KYBERNETIKA — VOLUME 38 (2002), NUMBER 6, PAGES 685 - 707

ARTIFICIAL NEURAL NETWORKS IN TIME SERIES FORECASTING: A COMPARATIVE ANALYSIS[1]

HÉCTOR ALLENDE, CLAUDIO MORAGA AND RODRIGO SALAS

Artificial neural networks (ANN) have received a great deal of attention in many fields of engineering and science. Inspired by the study of brain architecture, ANN represent a class of non-linear models capable of learning from data. ANN have been applied in many areas where statistical methods are traditionally employed. They have been used in pattern recognition, classification, prediction and process control. The purpose of this paper is to discuss ANN and compare them to non-linear time series models. We begin exploring recent developments in time series forecasting with particular emphasis on the use of non-linear models. Thereafter we include a review of recent results on the topic of ANN. The relevance of ANN models for the statistical methods is considered using time series prediction problems. Finally we construct asymptotic prediction intervals for ANN and show how to use prediction intervals to choose the number of nodes in the ANN.

1. INTRODUCTION

Artificial neural networks (ANN) have received a great deal of attention over the last years. They are being used in the areas of prediction and classification, areas where regression and other related statistical techniques have traditionally been used [13].

Forecasting in time series is a common problem. Using a statistical approach, Box and Jenkins [8] have developed the integrated autoregressive moving average (ARIMA) methodology for fitting a class of linear time series models. Statisticians in a number of ways have addressed the restriction of linearity in the Box–Jenkins approach. Robust versions of various ARIMA models have been developed. In addition, a large amount of literature on inherently non-linear time series models is available. The stochastic approach to non-linear time series outlined by [43] can not only fit non-linear models to time series data, but also provides measures of uncertainty in the estimated model parameters as well as forecasts generated by these models. It is the stochastic approach that again enables the specification of uncertainty in parameter estimates and forecasts.

[1]This research was supported in part by the Research Grant FONDECYT 1010101-7010101, in part by the Spanish State Secretary of the Ministry of Education, Culture and Sports (Grant SAB2000-0048) and by the Ministry of Education and Research of Germany (Grant CHL-99/023).

More recently, ANN have been studied as an alternative to these non-linear model-driven approaches. Because of their characteristics, ANN belong to the data-driven approach, i.e. the analysis depends on the available data, with little a priori rationalization about relationships between variables and about the models. The process of constructing the relationships between the input and output variables is addressed by certain general-purpose 'learning' algorithms [16]. Some drawbacks of the practical use of ANN are the possibly long time consumed in the modeling process and the large amount of data required by the present ANN technology. Speed-up is being achieved due to the impressive progress in increasing the clock rate of present processors. The demands on the number of observations remain however a hard open problem. One cause of both problems is the lack of a definite generic methodology that could be used to design a small structure. Most of the present methodologies use networks, with a large number of parameters ("weights"). This means lengthy computations to set their values and a requirement for many observations. Unfortunately, in practice, model parameters must be estimated quickly and just a small amount of data are available. Moreover, part of the available data should be kept for the validation and for performance-evaluation procedures.

This paper reviews recent developments in one important class of non-linear time series models, like the ANN's (model-free systems) and describe a methodology for the construction of prediction intervals which facilitates the estimation of forecast.

In the next section we provide a very brief review of the linear and non-linear ARMA models and the optimal prediction. Section 3 contains an overview of ANN terminology and describes a methodology for neural model identification. The multilayer feedforward ANN described can be conceptualized as a means of fitting a highly non-linear regression and time series prediction problem. In Section 4 we use the results of [24] to construct confidence intervals and prediction intervals in non-linear time series.

2. TIME SERIES ANALYSIS

2.1. Linear models

The statistical approach to forecasting involves the construction of stochastic models to predict the value of an observation x_t using previous observations. This is often accomplished using linear stochastic difference equation models, with random input. By far, the most important class of such models is the linear autoregressive integrate moving average (ARIMA) model. Here we provide a very brief review of the linear ARIMA-models and optimal prediction for these models. A more comprehensive treatment may be found for example in [8]. The seasonal ARIMA $(p,d,q)\times(P,D,Q)^S$ model for such time series is represented by

$$\Phi_P(B^S)\,\phi_p(B)\,\nabla_S^D\nabla^d x_t = \Theta_Q(B^S)\,\theta_q(B)\,\epsilon_t \tag{1}$$

where $\phi_p(B)$ is the nonseasonal autoregressive operator of order p, $\theta_q(B)$ is the nonseasonal moving average operator of order q, $\Phi_P(B^S)$, $\Theta_Q(B^S)$ are the seasonal autoregressive and moving average operator of order P and Q and the terms x_t and

ϵ_t are the time series and a white noise respectively. Moreover it is assumed that $E[\epsilon_t|x_{t-1}, x_{t-2}, \ldots] = 0$. This condition is satisfied for example when ϵ_t are zero mean, independent and identically distributed and independent of past x_ts. It is assumed throughout that ϵ_t has finite variance σ^2. The backshift operator B shifts the index of a time series observation backwards, e. g. $Bx_t = x_{t-1}$ and $B^k x_t = x_{t-k}$. The order of the operator is selected by Akaike's information criterion (AIC) or by Bayes information criterion (BIC) [10] and the parameters $\Phi_1, \ldots, \Phi_P$, $\phi_1, \ldots, \phi_p$, $\Theta_1, \ldots, \Theta_Q$ y $\theta_1, \ldots, \theta_q$ are selected from the time series data using optimization methods such as maximum likelihood [8] or using robust methods such as recursive generalized maximum likelihood [2]. The ARMA-model is limited by the requirement of stationarity and invertibility of the time series, i. e. the system generating the time series must be time invariant and stable. In addition, the residuals must be independent and identically distributed [7].

The ARMA models require a stationary time series in order to be useful for forecasting. The condition for a series to be weak stationary is that for all t

$$E[x_t] = \mu; \quad V[x_t] = \sigma^2; \quad COV[x_t, x_{t-k}] = \gamma_k. \tag{2}$$

Diagnostic checking of the overall ARMA models is done by the residuals. Several tests have been proposed, among them the most popular one seems to be the so-called portmanteau test proposed by [27] and its robust version by [1]. These tests are based on a sum of squared correlations of the estimated residuals suitably scaled.

2.2. Non-linear models

Theory and practice are mostly concerned with linear methods and models, such as ARIMA models and exponential smoothing methods. However, many time series exhibit features which cannot be explained in a linear framework. For example some economic series show different properties when the economy is going into, rather than coming out of, recession. As a result, there has been increasing interest in non-linear models.

Many types of non-linear models have been proposed in the literature, see for example bilinear models [40], classification and regression trees [9], threshold autoregressive models [43] and Projection Pursuit Regression [18]. The rewards from using non-linear models can occasionally be substantial. However, on the debit side, it is generally more difficult to compute forecasts more than one step ahead [25].

Another important class of non-linear models is that of non-linear ARMA models (NARMA) proposed by [14], which are generalizations of the linear ARMA models to the non-linear case. A NARMA model obeys the following equations:

$$x_t = h(x_{t-1}, x_{t-2}, \ldots, x_{t-p}, \epsilon_{t-1}, \ldots, \epsilon_{t-q}) + \epsilon_t \tag{3}$$

where h is an unknown smooth function, and as in Section 2.1 it is assumed that $E[\epsilon_t|x_{t-1}, x_{t-2}, \ldots] = 0$ and that variance of ϵ_t is σ^2. In this case the conditional mean predictor based on the infinite past observation is

$$\hat{x}_t = E[h(x_{t-1}, x_{t-2}, \ldots, x_{t-p}, \epsilon_{t-1}, \ldots, \epsilon_{t-q})|x_{t-1}, x_{t-2}, \ldots]. \tag{4}$$

Suppose that the NARMA model is invertible in the sense that there exists a function ν such that

$$x_t = \nu(x_{t-1}, x_{t-2}, \ldots) + \epsilon_t. \tag{5}$$

Then given the infinite past of observations $x_{t-1}, x_{t-2}, \ldots$, one can compute the ϵ_{t-j} in (3) exactly:

$$\epsilon_{t-j} = \kappa(x_{t-j}, x_{t-j-1}, \ldots), \qquad j = 1, 2, \ldots, q. \tag{6}$$

In this case the mean estimate is

$$\hat{x}_t = h(x_{t-1}, x_{t-2}, \ldots, x_{t-p}, \epsilon_{t-1}, \ldots, \epsilon_{t-q}) \tag{7}$$

where the ϵ_{t-j} are specified in terms of present and past x_u's. The predictor of (7) has a mean square error σ^2.

