Project Proposal - Andrew Landau (working with Eve Marder and Mark Goldman)

Background

What mechanisms do neurons use when they learn how to compute useful transformations of their inputs? Most studies of neural plasticity have focused on synaptic plasticity mechanisms that modulate the strength of synaptic connections with other neurons in various ways. However, neurons are characterized by many biophysical parameters that can potentially be utilized to learn transformation functions.

Neurons in the stomatogastric ganglion are exquisitely parameterized to generate the pyloric rhythm while maintaining robustness to a range of intrinsic and synaptic parameterizations. However, work from Fournier & Marder (2024) indicates that the circuit is more sensitive to perturbation of intrinsic properties than perturbation to synaptic connections. This argues that intrinsic properties may be a critical node for the computational properties of neurons.

Biophysical properties of neurons are known to be dynamically modulated by the state of the animal. For example, Williams & Fletcher (2019) showed that acetylcholine can increase the excitability of the apical dendritic compartment of L5 pyramidal neurons by increasing the single channel conductance of R-type calcium channels, which has large effects on the input-output curve of these cells. In addition to transient modulation, it has also been observed that dendritic excitability can exhibit long-term modulation through plasticity mechanisms. Losonczy & Magee (2008) showed that individual dendritic compartments can down-regulate A-type potassium currents over long-timescales through activity-dependent mechanisms, which improves the transmission of synaptic input to somatic output.

Proposal

I propose to study how intrinsic properties are used and/or learned for simple computational tasks in spiking recurrent neural networks built with integrate and fire neurons. I will use this as a model system for my question because it provides a sufficiently rich capacity for subthreshold dynamics that the intrinsic conductances can have a large impact on the circuit while being simple enough that the model can be trained and analyzed in a straightforward way (in comparison to a Hodgkin-Huxley formulation, for example). My project will address two interrelated questions:

1) How does the performance of spiking RNNs depend on intrinsic properties in comparison to synaptic properties?
2) Are intrinsic properties sufficient for learning simple computational tasks in spiking RNNs?

Methods

To specify and train spiking RNNs, I will use the publicly available python package snnTorch. The package contains models of integrate and fire neurons that I will extend to contain voltage-dependent and calcium dependent channels which will modulate subthreshold dynamics. I will specify the additional channels with standard alpha/beta rate parameters, and the maximum conductance of each channel will be a learnable parameter using standard auto differentiation.

Question 1: Dependence on Intrinsic Properties

To analyze how spiking RNNs depend on intrinsic properties, I will train spiking RNN models on several simple tasks in which they learn both through input weights, recurrent weights, readout weights, and intrinsic parameters. For my first pass, I will not enforce Dale's law and will let neurons have weights with an arbitrary sign at every synapse. The tasks I will use are the following: 1) Go/No-go, 2) Context-dependent Go/No-go, and 3) Delayed Match-to-sample. In general, the input to the network will be an N-dimensional set of currents with learned weights for each neuron. The output for each task will be a readout accumulator that performs perfect integration of spikes (scaled by their readout weights), followed by a softmax, since each task can be modeled as a classification problem.

To analyze the dependence on intrinsic properties, I will use two methods. First, I will simply directly measure the gradient using a backward pass in each task. To get a more complete picture, I will do this across a large batch of input and across an ensemble of identical networks that were all initialized randomly. I will interpret this result as an (poor) estimate of the hessian matrix of all the task parameters with respect to each task. Then, I will simply compare the amplitude of the gradient for intrinsic parameters with synaptic parameters. Secondly, I will perform perturbation analyses in which the learnable parameters of the network will be directly perturbed, then the performance on each task will be measured.

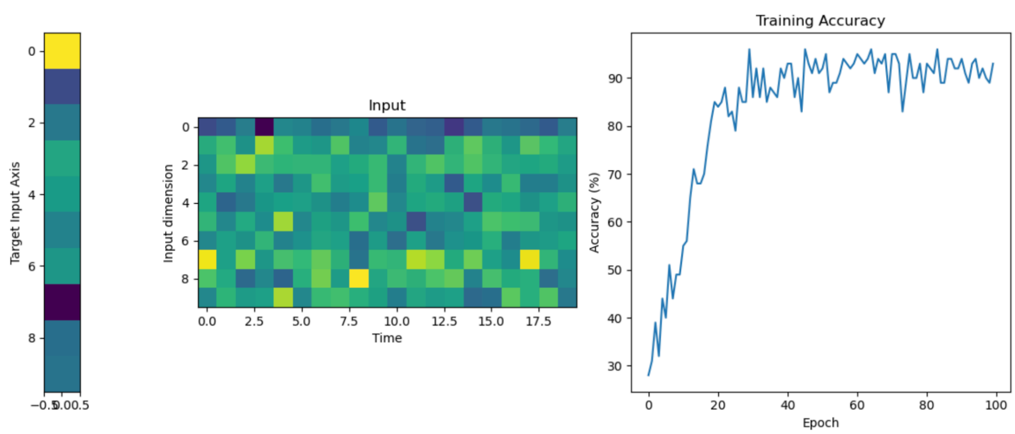


Figure 1: Example of a noisy detection Go/No-Go task.

