ELSEVIER

International Journal of Forecasting 14 (1998) 35–62

international journal of forecasting

Forecasting with artificial neural networks: The state of the art

Guoqiang Zhang, B. Eddy Patuwo, Michael Y. Hu*

Graduate School of Management, Kent State University, Kent, Ohio 44242-0001, USA

Accepted 31 July 1997

Abstract

Interest in using artificial neural networks (ANNs) for forecasting has led to a tremendous surge in research activities in the past decade. While ANNs provide a great deal of promise, they also embody much uncertainty. Researchers to date are still not certain about the effect of key factors on forecasting performance of ANNs. This paper presents a state-of-the-art survey of ANN applications in forecasting. Our purpose is to provide (1) a synthesis of published research in this area, (2) insights on ANN modeling issues, and (3) the future research directions. © 1998 Elsevier Science B.V.

Keywords: Neural networks; Forecasting

1. Introduction

Recent research activities in artificial neural networks (ANNs) have shown that ANNs have powerful pattern classification and pattern recognition capabilities. Inspired by biological systems, particularly by research into the human brain, ANNs are able to learn from and generalize from experience. Currently, ANNs are being used for a wide variety of tasks in many different fields of business, industry and science (Widrow et al., 1994).

One major application area of ANNs is forecasting (Sharda, 1994). ANNs provide an attractive alternative tool for both forecasting researchers and practitioners. Several distinguishing features of ANNs make them valuable and attractive for a forecasting task. First, as opposed to the traditional model-based methods, ANNs are data-driven self-adaptive methods in that there are few a priori assumptions about the models for problems under study. They learn from examples and capture subtle functional relationships among the data even if the underlying relationships are unknown or hard to describe. Thus ANNs are well suited for problems whose solutions require knowledge that is difficult to specify but for which there are enough data or observations. In this sense they can be treated as one of the multivariate nonlinear nonparametric statistical methods (White, 1989; Ripley, 1993; Cheng and Titterington, 1994). This modeling approach with the ability to learn from experience is very useful for many practical problems since it is often easier to have data than to have good theoretical guesses about the underlying laws governing the systems from which data are generated. The problem with the

*Corresponding author. Tel.: +1 330 6722772 ext. 326; fax: +1 330 6722448; e-mail: mhu@kentvm.kent.edu

PII S0169-2070(97)00044-7

data-driven modeling approach is that the underlying rules are not always evident and observations are often masked by noise. It nevertheless provides a practical and, in some situations, the only feasible way to solve real-world problems.

Second, ANNs can generalize. After learning the data presented to them (a sample), ANNs can often correctly infer the unseen part of a population even if the sample data contain noisy information. As forecasting is performed via prediction of future behavior (the unseen part) from examples of past behavior, it is an ideal application area for neural networks, at least in principle.

Third, ANNs are universal functional approximators. It has been shown that a network can approximate any continuous function to any desired accuracy (Irie and Miyake, 1988; Hornik et al., 1989; Cybenko, 1989; Funahashi, 1989; Hornik, 1991, 1993). ANNs have more general and flexible functional forms than the traditional statistical methods can effectively deal with. Any forecasting model assumes that there exists an underlying (known or unknown) relationship between the inputs (the past values of the time series and/or other relevant variables) and the outputs (the future values). Frequently, traditional statistical forecasting models have limitations in estimating this underlying function due to the complexity of the real system. ANNs can be a good alternative method to identify this function.

Finally, ANNs are nonlinear. Forecasting has long been the domain of linear statistics. The traditional approaches to time series prediction, such as the Box-Jenkins or ARIMA method (Box and Jenkins, 1976; Pankratz, 1983), assume that the time series under study are generated from linear processes. Linear models have advantages in that they can be understood and analyzed in great detail, and they are easy to explain and implement. However, they may be totally inappropriate if the underlying mechanism is nonlinear. It is unreasonable to assume a priori that a particular realization of a given time series is generated by a linear process. In fact, real world systems are often nonlinear (Granger and Terasvirta, 1993). During the last decade, several nonlinear time series models such as the bilinear model (Granger and Anderson, 1978), the threshold autoregressive (TAR) model (Tong and Lim, 1980), and the autoregressive conditional heteroscedastic (ARCH) model (Engle, 1982) have been developed. (See De Gooijer and Kumar (1992) for a review of this field.) However, these nonlinear models are still limited in that an explicit relationship for the data series at hand has to be hypothesized with little knowledge of the underlying law. In fact, the formulation of a nonlinear model to a particular data set is a very difficult task since there are too many possible nonlinear patterns and a prespecified nonlinear model may not be general enough to capture all the important features. Artificial neural networks, which are nonlinear data-driven approaches as opposed to the above model-based nonlinear methods, are capable of performing nonlinear modeling without a priori knowledge about the relationships between input and output variables. Thus they are a more general and flexible modeling tool for forecasting.

The idea of using ANNs for forecasting is not new. The first application dates back to 1964. Hu (1964), in his thesis, uses the Widrow's adaptive linear network to weather forecasting. Due to the lack of a training algorithm for general multi-layer networks at the time, the research was quite limited. It is not until 1986 when the backpropagation algorithm was introduced (Rumelhart et al., 1986b) that there had been much development in the use of ANNs for forecasting. Werbos (1974), (1988) first formulates the backpropagation and finds that ANNs trained with backpropagation outperform the traditional statistical methods such as regression and Box-Jenkins approaches. Lapedes and Farber (1987) conduct a simulated study and conclude that ANNs can be used for modeling and forecasting nonlinear time series. Weigend et al. (1990), (1992); Cottrell et al. (1995) address the issue of network structure for forecasting real-world time series. Tang et al. (1991), Sharda and Patil (1992), and Tang and Fishwick (1993), among others, report results of several forecasting comparisons between Box-Jenkins and ANN models. In a recent forecasting competition organized by Weigend and Gershenfeld (1993) through the Santa Fe Institute, winners of each set of data used ANN models (Gershenfeld and Weigend, 1993).

Research efforts on ANNs for forecasting are considerable. The literature is vast and growing. Marquez et al. (1992) and Hill et al. (1994) review

the literature comparing ANNs with statistical models in time series forecasting and regression-based forecasting. However, their review focuses on the relative performance of ANNs and includes only a few papers. In this paper, we attempt to provide a more comprehensive review of the current status of research in this area. We will mainly focus on the neural network modeling issues. This review aims at serving two purposes. First, it provides a general summary of the work in ANN forecasting done to date. Second, it provides guidelines for neural network modeling and fruitful areas for future research.

The paper is organized as follows. In Section 2, we give a brief description of the general paradigms of the ANNs, especially those used for the forecasting purpose. Section 3 describes a variety of the fields in which ANNs have been applied as well as the methodology used. Section 4 discusses the key modeling issues of ANNs in forecasting. The relative performance of ANNs over traditional statistical methods is reported in Section 5. Finally, conclusions and directions of future research are discussed in Section 6.

2. An overview of ANNs

In this section we give a brief presentation of artificial neural networks. We will focus on a particular structure of ANNs, multi-layer feedforward networks, which is the most popular and widely-used network paradigm in many applications including forecasting. For a general introductory account of ANNs, readers are referred to Wasserman (1989); Hertz et al. (1991); Smith (1993). Rumelhart et al. (1986a), (1986b), (1994), (1995); Lippmann (1987); Hinton (1992); Hammerstrom (1993) illustrate the basic ideas in ANNs. Also, a couple of general review papers are now available. Hush and Horne (1993) summarize some recent theoretical developments in ANNs since Lippmann (1987) tutorial article. Masson and Wang (1990) give a detailed description of five different network models. Wilson and Sharda (1992) present a review of applications of ANNs in the business setting. Sharda (1994) provides an application bibliography for researchers in Management Science/Operations Research. A bibliography of neural network business applications research is also given by Wong et al. (1995). Kuan and White (1994) review the ANN models used by economists and econometricians and establish several theoretical frames for ANN learning. Cheng and Titterington (1994) make a detailed analysis and comparison of ANNs paradigms with traditional statistical methods.

Artificial neural networks, originally developed to mimic basic biological neural systems– the human brain particularly, are composed of a number of interconnected simple processing elements called neurons or nodes. Each node receives an input signal which is the total ''information'' from other nodes or external stimuli, processes it locally through an activation or transfer function and produces a transformed output signal to other nodes or external outputs. Although each individual neuron implements its function rather slowly and imperfectly, collectively a network can perform a surprising number of tasks quite efficiently (Reilly and Cooper, 1990). This information processing characteristic makes ANNs a powerful computational device and able to learn from examples and then to generalize to examples never before seen.

Many different ANN models have been proposed since 1980s. Perhaps the most influential models are the multi-layer perceptrons (MLP), Hopfield networks, and Kohonen's self organizing networks. Hopfield (1982) proposes a recurrent neural network which works as an associative memory. An associative memory can recall an example from a partial or distorted version. Hopfield networks are non-layered with complete interconnectivity between nodes. The outputs of the network are not necessarily the functions of the inputs. Rather they are stable states of an iterative process. Kohonen's feature maps (Kohonen, 1982) are motivated by the self-organizing behavior of the human brain.

In this section and the rest of the paper, our focus will be on the multi-layer perceptrons. The MLP networks are used in a variety of problems especially in forecasting because of their inherent capability of arbitrary input–output mapping. Readers should be aware that other types of ANNs such as radial-basis functions networks (Park and Sandberg, 1991, 1993; Chng et al., 1996), ridge polynomial networks (Shin and Ghosh, 1995), and wavelet networks (Zhang and Benveniste, 1992; Delyon et al., 1995) are also very

useful in some applications due to their function approximating ability.

An MLP is typically composed of several layers of nodes. The first or the lowest layer is an input layer where external information is received. The last or the highest layer is an output layer where the problem solution is obtained. The input layer and output layer are separated by one or more intermediate layers called the hidden layers. The nodes in adjacent layers are usually fully connected by acyclic arcs from a lower layer to a higher layer. Fig. 1 gives an example of a fully connected MLP with one hidden layer.

For an explanatory or causal forecasting problem, the inputs to an ANN are usually the independent or predictor variables. The functional relationship estimated by the ANN can be written as

$$y = f(x_1, x_2, \cdots, x_p),$$

where $x_1, x_2, \cdots, x_p$ are p independent variables and y is a dependent variable. In this sense, the neural network is functionally equivalent to a nonlinear regression model. On the other hand, for an extrapolative or time series forecasting problem, the inputs are typically the past observations of the data series and the output is the future value. The ANN performs the following function mapping

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

where y_t is the observation at time t. Thus the ANN is equivalent to the nonlinear autoregressive model for time series forecasting problems. It is also easy to incorporate both predictor variables and time-lagged observations into one ANN model, which amounts to the general transfer function model. For a discussion on the relationship between ANNs and general ARMA models, see Suykens et al. (1996).

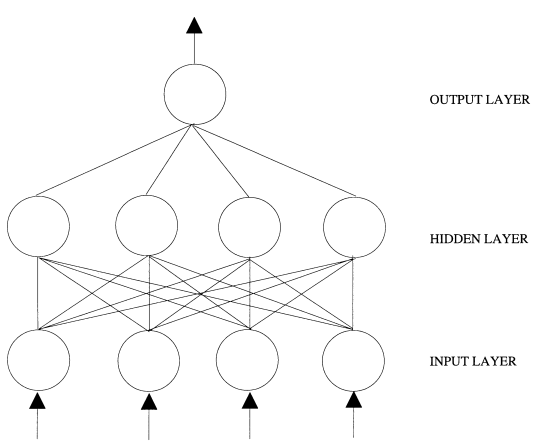

Fig. 1. A typical feedforward neural network (MLP).

Before an ANN can be used to perform any desired task, it must be trained to do so. Basically, training is the process of determining the arc weights which are the key elements of an ANN. The knowledge learned by a network is stored in the arcs and nodes in the form of arc weights and node biases. It is through the linking arcs that an ANN can carry out complex nonlinear mappings from its input nodes to its output nodes. An MLP training is a supervised one in that the desired response of the network (target value) for each input pattern (example) is always available.

The training input data is in the form of vectors of input variables or training patterns. Corresponding to each element in an input vector is an input node in the network input layer. Hence the number of input nodes is equal to the dimension of input vectors. For a causal forecasting problem, the number of input nodes is well defined and it is the number of independent variables associated with the problem. For a time series forecasting problem, however, the appropriate number of input nodes is not easy to determine. Whatever the dimension, the input vector for a time series forecasting problem will be almost always composed of a moving window of fixed length along the series. The total available data is usually divided into a training set (in-sample data) and a test set (out-of-sample or hold-out sample). The training set is used for estimating the arc weights while the test set is used for measuring the generalization ability of the network.

The training process is usually as follows. First, examples of the training set are entered into the input nodes. The activation values of the input nodes are weighted and accumulated at each node in the first hidden layer. The total is then transformed by an activation function into the node's activation value. It in turn becomes an input into the nodes in the next layer, until eventually the output activation values are found. The training algorithm is used to find the weights that minimize some overall error measure such as the sum of squared errors (SSE) or mean squared errors (MSE). Hence the network training is

actually an unconstrained nonlinear minimization problem.

For a time series forecasting problem, a training pattern consists of a fixed number of lagged observations of the series. Suppose we have N observations $y_1, y_2, \cdots y_N$ in the training set and we need 1-step-ahead forecasting, then using an ANN with n input nodes, we have $N-n$ training patterns. The first training pattern will be composed of $y_1, y_2, \cdots, y_n$ as inputs and y_{n+1} as the target output. The second training pattern will contain $y_2, y_3, \cdots, y_{n+1}$ as inputs and y_{n+2} as the desired output. Finally, the last training pattern will be y_{N-n}, $y_{N-n+1}, \cdots y_{N-1}$ for inputs and y_N for the target. Typically, an SSE based objective function or cost function to be minimized during the training process is

$$E = \frac{1}{2} \sum_{i=n+1}^{N} (y_i - a_i)^2,$$

where a_i is the actual output of the network and $1/2$ is included to simplify the expression of derivatives computed in the training algorithm.

