A Spiking Self-Organizing Map Combining STDP, Oscillations, and Continuous Learning

Timothy Rumbell, Susan L. Denham, and Thomas Wennekers

***Abstract*— The self-organizing map (SOM) is a neural network algorithm to create topographically ordered spatial representations of an input data set using unsupervised learning. The SOM algorithm is inspired by the feature maps found in mammalian cortices but lacks some important functional properties of its biological equivalents. Neurons have no direct access to global information, transmit information through spikes and may be using phasic coding of spike times within synchronized oscillations, receive continuous input from the environment, do not necessarily alter network properties such as learning rate and lateral connectivity throughout training, and learn through relative timing of action potentials across a synaptic connection. In this paper, a network of integrate-and-fire neurons is presented that incorporates solutions to each of these issues through the neuron model and network structure. Results of the simulated experiments assessing map formation using artificial data as well as the Iris and Wisconsin Breast Cancer datasets show that this novel implementation maintains fundamental properties of the conventional SOM, thereby representing a significant step toward further understanding of the self-organizational properties of the brain while providing an additional method for implementing SOMs that can be utilized for future modeling in software or special purpose spiking neuron hardware.**

***Index Terms*— Artificial neural networks, neural engineering, self-organizing feature maps, unsupervised learning.**

I. Introduction

TOPOLOGICALLY ordered spatial representations of features can be found in various sensory cortical areas [1], [2], such as ocular dominance bands [3], [4] and orientation maps [5], [6] in cat and primate primary visual cortex, a tonotopic map in the auditory cortex in cats [7], a gustotopic map in the primary taste cortex [8], an odor relationship representation in the olfactory bulb [9], a whisker map in the barrel cortex in rodents [10], [11], and a somatosensory map in the somatosensory cortex of primates [12]. Throughout development these cortical feature maps accrue several distinctive properties, such as disruptions that reflect actual discontinuities in the sensory periphery and disproportionate representation of early developing portions of the receptor sheet [1]. The relationship between the properties of the sensory input is reflected in the relationship between the physical areas of cortex that are tuned to represent those properties [13], and the physical substrate of representation is capable of reorganizing to a change in input properties [14], [15].

The self-organizing map (SOM) is a neural network algorithm inspired by the organizational structure of feature maps in the brain [16]. Throughout learning a SOM gradually maps statistical correlations present in a set of data onto a simple, low dimensional, topological output representation. The discovery of features and feature relations in a complex space is also a goal of principal component analysis [17], and the combination of winner-takes-all competition and a neighborhood function for learning allows generated representations to be sparse, orthogonalized, and analogous to the representations developed by clustering algorithms [18]. Learning in the SOM is unsupervised, making it useful in a variety of situations and easily modified to suit a variety of purposes (for a review of SOM applications, see [16]; for recent examples of SOM modifications, see [19] and [20]).

However, the SOM algorithm in its conventional form differs from the methods of learning and information coding present in cortical feature maps in several functionally important ways: biological neurons do not have direct access to global information, such as the distance of their weights from the current input data point relative to all other neurons in the layer, or the actual values of the current input data point, toward which their weights should be moved; information is transmitted in a sequence of postsynaptic potentials (PSPs), often synchronized within and between the layers [21], with coding of information in relative firing times contributing to rapid hierarchical processing abilities [22]–[24]; input enters the sensory periphery continuously, so must be converted from this stream into spike sequences; learning can be ongoing [25], without necessarily resorting to reductions in learning rate and neighborhood width at prescribed intervals throughout training. This paper presents a series of modifications to the original SOM that addresses these issues while retaining functionality, thereby contributing toward understanding of the self-organizational properties of the brain while providing a SOM implementation that can be used for future modeling in software or special purpose spiking neuron hardware.

Temporal coding can be incorporated into the SOM by inserting spiking neurons in place of neurons that make weight comparisons or communicate through time-continuous gradual activity values. An influential model [26] consists of two layers of integrate-and-fire neurons. The first layer encodes input in the firing times of neurons in an array that collectively represent a real value. The second layer represents the SOM: the best matching unit is determined locally through integration of the input spikes; the neighborhood function consists of local interactions through excitatory and inhibitory connections. The network self-organizes in an unsupervised manner according to a learning rule based on spike timing, but makes use of a global reference time to establish weight adjustments. The reliance on globally available information

Manuscript received September 3, 2012; revised July 1, 2013; accepted September 10, 2013. Date of publication October 17, 2013; date of current version April 10, 2014.

The authors are with the Cognition Institute, Plymouth University, Plymouth PL4 8AA, U.K. (e-mail: timothy.rumbell@plymouth.ac.uk; s.denham@plymouth.ac.uk; thomas.wennekers@plymouth.ac.uk).

Digital Object Identifier 10.1109/TNNLS.2013.2283140

TABLE I
NETWORK PARAMETERS FOR ALL SIMULATIONS DESCRIBED IN SECTION III

(A) Neuronal parameters, used in (1) and (4)									
Δt	$\tau_m^{u,v}$	$\tau_m^{Inh\ u}$	V_{rest}	θ^u	$\theta^{Inh\ u}$	θ^v	g		
0.1ms	1ms	0.5ms	0.0	0.5	0.01	1.0	0.0		
(B) Synaptic parameters, used in (2) and (3) for different synapse types									
u to v		u to $Inh\ u$ (exc.)		u to $Inh\ u$ (inh.)		$Inh\ u$ to u		v to v	
τ_r	τ_f	τ_r	τ_f	τ_r	τ_f	τ_r	τ_f	τ_r	τ_f
0.2ms	1.0ms	0.4ms	2.0ms	0.2ms	1.0ms	1.0ms	5.0ms	0.1ms	0.5ms
(C) Maximum magnitudes of synaptic connection strength									
$w_{max}^{u\ to\ v}$	$w_{max}^{u\ to\ Inh\ u\ (exc.)}$	$w_{max}^{u\ to\ Inh\ u\ (inh.)}$	$w_{max}^{Inh\ u\ to\ u}$	$w_{max}^{v\ to\ v}$					
2.2	1.0	1.0	100.0	1.0					
(D) Neighbourhood parameters, used in (6) and (7), for layer v									
a	b	X	X'						
3.0	3.0	3.0	3.0						
(E) Learning parameters, used in (5)									
A_+	A_-	τ_+	τ_-						
0.0016	0.0055	11ms	10ms						

makes the rule unsuitable for the current purpose. Additionally, the network is reset after each presentation of an input datum as it is unable to handle continuously presented input without this mechanism. Representative mappings of 1-D and 2-D input data are reported as a result suggesting that important functional properties of the SOM are present.

The network presented and analyzed in Sections II and III implements the principles of the SOM algorithm in a network of spiking neurons. Several features of the model are based in part on the model detailed above [26]. This spiking SOM can act as a base for future work, introducing additional biological constraints to replicate properties of the cortex. It can be used to help realize hardware and other technical applications of the SOM algorithm that can benefit from the use of temporal coding. Section II describes the model, explaining the combination of mechanisms used, and the method of evaluating output quality. Section III presents the results of testing the model, including extensive parameter analyses, demonstration of key features of the model and application to categorization tasks. Finally, Section IV summarizes this paper, discussing the current model in terms of its place within the field relative to existing models, and its limitations.

II. METHODS

The model described in this section is a two layer network of integrate-and-fire neurons similar to the model proposed in [26]. In both models, firing in the first layer encodes an actual data point as a temporal sequence of spikes. Neurons in the second layer respond as a result of this firing, and one of them will win the competition, i.e., fire first. Further firing in the second layer is influenced by lateral connections, representing the neighborhood. Neurons physically close to the winning unit fire sooner due to stronger excitatory lateral influence, and neurons further from the winning unit fire later due to weaker excitatory or inhibitory lateral influence.

The current model differs from that of [26] by incorporating realistic PSPs, spike-timing-dependent plasticity (STDP), and inhibitory control of the input layer to generate oscillatory behavior (facilitating continuous input presentation through temporal segmentation), while allowing for continuous, ongoing learning, and stable neighborhood size. The mechanisms controlling each of these aspects of the network will be detailed in this section. Parameters for the equations introduced below are listed in Table I.

Simulations were conducted using custom made C software.

A. Neuron Model

Leaky integrate-and-fire neurons were used for all neurons in the network, modeled by

$$\tau_m \frac{dV}{dt} = I(t) - V + g\eta(t), \quad \text{if } V \geq \theta \text{ then spike and reset } V = 0. \tag{1}$$

Each neuron has a membrane potential V that increases by integrating current input $I(t)$, and leaks toward a resting potential of 0 when there is no input from its afferent synapses. The membrane potential time constant, τ_m, is set to 1 ms for all neurons in the model, except u_{Inh}, in which $\tau_m = 0.5$ ms. A spike is generated when a neuron's membrane potential reaches a firing threshold θ, which varies by layer (see Section II-C). Neurons are also subjected to a Gaussian white noise process η, which is scaled by a factor g; $g = 0$ (i.e., no noise injection) for the majority of testing, but the robustness of network output to noise is tested by varying g in Section III.

All connections between the neurons are modeled synapses with weight w_{ij}, which transmit PSPs to the postsynaptic neuron when a spike is generated in the presynaptic neuron. PSPs are modeled as α-functions, using

$$\tau_r \frac{ds_1}{dt} = (s - s_1) \tag{2}$$

$$\tau_f \frac{ds_2}{dt} = (s_1 - s_2) \tag{3}$$

where s is a binary value representing instantaneous presence or absence of a presynaptic spike, s_1 is an internal state

variable, s_2 is the α-function output, and τ_r and τ_f are time constants for the rise and fall duration of the response. Time constants are set independently for each layer (see Table I), but the ratio of τ_r:τ_f is always set at 1:5.

Input current to a neuron at time t, $I(t)$, is calculated by

$$I(t) = \sum_j w_j s_{2j}(t) \tag{4}$$

where w_j represents the connection weight (or synaptic efficacy) between the neuron j (presynaptic) and the current neuron (postsynaptic), and $s_{2j}(t)$ represents the current α-function output from neuron j.

