Unsupervised Pattern Classification by means of Cellular Neural Networks

G. COSTANTINI, D. CASALI, M. CAROTA
Department of Electronic Engineering
University of Rome "Tor Vergata"
Viale Politecnico, 1. I-00133 Rome
ITALY

Abstract: - A novel algorithm for unsupervised classification of datasets made up of integer valued patterns by means of Cellular Neural Network (CNN) is proposed. The adopted CNN is n-dimensional and is based on a space-variant template - neighborhood order 1 - to cluster n-dimensional datasets. The choice of a CNN architecture allows a straightforward hardware implementation, particularly suited for bi-dimensional patterns.

Key-Words: - Cellular Neural Networks, Clustering, Pattern Classification.

1 Introduction

Data mining is one of the most successful applications of neural networks. It is also a powerful tool to find patterns and implicit relationships in datasets, as well as to infer rules to be used in a prediction task. This job can be carried out by somehow partitioning the dataset in two or more subsets called clusters.

Nevertheless, from a computational point of view, clustering algorithms are sometimes very onerous; hence, a hardware implementation is desirable.

CNN [1,2] are known to be an excellent example of very powerful and general-purpose networks that can be implemented in hardware [3]. Therefore, a clustering algorithm that can run on a CNN would be very helpful, just because of the real-time capabilities of the CNN chips. Other CNN based clustering algorithms have been proposed in literature, as in [4] and [5], but with respect to binary valued patterns only.

Our algorithm is based on an approach that differs both from [4] and [5]. It can be used to firstly cluster a given dataset of integer valued patterns and then to associate a new pattern to the clusters previously extracted. The main drawback of this method is its limitation to a low number of dimensions, in case we would implement it in hardware. In a first step, we will take into account bi-dimensional patterns, that is patterns consisting of vectors of two numbers. Afterwards, we will extend the algorithm to patterns of dimension n > 2.

2 Overview of the method

A vector of two elements can be represented by a point in a bi-dimensional space. The whole database is a set of points characterized by a well-defined spatial density function. The local maxima of this function will be taken as labels that identify the clusters found out by the CNN algorithm.

The proposed unsupervised classification algorithm can be divided into three different phases:

1. Preliminary phase: the density function and its maximum points are calculated.

2. Clustering phase: a CNN template is built up and all the patterns of the dataset are presented to the network one after the other. The point corresponding to each of these patterns is let moving along the direction where the gradient of the density function is higher, until it reaches one of the local maxima. This maximum will label the cluster to which the pattern taken into account belongs to.

3. Classification phase: new patterns not included in the original dataset are presented to the CNN, in order to associate them to the clusters found during the previous phases.

3 Implementation of the method

The unsupervised classification algorithm presented in this paper is designed for being implemented on a CNN. As proposed by Chua [1], CNN state equation is the following:

$$\frac{dx_{ij}(t)}{dt} = -x_{ij}(t) + \sum_{C(k,l)\in N_1(i,j)} A^{ij}_{kl} y_{kl}(t) + \sum_{C(k,l)\in N_1(i,j)} B^{ij}_{kl} u_{kl}(t) + I$$

where:

$\mathbf{U} = [u_{ij}]$	is the input matrix
$\mathbf{X} = [x_{ij}]$	is the state matrix
$\mathbf{Y} = [y_{ij}]$	is the output matrix
$\mathbf{A^{ij}} = [A^{ij}{}_{kl}]$	is the feedback template for cell (i,j)
$\mathbf{B_{ij}} = [B^{ij}{}_{kl}]$	is the control template for cell (i,j)
I	is the bias current.

As regards clustering, we adopt a CNN with first order neighborhood, using space-invariant templates for phase 1 of the algorithm and space-variant templates for phase 2.

A preliminary operation is required. It consists of the superimposition of the whole dataset on the state of the CNN: if pattern (*i,j*) belongs to the dataset, the state of the cell (*i,j*) is set to 1, while the state of all the other cells is set to -1. An example of the image related to a given data set is shown in Fig. 1, in the case of a 50 X 50 CNN.

