ADANN: Automatic Design of Artificial Neural Networks

Juan Peralta
Computer Science Department
University Carlos III
Av. Universidad 30 Leganes (Madrid)
(34) 91624 9424
jperalta@inf.uc3m.es

German Gutierrez
Computer Science Department
University Carlos III
Av. Universidad 30 Leganes (Madrid)
(34) 91624 9135
ggutierr@inf.uc3m.es

Araceli Sanchis
Computer Science Department
University Carlos III
Av. Universidad 30 Leganes (Madrid)
(34) 91624 9423
masm@inf.uc3m.es

ABSTRACT

In this work an improvement of an initial approach to design Artificial Neural Networks to forecast Time Series is tackled, and the automatic process to design Artificial Neural Networks is carried out by a Genetic Algorithm. A key issue for these kinds of approaches is what information is included in the chromosome that represents an Artificial Neural Network. In this approach new information will be included into the chromosome so it will be possible to compare these results with those obtained in a previous approach. There are two principal ideas to take into account: first, the chromosome contains information about parameters of the topology, architecture, learning parameters, etc. of the Artificial Neural Network, i.e. Direct Encoding Scheme; second, the chromosome contains the necessary information so that a constructive method gives rise to an Artificial Neural Network topology (or architecture), i.e. Indirect Encoding Scheme. The results for a Direct Encoding Scheme (in order to compare with Indirect Encoding Schemes developed in future works) to design Artificial Neural Networks to forecast Time Series are shown.

ACM Categories and Subject Descriptors

I.2.6 [Artificial Intelligence]: Learning - *Connectionism and neural nets;* G.3 [Mathematics of Computing]: Probability and Statistics - *Time series analysis;*

General Terms

Algorithms, Measurement, Performance, Design, Theory Verification, Experimentation.

Keywords

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

1. INTRODUCTION

In order to acquire knowledge, it is so interesting to know what the future will look like, i.e. forecast the future from the observed past. The forecasting task can be performed by several techniques as Statistical methods, and other based on Computational Intelligence like Immune Systems, and Artificial Neural Networks (ANN).

This contribution reports the methodology to carry out the automatic design of Artificial Neural Networks (ANN) that tackles the Forecasting Time Series problem taken from a referenced set of Time Series [1]. First Time Series is called Dow-Jones [2], second one is Temperature [3] and the last one is Passengers [4]. The task will consist of forecasting several Time Series, not all of them with the same ANN, but an automatic method will be used to obtain a different ANN to forecast each Time Series.

Two different steps will be done to get an ANN to forecast each Time Series. The first step will consist of setting the kind of ANN that will solve the forecasting task, and the learning algorithm used. According to [5], that show the approximation capability of Multilayer Perceptron (MLP), we have focused on full connected MLP with only one hidden layer and Backpropagation (BP) as learning algorithm to forecast time series as a first approach because ANNs with only one hidden layer are faster to train and easier to work with them.

In the second step the design of the ANN will be done setting the parameter values of the ANN. In the case of MLP with only one hidden layer and BP the parameters are: number of inputs nodes, number of hidden neurons (number of output neurons is placed by the problem), which is the connection pattern (how the nodes are connected), and as a new parameter, the whole set of connection weights. So, the process of designing ANN could be considered as a trial and error search problem within all possible designs, and this search could be done by Genetic Algorithms (GA). Different “pruning” and “growing” algorithms have been developed [3].

The paper is organized as follows. Sec 2 reviews the state of art about how to tackle forecast task with ANNs. Sec 3 explain how our system designs ANN with GA to forecast Time Series. In Sec 4 experimental setup and results are shown. And finally, conclusions and future works are described in Sec 5.

2. STATE OF ART

Several works have tackled the forecasting Time Series task with ANN, not only computer science researchers, but statistics as well [6]. This reveals the full consideration of ANN (as a data driven learning machine) into forecasting theory [7].

