Short-term load forecasting by artificial neural networks specified by genetic algorithms – a simulation study over a Brazilian dataset.

Samuel Belini Defilippo[1], Guilherme Guilhermino Neto[2], Henrique S. Hippert[3]

[1] PPG Modelagem Computacional, UFJF. samueldefilippo@outlook.com
[2] IFES – Campus Linhares. guilherme-gn@outlook.com
[3] Depto. Estatística, UFJF. henrique.hippert@ufjf.edu.br

Abstract. This paper studies the application of genetic algorithms in helping to select the proper architecture and training parameters, by means of evolutionary simulations done on a series of real load data, for a neural network to be used in electric load forecasting. Particularly, we investigate the application of a novel fitness function to the genetic algorithms, instead of the usual ones, based on the sum of the squares of the errors. We compare the results of the neural networks thus specified with that of four benchmarks: two naive forecasters, a linear method, and a neural network in which the parameter values are found by means of a grid search.

Keywords: short-term load forecasting; artificial neural networks; genetic algorithms

1 Introduction

Electricity is the form of energy most used throughout the world. It generates heat, light and power, and is directly linked to the technological and economic development of a country. The electricity generating system may be compared to a productive system in which plants, turbines and generators combined with fuel or water (inputs) produce electricity (output), which is then distributed to the clients. For proper and efficient operation of any production system, planning and production control is required; this requires forecasts for the long term, the middle term and the short term. For the electricity generating systems, long-term forecasts (for several years ahead) are needed for planning the expansion or reduction of production capacity - by the installation of new plants, for example. Medium term forecasts (for a few weeks or months ahead) are necessary to support decisions on the utilization of the existing system - the purchase of fuel, or scheduling of maintenance activities. Short-term forecasts (one hour to a few days ahead), are necessary to optimize the use of the plant and the machinery.

The production of electricity, however, differs in some ways from other forms of industrial production. The differences are derived mostly from the fact that energy

adfa, p. 1, 2011.
© Springer-Verlag Berlin Heidelberg 2011

cannot be stored in large quantities; thus the production has to meet the demand very accurately at all times. Energy produced in excess may be wasted, which means useless consumption of fuel or water, and losses for the company. On the other hand, if the production does not reach the demand, the system may fail, and this may lead blackouts. Thus, accurate short-time forecasts of the demand are essential for the operation of a system.

The privatization of utilities and the deregulation of energy systems, which started in several countries by the late 1980s, have also increase the importance of forecasting and raised the cost of the forecasting errors. Highly competitive energy markets have sprung up, focusing on energy production with high quality standards, at low costs. Particularly, one day ahead forecasts became extremely important, because besides being needed for the day-to-day operation of the system (scheduling the times for turning the generating units on and off, so as to minimize production costs), they are also one of the inputs needed for the definition of the price of electricity in the market [1]. Data on energy demand are usually obtained in the form of time series of electric loads. The forecasting of future values of these series has been tried by several different methods. Some of these are based on univariate (time-series) models, in which the forecast is a function of the past load only. Others are based on multivariate techniques, in which the forecast is a function not only of the past loads, but also of exogenous variables related to the weather or to social events. Many artificial intelligence techniques have been tried for this task, because of their flexibility and their ability to model complex nonlinear multivariate relationships. Among these, the most frequently researched have been the Artificial Neural Networks [2].

This paper describes simulations done with a method that combines artificial neural networks and genetic algorithm, and we propose a novel cost function to be minimized. Usually, the models minimize the mean square error (MSE) of the forecasts. Since, however, the forecasts are made in the same instant of time for different forecasting horizons (in this paper, for example, we consider that the forecasts for all the 24 hourly loads in the next day are done at today's midnight), the forecasting errors tend to be highly autocorrelated. This is undesirable, since the existence of autocorrelation in a series of errors means that the model is not extracting all the information available in the data. In this paper, we propose a cost function which is a variation of the well know Ljung-Box statistic used to test the autocorrelation of the errors generated by the statistical times series models, such as the ARIMA models.

2 Material and methods

2.1 Load data from Rio de Janeiro

For the simulations, we used a series of load (in MWh) measured in the city of Rio de Janeiro, by a local power company. This database contains the hourly loads for days 01/Jan/1996 to 30/Dec/1997, adding up to 17,472 observations. Loads on special days (such as holidays) are usually forecasted offline by the companies, by means of proprietary methods that tend to be largely empirical; since these loads are not the focus of this paper, we opted to smooth them out of the series, replacing them by the average of previous observations on the day of the week and at the same time.

