Recent Developments in Lattice Neural Networks

Gerhard X. Ritter[1,*] and Gonzalo Urcid[2]

[1] CISE Department, University of Florida, Gainesville FL 32611–6120, USA
ritter@cise.ufl.edu
[2] Optics Department, INAOE, Tonantzintla Pue. 72000, Mexico
gurcid@inaoep.mx

1 Introduction

This paper provides a brief summary of recent developments and some pertinent results in the evolving theory of lattice based neural networks. The first two sections deal with lattice based correlation memories while the remaining two sections focus on dendritic computing. The final section outlines some new research avenues and associated questions.

2 Matrix based associative memories

One of the first goals achieved in the development of lattice based neural networks also known as morphological neural networks (MNNs) was the establishment of a morphological associative memory or MAM for short. In its basic form, this model of an associative memory resembles the well-known correlation memory or linear associative memory [5]. As in correlation encoding, the morphological associative memory provides a simple method to add new associations. A weakness in linear correlation encoding is the requirement of orthogonality of the key vectors in order to exhibit perfect recall of the fundamental associations. The morphological auto-associative memory does not restrict the domain of the key vectors in any way. Thus, as many associations as desired can be encoded into the memory [1, 2].

Lattice based associative memories are in many ways similar to the classical correlation memories but also exhibit many properties that are drastically different. Before defining lattice based correlation, we need to define the corresponding matrix operations in the lattice domain. Given two $m \times n$ matrices $A = (a_{ij})$ and $B = (b_{ij})$, then then the *pointwise maximum*, $A \vee B$, of A and B, is the $m \times n$ matrix C defined by $A \vee B = C$, where $c_{ij} = a_{ij} \vee b_{ij}$. Similarly, the *pointwise minimum* $A \wedge B$ is defined as $A \wedge B = C$, where $c_{ij} = a_{ij} \wedge b_{ij}$. If A is $m \times p$ and B is $p \times n$, then the *max product* of A and B is the matrix $C = A \bigtriangledown B$, where $c_{ij} = \bigvee_{k=1}^{p} (a_{ik} + b_{kj})$. Observe that this product is analogous to the usual matrix product $c_{ij} = \sum_{k=1}^{p} (a_{ik} \times b_{kj})$, with the symbol $\sum$ replaced by $\bigvee$. Since

* Corresponding author. Fax: (352) 392–1220; Tel.: (352) 392-1200

© A. Gelbukh, C. Yáñez Márquez, O. Camacho Nieto (Eds.)
Advances in Artificial Intelligence and Computer Science
Research on Computing Science 14, 2005, pp. 1-12

$\bigvee$ replaces $\sum$ in our definition, the pointwise maximum can be thought of as matrix addition. The *min product* of A and B is the matrix $C = A \boxtimes B$, where $c_{ij} = \bigwedge_{k=1}^{p}(a_{ik} + b_{kj})$. Finally, the *lattice outer product* of two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ is defined as

$$\mathbf{y} \times \mathbf{x}' = \begin{pmatrix} y_1 + x_1 & \cdots & y_1 + x_n \\ \vdots & \ddots & \vdots \\ y_m + x_1 & \cdots & y_m + x_n \end{pmatrix} .$$

It is worthwhile to note that $\mathbf{y} \times \mathbf{x}' = \mathbf{y} \boxtimes \mathbf{x}' = \mathbf{y} \boxtimes \mathbf{x}'$.

Henceforth we shall let $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^k\} \subset \mathbb{R}^n$ and $Y = \{\mathbf{y}^1, \ldots, \mathbf{y}^k\} \subset \mathbb{R}^m$ denote two sets of pattern associations. An associative memory for the pair (X, Y) is a transform $W : \mathbb{R}^n \to \mathbb{R}^m$ with the property that $W(\mathbf{x}^\xi) = \mathbf{y}^\xi$ for $\xi = 1, \ldots, k$. In the special case where $X = Y$, the memory W is called an *auto-associative memory.*

For a given pair of pattern associations (X, Y) there are two natural lattice based $m \times n$ memories W_{XY} and M_{XY}. These are defined by

$$W_{XY} = \bigwedge_{\xi=1}^{k} [\mathbf{y}^\xi \times (\mathbf{x}^\xi)^*] \quad \text{and} \quad M_{XY} = \bigvee_{\xi=1}^{k} [\mathbf{y}^\xi \times (\mathbf{x}^\xi)^*] , \tag{1}$$

where $\mathbf{x}^*$ denotes the *lattice conjugate* of $\mathbf{x}$ defined by $\mathbf{x}^* = (-\mathbf{x})'$. It has been shown that if $X = Y$, then $W_{XX} \boxtimes \mathbf{x}^\xi = \mathbf{x}^\xi = M_{XX} \boxtimes \mathbf{x}^\xi$ for $\xi = 1, \ldots, k$; that is, recall is always perfect whenever the input is not distorted. The same cannot be said when $X \neq Y$. To assure perfect recall for perfect input, the hypothesis of the following theorem must be satisfied:

THEOREM 2.1 *W_{XY} is a perfect recall memory for the pattern association $(\mathbf{x}^\lambda, \mathbf{y}^\lambda)$ if and only if each row of the matrix $[\mathbf{y}^\lambda + (\mathbf{x}^\lambda)^*] - W_{XY}$ contains a zero entry. Similarly, M_{XY} is a perfect recall memory for the pattern pair $(\mathbf{x}^\lambda, \mathbf{y}^\lambda)$ if and only if each row of the matrix $M_{XY} - [\mathbf{y}^\lambda + (\mathbf{x}^\lambda)^*]$ contains a zero entry.*

