

Forecasting Level And Volatility Of High Frequency Exchange Rate Data

by Sabine Toulson (London School of Economics & Intelligent Financial Systems Limited)

Abstract

This paper compares the performance of a single hidden layer multi-layer perceptron (MLP) with two conventional statistical techniques, a structural time series model (SM) and the stochastic volatility model (SV). A comparison is made between the multi-layer perceptron and the structural model by investigating the predictive power of both models for a one-step ahead forecast of the USD/DEM exchange rate. A further study compares the performance of a multi-layer perceptron and a stochastic volatility model in predicting the volatility of the USD/DEM exchange rate. Finally, a combination of the multi-layer perceptron and the stochastic volatility models is proposed for forecasting volatility. This combination is shown to decrease the overall forecasting error.

1. Overview of the Models

1.1 The Multi-layer Perceptron

Neural networks have recently received considerable attention in the financial community (White (1988), Refenes *et al.* (1993)). The most commonly used neural network architecture is the multi-layer perceptron (Rummelhart *et al.* (1986)) shown in Figure 1.

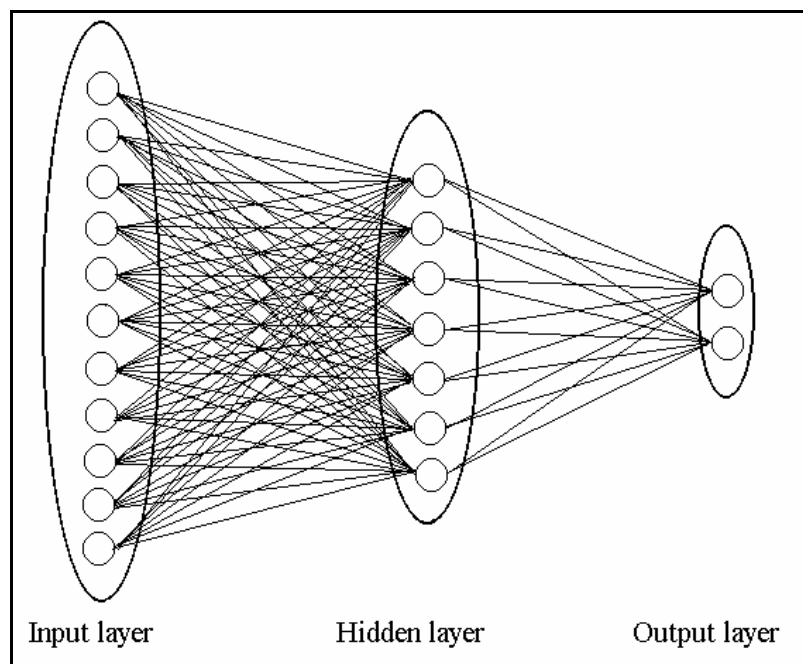


Figure 1: The multi-layer perceptron

The basic building block of the multi-layer perceptron is the artificial neuron shown in Figure 2. A neuron operates by summing the input it receives via weighted links from other neurons and then generates an output value that is a non-linear function of this input activation.

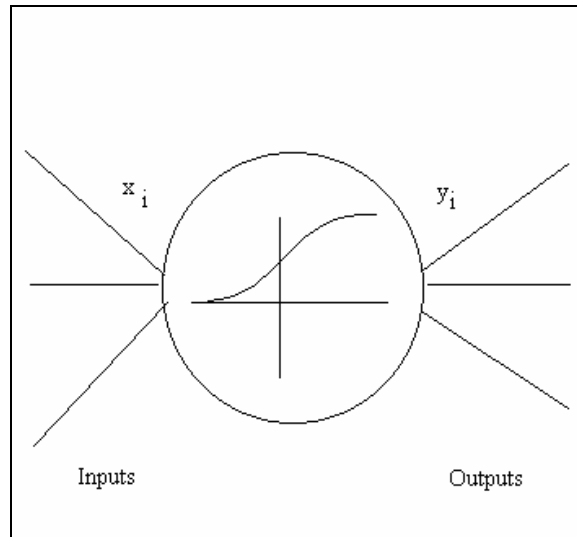


Figure 2: The neuron

Let x_i be the total activation of neuron i , y_i the output of neuron i and ω_{ji} the weighted link between neurons j and i then

