Journal of Economics and International Finance Vol. 3(4), pp. 204-210, April 2011
Available online at http://www.academicjournals.org/JEIF
ISSN 2006-9812 ©2011 Academic Journals

Full Length Research Paper

Recognition of good prediction of gold price between MLFF and GMDH neural network

Vida Varahrami

Kargar-e-Shomali Avenue, Faculty of Economics, University of Tehran, Tehran, Iran. E-mail: Vida7892000@yahoo.com, vvarahrami@ut.ac.ir.

Accepted 28 February, 2011

Studies show neural networks have better results in predicting of financial time series in comparison to any linear or non-linear functional form to model the price movement. Neural networks have the advantage of simulating the non-linear models when little a priori knowledge of the structure of problem domains exist or the number of immeasurable input variables are great and system has a chaotic characteristics. Among different methods, MLFF neural network with back-propagation learning algorithm and GMDH neural network with genetic learning algorithms are used to predict gold price of the NYMEX database covering 1st January, 2002 to 13th July, 2007 period. This paper uses moving average crossover inputs and the results confirms (1) the fact that there is short-term dependence in gold price movements, (2) the EMA moving average has better result and also (3) by means of the GMDH neural networks, prediction accuracy in comparison to MLFF neural networks, can be improved.

Key words: Artificial neural networks (ANN), multi layered feed forward (MLFF), group method of data handling (GMDH), simple moving average (SMA), exponential moving average (EMA), gold price.

INTRODUCTION

Neural networks are a class of generalized nonlinear models inspired by studies of the human brain. Their main advantage is that they can approximate any non-linear function to an arbitrary degree of accuracy with a suitable number of hidden units. Neural networks get their intelligence from learning process, and then this intelligence makes them have the capability of auto-adaptability, association and memory to perform certain tasks.

Problems of complex objects modelling such as analysis and prediction of stock market, gold price and other, cannot be solved by deductive logical-mathematical methods with needed accuracy. The task of knowledge extraction from data is to select mathematical description from data. But the required knowledge for designing of mathematical models or architecture of neural networks is not at the command of the users [Mirmirani and Li, 2004]. In mathematical statistics, there is need to have a priori information about the structure of the mathematical model. In neural networks, the user estimates this structure by choosing the number of layers, and the number and transfer functions of nodes of a neural network. This requires not only knowledge about the theory of neural networks, but also knowledge of the object's nature and time. Besides this, the knowledge from systems theory about the systems modelled is not applicable without transformation into an object in the neural network world. But the rules of translation are usually unknown (Ivakhnenko and Müller, 1995).

GMDH type neural networks can overcome these problems. It can pick out knowledge about object directly from data sampling. The GMDH is the inductive sorting-out method, which has advantages in the cases of rather complex objects, having no definite theory, particularly for the objects with fuzzy characteristics. GMDH algorithms found the only optimal model using full sorting-out of model-candidates and operation of evaluation of them, by external criteria of accuracy or difference types (Madala. and Ivakhnenko, 1994; Ivakhnenko and Müller, 1995).

In this paper, a method of non-parametric approach to predict the gold price over time in different condition of market is used. MLFF neural network with back-propagation algorithm and GMDH neural network with genetic algorithms are briefly discussed, and they are used to make appropriate model for predicting the gold price of the NYMEX database. As input variables to the neural networks, we use price time series separately with 2 lags of the 5, 10, 15, 30, 35, 50 and 55-day moving average crossover. These two models are coded and implemented in Matlab software package.

The prime aim here is to find out dependence in gold price and also which neural network model does best in forecasting when the input parameters are little or great.

GOLD PRICE PREDICTION - A LITERATURE SURVEY

Concerning gold price movements, Booth et al. (1991) revealed the existence of long-term dependence, while Ball et al. (Mirmirani and Li, 2004), on the other hand, detected the existence of short-term periodic variation. Using a GARCH model, Akgiray et al. (1991) also verified time dependency of gold price. The use of gold price as an economic indicator drew the attention of many economists. Mirmirani [2004] asserted that more than 70% of the change in inflation rate can be explained by the price movements of gold.

