Weighted Cross-validation Evolving Artificial Neural Networks to Forecast Time Series

Juan Peralta Donate¹, Paulo Cortez², German Gutierrez Sanchez¹, Araceli Sanchis de Miguel¹

Abstract. Accurate time series forecasting is a key tool to support decision making and for planning our day to-day activities. In recent years, several works in the literature have adopted evolving artificial neural networks (EANN) for forecasting applications. EANNs are particularly appealing due to their ability to model an unspecified non-linear relationship between time series variables. In this work, a novel approach for EANN forecasting systems is proposed, where a weighted cross-validation is used to build an ensemble of neural networks. Several experiments were held, using a set of six real-world time series (from different domains) and comparing both the weighted and standard cross-validation variants. Overall, the weighted cross-validation provided the best forecasting results.

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 effectiveness in human life. It is a discipline that finds each day more applications in areas such as planning, management, production, maintenance and control of industrial processes, economy, and weather forecasting. The forecasting task can be performed by several techniques, including Statistical methods [1] and Computational Intelligence based methods, such as Immune Systems [2] and Artificial Neural Networks (ANN) [3,4,5]. ANNs provide a methodology for solving nonlinear problems that are difficult to be modeled by traditional techniques. Most time series processes often exhibit temporal and spatial variability and are suffered by issues of nonlinearity of physical processes, conflicting spatial and temporal scale and uncertainty in parameter estimates.

¹ Computer Science Department, University Carlos III of Madrid. Avenida de la Universidad 30 28911 Leganes, SPAIN. {Jperalta, Ggutierr, Masm}@inf.uc3m.es

² Department of Information Systems/Algoritmi, University of Minho. Campus de Azurém 4800 - 058 Guimarões, PORTUGAL. pcortez@dsi.uminho.pt

Several works have addressed time series forecasting with ANNs showing the effectiveness of using ANNs in this domain [3,4,5]. Modeling ANNs for forecasting involves the design of the ANN structure (e.g. number of input and hidden nodes) and setting of the training algorithm parameters. If a manual design of is carry out, several ANN setups (e.g. with different number of inputs neurons and learning rates) need to be set. For each of them, training is performed and validation errors are obtained and the one with the best generalization capability is selected to forecast the future values. As an alternative, automatic ANN design methods have been proposed. In particular, Evolving ANN (EANN) algorithms are becoming a popular solution, since they perform a global multi-point search, quickly locating areas of high quality, even when the search space is very complex [6,7,8,9]. Therefore, EANNs have been rapidly applied to time series forecasting [4,10]. In this work, we adopt the recently proposed Automatic Design of ANN (ADANN) system [5], which obtained high quality forecasting results.

Another crucial issue when using ANNs for forecasting domain is related with the quality of the data provided to the ANN. This issue is particularly relevant when dealing with short time series (i.e. with few elements), as there are few training patterns and it is more difficult for the ANN to learn how to generalize well. In this paper, we address this second issue by proposing the use of a weighted cross-validation learning scheme to create an ensemble of ANNs. Such scheme has the advantage of allowing the forecasting system to obtain more training patterns and thus may be more suited for providing accurate forecasts when modeling short time series.

The paper is organized as follows. Section 2 describes how to address time series forecasting tasks with EANN and weighted cross-validation ensembles. Next, we describe the experiments held and analyze the obtained results (Section 3). Finally, closing conclusions are drawn in Section 4.

2 Time Series Forecasting with Artificial Neural Networks (ANN)

2.1 Automatic Design of Artificial Neural Networks (ADANN)

The problem of forecasting the time series $y_1, y_2, \dots, y_t$ with ANN is considered as modeling the relationship (function f) from the value at period t (as we adopt only one output neuron) and the values from previous elements of the time series (time lags $t-1, t-2, \dots, t-k$):

$$y_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-k}) + e_t \quad (1)$$

Where e_t is the forecasting error. As the first step, the original values of the time series need to be normalized (within [0,1]). After training, the inverse process is carried out, transforming the ANN responses back to the original scale. Only one neuron is chosen at the output layer. Multi-step forecasts (i.e., 1 to N ahead forecasts) are built by iteratively using 1-ahead predictions as inputs. By adopting a sliding time window of size k , the time series is transformed into a pattern set, where each pattern consists of:

- k input values - corresponding to the k normalized previous values: $y_{t-1}, y_{t-2}, \dots, y_{t-k}$.
- one output value - corresponding to the normalized time series at period t that is predicted.

