Integrated Intelligent Research (IIR)
International Journal of Computing Algorithm
Volume: 03 Issue: 03 December 2014 Pages: 316-323
ISSN: 2278-2397

Optimizing the Architecture of Artificial Neural Networks in Predicting Indian Stock Prices

A. Victor Devadoss[1], T. Antony Alphonnse Ligori[2]
[1]Head and Associate Professor, Department of Mathematics, Loyola College, Chennai, India.
[2]Ph. D Research Scholar, Department of Mathematics, Loyola College, Chennai, India.
hanivictor@ymail.com, antony_ligori2001@yahoo.com

Abstract - In forecasting, the design of an Artificial Neural Network (ANN) is a non-trivial task and choices incoherent with the problem could lead to instability of the network. So a Genetic Algorithm (GA) approach is used to find an optimal topology for the prediction. This paper presents a novel approach to Optimization of ANN topology that uses GA for the forecasting of Indian Stock Prices under Bombay Stock Exchange. After determining the optimum network determined by GA, forecasting of the stock prices is found by implementing MATLAB tool. The paper is organized as follows. The first Sectiondeals with the introduction to Genetic Algorithms; Section two reviews the literature on the optimization of neural network architectures and applications of genetic algorithms in doing so. Section three gives the proposed approach in the optimization of neural network architectures. Section four presents the experimental results by the methodology described in section three and followed by results and conclusion.

Keywords -Artificial Neural Network (ANN); Genetic Algorithm (GA); Optimization.

I.INTRODUCTION

Genetic algorithm (GA) is an adaptive and robust computational model based on natural evolution [1] and tries to emulate some of the processes observed in natural evolution. GA is based on Charles Darwin's theory of natural selection known as "survival of the fittest". In Darwin's theory, individuals in a population of reproductive organisms inherit traits from their parents during each generation. Over a time, desirable traits become more and more common than the undesirable ones since individuals with the desirable traits that fit better with the environment are more likely to reproduce [2]. Genetic algorithms are good at taking larger, potentially huge search spaces and navigating them looking for optimal combination of things and solutions which we might not find in a life time [3]. Genetic algorithms are different from traditional optimization methods in the sense that GA uses a population of points at one time in contrast to the single point approach.

To determine optimal neural network using Genetic Algorithm, one has to find an appropriate representation of the network as a string of parameters which makes it conformable to the application of genetic operators and evolutionary processes. The neural network is termed as phenotype and the string of parameters of the neural network is called the genotype which contains the necessary information of the problem domain to identify a solution. Selection, crossover, mutation are all performed on a population of candidate solutions represented in the form of strings. An encoding procedure is uniformly applied to all phenotypes. Decoding of string is essential for the measurement of fitness which is always measured on the phenotype. To measure the fitness, we will first convert the string representation back into a network representation, and then evaluate its performance on a training data set. Selection of parents and which parents are selected for mating is a critical component of the evolutionary techniques and an essential aspect of selection is the probability of an individual to be selected as a parent depending on its fitness. Maintaining diversity in the population retains the scope of searching wider regions of the search space. While maintaining diversity for a reasonable number of generations even the worst member of the current population has a finite probability of becoming a parent. It is this stochastic nature that helps the genetic algorithm escape from local optima. Crossover operator and mutation operators are responsible for transfer of genetic information from one string to another and thereby new candidate solutions are generated from generation to generation. There are many types of crossover operators available which includes, single-point crossover, multipoint crossover, uniform crossover etc.

II.LITERATURE REVIEW

Although ANNs have been used extensively in many fields for a variety of applications, their accuracy level is affected due to input data set and network structure, initial weights, number of iterations, transfer function and learning rate. Choosing the correct size of a neural network can increase its speed of response and improve the performance of the overall system. Also, the network size affects network complexity and learning time but most importantly, it affects the generalization capabilities of the network [4]. Understanding these factors and designing an optimum neural network is significant for a successful ANN application. Such optimal network must learn the input data, ability to generalize for similar input data that were not in the training set and have minimum size (neurons) to accomplish the first two tasks [5].

