Global Journal of Computer Science and Technology
Volume 11 Issue 15 Version 1.0 September 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Recognition of Tifinagh Characters Using Self Organizing Map And Fuzzy K-Nearest Neighbor

By S. Gounane, M. Fakir, B. Bouikhalene
FST SMS University Beni Mellal, Morocco

Abstract - In this paper we present a comparison between SOM (Self-Organization Map) neural network and Fuzzy K-nearest Neighbor algorithms and their application to handwriting Tifinagh character recognition. The Box approach proposed in [1] is used for features extraction. Experimentation is carried out on a limited database of nearly 200 samples. The results showed that Fuzzy K-Nearest Neighbor had a very good performance.

Keywords : Tifinagh characters, Fuzzy k-Nearest Neighbor, Self Organizing Map, Fuzzy k-means, Features extraction.

GJCST Classification : I.2.3, F.1.1

Recognition of Tifinagh Characters Using Self Organizing Map And Fuzzy K-Nearest Neighbor

S. Gounane[α], M. Fakir[Ω], B. Bouikhalene[β]

Abstract - In this paper we present a comparison between SOM (Self-Organization Map) neural network and Fuzzy K-nearest Neighbor algorithms and their application to handwriting Tifinagh character recognition. The Box approach proposed in [1] is used for features extraction. Experimentation is carried out on a limited database of nearly 200 samples. The results showed that Fuzzy K-Nearest Neighbor had a very good performance.

Keywords : Tifinagh characters, Fuzzy k - Nearest Neig - hbor, Self Organizing Map, Fuzzy k-means, Features Features extraction.

I. Introduction

The recognition of characters from scanned images of documents has been a problem that has received much attention in the fields of image processing, pattern recognition and artificial intelligence.

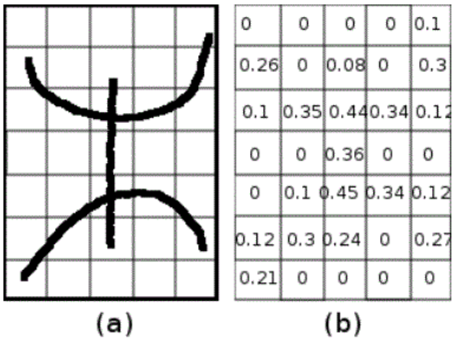
For many years, fuzzy logic and Artificial Neural Networks have been used in a wide range of problem domains: process control (where Fuzzy Controllers have been very popular), management and decision making, operations research, economics and pattern recognition and classification.

This paper presents an application of both SOM neural network and fuzzy k-Nearest Neighbor in recognition of handwritten Tifinagh characters. This paper is organized as follows: in section [1] a features extraction method, which is an essential step prior to pattern recognition, is described. Section [2] describes the architecture and the learning mechanism of the SOM neural networks. Section [3] presents the k-Nearest Neighbor algorithm. Section [4] gives results of the application of both SOM and Fuzzy k-NN on Tifinagh handwriting character recognition.

II. Features Extraction

Preprocessing techniques like thinning, slant correction and smoothening are applied. For extracting the features, the Box approach proposed in [1] is used here. This approach requires the spatial division of the character image. The major advantage of this approach stems from its robustness to variation, ease of implementation and high recognition rate. Each character image is divided into L × H boxes so that the portions of character will be in some of these boxes. There could be boxes that are empty (Figure 1(a)). The choice of number of boxes is arrived at by experimentation. Elements of a Normalized Vector that describe the character is obtained by dividing the number of all black pixels present in this box by their total number for each box, (Figure 1 (b)).

One can easily see that this characterization is invariant of the character image dimensions. Hence an image of whatever size gets transformed into a vector of L × H predetermined dimensions.



0	0	0	0	0.1
0.26	0	0.08	0	0.3
0.1	0.35	0.44	0.34	0.12
0	0	0.36	0	0
0	0.1	0.45	0.34	0.12
0.12	0.3	0.24	0	0.27
0.21	0	0	0	0

Figure 1 : Features extraction

III. Self Organizing Map (SOM)

The Self-Organizing Maps, abbreviated SOM, was developed by Professor Teuvo Kohonen in the early 1980s. Self-organizing maps are a special class of artificial neural network, because those are based on competitive learning and the learning itself is unsupervised.

Also SOM is considered as a special case in data-mining, it can be applied to both clustering and projecting the data onto a lower dimensional display at the same time.

