Binding Sparse Spatiotemporal Patterns In Spiking Computation

Steve K. Esser, Anthony Ndirango and Dharmendra S. Modha

Abstract— **Imagine a two-dimensional spatial array of detectors temporally driven via an unknown number of mutually overlapping, unknown patterns. One at a time, these patterns are randomly, partially, sparsely and repeatedly presented, superimposed with omnipresent noise. The challenge is to design a scheme for detecting and recalling these patterns in an unsupervised, online and computationally efficient fashion.**

As our main contribution, we propose a network of spiking neurons consisting of two reciprocally connected layers. The bottom layer receives stimulus from the detector array and serves as input/output. The top layer encodes, detects and recalls specific patterns. Feedforward projections are data-driven, bottom-up, and analytic, while feedback projections are model-driven, top-down, and synthetic. We judiciously select neuron dynamics and spike-timing dependent synaptic learning rules such that these feedforward and feedback views eventually converge to bind together the spatial extent of each pattern into a coherent, temporary assembly.

We present simulations demonstrating that our system is able to detect repeating patterns in an input stream with an impressive degree of tolerance to noise and pattern characteristics.

I. Introduction

Biological organisms have evolved a diverse set of sensory devices to deal with a wide array of stimulus features in the environment. Somewhat surprisingly, despite the wide variability in the types of sensors, every type of incoming sensory signal is transduced into a universal format, namely, the action potentials of firing neurons: "spikes". Millions of noisy spiking inputs impinge on the brain at a wide variety of spatial and temporal scales. Nonetheless, somehow, the brain integrates these spikes and extracts *relevant* pieces of information – in a completely unsupervised, parallel and distributed fashion – to give rise to cognition. Precisely how the brain achieves this marvelous feat is an outstanding and profound open problem. Towards the long-term goal of unlocking this mystery, we focus on a concrete problem that is difficult to solve via traditional neural network approaches [1], [2], [3], [4], [5], [6], [7], [8], [9], [10] but lends itself naturally to spiking computation.

Imagine a two-dimensional spatial array of detectors observing an environment. A *pattern* is a specific spatial shape corresponding to a subset of the detectors. The spatial detector array is temporally driven via an unknown number of mutually overlapping patterns that are sequentially, randomly, partially, sparsely and repeatedly presented. No side information about a pattern is given, specifically pattern start time, duration present, size and expected shape are unknown. The situation is further confounded by omnipresent, random detector and/or environmental noise.

Steve K. Esser, Anthony Ndirango and Dharmendra S. Modha are at the IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120 (e-mail: {sesser, andiran, dmodha}@us.ibm.com)

An observer viewing this detector array will thus see individual activations occurring rapidly over time, but due to the temporally sparse nature of their occurrence, will never see more than a small fragment of a pattern at a time. Further, the observer has no way to distinguish activations resulting from the presence of a pattern from those due to noise. To discern what stimuli are present, the observer might try simply averaging the activity in the detector array using a sliding time window. However, a problem arises in choosing the averaging window. If too short of a window is used, the stimulus will not be fully captured while if too long a window is used, multiple stimuli will be captured and appear as a single stimulus. Further, any attempt at averaging will be prone to underlying noise, ambiguity about start and stop times of patterns, the unknown number of patterns, and the unknown, random sequence of pattern presentation.

The challenge we set forth is to design a scheme that allows one to (a) extract the spatial structure of the patterns, that is, detect the patterns as presented, maintaining a distinction between different patterns, and setting them apart from the background noise, and (b) store the patterns so that when presented with a fragmentary and/or noisy version of the stored pattern, the system is able to faithfully retrieve the correct version from memory. Furthermore, this task is to be performed under the constraints delineated in the previous paragraph and carried out in a purely *unsupervised* manner and in an exclusively *online* fashion. In particular, *ad hoc* stages during learning, such as separate learning/un-learning stages [11], are not permitted.

In this paper, as our main contribution, we propose a two-layer network of spiking neurons that is designed to stylistically capture the massive recurrence observed in the cerebral cortex. The neurons in the bottom layer receive stimuli from the detector array and also serve as both input and output. Neurons in the top layer, via competitive dynamics, encode specific patterns and serve to detect and recall the patterns. Feedforward projections are data-driven, bottom-up, and analytic, while feedback projections are model-driven, top-down, and synthetic. We judiciously select neuron dynamics and spike-timing dependent synaptic learning rules such that these feedforward and feedback views of the world eventually converge, thereby extracting the presence and characteristics of persistent patterns in the input stream. The interplay between feedforward and feedback processing serves to bind together the spatial extent of each pattern into a coherent, temporary assembly [12], [13] in a completely unsupervised, online, computationally efficient fashion.

We present the network architecture and algorithm in Section II, detailed simulation results in Section III, and discussion in Section IV.

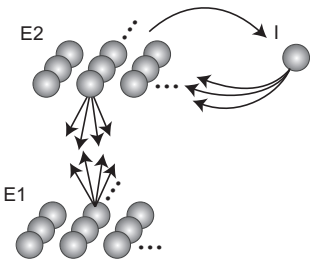

Fig. 1. Schematic representation of the network's architecture with some projections from typical $E1$ and $E2$ neurons and from the inhibitory neuron are shown.

II. Architecture and Algorithm

A. Architecture

As shown in Figure 1, the basic design features two layers – hereafter denoted by $E1$ and $E2$ – each comprising a population of *excitatory* neurons, as well as an additional *inhibitory* neuron, labeled I. We will restrict our attention to *autoassociative* tasks where the $E1$ layer serves as both an input and output layer. As will soon become apparent, the $E2$ neurons serve to primarily identify candidates for membership in putative assemblies, while the inhibitory neuron effectively induces lateral inhibition between the $E2$ neurons, culminating in a potent winner-take-all mechanism. In general, the sizes of the $E1$ and $E2$ populations can be chosen independently, but for simplicity, we will let them contain an equal number, N, of neurons. Each neuron in $E1$ connects to all $E2$ neurons, while each $E2$ neuron connects to all $E1$ neurons as well as making a single connection to the inhibitory neuron I. The synaptic weights of the connections from $E1$ to $E2$ and the connections from $E2$ to $E1$ are plastic. The dynamic synaptic weight update rules will be treated below. Finally, the inhibitory neuron makes a single connection to each $E2$ neuron. We note that no population is allowed to connect back to itself.

