Proceedings of the International Conference on Big Data Cloud and Applications
Tetuan, Morocco, May 25 - 26, 2015

Architecture optimization model for the probabilistic self-organizing maps

EN-NAIMANI Zakariae
Modeling and Scientific Computing Laboratory, Faculty of Science and Technology, Fez, MOROCCO
z.ennaimani@gmail.com

Mohamed LAZAAR
National School of Applied Sciences
Abdelmalek Essaadi University
Tetouan, MOROCCO
lazaarmd@gmail.com

Mohamed ETTAOUIL
Modeling and Scientific Computing Laboratory, Faculty of Science and Technology, Fez, MOROCCO
mohamedettaouil@yahoo.fr

***Abstract*— The PRobabilistic Self-Organizing Maps (PRSOM) become more and more interesting in many fields such as: pattern recognition, clustering, classification, speech recognition, data compression, medical diagnosis, etc. The PRSOM give an estimation of the density probability function of the data, which depends on the parameters of the PRSOM, such as the architecture of the network. When we take a random PRSOM architecture choice (the number of neurons or components), we could have degenerated solutions, called also singular solutions. Associated with a given problem, it is one of the most important research problems in the neural network research. In the present paper we describe a recent approach of probabilistic self-organizing maps (PRSOM) trying to propose a solution to this problem. We propose a speech compression technique based on vector quantization. The main innovation is the use of an optimal probabilistic self-organizing map to determine the optimal codebook, unlike in classical PRSOM. Also, we give an implementation and an evaluation of the proposed method; the numerical results are powerful and show the practical interest of our approach.**

Keywords*— Neural Network ; self-organization; classification; unsupervized learning; compression.

I. Introduction

Artificial Neural Network (ANN) often called Neural Network (NN) is a computational model or mathematical model based on biological neural networks.

Teuvo Kohonen has introduced the very interesting concept of self-organizing topological feature maps [18], The central property of this formalism is that it forms a nonlinear projection of a high-dimensional data manifold on a regular, low-dimensional (usually 2D) grid. In the display, the clustering of the data space as well as the metric-topological relations of the data items are clearly visible[17,19].

In the following we introduce the probabilistic Self-Organizing Maps (PRSOM) using a probabilistic formalism[1,2]. This algorithm gives a maximum approximation of the density distribution obtained by the learning phase. Since the training stage is very important in the probabilistic Self-Organizing Maps (PRSOM) performance, the selection of the architecture of PRobabilistic SOM, associated with a given problem, is one of the most important research problems in the neural network research. More precisely, the choice of components (neurons) number, the initial weights and covariances matrix has a great impact on the convergence of learning methods. The optimization of the artificial neural networks architectures, particularly PRSOM networks, is a recent problem. The first techniques consist in building the map in an evolutionary way: allowing, adding neurons and deleting some others. Some methods that have been proposed in the literature can be broadly classified into two categories: the first fixes a priori the size of the map in an evolutionary way [24]; the second category allows the data themselves to choose the dimension of the map. Recently, another method is introduced to determine the network parameters, in the supervised learning and in the Kohonen networks [8,9,10]. The mean purpose of this work is to model this choice problem of neural architecture in terms of a mixed-integer nonlinear problem with linear constraints. Because of its effectiveness in solving the optimization problems, the genetic algorithm approach is used to solve this nonlinear problem. It should be noted that a good local optimum of the obtained model permits to improve the performance of the PRSOM learning algorithm.

This paper is organized as follows: The section 2 presents the formalism of probabilistic self-organizing maps and vector quantization. In section 3 we introduce the model to optimize the probabilistic Self Organizing architecture Maps. And before concluding, experimental results are given in the section 5.

II. Probabilistic Self Organizing Map and Vector Quantization

A. *Probabilistic Self-Organizing Map*

In this section, we will briefly introduce the formal PRSOM model. It allows not only the quantification of data space, but also it does local densities estimation.

