

# Applications of Advanced Regression Analysis for Trading and Investment\*

CHRISTIAN L. DUNIS AND MARK WILLIAMS

## ABSTRACT

This chapter examines and analyses the use of regression models in trading and investment with an application to foreign exchange (FX) forecasting and trading models. It is not intended as a general survey of all potential applications of regression methods to the field of quantitative trading and investment, as this would be well beyond the scope of a single chapter. For instance, time-varying parameter models are not covered here as they are the focus of another chapter in this book and Neural Network Regression (NNR) models are also covered in yet another chapter.

In this chapter, NNR models are benchmarked against some other traditional regression-based and alternative forecasting techniques to ascertain their potential added value as a forecasting and quantitative trading tool.

In addition to evaluating the various models using traditional forecasting accuracy measures, such as root-mean-squared errors, they are also assessed using financial criteria, such as risk-adjusted measures of return.

Having constructed a synthetic EUR/USD series for the period up to 4 January 1999, the models were developed using the same in-sample data, leaving the remainder for out-of-sample forecasting, October 1994 to May 2000, and May 2000 to July 2001, respectively. The out-of-sample period results were tested in terms of forecasting accuracy, and in terms of trading performance via a simulated trading strategy. Transaction costs are also taken into account.

It is concluded that regression models, and in particular NNR models do have the ability to forecast EUR/USD returns for the period investigated, and add value as a forecasting and quantitative trading tool.

## 1.1 INTRODUCTION

Since the breakdown of the Bretton Woods system of fixed exchange rates in 1971–1973 and the implementation of the floating exchange rate system, researchers have been motivated to explain the movements of exchange rates. The global FX market is massive with

---

\* The views expressed herein are those of the authors, and not necessarily those of Girobank.

an estimated current daily trading volume of USD 1.5 trillion, the largest part concerning spot deals, and is considered deep and very liquid. By currency pairs, the EUR/USD is the most actively traded.

The primary factors affecting exchange rates include economic indicators, such as growth, interest rates and inflation, and political factors. Psychological factors also play a part given the large amount of speculative dealing in the market. In addition, the movement of several large FX dealers in the same direction can move the market. The interaction of these factors is complex, making FX prediction generally difficult.

There is justifiable scepticism in the ability to make money by predicting price changes in any given market. This scepticism reflects the efficient market hypothesis according to which markets fully integrate all of the available information, and prices fully adjust immediately once new information becomes available. In essence, the markets are fully efficient, making prediction useless. However, in actual markets the reaction to new information is not necessarily so immediate. It is the existence of market inefficiencies that allows forecasting. However, the FX spot market is generally considered the most efficient, again making prediction difficult.

Forecasting exchange rates is vital for fund managers, borrowers, corporate treasurers, and specialised traders. However, the difficulties involved are demonstrated by the fact that only three out of every 10 spot foreign exchange dealers make a profit in any given year (Carney and Cunningham, 1996).

It is often difficult to identify a forecasting model because the underlying laws may not be clearly understood. In addition, FX time series may display signs of nonlinearity which traditional linear forecasting techniques are ill equipped to handle, often producing unsatisfactory results. Researchers confronted with problems of this nature increasingly resort to techniques that are heuristic and nonlinear. Such techniques include the use of NNR models.

The prediction of FX time series is one of the most challenging problems in forecasting. Our main motivation in this chapter is to determine whether regression models and, among these, NNR models can extract any more from the data than traditional techniques. Over the past few years, NNR models have provided an attractive alternative tool for researchers and analysts, claiming improved performance over traditional techniques. However, they have received less attention within financial areas than in other fields.

Typically, NNR models are optimised using a mathematical criterion, and subsequently analysed using similar measures. However, statistical measures are often inappropriate for financial applications. Evaluation using financial measures may be more appropriate, such as risk-adjusted measures of return. In essence, trading driven by a model with a small forecast error may not be as profitable as a model selected using financial criteria.

The motivation for this chapter is to determine the added value, or otherwise, of NNR models by benchmarking their results against traditional regression-based and other forecasting techniques. Accordingly, financial trading models are developed for the EUR/USD exchange rate, using daily data from 17 October 1994 to 18 May 2000 for in-sample estimation, leaving the period from 19 May 2000 to 3 July 2001 for out-of-sample forecasting.<sup>1</sup> The trading models are evaluated in terms of forecasting accuracy *and* in terms of trading performance via a simulated trading strategy.

---

<sup>1</sup> The EUR/USD exchange rate only exists from 4 January 1999: it was retropolated from 17 October 1994 to 31 December 1998 and a synthetic EUR/USD series was created for that period using the fixed EUR/DEM conversion rate agreed in 1998, combined with the USD/DEM daily market rate.

Our results clearly show that NNR models do indeed add value to the forecasting process.

The chapter is organised as follows. Section 1.2 presents a brief review of some of the research in FX markets. Section 1.3 describes the data used, addressing issues such as stationarity. Section 1.4 presents the benchmark models selected and our methodology. Section 1.5 briefly discusses NNR model theory and methodology, raising some issues surrounding the technique. Section 1.6 describes the out-of-sample forecasting accuracy and trading simulation results. Finally, Section 1.7 provides some concluding remarks.

## 1.2 LITERATURE REVIEW

It is outside the scope of this chapter to provide an exhaustive survey of all FX applications. However, we present a brief review of some of the material concerning financial applications of NNR models that began to emerge in the late 1980s.

Bellgard and Goldschmidt (1999) examined the forecasting accuracy and trading performance of several traditional techniques, including random walk, exponential smoothing, and ARMA models with Recurrent Neural Network (RNN) models.<sup>2</sup> The research was based on the Australian dollar to US dollar (AUD/USD) exchange rate using half hourly data during 1996. They conclude that statistical forecasting accuracy measures do not have a direct bearing on profitability, and FX time series exhibit nonlinear patterns that are better exploited by neural network models.

Tyree and Long (1995) disagree, finding the random walk model more effective than the NNR models examined. They argue that although price changes are not strictly random, in their case the US dollar to Deutsche Mark (USD/DEM) daily price changes from 1990 to 1994, from a forecasting perspective what little structure is actually present may well be too negligible to be of any use. They acknowledge that the random walk is unlikely to be the optimal forecasting technique. However, they do not assess the performance of the models financially.

The USD/DEM daily price changes were also the focus for Refenes and Zaidi (1993). However they use the period 1984 to 1992, and take a different approach. They developed a hybrid system for managing exchange rate strategies. The idea was to use a neural network model to predict which of a portfolio of strategies is likely to perform best in the current context. The evaluation was based upon returns, and concludes that the hybrid system is superior to the traditional techniques of moving averages and mean-reverting processes.

El-Shazly and El-Shazly (1997) examined the one-month forecasting performance of an NNR model compared with the forward rate of the British pound (GBP), German Mark (DEM), and Japanese yen (JPY) against a common currency, although they do not state which, using weekly data from 1988 to 1994. Evaluation was based on forecasting accuracy and in terms of correctly forecasting the direction of the exchange rate. Essentially, they conclude that neural networks outperformed the forward rate both in terms of accuracy and correctness.

Similar FX rates are the focus for Gençay (1999). He examined the predictability of daily spot exchange rates using four models applied to five currencies, namely the French franc (FRF), DEM, JPY, Swiss franc (CHF), and GBP against a common currency from

---

<sup>2</sup> A brief discussion of RNN models is presented in Section 1.5.

1973 to 1992. The models include random walk, GARCH(1,1), NNR models and nearest neighbours. The models are evaluated in terms of forecasting accuracy and correctness of sign. Essentially, he concludes that non-parametric models dominate parametric ones. Of the non-parametric models, nearest neighbours dominate NNR models.

Yao *et al.* (1996) also analysed the predictability of the GBP, DEM, JPY, CHF, and AUD against the USD, from 1984 to 1995, but using weekly data. However, they take an ARMA model as a benchmark. Correctness of sign and trading performance were used to evaluate the models. They conclude that NNR models produce a higher correctness of sign, and consequently produce higher returns, than ARMA models. In addition, they state that without the use of extensive market data or knowledge, useful predictions can be made and significant paper profit can be achieved.

Yao *et al.* (1997) examine the ability to forecast the daily USD/CHF exchange rate using data from 1983 to 1995. To evaluate the performance of the NNR model, “buy and hold” and “trend following” strategies were used as benchmarks. Again, the performance was evaluated through correctness of sign and via a trading simulation. Essentially, compared with the two benchmarks, the NNR model performed better and produced greater paper profit.

Carney and Cunningham (1996) used four data sets over the period 1979 to 1995 to examine the single-step and multi-step prediction of the weekly GBP/USD, daily GBP/USD, weekly DEM/SEK (Swedish krona) and daily GBP/DEM exchange rates. The neural network models were benchmarked by a naïve forecast and the evaluation was based on forecasting accuracy. The results were mixed, but concluded that neural network models are useful techniques that can make sense of complex data that defies traditional analysis.

A number of the successful forecasting claims using NNR models have been published. Unfortunately, some of the work suffers from inadequate documentation regarding methodology, for example El-Shazly and El-Shazly (1997), and Gençay (1999). This makes it difficult to both replicate previous work and obtain an accurate assessment of just how well NNR modelling techniques perform in comparison to other forecasting techniques, whether regression-based or not.

Notwithstanding, it seems pertinent to evaluate the use of NNR models as an alternative to traditional forecasting techniques, with the intention to ascertain their potential added value to this specific application, namely forecasting the EUR/USD exchange rate.

### **1.3 THE EXCHANGE RATE AND RELATED FINANCIAL DATA**

The FX market is perhaps the only market that is open 24 hours a day, seven days a week. The market opens in Australasia, followed by the Far East, the Middle East and Europe, and finally America. Upon the close of America, Australasia returns to the market and begins the next 24-hour cycle. The implication for forecasting applications is that in certain circumstances, because of time-zone differences, researchers should be mindful when considering which data and which subsequent time lags to include.

In any time series analysis it is critical that the data used is clean and error free since the learning of patterns is totally data-dependent. Also significant in the study of FX time series forecasting is the rate at which data from the market is sampled. The sampling frequency depends on the objectives of the researcher and the availability of data. For example, intraday time series can be extremely noisy and “a typical off-floor trader...

would most likely use daily data if designing a neural network as a component of an overall trading system” (Kaastra and Boyd, 1996: 220). For these reasons the time series used in this chapter are all daily closing data obtained from a historical database provided by Datastream.

The investigation is based on the London daily closing prices for the EUR/USD exchange rate.<sup>3</sup> In the absence of an indisputable theory of exchange rate determination, we assumed that the EUR/USD exchange rate could be explained by that rate’s recent evolution, volatility spillovers from other financial markets, and macro-economic and monetary policy expectations. With this in mind it seemed reasonable to include, as potential inputs, other leading traded exchange rates, the evolution of important stock and commodity prices, and, as a measure of macro-economic and monetary policy expectations, the evolution of the yield curve. The data retained is presented in Table 1.1 along with the relevant Datastream mnemonics, and can be reviewed in Sheet 1 of the DataAppendix.xls Excel spreadsheet.

**Table 1.1** Data and Datastream mnemonics

Number	Variable	Mnemonics
1	<i>FTSE 100 – PRICE INDEX</i>	FTSE100
2	<i>DAX 30 PERFORMANCE – PRICE INDEX</i>	DAXINDX
3	<i>S&amp;P 500 COMPOSITE – PRICE INDEX</i>	S&PCOMP
4	<i>NIKKEI 225 STOCK AVERAGE – PRICE INDEX</i>	JAPDOWA
5	<i>FRANCE CAC 40 – PRICE INDEX</i>	FRCAC40
6	<i>MILAN MIB 30 – PRICE INDEX</i>	ITMIB30
7	<i>DJ EURO STOXX 50 – PRICE INDEX</i>	DJES50I
8	<i>US EURO-\$ 3 MONTH (LDN:FT) – MIDDLE RATE</i>	ECUS\$3M
9	<i>JAPAN EURO-\$ 3 MONTH (LDN:FT) – MIDDLE RATE</i>	ECJAP3M
10	<i>EURO EURO-CURRENCY 3 MTH (LDN:FT) – MIDDLE RATE</i>	ECEUR3M
11	<i>GERMANY EURO-MARK 3 MTH (LDN:FT) – MIDDLE RATE</i>	ECWGM3M
12	<i>FRANCE EURO-FRANC 3 MTH (LDN:FT) – MIDDLE RATE</i>	ECFFR3M
13	<i>UK EURO-£ 3 MONTH (LDN:FT) – MIDDLE RATE</i>	ECUK£3M
14	<i>ITALY EURO-LIRE 3 MTH (LDN:FT) – MIDDLE RATE</i>	ECITL3M
15	<i>JAPAN BENCHMARK BOND-RYLD.10 YR (DS) – RED. YIELD</i>	JPBRYLD
16	<i>ECU BENCHMARK BOND 10 YR (DS) ‘DEAD’ – RED. YIELD</i>	ECBRYLD
17	<i>GERMANY BENCHMARK BOND 10 YR (DS) – RED. YIELD</i>	BDBRYLD
18	<i>FRANCE BENCHMARK BOND 10 YR (DS) – RED. YIELD</i>	FRBRYLD
19	<i>UK BENCHMARK BOND 10 YR (DS) – RED. YIELD</i>	UKMBRYLD
20	<i>US TREAS. BENCHMARK BOND 10 YR (DS) – RED. YIELD</i>	USBD10Y
21	<i>ITALY BENCHMARK BOND 10 YR (DS) – RED. YIELD</i>	ITBRYLD
22	<i>JAPANESE YEN TO US \$ (WMR) – EXCHANGE RATE</i>	JAPAYES
23	<i>US \$ TO UK £ (WMR) – EXCHANGE RATE</i>	USDOLLR
24	<i>US \$ TO EURO (WMR) – EXCHANGE RATE</i>	USEURSP
25	<i>Brent Crude-Current Month, fob US \$/BBL</i>	OILBREN
26	<i>GOLD BULLION \$/TROY OUNCE</i>	GOLDBLN
27	<i>Bridge/CRB Commodity Futures Index – PRICE INDEX</i>	NYFECRB

<sup>3</sup>EUR/USD is quoted as the number of USD per euro: for example, a value of 1.2657 is USD1.2657 per euro. The EUR/USD series for the period 1994–1998 was constructed as indicated in footnote 1.

All the series span the period from 17 October 1994 to 3 July 2001, totalling 1749 trading days. The data is divided into two periods: the first period runs from 17 October 1994 to 18 May 2000 (1459 observations) used for model estimation and is classified in-sample, while the second period from 19 May 2000 to 3 July 2001 (290 observations) is reserved for out-of-sample forecasting and evaluation. The division amounts to approximately 17% being retained for out-of-sample purposes.

Over the review period there has been an overall appreciation of the USD against the euro, as presented in Figure 1.1. The summary statistics of the EUR/USD for the examined period are presented in Figure 1.2, highlighting a slight skewness and low kurtosis. The Jarque–Bera statistic confirms that the EUR/USD series is non-normal at the 99% confidence interval. Therefore, the indication is that the series requires some type of transformation. The use of data in levels in the FX market has many problems, “FX price movements are generally non-stationary and quite random in nature, and therefore not very suitable for learning purposes. . . Therefore for most neural network studies and analysis concerned with the FX market, price inputs are not a desirable set” (Mehta, 1995: 191).

