Self-organizing Maps for Speech Recognition

Caio Fernandes Araújo
Centro de Informática
Universidade Federal de Pernambuco
Recife, PE, Brasil
cfa4@cin.ufpe.br

Aluizio Fausto Ribeiro Araújo
Centro de Informática
Universidade Federal de Pernambuco
Recife, PE, Brasil
aluizioa@cin.ufpe.br

Abstract— **The spoken speech is the easiest and most natural way for the communication between human beings. So, the human-machine communication can be executed based on the way that human-human communication occurs. Researches in automatic speech recognition (ASR) have been developed for decades to produce communication as natural as possible. There some few attempts to use Self-organizing Maps to solve ASR problems, often working to execute pattern recognition. In this paper, we comparatively analyze the efficiency of two different neural networks, the Self-Organizing Maps (SOM) and the Time-Organized Maps (TOM), applied for the recognition of the American English phonemes. We considered phonological features to represent the input data. The results of the experiments suggest that the SOM is more efficient than TOM, even with simulations of disturbed data, including noise that may appear and harm the input signal quality.**

Keywords— Speech Recognition; Phonemes Recognition; Self-Organizing Maps; Time-Organized Maps; Phonological Features.

I. INTRODUCTION

As stated by Rios [1], speech is the prevalent way to occur communication between people in which the necessary skills are learnt along the lives of human beings, especially during our early childhood. Speech recognition is a complex task since one can argue that the nonlinear properties of a human voice can vary in pronunciation, volume, speed, accent among other features. Moreover, the speech broadcast process can be disturbed by different sources. People can naturally perform speech recognition; however, this is not the case for automatic speech recognition (ASR). Anusuya and Katti [2] present a number of relevant issues to design an ASR: environment, transducer, channel, speakers, speech styles, and vocabulary.

Automatic Speech Recognition (ASR) is the process of transforming a speech signal to a sequence of words, using a computer algorithm to do so [2]. The main objective of speech recognition is to develop systems for speech input to be used in machines. A recognition system can be designed to deal with four types of samples: isolated words, connected words, continuous speech and spontaneous speech.

There are three main approaches to speech recognition [2]: acoustic phonetic approach, pattern recognition approach and artificial intelligence approach. The acoustic phonetic was the first one, and it is based on finding and labeling speech sounds. The pattern recognition uses a mathematical framework to establish pattern representation, for comparison purposes. The artificial intelligence is a merge between the two previously mentioned. It uses information regarding linguistic, phonetic and spectrograms, and is the approach used in this paper

Anusuya and Katti [2] argue that ASR is useful for many applications whenever human-machine interface is necessary. The authors list a number of real world applications: automatic call processing, query information systems, stock price quotations, weather reports, voice dictation, avionics information, automobile portal, speech transcription, orientation for blind people, railway reservations. Hidden Markov Models (HMMs) are the most common approach to deal with ASR since the mid-1980s. A high number of commercial ASR systems are currently based on such statistical framework. Despite significant progresses along the years, HMM is not the only approach to handle ASR.

Anusuya and Katti [2] point out that since the late 1980s, neural networks (NNs) are applied to speech recognition problems. The main contributions of NNs involve knowledge representations and integration of distinct knowledge sources. Among the NNs, one can mention Support Vector Machine (SVM), recurrent versions of Multi-layer Perceptrons, Learning Vector Quantization (LVQ) as supervised neural networks models which were employed to ASR. The use of unsupervised models is still a quite unexplored alternative. Hence, a clustering algorithm used in ASR can be an alternative. A feedfoward version or a recurrent version of such an approach is one of the first issues to be considered.

This work aims to compare, using a same representation, storage and recognition of phonetic information based on clustering, taking into account both spatial and spatio-temporal features. More specifically, this is an initial study comparing the effectiveness of two algorithms to handle ASR: the Self-Organizing Maps (SOM) [3] and Maps Organized in Time (TOM) [4]. The former is an unsupervised learning spatio-based clustering algorithm and the latter is a spatio-temporal-based clustering algorithm. Both algorithms were previously used for speech recognition problems with different database [5-6], whereas we have considered phonological features to represent the data.

This paper is organized as follows. In Section II, we briefly discuss the use of Self-organizing Maps for ASR, considering the original model and recurrent versions of it. In Section III, we defined both chosen algorithms, SOM and TOM. The results are presented in Section IV and the conclusion in Section V.

