Genetic Algorithm Based Backpropagation Neural Network Performs better than Backpropagation Neural Network in Stock Rates Prediction

Asif Ullah Khan
Asst. Prof. Dept. of Computer Sc. & Engg.
All Saints' College of Technology, Bhopal.

T. K. Bandopadhyaya
Professor, Bansal Institute o
Science and Technology,
Bhopal.

Sudhir Sharma
Professor, RGPV,
Bhopal.

Abstract

The prevailing notion in society is that wealth brings comfort and luxury, so it is a challenging and daunting task to find out which is more effective and accurate method for stock rate prediction so that a buy or sell signal can be generated for given stocks. This paper presents a Back Propagation Neural Network and Genetic Based Backpropagation Neural Network to predict the stock price of the day. Stock rate prediction accuracy of backpropagation neural network and genetic algorithm based backpropagation neural network has been compared. The results showed that the genetic algorithm based backpropagation neural network predict stock price more accurately as compared to backpropagation neural network.

Key Words - Backpropagation Neural Network, Genetic Algorithm Based Backpropagation Neural Network, Technical Analysis.

1. Introduction.

Prediction of financial markets has long been an attraction in the minds of equity investors. Technical Analysis [1] provides a framework for studying investor behavior, and generally focuses only on price and volume data. Technical Analysis using this approach has short-term investment horizons, and access to only price and exchange data. With the advent of powerful computers much attention has been focused on this field. Equity market prices depend on many influences. Key factors that influence future equity prices can be broadly divided into quantitative and qualitative types. Primary quantitative factors include open rate, high rate, low rate, close rate and volume for individual equities. Qualitative factors include socio-economic, political, international, regional and performance factors to name but a few. The aim of this paper is to compare backpropagation neural network and genetic algorithm based back propagation neural network to find technique that can predict stock price more accurately. Preliminary research performed on Indian National Stock Exchange market has suggested that the inputs to the system may be taken as: previous day's closing rate and volume of last trading day for backpropagation neural network and genetic algorithm based backpropagation neural network. After the inputs have been determined, the data have been gathered of Maruti stock for the period of 01-Jan-2004 to 29-Dec-2006 for training backpropagation neural network and genetic algorithm based backpropagation neural network. For testing purpose we have used testing data of Maruti Stock for the period of 02-Jan-2007 to 30-Mar-2007. Training and testing is performed using two network architectures.

1). One Hidden Layer BPN Network
2). One Hidden Layer GA-BPN Network

The results are compared between BPNN(Backpropagation Neural Network) and GA-BPNN (Genetic Algorithm Based Backpropagation Neural Network). It has been found that GA-BPNN has better performance than BPNN which generally performs better than technical indicators.

3. Application of Neural Networks in Market Prediction.

3.1 Overview

The ability of neural networks to discover nonlinear relationships [2] in input data makes them ideal for modeling nonlinear dynamic systems such as the stock market.

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A neural network method can enhance an investor's forecasting ability [3]. Neural networks are also gaining popularity in forecasting market variables [4]. A trained neural network can be thought of as an expert in

Manuscript received July 5, 2008.
Manuscript revised July 20, 2008.

the category of information it has been given to analyze. be used to provide projections given new situations of interest and answer "what if" questions. Traditionally forecasting research and practice had been dominated by statistical methods but results were insufficient in prediction accuracy [5]. Monica et al's work [6] supported the potential of NNs for forecasting and prediction. Asif Ullah Khan et al [7] used the back propagation neural networks with different number of hidden layers to analyze the prediction of the buy/sell.Neural networks using back propagation algorithms having one hidden layer give more accurate results in comparison to two, three, four and five layers.

3.2. Backpropagation Neural Network Training.

Initialize all weights in network;
While terminating condition is not satisfied { for each training sample X in sample
{ // propagate the inputs forward:
for each hidden or output layer unit
j { $I_{j=}\Sigma_j W_{ij}O_{i\,;}$ //Compute the net input of unit j with respect to the previous layer, i
$O_{j=}1/(_{1+e}{}^{-Ij});\}$
//Compute the output of each unit j
//Backpropagate the errors
for each unit j in the output layer
$Err_{j=}O_j\,(1\text{-}O_j)\,(T_j - O_j)$;//Compute the error for each unit j in the hidden layers,from the last to the first hidden layer
$Err_{j=}O_j\,(1\text{-}O_j)\,\Sigma_k\,Err_k W_{jk\,;}$ //Compute the error with respect to the next higher layer,k
for each weight Wij in network
{ $\Delta W_{ij}=(l)Err\,_jO_i$
//Weight increment
$W_{ij}=W_{ij}+\Delta W_{ij}$;}//weight update } }