Since we have only a finite observation record, we cannot compute (6) and (7). It seems reasonable to approximate the conditional mean predictor (7) by the recursive algorithm

$$\hat{x}_t = h(x_{t-1}, x_{t-2}, \ldots, x_{t-p}, \hat{\epsilon}_{t-1}, \ldots, \hat{\epsilon}_{t-q}) \tag{8}$$

$$\hat{\epsilon}_{t-j} = x_{t-j} - \hat{x}_{t-j}, j = 1, 2, \ldots, q \tag{9}$$

with the following initial conditions

$$\hat{x}_0 = \hat{x}_{-1} = \ldots = \hat{x}_{-p+1} = \hat{\epsilon}_0 = \ldots = \hat{\epsilon}_{-q+1} = 0. \tag{10}$$

For the special case of non-linear autoregressive model (NAR), it is easy to check that (3) is given by

$$x_t = h(x_{t-1}, x_{t-2}, \ldots, x_{t-p}) + \epsilon_t. \tag{11}$$

In this case, the minimum mean square error (MSE) optimal predictor of x_t given x_{t-1}, $x_{t-2}, \ldots$, x_{t-p} is the conditional mean (for $t \geq p+1$).

$$\hat{x}_t = E[x_t | x_{t-1}, \ldots, x_{t-p}] = h(x_{t-1}, \ldots, x_{t-p}). \tag{12}$$

This predictor has mean square error σ^2.

3. ARTIFICIAL NEURAL NETWORKS

The brain is an enormously complex system in which information is distributed processed by mutual dynamical interactions of neurons. It is still difficult and challenging, to understand the mechanisms of the brain. The importance and effectiveness of brain-style computation has become a fundamental principle in the development of neural networks. There are three different research areas concerning neural networks. One is the experimental based on physiology and molecular biology. The second area is engineering applications of neural networks inspired by the brain-style computation where information is distributed as analog pattern signal, parallel computations are dominant and appropriate learning guarantees flexibility and robust computation. The third area is concerned with mathematical foundations of neuro-computing, which searches for the fundamental principles of parallel

distributed information systems with learning capabilities. Statistics has a close relation with the second application area of neuronal networks. This area has opened new practical methods of pattern recognition, time series analysis, image processing, etc.

Artificial Neural Networks (ANN) provide statistics with tractable multivariate non-linear methods to be further studied. On the other hand statistical science provides one of the crucial methods for constructing theoretical foundations of neurocomputing.

From a statistical perspective ANN are interesting because of their use in various kinds of problems, for example: prediction and classification. ANN have been used for a wide variety of applications, where statistical methods are traditionally employed. They have been used in classification problems as identifying underwater sonar contacts, and predicting heart problems of patients [4]. In time series applications they have been used in predicting stock market performance [23]. ANN are currently the preferred method in predicting protein secondary structures [17]. The statisticians would normally solve these problems through classical statistical models such as discriminant analysis, logistic regression, multiple regression and time series models such as ARIMA and forecasting methods.

Nowadays one can recognize ANN as a potential tool for data analysis. Several authors have done comparison studies between statistical methods and ANN (see e.g. [47] and [39]). These works tend to focus on performance comparisons and use specific problems as examples. ANN trained by error Backpropagation are examples of nonparametric regression estimators. In this paper we present the relations between nonparametric inference and ANN, we use the statistical viewpoint to highlight strength and weakness of neural models. There is a number of good introductory articles on ANN located in various scientific journals. For instance, [26] provides an excellent overview of ANN for the signal processing community. There have also been papers relating ANN and statistical methods [36] and [37]. One of the best for a general overview for statisticians is [13].

3.1. Elements of artificial neural networks

The three essential features of an artificial neural network (ANN) are the basic processing elements referred to as neurons or nodes; the network architecture describing the connections between nodes; and the training algorithm used to find values of the network parameters for performing a particular task.

An ANN consists of elementary processing elements (neurons), organized in layers (see Figure 1). The layers between the input and the output layers are called "hidden". The number of input units is determined by the application. The architecture or topology A_λ of a network refers to the topological arrangement of the network connections. A class of neural models is specified by

$$S_\lambda = \{g_\lambda(\underline{x}, \underline{w}), \underline{x} \in \Re^m, \underline{w} \in W\}, \quad W \subseteq \Re^\tau \tag{13}$$

where $g_\lambda(\underline{x}, \underline{w})$ is a non-linear function of $\underline{x}$ with $\underline{w}$ being its parameter vector, λ is the number of hidden neurons and τ is the number of free parameters determined by A_λ, i.e., $\tau = \rho(A_\lambda)$.

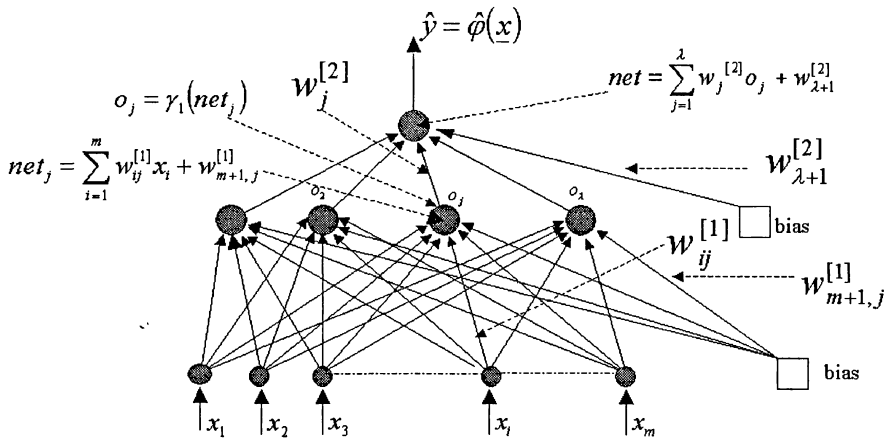

Fig. 1. A multilayer feedforward ANN for approximating an unknown function $\varphi(\underline{x})$.

A class (or family) of neural models is a set of ANN models which share the same architecture and whose individual members are continuously parameterized by the vector $\underline{w} = (w_1, w_2, \ldots, w_\tau)^T$. The elements of this vector are usually referred to as weights. For a single-hidden-layer architecture, the number of hidden units λ indexes the different classes of ANN models (S_λ) since it is an unambiguous descriptor of the dimension τ of the parameter vector $(\tau = (m+2)\lambda + 1)$.

Given the sample of observations, the task of neural learning is to construct an estimator $g(\underline{x}, \underline{w})$ of the unknown function $\varphi(\underline{x})$

$$g_\lambda(\underline{x}, \underline{w}) = \gamma_2 \left(\sum_{j=1}^{\lambda} w_j^{[2]} \gamma_1 \left(\sum_{i=1}^{m} w_{ij}^{[1]} x_i + w_{m+1,j}^{[1]} \right) + w_{\lambda+1}^{[2]} \right) \tag{14}$$

where $\underline{w} = (w_1, w_2, \ldots, w_\tau)^T$ is a parameter vector to be estimated, γ_s' represent linearity or non-linearity and λ is a control parameter (number of hidden units). An important factor in the specification of neural models is the choice of base functions γ, which are known as 'activation' functions. They can be any non-linear function as long as they are continuous, bounded and differentiable. Typically γ_1 is a sigmoidal or the hyperbolic tangent, and γ_2 is a linear function.

The estimated parameter $\hat{\underline{w}}$ is obtained by minimizing iteratively a cost functional $L_n(\underline{w})$ i. e.

$$\hat{\underline{w}} = \arg, \min\{L_n(\underline{w}) : \underline{w} \in W\}, W \subseteq \Re^\tau \tag{15}$$

where $L_n(\underline{w})$ is, for example, the ordinary mean square error function, i. e.

$$L_n(\underline{w}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - g_\lambda(\underline{x}_i, \underline{w}))^2. \tag{16}$$

The loss function in equation (16) gives us a measure of the accuracy with which an estimator A_λ, fits the observed data but it does not account for the estimator's (model) complexity. Given a sufficient large number of free parameters, $\tau = \rho(A_\lambda)$, a neural estimator A_λ, can fit the data with arbitrary accuracy. Thus, from the perspective of selecting between candidates, model expression (16) is an inadequate measure. The usual approach to the selection is the so-called discrimination approach, where the models are evaluated using a fitness criterion, which usually penalizes the in-sample performance of the model, as the complexity of the functional form increases and the degrees of freedom for error become less. Such criteria, commonly used in the context of regression analysis are: the R-Squared adjusted for degrees of freedom, Mallow's C_p criterion, Akaike's AIC criterion, etc.

