IJCSI International Journal of Computer Science Issues, Volume 13, Issue 1, January 2016
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org

Architecture Optimization Model of Probabilistic Neural Network

Zakariae EN-NAIMANI [1] and Mohamed ETTAOUIL [2]

[1] Modeling and Scientific Computing Laboratory, Faculty of Science and Technology, USMBA Fez, MOROCCO

[2] Modeling and Scientific Computing Laboratory, Faculty of Science and Technology, USMBA Fez, MOROCCO

Abstract

Random Probabilistic neural networks are more approximate to humans than determinist neural network. Therefore, it is trivial in our study to use random criterion. There exist several random tools, but the most popular is the Probabilistic Self Organizing Maps. For that reason we chose this latter as a classification tool in this research paper, where we describe, in a first time, our PRSOM model as a MINLP model with linear constraints. And we use the dynamic center method to resolve this model. Then in a second time, we describe our PRSOM model as a MINLP model with nonlinear constraints, that we resolve with the genetic algorithm. In order to validate the theoretical approach, we apply our methods to the domain of classification. Moreover, the results obtained are compared with other classification methods.

Keywords: *Neural Random Network, self-organization map, classification, unsupervised learning, MINLP model.*

1. Introduction

Neural models are digital systems that allow general process modeling by establishing functional models. Graphically, a neural network is a set of interconnected neurons.

The Artificial Neural Networks (ANN) are a very powerful tool to deal with many applications, and they have proved their effectiveness in several research areas such as analysis and image compression, handwriting recognition, speech recognition [8,11], speech compression [10], video compression, signal analysis, facial recognition [23], process control, robotics and Web searching.

There exist two kinds of Artificial Neural Networks (ANN): Determinist NN [10][11][17], and probabilistic NN [9][19]. In this paper, we focus especially on the Probabilistic Neural Networks.

The probabilistic self-organizing map (PRSOM) [2] uses a probabilistic formalism. This algorithm approximates the maximum density distribution of the data thanks to the learning phase of the PRSOM. Thus, we deduce that the learning stage is very important in the probabilistic Self-Organizing Maps (PRSOM) performance.

The neural models are now part of the optimization domain and are applied in retro propagation [12], in Kohonen maps models [17][18][20], etc.

The optimization is a branch of mathematics. In practice, we start from a concrete problem, we model it, and then we resolve it. An optimization problem consists on finding the optimal value associated to an optimal solution in a domain D, given the function f. The optimization area includes various types, like: linear optimization and nonlinear optimization…

In the present work, in the first step we present a new PRSOM algorithm based on a MINLP model with linear constraints [14], resolved by the Cluster center method (assignment-minimization). In the second, we propose a new model for architecture optimization based on a MINLP with linear and nonlinear constraints (quadratic). This latter is resolved by the genetic algorithm [16].

We recall some definitions and theoretical results on which this paper is based. We begin by describing the general formulation of Mathematical Optimization Problem (MOP):

$$\begin{cases} Min \quad f(x) \\ s.t. \quad g_i(x) \le b_i \quad \forall i \in \{1,\dots,m\} \\ \qquad x \in D \subset \mathbb{R}^n \end{cases} \tag{1}$$

Where f is the objective function and $g_i : i \in \{1,\dots,m\}$ are the m constraints, $S = \{x \in D \subset \mathbb{R}^n : g_i(x) \le b_i, i = 1,\dots,m\}$ is the feasible region.

We will focus on the Mixed Integer Non Linear Programming; this latter represents a powerful framework for mathematical modeling of many optimization problems that involve discrete and continuous variables. MINLP is a NP-complete problem which has been considered as a very complicated problem until now.

The MINLP formulation is stated as:

$$\begin{cases} Min \quad f(x,y) \\ s.t. \quad \mathrm{g}_i(x,y) \le 0 \quad \forall i \in 1,\dots,m \\ \qquad x \in D \subset \mathbb{R}^n \quad y \in E \subset \mathbb{N}^p \end{cases} \tag{2}$$

f, g_i are respectively nonlinear objective function and constraints, x is a n vector of continuous variables and y is a p-vector of integer variables.

The organization of this paper is as following: the section 2 presents the formalism of Probabilistic Self-Organizing Map. In section 3 we introduce the proposed mathematical model to Probabilistic Self-Organizing Maps based on MINLP with linear constraints. In section 4, we propose the model of architecture optimization based on a MINLP with linear and nonlinear constraints and the resolution of this model. And before concluding, experimental results are given in the section 5.

