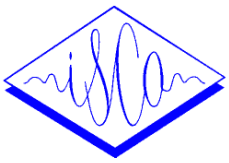

2nd International Conference on
Spoken Language Processing (ICSLP 92)
Banff, Alberta, Canada
October 12-16, 1992

SELF-ORGANIZING MAP WITH SUPERVISION FOR SPEECH RECOGNITION

Franck Poirier

TELECOM Paris
Département Signal
46 rue Barrault
75634 Paris Cedex 13

ABSTRACT

This paper proposes a new connectionist model of supervised learning called WYLINWYT map, based on Kohonen's Self Organizing Map (SOM). SOM has been early used in speech processing. It produces, by an unsupervised learning, a topological representation of the speech input space. The unsupervised learning mode is not fitted to obtain acceptable results in speech recognition. WYLINWYT shows a new learning scheme that improve speech recognition capabilities of SOM. WYLINWYT combines the advantages of a topological representation in the map space and the discriminating power of supervised learning. Phoneme recognition experiments are performed in a corpus of 200 phonetically balanced sentences. Compared to basic SOM, the proposed recognition method shows an improvement in the recognition accuracy of more than 5%.

10.21437/ICSLP.1992-114

1. INTRODUCTION

Acoustic-phonetic decoding is a central problem in speech recognition and is mainly a pattern-matching problem. In the past, many methods have been used such as Dynamic Time Warping (DTW), Hidden Markov Modeling (HMM) and more recently Artificial Neural Networks (ANN) [1]. These last methods are well-known for their ability of learning and their power of generalization on real-world data. They are more and more used in speech recognition.

ANN can be described as non-parametric models that learn some kind of internal representation of the input space from quite a lot examples. In other words, ANN find the hidden structure of the problem.

In speech processing, different models are applied, the most common are firstly feedforward networks such as Time Delay Neural Network (TDNN) or Radial Basis Function Network (RBF) and secondly competitive networks [2] such as Self-Organizing Map (SOM) or Learning Vector Quantizer (LVQ).

Only the SOM uses an unsupervised learning rule and is strongly biologically inspired. So, this model is particular, it self-organizes a map and produces a topological representation of the input space. In speech processing, SOM has early produced good results. In this field, SOM shows a hight discriminating power for vowels, but its performance deteriorates considerably for consonants. This degradation is caused on one hand by the nature of the speech sound (fast spectral variation) and on the other hand by the training mode (unsupervised learning).

2. SELF-ORGANIZING MAP

From the pattern recognition point of view, SOM is a vector quantizer (VQ). Unlike the other VQ, SOM organizes the codewords in a topological structure called map. In other words, SOM allows to approximate the distribution of the training vectors in an orderly fashion.

For coding or quantization problems, this characteristic is a great advantage. Unfortunately, for classification tasks, the SOM algorithm does not guaranty that the rate of missclassification is minimized.

To improve the recognition rate, different tracks are possible.

Firstly, the units of the map or codewords can be fine tuned using the Learning Vector Quantization (LVQ) algorithm [3], based on supervised learning.

Secondly, different versions of LVQ (LVQ1, LVQ2, LVQ3) can be used instead of the SOM method. In that case, a clustering

algorithm, such as the k-means method, is useful for inititializing the value of the codewords.

Thirdly, we have proposed an incremental version of LVQ called DVQ for Dynamic Vector Quantization [4] that solves the prolem of initialization and adapts the number of units to the intrinsic complexity of the training set . Moreover, DVQ improves the ability of generalization, decreases the number of codebook vectors, trains faster and reaches better performances than LVQ2.

All these approaches have the same drastic consequence: the topological organization of the map is lost after adaptation of the units by LVQ or DVQ.

In this paper, the main idea is to combine the advantages of a topological representation in the map space and the discriminating power of supervised learning.

3. WYLINWYT MAP

The new supervised learning paradigm for Self-organizing map involves some modifications.

Basic SOM consists of only two layers. An input layer and a clustering layer often called Kohonen layer or map layer. There is no output layer as for feedforward networks.

So, in the WYLINWYT map, supervision must be applied to the input layer. The input is therefore divided into two parts, firstly the feature vector called input vector and secondly the supervision vector called output vector which provides the corresponding desired output. The whole vector is called extended input vector and denoted E. In our speech experiment, the input vector is an n-dimension Mel cepstrum coefficients vector. The output vector is a p-dimension vector for a p-class problem. Each node of the map is associated with a n+p dimension weight vector.

Both learning scheme (Fig. 1) and recognition scheme (Fig. 2) have been modified. The learning phase is explained first, and then the recognition phase.

3.1 Learning phase

Learning consists of two different periods:

1. adaptation of the weight vectors by the standard learning rule,
2. adaptation of the weight vectors by the LVQ rule.

