Feed-forward Neural Nets as Models for Time Series Forecasting

Zaiyong Tang
BUS 351 Dept. of Decision & Information Sciences
University of Florida, Gainesville, FL 32611
Phone: 904-392-9600 *Email: zt@beach.cis.ufl.edu*

Paul A. Fishwick
301 CSE, Department of Computer & Information Sciences
University of Florida, Gainesville, FL 32611
Phone: 904-392-1414 *Email:fishwick@cis.ufl.edu*

Computer Science: Feed-forward neural networks, application

Forecasting: Time series

Abstract

We have studied neural networks as models for time series forecasting, and our research compares the Box-Jenkins method against the neural network method for long and short term memory series. Our work was inspired by previously published works that yielded inconsistent results about comparative performance. We have since experimented with 16 time series of differing complexity using neural networks. The performance of the neural networks is compared with that of the Box-Jenkins method. Our experiments indicate that for time series with long memory, both methods produced comparable results. However, for series with short memory, neural networks outperformed the Box-Jenkins model. Because neural networks can be easily built for multiple-step-ahead forecasting, they present a better long term forecast model than the Box-Jenkins method. We discussed the representation ability, the model building process and the applicability of the neural net approach. Neural networks appear to provide a promising alternative for time series forecasting.

Introduction

A time series is a sequence of time-ordered data values that are measurements of some physical process. For example, one can consider time-dependent sales volume or rain fall data to be examples of time series. Time series forecasting has an especially high utility for predicting economic and business trends. Many forecasting methods have been developed in the last few decades (Makridakis 1982). The Box-Jenkins method is one of the most widely used time series forecasting methods in practice (Box and Jenkins 1970).

Recently, artificial neural networks that serve as powerful computational frameworks have gained much popularity in business applications as well as in computer science, psychology and cognitive science. Neural nets have been successfully applied to loan evaluation, signature recognition, time series forecasting (Dutta and Shekhar 1988; Sharda and Patil 1990), classification analysis (Fisher and McKusick 1989; Singleton and Surkan 1990), and many other difficult pattern recognition problems (Simpson 1990). While it is often difficult or impossible to explicitly write down a set of rules for such pattern recognition problems, a neural network can be trained with raw data to produce a solution.

Concerning the application of neural nets to time series forecasting, there have been mixed reviews. For instance, Lapedes and Farber (1987) reported that simple neural networks can outperform conventional methods, sometimes by orders of magnitude. Their conclusions are based on two specific time series without noise. Weigend et al. (1990) applied feed-forward neural nets to forecasting with noisy real-world data from sun spots and computational ecosystems. A neural network, trained with past data, generated accurate future predictions and consistently out-performed traditional statistical methods such as the TAR (threshold autoregressive) model (Tong and Lim 1980). Sharda and Patil (1990) conducted a forecasting competition between neural network models and a traditional forecasting technique (namely the Box-Jenkins method) using 75 time series of various nature. They concluded that simple neural nets could forecast about as well as the Box-Jenkins forecasting system. Fishwick (1989) demonstrated that, for a ballistics trajectory function approximation problem, the neural network used offered little

competition to the traditional linear regression and surface response model.[1]

The potential of neural nets as models for time series forecasting has not been studied systematically. Sharda and Patil (1990) reported that the performance of neural nets and the Box-Jenkins method are on par. Out of the 75 series they tested, neural nets performed better than the Box-Jenkins method for 39 series, and worse for the other 36 series. One would ask when neural nets are better than the other method? Can the performance of neural nets be improved?

Neural net models are generally regarded as "black boxes." Few researchers have explored in detail neural nets as times series forecasting models. We would like to ask why neural nets can be used as forecasting models, and why should they be able to compete with conventional methods such as the Box-Jenkins method.

We will try to answer the questions raised above by performing a series of forecasting experiments and analysis. Different neural net structures and training parameters will be used. The performance of neural nets is compared with that of the Box-Jenkins method. In the following section, we give a brief review of the two approaches. The subsequent section presents the forecasting experiment results. We discuss the issues in applying neural nets in forecasting in Section 3. The last section presents a summary of the study and our conclusions.

1 Methodology Background

While the Box-Jenkins method is a fairly standard time series forecast method, neural nets as forecast models are relatively new. Thus, in the following, we present only a brief account of the Box-Jenkins method and give a more detailed description of the neural net approach. Complete treatments of the two methods can be found in (Hoff 1983) and (Rumelhart, McClelland and the PDP Research Group 1986).

[1] We later determined that the neural network approach (for the ballistics data) performed quite adequately as long as the initial data were randomized with respect to training order.

1.1 The Box-Jenkins method

The Box-Jenkins method is one of the most popular time series forecasting methods in business and economics. The method uses a systematic procedure to select an appropriate model from a rich family of models, namely, ARIMA models. Here AR stands for autoregressive and MA for moving average. AR and MA are themselves time series models. The former model yields a series value x_t as a linear combination of some finite past series values, $x_{t-1}, ..., x_{t-p}$, where p is an integer, plus some random error e_t. The latter gives a series value x_t as a linear combination of some finite past random errors, $e_{t-1}, ..., e_{t-q}$, where q is an integer. p and q are referred as orders of the models. Note that in the AR and MA models some of the coefficients may be zero, meaning that some of the lower-order terms may be dropped. AR and MA models can be combined to form an ARMA model. Most time series are "non-stationary" meaning that the level or variance of the series change over time. Differencing is often used to remove the trend component of such time series before a ARMA model can be used to describe the series. The ARMA models applied to the differenced series are called integrated models, denoted by ARIMA.

A general ARIMA model has the following form (Bowerman and O'connell 1987):

$$\Phi_p(B)\Phi_P(B^L)(1-B^L)^D(1-B)^d y_t = a + \Theta_q(B)\Theta_Q(B^L)\epsilon_t \tag{1}$$

where $\Phi(B)$ and $\Theta(B)$ are autoregressive and moving average operators respectively; B is the back shift operator; ϵ_t is called random error with normal distribution $N(0, \sigma^2)$; a is a constant, and y_t is the time series data, transformed if necessary.

The Box-Jenkins method performs forecasting through the following process:

1. *Model Identification:* The orders of the model are determined.

2. *Model Estimation:* The linear coefficients of the model are estimated (based on maximum likelihood).

3. *Model Validation:* Certain diagnostic methods are used to test the suitability of the estimated model. Alternative models may be considered.

4. *Forecasting:* The best model chosen is used for forecasting. The quality of the forecast is monitored. The parameters of the model may be reestimated or the whole modeling process repeated if the quality of forecast deteriorates.

