

A New Algorithm for Training Multi-layered Morphological Networks

Ricardo Barrón, Humberto Sossa, and Benjamín Cruz

Centro de Investigación en Computación-IPN
Av. Juan de Dios Bátiz esquina con Miguel Othón de Mendizábal
Mexico City, 07738, Mexico
rbarron@cic.ipn.mx, hsossa@cic.ipn.mx,
benjaminacruz@sagitario.cic.ipn.mx,

Abstract. In this work we present an algorithm for training an associative memory based on the so-called multi-layered morphological perceptron with maximal support neighborhoods. We compare the proposal with the original one by performing some experiments with real images. We show the superiority of the new one. We also give formal conditions for correct classification. We show that the proposal can be applied to the case of gray-level images and not only binary images.

Keywords: Associative memories, Morphological neural networks, maximal support neighborhoods.

1 Introduction

Neural networks have shown to be an excellent alternative to face problems where it is difficult to find no algorithmic solution. Based on the functioning of the human nervous system, lots of researchers have proposed different neural processing models. Probably the best know model is the Back-propagation Neural Network [1], [2] and [3].

The study of the internal structure of neural cells has revealed that all cells have the same simple structure, independently of their size and shape. Information from a cell voyages through the signals that neurons send to other neurons through their dendrites. It is believed that the cellular body adds up the received signals; when enough inputs are available a discharge is produced. Initially, this discharge occurs in cellular body, it then propagates between the axon until the synapses that sends a new signal to the other neurons.

An artificial neural network can be seen as a non-linear mapping between two pattern spaces: the input pattern set and the output pattern set. Normally the internal parameters of this mapping are determined by means of a training process and are denoted as synaptic weights.

In the decade of the 50's, Rosenblatt ([2], [3]) introduces the well-known perceptron, which is the classical model that has been used for most of the actual developments. However, in the 90's, Ritter et al., ([4], [5], [6]) and Sussner [12] presented a new kind of neural network model, the so-called morphological neural

network. Here, the classical operations of multiplication and addition are replaced by summations and **max** (or **min**), respectively. One difference between this model and the classical models is the computational cost when computing the value of i -th neuron at time $t + 1$ that in this model is less.

In the last years it has been found that, apparently the processing of information not only occurs at the cellular body but also at the dendrites [7]. This affirmation could give an explanation of the great efficiency of our nervous system, since processing of the information happens practically along the communication channel.

The material just presented along with the morphological paradigm is the departure point of this paper.

2 Associative Memory Based on the Morphological Perceptron

In [7] it is presented how a morphological perceptron allows classifying any compact set in the pattern's domain, which can be used to build an associative memory able to recall patterns affected by mixed noise. The idea is to build a three layer associative memory (an of input, one hidden, and of output) as can be appreciated in Figure 1.

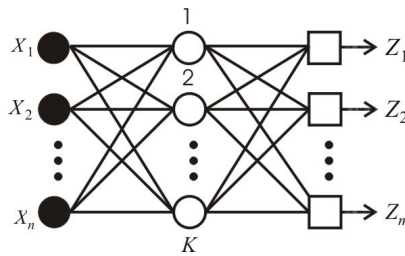


Fig. 1. Associative memory of three layers based on the morphological perceptron

First layer works as register of the number of elements of the input pattern. Second one (the hidden layer) is composed of morphological perceptrons [7], one for each class, while the output layer is formed by perceptrons based on the **max** operator with a linear gating function, one perceptron for each component of the output patterns.

The perceptrons of the hidden layer are morphological perceptrons with one input dendrite. The classified region is thus a hyper-rectangle in \mathbf{R}^n , whose goal is to classify the corresponding pattern inside the on-region. For the better functioning of this model, the perceptron outputs a zero in its on-region and $-\infty$ in its complements.

