New variant of the Self Organizing Map in Pulsed Neural Networks to Improve Phoneme Recognition in Continuous Speech

Tarek Behi
Signal, Image and Pattern Recognition Research Unit-ENIT
Université Tunis El Manar
BP 37, le Belvédère, Tunis 1002

Najet Arous
Signal, Image and Pattern Recognition Research Unit-ENIT
Université Tunis El Manar
BP 37, le Belvédère, Tunis 1002

Noureddine Ellouze
Signal, Image and Pattern Recognition Research Unit-ENIT
Université Tunis El Manar
BP 37, le Belvédère, Tunis 1002

ABSTRACT
Speech recognition has gradually improved over the years, phoneme recognition in particular. Phoneme recognition plays very important role in speech processing. Phoneme strings are basic representation for automatic language recognition and it is proved that language recognition results are highly correlated with phoneme recognition results.
Nowadays, many recognizers are based on Artificial neural networks have been applied successfully in speech recognition applications including multi-layer perceptrons, time delay neural network, recurrent neural network and self-organizing maps (SOM), but present some weaknesses if patterns involve a temporal component. Let's note for example in speech recognition or contextual information, where different of the time interval, is crucial for comprehension.
In this paper, we propose a new variant SOM made of spiking neurons, with a view to emphasising the temporal aspect of the data which might serve as an input, in order to improve phoneme classification accuracy. The proposed variant, the Leaky Integrators Neurons, is like the basic SOM, however it represents the characteristic to modify the learning function and the choice of the best matching unit (BMU). The proposed SOM variant, show good robustness and high phoneme classification rates.

General Terms
Spiking neural networks, Self-organizing map, Speech processing, Neural Networks.

Keywords
Kohonen map, Temporal self organizing map, Leaky Integrator neurons, phoneme classification.

1. INTRODUCTION
The generality of engineering of processing and pattern recognition suppose that the dynamic process to recognize is stationary. Nevertheless, the static classification of the pattern is not a sufficient method when the dynamics of the pattern is an important characteristic of the information; it means that when the last information is necessary for the interpretation of current information. Let's note for example in speech recognition or contextual information, on different of the time interval, is crucial for comprehension. Spontaneous speech production is a continuous and dynamic process. This continuity is reflected in the acoustics of speech sounds, and, in particular, in the transitions from one speech sounds to another [1]. To take account of time in a system of data processing poses two great constraints. First, this system must be able to manage the succession of the various events which must be treated in a sequential way, it is then a question of sequential treatment. Thus, if the duration of the events is relevant for the task to carry out, the system must be able to treat the temporal structure. However, in the context of the speech recognition, the use of the static networks of neurons is difficult sight the absence of the parameter time in their structure.

In order to classify temporal sequences many technique have been used to model temporal relation in connectionist model [2] [3] like the networks of recurring neurons [4], the temporal self-organizing map [5] [6] [7] and networks of impulse neurons [8] [3] which prove the existence of robust techniques of recognition and classification.

In our model the temporal information is taken into account by using spiking neurons. Spiking neural networks (SNN) have become quite popular recently, due to their biological plausibility. Using spiking neuron models, SNN are able to encode temporal information into both spike timing and spiking rates. The model which realizes the spiking neurons as coincidence detectors encodes the training input information in the connection delays.

Spiking neural models can account for different types of computations, ranging from linear temporal summation of inputs and coincidence detection to multiplexing, nonlinear operations and preferential resonance [9]. Several recent studies employing rigorous mathematical tools have demonstrated that through the use of temporal coding, a pulsed neural network may gain more computational power than a traditional network (i.e., consisting of rate coding neurons) of comparable size [10]. A simple spiking neural model can carry out computations over the input spike trains under several different modes [9]. Thus, spiking neurons compute when the input is encoded in temporal patterns, firing rates, firing rates and temporal correlations, and space–rate codes. An essential feature of the spiking neurons is that they can act as coincidence detectors for the incoming pulses, by detecting if they arrive in almost the same time [11] [12].

In the following, we present the self-organizing map of Kohonen (SOM), then, we present some temporal self-organizing map models, thereafter we propose the Leaky Integrator neurons model (LIN). Finally, we illustrate experimental results of the application of SOM variant in phoneme classification.

