دومین کنگره مشترک سیستمهای فازی و هوشمند ایران
2nd Joint Congress on Fuzzy and Intelligent Systems
28-30 October 2008 ۷ الی ۹ آبان ماه ۱۳۸۷

Iran's Stock Market Prediction By Neural Networks and GA

Mahmood Khatibi
MS. in Control Engineering
mahmood.khatibi@gmail.com

Habib Rajabi Mashhadi
Associate Professor
h_mashhadi@ferdowsi.um.ac.ir

Electrical Eng. Dep. / Faculty of Eng. / Ferdowsi University of Mashhad

Abstract: *Stock market prediction is one of the areas that had been very interesting for investors, economists and managers. For this purpose, classical and modern methods such as AR and ARIMA models, Neural Networks, GA, Fuzzy Logic, etc, have been proposed but among them NNs play an essential role. In this paper, the ability of three different neural networks, namely MLP, RBF and GRNN, are compared for stock market prediction. Unknown parameters of each network are optimized for minimum error by GA in training phase. Then trained networks are used for prediction of two and three monthly returns. In addition, for the first time in the literatures, the optimum order for each model, i.e. the number of input variables for each NN model is determined using trial and error.*

Keywords: GA, Prediction, Neural Networks, Returns, Stock Market.

1 Introduction

Financial or stock market prediction is one of the areas that had been very interesting for investors, economists and managers. The primary focus is to determine the market timing to buy, hold or sell.

Unfortunately, stock market prediction is not an easy task, due to the fact that stock market is essentially nonlinear, dynamic, nonparametric, complicated, and chaotic in the nature. The difficulty of the task has raised questions on whether the stock market price can be predicted. Efficient Market Hypothesis (EMH) or Random-Walk Hypothesis [1], states that market price is unpredictable. It is claimed that future changes in stock market prices cannot be predicted from information of the past prices. Viewing from the other hand, financial time series are shown to exhibit nonrandom behavior. The financial market is not always efficient and this suggests that, there is some predictability in financial time series, at least for weak form of EMH. For this purpose several tools have been used such as AR[1] and ARIMA[2] models which are based on Regression and Least Mean Square Error [2]. These methods are regarded as classical methods. However there are modern methods which use Artificial Intelligence methods such as Neural Networks, GA, Fuzzy Logic and etc, among them NNs play an essential role.

Among all NNs, Perceptrons are used widely for prediction [3] but training of them is time consuming and often has difficulty of local minimums [4], so the use of other NNs such as RBF networks would be noticeable.

One of the most valuable stock's data is its return. Return for a stock in period T is defined as follow:

$$r_i = \frac{p_{L1} - p_{L0} + div_{L0}}{p_{L0}} \tag{1}$$

where: p_{L0} is price of the stock in the beginning of period T, p_{L1} is price of the stock in the end of period T and div_{L0} is divided earning per share over the period T.

In this paper we want to predict stock return of a company in Iran's stock market. For this purpose we chose Iran Khodro Co. since it is a great and

[1]. Auto Regressive
[2]. Auto Regressive Integrated Moving Average

old company which its data are available for a long period.

Then three neural networks type, namely MLP, RBF and GRNN, are trained with all monthly return data of a period except two or three months of its end. Unknown parameters of each network are optimized for minimum error by GA in training phase. Finally these trained optimized networks are tested for prediction of two or three monthly returns which didn't present on the training set.

This paper is organized as follow: section 2 is brief review of the literature, section 3 is problem statement, section 4 describes MLP neural network and its results when applied to the problem, section 5 describes RBF neural network and its results when applied to the problem and finally section 6 describes GRNN neural network and its results when applied to the problem.

2 Literature Review

We can categorize almost all papers about stock market prediction in three main groups:

1. Papers that are trying to predict an index of a stock, based on previous values of it. The index usually is stock's price or stock's return. In this kind of analysis authors try to find a function like F so that:

$$\tilde{r}[n] = F(r[n-1]), r[n-2], \ldots, r[n-p]) \quad (2)$$

$$\varepsilon_n = r[n] - \tilde{r}[n] \quad (3)$$

where $r[n]$ denotes the value of index in time n, ε_n is the error of prediction and p is order of the model. A good function must lead to a less value of variance for ε_n.