$$\begin{aligned} x_i &= \sum_{j=1}^n \omega_{ji} y_j, \text{ with} \\ y_i &= \mathbf{q}(x_i) \end{aligned} \quad (1)$$

where $\theta(\cdot)$ is some non-linear function, usually taken to be sigmoidal,

$$\mathbf{q}(x_i) = \frac{1}{1 + \exp(-x_i)} \quad (2)$$

The multi-layer perceptron is composed of hierarchical layers of such neurons arranged so that information flows from the *input* layer to the *output* layer of the network, i.e. no feedback connections are allowed. For financial applications the multi-layer perceptron may be used in the following way. Lagged observations of the time series are encoded and *fed* into the input layer of the network. These inputs are propagated through a hidden layer to emerge from the final output layer as an encoded prediction

The network is trained using a set of desired input/output vector pairs termed the training set. This process involves the iterative adaptation of the weighted links between neurons to minimise the mean square error between the desired outputs and the actual ones produced by the network. A number of techniques may be used to achieve this. First-order

techniques¹ such as backprop (McClelland *et al.* (1986)) are often used, though in this work a faster second order technique, Quickprop (Fahlman (1988)), was preferred.

To avoid overfitting the data in the learning process a method of concurrent descent is used whereby the training data is split into training and validation sets. Training is halted at the point at which the error of the validation set is minimised. The performance of the network on unseen data is then assessed by applying it to a further test set.

1.2 The Structural Model

Most economic time series are characterised by following a trend, representing the long-term behaviour of the series. Seasonalities or cyclical patterns are often reported in the literature (Harvey (1993) and Taylor (1986)). A model which captures these characteristics is the structural time series model. The structural model used consists of a trend, a cyclical and an irregular (random) component, denoted by \mathbf{m} , \mathbf{y}_t and \mathbf{e}_t , respectively, given by

$$s_t = \mathbf{m}_t + \mathbf{y}_t + \mathbf{e}_t, \quad t = 1, K, T \quad (3)$$

Let \mathbf{h}_t and \mathbf{z}_t be independently distributed disturbance terms of the form:

$$\begin{aligned} \mathbf{h}_t &\sim NID(0, \mathbf{s}_h^2) \\ \mathbf{z}_t &\sim NID(0, \mathbf{s}_z^2) \end{aligned}$$

The trend component is then generated by the process

$$\begin{aligned} \mathbf{m}_t &= \mathbf{m}_{t-1} + \mathbf{b}_{t-1} + \mathbf{h}_t, \quad \text{with} \\ \mathbf{b}_t &= \mathbf{b}_{t-1} + \mathbf{z}_t \end{aligned} \quad (4)$$

Let I_c denote the frequency of a cyclical function with dampening factor \mathbf{r} then the cyclical component can then be written as

$$\begin{pmatrix} \mathbf{y}_t \\ \mathbf{y}_t^* \end{pmatrix} = \mathbf{r} \begin{pmatrix} \cos I_c & \sin I_c \\ -\sin I_c & \cos I_c \end{pmatrix} \begin{pmatrix} \mathbf{y}_{t-1} \\ \mathbf{y}_{t-1}^* \end{pmatrix} + \begin{pmatrix} \mathbf{k}_t \\ \mathbf{k}_t^* \end{pmatrix}, \quad t = 1, K, T \quad (5)$$

The irregular component is assumed to be stationary and consists of a white noise disturbance term of the form:

$$\mathbf{e}_t \sim NID(0, \mathbf{s}_e^2)$$

The structural model, when written in state space form can be used to make predictions of future observations. A likelihood function for the observations is obtained from the state space form via prediction error decomposition. Using the Kalman filter (see Harvey

¹ First-order techniques are based on the first derivative of the mean square error function whereas second-order techniques use information about the second derivative as well. The latter will result in speeding up the learning process of the multi-layer perceptron.