To overcome these problems, the EUR/USD series is transformed into rates of return. Given the price level  $P_1, P_2, \dots, P_t$ , the rate of return at time  $t$  is formed by:

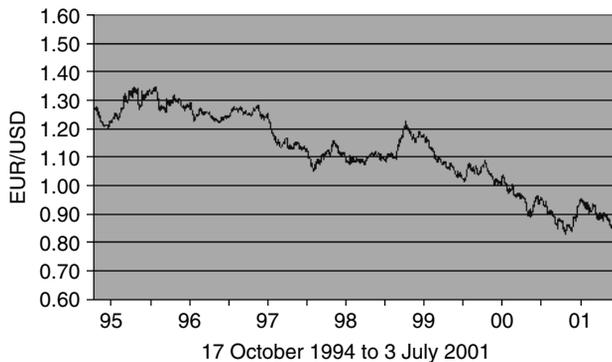
$$R_t = \left( \frac{P_t}{P_{t-1}} \right) - 1 \quad (1.1)$$

An example of this transformation can be reviewed in Sheet 1 column C of the oos\_Naïve.xls Excel spreadsheet, and is also presented in Figure 1.5. See also the comment in cell C4 for an explanation of the calculations within this column.

An advantage of using a returns series is that it helps in making the time series stationary, a useful statistical property.

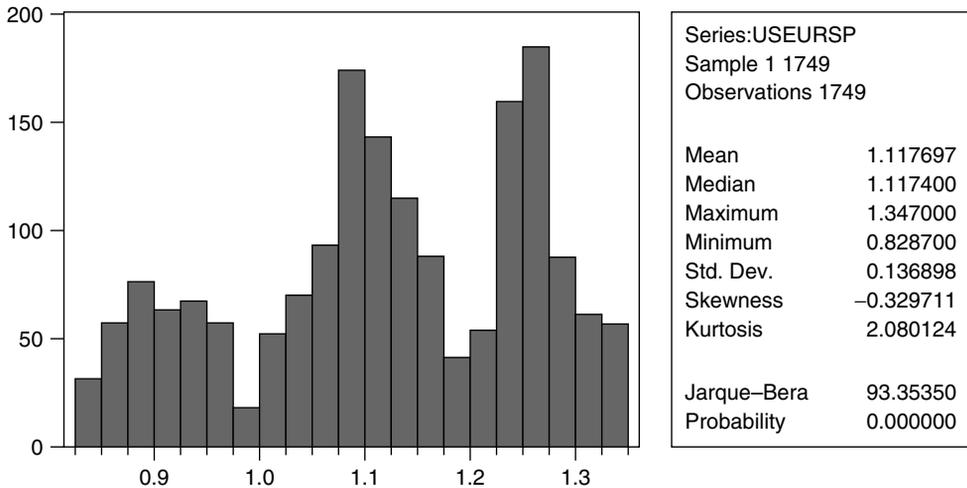
Formal confirmation that the EUR/USD returns series is stationary is confirmed at the 1% significance level by both the Augmented Dickey–Fuller (ADF) and Phillips–Perron (PP) test statistics, the results of which are presented in Tables 1.2 and 1.3.

The EUR/USD returns series is presented in Figure 1.3. Transformation into returns often creates a noisy time series. Formal confirmation through testing the significance of



**Figure 1.1** EUR/USD London daily closing prices (17 October 1994 to 3 July 2001)<sup>4</sup>

<sup>4</sup> Retropolated series for 17 October 1994 to 31 December 1998.



**Figure 1.2** EUR/USD summary statistics (17 October 1994 to 3 July 2001)

**Table 1.2** EUR/USD returns ADF test

ADF test statistic	<b>-18.37959</b>	1%	critical value <sup>a</sup>	<b>-3.4371</b>
		5%	critical value	-2.8637
		10%	critical value	-2.5679

<sup>a</sup>MacKinnon critical values for rejection of hypothesis of a unit root.

Augmented Dickey-Fuller Test Equation

Dependent Variable: D(DR\_USEURSP)

Method: Least Squares

Sample(adjusted): 7 1749

Included observations: 1743 after adjusting endpoints

Variable	Coefficient	Std. error	t-Statistic	Prob.
DR_USEURSP(-1)	-0.979008	0.053266	-18.37959	0.0000
D(DR_USEURSP(-1))	-0.002841	0.047641	-0.059636	0.9525
D(DR_USEURSP(-2))	-0.015731	0.041288	-0.381009	0.7032
D(DR_USEURSP(-3))	-0.011964	0.033684	-0.355179	0.7225
D(DR_USEURSP(-4))	-0.014248	0.024022	-0.593095	0.5532
C	-0.000212	0.000138	-1.536692	0.1246
<i>R</i> -squared	0.491277	Mean dependent var.		1.04E-06
Adjusted <i>R</i> -squared	0.489812	S.D. dependent var.		0.008048
S.E. of regression	0.005748	Akaike info. criterion		-7.476417
Sum squared resid.	0.057394	Schwarz criterion		-7.457610
Log likelihood	6521.697	<i>F</i> -statistic		335.4858
Durbin-Watson stat.	1.999488	Prob( <i>F</i> -statistic)		0.000000

**Table 1.3** EUR/USD returns PP test

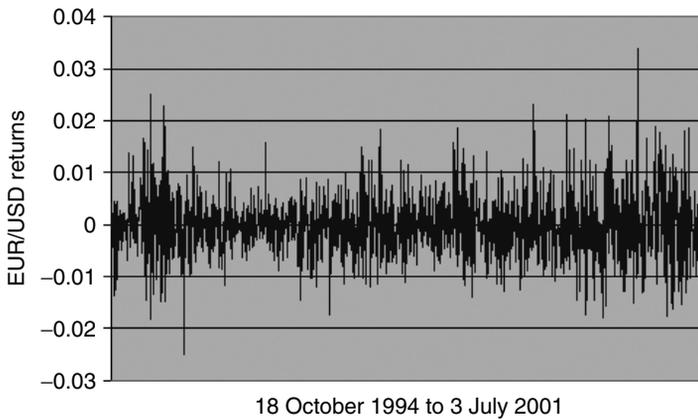
PP test statistic	<b>-41.04039</b>	1%	critical value <sup>a</sup>	<b>-3.4370</b>
		5%	critical value	-2.8637
		10%	critical value	-2.5679

<sup>a</sup>MacKinnon critical values for rejection of hypothesis of a unit root.

Lag truncation for Bartlett kernel: 7	(Newey–West suggests: 7)
Residual variance with no correction	3.29E-05
Residual variance with correction	3.26E-05

Phillips–Perron Test Equation  
 Dependent Variable: D(DR\_USEURSP)  
 Method: Least Squares  
 Sample(adjusted): 3 1749  
 Included observations: 1747 after adjusting endpoints

Variable	Coefficient	Std. error	<i>t</i> -Statistic	Prob.
DR_USEURSP(-1)	-0.982298	0.023933	-41.04333	0.0000
C	-0.000212	0.000137	-1.539927	0.1238
<i>R</i> -squared	0.491188	Mean dependent var.		-1.36E-06
Adjusted <i>R</i> -squared	0.490896	S.D. dependent var.		0.008041
S.E. of regression	0.005737	Akaike info. criterion		-7.482575
Sum squared resid.	0.057436	Schwarz criterion		-7.476318
Log likelihood	6538.030	<i>F</i> -statistic		1684.555
Durbin–Watson stat.	1.999532	Prob( <i>F</i> -statistic)		0.000000



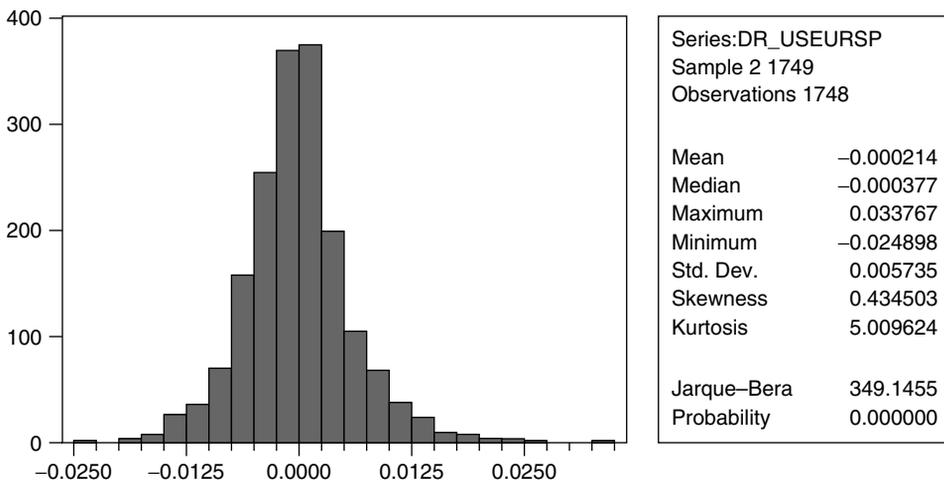
**Figure 1.3** The EUR/USD returns series (18 October 1994 to 3 July 2001)

the autocorrelation coefficients reveals that the EUR/USD returns series is white noise at the 99% confidence interval, the results of which are presented in Table 1.4. For such series the best predictor of a future value is zero. In addition, very noisy data often makes forecasting difficult.

The EUR/USD returns summary statistics for the examined period are presented in Figure 1.4. They reveal a slight skewness and high kurtosis and, again, the Jarque–Bera statistic confirms that the EUR/USD series is non-normal at the 99% confidence interval. However, such features are “common in high frequency financial time series data” (Gençay, 1999: 94).

**Table 1.4** EUR/USD returns correlogram

	Autocorrelation	Partial correlation	Q-Stat.	Prob.
1	0.018	0.018	0.5487	<b>0.459</b>
2	-0.012	-0.013	0.8200	<b>0.664</b>
3	0.003	0.004	0.8394	<b>0.840</b>
4	-0.002	-0.002	0.8451	<b>0.932</b>
5	0.014	0.014	1.1911	<b>0.946</b>
6	-0.009	-0.010	1.3364	<b>0.970</b>
7	0.007	0.008	1.4197	<b>0.985</b>
8	-0.019	-0.019	2.0371	<b>0.980</b>
9	0.001	0.002	2.0405	<b>0.991</b>
10	0.012	0.012	2.3133	<b>0.993</b>
11	0.012	0.012	2.5787	<b>0.995</b>
12	-0.028	-0.029	3.9879	<b>0.984</b>



**Figure 1.4** EUR/USD returns summary statistics (17 October 1994 to 3 July 2001)

A further transformation includes the creation of interest rate yield curve series, generated by:

$$yc = 10 \text{ year benchmark bond yields} - 3 \text{ month interest rates} \quad (1.2)$$

In addition, all of the time series are transformed into returns series in the manner described above to account for their non-stationarity.

## 1.4 BENCHMARK MODELS: THEORY AND METHODOLOGY

The premise of this chapter is to examine the use of regression models in EUR/USD forecasting and trading models. In particular, the performance of NNR models is compared with other traditional forecasting techniques to ascertain their potential added value as a forecasting tool. Such methods include ARMA modelling, logit estimation, Moving Average Convergence/Divergence (MACD) technical models, and a naïve strategy. Except for the straightforward naïve strategy, all benchmark models were estimated on our in-sample period. As all of these methods are well documented in the literature, they are simply outlined below.

### 1.4.1 Naïve strategy

The naïve strategy simply assumes that the most recent period change is the best predictor of the future. The simplest model is defined by:

$$\hat{Y}_{t+1} = Y_t \quad (1.3)$$

where  $Y_t$  is the actual rate of return at period  $t$  and  $\hat{Y}_{t+1}$  is the forecast rate of return for the next period.

The naïve forecast can be reviewed in Sheet 1 column E of the oos.Naïve.xls Excel spreadsheet, and is also presented in Figure 1.5. Also, please note the comments within the spreadsheet that document the calculations used within the naïve, ARMA, logit, and NNR strategies.

The performance of the strategy is evaluated in terms of forecasting accuracy and in terms of trading performance via a simulated trading strategy.

### 1.4.2 MACD strategy

Moving average methods are considered quick and inexpensive and as a result are routinely used in financial markets. The techniques use an average of past observations to smooth short-term fluctuations. In essence, “a moving average is obtained by finding the mean for a specified set of values and then using it to forecast the next period” (Hanke and Reitsch, 1998: 143).

The moving average is defined as:

$$M_t = \hat{Y}_{t+1} = \frac{(Y_t + Y_{t-1} + Y_{t-2} + \dots + Y_{t-n+1})}{n} \quad (1.4)$$

	A	B	C	D	E	F	G	H	I	J	K
	Trading Days	USD to 1 Euro	Daily returns	Rise or Fall	Naive Forecast	Trading Signal	# Transactions	Positions	Last Entry	Transaction Profit/Loss	Winning Trades
5	19/05/2000	0.8928	-0.10%	0	0.30%	1		0	0.8928	0	0
6	22/05/2000	0.904	1.25%	1	-0.10%	1		0	-0.904	0.0112	0
7	23/05/2000	0.9088	0.53%	1	1.25%	1		0	0.9088	-0.0048	0
8	24/05/2000	0.9074	-0.15%	0	0.53%	1		0	0.9088	0	0
9	25/05/2000	0.9069	-0.06%	0	-0.15%	1		0	-0.9069	-0.0019	0
10	26/05/2000	0.9259	2.10%	1	-0.06%	-1		0	-0.9069	0	0
11	29/05/2000	0.9277	0.19%	1	2.10%	1	1	0.9277	0.9277	-0.0208	0
12	30/05/2000	0.9325	0.52%	1	0.19%	1		0	0.9277	0	0
13	31/05/2000	0.9274	-0.55%	0	0.52%	1		0	0.9277	0	0
14	01/06/2000	0.9311	0.40%	1	-0.55%	-1	1	-0.9311	-0.9311	0.0034	1
15	02/06/2000	0.9442	1.41%	1	0.40%	1	1	0.9442	0.9442	-0.0131	0
16	05/06/2000	0.9445	0.03%	1	1.41%	1		0	0.9442	0	0
17	06/06/2000	0.9589	1.52%	1	0.03%	1		0	0.9442	0	0
18	07/06/2000	0.9571	-0.19%	0	1.52%	1		0	0.9442	0	0
19	08/06/2000	0.9535	-0.38%	0	-0.19%	-1	1	-0.9535	-0.9535	0.0093	1
20	09/06/2000	0.951	-0.26%	0	-0.38%	-1		0	-0.9535	0	0
21	12/06/2000	0.9539	0.30%	1	-0.26%	-1		0	-0.9535	0	0
22	13/06/2000	0.9607	0.71%	1	0.30%	1	1	0.9607	0.9607	-0.0072	0
23	14/06/2000	0.959	-0.18%	0	0.71%	1		0	0.9607	0	0
24	15/06/2000	0.9526	-0.67%	0	-0.18%	-1	1	-0.9526	-0.9526	-0.0081	0
25	16/06/2000	0.9608	0.86%	1	-0.67%	-1		0	-0.9526	0	0
26	19/06/2000	0.9603	-0.05%	0	0.86%	1	1	0.9603	0.9603	-0.0077	0
27	20/06/2000	0.9554	-0.51%	0	-0.05%	-1	1	-0.9554	-0.9554	-0.0049	0
28	21/06/2000	0.9462	-0.96%	0	-0.51%	-1		0	-0.9554	0	0
29	22/06/2000	0.9395	-0.71%	0	-0.96%	-1		0	-0.9554	0	0
30	23/06/2000	0.9365	-0.32%	0	-0.71%	-1		0	-0.9554	0	0
31	26/06/2000	0.9371	0.06%	1	-0.32%	-1		0	-0.9554	0	0
32	27/06/2000	0.9433	0.66%	1	0.06%	1	1	0.9433	0.9433	0.0121	1
33	28/06/2000	0.9459	0.28%	1	0.66%	1		0	0.9433	0	0
34	29/06/2000	0.9525	0.70%	1	0.28%	1		0	0.9433	0	0

Figure 1.5 Naive forecast Excel spreadsheet (out-of-sample)

where  $M_t$  is the moving average at time  $t$ ,  $n$  is the number of terms in the moving average,  $Y_t$  is the actual level at period  $t$ <sup>5</sup> and  $\hat{Y}_{t+1}$  is the level forecast for the next period.

