Optimizating Multilayer Perceptrons for Time Series Prediction

Ben Romdhane Ons & Abdelhahed Trabelsi

February 13, 2003

Abstract

Selection of the current parameters of a multilayer perceptron and the right learning constants is a tedious task for designing an optimal multilayer perceptron, which is smaller and with a better generalization performance. In this paper we propose a multilayer perceptron training algorithm which is used for time series prediction and combines genetic algorithm and backpropagation.

The results from the application of the proposed algorithm to several time series benchmarks show the e¢ ciency of this algorithm in optimizing the neural model and its capacity to achieve good prediction with small multilayer perceptron.

Keywords : Multilayer Perceptrons, Genetic Algorithms, Time series Prediction, Learning, Generalization.

Introduction

Arti…cial neural networks have been widely used for time series prediction, but their drawback is the lack of a systematic procedure in model building, mainly since the performance of the neural network highly depends on its architecture and its parameters.

The most popular training mechanism is the backpropagation, in its di¤erent versions, which is frequently and successfully used to solve a great variety of real-world problems. The backpropagation is an iterative gradient descent algorithm designed to gradually minimize the di¤erence between the actual output vector of the network and the desired output vector. However, despite its popularity, the backpropagation is widely criticized for its ine¢ ciency. In fact, it does encounter two major di¢ culties : slow convergence and getting stuck in local minima. That's why some extensions have been suggested to improve the backpropagation algorithm especially the introduction of a momentum term and the utilization of a synchronized decreasing learning rate. However, all these versions are not able to really overcome the existing problems.

Recently, some investigations proposed the adaptation of genetic algorithms into neural network training towards synthesizing appropriate neural structures,

as well as to optimize the values of learning parameters. Indeed, the genetic algorithms o¤er an e¢cient search method for a complex problem space which contains many local optima by carrying out a global search over simultaneous regions of the error surface. That's why they are less likely to be trapped in a based local optima than gradient based search algorithms.

However, one major problem of genetic algorithms is their ine¢ciency in …ne-tuned local search although they are good at global search. In fact, the genetic algorithms can locate a good region in the space which contains a near optimal solution, but they have a di¢culty to converge to this solution. The e¢ciency of evolutionary training can be signi…cantly improved by incorporating a local search procedure into the evolution. The evolutionary algorithm will be used to locate a good region in the space and then a local search procedure is used to …nd a near optimal solution in this region. The local algorithm could be the backpropagation learning algorithm, which is used to modify the weights and the bias of the learning networks.

The application of the genetic algorithms to the neural networks means that each chromosome in the population codes for the di¤erent parameters related to a neural network, that will compete with other neural networks, so that the genetic algorithms will …nd the …ttest neural network for a given problem.

The topic of this paper is the question of how can genetic algorithms and neural networks exactly be combined. In other words, how should the neural nets be represented to get good time series prediction from the genetic algorithms. In this paper, we propose a new way of using the genetic algorithms in the optimization of the neural networks which are trained by the backpropagation.

Multilayer Perceptron Optimization by Genetic Algorithms

The representation

The aim of this paper is to present a multilayer perceptron training algorithm that tries to found the optimal set of weights and bias and the optimal architecture,i.e. the number of layers, the number of nodes for each layer and the transfer function to be used for each layer. Moreover, when we face time series prediction, we have to establish how to sample the past outcomes of the phenomenon which are used in the prediction. This means we must …nd a constant lag sampling called sampling lag that represents the lag between two consecutive inputs of the neural networks. In other words, it is the lag of the autocorrelation function of the time series.

This algorithm also tries to optimize the backpropagation parameters which are the training rate and the momentum coe¢cient that have an impact on the neural network performance.

In our approach, which is unlike other approaches that represent all the parameters related to the neural network in the same chromosome, we represent the set of weights and bias in a chromosome and the architecture with the sampling lag and the two backpropagation parameters in another one. In such a way, we decrease the probability to getting stuck in a local minima and we

increase the probability to …nd a near optimal neural network which provides a good time series prediction. Thus, each neural network is represented by two chromosomes (…gure 1 and …gure 2).

