Implementation of Artificial Intelligence in the Time Series Prediction Problem

L. EKONOMOU
Public Power Corporation S.A.
22 Chalcocondyli Str.
104 32 Athens

S.SP. PAPPAS
University of the Aegean
Information & Communications Systems Engineering
26 Deligeorgi Str.
17456 Athens
GREECE

Abstract: This paper presents the use of artificial intelligence and more specifically artificial neural networks, genetic algorithms and evolutionary algorithms in the solution of the time series prediction problem. The time series prediction problem is formulated as a system identification problem, where the input to the system is the past values of a time series and its desired output is the future values of a time series. A method has been developed based on the well known from the literature Genetics-Based Self-Organising Network (GBSON) method and has been applied to various time series data producing satisfactory results.

Keywords: Time series prediction, Genetics-Based Self-Organising Network (GBSON) method, Artificial neural networks, Genetic algorithms, Evolutionary algorithms

1. Introduction

A time series is a set of observations x_t, each one being recorded at a specific time t. A discrete time series is one where the set of times at which observations are made is a discrete set. Continuous time series are obtained by recording observations continuously over some time interval. An example of a discrete time series can be seen in Figure 1.

Analysing time series data led to the decomposition of time series into components. Each component is defined to be a major factor or force that can affect any time series. Three major components of time series have been identified. *Trend* refers to the long-term tendency of a time series to rise or fall. *Seasonality* refers to the periodic behaviour of a time series within a specified period of time. The fluctuation in a time series after the trend and seasonal components have been removed is termed as the irregular component [1].

In this work artificial intelligence is used in order to give solution to the time series prediction problem. The time series prediction problem is formulated as a system identification problem, where the input to the system is the past values of a time series, and its desired output is the future values of a time series. A method has been developed based on the well known from the literature Genetics-Based Self-Organising Network (GBSON) method and has been applied to thunderstorm days time series data that have been collected from the National Meteorological Authority of Hellas producing satisfactory results.

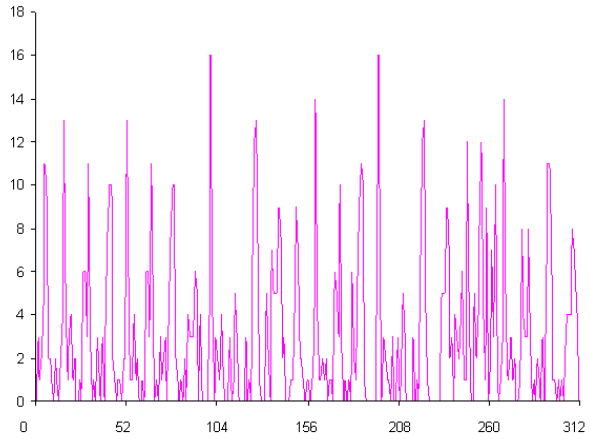

Figure 1: Monthly thunderstorm days, 01/1980-12/2005.

2. The Procedure of Time Series Signals Prediction

The prediction of time series signals is based on their past values. Therefore, it is necessary to obtain a data record. When obtaining a data record, the objective is to have data that are maximally informative and an adequate number of records for prediction purposes. Hence, future values of a time series $x(t)$ can be predicted as a function of past values $x(t\text{-}1)$, $x(t\text{-}2)$, …, $x(t\text{-}\varphi)$.

$$x(t+\tau) = f(x(t\text{-}1), x(t\text{-}2), \ldots, x(t\text{-}\varphi)) \quad (1)$$

The problem of time series prediction now becomes a problem of system identification. The unknown system to be identified is the function $f(\cdot)$ with inputs the past values of the time series.

While observing a system there is a need for a concept that defines how its variables relate to each other. The relationship between observations of a system or the knowledge of its properties is termed as the *model* of the system. Models can be given in several different forms. A *mental model* does not involve any mathematical formalisation, but the system's behaviour is summarised in a nonanalytical form in a person's mind. A mental model is a driver's perception of a car's dynamics. *Graphic models* make use of a graph or a table to summarise the properties of a system. *Mathematical models* are mathematic relationships among the system variables, often differential or difference equations. In system identification, a set of candidate models is specified, where the search for the most suitable one will be restricted.

