LETTER Communicated by Kenji Morita

A Low-Order Model of Biological Neural Networks

James Ting-Ho Lo
jameslo@umbc.edu
Department of Mathematics and Statistics, University of Maryland Baltimore County, Baltimore, MD 21250, U.S.A.

A biologically plausible low-order model (LOM) of biological neural networks is proposed. LOM is a recurrent hierarchical network of models of dendritic nodes and trees; spiking and nonspiking neurons; unsupervised, supervised covariance and accumulative learning mechanisms; feedback connections; and a scheme for maximal generalization. These component models are motivated and necessitated by making LOM learn and retrieve easily without differentiation, optimization, or iteration, and cluster, detect, and recognize multiple and hierarchical corrupted, distorted, and occluded temporal and spatial patterns.

Four models of dendritic nodes are given that are all described as a hyperbolic polynomial that acts like an exclusive-OR logic gate when the model dendritic nodes input two binary digits. A model dendritic encoder that is a network of model dendritic nodes encodes its inputs such that the resultant codes have an orthogonality property. Such codes are stored in synapses by unsupervised covariance learning, supervised covariance learning, or unsupervised accumulative learning, depending on the type of postsynaptic neuron. A masking matrix for a dendritic tree, whose upper part comprises model dendritic encoders, enables maximal generalization on corrupted, distorted, and occluded data. It is a mathematical organization and idealization of dendritic trees with overlapped and nested input vectors. A model nonspiking neuron transmits inhibitory graded signals to modulate its neighboring model spiking neurons. Model spiking neurons evaluate the subjective probability distribution (SPD) of the labels of the inputs to model dendritic encoders and generate spike trains with such SPDs as firing rates. Feedback connections from the same or higher layers with different numbers of unit-delay devices reflect different signal traveling times, enabling LOM to fully utilize temporally and spatially associated information.

Biological plausibility of the component models is discussed. Numerical examples are given to demonstrate how LOM operates in retrieving, generalizing, and unsupervised and supervised learning.

1 Introduction

A learning machine, called a temporal hierarchical probabilistic associative memory (THPAM), was recently reported (Lo, 2010). The goal to achieve in the construction of THPAM was to develop a learning machine that learns, with or without supervision, and retrieves easily without differentiation, optimization, or iteration; and recognizes corrupted, distorted, and occluded temporal and spatial information. In the process to achieve the goal, mathematical necessity took precedence over biological plausibility. This top-down approach focused first on minimum mathematical structures and operations that are required for an effective learning machine with the mentioned properties.

THPAM turned out to be a functional model of neural networks with many unique features that well-known models, such as the recurrent multilayer perceptron (Hecht-Nielsen, 1990; Principe, Euliano, & Lefebvre, 2000; Bishop, 2006; Haykin, 2009), associative memories (Kohonen, 1988b; Willshaw, Buneman, & Longet-Higgins, 1969; Nagano, 1972; Amari, 1989; Sutherland, 1992; Turner & Austin, 1997), spiking neural networks (Maass & Bishop, 1998; Gerstner & Kistler, 2002), and cortical circuit models (Martin, 2002; Granger, 2006; Grossberg, 2007; George & Hawkins, 2009) do not have.

These unique features indicated that THPAM might contain clues for understanding the operations and structures of biological neural networks. The components of THPAM were then examined from the biological point of view with the purpose of constructing a model of biological neural networks with biologically plausible component models. The components were identified with those of biological neural networks and reconstructed, if necessary, into biologically plausible models of the same.

This effort resulted in a low-order model (LOM) of biological neural networks. LOM is a recurrent hierarchical network of biologically plausible models of dendritic nodes and trees, synapses, spiking and nonspiking neurons, unsupervised and supervised learning mechanisms, a retrieving mechanism, a generalization scheme, and feedback connections with delays of different durations. All of these biologically plausible component models, except the generalization scheme and the feedback connections, are significantly different from their corresponding components in THPAM. More will be said about the differences.

Note that although a dendrite or axon is a part of a neuron and a dendrodendritic synapse is a part of a dendrite (thus, part of a neuron), they are treated, for simplicity, as if they were separate entities, and the word *neuron* refers essentially to the soma of a neuron in this letter.

A basic approximation made in the modeling effort reported here is that LOM is a discrete-time model and all the spike trains running through it are Bernoulli processes. The discrete-time approximation is frequently made in neuroscience. Mathematically, Bernoulli processes are discrete-time

approximations of Poisson processes, which are usually used to model spike trains. The discrete-time assumption seems similar to that made in the theory of discrete-time dynamical systems as exemplified by the standard time discretization of a differential equation into a difference equation. However, it is appropriate to point out that the discrete time for LOM and Bernoulli processes is perhaps more than simply a mathematical approximation. First, a spike (or action potential) is not allowed to start within the refractory period of another, violating a basic property of Poisson processes. Consecutive spikes in the brain cannot be arbitrarily close and are separated at least by the duration of a spike, including its refractory period, setting the minimum time length between two consecutive time points in a time line. Second, neuron or spike synchronization has been discovered in biological neural networks (von der Malsburg, 1981; Singer & Gray, 1995). Such synchronization may originate from or be driven by synchronous spikes from sensory neurons. Third, if a neuron and its synapses integrate multiple spikes in response to sensory stimuli being held constant (e.g., an image fixated by the retina for about one-third of a second) and the neuron generates multiple spikes in the process of each of such integrations, different timescales and their match-up can probably be reconciled for the discrete-time LOM and Bernoulli processes. Before more can be said, the discrete-time approximation is viewed upon as part of the low-order approximation by LOM.

Exogenous inputs to LOM and spike trains generated by spiking neurons propagate first through model dendritic encoders (see section 2.2), which are networks of model dendritic nodes and form the upper part of our model dendritic trees. Three different model dendritic nodes in LOM are each a hyperbolic polynomial with two variables, which acts approximately like an XOR (exclusive-OR) logic gate (Zador, Clairborne, & Brown, 1992; Fromherz and Gaede, 1993) with an accuracy depending on how close the two inputs to the node are to binary digits (see section 2.1). By combining such hyperbolic polynomials, which are commutative and associative binary operations, a model dendritic node may have more than two input variables. Prior results on structures and computation of dendritic trees can be found in Koch and Poggio (1982, 1992), Koch, Poggio, and Torre (1983), Rall and Sergev (1987), Shepherd and Brayton (1987), Mel (1992a, 1992b, 1993, 1994, 2008).

A model dendritic encoder, which is a mathematical composition of many hyperbolic polynomials, can be looked on as a function that encodes its input vector into an output vector with an orthogonal property (see section 2.3). The orthogonality property is proven in the appendix. Model dendritic encoders form groups. Output vectors from a group of model dendritic encoders are processed by trainable synapses at the output ends of these encoders for learning and retrieving. The strengths (or weights) of these model synapses are covariances between the encoder output vectors and the labels of the encoder input vectors. From the matrices

of these covariance, called expansion covariance matrices, subjective probability distributions of the labels can be extracted (see sections 3 and 4).

Model nonspiking neurons, called C-neurons, process outputs of trainable synapses and generate inhibitory graded signals (see section 7.1) transmitted to neighboring model spiking neurons. Model spiking neurons, called D-neurons, sum outputs of trainable synapses and use the inhibitory graded signal to compute subjective probability distributions (see section 4) and generate spike trains (see section 7.2), with firing rates being the subjective probability distributions. In computing a subjective probability distribution, the sum in a D-neuron is divided by the inhibitory graded signal received from a C-neuron. Discovery of division in neurons was reported in Tal and Schwartz (1997), Koch (1999), Gabbiani, Krapp, Koch, & Laurent (2002), Gabbiani et al. (2004), Mel (2008).

Three types of learning are performed in LOM: unsupervised covariance learning, supervised covariance learning, and unsupervised accumulation. The first is performed for trainable synapses with postsynaptic D-neurons whose outputs are teaching signals and the second for synapses receiving teaching signals from neurons that do not receive signals from these synapses. These two types of learning are variants of Sejnowski's covariance rule (see sections 3.1 and 3.2) and can be viewed as Hebbian-type learning (Sejnowski, 1977; Koch, 1999). The third type of learning is performed for accumulating deviations of the model dendritic encoders' output vectors from their averages over time in synapses with a postsynaptic C-neuron (see section 3.3). A forgetting factor and a normalizing constant are used to keep the entries in covariance matrices bounded.

Maximal generalization capability of LOM is achieved with masking matrices that each automatically find the largest subvector of the output vector of dendritic encoders that matches a subvector of an output vector having been stored in the expanded covariance matrix and set the rest of the components of the former output vector equal to 0. Masking matrices can be viewed as an idealization and organization of dendritic encoders with overlapped and nested input vectors (see section 5).

LOM can be looked on as a recurrent hierarchical network of processing units (PUs), each comprising a number of model dendritic encoders encoding subvectors of the vector input to the PU, a large number of model trainable synapses learning resultant codes jointly with their labels; a group of D-neurons retrieving information from model synapses, computing subjective probability distributions of these labels and generating spike trains; a C-neuron generating inhibitory graded signal to modulate these D-neurons (see section 6); and feedback connections with different numbers of unit-time delay devices receiving associated spatial and temporal information from the same or higher layers. At any time step, retrieving and learning are performed in the PU, and the output vector from the D-neurons in the PU (i.e., vector of spikes and nonspikes) is a

point estimate of the label of the input vector to the PU (i.e., the vector input to the dendritic encoders of the PU).

In unsupervised covariance learning, the output vector from the D-neurons is assigned as the label to be learned jointly with the input vector (see section 3.1). If the input vector has not been learned before, it is assigned a new label randomly selected by the D-neurons. If the input vector or a variation of it has been learned before, the output vector, a point estimate of the label or labels of the input vector based on the subjective probability distribution, is learned jointly with the input vector. Supervised covariance learning is performed when teaching signals (i.e., labels of input vector to the PU) from outside the PU are provided (see section 3.2). In the third type of learning, unsupervised accumulation learning, no label is needed.

LOM may have D-neurons with supervised learning synapses in PUs in the last few layers of LOM or in PUs on feedforward connections that branch out from the bulk of LOM's multilayer structure that perform unsupervised learning. Unsupervised learning can be performed all the time. What it actually does is clustering of spatial and temporal patterns or causes. When supervised learning occurs, the entire cluster is assigned with the same label. For example, a toddler puts apples of different colors, shapes, or orientations in the same cluster in unsupervised learning. Once he or she hears the word *apple* when he or she is looking at a certain apple, the entire cluster is assigned the label "apple" in sound by the toddler's neural networks. This provides an explanation of the "invariance" capability of the brain.

Biological plausibility of the foregoing component models of LOM is discussed in the section in which a model is described. Although these component models are worth looking at in their own right, a main contribution of this letter is the integration of these component models into a single mathematical model that is a low-order, yet structurally, functionally, and operationally faithful, description (to a degree) of biological neural networks with the following unique features that, regardless of biological plausibility, other well-known models such as the recurrent multilayer perceptron (Hecht-Nielsen, 1990; Principe et al., 2000; Bishop, 2006; Haykin, 2009), associative memories (Kohonen, 1988b; Willshaw et al., 1969; Nagano, 1972; Amari, 1989; Sutherland, 1992; Turner & Austin, 1997), spiking neural networks (Maass & Bishop, 1998; Gerstner & Kistler, 2002), and cortical circuit models (Martin, 2002; Granger, 2006; Grossberg, 2007; George & Hawkins, 2009) do not have:

1. Model dendritic nodes that are low-order polynomials, which act approximately like an XOR logic gate or a composition of these gates
2. Model dendritic encoders, which are networks of mentioned model dendritic nodes and whose outputs form approximately orthogonal vectors

3. Model spiking neurons containing a pseudo-random number generator
4. Firing rates of the spike trains being the subjective probability distributions of labels
5. Three learning rules: unsupervised covariance learning, supervised covariance learning, and unsupervised accumulation
6. A recurrent multilayer network learning by a Hebbian-type unsupervised learning rule
7. Decovariance retrieving for retrieving a subjective probability distribution
8. Masking matrices for maximizing generalization
9. Feedback connections with different delay durations (e.g., different numbers of unit-time delay devices) for fully utilizing temporally and spatially associated information

Note that THPAM, regardless of biological plausibility, does not have features 1, 2, 5, and 7. Major differences between LOM and THPAM are listed in the Conclusion.

To the best of my knowledge, LOM is the only single biologically plausible model of biological neural networks that provides logically coherent answers to the following questions:

1. What is the information carried by the spike rate in a spike train?
2. How is knowledge encoded?
3. In what form is the encoded knowledge stored in the synapses?
4. What does the dendritic node do? How does the dendritic node do it?
5. How are dendritic nodes organized into dendritic trees? Why is there compartmentalization in dendritic trees?
6. How do the dendritic trees contribute to the neural computation?
7. How is unsupervised learning performed by a multilayer neural network?
8. How is a piece of knowledge stored in the synapses retrieved and converted into spike trains?
9. How do neural networks generalize on corrupted, distorted or occluded patterns?

The terms *biologically plausible*, *functional model*, and *low-order model* used in this letter are clarified as follows. Following dictionary.com, the definition of the word *plausible* used in this letter means "having an appearance of truth or reason; seemingly worthy of approval or acceptance." *Biologically plausible* means "plausible from the biological viewpoint." For example, the NXOR logic gate, orthogonal encoder, and unsupervised and supervised learning mechanisms in THPAM are not biologically plausible. Before being confirmed with experimental results or biological implementations, a biologically plausible model is only a hypothesis. Conceiving hypotheses

is a useful step in mathematics and physics. It is equally useful in neuroscience.

The term *functional model* originated from the theory of system control (Kalman, Falb, & Arbib, 1969). Constructing a faithful model of a complex dynamical system (e.g., national and international economies, jet engines, chemical processing plants) by physics is often difficult. For the purpose of controlling a system, it was found that an "identification" of the input-output relation of the system is sometime sufficient regardless of the actual inner workings of the system. Depending on the proportions of system-theoretic identification and physics-based modeling involved, there are "gray boxes" with different gray levels. A functional model is such a gray box. A main purpose of building a functional model is utility. THPAM in Lo (2010) is a gray box intended to be a learning machine.

The term *low-order model* is usually used to deal with the mathematical intractability of a full model in mathematics, physics, and engineering. A well-known example of a low-order model is the ideal fluid—fluid that is nonviscous and incompressible (or obtained by setting viscosity and compressibility coefficients equal to 0 in the Navier-Stokes equations; Landau & Lifshitz, 1987). The ideal fluid is physically plausible only in regions far away from fluid boundaries and when the fluid velocity is subsonic. Nevertheless, it plays an important role in fluid mechanics even today. Studying effects of viscosity in the fluid led to the boundary-layer theory (Schlichting & Gersten, 2000), and considering compressibility of fluid in studying transonic and supersonic fluids yielded aerodynamics and gas dynamics, where higher-order models were studied. The higher the order of a physically or biologically plausible model is, the closer it is to a physical or biological implementation.

The biologically plausible low-order model, LOM, reported in this letter is a hypothesis to be biologically verified. Undoubtedly many higher-order phenomena in biological neural networks can be found missing in LOM. Nevertheless, LOM has many important features unique among existing models and provides logically coherent answers to many important long-standing questions jointly.

2 Dendritic Encoders

Dendrites use more than 60% of the energy consumed by the brain (Wong, 1989), occupy more than 99% of the surface of some neurons (Fox & Barnard, 1957) and are the largest component of neural tissue in volume Sirevaag & Greenough (1987). It was discovered in the 1980s and 1990s that dendrites are capable of performing information processing tasks (Koch & Poggio, 1982, 1992; Koch et al., 1983; Rall & Sergev, 1987; Shepherd & Brayton, 1987; Mel, 1992a, 1992b, 1993, 1994; Poirazi, Brannon, & Mel, 2003; Mel, 2008). Spratling and Johnson (2001, 2002), found dendritic inhibition to enhance neural coding properties

and unsupervised learning in neural networks. Computation by dendrites was also discussed in Goldman, Levine, Major, Tank, & Seung (2003), Morita, Okada, & Aihara (2007), Rhodes (2008), and Morita (2008, 2009).

However, dendritic trees are missing in well-known artificial neural networks (Rieke, Warland, de Ruyter van Steveninck, & Bialek, 1999; O'Reilly & Munakata, 2000; Dayan & Abbott, 2001; Hecht-Nielsen & McKenna, 2003; Hawkins, 2004; Hecht-Nielsen, 2007; Principe et al., 2000; Bishop, 2006; Haykin, 2009) , associative memories (Kohonen, 1988a; Hinton & Anderson, 1989; Hassoun, 1993) and models of cortical circuits (Martin, 2002; Granger, 2006; Grossberg, 2007; George & Hawkins, 2009), overlooking a large percentage of the neural circuit. However, Spratling and Johnson (2001, 2002) found that dendritic inhibition enhances neural coding properties and unsupervised learning. More discussion on computation by dendrites can be found in Goldman et al. (2003), Morita et al. (2007), Rhodes (2008), Morita (2008, 2009).

In the low-order model (LOM) proposed in this letter, an upper part of the model dendritic tree, called a dendritic encoder, is a network of model dendritic nodes, each of which is a low-order polynomial with two variables that acts like an XOR (exclusive-OR) logic gate when the two variables are binary digits. XOR gates are found in biological dendritic trees by Zador et al. (1992). Five types of single arborized neuron that perform XOR were reported by Fromherz and Gaede (1993).

In this section, model dendritic nodes and a model dendritic encoder are described. The model dendritic encoder is a function that encodes the vector input to the encoder into the vector output from the encoder. The output vectors have an orthogonality property proven in the appendix. The orthogonality property can be viewed as an extension of a theorem in the coding theory by Slepian (1956).

2.1 Binary Operation of a Dendritic Node. The use of logic gates and low-order polynomials in describing biological dendritic trees was discussed in Mel (1994). The low-order polynomial metaphor may be viewed as a smooth, analog version of the logical dendrites hypothesis.

Low-order models of three types of dendritic node and one type of axonal node are depicted and specified in Figure 1. The mini-networks in panels a to c each have a dendritic branch point and hence are called model dendritic nodes. The one in panel d has an axonal branch point and is called a model axonal node. The model dendritic node in panel a receives inputs v and u through dendrites and models "interior" dendritic nodes that are not in direct contact with axons. The synapses in the model are inhibitory dendro-dendritic synapses (Shepherd, 2004; Shepherd & Grillner, 2010). Dendritic preintegration inhibition was reported in Spratling and Johnson (2001, 2002). Dendritic spikes were observed in pyramidal cells of hippocampus (Wong, Prince, & Basbaum, 1979),

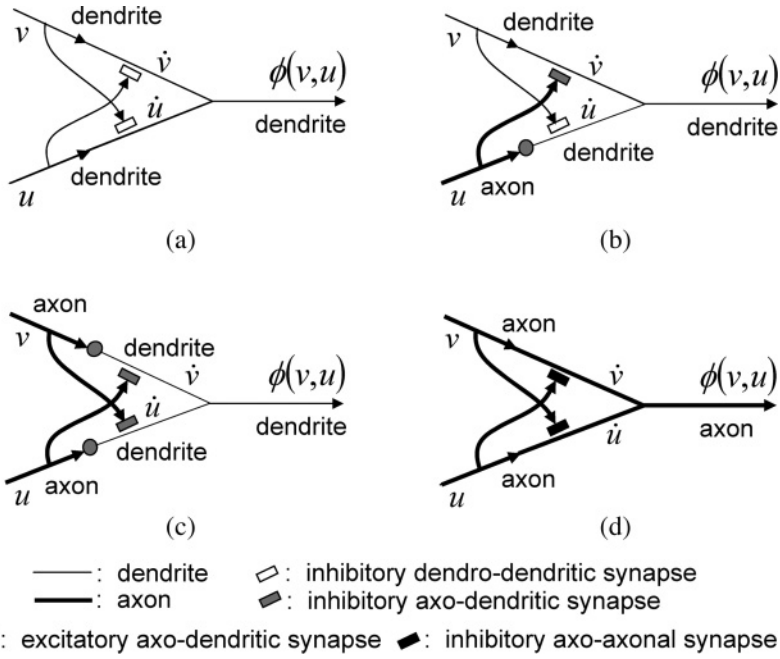

Figure 1: Four types of mini-network, whose output $\phi(v, u)$ is a hyperbolic polynomial $-2vu + v + u$ of its input variables, v and u, are shown. $\dot{u} = u(1 - v)$ is u inhibited by v. $\dot{v} = v(1 - u)$ is v inhibited by u. If v and u are binary digits, $\phi(v, u)$ acts like the logic gate XOR. (a) v and u are received by dendrites and inhibited by each other through dendro-dendritic synapses. (b) v and u are received, respectively, by a dendrite and an axon, and are inhibited by each other through an axo-dendritic synapse and a dendro-dendritic synapse, respectively. (c) v and u are received by axons and are inhibited by each other through axo-dendritic synapses. (d) v and u are received by axons and are inhibited by each other through axo-axonal synapses. The networks in panels a–c are called dendritic nodes. The network in panel d is called an axonal node.

