IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

A Comparative Study of Different Neural Networks Learning Algorithms for Forecasting Indian Gold Prices

Parminder Kaur

Asst. Professor, Dept. of Computer Science & Applications, Khalsa College for Women, Civil Lines, Ludhiana, India

Abstract: Artificial Neural Networks (ANN) have been widely used for forecasting purposes in various fields of science and engineering, economics, finance etc. This paper studies the Artificial Neural Networks based forecasting of Indian gold prices using three learning algorithms, namely, Standard Back Propagation (SBP), Back Propagation with Bayesian Regularization (BPR) and Levenberg-Marquardt (LM) algorithm. The performance of these three learning algorithms is compared using the statistical measures. The study found that the Levenberg-Marquardt learning algorithm based Artificial Neural Network model outperforms the other two learning algorithms based Artificial Neural Network Models in forecasting the Indian Gold prices.

Keywords: Artificial Neural Networks, Standard Back Propagation, Back Propagation with Bayesian Regularization, Lavenberg-Marquardt algorithm.

I. INTRODUCTION

India is the largest consumer of gold in the world. Gold holds religious and cultural importance in the lives of the Indians. It forms an important part of all the major festivities. At the same time, gold is also regarded as an important investment option as it is believed to be a reliable hedge in times of financial turmoil. Thus, prior forecast of the gold prices would immensely help the people in making intelligent investment decisions and would save them from the risk of losing on their market investments. Artificial neural networks are very versatile tools and have been widely used to tackle many issues [1-6].

In the recent past, the ANNs have been widely used for forecasting purposes. ANN can be viewed as a mathematical model or computational model that is inspired by the structure or functional aspects of biological neural networks. Neural networks are designed to extract existing patterns from noisy data. The procedure involves training a network (training phase) with a large sample of representative data, after which one exposes the network to data not included in the training set (validation or prediction phase) with the aim of predicting the new outcomes [7]. The interest in neural networks comes from the networks' ability to mimic human brain as well as its ability to learn and respond. As a result, neural networks have been used in a large number of applications and have proven to be effective in performing complex functions in a variety of fields [8].

Neural Networks have also been used to forecast the gold prices. But the forecasting efficiency of the ANNs depends upon the choice of the learning algorithms used to train the artificial neural networks. This paper discusses three learning algorithms- Standard Back Propagation (SBP), Back Propagation with Bayesian Regularization (BPR) and Levenberg-Marquardt (LM) algorithm. These algorithms are compared for their capability to predict Indian gold prices accurately.

II. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANN) are non linear models that have the potential of being used as effective forecasting tools in a large number of application areas.

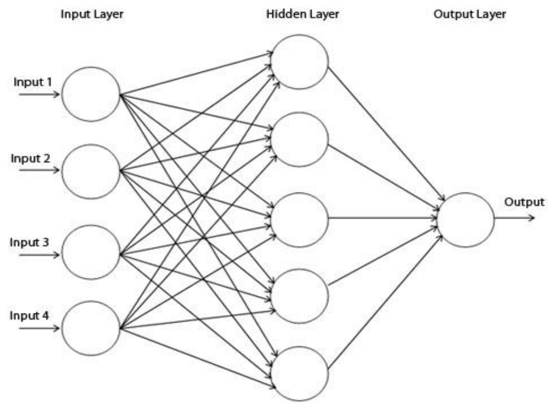

Fig. 1. Structure of MLPN.

The multilayer perceptron network (MLPN) model (also known as multilayer feed forward network) is chosen as the neural network architecture for modeling the Indian gold prices. These networks generally have one input layer, one or more than one hidden layers and one output layer. The number of layers and the number of units in each layer determine the functional complexity [9]. The nodes in each layer are connected to every node in the upper layer. The interconnection strengths between the

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

nodes are termed as weights. A training algorithm is then used to set the network's weights in order to minimize the difference between the actual output and the target values produced by the network.

III. NEURAL NETWORK LEARNING ALGORITHMS

The neural network learning algorithms play an important role in building an efficient forecasting model. An appropriate topology may still fail to give a better model, unless trained by a suitable learning algorithm. An appropriate learning algorithm would not only shorten the training time but would also produce more accurate results. These will tend to minimize the prediction errors of the networks. Thus, appropriate training of the networks is an important trait of the ANNs. During the training process of the network, data is iteratively presented to the network, so that the network can integrate within its structure, the knowledge gained from this repeated training. A number of training algorithms are available for training the Multi Layer Perceptron Network [10].