The theorem points to one obvious weakness of matrix correlation in the lattice domain: in real applications it is often impossible to satisfy the hypothesis of the theorem. Furthermore, we are still assuming perfect input for both the auto-associative as well as hetero-associative memories. Various problems arise when noise is introduced. It has been shown that the memories W_{XX} and M_{XX} are extremely robust in the presence of erosive and dilative noise, respectively [1, 2]. However, both memories fail miserably in the presence of mixed (i.e., erosive plus dilative) random noise. The method of kernels - originally proposed in [1], generalized in [2], and further generalized in [7] - was introduced in order to manage random noise. Although some excellent resuts were obtained in recalling patterns from randomly distorted input patterns in the auto-associative case using the kernel method [2], it can be easily demonstrated that the kernel method will fail in a variety of cases. Such demonstrations became possible after the connections between lattice dependence and the fixed points of lattice transforms was discovered. We review these recent discoveries in the next section.

3 New results concerning matrix based memories

Several important concepts from lattice theory need to be revisited in order to fully understand the weaknesses and strengths of lattice correlation memories. The first two concepts are those of *lattice dependence* and *lattice independence.* The particular lattices pertinent in our discussion are the *bounded lattice ordered group* or *blog* $(\mathbb{R}_{\pm\infty}, \vee, \wedge, +, +^*)$, its sublattice $(\mathbb{R}, \vee, \wedge, +)$, and the corresponding vector space lattices $(\mathbb{R}^n_{\pm\infty}, \vee, \wedge, +, +^*)$ and $(\mathbb{R}^n, \vee, \wedge, +)$. Here, $\mathbb{R}_{\pm\infty}$ denotes the extended set of real numbers with the symbols ∞ and $-\infty$ included; that is, $\mathbb{R}_{\pm\infty} = \mathbb{R} \cup \{-\infty, \infty\}$. The additions $+$ and $+^*$ are identical for real numbers as well as for the extensions $a + \infty = \infty + a = \infty = a +^* \infty = \infty +^* a$ $\forall a \in \mathbb{R}_\infty$ and $a + (-\infty) = (-\infty) + a = -\infty = a +^* (-\infty) = (-\infty) +^* a \; \forall a \in \mathbb{R}_{-\infty}$. They only differ when $a \in \{-\infty, \infty\}$ in which case the following anti-symmetry occurs:

$$-\infty + \infty = \infty + -\infty = -\infty \tag{2}$$

$$-\infty +^* \infty = \infty +^* -\infty = \infty. \tag{3}$$

The notions of lattice dependence and independence mirror the notions of linear dependence and independence. Since our application domain is pattern recognition, which deals with real valued vectors, we restrict our discussion to sets of vectors $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^k\} \subset \mathbb{R}^n_{\pm\infty}$ for which $\mathbf{x}^\xi \in \mathbb{R}^n$ for $\xi = 1, \ldots, k$.

Definition 3.1. *If* $\{\mathbf{x}^1, \ldots, \mathbf{x}^k\} \subset \mathbb{R}^n$, *then a* linear minimax combination *of vectors from the set* $\{\mathbf{x}^1, \ldots, \mathbf{x}^k\}$ *is any vector* $\mathbf{x} \in \mathbb{R}^n_{\pm\infty}$ *of form*

$$\mathbf{x} = \mathfrak{S}(\mathbf{x}^1, \ldots, \mathbf{x}^k) = \bigvee_{j \in J} \bigwedge_{\xi=1}^{k} (a_{\xi j} + \mathbf{x}^\xi), \tag{4}$$

where J *is a finite set of indices and* $a_{\xi j} \in \mathbb{R}_{\pm\infty}$ $\forall j \in J$ *and* $\forall \xi = 1, \ldots, k$. *The expression* $\mathfrak{S}(\mathbf{x}^1, \ldots, \mathbf{x}^k) = \bigvee_{j \in J} \bigwedge_{\xi=1}^{k} (a_{\xi j} + \mathbf{x}^\xi)$ *is called a* linear minimax sum.

The similarity with linear sums $\sum_{\xi=1}^{k} a_\xi \mathbf{x}^\xi$ in the vector space $(\mathbb{R}^n, +)$ is obvious; in the vector lattice space $(\mathbb{R}^n_{\pm\infty}, \vee, \wedge)$ the symbols $\bigvee \bigwedge$ (or $\bigwedge \bigvee$) replace the symbol $\sum$ and scalar addition replaces scalar multiplication. Also, if every scalar in the linear sum is zero, then the linear sum is the zero vector. Similarly, according to equation (4), if for every $j \in J$ there exists an index $\xi \in \{1, \ldots, k\}$ such that $a_{\xi j} = -\infty$, then $\mathbf{x} = \mathfrak{S}(\mathbf{x}_1, \ldots, \mathbf{x}^k)$ is the null vector $\omega = (\omega_1, \ldots, \omega_n)'$ of $(\mathbb{R}^n_{-\infty}, \vee, +)$, where $\omega_i = -\infty$ for $i = 1, \ldots, n$. Similarly, if for some $j \in J$ the scalar $a_{\xi j} = \infty$ $\forall \xi = 1, \ldots, k$, then $\mathbf{x} = \omega^*$, the null vector of $(\mathbb{R}^n_\infty, \wedge, +)$ whose coordinates are all ∞. Finally, in the definition of a linear minimax sum one can choose either expression $\bigvee_{j \in J} \bigwedge_{\xi=1}^{k} (a_{\xi j} + \mathbf{x}^\xi)$ or $\bigwedge_{i \in I} \bigvee_{\xi=1}^{k} (b_{\xi i} + \mathbf{x}^\xi)$ as one can be transformed into the other for appropriate indexing sets J and I [7].