II. Neural Networks for Speech Recognition

Speech recognition may be divided in five steps: Acquisition, pre-processing, feature extraction, categorization of the speech signal to produce its output. The first step consists in the acquisition of speech raw signal, either by using a microphone or other equipment. The second stage acts to address the problems of signal perturbation or any other interference in the acquired audio signal. The third step is to extract specific features of the processed signal. The fourth step is the categorization, i.e., to determine how a given input is related to the know vocabulary or set of signals. Finally, using the response from the previous step, the output is yield. NNs are typically considered for the fourth and fifth steps.

A. SOM and Speech Recognition

A pioneer work on speech recognition using Self-organizing Maps was the neural phonetic typewriter [7] in which phonemes of the Finnish idiom were clustered. Some years later, Somervuo and Kohonen [8] used a version of the Learning Vector Quantization – LVQ [3] to feature sequences recognition. In 2003, Wiemer proposed TOM [4] to deal with spatio-temporal relations between stimuli presented to it. Wiemer argues that the order in which a speaker utters a string of phonemes and words is relevant during training and thus it influences the recognition result. Such an algorithm presented encouraging results for mono-speaker, speaker-independent and unknown speaker speech recognition of isolated digits. The authors argued that spatio-temporal categorization seems to be suitable to handle speech recognition [5-6].

Speaking of categorization based on spatio-temporal features, here are a number of SOM variations to handle spatio-temporal data [9], i.e., unsupervised recurrent self-organizing networks for straight sequence processing. One can mention the use of the leaky integrator in the Temporal Kohonen Map (TKM) [10] and in the Recurrent SOM (RSOM) [11], and combination of context model with arbitrary lattice structures in Merge SOM (MSOM) [12] as typical samples of this approach.

TKM and RSOM take advantage of leaky integrator memory to create temporal context for the sequence of input signals. In both, TKM and RSOM, the memory is added to the outputs, in which the leaked quantity is the squared norm or the difference vector respectively. The recurrence in TKM and RSOM are locally defined. The context model of MSOM can be suitable for different lattice structures. In MSOM, the weight vector associated with the last winner and its context linearly combined. In turn, TOM transforms spatial distances into spatio-temporal distances, taking into account interactions between current and former signal through propagation waves. Because TOM has some degree of plausibility, stores spatio-temporal weight vectors and it was previously used for speech recognition, we chose to use TOM as our first attempt to evaluate the influence of the time over ASR.

III. Self-Organizing Maps for Speech Recognition

The solution proposed to this problem is the utilization of two artificial neural networks. The first one is the Self-organizing Map (SOM). The SOM is an artificial neural network [3] that uses unsupervised learning to perform a dimension reduction having the least possible loss of information. In order to keep the topological characteristics of the higher dimensional input space, a neighborhood function is used. The topological map is ordered automatically, when compared, several times, the input feature vectors and the weight vectors of each node in the network.

The SOM operates as follows: the weight vectors of the network nodes are initialized with random values, and preferably at very close positions, to prevent neighbors of being far apart in the beginning. After initialization, an input feature vector is presented to it, to determine the most similar weight vector to the feature vector. The selected node is called the winner. As training advances, the network updates the nodes to become more similar to the feature vectors and also updates its neighbors, according to the equation below:

$$\mathbf{w}_i(n) = \mathbf{w}_i(n-1) + h(n).h_{i,w}.(\mathbf{s}_k(n-1) - \mathbf{w}_i(n-1)) \qquad (1)$$

where $\mathbf{w}_i$ (n) is the weight vector of the i-th unit, ? is the learning rate ranging between 0 and 1, $h_{i,\mathrm{w}}$ is the neighborhood function which depends on the winning node, w, also varying between 0 and 1, and $\mathbf{s}_k$ (n) is the k-th input feature vector at instant n. Hence, the updating involves the winner node and its neighbors. The network induces that the generated map has topological characteristics similar to the input data

