Appendix: Emergence of Hierarchical Layers in a Single Sheet of Self-Organizing Spiking Neurons

Paul Bertens
Department of Brain and Cognitive Engineering
Korea University
Seoul, South Korea
paulbertens@korea.ac.kr

Seong-Whan Lee
Department of Artificial Intelligence
Korea University
Seoul, South Korea
sw.lee@korea.ac.kr

A Appendix

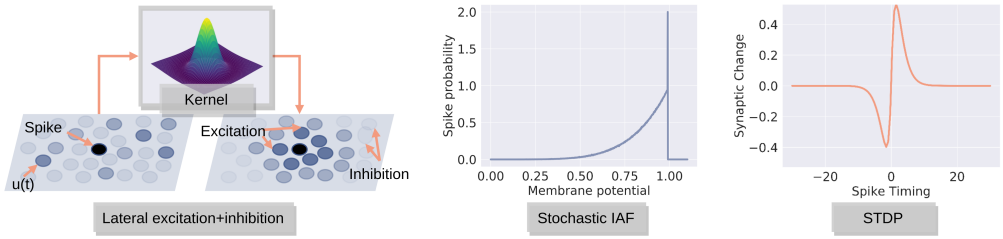


Figure 1: Detailed examples of the lateral excitation and inhibition kernel, stochastic Integrate and Fire (Stochastic IAF), and Spike-Timing-Dependent Plasticity (STDP). For lateral excitation and inhibition, the kernel ensures that neurons nearby spiked neurons increase in their membrane potential u(t). This increase then increases the probability of firing (through the Stochastic IAF model). It also causes inhibition on farther neurons, which reduces the probability of those neurons firing. This causes a type of local winner take all to take place within the kernel size, where only a single neuron fires. This kernel is the result of the difference of 2 Gaussians (DoG) with different sigma values.

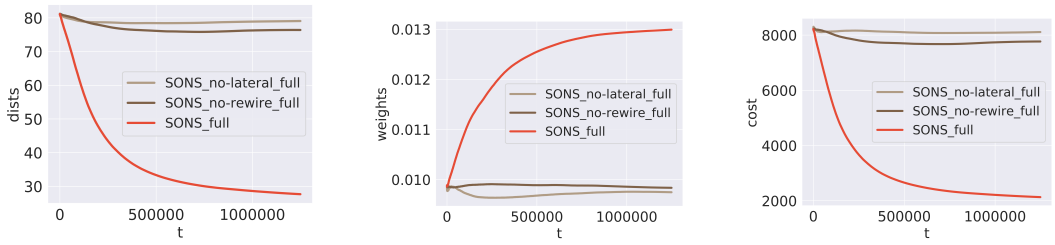
Lateral Excitation and Inhibition Equation of the underlying lateral excitation and inhibition kernel that we convolve over the neural sheet for each spiked neuron can be described as follows:

$$DoG_{2D}(x, y) = \frac{1}{2\pi\sigma_+^2} e^{-\frac{x^2+y^2}{2\sigma_+^2}} - \frac{1}{2\pi\sigma_-^2} e^{-\frac{x^2+y^2}{2\sigma_-^2}} \tag{1}$$

Where x and y are the offset coordinates relative to the spiked neuron, and σ_+ and σ_- the different sigmas for excitation and inhibition respectively. Please also refer to figure 1 for a more visual description.

Topographic map generation The topographic maps were generated by averaging the input patches for which each neuron responded to (i.e. spiked). The average was taken over 5000 patches. These topographic maps show classical pinwheel type patterns similar to the visual cortex in the brain, where edges of similar orientation are clustered closer together. For the topographic maps in the full image neural sheet experiment we directly visualize the synaptic strength of the connections that go to the input. Since in this case the input is no longer a patch but an entire image, and we are interested in visualizing what each neuron responds to locally (i.e. the localized feature kernels learned).

36th Conference on Neural Information Processing Systems (NeurIPS 2022).



(a) Mean wiring length (b) Mean wiring weights (c) Mean wiring cost

Figure 2: Ablation study results for Self-Organizing Neural Sheet (SONS) model. Shows SONS model without dynamic rewiring (i.e. no growth or pruning), and without lateral connectivity (i.e. no lateral excitation and inhibition). Training progress in terms of mean synaptic connection length ($d(p,q)$), mean synaptic strength ($w(p,q)$) and total resulting wiring cost ($\frac{d(p,q)}{w(p,q)}$).

Hyper-parameter selection Hyper-parameters were selected based on logical requirements. In the case of determining the sheet size a sufficiently shaped sheet was created that can contain the image size and have enough vertical space to let layers emerge (100x200). The learning rate of updating the synaptic weights were chosen such that synaptic updates were smooth over multiple iterations. For stochastic spiking in the IAF model, the noise distribution was chosen that provided sufficient minimal noise for synaptic growth to occur. No extensive hyper-parameter search was performed to obtain these parameter values, and initial choices were found to work well in practice (table 1).

	Patch	Full Image
Pre-σ^+	2	2
Pre-σ^-	3	3
n_spike_bins	8	8
n_sim_steps	20	60
n_total_steps	200k	3600k
in_constraint	64	64
out_constraint	1024	64
n_steps_prune	2000	6000
Lateral σ^-	32	3
Lateral σ^+	3	2
STDP: η	0.1	0.1
IAF: a	5	5
IAF: b	0.0001	0.0001

Table 1: Detailed experimental parameters. Pre-σ is the sigma used for the ON and OFF center surround cells in the preprocessing step. Spike bins indicate into how many temporal bins we obtain after converting the ON/OFF cell responses to spike patterns. Sim steps is how many steps we run the neural sheet for each stimuli image. Total steps is how many time steps we train in total. In and Out constraints indicate the number of allowed incoming and outgoing connections. Note how in the case of patches this constraint is much higher, as we have more space to generate much larger topographic maps per patch than per image. This also affects the lateral inhibition σ_-, which now has to be much larger to inhibit a larger area of neurons in the 32x32 grid, instead of just a localized small area as is done in the full image neural sheet setup. Steps until pruning indicates for how many steps a synapse cannot be pruned. This gives newly grown synapses sufficient time to undergo STDP and mature to overtake existing connections.