As the standard Self-Organizing Maps (SOM) [17,19,18], PRSOM consists of a discrete set C of formal neurons, which associates to each neuron $c \in (C)$ a spherical Gaussian density function f_c [5], which is defined by its mean (referent vector) $w_c \in \mathbb{R}^n$ and its covariance matrix. Thus we denote by $W = \{w_c; c \in C\}$ and $\sigma = \{\sigma_c; c \in C\}$ the two sets of parameters defining the PRSOM model [1].

In this probabilistic formalism presented in Figure 1, the classical map C is duplicated into two similar maps C^1 and C^2 provided with the same topology as C. It is assumed that the model satisfies the Markov chain hypothesis [7], thus for every input data $x \in D$ and every pair of neurons $(c_i^1, c_j^2) \in C^1 \times C^2$:

$$p(c_j^2 / x, c_i^1) = p(c_j^2 / c_i^1) \text{ and } p(x / c_i^1, c_j^2) = p(x / c_i^1)$$

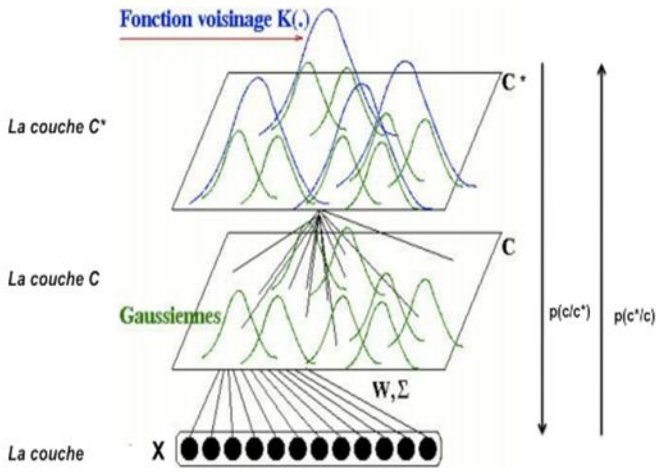

Figure 1: Probabilistic Self Organizing Map (PRSOM)

It is thus possible to compute the probability of any pattern x

$$p(x) = \sum_{j=1}^{K} p(c_j^2) p_{c_j^2}(x)$$

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

$$p_{c_j^2}(x) = p(x / c_j^2) = \sum_{i=1}^{K} p(c_i^1 / c_j^2) p(x / c_i^1)$$

The probability density $p_{c_j^2}(x)$ is a mixture of densities completely defined from the map given the conditional probability $p(c_i^1 / c_j^2)$ on the map and the conditional probability $p(x / c_i^1)$ on the data. In the following we deal with Gaussian densities and assume that:

$$p(c_i^1 / c_j^2) = \frac{K_T(d(c_j^2, c_i^1))}{\sum_{i=1}^{K} K_T(d(c_j^2, c_i^1))}$$

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

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

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

The curve of this likelihood is 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 [5].

The following algorithm ensures the convergence into a local maximum of data probability.

PRSOM learning algorithm:

- Initialization : k=0
- Initial parameters W^0 and σ^0, and the maximum number of iterations T_max is chosen.
- Let's compute $\chi^0(x) = \arg\max_{c_j^2} p_{c_j^2}(x) \quad j = 1, \dots, K$
- Iterative step k

$$w_{c_i}^k = \frac{\sum_{l=1}^{N} x_l K(d(c_i, \chi^{k-1}(x_l))) \dfrac{f_{c_i}(x_l, w_{c_i}^{k-1}, \Sigma_{c_i}^{k-1})}{p_{\chi^{k-1}(x_l)}(x_l)}}{\sum_{l=1}^{N} K(d(c_i, \chi^{k-1}(x_l))) \dfrac{f_{c_i}(x_l, w_{c_i}^{k-1}, \Sigma_{c_i}^{k-1})}{p_{\chi^{k-1}(x_l)}(x_l)}} \tag{1}$$

$$(\sigma_{c_i}^k)^2 = \frac{\sum_{l=1}^{N} \| w_{c_i}^{k-1} - x_l \|^2 K(d(c_i, \chi^{k-1}(x_l))) \dfrac{f_{c_i}(x_l, w_{c_i}^{k-1}, \Sigma_{c_i}^{k-1})}{p_{\chi^{k-1}(x_l)}(x_l)}}{n \sum_{l=1}^{N} K(d(c_i, \chi^{k-1}(x_l))) \dfrac{f_{c_i}(x_l, w_{c_i}^{k-1}, \Sigma_{c_i}^{k-1})}{p_{\chi^{k-1}(x_l)}(x_l)}} \tag{2}$$

