Pergamon

Omega, Int. J. Mgmt Sci. Vol. 26, No. 4, pp. 495–506, 1998

Printed in Great Britain
PII: S0305-0483(98)00003-6
0305-0483/98 $19.00 + 0.00

Neural Network Forecasting of the British Pound/US Dollar Exchange Rate

GIOQINANG ZHANG

Georgia State University, Atlanta, GA 30303, USA

MICHAEL Y. HU

Kent State University, Kent, OH 44242-0001, USA

(Received 4 April 1997; accepted after revision 8 December 1997)

Neural networks have successfully been used for exchange rate forecasting. However, due to a large number of parameters to be estimated empirically, it is not a simple task to select the appropriate neural network architecture for an exchange rate forecasting problem. Researchers often overlook the effect of neural network parameters on the performance of neural network forecasting. This paper examines the effects of the number of input and hidden nodes as well as the size of the training sample on the in-sample and out-of-sample performance. The British pound/US dollar is used for detailed examinations. It is found that neural networks outperform linear models, particularly when the forecast horizon is short. In addition, the number of input nodes has a greater impact on performance than the number of hidden nodes, while a larger number of observations do reduce forecast errors. © 1998 Elsevier Science Ltd. All rights reserved

Key words—foreign exchange rate, time series forecasting, neural networks

1. INTRODUCTION

EXCHANGE RATE FORECASTING is an important and challenging task for both academic researchers and business practitioners. Various theoretical models including both econometric and time series approaches have been suggested to model and forecast exchange rates. Unfortunately, empirical results often fail to meet theoretical expectations. Meese and Rogoff [1, 2] show that the out-of-sample performance of many structural and time series models is no better than that of a simple random walk model. Their findings discourage many researchers in the area since the superiority of the random walk means the unpredictability of the foreign exchange rate. However, as all models investigated in [1, 2] are linear, it is natural to conjecture that exchange rate data contain nonlinearities which may not be accounted for or approximated well by linear models. In fact, the linear unpredictability of exchange rates has been confirmed by many studies [3–11]. Much research efforts have been devoted to explore the nonlinearity of the exchange rate data and to develop specific nonlinear models to improve exchange rate forecasting.

Parametric nonlinear models such as autoregressive conditional heteroskedasticity (ARCH) [12, 13], general autoregressive conditional heteroskedasticity (GARCH) [14–16], chaotic dynamics [8, 17], and self-exciting threshold autoregressive [18] models have been proposed and applied to financial forecasting. While these models may be good for a particular situation, they perform poorly for other applications. The pre-specification of the model form restricts the usefulness of these parametric nonlinear models since there can be many other possible nonlinear patterns to be considered. One particular nonlinear specification will not be general enough to capture all the nonlinearities in the data. Diebold and Nason [19, p. 318] point out that "the overwhelming variety of plausible candidate nonlinear models makes the determination of a good approximation to the true data-generating process (DGP) a difficult task and the seemingly large variety of parametric nonlinear models is in fact a very small subset of the class of plausible nonlinear DGPs". A number of nonparametric methods have also been proposed to forecast exchange rates. Diebold and Nason [19] use a nearest neighbor technique called locally-weighted regression (LWR) to analyze both in-sample and out-of-sample behaviors of 10 exchange rate time series. Meese and Rose [10] also apply this nonparametric technique in five structural models to forecast four exchange rates. LWR is a general approach in estimating regression surfaces in a moving-average mode and is a powerful technique for detecting a very wide range of functions. Mizrach [20] explores the potential of multivariate nearest-neighbor methods in EMS exchange rate forecasting. However, nonparametric methods investigated in these studies are still unable to improve upon a simple random walk model in out-of-sample predictions of exchange rates.

Recently, artificial neural networks have received increasing attention as decision-making tools. Numerous studies have shown that neural networks can be one of the very useful tools in time series forecasting [21]. Neural networks have general nonlinear function mapping capability which can approximate any continuous function with arbitrarily desired accuracy [22, 23]. Neural networks are nonparametric data-driven models which impose little prior assumptions on the underlying processes from which data are generated. As such, they are less susceptible to the model misspecification problem than most parametric nonlinear methods. This is an important feature of neural networks since there are few exchange rate theories currently available that will prescribe the specific nonlinearities in exchange rate data.

Neural networks as models for forecasting exchange rates have been investigated in a number of studies. Weigend *et al.* [24, 25] have found that neural networks are better than random walk models in predicting the Deutsche mark/US dollar (DEM/USD) exchange rate. Refenes [26] develops a constructive learning algorithm to find the best neural network configuration in forecasting DEM/USD. Kuan and Liu [27] use both feedforward and recurrent neural networks to forecast five foreign exchange rates of British pound, Canadian dollar, Deutsche mark, Japanese yen, and Swiss franc against the US dollar. Wu [28] compares neural networks with ARIMA models in forecasting Taiwan/US dollar exchange rate and shows significantly better results with the neural network model. Hann and Steurer [29] make comparisons between the neural network and the linear model in USD/DEM forecasting. They report that if monthly data are used, neural networks do not show much improvement over linear models but for weekly data, neural networks are much better than both the monetary and random walk models. In all these neural network models, the researchers focus on the out-of-sample results and have not explicitly considered the effect of various factors on neural network performance.

Despite the many successful applications of neural networks in forecasting, building a neural network forecaster for a particular time series forecasting problem is a nontrivial task. In comparison with the traditional forecasting methods such as Box–Jenkins ARIMA models or regression models, there are many more modeling factors to be considered in neural networks. Zhang *et al.* [21] summarize the neural network modeling issues for forecasting. Kaastra and Boyd [30] propose an eight-step method in designing a neural network model for forecasting financial time series.

The purpose of this research is to provide an in-depth study of the effects of several important factors on the performance of neural networks in exchange rate forecasting. Specifically we will examine two neural network factors—the number of input nodes and the number of hidden nodes—on the forecasting performance of exchange rate between British pound and US dollar. Knowledge of the effects of these factors is helpful in the selection of the optimal neural network structure as well as in understanding the autocorrelation structure of exchange rate time series. In addition, the effect of training sample size is also investigated. This study employs the popular multi-layer feedforward neural networks. Although there exist many other types of neural networks, multi-layer feedforward networks are still the most commonly used network structures and are known to produce superior performance for many forecasting problems. We notice that most previous studies treat neural networks as a black box and focus only on the results. Few researchers discuss in detail how neural networks are constructed to perform a forecasting task for a time series.