The set of all linear minimax sums of vectors from $\{\mathbf{x}^1, \ldots, \mathbf{x}^k\}$ is the *linear minimax span* of the vectors and will be denoted by $LMS(\mathbf{x}^1, \ldots, \mathbf{x}^k)$. The subset of all real valued vectors in $LMS(\mathbf{x}^1, \ldots, \mathbf{x}^k)$ is denoted by $LMS_{\mathbb{R}}(\mathbf{x}^1, \ldots, \mathbf{x}^k)$.

Hence $LMS_{\mathbf{R}}(\mathbf{x}^1,\ldots,\mathbf{x}^k) = LMS(\mathbf{x}^1,\ldots,\mathbf{x}^k)\cap\mathbb{R}^n$ or, equivalently, $LMS_{\mathbf{R}}(\mathbf{x}^1,\ldots,\mathbf{x}^k) = \{\mathbf{x}\in\mathbb{R}^n : \mathbf{x} = \mathfrak{S}(\mathbf{x}^1,\ldots,\mathbf{x}^k)\}$.

Linear minimax sums provide for the definitions of lattice dependence and lattice independence that are in close analogy with these concepts as defined in linear algebra.

Definition 3.2. *Suppose* $X = \{\mathbf{x}^1,\ldots,\mathbf{x}^k\} \subset \mathbb{R}^n$. *A vector* $\mathbf{x}\in\mathbb{R}^n$ *is* lattice dependent *on* X *if and only if* $\mathbf{x} = \mathfrak{S}(\mathbf{x}^1,\ldots,\mathbf{x}^k)$ *for some linear minimax sum of vectors from* X. *The vector* $\mathbf{x}$ *is said to be* lattice independent *of* X *if and only if it is not lattice dependent on* X.

The set X *is said to be* lattice independent *if and only if* $\forall\lambda\in\{1,\ldots,k\}$, $\mathbf{x}^\lambda$ *is lattice independent of* $X\backslash\{\mathbf{x}^\lambda\} = \{\mathbf{x}^\xi\in X : \xi\neq\lambda\}$.

The fixed point set of W_{XX} is the set $F(X) = \{\mathbf{x}\in\mathbb{R}^n : W_{XX}\boxdot\mathbf{x} = \mathbf{x}\}$. It is easy to verify that $\{\mathbf{x}\in\mathbb{R}^n : W_{XX}\boxdot\mathbf{x} = \mathbf{x}\} = \{\mathbf{x}\in\mathbb{R}^n : M_{XX}\boxtimes\mathbf{x} = \mathbf{x}\}$. Hence, W_{XX} and M_{XX} share the same fixed point set. We now summarize some important properties of $F(X)$ that were proven in [7]:

1. If $\mathbf{x},\mathbf{y}\in F(X)$, then $(a+\mathbf{x})\in F(X)$, $(a+\mathbf{x})\vee(b+\mathbf{y})\in F(X)$, and $(c+\mathbf{x})\wedge(d+\mathbf{y})\in F(X)$ $\forall a,b,c,d\in\mathbb{R}$.
2. If $\mathbf{x}\in\mathbb{R}^n$, then $\mathbf{x}$ is a fixed point of W_{XX} if and only if $\mathbf{x}$ is lattice dependent on X.
3. If $X = \{\mathbf{x}^1,\ldots,\mathbf{x}^k\}\subset\mathbb{R}^n$ and $Y = \{\mathbf{y}^1,\ldots,\mathbf{y}^k\}\subset\mathbb{R}^n$, where $\mathbf{y}^\xi = a_\xi + \mathbf{x}^\xi$ and $a_\xi\in\mathbb{R}$ for $\xi = 1,\ldots,k$, then X is lattice independent if and only if Y is lattice independent. Furthermore, $F(X) = F(Y)$.
4. $F(X)$ is a prismatic beam in $\mathbb{R}^n$; that is, $F(X) = \bigcup_{a\in\mathbf{R}}(a+\mathfrak{B})$, where $\mathfrak{B} = F(X)\cap\{\mathbf{x}\in\mathbb{R}^n : x_n = 0\}$ is a convex polytope.
5. If $\mathbf{x}\in\mathfrak{B}$ and $L_\mathbf{x}\subset\mathbb{R}^n$ is the line defined by $L_\mathbf{x} = \{a+\mathbf{x} : a\in\mathbb{R}\}$, then $L_\mathbf{x}$ has direction $\mathbf{e} = \frac{\sum_{j=1}^n \mathbf{e}^j}{\|\sum_{j=1}^n \mathbf{e}^j\|}$, where $e_i^j = 1$ if $i = j$ and $e_i^j = 0$ if $i\neq j$. In particular, the convex beam $F(X) = \bigcup_{\mathbf{x}\in\mathfrak{B}} L_\mathbf{x}$ has direction $\mathbf{e}$.
6. $F(X)$ is a convex sublattice of $(\mathbb{R}^n,\vee,\wedge)$.

