2019 2019 [ICONS ’19]International Conference on Neuromorphic SystemsJuly 23–25, 2019Knoxville, TN, USA International Conference on Neuromorphic Systems (ICONS ’19), July 23–25, 2019, Knoxville, TN, USA 15.00 10.1145/3354265.3354276 978-1-4503-7680-8/19/07

Towards Self-Organizing Neuromorphic Processors: Unsupervised Dictionary Learning via a Spiking Locally Competitive Algorithm

Yijing Watkins[1], Austin Thresher[2], Peter F. Schultz[2], Andreas Wild[3], Andrew Sornborger[1] and Garrett T. Kenyon[1,2]
Los Alamos National Laboratory[1]
New Mexico Consortium[2]
Intel Labs[3]

***Abstract*—A new class of neuromorphic processors promises to provide fast and power-efficient execution of spiking neural networks with on-chip synaptic plasticity. This efficiency derives in part from the fine-grained parallelism as well as event-driven communication mediated by spatially and temporally sparse spike messages. Another source of efficiency arises from the close spatial proximity between synapses and the sites where their weights are applied and updated. This proximity of compute and memory elements drastically reduces expensive data movements but imposes the constraint that only local operations can be efficiently performed, similar to constraints present in biological neural circuits. Efficient weight update operations should therefore only depend on information available locally at each synapse as non-local operations that involve copying, taking a transpose, or normalizing an entire weight matrix are not efficiently supported by present neuromorphic architectures. Moreover, spikes are typically non-negative events, which imposes additional constraints on how local weight update operations can be performed. The Locally Competitive Algorithm (LCA) is a dynamical sparse solver that uses only local computations between non-spiking leaky integrator neurons, allowing for massively parallel implementations on compatible neuromorphic architectures such as Intel's Loihi research chip. It has been previously demonstrated that non-spiking LCA can be used to learn dictionaries of convolutional kernels in an unsupervised manner from raw, unlabeled input, although only by employing non-local computation and signed non-spiking outputs. Here, we show how unsupervised dictionary learning with spiking LCA (S-LCA) can be implemented using only local computation and unsigned spike events, providing a promising strategy for constructing self-organizing neuromorphic chips.**

***Keywords*-Sparse Coding; Unsupervised Dictionary Learning; Spiking Locally Competitive Algorithm; Neuromorphic Processor.**

I. Introduction

Spiking neural networks (SNNs) are computational models that mimic biological neural networks. Compared with artificial neural networks (ANN), SNNs incorporate integrate-and-fire dynamics that increase both algorithmic and computational complexity. The justification for such increased complexity is two-fold: First, by using dedicated, potentially analog, circuit elements to instantiate individual neurons and by exploiting the low-bandwidth event-based communication enabled by SNNs, such networks can be implemented in extremely low-power neuromorphic hardware [3], enabling real-time remote applications that depend on scavenged power sources such as solar recharge. Second, there is evidence that biological neural circuits utilize spike timing to transmit information more rapidly and to dynamically bind distributed features via synchronous oscillations [11] [15] [14] [6]. The potential for mimicking the dynamics of biological neural networks in fast, low-power neuromorphic processors has motivated several efforts to develop such devices [4] [5] [9] [2] [13].

To fully exploit the potential of neuromorphic hardware, it is likely that such devices must be able to learn from their environment in a manner similar to biological neural systems. In particular, these devices must be able to learn in an unsupervised manner how to infer representations that support subsequent processing tasks. The Locally Competitive Algorithm (LCA) describes a dynamical neural network that uses only local synaptic interactions between non-spiking leaky-integrator neurons to infer sparse representations of input stimuli [10]. Unsupervised dictionary learning using convolutional LCA [12] has been used to infer sparse representations that support a number of signal processing tasks [17][16][8][7]. However, as previously implemented, unsupervised learning with non-spiking LCA utilizes non-local computations, specifically transpose and normalization operations performed globally on the entire weight matrix, and further requires signed outputs in order to represent the sparse reconstruction error. Thus, it remains unclear how unsupervised dictionary learning can be accomplished using only local operations and unsigned spiking output.

In this work, we show that unsupervised dictionary learning can be accomplished using a modified Spiking Locally Competitive Algorithm (S-LCA) that employs only local computations and uses only unsigned spiking output from all neurons. Thus our results provide a proof-of-concept for how neuromorphic processors can be configured so as to self-organize in response to natural environmental stimuli without sacrificing efficiency.

