



FORECASTING VOLATILITY IN THE STOCK MARKET

BMI PAPER

Sergiy Ladokhin

Supervisor: Dr. Erik Winands

VU University Amsterdam

Faculty of Science

Business Mathematics and Informatics

De Boelelaan 1081a

1081 HV Amsterdam

February 2009

Contents

Management summary	4
Introduction	5
Volatility concept and its usage in financial risk management.....	6
Forecasting financial volatility	8
Historical volatility models	10
Black-Scholes formula and implied volatility.....	11
Autoregressive and Heteroskedastic models.....	13
Models based on artificial neural networks	15
Comparison of volatility forecasting models	17
Conclusions and Perspectives	25
Bibliography	26
Appendix A	29

Management summary

This paper focuses on the problem of volatility forecasting in the financial markets. It begins with a general description of volatility and its properties, and discusses its usage in financial risk management. The paper then examines the accuracy of several of the most popular methods used in volatility forecasting: historical volatility models (including Exponential Weighted Moving Average), the implied volatility model, and autoregressive and heteroskedastic models (including ARMA model and GARCH family of models). We also test two methods from a new class of models which utilizes the Artificial Neural Networks. The forecasting accuracy of the models is tested using the S&P 500 stock index; the advantages and disadvantages of each model are discussed. This research is designed to be of interest to both theoretical researchers and practitioners in the finance industry.

Introduction

The main characteristic of any financial asset is its return. Return is typically considered to be a random variable. An asset's volatility, which describes the spread of outcomes of this variable, plays an important role in numerous financial applications. Its primary usage is to estimate the value of market risk. Volatility is also a key parameter for pricing financial derivatives. All modern option-pricing techniques rely on a volatility parameter for price evaluation. Volatility is also used for risk management applications and in general portfolio management. It is crucial for financial institutions not only to know the current value of the volatility of the managed assets, but also to be able to estimate their future values. Volatility forecasting is especially important for institutions involved in options trading and portfolio management.

Accurate prediction of the values of financial indicators is complicated by complex interconnections, which are often convoluted and not intuitive. This makes the prediction of volatility a challenging task even for experts in this field. Mathematical modeling can assist in detecting the dependencies between current values of the financial indicators and their future expected values. Model-based quantitative forecasts can provide financial institutions with a valuable estimate of a future market trend. Although some experts believe that future events are unpredictable, evidence to the contrary exists. For example, financial volatility has a tendency to cluster and exhibits considerable autocorrelation (i.e., the dependency of future values on past values). These features provided the justification for formalizing the concept of volatility and creating volatility-forecasting mathematical techniques, which started appearing the late 70's. Since then, a number of successful models for volatility forecasting have been introduced.

The purpose of this work is to compare different mathematical methods used in volatility forecasting, and to describe their advantages and disadvantages. Another objective is to examine the concept of volatility and other related issues, which is expected to influence the forecasting ability of certain methods. Specifically, we tested several classes of volatility forecasting models that are widely used in modern practice: historical (including moving averages), autoregressive, conditional heteroscedastic models, and the implied volatility concept. In addition, we examined a relatively new class of models – those based on artificial neural networks. Their ability to capture a non-linear behavior is an important feature, which can improve current 'classical' methods. All of the models were tested on the volatility of the S&P 500 index, which is one of the main indexes of the US stock market and plays an important role not only for US economy but for the world economy as well. Although

the results of this work are applicable primarily to a well-diversified equity indexes, there are no stringent restrictions, and the principal results are applicable to other markets and products.

Volatility concept and its usage in financial risk management

Volatility refers to the spread of all outcomes of an uncertain variable. In finance, we are interested in the outcomes of assets returns. Volatility is associated with the sample standard deviation of returns over some period of time. It is computed using the following formula:

$$\hat{\sigma} = \sqrt{\frac{1}{T-1} \sum_{t=1}^T (r_t - \mu)^2}$$

where r_t is the return of an asset over period t and μ is an average return over T periods.

The variance, σ^2 , could also be used as a measure of volatility. But this is less common, because variance and standard deviation are connected by simple relationship.

Volatility is a quantified measure of market risk. Volatility is related to risk, but it is not exactly the same. Risk is the uncertainty of a negative outcome of some event (e.g. stock returns); volatility measures a spread of outcomes. This includes positive as well as negative outcomes.

Financial time series data sets have several characteristics which are crucial to note for the purposes of modeling and forecasting:

- *Fat tails.* Market returns have distributions with fatter tails than the normal distribution. This results to a higher kurtosis. The normal distribution has the fourth moment equal to three, however several studies, e.g. (J. Knight, 2007), have shown that the distribution of market returns have sample fourth moments larger than three.
- *Volatility Clustering.* Volatility is not constant over time. Moreover it exhibits certain patterns. This means that large movements in returns tend to be followed by further large movements. Thus the economy has cycles with high volatility and low volatility periods. High volatility periods usually refer to economic crises and recessions.

- *Leverage effect.* Price movements are negatively correlated with volatility. This means that the volatility of stock tends to increase when the prices drops. This effect is particularly important for options markets. (Black F. , 1976)
- *Co-movements in volatility.* Volatility of different markets tends to move in certain patterns. For example, a high volatility of one currency pair listed on the FOREX exchange will most likely cause higher volatility of the returns for holders of other currency pairs.

Volatility is a key parameter for derivatives pricing models. The market of financial derivatives is one of the biggest financial markets. Derivatives are financial contracts for which values are derived from the prices of the underlying assets. Major classes of derivatives are: futures/forwards, options and swaps. There is a great variety of derivative contracts within these classes. The Black-Scholes formula and the binomial tree models are the most widely used approaches for derivative evaluation. All approaches, however, have the volatility of the underlying asset as a key parameter. Moreover, an accurate forecast of the future volatility results in a more precise derivative pricing.

Volatility is used in many financial ratios. For example, the Sharp-ratio is a measure of excess return per unit of risk. It is one of the most frequently used measures for comparison of the investment performance.

$$S = \frac{E[R] - R_f}{\sigma}$$

where $E[R]$ is the expected return of the portfolio, R_f the risk free rate and σ the volatility of the portfolio.

Volatility can be used in some risk management applications, such as Value at Risk (VaR). VaR is a maximum loss over N days, that will not be exceeded, and a financial institution is X percent certain about this. The management of financial institutions (FI) is usually interested in such a risk characteristic: “We are X percent certain, that we will not lose more than V dollars, in the next N days” (Hull, 2002); Where V is VaR, X is a confidence level and N is the time horizon in days.

Market risk is the risk that an investment (equity, fixed income, commodity, etc.) will decrease in value due to movements in market factors. VaR is a key characteristic of a market risk. This methodology is a component of regulatory requirements for FI’s capital under market risk. Regulators typically use $N = 10$ and $X = 99\%$. However, some FI (e.g. clearing houses) can use a daily VaR with a higher confidence level. For example, Daily earnings at risk (DEaR) is a measure of VaR for a 1 day period, typically using a 95% confidence level. Under the model-

building approach (Hull, 2002) VaR is calculated using the daily volatility of an asset σ_d :

$$VaR = \sigma_d \cdot A \cdot Conf \cdot \sqrt{N}$$

Where σ_d is the daily volatility of an asset, A is the total amount of money in an open position on this asset, $Conf$ is the confidence level and N is the time horizon in days. As we can see, daily volatility is a key parameter in VaR calculations. A FI which can more accurately compute σ_d can produce a better estimation of its VaR.

Volatility is an input parameter in many financial models. For example, Modern Portfolio Theory (MPT) uses volatility (in the form of standard deviations) to compute the market portfolio. The Market portfolio is an optimal portfolio in a sense of reducing diversifiable risk and maximizing expected return. MPT is widely used in asset allocation decision making by many FI.

As we can see, volatility is used in many aspects of, among others, financial risk management, investment decisions, regulatory policies and capital budgeting.