1.2 Neural nets and backpropagation

Neural nets are a computational framework consisting of massively connected simple processing units. These units have an analog to the neurons in the human brain. Because of the highly connected structure, neural networks exhibit some desirable features, such as high speed via parallel computing, resistance to hardware failure, robustness in handling different types of data, graceful degradation (which is the property of being able to process noisy or incomplete information), learning and adaptation (Rumelhart, McClelland and the PDP Research Group 1986; Lippmann 1987; Hinton 1989).

One of the most popular neural net paradigms is the feed-forward neural network (FNN) and the associated back-propagation (BP) training algorithm. In a feed-forward neural network, the neurons (i.e., processing units) are usually arranged in layers. A feed-forward neural net is denoted as $I \times H \times O$, where I, H and O represent the number of input units, the number of hidden units, and the number of output units, respectively. Figure 1 gives a typical fully connected 2-layer feed-forward network (by convention, the input layer does not count) with a $3 \times 4 \times 3$ structure.

The input units simply pass on the input vector x. The units in the hidden layer and output layer are processing units. Each processing unit has an activation function which is commonly chosen to be the sigmoid function.

$$f(x) = \frac{1}{1 + e^{-\gamma x}} \tag{2}$$

where γ is a constant controlling the slope of the function. The net input to a processing unit j is given by

$$net_j = \sum_i w_{ij} x_i + \theta_j \tag{3}$$

where x_i's are the outputs from the previous layer, w_{ij} is the weight (connection strength) of the link connecting unit i to unit j, and θ_j the bias, which determines the location of the sigmoid

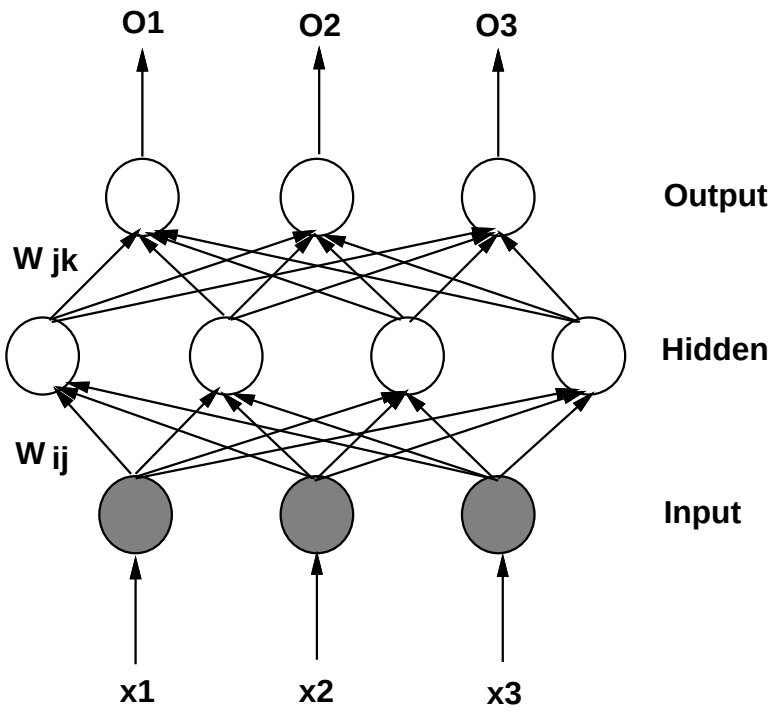

Figure 1: A $3 \times 4 \times 3$ feedforward neural network.

function on the x axis.

A feed-forward neural net works by training the network with known examples. A random sample (x_p, y_p) is drawn from the training set $\{(x_p, y_p)|p = 1, 2, \ldots, P\}$, and x_p is fed into the network through the input layer. The network computes an output vector o_p based on the hidden layer output. o_p is compared against the training target y_p. A performance criterion function is defined based on the difference between o_p and y_p. A commonly used criterion function is the sum of squared error (SSE) function

$$F = \sum_p F_p = \frac{1}{2} \sum_p \sum_k (y_{pk} - o_{pk})^2 \tag{4}$$

where p is the index for the pattern (example) and k the index for output units.

The error computed from the output layer is backpropagated through the network, and weights (w_{ij}) are modified according to their contribution to the error function.

$$\Delta w_{ij} = -\eta \frac{\partial F}{\partial w_{ij}} \tag{5}$$

where η is called learning rate, which determines the step size of the weight updating. The BP

algorithm is summarized below (for derivation details, see Rumelhart et al. 1986b).

Algorithm BP

1. *Initialize:*

   - Construct the feedforward neural network. Choose the number of input units and the number of output units equal to the length of input vector x and the length of target vector t, respectively.
   - Randomize the weights and bias in the range $[-.5, .5]$.
   - Specify a stopping criterion such as $F \leq F_{stop}$ or $n \geq n_{max}$. Set iteration number $n = 0$.

2. *Feedforward:*

   - Compute the output for the non-input units. The network output for a given example p is

   $$o_{pk} = f(\sum_j w_{jk} f(\sum_m w_{mj} f(\cdots f(\sum_i w_{il} x_i)))).$$

   - Compute the error using Equation 4.
   - If a stopping criterion is met, stop.

3. *Backpropagate:*

   - $n \leftarrow n + 1$.
   - For each output unit j, compute

   $$\delta_k = (o_k - y_k) f'(net_k).$$

   - For each hidden unit j, compute

   $$\delta_j = f'(net_j) \sum_k \delta_k w_{jk}.$$

4. *Update:*

$$\Delta w_{ij}(n+1) = \eta \delta_j o_i + \alpha \Delta w_{ij}(n)$$

where $\eta > 0$ is the learning rate (step size) and $\alpha \in [0, 1)$ is a constant called the momentum.

5. *Repeat:*

Go to Step 2.

The momentum term is used to accelerate the weight updating process when the error gradient is small, and to damper oscillation when the error gradient changes sign in consecutive iterations. A discussion of the convergence and the performance of BP is beyond the scope of this paper. Interested readers are referred to White (1989) and Hecht-Nielsen (1989).

2 Forecast Experiments

2.1 Data

Sharda and Patil (1990) used 75 series taken from the well known M-Competition (Makridakis 1982) series that are suitable for the Box-Jenkins analysis. Their results showed that the relative performance of the neural nets and the Box-Jenkins models vary substantially for different series. To explore why there were such differences, we took 14 series from the 75 time series used in their study. For 10 (group 1) of those 14 series, neural nets were reported to have performed significantly worse than the Box-Jenkins method in term of forecast MAPS (mean absolute percentage errors). For the other 4 series (group2), both methods gave large errors. The two groups of data are shown in figures 2 and 3, respectively. The data series are labeled as they appear in Sharda and Patil (1990). Series numbered less than 382 are quarterly and those numbered greater than 382 are monthly series.