With $c_i = 1, \dots, K$

$$\chi^k(x) = \arg\max_{c_j^2} p_{c_j^2}(x) \tag{3}$$

While (k>T_max)

The expression (1) is used to update the neurons weights (referents).

The expression (2) is used to update the neurons standard deviations.

The expression (3) is used to partition the data space.

B. Vector quantization

Vector quantization (VQ) is defined as follows: given a set of feature vectors Ω, find a partitioning of the feature vector space into the predefined K number of regions which $\Omega = \bigcup_{i=1}^{K} \Omega_i$ *with* $\Omega_i \bigcap_{i \neq j} \Omega_j = \Phi$. Every vector inside such region is represented by the corresponding centroid. These regions are called clusters and a set of centroids, which represents the whole vector space, is called a codebook[7].

In addition, vector quantization is considered as a data compression technique in the speech coding [9] [11]. Vector quantization has also been increasingly applied to reduce complexity problem like pattern recognition. The quantization method using the Artificial Neural Network, particularly in Probabilistic Self Organizing Maps, is more suitable in this case than the statistical distribution of the original data that changes with time, since it supports the adaptive data learning [11]. Also, the neural network has a huge parallel structure and the possibility for high speed processing.

But the main problems encountered in the probabilistic SOM formalism are:

- The risk to find degenerated solutions that present at least one neuron non adjusted to any input. But the likelihood of such Gaussian cannot be infinite, i.e. we get closer from a peak of Dirac.
- The problem of the network architecture choice, i.e. the number of neurons in the map and the initialization parameters.

III. PROPOSED MODEL TO OPTIMIZE THE PROBABILISTIC SELF-ORGANIZING ARCHITECTURE MAPS

A. Problem description

Generally, if the size of the probabilistic self-organizing map is chosen randomly, the PRSOM learning algorithm gives three classes of neurons as showing in Figure 2. The first class (red neurons) doesn't represent any observation (empty class), the second class (green) represents the neurons that contain few information data and the third class represents the important information data (blue).

In the above remark, we noticed that there exists a strong relation between the two problems mentioned in the previous section. In other words, we cannot distinguish between the two cases. When we take a random PRSOM architecture choice (the number of neurons or components), we could have degenerated solutions, called also singular solutions. More, the neurons (components) of the first class have a negative effect because they make the learning process heavier.

To overcome this problem we propose in this paper a new

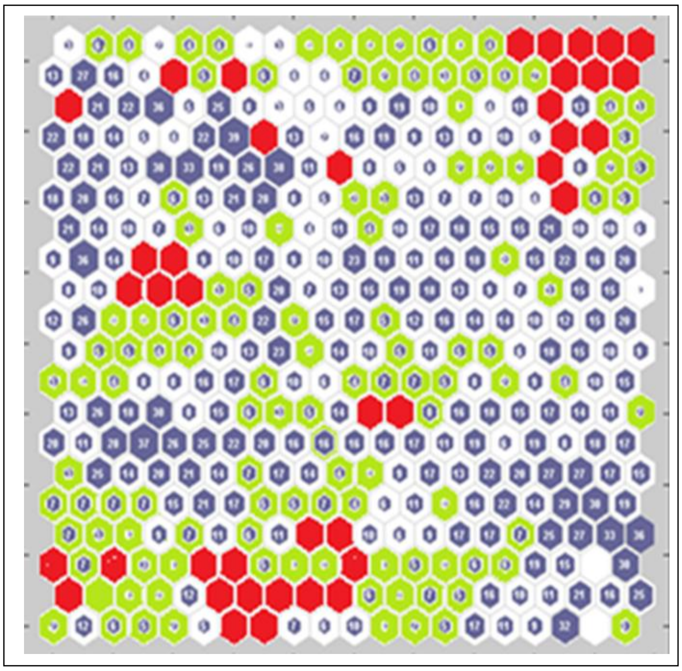

Figure 2: illustration of the three classes neurons of (PRSOM)

mathematical model of PRSOM that controls the size of the map. In this section, we will describe the construction steps of our model. The first one consists in integrating the special term which controls the size of the map. The second step gives the constraints which ensure the allocation of each data to only one neuron (component).