The basic requirement of any ANN training method is convergence of the performance error to a locally unique minimum. By introducing the requirement that for any particular architecture A_λ the network has to be trained to convergence, we perform a restricted search in the function space. From each class S_λ we select only one member with its parameter estimated from equation (15). In this setting the actual training algorithm used, is irrelevant provided that it satisfies the convergence requirement. The first step is to estimate the parameters $\underline{w}$ of the model by iteratively minimizing the empirical loss $L_n(\underline{w})$ (see (16)). This stage must not be confused with model selection, which in this framework employs a different fitness criterion for selecting among fitted models. The second step is to compute the error Hessian $\hat{A}_n$. (See Appendix A.) This is used to facilitate the test on convergence. The third step is intended to perform a test for convergence and uniqueness, basically by examining whether $\hat{A}_n$ has negative eigenvalues. The fourth step is to estimate the prediction risk $P_\lambda = E[L(\underline{w}_n)]$, which adjusts the empirical loss for complexity. The fifth step is to select a model by employing the minimum prediction risk principle which expresses the trade-off between the generalization ability of the network and its complexity. However it has to be noted that since the search is restricted, the selected network is the best among the alternatives considered and it does not necessarily represent a global optimum. The final step involves testing the adequacy of the selected model. Satisfying those tests is a necessary but not sufficient condition for model adequacy. Failure to satisfy those tests indicates that either a different numbers of hidden units is needed or some relevant variables were omitted.

3.2. Model-free forecast

Artificial neural networks are essentially devices for non-parametric statistical inference. From the statistical viewpoint, they have a simple interpretation: given a sample $D_n = \{(x_i, y_i)\}_{i=1}^n$ generated by an unknown function $f(\underline{x})$ with the addition of a stochastic component ε, i. e.

$$y_i = f(\underline{x}_i) + \varepsilon_i \tag{17}$$

the task of "neural learning" is to construct an estimator $g(\underline{x}, \underline{w}) \equiv \hat{f}(\underline{x})$ of $f(\underline{x})$, where $\underline{w} = (w_1, \ldots, w_\tau)^T$ is a set of free parameters (known as "connection weights"

in sub-section 3.1.). Since no a priori assumptions are made regarding the functional form of $f(\underline{x})$, the neural model $g(\underline{x}, \underline{w})$ is a non-parametric estimator of the conditional density $E[y|\underline{x}]$, as opposed to a parametric estimator where the functional form is assumed a priori, for example, in a linear model.

ANN of non-linear autoregressive models (NAR)

An ANN topology and dynamics define an approximator from input to output. The unknown function $g : \Re^m \to \Re$ produces the observed sample pattern pairs $(x_1, y_1), (x_2, y_2), \ldots$ The sample data modify parameters in the neural estimator and bring the neural system input-output responses closer to the input-output responses of the unknown estimate g. In psychological terms, the neural system "learns from experience". In the neural estimator process, we do not ask the neural engineer to articulate, write down or guess the mathematical shape of the unknown function g. This is why we call the ANN estimation model-free.

A central problem of non-linear autoregressive models (NAR) is to construct a function, $h : \Re^p \to \Re$ in a dynamical system with the form

$$x_t = h(x_{t-1}, x_{t-2}, \ldots, x_{t-p}) \tag{18}$$

or possibly involving a mixture of chaos and randomness $x_t = h(x_{t-1}, x_{t-2}, \ldots, x_{t-p}) + \varepsilon_t$ in which h is an unknown smooth function and ε_t denotes noise. Similar to Section 2.1 we assume that $E[\varepsilon_t | x_{t-1}, x_{t-2}, \ldots] = 0$, and that ε_t has finite variance σ^2. Under these conditions the MSE optimal predictor of x_t, given $x_{t-1}, x_{t-2}, \ldots, x_{t-p}$ is as shown in equation (12).

Feedforward ANN were proposed as a NAR model for time series prediction by [14]. A feedforward ANN provides a non-linear approximation to h given by

$$\hat{x}_t = h(x_{t-1}, x_{t-2}, \ldots, x_{t-p}) = \sum_{j=1}^{\lambda} w_j^{[2]} \gamma_1 \left(\sum_{i=1}^{p} w_{ij}^{[1]} x_{t-i} + w_{p+1,j}^{[1]} \right) \tag{19}$$

where the function $\gamma_1(\cdot)$ is a smooth bounded monotic function. (19) is similar to equation (14), where γ_1 is a sigmoide, γ_2 is the identity function and the output node has no bias.

The parameters $w_j^{[2]}$ and $w_{ij}^{[1]}$ are estimated from a training and estimate $\hat{h}$ of h. Estimates are obtained by minimizing the sum of the squared residuals, similar to (15). This is done for example by a gradient descent procedure known as "Backpropagation", by Super Self–Adapting Backpropagation or by a second-order method for learning (see [16]).

4. PREDICTION INTERVALS FOR ANN

A theoretical problem of the ANN is the unidentifiability of the parameters. That is, there are two sets of parameters such that the corresponding distributions $(\underline{x}, y)$ are identical. In this section we concentrate on the case of only one hidden layer.

Further, we assume a nonparametric statistical model that relates y and $g(\underline{x}, \underline{w})$ as follows:

$$y = g(\underline{x}, \underline{w}) + \varepsilon \tag{20}$$

where the random component ε has a normal distribution with mean zero and variance σ^2. The function $g(\underline{x}, \underline{w})$ is a non-linear function such as (13).

The network is trained on the dataset $D_n = \{(\underline{x}_i, y_i)\}_{i=1}^n$; that is, these data are used to predict the future output at a new input x_{n+1} by $\hat{y}_{n+1} = g(\underline{x}_{n+1}, \underline{\hat{w}})$. We assume that for every $1 \leq i \leq n+1$, (14) and (20) are satisfied, that is, $y_i = g(\underline{x}_i, \underline{w}) + \varepsilon_i$, where y_{n+1} is the unobservable random variable that is the target of prediction. Further, we assume that the x_i's are independent of the ε_i's and the $(\underline{x}, \varepsilon_i)$, $1 \leq i \leq n+1$ are independent identically distributed (i.i.d.). Our aim in this section is to construct prediction intervals for y_{n+1} and confidence intervals for $g(\underline{x}_{n+1}, \underline{w})$, the conditional expectation of y_{n+1} given $\underline{x}_{n+1}$.

To discuss the identifiability (or rather the unidentifiability) of parameters, we first discuss two concepts (as in [41]). We say that an ANN (with a fixed set of parameters) is "redundant" if there exists another ANN with fewer neurons that represents exactly the same relationship function $g(\cdot, \underline{w})$. A formal definition of the reducibility of $\underline{w}$, can be found in [41].

Definition 4.1. For γ_1 chosen as a sigmoidal function and γ_2 a linear function $\underline{w}$ is called "reducible" if one of following three cases holds for $j \neq 0$, (a) $w_j^{[2]} = 0$ for some $j = 1, \ldots, \lambda$; (b) $\underline{w}_j^{[1]} = (w_{1j}^{[1]}, \ldots, w_{mj}^{[1]}) = \underline{0}$ for some $j = 1, \ldots, \lambda$; or (c) $(\underline{w}_j^{[1]}, w_{m+1,j}^{[1]}) = (\underline{w}_l^{[1]}, w_{m+1,l}^{[1]})$ for some $j \neq l$, where $\underline{0}$ denotes the zero vector of the appropriate size.

If $\underline{w}$ is reducible and γ_1 is a sigmoidal function, then the corresponding ANN relative to (20) is obviously redundant. On the other hand, an irreducible $\underline{w}$ may not always lead to a nonredundant ANN. [41] proved that if the class of functions $\{\gamma_1(bx + b_0), b > 0\} \cup \{\gamma_1 \equiv 1\}$ is linearly independent then the irreducibility of $\underline{w}$ implies that the corresponding ANN is nonredundant.

