Departmental Papers (BE)   Department of Bioengineering

May 2006

Dynamic Computation in a Recurrent Network of Heterogeneous Silicon Neurons

Paul Merolla
University of Pennsylvania

Kwabena Boahen
Stanford University

Recommended Citation

Dynamic Computation in a Recurrent Network of Heterogeneous Silicon Neurons

Abstract

We describe a neuromorphic chip with a two-layer excitatory-inhibitory recurrent network of that exhibits localized clusters of neural activity. Unlike other recurrent networks, the clusters in our network are pinned to certain locations due to transistor mismatch introduced in fabrication. As described in previous work, our pinned clusters respond selectively to oriented stimuli and the neurons' preferred orientations are distributed similar to the visual cortex. Here we show that orientation computation is rapid when activity alternates between layers (staccato-like), dislodging pinned clusters, which promotes fast cluster diffusion.

Dynamic computation in a recurrent network of heterogeneous silicon neurons

Paul A. Merolla and Kwabena Boahen

Dept. of Bioengineering, UPenn, {pmerolla,boahen}@seas.upenn.edu

Abstract—We describe a neuromorphic chip with a two-layer excitatory-inhibitory recurrent network of spiking neurons that exhibits localized clusters of neural activity. Unlike other recurrent networks, the clusters in our network are pinned to certain locations due to transistor mismatch introduced in fabrication. As described in previous work, our pinned clusters respond selectively to oriented stimuli and the neurons' preferred orientations are distributed similar to the visual cortex. Here we show that orientation computation is rapid when activity alternates between layers (staccato-like), dislodging pinned clusters, which promotes fast cluster diffusion.

I. Pattern-forming recurrent networks

A 2-D recurrent network of spiking neurons with Mexican hat connectivity (local excitation and distal inhibition) can exhibit clusters of activity when the feedback is sufficiently strong. These clusters are an emergent property of the network and are identified by contiguous regions of activity surrounded by dead zones (no activity). In a homogeneous network (i.e., neurons and their connections are all identical), the locations of the clusters are unconstrained and have an equal likelihood of existing at any position. Therefore, clusters in a homogeneous network move in a random walk, constrained only by their interactions with nearby clusters [1].

In contrast, networks with heterogeneous neurons tend to bias the locations where clusters reside. Clusters do not wander freely but are instead pinned to the locations that maximize their local recurrent feedback. One intriguing possibility is that the interactions between spatio-temporal input patterns (e.g., visual scenes) and these pinned clusters can process information. For example, it has been shown that oriented stimuli are able to shift the clusters away from their preferred locations to produce orientation selective responses whose distribution resembles cortical maps of preferred orientation (PO) [2], [3]. The seemingly complicated task of assigning similar POs to nearby neurons is cleverly achieved by simply building an imprecise, recurrent network.

Transforming fixed-pattern noise into a smoothly changing feature map is an impressive feat, but this transformation is poorly understood. In particular, it is not known how cluster dynamics, which can range from fluid (mobile clusters) to crystalline (immobile clusters), influence PO map creation. We address this issue by characterizing how clusters diffuse over a range of network states and by examining the speed at which orientation maps converge for two disparate diffusion rates.

Exploring a detailed, large-scale recurrent network over a wide range of network parameters is a computationally daunting task that is poorly suited for software modeling.
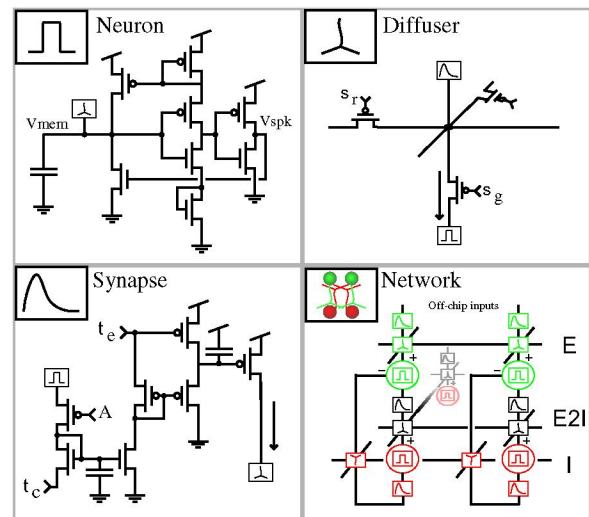