For this purpose several methods have been used such as AR model, ARIMA model [5-6], NN models [2-3], etc.

2. Another group of papers are trying to classify stocks, e.g. proper for purchase or no proper [7-8]. These two kinds of analysis also are called Technical analysis [9].

3. Another group of papers are trying to find a relationship between the value of an index and financial ratios, exchange rate, oil price, etc [4]. This kind of analysis also is called Fundamental analysis [9].

In Iran's Stock market case, several researches have been done. In [10], authors employed a neuro-fuzzy approach to forecast stock price and compared their results with ARIMA model. Their research shows that neuro-fuzzy works better than ARIMA model and has such properties as fast convergence, high precision and strong function approximation ability and it is suitable for real stock price forecasting. In [11] authors have examined efficiency of Iran stock market by using AR and Perceptron models. Their research shows AR model can't predict Iran's stock market accurately and so it is efficient but by using Perceptron model, a good function can be fitted to the data history and so it is inefficient. Other researches show Iran's stock market has weak efficiency.

3 Problem Statement

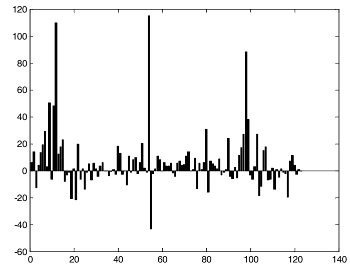


Figure 1: Stock Return of Iran Khodro Co. (1995-2005)

Fig. 1 shows the diagram of monthly stock return of Iran Khodro Co. through 10 years (1995-2005). This is the question:

"How accurate stock's returns of the L months (N, N-1, ... , N-L+1) can be predicted when we know stock's returns of previous months?" .

This question yields to finding function F in (2) by our data of previous months and analyzing its error for predicted returns in the duration of L months: (N, N-1, ... , N-L+1).

In this paper, function F is estimated for different p by three Neural Network types: Perceptron, RBF and GRNN. Each network is trained by data of previous months and then its error for predicted returns in the duration of L months is analyzed. Error in duration L months is defined as:

$$Er_F(L) = \frac{\sqrt{\sum_{i=1}^{L} \varepsilon_F^2 (N+i-L)}}{L} \quad (4)$$

Optimum parameters for three NNs, are obtained by GA in the training phase. The problem is studied for L=3 and L=2. For each L, two different situations are tested and the optimum model's order for prediction is determined using trial and error. So there are four cases:

Case1: In this case L=3 and three last monthly returns are: -10, 14.83 and 17.66.

Case2: In this case L=3 and three last monthly returns are : 4.21, -3.06, 0.75.

Case3: In this case L=2 and three last monthly returns are : 14.83, 17.66.

Case4: In this case L=2 and three last monthly returns are : -3.06, 0.75.

First of all, for neglecting spouts, every return more than 20 is assumed 20 and every return less than -10 is assumed -10. Then the data are normalized in the range -1 to 1.

4 Perceptron case and its results

In this section, we give a brief presentation of multi-layer Perceptron, which is the most popular and widely-used network paradigm in many applications including forecasting. An MLP is typically composed of several layers of nodes.

The first or the lowest layer is an input layer where external information is received. The last or the highest layer is an output layer where the problem solution is obtained. The input and output layers are separated by one or more intermediate layers called *hidden layers*. The nodes in adjacent layers are usually fully connected from a lower layer to a higher layer. Fig.2 gives an example of a fully connected MLP with one hidden layer.

