Shahid Beheshti University

Anita Soroush

Abstract

In the previous project we tried to implement a single Neuron.
In this project we are going to connect some of these neurons so that they can affect each other.

Introduction

In this report, I will discuss different types of neuron populations.
In all of these populations, I used the same LIF neuron model that I implemented in the previous project. The only modification is that I have added an integer to the LIF class that can contain the neuron's type; it takes the value of 1 if the neuron is excitatory and -1 if inhibitory.

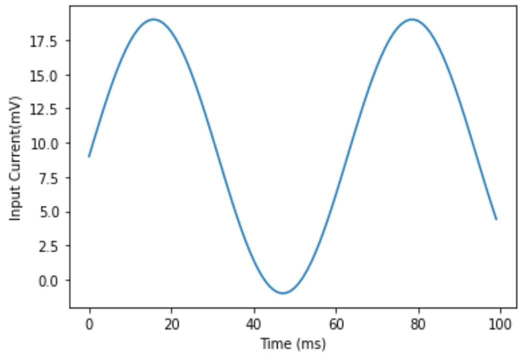
Implementation points:

- Since inhibitory neurons are fast spike so they have smaller time constant of τ. I found $\tau = 8$ for the excitatory and $\tau = 6$ for the inhibitory neurons to be the best values considering other parameters in my models.
- Another point that I have to mention is that in the connections between neurons, no matter what the postsynaptic neuron is, if the presynaptic neuron is inhibitory, the synaptic weight should be a negative value (to mimic inhibition) and if the presynaptic neuron is excitatory, the synaptic weight should be positive (to mimic excitation).

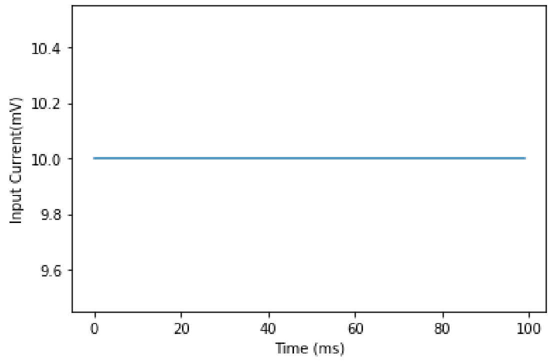
Part 1: A Population Made Up of 2 Neurons

I used 2 following currents:

I1 (sine wave):



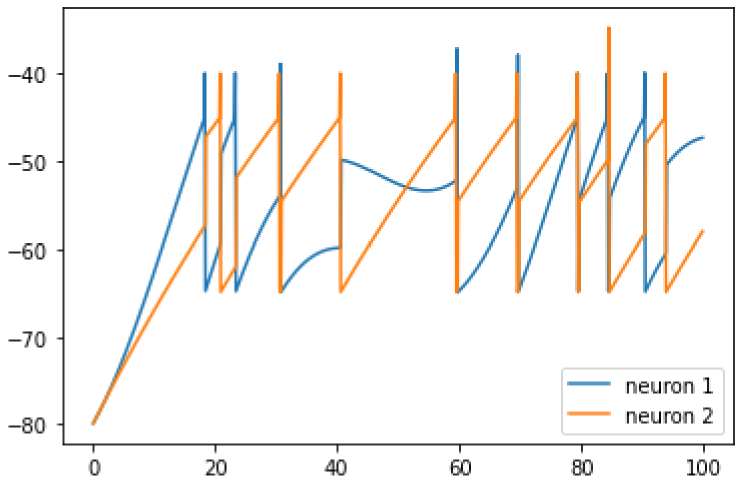
I2 (constant current):



a) Both of them excitatory:

neuron 1 (exc) : I1

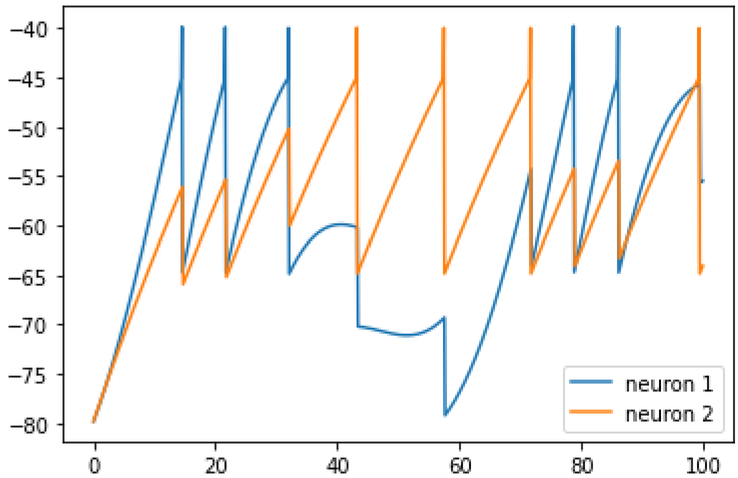
neuron 2 (exc) : I2



b) Both of them inhibitory:

neuron 1 (inh) : I1

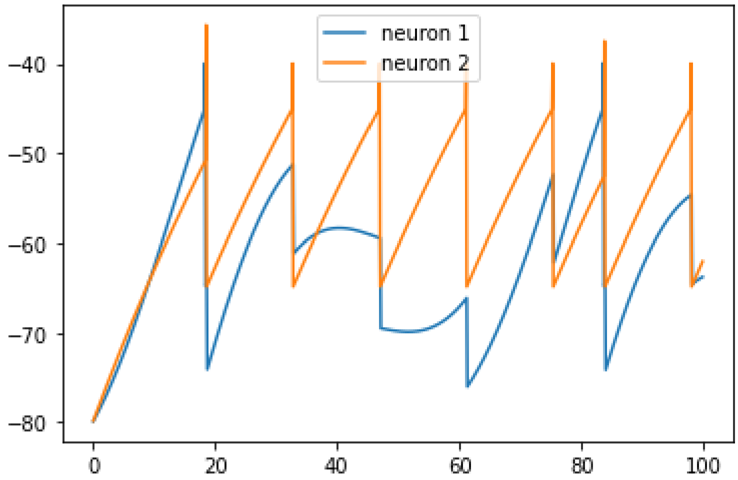
neuron 2 (inh) : I2



c) One of them excitatory and the other one inhibitory:

neuron 1 (exc) : I1

neuron 2 (inh) : I2

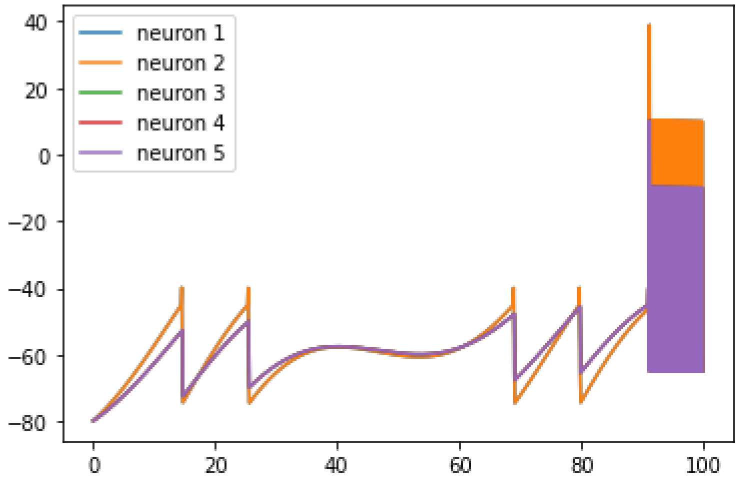


Part 2: A fully Connected Population Made up of 2 Inhibitory Neurons and 8 Excitatory Neurons:

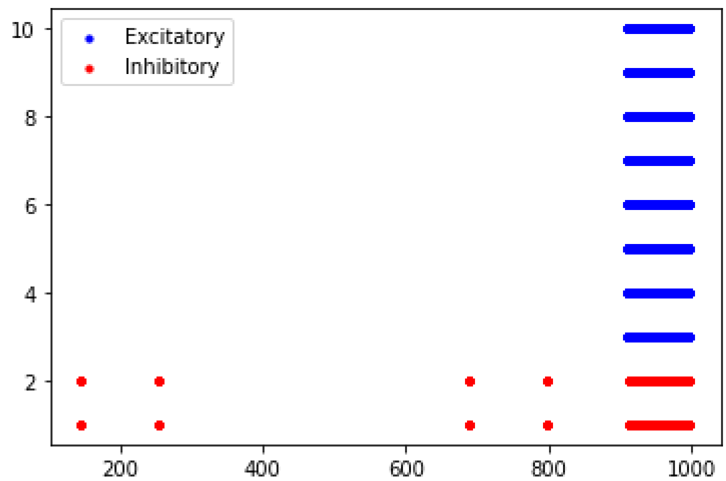
I used the sine wave current in the previous part as the input current of all neurons here.

u_t plot:

(neuron 1 and 2 are inhibitory and the others are excitatory)



raster plot:



Part 3: Simulation of decision making using 2 excitatory and 1 inhibitory population:

When connecting 2 populations, alongside all changes within a group, we should take care of the impacts that each group has on the other one.