The output of a perceptron at the output layer is given as

$$z_i = \bigvee_{k=1}^K [y_i^k + \varphi_k(x)] \tag{1}$$

where y_i^k is the i -th component of the k -th output pattern, and $\varphi_k(x)$ is the output of the k -th perceptron of the hidden layer. Thus, when taking the **max** of the

outputs of the perceptrons φ_k that output a $-\infty$ unqualify pattern y^k as possible output of the associative memory. To train this associative memory, we have simply to define the supports of each key pattern. In this case the on regions of the corresponding patterns represent these supports.

From [3], during training, all supports of the memory are built as hyper-cubes of side equal to α , where α is obtained as

$$\alpha < \frac{1}{2} \min_{i < j} d(x^i, x^j) \tag{2}$$

where $d(x^i, x^j)$ is defined as

$$d(x^i, x^j) = \max \left\{ |x_l^i - x_l^j|_{l=1, \dots, n} \right\} \tag{3}$$

and $(x^i)_{i=1, \dots, m}$ is the set of key patterns.

To avoid collisions at the moment of classification it is necessary that the supports are disjoint two by two. This is not demonstrated in [7]. Next a brief proof that this happens is given.

Proposition 1. *Let M a multi-layered associative memory. If each key pattern x^k has a support $S_k = \{x : d(x, x^k) < \alpha\}$, with α as defined in equation (2), then it hold that $S_i \cap S_j = \emptyset, \forall i \neq j$.*

Proof. Let us suppose that $\exists_{i \neq j}$ such that $S_i \cap S_j \neq \emptyset$, this means that there is a x such that $d(x, x^i) < \alpha$ and $d(x, x^j) < \alpha$. When summing and using triangles' inequality we get: $d(x^i, x^j) \leq d(x, x^i) + d(x, x^j) \leq (\alpha + \alpha = 2\alpha)$, then $d(x^i, x^j) < 2\alpha$, or in other words $\frac{d(x^i, x^j)}{2} < \alpha$!!, which is a contradiction, thus the proposition hold.

One of the drawbacks of this method to construct the supports is that if two of them are of them too close, the (radius) of the neighborhood's supports reduce drastically.

In [7] it is proposed another method to increase the neighborhoods of the supports by means of the kernels' method. According to [7], with this the range of permissible noise is increased. However, this method makes expensive the computational cost and besides it imposes restrictions over the patterns very difficult to get.

3 Proposed Training Algorithm

In the content of this work, let us suppose that a pattern is represented in terms of n object features; then at each coordinated axis we can compute the variation obtained

per feature by ordering all components. By computing the average of variability per pattern it is possible to define a threshold. For practical purposes, this threshold allows to consider if two patterns can be considered to be the same from the point of view of one of their components. This way we avoid having very tiny supports with respect to the coordinated axis. This way the drawback of the algorithm described in last section is surpassed.

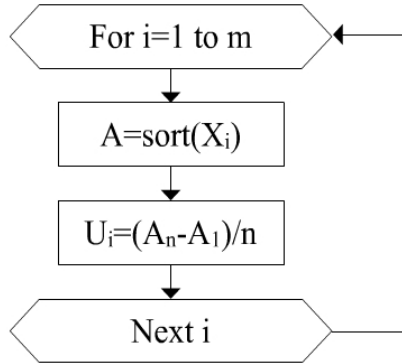


Fig. 2. Flowchart to get average variation threshold

The algorithm to find the average variation threshold by axis is observed in Fig. 2. In this case, X is a matrix of $n \times m$. At j -th column it is j -th pattern, while at i -th line are the m features of the i -th axis of each pattern. In U_i it is the average variation threshold for the i -th axis. The key point for the training of the multi-layered morphological perceptron consists on constructing the supports for each pattern. In general these supports must be disjoint two by two to avoid that two patterns be assigned to the same output. This is fulfilled in the proposed algorithm thanks to the following:

Proposition 2. Let Ω_i and Ω_j two arbitrary supports corresponding to different patterns in \mathbb{R}^n , built according to average variation threshold, then it holds that $\Omega_i = \Omega_j$ or $\Omega_i \cap \Omega_j = \emptyset$. [13].