(1991)) the parameters σ_η^2 , σ_ζ^2 and σ_ε^2 can be estimated by exact maximum likelihood in the time domain.

1.3 Stochastic Volatility Model

For most financial time series, groups of highly volatile observations are interspersed with tranquil periods. Modelling volatility becomes important if markets are efficient since the observations p_t of a particular financial time series should just follow a martingale (Fama (1965, 1970)). However, it is then possible to find non-linear models for the change in variance which might provide some predictive ability.

Changes in the variances v_t can be set up as a Gaussian white noise process \mathbf{e}_t which is multiplied by the standard deviation \mathbf{s}_t given by

$$v_t = \mathbf{s}_t \mathbf{e}_t, \quad \mathbf{e}_t \sim IID(0,1) \quad (6)$$

The stochastic volatility model assumes that the variance \mathbf{s}_t is an unobservable process and the volatility at time t given all the information up to time $t-1$ is random. Let the log volatility h_t denote a normally distributed unobservable random variable

$$h_t \sim NID(0, \mathbf{s}_h^2)$$

then:

$$\mathbf{s}_t^2 = \exp(h_t) \quad (7)$$

Thus, \mathbf{s}_t^2 can be generated by a linear stochastic process such as a first-order autoregression AR(1). The stationary AR(1) SV model is given by:

$$\begin{aligned} v_t &= \mathbf{e}_t \exp\left(\frac{h_t}{2}\right), \quad \text{with} \\ h_{t+1} &= \mathbf{g} + \mathbf{f}h_t + \mathbf{h}_t \end{aligned} \quad (8)$$

where $0 \leq \mathbf{f} \leq 1$ and

$$\mathbf{h}_t \sim (0, \mathbf{s}_h^2)$$

The non-linear equation of the stochastic volatility model can be modified to fit in a linear state space model. Assuming normality for \mathbf{e}_t it can be shown that $\log \mathbf{e}_t^2$ has a mean of -1.2704 and a variance of $\mathbf{p}^2/2$ (Abramovitz *et al.* (1972)).

Let:

$$\mathbf{e}_t^* = \log(\mathbf{e}_t^2) + 1.27 \quad (9)$$

then, by squaring the observations and taking logs, the stochastic volatility model equation is given in state space form as:

$$\begin{aligned}\log(v_t^2) &= -1.27 + h_t + \mathbf{e}_t^* \\ h_{t+1} &= \mathbf{g} + \mathbf{f}h_t + \mathbf{h}_t\end{aligned}\quad (10)$$

Assuming that $\log v_t^2$ is normally distributed, a quasi-likelihood estimation can be carried out by applying the Kalman filter to the state space form of the stochastic volatility model. An in-depth discussion of the Kalman filter and its application to stochastic volatility models can be found in Harvey (1991) and Harvey *et al.* (1994).

2. Forecasting The USD/DEM Exchange Rate

In this section we compare the performance of a multi-layer perceptron and a structural model in predicting the one step ahead level of the USD/DEM exchange rate. The data used consists of a three week period (1-19 March 1993) of FXFX USD/DEM indicative bid quotes provided by Olsen & Associates in tick-by-tick form. Weekends were excluded due to the negligible number of observations and price movements. The training sets for the structural model and the multi-layer perceptron were extracted from the first 10 working days. The last 5 working days were set aside as a test set to see how well both models generalise on unseen data. The data provided consisted of bid and ask quotes. Bid prices rather than mid-prices were used as extra noise is introduced into the data by considering mid-prices. This is because only the bid price is an actual quote whereas the ask price is artificially extrapolated.

A pre-processing step transformed the tick-by-tick data into regularly spaced observations. This was achieved by averaging the bid prices over periods of 5 minutes. Let n_t be the number of bid prices b in the t^{th} 5 minute interval then the t^{th} bid price average is given by:

$$p_t = \frac{\sum_n b}{n_t} \quad (11)$$

A graph of the equally spaced average bid prices p_t for the 10-day training period is given in Figure 3. The training and test sets contain 2864 observations and 1429 observations, respectively. For the purpose of training the multi-layer perceptron the training set was further subdivided into training and validation sets, the latter consisting of every third observation.