The MACD strategy used is quite simple. Two moving average series  $M_{1,t}$  and  $M_{2,t}$  are created with different moving average lengths  $n$  and  $m$ . The decision rule for taking positions in the market is straightforward. If the short-term moving average (SMA) intersects the long-term moving average (LMA) from below a “long” position is taken. Conversely, if the LMA is intersected from above a “short” position is taken.<sup>6</sup> This strategy can be reviewed in Sheet 1 column E of the is\_35&1MA.xls Excel spreadsheet, and is also presented in Figure 1.6. Again, please note the comments within the spreadsheet that document the calculations used within the MACD strategy.

The forecaster must use judgement when determining the number of periods  $n$  and  $m$  on which to base the moving averages. The combination that performed best over the in-sample period was retained for out-of-sample evaluation. The model selected was a combination of the EUR/USD series and its 35-day moving average, namely  $n = 1$  and  $m = 35$  respectively, or a (1,35) combination. A graphical representation of the combination is presented in Figure 1.7. The performance of this strategy is evaluated in terms of forecasting accuracy via the correct directional change measure, and in terms of trading performance.

Several other “adequate” models were produced and their performance evaluated. The trading performance of some of these combinations, such as the (1,40) combination, and

<sup>5</sup> In this strategy the EUR/USD levels series is used as opposed to the returns series.

<sup>6</sup> A “long” EUR/USD position means buying euros at the current price, while a “short” position means selling euros at the current price.

	A	B	C	D	E	F	G	H	I	J	K	L
1				Daily returns	Trading	# Transactions			Transaction	Winning		Daily
2	Trading Days	USD to 1 Euro	Daily returns	Rise or Fall	Signal	Calculation	Positions	Last Entry	Profit/Loss	Trades	Daily Change	Pr
37	02/12/1994	1.2093	-0.44%	0	-1	1	-1.2093	-1.2093	0	0	-0.0063	
38	05/12/1994	1.212	0.22%	1	-1		0	-1.2093	0		0.0027	-0.0
39	06/12/1994	1.2141	0.17%	1	-1		0	-1.2093	0		0.0021	-0.0
40	07/12/1994	1.2143	0.02%	1	-1		0	-1.2093	0		0.0002	-0.0
41	08/12/1994	1.2105	-0.31%	0	-1		0	-1.2093	0		-0.0038	0.0
42	09/12/1994	1.2113	0.07%	1	-1		0	-1.2093	0		0.0008	-0.0
43	12/12/1994	1.213	0.14%	1	-1		0	-1.2093	0		0.0017	-0.0
44	13/12/1994	1.2115	-0.12%	0	-1		0	-1.2093	0		-0.0015	0.0
45	14/12/1994	1.2142	0.22%	1	-1		0	-1.2093	0		0.0027	-0.0
46	15/12/1994	1.2127	-0.12%	0	-1		0	-1.2093	0		-0.0015	0.0
47	16/12/1994	1.2132	0.04%	1	-1		0	-1.2093	0		0.0005	-0.0
48	19/12/1994	1.211	-0.18%	0	-1		0	-1.2093	0		-0.0022	0.0
49	20/12/1994	1.2119	0.07%	1	-1		0	-1.2093	0		0.0009	-0.0
50	21/12/1994	1.2127	0.07%	1	-1		0	-1.2093	0		0.0008	-0.0
51	22/12/1994	1.2045	-0.68%	0	-1		0	-1.2093	0		-0.0082	0.0
52	23/12/1994	1.2058	0.11%	1	-1		0	-1.2093	0		0.0013	-0.0
53	26/12/1994	1.2058	0.00%	0	-1		0	-1.2093	0		0	-0.0
54	27/12/1994	1.2062	0.03%	1	-1		0	-1.2093	0		0.0004	-0.0
55	28/12/1994	1.2083	0.17%	1	-1		0	-1.2093	0		0.0021	-0.0
56	29/12/1994	1.225	1.36%	1	1		0	1.225	0	0	0.0167	-0.0
57	30/12/1994	1.2264	0.11%	1	1		0	1.225	0	0	0.0014	0.0
58	02/01/1995	1.2264	0.00%	0	1		0	1.225	0	0	0	-0.0
59	03/01/1995	1.2244	-0.16%	0	1		0	1.225	0	0	-0.002	-0.0
60	04/01/1995	1.221	-0.28%	0	1		0	1.225	0	0	-0.0034	-0.0
61	05/01/1995	1.2256	0.36%	1	1		0	1.225	0	0	0.0046	0.0
62	06/01/1995	1.2202	-0.44%	0	1		0	1.225	0	0	-0.0054	-0.0
63	09/01/1995	1.2229	0.22%	1	1		0	1.225	0	0	0.0027	0.0
64	10/01/1995	1.2326	0.81%	1	1		0	1.225	0	0	0.0099	0.0
65	11/01/1995	1.2326	-0.02%	0	1		0	1.225	0	0	-0.0002	-0.0
66	12/01/1995	1.2361	0.28%	1	1		0	1.225	0	0	0.0035	0.0

Figure 1.6 EUR/USD and 35-day moving average combination Excel spreadsheet

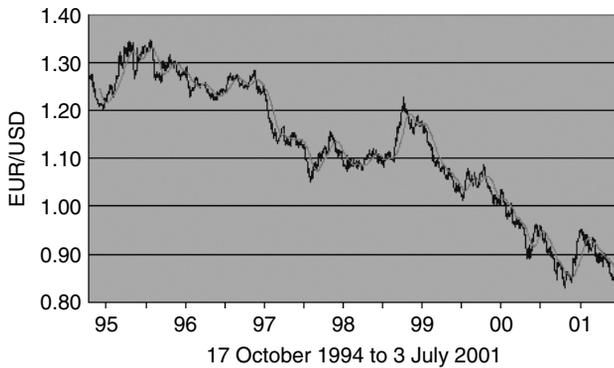


Figure 1.7 EUR/USD and 35-day moving average combination

the (1,35) combination results were only marginally different. For example, the Sharpe ratio differs only by 0.01, and the average gain/loss ratio by 0.02. However, the (1,35) combination has the lowest maximum drawdown at  $-12.43\%$  and lowest probability of a 10% loss at 0.02%.<sup>7</sup> The evaluation can be reviewed in Sheet 2 of the is\_35&1MA.xls and is\_40&1MA.xls Excel spreadsheets, and is also presented in Figures 1.8 and 1.9,

<sup>7</sup> A discussion of the statistical and trading performance measures used to evaluate the strategies is presented below in Section 1.6.

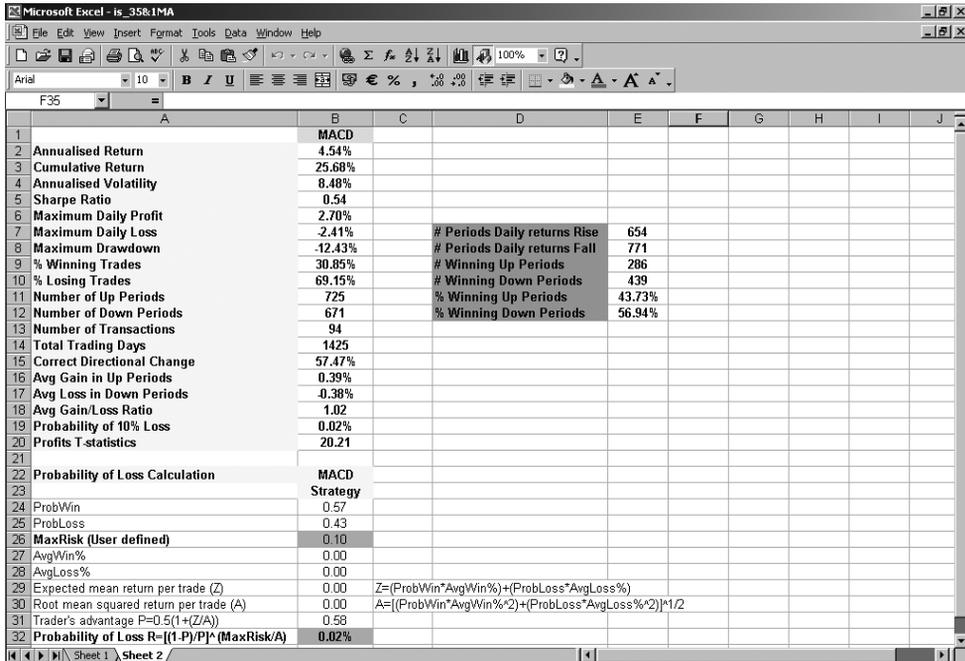


Figure 1.8 (1,35) combination moving average Excel spreadsheet (in-sample)

respectively. On balance, the (1,35) combination was considered “best” and therefore retained for further analysis.

### 1.4.3 ARMA methodology

ARMA models are particularly useful when information is limited to a single stationary series,<sup>8</sup> or when economic theory is not useful. They are a “highly refined curve-fitting device that uses current and past values of the dependent variable to produce accurate short-term forecasts” (Hanke and Reitsch, 1998: 407).

The ARMA methodology does not assume any particular pattern in a time series, but uses an iterative approach to identify a possible model from a general class of models. Once a tentative model has been selected, it is subjected to tests of adequacy. If the specified model is not satisfactory, the process is repeated using other models until a satisfactory model is found. Sometimes, it is possible that two or more models may approximate the series equally well, in this case the most parsimonious model should prevail. For a full discussion on the procedure refer to Box *et al.* (1994), Gouriéroux and Monfort (1995), or Pindyck and Rubinfeld (1998).

The ARMA model takes the form:

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \varepsilon_t - w_1 \varepsilon_{t-1} - w_2 \varepsilon_{t-2} - \dots - w_q \varepsilon_{t-q} \tag{1.5}$$

<sup>8</sup> The general class of ARMA models is for stationary time series. If the series is not stationary an appropriate transformation is required.

	A	B	C	D	E	F	G	H	I	J
1		MACD								
2	Annualised Return	4.49%								
3	Cumulative Return	25.28%								
4	Annualised Volatility	8.50%								
5	Sharpe Ratio	0.53								
6	Maximum Daily Profit	2.69%								
7	Maximum Daily Loss	-2.41%		# Periods Daily returns Rise	651					
8	Maximum Drawdown	-14.54%		# Periods Daily returns Fall	769					
9	% Winning Trades	30.49%		# Winning Up Periods	290					
10	% Losing Trades	69.51%		# Winning Down Periods	439					
11	Number of Up Periods	729		% Winning Up Periods	44.55%					
12	Number of Down Periods	662		% Winning Down Periods	57.09%					
13	Number of Transactions	82								
14	Total Trading Days	1420								
15	Correct Directional Change	57.04%								
16	Avg Gain in Up Periods	0.38%								
17	Avg Loss in Down Periods	-0.38%								
18	Avg Gain/Loss Ratio	1.00								
19	Probability of 10% Loss	0.06%								
20	Profits T statistics	19.89								
21										
22	Probability of Loss Calculation	MACD								
23		Strategy								
24	ProbWin	0.57								
25	ProbLoss	0.43								
26	MaxRisk (User defined)	0.10								
27	AvgWin%	0.00								
28	AvgLoss%	0.00								
29	Expected mean return per trade (Z)	0.00		Z=(ProbWin*AvgWin%)+(ProbLoss*AvgLoss%)						
30	Root mean squared return per trade (A)	0.00		A=[(ProbWin*AvgWin% <sup>2</sup> )+(ProbLoss*AvgLoss% <sup>2</sup> )] <sup>1/2</sup>						
31	Trader's advantage P=0.5*(1+(Z/A))	0.57								
32	Probability of Loss R=[(1-P)/P]*(MaxRisk/A)	0.06%								

Figure 1.9 (1,40) combination moving average Excel spreadsheet (in-sample)

where  $Y_t$  is the dependent variable at time  $t$ ;  $Y_{t-1}, Y_{t-2}, \dots, Y_{t-p}$  are the lagged dependent variables;  $\phi_0, \phi_1, \dots, \phi_p$  are regression coefficients;  $\varepsilon_t$  is the residual term;  $\varepsilon_{t-1}, \varepsilon_{t-2}, \dots, \varepsilon_{t-p}$  are previous values of the residual;  $w_1, w_2, \dots, w_q$  are weights.

Several ARMA specifications were tried out, for example ARMA(5,5) and ARMA(10,10) models were produced to test for any “weekly” effects, which can be reviewed in the arma.wf1 EViews workfile. The ARMA(10,10) model was estimated but was unsatisfactory as several coefficients were not even significant at the 90% confidence interval (equation arma1010). The results of this are presented in Table 1.5. The model was primarily modified through testing the significance of variables via the likelihood ratio (LR) test for redundant or omitted variables and Ramsey’s RESET test for model misspecification.

Once the non-significant terms are removed all of the coefficients of the restricted ARMA(10,10) model become significant at the 99% confidence interval (equation arma13610). The overall significance of the model is tested using the  $F$ -test. The null hypothesis that all coefficients except the constant are not significantly different from zero is rejected at the 99% confidence interval. The results of this are presented in Table 1.6. Examination of the autocorrelation function of the error terms reveals that the residuals are random at the 99% confidence interval and a further confirmation is given by the serial correlation LM test. The results of this are presented in Tables 1.7 and 1.8. The model is also tested for general misspecification via Ramsey’s RESET test. The null hypothesis of correct specification is accepted at the 99% confidence interval. The results of this are presented in Table 1.9.