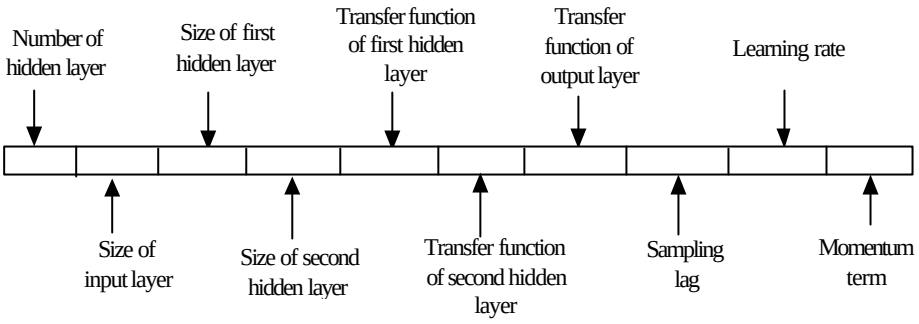

Figure 1: The graphical representation of the architecture chromosome

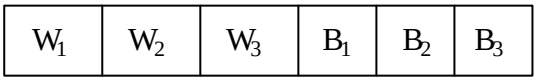

Figure 2: The graphical representation of the weights and bias chromosome.

Though such approach we suppose that all the neural networks are completely connected and all the hidden and output nodes have a bias. As for the number of output nodes it is initialized by the user in order to give the latter the possibility to choose the number of nodes that he needs. Indeed, this number describes how far in the future the neural network predict values.

The …tness function

Generally speaking in the multilayer perceptron training, the total time series is divided into two subsets : the training set which is used to train the network and the verifying set or the test set which is used to evaluate the generalization ability of the trained network.

To evaluate the goodness of an individual, we will use the Root Mean Square Error (RMSE) on the test set. We de…ne RMSE as :

$$RMSE = \frac{1}{N} \sqrt{\sum (predicted - actual)^2} \qquad (1)$$

The optimization algorithm

The proposed algorithm is speci…ed in the following pseudocode :

Step 1:
- Generate the initial population of architecture chromosomes (…gure 1).
- Choose the initial architecture, go to step 2.

Step 2 :
- If a weights-and-bias chromosome is a¤ected to each architecture in the population of architectures, go to step7. Else, create for the current architecture chromosome an initial population of weights and bias chromosome, go to step 4.

Step 3 :
- Train and evaluate the current architecture with all the choices of weights and bias using backpropagation algorithm, go to step 4.

Step 4:
- If the required error is found, a¤ect the best choice of weights and bias to the current architecture and stop the process (…rst termination criterion), else go to step 5.

Step 5:
- If the minimum number of generations of weights-and-bias chromosomes is reached, a¤ect the best choice of weights and bias to the current architecture and go to step 2 in order to repeat the same process to the next architecture. Else, create a new generation to try to pick up the best set of weights and bias for the current architecture, go to step 6.

Step 6 : Create a new generation of weights and bias chromosomes.
- Select the best n choices of weights and bias in the current population according to their …tness.
- Apply the genetic operators (crossover and mutation) on these choices, with the aim of creating new combinations of weights and bias, that will substitute the worst individuals in the current population in order to obtain the new one. The latter becomes the current generation, go to step 3.

Step 7 :
- If the maximum number of generations of architectures is reached return the best neural network and stop the process (second termination criterion). Else, go to step 8 to create a new generation of architectures.

Step 8 : Create a new generation of architecture chromosomes
- Select the best m architectures in the current population according to their …tness.
- Apply the genetic operators (crossover and mutation) on these architectures chromosomes with the aim of creating new architectures, that will substitute

the worst ones in the current population so as to obtain the new one. The latter becomes the current population of architectures.