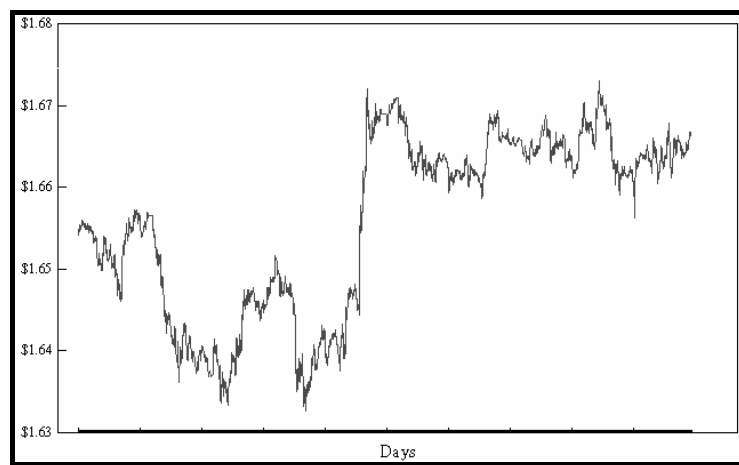


Figure 3: 10 days of the 5-minute intervals of the averaged bid prices

The multi-layer perceptron was used to predict the one-step ahead level of the USD/DEM exchange rate in the following manner. At time t first differences of the previous 12 observations were encoded as an input vector to the network. The inputs were normalised to have zero mean and unit variance² across the training set. The architecture contained a single output neuron. The output response was interpreted as the predicted *change* in the level of the exchange rate one-step ahead. The architecture therefore contained 12 input neurons, 1 output neuron and an undetermined number of hidden nodes.

To find the optimal number of hidden nodes, networks were trained varying the number of hidden neurons from 2 to 32. The architecture producing the lowest validation error was chosen. Architectures containing 2 hidden layers were also tested but their performance was found to be inferior (Toulson (1995)).

The structural model used contained a stochastic trend, a stochastic level and 2 cycles. The cycles had periods of 12 and 288 denoting 1 hour and 1 day, respectively. On the training period the structural model performed quite well (Table 1). This result seems to indicate some predictability contrary to the belief that the price should follow a martingale.

<i>Model</i>	<i>Validation error</i>	<i>Training error</i>	<i>Minimum iteration</i>	<i>% of successes</i>	<i>MAE</i>
Training set					
MLP 12-5-1	1.167	0.998	43	56.06	0.398
MLP 12-15-1	1.145	0.994	42	56.74	0.394
MLP 12-20-1	1.151	1.003	48	52.76	0.341
SM	n/a	n/a	50	58.35	0.365
Test set					
MLP 12-15-1	1.145	0.994	42	52.59	0.309
SM	n/a	n/a	50	51.93	0.399

Table1: Table comparing the different MLPs and the SM on the training and test sets

All of the multi-layer perceptrons seemed to train to roughly similar error levels, the best performance being obtained with 15 hidden nodes. However, the multi-layer perceptron was seen to be much better at predicting large changes than very small changes. This is illustrated in the graph shown in Figure 4. The graph shows the accuracy of the multi-layer perceptron in predicting the direction of increasingly large actual price movements. It can be seen that the multi-layer perceptron has an accuracy of about 65% in predicting the direction of price movements of greater than 8 pips. This evidence argues in favour of thresholding the MLP's response.

² The inputs were standardised by subtracting the mean and then dividing by the standard deviation.