II. Methods

A. Unsupervised Dictionary Learning with a Non-Spiking LCA

Given an overcomplete basis, non-spiking LCA [10] can be used to find a minimal set of active neurons that represent the input to some degree of fidelity. Each neuron can be thought of as a generator that adds its associated feature vector to the reconstructed input with an amplitude equal to its activation. For any particular input, the optimal sparse representation is given by a vector of neural activations that minimizes the following cost function:

$$E = \frac{1}{2}||\mathbf{I} - \{\mathbf{\Phi} * \mathbf{a}\}||_2^2 + \frac{1}{2}\lambda^2||\mathbf{a}||_0 \qquad (1)$$

where $\mathbf{I}$ is the input vector and $\mathbf{\Phi}$ is a dictionary of feature kernels that are convolved with the activation coefficients $\mathbf{a}$. The L_0 norm $||\mathbf{a}||_0$ simply counts the number of non-zero activation coefficients. The factor λ acts as a trade-off parameter; larger λ values encourage greater sparsity (fewer non-zero coefficients) at the cost of greater reconstruction error.

LCA finds a local minimum of the cost function defined in Eq. (1) by introducing the dynamical variables (membrane potentials) $\mathbf{u}$ such that the output $\mathbf{a}$ of each neuron is given by a hard-threshold transfer function, with threshold λ, of the membrane potential: $\mathbf{a} = T_\lambda(\mathbf{u}) = H(\mathbf{u} - \lambda)\mathbf{u}$, where H is the Heaviside function[10].

The cost function defined in equation (1) is then minimized by taking the gradient of the cost function with respect to $\mathbf{a}$ and solving the resulting set of coupled differential equations for the membrane potentials $\mathbf{u}$:

$$\dot{\mathbf{u}} \propto -\frac{\partial E}{\partial \mathbf{a}} = -\mathbf{u} + \mathbf{\Phi}^T\{\mathbf{I} - \mathbf{\Phi}T_\lambda(\mathbf{u})\} + T_\lambda(\mathbf{u}). \qquad (2)$$

An update rule for feature kernels can be obtained by taking the gradient of the cost function with respect to $\mathbf{\Phi}$:

$$\Delta\mathbf{\Phi} \propto -\frac{\partial E}{\partial \mathbf{\Phi}} = \mathbf{a} \otimes \{I - \mathbf{\Phi}\mathbf{a}\} = \mathbf{a} \otimes \mathbf{R} \qquad (3)$$

where we introduced an intermediate residual layer $\mathbf{R}$ corresponding to the sparse reconstruction error.

For non-spiking LCA, online unsupervised dictionary learning is achieved via a two step process: First, a sparse representation for a given input is obtained by integrating Eq. (2), after which Eq. (3) is evaluated to slightly reduce the reconstruction error given the sparse representation of the current input.

As illustrated in Figure 1, the weight update (3) resembles a local Hebbian learning rule for $\mathbf{\Phi}$ with pre- and post-synaptic activities $\mathbf{a}$ and $\mathbf{R}$ respectively. However, the computation of $\mathbf{\Phi}^T$ renders the overall dictionary learning process a non-local operation.

We have previously shown that our implementation of non-spiking LCA can be used to learn a convolutional dictionary in an unsupervised, self-organizing manner that factors a complex, high-dimensional natural image into an overcomplete set of basis vectors that capture the high-dimensional correlations in the data [12]. In the next section, we show how this implementation can be adapted to an S-LCA model that uses only local computations and unsigned spiking output.

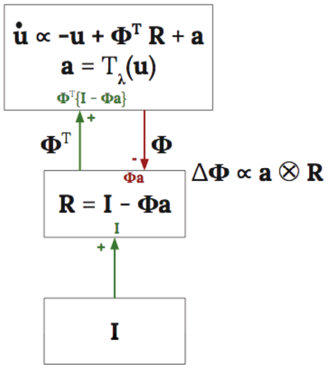

$\dot{\mathbf{u}} \propto -\mathbf{u} + \mathbf{\Phi}^T \mathbf{R} + \mathbf{a}$
$\mathbf{a} = T_\lambda(\mathbf{u})$
$\mathbf{\Phi}^T\{\mathbf{I} - \mathbf{\Phi}\mathbf{a}\}$
+
$\mathbf{\Phi}^T$
$\mathbf{\Phi}$
-
$\mathbf{\Phi}\mathbf{a}$
$\Delta\mathbf{\Phi} \propto \mathbf{a} \otimes \mathbf{R}$
$\mathbf{R} = \mathbf{I} - \mathbf{\Phi}\mathbf{a}$
$\mathbf{I}$
+
$\mathbf{I}$

Figure 1: A non-spiking LCA model that supports unsupervised dictionary learning via a residual or sparse reconstruction error layer.

