Journal of Official Statistics, Vol. 17, No. 4, 2001, pp. 447–456

A Neural Network Model for Predicting Time Series with Interventions and a Comparative Analysis

M.D. Cubiles-de-la-Vega[1], *R. Pino-Mejías*[1,2], *J.L. Moreno-Rebollo*[1,2] *and J. Muñoz-García*[1]

A procedure for designing a multilayer perceptron for predicting time series with interventions is proposed. It is based on the generation, according to a rule emerging from an ARIMA model with interventions previously fitted, of a set of nonlinear forecasting models with interventions. Each model is approximated through a three-layered perceptron, selecting the one minimizing the Bayesian Information Criterion. The training of the multilayer perceptron is performed by three alternative learning rules, incorporating multiple repetitions, and the hidden layer size is computed by means of a grid search. A comparative analysis using time series from the Active Population Survey in Andalusia, Spain, shows a better performance of these neural network models over ARIMA models with interventions.

Key words: Backpropagation; backpropagation with momentum; Levenberg-Marquardt; mean squared error; mean absolute deviation; MATLAB; SAS; active population survey.

1. Introduction

Artificial Neural Networks (ANNs) provide a great variety of mathematical nonlinear models, useful for tackling different statistical questions. Thus, a good review of this area from a statistical perspective is Cheng and Titterington (1994), while Ripley (1993, 1994, 1996) and Bishop (1995) cover classification problems. Nordbotten (1996) presents experimental research with neural network imputation models. In time series analysis, Hill et al. (1994) point out that the results achieved by neural network models can be similar to those obtained by traditional statistical methods, but it continues to be necessary to delve deeply into comparing the efficiency of ANN forecasting models, taking into account topics like architecture network, determining the size of hidden layers, learning algorithms, error measures and alternative statistical procedures.

This article belongs to this research area, but considers the problem of intervention analysis. Many economic and official time series present interventions, existing statistical procedures which take into account those phenomena, for example ARIMA models with interventions (Mills 1990). We outline in this article a methodology for fitting this sort of models, based on the previous fit of an ARIMA model with interventions. The constructed neural networks provided, over a collection of Andalusian labour force time series, a better performance than ARIMA models with interventions.

[1] Departamento de Estadística e Investigación Operativa, Universidad de Sevilla.
[2] Centro Andaluz de Prospectiva.
Correspondence: Rafael Pino-Mejías, Departamento de Estadística e I.O., Facultad de Matemáticas, Avda. Reina Mercedes, s/n, 41012 Sevilla, Spain. E-mail: rafaelp@cica.es.
Acknowledgments: The authors are very grateful to the JOS Associate Editor, Professor Estela Bee Dagum, and an anonymous referee, for very helpful comments.

2. ARIMA Models with Interventions

The general expression for a multiplicative seasonal ARIMA model with m interventions and periodicity s (Box and Tiao 1975), is:

$$\nabla^d \nabla_s^D x_t = \sum_{i=1}^{m} \frac{U_i(B)}{S_i(B)} \nabla^d \nabla_s^D \delta_i^t + \theta_0 + \frac{\phi(B)\Phi(B^s)}{\theta(B)\Theta(B^s)} a_t \tag{1}$$

The classical seasonal multiplicative ARIMA$(p, d, q)(P, D, Q)_s$, where we use the standard Box and Jenkins (1976) notation, including the nonseasonal ARIMA(p, d, q) model when $P = D = Q = 0$, is defined by the following components in (1):

the lag operator B, where $Bx_t = x_{t-1}$, and the seasonal lag operator B^s, $B^s x_t = x_{t-s}$,

the difference operator $\nabla = 1 - B$, and the seasonal difference operator $\nabla_s = 1 - B^s$, used d and D times, respectively, to achieve stationarity, and the constant θ_0