In general note that every ANN is unidentifiable. However [24] showed that ANN's with certain activation functions, leave the distribution of y invariant up to certain family $\Im$ of transformations of $\underline{w}$. That is, if there exist another $\underline{w}^*$ such that $g(\cdot, \underline{w}^*) = g(\cdot, \underline{w})$, then there is a transformation generated by $\Im$ that transforms $\underline{w}^*$ to $\underline{w}$. Further under the assumption that γ_1 is continuously differentiable, the matrix

$$\Sigma = E[\nabla_{\underline{w}} g(\underline{x}, \underline{w}) \nabla_{\underline{w}} g(\underline{x}, \underline{w})^T] \tag{21}$$

is non-singular.

In this section we construct confidence intervals and prediction intervals based on an ANN and show these to be asymptotically valid using the results.

From [41], [24], we first assume that the number of neurons λ is known. Specifically we assume that our observations $(\underline{x}_i, y_i)$, $1 \leq i \leq n$ satisfy (20), that is

$$y_i = g(\underline{x}_i, \underline{w}) + \varepsilon_i. \tag{22}$$

Furthermore, let y_{n+1} denote a future unknown observation that satisfies

$$y_{n+1} = g(\underline{x}_{n+1}, \underline{w}) + \varepsilon_{n+1}. \tag{23}$$

We then construct a prediction interval for y_{n+1} and a confidence interval for $g(\underline{x}_{n+1}, \underline{w})$, the conditional mean of y_{n+1} given $\underline{x}_{n+1}$.

Before doing so, we state general results about an invariant statistic, where $g(\cdot, \underline{w})$ may or may not correspond to an ANN. We write the parameter space W, a subset of $\Re^r$, as the union of W_i's where W_i may or may not be disjoint. We assume that there exist differentiable functions T_i, $1 \leq i \leq \lambda$, that map W_1 onto W_i that is,

$$T_i(W_1) = W_i. \tag{24}$$

Let $\underline{w}_0 \in W$ denote the true parameter and let $w_0^{[1]}$ be the point in W_1 that corresponds to $\underline{w}_0$. Assume that $\hat{\underline{w}}^{[1]}$ is a consistent estimator for $\underline{w}_0^{[1]}$ based on a sample size n and

$$\sqrt{n}\left(\hat{\underline{w}}^{(1)} - \hat{\underline{w}}_0^{(1)}\right) \sim N\left(0, \sigma^2 V\left(\underline{w}_0^{(1)}\right)\right) \tag{25}$$

where σ^2 is a scale parameter that can be estimated consistently by an estimator $\hat{\sigma}^2$ and $V\left(\underline{w}_0^{(1)}\right)$ is a square matrix (see [16]). Let $\hat{\underline{w}}$ be an arbitrary estimator that takes value from $\{\hat{\underline{w}}^{(1)}, \ldots, \hat{\underline{w}}^{(\lambda)}\}$ where $\hat{\underline{w}}^{(i)} = T_i(\hat{\underline{w}}^{(1)})$. This is to say that for every n and every dataset, there exists an i such that $\hat{\underline{w}}_0 = \hat{\underline{w}}^{(i)}$. A real-valued function $l(w)$ is said to be invariant with respect to all the transformations T_i if

$$l(\underline{w}) = l(T_i(\underline{w})) \qquad \text{for every } i. \tag{26}$$

One can show that the asymptotic variance of an invariant statistic is also invariant as stated in the following result (see [24]). Assume that $l(w)$ is differentiable and invariant. Then as $n \to \infty$, $\sqrt{n}[l(\hat{\underline{w}}_0) - l(\underline{w}_0)]$ converges to a normal with mean zero and variance $\nu^2(\underline{w}_0)$, where $\nu^2(\underline{w}_0) = \sigma^2[\nabla l(\underline{w}_0)^T V(\underline{w}_0) \nabla l(\underline{w}_0)]$. Furthermore, the function $\nu^2(\underline{w}_0)$ is invariant with respect to all of the transformations T_i.

Under additional continuity assumptions it can be proved that the asymptotic variance can be estimated consistently by $\hat{\sigma}^2[\nabla l(\hat{\underline{w}}^{(i)})^T V(\hat{\underline{w}}^{(i)}) \nabla l(\hat{\underline{w}}^{(i)})]$, which again by invariance equals

$$\hat{\sigma}^2[\nabla l(\hat{\underline{w}}_0)^T V(\hat{\underline{w}}_0) \nabla l(\hat{\underline{w}}_0)] \tag{27}$$

therefore if $\nabla l(\underline{w})$ and $V(\underline{w})$ are both continuous in $\underline{w}$, then (27) is a consistent estimator for the asymptotic variance of $\sqrt{n}[l(\hat{\underline{w}}_0) - l(\underline{w}_0)]$.

Returning to the neural network problem, we now apply the last results to model (20) and we assume that the true value $\underline{w}_0$ of $\underline{w}$ is irreducible. We may make this assumption without loss of generality, because otherwise the neural network is redundant and we may drop one neuron without changing the input-output relationship at all. We may continue this process until a nonredundant network is obtained. This corresponds to an irreducible $\underline{w}_0$.

Assume that W, the parameter space, is a compact set. Furthermore, make the following assumptions:

i) ε_i are i.i.d. with mean zero and variance σ^2, and ε_i's are statistically independent of x_i's.

ii) x_i are i.i.d. samples from an unknown distribution $F(\underline{x})$ whose support is $\Re^m$.

iii) γ_1 in (14) is a sigmoidal function with continuous second-order derivative. Let γ_1' denote its derivative. Furthermore the class of functions $\{\gamma_1(bx + b0), b > 0\} \cup \{\gamma_1'(bx+b0), b > 0\} \cup \{x\gamma_1'(bx+b0), b > 0\} \cup \{\gamma_1 \equiv 1\}$ is linearly independent.

iv) γ_2 in (14) is a linear function.

v) $\underline{w}_0$ is an interior point of W.

Let $\underline{\hat{w}}$ be a global minimizer of $\sum_{i=1}^{n} \left(y_i - g(\underline{x}_i, \underline{w})\right)^2$, which exists by the compactness of W and continuity of g. Then

$$g(\underline{x}_{n+1}, \underline{\hat{w}}) \pm t_{(1-\alpha/2);[n-(m+2)\lambda-1]} \hat{\sigma} \sqrt{S(\underline{\hat{w}})} \tag{28}$$

is a confidence interval for $g(\underline{x}_{n+1}, \underline{w})$ with asymptotic coverage probability $1 - \alpha$. Here $t_{(1-\alpha/2);[n-(m+2)\lambda-1]}$ denotes the $1 - \alpha/2$ quantile of a t-Student distribution with $[n - (m + 2)\lambda - 1]$ degrees of freedom,

$$\hat{\sigma}^2 = \frac{1}{[n - (m + 2)\lambda - 1]} \sum_{i=1}^{n} [y_i - g(\underline{x}_i, \underline{\hat{w}})]^2 \tag{29}$$

and

$$S(\underline{\hat{w}}) = \frac{1}{n} \left\{ \left[\nabla_{\underline{w}} g(\underline{x}_{n+1}, \underline{w}) \right]_{\underline{w}=\underline{\hat{w}}}^{T} \hat{\Sigma}^{-1}(\underline{\hat{w}}) \right) \left[\nabla_{\underline{w}} g(\underline{x}_{n+1}, \underline{w}) \right]_{\underline{w}=\underline{\hat{w}}} \right\} \tag{30}$$

where

$$\hat{\Sigma}(\underline{\hat{w}}) = \frac{1}{n} \left\{ \sum_{i=1}^{n} \left[\nabla_{\underline{w}} g(\underline{x}_i, \underline{w}) \nabla_{\underline{w}} g(\underline{x}_i, \underline{w})^T \right]_{\underline{w}=\underline{\hat{w}}} \right\}. \tag{31}$$

Furthermore, assume that ε_{n+1} is normally distributed. Then

$$I_{\underline{w}}(y_{n+1}) = g(\underline{x}_{n+1}, \underline{\hat{w}}) \pm t_{(1-\alpha/2);[n-(m+2)\lambda-1]} \hat{\sigma} \sqrt{1 + S(\underline{\hat{w}})} \tag{32}$$

is an asymptotic prediction interval for y_{n+1}; that is,

$$P_r[y_{n+1} \in I_{\underline{w}}(y_{n+1})] \to 1 - \alpha \tag{33}$$

the proof of these results are given in [24].

