Artificial Neural Networks and Automatic Time Series Analysis: methodological approach, results and examples using health-related time series

Belén García Cárceles , Jose M. Pavía Miralles, Bernardí Cabrer Borras

Economic Analysis, University of Valencia (Spain)

Abstract

The work summarized in this paper explores the possibilities offered by statistical tools based on artificial neural networks for pattern recognition. Results offered here come from a research line with which it is pretend to stablish under which conditions automatic forecasting tools for time series offer significant advantage. In this context, artificial neural networks are applied to detect and determine those conditions.

Keywords: Time series; Forecasting Accuracy; Health Economic modeling; TRAMO; SEATS; ARIMA.

MSC Codes: 91B84, 37M05, 62M10.

Acknowledgment: Belén García Cárceles acknowledges the "Atracció de Talent" scholarship provided by University of Valencia. This work was supported by the Spanish Ministry of Economics and Competitiveness under grant CSO2013-43054-R.

Mail Address: belen.garcia-carceles@uv.es. Economic Analysis, University of Valencia (Spain). Av Tarongers, s/n, 46022-Valencia. Phone: +34963828404, Fax: +34963828415.

INTRODUCTION

The work summarized in this paper explores the possibilities offered by statistical tools based on artificial neural networks for pattern recognition and continuous variables forecasting.

Results offered here come from a research line with which it is pretend to stablish under which conditions automatic forecasting tools for time series offer significant advantage. In this context, artificial neural networks are applied to detect and determine those conditions.

Background:

First step was designing the methodology: different automatic estimation ARIMA modeling tools were compared using a hold-out test strategy (García Cárceles, et al., 2013). In this first exercise, different error accuracy measures were computed and it was designed a novelty method to evaluate different measures using Receiving Operation Curves (ROC) (García Cárceles, et al., 2014). After evaluating different accuracy measures and its dependency with the typology of the data they refer, once it is stablish a method to objectively quantify the precision of each measure produced, compare accuracy of different automatic forecasting procedures (all of those questions addressed in previous work), some issues of concern remained for analysis. Specifically, the question whether there are a priori elements that may affect the quality of the prediction when using certain automatic procedure.

Goal:

So the idea here, is to "reverse" the previous analysis. That is, as it is already stablished what an accuracy forecast is, and even it is possible to use ROC curves to decide whether a forecast can be considered, in fact, a hit or a failure, these information can be used as a training data to let neural networks help to detect if those a priori elements do exist.

ARTIFICIAL NEURAL NETWORKS IN CONTEXT

General idea:

The application of artificial neural networks, as a concept, to data processing has its origin in artificial intelligence works from the 40-50 which central interest was the construction of intelligent machines: teaching a machine how to process information similarly to how the brain does[1].

[1] A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects: 1. Knowledge is acquired by the network through a learning process. 2. Interneuron connection strengths known as synaptic weights are used to store the knowledge (Aleksander & Morton, 1990).

Biological origins of the concept, outlines the idea of data analysis using neural structures, that is: a connectionist system where simple processing units (nodes similar to neurons) are linked by connections which transmit a changeable numerical value (weight as a synapse) from one node to another.

Its main feature is that process information in parallel, that is, several neurons may be working (deciding) simultaneously. In addition, these systems are not programmed to perform a certain task, but "trained" using labelled examples as the training set, and from these to distil the essence of grouping. Therefore, in this analysis, we consider neural network as a process of "Machine Learning"[2] for pattern recognition (classification).

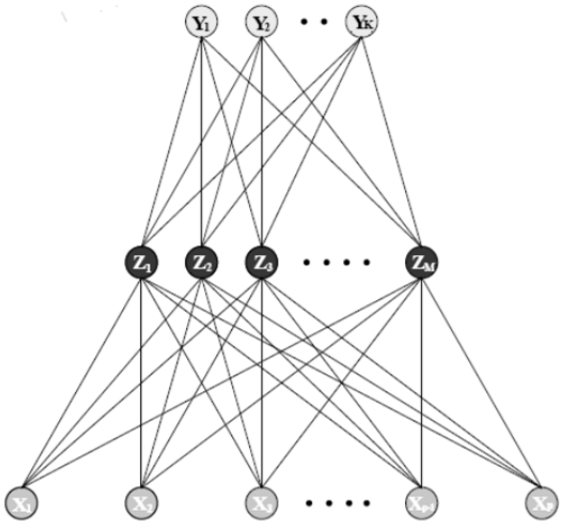

Figure 1: Schematic of a single hidden layer, feed-forward neural network. This is a typical representation of a two-stage regression or classification model (Hastie, et al., 2008, p. 393).