The weights in the network are initialized to small random numbers (e.g., ranging from –0.0 to 1.0).Then propagate the inputs forward, the input and output of each unit in the hidden and output layers are computed. First, the training sample is fed to the input layer of the network. For unit j in the input layer, its output is equal to its input, that is, $O_j= I_j$ for input unit j. The net input to each unit in the hidden and output layers is computed as a linear combination of its inputs. The inputs to the unit are, in fact, the outputs of the units connected to it in the previous layer. To compute the net
input to the unit, each input connected to the unit is multiplied by its corresponding weight, and this is summed. Given a unit j in a hidden or output layer, the net input, I_j, to unit j is $I_j= \Sigma_i W_{ij} O_i$, where w_{ij} is the weight of the connection from unit i in the previous layer to unit j; O_i is the output of unit I from the previous layer. Each This expert can then
unit in the hidden and output layers takes its net input and then applies
an activation function to it. The function symbolizes the activation of the neuron represented by the unit. The logistic or sigmoid function is used. Given the net input Ij to unit j, then O_j, the output of unit j,is computed as $O_j=1/(1+e^{-Ij})$.This function is also referred to as a squashing function[8], since it maps a large input domain onto the smaller range of 0 to 1. The error is then propagated backwards by updating the weights to reflect the error of the network's prediction. For a unit j in the output layer, the error Err_j is computed by $Err_j=O_j(1-O_j)(T_j-O_j)$, where O_j is the actual output of unit j, and T_j is the true output, based on the known class label of the given training sample. $O_j(1-O_j)$ is the derivative of the logistic function. To compute the error of a hidden layer unit j, the weighted sum of the errors of the units connected to unit j in the next layer is considered. The error of a hidden layer unit j is $Err_j = O_j (1-O_j) \Sigma_k Err_k W_{jk}$,where W_{jk} is the weight of the connection from unit j to a unit k in the next higher layer, and Err_k is the error of unit k. The weights are updated to reflect the propagated errors. Weights are updated by the following equations, where Δw_{ij} is the change in weight w_{ij}: $\Delta w_{ij}=(l)\, Err_jO_i$
$w_{i\,j}= w_{ij}+ \Delta w_{ij}$.
The variable l is the learning rate ,a constant typically having a value between 0.0 and 1.0.
Training stops when

- all Δw_{ij} in the previous epoch were so small as to below some specified threshold, or
- the percentage of samples misclassified in the previous epoch is below some threshold, or
- a prespecified number of epochs has expired.

3.3. Backpropagation Neural Network Organizations

Back propagation networks are the most commonly used network because they offer good generalization abilities and are relatively straightforward to implement. Although it may be difficult to determine the optimal network configuration and network parameters. The architecture of neural network used is as given below:-
1). Input layer with 2 nodes
2). One hidden layer with 2 nodes
3). Output layer with one node.

4. Genetics Algorithm.

4.1 An overview

A genetic algorithm is an iterative procedure maintaining a population of structures that are candidate solutions to specific domain challenges [9]. During each temporal increment (called a generation), the structures in the current population are rated for their effectiveness as mutation. Genetic Algorithms (GAs) are search algorithms based on the mechanics of the natural selection process (biological evolution). The most basic concept is that the strong tend to adapt and survive while the weak tend to die out. That is, optimization is based on evolution, and the "Survival of the fittest" concept. GAs has the ability to create an initial population of feasible solutions, and then recombine them in a way to guide their search to only the most promising areas of the state space. Each feasible solution is encoded as a chromosome (string) also called a genotype, and each chromosome is given a measure of fitness via a fitness (evaluation or objective) function. The fitness of a chromosome determines its ability to survive and produce offspring. A finite population of chromosomes is maintained.

GAs use probabilistic rules to evolve a population from one generation to the next. The generations of the new solutions are developed by genetic recombination operators

Biased Reproduction: selecting the fittest to reproduce

Crossover: combining parent chromosomes to produce children chromosomes

Mutation: altering some genes in a chromosome. Crossover combines the "fittest" chromosomes and passes superior genes to the next generation. Mutation ensures the entire state-space will be searched, (given enough time) and can lead the population out of a local minima. Determining the size of the population is a crucial factor. Choosing a population size too small increases the risk of converging prematurely to a local minimum, since the population does not have enough genetic material to sufficiently cover the problem space. A larger population has a greater chance of finding the global optimum at the expense of more CPU time. The population size remains constant from generation to generation. Fitness Function Drives the Population toward better solutions and is the most important part of the algorithm.