A practical problem that occurs in many applications of ANN's is how to choose the network structure. When restricted to feedforward networks with only one hidden layer, this problem becomes how to choose the number of hidden neurons. One possible approach, which can be called the "prediction interval approach", is to choose the number of nodes so that the prediction interval has coverage probability close to the nominal level (e. g. 95 % or 90 %) and has the shortest expected length. Because both quantities are unknown, they should be estimated. The delete-one

jackknife for the coverage probability could be used. Specifically, this involves deleting a pair $(\underline{x}_i, y_i)$ and using the rest of data together with x_i to construct a prediction interval for y_i. By letting i vary, we have n intervals. The coverage probability can then be estimated by counting the proportion of times the intervals cover y_i. One could also calculate the average length of n intervals and use it to estimate the expected length. Another possible approach, which can be called the "prediction error approach", is to choose the number of nodes to minimize the jackknife estimate of the prediction error.

Finally other possible approaches are bootstrap methods, or so-called "resampling techniques" that permit rather accurate estimate of finite sample distributions for $\underline{\hat{w}}_n$ when $\{(\underline{x}_i, y_i)\}_{i=1}^n$ is a sequence of independent identically distributed (i.i.d.) random variables. The basic idea is to draw a large number N of random samples of size n with replacement from $\{(\underline{x}_i, y_i)\}_{i=1}^n$, calculate $\underline{\hat{w}}_n$ for each of the N samples, say $\underline{\hat{w}}_n^{(i)}$; $i = 1, 2, \ldots, N$, and use the resulting empirical distribution of the estimates $\underline{\hat{w}}_n^{(i)}$, as an estimate of the sampling distribution of $\underline{\hat{w}}_n$. The bootstrap methods are not recommended, because computationally they are too time-consuming. Resampling techniques are beyond the scope of this paper.

5. APPLICATION TO DATA

In this section the ANN will be applied to two examples, the first one is the well-known 'airline' data and next we will deal with the 'RESEX' data. Both time series are monthly observations and have been analyzed by many scientists and are a baseline to compare different models.

The results reported in this paper were computed by separating each set of data in two subsets, were the first n monthly observations (data), corresponding from time 1 to time T, called samples or training set were used to fit the model and then use the last 12, called test set, corresponding from time $T+1$ to $T+12$, to make the forecast. The data used to fit the model are also used for the training of the neural network, this data were re-escaled in the interval $[-1, 1]$. The NN used to model the data and then used to forecast is a feedforward with one hidden layer and a bias in the hidden and output layer. The number of neurons m in the input layer is the same as the number of lags needed, these neurons do not perform any processing, they just distribute the input values to the hidden layer, they serve as a sense layer. In the hidden layer different number of neurons are used to choose the best architecture, the activation function used is the sigmoidal function $\gamma_1(z) = \frac{1}{1+e^{-z}}$. One neuron is used in the output corresponding to the forecast, and it uses a linear activation function to obtain values in the real space. The forecasts were obtained using the data available (samples), one-step forecast a time. The sample corresponding to the recent forecast is included for the next forecast. The model parameters were not re-estimated at each step when computing the forecasts.

The weights (parameters) to be used in the NN model are estimated from the data by minimizing the mean squared error $mse = \frac{1}{n_{\text{effective}}} \sum_i (y_i - g(\underline{x}_i, \underline{w}))^2$ of the within-sample one-step-ahead forecast errors, where $n_{\text{effective}}$ denotes the number of effective observations used in fitting the model, because some data may be lost by

differentiating. To train the network a backpropagation algorithm with momentum was used, which is an enhancement of the backpropagation algorithm. The network 'learns' by comparing the actual network output and the target; then it updates its weights by computing the first derivatives of the objective function, and uses momentum to avoid local minima.

The statistics computed for each model were the following:

- $S = \sum_i (e_i)^2$, the sum of squared residuals up to time T, where $e_i = y_i - g(\underline{x}_i, \underline{w})$ are the residuals, and $g(\underline{x}_i, \underline{w})$ is the output of the NN and y_i is the target (training set).
- The estimate of the residual standard deviation: $\hat{\sigma} = \sqrt{\frac{S}{n_{\text{effective}} - \tau}}$ where $\tau = (m+2)\lambda + 1$ is the number of parameters.
- The Akaike information criterion (AIC): $\text{AIC} = n_{\text{effective}} \ \ln(S/n_{\text{effective}}) + 2\tau$
- The Bayesian information criterion (BIC): $\text{BIC} = n_{\text{effective}} \ \ln(S/n_{\text{effective}}) + \tau + \tau \ \ln(n_{\text{effective}})$
- S_{pre} is the sum of squares of one-step-ahead forecast errors of the test set.

To choose the architecture of the model that best fits the data, one can use the residual sum of squares, S, but the larger the model is made (more neurons), the smaller becomes S and the residual standard deviation, and the model gets more complicated. Instead BIC and AIC as minimization criteria are used for choosing a 'best' model from candidates models having different number of parameters. In both criteria, the first term measures the fit and the rest is a penalty term to prevent overfitting, where BIC penalizes more severely the extra parameter than AIC does. Overfitting of the model is not wanted, because it produces a very poor forecast, giving another reason to choose AIC and BIC over S to select the best model. The lower value obtained by this criterion, the better is the model.

To identify different classes of neural models as expressed by equation (13), following notation $\text{NN}(j_1, \ldots, j_k; \lambda)$ was used, which denotes a neural network with inputs at lags $j_1, \ldots, j_k$ and with λ neurons in the hidden layer.

5.1. Artificial neural network for the airline data

In this section we show an example of the well-known airline data, listed by Box et al [8], series G, and earlier by Brown [11] (see Figure 2). The data of this series have an upward trend, a seasonal variation called multiplicative seasonality. The airline data comprises monthly totals of international airline passengers from January 1949 to December 1960.

The airline data was modeled by a special type of seasonal autoregressive integrated moving average model (ARIMA), of order $(0,1,1) \times (0,1,1)_{12}$ as described in Section 2.1 which has the form $(1 - B^{12})(1 - B)x_t = (1 - \Theta_1 B^{12})(1 - \theta_1 B)a_t$, after some operations the following equation is obtained, $x_t = x_{t-1} + x_{t-12} - x_{t-13} + a_t - \theta_1 a_{t-1} - \Theta_1 a_{t-12} + \theta_1 \Theta_1 a_{t-13}$, taking care of using the appropriate transformation

to make the seasonality additive, in this case natural logarithm is taken over the data.

We will use an ANN to fit and forecast the airline data, because of the non-linearity property of the NN models; this will allow us to deal with the multiplicative seasonality.

Choosing the architecture

Different neural network architectures were evaluated with the statistics described in Section 5, the best model using AIC and BIC is NN(1, 12, 13; 1), having the best forecast with a minimum S_{pred} (see Table 1), so the NN(1, 12, 13; 1) model was selected for further results. Using the Box–Jenkins airline model one can try to use the proposed lags, i.e. $(x_{t-1}, x_{t-12}, x_{t-13})$, as the input to the neural network and then see its performance.

Table 1. Results obtained for the NN model chosen for the airline data.

Lags	λ	τ	S	desv	AIC	BIC	S_{pred}	AIC prediction	BIC prediction
1, 12, 13	1	6	0.2583	0.0478	−715.8	−695.1	0.1556	−40.1	−31.2

Forecasting and prediction intervals

After selecting the model, it is used to forecast the rest of the data (test data) using one-step-ahead forecast. The result is shown in Table 2 and it is represented in Figure 2. By using equation (29) to (32) the asymptotic prediction interval is calculated for each one-step forecast. The prediction interval computed for $\alpha = 0.05$ and $\alpha = 0.10$ is shown in Figure 3 respectively, and the values obtained are shown in Table 2.

Table 2. Prediction of the NN, for the airline data.

