Time series forecasting using neural networks vs. Box-Jenkins methodology

Zaiyong Tang , Chrys de Almeida * Paul A. Fishwick **
* Department of Decision & Information Sciences
** Department of Computer & Information Sciences
University of Florida
Gainesville, FL 32611

We discuss the results of a comparative study of the performance of neural networks and conventional methods in forecasting time series. Our work was initially inspired by previously published works that yielded inconsistent results about comparative performance. We have experimented with three time series of different complexity using different feed forward, backpropagation neural network models and the standard Box-Jenkins model. Our experiments demonstrate 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. We note that some of the comparable results arise since the neural network and time series model appear to be functionally similar models. We have found that for time series of different complexities there are optimal neural network topologies and parameters that enable them to learn more efficiently. Our initial conclusions are that neural networks are robust and provide good long-term forecasting. They are also parsimonious in their data requirements. Neural networks represent a promising alternative for forecasting, but there are problems determining the optimal topology and parameters for efficient learning.

Introduction

Artificial neural networks have been widely studied and applied to a variety of areas. One of these areas is forecasting. Although there have been many encouraging reports, there are, at the same time, several questions remaining unanswered. Many reports fail to compare their neural network performances against those of traditional methods.

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. Sharda and Patil (1990) conducted a forecasting competition between a neural network model and a traditional forecasting technique (namely t he Box-Jenkins method) using 75 time series of various nature. They concluded that the simple neural network model could forecast about as well as the Box-Jenkins forecasting system. Each of the methods performed better then the other about half of the time. The experiment of Fishwick (1989) shows, however, for a ballistics trajectory function approximation problem, the neural network used offered little competition to the traditional linear regression and surface response model. Surveying those papers leads us to ask why there is such a discrepancy? Can neural networks really compete with conventional methods? Why, or why not?

Experimental results from Sharda and Patil (1990) show that periodicity of time series does not have significant influence on the relative performance of the neural ne tworks and the Box-Jenkins approach. But there are significant performance differences as these two methods are applied to some of their chosen time series. For example, in terms of MAPS (mean absolute percent error), neural networks sometimes outperformed the Box-Jenkins approach by more than 100 percent. In some other cases, the latter proved much better. The authors did not provide any explanations as to when and why one approach is superior to the other. In light of the controversy about neural network applications, we feel it is of both theoretic and practical importance to answer some of these questions.

In this paper, a comparative study is carried out to investigate the forecasting capability of neural networks and Box-Jenkins models which are among those forecasting models most successfully applied in practice. Three typical time series were selected: international airline passenger data, domestic car sale data in the U.S., and foreign car sale data in the U.S.. Box- Jenkins model forecasting was done with the time series analysis package *TIMESLAB* (Newton, 1988). Neural network simulations were performed using the *Back Propagation* (BP) program in the *Parallel Distributed Processing* (PDP) package by McClelland and Rumelhart (1988).

In the following section, we present the experiments of forecasting with Box-Jenkins models and the neural networks. Section 3 focuses on the effects of neural network parameters and structure on training and forecasting performance. Following that, in section 4, we will offer some insights that we gained from our experiments, and discuss the questions we raised above. Finally, in section 5, we present our conclusions and directions for further research.

Box-Jenkins Model vs. NN Forecasting

Data

The three time series we selected are shown in Figure 1. Airline Passenger data (Fig. 1a) shows a long memory pattern — there is an apparent increasing trend and seasonal pattern. Domestic car sale (Fig. 1b) and foreign car sale data (Fig. 1c) also show seasonal patterns, though not very clearly. Foreign car sales presents an increasing trend while domestic car sales appears more irregular, and can hence be classified as a short memory time series.

Box-Jenkins Models

The most popular ARMA-model based forecasting method is the Box-Jenkins approach, which involves the following steps (Newton, 1988):

(1) model identification

(2) parameter estimation

(3) consideration of alternative ARIMA models, if necessary

(4) forecasting based on the chosen models.