Proof. Let x^i and x^j the corresponding patterns to supports Ω_i and Ω_j , respectively, then we have two cases:

- a. If it holds that $|x_k^i - x_k^j| < U_k, \forall_{k=1, \dots, n}$, with U_k is the average variation. If this holds then $\Omega_i = \Omega_j$.
- b. If $\exists_{k=1, \dots, n}$ such that $|x_k^i - x_k^j| > U_k$ would imply that at k -th coordinate, the support does not coincide and on the axis they are disjoint, thus they are disjoint in \mathbb{R}^n .

Evidently, when the patterns are too close, the neighborhoods would present occlusions. In these cases $\Omega_i = \Omega_j$ for $i \neq j$, one variant of this algorithm would be to consider that $\Omega_i \cap \Omega_j \neq \emptyset$, but that each neighborhood is centered at its respective key pattern. The advantage of this enhancement is that neighborhoods allowing more noise to be added to the patterns will be enlarged.

4 Numerical Examples

Example No. 1. Let the following set of key patterns in \mathbb{R}^n :

$$\left\{ x^1 = \begin{bmatrix} 3 \\ -4 \end{bmatrix}, x^2 = \begin{bmatrix} -1 \\ -4 \end{bmatrix}, x^3 = \begin{bmatrix} -2 \\ 2 \end{bmatrix}, x^4 = \begin{bmatrix} -2 \\ -3 \end{bmatrix}, x^5 = \begin{bmatrix} 4 \\ 5 \end{bmatrix}, x^6 = \begin{bmatrix} -3 \\ 2 \end{bmatrix} \right\} \quad (4)$$

a) Solution obtained by means of algorithm proposed in [7]:

By applying the algorithm proposed in [7], according to equation (2) $\alpha = 0.5$. Figure 3 shows the neighborhoods obtained when using this value of α .

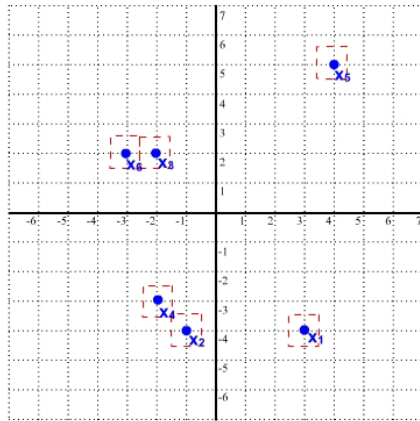


Fig. 3. Neighborhoods obtained when algorithm proposed in [7] is used

b) Solution obtained by means of algorithm proposed in this paper (first variant):

When using the algorithm to get the maximal support neighborhoods, by applying the flowchart shown in Figure 3, we get the threshold value U as:

$$U = \begin{bmatrix} 1.166 \\ 1.5 \end{bmatrix} \quad (5)$$

The neighborhoods obtained by using this threshold value are shown in Figure 4. The differences can be immediately appreciated. In this second case the range on noise for each pattern, as can be appreciated, is bigger.

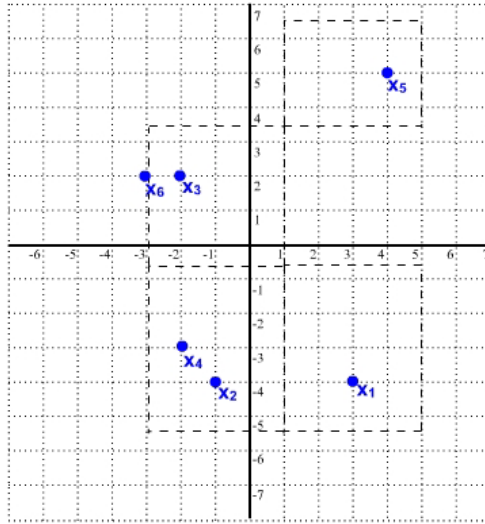


Fig. 4. Neighborhoods obtained when using first variant of the proposed algorithm

c) Solution obtained by means of algorithm proposed in this paper (second variant):

We apply the same procedure used by the first variant, but in this case the neighborhoods are centered at the corresponding key patterns. The threshold is the same given by equation (5).