3. Applications of ANNs as forecasting tools

Forecasting problems arise in so many different disciplines and the literature on forecasting using ANNs is scattered in so many diverse fields that it is hard for a researcher to be aware of all the work done to date in the area. In this section, we give an overview of research activities in forecasting with ANNs. First we will survey the areas in which ANNs find applications. Then we will discuss the research methodology used in the literature.

3.1. Application areas

One of the first successful applications of ANNs in forecasting is reported by Lapedes and Farber (1987), (1988). Using two deterministic chaotic time series generated by the logistic map and the Glass-Mackey equation, they designed the feedforward neural networks that can accurately mimic and predict such dynamic nonlinear systems. Their results show that ANNs can be used for modeling and forecasting nonlinear time series with very high accuracy.

Following Lapedes and Farber, a number of papers were devoted to using ANNs to analyze and predict deterministic chaotic time series with and/or without noise. Chaotic time series occur mostly in engineering and physical science since most physical phenomena are generated by nonlinear chaotic systems. As a result, many authors in the chaotic time series modeling and forecasting are from the field of physics. Lowe and Webb (1990) discuss the relationship between dynamic systems and functional interpolation with ANNs. Deppisch et al. (1991) propose a hierarchically trained ANN model in which a dramatic improvement in accuracy is achieved for prediction of two chaotic systems. Other papers using chaotic time series for illustration include Jones et al. (1990); Chan and Prager (1994); Rosen (1993); Ginzburg and Horn (1991), (1992); Poli and Jones (1994).

The sunspot series has long served as a benchmark and has been well studied in statistical literature. Since the data are believed to be nonlinear, non-stationary and non-Gaussian, they are often used as a yardstick to evaluate and compare new forecasting methods. Some authors focus on how to use ANNs to improve accuracy in predicting sunspot activities over traditional methods (Li et al., 1990; De Groot and Wurtz, 1991), while others use the data to illustrate a method (Weigend et al., 1990, 1991, 1992; Ginzburg and Horn, 1992, 1994; Cottrell et al., 1995).

There is an extensive literature in financial applications of ANNs (Trippi and Turban, 1993; Azoff, 1994; Refenes, 1995; Gately, 1996). ANNs have been used for forecasting bankruptcy and business failure (Odom and Sharda, 1990; Coleman et al., 1991; Salchenkerger et al., 1992; Tam and Kiang, 1992; Fletcher and Goss, 1993; Wilson and Sharda, 1994), foreign exchange rate (Weigend et al., 1992; Refenes, 1993; Borisov and Pavlov, 1995; Kuan and Liu, 1995; Wu, 1995; Hann and Steurer, 1996), stock prices (White, 1988; Kimoto et al., 1990; Schoneburg, 1990; Bergerson and Wunsch, 1991; Yoon and Swales, 1991; Grudnitski and Osburn, 1993), and others (Dutta and Shekhar, 1988; Sen et al., 1992; Wong et al., 1992; Kryzanowski et al., 1993; Chen, 1994; Refenes et al., 1994; Kaastra and

Boyd, 1995; Wong and Long, 1995; Chiang et al., 1996; Kohzadi et al., 1996).

Another major application of neural network forecasting is in electric load consumption study. Load forecasting is an area which requires high accuracy since the supply of electricity is highly dependent on load demand forecasting. Park and Sandberg (1991) report that simple ANNs with inputs of temperature information alone perform much better than the currently used regression-based technique in forecasting hourly, peak and total load consumption. Bacha and Meyer (1992) discuss why ANNs are suitable for load forecasting and propose a system of cascaded subnetworks. Srinivasan et al. (1994) use a four-layer MLP to predict the hourly load of a power system. Other studies in this area include Bakirtzis et al. (1995); Brace et al. (1991); Chen et al. (1991); Dash et al. (1995); El-Sharkawi et al. (1991); Ho et al. (1992); Hsu and Yang (1991a), (1991b); Hwang and Moon (1991); Kiartzis et al. (1995); Lee et al. (1991); Lee and Park (1992); Muller and Mangeas (1993); Pack et al. (1991a,b); Peng et al. (1992); Pelikan et al. (1992); Ricardo et al. (1995).

Many researchers use data from the well-known M-competition (Makridakis et al., 1982) for comparing the performance of ANN models with the traditional statistical models. The M-competition data are mostly from business, economics and finance. Several important works include Kang (1991); Sharda and Patil (1992); Tang et al. (1991); Foster et al. (1992); Tang and Fishwick (1993); Hill et al. (1994), (1996). In the Santa Fe forecasting competition (Weigend and Gershenfeld, 1993), six nonlinear time series from very different disciplines such as physics, physiology, astrophysics, finance, and even music are used. All the data sets are very large compared to the M-competition where all time series are quite short.

Many other forecasting problems have been solved by ANNs. A short list includes airborne pollen (Arizmendi et al., 1993), commodity prices (Kohzadi et al., 1996), environmental temperature (Balestrino et al., 1994), helicopter component loads (Haas et al., 1995), international airline passenger traffic (Nam and Schaefer (1995), macroeconomic indices (Maasoumi et al., 1994), ozone level (Ruiz-Suarez et al., 1995), personnel inventory (Huntley, 1991), rainfall (Chang et al., 1991), river flow (Karunanithi et al., 1994), student grade point averages (Gorr et al., 1994), tool life (Ezugwu et al., 1995), total industrial production (Aiken et al., 1995), trajectory (Payeur et al., 1995), transportation (Duliba, 1991), water demand (Lubero, 1991), and wind pressure profile (Turkkan and Srivastava, 1995).

3.2. Methodology

There are many different ways to construct and implement neural networks for forecasting. Most studies use the straightforward MLP networks (Kang, 1991; Sharda and Patil, 1992; Tang and Fishwick, 1993) while others employ some variants of MLP. Although our focus is on feedforward ANNs, it should be pointed out that recurrent networks also play an important role in forecasting. See Connor et al. (1994) for an illustration of the relationship between recurrent networks and general ARMA models. The use of the recurrent networks for forecasting can be found in Gent and Sheppard (1992); Connor et al. (1994); Kuan and Liu (1995).

Narendra and Parthasarathy (1990) and Levin and Narendra (1993) discuss the issue of identification and control of nonlinear dynamical systems using feedforward and recurrent neural networks. The theoretical and simulation results from these studies provide the necessary background for accurate analysis and forecasting of nonlinear dynamic systems.

Lapedes and Farber (1987) were the first to use the multi-layer feedforward networks for forecasting purposes. Jones et al. (1990) extend Lapedes and Farber (1987, (1988) by using a more efficient one dimensional Newton's method to train the network instead of using the standard backpropogation. Based on the above work, Poli and Jones (1994) build a stochastic MLP model with random connections between units and noisy response functions.

The issue of finding a parsimonious model for a real problem is critical for all statistical methods and is particularly important for neural networks because the problem of overfitting is more likely to occur with ANNs. The parsimonious models not only have the recognition ability, but also have the more important generalization capability. Baum and Haussler (1989) discuss the general relationship between the generalizability of a network and the size of the

training sample. Amirikian and Nishimura (1994) find that the appropriate network size depends on the specific tasks of learning.

Several researchers address the issue of finding networks with appropriate size for predicting real-world time series. Based on the information theoretic idea of minimum description length, Weigend et al. (1990), (1991), (1992) propose a weight pruning method called weight-elimination through introducing a term to the backpropagation cost function that penalizes network complexity. The weight elimination method dynamically eliminates weights during training to help overcome the network overfitting problem (learning the noise as well as rules in the data, see Smith, 1993). Cottrell et al. (1995) also discuss the general ANN modeling issue. They suggest a statistical stepwise method for eliminating insignificant weights based on the asymptotic properties of the weight estimates to help establish appropriate sized ANNs for forecasting. De Groot and Wurtz (1991) present a parsimonious feedforward network approach based on a normalized Akaike information criterion (AIC) (Akaike, 1974) to model and analyze the time series data.

Lachtermacher and Fuller (1995) employ a hybrid approach combining Box-Jenkins and ANNs for the purpose of minimizing the network size and hence the data requirement for training. In the exploratory phase, the Box-Jenkins method is used to find the appropriate ARIMA model. In the modeling phase, an ANN is built with some heuristics and the information on the lag components of the time series obtained in the first step. Kuan and Liu (1995) suggest a two-step procedure to construct the feedforward and recurrent ANNs for time series forecasting. In the first step the predictive stochastic complexity criterion (Rissanen, 1987) is used to select the appropriate network structures and then the nonlinear least square method is used to estimate the parameters of the networks. Barker (1990) and Bergerson and Wunsch (1991) develop hybrid systems combining ANNs with an expert system.

Pelikan et al. (1992) present a method of combining several neural networks with maximal decorrelated residuals. The results from combined networks show much improvement over a single neural network and the linear regression. Ginzburg and Horn (1994) also use two combined ANNs to improve time series forecasting accuracy. While the first network is a regular one for modeling the original time series, the second one is used to model the residuals from the first network and to predict the errors of the first. The combined result for the sunspots data is improved considerably over the one network method. Wedding and Cios (1996) describe a method of combining radial-basis function networks and the Box-Jenkins models to improve the reliability of time series forecasting. Donaldson and Kamstra (1996) propose a forecasting combining method using ANNs to overcome the shortcomings of the linear forecasting combination methods.

Zhang and Hutchinson (1993) and Zhang (1994) describe an ANN method based on a general state space model. Focusing on multiple step predictions, they doubt that an individual network would be powerful enough to capture all of the information in the available data and propose a cascaded approach which uses several cascaded neural networks to predict multiple future values. The method is basically iterative and one network is needed for prediction of each additional step. The first network is constructed solely using past observations as inputs to produce an initial one-step-ahead forecast; then a second network is constructed using all past observations and previous predictions as inputs to generate both one-step and two-step-ahead forecasts. This process is repeated until finally the last network used all past observations as well as all previous forecast values to yield the desired multi-step-ahead forecasts.

Chakraborty et al. (1992) consider using ANN approach to multivariate time series forecasting. Utilizing the contemporaneous structure of the trivariate data series, they adopt a combined approach of neural network which produces much better results than a separate network for each individual time series. Vishwakarma (1994) uses a two-layer ANN to predict multiple economic time series based on the state space model of Kalman filtering theory.

Artificial neural networks have also been investigated as an auxiliary tool for forecasting method selection and ARIMA model identification. Chu and Widjaja (1994) suggest a system of two ANNs for forecasting method selection. The first network is used for recognition of demand pattern in the data. The second one is then used for the selection of a

forecasting method among six exponential smoothing models based on the demand pattern of data, the forecasting horizon, and the type of industry where the data come from. Tested with both simulated and actual data, their system has a high rate of correct demand pattern identification and gives fairly good recommendation for the appropriate forecasting method. Sohl and Venkatachalam (1995) also present a neural network approach to forecasting model selection.

Jhee et al. (1992) propose an ANN approach for the identification of the Box-Jenkins models. Two ANNs are separately used to model the autocorrelation function (ACF) and the partial autocorrelation function (PACF) of the stationary series and their outputs give the orders of an ARMA model. In a latter paper, Lee and Jhee (1994) develop an ANN system for automatic identification of Box-Jenkins model using the extended sample autocorrelation function (ESACF) as the feature extractor of a time series. An MLP with a preprocessing noise filtering network is designed to identify the correct ARMA model. They find that this system performs quite well for artificially generated data and the real world time series and conclude that the performance of ESACF is superior to that of ACF and PACF in identifying correct ARIMA models. Reynolds (1993) and Reynolds et al. (1995) also propose an ANN approach to Box-Jenkins model identification problem. Two networks are developed for this task. The first one is used to determine the number of regular differences required to make a non-seasonal time series stationary while the second is built for ARMA model identification based on the information of ACF and PACF of the stationary series.

4. Issues in ANN modeling for forecasting

Despite the many satisfactory characteristics of ANNs, building a neural network forecaster for a particular forecasting problem is a nontrivial task. Modeling issues that affect the performance of an ANN must be considered carefully. One critical decision is to determine the appropriate architecture, that is, the number of layers, the number of nodes in each layer, and the number of arcs which interconnect with the nodes. Other network design decisions include the selection of activation functions of the hidden and output nodes, the training algorithm, data transformation or normalization methods, training and test sets, and performance measures.

In this section we survey the above-mentioned modeling issues of a neural network forecaster. Since the majority of researchers use exclusively fully-connected-feedforward networks, we will focus on issues of constructing this type of ANNs. Table 1 summarizes the literature on ANN modeling issues.

4.1. The network architecture

An ANN is typically composed of layers of nodes. In the popular MLP, all the input nodes are in one input layer, all the output nodes are in one output layer and the hidden nodes are distributed into one or more hidden layers in between. In designing an MLP, one must determine the following variables:

- the number of input nodes.
- the number of hidden layers and hidden nodes.
- the number of output nodes.

The selection of these parameters is basically problem-dependent. Although there exists many different approaches such as the pruning algorithm (Sietsma and Dow, 1988; Karnin, 1990; Weigend et al., 1991; Reed, 1993; Cottrell et al., 1995), the polynomial time algorithm (Roy et al., 1993), the canonical decomposition technique (Wang et al., 1994), and the network information criterion (Murata et al., 1994) for finding the optimal architecture of an ANN, these methods are usually quite complex in nature and are difficult to implement. Furthermore none of these methods can guarantee the optimal solution for all real forecasting problems. To date, there is no simple clear-cut method for determination of these parameters. Guidelines are either heuristic or based on simulations derived from limited experiments. Hence the design of an ANN is more of an art than a science.