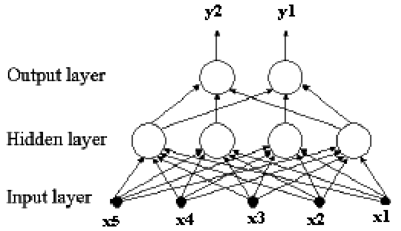
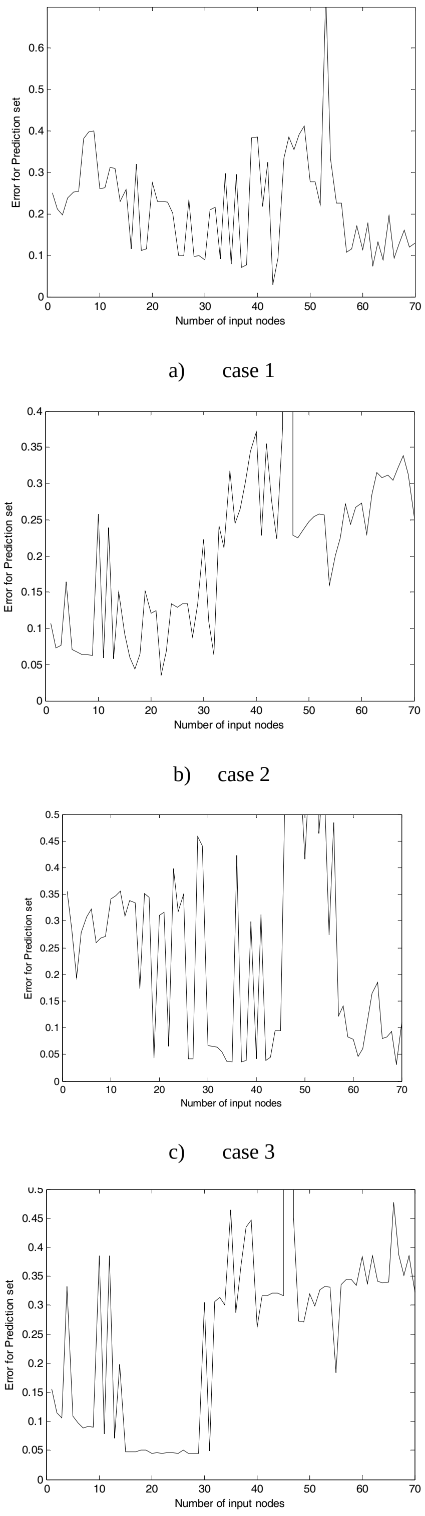


Figure 2: An MLP with one hidden layer

Input layer is only a fan-out layer and does not any processing. Output of every neuron in hidden and output layer is obtained by (5):

$$y_j = f(\sum_i w_{ij}.x_{ij} + \mathrm{b}_{\mathrm{j}}) \quad (5)$$

where x_{ij} is *i*-th input of *j*-th neuron and w_{ij} is its associated weight and b_{j} is the bias of *j*-th neuron. Training process includes determination of w_{ij} s and b_{j} s. Usually BP[3] algorithm or a modified version of it, is used for determination of weights and biases. Although the algorithm is fast



a) case 1

b) case 2

c) case 3

d) case 4

Figure 3: MLP Results

[3] . Back Propagation

but often falls in a local minimum. So using GA for determining weights is notice of worth. In this work a Perceptron with one hidden layer is used and weights and biases are determined for minimum error by GA.

The number of cells in the hidden layer is determined so that the number of training patterns is ten times greater than the number of weights. Moreover some good members which are obtained by BP are posed in initial population.

Fig.3 shows the results of the MLP for the 4 cases:

It is seen a Perceptron with "p"=22 has good performance in the four cases. In the case 1 (when the 3 last monthly returns are -10, 14.83, 17.66), the network has average error equal to 6.9. In the case 2 (when 3 last monthly returns are 4.21, -3.06, 0.75), it has average error equal to 1.05. In the case 3 (when the 2 last monthly returns are 14.83, 17.66), the network has average error equal to 1.95 and finally in case 4 (when 2 last monthly returns are -3.06, 0.75) it has average error equal to 1.35 .

5 RBF case and its results

RBF is a three layer NN [5]. Input layer is only a fan-out layer and does not any processing. Second layer or hidden layer do a nonlinear mapping from input space to often a higher dimension space and output layer do a simple weighted summation and then transfer result from a linear function. For classification a hard limiting function is required in output.

