Statistical Characterization and Optimization of Artificial Neural Networks in Time Series Forecasting: The One-Period Forecast Case

Caracterización Estadística y Optimización de Redes Neuronales Artificiales para Pronóstico de Series de Tiempo: Pronóstico de un Solo Período

María Angélica Salazar Aguilar[1], Guillermo J. Moreno Rodríguez[2] and Mauricio Cabrera-Ríos[4]
[1]División de Posgrado en Ingeniería de Sistemas, Facultad de Ingeniería Mecánica y Eléctrica, Universidad Autónoma de Nuevo León, Monterrey, Nuevo León, MÉXICO
[4] Corresponding Author, Ph: +52(81) 1492 0637
[2] Ingeniería de Tráfico y Optimización de Capacidad, Avantel, Monterrey, Nuevo León, MÉXICO
{angy@yalma.fime.uanl.mx[1], gmoreno@avantel.com.mx[2], mcabrera@mail.uanl.mx[4]}

Article received on Augost 09, 2005; accepted on September 08, 2006

Abstract

Time series forecasting is an active area for the application of Artificial Neural Networks (ANNs). Although the selection of an ANN has been greatly simplified, it remains a challenge to adequately determine the ANN's parameters. In this work a method based on statistical analysis and optimization techniques is proposed to select the ANN´s parameters for application in time series forecasting. The results on the successful application of the method in a real demand forecasting problem for the telecommunications industry are also reported. .
Keywords: Artificial Neural Networks, Time Series Forecasting, Design and Analysis of Experiments.

Resumen

Los pronósticos de series de tiempo constituyen un área activa para la aplicación de Redes Neuronales Artificiales (RNAs). Aunque la selección de una RNA para tal aplicación se ha simplificado grandemente, la falta de un método establecido para asignar los parámetros de las RNAs de una manera adecuada sigue siendo un reto. En este trabajo se propone una metodología basada en técnicas estadísticas y optimización para la selección de parámetros de una RNA para el pronóstico de series de tiempo. La metodología propuesta se demuestra por medio de su aplicación en un problema real de pronóstico de demanda en la industria de las telecomunicaciones.
Palabras Clave: Redes Neuronales Artificiales, Series de tiempo, Análisis y Diseño de Experimentos, Pronósticos.

1. Introduction

Throughout history, forecasting the behavior of phenomena of interest has been important. This fact is reflected in the diversity of forecasting applications in practically all knowledge areas. When quantitative information regarding the behavior of a particular phenomenon with respect to time is available, then one has a time series. Forecasts in time series are commonly generated through traditional statistical techniques. The body of knowledge that encompasses these techniques is called Time Series Analysis.

Forecasts are generally the bases for decision-making at all levels ranging from the day-to-day operational decisions to the long-term strategical ones in many organizations. Given its importance, it is not surprising that forecasting had become a very active research area (Makridakis and Wheelwright, 1987; Zhang, 2004).

Use of linear forecasting methods such as moving averages, exponential smoothing, linear regression and time series decomposition have dominated the arena for decades. Especially significant in terms of usage is the technique called Auto-Regressive Integrated Moving Average (ARIMA), developed by Box and Jenkins (1976). In spite of the widespread use of these linear methods, the existence of nonlinear relationships in the time series can limit their application in many cases (Makridakis et al., 1982). Nonlinear relationships are, indeed, not uncommon at all in reality, thus making it necessary to resort to techniques capable to adequately reflect such behavior. Artificial Neural Networks (ANNs) have been suggested as an alternative forecasting technique owing to their ability to approximate nonlinear behavior. In fact, ANNs have also been shown to be competitive even when the case is one of linearity (Widrow et al., 1994; Hwarng, 2001; Medeiros et al., 2001; Zhang, 2001).

The first use of ANNs in time series forecasting can be traced back to Hu in 1964 (Zhang, et al., 1998). However, Hu´s work could not be completed because a training algorithm for multilayer ANNs was not available at the time. Such algorithm, known now as the backpropagation algorithm, was developed in 1974 by Paul Werbos, nevertheless, its potential was not immediately recognized by the researchers in ANNs. It was until 1986 when the backpropagation algorithm was finally used by Rumelhart et al., (1986) to develop ANNs to be applied in time series forecasting. ANNs have been gaining ground year by year since then (Zhang et al., 1998). A work by Werbos (1988) supported the use of ANNs through results from a series of instances in which backpropagation-trained ANNs performed better than several traditional statistical techniques such as linear regression and Box-Jenkins' method.

In recent years, ANNs have become very popular as modeling and analysis aids in areas as diverse as finances, medicine, energy generation, engineering, and environmental sciences among many others, and including time series forecasting (Maier et al., 2000). A large number of papers that make use ANNs for prediction can be found in the literature. For instance, some works use ANNs to predict the quality of air (Gardner and Dorling, 1999; Kolehmainen et al., 2001; Niska et al., 2004); another one (Kuo et al., 1999) compares the performance of ANNs and the autoregressive mean average in forecasting the demand in a chain of Chinese supermarkets. The results in this last one once again favored the ANNs.