2. Probabilistic Self-Organizing Map

In this section, we will introduce the formal PRSOM model. In the probabilistic formalism, the classical map C of SOM [3][22] is duplicated into two similar maps C^1 and C^2 provided with the same topology as C, for every input data $x \in D \subset \mathbb{R}^d$ and every pair of neurons $(c_i^1, c_j^2) \in C^1 \times C^2$,which associates to each neuron c_k a Gaussian density function f_k [7,13], which is defined by its mean $w_k \in \mathbb{R}^d$ and its covariance matrix Σ_k.
Compute the probability of any pattern x with the joint probability $p(x) = \sum_{j=1}^{K} p(c_j^2) p_{c_j^2}(x)$.

Where K is the number of neurons for the two maps C^1 and C^2,

$$\begin{aligned} p_{c_j^2}(x) &= p(x / c_j^2) = \sum_{i=1}^{K} p(c_i^1 / c_j^2) p(x / c_i^1), \\ p(c_j^2 / x, c_i^1) &= p(c_j^2 / c_i^1) = \frac{K_T(d(c_j^2, c_i^1))}{\sum_{i=1}^{K} K_T(d(c_j^2, c_i^1))} \end{aligned} \tag{3}$$

And

$$p(x / c_i^1, c_j^2) = p(x / c_i^1) = f_{c_i^1}(x, w_{c_i^1}, \Sigma_{c_i^1}) \tag{4}$$

Where $f_{c_i^1}$ is the i^{th} Gaussian density with mean vector $w_{c_i^1}$ and covariance matrix $\Sigma_{c_i^1} = \sigma_{c_i^1}{}^2 \mathrm{I}$.

Then

$$p(x) = \sum_{j=1}^{K} p(c_j^2) \sum_{i=1}^{K} \frac{K_T(d(c_j^2, c_i^1))}{\sum_{k=1}^{K} K_T(d(c_j^2, c_k^1))} f_{c_i^1}(x, w_{c_i^1}, \Sigma_{c_i^1}) \tag{5}$$

The curve of this likelihood has a very complicated shape, which often has very numerous local maxima. Practically, it is impossible to maximize directly this likelihood, even to reach a local maximum [7]. The algorithm presented in [7] ensures the convergence into a local maximum of data probability.

3. Proposed Mathematical Model of PRSOM

3.1 Modeling of PRSOM via MINLP

We propose a new model of probabilistic self-organizing maps as an optimization problem in terms of a mixed-integer nonlinear problem with linear constraints. To formulate this model we need to define some parameters as follows:

Parameters:

n : number of data set observation;
N: Number of neurons in the topology map of PRSOM;
d : Dimension vector of data set observation;

Variables:

$X = (x_{ij})_{\substack{1 \le i \le n \\ 1 \le j \le d}}$: Matrix of Training base elements.

$U = (\mathrm{u}_{ij})_{\substack{1 \le i \le n \\ 1 \le j \le N}}$: Matrix of the binary variables.

$W = (\mathrm{w}_{ij})_{\substack{1 \le i \le N \\ 1 \le j \le d}}$: Matrix of referent vectors.

$\sigma = (\sigma_i)_{1 \le i \le N}$: Matrix of covariance.

With $u_{i,j} = \begin{cases} 1 & \text{if the } i^{th} \text{ data assigned to } j^{th} \text{ neuron} \\ 0 & \text{else} \end{cases}$, $u_{i,j}$ is the assignment variable that define the relationship between data and neuron.

Objective function:

Basing on the work of Bishop [4], we will define the objective function of the PRSOM mathematical model as:

$$\text{Max } p(U,W,\sigma) = \prod_{i=1}^{n}\prod_{j=1}^{N}(\pi_j * (\sum_{k=1}^{N} K^T(\delta(j,k))\theta_k(x_i, w_k, \sigma_k)))^{u_{ij}} \quad (6)$$

For reasons of convenience, the log level helps to reduce the volume of the digits representing a series. Moreover, the linear logarithm is a multiplicative relationship i.e. we transform a multiplicative series to an additive one. The log function is strictly increasing. It is then better to maximize log(p) than p.

