Evaluation of GARCH, RNN, and FNN models for forecasting volatility in the financial markets

David Enke
Missouri University of Science and Technology

Evaluation of GARCH, RNN and FNN Models for Forecasting Volatility in the Financial Markets

Ajitha Vejendla* and David Enke**

Volatility forecasting is an important task for those associated with the financial markets, and has occupied the attention of academics and practitioners over the last two decades. This research paper reflects the importance of volatility in option pricing, security valuation and risk management. It investigates the forecasting ability of Feed-Forward Neural Networks (FNN) using backpropagation learning, Recurrent Neural Networks (RNN), and a Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model based on empirical price and historical volatility data. The performance of the three models is studied and compared using data of S&P 500, DJIA, NYSE and NASDAQ indexes. The results obtained from these selected models are anticipated to have significant market directional ability and lower prediction errors.

Introduction

Volatility is the most important variable in the pricing of derivative securities, whose trading volume has continued to rise over the last decade. To price an option, it is imperative to know the volatility of the underlying asset from now until the option expires. The market convention is to list option prices in terms of volatility units. Hence, trading can be executed by buying derivatives that are written on volatility itself. In this case, the definition and measurement of volatility will be clearly specified in the derivative contracts. In these new contracts, volatility now becomes the underlying 'asset'. Volatility forecast and a second prediction on the volatility of volatility over the defined period are needed to price such derivative contracts.

This study applies performance measures to evaluate the neural network models for financial volatility measurement. The forecasted value determines how well the system fits the model. However, this information may not be adequate to assess financial applications. The Mean Squared Error (MSE) is one of the measurement methods used to compare the performance of the neural network models.

Background

The use of neural networks as nonlinear forecasters in the financial area seems very promising. However, the development of financial applications using neural network is a complex process. Extensive data pre-processing and experimentation with network parameters in the application development is a time-consuming process. In the area of stock prediction, the

* Independent Researcher, H.No. 10-3-905, Vijaya Nagar Colony, Hyderabad, India; and is the corresponding author. E-mail: ajitha.vejendla@gmail.com

** Professor and Chair, Department of Engineering Management and Systems Engineering, Missouri University of Science and Technology, 600 W 14th Street, Rolla, MO, 65409-0370, USA. E-mail: enke@mst.edu

use of neural networks has been investigated for feasibility by numerous researchers. Experiments have been conducted using market data throughout the global financial world. In addition, different input variables and neural network architectures were applied.

Neural networks have been studied in many financial areas, such as prediction of corporation bankruptcy (Fletcher and Goss, 1993), stock trading (Wilson, 1994), initial public offering pricing (Jain and Nag, 1995), currency option trading (Quah *et al.*, 1995), and foreign exchange rate forecasting (White and Racine, 2001).

Poon and Granger (2003) surveyed 39 studies comparing the out-of-sample forecasting ability of GARCH and historical volatility. The results showed that in 22 studies, historical volatility (including weighted standard deviation measures) forecasts better, while in the remaining 17 studies, GARCH forecasts better. Some very early papers using GARCH forecasts to value options simply assigned the forecast volatility for day $t+1$ to the entire period out to $t+N$. A volatility forecast over multiple periods' horizon is being used for the present analysis.

Volatility Measures

There are basically three notions of volatility in the literature: (1) Historical Volatility; (2) Implied Volatility; and (3) Model-Based Volatility (determined, for example, by models from the GARCH family).

1. Historical Volatility: The historical measure is similar to the basic volatility measure applied in RiskMetrics and it is well suited for the purpose of trading options on financial indexes.

   This measure estimates volatility using the historical stock price data, which is normally observed at fixed intervals of time, such as daily, weekly, or monthly. Historical volatility is usually called statistical volatility and is calculated as the standard deviation of a stock's return over a fixed period of time. Stock return is often defined as the natural logarithm of the closing prices between each interval of time.

   The return and historical volatility are calculated as shown in Equations (1) and (2) respectively:

$$X_i = \ln\left(\frac{P_i}{P_{i-1}}\right) \qquad \ldots(1)$$

$$HV = \sqrt{\frac{1}{n-1}\sum(X_i - \overline{X})^2} \qquad \ldots(2)$$

   where X_i = Return at the i^{th} interval, P_i = Stock close price at the end of i^{th} interval, and $n+1$ = Number of observations or number of observed days.

