

Lattice Based Dendritic Computing: A Biomimetic Approach to ANNs

Gerhard X. Ritter¹ 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`

Abstract. This paper presents an overview of the current status of lattice based dendritic computing. Roughly speaking, lattice based dendritic computing refers to a biomimetic approach to artificial neural networks whose computational aspects are based on lattice group operations. We begin our presentation by discussing some important processes of biological neurons followed by a biomimetic model which implements these processes. We discuss the reasons and rationale behind this approach and illustrate the methodology with some examples. Global activities in this field as well as some potential research issues are also part of this discussion.

1 Introduction

The study of artificial neural networks (ANNs) was originally inspired by advances in neuroscience [1, 2]. However, early research in ANNs came almost to a standstill in the 1970s. A widely published book [3] by Minsky and Papert showed the limitations of the highly touted neural network model known as a perceptron. Probably as much as any other single factor, the efforts of J.J. Hopfield during the early 1980's brought about a profound change in the perception of ANNs within the scientific community. As a well-known physicist of the California Institute of Technology, Hopfield's scientific credentials lent renewed credibility to the field of ANNs which had been badly tarnished by the hype of the mid-1960's. Several applications of Hopfield's early papers include associative or content-addressable memories [4–6]. Since these early days, ANNs have become a major tool in machine learning and artificial intelligence. They have been applied in such diverse areas as pattern recognition and pattern association, robotic control and image processing, speech processing and computer vision, data storage and retrieval, expert systems and many others.

The various ANN models in current use are intimately associated with a particular learning algorithm or learning rule. Thus, we have multilayer perceptrons (MLPs) and back propagation, kernel function based learning such as radial basis

* Corresponding author. G. Urcid thanks SNI-CONACYT for partial financial support, grant # 22036.

function (RBF) neural networks, support vector machines (SVMs), kernel Fisher discriminants (KFDs), and various other hybrid models with similar approaches. Consequently, the approach inspired by biology has been largely abandoned for a more practical approach based on statistics, probability theory, and other mathematical signal processing methods. It is therefore safe to say that these ANNs have very little in common with biological neural networks. It was this observation of the divergence of ANNs from their biological roots that inspired our goal of reversing this trend and led to the creation of a biomimetic model of a neuron [7]. In the next section we provide some basic information about biological neurons and their processes. In Section 3 we present the biomimetic model based on the neural processes discussed in Section 2 and also discuss the reasons and rationale for the basic computational processes. Section 4 provides an application example and a brief discussion of some global studies and applications of DLNNs. The section ends with five open research problems. In the concluding section (Section 5) we appeal to the reader to join our efforts in advancing the frontiers of biomimetic ANNs.

2 Biological Neurons and Their Processes

The term *biomimetic* refers to man-made systems of processes that imitate nature. Thus, a biomimetic approach to ANNs refers to imitating or mimicking biological neural networks. But in order to imitate biological neural structures one has to first understand the morphology and function of the fundamental component of the structure, namely the neuron. A neuron (or nerve cell) is a cell in the animal kingdom and as such contains numerous components common to all animal cells. These include a cell membrane, a cell nucleus, mitochondria, Golgi apparatus, ribosomes, and so on. Just as there are many different type of cells making up the overall structure of an animal, there are many different types of nerve cells making up the nervous system of an animal. These different types of neurons are classified according to their morphological differences such as their dendritic structures as well as their functionality. Nevertheless, every neuron consists of a cell body, called *soma*, and several processes. These processes are of two kinds and are called, respectively, *dendrites* and *axons*. The dendrites, which are usually multiple, conduct impulses toward the body of the cell; the axon conducts from the cell body. Dendrites typically have many branches that create large and complicated trees. Many (but not all) types of dendrites are studded with large numbers of tiny branches called *spines*. Dendritic spines, when present, are the major *postsynaptic* target for *excitatory synaptic* input. The soma and the dendrites constitute the input surface of the neuron. When a neuron fires, then all neurons receiving the fired signal are called the *postsynaptic neurons* while the firing neuron is called the *presynaptic neuron*. When the voltage profile of a fired signal is recorded, it usually consists of a sequence - better known as a *train* - of spikes. It has been conjectured that the number of and distances between the spikes in a train represents the encoded information that the neuron is transmitting to the recipient postsynaptic neuron [10]. The

axon, which usually arises from the opposite pole of the cell at a point called the *axon hillock*, consists of a long fiber whose branches form the *axonal arborization* or *axonal tree*. For some neurons the axon may have branches at intervals along its length in addition to its terminal arborization. The tips of the branches of the axon are called *nerve terminals* or *boutons* or *synaptic knobs*. The axon is the principal fiber branch of the neuron for the transmission of signals to other neurons. Figure 1 shows an image and a typical schematic representation of a biological neuron with its branching processes. An impulse traveling along an