Studies indicate that the relative performance of learning algorithms depends on the characteristics of the problem. This study makes use of three learning algorithms to train the Artificial Neural Network model built to forecast the Indian gold prices. A comparative analysis is then carried out to identify the best one, the one which trains the network most efficiently and gives minimum forecasting error. The three learning algorithms used in the study are briefly described below:

A. Standard Back Propagation

In Standard Back Propagation learning algorithm, the weights are iteratively updated to map $X=(x_1, x_2, x_3,\ldots\ldots,x_p)$ to $Y=(y_1,y_2,y_3,\ldots\ldots y_p)$ where X is a set of input vectors and Y is a set of output vectors [11]. This is done by multiplying the set of input vectors with the corresponding weight values to obtain weighted input. The weighted inputs are then summed up and the output is generated using the following equations:

$$y_p = f(W_o h_p + \theta_0)$$

where $h_p = f(W_h x_p + \theta_h)$

where W_o is the output weight matrix

W_h is the hidden layer weight matrix,

h_p is a vector that represents for pattern 'p', the response of hidden layer,

θ_h denotes the bias for the hidden layer,

θ_o represents the bias for the output layer,

f(.) represents the sigmoid activation function.

The SBP algorithm involves minimizing the cost function defined as sum of squared error by the following function:

$$E = \frac{1}{2}\sum_p (t_p - y_p)^T (t_p - y_p)$$

where t_p represents the target output vector for the given pattern 'p'.

The connection weights between the neurons are adjusted using the gradient descent algorithm. If w represents the weight vector, then at *t-th* epoch, the value of the weight vector w will be given by the following equation:

$$\Delta w_t = -\eta \nabla E(w) | w = w(t) + \alpha \Delta w_{t-1}$$

Where η represents the learning rate and it controls the incremental step size of the iterations.

α represents the momentum factor. The convergence in case of large scale problems depends on the choice of both the η and α values.

B. Bayesian Regularization (BPR)

This learning algorithm was proposed by MacKay. This algorithm is based on the principle of regularization which ultimately improves the generalization ability of a network which generates small errors not only for in- sample data but also for out- of -sample data [12]. The regularization principle involves constraining the magnitude of network parameters by restricting the size of biases and weights to small values. As a result, the output generated by the network is smoother with lesser chances of overfitting of data [13]. The cost function in this learning algorithm is defined by the following function:

$$F = \gamma E_d + (1-\gamma) E_w$$

where E_d is same as the cost function E, defined in the Standard Back Propagation learning algorithm.

$$E_w = \|w^2\| / 2$$

E_w represents the summation of squares of parameters of the network.

γ is the performance ratio parameter. Its value determines the emphasis of training on regularization. A relatively big value of γ will result in smaller error E_d values. On the other hand, a smaller value of γ will result in reduction of sizes of parameters but to a bigger error value. The resulting network response will hence be smoother.

Using the Bayesian framework approach, the optimal regularization parameters can be automatically determined [14]. The Bayesian framework approach assumes probability distribution. This distribution is assumed over a weight space. Initially, this space is initialized to a prior distribution. Then for the weight p(w | D, γ, M), then posterior probability distribution is given by the Bayesian rule as

$$p(w|D,\gamma,M) = \frac{p(D\,|w,\gamma,M)\; p(w|\gamma\,,M)}{p(D\,|\,\gamma,M)}$$

Where $p(D\,|\,\gamma,M)$ represents the likelihood function,

$p(D\,|\,\gamma,M)$ represents the normalization factor,

$p(w|\gamma\,,M)$ represents the prior distribution.

The optimal weight in Bayesian framework maximizes the posterior probability. The following Bayesian rule is used to optimize the value of γ:

$$p(\gamma|D,M) = \frac{p(D|\gamma,M)\; p(\gamma|M)}{p(D\,|\,M)}$$

C. Levenberg-Marquardt (LM) Algorithm

The LM algorithm is one of the most appropriate higher-order adaptive algorithms known for minimizing the errors of a neural network. It is a member of a class of learning algorithms called ''pseudo second order methods."

Standard gradient descent algorithms use only the local approximation of the slope of the performance surface (error versus weights) to determine the best direction to move the weights in order to lower the error. Second-order methods use the Hessian or the matrix of second derivatives (the curvature instead of just the slope) of the performance surface to determine the weight update, while pseudo second order methods approximate the Hessian. In particular the LM utilizes the so-called Gauss-Newton approximation that keeps the Jacobean matrix and discards second-order derivatives of the error.