4.1.1. The number of hidden layers and nodes

The hidden layer and nodes play very important roles for many successful applications of neural networks. It is the hidden nodes in the hidden layer that allow neural networks to detect the feature, to

Table 1
Summary of modeling issues of ANN forecasting

Researchers	Data type	Training/ test size	#input nodes	#hidden layer:node	#output nodes	Transfer fun. hidden:output	Training algorithm	Data normalization	Performance measure
Chakraborty et al. (1992)	Monthly price series	90/10	8	1:8	1	Sigmoid:sigmoid	BP*	Log transform.	MSE
Cottrell et al. (1995)	Yearly sunspots	220/?	4	1:2–5	1	Sigmoid:linear	Second order	None	Residual variance and BIC
De Groot and Wurtz (1991)	Yearly sunspots	221/35,55	4	1:0–4	1	Tanh:tanh	BP.BFGS LM** etc.	External linear to [0,1]	Residual variance
Foster et al. (1992)	Yearly and monthly data	*N-k/k***	5,8	1:3,10	1	N/A****	N/A	N/A	MdAPE and GMARE
Ginzburg and Horn (1994)	Yearly sunspots	220/35	12	1:3	1	Sigmoid:linear	BP	External linear to [0,1]	RMSE
Gorr et al. (1994)	Student GPA	90%/10%	8	1:3	1	Sigmoid:linear	BP	None	ME and MAD
Grudnitski and Osburn (1993)	Monthly S and P and gold	N/A	24	2:(24)(8)	1	N/A	BP	N/A	% prediction accuracy
Kang (1991)	Simulated and real time series	70/24 or 40/24	4,8,2	1,2:varied	1	Sigmoid:sigmoid	GRG2	External linear [−1,1] or [0.1,0.9]	MSE, MAPE MAD, *U*-coeff.
Kohzadi et al. (1996)	Monthly cattle and wheat prices	240/25	6	1:5	1	N/A	BP	None	MSE, AME, MAPE
Kuan and Liu (1995)	Daily exchange rates	1245/ varied	varied	1:varied	1	Sigmoid:linear	Newton	N/A	RMSE
Lachtermacher and Fuller (1995)	Annual river flow and load	100%/ synthetic	n/a	1:n/a	1	Sigmoid:sigmoid	BP	External simple	RMSE and Rank Sum
Nam and Schaefer (1995)	Monthly airline traffic	3,6,9 yrs/ 1 yr.	12	1:12,15,17	1	Sigmoid:sigmoid	BP	N/A	MAD
Nelson et al. (1994)	M-competition monthly	*N*−18/18	varied	1:varied	1	N/A	BP	None	MAPE
Schoneburg (1990)	Daily stock price	42/56	10	2:(10)(10)	1	Sigmoid:sine, sigmoid	BP	External linear to [0.1,0.9]	% prediction accuracy
Sharda and Patil (1992)	M-competition time series	*N*−*k*/*k***	12 for monthly	1:12 for monthly	1,8	Sigmoid:sigmoid	BP	Across channel linear [0.1,0.9]	MAPE
Srinivasan et al. (1994)	Daily load and relevant data	84/21	14	2:(19)(6)	1	Sigmoid:linear	BP	Along channel to [0.1,0.9]	MAPE
Tang et al. (1991)	Monthly airline and car sales	*N*−24/24	1,6,12,24	1:=input node #	1,6,12,24	Sigmoid:sigmoid	BP	N/A	SSE
Tang and Fishwick (1993)	M-competition	*N*−*k*/*k***	12:month 4:quarter	1:=input node #	1,6,12	Sigmoid:sigmoid	BP	External linear to [0.2,0.8]	MAPE
Vishwakarma (1994)	Monthly economic data	300/24	6	2:(2)(2)	1	N/A	N/A	N/A	MAPE
Weigend et al. (1992)	Sunspots exchange rate (daily)	221/59 501/215	12 61	1:8,3 1:5	1 2	Sigmoid:linear Tanh:linear	BP	None along channel statistical	ARV ARV
Zhang (1994)	Chaotic time series	100 000/ 500	21	2:(20)(20)	1–5	Sigmoid:sigmoid	BP	None	RMSE

* Backpropagation

** Levenberg-Marquardt

*** *N* is the number of training sample size; *k* is 6, 8 and 18 for yearly, monthly and quarterly data respectively.

**** Not available

capture the pattern in the data, and to perform complicated nonlinear mapping between input and output variables. It is clear that without hidden nodes, simple perceptrons with linear output nodes are equivalent to linear statistical forecasting models.

Influenced by theoretical works which show that a

single hidden layer is sufficient for ANNs to approximate any complex nonlinear function with any desired accuracy (Cybenko, 1989; Hornik et al., 1989), most authors use only one hidden layer for forecasting purposes. However, one hidden layer networks may require a very large number of hidden nodes, which is not desirable in that the training time and the network generalization ability will worsen.

Two hidden layer networks may provide more benefits for some type of problems (Barron, 1994). Several authors address this problem and consider more than one hidden layer (usually two hidden layers) in their network design processes. Srinivasan et al. (1994) use two hidden layers and this results in a more compact architecture which achieves a higher efficiency in the training process than one hidden layer networks. Zhang (1994) finds that networks with two hidden layers can model the underlying data structure and make predictions more accurately than one hidden layer networks for a particle time series from the Santa Fe forecasting competition. He also tries networks with more than two hidden layers but does not find any improvement. Their findings are in agreement with that of Chester (1990) who discusses the advantages of using two hidden layers over single hidden layer for general function mapping. Some authors simply adopt two hidden layers in their network modeling without comparing them to the one hidden layer networks (Vishwakarma, 1994; Grudnitski and Osburn, 1993; Lee and Jhee, 1994).

These results seem to support the conclusion made by Lippmann (1987); Cybenko (1988); Lapedes and Farber (1988) that a network never needs more than two hidden layers to solve most problems including forecasting. In our view, one hidden layer may be enough for most forecasting problems. However, using two hidden layers may give better results for some specific problems, especially when one hidden layer network is overladen with too many hidden nodes to give satisfactory results.

The issue of determining the optimal number of hidden nodes is a crucial yet complicated one. In general, networks with fewer hidden nodes are preferable as they usually have better generalization ability and less overfitting problem. But networks with too few hidden nodes may not have enough power to model and learn the data. There is no theoretical basis for selecting this parameter although a few systematic approaches are reported. For example, both methods for pruning out unnecessary hidden nodes and adding hidden nodes to improve network performance have been suggested. Gorr et al. (1994) propose a grid search method to determine the optimal number of hidden nodes.

The most common way in determining the number of hidden nodes is via experiments or by trial-and-error. Several rules of thumb have also been proposed, such as, the number of hidden nodes depends on the number of input patterns and each weight should have at least ten input patterns (sample size). To help avoid the overfitting problem, some researchers have provided empirical rules to restrict the number of hidden nodes. Lachtermacher and Fuller (1995) give a heuristic constraint on the number of hidden nodes. In the case of the popular one hidden layer networks, several practical guidelines exist. These include using ''$2n+1$'' (Lippmann, 1987; Hecht-Nielsen, 1990), ''$2n$'' (Wong, 1991), ''n'' (Tang and Fishwick, 1993), ''$n/2$'' (Kang, 1991), where n is the number of input nodes. However none of these heuristic choices works well for all problems.

Tang and Fishwick (1993) investigate the effect of hidden nodes and find that the number of hidden nodes does have an effect on forecast performance but the effect is not quite significant. We notice that networks with the number of hidden nodes being equal to the number of input nodes are reported to have better forecasting results in several studies (De Groot and Wurtz, 1991; Chakraborty et al., 1992; Sharda and Patil, 1992; Tang and Fishwick, 1993).

4.1.2. The number of input nodes

The number of input nodes corresponds to the number of variables in the input vector used to forecast future values. For causal forecasting, the number of inputs is usually transparent and relatively easy to choose. In a time series forecasting problem, the number of input nodes corresponds to the number of lagged observations used to discover the underlying pattern in a time series and to make forecasts for future values. However, currently there is no suggested systematic way to determine this number. The selection of this parameter should be included in the model construction process. Ideally, we desire a

small number of essential nodes which can unveil the unique features embedded in the data. Too few or too many input nodes can affect either the learning or prediction capability of the network.

Tang and Fishwick (1993), p. 376 claim that the number of input nodes is simply the number of autoregressive (AR) terms in the Box-Jenkins model for a univariate time series. This is not true because (1) for moving average (MA) processes, there are no AR terms; and (2) Box-Jenkins models are linear models. The number of AR terms only tells the number of linearly correlated lagged observations and it is not appropriate for the nonlinear relationships modeled by neural networks.

Most authors design experiments to help select the number of input nodes while others adopt some intuitive or empirical ideas. For example, Sharda and Patil (1992) and Tang et al. (1991) use 12 inputs for monthly data and four for quarterly data heuristically. Going through the literature, we find no consistent results for the issue of determining this important parameter. Some authors report the benefit of using more input nodes (Tang et al., 1991) while others find just the opposite (Lachtermacher and Fuller, 1995). It is interesting to note that Lachtermacher and Fuller (1995) report both bad effects of more input nodes for single-step-ahead forecasting and good effects for multi-step prediction. Some researchers simply adopt the number of input nodes used by previous studies (Ginzburg and Horn, 1994) while others arbitrarily choose one for their applications. Cheung et al. (1996) propose to use maximum entropy principles to identify the time series lag structure.

In our opinion, the number of input nodes is probably the most critical decision variable for a time series forecasting problem since it contains the important information about the complex (linear and/or nonlinear) autocorrelation structure in the data. We believe that this parameter can be determined by theoretical research in nonlinear time series analysis and hence improve the neural network model building process. Over the past decade, a number of statistical tests for nonlinear dependencies of time series such as Lagrange multiplier tests (Luukkonen et al., 1988; Saikkonen and Luukkonen, 1988), likelihood ratio-based tests (Chan and Tong, 1986), bispectrum tests (Hinich, 1982), and others (Keenan, 1985; Tsay, 1986; McLeod and Li, 1983; Lee et al., 1993) have been proposed. However, most tests are model dependent and none is superior to others in all situations. These problems also apply to the determination of the number of lags in a particular nonlinear model. One frequently used criterion for nonlinear model identification is the Akaike information criterion (AIC). However, there are still controversies surrounding the use of this criterion (De Gooijer and Kumar, 1992; Cromwell et al., 1994).

Recently, genetic algorithms have received considerable attention in the optimal design of a neural network (Miller et al., 1989; Guo and Uhrig, 1992; Jones, 1993; Schiffmann et al., 1993). Genetic algorithms are optimization procedures which can mimic natural selection and biological evolution to achieve more efficient ANN learning process (Happel and Murre, 1994). Due to their unique properties, genetic algorithms are often implemented in commercial ANN software packages.

4.1.3. The number of output nodes

The number of output nodes is relatively easy to specify as it is directly related to the problem under study. For a time series forecasting problem, the number of output nodes often corresponds to the forecasting horizon. There are two types of forecasting: one-step-ahead (which uses one output node) and multi-step-ahead forecasting. Two ways of making multi-step forecasts are reported in the literature. The first is called the iterative forecasting as used in the Box-Jenkins model in which the forecast values are iteratively used as inputs for the next forecasts. In this case, only one output node is necessary. The second called the direct method is to let the neural network have several output nodes to directly forecast each step into the future. Zhang (1994) cascaded method combines these two types of multi-step-ahead forecasting. Results from Zhang (1994) show that the direct prediction is much better than the iterated method. However, Weigend et al. (1992) report that the direct multi-step prediction performs significantly worse than the iterated single-step prediction for the sunspot data. Hill et al. (1994) conclude similar findings for 111 M-competition time series.

In our opinion, the direct multiple-period neural

network forecasting may be better for the following two reasons. First, the neural network can be built directly to forecast multi-step-ahead values. It has the benefits over the iterative method like the Box-Jenkins method in that the iterative method constructs only a single function which is used to predict one point each time and then iterates this function on its own outputs to predict points in the future. As the forecasts move forward, past observations are dropped off. Instead, forecasts rather than observations are used to forecast further future points. Hence it is typical that the longer the forecasting horizon, the less accurate the iterative method. This also explains why Box-Jenkins models are traditionally more suitable for short-term forecasting. This point can be seen clearly from the following k-step forecasting equations used in iterative methods such as Box-Jenkins:

$$\hat{x}_{t+1} = f(x_t, x_{t-1}, \cdots, x_{t-n}),$$

$$\hat{x}_{t+2} = f(\hat{x}_{t+1}, x_t, x_{t-1}, \cdots, x_{t-n+1}),$$

$$\hat{x}_{t+3} = f(\hat{x}_{t+2}, \hat{x}_{t+1}, x_t, x_{t-1}, \cdots, x_{t-n+2}),$$

$$\vdots$$

$$\hat{x}_{t+k} = f(x_{t+k-1}, \hat{x}_{t+k-2}, \cdots, \hat{x}_{t+1}, x_t, x_{t-1}, \cdots, x_{t-n+k-1}),$$

where x_t is the observation at time t, $\hat{x}_t$ is the forecast for time t, f is the function estimated by the ANN. On the other hand, an ANN with k output nodes can be used to forecast multi-step-ahead points directly using all useful past observations as inputs. The k-step-ahead forecasts from an ANN are

$$\hat{x}_{t+1} = f_1(x_t, x_{t-1}, \cdots, x_{t-n})$$

$$\hat{x}_{t+2} = f_2(x_t, x_{t-1}, \cdots, x_{t-n})$$

$$\vdots$$

$$\hat{x}_{t+k} = f_k(x_t, x_{t-1}, \cdots, x_{t-n})$$

where $f_1, \cdots, f_k$ are functions determined by the network.

Second, Box-Jenkins methodology is based heavily on the autocorrelations among the lagged data. It should be pointed out again that autocorrelation in essence measures only the linear correlation between the lagged data. In reality, correlation can be nonlinear and Box-Jenkins models will not be able to model these nonlinear relationships. ANNs are better in capturing the nonlinear relationships in the data. For example, consider an MA(1) model: $x_t = \varepsilon_t + 0.6\varepsilon_{t-1}$. Since the white noise ε_{t+1} is not forecastable at time t (0 is the best forecast value), the one-step-ahead forecast is $\hat{x}_{t+1} = 0.6(x_t - \hat{x}_t)$. However, at time t, we can not predict $x_{t+2} = \varepsilon_{t+2} + 0.6\varepsilon_{t+1}$ since both ε_{t+2} and ε_{t+1} are future terms of white noise series and are unforecastable. Hence the optimum forecast is simply $\hat{x}_{t+2} = 0$. Similarly, k-step-ahead forecasts: $\hat{x}_{t+k} = 0$ for $k \geq 3$. These results are expected since the autocorrelation is zero for any two points in the MA(1) series separated by two lags or more. However, if there is a nonlinear correlation between observations separated by two lags or more, the Box-Jenkins model can not capture this structure, causing more than one-step-ahead values unforecastable. This is not the case for an ANN forecaster.

