COGNITIVE MODELING OF TIME DATA BY COMPUTATIONAL ALGORITHMS

RANDRIANAIVOARIVONY Rija Luc[1], ANDRIAMANOHISOA Hery Zo[2], ROBINSON Matio[3]

[1] Doctoral Student in School of Science and Engineering and Technic Innovation Laboratory of Cognitive Science and Application, High School Of Polytechnical University in Antananarivo, Madagascar,

[2] Professor, in the Doctoral School of Science and Engineering and Technic Innovation Laboratory of Cognitive Science and Application, High School Of Polytechnical University in Antananarivo, Madagascar

[3] Doctor in the Doctoral School of Science and Engineering and Technic Innovation Laboratory of Cognitive Science and Application, High School Of Polytechnical University in Antananarivo, Madagascar

ABSTRACT

This article proposes a cognitive approach based on the use of computational algorithms for modeling time data in particular on Optimization of a neural predictor based on a real genetic algorithm. This method consists in the simultaneous optimization of the topology of neural networks (the number of layers, the number of neurons per layer, the form of the activation function used in layers, the presence or absence of bias, the number of predictor inputs, delays), neural network control parameters (learning parameter, inertia parameter), and initial intervals of synaptic weights of hidden layers and output layer

Keywords : *Cognitive approach, neural networks, genetic algorithm*

1. INTRODUCTION

Time series forecasting is a very active subject in the field of science and engineering. The main purpose of time series analysis is to analyze, describe and explain a phenomenon over time and to draw consequences for responsiveness in decision-making. This forecast requires robust techniques and tools for their treatment [1].
Recently, bio-inspired computational algorithms such as artificial neural networks and genetic algorithms have been proposed as a promising alternative approach for time series prediction .

The methods of modeling by neural networks despite their simplicity, are less effective because e girls only issue not always the most relevant results. Many local optima do not always provide the optimal solution.

However, we will present the method of optimizing a neural predictor based on a real genetic algorithm. This method consists in the simultaneous optimization of the topology of neural networks (the number of layers, the number of neurons per layer, the form of the activation function used in layers, the presence or absence of bias, the number of predictor inputs, delays), neural network control parameters (learning parameter, inertia parameter), and initial synaptic weights of hidden layer and output layer.
In this work, we will simulate inflation in Madagascar as a time series by the hybridization method of the genetic algorithm and the neural network that we will compare with other methods.

2. APPROACH WITH COMPUTATIONAL ALGORITHMS

2.1. Neural network

The formal neuron is a mathematical model that incorporates the principles of the functioning of biological neuron, especially the sum of inputs. Knowing that at the biological level, the synapses do not all have the same "value" (connections between neurons being more or less strong), the researchers created an algorithm that weights the sum of its inputs by synaptic weights (weighting coefficients). In general, a formal neuron is a processing element having n inputs

$x_1, x_2, \dots, x_n$ (which are the external inputs or the outputs of the other neurons) and one or more outlets. Its treatment is to perform at its output the result of a threshold function$y_i f$ (Also called the activation function) of the weighted sum.

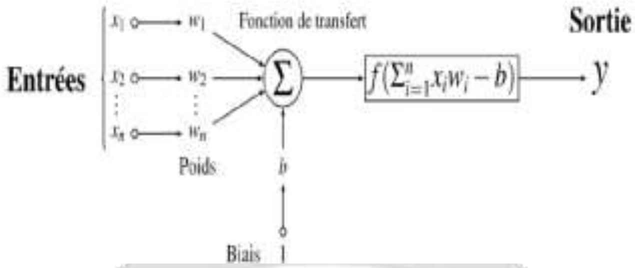

Fig-1: Formal neuron

An artificial neuron is composed of:

- A set of input values $x_1, x_2, \dots, x_n$
- A set of actual weights between neurons $w_1, w_2, \dots, w_n$
- A summing function P which calculates the weighted sum (weighted implied by weight) of the inputs: $(x_1 w_1) + (x_2 w_2) + \cdots + (x_n w_n)$
- An activation function that calculates the activity

A simple neural network consists of layers of neurons where each neuron of a layer is connected to all the neurons of the upper layer. The lower layer, called the "input layer" receives the data x_i, if the upper layer, called "output layer" gives the result or output of the neuron y. The intermediate layers are called "hidden layers" [2].