**Table 1.5** ARMA(10,10) EUR/USD returns estimation

Dependent Variable: DR\_USEURSP

Method: Least Squares

Sample(adjusted): 12 1459

Included observations: 1448 after adjusting endpoints

Convergence achieved after 20 iterations

White Heteroskedasticity-Consistent Standard Errors &amp; Covariance

Backcast: 2 11

Variable	Coefficient	Std. error	<i>t</i> -Statistic	Prob.
C	-0.000220	0.000140	-1.565764	0.1176
AR(1)	-0.042510	0.049798	-0.853645	<b>0.3934</b>
AR(2)	-0.210934	0.095356	-2.212073	0.0271
AR(3)	-0.359378	0.061740	-5.820806	0.0000
AR(4)	-0.041003	0.079423	-0.516264	<b>0.6058</b>
AR(5)	0.001376	0.067652	0.020338	<b>0.9838</b>
AR(6)	0.132413	0.054071	2.448866	0.0145
AR(7)	-0.238913	0.052594	-4.542616	0.0000
AR(8)	0.182816	0.046878	3.899801	0.0001
AR(9)	0.026431	0.060321	0.438169	<b>0.6613</b>
AR(10)	-0.615601	0.076171	-8.081867	0.0000
MA(1)	0.037787	0.040142	0.941343	<b>0.3467</b>
MA(2)	0.227952	0.095346	2.390785	0.0169
MA(3)	0.341293	0.058345	5.849551	0.0000
MA(4)	0.036997	0.074796	0.494633	<b>0.6209</b>
MA(5)	-0.004544	0.059140	-0.076834	<b>0.9388</b>
MA(6)	-0.140714	0.046739	-3.010598	0.0027
MA(7)	0.253016	0.042340	5.975838	0.0000
MA(8)	-0.206445	0.040077	-5.151153	0.0000
MA(9)	-0.014011	0.048037	-0.291661	<b>0.7706</b>
MA(10)	0.643684	0.074271	8.666665	0.0000
<i>R</i> -squared	0.016351	Mean dependent var.		-0.000225
Adjusted <i>R</i> -squared	0.002565	S.D. dependent var.		0.005363
S.E. of regression	0.005356	Akaike info. criterion		-7.606665
Sum squared resid.	0.040942	Schwarz criterion		-7.530121
Log likelihood	5528.226	<i>F</i> -statistic		1.186064
Durbin-Watson stat.	1.974747	Prob( <i>F</i> -statistic)		0.256910
Inverted AR roots	0.84 + 0.31i 0.07 + 0.98i -0.90 + 0.21i	0.84 - 0.31i 0.07 - 0.98i -0.90 - 0.21i	0.55 - 0.82i -0.59 - 0.78i	0.55 + 0.82i -0.59 + 0.78i
Inverted MA roots	0.85 + 0.31i 0.07 - 0.99i -0.90 + 0.20i	0.85 - 0.31i 0.07 + 0.99i -0.90 - 0.20i	0.55 - 0.82i -0.59 - 0.79i	0.55 + 0.82i -0.59 + 0.79i

**Table 1.6** Restricted ARMA(10,10) EUR/USD returns estimation

Dependent Variable: DR\_USEURSP  
 Method: Least Squares  
 Sample(adjusted): 12 1459  
 Included observations: 1448 after adjusting endpoints  
 Convergence achieved after 50 iterations  
 White Heteroskedasticity–Consistent Standard Errors & Covariance  
 Backcast: 2 11

Variable	Coefficient	Std. error	<i>t</i> -Statistic	Prob.
C	−0.000221	0.000144	−1.531755	0.1258
AR(1)	0.263934	0.049312	5.352331	<b>0.0000</b>
AR(3)	−0.444082	0.040711	−10.90827	<b>0.0000</b>
AR(6)	−0.334221	0.035517	−9.410267	<b>0.0000</b>
AR(10)	−0.636137	0.043255	−14.70664	<b>0.0000</b>
MA(1)	−0.247033	0.046078	−5.361213	<b>0.0000</b>
MA(3)	0.428264	0.030768	13.91921	<b>0.0000</b>
MA(6)	0.353457	0.028224	12.52307	<b>0.0000</b>
MA(10)	0.675965	0.041063	16.46159	<b>0.0000</b>
<i>R</i> -squared	0.015268	Mean dependent var.		−0.000225
Adjusted <i>R</i> -squared	0.009793	S.D. dependent var.		0.005363
S.E. of regression	0.005337	Akaike info. criterion		−7.622139
Sum squared resid.	0.040987	Schwarz criterion		−7.589334
Log likelihood	5527.429	<i>F</i> -statistic		2.788872
Durbin–Watson stat.	2.019754	Prob( <i>F</i> -statistic)		<b>0.004583</b>
Inverted AR roots	0.89 + 0.37i 0.08 − 0.98i −0.92 + 0.31i	0.89 − 0.37i 0.08 + 0.98i −0.92 − 0.31i	0.61 + 0.78i −0.53 − 0.70i	0.61 − 0.78i −0.53 + 0.70i
Inverted MA roots	0.90 − 0.37i 0.07 + 0.99i −0.93 + 0.31i	0.90 + 0.37i 0.07 − 0.99i −0.93 − 0.31i	0.61 + 0.78i −0.54 − 0.70i	0.61 − 0.78i −0.54 + 0.70i

The selected ARMA model, namely the restricted ARMA(10,10) model, takes the form:

$$Y_t = -0.0002 + 0.2639Y_{t-1} - 0.4440Y_{t-3} - 0.3342Y_{t-6} - 0.6361Y_{t-10} \\ - 0.2470\varepsilon_{t-1} + 0.4283\varepsilon_{t-3} + 0.3535\varepsilon_{t-6} + 0.6760\varepsilon_{t-10}$$

The restricted ARMA(10,10) model was retained for out-of-sample estimation. The performance of the strategy is evaluated in terms of traditional forecasting accuracy and in terms of trading performance. Several other models were produced and their performance evaluated, for example an alternative restricted ARMA(10,10) model was produced (equation arma16710). The decision to retain the original restricted ARMA(10,10) model is because it has significantly better in-sample trading results than the alternative ARMA(10,10) model. The annualised return, Sharpe ratio and correct directional change of the original model were 12.65%, 1.49 and 53.80%, respectively. The corresponding

**Table 1.7** Restricted ARMA(10,10) correlogram of residuals

Sample: 12 1459

Included observations: 1448

 $Q$ -statistic probabilities adjusted for 8 ARMA term(s)

	Autocorrelation	Partial correlation	$Q$ -Stat.	Prob.
1	-0.010	-0.010	0.1509	
2	-0.004	-0.004	0.1777	
3	0.004	0.004	0.1973	
4	-0.001	-0.001	0.1990	
5	0.000	0.000	0.1991	
6	-0.019	-0.019	0.7099	
7	-0.004	-0.004	0.7284	
8	-0.015	-0.015	1.0573	
9	0.000	0.000	1.0573	<b>0.304</b>
10	0.009	0.009	1.1824	<b>0.554</b>
11	0.031	0.032	2.6122	<b>0.455</b>
12	-0.024	-0.024	3.4600	<b>0.484</b>
13	0.019	0.018	3.9761	<b>0.553</b>
14	-0.028	-0.028	5.0897	<b>0.532</b>
15	0.008	0.008	5.1808	<b>0.638</b>

values for the alternative model were 9.47%, 1.11 and 52.35%. The evaluation can be reviewed in Sheet 2 of the *is\_arma13610.xls* and *is\_arma16710.xls* Excel spreadsheets, and is also presented in Figures 1.10 and 1.11, respectively. Ultimately, we chose the model that satisfied the usual statistical tests and that also recorded the best in-sample trading performance.

#### 1.4.4 Logit estimation

The logit model belongs to a group of models termed “classification models”. They are a multivariate statistical technique used to estimate the probability of an upward or downward movement in a variable. As a result they are well suited to rates of return applications where a recommendation for trading is required. For a full discussion of the procedure refer to Maddala (2001), Pesaran and Pesaran (1997), or Thomas (1997).

The approach assumes the following regression model:

$$Y_t^* = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \dots + \beta_p X_{p,t} + \varepsilon_t \quad (1.6)$$

where  $Y_t^*$  is the dependent variable at time  $t$ ;  $X_{1,t}, X_{2,t}, \dots, X_{p,t}$  are the explanatory variables at time  $t$ ;  $\beta_0, \beta_1, \dots, \beta_p$  are the regression coefficients;  $\varepsilon_t$  is the residual term.

However,  $Y_t^*$  is not directly observed; what is observed is a dummy variable  $Y_t$  defined by:

$$Y_t = \begin{cases} 1 & \text{if } Y_t^* > 0 \\ 0 & \text{otherwise} \end{cases} \quad (1.7)$$

Therefore, the model requires a transformation of the explained variable, namely the EUR/USD returns series into a binary series. The procedure is quite simple: a binary

**Table 1.8** Restricted ARMA(10,10) serial correlation LM test

Breusch–Godfrey Serial Correlation LM Test				
<i>F</i> -statistic	0.582234	Probability		<b>0.558781</b>
Obs* <i>R</i> -squared	1.172430	Probability		<b>0.556429</b>
Dependent Variable: RESID				
Method: Least Squares				
Presample missing value lagged residuals set to zero				
Variable	Coefficient	Std. error	<i>t</i> -Statistic	Prob.
C	8.33E-07	0.000144	0.005776	0.9954
AR(1)	0.000600	0.040612	0.014773	0.9882
AR(3)	0.019545	0.035886	0.544639	0.5861
AR(6)	0.018085	0.031876	0.567366	0.5706
AR(10)	-0.028997	0.037436	-0.774561	0.4387
MA(1)	-0.000884	0.038411	-0.023012	0.9816
MA(3)	-0.015096	0.026538	-0.568839	0.5696
MA(6)	-0.014584	0.026053	-0.559792	0.5757
MA(10)	0.029482	0.035369	0.833563	0.4047
RESID(-1)	-0.010425	0.031188	-0.334276	0.7382
RESID(-2)	-0.004640	0.026803	-0.173111	0.8626
<i>R</i> -squared	0.000810	Mean dependent var.		1.42E-07
Adjusted <i>R</i> -squared	-0.006144	S.D. dependent var.		0.005322
S.E. of regression	0.005338	Akaike info. criterion		-7.620186
Sum squared resid.	0.040953	Schwarz criterion		-7.580092
Log likelihood	5528.015	<i>F</i> -statistic		0.116447
Durbin–Watson stat.	1.998650	Prob( <i>F</i> -statistic)		0.999652

**Table 1.9** Restricted ARMA(10,10) RESET test for model misspecification

Ramsey RESET Test			
<i>F</i> -statistic	0.785468	Probability	0.375622
Log likelihood ratio	0.790715	Probability	<b>0.373884</b>

variable equal to one is produced if the return is positive, and zero otherwise. The same transformation for the explanatory variables, although not necessary, was performed for homogeneity reasons.

A basic regression technique is used to produce the logit model. The idea is to start with a model containing several variables, including lagged dependent terms, then through a series of tests the model is modified.

The selected logit model, which we shall name logit1 (equation logit1 of the logit.wf1 EViews workfile), takes the form:

	A	B	C	D	E	F	G	H	I
1		ARMA Strategy							
2	Annualised Return	12.65%		Mean Absolute Error	0.0039				
3	Cumulative Return	72.67%		Mean Absolute Percentage Error	110.34%				
4	Annualised Volatility	8.48%		Root Mean Squared Error	0.0053				
5	Sharpe Ratio	1.49		Theil's Inequality Coefficient	0.8721				
6	Maximum Daily Profit	2.49%							
7	Maximum Daily Loss	-2.51%		# Periods Daily returns Rise	663				
8	Maximum Drawdown	-9.62%		# Periods Daily returns Fall	785				
9	% Winning trades	50.24%		# Winning Up Periods	243				
10	% Losing trades	49.76%		# Winning Down Periods	536				
11	Number of Up Periods	779		% Winning Up Periods	36.65%				
12	Number of Down Periods	641		% Winning Down Periods	68.28%				
13	Number of Transactions	613							
14	Total Trading Days	1448							
15	Correct Directional Change	53.80%							
16	Avg Gain in Up Periods	0.41%							
17	Avg Loss in Down Periods	-0.39%							
18	Avg Gain/Loss Ratio	1.06							
19	Probability of 10% Loss	0.48%							
20	Profits T-statistics	56.72							
21									
22	Probability of Loss Calculation	ARMA Strategy							
23	ProbWin	0.54							
24	ProbLoss	0.46							
25	MaxRisk (User defined)	0.10							
26	AvgWin%	0.00							
27	AvgLoss%	0.00							
28	Expected mean return per trade (Z)	0.00		Z=(ProbWin*AvgWin%)+(ProbLoss*AvgLoss%)					
29	Root mean squared return per trade (A)	0.00		A=[(ProbWin*AvgWin%^2)+(ProbLoss*AvgLoss%^2)]*1/2					
30	Trader's advantage P=0.5*(1+Z/A)	0.55							
31	Probability of Loss R=(1-P)/P*(MaxRisk/A)	0.48%							

Figure 1.10 Restricted ARMA(10,10) model Excel spreadsheet (in-sample)

$$Y_t^* = 0.2492 - 0.3613X_{1,t} - 0.2872X_{2,t} + 0.2862X_{3,t} + 0.2525X_{4,t} - 0.3692X_{5,t} - 0.3937X_{6,t} + \varepsilon_t$$

where  $X_{1,t}, \dots, X_{6,t}$  are the JP\_yc(-2), UK\_yc(-9), JAPDOWA(-1), ITMIB30(-19), JAPAYE\$(-10), and OILBREN(-1) binary explanatory variables, respectively.<sup>9</sup>

All of the coefficients in the model are significant at the 98% confidence interval. The overall significance of the model is tested using the LR test. The null hypothesis that all coefficients except the constant are not significantly different from zero is rejected at the 99% confidence interval. The results of this are presented in Table 1.10.

To justify the use of Japanese variables, which seems difficult from an economic perspective, the joint overall significance of this subset of variables is tested using the LR test for redundant variables. The null hypothesis that these coefficients, except the constant, are not jointly significantly different from zero is rejected at the 99% confidence interval. The results of this are presented in Table 1.11. In addition, a model that did not include the Japanese variables, but was otherwise identical to logit1, was produced and the trading performance evaluated, which we shall name nojap (equation nojap of the logit.wf1 EViews workfile). The Sharpe ratio, average gain/loss ratio and correct directional change of the nojap model were 1.34, 1.01 and 54.38%, respectively. The corresponding values for the logit1 model were 2.26, 1.01 and 58.13%. The evaluation can be reviewed in Sheet 2 of the is\_logit1.xls and is\_nojap.xls Excel spreadsheets, and is also presented in Figures 1.12 and 1.13, respectively.

<sup>9</sup> Datastream mnemonics as mentioned in Table 1.1, yield curves and lags in brackets are used to save space.

	A	B	C	D	E	F	G	H	I
1		ARMA Strategy							
2	Annualised Return	9.47%		Mean Absolute Error	0.0039				
3	Cumulative Return	54.42%		Mean Absolute Percentage Error	105.57%				
4	Annualised Volatility	8.50%		Root Mean Squared Error	0.0053				
5	Sharpe Ratio	1.11		Theil's Inequality Coefficient	0.9040				
6	Maximum Daily Profit	2.49%							
7	Maximum Daily Loss	-2.51%		# Periods Daily returns Rise	663				
8	Maximum Drawdown	-8.57%		# Periods Daily returns Fall	785				
9	% Winning trades	52.60%		# Winning Up Periods	196				
10	% Losing trades	47.40%		# Winning Down Periods	562				
11	Number of Up Periods	758		% Winning Up Periods	29.56%				
12	Number of Down Periods	662		% Winning Down Periods	71.59%				
13	Number of Transactions	346							
14	Total Trading Days	1448							
15	Correct Directional Change	52.35%							
16	Avg Gain in Up Periods	0.41%							
17	Avg Loss in Down Periods	-0.39%							
18	Avg Gain/Loss Ratio	1.06							
19	Probability of 10% Loss	2.35%							
20	Profits T-statistics	42.40							
21									
22	Probability of Loss Calculation	ARMA Strategy							
23									
24	ProbWin	0.52							
25	ProbLoss	0.48							
26	MaxRisk (User defined)	0.10							
27	AvgWin%	0.00							
28	AvgLoss%	0.00							
29	Expected mean return per trade (Z)	0.00		Z=(ProbWin*AvgWin%)+(ProbLoss*AvgLoss%)					
30	Root mean squared return per trade (A)	0.00		A=[(ProbWin*AvgWin%^2)+(ProbLoss*AvgLoss%^2)]^1/2					
31	Trader's advantage P=0.5*(1+(Z/A))	0.54							
32	Probability of Loss R=(1-P)/P*(MaxRisk/A)	2.35%							

Figure 1.11 Alternative restricted ARMA(10,10) model Excel spreadsheet (in-sample)

The logit1 model was retained for out-of-sample estimation. As, in practice, the estimation of the model is based upon the cumulative distribution of the logistic function for the error term, the forecasts produced range between zero and one, requiring transformation into a binary series. Again, the procedure is quite simple: a binary variable equal to one is produced if the forecast is greater than 0.5 and zero otherwise.