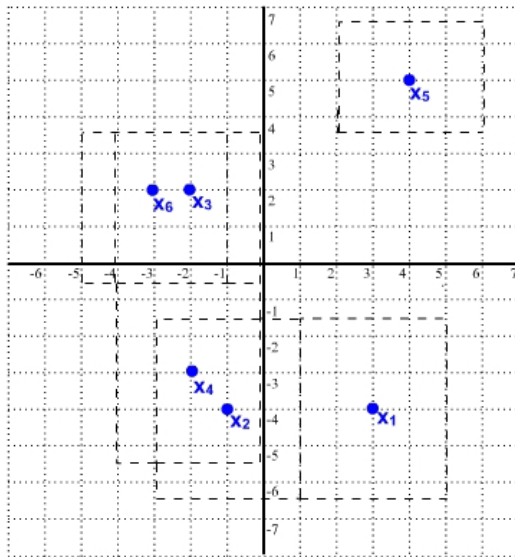


Fig. 5. Neighborhoods obtained when using second variant of the proposed algorithm

The neighborhoods obtained are shown in Fig. 5. Due to the neighborhoods are now centered at their respective key patterns, in this case the support for noise is bigger. Note also the occlusions between classes that do not occur in the first variant.

In the following section we show how the proposal described in this paper can be used not only to recall binary patterns but also gray-level patterns such images.

5 Experiments with Real Patterns

For this experiment we used the images shown in Fig. 6. These images are gray-level of 262×326 elements. They were perturbed with noise from 5% to 15%, in steps of 5%.



Fig. 6. Original images of experiments

The input key patterns where formed by describing each image by means of known Hu invariants ([10], [11]). Figures 7 to 9 show graphically the results obtained when using the algorithm proposed in [7], and the algorithm proposed in this paper to the images is shown in Figs. 10 to 12. From all of these figures we can observe that when adding, even small quantities of noise to the patterns, the original algorithm proposed in [7] fails to recall practically of the patterns, while the proposal, although with modest percentage, several of the desired patterns are correctly recalled.

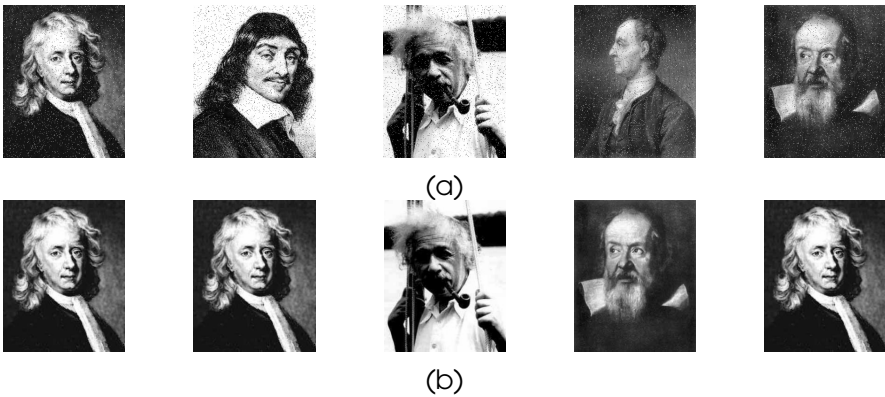


Fig. 7. (a) Images altered wit 5% of noise. (b) Images recalled using [7].