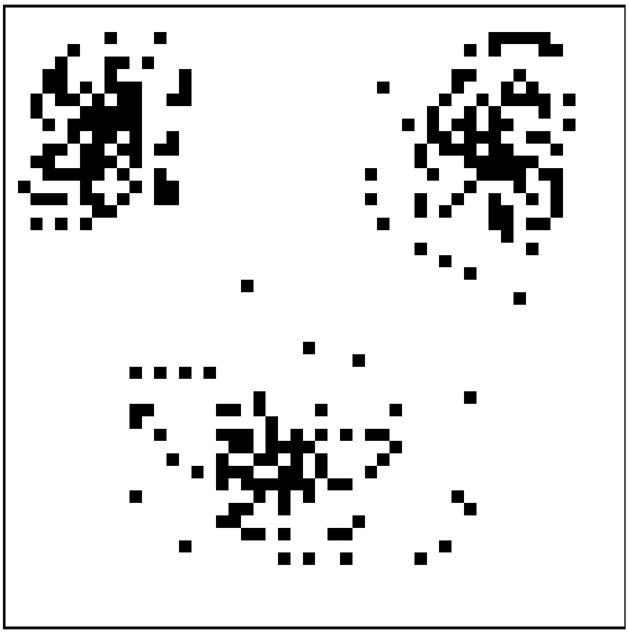

Fig. 1 - Example of a data set.

The evaluation of the density function can be simply accomplished by applying a heat-diffusion template [6]:

$$\mathbf{A}=\begin{bmatrix}0.1 & 0.15 & 0.1\\ 0.15 & 0 & 0.15\\ 0.1 & 0.15 & 0.1\end{bmatrix} \quad \mathbf{B}=\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix} \quad I=0$$

Input: $\mathbf{u}$ *= arbitrary*
Initial state: $\mathbf{x}(0)$ *= data set*
Output: $\mathbf{y}(\infty)$ *= density function*

Within a given time t_d, the state matrix becomes the image of the density function. In Fig. 2, this image, obtained after a time $t_d = 20\tau$, is depicted (τ is the time constant of the CNN). The higher is t_d, the smoother the density function will be, and a lower number of clusters will be found. To find the cluster labels as local maxima of the density function, we can make use of the following non linear template [6]:

$$\mathbf{A}=\begin{bmatrix}0 & 0 & 0\\ 0 & 3 & 0\\ 0 & 0 & 0\end{bmatrix} \quad \mathbf{B}=\begin{bmatrix}b & b & b\\ b & 0 & b\\ b & b & b\end{bmatrix} \quad I=-3.5$$

where:
$b = 0.5$ if $u_{ij} - u_{kl} \geq 0$
$b = 0$ otherwise

Input: $\mathbf{u}$ *= density function*
Initial state: $\mathbf{x}(0) = \mathbf{0}$
Output: $\mathbf{y}(\infty)$ *= local maxima*

When the network reaches the stability, black points (cells with output = 1) will identify the points of local maximum.

