Sparse approximation on a network of locally competitive integrate and fire neurons

Samuel Shapero, Daniel Brüderle, Paul Hasler, and Christopher Rozell

Sparse coding is a hypothesized coding strategy where a population of sensory neurons (e.g., V1) encode a stimulus using as few active neurons as possible. It is not currently clear what mechanisms a biophysically realistic neural population could use to solve the optimization problem proposed by this coding strategy. We present a network architecture of integrate and fire (I&F) neurons that calculate sparse approximations for natural images. This work incorporates spiking neural models into locally competitive algorithms (LCAs), which are analog dynamical systems that have been shown to calculate sparse approximations. In this work we show that I&F populations can approximate the necessary nonlinearities for the LCA, and we simulate in NEURON a network of 128 neurons that encode 8x8 pixel image patches to demonstrate that the network converges to nearly optimal encodings. We also show that when using larger (and more biophysically realistic) leakage parameters, the network dynamics optimize an objective function that encourages sparsity in the encoding. Finally, to demonstrate the network effectiveness in a physical implementation and gain computational speed, we construct a small network on a mixed-signal neuromorphic chip.

In sparse coding, a vector input $y \in \mathbb{R}^N$ is represented with an overcomplete dictionary $\Phi = [\phi_1, \dots, \phi_M]$ using neural activities $a \in \mathbb{R}^M$ by minimizing $\min_a \left(\frac{1}{2}\|y - \Phi a\|_2^2 + \lambda \Sigma_i C(a_i)\right)$, where $C(\cdot)$ is the sparsity-inducing cost function (e.g., the ℓ_1 norm). The LCA [3] is an analog dynamical system minimizing this objective via the dynamics

$$\tau \dot{u}(t) + u(t) = \Phi^t y - (\Phi^t \Phi - I)\ a(t),$$
$$a(t) = T(u(t)), \quad T^{-1}(a) = a + \frac{dC(a)}{da}.$$

For the common special case of $C(a) = |a|$, the necessary activation function is the soft-thresholder [3].

This structure lends itself well to an I&F network using a rate code to represent the inputs and outputs. We generate excitation $\Phi^t y$ and lateral inhibition $(\Phi^t \Phi - I)$ using variable weight conductive synapses. We tune the leakage currents of the I&F neurons to approximate the desired activation function $T(\cdot)$. When the reset voltage is close to the threshold voltage ($\beta = \frac{V_{TH} - V_{RESET}}{V_{TH} - V_{REST}} \ll 1$), $T(\cdot)$ approximates the soft-thresholder. Increasing ratio β to more biophysically realistic levels introduces an offset in the activation function. The resulting cost function is non-convex, but this objective encourages sparsity in the solution.

To test this I&F network, we simulated the encoding of 200 8x8 whitened image patches in a learned overcomplete dictionary [2]. The network was coded in PyNN, which allowed us to port our design to the NEURON simulator, and to the mixed-signal neuromorphic chip Spikey [1].

We compared the quality of the final sparse encoding to the results of solving the same objective using a specialized digital solver with thresholds $\lambda \in [0.01...0.1]$. We observed that the neural activity of the I&F network consistently converged to a solution that minimized the objective function to within 2% of the ideal optimal value, and to within 0.1% when $\lambda = 0.1$. When we increased the biophysical realism by decreasing the reset voltage, neural activities were less optimal in terms of minimizing the objective function (see fig. 1), but had fewer nonzero components. It is possible that the random fluctuations of the instantaneous spike rate actually aided the network in finding good local minima to the non-convex objective function.

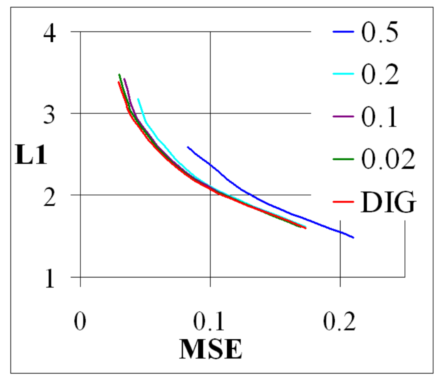

Fig. 1: Metrics of I&F network (sorted by $\beta = \frac{V_{TH} - V_{RESET}}{V_{TH} - V_{REST}}$) vs. digital solver

When we ported our design to the Spikey chip, the speed and intrinsic parallelism achievable with the physical I&F network allowed the network to converge in 70 µs. This fast implementation will be especially useful in future explorations of the role of sparse coding networks for biological systems.

[1] D. Brüderle, E. Müller, A. Davison, E. Muller, J. Schemmel, and K. Meier. Establishing a novel modeling tool: A Python-based interface for a neuromorphic hardware system. *Front. Neuroinform.*, 3(17), 2009.

[2] B. Olshausen and D. Field. Emergence of simple-cell receptive field properties by learning in a sparse code for natural images. *Nature*, 381:607–609, 1996.

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