Mathematically, Levenberg-Marquardt algorithm minimizes the error functions [15] of the form,

$$E=1/2\sum k(e_k)^2 = 1/2||e||^2$$

where e_k represents the error in the pattern k,
e represents an error vector with e_k as its element.

In case of small difference between two successive weight vectors, then e can be expanded using the Taylor series as:

$$e_{(j+i)} = e_{(j)} + \delta e_k / \partial_{w_i}(w_{(j+1)} - w_{(j)})$$

The error function can then be defined as:

$$E=1/2 \| e_{(j)} + \delta e_k / \partial_{w_i}((w_{(j+1)} - w_{(j)}))||^2$$

On minimizing the error function, the weight vector is then given by:

$$w_{(j+1)} = w_{(j)} - (Z^T Z)^{-1} Z^T e_{(j)}$$

where $(Z)_{ki} \equiv \partial_{e_k}/\partial_{w_i}$

To ensure that the linear approximation is validated, the error function in the Levenberg-Marquardt algorithm is modified as

$$E=1/2||e_{(j)} + \delta e_k / \partial_{w_i}((w_{(j+1)} - w_{(j)}))||^2 + \lambda\|w_{(j+1)} - w_{(j)}||^2$$

λ denotes the step size.

On minimization, the modified error in the LM algorithm with respect to $w_{(j+1)}$ becomes

$$w_{(j+1)} = w_{(j)} - (Z^T Z+\lambda I)^{-1} Z^T e_{(j)}$$

A key advantage of the LM approach is that it defaults to the gradient search for large values of λ and to Newton method for small values of λ. The LM algorithm combines the best features of the Gauss-Newton technique and the steepest-descent algorithm but avoids many of their limitations. In particular, it generally does not suffer from the problem of slow convergence [16].

IV. DATA AND METHODOLOGY

The study used historical Indian gold prices from Multi Commodity Exchange (MCX) of India for developing the ANN models. Matlab 8.0 was used for developing the ANN models. The input data consisted of daily closing prices of Gold. Data covering the period from January 3, 2007 to December 31, 2015 was used for the study.

TABLE I: DESCRIPTIVE STATISTICS OF THE GOLD PRICE DATA UNDER STUDY

Descriptive Statistics	Value
Average	20971.74
Standard Deviation	7583.54
Skewness	-0.22
Excess-Kurtosis	-1.46
Median	21710
Min Value	8525
Max Value	32943

A three-layer structure MLP (one input layer, one hidden layer, and one output layer) was selected with hyperbolic tangent (tanh) transfer function for hidden layer and linear transfer function for output layer. 67% of the data was used for training the network, and the remaining 33% of data was used for testing. Networks were trained for a fixed number of epochs. The optimal number of hidden neurons was obtained experimentally by changing the network design and running the training process several times until a good performance was obtained. A random number generator was used to assign the initial values of weights and thresholds with a small bias as a difference between each weight connecting two neurons together since similar weights for different connections may lead to a network that will never learn. Using three different learning algorithms, namely, Standard Back Propagation, Bayesian Regularization and Levenberg-Marquardt algorithms, three different Artificial Neural Network models were developed. The monthly gold prices for the year 2015 were forecast using the all the three ANN models. The performance of the three ANNs was compared using the statistical measures.

V. COMPARATIVE STUDY OF THE ARTIFICIAL NEURAL NETWORK MODELS USING DIFFERENT LEARNING ALGORITHMS

The ANN algorithm to forecast the Indian Gold closing prices is trained using the three different learning algorithms under study- Standard Back Propagation, Bayesian Regularization and Levenberg-Marquardt algorithms under the same set of conditions. The performance of the ANNs using the learning algorithms is compared using the widely used statistical measures, namely, Mean Absolute Error (MAE), Normalized Mean Square Error (NMSE), Correct Down Trend (CDT), Correct Up Trend (CUT). These metrics are calculated as follows:

$$\text{MAE [17][18][19][20]} = \frac{1}{N} |x_k - x\hat{}_k|$$

where x is the actual value and $x\hat{}_k$ is the predicted value.

DOI 10.17148/IJARCCE.2016.54265

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

NMSE [17] [18]= $\frac{1}{\sigma^2 N}\sum_k (x_k - x\hat{}_k)^2$

CDT [21] = $\frac{\sum_k d_k}{\sum_k t_k}$, $d_k = 1$ if$(x\hat{}_k - x\hat{}_{k-1})<0$,
$(x_k - x_{k-1})(x\hat{}_k - x\hat{}_{k-1}) \geq 0$;
$d_k = 0$, otherwise.
t_k =1 if (x_k - x_{k-1}) <0,
t_k =0, otherwise .