2. SELF ORGANIZING MAP OF KOHONEN
A self-organizing map (SOM) is an unsupervised neural network algorithm that uses competitive learning [13] [14].

Competitive learning means that as data is input to the SOM, there is a competition among the neurons or nodes of the map to determine which neurons will represent the input data. The neurons of competitive networks learn to recognize groups of similar input vectors. Self-organizing maps learn to recognize groups of similar input vectors in such a way that neurons physically near each other in the neuron layer respond to similar input vectors [15].

The self-organizing map output represents the result of a vector quantization algorithm that gives a fixed number of references or prototype vectors onto high dimensional data sets in an ordered fashion. A mapping from a high dimensional data space ($\Re^n$) onto a two dimensional lattice of units is thereby defined.

An input vector $x \in \Re^n$ is compared with all mi, in any metric; in practical applications, the smallest of the Euclidian distances is usually used to define the best matching unit (BMU). The BMU is the neuron whose weight vector mi is closest to the input vector x determined by:

$$||x - m_c|| = \min\{||x - m_i||\}, \forall i \in [1..n] \quad (1)$$

Where n is the number of map units and $\|x - m_i\|$ is a distance measure between x and m_i.

After finding the BMU, his weight vector is updated so that the BMU is moved closer to the current input vector. The topological neighbors of the BMU are also updated. This adaptation procedure stretches the BMU and its topological neighbors towards the sample vector. Kohonen update rule for weight vector of the unit i in the BMU neighborhood is:

$$m_i(t+1) = m_i(t) + \alpha(t)\, h_{ci}(t)\, [x(t) - m_i(t)], \forall i \in [1..n] \quad (2)$$

x(t) is the input vector randomly drawn from the input data set at time t, $h_{ci}(t)$ the neighborhood kernel around the winner unit c and α(t) the learning rate at time t [16].

The neighborhood function h_{ci} is usually a function that decreases with the distance (in the output space) to the winning unit, and is responsible for the interactions between different units. During training, the radius of this function will usually decrease, so that each unit will become more isolated from the effects of its neighbors. It is important to note that many implementations of SOM decrease this radius to 1, meaning that even in the final step of training each unit will have an effect on its nearest neighbors, while other implementations allow this parameter to decrease to zero.

3. TEMPORAL SELF ORGANIZING MAP MODELS

3.1 Spatio-temporal kohonen maps

Mozayyani [17], have proposed the method to encode the time dependent data explicitly by extending the field of the SOM inputs from the real domain R into the complex plane C.

Each event is represented as a complex number:

$$x(t) = p(t)e^{i\phi(t)} = p(t)e^{iarct(\mu t)} \quad (3)$$

Where μ constant.

In order to establish uniqueness between the time variable t and the phase Φ, the past events with the negative time are restricted to be in the interval (- π/2; 0) while the positive time values (future events) are in (0; + π /2). The data samples are restricted to be always positive, p > 0.

The ST-Kohonen map [18] algorithm works in the same manner as classical kohonen one however, the winner is chosen according to the Hermetien distance:

$$Dist(X,W) = \|X - W\| \quad (4)$$

$$Dist(X,W) = \sqrt{\sum_{j=1}^{n} (x_j - w_j)\overline{(x_j - w_j)}} \quad (5)$$

x desingnes the map input and w_i is the weighting vector of the neuron i. Both the input and the weight vectors are defined in the complex domain C. the adaptation rule for ST-Kohonen is the same the one presented in kohonen, yet we manage complex vectors instead of real.

3.2 Recurrent self organizing map

The recurrent SOM [19] [20] is an extension to the Kohonen's SOM that enables neurons to compete to represent temporal properties in the data. Therefore, the RSOM that allows storing information from the past input vectors. The information is stored in the form of difference vectors in the map units. The mapping that is formed during training has the topology preservation characteristic of the SOM. Recurrent SOM differs from the SOM only in its outputs. In the training algorithm, an episode of consecutive input vectors x(n) starting from a random point in the input space is presented to the map. The difference vector $y_i(n)$ in each neuron of the map is updated as follows:

$$y_i(t) = (1-\alpha)y_i(t-1) + \alpha(x(t) - m_i(t)) \quad (6)$$

Where $y_i(n)$ is the leaked difference vector in unit i, $0 < \alpha \leq 1$ is the leaking coefficient. x(t) is the input vector and $m_i(t)$ is the weight vector of the unit i.

