A New Auto-associative Memory Based on Lattice Algebra

Gerhard X. Ritter, Laurentiu Iancu, and Mark S. Schmalz

CISE Department, University of Florida, Gainesville, FL 32611-6120, USA
{ritter,liancu,mssz}@cise.ufl.edu

Abstract. This paper presents a novel, three-stage, auto-associative memory based on lattice algebra. The first two stages of this memory consist of correlation matrix memories within the lattice domain. The third and final stage is a two-layer feed-forward network based on dendritic computing. The output nodes of this feed-forward network yield the desired pattern vector association. The computations performed by each stage are all lattice based and, thus, provide for fast computation and avoidance of convergence problems. Additionally, the proposed model is extremely robust in the presence of noise. Bounds of allowable noise that guarantees perfect output are also discussed.

1 Introduction

The computational framework of morphological associative memories involves lattice algebraic operations, such as dilation, erosion, and max and min product. Using these operations, two associative memories can be defined. These memories can be either hetero-associative or auto-associative, depending on the pattern associations they store. The morphological auto-associative memories are known to be robust in the presence of certain types of noise, but also rather vulnerable to random noise [1–4].

The kernel method described in [1, 3, 5] allows the construction of auto-associative memories with improved robustness to random noise. However, even with the kernel method, complete reconstruction of exemplar patterns that have undergone only minute distortions is not guaranteed. In this paper we present a new method of creating an auto-associative memory that takes into account the kernel method discussed in [1, 3] as well as a two-layer morphological feed-forward network based on neurons with dendritic structures [6–8].

2 Auto-associative Memories in the Lattice Domain

The lattice algebra in which our memories operate is discussed in detail in [1]. In this algebraic system, which consists of the set of extended real numbers $\mathbb{R}_{\pm\infty}$ and the operations $+$, $\vee$ and $\wedge$, we define two matrix operations called *max product* and *min product*, denoted by the symbols $\boxdot\!\!\!\vee$ and $\boxdot\!\!\!\wedge$, respectively. For an $m \times p$ matrix A and a $p \times n$ matrix B with entries from $\mathbb{R}$, the $m \times n$ matrix

A. Sanfeliu et al. (Eds.): CIARP 2004, LNCS 3287, pp. 148–155, 2004.

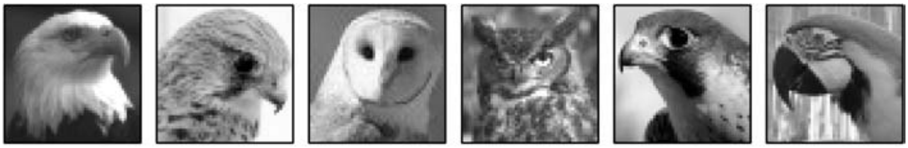

Fig. 1. Images $\mathbf{p}^1, \ldots, \mathbf{p}^6$, converted into column vectors $\mathbf{x}^1, \ldots, \mathbf{x}^6$, and stored in the morphological auto-associative memories W_{XX} and M_{XX}.

$C = A \boxed{\vee} B$ has the i,jth entry $c_{ij} = \bigvee_{k=1}^{p}(a_{ik} + b_{kj})$. Likewise, the i,jth entry of matrix $C = A \boxed{\wedge} B$ is $c_{ij} = \bigwedge_{k=1}^{p}(a_{ik} + b_{kj})$.

For a set of pattern vectors $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^k\} \subset \mathbb{R}^n$ we construct two natural auto-associative memories W_{XX} and M_{XX} of size $n \times n$ defined by $W_{XX} = \bigwedge_{\xi=1}^{k} \left[\mathbf{x}^\xi \times (-\mathbf{x}^\xi)'\right]$ and $M_{XX} = \bigvee_{\xi=1}^{k} \left[\mathbf{x}^\xi \times (-\mathbf{x}^\xi)'\right]$. Here the symbol $\times$ denotes the *morphological outer product* of two vectors, such that $\mathbf{x} \times \mathbf{x}' = \mathbf{x} \boxed{\vee} \mathbf{x}' = \mathbf{x} \boxed{\wedge} \mathbf{x}'$. In [1] we proved that

$$W_{XX} \boxed{\vee} X = X = M_{XX} \boxed{\wedge} X \; , \tag{1}$$

where X can consist of any arbitrarily large number of pattern vectors. In other words, morphological auto-associative memories have infinite capacity and perfect recall of undistorted patterns.