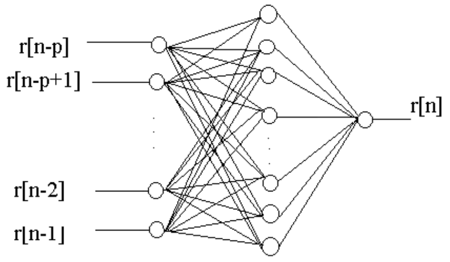
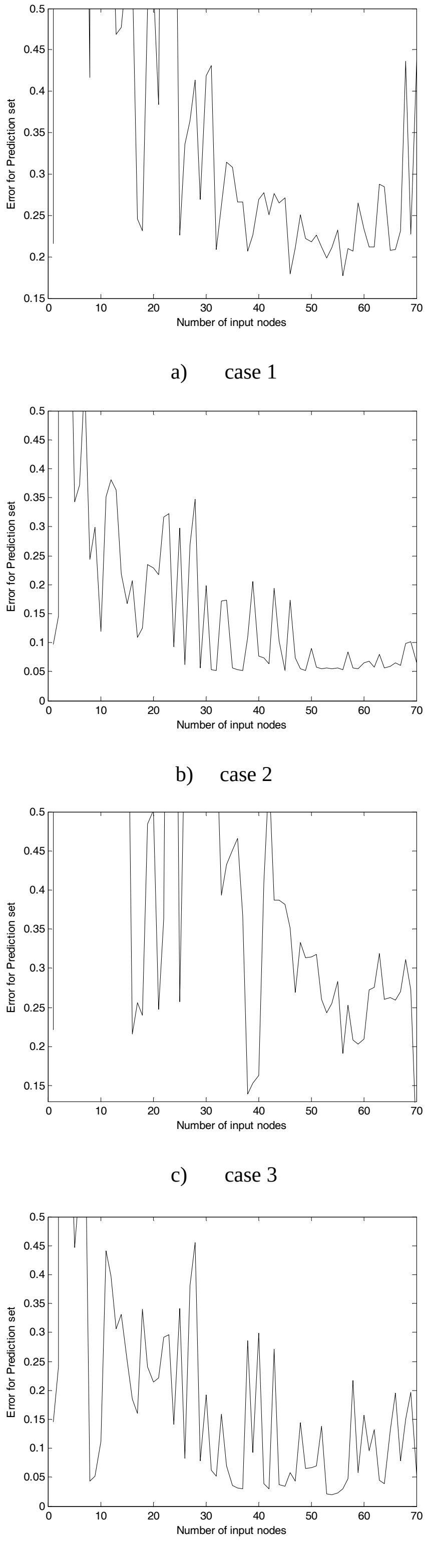


Figure 4: RBF Structure for the problem

The main difference between RBF and Perceptron is in hidden layer processing. The idea in RBF is that every neuron on hidden layer is a Gaussian function with a center and a variance:

$$y_i = \sum_{j=1}^{N} \exp(-\frac{\left|x_j - x_{0i}\right|^2}{2\sigma_i^2}) \qquad (6)$$

The centers x_{0i} are training data or centers of them and so there is not special training process. Variance σ_i^2 can be obtained by several methods. One suggestion is (7):



a) case 1

b) case 2

c) case 3

d) case 4

Figure 5: RBF Results

$$\sigma_i{}^2 = \frac{1}{P}\sum_{k=1}^{P}(d_{ik})^2 \qquad (7)$$

where d_{ik} is distance between x_k and x_{0i}.

The output layer is just as output layer of Perceptrons. **A**n RBF structure like Fig.4 is used to predict stock return. In this Fig. "p" is the order of model. For different "p", the corresponding network is trained for n=p+1, ..., N-L where N is the number of all members of training set and L is the length of the prediction period. For this purpose the first-layer weights is set to P where P is input vector of training set and the first-layer biases are all set to $0.8326/\sigma$, resulting in Gaussian functions that cross 0.5 at weighted inputs of $\pm\sigma$. The second-layer weights IW{2,1} and biases b{2} are found by simulating the first-layer outputs A{1} and then solving the following linear expression:

$$\left[W\{2,1\} \quad b\{2\}\right] * \left[A\{1\}; ones\right] = T \qquad (8)$$

T is the output vector of training set.

