Appl. Statist. (1998)
47, *Part* 2, *pp.* 231–250

Time series forecasting with neural networks: a comparative study using the airline data

Julian Faraway

University of Michigan, Ann Arbor, USA

and Chris Chatfield†

University of Bath, UK

[Received January 1996. Final revision April 1997]

Summary. This case-study fits a variety of neural network (NN) models to the well-known airline data and compares the resulting forecasts with those obtained from the Box–Jenkins and Holt–Winters methods. Many potential problems in fitting NN models were revealed such as the possibility that the fitting routine may not converge or may converge to a local minimum. Moreover it was found that an NN model which fits well may give poor out-of-sample forecasts. Thus we think it is unwise to apply NN models blindly in 'black box' mode as has sometimes been suggested. Rather, the wise analyst needs to use traditional modelling skills to select a good NN model, e.g. to select appropriate lagged variables as the 'inputs'. The Bayesian information criterion is preferred to Akaike's information criterion for comparing different models. Methods of examining the response surface implied by an NN model are examined and compared with the results of alternative non-parametric procedures using generalized additive models and projection pursuit regression. The latter imposes less structure on the model and is arguably easier to understand.

Keywords: Airline model; Akaike information criterion; Autoregressive integrated moving average model; Bayesian information criterion; Box–Jenkins forecasting; Generalized additive model; Holt–Winters forecasting; Projection pursuit regression

1. Introduction

Neural networks (NNs) have been vigorously promoted in the computer science literature for tackling a wide variety of scientific problems. Recently, statisticians have started to investigate whether NNs are useful for tackling various statistical problems (Ripley, 1993; Cheng and Titterington, 1994) and there has been particular attention to pattern recognition (Bishop, 1995; Ripley, 1996). NNs also appear to have potential application in time series modelling and forecasting but nearly all such work has been published outside the mainstream statistical literature. This work is reviewed in Section 7 and reveals that, contrary to some rather grandiose claims, the empirical evidence on NN forecasts indicates varying degrees of success.

It is pertinent to ask whether the success of NN modelling depends on

(a) the type of data
(b) the skill of the analyst in selecting a suitable NN model and/or
(c) the numerical methods used to fit the model and to compute predictions.

Experience regarding (a) can be built up with forecasting competitions, whereas case-studies

†*Address for correspondence*: Department of Mathematical Sciences, University of Bath, Bath, BA2 7AY, UK.
E-mail: cc@maths.bath.ac.uk

 0035–9254/98/47231

are better suited to assess (b) and (c), as well as throwing some light on (a). This paper describes one such case-study, based on the well-known airline data (see Section 2). After an introduction to NNs in Section 3, we describe our experiences in fitting and using NN models in Sections 4 and 6. Section 5 discusses ways of trying to understand the response surface implied by an NN model. Section 7 reviews our findings in the context of other studies and urges more understanding between statisticians and computer scientists.

We stress that we do not claim to propose any new methodology. Moreover, we deliberately chose to use public domain software to carry out our analyses to replicate the likely circumstances of an applied statistician trying out NNs for the first time. The 'novelty' of the paper lies in giving practical guidance on NN modelling and a comparison with alternative approaches from a statistical, as opposed to computing science, point of view.

2. Box–Jenkins analysis of the airline data

The main time series used in this paper is the so-called airline data, listed by Box *et al.* (1994), series G, and earlier by Brown (1962). Fig. 1 shows that the data have an upward trend together with seasonal variation whose size is roughly proportional to the local mean level (called *multiplicative* seasonality). The presence of multiplicative seasonality was one reason for choosing this data set. A common approach to dealing with this type of seasonality is to choose an appropriate transformation, usually logarithms, to make the seasonality *additive*. However, the discussion of Chatfield and Prothero (1973) demonstrated the difficult nature of such a choice, and if NN models could deal with the non-linearity that is inherent in multiplicative seasonality and allow the raw data to be analysed this would obviate one awkward step in the usual approach to time series analysis. Moreover, the length of the series was typical of data found in forecasting situations, the data were widely available and non-

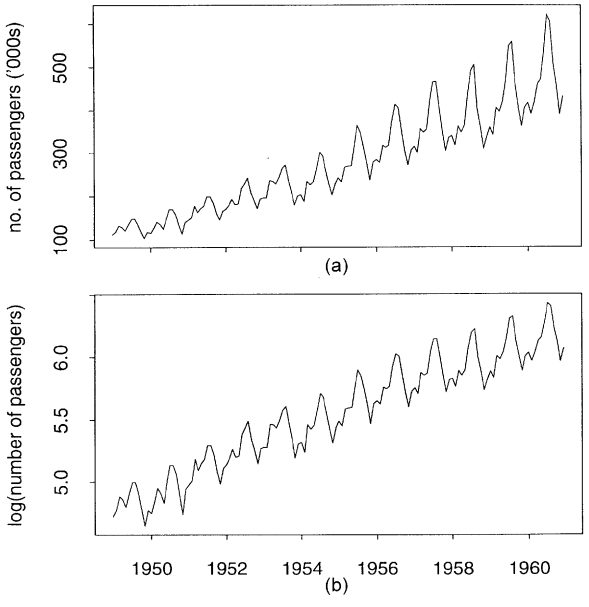

Fig. 1. Airline data; monthly totals (in thousands) of international airline passengers from January 1949 to December 1960: (a) raw data; (b) natural logarithms

confidential and we wanted to see whether we could replicate the promising results obtained by some computer scientists (Tang *et al*., 1991).

The standard Box–Jenkins analysis (e.g. Harvey (1993) and Box *et al*. (1994)) involves taking natural logarithms of the data followed by seasonal and non-seasonal differencing to make the series stationary. A special type of seasonal autoregressive integrated moving average (SARIMA) model, of order $(0, 1, 1) \times (0, 1, 1)_{12}$ in the usual notation (e.g. Box *et al*. (1994), p. 333), is then fitted. This model is often called the *airline model* and is used as the yardstick for future comparisons, though other SARIMA models could be found with a similar fit and forecast accuracy.

Most of the results reported in this paper were computed by fitting a model to the first 11 years of data and then making forecasts of the last 12 monthly observations. The forecasts were obtained either from the base month 132 using only data available at that time, giving *multistep* forecasts, or by bringing in the recent observed data one at a time, giving *one-step* forecasts. The model parameters were *not* re-estimated at each step when computing one-step forecasts.

For each model fitted, using data up to time T, we computed the following statistics:

(a) S, the sum of squared residuals up to time T (the residuals are the within-sample one-step-ahead forecast errors);
(b) $\hat{\sigma} = \sqrt{\{S/(n-p)\}}$, the estimate of residual standard deviation, where n denotes the number of *effective* observations used in fitting the model and p denotes the number of parameters fitted in the model; thus, when fitting the airline model to the airline data with $T = 132$, the value of n is $132 - 13 = 119$, since 13 observations are 'lost' by differencing;
(c) the Akaike information criterion (AIC), $n \ln(S/n) + 2p$;
(d) the Bayesian information criterion (BIC), $n \ln(S/n) + p + p \ln(n)$;
(e) SS_{MS}, the sum of squares of multistep-ahead forecast errors made at time T of the observations from time $T+1$ to the end of the series; these are the out-of-sample (genuine *ex ante*) forecasts;
(f) SS_{1S}, the sum of squares of one-step-ahead (out-of-sample) forecast errors of the observations from time $T+1$ to the end of the series.