Example 1. For a visual example, consider the six pattern images $\mathbf{p}^1, \ldots, \mathbf{p}^6$ shown in Fig. 1. Each $\mathbf{p}^\xi$, $\xi = 1, \ldots, 6$, is a 50×50 pixel 256-gray scale image. For uncorrupted input, perfect recall is guaranteed by Eq. (1) if we use the memory W_{XX} or M_{XX}. Using the standard row-scan method, each pattern image $\mathbf{p}^\xi$ can be converted into a pattern vector $\mathbf{x}^\xi = \left(x_1^\xi, \ldots, x_{2500}^\xi\right)'$ by defining $x_{50(r-1)+c}^\xi = p^\xi(r,c)$ for $r, c = 1, \ldots, 50$.

Morphological associative memories are extremely robust in the presence of certain types of noise, missing data, or occlusions. We say that a distorted version $\tilde{\mathbf{x}}^\xi$ of the pattern $\mathbf{x}^\xi$ has undergone an *erosive change* whenever $\tilde{\mathbf{x}}^\xi \leq \mathbf{x}^\xi$ and a *dilative change* whenever $\tilde{\mathbf{x}}^\xi \geq \mathbf{x}^\xi$. The morphological memory W_{XX} is a memory of dilative type (patterns are recalled using the max product) and thus is extremely robust in the presence of erosive noise. Conversely, the memory M_{XX}, of erosive type, is particularly robust to dilative noise.

Several mathematical results proved in [1] provide necessary and sufficient conditions for the maximum amount of distortion of a pattern that still guarantees perfect recall. In spite of being robust to the specific type of noise they tolerate, the morphological memories W_{XX} and M_{XX} can fail to recognize patterns that are affected by a different type of noise, even in a minute amount. Thus, W_{XX} fails rather easily in the presence of dilative noise, while M_{XX} fails in the presence of erosive noise. Additionally, both types of morphological memories are vulnerable to random noise, i.e. noise that is both dilative and erosive in nature.

The following experiment illustrates this behavior of the lattice auto-associative memories. Figure 2 shows the images $\mathbf{p}^1, \ldots, \mathbf{p}^6$ in which 75% of the pixels

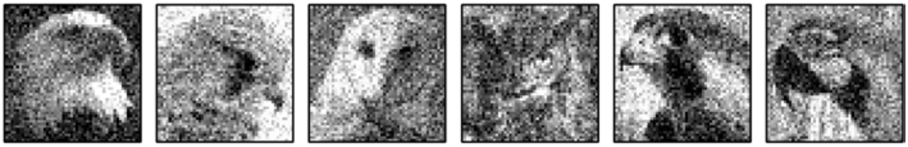

Fig. 2. Images corrupted with 75% random noise (both dilative and erosive) in the range $[-72, 72]$. Pixel values are in the range $[0, 255]$.

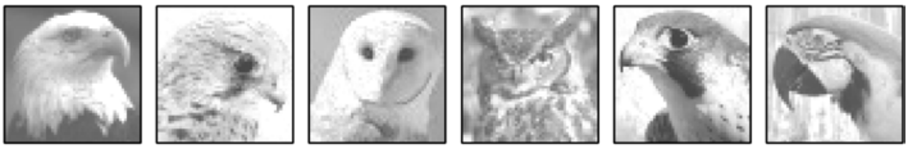

Fig. 3. Incorrect recall of memory W_{XX} when presented with the noisy input images from Fig. 2. The output appears shifted towards white pixel values.

have been corrupted by random noise. The noise has uniform distribution and is in the range $[-72, 72]$. When the pixel values affected by noise become less than 0 or greater than 255, the result is clamped at 0 and 255, respectively. The range of noise has been chosen by calculation, in order to compare the memories W_{XX} and M_{XX} to the ones based on the dendritic model, as discussed in the subsequent sections of this paper.