Recent studies regarding the application of ANNs in Operations Research, including forecasting in finances, can be found in Zhang (2004) and Smith et al., (2000).

The fact that there is positive evidence in favor of ANNs in forecasting, should not give the false impression that their use is without challenges. In fact, forecasting precision highly depends on key decisions regarding several parameters as well as the architecture of the ANN (Zhang, 2004). Some of these decisions can be made while the ANN is being built, but others must be made a priori. Due to the lack of a standard method to make these decisions, in this work a method is proposed that allows setting the ANN's parameters to guarantee an adequate forecasting performance.

The proposed method is based on statistical analysis and optimization of the ANNs' performance measures (PMs). These PMs are defined as functions of forecasting errors for our purposes. The method consists on carrying out a statistical design of experiments where each factor involved corresponds to one ANN adjustable parameter. The experiment's results are then characterized through one regression model fitted to each PM. Each regression model is later used as the objective function in an optimization problem where the parameters are the decision variables. The solution to the optimization problem specifies the levels at which the ANN parameters should be set to generate reliable forecasts.

In this work, the decisions regarding the application of ANNs in time series forecasting are described first, followed by a detailed explanation of the proposed method. Finally, the use of the method is demonstrated through a real case study in a local telecommunications company. Emphasis in this manuscript has been given to describing the method and demonstrating its use in cases of one-period forecasts. Future publications will approach the more complicated case of multiple-period forecasting.

2 Decisions involved in the application of ANNs in time series forecasting

Development of an ANN for time series forecasting involves the critical task of specifying an architecture (or topology) for the ANN. This architecture must be defined in terms of the number of neurons in each of the three layers of the ANN proposed for the stated purpose i.e. the input layer, the hidden layer, and the output layer.

The number of neurons in the input layer determines the amount of historical data points or lags that will be used to generate the forecast. Because only one-period forecasts are considered here, the output layer will contain only the neuron corresponding to the single forecast. On the other hand, the number of neurons in the hidden layer determines the capability of the ANN to approximate the nonlinear relationships between the lags and the resulting forecasts (Zhang et al., 1998). Several authors have studied the choice of an adequate number of neurons in the hidden layer (Zhang, 2004; Hansen et al., 2003; Sexton et al., 2005), however, to the date there is not a definitive way to solve this problem.

Preprocessing the input data to improve an ANN's performance has been recommended by several authors in the literature. Among them, Piramuthu et al. (1998) documented the importance of preprocessing. It has been observed that training an ANN –the process by which the ANN learns input-output patterns- can actually become easier through different transformations of the input data. Preprocessing can be used to reduce the spread of the data in the search space

(Hansen et al., 2003) and is therefore an important factor to consider when trying to improve the performance of an ANN.

On the other hand, selection of a training algorithm is another important factor for most applications. Time series forecasting is not the exception in this case.

It is our premise that the decisions here described can be made in a simultaneous and systematic manner through the use of statistical design of experiments and optimization techniques as laid out in the method detailed in the ensuing section.

3 Proposed method

Figure 1 schematically shows the method proposed to set the parameters of the ANN in a simultaneous and systematic manner. The method is presented here as applicable to generic feedforward backpropagation ANNs. The details concerning time series forecasting are covered later in this manuscript.

The basic idea behind the method was the representation of the ANN as a system with controllable variables that exert influence over certain ANN's outputs. In this way, by considering the ANN parameters as controllable variables, we can explore their respective ranges and characterize their effect on key PMs (outputs) through regression techniques. This characterization, in turn, will allow the use of analysis of variance (ANOVA) to screen significant controllable variables as well as determine if two or more PMs depend on disjoint subsets of these variables. In this last case, independent optimization problems can be created thereby simplifying this task.

Because the ultimate goal is to set these parameters to obtain the best possible PM values, the problem indeed falls in the optimization realm.

The steps of the method outlined in Figure 1 can be described as follows:

Description of the ANN as a system. The objective in this step is to specify the characteristics of the ANN to be used. It is necessary to identify the ANN controllable parameters as well as to define the ANN's PMs to be included in the study.

Design and Analysis of Experiments. In this step, the focus is on planning, executing and interpreting the results of a statistically designed experiment that includes the previously identified parameters and PMs.

Metamodeling. A regression model must be obtained through proper statistical techniques (including residual analysis) to describe the response surface corresponding to each PM as a function of the controllable parameters.

Optimization Problem. An optimization problem is built with the metamodels obtained in the previous step as objective functions.

Solution. Solve the optimization problem through proper techniques i.e. multiple criteria optimization if necessary and multiple starting points.

