CS/NEUR125 Brains, Minds, and Machines

Lab 1: Neurons and the Brain

Due: Wednesday, February 1

This lab explores some properties of neurons and the functional organization of the human brain. Questions about neurons build on background provided in the section entitled, "Neurons," from this online Introduction to Neurons and Neuronal Networks by John Byrne. Questions in the section called "The Brain" draw from this online tool for visualizing the brain in 3D that includes useful information about the structure and function of each area of the human brain.

You will work on this lab with a partner. To begin, one of you should create a copy of this Google document, share your copy between the two of you, and modify the title of your copy (displayed in the upper left corner of this window) to include both of your names. More information about this process can be found on this webpage that is linked from the Lab 1 spot on the course schedule. You and your partner can answer the bulleted numbered questions below by typing directly into your copy of this Google doc. When you have completed the Lab activities and write-up, you can then submit your work by sharing this document with Mike (mwiest@wellesley.edu) and Ellen (ehildreth@wellesley.edu). Be sure to give us permission to edit the document so that we can provide comments on your work directly in the electronic file.

In this lab we will

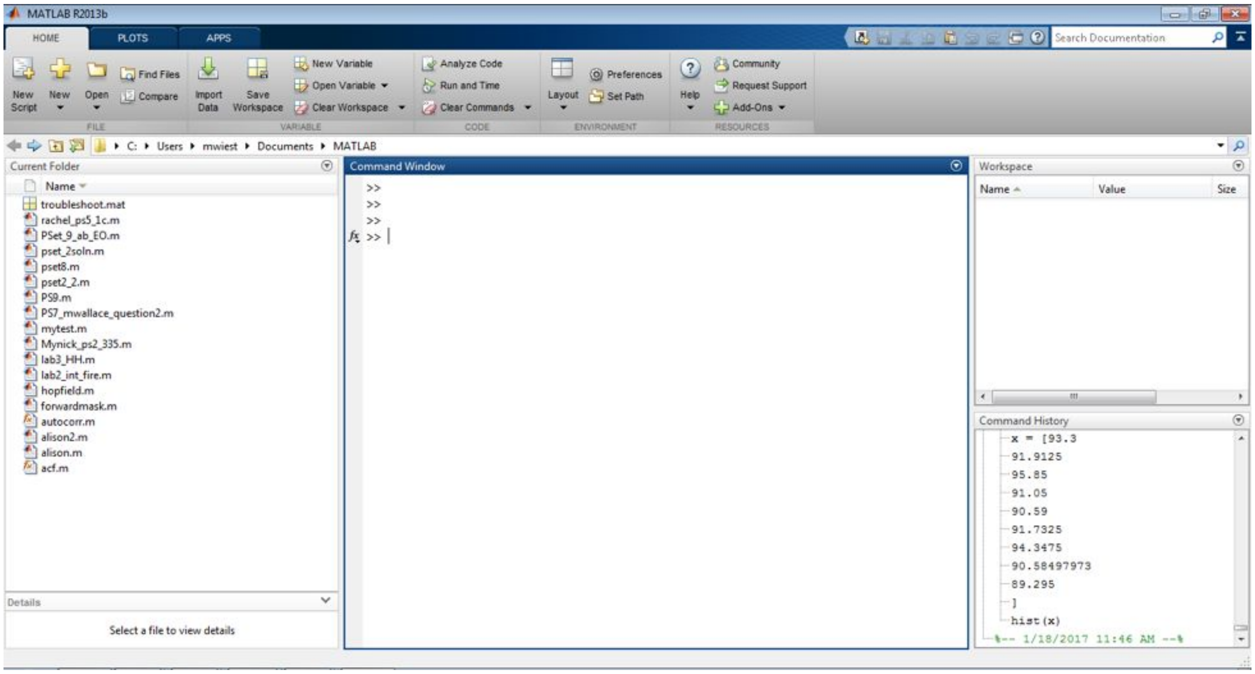
I. Get a first taste of the MATLAB programing environment and use it to calculate and plot **firing rates** that we use to quantify a neuron's electrical activity, and
II. Use an online 3D neuroanatomy explorer to begin to get a sense of the human brain's layout, and a few spots in it that we will focus on in this course.