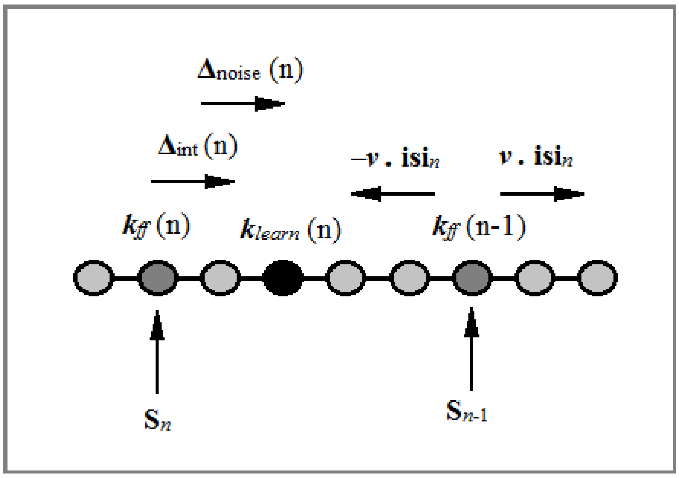


Fig. 1. Topology of the TOM. The input checks the node with the nearest features, the previous node and, in accordance with the noise, updates the value of a new node.

The second considered algorithm is the Time Organized Maps (TOM). The TOM [4] is also an unsupervised neural network, also capable of dimension reduction, Fig. 1, and having the least loss of information. Unlike the SOM, TOM aims to construct spatio-temporal maps. Calculating the shortest distance between the input vector and the features vectors is only the first step. After finding the winner node, the network uses memory information, considering the winner of the previous iteration, its spread in the network over time and problems due to noise. Thus, each map region considers both spatial and temporal proximity between patterns.

The TOM operates as follows: after the initialization equal to the SOM, an input feature vector is presented to determine the most similar weight vector. Next, the winner node is not updated. Instead of this, the processing of the temporal context is executed. Firstly, the position of the wave propagated by winner node in the previous iteration is calculated. The calculation takes into account the speed of propagation of this wave as well as the time between the current and the previous iteration. As the network used is one-dimensional, the wave can propagate in two different directions. The chosen direction is the one that leaves the winner node in the previous iteration into the direction of the current winner node. Then, a deviation due to noise is also determined. Such a term decreases over time to reach a constant value. After calculating the position of the propagated wave and the deviation due to noise, according to Table I [4], these values are added to the winner node. The resulting value is rounded to the nearest integer, chosen as the node to be updated according to the equation below:

$$\mathbf{w}_{k_{learn}}(n) = \mathbf{w}_{k_{learn}}(n-1) + h(\mathbf{s}_k(n-1) - \mathbf{w}_{k_{learn}}(n-1)) \qquad (2)$$

where $\mathbf{w}_{k_{learn}}(n)$ is the weight vector of the network, *?* is the value of the learning rate of the network, that was 0.01 and $\mathbf{s}_k$ is the input feature vector. This way, only the node k_{learn} is updated, unlike SOM that also updates the neighborhood.

TABLE I. EQUATIONS OF THETOM

Calculate the position of the propagated wave	
$\tilde{k}_{\pm}(n) = k_{ff}(n-1) \pm v.isi_n$	(3)
$isi_n = isi(i_n, i_{n-1}) = \dfrac{1}{V_s}\left\|\ln \dfrac{i_n}{i_{n-1}}\right\|$	(4)
Calculate the variation caused by the interaction with the propagated wave	
$\Delta_{\text{int}}(n) = f\left(\tilde{k}(n) - k_{ff}(n)\right)$	(5)
$\tilde{k}(n) = \begin{cases} \tilde{k}_+(n), & if \quad \left\|\tilde{k}_+(n) - k_{ff}(n)\right\| \le \left\|\tilde{k}_-(n) - k_{ff}(n)\right\| \\ \tilde{k}_-(n), & otherwise \end{cases}$	(6)
$f(x) = k\,.\tanh\left(\dfrac{x}{k}\right)\exp\left(-\dfrac{x^2}{2(s_x)^2}\right)$	(7)
Calculate the variance due to noise	
$\Delta_{noise}(n) \sim N(0, s_{noise}(n))$	(8)
$s_{noise}(n) = s_0 \cdot \left(\dfrac{s_f}{s_0}\right)^{n/n'_f} \quad if\ n \le n'_f$	(9)
$s_{noise}(n) = s_f \quad if\ n'_f < n \le n_f$	(10)
Find the position of the new winner node	
$k_{learn}(n) = round\left(k_{ff}(n) + \Delta_{\text{int}}(n) + \Delta_{noise}(n)\right)$	(11)

where $v = 1$ $V_s = 1$, $? = 5$, $s_0 = 15$, $s_f = 0.1$, $k_f = 10^6$ $k'_f = 0.9k_f$, $k_{ff}(n\text{-}1)$ is the closest node in the previous iteration, and i_k is the position of the chosen phoneme.