Forecasting financial volatility

Formally, forecasting volatility could be seen as finding such $\hat{\sigma}_t$ that will minimize the error $\varepsilon = f(\sigma_t - \hat{\sigma}_t)$, where σ_t is an actual (or observed) volatility over period t and $f(\cdot)$ is an error function. Discussion of different forms of error functions can be found below.

Volatility can be forecasted over different time ranges. Typically, it is divided into one day volatility, 10 day volatility, monthly volatility, and bigger time frames. Daily volatility is usually used for computing risk metrics (e.g. DEaR). 10 day volatility is most often used in risk management; under Basel II regulatory requirements a Financial Institution should track its 10-day VaR. Monthly and bigger timeframes could be used for options evaluation and in portfolio management. The methods, which are described in this work, do not depend on the choice of a timeframe. The forecasting performance of methods could, however, vary on different timeframes.

To estimate volatility on a certain timeframe one could use data of a smaller timeframe and compute the standard deviation. For example, if we are interested in monthly volatility, we can compute the standard deviation of daily returns. Sometimes it is difficult to find data for a smaller timeframe, in which case different methods for volatility estimation can be used. The simplest way of estimating

volatility is taking daily squared returns. Unfortunately, this method gives an inaccurate estimation of volatility (Taylor, 1986). Daily returns are calculated using the closing price of an asset in the end of a trading session. The intraday jumps of the prices are not reflected in daily returns. These jumps can be significant and can influence the estimate of volatility.

In order to estimate the forecasting performance of some methods or to compare several methods we should define error functions. The following are the most used error functions:

1. Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{\sigma}_t - \sigma_t)^2}$$

2. Mean Absolute Error (MAE):

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{\sigma}_t - \sigma_t|$$

3. Mean Absolute Percent Error (MAPE):

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|\hat{\sigma}_t - \sigma_t|}{\sigma_t}$$

4. Mean Error (ME):

$$MA = \frac{1}{N} \sum_{i=1}^N (\hat{\sigma}_t - \sigma_t)$$

5. Mean Square Error (MSE):

$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{\sigma}_t - \sigma_t)^2$$

6. Mean Logarithm of Absolute Error (MLAE):

$$MLAE = \frac{1}{N} \sum_{i=1}^N \ln |\hat{\sigma}_t - \sigma_t|$$

There are other error measures (e.g. Theil-U or LINEX loss function) but they are less intuitive and infrequently used (Poon, 2005).

There are many volatility models and forecasting methods. An overview of some of them will be given in the next chapters. All these models can work in

different time frames and on different markets. Historically most of these models were created for stock market analysis, but they can be easily applied to the analysis of commodities, foreign exchanges and other markets.

Historical volatility models

Historical volatility models (HIS) are one of the simplest classes of volatility models. The name, HIS, is different in different literature, but most often it is used to stress that these models differ from the implied volatility models. In this chapter the following models will be discussed: constant volatility model, simple moving average, exponential smoothing, and exponential weighted moving average.

Historical average model (HAM) is the simplest model. $\hat{\sigma}_{t+1}$ is a mean standard deviation calculated over some time interval and then used to forecast future values.

$$\hat{\sigma}_{t+1} = \frac{1}{t} \sum_{i=1}^t \sigma_i$$

Off course this method produces poor results, but it can be used as a quick method. The error of this method can be used as a benchmark for other methods.

Simple moving average (SMA) is a more advance method than HAM. This method uses the most recent information to build a prediction. Under this method the forecasted volatility at time $t + 1$ is computed using the following formula:

$$\hat{\sigma}_{t+1} = \frac{1}{\tau} \sum_{i=0}^{\tau-1} \sigma_{t-i}$$

where $\tau < t$. In this method information older than τ is not taken into consideration. The parameter τ can be arbitrary or taken in such a way that the error $\varepsilon = f(\sigma_t - \widehat{\sigma}_t)$ is minimal on some training set.

Exponential smoothing (ES) is another method to compute $\hat{\sigma}_{t+1}$ based on historical values. This method is described by the following formulas:

$$\hat{\sigma}_{t+1} = (1 - \beta)\sigma_t + \beta\hat{\sigma}_t$$

where $\hat{\sigma}_0 = \sigma_0$, and smoothing parameter β : $0 \leq \beta \leq 1$ are found by minimizing in-sample forecast error ξ_t of the relation $\sigma_t = (1 - \beta)\sigma_{t-1} + \beta\hat{\sigma}_{t-1} + \xi_t$. This method, unlike SMA, gives more weight to the recent volatility.

Exponential weighted moving average (EWMA) is another HIS volatility model. This moving average method is represented by the following formula:

$$\hat{\sigma}_{t+1} = \frac{\sum_{i=0}^{\tau-1} \beta^i \sigma_{t-i}}{\sum_{i=0}^{\tau-1} \beta^i}$$

Again the smoothing parameter β is estimated by minimizing the error on a training set. The JP Morgan *Riskmetrics* model is a procedure that uses Exponentially Weighted moving Average.

Black-Scholes formula and implied volatility

Implied volatility models are another important class of volatility models. A number of definitions should be introduced first.

European call option is a financial contract that gives the holder the right but not an obligation to buy an underlying asset at a certain date (expiry date) for a certain price (exercise or strike price).

European put option, unlike call option, gives to its holder right to sell an underlying asset at a certain date for a certain price.

In early 1970s, Fischer Black, Myron Scholes and Robert Merton made a major breakthrough into stock option pricing (Hull, 2002). This involved the development of what became known as the Black-Scholes model. In 1997 Myron Scholes and Robert Merton received a Nobel Prize in economics for their contribution in derivatives pricing. Let us introduce the assumptions and some important results of this model. The original assumptions are (Black & Scholes, 1973):

- The underlying stock price (S) is describe by the following process:
 $\frac{dS}{S} = \mu dt + \sigma dW$, where μ is expected rate of return (the drift), σ is the constant volatility of the returns, W is a Brownian Motion.
- There are no transaction costs and all securities are perfectly divisible.
- There are no arbitrage opportunities.
- The risk free interest rate, r , is constant and the same for all maturities. Investors can freely borrow and lend money for the risk free interest rate.

The assumptions of the Black-Scholes model are rather strong and unrealistic. Extensions of the Black-Scholes model manage to overcome most of these

restrictions (Wilmott, Howison, & Dewynne, 1995). Still the assumption of constant volatility σ is one of the strongest.

Let us denote price of a European call option as C , P – European put and K strike price. Then C and P could be found by the following formulas:

$$C = S_0 e^{-qT} N(d_1) - K e^{-rT} N(d_2)$$

$$P = K e^{-rT} N(-d_2) - S_0 e^{-qT} N(-d_1)$$

where $d_1 = \frac{\ln(S_0/K) + (r - q + \sigma^2/2)T}{\sigma\sqrt{T}}$

$$d_2 = d_1 - \sigma\sqrt{T}$$

$N(x)$ - is the cumulative probability distribution function for standard normal distribution.

S_0 - is a price of underlying asset at time $t = 0$.

q - is a dividend rate.

Other types of options (American, Asian, etc.) can be priced using the Black-Scholes partial differential equation. The Black-Scholes model and its extensions are widely used in practice. The one parameter of these formulas that cannot be directly observed is the volatility; **implied volatility** is used as a proxy. Implied volatilities are the volatilities implied by the market prices of the options. Let us assume that we have the market prices of call and put options for different maturities and all other parameters are known (except volatility). Using those prices we can compute Black-Scholes formulas by working ‘backwards’ and estimate the volatility. This volatility will be the implied volatility. Unfortunately there is no direct formula for computing the implied volatility (Hull, 2002). However, a method of trial-and-error can be introduced which allows us to compute implied volatility with a good accuracy.

The implied volatility of options of different maturities has an interesting characteristic. There is a pattern that implied volatility is not constant for different strike prices. This pattern is called *volatility skew* or *volatility smile*. An example of this pattern can be found in figure 1.