The search for the most suitable model for a system is guided by an assessment criterion of the goodness of a model. In the prediction of time series, the assessment of the goodness of a model is based upon the prediction error of the specific model [2, 3].

After the most suitable model of a system has been determined, it has to be validated. The validation step in the system identification procedure is very important because in the model identification step, the most suitable model obtained was chosen among the predefined candidate models set. This step will certify that the model obtained describes the true system. Usually, a different set of data than the one used during the identification of the model, the *validation set*, is used during this step [4, 5].

In this paper a developed by the authors' method has been applied to various time series data producing very satisfactory results.

3. Genetics-Based Self-Organising Network (GBSON)

In [6], Kargupta and Smith proposed a method for system identification using evolving polynomial networks. This approach was motivated from the work of Ivakhnenko who introduced the Group Method of Data Handling (GMDH).

The method introduced by Kargupta and Smith is the Genetics-Based Self-Organising Network (GBSON). It is a hybrid method of the GMDH and Genetic Algorithms [7]. The GBSON method was introduced to overcome the drawbacks of the original GMDH algorithms, since they use local search techniques to obtain an optimal solution [8, 9].

The GBSON uses polynomial neural networks to represent the model of the system to be identified. Each layer of the polynomial neural network is regarded as a separate optimisation problem. The input to the first layer of the network is the independent variables of the data sample. The output of each layer is the peak nodes obtained by the use of a multi-modal Genetic Algorithm [10]. The peak nodes selected to be the output of a layer are also the inputs for the next layer.

The population members of the GA are network nodes represented by an eightfield bit string. The two first fields are used to represent the nodes from the previous layer connected to the present node. The other six fields are used to represent the coefficients of a quadratic function that determines the output of the node y:

$$y = a + bz_1 + cz_2 + dz_1z_2 + ez_1^2 + fz_2^2 \quad (2)$$

where z_1 and z_2 are the outputs of the connected nodes in the previous layer.

The fitness measure of a node is given by calculating its description length. The description length gives a trade off between the accuracy of the prediction and the complexity of the network. The equation used by Kargupta and Smith for calculating the description length is:

$$I = 0.5n \log D_n^2 + 0.5m \log n \quad (3)$$

where D_n^2 is the mean-square error, m is the number of coefficients in the model selected and n is the number of observations used to determine the mean-square error.

The multi-modal GA used in GBSON incorporates the fitness-sharing scheme, where the shared fitness is given by:

$$f_i' = \frac{f_i}{m_i} \quad (4)$$

f_i is the original fitness of the node and m_i is the niche count defined by:

$$m_i = \sum_{j=1}^{N} sh(d_{ij}) \quad (5)$$

where

$$sh(d_{ij}) = \begin{cases} 1 - \left(\dfrac{d_{ij}}{\sigma_s}\right)^a & if\ d_{ij} < \sigma_s \\ 0 & otherwise \end{cases} \quad (6)$$

N is the population size and d_{ij} is the Hamming

distance between the members of the population i and j. The niche radius σ_s is determined by the equation:

$$\frac{1}{2^l}\sum_{i=0}^{\sigma_s}\binom{l}{i}=\frac{l}{q} \quad (7)$$

where l is the string length and q is the number of nodes in the previous network layer.

New populations are obtained after applying the genetic operators of tournament selection, single-point crossover and point mutation. A mating restriction is also applied to the members to be crossed. If a member i is to be crossed, its mate j is selected such that $d_{ij} < \sigma_s$. If no such mate can be found then j is selected randomly.

The GBSON procedure continues until the GA converges to a layer with a single node.

