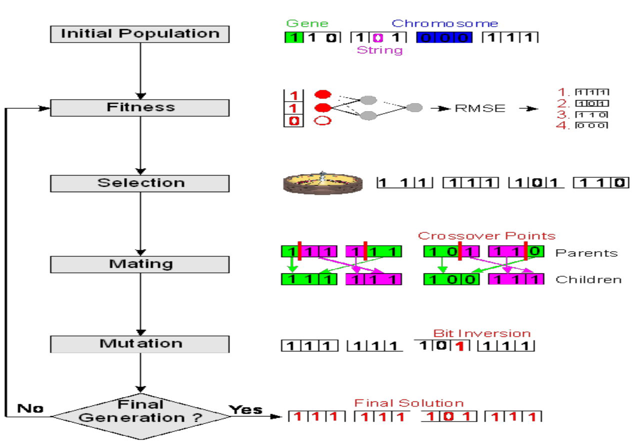

Figure 2. Genetic algorithm.

- **Open:** The price at which a security first trades upon the opening of an exchange on a given trading day. A security's opening price is an important marker for that day's trading activity, especially for those interested in measuring short-tem results, such as day traders. Additionally, securities, which experience very large intra-day gains and losses, will have those swings measured relative to their opening price for the day.
- **Close:** The final price at which a security is traded on a given trading day. The closing price represents the most up-to-date valuation of a security until trading commences again on the next trading day.
- **Low:** The lowest price to which the stock value drops in the course of the day. It determines the maximum dip in the stock price.
- **High:** The highest price to which the stock value rises in the course of the day. It determines the highest peak in the stock price.
- **Volume:** The number of stocks that have been traded during the day.

MATLAB is used to create network architecture for the prediction, train it and then predict future values accordingly.

5. Data Pre-Processing

Data pre-processing is an essential part of prediction models in Neural Network. It is essential that normalization of data is carried out prior to subjecting the data into the learning phase. The data pre-processing of the input variable of the neural network model facilitates de-trending of the data and highlight essential relationship, so as to facilitate proper network learning process.

The following linear scaling function is used in order to scale down the input values to an interval between [0, 1]:

$$X^{*}_{k,n} = \frac{X_{k,n} - \min(X_k)}{\max(X_k) - \min(X_k)}$$

Where X_k is the data series of each type of input parameters taken separately and $X_{k,n}$ is the n^{th} day's input parameter of that type.

6. Training Process

In the training process, the pre-processed data is fed into the Neural Network architecture. This phase is called the learning phase. The neural network uses weights and biases to update the input according to the desired output. The patterns in the data are established and the neural network is trained to be able to predict recurring instances of such patterns.

For this model, a 5-4-1 Neural Network architecture has been adopted. It comprises of 5 inputs, namely Open, Close, High, Low and Volume of the Sensex. There is one hidden layer, which comprises of 4 hidden neurons and there is one output neuron that gives closing Sensex value.

6.1 Training Algorithm

The following training algorithms have been considered for network training:

6.1.1 TrainGDA

It's a backpropagation gradient descent training method with adaptive learning, to prevent over-learning or under-learning in a network. This procedure increases the learning rate, but only to the extent that the network can learn without large error increases. Thus, a near-optimal learning rate is obtained for the local terrain. When a larger learning rate could result in stable learning, the learning rate is increased. When the learning rate is too high to guarantee a decrease in error, it gets decreased until stable learning resumes[23] .

Backpropagation is used to calculate derivatives of performance dperf with respect to the weight and bias variables X. Each variable is adjusted according to gradient descent:

dX = lr*dperf/dX. The following occurrences lead to the termination of training:

- Maximum number of epochs (repetitions) is reached.
- Maximum amount of time is exceeded.
- Performance is minimized to the goal.
- Performance gradient falls below min_grad.

6.1.2 TrainGDX

Traingdx is an extension of traingda. Traingdx is a network training function that updates weight and bias values according to gradient descent momentum and an adaptive learning rate. It has an additional parameter, momentum constant (mc), which helps in setting the momentum of the learning rate. By default, the value of mc is 0.9.