The performance of the strategy is evaluated in terms of forecast accuracy via the correct directional change measure and in terms of trading performance. Several other adequate models were produced and their performance evaluated. None performed better in-sample, therefore the logit1 model was retained.

### 1.5 NEURAL NETWORK MODELS: THEORY AND METHODOLOGY

Neural networks are “data-driven self-adaptive methods in that there are few *a priori* assumptions about the models under study” (Zhang *et al.*, 1998: 35). As a result, they are well suited to problems where economic theory is of little use. In addition, neural networks are universal approximators capable of approximating any continuous function (Hornik *et al.*, 1989).

Many researchers are confronted with problems where important nonlinearities exist between the independent variables and the dependent variable. Often, in such circumstances, traditional forecasting methods lack explanatory power. Recently, nonlinear models have attempted to cover this shortfall. In particular, NNR models have been applied with increasing success to financial markets, which often contain nonlinearities (Dunis and Jalilov, 2002).

**Table 1.10** Logit1 EUR/USD returns estimation

Dependent Variable: BDR\_USEURSP

Method: ML – Binary Logit

Sample(adjusted): 20 1459

Included observations: 1440 after adjusting endpoints

Convergence achieved after 3 iterations

Covariance matrix computed using second derivatives

Variable	Coefficient	Std. error	z-Statistic	Prob.
C	0.249231	0.140579	1.772894	0.0762
BDR_JP_YC(-2)	-0.361289	0.108911	-3.317273	<b>0.0009</b>
BDR_UK_YC(-9)	-0.287220	0.108397	-2.649696	<b>0.0081</b>
BDR_JAPDOWA(-1)	0.286214	0.108687	2.633369	<b>0.0085</b>
BDR_ITMIB31(-19)	0.252454	0.108056	2.336325	<b>0.0195</b>
BDR_JAPAYE\$(-10)	-0.369227	0.108341	-3.408025	<b>0.0007</b>
BDR_OILBREN(-1)	-0.393689	0.108476	-3.629261	<b>0.0003</b>
Mean dependent var.	0.457639	S.D. dependent var.		0.498375
S.E. of regression	0.490514	Akaike info. criterion		1.353305
Sum squared resid.	344.7857	Schwarz criterion		1.378935
Log likelihood	-967.3795	Hannan-Quinn criterion		1.362872
Restr. log likelihood	-992.9577	Avg. log likelihood		-0.671791
LR statistic (6 df)	51.15635	McFadden R-squared		0.025760
Prob(LR statistic)	<b>2.76E-09</b>			
Obs. with dep = 0	781	Total obs.		1440
Obs. with dep = 1	659			

Theoretically, the advantage of NNR models over traditional forecasting methods is because, as is often the case, the model best adapted to a particular problem cannot be identified. It is then better to resort to a method that is a generalisation of many models, than to rely on an *a priori* model (Dunis and Huang, 2002).

However, NNR models have been criticised and their widespread success has been hindered because of their “black-box” nature, excessive training times, danger of overfitting, and the large number of “parameters” required for training. As a result, deciding on the appropriate network involves much trial and error.

For a full discussion on neural networks, please refer to Haykin (1999), Kaastra and Boyd (1996), Kingdon (1997), or Zhang *et al.* (1998). Notwithstanding, we provide below a brief description of NNR models and procedures.

### 1.5.1 Neural network models

The will to understand the functioning of the brain is the basis for the study of neural networks. Mathematical modelling started in the 1940s with the work of McCulloch and Pitts, whose research was based on the study of networks composed of a number of simple interconnected processing elements called neurons or nodes. If the description is correct,

**Table 1.11** Logit1 estimation redundant variables LR test

---

Redundant Variables: BDR\_JP\_YC(-2), BDR\_JAPDOWA(-1), BDR\_JAPAYE\$(-10)

---

<i>F</i> -statistic	9.722023	Probability	0.000002
Log likelihood ratio	28.52168	Probability	<b>0.000003</b>

---

Test Equation:  
 Dependent Variable: BDR\_USEURSP  
 Method: ML – Binary Logit  
 Sample: 20 1459  
 Included observations: 1440  
 Convergence achieved after 3 iterations  
 Covariance matrix computed using second derivatives

---

Variable	Coefficient	Std. error	z-Statistic	Prob.
C	-0.013577	0.105280	-0.128959	0.8974
BDR_UK_YC(-9)	-0.247254	0.106979	-2.311245	0.0208
BDR_ITMIB31(-19)	0.254096	0.106725	2.380861	0.0173
BDR_OILBREN(-1)	-0.345654	0.106781	-3.237047	0.0012
Mean dependent var.	0.457639	S.D. dependent var.		0.498375
S.E. of regression	0.494963	Akaike info. criterion		1.368945
Sum squared resid.	351.8032	Schwarz criterion		1.383590
Log likelihood	-981.6403	Hannan–Quinn criterion		1.374412
Restr. log likelihood	-992.9577	Avg. log likelihood		-0.681695
LR statistic (3 df)	22.63467	McFadden <i>R</i> -squared		0.011398
Prob(LR statistic)	4.81E-05			
Obs. with dep = 0	781	Total obs.		1440
Obs. with dep = 1	659			

---

they can be turned into models mimicking some of the brain’s functions, possibly with the ability to learn from examples and then to generalise on unseen examples.

A neural network is typically organised into several layers of elementary processing units or nodes. The first layer is the input layer, the number of nodes corresponding to the number of variables, and the last layer is the output layer, the number of nodes corresponding to the forecasting horizon for a forecasting problem.<sup>10</sup> The input and output layer can be separated by one or more hidden layers, with each layer containing one or more hidden nodes.<sup>11</sup> The nodes in adjacent layers are fully connected. Each neuron receives information from the preceding layer and transmits to the following layer only.<sup>12</sup> The neuron performs a weighted summation of its inputs; if the sum passes a threshold the neuron transmits, otherwise it remains inactive. In addition, a bias neuron may be connected to each neuron in the hidden and output layers. The bias has a value of positive

---

<sup>10</sup> Linear regression models may be viewed analogously to neural networks with no hidden layers (Kaastra and Boyd, 1996).

<sup>11</sup> Networks with hidden layers are multilayer networks; a multilayer perceptron network is used for this chapter.

<sup>12</sup> If the flow of information through the network is from the input to the output, it is known as “feedforward”.

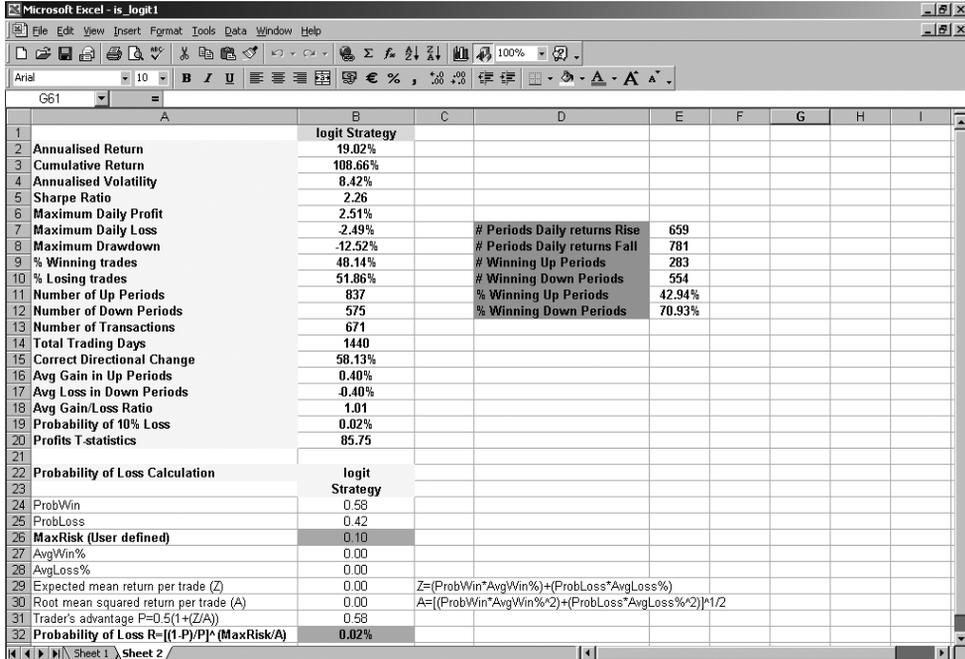


Figure 1.12 Logit1 estimation Excel spreadsheet (in-sample)

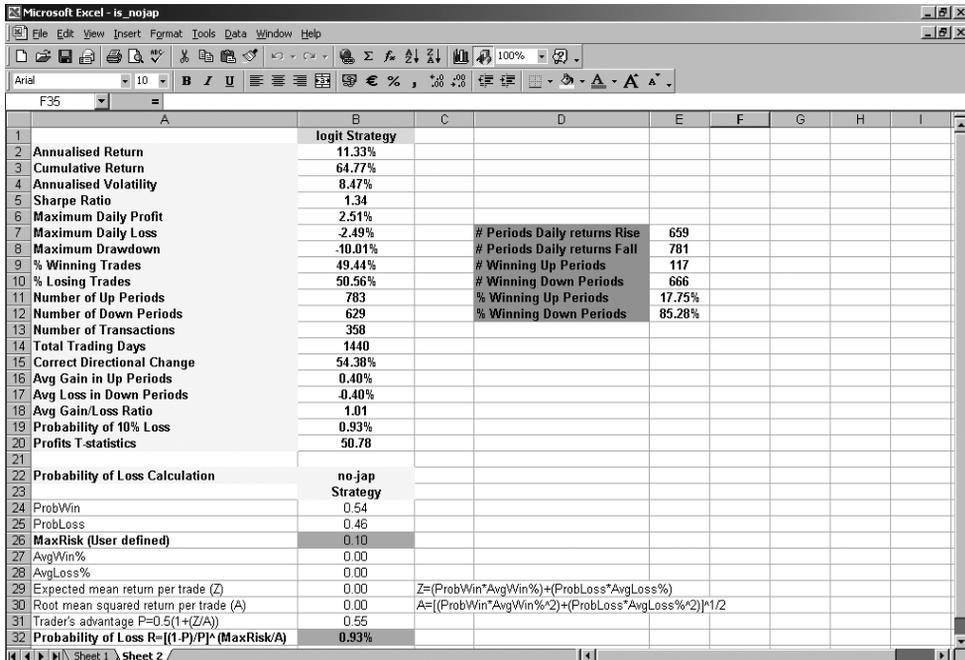
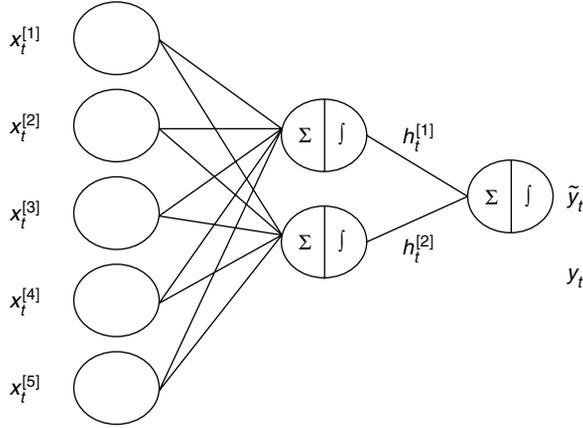


Figure 1.13 Nojap estimation Excel spreadsheet (in-sample)



where  $x_t^{[i]}$  ( $i = 1, 2, \dots, 5$ ) are the NNR model inputs at time  $t$   
 $h_t^{[j]}$  ( $j = 1, 2$ ) are the hidden nodes outputs  
 $y_t$  and  $\tilde{y}_t$  are the actual value and NNR model output, respectively

**Figure 1.14** A single output fully connected NNR model

one and is analogous to the intercept in traditional regression models. An example of a fully connected NNR model with one hidden layer and two nodes is presented in Figure 1.14.

The vector  $A = (x^{[1]}, x^{[2]}, \dots, x^{[n]})$  represents the input to the NNR model where  $x_t^{[i]}$  is the level of activity of the  $i$ th input. Associated with the input vector is a series of weight vectors  $W_j = (w_{1j}, w_{2j}, \dots, w_{nj})$  so that  $w_{ij}$  represents the strength of the connection between the input  $x_t^{[i]}$  and the processing unit  $b_j$ . There may also be the input bias  $\varphi_j$  modulated by the weight  $w_{0j}$  associated with the inputs. The total input of the node  $b_j$  is the dot product between vectors  $A$  and  $W_j$ , less the weighted bias. It is then passed through a nonlinear activation function to produce the output value of processing unit  $b_j$ :

$$b_j = f \left( \sum_{i=1}^n x^{[i]} w_{ij} - w_{0j} \varphi_j \right) = f(X_j) \tag{1.8}$$

Typically, the activation function takes the form of the logistic function, which introduces a degree of nonlinearity to the model and prevents outputs from reaching very large values that can “paralyse” NNR models and inhibit training (Kaastra and Boyd, 1996; Zhang *et al.*, 1998). Here we use the logistic function:

$$f(X_j) = \frac{1}{1 + e^{-X_j}} \tag{1.9}$$

The modelling process begins by assigning random values to the weights. The output value of the processing unit is passed on to the output layer. If the output is optimal, the process is halted, if not, the weights are adjusted and the process continues until an optimal solution is found. The output error, namely the difference between the actual value and the NNR model output, is the optimisation criterion. Commonly, the criterion

is the root-mean-squared error (RMSE). The RMSE is systematically minimised through the adjustment of the weights. Basically, training is the process of determining the optimal solutions network weights, as they represent the knowledge learned by the network. Since inadequacies in the output are fed back through the network to adjust the network weights, the NNR model is trained by backpropagation<sup>13</sup> (Shapiro, 2000).

A common practice is to divide the time series into three sets called the training, test and validation (out-of-sample) sets, and to partition them as roughly  $\frac{2}{3}$ ,  $\frac{1}{6}$  and  $\frac{1}{6}$ , respectively. The testing set is used to evaluate the generalisation ability of the network. The technique consists of tracking the error on the training and test sets. Typically, the error on the training set continually decreases, however the test set error starts by decreasing and then begins to increase. From this point the network has stopped learning the similarities between the training and test sets, and has started to learn meaningless differences, namely the noise within the training data. For good generalisation ability, training should stop when the test set error reaches its lowest point. The stopping rule reduces the likelihood of overfitting, i.e. that the network will become overtrained (Dunis and Huang, 2002; Mehta, 1995).

An evaluation of the performance of the trained network is made on new examples not used in network selection, namely the validation set. Crucially, the validation set should never be used to discriminate between networks, as any set that is used to choose the best network is, by definition, a test set. In addition, good generalisation ability requires that the training and test sets are representative of the population, inappropriate selection will affect the network generalisation ability and forecast performance (Kaastra and Boyd, 1996; Zhang *et al.*, 1998).