Load series usually have complex structures, including several superimposed seasonal patterns. The lineplot in Figure 1 highlights a weekly cycle: the demand is higher from Monday to Friday and lower on weekends.

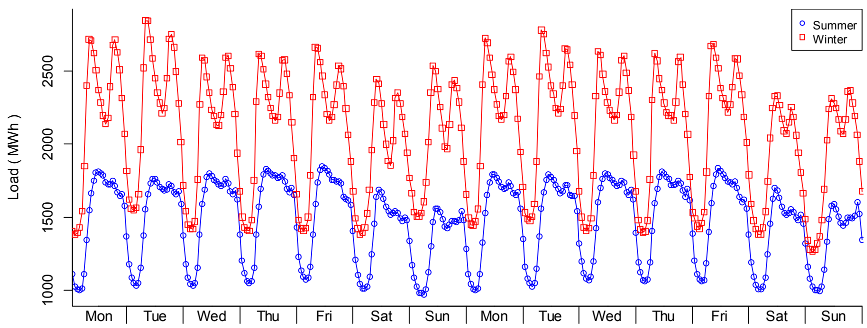

Fig. 1. Typical demand curves for summer and winter (Rio de Janeiro, Brazil)

Also, daily cycles are clearly seen: in winter, all weekdays tend to exhibit the same profile (series of 24 hourly demands), with a peak at 19h; in summer, the profile is different, with a peak at 15h and another at 23h. Also, there is a yearly cycle, linked to the seasons, that can be noticed both in Figure 1 and Figure 2; the increase in load during summer is due to the intense use of air-conditioning.

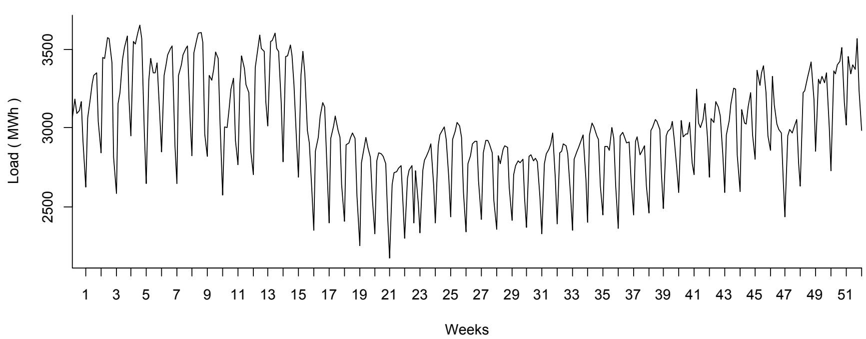

Fig. 2. Weekly average loads in one year (Rio de Janeiro, Brazil)

This load series was partitioned into three subseries. Data ranging from week 15 to 64 were used for training the ANNs (the *training* sample). Data from week 65 to 84 were used by the genetic algorithms to check the fitness of the competing models (the *validation* sample). Finally, data from weeks 85 to 104 were used to test the models selected by GA, against four different benchmarks (the *test* sample).

2.2 Statistical model load curve

In this paper we use a standard "load curve model" for forecasting. These models are frequently used in the electricity industry, mainly due to their simplicity of implementation and the relative ease of interpretation of their results [3]. The idea is to model the expected profile of a day assuming a priori a functional mathematical form for its shape. After a model is estimated, it is fed at the end of each day with new data, and then used to forecast the profile for the next 24 hours.

In the most common univariate additive model, the demand at time h on day d, L(d,h), is given by the sum of a base component B(d,h), which is a function of the most recent previous observations, with a random error R(d,h), as in eq. (1):

$$L_{(d,h)} = B_{(d,h)} + R_{(d,h)} \quad (1)$$

As the statistical model above is, in principle, a single univariate model for all hours of the day, it is necessary to adopt some method to consider the multiple layers of seasonality of the data, and thus obtain better predictions. Some authors, such as [4] and [5], recommend splitting the data into 24 separate time series, and then adjusting one model for each hour of the day. Others, such as [6] and [7] advise splitting the data into 168 separate series, one for each of the 168 hours of a week. A third option figures in the recent literature: keeping a single model, but choosing one that can model simultaneously the triple seasonality present in load series. A more recent such development was the adaptation of the well-known Holt-Winters exponential smoothing method [8] and [9]; this method was used on the same data series as in the present paper, and its results are used as a benchmark.