In the field of engineering decision systems, techniques based on inductive human behavior have been developed. Among such systems are neural networks, the genetic algorithm and their integration [Mirmirani and Li, 2004]. These engineering-based systems have gradually found their way into business and economics [Dawid, 1996]. A thorough literature review of neural network applications in finance and business are provided by Wong and Yakup [1998].

In time series analysis, a review of the methodological linkage between statistical techniques and neural networks is given by Cheng and Titterington (1994). In comparison with statistical techniques, neural networks make less restrictive assumptions on the underlying distributions and provide a higher degree of robustness (Brauner et al., 1997). As pointed out by Kuo and Reitsch (Ripley, 1993), neural networks provide meaningful predictions when independent variables were correlated or missing. It is also known that neural networks tended to outperform the conventional regression analysis at the presence of ambiguity in independent variables. It is not surprising to learn that neural networks are superior to traditional approaches in terms of parsimony of parameterization. In addition, a network structure is trained by using part of the data and then tested by using the rest of the data. A well-trained network is therefore expected to provide robust predictions (Cheng and Titterington, 1994). Ongsritrakul and Soonthornphisaj (2003) demonstrate the use of SVR (support vector regression) techniques for predicting the cost of gold by using factors that have an effect on gold to estimate its price. They have applied a decision tree algorithm for the feature selection task and then performed the regression process using forecasted indexes. Their experimental results show that the combination of the decision tree and SVR leads to a better performance (Ongsritrakul and Soonthornphisaj, 2003).

Recently, some studies have been done with neural networks. One of them has been done with Mirmirani and LI (2004) who concentrates on a non-parametric study of gold prediction with neural network and genetic algorithm. Sarafraz and Afsar (2005) have done another study with neuro-fuzzy networks. They have studied gold price in Iran.

MODELING USING NEURAL NETWORK

Artificial neural networks (ANN) are biologically inspired network based on the organization of neurons and decision making process in the human brain (Madala. and Ivakhnenko, 1994). In other words, it is the mathematical analogue of the human nervous system. This can be used for prediction, pattern recognition and pattern classification purposes. It has been proved by several authors that ANN can be of great used when the associated system is so complex that the underline processes or relationship are not completely understandable or display chaotic properties (Priestley, 1988). Development of ANN model for any system involves three important issues: (1) Topology of the network, (2) A proper training algorithm and (3) Transfer function. Basically an ANN involves an input layer and an output layer connected through one or more hidden layers. The network learns by adjusting the inter connections between the layers. When the learning or training procedure is completed, a suitable output is produced at the output layer. The learning procedure may be supervised or unsupervised. In prediction problem, supervised learning is adopted where a desired output is assigned to network before hand [Jang and Sun, 1995].

MLFF neural network

MLFF neural network is one the famous and it is used at more than 50% of researches that are doing in financial and economy field recently [Akgiray et al., 1991]. This class of networks consists of multiple layers of computational units, usually interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the subsequent layer. In many applications, the units of these networks apply a sigmoid function as a transfer function ($f(x)=\frac{1}{1+e^{-x}}$). It has a continuous derivative, which allows it to be used in back-propagation. This function is also preferred because its derivative is easily calculated: $y' = y(1-y)$ Multi-layer networks use a variety of learning techniques; the most popular is back-propagation algorithm (BPA). The BPA is a supervised learning algorithm that aims at reducing overall system error to a minimum [Madala. and Ivakhnenko, 1994; Kamarthi and Pittner, 1999]. This algorithm has made multilayer neural networks suitable for various prediction problems. In this learning procedure, an initial weight vectors w_0 is updated according to [Kartalopoulos, 2000]:

$$w_i(k+1) = w_i(k) + \mu(T_i - O_i) f'(w_i x_i) x_i \qquad (1)$$

where, $w_i \Rightarrow$ The weight matrix associated with ith neuron; $x_i \Rightarrow$ Input of the ith neuron; $O_i \Rightarrow$ Actual output of the ith neuron; $T_i \Rightarrow$ Target output of the ith neuron, and μ is the learning rate parameter.