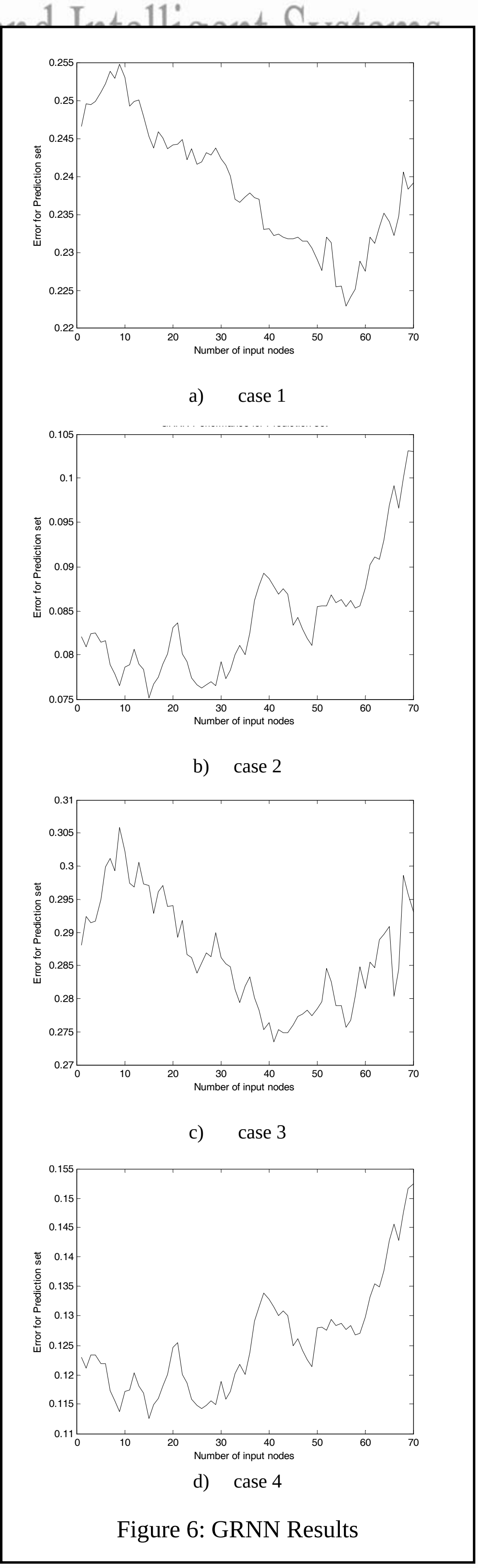
For easiness, it is presumed a constant σ for all Gaussian functions and it is optimized by GA for minimum error for every "p". Assuming a constant σ, improve the speed of optimization significantly while don’t has significant effect on results as will seen. Also it is very useful to impose constriction $\sigma > 1$ to the problem for avoiding overtraining. The optimized network is tested for L month of the end that didn’t present in training set. Fig.6 shows the results of the RBF for the 4 cases:

It is seen an RBF with "p"=56 has good performance in the four cases. In case 1 (when the 3 last monthly returns are -10, 14.83, 17.66), the network has average error equal to 5.4. In case 2 (when 3 last monthly returns are 4.21, -3.06, 0.75), it has average error equal to 1.5. In case 3 (when the 2 last monthly returns are 14.83, 17.66), the network has average error equal to 5.4 and finally in case 4 (when 2 last monthly returns are -3.06, 0.75) it has average error equal to 0.9 .

These results are comparable with MLP results when we remember training of RBF is less time consuming than MLP training.

6 GRNN case and its Results

A GRNN network is very similar to RBF *NN*. The difference is in the output layer. The weights of output layer are corresponding outputs divided to a coefficient. Mathematically:



a) case 1

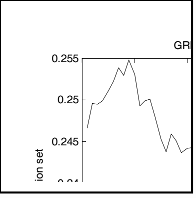
b) case 2

c) case 3

d) case 4

Figure 6: GRNN Results

$$y_i = \frac{\sum_{j=1}^{N} y_j \exp\left(-\frac{(x_i - x_{0j})^2}{2\sigma_j{}^2}\right)}{\sum_{j=1}^{N} \exp\left(-\frac{(x_i - x_{0j})^2}{2\sigma_j{}^2}\right)} \qquad (9)$$