Instead of using a univariate model, another option is using a multivariate one, capable of dealing simultaneously with several inputs and 24 outputs (the forecasts). This has been frequently done in recent years by means of artificial neural networks; we discuss them below.

2.3 Artificial neural networks (ANNs)

The ANN models we use are Multi-Layer Perceptrons, with a single hidden layer. They are very adaptable, the can incorporate the several levels of seasonality in a single model.

The ANNs we used had only one hidden layer, with sigmoidal activation functions. To estimate the neural network weights, we used a training function that incorporated regularization [10]. We chose this function because it led to an improved accuracy, as compared to a few others which were also tried out. The neural network has 24 output nodes, representing the load profile of the next day. The choice of the input variables, the number of hidden layers and the number of training epochs were done by a genetic algorithm (section 2.4).

For forecasting next day's profile, the available input variables were the temperature differences (the differences between the temperature at each hour in a day, and the temperatures at the same hours on the previous day), the measured hourly loads, and seven dummies to represent the days of the week. So, we experimented with a total of 103 possible input variables, as follows:

24 differences between the hourly temperatures measured today, and those measured yesterday

24 differences between the hourly temperatures measured today, and those forecasted for tomorrow

48 hourly loads measured today and yesterday.

06 bits, representing the days of the week.

2.4 Genetic Algorithms (GAs)

One of the difficulties in the implementation of ANNs is the lack of straight-forward criteria to help the researcher in defining the parameters of the model or of the training process. Just as there is still no generally accepted methodology for the selection of input variables, there are also no rules for selecting the number of layers and of hidden neurons, etc. For the definition of these parameters, GAs have been tried by several authors [11], and we used them in this paper.

The GA we used was of a binary generational type. The stopping criterion was the stabilization of the best genes, for 200 generations, and the existence of 50 individuals in the population without crossover operator, and with mutation operator with probability 0.2. The selection was made through one tournament, and "elitism" was implemented (i.e., 49 individuals are passed on to the next generation; only the worst individual is discarded).

Each individual (i.e., each ANN model) was represented by a vector of bits. The first 103 bits were used to represent the candidate input variables; the variables eventually included are listed in Table 1. The next five bits are used to represent the number of hidden neurons (01 to 32); the final bits represent the number of epochs (01 to 32) for ANN training; since we used a training function based on regularization, we did not use cross-validation for control early stopping.

A GA minimizes a chosen cost function or criterion. Usually, this is the mean square error (MSE) of the fitted model; in this paper, however, we experimented with the function defined below.

2.5 The cost function

As explained in the introduction, profile forecasting methods frequently result in forecasting errors that exhibit marked autocorrelation, instead of being white noise, as would be ideally expected. In order to reduce this autocorrelation, we experimented using a GA for minimizing a cost function Q defined by:

$$e_t = z_t - \hat{z}_t \tag{2}$$

$$Q = \sum_{i=1}^{24} (e_t \times e_{t-i})^2 \quad \forall t \tag{3}$$

In this equation, each product e(t) x e(t-i) may be considered as an approximation to the autocovariance of the series, for lag = i. This sum of the squares of autocovariances Q is analogous to the statistics used in the "port-manteau" tests (commonly employed in ARIMA modeling), which are based on the sum of the squares of the autocorrelations of the errors for all lags in a given interval - if the error is a white noise, all the autocorrelations are equal to zero, and the sum tends to zero. By minimizing Q, therefore, we expect to minimize the autocovariances present in the error series. We used this cost function in two ways.

First, we run the GAs three times, searching to minimize the cost function as defined in eqs. (2) and (3), with et representing the forecasting error at each hour. The models that resulted of these three runs are listed below as GA01, GA02, GA03. Second, we did another three runs of the GAs, this time using et to represent the total daily forecasting error (the sum of the hourly errors); the models that resulted are listed below as G04, GA05 and GA06.

2.6 Other details

Benchmarks

The different models we experimented with were compared among themselves, and also against four benchmarks, in terms of their forecasting accuracy. The benchmarks were two naïve forecasters (Naive1, Naive2), a modification of the well-known Holt-Winters exponential smoothing method (HWT), and an artificial neural network (ANN). These benchmarks are detailed below.