The training sequence is equal to that of SOM and TOM follow the same random order to present the feature patterns.

IV. RESULTS OF THE SIMULATIONS

This section shows the results of the simulations for the trained networks. Such experiments determine the different clusters, aiming to show each group individually, and evaluate the capacity of individual recognition under noise. Besides that, the results evaluate the recognition of each sentence of phonemes and the difference that a training strategy causes during the recognition stage.

Waveform is one of the most used representations of sounds of a language. This is a long representation in spite of being accurate; hence other alternatives can be welcome. Ladefoged [13] argues that both speech synthesis and speech recognition can work properly using representation different from conventional linguistic units. In particular, ASR may interpret different contents employing phonemes as such units. Ladefoged [13] points out the existence of a conflict between the representation of lexical contrasts in a language and both, the definition of a well formed syllable and the pronunciation of sound samples of a language. Considering the definition of phonetic components based on phonological features [14-15], the components can be defined in terms of phonetic properties of a language. Then, in spite of neglecting lexical contrasts, adequate syllables and sounds can be represented when considering correct statements of phonological patterns.

Ladefoged [16] proposed a way to represent the sounds of a word in English. The author used distinctive features, i.e., he organized the lexical contrasts according to a hierarchical arrangement of articulatory characteristics subject to constraints. Each input is represented by a set of articulatory features. This representation type is similar to that proposed by Araujo et al [17] as an adaptation of that one previously introduced by Hinton and Shallice [18].

The database used in the following experiments, created specifically for our simulations, to analyze English phonemes. The representations of the phonemes were separated in two classes: the vowel feature set and the [17]. The former has four attributes whereas the latter has eight, all of them varying between -1 and 1. The vowels features are *Open-Close, Front-Back, Tense-Lax* and *Rounded-Unrounded*. The consonant features are *Bilabial-Dental-Glottal, Nasal, Plosive, Fricative, Voiced, Glides, Retroflex* and *Lateral*. Hence, 42 phonemes were mapped in which 18 were considered vowel phonemes and 24 were consonant phonemes. After defining the representation of the phonemes, a set of 10 sentences was defined to train the network, chosen to be simple orders, to be used in the future to control a robot. Such sentences are orders of simple instructions, for more details see Appendix.

We defined two ninety-unit one-dimensional maps (SOM and TOM) in which their weight vectors were randomly initialized. Despite the actual initial values of the weight vectors were randomly determined, both, SOM and TOM had the same initialization to compare them under equal conditions. The presentation order of the sentences and of the phonemes in each phrase, used during the training stage, were also random. SOM's values of ?(n) and $h_{i,w}$ are defined by:

$$h(n) = 1/2 * (1/2)^{n/100000} \tag{12}$$

$$h_{i,w} = \begin{cases} 1, & \text{if winner} \\ 1/3, & \text{if first neighbor} \\ 1/10, & \text{if second neighbor} \\ 1/50, & \text{if third neighbor} \end{cases} \tag{13}$$

After training, both networks had their learned prototypes represented by their weight vectors were very distinct from the initial ones and the corresponding phonemes are shown in the Table II for SOM and in Table III for TOM. The phonemes for each cluster can vary significantly between the models and so can the neighborhood relations.

TABLE II. PHONEMES CLUSTERED BY SOM.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ao	ao	ao	ao	ao	ao	ao	ao	ao	ao	ao	ao	ao	ao	ao
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
ao	ao	ao	ao	b	b	b	b	d	d	d	g	g	g	ng
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
ng	n	n	n	m	l	l	l	r	r	r	w	y	y	jh
46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
dh	z	zh	h	sh	s	th	p	t	t	k	k	k	aa	er
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
ow	ow	ow	uw	ow	ay	ay	ey	ey	ey	iy	iy	ih	ih	ih
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
eh	eh	eh	eh	ae	ae	ae	ah	ah	ah	ah	ah	ah	ao	ao

TABLE III. PHONEMES CLUSTERED BY TOM.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ao	ao	ao	ao	ao	s	ao	s	ao	s	s	s	s	s	s
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
sh	sh	th	h	dh	z	z	z	v	p	t	t	k	t	t
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
t	k	k	d	d	k	g	d	d	d	d	d	uw	uw	ow
46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
ow	ow	ow	ow	ow	ow	ey	ey	ey	ey	ey	ey	ey	ey	ey
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
ey	ey	iy	iy	iy	aa	aa	aa	aa	er	er	aa	aa	ay	eh
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
ae	ix	eh	eh	ah	ah	n	eh	l	n	r	l	y	r	r

To verify the robustness of the system, we need to do tests using inputs with noise, since a natural environment is often subject to disturbances. Such noise, added to original inputs, was determined by a normal probability distribution function, with mean zero and standard deviation according to the maximum allowed noise. Thus, low noise levels are more likely to occur.

