CS 425/525 Assignment 1

Alfonso Buono (ajb393)
Andrea Cho (aac233)
Kathleen Wilcox (krw93)
Shyam Patel (spp128)

October 27th, 2020

1 Division of Labor

All members of the team contributed in discussions and peer reviews of each others work throughout the project effort. All member participated in the implementation and research of the various models. Upon request, we can grant the grader or the professor access to the private git repository where our source code and commit log can be accessed. The python and notebook files are provided alongside this report in our submission.

2 Questions

1. What do you expect to happen if an IF neuron is fed a very low input current? An LIF neuron?

   If an IF neuron is fed a very low input current overtime the IF neuron will fire but the LIF neuron would not. The IF neuron will fire because it simply compounds current until the threshold has been reached. The LIF neuron will not fire because since current is essentially "leaving" the neuron we approach the spike threshold at a much slower rate than that of the IF neuron.

2. What do you expect to happen if an IF neuron is fed a larger input current? An LIF neuron?

   If either the IF or LIF are fed a larger input current then, depending on the current, they both should fire. The firing rate of both will increase according to the increase in current. This is because at higher currents the LIF is able to minimize the effect of the "leak" due to the increase in current.

3. What are the limitations of an LIF neuron?

   The Leaking Integrate and Fire model seeks to simplify a highly dynamic and complex system. As such, it neglects a multiude of key aspects of neuronal dynamics. One key simplification it makes is with its input; in leaky integrate and afire the input current is integrated linearly when in reality it should consider the postsynaptic state of the input neuron. One other issue with there being very little granulaity pertaining to spikes is the inability maintain a form of "memory" aside from the most recent spike.

3 Programming

3.1 Libraries

We used the following external Python3 libraries: Numpy, Plotly, and Odeint

3.2 Structure

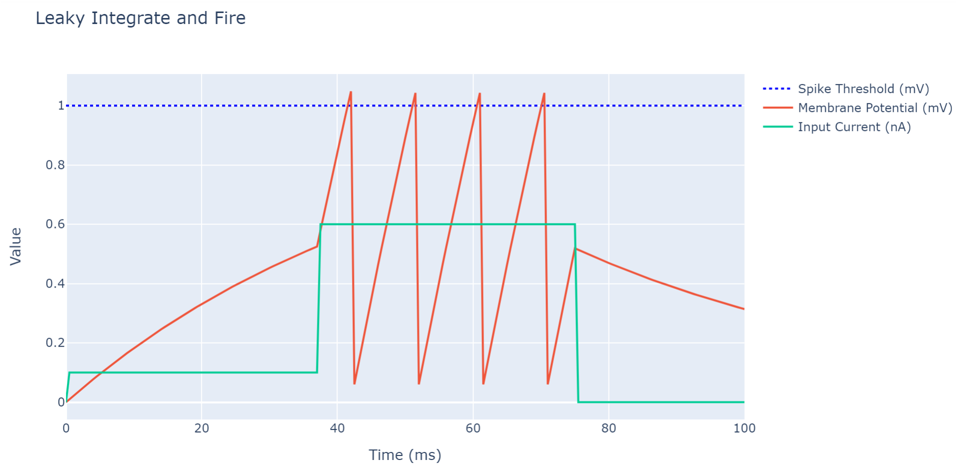
Code Structure
We opted to use an object oriented design where we maintain the following classes: neuron, lif, izhikevich and hodgkinhuxley.
We defined the following classes: lif, izhikevich and hodgkinhuxley. These classes contain the respective constants, values, and helper functions needed for each model.
A fourth class, the neuron class, contains all necessary components for our simulations.

Simulation Structure
In order to simulate a neuron firing for some length of time, t, in milliseconds, we represent a single time step via an array of zeroes that is of length $\frac{1}{step\ size} * t$.
For example, if we opted for 1 time step to be 0.5 milliseconds in a simulation that runs for 100 milliseconds, the array will be of length 200. We also maintain an array of membrane potentials, spikes and input currents that have a 1 to 1 correspondence with our time array. All of these parts are stored within a neuron object. The object also contains the step size, spike threshold, and total time, in milliseconds, the simulation will run for.
All simulations can be easily graphed via the respective plot_graph() function defined within the neuron class

3.3 Questions

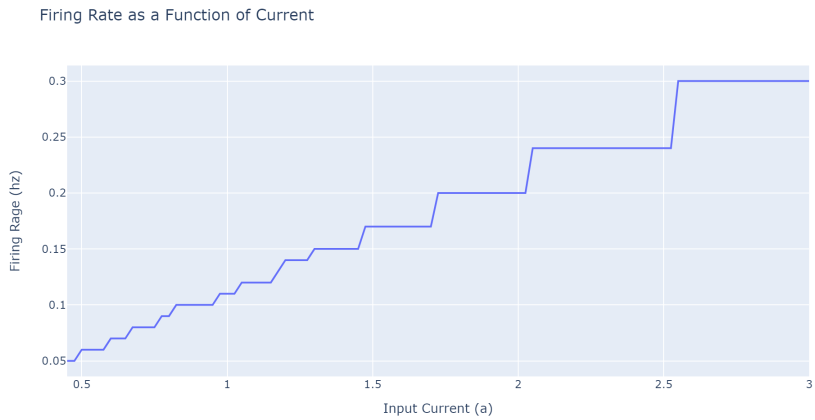
1. Simulate an LIF neuron with different input currents and plot the membrane potential, showing (a) potential decay over time and (b) spiking behavior.

   The graph below is simulated model for 100ms with a timestep of 0.5ms with a resistence of 100 milliohms and an initial input current of 0.1 mv which is then increased to 0.6 mv. The input currents and necessary constants were decided arbitrarily from reading the provided paper, and research online.



