

A COMPARATIVE STUDY OF EVOLUTIONARY ALGORITHMS FOR TRAINING ELMAN RECURRENT NEURAL NETWORKS TO PREDICT AUTONOMOUS INDEBTEDNESS

Cuéllar M.P., Navarro A., Pegalajar M.C and Pérez-Pérez R.

Dpto. Ciencias de la Computación e Inteligencia Artificial, ETS Ingeniería Informática, C/. Daniel Salcedo Aranda s.n. (18071), Universidad de Granada, Granada (Spain)

Keywords: Recurrent Neural Networks, Genetic Algorithms, Niching, CHC, Time Series Prediction

Abstract: This paper presents a training model for Elman recurrent neural networks, based on evolutionary algorithms. The proposed evolutionary algorithms are classic genetic algorithms, the multimodal clearing algorithm and the CHC algorithm. These training algorithms are compared in order to assess the effectiveness of each training model when predicting Spanish autonomous indebtedness.

1 INTRODUCTION

Many techniques have been used to predict a time series. The use of neural networks applied to this problem has increased in recent years, and these have obtained excellent results. Well-known examples of such networks are the FIR neural network (Wan, 1993; Cuéllar, 2003) and the recurrent neural network (Mandic, 2001; Haykin). In this paper, we shall focus on recurrent neural networks and in particular, the Elman model.

Traditionally, recurrent neural networks have been trained with gradient-based algorithms such as RTRL (Mandic, 2001; Haykin) or BPTT (Wan, 1993). However, one disadvantage of such algorithms is that they may easily become trapped in a local minimum. While genetic algorithms may also become trapped in local optimums, these algorithms have enough resources to avoid the problem (Blanco, 2001). Each process consists of two stages: training a neural network using evolutionary algorithms (Blanco, 2000, 2001), and predicting the time series with the trained neural networks.

Section 2 describes the evolutionary algorithms proposed. In Section 3, we introduce the Elman recurrent neural network model, and in Section 4, we explain how this may be trained with evolutionary algorithms. The results obtained are

shown in Section 5, and the conclusions in Section 6.

2 EVOLUTIONARY ALGORITHMS

Evolutionary Algorithms are optimization, searching and learning algorithms, based on nature and genetic evolution processes. In this paper, we work with three different types of Evolutionary algorithms.

2.1 Genetic Algorithms

Genetic algorithms (GA) (Cuéllar, 2003; Goldberg, 1989; Blanco, 2000, 2001; Back, 1996, 1997) base the evolution process on the recombination of individuals from a population of solutions, and also on the probability that new generated individuals can mutate to construct non-explored solutions. The basic scheme of a genetic algorithm is:

0. $t = 0$; $P(t)$ = population at time t .
1. While stopping condition is not satisfied, do:
 - 1.1. Selection Operator
 - 1.2. Cross Operator
 - 1.3. Mutation Operator
 - 1.4. $P(t+1)$ = replacement on $P(t)$

2.2 Multimodal Clearing Algorithm

Multimodal algorithms (Pérowski) are very similar to genetic algorithms. The main difference between them lies in the fact that multimodal algorithms evolve different areas (niches) in the search space. A niche is a set of individuals which can share certain resources or properties. In the problem explained in this paper, the property used as a relationship between individuals is that the Euclidean distance between their genes must be below a certain threshold, called the niching ratio. In order to carry out the cross operation, k best individuals of each niche are selected, and we allow the N best individuals among these to be the parents in the cross operator, where N is the size of the population.

2.3 CHC Algorithm.

Generally, the CHC algorithm (Rawlins, 1991) is used to solve binary-coded problems. It was one of the first proposals of evolutionary algorithms that introduced a balance between diversity and convergence factors. This algorithm combines an elitist selection which preserves the best individuals in the population with a cross operator that generates descendants which are as different as possible from the parents. As our problem must be real-coded, we propose a CHC variation so that we can work with real-coded solutions.

3 ELMAN RECURRENT NEURAL NETWORKS

In this paper, we use the Elman recurrent neural network model. This model has three neuron layers: one is used as the input data layer, another as the hidden neuron layer, and the third as the output data neuron layer. Furthermore, at the current time, the network saves the values (which previously had hidden neurons) on a layer called the state neuron layer. There is the same number of state neurons as hidden ones. Below, we shall show the equations which govern the behaviour of the network:

$$neth_h(t) = \sum_{h=1}^{NHID} U_{jh} S_h(t-1) + \sum_{i=1}^{NIN} V_{ji} X_i(t) \quad (\text{Eq. 1})$$