B. Neuromorphic Constraints and Solutions

Recasting LCA in terms of spiking neurons producing unsigned spike events and using only local computations in a manner that still supports unsupervised dictionary learning introduces a number of challenges when mapping this algorithm to neuromorphic architectures. First, the non-spiking LCA model illustrated in Figure 1 relies on a non-local transpose operation of $\mathbf{\Phi}$ which is only inefficiently or not at all supported by present neuromorphic architectures. Second, to support local Hebbian learning, the reconstruction error represented by the residual layer in Figure 1 must be signed, to encode both positive and negative errors. However, most current neuromorphic architectures support only unsigned spike events. Third, with an L_0 sparsity penalty, there exists degenerate solutions in which the weights may be arbitrarily small while the activation coefficients become correspondingly large. Conversely, with an L_1 sparsity penalty, there exist degenerate solutions in which the weights can be arbitrarily large and activation coefficients correspondingly small. The non-spiking LCA model illustrated in Figure 1 avoids such degenerate solutions by normalizing the feature kernels after each weight update, but this again represents a non-local computation that cannot be efficiently implemented on neuromorphic architectures. Finally, neuromorphic architectures typically employ low precision representations of numerical quantities whereas the algorithm illustrated in Figure 1 assumes conventional floating point values are available.

Challenge A: Computations must be local, which prevents employing a transpose operation.
Proposed Solution: Replace $\mathbf{\Phi}^T$ with a separate connection $\mathbf{\Psi}$ that obeys its own local Hebbian learning rule. Since both the $\mathbf{\Phi}$ and $\mathbf{\Psi}$ connections link the same set of pre- and post-synaptic layers but with pre and post swapped, a local Hebbian learning rule that depends on pre × post results in the same change for both $\Delta\mathbf{\Phi}$ and $\Delta\mathbf{\Psi}$. We initialize $\mathbf{\Phi}$ and $\mathbf{\Psi}$ to be transposes of each other but beyond that no further formal synchronization is allowed or needed.

Challenge B: Both positive and negative reconstruction errors must be represented by unsigned spike events.
Proposed Solution: Modulate firing relative to a baseline rate for representing positive and negative values. Establish a baseline firing rate for the residual error layer, so that signed quantities can be encoded relative to the baseline rate. Firing rates below the baseline firing rate correspond to negative error whereas firing rates above the baseline firing rate correspond to positive error. This baseline firing rate can be maintained via local homeostatic regulation.

Challenge C: Weight normalization during unsupervised learning is non-local.
Proposed Solution: The dynamics of spiking neurons naturally imposes bounds on neuronal firing rates. Firing rates are bounded from below because there cannot be a fractional number of spikes, and are bounded from above because neurons have a finite refractory period. Weights naturally remain bounded because the activity is bounded and thus no normalization is needed.

Challenge D: Weight values have limited precision.
Proposed Solution: Sparse coding in general is very robust to noise due to the underlying attractor dynamics. Even with low precision representations of dynamical values, the system should still settle close to a local minimum of cost function Eq. 1.

C. S-LCA Unsupervised Dictionary Learning

Our S-LCA model for unsupervised dictionary learning is shown in Figure 2, where the superscripts indicate the different layers (e.g. $\mathbf{a^I}$ denotes the input spikes). We replace the non-spiking leaky-integrator model of Eq. 2 with a leaky integrate-and-fire (LIF) model consisting of a membrane potential, $\mathbf{u}$, and a binary spiking output, $\mathbf{a}$:

$$\dot{\mathbf{u}} \propto -\mathbf{u} + \mathbf{u}_{input} \tag{4}$$

$$\mathbf{a} = \begin{cases} 0 & \mathbf{u} < \lambda \\ 1; \mathbf{u} \rightarrow 0 & \mathbf{u} \geq \lambda \end{cases} \tag{5}$$

where λ again plays the role of a threshold that controls the level of sparsity and $\mathbf{u}_{input}$ is the sum of the input received from connected neurons. When $\mathbf{u}$ crosses the threshold λ a spike is generated and u is reset to zero.