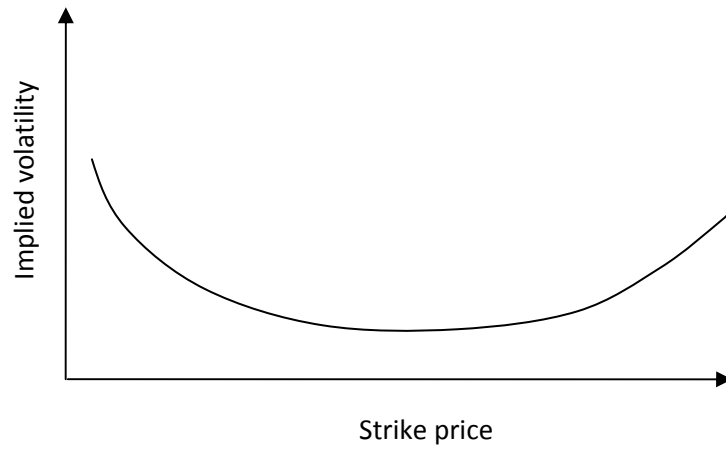


Figure 1. A common shape of the volatility skew pattern. A non linear relationship between the moneyness of the option and it's implied volatility.

The volatility skew is used by the investors to price options in a foreign currency market and the equity options market. Investors typically calculate implied volatility for actively traded options and then use the volatility smile to price more exotic options. Nowadays, implied volatility is an important market indicator. It helps to price financial derivatives and is an overall indicator of investors 'moods'.

Autoregressive and Heteroskedastic models

Financial market volatility is known to cluster (Tsay, 2005). A highly volatile period tends to persist for some time before the market returns to a more stable environment. An autoregressive approach helps to build more accurate and reliable volatility models.

The **Autoregressive Moving Average (ARMA)** model is a combination of the SMA model and Autoregressive model. It consists of two components and can be expressed by the following formula:

$$\hat{\sigma}_{t+1} = \sum_{i=1}^p \beta_i \sigma_{t+1-i} + \sum_{i=1}^q \gamma_i \xi_{t+1-i}$$

where $\beta_1, \beta_2, \dots, \beta_p$ and $\gamma_1, \gamma_2, \dots, \gamma_q$ are parameters of the model and $\xi_1, \xi_2, \dots, \xi_t$ are the error terms of the model $\sigma_i = \hat{\sigma}_i + \xi_i$, $i = \overline{1, t}$. We will refer to the first model as the $ARMA(p, q)$ model. The length of the moving average (p) and autoregressive term (q) as well as the parameters could be found by minimizing the error on the testing set. The ARMA model is a popular model for forecasting not only in financial markets but in many other applied areas as well.

The **Autoregressive Conditional Heteroskedasticity (ARCH)** model was first introduced by Engle in 1982 (Engle, 1982). ARCH model and its extensions (GARCH, EGARCH, etc.) are among the most popular models for forecasting market returns and volatility. Originally, the ARCH model rather than using standard deviations used the variance. Let us call the variance of the returns σ^2 as h . The ARCH model can be defined as following:

$$\begin{aligned} r_t &= \mu + \varepsilon_t \\ \varepsilon_t &= \sqrt{h_t} z_t \\ h_t &= \varpi + \sum_{j=1}^q \alpha_j \varepsilon_{t-j}^2 \end{aligned}$$

where r_t - return at time t

μ - mean return

ε_t - residuals (or error terms)

$z_t \sim iid N(0,1)$ normally distributed random variable

ϖ and $\alpha_1, \alpha_2, \dots, \alpha_q$ - parameters of the model

The process z_t is scaled by h_t , the conditional variance, which follows an autoregressive regression process. The parameters $\varpi > 0, \alpha_j \geq 0$ insure that the variance h_t is positive. The one step ahead forecast is simply the square root of the variance $\hat{\sigma}_{t+1} = \sqrt{h_{t+1}}$. The parameter q could be estimated by minimizing the error on a particular training set. The discussion of theoretical properties of the ARCH process could be found in (Tsay, 2005).

The name ARCH refers to this structure: the model is autoregressive, since ε_t clearly depends on previous ε_{t-i} , and conditionally heteroscedastic, since the conditional variance changes continually.

The **Generalized Autoregressive Conditional Heteroskedasticity (GARCH)** is a general version of the ARCH model. It differs from ARCH by the form of h_t . Formally, the $GARCH(p,q)$ model can be defined as following:

$$\begin{aligned} r_t &= \mu + \varepsilon_t \\ \varepsilon_t &= \sqrt{h_t} z_t \end{aligned}$$

$$h_t = \varpi + \sum_{i=1}^p \beta_i h_{t-i} + \sum_{j=1}^q \alpha_j \varepsilon_{t-j}^2$$

where r_t - return at time t

μ - mean return

ε_t - residuals (or error terms)

$z_t \sim iid N(0,1)$ - normally distributed random variable

$\varpi, \alpha_1, \alpha_2, \dots, \alpha_q, \beta_1, \beta_2, \dots, \beta_p$ - parameters of the model

As before, parameters $\varpi > 0, \beta_i \geq 0, \alpha_j \geq 0$ are positive. There are additional constraints on β_i, α_j for models with higher orders than $GARCH(1,1)$ (Tsai, 2006). An optimization procedure can be introduced to find the parameters p and q . As in the *ARCH* model, at time t all the parameters are known, and h_t can be easily computed. The one-step ahead forecast of volatility is again, just $\hat{\sigma}_{t+1} = \sqrt{h_{t+1}}$.

There are a number of extensions of the GARCH model, such as Integrated GARCH, Exponential GARCH, GJR- GARCH and others (J. Knight, 2007). The GARCH family of models is widely used in practice for prediction of financial market volatility and returns.

Models based on artificial neural networks

Artificial Neural Networks (or Neural Networks) are popular statistic techniques for machine learning. Originally, they were created as an attempt to model the biological neuron system. This attempt was made to create a new approach to the computing, and to possibly mimic the behavior of a human brain. This field of science was created in a late 1950s, and was extensively developed in 1980s. Some definitions have to be given, before introducing the volatility model based on the Neural Networks.

Informally, Neural Network is a black box which takes some variables as an input, and gives other as an output. The network consists of neurons that are connected with each other. A neuron transforms the input information according to some rule and propagates the result further to other neurons. The Neural Network can be considered as an algorithm for an approximation of unknown functions. A number

of definitions should be introduced first. A reader who is not interested in the technical details can skip the following definitions.

The Neural Networks are not a new technique for the financial applications. They were successfully used for the prediction of the stock/index returns (White, 1988), (Sharda, 1990), (Kimoto, 1990), (Brown, 1998), (Gencay, 1998); the economic time series forecasting (Swanson, 1995), (Zhang, 1994); interest rate prediction (Swanson, 1995), (Kim S. H., 1997); earnings prediction (Charitou, 1996); fraud detection (Fanning, 1998); bond risk analysis (Dutta, 1988), (Kim J. W., 1997), (Maher, 1997); and other fields.

There is a number of Neural Network models used for volatility forecasting. The (Donaldson, 1997) proposes a Neural Network modification of a GARCH model. In this paper a multilayer perception and density estimating neural network is used to forecast the returns and volatility of the DAX stock index. This paper shows that Neural Networks can be successfully used for volatility forecasting. In the (Gonzales, 1997) authors are investigating the use of Neural Networks for predicting volatility for Ibex35 index. This paper shows advantages of the non-linear Neural Networks over the linear models for volatility forecasting. This paper describes a successful application of Neural Networks for volatility forecasting on an hourly basis. The (Hamid, 2004) describes the usage of a feed forward Multilayer Perceptron for forecasting the volatility of S&P 500 futures. The network uses 13 different financial indicators as inputs, and gives the forecast for futures of different maturities. This paper had shown the advantages of the Neural Networks over the implied volatility.

Forecasting accuracy of the Neural Networks compared to other methods is different in the literature. The performance of the networks is heavily influenced by the choice of the testing asset, the timeframe, the type and the number of the inputs. However, the majority of papers are suggesting better forecasting accuracy of NN compared to the linear algorithms. The Neural Network model described in this paper is a generalization of approaches suggested in the literature. It describes the general framework for using the feed-forward NN in one period ahead volatility forecasting. We do not make any constraints on the number of input parameters and their nature. We implicitly specify the optimal number of hidden layers and the reason for this (The Universal Approximation Theorem).