Fig. 1. Transistor implementations for a neuron, diffuser, and a synapse are shown together along with their input–output interactions; the network panel shows how these individual blocks are organized into neural circuits. We refer to circuit parameters as type_{par}, where type specifies the section of the neural circuit, and par specifies the circuit parameter (e.g., the excitatory cell synapse parameter A is referenced as E_A).

Previous attempts to model such networks in software have sacrificed low-level details that are known to affect cluster dynamics (e.g., they replace spiking with rate-based equations) [4], and have explored only a handful of network states [2], [1]. For these reasons, we have chosen to build our network in silicon using a neuromorphic implementation [5]. This approach has the advantage of: I operating in real-time, II retaining important low-level details [6] (i.e., electronic currents flowing through MOS transistors are analogous to ionic currents in a biological membrane), and III supporting large-scale networks ($\approx 10,000$ neurons).

Our two-layer excitatory–inhibitory recurrent network is built from canonical transistor circuits that capture the behavior of their biological counterparts (Fig. 1). These circuits, which are designed to operate in the subthreshold region, perform the following functions:

- A **neuron** integrates excitatory and inhibitory currents on its membrane capacitor and generates a spike (brief voltage pulse) after reaching threshold [7].
- A **diffuser**, which models axonal arbors, spreads synaptic current to local regions in space with exponential decay [8].

- A **synapse** converts spikes into postsynaptic currents that decay exponentially [3].

A computational block of our network, which is tiled across the chip, consists of an excitatory (EXC) and an inhibitory (INH) neuron reciprocally connected through synapse and diffuser circuits (Fig. 1 *network*). Note that feedback excitation (not shown) is implemented by routing EXC spikes transmitted off-chip back into neighboring synapse circuits (see Section II).

II. MODEL IMPLEMENTATION

We implemented our recurrent network in silicon using a 0.25μm 5-metal layer CMOS process. The fabricated chip consists of a 2-D core of 48×48 blocks, surrounded by asynchronous digital circuitry that transmits and receives spikes, encoded using the address-event representation (AER), in real-time [9]. Our custom chip has $\approx 460,000$ transistors packed in 10 mm^2 of silicon area for a total of 9,216 neurons (only half of which are used in this paper for the recurrent network).

To implement spatial interactions in our network, we use diffuser circuits to spread post-synaptic currents (described previously). However, we must use caution when diffusers implement connectivity between cells in the same layer. The caveat is that in addition to spreading current to neighboring nodes, they also implement an autapse (self-connection) that is weighted by the peak of the exponential. For this reason, we implement EXC to EXC connections through a FPGA–RAM interface, which generates multiple events targeted to neighboring neurons (whose addresses are stored in the RAM). Self-excitation is still present through neighboring diffuser nodes (see Fig. 1), but the strength of the autapse is significantly reduced.

A bi-directional AER–USB (universal serial bus) interface allows us to view, process, and save neuron activity from our chip in real-time while simultaneously sending events (i.e., afferent spikes) to drive the network. Utilizing the USB 2.0 high-speed standard (asynchronous mode), we achieved ≈ 7 million address-events/sec. With a custom GUI, we were able to interact with our network in real-time by: I changing the level of input activity to the chip with a keystroke, II viewing spike rasters to gauge the extent of network synchronization, III viewing network activity over space, and IV determining when neural activity becomes unstable by monitoring the most-excitable cells.