The activation function (or threshold function,) is used to introduce a non-linearity in the operation of the neuron. The functions thresholding generally have three intervals:

- below the threshold, the neuron is not active (often in this case, its output is 0 or - 1).
- around the threshold, a transition phase.
- above the threshold, the neuron is active (often in this case, its output is 1) [3]

The architecture a neural network is a network of several neurons, usually organized in layers.

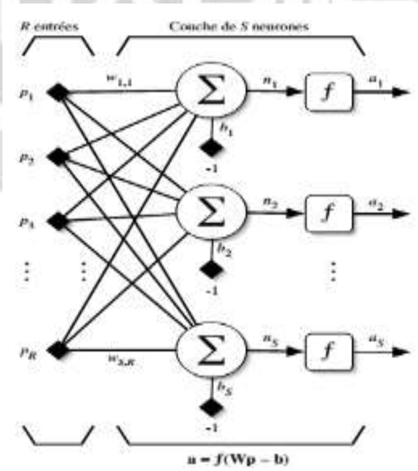

Fig-2: Layer of S neurons

The *S* neurons of the same layer are all connected to the *R* inputs. It is said that the layer is fully connected. A weight $w_{i,j}$ is associated with each of the connections. We will always note the first index by *i* and the second by *j*. The first index (row) always refers to the neuron number on the layer, while the second index (column) specifies the number of the entry. Thus, $w_{i,j}$ denotes the weight of the connection that connects the neuron *i* to its input *j*. The set of weights of a layer thus forms a matrix **W** of dimension $S \times R$:

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,1} \\ \vdots & \vdots & & \ddots \ \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix}$$

Finally, to build a network, it is only necessary to combine layers [4].

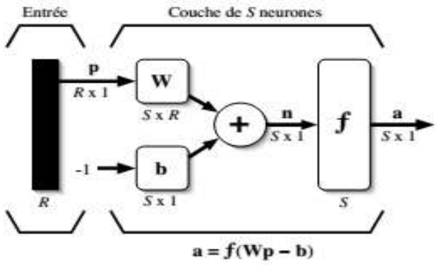

Fig-3: Matrix representation of a layer of S neurons

2.2. Genetic algorithm

The Genetic algorithms are based on solutions populations evolution simulation. The goal of genetic algorithms (GA) is to evolve a population P in order to find the optimum. To do this, at each generation t, the individuals of the population are mutated and crossed with a probability and it's the fittest who survive for the next generation.

This process is repeated for a number of generations, in the hope that the fitness function solutions will appear in the population [5].

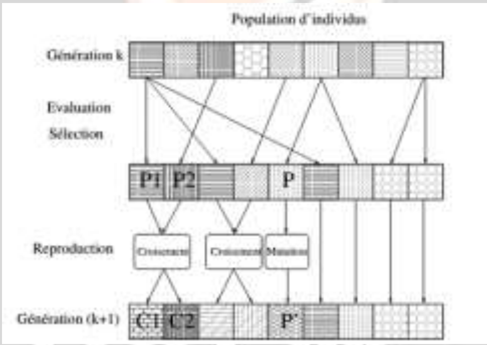

Fig-4: General principle of genetic algorithms

These algorithms are part of the class of so-called stochastic algorithms. Indeed, much of their operation is based on chance. Although using chance, GAs are not purely random. They effectively leverage the information obtained previously to speculate on the position of new points to explore, with the hope of improving performance[6].