4.1.4. The interconnection of the nodes

The network architecture is also characterized by the interconnections of nodes in layers. The connections between nodes in a network fundamentally determine the behavior of the network. For most forecasting as well as other applications, the networks are fully connected in that all nodes in one layer are only fully connected to all nodes in the next higher layer except for the output layer. However it is possible to have sparsely connected networks (Chen et al., 1991) or include direct connections from input nodes to output nodes (Duliba, 1991). Adding direct links between input layer and output layer may be advantageous to forecast accuracy since they can be used to model the linear structure of the data and may increase the recognition power of the network. Tang and Fishwick (1993) investigate the effect of direct connection for one-step-ahead forecasting but no general conclusion is reached.

4.2. Activation function

The activation function is also called the transfer function. It determines the relationship between

inputs and outputs of a node and a network. In general, the activation function introduces a degree of nonlinearity that is valuable for most ANN applications. Chen and Chen (1995) identify general conditions for a continuous function to qualify as an activation function. Loosely speaking, any differentiable function can qualify as an activation function in theory. In practice, only a small number of ''well-behaved'' (bounded, monotonically increasing, and differentiable) activation functions are used. These include:

1. The sigmoid (logistic) function:

   $f(x) = (1 + \exp(-x))^{-1}$;

2. The hyperbolic tangent (tanh) function:

   $f(x) = (\exp(x) - \exp(-x))/(\exp(x) + \exp(-x))$;

3. The sine or cosine function:

   $f(x) = \sin(x)$ or $f(x) = \cos(x)$;

4. The linear function:

   $f(x) = x$.

   Among them, logistic transfer function is the most popular choice.

There are some heuristic rules for the selection of the activation function. For example, Klimasauskas (1991) suggests logistic activation functions for classification problems which involve learning about average behavior, and to use the hyperbolic tangent functions if the problem involves learning about deviations from the average such as the forecasting problem. However, it is not clear whether different activation functions have major effects on the performance of the networks.

Generally, a network may have different activation functions for different nodes in the same or different layers (see Schoneburg (1990) and Wong (1991) for examples). Yet almost all the networks use the same activation functions particularly for the nodes in the same layer. While the majority of researchers use logistic activation functions for hidden nodes, there is no consensus on which activation function should be used for output nodes. Following the convention, a number of authors simply use the logistic activation functions for all hidden and output nodes (see, for example, Tang et al., 1991; Chakraborty et al., 1992; Sharda and Patil, 1992; Tang and Fishwick, 1993; Lachtermacher and Fuller, 1995; Nam and Schaefer, 1995). De Groot and Wurtz (1991) and Zhang and Hutchinson (1993) use the hyperbolic tangent transfer functions in both hidden and output layer. Schoneburg (1990) uses mixed logistic and sine hidden nodes and a logistic output node. Notice that when using these nonlinear squashing functions in the output layer, the target output values usually need to be normalized to match the range of actual outputs from the network since the output node with a logistic or a hyperbolic tangent function has a typical range of [0,1] or [−1,1] respectively.

Conventionally, the logistic activation function seems well suited for the output nodes for many classification problems where the target values are often binary. However, for a forecasting problem which involves continuous target values, it is reasonable to use a linear activation function for output nodes. Rumelhart et al. (1995) heuristically illustrate the appropriateness of using linear output nodes for forecasting problems with a probabilistic model of feedforward ANNs, giving some theoretic evidence to support the use of linear activation functions for output nodes. Researchers who use linear output nodes include Lapedes and Farber (1987), (1988); Weigend et al. (1990), (1991), (1992); Wong (1991); Ginzburg and Horn (1992), (1994); Gorr et al. (1994); Srinivasan et al. (1994); Vishwakarma (1994); Cottrell et al. (1995); Kuan and Liu (1995), etc. It is important to note that feedforward neural networks with linear output nodes have the limitation that they cannot model a time series containing a trend (Cottrell et al., 1995). Hence, for this type of neural networks, pre- differencing may be needed to eliminate the trend effects. So far no research has investigated the relative performance of using linear and nonlinear activation functions for output nodes and there have been no empirical results to support preference of one over the other.

4.3. Training algorithm

The neural network training is an unconstrained nonlinear minimization problem in which arc

weights of a network are iteratively modified to minimize the overall mean or total squared error between the desired and actual output values for all output nodes over all input patterns. The existence of many different optimization methods (Fletcher, 1987) provides various choices for neural network training. There is no algorithm currently available to guarantee the global optimal solution for a general nonlinear optimization problem in a reasonable amount of time. As such, all optimization algorithms in practice inevitably suffer from the local optima problems and the most we can do is to use the available optimization method which can give the ''best'' local optima if the true global solution is not available.

The most popularly used training method is the backpropagation algorithm which is essentially a gradient steepest descent method. For the gradient descent algorithm, a step size,, which is called the learning rate in ANNs literature, must be specified. The learning rate is crucial for backpropagation learning algorithm since it determines the magnitude of weight changes. It is well known that the steepest descent suffers the problems of slow convergence, inefficiency, and lack of robustness. Furthermore it can be very sensitive to the choice of the learning rate. Smaller learning rates tend to slow the learning process while larger learning rates may cause network oscillation in the weight space. One way to improve the original gradient descent method is to include an additional momentum parameter to allow for larger learning rates resulting in faster convergence while minimizing the tendency to oscillation (Rumelhart et al., 1986b). The idea of introducing the momentum term is to make the next weight change in more or less the same direction as the previous one and hence reduce the oscillation effect of larger learning rates. Yu et al. (1995) describe a dynamic adaptive optimization method of the learning rate using derivative information. They also show that the momentum can be effectively determined by establishing the relationship between the backpropagation and the conjugate gradient method.

The standard backpropagation technique with momentum is adopted by most researchers. Since there are few systematic ways of selecting the learning rate and momentum simultaneously, the ''best'' values of these learning parameters are usually chosen through experimentation. As the learning rate and the momentum can take on any value between 0 and 1, it is actually impossible to do an exhaustive search to find the best combinations of these training parameters. Only selected values are considered by the researchers. For example, Sharda and Patil (1992) try nine combinations of three learning rates (0.1, 0.5, 0.9) and three momentum values (0.1, 0.5, 0.9).

Tang and Fishwick (1993) conclude that the training parameters play a critical role in the performance of ANNs. Using different learning parameters, they re-test the performance of ANNs for several time series which have been previously reported to have worse results with ANNs. They find that for each of these time series there is an ANN with appropriate learning parameters, which performs significantly better. Tang et al. (1991) also study the effect of training parameters on the ANN learning. They report that high learning rate is good for less complex data and low learning rate with high momentum should be used for more complex data series. However, there are inconsistent conclusions with regard to the best learning parameters (see, for example, Chakraborty et al., 1992; Sharda and Patil, 1992; Tang and Fishwick, 1993), which, in our opinion, are due to the inefficiency and unrobustness of the gradient descent algorithm.

In light of the weakness of the conventional backpropagation algorithm, a number of variations or modifications of backpropagation, such as the adaptive method (Jacobs, 1988; Pack et al., 1991a,b), quickprop (Falhman, 1989), and second-order methods (Parker, 1987; Battiti, 1992; Cottrell et al., 1995) etc., have been proposed. Among them, the second-order methods (such as BFGS and Levenberg-Marquardt methods) are more efficient nonlinear optimization methods and are used in most optimization packages. Their faster convergence, robustness, and the ability to find good local minima make them attractive in ANN training. De Groot and Wurtz (1991) have tested several well-known optimization algorithms such as quasi-Newton, BFGS, Levenberg-Marquardt, and conjugate gradient methods and achieved significant improvements in training time and accuracy for time series forecasting.

Recently, Hung and Denton (1993), and Subramanian and Hung (1993) propose to use a general-

purpose nonlinear optimizer, GRG2 (Lasdon and Waren, 1986), in training the networks. The benefits of GRG2 have been reported in the ANN literature for many different problems (Patuwo et al., 1993; Subramanian and Hung, 1993; Lenard et al., 1995). GRG2 is a widely available optimization software which solves nonlinear optimization problems using the generalized reduced gradient method. With GRG2, there is no need to select learning parameters such as learning rate and momentum. Rather, a different set of parameters, such as stopping criteria, search direction procedure, and the bounds on variables, need to be specified and they can be set at their default values.

Another relevant issue in training an ANN is the specification of an objective function or a cost function. Typically SSE and MSE are used since they are defined in terms of errors. Other objective functions such as maximizing the return, profit or utility may be more appropriate for some problems like financial forecasting. Refenes (1995) (pp. 21–26) shows that the choice of a cost function may significantly influence the network predictive performance if the learning algorithm (backpropagation) and other network parameters are fixed. Thus, one possible way to deal directly with the ultimate objective function is to change the search algorithm from backpropagation type to genetic algorithms, simulated annealing, or other optimization methods which allow search over arbitrary utility functions.

4.4. Data normalization

Nonlinear activation functions such as the logistic function typically have the squashing role in restricting or squashing the possible output from a node to, typically, (0,1) or (−1,1). Data normalization is often performed before the training process begins. As mentioned earlier, when nonlinear transfer functions are used at the output nodes, the desired output values must be transformed to the range of the actual outputs of the network. Even if a linear output transfer function is used, it may still be advantageous to standardize the outputs as well as the inputs to avoid computational problems (Lapedes and Farber, 1988), to meet algorithm requirement (Sharda and Patil, 1992), and to facilitate network learning (Srinivasan et al., 1994).

Four methods for input normalization are summarized by Azoff (1994):

1. Along channel normalization: A channel is defined as a set of elements in the same position over all input vectors in the training or test set. That is, each channel can be thought of as an ''independent'' input variable. The along channel normalization is performed column by column if the input vectors are put into a matrix. In other words, it normalizes each input variable individually.
2. Across channel normalization: This type of normalization is performed for each input vector independently, that is, normalization is across all the elements in a data pattern.
3. Mixed channel normalization: As the name suggests, this method uses some kind of combinations of along and across normalization.
4. External normalization: All the training data are normalized into a specific range.

The choice of the above methods usually depends on the composition of the input vector. For a time series forecasting problem, the external normalization is often the only appropriate normalization procedure. The time lagged observations from the same source are used as input variables and can retain the structure between channels as in the original series. For causal forecasting problems, however, the along channel normalization method should be used since the input variables are typically the independent variables used to predict the dependent variable. Sharda and Patil (1992) use the across channel normalization method for the time series data which may create a serious problem in that the same data in different training patterns are normalized differently and hence valuable information in the underlying structure of the original time series may be lost.

For each type of normalization approach discussed above, the following formulae are frequently used:

- linear transformation to [0,1]: $x_n=(x_0-x_{\min})/(x_{\max}-x_{\min})$ (Lapedes and Farber, 1988);
- linear transformation to $[a,b]$: $x_n=(b-a)(x_0-x_{\min})/(x_{\max}-x_{\min})+a$ (Srinivasan et al., 1994);
- statistical normalization: $x_n=(x_0-\bar{x})/s$ (Weigend et al., 1992);

- simple normalization: $x_n = x_0 / x_{\max}$ (Lachtermacher and Fuller, 1995),

where x_n and x_0 represent the normalized and original data; $x_{\min}$, $x_{\max}$, $\bar{x}$ and s are the minimum, maximum, mean, and standard deviation along the columns or rows, respectively.

It is unclear whether there is a need to normalize the inputs because the arc weights could undo the scaling. There are several studies on the effects of data normalization on the network learning. Shanker et al. (1996) investigate the effectiveness of linear and statistical normalization methods for classification problems. They conclude that, in general, data normalization is beneficial in terms of the classification rate and the mean squared error, but the benefit diminishes as network and sample size increase. In addition, data normalization usually slows down the training process. Engelbrecht et al. (1994) conclude similar results and propose an automatic scaling method called gamma learning rule to allow network self-scaling during the learning process and eliminate the need to normalize the data before training.

Normalization of the output values (targets) is usually independent of the normalization of the inputs. For time series forecasting problems, however, the normalization of targets is typically performed together with the inputs. The choice of range to which inputs and targets are normalized depends largely on the activation function of output nodes, with typically [0, 1] for logistic function and [−1, 1] for hyperbolic tangent function. Several researchers scale the data only to the range of [0.1, 0.9] (Srinivasan et al., 1994) or [0.2, 0.8] (Tang and Fishwick, 1993) based on the fact that the nonlinear activation functions usually have asymptotic limits (they reach the limits only for infinite input values) and the guess that possible outputs may lie, for example, only in [0.1, 0.9], or even [0.2, 0.8] for a logistic function (Azoff, 1994). However, it is easy to see that this is not necessarily true since the output from a logistic node can be as small as 0.000045 or as large as 0.99995 for the net input of only −10 or 10, respectively.

It should be noted that, as a result of normalizing the target values, the observed output of the network will correspond to the normalized range. Thus, to interpret the results obtained from the network, the outputs must be rescaled to the original range. From the user's point of view, the accuracy obtained by the ANNs should be based on the rescaled data set. Performance measures should also be calculated based on the rescaled outputs. However only a few authors clearly state whether the performance measures are calculated on the original or transformed scale.

4.5. Training sample and test sample

As we mentioned earlier, a training and a test sample are typically required for building an ANN forecaster. The training sample is used for ANN model development and the test sample is adopted for evaluating the forecasting ability of the model. Sometimes a third one called the validation sample is also utilized to avoid the overfitting problem or to determine the stopping point of the training process (Weigend et al., 1992). It is common to use one test set for both validation and testing purposes particularly with small data sets. In our view, the selection of the training and test sample may affect the performance of ANNs.

The first issue here is the division of the data into the training and test sets. Although there is no general solution to this problem, several factors such as the problem characteristics, the data type and the size of the available data should be considered in making the decision. It is critical to have both the training and test sets representative of the population or underlying mechanism. This has particular importance for time series forecasting problems. Inappropriate separation of the training and test sets will affect the selection of optimal ANN structure and the evaluation of ANN forecasting performance.