Backpropagation is used to calculate derivatives of performance perf with respect to the weight and bias variables X. Each variable is adjusted according to gradient descent with momentum,

dX = mc*dXprev + lr*mc*dperf/dX. The following occurrences lead to the termination of training:

- Maximum number of epochs (repetitions) is reached.
- Maximum amount of time is exceeded.
- Performance is minimized to the goal.
- Performance gradient falls below min_grad.

6.1.3 TrainRP

In sigmoid functions (also called 'squashing functions'), an infinite input range is compressed into a finite output range. When steepest descent method is used to train a multilayer network with sigmoid functions, the gradient can have a very small magnitude; and therefore, cause small changes in the weights and biases, even though the weights and biases are far from their optimal values[23].

To overcome the effects of partial derivatives, resilient backpropagation algorithm is used. Only the sign of the derivative is used to determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update[23].

Backpropagation is used to calculate derivatives of performance perf with respect to the weight and bias variables X. Each variable is adjusted according to the following:

dX = delX.*sign(gX); where the elements of delX are all initialized to del0, and gX is the gradient. Elements of delX are modified at each iteration. If an element of gX changes sign from one iteration to the next, then the corresponding element of delX is decreased by delta_dec. If it maintains the same sign, then the corresponding element of delX is increased by delta_inc.

Training stops when any of these conditions occur:

- Maximum number of epochs (repetitions) is reached.
- Maximum amount of time is exceeded.
- Performance is minimized to the goal.
- Performance gradient falls below min_grad.

6.2 Activation Functions

The following activation functions have been considered for the layers:

6.2.1 Tan-Sigmoid Activation Function

Tan-sigmoid function generates values between -1 and +1. Mathematically, it can be written as:

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

6.2.2 Log-Sigmoid Activation Function

Log-sigmoid function generates values between 0 and +1 (not inclusive). Mathematically, it can be written as:

$$\sigma(t) = \frac{1}{1 + e^{-\dagger t}}$$

6.2.3 Pure-Linear Activation Function

If linear output neurons are used in the last layer of the multilayer network, the network outputs can take on any value, unlike that of sigmoid functions. Mathematically, it can be written as: $f(x) = x$

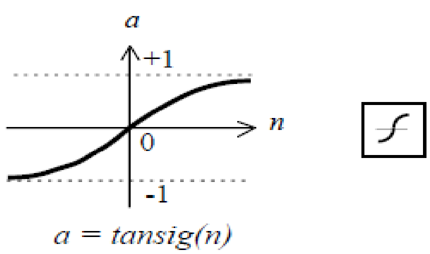

Figure 3. Tan-sigmoid.

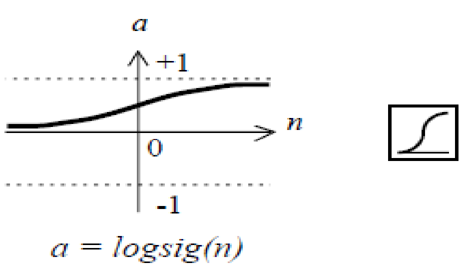

Figure 4. Log-sigmoid.

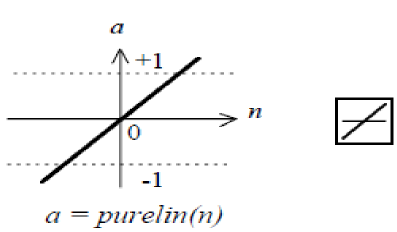

Figure 5. Pure-linear.

7. Experiments and Results

The output obtained from training the network is compared with the actual values. The performance of the network is measured by NMSE (Normalized Mean Squared Error)[24]. Mathematically,

$$NMSE = \frac{\sum_{t=1}^{N1} (P_i - O_i)^2}{\sum_{t=1}^{N1} (P_i - \bar{P}_i)^2}$$

Where P_t is the actual value of the pre-processed data series, O_t is the observed value or the predicted value for the same days closing prices of the index and P_t is the mean of the actual values.