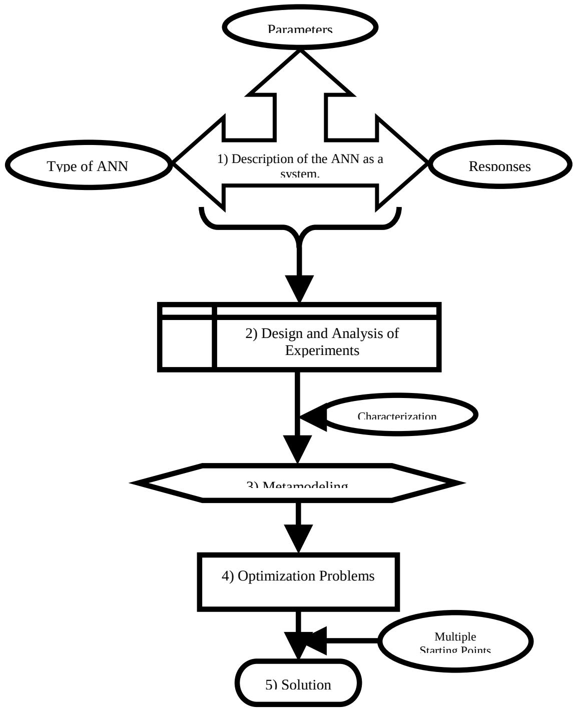

Fig.1. Proposed Method for setting an ANN's parameters

The rationale for the proposed method obeys the point of view of an experiment, where planned changes are introduced in a system with the objective to analyze the variation induced in PMs of interest. Because the size of the experiment to be run depends on the number of controllable parameters, the use of an adequate experimental design is critical. If there are a few parameters, then a full factorial design will suffice. A factorial design contains as many experimental runs as the complete enumeration of the combinations of values to be sampled per parameter (factor). For example, if four parameters are to be studied, each varied at three, four, three and five values or levels, then the factorial design will have 3×4×3×5=80 experimental runs.

For our purposes, then, under each combination of controllable parameters (run) an ANN will be built and trained to obtain a quantification of the prediction quality through the chosen PMs as the response of the experiment.

It is advisable to consider at least three different values for each parameter where possible to allow the determination of response curvature right from the beginning

Once the experiment is complete, it is followed by its analysis aiming to characterize the variation induced in the PMs. In order to achieve this, an ANOVA is run based on an underlying full quadratic regression model of each PM as a function of the controllable parameters. The regression model proposed is shown in Equation (1), where the residuals, ε, are assumed to be identically independently normally distributed with a mean of 0 and an unknown, but constant variance.

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^{\,2} + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \beta_{ij} x_i x_j + \varepsilon \qquad (1)$$

In Equation (1), dependent variable *y* represents the value of the PM of interest, x_i corresponds to the ith parameter, β_0 is the intercept of the model, β_i is the regression coefficient associated to the variation of x_i, β_{ii} is the regression coefficient for x_i^2, and β_{ij} is the regression coefficient for the interaction between x_i and x_j; and k is the total number of controllable parameters in the experiment.

The regression coefficients are typically computed through a squares reduction technique, available in most commercial statistical software packages. Checking the residuals' assumptions to verify the adequacy of the model is also conveniently carried out through these same packages.

Finally, each of the resulting regression models (one per PM) is included as an objective function in an optimization problem, where finding the parameter settings to achieve the best possible value for the objective function is sought.

At the end, the optimization problem will most likely be nonlinear and nonconvex. This makes the optimization problem hard to solve, and therefore the highest aspiration tends to be finding an attractive local optimal solution.

The proposed method was put in practice in a local telecommunications company for testing purposes. The case study that demonstrates the application of the method and the results is discussed next.

4 Case study

In this case study, a real problem in a local telecommunications company is approached. The company's ultimate objective, as the vast majority of companies in the world, is to be profitable through providing a high service level to its customers. Its major resource is a transmission network, loosely defined as a set of interconnected devices with a finite transmission capacity.

Customers demand different services at different levels through such network in a stochastic fashion and at some points in time challenging the installed network capacity. Because decisions on capacity expansion take time and equipment purchase and installation are not immediate, it is critical for the company to be able to estimate the demand behavior at least for the next period of time. Reliable forecasts should allow the decision-maker to plan the required capacity expansions to meet the expected demand without having too much unutilized installed capacity.

The company provided historic network utilization data organized in monthly periods i.e. a time series. Nonlinearity seemed evident from the beginning. This observation was further confirmed with the poor forecasts from a previously tried linear regression. However, as none of the previous was completely conclusive, we relied on the fact that ANNs have been shown to be competitive even in cases of linearity (Hwarng, 2001; Medeiros et al., 2001; Zhang, 2001). Based on these pieces of information, it was decided that an ANN would be a sensible choice. The parameter selection was approached through the method previously proposed in section 3 as described in the following section.

4.1. Description of the ANN as a system

Figure 2 shows the ANN selected for its use as a forecasting model. Such choice follows from this ANN's well-studied and demonstrated universal approximation capabilities (White, 1990; Hornik, 1989); as well as its documented good performance in forecasting (Zhang, et al., 1998; Zhang, 2004; Liao et al., 2005; Hansen, et al., 2004).