Although *AUTOBOX*, the computer expert system used in Sharda and Patil (1990), was reported to be comparable to real experts (Sharda and Ireland 1987), we feel that AUTOBOX might not give the best results derived from the Box-Jenkins method. To obtain a more fair comparison, two more standard test series are also used. The first one is the international airline passenger data from 1949 to 1960. This series has been used in the classic work by Box and

Jenkins (1970). The other series is company sales data taken from *TIMESLAB* (Newton 1988) (Figure 4).

2.2 Experimental Design

Following convention, the last 8 terms of the quarterly series and the last 18 terms of the monthly series were used as hold-out samples to test the forecasting performance of the models. Different neural net structures and training parameters were tested. A forecast problem can be easily mapped to a feed-forward neural net. The number of input units corresponds to the number of input data terms. In the univariate time series case, this number is also the number of AR terms used in the Box-Jenkins model. The number of output units represents the forecast horizon. One-step-ahead forecast can be performed by a neural net with one output unit, and k-step-ahead forecast can be mapped to a neural net with k output units. A neural net forecast model is shown in figure 5, where 8 terms of past series values are used to predict the series values in 2 steps ahead. The effect of hidden units on forecasting performance will be discussed in the next section.

For each neural net structure and parameter setting, 10 experiments were run with different random initial weights. The reported forecasting errors (MAPS) are the averages of the 10 runs. The data series are normalized to the range of $[0.2, 0.8]$ before feeding into the neural nets. This normalization range was found to facilitate neural net training better than the range of $[0, 1]$. Forecasts from the neural net output were transformed to the original data scale before the MAPS were computed.

For the 14 series taken from the M-Competition, the forecasting results reported by Sharda and Patil were used to compare with our neural net models. The other two test series were used to examine the performance of the Box-Jenkins method and the neural net approach for forecasting with different forecast horizons. Box-Jenkins model forecasting was carried out with the time series analysis package *TIMESLAB* (Newton 1988).

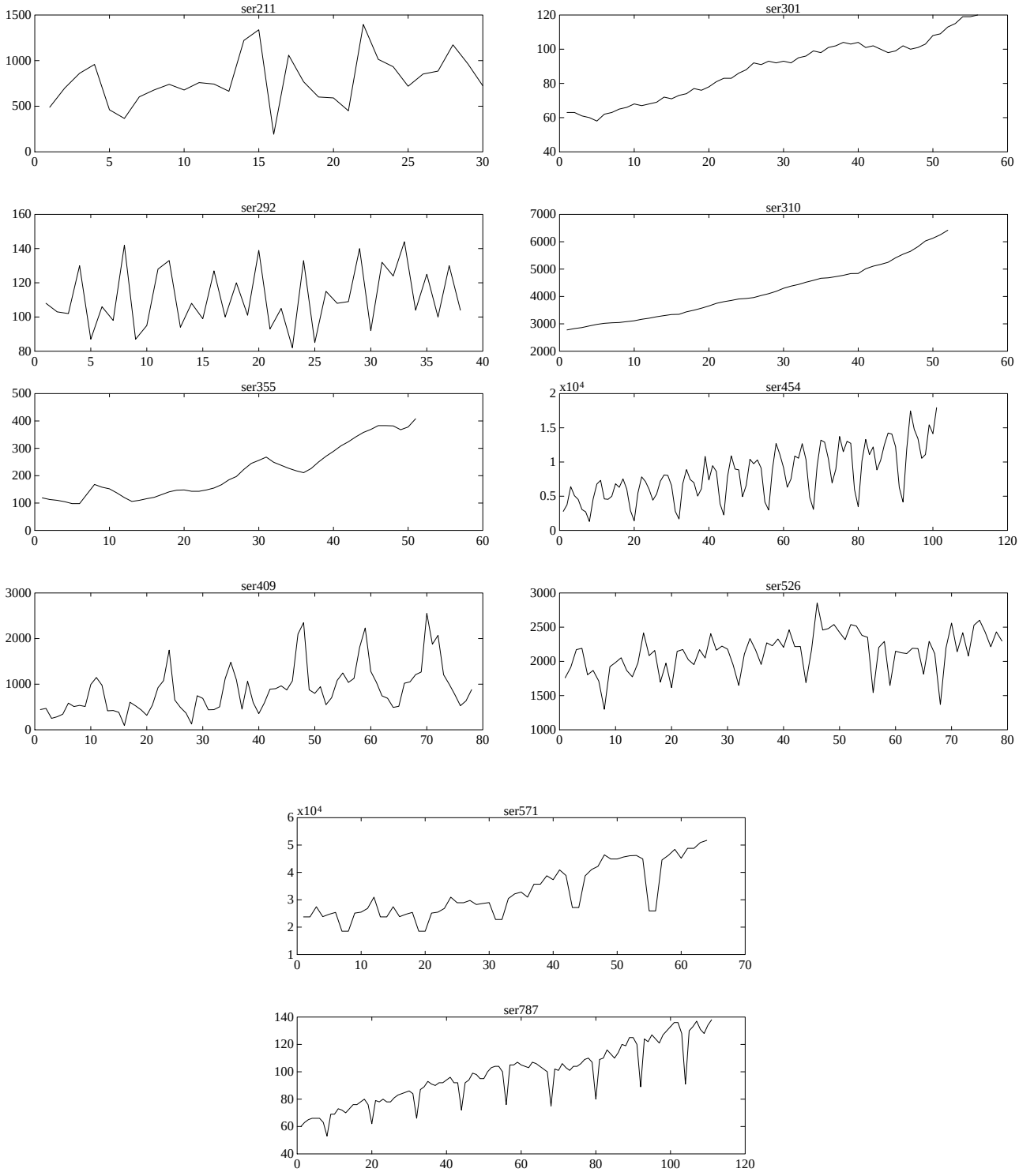

Figure 2: Time series plots for series group 1.