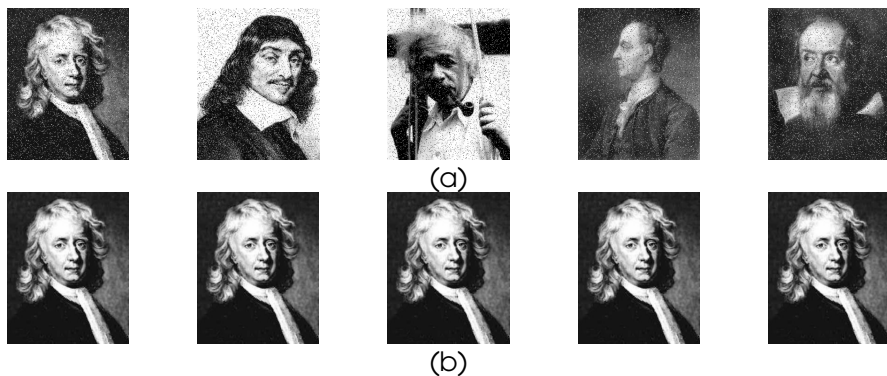


Fig. 8. (a) Images altered with 10% of noise. (b) Images recalled using [7].



Fig. 9. (a) Images altered with 15% of noise. (b) Images recalled using [7].

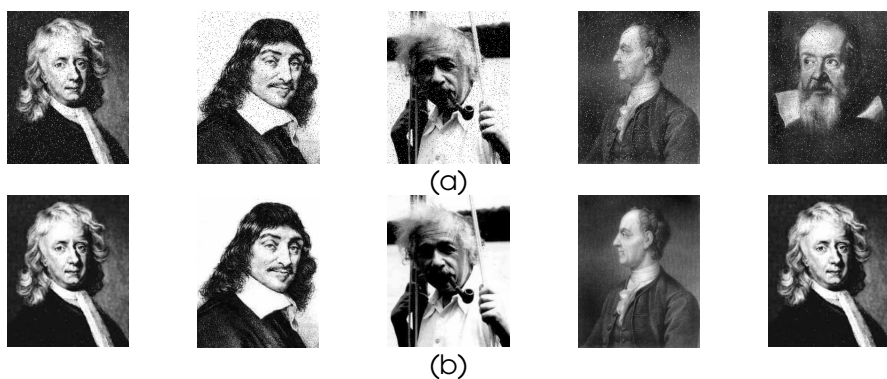


Fig. 10. (a) Images altered with 5% of noise. (b) Images recalled using the proposal.

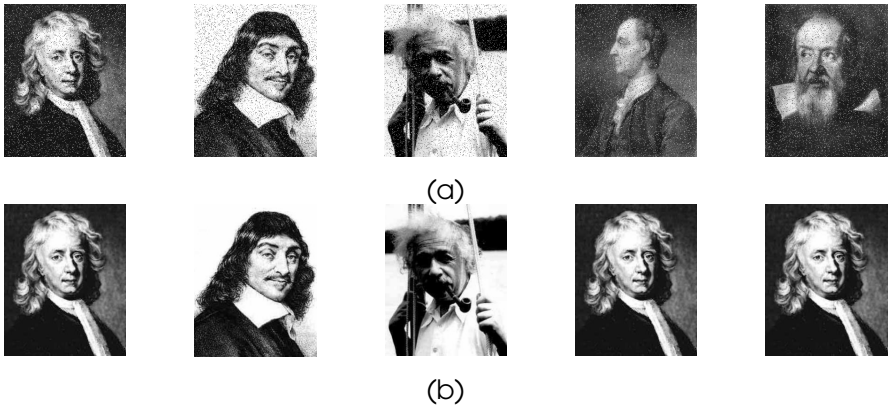


Fig. 11. (a) Images altered with 10% of noise. (b) Images recalled using the proposal.