The first period corresponds to the initial formation of the map. It situates the nodes in the correct order. As for basic SOM, at the end, the map is quasi-organized. The n+p weights are adapted for the nodes that lie within the neighborhood of the winning node. The only difference is that unlike n feature parameters, n+p parametters are applied to the input layer. So, n+p weights are adapted for the nodes that lie within the neighborhood of the image node.

The second period corresponds to the final convergence of the map. The statistical distribution of the nodes approaches the statistical distribution of the input vectors. As the map is quasi-organized, the p parameters of the weight vector are significant. They bring a class information. So, it is sensible to replace the standard learning rule by a supervised learning rule. We have chosen the LVQ rule.

For both periods, the n+p parameters are processed in the same way. From the algorithm point of view, there is no distinction between the feature parameters and the class parameters.

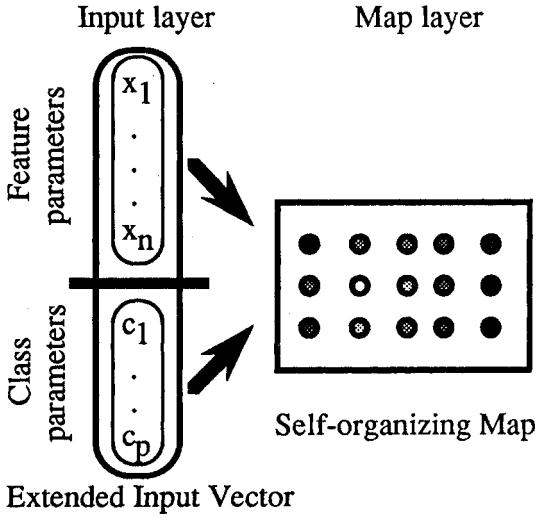

Fig. 1 Training phase

3.2 Recognition phase

During the recognition phase, only the feature vector $X=[x_1,\ldots,x_n]^T$ is presented to the map. A competition is organized between all the nodes of the map and the most active node, exactly the closest node to the feature vector X, wins. The response of the network is the output vector of the winning node. The decision rule is the following: an input pattern X is classified in the i^{th} class, if the i^{th} component of the output vector of the winning node (the $n+i^{th}$

component of the weight vector) has the maximum value.

Compared to the training phase, the input pattern in the recognition or test phase can be seen as an incomplete pattern, with the network recalling the missing information about the phonetic class of the pattern. What the network tests (the feature vector) is not the same as what the network learns (the extended input vector), so our model is called WYLINWYT for What You Learn Is Not What You Test.

Contrary to the back-propagation classifier where the network is trained to minimize the mean-squared error, the nodes of the clustering layer are self-organized on the basis of the correlation between the feature vector and the output vector. During the training phase, the map succeeds in associating the feature parameters with the supervision parameters.

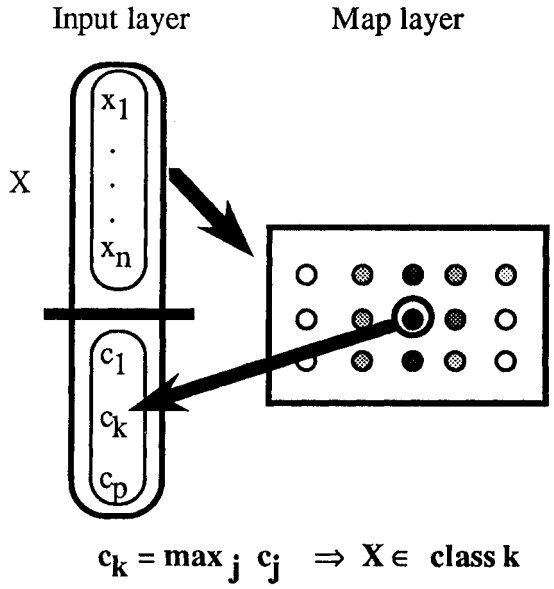

Fig. 2 Test phase

Contrary to SOM, no calibration phase is required. The map is implicitly labeled because the weight vector of each node is composed of p weights that show the distance between a node and the p classes. In our experiment, nodes are labeled into phonetic classes according to the maximum value of the output weight coefficients c_i.

3.3 Class information coding

During the learning process, for each input pattern $X=[x_1,\ldots,x_n]^T$, a class vector $c=[c_1,\ldots,c_p]^T$ is added. If X belongs to class k, class coefficients are of the form:

$$c_k = K_\mu . \|\mu\|$$
$$c_i = 0 \quad \text{for } i \neq k$$

with μ the mean vector of all X in the training database.

If $K_\mu = 0$, the proposed model is equivalent to the basic SOM. The value of K_μ is not critical in a large interval.

Compared to the basic SOM, the map is better organized, i. e. the area for each label is more continuous and compact. Unlike with the basic SOM, each class is guaranted to be associated with some nodes.