3. Simulation Results

3.1 Thunderstorm Days Series

The first set of experiments was conducted on monthly thunderstorm days numbers, recorded by the National Meteorological Authority of Hellas [11], from January 1980 to December 2005. These numbers are indicative of the average relative number of thunderstorm days observed every month of the year. The thunderstorm days are strongly related to the lightning. In result, the thunderstorm days can determine the lightning level of an area, i.e. the number of lightning flashes to earth. The prediction of the thunderstorm days is therefore essential to the studies of transmission and distribution lines' designers, since the knowledge of the future lightning level of an area can result in a better design and consequently to the reduction of the lightning faults in lines.

The thunderstorm days time series has been classified as quasiperiodic, and it has been found that the period varies between 8 to 12 years with irregular amplitudes, making the time series hard to predict.

The objective of the experiment is to generate a single-step prediction based on past observations. The data were normalised to take values from zero to one, before using them as input data to the polynomial neural networks. The input pattern was assigned as ($x(t$-1), $x(t$-2), $x(t$-3)) and the desired output was:

$$x(t) = f((x(t-1),\ x(t-2),\ x(t-3))$$

From the 312 available data points, 52 points (208 to 260) were used for the validation of potential models. The experiments were run with a population size of 20 for 100 generations, with tournament size 4, probability of crossover 0.95 and probability of mutation 0.01.

GBSON resulted to a network with three layers to model the thunderstorm days series. The most significant term in the partial descriptions,

$$y = a + bx_i + cx_j + dx_i x_j + ex_i^2 + fx_j^2 \quad (8)$$

of the model was the term x_j and the less significant term was the constant term.

The past values of the thunderstorm days series, ($x(t$-1), $x(t$-2), $x(t$-3)), contributed equally to obtain the final model.

The results of the prediction can be seen in Figure 2. The actual error of the prediction is shown in Figure 3. The percent square error (PSE) over the whole data set is 0.043825 and the root mean square error (RMSE) is 0.007243. The PSE over the validation data set is 0.067253. The difference of the PSE over the whole data set and the validation data set is small, and thus the model obtained performs with approximately the same accuracy in data points that have not been used in any part of the modelling process.

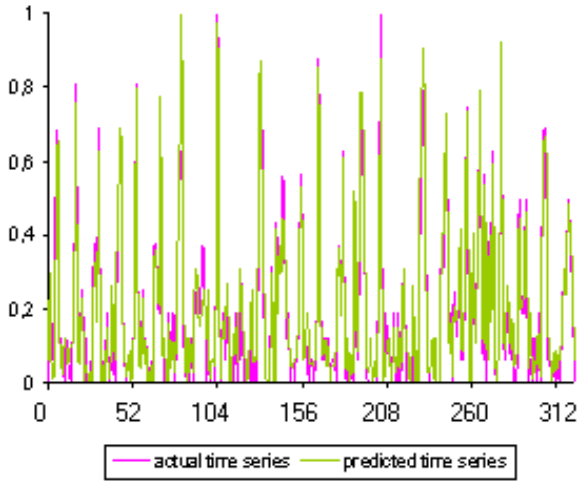

Figure 2: The actual thunderstorm days series and the predicted with the proposed method.

3.2 Lorentz Attractor Series

Edward Lorentz obtained the Lorentz attractor system, in his attempt to model how an air current rises and falls while it is heated by the sun. The Lorentz attractor system is defined by the following three ordinary differential equations.

$$\frac{dx(t)}{dt} = \sigma\, x(t) - \sigma\, y(t)$$

$$\frac{dy(t)}{dt} = -y(t) + r\, x(t) - x(t)\, y(t)$$

$$\frac{dz(t)}{dt} = -bz(t) + x(t)\,y(t)$$

The Lorentz attractor system has also been used to model a far-infrared NH_3 laser that generates chaotic intensity fluctuations [12]. The far-infrared NH_3 laser is described by exactly the same equations, only the variables and constants have different physical meaning.