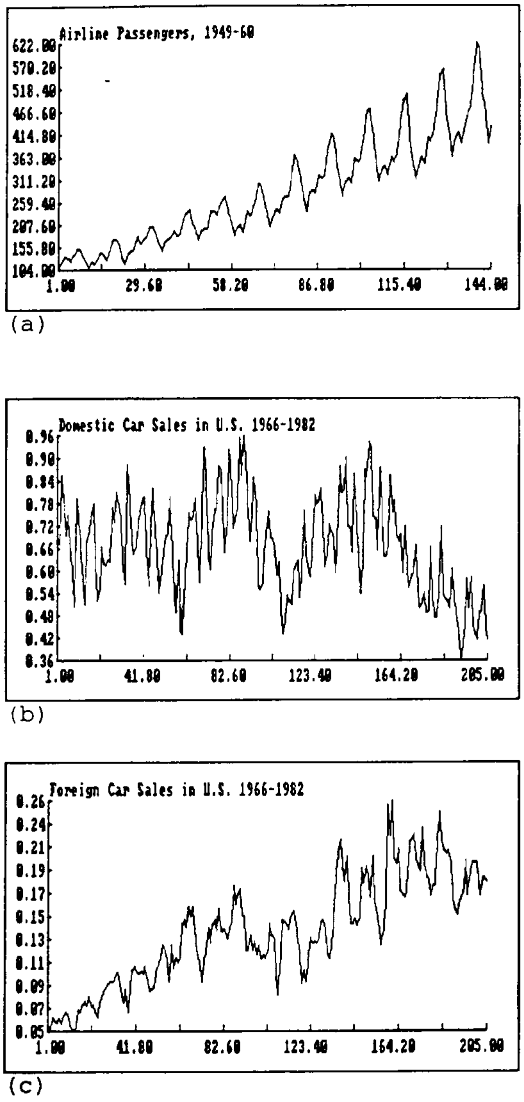


Figure 1. Time Series Data

The general Box-Jenkins model has the following form (Bowerman, 1987, P. 120).

$$\phi_p(B)\,\phi_P(B^L)\,(1-B^L)^D(1-B)^d y_t = \delta + \theta_q(B)\theta_Q(B^L)a_t \quad (1)$$

where $\phi(B)$ and $\theta(B)$ are autoregressive and moving average operators respectively ; B is the back shift operator; a_t is called white noise with normal distribution $N(0,\sigma^2)$; δ is a constant, and y_t is the time series data, transformed if necessary.

Using *TIMESLAB*'s model identification macro, we determined the Box-Jenkins model for the airline data to be:

$$(1-B^{12})(1-B)y_t = (1-\theta_1 B)(1-\theta_{1,12}B)\,a_t \qquad (2)$$

and the model for the car sales data to be:

$$(1-\phi_1 B+\phi_2 B^2+\phi_3 B^3)(1-B_{12})(1-B)\,y_t = (1-\theta_{1,12}B)a_t \quad (3)$$

Neural Network Models

Although there has been some research on the design of optimal neural network structures (Kurg and Hwang, 1988), it is still largely an art to determine the number of hidden layers and number of units in each hidden layer. The effects of neural network structure on the training of and forecasting by the neural network will be discussed in the next section. To ensure an effective comparison, in this section we use one hidden layer for all of the neural networks considered. This configuration is also the choice of Sharda and Patil (1990). The number of units in the hidden layer equals the number of inputs, which are set to be 1, 6, 12 and 24, corresponding to one month, half a year, one year and two year input data respectively. The number of output units are set to be 1, 6, 12 and 24, corresponding to 1, 6, 12 and 24-period- ahead forecasting respectively. Neural network structure is denoted as IxHxO, where I, H, and O represent number of input units, hidden units, and output units respectively.

The training parameters are initially set to be the same for different neural network structures, that is, learning rate = 0.5 and momentum = 0.9. The discussion on how to choose the best combination of the parameters will be left to the next section. In this part of the experiment, instead of using fixed training parameters, we adjusted the parameters during training to facilitate faster convergence. We observed that a training schedule, just like the cooling schedule in simulated annealing, plays an important role in neural network learning.

Forecasting results

Three cases were investigated in the forecasting performance comparison. (1) the amount of data used; (2) the number of periods for the forecast; and (3) the number of input variables. Of the original time series, the last 24 items were saved to compare with forecasted values. The forecasting results are summarized in the following tables. Note that all real values represent the total sum of square error (tss) for the 24 period forecast.