The connectivity is designed to stylistically capture the massive recurrence observed in the cerebral cortex within an artificial neural network architecture. We think of $E1 \rightarrow E2$ projections as data-driven, feedforward, bottom-up, analytic, and $E2 \rightarrow E1$ projections as model-driven, feedback, top-down, and synthetic. Our key claim is that these feedforward and feedback views of the world eventually converge via learning.

B. Neuron Dynamics

Our simulations were carried out using conductance-based integrate-and-fire spiking neurons. A key advantage of these neurons is that they can progressively accumulate input, allowing them to respond to patterns that are spread out in time. Furthermore, while input accumulation is gradual, the response spike occurs at a specific time, enabling spike-timing dependent plasticity rules that can produce response selectivity to a specific pattern to the exclusion of background noise and other patterns.

The dynamics of each excitatory spiking neuron, indexed by j, is determined in terms of its depolarization variable V_j via

$$\begin{aligned}
C_0\dot{V}_j(t) &= G_L^+\left[E_L^+ - V_j(t)\right] + I_j^+(t) + I_j^-(t) + I_j^{\text{ext}}(t) \\
I_j^+(t) &\equiv \left[E^+ - V_j(t)\right] \cdot \\
&\qquad \sum_{k\in E1\cup E2} w_{kj}(t) \sum_{n\geq 0} A\left(t - t_k^{(n)}; \tau_1, \tau_2\right) \\
I_j^-(t) &\equiv \left[E^- - V_j(t)\right] w_{Ij} \sum_{n\geq 0} A\left(t - t_k^{(n)}; \tau_3, \tau_4\right) \\
A(s;\tau_a,\tau_b) &\equiv H(s)\frac{e^{-s/\tau_a} - e^{-s/\tau_b}}{e^{-\tau_{ab}/\tau_a} - e^{-\tau_{ab}/\tau_b}}, \\
&\qquad \tau_{ab} \equiv \frac{\tau_a\tau_b}{\tau_a - \tau_b}\ln\left(\frac{\tau_a}{\tau_b}\right). \qquad (1)
\end{aligned}$$

The inhibitory spiking neuron is implemented by replacing G_L^+ with G_L^- and by replacing E_L^+ with E_L^-. I_j^{ext} is the external current, if any, delivered to neuron j, while I_j^+ and I_j^- are the net excitatory and inhibitory synaptic currents impinging on neuron j from other neurons in the network. $A(\cdot;\tau_a,\tau_b)$ denotes the synaptic conductance and H is Heaviside's function. Connections originating from excitatory neurons are fast and are modeled using AMPA-like dynamics, while connections from inhibitory to excitatory neurons are slow and are modeled using GABA$_A$-like dynamics. The variables $\{t_k^{(n)}\}_{n\geq 0}$ represent the discrete spike times associated with neuron k. Finally, w_{kj} refers to the plastic synaptic weight assigned to a synapse shared between a pre-synaptic neuron k and a post-synaptic neuron j and w_{Ij} refers to static synaptic weight from the inhibitory neuron to a post-synaptic neuron j. Three additional parameters, V_{reset}, $V_{\text{threshold}}$ and τ_{refr} are required to round out the description. A neuron j is said to spike as soon as $V_j \geq V_{\text{threshold}}$. Once it spikes, it is instantaneously reset to the value V_{reset}, and subsequently held at that value for a duration of its refractory period τ_{refr}. Parameters in equation (1), together with typical values, are defined in Table I. Most parameter values were set to loosely reflect observations of biological neuron behavior, and then fine-tuned through trial-and-error to achieve desired network performance. We used a 0.25 millisecond interval for updating neuron dynamics.

C. Input Process

The input layer $E1$ consists of N neurons. We define a *pattern* as simply a subset of these N neurons. We suppose that there are M patterns.

The spatiotemporal input stream is generated as follows. A pattern is randomly selected, stays active for d milliseconds, is followed by an inter-pattern interval of 150 milliseconds where no patterns are activated, and then the process is repeated.

When a pattern becomes active, a fraction f of the pattern neurons are randomly selected. Each selected neuron receives superthreshold current $I_\star$ exactly once during the d millisecond window. The timing of current injection is selected randomly over the d millisecond duration for each

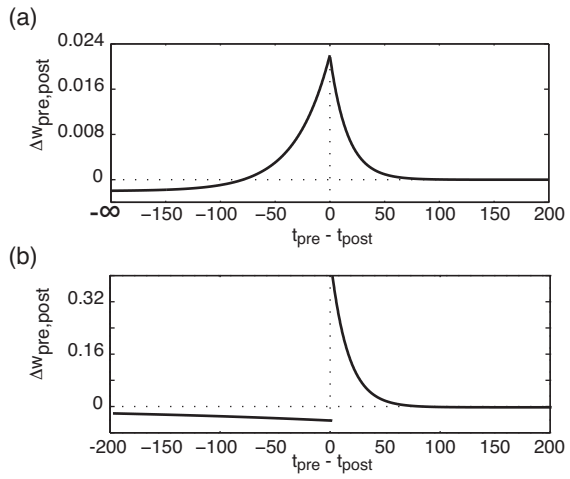

Fig. 2. The spike timing based learning rule governing synaptic plasticity in the network. Changes in synaptic weight are determined by the relative timing of spikes occurring in the post-synaptic and pre-synaptic neurons. Different learning rules govern the connections from $E1$ to $E2$ (a) and from $E2$ to $E1$ (b).

neuron. The input stream also contains an ever present noise component that randomly injects superthreshold current into each of the N neurons with probability a over a millisecond interval.