2. Plot the firing rate as a function of the input current.

   The graph below is a plot of the number of spikes in multiple 100ms simulation where out timestep = 0.5ms. The start voltage of every simulation was 0.2 mv. The output voltage for the various simulations can be defined via the set, $X = \{\forall i \in Y | 0.45 + i * 0.025\}$ where, $Y = \{\forall z \in Z | 0 \leq z \leq 102\}$

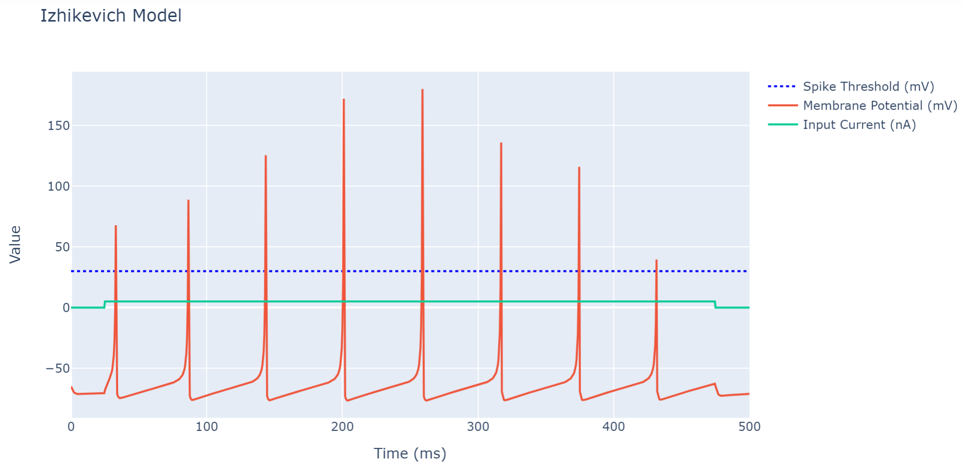


3. What happens to the firing rate as you continue to increase the input current? Why?

   Increasing the current will lead to an increase in the number of spikes because a higher current will make it so that the neuron can increase its membrane potential faster than it would in a lower current situation. Also, it would allow the neuron to increase it's membrane potential quicker than "leak out."

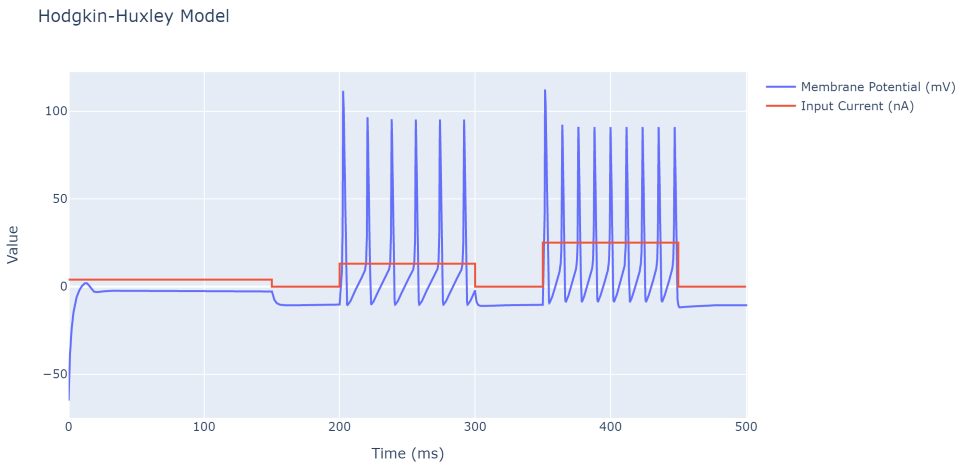
4. Simulate a neuron using the Izhikevich model.

   See the graph below where we simulated the model for 500 ms with a timestep of 0.5 ms. Our constants for the following graph are as follows: a = 0.02, b = 0.2, c = -65 mv, d = 8, spike threshold = 30 mv. Our input current was set to 5 nAmps. The constants were all obtained from Izhikevich's paper.



5. Simulate a neuron using the Hodgkin-Huxley model.

   See the graph below where we simulated the model for 500 ms with a timestep of 0.01 ms. Our constants for the following graph are as follows: membrane capacitance is 1, initial membrane voltage is -65 mV, initial probability for an n gate being open was 0.32, initial proability for an h gate being open was 0.6, initial probability for an m gate being open was 0.05, equilibrium potentials for Na, K, and L are as follows: 115, -12, and -10.613. Lastly, the conductances for Na, K, and L are as follows: 120, 36, and 0.3. These values were taken from "A quantitative description of membrane current and its application to conduction and excitation in nerve" by Hodgekin and Huxley.
   As one can see in the graph below depending on the input current, different amount of spikes occured. In the first input current area, the membrane voltage plateaued and was unable to spike. Once the input current dropped to 0 and the neuron was able to reset when we gave it a higher input current the membrane voltage spiked. As you can see the first spike has a higher voltages but as it goes on it has a little bit less Volts. Lastly, after the input current drops, to reset the neuron, and then jumps to the highest amperage, you can see that the membrane voltage spikes even more frequently.



   Some sources that were useful and utilized during the implementation of this model are:

   - Heitler, W J. The Hodgkin-Huxley Model for the Generation of Action Potentials. The Hodgkin-Huxley Model, www.st-andrews.ac.uk/~wjh/hh_model_intro/.
   - Information and Tutorials. Information, users.sussex.ac.uk/~tn41/HHNeuronHellawell/info.html.