Table 1. Airline passenger forecast (tss)

forecast period	Box-Jenkins model	Neural network
1	0.0071	0.0105
6	0.0168	0.0285
12	0.0340	0.0336
24	0.1318	0.0129

Table 2. Training error and forecast error (tss with different input pattern)

input pattern	training error	forecast error
1 × 6 × 1	0.1008	0.0614
6 × 6 × 1	0.0940	0.0613
12 × 12 × 1	0.0338	0.0113
24 × 24 × 1	0.0221	0.0105

Table 3. Input data amount and forecast error (tss)

input data	training error	forecast error
2 yr.	0.0044	0.1524
5 yr.	0.0156	0.0208
10 yr.	0.0338	0.0113

Table 4. Car Sale Forecast (tss)

input data	Box-Jenkins model	Neural net (12 inputs)
domestic car		
9 yr.	0.19439	0.10130
15 yr	0.18156	0.10110
foreign car		
9 yr.	0.00724	0.00590
15 yr.	0.00670	0.00630

Table 1 shows that with one-period-ahead and six-period-ahead forecasts, the Box-Jenkins model outperforms the neural network for the selected structures and training methods, while for the 12-period-ahead and 24-period-ahead forecasts the neural network is better. It is not surprising that as the forecast horizon extends, Box-Jenkins model performs less well, since the model is best suited for short term forecast. 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-period-ahead forecast, the neural network resulted in much smaller error then the Box-Jenkins model. The fact that the 24- period-ahead forecast is better than the 6-period-ahead and the 12-period-ahead forecast suggest that in the later case, the networks may not have been trained to produce the optimal results.

Examining the forecast errors resulting from the Box-Jenkins model and the neural network, we found that the neural network provides very nice forecasts except for the small bumps at each yearly cycle. The Box-Jenkins model can nicely reproduce the 'details' of the original series, while the neural network we trained