Properties one through six provide for a complete algebraic as well as geometric description of $F(X)$ with the exception that the dimensionality of $F(X)$ is not specified. The reason for this is that if $X = \{\mathbf{x}^1,\ldots,\mathbf{x}^k\}\subset\mathbb{R}^n$, then it is possible to have $dim F(X) < k$ as well as $dim F(X) > k$ even if X is lattice independent. Of course, we always have $dim F(X)\leq n$, the dimension of the space. Moreover, it is even possible to have $k > n$ when X is lattice independent. This is in contrast to sets of linearly independent vectors as well as affinely independent sets of vectors. The notion of *strong lattice independence* circumvents these undesirable properties.

Definition 3.3. *A set of lattice independent vectors* $\{\mathbf{x}^1,\ldots,\mathbf{x}^k\}\subset\mathbb{R}^n$ *is said to be* strong lattice independent *if and only if for every* $\lambda\in\{1,\ldots,k\}$ *there exists an index* $j_\lambda\in\{1,\ldots,n\}$ *such that*

$$x_{j_\lambda}^\xi - x_i^\xi \leq x_{j_\lambda}^\lambda - x_i^\lambda \tag{5}$$

$\forall i\in\{1,\ldots,n\}$ *and* $\forall\xi\in\{1,\ldots,k\}$.

The next set of properties have recently been established as a result of exploring the relationships between affine independence and strong lattice independence [7, 8].

7. Any two lattice independent vectors are strongly lattice independent.
8. If $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^k\} \subset \mathbb{R}^n$ is strongly lattice independent, then X is affinely independent.
9. If $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^k\} \subset \mathbb{R}^n$ is strongly lattice independent, then $k \leq dim F(X) \leq n$.
10. If $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^k\} \subset \mathbb{R}^n$ is strongly lattice independent, then the points of X are extreme points of the convex set $F(X)$.
11. If $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^k\} \subset \mathbb{R}^n$, then the vectors corresponding to the columns of W_{XX} satisfy inequality (5).
12. If $V \subset W_{XX}$ is the smallest set of column vectors of W_{XX} satisfying $F(V) = F(X)$, then V is strongly lattice independent.

According to property eight, every strongly lattice independent set of vectors is affinely independent. The converse is generally not true; e.g., if $\mathbf{x}^1 = (0, 0)'$, $\mathbf{x}^2 = (1, 1)'$, then $\{\mathbf{x}^1, \mathbf{x}^2\}$ is affinely independent, but not strong lattice indendent since $\mathbf{x}^2 = 1 + \mathbf{x}^1$. More importantly, the above properties have some significant implications in the theory of linear spectral mixing as employed in hyperspectral and multispectral image analysis. There exist effective methods for deriving the set V mentioned in property twelve [8]. According to property eight, V is affinely independent. As a result, there now exist a computationally efficient way of computing endmembers, which correspond to affinely independent points in hyperspectral image cubes.

On the down side, the above listed properties seem of little help in overcoming the inherent drawbacks of lattice correlation matrix memories in the presence of noise. Figure 1 provides a simple visual example that exposes the core of the problem. Here we are dealing with four patterns $X = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4\} \subset \mathbb{R}^2$. The fixed point set $F(X)$ of W_{XX} is represented as the shaded infinite strip. Since $\tilde{\mathbf{x}}^2 \in F(X)$, $W_{XX} \boxtimes \tilde{\mathbf{x}}^2 = \tilde{\mathbf{x}}^2 = M_{XX} \boxtimes \tilde{\mathbf{x}}^2$. Hence, even a minutely distorted version $\tilde{\mathbf{x}}^2$ of $\mathbf{x}^2$ will not be recovered by these lattice matrix memories whenever $\tilde{\mathbf{x}}^2 \in F(X)$. The point $\mathbf{x}^3$ is in the interior of $F(X)$. Thus *any* distorted version of $\tilde{\mathbf{x}}^3$, no matter how small the distortion, will ever be correctly recalled. Figure 1 indicates this for a distorted version of $\tilde{\mathbf{x}}^3$. The arrows in the figure indicate the orbits of points under the action of W_{XX}. Since the kernel method does not affect the fixed point set $F(X)$, these problems cannot be resolved through the use of the kernel method.

4 Dendritic computing within the lattice domain

In order to take advantage of recent advances in neurobiology and the biophysics of neural computation and to nudge the field of ANNs back to its original roots, we proposed a model of single neuron computation that takes into account the

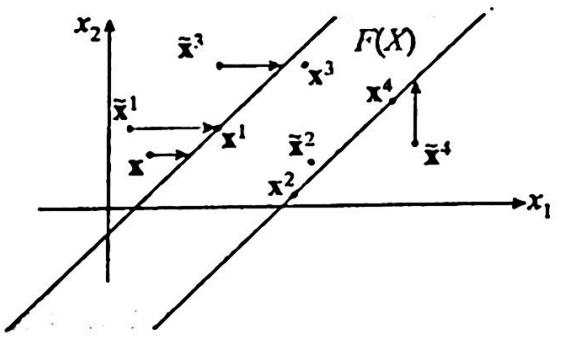

Fig. 1. The orbits of points when using the transform W_{XX}: the set of fixed points of W_{XX} is indicated by the shaded area, while the points resting at the end of the arrows are transformed under the action of W_{XX} to points at the tips of the arrrows.