B. Learning

The learning rule used in [26] makes use of some artificial features. Neurons have access to a global time stamp, which allows the gap between the firing time of the best matching unit and the firing time of the current neuron to be calculated. In addition, the actual input value is compared with the current synaptic weight to determine the weight change, meaning that the synapse has knowledge of the input patterns. Replacing this learning rule with a standard STDP rule removes these issues, providing a basis for learning that is more biologically plausible [27], and more robust due to reliance only on local information to which each neuron already has access.

STDP [28] provides a function for long-term potentiation (LTP) or depression (LTD) of synapses based on the time difference Δt between a single pair of pre and postsynaptic spikes, in neurons i and j, respectively, according to $w_{ij} \rightarrow w_{ij}(t) + f(\Delta t)$. A linear multiplicative rule for LTD and exponential multiplicative rule for LTP are used, according to [29]

$$f(\Delta t) = \begin{cases} \exp^{-w_{ij}} A_+ \left(1 - \frac{1}{\tau_+}\right)^{\Delta t}, & \text{if } \Delta t > 0 \\ -w_{ij} A_- \left(1 - \frac{1}{\tau_-}\right)^{\Delta t}, & \text{if } \Delta t \leq 0. \end{cases} \tag{5}$$

A_+ and A_- are both positive and determine the maximum amount of synaptic strengthening and weakening that can occur, respectively. τ_+ and τ_- are time constants determining the range of time in which synaptic strengthening and weakening will occur, respectively. Weights are bounded between 0 and $w_{\max}$. The specific values these five variables are set to, along with the motivation for differing forms of the rules for LTD and LTP, are discussed in Section III-B, which details an extensive parameter search conducted to optimize learning.

C. Network Structure

The spiking SOM network structure is shown schematically in Fig. 1. Conventional instantiations of the SOM receive input (numerical values for each dimension in the input data set) directly into the SOM neurons. In the spiking version presented here these values (represented by node I) feed into a bank of neurons within an intermediate input layer, u. The actual input values are converted into a temporal spike sequence within each bank through the use of an inhibitory mechanism, described in Section II-C.1. This spike sequence then drives the SOM layer, v, through all-to-all feedforward

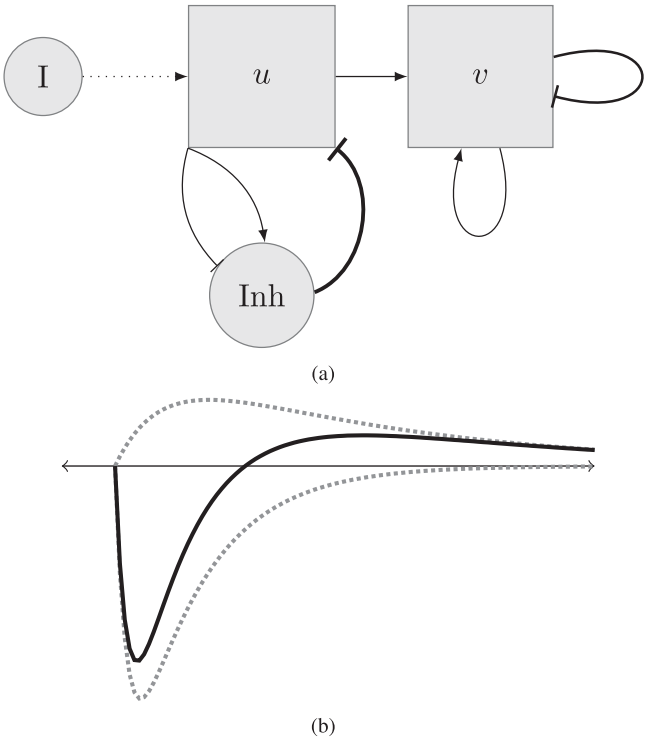

Fig. 1. (a) Spiking SOM network structure: an actual input value I is encoded into spikes times by nodes in layer u; the inhibitory unit causes oscillations, allowing continuous input to be presented to u (see Section II-C.1 for details); feedforward connections from u to v drive firing in v; early firing in v determines the location of output activity in v through lateral (neighborhood) connections (see Section II-C.2 for details). (b) Time course of the PSP from both of the u to Inh synapses (inhibitory is the lower dotted line, excitatory is the upper dotted line), together with the combined effect on the membrane potential of Inh_u (solid line).

synaptic connections. All-to-all lateral synaptic connections in v implement the neighborhood function.

1) Input Encoding Within Oscillations: Input to this model is in the form of an m-dimensional vector of real numbers; each dimension I_n of this vector needs to be encoded in the firing of neurons in layer u. This can be achieved by representing each I_n with a bank of neurons u_n from layer u. Each neuron u_i in that bank is tuned around a point from within the range of values that I_n can take [30]. A Gaussian function is used such that the closer the actual value I_n is to the tuned value of u_i, the higher the activation to u_i, as shown in Fig. 2. The use of integrate-and-fire neurons means that u_i with higher activation levels will reach threshold earlier and fire faster than u_i with low activation levels. This creates a unique yet structured temporal pattern of spikes for each value of I_n.

Continuous stimulus presentation is an important feature in constructing a versatile and general network, allowing network operation to be ongoing, with no need for discretization of temporal aspects, such as automatic resetting of the network state at each training step. Using the current method of input encoding, continuous input presentation provides constant activation to layer u. This disrupts the temporal representation of each input pattern: the first u_i to fire will begin integrating input again first after resetting, and fire sooner in the next cycle of firing. This quickly desynchronizes the input neurons

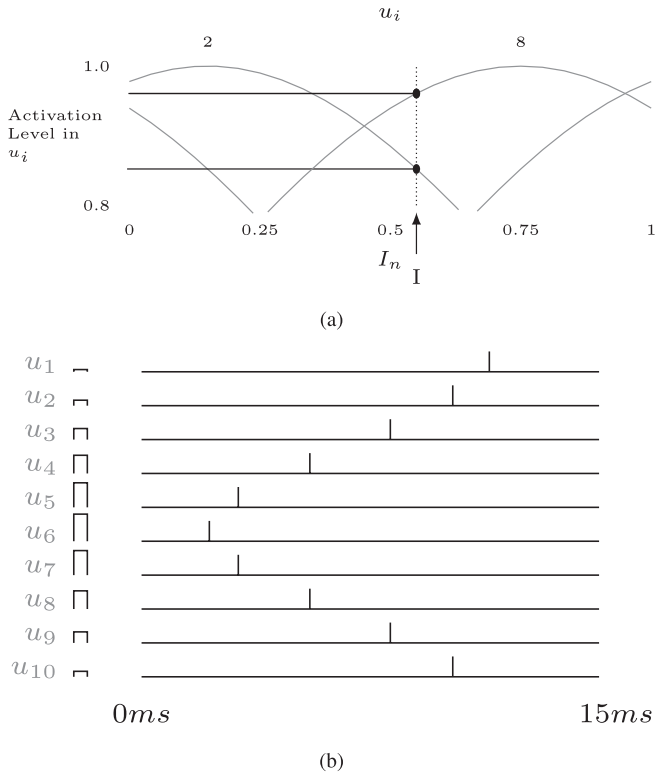

Fig. 2. Generation of a temporal sequence from a value in an input dimension and a bank of neurons tuned to points within that dimension. (a) Generation of an activation value for each neuron (u_i) from within the bank of input neurons representing input dimension (I_n). An example input value, at arrow I, of 0.55 is shown by the vertical dotted line. Activation values are established by the vertical point at which this dotted line crosses the tuning curve for a (u_i) neuron. Example tuning curves are shown for neurons 2 and 8, tuned to values of 0.15 and 0.75, respectively. The vertical locations of the horizontal lines from this curve then represent the activation levels. (b) Example of a generated output firing sequence. The height of the bars shown next to each u_i represents the activation level generated from the intersection of a given input value and the neuron's tuning curve.

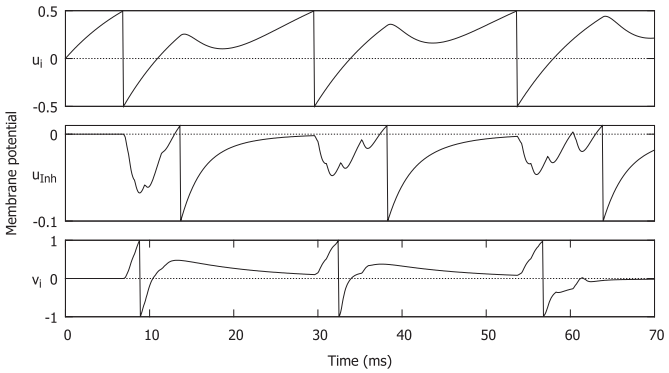

Fig. 3. Time course of membrane potentials in examples of the neurons u_i (top), u_{Inh} (middle), and v_i (bottom). Firing threshold is the top of the figure in each case. A membrane potential reaching threshold generates a spike and the membrane potential is reset, indicated by vertical lines. Potentials in layer u neurons increase, generating spikes, at which time the potential in u_{Inh} is depressed temporarily, due to the inhibitory connection from each u_i. The excitatory connections from each u_i then result in an increase in potential in u_{Inh}, which fires. This causes a temporary stagnation of the membrane potential in u_i, which would normally increase continuously due to continuous exposure to the actual input. The membrane potentials of neurons in layer v begin increasing after the onset of firing in u, some of which will spike (shown here) if they have strong weights to the early firing layer u neurons.

from one another, meaning that information is encoded in the firing rates rather than the spike times.

This problem is remedied through an inhibitory neuron, Inh_u, which responds to firing in u with a slow inhibitory PSP fed back to all the neurons in u. The inhibition depresses the membrane potential of all u_i after firing, establishing an approximate baseline for the effects of activation from I, creating a close to identical repetition of the temporal sequence. Inhibition in response to excitation creates an oscillatory behavior, with a period of firing across the layer followed by a period of inhibition ahead of the next spikes. The temporal structure of spiking within each oscillation is maintained, meaning that information is now encoded in spike times rather than firing rates. Under the parameters in Table I, inhibition in layer u commences 10–15 ms after the start of firing in the layer, and the inhibitory PSP depresses membrane potentials sufficiently to prevent firing for approximately 10 ms, when the temporal sequence begins again. This behavior is shown between 15 and 30 ms in the upper chart in Fig. 3, which shows the membrane potential time course for a sample u_i. This inhibitory effect could also be reached by a population of spiking neurons, but is simplified to a single neuron here. Furthermore, firing in v synchronizes with oscillations in u, as the feedforward connections between the layers drive activity in v.