Here the output values (O_i) are compared with the correct answer to compute the value of some predefined error-function. The neural network is learned with the weight update Equation (1) to minimize the mean squared error given by [Kartalopoulos, 2000]:

$$E = \frac{1}{2}(T_i - O_i)^2 = \frac{1}{2}[T_i - f(w_i x_i)]^2$$

By various techniques, the error is then fed back through the network. Using this information, the algorithm adjusts the weights of each connection in order to reduce the value of the error function by some small amount. After repeating this process for a sufficiently large number of training cycles, the network will usually converge to some state where the error of the calculations is small. In this case, one says that the network has learned a certain target function. To adjust weights properly, one applies a general method for non-linear optimization that is called gradient descent. For this, the derivative of the error function with respect to the network weights is calculated and the weights are then changed such that the error decreases (Lippmann, 1987).

The gradient descent back-propagation learning algorithm is based on minimizing the mean square error. An alternate approach to gradient descent is the exponentiated gradient descent algorithm which minimizes the relative entropy (Srinivasan et al., 2002).

GMDH neural network

By means of GMDH algorithm, a model can be represented as sets of neurons in which different pairs of them in each layer are connected through a quadratic polynomial and thus produce new neurons in the next layer. Such representation can be used in modeling to map inputs to outputs [Madala. and Ivakhnenko, 1994; Ivakhnenko and Müller, 1995; Farlow, 1984]. The formal definition of the identification problem is to find a function $\hat{f}$ so that can be approximately used instead of actual one, f , in order to predict output $\hat{y}$ for a given input vector $X = (x_1, x_2, x_3, ..., x_n)$ as close as possible to its actual output y (Atashkari et al., *2007)*. Therefore, given M, observation of multi-input-single-output data pairs so that

$$y_i = f(x_{i1}, x_{i2}, x_{i3}, ..., x_{in}) \qquad (i=1,2,…,M)$$

It is now possible to train a GMDH-type neural network to predict the output values $\hat{y}_i$ for any given input vector $X = (x_{i1}, x_{i2}, x_{i3}, ..., x_{in})$, that is

$$\hat{y}_i = \hat{f}(x_{i1}, x_{i2}, x_{i3}, ..., x_{in}) \qquad (i=1,2,…,M).$$

The problem is now to determine a GMDH-type neural network so that the square of difference between the actual output and the predicted one is minimized, that is

$$\sum_{i=1}^{M} [\hat{f}(x_{i1}, x_{i2}, x_{i3}, ..., x_{in}) - y_i]^2 \rightarrow \min.$$

General connection between inputs and output variables can be expressed by a complicated discrete form of the Volterra functional series in the form of

$$y = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} x_i x_j + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} a_{ijk} x_i x_j x_k + ... \qquad (2)$$

which is known as the Kolmogorov-Gabor polynomial (Madala. and Ivakhnenko, 1994; Ivakhnenko and Müller, 1995, 1995; Farlow, 1984). This full form of mathematical description can be represented by a system of partial quadratic polynomials consisting of only two variables (neurons) in the form of

$$\hat{y} = G(x_i, x_j) = a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j + a_4 x_i^2 + a_5 x_j^2 \qquad (3)$$

In this way, such partial quadratic description is recursively used in a network of connected neurons to build the general mathematical relation of inputs and output variables given in Equation (2). The coefficient a_i in Equation (3) are calculated using regression techniques so that the difference between actual output, y, and the calculated one, $\hat{y}$, for each pair of x_i, x_j as input variables is minimized. Indeed, it can be seen that a tree of polynomials is constructed using the quadratic form given in Equation (3) whose coefficients are obtained in a least-squares sense. In this way, the coefficients of each quadratic function G_i are obtained to optimally fit the output in the whole set of input-output data pair, that is [Ivakhnenko and Muller, 1995]