Once the performance measurement produces optimal result for a particular network architecture and training algorithm combination, the network is used for future value prediction. The network is simulated to predict the $(n+1)^{th}$ day's closing index value, which is then compared to the actual obtained value, the next day. Hence, prediction is done for a particular period of time to test the consistency of the model. The accuracy of the predicted values is what will determine the reliability of this model.

The results obtained from various models are compared and analyzed. The model which provides the lowest Normalized Mean Squared Error is the most accurate model. This model is chosen over others for Stock Market prediction.

NMSE for traingda algorithm using tansig as activation function = 0.020845

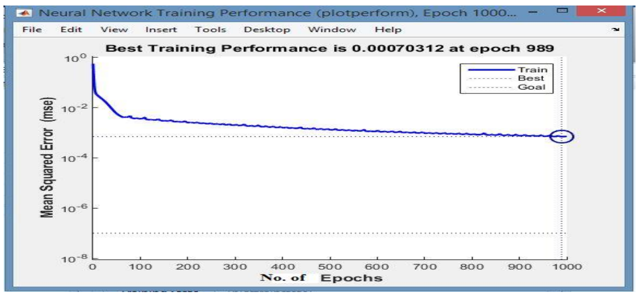

Figure 6. Training performance graph for tansig, traingda.

As seen in the training graph above, there's a steep decrease in the Mean Squared Error (mse) at the 0th epoch itself. After that, there's a gradual decline in the slope till it reaches the given epoch limit. Also, the training is not smooth as it is marked by crests and troughs throughout the training. Thus, even though it achieves an NMSE of 0.020845, the network is still not trained completely in compliance with the given data. This uncertainty is the main drawback of using traingda algorithm with tan sigmoid as the activation function.

NMSE for traingdx algorithm using tansig as activation function = 0.006599

As can be seen from the performance graph, traingdx algorithm, when using tan sigmoid as the activation function, provides a smoother training of the Neural network, as indicated by the absence of crests and troughs. There's absence of smoothness in the initial iterations, which slightly reduces the accuracy of the Neural Network. However, in comparison to traingda, traingdx is much more efficient for Stock Prediction.

NMSE for trainrp algorithm using tansig as activation function = 0.003917076

As seen in the above graph, the training achieves a very low MSE in the beginning itself. The training curve is smooth, indicating proper learning by the Neural Network. Thus trainrp training algorithm, by far, has been the most efficient in obtaining the lowest NMSE,

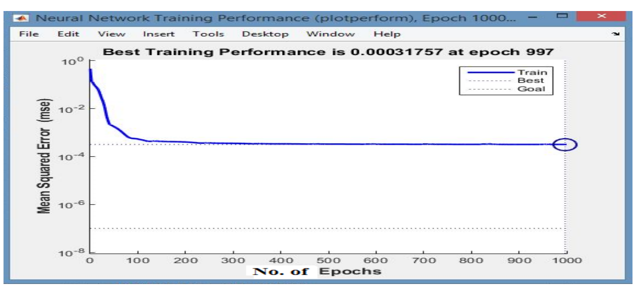

Figure 7. Training performance graph for tansig, traingdx.

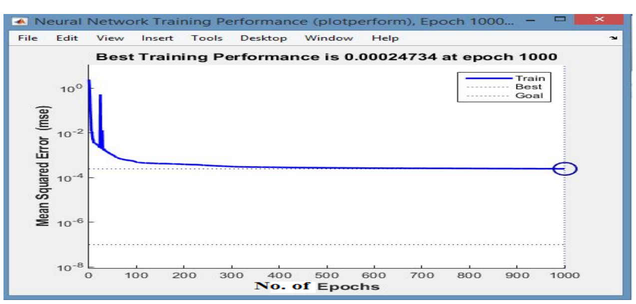

Figure 8. Training performance graph for tansig, trainrp.

and as evient from the graph, it facilitates faster and more accurate training.