$$netout_k(t) = \sum_{j=0}^{NHID} W_{kj} S_j(t) \quad (\text{Eq. 2})$$

$$S_j(t) = f(neth_j(t)) \quad (\text{Eq. 3})$$

$$Y_k(t) = g(netout_k(t)) \quad (\text{Eq. 4})$$

- $Y_k(t)$: output of neuron k on the output layer at time t
- $Netout_k(t)$: output of neuron k on the output layer, while the activation function has not yet been applied
- $Neth_k(t)$: output of neuron k on the hidden layer when the activation function has not yet been applied
- $S_k(t)$: output of neuron k on the hidden layer
- **NHID**: number of hidden neurons
- **NIN**: number of input data neurons
- The net state is defined by the $S_k(t-1)$ values for the **NHID** hidden neurons at time t
- $F(\cdot)$: activation function for a hidden neuron
- $G(\cdot)$: activation function for an output neuron
- V : weights from the input to the hidden neurons
- U : weights from the state to the hidden neurons
- W : weights from the hidden to the output neurons
- $X(t)$: net inputs at time t

Values V_{ij} , U_{ij} , W_{ij} are the weights where j neuron is the source and i neuron is the target.

4 TRAINING MODEL

The architecture of the Elman neural network has three kinds of weights: input-hidden, state-hidden, and hidden-output weights. If the input-hidden weights are labelled V , the state-hidden weights U , and the hidden-output weights W , then Figure 1 shows the structure to encode an Elman network as a chromosome (Delgado; Blanco, 2000, 2001).

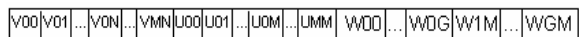


Figure 1: Structure of a chromosome

5 EXPERIMENTAL RESULTS

Having presented the models used in this paper, we shall compare their performance when used to predict autonomous indebtedness in Spain.

5.1 Indebtedness Data

We have the GDP data for each autonomous community in Spain between 1986 and 2000, and we will attempt to predict the 2001 GDP value.

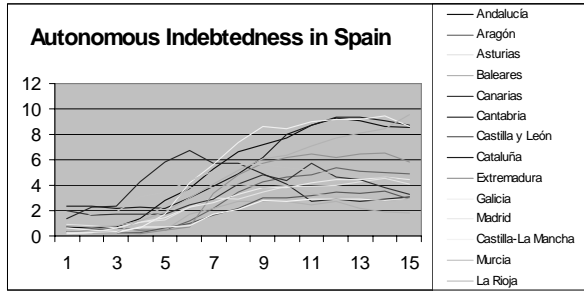


Figure 2: GDP data for each autonomous community between 1986 and 2000

5.2 Results Obtained

The algorithms has been run four times. The best results obtained with each algorithm are shown in Tables 1-3:

Table 1. Mean Square Error and GDP prediction for the year 2001, obtained with genetic algorithms

COMMUNITY	MSE	PREDICTION
Andalucía	8.943364e-03	8,531215
Aragón	6.640314e-03	4,975276
Asturias	4.125895e-03	4,162059
Baleares	5.479581e-03	1,68755
Canarias	2.758203e-02	2,629222
Cantabria	1.949228e-02	3,518876
Castilla y León	9.715380e-03	3,701863
Cataluña	1.417142e-03	8,300801
Extremadura	2.988611e-02	6,285624
Galicia	3.228254e-02	9,140918
Madrid	2.781795e-03	4,556016
Castilla - La Mancha	3.512358e-04	2,896469
Murcia	2.097112e-03	4,651921
La Rioja	4.581175e-03	2,572255
Valencia	1.904009e-02	10,239502

Table 2: Mean Square Error and GDP prediction for the year 2001, obtained with a real-coded modification of the CHC algorithm

COMMUNITY	MSE	PREDICTION
Andalucía	4.822338e-04	9.052394
Aragón	9.599990e-04	5.984290
Asturias	5.168292e-04	3.957864
Baleares	1.167222e-06	2.667783
Canarias	1.313612e-03	5.317874
Cantabria	1.152270e-04	5.253256
Castilla y León	8.388055e-05	4.916869
Cataluña	7.945567e-03	9.408566
Extremadura	1.060049e-03	6.441150
Galicia	1.077279e-02	9.140441
Madrid	9.207039e-10	4.770146
Castilla - La Mancha	4.644942e-04	2.870365
Murcia	1.416471e-02	3.436471
La Rioja	9.128470e-04	1.869554
Valencia	2.499737e-03	7.651457

Table 3: Mean Square Error and GDP prediction for the year 2001, obtained with the multimodal clearing algorithm

COMMUNITY	MSE	PREDICTION
Andalucía	1.656947e-03	9.429503
Aragón	9.672948e-04	5.605032
Asturias	6.392529e-04	3.955628
Baleares	1.289698e-03	2.772098
Canarias	2.655752e-03	9.388795
Cantabria	1.204943e-03	2.898273
Castilla y León	5.458950e-04	2.783321
Cataluña	3.320007e-03	9.310118
Extremadura	2.619883e-03	6.101216
Galicia	2.769566e-03	8.398488
Madrid	3.640982e-03	6.261248
Castilla - La Mancha	4.247670e-04	2.868603
Murcia	9.254246e-04	4.655863
La Rioja	6.494098e-03	0.112020
Valencia	2.367001e-05	8.290122