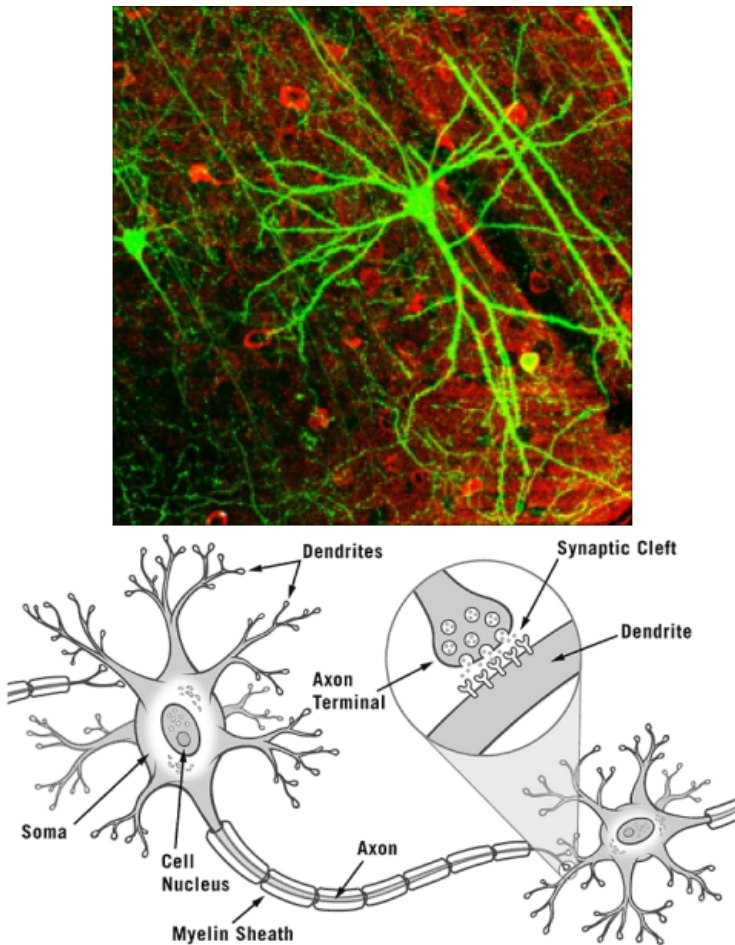


Fig. 1. Merged color image of a biological neuron cell showing dendrites, dendritic trees, soma, axon, and thin terminal branches [8], and schematic drawing of biological neuron cells (pre- and postsynaptic) showing dendrites, dendritic trees, soma with nucleus, axon, myelin sheath, synaptic cleft, and terminal branches with boutons [9]

axon from the axon hillock propagates through the axonal tree all the way to the nerve terminals. The terminals of the branches make contact with the soma and the many dendrites of other neurons. The sites of contact are the *synaptic sites* where the *synapses* take place. The *synapse* is a specialized structure whereby neurons communicate but there is no actual structural union of the two neurons at the synaptic site. The synaptic knob is separated from the surface of the dendrite or soma by an extremely narrow space called the *synaptic cleft*. The exact mechanism of synaptic structures is fairly well understood and there exist two kinds of synapses; *excitatory synapses* which tend to depolarize the postsynaptic membrane and consequently exciting the postsynaptic cell to fire impulses, and *inhibitory synapses* that try to prevent the neuron from firing impulses in response to excitatory synapses. Inhibitory action affects the postsynaptic membrane and lowers its potential [11, 12].

3 The Biomimetic Neuron and Neural Network Model

The number of synapses on a *single* neuron in the cerebral cortex ranges between 500 to 200,000. Most of the synapses occur on the dendritic tree of the neuron, and it is here where information is processed [12–15]. Dendrites make up the largest component in both surface area and volume of the brain. Part of this is due to the fact that pyramidal cell dendrites span all cortical layers in all regions of the cerebral cortex [11, 14, 15]. Thus, when attempting to model artificial brain networks that bear more than just a passing resemblance to biological brain networks, one cannot ignore dendrites (and their associated spines) which can make up more than 50% of the neuron's membrane. This is especially true in light of the fact that some brain researchers have proposed that dendrites and not the neuron are the elementary computing devices of the brain. Neurons with dendrites can function as many, almost independent, functional subunits with each unit being able to implement a rich repertoire of logical operations [13–17]. Possible mechanisms for dendritic computation of such logical functions as XOR, AND, and NOT have been proposed by several researchers [13–15, 19–21]. For a more thorough background in dendritic computing, we refer the reader to [11, 14, 22, 23].

It is for the above observations that a biomimetic model of a neuron needs to include both dendrites and an axon with arborization. Also, the operations of AND, OR, NOT, and XOR are operations common to lattice theory and can be achieved in the dendrites starting at the synapses and accumulating in branches of the dendritic tree. They are just as easy to implement on the gate array level and therefore provide for fast computational results. Additionally, for additive lattice groups the operation of multiplication is generally absent and thus yields extremely fast convergence in lattice based learning algorithms. In light of these observations, we constructed a biomimetic model of a neuron with a dendritic process in which basic lattice operations occur after information transfer at the synaptic sites. In order to describe this model in more detail, it is necessary to briefly discuss the concept a lattice group and the associated lattice computations.

A partially ordered set L is often specified as a pair (L, \leq) , where \leq denotes the partial order. If for any two elements $x, y \in L$ the greatest lower bound or supremum $\sup\{x, y\}$ and least upper bound or infimum $\inf\{x, y\}$ exist, then L is called a *lattice*. If L is a lattice, then it is also common to specify L as a triple (L, \vee, \wedge) , where \vee and \wedge denote the binary operation $x \vee y = \sup\{x, y\}$ and $x \wedge y = \inf\{x, y\}$. By a lattice ordered group we mean a set L with an associated algebraic structure $(L, \vee, \wedge, +)$, where (L, \vee, \wedge) is a lattice and $(L, +)$ is a group with the property that every group translation is isotone; that is, if $x \leq y$, then $a + x + b \leq a + y + b \forall a, b \in L$. Given the set $\mathcal{O} = \{\vee, \wedge, +\}$ of lattice group operations, then the symbols $\oplus, \otimes,$ and \odot will mean that $\oplus, \otimes, \odot \in \mathcal{O}$ but are not explicitly specified operations. Similarly, symbols of the form $\bigoplus, \bigotimes,$ or \bigodot will denote generalized operations derived from $\oplus, \otimes,$ and \odot , respectively. For example, $\bigoplus_{i=1}^n a_i = a_1 \oplus \dots \oplus a_n$. Hence, if $\oplus = \vee$ and $\odot = +$, then $\bigoplus_{i=1}^n a_i = \bigvee_{i=1}^n a_i = a_1 \vee \dots \vee a_n$, and $\bigodot_{i=1}^n a_i = \sum_{i=1}^n a_i = a_1 + \dots + a_n$.