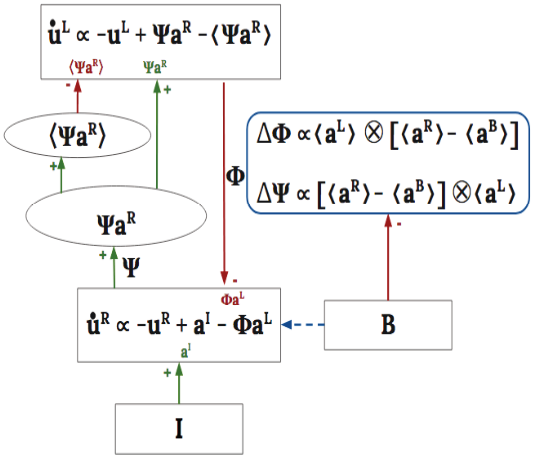

$\dot{\mathbf{u}}^{\mathbf{L}} \propto -\mathbf{u}^{\mathbf{L}} + \mathbf{\Psi a^R} - \langle\mathbf{\Psi a^R}\rangle$
$\langle\mathbf{\Psi a^R}\rangle$
$\mathbf{\Psi a^R}$
-
+
$\langle\mathbf{\Psi a^R}\rangle$
+
$\mathbf{\Psi a^R}$
+
$\mathbf{\Psi}$
$\mathbf{\Phi}$
$\Delta\mathbf{\Phi} \propto \langle\mathbf{a^L}\rangle \otimes [\langle\mathbf{a^R}\rangle - \langle\mathbf{a^B}\rangle]$
$\Delta\mathbf{\Psi} \propto [\langle\mathbf{a^R}\rangle - \langle\mathbf{a^B}\rangle] \otimes \langle\mathbf{a^L}\rangle$
-
-
$\mathbf{\Phi a^L}$
$\dot{\mathbf{u}}^{\mathbf{R}} \propto -\mathbf{u}^{\mathbf{R}} + \mathbf{a^I} - \mathbf{\Phi a^L}$
$\mathbf{a^I}$
B
+
I

Figure 2: S-LCA with unsupervised dictionary learning.

As with non-spiking LCA, the residual layer $\mathbf{R}$ in S-LCA is driven by the difference between the input and the reconstructed input generated by the LCA layer, which for S-LCA is given by $\mathbf{a^I} - \mathbf{\Phi a^L}$. In addition, S-LCA includes a homeostatic mechanism to set the baseline firing rate of the residual layer to a target value $\langle\mathbf{a^B}\rangle$. Values of $\langle\mathbf{a^R}\rangle$ above and below the target baseline firing rate encode positive and negative errors, respectively. Eq. 4 for the residual layer $\mathbf{R}$ then becomes

$$\dot{\mathbf{u}}^{\mathbf{R}} \propto -\mathbf{u}^{\mathbf{R}} + \mathbf{a^I} - \mathbf{\Phi a^L}. \tag{6}$$

We augment Eq. 6 with a firing condition analogous to Eq. 5 with $\lambda \rightarrow \lambda^{\mathbf{R}}$.

The input to the sparse coding layer $\mathbf{L}$ in Figure 2 is denoted by $\mathbf{u}^{\mathbf{L}}_{input}$, given by,

$$\mathbf{u}^{\mathbf{L}}_{input} = \mathbf{\Psi a^R} - \langle\mathbf{\Psi a^R}\rangle. \tag{7}$$

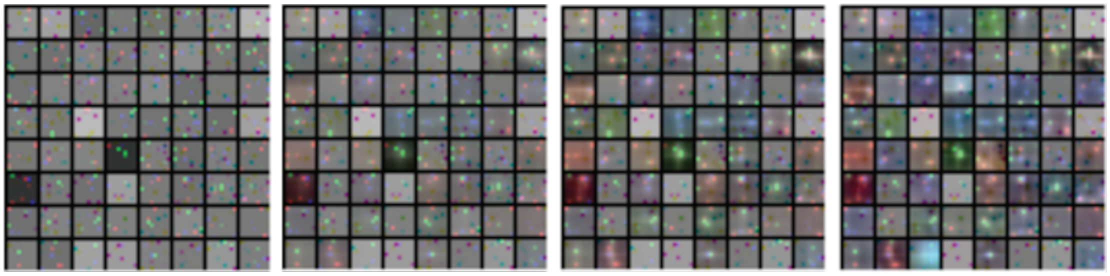
In Figure 2, $\langle\mathbf{\Psi a^R}\rangle$ is an exponentially moving temporal average (i.e. low-pass filtered output) of $\Psi\mathbf{a^R}$, so that the average input to the LCA layer $\mathbf{L}$ is zero.

