Leaky Integrate and Fire Neuron.

Abstract

Spiking Neural Network (SNN) influenced by neurobiology provides efficient learning and recognition tasks. A power and area efficient electronic neuron is required to produce a large scale network similar to biology. In this project we are trying to understand and mimic the biological working of SNN by mathematical representation and python implementation

Introduction

Spiking neural network (SNN) is an attempt to comprehend and emulate human brain functions — a critical issue of next-generation computing. For recognition and classification applications, SNN outperforms the von-Neumann architecture in terms of energy efficiency. An efficient counterpart to the biological neuron is required to build SNN in hardware.

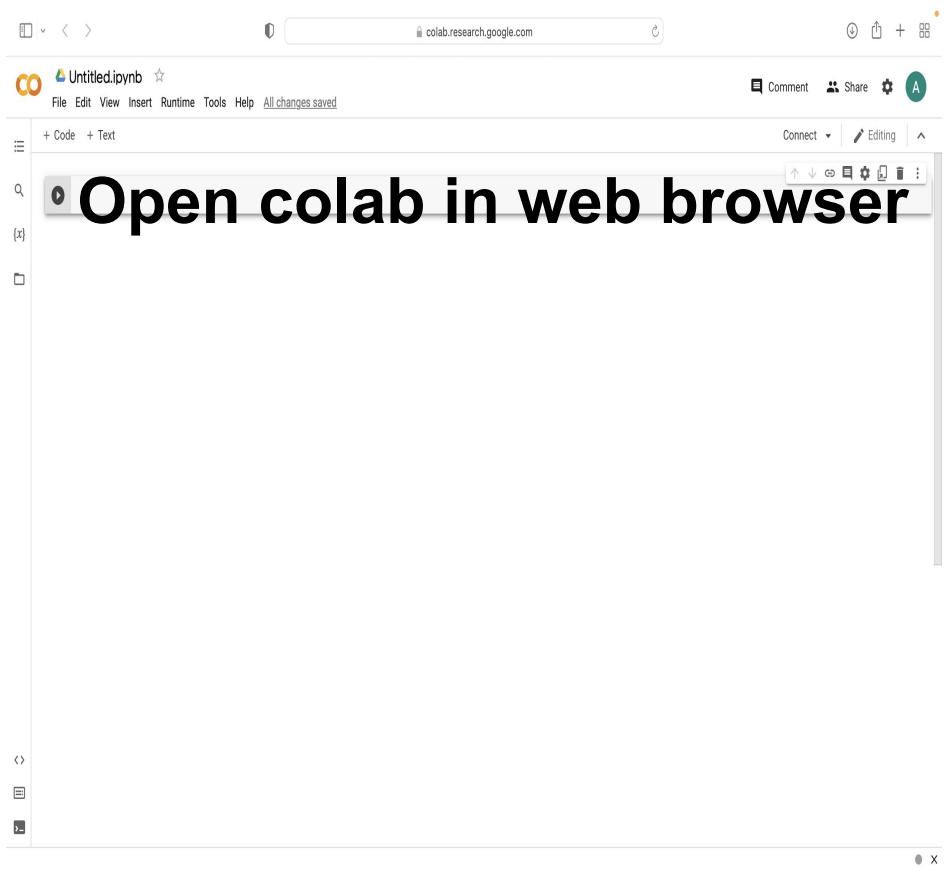
In addition, analog neuron circuits save space and power as compared to digital implementation. However, the human brain's enormous neuronal density (1011 neurons vs. 109 transistors/chip) and connection imposes fundamental limits