III. CLUSTER DYNAMICS

Our network exhibits spatio-temporal patterns in certain parameter regimes. In particular, when local feedback excitation (E_A) is scaled relative to distal inhibition (I_A), clusters of neural activity form across the network (Fig. 2). These clusters are dynamic: over time they can change shape, coalesce with neighbors, disappear, or diffuse. However clusters do not move far; this confinement is the result of transistor mismatch. Without this heterogeneity, clusters have no preferred locations.
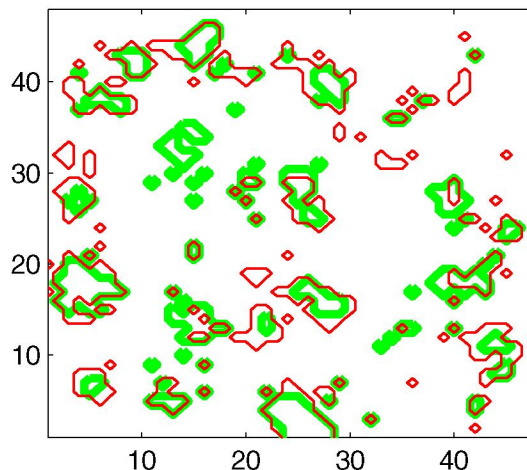

Fig. 2. Contours of EXC cell activity (clusters) are shown for two 10ms intervals 50ms apart (thin and thick lines). In this example, the network parameters (in volts) are: for INH, $I_{sr} = 1.844$, $I_{sg} = 1.897$, $I_{te} = 2.374$, $I_A = 1.45$, $I_{tc} = 0.061$; for EXC, $E_{sr} = 2.046$, $E_{sg} = 2.045$, $E_{te} = 2.4$, $E_A = 1.855$, $E_{tc} = 0.052$; for their interaction, $E2I_{sr} = 2.051$, $E2I_{sg} = 2.098$, $E2I_{te} = 2.387$, $E2I_A = 1.602$, $E2I_{tc} = 0.05$, and Vdd $= 2.5$. Each cell in the network is driven with a 75Hz poisson background input. These setting are used throughout the paper except where noted.

Our goal is to characterize the dynamics observed in our heterogeneous network over a range of parameters. To accomplish this, we first introduce metrics that quantify how clusters diffuse in time. We then compare cluster dynamics over a range of different network states — a daunting task considering the extent of the parameter space. We whittle the astronomic parameter space down to an interesting subspace by taking advantage of the real-time visualization offered in our neuromorphic system. Finally, we gauge the speed of computation by exploring how cluster diffusion relates to PO map convergence.

A. Confined cluster diffusion

We characterize our network dynamics by measuring the mean-squared displacement of clusters $\langle \Delta^2 r \rangle$ over time. The measurement is performed as follows: First, we identify the centers of clusters for two frames separated by time t_0.[1] Next, we calculate the pair-wise distances of all the centroids from one frame to the next (Fig. 3). The mean of this distribution, $\langle \Delta^2 r(t_0) \rangle$, is calculated using the region around the initial peak (Fig. 3, lighter region), which corresponds to local cluster diffusion (explained below). Repeating this measurement for different values of t, we obtain $\langle \Delta^2 r(t) \rangle$ (Fig. 3 *Inset*).

The multiple humps observed in the $|\Delta|r(t_0)$ distribution (Fig. 3) are characteristic of clusters diffusing in a confined area. The initial peak is the result of clusters moving a small distance (or not moving at all), whereas the nearby trough