The literature offers little guidance in selecting the training and the test sample. Most authors select them based on the rule of 90% vs. 10%, 80% vs. 20% or 70% vs. 30%, etc. Some choose them based on their particular problems. Gorr et al. (1994) employ a bootstrap resampling design method to partition the whole sample into ten independent subsamples. The estimation of the model is implemented using nine subsamples and then the model is tested with the remaining subsample. Lachtermacher and Fuller (1995) use all the available data for training and use so-called synthetic time series

for test so as to reduce the data requirement in building ANN forecasters. Following the convention in M-competition, the last 18, 8 and 6 points of the data series are often used as test samples for monthly, quarterly and yearly data respectively (Foster et al., 1992; Sharda and Patil, 1992; Tang and Fishwick, 1993). Granger (1993) suggests that for nonlinear forecasting models, at least 20 percent of any sample should be held back for a out-of-sample forecasting evaluation.

Another closely related factor is the sample size. No definite rule exists for the requirement of the sample size for a given problem. The amount of data for the network training depends on the network structure, the training method, and the complexity of the particular problem or the amount of noise in the data on hand. In general, as in any statistical approach, the sample size is closely related to the required accuracy of the problem. The larger the sample size, the more accurate the results will be. Nam and Schaefer (1995) test the effect of different training sample size and find that as the training sample size increases, the ANN forecaster performs better.

Given a certain level of accuracy, a larger sample is required as the underlying relationship between outputs and inputs becomes more complex or the noise in the data increases. However, in reality, sample size is constrained by the availability of data. The accuracy of a particular forecasting problem may be also affected by the sample size used in the training and/or test set.

Note that every model has limits on accuracy it can achieve for real problems. For example, if we consider only two factors: the noise in the data and the underlying model, then the accuracy limit of a linear model such as the Box-Jenkins is determined by the noise in the data and the degree to which the underlying functional form is nonlinear. With more observations, the model accuracy can not improve if there is a nonlinear structure in the data. In ANNs, noise alone determines the limit on accuracy due to its capability of the general function approximation. With a large enough sample, ANNs can model any complex structure in the data. Hence, ANNs can benefit more from large samples than linear statistical models can. It is interesting to note that ANNs do not necessarily require a larger sample than is required by linear models in order to perform well. Kang (1991) finds that ANN forecasting models perform quite well even with sample sizes less than 50 while the Box-Jenkins models typically require at least 50 data points in order to forecast successfully.

4.6. Performance measures

Although there can be many performance measures for an ANN forecaster like the modeling time and training time, the ultimate and the most important measure of performance is the prediction accuracy it can achieve beyond the training data. However, a suitable measure of accuracy for a given problem is not universally accepted by the forecasting academicians and practitioners. An accuracy measure is often defined in terms of the forecasting error which is the difference between the actual (desired) and the predicted value. There are a number of measures of accuracy in the forecasting literature and each has advantages and limitations (Makridakis et al., 1983). The most frequently used are

- the mean absolute deviation (MAD) = $\frac{\Sigma|e_t|}{N}$;
- the sum of squared error (SSE) = $\Sigma(e_t)^2$;
- the mean squared error (MSE) = $\frac{\Sigma(e_t)^2}{N}$;
- the root mean squared error (RMSE) = $\sqrt{\text{MSE}}$;
- the mean absolute percentage error (MAPE) = $\frac{1}{N}\Sigma\left|\frac{e_t}{y_t}\right|$ (100),

where e_t is the individual forecast error; y_t is the actual value; and N is the number of error terms.

In addition to the above, other accuracy measures are also found in the literature. For example, the mean error (ME) is used by Gorr et al. (1994), Theil's U-statistic is tried by Kang (1991) and Hann and Steurer (1996), and the median absolute percentage error (MdAPE) and the geometric mean relative absolute error (GMRAE) are used by Foster et al. (1992). Weigend et al. (1990), (1991), (1992) use the average relative variance (ARV). Cottrell et al. (1995) and De Groot and Wurtz (1991) adopt the residual variance and Akaike information criterion and Bayesian information criterion (BIC).

Because of the limitations associated with each individual measure, one may use multiple perform-

ance measures in a particular problem. However, one method judged to be the best along one dimension is not necessarily the best in terms of other dimensions. The famous M-competition results (Makridakis et al., 1982) consolidate this point. Kang (1991) finds that ANNs do not significantly depend on the performance criteria for simulated data but appear to be dependent on the accuracy measure for actual data.

It is important to note that the first four of the above frequently used performance measures are absolute measures and are of limited value when used to compare different time series. MSE is the most frequently used accuracy measure in the literature. However, the merit of using the MSE is much debated in evaluating the relative accuracy of forecasting methods across different data sets (see, for example, Clements and Hendry (1993) and Armstrong and Fildes (1995)). Furthermore, the MSE defined above may not be appropriate for ANN model building with training sample since it ignores the important information about the number of parameters (arc weights) the model has to estimate. From the point of view of statistics, as the number of estimated parameters in the model goes up, the degrees of freedom for the overall model goes down, thus raising the possibility of overfitting in the training sample. An improved definition of MSE for the training part is the total sum of squared errors divided by the degrees of freedom, which is the number of observations minus the number of arc weights and node biases in an ANN model.

5. The relative performance of ANNs in forecasting

One should note the performance of neural networks in forecasting as compared to the currently widely-used well-established statistical methods. There are many inconsistent reports in the literature on the performance of ANNs for forecasting tasks. The main reason is, as we discussed in the previous section, that a large number of factors including network structure, training method, and sample data may affect the forecasting ability of the networks. For some cases where ANNs perform worse than linear statistical models, the reason may simply be that the data is linear without much disturbance. We can not expect ANNs to do better than linear models for linear relationships. In other cases, it may simply be that the ideal network structure is not used for the data set. Table 2 summarizes the literature on the relative performance of ANNs.

Several papers are devoted to comparing ANNs with the conventional forecasting approaches. Sharda and Patil (1990), (1992) conduct a forecasting competition between ANN models and the Box-Jenkins method using 75 and 111 time series data from the M-competition. They conclude that simple ANN models can forecast as well as the Box-Jenkins method. Tang et al. (1991), and Tang and Fishwick (1993), using both ANN and ARIMA models, analyze several business time series and re-examine 14 series from 75 series used in Sharda and Patil (1990) which are reported to have larger errors. They conclude that ANNs outperform the Box-Jenkins for time series with short memory or with more irregularity. However, for long memory series, both models achieve about the same performance. Kang (1991) obtains similar results in a more systematic study. Kohzadi et al. (1996) compare ANNs with ARIMA models in forecasting monthly live cattle and wheat prices. Their results show that ANNs forecast considerably and consistently more accurately and can capture more turning points than ARIMA models. Hill et al. (1996) compare neural networks with six traditional statistical methods to forecast 111 M-competition time series. Their findings indicate that the neural network models are significantly better than traditional statistical and human judgment methods when forecasting monthly and quarterly data. For the annual data, neural networks and traditional methods are comparable. They also conclude that neural networks are very effective for discontinuous time series. Based on 384 economic and demographic time series, Foster et al. (1992) find that ANNs are significantly inferior to linear regression and a simple average of exponential smoothing methods. Brace et al. (1991) also find that the performance of ANNs is not as good as many other statistical methods commonly used in the load forecasting.

Nelson et al. (1994) discuss the issue of whether ANNs can learn seasonal patterns in a time series. They train networks with both deseasonalized and

Table 2
The relative performance of ANNs with traditional statistical methods

Study	Data	Conclusions
Brace et al. (1991)	8 electric load series (daily)	ANNs are not as good as traditional methods.
Caire et al. (1992)	One electric consumption data (daily)	ANNs are hardly better than ARIMA for 1-step-ahead forecast, but much more reliable for longer step-ahead forecasts.
Chakraborty et al. (1992)	One trivariate price time series (monthly)	ANNs outperform statistical model by at least one order of magnitude.
De Groot and Wurtz (1991)	Sunspots activity time series (yearly)	ANNs are not the best but comparable to the best linear or nonlinear statistical model.
Denton (1995)	Several computer generated data sets	Under ideal situations, ANNs are as good as regression; under less ideal situations, ANNs perform better.
Duliba (1991)	Transportation data (quarterly)	ANNs outperform linear regression model for random effects specification; but worse than the fixed effects specification.
Fishwick (1989)	Ballistic trajectory data	ANNs are worse than linear regression and surface response model.
Foster et al. (1992)	384 economic and demographic time series (quarterly and yearly)	ANNs are significantly inferior to linear regression and simple average of exponential smoothing methods.
Gorr et al. (1994)	Student grade point averages	No significant improvement with ANNs in predicting students' GPAs over linear models.
Hann and Steurer (1996)	Weekly and monthly exchange rate	ANNs outperform the linear models for weekly data and both give almost the same results for monthly data.
Hill et al. (1994) and Hill et al. (1996)	A systematic sample from 111 M-competition time series (monthly, quarterly and yearly)	ANNs are significantly better than statistical and human judgment methods for quarterly and monthly data; about the same for yearly data; ANNs seem to be better in forecasting monthly and quarterly data than in forecasting yearly data.
Kang (1991)	50 M-competition time series	The best ANN model is always better than Box-Jenkins; ANNs perform better as forecasting horizon increases; ANNs need less data to perform as well as ARIMA.
Kohzadi et al. (1996)	Monthly live cattle and wheat prices	ANNs are considerably and consistently better and can find more turning points.
Lachtermacher and Fuller (1995)	4 stationary river flow and 4 nonstationary electricity load time series (yearly)	For stationary time series, ANNs have a slightly better overall performance than traditional methods; for nonstationary series, ANNs are almost much better than ARIMA.
Marquez et al. (1992)	Simulated data for 3 regression models	ANNs perform comparatively as well as regression models.
Nam and Schaefer (1995)	One airline passenger data (monthly)	ANNs are better than time series regression and exponential smoothing.
Refenes (1993)	One exchange rate time series (hourly)	ANNs are much better than exponential smoothing and ARIMA.
Sharda and Patil (1990) and Sharda and Patil (1992)	75 and 111 M-competition time series (monthly, quarterly, and yearly)	ANNs are comparable to Box-Jenkins models.
Srinivasan et al. (1994)	One set of load data	ANNs are better than regression and ARMA models.

Table 2. Continued

Study	Data	Conclusions
Tang et al. (1991)	3 business time series (monthly)	For long memory series, ANNs and ARIMA models are about the same; for short memory series, ANNs are better.
Tang and Fishwick (1993)	14 M-competition time series and 2 additional business time series (monthly and quarterly)	Same as Tang et al. (1991) plus ANNs seem to be better as forecasting horizon increases.
Weigend et al. (1992)	Sunspots activity	ANNs perform better than TAR and bilinear models.
	Exchange rate (daily)	ANNs are significantly better than random walk model.

the raw data, and evaluate them using 68 monthly time series from the M-competition. Their results indicate that the ANNs are unable adequately to learn seasonality and that prior deseasonalization of seasonal time series is beneficial to forecast accuracy. However, Sharda and Patil (1992) conclude that the seasonality of the time series does not affect the performance of ANNs and ANNs are able implicitly to incorporate seasonality.

Several empirical studies find that ANNs seem to be better in forecasting monthly and quarterly time series (Kang, 1991; Hill et al., 1994, 1996) than in forecasting yearly data. This may be due to the fact that monthly and quarterly data usually contain more irregularities (seasonality, cyclicity, nonlinearity, noise) than the yearly data, and ANNs are good at detecting the underlying pattern masked by noisy factors in a complex system.

Tang et al. (1991) and Tang and Fishwick (1993) try to answer the question: under what conditions ANN forecasters can perform better than the traditional time series forecasting methods such as Box-Jenkins models. The first study is based on only three and the second on 16 time series. Their findings are that (1) ANNs perform better as the forecast horizon increases, which is also confirmed by other studies (Kang, 1991; Caire et al., 1992; Hill et al., 1994); (2) ANNs perform better for short memory series (see also Sharda and Patil, 1992); and (3) ANNs give better forecasting results with more input nodes.

Gorr et al. (1994) compare ANNs with several regression models such as linear regression and stepwise polynomial regression in predicting student grade point averages. They do not find any significant statistical difference in the improvement of prediction accuracy among the four methods considered even if there is some evidence of nonlinearities in the data. As the authors discussed, the reasons that their simple ANNs do not perform any better are (1) there are no underlying systematic patterns in the data and/or (2) the full power of the ANNs has not been exploited.

Experimenting with computer generated data for several different experimental conditions, Denton (1995) shows that, under ideal conditions with all regression assumptions, there is little difference in the predictability between ANNs and regression models. However, under less ideal conditions such as outliers, multicollinearity, and model misspecification, ANNs perform better. On the other hand, Hill et al. (1994) report that ANNs are vulnerable to outliers.

Most other researchers also make comparisons between ANNs and the corresponding traditional methods in their particular applications. For example, Fishwick (1989) reports that the performance of ANNs is worse than that of the simple linear regression and the response surface model for a ballistics trajectory function approximation problem. De Groot and Wurtz (1991) compare ANNs with the linear (Box-Jenkins) and nonlinear (bilinear and TAR) statistical models in forecasting the sunspots data. Chakraborty et al. (1992) contrast their ANNs with the multivariate ARMA model for a multivariate price time series. Weigend et al. (1992) study the sunspots activity and exchange rate forecasting problems with ANN and other traditional methods

popular in these areas. Refenes (1993) compares his ANN model with exponential smoothing and ARIMA model using hourly tick data of exchange rate. Srinivasan et al. (1994) evaluate the performance of ANNs in forecasting short-term electrical load and compare it to popular traditional methods of linear regression and ARIMA models. In forecasting exchange rate, Hann and Steurer (1996) find that ANNs outperform the linear models when weekly data are used and if monthly data are used, ANNs and linear methods yield similar results.

6. Conclusions and the future

We have presented a review of the current state of the use of artificial neural networks for forecasting application. This review is comprehensive but by no means exhaustive, given the fast growing nature of the literature. The important findings are summarized as follows:

- The unique characteristics of ANNs – adaptability, nonlinearity, arbitrary function mapping ability – make them quite suitable and useful for forecasting tasks. Overall, ANNs give satisfactory performance in forecasting.
- A considerable amount of research has been done in this area. The findings are inconclusive as to whether and when ANNs are better than classical methods.
- There are many factors that can affect the performance of ANNs. However, there are no systematic investigations of these issues. The shotgun (trial-and-error) methodology for specific problems is typically adopted by most researchers, which is the primary reason for inconsistencies in the literature.