$$E = \frac{\sum_{i=1}^{M} (y_i - G_i)^2}{M} \rightarrow \min \tag{4}$$

In the basic form of the GMDH algorithm, all the possibilities of two independent variables out of total n input variables are taken in order to construct the regression polynomial in the form of Equation (3) that best fits the dependent observations (y_i , i=1, 2, …, M) in a least-squares sense. Consequently, $\binom{n}{2} = \frac{n(n-1)}{2}$ neurons will be built up in the first hidden layer of the feed forward network from the observations { (y_i, x_{ip}, x_{iq}); (i=1, 2,… M)} for different $p,q \in \{1,2,...,n\}$. In other words, it is now possible to construct M data triples { (y_i, x_{ip}, x_{iq}); (i=1, 2,…, M)} from observation using such $p,q \in \{1,2,...,n\}$ in the form

$$\left[\begin{array}{cc|c} x_{1p} & x_{1q} & y_1 \\ x_{2p} & x_{2q} & y_2 \\ \hline x_{Mp} & x_{Mq} & y_M \end{array}\right].$$

Using the quadratic sub-expression in the form of Equation (3) for each row of M data triples, the following matrix equation can be readily obtained as

$$A\mathbf{a} = Y$$

where $\mathbf{a}$ is the vector of unknown coefficients of the quadratic polynomial in Equation (3)

$$\mathbf{a} = \{a_0, a_1, a_2, a_3, a_4, a_5\} \tag{5}$$

and $Y = \{y_1, y_2, y_3, ..., y_M\}^T$ is the vector of output's value from observation. It can be readily seen that

$$A = \left[\begin{array}{cccccc} 1 & x_{1p} & x_{1q} & x_{1p}x_{1q} & x_{1p}^2 & x_{1q}^2 \\ 1 & x_{2p} & x_{2q} & x_{2p}x_{2q} & x_{2p}^2 & x_{2q}^2 \\ \hline 1 & x_{Mp} & x_{Mq} & x_{Mp}x_{Mq} & x_{Mp}^2 & x_{Mq}^2 \end{array}\right]$$

The least-squares technique from multiple-regression analysis leads to the solution of the normal equations in the form of

$$\mathbf{a} = (A^T A)^{-1} A^T Y \tag{6}$$

which determines the vector of the best coefficients of the quadratic Equation (3) for the whole set of M data triples. It should be noted that this procedure is repeated for each neuron of the next hidden layer according to the connectivity topology of the network. However, such a solution directly from normal equations is rather susceptible to round off errors and, more importantly, to the singularity of these equations (Madala. and Ivakhnenko, 1994; Ivakhnenko and Müller, 1995; Amanifard et al., 2008; Atashkari et al., 2007).

TECHNICAL INDICATORS USED FOR THE STUDY

Simple moving average (SMA)

This is perhaps the oldest and the most widely used technical indicator. It shows the average value of price over time. A simple moving average with the time period n is calculated by:

$$SMA(t) = \frac{1}{n}\sum_{i=0}^{i=n} C_{t-i}$$

where C_t is a price at time t (Vandaele, 1983). The shorter the time period, the more reactionary a moving average becomes. A typical short term moving average ranges from 5 to 25 days, an intermediate-term from 5 to 100, and long-term 100 to 250 days. In our experiment, the window of the time interval n is 5.

Exponential moving average (EMA)

An exponential moving average gives more weight to recent prices, and is calculated by applying a percentage of today's closing price to yesterday's moving average. The equation for n time period is [Vandaele, 1983]:

$$EMA(t) = \frac{2}{n+1}C_t + (1 - \frac{2}{n+1}) * SMA(t)$$

The longer the period of the exponential moving average, the less total weight is applied to the most recent price. The advantage to an exponential average is its ability to pick up on price changes more. In our experiment, the window of the time interval n is 5.

Table 1. MLFF network characteristics

NH	N	TF1	TF2	TF3	TFO
3	15- 8 -10	Logistic	Tanh	Tanh	Linear

NH, Number of hidden layers; N, neurons in hidden layers; TFl, transfer function in hidden layer 1; TF2, transfer function in hidden layer 2; TF3, transfer function in hidden layer, TFO, transfer function in output layer.

Table 2. GMDH network characteristics.

NH	N	TF1	TF2	TFO
2	8-4	Volterra	Volterra	Volterra

NH, Number of hidden layers; N, neurons in hidden layers; TFl, transfer function in hidden layer 1; TF2, transfer function in hidden layer 2; TF3, transfer function in hidden layer, TFO, transfer function in output layer.