computation performed by dendrites [3]. Extrapolating on this model, we constructed a single-layer, feedforward neural network based on dendritic computing within the lattice domain [4]. In this model, a set of n input neurons $N_1, \ldots, N_n$ provides information through its axonal arborization to the dendritic trees of a set of m neurons $M_1, \ldots, M_m$. Explicitely, the state value of a neuron N_i $(i = 1, \ldots, n)$ propagates through its axonal tree all the way 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^{ℓ}_{ijk}, where the superscript $\ell \in \{0, 1\}$ distinguishes between *excitatory* $(\ell = 1)$ and *inhibitory* $(\ell = 0)$ input to the dendrite (see also Figure 2). 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

$$\tau^j_k(\mathbf{x}) = p_{jk} \bigwedge_{i \in I(k)} \bigwedge_{\ell \in L(i)} (-1)^{1-\ell} \left(x_i + w^{\ell}_{ijk}\right) \ , \tag{6}$$

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.

The value $\tau^j_k(\mathbf{x})$ 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

$$\tau^j(\mathbf{x}) = p_j \bigwedge_{k=1}^{K_j} \tau^j_k(\mathbf{x}) \ , \tag{7}$$

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, while $p_j = -1$ means that the cell rejects the received input. The *next* state of M_j is then determined by an activation

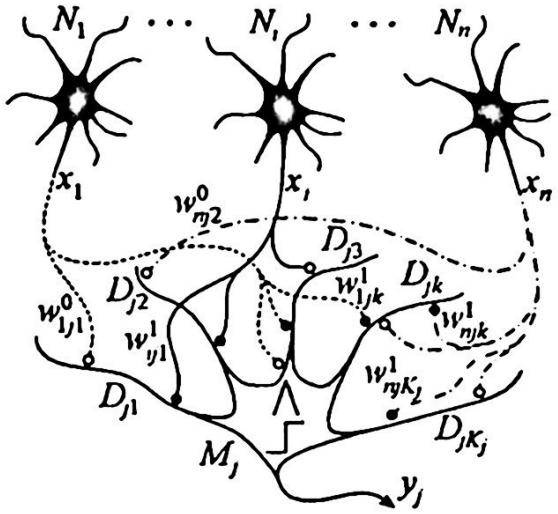

Fig. 2. Morphological perceptron with dendritic structure. Terminations of excitatory and inhibitory fibers are marked with • and o, respectively. Symbol D_{jk} denotes dendrite k of M_j and K_j its number of dendrites. Neuron N_i can synapse D_{jk} with excitatory or inhibitory fibers, e.g. weights w^1_{ijk} and w^0_{nj2} respectively denote excitatory and inhibitory fibers from N_1 to D_{jk} and from N_n to D_{j2}.

function f, namely $y_j = f\left(\tau^j(\mathbf{x})\right)$. Figure 2 provides a graphical representation of this model.

A modification of the lattice based perceptron led to a novel auto-associative memory [6]. A slight modification of auto-associative memory yields a hetero-associative memory which is robust in the presence of noise. Let $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^k\} \subset \mathbb{R}^n$ and $Y = \{\mathbf{y}^1, \ldots, \mathbf{y}^k\} \subset \mathbb{R}^m$ denote the sets of desired association patterns, and $N_1, \ldots, N_n$ the set of sensory (input) neurons that receive input $\mathbf{x}$ from the space $\mathbb{R}^n$ with N_i receiving input x_i. The input neurons will propagate their input values x_i to a set of hidden neurons $H_1, \ldots, H_k$, where each H_j has exactly one dendrite. Every input neuron N_i has exactly two axonal fibers terminating on the dendrite of H_j. The weights of the terminal fibers of N_i terminating on the dendrite of H_j are given by

$$w^\ell_{ij} = \begin{cases} -(x^j_i - \alpha) \text{ if } \ell = 1, \\ -(x^j_i + \alpha) \text{ if } \ell = 0, \end{cases} \tag{8}$$

where $i = 1, \ldots, n$ and $j = 1, \ldots, k$. The parameter $\alpha > 0$ is a user defined noise parameter that must satisfy the inequality $\alpha < \frac{1}{2} d_{\min}$, where

$$d_{\min} = \min\{d(\mathbf{x}^\xi, \mathbf{x}^\gamma) : \xi < \gamma,\ \xi, \gamma \in \{1, \ldots, k\}\} \tag{9}$$

and $d(\mathbf{x}^\xi, \mathbf{x}^\gamma)$ denotes the Chebyshev (checkerboard) distance between the patterns $\mathbf{x}^\xi$ and $\mathbf{x}^\gamma$ defined by $d(\mathbf{x}^\xi, \mathbf{x}^\gamma) = \max\left\{\left|x^\xi_i - x^\gamma_i\right| : i = 1, \ldots, n\right\}$. For a given input $\mathbf{x} \in \mathbb{R}^n$, the dendrite of the hidden unit H_j computes

$$\tau^j(\mathbf{x}) = \bigwedge_{i=1}^{n} \bigwedge_{\ell=0}^{1} (-1)^{1-\ell} \left(x_i + w^\ell_{ij}\right). \tag{10}$$

The state of the neuron H_j is determined by the hard-limiter activation function

$$f(z) = \begin{cases} 0 & \text{if } z \geq 0, \\ -\infty & \text{if } z < 0. \end{cases} \tag{11}$$

The output of H_j is given by $f\left[\tau^j(\mathbf{x})\right]$ and is passed along its axonal fibers to m output neurons $M_1, \ldots, M_m$. Similar to the hidden layer neurons, each output neuron M_h, where $h = 1, \ldots, m$, 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. The excitatory fiber of H_j terminating on M_h has synaptic weight $v_{jh} = y_h^j$. The computation performed by M_h is given by

$$\tau^h(\mathbf{s}) = \bigvee_{j=1}^{k} (s_j + v_{jh}), \tag{12}$$

where s_j denotes the output of H_j, namely $s_j = f\left[\tau^j(\mathbf{x})\right]$, with f defined in equation (11). The activation function g for each output neuron M_h is the linear identity function $g(z) = z$.