Month	Target	Prediction $\alpha = 0.05$	Prediction $\alpha = 0.10$
133	417	428.9431 ± 26.9128	428.9431 ± 21.3724
134	391	398.4319 ± 26.9475	398.4319 ± 21.3999
135	419	461.1852 ± 26.8404	461.1852 ± 21.3152
136	461	417.7294 ± 28.4284	417.7294 ± 22.5760
137	472	482.4761 ± 29.9372	482.4761 ± 23.7742
138	535	517.8650 ± 29.8691	517.8650 ± 23.7202
139	622	573.9760 ± 29.9452	573.9760 ± 23.7806
140	606	581.7285 ± 31.6975	581.7285 ± 25.1721
141	508	598.4535 ± 32.0805	498.4535 ± 25.4763
142	461	450.3307 ± 32.0715	450.3307 ± 25.4692
143	390	414.2488 ± 32.0719	414.2488 ± 25.4695
144	432	442.6636 ± 32.3828	442.6636 ± 25.7163

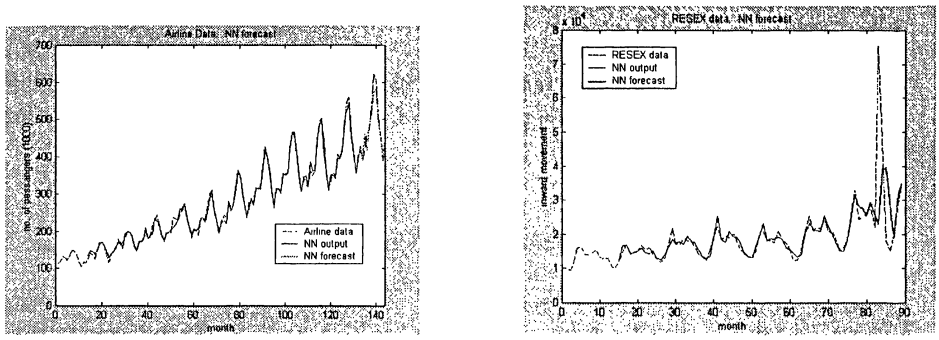

Fig. 2. (left) Airline data and its NN model and prediction. (right) RESEX data and its NN model and prediction.

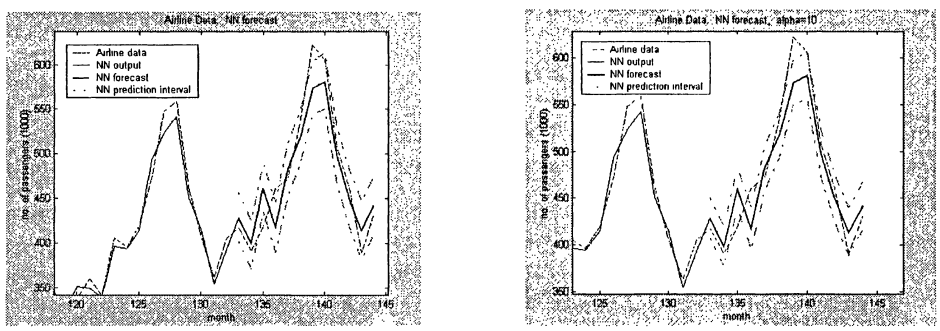

Fig. 3. Asymptotic prediction interval for the Airline data with $\alpha = 0.05$ (left) and $\alpha = 0.10$ (right).

5.2. Artificial Neural Network for the RESEX data

In this section the procedure is applied to the Residence Telephone Extensions Inward Movement (Bell Canada) known as RESEX data. The chosen series is a monthly series of "inward movement" of residential telephone extensions of a fixed geographic area in Canada from January 1966 to May 1973, a total of 89 data points. This series has two extremely large values in November and December 1972 as it is shown in Figure 2. The two obvious outliers have a known cause, namely a bargain month (November) in which residence extensions could be requested free of charge. Most of the orders were filled during December, with the remainder being filled in January.

Brubacher (1974) identified the stationary series as an ARIMA $(2,0,0)\times(0,1,0)_{12}$ model, i. e., the RESEX data is represented by an $AR(2)$ model after differentiating. As described in 2.1 it has the form $(1-\phi_1 B-\phi_2 B^2)(1-B^{12})x_t = a_t$ and after some operations $x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + x_{t-12} - \phi_1 x_{t-13} - \phi_2 x_{t-14} + a_t$.

Choosing the Architecture

Different architectures were tested, and the best result is obtained by NN(1, 2, 12, 13, 14; 1) (see Table 3), the NN using the lags of the ARIMA model. So this model was chosen for further results.

Table 3. Results obtained for the NN model chosen for the RESEX data.

Lags	λ	τ	S	desv	AIC	BIC	S_{pred}	AIC prediction	BIC prediction
1,2,12,13,14	1	8	0.6876	0.1118	−268.6	−243.5	25.6	25.1	37.0

Forecasting and prediction intervals

After selecting the model, it is used to forecast the rest of the data (test data) using one-step-ahead-forecast. The result is shown in Table 4, and it is represented in Figure 2.

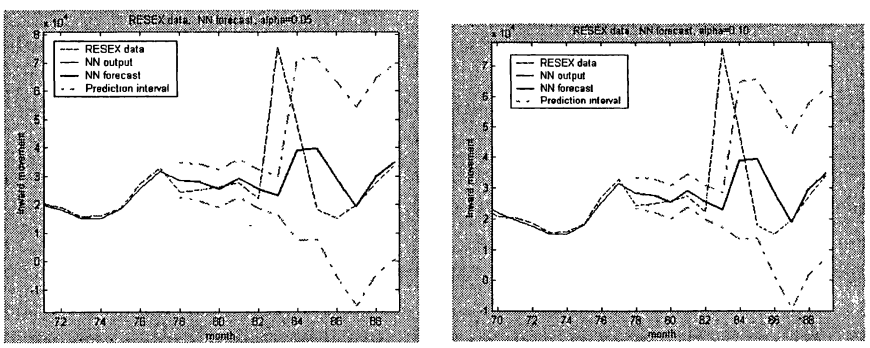

Fig. 4. Asymptotic prediction interval for the RESEX data with $\alpha = 0.05$ (left) and $\alpha = 0.10$ (right).

By using equation (29) to (32) the asymptotic prediction interval is calculated for each one-step forecast.The results are shown in Table 4 and in Figure 4 for $\alpha = 0.05$, and $\alpha = 0.10$. Both graphics in Figure 4 have a prediction interval that is large, because of the huge outliers presented, giving a poor forecast with a lot of error and variance. But the NN model at least tried to follow the trend of the data in the outlier part.

Table 4. Prediction of the NN for the RESEX data.

month	Target	prediction $\alpha = 0.05$	prediction $\alpha = 0.10$
78	24309	28360.2 ± 6171.8	28360.2 ± 4976.9
79	24998	27717.7 ± 6616.9	27717.7 ± 5335.8
80	25996	25508.0 ± 6786.7	25508.0 ± 5472.7
81	27583	29374.2 ± 6718.8	29374.2 ± 5418.0
82	22068	25364.3 ± 6757.6	25364.3 ± 5449.3
83	75344	22993.4 ± 6997.7	22993.4 ± 5642.9
84	47365	39153.2 ± 31888.5	39153.2 ± 25714.7
85	18115	39670.5 ± 31993.2	39670.5 ± 25799.2
86	15184	28882.0 ± 34279.9	28882.0 ± 27643.1
87	19832	19117.6 ± 35184.9	19117.6 ± 28372.8
88	27597	29709.1 ± 34676.1	29709.1 ± 27962.6
89	34256	35221.7 ± 34300.1	35221.7 ± 27659.4

6. CONCLUSIONS

It is the premise of this paper that the learning methods in ANN are sophisticated statistical procedures and that tools developed for the study of statistical procedures generally do not only yield useful insights into the properties of specific learning procedures but also suggest valuable improvements in alternatives to and generalizations of existing learning procedures.

Particularly applicable are asymptotic analytical methods that describe the behavior of statistics when the size n of the training set is large. At present, there is no easy answer to the question of how large "n" must be for the approximators described earlier to be "good".

The advantage of the ANN technique proposed in this paper is that it provides a methodology for model-free approximation; i. e. the weighted vector estimation is independent of any model. It has liberated us from the procedures of the model-based selection and the sample data assumptions. When the non-linear systems are still in the state of development, we can conclude that the ANN approach suggests a competitive and robust method for the system analysis, forecast and control. The ANN present is a superior technique in the modeling of another non-linear time series such as: bilinear models, threshold autorregressive models and regression trees. And the connections between forecasting, data compression, and neurocomputing shown in this paper seems very interesting in the time series analysis.