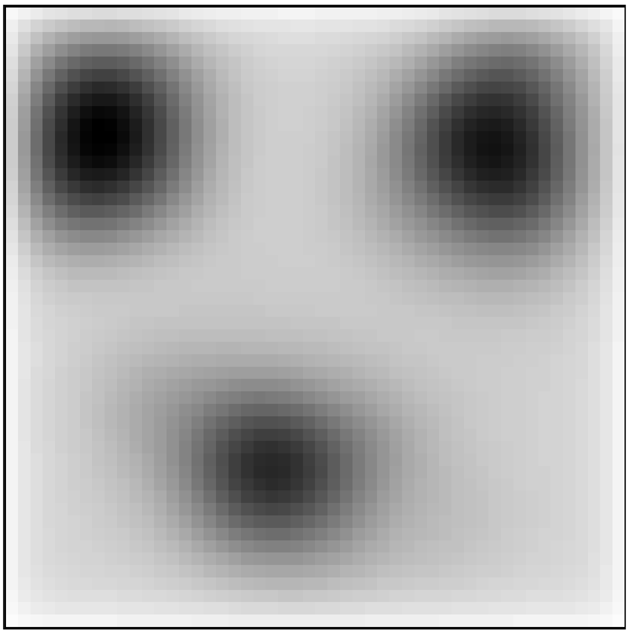

Fig. 2 - Density function of data set in Fig. 1.

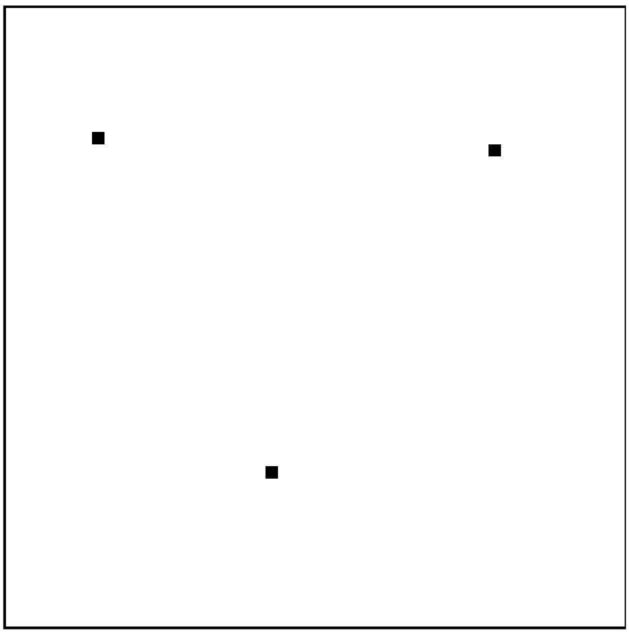

Fig. 3 - Maximum points of density function in Fig. 2.

In Fig. 3, the result of the application of this template to the density in Fig. 2 is shown. Three clusters, respectively with centres (11, 8), (12, 40) and (38, 22), can be identified.
The clustering phase can be implemented by constructing a space-variant template that drives any point superimposed on the state towards the direction of maximum gradient of the density function. This task can be carried out by deriving the space-variant template $\mathbf{A}^{ij}$ from the density function evaluated in the previous step and that at this point can be found at the state matrix **X**. The element A^{ij}_{kl} of the template $\mathbf{A}^{ij}$ is set to 1 if $X_{ij} > X_{i+k-2, j+l-2}$, or if (i,j) is a maximum; otherwise, $A^{ij}_{hl} = 0$. Cells at position $(I + k - 2, j + l - 2)$ are always located in the order-one neighbourhood of (i,j). Therefore, this operation is local and can be carried out in parallel, that is to say implemented in hardware. Actually, this template doesn't move the point towards the direction of maximum gradient only. As a matter of fact, it extends the point to a spot that covers all the cells with a positive gradient. Afterwards, the spot spreads out, moves toward the nearest maximum and finally collapses again to a point, that's nothing but the maximum point.