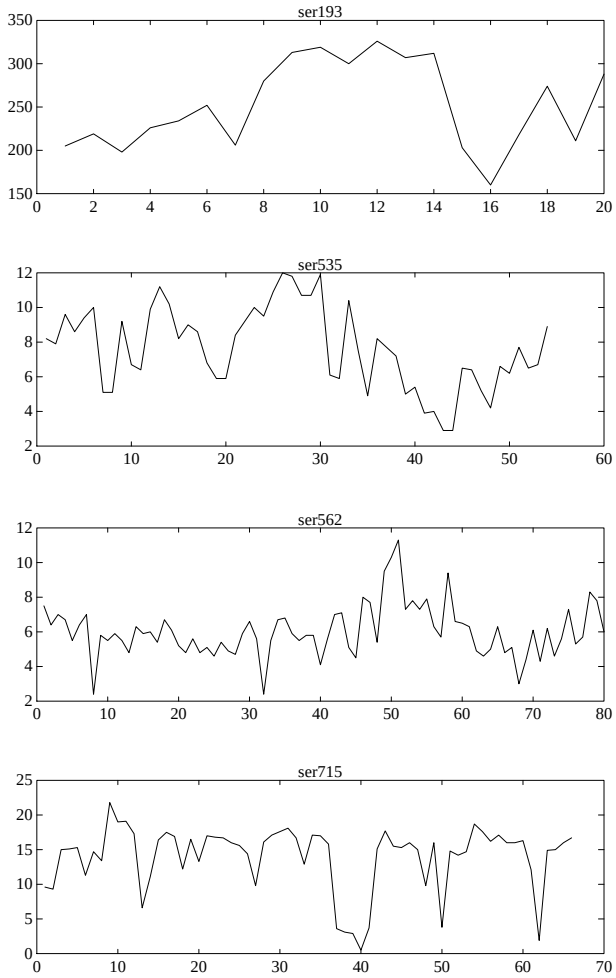

Figure 3: Time series plots for series group 2.

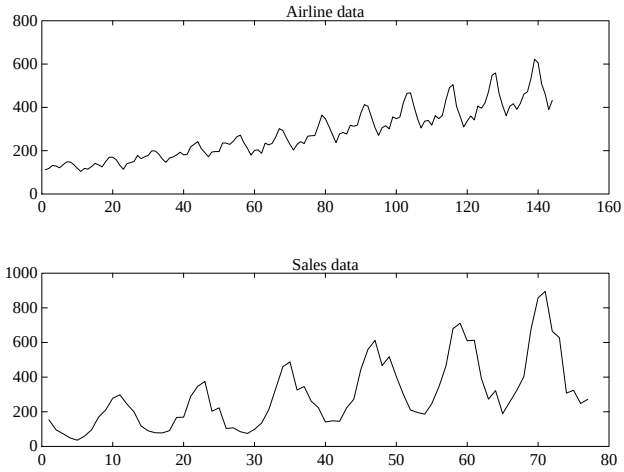

Figure 4: Airline passenger and sales data.

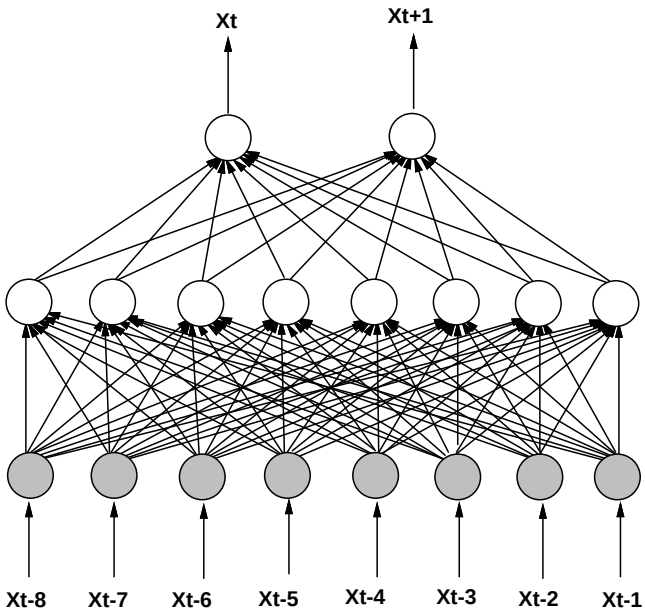

Figure 5: A neural net model for 2-step-ahead forecasting.

2.3 Forecasting results

We first tested the performance of general neural net models that are used by most other researchers. They are 2-layer feed-forward neural nets with the number of hidden units equal the number of input units. Taking the time series data structure into account, we used one year's data as an input pattern. Hence there are 4 input units for quarterly series, and 12 input units for monthly series. Three training parameter pairs were used. For each parameter set, the neural net was trained for 100 epochs and 500 epochs. (An epoch is one presentation of all the training examples).

Table-1: Forecast error (MAPS) for series group 1.

Series	100 Epochs			500 Epochs			Reported
	$\eta = \alpha = .5$	$\eta = \alpha = .25$	$\eta = \alpha = .1$	$\eta = \alpha = .5$	$\eta = \alpha = .25$	$\eta = \alpha = .1$	
ser211	17.03	17.16	**16.42**	22.45	18.48	17.64	(26.31,50.36)
ser292	**10.53**	11.21	12.18	18.07	12.14	10.81	(12.68,24.86)
ser301	8.22	9.53	15.21	8.47	**6.89**	7.36	(2.91,10.77)
ser310	7.08	14.43	25.17	**5.46**	6.39	10.26	(4.46,11.30)
ser355	4.48	19.86	40.23	**3.15**	4.84	9.76	(4.75,9.75)
ser409	**17.22**	51.42	44.13	18.68	19.71	35.60	(42.80,97.43)
ser454	29.66	28.06	31.43	18.21	**11.74**	17.48	(8.48,16.17)
ser526	17.31	11.29	10.04	14.35	10.67	**9.81**	(9.95,20,28)
ser571	24.07	24.72	29.54	**6.78**	15.12	26.84	(5.57,10.39)
ser787	13.83	11.48	14.52	**2.24**	3.19	10.04	(3.17,7.90)

Table 1 shows the forecast errors for the 10 series (group 1) that neural nets were reported to have performed significantly worse than the Box-Jenkins method. The number in the parenthesis are the MAPS of the Box-Jenkins method (first one) and the MAPS of the neural nets (second one) obtained by Sharda and Patil (1990). Table 2 shows the forecast errors for the 4 series (group 2) for which both the Box-Jenkins method and the neural net method were reported as performing poorly.

Table-2: Forecast error (MAPS) for series group 2.

Series	100 Epochs			500 Epochs			Reported
	$\eta = \alpha = .5$	$\eta = \alpha = .25$	$\eta = \alpha = .1$	$\eta = \alpha = .5$	$\eta = \alpha = .25$	$\eta = \alpha = .1$	
ser193	**25.14**	25.49	24.53	31.66	26.05	25.37	(44.43,40.79)
ser535	**35.31**	54.29	65.15	120.17	83.64	75.08	(69.14,48.26)
ser562	33.41	30.82	32.29	69.81	38.37	**29.24**	(30.63,34.09)
ser715	64.11	59.89	62.38	72.16	**55.06**	61.32	(61.31,76.65)

Results in table 1 show that our neural net models, in appropriate training setting, performed significantly better than those reported in Sharda and Patil (1990). The best results in tables 1 and 2 are highlighted with boldface. Note that the important role played by the training parameters η and α, and the training length. This explains to some extent the discrepancy in the results reported in the literature. Choosing a single training setting for different time series is –at best– unreliable. Two events may lead to poor forecasting performance. One is training failure, and the other is overfitting. These events will be discussed in the next section.