B. Modeling of PRSOM architecture optimization

We propose a new modeling of neural architecture optimization problem 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 : Optimal number of neurons (components) in the topology map of PRSOM,
- N_{max} : Maximal number of neurons in the topology map of PRSOM.

Variables

- $X = (x_{ij})_{1\le i\le n}$: Matrix of Training base elements;
- $U = (u_{ij})^{1\le j\le p}_{1\le i\le n}$:matrix of the binary variables
- $W = (w_{ij})^{1\le j\le N_{max}}_{1\le i\le N_{max}}$ Matrix of referent vectors
- $\sigma = (\sigma_i)^{1\le j\le p}_{1\le i\le N_{max}}$ matrix of covariance

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

$$(P_{Max}) = \begin{cases} \text{Max } \ p(U,W,\sigma) = \prod_{i=1}^{n}\prod_{j=1}^{N_{\max}} \left(\pi_j * \left(\sum_{k=1}^{N_{\max}} K^T(\delta(j,k))\theta_k(x_i, w_k, \sigma_k)\right)\right)^{u_{ij}} \ (1) \\ \textit{Subject to}: \\ \sum_{j=1}^{N_{\max}} u_{ij} = 1;\ldots;1 \le i \le n \ \ (2) \\ U \in \{0,1\}^{n \times N_{\max}} \\ W \in \mathbb{R}^{N_{\max} \times p} \\ \sigma \in \mathbb{R}^{N_{\max}} \end{cases}$$

The mathematical problem Pmax is equivalent to the problem P’max

$$(P'_{Max}) = \begin{cases} Max \ln(p(U,W,\sigma)) = \sum_{i=1}^{n}\sum_{j=1}^{N_{\max}} u_{ij}[\ln(\pi_j) + \ln(\sum_{k=1}^{N_{\max}} K^T(\delta(j,k))\theta_k(x_i, w_k, \sigma_k))] \ (1) \\ \textit{Subject to}: \\ \sum_{j=1}^{N_{\max}} u_{ij} = 1;\ldots;1 \le i \le n \ \ (2) \\ U \in \{0,1\}^{n \times N_{\max}} \\ W \in \mathbb{R}^{N_{\max} \times p} \\ \sigma \in \mathbb{R}^{N_{\max}} \end{cases}$$

The research for a maximum can always be transformed to the research of a minimum.

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

In the following section, we study the resolution of the last mathematical program.

C. *Resolution of the obtained nonlinear model*

We use the Genetic Algorithm approach to solve this mathematical model.

1) Genetic algorithm

Genetic Algorithm belongs to a class of stochastic methods called “evolutionary algorithms”. Introduced by J. HOLLAND [16], they are efficient and robust adaptive search techniques based on the idea of natural evolution (Darwin theory). This algorithm has been applied in a large number of optimization problems in several domains: telecommunication, routing, scheduling, and it proves its efficiency to obtain good solutions [24].

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 fixed number of individuals is generated, 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 attained.

The relevant steps of GA are:

Step 1: Coding individuals

Step 2: Randomly generate an initial population.

Step 3: Evaluate the fitness of each individual in the current population.

Step 4: Execute genetic operators including selection, crossover and mutation.

Step 5: Generate the next population using genetic operators.

Step 6: Return to Step 2 until the maximum of the fitness function is obtained.

2) Solving the optimized model

A specially designed genetic algorithm is applied to solve the optimization problem of the Architecture optimization model of the probabilistic self-organizing maps described in Section 3.2.