I. Neurons: Action potential, Firing rate, and a Taste of MATLAB

Neurons are a special type of cell in the brain and peripheral nervous system, which are able to communicate rapidly with each other and with muscle cells at large distances by pulses of electric polarization across their membrane, called **action potentials** (or **spikes**). We think our behavior (intelligent or otherwise) is controlled by these action potentials and our mental life is intimately—but mysteriously—related to them.

MATLAB is a programming environment that we can use for making plots, analyzing data, and even simulating neurons or intelligent behavior. Today we will just learn to open MATLAB, do a few simple calculations, and make a figure of our results. To open MATLAB, go to the Go menu at the top of your screen and select Applications. Inside the Applications folder, find the icon labeled MATLAB_R2015b and double-click on this icon.

When you open MATLAB, the following screen (or something very similar) should appear:



```
>>
>>
>>
fx >> |
```

```
x = [93.3
91.9125
95.85
91.05
90.59
91.7325
94.3475
90.58497973
89.295
]
hist(x)
%-- 1/18/2017 11:46 AM --%
```

The most important of the inner windows is the **Command** window, where you will type all instructions to MATLAB.

The **Workspace** window lists all variables and their sizes and class (array, string, etc.). This window can be useful in debugging (i.e. finding the errors in) your code.

The **Command History** window tells you the most recent commands that were entered. You can double-click on a previous command to run it again (to save you re-typing).

You can open or close these windows by choosing the items from the **Layout** menu.

The **Current Directory** is displayed in the long horizontal box below the top toolbar of the program. This is the default directory that MATLAB uses when you save your work or open files. You should set this to the directory (e.g. the Desktop or a folder) in which you plan to locate your work.

Using MATLAB as your calculator

At the most basic level, MATLAB can be used as your calculator with **+**, **-**, *****, and **/** representing addition, subtraction, multiplication, and division, respectively.

Let’s play a bit with this (type along when you see the prompt “>>”):

```
>> 2+2

ans =
        4
```

“ans=” is MATLAB shorthand for “the answer is…”.

Assigning variables

A *variable* is a storage space in a computer for numbers or characters. The equal sign “**=**” is used to assign a numerical value to a variable:

At the command prompt, type:

```
>> a = 5
```

You just created a variable named a and assigned it a value of 5. If you now type

```
>> a*5
```

you get the answer 25.

Now type

```
>> b = a*5;
```

You just made a new variable b, and assigned it a value equal to the product of the value of a and 5. By adding a **semicolon**, you suppressed the output from being printed. Suppressing the output will be important later. To retrieve the value of b, just type

```
>> b
```

without a semicolon.

Now try typing

```
>> a = a + 10
```

This re-assigns to the variable a the sum of the previous value of a (which was 5), plus 10, resulting in the new value of the variable a being 15. *Note: this illustrates that ‘=’ means ‘assign’ in MATLAB. It is not a declaration that the expressions on the two sides of the ‘=’ sign are ‘equal’ (because certainly a cannot equal itself plus 10!). We will see later that a double equal sign “==” is used when we want to test if two quantities are equal to each other.*

Vectors and Matrices

In MATLAB you can store and manipulate numbers in tables called arrays or matrices. The numbers are arranged in rows and columns and so the size of a matrix is the number of rows and columns it has. If an array is just one column or one row we call it a vector. (These terms have more technical meanings in more advanced math.)