We define the instantaneous signal to noise ratio, when the pattern is active, as:

$$\mathrm{SNR} = \frac{f \times (\text{pattern size})/d}{N \times a}, \tag{2}$$

and as zero otherwise.

D. Learning Rules

Initially, every synaptic weight from $E2 \rightarrow E1$ is initialized to zero, every synaptic weight from $E1 \rightarrow E2$ is selected uniformly randomly from the interval $[0.02, 0.06]$ and constrained such that the mean value of the synaptic weights into each $E2$ neuron is 0.04. Throughout, every synaptic weight from $E2 \rightarrow I$ is set to 2, and every synaptic weight from $I \rightarrow E2$ is set to 10. Throughout the simulation, every $E1 \rightarrow E2$ synaptic weight is constrained within $[0, 0.4]$ and every $E2 \rightarrow E1$ synaptic weight is constrained within $[0, 4]$. Note that a synaptic weight of $w_{\text{threshold}} = 1.2$ is enough to cause a neuron to fire. Finally, throughout, to guarantee stability, we ensure that the sum of dendritic weights incoming into every $E2$ neuron from $E1$ neurons is maintained at a constant value via weight renormalization. Specifically, for a neuron j in $E2$, let $\{w_{kj}(t)\}_{k \in E1}$ denote the incoming dendritic weights from $E1$ at time t. Let the synaptic change at time t be $\{\Delta w_{kj}(t)\}_{k \in E1}$. Then, for every $k \in E1$, we compute:

$$\begin{aligned} w'_{kj}(t) &= \min[0.4, \max(0, w_{kj}(t) + \Delta w_{kj}(t))] \\ w_{kj}(t+1) &= w'_{kj}(t) \times \frac{\sum_{k \in E1} w_{kj}(t)}{\sum_{k \in E1} w'_{kj}(t)}. \end{aligned}$$

We first describe a generic spike-timing dependent plasticity rule that will be judiciously specialized for feedforward and feedback connections. Every registered pair of pre- and post-synaptic spike events, say at t_{pre} and t_{post}, induces a change, $\Delta w_{\text{pre,post}}$, in the strength of the synapse between the given pair of neurons (labeled here by pre and post), given by

$$\Delta w_{\text{pre,post}} = \begin{cases} A_1 \left[e^{-(t_{\text{post}} - t_{\text{pre}})/\tau_5} - e^{-\tau_7} \right], \\ \quad \text{if } t_{\text{post}} \geq t_{\text{pre}} \\ A_2 \left[e^{-(t_{\text{pre}} - t_{\text{post}})/\tau_6} \right], \\ \quad \text{if } t_{\text{post}} < t_{\text{pre}} \end{cases} \tag{3}$$

where A_1, A_2, τ_5, τ_6, and τ_7 are constants.

For the feedforward learning rule, our general strategy is to detect repeated sequences of spatiotemporal patterns encoded in streams of spikes. Therefore, we will use a conservative learning rules that rewards correlations between spike-pairs that occur within short intervals of time as such pairing are likely due to firing within a pattern, and penalizes any spike-pairs outside the specified window as such pairings are likely due to unrelated underlying events caused by noise. The feedforward rule is displayed in Figure 2(a) and is obtained from (3) by setting:

$$A_1 = 0.024, A_2 = 0.022, \tau_5 = 32, \tau_6 = 16, \tau_7 = 6.21.$$

Turning to the feedback path, if neuron $\alpha \in E2$ fires after a given neuron $j \in E1$, our learning rule assumes that, in a mirror-symmetric world, j might have had something to do with α's firing, and so we increase the strength of the feedback connection $w_{\alpha j}$. If, instead, α fires before j, we penalize the connection. The feedback learning rule, which might be counterintuitive, is imposed in order to discourage the modification of $E1$ activity by $E2$. The feedback rule is displayed in Figure 2(b) and is obtained from (3) by setting:

$$A_1 = -0.04, A_2 = 0.4, \tau_5 = 256, \tau_6 = 16, \tau_7 = \infty.$$

E. Feedforward, Feedback, and Winner-Take-All

Inputs generated in $E1$ are propagated in the form of spikes to neurons in $E2$. $E2$ neurons whose receptive fields (that is, input weights from $E1$) closely match this input stream will respond quickly. Once any $E2$ neuron fires, it triggers the inhibitory neuron, producing inhibition of sufficient strength to reset all $E2$ neurons roughly to their baseline values. Hence, $E2$ neurons staging a strong and fast response to the onset of a particular stimulus will tend to emerge as *winners* [14], [15], [6], [16], [17], and will prevent other neurons in $E2$ from responding to the presented stimulus via a winner-take-all mechanism induced by lateral inhibition. When an $E2$ neuron fires, the synapses associated with the currently active pattern, which is likely to have caused the $E2$ neuron to fire, will be strengthened, thus making that $E2$ neuron more selective for that pattern the next time it occurs, and making it even more likely that it will be a winner again in the context of the specific stimulus encoded in its receptive field. In tandem, the feedback learning rule

TABLE I

PARAMETERS USED IN SIMULATIONS

Parameter	Description	Value (dimensionless units)
C_0	Membrane capacitance	1
G_L^+, G_L^-	Leak conductances	$0.01, 1$
V_{reset}	Reset potential	-65
$V_{\text{threshold}}$	Threshold potential	-51
E_L^+, E_L^-, E^+, E^-	Reversal potentials	$-65, -55, 0, -70$
$I_\star$	Input (superthreshold) current	1000
$\tau_{\text{refr}}^+, \tau_{\text{refr}}^-$	Refractory periods	$50, 1$
$\tau_1, \tau_2, \tau_3, \tau_4$	Decay constants for synaptic conductance channels	$0.5, 2.4, 1.0, 7.0$
$\tau_5^{FF}, \tau_6^{FF}, \tau_7^{FF}$	Feedforward learning rule constants	$32, 16, 6.2$
$\tau_5^{FB}, \tau_6^{FB}, \tau_7^{FB}$	Feedback learning rule constants	$256, 16, \infty$
$A_1^{FF}, A_2^{FF}, A_1^{FB}, A_2^{FB},$	Learning Rule Parameters	$0.024, 0.022, -0.04, 0.4$
$w_{\max}^{FF}, w_{\max}^{FB}$	Feedforward and feedback max weights	$0.4, 4$
$\{w_{jk}\}_{j\in E1, k\in E2}$	Initial $E1$-to-$E2$ weights uniformly distributed	$w_{jk} \in [0.02, 0.06]$
$\{w_{kj}\}_{k\in E2, j\in E1}$	Initial $E2$-to-$E1$ weights	$w_{kj} = 0, \forall j, k$
$\{w_{Ij}\}_{j\in E2}$	Inhibitory-to-Excitatory weights	$w_{Ij} = 10, \forall j \in E2$
$\{w_{jI}\}_{j\in E2}$	Excitatory-to-Inhibitory weights	$w_{jI} = 2, \forall j \in E2$