The one shown in figure 2 is a three-layer feedforward backpropagation ANN. Its first layer of neurons receives historic data, which is then sent to the intermediate layer (the hidden layer) for processing. The hidden layer determines the degree of flexibility in the forecasting model according to the number of neurons that it contains. Once processed by the hidden layer, the information is finally sent to the output layer, which in this case has a single output neuron (for the single-period forecast).

Referring to figure 2, ANN inputs $Y_i \quad i = t-m, t-m+1, \ldots t$ in the current forecasting application, corresponded to the *m* lags previous to period $t+1$, which is to be forecasted. Determining *m* is crucial because it indicates the extent in historic data to which a particular data point is potentially correlated. The number of *neurons* in the hidden layer allows modeling nonlinearity, hence its importance as a decision variable. A number of neurons larger than necessary would result in overtraining, which implies the loss of the ANN's prediction capability. If too few neurons are included, then the model will not be flexible enough to accommodate certain degree of nonlinearity. Finally, the neuron in the output layer computes the forecast for period $t+1$, represented by $\hat{Y}_{t+1}$. The transfer function for the hidden neurons was the hyperbolic tangent, and the identity function for the output neuron.

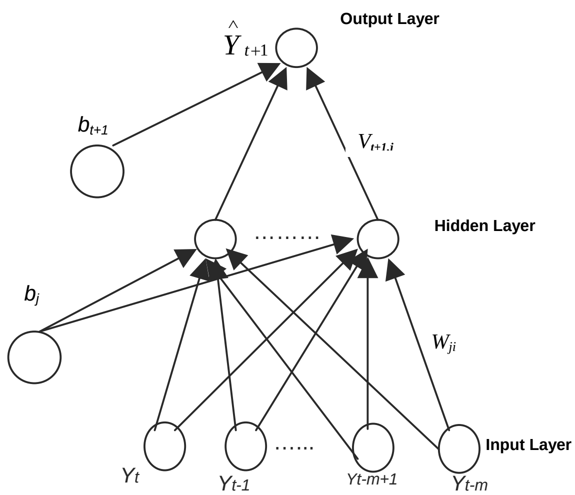

Fig. 2. Three-layer feedforward backpropagation ANN

Additionally in Figure 2, W_{ji}, $i = 1,2,\ldots, neurons$, $j = 1,\ldots, m$ is the weight applied in the incoming arc to hidden neuron j from input neuron i, $V_{t+1,i}$ $i = 1,2,\ldots, neurons$ is the weight applied in the incoming arc to neuron $t+1$ (output neuron) from neuron j (hidden neuron). These weights modify the information that passes through their respective arc and can be understood as fitting parameters. Special weights b_j and b_{t+1} are known as biases.

There is a large collection of algorithms in the literature to train an ANN, i.e. finding all ANN weight values. In this case, two algorithms were tried: Levenberg-Marquardt's (*lm*), which is valued for its convergence speed, and Bayesian Regularization (*br*), valued for its capability to avoid overtraining. Both of these algorithms have been shown to be quite competitive, although for different reasons as explained before (Bishop, 1995; Hagan et al., 1996).

Two parameters that are associated with the historic data for analysis were included: *transformation* and *scale; transformation* included two options: either leave the data as demand points on each period or used them as the difference between two adjacent periods; on the other hand, *scale* implied the option to either normalize the data to fall within the range [-1, 1], or use their original scale., The levels on both parameters have been suggested as analysis options in the ANN literature.

In order to measure the ANN forecasting error, the mean square error (MSE), the mean absolute error (MAE), the largest-in-magnitude positive error or over-prediction (*S_Pred*), and the largest-in-magnitude negative error or under-prediction (*B_Pred*) were used. Both *MSE* and *MAE* have been widely utilized in the literature, while *S_Pred* and *B_Pred* were important measures in this case to show the worst case of unsatisfied demand and the worst case of unutilized installed capacity in the network respectively. .

4.2. Design and Analysis of Experiments

The next step in the method was to carry out a statistically designed experiment. To this end, number of *lags* was varied within the range [2,6]; number of hidden *neurons* in the range [2,7]; *transformation* in {none, differences}; *scale* in {original, [-1,1] }, and *algorithm* in {lm,br}. Three levels were considered for *lags* and *neurons*, while two levels for the rest of the parameters as specified above. The PMs were *MSE, MAE, and B_Pred* y *S_Pred*.

Specific values for the ANN parameters are as follows: *lags* = {2,3,6}, *neurons* = {2,5,7}, *transformation* = {none, differences}, *scale* = {original , [-1,1] } and *algorithm* = {lm, br }. The labels "1", "2" and "3" were used to denote the levels of these parameters and correspond to the order of the listed values.

When conducting the experiment according to section 4.1, it was deemed adequate to use a factorial design $3^2 \times 2^3$ which resulted in a total of 72 combinations. Table 1 shows the experimental results when the training algorithm was Levenberg-Marquardt's.