The objective function becomes:

$$Max \ln(p(U,W,\sigma)) = \sum_{i=1}^{n}\sum_{j=1}^{N} u_{ij}[\ln(\pi_j) + \ln(\sum_{k=1}^{N} K^T(\delta(j,k))\theta_k(x_i, w_k, \sigma_k))] \quad (7)$$

Constraints:

Each data element must be allocated to one neuron (component). In consequence we obtain the following n constraints: $\sum_{j=1}^{N} u_{ij} = 1 \quad \forall i \in 1,....,n$

PRSOM model:

A general formulation for the (MINLP) is given by (P_{Max}).

$$(P_{Max}) = \begin{cases} Max \ln(p(U,W,\sigma)) = \\ \sum_{i=1}^{n}\sum_{j=1}^{N} u_{ij}[\ln(\pi_j) + \ln(\sum_{k=1}^{N} K^T(\delta(j,k))\theta_k(x_i, w_k, \sigma_k))] \\ Subject\ to: \\ \sum_{j=1}^{N} u_{ij} = 1; ...; 1 \le i \le n \\ U \in \{0,1\}^{n\times N} \\ W \in \mathbb{R}^{N\times d} \\ \sigma \in \mathbb{R}^{N} \end{cases} \quad (8)$$

The research for a maximum can always be transformed to the research of a minimum, the mathematical model is thus as following:

$$(P_{Min}) = \begin{cases} Min\ E(U,W,\sigma) = \\ -[\sum_{i=1}^{n}\sum_{j=1}^{N} u_{ij}[\ln(\pi_j) + \ln(\sum_{k=1}^{N} K^T(\delta(j,k))\theta_k(x_i, w_k, \sigma_k))]] \\ Subject\ to: \\ \sum_{j=1}^{N} u_{ij} = 1; ...; 1 \le i \le n \\ U \in \{0,1\}^{n\times N} \\ W \in \mathbb{R}^{N\times d} \\ \sigma \in \mathbb{R}^{N} \end{cases} \quad (9)$$

In the following section, we study the resolution of the last mathematical program Eq.(9).

3.2 Resolution of the obtained mixed-integer nonlinear problem

We use the dynamic clusters approach to solve this mathematical model. We will solve it basing on two steps:

- Assignment phase: we fix the weight vectors and we solve the obtained problem;
- Minimization phase: we fix the assignment vectors and we solve the obtained problem.

Assignment phase:

If we fix the variables W and σ in (P_{Min}), we find a linear model of binary variables under linear constraints.

The obtained model $(P_{W,\sigma})$ is defined by:

$$(P_{W,\sigma}) = \begin{cases} Min\ E_{W,\sigma}(U) = \\ -\sum_{i=1}^{n}\sum_{j=1}^{N} u_{ij} \ln[\pi_j \sum_{k=1}^{N} K^T(\delta(j,k))\theta_k(x_i, w_k, \sigma_k)] \\ Subject\ to: \\ \sum_{j=1}^{N} u_{ij} = 1; ...; 1 \le i \le n \\ U \in \{0,1\}^{n\times N} \end{cases} \quad (10)$$

The matrix U can be transformed into a vector X of size m, with $m = n \times N$

$$X = \begin{matrix} u_{1,1} & u_{1,2} & \cdots & u_{1,N} & \cdots & u_{i,1} & \cdots & u_{i,j} & \cdots & u_{i,N} & \cdots & u_{n1} & \cdots & u_{nN} \end{matrix}$$

Afterwards we can define the objective function as follows:

$E(\mathrm{X}) = C^t\,\mathrm{X}$

With:

$$C = \begin{pmatrix} -\ln[\pi_1 \sum_{k=1}^{N} K^T(\delta(1,k))\theta_k(\mathrm{x}_1, w_k, \sigma_k)] \\ -\ln[\pi_2 \sum_{k=1}^{N} K^T(\delta(2,k))\theta_k(\mathrm{x}_1, w_k, \sigma_k)] \\ \vdots \\ -\ln[\pi_N \sum_{k=1}^{N} K^T(\delta(N,k))\theta_k(\mathrm{x}_1, w_k, \sigma_k)] \\ \vdots \\ -\ln[\pi_1 \sum_{k=1}^{N} K^T(\delta(1,k))\theta_k(\mathrm{x}_i, w_k, \sigma_k)] \\ \vdots \\ -\ln[\pi_N \sum_{k=1}^{N} K^T(\delta(N,k))\theta_k(\mathrm{x}_i, w_k, \sigma_k)] \\ \vdots \\ -\ln[\pi_1 \sum_{k=1}^{N} K^T(\delta(1,k))\theta_k(\mathrm{x}_n, w_k, \sigma_k)] \\ \vdots \\ -\ln[\pi_N \sum_{k=1}^{N} K^T(\delta(N,k))\theta_k(\mathrm{x}_n, w_k, \sigma_k)] \end{pmatrix}$$