The inhibitory neuron receives input from each u_i through both an excitatory and an inhibitory synapse, as shown in Fig. 1(a). The excitatory synapse has a relatively long time constant, making it slow, and the inhibitory synapse has a relatively short time constant, making it fast, as shown in Fig. 1(b). The overall effect of the pair of connections on the membrane potential of Inh_u is an initial dip, followed by a recovery into the positive region, shown in the middle chart of Fig. 3. Combined with a resting potential fractionally below the firing threshold, the effect is that the membrane potential of Inh_u will stay subthreshold as long as there are spikes in u within a reasonably short time of each other, and then reach threshold when there is a sufficient gap in activity in u. Thus, membrane potentials in u_i are reset when there is a large gap in firing, or when the pattern ends.

In summary, oscillations are induced in the layer through excitatory followed by inhibitory firing, as in a classic excitatory-inhibitory feedback loop [31]. Oscillations of this type allow input neurons to be constantly excited and maintain a reliable firing pattern for an input. In turn, this allows for a stimulus, or input datum, to be continuously presented to the network, resulting in a versatile and reliable input coding mechanism.

2) Neighborhood Function: Self-organization in the spiking SOM is produced through the use of a lateral interaction profile (analogous to a neighborhood function), and STDP (see Section II-B). Learning in the spiking SOM occurs when an output node fires in response to the input sequence; in particular, learning the current input values is strongest when an output neuron fires soon after the start of the input sequence, causing greater strengthening of the afferent synapses from nodes that better represent the actual input

values. Lateral synaptic connections in the output layer v send excitatory signals to neurons that are within a certain distance and inhibitory signals to more distant neurons. This lateral profile encourages neurons within a spatial region to fire and discourages neurons outside of that region from firing.

A suitable neighborhood kernel, both in terms of capturing qualitative properties of cortical structure and functional properties of the SOM network, is a Mexican-hat function [32]. Keith–Magee [33] discussed a lateral connection initialization function for a SOM based on a Laplacian kernel

$$\bar{w}_{ij} = (1+a)G(||i-j||, r) - aG(||i-j||, br) \quad (6)$$

in which the lateral connection strength $\bar{w}_{ij}$ between the output neurons at locations i and j in the grid is determined by (6), where a represents the magnitude of the negative component of the function, b determines the decay of the negative component of the function, r determines the radius of the positive component of the function, and the function G is a Gaussian function of the distance between i and j.

The traditional SOM formulation includes a decaying neighborhood width over time to produce a more finely tuned output mapping. An appropriate decay function for the width (r) of this lateral connection kernel is established, through a series of experiments [33], as a step function with a filter to smooth the step function over time

$$r(t) = X - \frac{X - X'}{1 + (\sqrt{2}-1)((T/t)^{2n})} \quad (7)$$

where $r(t)$ gives the value of r to use in (6) at training step t in the simulation, X and X' are values of r at the start and end of training, respectively, T represents the value of t that the step is centered around, and n is the order, or amount of smoothing, of the smoothed step function.

Compared with classic linear decay schemes and a non-smoothed step decay, use of (7) results in an accurate output mapping being reached more quickly [33]. In addition, identical values for X and X' results in no neighborhood decay, facilitating a simple transition between the regimes.

The current spiking SOM model uses (6) and (7) to establish the lateral synaptic weights. For a 10×10 grid of neurons in layer v, parameter searching reveals that setting a, b, X, and X' all to 3.0 provides a lateral connection profile capable of topological map formation. Identical values for X and X' lead to a constant r in (6); this was found to be capable of topological map formation, demonstrated in Section III, although it is possible that more accurate mappings can be obtained using a larger X and smaller X' [33]. This decision was made to ensure continuous learning in the output map, identified as a goal of the current system. Decay of neighborhood size throughout training was used for results in Sections III-F and III-G, however, and in these cases the step function in 7 was used to modify the lateral weights.

3) Self-Organization: The inhibitory current generating an oscillation, the temporal coding of each input dimension in spike times, the neighborhood function, and multiplicative STDP all contribute to the self-organization of the output map. At the start of an oscillation, the input neurons have depressed membrane potentials due to inhibition from the previous oscillation. Membrane potentials increase through constant input current and early spikes within an oscillation indicate neurons that represent the actual input well. Fig. 3 shows the relationship between these membrane potentials, with a spike in a u_i leading to temporary depression of the membrane potential of the inhibitory neuron. The inhibitory neuron then fires, temporarily preventing the increase of the membrane potential of all u_i. This first part of the input firing pattern generates a spike in the output layer from the neuron best matching the input firing. This is followed by the firing of nearby output neurons due to lateral activity, all before firing of neurons in the input layer that are relatively poor representatives of the actual input. STDP causes the synaptic connections from neurons in the early part of the input pattern to be strengthened, and the later part of the input pattern to be weakened, for output neurons within the neighborhood of the winning neuron. This is shown schematically in Fig. 4.

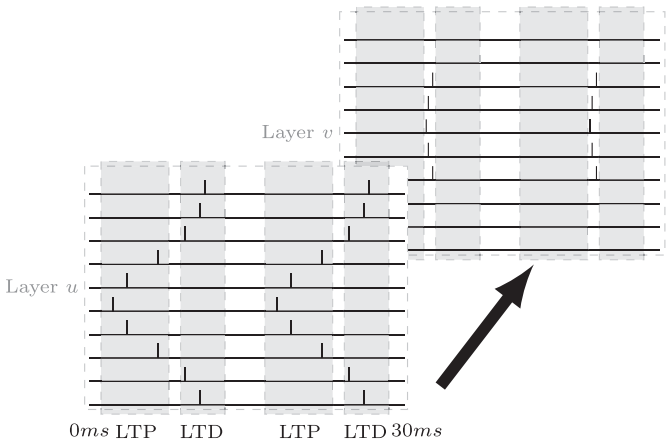

Fig. 4. Mechanics of self-organization in the spiking SOM. A spike sequence in a bank of neurons in layer u represents the actual input value, with early firing neurons being well tuned to the actual value, and late firing neurons being poorly tuned to the actual value. Black arrow: all-to-all feedforward synaptic connections from u to v. At some point in the firing of the pattern in u, a neuron in v fires, winning the competition and becoming the best matching unit. Neighboring neurons in v are caused to fire within close temporal proximity. The gap between the LTP and LTD boxes in layer u represents the time at which firing in v occurs relative to the firing in u. Synapses from any neurons that have fired before that point are strengthened (the LTP box) and synapses from any neurons that fire after that point are weakened (the LTD box).

A multiplicative form of STDP helps to ensure that weights reach a stable point roughly proportional to how often an input neuron fires before an output neuron relative to how often it fires after an output neuron. Output neurons will respond for actual input values that are a distance away from their preferred input value, due to lateral excitation. As such, weights will be increased at synapses from input neurons that normally fire after the output neuron in the output neuron's preferred input pattern. Weights will be decreased from input neurons that normally fire before the output neuron in the output neuron's preferred input pattern. These changes are weight dependent, so for a certain weight value a few instances of depression balance with a greater number of instances of excitation, and vice versa, creating stability. This stability means that there can be precise differentiation between the

winning output neurons; adjacent output neurons will prefer similar input neurons, so stable weights between the maximum and the minimum are important in determining, which of several neurons with similar preferences reaches threshold first.

D. Quality of Map Formation Metric

The ability of a SOM to map an input dataset can be assessed by checking the topographic mapping error of the output map given the input data. For a map with no mapping error, the relative distance between any pair of data in the input space is the same as the relative distance between the locations activated by that pair of input data in the output map. Metric multidimensional scaling (MDS) can be used to assess this, according to

$$E_{\mathrm{MDS}} = \sum_{i=1}^{N} \sum_{j<i} (F(i, j) - G(M(i), M(j)))^2 \qquad (8)$$

where N is the number of input patterns, $F(i, j)$ represents the actual dissimilarity of the pair of input patterns i and j (measured as Euclidean distance), and $G(M(i), M(j))$ represents the dissimilarity between the locations in the output map representing patterns i and j (measured as Euclidean distance), where $M(i)$ and $M(j)$ are the locations of the winning nodes in layer v for input patterns i and j, respectively, [34]. The value of E_{MDS} represents how well the final network mapping preserves the topology of the input dataset. The most accurate mapping achieved is one in which relative distances between the patterns in the input space are reflected exactly by relative distances between the neurons representing those patterns in the output space, resulting in a minimum E_{MDS} value of 0. The least accurate mapping observed in practice is one in which all the input patterns result in activation of the same location in the output map; the E_{MDS} value for this situation will vary depending on the distribution of input patterns. This test is used in [26] to analyze their spiking SOM, with final E_{MDS} values of under 20% of the starting value being reported. Values in Section III are reported using the mean values for E_{MDS}, with the final summed E_{MDS} divided by the total number of pairs of input patterns compared, with give a value that can be compared regardless of the number of patterns in the input space.

Alternate methods for analyzing the quality of output mapping produced by a SOM are available (see [34]–[36] for reviews). MDS has been selected as an analytical tool ahead of other techniques for two primary reasons. First, it meets two criteria proposed in [37] that are required for a SOM analysis tool: it should provide evidence of the self-organizing process during training (shown through a reduction in error value); and it should measure the embedding of the set of neurons into the data manifold (the error value measures how well changes in the input space are mapped by changes in the output space). Second, given that it represents an appropriate SOM analysis tool, it is important that results described here are generalizable for potential comparison with other spiking neuron network implementations. Other spiking neuron networks may not be specifically designed as SOM implementations, and specific SOM analysis metrics would lose their relevance when comparing the topographic mapping capabilities of one spiking neuron network with another.