A detailed description of the volatility models based on Neural Networks can be found in Appendix A.

Comparison of volatility forecasting models

The empirical tests of the volatility models were performed on the volatility of the S&P 500 index. This index includes 500 stocks of US companies with the largest capitalization; it is widely diversified and actively traded on the markets. All the models are tested using the monthly volatility which is constructed from the daily returns. We use 360 observations for the tests. The data is divided into two data sets:

- Training set – December 1978 to October 2000 (263 observations);
- Testing set – November 2000 to November 2008 (97 observations).

The parameters of all the models are optimized on a training set; the testing set is used to compare quality of the models. Training and testing sets are shown in Figure 2.

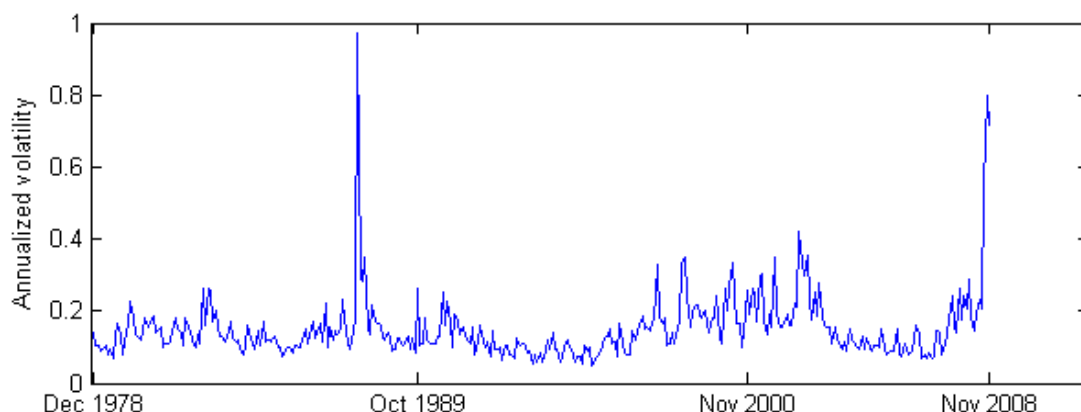


Figure 2. Observed monthly volatility of the S&P 500. Training and testing sets.

The volatility for the data sets was computed using daily data from *finance.yahoo.com*. The volatility was computed in annualized terms (a standard deviation of a daily returns is scaled by multiplying by $\sqrt{252}$). We then used the following error functions: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percent Error (MAPE). All the models performed one-step a head forecast (e.g. forecasting volatility for ‘next’ month).

The following historical volatility models were tested using the testing set: Historical average model, Simple moving average, Exponential smoothing and Exponential weighted moving average (EWMA). The results can be found in Table 1 and Figures 3-4.

Model	RMSE	MAE	MAPE
Historical average	0.1235	0.0718	0.3745
Simple moving average	0.0773	0.0477	0.2580
Exponential smoothing	0.0886	0.0517	0.2660
Exponential weighted moving average	0.0784	0.0473	0.2497

Table 1. Performance of simple models.

Simple moving average was used with $\tau = 2$ previous observations taken into consideration. Exponential weighted moving average had $\tau = 3$. The optimal parameter of Exponential smoothing yielded $\beta = 0.771$ while for EWMA $\beta = 0.744$. All the parameters were found by minimizing RMSE on the training set.

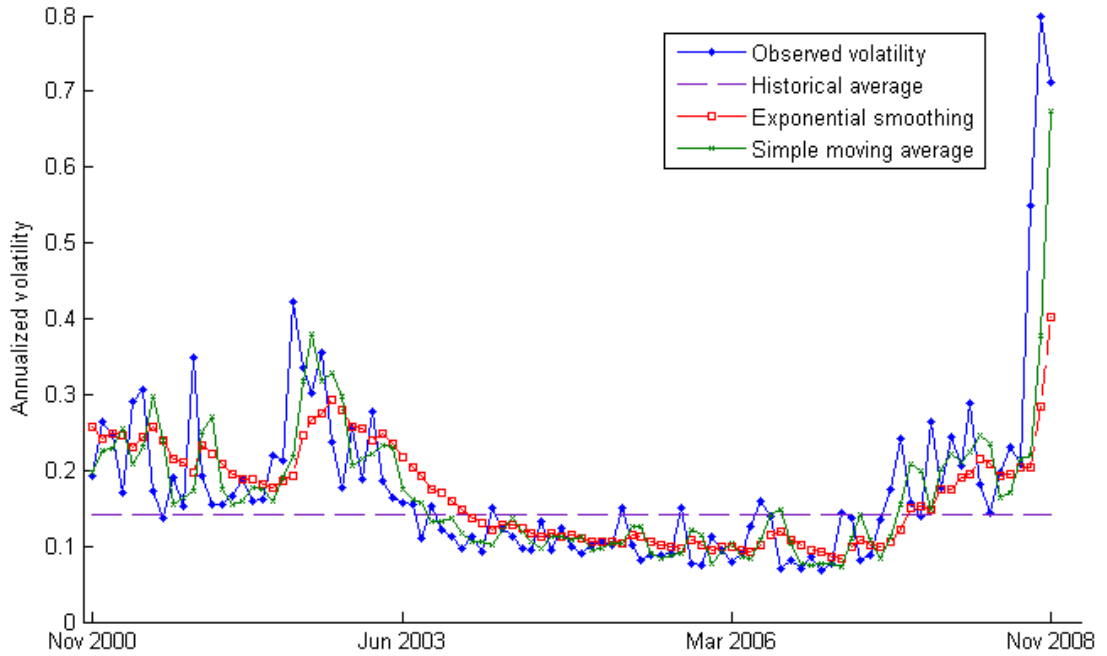


Figure 3. Observed volatility compared to simple volatility models.

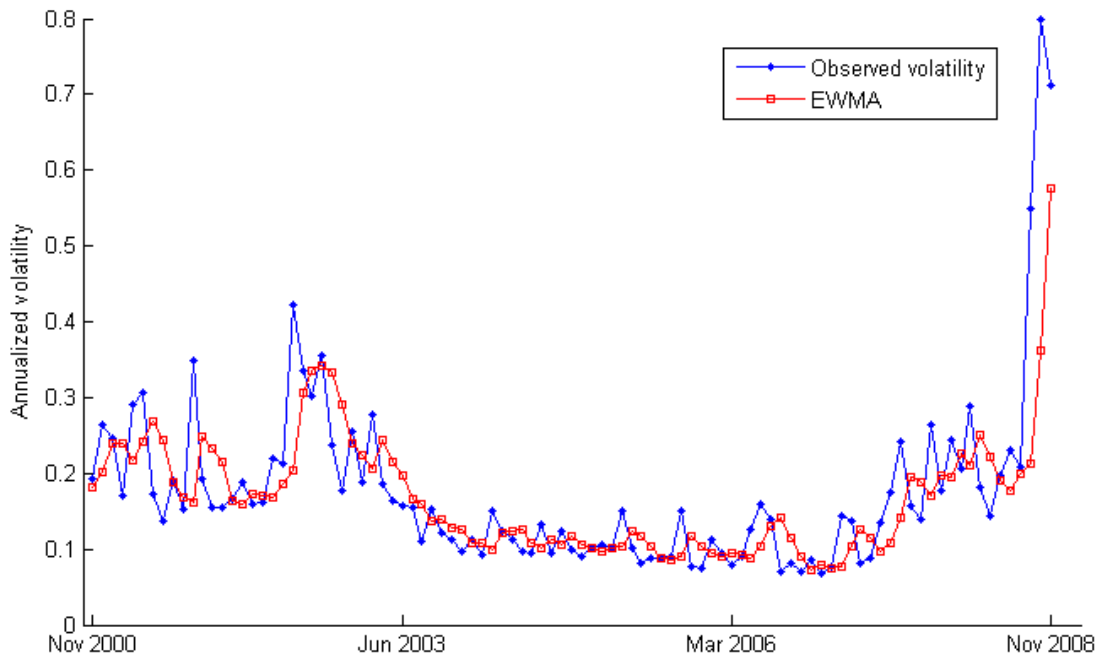


Figure 4. Exponentially weighted moving average and observed volatility over testing set.