$\phi(B)$ and $\Phi(B^s)$ are the autoregressive polynomial terms, with p and Ps degrees, being

$$\phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p, \Phi(B^s) = 1 - \Phi_1 B^s - \ldots - \Phi_P B^{Ps}$$

$\phi(B)$ and $\Phi(B^s)$ are the moving average polynomial terms, with q and Qs degrees, being

$$\theta(B) = 1 - \theta_1 B - \ldots - \theta_q B^q, \Theta(B^s) = 1 - \Theta_1 B^s - \ldots - \Theta_Q B^{Qs}$$

a_t is a white noise process, so a_t $i.i.d \sim N(0, \sigma^2)$

We can also consider an additive seasonal ARIMA model, where the product $\phi(B)\Phi(B^s)$ is replaced by a polynomial term $1 - \phi_1 B - \ldots - \phi_p B^p - \Phi_1 B^s - \ldots - \Phi_P B^{Ps}$, and similarly, the product $\theta(B)\Theta(B^s)$ is replaced by $1 - \theta_1 B - \ldots - \theta_q B^q - \Theta_1 B^s - \ldots - \Theta_Q B^{Qs}$.

In addition to the classical ARIMA$(pd, q)(P, D, Q)_s$ model, we see in (1) m terms, one for each intervention. The m auxiliary variables $\delta_1^t, \delta_2^t, \ldots, \delta_m^t$ are associated with interventions occurring in time periods $r_1, r_2, \ldots, r_m$, respectively, and their definition depends on the nature of the intervention.

For a permanent intervention;

$$\delta_i^t = \xi_i^t = \begin{cases} 1 & if \quad t \geq r_i \\ 0 & if \quad t < r_i \end{cases} \tag{2}$$

For a transitory intervention;

$$\delta_i^t = (1 - B)\xi_i^t = \begin{cases} 1 & if \quad t = r_i \\ 0 & if \quad t \neq r_i \end{cases} \tag{3}$$

There exist diverse forms for $U_i(B)$ and $S_i(B)$, so it is possible to accommodate in the model (1) several types of responses to the interventions (see Mills 1990 for example), but it is not an easy task to arrive at the best ARIMA model with interventions. Given that in our comparative study we only had two possible interventions, we decided to follow a straight procedure to identify an ARIMA model with interventions, as is

described by Liu (1992). Next we summarize this procedure for a possible intervention in time period r:

1) Try a sudden and transitory intervention:
$$\frac{U_0}{1 - S_1 B}(1 - B)\xi^t$$
If S_1 is larger than or equal to 1, or near 1, we can exclude a transitory effect.
2) Try a gradual and permanent intervention:
$$\frac{U_0}{1 - S_1 B}\xi^t$$
If S_1 is not significantly different from 0, we discard a gradual effect. If $S_1 = 1$, there exists a linear trend.
3) Finally, try a sudden and permanent effect:
$$U_0 \xi^t$$
If this model is not valid, we discard an intervention in time period r.

Thus we can define interventions of Types 1, 2, and 3 corresponding to each of the three intervention models defined in the three steps of the last algorithm.

3. A General Nonlinear Forecasting Model with Treatment of Interventions

Given a univariate time series $\{x_t, t = 1, 2, \ldots, n\}$, where $x_t \in R$, ANN forecasting models usually suppose that each observed value is an unknown nonlinear function F of c lags $t_1, t_2, \ldots, t_c$:

$$x_t = F(x_{t-t_1}, x_{t-t_2}, \ldots, x_{t-t_c}) + \varepsilon_t$$

where the error ε_t is of zero mean.