The output of the memory W_{XX} when presented with the patterns corrupted with random noise is illustrated in Fig. 3. When compared to the original images from Fig. 1 stored in the memory, the patterns recalled by W_{XX} appear to be different from the original $\mathbf{p}^\xi$, $\xi = 1, \ldots, 6$. The output of W_{XX} is offset toward white (high pixel values), as W_{XX} is applied dilatively via the max product. A similar experiment will show that the output of M_{XX} will be shifted toward black (low pixel values), as M_{XX} is a memory of erosive type, used in conjunction with the min product.

Because of this failure of the memories W_{XX} and M_{XX}, we developed the method of *kernels* to treat random noise. This method is discussed in detail in [3]. Basically, a *kernel* for X is a set of vectors $Z = \{\mathbf{z}^1, \ldots, \mathbf{z}^k\} \subset \mathbb{R}^n$ such that $\forall \gamma = 1, \ldots, k$,

1. $\mathbf{z}^\gamma \wedge \mathbf{z}^\xi = 0 \;\; \forall \xi \neq \gamma$,
2. $\mathbf{z}^\gamma$ contains exactly one non-zero entry, and
3. $W_{XX} \boxvee \mathbf{z}^\gamma = \mathbf{x}^\gamma$.
4. If z_i^γ denotes the non-zero entry of $\mathbf{z}^\gamma$, then $z_i^\gamma = x_i^\gamma$.

Now if Z satisfies the above conditions and $\tilde{\mathbf{x}}^\gamma$ denotes a distorted version of $\mathbf{x}^\gamma$ such that $\tilde{x}_i^\gamma = z_i^\gamma$, where z_i^γ denotes the non-zero entry of $\mathbf{z}^\gamma$, then $W_{XX} \boxvee (M_{ZZ} \boxwedge \tilde{\mathbf{x}}^\gamma) = \mathbf{x}^\gamma$. Here we assume that the set X of exemplar patterns have non-negative coordinates, which is generally the case in pattern recognition problems. If, however, $\tilde{x}_i^\gamma \neq z_i^\gamma$, then perfect recall cannot be achieved. To overcome this shortcoming, we developed an extended model that takes into account dendritic neural structures.

3 The Dendritic Model

The artificial neural model that employs dendritic computation has been motivated by the fact that several researchers have proposed that dendrites, and not the neurons, are the elementary computing devices of the brain, capable of implementing logical functions such as AND, OR, and NOT [9–14]. In the mammalian brain, dendrites span all cortical layers and account for the largest component in both surface and volume. Thus, dendrites cannot be omitted when attempting to build artificial neural models.

Inspired by the neurons of the biological brain, we developed a model of *morphological neuron* that possesses *dendritic structures*. A number of such neurons can then be arranged on one layer, similarly to the classical single layer perceptron (SLP), in order to build a single layer morphological perceptron with dendritic structures (SLMP). This artificial model is described in detail in [8, 15] and only briefly summarized below due to page limitation.

Let $N_1, \ldots, N_n$ denote a set of input neurons, which provide synaptic input to the main layer of neurons with dendritic structures, $M_1, \ldots, M_m$, which is also the output layer. The value of an input neuron N_i $(i = 1, \ldots, n)$ propagates through its axonal tree to the terminal branches that make contact with the neuron M_j $(j = 1, \ldots, m)$. The weight of an axonal branch of neuron N_i terminating on the kth dendrite of M_j is denoted by w_{ijk}^{ℓ}, where the superscript $\ell \in \{0, 1\}$ distinguishes between *excitatory* $(\ell = 1)$ and *inhibitory* $(\ell = 0)$ input to the dendrite. The kth dendrite of M_j will respond to the total input received from the neurons $N_1, \ldots, N_n$ and will either accept or inhibit the received input. The computation of the kth dendrite of M_j is given by