In the dendritic model of ANNs, a finite set of presynaptic neurons N_1, \dots, N_n provides information through its axonal arborization to the dendritic trees of some other finite set of postsynaptic neurons M_1, \dots, M_m . The dendritic tree of a postsynaptic neuron M_j is assumed to consist of a finite number of branches d_{j1}, \dots, d_{jK_j} which contain the synaptic sites upon which the axonal fibers of the presynaptic neurons terminate. The *strength* of the synapse on the k -th dendritic branch d_{jk} ($k \in \{1, \dots, K(j)\}$) which serves as a synaptic site for a terminal axonal branch fiber of N_i is denoted by w_{ijk}^ℓ and is also called its *synaptic weight*. The superscript ℓ is associated with the postsynaptic response that is generated within and in close proximity of the synapse. Specifically, $\ell = 0$ and $\ell = 1$ denote an inhibitory or excitatory postsynaptic response, respectively. It is possible for several axonal fibers to synapse on the same or on different synaptic sites on a given branch d_{jk} , with the former case implying that $w_{ijk}^\ell = w_{hjk}^\ell$. The total response (or output) of d_{jk} to the received input at its synaptic sites is given by

$$\tau_k^j(\mathbf{x}) = p_{jk} \bigoplus_{i \in I(k)} \bigotimes_{\ell \in \mathcal{L}(i)} (-1)^{1-\ell} l(x_i + w_{ijk}^\ell), \tag{1}$$

where $\mathbf{x} = (x_1, \dots, x_n) \in L^n$ with L^n denoting the n -fold Cartesian product of L , $x_i \in L$ denotes the information propagated by N_i via its axon and axonal branches, $\mathcal{L}(i) \subseteq \{0, 1\}$ corresponds to the postsynaptic response generated at the synaptic region to the input received from N_i , and $I(k) \subseteq \{1, \dots, n\}$ corresponds to the set of all presynaptic neurons with terminal axonal fibers that synapse on the k -th dendritic branch of M_j . The value $p_{jk} \in \{-1, 1\}$ marks the final signal outflow from the k -th branch as inhibitory if $p_{jk} = -1$ and excitatory if $p_{jk} = 1$. The value $\tau_k^j(\mathbf{x})$ is passed to the cell body of M_j and the state of M_j is a function of the combined values received from its dendritic structure and is computed as

$$\tau^j(\mathbf{x}) = p_j \bigodot_{k=1}^{K_j} \tau_k^j(\mathbf{x}), \tag{2}$$

where K_j denotes the total number of dendritic branches of M_j and $p_j = \pm 1$ denotes the response of the cell to the received input. Here again $p_j = -1$ means rejection (inhibition) and $p_j = 1$ means acceptance (excitation) of the received input. Figure 2 illustrates the neural pathways from the presynaptic neurons N_i to the postsynaptic neuron M_j . An open circle \circ in Fig. 2 means that the synaptic weight is inhibitory while a solid circle \bullet indicates an excitatory synapse. The value x_i denotes the information transferred from neuron N_i to the synaptic sites of neuron M_j .

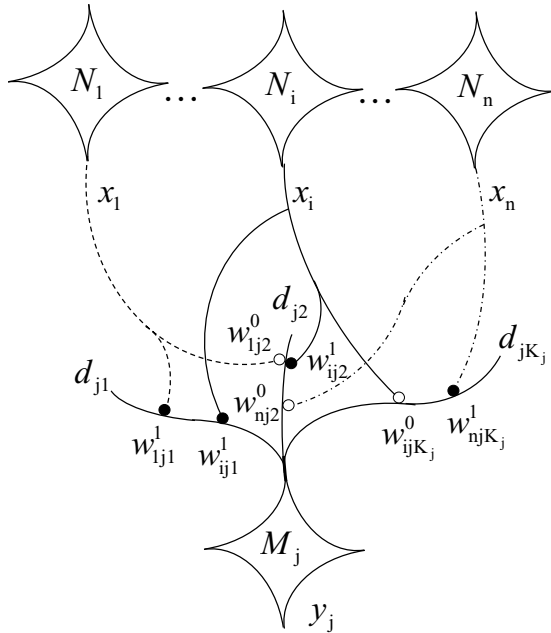


Fig. 2. Terminal branches of axonal fibers originating from the presynaptic neurons N_i make contact with synaptic sites on dendritic branches of M_j . Excitatory and inhibitory inputs are indicated, respectively, by solid (\bullet) and open circles (\circ).