4.2. Genetic Algorithm Based BPN Network Training.

Step1: Randomly generate an initial population of, say, P strings of length d: $S(0)=\{\mathbf{s}_1, ..., \mathbf{s}_P\}. \subset \Omega$.

Step 2: Compute the fitness score $\mathbf{f}(\mathbf{s}_k)$ of each individual string $\mathbf{s}_k$ of the current population S(t).

Step 3: Generate an intermediate population [termed mating pool] by applying the *selection* operator.

Step 4: Generate S(t+1) by applying recombination operators (*crossover* and *mutation*) to the intermediate population.

domain solutions, and on the basis of these evaluations, a new population of candidate solutions is formed using specific genetic operators such as reproduction, crossover, and

Step 5: t:=t+1 and continue with Step 2 until some stopping criterion applies [in this case

designate the best-so-far individual as the result of the GA].

The first step generates an initial population S(0), i.e. $S(0)=\{\mathbf{s}_1, ..., \mathbf{s}_P\}\subset \Omega$.. In GA each member of S(0) is a string of length d that corresponds to the problem coding. S(0) is usually generated randomly, because it is not known a priori where the globally optimal strings in Ω. are likely to be found. From this initial population, subsequent populations S(1), ..., S(t), ... will be computed by employing the three genetic operators of selection (reproduction), crossover and mutation. After calculating the relative fitness for all the strings in the current population S(t) (Step 2), selection is carried out and then strings in the current population are copied (i.e. duplicated) and placed in the intermediate population proportional to their fitness relative to other individuals in the population. After selection has been carried out the construction of the intermediate population is completed. Then crossover and mutation are applied to the intermediate population to create the next population S(t+1) (Step 4). Crossover and mutation provide a means of generating new sample points in Ω. while partially preserving distribution of strings across hyperplanes which is observed in the intermediate population. Crossover is a recombination mechanism to explore new regions in Ω. The two new strings, called offspring, are formed by the juxtaposition of the first part of one parent and the last part of the other parent. Continue with Step 2 until some stopping

criterion applies to find final population. Then final weights are determined from it for backpropagation algorithms.

4.3. Genetic Algorithm Based Backpropagation Neural Network Organizations.

Hybrid approach offer strong advantage over either rule based or unaided neural network approaches [10]

Genetic algorithm based back propagation neural network offer good generalization abilities although it may be difficult to determine the optimal network configuration and network parameters. The architecture of GA based neural network used is as follows:

1). Input layer with 2 nodes

2). One hidden layer with 2 nodes

3). Output layer with one node.

5. Experimental Results.

The system has been developed and tested on Windows XP operating system .We have used Visual Basic and Microsoft Access as Front End and Back End Tool. Normalisation is a key part of data pre-processing for neural networks and should enable more accurate predictable rates. Normalised data is used for training backpropagation neural network and Genetic algorithm based backpropagation neural network. We normalize inputs so that input values lies between 0 and 1.Simulation data was sourced from Indian National Stock Exchange (NSE). Input attributes should be carefully selected to keep the dimensionality of input vectors relatively small [11]. As we know close rate and volume are primary quantitative factors for individual equities and from quantitative factors the key qualitative factor of the market sentiment can be derived. So we used close rate and volume of stocks as our input in backpropagation neural network and genetic algorithm based backpropagation neural network and next stock rate as our target for training networks. BPNN and GA-BPNN is trained on data set of Maruti for the years of Jan 2004 to Dec 2006 after training testing is done on data set of 2nd Jan 2007 to 30th Mar 2007 of Maruti stock. GA-BPNN and BPNN performance is compared on stock rates of Maruti for a specified period. The GA's Based BPNN system demonstrated better performance in comparison to BPNN on stock rates of Maruti for a specified period. On the bases of the comparison it has been found that, GA's Based BPNN system is able to predict stock price movement of Maruti correctly 98.31% as shown in Fig 2, while performance of BPNN is 93.22. as shown in Fig 1