NMSE for traingda algorithm using logsig as activation function = 0.01466

As seen in the training graph above, there's a steep decrease in the Mean Squared Error (MSE) in the beginning. After that, it is marked by numerous crests and troughs till it reaches the epoch limit. It achieves an NMSE of 0.01466, which indicates that the network is still not trained completely in compliance with the given data. Thus, traingdx and traingda have better accuracy when predicting Stock Market.

NMSE for traingdx algorithm using logsig as activation function = 0.010221

It is seen that traingdx algorithm, with log sigmoid as the activation function, provides erratic training for the Network. The 0th iteration indicates a steep drop, followed by a bulging drop, which slightly reduces the accuracy of the Neural Network. Also, certain discontinuity exists in the higher epochs at certain places. This considerably reduces its accuracy, and hence the NMSE.

NMSE for trainrp algorithm using logsig as activation function = 0.003745

As seen in the above graph, it provides the smoothest training curve. The training achieves a very low MSE in

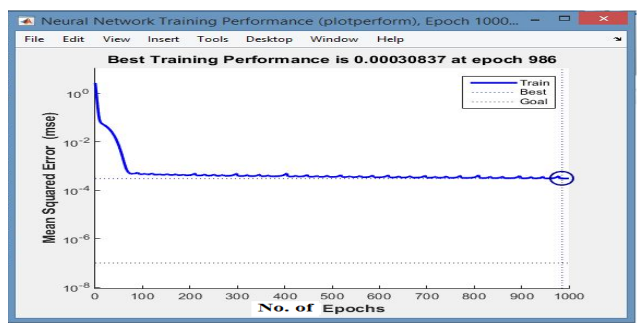

Figure 9. Training performance graph for logsig, traingda.

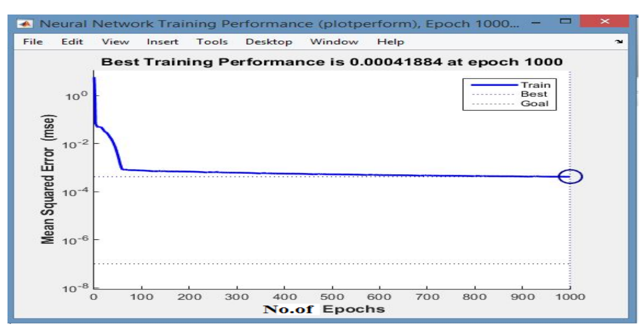

Figure 10. Training performance graph for logsig, traingdx.

the beginning itself, without any erratic training behavior or discontinuity. The training curve is smooth, indicating proper learning by the Neural Network. Thus trainrp training algorithm along with log-sigmoid is the most efficient in obtaining the lowest NMSE, and as evident from the graph, it facilitates faster and more accurate training.

7.1 Optimization using Genetic Algorithm

When stock prediction is carried out using Artificial Neural Networks, the initial values for synaptic weights are randomly chosen. However, if the initial weights as taken by the system are not in synchronization with the direction of the training, the training process will be lengthened. In order to choose the best values for the synaptic weights Genetic Algorithm is used. In this model, genetic crossover is used. An initial set of random population of size 400 is taken, and is subjected to crossover, by increasing its fitness value, where the fitness function is the Mean Squared Error (MSE) for the population. The lesser the MSE, the more is the fitness value of the individual belonging to the population. The algorithm terminates when the average change in the fitness value is less than 1e-08. The MLP network is now trained by using the optimized synaptic weight values using trainrp and logsig.

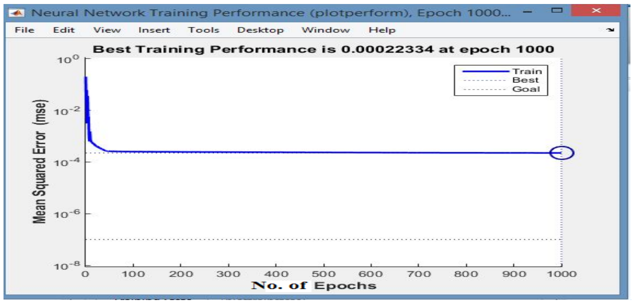

Figure 11. Training performance graph for logsig, trainrp.