Naive1: this method forecasts the load at the hour h of day d, by using the load at the same hour on the previous day; that is:

$$\hat{z}_{d,h} = z_{d-1,h}$$

This is a naïve method takes into account the daily seasonality (the daily load profiles tend to repeat themselves from day to day). However, this daily seasonality is broken at the weekends: since this method predicts Monday loads with basis on Sunday loads, its accuracy is expected to be poor.

Naive2: this method forecasts the load at the hour h of day d, by using the load at the same hour on day with the same denomination on the week before; that is:

$$\hat{z}_{d,h} = z_{d-7,h}$$

This method takes into account the weekly seasonality (Monday loads, for example, are predicted with basis on the previous Monday loads), but it does not take into account the variation in average level caused by the yearly seasonality.

HWT: Taylor [8] and [9] adapted the Holt-Winters exponential smoothing method, adding two new equations, to allow it to deal with series which show three levels of seasonality. We experimented with this method on the same load series on a previous study [12], and the results are here used as benchmarks.

ANN2001: In a previous study [13], we used a large neural network to forecast the same load series as in this paper. This ANN was a MLP, and its architecture and training parameters were chosen with basis of empirical validations, instead of by a genetic algorithm. These early results are here used as benchmarks.

Error measure

The forecasting accuracy of the different methods were compared by means of the the mean absolute percent error (MAPE) defined as:

$$MAPE = 100 \times \frac{1}{n} \sum_{t=1}^{n} \left| \frac{z_t - \hat{z}_t}{z_t} \right| \tag{4}$$

We chose this error measure because it has proved to be the most commonly used in the load forecasting literature, due to its easy interpretation.

3 Results

Table 1 shows the details of the three ANNs selected in the three GA runs we did using the cost function in (3) as the fitness function (GA01 to GA03). Table 1 also shows the details of the ANNs selected by the GA runs using the second cost function defined in Sect. 2.6.2, the one based on the daily sum of errors, instead of on the hourly errors (GA04 to GA06).

Method	# neurons	Training epochs	Temperature differences: today-yesterday	Temperature differences: yesterday-day before	Loads on last two days	Dummies
GA01	29	11	11	17	19	4
GA02	18	15	10	14	23	6
GA03	29	13	10	13	29	5
GA04	31	13	14	23	31	5
GA05	17	10	11	15	21	6
GA06	18	11	10	17	23	5

Table 1. ANNs selected by each run of a GA

We run every one of these six models 30 times each on the training samples, to train their weights, and then simulated their forecasts over the test sample, and computed their MAPEs. These are described summarily on Table 2: the minimum MAPE obtained on the 30 runs (i.e., the best forecasts), the maximum MAPE (the worst forecasts), and the average MAPE.

ANNs	Maximum MAPE	Minimum MAPE	Average MAPE	Median MAPE
GA01	2.49	2.01	2.24	2.27
GA02	2.51	2.07	2.23	2.22
GA03	2.81	2.15	2.34	2.32
GA04	2.26	1.99	2.10	2.09
GA05	2.82	2.23	2.39	2.38
GA06	2.69	2.14	2.25	2.23
ANN2001	2.75	2.26	2.44	-
Naive1	3.35			
Naive2	6.12			
HWT	2.47			

Table 2. – Results: MAPEs on weeks 85-104

These same results are shown by means of boxplots on Figure 3. The MAPE produced by HWT, and the average MAPE produced by ANN2001, are shown by the blue and red vertical lines across the boxplots, respectively.