2. NEURAL NETWORKS FOR TIME SERIES FORECASTING

Neural networks are computing models often used for pattern classification and pattern recognition problems. They can learn from examples or experiences and are particularly noted for their flexible function mapping ability. Neural networks are data-driven self-adaptive methods in that there are few *a priori* assumptions about the model form for a problem under study. These unique features make them valuable for solving many practical forecasting problems. Any time series forecasting model assumes that there is an underlying process from which data are generated and the future value of a time series is solely determined by the past and current observations. Neural networks are able to capture the underlying pattern or autocorrelation structure within a time series even when the underlying law governing the system is unknown or too complex to describe.

Numerous neural network models have been proposed and used for the forecasting purpose. The most popular and successful one is the feedforward multilayer network or the multi-layer perceptron (MLP). An MLP is typically composed of several layers of nodes. The first or the lowest layer is an input layer where external information is received. The last or the highest layer is an output layer where the problem solution is obtained. The input layer and output layer are separated by one or more intermediate layers called the hidden layers. The nodes in adjacent layers are usually fully connected by acyclic arcs from a lower layer to a higher layer. The knowledge learned by a network is stored in the arcs and the nodes in the form of arc weights and node biases which will be estimated in the neural network training process. Figure 1 is an example of a fully connected MLP with one hidden layer.

For a univariate time series forecasting problem, the inputs of the network are the past, lagged observations of the data series and the outputs are the future values. Each input pattern is composed of a moving window of fixed length along the series. The network represented by Fig. 1 is a mapping function of the form:

$$y_{t+1} = f(y_t, y_{t-1}, \ldots, y_{t-p}) \tag{1}$$

where y_t is the observation at time t and p is the dimension of the input vector or the number of past observations related to the future value. In this sense, the feedforward network used for time series forecasting is a general autoregressive model.

Suppose we have N time-lagged observations $y_1, y_2, \ldots, y_N$ in the training set and we need the one-step-ahead forecasts, then using a network with p input nodes and one output node, we have $N-p$ training patterns. The first training pattern is composed of y_1, $y_2, \ldots, y_p$ as the inputs and y_{p+1} as the target output. The second training pattern contains $y_2, y_3, \ldots, y_{p+1}$ for the inputs and y_{p+2} for the desired output. Finally, the last training pattern is $y_{N-p}, y_{N-p+1}, \ldots, y_{N-1}$ for the inputs and y_N for the target. See Fig. 2 for an illustration. The neural network training objective is to find the weights in order that some overall predictive error measure such as the sum of the squared errors (SSE) is minimized. For this network structure, SSE can be written as

$$SSE = \sum_{i=p+1}^{N} (y_i - a_i)^2 \tag{2}$$

where a_i is the output from the network.

3. RESEARCH DESIGN

3.1. Data

Bilateral exchange rates between British pound (GBP) and US dollar (USD) are

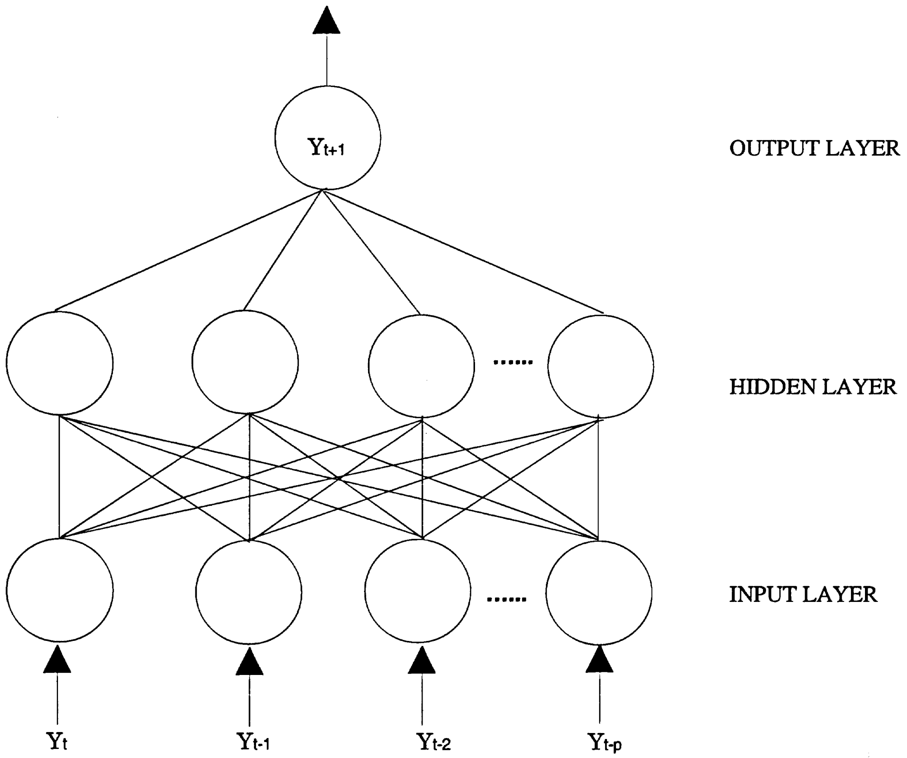

Fig. 1. A typical fully connected feedforward neural network for time series forecasting

obtained from Datastream International. Data is composed of daily rates from the beginning of 1976 through the end of 1993. The rates are quotations at 3:00 pm Eastern time from Banker Trust, USA. Weekly observations are compiled by taking the closing rates on Wednesday as the representative rates of that week to avoid potential biases caused by the weekend effect documented for many asset pricing studies. If a particular Wednesday happens to be a nontrading day, then either Tuesday or Thursday closing rates will be retained. Based on [19], there is little variation in results from one exchange rate to another when nonparametric methods are used, we use the GBP/USD exchange rate as an illustration of our analysis. Following Meese and Rogoff [1], we take the natural logarithmic transformation to stabilize the time series.