Also, in the general lattice group case, the values ± 1 denote the group identity and its inverse and not necessarily the numbers one and minus one. The prime example of a lattice ordered group is the set \mathbb{R} of real numbers together with the binary operations of the maximum (\vee) and minimum (\wedge) of two numbers, and the group operation of addition; it is denoted by $(\mathbb{R}, \vee, \wedge, +)$ and is the lattice employed in this paper. Thus, Eqs. (1) and (2) assume, respectively, the forms

$$\tau_k^j(\mathbf{x}) = p_{jk} \bigvee_{i \in I(k)} \bigwedge_{\ell \in \mathcal{L}(i)} (-1)^{1-\ell} (x_i + w_{ijk}^\ell) \quad \text{and} \quad \tau^j(\mathbf{x}) = p_j \sum_{k=1}^{K_j} \tau_k^j(\mathbf{x}), \quad (3)$$

where $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, $x_i \in \mathbb{R}$, and $j = 1, \dots, m$. The value $\tau_k^j(\mathbf{x})$ activates the neuron M_j via an activation function f . The activation function

depends on the task assigned to this neuron and can be the identity function, a hard-limiter function, a ramp function, or any other appropriate function. The activated neuron will then fire and provide an output $f[\tau_k^j(\mathbf{x})]$ to postsynaptic neurons of its receptive field.

Figure 3 shows one possible feed-forward structure of such interconnected biomimetic neurons. The mimetic network consists of three layers, namely an input layer N of neurons N_1, \dots, N_n without dendrites but axonal arborization, two hidden layers A and B consisting of neurons A_1, \dots, A_p and B_1, \dots, B_q , respectively, and an output layer M consisting of m neurons M_1, \dots, M_m . Networks of this type are known as *dendritic lattice neural networks* (DLNNs).

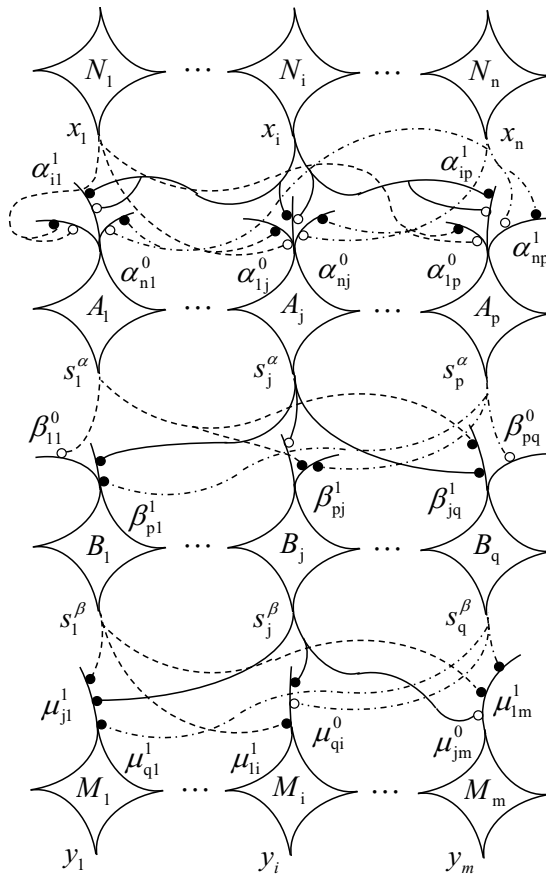


Fig. 3. One possible two-layer structure of a whole or part of a DLNN

4 Examples, Observations, and Open Problems

Although DLNNs are still in their infancy, lattice based neural networks have been around for a couple of decades and investigated as well as applied to various problem domains around the globe [18, 28, 29, 32–45, 47–52]. These early lattice based neural network led to a deeper understanding of how a variety of problems could be solved using only lattice group operations. However, certain limitations of these networks, a majority of which were matrix based correlation approaches, also came to light. Many of these problems are easily removed when using the dendritic approach. In particular problems in the area of associative memories such as discussed in [40, 42, 44, 46, 49] were readily overcome using DLNNs [59, 60, 63, 66].

To gain better insight in dendritic computing we consider a specific example. Suppose we have a set $X = \{\mathbf{x}^1, \dots, \mathbf{x}^K\} \subset \mathbb{R}^n$ of exemplar patterns. Then given a vector $\mathbf{x} \in \mathbb{R}^n$, we would like to know whether \mathbf{x} is “close” to some $\mathbf{x}^j \in X$ in terms of the L_1 metric d_1 . To solve this problem with a DLNN we know that we need n input neurons N_1, \dots, N_n since inputs will be real valued vectors of length n . Since the number of exemplars is K , K neurons A_1, \dots, A_K are required in the first hidden layer, with each neuron A_j storing the structure of pattern \mathbf{x}^j as synaptic weights at the synaptic sites in its dendritic branches. More precisely, for a given input pattern $\mathbf{x} = (x_1, \dots, x_n)$ the i -th neuron N_i will assume as its value the i -th coordinate x_i of \mathbf{x} and will propagate this value through its axonal arborization to the dendrites of A_j for $j = 1, \dots, K$. The dendritic tree of each hidden neuron A_j has n single branches d_{j1}, \dots, d_{jn} , and each neuron N_i has two axonal fibers terminating on the synaptic sites located on the corresponding branch d_{ji} of the hidden layer neuron A_j as depicted in Fig. 3. Observe that in this formulation the dendritic branch counter k equals i , making the extra counter k unnecessary. The two synaptic weights associated with the two synaptic sites of d_{ji} are defined by $a_{ij}^\ell = -x_i^j$ for $\ell = 0, 1$. The output of each dendritic branch is denoted by $\tau_i^j(\mathbf{x})$. Here we use the formula given by Eq. 3 in order to compute this value. Setting $p_{jk} = -1$ and using the fact that $I(k) = I(i) = \{i\}$, Eq. 3 reduces to

$$\tau_i^j(\mathbf{x}) = - \bigwedge_{\ell=0}^1 (-1)^{1-\ell} (x_i + a_{ij}^\ell) = (x_i - x_i^j) \vee (x_i^j - x_i). \tag{4}$$

