Design of Artificial Neural Networks based on Genetic Algorithms to Forecast Time Series

Juan Peralta, German Gutierrez, Araceli Sanchis
{jperalta,ggutierr,masm}@inf.uc3m.es
CAOS Group, Computer Science Department, University Carlos III of Madrid

***Abstract*— In this work an initial approach to design Artificial Neural Networks (ANN) using Genetic Algorithms (GA) is tackle. A key issue for these kind of approaches is what information contains, or is included, in the chromosome that represent an ANN, and there are two principal ideas about these question: first, information about parameters of the topology, architecture, learning parameters, etc. of ANN, i.e. Direct Encoding; second, initial information related to a constructive method that give rise to an ANN topology or architecture, i.e Indirect Encoding. The results for an Direct Encoding (in order to compare with Indirect Encoding developed in future works), for design ANN to NN3 Forecasting Time Series Competition.**

I. Introduction

THIS document report the methodology to carry out the automatic design of ANN that tackle the Forecasting Time Series problem taken from NN3 Forecasting Time Series Competition [1].

According to [2], that show the approximation capability of Multilayer Perceptron (MLP), we have focused on MLP with only one hidden layer, and Backpropagation (BP) to forecasting time series.

Several works approach the design of ANN using Evolutionary Techniques. Some of them uses Direct Encoding Scheme [3,4], the others using Indirect Encoding Scheme [5,6,7]. Ajith Abraham [8] shows an automatic framework for optimization ANN in an adaptive way. And [9] try to spell out the future trends of the field.

II. Forecasting Time Series With ANN

In order to make that a single ANN could work, and forecasting, with time series values, an initial step from original values of time series have to be done, i.e. normalize the data. And, once the ANN gives those values, the inverse process is done. This step is important as the ANN will learn just the normalized values.

The problem of forecasting time series with ANN is considered as obtain the relationship of the value of period "*t*" (i.e. the net has only one output neuron) and the values of previous periods, i.e the function:

$$a_t = f(a_{t-1}, a_{t-2}, \cdots, a_{t-k})$$

Therefore, the time series will be transform into a pattern set, it depend on the *k* inputs nodes of a particular ANN. Each pattern consists in:

- "*k*" inputs values, that correspond to "*k*" normalized previous values of period *t*: $a_{t-1}, a_{t-2}, \cdots, a_{t-k}$
- One output value, that corresponds to normalized time series value of period *t*.

The complete patterns set are ordered into the same way the time series is. This patterns set will be used to train and test the ANN, then it will be split into two sets, train and test sets. The train set will be obtained from the first *m*% (e.g *m*=70) and the test set will be obtained from the rest of the complete patterns set.

If hand design of ANN is carry out, several topologies (i.e. different number of inputs nodes and number of different number of neurons of only one hidden layer), with different learning rates are trained. For each of them, train and test error are obtained, and one with better generalization capability (i.e. less test error and a good train error) is selected to generate forecast values.

III. ANN Design with Genetic Algorithms

The problem of design ANN could be seen as a search problem into the space of all possible ANN. And that search can be done by a GA [10] using exploitation and exploration.

There three crucial issues: the solution's space; how each solutions is codified into a chromosome, i.e. encoding scheme; and what is looking for, translated into the function fitness.

The final object of our approach is consider Sparsely Connected MLP to forecast time series, and use both Indirect Encoding Scheme, one based on Cellular Automata [11] and other based on Bidimensional Grammar [12], to design ANN using GA.

But as a first approach to design ANN to forecasting time series, a Direct Encoding Scheme for Full Connected MLP has been considered. For this Direct Encoding Scheme the information placed into the chromosome is:

- Number of inputs nodes (i).

Manuscript received May 14, 2007.

Juan Peralta, is with the CAOS Group, as part of Computer Science Department of Univesity Carlos III of Madrid, Leganes, 28911 Spain (corresponding author to provide phone: +34 91 624 94 24; fax: +34 91 624 94 24; e-mail: jperalta@ inf.uc3m.es).

German Gutierrez and Araceli Sanchis are with the CAOS Group, as part of Computer Science Department of Univesity Carlos III of Madrid, Leganes, 28911 Spain ({ggutierr,masm}@ inf.uc3m.es)

- Number of hidden nodes (h).
- Learning rate, for BP learning algorithm (α).