Encoding

In our model, we have encoded an individual by three chromosomes see Figure 3 , the first one (a) represent the matrix of decision variables U, the second one (b) represents the matrix of weights W and the last one (c) represents the vector of variances σ .

0.2	0.8	…	…	0.5
0.1	…	…	…	0.6
…	…	…	…	…
0.9	…	…	…	0.3

(a)

1	0	0	…	0	0	0
0	0	1	0	…	0	0
0	0	1	0	…	0	0
…	…	…	…	…	…	…
0	0	…	…	…	0	1
1	0	0	…	0	0	0
0	1	0	0	…	0	0

(b)

1200	780	395	…	…	702	2000

(c)

Figure 3: Genetic representation of an individual Initial Population

An initial population is built such that each individual must at least be possible solution, i.e., every component (U, W, σ) in the initial population must be feasible solution. The initial population could be randomly generated, but there exist other ways to generate the initial population like applying other heuristics. In our case, we do not use the random initialization of the variable U. When we set the variables W and Sigma in (P_{Min}), we find a linear model of binary variables under linear constraints. Thus, the initialization of the variable U is obtained by the resolution of the model (P_U), with W and Sigma randomly initialized.

The obtained model (P_U) is defined by:

$$(P_U) = \begin{cases} Min\ E(U) = -\sum_{i=1}^{n} \sum_{j=1}^{N_{\max}} u_{ij} \ln[\pi_j \sum_{k=1}^{N_{\max}} K^T(\delta(j,k))\theta_k(\mathrm{x}_i, w_k, \sigma_k)] (1) \\ Subject\ to: \\ \sum_{j=1}^{N_{\max}} u_{ij} = 1; ...; 1 \le i \le n \quad (2) \\ U \in \{0,1\}^{n \times N_{\max}} \end{cases}$$

The matrix U can be transformed into a vector X of size m, with m=n*Nmax

$$X = \begin{pmatrix} u_{1,1} & u_{1,2} & \dots & u_{1,k} & \dots & u_{i,1} & \dots & u_{i,k} & \dots & u_{n1} & \dots & u_{nk} \end{pmatrix}$$

Afterwards we can define the objective function as follows:

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

With:

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

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

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

$$\sum_{j=1}^{N_{\max}} u_{ij} = 1; ...; 1 \le i \le n \Leftrightarrow AX = b$$

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

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

$$b = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

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

$$(P_U) = \begin{cases} Min\ E(\mathrm{X}) = < C, \mathrm{X} > \\ Subject\ to: \\ \mathrm{AX} = \mathrm{b} \\ X \in \{0,1\}^{nN_{\max}} \end{cases}$$

Evaluating individuals

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 corresponding to this individual. An individual who has a great fitness is the most adapted to the problem.

The fitness suggested in our work is the following function:

$$f_i = \frac{1}{E_i + 1}$$

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

Selection

The application of the fitness criterion is intended to select which individuals from a population will go on to reproduce. Where:

$$P_i = \frac{f_i}{\sum_{j=1}^{n} f_j}$$

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 see Figure 4 :

We fix three points of crossover, the parents are cut from these points, the first part of parent 1 and the second of parent 2 goes to child 1 and the rest goes to child 2.

In the crossover that we adopted, we choose 4 different crossover points: the first one corresponds to the matrix of weights, the second one is for vector U and the last one corresponds to the vector of variances σ .

Mutation

The rule of mutation is to keep the diversity of solutions in order to avoid local optimums. It corresponds to changing the values of one (or several) value (s) of the individuals who are (s) chosen randomly.