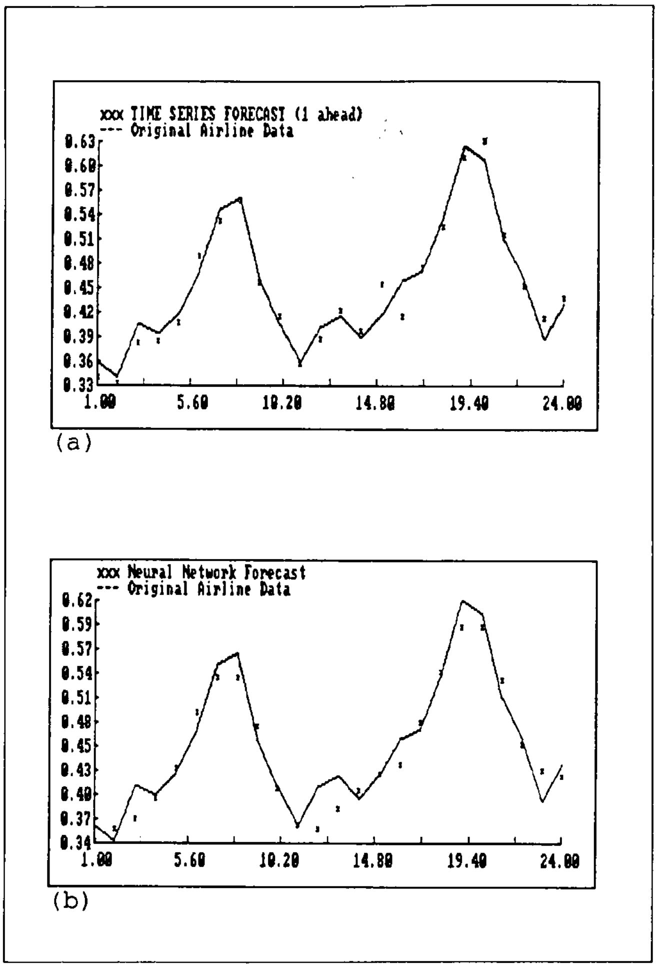


Figure 2. Box-Jenkins vs NN Forecasts

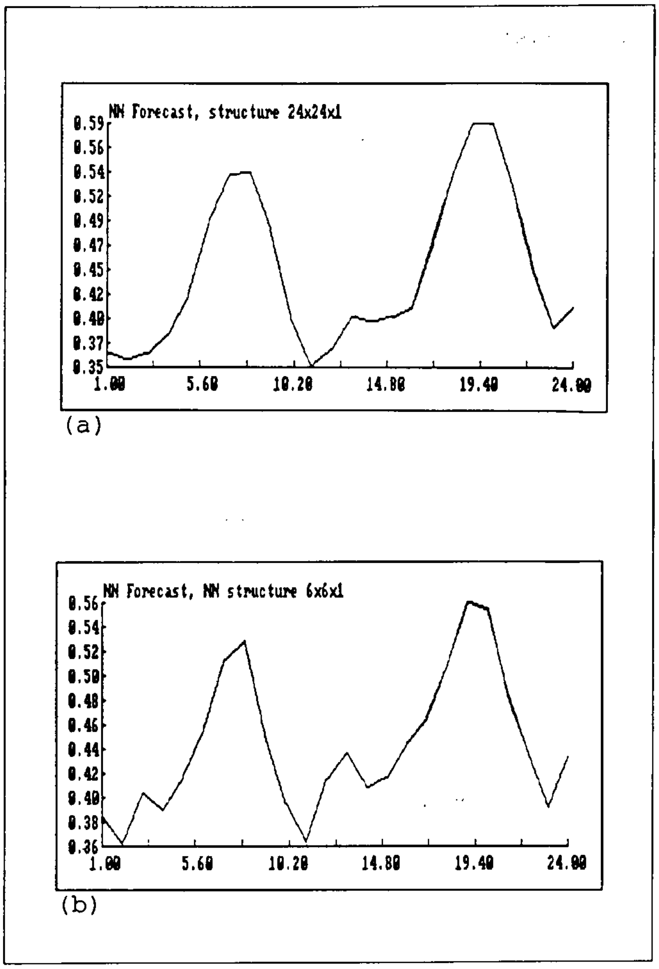


Figure 3. NN Forecasts with Different Input Patterns

provides somewhat smooth curve, ignoring the 'details'. There could be many factors attributing to the result. We would expect that changing the network structure and/ or the training parameter could improve the forecast of those 'details'. Changing the input pattern also helps in this aspect as can be seen from Figure 3.

When the input number increases, the forecasting performance of neural network improves, as shown in Table 2. Figure 3 depicts the forecasting differences with six and twenty four inputs. More inputs will provide more information, and hence will provide more accurate forecasts. It is interesting to note that with six inputs, the network learned the time series pattern better than with twenty four inputs, but it did not learn the increasing trend very well.

The amount of data, or equivalently, the number of training patterns also affects the forecast performance (Table 3). Although for the long memory series, more training patterns results in more accurate forecasts, the neural network can perform reasonably well with short series of input data. The Box-Jenkins model does not work well or does not work at all for short input series. This can be regarded as one of the advantages of the neural network over Box- Jenkins model.

For domestic car sale data, the Box-Jenkins model did less well than the neural network, although both result in relatively large errors due to the irregular nature of the series (Table 4). The two methods produced comparable results for foreign car sale forecast. Comparing the forecasting results for the three time series with different memory patterns, we noticed that, as the complexity of the time series increase, the Box-Jenkins model become less competitive against the neural network model.

Neural Network Structure and Training Parameter Analysis

As we mentioned in section 2, neural network structure affects its forecasting ability. For instance, with different input patterns, which correspond to different neural network structures, the trained neural network present different forecast patterns. We also mentioned that some of the neural networks used in our forecasting may not have been trained optimally. The following explores more on how neural network structure and training parameters affect their performance.

The Effect of Hidden Layer

We compare two neural network structures: (1) with hidden layer, and (2) without hidden layer. The training parameters are the same as described in section 2. Table 5 summarizes the forecasting results.

Table 5. The effect of NN structure on forecast

NN Structure	Forecast error
12 × 12 × 1	0.0113
12 × 1	0.0131
24 × 24 × 1	0.0105
24 × 1	0.0215

The results show that adding a hidden layer improves the forecasting performance of neural networks. While plotting the forecast results, we found that the neural networks without a hidden layer actually learned the pattern better. The forecast data present a pattern very close to those resulted from the Box-Jenkins model. This is not too surprising since without a hidden layer, the neural network is close functionally to a linear model. Larger errors of neural networks without the hidden layer result from improper estimation of the time series trend. With a hidden layer, the neural networks learned a smoother mapping (Figure 4a).