Linear constraints associated with this problem are defined by the following statement:

Each element $x_i \; ; i = 1,\dots,n$ is affected to a single neuron j. These constraints are given by:

$$\sum_{j=1}^{N} u_{ij} = 1;\dots;1 \le i \le n \Leftrightarrow AX = b$$

The matrix $A \in \{0,1\}^{n\times N}$ and the vector $b \in \mathbb{R}^m$ are defined by:

$$A = \begin{pmatrix} 1 & \cdots & 1 & 0 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \ddots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & \cdots 0 & 1 & \cdots & 1 \end{pmatrix} \qquad b = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

Finally we obtain a linear program with variables 0-1, and with linear constraints.

$$(P_{W,\sigma}) = \begin{cases} Min\ E(\mathrm{X}) = < C, \mathrm{X} > \\ Subject\ to: \\ \mathrm{AX} = \mathrm{b} \\ X \in \{0,1\}^{nN} \end{cases} \tag{11}$$

U* is optimal solution of the model $(P_{W,\sigma})$.

Minimization phase:

In this step, we fix the variables vector u, and we solve the following optimization problem with continuous variables:

$$(P_U) = \begin{cases} Min\ E_U(W,\sigma) = \\ -[\sum_{i=1}^{n}\sum_{j=1}^{N} \mathrm{u}_{ij}[\ln(\pi_j) + \ln(\sum_{k=1}^{N} K^T(\delta(j,k))\theta_k(\mathrm{x}_i, w_k, \sigma_k))]] \\ W \in \mathbb{R}^{N\times d} \end{cases} \tag{12}$$

The solution of the problem (P_U) is given by the following system: $\frac{\partial E_U}{\partial w_k} = 0$ *and* $\frac{\partial E_U}{\partial \sigma_k} = 0$, since it is sufficient to ensure, that in every iteration, we use a simple gradient method.

$$w_k = \frac{\sum_{i=1}^{n}\sum_{j=1}^{N} u_{ij} x_i \frac{K^T(\delta(j,k))\theta_k(\mathrm{x}_i)}{\sum_{r=1}^{N} K^T(\delta(j,\mathrm{r}))\theta_r(\mathrm{x}_i)}}{\sum_{i=1}^{n}\sum_{j=1}^{N} u_{ij} \frac{K^T(\delta(j,k))\theta_k(\mathrm{x}_i)}{\sum_{r=1}^{N} K^T(\delta(j,\mathrm{r}))\theta_r(\mathrm{x}_i)}} \tag{13}$$

$$\sigma_k^2 = \frac{\sum_{i=1}^{n}\sum_{j=1}^{N} u_{ij} \frac{\| w_k - x_i \|^2 K^T(\delta(j,k))\theta_k(\mathrm{x}_i)}{\sum_{r=1}^{N} K^T(\delta(j,\mathrm{r}))\theta_r(\mathrm{x}_i)}}{d \times \sum_{i=1}^{n}\sum_{j=1}^{N} u_{ij} \frac{K^T(\delta(j,k))\theta_k(\mathrm{x}_i)}{\sum_{r=1}^{N} K^T(\delta(j,\mathrm{r}))\theta_r(\mathrm{x}_i)}} \tag{14}$$

3.3 Proposed New algorithm PRSOM based on the resolution of MINLP

This algorithm is probabilistic self-organizing based on solving the optimization problem (P_{Min}).

Algorithm 1:

Input:

n, p, X, N_{iter}, N, $[T_{\min}, T_{\max}]$ the interval of the parameter T;

Output:

Optimal probabilistic topological map

Initialization:

$w^1(0),...,w^N(0)$ randomly initialized, $\sigma^1(0),...,\sigma^N(0)$ randomly initialized with the great values, $T \leftarrow T_{\max}$ $t \leftarrow 0$

While $t < N_{iter}$

Assignment-decision phase via resolution of the model $(P_{W,\sigma})$.