Let’s see more about how arrays work. Type

```
>> a = [1  2  3  4  5]

a =

     1     2     3     4     5
```

You have just created an array containing the numbers 1 through 5. This array can be thought of as a matrix that is 1 row deep and 5 columns wide (often called a *row vector*). To confirm this, use the **size** command which tells you the number of rows (first element returned) and number of columns (second element returned) in an array:

```
>> size(a)

ans =

     1     5
```

To obtain the number of elements in a vector, use the **length** command:

```
>> length(a)

ans =

     5
```

You can also add or multiply arrays by a constant, or add arrays. Try:

```
>> b = 2*a

b =
     2     4     6     8    10

>> b+a

ans =

     3     6     9    12    15
```

You can access the value of a given element of a or b by using parentheses. Type

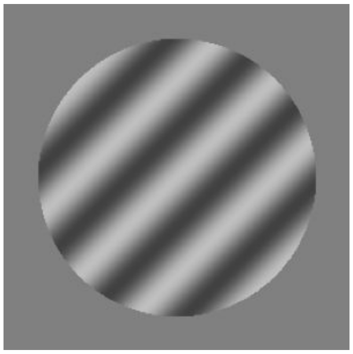
```
>> a(3)
>> b(4)
```

and you see the values of the 3rd element of the a array and the 4th element of the b array, respectively. You can also use b(end) or a(end) to see the last element.

Stimulus-response plot a.k.a. Tuning curve

Now that you know some MATLAB basics, let's apply them to calculate and plot the firing rate response to each different stimulus in a pretend electrophysiology experiment. Imagine that we have surgically implanted a thin microwire into the brain of an awake monkey, and the microwire sits close enough to a particular sensory neuron that we can record action potentials—1 millisecond-long spikes in voltage—from that neuron. Our experiment is to present a series of visual stimuli to the monkey and note the number of action potentials generated by our cell in the 5 seconds after the presentation of each stimulus. The *purpose* of our experiment is to characterize what stimuli are "preferred" by this neuron: that is, what stimuli cause it to fire the largest number of spikes.

Suppose our stimuli are oriented visual "gratings" like the one shown below, which illustrates a grating tilted 45 degrees from horizontal.



Suppose we present gratings with the following tilt angles, in order: -90, -85, -80, -75, …, +85, +90. MATLAB makes it easy to make a list of numbers from one number to another in regular steps. Here's how you would make a vector x that counted by twos from 10 to 18:

```
>> x = 10:2:18;
```

Using colons as above create a vector called "stimangles" that contains the correct list of stimulus angles given above.

How many stimulus angles are there? (Hint: use the length command.)

Next create a variable called "counts" to store your spike counts in it, in the same order as the stimuli that generated each response. Suppose the number of spikes we recorded in the 5 seconds after each stimulus were, in order: 10 11 9 10 10 10 9 10 9 9 9 11 9 10 9 10 7 9 9 9 14 20 26 29 31 30 27 20 14 6 10 9 8 11 12 10 10.

The spike counts do reflect the neuron's stimulus preference. However, they would be hard to compare to the results of someone else's experiment if they stimulated for a different length of time. For this reason it is often useful to express a neuron's response in terms of a **firing rate**, which is the average number of spikes fired per unit time. Its units are Hertz, abbreviated Hz, which means "per second." In this context it means "spikes per second." To calculate the firing rate responses define a variable called “rates” and set it equal to the spike counts divided by the duration of the stimulation.

We can plot the rates as a function of the stimulation orientation angle by using the plot command:

```
>> plot(stimangles, rates)
```

This plot function treats the stimangles numbers as the x-axis and the rates as the y-axis values and connects the points with lines. You can plot with different symbols and styles using options like

```
>> plot(stimangles, rates, ‘r--‘)
```

The ‘r--‘ optional input causes the plotted curve to be a dashed red line. Type ‘help plot’ for more information about the plot command.