Block-Sparse matrix details An example of the custom block-sparse data structure used to efficiently prune and grow connections is given in figure 4. We can store both incoming and outgoing connection indices per neuron, and keep both matrices in sync when pruning or growing new connections. Two sub-matrices are required to efficiently pruning either incoming or outgoing connections.

This fixed block-structure allows for efficient memory usage, efficient fetching of incoming and outgoing connections per neuron, and an efficient forward pass (since we have far fewer synaptic weights). This makes such block-sparse matrices much more scalable, while growing/pruning connections maintains learning flexibility.

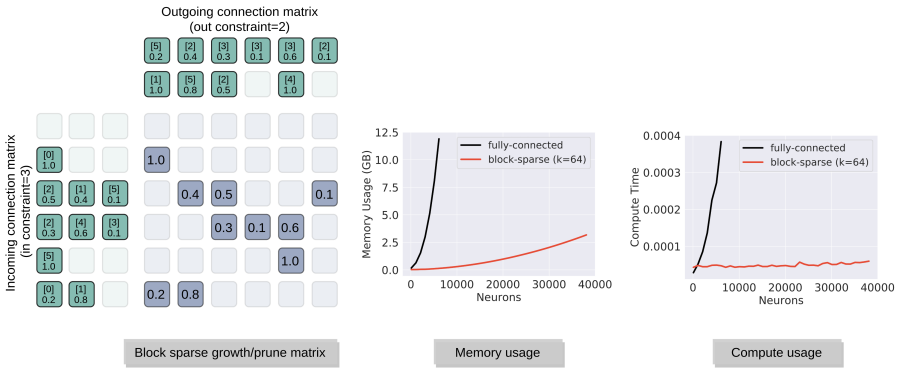


Figure 3: Block-sparse growth prune matrix example. Each entry includes the synaptic weight value and associated index in the matrix. Simulated memory and GPU usage is also given for different number of neurons (different neural sheet sizes) for both fully connected matrix and block-sparse matrix. Implementation was done in PyTorch running on a Titan V GPU and results were averaged over 10000 runs. Memory usage is the result of storing and caching weight values, normalized weights, connection distances, insert/growth time and intermediate copies. GPU compute time is the result of forward pass of spiked neurons. Memory and compute time can be seen to exponentially increase with fully connected weight matrix as expected, but stays approximately linear with a block-sparse implementation as we constrain the number of allowed connections (k=64).

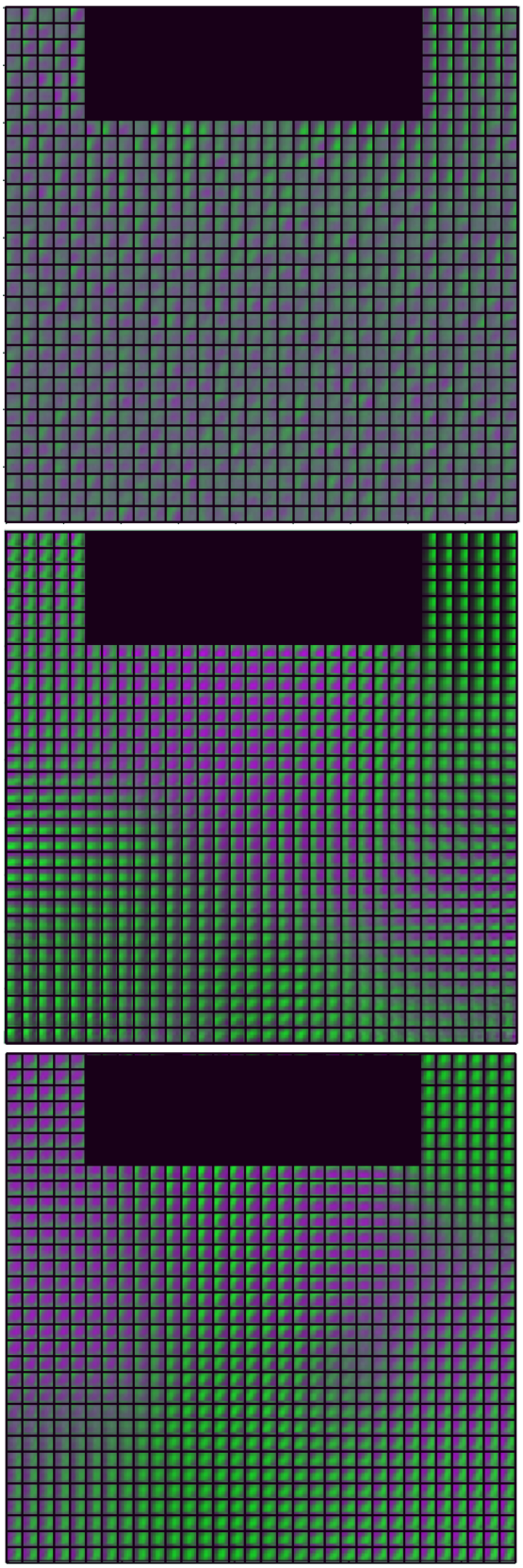


Figure 4: Zoomed in version of obtained topographic maps. SNN (top), SOM (middle), SONS (bottom). The SOM and SONS models perform similarly in this setting, but the SONS is able to scale to much larger sheet sizes.