Before using an ANN to forecast, it has to be designed, i.e. establishing the suitable value for each freedom degree of the ANN [8] (kind of net, number of input nodes, number of outputs neurons, number of hidden layer, number of hidden neurons, the connections from one node to another , connection weights, etc). The design process is more an "art" based on test and error and the experience of human designer, than an algorithm. In [9] Zhang, Patuwo and Hu present a "state of the art" of ANN into forecasting task, in [10] is proposed an "extensive modeling approach" to review several designs of ANNs, and finally Crone and Preβmar show in [11] an evaluation framework to NN modeling in forecasting, and literature review.

2.1 Time Series and ANN

In order that a single ANN could work forecasting 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 obtaining 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 to obtain a function as it is shown in (1):

$$\mathbf{a_t = \mathit{f}\,(a_{t-1},a_{t-2},\ldots,a_{t-k}).} \qquad (1)$$

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:

(i) "k" inputs values, that correspond to "k" normalized previous values of period t: $a_{t-1},a_{t-2},\ldots,a_{t-k}$;

(ii) 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 70%) and the test set will be obtained from the rest of the complete patterns set.

If hand design of ANN is carried out, several topologies (i.e. different number of inputs nodes and number of hidden neurons in 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 forecasted values.

2.2 ANN and Evolutionary Computation

Several works approach the design of ANN using Evolutionary Techniques. Some of them use Direct Encoding Schemes (DES) [12,13], the others using Indirect Encoding Scheme (IES) [14,15,16]. For DES the chromosome contains information about parameters of the topology, architecture, learning parameters, etc. of the Artificial Neural Network. In IES the chromosome contains the necessary information so that a constructive method gives rise to an Artificial Neural Network topology (or architecture). Ajith Abraham [17] shows an automatic framework for optimization ANN in an adaptive way, and Xin Yao et. al. [18] try to spell out the future trends of the field.

3. ANN DESIGN WITH GA

The problem of designing ANN could be seen as a search problem into the space of all possible ANN. And that search can be done by a GA [19] using exploitation and exploration. Therefore there are three crucial issues: the solution's space (how each solution is codified into a chromosome), Encoding Scheme and what is looking for, translated into the fitness function.

In this first approach it has been chosen MLP as computational model due to its approximation capability, according to [5], and inside this group, Full Connected MLP with only a hidden layer because they are easier to encode into a chromosome and to work with them. The final object of our approach is to consider Sparsely Connected MLP to forecast Time Series and use both Indirect Encoding Schema, one based on Cellular Automata [20] and other based on Bidimensional Grammar [21], to design ANN using GA. A previous work for designing ANN to forecast using GA based on Cellular Automata is developed by Salah and Al-Salqan in [22].

But as a first approach to design ANN to forecast Time Series, a Direct Encoding Scheme for Full Connected MLP has been considered. Between all the different parameters that define an ANN (hidden layers, number of input nodes, number of output nodes, learning algorithm, algorithm to specify the initial connection weights values, etc), some of the are previously fixed (i.e. Back Propagation as learning algorithm) and others will be set during the search process. For this Direct Encoding Scheme the information placed into the chromosome is:

- Number of inputs nodes (*i*), corresponding to "k" (k=i) previous periods in (1) for each individual.
- Number of hidden nodes (*h*).
- Learning rate, for BP learning algorithm (α).
- Seed to initialize the weights of the MLP (s).

The value of learning rate "α" is between 0 and 1, and the value of "i" and "h" will be limited by a maximum number of inputs nodes (max_inputs) and a maximum number of hidden nodes (max_hidden) respectively.

One important characteristic for the performance of a MLP is the set of values known as connection weights. For each ANN the connection weights are obtained by means of a learning algorithm, in the case of MLP the algorithm is BP [8] The set of connection weights could be information to be included into the chromosome but in this case, the final chromosome would be to long and the search process carried out by the GA would be to slow and ineffective [23].