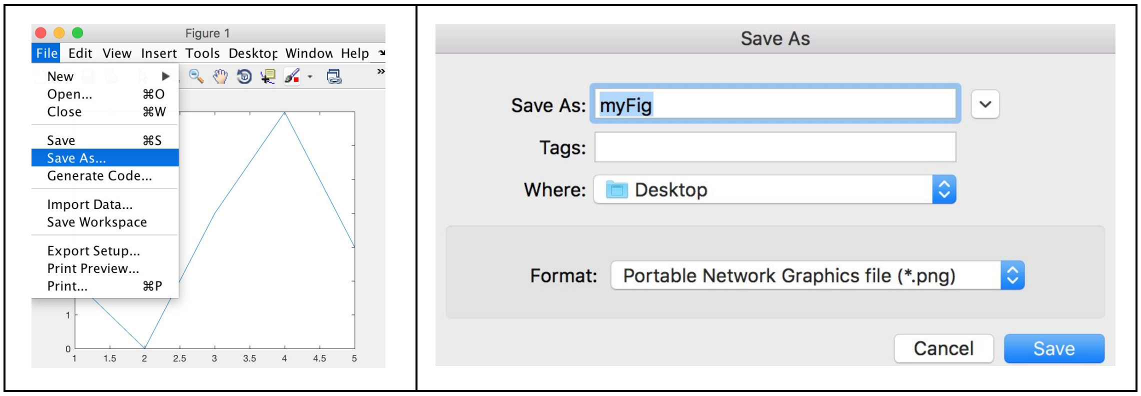
You can label the x-axis and y-axis like this (note the use of single quotes for text):

```
>> xlabel(‘Orientation relative to horizontal (deg)’)
>> ylabel(‘Firing rate (Hz)’)
```

And you can give your plot a title like this:

```
>> title(‘Firing rate response vs. Grating angle’)
```

- **Q1.** Please **paste this figure** into your lab document to turn in with answers to the other numbered questions. Go to the File menu on the Figure window and select Save As to save the figure as a .png file on your Desktop with a name of your choosing, as shown in the pictures below. Then drag the file from your Desktop into your open Google doc.



Save As: myFig

Tags:

Where: Desktop

Format: Portable Network Graphics file (*.png)

We can use the max function to determine the preferred stimulus angle for the neuron we recorded from. In the Command window type

```
>> [ mx, mx_ind ] = max(counts)
```

The max function takes a vector as input (in our case the input vector is counts), and returns two outputs, called ‘mx’ and ‘mx_ind’. mx is the maximum value out of the numbers stored in counts--that is, the maximum firing rate we recorded in our experiment. But what we’re looking for is which stimulus (which angle) caused the neuron to fire that maximum rate. To find that we’ll use the other output of the max function, mx_ind. That one tells us the “index” of the maximum count in the counts vector. That means whether it is the first or eighth or fourteenth number (“element”) in the vector.

Suppose the index of the maximum, mx_ind, is 14. That means the max count was the fourteenth count, which means it was the response to the fourteenth stimulus angle, since we kept them in order. So the preferred angle for this neuron would be the 14th angle in the list (vector) of stimulus angles.

- **Q2.** What is the preferred stimulus orientation for our neuron based on the actual data you analyzed?

Action potential

Let’s make another plot to practice a bit more with MATLAB and review some features of action potentials. This time instead of typing our data we’ll load them from a file. Cut and paste the following two lines into your Command window:

```
urlwrite('http://cs.wellesley.edu/~cs125/spikeeg.mat', 'spikeeg.mat');
load('spikeeg.mat');
```

The first line copies a file stored online to your computer; the second line loads that file into your MATLAB workspace. Data files containing MATLAB variables generally end with the suffix ‘.mat’.

Once you’ve loaded spikeeg.mat into your workspace you should see its variable appear in your Workspace panel. You can also type “who” to see a list of variables in your workspace, or “whos” for a list that also shows you the size of each variable. The vector ‘spike’ contains the membrane potential in millivolts during a 5 millisecond simulated recording. The vector ‘tax’ contains a time axis for plotting the voltages in spike.

Plot the spike using the plot command and the given time axis. Use the xlabel and ylabel commands to label the axes, including units.

At t=1ms the neuron was stimulated by an injection of positive current from a stimulating electrode for 1ms, resulting in the single action potential that we see in the plot.

- **Q3.** What is the resting potential of this simulated neuron?