Historical volatility models are the simplest class of volatility forecasting methods. It is easy to implement and to optimize these models. At the same time Exponential smoothing and Exponentially weighted moving average give one of the best results among all volatility forecasting models. These two models have the lowest RMSE using the testing set. They do have one disadvantage, however, in that they purely manage to adjust for the sudden jumps or drops in the volatility. It takes some ‘time’ for these models to switch to another volatility cluster.

Implied volatility is an important class of volatility models. The Chicago Board Options Exchange is tracking VIX index – an index of implied volatilities for the S&P500 stock index. VIX is a key barometer of volatility for the S&P 500. VIX is calculated as a weighted average of implied volatilities of call and put options of the S&P 500 index with different maturities. The main goal of VIX is to estimate 30 days implied volatility. We are using two versions of the implied volatility: VIX and adjusted VIX. The adjustment is made by multiplying the value of VIX by $\sqrt{22}$ and dividing by $\sqrt{30}$. This adjustment is done in order to switch to daily volatility and then scale it back by the average number of trading days in a month (22). The results of these methods are presented in Table 2 and Figure 5.

Model	RMSE	MAE	MAPE
VIX	0.0792	0.0546	0.3610
VIX adjusted	0.0783	0.0439	0.2396

Table 2. Implied volatility performance.

The implied volatility is a popular method for measuring volatility. It gives good results in terms of the error. The ‘shape’ of the implied volatility tends to mimic the realized volatility. The main disadvantage of this method is that on average it tends to overestimate the observed volatility. Implied volatility reflects the ‘fears’ of market players. Another disadvantage is that for some assets this method is hard to implement as there are no options traded or traded options are not liquid. Off course, this is never the case with the S&P 500.

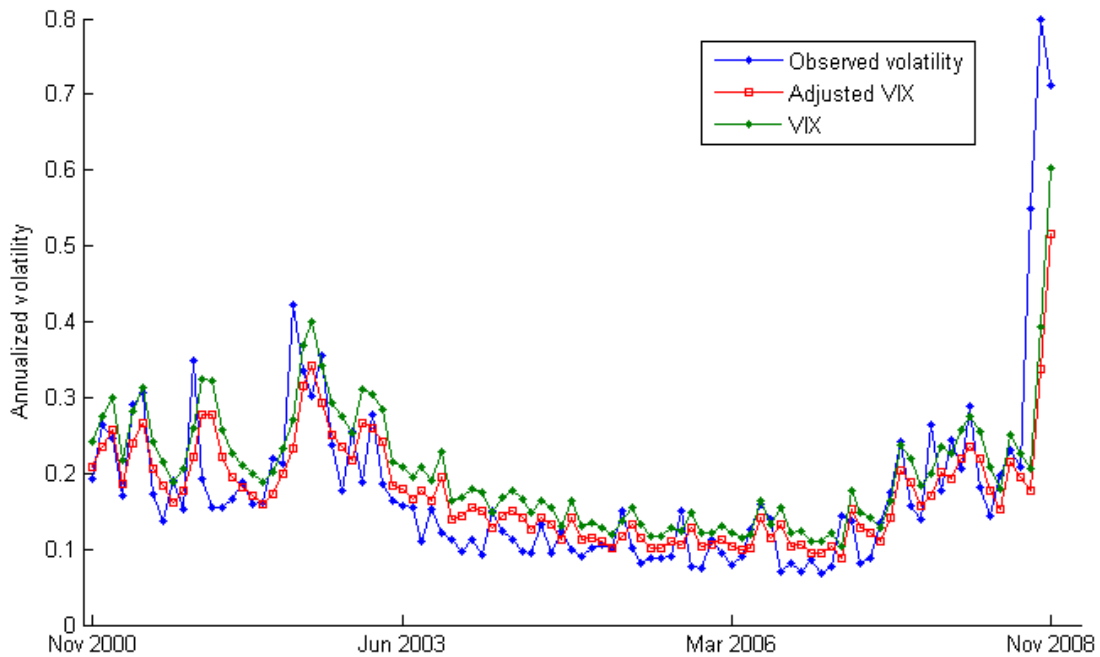


Figure 5. Index of implied volatility VIX and its adjusted version in comparison with the observed volatility.

We have also tested a number of Autoregressive and Heteroskedastic models. Their performance could be found in the Table 3. The parameters of the models were optimized using the Gradient descent algorithm.

Model	RMSE	MAE	MAPE
ARMA(1,1)	0.0890	0.0505	0.2597
ARMA(5,1)	0.0871	0.0503	0.2620
ARCH(1)	0.1029	0.0905	0.7045
GARCH(1,1)	0.1014	0.0857	0.6104
GARCH(2,2)	0.1011	0.0856	0.6117

Table 3. Performance of ARMA and ARCH/GARCH methods

These methods are complex to implement. In addition they yield a poor level of accuracy relative to their complexity and theoretical background. The main disadvantage of these models is their complex procedure of parameter estimation.

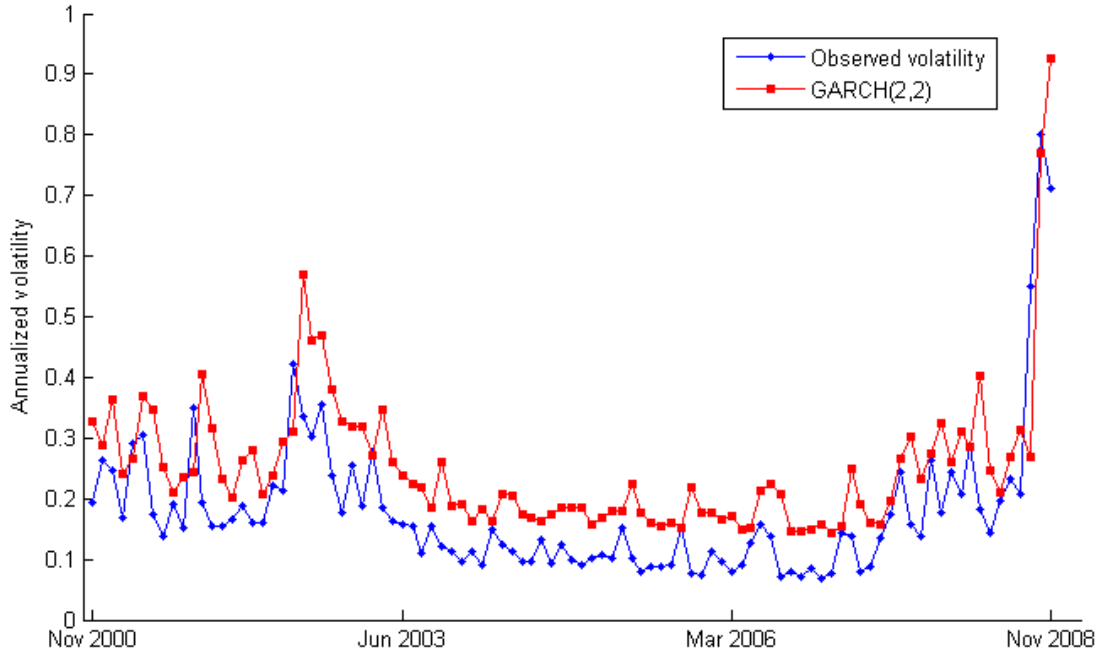


Figure 6. GARCH(2,2) performance on the testing set.

The Neural Network model was proposed to improve on the forecasting performance of ‘classical’ methods. We have tested two architectures of NN for volatility forecasting. Setups of these models are presented in Table 4.

Neural networks are non linear adaptive algorithms. For a given setup and data set they can give different results after training. In order to produce an accurate estimate, we calculate errors on the average forecast produced by 10 networks of the same architecture. An average error is presented in Table 5.