3.2. Experimental design

The major purpose of this study is to evaluate the effects of several factors on the in-sample fit and out-of-sample forecasting capabilities of neural networks. The neural network factors investigated are the number of input and hidden nodes which are two critical parameters in the design of a neural network. The number of input nodes is perhaps the most important factor in neural network analysis of a time series since it corresponds to the number of past lagged observations related to future values. It also plays a role in determining the autocorrelation structure of a time series. The number of hidden nodes allows neural networks to capture nonlinear patterns and detect complex relationships in the data. Without hidden nodes, simple perceptrons are equivalent to linear statistical models. In general, networks with too few hidden nodes may not have enough power to model and learn the data. But networks with too many hidden nodes may cause overfitting problems, leading to poor forecasting ability.

Ten levels of the number of input nodes ranging from 1 to 10 will be used in this study. For a linear time series problem, previous research shows that the most common order of autoregressive terms is 1 or 2 and very few problems have order of 3 or more [31]. For nonlinear time series, there are no reports on the number of autoregressive terms typically used for real life applications. We experiment with a relatively large number of input nodes. There is no upper limit on the possible number of hidden nodes in theory. However, it is rarely seen in the literature that the number of hidden nodes is more than double the number of input nodes [21]. In addition, previous research [32] indicates that the forecasting performance of neural networks is not as sensitive to the number of hidden nodes as it is to the number of input nodes. Thus, five levels of hidden nodes, 4, 8, 12, 16 and 20 will be experimented. The combination of ten input nodes and five hidden nodes yields a total of 50 neural network architectures being considered for each in-sample training data set.

Sample size is another factor which can affect the forecasting ability of neural networks. In the neural network literature, large samples are often claimed to be beneficial in training neural networks due to the large set of parameters involved in a neural network. However, Kang [33], in a comprehensive study of neural network time series forecasting, finds that neural network models do not necessarily require large data set to perform well. To test if there is a significant difference between large and small training samples in modeling and forecasting exchange rates, we use two training sample sizes in our study. The large sample is consisted of 887 observations from 1976 to 1992 and the small one includes 261 data points from 1988 to 1992. The test sample for both cases is the 1993 data which has 52 observations. For this out-of-sample, we adopt three time horizons of one month, six months, and twelve months to examine the forecast horizon effect. Forecasts are generated for the first 4 weeks, 26 weeks and 52 weeks of the test sample corresponding to the three forecast horizons of one month, six months and twelve months, respectively.

We will focus on one-step-ahead forecasts as in [19]. One-step-ahead prediction is useful in evaluating the adaptability and robustness of a forecasting method. In one-step-ahead forecasting, the exchange rate values are forecasted one step at a time and the actual rather than the forecasted values are then used for the next prediction in a forecasting horizon. One poor prediction will not have serious consequences in a one-step-ahead forecasting system since the actual value will be used to correct itself for the future forecasts.

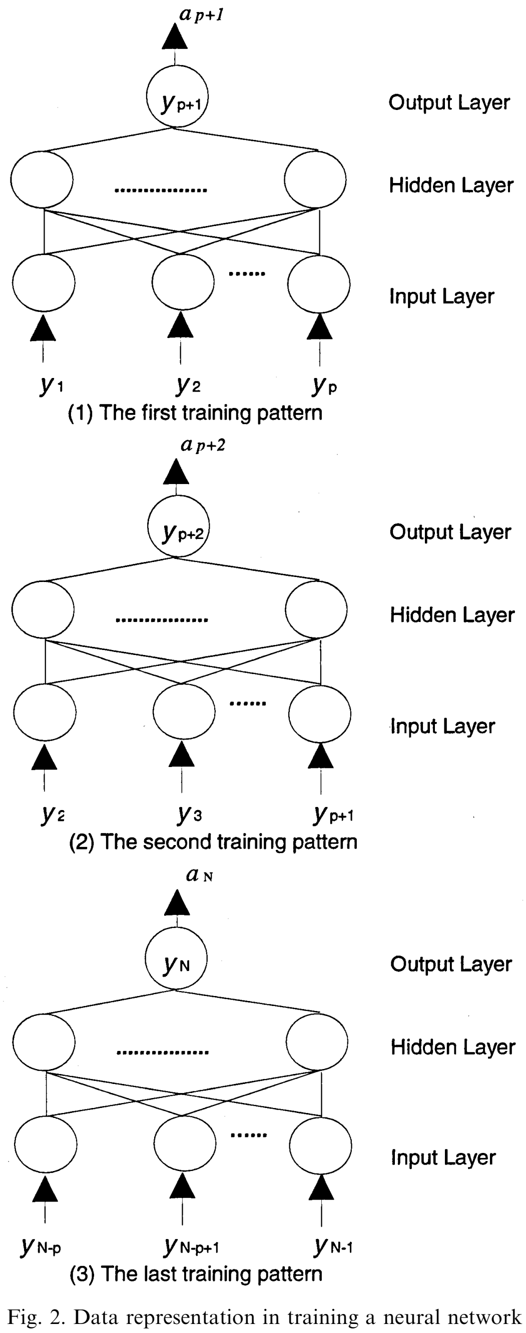

Fig. 2. Data representation in training a neural network

Furthermore, the random walk model will be used as a benchmark for comparison. The random walk is a one-step-ahead forecasting model since it uses the current observation to predict the next one. It is easy to see that random walk models will produce more accurate forecasts in one-step-ahead than in multi-step-ahead forecasts which rely on one single observation as predictions for all future values in a time horizon.

Three-layer feedforward neural networks are used to forecast the GBP/USD exchange rate. Logistic activation functions are employed in the hidden layer and the linear activation function is utilized in the output layer. Given that we are interested in one-step-ahead forecasts, one output node is deployed in the output layer.

We use node biases only in the output nodes. Based on the proof in [22, 23] that for neural networks to be universal function approximators, it is not necessary to have hidden node biases. Moreover, without hidden node biases, a more parsimonious model is achieved since fewer model parameters (weights) have to be estimated. Parsimony is a critical model characteristic for a forecasting situation where generalization of past patterns into the future is of utmost importance. We have found [34] that neural networks without hidden node biases often predict more accurately than those with hidden node biases.