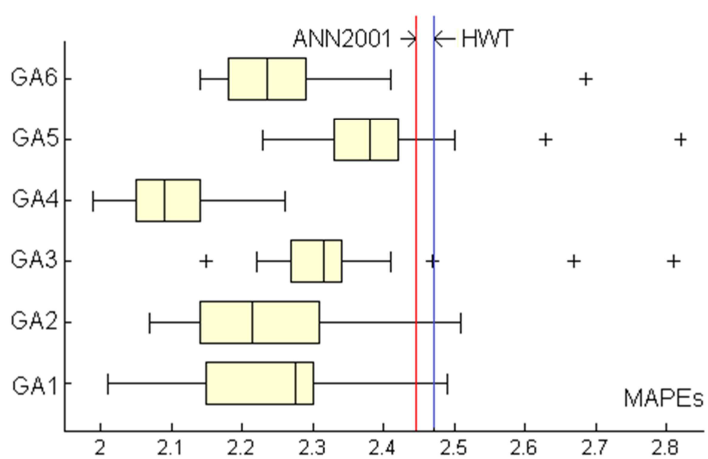

Fig. 3. MAPEs on weeks 85-104

4 Discussion and conclusion

As can be seen in Figure 3 and in Table 2, the GA-based method we used produced in general much better results than the four benchmarks. The MAPEs in all GA runs were much lower than the ones of the two naive forecasters (Naive1, Naive2). As compared to the other two benchmarks, we notice, first, that the mean and the median MAPEs we obtained in the six GA runs were always lower than the MAPE of the HWT, or the mean and median MAPEs of the ANN2001. Actually, in half of the GAs we tried (three out of six), the median MAPEs we obtained were even better than the *best* result obtained by the ANN2001. The median MAPE in all 6 GA runs was 2.24% which was about 0.20% below the ANN2001 median.

However, one of the difficulties in applying ANNs is the wide variation of the results. Since there are a great many design parameters to be specified, and the network weights have to be found by optimization processes, the results of the GAs are always unpredictable. Models with very different structures may be found at the end of each run. Usually, these models result in more or less the same performance in forecasting (since ANNs are such large models, a single 'perfect' model is never found); it occasionally happens, however, that some models are found that perform very poorly when applied to test data; such are for instance the ones whose MAPEs are shown as outliers in the boxplots in Figure 3.

Because of that, one should always consider, when analyzing the results of ANN forecasters, not only the mean or median error measures, but also the worst cases – the very large prediction errors, the ones that, had they occurred in real life, would have resulted in serious losses for the electricity companies. In this respect, we find that the models selected by the GA runs performed comparatively well. Out of 180 runs (30 of each GA), only six values were considered as outliers (see Figure 3); but even these were well below the limits set by the Naive1 and Naive2 forecasters.

We think, therefore, that the results of the GA-based method we tried were very promising. For future research, we believe it would be interesting to experiment with the use of ANN committees, instead of individual ANNs, since this might lead to a further reduction in the variation of the MAPEs.

5 References

1. Bunn, D. W. Forecasting loads and prices in competitive power markets. Proc. IEEE, 2, 163-169. (2000)
2. Hippert, H.S.; Pedreira, C.E.; Souza, R.C. Neural networks for short-term load forecasting: A review and evaluation. IEEE Trans Power Syst, 1, 44-55. (2001)
3. Bunn, D.W.; Farmer, E.D. Comparative Models for Electrical Load Forecasting. 1. ed. New York: Wiley (1985).
4. Ramanathan, R.; Engle, R.; Granger, C.W.J.; Vahid-Araghi, F.; Brace, C. Short-run forecasts of electricity loads and peaks. Int J Forec, 14, 161-174. (1997)
5. Harvey, A.; Koopman, S.J. (1993). Forecasting hourly electricity demand using time-varying splines. J American Stat Assoc, 424, 1228-1236. (1993)

6. Papalexopoulos, A.D.; Hesterberg, T.C. A regression-based approach for short-term load forecasting. IEEE Trans Power Syst, 4, 1535-1550. (1990)
7. Infield, D.G; Hill, D.C. Optimal smoothing for trend removal in short-term electricity demand forecasting. IEEE Trans Power Syst, 3, 1115-1120. (1998)
8. Taylor, J.W. Exponentially weighted methods for forecasting intraday time series with multiple seasonal cycles. Int J Forecasting, **26**, 627-646 (2010)
9. Taylor, J.W. Triple seasonal methods for short-term load forecasting. European J Oper Research, 204, 139-152. (2010)
10. Bishop, C. M. Neural networks for pattern recognition. Oxford: Oxford Univ Press (1995).
11. Dréo, J.; Pétrowski, A.; Siarry, P.; Taillard, E. Metaheuristics for Hard Optimization. Springer. (2005)
12. Carneiro, A. C. M. da S.; Guilhermino Neto, G.; DeFilippo, S. B.; Hippert, H. S. Previsão de carga a curto prazo por combinações de métodos lineares. Anais do XXI SIMPEP – Simpósio de Eng de Produção, UNESP – Bauru. (2014)
13. Hippert, H. S. Previsão de cargas a curto prazo – uma avaliação da viabilidade do uso de redes neurais artificiais. Tese de doutorado, Depto. Engenharia Elétrica, PUC-Rio. (2001)