III. Results

This section covers the results of testing conducted to confirm the behavior of the spiking SOM. A measure of the quality of map formation is introduced in Section II-D, to be used to interpret the rest of the testing results. The parameters used during testing are described in Section III-A, and a parameter search on the variables involved in (5) is described in Section III-B, to determine the range of values that result in good map formation. Section III-C demonstrates the robustness of learning under the chosen parameters in the presence of noise. The spiking SOM has been tested in common scenarios used to test the conventional SOM: the response of the spiking SOM to evenly distributed, randomly selected, 2-D input data is analyzed in Section III-D. Finally, the results of categorization tests carried out with the spiking SOM are reported in Sections III-E, III-F, and III-G.

A. Network Parameters

The testing conducted in Sections III-B and III-D made use of standardized parameters for the network, shown in Table I. In summary, each dimension n in the input I was associated with a bank u_n of 10 neurons in layer u. The value of n was set to 2 for 2-D input. The preferential values of the neurons in each u_n were equally spaced between 0.05 and 0.95. Gaussian tuning curves around these preferential points were used for calculation of the activation values, with distance calculations including circular wrapping from 1 to 0.

Layer v, the SOM layer, was initialized with 100 neurons, arranged in a 10×10 grid through the lateral connection weights. Feedforward connections were initialized from all layer u neurons to all layer v neurons with a randomized weight between 0.4 and 0.6 of the maximum synaptic weight $w_{\max}^{u \text{ to } v}$. The radius r in (6) was set to 3.0, and the distance between the neurons in the layer were calculated with toroidal structure. Values of τ_+ and τ_- in (5) were set to 11 and 10 ms, respectively. This width of learning window approximately matches the temporal width of a network oscillation, leading to a negligible influence on learning of spikes from within neighboring oscillations.

At the start of each training step an input value was determined by selecting randomly from 10 values for each dimension, equally spaced between 0.05 and 0.95, making a total of 100 input patterns from within the 2-D input space. A training step lasted through five oscillations of the network (approximately 125 ms) before the input pattern was changed. The network was allowed to learn for 4000 training steps. Quality of map formation was assessed using mean E_{MDS} for each pair of input patterns. Given that for the 2-D case both the input and output space wrap toroidally, this situation results in a maximum mean E_{MDS} value of $(1/6)$.

The maximum connection strength values for synapses with presynaptic neurons in layer u are varied for the categorization tests conducted in Sections III-F and III-G. These datasets contain four and nine input dimensions, resulting in 40 and

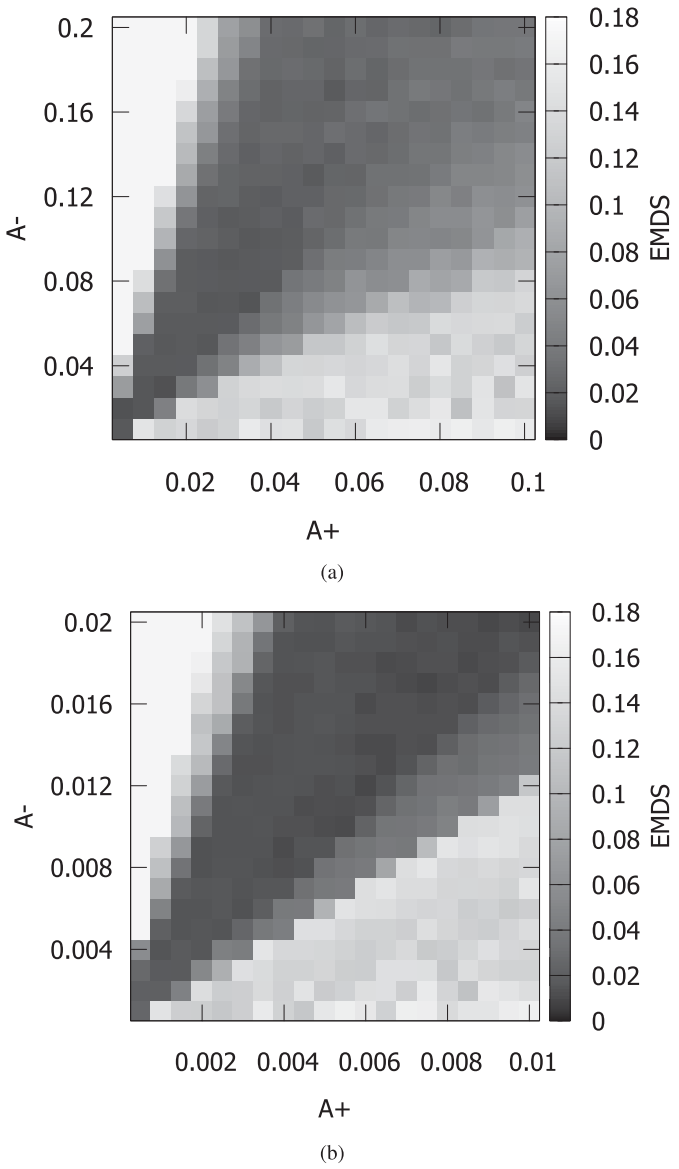

Fig. 5. Parameter search results for A_+ to A_- values: normalized final E_{MDS} values averaged from 30 trials are presented, for ranges of A_+ and A_- values at (a) low resolution and (b) high resolution.

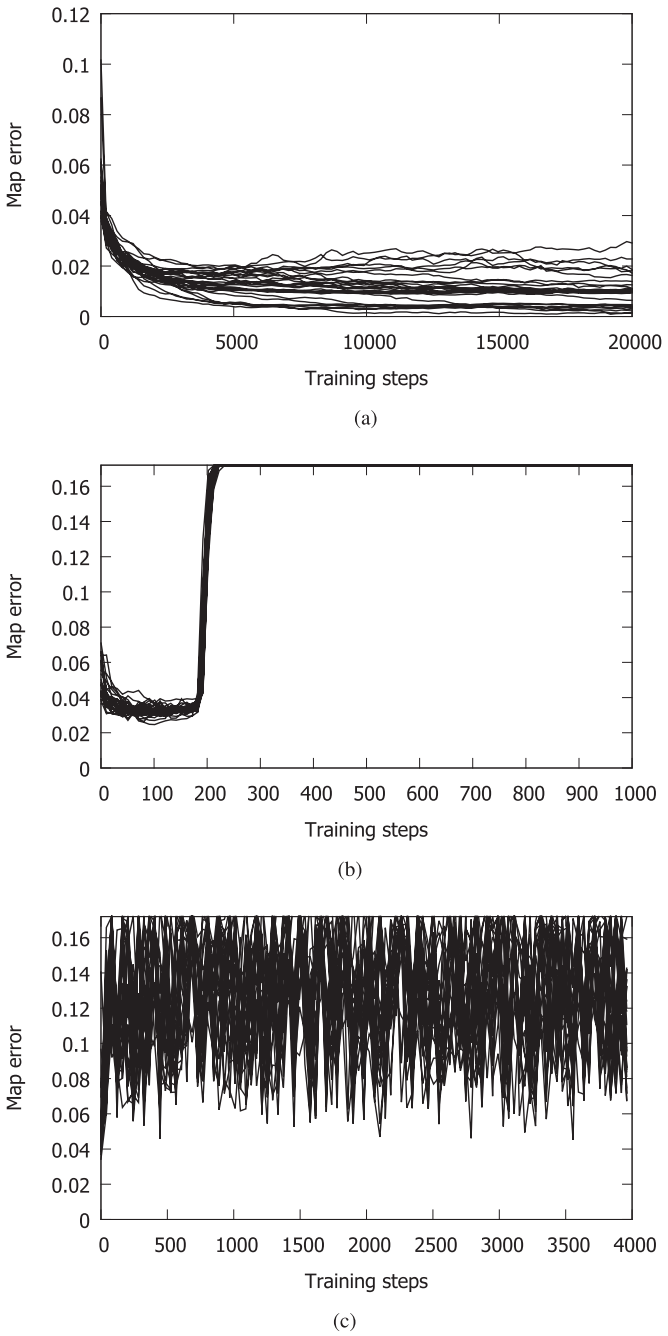

Fig. 6. Time course of E_{MDS} values for three points in the parameter space. (a) Normal learning ($A_+ = 0.002$, $A_- = 0.007$). (b) Dominance of depression ($A_+ = 0.0005$, $A_- = 0.01$), and (c) dominance of potentiation ($A_+ = 0.02$, $A_- = 0.005$).

90 neurons in u, respectively. As such, the value of $w_{\max}^{u\ \mathrm{to}\ v}$ is scaled down to 1.5 in Section III-F, and 0.7 in Section III-G, and the $w_{\max}^{u\ \mathrm{to}\ Inh\ u}$ values to 0.4 in Section III-G.

B. Learning Parameter Analysis

Parameter testing was conducted to establish suitable values for the maximum and minimum weight change parameters, A_+ and A_- from (5). A test of a parameter set consisted of 30 randomly initialized maps, trained using the method described in Section III-A involving random selection from 100 input patterns evenly spaced across the 2-D surface. An average normalized E_{MDS} value taken at the end of training used to gauge the quality of maps formed with those parameters. Coarse- and fine-grained searches were conducted, the results of which are shown in Fig. 5(a) and (b), respectively.

The fine-grained search results establish that, for a range of A_+ values up to 0.01, and A_- values up to 0.02, a ratio between 1:2.5 and 1:3.5 of A_+ to A_- will result in good map formation. The coarse-grained search result establishes that there is little performance degradation up to A_+ values of 0.045 and A_- values of 0.11, meaning that large weight changes relative to the maximum weight can still result in map formation. Only 500 training steps were used in the coarse-grained simulation results; the high learning rates involved result in a fluctuating error value after this point, rather than increased convergence of error values.

Fig. 6 shows the progression of E_{MDS} values throughout training for important locations in the parameter space.