Table 3. MLFF and GMDH network results with SMA.

Variable combination	MLFF		GMDH	
	RMSE	MAPE	RMSE	MAPE
MA_5,MA_5(-1),MA_5(-2)	5.782	0.824	4.161	0.398
MA_{10},MA_{10}(-1),MA_{10}(-2)	5.583	0.872	4.172	0.39
MA_{15},MA_{15}(-1),MA_{15}(-2)	6.341	0.889	5.452	0.487
MA_{30},MA_{30}(-1),MA_{30}(-2)	8.125	1.115	6.283	0.963
MA_{35},MA_{35}(-1),MA_{35}(-2)	9.529	1.211	6.743	1.206
MA_{50},MA_{50}(-1),MA_{50}(-2)	13.364	1.349	10.278	1.208
MA_{55},MA_{55}(-1),MA_{55}(-2)	14.461	1.732	12.235	1.292
MA_{60},MA_{60}(-1),MA_{60}(-2)	14.523	2.519	12.227	1.975

RMSE stands for root mean square error, and MAPE for mean Absolute percentage error.

EMPIRICAL RESULTS

As input variables to the neural networks, we use price time series covering 01/01/2002-13/7/2007 period separately, with 2 lags of the 5-day [MA_5,MA_5(-1),MA_5(-2)], 10-day [MA_{10},MA_{10}(-1),MA_{10}(-2)], 15-day [MA_{15},MA_{15}(-1),MA_{15}(-2)], 30-day [MA_{30},MA_{30}(-1),MA_{30}(-2)], 35-day [MA_{35},MA_{35}(-1),MA_{35}(-2)], 50-day [MA_{50},MA_{50}(-1),MA_{50}(-2)], 55-day [MA_{55},MA_{55}(-1),MA_{55}(-2)] and 60-day [MA_{60},MA_{60}(-1),MA_{60}(-2)] moving average crossover. 5, 10 and 15-day simple moving average and EMA are used to measure the short-term dependency, 30, 35 and 40-day moving average for intermediate-term dependency and 50, 55 and 60-day moving average are used to measure the long-term dependency in gold price.
Training set is determined to include about 50% of the data set and the 25% will be used for testing and 25% for validating purposes.

During the designing of MLFF model we have changed the number of layers and neurons. Finally, we found the best network in this research as a network with 3 hidden layers and 15-8-10 neurons. Almost 100 transfer function has been used and sigmoid is used for the input function and the linear function is used for output. Table 1 summarizes the characteristics of the best network.