A GRNN structure is used to predict stock return. For different "p" the corresponding Network is trained for n=p+1, ... , N-L where N is the number of all members of training set and L is the length of prediction. For this purpose the first-layer weights are set to P^T where P is input vector of training set and the first-layer biases are all set to $0.8326/\sigma$, resulting in Gaussian functions that cross 0.5 at weighted inputs of $\pm\sigma$. The second-layer weights are set to T where T is output vector of training set. Again for easiness and speed, it is presumed a constant σ for all Gaussian functions and it is optimized by GA for minimum error for each "p". Also for avoiding overtraining it is very useful to impose constriction $\sigma > 1$ to the problem. The optimized network is tested for L month of the end that didn’t present in training set. Fig. 7 shows the results of the GRNN for the 4 cases:

It is seen a GRNN with "p"=49 has good performance in the four cases. In case 1 (when the 3 last monthly returns are -10, 14.83, 17.66), the network has average error equal to 6.9. In case 2 (when 3 last monthly returns are 4.21, -3.06, 0.75), it has average error equal to 2.4. In case 3 (when the 2 last monthly returns are 14.83, 17.66), the network has average error equal to 8.3 and finally in case 4 (when 2 last monthly returns are -3.06, 0.75) it has average error equal to 3.6. It is obvious that these results are not comparable with MLP and RBF results.

Table 1 summarizes the results:

Table 1: Summarized Results

NN	Best Deg	Error for Prediction set			
		Case1	Case2	Case3	Case4
MLP	22	6.9	1.05	1.95	1.35
RBF	56	5.4	1.5	5.4	0.9
GRNN	49	6.9	2.4	8.3	3.6

Conclusion

In this paper, stock market return of Iran Khodro Co. has been predicted. For this purpose, monthly returns of 10 years are collected as training set. Then three neural networks namely MLP, RBF and GRNN are compared for stock market prediction.

The parameters of each network have been optimized in order to minimize the prediction error using GA. Then trained networks have been used for prediction of two and three monthly returns which didn’t present on the training set. Furthermore the optimum order for each model, has been determined. Results show combination of RBF or MLP neural networks with GA, can predict the complex time series in a sufficient manner.

References

[1] Hellström, T., Holmström, K. Predicting the stock market. Technical Report Ima-TOM-1997-07, Center of Mathematical Modeling, Department of Mathematics and Physics, Mälardalen University,Västeras, Sweden.

[2] Guoqiang Zhang, B. Eddy Patuwo, Michael Y. Hu, Forecasting with artificial neural networks:The state of the art, International Journal of Forecasting ,vol.14, pp. 35–62, 1998.

[3] A. Vellidoa, P.J.G. Lisboaa, J. Vaughanb, Neural networks in business: a survey of applications (1992–1998), Expert Systems with Applications, Vol. 17, pp. 51–70, 1999.

[4] Dennis Olson , Charles Mossman, Neural network forecasts of Canadian stock returns using accounting ratios, International Journal of Forecasting, Vol. 19, pp. 453–465, 2003.

[5] Pai, P.F., Lin, C.S., A hybrid ARIMA and support vector machines model in stock price forecasting, Omega, Vol. 33, 2005, P.P. 497 – 505.

[6] Wang,J.-H.&Leu,J.-Y.(1996). Stock market Trend Prediction using ARIMA based neural networks. IEEE international Conference on Neural networks, 2160-2165.

[7] Leung,M.T., Daouk, H.,& Chen,A.-S.(2001).Forecasting stock indices: A comparison of classification and level estimation models. *International Journal of Forecasting*, 16,173-190.

[8] Swales, G.S.Jr., Yoon, Y., *Applying Artificial Neural Networks to Investment Analysis*, Financial Analysts Journal, September-October, 1992, pp. 78-80.

[9] Mohammed Omran, Linear Versus Non-linear relationships Between Financial Ratios and Stock Returns: Empirical Evidence from Egyptian Firms, Review of Accounting and Finance, Vol. 3, No 2, pp. 84-102, 2004.

[10] A.Afsar, A.Azar, Strengthening Neural networks for prediction stock market prices by fuzzy logic systems (In Persian).

[11] Patterning process of price's behavior in Tehran stock market, comparing nonlinear and linear models, Oct. 1998, (In Persian).