- Choose an initial architecture in this population and go to step 2, to try to found the optimal set of weights and bias for each architecture.

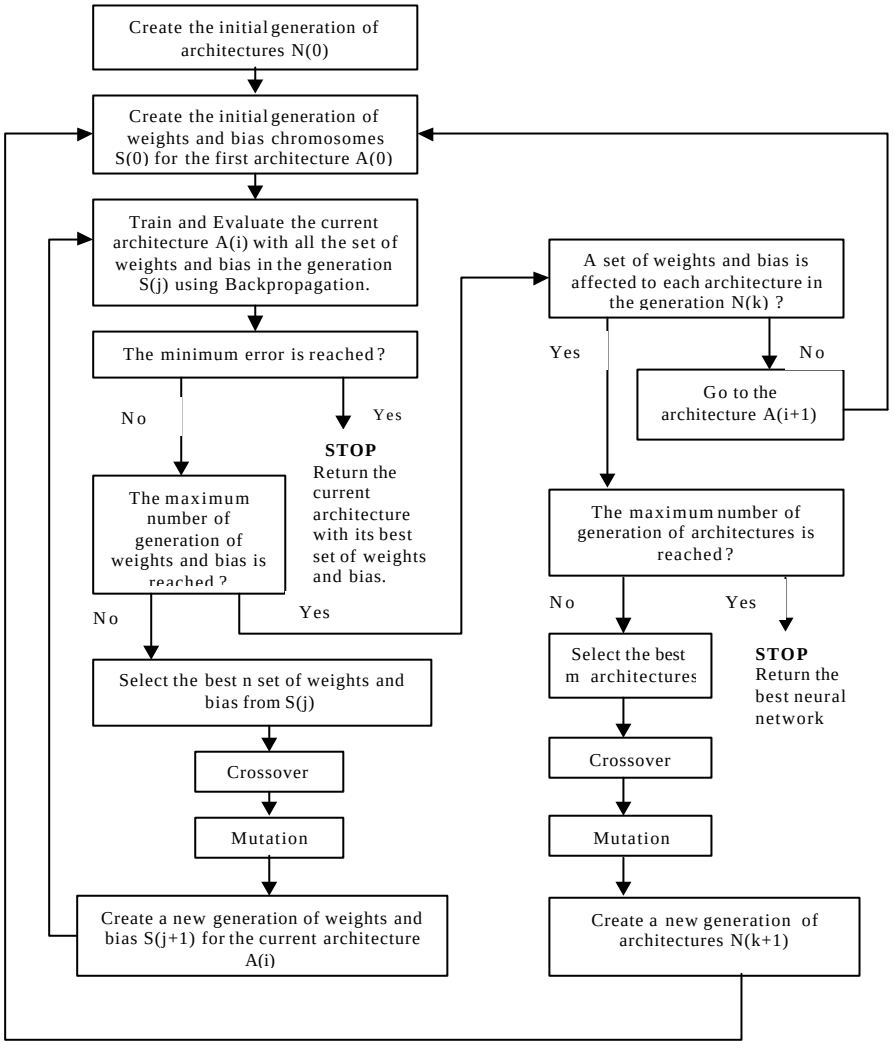

Figure 3: The different steps of the proposed algorithm.

Experiments and Results

To evaluate the performance of the proposed algorithm, we have used the following two time series Benchmarks [1].

a) Waste Water Flow Prediction

The problem is to predict the waste water ‡ow into a sewage plant. The water ‡ow was measured every hour. It is important to be able to predict the volume of ‡ow f(t+1) as the collecting tank has a limited capacity and a sudden increase in ‡ow will cause an excess in the water over‡ow. The water ‡ow prediction is to assist an adaptive on-line controller. The time series consists of 475 data points. The …rst 240 data sets were used for training and remaining data for testing.

b) Mackey Glass Chaotic Time Series

The Mackey Glass di¤erential equation [9] is a chaotic time series for some values of the parameters x(0) and ¿:

$$\frac{dx(t)}{dt} = \frac{0{:}2x(t - ¿)}{1 + x^{10}(t - ¿)} - 0{:}1x(t) \qquad (2)$$

Fourth order Runge-Kutta method was used to generate 1000 data series. The time step used in the method is 0.1 and initial condition were x(0) = 1:2; ¿(t) = 0 for t<0. First 500 data were used for training and remaining data for testing [ALEC].