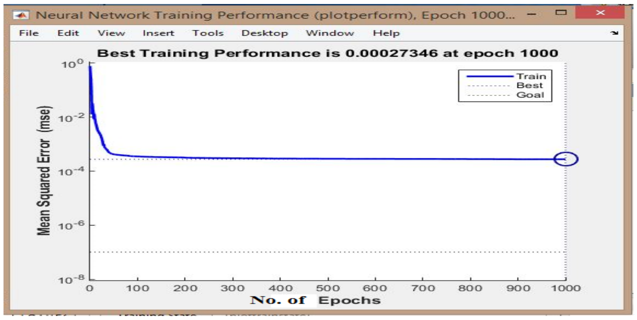

Figure 12. Training performance graph for logsig, trainrp after GA.

NMSE for training using Genetic Algorithm = 0.003092121

It is observed that the training is smoother and faster, and hence more efficient.

Once the Neural Network is trained, it is tested for future stock market predictions. The nth day's input is entered, and the network is simulated with this input data. Upon doing so, the network gives the prediction for next day's closing value. This is tabulated in the excel sheet. When the actual value is obtained the next day, it is compared against the predicted value and its accuracy is analyzed.

7.2 Testing

Testing is carried out for the networks using trainrp as the training algorithm, with tan-sigmoid and log-sigmoid as the activation functions. The data for all working days of April, is predicted for the purpose of testing.

7.2.1 Testing for Trainrp in Combination with Tan-Sigmoid

Upon testing for tan-sigmoid with Trainrp, the network provides somewhat accurate training in the beginning, with a minimum difference between the actual and predicted value being 10.5509. However, the difference increases with passing days, except for one instance where the difference reduces to 15.2722. It has a NMSE of 0.276605.

7.2.2 Testing for Trainrp in Combination with Log-Sigmoid

Upon testing for log-sigmoid with Trainrp, the network provides a more accurate result than tan-sigmoid. The minimum difference between the actual and predicted value is 5.05754187. As seen in the graph, the predicted trend almost reflects the actual trend of the stock market index. Given that human factors also play a role in deter-

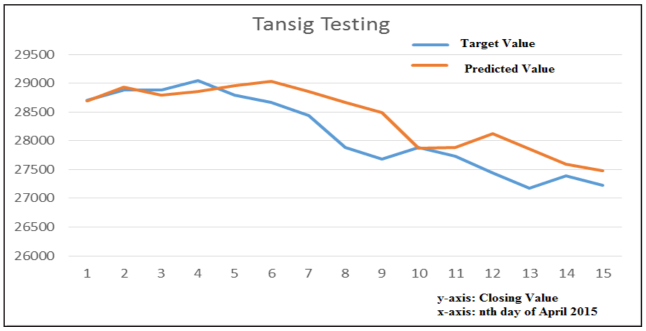

Figure 13. Testing graph for tansig, trainrp.

mining the trends of stock market (which is beyond the understanding of machine), the Neural Network architecture adopted for stock market prediction more or less predicts the next day's value as close to the original value as possible. It has a NMSE of 0.053318.

7.2.3 Testing for Trainrp with Log-Sigmoid using Genetic Algorithm

Upon testing with trainrp while including Genetic Algorithm, the following graph is obtained, showing two non-overlapping lines. The values predicted by this network do not coincide with the original values completely. Hence, the NMSE is 0.248067946. However, the network very efficiently follows the trends of the stock market index. The highs and lows, the gradient of the slopes, are all accurately depicted by improvising with the help of Genetic Algorithm. This indicates that the network training successfully inculcates the nature of the stock market and hence is able to accurately predict future trends.