International Journal of Computing Algorithm
Volume: 03 Issue: 03 December 2014 Pages: 316-323
ISSN: 2278-2397

Generalization is the ability of the neural network to interpolate and extrapolate data that it has not seen before. The best generalization performance is obtained by trading the training error against network complexity [6]. If a neural network is too small for a given application, it may never be able to learn the desired function and produces unacceptably large errors. On the other hand, if a neural network is too large, it may learn the training samples too well and not be able to generate the appropriate output for the inputs not included in the training set. The accuracy does not increase gradually when the size of the network is increased and do not give considerably better results [4].

As the size of input layer is equal to the number of input features and that of output layer is equal to the number of outputs, the only adjustable part in the designing of neural network is the number of hidden layers and the number of neurons in the respective hidden layers. Generally researchers use trial and error method to fix the optimum number of hidden layers and neurons. For a three layered network having n input elements in the input layers, m processing elements in the output layer, $(2n+1)$ processing elements in the hidden layer are able to reveal all the characteristic features present on the input nodes of the network [7]. So, introducing additional hidden layers increases the feature extraction capability of the network at the cost of significantly extended training and operational time of the forecaster. The study [8] shows that networks never need more than two hidden layers to solve most complex problems. But the study [9] shows that both networks with one and two hidden layers perform similarly in all other respects. In the literature there are two approaches to determine the proper size of the neural network in the literature. One is to start with a small network and iteratively increase the number of nodes in the hidden layer(s) until satisfactory learning is achieved. The techniques that are based on this approach are called constructive techniques [10] but more likely to become trapped in a local minimum. The second approach is to begin with a larger network and make it smaller by iteratively eliminating nodes in the hidden layer(s) or interconnection between nodes. These types of algorithms are called pruning. In the pruning method, the disadvantage is that the networks chosen are larger than the required size as starting point and a lot of time is spent on training the network before the actual training starts. Since there are a lot of medium-size networks that can learn the same problem, the pruning procedure may not be able to find a small-size network because it may get stuck with one of these medium-size networks.

The present study deals with the designing of network size using Genetic Algorithms. Usually a number of different size networks are encoded into a bit-string in order to form the initial population. The genetic operators act on this population and new populations are formed. A decoding procedure is applied on each member of the population in order to transform a bit-string into legitimate network architecture. The fitness of each network is evaluated by training it for a certain number of epochs and recording the network's error. The networks with low fitness values are discarded, while networks with good evaluation survive in future populations. As Genetic Algorithms are structured search and optimization techniques [11] they are employed to find the optimal number of neurons in each layer of the hidden layers.

III. THE PROPOSED APPROACH

This section deals with a method of optimizing the architecture of the network by GA. The optimal network is expected to have the capacity of learning and generalization with minimum number of neurons. According to the evolutionary metaphor, a genetic algorithm starts with a population of individuals randomly selected, which evolves toward optimum solutions through the genetic operators (selection, crossover, mutation), inspired by biological processes. Each element of the population is called chromosome and codifies a point from the search space. The search is guided by a fitness function meant to evaluate the equality of each individual in the population. The efficiency of a genetic algorithm is connected to the ability of defining a good fitness function which is either to be maximized or minimized. The fitness function defined in the study and is given as, $fitness\ \ function = \frac{1}{(SSE)^2}$, where SSE denotes Sum of Squared Errors; $SSE = \sum_{i=1}^{N} \left(a_i - f_i\right)^2$, where a_i denotes the actual value and f_i denotes the forecasted value.

The topologies that this study attempted to determine have the general structure of the form $\left(x - H_1 - H_2 - 1\right)$; where x denote the number of neurons in the input layer; H_1 and H_2, denote the number of neurons in the first and second hidden layers respectively; the output neuron in this case it is understood that only one as it is the stock price to be predicted.

The prediction of optimal neural network is done individually with single hidden layer and two hidden layers to reduce the computational complexities. The following three methods are adopted in determining the optimal architecture.

A) Moving Method
B) Factor Method
C) Mixed Method

In all the methods, an optimum architecture is constructed in predicting the stock price of Tata Consultancy Service (TCS) is estimated with the help of genetic algorithms and MATLAB. In moving method, two different studies are carried out by considering three and five consecutive moving prices. The 'moving' refers to the change in the stock prices carried out each. In the factor method, the number '3' in the input layer denotes the three inputs High Price, Low Price, Closing price of the stock and the number '5' denotes the five inputs which are Opening Price, High Price, Low Price, Closing Price and the number of transacted volumes for the stock. In the mixed method, the number '3' in the input layer denotes the three inputs which are the three consecutive closing prices of the stock and the number '5' denotes the five inputs which are Opening Price, High Price, Low Price, Closing Price and the number of Volumes transacted for the stock.