Purkinje cells of the cerebellum (Llinas & Sugimori, 1980), and neocortical pyramidal cells (Antic, Zhou, Moore, Short, & Ikonomu, 2010). The mini-networks in panels b and c receive at least one of the inputs directly from an axon. The dendrite that said axon synapses on is therefore on the boundary of the dendritic tree in which the dendrites in the mini-network belong.

In addition to dendritic trees performing information processing, it is known that there are extensive axonal arbors (Wittner, Henze, & Zaborszky, 2007; Matsuda et al., 2009), which also perform information processing

(Debanne, 2004). The model axonal node in Figure 1d does not involve a dendrite and is "interior" in an axonal arbor. Axo-axonal synapses are discussed in Milokhin and Reshetnikov (1968) and Pannese (1994). It is unclear if the common belief that axo-axonal synapses, excitatory or inhibitory, are rare in the biological neural networks is correct or a misunderstanding due to lack of knowledge. If the belief is correct, axonal nodes modeled by the model axonal node must be rare in the brain.

The input variables, v and u, are usually spikes in spike trains modeled as Bernoulli processes. After mutual inhibition, v and u are transformed into $\dot{v} = v(1-u)$ and $\dot{u} = u(1-v)$, respectively. The output is the sum of $\dot{v}$ and $\dot{u}$,

$$\phi(v,u) = -2vu + v + u, \tag{2.1}$$

which is a hyperbolic polynomial depicted in Figure 2. If v and u are binary digits, $\phi(v,u)$ is the XOR function. If not, $\phi(v,u)$ approximates the XOR function nicely. As shown in Figure 2, the closer v and u are to binary values, the more ϕ acts like XOR. The further they are from binary values, the less ϕ acts like XOR. For example, $\phi(.9,.9) = .18$, $\phi(.9,.1) = .82$, $\phi(.9,.75) = .3$, and $\phi(.75,.1) = .7$. ϕ acts more like XOR at (.9, .9) and (.9, .1) than at (.9, .75) and (.75, .1). Note that there are other polynomials (e.g., the elliptic polynomial $\eta(v,u) = 2v^2 + 2u^2 - 2vu - v - u$) that act like the XOR function at binary inputs, but ϕ involves the least number of arithmetic operations and approximates XOR in the most reasonable manner as shown in Figure 2. $\phi(v,u)$ is henceforth called the XOR polynomial.

2.2 Composition of Operations of Dendritic Nodes. The algebraic binary operation $\phi(v,u) = -2vu + v + u$ is commutative and also associative:

$$\begin{aligned}\phi(w,\phi(v,u)) &= \phi(\phi(w,v),u)\\ &= (-2)^2 wvu - 2(wv + wu + vu) + w + v + u.\end{aligned}$$

Hence, we can define a symmetric function ϕ_k by applying the binary operation repeatedly as follows:

$$\phi_k(v_1, v_2, \ldots, v_k) = \phi(\ldots\phi(\phi(v_1, v_2), v_3), \ldots, v_k),$$

where $\phi_1(v_i) = v_i$ and $\phi_2(v_i, v_j) = \phi(v_i, v_j)$.

It follows that

$$\phi_k(v_1, v_2, \ldots, v_k) = \phi(\phi_i(v_1, v_2, \ldots, v_i), \phi_{k-i}(v_{i+1}, v_{i+2}, \ldots, v_k)).$$

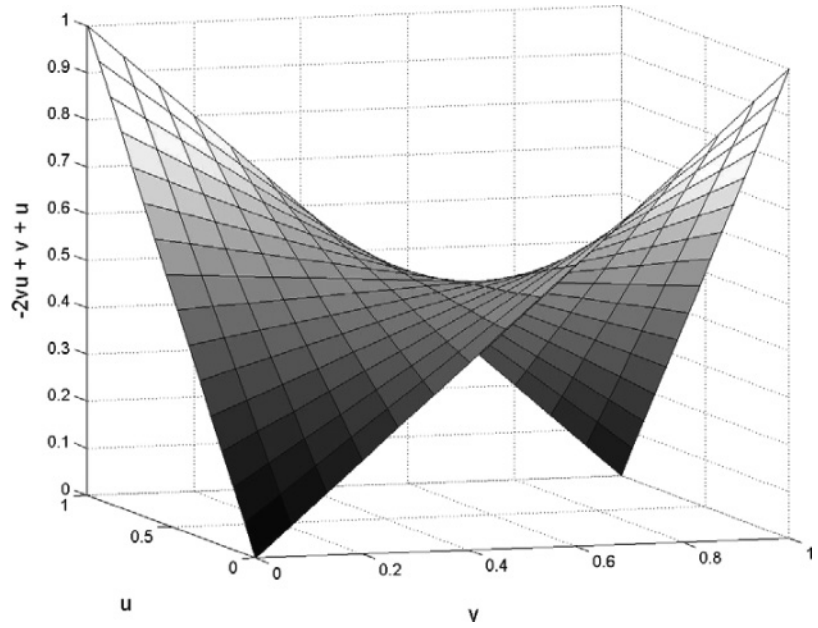

Figure 2: The three-dimensional graph shows the output $\phi(v, u)$ of a model dendritic node over the input domain, the unit square $[0, 1]^2$. The saddle shape of $\phi(v, u)$ shows that the model dendritic node is robust. If strengths of spikes change (e.g., weakening) in traveling through dendrites or if the spikes are corrupted by biological noise in the dendrites, the outputs of the model dendritic node suffer only from a graceful degradation. The hyperbolic polynomial $-2vu + v + u$ is an ideal approximation of the XOR logic gate and henceforth called the XOR polynomial.

Therefore, there are many different ways to obtain $\phi_k(v_1, v_2, \ldots, v_k)$. For example, $\phi_4(v_1, v_2, v_3, v_4)$ can be obtained by $\phi(\phi(v_1, v_2), \phi(v_3, v_4))$ or by $\phi(\phi_3(v_3, v_2, v_4), v_1)$.

An upper part of an example model dendritic tree proposed in this letter is depicted in Figure 3. It is a function whose outputs are determined by its inputs. Such an upper part of a model dendritic tree is called a model dendritic encoder. If the model dendritic encoder has m inputs forming an input set $\{v_1, v_2, \ldots, v_m\}$, then the input set has 2^m subsets. On each of these subsets, say, $\{v_{k_1}, v_{k_2}, \ldots, v_{k_i}\}$, an output of the dendritic encoder is defined to be $\phi_i(v_{k_1}, v_{k_2}, \ldots, v_{k_i})$. For example, if the input set is $\{v_1, v_2, v_3\}$, then the subsets are Φ, $\{v_1\}$, $\{v_2\}$, $\{v_2, v_1\}$, $\{v_3\}$, $\{v_3, v_1\}$, $\{v_3, v_2\}$, $\{v_3, v_2, v_1\}$, where Φ is the empty set. The outputs of the dendritic encoder are $\phi_0(\Phi)$, $\phi_1(v_1)$, $\phi_1(v_2)$, $\phi_2(v_2, v_1)$, $\phi_1(v_3)$, $\phi_2(v_3, v_1)$, $\phi_2(v_3, v_2)$, $\phi_3(v_3, v_2, v_1)$, where $\phi_0(\Phi)$ is defined to be 0.

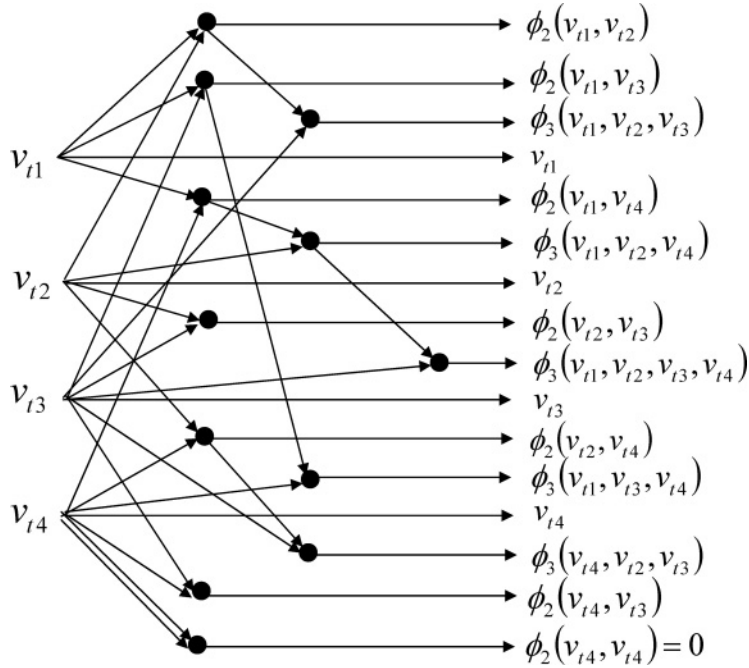

Figure 3: An upper part of a dendritic tree with 4 inputs and 16 branches that performs dendritic encoding. This part is called a dendritic encoder. The solid dots represent the dendritic nodes shown in Figure 1. Because of the commutativity and associativity of the XOR polynomial, there are many possible branching structures with the same inputs and branch outputs listed on the right side of the graph. Some branches are axons.

Similarly, if the input set is $\{v_1, v_2, v_3, v_4\}$, the model dendrite encoder has 16 outputs $\phi_i(v_{k_1}, \ldots, v_{k_i})$, where $\{v_{k_1}, \ldots, v_{k_i}\}$ are subsets of $\{v_1, v_2, v_3, v_4\}$. Figure 3 shows such a model dendritic encoder with 4 inputs and 16 outputs, where the four inputs v_{ti}, $i = 1, 2, 3, 4$, at time t are each close to a binary digit. $\phi_i(v_{k_1}, \ldots, v_{k_i})$ can be evaluated by binary ϕ_2 or other operations ϕ_k in more than one way if $i > 2$. Therefore, the structure of a model dendritic encoder for more than two inputs is not unique.

Note that $\phi(v_i, v_i) = 0$ and $\phi(0, v_i) = v_i$. It follows that for $v_{k_1}, v_{k_2}, \ldots, v_{k_j}$, and v_{k_1},

$$\begin{aligned}\phi_{j+1}\left(v_{k_1}, v_{k_1}, v_{k_2}, \ldots, v_{k_j}\right) &= \phi\left(\phi_2\left(v_{k_1}, v_{k_1}\right), \phi_{j-1}\left(v_{k_2}, \ldots, v_{k_j}\right)\right) \\ &= \phi\left(0, \phi_{j-1}\left(v_{k_2}, \ldots, v_{k_j}\right)\right) \\ &= \phi_{j-1}\left(v_{k_2}, \ldots, v_{k_j}\right).\end{aligned}$$

Hence, a function ϕ_j with repeated variables can be identified with a function ϕ_{j-2i} with different variables for some $i > 0$. Using $\{v_1, v_2, \ldots, v_m\}$ as the input set and $\phi(v, u)$ to compose functions, we can obtain only 2^m different functions for input variables with binary values.

This explains mathematically that a biological dendritic encoder for a given set of m inputs may grow by forming nodes each with two branches meeting at a time, in a "random" manner. At most 2^m different outputs ϕ_k (including ϕ_0) are generated by the dendritic encoder in response to the m inputs.

Notice that 2^m is an exponential function of m. This implies that a biological dendritic encoder has a reasonably small number m of inputs and a large number of inputs must be processed by multiple dendritic encoders, each inputting a small subset of these inputs. These dendritic encoders are believed to be called compartments or subunits in the literature (Koch & Poggio, 1982, 1992; Koch et al., 1983; Rall & Sergev, 1987; Shepherd & Brayton, 1987; Mel, 1992a, 1992b, 1993, 1994, 2008; Poirazi et al., 2003).

Replacing the dendrites and dendritic nodes in Figure 3 with axons and axonal nodes, respectively, we obtain a model axonal encoder. Operations of model axonal nodes can be composed in the same way operations of model dendritic nodes are composed as described above.

An orthogonality property of a model dendritic encoder's outputs is discussed in the next section and proven in the appendix. The discussion and proof are equally valid for a model axonal encoder. In fact, a model encoder with both model dendritic nodes and model axonal nodes may be employed so that they have all the 2^m outputs ϕ_k that jointly have the orthogonality property. The rest of the letter is still valid if model dendritic encoders are replaced with model axonal encoders or mixtures of them. This indicates there is much flexibility in the growth of biological dendritic and axonal encoders, which might have contributed to the mystery about their morphologies.

2.3 An Orthogonality Property of a Dendritic Encoder's Outputs. To describe an orthogonality property of the outputs of a model dendritic encoder with input variables $\{v_1, v_2, \ldots, v_m\}$, we organize its 2^m outputs into a vector as follows. Let u denote a scalar and $v = [v_1 \quad v_2 \quad \ldots \quad v_k]$ a k-dimensional vector. Define a k-dimensional vector $\phi(u, v)$ of polynomials by

$$\phi(u, v) = \Big[\phi(u, v_1) \quad \phi(u, v_2) \quad \ldots \quad \phi(u, v_k)\Big].$$

The 2^m different functions that can be defined by compositions of the binary operation $\phi(v, u)$ on the input set $\{v_1, v_2, \ldots, v_m\}$ are generated and organized into a 2^m-dimensional column vector $\check{v}$ by recursively generating

row vectors $\breve{v}(1, \ldots, k)$, for $k = 1, 2, \ldots, m$, as follows:

$$\breve{v}(1) = \begin{bmatrix} 0 & v_1 \end{bmatrix}, \tag{2.2}$$

$$\breve{v}(1,2) = \begin{bmatrix} \breve{v}(1) & \phi(v_2, \breve{v}(1)) \end{bmatrix}$$

$$= \begin{bmatrix} 0 & v_1 & v_2 & -2v_2v_1 + v_2 + v_1 \end{bmatrix}, \tag{2.3}$$

$$\breve{v}(1, \ldots, k+1) = \begin{bmatrix} \breve{v}(1, \ldots, k) & \phi(v_{k+1}, \breve{v}(1, \ldots, k)) \end{bmatrix} \tag{2.4}$$

$$\breve{v} = \breve{v}'(1, \ldots, m). \tag{2.5}$$

Denoting the kth component of $\breve{v}$ by $\breve{v}_k$, the vector $\breve{v} = \begin{bmatrix} \breve{v}_1 & \breve{v}_2 & \ldots & \breve{v}_{2^m} \end{bmatrix}'$ is called the dendritic expansion of v. Setting the first component of $\breve{v}(1)$ equal to 0 above (instead of 1) yields two properties. First, because $\phi(v, v) = 0$ and $\phi(v, 0) = v$, two equal binary signals meeting at a dendritic node produce the first component of $\breve{v}$, and this first component will not change other components through dendritic nodes down the stream. Second, this makes $\breve{0} = 0$. Here, 0 are the zero vectors.

It is proven in the appendix that given two m-dimensional binary vectors, v and u, their dendritic expansions, $\breve{v}$ and $\breve{u}$, satisfy

$$\left(\breve{v} - \frac{1}{2}\mathbf{I}\right)' \left(\breve{u} - \frac{1}{2}\mathbf{I}\right) = 0, \text{ if } v \neq u \tag{2.6}$$

$$= 2^{m-2}, \text{ if } v = u, \tag{2.7}$$

where $\mathbf{I} = [1 \ 1 \ \ldots \ 1]'$, which we note is not the identity matrix I.

Example 1a. If $v = [1 \ 0 \ 1 \ 0]'$ and $u = [1 \ 0 \ 1 \ 1]'$, then $\breve{v}(1) = [0 \ 1]$, $\breve{v}(1,2) = [0 \ 1 \ \phi(0,0) \ \phi(0,1)] = [0 \ 1 \ 0 \ 1]$, $\breve{v}(1,2,3) = [0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0]$, and

$$\breve{v} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}',$$

$$\breve{u} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}'.$$

It follows that $\breve{v} - \frac{1}{2}\mathbf{I}$ and $\breve{u} - \frac{1}{2}\mathbf{I}$ are, respectively,

$$\begin{bmatrix} -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \end{bmatrix}',$$

$$\begin{bmatrix} -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix}'.$$

Hence, $(\check{v} - \frac{1}{2}\mathbf{I})'(\check{v} - \frac{1}{2}\mathbf{I}) = 2^{4-2}$ and $(\check{v} - \frac{1}{2}\mathbf{I})'(\check{u} - \frac{1}{2}\mathbf{I}) = 0$, as predicted by equations 2.7 and 2.6.

3 Learning by Synapses on a Dendritic Tree

Three learning rules—the unsupervised covariance rule, supervised covariance rule, and unsupervised accumulation rule—are described in this section. The first two are essentially Sejnowski's covariance rule Sejnowski (1977), whose biological plausibility is also discussed on in Koch (1999). However, the unsupervised covariance rule and supervised covariance rule proposed here do not build up the covariance between the outputs of the presynaptic and postsynaptic neurons as Sejnowski's covariance rule does. The unsupervised covariance rule builds up, in synapses, the covariance between the outputs of the presynaptic dendritic encoder and the postsynaptic neurons. The supervised covariance rule builds up the covariance between the outputs of the presynaptic dendritic encoder and the outputs of neurons that do not receive signals from the synapses on the dendritic encoder, but act as teachers to the synapses. Like Sejnowski's covariance learning rule, the unsupervised and supervised covariance learning rules here, especially the former, can be looked as variants of what is commonly known as Hebb's rule. The unsupervised accumulation rule simply accumulates deviations of the dendritic encoder outputs from their averages over a certain time window.

3.1 Unsupervised Covariance Rule. There are two types of model neuron in LOM: D-neurons and C-neurons. A D-neuron is a model spiking neuron generating a spike train, and a C-neuron is a model nonspiking neuron outputting inhibitory graded signals that are transmitted to its neighboring D-neurons. Computations performed in these two types of neuron are described in section 7.

Each of the 2^m outputs, $\check{v}_{t1}, \check{v}_{t2}, \ldots, \check{v}_{t2^m}$, from a dendritic encoder at time (or numbering) t passes through a synapse to reach each of a number of, say, R, postsynaptic D-neurons and a postsynaptic C-neuron. Figure 4 shows the outputs of the dendritic encoder going through synapses represented by $\otimes$ to reach a D-neuron, say D-neuron i, whose output at time t is u_{ti}.

The unsupervised covariance rule that updates the strength D_{ij} of the synapse receiving $\check{v}_{tj}$ and feeding D-neuron i whose output is u_{ti} follows:

$$D_{ij} \leftarrow \lambda D_{ij} + \Lambda \left(u_{ti} - \langle u_{ti} \rangle\right) \left(\check{v}_{tj} - \left\langle \check{v}_{tj} \right\rangle\right), \tag{3.1}$$

where Λ is a proportional constant, λ is a forgetting factor that is a positive number less than 1, and $\langle \check{v}_{tj} \rangle$ and $\langle u_{ti} \rangle$ denote, respectively, the average activities of the presynaptic dendritic node j and postsynaptic D-neuron i over some suitable time intervals.

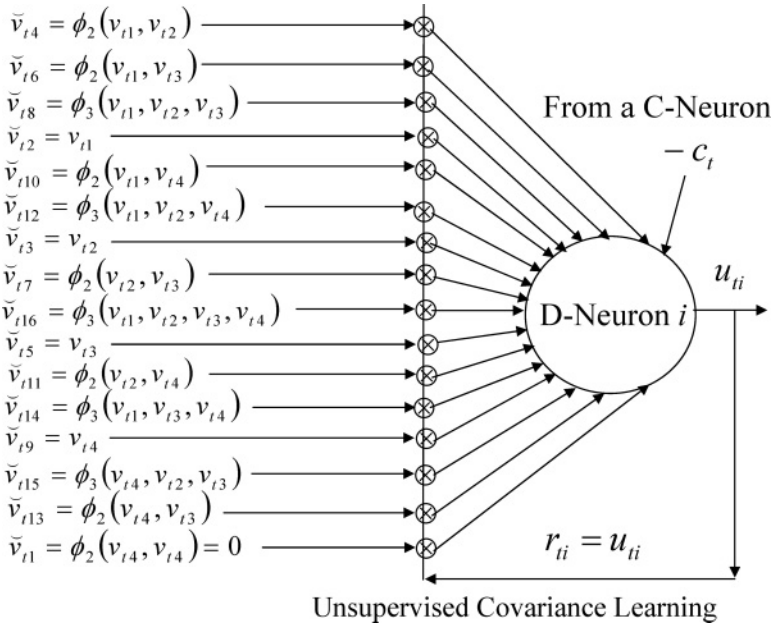

Figure 4: A D-neuron, which is a model of spiking neurons, and an unsupervised convariance learning mechanism are shown. D-neuron i receives an inhibitory graded signal $-c_t$ from a C-neuron, which is a model of nonspiking neurons and the 16 outputs from the model dendritic encoder depicted in Figure 3. The synapses $\otimes$ are dendro-dendritic or axo-dendritic, depending on where the signals come from, and are capable of learning (i.e., trainable). The postsynaptic dendrites from the synapses $\otimes$ to D-neuron i may merge in many ways. Dendrites from synapses $\otimes$ to D-neuron i form the lower parts of a dendritic tree. The upper (the dendritic encoder in Figure 3) and the lower parts morphologically resemble a biological dendritic tree (more convincingly shown in Figure 9). Note that the binary output u_{ti} of the postsynaptic D-neuron is used as the teaching signal (i.e., desired output) for all the synapses $\otimes$. The unsupervised covariance learning mechanism is essentially Sejnowski's covariance learning rule, except that the covariance matrix here is not that between pre- and postsynaptic outputs.