In fact, RSOM defines a difference vector for each unit of the map which is used for selecting the best matching unit and also for adaptation of weights of the map. Difference vector captures the magnitude and direction of the error in the weight vectors and allows learning temporal context. Weight update is similar to the SOM algorithm, except that weight vectors are moved towards recursive linear sum of past difference vectors and the current input vector [21]

3.3 Merge self organizing map

Barbara Hammer and Marc Strickert, have proposed a new method, merge SOM (MSOM), for unsupervised sequence processing for temporal data [22] [23]. The MSOM model combines a noise-tolerant learning architecture which implements a compact back-reference to the previous winner with separately controllable contribution of the current input and the past with arbitrary lattice topologie. In general, the merge SOM context refers to a fusion of two properties characterizing the previous winner: the weight and the context of the last winner neuron are merged by a weighted linear combination. During MSOM training, this context descriptor is kept up-to-date and it is the target for the folowing context vector of the winner neuron and its neighbourhood. In fact, MSOM accounts for the temporal context by an explicit vector attached to each neuron which stores the preferred context of this neuron. The way in which the context is represented is crucial for the result, since the representation determines the induced similarity measure of sequences [24].

For training, data vectors are iteratively presented, and the weights of the closest neuron and its neighbors in the grid are adapted towards the currently processed data point. The update formula for neuron I given pattern $\mathbf{x}_j$ is given by the formula [23]:

$$\Delta w_i = \alpha.h(d(i,I).(x_j - w_i) \quad (7)$$

where I is the winner, i.e. the neuron with smallest distance from $\mathbf{x}_j$, and h is a decreasing function of increasing neighborhood distance, e.g. a Gaussian bell-shaped curve or a derivative of the Gaussian.

4. THE PROPOSED VARIANT OF THE SOM

We present an algorithm to train the temporal behavior of leaky integrator networks in the context of spiking neuron networks. In the algorithm proposed here, we given a set S of m-dimensional input vectors $s = (s_1, ..., s_m)$ and a spiking neuron network with m input neurons and n output neurons, where each output neuron v_j receives synaptic feedforward input from each input neuron u_i with weight w_{ij} and lateral synaptic input from each output neuron v_k, with weight w_{kj}. At every epoch of the learning procedure one sample is chosen and the input neurons are made fire such that they temporally encode input vectors [25] [26].

In order to use leaky integrator units to create spiking neural network models for simulation experiments, a learning rule that works in continuous time is needed. The following formulation is motivated by [27] and describes how a backpropagation algorithm for leaky integrator units can be derived.

In this approach, the state of each neuron (*i)* is represented by a membrane potential $P_i(T)$, which, is a function of the input I(t) which measures the degree of matching between the neuron's weight vector and the current input vector.

The differential equation of a membrane potential is:

$$\frac{dP_i}{dt} = hP_i(t) + I_i(t) \qquad (8)$$

Where $\eta < 0$.

Particularly, the discrete version of the equation (8), often written as:

$$P_i(t) = l\, P_i(t - 1) + I_i(t) \qquad (9)$$

LIN memorise the last activation of each neuron i by means of a Leaky Integrators potential noted $a_i(t)$ [28] [29] [30]:

$$a_i(t) = l\, a_i(t - 1) - \frac{1}{2} \| x(t) - w_i(t) \|^2 \qquad (10)$$

where λ is a depth memory constant($0 \le \lambda \le 1$), x(t) is the input vector, and $w_i(t)$ is the weight vector of neuron *i.* Comparing equations (9) and (10), we find that $I_i(t) = -(½) \| x(t) – w_i(t)\|^2$.

5. EXPERIMENTAL RESULTS

5.1 Representation of speech data

We have used the TIMIT corpus for the purpose of developing and evaluating the proposed SOM variant for phonemes recognition in continuous speech and speaker independent. Speech utterance was sampled at a sampling rate of 16 KHz using 16 bits quantization. Speech frames are filtered by a first order filter whose transfer function is:

$$H(z) = 1 – a.z^{-1}, 0.9 \le a \le 1.0 \qquad (11)$$

Where z^{-1} is the delay operator. In our experiments, a is chosen to be 0.95.