Initially it was found the mean square error between each phoneme and the weight vector closest to it, shown in Fig. 2. The results suggest that the TOM is greater than the error of the SOM. This is due to the fact that the error is calculated by the distance between the winner prototype and the original representation of the phoneme, considering only the patterns spatial differences.

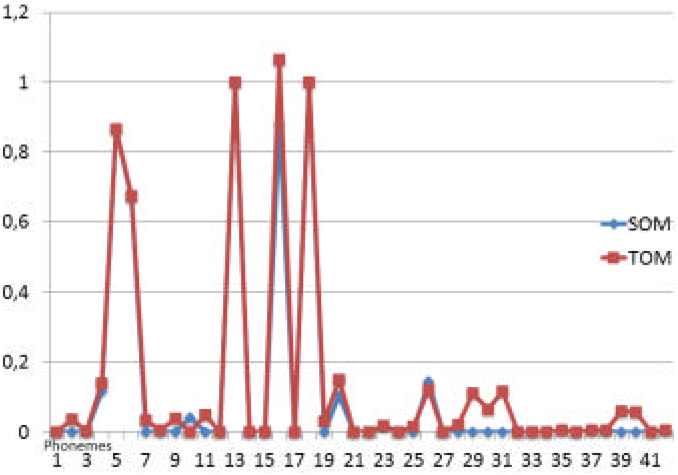


Fig. 2. Mean square error of each phoneme for SOM and TOM.

The next tests involve recognition for entries with different noise levels, ranging from zero to a maximum value of 20%, and the 10 sentences used for training and testing have 29 distinct phonemes out of a total of 152 phonemes. The tests were divided into two parts: one where only the winner nodes in the network were maintained for verification and other where the neighboring nodes were also maintained. The Fig. 3

shows the number of distinct phonemes that were incorrectly detected when only the winner nodes in the network are considered.

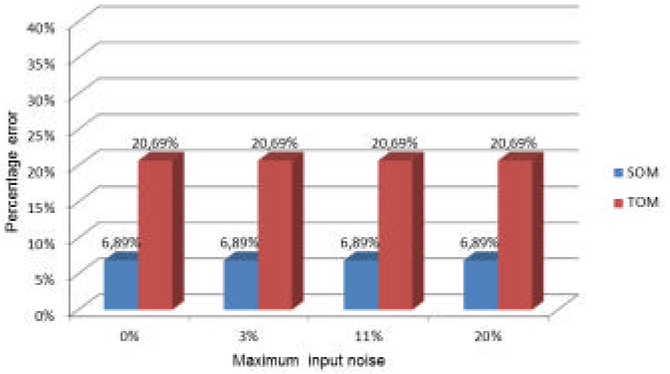


Fig. 3. Percentage of distinct phonemes incorrectly detected only by the winner.

The Fig. 4 shows the number of distinct phonemes that are incorrectly recognized when considering the neighborhood, according to the level of input noise.

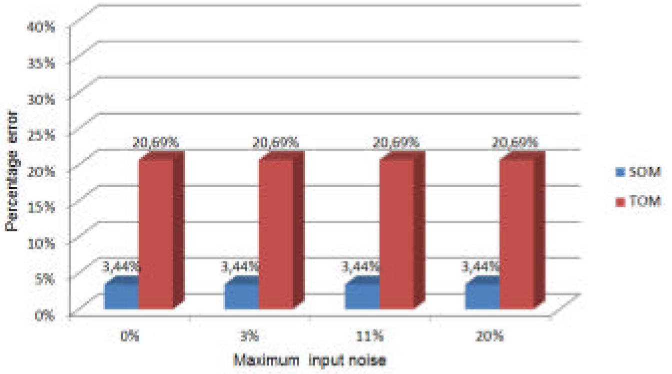


Fig. 4. Percentage of distinct phonemes incorrectly detected with neighborhood.