In a SOM network, there is an input layer and an output layer which is usually designed as two-dimensional arrangement of neurons that maps n dimensional input to two dimensional (Figure 2). It is basically a competitive network with the characteristic of self-organization providing a topology-preserving mapping from the input space to the clusters.

Author [α] : Information Processing & Telecommunication Team, Dep. Of Computer Sciences – FST SMS University Beni Mellal, Morocco. E-mail : Gounane.said@gmail.com

Author [Ω] : Information Processing & Telecommunication Team, Dep. Of Computer Sciences – FST SMS University Beni Mellal, Morocco. E-mail : fakfad@yahoo.fr

Author [β] : Information Processing & Telecommunication Team – FP SMS University Beni Mellal, Morocco. E-mail : bouikhalene@yahoo.fr

The algorithm proceeds first by initializing the synaptic weights in the map for the neurons. This can be done by assigning them small values picked from a random number generator; in so doing no prior order is imposed on the feature map. Once the map has been properly initialized, there are three essential processes involved in the formation of the self-organizing map: Competition, cooperation and synaptic adaptation.

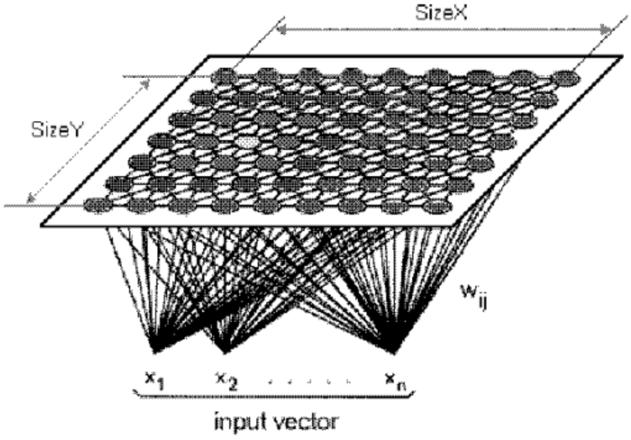


Figure 2 : SOM architecture

a) Competition

Let $X = [x_1, \ldots, x_n]$ be the input pattern and $W_j = [w_{j1}, \ldots, w_{jn}]$ the synaptic weight vector of each neuron j in the map. To find the best match of the input vector X with the synaptic weight vectors W_j, compare the inner products $W_j^T.X$ for j=1,2,…,l (l is the number of output neurons) and select the largest denoted $i(X)$. Maximizing $W_j^T.\ X$ is equivalent to minimizing $W_j - X$ [2] then one can have:

$$\underset{j}{i(X)} = \arg\min\left(W_j - X\right) \tag{1}$$

The neuron number $i(X)$ is called the winning neuron.

b) Cooperation

In neurobiology, a neuron that is firing tends to excite neurons in its immediate neighborhood more than those farther away. The same with output neurons in the SOM, a topological neighborhood around the winning neuron i is made and it decay smoothly with lateral distance. Another unique feature of the SOM algorithm is that the size of the topological neighborhood shrinks with time. Let $h_{j,i}(k)$ denote the topological neighborhood at time k, centered on the winning neuron i, and j denote a typical neuron of a set of excited (cooperating) neurons around winning neuron i. On can assume that the topological neighborhood $h_{j,i}$ is unimodal function of the lateral distance $d_{i,j}$, such that it satisfies:

1. $h_{j,i}$ is symmetric and attains its maximum value at the winning neuron i for which $d_{j,i} = 0$.
2. The amplitude of $h_{j,i}$ decreases monotonically with increasing $d_{i,j}$, decaying to zero for $d_{i,j} \to \infty$.

A typical choice of $h_{j,i}$ is the Gaussian function

$$h_{j,i}(k) = e^{\left(\frac{d_{i,j}^2}{2\sigma^2(k)}\right)} \tag{2}$$

$$\sigma(k) = \sigma_0 e^{\left(-\frac{k}{\tau_1}\right)} \tag{3}$$

Where τ_1 is the time constant of the algorithm.

c) Synaptic adaptation

By definition, for the network to be self-organizing (and unsupervised), the synaptic weight vector W_j of neuron *j* in the network is required to change in relation to the input vector X. This change is expressed by the equation as follows:

$$W_j(k+1) = W_j(k) + \alpha(k) h_{j,i(X)}(k)\, X\left(-W_j(k)\right) \tag{4}$$

Where $\alpha(k)$ is the learning-rate.