ANNs offer a promising alternative approach to traditional linear methods. However, while ANNs provide a great deal of promises, they also embody a large degree of uncertainty. There are several unsolved mysteries in this area. Since most results are based on limited empirical studies, the words ''seem'' and ''appear'' are used quite commonly in the literature. Few theoretical results are established in this area. Many important research questions still remain. Among them:

(1) How do ANNs model the autocorrelated time series data and produce better results than conventional linear and nonlinear methods?

(2) Given a specific forecasting problem, how do we systematically build an appropriate network that is best suited for the problem?

(3) What is the best training method or algorithm for forecasting problems, particularly time series forecasting problems?

(4) How should we go about designing the sampling scheme, pre- and post-processing the data? What are the effects of these factors on the predictive performance of ANNs?

These problems are not easy to tackle. However, given too many factors which could affect the performance of the ANN method, limited empirical study may not be sufficient to address all the issues.

Like statistical models, ANNs have weaknesses as well as strengths. While ANNs have many desired features which make them quite suitable for a variety of problem areas, they will never be the panacea. There cannot be a universal model that will predict everything well for all problems (Gershenfeld and Weigend, 1993). Indeed there will probably not be a single best forecasting method for all situations (Bowerman and O'Connell, 1993). The mixed results of M-competition as well as the results from Section 5 of this paper give clear evidence. ANNs' capabilities make them potentially valuable for some forecasting problems, but not for others. Gorr (1994) believes that ANNs can be more appropriate for the following situations: (1) large data sets; (2) problems with nonlinear structure; (3) the multivariate time series forecasting problems.

To best utilize ANNs for forecasting problems as well as other tasks, it is important to understand the limitations of ANNs and what they can do as well as what they cannot do. Several points need to be emphasized:

- ANNs are nonlinear methods per se. For static linear processes with little disturbance, they may not be better than linear statistical methods.
- ANNs are black-box methods. There is no explicit form to explain and analyze the relationship between inputs and outputs. This causes difficulty

in interpreting results from the networks. Also no formal statistical testing methods can be used for ANNs.
- ANNs are prone to have overfitting problems due to their typical, large parameter set to be estimated.
- There are no structured methods today to identify what network structure can best approximate the function, mapping the inputs to outputs. Hence, the tedious experiments and trial-and-error procedures are often used.
- ANNs usually require more data and computer time for training.

Overall, there may be a limit on what ANNs can learn from the data and make predictions. This limitation of ANNs may come from their nonparametric property (Geman et al., 1992). Since they are data-driven and model-free, ANNs are quite general but can suffer high variance in the estimation, that is, they may be too dependent on the particular samples observed. On the other hand, the model-based methods such as Box-Jenkins are bias-prone. They are likely to be incorrect for the task on hand. It is clear that ANNs as well as traditional linear and nonlinear methods can not do everything equally well.

Given the current status of the ANN forecasting, what will be the future of the area? Will it be a passing fad (Chatfield, 1993)? Many successful forecasting applications of ANNs suggest that they can be one of the very useful tools in forecasting researchers' and practitioners' arsenal. Yet as Amari (1994) comments, without theoretical foundations, it is difficult for ANN technology to take off from the current rather ''easy and shallow'' technology to a more fundamental one. Furthermore, the current ANN model building needs lengthy experimentation and tinkering which is a major roadblock for the extensive use of the method. New modeling methodology is needed to ease model building and to make ANNs more acceptable to forecasting practitioners. Recently, fuzzy expert system (Bakirtzis et al., 1995; Dash et al., 1995; Kim et al., 1995; Bataineh et al., 1996) and wavelet analysis (Zhang and Benveniste, 1992; Delyon et al., 1995; Zhang et al., 1995; Szu et al., 1996; Yao et al., 1996) have been proposed as supplementary tools to ANNs. They can aid ANNs to extract the features of time series data and make better forecasting. Particularly, the discovery of wavelet functions that serve as basis functions has reawakened the interest in the time–frequency analysis of nonstationary nonlinear time series (Rioul and Vetterli, 1991; Wallich, 1991). As space, time, frequency, and phase are the four mathematical domains by which signals or indicators about time series are analyzed, wavelet networks which combine the wavelet theory and ANNs can be very promising tools for understanding nonlinear nonstationary processes. We believe that the future of ANN forecasting will be even brighter as more and more research efforts are devoted to this area.

Acknowledgements

We would like to thank Dr. Pelikan, the associate editor, and three anonymous referees for their constructive and helpful comments.

References

Aiken, M., Krosp, J., Vanjani, M., Govindarajulu, C., Sexton, R., 1995. A neural network for predicting total industrial production. Journal of End User Computing 7 (2), 19–23.

Akaike, H., 1974. A new look at the statistical model identification. IEEE Transactions on Automatic Control 19 (6), 716–723.

Amari, S., 1994. A comment on ''Neural networks: A review from a statistical perspective''. Statistical Science 9 (1), 31–32.

Amirikian, B., Nishimura, H., 1994. What size network is good for generalization of a specific task of interest?. Neural Networks 7 (2), 321–329.

Arizmendi, C.M., Sanchez, J.R., Ramos, N.E., Ramos, G.I., 1993. Time series prediction with neural nets: Application to airborne pollen forecasting. International Journal of Biometeorology 37, 139–144.

Armstrong, J.S., Fildes, R., 1995. Correspondence: On the selection of error measures for comparisons among forecasting methods. Journal of Forecasting 14, 67–71.

Azoff, E.M., 1994. Neural Network Time Series Forecasting of Financial Markets. John Wiley and Sons, Chichester.

Bacha, H., Meyer, W., 1992. A neural network architecture for load forecasting. In: Proceedings of the IEEE International Joint Conference on Neural Networks, 2, pp. 442–447.

Bakirtzis, A.G., Theocharis, J.B., Kiartzis, S.J., Satsios, K.J., 1995. Short term load forecasting using fuzzy neural networks. IEEE Transactions on Power Systems 10 (3), 1518–1524.

Balestrino, A., Bini Verona, F., Santanche, M., 1994. Time series analysis by neural networks: Environmental temperature forecasting. Automazione e Strumentazione 42 (12), 81–87.

Barker, D., 1990. Analyzing financial health: Integrating neural networks and expert systems. PC AI 4 (3), 24–27.

Barron, A.R., 1994. A comment on ''Neural networks: A review from a statistical perspective''. Statistical Science 9 (1), 33–35.

Bataineh, S., Al-Anbuky, A., Al-Aqtash, S., 1996. An expert system for unit commitment and power demand prediction using fuzzy logic and neural networks. Expert Systems 13 (1), 29–40.

Battiti, R., 1992. First- and second-order methods for learning: Between steepest descent and Newton's method. Neural Computation 4 (2), 141–166.

Baum, E.B., Haussler, D., 1989. What size net gives valid generalization?. Neural Computation 1, 151–160.

Bergerson, K., Wunsch, D.C., 1991. A commodity trading model based on a neural network–expert system hybrid. In: Proceedings of the IEEE International Conference on Neural Networks, Seattle, WA, pp. 1289–1293.

Borisov, A.N., Pavlov, V.A., 1995. Prediction of a continuous function with the aid of neural networks. Automatic Control and Computer Sciences 29 (5), 39–50.

Bowerman, B.L., O'Connell, R.T., 1993. Forecasting and Time Series: An Applied Approach, 3rd ed. Duxbury Press, Belmont, CA.

Box, G.E.P., Jenkins, G.M., 1976. Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco, CA.

Brace, M.C., Schmidt, J., Hadlin, M., 1991. Comparison of the forecasting accuracy of neural networks with other established techniques. In: Proceedings of the First Forum on Application of Neural Networks to Power Systems, Seattle, WA, pp. 31–35.

Caire, P., Hatabian, G., Muller, C., 1992. Progress in forecasting by neural networks. In: Proceedings of the International Joint Conference on Neural Networks, 2, pp. 540–545.

Chakraborty, K., Mehrotra, K., Mohan, C.K., Ranka, S., 1992. Forecasting the behavior of multivariate time series using neural networks. Neural Networks 5, 961–970.

Chan, D.Y.C., Prager, D., 1994. Analysis of time series by neural networks. In: Proceedings of the IEEE International Joint Conference on Neural Networks, 1, pp. 355–360.

Chan, W.S., Tong, H., 1986. On tests for non-linearity in time series analysis. Journal of Forecasting 5, 217–228.

Chang, I., Rapiraju, S., Whiteside, M., Hwang, G., 1991. A neural network to time series forecasting. In: Proceedings of the Decision Science Institute., 3, pp. 1716–1718.

Chatfield, C., 1993. Neural networks: Forecasting breakthrough or passing fad?. International Journal of Forecasting 9, 1–3.

Chen, C.H., 1994. Neural networks for financial market prediction. In: Proceedings of the IEEE International Conference on Neural Networks, 2, pp. 1199–1202.

Chen, S.T., Yu, D.C., Moghaddamjo, A.R., 1991. Weather sensitive short-term load forecasting using nonfully connected artificial neural network. In: Proceedings of the IEEE/Power Engineering Society Summer Meeting, 91 SM 449–8 PWRS.

Chen, T., Chen, H., 1995. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. IEEE Transactions on Neural Networks 6 (4), 911–917.

Cheng, B., Titterington, D.M., 1994. Neural networks: A review from a statistical perspective. Statistical Science 9 (1), 2–54.

Chester, D.L., 1990. Why two hidden layers are better than one? In: Proceedings of the International Joint Conference on Neural Networks, pp. 1265–1268.

Cheung, K.H., Szeto, K.Y., Tam, K.Y., 1996. Maximum-entropy approach to identify time-series lag structure for developing intelligent forecasting systems. International Journal of Computational Intelligence and Organization 1 (2), 94–106.

Chiang, W.-C., Urban, T.L., Baldridge, G.W., 1996. A neural network approach to mutual fund net asset value forecasting. Omega 24, 205–215.

Chng, E.S., Chen, S., Mulgrew, B., 1996. Gradient radial basis function networks for nonlinear and nonstationary time series prediction. IEEE Transactions on Neural Networks 7 (1), 190–194.

Chu, C.H., Widjaja, D., 1994. Neural network system for forecasting method selection. Decision Support Systems 12, 13–24.

Clements, M.P., Hendry, D.F., 1993. On the limitations of comparing mean square forecast errors. Journal of Forecasting 12, 615–637.

Coleman, K.G., Graettinger, T.J., Lawrence, W.F., 1991. Neural networks for bankruptcy prediction: The power to solve financial problems. AI Review 5, July/August, 48–50.

Connor, J.T., Martin, R.D., Atlas, L.E., 1994. Recurrent neural networks and robust time series prediction. IEEE Transaction on Neural Networks 51 (2), 240–254.

Cottrell, M., Girard, B., Girard, Y., Mangeas, M., Muller, C., 1995. Neural modeling for time series: a statistical stepwise method for weight elimination. IEEE Transactions on Neural Networks 6 (6), 1355–1364.

Cromwell, J.B., Labys, W.C., Terraza, M., 1994. Univariate Tests for Time Series Models. Saga Publications, Thousand Oaks.

Cybenko, G., 1988. Continuous Valued Neural Networks with Two Hidden Layers are Sufficient. Technical Report, Tuft University.

Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. Mathematical Control Signals Systems 2, 303–314.

Dash, P.K., Ramakrishna, G., Liew, A.C., Rahman, S., 1995. Fuzzy neural networks for time-series forecasting of electric load. IEE Proceedings – Generation, Transmission and Distribution 142 (5), 535–544.

De Gooijer, J.G., Kumar, K., 1992. Some recent developments in non-linear time series modelling, testing, and forecasting. International Journal of Forecasting 8, 135–156.

De Groot, C., Wurtz, D., 1991. Analysis of univariate time series with connectionist nets: a case study of two classical examples. Neurocomputing 3, 177–192.

Delyon, B., Juditsky, A., Benveniste, A., 1995. Accuracy analysis for wavelet approximations. IEEE Transactions on Neural Networks 6 (2), 332–348.

Denton, J.W., 1995. How good are neural networks for causal forecasting? The Journal of Business Forecasting 14 (2), Summer, 17–20.

Deppisch, J., Bauer, H.-U., Geisel, T., 1991. Hierarchical training of neural networks and prediction of chaotic time series. Physics Letters 158, 57–62.

Donaldson, R.G., Kamstra, M., 1996. Forecasting combining with neural networks. Journal of Forecasting 15, 49–61.

Duliba, K.A., Contrasting neural nets with regression in predicting performance in the transportation industry. In: Proceedings of the Annual IEEE International Conference on Systems Sciences., 25, pp. 163–170.

Dutta, S., Shekhar, S., 1988. Bond rating: A non-conservative application of neural networks. In: Proceedings of the IEEE International Conference on Neural Networks. San Diego, California, 2, pp. 443–450.

El-Sharkawi, M.A., Oh, S., Marks, R.J., Damborg, M.J., Brace, C.M., 1991. Short-term electric load forecasting using an adaptively trained layered perceptron. In: Proceedings of the 1st International Forum on Application of Neural Networks to Power Systems, 3–6.

Engelbrecht, A.P., Cloete, I., Geldenhuys, J., Zurada, J.M., 1994. Automatic scaling using gamma learning for feedforward neural networks. In: Anderson, D.Z. (Ed.), Neural Information Processing Systems, American Institute of Physics, New York, pp. 374–381.

Engle, R.F., 1982. Autoregressive conditional heteroskedasticity with estimates of the variance of UK inflation. Econometrica 50, 987–1008.

Ezugwu, E.O., Arthur, S.J., Hins, E.L., 1995. Too-wear prediction using artificial neural networks. Journal of Materials Processing Technology 49, 255–264.

Falhman, S., 1989. Faster-learning variations of back-propagation: An empirical study. In: Touretzky, D., Hinton, G., Sejnowski, T., (Eds.), Proceedings of the 1988 Connectionist Models Summer School, pp. 38–51.

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

Fletcher, D., Goss, E., 1993. Forecasting with neural networks – An application using bankruptcy data. Information and Management 24, 159–167.