To decide which architecture one may use to model some time series, first, it is possible to try traditional methods, by using a simple autocorrelation function, to find the kind of time series that we are dealing with, and indeed, the lags that are used as input in the NN. Second, to select the number of hidden neurons, we start with one and then we increase it until the performance evaluated by AIC and BIC becomes worse. Then, we train the network with the first data, and finally use the last data to forecast. Asymptotic predictions intervals are computed for each

one-step-ahead-forecast, to show the limits where the data is moving.

The study of the stochastic convergence properties (consistency, limiting distribution) of any proposed new learning procedure is strongly recommended, in order to determine what it is that the ANN eventually learns and under what specific conditions. Derivation of the limiting distribution will generally reveal the statistical efficiency of the new procedure relative to existing procedures and may suggest modifications capable of improving statistical efficiency. Furthermore, the availability of the limiting distribution makes possible valid statistical inferences. Such inferences can be of great value in the research of the optimal network architectures in particular applications. A wealth of applicable theory is already available in the statistics, engineering, and system identification and optimization theory literature.

It is also evident that the fields of statistics has much to gain from the neurocomputing techniques. Analyzing neural network learning procedures pose a host of interesting theoretical and practical challenges for statistical methods; all is not cut and dried. Most important, however, neural network models provide a novel, elegant and rich class of mathematical statistical methods for data analysis.

In spite of the robust forecast performance for ANN some problems remain to be solved. For example: (i) How many input nodes are required for a seasonal time series? (ii) How to treat the outlier data? (iii) How to avoid the problem of overfitting? (iv) How to find the $(1-\alpha)\,\%$ confidence interval for the forecast? (v) How to treat the missing data?

In general the following conclusions and guidelines can be stated concerning the use of statistical methods and ANN:

1. If the functional form linking inputs and output is unknown, only known to be extremely complex, or of no interest to the investigator, an analysis using ANN may be best. The availability of large training datasets and powerful computing facilities are requirements for this approach.

2. If the underlying physics of the data generating process are to be incorporated into the analysis, a statistical approach may be the best. Generally, fewer parameters need to be estimated and the training datasets can be substantially smaller. Also, if measures of uncertainty are desired, either in parameter estimates or forecasts, a statistical analysis is mandatory. If the models fit to data are to be used to delve into the underlying mechanisms, and if measures of uncertainty are sought, a statistical approach can give more insight. In this sense, statistics provides more value added to a data analysis; it probably will require a higher level of effort to ascertain the best fitting model, but error in predictions, error in parameter estimate, and assessment of model adequacy are available in statistical analysis. In addition to providing measures of parameter and prediction uncertainty, statistical models inherently possess more structure than ANN do, which are often regarded as "black boxes". This structure is manifested as specification of a random component in statistical models. As such, statistical methods have more limited application. If a non-linear relationship exists between inputs and outputs, then data of this complexity may best modeled by an ANN. A summary of these considerations can be found in Table 7. (See Appendix C.)

APPENDIX A: ASYMPTOTIC DISTRIBUTION OF $\hat{\underline{w}}_n$

Under certain mild regularity assumptions, it can be shown [24] that the asymptotic distribution of the standardized quantity $\sqrt{n}(\hat{\underline{w}}_n - \underline{w}_0)$ is zero mean multivariate normal with covariance matrix $C = A^{-1}BA^{-1}$ where $\hat{\underline{w}}_n$ is the estimated and $\underline{w}_0$ the true parameter vector and

$$A = E[\nabla\nabla r(x, \underline{w}_0)] \qquad \text{and} \qquad B = E[\nabla r(z, \underline{w}_0)\nabla r(z, \underline{w}_0)^r].$$

The matrices A and B are non-singular with ∇ and $\nabla\nabla$ denoting the $(\tau \times 1)$ gradient and $(\tau \times \tau)$ Hessian operator with respect to w (τ is the number of network parameters). However, since the true parameters $\underline{w}_0$ are not known, the weakly consistent estimator $\hat{C} = \hat{A}_n^{-1}\hat{B}_n\hat{A}_n^{-1}$ of the covariance matrix C has been used instead, where

$$\hat{A} = n^{-1}\sum_{i=1}^{n} \nabla\nabla r(z_i, \hat{\underline{w}}_n) \tag{34}$$

$$\hat{B}_n = n^{-1}\sum_{i=1}^{n} \nabla r(z_i, \hat{\underline{w}}_n)\nabla r(z_i, \hat{\underline{w}}_n)^T \tag{35}$$

$$r(z_i, \hat{\underline{w}}_n) = \frac{1}{n}[y_i - g(x_i; \hat{\underline{w}}_n)]^2. \tag{36}$$

This has no effect on the asymptotic distribution of the network's parameters, although larger n will be needed to obtain an approximation as good as if C itself were available. The single most important assumption made is that $\hat{\underline{w}}_n$ is a locally unique solution, i. e. none of its parameters can be expressed in terms of the others, or equivalently, the network is not overparameterized. This is reflected in the natural requirement that matrices A and B are non-singular.

The fact that $\sqrt{n}(\hat{\underline{w}}_n - \underline{w}_0) \sim N(0, C)$ can be used to robustly estimate the standard error of any complex function of $\hat{\underline{w}}_n$ i. e. $\theta = \rho(\hat{\underline{w}}_n)$, without the need for an analytic derivation. By stochastically sampling from the distribution of $\hat{\underline{w}}_n$, we can inexpensively create a sufficient large number r of parameter vectors $\hat{\underline{w}}_n^{(s)}$, where $s = 1, 2, \ldots, r$ and then compute the estimate $\hat{\sigma}_A$ of the standard error as follows:

$$\hat{\sigma}_A = \left[(r-1)^{-1}\sum_{s=1}^{r}\left(\hat{\theta}^{(s)} - \hat{\theta}(0)\right)^2\right]^{\frac{1}{2}} \tag{37}$$

where

$$\hat{\theta}(0) = r^{-1}\sum_{s=1}^{r}\hat{\theta}^{(s)} = r^{-1}\sum_{s=1}^{r}\rho(\hat{\underline{w}}_n^{(s)}). \tag{38}$$

The scheme is independent of the functional $\rho(\cdot)$ and much less computationally demanding, compared to bootstrap for example, since the estimate $\hat{w}_n$ has to be obtained only once (see [34]).

APPENDIX B: TIME SERIES DATA

SERIES G: International Airline Passengers: monthly totals (Thousands of passengers).

Table 5. Series G.

	JAN	FEB	MAR	ABR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
1949	112	118	132	129	121	135	148	148	136	119	104	118
1950	115	126	141	135	125	149	170	170	158	133	114	140
1951	145	150	178	163	172	178	199	199	184	162	146	166
1952	171	180	193	181	183	218	230	242	209	191	172	194
1953	196	196	236	235	229	243	264	272	237	211	180	201
1954	204	188	235	227	234	264	302	293	259	229	203	229
1955	242	233	267	269	270	315	364	347	312	274	237	278
1956	284	277	317	313	318	374	413	405	355	306	271	306
1957	315	301	356	348	355	422	465	467	404	347	305	336
1958	340	318	362	348	363	435	491	505	404	359	310	337
1959	360	342	406	396	420	472	548	559	463	407	362	405
1960	417	391	419	461	472	535	622	606	508	461	390	432

RESEX: Residence Telephone Extensions inward Movement (Bell Canada).

Table 6. RESEX.

	JAN	FEB	MAR	ABR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
1966	10165	9279	10930	15876	16485	14075	14168	14535	15367	13396	12606	12932
1967	10545	10120	11877	14752	16932	14123	14777	14943	16573	15548	15838	14159
1968	12689	11791	12771	16952	21854	17028	16988	18797	18026	18045	16518	14425
1969	13335	12395	15450	19092	22301	18260	19427	18974	20180	18395	15596	14778
1970	13453	13086	14340	19714	20796	18183	17981	17706	20923	18380	17343	15416
1971	12465	12442	15448	21402	25437	20814	22066	21528	24418	20853	20673	18746
1972	15637	16074	18422	27326	32883	24309	24998	25996	27583	22068	75344	47365
1973	18115	15184	19832	27597	34256							

APPENDIX C: COMPARISON BETWEEN STATISTICAL METHODS (SM) AND ARTIFICIAL NEURAL NETWORKS (ANN)

Table 7. Statistical analysis versus ANN.

