

Time Series Forecasting by Evolving Artificial Neural Networks Using “*Shuffle*”, Cross-Validation and Ensembles



Juan Peralta, German Gutierrez, and Araceli Sanchis

Computer Science Department, University Carlos III of Madrid
Avenida de la Universidad 30 28911 Leganes, Spain
{jperalta,ggutierr,masm}@inf.uc3m.es

Abstract. Accurate time series forecasting are important for several business, research, and application of engineering systems. Evolutionary Neural Networks are particularly appealing because of their ability to design, in an automatic way, a model (an Artificial Neural Network) for an unspecified nonlinear relationship for time series values. This paper evaluates two methods to obtain the pattern sets that will be used by the artificial neural network in the evolutionary process, one called “shuffle” and another one carried out with cross-validation and ensembles. A study using these two methods will be shown with the aim to evaluate the effect of both methods in the accurateness of the final forecasting.

Keywords: Evolutionary Computation, Genetic Algorithms, Artificial Neural Networks, Time Series, Forecasting, Ensembles.

1 Introduction

Time series forecasting is an essential research field due to its applications in several research, commercial and industry areas, and can be performed by Statistical methods or Artificial Neural Networks (ANN) [1]. The ANN have the capability, without any information but the data, of extracting the nonlinear relationship between the inputs and outputs of a process. There are, in the literature some “state of art” by Abraham [2] and Yao [3] about automatic methods to design ANN based on Evolutionary Computation (EC).

In order to deal with model uncton $x_t = f(x_{t-1}, x_{t-2}, \dots, x_{t-k})$, time series known values will be transformed into a patterns set, depending on the k inputs nodes of a particular ANN. If the number of input nodes are different their pattern set are different and will be used to train and validate each ANN generated in the GA. The fitness value for each individual will be then the minimum validation error along the training of ANN topology. Once that GA reaches the last generation, the best individual (i.e. ANN) is used to forecast the future (and unknown) time series values (a_t) one by one using the k previous known values (a_{t-1}, \dots, a_{t-k}). Value k is the number of input nodes of the best individual.

This contribution reports two methods, “*Shuffle*” and cross-validation, to obtain the pattern sets used for ANN learning algorithm in a previous approach [4] based on Genetic Algorithms (GA). “*Shuffle*” refers to the way the whole pattern set will be split between train pattern set and validation pattern set. Cross-validation will be used for time series with few elements, so that cross-validation will be used to obtain several pattern subsets which will help to evaluate more accurately every specific ANN obtained in the GA.

2 “*Shuffle*”, Cross-Validation and Ensembles

In previous work, train and validation sets are obtained in a sequentially manner (train first 70%, validation last 30%). But, in this new approach, “*shuffle*”, the process of splitting the patterns set will consist of obtaining train and validation sets in a random way from time series data, see (Fig. 1). So it will let different parts of the time series to train the ANN and also different parts of the time series to validate the ANN, in order to obtain better generalization ability.

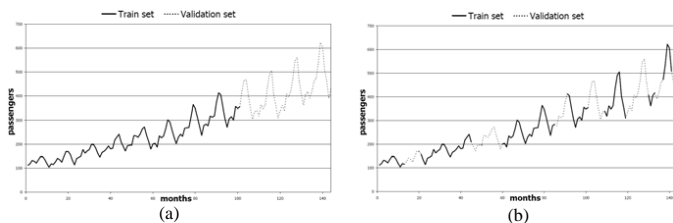


Fig. 1. Passengers: train and validation patterns sequentially (a) and randomly (b) obtained

Cross-validation has been used to forecast time series. In this study, the total pattern set will be split into n complementary pattern subsets (n from 2 to 8). Fig.2 shows an example of cross-validation with three pattern subsets. So that a individual in GA is an ANN topology, applying cross-validation to this individual gives n different ANN architectures (i.e. topology plus connection weights) and it n different fitness values depending on which patterns are used to train and validate the topology. So, applying cross-validation, the final fitness value for an individual will be the average of all its fitness values from each of its architectures.

When last generation of the GA is reached and its best individual have to be used to forecast, what of the its n architecture from cross-validation should be used? To solve this new problem Ensembles similar to Yao in [5] will be used to obtain only one model to carry out the final forecasting send to any competition or any company that need just a forecast (not several). In [5] the ensemble takes the different architectures obtained in last generation of the evolving cross-validation process, rather than an individual, to form the final result. But in our work, the different architectures (the same input and hidden nodes but different weight values) from the best individual in

the last generation using cross-validation are linearly combined as an ensemble. And this ensemble will be used to give only one forecast. The weight for each model of the linear combination is given by the eq. 1 (a), and the ensemble output is given by eq. 1 (b). (n is the number of models into the ensemble and β is a scaling factor).