Moving Method

Case (i) Finding optimal size of the form $(x - H_1 - 1)$

The objective of this case is to find the optimal number of neurons by considering the only one hidden layer and making available the number of neurons in the input layer either 3 or 5. The choice of the input neurons is merely chosen at random which are either the previous three day closing prices of the stock under study or the five parameters pertaining to the stock (opening price, high, low, closing price and volume).

Step 1: Initial Population

The initial population of four candidate solutions is generated randomly called the network candidates (chromosomes). These candidates are random and it is selected in such a way that their binary conversions consist of equal (or almost equal) number of 0's and 1's. The size of the population is considered as $4n$, where n is the number of unknowns. To paraphrase, binary coding of these chromosomes is done in this step.

Step 2: Evaluation of Fitness function

In this step, the fitness of the networks is evaluated and the least fit network is eliminated and the remaining networks are given counts to be considered for the mating pool, a place for creation of parents for next generation (iteration).

Step 3: Cross over site

Randomly selected single point cross over is applied for the chromosomes in the mating pool and thereby new population is made ready for evaluating the corresponding fitness function. The process is continued until a specified number of generations or desired criterion is met.

Step 4: Mutation

Mutation is done at 5% also dependent on the number of 0's and 1's in the new population. Number of generation may also be taken into consideration during mutation.

Case (ii) Finding optimal size of the form $(x - H_1 - H_2 - 1)$

In this case the objective is to find the optimal number of neurons in the first and second hidden layers and the procedure being the same as case (i). The following two methods also use the above algorithm but the input variables are different as shown below.

Factor Method

In this method two cases are dealt one with three inputs (High, Low, and Closing Price) and the other with the inputs Opening price, High Price, Low Price, Close Price and Volume transacted in determining the optimal network.

Mixed Method

Here the mixed method takes 3 inputs as three consecutive prices and 5 inputs as Opening Price, High Price, Low Price, Closing Price and the number of Volumes transacted for the stock.

The calculations of reaching optimal networks are shown in the subsequent tables below.

Experimental Results of the Proposed Method

(a)Moving Method

(i) Single Hidden Layer Calculations

Here the number of neurons in the single hidden layer is determined and the optimal network of the form $(x - H_1 - 1)$ is found. The calculations are shown below.

In the fourth iteration we find the sum of SSE is increasing when it is compared with its previous calculations which show an indication of optimality in the network structure. The optimal networks identified in this method are 3-10-1 with the least SSE of 1.38 and 5-11-1 with its SSE being 1.50.

(ii) Two Hidden Layer Calculations

The number of neurons in the first and second hidden layers is determined and thereby an optimal network of the form $(x - H_1 - H_2 - 1)$ is found. The first iteration calculations are shown in this section.

The optimal networks identified in the final iteration by this method are 3-5-20-1 and 5-13-20-1 with SSE of 1.50 each. Hence by using the moving method, considering single and two hidden layer calculations shown above, the optimal networks are identified are 3-10-1, 5-11-1, 3-5-20-1, and 5-13-20-1

TABLE I. FIRST ITERATION

Initial Population	Network	SSE	Fitness Function	Pselect	Actual Count	Mating Pool	Mate randomly selected	Cross Over site	New Population
00011	3-3-1	1.64	0.3718	0.235	0	01010	1 and 4	1	00110
00110	3-6-1	1.60	0.3906	0.247	1	01010	2 and 3	3	01011
01010	5-10-1	1.56	0.4109	0260	2	01111		3	01110
01111	5-15-1	1.57	0.4057	0.257	1	00110		1	01010
	Sum	6.37	1.579	1					

International Journal of Computing Algorithm
Volume: 03 Issue: 03 December 2014 Pages: 316-323
ISSN: 2278-2397

TABLE II. SECOND ITERATION

Initial Population	Network	SSE	Fitness Function	Pselect	Actual Count	Mating Pool	Mate randomly selected	Cross Over site	New Population
00110	3-6-1	1.55	0.4162	0.250	1	01110	1 and 4	2	01010
01011	3-11-1	1.57	0.4057	0.244	0	01110	2 and 3	1	01110
01110	5-14-1	1.52	0.4328	0.260	2	00110		1	01011
01010	5-10-1	1.56	0.4109	0.247	1	01010		2	00110
	Sum	6.20	1.666	1					