Each neuron H_j will have the output value $s_j = 0$ if and only if $\mathbf{x}$ is an element of the hypercube $B^j = \left\{\mathbf{x} \in \mathbb{R}^n : x_i^j - \alpha \leq x_i \leq x_i^j + \alpha\right\}$ and $s_j = -\infty$ whenever $\mathbf{x} \in \mathbb{R}^n \setminus B^j$. Thus, the output of this network will be $\mathbf{y} = (y_1, \ldots, y_m)' = (y_1^j, \ldots, y_m^j)' = \mathbf{y}^j$ 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 associated with $\mathbf{y}^j$. If the amount of noise exceeds the level α, then the network rejects the input by yielding the output vector $(-\infty, \ldots, -\infty)'$. Obviously, each uncorrupted pattern $\mathbf{x}^\xi$ will be associated with $\mathbf{y}^\xi$.

A further improvement of this model is to make the noise parameter α dependent on each $\xi = 1, \ldots, k$ and each coordinate $i = 1, \ldots, n$. This will allow more freedom in order to taylor the net to a specific problem. In particular, one can define the weights

$$w_{ij}^\ell = \begin{cases} -\left(x_i^j - \alpha_i^1(j)\right) & \text{if } \ell = 1, \\ -\left(x_i^j + \alpha_i^0(j)\right) & \text{if } \ell = 0, \end{cases} \tag{13}$$

In this way, the hyperboxes are not centered at $\mathbf{x}^\xi$ allowing for various schemes for creating large noise surrounds about a pattern. Care needs to be taken that the boxes do not overlap. Figure 3 shows an example of such generalized boxes that will associate each point in the box $B^j = \left\{\mathbf{x} \in \mathbb{R}^n : x_i^j - \alpha_i^1(j) \leq x_i \leq x_i^j + \alpha_i^0(j)\right\}$ with the pattern $\mathbf{y}^j$. The figure illustrates the superiority of this model over the lattice correlation memory.

To illustrate the performance of this associative memory, we use a visual example consisting of the associated image pairs $P = \{\mathbf{p}^1, \mathbf{p}^2, \mathbf{p}^3\}$ and $Q = \{\mathbf{q}^1, \mathbf{q}^2, \mathbf{q}^3\}$ shown in Figure 4. Each $\mathbf{p}^\xi$ is a 50×50-pixel 256-gray scale image, whereas each $\mathbf{q}^\xi$ is a 30×50-pixel 256-gray scale image, where $\xi = 1, 2, 3$. Using the standard row-scan method, each pattern image $\mathbf{p}^\xi$ and $\mathbf{q}^\xi$ was converted

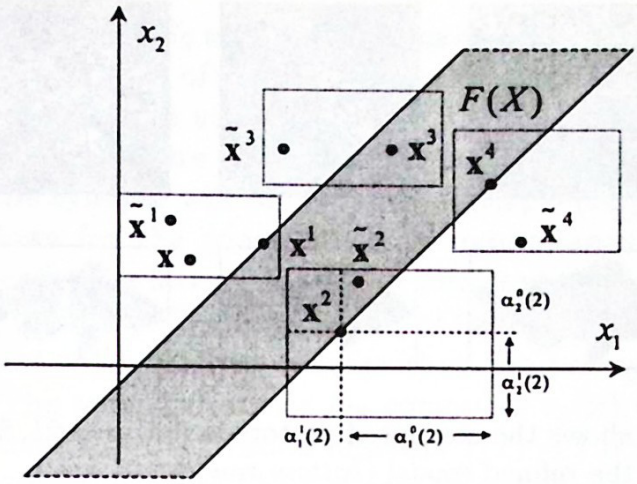

Fig. 3. Boxes with variable noise parameters $\alpha_i^\ell(j)$. Each pattern in the box containing $\mathbf{x}^j$ will be associated with $\mathbf{y}^j$.

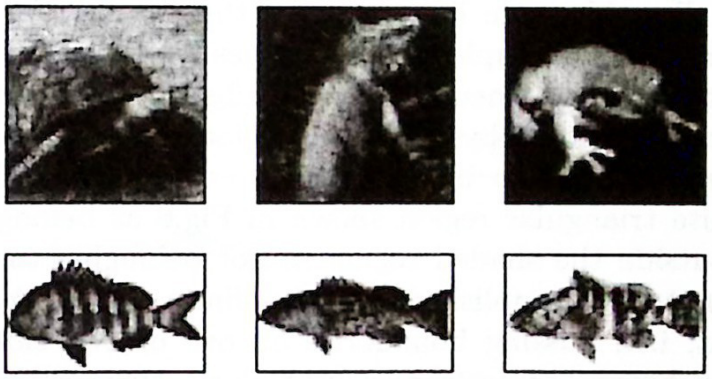

Fig. 4. The *top row* depicts the images $\mathbf{p}^1, \mathbf{p}^2, \mathbf{p}^3$, which were converted into the prototype patterns of the set $X = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3\}$, while the *bottom row* shows the images used to generate the corresponding association patterns from the set $Y = \{\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3\}$.

