GENETICALLY TUNED ARTIFICIAL NEURAL NETWORK FOR STOCK INDEX PREDICTION

Dinesh K. Sharma, University of Maryland Eastern Shore, USA
dksharma@umes.edu
H.S. Hota, Guru Ghasidas Central University, Bilaspur (C.G.), India
proffhota@gmail.com

ABSTRACT

In this paper, we propose an intelligent system by combining an artificial neural network (ANN) with genetic algorithm (GA) for stock price prediction. The proposed system and ANN are tested ANN on the index funds of DOW30 and NASDAQ100, and the results are compared. Index data are partitioned to check the robustness of the hybrid model. Simulation results show that the hybrid system outperforms the ANN model.

Keywords: Genetically Tuned Artificial Neural Network, Back Propagation Artificial Neural Network.

INTRODUCTION

Forecasting the stock market is an extremely difficult and complex process due to the interaction of many variables. Investment professionals rely heavily upon intelligent systems to forecast growth rates, net income, price-earnings ratios, etc. Many traditional statistical methods and expert systems have been used to predict stock prices. However, they have not proved to be highly reliable. These models require many simplifying assumptions and continued review and refinement as economic conditions change (Trippi & Turban, 1996).

In the literature, several authors have reported that no single technique can capture prices entirely in the presence of many variables. Individual techniques have some strengths and weaknesses (Branke, 1995; Yao, 1999; Kwon & Moon 2007; Mandziuk & Marcin, 2011). Therefore, there is a need to develop hybrid systems using two or more techniques to combine the strengths. The stock market movements can be predicted better combining more than one technique instead of a single forecasting technique because of multiplicity of the variables. Since back propagation artificial neural network (BPANN) is one of the popular ANN techniques, which is suitable for learning large amount of input – output patterns, but may face a problem of local minima and network paralysis which will affect the accuracy of the system. On the other hand, GA does not guarantee the global optimum solution. A suitable integration of ANN and GA may overcome the problems of the both techniques where ANN is used for learning while GA is used for optimization of weights ((Rajasekaran & Pai, 1996).

In this study, we have applied GA for optimizing the weights of ANN so that result produced by ANN will be more accurate, and hence error will be minimized. Also, results and trends of actual and predicted values for DOW30 and NASDAQ100 indices are compared for both genetically tuned artificial neural network (GANN) and BPANN.

GENETICALLY TUNED ARTIFICIAL NEURAL NETWORK

Back propagation artificial neural network (BPANN) is a neural network technique which is able to train nonlinear data and is based on gradient descent. This network is trained with popular error back propagation algorithm (EBPA). This algorithm has two passes: feed forward phase in which output is calculated and feed backward phase in which the calculated error is propagated back to the network to adjust the weights. On the other hand, GAs are well known optimization algorithms based on survival of fittest concept. As discussed above, both the techniques have own strength and weakness. To overcome these problems and to utilize best features of both, a suitable integration of ANN with GA is required where ANN is used for learning while GA is used for optimization of weights. A 4-4-1 neural network is constructed for forecasting stock index and trained in a supervised manner. A complete algorithm (Rajasekaran & Pai, 1996) to optimize weights of ANN using GA is shown in table 1.This algorithm has two sub algorithms namely fitness-value () and GANN(), where fitness-value () algorithm is used to calculate fitness value of chromosomes in the population, while GANN() is used to obtain the weights from the chromosomes by presenting input and output pattern.

Table 1: Algorithm for genetically tuned artificial neural network (GANN)

Algorithm: Fitness-Value (I_i,O_i,C_i,P_i,W_i,N,E,F_i)

Step 0: Start.

Step 1: For each chromosome C_i belongs to the current population P_i performs step 2 to 4.

Step 2:Extract weight W_i .

Step 3:Keeping weight W_i fixed and then training the network in a supervised manner by presenting N input instance with I_i as input and O_i as output.

Step 4:Calculate error (E) as Root mean square error (RMSE).

Step 5:Output fitness value F_i for each C_i.

Step 6:Stop.

Algorithm: GANN (P_i)

Step 0: Start and set all initial parameters as default.