The Effect of Training Parameters

The effect of training parameters on the neural network learning were studied. The same randomly generated initial weight set was used in all the experiments. Table 6 and table 7 present the results, where LR is the learning rate, MO is the momentum value, TSS is the total sum of square error, and EPO is the number of training epochs.

For the airline passenger data, which is the least complex, the network converges quite rapidly and reaches error level specified by the global error criterion. At higher learning rates the convergence is much faster. For the other two series, the convergence towards a global minimum appears to be taking place for very small values of LR while the momentum is high. As the complexity of the pattern decreases the occurrence of local minima are less frequent, and a larger LR is accommodated.

Discussions

We began our study by asking when and why the neural networks offer superior performance to conventional methods. We have answered, at least partially, the first part of the question in the last two sections. Now let us try to answer the second part of the question by examining how the Box-Jenkins model and the neural network model perform input-output mapping.

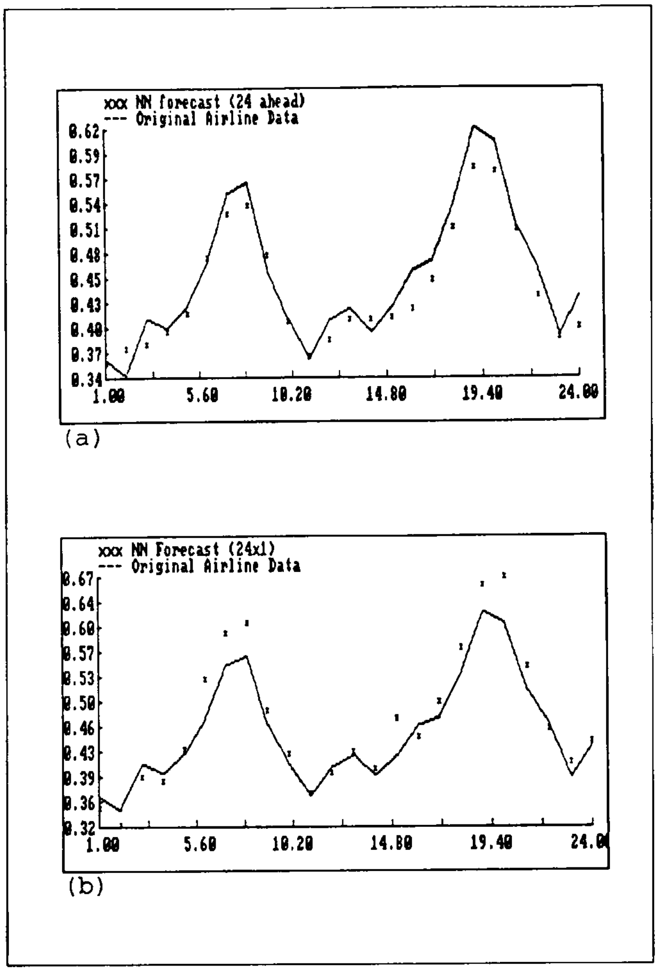


Figure 4. Effect of Hidden Layer on Forecast.

Table 6. Effect of Momentum for Fixed Learning Rate

LR	=	.1		
Domestic Car Sales:				
MO	=	.9	.5	.1
TSS	=	.2333	.2386	.2547
EPO	=	1000	1000	1000
Foreign Car Sales:				
MO	=	.9	.5	.1
TSS	=	.0225	.0302	.0362
EPO	=	1000	1000	1000