It follows from Eq. 4 that $\tau_i^j(\mathbf{x}) = 0 \Leftrightarrow x_i = x_i^j$ and $\tau_i^j(\mathbf{x}) > 0 \Leftrightarrow x_i \neq x_i^j$. The value $\tau_i^j(\mathbf{x})$ is passed to the cell body of A_j and its state is a function of the combined values received from its dendritic structure. This state is computed using Eq. 3 with $p_j = 1$. Specifically, we have

$$\tau_A^j(\mathbf{x}) = \sum_{i=1}^n \tau_i^j(\mathbf{x}) = \sum_{i=1}^n [(x_i - x_i^j) \vee (x_i^j - x_i)] = \sum_{i=1}^n |x_i - x_i^j|. \tag{5}$$

It follows that each neuron A_j in the A -layer computes the L_1 -distance between the input pattern \mathbf{x} and the j -th exemplar pattern \mathbf{x}^j , i.e., $\tau_A^j(\mathbf{x}) = d_1(\mathbf{x}, \mathbf{x}^j)$.

A threshold for defining an activation function for the A -layer neurons can be user defined or obtained during training, which depends on the specific problem at hand. Employing the network just described as a first part of a two hidden layer network similar to the one shown Fig. 3, we designed a DLNN based heteroassociative memory that proved to be extremely robust in the presence of various types of noise [51, 63]. For example, five predator images each associated with a specific prey are shown in Fig. 4.



Fig. 4. Five predators in the first row and corresponding preys in the second row

The predator exemplar features are stored as synaptic weights in the synaptic sites of the dendrites of the A -layer neurons while the features of the prey exemplars, $Y = \{\mathbf{y}^1, \dots, \mathbf{y}^5\}$, were stored as synaptic weights in the M -layer neurons. For a given input \mathbf{x} , the A -layer neurons compute the L_1 distance between \mathbf{x} and each exemplar \mathbf{x}^j . The output of the A -layer serves as input to the B -layer neurons whose task is to find the minimum of the five distances $d_1(\mathbf{x}, \mathbf{x}^j)$ where $j = 1, \dots, 5$. The minimum distance values are then forwarded by the axons of the B -layer neurons to the dendrites of the output neurons of M . The output of the M -layer will be \mathbf{x}^j such that $d_1(\mathbf{x}, \mathbf{x}^j)$ is minimal. It is noteworthy to mention that the activation function for the A layer neurons is a ramp function while the activation function for the B layer neuron is a hard limiter function. These two functions correspond roughly to the measured excitatory post-synaptic potential (EPSP) of cerebellar interneurons (ramp function) and pyramidal neurons (hard limiter) [64, 65].

The inputs to the heteroassociative memory predator-prey DLNN were noisy and corrupted versions of the predator images. We simulated noise pattern acquisition by increasing and decreasing image contrast, approximating linear camera motion, applying circular averaging filters, employing the morphological operations of dilation and erosion with different structuring elements, and by using Gaussian as well as uniform noise. Figure 5 shows some of the tested image corruption changes. Different types of noise corruption have been applied to different images. The first column presents motion blur, the 2nd shows Gaussian noise, the 3rd displays the application of a circular averaging filter, the 4th illustrates a morphological erosion with a line as structuring element, and the 5th presents a morphological dilation with a structuring element of elliptical shape.



Fig. 5. Top row displays the exemplar image patterns (predators), 2nd through the 4th column below a given predator show the increase in noise level or image corruption of a predator and bottom row illustrates the DLNN recall performance of a prey when presented with a noisy predator image above it

In many applications more accurate results can be obtained by using more than one type of metric. Examples of pattern classification DLNNs that use two lattice metrics are given in [69]. Various other DLNNs have been proposed to solve a variety of problems in areas ranging from the early detection of Alzheimer's disease to the segmentation of blood vessels in retinal images [58, 61, 62, 67, 68, 71]. An intriguing application a DLNN in the area of formal concept analysis (FCA) was recently proposed by researchers at CINVESTAV-Guadalajara [72]. A major aim of FCA is to support the rational communication of humans by mathematically developing appropriate structures which can be logically activated [73]. Applications of FCA are manifold and may be a key technique in the storage, retrieval, and analysis of big data [74].

In spite of all the above mentioned research and applications of DLNNs, dendritic computing is not yet part of mainstream ANNs even though any Google search on dendritic computing shows that there is a consensus among neuroscientists that basic information changes and calculations occur in the dendrites and their synapses. However, this is good news for the researcher interested in using some of these novel biological discoveries in order to create more powerful

biomimetic mathematical models of neural networks. Since the theory of DLNNs is still in its infancy, the field of research remains wide open for further exploration and major breakthroughs. With this in mind, we conclude this section by listing five important areas that are in need of further exploration.