$$\tau_k^j(\mathbf{x}) = p_{jk} \bigwedge_{i \in I(k)} \bigwedge_{\ell \in L(i)} (-1)^{1-\ell} \left(x_i + w_{ijk}^{\ell} \right) \ , \tag{2}$$

where $\mathbf{x} = (x_1, \ldots, x_n)'$ denotes the input value of the neurons $N_1, \ldots, N_n$ with x_i representing the value of N_i; $I(k) \subseteq \{1, \ldots, n\}$ corresponds to the set of all input neurons with terminal fibers that synapse on the kth dendrite of M_j; $L(i) \subseteq \{0, 1\}$ corresponds to the set of terminal fibers of N_i that synapse on the kth dendrite of M_j; and $p_{jk} \in \{-1, 1\}$ denotes the excitatory $(p_{jk} = 1)$ or inhibitory $(p_{jk} = -1)$ response of the kth dendrite of M_j to the received input.

It follows from the formulation $L(i) \subseteq \{0, 1\}$ that the ith neuron N_i can have at most two synapses on a given dendrite k. Also, if the value $\ell = 1$, then the input $\left(x_i + w_{ijk}^1\right)$ is excitatory, and inhibitory for $\ell = 0$ since in this case we have $-\left(x_i + w_{ijk}^0\right)$.

The value $\tau_k^j(\mathbf{x})$ is passed to the cell body and the state of M_j is a function of the input received from all its dendrites. The total value received by M_j is given by $\tau^j(\mathbf{x}) = p_j \bigwedge_{k=1}^{K_j} \tau_k^j(\mathbf{x})$, where K_j denotes the total number of dendrites of M_j

and $p_j = \pm 1$ denotes the response of the cell body to the received dendritic input. Here again, $p_j = 1$ means that the input is accepted, whereas $p_j = -1$ means that the cell rejects the received input. The *next* state of M_j is then determined by an activation function f, namely $y_j = f\left(\tau^j(\mathbf{x})\right)$. Typical activation functions used with the dendritic model include the hard-limiter and the pure linear identity function. The single layer morphological perceptron usually employs the former.

For a more thorough understanding of this model as well as its computational performance, we refer the reader to examples and theorems given in [7, 8, 15].

4 An Auto-associative Memory Based on the Dendritic Model

Based on the dendritic model described in the previous section, we construct an auto-associative memory that can store a set of patterns $X = \left\{\mathbf{x}^1, \ldots, \mathbf{x}^k\right\} \subset \mathbb{R}^n$ and can also cope with random noise. The memory we are about to describe will consist of n input neurons $N_1, \ldots, N_n$, k neurons in the hidden layer, which we denote by $H_1, \ldots, H_k$, and n output neurons $M_1, \ldots, M_n$. Let $\mathrm{d}\left(\mathbf{x}^\xi, \mathbf{x}^\gamma\right) = \max\left\{\left|x_i^\xi - x_i^\gamma\right| : i = 1, \ldots, n\right\}$ and choose an allowable noise parameter α with α satisfying

$$\alpha < \frac{1}{2} \min\left\{\mathrm{d}\left(\mathbf{x}^\xi, \mathbf{x}^\gamma\right) : \xi < \gamma,\ \xi, \gamma \in \{1, \ldots, k\}\right\} . \tag{3}$$

For $\mathbf{x} \in \mathbb{R}^n$, the input for N_i will be the ith coordinate of $\mathbf{x}$. Each neuron H_j in the hidden layer has exactly one dendrite, which contains the synaptic sites of the terminal axonal fibers of N_i for $i = 1, \ldots, n$. The weights of the terminal fibers of N_i terminating on the dendrite of H_j are given by

$$w_{ij}^\ell = \begin{cases} -\left(x_i^j - \alpha\right) & \text{if } \ell = 1 \\ -\left(x_i^j + \alpha\right) & \text{if } \ell = 0 \end{cases} ,$$

where $i = 1, \ldots, n$ and $j = 1, \ldots, k$. For a given input $\mathbf{x} \in \mathbb{R}^n$, the dendrite of H_j computes $\tau^j(\mathbf{x}) = \bigwedge_{i=1}^n \bigwedge_{\ell=0}^1 (-1)^{1-\ell}\left(x_i + w_{ij}^\ell\right)$. The state of the neuron H_j is determined by the hard-limiter activation function

$$f(z) = \begin{cases} 0 & \text{if } z \geq 0 \\ -\infty & \text{if } z < 0 \end{cases} .$$

Thus, the output of H_j is given by $f\left[\tau^j(\mathbf{x})\right]$ and is passed along its axon and axonal fibers to the output neurons $M_1, \ldots, M_n$.