will favor strengthening of synapses from the winning $E2$ neuron back to recently active $E1$ neurons, which were presumably responsible for triggering that particular $E2$ neuron. Through the interplay between the feedforward and feedback learning rules, the associations between patterns and the strongest corresponding winners in $E2$ will become stronger iteratively. Further, the omnipresent noise component in the input stream will generally cause the feedforward learning to forget or unlearn incidental connections from $E1$ to $E2$ and non-pattern related $E2$ to $E1$ connections. Thus, a particular pattern in $E1$ and a small set of neurons in $E2$ will become mutually associated forming an assembly.

In summary, the winner-take-all mechanism via lateral inhibition, the spike-timing dependent learning rules, dendritic weight normalization, heterogeneous mix of initial weight distributions, and background noise come together to induce the emergence of a minimal number of representative $E2$ neurons selective to particular stimuli.

III. Simulation Results

A. First Experiment

We tested our algorithm's capacity to learn by presenting it with a simple test case of 9 distinct patterns (see Figure 3). Each pattern was composed of 16 $E1$ neurons. Each pattern overlapped with at least three other patterns and one pattern overlapped with all other patterns. Median overlap between two patterns was 4 neurons. Once a pattern became active, it stayed active for $d = 50$ milliseconds. Noise was continually present, producing activations in each $E1$ neuron with a probability of $a = 0.002$ in a given millisecond. In the training phase, signal fraction f was set to 1.0, producing an SNR of 1.6. For testing, we reduced signal fraction to $f = 0.5$, resulting in an SNR of 0.8.

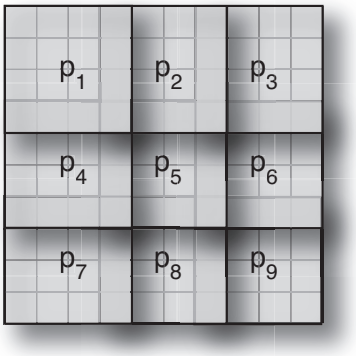

Fig. 3. The network was trained on a set of nine overlapping patterns each of size 16, represented as 4x4 squares evenly distributed across $E1$ (for visualization purposes).

The network parameters are detailed in Table I, with 100 $E1$ and 100 $E2$ neurons. Early in the training phase, $E2$ neuron receptive fields (that is, $E1 \to E2$ synaptic weights) were nonspecific, meaning that any $E2$ neuron could fire in response to a given pattern. However, as training progressed, one $E2$ neuron would begin to fire more frequently in response to a particular pattern, initially by chance and then due to its receptive field becoming more selective for that particular pattern as a result of the learning rule. To quantify such changes, we defined the *receptive field selectivity* of a neuron j in $E2$ neuron with respect to a pattern π as

$$\frac{\sum_{k\in\pi} w_{kj}(t)}{\sum_{k\in E1} w_{kj}(t)}.$$

For each pattern, neurons in $E2$ initially exhibited varying levels of selectivity until about the 50th stimulus presenta-