Architecture	NN_1	NN_2
Type	Multi Layer Perceptron	
Number of hidden layers	1	
Activation function	Linear	
Number of neurons in hidden layer	20	10
Input	30 input neurons: 10 previous observations of S&P 500 returns, 10 previous observations of S&P 500 volatility, 10 previous observations of S&P 500 trading volume	8 input neurons: 2 previous observations of S&P 500 returns, 2 previous observations of S&P 500 volatility, 2 previous observations of S&P 500 trading volume , 2 previous VIX value
Number of epochs during training	2000	
Learning algorithm	Scaled conjugate gradient optimization	

Table 4. Two networks for volatility forecasting

Model	RMSE	MAE	MAPE
NN_1	0.0919	0.0527	0.2734
NN_2	0.0820	0.0435	0.2029

Table 5. Error of mean forecast produced by two neural network models

Neural networks generate small forecasting errors. They capture non-linear dependences in the index reruns and volatility clusters. The main disadvantage of the Neural Network approach is that the network can only ‘learn’ patterns that appeared in the past. This approach requires a skilled tuning of the parameters.

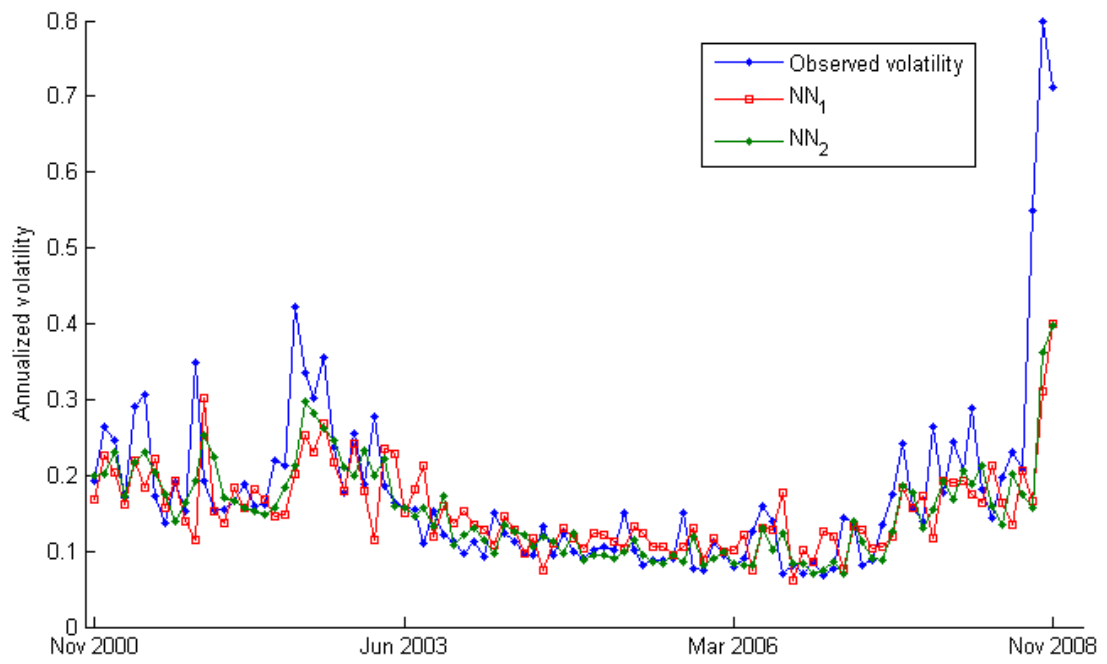


Figure 7. One period forecast of volatility of two implementations of the models based on Neural Networks.

The comparison of the volatility forecasting models is summarized in Table 6.

Class of the models	RMSE	MAE	MAPE
Historical volatility models			
Historical average	0.1235	0.0718	0.3745
Simple moving average	0.0773	0.0477	0.2580
Exponential smoothing	0.0886	0.0517	0.2660
Exponential weighted moving average	0.0784	0.0473	0.2497
<i>Advantages:</i> Easy to use, relatively good results (except historical average).			
<i>Disadvantages:</i> after sudden price movements these models tend to significantly overestimate the volatility			
Implied volatility			
VIX	0.0792	0.0546	0.3610
VIX adjusted	0.0783	0.0439	0.2396
<i>Advantages:</i> Good results, these model are based on strong theoretical results.			
<i>Disadvantages:</i> complex to implement; not suitable for all products (only for that, which are an underlying for an actively trading options); are dependent on sometimes irrational expectations of investors.			
Autoregressive and Heteroskedastic models			
ARMA(1,1)	0.0890	0.0505	0.2597
ARMA(5,1)	0.0871	0.0503	0.2620
ARCH(1)	0.1029	0.0905	0.7045
GARCH(1,1)	0.1014	0.0857	0.6104
GARCH(2,2)	0.1011	0.0856	0.6117
<i>Advantages:</i> an existence of a big number of theoretical researches of this models;			
<i>Disadvantages:</i> moderate forecasting accuracy; complex to implement and optimize;			
Models based on Neural Networks			
NN_1	0.0919	0.0527	0.2734
NN_2	0.0820	0.0435	0.2029
<i>Advantages:</i> relatively good results; an ability to build models which use not only historical returns as an input, but also other related financial time series and variables.			
<i>Disadvantages:</i> complex to implement and to find suitable architecture; can forecast only dependencies from a previous observations			

Table 6. A summary of the results

Conclusions and Perspectives

This project is dedicated to addressing the problem of forecasting volatility in financial markets. We have selected several methods that are heavily used in practice and tested their accuracy using real data (i.e., S&P 500 stock index). Each family of methods has its advantages and disadvantages, which are described in this work. Some methods are simple but yield poor results (e.g., historical average model). Other methods provide improved results but are difficult to implement (e.g., Implied Volatility method). In short, there is no single perfect approach. Nevertheless, we found that the Exponentially Weighted and Simple Moving Average methods are both efficient and are relatively easy to implement. Our results are also consistent with those published by other researches who examined these methods (McNeil, Frey, & Embrechts, 2005). We suggest that Moving Average can be used for a quick approximation of the volatility forecast. Although it can give a good initial benchmark of a forecast, the final estimations should always rely on several models. We have also tested models based on Neural Networks. This is a relatively new class of models, and we confirmed that this class of models can be successfully used for volatility forecasting.

A logical continuation of this work would be to combine several volatility-forecasting models into a single predictor. Application of such a predictor can potentially overcome the disadvantages of individual models and provide the best forecast. The predictor can be viewed as a linear combination of outputs of selected best models:

$$\hat{\sigma}_{t+1} = \alpha_0 + \alpha_1 P_1(d_t, \gamma_1) + \alpha_2 P_2(d_t, \gamma_2) + \dots + \alpha_N P_N(d_t, \gamma_N)$$

where P_i - models for volatility forecasting (e.g. SMA, EWMA, GARCH, VIX, etc.)

α_i – weight of the i -th model

γ_i – optimal set of parameters and their values utilized by P_i

d_t – available market data at time t

The practical application of the outlined approach, however, will require further theoretical and empirical testing that goes beyond the scope of this paper.