A given time series is first split into training and test data. The training data is used to create the pattern set, which is further split into training and validation sets. The former, with $x\%$ of the pattern set elements (i.e. training), is used to fit the ANN (i.e. adjust the connection weights), while the latter (i.e. validation), with the remaining patterns, is used to estimate the ANN generalization capabilities (i.e. how accurate is the ANN at predicting future values).

The problem of designing the best ANN can be seen as a search problem within the space of all possible ANN setups. Then, this search can be performed by a Genetic Algorithm (GA) [11], using both exploitation and exploration. Recently, we have proposed the ADANN system [4], which adopts such GA as follows:

1. Each chromosome consists of 16 digits (from 0 to 9), where: the first two digits set k (number of ANN inputs), the next two digits set the number of hidden nodes, the next two digits set up the back-propagation learning rate (divided by 100, i.e. from 0 to 1); and remaining digits set the random seed to generate the ANN initial weights.
2. A randomly generated population (with 50 individuals) is obtained.
3. The phenotype (i.e. ANN architecture) and fitness value of each individual of the current generation is obtained. To obtain the phenotype associated to a chromosome and its fitness value:
 - 3.1. The topology of an individual i from the actual generation is obtained.
 - 3.2. Then, the training and validation patterns subsets for this ANN i are obtained from time series data.
4. Once each ANN is initialized with its topology and connection weights values with the information from the chromosome, it is trained with the standard back-propagation algorithm. The architecture (topology and connections weights) of the ANN when the validation error is minimum during the training process is saved (i.e. early stopping). The validation pattern subset is used to

estimate the generalization capability, obtaining its fitness value. In this paper, this fitness is given by the Mean Squared Error (MSE).

5. Once the fitness values for whole population have been already obtained, the GA operators such as elitism, selection, one-point crossover and mutation (with a 0.07% rate) are applied in order to generate the population of the next generation, i.e. a new set of chromosomes.

The steps 2 and 3 are iteratively executed until a maximum number of generations are reached. Finally, the best individual from the last generation is used to compute the 1 to N ahead forecasts.

2.2 Cross-validation

Cross-validation, sometimes called rotation estimation, is a technique for assessing how a learning model will generalize to an independent data set. It is mainly used when the goal is prediction and one wants to estimate how accurately a predictive model will perform in practice. One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the validation set). To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds. Cross-validation is important in guarding against testing hypotheses suggested by the data (called Type III errors [12], especially where further samples are hazardous, costly or difficult to collect. Cross-validation has been used in previous works to forecast time series [13,14]. In [13] it was used to determine densities of noise terms when applying a Kalman smoother, which is a classical statistical tool to forecast time series. More closed to what is proposed in this work, in [14] a time ordered cross-validation is used to validate the training process of ANN when forecasting time series, by testing different number of pattern subsets (or folds), ranging from 2 to 8. Every time a unique fold is used as validation subset, the remaining pattern examples are used to train the ANN. In the next round, another fold is selected for validation and this process is repeated as many times as folds we have. Fig. 1 shows an example of a 5-fold cross-validation.

Fig. 1. Example of a 5-fold cross-validation.

While one individual (i.e. genotype) leads to a single ANN topology (i.e. phenotype), applying cross-validation to this individual results in n different ANNs (i.e. different connection weights for the same topology), where n is the number of folds. Given such scheme, there are two relevant questions: How to evaluate the fitness of these n ANNs and How to combine the outputs of the n different ANNs (explained in Section 2.3).

To measure the fitness of a cross-validation ensemble, the most common procedure is to weight equally each fold fitness, i.e. calculate the fitness value as the average of all n validation set errors (Fig. 1) [14]. Yet, in the forecasting domain, recent patterns should have a higher importance when compared with older ones. Following this assumption, in this paper, we propose the following weighted

cross-validation: $w_j = \frac{1}{2^{n+1-j}}$, for all $j \in \{2, \dots, n\}$ and $w_1 = 1 - \sum_j w_j$. For example, for $n=4$, then $w_1=w_2=0.125$, $w_3=0.25$ and $w_4=0.5$.