While $k < N$

Minimization phase via resolution of the model (P_U); update the w_k via the Eq.(13) and update the σ_k via the equation Eq.(14).

Done $t \leftarrow t+1$;

Done $T \leftarrow T_{\max}(\frac{T_{\min}}{T_{\max}})^{\frac{t}{N_{iter}-1}}$;

Return:

Optimal parameters of PRSOM.

Unfortunately, even after the mathematical modeling of the PRSOM in the previous section, the major problem of this latter is the choice of the architecture, i.e. the initial choice of the model's parameters. For that reason we propose in the next section a mathematical model to resolve this problem.

4. Proposed model to optimization of the architecture of PRSOM

4.1 Mathematical model

Generally, if the size of the probabilistic self-organizing map is chosen randomly, the PRSOM learning algorithm gives two classes of neurons as showing in Fig. 1, the first class that doesn't represent any observation (empty class), and the second class that represents the important information data. The mean purpose is to delete the useless components from the PRSOM.

Basing on the previous mathematical model, we add the control variable v_j which allows controlling the size of the PRSOM map, with $v_j = \begin{cases} 1 & \textit{if the } j^{th} \textit{ neuron is used} \\ 0 & \textit{else} \end{cases}$

Objective function:

$$\text{Max } \; p(\text{V},U,W,\sigma) = \prod_{i=1}^{n}\prod_{j=1}^{N}(\pi_j * (\sum_{k=1}^{N} K^T(\delta(j,k))\theta_k(\text{x}_i, w_k, \sigma_k)))^{u_{ij}\times v_j} \tag{15}$$

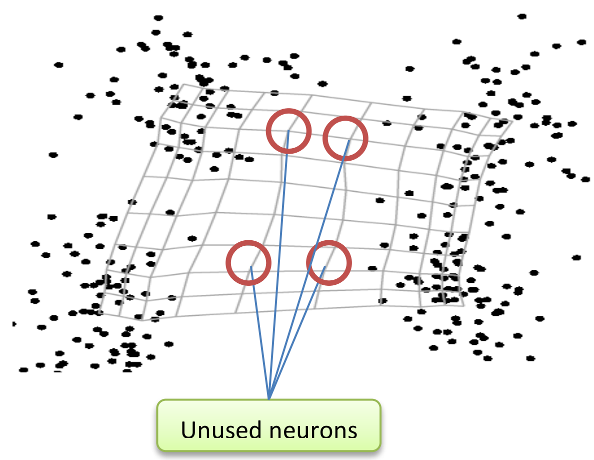

Fig. 1 Illustration of two classes' neurons of PRSOM.

As well the function becomes:

$$Max \ln(p(\text{V},U,W,\sigma)) = \sum_{i=1}^{n}\sum_{j=1}^{N} v_j u_{ij}[\ln(\pi_j) + \ln(\sum_{k=1}^{N} K^T(\delta(j,k))\theta_k(\text{x}_i, w_k, \sigma_k))] \tag{16}$$

Constraints:

Besides assignment constraints, we add another one called transmission constraint.

If the neuron j is not used $v_j = 0$, i.e., the summation on i of u_{ij} takes 0; else $v_j = 1$, i.e. the summation on i of u_{ij} is strictly greater than 0 then the constraint is: $\sum_{j=1}^{N}(1-v_j)\sum_{i=1}^{n}u_{i,j} = 0$

Mathematical model:

$$(P_{Min}) = \begin{cases} Min\ E(U,\mathrm{V},W,\sigma) = \\ -[\sum_{i=1}^{n}\sum_{j=1}^{N} v_j u_{ij}[\ln(\pi_j) + \ln(\sum_{k=1}^{N} K^T(\delta(j,k))\theta_k(\mathrm{x}_i, w_k, \sigma_k))]] \\ Subject\ to: \\ \sum_{j=1}^{N} u_{ij} = 1; ...; 1 \le i \le n \\ \sum_{j=1}^{N}(1-v_j)\sum_{i=1}^{n}u_{i,j} = 0 \\ U \in \{0,1\}^{n\times N} \\ V \in \{0,1\}^{1\times N} \\ W \in \mathbb{R}^{N\times d} \\ \sigma \in \mathbb{R}^{N} \end{cases} \quad (17)$$

4.2 Solving the Optimization Model Using Genetic Algorithm

In this part, we use the genetic algorithm to solve the non-linear optimization model obtained in the previous section.