Input-Output Mapping with Box-Jenkins Model

As we have stated in section 2, the model we identified for the airline data is:

$$(1 - B^{12})(1 - B)y_t = (1 - \theta_1 B)(1 - \theta_{1,12} B)a_t \qquad (4)$$

The same model has been identified by other re-

Table 7. Effect of Learning Rate for Fixed Momentum

MO	=	.9			
Domestic Car Sales:					
LR	=	.05	.1	.2 .	5
TSS	=	.2350	.2333	5.7897	5.7897
EPO	=	1000	1000	1000	1000
Foreign Car Sales:					
LR	=	.05	.1 .	2	.5
TSS	=	.0242	.0225	1.6921	1.6921
EPO	=	1000	1000	1000	1000
Airline Passenger Data:					
LR	=	.1	.2 .	5	
TSS	=	.01	.01	.01	
EPO	=	324	163	68	

searchers for the airline passenger data (Newton, 1988). Rewriting the model, we have the following:

$$(1 - B^1 - B^{12} + B^{13})y_t = (1 - \theta_1 B - \theta_{1,12} B^{12} + \theta_1 \theta_{1,12} B^{13})a_t \quad (5)$$

or

$$y_t = y_{t-12} + (y_{t-1} - y_{t-13}) + (a_t - \theta_1 a_{t-1} - \theta_{1,12} a_{t-12} + \theta_1 \theta_{1,12} a_{t-13}) \quad (6)$$

Equation (6) 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 shocks (or residuals) of period t, t – 1, t – 12 and t – 13 on the forecast.

From above analysis, it is then easy to see why the Box-Jenkins model can provide an accurate forecast for long memory time series, even for small bumps between the large seasonal peaks (Figure 2a), as long as there is a fixed pattern of the time series. But for short memory series, there is no definite pattern. Using a model similar to (6) is apparently not sufficient.

Input-output Mapping with Neural Network

Most neural networks use sigmoidal activation functions which make it possible for the neural network to perform a complicated input-output mapping through the back propagation procedure. In essence, a neural network model is equivalent to a set of algebraic equations arranged in a hierarchical order to form a input-output mapping. Changing the structure of a neural network is nothing more than changing the hierarchical order of the algebraic equations. Training a neural network is just another way to say estimating the parameters in the complex input-output transformation function (Fishwick, 1989), formed by activation functions.

It is difficult to study the input-output transformation function of a neural network with hidden layer(s). Without a hidden layer, the neural network output is a function of a linear combination of the input variables. Then, we would like to know how this function is related to the Box-Jenkins model. Since the Box-Jenkins model for the airline data says that the time series value at time t is determined by a the value at time t – 1, t – 12, and t – 13, and some random shocks, we would expect that these data items also play an important role in the neural network model. This is indeed true as shown in Table 8. To compare with the Box-Jenkins model, we used 13 inputs (corresponding data at time t – 1 through t – 13) to forecast the value at time t. The training methods are the same as those described in section 2.

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 lager value of coefficients), as the Box-Jenkins model did. Note that those inputs are identified only after sufficient iterations of training epochs.

One important issue in understanding the neural network's ability to learn complicated mapping was brought up by Lapedes and Farber (1987). That is, the mode decomposition perspective. They made it clear that a neural network learns the relationship of the input-output pairs by using combinations of sigmoid functions to approximate the underlying time series mapping. Consider a series x(t + 1) = 4*x (t) (1 – x(t)), for instance, Lapedes and Farber trained a simple 1 × 5 × 1 (1 input unit, 5 hidden units, 1 output unit) neural network, and t hey showed that using the weights and

Table 8. Weights and bias associated with the neural network

Training	**weights**													**bias**
100 epo.	1.7	–.2	.2	–.6	.5	.0	.3	–.6	.0	.0	1.5	2.3	.0	–2.33
500	2.9	.0	.1	–.4	.4	–.1	.4	–.6	.2	.0	.9	3.6	–2.33	–2.43
1000	34.5	–.1	.3	–.4	.7	–.7	.7	–.7	.5	–.2	.6	4.3	–4.63	–2.43

biases from the trained network, they could explicitly write out an input-output mapping as a linear combination of 5 sigmoid functions and a linear function of the input variable. This linear combination of functions can approximate the underlying mapping, *i.e.*, x(t + 1) = 4* x(t) (1 – x(t)), very accurately within the domain.

Figure 5 shows the neural network mapping of the airline passenger time series. Different network structures result in different mapping. The one with a hidden layer is flatter than the one without a hidden layer. We see that a neural network is capable of reproducing the underlying pattern of a time series, with a proper network structure and appropriate training.