The improvement of the forecasting performance of the series group 2 is less significant than that of the group 1. Note that, in this group, longer training length leads to deteriorated forecasts in virtually all cases. The reason is that all the series in group 2 are highly irregular and subject to pattern shifting in the forecasting period (see Figure 3). For example, the forecast for `ser715` is poor because the training pattern given to the network has a cycle length that is different from the pattern to be forecasted.

2.4 The effect of forecast horizon

There are two ways to conduct a long term foreacasting. The first is to directly build an multiple-step-ahead forecast model. This can be easily implemented with neural nets by simply adding

more output units. The second approach is to do a stepwise forecast through using a model built for shorter forecast horizon. That is, the forecast values are fed back as inputs (which may be partial) to forecast the values in the next forecast horizon of the model. Using the Box-Jenkins method for multiple-step-ahead forecasting falls into the second approach, where the models have a forecast horizon of one step.

Table-3: Forecast error (MAPS) for different forecasting horizons.

Net structure and forecast horizon	Airline data		Sales data	
	Box-Jenkins	Neural nets	Box-Jenkins	Neural nets
$12 \times 12 \times 1$	3.11	4.19	16.69	16.36
$12 \times 12 \times 6$	5.02	5.42	19.43	17.68
$24 \times 24 \times 12$	7.30	6.87	16.87	19.87
$24 \times 24 \times 24$	14.72	5.31	26.96	31.05

Table 3 shows that the performance of neural net models for multiple-step-ahead forecasting are different for the two series. For the airline data, with one-step-ahead and six-step-ahead forecasts, the Box-Jenkins model outperforms the neural network for the selected structures and training methods, while for the 12-step-ahead and 24-step-ahead forecasts, the neural network is better. For the sales data, the neural nets seems worse for long term forecasting. This poor performance results from the fact that for the $24 \times 24 \times 24$ network with 24 periods of data hold for testing, there are only 6 training patterns (the sales series has a total of 77 data). This 6-pattern training set is not enough to represent the yearly cycle of the series.

We tested stepwise forecasting with 3 neural net models, having a forecast horizon equal to 1, 6, and 12 steps, respectively. Table 4 shows that when stepwise forecasting is used for 24-step-ahead forecast, the neural nets models outperformed the Box-Jenkins method in all cases.

Table-4: 24-step-ahead forecast error (MAPS) with stepwise forecasting.

Data	Neural nets			Box-Jenkins
	$12 \times 12 \times 1$	$12 \times 12 \times 6$	$12 \times 12 \times 12$	
airline	14.49	8.52	7.19	14.74
sales	22.37	24.91	17.46	26.96

It is not surprising that as the forecast horizon extends, the Box-Jenkins model performs less well, since the model is best suited for short term forecasting. The relative performance of the neural network improves as the forecast horizon increases. This suggests that the neural network

is a better choice for long term forecasting. Note that for the 24-step-ahead forecast, the neural network resulted in much a smaller error then the Box-Jenkins model (except for the case where training set is insufficient). The success of the 24-step-ahead forecast may be surprising. The reason may be that for the NN model, it is actually easier to learn the underlying mechanism of the time series with similar input and output patterns.

2.5 The effect of hidden units

Tests in this and the next subsection were done with training parameters set to 0.25 and 0.1. The training length was set to 500 epoches. The neural nets are fully connected with structures shown in the tables.

Table-5: Forecast error (MAPS) with different numbers of hidden units.

Net structure	Airline data	Sales data	ser454	ser562
$12 \times 12 \times 1$	4.19	16.36	11.74	29.24
$12 \times 6 \times 1$	4.46	16.21	9.77	26.10
$12 \times 3 \times 1$	4.61	17.76	9.75	20.65
$12 \times 2 \times 1$	4.24	17.20	9.14	21.12
$12 \times 1 \times 1$	3.64	18.85	15.40	24.78

The effect of number of hidden units on neural net model performance is not the same for series with different nature. Table 5 shows that the effect on airline and the sales data is not significant. For the other two series, there is a trend of forecast error. Up to a certain point, the error decreases until it reaches a lower value, and then the error increases. This phenomenum suggests that there is an optimal number of hidden units for different series. This is understandable as too few hidden units make the net not able to learn the underlying pattern of the training set, while too many hidden units tend to overfit the training set. We will discuss how to determine the optimal number of hidden units in the next section.

2.6 The effect of direct connection

A feed-forward neural net can be modified to have direct connections from the input units to the output units (figure 6). Theoretically, direct connections increase the recognition power of a feed-forward neural net (Sontag 1990). The insertion of direction connections is equivalent to adding linear components to the net (see the discussion in 3.1).

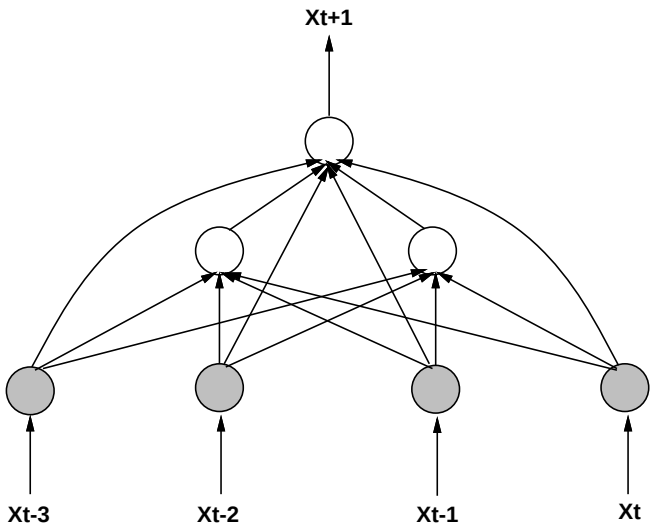

Figure 6: A neural net model with direct connections.

Table-6: Forecast error (MAPS) of neural nets with direct connections.

Net structure	Airline data	Sales data	ser454	ser562
$12 \times 12 \times 1$	4.96	19.47	12.49	22.47
$12 \times 6 \times 1$	4.89	20.16	10.92	22.38
$12 \times 3 \times 1$	3.86	20.24	10.37	19.40
$12 \times 2 \times 1$	3.51	19.86	10.18	19.54
$12 \times 1 \times 1$	2.87	18.38	10.35	19.27
12×1	3.04	17.87	10.15	19.31

Our test using direct connections from input units to output units had mixed results. For the airline data and `ser562`, neural nets with direct connections performed better. For the other two series, the direct connection method was not as successful, except for the one-hidden-unit neural net. However, we feel a general conclusion cannot be drawn as we used only fixed training parameters. Other combinations of training parameters may result in better forecasts.