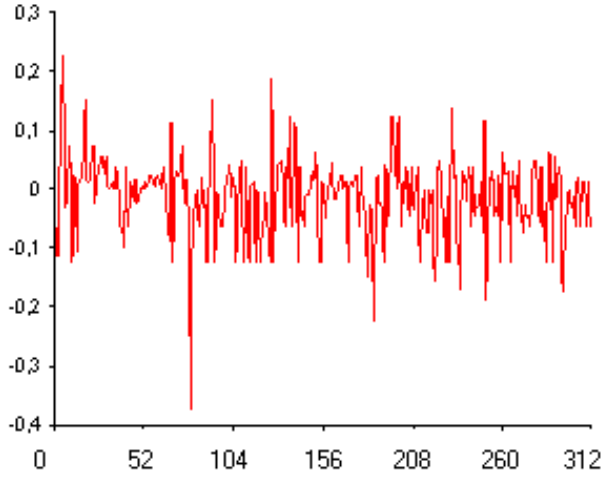

Figure 3: The actual error for each point of the thunderstorm days series predicted with the proposed method.

The time series used in this experiment, is the *x-component* in the Lorentz equations. The data were generated by solving the system of differential equations, that describe the Lorenz attractor, with the initial conditions of σ = 10, r = 50 and b = 8/3. The data were again normalised to take values from zero to one, before they were used as inputs to the polynomial neural networks [13-16].

The objective is to make one-step ahead prediction. The prediction is based on four past values ($x(t$-1), $x(t$-2), $x(t$-3), $x(t$-4)) and thus the output pattern is:

$$x(t) = f((x(t-1), x(t-2), x(t-3), x(t-4))$$

The experiments were performed with 100 members in each population for 500 generations, with tournament size 6, probability of crossover 0.95 and probability of mutation 0.03. The data points 2000 to 2500 were used for model validation.

The network constructed by the GBSON method to model the Lorentz attractor has eight layers. The most significant term in the partial descriptions:

$$y = a + bx_i + cx_j + dx_i x_j + ex_i^2 + fx_j^2$$

of the model was the term x_i^2 and the less significant term was the constant term. The input variables x_3 and x_4, were the most significant variables in the model. The results of the prediction and the actual system can be seen in Figure 3. The actual error of the prediction for each data point is shown in Figure 4. The PSE over the whole data set is 0.000244 and the RMSE is 0.000050. The PSE over the validation data set is 0.000231. The difference of the PSE over the whole data set and the validation data set is small, and thus the generalisation of the network is very good.

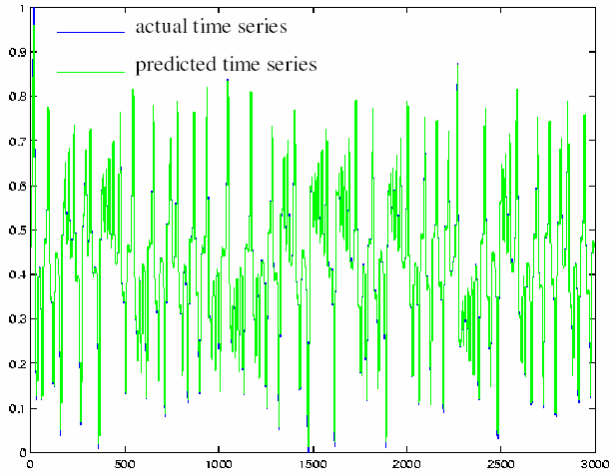

Figure 3: The predicted with the proposed method time series and the actual Lorenz attractor system time series.

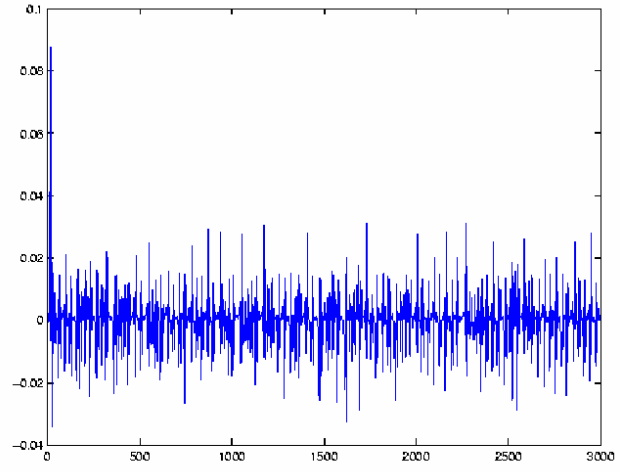

Figure 4: The actual error for each data point obtained from the prediction of the Lorenz attractor system time series.