### 1.5.2 Issues in neural network modelling

Despite the satisfactory features of NNR models, the process of building them should not be taken lightly. There are many issues that can affect the network's performance and should be considered carefully.

The issue of finding the most parsimonious model is always a problem for statistical methods and particularly important for NNR models because of the problem of overfitting. Parsimonious models not only have the recognition ability but also the more important generalisation ability. Overfitting and generalisation are always going to be a problem for real-world situations, and this is particularly true for financial applications where time series may well be quasi-random, or at least contain noise.

One of the most commonly used heuristics to ensure good generalisation is the application of some form of Occam's Razor. The principle states, "unnecessary complex models *should not* be preferred to simpler ones. However . . . more complex models always fit the data better" (Kingdon, 1997: 49). The two objectives are, of course, contradictory. The solution is to find a model with the smallest possible complexity, and yet which can still describe the data set (Haykin, 1999; Kingdon, 1997).

A reasonable strategy in designing NNR models is to start with one layer containing a few hidden nodes, and increase the complexity while monitoring the generalisation ability. The issue of determining the optimal number of layers and hidden nodes is a crucial factor

---

<sup>13</sup> Backpropagation networks are the most common multilayer network and are the most used type in financial time series forecasting (Kaastra and Boyd, 1996). We use them exclusively here.

for good network design, as the hidden nodes provide the ability to generalise. However, in most situations there is no way to determine the best number of hidden nodes without training several networks. Several rules of thumb have been proposed to aid the process, however none work well for all applications. Notwithstanding, simplicity must be the aim (Mehta, 1995).

Since NNR models are pattern matchers, the representation of data is critical for a successful network design. The raw data for the input and output variables are rarely fed into the network, they are generally scaled between the upper and lower bounds of the activation function. For the logistic function the range is  $[0,1]$ , avoiding the function's saturation zones. Practically, as here, a normalisation  $[0.2,0.8]$  is often used with the logistic function, as its limits are only reached for infinite input values (Zhang *et al.*, 1998).

Crucial for backpropagation learning is the learning rate of the network as it determines the size of the weight changes. Smaller learning rates slow the learning process, while larger rates cause the error function to change wildly without continuously improving. To improve the process a momentum parameter is used which allows for larger learning rates. The parameter determines how past weight changes affect current weight changes, by making the next weight change in approximately the same direction as the previous one<sup>14</sup> (Kaastra and Boyd, 1996; Zhang *et al.*, 1998).

### 1.5.3 Neural network modelling procedure

Conforming to standard heuristics, the training, test and validation sets were partitioned as approximately  $\frac{2}{3}$ ,  $\frac{1}{6}$  and  $\frac{1}{6}$ , respectively. The training set runs from 17 October 1994 to 8 April 1999 (1169 observations), the test set runs from 9 April 1999 to 18 May 2000 (290 observations), and the validation set runs from 19 May 2000 to 3 July 2001 (290 observations), reserved for out-of-sample forecasting and evaluation, identical to the out-of-sample period for the benchmark models.

To start, traditional linear cross-correlation analysis helped establish the existence of a relationship between EUR/USD returns and potential explanatory variables. Although NNR models attempt to map nonlinearities, linear cross-correlation analysis can give some indication of which variables to include in a model, or at least a starting point to the analysis (Diekmann and Gutjahr, 1998; Dunis and Huang, 2002).

The analysis was performed for all potential explanatory variables. Lagged terms that were most significant as determined via the cross-correlation analysis are presented in Table 1.12.

The lagged terms  $SPCOMP(-1)$  and  $US\_yc(-1)$  could not be used because of time-zone differences between London and the USA, as discussed at the beginning of Section 1.3. As an initial substitute  $SPCOMP(-2)$  and  $US\_yc(-2)$  were used. In addition, various lagged terms of the EUR/USD returns were included as explanatory variables.

Variable selection was achieved via a forward stepwise NNR procedure, namely potential explanatory variables were progressively added to the network. If adding a new variable improved the level of explained variance (EV) over the previous "best" network, the pool of explanatory variables was updated.<sup>15</sup> Since the aim of the model-building

<sup>14</sup> The problem of convergence did not occur within this research; as a result, a learning rate of 0.1 and momentum of zero were used exclusively.

<sup>15</sup> EV is an approximation of the coefficient of determination,  $R^2$ , in traditional regression techniques.

**Table 1.12** Most significant lag of each potential explanatory variable (in returns)

Variable	Best lag
<i>DAXINDEX</i>	10
<i>DJES50I</i>	10
<i>FRCAC40</i>	10
<i>FTSE100</i>	5
<i>GOLDBLN</i>	19
<i>ITMIB</i>	9
<i>JAPAYE\$</i>	10
<i>OILBREN</i>	1
<i>JAPDOWA</i>	15
<i>SPCOMP</i>	1
<i>USDOLLR</i>	12
<i>BD_yc</i>	19
<i>EC_yc</i>	2
<i>FR_yc</i>	9
<i>IT_yc</i>	2
<i>JP_yc</i>	6
<i>UK_yc</i>	19
<i>US_yc</i>	1
<i>NYFECRB</i>	20

procedure is to build a model with good generalisation ability, a model that has a higher EV level has a better ability. In addition, a good measure of this ability is to compare the EV level of the test and validation sets: if the test set and validation set levels are similar, the model has been built to generalise well.

The decision to use explained variance is because the EUR/USD returns series is a stationary series and stationarity remains important if NNR models are assessed on the level of explained variance (Dunis and Huang, 2002). The EV levels for the training, test and validation sets of the selected NNR model, which we shall name *nnr1* (*nnr1.prv* Previa file), are presented in Table 1.13.

An EV level equal to, or greater than, 80% was used as the NNR learning termination criterion. In addition, if the NNR model did not reach this level within 1500 learning sweeps, again the learning terminates. The criteria selected are reasonable for daily data and were used exclusively here.

If after several attempts there was failure to improve on the previous “best” model, variables in the model were alternated in an attempt to find a better combination. This

**Table 1.13** *nnr1* model EV for the training, test and validation sets

Training set	Test set	Validation set
3.4%	2.3%	2.2%

procedure recognises the likelihood that some variables may only be relevant predictors when in combination with certain other variables.

Once a tentative model is selected, post-training weights analysis helps establish the importance of the explanatory variables, as there are no standard statistical tests for NNR models. The idea is to find a measure of the contribution a given weight has to the overall output of the network, in essence allowing detection of insignificant variables. Such analysis includes an examination of a Hinton graph, which represents graphically the weight matrix within the network. The principle is to include in the network variables that are strongly significant. In addition, a small bias weight is preferred (Diekmann and Gutjahr, 1998; Kingdon, 1997; Previa, 2001). The input to a hidden layer Hinton graph of the nnr1 model produced by Previa is presented in Figure 1.15. The graph suggests that the explanatory variables of the selected model are strongly significant, both positive (green) and negative (black), and that there is a small bias weight. In addition, the input to hidden layer weight matrix of the nnr1 model produced by Previa is presented in Table 1.14.

The nnr1 model contained the returns of the explanatory variables presented in Table 1.15, having one hidden layer containing five hidden nodes.

Again, to justify the use of the Japanese variables a further model that did not include these variables, but was otherwise identical to nnr1, was produced and the performance evaluated, which we shall name nojap (nojap.prv Previa file). The EV levels of the training

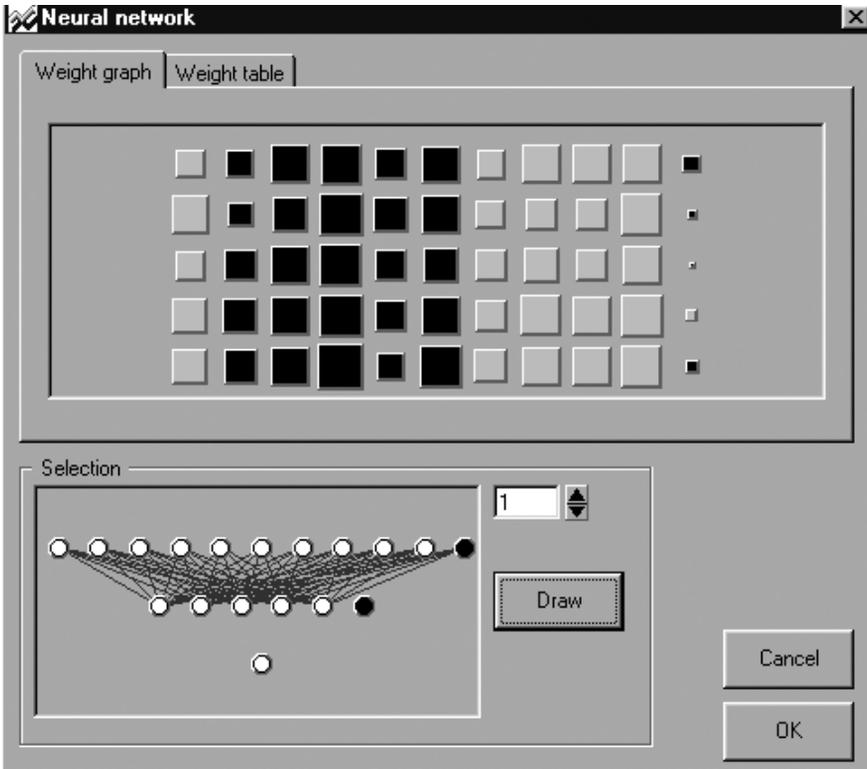


Figure 1.15 Hinton graph of the nnr1 EUR/USD returns model

**Table 1.14** Input to hidden layer weight matrix of the nnr1 EUR/USD returns model

	GOLD BLN (-19)	JAPAY E\$ (-10)	JAP DOWA (-15)	OIL BREN (-1)	US DOLLR (-12)	FR_yc (-2)	IT_yc (-6)	JP_yc (-9)	JAPAY E\$ (-1)	JAP DOWA (-1)	Bias
<b>C[1,0]</b>	0.2316	-0.2120	-0.4336	-0.4579	-0.2621	-0.3911	0.2408	0.4295	0.4067	0.4403	-0.0824
<b>C[1,1]</b>	0.4016	-0.1752	-0.3589	-0.5474	-0.3663	-0.4623	0.2438	0.2786	0.2757	0.4831	-0.0225
<b>C[1,2]</b>	0.2490	-0.3037	-0.4462	-0.5139	-0.2506	-0.3491	0.2900	0.3634	0.2737	0.4132	-0.0088
<b>C[1,3]</b>	0.3382	-0.3588	-0.4089	-0.5446	-0.2730	-0.4531	0.2555	0.4661	0.4153	0.5245	0.0373
<b>C[1,4]</b>	0.3338	-0.3283	-0.4086	-0.6108	-0.2362	-0.4828	0.3088	0.4192	0.4254	0.4779	-0.0447

**Table 1.15** nnr1 model explanatory variables (in returns)

Variable	Lag
<i>GOLDBLN</i>	19
<i>JAPAYE\$</i>	10
<i>JAPDOWA</i>	15
<i>OILBREN</i>	1
<i>USDOLLR</i>	12
<i>FR_yc</i>	2
<i>IT_yc</i>	6
<i>JP_yc</i>	9
<i>JAPAYE\$</i>	1
<i>JAPDOWA</i>	1

and test sets of the nojap model were 1.4 and 0.6 respectively, which are much lower than the nnr1 model.

The nnr1 model was retained for out-of-sample estimation. The performance of the strategy is evaluated in terms of traditional forecasting accuracy and in terms of trading performance.

Several other adequate models were produced and their performance evaluated, including RNN models.<sup>16</sup> In essence, the only difference from NNR models is the addition of a loop back from a hidden or the output layer to the input layer. The loop back is then used as an input in the next period. There is no theoretical or empirical answer to whether the hidden layer or the output should be looped back. However, the looping back of either allows RNN models to keep the memory of the past,<sup>17</sup> a useful property in forecasting applications. This feature comes at a cost, as RNN models require more connections, raising the issue of complexity. Since simplicity is the aim, a less complex model that can still describe the data set is preferred.

The statistical forecasting accuracy results of the nnr1 model and the RNN model, which we shall name rnn1 (rnn1.prv Previa file), were only marginally different, namely the mean absolute percentage error (MAPE) differs by 0.09%. However, in terms of

<sup>16</sup> For a discussion on recurrent neural network models refer to Dunis and Huang (2002).

<sup>17</sup> The looping back of the output layer is an error feedback mechanism, implying the use of a nonlinear error-correction model (Dunis and Huang, 2002).

	A	B	C	D	E	F	G	H	I
1		nnr Strategy							
2	Annualised Return	18.18%		Mean Absolute Error	0.0038				
3	Cumulative Return	103.90%		Mean Absolute Percentage Error	137.82%				
4	Annualised Volatility	8.42%		Root Mean Squared Error	0.0053				
5	Sharpe Ratio	2.16		Theil's Inequality Coefficient	0.8405				
6	Maximum Daily Profit	2.51%							
7	Maximum Daily Loss	-2.49%		# Periods Daily returns Rise	675				
8	Maximum Drawdown	-9.11%		# Periods Daily returns Fall	765				
9	% Winning trades	48.12%		# Winning Up Periods	245				
10	% Losing trades	51.88%		# Winning Down Periods	569				
11	Number of Up Periods	814		% Winning Up Periods	36.30%				
12	Number of Down Periods	626		% Winning Down Periods	74.38%				
13	Number of Transactions	611							
14	Total Trading Days	1440							
15	Correct Directional Change	56.53%							
16	Avg Gain in Up Periods	0.41%							
17	Avg Loss in Down Periods	-0.37%							
18	Avg Gain/Loss Ratio	1.12							
19	Probability of 10% Loss	0.01%							
20	Profits T-statistics	81.91							
21									
22	Probability of Loss Calculation	nnr1							
23		Strategy							
24	ProbWin	0.57							
25	ProbLoss	0.43							
26	MaxRisk (User defined)	0.10							
27	AvgWin%	0.00							
28	AvgLoss%	0.00							
29	Expected mean return per trade (Z)	0.00	$Z=(ProbWin*AvgWin%)+(ProbLoss*AvgLoss%)$						
30	Root mean squared return per trade (A)	0.00	$A=[(ProbWin*AvgWin%^2)+(ProbLoss*AvgLoss%^2)]^{1/2}$						
31	Trader's advantage $P=0.5(1+(Z/A))$	0.59							
32	Probability of Loss $R=[(1-P)/P]^A*(MaxRisk/A)$	0.01%							

Figure 1.16 nnr1 model Excel spreadsheet (in-sample)

	A	B	C	D	E	F	G	H	I
1		rnn Strategy							
2	Annualised Return	18.38%		Mean Absolute Error	0.0038				
3	Cumulative Return	105.04%		Mean Absolute Percentage Error	137.91%				
4	Annualised Volatility	8.42%		Root Mean Squared Error	0.0053				
5	Sharpe Ratio	2.18		Theil's Inequality Coefficient	0.8405				
6	Maximum Daily Profit	2.51%							
7	Maximum Daily Loss	-2.49%		# Periods Daily returns Rise	675				
8	Maximum Drawdown	-9.11%		# Periods Daily returns Fall	765				
9	% Winning trades	48.04%		# Winning Up Periods	247				
10	% Losing trades	51.96%		# Winning Down Periods	569				
11	Number of Up Periods	816		% Winning Up Periods	36.59%				
12	Number of Down Periods	624		% Winning Down Periods	74.38%				
13	Number of Transactions	612							
14	Total Trading Days	1440							
15	Correct Directional Change	56.67%							
16	Avg Gain in Up Periods	0.41%							
17	Avg Loss in Down Periods	-0.37%							
18	Avg Gain/Loss Ratio	1.11							
19	Probability of 10% Loss	0.01%							
20	Profits T-statistics	82.83							
21									
22	Probability of Loss Calculation	rnn1							
23		Strategy							
24	ProbWin	0.57							
25	ProbLoss	0.43							
26	MaxRisk (User defined)	0.10							
27	AvgWin%	0.00							
28	AvgLoss%	0.00							
29	Expected mean return per trade (Z)	0.00	$Z=(ProbWin*AvgWin%)+(ProbLoss*AvgLoss%)$						
30	Root mean squared return per trade (A)	0.00	$A=[(ProbWin*AvgWin%^2)+(ProbLoss*AvgLoss%^2)]^{1/2}$						
31	Trader's advantage $P=0.5(1+(Z/A))$	0.59							
32	Probability of Loss $R=[(1-P)/P]^A*(MaxRisk/A)$	0.01%							

Figure 1.17 rnn1 model Excel spreadsheet (in-sample)

trading performance there is little to separate the `nnr1` and `rnn1` models. The evaluation can be reviewed in Sheet 2 of the `is_nnr1.xls` and `is_rnn1.xls` Excel spreadsheets, and is also presented in Figures 1.16 and 1.17, respectively.