The stochastic procedure used in a genetic algorithm is based on the following phases :

– Initialization: an initial population of N chromosomes is randomly drawn.
– Assessment: Each chromosome is decoded and evaluated.
– Selection: creation of a new population of N chromosomes by the use of an appropriate selection method.
– Reproduction: possibility of crossing and mutation within the new population
– Return to the evaluation phase until the algorithm stops [7].

3. APPROACH HYBRIDIZATION NEURONE NETWORK AND GENETIC ALGORITHM

The study of Artificial Neural Networks (ANN) and the study of Genetic Algorithms (AG's) have developed in parallel, they have often been in interaction during this last decade. The neural predictor is defined by a fairly large number of parameters and all these parameters can be optimized by the GA.

1. The weights of the connections are most often variables whose values are optimized by the GA : the range of variation synaptic weights of hidden layers s and s layer s output is one of the control parameters of the network complexity
2. The structure of prédicteur neuronal: the number of hidden layer , the numbers of neurons per layer, the shape of the activation function used in layers, the presence or absence of bias, the number of input predictor and delays (lag) are data that we can optimize by GAs .
3. The AG's can be used for optimization of neural predictor controls; it can be the learning coefficient, the coefficient of inertia, the number of learning steps. In our work, we are interested in the three simultaneous strategies that prove to be the most effective.

3.1. Representation of chromosome

It is proposed the use of genetic algorithms do not rely on the length of a predetermined chromosome. Using an actual coding of the variation interval of the synaptic weights of the neural network. This encoding is used with an objective function that measures the network efficiency by calculating a mean square error between the network output and the actual output. We consider a specific representation of parameters related to the synaptic weights, the neural predictor structure as well as the control parameters: the number of hidden layers (NHL), the numbers of neurons in each layer (*Nhl*) the form of the hidden layer and output layer activation function, the number of neurons of the input layer (N), the presence or absence of bias (b) the time between two inputs successive delay (δ), learning coefficient (η), inertia coefficient (α), synaptic weight variation range for hidden layers $[low_1 high_1]$, $[low_2 high_2]$ and output layer $[low_3 high_3]$.

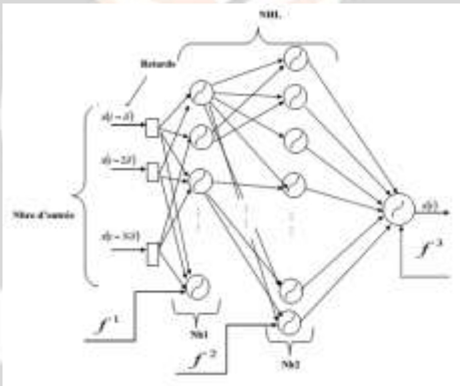

Fig-5: Representation of the parameters to optimize

Structure of the chromosome

Each allele in the chromosome is defined in a subset S_i, $i = \{1, \ldots, 7\}$presented in the following table:

	NHL	η	α	b	N	$Nh1$	$Nh2$	δ	f^1	f^2	f^{out}	$high_1$	low_1	$high_2$	low_2	$high_3$	low_3
Set	S_1	S_2	S_2	S_3	S_4	S_5	S_5	S_6	S_7	S_7	S_7	S_2	S_2	S_2	S_2	S_2	S_2

Table-1: The subset of each gene

Since it is sufficient in most neural network applications to use a small number of hidden layer, the value of (NHL) is taken between 1 and 2 while the maximum value for (*Nhl*) is calculated by equation:

$$max_{Nhl} = \frac{N+1}{2} + \sqrt{n\_data}$$

n_data : Number of data used in the learning phase

All alleles are defined in a subset.