TABLE III. THIRD ITERATION

Initial Population	Network	SSE	Fitness Function	Pselect	Actual Count	Mating Pool	Mate randomly selected	Cross Over site	New Population
01010	3-10-1	1.38	0.5251	0.292	2	01010	1 and 4	2	01110
01110	3-14-1	1.56	0.4006	0.229	0	01010	2 and 3	1	01011
01011	5-11-1	1.50	0.4109	0.247	1	01011		1	01010
00110	5-6-1	1.55	0.4162	0.232	1	00110		2	00110
	Sum	5.99	1.797	1					

TABLE IV. FINAL ITERATION

Initial Population	Network	SSE	Fitness Function
01110	3-14-1	1.51	0.4386
01011	3-11-1	2.37	0.1780
01010	5-10-1	1.68	0.3543
00110	5-6-1	1.64	0.3718
	Sum	7.20	1.343

TABLE V. FIRST ITERATION (TWO HIDDEN LAYERS)

Initial Population	Network	SSE	Fitness Function	Pselect	Actual Count	Mating Pool
00101-01100	3-5-12-1	1.60	0.3906	0.136	1	01101-10110
10010-00011	3-18-3-1	1.65	0.3673	0.128	1	01101-10110
00110-00101	3-6-5-1	2.53	0.1562	0.054	0	10101-10100
01010-00111	3-10-7-1	1.56	0.4109	0.143	1	10101-10100
01110-01001	5-14-9-1	1.61	0.3858	0.134	1	01010-00111
01011-01111	5-11-15-1	2.31	0.1874	0.065	0	00101-01100
01101-10110	5-13-22-1	1.34	0.5569	0.194	2	01110-01001
10101-10100	5-21-20-1	1.55	0.4162	0.145	2	10010-00011
	Sum	14.15	2.8714	1		

International Journal of Computing Algorithm
Volume: 03 Issue: 03 December 2014 Pages: 316-323
ISSN: 2278-2397

TABLE VI. FORECASTED PRICES USING OPTIMAL NETWORKS (MOVING METHOD)

Date	Actual Value (in Rs).	Network Structure			
		(3-10-1)	(5-11-1)	(3-5-20-1)	(5-13-20-1)
17-04-2013	1459.20	1501.73	1497.00	1506.70	1526.12
18-04-2013	1450.70	1480.08	1457.59	1448.71	1490.12
22-04-2013	1425.25	1481.43	1463.32	1451.33	1493.63
23-04-2013	1429.60	1441.46	1420.42	1399.57	1453.75
25-04-2013	1402.10	1420.04	1441.77	1428.57	1439.74

TABLE VII. PERFORMANCE EVALUATION OF THE OPTIMAL NETWORKS (MOVING METHOD)

Forecasting Performance	Optimal Networks			
	(3-10-1)	(5-11-1)	(3-5-20-1)	(5-13-20-1)
MAPE	2.20	1.84	1.84	3.29
MAD	31.58	26.332	26.41	47.30
RMSE	35.47	30.28	30.14	50.42

TABLE VIII. FORECASTING ACCURACY OF THE IMPLEMENTED ARCHITECTURE

Optimal Networks	Forecasting Accuracy
(3-10-1)	97.80%
(5-11-1)	98.16%
(3-5-20-1)	98.16%
(5-13-20-1)	96.71%

(b) Factor Method

By using the factor method, considering single and two hidden layer calculations shown above, the optimal networks identified3-3-1, 5-6-1, 3-9-12-1, and 5-18-9-1.