The outputs u_{ti}, $i = 1, \ldots, R$, of the R D-neurons can be assembled into a vector, $u_t = [\, u_{t1}\ u_{t2} \ldots u_{tR}\,]'$, and the strengths D_{ij} into a $R \times 2^m$ matrix D whose $i \times j$th entry is D_{ij}. This matrix D is called an expansion covariance matrix. Using these notations, the covariance rule can be expressed as follows:

$$D \leftarrow \lambda D + \Lambda\,(u_t - \langle u_t \rangle)\,(\breve{v}_t - \langle \breve{v}_t \rangle)'. \tag{3.2}$$

If the vector pairs, (v_s, u_s), $s = 1, \ldots, t$, have been learned by the $2^m R$ synapses, their expansion covariance matrix D is

$$D = \Lambda \sum_{s=1}^{t} \lambda^{t-s} \left(u_s - \langle u_s \rangle\right) \left(\breve{v}_s - \langle \breve{v}_s \rangle\right)'. \tag{3.3}$$

In addition to the biological plausibility of Sejnowski's covariance rule (Sejnowski, 1977; Koch, 1999), which is shared by the unsupervised covariance rule 3.1, this rule 3.1 makes LOM more fault tolerant and efficient than THPAM (Lo, 2010):

1. If the D-neuron outputting u_{ti} or the dendritic node outputting $\breve{v}_{tj}$ is out of order, causing u_{ti} or $\breve{v}_{tj} = 0$ or 1 or any constant for too long, then $u_{ti} - \langle u_{ti} \rangle = 0$ or $v_{ti} - \langle v_{ti} \rangle = 0$, whence $D_{ij} \leftarrow \lambda D_{ij}$ and D_{ij} shrinks to 0, eliminating the effect of the faulty D-neuron or dendritic node.
2. If $\breve{v}_{tj}$ takes on 1 (or 0) significantly more often than 0 (or 1), then $\langle \breve{v}_{tj} \rangle$ is closer to 1 (or 0), $\breve{v}_{tj} - \langle \breve{v}_{tj} \rangle$ is smaller for $\breve{v}_{tj} = 1$ (or 0) than for $\breve{v}_{tj} = 0$ (or 1), and D learns $\breve{v}_{tj}$ with less intensity. The same happens if u_{ti} takes on 1 (or 0) significantly more often than 0 (or 1). This automatically balances out the number of additions (to store 1's) to and subtractions (to store 0's) from D to avoid memory saturation at a synapse.
3. If $\breve{v}_{tj}$ is replaced with an inhibitory $\breve{v}_{tj}$ taking on -1 or 0, then $\breve{v}_{tj} - \langle \breve{v}_{tj} \rangle = -(-\breve{v}_{tj} - \langle -\breve{v}_{tj} \rangle)$, where $-\breve{v}_{tj}$ is excitatory, and the orthogonality property among $\breve{v}_t$ remains valid. This will be discussed more in section 4.

These advantages of the unsupervised covariance rule are valid also for the supervised covariance rule and the unsupervised accumulation rule to be described below. They are actually fundamental advantages of Sejnowski's covariance rule.

How a vocabulary is built by the unsupervised covariance rule is discussed at the end of section 7.

3.2 Supervised Covariance Rule. In supervised learning, an output from a D-neuron that does not receive a signal from the synapse is used as the teaching signal for the synapse. For example, D-neurons in the auditory cortex may provide teaching signals to the synapses in the visual cortex, and vice versa. Figure 5 shows a D-neuron from elsewhere whose output spike train w_{ti}, $t = 1, 2, \ldots$, is used to update the synapses $\otimes$ of neuron i. The D-neuron from elsewhere is called a teaching D-neuron. Note that it is not a postsynaptic neuron to the synapses $\otimes$ in Figure 5.

Assume that there are R teaching D-neurons outputting R spike trains w_{ti}, $t = 1, 2, \ldots, i = 1, \ldots, R$. A supervised covariance rule that updates the strength D_{ij} of the synapse receiving $\breve{v}_{tj}$ and w_{ti} is the following:

$$D_{ij} \leftarrow \lambda D_{ij} + \Lambda \left(w_{ti} - \langle w_{ti} \rangle\right) \left(\breve{v}_{tj} - \left\langle \breve{v}_{tj} \right\rangle\right), \tag{3.4}$$

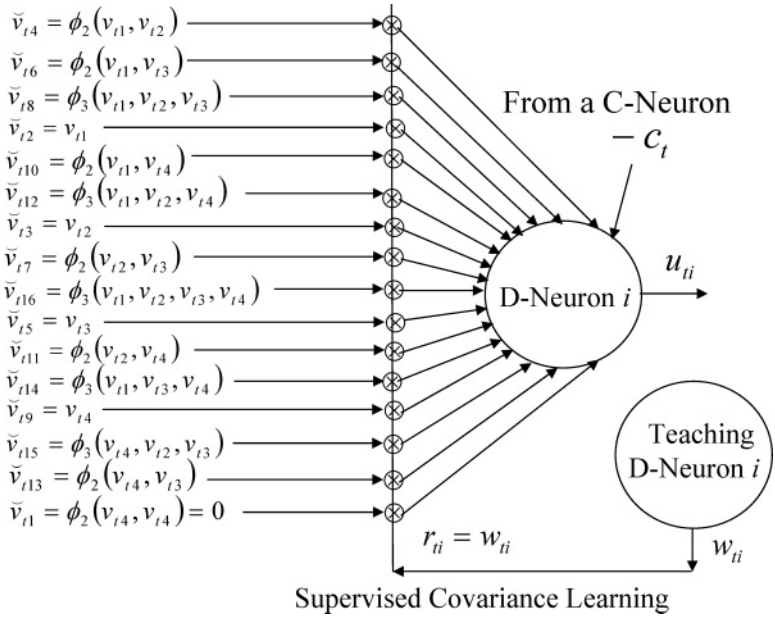

Figure 5: The graph is similar to that in Figure 4 except that the learning mechanism shown here is a supervised convariance learning mechanism. Note that the binary output u_{ti} of the postsynaptic D-neuron is not used as the teaching signal. The teaching signal w_{ti} comes from another D - neuron, called a teaching D-neuron. This supervised covariance learning mechanism is a variant of Sejnowski's covariance learning rule. The covariances learned here do not involve output of either pre- or postsynaptic neurons.

for $j = 1, \ldots, 2^m$ and $i = 1, \ldots, R$, where Λ and λ are a proportion constant and a forgetting factor, and $\langle \breve{v}_{tj} \rangle$ and $\langle w_{ti} \rangle$ denote, respectively, the average activities of the presynaptic dendritic node j and teaching D-neuron i over some suitable time intervals.

The outputs w_{ti} of the R teaching D-neurons can be assembled into a vector, $w_t = [\, w_{t1}\ w_{t2}\ \ldots\ w_{tR}\,]'$, and the synaptic strengths D_{ij} into a $R \times 2^m$ matrix D whose $i \times j$th entry is D_{ij}. This matrix D is again called an expansion covariance matrix. Using these notations, the covariance rule can be expressed as

$$D \leftarrow \lambda D + \Lambda\,(w_t - \langle w_t \rangle)\,(\breve{v}_t - \langle \breve{v}_t \rangle)'\,. \tag{3.5}$$

If the pairs, (v_s, w_s), $s = 1, \ldots, t$, have been learned by the $R(2^m)$ synapses, their expansion correlation matrix D is

$$D = \Lambda \sum_{s=1}^{t} \lambda^{t-s}\,(w_s - \langle w_s \rangle)\,(\breve{v}_s - \langle \breve{v}_s \rangle)'\,. \tag{3.6}$$

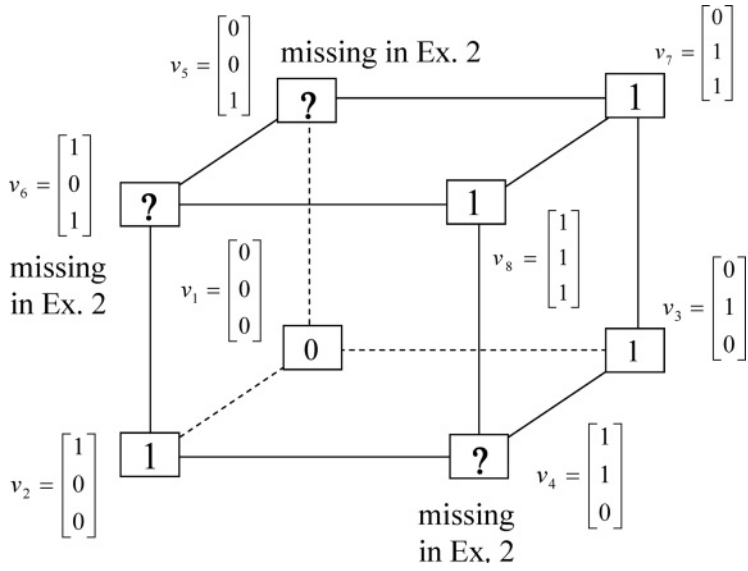

Figure 6: Data for supervised learning in example 2a, unsupervised learning in example 2b, and unsupervised learning in example 2c are shown as the vertices of a cube. The digits in the squares at the vertices are labels for supervised learning. The question marks indicate that the labels are unknown and unsupervised learning is necessary.

Note that the update formulas, 3.4 and 3.5, and equation 3.6 are the same as the update formulas, 3.1 and 3.2, and equation 3.3, respectively, except that u_t and $< u_t >$ are, respectively, replaced with w_t and $< w_t >$.

Example 2a. Consider a unit cube shown in Figure 6. The vectors, v_t, t =1, 2, . . . , 8, to be input to dendritic encoders in Examples 2a, 2b, and 2c, are shown at the vertices. The signals from a teaching D-neuron corresponding to v_t, $t = 1, 2, 3, 7, 8$, are available for supervised learning. They are binary digits w_t, $t = 1, 2, 3, 7, 8$, respectively, enclosed in the square boxes. The supervised training data consist of the pairs, (v_t, w_t), t = 1, 2, 3, 7, 8. The question marks in the square boxes indicate that no teaching signal is available for supervised learning.

The pairs, $(\check{v}'_t, w_t)$, $t = 1, 2, 3, 7, 8$, are listed as rows in Table 1.

Assume $\lambda = \Lambda = 1$, $\langle \check{v}_t \rangle = \mathbf{I}/2$, and $\langle w_t \rangle = \mathbf{I}/2$ in equation 3.5. The expansion correlation matrix D is

$$D = \left[-\tfrac{3}{4} \;\; \tfrac{1}{4} \;\; \tfrac{3}{4} \;\; \tfrac{3}{4} \;\; \tfrac{1}{4} \;\; \tfrac{1}{4} \;\; -\tfrac{1}{4} \;\; \tfrac{3}{4} \right]. \tag{3.7}$$

Table 1: Input Vectors, Their Dendritic Codes, and Labels for Supervised Covariance and Unsupervised Accumulation Learning.

$\breve{v}'_t$	0	v_{t1}	v_{t2}	$\phi(v_{t2}, v_{t1})$	v_{t3}	$\phi(v_{t3}, v_{t1})$	$\phi(v_{t3}, v_{t2})$	$\phi_3(v_{t3}, v_{t2}, v_{t1})$	w_t
$\breve{v}'_1$	0	0	0	0	0	0	0	0	0
$\breve{v}'_2$	0	1	0	1	0	1	0	1	1
$\breve{v}'_3$	0	0	1	1	0	0	1	1	1
$\breve{v}'_7$	0	0	1	1	1	1	0	0	1
$\breve{v}'_8$	0	1	1	0	1	0	0	1	1

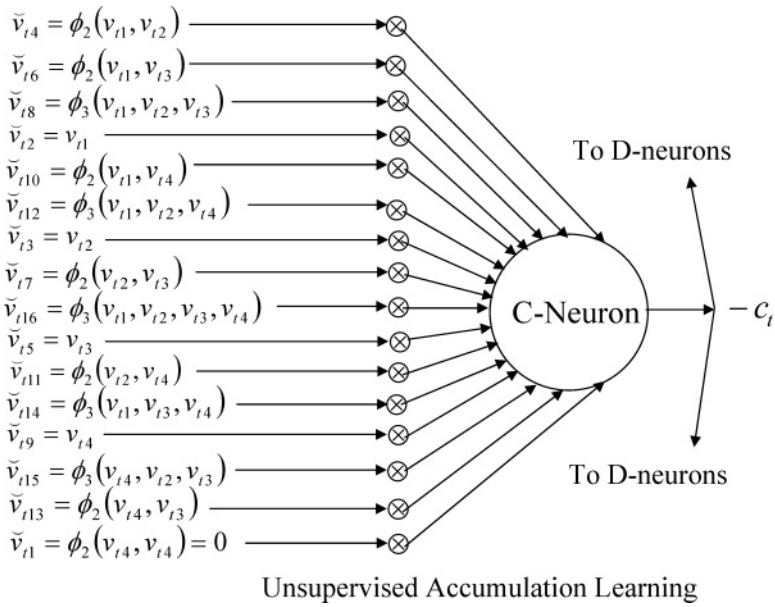

Figure 7: A C-neuron with an unsupervised accumulation learning mechanism is shown. The inhibitory graded output $-c_t$ is distributed to neighboring D-neurons.

3.3 Unsupervised Accumulation Rule. The 2^m synapses on the connections from the output terminals of a dendritic encoder to a C-neuron are updated by the unsupervised accumulation rule

$$C \leftarrow \lambda C + \frac{\Lambda}{2}(\breve{v}_t - \langle \breve{v}_t \rangle)'. \tag{3.8}$$

Figure 7 shows a C-neuron and its synapses $\otimes$. Note that the inhibitory graded output $-c_t$ from the C-neuron is not fedback for updating the synaptic strengths C, which is a 2^m-dimensional row vector. If the

deviations $\breve{v}_s - \langle \breve{v}_s \rangle$, $s = 1, \ldots, t$, have been accumulated in the 2^m synapses, the strengths or weights in them are the row vector,

$$C = \frac{\Lambda}{2} \sum_{s=1}^{t} \lambda^{t-s} (\breve{v}_s - \langle \breve{v}_s \rangle)'. \tag{3.9}$$

Example 2b. This is a continuation of example 2a. For the training data, x_t, $t = 1, 2, 3, 7, 8$, from example 2a, which is shown in Figure 6, the expansion correlation matrix C is

$$C = \left[-\tfrac{5}{4} \ -\tfrac{1}{4} \ \tfrac{1}{4} \ \tfrac{1}{4} \ -\tfrac{1}{4} \ -\tfrac{1}{4} \ -\tfrac{3}{4} \ \tfrac{1}{4} \right]. \tag{3.10}$$

Note that the teaching signals, $w_t, t = 1, 2, 3, 7, 8$, are not needed in obtaining C by the unsupervised accumulation learning, equation 3.8.

How the dendritic encoders, synapses, D-neuron, C-neuron, and three learning mechanisms in examples 2a and 2b are connected is shown in Figure 8. The inputs to the dendritic encoder are $v_{\tau 1}$, $v_{\tau 2}$, $v_{\tau 3}$. The masking matrices M in the figure will be used in example 2c in section 5. The outputs of the dendritic encoder are 0, $v_{\tau 1}$, $v_{\tau 2}$, $\phi(v_{\tau 2}, v_{\tau 1})$, v_3, $\phi(v_{\tau 3}, v_{\tau 1})$, $\phi(v_{\tau 3}, v_{\tau 2})$, $\phi_3(v_{\tau 3}, v_{\tau 2}, v_{\tau 1})$. For the synapses preceding the D-neuron to perform supervised covariance learning, the selection lever, represented by the thick line segment with a circle at its end, is placed in the top position to receive a teaching signal r_τ provided from outside. For these synapses to perform unsupervised covariance learning, the lever is placed in the bottom position to receive a spike or nonspike $v\{y_\tau\}$ output from the D-neuron. For these synapses to perform no learning, the lever is placed in the middle position to receive the signal 1/2. The selection lever of a biological synapse is usually permanently set at one position for one type of learning. It is not clear if a biological synapse can perform supervised, unsupervised, and no learning alternately.

In Figure 8, the C-neuron and D-neuron share the same dendritic encoder and its outputs. This is not necessary in modeling biological neural networks. As long as the C-neuron and D-neuron jointly provide enough information to produce a good estimate of the subjective probability distribution, they may have different dendritic encoders.

4 Retrieving Information from Synapses

Once a vector v_τ is received by a dendritic encoder, v_τ is encoded by the dendritic encoder into $\breve{v}_\tau$, which is made available to synapses for learning as well as retrieving of a point estimate of a representation of the input v_τ. This representation is called the label of v_τ and denoted by r_τ. Recall that learned information is stored in the expansion covariance matrices, D and

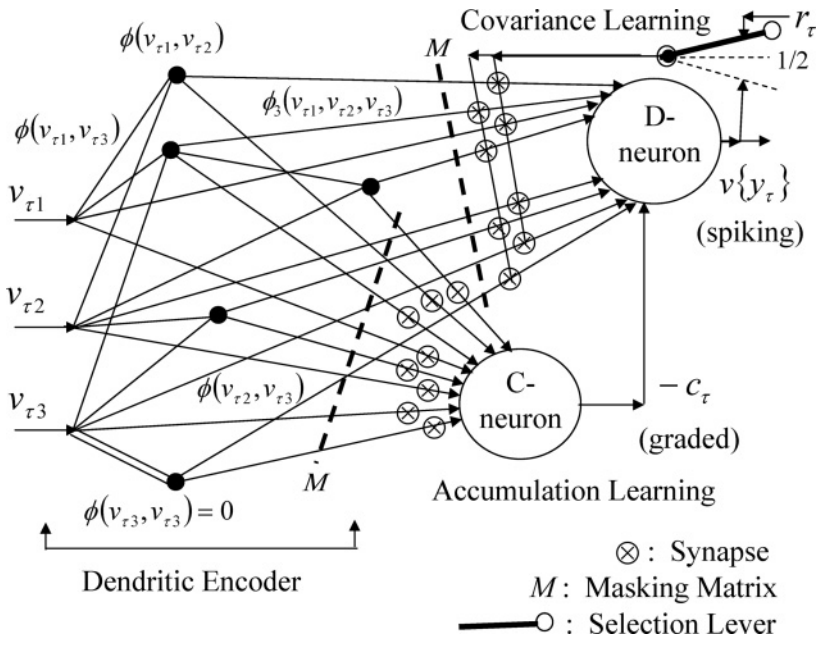

Figure 8: The dendritic trees, the C-neuron, the D-neuron, the synapses, the unsupervised accumulation and unsupervised and supervised covariance learning mechanisms, and the masking matrix M that are used in examples 2a, 2b, and 2c are shown. The selection lever represented by a thick line segment with a circle at its end can be placed in the top, bottom, or middle position for supervised, unsupervised, and no learning, respectively.

C. Upon the arrival of $\check{v}_\tau$, the following products, d_τ and c_τ, are computed by the synapses preceding the R D-neurons and 1 C-neuron:

$$d_\tau = D\left(\check{v}_\tau - \langle \check{v}_\tau \rangle\right), \tag{4.1}$$

$$c_\tau = C\left(\check{v}_\tau - \langle \check{v}_\tau \rangle\right), \tag{4.2}$$

where d_τ is an R-dimensional vector and c_τ is a scalar.

To gain some intuitive understanding of the meanings of d_τ and c_τ, let us assume that $\lambda = 1$ and that the averages, $\langle \check{v}_s \rangle$, $\langle \check{v}_\tau \rangle$, and $\langle r_s \rangle$, are all equal to $\mathbf{I}/2$, where r_s denotes u_s or w_s for unsupervised and supervised learning, respectively, and is an R-dimensional binary vector. Note that the time averages of biological spike trains are usually close to $1/2$ and that the forgetting factor λ is believed to be very close to 1 (fortunately). Here, $\mathbf{I} = \left[1\ 1\ \ldots\ 1\right]'$, which we note is not the identity matrix I. The only

problem with the assumption is that $\breve{v}_{s1} = 0$ and, hence, $\langle \breve{v}_{s1} \rangle = 0 \neq 1/2$. If the dimensionality of the vector $\breve{v}_s$ is large, the effect of missing this one term of $1/2$ is negligible. Nevertheless, it would be interesting to see whether and how this one term exists in biological neural networks.

