Feature Maps for Non-supervised Classification of Low-Uniform Patterns of Handwritten Letters

Pilar Gómez-Gil, Guillermo de-los-Santos-Torres, and Manuel Ramírez-Cortés

Department of Computer Science and CENTIA
Universidad de las Américas, Puebla
Santa Catarina M. Cholula Puebla, 72820, México
pgomez@mail.udlap.mx

Abstract. When input data is noisy and with a lack of uniformity, classification is a very difficult problem, because decision regions are hard to define in an optimal way. This is the case of recognition of old handwritten manuscript characters, where patterns of the same class may be very different from each other, and patterns of different classes may be similar in terms of Euclidian distances between their feature vectors. In this paper we present the results obtained when a non-supervised method is used to create feature maps of possible classes in handwriting letters. The prototypes generated in the map present a topological relationship; therefore similar prototypes are near each other. This organization helps to solve the problem of variance in the patterns, allowing a better classification when compared with other supervised classification method, a nearest-neighbor algorithm. The feature map was built using a Self-organized Feature Map (SOFM) neural network.

1 Introduction

Today the use of OCR´s is very common for printed documents. However, the automatic transcription of handwritten documents is still a challenge. In the other hand, there is a huge amount of old handwritten documents with valuable information that need to be digitized and translated in order to preserve them and make them available to a large community of historians. Currently this task is made by expert humans, making it expensive and slow.

The problems found when building an OCR that work for handwritten old documents are several: cleaning the digitized image, segmentation of words, segmentation of characters, recognition of characters and recognition of words. In this paper only the problem of recognition of handwritten characters is addressed, looking it as a classification of ill-defined patterns. We understand by ill-defined patterns those that:

- do not show an evident prototype to represent each class,
- the variance among all members of one class is greater than a threshold,
- a metric of similarity over the patters, such as Euclidian distance, may be greater among members of two different classes than members of the same class,
- noise in the patterns is high.

It is clear that these characteristics make the solution space of classification difficult to be defined using any classifier. A classifier for this type of patterns should be able to represent, in some way, the ambiguity imbedded in the training data. One possibility is that the classifier defines by itself the prototypes and number of classes.

A. Sanfeliu et al. (Eds.): CIARP 2004, LNCS 3287, pp. 203–207, 2004.

2 SOFM Network

There is evidence that cerebral cortex of the human brain is organized in computational maps [1]. This organization provides, among other benefits, efficient information processing and simplicity of access to processed information. Inspired on this property, in 1982 T. Kohonen developed the self organizing feature mapping algorithm (SOFM) [2]. The goal of SOFM algorithm is to store a set of input patterns $\mathbf{x} \in X$ by finding a set of prototypes $\{\mathbf{w}j \mid j = 1, 2\ldots N\}$ that represent the best feature map $\mathbf{\Phi}$, following some topological fashion. The map is formed by the weights connection $\mathbf{w}_j$ of a one or two-dimensional lattice of neurons, where the neurons are also related each other in a competitive way.

This learning process is stochastic and off-line; that is, two possible stages are distinguished for the net: learning and evaluation. It is important to notice that the success of map forming is highly dependent on the learning parameters and the neighborhood function defined in the model. The map is defined by the weights connecting the output neurons to the input neurons. The learning SOFM algorithm is: [1]

1. Initialize the weights with random values:

$$\mathbf{w}_j(0) = \text{random}() \ , \ j = 1..N \text{ (number of neurons)} \tag{1}$$

2. Choose randomly a pattern $\mathbf{x}(t)$ from the training set X at iteration t.
3. For each neuron i in the map feature map $\mathbf{\Phi}$ calculate the similarity among its corresponding weight set $\mathbf{w}_i$ and $\mathbf{x}$. The Euclidian distance may be used:

$$d^2(\mathbf{w}_i, \mathbf{x}) = \sum_{k=1}^{n} \left(w_{ik} - x_k\right)^2 \quad i = 1..N \tag{2}$$

4. Find a wining neuron i^* which is the one with maximum similarity (minimum distance).
5. Update the weights of winning neuron i^* and their neighbors as:

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(t)\left(\mathbf{x}(t) - \mathbf{w}_i(t)\right) \text{ for } j \in \Lambda_{i^*}(t) \tag{3}$$

Where $\Lambda_{i^*}(t)$ corresponds to a neighborhood function centered on the winning neuron. For this problem, we choose a neighborhood distance of 0 neurons. $\alpha(t)$ is a learning rate function depending on time. We choose: $\alpha(t) = 1/t$.

6. Go to step 2 until no more changes in the feature map are observed or a maximum number of iterations is reached.

3 Characteristics of the Patterns

This algorithm was tested with an ill-defined set of patterns: manuscript characters extracted from a collection of telegrams written by a famous person, Gral. Porfirio Díaz, at the beginning of 20th century in México. Figure 1 shows an example of one of these telegrams. Figure 2 shows some examples of words taken from such documents. Notice that the same class of letter looks very different when it appears in different positions of the word or in different words, as in the first and third word in the figure. Also notice that different classes of letters may look very similar.