The decision to retain the `nnr1` model over the `rnn1` model is because the `rnn1` model is more complex and yet does not possess any decisive added value over the simpler model.

## 1.6 FORECASTING ACCURACY AND TRADING SIMULATION

To compare the performance of the strategies, it is necessary to evaluate them on previously unseen data. This situation is likely to be the closest to a true forecasting or trading situation. To achieve this, all models retained an identical out-of-sample period allowing a direct comparison of their forecasting accuracy and trading performance.

### 1.6.1 Out-of-sample forecasting accuracy measures

Several criteria are used to make comparisons between the forecasting ability of the benchmark and NNR models, including mean absolute error (MAE), RMSE,<sup>18</sup> MAPE, and Theil's inequality coefficient (Theil-U).<sup>19</sup> For a full discussion on these measures, refer to Hanke and Reitsch (1998) and Pindyck and Rubinfeld (1998). We also include correct directional change (CDC), which measures the capacity of a model to correctly predict the subsequent actual change of a forecast variable, an important issue in a trading strategy that relies on the direction of a forecast rather than its level. The statistical performance measures used to analyse the forecasting techniques are presented in Table 1.16.

### 1.6.2 Out-of-sample trading performance measures

Statistical performance measures are often inappropriate for financial applications. Typically, modelling techniques are optimised using a mathematical criterion, but ultimately the results are analysed on a financial criterion upon which it is not optimised. In other words, the forecast error may have been minimised during model estimation, but the evaluation of the true merit should be based on the performance of a trading strategy. Without actual trading, the best means of evaluating performance is via a simulated trading strategy. The procedure to create the buy and sell signals is quite simple: a EUR/USD buy signal is produced if the forecast is positive, and a sell otherwise.<sup>20</sup>

For many traders and analysts market direction is more important than the value of the forecast itself, as in financial markets money can be made simply by knowing the direction the series will move. In essence, "low forecast errors and trading profits are not synonymous since a single large trade forecasted incorrectly . . . could have accounted for most of the trading system's profits" (Kaastra and Boyd, 1996: 229).

The trading performance measures used to analyse the forecasting techniques are presented in Tables 1.17 and 1.18. Most measures are self-explanatory and are commonly used in the fund management industry. Some of the more important measures include the Sharpe ratio, maximum drawdown and average gain/loss ratio. The Sharpe ratio is a

---

<sup>18</sup> The MAE and RMSE statistics are scale-dependent measures but allow a comparison between the actual and forecast values, the lower the values the better the forecasting accuracy.

<sup>19</sup> When it is more important to evaluate the forecast errors independently of the scale of the variables, the MAPE and Theil-U are used. They are constructed to lie within [0,1], zero indicating a perfect fit.

<sup>20</sup> A buy signal is to buy euros at the current price or continue holding euros, while a sell signal is to sell euros at the current price or continue holding US dollars.

**Table 1.16** Statistical performance measures

Performance measure	Description
Mean absolute error	$\text{MAE} = \frac{1}{T} \sum_{t=1}^T  \tilde{y}_t - y_t  \quad (1.10)$
Mean absolute percentage error	$\text{MAPE} = \frac{100}{T} \sum_{t=1}^T \left  \frac{\tilde{y}_t - y_t}{y_t} \right  \quad (1.11)$
Root-mean-squared error	$\text{RMSE} = \sqrt{\frac{1}{T} \sum_{t=1}^T (\tilde{y}_t - y_t)^2} \quad (1.12)$
Theil's inequality coefficient	$U = \frac{\sqrt{\frac{1}{T} \sum_{t=1}^T (\tilde{y}_t - y_t)^2}}{\sqrt{\frac{1}{T} \sum_{t=1}^T (\tilde{y}_t)^2} + \sqrt{\frac{1}{T} \sum_{t=1}^T (y_t)^2}} \quad (1.13)$
Correct directional change	$\text{CDC} = \frac{100}{N} \sum_{t=1}^N D_t \quad (1.14)$ where $D_t = 1$ if $y_t \cdot \tilde{y}_t > 0$ else $D_t = 0$

$y_t$  is the actual change at time  $t$ .

$\tilde{y}_t$  is the forecast change.

$t = 1$  to  $t = T$  for the forecast period.

risk-adjusted measure of return, with higher ratios preferred to those that are lower, the maximum drawdown is a measure of downside risk and the average gain/loss ratio is a measure of overall gain, a value above one being preferred (Dunis and Jalilov, 2002; Fernandez-Rodriguez *et al.*, 2000).

The application of these measures may be a better standard for determining the quality of the forecasts. After all, the financial gain from a given strategy depends on trading performance, not on forecast accuracy.

### 1.6.3 Out-of-sample forecasting accuracy results

The forecasting accuracy statistics do not provide very conclusive results. Each of the models evaluated, except the logit model, are nominated “best” at least once. Interestingly, the naïve model has the lowest Theil-U statistic at 0.6901; if this model is believed to be the “best” model there is likely to be no added value using more complicated forecasting techniques. The ARMA model has the lowest MAPE statistic at 101.51%, and equals the MAE of the NNR model at 0.0056. The NNR model has the lowest RMSE statistic, however the value is only marginally less than the ARMA model. The MACD model has the highest CDC measure, predicting daily changes accurately 60.00% of the time. It is difficult to select a “best” performer from these results, however a majority decision rule

**Table 1.17** Trading simulation performance measures

Performance measure	Description
Annualised return	$R^A = 252 \times \frac{1}{N} \sum_{t=1}^N R_t$ (1.15)
Cumulative return	$R^C = \sum_{t=1}^N R_t$ (1.16)
Annualised volatility	$\sigma^A = \sqrt{252} \times \sqrt{\frac{1}{N-1} \sum_{t=1}^N (R_t - \bar{R})^2}$ (1.17)
Sharpe ratio	$SR = \frac{R^A}{\sigma^A}$ (1.18)
Maximum daily profit	Maximum value of $R_t$ over the period (1.19)
Maximum daily loss	Minimum value of $R_t$ over the period (1.20)
Maximum drawdown	Maximum negative value of $\sum (R_T)$ over the period
	$MD = \min_{t=1, \dots, N} \left( R_t^c - \max_{i=1, \dots, t} (R_i^c) \right)$ (1.21)
% Winning trades	$WT = 100 \times \frac{\sum_{t=1}^N F_t}{NT}$ (1.22) where $F_t = 1$ if transaction profit $_t > 0$
% Losing trades	$LT = 100 \times \frac{\sum_{t=1}^N G_t}{NT}$ (1.23) where $G_t = 1$ if transaction profit $_t < 0$
Number of up periods	Nup = number of $R_t > 0$ (1.24)
Number of down periods	Ndown = number of $R_t < 0$ (1.25)
Number of transactions	$NT = \sum_{t=1}^N L_t$ (1.26) where $L_t = 1$ if trading signal $_t \neq$ trading signal $_{t-1}$
Total trading days	Number of all $R_t$ 's (1.27)
Avg. gain in up periods	$AG = (\text{Sum of all } R_t > 0) / \text{Nup}$ (1.28)
Avg. loss in down periods	$AL = (\text{Sum of all } R_t < 0) / \text{Ndown}$ (1.29)
Avg. gain/loss ratio	$GL = AG/AL$ (1.30)
Probability of 10% loss	$PoL = \left[ \frac{(1-P)}{P} \right]^{\left( \frac{\text{MaxRisk}}{\Lambda} \right)}$ (1.31) where $P = 0.5 \times \left( 1 + \left( \frac{((WT \times AG) + (LT \times AL))}{\sqrt{[(WT \times AG^2) + (LT \times AL^2)]}} \right) \right)$ and $\Lambda = \sqrt{[(WT \times AG^2) + (LT \times AL^2)]}$ MaxRisk is the risk level defined by the user; this research, 10%
Profits $T$ -statistics	$T\text{-statistics} = \sqrt{N} \times \frac{R^A}{\sigma^A}$ (1.32)

**Table 1.18** Trading simulation performance measures

Performance measure	Description
Number of periods daily returns rise	$NPR = \sum_{t=1}^N Q_t \tag{1.33}$ where $Q_t = 1$ if $y_t > 0$ else $Q_t = 0$
Number of periods daily returns fall	$NPF = \sum_{t=1}^N S_t \tag{1.34}$ where $S_t = 1$ if $y_t < 0$ else $S_t = 0$
Number of winning up periods	$NWU = \sum_{t=1}^N B_t \tag{1.35}$ where $B_t = 1$ if $R_t > 0$ and $y_t > 0$ else $B_t = 0$
Number of winning down periods	$NWD = \sum_{t=1}^N E_t \tag{1.36}$ where $E_t = 1$ if $R_t > 0$ and $y_t < 0$ else $E_t = 0$
Winning up periods (%)	$WUP = 100 \times (NWU/NPR) \tag{1.37}$
Winning down periods (%)	$WDP = 100 \times (NWD/NPF) \tag{1.38}$

**Table 1.19** Forecasting accuracy results<sup>21</sup>

	Naïve	MACD	ARMA	Logit	NNR
Mean absolute error	0.0080	–	<b>0.0056</b>	–	<b>0.0056</b>
Mean absolute percentage error	317.31%	–	<b>101.51%</b>	–	107.38%
Root-mean-squared error	0.0102	–	0.0074	–	<b>0.0073</b>
Theil’s inequality coefficient	<b>0.6901</b>	–	0.9045	–	0.8788
Correct directional change	55.86%	<b>60.00%</b>	56.55%	53.79%	57.24%

might select the NNR model as the overall “best” model because it is nominated “best” twice and also “second best” by the other three statistics. A comparison of the forecasting accuracy results is presented in Table 1.19.

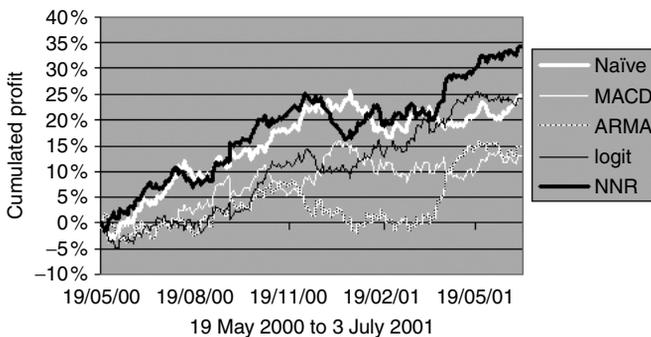
**1.6.4 Out-of-sample trading performance results**

A comparison of the trading performance results is presented in Table 1.20 and Figure 1.18. The results of the NNR model are quite impressive. It generally outperforms the benchmark strategies, both in terms of overall profitability with an annualised return of 29.68% and a cumulative return of 34.16%, and in terms of risk-adjusted performance with a Sharpe ratio of 2.57. The logit model has the lowest downside risk as measured by maximum drawdown at –5.79%, and the MACD model has the lowest downside risk

<sup>21</sup> As the MACD model is not based on forecasting the next period and binary variables are used in the logit model, statistical accuracy comparisons with these models were not always possible.

**Table 1.20** Trading performance results

	Naïve	MACD	ARMA	Logit	NNR
Annualised return	21.34%	11.34%	12.91%	21.05%	<b>29.68%</b>
Cumulative return	24.56%	13.05%	14.85%	24.22%	<b>34.16%</b>
Annualised volatility	11.64%	11.69%	11.69%	11.64%	<b>11.56%</b>
Sharpe ratio	1.83	0.97	1.10	1.81	<b>2.57</b>
Maximum daily profit	<b>3.38%</b>	1.84%	<b>3.38%</b>	1.88%	<b>3.38%</b>
Maximum daily loss	-2.10%	-3.23%	-2.10%	-3.38%	<b>-1.82%</b>
Maximum drawdown	-9.06%	-7.75%	-10.10%	<b>-5.79%</b>	<b>-9.12%</b>
% Winning trades	37.01%	24.00%	52.71%	49.65%	<b>52.94%</b>
% Losing trades	62.99%	76.00%	47.29%	50.35%	<b>47.06%</b>
Number of up periods	162	149	164	156	<b>166</b>
Number of down periods	126	138	124	132	<b>122</b>
Number of transactions	127	<b>25</b>	129	141	136
Total trading days	290	290	290	290	290
Avg. gain in up periods	0.58%	0.60%	0.55%	<b>0.61%</b>	0.60%
Avg. loss in down periods	-0.56%	-0.55%	-0.61%	<b>-0.53%</b>	-0.54%
Avg. gain/loss ratio	1.05	1.08	0.91	<b>1.14</b>	1.12
Probability of 10% loss	0.70%	<b>0.02%</b>	5.70%	0.76%	0.09%
Profits <i>T</i> -statistics	31.23	16.51	18.81	30.79	<b>43.71</b>
Number of periods daily returns rise	128	128	128	128	128
Number of periods daily returns fall	162	162	162	162	162
Number of winning up periods	<b>65</b>	45	56	49	52
Number of winning down periods	97	104	108	106	<b>114</b>
% Winning up periods	<b>50.78%</b>	35.16%	43.75%	38.28%	40.63%
% Winning down periods	59.88%	64.20%	66.67%	66.05%	<b>70.37%</b>

**Figure 1.18** Cumulated profit graph

as measured by the probability of a 10% loss at 0.02%, however this is only marginally less than the NNR model at 0.09%.

The NNR model predicted the highest number of winning down periods at 114, while the naïve model forecast the highest number of winning up periods at 65. Interestingly, all models were more successful at forecasting a fall in the EUR/USD returns series, as indicated by a greater percentage of winning down periods to winning up periods.

The logit model has the highest number of transactions at 141, while the NNR model has the second highest at 136. The MACD strategy has the lowest number of transactions at 25. In essence, the MACD strategy has longer “holding” periods compared to the other models, suggesting that the MACD strategy is not compared “like with like” to the other models.

More than with statistical performance measures, financial criteria clearly single out the NNR model as the one with the most consistent performance. Therefore it is considered the “best” model for this particular application.

### 1.6.5 Transaction costs

So far, our results have been presented without accounting for transaction costs during the trading simulation. However, it is not realistic to account for the success or otherwise of a trading system unless transaction costs are taken into account. Between market makers, a cost of 3 pips (0.0003 EUR/USD) per trade (one way) for a tradable amount, typically USD 5–10 million, would be normal. The procedure to approximate the transaction costs for the NNR model is quite simple.