Similar to the hidden layer neurons, each output neuron M_h, $h = 1, \ldots, n$, has one dendrite. However, each hidden neuron H_j has exactly one excitatory axonal fiber and no inhibitory fibers terminating on the dendrite of M_h. Figure 4 illustrates this dendritic network model. The excitatory fiber of M_j terminating on M_h has synaptic weight $v_{jh} = x_h^j$. The computation performed by M_h is

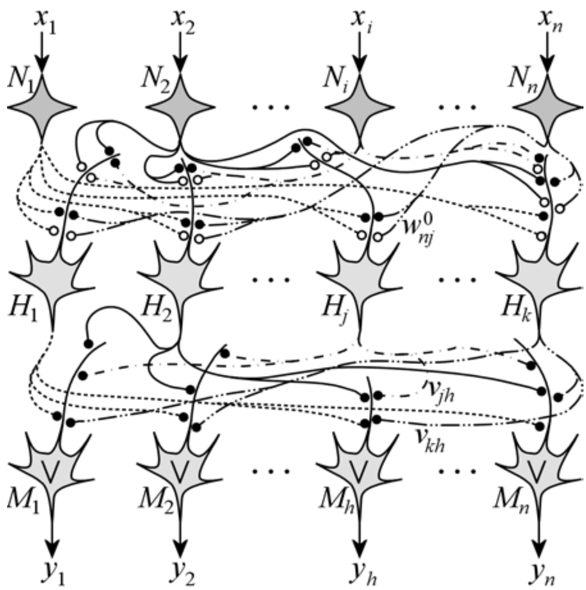

Fig. 4. The topology of the morphological auto-associative memory based on the dendritic model. The network is fully connected; all axonal branches from input neurons synapse via two fibers on all hidden neurons, which in turn connect to all output nodes via excitatory fibers.

given by $\tau^h(\mathbf{q}) = \bigvee_{j=1}^{k} (q_j + v_{jh})$, where q_j denotes the output of H_j, namely $q_j = f\left[\tau^j(\mathbf{x})\right]$. The activation function for each output neuron M_h is the simple linear identity function $f(z) = z$.

Each neuron H_j will have output value $f(q_j) = 0$ if and only if $\mathbf{x}$ is an element of the hypercube $B^j = \left\{ (x_1, \ldots, x_n) \in \mathbb{R}^n : x_i^j - \alpha \leq x_i \leq x_i^j + \alpha,\ i = 1, \ldots, n \right\}$ and $f(q_j) = -\infty$ whenever $\mathbf{x} \in \mathbb{R}^n \setminus B^j$. Thus, the output of this network will be $\mathbf{y} = (y_1, \ldots, y_n) = \left(x_1^j, \ldots, x_n^j\right) = \mathbf{x}^j$ if and only if $\mathbf{x} \in B^j$. That is, whenever $\mathbf{x}$ is a corrupted version of $\mathbf{x}^j$ with each coordinate of $\mathbf{x}$ not exceeding the allowable noise level α, then $\mathbf{x}$ will be identified as $\mathbf{x}^j$.