2.3 Ensembles

Ensemble building is a common way to improve the performance of the resulting model for classification and regression tasks, since it was noticed that an ensemble of individual models performs often better than a single predictor [15,16]. Usually, an ensemble consists of models taken from one single type of model, i.e. ANN. Unlike most EANN systems, which optimize a single ANN, the final result of the ADANN system consists of the best ANN ensemble, built using the n -fold cross-validation scheme without involving much extra computational cost. To combine the individual forecasts, we adopt the approach proposed in [17], where the final output depends on a linear combination of the individual predictions, such that the responses of each ANN are weighted proportionally to their fitness values. That is,

given n sorted architectures with an increasing error rate, where n is the population size, β is a scaling factor (set using preliminary experiments) and their outputs $O_1, O_2, \dots, O_n$, then the weight for the i_{th} architecture is:

$$W_i = \frac{\exp(\beta(n+1-i))}{\sum_{j=1}^n \exp(\beta_j)} \quad (2) \quad O = \sum_{j=1}^n w_j o_j \quad (3)$$

3 Experiments and Results

Six real-world time series, from distinct domains, were selected to evaluate the proposed methods [18]: Passengers, Temperature, Dow-Jones, IBM, Ozone and Paper. All these time series can be considered as small sized, as the number of elements range from 120 (paper) to 369 (IBM). We set a fixed number of 19 elements for the test set, thus for each series we perform from 1 to 19 ahead forecasts. The remaining data is used to train and validate the cross-validation ensembles. The global forecasting performance is evaluated the popular Symmetric Mean Absolute Percentage Error (SMAPE) metric (the lower the value, the better the forecast) [19].

Experiments have been run for each time series by applying cross-validation without and with weights. The number of subsets (i.e. folds) in which the total pattern set has been split ranges from 2 to 8. The obtained results are shown in Tables 1 and 2.

In these tables, rows will show forecasting (SMAPE error) for each time series and each column represents the number of subsets the whole pattern set has been split.

Table 1. Cross-validation without weights results (% SMAPE values, best values in **bold**)

↓Series\Folds→	0	2	3	4	5	6	7	8
Passengers	3.18	16.22	11.10	6.77	4.51	9.13	4.19	2.79
Temperature	4.22	3.38	3.41	3.62	3.73	3.95	3.84	3.81
Dow-Jones	6.67	6.19	6.72	7.58	7.11	5.09	6.96	6.12
IBM	5.10	4.65	3.18	4.03	3.45	3.38	2.10	10.78
Ozone	16.60	14.02	14.38	14.37	15.34	15.02	13.64	15.79
Paper	8.15	7.85	8.45	7.92	8.09	7.51	8.82	8.03
Average	7.32	8.71	7.87	7.38	7.03	7.34	6.59	7.88

Table 2. Cross-validation with weights results (% SMAPE values, best values in bold)

↓Series\Folds→	0	2	3	4	5	6	7	8
Passengers	3.18	16.22	10.74	5.08	10.16	5.45	4.07	3.39
Temperature	4.22	3.38	3.67	3.64	3.72	3.75	3.69	3.47
Dow-Jones	6.67	6.19	6.96	4.46	4.92	6.07	5.58	4.91
IBM	5.10	2.83	2.49	2.34	3.30	2.42	2.19	2.11
Ozone	16.60	15.31	16.12	15.20	13.76	15.59	15.66	15.44
Paper	8.15	7.89	8.41	7.52	7.84	7.53	7.51	8.82
Average	7.32	8.63	8.06	6.37	7.28	6.80	6.45	6.35

When analyzing both tables, it can be concluded that using weights in cross-validation is in general a better strategy, as the overall average (average of last row) for all time series is lower with weights (7.14%) than without weights (7.55%). Also, the average weighted cross-validation results, when adopting a fixed number of folds, is better in 5 of the 7 cases when weights are used, when compared with the no cross-validation (0 folds, 30% of the training data is used as the validation set) setup. In particular, when using weighted cross-validation in 3 cases (i.e. Temperature, IBM and Ozone), whatever the number of folds is used, the weighted cross-validation always outperforms the 0 fold method (i.e. simple hold-out validation). Regarding the best n value, it is dependent on the time series considered. Given the trade-off between the accuracy and computational cost, we suggest the use of a weighted 4-fold validation scheme, which obtains the second best overall performance (SMAPE=6.37%).