BP begins with a set of pseudorandom weights called initial weights, so always that it begins with the same set of initial weights, it will always arrive to the same set of final weights at the end of the training process. It means that one full connected MLP with the same topology and same seed value trained during "*n*" cycles will get the same connection weights values at the end of the process, each time that BP is carried out. Then, an approach to include the set of connections weights of the ANN into the chromosome will be done encoding the initialization seed "*s*" of the connection weights for each individual.

As it was said before values of "*i*" and "*h*" are 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 to a factor "A" (given by an expert as a parameter using partial autocorrelation function), and with the number of periods of the Time Series, it means, the known values of the Time Series (nts). See ec. (2)

$$\begin{aligned}&\mathbf{max\_inputs = A \times nts.}\\&\mathbf{max\_hidden = 2 \times max\_inputs.}\end{aligned} \qquad (2)$$

Into the chromosome, two decimal digits, i.e two genes, are used to codify the value "*i*", other two for "*h*", two for "*α*", and the last ten genes for "*s*" (seed in SNNS is of "long int type", that is why it has been used 10 genes to encode "*s*"). This way, the values of "*i*", "*h*", "*α*" and "*s*" are obtained, from the chromosome, as it can be seen in (3)

$$\begin{aligned}&\textbf{Chrom:}\\&g_{i1}\ g_{i2}\ g_{h1}\ g_{h2}\ g_{\alpha1}\ g_{\alpha2}\ g_{s1}\ g_{s2}\ g_{s3}\ g_{s4}\ g_{s5}\ g_{s6}\ g_{s7}\ g_{s8}\ g_{s9}\ g_{s10}\\&0 \le g_{xy} \le 9\ ;\ x = i,h,\alpha\ ;\ y=1..10\\&i = max\_inputs\ x\ ((g_{i1} \bullet 10 + g_{i2})/100)\\&h = max\_hiddens\ x\ ((g_{h1} \bullet 10 + g_{h2})/100)\\&\alpha = ((g_{\alpha1} \bullet 10 + g_{\alpha2}))/100\\&s = g_{s1}\ g_{s2}\ g_{s3}\ g_{s4}\ g_{s5}\ g_{s6}\ g_{s7}\ g_{s8}\ g_{s9}\ g_{s10}\end{aligned} \qquad (3)$$

Although there are sixteen possible genes into the chromosome, as it will be seen into the experimental results (section 4.2), two different codification of the chromosome will be used. One just composed of genes to encode "*i*", "*h*" and "*α*", and the other one with these six genes plus the ten genes used to encode "*s*".

The search process (GA) 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 maximum number of generations are reached. To obtain the fitness value of a chromosome:

1. The phenotypes (i.e. ANN) of all the individuals of the actual generation are obtained.

2. The train patterns and test patterns sets are obtained for each individual, depending on the number of inputs nodes of each net, as it was said above (sec.2).

3. Then, depending of which of the two different experiments are going to be done, the connection weights are randomly initialized (first experiment), or they are taken from the chromosome with the seed (second experiment).

4. Finally, the net is trained with BP (using Stuttgart Neural Network Simulator (SNNS) binary tools [24]) a maximum training cycles.

The fitness value will be the minimum test error reached during the learning 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 to forecast. Once that GA reaches the last generation, the best individual from all generations is used to forecast the future Time Series values (i.e. validation set).