Bibliography

- Black, F. (1976). Studies of stock price volatility of changes. *American Statistical Association Journal* , 177-181.
- Black, F., & Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *Journal of Political Economy* , 637–654.
- Brown, S. G. (1998). The Dow theory: William Peter Hamilton's track Record Reconsidered. *Journal of Finance* , 1311-1333.
- CBOE. <http://www.cboe.com/micro/vix/faq.aspx#2>.
- Charitou, A. a. (1996). The Prediction of Earnings Using Financial Statement Information: Empirical Evidence using Logit Models & Artificial Neural Networks. *International Journal of Intelligent Systems in Accounting, Finance & Management* .
- Cybenko, G. (1989). Approximation by superposition of a sigmoidal functions. *Mathematics of Control, Signals and Systems* , 303-314.
- Donaldson, R. G. (1997). An Artificial Neural Network- GARCH Model for International Stock Return Volatility. *Journal of Empirical Evidence* , 17-46.
- Dutta, S. a. (1988). Bond Rating: A Non-conservative Application of Neural Networks. *Proceedings of the IEEE International Conference on Neural Networks* , 443-450.
- Engle, R. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 987–1007.
- Fanning, K. M. (1998). Neural Network Detection of Management Fraud Using Published Financial Data. *International Journal of Intelligent Systems in Accounting, Finance & Management* .
- Gencay, R. (1998). The Predictability of Security Returns with Simple Technical Trading Rules. *Journal of Empirical Finance* , 347-359.
- Gonzales, M. F. (1997). Modeling Market Volatilities: The Neural Network Perspective. *European Journal of Finance* , 137-157.
- Hamid, S. A. (2004). *Primer on using Neural Networks for forecasting market variables*. New Hampshire: New Hampshire University.
- Haykin, S. (2003). *Neural Networks: A comprehensive foundation* . New Jersey: Prentice Hall.
- Hull, J. C. (2002). *Options, futures and other derivatives*. New Jercey: Prentice Hall.

- J. Knight, S. S. (2007). *Forecasting volatility in the financial markets*. London: Elsevier.
- Kim, J. W. (1997). Expert System for Bond Rating: A Comparative Analysis of Statistical, Rule-based, and Neural Network Systems. *Expert Systems* , 167-171.
- Kim, S. H. (1997). Predictability of Interest Rates Using Data Mining Tools: A Comparative Analysis of Korea and the US. *Expert Systems with Applications* , 85-95.
- Kimoto, T. A. (1990). Stock Market Prediction System with Modular Neural Networks. *Proceedings of the International Joint Conference on* .
- Kriesel, D. (1998). *A brief Introduction on Neural Networks*.
- Maher, J. J. (1997). Predicting Bond Ratings Using Neural Networks: A Comparison with Logistic Regression. *International Journal of Intelligent Systems in Accounting, Finance & Management* .
- McNeil, Frey, & Embrechts. (2005). *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton University Press.
- Nelson, D. a. (1992). Inequality constraints in the univariate GARCH model. *Journal of Business and Economic Statistics* , 229-235.
- Poon, S.-H. (2005). *A practical guide to forecasting financial market volatility*. Chichester: Jhon Wiley and Sons, ltd.
- Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Cornell Aeronautical Laboratory, Psychological Review* , 386-408.
- Sharda, S. A. (1990). Neural Networks as Forecasting Experts: An Empirical test. *Proceedings of the International Joint Conference on Neural Networks* , 491-494.
- Swanson, N. R. (1995). A model selection approach to assessing the information in the term structure using linear models and artificial neural networks. *Journal of Business and Economic Statistic* , 265-275.
- Taylor, S. (1986). *Modelling Financial Time Series*. Chichester: JohnWiley&Sons Ltd.
- Tsai, H. (2006). *A Note on Inequality Constraints in the GARCH model*. Iowa: The University of Iowa.

Tsay, R. S. (2005). *Analysis of Financial Time Series*. Wiley-Interscience; 2nd edition .

White, H. (1988). Economic Prediction Using Neural Networks: The Case of IBM Daily Stock. *Proceedings of the IEEE International Conference on Neural Networks* , 451-458.

Wilmott, P., Howison, S., & Dewynne, J. (1995). *The Mathematics of Financial Derivatives*. Cambridge: Cambridge University Press.

Zhang, X. (1994). Non-linear Predictive Models for Intra-day Foreign Exchange Trading. *International Journal of Intelligent Systems in Accounting, Finance and Management* , 293-302.

Appendix A

A *Neural Network* (NN) is a sorted triple (N, V, w) . Here N is a set of *neurons* and $V = \{(i, j) | i, j \in \mathbb{N}\}$ is a sorted set of *connections* between the neurons, while the function $w: V \rightarrow \mathbb{R}$ defines the *weights* of the connections. $w((i, j))$ is a weight of a connection between neuron i and j , it can be also denoted as w_{ij} .

A neuron is a computational unit, which transforms the input information based on some rule. The neuron is characterized by this ‘rule’ which is described by the propagation and activation functions.

Let $I = \{i_1, i_2, \dots, i_n\}$ be the set of neurons such that $\forall z \in \{1, \dots, n\} \exists w_{i_z j}$ and $\{o_{i_1}, o_{i_2}, \dots, o_{i_n}\}$ be the set of outputs of neurons I . Then *the propagation function* of the neuron j is defined as:

$$f_{prop}: \{o_{i_1}, o_{i_2}, \dots, o_{i_n}\} \times \{w_{i_1 j}, w_{i_2 j}, \dots, w_{i_n j}\} \rightarrow net_j$$

Most often the propagation function is a weighted sum of the neuron’s inputs

$$net_j = \sum_{i \in I} o_i w_{ij}$$

A neural network is a discrete time model. In this chapter by *time* I mean the step t of the work of a neural network. It should be not confused with the time scale of the financial time-series.

The activation state of a neuron is inspired by the properties of the biological neuron. It can be in an “active state” when the electric impulse is running through it. Let j be a neuron, *the activation state* $a_j(t)$ is a level of activation, explicitly assigned to j . *The activation function* transforms the activation state of the neuron from the previous state $a_j(t-1)$ to a new one $a_j(t)$. The activation function is defined as: $\varphi: net_j \times a_j(t-1) \rightarrow a_j(t)$. The activation function can be a binary threshold, step function, sigmoid function, Gaussian function or identical function among others. For example, a binary threshold function is:

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ -1 & \text{if } v < 0 \end{cases}$$

The output function of a neuron $f_{out}: a_j \rightarrow o_j$ calculates the output value based on the activation state. Often, the output function is taken as an identity, so $o_j = a_j$.

A *layer* of a NN is a group of neurons which have a common property. Usually, a layer refers to topological similarities of neurons. The *network topology* is

defined by the way the neurons are connected. The most popular topologies are a *feed forward* and a *recurrent* topology. In a feed forward network there is one input layer of neurons, n processing layers and one output layer. The neurons of each layer are connected only with the neurons of a following layer. The layers are clearly separated in this topology. A recurrent network is characterized by more complex connections between the neurons. The layers could be connected with following as well as previous layers.

In general, the *learning* of a neural network can be described as a changing of the weights w of a network to adopt its outputs to certain inputs. One of the concepts of machine learning is so called supervised learning. Under supervised learning the set of training patterns is given. Each pattern $p \in P_t$ consists of the inputs and the desired outputs $p = \{x; y\}$. By *training* of the neural network we will call the procedure of finding such weights w^* that will minimize the error function between outputs of a network and desired outputs.

$$w^* = \underset{w \in W}{\operatorname{argmin}} \operatorname{Err}(o_{\text{net}}(x), y)$$

where $\operatorname{Err}(\cdot, \cdot)$ is a given error function, $o_{\text{net}}(x)$ is an output of the network for an input x , y is a desired output, W is a set of all possible combinations of weights.

An *Input neuron* is an identity neuron. It forwards the signal received. An *information processing neuron* (or *information processing unit*) is a neuron that changes the input information according to some rule. For example, a binary neuron sums up the weighted input, and then applies a binary activation function. A linear activation function is actually the identity activation function. An *information processing neuron* with this activation function generates the weighted sum of the input signals as an output. An example of an information processing neuron is given on Figure 8. The layer of information processing neurons is called trainable layer.

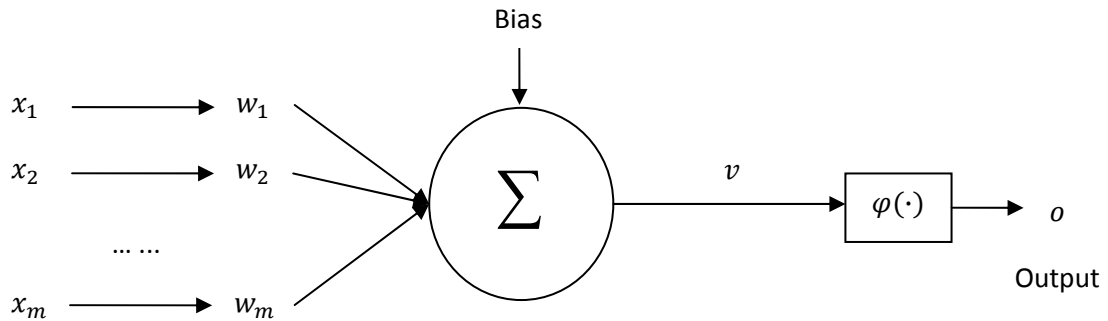


Figure 8. A scheme of a simple information processing neuron.