Table 1 Experimental results with Levenberg-Marquardt's training algorithm

Neurons	Lags	Scale	Transformation	Algorithm	MAE	MSE	B_Pred	S_Pred
2	2	1	1	1	15.38	717.49	69.89	125.52
2	2	1	2	1	14.83	822.03	73.26	130.74
2	2	2	1	1	14.99	609.14	66.36	105.76
2	2	2	2	1	14.41	808.62	71.63	132.36
2	3	1	1	1	15.39	725.56	70.07	125.99
2	3	1	2	1	15.25	724.62	68.28	135.72
2	3	2	1	1	14.89	613.74	66.36	105.42
2	3	2	2	1	12.53	523.33	69.62	135.46
2	6	1	1	1	15.31	566.63	69.56	116.49
2	6	1	2	1	15.02	578.48	69.35	98.61
2	6	2	1	1	13.43	501.43	69.18	112.18
2	6	2	2	1	10.60	248.54	65.77	41.95
5	2	1	1	1	15.09	698.58	70.89	125.17
5	2	1	2	1	11.80	448.97	65.35	128.56
5	2	2	1	1	12.47	407.59	43.30	114.36
5	2	2	2	1	9.93	440.55	69.97	134.07
5	3	1	1	1	15.01	715.72	69.37	126.64
5	3	1	2	1	14.69	759.51	69.42	142.16
5	3	2	1	1	10.21	355.73	65.85	108.67
5	3	2	2	1	9.78	251.53	70.43	55.35
5	6	1	1	1	15.46	553.00	59.24	101.95
5	6	1	2	1	12.71	545.57	68.51	120.65
5	6	2	1	1	7.04	169.25	30.79	74.33
5	6	2	2	1	3.29	32.99	21.04	13.86
7	2	1	1	1	14.81	700.10	70.41	125.41
7	2	1	2	1	9.09	178.11	65.23	26.45
7	2	2	1	1	8.89	218.43	63.52	57.49
7	2	2	2	1	7.30	147.09	65.69	32.18
7	3	1	1	1	15.28	722.19	69.49	126.43
7	3	1	2	1	11.85	443.52	68.94	114.13
7	3	2	1	1	9.20	310.22	48.55	108.67
7	3	2	2	1	5.49	68.57	17.46	29.64
7	6	1	1	1	14.45	607.09	64.92	121.57
7	6	1	2	1	13.90	461.54	58.44	90.33
7	6	2	1	1	3.59	19.41	13.88	10.82
7	6	2	2	1	0.80	1.68	5.04	4.68

4.3. Metamodeling

The resulting experiment was then used to characterize the results through the use of ANOVA. This analysis, along with the factorial structure and the regression metamodel, allows the assessment of the statistical contribution of the parameters independently, as well as in second-order interactions. Second order regression models were obtained for each of the PMs. The resulting coefficients are shown in Table 2.

Table 2 Regression coefficients for each ANN performance measure

Regression Term	MAE	MSE	B_Pred	S_Pred
Constant	29.82	1726.89	102.61	219.74
x_1	-1.08	-84.40	0.13	4.36
x_2	0.55	-4.72	1.58	20.73
x_3	-1.92	-178.61	0.88	-23.03
x_4	-1.67	-100.62	9.10	-9.47
x_5	-14.15	-935.46	-61.57	-145.98
x_1^2	0.02	-0.15	0.07	-1.41
x_2^2	-0.08	-6.00	-0.13	-2.83
x_1x_2	0.01	7.41	-0.61	0.61
x_1x_3	-0.52	-25.08	-2.99	-4.35
x_1x_4	-0.21	-21.05	-1.08	-3.57
x_1x_5	1.00	68.57	3.94	9.57
x_2x_3	-0.74	-37.02	-3.41	-7.30
x_2x_4	0.06	4.64	-0.61	-2.19
x_2x_5	0.52	25.07	4.13	6.03
x_3x_4	-1.71	-101.01	-6.87	-7.87
x_3x_5	4.84	291.74	19.42	42.71
x_4x_5	2.91	237.47	7.45	27.82

Because using regression metamodels and ANOVA implies assumptions regarding the residuals, namely normality, independence and constant variance, it is important that these assumptions be verified through residual analysis to make sure that the subsequent conclusions are statistically valid.

Figure 3 shows an example on how residual analysis can be carried out graphically as a first approximation. This figure shows the analysis pertaining to *B_Pred*. The graphs on the left allow verifying if the residuals are following a normal distribution. In the graph in the upper left corner, a straight line pattern is sought and in the graph in the lower left corner a bell shape similar to the normal distribution should be approximately evident. The graphs in the right have the purpose of checking for independence of the residuals vs. the model predictions (fits) and vs. the run order. The behavior of the residuals in these graphs should be random. These two graphs also help to see if there was a dramatic change in the spread of the residuals (nonconstant variance). A series of formal statistical tests usually follows the graphical analysis. In this case, after statistical testing, the conclusion was that there was no significant violation to the residual assumptions. Further details on these tests can be found in Devore (1995). A similar analysis was carried out for each PM, with similar conclusions.

