18-847 Lab Assignment 1

Due 2019-01-31 11:59pm PST
Due 2019-02-01 02:59am EST
100 Points

This lab assignment will ensure that you are able to use Neuromorphic Computing terminology, help you get started on manipulating images from the MNIST dataset, and ensure that you are able to visualize your outputs.

This lab is an individual assignment. Teams are not permitted. You may discuss concepts pertaining to this homework with classmates, but your submission must reflect your own work.

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

- firstlayer.py
- lab1.py
- spiketimes.csv
- lab1.pdf
  - Please place each answer on a new page

1 A Bit About Columns (25 points)

Macro-columns have a number of different names, depending on where they are found. In the somatosensory cortex of rodents, they are referred to as "barrels". In the visual cortex, they are "hypercolumns".

1. How many neurons are in a barrel column? What fractions are excitatory and inhibitory? How many micro-columns are in a barrel column?

   Document your answer. What sources from the literature are used? Provide tables or other data from papers along with the method used to arrive at the numbers. The numbers themselves are less important than the way you arrive at and document the result. Do this entirely on your own. Google scholar is a good place to start.

2. Repeat for hypercolumns.

2 Set Up Your Environment

Starter code can be found on Canvas and sections that you are required to complete are labeled. You can receive full credit on this assignment without changing any additional sections of the starter code.

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 starter code was written using Python 3.6. We use the following Python libraries, all of which can be installed using standard Python package managers and will work with Anaconda.

- numpy
- matplotlib.pyplot
- sklearn.datasets
- scipy

Running the starter code should exit cleanly when run from a terminal:

```
> python lab1.py
```

3 On/Off Center Filtering (25 points)

The first step in the network that we will build across the homeworks in this class is going to be to apply On Center and Off Center filters to the MNIST dataset in order to represent computation done in the retina. The preprocessing step should perform the following:

3.1

In the file `firstlayer.py`, define the function `preprocess`. The function should assume that input data is formatted as an array of shape `(num_images, image_height, image_width)`, and that the filter for the receptive field (On Center, Off Center, etc.) will be provided as an input to the function.

1. Scale the data to be 3 bits. In `lab1.pdf`, explain why we might want to perform this step.
2. Apply the filter to each receptive field in the image, and apply a reasonable strategy to handle the border of the image.

3.2

Define three filters, `oncenter`, `offcenter`, and a filter of your choosing. Apply your pre-processing function to the data using your filters. Explain what your own filter brings out in the image.

(For example: "I implemented a filter that highlights vertical left edges. This might be an interesting filter if....")

4 Visualize Your Outputs (20 points)

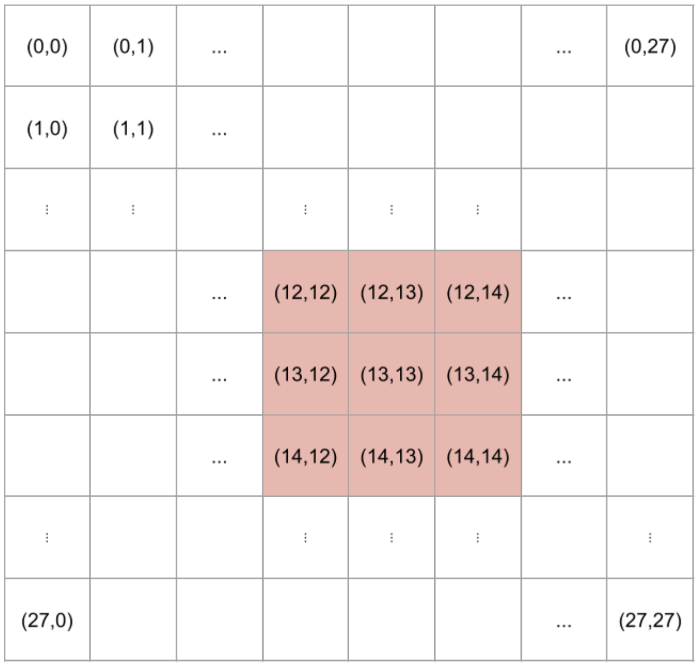
4.1

Select two images from the dataset, and in `lab1.pdf` show the original image and the resulting image after applying all three filters to the image. Please ensure that each image is labled. (4 images in total).

5 Spiketimes (30 points)

5.1

In `firstlayer.py` write a function that will generate a volley of spiketimes for the 3x3 receptive field whose top left corner is at (12,12). In other words, if we index into the image as shown in the diagram below, use the highlighted pixels as your receptive field:



At this time, you do not need to implement any learning or structures to correlate spikes.

We will be looking to see that you implement a reasonable representation of time.

A suggested way to structure this is:

1. Write a function that will convert all the on-off filtered images into a spiketime format.
2. Write a function that will output the spiketimes for a given receptive field

5.2

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

image_number	**spike_position**	**spike_time**