4. EXPERIMENTAL SETUP AND RESULTS

4.1 Experimental Setup

To normalize the Time Series values (sec 2.2), so the nets will be able to work with them, it has to be considered the Time Series shape [25], especially trend (upward or downward) component. As the Time Series values have to be rescale, into the numerical range value [0,1], considering not only the known values, but the future values (those to be forecasted), the maximum and minimum limits for normalizing (max2norm, min2norm respectively) cannot be just the maximum (max) and minimum(min) known Time Series values. A margin from max and min has to be set if future values are higher or lower than they already are. This margin will depend on another parameter given by the expert (Prct_inc), in those cases in which the Time Series is stationary a Prct_inc of 10% will be enough, but when the Time Series is increasing or decreasing Prct_inc should be at least of 50%, because if it is going to be forecasted new values for a Time Series that in growing up, it is needed a margin enough big so the new values can be into the numerical range [0,1]. This Equation (4) show how are obtained max2norm and min2norm.

$$\begin{aligned}&\mathbf{max2norm = max + (Prct\_inc \times (max\text{-}min))}\\&\mathbf{min2norm = min - (Prct\_inc \times (max\text{-}min))}\end{aligned} \qquad (4)$$

Once that all the normalized Time Series values are obtained, the set of all the patterns is got as it was explained in sec. 2.1. After that the whole set of patterns will be split into train pattern set (as it was mentioned before first 70% of the whole pattern set, with it we will train the net during "m" (10^4) cycles), test pattern set (as it was mentioned before the rest 30% of the whole pattern set, with it we will be able to obtain the fitness value testing the net each "n" cycles). The validation pattern set (apart from train and test sets) will be the future values to be forecasted and used to evaluate nets obtained from GA results.

The parameters of GA described in sec 3 are: population size, 50; maximum number of generations, 100; percentage of the best individual that stay unchangeable to the next generation (percentage of elitism), 10%; crossover method; mutation probability, (1/length_chrom) = 0.17. The final goal of the search process is an ANN with the best generalization capacity (i.e. minimum test error) so it will be able to forecast better the validation pattern set values. Therefore, as it was said before, the fitness function is the minimum error test reached during the training process (sec 3).

4.2 Results of GA

As it was said before, three different Time Series will be used for the experiments. This set is composed of Dow-Jones, Temperature and Passengers. Passengers Time Series tends to grow up, Temperature and Dow-Jones are stationary (i.e. they have no trend).

The GA to design ANN has been carried out for this 3 selected Time Series. At the beginning it was only considered "i", "h" and "α" genes when encoding the topologies of the ANN into the chromosome, later it was also considered the ten genes of the initialization seed of the connection weights into the chromosome. After having run our system (i.e. GA) several times (10 times, although only two of them will be shown in the graphs due to a space problems) for each Time Series with and without seed, a graph is obtained for each one of them. It is the fitness evolution graph along the generations (i.e. the fitness value of the best individual from each generation). We can then observe a saw shape in the fitness evolution graph of the three Time Series when it is not encoded "s" into the chromosome. Dow-Jones Time Series is shown in Fig 1.

It can be said that the GA does not improve (i.e. it does not decrease) the fitness value along 100 generations when experiments without seed are launched for the three Time Series. This saw shape can be explained because a same chromosome (i.e. same topology) in generation "*i*" and generation "*i+1*" will have a different architecture due to that the connection weights are randomly initialized each generation by the system (i.e SNNS get a random seed at the beginning), so we can say this process of getting the fitness from genotype is non deterministic. Anyway, as it can be observed there is only a decreasing trend in the fitness value along the time the first ten generations, from here till the end it is stationary but still with the saw shape. That is why it was decided to include "*s*" (i.e. initialization seed of the connection weights) into the codification of the chromosome, so initialization seed of the connection weights will be kept generation after generation, eliminating the saw shape and obtaining a decreasing trend in the fitness evolution graph.

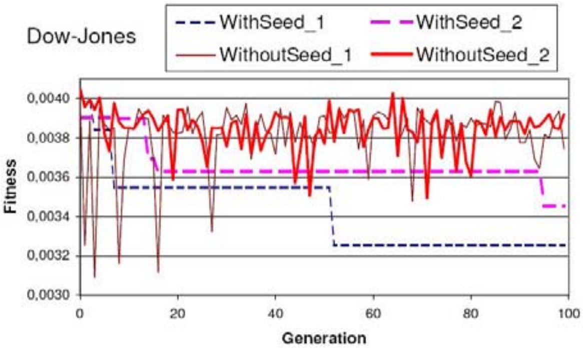

Figure 1 Dow-Jones Fitness graph with and without seed.