Next, we suppose that m interventions, in time periods $r_1, r_2, \ldots, r_m$, have been detected. We define m auxiliary variables $\delta_1^t, \delta_2^t, \ldots, \delta_m^t$, depending on the nature of the intervention, as in (2) and (3). So we consider the next nonlinear forecasting model with c lags $t_1, t_2, \ldots, t_c$ and m interventions:

$$x_t = F(x_{t-t_1}, x_{t-t_2}, \ldots, x_{t-t_c}, \delta_1^t, \delta_2^t, \ldots, \delta_m^t) + \varepsilon_t \tag{4}$$

There remains the problem of obtaining $c, t_1, t_2, \ldots, t_c$. Following Cubiles de la Vega (1999), we considered the nonintervention component in the expression of the fitted ARIMA model. Thus several models were constructed, introducing some variability in the complexity of the ANN models. Concretely, we consider:

i) All of the possible models where c varies between 1 and k, being $k = Max\{p + Ps, q + Qs\}$.
ii) The model with every lag between 1 and p, and every lag between bs and $bs + p$, $b = 1, 2, \ldots, P$.
iii) The model with every lag between 1 and q, and every lag between bs and $bs + q$, $b = 1, 2, \ldots, Q$.
iv) The model with every lag between 1 and p, and every lag of the form bs, $b = 1, 2, \ldots, P$.
v) The model with every lag between 1 and q, and every lag of the form bs, $b = 1, 2, \ldots, Q$.

vi) When $p = q = P = Q = 0$, we construct models similar to those of ii) and iv), but based on d (instead of p) and D (instead of P).

For example, from the model ARIMA(0 1 1)(0 1 1)$_4$, we considered next lag sets: {1}, {1 2}, {1 2 3}, {1 2 3 4}, {1 2 3 4 5} (i), and also, {1 4 5} (iii), and {1 4} (v).

The unknown function F of (4) must be approximated by an appropriate model. Given the nonlinear nature of this task, ANNs provide a convenient framework to attack the job, standing out the multilayer perceptron.

4. The Multilayer Perceptron

4.1. Definition

A multilayer perceptron is a feedforward artificial neural network with three or more layers. The layers between the first and the last one are the hidden layers. The first layer, or input layer, is formed by **k** nodes, corresponding to an input vector $(x_1, x_2, \ldots, x_k)$. The last layer, or output layer, is formed by q nodes, so the network output is a vector $y = (y_1, y_2, \ldots, y_q)$. The multilayer perceptron was devised as a mathematical model for approximating a function ϕ: $A \subseteq R^k \rightarrow R^q$. The approximation is based on the network training (or learning) starting from n training patterns $(x^{(l)}, y^{(l)})$, being $y^{(l)} = \phi(x^{(l)})$, $l = 1, 2, \ldots, n$.

Let H be the hidden layer size, $\{v_{ih}, i = 0, 1, 2, \ldots, k, h = 1, 2, \ldots, H\}$ the synaptic coefficients for the connections between the input and the hidden nodes, and let $\{w_{hj}, h = 0, 1, 2, \ldots, H, j = 1, 2, \ldots, q\}$ be the synaptic coefficients for the connections between the hidden and the output layers. The output of each hidden layer, s_h, $h = 1, 2, \ldots, H$, is computed by applying an activation (or transfer) function, g, to its net input, $m_h = v_{0h} + \sum_{i=1}^{k} v_{ih} x_i$, so $s_h = g(m_h)$. Similarly, each output node produces a value $y_j, j = 1, 2, \ldots, q$, obtained by means of an activation function f, $y_j = f(t_j)$, being t_j the net input of the output node j:

$$y_j = f(t_j) = f\left(w_{0j} + \sum_{h=1}^{H} w_{hj} s_h\right) = f\left(w_{0j} + \sum_{h=1}^{H} w_{hj} g\left(v_{0h} + \sum_{i=1}^{k} v_{ih} x_i\right)\right)$$

The last expression shows clearly that each network output, y_j, $j = 1, 2, \ldots, q$, is a nested function, usually nonlinear, of the input values $(x_1, x_2, \ldots, x_{\mathbf{k}})$. The total number of parameters for a three-layered perceptron, M, is

$$M = (k + 1)H + (H + 1)q = (k + q + 1)H + q$$

Several theoretic properties support the multilayer perceptron, among which we can point out the universal approximate one (Ripley 1996), where the sigmoid activation function is:

$$g(x) = \frac{e^x}{1 + e^x}$$

THEOREM. Any continuous function ϕ: $A \subseteq R^k \rightarrow R^q$, A being a compact set, can be uniformly approximated by means of a three layered perceptron, with sigmoid activation functions in the hidden layers and identify activation function in the output layer.