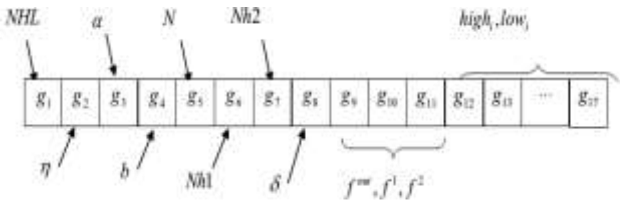

Table-2: Chromosome representation

$S_1 = \{1,2\}$,,,,,,.$S_1 = [0,1]$$S_3 = \{0,1\}$$S_4 = \{2,3,\ldots,30\}$$S_5 = \{3,4,\ldots,40\}$$S_6 = \{2,3,\ldots,10\}$$S_7 = \{1,2,3\}$
S_7 Means that the activation function can take the value 1≡ f_1 for sigmoid shape, 2 ≡ f_2 for the hyperbolic tangent form, 3 ≡ f_3 for the pure linear.
S_3 means 0 ≡ lack of bias, 1 ≡ the presence of bias
$[low_i high_i]$ have the range of variation of the synapse weight in both hidden layers and the layer production.

Genetic operators
The choice of GO use is inseparable from the choice of representation of RNA.

- **Selection**

Selection of a pair of individuals from the population. He has several types of selection in this work we used " *remainder selection* ".

- **crossing**

Crossing is the main operator acting on two chromosomes. It makes it possible to create new individuals by exchanging information between chromosomes through their combination. In our case, we used heuristic growth in the equation:

$$Child = parent_2 + R'(parent_1 - parent_2)$$

- **Mutation**

The role of the operator is to randomly modify the value of a gene in a chromosome. In the case of an real coding, the uniform mutation is used. Assuming the probability of mutation P_m, a draw of an x gene chromosome used to decide whether this gene is to be modified or not. We assume that the gene takes these values$[x_{min}, x_{max}]$ for the uniform mutation, which is a simple extension of the binary mutation, it is replaced by any selected x value x 'randomly chosen in the interval $[x_{min}, x_{max}]$

Purpose Function
It sets a goal, minimization of error $e(t)$ between the current value of time series and the predicted value. This can be defined by several numerical indices (NMSE, MAE, MBE, MSE).
In our work, we adopted for the minimizing of the mean square error (MSE).

$$fitness = MSE$$

APPLICATION

The growth in the Consumer Price Index (CPI) measures the monthly percentage change in the Consumer Price Index (CPI) to determine the rate of inflation. This rate reflects the change in the prices paid by the average consumer during a given period of time when purchasing goods and services. Obviously, the basket of goods and services on which the calculations are based changes over time due to changes in consumption habits. The Consumer Price Index must represent the cost of living through surveys and research that will paint a picture of the average consumer. According to the spending of this typical consumer, a basket of about 539 varieties depending on the type of products was created.
In our work, we will take the variation of the consumer price according to the following type products :

- product premises ;
- product semi-imported ;
- imported product .

For this study, we looked for and linked the monthly CPI series from November 2007 to December 2018, whose change in the consumer price is grouped by type of product (local, semi imported and imported products). According to the methodological summaries found in Razafimanantena (2003) and the monthly publication of INSTAT, the CPIs are calculated from more than 9000 records from 1400 points of sale distributed in the 6 former provincial capitals of Madagascar. .

3.2. Simulation of inflation with hybridization of genetic algorithm and neural network

In this simulation, each predictor consists principally of three layers (input layer, hidden layer and output layer). Neural predictors are developed through a real genetic algorithm and that by working to minimize the error of prediction. The genetic algorithm is used to optimize the architecture, control parameters and the range of weight variation.
The parameters of genetically selected predictors are:

- The number of hidden layers (NHL),
- the numbers of neurons in each layer $l(Nhl)$
- the activation function of the hidden layer and output layer ($f^l f^{out}$)
- the number of neurons in the input layer (N)
- the presence or absence of bias (b), the time delay between two successive input (δ)
- the learning coefficient (η)
- the coefficient of inertia (α)
- the range of variation of the synaptic weights for the hidden layers $[low_1, high_1]$, $[low_2, high_2]$and the output layer $[low_3, high_3]$.