Fig. 2. Example of cross validation with 3 patterns subsets

$$(a) w_i = \frac{\exp(\beta(n+1-i))}{\sum_{j=1}^n \exp(\beta_j)}; (b) O = \sum_{j=1}^n w_j o_j \quad (1)$$

3 Experimental Results and Conclusions

Five time series [6] have been used to evaluate our methods. Sequential and “*Shuffle*” ways to obtain train and validation subsets are evaluated into the system. Forecasted values are compared with real values (i.e. test set) and two error values are used: MSE (mean squared error) and SMAPE (symmetric mean absolute percent error) [7]. The results are shown in Table 1.

As it can be observed applying shuffle method to these time series does not achieve better forecasting in Passengers and Dow-Jones time series. It could be explained because of the few elements of those time series (less than 200). If train and validation pattern subsets obtained are split in a random way, then all the patterns used to adjust the connection’s weights does not correspond to consecutive time series values. So the relationship between inputs and output could be harder to learn if there are few patterns for learning and they are not consecutive (i.e. mixing up the training and validation patterns). On the other hand, the same experiment was also carried out with Quebec and Mackey-Glass time series, larger than previous ones (about 730 elements) Applying shuffle to these time series gets better results, specially for Mackey-Glass.

Our approach does not seem to achieve an improvement using “*Shuffle*” with short time series (i.e. Passengers, Temperature and Dow-Jones), so cross-validation, usually used when not too many data are given, have been tried for these time series. The number of subsets in which the total pattern set has been split goes from two to eight. All forecasted values, obtained from the ensemble of the ANN architectures are compared with real values and SMAPE error is shown. Results are shown in Table 2. We can observe that applying cross-validation to these time series obtain different results depending on the time series and the number of subsets the total pattern set has been split. The problem now arise in which is the optimum number of subsets which should be used to forecast a time series using cross-validation.

Table 1. SMAPE and MSE error for the best individual in the last generations

	Sequentially		Shuffle	
	MSE	%SMAPE	MSE	%SMAPE
Passengers (120 values)	0.00039	3.065	0.00521	8.999
Temperature (206 values)	0.00406	4.845	0.00384	4.441
Dow-Jones (129 values)	0.01437	5.512	0.02065	6.689
Quebec (735 values)	0.02149	12.121	0.01312	9.218
Mackey-Glass (731 values)	0.00363	8.672	0.00016	1.818

Table 2. SMAPE error using Cross-validation and Ensembles

	0 Sub	2 Sub	3 Sub	4 Sub	5 Sub	6 Sub	7 Sub	8 Sub
Passengers	3.065	<i>16.227</i>	11.104	6.774	4.519	9.136	4.198	2.790
Temperature	<i>4.845</i>	3.385	3.413	3.625	3.730	3.952	3.842	3.817
Dow-Jones	5.512	6.194	6.728	<i>7.581</i>	7.112	5.093	6.962	6.125

The results disclose that shuffle only improves forecasting for not short time series. An issue arises at this point: how the positive/negative effect of shuffle depends on the number of time series elements (i.e. size of training/validation subsets). On the other hand, cross-validation let us improve the result for short time series, but another issue arise, the optimum number of subsets to split the total pattern set. As it is a totally automatic method, it will not be necessary any previous knowledge from the user. The user just have to give the time series he wants to forecast and the number of future elements he wants to be forecasted to the system; and this method will give these forecasted values as result to the user. This approach got 6th position in NN5 Forecasting Competition [7].

Acknowledgments. The research reported here has been supported by the Spanish Ministry of Science and Innovation under project TRA2007-67374-C02- 02.

References

1. Zhang, G., Eddy Patuwo, B., Hu, M.Y.: Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting* 14(1), 35–62 (1998)
2. Abraham, A.: Meta learning evolutionary artificial neural networks. In: 3rd Int. Conf. on Neural, Parallel and Scientific Computations, Atlanta, USA (2006)
3. Yao, X.: Evolving artificial neural networks. *Proceedings of the IEEE* 87(9), 1423–1447 (1999)
4. Peralta, J., Gutierrez, G., Sanchis, A.: Shuffle design to improve time series forecasting accuracy. In: CEC 2009: Proceedings of the 11th Conference on Evolutionary Computation, pp. 741–748. IEEE Press, Los Alamitos (2009)
5. Yao, X.: Evolving artificial neural network ensembles. In: *Studies in Computational Intelligence*, pp. 851–880 (1993)
6. Hyndman, R.J. (n.d.): Time Series Data Library, <http://www.robjhyndman.com/TSDL> (accessed on June 2010)
7. ANN & CI Time Series Forecasting, <http://www.neural-forecasting-competition.com> (accessed on June 2010)