CS 425/525: Brain Inspired Computing **(Due: 02/29/20)**

Assignment 1 - Report

Members of the group:

Nicolas Carchio (nfc28), undergraduate
Dan Sumetsky (ds1461), undergraduate
Zain Sayed (zjs15), undergraduate
Yashshree Patil (yap14), graduate

1. **Contributions of each team member:**

   Dan - code for LIF, Izhikevich, HH
   Yashshree - research on HH, compiling report
   Nicolas - code for LIF, Izhikevich, HH
   Zain - research on HH, theoretical questions

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

   An IF neuron would eventually fire because the voltage always increases at a constant rate until a spike occurs if the current is constant. For an LIF neuron, the membrane potential could end up asymptoting at a voltage below the spiking threshold if the constant current is too low. For both neurons, the membrane potential never goes down unless there is a spike or the current is decreased.

   b)What do you expect to happen if an IF neuron is constantly fed a larger input current? An LIF neuron?

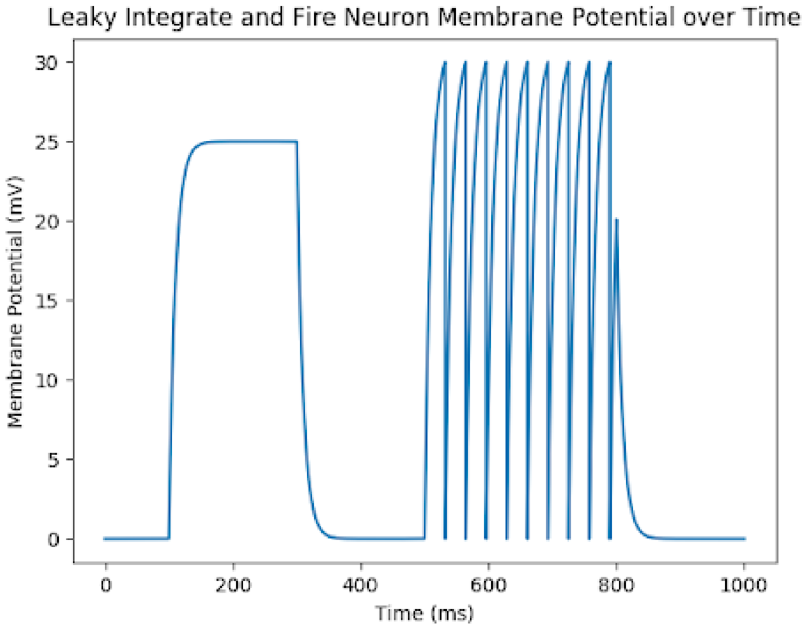
   An IF neuron would fire frequently because the voltage always increases at a constant rate until a spike occurs if the current is constant. A LIF neuron would also most likely fire frequently because it is unlikely that the leak could keep up with the constant increase in voltage.

   c)What are the limitations of an LIF neuron?

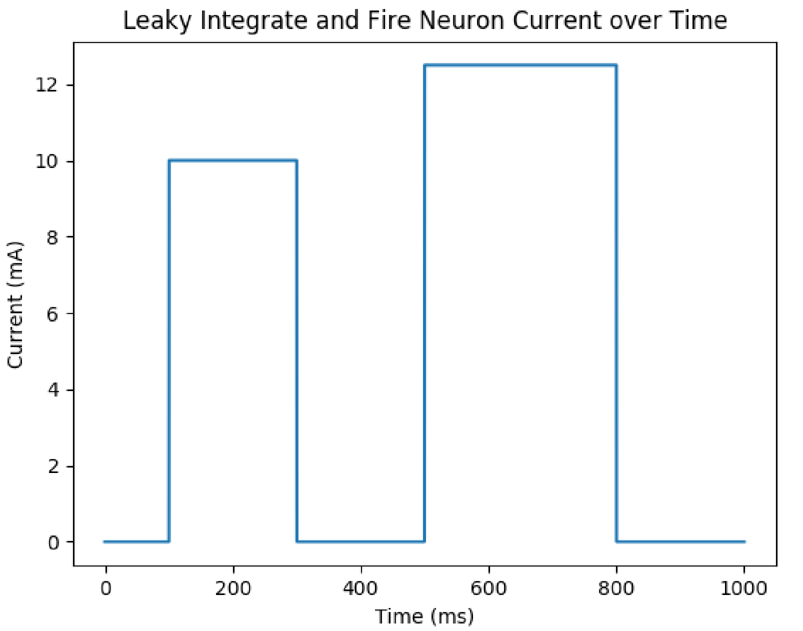
   After each output spike, the membrane potential is reset, no memory of previous spikes are kept. Most neurons take into account a spike train where it waits for the frequency and amplitude of spikes to be consistent before acting on it while an LIF neuron has no memory and cannot account for steady state spiking therefore inhibiting it from capturing adaptation. An LIF neuron does not accurately model a real neuron as it does not have a refractory period.