There is no consensus on whether data normalization should be used [21]. For example, it is still unclear that whether there is a need to normalize the inputs because the arc weights can undo the scaling. Shanker *et al.* [35] investigate the effectiveness of linear and statistical normalization methods for classification problems. They find that data normalization does not necessarily lead to better performance particularly when the network and sample size are large. Based on our experience with the exchange rate data, we find no significant difference between using normalized and original data. Hence, raw data are used in this study.

Neural network training is a nonlinear nonconvex minimization problem because the estimation of arc weights entails nonlinear optimization. Therefore, global optimal solution cannot be guaranteed. In this study, we use a GRG2 based system recently proposed by Hung and Denton [36] and Subramanian and Hung [37] to train neural networks. GRG2 [38] is a general-purpose nonlinear optimizer which is widely available even in popular spreadsheet programs. The GRG2 based system has been shown to be particularly effective for highly nonlinear problems like those in neural network training. Using GRG2 eliminates the need to specify learning rate and the momentum term which are usually selected through lengthy experimentation in training neural networks with the popular backpropagation method. The benefit of using direct optimization procedure in neural network training is also illustrated in [39]. To be more certain of getting the true global optima, a common practice is to solve the problem using a number of randomly generated initial solutions. We train each network 50 times by using 50 sets of different initial arc weights (We have tried using 100 runs in a preliminary study and found practically no difference in solution from using 50 runs or 100 runs). The best solution among 50 runs is used as the optimal neural network training solution.

3.3. Performance measures and comparison

There is no consensus on the most appropriate measure to assess the performance of a forecasting technique. Here, we use three popular criteria: RMSE, MAE, and MAPE, to evaluate the predictive performance of neural networks. These forecasting accuracy measures are listed as follows:

$$RMSE = \sqrt{\frac{\sum (y_t - \hat{y}_t)^2}{T}} \tag{3}$$

$$MAE = \frac{\sum |y_t - \hat{y}_t|}{T} \tag{4}$$

$$MAPE = \frac{1}{T}\sum \left| \frac{y_t - \hat{y}_t}{y_t} \right| \times 100 \tag{5}$$

where y_t is the actual observation, $\hat{y}_t$ is the predicted value, and T is the number of predictions. These criteria are mean based and are frequently used performance measures in the literature.

We also apply the traditional Box–Jenkins ARIMA models to the exchange rate data. The Box–Jenkins models will be implemented with the SCA Statistical System [40]. SCA is a general statistical package which has extensive capabilities for forecasting and time series analysis [41]. The SCA system is widely available on mainframes, workstations, and PCs. In this paper, we use the automatic modeling capabilities of SCA by implementing the SCA–EXPERT function [42]. The SCA–EXPERT function employs an expert system technology to facilitate automatic ARIMA modeling. Through the iterative process of model identification, parameter estimation and diagnostic checking, SCA–EXPERT produces the 'best' linear model fitted to the data at hand. Neural network forecasts will then be compared with the forecasts obtained from SCA–EXPERT.

4. RESULTS

Since the major purpose of this study is to investigate the effects of neural network factors on the modeling and forecasting performance of neural networks, both in-sample (training set) and out-of-sample (test set) results will be discussed in this section. The focus will be on the out-of-sample analysis because it is the forecasting capability that researchers and practitioners are most interested in. With most traditional forecasting models such as linear regression and Box–Jenkins models, in-sample fitting may not be an issue because of the parametric characteristic of the linear model and the existence of the optimal solution using least squares method. For neural network method, however, due to the potential problem of overfitting, we need to study the conditions under which overfitting may occur. An overfitted model gives good in-sample fit to the training data, yet poor predictive out-of-sample performance. Therefore, the examination of both in-sample and out-of-sample behaviors will provide us information on when and how overfitting occurs. In addition, the effects of neural network model parameters are evaluated along with two sample sizes. The purpose is to examine whether the pattern of the effects changes with the sample size.

Table 1 contains the in-sample results for the large training sample of 887 observations. It is quite evident that as the number of hidden nodes increases, RMSE decreases. This pattern is observed consistently in each level of the input node. This phenomenon is to be expected since the objective in training a neural network is to minimize the SSE or equivalently RMSE. As more hidden nodes are used, the neural network becomes more 'powerful' in modeling the data. Note that too powerful networks are undesirable because they tend to memorize the specific features of

Table 1. Effects of neural network factors on training performance (training period: 1976–92)

Input	Hidden	RMSE	MAE	MAPE	Input	Hidden	RMSE	MAE	MAPE
1	4	0.006931	0.005003	2.587459	6	4	0.006851	0.004992	2.579090
1	8	0.006931	0.005002	2.587186	6	8	0.006821	0.004993	2.576254
1	12	0.006931	0.005002	2.587236	6	12	0.006790	0.004923	2.547637
1	16	0.006930	0.005002	2.587389	6	16	0.006773	0.004975	2.581466
1	20	0.006930	0.005002	2.587291	6	20	0.006737	0.004952	2.565525
	Avg	0.006931	0.005002	2.587312		Avg	0.006795	0.004967	2.569994
2	4	0.006919	0.004991	2.586074	7	4	0.006860	0.004999	2.601984
2	8	0.006912	0.004995	2.595125	7	8	0.006819	0.005004	2.600759
2	12	0.006911	0.004994	2.591365	7	12	0.006812	0.004959	2.581528
2	16	0.006908	0.004997	2.592680	7	16	0.006770	0.004977	2.587919
2	20	0.006908	0.004995	2.591257	7	20	0.006757	0.004965	2.575223
	Avg	0.006912	0.004994	2.591300		Avg	0.006803	0.004981	2.589483
3	4	0.006893	0.004979	2.582800	8	4	0.006848	0.005002	2.577210
3	8	0.006891	0.004984	2.596662	8	8	0.006825	0.004987	2.595661
3	12	0.006882	0.004989	2.599580	8	12	0.006798	0.004991	2.587310
3	16	0.006881	0.004973	2.588106	8	16	0.006773	0.004960	2.567285
3	20	0.006880	0.004983	2.591278	8	20	0.006748	0.004965	2.553722
	Avg	0.006885	0.004982	2.591685		Avg	0.006799	0.004981	2.576238
4	4	0.006900	0.004968	2.593493	9	4	0.006843	0.004999	2.589117
4	8	0.006878	0.004955	2.575098	9	8	0.006835	0.004993	2.587539
4	12	0.006865	0.004965	2.593208	9	12	0.006754	0.004983	2.577673
4	16	0.006857	0.004955	2.575538	9	16	0.006752	0.004959	2.569639
4	20	0.006844	0.004971	2.593350	9	20	0.006675	0.004900	2.513428
	Avg	0.006869	0.004963	2.586137		Avg	0.006772	0.004967	2.567479
5	4	0.006872	0.004978	2.586478	10	4	0.006770	0.004923	2.537077
5	8	0.006850	0.004955	2.568201	10	8	0.006773	0.004940	2.557905
5	12	0.006836	0.004950	2.572008	10	12	0.006699	0.004886	2.524505
5	16	0.006803	0.004964	2.586817	10	16	0.006715	0.004948	2.563729
5	20	0.006778	0.004924	2.563512	10	20	0.006690	0.004945	2.553156
	Avg	0.006828	0.004954	2.575403		Avg	0.006729	0.004928	2.547274