The residual sum of squares, S, can only become smaller and the residual standard deviations $\hat{\sigma}$ will tend to become smaller as a model is made 'larger'. Thus the minimization of a criterion such as the AIC or BIC is more satisfactory for choosing a 'best' model from candidate models having different numbers of parameters. Strictly speaking (c) and (d) above are approximations to the variable parts of the AIC and BIC respectively. In both cases the first term is a measure of (lack of) fit and the remainder is a penalty term to prevent over-fitting. The BIC penalizes extra parameters more severely than the AIC does, leading to 'smaller' models. Several similar criteria have been proposed including alternative closely related Bayesian criteria which depend on different priors on model size. In particular Schwarz's Bayesian criterion (SBC) has the penalty term $p \ln(n)$ rather than $p + p \ln(n)$ (Priestley (1981), pages 375–376). The SBC gives results that are qualitatively similar to the BIC used in this paper and its use here for the airline data would not change the models that we select. The reader should note that the SBC is sometimes (confusingly) abbreviated as BIC.

For the airline model fitted to the airline data with $T = 132$, the MINITAB package (release 9.1) gave the following values (after back-transforming all forecasts from the model for the logged data into the original units):

(i) $S = 10789$;
(ii) $\hat{\sigma} = 9.522$;
(iii) AIC $= 540.35$;
(iv) BIC $= 547.91$;
(v) $\text{SS}_{\text{MS}} = 3910$;
(vi) $\text{SS}_{\text{1S}} = 4328$.

The one-step forecasts have slightly worse accuracy than the multistep forecasts as will happen occasionally. If the airline model is fitted to the *raw* data, rather than to the logarithms, then the fit is about 20% worse ($S = 12920$) while the accuracy of forecasts suffers even more (e.g. $\text{SS}_{\text{MS}} = 5230$ is 34% worse).

An alternative way to compute forecasts for the airline data, without taking logarithms, is to use the multiplicative version of Holt–Winters exponential smoothing (e.g. Chatfield and Yar (1988)) and this gave forecasts with comparable accuracy to Box–Jenkins forecasts. Although applied to the raw data, the multiplicative Holt–Winters method is inherently non-linear in that the formula for a point forecast is a non-linear function of past observations.

3. Neural networks

The following brief account of NNs is intended to make this paper as self-contained as possible. However, the reader may also find it helpful to read Ripley (1993), Sarle (1994), Stern (1996), Warner and Misra (1996) and/or Chatfield (1996a), section 11.4. Alternatively an introduction from a computer science perspective such as Hertz *et al.* (1991) and Gershenfeld and Weigend (1994) may be helpful, while econometric and financial perspectives are provided by Kuan and White (1994) and Azoff (1994) respectively.

This paper considers one popular form of (artificial) NN called the *feed-forward* NN with one hidden layer. In time series forecasting, we wish to predict future observations by using some function of past observations. One key point about NNs is that this function *need not be linear*, so that an NN can be thought of as a sort of non-linear (auto)regression model.

Fig. 2 depicts a typical *architecture* as applied to time series forecasting with monthly data. The value at time t is to be forecasted using the values at lags 1 and 12. The latter are regarded as *inputs* whereas the forecast is the *output*. The illustrated example includes one hidden layer of two *neurons* (often called *nodes* or *processing units* or just *units*). In addition there is a

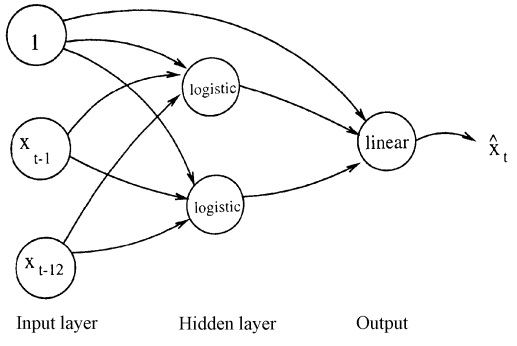

Fig. 2. Architecture of a typical NN for time series forecasting with one hidden layer of two neurons: the output (the forecast) depends on the lagged values at times $t - 1$ and $t - 12$

constant input term which for convenience may be taken as 1. Each input is connected to both the (hidden) neurons, and both neurons are connected to the output. There is also a direct connection from the constant input to the output. The 'strength' of each connection is measured by a quantity called a *weight*. A numerical value is calculated for each neuron as follows. First a linear function of the inputs is found, say $\Sigma\, w_{ij} y_i$ where w_{ij} denotes the weight of the connection between input y_i and the jth neuron. The values of the inputs in our example are $y_1 = 1$, $y_2 = x_{t-1}$ and $y_3 = x_{t-12}$. The linear sum, say ν_j, is then transformed by applying a function called an *activation function*, which is typically non-linear. A commonly used function is the *logistic function*, $z_j = 1/\{1 + \exp(-\nu_j)\}$, which gives values in the range (0, 1). In our example this gives values z_1 and z_2 for the two neurons. A similar operation can then be applied to the values of z_1, z_2 and the constant input to obtain the predicted output. However, the logistic function should not be used at the output stage in time series forecasting unless the data are suitably scaled to lie in the interval (0, 1). Instead a linear function of the neuron values may be used, which implies the identity activation function at the output stage.

The introduction of a constant input unit, connected to every neuron in the hidden layer and also to the output, avoids the necessity of separately introducing what computer scientists call a *bias*, and what statisticians would call an intercept term, for each unit. Essentially the biases just become part of the set of weights (the model parameters).

For an NN model with one hidden level, the general prediction equation for computing a forecast of x_t (the output) using selected past observations, $x_{t-j_1}, \ldots, x_{t-j_k}$, as the inputs, may be written (rather messily) in the form

$$\hat{x}_t = \phi_o\left\{w_{co} + \sum_h w_{ho}\,\phi_h\left(w_{ch} + \sum_i w_{ih} x_{t-j_i}\right)\right\} \tag{1}$$

where $\{w_{ch}\}$ denote the weights for the connections between the constant input and the hidden neurons and w_{co} denotes the weight of the direct connection between the constant input and the output. The weights $\{w_{ih}\}$ and $\{w_{ho}\}$ denote the weights for the other connections between the inputs and the hidden neurons and between the neurons and the output respectively. The two functions ϕ_h and ϕ_o denote the activation functions used at the hidden layer and at the output respectively. One minor point is that the labels on the hidden neurons can be permuted without changing the model.

We use the notation NN($j_1, \ldots, j_k$; h) to denote the NN with inputs at lags $j_1, \ldots, j_k$ and with h neurons (or units) in the one hidden layer. Thus Fig. 2 represents an NN(1, 12; 2) model.

The weights to be used in the NN model are estimated from the data by minimizing the sum of squares of the within-sample one-step-ahead forecast errors, namely $S = \Sigma_t\,(\hat{x}_t - x_t)^2$, over the first part of the time series, called the *training set* in NN jargon. This is not an easy task as the number of weights may be large and the objective function may have local minima. Various algorithms have been proposed, but even the better procedures may take several hundred iterations to converge, and yet may still converge to a local minimum. The NN literature tends to describe the iterative estimation procedure as being a 'training' algorithm which 'learns by trial and error'. Our software used a popular algorithm called *back-propagation* for computing the first derivatives of the objective function. There are many ways to use these derivatives for optimization and our fitting method relied on the Broyden–Fletcher–Goldfarb–Shanno algorithm (Fletcher, 1987) which is a quasi-Newton method. The starting values chosen for the weights can be crucial and it is advisable to try several different

sets of starting values to see whether consistent results are obtained. Other optimization methods are still being investigated and different packages may use different fitting procedures. For example, a technique called *simulated annealing* (e.g. van Laarhoven and Aarts (1987)) can be used to try to avoid local minima but this requires the analyst to set numerical parameters with names like 'the cooling rate', and even then there is no guarantee that convergence to a global minimum will occur.