This is the network architecture we adopted for the multilayer perceptron in our work.

4.2. Learning rules

Once the parameters k, H, q, and the activation functions are fixed, we need an algorithm to obtain appropriate values for the synaptic coefficients. In this article we consider three of the learning rules we can find in MATLAB 5.1.0.421 (Demuth and Beale 1994): Backpropagation, Backpropagation with momentum, and Levenberg-Marquardt, all of them intended to minimize the mean squared error over the training set.

From some preliminary studies we performed over the time series analyzed in our comparative study, we did not find any important effect in varying the parameters controlling the algorithms, so our programs worked with the default parameter values assigned by MATLAB. However, the initial values of the synaptic coefficients showed a more important effect on the final mean squared error. So we wrote MATLAB programs to repeat each algorithm 50 times, selecting the set of synaptic coefficients leading to the minimum mean squared error. This procedure is intended to avoid the lack of global minimum property of these algorithms also.

4.3. Determining the hidden layer size

The multilayer perceptron we used in our comparative study has the following features:

1. The number of input nodes k is equal to the sum of the number of lags c and interventions m, $k = c + m$. The input vector is $(x_{t-t_1}, x_{t-t_2}, \ldots, x_{t-t_c}, \delta_1^t, \delta_2^t, \ldots, \delta_m^t)$.
2. The sigmoid activation functions in the hidden layer.
3. An output node, $q = 1$, with the identity activation function, providing the forecast $\hat{x}_t$ of x_t.
4. A learning rule as is described in Section 4.2.

Smith (1993) suggests as the hidden layer size the average of k and q. From this advice, we introduced in our work a grid search for H around that value:

i) Compute $a = [(c + m + 1)/2]$ where [ ] denotes the nearest integer.
ii) For $h = \max\{0, a - 3\}$ to $a + 3$, construct a multilayer perceptron with $c + m$ input nodes, h hidden layers, and one output layer. Apply the learning rule, with 50 repetitions, over the training set.
iii) Select for H that value of h yielding the minimum MSE over the training set.

The MATLAB code we wrote is contained in Cubiles de la Vega (1999).

5. A Comparative Study

5.1. Outline of the study

We considered 33 quarterly time series from the Active Population Survey in Andalusia, Spain, from 1977 to 1997. These series contain the total of active, employed and unemployed people, each of the three classes being classified into 11 categories: total, total of men, total of women, 16 to 19 years old, 20 to 24 years old, 25 to 54 years old, older than 54, Agriculture, Industry, Construction and Services. The 33 series were identified, following this classification, by A1 to A11, E1 to E11, and U1 to U11. Each series

was split into a training set and a test set: the test set is formed by the 8 cases corresponding to the years 1996 and 1997, while the 76 cases corresponding to the period 1997–1995 form the training set, used for fitting the forecasting models.

Some definitions of the variables measured by the Active Population Survey were changed in the second quarter of 1987. Though the series of the Survey were linked by the Spanish National Statistical Institute, we could expect the presence of an intervention in that time period. Furthermore, the time series plots revealed a clear intervention in the first quarter of 1984 for several series. So we considered the analysis of the possible presence of interventions in both time periods. Thus, for each of the 33 series, when it came to the training set, a multiplicative or additive and possibly seasonal ARIMA model was fitted. Box-Jenkins' methodology for identifying one or more ARIMA models (Box and Jenkins 1976) was used, and we incorporated, in the case of each of these models, the procedure outlined in Section 2 in order to identify interventions. So for each series we had one or more ARIMA models with interventions. A model was considered valid when i) all the coefficients were five percent significative, and ii) the residuals were not found to be correlated by means of Ljung-Box test. When several models were considered valid, the minimum Bayesian Information Criterion was used to select the best model.