As we can see in the previous tables, the CHC and the clearing algorithms obtained the minimum MSE. Nevertheless, these results could be confusing. For instance, Figure 4 shows that although in the first data points the adjustment between the prediction and the real data is excellent, when we reach the last points, the error increases and the prediction cannot be trusted. The clearing and the CHC algorithms obtain a great search depth in the

solution space because of their performance. When there is a few amount of input data, the CHC and clearing algorithms overfit the data and the results are worse than if genetic algorithms were used. This is what happens in our case: although the mean square error obtained with genetic algorithms is worse than that obtained with the other algorithms, the adjustment of real and prediction data points is better at a general stage and prediction is therefore more trustworthy. Figures 3-4 show an example of

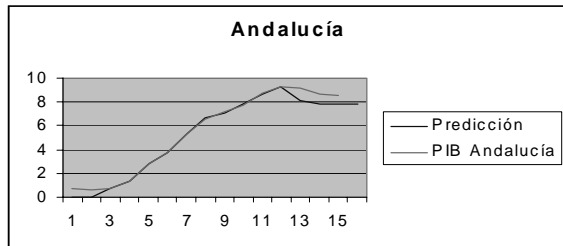


Figure 3: GDP prediction for Andalucía with CHC

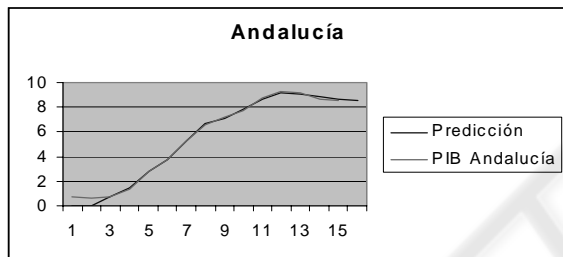


Figure 4: GDP prediction for Andalucía with GA

GDP prediction with CHC and genetic algorithms, for the community of Andalucía.

6 CONCLUSIONS

In this paper, we have studied a set of evolutionary models to train an Elman recurrent neural network, applied to time series prediction. These models have proved to be a good tool to predict Spanish autonomous indebtedness. Furthermore, genetic algorithms enable Elman networks to be trained easily, and prediction to be the most approximate possible. The average MSE obtained between each community is 0.0116. This means that not only is prediction good, but also that the model works uniformly in every community. The standard deviation of output data is 0.011056, which corroborates what we have just said.

REFERENCES

- Pétrowski A. A Clearing Procedure as a Niching Method for Genetic Algorithms. In IEEE pp. 798-803. 1996.
- Cuéllar M.P., M.A. Navarro, M.C. Pegalajar, R. Pérez. A FIR Neural Network to model the autonomous indebtedness In *SIGEF'03 Congress*, vol I. Leon. Pp. 199-209
- Cuéllar M.P., Delgado M., Pegalajar M. y Pérez R.. Predicción del endeudamiento económico español utilizando modelos bioinspirados. In *SIGEF'03 Congress*, vol II. Leon Pp.
- Danilo P. Mandic, Jonathon A. Chambers; *Recurrent Neural Networks for Prediction*. Wiley, John & Sons, Incorporated. 2001.
- Delgado M., Pegalajar M.C.. A Multiobjective Genetic Algorithm for obtaining the optimal size of a Recurrent Neural network for Grammatical Inference. In *Pattern Recognition, Special Issue of Grammatical Inference*. (in press)
- D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, 1989
- Blanco, Delgado y Pegalajar (2001). A Real-Coded genetic algorithm for training recurrent neural networks. In *Neural Networks 14*, 93-105
- Blanco, Delgado y Pegalajar. A genetic algorithm to obtain the optimal recurrent neural network. In *International Journal of Approximate Reasoning*, 23, pp. 67-83. 2000
- Gregory JE Rawlins, *Foundations of Genetic Algorithms*. ED. Morgan Kauffman. 1991.
- Schmidhuber J.. A fixed Size Storage $O(n^3)$ Time Complexity Learning Algorithm for Fully Recurrent Continually Running Networks. In *Neural Computation 4*, pp. 243-248. 1992.
- Simon Haykin, *Neural Networks (a Comprehensive foundation)* Second Edition, Prentice Hall
- T. Back, D. Fogel, Z. Michalewicz, *Handbook of Evolutionary Computation*. Institute of Physics Publishing and Oxford University Press. 1997.
- T. Back, *Evolutionary Algorithms in Theory and Practice*. Oxford. 1996.
- Wan. *PhD Dissertation*. Stanford University. Nov. 1993.