CUT [21] = $\frac{\sum_k d_k}{\sum_k t_k}$, $d_k = 1$ if$(x\hat{}_k - x\hat{}_{k-1})>0$,
$(x_k - x_{k-1})(x\hat{}_k - x\hat{}_{k-1}) \leq 0$,
$d_k = 0$, otherwise.
t_k =1 if (x_k - x_{k-1}) >0,
t_k =0, otherwise.

Smaller values of MAE and NMSE indicate higher accuracy. CDT and CUT performance metrics measure the accuracy of the predicted downward and upward trend, respectively.

TABLE III: COMPARATIVE ANALYSIS OF THE LEARNING ALGORITHMS

Learning Algorithm applied to Neural Network Model	**Performance Metrics**			
	MAE	**NMSE**	**CDT**	**CUT**
Standard Back Propagation	0.0058	0.653	79.24	80.23
Bayesian Regularization	0.0049	0.349	85.12	84.68
Lambert - Marquardt (LM)	0.0038	0.231	89.58	87.49

The results obtained clearly depict that the Levenberg-Marquardt (LM) learning algorithm has the smallest Normalized Mean Square Error (NMSE) value of 0.231 and Mean Absolute Error (MAE) of 0.0038.

The LM algorithm also gives highest accuracy of the predicted downward and upward trends with accuracy levels upto 89.58 and 87.49 respectively. This implies that the Indian gold prices can be best predicted by training the Artificial Neural Network using the Levenberg-Marquardt (LM) algorithm.

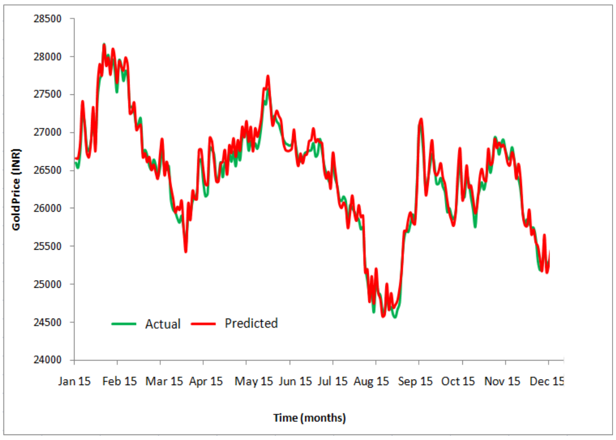

Fig. 2. Comparative graph showing the predicted Neural Network output obtained by training the ANNs using LM algorithm with the actual gold.

Fig. 3. Comparative graphs showing the predicted Neural Network output obtained by training the ANNs using BR algorithm with the actual gold.

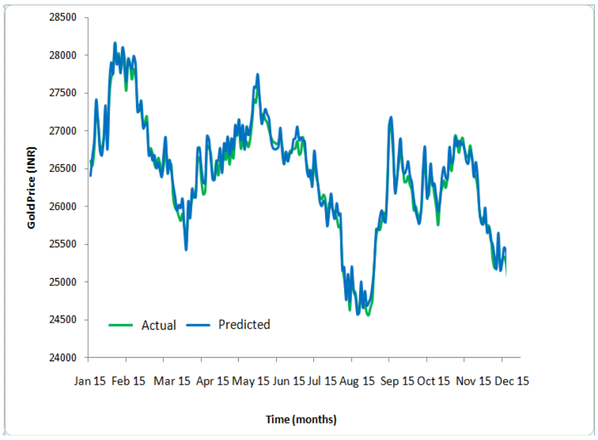

Fig. 4. Comparative graphs showing the predicted Neural Network output obtained by training the ANNs using SBP algorithm with the actual gold.

Figures 2, 3 and 4 depict the comparative graphs showing the predicted Neural Network output obtained by training the ANNs with Levenberg-Marquardt (LM) algorithm, Bayesian Regularization (BR) and Standard Back Propagation (SBP) algorithms.

VI. CONCLUSION

The results obtained using the published gold price data of MCX on the performance of the three learning algorithms-SBP, BPR and LM to predict the Indian Gold prices have been presented in the paper. The main aim of the study was to study the impact of the learning algorithms on prediction of the prices in the Indian Gold market using Artificial neural Networks. When analyzing three different learning optimization algorithms, it was found that the Levenberg–Marquardt was superior compared to the tested set of Standard Back Propagation (SBP) and Back Propagation with Bayesian Regularization (BPR) algorithms.