The chromosomes are composed of the real values of the networks parameters and any codi…cation of the networks is needed. The parameters used in our experiments are set to be the same for both problems, and they are shown in table 1.

Parameters	Values
Maximum number of gen. of architectures	10
Size of the population of architectures	20
Maximum number of gen. of weights and bias	5
Size of the population of weights and bias	10
Number of input nodes	1 - 20
Number of nodes for each hidden layer	1 - 5
Number of output nodes	1
Activation Function	Tanh, Log-Sigmoidal
Initialization of weights and bias	+/- 0.5
Training rate	[1, 0.1]
Momentum term	[0.85, 0.95]
Crossover rate	0.8
Crossover kind	one-point
Mutation rate	0.06
Ranked based selection	0.5

Table 1 : Parameters used in the proposed algorithm.

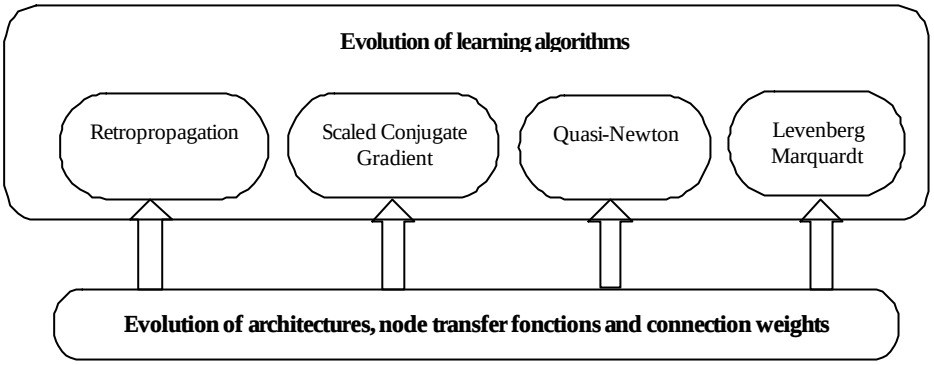

Figure 4: Interaction of various evolutionary search mechanisms in ALEC.

In our research, we are interested in comparing the proposed algorithm with another evolutionary algorithm which is based on the genetic algorithm to optimize multilayer perceptrons. This algorithm is ALEC [1].

ALEC (Adaptive Learning by evolutionary Computation) is an automatic computational framework for optimizing neural networks wherein an optimal design of neural network can only be achieved by the adaptive evolution of connection weights, architecture and learning rules which progress on different time scales. Figure 4 illustrates the general interaction mechanism with the learning algorithms of the neural network evolving at the highest level on the slowest time scale.

Table 2 shows the results obtained with the proposed algorithm and compares them to the ones shown in [1]. The experiments were repeated three times for each algorithm.

Time series	Proposed Algorithm		ALEC		
	RMSE	Arch.+		RMSE	Arch.+
Waste Water	0.00515	4(T++)	RP	0.0547	6(T),5(T),1(SL)
			SCG	0.0579	6(T), 4(SL)
			QN	0.0823	5(T), 5(T)
			LM	0.0521	8(T), 1(SL)
Mackey Glass	0.00203	4(T)	RP	0.0077	7(T), 3(LS)
			SCG	0.0031	11 (T)
			QN	0.0027	6(T), 4(T)
			LM	0.0004	8(T),2(T),1(LS)

Table 2 : Results of the proposed algorithm and ALEC

+ Arch. : hidden neurons distribution and activation functions

++ Activation function (T=tanh, TS=tanh-Sigmoidal)

The learning rate found by the proposed algorithm is 0.04982 and the momentum term is 0.9139. As for the sampling lag is 4.