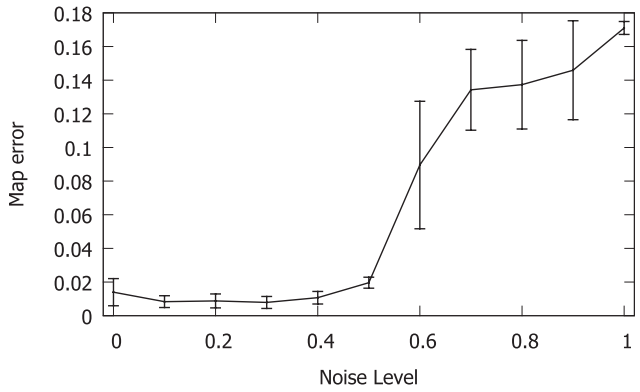

Fig. 7. Average E_{MDS} value after training plotted against the value of variable g, magnitude of noise in the neuron model. Error bars represent one standard deviation. Accuracy of the output mapping stays high until noise levels reach a critical point around 0.5, with accuracy of the final mapping ending up close to the maximum error value as g reaches 1.

Fig. 8. Final u (y-axis) to v (x-axis) synaptic weights after training with 2-D data (light: strong connection and dark: weak connection; the top half of the graphic is the bank of input neurons encoding the first dimension and the bottom half is the bank encoding the second dimension; and each 10 steps along the x-axis represents a row of output neurons, then the next 10 represent the next adjacent row, etc.). The connections to the output layer vary in 1-D across an individual row (with consistency throughout the map), and vary in the other dimension across the rows, encoding the current input as a location in the output map in an organized way. A change in one input dimension (the bottom 10 rows) as one moves through each row of the output layer (each block of 10 columns), and a change in the second dimension (the top 10 rows) as one moves down through rows in the output layer (changes between each block of 10 columns in the figure) can be observed. For a given output neuron (one column), a gradual decrease in weights either side of a central point in each input dimension is visible.

Fig. 6(a) shows the mapping error attained for multiple trials for an A_+ to A_- ratio of 1:3.5, a ratio that reliably results in good map formation. Fig. 6(b) shows the degradation of map quality for an A_+ to A_- ratio of 1:20. Depression dominates, and weights are gradually lowered until activity in u no longer evokes any spikes in v. If no neurons win the competition for any input pattern, there is no distance between the winning nodes for any input pattern, so a maximum error value is reached. This outcome is characteristic of all A_+ to A_- ratio smaller than 1:6. Progression of error values throughout training for the opposite situation, a dominance of potentiation, is shown in Fig. 6(c), with an A_+ to A_- of 4:1. In this regime, there is too little depression of weights for connections from input neurons representing less preferred input patterns, eventually resulting in one output region of the map dominating for all input patterns. Wild fluctuations are observed in the error value; this is caused by slightly different neurons, still close to the dominant region, winning the competition for different input patterns. A brief change in winning neuron can result in a temporary large variation in error value. This outcome is characteristic of all A_+ to A_- ratio greater than 1:1.

For the following simulations, an A_+ value of 0.0016 and an A_- value of 0.0055 will be used. These values are situated within the acceptable ratio of these parameters, and represent a low learning rate compared to the maximum acceptable rate.

C. Robustness to Noise

The noise scaling factor g was tested for 11 values between 0.0 and 1.0, with the same value used for both layers u and v, to analyze the robustness of the SOM formation to variable spike times. These values of g resulted in spike time variations in the input pattern of up to around 3 ms. Other network parameters, input data set and input pattern selection remained as described in Section III-A and used in Section III-B. Fig. 7 shows the average E_{MDS} values for training with variation in g. The average values of E_{MDS} ranged from 0.008 to 0.020 for values of g up to 0.5, and between 0.090 and 0.146 for g from 0.6 to 0.9, showing that noise in the neuron model does not prevent the learning mechanism from picking up the statistical correlations present in the input data until a value of g greater than 0.5.

D. 2-D Input

A test often applied to SOM algorithms is to present random samples from a range [0, 1] in two dimensions as input to the network, and test the ability of the output to organize itself into a formation capable of representing this input data.

An average final normalized E_{MDS} value of 0.00554 was achieved, over 64 trials (standard deviation 0.00483). The final weight matrix for the feedforward connections from input to output neurons is shown in Fig. 8; neurons in one row of the output layer are strongly connected to a specific range of input neurons in 1-D, while varying their connection strength to input neurons in the other dimension uniformly across the row (with the inverse pattern seen within and between columns). Fig. 9 shows the output layer neurons as circles, with nearest neighbor connections indicated by connecting lines, positioned in the input space according to the input value to which they respond most quickly (i.e., are best tuned to); this preferential tuning is initially random, and throughout training organizes to mirror the inputs received. It is worth noting that Fig. 9(c) and (d) both show the network state after a good mapping has been achieved, but the position of the neurons is quite variable; this is because the network is learning at the same rate throughout training in this example, and will morph slightly depending on the distribution of the most recent input data.

The spiking SOM is also capable of generating a map representation skewed to fit the input distribution. To test this the map was trained using six distributions of 2-D input data. In each distribution, the likelihood of either dimension being drawn from the range [0, 0.5] was altered to a value in the range of [0, 0.5] at increments of 0.1. A likelihood of 0.5 represents an even distribution across the 2-D space, and a likelihood of 0 represents a distribution entirely in the quadrant of the 2-D space between 0.5 and 1.0 in both dimensions.

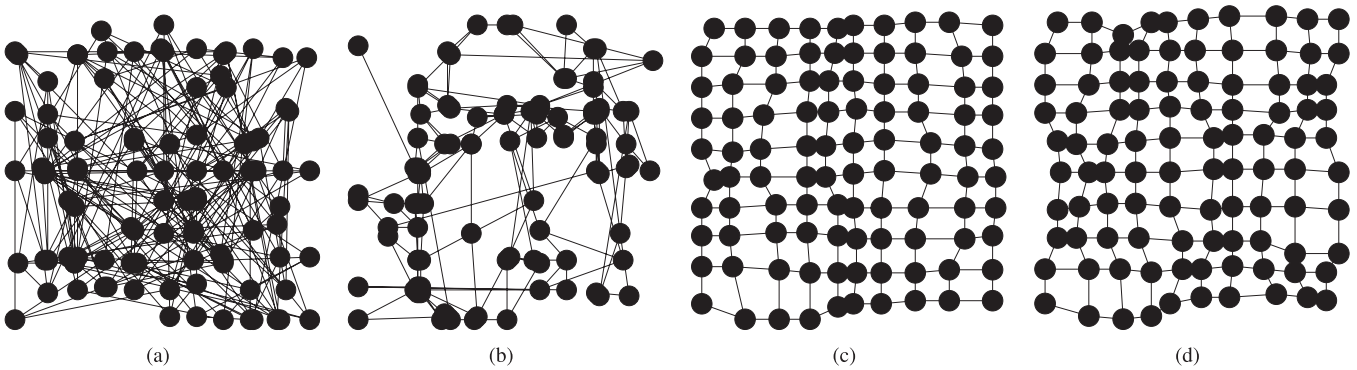

Fig. 9. Representation of u-v feedforward weights in the 2-D input space, where black dots represent output layer neurons and black lines represent nearest neighbor synaptic connections (input and output dimensions are toroidal, but nearest neighbor edges to opposite sides of the input space are omitted for clarity of the figure). (a) Random starting distribution; after 200 training steps. (b) Nodes begin to align to the input data. (c) and (d) Trained map after 2800 and 3600 training steps, respectively—learning is ongoing, so stochastic fluctuations in the distribution of recent input patterns are reflected by minor modulations in the map weights.

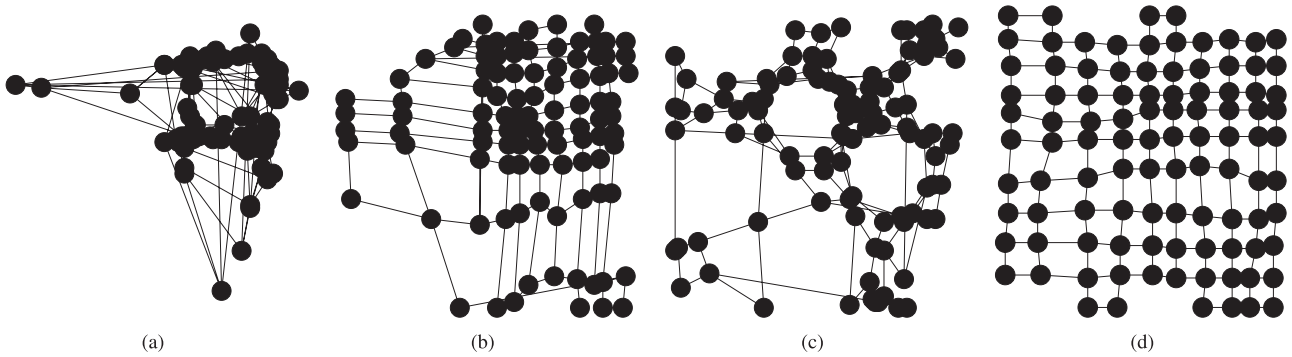

Fig. 10. Representation of u-v feedforward weights in the 2-D input space, where black dots represent output layer neurons and black lines represent nearest neighbor synaptic connections (input and output dimensions are toroidal, but nearest neighbor edges to opposite sides of the input space are omitted for clarity of the figure; neurons on the map border are alternating from one side of the figure to the other because the toroidal nature of the input space means that input values of 0 and 1 are essentially identical, and the row or column that the neurons are a part of is lined up along the 0–1 divide, with some neurons placed just on one side and some just on the other). (a)–(d) Final mapping for input distributions in which the probability in both dimensions of an input value being between 0 and 0.5 is 0.1, 0.2, 0.3, and 0.4, respectively.

Within each half of each dimension, the distribution is even across the range. The representation of the output nodes in the input space is shown in Fig. 10, as in Fig. 9. In Fig. 10(a)–(d), the input distribution can effectively be split into quadrants of likelihood. At any training step the input pattern is least likely to be selected from the lower-left quadrant (between 0 and 0.5 in both dimensions), most likely to be selected from the upper-right quadrant (between 0.5 and 1 in both dimensions), with each of the remaining quadrants at an intermediate likelihood (between 0 and 0.5 in only one dimension). Discontinuities can be observed in the map representation in Fig. 10(d); these are at the boundaries of the toroidal space, and are actually adjacent such that the extended sections on the top are interlocked with the gaps at the bottom. Fig. 11 shows that map formation on average results in good representations of input distributions by the proportion of nodes in the final output mapping that represent a quadrant of the input space.