If $\mathbf{x}$ does not fall within the allowable noise level α specified by Eq. (3), then the output will not be $\mathbf{x}^j$. We can, however, increase the geometric territory for distorted versions of $\mathbf{x}^j$ by first employing the kernel method. In particular, suppose that X is strongly lattice independent and Z is a kernel for X satisfying properties 1–4 specified earlier. Then, for each pattern $\mathbf{x}^j$, the user can add the noise parameter α about $\mathbf{z}^j$ as well, as shown in Fig. 5. This increases the allowable range of noise. In particular, if $|z_i^j - x_i| > \alpha\ \forall i$, then $\mathbf{x}$ is rejected as an input vector. However, if $\mathbf{x}$ falls within any of the shaded regions illustrated in Fig. 5, then the memory flow diagram $\mathbf{x} \to M_{ZZ} \to W_{XX} \to M \to \mathbf{x}^j$, where M denotes the two-layer feed-forward dendritic network, provides perfect recall output. That is, we first compute $\mathbf{y} = W_{XX} \boxdot (M_{ZZ} \boxtimes \mathbf{x})$ and then use $\mathbf{y}$ as the input vector to the feed-forward network M. For purpose of illustration, we

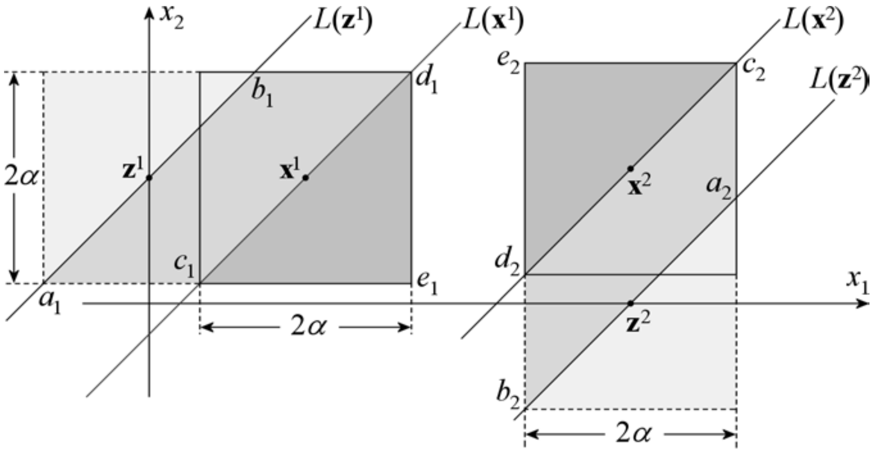

Fig. 5. The two patterns $\mathbf{x}^1, \mathbf{x}^2$ with corresponding kernel vectors $\mathbf{z}^1, \mathbf{z}^2$. The non-zero entries of $\mathbf{z}^1$ and $\mathbf{z}^2$ are z_2^1 and z_1^2, respectively. Every point on and between the lines $L(\mathbf{x}^1)$ and $L(\mathbf{x}^2)$ is a fixed point of W_{XX}. Similarly, every point on and between $L(\mathbf{z}^1)$ and $L(\mathbf{z}^2)$ is a fixed point of M_{ZZ}.

used two independent vectors in $\mathbb{R}^2$. The corresponding kernel vectors lie on the coordinate axes. Observe that if $\mathbf{x}$ lies in the lightly shaded area above the line $L(\mathbf{z}^1)$, then $M_{ZZ} \boxtimes \mathbf{x}$ lies on the segment $[a_1, b_1] \subset L(\mathbf{z}^1)$. If $\mathbf{x}$ lies within the other shaded regions, then $M_{ZZ} \boxtimes \mathbf{x} = \mathbf{x}$. Everything within the parallelogram $\langle a_1, b_1, d_1, c_1 \rangle$ (including $[a_1, b_1]$) will be mapped under W_{XX} onto the segment $[c_1, d_1] \subset L(\mathbf{x}^1)$ and any point within the triangle specified by $\langle c_1, d_1, e_1 \rangle$ will be mapped by M to $\mathbf{x}^1$. This schema can be easily extended to any dimension.