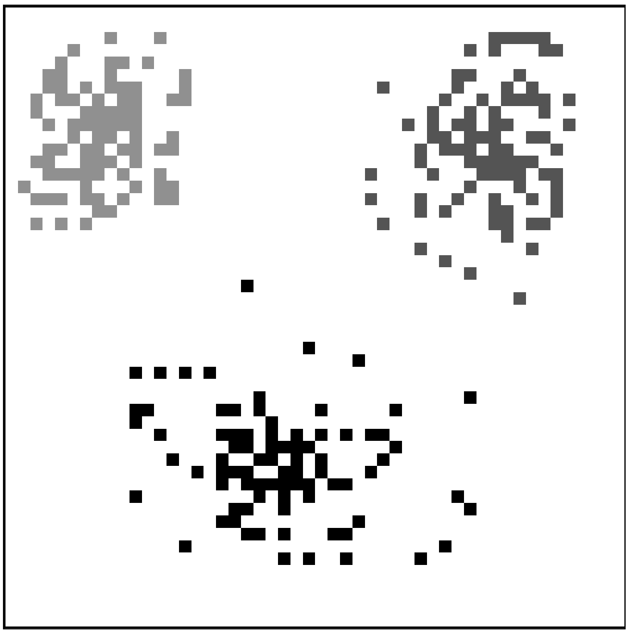

Fig. 4 - Data set in Fig. 1 clustered.

In Fig. 4 we show the same data set as in Fig. 1 after clustering: pixels of the data set are represented with a different grey level, according to the cluster in which they have been allocated.
Classification of new patterns can be achieved by simply presenting every pattern not found in the training data set at the network input. The network will evolve and, once the stability is achieved, the maximum point that labels one of the identified clusters will indicate the class to which the presented pattern belongs to.

4 Extension to n dimensions

To extend the CNN classification method presented in this paper to n-dimensional data sets, we can use an *n*-dimensional CNN.
For example, in three dimensions, the diffusion task can be accomplished by generalizing the approximation of the diffusion template:

$$\mathbf{A} = [[0.01\ 0.05\ 0.01;\ 0.05\ 0.07\ 0.05;\ 0.01\ 0.05\ 0.01]\ [0.05\ 0.07\ 0.05;\ 0.07\ {-0.1}\ 0.07;\ 0.05\ 0.07\ 0.05]\ [0.01\ 0.05\ 0.01;\ 0.05\ 0.07\ 0.05;\ 0.01\ 0.05\ 0.01]]$$

$\mathbf{B} = \mathbf{0}$ and $I = 0$.

The template for local maxima extraction can be generalized in the form $A_{222} = 2$ and $A_{klm} = 0$ for the remaining elements of **A,** $B_{222} = 0$ and $B_{klm} = b$ for the remaining elements of **B**, where $b = 0.5$ if $u_{ijh} - u_{klm} \geq 0$, $b = 0$ otherwise, I= -24.5. Finally, the space-variant template can be constructed in this way: element *(k,l,m)* of cell *(i,j,h),...* is set to 1, if $S_{ijh} > S_{i-k-2,i-l-2,h-m-2,...}$ or if (i,j,h) is a maximum of the density function.

5 Conclusion

A CNN based unsupervised classification algorithm is presented. The algorithm shares advantages and limits of CNN: a fast and simple hardware implementation, for two dimensions pattern, but a highly growing complexity when the number of dimensions increases.

References:

[1] Chua, L.O., and Yang, L.: Cellular neural networks: Theory, *IEEE Trans. Circuits and Systems*, Vol. 35, pp. 1257 -1272, 1988.

[2] Chua, L.O., and Yang, L., Cellular neural networks: applications, *IEEE Trans. on Circuits Systems*, 1988, vol. 35, pp. 1273-1290

[3] Linan, G., Espejo, S., Dominguez-Castro, R., and Rodriguez-Vazquez, A., The CNNUC3: an analog I/O 6464 CNN universal machine chip prototype with 7bit analog accuracy, Proc. of *IEEE International Workshop on Cellular Neural Networks and Their Applications Proceedings,* 2000.

[4] Bálya, D., CNN Universal Machine as classification platform: an art-like clustering algorithm, *International Journal of Neural Systems*, Vol. 13, No. 6, pp. 415-425, 2003.

[5] Grassi, G., and Di Sciascio, E., Learning Algorithm for Pattern Classification using Cellular Neural Networks, *Electronics Letters, IEE,* vol. 36, n. 23, pp. 1941-1943, 2000.

[6] Analogic CNN program library 6.1, *Analogical and Neural Computing Lab.*, Hungarian Academy of Sciences, Budapest, 1994