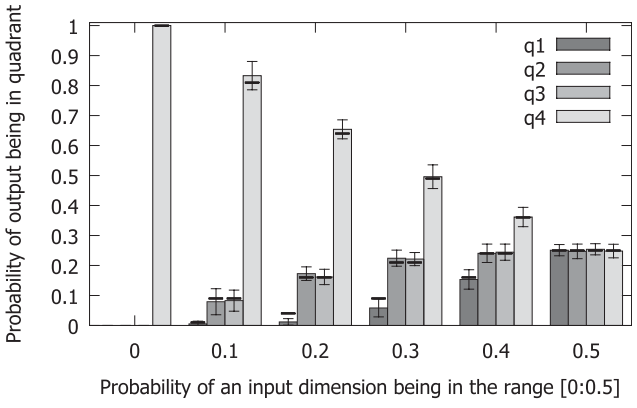

Fig. 11. Probability of an output node in a trained map representing each quadrant of the input space, plotted against the likelihood of an input dimension having a value within the range [0, 0.5]. Bar height: average over 30 trials. Error bars: standard deviation. Black horizontal marks: actual probability of an input datum being within a quadrant of the input space. q1, q2, q3, and q4: lower left, upper left, lower right, and upper right quadrants, respectively.

Another feature of the mapping shown by some SOM algorithms that attempt continuous learning is the ability of the network to reconfigure to a new input distribution midway through training, after a mapping has been established to an existing input distribution. This was tested by training the network using only 75% of the input data space, leaving out the quarter of the input space square covered by values of greater than 0.5 in both dimensions. This reduced data set

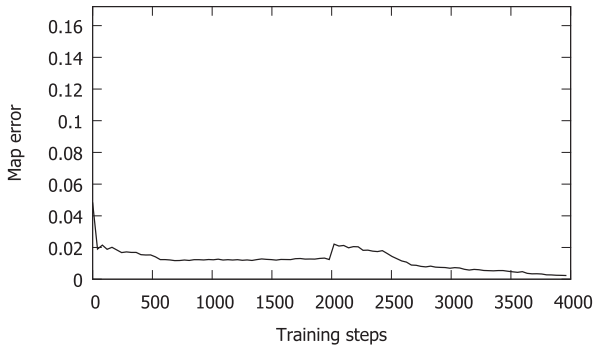

Fig. 12. Typical evolution of average E_{MDS} value for 2-D input: the error in the output map reduces up to the halfway point in training, at which point the input distribution is expanded; the error in the map jumps up as the error measure is now relative to the new distribution; the error then decreases again as the map adjusts to the new distribution.

was used for 2000 training steps, then the full range of input data, including the previously omitted quarter, was used for a further 2000 training steps.

The evolution of the topographic error (Fig. 12) shows that the network adjusts to the initial input range as normal, but settles at a slightly higher E_{MDS} value, most likely due to the discrepancy between the shape of input space and shape of lateral connections in the output layer. The extra data is introduced half way through training, resulting in a spike in E_{MDS} value, as the output mapping is no longer suitable for the input data, and the error is then reduced to a lower value as the output map reorganizes to the new data. The mapping of the output nodes in the input space is shown in Fig. 13. The output nodes map to the original input space during the first half of training, and reorganize in the second half such that the final mapping is qualitatively identical to that observed in Fig. 9. The final weight matrix for the feedforward synapses is also qualitatively identical to the one obtained when training using the full data range from the start (Fig. 8). This simulation demonstrates that the map has learnt to represent the initial input distribution, but when a new distribution is presented the map is capable of adjusting appropriately.

E. Categorization

SOMs can cluster input patterns, creating a specific spatial location that is activated by incoming members of a specific category. If there are category divisions in the input data, nodes in the output layer will respond more reliably to one category of input than to others. A trained SOM can therefore be used as a categorization tool by assigning each output node a category to represent based on whether that node fires reliably for one particular category. The capacity of the spiking SOM to be used as a categorizer in this way has been tested through training with two datasets commonly used for assessing the categorization ability of a system. This capacity is demonstrated to provide evidence that organization to datasets that contain relatively distinct categories within a high-dimensional space is possible using this network, alongside representations of less discrete input data.

In the examples in Sections III-F and III-G five-way cross validation is performed: a dataset is split into five chunks and the network is initialized and trained five times, using a different set of four chunks as training data and one chunk as testing data each time, so that in total all data points are used for testing once. Each training phase lasted for 4000 training steps with random selection of input pattern after each five oscillations of the network, as described in Section III-A. At the end of a training phase, the output nodes were designated as representing a category based on the input category to which they responded most frequently during training. The testing patterns were then presented to the trained network one at a time, and the output activity recorded. Lateral connections were still used in this testing phase, so multiple neurons in the output layer fired for each testing pattern, in an area with width determined by the lateral connection profile at the end of the training phase. The testing pattern was categorized by the network as belonging to the category to which the highest number of output neurons firing in response to that input pattern had been designated as representing. If more of the output neurons firing in response to a testing pattern had been designated with that pattern's category during the training phase than any other single category, then the pattern was considered to have been correctly classified.

F. Iris Dataset

The first dataset used to test the categorization performance of the spiking SOM was the Iris dataset [38]. This dataset is of sizes of flowers of the Iris plant; it consists of three categories each with 50 members, and each data point has four values, petal length, petal width, sepal length, and sepal width. One of these categories Iris Setosa Canadensis, is fairly distinct from the other two, Iris Virginica and Iris Versicolor.

For the current purpose, the values for each dimension were normalized in the range [0–1], and the spiking SOM model was initialized as in Section III-D but with 40 input neurons, making one bank of 10 for each dimension. This increase in number of input neurons meant that the feedforward connection strength from layer u to layer v had to be reduced; the value of $w_{\max}^{u\ \mathrm{to}\ v}$ was scaled down to 1.5. Remaining parameters remained identical, with the exception of the X and X' values in (7), controlling the evolution of the width of the neighborhood function. For continuous learning these values are identical, meaning no change in lateral connection strengths over time. The standard neighborhood parameters were used as one condition in the categorization performance tests ($X = 3.0$ and $X' = 3.0$). However, output map quality can potentially be improved by starting with a large neighborhood and reducing it throughout training. This approach can obtain a globally ordered topology initially, and refine details later on. This regime was used as a second condition in the categorization tests, with X set to 4.0 and X' to 2.5. These values are in numbers of neurons, so for a 10×10 map a radius of greater than 4.0 is the majority of the map.

Categorization accuracy, averaged over 9 trials, was 87.8% (standard deviation = 1.3%) in the without-NR condition, and 90.9% (standard deviation = 1.7%) in the with-NR condition. Table II shows these results in comparison with the results achieved for other categorization algorithms using this dataset. The spiking SOM categorizes better than MATLAB

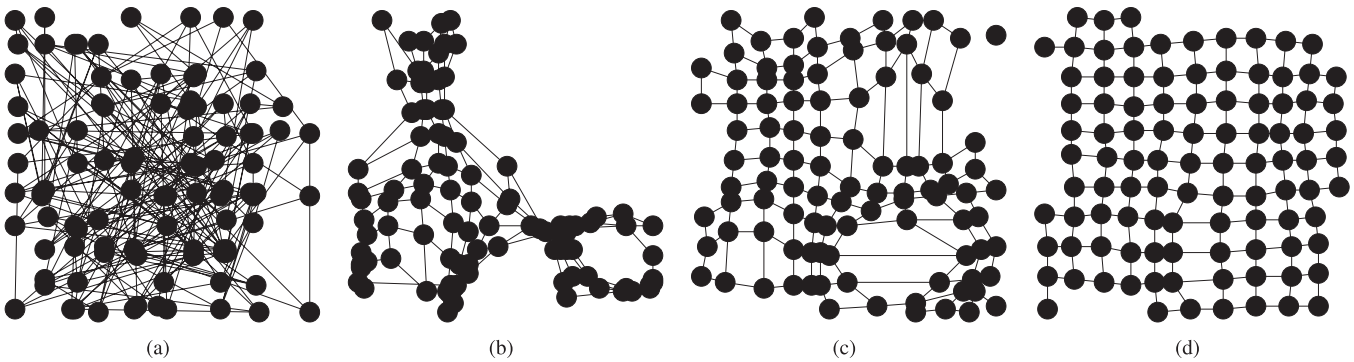

Fig. 13. Representation of u-v feedforward weights in the 2-D input space (arranged as per Fig. 9). (a) Random starting distribution; after 8000 training steps with the partial input data. (b) Nodes are aligned to the input data space, leaving a gap in the input space from which no training examples have yet been received. (c) Expansion of the map to the newly increased range of input data after 800 training steps with the full input distribution. (d) Trained map after 2000 training steps with the full data range—the map has adjusted to the new input data.

TABLE II
CATEGORIZATION ACCURACY (%) OF SPIKING NEURON AND NONSPIKING NEURON ALGORITHMS FOR THE IRIS DATASET. THE CURRENT APPROACH IS SHOWN AS SPIKING SOM WITH NEIGHBORHOOD REDUCTION (NR) AND SPIKING SOM WITHOUT NR

Non-spiking	% acc.	Spiking	% acc.
k-Means [30]	88.6	RBF [30]	92.6
SOM [30]	85.33	SpikeProp [39]	96.1
Matlab BP [40]	95.5	SWAT [40]	95.3
Matlab LM [40]	95.7	RBF [41]	89
TEST [42]	91.7	SNN_{Bako} [43]	83.4
		Spiking SOM with NR	90.9
		Spiking SOM without NR	87.8

implementations of the k-means and SOM algorithms [30] (although parameters used and the extent of parameter searching conducted to achieve these results are unclear), the field-programmable gate array (FPGA) implemented classification network of [43], and the spiking neuron radial basis function (RBF) network of [41].

Categorization performance is slightly worse than several other networks: the spiking RBF model of [30], the SpikeProp model of [39], the synaptic weight associations training algorithm model of [40], the training-estimation-training algorithm [42], and MATLAB implementations of the backpropagation and Levenberg–Marquardt training algorithms [40]; it is worth noting, however, that these are designed specifically for data classification purposes, and do not feature the topographical ordering properties of the SOM model.