A cost of 3 pips per trade and an average out-of-sample EUR/USD of 0.8971 produce an average cost of 0.033% per trade:

$$\frac{0.0003}{0.8971} = 0.033\%$$

The NNR model made 136 transactions. Since the EUR/USD time series is a series of bid rates and because, apart from the first trade, each signal implies two transactions, one to close the existing position and a second one to enter the new position indicated by the model signal, the approximate out-of-sample transaction costs for the NNR model trading strategy are about 4.55%:

$$136 \times 0.033\% = 4.55\%$$

Therefore, even accounting for transaction costs, the extra returns achieved with the NNR model still make this strategy the most attractive one despite its relatively high trading frequency.

## 1.7 CONCLUDING REMARKS

This chapter has evaluated the use of different regression models in forecasting and trading the EUR/USD exchange rate. The performance was measured statistically and financially via a trading simulation taking into account the impact of transaction costs on models with higher trading frequencies. The logic behind the trading simulation is, if profit from a trading simulation is compared solely on the basis of statistical measures, the optimum model from a financial perspective would rarely be chosen.

The NNR model was benchmarked against more traditional regression-based and other benchmark forecasting techniques to determine any added value to the forecasting process. Having constructed a synthetic EUR/USD series for the period up to 4 January 1999, the models were developed using the same in-sample data, 17 October 1994 to 18 May 2000, leaving the remaining period, 19 May 2000 to 3 July 2001, for out-of-sample forecasting.

Forecasting techniques rely on the weaknesses of the efficient market hypothesis, acknowledging the existence of market inefficiencies, with markets displaying even weak signs of predictability. However, FX markets are relatively efficient, reducing the scope of a profitable strategy. Consequently, the FX managed futures industry average Sharpe ratio is only 0.8, although a percentage of winning trades greater than 60% is often required to run a profitable FX trading desk (Grabbe, 1996 as cited in Bellgard and Goldschmidt, 1999: 10). In this respect, it is worth noting that only one of our models reached a 60% winning trades accuracy, namely the MACD model at 60.00%. Nevertheless, all of the models examined in this chapter achieved an out-of-sample Sharpe ratio higher than 0.8, the highest of which was again the NNR model at 2.57. This seems to confirm that the use of quantitative trading is more appropriate in a fund management than in a treasury type of context.

Forecasting techniques are dependent on the quality and nature of the data used. If the solution to a problem is not within the data, then no technique can extract it. In addition, sufficient information should be contained within the in-sample period to be representative of all cases within the out-of-sample period. For example, a downward trending series typically has more falls represented in the data than rises. The EUR/USD is such a series within the in-sample period. Consequently, the forecasting techniques used are estimated using more negative values than positive values. The probable implication is that the models are more likely to successfully forecast a fall in the EUR/USD, as indicated by our results, with all models forecasting a higher percentage of winning down periods than winning up periods. However, the naïve model does not learn to generalise *per se*, and as a result has the smallest difference between the number of winning up to winning down periods.

Overall our results confirm the credibility and potential of regression models and particularly NNR models as a forecasting technique. However, while NNR models offer a promising alternative to more traditional techniques, they suffer from a number of limitations. They are not the panacea. One of the major disadvantages is the inability to explain their reasoning, which has led some to consider that “neural nets are truly black boxes. Once you have trained a neural net and are generating predictions, you still do not know why the decisions are being made and can’t find out by just looking at the net. It is not unlike attempting to capture the structure of knowledge by dissecting the human brain” (Fishman *et al.*, 1991 as cited in El-Shazly and El-Shazly, 1997: 355). In essence, the neural network learning procedure is not very transparent, requiring a lot of understanding. In addition, statistical inference techniques such as significance testing cannot always be applied, resulting in a reliance on a heuristic approach. The complexity of NNR models suggests that they are capable of superior forecasts, as shown in this chapter, however this is not always the case. They are essentially nonlinear techniques and may be less capable in linear applications than traditional forecasting techniques (Balkin and Ord, 2000; Campbell *et al.*, 1997; Lisboa and Vellido, 2000; Refenes and Zaidi, 1993).

Although the results support the success of neural network models in financial applications, there is room for increased success. Such a possibility lies with optimising the neural network model on a financial criterion, and not a mathematical criterion. As the profitability of a trading strategy relies on correctly forecasting the direction of change, namely CDC, to optimise the neural network model on such a measure could improve trading performance. However, backpropagation networks optimise by minimising a differentiable function such as squared error, they cannot minimise a function based on loss,

or conversely, maximise a function based on profit. Notwithstanding, there is possibility to explore this idea further, provided the neural network software has the ability to select such an optimisation criterion.

Future work might also include the addition of hourly data as a possible explanatory variable. Alternatively, the use of first differences instead of rates of return series may be investigated, as first differences are perhaps the most effective way to generate data sets for neural network learning (Mehta, 1995).

Further investigation into RNN models is possible, or into combining forecasts. Many researchers agree that individual forecasting methods are misspecified in some manner, suggesting that combining multiple forecasts leads to increased forecast accuracy (Dunis and Huang, 2002). However, initial investigations proved unsuccessful, with the NNR model remaining the “best” model. Two simple model combinations were examined, a simple averaging of the naïve, ARMA and NNR model forecasts, and a regression-type combined forecast using the naïve, ARMA and NNR models.<sup>22</sup> The regression-type combined forecast follows the Granger and Ramanathan procedure (gr.wf1 EViews workfile). The evaluation can be reviewed in Sheet 2 of the oos\_gr.xls Excel spreadsheet, and is also presented in Figure 1.19. The lack of success using the combination models was undoubtedly because the performance of the benchmark models was so much weaker than that of the NNR model. It is unlikely that combining relatively “poor” models with an otherwise “good” one will outperform the “good” model alone.

The main conclusion that can be drawn from this chapter is that there are indeed nonlinearities present within financial markets and that a neural network model can be

	B	C	D	E	F	G	H	I
1	GR Strategy							
2	Annualised Return	26.06%	Mean Absolute Error	0.0056				
3	Cumulative Return	29.99%	Mean Absolute Percentage Error	108.65%				
4	Annualised Volatility	11.60%	Root Mean Squared Error	0.0073				
5	Sharpe Ratio	2.25	Theil's Inequality Coefficient	0.8644				
6	Maximum Daily Profit	3.38%						
7	Maximum Daily Loss	-1.76%	# Periods Daily returns Rise	128				
8	Maximum Drawdown	-7.75%	# Periods Daily returns Fall	162				
9	% Winning trades	54.23%	# Winning Up Periods	62				
10	% Losing trades	45.77%	# Winning Down Periods	97				
11	Number of Up Periods	159	% Winning Up Periods	48.44%				
12	Number of Down Periods	129	% Winning Down Periods	59.88%				
13	Number of Transactions	142						
14	Total Trading Days	290						
15	Correct Directional Change	54.83%						
16	Avg Gain in Up Periods	0.61%						
17	Avg Loss in Down Periods	-0.52%						
18	Avg Gain/Loss Ratio	1.17						
19	Probability of 10% Loss	0.22%						
20	Profits T-statistics	38.26						
21								
22	Probability of Loss Calculation	regress1						
23		Strategy						
24	ProbWin	0.55						
25	ProbLoss	0.45						
26	MaxRisk (User defined)	0.10						
27	AvgWin%	0.01						
28	AvgLoss%	-0.01						
29	Expected mean return per trade (Z)	0.00	Z=(ProbWin*AvgWin%)+(ProbLoss*AvgLoss%)					
30	Root mean squared return per trade (A)	0.01	A=[(ProbWin*AvgWin%^2)+(ProbLoss*AvgLoss%^2)]^1/2					
31	Trader's advantage P=0.5*(1+(Z/A))	0.59						
32	Probability of Loss R=((1-P)/P)^(MaxRisk/A)	0.22%						

Figure 1.19 Regression-type combined forecast Excel spreadsheet (out-of-sample)

<sup>22</sup> For a full discussion on the procedures, refer to Clemen (1989), Granger and Ramanathan (1984), and Hashem (1997).

trained to recognise them. However, despite the limitations and potential improvements mentioned above, our results strongly suggest that regression models and particularly NNR models can add value to the forecasting process. For the EUR/USD exchange rate and the period considered, NNR models clearly outperform the more traditional modelling techniques analysed in this chapter.

## REFERENCES

- Balkin, S. D. and J. K. Ord (2000), "Automatic Neural Network Modelling for Univariate Time Series", *International Journal of Forecasting*, **16**, 509–515.
- Bellgard, C. and P. Goldschmidt (1999), "Forecasting Across Frequencies: Linearity and Non-Linearity", *University of Western Australia Research Paper, Proceedings of the International Conference on Advanced Technology*, Australia, (www.imm.ecel.uwa.edu.au/~cbellgar/).
- Box, G. E. P., G. M. Jenkins and G. C. Reinsel (1994), *Time Series Analysis: Forecasting and Control*, Prentice Hall, Englewood Cliffs, NJ.
- Campbell, I. Y., A. W. Lo and A. C. MacKinley (1997), "Nonlinearities in Financial Data", in *The Econometrics of Financial Markets*, Princeton University Press, Princeton, NJ, pp. 512–524.
- Carney, J. C. and P. Cunningham (1996), "Neural Networks and Currency Exchange Rate Prediction", *Trinity College Working Paper, Foresight Business Journal web page*, (www.maths.tcd.ie/pub/fbj/forex4.html).
- Clemen, R. T. (1989), "Combining Forecasts: A Review and Annotated Bibliography", *International Journal of Forecasting*, **5**, 559–583.
- Diekmann, A. and S. Gutjahr (1998), "Prediction of the Euro–Dollar Future Using Neural Networks – A Case Study for Financial Time Series Prediction", *University of Karlsruhe Working Paper, Proceedings of the International Symposium on Intelligent Data Engineering and Learning (IDEAL'98)*, Hong Kong, (<http://citeseer.nj.nec.com/diekmann98prediction.html>).
- Dunis, C. and X. Huang (2002), "Forecasting and Trading Currency Volatility: An Application of Recurrent Neural Regression and Model Combination", *The Journal of Forecasting*, **21**, 317–354.
- Dunis, C. and J. Jalilov (2002), "Neural Network Regression and Alternative Forecasting Techniques for Predicting Financial Variables", *Neural Network World*, **2**, 113–139.
- El-Shazly, M. R. and H. E. El-Shazly (1997), "Comparing the Forecasting Performance of Neural Networks and Forward Exchange Rates", *Journal of Multinational Financial Management*, **7**, 345–356.
- Fernandez-Rodriguez, F., C. Gonzalez-Martel and S. Sosvilla-Rivero (2000), "On the Profitability of Technical Trading Rules Based on Artificial Neural Networks: Evidence from the Madrid Stock Market", *Economics Letters*, **69**, 89–94.
- Fishman, M. B., D. S. Barr and W. J. Loick (1991), "Using Neural Nets in Market Analysis", *Technical Analysis of Stocks and Commodities*, **9**, 4, 135–138.
- Gençay, R. (1999), "Linear, Non-linear and Essential Foreign Exchange Rate Prediction with Simple Technical Trading Rules", *Journal of International Economics*, **47**, 91–107.
- Gouriéroux, C. and A. Monfort (1995), *Time Series and Dynamic Models*, translated and edited by G. Gallo, Cambridge University Press, Cambridge.
- Grabbe, J. O. (1996), *International Financial Markets*, 3rd edition, Prentice Hall, Englewood Cliffs, NJ.
- Granger, C. W. J. and R. Ramanathan (1984), "Improved Methods of Combining Forecasts", *Journal of Forecasting*, **3**, 197–204.
- Hanke, J. E. and A. G. Reitsch (1998), *Business Forecasting*, 6th edition, Prentice Hall, Englewood Cliffs, NJ.
- Hashem, S. (1997), "Optimal Linear Combinations of Neural Networks", *Neural Networks*, **10**, 4, 599–614 ([www.emsl.pnl.gov:2080/people/bionames/hashem\\_s.html](http://www.emsl.pnl.gov:2080/people/bionames/hashem_s.html)).
- Haykin, S. (1999), *Neural Networks: A Comprehensive Foundation*, 2nd edition, Prentice Hall, Englewood Cliffs, NJ.
- Hornik, K., M. Stinchcombe and H. White (1989), "Multilayer Feedforward Networks Are Universal Approximators", *Neural Networks*, **2**, 359–366.
- Kaasra, I. and M. Boyd (1996), "Designing a Neural Network for Forecasting Financial and Economic Time Series", *Neurocomputing*, **10**, 215–236.

- Kingdon, J. (1997), *Intelligent Systems and Financial Forecasting*, Springer, London.
- Lisboa, P. J. G. and A. Vellido (2000), "Business Applications of Neural Networks", in P. J. G. Lisboa, B. Edisbury and A. Vellido (eds), *Business Applications of Neural Networks: The State-of-the-Art of Real-World Applications*, World Scientific, Singapore, pp. vii–xxii.
- Maddala, G. S. (2001), *Introduction to Econometrics*, 3rd edition, Prentice Hall, Englewood Cliffs, NJ.
- Mehta, M. (1995), "Foreign Exchange Markets", in A. N. Refenes (ed.), *Neural Networks in the Capital Markets*, John Wiley, Chichester, pp. 176–198.
- Pesaran, M. H. and B. Pesaran (1997), "Lessons in Logit and Probit Estimation", in *Interactive Econometric Analysis Working with Microfit 4*, Oxford University Press, Oxford, pp. 263–275.
- Pindyck, R. S. and D. L. Rubinfeld (1998), *Econometric Models and Economic Forecasts*, 4th edition, McGraw-Hill, New York.
- Previa (2001), *Previa Version 1.5 User's Guide*, ([www.elseware.fr/previa](http://www.elseware.fr/previa)).
- Refenes, A. N. and A. Zaidi (1993), "Managing Exchange Rate Prediction Strategies with Neural Networks", in P. J. G. Lisboa and M. J. Taylor (eds), *Techniques and Applications of Neural Networks*, Ellis Horwood, Hemel Hempstead, pp. 109–116.
- Shapiro, A. F. (2000), "A Hitchhiker's Guide to the Techniques of Adaptive Nonlinear Models", *Insurance, Mathematics and Economics*, **26**, 119–132.
- Thomas, R. L. (1997), *Modern Econometrics. An Introduction*, Addison-Wesley, Harlow.
- Tyree, E. W. and J. A. Long (1995), "Forecasting Currency Exchange Rates: Neural Networks and the Random Walk Model", *City University Working Paper, Proceedings of the Third International Conference on Artificial Intelligence Applications*, New York, (<http://citeseer.nj.nec.com/131893.html>).
- Yao, J., H. Poh and T. Jasic (1996), "Foreign Exchange Rates Forecasting with Neural Networks", *National University of Singapore Working Paper, Proceedings of the International Conference on Neural Information Processing*, Hong Kong, (<http://citeseer.nj.com/yao96foreign.html>).
- Yao, J., Y. Li and C. L. Tan (1997), "Forecasting the Exchange Rates of CHF vs USD Using Neural Networks", *Journal of Computational Intelligence in Finance*, **15**, 2, 7–13.
- Zhang, G., B. E. Patuwo and M. Y. Hu (1998), "Forecasting with Artificial Neural Networks: The State of The Art", *International Journal of Forecasting*, **14**, 35–62.