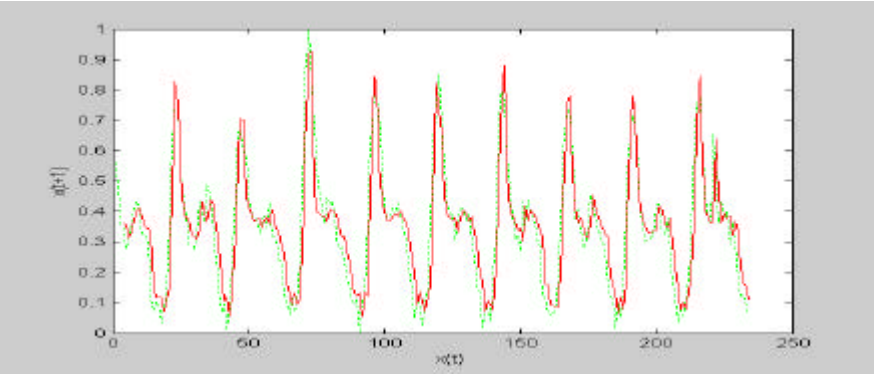

Figure 5: The proposed algorithm test results for Waste Water series:Predicted Values (...) and Desired Values (—). .

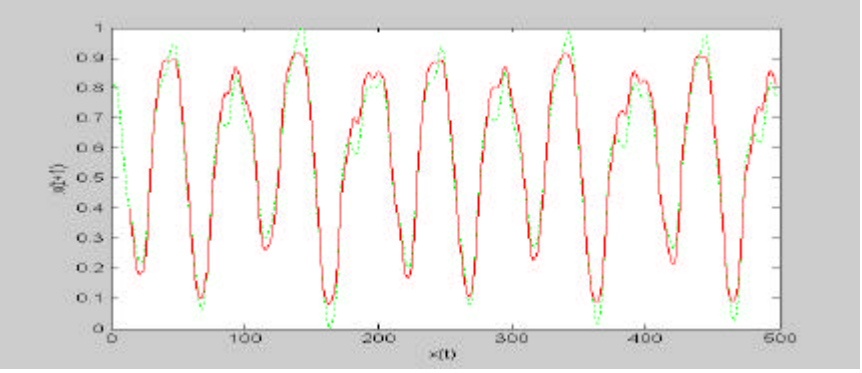

Figure 6: The proposed algorithm test results for Mackey-Glass series:Predicted Values (...) and Desired Values (—).

Discussion

Table 2 shows that the proposed algorithm outperforms ALEC, although the latter uses four learning algorithms. As for waste water ‡ow series, the proposed algorithm achieves an error of 0.00515 while the lowest error achieved by ALEC is 0.0521. As for the Mackey Glass series, the proposed algorithm obtains less error than the learning algorithms used in ALEC, except for the Levenberg-Marquardt algorithm. As it has been shown in both time series, the network size for the proposed algorithm is much smaller than that for ALEC.

To sum up, I would like to say that these experiments show that the proposed algorithm obtains, for di¤erent time series, less error and smaller networks than ALEC.

Application of the proposed algorithm to …nancial data sets

Experimentation data

The problem is to predict the opening currency of Tunis Air Stock. The quotation was measured every day from Monday to Friday and the data set consists of 1004 data points observed during four years stretching from 01/01/1998 until 31/12/2001. The …rst 502 data sets were used for training and remaining data for testing.

Results

We have used the same parameters of the proposed algorithm that were used in the previous experiments. Fitness value was calculated based on the RMSE achieved on the test set. The experiments were repeated three times and the worst RMSE value was reported (Table 3).