Fletcher, R., 1987. Practical Methods of Optimization, 2nd ed. John Wiley, Chichester.

Foster, W.R., Collopy, F., Ungar, L.H., 1992. Neural network forecasting of short, noisy time series. Computers and Chemical Engineering 16 (4), 293–297.

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

Gately, E., 1996. Neural Networks for Financial Forecasting. John Wiley, New York.

Geman, S., Bienenstock, E., Doursat, T., 1992. Neural networks and the bias/variance dilemma. Neural Computation 5, 1–58.

Gent, C.R., Sheppard, C.P., 1992. Predicting time series by a fully connected neural network trained by back propagation. Computing and Control Engineering Journal 3 (3), May, 109–112.

Gershenfeld, N.A., Weigend, A.S., 1993. The future of time series: learning and understanding. In: Weigend, A.S., Gershenfeld, N.A. (Eds.), Time Series Prediction: Forecasting the Future and Understanding the Past. Addison-Wesley, Reading, MA, pp. 1–70.

Ginzburg, I., Horn, D., 1991. Learnability of time series. In: Proceedings of the IEEE International Joint Conference on Neural Networks, 3, pp. 2653–2657.

Ginzburg, I., Horn, D., 1992. Learning the rule of a time series. International Journal of Neural Systems 3 (2), 167–177.

Ginzburg, I., Horn, D., 1994. Combined neural networks for time series analysis. Advances in Neural Information Processing Systems 6, 224–231.

Gorr, L., 1994. Research prospective on neural network forecasting. International Journal of Forecasting 10, 1–4.

Gorr, W.L., Nagin, D., Szczypula, J., 1994. Comparative study of artificial neural network and statistical models for predicting student grade point averages. International Journal of Forecasting 10, 17–34.

Granger, C.W.J., 1993. Strategies for modelling nonlinear time-series relationships. The Economic Record 69 (206), 233–238.

Granger, C.W.J., Anderson, A.P., 1978. An Introduction to Bilinear Time Series Models. Vandenhoeck and Ruprecht, Göttingen.

Granger, C.W.J., Terasvirta, T., 1993. Modelling Nonlinear Economic Relationships. Oxford University Press, Oxford.

Grudnitski, G., Osburn, L., 1993. Forecasting S and P and gold futures prices: An application of neural networks. The Journal of Futures Markets 13 (6), 631–643.

Guo, Z., Uhrig, R., 1992. Using genetic algorithm to select inputs for neural networks. In: Proceedings of the Workshop on Combinations of Genetic Algorithms and Neural Networks, COGANN92, pp. 223–234.

Haas, D.J., Milano, J., Flitter, L., 1995. Prediction of helicopter component loads using neural networks. Journal of the American Helicopter Society 40 (1), 72–82.

Hammerstrom, D., 1993. Neural networks at work, IEEE Spectrum, June, 26–32.

Hann, T.H., Steurer, E., 1996. Much ado about nothing? Exchange rate forecasting: Neural networks vs. linear models using monthly and weekly data. Neurocomputing 10, 323–339.

Happel, B.L.M., Murre, J.M.J., 1994. The design and evolution of modular neural network architectures. Neural Networks 7, 985–1004.

Hecht-Nielsen, R., 1990. Neurocomputing. Addison-Wesley, Menlo Park, CA.

Hertz, J., Krogh, A., Palmer, R.G., 1991. Introduction to the Theory of Neural Computation. Addison-Wesley, Reading, MA.

Hill, T., Marquez, L., O'Connor, M., Remus, W., 1994. Artificial neural networks for forecasting and decision making. International Journal of Forecasting 10, 5–15.

Hill, T., O'Connor, M., Remus, W., 1996. Neural network models for time series forecasts. Management Sciences 42 (7), 1082–1092.

Hinich, M.J., 1982. Testing for Gaussianity and linearity of a statistionary time series. Journal of Time Series Analysis 3, 169–176.

Hinton, G.E., 1992. How neural networks learn from experience, Scientific American, September, 145–151.

Ho, K.L., Hsu, Y.Y., Yang, C.C., 1992. Short term load forecasting using a multilayer neural network with an adaptive learning algorithm. IEEE Transactions on Power Systems 7 (1), 141–149.

Hopfield, J.J., 1982. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of the Sciences of the U.S.A. 79, 2554–2558.

Hornik, K., 1991. Approximation capabilities of multilayer feedforward networks. Neural Networks 4, 251–257.

Hornik, K., 1993. Some new results on neural network approximation. Neural Networks 6, 1069–1072.

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Networks 2, 359–366.

Hsu, Y.Y., Yang, C.C., 1991. Design of artificial neural networks for short-term load forecasting, Part I: selforganising feature maps for day type identification. IEE Proceedings-C: Generation, Transmission and Distribution 138 (5), 407–413.

Hsu, Y.Y., Yang, C.C., 1991. Design of artificial neural networks for short-term load forecasting, Part II: Multilayer feedforward networks for peak load and valley load forecasting. IEE Proceedings- C; Generation, Transmission and Distribution 138 (5), 414–418.

Hu, M.J.C., 1964. Application of the adaline system to weather forecasting. Master Thesis, Technical Report 6775-1, Stanford Electronic Laboratories, Stanford, CA, June.

Hung, M.S., Denton, J.W., 1993. Training neural networks with the GRG2 nonlinear optimizer. European Journal of Operational Research 69, 83–91.

Huntley, D.G., 1991. Neural nets: An approach to the forecasting of time series. Social Science Computer Review 9 (1), 27–38.

Hush, D.R., Horne, B.G., 1993. Progress in supervised neural networks: What's new since Lippmann? IEEE Signal Processing Magazine, January, 8–38.

Hwang, J.N. and S. Moon, 1991. Temporal difference method for multi-step prediction: Application to power load forecasting. In: Proceedings of the First Forum on Application of Neural Networks to Power Systems, pp. 41–45.

Irie, B., Miyake, S., 1988. Capabilities of three-layered perceptrons. In: Proceedings of the IEEE International Conference on Neural Networks, I, pp. 641–648.

Jacobs, R.A., 1988. Increased rates of convergence through learning rate adaptation. Neural Networks 1 (4), 295–308.

Jhee, W.C., Lee, K.C., Lee, J.K., 1992. A neural network approach for the identification of the Box-Jenkins model. Network: Computation in Neural Systems 3, 323–339.

Jones, A.J., 1993. Genetic algorithms and their applications to the design of neural networks. Neural Computing and Applications 1, 32–45.

Jones, R.D., Lee, Y.C., Barnes, C.W., Flake, G.W., Lee, K., Lewis, P.S., et al., 1990. Function approximation and time series prediction with neural networks. In: Proceedings of the IEEE International Joint Conference on Neural Networks, San Diego, CA, 1, pp. 649–665

Kaastra, I., Boyd, M.S., 1995. Forecasting futures trading volume using neural networks. The Journal of Futures Markets 15 (8), 953–970.

Kang, S., 1991. An Investigation of the Use of Feedforward Neural Networks for Forecasting. Ph.D. Thesis, Kent State University.

Karnin, E.D., 1990. A simple procedure for pruning back-propagation trained neural networks. IEEE Transactions on Neural Networks 1 (2), 239–245.

Karunanithi, N., Grenney, W.J., Whitley, D., Bovee, K., 1994. Neural networks for river flow prediction. Journal of Computing in Civil Engineering 8 (2), 201–220.

Keenan, D.M., 1985. A Turkey nonadditivity-type test for time series nonlinearity. Biometrika 72 (1), 39–44.

Kiartzis, S.J., Bakirtzis, A.G., Petridis, V., 1995. Short-term load forecasting using neural networks. Electric Power Systems Research 33, 1–6.

Kim, K.H., Park, J.K., Hwang, K.J., Kim, S.H., 1995. Implementation of hybrid short-term load forecasting system using artificial neural networks and fuzzy expert systems. IEEE Transactions on Power Systems 10 (3), 1534–1539.

Kimoto, T., Asakawa, K., Yoda, M., Takeoka, M., 1990. Stock Market prediction system with modular neural networks. In: Proceedings of the IEEE International Joint Conference on Neural Networks. San Diego, California, 2, pp. 11–16.

Klimasauskas, C.C., 1991. Applying neural networks. Part 3: Training a neural network, PC-AI, May/June, 20–24.

Kohonen, T., 1982. Self-organized formation of topologically correct feature maps. Biological Cybernetics 43, 59–69.

Kohzadi, N., Boyd, M.S., Kermanshahi, B., Kaastra, I., 1996. A comparison of artificial neural network and time series models for forecasting commodity prices. Neurocomputing 10, 169–181.

Kryzanowski, L., Galler, M., Wright, D.W., 1993. Using artificial neural networks to pick stocks. Financial Analysts Journal, July/August, 21–27.

Kuan, C.-M., Liu, T., 1995. Forecasting exchange rates using feedforward and recurrent neural networks. Journal of Applied Econometrics 10, 347–364.

Kuan, C.-M., White, H., 1994. Artificial neural networks: an economic perspective. Economic Reviews 13 (1), 1–91.

Lachtermacher, G., Fuller, J.D., 1995. Backpropagation in time-series forecasting. Journal of Forecasting 14, 381–393.

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

Lapedes, A., Farber, R., 1988. How neural nets work. In: Anderson, D.Z., (Ed.), Neural Information Processing Systems, American Institute of Physics, New York, pp. 442–456.

Lasdon, L.S., Waren, A.D., 1986. GRG2 User's Guide, School of Business Administration, University of Texas at Austin.

Lee, J.K., Jhee, W.C., 1994. A two-stage neural network approach for ARMA model identification with ESACF. Decision Support Systems 11, 461–479.

Lee, K.Y., Cha, Y.T., Ku, C.C., 1991. A study on neural networks for short-term load forecasting. In: Proceedings of the First Forum on Application of Neural Networks to Power Systems, Seattle, WA, pp. 26–30.

Lee, K.Y., Park, J.H., 1992. Short-term load forecasting using an artificial neural network. IEEE Transactions on Power Systems 7 (1), 124–132.

Lee, T.W., White, H., Granger, C.W.J., 1993. Testing for neglected nonlinearity in time series meodels. Journal of Econometrics 56, 269–290.

Lenard, M.J., Alam, P., Madey, G.R., 1995. The application of neural networks and a qualitative response model to the auditor's going concern uncertainty decision. Decision Science 26 (2), 209–226.

Levin, A.U., Narendra, K.S., 1993. Control of nonlinear dynamical systems using neural networks: Controllability and Stabilization. IEEE Transactions on Neural Networks 4 (2), 192–206.

Li, M., Mehrotra, K., Mohan, C., Ranka, S., 1990. Sunspots numbers forecasting using neural networks. In: Proceedings of the 5th IEEE International Symposium on Intelligent Control, pp. 524–529.

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

Lowe, D., Webb, A.R., 1990. Time series prediction by adaptive networks: A dynamical systems perspective. IEE proceedings-F 138 (1), 17–24.

Lubero, R.G., 1991. Neural networks for water demand time series forecasting. In: Proceedings of the International Workshop on Artificial Neural Networks, pp. 453–460.

Luukkonen, R., Saikkonen, P., Terasirta, T., 1988. Testing linearity in univariate time series models. Scandinavian Journal of Statistics 15, 161–175.

Maasoumi, E., Khotanzad, A., Abaye, A., 1994. Artificial neural networks for some macroeconomic series: A first report. Econometric Reviews 13 (1), 105–122.

Makridakis, S., Anderson, A., Carbone, R., Fildes, R., Hibdon, M., Lewandowski, R. et al., 1982. The accuracy of extrapolation (time series) methods: Results of a forecasting competition. Journal of Forecasting 1 (2), 111–153.

Makridakis, S., Wheelwright, S.C., McGee, V.E., 1983. Forecasting: Methods and Applications, 2nd ed. John Wiley, New York.

Marquez, L., Hill, T., O'Connor, M., Remus, W., 1992. Neural network models for forecast a review. In: IEEE proceedings of the 25th Hawaii International Conference on System Sciences., 4, pp. 494–498.

Masson, E., Wang, Y.-J., 1990. Introduction to computation and learning in artificial neural networks. European Journal of Operational Research 47, 1–28.

McLeod, A.I., Li, W.K., 1983. Diagnostic checking ARMA time series models using squared residual autocorrelations. Journal of Time Series Analysis 4, 169–176.

Miller, G.F., Todd, P.M., Hegde, S.U., 1989. Designing neural networks using genetic algorithms. In: Schaffer, J.D. (Ed.), Proceedings of the Third International Conference on Genetic Algorithms. Morgon Kaufman, San Francisco, pp. 370–384.

Muller, C., Mangeas, M., 1993. Neural networks and time series forecasting: a theoretical approach. In: IEEE Systems, Man, and Cybernetics Conference Proceedings, pp. 590–594.

Murata, N., Yoshizawa, S., Amari, S., 1994. Network information criterion-determining the number of hidden units for an artificial neural network model. IEEE Transactions on Neural Networks 5 (6), 865–872.

Nam, K., Schaefer, T., 1995. Forecasting international airline passenger traffic using neural networks. Logistics and Transportation 31 (3), 239–251.

Narendra, K.S., Parthasarathy, K., 1990. Identification and control of dynamical systems using neural networks. IEEE Transactions on Neural Networks 1 (1), 4–27.

Nelson, M., Hill, T., Remus, B., O'Connor, M., 1994. Can neural networks be applied to time series forecasting and learn seasonal patterns: An empirical investigation. In: Proceedings of the Twenty seventh Annual Hawaii International Conference on System Sciences, pp. 649–655.

Odom, M.D., Sharda, R., 1990. A neural network model for bankruptcy prediction. In: Proceedings of the IEEE International Joint Conference on Neural Networks. San Diego, CA, 2, pp. 163–168

Pack, D.C., El-Sharkawi, M.A., Marks II, R.J., 1991a. An adaptively trained neural network. IEEE Transactions on Neural Networks 2 (3), 334–345.

Pack, D.C., El-Sharkawi, M.A., Marks II, R.J., Atlas, L.E., Damborg, M.J., 1991b. Electric load forecasting using an artificial neural network. IEEE Transactions on Power Systems 6 (2), 442–449.