4. ALGORITHM

Let be denoted $E=[x_1,\ldots,x_n,c_1,\ldots,c_p]^T \in \Re^{n+p}$ the extended input vector, $X=[x_1,\ldots,x_n]^T$ the input vector, w_k a n+p dimensional extended weigt wector assigned to node k, i the index of the nearest nodes, $N_k(t)$ a topological neighbourhood from node k, $\varepsilon(t)$ a gain function, C_k the class associated to the node k.

Learning period 1:
$T_{end-1} < T_{end-2}$
t=0
initialize extended weights w
while $t < T_{end-1}$
  t=t+1
  present extended input vector E(t)
  update extended weight vectors
    for $k \in N_i(t)$
      $w_k(t+1) = w_k(t) + \varepsilon(t)\,(E(t) - w_k(t))$

Learning period 2
while $t < T_{end-2}$
  t=t+1
  present extended input vector E(t)
  update extended weight vectors
    for $k \in N_i(t)$
      if $X(t) \in C_k$
        $w_k(t+1) = w_k(t) + \varepsilon(t)\,(E(t) - w_k(t))$
      else
        $w_k(t+1) = w_k(t) - \varepsilon(t)\,(E(t) - w_k(t))$

Map calibration:
the weight vector $w = [x'_1,\ldots,x'_n,c'_1,\ldots,c'_p]^T$ is a reference of class k if $c'_k = \max_j c'_j$.

5. SPEECH DATABASE

The corpus consists of 200 french phonetically balanced sentences uttered by a male speaker. It contains 5270 phonemes, each one was labeled by hand at the center. The corpus is halfed between a training set and a test set.

The speech waveform is sampled at 10 kHz. Every 10 ms, using 33 ms overlapping Hamming window, an 8-dimension MFCC vector is computed. Each phoneme is represented by 3 frames, one frame on the left and one frame on the right of the label, which corresponds to 24 coefficients (3*8).

It can be noticed that a phonetically balanced corpus is not well fitted for learning. In fact, some phonemes are very little frequent (/ɲ/, /g/, /ɳ/ ...), for that reason it is far from obvious that they are correctly learnt. So, it would be better to modify the training set in order to increase the occurence of some phonemes. A minimal number of those in the training set is required to adequately learn some of their references .

6. EXPERIMENTS AND RESULTS

Table 1 shows the recognition rate on the training and test sets for all the experiments.

Method	Pho-nemes	Learning base (%)	Test base (%)	Number of nodes
SOM	V	79	72	100
	C	73	65	196
	V+C	58	57	400
WY LIN WYT	V	83	**77**	**100**
	C	71	**70**	**196**
	V+C	60	**60**	**400**

Table 1. Results on phonemes

SOM and WYLINWYT map are compared. For both methods, same number of nodes is used.

On 27 phonemes (V+C), 10 vowels and 17 consonants for 2348 examples in the training set, SOM performs 3% worst. It can be noticed that the energy coefficient or the temporal derivative of the MFCC are not used. So a number of errors occurs between consonants and vowels. It is more interesting to classify separately vowels and consonants.

On 10 vowels (V), for 1074 examples in the training set, WYLINWYT map shows an improvement of 5% on the test set.

On 17 consonants (C), for 1274 examples in the training set, the difference between both algorithms is the same (5%).

For all the experiments, the difference between the recognition rate on the learning set and the test set is fewer with the WYLINWYT map. In other words, this method induces better generalization.

7. CONCLUSION

WYLINWYT map shows, on the test set, an improvement in the recognition accuracy of about 5%. On the same speech database for the preliminary results, the Time Delay Neural Network (TDNN) method [5] and the DVQ method [6] yield a slightly higher performance. Yet, both last methods don't produce a topological representation.

Acknowledgement

We would like to thank F. Bimbot who made available to us the phonetically-balanced sentences database.

References

[1] J. Hertz, A. Krogh, R. Palmer. Introduction to the theory of neural computation. Addison-Wesley, 1991.

[2] T. Kohonen. Self-Organization and Associative Memory. Third Edition. Springer Verlag, Series in Information Sciences, 1989.

[3] T. Kohonen. The "Neural" Phonetic Typewriter. Computer, vol. 22, March 1988.

[4] F. Poirier et A. Ferrieux. DVQ: Dynamic Vector Quantization - An incremental LVQ. ICANN, Helsinki, 1991.

[5] S. Midenet. Modèle connexionniste d'apprentissage d'associations par carte auto-organisatrice. Contribution à l'étude d'une mise en correspondance des représentations connexionnistes et symboliques. Doctorat of Telecom Paris, 1991.

[6] F. Poirier. DVQ: Dynamic Vector Quantization. Application to Speech Processing. Eurospeech 91, Genova.