Under the above assumptions,

$$\begin{aligned} d_\tau &= \Lambda \sum_{s=1}^{t} (r_s - \langle r_s \rangle) (\breve{v}_s - \langle \breve{v}_s \rangle)' (\breve{v}_\tau - \langle \breve{v}_\tau \rangle) \\ &= \Lambda \sum_{s=1}^{t} \left(r_s - \frac{1}{2}\mathbf{I} \right) \left(\breve{v}_s - \frac{1}{2}\mathbf{I} \right)' \left(\breve{v}_\tau - \frac{1}{2}\mathbf{I} \right) \end{aligned}$$

and

$$\begin{aligned} c_\tau &= \frac{\Lambda}{2} \sum_{s=1}^{t} (\breve{v}_s - \langle \breve{v}_s \rangle)' (\breve{v}_\tau - \langle \breve{v}_\tau \rangle) \\ &= \frac{\Lambda}{2} \sum_{s=1}^{t} \left(\breve{v}_s - \frac{1}{2}\mathbf{I} \right)' \left(\breve{v}_\tau - \frac{1}{2}\mathbf{I} \right). \end{aligned}$$

Under the further assumption that v_s and v_τ are binary vectors, by equations 2.6 and 2.7, which are proven in the appendix, if $v_s \neq v_\tau$, then

$$\left(\breve{v}_s - \frac{1}{2}\mathbf{I} \right)' \left(\breve{v}_\tau - \frac{1}{2}\mathbf{I} \right) = 0, \tag{4.3}$$

and if $v_s = v_\tau$, then

$$\left(\breve{v}_s - \frac{1}{2}\mathbf{I} \right)' \left(\breve{v}_\tau - \frac{1}{2}\mathbf{I} \right) = 2^{m-2}. \tag{4.4}$$

Therefore, under the above assumptions and the assumption that v_s and v_τ are binary vectors, for $j = 1, \ldots, R$,

$$\begin{aligned} d_{\tau j} &= \Lambda \left(\sum_{\substack{s=1 \\ v_s = v_\tau, r_{sj} = 1}}^{t} + \sum_{\substack{s=1 \\ v_s = v_\tau, r_{sj} = 0}}^{t} \right) \left(r_{sj} - \frac{1}{2} \right) 2^{m-2} \\ &= \Lambda \sum_{\substack{s=1 \\ v_s = v_\tau, r_{sj} = 1}}^{t} \left(\frac{1}{2} \right) 2^{m-2} + \Lambda \sum_{\substack{s=1 \\ v_s = v_\tau, r_{sj} = 0}}^{t} \left(-\frac{1}{2} \right) 2^{m-2} \end{aligned}$$

$$= 2^{m-3}\Lambda \left|\left\{v_s | v_s = v_\tau, r_{sj} = 1, s \in \{1, \ldots, t\}\right\}\right|$$
$$-2^{m-3}\Lambda \left|\left\{v_s | v_s = v_\tau, r_{sj} = 0, s \in \{1, \ldots, t\}\right\}\right|$$

and

$$c_\tau = \frac{\Lambda}{2} \sum_{\substack{s=1 \\ v_s = v_\tau}}^{t} 2^{m-2}$$
$$= 2^{m-3}\Lambda \left|\{v_s | v_s = v_\tau, s \in \{1, \ldots, t\}\}\right|$$
$$= 2^{m-3}\Lambda \left|\left\{v_s | v_s = v_\tau, r_{sj} = 1, s \in \{1, \ldots, t\}\right\}\right|$$
$$+ 2^{m-3}\Lambda \left|\left\{v_s | v_s = v_\tau, r_{sj} = 0, s \in \{1, \ldots, t\}\right\}\right|,$$

where $|S|$ denotes the number of elements in the set S.

Denoting $(c_\tau + d_{\tau j})/2$ by $a_{\tau j}$,

$$\frac{a_{\tau j}}{c_\tau} = \frac{\left|\left\{v_s | v_s = v_\tau, r_{sj} = 1, s \in \{1, \ldots, t\}\right\}\right|}{\left|\{v_s | v_s = v_\tau, s \in \{1, \ldots, t\}\}\right|}.$$

This is the relative frequency that $r_{sj} = 1$ has been learned for $v_s = v_\tau$ for $s = 1, \ldots, t$. Let $a_\tau = [\, a_{\tau 1}\ a_{\tau 2}\ \ldots\ a_{\tau R}\,]'$. Then a_τ / c_τ is a relative frequency distribution of r_τ given v_τ.

v_τ may be shared by more than one cause (or pattern) and may contain corruption, distortion, occlusion, or noise caused directly or indirectly by the sensor measurements such as image pixels and sound recordings. In this case, the label r_τ of v_τ should be a random variable, which can be described or represented only by a probability distribution (or a relative frequency distribution). On the other hand, v_τ may contain parts from more than one cause. In this case, the label r_τ of v_τ should be a fuzzy logic variable, which can be described or represented only by its membership function (Zadeh, 1965; Zadeh, Klir, & Yuan, 1996). Fortunately, both the probabilities and truth values range between 0 and 1. The former can be learned mainly as relative frequencies over time and the latter mainly as relative proportions in each v_τ as represented by relative frequencies. $a_{\tau j}/c_\tau$ evaluated above is a relative frequency representing a probability or a truth value. The three learning rules in the preceding section facilitate learning both the subjective probabilities and truth values and sometimes a combination of them. For simplicity, truth value and membership function will be referred to as subjective probability and subjective probability distribution in this letter. The fact that the subjective probability distribution (or membership function)

of r_τ can be retrieved from the synapses is striking but mathematically and naturally necessary Lo (2010).

As remarked at the end of the appendix, if any number of components in $\breve{v}_\tau$ change their signs and the corresponding components in $\breve{v}_s$ and $\mathbf{I}$ change their signs, then the orthogonality property, equations 2.6 and 2.7, still holds. If $\breve{v}_{tq}$ is inhibitory, the qth components of $\breve{v}_s - \langle \breve{v}_s \rangle$ and $\breve{v}_\tau - \langle \breve{v}_\tau \rangle$ change their signs. If additionally $\langle \breve{v}_s \rangle = \langle \breve{v}_\tau \rangle = -\mathbf{I}/2$, then $a_{\tau j}/c_\tau$ is still the relative frequency that $r_{sj} = 1$ has been learned for $v_s = v_\tau$. In general, $a_{\tau j}/c_\tau$ above is this relative frequency regardless of how many of the dendritic encoder' outputs $\breve{v}_t$ are inhibitory.

Example 1b. This example is a continuation of example 1a. With the v and u from example 1a, let a supervised training data set consist of 8 copies of u with label 1 and 2 copies of u with label 0 and 3 copies of v with label 1 and 27 copies of v with label 0. By equations 3.6 and 3.9, this supervised training data set is learned with $\lambda = \Lambda = 1$ (in equations 3.6 and 3.9) to form the ECMs (expansion correlation matrices):

$$D = \frac{1}{2}(8-2)\left(\breve{u} - \frac{1}{2}\mathbf{I}\right)' + \frac{1}{2}(3-27)\left(\breve{v} - \frac{1}{2}\mathbf{I}\right)',$$

$$C = \frac{1}{2}(8+2)\left(\breve{u} - \frac{1}{2}\mathbf{I}\right)' + \frac{1}{2}(3+27)\left(\breve{v} - \frac{1}{2}\mathbf{I}\right)'.$$

By equations 2.6 and 2.7, $D(\breve{u} - \frac{1}{2}\mathbf{I}) = 3(4)$, $C(\breve{u} - \frac{1}{2}\mathbf{I}) = 5(4)$, $D(\breve{v} - \frac{1}{2}\mathbf{I}) = -12(4)$, $C(\breve{v} - \frac{1}{2}\mathbf{I}) = 15(4)$. It follows that $(D(\breve{u} - \frac{1}{2}\mathbf{I}) + C(\breve{u} - \frac{1}{2}\mathbf{I}))/(2C(\breve{u} - \frac{1}{2}\mathbf{I})) = 8/10$ is the relative frequency that u has been learned with label 1, and $1 - 8/10 = 2/10$ is the relative frequency that u has been learned with label 0. Similarly, $(D(\breve{v} - \frac{1}{2}\mathbf{I}) + C(\breve{v} - \frac{1}{2}\mathbf{I}))/(2C(\breve{v} - \frac{1}{2}\mathbf{I})) = 3/30$ is the relative frequency that v has been learned with label 1, and $1 - 3/30 = 27/30$ is the relative frequency that v has been learned with label 0.

A few words about the morphology of a model dendritic tree are useful here. Each model dendritic tree consists of the upper parts and lower parts connected by trainable synapses $\otimes$. The upper parts form dendritic encoders, and dendrites in the lower parts may merge in many ways before reaching the postsynaptic neuron. In Figures 3, 4, 5, 7, and 8 the tree structures of model dendritic trees are partly suppressed to keep the graph sizes small. In Figure 9, a small model dendritic tree is shown as a binary tree. The inputs, $v_{\tau 1}$, $v_{\tau 1}$, and $v_{\tau 1}$ are distributed to the 14 "tree tips" by axons preceding the dendritic tree. Both the lower and upper parts of the model dendritic tree resemble morphologically their corresponding parts in a biological dendritic tree. Neither the lower nor the upper parts of the model dendritic tree are unique.

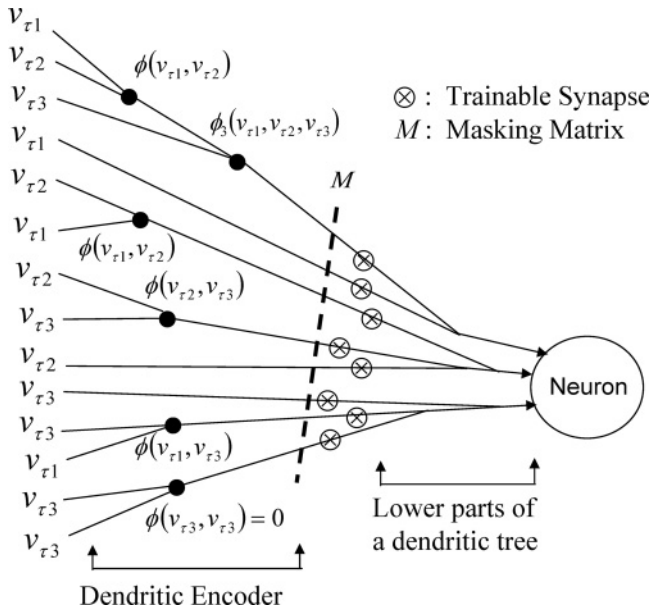

Figure 9: A model dendritic tree is shown here, which is a binary tree. It comprises upper and lower parts connected by trainable synapses ⊗. The upper parts form a model dendritic encoder, where the branch points are model dendritic nodes. The branch points in the lower parts serve simply for summing. The model dendritic encoder is not unique for the same input vector $v_\tau = [v_{\tau 1} \quad v_{\tau 2} \quad v_{\tau 3}]'$ and the same output vector $\breve{v}_\tau$ input to the trainable synapses ⊗. The outputs of the synapses are a row vector $[c_{\tau j}]$ or a matrix $[d_{\tau kj}]$ depending on whether the post - synaptic neuron is a C-neuron or a D-neuron. The summation, $c_\tau = \sum_j [c_{\tau j}]$ or $d_{\tau k} = \sum_j [d_{\tau kj}]$, is computed by the lower parts of the model dendritic tree and the postsynaptic neuron. The division of computation between the lower parts and the neuron is not unique. Given a division, the merges of dendrites in the lower parts are not unique.

5 Maximal Generalization in Information Retrieval

Let a vector v_τ that deviates from each of the vectors v_s that have been learned by the synapses on a dendritic encoder due to corruption, distortion, or occlusion be presented to the dendritic encoder. The dendritic tree and its synapses are said to have a maximal generalization capability in their retrieval of information if they are able to automatically find the largest subvector of v_τ that matches at least one subvector among the vectors v_s stored in the synapses and enable postsynaptic neurons to generate the subjective probability distribution of the label of the largest subvector. This maximal capability is achieved by the use of a masking matrix described in

this section. A biological interpretation of such a matrix is given at the end of this section.

Slepian (1956) discovered the following orthogonal expansion of bipolar binary vectors $a = [a_1 \quad a_2 \quad \dots \quad a_m]'$:

$$\begin{aligned} \hat{a}(1) &= \begin{bmatrix} 1 & a_1 \end{bmatrix}', \\ \hat{a}(1,\dots,j+1) &= \begin{bmatrix} \hat{a}'(1,\dots,j) & a_{j+1}\hat{a}'(1,\dots,j) \end{bmatrix}' \\ &\qquad \text{for } j = 1,\dots,m-1, \\ \hat{a} &= \hat{a}(1,\dots,m). \end{aligned} \tag{5.1}$$

$\hat{a}$ is called the orthogonal expansion of a. For example, if $a = [a_1 \quad a_2 \quad a_3]$, then

$$\hat{a} = \begin{bmatrix} 1 & a_1 & a_2 & a_2a_1 & a_3 & a_3a_1 & a_3a_2 & a_3a_2a_1 \end{bmatrix}.$$

Let us denote the vector $a = [a_1 \quad a_2 \quad \dots \quad a_m]'$ with its i_1th, i_2th, ..., and i_jth components set equal to 0 by $a(i_1^-, i_2^-, \dots, i_j^-)$, where $1 \le i_1 < i_2 < \cdots < i_j \le m$, and the dendritic and orthogonal expansions of $a(i_1^-, i_2^-, \dots, i_j^-)$ by $\check{a}(i_1^-, i_2^-, \dots, i_j^-)$ and $\hat{a}(i_1^-, i_2^-, \dots, i_j^-)$, respectively. Denoting the m-dimensional vector $[1 \quad 1 \quad \dots \quad 1]'$ by $\mathbf{I}$, the vector $\mathbf{I}$ with its i_1th, i_2th, ..., and i_jth components set equal to 0 is $\mathbf{I}(i_1^-, i_2^-, \dots, i_j^-)$ and the orthogonal expansion of $\mathbf{I}(i_1^-, i_2^-, \dots, i_j^-)$ is $\hat{\mathbf{I}}(i_1^-, i_2^-, \dots, i_j^-)$. For example, if $\mathbf{I} = [1 \quad 1 \quad 1]'$, then

$$\begin{aligned} \hat{\mathbf{I}}(1^-) &= \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}, \\ \hat{\mathbf{I}}(2^-) &= \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}, \\ \hat{\mathbf{I}}(3^-) &= \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \end{aligned} \tag{5.2}$$

Notice that for the vector $a = [a_1 \quad a_2 \quad a_3]'$,

$$\begin{aligned} \check{a}(1^-) &= \begin{bmatrix} 0 & 0 & a_2 & 0 & a_3 & 0 & \phi(a_3,a_2) & 0 \end{bmatrix} = \left(\text{diag}\hat{\mathbf{I}}(1^-)\right)\check{a}, \\ \check{a}(2^-) &= \begin{bmatrix} 0 & a_1 & 0 & 0 & a_3 & \phi(a_3,a_1) & 0 & 0 \end{bmatrix} = \left(\text{diag}\hat{\mathbf{I}}(2^-)\right)\check{a}, \\ \check{a}(3^-) &= \begin{bmatrix} 0 & a_1 & a_2 & \phi(a_2,a_1) & 0 & 0 & 0 & 0 \end{bmatrix} = \left(\text{diag}\hat{\mathbf{I}}(3^-)\right)\check{a}. \end{aligned}$$

In general, for the vector $a = [\,a_1 \quad a_2 \quad \dots \quad a_m\,]'$,

$$\breve{a}\left(i_1^-, i_2^-, \dots, i_j^-\right) = \left(\text{diag}\hat{\mathbf{I}}\left(i_1^-, i_2^-, \dots, i_j^-\right)\right)\breve{a}.$$

Notice that $\text{diag}\hat{\mathbf{I}}(i_1^-, i_2^-, \dots, i_j^-)$ eliminates components in $\breve{a}$ that involve a_{i_1}, $a_{i_2}, \dots, a_{i_j}$. Therefore, $\text{diag}\hat{\mathbf{I}}(i_1^-, i_2^-, \dots, i_j^-)$ is called a masking matrix.

An important property of the masking matrix $\text{diag}\hat{\mathbf{I}}(i_1^-, i_2^-, \dots, i_j^-)$ is the following. Assume that v_s and v_τ are binary vectors. If

$$\left(\text{diag}\hat{\mathbf{I}}(i_1^-, i_2^-, \dots, i_j^-)\right)\left(\breve{v}_s - \frac{1}{2}\mathbf{I}\right)$$
$$= \left(\text{diag}\hat{\mathbf{I}}(i_1^-, i_2^-, \dots, i_j^-)\right)\left(\breve{v}_\tau - \frac{1}{2}\mathbf{I}\right),$$

then

$$(\breve{v}_s - \langle\breve{v}_s\rangle)'\left(\text{diag}\hat{\mathbf{I}}(i_1^-, i_2^-, \dots, i_j^-)\right)(\breve{v}_\tau - \langle\breve{v}_\tau\rangle) = 2^{m-2-j}. \tag{5.3}$$

If

$$\left(\text{diag}\hat{\mathbf{I}}(i_1^-, i_2^-, \dots, i_j^-)\right)\left(\breve{v}_s - \frac{1}{2}\mathbf{I}\right)$$
$$\neq \left(\text{diag}\hat{\mathbf{I}}(i_1^-, i_2^-, \dots, i_j^-)\right)\left(\breve{v}_\tau - \frac{1}{2}\mathbf{I}\right),$$

then

$$(\breve{v}_s - \langle\breve{v}_s\rangle)'\left(\text{diag}\hat{\mathbf{I}}(i_1^-, i_2^-, \dots, i_j^-)\right)(\breve{v}_\tau - \langle\breve{v}_\tau\rangle) = 0. \tag{5.4}$$

Using this property, we combine all such masking matrices that set less than or equal to a selected positive integer J of components of v_s equal to 0 into the following masking matrix,

$$M = I + \sum_{j=1}^{J}\sum_{i_j=j}^{m}\cdots\sum_{i_2=2}^{i_3-1}\sum_{i_1=1}^{i_2-1} 2^{-5j}2^j\,\text{diag}\hat{\mathbf{I}}\left(i_1^-, i_2^-, \dots, i_j^-\right), \tag{5.5}$$

where 2^j is used to compensate for the factor 2^{-j} in 2^{m-2-j} in the important property 5.3, and 2^{-5j} is an example weight selected to differentiate between different levels j of maskings.

The information retrieval formula, equations 4.1 and 4.2, is replaced with

$$d_\tau = DM(\breve{v}_\tau - \langle \breve{v}_\tau \rangle), \tag{5.6}$$

$$c_\tau = CM(\breve{v}_\tau - \langle \breve{v}_\tau \rangle). \tag{5.7}$$

Note that for $k = 1, \ldots, R$, we have the following:

- If $C(\breve{v}_\tau - \langle \breve{v}_\tau \rangle) \neq 0$, then

$$D_k(\breve{v}_\tau - \langle \breve{v}_\tau \rangle) \approx D_k M(\breve{v}_\tau - \langle \breve{v}_\tau \rangle),$$

$$C(\breve{v}_\tau - \langle \breve{v}_\tau \rangle) \approx CM(\breve{v}_\tau - \langle \breve{v}_\tau \rangle).$$

- If $C(\breve{v}_\tau - \langle \breve{v}_\tau \rangle) = 0$, but $C \sum_{i_1=1}^{m} (\mathrm{diag}\hat{\mathbf{I}}(i_1^-))(\breve{v}_\tau - \langle \breve{v}_\tau \rangle) \neq 0$, then

$$D_k \sum_{i_1=1}^{m} (\mathrm{diag}\hat{\mathbf{I}}(i_1^-))(\breve{v}_\tau - \langle \breve{v}_\tau \rangle) \approx D_k M(\breve{v}_\tau - \langle \breve{v}_\tau \rangle),$$

$$C \sum_{i_1=1}^{m} (\mathrm{diag}\hat{\mathbf{I}}(i_1^-))(\breve{v}_\tau - \langle \breve{v}_\tau \rangle) \approx CM(\breve{v}_\tau - \langle \breve{v}_\tau \rangle).$$

- If $C(\breve{v}_\tau - \langle \breve{v}_\tau \rangle) = 0$, $C \sum_{i_1=1}^{m} (\mathrm{diag}\hat{\mathbf{I}}(i_1^-))(\breve{v}_\tau - \langle \breve{v}_\tau \rangle) = 0$, but $C \sum_{i_2=2}^{m} \sum_{i_1=1}^{i_2-1} (\mathrm{diag}\hat{\mathbf{I}}(i_1^-, i_2^-))(\breve{v}_\tau - \langle \breve{v}_\tau \rangle) \neq 0$, then

$$D_k \sum_{i_2=2}^{m} \sum_{i_1=1}^{i_2-1} (\mathrm{diag}\hat{\mathbf{I}}(i_1^-, i_2^-))(\breve{v}_\tau - \langle \breve{v}_\tau \rangle) \approx D_k M(\breve{v}_\tau - \langle \breve{v}_\tau \rangle),$$

$$C \sum_{i_2=2}^{m} \sum_{i_1=1}^{i_2-1} (\mathrm{diag}\hat{\mathbf{I}}(i_1^-, i_2^-))(\breve{v}_\tau - \langle \breve{v}_\tau \rangle) \approx CM(\breve{v}_\tau - \langle \breve{v}_\tau \rangle).$$

Continuing in this manner, it is seen that $D_k M(\breve{v}_\tau - \langle \breve{v}_\tau \rangle)$ and $CM(\breve{v}_\tau - \langle \breve{v}_\tau \rangle)$ always use the greatest number of uncorrupted, undistorted, or unoccluded components of v_τ in estimating $d_{\tau k}$, c_τ, and $a_{\tau k}$. The vector $a_{\tau j}/c_\tau$ is now an estimate of the subjective conditional probability distribution of r_τ given v_τ, using the greatest number of uncorrupted, undistorted, or unoccluded components of v_τ.