Note that with a direct connection, the $12 \times 1 \times 1$ neural net produced a MAPS less than that of the Box-Jenkins model. There are several cases in table 1 where the neural net models have a larger forecast error than the Box-Jenkins model. We were be able to test different net structures and training settings and find neural net models that outperformed the Box-Jenkins method for virtually all series. For example, a $1 \times 2 \times 1$ net with direct connection resulted in a MAPS of 2.55 for `ser301`, and a $8 \times 4 \times 1$ net with direct connection resulted a MAPS of 3.24 for `ser310` (cf. table 1).

3 Discussion

The experiments in the last section have shown that neural nets can, indeed, provide better forecasts than the Box-Jenkins method in many cases. However, the performance of neural nets is affected by many factors, including the network structure, the training parameters and the nature of the data series. To further understand how and why neural nets may be used as models for time series forecasting, we examine the representation ability, the model building process and the applicability of the neural net approach in comparison with the Box-Jenkins method.

3.1 Representation

As discussed in the review section, Box-Jenkins models are a family of linear models of autoregressive and moving average processes. For the airline passenger data, the Box-Jenkins model that we identified (the same as identified by other researchers, e.g., Newton (1988)) takes the following form:

$$(1 - B^{12})(1 - B)x_t = (1 - \theta_1 B)(1 - \theta_{1,12} B^{12})\epsilon_t \tag{6}$$

Rewriting the model, we have the following:

$$(1 - B^1 - B^{12} + B^{13})x_t = (1 - \theta_1 B - \theta_{1,12} B^{12} + \theta_1 \theta_{1,12} B^{13})\epsilon_t \tag{7}$$

or

$$x_t = x_{t-12} + (x_{t-1} - x_{t-13}) + (\epsilon_t - \theta_1 \epsilon_{t-1} - \theta_{1,12}\epsilon_{t-12} + \theta_1 \theta_{1,12}\epsilon_{t-13}) \tag{8}$$

Equation 8 says that the forecast for the time period t is the sum of 1) the value of the time series in the same month of the previous year; 2) a trend component determined by the difference of previous month's value and last year's previous month's value; and 3) the effects of random errors (or residual) of period t, $t-1$, $t-12$ and $t-13$ on the forecast.

If a time series is determined by a linear model as described above, the Box-Jenkins method can do well as long as the pattern does not change.[2] If the series is determined by a nonlinear

[2]For series with pattern shift, more sophisticated procedures are needed to identify the shift and modify the model (Lee and Chen 1990).

process, for instance, the logistic series generated by $x(t+1) = ax(t)(1-x(t))$. The Box-Jenkins method is likely to fail since no higher order terms exist in the models. On the other hand, a neural net with a single hidden layer can capture the nonlinearity of the logistic series. Lapedes and Farber (1988) reported that their neural net models produced far better forecasts than conventional methods for some chaotic time series, including the logistic series.

A feedforward neural network can be regarded as a general, non-linear model. In effect, it is a complex function consisting of a convoluted set of activation functions $f \in C$, where C is a set of continuously differentiable functions, and the parameter set W called weights. In particular, the activation functions are sigmoid functions as used in our neural net models. The output of a feedforward neural net can be written as:

$$o = f(\sum_j w_{jk} f_j(\sum_m w_{mj} f_m(\cdots f_l(\sum_i w_{il} x_i)))) \tag{9}$$

where x_i is the i^{th} element of the input vector x.

It has been proven that —with an unlimited number of processing units in the hidden layer— a feed-forward neural net with a single hidden layer can serve as a universal approximator to any continuous function(Funahashi 1989; Hornik, Stinchcombe and White 1990). A theorem by Kolmogorov (1957) can be applied to feed-forward neural nets to yield a two layer neural network that —with a finite number of hidden layer units— can represent any continuous function (Hecht-Nielsen 1989). These theoretic results make feed-forward neural nets reasonable candidates for any complex function mapping. Neural nets have the potential to represent any complex, nonlinear underlying mapping that may govern changes in a time series.

Without a hidden layer, the neural net model becomes a function of a linear combination of the input variables.

$$o = f(\sum_i w_{il} x_i). \tag{10}$$

In our case, x_i is defined as x_{t-i}, thus equation 10 is similar to the Box-Jenkins model except that it 1) contains a nonlinear activation function $f(x)$, and 2) contains no white noise. However, the differences can be negligible as the activation function can be changed to a linear function. The sigmoid function is fairly linear in the middle of its range, and more input units can be

created that take into account random errors as input variables. If we consider a general feed-forward neural net (with hidden layer) that has direct connections from the input units to the output units (figure 6), then we essentially have a model that combines a linear model (direct connections) and a nonlinear model (through the hidden layer). Hence we may conclude that neural net models are super sets of the Box-Jenkins models. Neural nets have the ability to encompass more complicated time series. Neural nets can also be used as forecast models other than self-projection used in time series forecasting. For instance, causal model and combined causal and self-projecting models can be easily built with neural nets.[3] Utans and Moody (1991) have studied neural net causal models for corporate bond rating prediction.

3.2 Robustness of the neural net model

With different training settings, the neural net models were shown to produce good forecasts. Although there are many factors that affect the performance of neural net models, there seems to exist a trade-off between those factors. For example, the forecast errors are large for many series when the training parameters are small (table 1, $\eta = \alpha = 0.1$). However, extended training length can often reduce forecast errors for those cases. Results in table 5 and table 6 show that neural net structure (number of hidden units) has an effect on forecast performance, but the effect is not significant for a wide range of options. Even with fixed structures —as used by many researchers— neural nets were shown to be able to compete with conventional methods (Lapedes and Farber 1987; Sharda and Patil 1990). Thus, the neural net models are fairly robust.

The robustness of neural net models is also reflected by the fact that they are essentially assumption-free models, although assumptions about the training set can be made and statistic inferences can be carried out (White 1989). This assumption-free property makes them applicable to a wide range of pattern recognition problems. The Box-Jenkins model, as other statistic based models, is subject to satisfiablity of the assumptions of the data series (e.g., the random errors follow a normal distribution).

[3]The causal model with the Box-Jenkins method is known as Box-Jenkins transfer function model. The transfer function model requires repeated applications of univariate Box-Jenkins method. The comparative performance of the neural net causal model and the Box-Jenkins transfer function model is a sublect of future study.