The last part of the time series, called the *test set*, is kept in reserve so that genuine out-of-sample (*ex ante*) forecasts can be made and compared with the actual observations. Equation (1) effectively gives a *one-step-ahead forecast* as it uses the actual observed values of all lagged variables as inputs. If *multistep-ahead forecasts* are required, then it is possible to proceed in one of two ways. Firstly, we could construct a new architecture with several outputs, giving $\hat{x}_t, \hat{x}_{t+1}, \hat{x}_{t+2}, \ldots,$ where each output would have separate weights for each connection to the neurons. Secondly, we could 'feed back' the one-step-ahead forecast to replace the lag 1 value as one of the input variables, and the same architecture could then be used to construct the two-step-ahead forecast, and so on. We adopted the latter iterative approach because of its numerical simplicity and because it requires fewer weights to be estimated. Some analysts fit NN models to obtain the best forecasts of the test set data, rather than the best fit to the training data. Then a third section of data needs to be kept in reserve so that genuine out-of-sample forecasts can be assessed.

The number of parameters in an NN model is typically much larger than in traditional time series models and for a single-layer NN model is given by $p = (n_i + 2)n_u + 1$ where n_i denotes the number of input variables (excluding the constant) and n_u denotes the number of hidden neurons (or units). For example, the architecture in Fig. 2 (where n_i and n_u are both 2) contains nine connections and hence has nine parameters (weights). Because of this large number, there is a real danger that the algorithm may 'overtrain' the data and produce a spuriously good fit which does not lead to better forecasts. This motivates the use of model comparison criteria, such as the BIC, which penalize the addition of extra parameters. It also motivates the use of an alternative fitting technique called *regularization* (e.g. Bishop (1995), section 9.2) wherein the 'error function' is modified to include a penalty term which prefers 'small' parameter values (analogous to the use of a 'roughness' penalty term in nonparametric regression with splines). We did not pursue this approach.

NN modelling is nonparametric in character and it has been suggested that the whole process can be completely automated on a computer

> 'so that people with little knowledge of either forecasting or neural nets can prepare reasonable forecasts in a short space of time'

(Hoptroff, 1993). This *black box* character can be seen as an advantage but we think it potentially dangerous. Certainly black boxes can sometimes give silly results and NN models obtained like this are no exception. Thus Gershenfeld and Weigend (1994), p. 7, found that

> 'there was a general failure of simplistic ''black-box'' approaches — in all successful entries (in the Sante Fe competition), exploratory data analysis preceded the algorithm application'.

Our case-study demonstrated that a good NN model for time series data must be selected by combining traditional modelling skills with knowledge of time series analysis and of the particular problems involved in fitting NN models. Problem formulation is, as always, critical and it is 'unlikely that applied statistics will be reduced to an automatic process in the foreseeable future' (Sarle, 1994).

4. Fitting neural network models to the airline data

Many commercial packages are available for fitting NN models. James (1994) reviewed 12 such items. We deliberately eschewed commercial software, partly for financial reasons (some are very expensive), and partly because they are typically written for the business user. For example, whereas James (1994) would 'strongly recommend' a package called 4Thought, Harvey and Toulson (1994) described the claims made in its publicity material as being 'totally unsubstantiated'. Appendix A gives information on the software that we developed based on public domain S-PLUS functions, and we think that this sort of software will appeal more to the average statistician. We realize that different software packages use different optimization methods for fitting NN models but, although the implementations differ, we think that the problems encountered below are representative of those that are likely to occur with all packages, given the difficulty in choosing an appropriate architecture, the large number of parameters which must be estimated, the non-linearity of the model and the existence of multiple minima in the sum-of-squares surface.

Our case-study will focus on some general issues including

(a) the choice of input variables,
(b) the choice of architecture and activation function and
(c) the criterion for selecting the 'best' model.

Our software is designed to handle NNs with a single hidden layer (the usual situation), and so the choice of architecture is primarily about choosing the number of neurons in the hidden layer.

As well as these general issues, the case-study also focuses on some more *ad hoc* problems such as

(d) deciding whether the data need scaling and
(e) choosing starting values for the weights.

We think that these particular practical difficulties are of more general interest as being illustrative of the sort of problems that the analyst is likely to encounter in NN modelling. There were two immediate such problems. The untransformed airline data lie in the range 104–622. The default choice of the activation function at the output stage is the logistic function, but this constrains the output (the forecasts) to be in the range (0, 1). Failure to specify the identity activation function gave ridiculous results. Furthermore the starting values used in the algorithm are out of scale with the input values so that the fitting algorithm failed to converge in a sensible way. Thus we found it necessary to rescale the data, and dividing by 100 was found to work satisfactorily. All subsequent numbers, including forecasts and comparisons with other models, refer to these scaled data. We find it difficult to offer general advice on the choice of scaling except to use trial and error and to try different values. The starting weights must vary over a reasonable range, neither too wide nor too narrow, compared with the range of the data. We found that dividing by 1000, rather than 100, also led to convergence problems, although dividing by 1000 would (temporarily at least) have ensured that all short-term forecasts were in the range (0, 1) so that the logistic activation function could have been used at the output stage. The need for attention to such numerical details is clear.

The next serious problem that we found, which appears to apply more generally to NN model fitting, was that consecutive restarts of the fitting algorithm, with different random starting points for the weights, typically found several different local minima (or even

Table 1. Fit and forecast accuracy for seven local minima for the NN(1, 12; 2) model with $T = 132$ in ascending order of fit accuracy (S)†

Fit S	*Forecast accuracy*	
	SS_{MS}	SS_{1S}
2.30	0.35	0.34
2.32	0.38	0.38
2.38	0.34	0.31
2.41	0.35	0.31
2.49	0.33	0.33
2.49	0.35	0.35
2.51	0.44	0.40

†The notation is defined in Sections 2 and 3 and all numbers refer to the scaled data.

saddlepoints). For example, seven distinct local minima were found for the NN(1, 12; 2) model and the resulting fit and forecast accuracy are given in Table 1. As the algorithm can converge to a saddlepoint, it is important to check that the Hessian is positive definite in each case (though this can be difficult to do because the minima tend to be flat) to ensure that they all represent local minima. Even though the algorithm was restarted many times from different starting points, there is no guarantee that the model in Table 1 with the smallest S-value gives the global minimum. All the models reported hereafter are the result of refitting the model at least 50 times from different random starting points and taking the best of the resulting minima.

The problem of multiple minima is important because the fitted weights are not stable across different minima. Table 2 shows the fitted weights for the fifth and seventh minima in Table 1 and large differences are apparent. In particular, the second neuron in the first model has much more effect on the output than the first neuron has, whereas the two neurons in the second model have nearly equal weights. This instability suggests that it is unwise to try to interpret the fitted weights since, even at the global minimum, the weights could be changed substantially without changing the fit or the forecasts very much (as in near multicollinear regression).

We now compared various NN models having different input (lagged) variables and different architectures. All the models had one hidden layer, were fitted using the first 132 observations (the training set) and were used to give forecasts of the last 12 observations (the test set). As noted in Section 2, the AIC and BIC provide a fairer comparison of models with different numbers of parameters than does the residual (fit) root mean square. For NN models, the number of parameters, p, is the number of weights, while n, the number of effective observations, depends on the maximum lag. The results for a range of models are shown in Table 3 together with the corresponding values for the Box–Jenkins airline model.