In Dow-Jones Time Series it can be also possible to check that not always the best individual taken from last generation of the experiment with seed (i.e. generation 99 with fitness 0,003254) has a better fitness value than the best individual that can be find in any generation of the experiment without seed (i.e. generation 3 with fitness 0,003091) encoded into the chromosome. That happens because looking for a good individual between a set of 50 individuals multiplied by 100 generations with only six genes and a set with the same number of individuals but including the seed (with sixteen genes into the chromosome) is easier for the first process to find a good one because the search space is smaller. If more generations were used to carry out the experiment, or a chromosome reduction was done the set of individuals that have the seed included would obtain a better result. Carrying out the same process for Temperature and Passengers these are the graphs obtained (Fig 2 & 3).

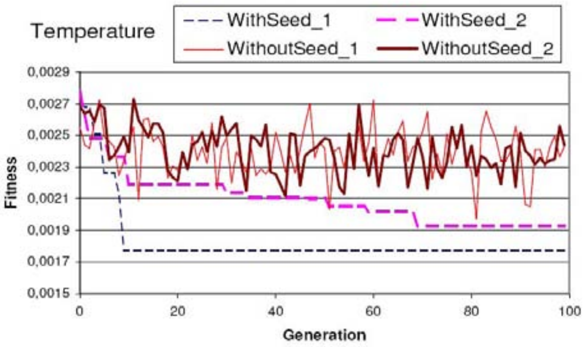

Figure 2 Temperature Fitness graph with and without seed.

For Passengers and Temperature it can be observed that those in which the seed has been encoded into the chromosome, the best individual from the last generation (i.e. generation 99 with fitness 0,000156 for Passengers and generation 99 with fitness 0,001772 for Temperature) obtains a better fitness value than any individual without seed along all generations (i.e. generation 75 with fitness 0,000162 for Passengers and generation 81 with fitness 0,001971 for Temperature).

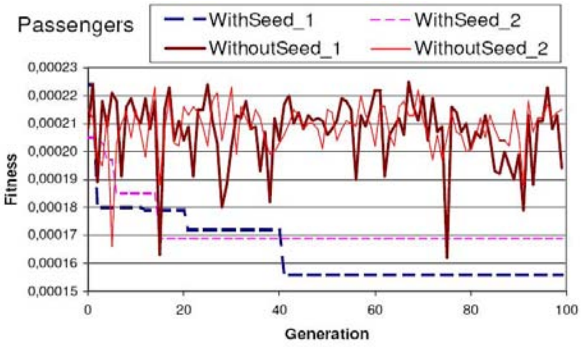

Figure 3 Passengers Fitness graph with and without seed.

Once that all processes have finished, and best individuals with seed and without seed (i.e ANN already trained) have been obtained for each Time Series, forecasting step can be done. As it was said before, due to a space problem, only two forecast with seed (in the graph called WS_X where X is a number) and other two carried out without seed (named WoS_X where X is a number) were done. These forecasts are carried out with the same best nets obtained from GA runs shown in Fig 1, Fig 2 and Fig 3.

A zoom of forecasted normalized values (i.e. forecasted values between [0, 1]) with the already known values also normalized (i.e. NTS) for Dow-Jones, Temperature and Passengers Time Series are shown together in Fig 4, Fig 5 and Fig 6. For Passengers and Temperature Time Series the forecasted values tend to follow the real values they should take. But for Dow Jones Time Series forecasted values resulting from only one of the 4 nets obtained from GA runs (WS_2) tend to follow real values (not close as Passenger and Temperature) of validation subset.