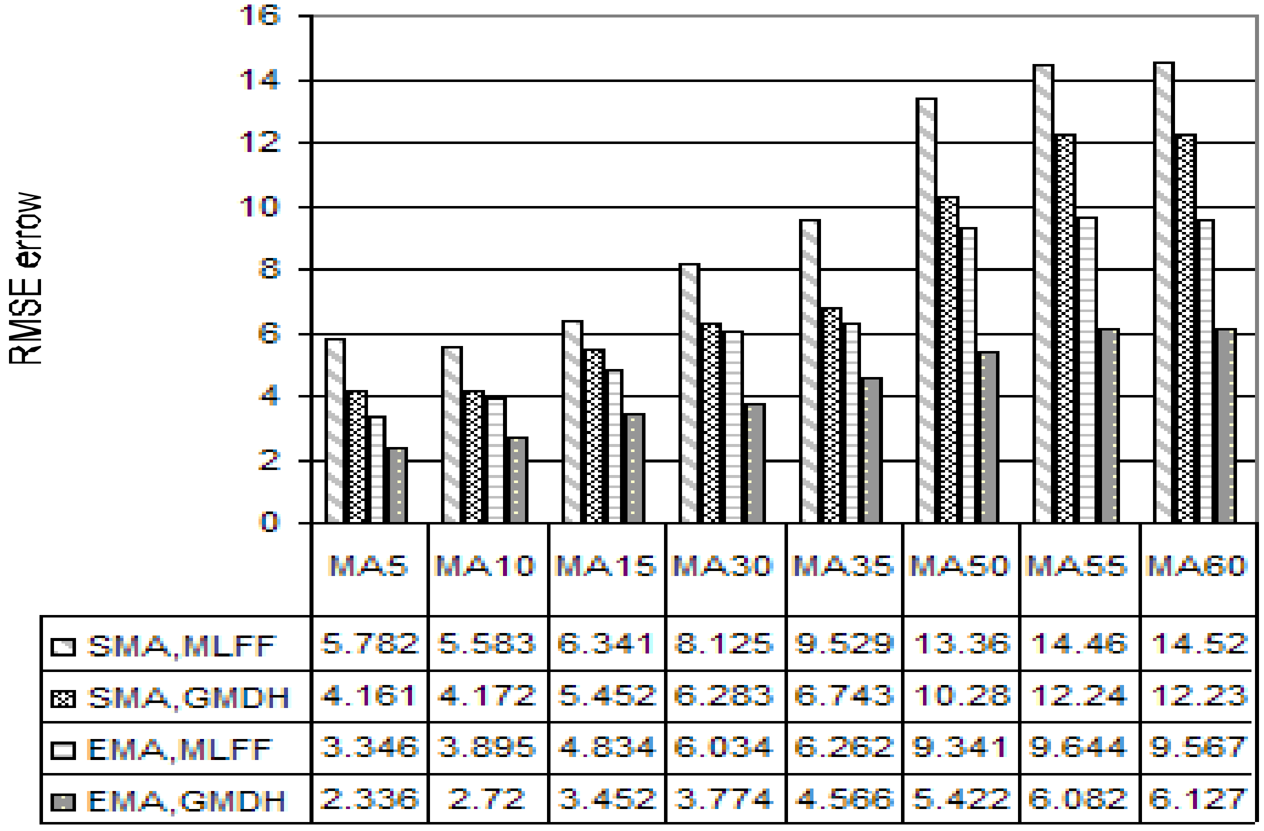
The same way, the GMDH neural network model characteristics is summarized in Table 2.

Table 3 shows error rates on the testing data of the best architecture for each variables combination with simple moving average calculation.

In Figures 1 and 2, RMSE and MAPE rates are shown through a chart. They make the comparison easier.

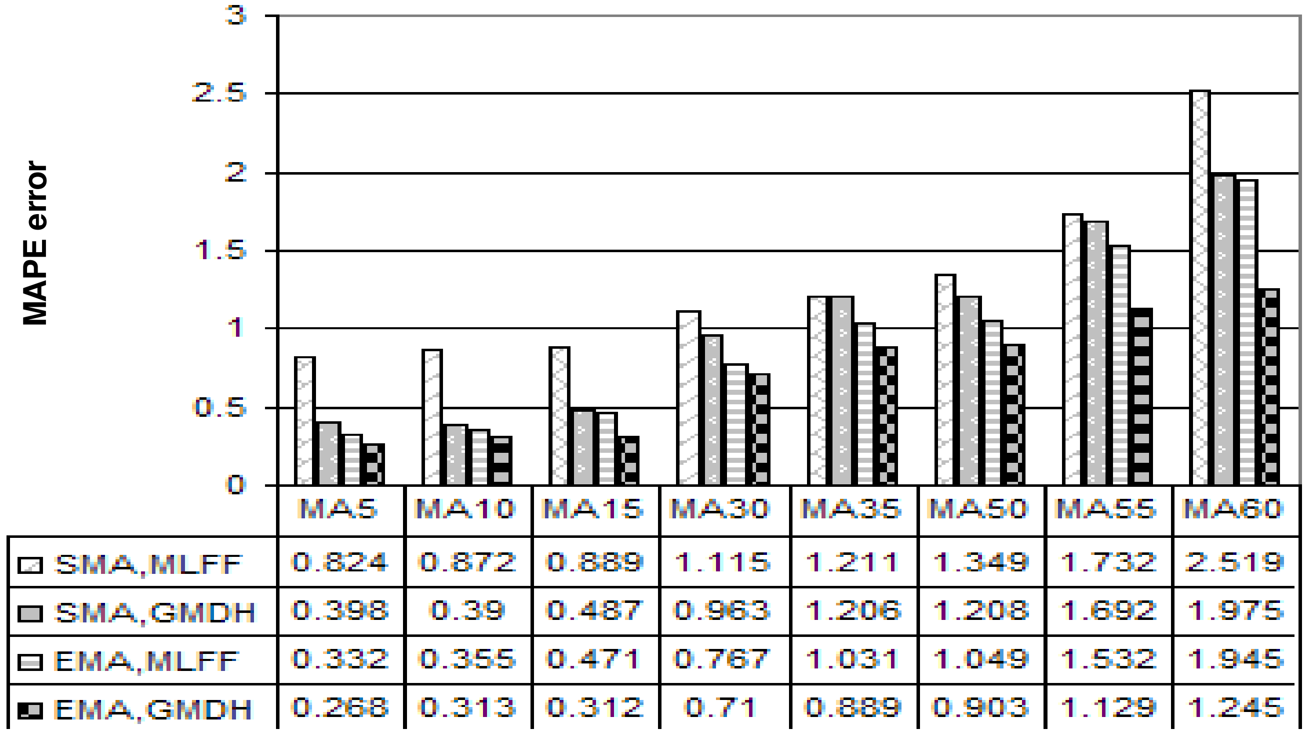
Table 4 shows error rates of the best architecture for each variables combination with exponential moving average calculation.