Genetic algorithm:

The Genetic Algorithm (GA) is a revolutionary method introduced by J. HOLLAND since 1950. This method aims to solve a large number of complex optimization problems [16][6]. This latter has been applied in a large number of optimization problems in several domains, such as telecommunication, routing, scheduling, and it has proven great efficiency to obtain good solutions [8][11].

Let’s give a little explanation for this method: the space of feasible solutions (individuals) is coded by chromosomes, i.e., each solution represents an individual who is coded in one or several chromosomes. These chromosomes represent the problem's variables. First, an initial population composed by a fix number of individuals is generated; this number is fixed randomly, or through a preprocessing of the problem to solve. Then, operators of reproduction are applied to a number of individuals selected according to their fitness. This procedure is repeated until the maximum number of iterations is reached.

To make this resolution, we define an encoding, a crossing operator, a mutation operator and a function Fitness according to the particularities of this problem. This resolution allows, on the one hand, defining the optimal number of neurons in the map, and in the other hand, it allows finding the weighting matrix and the variances matrix.

Genetic algorithm for mathematical model:

In this section, we will describe the genetic algorithm to solve the proposed model for PRSOM architecture optimization (OAPRSOM). For this purpose, we have coded an individual by four chromosomes; moreover, the fitness of each individual depends on the objective function value.

Encoding: In our model, we have encoded an individual by four chromosomes , the first one represents control vector V , the second one represents the matrix of decision variables U, the third one represents the matrix of weights W and the last one represents the vector of variances σ .

Initial population: The first step in the functioning of a GA is, then, the generation of an initial population. Each member of this population encodes a possible solution to a problem.

The individuals of the initial population are randomly generated, u_{ij} and v_j take the value 0 or 1, and the weights matrix takes random values, in addition, vector of variances is initialized with the great values.

Evaluating individuals: After creating the initial population, each individual is evaluated and assigned a fitness value according to the fitness function.

In this step, to each individual is assigned a numerical value called fitness which corresponds to its performance; it depends essentially on the value of objective function in this individual. An individual who has a great fitness is the one who is the most adapted to the problem.

The fitness suggested in our work is the following function:

$$f(i) = \frac{1}{1 + E(i)} \tag{18}$$

Minimizing the value of the objective function is equivalent to maximizing the value of the fitness function.

Selection: The application of the fitness criterion helps to choose which individuals from a population will go on to reproduce.

Let's define: $P(i) = \frac{f(i)}{\sum_{i=1}^{nbid} f(i)}$ the selection criterion; where nbid is the cardinal of the population.

The individuals with greater fitness are thus more likely to be chosen. We can talk then about a proportional selection.

Crossover: The crossover is a very important phase in the genetic algorithm, in this step, new individuals called children are created by individuals selected from the population called parents. Children are constructed as follows:

- We fix a point of crossover, the parent are cut switch this point, the first part of parent 1 and the second of parent 2 go to child 1 and the rest go to child 2.

- In the crossover that we adopted, we choose 4 different crossover points, the first for the matrix of weights and the second is for vector U.

Mutation: The mutation's principle is to modify the values of each individual, chosen randomly. The mutation ensures the diversity of research to reduce the risk of finding local optima. Indeed, the genes of children are limited by the genes of the parents, and if a gene is not present in the initial population (or it disappears because of reproductions), it will never develop in the progeny. The aim of the mutation operator is to bypass this problem. Each gene has a low probability to mutate, i.e. to be randomly replaced by another incarnation of that gene. The purpose of this precaution is to maintain genetic diversity. For the matrix U and the vector V, we used a binary encoding. The mutation is to change the values of one or more genes ($0 \rightarrow 1$; $1 \rightarrow 0$). Concerning the matrix of weights W, we change one or more components of the matrix by a randomly generated value.

4.3 Training algorithm 2

The Fig. 2 presents the successive steps of the OAPRSOM algorithm. Firstly, we will build the mathematical model MINLP $(P_{\min})$. Afterwards, we will resolve it through the stochastic method of resolution: Genetic Algorithm. Then, as result, we obtain both the usable optimal number of the two PRSOM maps and the optimal initial parameters of the algorithm. Finally, we make the training of the OAPRSOM so as to obtain the optimal topology.