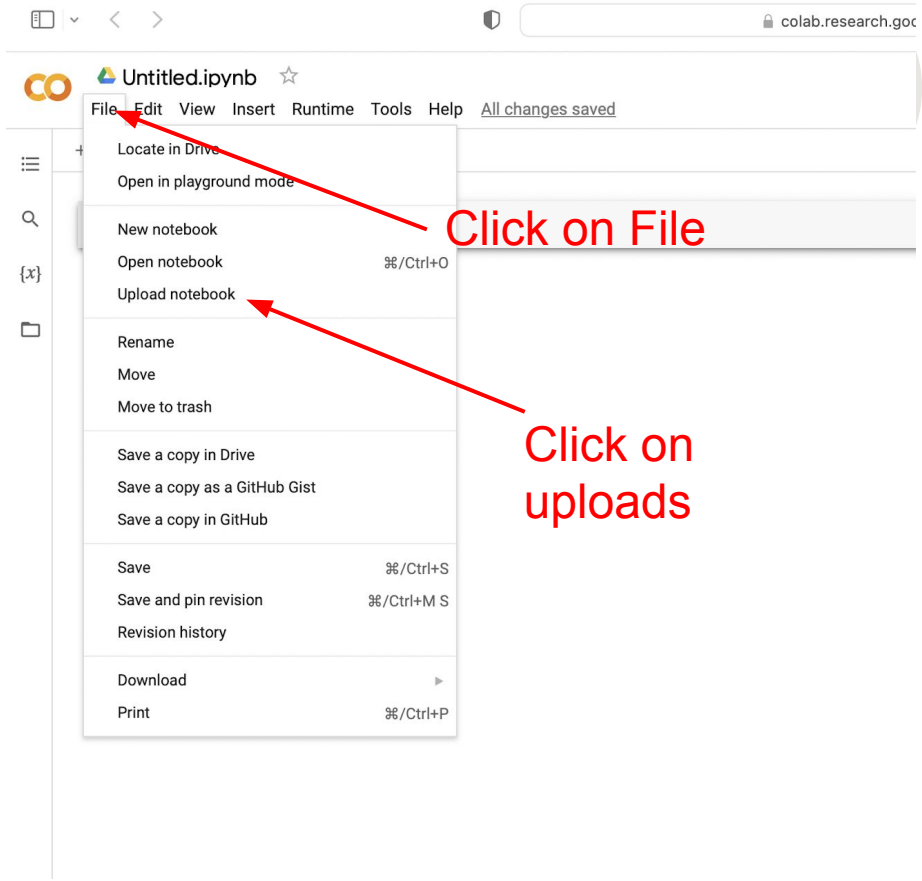
Steps to load and demonstrate the python code

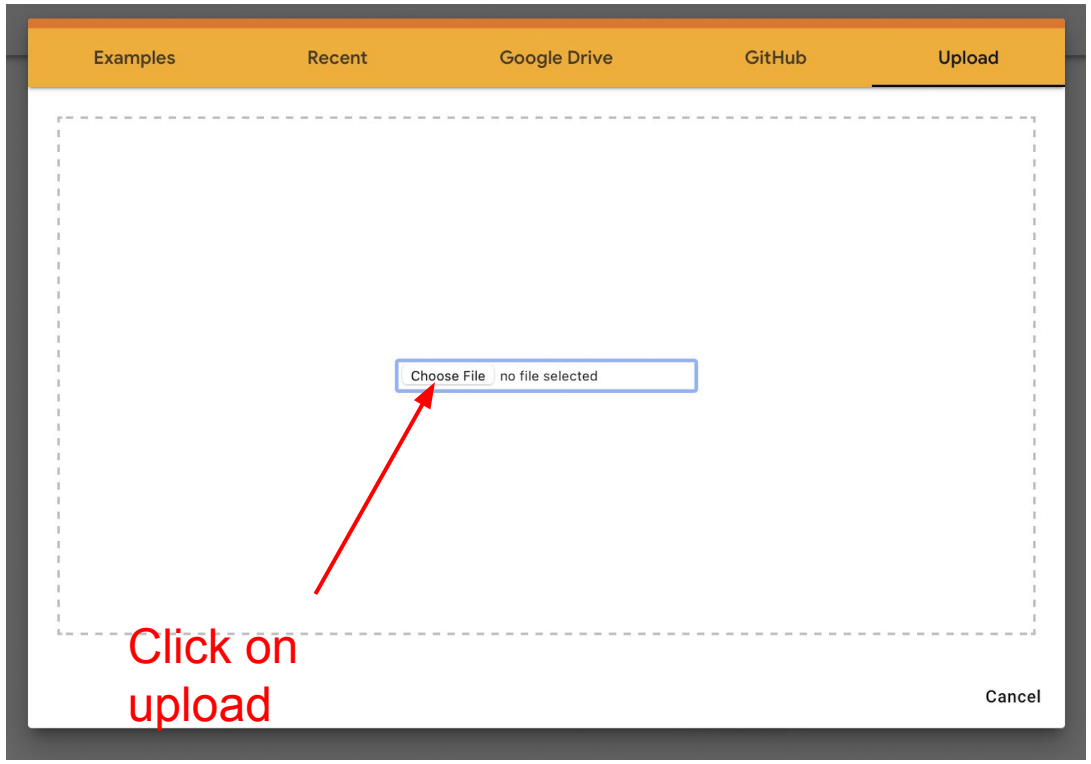
- Uploading the .ipynb file into the colab
- Run the whole code
- Set the threshold values
- Use the widgets to set the firing