- **Q4.** When the membrane potential is negative, does that mean there is more net negative charge inside the cell or outside the cell?

- **Q5.** From your plot estimate the threshold of this neuron.

Use the text command to label the depolarization and repolarization phases of the action potential. For example, text(2,20, ‘myLabel’) will put the text “myLabel” at the coordinate (2, 20) on your plot. (Or you can type ‘help text’ to see the **syntax** for the text function.)

- **Q6.** Include this figure in your Lab 1 document as you did with the figure in Q1.
- **Q7.** What do we call it in general when the membrane potential moves towards a more negative voltage (hint: NOT repolarization)?

OK, let’s close MATLAB and turn to consider the whole brain, with its hundreds of billions of neurons connected in intricate architectures.

Additional questions to answer later (or now):

- **Q8.** Briefly describe how an action potential in a presynaptic neuron can cause a change in a postsynaptic neuron’s membrane potential even though the two cells are not touching.
- **Q9.** True or False: Both neurons and liver cells have nonzero membrane potentials.
- **Q10.** What do action potentials allow neurons to do that other cell types cannot do?

II. The Brain

Open the online tool for visualizing the brain in 3D. Review the locations and basic features of the four lobes of the outer cortex of the brain by looking at the Whole Brain and frontal, parietal, occipital, and temporal lobes from the 3D Brain menu. The brain can be considered to be a sensory-motor organ, *perceiving* the state of the world and *acting* back on it.

- **Q11.** Read the Overview section for each of the four lobes. Which lobes seem to be mostly for perception, and which specializes in action? Briefly support your answer for each lobe. Which brain lobe houses the part of cortex that specializes in vision?
- **Q12.** Click on Main Menu in the upper left corner, go to Cognitive Processes-Language, and then click on Brain Anatomy in the lower “subway line” menu near the middle of the page, to find two brain areas involved in language. Then go back to the 3D Brain to locate those structures (use the Main Menu or use the subway menu at the top of the page to go directly to each area). Which cortical lobe is each structure found in? What specific language functions are attributed to each of the two areas? Do these two sub-functions correspond to “action” or “perception” according to how you labelled these two brain-lobes in the question above? Can you imagine what the first evidence for these areas having these functions might have been?

- **Q13.** Go to Main Menu-Cognitive Processes-Learning and Memory and click on Brain Anatomy in the mid-page subway line to find two brain areas involved in learning. What are they?

Go back to the 3D brain to locate them.

- **Q14.** What lobe of the brain is the hippocampus most closely associated with? It is difficult to see the answer on the 3D brain, so you may want to Google this question.

- **Q15.** The amygdala is a small almond-shaped structure near one end of the hippocampus on both sides of the brain. Is the amygdala near the **anterior** or **posterior** hippocampus?

- **Q16.** Aside from learning and memory, what is another primary function of the hippocampus? Can you see the different functions as related in any way?

While we think of the brain's cortex (including hippocampus) as responsible for our conscious mental life and higher cognitive functions, the brainstem (and hypothalamus) controls some of our most basic life-sustaining automatic and primal functions like breathing. Select Brainstem from the 3D Brain menu to see where it is and what it looks like.

Now go back to the Temporal Lobe and click on Show Labels in the lower right. Notice that the label for "Fusiform cortex" appears to be pointing at the brainstem rather than to any part of cortex. To see what that label is attempting to point at, use the left arrow button in the lower right to turn the brain until you can see through the transparent hemisphere to the inside--**medial**--part of the temporal lobe that was curled under and out of view from the other side. Then you can see the fusiform gyrus as the lowest **gyrus** in green. If you hover your cursor over it you will see the label.

In this course we'll spend some time learning about the Fusiform Face Area, which is located in the fusiform gyrus. Neurons there appear to be specialized for recognizing faces. More concretely, using terms we reviewed in lab today, the **preferred stimuli** for neurons in the Fusiform Face Area are faces. That means those neurons fire action potentials at higher **firing rates** when the brain is presented with pictures of faces as compared to when it is presented with pictures of other things.