4. Conclusions

The paper has presented the use of artificial intelligence and more specifically artificial neural networks, genetic algorithms and evolutionary algorithms in the solution of the time series prediction problem. The time series prediction problem has been formulated as a system identification problem, where the input to the system was the past values of a time series, and its desired output is the future values of a time series. A method has been developed based on the well known from the literature Genetics-Based Self-Organising Network (GBSON) method and has been applied to two different sets of significant time series data producing very satisfactory results.

References:

[1] T. Izumi, Y. Iiguni, "Data compression of nonlinear time series using a hybrid linear/nonlinear predictor", Signal Processing, vol. 86, no. 9, 2006, pp. 2439-2446.

[2] Z. Lu, "A regularized minimum cross-entropy algorithm on mixtures of experts for time series prediction and curve detection", Pattern Recognition Letters, vol. 27, no. 9, 2006, pp. 947-955.

[3] J.M. Matías, W. González-Manteiga, J. Taboada, C. Ordóñez, "Managing distribution changes in time series prediction", Journal of Computational and Applied Mathematics, vol. 191, no. 2, 2006, pp. 206-215.

[4] A. Yadav, D. Mishra, R.N. Yadav, S. Ray, P.K. Kalra, "Time-series prediction with single integrate-and-fire neuron", Applied Soft Computing, In Press, Corrected Proof, available online 18 April 2006.

[5] R.N. Yadav, P.K. Kalra, J. John, "Time series prediction with single multiplicative neuron model", Applied Soft Computing, In Press, Corrected Proof, available online 9 March 2006.

[6] H. Kargupta, R.E. Smith, "System identification with evolving polynomial networks", Proceedings of the 4th International Conference on Genetic Algorithms, 1991, pp. 370-376.

[7] http://www.inf.kiev.ua/GMDH-home/GMDH_res.htm

[8] X. Yao, Y. Liu, "Making use of population information in evolutionary artificial neural networks", IEEE Trans on Systems, Man, and Cybernetics-Part B: Cybernetics, vol. 28, no. 3, 1998, pp. 417-425.

[9] D. Quagliarella, J. Periaux, C. Poloni, G. Winter, "Generic algorithms and evolution strategies in engineering and computer science", Wiley, 1998.

[10] O.V. Pictet, M.M. Dacorogna, R.D. Dave, B. Chopard, R. Schirru, M. Tomassini, "Genetic algorithms with collective sharing for robust optimization in financial applications", Neural Network World, vol. 5, no. 4, 1995, pp. 573-587.

[11] Data supplied from the National Meteorological Authority of Hellas, 2005.

[12] H. Kantz, T. Schreiber, "Nonlinear time series analysis", Cambridge University Press, 1997.

[13] N.E. Mastorakis, "Solving differential equations via genetic algorithms", Proc. of the Circuits, Systems and Computers '96 (CSC'96), Piraeus, Greece, July 15-17, 1996, 3rd Volume: Appendix, pp.733-737.

[14] N.E. Mastorakis, "Genetic Algorithms and Nelder-Mead Method for the solution of boundary value problems with the collocation method", 5th WSEAS Int. Conf. on Simulation, Modeling & Optimization, Corfu, Greece, August 17-19, 2005 (pp690-694).

[15] N.E. Mastorakis, "Unstable ordinary differential equations: solution via genetic algorithms and the method of Nelder-Mead", 6th WSEAS Int. Conf. on Systems Theory & Scientific Computation (ISTASC'06), Crete, Greece, August 18-20, 2006.

[16] N.E. Mastorakis, "The singular value decomposition (SVD) in tensors (multidimensional arrays) as an optimization problem. solution via genetic algorithms and method of Nelder-Mead", 6th WSEAS Int. Conf. on Systems Theory & Scientific Computation (ISTASC'06), Crete, Greece, August 18-20, 2006.