2. **Programming**

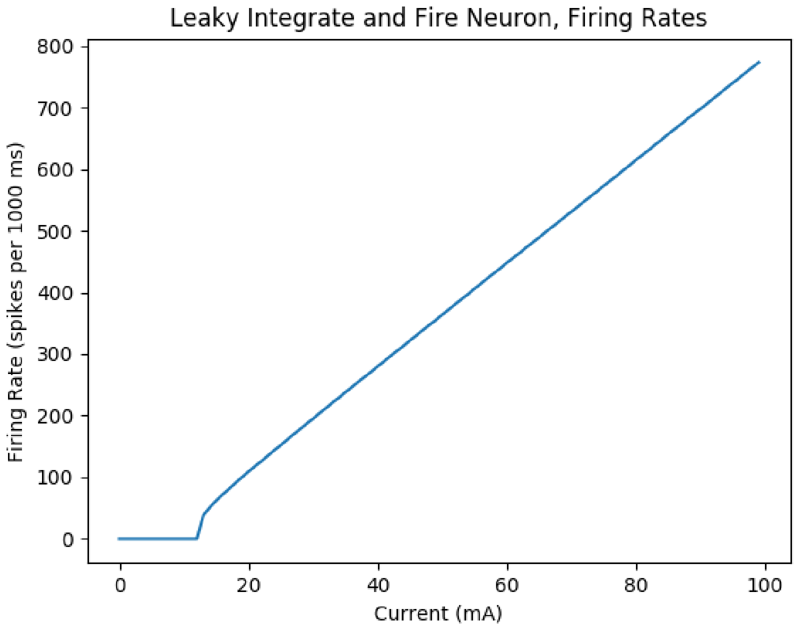
a) i. Plot of voltage over time



ii. Plot of current over time



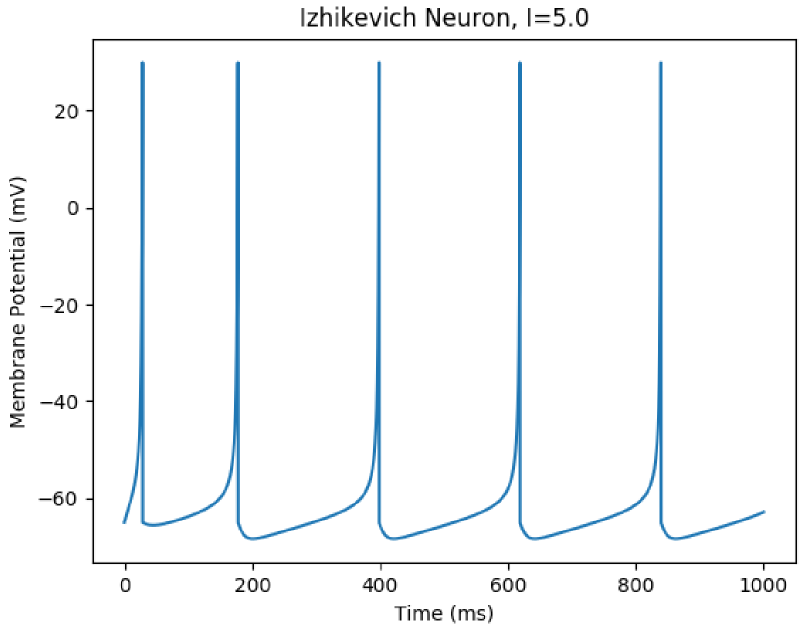
b) Plot of current vs firing rate



c) In a LIF neuron, the firing rate initially doesn't increase at all because the current isn't large enough to beat the leak. Then, the firing rate starts to constantly increase as the current increases. This is because the higher the current, the faster the membrane potential rises from the resting potential to the threshold, which can lead to a greater number of spikes in a period of time.