This procedure was developed using the ETS module of SAS v7.5.2, SAS Institute Inc. (1993), available in Asterix node of Andalusian Scientific Computing Centre (CICA), and employing maximum-likelihood estimation. From it, we found one or two interventions in 25 of the 33 series.

For each series, and from the identified ARIMA model with interventions, we constructed several lag sets as in Section 3. The nonlinear models so constructed were approximated by means of a multilayer perceptron, constructed as is described in Section 4. The forecasting model with the minimum Bayesian Information Criterion was selected.

5.2. Results

Following the procedure of Section 5.1, for each of the 25 series where we identified at least one intervention, we analyzed and compared the results obtained with ARIMA models with interventions and the multilayer perceptron.

Tables 1, 2 and 3 contain, for each of these 25 series, the fitted ARIMA model with interventions, all of them with log transformation, where (+) denotes that the ARIMA model is additive. The second column contains the lags that, with the corresponding auxiliary variables, define the input layer of the selected multilayer perceptron. The characters 1, 2 and 3 in the intervention columns denote the type of intervention, as is described in Section 2, using a 0 for no intervention in the associated time period. Numbers surrounded by parentheses are the only nonzero coefficients in the corresponding term.

We observe that the unemployed series need more complex multilayer perceptrons, whereas for the active series simpler multilayer perceptrons are sufficient, thus revealing a dependence on a larger number of lagged values for the unemployed series.

Table 4 contains the numbers of series where each method has the minimum test forecasting error among the four methods, as measured by the mean squared error (MSE) and

Table 1. Lag sets and ARIMA models for the Active Population Series

Series	Lags	ARIMA model (log)						Interventions	
		P	*d*	*q*	*P*	*D*	*Q*	1984	1987
A1	1, 4	0	1	1	0	1	1	3	3
A2	1	0	1	0	0	0	0	3	0
A4	1, 2, 3	0	1	3	0	0	0	0	3
A5	1, 4	0	1	1	0	0	1	0	3
A6	1, 4	0	1	1	0	1	1	3	2
A10(+)	1, 4, 5	1	1	0	1	1	0	3	1
A11	1, 4	0	1	1	0	1	1	0	3

Table 2. Lag sets and ARIMA models for the Employment Series

Series	Lags	ARIMA model (log)						Interventions	
		p	*d*	*q*	*P*	*D*	*Q*	1984	1987
E1	1, 4	0	1	1	0	0	1	3	0
E2(+)	1	0	1	(2)	0	0	1	3	0
E3	1, 4	0	1	1	0	1	1	0	2
E5	1	0	1	(2 5)	0	0	0	3	0
E6 (+)	1, 4	0	1	1	0	0	1	3	0
E7	1, 4	0	1	1	0	1	1	0	1
E8	1, 4	0	1	1	0	1	1	1	2
E10	1, 2, 3, 4, 5	0	1	(2)	0	1	1	2	0
E11	1, 2, 3, 4	0	1	0	0	1	1	2	0

Table 3. Lag sets and ARIMA models for the Unemployment Series

Series	Lags	ARIMA model (log)						Interventions	
		P	*d*	*Q*	*P*	*D*	*Q*	1984	1987
U1	1, 2, 3, 4, 5	0	1	(3 5)	0	0	0	3	0
U2	1, 2, 3	0	1	3	0	0	0	3	0
U4	1, 4	0	1	1	0	0	1	0	3
U5	1, 2, 3, 4, 5	0	1	(5)	0	0	1	3	0
U6(+)	1, 2, 3, 4, 5	0	1	1	0	0	1	3	0
U7	1, 2, 3, 4, 5	0	1	1	0	1	1	2	3
U8	1, 4	0	1	1	0	1	1	2	0
U9	1	0	1	0	0	0	0	0	3
U10	1, 2, 3, 4, 5, 6, 7	(3)	1	0	1	0	0	3	3

Table 4. Number of series minimizing the tests MSE and MAD

Method	MSE	MAD
Backpropagation	6	7
Backpropagation with momentum	9	7
Levenberg-Marquardt	2	4
ARIMA models	8	7

the mean absolute deviation criteria (MAD), so obtaining a comparison of the generalization capacity of the four methods.