Table 2. Comparison of weights from two different local minima†

w_{c1}	w_{11}	w_{21}	w_{c2}	w_{12}	w_{22}	w_{co}	w_{1o}	w_{2o}
−27.61	4.19	4.62	−0.86	0.05	0.23	−4.62	−0.14	16.10
−4.71	0.01	0.74	−0.93	0.20	0.53	−1.40	6.73	5.32

†The notation is defined in Section 3.

Table 3. Comparison of various NN models together with the corresponding values for the Box–Jenkins airline model†

Lags	*Number of hidden neurons*	*Number of parameters*	*Measures of fit*				*Forecast accuracy*	
			S	$\hat{\sigma}$	*AIC*	*BIC*	SS_{MS}	SS_{IS}
1, 2, 3, 4	2	13	7.74	0.245	−333	−283	58.52	1.03
1–13	2	31	0.73	0.091	−545	−428	1.08	0.71
1–13	4	61	0.26	0.067	−605	−375	4.12	1.12
1, 12	2	9	2.30	0.144	−456	−422	0.35	0.34
1, 12	4	17	2.16	0.145	−448	−383	0.38	0.44
1, 12	10	41	1.77	0.150	−424	−268	0.51	0.59
1, 2, 12	2	11	2.17	0.141	−459	−418	0.34	0.29
1, 2, 12	4	21	1.91	0.139	−455	−375	6.82	1.03
1, 2, 12, 13	2	13	0.99	0.097	−543	−494	0.37	0.52
1, 2, 12, 13	4	25	0.81	0.093	−543	−449	0.34	0.52
1, 12, 13	1	6	1.18	0.102	−537	−514	0.33	0.50
1, 12, 13	2	11	1.03	0.098	−543	−501	0.33	0.50
1, 12, 13	4	21	0.84	0.093	−547	−467	0.54	0.62
Box–Jenkins model		2	1.08	0.095	−556	−546	0.39	0.43

†The notation is defined in Section 2 and all numbers refer to the scaled data.

(All forecasts have been back-transformed and all statistics suitably scaled to make them comparable with the NN values fitted to the data divided by 100. In particular, the scaled AIC and BIC values were obtained by subtracting $119\ln(10000)$.)

For illustration, we tried a wide variety of NN models. The NN(1–4; 2) model is the sort of model which might be tried by someone with no training in time series; its fit is poor and its predictive performance, particularly in the long term, is awful. Clearly NN models cannot compensate for a poor choice of input variables. Someone with slightly more experience, realizing that these are annual data, might use all lags up to lag 12 or 13, or perhaps even higher. These models, with many parameters, achieve small S-values which might lead the naïve to suppose that they had found a good model, especially in view of the AIC values. However, the BIC values tell a different story and the predictive performance is poor. With still more knowledge of time series, it will seem reasonable to include lag 12 without necessarily including all intervening lags, and NN models with inputs at lags (1, 2, 12, 13) or lags (1, 12, 13) generally lead to better forecasts and better BIC values, provided that not too many hidden neurons are used.

If the model is chosen on the basis of minimizing the AIC or $\hat{\sigma}$, then the NN(1–13; 4) model will be selected, leading to poor predictions. Experienced time series analysts would guess intuitively that the use of so many inputs is unlikely to give good results and so it is alarming that this model gives the 'best' AIC. Clearly, in this context, the use of the AIC does not do enough to penalize extra parameters. In contrast, the BIC picks the NN(1, 12, 13; 1) model which is much more plausible and gives sensible results (but notice that the BIC for the Box–Jenkins airline model is even better!).

It has been suggested that we try the bias-corrected version of the AIC (AIC_C), as recommended by Brockwell and Davis (1993), section 9.3, which is obtained by adding $2(p+1)(p+2)/(n-p-2)$ to the AIC in our notation. This makes little difference for small values of p but, for larger values of p, penalizes extra parameters (much) more severely than the AIC and selected the 11-parameter NN(1, 12, 13; 2) model. Thus it is more akin to the BIC, than the AIC, in its effect on model selection. However, many statisticians are not

Table 4. Weights in the NN(1–13; 2) model for the connections from the 13 lagged values to the first hidden neuron (first row) and second hidden neuron (second row)

Weights for the following lags:												
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*
0.26	−0.06	0.02	−0.01	0.05	−0.05	0.06	0.01	0.06	−0.09	−0.03	0.21	−0.20
2.70	−3.47	2.53	−0.69	−1.39	−0.36	2.52	4.27	2.16	−4.25	−3.05	2.43	−3.31

familiar with AIC_C and most software computes the ordinary AIC. Thus we retain the AIC values for comparative purposes, while noting that the AIC_C deserves more attention.

An obvious question is whether the better NN models found above (i.e. models that include lags 1, 12 and maybe 13, but exclude intervening lags) could have been discovered without prior knowledge of seasonal ARIMA models. More generally how can we choose the input variables for an NN model? One possibility is to examine the fitted weights for a model with many inputs and to see which weights are 'large'. For example, Table 4 shows the weights of the connections between the 13 lagged values and the two neurons in the NN(1–13; 2) model. Notice that the weights connecting the lagged values to the first hidden neuron show small values for lags 2–11. This suggests that we drop the intermediate lags. Admittedly, for connections to the second hidden neuron, lags 2–11 have larger weights, but the weights from the two hidden neurons to the output unit are 16.04 and −0.90 respectively, so that the second hidden neuron has much less effect on the output. Thus an examination of the weights helps to identify important input variables in this example. However, in view of the instability in estimating the weights mentioned earlier, this may be the exception rather than the rule, and our experience in Chatfield and Faraway (1996) unfortunately suggests the former. Thus, as yet, we know of no general way to select input variables except to use the context and knowledge about other models fitted to similar data.

Why does having a high number of hidden neurons lead to poor forecasts? Consider the NN(1, 2, 12; 2) and NN(1, 2, 12; 4) models which have the same lags but different numbers of neurons. The prediction surfaces for these two models vary over the three lags and so cannot be plotted directly, but can be viewed along specified directions. Fig. 3 plots the predicted one-step response along the line $x_{t-1} = x_{t-2} = 0.87x_{t-12} + 0.07$, chosen because the data are

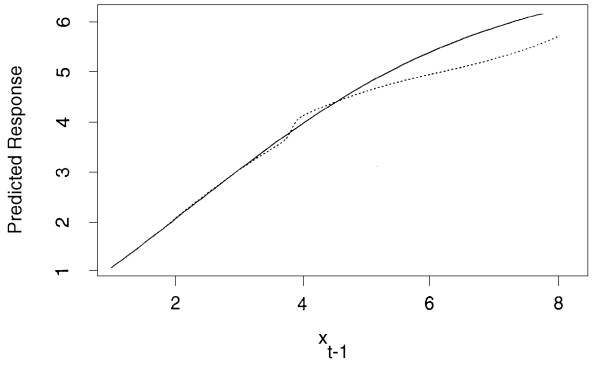

Fig. 3. Predicted response (in scaled units) for NN(1, 2, 12; 2) (———) and NN(1, 2, 12; 4) (·····) models along the direction $x_{t-1} = x_{t-2} = 0.87x_{t-12} + 0.07$

dense in this direction. It shows how the model with the higher number of neurons (and parameters) contorts to achieve a better fit to the middle range of the data, only to give predictions for higher values which are intuitively pulled down too far.