Specification and fitting the Back-propagation network structure model:

Typical network structure is proposed, named "back-propagation network" or "single layer perceptron" (see Figure 1), defined as a two-step classification or regression, described in detail in (Hastie, et al., 2008, pp. 392-396). We'd summarize here its approach in order to help the reader follow the conceptual line developed here and introduce the nomenclature used.

For K-class classification:

1- There are K target measurements Y_k, $k = 1,\ldots,K$, each being coded as a 0-1 variable for the kth class.

[2] "Machine Learning is generally taken to encompass automatic learning procedure based on logical or binary operations, that learn a task from a series of examples" (Michie, et al., 1994)

2- Derived features Z_m are created from linear combinations of the inputs, and then the target Y_k is models as a function of linear combinations of the Z_m. Those are the features in the middle of the network and they are usually called "hidden units" due to the fact that are not directly observed.

3- The output function $g_k(T)$ allows the final transformation of the vector of outputs T, concretely, for our K-class classification model, we'd use softmax function, because it's also the transformation used in the multilogit model, which produces positive estimates that sum to one and represent the probability of class k.

Formally:

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T X), \qquad m = 1, \dots, M,$$

$$T_k = \beta_{0k} + \beta_k^T Z, \qquad k = 1, \dots, K,$$

$$f_k(X) = g_k(T), \qquad k = 1, \dots, K,$$

Where: $Z = (Z_1, Z_2, \dots, Z_M)$, and $T = (T_1, T_2, \dots, T_K)$. The activation function is chosen to be sigmoid $\sigma(v) = 1/(1 + e^{-v})$, and so, the output function is defined as:

$$g_k(T) = \frac{e^{T_k}}{\sum_{l=1}^{K} e^{T_l}}$$

It is important to note that, in this context, parameters of the basis functions are learned from the data. Hence, a neural network can be thought of as a nonlinear generalization of the linear model.

So far, we have introduce model specification as a neural network model which has unknown parameters (weights) and the goal is to obtain values for them that make the model fit the training data well. As in (Hastie, et al., 2008, p. 395),We denote the complete set of weights by Θ, which consists of:

$$\{\alpha_{0m}, \alpha_m; m = 1,2, \dots, M\}\ M(p+1)\ weights,$$

$$\{\beta_{0k}, \beta_k; k = 1,2, \dots, K\}\ K(M+1)\ weights,$$

For classification we use cross-entropy (deviance):

$$R(\theta) = -\sum_{i=1}^{N}\sum_{k=i}^{K} y_{ik}\ log\ f_k(x_i)$$

Avoiding over fitting problems: Penalties

To avoid over fitted solutions, it is introduced a penalty term in the error function in two manners (see (Hastie, et al., 2008, p. 398) for more details), as neural networks have too many weights and are likely to over fit the data at the global minimum of $R(\theta)$.

1- Penalty error using *weight decay*: $R(\theta) + \lambda J(\theta)$, where $J(\theta) = \sum_{km} \beta_{km}^2 + \sum_{ml} \alpha_{ml}^2$

2- Penalty error using *weight elimination*: $R(\theta) + \lambda J(\theta)$, where $J(\theta) = \sum_{km} \frac{\beta_{km}^2}{1+\beta_{km}^2} + \sum_{ml} \frac{\alpha_{ml}^2}{1+\alpha_{ml}^2}$.

$\lambda \geq 0$ is a tuning parameter, usually estimated using cross-validation.

ACCURACY MEASURES AS TRAINING DATA

Preparation of data sets:

The preparation of data sets was performed as follows:

1- First place. Following methodology developed in previous work, real time series are used coming from economic and health area and obtained from different open-access sources. It is not desired to control neither by typology of ARIMA signal detected, nor by "decisions" made by a concrete forecast tool to fit it, but, exclusively, by its capacity to produce an accurate forecast. So it is selected an heterogeneous group of data sets, considering:

   a. Different frequency: annual (299 series), quarterly (6,039 series) and monthly (6,773 series).

   b. Different number of observations (from 40 to 600 observations).

   c. Different magnitude (scale ranges from 10^{-4} to 10^{9}).