into a pair of pattern vectors $\mathbf{x}^\xi = \left(x_1^\xi, \ldots, x_{2500}^\xi\right)'$ and $\mathbf{y}^\xi = \left(y_1^\xi, \ldots, y_{1500}^\xi\right)'$ by defining $x_{50(r-1)+c}^\xi = p^\xi(r,c)$ for $r, c = 1, \ldots, 50$ and, respectively, $y_{50(r-1)+c}^\xi = q^\xi(r,c)$ for $r = 1, \ldots, 30$ and $c = 1, \ldots, 50$. Thus, $X = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3\} \subset \mathbb{R}^{2500}$ and $Y = \{\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3\} \subset \mathbb{R}^{1500}$.

The patterns illustrated in the top row of Fig. 5 were obtained by distorting 100% of the vector components of each $\mathbf{x}^\xi$ randomly within the noise level α_ξ. To simplify the algorithm, each $\alpha(\xi)$ only depends only on ξ and not on individual coordinates. In particular, $\alpha_\xi < \frac{1}{2} min\{d(\mathbf{x}^\xi, \mathbf{x}^\gamma) : \gamma \in K(\xi)\}$, where $K(\xi) = \{1, \ldots, k\} \setminus \{\xi\}$. Figure 5 shows that the refined model achieves perfect recall association, whereas the previous model which uses one single constant α does not with this amount of noise.

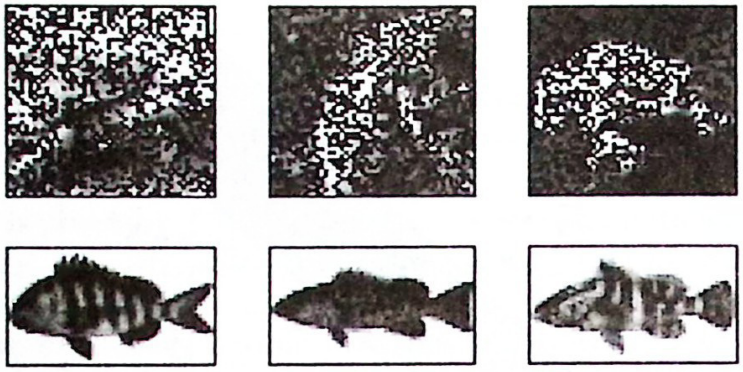

Fig. 5. The *top row* shows the corrupted prototype patterns $\tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}^2, \tilde{\mathbf{x}}^3$, which are correctly associated by the refined model (*bottom row*).

5 New avenues and open questions

The number of open problems associated with lattice based neural networks far exceeds the allowable page limitation for this article. For this reason, we can only briefly mention a couple of new avenues for future investigations. One obvious limitation of lattice neural networks discussed in the literature thus far is the reliance on the Chebyshev distance for defining axonal weights. For example, suppose we want to build a neural net that classifies all points in the shaded infinite triangular region shown in Fig.6 as belonging to one class and all points outside the shaded region as not belonging to that class. This could obviously not be accomplished using any finite union of boxes (even boxes of infinite extent; i.e., missing boundaries on one or two sides). In this case we again need to borrow ideas from neurobiology. Neuronal signals consist of short electrical pulses called spike trains. If two terminal axonal fibers from two presynaptic neurons terminate in close vicinity, such as on the same dendritic spine, a temporal summation of received spike trains seem to be taking place.

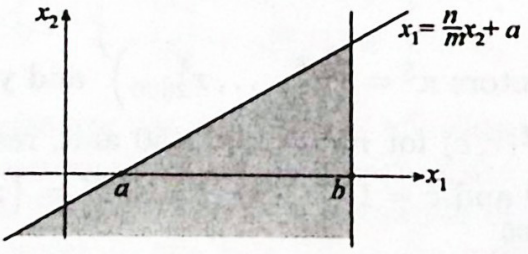

Fig. 6. The shaded region bounded by the lines $x_1 = \frac{n}{m}x_2 + a$ and $x_1 = b$. The region cannot be specified by a lattice neural net employing the Chebyshev distance.

Incorporating these observations into the lattice model takes the following form: suppose that two neurons N_1 and N_2 transmit information x_1 and x_2, respectively, to the *same* terminal location (spine, etc.) on a given dendrite.

Further suppose that N_1 and N_2 send spike trains of length m and n, respectively, and that the axonal branches of N_1 and N_2 terminating on the same terminal (spike) have respective weights w_1 and w_2. If N_2 is a inhibitory neuron, then its spike train will directly interfere with the spike train of N_1. If N_1 is excitatory, then the signal at the synapse will sum to $mx_1 + mw_1 - (nx_2 + nw_2)$. This type of computation allows for the construction of hyperplanes with any rational orientation. Specifically, the lattice neural net depicted in Fig.7 will classify the region shown in Fig.6 exactly. The dendrite D_1 of the output neuron M computes $\tau_1(\mathbf{x}) = -[mx_1 + m(-b)]$, since we assume that the postsynaptic response is negative; i.e., -1. The second dendrite D_2 computes $\tau_2(\mathbf{x}) = mx_1 + m(-a) - (nx_2)$. Using the hard-limiter activation function $f(\tau) = 1$ if $\tau \geq 0$ and $f(\tau) = 0$ if $\tau < 0$, then the output of M will be 1 if and only if $\tau(\mathbf{x}) = \tau_1(\mathbf{x}) \wedge \tau_2(\mathbf{x}) \geq 0$. But this is the case if and only if $x_1 \geq b$ and $x_1 \geq \frac{n}{m}x_2 + a$. Hence, this network solves the problem posed by the example illustrated in Fig.6.