A *Perceptron* is a feed forward neural network which consists of the following layers: one layer of the input layer, and one or more sequentially connected layers of the trainable layers (Rosenblatt F. , 1958). One neuron layer is completely linked with the following layer. The input layer consists of input neurons, and the trainable layers consist of information processing neurons. The information processing units of this network use summation as a propagation function.

A *Multilayer Perceptron* (MLP) is a Perceptron with more than one layer of information processing neurons. An example of a MLP network is shown in Figure 9.

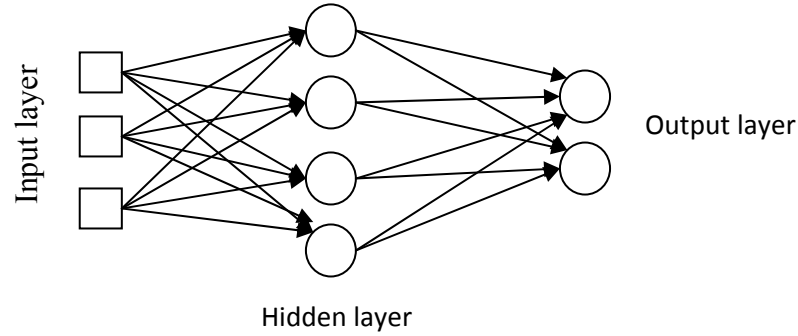


Figure 9. A Multilayer Perceptron with 3 inputs, 4 neurons in a hidden layer, and 2 outputs.

A multilayer Perceptron can be trained with the *Back Propagation algorithm* (Haykin, 2003). We will not go into details of the learning procedure in this paper. For us it is important that there is a procedure of finding the optimal weights w^* .

Another important result from the theory of Neural Networks is *the Universal approximation theorem* (Cybenko, 1989).

Theorem: Let $\varphi(\cdot)$ be the bounded, monotonically increasing and continuous function. Let I_{m_0} be the m_0 dimensional cube $[0,1]^{m_0}$ and $C(I_{m_0})$ is a space of all continuous functions on I_{m_0} . Than $\forall f \in C(I_{m_0})$ and $\forall \varepsilon > 0$ there exists m_1 and $\alpha_i, \beta_j, w_{ij}$, $i = \overline{1, m_1}$, $j = \overline{1, m_0}$ such that the function

$$F(x_1, \dots, x_{m_0}) = \sum_{i=1}^{m_1} \alpha_i \varphi\left(\sum_{j=1}^{m_0} w_{ij} x_j + \beta_j\right)$$

is an approximation of the function f , e.g.

$$|F(x_1, \dots, x_{m_0}) - f(x_1, \dots, x_{m_0})| < \varepsilon$$

for any x_1, \dots, x_{m_0} from an input space.

For the derivation and the discussion of the theorem see (Haykin, 2003), (Cybenko, 1989). As we can see, the theorem suggests that a Multilayer Perceptron with one hidden layer can be used as a universal approximator. This is one of the reasons for building a volatility model based on the MLP.

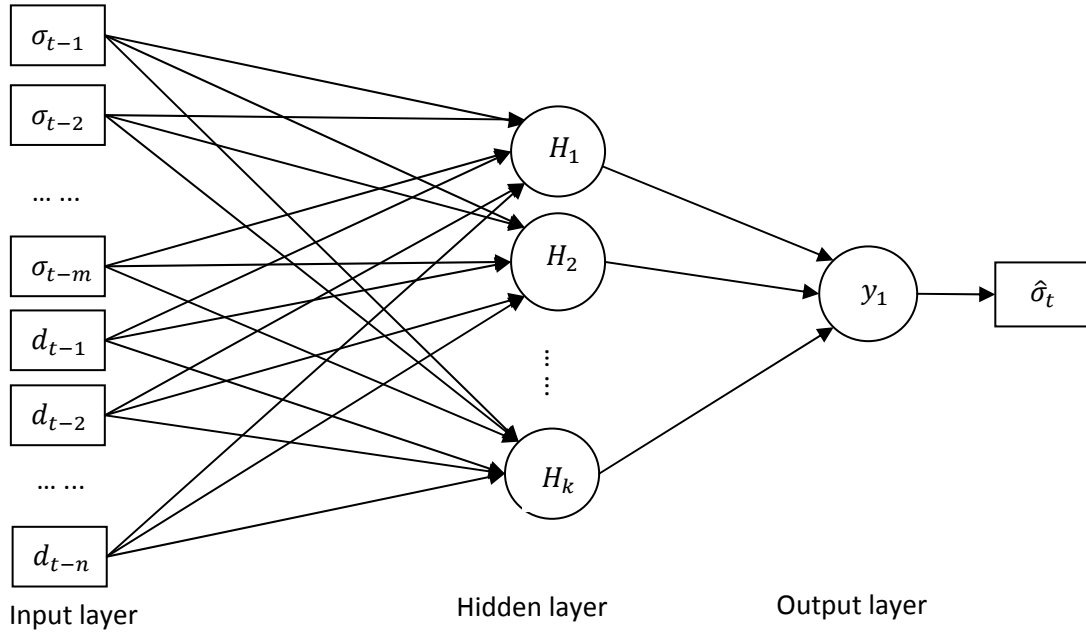


Figure 10. A Multilayer Perceptron with one hidden layer for volatility forecasting. The input consists of previous values of volatility and other market data available on time t . The output of the network is a value of expected volatility.

Volatility can be modeled as a function of previous market data¹: $\sigma_t = \mathcal{F}(\mathcal{D}_{t-1})$ where σ_t is observed volatility and \mathcal{D}_{t-1} is all market data available at time $t - 1$. We will try to approximate the unknown function $\mathcal{F}(\cdot)$ with the known function $F(\cdot)$. Then the model for volatility will be: $\hat{\sigma}_t = F(\mathcal{D}_{t-1})$. The Universal approximation theorem states that such a function $F(\cdot)$ exists and can be found. From the application of the theorem we will take the Multilayer Perceptron with one hidden layer as a realization of $F(\cdot)$.

Let us now describe the architecture of the neural network for volatility forecasting. As was mentioned before, this is a Multilayer Perception (e.g. a feed forward network) with the one hidden layer of the information processing neurons and the one output layer. The architecture of this network is presented in Figure 10.

The input layer consists of the $m + n$ neurons. The output layer is one neuron which corresponds to the forecast. Information processing neurons use summation as a propagation function and an identity as an activation function. This network takes

¹ The assumptions of this model are based on violation of hypothesis of the market efficiency in the weak or the semi-strong form. But this discussion is beyond this paper.

the previously observed volatilities $\sigma_{t-1}, \dots, \sigma_{t-m}$ and other market information d_{t-1}, \dots, d_{t-n} as inputs to produce the forecast $\hat{\sigma}_t$. The structure of the input of the network is flexible: d_{t-1}, \dots, d_{t-n} and could be the previous returns, the trading volumes, any indicators that influence $\hat{\sigma}_t$. In the simplest case d_{t-1}, \dots, d_{t-n} could be dropped. This simplification will not influence the networks architecture.

This network is trained on a previous history of the time series. The training patterns are the following $p_t = \{\sigma_{t-1}, \dots, \sigma_{t-m}, d_{t-1}, \dots, d_{t-n}; \sigma_t\}$ $t = \overline{1, T}$. The training procedure is the same as for MLP network. This type of MLP could be trained using the back propagation algorithm (Kriesel, 1998).