d)**Izhikevich Model**

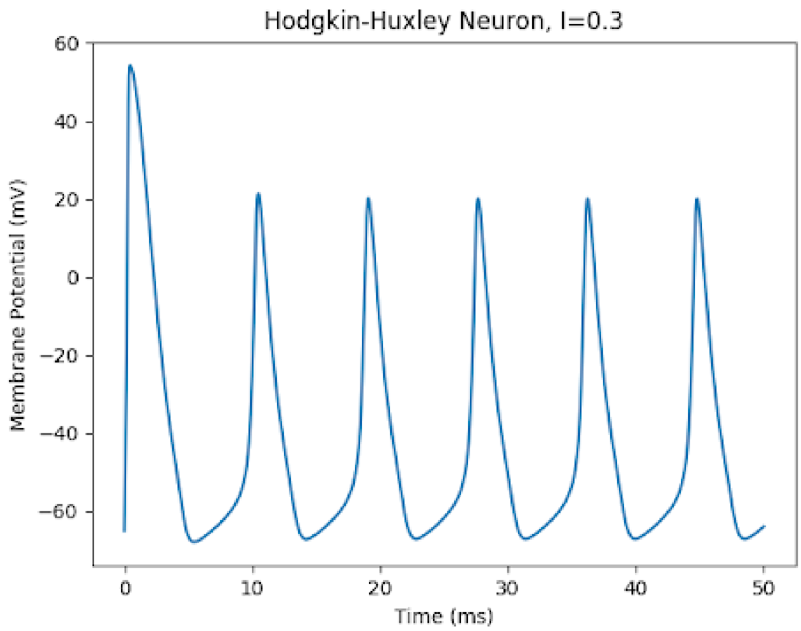
```
class IzhikevichNeuron():
    def __init__(self,a,b,c,d):
        self.a = a
        self.b = b
        self.c = c
        self.d = d
        self.v = self.c
        self.vt = 30.0
        self.u = self.b * self.v

    # run the izhikevich algorithm
    def take_current(self,current,time_diff):
        dvdt = ((0.04* (self.v**2))+ (5*self.v) + 140-self.u+current) / 4
        dudt = self.a * ((self.b * self.v) - self.u) / 4

        # integrate
        self.v += dvdt*time_diff
        self.u += dudt*time_diff

        # if the current voltage has reached the threshold, it spikes and then resets
        if self.v > self.vt:
            self.v = self.c
            self.u += self.d
```



Izikevich improves upon the shortcomings of Leaky Integrate and Fire by introducing two variables for voltage: one to track the cell voltage and another to act as a membrane recovery variable. This is used to model a system more similar to a biological neuron as it introduces the concept of a refractory period, which is shown in a gradual recovery and then buildup before the next spike in the figure above.