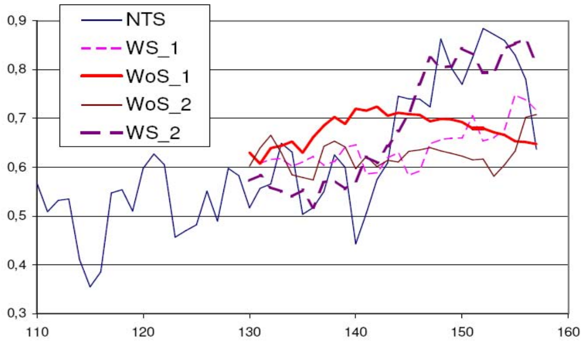

Figure 4 Dow-Jones forecasted values.

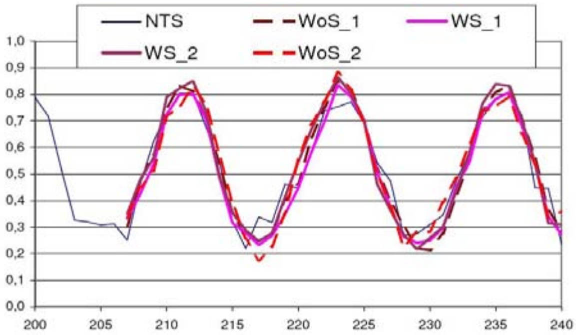

Figure 5 Temperature forecasted values.

Once that all the forecasted values are done it can be check how well this system works. This task will be carried out using MSE [26] and SMAPE error [27] for validation set, both of them will be calculated for the two best individuals with seed and without seed of each Time Series and we will be able to see which individual (i.e with or without seed) is better for each problem. Results of the four best topologies for Dow-Jones, Temperature and Passengers Times Series are shown in Table 1, Table 2 and Table 3 respectively. In these tables, first column will be used to represent the topology obtained, i.e. inputs_hidden_learningrate_seed, second column will represent the Fitness (test error value), third and fourth will be for MSE and SMAPE of forecasted values. The tables are ordered by the forth column value.

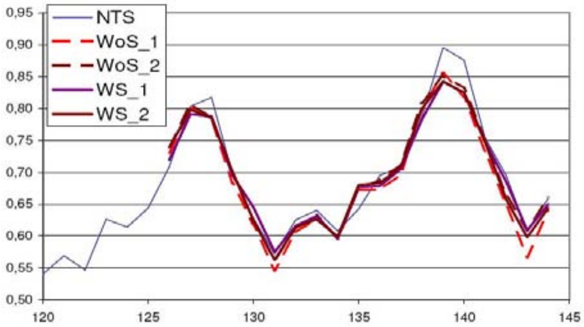

Figure 6 Passengers forecasted values.

Table 1.Dow-Jones MSE/SMAPE

Best ANN	Fitness	MSE	SMAPE
44_24_19_1653845331	0,003452	0,00499	9,0877203
45_71_04	0,003091	0,01968	13,478619
44_24_15_1241091545	0,002914	0,01411	13,615619
13_82_01	0,003492	0,01709	15,633814

Table 2.Temperature MSE/SMAPE

Best ANN	Fitness	MSE	SMAPE
62_47_31	0,002117	0,00277	8,642285
61_27_11_411555096	0,001929	0,00296	10,333577
66_86_60_1926843532	0,002067	0,00421	12,209635
80_53_72	0,002033	0,00545	15,966226

Table 3.Passengers MSE/SMAPE

Best ANN	Fitness	MSE	SMAPE
49_69_39_1588734993	0,000156	0,00046	2,0891843
50_29_40	0,000162	0,00046	2,1729044
49_62_58_570946223	0,000169	0,00059	2,5386781
50_44_71	0,000166	0,00075	3,3733351

New information can be gathered from those three tables. As it can be seen there is a correlation between MSE and SMAPE, but if Fitness value is compared with MSE and SMAPE, we check that not always the individual with the best Fitness value has also the minimum MSE and SMAPE error. It can be explained because when the forecast is done we are trying to obtain the future real unknown values with an ANN that has already been trained and test with a reduced number of data of the Time Series (i.e. Time Series data already known). But not always the best ANN obtained (i.e. ANN with best Fitness) will get the best forecast because it has the best generalization capacity for the known values but it doesn’t mean that it will also have the best generalization capacity for the future unknown values of the Time Series.