The exact form of learning-rate is not important. It can be linear, exponential or inversely proportional. However it should be time varying. In particular, it should start at an initial value α_0, and then decrease gradually with increasing time n. This requirement can be satisfied by using an exponential learning-rate, as shown by:

$$\alpha(k) = \alpha_0 e^{\left(-\frac{k}{\tau_2}\right)} \tag{5}$$

Where τ_2 is another time constant of the SOM algorithm.

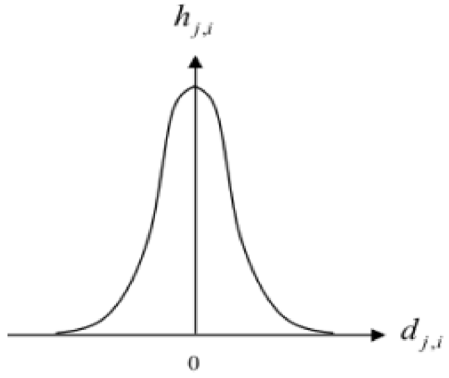


Figure 3 : example of topological neighbourhoods (Gaussian)

IV. Fuzzy K-Nearest Neighbor

Fuzzy kNN is part of supervised learning that has been used in many applications in the field of data mining, statistical pattern recognition, image processing and many others. Some successful applications are including recognition of handwriting and satellite image. Fuzzy K nearest neighbor algorithm is very simple. It works based on an unsupervised clustering algorithm like fuzzy k-means algorithm to determine prototypes representing clusters. Then the minimum distance from the query instance to these prototypes is used to determine the K-nearest neighbors. After and basing on membership function we take the neighbor with the maximum membership to be the prediction of the query instance.

a) Fuzzy k-means

The Fuzzy k-means clustering algorithm is an improvement of the k-means algorithms. K-Means are the simplest methods of clustering data. The fuzzy K-means algorithm uses a set of N unlabeled feature vectors and classifies them into k classes, where k is given by the user.

From these N feature vectors, k of them are randomly selected as initial seeds. The feature vectors are assigned to the closest seeds depending on its distance from it and on a given membership function. The mean of features belonging to a class is taken as the new center. The features are reassigned; this process is repeated until convergence.

In a fuzzy clustering, a pattern vector can belong to all clusters with different degrees given by a membership function [figure 4]. One can prove that such a function exists [5], and for each cluster C_i it is defined as follows:

$$f_i(X) = \frac{1/d^2\left(X,c_i\right)^{\frac{1}{m-1}}}{\sum_{i=1}^{N} 1/d^2(X,c_i)^{\frac{1}{m-1}}} \qquad m \in \Re, \quad m \geq 1 \qquad (6)$$

Where c_i is the center of the class C_i, and The parameter m is defined by the user to adjust the fuzziness of the clustering. The hard clustering case is obtained by taking m = 1.

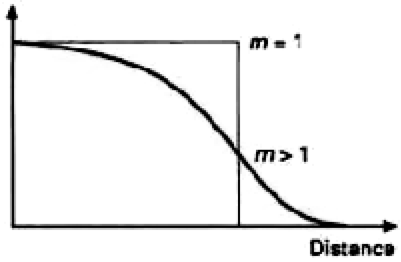


Figure 4 : Membership function

b) Fuzzy K-means algorithm

1. Choose randomly the k prototypes c_i .
2. Compute the degree of membership of all feature vectors X_j in all clusters C_i : $\mu_{ij} = f_i(X_j)$.
3. Compute new cluster prototypes as follows:

$$c_i = \frac{\sum_{j=1}^{N} \left(\mu_{ij}\right)^m x_j}{\sum_{j=1}^{N} \left(\mu_{ij}\right)^m} \qquad (7)$$

4. Iterate back and force between (2) and (3) until the memberships or cluster centers for successive iteration differ by more than some prescribed value ε.

c) Fuzzy K-Nearest Neighbor

The Fuzzy k-NN classifiers consist on proximity measures. They are ideally suited for modeling the non parametric distribution on handwritten word recognition data.