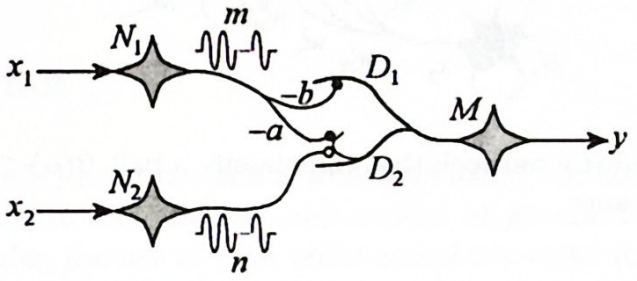

Fig. 7. The spike trains of lengths m and n travelling along the axons of N_1 and N_2, respectively, carry m copies of the input x_1 and n copies of the input x_2.

A particularly interesting application of the summation of a signals impinging on the same synaptic site of a dendrite is the implementation of the city-block distance $d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|$, which reduces to the Hamming distance for boolean patterns. For example, the *ball* $B(\alpha)$ of radius α about the point $\mathbf{x}^\xi \in \mathbb{R}^2$ using the city-block distance is shown in Fig.8. By setting weights $w_{11}^1 = x_2^\xi - x_1^\xi = w_{12}^0$, $w_{13}^1 = -(x_2^\xi + x_1^\xi) = w_{14}^0$, $w_{21}^0 = w_{24}^0 = \alpha$, and $w_{22}^1 = w_{23}^1 = -\alpha$, and assuming that for each $i \in \{1, 2, 3, 4\}$ the terminal fiber defined by w_{1i}^ℓ shares the same synaptic location as the fiber with weight w_{2i}^ℓ, then the simple network shown in Fig.9 has output $f(\tau(\mathbf{x})) = 1$ if and only if $\mathbf{x} \in B(\alpha)$.

Now consider a network that incorporates both the neurons discussed in this section and the neurons defined in the preceding. What training algorithms can be devised for such a network? Such a network should be more powerful than the morphological perceptron discussed earlier. How do you generalize the neuron discussed here by allowing for spike trains of variable lengths during training as well as testing?

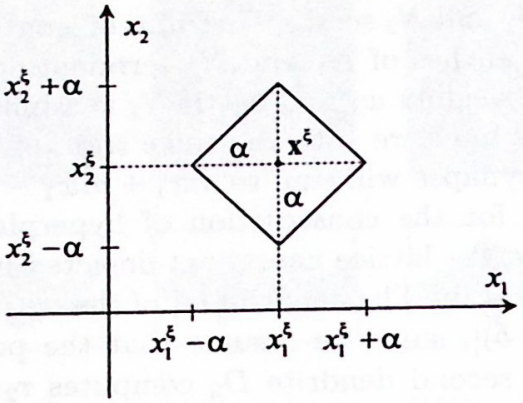

Fig. 8. The ball $B(\alpha) \subset \mathbb{R}^2$ of radius *alpha* and center $\mathbf{x}^\xi$ defined by the city-block distance.

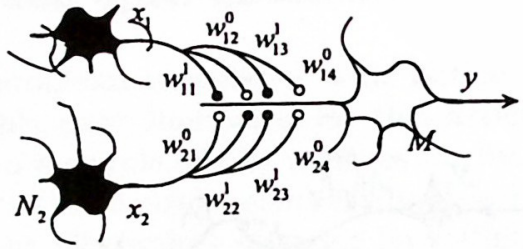

Fig. 9. A simple dendritic network that can classify a ball $B(\alpha) \subset \mathbb{R}^2$ defined in terms of the city-block distance.

References

1. Ritter GX, Sussner P, Diaz de Leon JL (1998) Morphological Associative Memories. IEEE Trans. on Neural Networks 9(2):281–293
2. Ritter GX, Urcid G, Iancu L (2003) Reconstruction of Noisy Patterns Using Morphological Associative Memories. J. of Mathematical Imaging and Vision 19(2):95–111
3. Ritter GX, Urcid G (2003) Lattice Algebra Approach to Single Neuron Computation. IEEE Trans. on Neural Networks 14(2):282–295
4. Ritter GX, Iancu L, Urcid G (2003) Morphological Perceptrons with Dendritic Structure. FUZZ-IEEE2003, St. Louis, Missouri, 1296–1301
5. Kohonen T (1972) Correlation matrix memory. IEEE Trans. on Computers C-21:353–359
6. Ritter GX, Iancu L, Schmalz BS (2004) A new auto-associative memory based on lattice algebra. Proc. of 9th Ibero-American Congress on Pattern Recognition, Puebla, Mexico, LNCS 3287, Springer Verlag-Heidelberg, 148–155
7. Ritter GX, Gader P (2005) Fixed points of lattice transforms and Lattice associative Memories, to appear in Advances in Imaging and Electron Physics, Academic Press.
8. Ritter GX, Selfridge R (2005) A novel approach to the computation of endmembers. (to appear - preprint available from the authors)