G. Wisconsin Breast Cancer Dataset (WBCD)

The second dataset used for categorization testing was the WBCD, consisting of 683 samples from two categories (444 benign and 239 malignant tumors), with nine measures of features of cytology. Each of the nine measures is a discrete value from 1–10, converted into a value in the range 0:1 for the current purpose, and represented using a bank of 10 input neurons, meaning that layer u consisted of 90 neurons. As such, the value of $w_{\max}^{u\text{ to }v}$ is scaled down to 0.7, the value of $w_{\max}^{u\text{ to }Inhu}$ to 0.4, and the X and X' values were adjusted to 3.5 in the without-NR condition, to account for only having two categories occupying the 10×10 output map; all other parameters remained identical to those used in Section III-F.

TABLE III
CATEGORIZATION ACCURACY (%) OF SPIKING NEURON AND NONSPIKING NEURON ALGORITHMS FOR THE WBC DATASET

Non-spiking	% acc.	Spiking	% acc.
Matlab BP [40]	96.3	SpikeProp [39]	97
Matlab LM [40]	96.7	SWAT [40]	95.3
		SNN_{Bako} [43]	89.5
		Spiking SOM with NR	96
		Spiking SOM without NR	97

Categorization accuracy, averaged over eight trials, was 96.4% (standard deviation = 0.4%) in the without-NR condition and 97.0% (standard deviation = 0.1%) in the with-NR condition. Again, these results are compared with the categorization accuracy of other algorithms using this dataset, shown in Table III. The spiking SOM again outperforms the FPGA categorization algorithm implementation of [43], and achieves a very similar level of accuracy to models that have been designed specifically for clustering and categorization operations.

IV. DISCUSSION

The spiking neuron SOM model implemented here has been demonstrated to produce qualitatively and quantitatively similar output to the traditional SOM algorithm. This implementation combines continuously presented input, regular oscillatory firing, phase coding of input values, α-function PSPs, leaky integrate-and-fire neurons, and STDP. The current model exhibits good categorization performance for generic datasets without resorting to additional fine-tuning of parameters. Furthermore, it can function similarly to the traditional SOM algorithm without necessarily decaying learning rate or neighborhood size throughout training. This section contains a set of comparisons of the current model with related existing approaches (Section IV-A), and a discussion of some novel

aspects of the mechanisms used, and limitations of those within the current model (Section IV-B).

A. Related Approaches

The current model improves on the spiking SOM model of [26], described in Section I, in several significant ways. The incorporation of continuous input and oscillatory firing means that the current network does not need resetting. The change to STDP for learning and introduction of α-function PSPs both contribute to an improvement in the biological plausibility and performance of the model. Additionally, testing has established the robustness of the new approach to changes in the learning parameters, noise, and input data.

Other SOM-like networks have been implemented using spiking neurons. In [44], a two layer SOM structure similar to [26] made up of MacGregor neurons [45] is used to test a pair of Hebbian learning rules, one with learning based on strength of PSPs and the other based on temporal correlations. It is demonstrated that, with an appropriate lateral connection neighborhood, either of these learning approaches can result in output space segregation that is related to properties of the input space. However, properties of the conventional SOM such as smooth mapping of input to output space and categorization are not demonstrated. In addition, this model does not process continuous input, encode specific input values in temporal sequences, or use oscillatory behavior.

In [46], a three-layer feedforward network of integrate-and-fire neurons with a STDP-like LTP window is used to produce a self-organized map of orientation preference, given appropriate receptive field shape and input properties. This map does not feature lateral connections or direct competition between the neurons, instead relying on those receptive fields and input properties for the self-organization to occur. As such, it is likely that an organized output map will only result from a limited range of inputs. The current approach can produce an output map based on organization to input data with any properties, due to the use of lateral connections for competition.

The laterally interconnected synergetically self-organizing map model has been modified to incorporate spiking neuron properties in [47], with Hebbian learning dependent on average activity rather than spike timing. This learning method is less biologically relevant than the STDP used in the current model, and using a network structure based on the visual system means that the map is not necessarily capable of mapping a wide variety of input data as in the current model and the conventional SOM.

In other recent work, a pair of self-organizing models have been presented that learn spatiomotor [48] and visual [49] representations using leaky integrate-and-fire neurons and STDP. Again, these models do not tackle the general problem addressed by the current model, but they are based on a shared core model, which can be compared with the current method. This underlying model features Gaussian tuning of responsiveness to input properties, but using a firing-rate rather than spike-timing encoding. This leads to a slightly different role for STDP, in that it picks average pre and postsynaptic combinations out of noisier activity, as opposed to being a method for storing phase-of-firing relationships in the current model. The visual representation model [49] has separate STDP rules for excitatory and inhibitory synapses in the map layer. The inhibitory plasticity is found to be crucial for emergence of the representations, while preventing recurrent excitation from increasing firing rates. The current model avoids this through control of the relationship between maximum synaptic strength and number of neurons, but this method warrants investigation for an automatic control of these properties in SOMs of different scales.

Each of these models suggest ways in which spiking neuron models can facilitate self-organizational network properties, but none of them represents a solution that incorporates the ideas of temporal coding of input data through relative phases of spikes within oscillations, continuous presentation of input data points to the network, and learning via STDP simultaneously. In addition, none of these models present a network structure possessing self-organizational and classification properties comparable with the traditional SOM algorithm.

B. Mechanisms, Limitations, and Future Directions

The inhibitory mechanism introduced to produce oscillatory firing with phase-of-firing coding from a continuously presented input activity level (see Section II-C.1) is a versatile and useful neural function in its own right. A drawback is that it is necessary to predetermine the required size of a gap in the input pattern that will allow the inhibitory neuron to fire and the length and magnitude of the inhibition, which controls the rate of oscillation, by setting PSP values in (2) and (3). These values do not require rigorous fine-tuning provided the range of input activity levels across the pool of input neurons is relatively low, such that all neurons in the pattern will fire within a restricted time, followed by a gap before the neuron with the strongest input will fire again. In addition, it is also necessary to predetermine the strength of feedforward connections between the layers when the number of input neurons changes. Currently, there is no general method for deriving maximum weight values for different network scenarios, and the development of such a method would represent an important step toward a generalized spiking SOM.

Crucial future research will revolve around establishing more precisely the extent of the qualities of self-organization that the spiking SOM possesses. It is currently unclear exactly how well the network can reorganize to shifting, nonstationary inputs, either varying distributions of a continuous multidimensional space, or in the form of correct incorporation and classification of additional input categories introduced after the network has been allowed to learn for some time. It can be speculated that the network will respond well to these challenges, a claim supported by the impressive ability of the network to accurately model input distributions and to adjust to one change in input data during training, results demonstrated in Section III-D. A more detailed comparison of the capabilities of the current network with the capabilities

of the conventional SOM would also require the introduction of analytical techniques that can assess more features of the mapping than the topographic representation. Measures such as quantization error can assess whether a minimum distance from input patterns to their respective output layer representatives is achieved, and integrated measures, such as CQoCO, incorporate the extent to which nodes map regions from outside of the desired input space and the twistedness of the representation as it untangles throughout training into the quality of mapping metric [36].

Additional future research could also improve on the biological plausibility of the network structure and connection profile by basing these on knowledge of cortical areas. The current model is not based on a specific brain region, but the all-to-all connectivity used between the layers is not observed in cortex, so represents a deviation from the way in which cortical functions are generated. As introduced in Section IV-A, other work on self-organization with spiking neurons has focused on generating maps using specific brain regions as inspiration. A synthesis of those approaches with the current one could lead to biologically plausible models with increased functionality and versatility in the future.

The use of phase-of-firing coding and an STDP learning rule has some limitations in the current context. Synaptic connections from neurons involved in the input pattern are weakened if the neuron fires late within the temporal sequence (i.e., after firing in the output layer). However, a neuron with much lower input activity would not fire at all for the current input; with STDP learning, synaptic connections from such neurons to the input layer are not weakened. This problem is not relevant if the input neurons that have connections to an output map represent a narrow domain of actual input stimuli, in which case any stimulus from the associated domain presented to that bank of input neurons would generate some relatively high activity level (and therefore firing) for the entire bank of neurons. In addition, if output neurons have connections from input neurons that never fire in correlation with the dominant banks of input neurons connected to an output neuron, there is a chance that spontaneous activity would gradually weaken the connection, although the current model does not support this.

A second problem with the use of phase coded input is that hierarchical layers of these SOMs are not possible. Output representation is spatial, within a relatively narrow temporal window. The same learning rule will not work if the SOM output is taken as input to a downstream SOM layer. Spreading the firing in the SOM layer into a temporal code (e.g., by adjusting synaptic time constants) would cause the self-organization to fail, as neighboring neurons in the output layer need to fire in close temporal proximity to ensure their weights are adjusted toward the same point. Of course, multiple independent SOMs could be set up with a temporal sequence between the SOMs, which could become input to a downstream SOM layer, creating a multimodal association/integration of spatially coded features.

This type of spiking neuron network has the potential to be used to explore the connectivity and learning mechanisms involved in formation of networks analogous to cortical maps that display topological organizational structures, in artificial intelligence mechanisms that perform tasks like clustering, categorization and concept formation in a biologically plausible manner, and in the development of spiking neuron hardware that physically represents neurons and networks in digital or analogue circuits.

ACKNOWLEDGMENT

This work has been supported by the Engineering and Physical Sciences Research Council of the United Kingdom (EPSRC) under grants EP/C010841/1 (“COLAMN: A Novel Computing Architecture for Cognitive Systems based on the Laminar Microcircuitry of the Neocortex”) and EP/J004561/1 (“BABEL: Bio-inspired Architecture for Brain Embodied Language”).