1. **Extension to fuzzy DLNNs.** The fields of fuzzy set theory and lattice theory are mathematically closely related [47, 48]. The idea of employing fuzzy techniques in DLNNs was first discussed in [30, 70], but further exploration of the utility of fuzzy set theory in DLNNs is still needed and have the potential of new breakthroughs.
2. **Multitasking DLNNs.** This area has yet to be explored. Specifically, a DLNN whose subunits solve different tasks and may have different sensory input neurons (e.g., sound, visual, smell, etc.). In this setup the different subunits interact in order to solve a task that depends on the outcome of the various tasks solved by the subunits.
3. **Research in the utility of FCA in DLNNs.** An integral part of FCA is the *concept lattice of a context*. This lattice provides a possible link to dendritic lattice computing. Since this constitutes a new area of research, it presents an excellent opportunity to obtain novel results fairly quickly.
4. **Training and learning paradigms for DLNNs.** Learning in DLNNs means the generation of axonal fibers, dendritic branches, synaptic sites and synaptic weights for these sites. Although several learning and training methods for DLNNs have been proposed and implemented, they mostly rely on geometric considerations and are collectively known as hyperbox approaches [53, 54, 56, 58]. Thus far, there do not exist dynamical systems approaches, strategies relying on statistical methods, or some other innovative approaches. This is a difficult and challenging problem, but the payoff would be very big for any truly novel learning paradigm.
5. **Exploring the utility of spike trains in DLNNs.** Axonal spike trains are a common phenomenon of firing neurons. Recent research supports the idea that the position of spikes, gaps, and spike bursts within a small time interval Δt (measured in milliseconds) is the key to understanding the coded language by which neurons communicate. For this reason it is of great interest to know the utility of incorporating the current theory of spike trains in the biomimetic model. First attempts showed great promise [57] and we believe that it is a worthwhile endeavor to further pursue this area of research.

5 Conclusions

We presented a brief overview of a biomimetic model for ANNs which included reasons and rationale in support of this model as well as application examples. Our presentation concluded with a list of five critically important open problem areas. We hope that our discussion and list of problems will generate sufficient interest to entice other researchers to join our efforts in advancing the frontiers of biomimetic ANNs.

References

1. McCulloch, W., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5, 115–133 (1943)
2. Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65, 386–408 (1958)
3. Minsky, M.L., Papert, S.A.: *Perceptrons*. MIT Press, Cambridge (1969)
4. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Academy of Sciences USA* 79, 2554–2558 (1982)
5. Hopfield, J.J.: Neurons with graded response have collective computational properties like those of two state neurons. *Proc. Nat. Academy of Sciences USA* 81, 3088–3092 (1984)
6. Hopfield, J.J., Tank, D.W.: Computing with neural circuits. *Science* 233, 625–633 (1986)
7. Ritter, G.X., Urcid, G.: Lattice algebra approach to single neuron computation. *IEEE Trans. on Neural Networks* 14(2), 282–295 (2003)
8. Lee, W.C.A., et al.: Dynamic remodeling of dendritic arbors in GABAergic interneurons of adult visual cortex. *PLoS Biology* 4(2), 271–280 (2006)
9. The Third Source: A message of hope for education. *The Neuron*, in Media Gallery (2010), <http://www.thethirdsource.org/media/charts-and-graphs>
10. Dayan, P., Abott, L.F.: *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, Boston (2006)
11. Eccles, J.C.: *The Understanding of the Brain*. McGraw-Hill, New York (1977)
12. Sejnowski, T.J., Qian, N.: Synaptic integration by electro-diffusion in dendritic spines. In: McKenna, T., Davis, J., Zornetzer, S.F. (eds.) *Single Neuron Computation*. Academic Press, New York (1992)
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*, pp. 605–636. John Wiley & Sons, New York (1987)
14. Koch, C., Segev, I. (eds.): *Methods in Neuronal Modeling: From Synapses to Networks*. MIT Press, Cambridge (1989)
15. Segev, I.: Dendritic Processing. In: Arbib, M.A. (ed.) *The Handbook of Brain Theory and Neural Networks*, pp. 282–289. MIT Press, Boston (1998)
16. Mel, B.W.: Why have Dendrites? A Computational Perspective. In: Dendrites, S.G., Spruston, N., Häusser, M.D. (eds.), pp. 271–289. Oxford University Press (1999)
17. Wei, D.S., et al.: Compartmentalized and binary behavior of terminal dendrites in hippocampal pyramidal neurons. *Science* 293, 2272–2275 (2001)
18. Won, Y., Gader, P.D., Coffield, P.: Morphological shared-weight networks with applications to automatic target recognition. *IEEE Trans. Neural Networks* 8(5), 1195–1203 (1997)
19. Mel, B.W.: Synaptic integration in excitable dendritic trees. *Journal of Neurophysiology* 70, 1086–1101 (1993)
20. Holmes, W.R., Rall, W.: Electronic Models of Neuron Dendrites and Single Neuron Computation. In: McKenna, T., Davis, J., Zornetzer, S.F. (eds.) *Single Neuron Computation*, pp. 7–25. Academic Press, New York (1992)
21. Shepherd, G.M.: Canonical Neurons and their Computational Organization. In: McKenna, T., Davis, J., Zornetzer, S.F. (eds.) *Single Neuron Computation*, pp. 27–55. Academic Press, San Diego (1992)