The procedure for determining the appropriate number of observations is not simple. Normally, more observations can lead to increased accuracy, however, volatility changes over time, and data from deep in the past may not be relevant for predicting the future. A basic rule is to set *n* equal to the number of days for which the volatility is to be applied (Hull, 2003). If historical volatility is high, it implies that the stock has previously been displaying increased movement in price. And if it is low, the volatility implies quiet trading or low movement in price. Benchmarking with the historical volatility of other stocks allows one to compare the volatility of a stock with the general market or the stocks within a specific sector. In the same way, one can estimate the increased or decreased movement of volatility by benchmarking the volatility of a stock with its own historical volatility. For example, assuming that the 15-day historical volatility of a stock is 10% and the 60-day historical volatility is 30%, this implies that the stock has recently had a sharp decrease in volatility.

2. Implied Volatility: This measure is estimated from the extracted options' volatilities implied by the Black-Scholes model.

3. Model-Based Volatility: The GARCH models try to reconstruct volatilities from the series of returns. Basic to these models is the notion that the returns can be decomposed into a predictable component and an unpredictable component, which is assumed to be a zero mean Gaussian (or *t*-distributed) noise of finite variance.

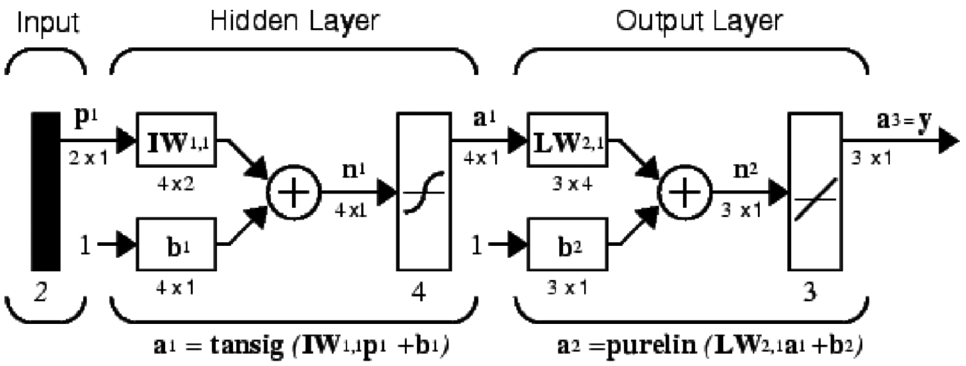
   The models are thus characterized by time-varying conditional variances and are therefore well suited to explain volatility clusters typically present in the series of returns. The conditional mean is modeled as a linear function of the previous value. The GARCH model estimates the volatility at time as the conditional variance. Given a training set of historical returns, the free parameters of the GARCH model are estimated by the maximum likelihood method. Note that unlike recurrent neural networks that are trained on quantized volatility differences, the GARCH models that define a volatility measure are trained on a series of returns of the underlying index.

Performance Evaluation of Neural Network Models

Feed-Forward Neural Networks with Backpropagation Learning

Feed-forward networks (Figure 1) often have one or more hidden layers of sigmoid neurons, followed by an output layer of linear neurons. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear and linear relationships between input and output vectors. The linear output layer lets the network produce values outside the range –1 to +1.

Figure 1: Feed-Forward Neural Network with Backpropagation



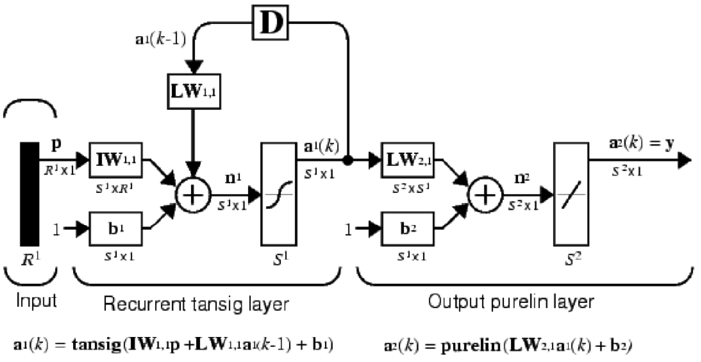
On the other hand, in order to constrain the outputs of a network (such as between 0 and 1), the output layer should use a sigmoid transfer function (such as logsig) (MATLAB Version 7.2, 2006). In the present simulations, two hidden layers (tansig, tansig) have been considered with 4 and 4 neurons in each of the layers.

Recurrent Neural Networks with Elman Network Architecture

The Elman network is commonly a two-layer network with feedback from the first-layer output to the first-layer input. This recurrent connection allows the Elman network to both detect and generate time-varying patterns.

The Elman network (Figure 2) has tansig neurons in its hidden (recurrent) layer, and purelin neurons in its output layer. This combination is special as two-layer networks with these transfer functions can approximate any function (with a finite number of discontinuities) with arbitrary accuracy. The only requirement is that the hidden layer must have enough neurons. More hidden neurons are needed as the function being fitted increases in complexity (MATLAB Version 7.2, 2006).