Example 2. To illustrate the performance of this auto-associative memory, we stored the same exemplar patterns $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^6 \in \mathbb{R}^{2500}$ used in Example 1 and shown in Fig. 1. The images were then distorted by randomly corrupting 75% of the coordinates within a noise level α, chosen to satisfy the inequality in (3). Letting $\alpha = \frac{2}{5} \min \left\{ \mathrm{d}\left(\mathbf{x}^\xi, \mathbf{x}^\gamma\right) : 1 \leq \xi < \gamma \leq 6 \right\}$ we obtain $\alpha = \frac{2}{5} \cdot 180 = 72$. This is how the allowable amount of distortion $[-72, 72]$ was chosen in Example 1, and applied to the images resulted in the noisy patterns of Fig. 2. Using the same corrupted patterns as input to the memory based on the model described here, we obtain perfect recall, i.e. patterns identical to the input patterns in Fig 1.

5 Conclusions

We presented a new paradigm for an auto-associative memory based on lattice algebra that combines correlation matrix memories and a dendritic feed-forward network. We gave a brief overview of correlation matrix memories in the lattice domain as well as single layer morphological perceptrons with dendritic structures, whose computational capability exceeds that of the classical single layer perceptrons. Using a two-layer dendritic model, we defined an auto-associative memory that is able to store and recall any finite collection of n-dimensional pattern vectors. We showed by example that this memory is robust in the presence of noise where the allowable noise level depends only on the minimum Chebyshev distance between the patterns.

The allowable noise level can be increased dramatically if the set of patterns is strongly lattice independent. It follows from the description of this model that recognition does not involve any lengthy training sessions but only straightforward computation of weights in terms of pattern distances. Convergence problems are non-existent as recognition is achieved in one step in terms of information feed-forward flow through the network.

References

1. Ritter, G.X., Sussner, P., Diaz de Leon, J.L.: Morphological Associative Memories. IEEE Trans. on Neural Networks **9**(2) (March 1998) 281–293
2. Ritter, G.X., Diaz de Leon, J.L., Sussner, P.: Morphological Bidirectional Associative Memories. Neural Networks **12** (March 1999) 851–867
3. Ritter, G.X., Urcid, G., Iancu, L.: Reconstruction of Noisy Patterns Using Morphological Associative Memories. J. of Mathematical Imaging and Vision **19**(2) (2003) 95–111
4. Urcid, G., Ritter, G.X., Iancu, L.: Kernel Computation in Morphological Bidirectional Associative Memories. Proc. 8th Iberoamerican Congress on Pattern Recognition CIARP'03. Havana, Cuba (November 2003) 552–559
5. Sussner, P.: Observations on Morphological Associative Memories and the Kernel Method. Neurocomputing **31** (2000) 167–183
6. Ritter, G.X., Iancu, L., Urcid, G.: Neurons, Dendrites, and Pattern Classification. Proc. 8th Iberoamerican Congress on Pattern Recognition CIARP'03. Havana, Cuba (November 2003) 1–16
7. Ritter, G.X., Iancu, L.: Lattice Algebra Approach to Neural Networks and Pattern Classification. Pattern Recognition and Image Analysis **14**(2) (2004) 190–197
8. Ritter, G.X., Iancu, L.: Morphological Perceptrons. Preprint submitted to IEEE Trans. on Neural Networks.
9. Eccles, J.C.: The Understanding of the Brain. McGraw-Hill, New York (1977)
10. Koch, C., Segev, I. (eds.): Methods in Neuronal Modeling: From Synapses to Networks. MIT Press, Boston (1989)
11. McKenna, T., Davis, J., Zornetzer, S.F. (eds.): Single Neuron Computation. Academic Press, San Diego (1992)
12. Mel, B.W.: Synaptic Integration in Excitable Dendritic Trees. J. of Neurophysiology **70** (1993) 1086–1101
13. Rall, W., Segev, I.: Functional Possibilities for Synapses on Dendrites and Dendritic Spines. In: Edelman, G.M., Gall, E.E., Cowan, W.M. (eds.): Synaptic Function. Wiley, New York (1987) 605–636
14. Segev, I.: Dendritic Processing. In: Arbib, M. (ed.): The Handbook of Brain Theory and Neural Networks. MIT Press, Boston (1998) 282–289
15. Ritter, G.X., Urcid, G.: Lattice Algebra Approach to Single Neuron Computation. IEEE Trans. on Neural Networks **14**(2) (March 2003) 282–295