

Class & home assignment: modeling single neuron as deep networks

Code for this assignment:

https://github.com/SelfishGene/tutorial_NEURON_pytorch

Data for this assignment (optional):

<https://www.kaggle.com/datasets/selfishgene/ball-and-stick-neuron-data>

Reference papers:

<https://www.sciencedirect.com/science/article/pii/S0896627321005018>

<https://pmc.ncbi.nlm.nih.gov/articles/PMC11702691/>

<https://arxiv.org/abs/2309.15090>

Overview:

In this assignment we will learn how to use the NEURON simulation environment to simulate detailed biophysical multi compartment neuron models, and how to train deep artificial neural networks using pytorch.

We will be simulating and fitting deep networks to a Ball and Stick neuron model.

i.e. a soma and a single dendritic cable attached to it.

Specifically, we will be trying to understand the differences between the functional input-output complexities of two variants of these models:

A simple model: a short and wide dendrite (128um length, 4um diameter) with AMPA synapses.

This is essentially a point neuron model as the cable is very short and wide and all locations on the neurons have nearly identical voltages.

A more complex model: a long and narrow dendrite (2048um length, 1um diameter) with human NMDA synapses.

This is no longer a point neuron as the cable is long and narrow enough that if something happens at a specific location on the dendrite, the voltage propagates only to its local environment. The NMDA synapses also contain voltage dependent behavior of the input. In particular, the same input will result in vastly different amounts of current injected into the cell if the voltage at that particular location is near rest vs. if the voltage is sufficiently depolarized.

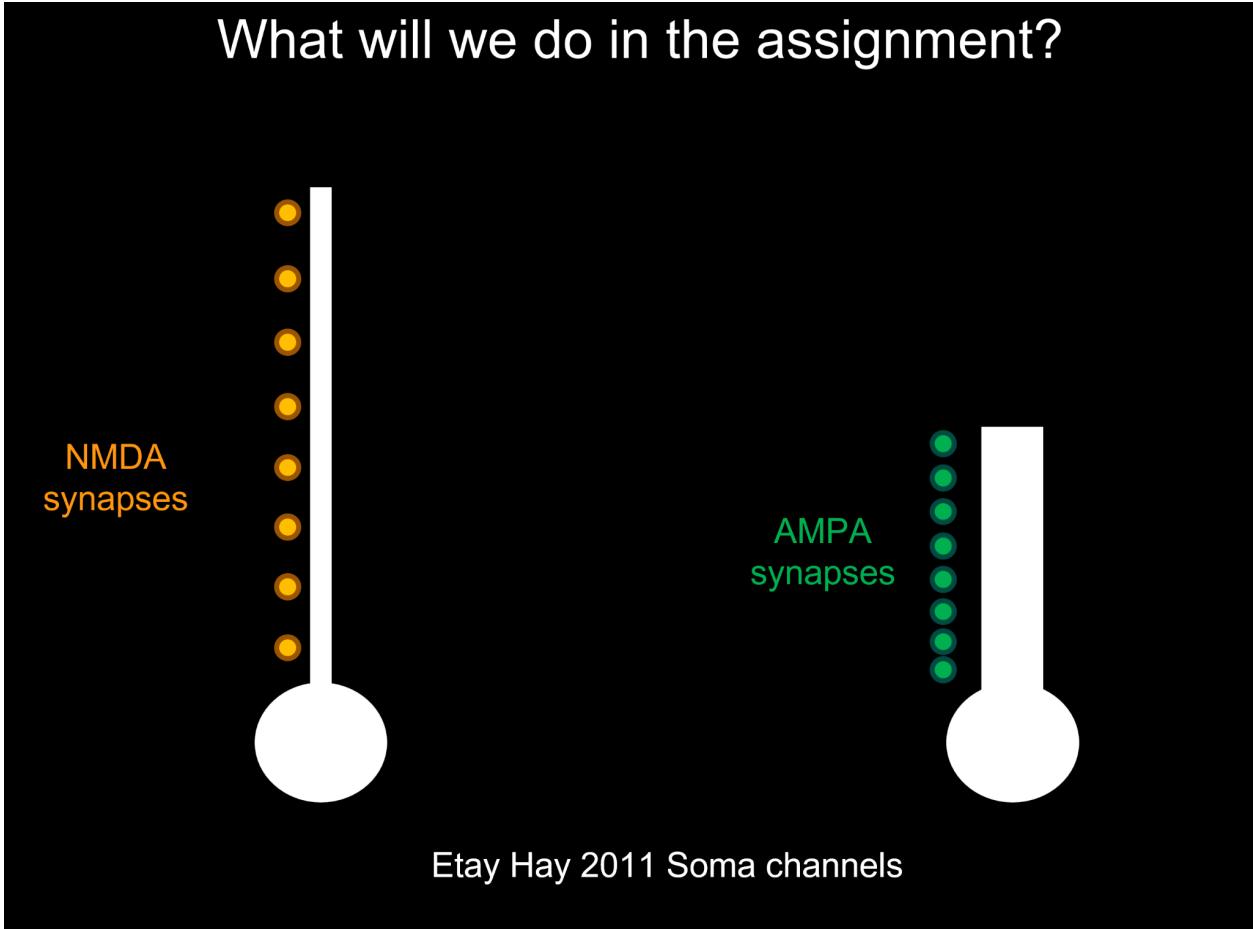
For each of these cells, we will first generate a large dataset of random inputs sent to the neuron model, and record their output.