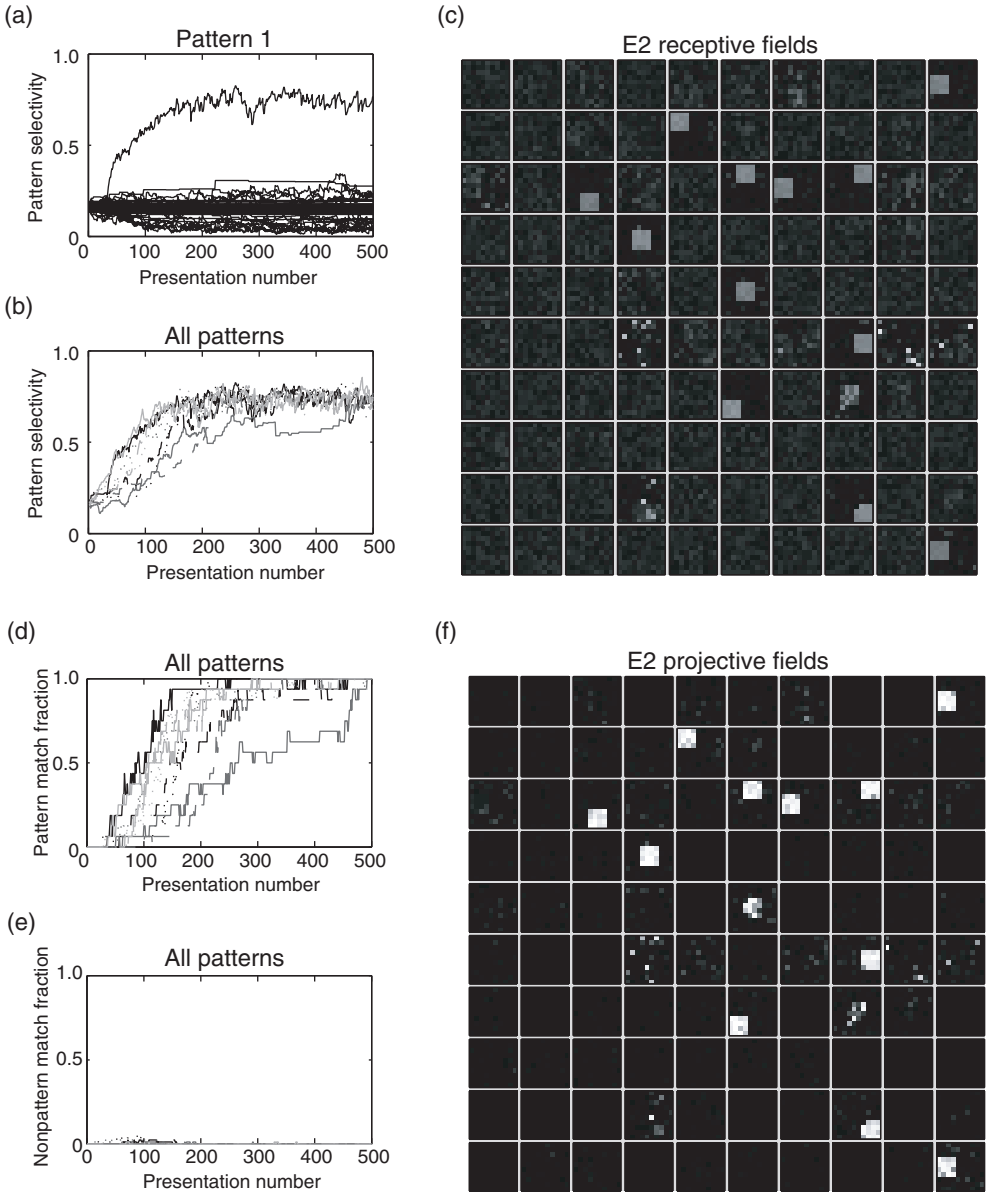

Fig. 4. a) The development over time of each $E2$ neuron's receptive field selectivity for one of the patterns. b) For each pattern, the development of the most selective neurons is shown over time. c) The receptive field of each $E2$ neuron at the end of the 500th presentation training period (weights scaled from 0 to $w_{\max}$). d) The development of pattern selective $E2$ neurons' pattern match fraction in $E1$. e) The development of the pattern selective $E2$ neurons' nonpattern match fraction (that is, false-positive pairings). f) The projective fields of each $E2$ neuron at the end of the 500th presentation training period. It can be seen that the projective and receptive fields of pattern selective $E2$ neurons converge.

tion when a neuron with clear preference began to emerge (Figures 4a and b). At the end of the training phase, a subset of $E2$ neurons became highly selective for single patterns (Figure 4c). At the end of training, we labeled a neuron in $E2$ as selective for a pattern if it had the highest receptive field selectivity for the pattern amongst all $E2$ neurons. These neurons achieved a final mean receptive field selectivity of 0.74 ± 0.03 (mean $\pm$ standard deviation). The presence of noise in the input stream prevented the selectivity from reaching 1.0, but neuron thresholds are set such that pattern detection happens quite reliably with selectivity less than 1 (see Figure 5 and accompanying text below).

As certain $E2$ neurons developed receptive field selectivity for the patterns in the input, the projective field of the $E2$ neurons (that is, the weights from those $E2$ neurons back to $E1$) began to match the pattern such that when a pattern selective $E2$ neuron fired, weights were strong enough (that is, greater than $w_{\text{threshold}}$) to produce firing in $E1$ neurons involved in the pattern. For a neuron j in $E2$ that is selective for a pattern π, we defined the *pattern match fraction* as the fraction of pattern neurons in $E1$ that receive synaptic weight greater than $w_{\text{threshold}}$. Conversely, we defined the *non-pattern match fraction* as the fraction of non-pattern neurons in $E1$ that receive synaptic weight greater than

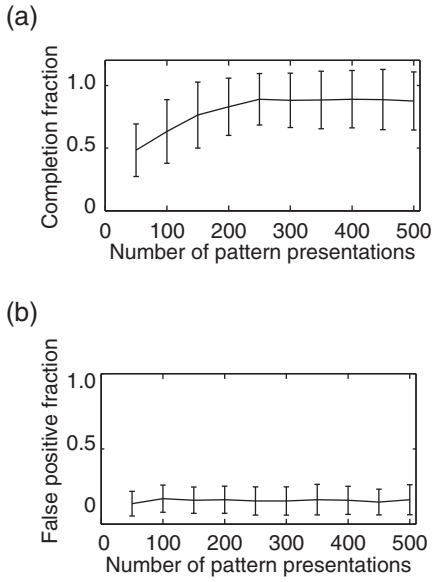

Fig. 5. Pattern completion in the network. a) The completion fraction (fraction of pattern completed by feedback from $E2$). b) The false positive fraction (number of spikes produced in $E1$ outside of the pattern due to input form $E2$).

$w_{\text{threshold}}$. We found that the pattern match fraction reached levels of 0.99 $\pm$ 0.02 following training (Figure 4d), while the non-pattern match fraction for all selective neurons was zero following training (Figure 4e). The high level of pattern matching in the $E2$ projective fields following training can be seen in Figure 4f.

The changes to synaptic weights described above provide a clear measure of the learning occurring in the system. In particular, the *observation that the receptive and projective fields of $E2$ neurons converged over the course of training* (as can be seen in Figure 4c and f) provides a strong demonstration of the formation of pattern specific assemblies. This crucial point provides a reassuring validation of the predictions made for our learning algorithm. To test such assemblies, we also sought to measure the system's ability to perform pattern completion. Following every 50 presentations of all patterns in the training session we switched to a testing phase where we locked synaptic weights at their current values and then presented half of each pattern ($f = 0.5$) 50 times, intermingled with the same level of background noise used in training. To assess pattern recall, we created a measure called the *completion fraction*. For each pattern presentation, this measure equalled the maximum fraction of pattern neurons in $E1$ coactivated within a short time window (defined as being simultaneously in their refractory period) out of those not already activated due to noise. Conversely, we also measured what portion of non-pattern $E1$ neurons were activated by the firing of an $E2$ neuron (the false positive fraction). We found that pattern completion improved over the course of training, achieving a completion fraction of 0.93 $\pm$ 0.13 with a false positive fraction of 0.16 $\pm$ 0.08 (Figure 5).