Some other NN models were considered that were not the result of a pure NN approach, but rather involved the sort of intelligent pre-examination of the data that would naturally be carried out by a sensible statistician. We had hoped that NN models could handle the data without the necessity of a power transformation, but Fig. 1 suggests that we take logarithms of the data to stabilize the seasonality and variance. So three plausible NN models for the logged data were explored:

(a) an NN(1, 12, 13; 2) model (for the logarithms) (model 1);
(b) remove the linear trend (from the logarithms) and then remove the seasonal trend by subtracting the monthly averages; then apply an NN(1–4; 2) model (model 2);
(c) apply first and seasonal differencing ($\nabla\nabla_{12}$) to the logarithms and then use an NN(1–4; 2) model (model 3).

Model 1 has the same NN structure as one of the best models for the raw data but is now applied to the logarithms. Models 2 and 3 have the trend and seasonal variation removed before fitting an NN model. The choice of inputs at lags 1–4 was not sensible for the raw data (see Table 3), but it seems reasonable to ignore longer lags for the detrended, deseasonalized data (though we could have tried other alternatives such as lags 1–3). Model 3 is akin to the airline model in that we apply an NN model to what is left after differencing the logged data. The results are shown in Table 5. Natural logarithms were used throughout, but fitted values and forecasts were appropriately transformed so that the entries in Table 5 are comparable with those appearing in Table 3. The fits of the NN(1, 12, 13; 2) models for the untransformed and the log-transformed data are similar (in contrast with the findings for the Box–Jenkins models). A possible explanation is the local linearity of the log-transformation — the ratio of the maximum to the minimum values for the whole series is about 5. But, as the current prediction depends primarily on the values at lags 12 and 13, it is the ratio x_t/x_{t-12} or x_t/x_{t-13} that really matters. The maximum value of this ratio is 1.4 indicating that the curvature introduced by the log-transform is not important.

One general problem with taking differences and making transformations is that it is not clear how many parameters should be counted for such operations. For example should taking logarithms count as an extra parameter? More generally for model 2 above, the fit criteria look very good but one could argue that detrending and deseasonalizing correspond to including $2 + 12 = 14$ more parameters, so that there are 27 parameters, not 13. Recalculating the AIC, BIC and $\hat{\sigma}$ on this basis gives -572, -468 and 0.097 respectively, which makes the model less persuasive. Sadly we can offer no general guidance on how many parameters

Table 5. Results for the three NN models fitted to the logged data†

Model	*Number of parameters*	*Measures of fit*				*Forecast accuracy*	
		S	$\hat{\sigma}$	*AIC*	*BIC*	SS_{MS}	SS_{IS}
1	11	1.04	0.098	−542	−500	0.44	0.63
2	13	0.96	0.091	−600	−550	0.66	0.56
3	13	1.14	0.105	−504	−456	4.35	0.67

†The notation is defined in Section 2 and all numbers refer to the scaled data.

should be allowed for. Model 3 gives a worse fit and worse forecasts than both model 2 and the airline model (see the Box–Jenkins model in Table 3) (and even gives worse forecasts than the 'naïve' model $\nabla\nabla_{12}\ \log(x_t)$ for which $SS_{MS} = 2.16$ and $SS_{1S} = 0.683$). Of course, once the data have been log-transformed, there is very little non-linearity for an NN model to capture and so it is perhaps not surprising that NN models for the logarithms are unlikely to be able to improve on a linear model.

We close with a general comment on the results for all the models considered in this section. An inspection of Tables 3 and 5 shows that the within-sample (fit) estimate of the error standard deviation, $\hat{\sigma}$, is typically much less than the prediction error standard deviation, namely $\sqrt{(SS_{1S}/12)}$. The best values of $\hat{\sigma}$ are around 0.1, whereas the best estimate of the prediction error standard deviation is 0.15 (for the NN(1, 2, 12; 2) model) but is more typically around 0.2. This is common in time series forecasting where the within-sample fit usually appears better than the out-of-sample prediction. The overoptimism caused by choosing the best of many fitted models and then behaving as if the selected model were known to be true has been well documented, both theoretically and empirically, in work on model uncertainty (Chatfield, 1995a, 1996b). The wrong model may be selected, the model may change in the future or there may not be a 'true' model anyway. In each situation forecasts are likely to be worse than expected. This is particularly important in regard to NN modelling as it can appear persuasive to add extra hidden units and layers (and hence extra parameters) to improve the fit, forgetting that out-of-sample forecasts may become worse (see Stern (1996), section 3.2, for a recent example). This re-emphasizes that all forecasting comparisons should be made on an out-of-sample basis.

5. Alternative ways of looking into the black box

One major criticism that is levelled at the NN models is that they provide a black box approach which may produce satisfactory forecasts but give little insight into the structure of the data. It can be very hard to interpret an NN model by just looking at the architecture and weights. This section examines alternative ways of 'looking inside the black box', including procedures based on alternative methodology. When there are only two inputs, the prediction surface can be plotted as in Fig. 4 for the NN(1, 12; 2) model. Notice that the surface is non-linear; however, if attention is focused on the region where most of the data occur (which is around the main diagonal) then the surface is close to linear.

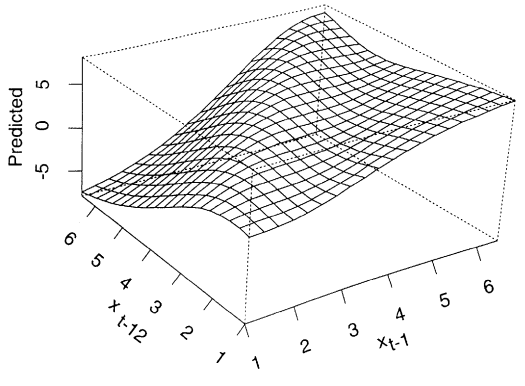

Fig. 4. Prediction surface (in scaled units) for the NN(1, 12; 2) model

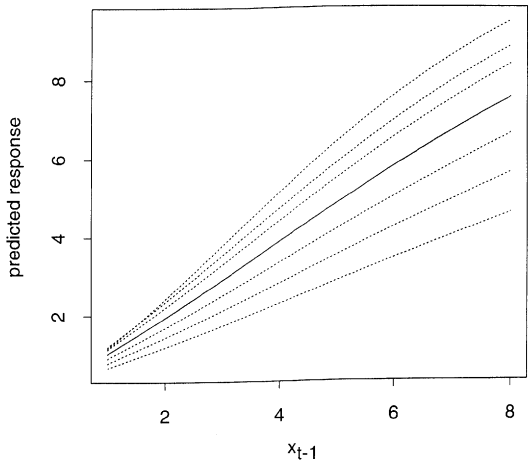

Fig. 5. Predicted response (in scaled units) for the NN(1, 2, 12, 13; 2) model along the direction of the first principal component of $\{x_{t-1}, x_{t-2}, x_{t-12}, x_{t-13}\}$ (———) and predicted responses along the first principal component perturbed by ± 0.1, ± 0.2 and ± 0.3 in the direction of the second principal component (·····)

When there are more than two inputs, it is not possible to plot the prediction surface so easily and we can only view the predictions along specified directions. For the NN(1, 2, 12, 13; 2) model, Fig. 5 shows the predicted response along the direction of the first principal component of $\{x_{t-1}, x_{t-2}, x_{t-12}, x_{t-13}\}$ as x_{t-1} varies (in scaled units). Most of the data lie within a distance of ± 0.2 in the direction of the second principal component where we see that the prediction surface is close to linear. This happens when the weighted sum of inputs into those hidden neurons that have much effect on the output happen to fall in the central, nearly linear, region of the logistic activation function. Thus these plots (which are not usually considered in NN analysis) demonstrate that the prediction surface is approximately linear in the region of interest. Such plots will not necessarily be useful for all data and all NN models but they are valuable here.