From Table 4 we see that in 17 of the 25 series the minimum MSE test was found in a multilayer perceptron model, 9 of these 17 models being trained with backpropagation with momentum. In 18 of the 25 series the minimum MAD test was found in a multilayer perceptron model, and 14 of these 18 models were trained with backpropagation (7) or backpropagation with momentum (7). Further, in the series where Levenberg-Marquardt trained models achieved the minimum MSE or MAD test, the second best model were a backpropagation or backpropagation with momentum multilayer perceptron model. This brings out the superiority of the forecasting models based on the multilayer perceptron, trained with backpropagation and backpropagation with momentum particularly.

Figure 1 exhibits the boxplots for the MSE in the training and test sets. A similar graphic for the MAD is shown in Figure 2. From Figures 1 and 2, we observe that backpropagation and backpropagation with momentum tend towards lower MSE and MAD values in the test set, in comparison with the other methods. Levenberg-Marquardt achieves very low values in the training set, but at the price of a poor capacity of generalization, revealing a clear overfitting with regard to the training test.

From Figure 2, backpropagation with momentum offers poor results with regard to the training set, but its capacity of generalization, measured by the MAD, is similar to that of the other methods. There is a clear difference in Figure 2 between the performances of the three learning rules, the multilayer perceptron trained with backpropagation being the method yielding the best compromise between the training and test set.