[1]Clusters are identified by finding contiguous regions of activity within a frame (i.e., cells that have at least half of their neighbors active). Then, the center of mass of each cluster is computed, where the mass of each cell is determined by its firing-rate. We cap the max firing-rate to 500Hz (i.e., 5 spikes in a 10ms frame), which limits the contribution of over active cells.
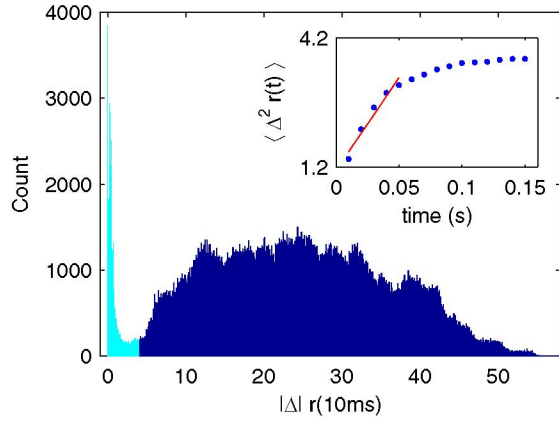
Fig. 3. The distances that clusters travel in 10ms, or $|\Delta|r(10\text{ms})$ (same network state as in Fig. 2). We show $|\Delta|r(10\text{ms})$ instead of $\Delta^2 r(10\text{ms})$ because the former distribution is easier to visualize. All computations however use $\Delta^2 r(t)$. *Inset* Mean-squared displacement, $\langle\Delta^2 r(t)\rangle$, only considering the region around the initial peak. The slope of the fit (D) is 43.02 pix^2/s.
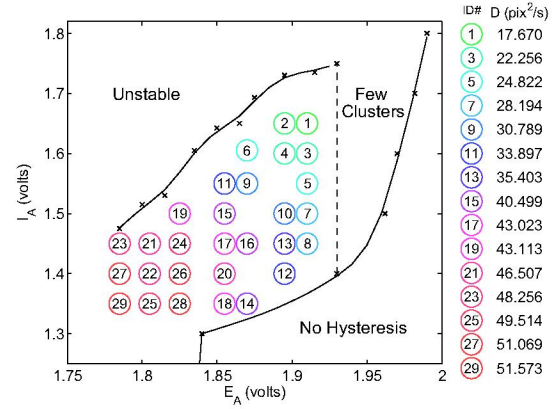
Fig. 4. Chip behaviors (numbered) are shown for different voltages I_A and E_A. Corresponding diffusion rates D are shown to the right. Note that lowering the voltage corresponds to increased synaptic strength (since A connects to a PMOS transistor).

represents distances where clusters are unlikely to travel. The n subsequent peaks correspond to the mean distance between n-th nearest neighbors; the peaks exist because, on average, many clusters are consistently the same distance from each other. In this paper, we only consider the distribution near the initial peak.

The shape of $\langle\Delta^2 r(t)\rangle$ (inset of Fig. 3) provides further evidence that, in this particular network state, clusters diffuse in a confined area. Indeed, over short time intervals (10 to 50ms), the mean-squared displacement varies linearly with time. In other words, the expected distance a cluster will travel is proportional to how long we wait (up until $\approx$ 50ms). We denote the slope of this linear region as D (traditionally called the diffusion constant). For longer time intervals ($>$ 90ms), the expected distance tends to saturate. This behavior is indicative of a confined cluster; it will never travel beyond a certain distance regardless of how long we wait. An analogy between cluster dynamics and particles vibrating in a crystalline lattice can be drawn — but we must be careful not to take the analogy literally. Unlike particles, clusters are not required to follow physical laws (e.g., the probability that a cluster must exist somewhere is not required to be 1).

The method that we have described allows us to obtain D without explicitly tracking clusters through time. By using the distance distribution directly, we avoid the task of characterizing non-diffusive cluster behavior. For instance, a cluster that vanishes (or merges with a neighbor) from one frame to the next does not affect the distance distribution near the initial peak; hence, non-diffusive behavior is automatically segregated from our analysis.[2]

B. Dependence on network parameters

Here, we explore how cluster dynamics, and in particular the diffusion constant D, depend on parameter choices. We

[2]This method was inspired in part by Tristan Ursell's web document *Diffusion of solid particles in viscous fluid*, and by [10].

accomplish this by first finding a suitable network operating point in the high-dimensional parameter space with interesting cluster dynamics; the chosen settings are indicated in Fig. 2. Our real-time GUI facilitated this search since we were able to monitor relevant network properties (e.g., network stability) while tuning the parameters.