2- N periods are removed from original time series (n=12 for monthly time series, n=4 for quarterly time series and n=5 for annual), because it is followed a *hold-out test* strategy. It is important to note, that this cut is not performed by the forecast tool used, so it is granted that the only information used by the tool to produce its forecasts are the data without the n periods removed.

3- Then, the forecast tool is applied. We will use TRAMO (Time Series Regression with ARIMA Noise Rev: 934 Build: 2014/12/17 17:32:59) and SEATS (Signal Extraction ARIMA Time Series Rev: 934 Build: 2014/12/17 17:32:59), software tools which are described below.

Automatic Forecast Tool description:

TRAMO is a program for estimation and forecasting of regression models with possibly nonstationary (ARIMA) errors and any sequence of missing values. SEATS is a program for estimation of unobserved components in time-series following the so-called ARIMA-model-based method, extracting the trend, seasonal, irregular, and cyclical components.

When using both applications assembled in its full automatic mode, this tool produces the following procedures:

1- Program test for log-level specification.

2- Pretest for the presence of Trading Day (TD), Leap Year (LY) and Eastern Effect (EE).

3- Automatic ARIMA model identification: (P D Q) (BP BD BQ)

4- Interpolates missing observations if any and computes their associated MSE (Mean Squared Error). No restriction is imposed on the location of missing observations in the series.

5- Outlier detection. Three types of outliers are considered: Additive (AO), Transitory Changes (TC) and Level Shifts (LS). The level of significance is set by the program and depends on the length of the series.

6- The full model is estimated by maximum likelihood.

7- Forecasts of the series up to two years horizon are computed, as well as their MSE.

8- The model is decomposed and optimal estimators and forecasts of the components are obtained. Components: trend-cycle, seasonal, irregular and transitory components.

A brief description of automatic procedure functions of TRAMO and SEATS can be found in (Gómez & Maravall, 1997, pp. 1,57). Si bien no es el objetivo de este análisis entrar a valorar la calidad del ajuste del modelo obtenido mediante el procedimiento automático de TRAMO SEATS, describiremos (aunque brevemente) aquí la base teórica que subyace al procedimiento de ajuste realizado por ambos programas para ofrecer una visión condensada de la cuestión en este documento.

TRAMO:

Given the vector of observations $z = (z_{t_1}, \dots, z_{t_M})$ where $0 < t_1 < \dots < t_M$, the program fits the regression model:

$$z_t = y'_t\beta + v_t, \qquad (1)$$

where $\beta = (\beta_1, \dots, \beta_n)'$ is a vector of regression coefficients, $y'_t = (y_{1t}, \dots, y_{nt})$ denotes n regression variables, and v_t follows the general ARIMA process:

$$\phi(B)\delta(B)v_t = \theta(B)a_t, \qquad (2)$$

Where B is the back shift operator; $\phi(B)$, $\delta(B)$ and $\theta(B)$, are finite polynomials in B, and a_t is assumed a n.i.i.d. $(0\ ,\ \sigma_a^2)$ white-noise innovation. The polynomial $\delta(B)$ contains the unit roots associated with differencing (regular and seasonal), $\phi(B)$ is the polynomial with stationary autoregresive roots (and the complex uni roots, if present), and $\theta(B)$ denotes the (invertible) moving average polynomial. In TRAMO, they assume the following multiplicative form:

$$\delta(B) = (1-B)^d(1-B^s)^D$$

$$\phi(B) = (1+\phi_1 B + \dots + \phi_p B^p)(1+\Phi_1 B^s + \dots + \Phi_P B^{sxP})$$

$$\theta(B) = (1 + \theta_1 B + \cdots + \theta_q B^q)(1 + \Theta_1 B^s + \cdots + \Theta_Q B^{sxQ})$$

where s denotes the number of observations per year. The model may contain a constant μ, equal to the mean of the differenced series $\delta(B)z_t$. In practice is estimated as one of the regression parameters in (1).