B. Robustness to Input Characteristics

To test the network's robustness to variations in the characteristics of the input patterns, we focused on the completion fraction and false positive fraction as performance metrics since these provide a direct indication of the network's success at pattern learning and recall. These measures were recorded in 50 test presentations of each pattern following 500 training presentations. We used the same network and stimulus parameters as in the above simulations, modified as described for each test case.

We tested how changes to pattern duration affected network performance by progressively increasing the temporal spread of the stimuli, d, across a series of simulations. Of the pattern durations tested, we found that the network was best at detecting patterns spread across 50 ms, but remained quite capable of detecting and completing patterns with durations between 10 and 100 ms (performance fell by at most 18% from its peak value for these cases). The network's ability to perform pattern completion began to drop off for patterns longer than 100 ms (Figure 6a).

It is not surprising that performance fell with longer pattern duration, as the learning rules look for spikes within restricted temporal windows, defined by τ_5, τ_6 and τ_7. To test whether the network could be tuned to improve responses to longer stimuli, we increased τ_5 and τ_6 by 4x each and set τ_7 to 4.82 (such that $e^{-\tau_7}$ increased by 4x; see equation 3) and repeated the simulations presented in Figure 6a. We found that performance increased somewhat for longer duration patterns, showing the greatest gain for pattern duration 200 ms, at which the completion fraction increased by 60% from the value achieved using the base parameters. However, this change to the network parameters reduced performance for shorter duration patterns, particularly evidenced by a 522% increase in the false positive fraction for patterns of 10 ms. Collectively, this demonstrates that network parameters can be fine tuned to improve performance *if* information is known about the temporal characteristics of the input stimuli (Figure 6b).

We next examined how well the algorithm dealt with patterns of different size by performing a series of simulations that used patterns involving progressively more $E1$ neurons. We found that network performance wasn't dramatically affected by pattern size, remaining relatively high (completion fraction at or above 0.92) for patterns ranging in size from covering 5 $E1$ neurons to half of the $E1$ population (50 neurons) (Figure 7).

C. Effect of Noise

To study how changes to signal and noise levels would influence learning and recall, we performed a series of simulations in which the network was trained on incomplete patterns, using signal fractions ranging from 1.0 (a complete pattern) to 0.1 (only 10% of a stimulus would be shown over the course of a particular pattern presentation). Surprisingly, the network exhibited very robust performance even when trained on degraded stimuli. Even with a signal fraction as

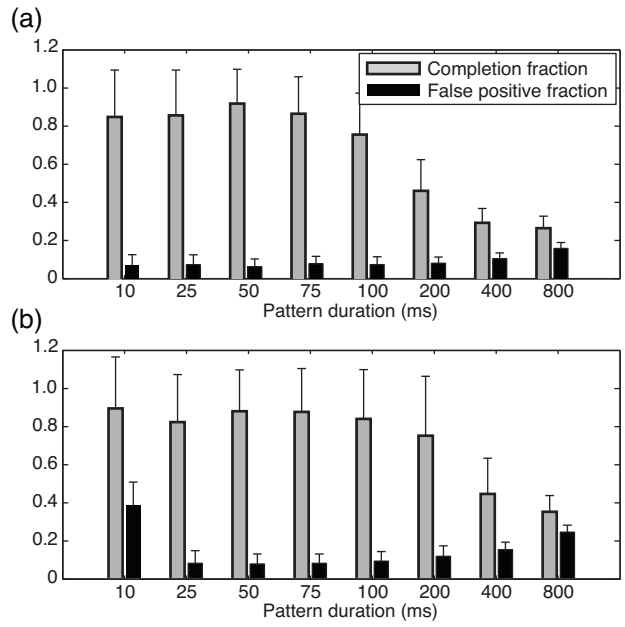

Fig. 6. a) The effect of changing pattern duration on pattern completion in the network. b) Changing pattern duration in a network employing wider temporal learning windows.

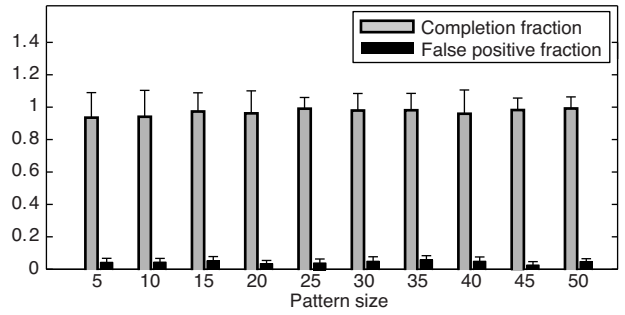

Fig. 7. The network was trained and tested using a pattern with spatial size ranging from 5 to 50 $E1$ neurons.

low as 0.6, the completion fraction was 0.89. This demonstrates that the network can piece together complete patterns even if never shown an entire pattern during any particular stimulus presentation (Figure 8a).

An additional set of simulations was performed to directly test the influence of noise on network performance. Noise levels were progressively increased from base levels across a series of simulations, resulting in a steady decrease in the network's ability to learn and complete patterns. Specifically, the completion fraction decreased until noise was 3 times base levels. For higher levels of noise, the random firings due to noise began to trigger $E2$ neurons more frequently, which produced more frequent recall of some form of stored pattern, although often not the correct one. Thus, as noise levels increased above 3 times base levels, the completion fraction increased but so did the false positive fraction, the latter reaching 0.42 for noise 4 times base levels. (Figure 8b).