individual observations rather than the general pattern in the data, causing poor generalizations. In addition, as the number of input nodes increases, RMSE also decreases as reflected by the average RMSEs across the five hidden node levels at each level of the input node.

We observe a different pattern for the effects of the input node and the hidden node with MAE and MAPE. MAE and MAPE do not decrease in general as the number of hidden nodes increases within each level of input node. However, as the number of input nodes increases from 1 to 5, the mean MAE steadily decreases from 0.005002 to 0.004954. When the number of input nodes is in the range of 6 to 10, overall MAE increases first and then decreases. MAPE does not show a clear input node effect. This is reasonable since our purpose of neural network training is to minimize the SSE or RMSE, and not MAE and MAPE. It is important to note that there is less variation among different hidden node levels within each input node level than among different input node levels, suggesting that the number of input nodes has greater impact on the model fitting process of neural networks.

Table 2 gives similar results for the small training set with 261 observations. From Table 1 and Table 2, we find that lower RMSEs and MAEs are always achieved with the large sample while lower MAPEs are associated with the small sample. The implication is that if a mean based absolute measure such as RMSE or MAE is more appropriate, then including more observations in the training set is beneficial. However, if a relative performance measure like MAPE is used to evaluate model fitting, then a small data set in the training phase is desirable.

Next, we turn to the one-step-ahead out-of-sample analysis to examine the predictive capabilities of neural networks as the neural network structure changes. Neural networks performance will also be compared with that of the forecasting models selected by SCA which happen to be the random walk model in our study.

Table 3 shows the out-of-sample prediction results from using the large training sample. We use index 1, 2 and 3 for each performance

Table 2. Effects of neural network factors on training performance (training period: 1988–92)

Input	Hidden	RMSE	MAE	MAPE	Input	Hidden	RMSE	MAE	MAPE
1	4	0.0076932	0.005664	2.387574	6	4	0.0074764	0.005606	2.372438
1	8	0.0076932	0.005664	2.387650	6	8	0.0073456	0.005485	2.322072
1	12	0.0076932	0.005664	2.387650	6	12	0.0071967	0.005410	2.282050
1	16	0.0076932	0.005664	2.387662	6	16	0.0071223	0.005258	2.218107
1	20	0.0076932	0.005664	2.387606	6	20	0.0070857	0.005306	2.240919
	Avg	0.007693	0.005664	2.387628		Avg	0.0072453	0.005413	2.2871172
2	4	0.0076074	0.00562	2.370522	7	4	0.007372	0.005638	2.373611
2	8	0.0075907	0.005605	2.364150	7	8	0.0073178	0.005540	2.336563
2	12	0.0075928	0.005610	2.366275	7	12	0.0072313	0.005535	2.337084
2	16	0.007590	0.005617	2.369127	7	16	0.0071128	0.005359	2.254627
2	20	0.007593	0.005604	2.363875	7	20	0.0070233	0.005292	2.223896
	Avg	0.007595	0.005611	2.366790		Avg	0.0072115	0.005473	2.3051562
3	4	0.0075279	0.005576	2.361224	8	4	0.0073025	0.005501	2.326813
3	8	0.007491	0.005515	2.331886	8	8	0.0072754	0.005513	2.332671
3	12	0.007480	0.005546	2.343020	8	12	0.007200	0.005497	2.319150
3	16	0.0074725	0.005497	2.323146	8	16	0.0070757	0.005312	2.232497
3	20	0.0074759	0.005533	2.338162	8	20	0.0070577	0.005321	2.248040
	Avg	0.007489	0.005533	2.339488		Avg	0.0071823	0.005429	2.2918342
4	4	0.0075844	0.005663	2.398957	9	4	0.0073694	0.005502	2.316137
4	8	0.007448	0.005514	2.325359	9	8	0.0072172	0.005486	2.309043
4	12	0.007417	0.005526	2.328228	9	12	0.0072084	0.005432	2.292847
4	16	0.0073954	0.005472	2.306762	9	16	0.0070444	0.005279	2.233593
4	20	0.0074131	0.005470	2.306587	9	20	0.0070594	0.005239	2.199270
	Avg	0.007452	0.005529	2.333179		Avg	0.007180	0.005388	2.270178
5	4	0.0075233	0.005592	2.368445	10	4	0.0073141	0.005582	2.362976
5	8	0.0074219	0.005437	2.294986	10	8	0.0071788	0.005525	2.324372
5	12	0.0073194	0.005462	2.302534	10	12	0.0071592	0.005452	2.300672
5	16	0.0072664	0.005361	2.265338	10	16	0.0070825	0.005433	2.292739
5	20	0.0072719	0.005373	2.265093	10	20	0.006967	0.005286	2.210740
	Avg	0.007361	0.005445	2.299279		Avg	0.007140	0.005456	2.298300

Table 3. Out-of-sample analysis: effects of input nodes (training period: 1976–92)