From above discussions, we see that training the neural network is essentially finding a set of hierarchically ordered activation functions which could best approximate the underlying mapping of the time series. With this perspective, we raise the following questions:

1. How accurate is the approximation?
2. Can we use analytical approach to find the approximation?

It is beyond the scope of this project to provide rigorous answer to above questions. Here we offer some of our initial thoughts gained from this project.

(1). Theoretically, the neural network mapping approximation could be made arbitrarily accurate. In practice, however, the accuracy is affected by many factors, such as the neural network structure, the accuracy of the input data. The fact that there is no established theory concerning neural network structure and training procedure makes accurate mapping more difficult. This could be one explanation to the discrepancy of reports on neural network performance in forecasting. In time series forecasting, suppose the series has a long memory, that is, a deterministic pattern (e.g., airline passenger), then the series can be described by the Box-Jenkins model very accurately. Using neural networks to approximate the mapping can hardly compete with Box-Jenkins model. However, Box-Jenkins model is sensitive to noise, and since it builds its forecast on previous observations, the method is only good for short term forecasting. A neural network (with a hidden layer) bases its forecasting on the approximated underlying mapping. Hence it is more robust and better in the case of long term forecasting as shown in sections 2.

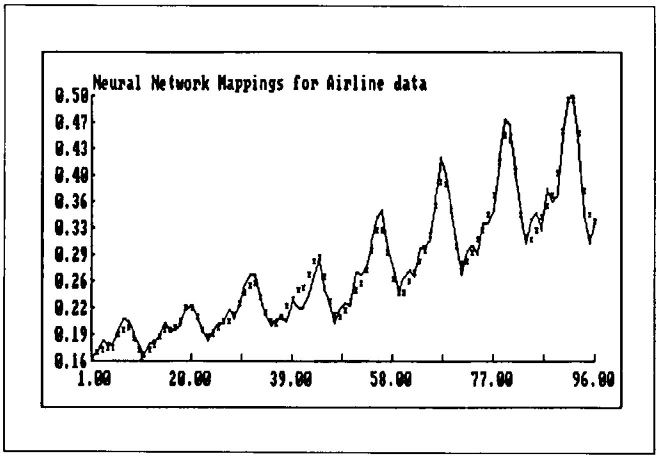


Figure 5. NN Mapping of the Time Series

(2). Training a neural network is time consuming. Since it can be regarded as an optimization problem, one would think analytical methods could be applicable, at least when the network structure is fixed. The problem is then finding a set of parameters, corresponding weights and biases in the neural network, such that the error function is minimized. Admittedly, this would be difficult since we do not have the experience of working on hierarchically ordered sigmoidal functions. But it may not be impossible. Research in improving neural network learning, such as fast learning rules, stepwise training, and so on, could lend insights to the analysis of the hierarchical mapping function. On the other hand, analytical study of the mapping approximation may, in turn, bring new ideas to improve neural network learning.

Effect of Training on NN Learning

A proper training schedule can greatly improve both learning speed and the forecasting accuracy. Our experiments show that stepwise training does indeed help learning as reported by others in the literature. In our stepwise training case, we use, instead of different pattern, part of the total training series to train the network first, then the whole series.

Few papers have talked about changing training parameters during the training process. We found that a properly designed training schedule can improve learning speed, and increase chance of convergence. This can be thought as an analog to simulated annealing, in which the cooling schedule plays a key role in avoiding local optima and finding good solutions.

Conclusions

Neural networks provide a promising alternative approach to time series forecasting. For time series with long memory, both Box-Jenkins model and neural network performs well, with Box-Jenkins model slightly better for short term forecasting. Neural network proved to be better in long term forecasting; however, we are still performing comparisons with alternative long term forecasting methods. For short memory time series, neural networks appear to be superior to Box-Jenkins model.

By approximating the underlying mapping of the time series, a neural network provides robust forecasting in the cases of irregular time series. The neural network model is more parsimonious in data requirement than the Box-Jenkins model.

Neural networks can be trained to approximate the underlying mapping of a time series, albeit the accuracy of the approximation depends on a number of factors such as the neural network structure, learning method, and training procedure. Without a hidden layer, the neural network model is functionally similar to the Box-Jenkins model.