Step 1

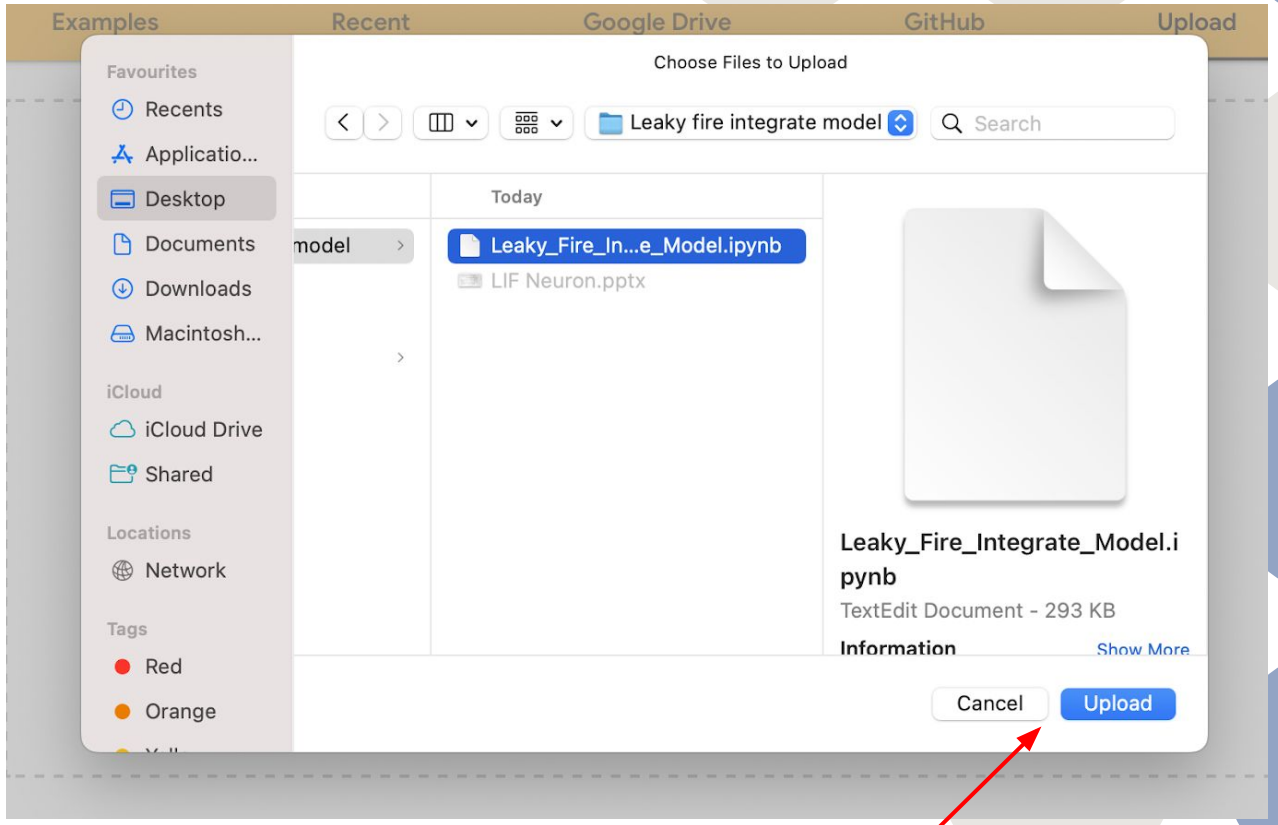
Uploading the .ipynb file into the colab