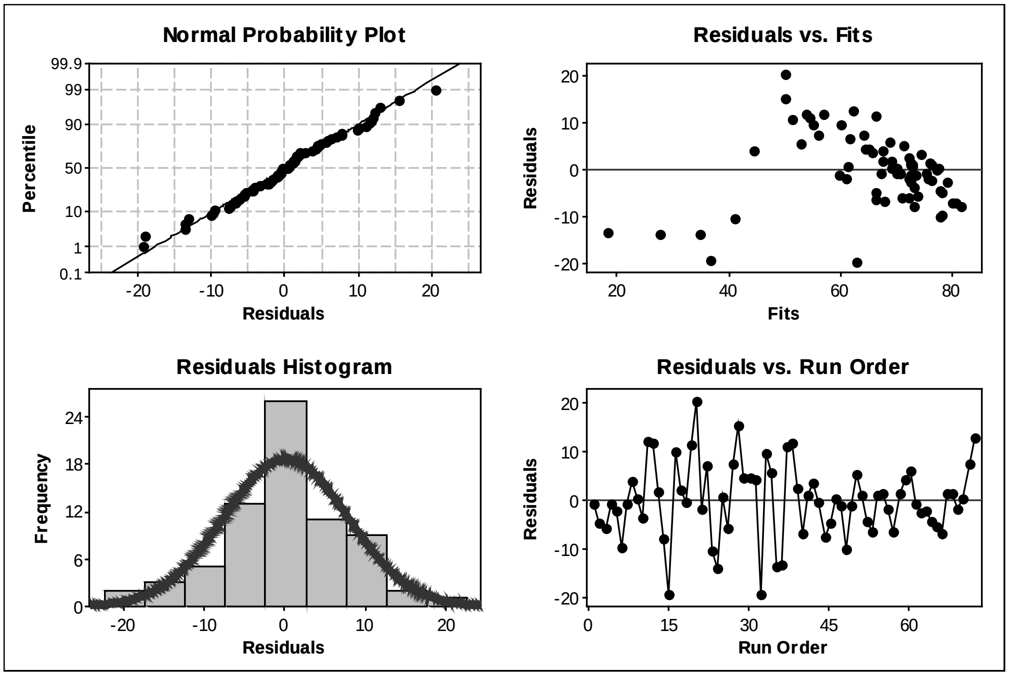

Fig. 3. Residual Analysis for the regression model corresponding to *B_Pred*

According to the proposed method, metamodeling must be followed by the creation of optimization problems. This is described in the following section.

4.4. Optimization problems and solution

On each of the optimization problems, the objective function z is defined by the metamodel representing the PM of interest. In general, the resulting optimization problems will have the following format:

Find $\quad x_i \quad i \in I = \{1, 2, ..., k\}$ to

Minimize

$$z = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^{\,2} + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \beta_{ij} x_i x_j$$

Subject to

$$x_{li} \le x_i \le x_{ui} \qquad \forall\, i \in I$$

$$x_i \in Z^+ \qquad \forall\, i \in I$$

where the decision variables x_i, represent the ith ANN controllable parameter through the variation of which the objective function z is to be minimized. Scalar k stands for the number of controllable parameters whose values must fall within a preset lower bound x_{li} and a preset upper bound x_{ui}. These bounds usually correspond to the limits of the region explored in the experimental design. Finally, the decision variables must be positive integers as indicated by the last constraint.

In our case study $k = 5$, corresponds to the parameters: *neurons, lags, scale, transformation* and *algorithm.*

Solving each optimization problem independently and using multiple starting points to increase the probability of finding an attractive local optimum, it was found that the input data should be handled in differences and scaled to fall within the range [-1,1]. The number of *lags* (now in differences) was determined to be 6 periods, and the ANN should have 7 hidden neurons. The training algorithm was prescribed to be Levenberg-Marquardt's. An ANN with these resulting characteristics provides forecasts with performance as shown in Figure 4. It can be observed that the model is indeed a good approximation to the time series. In fact, this configuration coincided with the best experimental combination in Table 1. An adequate forecasting performance is highly probable after the application of the proposed method.

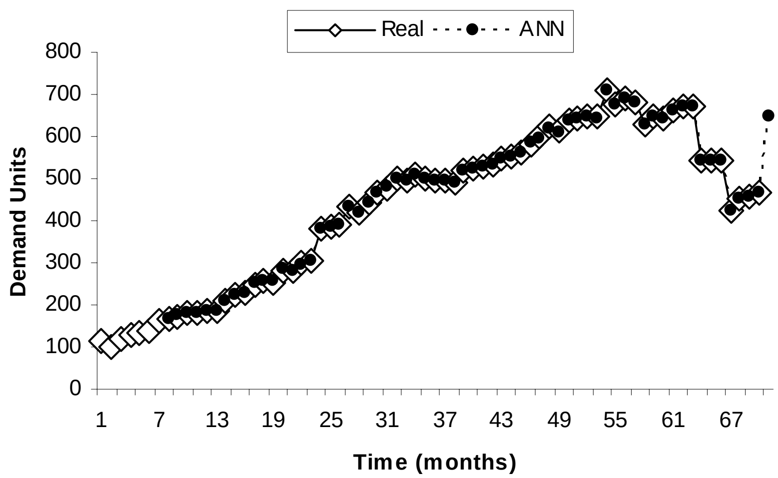

Fig. 4. ANN predictions vs. the real time series

It is necessary to point out, however, that although we discussed the issue of overtraining previously, in this particular case it was not explicitly considered. The reason had to do with keeping the objective at hand simple: testing the methodology. A future publication will include our complete results for multiple-period forecasting where cross-validation was used to avoid overtraining.