TABLE IX. FORECASTED PRICES USING OPTIMAL NETWORKS (FACTOR METHOD)

Date	Actual Value (in Rs).	Network Structure			
		(3-3-1)	(5-6-1)	(3-9-12-1)	(5-18-9-1)
17-04-2013	1459.20	1451.99	1454.38	1405.41	1429.98
18-04-2013	1450.70	1445.62	1480.93	1407.93	1392.54
22-04-2013	1425.25	1414.52	1389.43	1387.33	1417.36
23-04-2013	1429.60	1426.99	1413.95	1416.20	1432.75
25-04-2013	1402.10	1391.97	1365.33	1372.92	1372.83

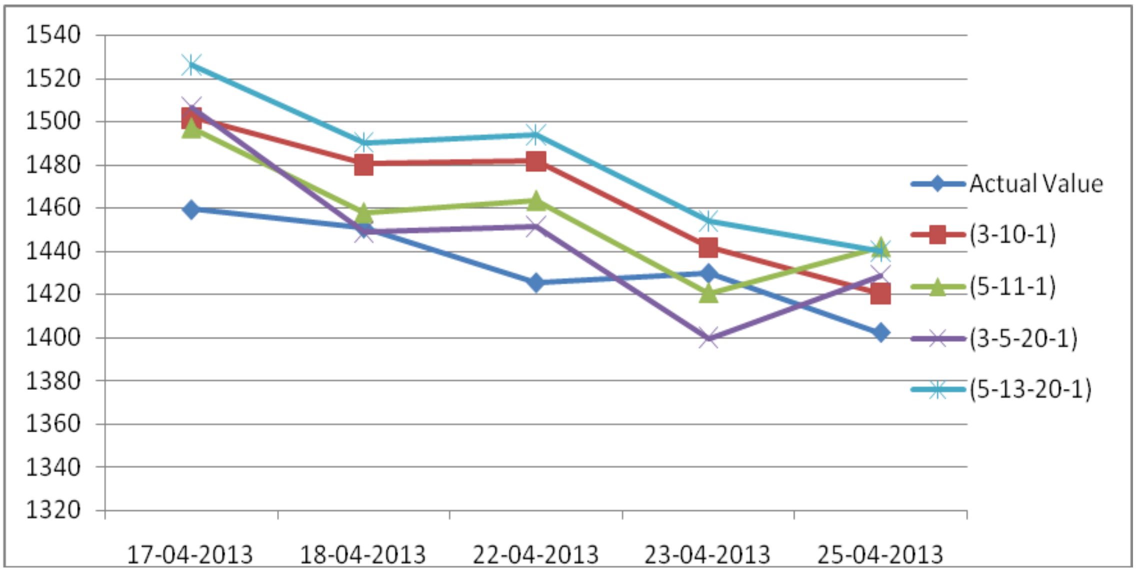

Figure 1: Graph of the forecasted Prices using Moving Method

TABLEX. PERORMANCE EVALUATION OF THE OPTIMAL NETWORKS (FACTOR METHOD)

Forecasting Performance	Optimal Networks			
	(3-3-1)	(5-6-1)	(3-9-12-1)	(5-18-9-1)
MAPE	0.50	1.73	2.46	1.78
MAD	7.15	24.66	35.41	25.54
RMSE	7.78	27.63	37.92	32.14

TABLE XI. FORECASTING ACCURACY OF THE IMPLEMENTED ARCHITECTURE

Optimal Networks	Forecasting Accuracy
(3-3-1)	99.50%
(5-6-1)	98.27%
(3-9-12-1)	97.54%
(5-18-9-1)	98.23%