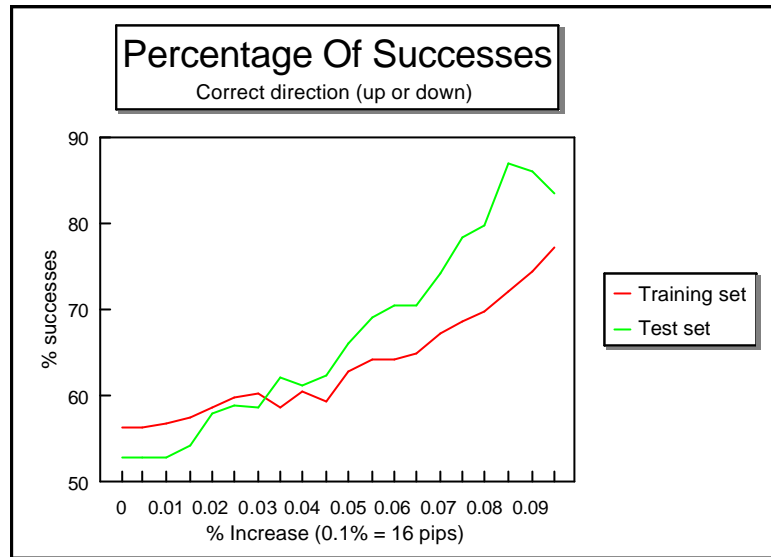


Figure 4: Success rate of correctly predicting the price movement versus the predicted absolute change.

3. Forecasting of the Volatility of the Bid Prices

Let D_t denote the first difference of the log of the t^{th} average bid price p_t . Then the volatility can be expressed as

$$\Delta_t = \log p_t - \log p_{t-1}, \quad t = 1, K, T. \quad (12)$$

Volatility was measured using the absolute value $|D_t|$. Again, 1429 observations (5 days of 5 minute intervals) were held back in the test set and were used for post-sample predictive testing.

A number of multi-layer perceptrons were compared with the stochastic volatility model. As before, all multi-layer perceptrons used 12 lagged observations to model one time step (5 minutes) ahead. The results suggest that both the stochastic volatility model and the multi-layer perceptron seem to capture the underlying pattern in changing volatility. Table 2 shows the results for modelling $|D_t|$ for the training and test periods. The results for the test set show that the mean average error (MAE) is lower for the multi-layer perceptron whereas the success rate of forecasting the increase or decrease in volatility is slightly higher for the stochastic volatility model.

<i>Model</i>	<i>% of successes</i>	<i>MAE</i>	<i>Training error</i>	<i>Validation error</i>
Training set				
MLP 12-10-1	73.18	16.609	0.021	0.013
MLP 12-15-1	73.23	16.581	0.020	0.011
MLP 12-20-1	72.88	16.735	0.021	0.013
SV	78.45	16.265	n/a	n/a
Test set				
MLP 12-15-1	72.97	13.478	0.020	0.011
SV	73.35	14.730	n/a	n/a

Combined MLP 13-15-1	75.94	13.474	0.011	0.008
----------------------	-------	--------	-------	-------

Table 2: Table comparing the MLP, the SV and the MLP with SV input for $|D_t|$ on the training and test sets

Based on these results it was proposed to combine the two models in order to try to reduce the overall errors. This was achieved by adding an extra input to the multi-layer perceptron containing the prediction of the stochastic volatility model for the next time-step ahead. It can be seen that the training and validation errors as well as the mean average error are lower for this augmented multi-layer perceptron (Combined MLP 13-15-1). Figure 6 shows the performance of the augmented multi-layer perceptron modelling $|D_t|$ for the test set. The improved performance of the augmented multi-layer perceptron seems to be due to additional information provided by the stochastic volatility model's forecast.