If some terms in equation 5.5 are missing, the generalization effect of M degrades only gracefully. The example weight 2^{-5j} in equation 5.5 is used to illustrate M generalizing maximally in example 2c. The weight is believed to be 2^{-j} in biological neural networks for a reason to be discussed. J in equation 5.5 is believed to vary from brain region to brain region. The range of J can be found by biological experiments.

Table 2: Input Vectors and Their Dentritic Codes for Retrieval and Unsupervised Covariance and Unsupervised Accumulation Learning.

$\check{v}'_t$	0	v_{t1}	v_{t2}	$\phi(v_{t2}, v_{t1})$	v_{t3}	$\phi(v_{t3}, v_{t1})$	$\phi(v_{t3}, v_{t2})$	$\phi(v_{t3}, v_{t2}, v_{t1})$
$\check{v}'_4$	0	1	1	0	0	1	1	0
$\check{v}'_5$	0	0	0	0	1	1	1	1
$\check{v}'_6$	0	1	0	1	1	0	1	0

Example 2c. Let the expansion covariance matrices, D and C, be those obtained in examples 2a and 2b. Using equation 5.2, we construct the masking matrix by formula 5.5 for $J = 1$:

$$M = I + 2^{-5}2\text{diag}\left(\hat{\mathbf{I}}(1^-) + \hat{\mathbf{I}}(2^-) + \hat{\mathbf{I}}(3^-)\right)$$
$$= I + 2^{-4}\text{diag}\begin{bmatrix} 3 & 2 & 2 & 1 & 2 & 1 & 1 & 0 \end{bmatrix} \tag{5.8}$$

Recall that with the masking matrix M, we use $d_{\tau j} = D_{\tau j}M(\check{v}_\tau - \langle\check{v}_\tau\rangle)$ and $c_\tau = C_\tau M(\check{v}_\tau - \langle\check{v}_\tau\rangle)$ in general, where D_j denotes the jth row of D. If $c_\tau \neq 0$, the subjective probability $p_{\tau j} = (d_{\tau j}/c_\tau + 1)/2$. The masking matrix M for this example is shown in Figure 8.

Assume that $\check{v}_1$ is presented to the synapses containing the expansion covariance matrices through M. By matrix multiplication,

$$DM(\check{v}_1 - \langle\check{v}_1\rangle) = -1 + 2^{-4}(-1/2) = -1.0312,$$
$$CM(\check{v}_1 - \langle\check{v}_1\rangle) = 1 + 2^{-4}(5/2) = 1.1563.$$

The subjective probability that the label of v_4 is 1 is $(DM(\check{v}_1 - \langle\check{v}_1\rangle)/(CM(\check{v}_1 - \langle\check{v}_1\rangle)) + 1)/2 = 0.054$, and the subjective probability that the label of v_4 is 0 is 0.946. Note that v_1 with a label of 0 has been learned. The subjective probability that the label of v_4 is 0 should be 1. The use of M causes a very small amount of error to the subjective probability, which can be controlled by changing the weight, 2^{-5}.

The dendritic expansions of the three vertices, v_4, v_5, and v_6, of the cube in Figure 4, which are not included in the supervised learning data, are listed in Table 2.

Simple matrix-vector multiplication yields $D(\check{v}_t - \langle\check{v}_t\rangle) = 0$ and $C(\check{v}_t - \langle\check{v}_t\rangle) = 0$ for $t = 4, 5, 6$. Hence no information is provided on v_t by $D(\check{v}_t - \langle\check{v}_t\rangle)$ and $C(\check{v}_t - \langle\check{v}_t\rangle)$ for $t = 4, 5, 6$. This shows that if v_t has not been learned, then generalization is necessary to get information on it from the expansion covariance matrices.

Assume that $\check{v}_4$ is presented to the synapses containing the expansion covariance matrices. By matrix multiplication,

$$DM(\check{v}_4 - \langle\check{v}_4\rangle) = 0 + 2^{-10}(9 + 2 + 6 - 3 - 2 + 1 - 1) = 2^{-10}(12),$$
$$CM(\check{v}_4 - \langle\check{v}_4\rangle) = 0 + 2^{-10}(15 - 2 + 2 - 1 + 2 - 1 - 3) = 2^{-10}(12).$$

The subjective probability that the label of v_4 is 1 is $(DM(\check{v}_4 - \langle\check{v}_4\rangle)/(CM(\check{v}_4 - \langle\check{v}_4\rangle)) + 1)/2 = 1$. From Figure 4, we see that the three vertices neighboring v_4 have been learned and have a label of 1. It is a good generalization that a label of 1 is assigned to v_4.

Assume that $\check{v}_6$ is presented to the expansion covariance matrices. By matrix multiplication,

$$DM(\check{v}_6 - \langle\check{v}_6\rangle) = 0 + 2^{-7}(9 + 2 - 6 + 3 + 2 - 1 - 1) = 2^{-7}(8), \quad (5.9)$$
$$CM(\check{v}_6 - \langle\check{v}_6\rangle) = 0 + 2^{-7}(15 - 2 - 2 + 1 - 2 + 1 - 3) = 2^{-7}(8). \quad (5.10)$$

Then the subjective probability that the label of v_6 is 1 is $(DM(\check{v}_6 - \langle\check{v}_6\rangle)/(CM(\check{v}_6 - \langle\check{v}_6\rangle)) + 1)/2 = 1$. From Figure 4, we see that only two vertices neighboring v_4 have been learned, and both have a label of 1. It is a good generalization that a label of 1 is assigned to v_6.

Assume that $\check{v}_5$ is presented to the synapses containing the expansion covariance matrices. By matrix multiplication,

$$DM(\check{v}_5 - \langle\check{v}_5\rangle) = 0 + 2^{-7}(9 - 2 - 6 - 3 + 2 + 1 - 1) = 2^{-7}(0),$$
$$CM(\check{v}_5 - \langle\check{v}_5\rangle) = 0 + 2^{-7}(15 + 2 - 2 - 1 - 2 - 1 - 3) = 2^{-7}(8).$$

Then the subjective probability that the label of v_5 is 1 is $(DM(\check{v}_5 - \langle\check{v}_5\rangle)/(CM(\check{v}_5 - \langle\check{v}_5\rangle)) + 1)/2 = 1/2$. From Figure 4, we see that only two vertices neighboring v_4 have been learned, and one of them has a label of 1 and the other a label of 0. No generalization is possible. A label of 1 is assigned to v_6 with a subjective probability of 1/2, and a label of 0 is assigned to v_6 with equal subjective probability.

The example shows that the weight 2^{-5j} in equation 5.5 is more than adequate to differentiate levels j of maskings for a dendritic encoder with only three inputs. The greater the number m of inputs to a dendritic encoder, the less that two adjacent levels, j and $j + 1$, of maskings need to be differentiated. For example, if the number m of components in the input vector is 12, any 11 of the 12 components should be almost as good as the 12 in determining the label of the input vector. A reduction by 50% of emphasis on the subvector as shown in equation 2.7 is usually adequate.

Therefore, if m is 12 or larger, the weight in equation 5.5 can be set equal to 2^{-j} so that the reduction of emphasis by 50% from level $j + 1$

and j is adequate. In this case, the masking matrix M is a mathematical idealization and organization of a large number of dendritic trees with nested and overlapped inputs. The following example illustrates this biological interpretation of the masking matrix M with the weight being 2^{-j} in equation 5.5.

Example 3. Using equation 5.2, we construct the masking matrix by formula 5.5 for $J = 1$,

$$\begin{aligned} M &= I + 2^{-1}2\text{diag}\left(\hat{\mathbf{I}}\left(1^{-}\right) + \hat{\mathbf{I}}\left(2^{-}\right) + \hat{\mathbf{I}}\left(3^{-}\right)\right) \\ &= I + \text{diag}\hat{\mathbf{I}}\left(1^{-}\right) + \text{diag}\hat{\mathbf{I}}\left(2^{-}\right) + \text{diag}\hat{\mathbf{I}}\left(3^{-}\right). \end{aligned} \tag{5.11}$$

In the product

$$M\check{v}_\tau = \check{v}_\tau + \left(\text{diag}\hat{\mathbf{I}}(1^{-})\right)\check{v}_\tau + \left(\text{diag}\hat{\mathbf{I}}(2^{-})\right)\check{v}_\tau + \left(\text{diag}\hat{\mathbf{I}}(3^{-})\right)\check{v}_\tau,$$

$\left(\text{diag}\hat{\mathbf{I}}(k^{-})\right)\check{v}_\tau$ eliminates all terms in $\check{v}_\tau$ that contain $v_{\tau k}$ and can be viewed as the output vector of a model dendritic encoder for $k = 1, 2, 3$. These model dendritic encoders are shown in Figure 10. $M\check{v}_\tau$ can be viewed as the sum of the output vectors from four model dendritic encoders with nested and overlapped input vectors, $v_\tau = [\, v_{\tau 1}\ v_{\tau 2}\ v_{\tau 3}\,]'$, $[\, v_{\tau 2}\ v_{\tau 3}\,]'$, $[\, v_{\tau 1}\ v_{\tau 3}\,]'$, and $[\, v_{\tau 1}\ v_{\tau 2}\,]'$.

6 Processing Units and Multiple Dendritic Trees

It is hypothesized that biological neural networks are organized with processing units (PU), each comprising dendritic encoders, synapses, a nonspiking neuron, spiking neurons, and learning and retrieving mechanisms. Let us denote the vector input to a model PU by v_t and the number of model spiking neurons (i.e., D-neurons) by R. In the rest of this letter, model PUs will be called PUs. PUs may have different numbers R of D-neurons. Recall that if $v_t = [\, v_{t1}\ v_{t2}\ \cdots\ v_{tN}\,]$, then the dendritic expansion $\check{v}_t$ of v_t is 2^N-dimensional. If there were only one dendritic encoder in a PU, there would be two difficulties. First, 2^N grows exponentially as N increases. Second, a large number of terms in the masking matrix M in equation 5.5 are required for masking even a small proportion J/N of the components of v_t if N is large. The corresponding biological difficulties are evident.

The inputs to a biological PU form dendritic nodes and grow into dendritic encoders through binary operations ϕ and compositions of binary operations ϕ_k as described in section 2. The growth occurs essentially in various neighborhoods simultaneously. As the growth continues, smaller dendritic encoders grow into larger encoders. If the number of inputs to a

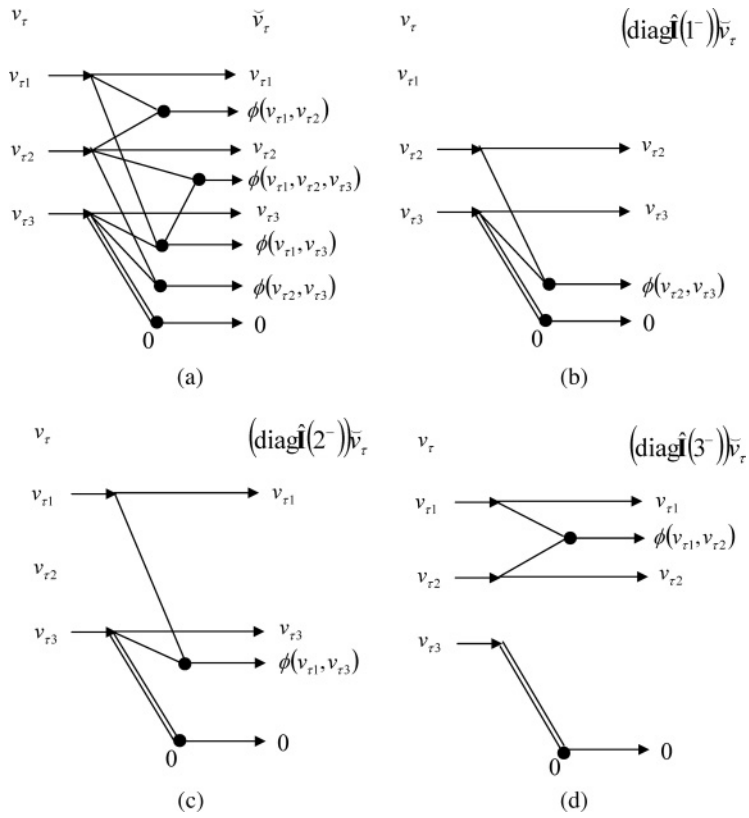

Figure 10: The product $M\check{v}_\tau$ in example 3 consists of four terms—$\check{v}_\tau$, $(\text{diag}\hat{\mathbf{I}}(1^-))\check{v}_\tau$, $(\text{diag}\hat{\mathbf{I}}(2^-))\check{v}_\tau$, and $(\text{diag}\hat{\mathbf{I}}(3^-))\check{v}_\tau$—where $(\text{diag}\hat{\mathbf{I}}(j^-))\check{v}_\tau$ has the components in $\check{v}_\tau$ that involve v_{τ_j} eliminated. These four terms can be viewed as the outputs of four model dendritic encoders shown in panels a–d in response to four overlapped and nested input vectors, $v_\tau = [v_{\tau 1}\ v_{\tau 2}\ v_{\tau 3}]$; $[v_{\tau 2}\ v_{\tau 3}]$; $[v_{\tau 1}\ v_{\tau 3}]$; and $[v_{\tau 1}\ v_{\tau 2}]$, respectively. The product $M\check{v}_\tau$ is the sum of these outputs. Although there are two parts for $(\text{diag}\hat{\mathbf{I}}(3^-))\check{v}_\tau$ in panel d, they can be replaced with a single dendritic encoder: the upper dendritic part in panel d with $\phi(v_{\tau 2}, v_{\tau 2}) = 0$ added to replace $\phi(v_{\tau 3}, v_{\tau 3}) = 0$ in the lower dendritic part.

biological PU is large, the growth falls short of forming one large single encoder for inputting all the inputs. The resultant smaller dendritic encoders are called compartments in neuroscience (Koch & Poggio, 1982, 1992; Koch et al., 1983; Rall & Sergev, 1987; Shepherd & Brayton, 1987; Mel, 1992a, 1992b, 1993, 1994, 2008).

Reflecting multiple dendritic encoders in a biological PU, let the number of model dendritic encoders in a (model) PU be denoted by Ψ; the vector

sequences input to them by $v_t(\psi)$, $\psi = 1, \ldots, \Psi$; the vectors output from them by $\breve{v}_t(\psi)$, $\psi = 1, \ldots, \Psi$; and the averages of these output vectors by $\langle \breve{v}_t(\psi) \rangle$, $\psi = 1, \ldots, \Psi$. We assemble these vectors into

$$\breve{v}_t = \left[\breve{v}_t'(1) \quad \breve{v}_t'(2) \quad \cdots \quad \breve{v}_t'(\Psi) \right]', \tag{6.1}$$

$$\langle \breve{v}_t \rangle = \left[\langle \breve{v}_t'(1) \rangle \quad \langle \breve{v}_t'(2) \rangle \quad \cdots \quad \langle \breve{v}_t'(\Psi) \rangle \right]'. \tag{6.2}$$

Notice that $\breve{v}_t$ here is not the single dendritic expansion of v_t defined in equation 2.5 but is called the general dendritic expansion of v_t. This dual use of the symbol is not expected to cause confusion. Note that the vectors $v_t(\psi)$, $\psi = 1, \ldots, \Psi$, may have common components and different dimensionalities, $\dim v_t(\psi)$, $\psi = 1, \ldots, \Psi$, but every component of v_t is included in at least one $v_t(\psi)$. Furthermore, the components of $v_t(\psi)$ are selected at random from those of v_t.

For the dendritic encoder output vectors, $\breve{v}_t(\psi)$, $\psi = 1, \ldots, \Psi$, let the expansion covariance matrices be denoted by $D(\psi)$ and $C(\psi)$, $\psi = 1, \ldots, \Psi$ and the masking matrices by $M(\psi)$, $\psi = 1, \ldots, \Psi$. We assemble these matrices into

$$D = \left[D(1) \quad D(2) \quad \cdots \quad D(\Psi) \right], \tag{6.3}$$

$$C = \left[C(1) \quad C(2) \quad \cdots \quad C(\Psi) \right], \tag{6.4}$$

$$M = \operatorname{diag} \left[M(1) \quad M(2) \quad \cdots \quad M(\Psi) \right]. \tag{6.5}$$

Here D and C are called general expansion covariance matrices, and M is called the general masking matrix for the Ψ dendritic encoders and the synapses on them. Note that D, C, and M above are not those defined for $2^{\dim v_t}$-dimensional dendritic expansions of v_t. The dual use of the symbols here is not expected to cause confusion.

D, C, and M above are $R \times \sum_{\psi=1}^{\Psi} 2^{\dim v_t(\psi)}$, $1 \times \sum_{\psi=1}^{\Psi} 2^{\dim v_t(\psi)}$, and $\sum_{\psi=1}^{\Psi} 2^{\dim v_t(\psi)} \times \sum_{\psi=1}^{\Psi} 2^{\dim v_t(\psi)}$ matrices, respectively. Note that by choosing $\dim v_t(\psi)$ smaller than $\dim v_t$, $2^{\dim v_t(\psi)}$ is much smaller than $2^{\dim \breve{v}_t(\psi)}$, and the dimensionalities of the general expansion covariance matrices, D and C, and the general masking matrix M are much smaller than those obtained from using $2^{\dim \breve{v}_t}$-dimensional dendritic expansions of v_t. Therefore, the two difficulties with a single dendritic encoder with a $2^{\dim \breve{v}_t}$-dimensional output vector are alleviated by the use of multiple dendritic encoders in a PU. A third advantage of multiple dendritic encoders in a PU is the enhancement of generalization capability: if the output vector from one dendritic encoder fails to retrieve any useful information from the general expansion matrices, those from other dendritic encoders may still retrieve enough information for detecting and recognizing the feature subvector v_t input to the PU.

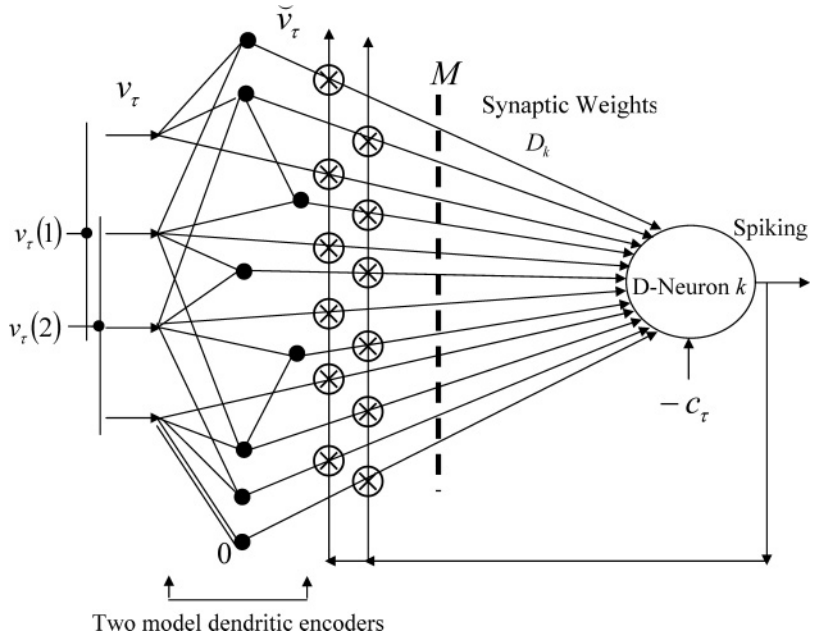

Figure 11: Two model dendritic encoders input subvectors, $v_\tau(1)$ and $v_\tau(2)$, of the feature subvector v_τ are shown. Their inputs are overlapped. Multiple model dendritic encoders are used to encode one input vector v_τ to reduce the number of synapses required and increase the generalization capability.