For the learning phase 75% of the CPI values were used, 15% for the test phase and 15% for the test phase. The number of iteration of learning used is from 1000 to 10000.

The results obtained during the execution of AG 's relate to the following one

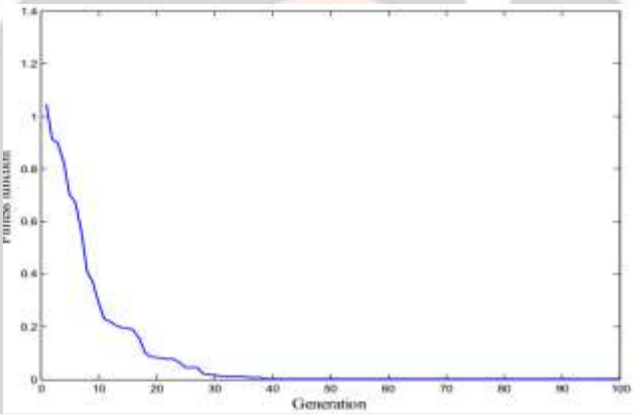

Fig-6: : Evolution of objective function across generations

The value of the ***fitness*** performance criterion is rapidly improved in the first generations, and is reduced from 1.0429 to only 50 generations for the prediction. The following **Tabe-3** gives the new neural predictor structure parameter corresponding to the best chromosomes obtained at the end of optimizations.

Nneural predictor parameter	Inflation
Number of hidden layer	1
Coefficient learning	0.9307
Coefficient of inertia	0.5457
Presence or through abscence	Presence (1)
Number of input	3
Delay	4
Number of neurons in layer cachée1	12
Form of hidden layer activation function	Semi-linear 3
Form of output layer activation function	Semi-linear 3
intout	[0.1634, 0.4703]
int1	[0.06360, 0.8137]

Table-3 : Optimal Structure received AG for inflation

The Fig-7 shows the evolution of MSE during the learning phase.

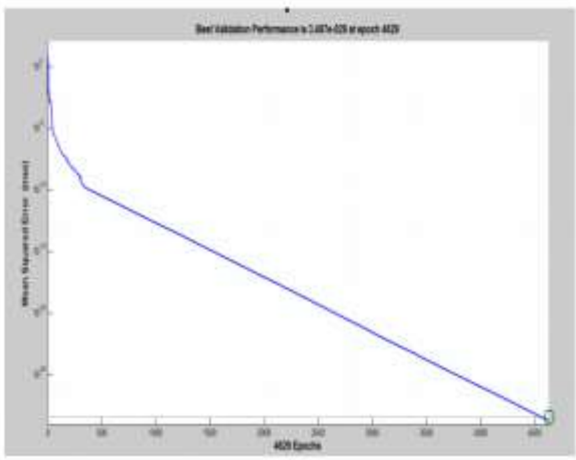

Fig-7 : Evolution of MSE during the learning phase

We note that the various results of iteration to another. The value of MSE improved during the first iterations, and reduced to3.487×10^{-29} after 4629 iterations for prediction

The Fig-8 shows the current and predicted CPA generated at the end of the learning phase and the Fig-9 shows the prediction error generated at the end of learning phase