Likewise, in Figure 2, the LCA layer $\mathbf{L}$ is again an LIF layer whose equation of motion is given by:

$$\dot{\mathbf{u}}^{\mathbf{L}} \propto -\mathbf{u}^{\mathbf{L}} + \mathbf{\Psi a^R} - \langle\mathbf{\Psi a^R}\rangle, \tag{8}$$

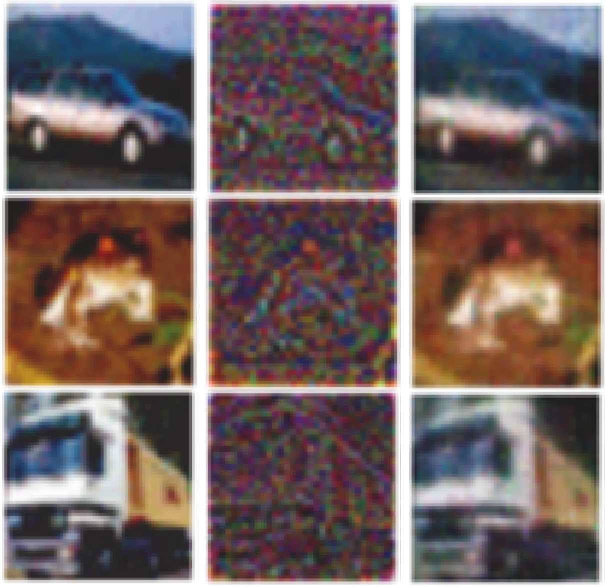
where we again augment Eq. 8 with a firing condition analogous to Eq. 5 with $\lambda \rightarrow \lambda^{\mathbf{L}}$. Recent neuromorphic architectures such as the Intel Loihi research chip [4] support the computations in Eq. 7, 8 if the sparse code layer is implemented in terms of multi-compartment neurons, allowing the instantaneous and low-pass filtered inputs from the residual layer to be combined.

Unsupervised dictionary learning can be used to update the weight matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$ efficiently on-chip given only



(a) (b) (c) (d)

Figure 3: (a) Initial stage dictionary at one epoch of training. (b-c) Middle stages dictionary during one epoch of training.(d) Final stage dictionary within one epoch of training.



(a) Original Images (b) Residuals (c) Reconstructions

Figure 4: Image reconstruction examples based on the learned dictionary.

information locally available at each synapse. To compute the weight updates, we introduce the low-pass filtered spike trains, or instantaneous firing rates, of the LIF neurons in the residual layer $\overline{\langle \mathbf{a^R} \rangle} = \langle \mathbf{a^R} \rangle - \langle \mathbf{a^B} \rangle$ computed relative to the target baseline firing rate of the residual layer. The firing rates of the LIF neurons in the sparse coding layer $\langle \mathbf{a^L} \rangle$ are likewise represented as low-pass filtered versions of the corresponding spike trains. In terms of these local firing rates, the update of $\mathbf{\Phi}$ and $\mathbf{\Psi}$ is given by a local Hebbian learning rule:

$$\Delta\Phi \propto \langle \mathbf{a^L} \rangle \otimes \overline{\langle \mathbf{a^R} \rangle},\ \Delta\Psi \propto \overline{\langle \mathbf{a^R} \rangle} \otimes \langle \mathbf{a^L} \rangle. \tag{9}$$

III. Results

We implemented the S-LCA unsupervised dictionary learning model in PetaVision[1], an open source neural simulation toolbox that enables multi-node, multi-core and GPU accelerated high-performance implementations of both non-spiking LCA and S-LCA.

We simulated an S-LCA model consisting of a single convolutional layer containing 64 features with a patch size of 9×9 pixels and a stride of 1 ($\simeq 10.7$ times overcomplete). In our implementation, we chose a convolutional model to greatly decrease the time required to train a dictionary, as a convolutional model allowed us to average weight changes over multiple patches and to take advantage of GPU acceleration. Although a convolutional network architecture requires non-local operations, our results and conclusions apply to both convolutional and non-convolutional models in general. In addition, recent attempts have been made to support connection-sharing in neuromorphic architectures [4].

We initialized the dictionary by assigning 10% of the synaptic weights a nonzero random value, and then rescaled each dictionary element to have zero mean and unit standard deviation. We used CIFAR-10 images as input, where the input spike firing rate for each pixel is proportional to the pixel value. The evolution of the dictionary learning over one epoch is illustrated in Figure 3. Figure 4 shows examples of original, residual error and reconstruction images based on the trained unsupervised dictionary from our S-LCA model.