Step 1: Generate initial population P_i.

Step 2: Perform step 3 to step 8 while current population(P_i) has not converged.

Step 3: Calculate fitness value using algorithm Fitness-Value ().

Step 4:Apply replaces operator

Step 5: Apply evolution as a combination of three operators: selection,cross over and mutation to produce offspring.

Step 6: set i=i+1.

Step 7: Designate this offspring as the current population P_i.

Step 8: Calculate fitness value of all the chromosomes of current population P_i.

Step 9: Weight extraction.

Step 10: Stop.

In order to understand the above algorithm more clearly, important steps are explained in more detail these are:

Initial Population and Coding: GA always requires initial population to start the evolution, which is generated randomly. The coding is represented as strings of bits containing all the

necessary information to describe a point of space in the population. Real coded genes are used in this study and initial population is generated randomly.

Fitness Value: The fitness value in this study is calculated with the help of the following formula

$$F=\frac{1}{E} \quad (1)$$

Where, E isthe root mean square of the error of the form given below.

$$E=\sqrt{\frac{(E_1+E_2+E_3+\ldots.+E_n)}{n}} \quad (2)$$

Where, E_n for i=1,2,….n is the error for the n^{th} instance of the data set

Weight Extraction: In order to train the ANN, weights must be extracted from each of the chromosomes from the current population. Using these weights, all the training data are supplied to the ANN which checks for the error. If the results are not satisfactory, process will be continued. Let $x_1, x_2, \ldots x_d, \ldots, x_L$ represent a chromosomes and $x_{kd+1}, x_{kd+2}\ldots, x_{(k+1)d}$ represent the k^{th} gene (k>=0) in the chromosome. Then the weight extraction formula (Rajasekaran & Pai, 1996) is given by the following equation:

$$W_k=\begin{cases} +\dfrac{X_{kd+2}10^{d-2}+X_{kd+3}10^{d-3}+---+X_{(k+1)d}}{10^{d-2}} \text{ if } 5\le X_{kd+1}\le 9 \\ -\dfrac{X_{kd+2}10^{d-2}+X_{kd+3}10^{d-3}+---+X_{(k+1)d}}{10^{d-2}} \text{ if } 0\le X_{kd+1}\le 5 \end{cases} \quad (3)$$

Replace: The worst fit chromosome in the population is replaced with best fit chromosome in order to produce better set of chromosomes for the next generation.

Evolution: Evolution is a combination of three genetic operators: selection, crossover and mutation. Chromosomes are selected from the population using roulette wheel selection to be parents to cross over and will produce offspring. Crossover is a way of exchanging genetic material of two parent chromosomes. Two point cross over is applied here. Mutation is an optional operator which permits to change a small part of chromosomes to give something new to the individual with very low mutation probability. We have applied two points mutation operator in the evolution process to optimize the weights of ANN.

DATA DESCRIPTION AND EVALUATION CRITERIA

The original data set covers the period from March 01, 2000 to Feb 02, 2012 was downloaded (http://www.finance.yahoo.com), which consists of 3000 samples of daily index price of DOW30 and NASDAQ100.A smooth trajectory of weight space can tend towards the optimum solution. To achieve smooth weight updation, data normalization is required, hence the data is normalized in the range of [0 1] using following formula:

$$D(Norm)=\frac{D(i)}{\max(D)}$$

Where, *D(i)* is the i^{th} instance and max(*D)* is the maximum value of a particular feature.

Training and testing sample size of data must be optimum because the result of ANN based system is highly depending on testing and training size, accuracy may vary based on this, hence in order to choose best sample size for a particular model and particular data, index data is partitioned into two different partitions for both training and testing: Partition1 (60:40) and Partition 2 (80:20).