After the pre-emphasis, speech data consists of a large amount of samples that present the original utterance. Windowing is introduced to effectively process these samples. This is done by regrouping speech data into several frames. In our system, a 256 sample window that could capture 16 ms of speech information is used. To prevent information lost during the process, an overlapping factor of 50% is introduced between adjacent frames.

After regrouping, each individual frame needs to be further pre-processed to minimize signal discontinuities at the beginning and at the end of each frame. A commonly used technique is to multiply the signal data with the hamming function. The earlier has smoothing effects at edges of the filter. This function can be described by the following equation:

$$h(n) = 0.54 - 0.46 * \cos(\frac{2\pi n}{N - 1}) \quad 0 \le n \le N,\ N > 1 \qquad (12)$$

Where n is the sample number and N is the total number of samples per window. In our case, N is 256.

Thereafter, mel frequency cepstral analysis was applied to extract the 12 mel cepstrum coefficients. The mel scale is an equi-pitch scale describing the subjective and perceptual response to frequency of human listener. The implemented neural networks are trained by presenting them with 12 input values from 9 frames selected at the middle of each phoneme. Table 1 shows the list of phonemes of each macro-class of TIMIT data base.

Table 1. List of phonemes of phonemic classes of each macro-class

Macro-class	Phonemes
affricates	/jh/, /ch/
Stops	/b/, /d/, /g/, /p/, /t/, /k/, /dx/, /q/, /bcl/, /dcl/, /gcl/, /pcl/, /tcl/, /kcl/
Others	/pau/, /epi/, /h#/
Nasals	/m/, /n/, /ng/, /em/, /en/, /eng/, /nx/
Semi-vowels	/l/, /r/, /w/, /y/, /hh/, /hv/, /el/
Fricatives	/s/, /sh/, /z/, /zh/, /f/, /th/, /v/, /dh/
Vowels	/iy/, /ih/, /eh/, /ey/, /ae/, /aa/, /aw/, /ay/, /ah/, /ao/, /oy/, /ow/, /uh/, /uw/, /ux/, /er/, /ax/, /ix/, /axr/, /axh/

5.2 Classification process

Classification is performed at frame level and performance is evaluated by comparing each classified frame with reference one.

In the case of the classic Kohonen model, a classification decision is operated as follows: for a given sample input vector, we search for its BMU. Thereafter, we look for its label.

In the case of the proposed SOM variant, a classification decision is operated in two steps. At a first step, for a test sample vector presented to a SOM variant we search for the BMU among all general centroid prototype vectors (GCPV) of

a map. Thereafter, inside selected BMU unit, we search for the best prototype vector of different classes, in terms of maximal activities.

5.3 Results and discussions

We have implemented the Kohonen model based on sequential learning and the proposed SOM variant. The realized system is composed of three main components [31] [32]: a pre-processor sounds and producing mel cepstrum vectors. The sound input space is composed by 12 mel cepstrum coefficients each 16 ms frame. 9 frames are selected at the middle of each phoneme to generate data vectors. The second component is a competitive learning module. The third component is a phoneme recognition module.

All maps are trained for 80 iterations using a data set consisting of 31070 sample vectors. For all maps, the learning rate decrease linearly from 0.9 to 0.05. The radius width decrease also linearly from half the diameter of the lattice to one. All maps of the same size have same initial conditions (that is the same m_i (0)). The neural lattice was bidimensional.

In our experiments, we have used the New England dialect region (DR1) composed of 31 male and 18 female. The corpus contains 31070 phonetic units for training and test. Each phonetic unit is represented by 9 frames selected at the middle of each phoneme to generate data vectors. Training has been made on phonemes for the seven macro classes of TIMIT data base. Table 2 shows the frame number of training and test data set of TIMIT speech corpus and the size of map for each macro classes.

Table 2. Number of samples of training and test data set of TIMIT speech corpus and the size of map

Macro-class	Frame number of training set	Frame number of test set	Size of map
Affricates	268	209	10*7
Stops	6206	2839	22*15
Nasals	1666	913	16*12
Semivowels	3959	1423	20*15
Fricatives	3899	1227	19*15
Vowels	12329	4036	26*19
Others	2743	1211	18*14

According to table 3, Leaky Integrator neurons (LIN) provides the best classification rate 93.20%. The LIN provides best rate for the phoneme /ch/ 94.39% in test set.