SEATS:

The model for the differenced series from TRAMO (presumably, the differences taken on the original series x_t achieves stationarity) can be expressed as:

$$\phi(B)(z_t - \bar{z}) = \theta(B)a_t, \qquad (3)$$

Where $\bar{z}$ is the mean of z_t, a_t is a white-noise series of innovations, normally distributed with zero mean and variance σ_a^2, $\phi(B)$ and $\theta(B)$are autoregressive (AR) and moving average (MA) polynomial in B, respectively. SEATS decomposes (decomposition can be multiplicative or additive) a series that follows model (3) into several components, considering: a) the trend component (x_{pt}); b) the seasonal component (x_{st}); c) the cyclical component (x_{ct}); and d) the irregular component (x_{ut}).

All components are derived from the ARIMA model detected: the trend component represent long-term evolution of the series, the seasonal component, captures the spectral peaks at seasonal frequencies; the cyclical component captures both fluctuations longer than a year and short term variations. The irregular component captures erratic, white noise behavior.

As a conclusion, automatic forecasting procedure performed by TRAMO and SEATS provide a fully model-based method for forecasting and signal extraction in univariate time series.

Description of the procedure for extracting forecasts:

The procedure is fully transparent and has been designed using the software R (R Development Core Team, 2008). The sequence of tasks run under R is as follows:

There are two processes in batch; the first one reads original files (downloaded in text format separated by coma, .csv file, from different data sources), extracts each time series to a separate file deleting last n data (which would be used later to quantify forecast accuracy) and the necessary strings to run TRAMO program are introduced. Second process runs TRAMO and stores its results, runs SEATS (using the input file automatically generated by TRAMO) and, again results are stored from the output files.

On completion, we obtain a database with the results of the estimation and forecasting carried out by TRAMO and SEATS on the automatic process. Specifically we are interested in forecast at different horizons, and its standard deviation to compute confidence interval.

In tables 1, 2 and 3, there are some of the results stored from the previous process. Each table refers to time series grouped by frequency (annual, quarterly and monthly data):

Table 1: Annual Series Summary Statistics

	Mean	SD	Max	Min	Approx 1% CV	Beyond 1% CV	% of series that pass the test (99%)
Length	34.95	3.56	36	22			
Num. of ARMA param. per serie	1.40	0.92	5	0			
Num. of outliers per serie	1.34	1.56	6	0			
Q	7.65	10.62	147.34	0.15	18.48	1.00	99.00
N	36.56	103.01	783.11	0.00	9.21	36.79	63.21
SK	-0.52	2.72	9.83	-10.86	2.58	25.75	74.25
Kur	2.68	4.66	25.84	-1.24	2.58	36.12	63.88
QS			0.00	0.00	9.21	0.00	100.00
Q2	6.11	4.28	23.61	0.08	20.09	1.67	98.33
Runs	0.03	1.86	5.12	-5.22	2.58	16.72	83.28

Source: Own elaboration.

Table 2: Quarterly Series Summary Statistics

	Mean	SD	Max	Min	Approx 1% CV	Beyond 1% CV	% of series that pass the test (99%)
Length	88.92	42.45	221	17			
Num. of ARMA param. per serie	1.80	0.99	7	0			
Num. of outliers per serie	1.75	2.24	28	0			
Q	14.81	5.75	85.42	1.13	29.14	1.01	98.99
N	13.05	153.30	8959.11	0.00	9.21	12.09	87.91
SK	-0.20	1.43	10.31	-21.92	2.58	5.13	94.87
Kur	0.97	3.17	92.08	-2.01	2.58	11.39	88.61
QS			28.44	0.00	9.21	0.58	99.42
Q2	17.76	13.86	180.55	0.00	30.58	8.41	91.59
Runs	-0.03	1.07	5.66	-7.42	2.58	2.07	97.93

Source: Own elaboration.

Table 3: Monthly Series Summary Statistics

	Mean	SD	Max	Min	Approx 1% CV	Beyond 1% CV	% of series that pass the test (99%)
Length	115.64	34.52	158	59			
Num. of ARMA param. per serie	2.19	0.81	7	0			
Num. of outliers per serie	1.17	1.54	14	0			
Q	22.78	6.94	90.14	4.28	40.29	0.81	99.19
N	2.13	3.76	95.03	0.00	9.21	3.16	96.84
SK	0.08	1.07	5.00	-4.52	2.58	1.99	98.01
Kur	0.11	0.99	8.68	-2.37	2.58	2.19	97.81
QS			18.96	0.00	9.21	0.27	99.73
Q2	23.38	9.51	224.43	2.01	42.98	3.37	96.63
Runs	0.00	0.88	4.80	-3.25	2.58	0.32	99.68

Source: Own elaboration.