3.3 Generalization

Unless the neural net solution oscillates due to large training parameters, the training error (fitting error) always decreases as the training length increases. This is not true for forecasting. The results in table 1 show that in most cases increasing training length led to improved forecast. In some cases in table 1 and most cases in table 2, longer training of the network led to deteriorated forecasting. This is the results of overfitting. That is, the model fits too much to the peculiar features of the training set and loses its generalization ability.

The Box-Jenkins method circumvents the overfitting problem through the diagnostic model validation process. The number of parameters in the Box-Jenkins models are controlled in that only those parameters that contribute to the data fitting with certain statistical significance are retained. There are no established procedures for preventing overfitting in neural net models, although a number of techniques have been suggested in the literature (Weigend, Huberman and Rumelhart 1990).

The significance of weights can also be examined. Those weights that do not significantly affect the training error can be set to zero. When a weight associated with a input unit is set to zero, the corresponding input ceases to contribute to the output. Hence, a reduced model is obtained.

To test the efficacy of this technique, a 13×1 neural net model for the airline data was used. Table 7 shows the weight changes of the neural net during the training process. The weights of the neural network correspond to the coefficients of the input variables (since there is no hidden layer), and the bias in the output unit corresponds to a constant term. Although the neural network model is not a linear model —because of the sigmoidal activation function of the output unit— it nevertheless identified the most important inputs (inputs with the larger value of coefficients), as the Box-Jenkins model did. Note that those inputs are identified only after a sufficient number of training epochs.

Table-7: Weights and bias of the trained 13×1 neural net.

epochs	Connection weights w_{t-i}													bias
	t-1	t-2	t-3	t-4	t-5	t-6	t-7	t-8	t-9	t-10	t-11	t-12	t-13	
100	1.7	-.2	.2	-.6	.5	.0	.3	-.6	.0	.0	1.5	2.3	.0	-2.3
500	2.9	.0	.1	-.4	.4	-.1	.4	-.6	.2	.0	.9	3.6	-2.3	-2.4
1000	4.5	-.1	.3	.4	.7	-.7	.7	-.7	.5	-.2	.6	4.3	-4.6	-2.4

With the three inputs x_{t-1}, x_{t-12} and x_{t-13} identified as the most significant variables, a reduced neural net model with only the three input variables was tested. Table 8 shows the forecast errors of the full model and the reduced model. The reduced model had smaller forecast errors.

Table-8: Forecast error (MAPS) for full and reduced models.

Data	Model	500 Epochs		1000 Epochs	
		$\eta = \alpha = .5$	$\eta = \alpha = .1$	$\eta = \alpha = .5$	$\eta = \alpha = .1$
airline	full	5.87	6.06	4.26	4.35
	reduced	4.21	5.43	3.78	4.04

3.4 Learning in neural nets

The representation power of neural nets cannot be utilized unless efficient learning algorithms exist. The back-propagation (BP) algorithm for training a feed-forward neural net is essentially a gradient descent based algorithm. Two major deficiencies of the BP algorithm are (1) slow convergence and (2) getting stuck in local minima. The first problem has been investigated by many researchers (Becker and Le Cun 1989; Solla, Levin and Fleisher 1988) and much progress has been made. Less progress has been made in attacking the second problem. Fortunately, the problem is rare for most applications (Hecht-Nielsen 1989) and can be alleviated by changing network structures (Rumelhart, Hinton and Williams 1986).

Most of the time needed to build a neural net model is spent on the training process. Some experiments might also need to be done to determine a good neural net structure. But the neural net models are fairly robust as shown above. Choosing a neural net is relatively easy compared to the building of a Box-Jenkins model. The model identification and validation processes of the Box-Jenkins method need expert knowledge input. Additionally, it is not easy to understand the statistical mechanism used in the model identification procedure. In the sense that less statistic

background and user input is required, neural net models are relatively easy to use than the Box-Jenkins method.

However, compared to the Box-Jenkins method, the neural net approach has not fully matured. Unlike the Box-Jenkins method, neural nets lack a systematic procedure for model building. Often, ad hoc or even arbitrary neural net structures and training procedures are used. Due to the robustness of the model, good performance can still be obtained, however, the lack of a theoretical background and systematic procedures in model building has hindered their applicability. Recent theoretical studies of neural nets has cast some hope in developing automatic neural net forecasting systems. More efficient training algorithms have been developed (Fahlman and Lebiere 1990). Some researchers have studied dynamically constructed neural nets. That is, the network structure is determined during the learning process of a particular problem (Fahlman and Lebiere 1990; Frean 1990). The training parameters can also be determined automatically, hence freeing the user from having to make an educated guess (Jacobs 1988).

Another disadvantage of neural net models is that the trained neural nets do not offer us much information about the underlying structure (function) of a time series. In other words, compared to the Box-Jenkins method, which gives us simple linear equations, neural nets lack the ability to elucidate or describe concisely the time series that they have learned. This shortcoming might be overcome though. Recently, many researchers have studied extract rules (e.g., production rules) from a trained neural net (Bochereau and Bourgine 1990; Fu 1991).

4 Summary and Conclusions

Our study on neural nets as models for time series forecasting was inspired by the inconsistency of reported neural net performance. In trying to answer the questions as to when and why neural nets may be used as good forecast models, we performed a series of experiments on 16 time series. Fourteen of those series were chosen from a published source where the neural net models were found to result in large forecast errors for those series. Our experiments showed that the performance of neural net models depends on many factors such as the nature of the data series, the network structure, and the neural net training procedure. The performance of neural net

models can be greatly improved with the proper combination of those factors.

For all of the series, neural nets were shown to be able to compete with or outperform the Box-Jenkins method. However, this happens only when appropriate combinations of neural net structure and training procedures were used. For long memory series (airline and sales data) and one-step-ahead forecasting, a neural net with a general structure was comparable to the Box-Jenkins method, with the Box-Jenkins method performing slightly better for the airline data. For series with more irregularity, neural net models generally performed better.

For series with structure change (series group 2), both the neural nets and the Box-Jenkins method resulted in large errors, although the neural net models were slightly better. This result indicates that the simple neural net models, as well as the Box-Jenkins model, are not able to deal with pattern shifting in the data series. Other models, such as causal forecast models should be used for those series.

Neural nets were shown to outperform the Box-Jenkins method in longer term forecasting (12-step-ahead and 24-step-ahead). The superiority of neural nets in long term forecasting is due to the fact that they can be easily built into direct multiple-step-ahead forecast models, while the Box-Jenkins model is inherently a single-step-ahead forecast model. When a stepwise multiple-step-ahead forecast is used, neural net models with a single-step forecast horizon are comparable to the Box-Jenkins method.