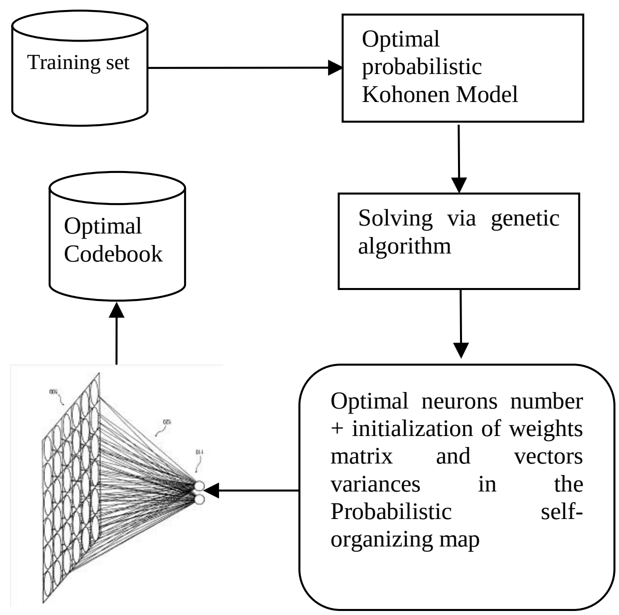

Fig. 1 Training Model OAPRSOM.

5. Experiments and discussion

To illustrate the advantages of the proposed method in this section, we apply our algorithm to the data base "Data Iris" to accomplish the classification task [25].

This data base is divided into three groups: Iris Setosa, Iris Verginica and Iris Versicolor. Each group contains 50 elements. Each element is characterized by four values: width of the petal, length of the petal, width of the sepal and length of the sepal.

Both the learning and the test are performed with 75 elements.

#Nmax: Maximal neurons number of the initial map.
#N: Optimal number of map's neurons.

Concerning the optimization algorithm, we have performed many tests so as to estimate the adequate number of neurons with the different data bases. Starting with the numbers 20, 30, 40 and 50; the results are shown in Table 1.

Table 1: The optimal neurons of the map using OAPRSOM

#Nmax	20	30	40	50
#N	7	7	8	7

We notice that the number of neurons retained, each time, converges to 7 in a decreasingly.

After determining the empirical number of neurons needed to project data, we are now interested in the quality of the training and the test of the proposed algorithm; the results are reported in Table 2 and Table 3.

Table 2: Numerical results for clustering the Training Data

Type	*Nr. Tr. D.*	*C.C*	*M.C*
Setosa	25	25	0
Virginica	25	24	1
Versicolor	25	23	2
Overall	75	72	3

The Table 2 presents the obtained clustering results of training data. We remark that this architecture permits to classify all the training data only three data; one from Virginica and two from Versicolor.

Table 3: Numerical results for clustering the Testing Data

Type	*Nr. Tr. D.*	*C.C*	*M.C*
Setosa	25	25	0
Virginica	25	24	1
Versicolor	25	25	0
Overall	75	74	1

The Table 3 presents the obtained clustering results of testing data. This table shows that our architecture gives good results, because all the testing data were correctly classified except one. In fact; this element (misclassified) is from the Virginica class.

Table 4: Comparison for Iris Data classification

Méthode	*N.Cl. App*	*N.Cl. Test*	*Pr.Cl.App. (%)*	*Pr.Cl.Tes. (%)*
EBP	3	2	96.0	97.3
RBF	4	4	94.6	94.6
RBF	4	2	96.0	97.3
SVM	3	5	94.6	93.3
AO PRSOM	3	1	96.0	98.6

Table 4 gives results of our AOPRSOM in comparison with other existing classification methods in the literature. We can tell that our model gives satisfactory results.

6. Conclusions and outlook to future work

In this paper, we have presented an approach to determine the optimal codebook and covariance matrix by the new Probabilistic Self Organizing Maps. As a first step we build a mathematical model in the form of MINLP with linear constraints, afterwards we solve it through dynamic clusters methods. After, we have introduced the optimization architecture model to PRSOM; this mathematical model is MINLP with linear constraints and nonlinear constraint resolved via genetic algorithm.

Using the data set Iris which is widely used in the clustering area, we have shown that our model outperform the other methods.