Characteristics	SM	ANN
General	Randomness Variability Structured Model Single or few outputs	Complex, non-linear Input /output relationships Multiple Outputs.
Data required	Relatively small Training datasets May require probability Distributions	Massive training Datasets needed to estimate weights
Model specifications	Physical Law Models Linear discrimination	No process Knowledge required Non-linear discrimination
Goodness of fit Criterion	Many possibilities Best fit can be tested	Few possibilities Least squares No best fit test
Parameter Estimator	Relatively few iterative training for non-linear; else noniterative computer time	Relatively many (weights) Iterative training Severe demands on computer time
Outputs	Calculate uncertainties for parameter estimates and predicted values. Residual diagnostic can provide physical insight	Response surfaces (splines) can be multivariate vectors No uncertainty computations Minimal diagnostics
Computer power Required	Low	High Parallel processing possible
Trends	Evolutionary techniques not yet used.	Evolutionary design possible

ACKNOWLEDGEMENT

The authors thank the referees for their valuable comments and suggestions.

(Received December 17, 2001.)

REFERENCES

[1] H. Allende and J. Galbiati: Robust test in time series model. J. Interamerican Statist. Inst. *1* (1996), 48. 35–79.
[2] H. Allende and S. Heiler: Recursive generalized M-estimates for autoregressive moving average models. J. Time Ser. Anal. *13* (1992), 1–18.
[3] B. Anderson and J. Moore: Optimal Filtering. Prentice Hall, Englewood Cliffs, N.J. 1979.
[4] W. G. Baxt: Use of an artificial neural network for data analysis in clinical decision marking: The diagnosis of acute coronary occlusion. Neural Computational *2* (1990), 480–489.
[5] J. M. Benitez, J. L. Castro, and J. Requena: Are neural network black boxes? IEEE Trans. Neural Networks *8* (1997), 1156–1163.
[6] J. Beran: Statistics for Long-memory Processes. Chapman and Hall, London 1994.
[7] B. L. Bowerman and R. T. O'Connell: Forecasting and time series: an applied approach. Third edition. Duxbury Press, 1993.
[8] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel: Time Series Analysis, Forecasting and Control. Third edition. Prentice Hall, Englewood Cliffs, N.J. 1994.
[9] L. Breiman, J. Friedman, R. Olshen, and C. J. Stone: Classification and Regression Trees. Belmont, C. A. Wadsworth, 1984.
[10] P. J. Brockwell and R. A. Davis: Time Series Theory and Methods. Springer Verlag, New York 1991.
[11] R. G. Brown: Smoothing, Forecasting and Prediction of Discrete Time Series. Prentice Hall, Englewood Cliffs, N.J. 1962.
[12] C. Chatfield: Forecasting in the 1990s. Statistician *4* (1997), 46, 461–473.
[13] B. Cheng and D. M. Titterington: Neural networks: review from a statistical perspective. Statist. Sci. *1* (1994), 2–54.
[14] J. T. Connor and R. D. Martin: Recurrent neural networks and robust time series prediction. IEEE Trans. Neural Networks *2* (1994), 5, 240–253.
[15] N. Crato and B. K. Ray: Model selection and forecasting for long-range dependent processes. Internat. J. Forecasting *15* (1996), 107–125.
[16] T. L. Fine: Feedforward Neural Network Methodology. Springer, New York 1999.
[17] B. Flury and H. Riedwyl: Multivariate Statistics: A Practical Approach. Chapman Hall, London 1990.
[18] J. H. Friedman: Multivariate adaptive regression spline. Ann. Statist. *19* (1991), 1–141.
[19] K. I. Funahashi: On the approximate realization of continuous mappings by neural networks. Neural Networks *2* (1989), 183–192.
[20] J. Han, C. Moraga, and S. Sinne: Optimization of feedforward neural networks. Engrg. Appl. Artificial Intelligence *2* (1996), 9, 109–119.
[21] K. Hornik, M. Stinchcombe, and H. White: Multilayer feedforward networks are universal approximators. Neural Networks *2* (1989), 359–366.
[22] R. M. Hristev: Artificial Neural Networks. Preprint of a book obtained via Internet from the author, 1998.
[23] J. M. Hutchinson: A Radial Basis Function Approach to Financial Time Series Analysis. Ph.D. Thesis. Massachusetts Institute of Technology, 1994.
[24] J. T. G. Hwang and A. A. Ding: Prediction for artificial neural networks. J. Amer. Statist. Assoc. *92* (1997), 438, 748–757.
[25] J. L. Lin and C. W. Granger: Forecasting from non-linear models in practice. Internat. J. Forecasting *13* (1994), 1–9.
[26] R. P. Lippmann: An introduction to computing with neural nets. IEEE ASSP Magazine (1997), 4–22.

[27] G. M. Ljung and G. E. P. Bax: On a measure of lack of fit in time series models. Biometrika *65* (1978), 297–303.
[28] P. McCullagh and J. A. Nelder: Generalized Linear Models. Chapman Hall, London 1989.
[29] J. S. Meditch: Stochastic Optimal linear Estimation and Control. MacGraw–Hill, New York 1969.
[30] J. E. Moody and J. Utans: Architecture selection strategies for neural networks. In: Refenes A. P. N. Neural Networks in the Capital Markets, Wiley, New York 1995.
[31] C. Moraga: Properties of parametric feedforward neural networks. In: XXIII Conferencia Latinoamericana de Informática, Valparaíso 1997, Vol. 2, pp. 861–870.
[32] F. J. Pineda: Generalization of Backpropagation to recurrent and higher order networks. In: Proc. IEEE Conf. Neural Inform. Proc. Syst., 1987.
[33] R. Poli, S. Cagnoni, G. Coppini, and G. Walli: A neural network expert system for diagnosing and treating hypertension. Computer (1991), 64–71.
[34] A. P. N. Referes and A. D. Zapranis: Neural model identification, variable selection and model adequacy. J. Forecasting *18* (1999), 299–322.
[35] G. C. Reinsel: Elements of Multivariate Time Series Analysis. Springer Verlag, New York 1993.
[36] B. D. Ripley: Statistical aspects of neural networks. In: Networks and Chaos-Statistical and Probabilistic Aspect (O. E. Barndorf–Nielsen, J. L. Jensen, and W. S. Kendall, eds.), Chapman and Hall, London 1993.
[37] W. S. Sarle: Neural networks and statistical methods. In: Proc. of the 19th Anual SAS Users Group International Conference, 1994.
[38] J. Smith and S. Yadav: Forecasting cost incurred from unit differencing fractionally integrated processes. Internat. J. Forecasting *10* (1994), 507–514.
[39] H. S. Stern: Neural networks in applied statistics. Technometrics *38* (1996), 3, 205–214.
[40] T. Subba Rao: On the theory of bilinear models. J. Roy. Statist. Soc. Ser. B *43* (1981), 244–255.
[41] H. J. Sussmann: Uniqueness of the weights for minimal feedforward nets with a given input-output map. Neural Networks *5* (1992), 589–593.
[42] K. H. Temme, R. Heider, and C. Moraga: Generalized neural networks for fuzzy modeling. In: Proc. Internat. Conference of European Society of Fuzzy Logic and Technology, EUSFLAT'99 Palma de Mallorca 1999.
[43] H. Tong: Non-linear Time Series. Oxford University Press, Oxford 1990.
[44] V. Vapnik: The Nature of Statistical Learning Theory. Springer Verlag, Berlin 1995.
[45] V. Vapnik and A. Chervoneski: The necessary and sufficient conditions for consistency of the method of empirical risk minimization. Pattern Recognition Image Anal. *1* (1991), 284–305.
[46] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang: Phoneme recognition using time delay neural networks. IEEE Trans. Acoust. Speech Signal Process. *37* (1989), 324–329.
[47] F. Y. Wu and K. K. Yen: Application of neural network in regression analysis. In: Proc. 14th Annual Conference on Computers and Industrial Engineering, 1992.

Prof. Dr. Héctor Allende O and MsCs. Rodrigo Salas, Universidad Técnica Federico Santa María, Departamento de Informática, Casilla 110-V, Valparaíso. Chile.
e-mail: hallende, rsalas@inf.utfsm.cl

Prof. Dr. Claudio Moraga, University of Dortmund, Department of Computer Science, D-44221 Dortmund, Germany and Technical University of Madrid, Department of Artificial Intelligence. Spain.
e-mail: moraga@cs.uni-dortmund.de