Tests using temporal characteristics were also carried out for the recognition by the two networks. In these tests, shown in Fig. 5, it is intended to assess whether the pattern entered in their temporal context is recognized differently by the networks. The following equation was used for the pattern recognition by inserting the temporal context:

$$h(t) = a\, x(t) + (1 - a)\, h(t\text{-}1) \qquad (14)$$

where 0.5 = a = 0.95 and $x(t)$ is the input. This way, the euclidean distance is calculated between the phoneme database and the value of $h(t)$ and not the value of $x(t)$ anymore.

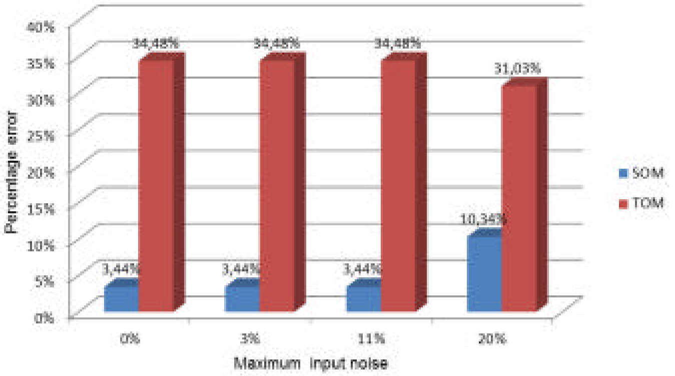


Fig. 5. Percentage of distinct phonemes incorrectly detected with neighborhood using the Eq. (14).

The SOM showed a drop in its performance when there was an increase in noise, since, in their training, temporal aspects are not considered. However, contrary to what was expected, the TOM also had a decrease of performance, even with different values of a. The Fig. 6 shows the number of distinct phonemes that are recognized incorrectly for temporally contextualized patterns by Eq. (14).

V. CONCLUSION

The results suggest that SOM does better than TOM at recognition of phonemes for all types of training and tests. In SOM, the neighborhood is similar in terms of spatial characteristics, with all vowel phonemes together, without consonant phonemes in the middle and vice versa, and only one phoneme was not learned. In TOM, four phonemes were not learned and even one phoneme that was not used in the training was learned, due to the temporal aspects. After training, the recognition was done using Euclidean distances and therefore this worked in favor of the SOM. With the inclusion of noise in the inputs, the SOM continued with better performance, since, being widely spread in space, most phonemes end up getting relatively far from each other. As TOM also considers temporal features, the consequence was the production of more errors.

The use of Eq. (14) for recognition did not show satisfactory results as well. The SOM had a similar result when using only the spatial features, however the TOM presented a result below the SOM recognition. It was expected that TOM, using an equation which also took into consideration the previous winner, could obtain a better result. The TOM worse performance probably occurred because of the training method, since it failed to correctly map all phonemes. TOM created temporal combinations of phonems presented in sequence, as two diferent phonems can preeced a considered phoneme, the prototype of the latter can be influenced by very distinct previous stimuli. Thus, the temporal context seems toencode more conflict into the prototype representations than to solve it. In sum, the results suggest that under the described circumstances, TOM was not as suitable as SOM for automatic speech recognition.

The measure of temporal evaluation can be very important for the recognition of patterns. Hence, an approach that considers the temporal aspects using probabilities to estimate the next pattern, from a comparison with a trained neural network seems a good and viable solution to tested in future works.

REFERENCES

[1] D. Rios, "Speech Recognition", available at http://www.learnartificialneuralnetworks.com/speechrecognition.html, 2010.

[2] M. A. Anusuya and S. K. Katti, "Speech Recognition by Machine: A review", International Journal of Computer Science and Information Security, vol. 6, no. 3, 2009, pp. 181-205.

[3] T. Kohonen, "The Self-Oganizing Map", Proceedings of the IEEE, Vol. 78, Issue 9, 1990, pp. 1464 – 1480.

[4] J. C. Wiemer, "The Time-Organized Map algorithm: Extending the Self-Organizing Map to spatiotemporal signals", Neural Computation, vol. 15, no. 5, 2003, pp. 1143-1171.

[5] Z. N. Ben Salema, L. Bougrain b, and F. Alexandre, "Spatio-temporal biologically inspired models for clean and noisy speech recognition", Neurocomputing, Vol. 71, no. 1–3, 2007, pp. 131–136.