We now consider two alternative statistical approaches, under the general title of *non-parametric identification of non-linear time series* (Tjøstheim and Auestad, 1994), which enable prediction surfaces to be looked at more directly. *Generalized additive models* (GAMs) (Hastie and Tibshirani, 1990) can be used to represent the predicted response in terms of a sum of functions of the chosen inputs. Choosing lags 1, 2, 12 and 13 for example, this may be written

$$\hat{x}_t = f_1(x_{t-1}) + f_2(x_{t-2}) + f_{12}(x_{t-12}) + f_{13}(x_{t-13})$$

where the functions f_i are estimated nonparametrically from the data, rather than being prespecified. Using S-PLUS software, the GAM was fitted as described in Chambers and Hastie (1992) with the functions f_i estimated using splines. The estimated functions, along with ± 2 standard error pointwise confidence bands, are shown in Fig. 6.

The fitted function for x_{t-2} is not significantly different from constant zero so we have the immediate message that this variable may be excluded from the model. In addition, the other fitted functions are all close to linear. If x_{t-2} is excluded, the refitted GAM gives a very similar fit. The GAM can be used to make predictions when splines are used for the fitting since these fits can be extrapolated. Of course, it is dangerous to extrapolate too far but the same

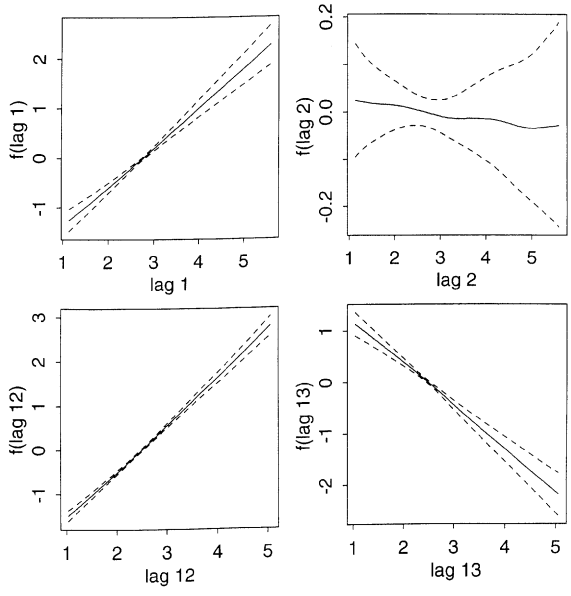

Fig. 6. Four fitted additive functions for the terms of a GAM in scaled units, corresponding to lags 1, 2, 12 and 13 (———) and ±2 standard error confidence bands (- - - -)

problem would arise with an NN model. The results, shown in Table 6, are comparable with the best NN models obtained earlier.

We tried GAMs on other time series and found them to be a useful exploratory technique. In particular, when applied to the famous sunspots data (Box *et al.* (1994), series E) which nearly everyone agrees are *not* linear, the functions were indeed found not to be linear but to be approximately two straight lines with a bend in the middle. This suggests that a threshold model (e.g. Tong (1990)) might be appropriate rather than an NN model.

An alternative approach is provided by *projection pursuit regression* (PPR) (see Jones and Sibson (1987)). Given inputs $\mathbf{x} = (x_1, \ldots, x_p)^{\mathrm{T}}$ and output y, then the PPR model is

$$\hat{y} = \bar{y} + \sum_{m=1}^{M_0} \beta_m \, \phi_m(\boldsymbol{\alpha}_m^{\mathrm{T}} \mathbf{x}),$$

where $\boldsymbol{\alpha}_m$ denotes a vector of constants of appropriate length, β_m is a constant and the 'activation' functions ϕ_m, scaled to have zero mean and unit variance, are estimated non-parametrically from the data, rather than given some prespecified form as in NN modelling.

Setting $(x_{t-1}, x_{t-2}, x_{t-12}, x_{t-13})$ as the input and x_t as the output, we first determined that the best choice of M_0 is 1 as larger values improve the fit to a negligible extent. The function ϕ_1 was

Table 6. Results for the GAM

Lags	*Number of parameters*	*Measures of fit*				*Forecast accuracy*	
		S	$\hat{\sigma}$	*AIC*	*BIC*	SS_{MS}	SS_{IS}
1, 12, 13	13	1.14	0.104	−527	−478	0.43	0.55

Table 7. Results for the linear regression model†

Number of parameters	*Measures of fit*				*Forecast accuracy*	
	S	$\hat{\sigma}$	*AIC*	*BIC*	SS_{MS}	SS_{IS}
4	1.18	0.101	−541	−526	0.32	0.51

†The notation is defined in Section 2.

found to be virtually linear with $\boldsymbol{\alpha}_1 = (0.504, -0.008, 0.681, -0.531)$. The small coefficient for x_{t-2} implies again that this term is not needed and the PPR model obtained is very similar to the earlier GAM model. It is rather more difficult in practice to make predictions using a PPR model since the 'activation function' is estimated using the 'supersmoother' method (Friedman, 1984) which does not lend itself readily to extrapolation. This smoother could be replaced but this would involve a difficult rewriting of Friedman's software. The fitted sum of squares S for the PPR model is 1.145 which is comparable with the value for the GAM model.

It may seem surprising at first that the GAM and PPR models give 'activation' functions which are approximately linear, when the NN models fitted earlier used the (non-linear) logistic function. In fact the logistic function *is* close to linear in its midrange, and this is where the bulk of the data lie.

All this suggests that we may be able to use a simple linear regression model with $\{x_{t-1}, x_{t-12}, x_{t-13}\}$ as explanatory variables. The fitted linear regression equation (for the data after scaling by dividing by 100 which changes the constant but not the other coefficients) is

$$\hat{x}_t = 0.0322 + 0.7824x_{t-1} + 1.0720x_{t-12} - 0.8394x_{t-13}$$

and the estimated coefficients are nearly proportional to the corresponding values obtained by PPR. Predictions are easily made using a linear regression model and the results are shown in Table 7. A comparison with Table 3 shows that the forecasts are excellent and the model, with only four parameters, gives the smallest value for the BIC yet seen!

6. Results for a different time period

To see whether there is any qualitative change in the results when we predict from a different part of the seasonal cycle, we extended the test set to the last 18 monthly observations (giving six fewer observations in the training set). The results for various models are shown in Table 8.

As before, the best BIC value is given by the NN(1, 12, 13; 1) model. The one-step forecasts are reasonable considering that the prediction sum of squares is now computed over 18 steps (so that it needs to be discounted by a third to compare with the previous values for the last 12 observations). The multistep forecasts are much worse than before, partly because some lead times are more than 1 year ahead. It is worrying that none of the NN models gives such a good BIC value or such good forecasts as the airline model. In other respects the results are qualitatively similar to those in Table 3. NN models with higher numbers of parameters tend to give better fits but worse forecasts, and the AIC does not penalize additional parameters sufficiently.