The value of learning rate, "*α*", is between 0 and 1, and the value of "*i*" and "*h*" is limited by a maximum number of inputs nodes (max_inputs) and a maximum number of hidden nodes (max_hidden), respectively. These maximum values are related, by a factor "*a*", with the number of periods of time series what values are known (nts), see (1)

$$\max\_inputs = a \times nts;$$
$$\max\_hidden = 2 \times \max\_inputs; \quad (1)$$
$$a = 0.3$$

Into the chromosome, two decimal digits, i.e two genes, are used to codified the value "*i*", other two for "*h*" and the last two for "*α*". This way,the values of "*i*", "*h*" and "*α*" are obtained, from the chromosome, as it can see in (2)

$$chrom: gi_1\ gi_2\ gh_1\ gh_2 g\alpha_1 g\alpha_2$$
$$i = \max\_inputs \times \frac{(gi_1 \cdot 10 + gi_2)}{100};$$
$$h = \max\_hiddens \times \frac{(gh_1 \cdot 10 + gh_2)}{100}; \quad (2)$$
$$\alpha = \frac{(g\alpha_1 \cdot 10 + g\alpha_2)}{100}$$

The search process that GA is, begins with a random population, i.e set of randomly generated chromosomes. Later, the fitness value for each one of the individual of the population is obtained (a). Once that it is already done the GA operators as Elitism, Selection, Crossover and Mutation are applied in order to generate the population of next generation, i.e. set of chromosomes (b). The steps (a) and (b) are iteratively executed till a maximun number of generations is reached.

To obtained the fitness value of a chromosome:

1. The train patterns and test patterns sets are obtained, dependending on the number of inputs nodes of the net, as was said above (sec.II).
2. Then, the connection wegihts are randomly initialized, and the net is trained (using Stuttgart Neural Network Simulator (SNNS) binary tools [13]) a maximum training cycles.

The fitness value will be the minimun error test reached in training process, it doesn´t have to be in the last training cycle. The architecture of the net (topology + connections weights) when the test error is minimum in training process is saved to be used later for forecasting time series values.

Once that GA reaches the last generation, the best individual from all generations is used to forecast the time series.

IV. Conclusion

The final object of our approach is consider Sparsely Connected MLP to forecast time series, and use both Indirect Encoding Scheme, one based on Cellular Automata [11] and other based on Bidimensional Grammar [12], to design ANN using GA.

Appendix

Stuttgart Neural Network Simulator (SNNS) binary tools [13], ff_bignet and batchman, has been used for generate ánd training ANN.

The whole source code is available in {ggutierr,jperalta}@inf.uc3m.es.

References

[1] 2006/07 Forecasting Competition for Neural Networks & Computational Intelligence. http://www.neural-forecasting-competition.com/
[2] G. Cybenko. Approximation by superposition of a sigmoidal function. Mathematics of Control, Signals and Systems, 2, 303-314, 1989
[3] T. Ash. Dynamic Node Creation in Backpropagation Networks ICS Report 8901, The Institute for Cognitive Science, University of California, San Diego (Saiensu-sh, 1988), 1988
[4]] D.B. Fogel, Fogel L.J. and Porto V.W. Evolving Neural Network, Biological Cybernetics, 63, 487-493, 1990.
[5] Gruau F. "Genetic Synthesis of Boolean Neural Netorks with a Cell Rewriting Developmental Process". Proceedings of COGANN-92 International Workshop on Combinations of Genetic Algorithms and Neural Networks, pp. 55-74, IEEE Computer Society Press, 1992.
[6] Yao, X. and Lin, Y. A new evolutionary system for evolving artificial neural networks, Transactions on Neural Networks, 8(3): 694-713, 1997.
[7] Kitano, H.: Designing Neural Networks using Genetic Algorithms with Graph Generation System, Complex Systems, 4, 461-476, 1990.
[8] Ajith Abraham, Meta-Learning Evolutionary Artificial Neural Networks, Neurocomputing Journal, Elsevier Science, Netherlands, Vol. 56c, pp. 1-38, 2004.
[9]] H. Garis, D. Fogel, M. Conrad and X. Yao. Special Issue on evolutionary neural systems. Neurocomputing, 42, 1-8, 2002
[10] Fogel, D. Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. Wiley-IEEE Press, 1998
[11] Non-direct Encoding method based on Cellular Automata to design Neural Network Architectures. G. Gutiérrez, A. Sanchis, P. Isasi, J.M. Molina and I. M. Galván. Computer and Informatics, 24 2005, No3.
[12] Studying the Capacity of Grammatical Encoding to Generate FNN Architectures. Germán Gutiérrez, Beatriz García, José M. Molina and Araceli Sanchis. Computational Methods in Neural Modeling, Lecture Notes in Computer Science, Volume 2686/2003. IWANN 2003
[13] Prof. Dr. Andreas Zell,, WSI Computer Science Department, Computer Arquitecture, Software, Artificial Neural Networks <http://www-ra.informatik.uni-tuebingen.de/SNNS/>