A D-neuron, D-neuron k, with its masking matrix M, synapses, two dendritic encoders, and unsupervised covariance learning mechanism in a PU, is shown in Figure 11. The two dendritic encoders input subvectors, $v_\tau(1)$ and $v_\tau(2)$, of the feature subvector v_τ input to the PU containing the D-neuron k.

By the information retrieval formula, equations 5.6 and 5.7, applied to each of the Ψ dendritic encoder and its synapses, upon the arrival of $\breve{v}_\tau$, the following products, $d_\tau(\psi)$ and $c_\tau(\psi)$, $\psi = 1, \ldots, \Psi$, are obtained for $\psi = 1, \ldots, \Psi$:

$$d_\tau(\psi) = D(\psi)\, M(\psi) \left(\breve{v}_\tau(\psi) - \left\langle \breve{v}_\tau(\psi) \right\rangle\right), \tag{6.6}$$

$$c_\tau(\psi) = C(\psi)\, M(\psi) \left(\breve{v}_\tau(\psi) - \left\langle \breve{v}_\tau(\psi) \right\rangle\right), \tag{6.7}$$

which are a vector with R components and a scalar, respectively.

For each $\psi = 1, \ldots, \Psi$, the ratio $a_\tau(\psi)/c_\tau(\psi) = (d_\tau(\psi)/c_\tau(\psi) + 1)/2$ is an estimate of the subjective conditional probability that the label $r_\tau(\psi)$ of $v_t(\psi)$ is equal to 1 given $v_t(\psi)$. If $v_t(\psi)$, $\psi = 1, \ldots, \Psi$, share the same label r_τ, that

is the label of v_t (i.e., $r_\tau(\psi) = r_\tau$ for $\psi = 1, \ldots, \Psi$); then the best estimate of the subjective conditional probability that the label r_τ of v_t is equal to 1 given $v_t(\psi)$, $\psi = 1, \ldots, \Psi$, is

$$\left(\sum_{\psi=1}^{\Psi} a_\tau(\psi)\right) \Big/ \left(\sum_{\psi=1}^{\Psi} c_\tau(\psi)\right)$$
$$= \left(\left(\sum_{\psi=1}^{\Psi} d_\tau(\psi)\right) \Big/ \left(\sum_{\psi=1}^{\Psi} c_\tau(\psi)\right) + \mathbf{I}\right)/2,$$

or, equivalently,

$$a_\tau / c_\tau = (d_\tau / c_\tau + \mathbf{I})/2,$$

where

$$d_\tau = DM(\check{v}_\tau - \langle \check{v}_\tau \rangle), \tag{6.8}$$
$$c_\tau = CM(\check{v}_\tau - \langle \check{v}_\tau \rangle), \tag{6.9}$$
$$a_\tau = d_\tau + c_\tau \mathbf{I}, \tag{6.10}$$
$$\mathbf{I} = \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}'. \tag{6.11}$$

The first two formulas here resemble those in equations 5.6 and 5.7. However, D, C, M, $\check{v}_\tau$, and $\langle \check{v}_\tau \rangle$ are the general versions assembled in equations 6.4, 6.3 and 6.5.

The R D-neurons and the C-neuron, their Ψ common dendritic encoders with $\sum_{\psi=1}^{\Psi} 2^{\dim v_t(\psi)}$ synapses, the learning mechanisms with the three training rules, and the retrieving mechanism form a PU. A flowchart of a PU is shown in Figure 12. At time or numbering τ, the PU receives a feature subvector v_τ. The dendritic encoders in the PU encode v_τ into $\check{v}_\tau$ by equation 6.1. The synapses compute and output $c_{\tau j} = C_j M_{jj}(\check{v}_{\tau j} - \langle \check{v}_{\tau j} \rangle)$ and $d_{\tau kj} = D_{kj} M_{jj}(\check{v}_{\tau j} - \langle \check{v}_{\tau j} \rangle)$ for all k and j, or equivalently the vector $[c_{\tau j}]$ and matrix $[d_{\tau kj}]$, where C, D, and M are those in equations 6.4, 6.3, and 6.5, respectively. The C-neuron sums up $c_{\tau j}$ over all j to form $c_\tau = \sum_j c_{\tau j}$ in equation 6.9. D-neuron k sums up $d_{\tau kj}$ over all j to form $d_{\tau k} = \sum_j d_{\tau kj}$, computes the subjective probability distribution $p_{\tau k} = (y_{\tau k} + 1)/2$, where $y_{\tau k} = d_{\tau k}/c_\tau$, and generates a pseudo-random vector $v\{y_{\tau k}\}$, whose components are 1's (spikes) or 0's (nonspikes). In supervised learning, the teaching signal r_τ is provided from outside the PU, and the selection lever is set in the top position. In unsupervised learning, the selection lever is set in the bottom position to use $v\{y_\tau\}$ as the teaching signal r_τ. Using r_τ, the synapses adjuster updates the synaptic weight matrices, C and D, by the three learning

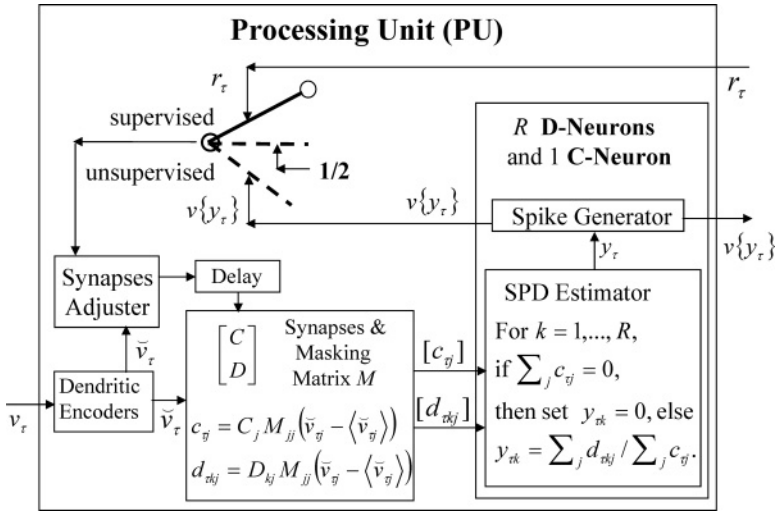

Figure 12: A flowchart of a PU. At τ, the PU receive v_τ. The dendritic encoders encode it into $\breve{v}_\tau$. The synapses compute $c_{\tau j}$ and $d_{\tau kj}$. There are R D-neurons and one C-neuron in the PU. The C-neuron sums up $c_{\tau j}$ to form c_τ. For $k = 1, ..., R$, D-neuron k sums up $d_{\tau kj}$ to form $d_{\tau k}$ and divides $d_{\tau k}$ by c_τ to get $y_{\tau k}$. The vector p_τ with components $p_{\tau k} = (y_{\tau k} + 1)/2$ represents a subjective probability distribution of the label of v_τ. For $k = 1, ..., R$, D-neuron k generates a pseudo-random number $v\{y_{\tau k}\}$. The pseudo-random vector $v\{y_\tau\}$ is a point estimate of the label of v_τ according to the distribution p_τ. Over time, the R components of v_τ form R spike trains. In supervised, unsupervised, or no learning, the selection lever is set in the top, middle, or bottom position, respectively. The synapses adjuster updates the covariance matrices, C and D, according to the unsupervised cumulative, and unsupervised or supervised covariance learning rules.

rules described in section 3 as appropriate. The computations performed by the neurons are described in section 7.

A recurrent multilayer network of PUs is called a low-order model (LOM) of biological neural networks. The vector v_t input to a PU in a layer contains not only feedforward components from outputs of D-neurons in the lower layers but also feedback components from outputs of D-neurons in the same or higher layers. Feedback components are delayed for at least one unit of time to ensure stability. Feedforward components may come from more than one layer preceding the layer in which the PU belongs to.

A possible architecture of LOM comprises an unsupervised main network and supervised offshoot PUs. The unsupervised main network is a multilayer network of PUs with feedback connections, whose PUs all perform unsupervised covariance learning. The supervised offshoot PUs may

or may not form layers, but learn by the supervised covariance rule. An offshoot PU in a layer immediately connecting to a layer of the unsupervised main network receives a certain number of components of the feature vector input to the layer.

An advantage of this architecture is as follows. The unsupervised main network learns all the time regardless of whether signals for supervised learning are available. As soon as a teaching signal is available for supervised learning by an offshoot PU, the relevant information that has been learned and accumulated through unsupervised learning and shows up in a feature vector is learned jointly with the teaching signal in the offshoot PU by the supervised covariance rule. This makes supervised learning extremely effective. For example, assume that a toddler sees many variants of an object without being told the name of the object, and subsequently we point to one such object and tell the toddler the name of it. The child will immediately be able to call out the name of all the variants of the object.

7 Computation by Neurons

A "neuron" in this section consists of the soma of a neuron and the dendrites from the trainable synapses to the soma. Only two types of model neuron are used in LOM: model spiking neurons and model nonspiking neurons. The former is called D-neurons and the latter C-neurons. In each PU, there are R D-neurons and 1 C-neuron. The output of the C-neuron is inhibitory and transmitted to the R D-neurons in the same PU to modulate the processing of the D-neurons.

7.1 Nonspiking Neurons. The C-neuron in a PU receives signals from $\sum_{\psi=1}^{\Psi} 2^{\dim v_t(\psi)}$ synapses connected to the Ψ dendritic encoders in the PU. The strengths of these synapses form the expansion covariance vector C. In response to v_τ, the C-neuron performs addition of the $\sum_{\psi=1}^{\Psi} 2^{\dim v_t(\psi)}$ signals $C_j(\psi)M_{jj}(\psi)(\breve{v}_\tau(\psi) - \langle \breve{v}_\tau(\psi)\rangle)_j$, $j = 1, 2, \ldots, 2^{\dim v_t(\psi)}$, $\psi = 1, 2, \ldots, \Psi$, from the $\sum_{\psi=1}^{\Psi} 2^{\dim v_t(\psi)}$ synapses. The output of the C-neuron is

$$\begin{aligned} -c_\tau &= -\sum_{\psi=1}^{\Psi} C(\psi) M(\psi) \left(\breve{v}_\tau(\psi) - \left\langle \breve{v}_\tau(\psi)\right\rangle\right) \\ &= -CM(\breve{v}_\tau - \langle \breve{v}_\tau\rangle), \end{aligned}$$

which is an estimate of the total number of times v_τ or its variants have been encoded and stored in C with effects of the forgetting factor, normalizing constant, and input corruption, distortion and occlusion included. $-c_\tau$ is a graded inhibitory signal transmitted to the neighboring R D-neurons in the same PU.

7.2 Spiking Neurons. Like the C-neuron, each of the R D-neurons in the PU receives signals from $\sum_{\psi=1}^{\Psi} 2^{\dim v_t(\psi)}$ trainable synapses connected to the dendritic encoders in the PU. The entries of the kth row D_k of D are the strengths of these synapses for D-neuron j for $j = 1, \ldots, R$. In response to v_τ, D-neuron k performs addition of the $\sum_{\psi=1}^{\Psi} 2^{\dim v_t(\psi)}$ signals $D_{kj}(\psi)M_{jj}(\psi)(\check{v}_\tau(\psi) - \langle\check{v}_\tau(\psi)\rangle)_j$, $j = 1, 2, \ldots, 2^{\dim v_t(\psi)}$, $\psi = 1, 2, \ldots, \Psi$, from the $\sum_{\psi=1}^{\Psi} 2^{\dim v_t(\psi)}$ synapses. The output of D-neuron k is

$$\begin{aligned} d_{\tau k} &= \sum_{\psi=1}^{\Psi} D_k(\psi)\, M(\psi)\left(\check{v}_\tau(\psi) - \left\langle\check{v}_\tau(\psi)\right\rangle\right) \\ &= D_k M(\check{v}_\tau - \langle\check{v}_\tau\rangle), \end{aligned}$$

which is an estimate of the total number of times v_τ and its variants have been encoded and stored in D with the kth component $r_{\tau k}$ of r_τ being $+1$ minus the total number of times v_τ and its variants have been encoded and stored in D with the kth component $r_{\tau j}$ being -1. Included in this estimate are the effects of the forgetting factor, normalizing constant, and input corruption, distortion, and occlusion.

Therefore, $(c_\tau + d_{\tau k})/2$ is an estimate of the total number of times v_τ and its variants have been encoded and stored in C with the kth component $r_{\tau k}$ of r_τ being 1. Consequently, $(d_{\tau k}/c_\tau + 1)/2$ is the subjective probability $p_{\tau k}$ that $r_{\tau k}$ is equal to 1 given v_t. D-neuron k then uses a pseudo-random number generator to generate a spike with probability $p_{\tau k}$ and no spike with probability $1- p_{\tau k}$. This 1 or 0 is the output $u_{\tau k}$ of D-neuron k at time or numbering τ. $u_{\tau k}$ is thus a point estimate of the kth component $r_{\tau k}$ of the label r_τ of v_τ.

Note that the vector

$$p_\tau = \begin{bmatrix} p_{\tau 1} & p_{\tau 2} & \cdots & p_{\tau R} \end{bmatrix}'$$

is a representation of a subjective probability distribution of the label r_τ. Note also that the outputs of the R D-neurons in response to v_τ form a binary vector u_τ, which is a point estimate of the label r_τ of v_τ.

The D-neuron is a low-order model of spiking neurons that performs three operations: (1) summing the synaptic outputs to obtain $d_{\tau k}$, (2) dividing c_τ of the inhibitory graded signal $-c_\tau$ from the C-neuron into $d_{\tau k}$ to obtain $d_{\tau k}/c_\tau$ and evaluate the subjective probability $p_{\tau k}$, and (3) generating pseudo-random binary number $u_{\tau k}$ according to the subjective probability distribution $p_{\tau k}$. Operation 1 is the best-known operation of neurons. Division in neurons has been experimentally confirmed and discussed in Tal and Schwartz (1997), Koch (1999), Gabbiani et al. (2002, 2004), and Mel (2008). Koch (1999) lists 5 ways on pages 471-472 for neurons to multiply

(and divide). The division $d_{\tau k}/c_\tau$ in operation 2 is therefore biologically plausible. Spike trains have been modeled as a stochastic point process—namely, the Poisson process, whose firing rate is a random variable with a time-varying average. Pseudo-random binary number generation in operation 3 produces a Bernoulli process whose average firing rate is $p_{\tau k}$, which is a discrete-time approximation of a Poisson process with the same average firing rate. These operations performed by the D-neuron were first proposed in Lo (2008), and the term D-neuron was adopted in Lo (2010). The necessity of using the average firing rate of D-neuron k to carry the subjective probability $p_{\tau k}$ was discussed from the mathematical viewpoint Lo (2010).

A recent paper (London, Roth, Beeren, Hausser, & Latham, 2010) demonstrated the sensitivity of an intact network to perturbation in vivo and concluded from it that the rat barrel cortex "is likely to be using a code that is robust to perturbations such as a rate code in which it is the average firing rate over large populations of neurons that carries information." This conclusion provides a biological confirmation of the assertion in Lo (2010) that information is carried by the average firing rates of spike trains.

London et al. (2010) combined the data (i.e., in vivo whole-cell patch-clamp recordings in rat barrel cortex) from the different current amplitudes and plotted the probability of an extra spike within 5 ms of the current pulse versus total injected charge into a soma in its Figure 3b, and discovered these two variables are approximately linearly related. As either $d_{\tau k}/c_\tau$ or $p_{\tau k}$ is the total injected charge into D-neuron k, this discovery provides a biological support for using the total injected charge, $d_{\tau k}/c_\tau$ or $p_{\tau k}$, in step 3 as the probability for generating a spike.

The D-neuron is the first model of biological spiking neurons with a pseudo-random number generator (PRNG). Apparently a biological spiking neuron does not come with a PRNG. Although it makes the relation between the total injected charge in the D-neuron and the probability of generating a spike linear and makes the spike train a Bernoulli sequence, questions arise: Is the D-neuron with a PRNG biologically meaningful? Does the D-neuron reflect the essence of biological spiking neurons? Is the biological spiking neuron a deterministic device (like a PRNG) whose output spike trains appear like a Poisson process (or Bernoulli sequence)? And so on.

To answer these questions, let us first note that there are roughly two types of models: deterministic and stochastic. The purpose of a deterministic model is to predict the consequence under given conditions. Newton's laws and Einstein's relativity theory are about deterministic models. To check a deterministic model, one compares the prediction by the model with experimental results. The purpose of a stochastic model is to reproduce statistics of a phenomenon. It is usually used when it is too expensive or too hard to develop or use a deterministic model. In this case, a stochastic model is a low-order model.

Let us consider the following example. If we flip a coin (possibly loaded) repeatedly, the outcome is usually stochastically modeled as a Bernoulli sequence. There is not a PRNG or even randomness in the process of repeatedly flipping a coin. The dynamical evolution of the flipped coin is governed by Newton's laws. In theory, each flipping of the coin is a deterministic process, and the outcome is deterministically and perfectly predictable. A deterministic model of coin flipping should describe how your hand releases the coin, how the coin flies through and interacts with the viscous and compressible air, and how it lands, bounces, and comes to rest on the table with certain frictional and elastic coefficients. Such a deterministic model is possible with enough X-ray, MRI, or optical video cameras and computing power. However, the Bernoulli model is preferable for understanding the process of repeated coin flipping and answering many questions. For example, what is the probability that more than 1000 heads show up in 10,000 trials, and what is the expected value for a game based on coin flipping? To simulate coin flipping using the Bernoulli model, a PRNG is used. The outcome of the simulation cannot be used to predict that of repeated coin flipping. Nevertheless, the simulation with a PRNG can be used to provide evidence for the law of large numbers or the central limit theorem.

In fact, all the physical (including biological) phenomena are deterministic, and there are no truly random events except in the quantum world. We use a stochastic model not because the phenomenon involved is not deterministic. What are usually called random events or pure noises are simply events and quantities that are too hard or expensive to model or monitor or do not contain information that can improve what one does or understands. In this letter, however, we will continue using such ordinary words as *random*, and *noise* as they are widely understood and accepted.

Biological experiments found substantial variability in neural activity in the sense that identical sensory stimuli produce different responses (Tolhurst, Movshon, & Dean, 1983; Richmond, Optican, Podell, & Spitzer, 1987; Victor & Purpura, 1996). London et al. (2010) pointed out that there are two possible sources for it. One is the variability usually called truly random events, such as ion channel noise and stochastic synaptic release. This is intrinsic noise: *intrinsic* because it cannot be eliminated and *noise* because it contributes to the neuronal variability but carries no information whatsoever. The experimental results in London et al. (2010) indicate a large amount of intrinsic noise, such as ion channel noise and stochastic synaptic release, in cortex, and this noise puts severe constraints on spike timing codes. The other source of variability is activity from other brain areas. That activity might provide information about, say, the degree of arousal or some other internal state, but it would not be related to the stimulus. This variability is signal, even though it would look like noise to an observer trying to relate the neural activity to the stimulus. Since in our modeling of the biological neural networks by LOM we do not consider activity from

other brain areas, this second type of variability is regarded as noise. (For details of this paragraph, see London et al., 2010.)

This noise from other brain areas and the intrinsic noise are an intrinsic property of biological neural networks and of biological spiking neurons in particular, which causes the input-output variability of biological spiking neurons. To model this intrinsic property, a PRNG is needed in a stochastic model of biological spiking neurons. The PRNG in the D-neuron generates a Bernoulli sequence that emulates the spike train output from a biological spiking neuron. As the same sequence input to a biological spiking neuron twice does not elicit the same output sequences, we do not require the D-neuron to generate the same output sequence that the biological spiking neuron does in response to the same input sequence. Incidentally, circumstantial evidence to support the use of a PRNG in the D-neuron is that the Bernoulli sequences generated by D-neurons play a pivotal role in the operations of LOM. An example is the creation of a vocabulary by the unsupervised covariance rule of learning described in sections 3.1 and 7.3.