With LIN we obtained an improvement of the classification rate in comparison with SOM in order to 16 % in training and test set. From table 6, LIN provides best classification accuracy in comparison with SOM. With LIN we obtained an improvement of the classification rate in comparison with SOM in order to 19 % in training set and 33% in test set

.

Table 3. Affricates recognition rates

Phonemes	SOM		LIN	
	Training	test	Training	test
jh	74.03	64.70	92.30	84.31
ch	80.39	80.37	94.11	94.39
Average	**77.18**	**72.72**	**93.20**	**89.47**

According to table 4, Leaky Integrator neurons (LIN) provides best classification rate in order to 93% in training set.

With LIN we obtained an improvement of the classification rate in comparison with SOM in order to 31 % in test set.

However, LIN model reach good recognition rates (in the range of 85 and 100%).

Table 4. Semi-vowels recognition rates

Phonemes	SOM		LIN	
	Training	test	Training	test
l	70.31	67.32	79.88	80.69
r	91.10	86.00	95.25	95.00
w	72.38	57.76	88.56	89.32
y	92.00	76.47	91.80	97.05
hh	91.23	62.43	100	92.68
hv	64.61	26.60	96.81	92.61
el	93.62	41.37	97.01	89.65
Average	**82.16**	**59.66**	**92.76**	**91.00**

From table 5, with LIN we obtained an improvement of the classification rate in comparison with SOM in order to 18 % in training set and 21% in test set.

The LIN model provides the best recognition accuracy in test set 86.85%.

Table 5. Nasals recognition rates

Phonemes	SOM		LIN	
	Training	test	Training	test
m	57.14	55.26	80.29	90.13
n	48.51	52.90	79.20	69.67
ng	55.82	40.00	85.92	80.00
em	100	100	100	100
en	44.5	70.19	77.50	88.74
eng	100	0	100	0
nx	82.50	74.17	98.00	92.71
Average	**69.85**	**65.49**	**88.73**	**86.85**

The LIN model provides the best rate 100 % for the phoneme /axh/ in training and test set.

Table 6. Vowels recognition rates

Phonemes	SOM		LIN	
	Training	test	Training	test
iy	85.14	65.04	87.92	65.04
ih	39.72	15.19	44.86	42.64
eh	23.95	30.24	35.72	57.56
ey	47.51	24.27	64.21	69.41
ae	67.40	36.27	79.60	69.11
aa	50.59	54.68	65.53	79.80
aw	100	51.20	100	89.85
ah	39.20	44.55	64.55	40.09
ay	40.12	16.01	48.50	56.79
ao	65.94	13.93	72.04	73.63
oy	30.15	39.02	62.50	78.53
ow	13.43	26.36	44.66	60.19
uh	47.20	32.67	87.60	89.10
uw	86.70	38.00	95.04	67.00
ux	41.20	48.25	83.60	80.10
er	64.61	15.68	84.89	76.96
ax	26.74	18.81	57.48	42.57
ix	29.22	24.50	36.97	47.50
axr	41.61	17.64	60.75	60.78
axh	91.20	75.50	100	100
Average	**51.58**	**34.38**	**68.80**	**67.33**

Table 7 shows that LIN provides the best recognition accuracy both in training and test set. With LIN we obtained an improvement of the classification rate in comparison with SOM in order to 9% in training set and 20% in test set.
LIN reaches good classification rate in order to 81%in training set and 80% in test set.

The SOM variant provides the best classification rate for the phoneme /epi/ in order to 91.79% in training set and 81.75% in test set.

Table 7. Others recognition rates

Phonemes	SOM		LIN	
	Training	test	Training	test
pau	0	0	0	0
epi	74.50	53.00	91.79	81.75
h#	59.62	56.26	67.73	76.90
Average	**71.48**	**59.70**	**80.63**	**79.52**

From table 8, with LIN we obtained an improvement of the classification rate in comparison with SOM in order to 11% in training set and 17% in test set.

6. CONCLUSION

In this paper, we have proposed a new variant of self organizing neural network algorithm in the unsupervised learning category, and we are interested in phoneme classification from TIMIT data base by means of new SOM variant with impulse neurons, named Leaky Integrator neurons.

The Leaky Integrator neurons model is based on the conservation of information, which makes it possible to consider the temporal order between the successive samples

The LIN model provides the best classification rate for the phoneme /g/ in order to 77.22 % in test set.