I set a spike threshold of 2 which means that in each iteration if the neurons inside a population have more that 2 spikes, the population should affect its connected populations:

This effect is simulated in the following way:

- if the population is excitatory (most of its neurons are excitatory), it should excite the other connected populations (increase their potential)
- if the population is inhibitory (most of its neurons are inhibitory), it should inhibit the other connected populations (decrease their potential)

```
if spikes_count >= self.spikes_threshold:
        if self.majority_type > 0:
          for j in self.connected_neuron_groups:
            for neuron in j.neurons:
              neuron.v += self.connected_neuron_groups_effect
        else:
          for j in self.connected_neuron_groups:
            for neuron in j.neurons:
              neuron.v -= self.connected_neuron_groups_effect
```

I found the best value for `connected_neuron_groups_effect` to be 5.

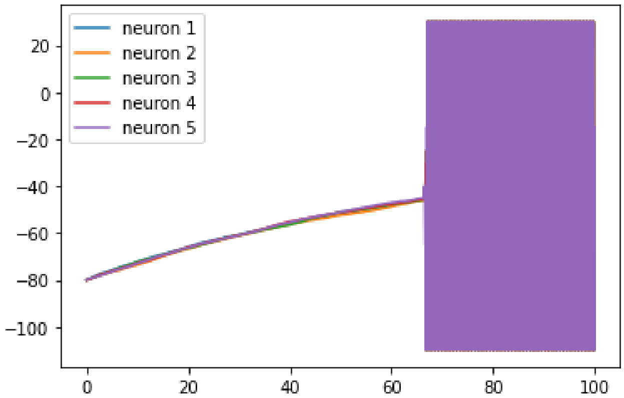
In this part we are asked to make 3 fully connected populations, each of which is made up of 10 neurons. 2 of them should be excitatory and the other one inhibitory. The excitatory populations can receive any external current but the only source of the inhibitory population current is the effect of excitatory populations.

These are the input currents:

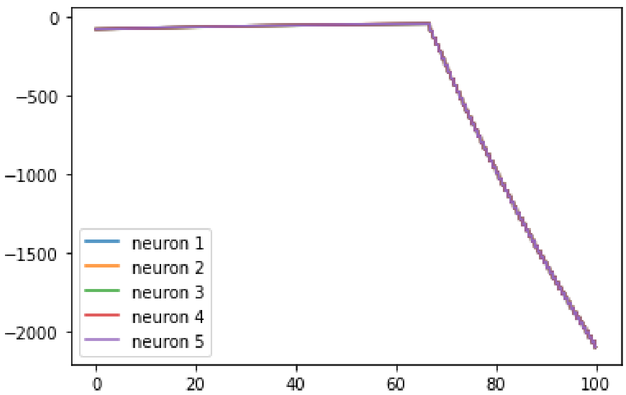
```
I_E1 = lambda x: random.random()*10

I_E2 = lambda x: random.random()*10

I_inh = lambda x: 0
```

These are the u_t plots of the populations:

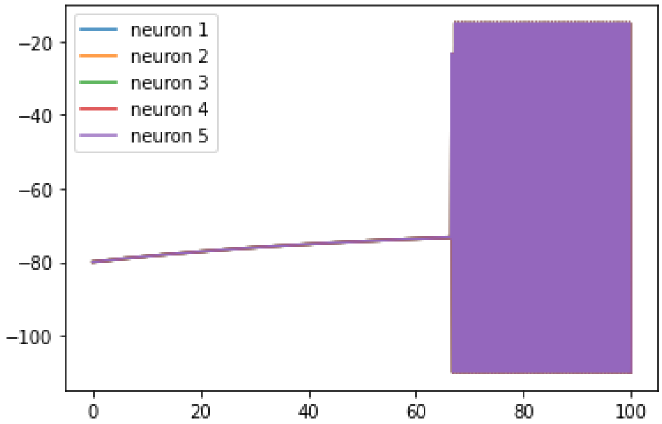
first excitatory population:



second excitatory population:

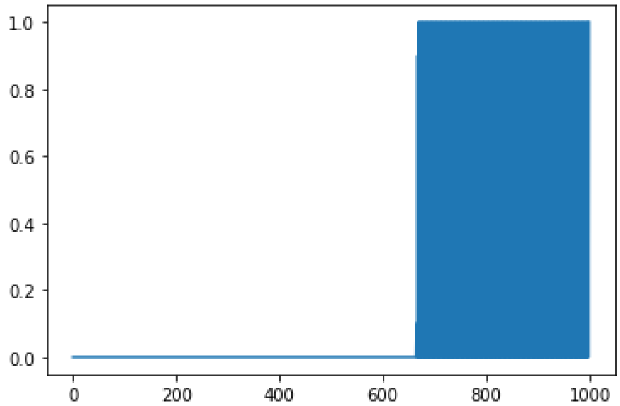


inhibitory population:

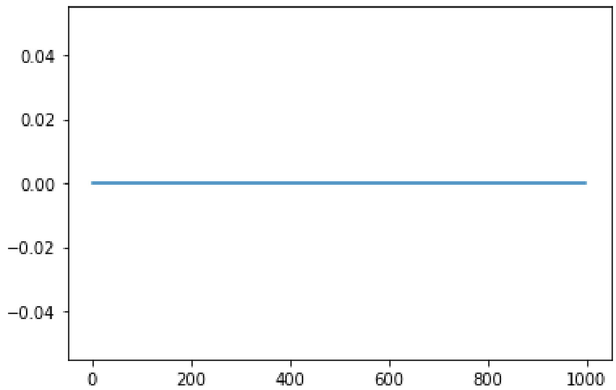


These are the related activity plots:

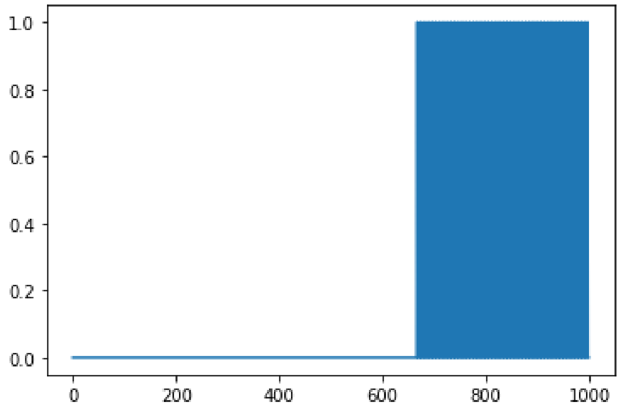
first excitatory population:



second excitatory population:

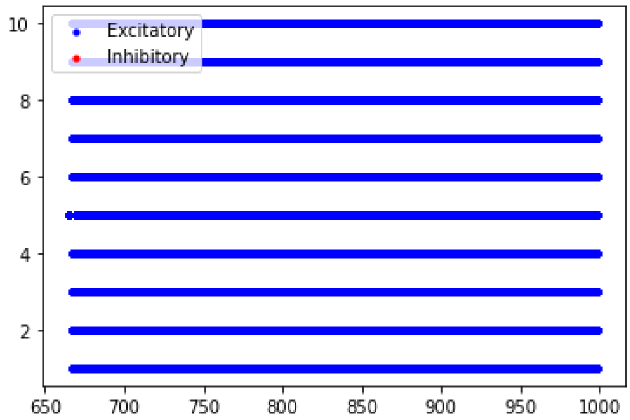


inhibitory population:

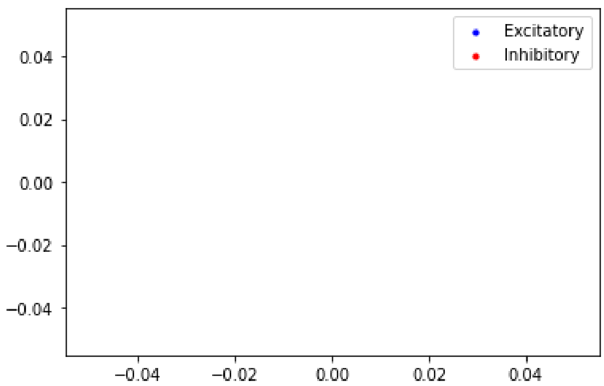


Raster plots:

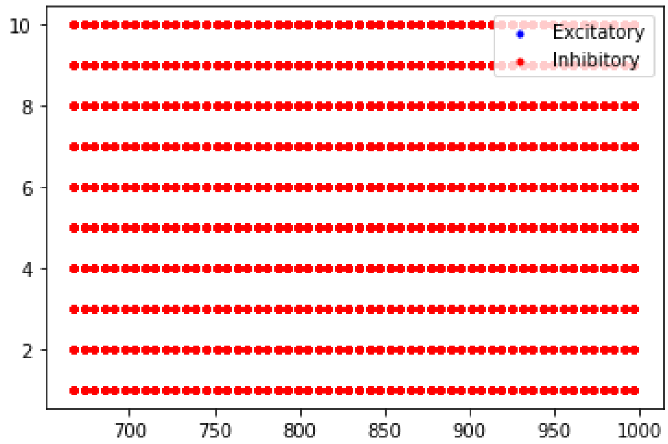
first excitatory population:



second excitatory population:



inhibitory population:



So it seems that the first population is the winner in this decision making.