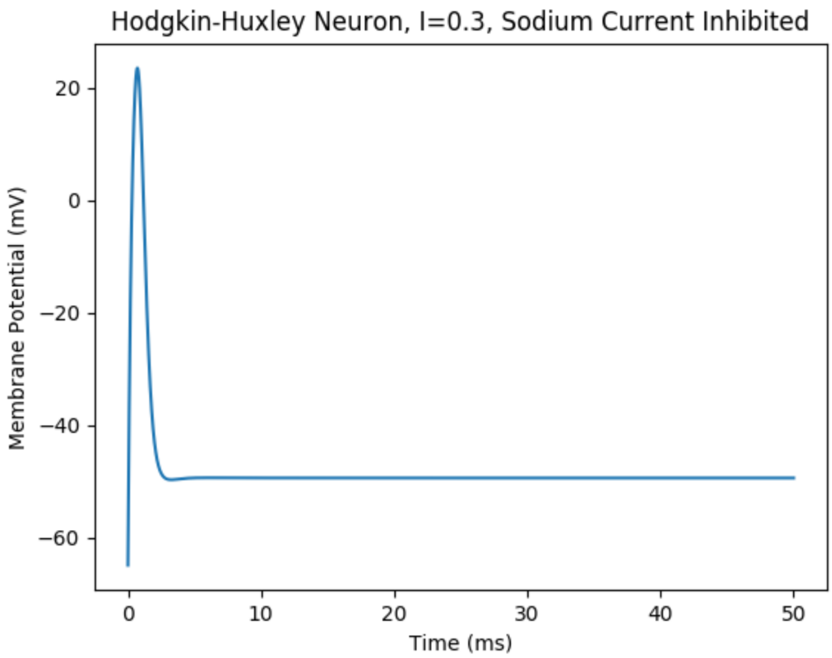
e)**Hodgkin-Huxley Model**

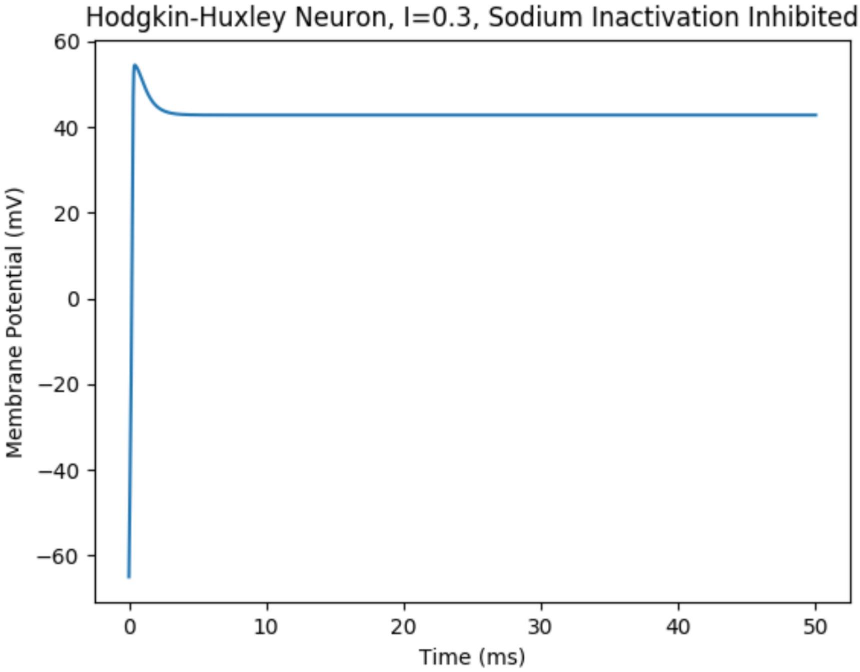
```
# run the Hodgkin-Huxley algorithm
def take_current(self, current, time_diff):
    # define alphas and beta (n, m, h)
    self.alpha_n = (-0.01 * (self.Vm + 60)) / (math.exp((60 + self.Vm) / -10) - 1)
    self.beta_n = 0.125 * math.exp((self.Vm + 70) / 80)
    self.alpha_m = (-0.1 * (self.Vm + 45)) / (math.exp((45 + self.Vm) / -10) - 1)
    self.beta_m = 4 * math.exp((self.Vm + 70) / -18)
    self.alpha_h = 0.07 * math.exp((self.Vm + 70) / -20)
    self.beta_h = 1 / (math.exp((40 + self.Vm) / -10) + 1)

    # differentiate the Vm (voltage)
    dVmdt = (current - (self.gK * self.n**4 * (self.Vm-self.EK) +
                        self.gNa * self.m**3 * self.h * (self.Vm-self.ENa) +
                        self.gL * (self.Vm-self.EL))) / self.capacitance

    # differentiate (n, m, h)
    dndt = self.alpha_n * (1 - self.n) - self.beta_n * self.n
    dmdt = self.alpha_m * (1 - self.m) - self.beta_m * self.m
    dhdt = self.alpha_h * (1 - self.h) - self.beta_h * self.h

    # update vars
    self.Vm += dVmdt*time_diff
    self.n += dndt*time_diff
    self.m += dmdt*time_diff
    self.h += dhdt*time_diff
```



When simulating a Hodgkin-Huxley neuron, we found that after the initial large spike, the resulting spikes share a smaller amplitude and similar membrane potentials. The Hodgkin-Huxley model is clearly a more realistic model than the Leaky Integrate and Fire neuron as it does not reset to a starting voltage of -65mV (causing a steep drop off). Rather, it demonstrates how the ions actually affect the membrane potential in a more gradual, natural way.

3. **Bonus Question:**





Above we have graphs depicting the membrane potential over time of a Hodgkin-Huxley neuron with the sodium current inhibited by the drug TTX and a Hodgkin-Huxley neuron with the sodium inactivation inhibited by the drug pronase, both having a current of 0.3 mA running through them. The way we inhibited the sodium current was to set dm/dt to 0, so that m, the variable controlling sodium activation, is never changed. This resulted in the membrane potential leveling out slightly above the resting membrane potential right after the initial peak. The way we inhibited the sodium inactivation was to set dh/dt to 0, so that h, the variable controlling sodium inactivation, is never changed. This resulted in the membrane potential levelling out slightly below the initial peak right after that peak.

Works Cited:

Izhikevich, E.m. "Simple Model of Spiking Neurons." IEEE Transactions on Neural Networks, vol. 14, no. 6, 2003, pp. 1569–1572., doi:10.1109/tnn.2003.820440.

Hodgkin, A. L., and A. F. Huxley. "A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve." The Journal of Physiology, vol. 117, no. 4, 1952, pp. 500–544., doi:10.1113/jphysiol.1952.sp004764.

Gerstner, Wulfram. "2.2 Hodgkin-Huxley Model." 2.2 Hodgkin-Huxley Model — Neuronal Dynamics Online Book, neuronaldynamics.epfl.ch/online/Ch2.S2.html.

"The Hodgkin-Huxley Model for the Generation of Action Potentials." The Hodgkin-Huxley Model, www.st-andrews.ac.uk/ wjh/hh_model_intro/.