For a given character X, the fuzzy classifier computes the membership of X in different classes $C_1, C_2, \ldots\ldots C_N$. The character X is allocated to the class for which the membership function yields the maximum value. By a learning process (Fuzzy k-mean, k-mean, LVQ ...) we assign a number of prototypes P_j for each class C_i . After generating the k-Nearest Neighbor prototypes P_j for a character (by distance similarity), the degree of membership of the vector X to the class C_i : $\mu_i(X)$, can be calculated as follow:

$$\mu_i(X) = \frac{\sum_{i=1}^{k} \mu_{ij} / d^2\left(X,P_j\right)^{\frac{1}{m-1}}}{\sum_{i=1}^{N} 1/d^2\left(X,P_j\right)^{\frac{1}{m-1}}} \qquad (8)$$

Where μ_{ij} is the membership of the prototype P_j to the class C_i. To compute this value tow methods are proposed:

Using a crisp assignment of P_j to C_j :

$$\mu_{ij} = \begin{cases} 1 & P_j \in C_i \\ 0 & P_j \notin C_i \end{cases} \qquad (9)$$

Or by using the k-NN to determine k-nearest neighbor of P_j, then the number of neighbor belonging to the same class as P_j denoted n_j is used to compute μ_{ij} as follows:

$$\mu_{ij} = \begin{cases} 0.51 + 0.49 n_j / k & j = i \\ 0.49 n_j / k & j \neq i \end{cases} \qquad (10)$$

V. Results

A java application was developed in order to test and compare those two algorithms. To extract features each, character was divided into boxes, the input layer of the SOM neural network was made of 63 neurons and 35 for the output layer (number of character to recognize), for the fuzzy k-nearest neighbor the fuzziness factor m is equal to 2 and number of neighbors' k is equal to 3.

As there is no standard database available for handwritten Tifinagh characters, a database was created from samples of different writing styles with different sizes. The database also includes some complex samples that are even hard to recognize by careful inspection Figure 5: Samples used for training (figure 5). This database contains 200 samples (about 5 samples per character) divided into two disjoint sets, one for training both SOM and Fuzzy k-NN, and another for testing these two algorithms.

The features extraction method used here is scaling invariant, that's why there is always a recognition problem between Tifinagh character ⴰ and ⵔ, where ⴰ 's are mistaken as ⵔ 's and vice versa. Some other recognition problems are with and , ⵔ and ⵙ but not persistent as the first one.

Experiments shows that the best results are given by the Fuzzy k-NN where in many cases it success to recognize handwritten characters that SOM algorithm fails to (figure 6).

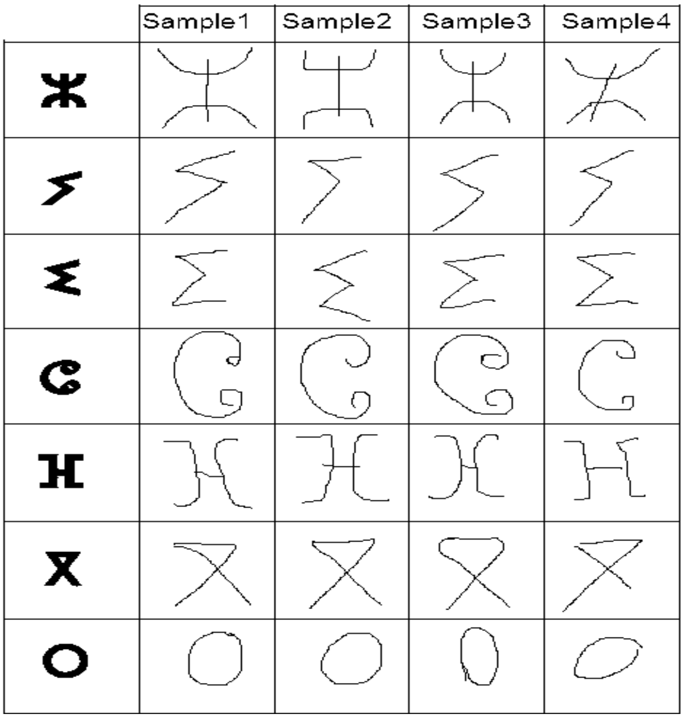
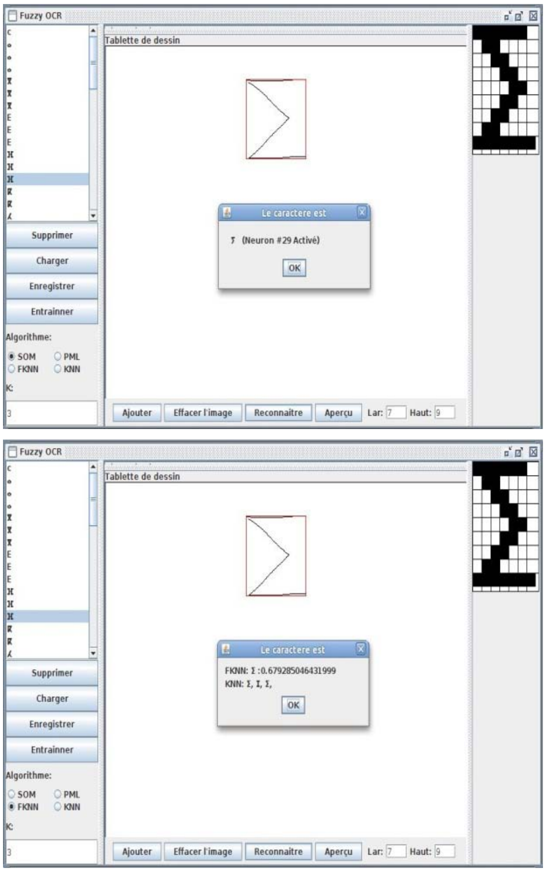