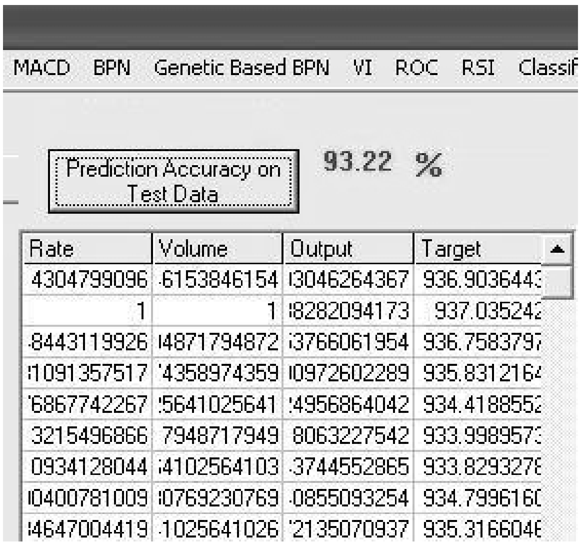

Fig.1: Prediction accuracy using BPNN.

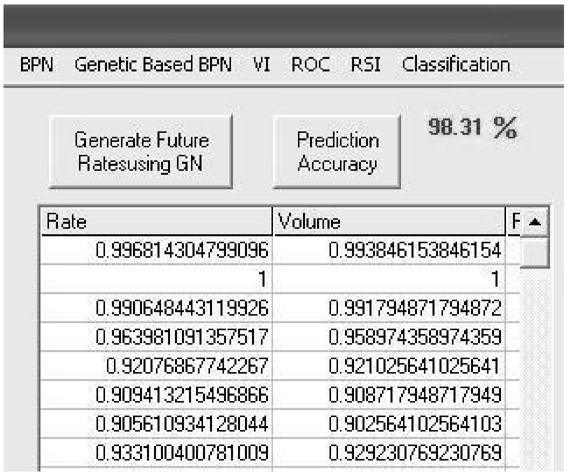

Fig.2:Prediction accuracy using GABPNN

Prediction accuracy of BPNN and GA-BPNN on stock – Maruti for the testing period of 02 Jan 07 to 30 Mar 07 is given below

Backpropagation Neural Network (BPNN)	***Genetic Algorithm Based Backpropagation Neural Network. (GABPNN)***
93.22%	98.31%

6. CONCLUSION

This Paper has compared the forecasting accuracies of backpropagation neural network and. genetic algorithm based backpropagation neural network. Results showed that for this stock, genetic algorithm based backpropagation neural network has given more accurate prediction in comparison to backpropagation neural network.

References.

[1] Mizuno, H., Kosaka, M., Yajima, H. and Komoda N. (1998), "Application of Neural Network to Technical Analysis of Stock Market Prediction", *Studies in Informatic and Control, Vol.7, No.3, pp.111-120.*

[2]. Phillip D. Wasserman, Van Nostrand "Neural Computing: Theory and Practice", *Van Nostrand Reinhold, New York, 1989*

[3] Youngohc yoon and George swales. "Predicting stock price performance: a neural network approach". *IEEE publishing, 1991*.

[4] Shaikh A. Hamid. Primer on using neural networks for forecasting market variables. In *proceedings of the a conference at* school of business, *southern new hampshire university*, *2004.*
[5] Ramon Lawrence "Using Neural Networks to Forecast Stock Market Prices",*Course Project, University of Manitoba Dec. 12, 1997.*
[6] Monica Adya and Fred Collopy. "How Effective are Neural Networks at Forecasting and Prediction? A Review and Evaluation", *Journal of Forecasting 1998.*
[7] Asif Ullah Khan et al "Stock Rate Prediction Using Back Propagation Algorithm: Analyzing the prediction accuracy with different number of hidden layers", *Glow gift, Bhopal*, 2005
[8] Haykin, Simon, "Neural Networks: A Comprehensive Foundation", *Macmillian CollegePublishing Company, New York, 1994.*
[9] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning." *New York: Addison-Wesley, 1989*.
[10] K. Bergerson and D. Wunsch. "A commodity trading model based on a neural network- expert system hybrid", *IJCNN-91- Seattle International Joint Conference on Volume i, Issue , 8-14 Jul 1991 Page(s):289 - 293 vol.1*
[11] H. White. "Economic prediction using neural networks: The case of IBM daily stock returns". In *Neural Networks in Finance nd Investing*, chapter18, pages 315–328. *Probus Publishing Company, 1993.*

Asif Ullah Khan received the B.E. degree in Electronics And Instrumentation Engineering in 1990 from Samrat Ashok Technological Institute , Vidisha. He was the Gold Medalist in M.Tech Degree in Computer Technology and Application from Rajiv Gandhi Technological University in 2005. He is having more than 18 years of industrial and academics experience. Presently he is Dean (Academics) in All Saints' College of Technology, Bhopal, (M.P.), India.