Architecture	RMSE
1:3:1	0.001613

Table 3: Performance of ANN trained using the proposed algorithm on the …nancial data set.

Figure 7: The proposed algorithm test results for the …nancial series:Predicted Values (...) and Desired Values (—).

Conclusion

In this paper an algorithm to automatic design of optimal neural networks models for time series prediction has been proposed. This algorithm, based on the genetic algorithms and the backpropagation, tries to simultaneously optimize the parameters of the multilayer perceptrons and the learning constants.

Several time series benchmarks have been used to test the proposed algorithm and to compare it with others [1]. The experimental results are promising

and show the importance and e¢cacy of the technique. The results show that this algorithm obtains smaller networks with less generalization error than others procedures.

In the proposed algorithm, we have supposed that all the neural networks are full connected and every neuron in the network has a bias. That's why in the future works we will focus on the optimization of the connectivity and the bias of the neural networks to reduce the dimension of the problem. We will concentrate also on comparing the proposed algorithm with other procedures for long-term predictions.

References

[1] Ajith Abraham, Baikunth Nath. ALEC : An Adaptive Learning Framework for Optimizing Arti…cial Neural Networks. Computational Science, Springer-Verlag. Germany, Vassil N Alexanrov et al (Editors, ISBN 3-540-42233-1, San Fransisco, USA, pp: 171-180,2002.

[2] Ajith Abraham, Baikunth Nath. Optimal Design of Neural Nets Using Hybrid Algorithmsp. 6$^{\text{ème}}$ Conférence Internationale d'Intelligence Arti…cielle, pp, 510-520, 2000.

[3] Ajith Abraham. Optimization of Evolutionary Neural Networks Using Hybrid Learning Algorithms. 2002 IEEE International Joint Conference on Neural Networks (IJCNN02), 2002 IEEE World Congress on Computational Intelligence, Hawaii, ISBN 0-7803-7278-6, volume 3, pp : 2797-2802, 2002.

[4] Antonio Fiordaliso. Une application des réseaux de neurones arti…ciels MLP à la prévision du prix d'une option négociable. Economie et prévision, n$^{\pm}$127; 1997.

[5] Gleb Beliakov and Ajith Abraham. Global Optimization of Neural Networks Using Deterministic Hybrid Approach. Hybrid Information systems, Studies in Fuzziness and Soft Computing, Springer Verlag, Germany, ISBN 3-7908-1408-6, pp : 79-92, 2002.

[6] Goldberg David.E. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Company, Inc, USA 1989.

[7] I.De Falco, A.Della Cioppa, A.Iazzetta, P.Natale and E.Tarantino. Optimizing Neural Networks for Time Series Prediction. Advances in Soft Computing Engineering Design and Manufacturing, 1998.

[8] Kate A.Smith, Jatinder N.D Gupta. Neural networks in Business : techniques and applications for the operations researcher. Computers & Operations Research, 1023-1044, 2000.

[9] Mackey MC, Glass L. Oscillation and Chaos in Physiological Control System. Science Vol 197, pp : 287-289, 1977.

[10] P.A.Castillo, J.J.Merelo, A.Prieto, V.Rivas, G.Romero. G-Prop : Global Optimization of Multilayer Perceptrons using GAs. Neurocomputing,

(2000), pp : 149-163.
[11] Perambur S.Neelakanta. Information-Theoretic Aspects of Neural Networks. CRC Press LLC, 1999.
[12] Philipp Koehn. Combining Genetic Algorithms and Neural Networks : The Encoding Problem. December 1994.
[13] Udo Seiffert. Multiple Layer Perceptron Training Using Genetic Algorithms. European Symposium on Artificial Neural Networks, Burges (Belgium), 25-27 April 2001.
[14] Xin Yao. Evolutionnary Artificial Neural Networks. Encyclopedia of Computer Science and Technology, Vol 33, pp : 137-170, New York, 1995.