IV. PROPOSED MODEL TO OPTIMIZE THE PROBABILISTIC SELF-ORGANIZING ARCHITECTURE MAPS

This algorithm is probabilistic self-organizing based on solving the optimization problem (P_{Min}) that gives in output: weights initialization (vectors referents), covariance matrix and the optimal neurons number. This is summarized in the following scheme Fig 5:

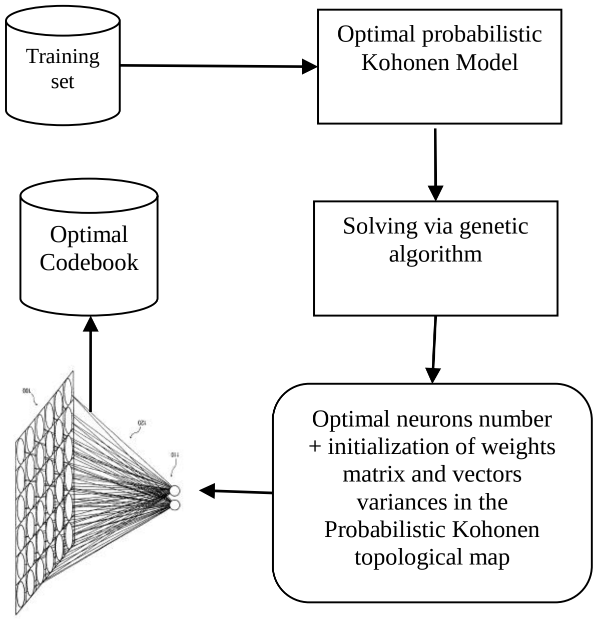

Figure 4: Training Model OPRSOM

To more understand the previous scheme, we explain it using the following iterative algorithm

Input:

n, p, X, $N_{iter}, N_{\max}$;

$[T_{\min}, T_{\max}]$ the interval of the parameter T;

Output:

Optimal probabilistic topological map

Initialization:

$w^1(0),\ldots,w^{N_{\max}}(0)$ Randomly initialized

$\sigma^1(0),\ldots,\sigma^{N_{\max}}(0)$ Randomly initialized with the great values

U initialized via resolution of the model (P_U).

$T \leftarrow T_{\max}$ $t \leftarrow 0$

Step 1:

Construction the model of PRSOM (P_{Min})

Step 2:

- Solving the model of PRSOM via Genetic algorithm.
- Outcome : the optimal number of neurons N used.
- Initial weights matrix Initial variances vector.

Step 3:

- Optimized model outputs, constructed in the initialization phase of OPRSOM.
- Training phase of OPRSOM.
- Assignment-decision phase (Equation 3).
- Minimization phase (Equation 1 and Equation 2).

Return

Optimal parameters of OPRSOM.

V. PROPOSED MODEL TO OPTIMIZE THE PROBABILISTIC SELF-ORGANIZING ARCHITECTURE MAPS

A. Data set Description

The experiments were performed using the Arabic digit corpus collected by the laboratory of automatic and signals, University of Badji-Mokhtar - Annaba, Algeria. A number of 88 individuals (44 males and 44 females), Arabic native speakers were asked to utter all digits ten times [27]. Depending on this, the database consists of 8800 tokens (10 digits x 10 repetitions x 88 speakers). In this experiment, the data set is divided into two parts: a training set with 75% of the samples and test set with 25% of the samples

Table 1. Arabic Digits

Arabic	English	Symbol
صفر	ZERO	‘0’
واحد	ONE	‘1’
اثنان	TWO	‘2’
ثلاثة	THREE	‘3’
أربعة	FOUR	‘4’
خمسة	FIVE	‘5’
ستة	SIX	‘6’
سبعة	SEVEN	‘7’

ثمانية	EIGHT	‘8’
تسعة	NINE	‘9’

Table 1 shows the Arabic digits, the first column presents the digits in Arabic language, the second column presents the digits in English language and the last column shows the symbol of each digit.

B. Experiments and discussion

In this section, we extensively study the performance of the proposed approach of speech compression using OPRSOM algorithm, Arabic digits set is considered.

The evaluation of the proposed approach in speech data compression was performed using the following measure,

Peak Signal-to-Noise Ratio (PSNR) is given by:

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\mathrm{n}X^2}{\mathrm{MSE}}\right)$$

Where n is the length of the reconstructed signal, X is the maximum absolute square value of the signal x, and Mean Squared Error (MSE) is defined as follows:

$$\mathrm{MSE} = \frac{1}{\mathrm{n}}\sum_{i=1}^{n}(\hat{x}(i) - x(i))^2$$

Where $\hat{x}$ is the original speech signal, and x is the reconstructed speech signal.