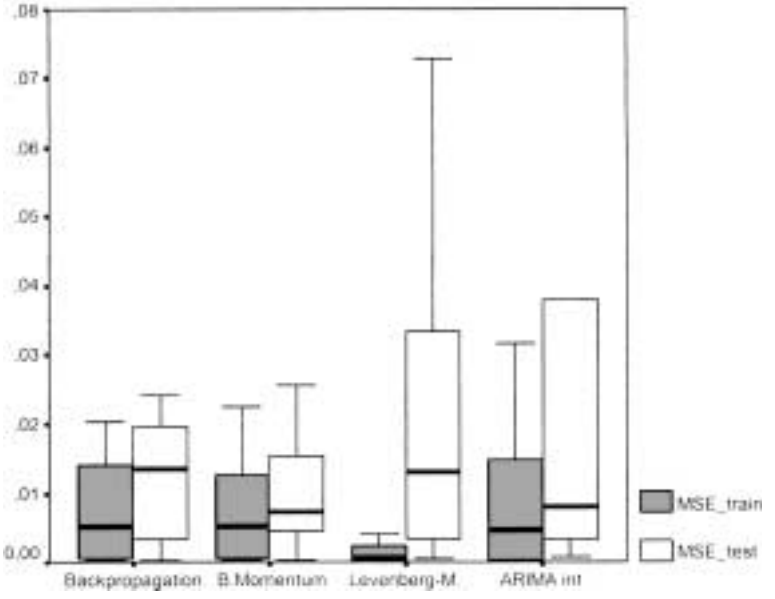

Fig. 1. Mean squared error in the training and test sets

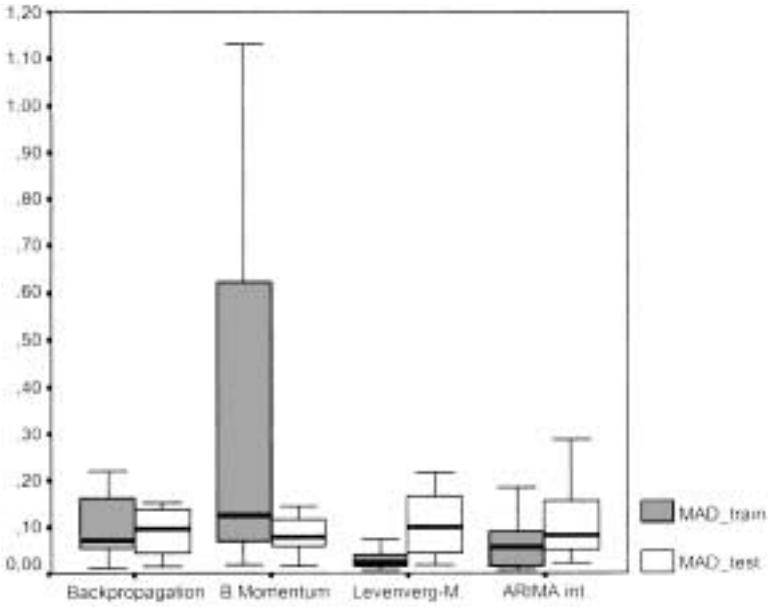

Fig. 2. Mean absolute deviation in the training and test sets

6. Concluding Remarks

i) In a majority of the considered series, the presented procedure for designing forecasting ANN models, with treatment of interventions, provided better results than those obtained with ARIMA models with treatment of interventions.

ii) The multilayer perceptron, trained with MATLAB default parameters for the backpropagation and backpropagation with momentum learning rules, but introducing multiple repetitions and a grid search for the hidden layer size, offers a good performance. However, Levenberg-Marquardt trained models exhibit an overfitting behaviour.

iii) The best compromise between the test and training errors, measured with MSE and MAD criteria, is achieved by the multilayer perceptron trained with backpropagation.

iv) Artificial neural networks trained by several learning rules, combined with the preview study of the series with Box-Jenkins methodology, and selecting the model through the Bayesian Information Criterion, provide a valuable framework for obtaining univariate predictions of the labour force time series, incorporating the treatment of interventions in the forecasting model.

7. References

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

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

Box, G.E.P. and Tiao, G.C. (1975). Intervention Analysis with Application to Economic

and Environmental Problems. Journal of the American Statistical Association, 70, 70–79.

Cheng, B. and Titterington, D.M. (1994). Neural Networks: A Preview from a Statistical Perspective. Statistical Science, 9, 1, 2–54.

Cubiles de la Vega, M.D. (1999). Redes de Neuronas Artificiales en el Análisis de Series Temporales. Ph.D. thesis, Universidad de Sevilla.

Demuth, H. and Beale, M. (1994). Neural Network TOOLBOX for Use with MATLAB, User's Guide. The Math Works Inc.

Hill, T., Marquez, L., O'Connor, M., and Remus, W. (1994). Artificial Neural Networks for Forecasting and Decision Making. International Journal of Forecasting, 10, 5–15.

Mills, T.C. (1990). Time Series Techniques for Economists. Cambridge: University Press.

Nordbotten, S. (1996). Time Network Imputation Applied to the Norwegian 1990 Population Census Data. Journal of Official Statistics, 12, 4, 385–401.

Ripley, B.D. (1993). Statistical Aspects of Neural Networks. In Networks and Chaos, Statistical and Probability Aspects, eds O. Barndorf-Nielsen et al., Chapman and Hall.

Ripley, B.D. (1994). Neural Networks and Related Methods for Classification. Journal of the Royal Statistical Society, Series B, 56, 3, 409–456.

Ripley, B.D. (1996). Pattern Recognition and Neural Networks. Cambridge: University Press.

SAS Institute Inc. (1993). SAS/ETS User's Guide, Version 6, Second Edition. Cary, NC: SAS Institute Inc.

Smith, M. (1993). Neural Networks for Statistical Modeling. Van Nostrand Reinhold.

Received May 2000
Revised March 2001