To evaluate the models, the mean absolute percentage error (MAPE) as written in equation (4) is considered.

$$MAPE = \frac{1}{N}\left[\sum_{i=1}^{N}\frac{|AV_i - PV_i|}{AV_i}\times 100\right] \quad (4)$$

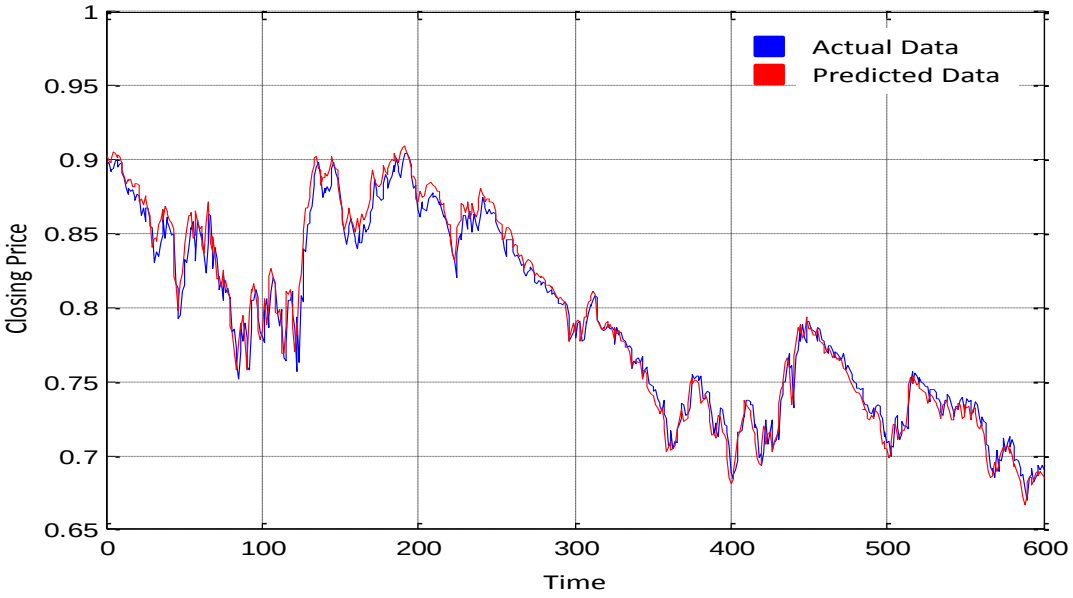
Where AV_i and PV_i are actual value and predicted value respectively for ith instance and N is the total number of instances (Samples).

EXPERIMENTAL WORK AND RESULT DISCUSSION

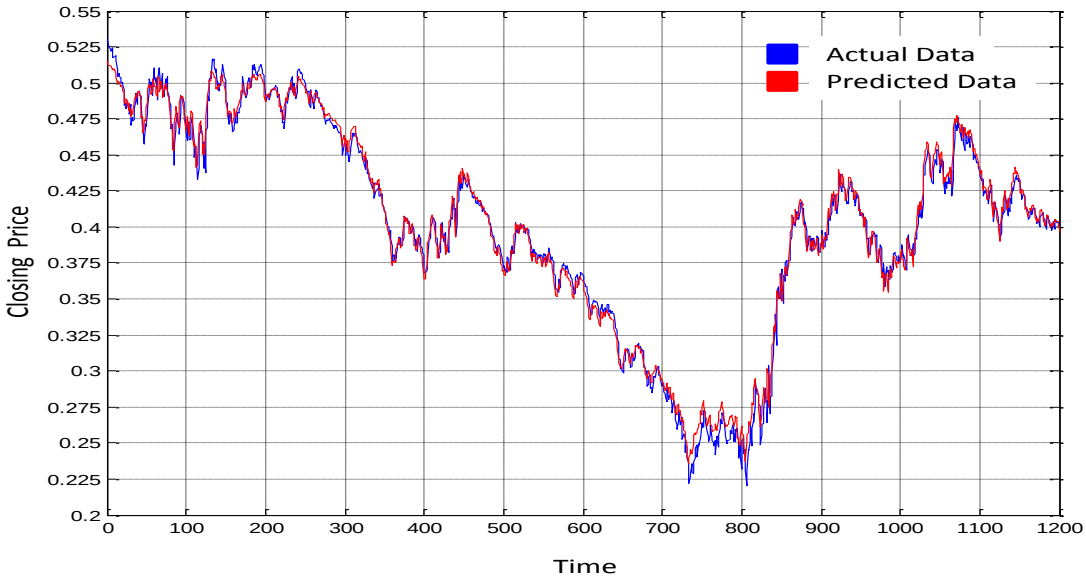
To simulate the entire work, custom software as developed using JAVA programming language. This experimental work and result discussion are explained with the help of two different subsections these are: (a) Experiment with DOW30 and NASDAQ100 data sets and (b) Error measures. The study investigated the robustness of the GANN as compare to BPANN using two partitions of both the data sets. The system training took place using each partition with the help of GA to optimize the weights and independently to BPANN.