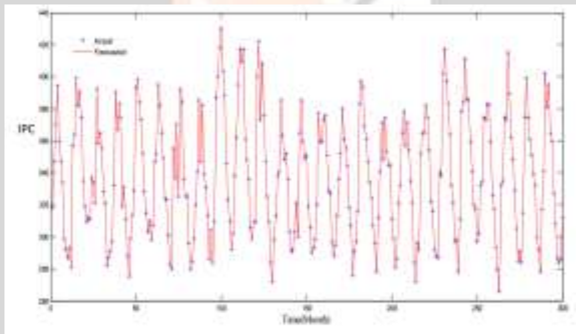

Fig-8: Current and predicted CPI generated at the end of the learning phase

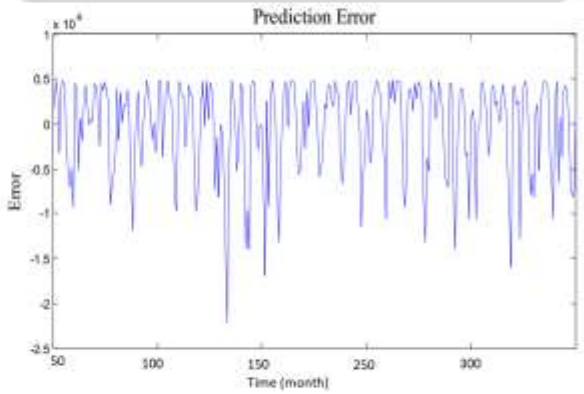

Fig-9: Prediction error generated at the end of the learning phase

We note that the prediction error is lower than $2{,}5 \times 10^{-8}$ for the CPI predictor and we find that the values of the prediction error are very low is these results are very satisfactory

The Fig-10 shows current monthly CPI and predicted the testing phase

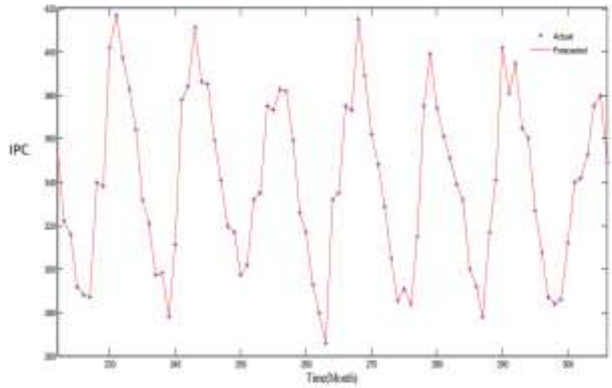

Fig-10: Current monthly CPI and predicts the test phase

3.3. Comparison with neural network

A comparative analysis of the results of the prediction of the consumer price index with RNA / AG hybridization and the RN will be shown in Table-4 according to the Mean Square Error (MSE).

Method	MSE
RN	0,321
RN / AG	3.487×10^{-29}

Table-4: Comparison of results

From the table above, it can be said that the proposed method produces ever weaker performance indicators. The reason for the predictive quality provided by this method is the efficiency of the GA in finding the best neural network parameters: the architecture, control parameters, and weight variation range that is appropriate with the values of time series to predict

4. CONCLUSION

A comparison of modeling with the neural network with the methodology of hybridization AG / RN shows that it allows a better model fit the data. A search path destined for a great future is based on unrestricted genetic algorithms, called "genetic programming", to search for functional forms best reproducing a data series.

A comparison of the modeling with the neural network with the AG / RN hybridization methodology shows that this allows a better fit of the model to the data. A search path destined for a great future is that based on unrestricted genetic algorithms, called " genetic programming ," to search for functional forms that best replicate a series of data.

Reference :

[1] A. Fiordaliso, "Fuzzy Systems and Prediction of Time Series", Science Publishing, 1999.
[2] PK Simpson, "Artificial Neural Systems" Pergmon Press Elmsford, New York, 1989.
[3] G. Dreyfur, "Neural Networks: Methodology and Application", Eyrolles edition, 2004.
[4] D. Martinez, "Statistical Learning: Neural networks, topological maps, vector machines", Edition Eyrolles, 2008
[5] DE Goldberg, "AG, Exploration, Optimization and Machine Learning", Addisionwesly 1994.
[6] N. Zerari, "Genetic Algorithms in Maintenance," El Hadj Lakhadar University, Batna. 2006.
[7] RL Haupt, SE Haupt, "Practical genetic algorithms", John Wiley, 2004.