Having found a suitable operating point, we now constrain our analysis to the post-synaptic current amplitudes (E_A and I_A), for the simple reason that these two parameters sample a wide-range of different dynamics (from inspection). Because two parameters still constitute a large space, we further reduce their range by cutting out regions where: I clusters do not self-sustain for more than 10 seconds in response to an impulse (no hysteresis); II clusters do not pack the entire network (few clusters); and III more than 28 neurons have an average firing-rate above 1,000Hz (unstable) (I and III are demarcated by x's in Fig. 4). It is important to note that these regions are empirically defined and do not imply underlying bifurcations.

Having carved out a parameter range for E_A and I_A, we compute D for points in that range (Fig. 4). For each point $\langle\Delta^2 r(t)\rangle$ has been measured over 10ms to 150ms in 10ms intervals. D is the slope of the linear fit from 10 to 50ms; the lowest R^2 (i.e., percentage of variance explained), considering the entire dataset, was 0.885.

C. Orientation map convergence

We track PO maps over time in two different network states to gauge how cluster diffusion affects computational speed. First, maps are computed by presenting four moving gratings with orientations ranging from 0 to 135 degrees for 1,000 seconds and then computing the vector sum — the resulting angle is a neuron's PO (see [3] for an example). Next, we compare short-term PO maps (considering part of the trial) with the final PO map (entire trial); comparisons are made using a map similarity index (SI) that computes the normalized angle difference between maps, (adopted from [11]). An SI of 1 signifies the maps have identical orientation preferences; 0.5
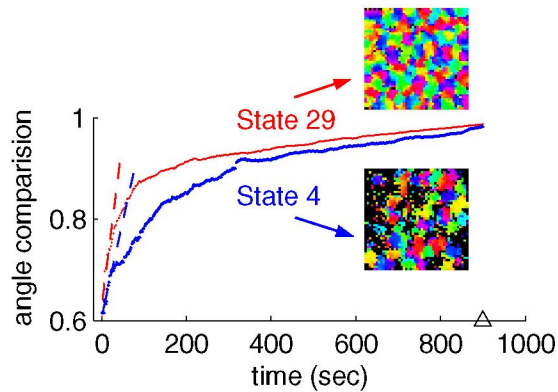
Fig. 5. Map convergence rates, measured as the SI between short-term PO maps and the final map, for two states. Fits for the first 10s of each state are shown, and extrapolated out to an SI of 0.9 (dashed lines). PO maps are also shown for the time indicated by the △; colors are determined by each cell's PO. Dark regions indicate cells that have not fired; note that more cells are active when cluster diffusion is high.
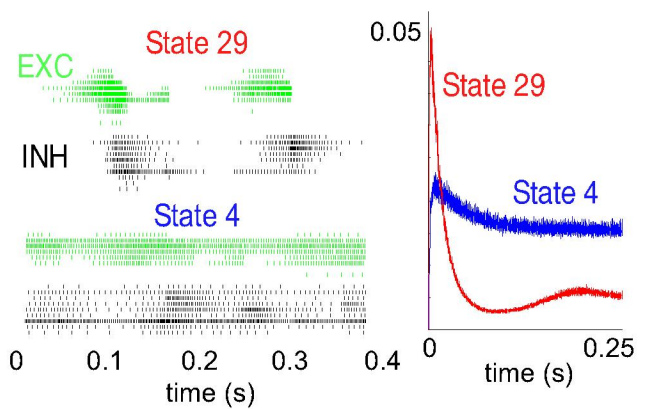

Fig. 6. Spike rasters (left) for 10 EXC (green) and INH (black) neurons for two network states and their average AC's (right). AC is calculated by considering EXC spikes over a 50s interval at a 0.1ms bin resolution; it is normalized for each neuron individually and averaged over 100 neurons.

signifies no correlation (i.e., the average PO difference is 45 degrees); and 0 signifies anti-correlation (i.e., the average PO difference is 90 degrees). The convergence rates for the two distinct network states 29 and 4 (red and blue traces in Fig. 5, respectively) show that greater cluster diffusion results in faster convergence. For example, the initial rate is 6.6×10^{-3} in SI/s (dashed red) when $D = 51$ pix^2/s (state 29), whereas the rate is 3.8×10^{-3} (dashed blue) when $D = 23$ pix^2/s (state 4).