22. McKenna, T., Davis, J., Zornetzer, S.E. (eds.): *Single Neuron Computation*. Academic Press, San Diego (1992)
23. Arbib, M.A. (ed.): *The Handbook of Brain Theory and Neural Networks*. MIT Press, Boston (1998)
24. Koch, C., Poggio, T.: *Multiplying with Synapses*. In: McKenna, T., Davis, J., Zornetzer, S.F. (eds.) *Single Neuron Computation*, pp. 315–345. Academic Press, New York (1992)
25. Koch, C.: *Biophysics of Computation: Information Processing in Single Neurons*. Oxford University Press (1999)
26. Yu, A.J., Giese, M.A., Poggio, T.A.: *Biophysically plausible implementations of the maximum operation*. *Neural Computation* 14(12), 2857–2881 (2002)
27. Lampl, I., Ferster, D., Poggio, T., Riesenhuber, M.: *Intracellular measurements of spatial integration and the MAX operation in complex cells of the cat primary visual cortex*. *Journal of Neurophysiology* 92, 2704–2713 (2004)
28. Ritter, G.X., Urcid, G., Selfridge, R.: *Minimax dendrite computation*. In: Proc. ANNIE, ASME, St. Louis, Missouri, vol. 12, pp. 75–80 (2002)
29. Ritter, G.X., Iancu, L., Urcid, G.: *Morphological perceptrons with dendritic structure*. In: Proc. FUZZ-IEEE, St. Louis, Missouri, pp. 1296–1301 (2003)
30. Ritter, G.X., Iancu, L.: *Lattice algebra approach to neural networks and pattern classification*. In: IAPR Proc. 6th Open German-Russian Workshop on Pattern Recognition and Image Understanding, Katun Village, Altai Region, Russian Fed., pp. 18–21 (2003)
31. Ritter, G.X., Iancu, L., Urcid, G.: *Neurons, dendrites, and pattern classification*. In: Sanfeliu, A., Ruiz-Shulcloper, J. (eds.) *CIARP 2003*. LNCS, vol. 2905, pp. 1–16. Springer, Heidelberg (2003)
32. Ritter, G.X., Iancu, L.: *A morphological auto-associative memory based on dendritic computing*. In: Proc. IEEE, IJCNN, vol. 2, pp. 915–920 (2004)
33. Urcid, G., Ritter, G.X., Iancu, L.: *Single layer morphological perceptron solution to the N -bit parity problem*. In: Sanfeliu, A., Martínez Trinidad, J.F., Carrasco Ochoa, J.A. (eds.) *CIARP 2004*. LNCS, vol. 3287, pp. 171–178. Springer, Heidelberg (2004)
34. Urcid, G.: *Transformations of neural inputs in lattice dendrite computation*. In: Proc. SPIE: Math. Methods in Pattern and Image Analysis, San Diego, CA, vol. 5916, pp. 201–212 (2005)
35. Ritter, G.X., Urcid, G., Schmalz, M.S.: *Lattice associative memories that are robust in the presence of noise*. In: Proc. SPIE: Math. Methods in Pattern and Image Analysis, San Diego, CA, vol. 5916, pp. 255–260 (2005)
36. Myers, D.S.: *The synaptic morphological perceptron*. In: Proc. SPIE: Math. of Data/Image Pattern Recognition, Compr. and Encryp., San Diego, CA, vol. 6315, pp. 63150B:1–63150B:11 (2006)
37. Davidson, J.L., Ritter, G.X.: *A theory of morphological neural networks*. In: Proc. SPIE: Digital Optical Computing II, San Diego, CA, vol. 1215, pp. 378–388 (1990)
38. Davidson, J.L.: *Simulated annealing and morphological neural networks*. In: Proc. SPIE: Image Algebra and Morphological Image Proc. III, San Diego, CA, vol. 1769, pp. 119–127 (1992)
39. Davidson, J.L., Hummer, F.: *Morphology neural networks: An introduction with applications*. *IEEE Systems and Signal Processing* 12(2), 177–210 (1993)
40. Ritter, G.X., Sussner, P.: *An introduction to morphological neural networks*. In: Proc. 13th Int. Conf. on Pattern Recognition, Vienna, Austria, pp. 709–717 (1996)
41. Ritter, G.X., Sussner, P.: *Associative memories based on lattice algebra*. In: IEEE Int. Conf. Systems, Man, and Cybernetics, Orlando, FL, pp. 3570–3575 (1997)

