18-847 Lab Assignment 2

Due 2019-02-14 11:59pm PST
Due 2019-02-15 2:59am EST
100 Points

This lab assignment will require you to think a bit about computational primitives and help you get started on building the core functionality of your temporal neural network simulator.

You may work in teams of up to 3 for this assignment and may submit as a group for this assignment.

You will turn in your code and a pdf writeup to Gradescope. Please ensure sure that you are able to access Gradescope well in advance of the deadline. You may submit an unlimited number of times before the deadline.

In Gradescope, we will be looking for the following files

- layer.py
- firstlayer.py
- lab2.py
- lab2.pdf
  - Please place each answer on a new page

As this course covers an active research area, we anticipate that you may think of a better way to go about solving a problem than the one that we have come up with. You may edit the starter code and add libraries. Please explain code changes in your submission. If your solution adds undue complexity or trivializes any problem, we will deduct points from the corresponding problem. If in doubt, ask the TAs!

The grading rubric for your deliverables will be based on their correctness, the quality of your justifications, and style/readability.

1 Teamwork is the ability to work together toward a common vision - Andrew Carnegie (10 points)

You may work in teams of up to 3 contributors for this assignment. Please indicate who you worked with in your submission.

You should each individually fill out the below survey by the submission deadline. You should fill out the survey even if you choose to work on this assignment as an individual.

Survey responses will be kept confidential to the course staff.

The link for the survey is: https://goo.gl/forms/AvwpxOY7u4kpRJUh2

2 Spike Response Models (20 points)

Hodgkin and Huxley won the Nobel Prize in Physiology or Medicine for their foundational work with neurons. They developed a model, namely, the Hodgkin-Huxley model, which is the classic neuron model adopted widely in neuroscience research. Although it is biologically accurate, a simpler model which can characterize a neuron's functional input-output spiking behaviour would be more useful, from a computer architecture perspective.

As you learned in your lectures, a simpler Spike Response Model was derived from discretized Hodgkin-Huxley model by removing certain "second-order" terms. To what extent is it actually second-order? Perform a literature search to find realistic constant values and compare the "second-order" terms with the "first-order" terms retained in the Spike Response Model.

Please provide your comparison analysis along with the citations in `lab2.pdf`.

3 Getting Into Inhibition! (25 points)

3.1

In `lab2.pdf` please explain the functional difference between FeedForward Inhibition (FFI) and Lateral Inhibition (LI).

3.2

In `Layer.py` we have defined a class for you called `Layer`, and have included the names of a few functions to complete. You may create additional functions as required.

For this lab assignment, you can consider your network as composing of two layers

1. A first layer that you mostly wrote in lab 1. You should be able to use it here with minor tweaks to your code. Discuss your Lab 1 solution with your teammates and come to a consensus on how you would like to implement your first layer.

2. A second layer that we are going to start building in this assignment. You are not required to implement Spike Timing Dependent Plasticity (STDP) or a correlator in this assignment.

Throughout this lab (Sections 3 4), use a 3x3 receptive field, tiled across the inputs. If 2 filters are applied at the first layer, this would thus be $(3 * 3) * 2 = 18$ inputs to a neuron.

3.3

Consider a function `process_image`, that will process a single image. This function will control the different steps that we would like to take for a single image, whether it is feed forward inhibition followed by an excitatory column, or another network structure.

As one step within this function, implement feed forward inhibition.

3.4

In `lab2.pdf` list any parameters that you had to pick, the values that you picked for them, and why you picked those values.

4 Extra Excited about Excitatory Columns! (45 points)

4.1

Finish up the function `process_image`. Calculate the levels of excitation in a layer for a single image and generate an output spike whenever a threshold is reached.

You may set the weights associated with synapses to be a constant, and do not need to learn new weights in this assignment.

4.2

Write a helper function `reset` that will allow you to reset the network (clear out accumulator variables, etc). In particular, you will want to use this function between presenting different images to the network.

4.3

In `lab2.pdf` report what response function you used. We recommend starting with a Step response function with no leak. However, you are free to implement other response functions as well. If you use a response function not discussed in lecture, please provide any relevant citations.

4.4

In `lab2.pdf` report the spiking threshold you chose. Why did you choose this threshold?

4.5

Handin a file `spiketimes.csv` that represents the spiketimes that your new layer outputs for the 3x3 receptive field at (12,12) for all images. Use the following format:

image_number	spike_position	spike_time