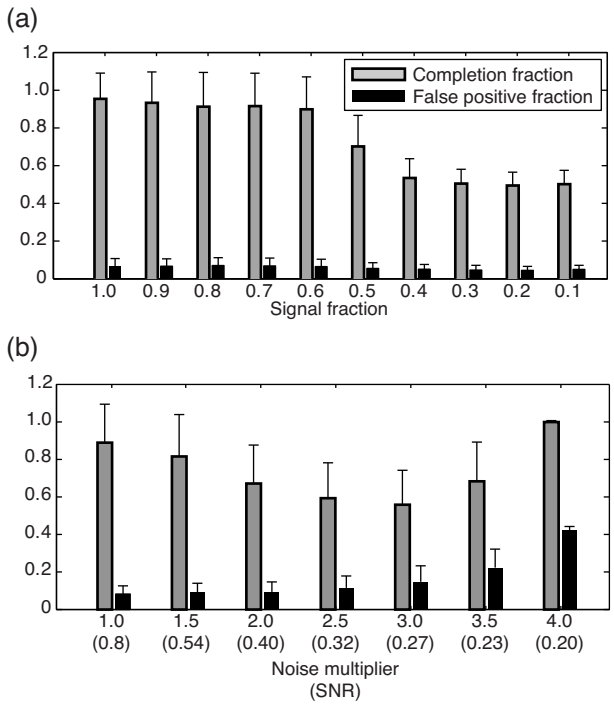

Fig. 8. a) The fraction of the stimulus shown during each pattern presentation in the training phase was decreased across a series of simulations. In each case, pattern completion was tested by presentation of stimuli using the default pattern fraction of 0.5. b) Noise was progressively increased across a series of simulations. The amount noise was multiplied from the base level is indicated on the x-axis along with the resulting SNR.

D. Sensitivity to changes in feedforward synaptic weight

As presented in Section II, the connections from $E1$ to $E2$ undergo synaptic weight normalization that enforces a constant total synaptic weight in the receptive field of each $E2$ neuron. That is, for all $j \in E2$, $\sum_{k \in E1} w_{kj}(t)$ is constant. We were interested in understanding how network behavior changes for different values of this total weight. We expected that decreasing the total $E2$ receptive field weight would result in $E2$ neurons firing less often in response to the presence of a pattern, but also to reduce their chance of responding (incorrectly) to noise. We expected that increasing this total weight would have the opposite effect. To test this, we performed a series of simulations in which we varied the total receptive field weight of each $E2$ neuron from the base value of 4. We indeed found that decreasing the total weight lowered both the completion fraction and the false positive fraction, while increasing this number had the opposite effect. Thus, the total $E2$ neuron receptive field weight can be tuned to achieve a particular balance between successful detections and false positives as dictated by the value and cost of each of these (Figure 9).

IV. Conclusions

We have described a two-layer network and an algorithm that learns repeating, sparse, partially and randomly presented patterns. We demonstrate through simulations that the layer $E2$ neuron receptive and projective fields, respectively

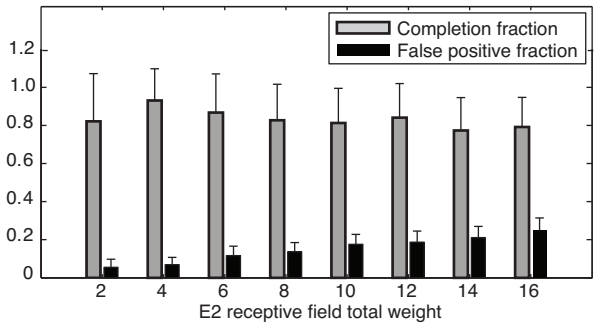

Fig. 9. The total receptive field weight of each $E2$ neuron was varied across a series of simulations.

corresponding to the feedforward and feedback projections in the network, converge to provide pattern specific assemblies. This allows the network to detect and recall patterns that can vary widely in their spatial and temporal extent, even in the presence of high levels of noise.

To the best of our knowledge, the only other published body of work that has broached the issue as posed can be found in [18], [19], though the approach there is entirely different – in both implementation and philosophy – to that presented here. In [18], [19], the patterns are learned by forming persistent assemblies – once ignited, they remain active over relatively long periods (several seconds) – a concept which many believe to be implicated in *working memory*. Our system takes a complementary approach by forming assemblies that are transient, that is, they are extinguished as soon as a successful retrieval has been carried out and once the pattern is deactivated. Such transient assemblies allow for an efficient way to perform rapid successive activation and retrieval of memories. A system with a dedicated circuit for pattern completion like the one proposed here has the ability to pinpoint events that require attention and subsequent analysis.

One advantage offered by our architecture is that it is readily amenable to incorporation into a modular framework, with each module having the basic two-layer design studied here. We plan to build on our construction by stacking the modules in a hierarchical fashion, borrowing the idea from the hierarchical organization exhibited by biological perceptual systems [20], with each level in the hierarchy representing features at varying degrees of abstraction. In a related vein, we can also add more top level layers, say, for example, $E2a$, $E2b$, and so on, to the basic two-layer design, with each layer responding, in parallel, to different features in the same input stream. This can be achieved by using different receptive field profiles for each layer [21]. Alternatively, we could have multiple bottom level layers, say, for example, $E1a$, $E1b$, and so on, with distinct input streams all feeding into one or more top level layers. Further, we plan to investigate ways in which the system can consolidate previously-learned patterns into complex composites by taking various permutations and combinations of these alternatives.

While our system shows remarkable performance (Section III), we would like to formally investigate its capacity, that is, the number of memories that can be reliably stored and retrieved. We would also like to know how capacity scales with network size. On a related note, working with fully connected networks tends to become prohibitively difficult as one scales up the network size. The computational requirements of the network presented are largely determined by its synapse count, which grows at a rate slightly greater than the product of the number of neurons in $E1$ and the number of neurons in $E2$. We are currently investigating means to expand the network size by limiting the (size of the) neurons' receptive and projective fields. This is in fact nature's preferred method of connectivity at larger scales. Preliminary results suggest that in order to compensate for this reduction, it is useful to allow for k-winner-take-all and structural plasticity, that is, a mechanism that allows a given neuron to re-distribute its weights along either its projective or receptive fields [22], [23].