These tables suggest that the GMDH network with 3 hidden layers is a better architecture. It is also revealed that exponential moving average (EMA) has a better response for the gold price time series in this period. The result of the neural network accuracy measure (RMSE) is noticeably decreased with short-term moving like 5 and 10. This confirms the fact that there is short-term



	MA5	MA10	MA15	MA30	MA35	MA50	MA55	MA60
□ SMA,MLFF	5.782	5.583	6.341	8.125	9.529	13.36	14.46	14.52
▩ SMA,GMDH	4.161	4.172	5.452	6.283	6.743	10.28	12.24	12.23
□ EMA,MLFF	3.346	3.895	4.834	6.034	6.262	9.341	9.644	9.567
▨ EMA,GMDH	2.336	2.72	3.452	3.774	4.566	5.422	6.082	6.127

Figure 1. RMSE of different models and different MAs.



	MA5	MA10	MA15	MA30	MA35	MA50	MA55	MA60
□ SMA,MLFF	0.824	0.872	0.889	1.115	1.211	1.349	1.732	2.519
▨ SMA,GMDH	0.398	0.39	0.487	0.963	1.206	1.208	1.692	1.975
□ EMA,MLFF	0.332	0.355	0.471	0.767	1.031	1.049	1.532	1.945
■ EMA,GMDH	0.268	0.313	0.312	0.71	0.889	0.903	1.129	1.245

Figure 2. MAPE of different models and different MAs.

Table 4. MLFF and GMDH network results with EMA.

Variable combination	MLFF		GMDH	
	RMSE	**MAPE**	**RMSE**	**MAPE**
MA_5,MA_5(-1),MA_5(-2)	3.346	0.332	2.336	0.268
MA_{10},MA_{10}(-1),MA_{10}(-2)	3.895	0.355	2.72	0.313
MA_{15},MA_{15}(-1),MA_{15}(-2)	4.834	0.471	3.452	0.312
MA_{30},MA_{30}(-1),MA_{30}(-2)	6.034	0.767	3.774	0.71
MA_{35},MA_{35}(-1),MA_{35}(-2)	6.262	1.031	4.566	0.889
MA_{50},MA_{50}(-1),MA_{50}(-2)	9.341	1.049	5.422	0.903
MA_{55},MA_{55}(-1),MA_{55}(-2)	9.644	1.532	6.082	1.129
MA_{60},MA_{60}(-1),MA_{60}(-2)	9.567	1.945	6.127	1.245

RMSE stands for root mean square error, and MAPE for mean Absolute percentage error.

dependence in gold price.

This research has been done with a Matlab's neural network toolbox that is designed based on GEovM. We have changed some part of its code to be more flexible and dynamic.