Input	RMSE1	MAE1	MAPE1	RMSE2	MAE2	MAPE2	RMSE3	MAE3	MAPE3
1	0.006050	0.004998	2.711644	0.007500	0.006018	3.472401	0.006712	0.005334	3.058212
2	0.006077	0.005052	2.740974	0.007525	0.006026	3.475833	0.006722	0.005339	3.060681
3	0.005822	0.004875	2.64527	0.007481	0.005983	3.453581	0.006701	0.005315	3.047933
4	0.005710	0.004778	2.593425	0.007508	0.005961	3.440397	0.006710	0.005301	3.039613
5	0.005545	0.004379	2.380488	0.007533	0.005941	3.435068	0.006717	0.005304	3.044636
6	0.004997	0.003838	2.088325	0 007553	0.005843	3.382352	0.006744	0.005285	3.036702
7	0.005212	0.004144	2.250345	0.007664	0.005978	3.455515	0.006798	0.005337	3.064864
8	0.005116	0.004068	2.210713	0.007724	0.006001	3.464358	0.006816	0.005306	3.045748
9	0.005435	0.004202	2.285952	0.007830	0.006047	3.493602	0.006889	0.005361	3.078797
10	0.005537	0.004292	2.334496	0.007827	0.006156	3.559376	0.006918	0.005428	3.118488
RW	0.006070	0.005016	2.720807	0.007522	0.006045	3.486385	0.006730	0.005360	3.072079

Note: neural network results are the averages across five levels of hidden nodes.

measure of the three time horizons. That is, RMSE1, MAE1 and MAPE1 are used for the one-month time horizon; RMSE2, MAE2 and MAPE2 are for the six-month time horizon; and RMSE3, MAE3, and MAPE3 are for the 12-month horizon. The effect of hidden nodes is first aggregated out by computing the averages for the various measures at each level of input nodes. For one-month horizon, all three measures of performance indicate 6 input nodes produce the best predictions. Average RMSE, MAE and MAPE take on values of 0.004997, 0.003838 and 2.088325 respectively. As the length of time horizon increases, the effects of input nodes on MAE and MAPE are quite consistent. The network with six input nodes is still the overall best architecture. The observed pattern in RMSE for 6- and 12-month horizons is not the same as in the case of one-month horizon. For 6- and 12-month time horizons, the lowest average RMSE is 0.007481 and 0.006701 which occur with three input nodes, indicating that the specification of the number of input nodes may be sensitive to the performance measure and the forecast horizon.

It is not surprising to find that the SCA–EXPERT picks the random walk model for the time series under study. This is consistent with what has been reported in the literature as no *linear* time series models are better than the random walk model. Results from the random walk model in three performance measures across the three time horizons are reported at the bottom of Table 3. It is clear that neural networks predict much better than the random walk model in terms of all three measures across the three time horizons not only for the 'best' neural network models but also for most other network architectures. For example, in one-month horizon, except for the two-input-node case, all other cases have smaller error measures than the random walk.

Out-of-sample results using the 1988–92 small training sample are given in Table 4. For one-month forecast horizon, the best average performance across the three different measures occurs at 8 input nodes with RMSE of 0.005918, MAE of 0.004535, and MAPE of 2.470946, which are all lower than the corresponding results from the random walk model listed at the bottom of Table 4. Note that one

Table 4. Out-of-sample analysis: effects of input nodes (training period: 1986–92)

Input	RMSE1	MAE1	MAPE1	RMSE2	MAE2	MAPE2	RMSE3	MAE3	MAPE3
1	0.005973	0.004562	2.493127	0.009644	0.007519	4.464940	0.008677	0.006962	4.080338
2	0.006582	0.005409	2.945701	0.012497	0.010096	6.020766	0.011500	0.009594	5.636496
3	0.007376	0.006319	3.432532	0.012697	0.010222	6.084185	0.011665	0.009814	5.750478
4	0.007303	0.005895	3.209151	0.013104	0.010567	6.283370	0.012042	0.010107	5.920922
5	0.007220	0.005537	3.020568	0.011394	0.009142	5.397156	0.010470	0.008634	5.038241
6	0.007029	0.005438	2.967680	0.012926	0.010118	5.990536	0.011282	0.009099	5.322131
7	0.006544	0.005100	2.780468	0.011167	0.008581	5.012582	0.009447	0.007261	4.218160
8	0.005918	0.004535	2.470946	0.013851	0.010636	6.272102	0.011341	0.008663	5.060446
9	0.008702	0.007465	4.034789	0.012657	0.009725	5.596404	0.010543	0.008045	4.613580
10	0.008502	0.007763	4.181009	0.012939	0.009968	5.685836	0.010843	0.008118	4.613877
RW	0.006070	0.005016	2.720807	0.007522	0.006045	3.486385	0.006730	0.005360	3.072079

Note: neural network results are the averages across 5 levels of hidden nodes.

input node gives almost the same results as the eight input node model. For longer forecast horizons, the neural network model with one input node provides the best overall prediction along all three performance criteria. However, at these longer forecast horizons, neural networks do not outperform the random walk model although the difference between the best neural network and the random walk is not significant. This observation suggests that there are structure changes in the exchange rate data, a specific neural network model for the past may not be appropriate for long term forecasting.

Table 5 shows the effects of hidden nodes on the out-of-sample performance for both the large and small samples. Because of the similarity in the effects with different levels of input nodes, only the results with six input nodes for the large sample and one input node for the small sample are reported. Results show no clear effects of hidden nodes across different time horizons along different performance measures. For example, with six input nodes for the large sample in one-month horizon, the network architecture with 16 hidden nodes yields the smallest RMSE, MAE and MAPE. For longer forecast horizons, 20 hidden nodes yield the smallest MAE and MAPE, and 16 hidden nodes produce the lowest RMSE. The differences in performance measures across the levels of hidden nodes are very small, indicating the number of hidden nodes does not occupy an important role in the out-of-sample performance of neural networks.

Overall, we see a different pattern in the out-of-sample predictive performance as in the in-sample characterization with respect to the effects of both input and hidden nodes. In the out-of-sample, there is clear indication of overfitting effect in neural networks. From Table 3, we observe that as the number of input nodes increases, all three measures across three forecast horizons exhibit decreasing first and then increasing trends. This implies that including more input nodes than necessary will adversely affect the predictive ability of neural network. Another observation is that within each input node level, the results are not very sensitive to the change in the number of hidden nodes. Hence, correctly identifying the number of input nodes is more important than identifying the number of hidden nodes. Our results are in line with the findings reported by Lachtermacher and Fuller [32] who apply neural networks to predict one-step-ahead river flow.