Pankratz, A., 1983. Forecasting with Univariate Box-Jenkins Models: Concepts and Cases. John Wiley, New York.

Park, J., Sandberg, I.W., 1991. Universal approximation using radial basis function networks. Neural Computation 3, 246–257.

Park, J., Sandberg, I.W., 1993. Approximation and radial basis function networks. Neural Computation 5, 305–316.

Parker, D.B., 1987. Optimal algorithm for adaptive networks: Second order back propagation, second order direct propagation, and second order Hebbian learning. In: Proceedings of the IEEE International Conference on Neural Networks, 2, pp. 593–600.

Patuwo, E., Hu, M.Y., Hung, M.S., 1993. Two-group classification using neural networks. Decision Science 24 (4), 825–845.

Payeur, P., Le-Huy, H., Gosselin, C.M., 1995. Trajectory prediction for moving objects using artificial neural networks. IEEE Transactions on Industrial Electronic 42 (2), 147–158.

Pelikan, E., de Groot, C., Wurtz, D., 1992. Power consumption in West-Bohemia: Improved forecasts with decorrelating connectionist networks. Neural Network World 2 (6), 701–712.

Peng, T.M., Hubele, N.F., Karady, G.G., 1992. Advancement in the application of neural networks for short-term load forecasting. IEEE Transactions on Power Systems 7 (1), 250–257.

Poli, I., Jones, R.D., 1994. A neural net model for prediction. Journal of American Statistical Association 89 (425), 117–121.

Reed, R., 1993. Pruning algorithms – A survey. IEEE Transactions on Neural Networks, 4 (5), 740–747.

Refenes, A.N., 1993. Constructive learning and its application to currency exchange rate forecasting. In: Trippi, R.R., Turban, E. (Eds.), Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real-World Performance. Probus Publishing Company, Chicago.

Refenes, A.N., 1995. Neural Networks in the Capital Markets. John Wiley, Chichester.

Refenes, A.N., Zapranis, A., Francis, G., 1994. Stock performance modeling using neural networks: A comparative study with regression models. Neural Networks 7 (2), 375–388.

Reilly, D.L., Cooper, L.N., 1990. An overview of neural networks: early models to real world systems. In: Zornetzer, S.F., Davis, J.L., Lau, C. (Eds.), An Introduction to Neural and Electronic Networks. Academic Press, New York, pp. 227–248.

Reynolds, S.B., 1993. A Neural Network Approach to Box-Jenkins Model Identification. Ph.D. Thesis, University of Alabama.

Reynolds, S.B., Mellichamp, J.M., Smith, R.E., 1995. Box-Jenkins forecast model identification. AI Expert, June, 15–28.

Ricardo, S.Z., Guedes, K., Vellasco M., Pacheco, M.A., 1995. Short-term load forecasting using neural nets. In: Mira, J., Sandoval, F. (Eds.), From Natural to Artificial Neural Computation. Springer, Berlin, pp. 1001–1008.

Rioul, O., Vetterli, M., 1991. Wavelet and signal processing. IEEE Signal Processing Magazine 8 (4), 14–38.

Ripley, B.D., 1993. Statistical aspects of neural networks. In: Barndorff-Nielsen, O.E., Jensen, J.L., Kendall, W.S. (Eds.), Networks and Chaos-Statistical and Probabilistic Aspects. Chapman and Hall, London, pp. 40–123.

Rissanen, J., 1987. Stochastic complexity (with discussion). Journal of the Royal Statistical Society, B, 49, 223–239 and 252–265.

Rosen, B.E., 1993. Neural network moving averages for time series prediction. In: SPIE, Vol. 1966, Science of Artificial Neural Networks, 2, 448–456.

Roy, A., Kim, L.S., Mukhopadhyay, S., 1993. A polynomial time algorithm for the construction and training of a class of multilayer perceptrons. Neural Networks 6, 535–545.

Ruiz-Suarez, J.C., Mayora-Ibarra, O.A., Torres-Jimenez, J., Ruiz-Suarez, L.G., 1995. Short-term ozone forecasting by artificial neural networks. Advances in Engineering Software 23, 143–149.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by backpropagating errors. Nature 323 (6188), 533–536.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning internal representation by back- propagating errors. In: Rumelhart, D.E., McCleland, J.L., the PDP Research Group (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition. MIT Press, MA.

Rumelhart, D.E., Widrow, B., Lehr, M.A., 1994. The basic ideas in neural networks. Communications of the ACM 37 (3), 87–92.

Rumelhart, D.E., Durbin, R., Golden, R., Chauvin, Y., 1995. Backpropagation: the basic theory. In: Chauvin, Y., Rumelhart, D.E. (Eds.), Backpropagation: Theory, Architectures, and Applications. Lawrence Erlbaum Associates, New Jersey, pp. 1–34.

Saikkonen, P., Luukkonen, R., 1988. Lagrange multiplier tests for testing non-linearities in time series models. Scandinavian Journal of Statistics 15, 55–68.

Salchenkerger, L.M., Cinar, E.M., Lash, N.A., 1992. Neural networks: A new tool for predicting thrift failures. Decision Science 23 (4), 899–916.

Schiffmann, W., Joost, M., Werner, R., 1993. Application of genetic algorithms to the construction of topologies for multilayer perceptron. In: Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms, pp. 675–682.

Schoneburg, E., 1990. Stock price prediction using neural networks: A project report. Neurocomputing 2, 17–27.

Sen, T.K., Oliver, R.J., Sen, N., 1992. Predicting corporate mergers using backpropagating neural networks: A comparative study with logistic models. Working paper, The R.B. Pamplin College of Business, Virginia Tech, Blacksburg, VA.

Shanker, M., Hu, M.Y., Hung, M.S., 1996. Effect of data standardization on neural network training. Omega 24 (4), 385–397.

Sharda, R., 1994. Neural networks for the MS/OR analyst: An application bibliography. Interfaces 24 (2), 116–130.

Sharda, R., Patil, R.B., 1990. Neural networks as forecasting experts: An empirical test. In: Proceedings of the International Joint Conference on Neural Networks. Washington, D.C., 2, pp. 491–494.

Sharda, R., Patil, R.B., 1992. Connectionist approach to time series prediction: An empirical test. Journal of Intelligent Manufacturing 3, 317–323.

Shin, Y., Ghosh, J., 1995. Ridge polynomial networks. IEEE Transactions on Neural Networks 6 (3), 610–622.

Sietsma, J., Dow, R., 1988. Neural net pruning–Why and how? In: Proceedings of the IEEE International Conference on Neural Networks, 1, pp. 325–333.

Smith, M., 1993. Neural Networks for Statistical Modeling. Van Nostrand Reinhold, New York.

Sohl, J.E., Venkatachalam, A.R., 1995. A neural network approach to forecasting model selection. Information and Management 29, 297–303.

Srinivasan, D., Liew, A.C., Chang, C.S., 1994. A neural network short-term load forecaster. Electric Power Systems Research 28, 227–234.

Subramanian, V., Hung, M.S., 1993. A GRG2-based system for training neural networks: Design and computational experience. ORSA Journal on Computing 5 (4), 386–394.

Suykens, J.A.K., Vandewalle, J.P.L., De Moor, B.L.R., 1996. Artificial Neural Networks for Modelling and Control of Nonlinear Systems. Kluwer, Boston.

Szu, H., Telfer, B., Garcia, J., 1996. Wavelet transforms and neural networks for compression and recognition. Neural Networks 9 (4), 695–708.

Tam, K.Y., Kiang, M.Y., 1992. Managerial applications of neural networks: The case of bank failure predictions. Management Science 38 (7), 926–947.

Tang, Z., Almeida, C., Fishwick, P.A., 1991. Time series forecasting using neural networks vs Box-Jenkins methodology. Simulation 57 (5), 303–310.

Tang, Z., Fishwick, P.A., 1993. Feedforward neural nets as models for time series forecasting. ORSA Journal on Computing 5 (4), 374–385.

Tong, H., Lim, K.S., 1980. Threshold autoregressive, limit cycles and cyclical data. Journal of the Royal Statistical Society Series B 42 (3), 245–292.

Trippi, R.R., Turban, E., 1993. Neural Networks in Finance and Investment: Using Artificial Intelligence to Improve Real-world Performance. Probus, Chicago.

Tsay, R.S., 1986. Nonliearity tests for time series. Biometrika 73 (2), 461–466.

Turkkan, N., Srivastava, N.K., 1995. Prediction of wind load distribution for air-supported structures using neural networks. Canadian Journal of Civil Engineering 22 (3), 453–461.

Vishwakarma, K.P., 1994. A neural network to predict multiple economic time series. In: Proceedings of the IEEE International Conference on Neural Networks, 6, pp. 3674–3679.

Wallich, P., 1991. Wavelet theory: An analysis technique that's creating ripples. Scientific American, January, 34–35.
Wang, Z., Massimo, C.D., Tham, M.T., Morris, A.J., 1994. A procedure for determining the topology of multilayer feedforward neural networks. Neural Networks 7 (2), 291–300.
Wasserman, P.D., 1989. Neural Computing: Theory and Practice. Van Nostrand, Reinhold, New York.
Wedding II, D.K., Cios, K.J., 1996. Time series forecasting by combining RBF networks, certainty factors, and the Box-Jenkins model. Neurocomputing 10, 149–168.
Weigend, A.S., Gershenfeld, N.A., 1993. Time Series Prediction: Forecasting the Future and Understanding the Past. Addison-Wesley, Reading, MA.
Weigend, A.S., Huberman, B.A., Rumelhart, D.E., 1990. Predicting the future: A connectionist approach. International Journal of Neural Systems 1, 193–209.
Weigend, A.S., Huberman, B.A., Rumelhart, D.E., 1992. Predicting sunspots and exchange rates with connectionist networks. In: Casdagli, M., Eubank, S. (Eds.), Nonlinear Modeling and Forecasting. Addison-Wesley, Redwood City, CA, pp. 395–432.
Weigend, A.S., Rumelhart, D.E., Huberman, B.A., 1991. Generalization by weight-elimination with application to forecasting. Advances in Neural Information Processing Systems 3, 875–882.
Werbos, P.J., 1974. Beyond regression: new tools for prediction and analysis in the behavioral sciences. Ph.D. thesis, Harvard University.
Werbos, P.J., 1988. Generalization of backpropagation with application to a recurrent gas market model. Neural Networks 1, 339–356.
White, H., 1988. Economic prediction using neural networks: The case of IBM daily stock returns. In: Proceedings of the IEEE International Conference on Neural Networks, 2, pp. 451–458.
White, H., 1989. Learning in artificial neural networks: A statistical perspective. Neural Computation 1, 425–464.
Widrow, B., Rumelhart, D.E., Lehr, M.A., 1994. Neural networks: Applications in industry, business and science. Communications of the ACM 37 (3), 93–105.
Wilson, R., Sharda, R., 1992. Neural networks. OR/MS Today, August, 36–42.
Wilson, R., Sharda, R., 1994. Bankruptcy prediction using neural networks. Decision Support Systems 11, 545–557.
Wong, B.K., Bodnovich, T.A., Selvi, Y., 1995. A bibliography of neural networks business application research: 1988–September 1994. Expert Systems 12 (3), 253–262.
Wong, F.S., 1991. Time series forecasting using backpropagation neural networks. Neurocomputing 2, 147–159.
Wong, F.S., Wang, P.Z., Goh, T.H., Quek, B.K., 1992. Fuzzy neural systems for stock selection. Financial Analysis Journal, Jan./Feb., 47–52.
Wong, S.Q., Long, J.A., 1995. A neural network approach to stock market holding period returns. American Business Review 13 (2), 61–64.
Wu, B., 1995. Model-free forecasting for nonlinear time series (with application to exchange rates). Computational Statistics and Data Analysis 19, 433–459.
Yao, S., Wei, C.J., He, Z.Y., 1996. Evolving wavelet neural networks for function approximation. Electronics Letters 32 (4), 360–361.
Yoon, Y., Swales, G., 1991. Predicting stock price performance: A neural network approach. In: Proceedings of the 24th Hawaii International Conference on System Sciences., 4, pp. 156–162.
Yu, X.H., Chen, G.A., Cheng, S.X., 1995. Dynamic learning rate optimization of the backpropagation algorithm. IEEE Transactions on Neural Networks 6 (3), 669–677.
Zhang, J., Walter, G.G., Miao, Y., Wayne, W.N., 1995. Wavelet neural networks for function learning. IEEE Transactions on Signal Processing 43 (6), 1485–1497.
Zhang, Q., Benveniste, A., 1992. Wavelet networks. IEEE Transactions on Neural Networks 3 (6), 889–898.
Zhang, X., 1994. Time series analysis and prediction by neural networks. Optimization Methods and Software 4, 151–170.
Zhang, X., Hutchinson, J., 1993. Simple architectures on fast machines: Practical issues in nonlinear time series prediction. In: Weigend, A.S., Gershenfeld, N.A. (Eds.), Time Series Prediction: Forecasting the Future and Understanding the Past. Addison-Wesley, Reading, MA.

Biographies: Guoqiang ZHANG received a B.S. in Mathematics and an M.S. in Statistics from East China Normal University, and is currently a Ph.D. candidate at Kent State University. His research interests are forecasting, neural networks applications, inventory systems, and statistical quality control. In 1997, he received the Best Student Paper Award at the Midwest Decision Sciences Institute Annual Meeting.

B. Eddy PATUWO is an Associate Professor in the Administrative Sciences Department at Kent State University. He earned his Ph.D. in IEOR from Virginia Polytechnic Institute and State University. His research interests are in the study of stochastic inventory systems and neural networks. His research has been published in Decision Sciences, IIE Transactions, Journal of Operational Research Society, Computers and Operations Research, among others.

Michael Y. HU is a Professor of Marketing at Kent State University. He earned his Ph.D. in Management Science from the University of Minnesota in 1977. He has published extensively (about 80 research papers) in the areas of neural networks, marketing research, international business, and statistical process control. His articles have been published in numerous journals including Decision Sciences, Computers and Operations Research, OMEGA, Journal of Academic of Marketing Science, Journal of International Business Studies, Journal of Business Research, Financial Management, and many others.