Conclusion

Gold is an important traded commodity and forecasting its price, has important implications not only monetarily but also economically. Despite the significant number of studies on gold and its price, the precise pricing mechanism in the gold market has not been deciphered. Many of them are based on neural network and it shows they are much better than the other methods. This paper used two types of neural networks to investigate the price of gold. It has been shown that GMDH type neural networks have better results when the associated system is so complex and the underline processes or relationship are not completely understandable or display chaotic properties.

We have used gold price time series and its moving average as the input to the neural network and the results show:

1. The GMDH has better response relative to MLFF neural network.
2. Gold price has short-term dependence to its history.
3. The exponential moving average is a better choice relative to simple moving average, if the moving average techniques are used.

REFERENCES

Akgiray V, Booth G, Geoffrey H, John J, Mustafa C (1991). Conditional dependence in precious metal prices. Fin. Rev., 26: 367-386,

Amanifard N, Nariman-Zadeh N, Borji M, Khalkhali A, Habibdoust A, (2008). Modelling and Pareto optimization of heat transfer and flow coefficients in microchannels using GMDH type neural networks and genetic algorithms; 49(2): 311-325.

Atashkari K, Nariman-Zadeh N, Gölcü M, Khalkhali A, Jamali A, (2007). Modelling and multi-objective optimization of a variable valve-timing spark-ignition engine using polynomial neural networks and evolutionary algorithms; Energy Conversion Manage., 48(3): 1029-1041.

Brauner EO, Dayhoff JE, Xiaoyun S, Hormby S (1997). Neural network training techniques for a gold trading model;Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceed. IEEE/IAFE, 23(25): 57 – 63

Cheng B, Titterington DM (1994). Neural networks: A review from a statistical perspective; Statistical Science, 9 (1): 2–30.

Dawid (1996). Herbert; Adaptive Learning by Genetic Algorithms, Springer, Berlin,

Farlow SJ (1984). Self-organizing Method in Modelling: GMDH type algorithm; Marcel Dekker Inc.,

Ivakhnenko AG, Müller JA (1995). Recent Developments of Self-Organising Modeling in Prediction and Analysis of Stock Market. 20(1-2): 29-50.

Ivakhnenko AG, Muller JA (1995). Self-organization of nets of active neurons: System Analysis Modeling and Simulation; (SAMS), 20(1-2): 29-50

Ivakhnenko, AG, Müller JA. (1995). Present state and new problems of further GMDH development; SAMS, p. 20

Jang JR, Sun C (1995). Nero Fuzzy Modelling and Control; Proc. of the IEEE, pp. 378-405,

Kamarthi SV, Pittner S (1999). Accelerating neural network training using weight extrapolation; Neural Network, 12: 1285-1299

Kartalopoulos SV (2000). Understanding Neural Networks and Fuzzy Logic- Basic Concepts and Applications; Prentice Hall, New-Delhi,

Lippmann RP (1987). An introduction to computing with neural nets; IEEE Mag., 3(4): 4-2

Madala HR, Ivakhnenko AG (1994). Inductive Learning Algorithms for Complex Systems Modeling; CRC Press Inc., Boca Raton, , p.384.

Mirmirani S, Li HC (2004). Gold Price, Neural Networks and Genetic Algorithm; Comput. Econ. 23: 193–200,.

Ongsritrakul P, Soonthornphisaj N (2003). Apply decision tree and support vector regression to predict the gold price; Proceedings of the International Joint Conference on Neural Networks, 4: 2488-2492

Priestley MB (1988). Non-linear and non-stationary time series analysis; Academic Press.

Ripley BD (1993). Statistical aspects of neural networks. In J. Barndorff-Nielsen, L. Jensen and W.S. Kendall (eds.), Networks and Chaos – Statistical and Probabilistic Aspects, Chapman & Hall, London. pp. 40-123

Srinivasan N, Ravichandran V, Chan KL, Vidhya JR, Ramakirishnan S, Krishnan SM (2002). Exponentiated backpropagation algorithm for multilayer feedforward neural networks; Neural Information Processing,. ICONIP apos;02. Proceedings of the 9th International Conference, Nov. 2002, 1(18-22): 327 – 331.

Vandaele W (1983). Applied time series and Box-Jenkins models; Academic Press,