REFERENCES

[1] J. H. Kaas and K. C. Catania, “How do features of sensory representations develop?” *BioEssays*, vol. 24, no. 4, pp. 334–343, 2002.
[2] B. A. Wandell, “The neurobiological basis of seeing words,” *Ann. New York Acad. Sci.*, vol. 1224, no. 1, pp. 63–80, 2011.
[3] V. A. Casagrande and J. H. Kaas, “The afferent, intrinsic, and efferent connections of primary visual vortex in primates,” in *Primary Visual Cortex in Primates* (Cerebral Cortex), vol. 10. New York, NY, USA: Springer-Verlag, 1994, pp. 201–259.
[4] J. C. Horton and D. R. Hocking, “Anatomical demonstration of ocular dominance columns in striate cortex of the squirrel monkey,” *J. Neurosci.*, vol. 16, no. 17, pp. 5510–5122, 1996.
[5] G. G. Blasdel and G. Salama, “Voltage-sensitive dyes reveal a modular organization in monkey striate cortex,” *Nature*, vol. 321, pp. 579–585, Jun. 1986.
[6] D. H. Hubel, T. N. Wiesel, and M. P. Stryker, “Anatomical demonstration of orientation columns in macaque monkey,” *J. Comparat. Neurol.*, vol. 177, pp. 361–380, Feb. 1978.
[7] M. M. Merzenich, P. L. Knight, and G. L. Roth, “Representation of cochlea within primary auditory cortex in the cat,” *J. Neurophysiol.*, vol. 38, no. 2, pp. 231–249, 1975.
[8] X. Chen, M. Gabitto, Y. Peng, N. J. P. Ryba, and C. S. Zuker, “A gustotopic map of taste qualities in the mammalian brain,” *Science*, vol. 333, pp. 1262–1266, Sep. 2011.
[9] F. Wang, A. Nemes, M. Mendelsohn, and R. Axel, “Odorant receptors govern the formation of a precise topographic map,” *Cell*, vol. 93, no. 1, pp. 47–60, 1998.
[10] T. A. Woolsey, C. Welker, and R. H. Schwartz, “Comparative anatomical studies of the SmL face cortex with special reference to the occurrence of ‘barrels’ in layer IV,” *J. Comparat. Neurol.*, vol. 164, no. 1, pp. 79–94, 1975.
[11] K. Fox, *Barrel Cortex*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[12] R. M. Friedman, L. M. Chen, and A. W. Roe, “Modality maps within primate somatosensory cortex,” *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 34, pp. 12724–12729, 2004.
[13] D. E. Feldman and M. Brecht, “Map plasticity in somatosensory cortex,” *Science*, vol. 310, pp. 810–815, Nov. 2005.
[14] M. M. Merzenich, J. H. Kaas, J. Wall, R. J. Nelson, M. Sur, and D. Felleman, “Topographic reorganization of somatosensory cortical areas 3b and 1 in adult monkeys following restricted deafferentation,” *Neuroscience*, vol. 8, no. 1, pp. 33–55, 1983.
[15] M. B. Calford, “Dynamic representational plasticity in sensory cortex,” *Neuroscience*, vol. 111, no. 4, pp. 709–738, 2002.
[16] T. Kohonen, *Self-Organizing Maps*, 3rd ed. New York, NY, USA: Springer-Verlag, 2001.
[17] J. Hertz, A. S. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*, 1st ed. New York, NY, USA: Perseus, 1991.
[18] E. T. Rolls and G. Deco, *Computational Neuroscience of Vision*. Oxford, U.K.: Oxford Univ. Press, 2002.
[19] N. Manukyan, M. J. Eppstein, and D. M. Rizzo, “Data-driven cluster reinforcement and visualization in sparsely-matched self-organizing maps,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 5, pp. 846–852, May 2012.

[20] C.-C. Hsu and S.-H. Lin, "Visualized analysis of mixed numeric and categorical data via extended self-organizing map," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 1, pp. 72–86, Jan. 2012.
[21] P. Maldonado, C. Babul, W. Singer, E. Rodriguez, D. Berger, and S. Grün, "Synchronization of neuronal responses in primary visual cortex of monkeys viewing natural images," *J. Neurophysiol.*, vol. 100, no. 3, pp. 1523–1532, 2008.
[22] P. Fries, D. Nikolić, and W. Singer, "The gamma cycle," *Trends in Neurosci.*, vol. 30, no. 7, pp. 309–316, 2007.
[23] S. J. Thorpe, D. Fize, and C. Marlot, "Speed of processing in the human visual system," *Nature*, vol. 381, no. 6582, pp. 520–522, 1996.
[24] R. Van Rullen and S. J. Thorpe, "Rate coding versus temporal order coding: What the retinal ganglion cells tell the visual cortex," *Neural Comput.*, vol. 13, no. 6, pp. 1255–83, 2001.
[25] T. K. Hensch, "Critical period regulation," *Annu. Rev. Neurosci.*, vol. 27, pp. 549–579, Jul. 2004.
[26] B. Ruf and M. Schmitt, "Self-organization of spiking neurons using action potential timing," *IEEE Trans. Neural Netw.*, vol. 9, no. 3, pp. 575–578, May 1998.
[27] S. Song and L. F. Abbott, "Cortical development and remapping through spike timing-dependent plasticity," *Neuron*, vol. 32, no. 2, pp. 1–20, 2001.
[28] S. Song, K. D. Miller, and L. F. Abbott, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neurosci.*, vol. 3, no. 9, pp. 919–926, 2000.
[29] D. Bush, A. Philippides, P. Husbands, and M. O'Shea, "Reconciling the STDP and BCM models of synaptic plasticity in a spiking recurrent neural network," *Neural Comput.*, vol. 22, no. 8, pp. 2059–2085, 2010.
[30] S. M. Bohte, H. La Poutré, and J. N. Kok, "Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer RBF networks," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 426–435, Mar. 2002.
[31] X.-J. Wang, "Neurophysiological and computational principles of cortical rhythms in cognition," *Physiol. Rev.*, vol. 90, no. 3, pp. 1195–1268, 2010.
[32] J. S. Law, "Modeling the development of organization for orientation preference in primary visual cortex," Ph.D. dissertation, School of Informatics, Univ. Edinburgh, Edinburgh, Scotland, 2009.
[33] R. Keith-Magee, "Learning and development in Kohonen-style self-organising maps," Ph.D. dissertation, School of Computing, Curtin Univ. Technol., Bentley, Australia, 2001.
[34] G. J. Goodhill and T. J. Sejnowski, "A unifying objective function for topographic mappings," *Neural Comput.*, vol. 9, no. 6, pp. 1291–1303, 1997.
[35] D. Vidaurre and J. Muruzábal, "A quick assessment of topology preservation for SOM structures," *IEEE Trans. Neural Netw.*, vol. 18, no. 5, pp. 1524–1528, Sep. 2007.
[36] D. Beaton, I. Valova, and D. Maclean, "CQoCO: A measure for comparative quality of coverage and organization for self-organizing maps," *Neurocomputing*, vol. 73, nos. 10–12, pp. 2147–2159, 2010.
[37] D. Polani, "Measures for the organization of self-organizing maps," in *Self-Organizing Neural Networks*, U. Seiffert and L. C. Jain, Eds. Berlin, Germany: Springer-Verlag, 2002, pp. 13–44.
[38] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugen.*, vol. 7, no. 2, pp. 179–188, 1936.
[39] Q. Wu, T. McGinnity, L. Maguire, B. Glackin, and A. Belatreche, "Learning under weight constraints in networks of temporal encoding spiking neurons," *Neurocomputing*, vol. 69, nos. 16–18, pp. 1912–1922, 2006.
[40] J. Wade, L. McDaid, J. Santos, and H. Sayers, "SWAT: A spiking neural network training algorithm for classification problems," *IEEE Trans. Neural Netw.*, vol. 21, no. 11, pp. 1817–1830, Nov. 2010.
[41] N. Gueorguieva, I. Valova, and G. Georgiev, "Learning and data clustering with an RBF-based spiking neuron network," *J. Experim. Theoretical Artif. Intell.*, vol. 18, no. 1, pp. 73–86, 2006.
[42] S.-Y. Yoon and S.-Y. Lee, "Training algorithm with incomplete data for feed-forward neural networks," *Neural Process. Lett.*, vol. 10, no. 3, pp. 171–179, 1999.
[43] L. Bako, "Real-time classification of datasets with hardware embedded neuromorphic neural networks," *Briefings Bioinformat.*, vol. 11, no. 3, pp. 348–363, 2010.
[44] D. M. Sala, K. J. Cios, and J. T. Wall, "Self-organization in networks of spiking neurons," *Austral. J. Intell. Inf. Process. Syst.*, vol. 5, no. 3, pp. 161–170, 1998.
[45] R. J. MacGregor, *Neural and Brain Modeling*. San Francisco, CA, USA: Academic, 1987.
[46] F. Veredas, H. Mesa, and L. A. Martínez, "Imprecise correlated activity in self-organizing maps of spiking neurons," *Neural Netw.*, vol. 21, no. 6, pp. 810–816, 2008.
[47] Y. Choe and R. Miikkulainen, "Self-organization and segmentation in a laterally connected orientation map of spiking neurons," *Neurocomputing*, vol. 21, nos. 1–3, pp. 139–157, 1998.
[48] N. Srinivasa and Y. Cho, "Self-organizing spiking neural model for learning fault-tolerant spatio-motor transformations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 10, pp. 1526–1538, Oct. 2012.
[49] N. Srinivasa and Q. Jiang, "Somatosensory cortex: Structural alterations following early injury to sense organs," *Frontiers Comput. Neurosci.*, vol. 7, no. 10, pp. 1–24, 2013.

Timothy Rumbell received the Ph.D. degree from Plymouth University, Plymouth, U.K., in 2013, investigating unsupervised learning of concepts in networks of spiking neurons.

He is currently a Post-Doctoral Fellow with the Mount Sinai School of Medicine, New York, NY, USA, researching the mechanisms underlying cognitive decline with aging using single cell models. His current research interests include the computational properties of neural systems.

Susan L. Denham is a Professor of cognitive neuroscience and she was appointed the director of the Plymouth Cognition Institute, Plymouth, U.K., in 2012. Her research focuses on understanding the computational principles of sensory perception and perceptual organization in the brain using perceptual experiments, computational models, and neuromorphic hardware implementations.

Thomas Wennekers is a Reader in Computational Neuroscience with Plymouth University, Plymouth, U.K. His research aims at an understanding of brain function at a systemic level and its application to future brainlike computing architectures in soft and hardware. His current research interests include cortical sensory and cognitive function.