In the future works, we will use exact approaches or others heuristic methods to resolve this problem and determine the optimal solution for the MINLP. The proposed method can be applied to solve the pattern recognition problems and speech recognition problems.

References

[1] F. Anouar. Modélisation probabiliste des auto-organisées, Application en classification et en régression. Thèse de doctorat soutenue au conservatoire national des arts et métiers. 1996.
[2] F. ANOUAR, F.BADRAN and S.THIRIA, Self-Organized Map, A Probabilistic Approach proceedings of the Workshop on Self-Organized Maps, Helsinki University of Technology, Espoo, Finland, June 4-6,1997.
[3] F. Anouar, F. Badran, S. Thiria, "Probabilistic self-organizing map and radial basis function networks" Neurocomputing 20 (1998) 83-96
[4] C.M. Bishop. Pattern Recognition and Machine Learning, Springer,2006
[5] P. Bruneau, Marc Gelgon and F. Picarougne, Parameter-based reduction of Gaussian mixture models with a variational-Bayes approach, IEEE, 2008.

[6] J. Dréo, A. Pétrowski, P. Siarry and E. Taillard. "Métaheuristiques pour l'optimisation difficile". Eyrolles, 2003.
[7] algorithm for a class of mixed-integer nonlinear programs". Mathematical Programming, 1986, pp. 307-339.
[8] Z. En-Naimani, M. Lazaar, M. Ettaouil 'Hybrid System of Optimal Self Organizing Maps and Hidden Markov Model for Arabic Digits Recognition". Journal of WSEAS Transactions on Systems, Volume 13, pp. 606-616, 2014.
[9] Z. En-Naimani, M. Lazaar, M. Ettaouil, "Architecture optimization model for the probabilistic self-organizing maps and classification", 9th International Conference on Intelligent Systems: Theories and Applications, Rabat (Morocco), May 07-08, 2014.
[10] M. Ettaouil, M. Lazaar, "Improved Self-Organizing Maps and Speech Compression", International Journal of Computer Science Issues (IJCSI), Volume 9, Issue 2, No 1, pp. 197-205, 2012.
[11] Ettaouil, M. ; Lazaar, M. ; En-Naimani, Z. A hybrid ANN/HMM models for arabic speech recognition using optimal codebook , Intelligent Systems: Theories and Applications (SITA), 2013 8th International Conference on Mai 5-6.
[12] M. Ettaouil and Y. Ghanou, "Neural architectures optimization and Genetic algorithms", Wseas Transactions On Computer, Issue 3, Volume 8, 526-537, 2009.
[13] Ezequiel L. R. "Probabilistic self-organizing maps for qualitative data". Neural Networks 23 (2010) 1208–1225
[14] R. Fletcher and S. Leyffer. "Solving Mixed Integer Programs by Outer Approximation", Math. Program. 66, 1994, 327–349.
[15] O.K. Gupta and A. Ravindran. "Branch and Bound Experiments in Convex Nonlinear Integer Programming", Manage Sci., 31 (12) , 1985, pp. 1533–1546.
[16] T. Kohonen, S. Kaski, K. Lagus, J. Salojr , J. Honkela, V. Paatero, A. Saarela. "Self-organization of a massive document collection". IEEE transaction on neural networks, 11, No. 3, 2000.
[17] T. Kohonen. "Self-Organizing Maps". Springer, 3th edition, 2001.
[18] T. Kohonen. "The Self Organizing Maps". Proceedingsof IEEE, 78, No. 9, 1990, pp. 1464-1480
[19] A.Lotfi, A. Benyettou, "A reduced probabilistic neural network for the classification of large databases", Turk J Elec Eng & Comp Sci, 2014. 979-989.
[20] S.P Luttrel. A bayesian analysis of self- organizing maps. Neural Computing 6: 767-794, 1994.
[21] I. Quesada and I.E. Grossmann. "An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems", Computers Chem. Eng., 16 (10/11), 1992, pp. 937–947.
[22] N. Rogovschi. Classification à la base de modèles de mélanges topologiques des données catégorielles et continues. Thèse de doctorat soutenue à l'université Paris 13- Insitut Galilée. 2009.
[23] H. Soyel, H. Demirel, "optimal feature selection for 3D facial expression recognition using coarse-to-fine classification" Turk J Elec Eng & Comp Sci, vol.18, 2010.1031-1040.
[24] www.ics.uci.edu/mlearn/MLRepository.html.