(a)Experiment with DOW30 and NASDAQ100 data set: Both the models (GANN and BPANN) are trained and then tested using both the partitions one by one. A comparative graph in between actual and predicted index closing price by GANN at the testing stage for best partition i.e. partition 1 for DOW30 and partition 2 for NASDAQ100 are shown respectively in figure 1 (a) and 1(b). From these figures, it is clear that predicted value is very close to the actual value.

(b) Error Measures: MAPE is used as error measures, calculated using equation(4) after simulation of BPANN and GANN with both the partitions of DOW30 and NASDAQ100 data sets and the results are tabulated in table 2 for both training and testing samples .We can observe from this table that the training error is less than testing error for both the partitions of DOW30 data set .The same trend can be seen in case of NASDAQ100 data set, range of MAPE in this case is higher than that of DOW30. This may be due to highly nonlinear trends of NASDAQ100 time series data. Minimum testing error 1.0 for DOW30 is found in case of partition 2 while it is 1.55 for NASDAQ100 data set using partition 1, which are lower than corresponding values of BPANN i.e. 1.10 for DOW30 and 1.63 for NASDAQ100 data set. These results clearly show that GANN is performing well as compare to traditional BPANN model.



(a)



(b)

Figure 1: Actual Vs. Predicted output using GANN at the testing stage for best partition. (a) Partition 2 of DOW30, (b) Partition 1 of NASDAQ100.

Table 2: Comparative MAPE of two models

Data Partition	DOW30				NASDAQ100			
	BPANN		GANN		BPANN		GANN	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Partition 1	1.23	1.69	1.07	1.56	1.41	1.63	1.37	1.55
Partition 2	1.21	1.10	1.11	1.0	1.95	1.99	1.90	1.80

CONCLUSION

Prediction of a stock market index more accurately is a challenging task due to high uncertainty in this area. Successful prediction of a stock index may be beneficial for an individual or any organization in terms of money. In this piece of research work, a back propagation artificial neural network (BPANN) and a genetically tuned artificial neural network (GANN) are tested on two different partitions of DOW30 and NASDAQ100 data sets. Results obtained from both models are compared in terms of MAPE and it is found that GANN is performing well for both the partitions. Due to the problem of local minima, sometimes BPANN may fail to optimize the weights of ANN. In those cases, we can rely on GANN, because GANN works in global optimizations. Also, GANN can capture the nonlinearity situation of stock market in a more intelligent way.

REFERENCES

Branke, J. (1995). *Evolutionary algorithms for neural network design and training*, Univ. Karlsruhe, Inst. AIFB, Karlsruhe, Germany, Tech. Rep. 322.

Kwon, Y-K, & Moon, B-R. (2007). Hybrid neurogenetic approach for stock forecasting. *IEEE Transactions on Neural Networks*, 18(3), 851-864.

Mandziuk, J., & Marcin, J. (2011). Neuro-genetic system for stock index prediction. *Journal of Intelligent & Fuzzy*, 22(2-3), 93-123.

Rajasekran, S., & Pai, G.A., Vijayalaxmi, (1996). Genetic algorithm based weight determination for backpropagation network. Proc of the Fourth International Conference on Advanced Computing, 73-79.

Trippi, R., & Turban, E. (1996). *Neural networks in finance and investing*. Chicago: Probus Publishing Company.

Yahoo Finance (2012). Web Source http://www.finance.yahoo.com last accessed on March 2012.

Yao, X. (1999). Evolving artificial neural networks. *Proc. IEEE*, 87(9), 1423–1447.