The neural network structures and training procedures have great impact on its forecasting performance. Since the structures and training procedures used in our study are by no means the best, we believe there is still much room for improvement of neural network forecasting. We consider the following topics worth studying:

(1) Applying different neural network models, for instance, recurrent and cascade network, to forecasting.

(2) Building neural network causal models for time series forecasting.

(3) Comparing the neural network model with other conventional forecasting approaches.

Acknowledgement

Dr. Chung Chen at Syracuse University provided the car sales data and some insightful discussions. The airline data is taken from *TIMESLAB* (Newton, 1988). Zaiyong Tang would like to thank the DIS Department at University of Florida for financial support. Paul Fishwick would like to thank the National Science Foundation (Award IRI8909152) for partial support during this research period.

References

Bowerman, B.L. and O'Connell, R.T., 1987, *Time Series Forecasting*, 2nd. Ed. PWS Publishers, pp. 120.

Fishwick, P.A., 1989, "Neural Network Models in Simulation: A Comparison with Traditional Modeling Approaches," *Proceedings of Winter Simulation Conference*, Washington, DC, pp. 702-710.

Kung, S.Y. and Hwang, J.N., 1988, "An Algebraic Projection Analysis for Optimal Hidden Units Size and Learning Rates in Back-Propagation Learning," *Proceedings of IEEE International Neural Network Conference*, Vol. II.

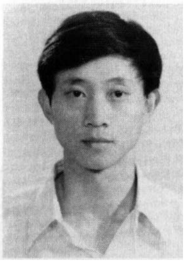
Lapedes, A. and Farber, R., 1987, "Nonlinear Signal Processing Using Neural Network: Prediction and System Modelling," Los Alamos National Lab Technical Report, LA-UR-87 -2662.

Rumelhart, D.E. and McClelland, J.L., 1986, *Parallel Distributed Processing, Vol . 1*, MIT Press, Cambridge, Massachusetts.

McClelland, J.L. and Rumelhart, D.E., 1988, *Explorations in Parallel Distributed Processing*, MIT Press, Cambridge, Massachusetts.

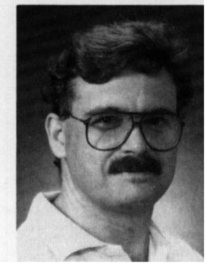
Newton, H.J., 1988. *TIMESLAB: A Time Series Analysis Laboratory*, Wadsworth & Brooks/Cole Publishing Company, California.

Sharda, R. and Patil, R.B., 1990, "Neural Networks as Forecasting Experts : An Empirical Test," *Proceedings of the IJCNN Meeting*, Washington, pp. 491-494.



ZAIYONG TANG is a Ph.D. candidate in the Department of Decision and Information Sciences at the University of Florida. He received his B.E. in Mechanical Engineering from Chongqing University, China, in 1982; M.S. in MAterials Science from Chengdu University of Sci. & Tech., China, in 1984; M.S. in Transportation from Washington State University in 1987. His research interests include neural networks, machine learning, global optimization and forecasting.

CHRYSANTHUS S. DE ALMEIDA is reading for his Ph.D. in Decision and Information Sciences at the University of Florida. He obtained the B.Sc. in Electrical Engineering from the University of Sri Lanka in 1975, and the M.B.A. from the University of Florida in 1988. He has held positions of varied responsibilities, both engineering and managerial, in the national radio and television broadcasting organizations in Sri Lanka. His research interests are in knowledge-based information systems.



PAUL A. FISHWICK is an associate professor in the Department of Computer and Information Sciences at the University of Florida. He received a PhD in Computer and Information Science from the University of Pennsylvania in 1986. His research interests are in computer simulation modeling and analysis methods for complex systems. He is a member of IEEE, IEEE Society for Systems, Man and Cybernetics, IEEE Computer Society, The Society for Computer Simulation, ACM and AAAI. Dr. Fishwick was chairman of the IEEE Computer Society technical committee on simulation (TCSIM) for two years (1988-1990) and he is on the editorial boards of several journals including the *ACM Transactions on Modeling and Computer Simulation, The Transactions of The Society for Computer Simulation, International Journal of Computer Simulation,* and the *Journal of Systems Engineering.*