To choice of optimal neural network (N), we tried five different sizes of topological maps (Nmax). In each map, we compute the optimal size by our model (P). Numerical results obtained on dataset of Arabic digits are presented in the Table2. We note that the optimal size is between 3 and 7 neurons whatever the initial size is. For example, for a map of 50 neurons on digits 1,2,3,7 the optimal size is 5 neurons.

Table 2. Optimal results of topological map

	N_{max}	20	30	50
SYFR 0	N	7	6	7
WAHID 1	N	5	5	5
ITNAN 2	N	5	5	5
THALATA 3	N	5	5	5
ARBAA 4	N	3	4	3
KHAMSA 5	N	7	7	6
SITA 6	N	6	6	6
SABAA 7	N	5	5	5
THAMANIA 8	N	5	5	5
TISAA 9	N	7	6	7

The compression numerical results using optimal size are presented in Table 3. This table list all Arabic digits, The PSNR and the MSE calculated by classical approach (PRSOM) a map of T=20 (Because for the other choices of map we find degenerated solutions) neurons and the MSE calculated by a new size of map (mean of N for each digit) for example N=5 for WAHID (1), N=3 for ARBAA (4) neurons which determined by the proposed approach.

Table 3. MSE and PSNR obtained for Arabic digit by PRSOM and OPRSOM

ARABIC DIGITS	PSNR OPRSOM	PSNR PRSOM	MSE OPRSOM	MSE PRSOM
0	17.95	20.36	0.95	0.85
1	17.65	17.80	0.98	0.95
2	14.91	15.36	1.64	1.50
3	16.60	17.18	1.25	1.10
4	15.55	16.22	1.45	1.25
5	19.77	19.72	0.73	0.74
6	18.44	18.69	1.26	1.20
7	20.00	21.00	1.00	0.79
8	16.09	15.68	1.20	1.31
9	18.80	18.80	1.09	1.09

Figure 6 and Figure 7 show the MSE and the PSNR comparison of digits Arabic between both approaches PRSOM and OPRSOM. We can see that the MSE and PSNR very close between both approaches. But proposed method can reduce the training time and the number of neurons, from the 20 to 5 neurons, rate of reduction is about 75%.

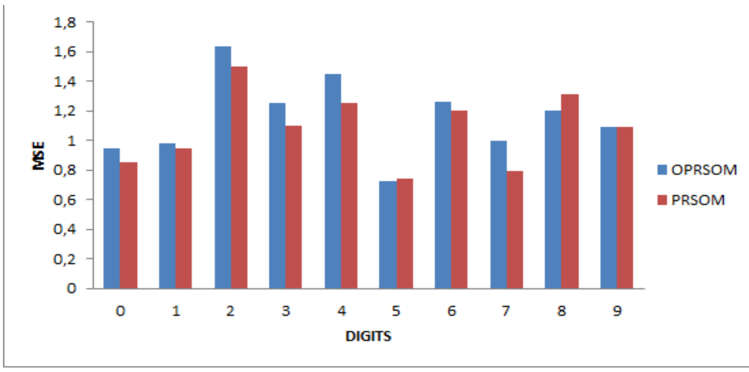

Figure 5: Comparison between both approaches for the MSE

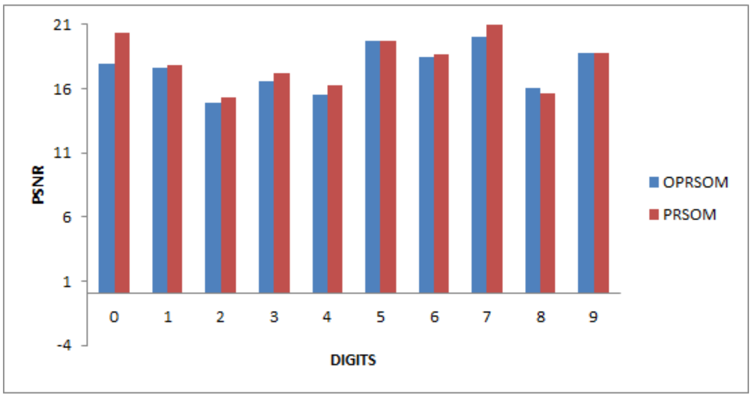

Figure 6: Comparison between both approaches for the PSNR

Recall that the proposed method contains an additional phase; this phase consists on solving the proposed model in order to remove the unnecessary neurons from the initial map. For example, for a map with 50 neurons we get a map of 5, the proposed approach can thus remove about 90% neurons from initial map to construct the optimal PRSOM.