About Topologies, it can be also observed that for the three Time Series our system obtains different topologies as a solution, some of them with seed and others without seed. For Passenger Time Series, there is not a great difference of the number of inputs nodes among all ANN obtained, but having a look to the number of hidden nodes we can see that they have no relation (i.e. 69, 62, 29 and 44). In the case of Dow-Jones and Temperature solutions, ANN obtained by the system have different number of hidden nodes as used to happen to Passengers Time Series but if we pay attention to the number of input nodes those ANN obtained with a similar number of input nodes (i.e. 44, 44 and 45 for Dow-Jones and 61, 66 and 62 for Temperature) get a smaller SMAPE and MSE error than those which number of input nodes have no relation, for example in Dow-Jones Table the last ANN (i.e. 13_82_01) is a solution, but the worst of all of them.

So, in some Time Series forecasting including the initialization seed of the connection weights into the chromosome is a good idea to get a smaller forecasting error (i.e. Dow-Jones and Passengers), but in Temperature Time Series there is not too much difference between including or not the seed into the chromosome, because although smaller SMAPE error is obtained with an individual without seed (i.e. SMAPE 8,6422% with individual 62_47_31) second smaller SMAPE is got by an individual with seed (i.e. SMAPE 10,3333% with individual 61_27_11_411555096).

5. CONCLUSIONS AND FUTURE WORKS

Including the seed into the chromosome, that allows us to get the initial pseudorandom set of connection weights for each ANN, decrease the fitness value during the generations and finish with the “saw shape”. On the other hand, including the initialization seed of the connection weights into the chromosome not always get a better individual (i.e. individual with better Fitness value and smaller SMAPE error) than those ANN in which the seed is not included into the chromosome. That is why the system (i.e. GA) should be launched several times to get different solutions, some with seed and other without seed so they can be compared and use the best one to forecast the future values of the Time series.

To avoid the discovered problem in which individuals with better fitness do not carry out a better forecasting (i.e. there is not a direct correlation between fitness and SMAPE obtained for each ANN), in future works shuffling the patterns before train and test set are created (till the moment train pattern set is obtained from the first 70% of the total pattern set and test pattern set is the 30% remaining) could be an improvement.

In this new system the only pre-process that has to be done on the data is normalizing them, so it is easier for an inexperienced user to work with it than working with other systems in which more variables have to be set before it is carried out.

The results show that ANN has more difficulty predicting Dow-Jones time series than Temperature and Passengers, due to the greater chaotic component of Dow-Jones. An improvement, so Dow-Jones Time Series could be accurately forecasted, is to consider, not full connected nets, but Sparsely Connected MLP. For this kind of MLP design, both Indirect Encoding Scheme, one based on Cellular Automata [20] and other based on Bidimensional Grammar [21], will be studied. Besides, more parameters of ANN design, e.g. moment value for BP, activation function of artificial neuron, etc. could be include into the chromosome.

6. ACKNOWLEDGMENTS

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

7. REFERENCES

[1] www-personal.buseco.monash.edu.au/~hyndman/TSDL/

[2] K.W. Hipel and A.I. McLeod (1994), "Time Series Modelling of Water Resources and Environmental Systems" published by Elsevier, Amsterdam. ISBN 0-444-89270-2.

[3] Time Series Analysis and Forecasting: the Box-Jenkins Approach, 1976, Butterworth.

[4] Box, G. E. P., Jenkins, G. M. and Reinsel, G. C. (1976) Time Series Analysis, Forecasting and Control. Third Edition. Holden-Day. Series G.