Recent studies observed that precisely timed input to dendritic branches can yield precisely timed output spikes in the axon without large somatic subthreshold voltage excursions (Larkum, Zhu, & Sakmann, 1999; Ariav, Polsky, & Schiller, 2003; Hausser & Mel, 2003; London & Hausser, 2005). However, as pointed out in London et al. (2010), "These mechanisms, though, have only been demonstrated *in vitro*, and only when input to the dendrites was carefully regulated in both space and time." Nevertheless, before the observation is clarified, the possibility is there that the relation between the input and output of a biological neuron is deterministic under certain circumstances. (Recall that all phenomena are deterministic except in the quantum world.) Besides, theoretical work has proposed that biological neural networks are chaotic (van Vreeswijk & Sompolinsky, 1998; Banerjee, Seriès, & Pouget, 2008; Izhikevich & Edelman, 2008). Chaos is deterministic, but unpredictable and statistically indistinguishable from randomness. Although the observation made in vitro in Larkum et al. (1999), Ariav et al. (2003), Hausser and Mel (2003), London and Hausser (2005) and chaotic neural network work van Vreeswijk and Sompolinsky (1998), Banerjee et al. (2008), and Izhikevich and Edelman (2008) suggest that biological neural networks are deterministic, no adequate deterministic model of biological spiking neurons has been reported. Presumably, in the in vivo study reported in London et al. (2010), measurements were taken while the deterministic behaviors were in progress and thus included. Analogous to the stochastic model of the basically deterministic process of repeated coin flipping discussed earlier on, the D-neuron using a PRNG to generate a Bernoulli sequence is a low-order stochastic model that we prefer at the present time, especially considering of the pivotal role it plays in the operations of LOM.

7.3 Creating a Vocabulary by Unsupervised Covariance Learning. Pseudo-random binary number generation performed by the R D-neurons in a PU is indispensable in making the unsupervised covariance learning rule work for the PU. Let us now see how a "vocabulary" is created by unsupervised covariance learning rule for the PU. If a feature subvector v_τ or a slightly different version of it, has not been learned by the PU, and $CM\check{v}_\tau = 0$, then d_τ/c_τ is set equal to 0 and $p_\tau = (1/2)\mathbf{I}$, where $\mathbf{I} = [1\ 1\ \cdots\ 1]'$. The R D-neurons use this subjective probability vector to generate a purely random label r_τ. Once this r_τ and the output vector $\check{v}_\tau$ have been learned and stored in C and D, if v_τ is input to the PU for a second time, then $u_\tau = r_\tau$ with probability 1, and one more copy of the pair (v_τ, r_τ) is included in C and D.

If an input vector v_τ or a slightly different version of it has been learned by a PU with different labels for different numbers of times, then $y_\tau \neq 0$ and $p_\tau \neq (1/2)\mathbf{I}$. Since v_τ may contain different parts from different causes and are assigned different labels in different rounds of unsupervised learning, p_τ may not be a binary vector. For example, assume that two labels, r_τ^1 and r_τ^2, of the same input vector v_τ have been learned with relative frequencies, 0.7 and 0.3, respectively. Then in response to v_τ, each component of u_τ that is output from the PU is equal to r_τ^1 with probability 0.7 and is equal to r_τ^2 with probability 0.3. Since these two labels may have common components, the point estimate of the label resembles r_τ^1 with a probability of greater than 70% and resembles r_τ^2 with a probability of greater than 30%.

8 Spike Trains for Each Exogenous Feature Vector

Recall that a binary vector u_t output from a PU, is obtained by a pseudo-random binary number generator using the subjective probability distribution p_t (or $y_t = d_t/c_t$) of the label r_t of a vector v_t input to the PU. Components of such binary vectors u_t with uncertainty form vectors input to PUs through feedforward or feedback connections. Upon receiving a vector with uncertainty, a PU uses masking matrices to suppress, or filter out, some components so that the remaining components are consistent with those stored in the expansion covariance matrices. (Masking matrices are described in section 5.)

However, there is a chance for the pseudo-random number generator to generate a binary vector u_t that is such an outlier for the subjective probability distribution p_t (or d_t/c_t) that causes undesirable effects on learning and retrieving of PUs receiving components of u_t in spite of masking matrices. To minimize such undesirable effects and represent the subjective probabilities involved in the PUs, our model, LOM, completes a certain number of rounds of retrieving and learning for each exogenous vector v_t^{ex} input to LOM so that many pseudo-random versions of u_t are generated and learned by each PU for the same v_t^{ex}.

To have ζ rounds of retrieving and learning for each exogenous vector v_t^{ex}, the exogenous vector must be held constant for ζ units of time. In other words, the exogenous vector v_t^{ex} is presented to LOM with a different timescale. More specifically, v_t^{ex} changes at $t = i\zeta + 1$, $i = 0, 1, 2, \ldots$. Consequently, a PU generates a sequence of binary vectors denoted by u_t, $t = i\zeta + j$, $j = 1, 2, \ldots, \zeta$, for each exogenous feature vector v_t^{ex}, which remains constant for $t = i\zeta + j$, $j = 1, 2, \ldots, \zeta$. More specifically, once a new exogenous vector $v_{i\zeta+1}^{ex}$ is presented, $v_{i\zeta+j}^{ex} = v_{i\zeta+1}^{ex}$ for $j = 2, \ldots, \zeta$. The sequence u_t, $t = i\zeta + j$, $j = 1, 2, \ldots, \zeta$, output by the PU consists of R spike trains, each having ζ spikes during the period of time.

9 Multilayer Networks of Processing Units with Feedbacks

The LOM of biological neural networks proposed in this letter is a multilayer network of processing units (PUs) with time-delayed feedbacks. Lo (1996) proved that a discrete-time model of biological neural networks with feedbacks is a valid model if and only if every cycle in the network has a delay device and every valid model is necessarily a layered network. LOM has these properties and can be looked on as an organization of a biological neural network into a multilayer network of PUs with feedback connections from PUs in a layer to PUs in the same or lower layers. The delay devices on the feedback connections to ensure stability of LOM make it a discrete-time dynamical system whose dynamical state at a time consists of the feedbacks held in the delay devices at the time.

An external vector input to LOM is called an exogenous feature vector, and a vector input to a layer of PUs is called a feature vector. A feature vector input to a layer usually contains not only feedforward outputs from the PUs in preceding layers but also feedback outputs from the PUs in the same or higher layers with time delays. A feature vector may contain components from an exogenous feature vector. For simplicity, we assume that the exogenous feature vector is input only to layer 1 and is thus a subvector of a feature vector input to layer 1. All of these feature vectors and output vectors over time usually form spike trains.

A subvector of a feature vector that is input to a PU is called a feature subvector. Trace the feedforward connections backward from neurons of a PU to a subvector of the exogenous feature vector. This subvector is called the receptive domain of the PU (not the receptive field, because the exogenous feature vector is usually not outputs of sensory receptors). The collection of neurons in layer $l - i, i = 1, 2, \ldots$, that have a direct feedforward connection (without going through another neuron) to a neuron in a PU in layer l and the unit time delay devices that hold a feedback that is directly input (without going through another unit time delay device) to the same PU are called the immediate receptive domain of the PU.

The feature vector input to layer l at time or numbering τ is denoted by v_τ^{l-1}, and the output from layer l at τ is denoted by $v\{y_\tau^l\}$. The feature vector

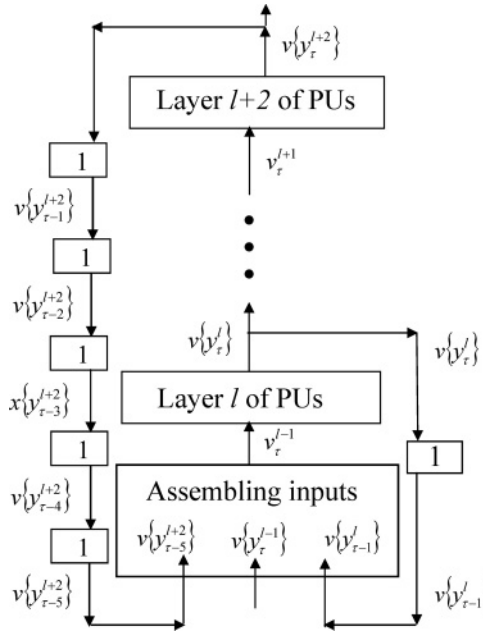

Figure 13: Layers l and $l+2$ of an example LOM with a feedback connection from layer l to layer l and another from layer $l+2$ to layer l. The former contains one unit time delay device and the latter five unit time delay devices. The box under layer l of PUs does not model a biological entity and shows only how both the feedforward and feedback spike trains are assembled into the feature vector input to a layer. Note that feedbacks from layer l and layer $l+2$ are shown here. There can be feedback from other layers. For notational simplicity, we feedback all outputs from layer l and layer $l+2$, but this is not necessary.

v_τ^{l-1} consists of components of the feedforward vector $v\{y_\tau^{l-1}\}$ and feedback vector $v\{y_{\tau-z(k)}^{l+k}\}$ fed back from the same layer l and higher layers $l+k$ and after $z(k)$ time units of delay for $k = 0, 1, \ldots$, where $z(k)$ is a function of k.

Figure 13 shows layer l and layer $l+2$ of PUs of LOM. In Figure 13, same-layer feedback connections with one unit time delay device from layer l to itself and two-layer feedback connections with five unit time delay devices from layer $l+2$ to layer l are shown. The box under layer l of PUs does not model a biological entity but illustrates that the feature vector input to layer l comprises feedforward vector $v\{y_\tau^{l-1}\}$, the same-layer feedback $v\{y_{\tau-1}^{l}\}$, and the two-layer feedback $v\{y_{\tau-5}^{l+2}\}$.

Once an exogenous feature vector is received by LOM, the PUs perform functions of retrieving or learning from layer to layer starting with layer 1, the lowest layer. After the PUs in the highest layer, layer L, complete performing their functions, LOM is said to have completed one round

of retrieving or learning (or both). Each exogenous feature vector is held constant for a certain number ζ of time units, during which LOM completes ζ of retrieving or learning (or both).

We note that retrieving and learning by a PU are performed locally, meaning that only the feature subvector input to the PU and its label are involved in the processing by the PU. Causes in patterns, temporal or spatial, usually form a hierarchy—for example:

- Phonemes, words, phrases, sentences, and paragraphs in speech
- Musical notes, intervals, melodic phrases, and songs in music
- Bananas, apples, peaches, salt shaker, pepper shaker, Tabasco, fruit basket, condiment tray, table, refrigerator, water sink, and kitchen in a house.

Note that although the final example is a spatial hierarchy, when one looks around in the kitchen, the images scanned and received by the person's retina form a temporal hierarchy.

The higher a layer in LOM is, the higher in the hierarchy the causes the PUs in the layer handle, and the more time it takes for the causes to form and be detected and recognized by the PUs. Therefore, the number of unit time delay devices on a feedback connection is a monotone increasing function $z(k)$ of k, which are defined above. This requirement is consistent with the workings in a biological neural network in the cortex. Note that it takes time (1) for biological PUs to process feature subvector, (2) for spikes to travel along feedforward neural fibers from a layer to the next layer, and (3) for spikes to travel along feedback neural fibers from a layer to the same or a lower-numbered layer. Note also that the subscripts of the input vector v_τ^{l-1} and output vector $v\{y_\tau^l\}$ of all layers l are the same, indicating the same exogenous feature vector v_τ^{ex} is processed or propagated in all layers. The common subscript τ does not represent the time that the signals in the biological network reach or are processed by its layers. However, a feedback $v_{\tau-z(k)}^{l+k}$ from layer $l+k$ to layer l for inclusion in v_τ^{l-1} must have a delay $z(k)$ that reflects the sum of the times taken for processing and traveling from the input terminals of layer l back to the same input terminals.

Note that the traveling depends not only on the total length of the feedforward and feedback connections, but also the propagation speed of action potentials in them. Spikes travel along a nerve fiber at a speed of anywhere from 1 to 120 meters per second, depending on the diameter of a fiber, the presence or absence of myelin in the fiber, and the geometrical ratio (Manor, Koch, & Segev, 1991; Debanne, 2004). In general, the greater the diameter or the more myelin, the faster the spikes travel. LOM is driven by the exogenous feature vectors. The time unit for LOM is the spike duration, including the refractory period, and the interspike interval (ISI) between a spike and a following spike that is immediately fired right after the refractory period of the former in one of the spike trains whose snapshots are the

exogenous feature vectors to LOM. For presentational simplicity, the spike duration and propagation speed are assumed to be constant in LOM. The spatial length of a spike in a neural fiber is the product of the spike speed and spike duration. For illustration, an example is given:

Example 4. Let us set the number $z(k)$ of unit time delay devices equal to $4(k+1)$ for $k = 0, \ldots, 7$ and set the number ζ time units that each exogenous feature vector is held constant equal to 16.

For $k = 1$ and $z(1) = 8$, the first 8 feedbacks used by layer l in processing an exogenous feature vector v_t^{ex} are output from layer $l+1$ in response to v_{t-1}^{ex}, which provides temporally and spatially associated information from the preceding exogenous feature vector v_{t-1}^{ex}.

For $k = 5$ and $z(5) = 24$, the first 8 feedbacks used by layer l in processing an exogenous feature vector v_t^{ex} are output from layer $l+5$ in response to v_{t-2}^{ex}, and the next 8 feedbacks are output from layer $l+5$ in response to v_{t-1}^{ex}, which provides temporally and spatially associated information from the preceding exogenous feature vectors, v_{t-2}^{ex} and v_{t-1}^{ex}.

For $k = 8$ and $z(8) = 36$, the first 4 feedbacks used by layer l in processing an exogenous feature vector v_t^{ex} are output from layer $l+8$ in response to v_{t-3}^{ex}, and the next 12 feedbacks are output from layer $l+8$ in response to v_{t-2}^{ex}, which provides temporally and spatially associated information from the preceding exogenous feature vectors, v_{t-3}^{ex} and v_{t-2}^{ex}.

Note that the greater k is, the larger the number of unit delay devices, and the further back the feedback information is in processing the current exogenous feature vector v_t^{ex}. Note also that the further back the feedback information is, the less spatially but more temporally associative information is used in processing v_t^{ex}. Moreover, given the same numbers of unit delay devices on each feedback connection, if an exogenous feature vector is presented to LOM for a larger number of time units, then more recent information and less further back information is used in processing v_t^{ex}. This means more spatially associated information but less temporally associated information is brought back by the feedback connections and utilized by LOM.

An example LOM with three layers of PUs and feedbacks is shown in its entirety in Figure 14. There are three types of feedback connection: same-layer feedbacks, one-layer feedbacks and two-layer feedbacks. The numbers of unit time delay devices on the feedback connections are not specified for simplicity. The second delay box on a feedback connection represents an additional delay.

10 Conclusion

A low-order model (LOM) of biological neural networks is proposed in this letter. It comprises model dendritic nodes, encoders and trees, model synapses capable of learning, model spiking and nonspiking neurons,

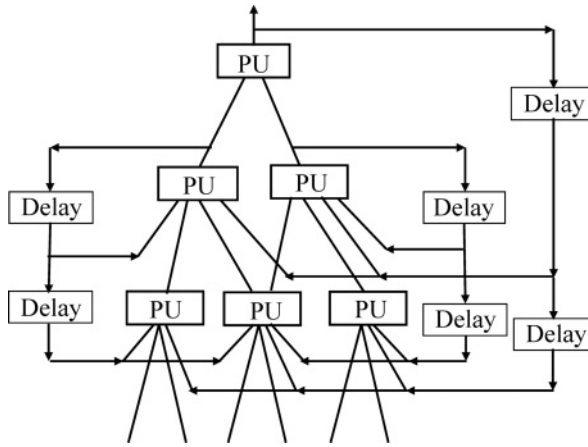

Figure 14: An example LOM with three layers of PUs is shown in its entirety. There are three types of feedback connection: same-layer feedback, one-layer feedback, and two-layer feedback. The delay durations on the feedback connections are not specified. The PU boxes are those shown in Figure 12. The second delay box on a feedback connection represents an additional delay.

model unsupervised and supervised learning mechanisms, model retrieving and generalizing mechanisms, and model feedback nerve fibers with different lengths. These model components are each a structural, functional, and operational model that is biologically plausible.

Consistent with existing understanding of dendritic trees (Koch & Poggio, 1982, 1992; Koch et al., 1983; Rall & Sergev, 1987; Shepherd & Brayton, 1987; Mel, 1992a, 1992b, 1993, 1994, 2008), the model dendritic nodes are each a hyperbolic polynomial in two variables that acts like an XOR logic gate when the inputs are binary digits (Zador et al., 1992; Fromherz & Gaede, 1993). Compositions of such model dendritic nodes form a model dendritic encoder, which is a high-order multivariable polynomial function encoding its inputs. If the inputs are binary digits, the codes output from the model dendritic encoder have an orthogonal property. This property allows the learning and retrieving of subjective probability distributions that are firing rates of spike trains that spiking neurons generate and transmit.

In recent years, the morphology of and information processing in axons have attracted much attention (Wittner et al., 2007; Matsuda et al., 2009; Debanne, 2004). It is conceivable that the network of hyperbolic polynomial nodes with the XOR effect comprises not only dendritic nodes but also axonal nodes.

The model unsupervised and supervised learning mechanisms proposed here are variants of Sejnowski's covariance learning rule (Sejnowski, 1977;

Koch, 1999). Together with a model unsupervised accumulation learning mechanism, they execute unsupervised and supervised learning of the subjective probability distributions. The learned subjective probability distributions, which are actually subjective conditional probability distributions, are stored in model synapses.

The model generalizing mechanism, which is represented by a simple masking matrix, is actually an idealization and organization of a large number of dendritic encoders with overlapped and nested inputs. Although somewhat unsophisticated, the mechanism in its ideal form maximizes generalization capability in retrieving corrupted, distorted, and occluded patterns.

The model nonspiking and spiking neurons jointly retrieve the subjective probability distributions, and the latter further generates stochastic spike trains with the subjective probability distributions as firing rates. The inhibitory graded output of a model nonspiking neuron divides into the sum of the outputs of trainable synapses in the process. This operation was discovered and reported in Tal and Schwartz (1997), Koch (1999), Gabbiani et al. (2002, 2004), and Mel (2008). Its use in computing subjective probability distributions shows a reason that the operation is necessary in neuronal computation.

Modeling nerve fibers of different lengths by including different numbers of unit time delay devices in feedback connections in LOM, instead of a single delay device with the same time delay in each feedback connection, allows LOM to mimic biological neural networks' capabilities to fully utilize spatial and temporal associative informations.

Although LOM was motivated by THPAM, they have the following significant differences:

1. The model dendritic node in THPAM input two ternary digits (i.e., −1, 0, 1) and perform NXOR (NOT-exclusive-OR). How the model dendritic node performs NXOR is unknown and not described in Lo, (2010). In contrast, the four types of model dendritic-axonal node in LOM input two real numbers and evaluate a hyperbolic polynomial. If the real numbers are close to binary digits, the model dendritic/axonal nodes act like an XOR logic gate. Each model dendritic/axonal mode is a mini-network of dendrites/axons and inhibitory/excitatory synapses that evaluate the hyperbolic polynomial.
2. The orthogonal expander in THPAM performs orthogonal expansion of its input ternary vector. The model dendritic/axonal encoders in LOM perform dendritic/axonal encoding of the input real vector. The resultant codes have an orthogonality property suited for the unsupervised and supervised covariance learning rules herein propos. A mathematical proof of the orthogonality property is provided in the appendix.

3. The learning rules in LOM are variants of Sejnowski's covariance rule, while those in THPAM are not.
4. Information retrieval methods of LOM and THPAM are different.

As mentioned in section 1, although the biologically plausible component models are worth looking at in their own right, a main contribution of this letter is the integration of these component models into a logically coherent and biologically plausible description of biological neural networks. The integration is shown in Figures 12 and 13. More specifically, (1) the model dendritic encoders provide appropriate inputs to the model synapses for learning subjective probability distributions with or without supervision; (2) the model unsupervised and supervised covariance learning mechanisms effect such learning of subjective probability distributions; (3) the masking matrix allows model spiking and nonspiking neurons to generalize maximally on inputs to the processing units; (4) the two types of neuron use signals from synapses to compute subjective probability distributions and generate spikes accordingly; and (5) the generated spikes are new labels or estimates of old labels for use by the unsupervised learning mechanism and for transmitting information to other dendritic trees, synapses and neurons.

Another main contribution is logically coherent answers to the eight questions posed at the end of section 1 provided by a single biologically plausible model, LOM.

The work reported in this letter points to four research directions for the near future:

1. Further examine the components and processing operations of LOM as biological hypotheses. Confirm those that are faithful low-order descriptions of the corresponding components and processing operations of biological neural networks and modify or dismiss those that are not.
2. Find biologically plausible models of more anatomical features and functional capabilities of biological neural networks such as attention selection and motion detection in order to further develop LOM into a model of a higher order.
3. Expand and modify LOM into low-order models of the visual, auditory, somatosensory, and (premotor, primary and supplementary) motor cortices.
4. Test and apply LOM to such applications as handwriting recognition, speech recognition, face detection and recognition, radiograph reading, baggage and container examination, license plate recognition, automatic target recognition, satellite and internet mass data processing, video monitoring, text understanding, prostheses, time-series prediction, and others.