Table 8. Results for various NN models and the Box–Jenkins airline model fitted to the first 126 observations†

Lags	*Number of hidden neurons*	*Number of parameters*	*Measures of fit*				*Forecast accuracy*	
			S	$\hat{\sigma}$	*AIC*	*BIC*	SS_{MS}	SS_{1S}
1, 12	1	5	2.38	0.147	−431	−412	1.73	0.96
1, 12	2	9	2.15	0.143	−435	−401	1.33	0.99
1, 12	4	17	1.81	0.137	−438	−374	3.27	3.30
1, 12, 13	1	6	1.09	0.101	−512	−490	2.59	0.66
1, 12, 13	2	11	0.90	0.094	−524	−483	2.05	0.60
1, 12, 13	4	21	0.69	0.086	−534	−456	4.93	2.02
1, 2, 12, 13	1	7	1.08	0.101	−511	−485	2.98	0.67
1, 2, 12, 13	2	13	0.91	0.095	−519	−471	18.10	1.48
1, 2, 12, 13	4	25	0.64	0.086	−534	−441	5.24	4.18
Box–Jenkins model		2	1.05	0.097	−525	−519	0.61	0.47

†The notation is defined in Section 2 and all numbers refer to the scaled data.

7. Discussion

We carried out a second analysis using the sales data from Chatfield and Prothero (1973) and the results are reported in Chatfield and Faraway (1996). Broadly speaking, the difficulties in fitting NN models were qualitatively similar, but the NN forecasts were even less persuasive, partly no doubt because that series is even shorter than the airline data.

We realize that analysing two time series gives little insight into the general forecasting ability of NN models, and so we now give a brief, and necessarily selective, review of the voluminous and rapidly growing literature applying NNs to time series. Earlier reviews by Chatfield (1993) and Hill *et al.* (1996), section 3, bear out our empirical experience that NN forecasts need not be clearly better than alternatives. It should also be remembered that researchers are more likely to publish results in favour of a new method, such as NN modelling, than the reverse (Chatfield, 1995b). A recent competition, using real life data, was described by Hill *et al.* (1996), who claimed that NNs 'did significantly better than traditional methods'. This finding contrasts with that of Foster *et al.* (1992) using some of the same data and may be overstated. Another recent competition (Callen *et al.*, 1996), using 296 short ($n = 89$) accounting series, found that linear methods are better than NNs 'even when data are financial, seasonal and non-linear'. The Sante Fe competition (Gershenfeld and Weigend, 1994) used much longer series (several thousand observations) which were more clearly non-linear. There NN models generally did well, though, for the one financial time series (exchange rate data), a 'crucial difference between training and test set performance' was found (Gershenfeld and Weigend (1994), p. 40). Out-of-sample predictions from the fitted NN model were no better than those from a random walk!

Simulations are complementary to the use of real life data and the results of Kuan and White (1994), section 1.4, show that NNs can be successful in handling non-linear data. However, even though NNs can be shown to provide a 'universal function approximator', simulations (e.g. Stern (1996), section 3) suggest that linear models give better forecasts of linear data than NNs do, as would be expected.

An assessment of the literature is complicated by the fact that many studies in the computer science literature, where the vast majority of work has appeared, can be criticized from a statistical point of view because

(a) they fail to compare NNs with alternative statistical methods (e.g. Hoptroff (1993)), or

(b) because the comparisons appear statistically flawed (for example see the discussion in Chatfield (1993)), or
(c) because forecasts may not be genuine out-of-sample forecasts or
(d) not enough detail is given to make a proper assessment of the results.

The earlier case-study of the airline data by Tang *et al.* (1991) is a good example of (d) as few details were given on how the NN models were actually fitted and the reader is left to guess for example that the data have been scaled by dividing by 1000 so that the logistic function can be used at the output stage. Given the stated number of hidden neurons, the number of weights for some of Tang's NN models apparently exceeded the number of observations. The paper is written from the point of view of computer scientists and, for us, raised more questions than it answered.

A better recent example from the connectionist literature is Park *et al.* (1996) which compares NN and Box–Jenkins forecasts for the sunspots data. However, they used a second-order autoregressive (AR(2)) model, as fitted in the 1970 edition of Box *et al.* (1994), even though it is now generally agreed that the 'series is almost certainly not adequately fitted by an AR(2) model' (Box *et al.* (1994), p. 205). Thus the greater accuracy of NN forecasts may be overstated. A subset AR(9) model (e.g. Morris (1977)) or a non-linear time series model, such as a threshold model (Tong, 1990), should have been used for comparison as in de Groot and Würtz (1991).

In reply, a computer scientist might criticize us for what we have done, for example by not considering elaborations such as skip layer connections, extra hidden layers and feed-back connections, but we think that the price of extra versatility is to increase the potential for going astray. With such short series, we think that there are already more than enough choices to be made.

The above remarks raise the issue of how statistics and NNs should interact. Sarle (1994) rightly said that they 'are not competing methodologies' and there should be better communication between the two fields. Flexer (1995), who is himself a connectionist, went further in lamenting the rivalry that often seems to exist and calling for more co-operation. In particular Flexer argued that the 'quality of research in the area of connectionism needs improvement' and that NNs are not compared with statistical methods as often as they should be. We naturally support this.

Although unable to make definitive statements about the forecasting accuracy of NN models, our experience does allow us to comment on the general difficulties involved in NN modelling. We agree with Sarle (1994) that the

> 'marketing hype claims that neural networks can be used with no experience and automatically learn whatever is required . . . is nonsense'.

We also agree with Gershenfeld and Weigend (1994), p. 59, that 'there are unprecedented opportunities to go astray'. Great care is needed to choose

(a) an appropriate set of input variables,
(b) an appropriate architecture,
(c) appropriate activation functions (which need not be the same at the hidden layers as at the output unit(s)) and
(d) an appropriate numerical procedure for fitting an NN model.

In particular sensible starting values need to be chosen for the weights and it may be necessary to scale the data. As statisticians, we also find NN models less revealing than alternative time

series models. The complicated black box structure of NN models is hard to understand and interpret, and, as there is no description of the 'error' term, there is no obvious way to compute interval forecasts.

In conclusion this case-study suggests the following general points.

(a) There is plenty of scope for going badly wrong with NN modelling (as there is for many other sophisticated statistical techniques). Without careful choices of the architecture, the activation functions and appropriate starting values for the weights, fitting routines may not converge, may converge to a local minimum or may lead to forecasts which are not sensible.
(b) Adding extra hidden units increases the number of parameters in an NN model. This may lead to an improvement in fit but may lead to a deterioration in out-of-sample predictions.
(c) When comparing models with different numbers of parameters, the use of the ordinary AIC does not do enough to penalize the addition of extra parameters, and the BIC is recommended instead. An alternative possibility is the bias-corrected version of the AIC (AIC_C).
(d) NN models are hard to interpret. GAMs and PPR provide alternative nonparametric approaches to the identification of non-linear time series which require less structure to be imposed on the data, are more familiar to statisticians and are arguably more helpful in exploring a set of data to understand the nature of the response surface.

Acknowledgement

This work was carried out while the first author was visiting the University of Bath.

Appendix A

Software for fitting neural nets to time series data may be found on the World Wide Web at

`http://www.stat.lsa.umich.edu/~faraway/`

together with full details on installation and use. We used various S-PLUS functions from Venables and Ripley (1994) (including `nnet` and `nnet.Hess`) in conjunction with some functions written by the first author. Some knowledge of S-PLUS is required for the software to be useful.