The final ANN, whose performance is depicted in Figure 4 was the best one found by the optimization procedure when minimizing each one of the PMs. This result is a clear indication of a large degree of correlation without conflict among the PMs. Had the optimization problems arrived to solutions different to each other, selecting a combination would have not been a trivial task. In such case it would have been necessary to use multiple criteria optimization techniques. These techniques try to find the best compromises between all PMs, which are formally called Pareto-efficient solutions. Because treatment of multiple criteria optimization procedures is beyond the scope of this work, the interested reader is referred to Deb (2004) and Hellermeier (2001). Also, some recent applications of multiple criteria optimization in manufacturing using Data Envelopment Analysis can be found in Cabrera-Ríos et al. (2002,2004), Castro J. M. et al. (2004) and in Castro C. E et al. (2003).

In this case study, the computational tools were Matlab™, Minitab™ and MS Excel™. The first one was used to build the ANNs, the second one was used for statistical analysis, and the third one for optimization purposes through its XL Solver module. The complete results of this work are available upon request to the authors.

5 Conclusions and future work

In this work, a method to set the parameters for an ANN in the context of time series forecasting was proposed. The method involves the coordinated use of design of experiments, analysis of variance, linear regression and optimization. A case study where the method was implemented and tested in a local telecommunications company was presented to demonstrate the use and performance of the method. Among the most attractive features of the proposed method we can list the following: (1) it uses well-established and reliable analytical techniques, (2) its implementation does not require highly specialized code, and (3) it makes the interrelationships between several key ANN parameters transparent.

The proposed method can be applied to ANNs with more complicated architectures, such as those used to forecast *n* future periods. A follow-up publication considering this case is under preparation in our research group.

References

1. Bishop C. M., Neural Networks for Pattern Recognition, Oxford University Press, Oxford, UK, 1995.
2. Box G. E. P., and Jenkins G. M., Time Series Analysis: Forecasting and Control, San Francisco, CA: Holden-Day, EUA, 1976.
3. Cabrera-Ríos M., Castro J. M., and Mount-Campbell C. A., “Multiple quality criteria optimization in reactive in-mold coating with a data envelopment analysis approach II: a case with more than three performance measures”, *Journal of Polymer Engineering*, Vol. 24, No. 4 , 2004, 435-450.
4. Cabrera-Ríos M., Castro J. M., and Mount-Campbell C. A., “Multiple quality criteria optimization in in-mold coating (IMC) with a data envelopment analysis approach”, *Journal of Polymer Engineering*, Vol. 22, No. 5, 2002, 305- 340.
5. Castro C. E., Cabrera-Ríos M., Lilly B., Castro J. M., and Mount-Campbell C. A., “Identifying the best compromise between multiple performance measures in injection holding (IM) using data envelopment analysis (DEA)”, *Journal of Integrated Design and Process Science*, Vol. 7, No. 1, 2003, 77-87.
6. Castro J. M., Cabrera-Ríos M., and Mount-Campbell C. A., “Modelling and Simulation in reactive polymer processing”, *Modelling and Simulation in Materials Science and Engineering*, Vol. 12, No. 3, 2004, S121-S149.
7. Deb K., Multi-Objective Optimization Using Evolutionary Algorithms, Editorial Wiley, NY, EUA, 2004.
8. Devore J. L., Probability and Statistics for Engineering the Sciences, 4ta Edition, Duxbury Press, California Polytechnic State University, EUA, 1995.
9. Gardner M. W., and Dorling S. R., “Artificial neural networks (the multi-layer perceptron) - a review of applications in the atmospheric sciences”, *Atmospheric Environment*, Vol. 33, 1999, 709-719.
10. Hagan M. T., Demuth H. B., and Beale M., Neural Network Design, PWS Publishing Company, EUA, 1996.
11. Hansen J. V., and Nelson R. D., “Forecasting and recombining time-series components by using neural networks”, Journal of the Operations Research Society, No. 54, 2003, 307-317.
12. Hillermeier C., Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach, Basel: Birkhauser Verlag, 2001.
13. Hornik K., Stinchcombe M., and White H., “Multilayer feedforward networks are universal approximators”, *Neural Networks*, Vol. 2, No. 5, 1989, 359-366.
14. Hu M. J. C., “Application in the Adaline system to weather forecasting”, Master Thesis, Technical Report 6775-1, Stanford electronic Laboratories, Stanford, CA, June 1964.
15. Hwarng H. B., “Insights into neural-network forecasting of time series corresponding to ARMA (p,q) structures”, *Omega: The International Journal of Management Science*, Vol. 29, No. 3, 2001, 273-289.