4 Conclusions

In this paper, we proposed an Evolving Artificial Neural Network (EANN) that is based on a weighted cross-validation scheme for time series forecasting. As shown in the previous section, cross-validation with weights improves the accuracy of the forecasts for short time series, outperforming both the cross-validation without weights and the standard hold-out validation methods. As a compromise between accuracy and computational cost, based on the presented results, we advise the use of a 4-fold weighted cross-validation. In future work, we intend to address additional time series. Another interesting future work direction is the use of sparsely connected [4].

References

1. Makridakis, S., Wheelwright, S., Hyndman, R.: Forecasting methods and applications. John Wiley & Sons, USA (2008)
2. Nunn, I. White, T.: The application of antigenic search techniques to time series forecasting. In: Proceedings of GECCO, 2005, pp. 353-360, ACM (2005)
3. Zhang, G., Patuwo, B.E., Hu, M.Y.: Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14: 35-62 (1998)
4. Cortez, P., Rocha, M. and Neves, J.: Time Series Forecasting by Evolutionary Neural Networks. In: J. Rubuñal and J. Dorado (Eds.), *Artificial Neural Networks in Real-Life Applications*. chapter III, pp. 47-70, Hershey, USA, Idea Group Publishing (2006)
5. Peralta, J., Li, X., Gutierrez, G., Sanchis, A.: Time series forecasting by evolving artificial neural networks using genetic algorithms and differential evolution. In: Proceedings of IJCNN, pp. 3999-4006, IEEE (2010)
6. Kitano, H.: Designing Neural Networks using Genetic Algorithms with Graph Generation System. *Complex Systems*, 4: 461-476 (1990)
7. Yao, X.: Evolving Artificial Neural Networks. *Proceedings of IEEE*, 9(87):1423-1447 (2002)
8. Abraham, A.: Meta-Learning Evolutionary Artificial Neural Networks. *Neurocomputing* 56(c):1-38 (2004)
9. Rocha, M., Cortez, P., Neves, J.: Evolution of Neural Networks for Classification and Regression. *Neurocomputing*, 70(16-18): 2809-2816 (2007)
10. Chena, Y., Chang, F.: Evolutionary artificial neural networks for hydrological systems forecasting. *Journal of Hydrology*, 367(1-2):125-137 (2009)
11. Fogel, D.: *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Wiley-IEEE Press (1998)
12. Mosteller, F.: A k-sample slippage test for an extreme population, *Annals of Mathematical Statistics*, 101-109 (2006)
13. Sarkka, S., Vehtari, A., Lampinen, J.: CATS benchmark time series prediction by kalman smoother with cross-validated noise density, *Neurocomputing*, 70(13-15): 2331-2341 (2007)
14. Wah, B., Qian, M.: Time-series predictions using constrained formulations for neural-network training and cross validation. In: *Proc. of 16th Int. Conf. on Intelligent Information Processing*, pp. 220-226. Kluwer Academic Press (2000)
15. Geman, S., Bienenstock, E., Doursat, R.: Neural networks and the bias/variance dilemma, *Neural Computing*, 4(1): 1-58 (1992)
16. Krogh, A., Sollich, P.: Statistical mechanics of ensemble learning, *Physical Review E*, 55(1): 811-825 (1997)
17. Yao, X., Islam, M.: Evolving artificial neural network ensembles. *IEEE Computational Intelligence Magazine*, 3(1): 31-42 (2008)
18. Hyndman, R.: Time series data library. In: <http://robjhyndman.com/TSDL/>, Accessed September (2010)
19. Hyndman, R., Koehler, A.: Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22 (4): 679-688 (2006)