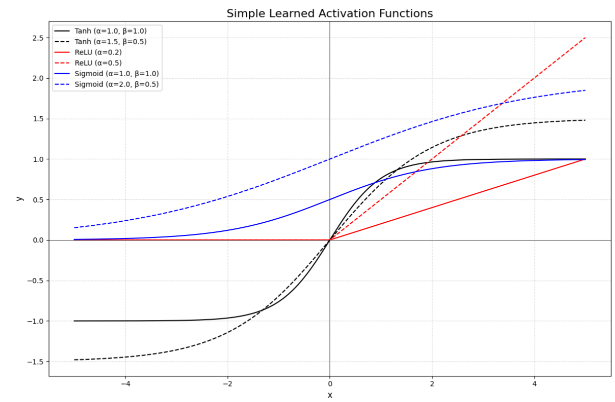
Left: target input axis - if the input has a positive projection onto this axis, the network should "GO", and if it has a negative input then the network should "NOGO".

Middle: example data with 20 timesteps for a NOGO trial.

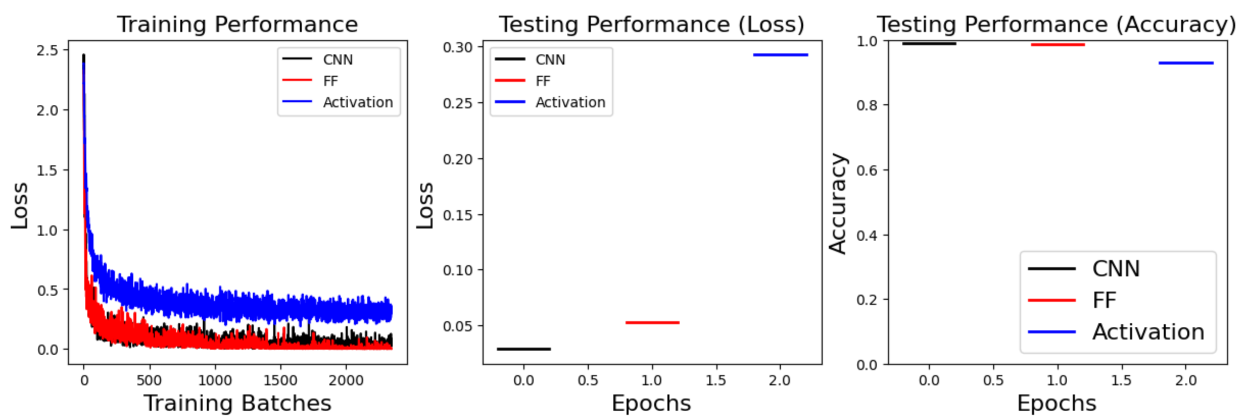
Right: accuracy across training for signal amplitude=1.0 and noise amplitude=3.0

Question 2: Learning with Intrinsic Properties

Another relevant question is if intrinsic properties are sufficient to learn tasks, independent of whether they are important for a task that also learns through synaptic connections. To study this problem, I will repeat the above experiments from the first question, but use fixed synaptic weights and learn exclusively through intrinsic parameters. For this question, I will likely need to use a network with many neurons such that the existing weights contain enough information about the input for the network to discover a latent solution.



As a proof of principle, I demonstrated this phenomenon using a multilayer perceptron trained to classify MNIST. I created a wide, 3 layer fully connected network where the weights were fixed from initialization. Each neuron was equipped with 5 parameters that specified learnable activation functions. The activation function was a linear combination of a ReLU, a sigmoid, and a hyperbolic tangent, where each had a gain parameter and the sigmoidal components had a stretching parameter. The “Activation” network doesn’t perform as well as a convolutional network or a standard feedforward network, but it achieves greater than 90% performance on the task for testing data. If I have time, I may try to extend this branch of the project for a secondary perspective on the problem, but it will not be my main focus.



Fournier, Alonso, Marder (2024), Her Powerpoint for MCN

Williams, Stephen R., and Lee N. Fletcher. "A dendritic substrate for the cholinergic control of neocortical output neurons." Neuron 101.3 (2019): 486-499.

Losonczy, Attila, Judit K. Makara, and Jeffrey C. Magee. "Compartmentalized dendritic plasticity and input feature storage in neurons." Nature 452.7186 (2008): 436-441.

Jason K. Eshraghian, Max Ward, Emre Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D. Lu “Training Spiking Neural Networks Using Lessons From Deep Learning”. Proceedings of the IEEE, 111(9) September 2023.