Appendix: Proof of an Orthogonality Property of Dentritic Codes

A proof of the formulas stated at the end of section 2.3 is provided here. Given the hypothesized binary function $\phi(v, u) = -2vu + v + u$ of the dendritic node, it is easy to see that

$$\phi(v, u) - 1/2 = -2(v - 1/2)(u - 1/2).$$

Recall the notation

$$\phi\left(v, \begin{bmatrix} u_1 & \cdots & u_k \end{bmatrix}'\right) = \begin{bmatrix} \phi(v, u_1) \cdots \phi(v, u_k) \end{bmatrix}'.$$

Simple calculation yields, for $k = 1, 2, \ldots$,

$$\begin{aligned}
&\phi\left(v, \begin{bmatrix} u_1 & \cdots & u_k \end{bmatrix}'\right) - \frac{1}{2}\mathbf{I} \\
&= \begin{bmatrix} \phi(v, u_1) - \frac{1}{2} & \cdots & \phi(v, u_k) - \frac{1}{2} \end{bmatrix}' \\
&= \begin{bmatrix} -2\left(v - \frac{1}{2}\right)\left(u_1 - \frac{1}{2}\right) & \cdots & -2\left(v - \frac{1}{2}\right)\left(u_k - \frac{1}{2}\right) \end{bmatrix}' \\
&= -2\left(v - \frac{1}{2}\right)\begin{bmatrix} u_1 - \frac{1}{2} & \cdots & u_k - \frac{1}{2} \end{bmatrix}' \\
&= -2\left(v - \frac{1}{2}\right)\left(\begin{bmatrix} u_1 & \cdots & u_k \end{bmatrix}' - \frac{1}{2}\mathbf{I}\right)
\end{aligned} \tag{A.1}$$

where $\mathbf{I} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}'$.
Recall that

$$\breve{v}(1) = \begin{bmatrix} 0 & v_1 \end{bmatrix}',$$

$$\breve{v}(1, \ldots, k+1) = \begin{bmatrix} \breve{v}'(1, \ldots, k) & \phi(v_{k+1}, \breve{v}'(1, \ldots, k)) \end{bmatrix}'$$

for $k = 1, 2, \ldots$. By equation A.1,

$$\begin{aligned}
&\breve{v}(1, \ldots, k+1) - \frac{1}{2}\mathbf{I} \\
&= \begin{bmatrix} \breve{v}'(1, \ldots, k) - \frac{1}{2}\mathbf{I}' & \phi(v_{k+1}, \breve{v}'(1, \ldots, k)) - \frac{1}{2}\mathbf{I}' \end{bmatrix}' \\
&= \begin{bmatrix} \breve{v}'(1, \ldots, k) - \frac{1}{2}\mathbf{I}' & -2\left(v_{k+1} - \frac{1}{2}\right)\left(\breve{v}'(1, \ldots, k) - \frac{1}{2}\mathbf{I}'\right) \end{bmatrix}'.
\end{aligned}$$

By simple substitution and factorization, we obtain for $k = 1, 2, \ldots$,

$$\begin{aligned}
&\left(\breve{v}\,(1,\ldots,k+1)-\frac{1}{2}\mathbf{I}\right)'\left(\breve{u}\,(1,\ldots,k+1)-\frac{1}{2}\mathbf{I}\right)\\
&=\left(\breve{v}\,(1,\ldots,k)-\frac{1}{2}\mathbf{I}\right)'\left(\breve{u}\,(1,\ldots,k)-\frac{1}{2}\mathbf{I}\right)+\\
&2^2\left(v_{k+1}-\frac{1}{2}\right)\left(u_{k+1}-\frac{1}{2}\right)\left(\breve{v}\,(1,\ldots,k)-\frac{1}{2}\mathbf{I}\right)'\left(\breve{u}\,(1,\ldots,k)-\frac{1}{2}\mathbf{I}\right)\\
&=\left(1+2^2\left(v_{k+1}-\frac{1}{2}\right)\left(u_{k+1}-\frac{1}{2}\right)\right)\left(\breve{v}\,(1,\ldots,k)-\frac{1}{2}\mathbf{I}\right)'\\
&\quad\times\left(\breve{u}\,(1,\ldots,k)-\frac{1}{2}\mathbf{I}\right). \qquad \text{(A.2)}
\end{aligned}$$

For m-dimensional vectors, v and u, applying equation A.2 repeatedly for $k = m, m-1, \ldots, 1$ yields

$$\begin{aligned}
&\left(\breve{v}-\frac{1}{2}\mathbf{I}\right)'\left(\breve{u}-\frac{1}{2}\mathbf{I}\right)\\
&=\left[\prod_{k=2}^{m}\left(1+2^2\left(v_k-\frac{1}{2}\right)\left(u_k-\frac{1}{2}\right)\right)\right]\left(\breve{v}\,(1)-\frac{1}{2}\mathbf{I}\right)'\left(\breve{u}\,(1)-\frac{1}{2}\mathbf{I}\right)\\
&=2^{-2}\prod_{k=1}^{m}\left(1+2^2\left(v_k-\frac{1}{2}\right)\left(u_k-\frac{1}{2}\right)\right) \qquad \text{(A.3)}
\end{aligned}$$

because $(\breve{v}(1)-\frac{1}{2}\mathbf{I})'(\breve{u}(1)-\frac{1}{2}\mathbf{I}) = 2^{-2}(1+2^2(v_1-\frac{1}{2})(u_1-\frac{1}{2}))$.

Note that $1+2^2(v_k-\frac{1}{2})(u_k-\frac{1}{2}) = 2$ if $(v_k, u_k) = (0,0)$ or $(1,1)$, and $1+2^2(v_k-\frac{1}{2})(u_k-\frac{1}{2}) = 0$ if $(v_k, u_k) = (1,0)$ or $(0,1)$. Therefore, if v and u are binary vectors, we have the orthogonality property, equations 2.6 and 2.7:

$$\begin{aligned}
\left(\breve{v}-\frac{1}{2}\mathbf{I}\right)'\left(\breve{u}-\frac{1}{2}\mathbf{I}\right) &= 2^{m-2}, \text{ if } u = v\\
&= 0, \text{ if } u \neq v.
\end{aligned}$$

Remark. If any component of $\breve{v}$, say $\breve{v}_q$ is replaced with $-\breve{v}_q$, the corresponding component $\breve{u}_q$ of $\breve{u}$ is replaced with $-\breve{u}_q$, and the corresponding component $\mathbf{I}_q$ with -1 then the above orthogonality property, equations 2.6 and 2.7, remains true. In fact, if any number of components in $\breve{v}$ change their signs and the corresponding components in $\breve{u}$ and $\mathbf{I}$ change their signs, then the orthogonality property still holds.

Acknowledgments

Thanks to Paul Werbos, a program manager in NSF; Jonathan Bell, a neuromathematician in the Department of Mathematics and Statistics; and Frank Hanson and Thomas Cronin, neuroscientists in the Department of Biological Sciences, University of Maryland Baltimore County, for helpful discussions. I also thank two anonymous reviewers for their comments, which led to improvements of the manuscript.

References

Amari, S. (1989). Characteristics of sparsely encoded associative memory. *Neural Networks 2*(6), 451–457.

Antic, S. D., Zhou, W.-L., Moore, A. R., Short, S. M., & Ikonomu, K. D. (2010). The decade of the dendritic NMDA spike. *Journal of Neuroscience Research, 88*, 2991–3001.

Ariav, G., Polsky, A., & Schiller, J. (2003). Submillisecond precision of the input-output transformation function mediated by fast sodium dendritic spikes in basal dendrites of CA1 pyramidal neurons. *Journal of Neuroscience, 23*, 7750–7758.

Banerjee, A., Seriès, P., & Pouget, A. (2008). Dynamical constraints on using precise spike timing to compute in recurrent cortical networks. *Neural Computation, 20*, 974–993.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer.

Dayan, P., & Abbott, L. F. (2001). *Theoretical neuroscience: Computational and mathematical modeling of neural systems*. Cambridge, MA: MIT Press.

Debanne, D. (2004). Information processing in the axon. *Nature Reviews Neuroscience, 5*, 304–316.

Fox, C. A., & Barnard, J. W. (1957). A quantitative study of the Purkinje cell dendritic branches and their relationship to afferent fibers. *Journal of Anatomy, 91*, 299–313.

Fromherz, P., & Gaede, V. (1993). Exclusive-OR function of single arborized neuron. *Biological Cybernetics, 69*, 337–344.

Gabbiani, F., Krapp, H. G., Hatsopoulos, N., Mo, C.-H., Koch, C., & Laurent, G. (2004). Multiplication and stimulus invariance in a looming-sensitive neuron. *Journal of Physiology, 98*, 19–34.

Gabbiani, F., Krapp, H. G., Koch, C., & Laurent, G. (2002). Multiplication computation in a visual neuron sensitive to looming. *Nature, 420*, 320–324.

George, D., & Hawkins, J. (2009). Towards a mathematical theory of cortical micro-circuits. *PLoS Computational Biology, 5*, 1–26.

Gerstner, W., & Kistler, W. M. (2002). *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge: Cambridge University Press.

Goldman, M. S., Levine, J. H., Major, G., Tank, & Seung. (2003). Robust persistent neural activity in a model integrator with multiple hysteretcic dendrites per neuron. *Cerebral Cortex, 13*, 1185–1195.

Granger, R. (2006). Engines of the brain: The computational instruction set of human cognition. *AI Magazine, 27*, 15–31.

Grossberg, S. (2007). Towards a unified theory of neocortex: Laminar cortical circuits for vision and cognition. *Progress in Brain Research, 165*, 79–104.

Hassoun, M. H. (1993). *Associative neural memories: Theory and implementation*. New York: Oxford University Press.

Hausser, M., & Mel, B. (2003). Dendrites: Bug or feature? *Current Opinions on Neurobiology, 13*, 372–383.

Hawkins, J. (2004). *On intelligence*. New York: Holt.

Haykin, S. (2009). *Neural networks and learning machine* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

Hecht-Nielsen, R. (1990). *Neurocomputing*. New York: Addison-Wesley.

Hecht-Nielsen, R. (2007). *Confabulation theory*. New York: Springer-Verlag.

Hecht-Nielsen, R., & McKenna, T. (2003). *Computational models for neuroscience*. New York: Springer-Verlag.

Hinton, G. E., & Anderson, J. A. (1989). *Parallel models of associative memory*. Mahwah, NJ: Erlbaum.

Izhikevich, E. M., & Edelman, G. M. (2008). Large-scale model of mammalian thalamocortical systems. *Proceedings of National Academy of Sciences, 105*, 3593–3598.

Kalman, R. E., Falb, P., & Arbib, M. A. 1969. *Topics in mathematical system theory*. New York: McGraw-Hill.

Koch, C. (1999). *Biophysics of computation*. New York: Oxford University Press.

Koch, C., & Poggio, T. (1982). Retina ganglion cells: A functional interpretation of dendritic morphology. *Philos. Trans. R. Soc. Lond. B Biol. Sci., 298*(1090), 227–263.

Koch, C., & Poggio, T. (1992). Multiplying with synapses and neurons. In T. McKenna, J. Davis, & S. F. Zornetzer (Eds.), *Single neuron computation*. Orlando, FL: Academic Press.

Koch, C., Poggio T, & Torre V. (1983). Nonlinear interactions in a dendritic tree: Localization, timing, and role in information processing. *Proceedings of National Academy of Sciences, U.S.A., 80*(9), 2799–2802.

Kohonen, T. (1988a). *Self-organization and associative memory*. New York: Springer-Verlag.

Kohonen, T. (1988b.) *Self-organization and associative memory* (2nd ed.). New York: Springer-Verlag.

Landau, L. D., & Lifshitz, E. M. (1987). *Fluid mechanics*. New York: Pergamon Press.

Larkum, M. E., Zhu, J. J., & Sakmann, B. (1999). A new cellular mechanism for coupling inputs arriving at different cortical layers. *Nature, 398*, 338–341.

Llinas, R., & Sugimori, M. (1980). Electrophysiological properties of in vitro Purkinje cell somata in mammalian cerebellar slices. *Journal of Physiology, 305*, 171–195.

Lo, J. T.-H. (1996). Layer and recursive structures of neural networks. In *Proceedings of the 1996 World Congress on Neural Networks*. Piscataway, NJ: IEEE Press.

Lo, J. T.-H. (2008). Probabilistic associative memories. In *Proceedings of the 2008 International Joint Conference on Neural Networks* (pp. 3895–3903). Piscataway, NJ: IEEE Press.

Lo, J. T.-H. (2010). Functional model of biological neural networks. *Cognitive Neurodynamics, 4*, 295–313.

London, M., & Hausser, M. (2005). Dendritic computation. *Annual Review of Neuroscience, 28*, 503–532.

London, M., Roth, A., Beeren, L., Hausser, M., & Latham, P. E. (2010). Sensitivity to perturbations in vivo implies high noise and suggests rate coding in cortex. *Nature, 466*, 123–128.

Maass, W., & Bishop, C. M. (1998). *Pulsed neural networks*. Cambridge, MA: MIT Press.

Manor, Y., Koch, C., & Segev, I. (1991). Effect of geometrical irregularities on propagation delay in axonal trees. *Biophysical Journal, 60*, 1424–1437.

Martin, K. A. C. (2002). Microcircuits in visual cortex. *Current Opinion in Neurobiology, 12*, 418–442.

Matsuda, N., Lu, H., Fukata, Y., Noritake, J., Gao, H., Mukherjee, S., et al. (2009). Differential activity-dependent secretion of brain-derived neurotrophic factor from axon and dendrite. *Journal of Neuroscience, 29*, 14185–14198.

Mel, B. W. (1992a). The clusteron: toward a simple abstraction in a complex neuron. In J. Moody, S. Hanson, & R. Lippmann, R. (Eds.), *Advances in neural information processing systems*, 4 (pp. 35–42). San Mateo, CA: Morgan Kauffmann.

Mel, B. W. (1992b). NMDA-based pattern discrimination in a modeled cortical neuron. *Neural Computation, 4*, 502–516.

Mel, B. W. (1993). Synaptic integration in an excitable dendritic tree. *Journal of Neurophysiology, 70*, 1086–1101.

Mel, B. W. (1994). Information processing in dendritic trees. *Neural Computation, 6*, 1031–1085.

Mel, B. W. (2008). Why have dendrites? A computational perspective. In G. Stuart, N. Spruston & M. Hausser (Eds.), *Dendrites* (2nd ed.). New York: Oxford University Press.

Milokhin, A. A., & Reshetnikov, S. S. (1968). Axo-axonal synapses in the ganglia of Auerbach's plexus. *Arkhiv Anatomii, Gistologii i Embriologii, 54*, 25–30.

Morita, K. (2008). Possible role of dendritic compartmentalization in the spatial working memory circuit. *Journal of Neuroscience, 28*, 7699–7724.

Morita, K. (2009). Possible dendritic contribution to unimodal numerosity tuning and Weber-Fechner law-dependent numerical cognition. *Frontiers in Computational Neuroscience, 3*, 1–14.

Morita, K., Okada, M., & Aihara, K. (2007). Selectivity and stability via dendritic nonlinearity. *Neural Computation, 19*, 1798–1853.

Nagano, K. (1972). Association—a model of associative memory. *IEEE Transactions on Systems, Man and Cybernetics, SMC-2*, 68–70.

O'Reilly, R., & Munakata, Y. (2000). *Computational explorations in cognitive neuroscience*. Cambridge, MA: MIT Press.

Pannese, E. (1994). *Neurocytology: Fine structure of neurons, nerve processes, and neuroglial cells*. New York: Thieme.

Poirazi, P., Brannon, T., & Mel, B. W. (2003). Pyramidal neuron as two-layer neural network. *Neuron, 37*, 989–999.

Principe, J. C., Euliano, N. R., & Lefebvre, W. C. (2000). *Neural and adaptive systems: Fundamentals through simulations*. Hoboken, NJ: Wiley.

Rall, W., & Sergev, I. (1987). Functional possibilities for synapses on dendrites and on dendritic spines. In G. M. Edelman, W. F. Gail, & W. M. Cowan (Eds.), *Synaptic function* (pp. 603–636). New York: Wiley.

Rhodes, P. A. (2008). Recoding patterns of sensory input: Higher-order features and the function of nonlinear dendritic trees. *Neural Computation, 20*, 2000–2036.

Richmond, B. J., Optican, L. M., Podell, M., & Spitzer, H. (1987). Temporal encoding of two dimensional patterns by single units in primate inferior temporal cortex. I. Response characteristics. *Journal of Neurophysiology, 57*, 132–146.

Rieke, F., Warland, D., de Ruyter van Steveninck, R., & Bialek, W. (1999). *Spikes: Exploring the neural code*. Cambridge, MA: MIT Press.

Schlichting, H., & Gersten, K. (2000). *Boundary-layer theory*. New York: Springer.

Sejnowski, T. J. (1977). Storing covariance with nonlinearly interacting neurons. *Journal of Mathematical Biology, 69*, 303–321.

Shepherd, G. M. (2004). *The synaptic organization of the brain* (5th ed.). New York: Oxford University Press.

Shepherd, G. M., & Brayton, R. K. (1987). Logic operations are properties of computer-simulated interactions between excitable dendritic spines. *Neuroscience, 21*(1), 151–165.

Shepherd, G. M., & Grillner, S. (2010). *Handbook of brain microcircuits*. New York: Oxford University Press.

Singer, W., & Gray, C. M. (1995). Visual feature integration and the temporal correlation hypothesis. *Annual Review of Neuroscience, 18*, 555.

Sirevaag, A. M., & Greenough, W. T. (1987). Differential rearing effects on rat visual cortex synapses, III neuronal and glial nuclei, boutons, dendrites, and capillaries. *Brain Research, 424*, 320–332.

Slepian, D. (1956). A class of binary signaling alphabets. *Bell Systems Technical Journal, 35*, 203.

Spratling, M. W., & Johnson, M. H. (2001). Dendritic inhibition enhances neural coding properties. *Cerebral Cortex, 11*, 1144–1149.

Spratling, M. W., & Johnson, M. H. (2002). Pre-integration lateral inhibition enhances unsupervised learning. *Neural Computation, 14*, 2157–2179.

Sutherland, J. G. (1992). The holographic neural method. In S. Branko (Ed.), *Fuzzy, holographic, and parallel intelligence*. Hoboken, NJ: Wiley.

Tal, D., & Schwartz, E. L. (1997). Computing with the leaky integrate and fire neuron: Logarithmic computation and multiplication. *Neural Computation, 9*, 305–318.

Tolhurst, D. J., Movshon, J. A., & Dean, A. F. (1983). The statistical reliability of signals in single neurons in cat and monkey visual cortex. *Vision Research, 23*, 775–785.

Turner, M., & Austin, J. (1997). Matching performance of binary correlation matrix memories. *Transactions of the Society for Computer Simulation International, 14*(4).

van Vreeswijk, C., & Sompolinsky, H. (1998). Chaotic balanced state in a model of cortical circuits. *Neural Computation, 10*, 1321–1371.

Victor, I. D., & Purpura, K. P. (1996). Nature and precision of temporal coding in visual cortex: A metric-space analysis. *Journal of Neurophysiology, 76*, 1310–1326.

von der Malsburg, C. (1981). The correlation theory of brain function. In E. Domani, J. L. van Hemmen, & K. Schulten (Eds.), *Models of neural networks II*. Berlin: Springer.

Willshaw, D. J., Buneman, O. P., & Longet-Higgins, H. C. (1969). Non-holographic associative memory. *Nature, 222*, 960–962.

Wittner, L., Henze, D. A., & Zaborszky, L. (2007). Three-dimensional reconstruction of the axon arbor of a CA3 pyramidal cell recorded and filled in vivo. *Brain Structure and Function, 212*, 75–83.

Wong, M. T. T. (1989). Cytochrome oxidase: An endogenous metabolic marker for neuronal activity. *Trends in Neuroscience 12*(3), 94–101.

Wong, R. K. S., Prince, D. A., & Basbaum, A. I. (1979). Intradendritic recording from hippocampal neurons. *Proceedings of the National Academy of Sciences, 76*, 986–990.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control 8*(3), 338–353.

Zadeh, L. A., Klir, G. J., & Yuan, B. (1996). *Fuzzy sets, fuzzy logic, and fuzzy systems*. Singapore: World Science Press.

Zador, A. M., Clairborne, B. J., & Brown, T. H. (1992). Nonlinear pattern separation in single hippocampal neurons with active dendritic membrane. In J. Moody, S. Hanson, & R. Lippmann (Eds.), *Advances in neural information processing systems, 4* (pp. 51–58). San Mateo, CA: Morgan Kaufmann.

Received August 23, 2010; accepted February 18, 2011.