[5] G. Cybenko. *Approximation by superposition of a sigmoidal function*. Mathematics of Control, Signals and Systems, 2, 303-314, 1989.

[6] Spyros G. Makridakis, Steven C. Wheelwright, Rob J Hyndman. *Forecasting: Methods and Applications.*

[7] Zhang, G., B. Eddy Patuwo, et al. (1998). "*Forecasting with artificial neural networks: The state of the art.*" International Journal of Forecasting 14(1): 35-62.

[8] Haykin, S. Simon & Schuster (ed.) *Neural Networks. A Comprehensive Foundatio*n Prentice Hall, 1999.

[9] Zhang, G.; Patuwo, B.E. & Hu, M.Y. *Forecasting with artificial neural networks: The state of the art International Journal of Forecasting*, 1998, 14, 35-62.

[10] Crone, S. F. *Stepwise Selection of Artificial Neural Networks Models for Time Series Prediction Journal of Intelligent Systems*, Department of Management Science Lancaster University Management School Lancaster, United Kingdom, 2005.

[11] Crone, S. F. and Preßmar, D. B. 2006. *An extended evaluation framework for neural network publications in sales forecasting*. In Proceedings of the 24th IASTED international Conference on Artificial intelligence and Applications (Innsbruck, Austria, February 13 - 16, 2006). V. Deved, Ed. International Association Of Science And Technology For Development. ACTA Press, Anaheim, CA, 179-186.

[12] 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.

[13] D.B. Fogel, Fogel L.J. and Porto V.W. *Evolving Neural Network, Biological Cybernetics*, 63, 487-493, 1990.

[14] 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.

[15] Yao, X. and Lin, Y. *A new evolutionary system for evolving artificial neural networks*, Transactions on Neural Networks, 8(3): 694-713, 1997.

[16] Kitano, H.: *Designing Neural Networks using Genetic Algorithms with Graph Generation System*, Complex Systems, 4, 461-476, 1990.

[17] Ajith Abraham, *Meta-Learning Evolutionary Artificial Neural Networks*, Neurocomputing Journal, Elsevier Science, Netherlands, Vol. 56c, pp. 1-38, 2004.

[18] H. Garis, D. Fogel, M. Conrad and X. Yao. *Special Issue on evolutionary neural systems*. Neurocomputing, 42, 1-8, 2002.

[19] Fogel, D. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Wiley-IEEE Press, 1998.

[20] G. Gutiérrez, A. Sanchis, P. Isasi, J.M. Molina and I. M. Galván. Non-direct Encoding method based on Cellular Automata to design Neural Network Architectures. Computer and Informatics, 24 2005, No3.

[21] Germán Gutiérrez, Beatriz García, José M. Molina and Araceli Sanchis. Studying the Capacity of Grammatical Encoding to Generate FNN Architectures. Computational Methods in Neural Modeling, Lecture Notes in Computer Science, Volume 2686/2003. IWANN 2003.

[22] Asma Abu Salah, Y.A. *Meta-Learning Evolutionary Artificial Neural Networks Using Cellular Configurations: Experimental Works Intelligent Computing*, 2006, Volume 4113/2006, pages 178-193.

[23] Arthur L. Corcoran and Roger L. Wainwright.“*Chromosome Reduction in Genetic Algorithms*”

[24] Prof. Dr. Andreas Zell,, WSI Computer Science Department, *Computer Arquitecture, Software, Artificial Neural Networks* http://www-ra.informatik.uni-tuebingen.de/SNNS/

[25] Valerie J. Easton & John H.McColl. *Statistics Glossary,Time series data*. http://www.stats.gla.ac.uk/steps/glossary/time_series.html

[26] http://www.appliedforecasting.com/index.php?option=com_content&task=view&id=401&Itemid=42

[27] 2006/07 Forecasting Competition for Neural Networks & Computational Intelligence. http://www.neural-forecasting-competition.com