Thus the prediction accuracy of the Indian gold prices can be considerably enhanced by proper choice of the learning optimization algorithm while training the Artificial Neural Network.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

REFERENCES

[1] D. Samek, P. Varacha, "Time Series Prediction Using Artificial Neural Networks", International Journal Of Mathematical Models And Methods In Applied Sciences, Issue 1, Vol. 7, 2013.

[2] Yodele et al., "Stock Price Prediction using Neural Network with Hybridized Market Indicators", Journal of Emerging Trends in Computing and Information Sciences, Vol. 3, No 1, January 2012 ISSN 2079-8407.

[3] Y. Kara, M, A. Boyacioglu, O. K, Baykan, "Predicting Direction of Stock Price Index Movement Using Artificial Neural Networks and Support Vector Machines: The sample of the Istanbul Stock Exchange", Expert Systems with Applications 38, pp 5311–5319, 2011

[4] S. Agrawal, M. Jindal, G. N. Pillai , "Momentum Analysis based Stock Market Prediction using Adaptive Neuro-Fuzzy Inference System (ANFIS)", Proceedings of IMECS, 2010.

[5] M. Qi., G. P. Zhang, ''Trend time series modeling and forecasting with neural networks'' IEEE Transactions on Neural Networks, 19(5), pp.808–816, 2008

[6] H. K. Cigizoglu, "Estimation, forecasting and extrapolation of river flows by artificial neural networks" , Hydrological Sciences Journal, vol. 48, no. 3, pp. 349–361, 2003.

[7] C. M. Bishop, "Neural Networks for Pattern Recognition", Oxford University Press, Oxford, UK, 1995.

[8] L. Fausett, "Fundamental of Neural Networks", Prentice Hall, New York, NY, USA, 1994.

[9] S. Haykin, "Neural Networks: A Comprehensive Foundation", Macmillan Publishing Company, New York, NY, USA, 1994.

[10] S. E. Fahlman, "An empirical study of learning speed in backpropagation networks," Computer Science Technical Report CMU-CS-88-162, Carnegie Mellon University, Pittsburgh, Pa, USA, 1988.

[11] D.E Rumelhart, J.L. McClelland and the PDP research group, Parallel Distributed Processing, vol. 1, MIT Press, 1986.

[12] D.J.C. Mackay, "Bayesian interpolation", Neural Computation, vol. 4, pp. 415-447, 1992.

[13] M.T. Hagan, H.B. Demuth and M.H. Beale, Neural Network Design, PWS Publishing, Boston, MA, 1996

[14] B. D. Klein , D. F. Rossin, "Data quality in neural network models: effect of error rate and magnitude of error on predictive accuracy", OMEGA: Int. Journal of Management Science, vol. 27, pp 569-582, 1999.

[15] Aldrich .C , "Exploratory Analysis of Metallurgical Process Data with Neural Networks and Related Methods", Elsevier Science, British Library Cataloguing in Publication Data, Netherlands, Vol.1, pp.56-57, 2002

[16] M. T. Hagan , M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 989–993, 1994.

[17] C. Hamzacebi, "Improving artificial neural networks' performance in seasonal time series forecasting", Information Sciences 178 , pp. 4550-4559, 2008.

[18] L.J. Cao , F. E.H. Tay, "Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting", IEEE Transaction on Neural Networks, Vol. 14, No. 6, pp. 1506-1518, 2003.

[19] G.P. Zhang, "A neural network ensemble method with jittered training data for time series forecasting", Information Sciences 177, pp: 5329–5346, 2007.

[20] H. Park, "Forecasting Three-Month Treasury Bills Using ARIMA and GARCH Models", Econ 930, Department of Economics, Kansas State University, 1999

[21] D.E. Rumelhart., G.E. Hinton, R.J. Williams, " Learning internal representations by error propagation.", In D. E. Rumelhart & G. E. Hinton (Eds.), Parallel Distributed Processing (Vol. 1, pp. 318-362). Cambridge, MA: MIT Press, 1986.

[22] J. H.Wang , J. Y. Leu, "Stock market trend prediction using ARIMA-based neural networks," Proc. of IEEE Int. Conf. on Neural Networks, vol. 4, pp. 2160-2165, 1996.