IV. Conclusion

An algorithm must observe locality of computation in order to map successfully to recent neuromorphic hardware architectures [4] [5] [9]. Furthermore, to enable communication-efficient implementation, algorithms must exploit the low-bandwidth spike-based communication that such architectures provide. Utilizing these features promises vastly lower energy consumption and/or faster execution time on neuromorphic architectures compared to traditional CPU/GPU architectures. In this work, we demonstrated for the first time, how a spiking locally competitive algorithm (S-LCA) can be implemented using only unsigned spike events and local computations. S-LCA allows for both unsupervised dictionary learning and inference to be performed in a manner compatible with the constraints of recent architectures, such as the Intel Loihi research chip. As unsupervised dictionary learning via sparse coding accounts for many aspects of cortical development, our results thus suggest a viable strategy through which neuromorphic processors can self-organize efficiently in response to unlabeled natural environmental stimuli.

ACKNOWLEDGMENT

This work was funded by NSF #1734980, the NNSA ASC Beyond Moore's Law program, and the DOE LDRD program at LANL.

REFERENCES

[1] PetaVision. Software available from petavision.github.io.

[2] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, G.-J. Nam, et al. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(10):1537–1557, 2015.

[3] K. Boahen. A neuromorph's prospectus. *Computing in Science Engineering*, 19(2):14–28, Mar. 2017.

[4] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018.

[5] S. Friedmann, J. Schemmel, A. Grübl, A. Hartel, M. Hock, and K. Meier. Demonstrating hybrid learning in a flexible neuromorphic hardware system. *IEEE transactions on biomedical circuits and systems*, 11(1):128–142, 2017.

[6] G. T. Kenyon. Extreme synergy: Spatiotemporal correlations enable rapid image reconstruction from computer-generated spike trains. *Journal of vision*, 10(3):21–21, 2010.

[7] S. Y. Lundquist, M. Mitchell, and G. T. Kenyon. Sparse coding on stereo video for object detection. In *workshop on Learning with Limited Labeled Data: Weak Supervision and Beyond, NIPS 2017*. NIPS, 2017.

[8] S. Y. Lundquist, D. M. Paiton, P. F. Schultz, and G. T. Kenyon. Sparse encoding of binocular images for depth inference. In *Image Analysis and Interpretation (SSIAI), 2016 IEEE Southwest Symposium on*, pages 121–124. IEEE, 2016.

[9] E. Painkras, L. A. Plana, J. Garside, S. Temple, S. Davidson, J. Pepper, D. Clark, C. Patterson, and S. Furber. Spinnaker: A multi-core system-on-chip for massively-parallel neural net simulation. In *Proceedings of the IEEE 2012 Custom Integrated Circuits Conference*, pages 1–4, Sept 2012.

[10] C. J. Rozell, D. H. Johnson, R. G. Baraniuk, and B. A. Olshausen. Sparse coding via thresholding and local competition in neural circuits. *Neural Comput.*, 20(10):2526–2563, Oct. 2008.

[11] R. V. Rullen and S. J. Thorpe. Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex. *Neural computation*, 13(6):1255–1283, 2001.

[12] P. Schultz, D. Paiton, W. Lu, and G. Kenyon. Replicating kernels with a short stride allows sparse reconstructions with fewer independent kernels, 2014.

[13] P. M. Sheridan, F. Cai, C. Du, W. Ma, Z. Zhang, and W. Lu. Sparse coding with memristor networks. *Nature Nanotechnology*, 12, 05 2017.

[14] G. J. Stephens, S. Neuenschwander, J. S. George, W. Singer, and G. T. Kenyon. See globally, spike locally: oscillations in a retinal model encode large visual features. *Biological cybernetics*, 95(4):327–348, 2006.

[15] S. Thorpe, A. Delorme, and R. Van Rullen. Spike-based strategies for rapid processing. *Neural networks*, 14(6-7):715–725, 2001.

[16] Y. Watkins, O. Iaroshenko, M. R. Sayeh, and G. T. Kenyon. Image compression: Sparse coding vs. bottleneck autoencoders. In *2018 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, SSIAI '18, 2018.

[17] Y. Watkins, A. Thresher, D. Mascarenas, and G. T. Kenyon. Sparse coding enables the reconstruction of high-fidelity images and video from retinal spike trains. In *Proceedings of the International Conference on Neuromorphic Systems*, ICONS '18, pages 8:1–8:5, New York, NY, USA, 2018. ACM.