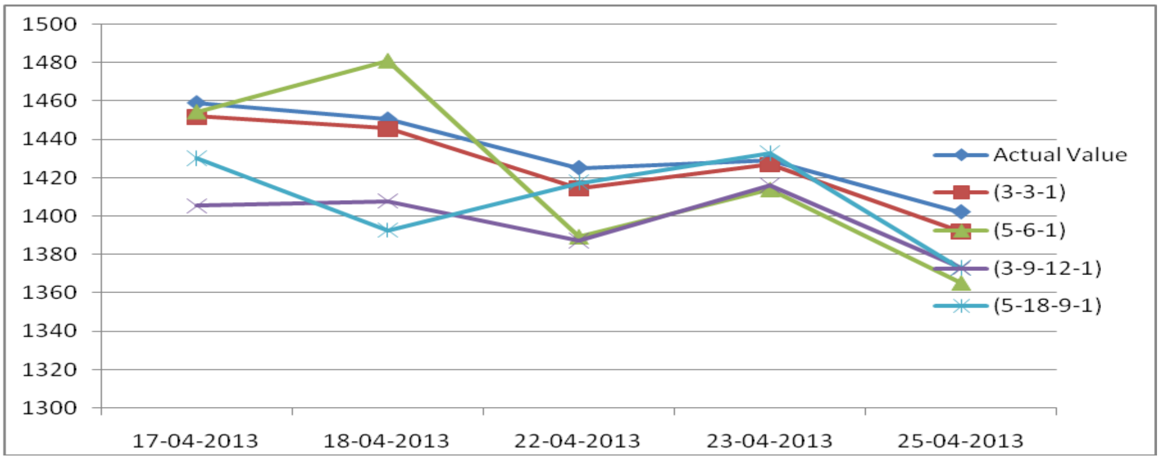

Figure 2: Graph of the forecasted Prices using Factor Method

(c) Mixed Method

The optimal networks determined by this method are 3-13-1, 5-3-1, and3-6-5-1.

International Journal of Computing Algorithm
Volume: 03 Issue: 03 December 2014 Pages: 316-323
ISSN: 2278-2397

TABLE XII. FORECASTING PERFORMANCE OF THE OPTIMAL NETWORKS (MIXED METHOD)

Date	Actual Value (in Rs).	Optimal Networks					
		3-13-1		5-3-1		3-6-5-1	
		Predicted Value	Forecasting Error	Predicted Value	Forecasting Error	Predicted Value	Forecasting Error
17-04-2013	1459.20	1519.53	4.13%	1441.42	1.22%	1487.02	1.91%
18-04-2013	1450.70	1464.49	0.95%	1428.78	1.51%	1459.66	0.62%
22-04-2013	1425.25	1479.59	3.81%	1411.72	0.95%	1451.17	1.82%
23-04-2013	1429.60	1433.49	0.27%	1424.19	0.38%	1419.03	0.74%
25-04-2013	1402.10	1446.48	3.17%	1388.42	0.98%	1429.72	1.97%

TABLE XIII. FORECASTING ACCURACY OF THE IMPLEMENTED ARCHITECTURE

Optimal Networks	Forecasting Accuracy
(3-13-1)	97.53%
(5-3-1)	98.99%
(3-6-5-1)	98.59%

TABLE XIV. PERFORMANCE EVALUATION OF THE OPTIMAL NETWORKS (MIXED METHOD)

Sl No	Forecasting Performance	Optimal Networks		
		3-13-1	5-3-1	3-6-5-1
1	MAPE	2.467	1.007	1.410
2	MAD	35.350	14.465	20.177
3	RMSE	41.877	15.467	21.911

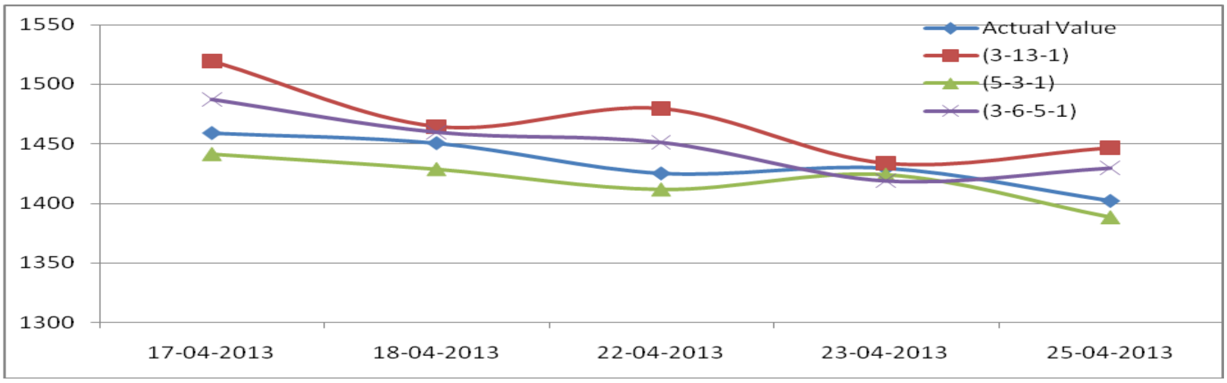

Figure 3: Graph of the forecasted Prices using Mixed Method

TABLE XV. COMPARATIVE STUDY OF MOVING, FACTOR AND MIXED METHODS NETWORKS (MOVING METHOD)