7.3 Comparison between Genetic Algorithm and Non-Genetic Algorithm Model

The Figure below shows the results of the Genetic Algorithm Model and the non-Genetic Algorithm model overlapped simultaneously over the actual closing value. As seen in the Figure, the graph corresponding to Non

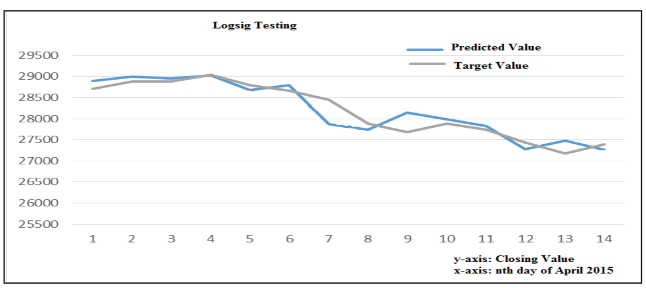

Figure 14. Testing graph for logsig, trainrp.

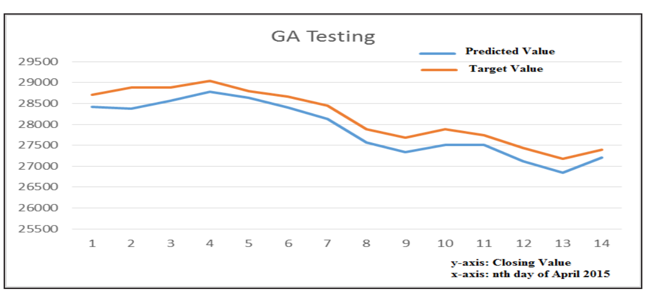

Figure 15. Testing graph for logsig, trainrp with GA.

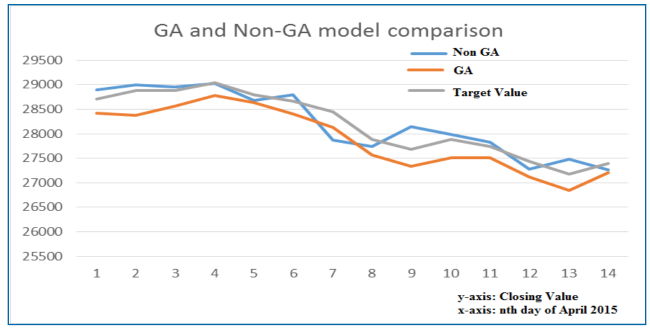

Figure 16. Comparison graph for logsig, trainrp with GA and without GA models.

GA model (Trainrp with log-sigmoid model) almost converges with the actual closing value on Day 4 and Day 11. It is very close on Day 2, 3, 5 and 10. However, it does not completely follow the trend of the actual stock index. For instance, on Day 6, the actual closing price value is shown to be decreasing from the previous day's value. Hence there's a downward slope in the graph from Day 5 to Day 6. But, the predicted value is misleading, as it shows that the closing value of Day 6 is more than that of Day 5. Similar anomalies are visible in Day 7, 9 and 13.

In contrast to this, the Genetic algorithm model, which also implements Trainrp with log sigmoid, is in compliance with the actual stock market trend. The highs and lows in the graph are in agreement with the actual closing prices. Even though the values do not converge at any point, the difference in the values is a small percentage of the actual value of the closing prices. It is the trend of the stock market which is more important to investors.

8. Conclusion

Artificial Neural Network is a successful soft-computing technique to predict the trends of the stock market index. Various training algorithms based on backpropagation of errors have been tried, out of which the Resilient Propagation training algorithm (trainrp) provides the most efficient training results, as it undoes the error that occurs due to squashing of inputs. Two activation functions, tan-sigmoid and log-sigmoid have been used in the model, out of which log-sigmoid fares better than tan-sigmoid, according to the results obtained.

The Neural Network model is further improvised by using Genetic Algorithm to initialize synaptic weights. By introducing Genetic Algorithm, the network is more effectively trained.