Here is an example of its use. The airline data are read and rescaled by dividing by 100 in the first command line. An NN(1, 12; 2) model is fitted to the first 132 observations in the second line, including an instruction to restart the algorithm 50 times from different random starting weights and to take the best of the models found. A summary of the fit is requested in the third line.

```
> air <- scan ("air.data")/100
> g <- nnts(air[1:132],c(1,12),2,retry=50)
> summary(g)
a 2-2-1 network with 9 weights

Unit 0 is constant one input
Input units: Lag 1=1, Lag 12=2,
Hidden units are 3 4
Output unit is 5

 0->3   1->3   2->3    0->4   1->4    2->4   0->5     3->5    4->5
-0.10   1.26  -1.31   -0.10   0.66   -0.55  -7.43   -15.60   31.27
Sum of squares is 2.310301
AIC : -456.0137 , BIC : -421.9262 , residual se : 0.1442689
```

We can now predict the next 12 observations.

```
> predict (g,12)
3.997191 3.803972 4.383948 4.370344 4.652647 5.185713 5.889187 6.055508
4.930448 4.410727 3.994916 4.466562
```

An alternative public domain software package for NN modelling is available from the University of Nevada via the World Wide Web at

`http://www.scs.unr.edu/~cbmr/research/local/res-local.html`

References

Azoff, E. M. (1994) *Neural Network Time Series Forecasting of Financial Markets*. New York: Wiley.

Bishop, C. M. (1995) *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press.

Box, G. E. P., Jenkins, G. M. and Reinsel, G. C. (1994) *Time Series Analysis, Forecasting and Control*, 3rd edn. Englewood Cliffs: Prentice Hall.

Brockwell, R. J. and Davis, R. A. (1991) *Time Series: Theory and Methods*, 2nd edn. New York: Springer.

Brown, R. G. (1962) *Smoothing, Forecasting and Prediction of Discrete Time Series*. Englewood Cliffs: Prentice Hall.

Callen, J. L., Kwan, C. C. Y., Yip, P. C. Y. and Yuan, Y. (1996) Neural network forecasting of quarterly accounting earnings. *Int. J. Forecast.*, **12**, 475–482.

Chambers, J. and Hastie, T. (1992) *Statistical Models in S*. Pacific Grove: Wadsworth and Brooks/Cole.

Chatfield, C. (1993) Neural networks: forecasting breakthrough or passing fad? *Int J. Forecast.*, **9**, 1–3.

———(1995a) Model uncertainty, data mining and statistical inference (with discussion). *J. R. Statist. Soc.* A, **158**, 419–466.

———(1995b) Positive or negative? *Int J. Forecast.*, **11**, 501–502.

———(1996a) *Time Series Analysis*, 5th edn. London: Chapman and Hall.

———(1996b) Model uncertainty and forecast accuracy. *J. Forecast.*, **15**, 495–508.

Chatfield, C. and Faraway, J. (1996) Forecasting sales data with neural nets: a case study (in French). *Rec. Applic. Marktng*, **11**, no. 2, 29–41.

Chatfield, C. and Prothero, D. L. (1973) Box–Jenkins seasonal forecasting: problems in a case-study (with discussion). *J. R. Statist. Soc.* A, **136**, 295–336.

Chatfield, C. and Yar, M. (1988) Holt-Winters forecasting: some practical issues. *Statistician*, **37**, 129–140.

Cheng, B. and Titterington, M. (1994) Neural networks: a review from a statistical perspective (with discussion). *Statist. Sci.*, **9**, 2–54.

Fletcher, R. (1987) *Practical Methods of Optimization*, 2nd edn. Chichester: Wiley.

Flexer, A. (1995) Connectionists and statisticians: friends or foes? In *From Natural to Artifical Neural Computation: Proc. Int. Wrkshp Artificial Neural Networks* (eds J. Mira and F. Sandoval), pp. 454–461. Torremolinos: Springer.

Foster, W. R., Collopy, F. and Ungar, L. H. (1992) Neural network forecasting of short, noisy time series. *Comput. Chem. Engng*, **16**, 293–297.

Friedman, J. (1984) A variable span smoother. *Technical Report 5*. Laboratory for Computational Statistics, Stanford University, Stanford.

Gershenfeld, N. A. and Weigend, A. S. (1994) The future of time series: learning and understanding. In *Time Series Prediction* (eds A. S. Weigend and N. A. Gershenfeld), pp. 1–70. Reading: Addison-Wesley.

de Groot, C. and Würtz, D. (1991) Analysis of univariate time series with connectionist nets: a case study of two classical examples. *Neurocomputing*, **3**, 177–192.

Harvey, A. (1993) *Time Series Models*, 2nd edn. Hemel Hempstead: Harvester Wheatsheaf.

Harvey, A. and Toulson, S. (1994) Review of 4Thought. *Int. J. Forecast.*, **10**, 35–46.

Hastie, T. and Tibshirani, R. (1990) *Generalized Additive Models*. London: Chapman and Hall.

Hertz, J., Krogh, A. and Palmer, R. (1991) *Introduction to the Theory of Neural Computation*. Redwood City: Addison-Wesley.

Hill, T., O'Connor, M. and Remus, W. (1996) Neural network models for time series forecasts. *Mangmnt Sci.*, **42**, 1082–1092.

Hoptroff, R. (1993) The principles and practice of time series forecasting and business modelling using neural nets. *Neur. Comput. Applic.*, **1**, 59–66.

James, H. (1994) Software for studying and developing applications of artificial neural networks. *Econ. J.*, **104**, 181–196.

Jones, M. C. and Sibson, R. (1987) What is projection pursuit (with discussion)? *J. R. Statist. Soc.* A, **150**, 1–36.

Kuan, C.-M. and White, H. (1994) Artificial neural networks: an econometric perspective (with discussion). *Econometr. Rev.*, **13**, 1–143.

van Laarhoven, P. J. M. and Aarts, E. H. L. (1987) *Simulated Annealing: Theory and Applications*. Dordrecht: Reidel.

Morris, M. J. (1977) Forecasting the sunspot cycle. *J. R. Statist. Soc.* A, **140**, 437–448.

Park, Y. R., Murray, T. J. and Chen, C. (1996) Predicting sun spots using a layered perceptron neural network. *IEEE Trans. Neur. Netwrks*, **7**, 501–505.
Priestley, M. B. (1981) *Spectral Analysis and Time Series*. London: Academic Press.
Ripley, B. (1993) Statistical aspects of neural networks. In *Chaos and Networks — Statistical and Probabilistic Aspects* (eds O. Barndorff-Nielsen, J. Jensen and W. Kendall), pp. 40–123. London: Chapman and Hall.
———(1996) *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press.
Sarle, W. (1994) Neural networks and statistical models. In *Proc. 19th A. SAS Users Group Int. Conf.*, pp. 1538–1550. Cary: SAS Institute.
Stern, H. (1996) Neural networks in applied statistics (with discussion). *Technometrics*, **38**, 205–220.
Tang, Z., de Almeida, C. and Fishwick, P. A. (1991) Time series forecasting using neural networks versus Box–Jenkins methodology. *Simulation*, **57**, 303–310.
Tjøstheim, D. and Auestad, B. (1994) Nonparametric identification of nonlinear time series: I, Projections; II, Selecting significant lags. *J. Am. Statist. Ass.*, **89**, 1398–1419.
Tong, H. (1990) *Non-linear Time Series*. Oxford: Clarendon.
Venables, W. N. and Ripley, B. D. (1994) *Modern Applied Statistics with S-PLUS*. New York: Springer.
Warner, B. and Misra, M. (1996) Understanding neural networks as statistical tools. *Am. Statistn*, **50**, 284–293.