We have shown that cluster diffusion (D) can influence map convergence, but D only quantifies how far clusters diffuse over time; the fundamental question of how clusters diffuse in the presence of heterogeneity is still open. Viewing spikes from EXC and INH neurons directly (Fig. 6) provides a clue. In the high diffusive state (state 29) activity alternates between the two layers — first EXC responses build for about 80ms (green rasters), which then recruit massive activity in INH cells (black rasters). These staccato-like dynamics can also be seen in the autocorrelation (AC) (Fig. 6 *right*, red trace); the AC is high for short times (activity is initially correlated with itself), forms a trough at around 80ms (activity is anti-correlated 80ms later), and rebounds near 200ms. In contrast, activity in the low-diffusive state (state 4) is more regular, and its AC decays monotonically as a power law with an exponent of -0.10 (Fig. 6 *right*, blue).

IV. CONCLUSION

We have presented a chip that implements a recurrent network of spiking neurons with Mexican hat connectivity. The network displays clusters of neural activity that are biased to certain locations as a result of network heterogeneity. Previous work has shown that these pinned clusters and their interactions with spatio-temporal patterns can perform a useful vision computation (orientation extraction). In this paper, we have elaborated on how the dynamics of clusters influence network computation; specifically, we show that diffusive clusters lead to faster-converging orientation maps.

The cluster dynamics that we observed in our network, quantified by the local diffusion constant D, ranged from 17 to 51 pix^2/s. To reach the high-diffusion regimes, the network operated with staccato-like interactions between EXC and INH neurons. We propose that this staccato activity state is one way to achieve cluster diffusion in the presence of heterogeneity. Furthermore, this staccato state could provide a way to balance computational speed and efficacy in a heterogeneous network.

REFERENCES

[1] M. A. Usher and M. Stemmler. Dynamic pattern formation leads to 1/f noise in neural populations. *Physical Review Letters*, vol. 74(2):326–329, January 1995.

[2] U. A. Ernst, K. R. Pawelzik, C. Sahar-Pikielny, and M. V. Tsodyks. Intracortical origin of visual maps. *Nat Neurosci*, vol. 4(4):431–6, 2001.

[3] P. A. Merolla and K. Boahen. A recurrent model of orientation maps with simple and complex cells. pages 995–1002. MIT Press, 2004.

[4] C. R. Laing and C. C. Chow. Stationary bumps in networks of spiking neurons. *Neural Comput*, vol. 13(2):1473–94, 2001.

[5] C. A. Mead. *Analog VLSI and Neural Systems*. Addison Wesley, Reading MA, 1989.

[6] J. V. Arthur and K. Boahen. Recurrently connected silicon neurons with active dendrites for one-shot learning. In *IJCNN'04 International joint conference on neural networks*, pages 1699–1704. IEEE Press, 2004.

[7] E. Culurciello, R. Etienne-Cummings, and K. A. Boahen. A biomorphic digital image sensor. *Ieee Journal of Solid-State Circuits*, 38(2):281–294, 2003.

[8] A. Andreou and K. Boahen. Translinear circuits in subthreshold mos. *Journal of analog integrated circuits ans signal processing*, vol. 9:141–166, 1996.

[9] K. A. Boahen. A burst-mode word-serial address-event channel. *Ieee Transactions on Circuits and Systems I-Regular Papers*, 51(7):1269–1300, July 2004.

[10] J. C. Crocker and D. G. Grier. Methods of digital video microscopy for colloidal studies. *J. Colloid Interface Sci*, 179:298–310, 1996.

[11] B. Chapman, M. P. Stryker, and T. Bonhoeffer. Development of orientation preference maps in ferret primary visual cortex. *J Neurosci*, vol. 16(20):6443–53, 1996.