From Table 3 and Table 4, it is interesting to note that in general, the performance measures for the neural network model in the 1988–92 series (the small training sample) take on larger values than the corresponding values from the 1976–92 series (the large training sample). For example, RMSE1, MAE1, and MAPE1 in the small sample with six input nodes have values of 0.007029, 0.005438, and 2.96768 respectively, as compared to 0.004997, 0.003838, and 2.088325 for these measures at one-month horizon in the long series. This observation implies that forecasts based on long series will lead to a lesser amount of forecast error.

5. SUMMARY AND CONCLUSION

Exchange rates forecasting using neural networks has recently received much attention. In

Table 5. Effect of hidden nodes

Sample size	Input	Hidden	RMSE1	MAE1	MAPE1	RMSE2	MAE2	MAPE2	RMSE3	MAE3	MAPE3
Large	6	4	0.005349	0.004106	2.234398	0.007523	0.005954	3.44287	0.006758	0.005356	3.076322
	6	8	0.00482	0.003637	1.977423	0.007568	0.005847	3.383286	0.006754	0.005288	3.03719
	6	12	0.005295	0.004215	2.291136	0.007503	0.005878	3.39638	0.006726	0.005302	3.042675
	6	16	0.004539	0.003423	1.865175	0.007645	0.005771	3.346312	0.006800	0.005293	3.044094
	6	20	0.004982	0.003807	2.073491	0.007524	0.005765	3.342913	0.006684	0.005185	2.983227
Small	1	4	0.005973	0.004563	2.493249	0.009614	0.007493	4.448447	0.008652	0.006937	4.065202
	1	8	0.005973	0.004561	2.492704	0.009667	0.007540	4.477813	0.008696	0.006981	4.091854
	1	12	0.005974	0.004562	2.493145	0.009651	0.007526	4.468852	0.008683	0.006968	4.084014
	1	16	0.005974	0.004564	2.493997	0.009627	0.007505	4.455806	0.008664	0.006950	4.072758
	1	20	0.005973	0.004561	2.492539	0.009659	0.007533	4.473783	0.008689	0.006975	4.087861

this study, we investigate the effects of three important factors on neural network modeling and forecasting performance for weekly British pound/US dollar exchange rate. The number of input nodes and the number of hidden nodes are the experimental neural network factors. Both in-sample fitting ability and out-of-sample predictive performance with three forecast horizons are evaluated along three criteria, RMSE, MAE, and MAPE. In addition, the effect of two training sample sizes are examined with the identical forecast horizons.

Our study shows that the appropriate selection of neural network inputs and architecture structure is critical for the predictive accuracy of exchange rates. The purpose of this paper is not to provide a clear-cut guidance for selecting an optimal neural network structure for forecasting. Rather, our purpose is to examine the effects of several important neural network factors on model fitting and forecasting behaviors. Obviously, for the forecasting purpose, it is not appropriate to evaluate the neural network capability with the training sample alone. Currently there are no broadly accepted methods for construction of the best predictive model using strictly in-sample training data. The selection of the optimal network architecture should be based on test sample results. The optimal neural network model is the one that gives the best results in the test sample. If this strategy is used for model selection, then a true out-of-sample or validation sample should be employed for model validity purposes. We have also found that the number of input nodes plays an important role in neural network time series forecasting. This is consistent with the theoretical expectation that it is the number of input nodes that determines the (nonlinear) autocorrelation structure. Correct identification of the underlying autocorrelation structure is fundamental in time series forecasting. Neural network performance is further shown to be insensitive to the number of hidden nodes. In addition, neural networks fail to outperform the random walks for longer forecast horizons, suggesting possible structure changes in exchange rate data. Therefore as more observations are available, they should be used to revise the neural network model to better reflect the change of the underlying patterns.

The limitation of this study lies in two aspects. First, only the GBP/USD exchange rate is selected for detailed examination. Although Diebold and Nason [19] and many other studies in exchange rate forecasting show that there is little difference in in-sample fitting and out-of-sample predictive results from one exchange rate to another, cautions may still need to be exercised for generalization of the results to other exchange rates. Second, we investigate only one-step-ahead forecasting strategy. Results for multiple-step forecasting may be different from those obtained from this study. Future research should include the cross validation methodology that can be used to identify the best structure of the neural networks. Robustness of neural networks to the changing structures, or turning points typically associated with exchange rate series, can be investigated by using multiple training and test samples systematically chosen from the original series.

REFERENCES

1. Meese, R. A. and Rogoff, K., Empirical exchange rate models of the seventies: Do they fit out of sample? *Journal of International Economics*, 1983, **14,** 3–24.
2. Meese, R. A. and Rogoff, K. (1983) The out of sample failure of empirical exchange rate models: Sampling error or misspecification?. In *Exchange Rates and International Economics*, ed. J. Frankel. University of Chicago Press, Chicago.
3. Brooks, C., Testing for non-linearity in daily sterling exchange rates. *Applied Financial Economics*, 1996, **6,** 307–317.
4. Chiarella, C., Peat, M. and Stevenson, M., Detecting and modelling nonlinearity in flexible exchange rate time series. *Asia Pacific Journal of Management*, 1994, **11**(2), 159–186.
5. Chinn, M. D., Some linear and nonlinear thoughts on exchange rates. *Journal of International Money and Finance*, 1991, **10,** 214–230.
6. Hsieh, D. A., The statistical properties of daily foreign exchange rates: 1974–83. *Journal of International Economics*, 1988, **24,** 129–145.
7. Hsieh, D. A., Testing for nonlinear dependence in daily foreign exchange rates. *Journal of Business*, 1989, **62,** 329–368.
8. Hsieh, D. A., Chaos and nonlinear dynamics: application to financial markets. *Journal of Finance*, 1991, **46,** 1839–1877.
9. Meese, R. A. and Rogoff, K., Was it real? The exchange rate-interest differential relation over the modern floating-rate period *Journal of Finance*, 1988, **43,** 933–947.
10. Meese, R. A. and Rose, A., An empirical assessment of non-linearities in models of exchange rate determination. *Review of Economic Studies*, 1991, **58,** 601–619.
11. Satchell, S. and Timmermann, A., An assessment of the economic value of non-linear foreign exchange

rate forecasts. *Journal of Forecasting*, 1995, **14,** 477–497.