Moving Method		Factor Method		Mixed Method	
Optimal Networks	MAPE Performance	Optimal Networks	MAPE Performance	Optimal Networks	MAPE Performance
(3-10-1)	2.20	(3-3-1)	0.50	(3-13-1)	2.467
(5-11-1)	1.84	(5-6-1)	1.73	(5-3-1)	1.007
(3-5-20-1)	1.84	(5-18-9-1)	1.78	(3-6-5-1)	1.410
Average	1.96		1.34		1.63

International Journal of Computing Algorithm
Volume: 03 Issue: 03 June 2015 Pages: 316-323
ISSN: 2278-2397

From the above table it is observed that the network 5-3-1 performs better than the other networks by observing forecasting performance measures. Moreover, among the optimal networks the smallest network 5-3-1 appeared to be the best choice, which has large generalization abilities [4].

IV.RESULTS AND DISCUSSION

Here the top three optimal networks from each of the methods were displayed in the table. The average of the MAPE performance is also calculated in the table. The forecasting accuracy and the implemented ANN architectures are shown in the tables XI, table XIII and table IV. The table shows the optimal networks obtained by Moving, Factor and Mixed methods. Corresponding performance measure Mean Absolute Percentage Error (MAPE) is also shown for the analysis. In each method, instead of finding a single optimal network, a group of networks have been identified. From the table, it is observed that the MAPE performance in the case of Factor Method is the minimum.

V.CONCLUSION

Genetic algorithm is applied in order to determine the optimal size of a neural network for forecasting stock prices for TCS under Bombay Stock Exchange. Genetic algorithm renders an efficient way in determining optimal network. The neural network thus constructed forecasts the stock prices of TCS with good accuracy. The results show that a company's fundamentals are significant in determining its stock prices. Hence it is enough for investors to look into fundamental analysis before investing in a company. The goal of identifying optimal architecture for artificial neural networks is successfully achieved by incorporating Genetic Algorithm with ANNs. The predicted results demonstrate that the integration of Genetic Algorithm and Artificial Neural Networks brings forecasters to predict with better accuracies. This hybridization brings a lot of hope for the researchers in determining optimal networks for the stock price prediction. Above all, the Factor Method gives a scope of improving the forecasting accuracy by varying the activation functions and learning algorithms and in all possible ways.

REFERENCES

[1] E Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison –Wesley Publishing Company, Inc. Reading MA, 1989.

[2] A.Kopel and X.H. Yu, "Optimize Neural Network Controller Design using Genetic Algorithm", Proceedings of the 7th World Congress on Intelligent Control and Automation, China, pp 2012-2016, 2008.

[3] S.Rajasekaran and G.A. Vijayalakshmipai, "Neural Networks, Fuzzy Logic and Genetic Algorithms", Prentice Hall of India, 2012.

[4] T.Kavzoglu, "Determining Optimum Structure for Artificial Neural Networks", Proceedings of the 25th Annual Technical Conference and Exhibition of the Remote Sensing Society, Cardiff, UK, pp 675-682, 1999.

[5] I.Illeana, C. Rotor and A. Incze, "The Optimization of Feed Forward Neural Networks Structure Using Genetic Algorithms", Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics, Thessaloniki, Greece, pp 223-234, 2004.

[6] Y.LeCun and J.S Denker and S.A. Solla, "Optimal Brain Damage", In advances in Neural Information Processing Systems, No 2, pp 498-505, 1990.

[7] H.R. Nielsen, "Kolmogorov's Mapping Neural Network Existence Theorem", IEEE Conference on Neural Networks: San Diego, CA, III, pp 11-14, 1987.

[8] G.Cybenko,"Approximation by Superpositions of a Sigmoidal Function", Mathematical Control Signal Systems, No. 2, pp. 303-314, 1989.

[9] J.DeVilliers and E. Bernard, "Backpropagation Neural Nets with One and Two Hidden Layers", IEEE Trans Neural Netw, Vol 4, No.1, pp 136-141, 1993.

[10] Y. Hirose, K. Yamashita, and S. Hijiya, "Back-Propagation Algorithm which varies the number of hidden units", Neural Networks, No.4, pp 61-66, 1991.

[11] D.Susmita, K.P. Sankar and A.Ghosh, "Genotypic and Phenotypic Assortative Mating in Genetic Algorithm", Journal of Information Sciences, Vol. 105, pp. 209-226, 1998.

[12] http://neuroph.sourceforge.net

[13] http://in.finance.yahoo.com