9. References

1. Philip MT, Paul K, Choy SO, Reggie K, Ng SC, Mak J, Jonathan T, Kai K, Lam WT. Design and implementation of NN5 for Hong Kong stock price forecasting. International Journal of Engineering Applications of Artificial Intelligence, Elsevier Publications. 2007 Jun; 20(4):453–61.
2. Ghezelbash A, Keynia F. Design and implementation of artificial neural network system for stock exchange prediction. African Journal of Computing and ICT. 2014 Mar; 7(1):153–60. ISSN: 2006-1781.
3. Haykin S. Neural Networks-A comprehensive foundation. 2nd ed. Pearson Publications; 2001.
4. Majumder M, Hussian MD. Forecasting of Indian stock market index using artificial neural network. Available from: www.nse-india.com/content/research/FinalPaper206.pdf
5. Refenes AN, Zapranis A, Francis G. Stock performance modeling using neural networks: A comparative study with regression models. Journal of Neural Networks. 1994; 7(2):375–88.
6. Schoeneburg E. Stock price prediction using neural networks: A project report. Journal of Neurocomputing. 1990 Jun; 2(1):17–27.
7. Wu SI, Zheng H. Can profits still be made using neural networks in stock market. Journal of Applied Simulation and Modeling. 410-194, 03-09-2003. ISSN (on line): 1925-8011.
8. Maciel LS, Ballini R. Neural networks applied to stock market forecasting: An empirical analysis. Journal of Brazilian Neural Network Society. 2010; 8(1):3–22.
9. Zhang Q, Wang C. Using genetic algorithm to optimize artificial neural network: A case study on earthquake prediction. Proceedings of the 2nd international conference on genetic and evolutionary computing, 2008 Sep; 25-26:128–31.
10. Weihong Z, Shunqing X. Optimization of BP neural network classifier using genetic algorithm. Intelligence Computation and Evolutionary Computation, Springer publications. 2013; 180:599–605.
11. Rajasekaran S, Pai GAV. Neural networks, fuzzy logic and genetic algorithms: Synthesis and applications. PHI; 2003.
12. Patel J, Shah S, Thakkar P, Kotecha K. Predicting stock market index using fusion of machine learning techniques. Journal of Expert Systems with Applications (Elsevier). 2015; 42(4):2162–72.
13. Tang Z, Fishwick PA. Back propagation neural nets as models for time series forecasting. ORSA Journal on Computing. 1993; 5(4):374–84.
14. Wang JH, Leu JY. Stock market trend prediction using ARIMA based neural network. Proceedings of IEEE Conference on Neural Networks; 1996. p. 2160–5.
15. Sexton C, Dorsey RSRE, Dohnson JD. Toward global optimization of neural networks: A comparison of the genetic algorithm and backpropagation. Decision Support Systems. 1998 Feb; 22(2):171–85.
16. Phua PKH, Ming D, Lin W. Neural network with genetic algorithms for stocks prediction. 5th Conference of the Association of Asian-Pacific Operations Research Societies; Singapore. 2000 Jul 5-7.
17. Abhishek K, Khairwa A, Pratap T, Prakash S. A stock market prediction model using artificial neural network. IEEE ICCCNT; 2012 Jul 26-28. p. 1–5.
18. Rumelhart D, McClelland. Parallel distributed processing. Cambridge, MA: MIT Press; 1986 Jan.
19. Caudill M, Butler C. Understanding neural networks: Computer explorations. Cambridge, MA: MIT Press; 1983.
20. Hawley D, Johnson J, Raina D. Artificial neural systems: A new tool for financial decision-making. Financial Analysts Journal. 1990 Nov-Dec; 46(6):63–72.
21. Medsker L, Turban E, Trippi R. Neural network fundamentals for financial analysts. In: Trippi R, Turban E, editors. Neural networks in finance and investing, Chicago. Probes Publishing. 1993; 2(1):59–68.
22. Wikipedia contributors. Artificial neural networks. 2015 Mar 23; Available from: http://en.wikipedia.org/wiki/Artificial_neural_network
23. Demuth H, Beale M. Back propagation, in neural networks toolbox for use with MATLAB, the Math Works. 2002; 5:4–16.
24. Sorayaei A, Salimi MR, Divkolaii MS. Probing efficiency scale of fuzzy neural network on forecasting stock exchange of the automobile industries in Iran. Indian Journal of Science and Technology. 2012 Apr; 5(4):2590–2. ISSN: 0974-6846.