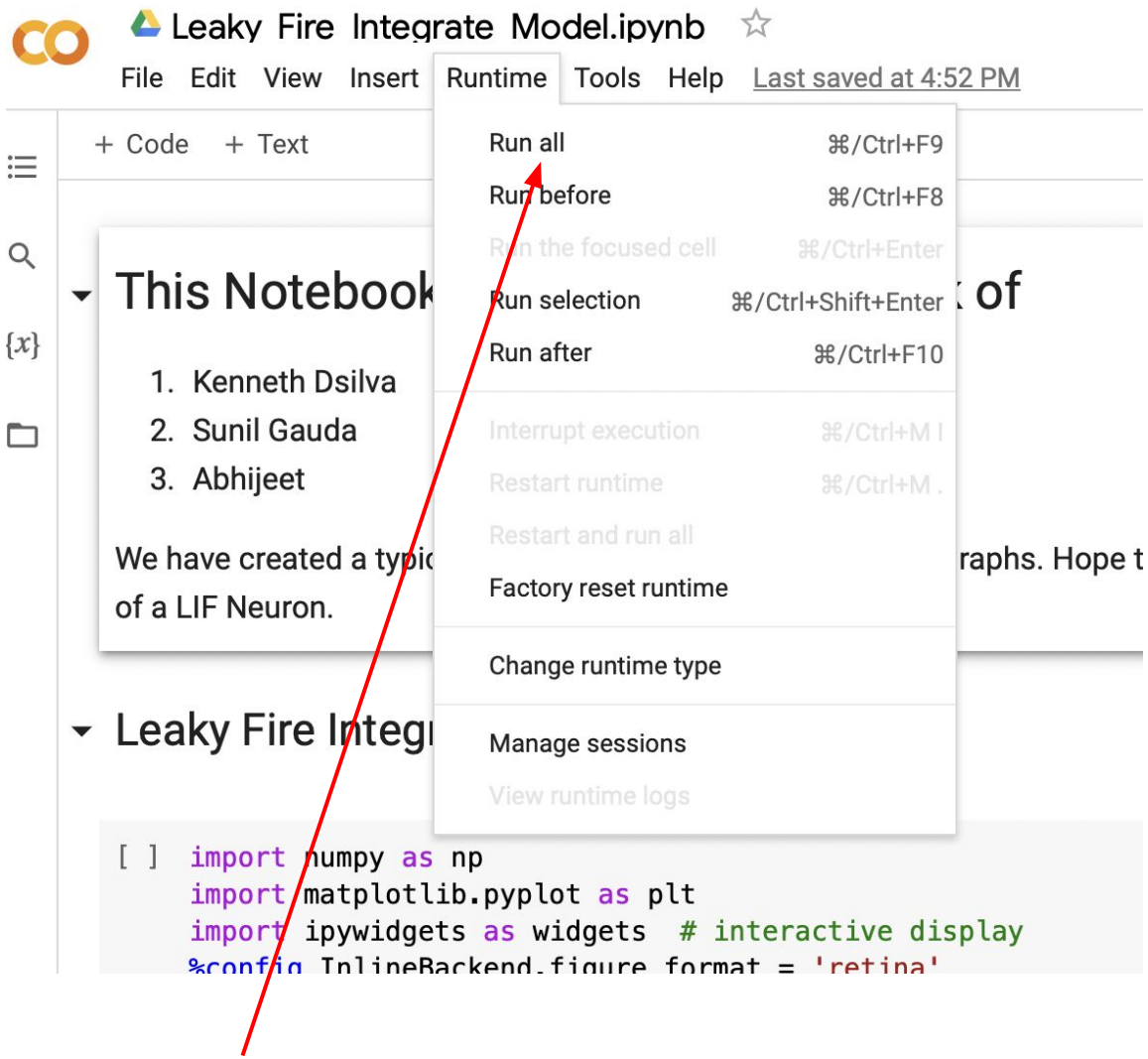




Click on upload



Upload the .ipynb file from