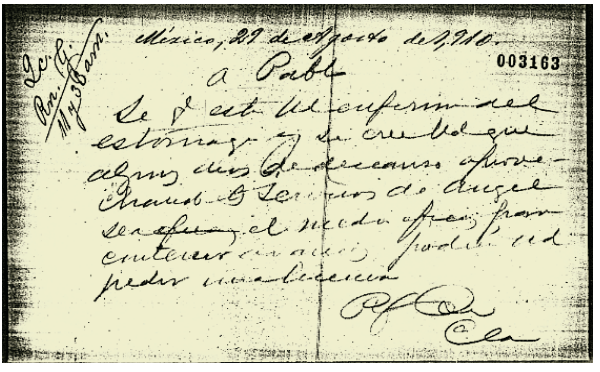

Fig. 1. A telegram written at the beginning of 20th century by Porfirio Díaz.

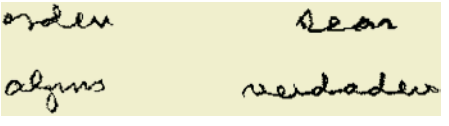

Fig. 2. Some examples of manuscript words written by Díaz.

To build the data set, words were manually extracted from the telegrams and their characters segmented also by a person. After that, character images were represented as bit maps, normalized with respect to a fixed number of rows and columns, and cleaned out of any blank rows or columns in their binary representation. The resulting patterns have 12 rows and 18 columns giving a total of 216 pixels. Figure 3 shows an example of the output of the software used to generate the patterns.

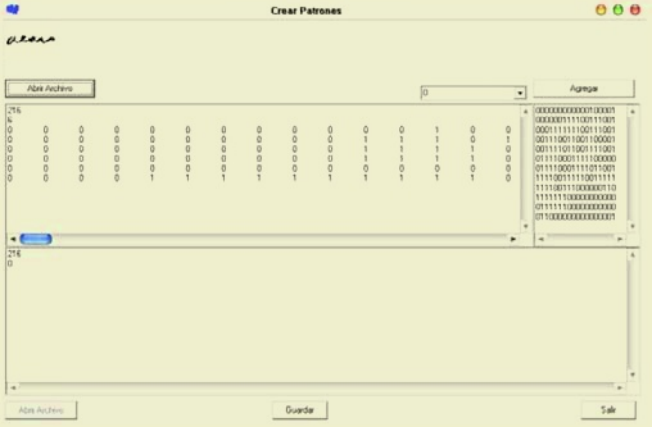

Fig. 3. Building patters from .gif files. In the top left the original word is shown; the map bit at the right shows the normalized pattern of the first character.

4 Building Feature Maps for Manuscript Letters

We used a 2-dimensional SOFM network. The formation of a feature map in a SOFM is stochastic, therefore several trails of learning using different sets of initial weights,

with different topologies in the output layer (feature map) need to be tested in order to find good results. Up today, there is no a formal way to find the best architecture of a network for a specific problem.

To analyze the behavior of SOFM network for the problem described in this article, we followed the next strategies:

1. We created several sub-sets of the problem, going from a simple problem with only 3 classes (instead of whole alphabet) up to 21 classes. It must be pointed out that, for the time of implementation of these experiments, we did not count with enough processed data to test the 27 classes forming the Spanish alphabet.
2. We compared the results obtained by SOFM network with a supervised classification algorithm, the nearest-neighbor, using a K-means algorithm to find the prototypes required, both as described at [3].

Table 1 shows a summary of the best results found during our experiments for 3, 5, and 21 classes. The numbers in parenthesis following the word "Kohonen" at column 3 represent the topology of the feature map tested in that case. For example (2x30) represents a feature map with 2 rows and 30 columns.

Table 1. Some results of Classification

Number of Classes	Number of training patterns	Type of Recognizer	Recognition rate
3	13	Nearest neighbor	84%
		Kohonen (3x3)	**92%**
5	56	Nearest neighbor	58%
		Kohonen (5x1)	58%
		Kohonen (5x2)	71%
		Kohonen (5x5)	**73%**
21	86	Nearest neighbor	6%
		Kohonen (5x12)	63%
		Kohonen (2x30)	**70%**

Notice that for all cases, particularly with the most difficult case (21 classes), SOFM performs much better than Nearest algorithm. Figures 4 and 5 show the 2 feature maps formed for the two Kohonen experiments performed with 21 classes. As expected, both maps follow a topological order, locating similar prototypes near each other. More details may be found at [4].

Fig. 4. Feature map with 5 rows and 12 columns generated for 21 classes.

Fig. 5. Feature map with 2 rows and 20 columns generated for 21 classes.

5 Conclusions and Perspectives

The advantages on the use of non-supervised generation of feature maps for an ill-defined set of patterns have been shown in these results. The behavior of SOFM network for the experiments executed in this research was as expected, creating well defined prototypes, and being able to classify better than a popular supervised algorithm. Future research work may include the determination of heuristics to help with the definition of the best topology of the feature maps, and the inclusion of weights in the feature maps marking the importance of each feature prototype, based on the frequency of patterns in the training set.

References

1. Haykin S.: Neural Networks: a Comprehensive Foundation. Macmillan College Publishing Company. New York. (1994).
2. Kohonen, T.: Self-Organized formation of topologically correct feature maps. Biological Cybernetics, 43, (1982) 59-69.
3. Tao, J.T. and Gonzalez, R.C. Pattern Recognition Principles. Addison-Wesley (1974)
4. De-los-Santos-Torres, G.: Reconocedor de Caracteres Manuscritos. Master thesis. Departamento de Ingeniería en Sistemas Computacionales. Universidad de las Américas, Puebla. (2003).