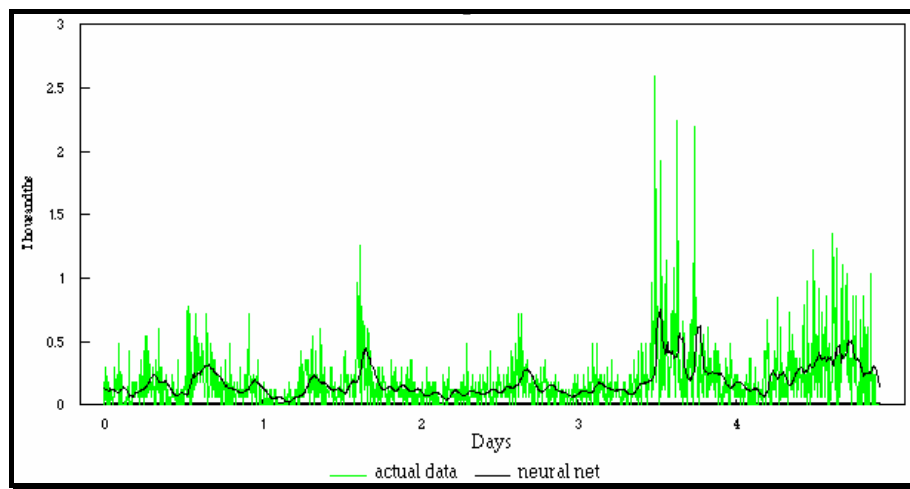


Figure 6: Graph of $|D_t|$ and predictions of the augmented MLP averaged over 12 lags for the test period.

4. Summary

In this paper we evaluated the structural model and the multi-layer perceptron in order to forecast the bid prices of the USD/DEM indicative quotes 5-minutes ahead and evidence presented herein tends to indicate that the multi-layer perceptron's performance is superior to that of the structural model. This may be attributed largely to the fact that the multi-layer perceptron is better able to predict large price movements. The results for both models seem to suggest that the price movement does not follow a random walk. So far predictions of only one time-step ahead have been considered. Work is currently being carried out to investigate predictive ability of both models to make longer term predictions.

Furthermore, the evidence presented in this paper lends to suggest that both the stochastic volatility model and the multi-layer perceptron give similar performances when trying to predict changes in volatility. Nevertheless, a combination of the multi-layer perceptron and the stochastic volatility model achieved superior results than either model individually. For this combined multi-layer perceptron the volatility forecast of the stochastic volatility model was used as an additional input to the network.

References

- Abramovitz M. and N.C. Stegun (1972)**, *“Handbook of Mathematical Functions”*, Dover Publications, New York.
- Fama E.F. (1965)**, *“The Behaviour of Stock Market Prices”*, Journal of Business, 38, 34-105.
- Fama E.F. (1970)**, *“Foundations of Finance”*, Basil Blackwell, Oxford.
- Fahlman S.E. (1988)**, *“Faster Learning Variations On Back-Propagation: An Empirical Study”*, Proceedings of the 1988 Connectionist Models Summer School, Morgan Kaufman, 38-51.
- Harvey A.C. (1991)**, *“Forecasting, Structural Time Series Models and the Kalman Filter”*, Cambridge University Press.
- Harvey A.C. (1993)**, *“Time Series Models”*, Second Edition, Harvester Wheatsheaf.
- Harvey A.C., E. Ruiz and N. Shephard (1994)**, *“Multivariate Stochastic Variance Models”*, Review of Economic Studies, 61, 247-264.
- McClelland J.L. and D.E. Rummelhart (1986)**, *“Parallel Distributed Processing”*, MIT Press.
- Refenes A.N., M. Azema-Barac and S.A. Karoussos (1993)**, *“Currency Exchange Rate Prediction And Neural Network Design Strategies”*, Neural Computing And Applications, 1 1 46-58.
- Rummelhart D.E., G.E. Hinton and R.J. Williams (1986)**, *“Learning Internal Representations by Error Propagation”*, in Parallel Distributed Processing, Vol 1, Ch 8. MIT Press.
- Taylor S. (1986)**, *“Modelling Financial Time Series”*, John Wiley & Sons.
- Toulson S.P. (1995)**, *“Comparison of Neural Network and Statistical Techniques Using High Frequency Exchange Rate Data”*, Proceedings First International Conference On High Frequency Data In Finance (HFDF-1), Zurich.
- White H. (1988)**, *“Economic Prediction Using Neural Networks: The Case OF IBM Daily Stock Returns”*, Proc. IEEE International Conference on Neural Networks, San Diego, II-451-459.