The Box-Jenkins method has been successfully used in practice, however, its modeling process is complicated and it is generally regarded as difficult to understand. Expert knowledge is needed to successfully apply the Box-Jenkins method. Neural nets are more general non-linear models than the Box-Jenkins models. They are relatively easy to use and require fewer user inputs, especially when dynamic training algorithms are used.

Neural nets presents a promising alternative approach to time series forecasting. They represent a powerful and robust computation paradigm. The neural network structures and training procedures will have a great impact on forecasting performance. Since the structures and training procedures used in our study are by no means the best, we believe that there is still much room for improvement of neural network forecasting.

Currently, there are no systematic procedures for building neural net forecasting models. However, neural network research is still a fast growing discipline. New theoretic and algorithmic results make it possible to build automatic neural net forecasting systems. The potential of adaptive learning and the emergence of high speed parallel processing architectures make neural nets a powerful alternative for forecasting.

Acknowledgement

We are indebted to Dr. Chung Chen at Syracuse University and Dr. Gary Koehler at University of Florida for some insightful discussions. Thanks are due to Dr. Ramesh Sharda at Oklahoma State University for the time series data. We are also thankful to two anonymous referees who have given us helpful comments and suggestions on an earlier draft of the paper. Paul Fishwick would like to thank the National Science Foundation (Award IRI8909152) for partial support during this research period.

REFERENCES

Becker, S. and Le Cun, Y. 1989. Improving the convergence of back-propagation learning with second order methods. In Touretzky, D., Hinton, G., and Sejnowski, T., editors, *Proceedings of the 1988 Connectionist Models Summer School*, pages 29–37, San Mateo. (Pittsburg 1988), Morgan Kaufmann.

Bochereau, L. and Bourgine, P. 1990. Rule extraction and validity domain on a multilayer neural network. In *International Joint Conference on Neural Networks*, volume 1, pages 97–100, San Diego.

Bowerman, B. L. and O'connell, R. T. 1987. *Time Series Forecasting: Unified Concepts and Computer Implementation, 2nd ed.* Duxbury Press, Boston, TX.

Box, G. E. P. and Jenkins, G. M. 1970. *Time Series Analysis: Forecasting and Control.* Holden-Day, San Francisco, CA.

Dutta, S. and Shekhar, S. 1988. Bond rating: A non-conservative application of neural networks. In *International Joint Conference on Neural Networks*, volume 2, pages 443–450.

Fahlman, S. and Lebiere, C. 1990. The cascade-correlation learning architecture. In Touretzky, D., editor, *Advances in Neural Information Processing Systems*, volume 2, pages 524–532, San Mateo. (Denver 1989), Morgan Kaufmann.

Fisher, D. H. and McKusick, K. B. 1989. An empirical comparison of id3 and back-propagation. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, MI.

Fishwick, P. A. 1989. Neural network models in simulation: A comparison with traditional modeling approaches. In *Proceedings of Winter Simulation Conference*, pages 702–710, Washington, DC.

Frean, M. 1990. The upstart algorithm: A method for constructing and training feedforward neural networks. *Neural Computation*, 2:198–209.

Fu, L. 1991. Rule learning by searching on adapted nets. In *Proceedings of AAAI-91 Conference*, pages 590–595, Anaheim, CA.

Funahashi, K. 1989. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183–192.

Hecht-Nielsen, R. 1989. Theory of the backpropagation neural network. In *International Joint Conference on Neural Networks*, volume 1, pages 593–605, New York. (Washington 1989), IEEE.

Hinton, G. E. 1989. Connectionist learning procedures. *Artificial Intelligence*, 40:185–234.

Hoff, C. J. 1983. *A Practical Guide to BOX-JENKINS Forecasting.* Lifetime Learning Publications, Belmont, CA.

Hornik, K., Stinchcombe, M., and White, H. 1990. Using multilayer feedforward networks for universal approximation. *Neural Networks*, 3:551–560.

Jacobs, R. 1988. Increased rates of convergence through learning rate adaptation. *Neural Networks*, 1:295–307.

Lapedes, A. and Farber, R. 1987. Nonlinear signal processing using neural networks: Prediction and system modelling. Technical Report LA–UR–87–2662, Los Alamos National Laboratory, Los Alamos, NM.

Lee, C. J. and Chen, C. 1990. Structural changes and the forecasting of quarterly accounting earnings in the utility industry. *Journal of Accounting and Economics*, 13:93–122.

Lippmann, R. 1987. An introduction to computing with neural nets. *IEEE ASSP Magazine*, pages 4–22.

Makridakis, S. e. 1982. The accuracy of extrapolation (time series) mehtods: Results of a forecasting competition. *Journal of Forecasting*, 1:111–153.

Newton, H. 1988. *TIMESLAB: A Time Series Analysis Laboratory*. Wadsworth & Brooks Cole, California.

Rumelhart, D., Hinton, G., and Williams, R. 1986. Learning representations by back-propagating errors. *Nature*, 323:533–536. Reprinted in Anderson 1988.

Rumelhart, D., McClelland, J., and the PDP Research Group 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. MIT Press, Cambridge.

Sharda, R. and Ireland, T. 1987. A empirical test of automatic forecasting systems. Technical Report ORSA/TIMS Meeting, New Orleans.

Sharda, R. and Patil, R. 1990. Neural networks as forecasting experts: An empirical test. In *International Joint Conference on Neural Networks*, volume 1, pages 491–494, Washington, D.C. IEEE.

Simpson, P. 1990. *Artificial Neural Systems: Foundations, Paradigms, Applications, and Implementations*. Pergamon Press.

Singleton, J. C. and Surkan, A. J. 1990. Modeling the judgement of bond rating agencies: Artificial intelligence applied to finance. In *1990 Midwest Finance Association Meetings*, Chicago, IL.

Solla, S., Levin, E., and Fleisher, M. 1988. Accelerated learning in layered neural networks. *Complex Systems*, 2:625–639.

Sontag, E. 1990. On the recognition capabilities of feedforward nets. Technical Report Report SYCON-90-03, Rutgers Center for System and Control.

Tong, H. and Lim, K. 1980. Threshold autoregression, limit cycle and cyclical data. *Journal of Royal Statistical Society B*, 42:245.

Utans, J. and Moody, J. 1991. Selecting neural network architectures via the prediction risk: Application to corporate bond rating prediction. In *The First International Conference on Artificial Intelligence Applications on Wall Street*, Los Alamitos, CA. IEEE.

Weigend, A., Huberman, B., and Rumelhart, D. 1990. Predicting the future: A connectionist approach. Technical Report Stanford-PDP-90-01, Stanfor University, Stanford, CA.

White, H. 1989. Some asymptotic results for learning in single hidden-layer feedforward neural models. *Journal of the American Statistical Association*, 184:1003–1013.