VI. PROPOSED MODEL TO OPTIMIZE THE PROBABILISTIC SELF-ORGANIZING ARCHITECTURE MAPS

In this paper, we have presented an approach to determine the optimal codebook and covariance matrix by the Optimal Probabilistic Self Organizing Maps (OPSOM). As a first step we construct a mathematical model, after we solve via genetic algorithm, therefore we obtain the optimal number used in the card and the best initialization parameters of the network.

This approach has been compared to speech compression problem using a datasets of Arabic digit. The obtained results demonstrate the performance of our proposed method.

In the future works, we will use exact approaches or others heuristics methods to resolve this problem and determine the optimal solution for the optimization of neural networks architectures. The proposed method can be applied to solve the pattern recognition problems, speech recognition problems and image compression problems.

REFERENCES

1. *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.*
2. *F. Anouar. Modélisation probabiliste des auto-organisées : Application en classification et en régression. Thèse de doctorat soutenue au conservatoires national des arts et métiers. 1996.*
3. *M. Ayache, M. Khalil and F. Tranquart. "Artificial Neural Network for Transfer Function Placental Development: DCT and DWT Approach". IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 3, September 2011.*
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. *M. Duran, I. E. Grossmann. "An outer-approximation algorithm for a class of mixed-integer non linear programs". Mathematical Programming, 1986, pp. 307-339.*
7. 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.
8. 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.
9. *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.*
10. *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.*
11. *M. Ettaouil, M. Lazaar, K. Elmoutaouakil, K. Haddouch, A New Algorithm for Optimization of the Kohonen Network Architectures Using the Continuous Hopfield Networks, WSEAS TRANSACTIONS on COMPUTERS,Issue 4, Volume 12, April 2013.*
12. *R. Fletcher and S. Leyffer. "Solving Mixed Integer Programs by Outer Approximation", Math. Program. 66, 1994, 327–349.*
13. *Gascuel O. Canu S. Thiria .S and Lechevallier Y. Statistique et méthodes neuronales. 1997.*
14. *D.E Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning". Addison-Wesley, 1989.*
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. *Holland J. "Adaptation in natural and artificial systems". Ann Arbor,MI: University of Michigan Press, 1992*
17. *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.*
18. *T. Kohonen. "Self Organizing Maps". Springer, 3th edition, 2001.*
19. *T. Kohonen. "The Self Organizing Maps". Proceedingsof IEEE, 78, No. 9, 1990, pp. 1464-1480.*
20. *S.P Luttrel. A bayesian analysis of self- organizing maps. Neural Computing 6: 767-794, 1994.*
21. *S. Manuel, M. L. José, M. B. Victor, M.R. José, GENES, "a Genetic Algorithms and Fast Time Simulation'', 3nd ATM R&D Symposium, Spain, 2002.*
22. *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.*
23. *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.*
24. *E. Taillard, J. Dréo, A.Pétrowski, P. Siarry. Métaheuristiques pour l'optimisation difficile, Eyrolles,2003.*
25. *D. Wang. "Fast Constructive-Coverting Algorithm forneural networks and its implement in classification".Applied Soft Computing, 8, 2008, pp. 166-173.*
26. *M. YACOUB and al, Clustering and classification based on expert knowledge propagation using Probabilistic Self Organizing Map: application to geophysics, Data AnalysisStudies in Classification, Data Analysis, and Knowledge Organization 2000, pp 67-78 .*
27. *http://archive.ics.uci.edu/ml/datasets/Spoken+Arabic+Digit*