We will then fit deep networks of various sizes to each of these datasets and record their accuracies.

Finally, we hope to reach some interesting conclusions.

Schematic illustration of the two cells below:

What will we do in the assignment?



Class Assignment:

Follow the notebook at:

https://github.com/SelfishGene/tutorial_NEURON_pytorch/blob/main/notebooks/simulate_ball_and_stick_neuron.ipynb

Run it cell by cell and try to understand what is going on.

If you wish, you can use the following starter colab network and work in the cloud instead of your own computers:

<https://colab.research.google.com/drive/1wgmJzoowipmzl5aUEvWtPohrGMSUJ93k?usp=sharing>

Home Assignment:

Create the datasets

Follow the notebook of:

https://github.com/SelfishGene/tutorial_NEURON_pytorch/blob/main/notebooks/create_dataset_BS_neuron_AMPA_point.ipynb

Alter the “num repeats” variable and generate a large dataset of simulations for the **simple** model of the two. (at least ~3000 training simulations is required)

Follow the notebook of:

https://github.com/SelfishGene/tutorial_NEURON_pytorch/blob/main/notebooks/create_dataset_BS_neuron_NMDA_cable.ipynb

Alter the “num repeats” variable and generate a large dataset of simulations for the **complex** model of the two. (at least ~5000 training simulations is required)

Optional: it should take around 1-4 hours to create the datasets on a laptop (depending on the laptop). If you are impatient, it's possible to download the two datasets from:

<https://www.kaggle.com/datasets/selfishgene/ball-and-stick-neuron-data>

Train deep networks of various sizes on the datasets

Follow the notebooks of:

https://github.com/SelfishGene/tutorial_NEURON_pytorch/blob/main/notebooks/train_DN_N_twin_of_BS_neuron_AMPA_point.ipynb

https://github.com/SelfishGene/tutorial_NEURON_pytorch/blob/main/notebooks/train_DN_N_twin_of_BS_neuron_NMDA_cable.ipynb

These two notebooks are identical, but were run on the two different datasets and one can get a feel of what one should expect to get using them.

For each dataset, we will train 5 different hyperparam configurations and record the results.

In code cell #8 it's possible to change the TCN depth width and temporal kernel sizes

Lets mark with **D**, **W**, **T** the various hyperparams:

```
first_layer_temporal_kernel_size = T  
num_layers_per_block_list = [D]  
num_features_per_block_list = [W]  
temporal_kernel_size_per_block_list = [T]
```

For each dataset, the 5 model hyperparam configurations we will train are:

Config A: D=1, W= 8, T=91

Config B: D=2, W=16, T=61

Config C: D=4, W=32, T=41

Config D: D=7, W=64, T=31

Config E: D=7, W=64, T=11

each training run should take between 5-30 minutes on a laptop (depending on the GPU and network size)

NOTE: when we use D, the network that will be trained will actually be of depth D+1. This is OK for our purposes here.

Analyze the results

Plot a side by side bar plot with the AUC value of for each of the 5 hyperparam config (on the x axis), for each of the neuron models (use different colored bars for each neuron).

Inspect the figure. Can you explain what we see? What does this imply?

Can you explain what happens in E compared to D?

Optional: If you are having technical difficulties (or just lazy), it's possible to download the 10 pretrained models and collect their accuracies from:

<https://www.kaggle.com/datasets/selfishgene/ball-and-stick-neuron-data>

Bonus Assignment:

1) Try to plot the weights of the first layer of one of the trained TCNs of configs A or B for both the simple and complex neuron models. This will require some basic understanding of pytorch and the problem setup (or it's possible to use chatgpt/claudie/cursor to help with this).

2) Play around with ELM, ResNet, Transformer hyperparams, can you surpass the accuracy of the biggest TCN we trained on the complex neuron model?

What is the smallest possible network to beat the TCN? (there 3 possible ways to refer to "small" here: A) in terms of number of parameters the deep network has, B) in terms of the speed to run inference on a single simulation file, C) in terms of total GPU memory consumption, what is the answer for each of these? Are the answers the same?)

3) Think and suggest how can we, using the code repo, design an experiment to understand
A) the importance of SKE2 ion channels at the soma?

B) the importance of Human NMDA synapses vs. rodent NMDA synapses?