12. Engle, R. F., Autoregressive conditional heteroscedasticity with estimates of the variance of UK inflation. *Econometrica*, 1982, **50,** 987–1008.
13. Hsieh, D. A., Modeling heteroscedasticity in daily foreign-exchange rates. *Journal of Business and Economic Statistics*, 1989, **7,** 307–317.
14. Bollerslev, T., Generalized autoregressive conditional heteroscedasticity. *Journal of Econometrics*, 1986, **31,** 307–327.
15. Bollerslev, T., A conditionally heteroscedastic time series model for speculative prices and rates of return. *Review of Economics and Statistics*, 1987, **9,** 542–547.
16. Bollerslev, T., Modeling the coherence in short-run nominal exchange rates: a multivariate generalized ARCH model. *Review of Economics and Statistics*, 1990, **72,** 498–505.
17. Peel, D. A. and Yadav, P., The time series behaviour of spot exchange rates in the German hyper-inflation period: (Was the process chaotic?) *Empirical Economics*, 1995, **20**(3), 455–463.
18. Chappel, D., Padmore, J., Mistry, P. and Ellis, C., A threshold model for the French franc/Deutschmark exchange rate. *Journal of Forecasting*, 1996, **15**(3), 155–164.
19. Diebold, F. X. and Nason, J. A., Nonparametric exchange rate prediction? *Journal of International Economics*, 1990, **28,** 315–332.
20. Mizrach, B., Multivariate nearest-neighbour forecasts of EMS exchange rates. *Journal of Applied Econometrics*, 1992, **7,** S151–S164.
21. Zhang, G., Patuwo, B. E. and Hu, M. Y. (1997) Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, forthcoming.
22. Cybenko, G., Approximation by superpositions of a sigmoidal function. *Mathematical Control Signals Systems*, 1989, **2,** 303–314.
23. Hornik, K., Stinchcombe, M. and White, H., Multilayer feedforward networks are universal approximators. *Neural Networks*, 1989, **2,** 359–366.
24. Weigend, A. S., Huberman, B. A. and Rumelhart, D. E. (1992) Predicting sunspots and exchange rates with connectionist networks. In *Nonlinear modeling and forecasting*, eds. M. Casdagli and S. Eubank. Addison-Welsey, Redwood City, 395–432.
25. Weigend, A. S., Rumelhart, D. E. and Huberman, B. A., Generalization by weight-elimination with application to forecasting. *Advances in Neural Information Processing Systems*, 1991, **3,** 875–882.
26. Refenes, A. N. (1993) Constructive learning and its application to currency exchange rate forecasting. In *Neural networks in finance and investing: using artificial intelligence to improve real-world performance*, eds. R. R. Trippi and E. Turban. Chicago: Probus Publishing Company, 465–493.
27. Kuan, C. M. and Liu, T., Forecasting exchange rates using feedforward and recurrent neural networks. *Journal of Applied Econometrics*, 1995, **10,** 347–364.
28. Wu, B., Model-free forecasting for nonlinear time series (with application to exchange rates). *Computational Statistics and Data Analysis*, 1995, **19,** 433–459.
29. Hann, T. H. and Steurer, E., Much ado about nothing? Exchange rate forecasting: neural networks vs. linear models using monthly and weekly data *Neurocomputing*, 1996, **10,** 323–339.
30. Kaastra, I. and Boyd, M., Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, 1996, **10,** 215–236.
31. Pankratz, A. (1983) *Forecasting with Univariate Box-Jenkins Models: Concepts and Cases*, Wiley, New York.
32. Lachtermacher, G. and Fuller, J. D., Backpropagation in time-series forecasting. *Journal of Forecasting*, 1995, **14,** 381–393.
33. Kang, S. (1991) An investigation of the use of feedforward neural network for forecasting. Ph.D Thesis, Kent State University, Kent, OH 44240, USA.
34. Hu, M. Y., Hung, M. S., Shanker, M. S. and Chen, H., Using neural networks to predict performance of sino-foreign joint ventures. *International Journal of Computational Intelligence and Organization*, 1996, **1**(3), 134–143.
35. Shanker, M., Hu, M. Y. and Hung, M. S., Effect of data standardization on neural network training. *Omega*, 1996, **24**(4), 385–397.
36. Hung, M. S. and Denton, J. W., Training neural networks with the GRG2 nonlinear optimizer. *European Journal of Operational Research*, 1993, **69,** 83–91.
37. Subramianian, V. and Hung, M. S., A GRG2-based system for training neural networks: Design and computational experience. *ORSA Journal on Computing*, 1993, **5**(4), 386–394.
38. Lasdon, L. S. and Waren, A. D. (1986) *GRG2 User's Guide*; School of Business Administration: University of Texas at Austin, Austin, TX.
39. Curry, B. and Morgan, P., Neural networks: a need for caution. *Omega*, 1997, **25**(1), 123–133.
40. Liu, L. M., Hudak, G. B., Box, G. E. P., Muller, M. E. and Tiao, G. C. (1991) *The SCA Statistical System: Reference Manual for Fundamental Capabilities. Second edition*. Scientific Computing Associates, Chicago.
41. Liu, L. M., Hudak, G. B., Box, G. E. P., Muller, M. E. and Tiao, G. C. (1986) *The SCA Statistical System: Reference Manual for Forecasting and Time Series Analysis*. Scientific Computing Associates, Chicago.
42. Liu, L. M. (1996) *Modeling and Forecasting Time Series Using SCA-Expert Capabilities*. Scientific Computing Associates, Chicago.

ADDRESS FOR CORRESPONDENCE: *Michael Y. Hu, Department of Marketing, Kent State University, Kent, OH 44242, USA. Tel.: (330) 672-2170 ext. 326. Fax: (330) 672-2448. E-mail: mhu@kentvm.kent.edu.*