Figure 5 : Samples used for training



Fuzzy OCR

Tablette de dessin

Le caractere est

(Neuron #29 Activé)

OK

Algorithme: ● SOM ○ PML ○ FKNN ○ KNN

K: 3

Lar: 7 Haut: 9

Fuzzy OCR

Tablette de dessin

Le caractere est

FKNN: Σ :0.679285046431999

KNN: Σ, Σ, Σ,

OK

Algorithme: ○ SOM ○ PML ● FKNN ○ KNN

K: 3

Lar: 7 Haut: 9

Figure 6 : handwritten character. SOM algorithm (left) fails ( ) but not Fuzzy k-NN(right) ( )

VI. Conclusion

In this paper two clustering algorithms are presented, Self Organizing Map neural network and the Fuzzy k-Nearest Neighbor. We applied these two algorithms to Tifinagh character recognition. The box approach was used to extract features from the character image. Fuzzy k-NN was more robust and fast than the SOM.

References Références Referencias

1. M.Hanmandlu, A.V.Nath, A.C.Mishra, & V.K.Madasu. (2007). Fuzzy Model Based Recognition of Handwritten Hindi Numerals using bacterial foraging. 6th IEEE/ACIS ICIS.
2. Haykin, & Simon. (1999). Neural Networks: a comprehensive fondation. New jersey: Prentice-hall inc.
3. A.Mingoti, S., & Lima, J. O. (2005). Comparing SOM neural network with Fuzzy c-means, K-means and traditional hierarchical clustering algorithms. European Journal of Operational Research.
4. D.Slezak, B.Kang-Junzhong, T.Kim, & G.Kuroda. Segnal processing, Image processing and Pattern recognition. New York: Springer.
5. J.Sagau. (n.d.). Logique flou en classification. Techniques de l'Ingénieur , H3638.

6. (J.C.), & BEZDEK. (1981). Pattern recognition with fuzzy objective function algorithms. Plenum Press.
7. A.Tellaeche, Burgos-Artizzu, X. P., G.Pajares, & A.Ribeiro. (2007). On Combining Support Vector Machines and Fuzzy K-Means in Vision-based Precision Agriculture. World Academy of Science, Engineering and Technology.
8. F.Zheng. (n.d.). K-Means-based fuzzy classifier.
9. Nasser, S., Alkhaldi, R., & Vert, G. (n.d.). A Modified Fuzzy K-means Clustering using Expectation Maximization.
10. Nassery, P., & Faez, K. (n.d.). Signature pattern recognition using psudo Zernike moments and fuzzy logic classifier.
11. O.Matan, R.K.Kiang, C.E.Stenard, & B.Boser. (1990). Handwritten character recognition using neural network architectures. 4th USPS advanced Technology Conference, Washington D.C, 1003-1011.
12. S.Nasser, R.Alkhaldi, & G.Vert. (n.d.). A Modified Fuzzy K-means clustering using expectation Maximization.
13. S.Theodoridis, & Koutroumbas, K. (2009). Pattern recognation (fort edition). Elsevier.
14. V.Ganapathy, & Liew, K. L. (2008). Handwritten Character Recognition Using Multiscale Neural.

This page is intentionally left blank