Figure 2: Recurrent Neural Network – Elman



Mean Squared Error

The MSE is one of the measurement methods used to compare the performance of the neural network models. It is used to evaluate and compare the productive power of the models by finding the error between the predictive output value and the actual value. The equation used for MSE is:

$$MSE = \frac{\sum \left(Y_t - \hat{Y}_t\right)^2}{N}$$

where

Y_t is the actual price;

$\hat{Y}_t$ is the price forecasts produced by the neural network models; and

N is the number of test datasets.

Low MSE usually means high accuracy and better prediction ability, although it is not necessarily a sign of high return or profitability in the stock market.

The forecasting ability of Feed-Forward Neural Networks (FNN) using backpropagation learning and Recurrent Neural Networks (RNN) with Elman network architecture have been investigated using the data from the stock markets. The data used is of price and volatility over a period of 8 years for the NASDAQ (COMPQX), DJIA, NYSE and the S&P 500 (SPX) indexes (see Tables 1 and 2).

Table 1: Datasets

Dataset	Period	No. of Days of Study
Overall Period	1996-2004	2016
Training Set	1996-2001	1512
Testing Set	2002-2004	504

Table 2: Functions Used in Simulations

Inputs	Open Volatility, High, Low, Close Output
Layers	4
Hidden Layers	4
Output	1
Functions	tansig, tansig, logsig logsig, logsig, purelin
Training functions	trainbfg

From the simulations (Table 3), it can be concluded that for a given set of data, RNN gives a better prediction of volatility as compared to FNN with backpropagation learning.

Table 3: Volatility Forecasts Using FNN and RNN

	SPX	COMPQX	NYSE	DJIA	
MSE	0.0066132	0.023578	0.00536508	0.006351	FNN
MSE INV	151.21	4.24	186.39	157.45	FNN
MSE	0.0059793	0.0174518	0.00377881	0.00584524	RNN
MSE INV	167.24	57.3	264.63	171.08	RNN

Comparison of Historical Volatility to GARCH Implied Volatility GARCH(1, 1) Models

The GARCH models try to reconstruct volatilities from the series of returns. Basic to these models is the notion that the returns can be decomposed into a predictable component and an unpredictable component, which is assumed to be a zero mean Gaussian (or *t*-distributed) noise of finite variance. The models are thus characterized by time-varying conditional variances and are therefore well suited to explain volatility clusters typically present in the series of returns.

The conditional mean is modeled as a linear function of the previous value. The GARCH model estimates the volatility at time as the conditional variance. Given a training set of historical returns, the free parameters of the GARCH model are estimated by the maximum likelihood method. Note that, unlike RNN that are trained on quantized volatility differences, the GARCH models that define a volatility measure, are trained on a series of returns of the underlying index.

The output of the garchpred GARCH model, SigmaTotal, is a matrix of volatility forecasts of returns over multi-period holding intervals, i.e., the first row contains the expected standard deviation of returns for assets held for one period for each realization of series; the second row contains the standard deviation of returns for assets held for two periods, and so on. Thus, the last row contains the forecast of the standard deviation of the cumulative return obtained if an asset was held for the entire forecast horizon (MATLAB Version 7.2, 2006).

Garchpred computes the elements of SigmaTotal by taking the square root of:

$$\mathrm{var}_t\left[\sum_{i=1}^{g} y_{t+i}\right]=\sum_{i=1}^{g}\left[\left(1+\sum_{j=1}^{g-i}\psi_j\right)^2 E_t\left(\sigma_{t+i}^2\right)\right] \quad \ldots(3)$$

where g is the forecast horizon of interest (NumPeriods), and ψ_j is the coefficient of the j^{th} lag of the innovations process in an infinite-order MA representation of the conditional mean model (see the function garchma).

In the special case of the default model for the conditional mean, $y_t = C + \varepsilon_t$ this reduces to:

$$\text{var}_t\left[\sum_{i=1}^{g} y_{t+i}\right] = \sum_{i=1}^{g} E_t\left(\sigma_{t+i}^{2}\right) \qquad \text{...(4)}$$

The SigmaTotal forecasts are correct for continuously compounded returns, and approximate for periodically compounded returns. SigmaTotal is the same size as SigmaForecast, if the conditional mean is modeled as a stationary invertible ARMA process (MATLAB Version 7.2, 2006).