16. Kolarik T., and G. Rudorfer, “Time series forecasting using neural networks”, *Conference Proceedings of the International Conference on APL 1994. APL Quote Quad*, Vol. 25, No. 1, 1994, 86-94.
17. Kolehmainen M., Martikainen H., Russkanen J., “Neural networks and periodic components used in air quality forecasting“, *Atmospheric Environment*, Vol. 35, 2001, 815-825.
18. Liao K-P, and Fildes R., “The accuracy of a procedural approach to specifying feedforward neural networks for forecasting”, *Computers & Operations Research*, Vol. 32, No. 2, 2005, 151-2169.
19. Maier H. R., and Dandy G. C., “Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications”, *Environment Modelling & Software*, Vol. 15, 2000, 101-124.
20. Makridakis S., Anderson A., Carbone R., Fildes R., Hibbon M., Lewandowski R., Newton J., Parsen E., and Winkley R., “The accuracy of extrapolation (time series) methods: Results of a forecasting competition”, *Journal of Forecasting*, Vol. 1, 1982, 111-153.
21. Makridakis S., and Wheelwright S. C., The Handbook of Forecasting a Manager’s Guide, 2da Edition, Editorial Wiley, NY, EUA, 1987.
22. Medeiros M. C., and Pedreira C. E., “What are the effects of forecasting linear time series with neural networks?” *Logistic and Transportation Review*, Vol. 31, No. 3, 2001, 239-251.
23. Niska Harri, Hiltunen Teri, Karppinen Ari, Russkanen J. and Kolehmainen M., “Evolving the neural network model for forecasting air pollution time series”, *Engineering Applications of Artificial Intelligence*, Vol. 17, 2004, 159-167.
24. Piramuthu S., H. Ragavan, and M. Shaw, “Using feature construction to improve the performance of neural networks”, *Management Science*, Vol. 44, No. 3, 1998, 416-430.
25. R. J. Kuo and k. C. Xue, “Fuzzy neural networks with application to sales forecasting”, *Fuzzy sets and y systems*, Vol. 108, No. 2, 1999, 123-143.
26. Rumelhart D-E., Hinton G. E., and Willians R. J., “Learning representations by backpropagating errors”, Nature, 323 (6188), 1986, 533-536.
27. Sexton R. S., McMurtrey S., Michalopoulos J. O., and Smith A. M.,“Employee turnover : a neural network solution”, *Computers & Operations Research*, Vol. 32, No. 10, 2005, 2635-2651.
28. Smith K. A. and Gupta JND, “Neural networks in business: techniques and applications for the operations researcher”, *Computers and Operations Research*, Vol. 27, Num. 11-12, 2000, 1023-1044.
29. Werbos P. J., “Generalization of backpropagation with applications to a recurrent gas market model”, *Neural Networks*, Vol. 1, 1988, 339-356.
30. White H., “Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings”, *Neural Networks*, Vol. 3, No. 5, 1990, 535-549.
31. Widrow B., Rumelhart D., and Lehr M. A., “Neural networks: Applications in industry, business and science”, *Communications of the ACM*, Vol. 37, No. 3, 1994, 93-105.
32. Zhang G., Patuwo E., and Hu Y. M., “Forecasting with artificial neural networks the state of the art”, *International Journal of Forecasting*, Vol.14, No. 1, 1998, 35-62.
33. Zhang G. P., Neural Networks in Business Forecasting, Idea Group Publishing, Georgia State University, EUA, 2004.
34. Zhang G. P., and Hu M. Y., “A simulation study of artificial neural networks for nonlinear time series forecasting”, *Computers & Operations Research*, Vol. 28, 2001, 381-396.

Acknowledgements

The authors are grateful to the CONACYT for the scholarship granted to Ms. Salazar for her graduate studies.

María Angélica Salazar Aguilar *obtained her B.S. in Computer Systems Engineering from Instituto Tecnológico de Querétaro (ITQ) (2004), and her M.S. in Systems Engineering from the Graduate Program in Systems Engineering at Universidad Autónoma de Nuevo León (2005). Her interests include Artificial Intelligence and Operations Research. She currently holds an appointment as an Assistant Professor at ITQ in the Department of Systems and Computing.*

Guillermo J. Moreno Rodríguez *obtained his B.S. in Electrical and Communications Engineering from ITESM-Monterrey (1994). His interests include all aspects of telecommunications network management. He is currently the manager of the division of Traffic Engineering and Capacity Optimization of Avantel, México.*

Mauricio Cabrera Ríos *obtained his B.S. in Industrial and Systems Engineering from ITESM-Monterrey (1996), and his M.S. and Ph.D. in the same discipline from The Ohio State University (1999, 2002). His research interests are in general related to the application of optimization to manufacturing and services. Currently, he holds an appointment as an Associate Professor at the Graduate Program in Systems Engineering, FIME, Universidad Autónoma de Nuevo León. He is a member of the National System of Researchers, Level 1. http: //yalma.fime.uanl.mx/~mauricio/*