Description of the analysis process using neural networks:

Accuracy of the each forecast is evaluated straightforward by checking if the real value is actually include in the forecast confidence interval, then we have a dichotomous result variable, which inform, in each case, if the automatic procedure has succeed with its forecast.

On the other hand, we have a set of characteristics which have been already described in the literature as the "necessary conditions" to produce accurate forecasts, and that are stored in the output data base produced by TRAMO and SEATS. Some of those characteristics are the accuracy of the model fit (Diebold & Mariano, 1995) (Peña, 2010) (Hamilton, 1994), the way the series is filtered by signal detection to obtain a stylized series to produce forecasts, the size of the series, the magnitude of the errors (Armstrong & Fildes, 1995), parameters considered from the regular and the irregular adjustment (Findley, 2005) (Mc Donald-Johnson, et al., 2007) (Tashman, 2000), outlier detection and its typology (Pavía Miralles, et al., 2012), among others.

After classifying forecast as good (1 = hit) or not (0 = fail), we can label each series to be introduced as a training example in the network model. A sensitive analysis is performed to assess the nature of the internal representations generated by the neural network to determine the importance or effect of each input variable on the output: the probability of produce an accurate forecast by the automatic tool.

All results will be shown and at the conference.

BIBLIOGRAPHY

Aleksander, I. & Morton, H., 1990. *An Introduction to neural computing.* London: Chapman & Hall.

Armstrong, J. S. & Fildes, R., 1995. On the selection of Error Measures for Comparisons Among Forecasting Methods. *Journal of Forecasting,* January, 14(1), pp. 67-71.

Diebold, F. X. & Mariano, R. S., 1995. Comparing Predictive Accuracy. *ournal of Business & Economic Statistics,* 13(3), pp. 253-263.

Findley, D. F., 2005. Some Recent Developments and Directions in Seasonal Adjustment. *Journal of Official Statistics,,* 21(2), pp. 343-365.

García Cárceles, B., Cabrer Borrás, B. & Pavía Miralles, J. M., 2014. Time Series Automatic Forecasting Procedure: Reassessing Accuracy Measures using ROC curves. *mimeo,* pp. 1-28.

García Cárceles, B., Pavía Miralles, J. M. & Cabrer Borrás, B., 2013. Automatic Forecasting: A comparison between TRAMO/SEATS and X-13-ARIMA performance. *mimeo,* pp. 1-28.

Gómez, V. & Maravall, A., 1997. *Programa TRAMO and SEATS, Instructions for the user,* Madrid: Banco de España.

Hamilton, J. D., 1994. Forecasting. In: *Time Series Analysis.* Princeton: Princeton University Press, pp. 72-116.

Hastie, T., Tibshirani, R. & Friedman, J., 2008. *The Elements of Statistical Learning. Data Mining, Infrence, and Prediction.* 2nd ed. Standfor: Srpinger.

Mc Donald-Johnson, K. M., Harvill Hood, C. C., Monsell, B. C. & Li, C., 2007. Comparing Automatic Modeling procedures of TRAMO and X-12-ARIMA, an Update. *U.S. Thensus Bureau.*

Michie, D., Spiegelhalter, D. J. & Taylor, C. C., 1994. *Machine Learning, Neural and Statistical.* New York: Ellis Horwood.

Pavía Miralles, J. M., Cabrer Borras, B. & Iranzo Pérez, D., 2012. Outlier detection with automatic modelling: TRAMO/SEATS versus X-12-ARIMA. *Model Assisted Statistics and Applications,* Issue 7, pp. 229-244.

Peña, D., 2010. *Análisis de series temporales.* 2ª ed. Madrid: Alianza.

R Development Core Team, 2008. *R: A language and environment for statistical computing.,* Vienna: R Foundation for Statistical Computing.

Tashman, L. J., 2000. Out-of-sample tests of forecasting accuracy: an analysis and review. *International Journal of Forecasting,* 16(4), pp. 437-450.