Table 8. Stops recognition rates

Phonemes	SOM		LIN	
	Training	test	Training	test
k	33.55	25.36	48.83	56.58
kcl	14.00	6.86	20.00	33.33
dcl	56.62	22.77	56.29	40.59
q	47.03	24.87	68.42	65.17
g	51.65	45.05	64.57	77.72
p	36.75	40.09	70.53	45.54
t	47.40	23.76	61.36	57.92
b	68.10	71.14	54.48	68.15
d	41.72	43.5	50.33	55.50
bcl	61.71	53.88	63.03	48.05
dx	63.27	68.31	72.13	80.19
gcl	39.60	28.57	48.18	55.66
pcl	41.11	58.93	61.51	53.13
tcl	15.89	16.83	27.48	20.79
Average	**44.20**	**37.86**	**54.84**	**54.13**

According to table 9, The SOM variant provides the best classification rate in order to 78.27% in training set and 75.95% in test set.

With LIN we obtained an improvement of the classification rate in comparison with SOM in order to 10 % in training and 20% in test set.

Table 9. Fricatives recognition rates

Phonemes	SOM		LIN	
	training	test	training	test
s	74.75	63.39	76.71	73.85
sh	66.42	55.62	85.18	90.06
z	53.31	51.89	75.67	70.88
zh	100	95.45	100	90.90
f	70.27	57.41	71.49	87.09
th	84.31	34.66	92.15	52.00
v	68.31	76.77	75.99	90.32
dh	30.67	14.57	48.62	51.65
Average	**68.56**	**56.39**	**78.27**	**75.95**

by using a mechanism called Leaky Integrators. In this approach, the state of each neuron is performed by a membrane potential which is function of the input, this potential measure the adaptation degree between the neuron weight vector and the current input vector.

The use of impulse neurons in the SOM variant makes it possible to establish temporal associations between the consecutive models in a temporal order through the impulses produced according to the entry and makes it possible to improve the taking into account of the temporal parameters in the recurring SOM.

The case study of such learning algorithms is phoneme classification in continuous speech and speaker independent.

The proposed SOM variant provides best classification rates in comparison with the basic SOM model. The LIN provides the best general recognition rates of the 7 macro-classes of TIMIT data base in order to 80% in training set and 75% in test set.

As a future work, we propose to implement a cooperative system of SOM for phoneme recognition in order to improve classification rates. The system of SOM is based on the association of different SOM variants of supervised and unsupervised learning algorithms. We suggest also to hybridize SOM and genetic algorithm on one hand to fine tune SOM parameters and on the other hand for training data set input in the objective to ameliorate recognition rates

7. REFERENCES

[1] Santiage, F., Alex, G. and Jurgen, S. 2008. Phoneme recognition with BLSIM-CIC. In IDSIA.

[2] Durand, S. 1994. Réseaux neuromimétiques spatio-temporels pour l'organisation des sens. Application à la parole. Dans Actes Rencontres Nationales des Jeunes chercheurs en Intelligence Artificielle. Marseille.

[3] Durand, S. 1995. TOM, une architecture connexionniste de traitement de séquence. Application à la reconnaissance de la parole. PhD thesis, Université Henri Poincaré, Nancy I.

[4] Danilo, P., Mandic. and Jonathon, A. 2001. Recurrent Neural Networks for Prediction, John Wiley and Sons Ltd.

[5] Vaucher, G. 1993. Un modèle de neurone artificiel conçu pour l'apprentissage non supervise de séquences d'événements asynchrones. In Revue VALGO, ISSN 1243-4825. 1, 66–107.

[6] Behi, T. and Arous, N. 2008. Modèle auto-organisateur à composante temporelle pour la reconnaissance de la parole continue. Huitième journée scientifiques des jeunes chercheurs en génie électrique et informatique, GEI2008, Sousse-Tunisie.

[7] Behi, T. and Arous N. 2008. Modèles auto-organisateur à apprentissage spatio-temporels Evaluation dans le domaine de la classification phonémique. Cinquième conférence internationale JTEA2008, Hammamet-Tunisie.

[8] Brette, R. 2003. Modèles Impulsionnels de Réseaux de Neurones Biologiques. Thèse de Doctorat, Ecole Doctorale Cerveau-Cognition – Comporteme.