42. Ritter, G.X., Sussner, P., Diaz de Leon, J.L.: Morphological associative memories. *IEEE Trans. Neural Networks* 9(2), 281–293 (1998)
43. Petridis, V., Kaburlasos, V.G.: Fuzzy lattice neural network (FLNN): A hybrid model for learning. *IEEE Trans. Neural Networks* 9(5), 877–890 (1998)
44. Ritter, G.X., Diaz de Leon, J.L., Sussner, P.: Morphological bidirectional associative memories. *Neural Networks* 12, 851–867 (1999)
45. Gabrys, B., Bargiela, A.: General fuzzy min-max neural network for clustering and classification. *IEEE Trans. on Neural Networks* 11(3), 769–783 (2000)
46. Sussner, P., Valle, M.E.: Gray-scale morphological associative memories. *IEEE Trans. on Neural Networks* 17(3), 559–570 (2006)
47. Kaburlasos, V.G.: Towards a Unified Modeling and Knowledge Representation Based on Lattice Theory, Computational Intelligence. *SCI*, vol. 27. Springer, Heidelberg (2006)
48. Kaburlasos, V.G., Ritter, G.X. (eds.): Computational Intelligence Based on Lattice Theory. *SCI*, vol. 67. Springer, Heidelberg (2007)
49. Sossa-A., J.H., Barron-F., R.: Extended $\alpha - \beta$ associative memories. *Revista Mexicana de Física* 53(1), 10–20 (2007)
50. Sussner, P., Laureano, E.E.: Morphological perceptrons with competitive learning: Lattice-theoretical framework and constructive learning algorithm. *Information Sciences* 181(10), 1929–1950 (2011)
51. Urcid, G., Ritter, G.X.: C -means clustering of lattice auto-associative memories for endmember approximation. In: *Adv. in Knowledge-Based and Intelligent Inf. and Eng. Systems*, San Sebastian, Spain, pp. 2140–2149. IOS Press (2012)
52. Valle, M.E., Sussner, P.: Quantale-based autoassociative memories with an application to the storage of color images. *Pattern Recognition Letters* 34(14), 1589–1601 (2013)
53. Iancu, L.: Lattice Algebra Approach to Neural Computation. Ph.D. Thesis, University of Florida, Gainesville, FL (2005)
54. Ritter, G.X., Gader, P.: Fixed Points of Lattice Transforms and Lattice Associative Memories. In: Hawkes, P. (ed.) *Advances in Imaging and Electron Physics*, vol. 144, pp. 165–242. Elsevier, San Diego (2006)
55. Ritter, G.X., Schmalz, M.S.: Learning in lattice neural networks that employ dendritic computing. In: *Proc. IEEE Inter. Conf. on Fuzzy Systems*, Vancouver, BC, Canada, pp. 209–215 (2006)
56. Ritter, G.X., Urcid, G.: Learning in lattice neural networks that employ dendritic computing. In: Kaburlasos, V.G., Ritter, G.X. (eds.) *Computational Intelligence based on Lattice Theory. SCI*, vol. 67, pp. 25–44. Springer, Heidelberg (2007)
57. Ritter, G.X., Urcid, G.: Lattice neural networks with spike trains. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) *HAIS 2010, Part II. LNCS*, vol. 6077, pp. 367–374. Springer, Heidelberg (2010)
58. Chyzhyk, D., Graña, M.: Optimal hyperbox shrinking in dendritic computing applied to Alzheimer’s disease detection in MRI. In: Corchado, E., Snášel, V., Sedano, J., Hassani, A.E., Calvo, J.L., Ślęzak, D. (eds.) *SOCO 2011. AISC*, vol. 87, pp. 543–550. Springer, Heidelberg (2011)
59. Ritter, G.X., Urcid, G.: Perfect recall from noisy input patterns with a dendritic lattice associative. In: *Proc. IEEE, IJCNN*, San Jose, CA, pp. 503–510 (2011)
60. Urcid, G., Ritter, G.X., Valdiviezo-N., J.C.: Grayscale image recall from imperfect inputs with a two layer dendritic lattice associative memory. In: *Proc. IEEE, 3rd Congr. on Nature and Biologically Inspired Computing*, Salamanca, Spain, pp. 268–273 (2011)

61. Chyzhyk, D.: Bootstrapped dendritic classifiers for Alzheimer's disease classification on MRI features. In: *Adv. in Knowledge-Based and Intelligent Inf. and Eng. Systems*, pp. 2251–2258. IOS Press (2012)
62. Chyzhyk, D., Graña, M., Savio, A., Maiora, J.: Hybrid dendritic computing with kernel-LICA applied to Alzheimer's disease detection in MRI. *Neurocomputing* 75, 72–77 (2012)
63. Ritter, G.X., Chyzhyk, D., Urcid, G., Graña, M.: A novel lattice associative memory based on dendritic computing. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) *HAIS 2012, Part II. LNCS*, vol. 7209, pp. 491–502. Springer, Heidelberg (2012)
64. Abrahamsson, T., Cathala, L., Matsui, K., Shigemoto, R., DiGregorio, D.A.: Thin dendrites of cerebellar interneurons confer sublinear synaptic integration and a gradient of short-term plasticity. *Neuron* 73, 1159–1172 (2012)
65. Caze, R.D., Humphries, M., Gutkin, B.: Passive dendrites enable single neurons to compute linearly non-separable functions. *PLoS Computational Biology* 9(2) (2013)
66. Urcid, G., Ritter, G.X., Valdiviezo-N., J.C.: Dendritic lattice associative memories for pattern classification. In: *Proc. IEEE, 4th Congr. on Nature and Biologically Inspired Computing, Mexico DF*, pp. 181–187 (2012)
67. Graña, M., Gonzalez-A., A.I.: Learning parsimonious dendritic classifiers. *Neurocomputing* 109, 3–8 (2013)
68. Sossa, H., Guevara, E.: Efficient training for dendrite morphological neural networks. *Neurocomputing* 131, 132–142 (2014)
69. Ritter, G.X., Urcid, G., Valdiviezo-N, J.C.: Two lattice metrics dendritic computing for pattern recognition. Accepted for publication in *FUZZ-IEEE* (2014)
70. Ritter, G.X., Iancu, L.: A lattice algebra approach to neural computation. In: *Handbook of Computational Geometry for Pattern Recognition, Computer Vision, Neurocomputing and Robotics*, pp. 97–129. Springer, Heidelberg (2005)
71. Vega, R., Guevara, E., Falcon, L.E., Sánchez-Ante, G., Sossa, H.: Blood vessel segmentation in retinal images using lattice neural networks. In: Castro, F., Gelbukh, A., González, M. (eds.) *MICAI 2013, Part I. LNCS*, vol. 8265, pp. 532–544. Springer, Heidelberg (2013)
72. Caro-C, D.E.: Méndez-V A.: Computing the concept lattice using dendritical neural networks. In: *Proc. 10th Int. Conf. on Concept Lattices and Their Applications*, pp. 141–152 (2013)
73. Wille, R.: Formal concept analysis as mathematical theory of concepts and concept hierarchies. In: Ganter, B., Stumme, G., Wille, R., et al. (eds.) *Formal Concept Analysis. LNCS (LNAI)*, vol. 3626, pp. 1–33. Springer, Heidelberg (2005)
74. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Fast computation of concept lattices using data mining techniques. In: *Proc. 7th Int. Workshop on Knowledge Representation and Databases*, pp. 129–139 (2000)