[6] X.-P. Du and P.-L. He, "The clustering solution of speech recognition models with SOM", Proc. IEEE International Symposium on Neural Networks (ISNN 2006), vol. 3972, 2006, pp. 150-157.

[7] T. Kohonen, "The Neural Phonetic Typewriter", Computer, vol. 21, no. 3, 1988, pp.11-22.

[8] P. Somervuo and T. Kohonen, "Self-organizing maps and learning vector quantization for feature sequences", Neural Processing Letters, 10 (2) , 1999, pp. 151–159.

[9] G. de A. Barreto and A. F. R. Araújo, "Time in self-organizing maps: An overview of models", International Journal of Computer Research, vol. 10, n 02, 2001, pp. 139-179.

[10] G. J. Chappell and J. G. Taylor, "The temporal kohonen map", Neural Networks, Vol. 6, 1993, pp.441-445.

[11] M. Varsta, J. R. Millan and J. Heikkonen , "A recurrent self-organizing map for temporal sequence processing", International Conference on Artificial Neural Networks — ICANN'97, vol. 1327, 1997, pp. 421-426.

[12] M. Strickert and B. Hammer, "Merge SOM for temporal data", Neurocomputing, vol. 64, 2005, pp. 39-71.

[13] P. Ladefoged, Features and parameters for different purposes, UCLA Working Papers in Phonetics, n. 104, 2005, pp. 1-13.

[14] N. Chomsky and M. Halle, The Sound Pattern of English. New York: Harper and Row, 1968.

[15] G. Clements and E. Hume, "The internal Organization of Speech Sounds", In The Handbook of Phonological Theory, J. Goldsmith ed.. Cambridge, Mass.: Blackwell, 1995, pp. 245-306.

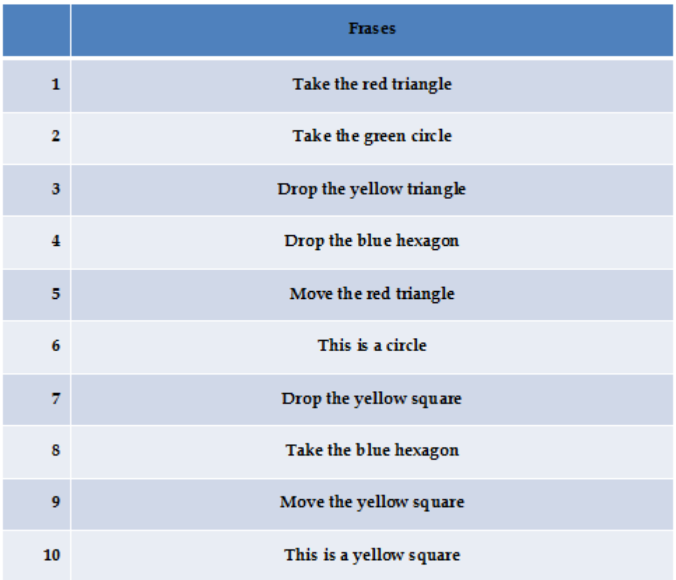
[16] P. Ladefoged, "Articulatory features for describing lexical distinctions", Language, vol. 83, n. 1, 2007, pp. 161-180.

[17] A. F. R. Araújo, H. F. Bassani and R. F. Pacheco, "Occurrence of false memories: A neural module considering context for memorization of words lists", in Proceedings of the 2010 IEEE International Joint Conference on Neural Networks, 2010, pp. 3115-3122.

[18] G. E. Hinton and T. Shallice, "Lesioning an attractor network: investigations of acquired dyslexia", Psychological Review, vol. 98, n. 1, 1991, pp. 74–95.

APPENDIX

TABLE IV. PHRASES USED FOR TRAINING.



	Frases
1	Take the red triangle
2	Take the green circle
3	Drop the yellow triangle
4	Drop the blue hexagon
5	Move the red triangle
6	This is a circle
7	Drop the yellow square
8	Take the blue hexagon
9	Move the yellow square
10	This is a yellow square

TABLE V. VOWEL PHONEMES MAPPED.

AA	AE	AH	AO	AW	AX
AY	EH	ER	IX	IH	IY
OY	OW	EY	UH	UW	JU

TABLE VI. CONSONANT PHONEMES MAPPED.

B	CH	D	DH	F	G
H	JH	K	L	M	N
NG	P	R	S	SH	T
TH	V	W	Y	Z	ZH