A final point worth considering is the biological plausibility of our spiking microcircuit. The mammalian visual system is hierarchically organized, with incoming sensory information passing through a number of processing stages before culminating in object recognition [24]. Electrophysiological and behavioral experiments suggest that information can travel through this hierarchy quite rapidly, reaching upper levels in 100-150 ms [25], [26], with category discrimination specific activity appearing after 150 ms [27]. This rapid propagation leaves little time for processing at each step, allowing for perhaps 1-2 spikes to be passed between layers within a processing stage before information is relayed forward [28]. The circuit presented here is by no means purported to capture the complexity and richness exhibited by the cerebral cortex, but it does contain several key features present in cortical networks, namely spiking neurons, spike-time driven learning rules and recurrent connections between neurons. As shown above, our system performs input pattern identification and subsequent pattern recall very rapidly, requiring only a single spike per neuron for these tasks following an unsupervised training session. If we envision our circuit serving as an intermediate processing stage of a larger system, then we can interpret our results as a demonstration that a functionally significant amount of processing can occur under time constraints similar to those suggested by neurobiological experiments.

REFERENCES

[1] K. Steinbuch, *Automat und Mensch*. Springer: Berlin, Heidelberg, New York, 1963.
[2] D. Willshaw, O. Buneman, and H. Longuet-Higgins, "Non-holographic associative memory," *Nature*, vol. 222, pp. 960–962, (1969).
[3] G. Palm, *Biological Cybernetics*, vol. 36, pp. 19–31, 1980.
[4] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci USA*, vol. 79, p. 2554, 1982.
[5] S. Grossberg, *Studies of Mind and Brain*. Reidel, Boston, 1982.
[6] D. Rumelhart and D. Zipser, "Feature discovery by competitive learning," *Cognitive Science*, vol. 9, pp. 75–112, 1985.
[7] A. Lansner and O. Ekeberg, "Reliability and speed of recall in an associative network," *IEEE*, vol. 7, p. 490, 1985.

[8] T. Kohonen, *Content-Addressable Memories*. Berlin, Heidelberg, New York: Springer, 1987.
[9] R. Scalettar and A. Zee, "Emergence of grandmother memory in feed forward networks: Learning with noise and forgetfulness," in *Readings from Cognitive Science*, D. Waltz and J. A. Feldman, Eds., 1988, p. 309.
[10] E. B. Baum, J. Moody, and F. Wilczek, "Internal representations for associative memory," *Biological Cybernetics*, vol. 59, pp. 217–228, (1988).
[11] G. Chechik, "Spike-timing-dependent plasticity and relevant mutual information maximization," *Neural Computation*, vol. 15, pp. 1481–1510, (2003).
[12] D. Hebb., *The Organization of Behavior: A Neuropsychological Theory*. Wiley, New York, 1949.
[13] W. James, *The Principles of Psychology*. Cambridge, MA.: Harvard University Press, 1890.
[14] C. V. der Malsburg, "Self-organization of orientation sensitive cells in the striate cortex," *Kybernetik*, vol. 14, pp. 85–100, (1973).
[15] S. Grossberg, "Adaptive pattern classification and universal recoding: I. parallel development and coding of neural feature detectors," *Biological Cybernetics*, vol. 23, pp. 121–134, (1976).
[16] G. Carpenter and S. Grossberg, "The art of adaptive pattern recognition by a self-organizing neural network," *Cognitive Science*, pp. 77–88, (1988).
[17] R. Coultrip, R. Granger, and G. Lynch, "A cortical model of winner-take-all competition via lateral inhibition," *Neural Networks*, vol. 5, pp. 47–54, (1992).
[18] D. J. Amit and G. Mongillo, "Spike-driven synaptic dynamics generating working memory states," *Neural Computation*, vol. 15, pp. 565–596, (2003).
[19] G. Mongillo, E. Curti, S. Romani, and D. J. Amit, "Learning in realistic networks of spiking neurons and spike-driven plastic synapses," *European Journal of Neuroscience*, vol. 21, pp. 3143–3160, (2005).
[20] D. J. Felleman and D. C. V. Essen, "Distributed hierarchical processing in the primate cerebral cortex," *Cerebral Cortex*, vol. 1, pp. 1–47, (1991).
[21] J. J. Koenderink and A. J. van Doorn, "Receptive field families," *Biological Cybernetics*, vol. 63, pp. 291–297, (1990).
[22] M. P. Meyer and S. J. Smith, "Evidence from in vivo imaging that synaptogenesis guides the growth and branching of axonal arbors by two distinct mechanisms," *Journal of Neuroscience*, vol. 26, pp. 3604–3614, (2006).
[23] S. R. Williams, C. Wozny, and S. J. Mitchell, "The back and forth of dendritic plasticity," *Neuron*, vol. 56, pp. 947–953, (2007).
[24] E. T. Rolls, "Neural organization of higher visual functions," *Currrent Opinion in Neurobiology*, vol. 1, pp. 274–278, (1991).
[25] H. Liu, Y. Agam, J. R. Madsen, and G. Kreiman, "Timing, timing, timing: fast decoding of object information from intracranial field potentials in human visual cortex." *Neuron*, vol. 62, pp. 281–290, (2009).
[26] B. J. Richmond, R. H. Wurtz, and T. Sato, "Visual responses of inferior temporal neurons in awake rhesus monkey," *Journal of Neurophysiology*, vol. 50, pp. 1415–1432, (1983).
[27] R. VanRullen and S. J. Thorpe, "The time course of visual processing: from early perception to decision-making," *Journal of Cognitive Neuroscience*, vol. 13, pp. 454–461, (2001).
[28] R. Guyonneau, R. VanRullen, and S. J. Thorpe, "Temporal codes and sparse representations: a key to understanding rapid processing in the visual system," *Journal of Physiology – Paris*, vol. 98, pp. 487–497, (2004).

V. Acknowledgements

The research reported in this paper was sponsored by Defense Advanced Research Projects Agency, Defense Sciences Office (DSO), Program: Systems of Neuromorphic Adaptive Plastic Scalable Electronics (SyNAPSE), Issued by DARPA/CMO under Contract No. HR0011-09-C-0002. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressly or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

Approved for public release, distribution unlimited.