In the present analysis, the historical volatility data is compared to the GARCH implied volatility obtained over a period of 6 months (Table 4). Based on the prior conclusion that RNN is better suited for the analysis for a specified set of data, the simulations have been done using the Elman network architecture of RNN (Table 5). The returns data of NASDAQ, DJIA, NYSE and S&P 500 markets have been used to obtain the 180-day implied volatility data using GARCH models. A volatility forecast over multiple period horizon has been used. This data has been compared to the historical volatility over the period of 180 days. The results obtained are presented in Table 6.

Table 4: Datasets

Dataset	Period	No. of Days of Study
Overall Period	February-July, 2004	180
Training Set	February-May, 2004	120
Testing Set	June-July, 2004	60

Table 5: Functions Used in Simulation Using RNN

Inputs	Historical Volatility, Implied Volatility
Output	
MSE Layers	2
Hidden Layers	4
Output	1
Functions	tansig, logsig
Training Functions	trainscg

Table 6: Comparison of Historical Volatility and GARCH Implied Volatility Using RNN

	COMPQX	DJIA	NYSE	SPX
MSE	0.579048	0.312141	0.0421336	0.249464
MSE INV	1.727	3.2	23.73	4.008

Conclusion

From the analysis, it can be concluded that GARCH models give a better forecast of volatility than historical volatility. But these indications definitely depend on the data, because the models perform differently for different stocks, and sometimes may be market-specific. MSE was also utilized to evaluate the results of the neural network, but the research has shown that low MSE does not always produce better return in a stock market when the model is a combination of a neural network prediction.

Volatility is an important metric in the financial markets. Forecasting volatility will have broader applications in portfolio management, options trading, demand forecasting, etc. Using neural networks to forecast volatility of stock market indexes and the performance evaluation of different models under dynamic market conditions could be very useful in the study of option strategies and option valuation. ❖

Bibliography

1. Akgiray Vedat (1989), "Conditional Heteroskedasticity in Time Series of Stock Returns: Evidence and Forecasts", *J. Bus.,* Vol. 62, pp. 55-80.

2. Amornwattana S and Enke D (2004), "A Comparison of Methods for Derivative Trading Using Stock Return Forecasts", *Artificial Neural Networks in Engineering*, Vol. 14, pp. 741-746.

3. Amornwattana S, Cihan H Dagli and Enke D (2004), "A Hybrid Option Pricing Model Using a Neural Network for Estimating Volatility", *International Journal of General Systems*, Vol. 36, pp. 558-573.

4. Bollerslev T (1986), "Generalized Autoregressive Conditional Heteroskedasticity", *Journal of Econometrics*, Vol. 31, pp. 307-327.

5. Fletcher D and Goss E (1993), "Forecasting with Neural Networks: An Application Using Bankruptcy Data", *Information and Management*, Vol. 24, pp. 159-167.

6. Gately Edward (1996), *Neural Network for Financial Forecasting,* John Wiley &Sons, Inc.

7. Hull J (2003), Options, Futures and Other Derivatives, 5th Edition, Prentice Hall.

8. Jain B A and Nag B N (1995), "Artificial Neural Network Models for Pricing Initial Public Offerings", *Decision Sciences*, Vol. 26, pp. 283-302.

9. Leung T M, Daouk H and Chen A S (2000), "Forecasting Stock Indices: A Comparison of Classification and Level Estimation Models", *International Journal of Forecasting*, Vol. 16, pp. 173-190.

10. MATLAB Version 7.2 (2006), The MathWorks Inc., Natick, Massachusetts.

11. Mayo Herbert B (2003), *Investments: An Introduction,* Thomson Southwestern.

12. McNelis Paul D (2005), *Neural Networks in Finance: Gaining Predictive Edge in the Market,* Elsevier Academic Press, Burlington, MA.

13. Poon Ser-Huang and Granger Clive W J (2003), "Forecasting Volatility in Financial Markets: A Review", *Journal of Economic Literature*, Vol. 41, pp. 478-539.

14. Quah T-S, Tan C-L, Teh H-H and Srinivasan B (1995), "Utilizing a Neural Logic Expert System in Currency Option Trading", *Expert Systems with Applications*, Vol. 9, No. 2, pp. 213-222.

15. White H and Racine J (2001), "Statistical Inference: The Bootstrap, and Neural-Network Modeling with Application to Foreign Exchange Rates", *IEEE Trans. Neural Networks,* Vol. 12, No. 4, pp. 657-673.

16. Wilson C L (1994), "Self-Organizing Neural Network System for Trading Common Stocks", IEEE International Conference on Neural Networks, June 27-July 2, Vol. 6, pp. 3651-3654.

Reference # 37J-2013-03-03-01