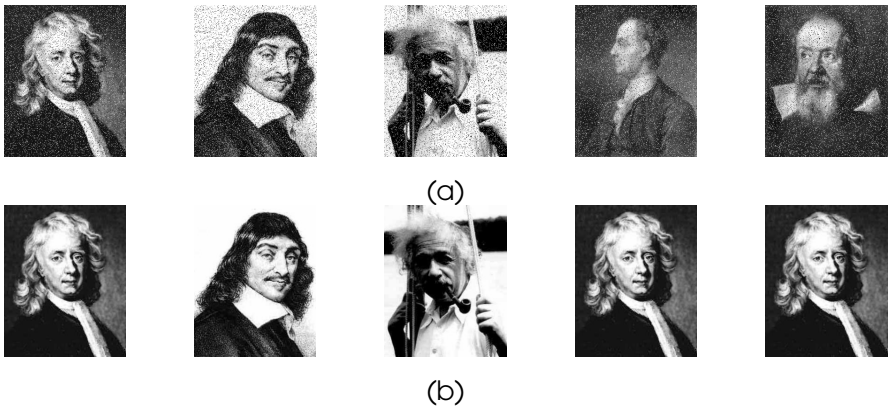


Fig. 12. (a) Images altered with 15% of noise. (b) Images recalled using the proposal

6 Conclusions

In this paper we have presented an algorithm to train the multi-layered morphological perceptron that allows to build more efficient support neighborhoods, since the point of view of pattern recall, from the set of key pattern patterns of the training set of an associative memory in both its auto-associative or hetero-associative way of operation. By several experiments with real patterns, we have shown that the proposal can be used to recall gray-level images and not only binary images. We show the superiority of the new one.

Acknowledgements. This work was economically supported by CIC-IPN, COFFAA-IPN and CONACYT under grant 46805 and SIP under grants 20071438 and 20071084, respectively.

References

- [1] Bishop, C.: *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford England (1995)
- [2] Rostenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review* 65, 386–408 (1958)
- [3] Rostenblatt, F.: *Principles of Neurodynamics: Perceptrons and the theory of brain mechanism*, Spartans Books, Washington D.C (1962)
- [4] Ritter, G.X., et al.: An introduction to morphological Neural Networks. In: *Proceedings of the 13th International Conference on Pattern Recognition*, pp. 709–717 (1996)
- [5] Ritter, G.X., et al.: Morphological associative memories. *IEEE Transactions on Neural Networks* C-9, 281–293 (1998)
- [6] Ritter, G.X.: *Morphological Perceptrons*. ISAS'97, Intelligent Systems and Semiotics, Gaithersburg, Maryland (1997)
- [7] Ritter, G.X.: A new auto-associative memory based on lattice algebra. In: *Proc. 9th CIARP 2004*, La Havana, Cuba pp. 148–155 (2004)
- [8] Kishan, M., et al.: *Elements of Artificial Neural Networks*. The MIT Press, Cambridge, Massachusetts, London, England (1997)
- [9] Pessoa, L.F.C., et al.: Morphological/Rank Neural Networks and their adaptive optimal image processing. *IEEE International Conference on Acoustic, Speech, and Signal Processing* 6, 3399–3402 (1996)
- [10] Hu, M.K.: Pattern recognition by moments invariants. *Proceeding of the IRE* 49, 1428 (1961)
- [11] Hu, M.K.: Visual pattern recognition by moments invariants, *IRE Transactions on Information Theory*, 179–187 (1962)
- [12] Sussner, P.: *Morphological Perceptron Learning*. In: *Proceedings of the 1998 International Symposium on Intelligent Systems and Semiotics*, pp. 477–482, Gaithersburg, Maryland (1998)
- [13] Barron, R., et al.: New Improved Algorithm for the training of a Morphological Associative Memory, *Research in Computer Science. Special Issue: Neural Networks and Associative Memories* 21, 49–59 (2006)