[9] Maass, W. and Bishop, CM. 1999. Pulsed Neural Networks. MIT Press.

[10] Maass, W. and Schmitt, M. 1997. On the complexity of learning for a spiking neuron. In COLT'97, Conf. on Computational Learning Theory, ACM Press. 54–61.

[11] Kempter, R., Gerstner, W., Van Hemmen, JL. and Wagner, H. 1998. Extracting oscillations: Neuronal coincidence detection with noisy periodic spike input. Neural Comput. 10, 1987-2017.

[12] Softky, WR. and Koch, C. 1993. The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. J. Neurosci. 13, 334–350.

[13] Kohonen, T. 1982. Self_Organized Formation of Topologically Correct Feature Maps. Biological Cybernetics. 43, 59-69.

[14] Arous, N. 2003. Hybridation des Cartes de Kohonen par les algorithmes génétiques pour la classification phonémique. Thèse de Doctorat en Génie Electrique, Ecole Nationale d'ingénieurs de Tunis.

[15] Haykin, S. 1999. Neural Network A Comprehensive Foundation, Prentice Hall Upper Saddle River, New Jersey.

[16] Kohonen, T. 2003. Self-organizing map, third edition, Springer.

[17] Mozayyani, N., Alanou, V., Derfus, J. and Vaucher, G. 1995. A spatio-temporal data coding applied to kohonen maps, in Proceeding of International Conference on Artificial Neural Network. 75–79.

[18] Zouhour, N., Laurent, B. and Frédéric, A. 2007. Spatio-temporal biologically inspired models for clean and noisy speech recognition Elsevier Science, Neurocomputing. 71, 131-136.

[19] Varsta, M., Heikkonen, J. and Milan, R. 1997. A recurrent self-organizing map for temporal sequence processing. Proc. Int. Conf. on Artificial Neural Networks (ICANNP'P97), Lausanne, Switzerland.421-426.

[20] Koskela, T., Varsta, M., Heikkonen, J. and Kaski, K. 1998. Time Series prediction using recurrent SOM with local linear models. International Journal of Knowledge-based Intelligent Engineering Systems. 2(1), 60-68.

[21] Koskela, T., Varsta, M., Heikkonen, J. and Kaski, K. 1998. Temporal sequence processing using recurrent SOM. KES. 1, 290-297

[22] Hammer, B., Micheli, A., Sperduti, A. and Strickert, M. 2004. A general framework for unsupervised processing of structured data Neurocomputing. 57, 3-35.

[23] Marc, S. and Barbara, H. 2005. Merge SOM for temporal data. In Neurocomputing. 64, 39-71.

[24] Salhi, M. S., Arous, N. and Ellouze, N. 2009. Principal temporal extensions of SOM: Overview. International Journal of Signal Processing, Image Processing and Pattern Recognition. 2(4), 61-84.

[25] Maass, W. 1998. Computing with spiking neurons .In Maass, W. and Bishop, C. M., editors, Pulsed Neural Networks, chapter 2, MIT-Press. 55-85.

[26] Maass, W. and Bishop, C. M. 1998. Pulsed Neural Networks. The MIT Press, 1st edition, Cambridge.

[27] Taylor, J. G. 1990. Temporal patterns and leaky integrator neurons. Proc. Int. Conf. Neural Networks (ICNN'90), Paris, 952-955.

[28] Koskela, T. and Varsta, M. 1998. Recurrent SOM with local linear models in time series prediction. Helsinki university of technologie-labo of computational engineering-Finland. (April 1998).

[29] Varsta, M. 1998. Temporal sequence processing using recurrent SOM. Helsinki university of technologie labo of computational engineering-Finland.

[30] Voegtlin, T. 2004. Réseaux de neurones et autoréférence. Thèse de Doctorat, université lumière lyon II.

[31] Arous, N. and Ellouze, N. 2002. Phoneme classification accuracy improvements by means of new variants of unsupervised learning neural networks, 6th World Multiconference on Systematics, Cybernetics and Informatics. Floride, USA, 14 – 18.

[32] Arous, N. and Ellouze, N. 2003. Cooperative supervised and unsupervised learning algorithm for phoneme recognition in continuous speech and speaker-independent context. Elsevier Science, Neurocomputing, Special Issue on Neural Pattern Recognition. 51, 225 – 235.