the given zip



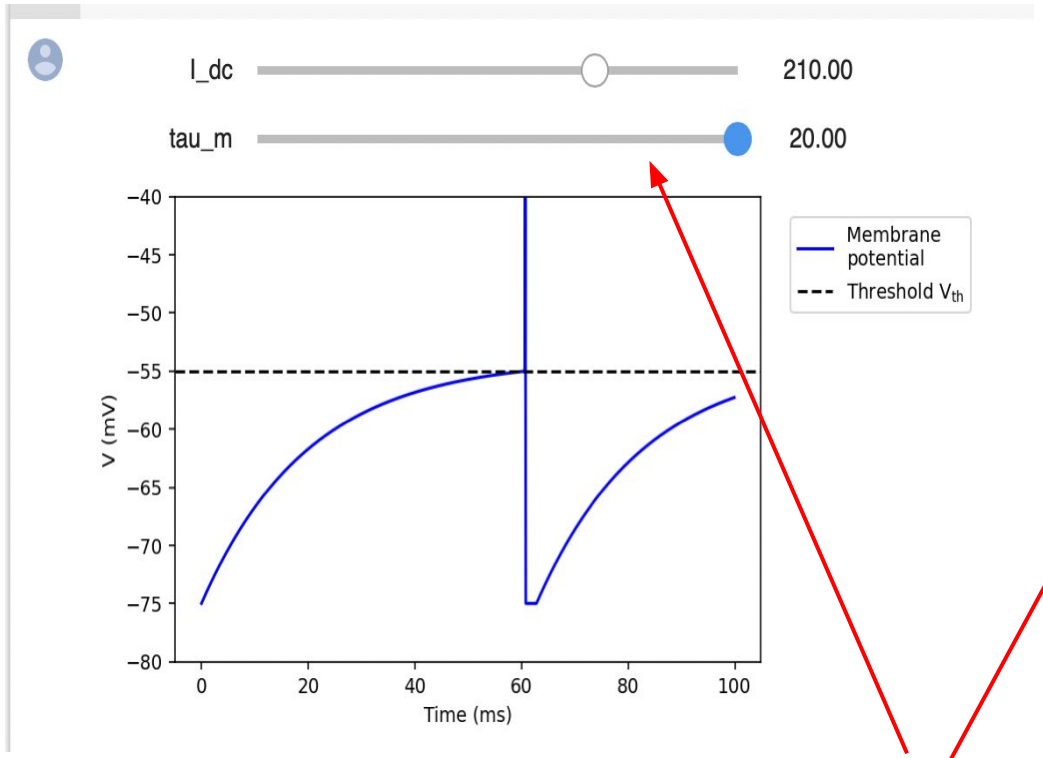
Go to Runtime Menu
click on Run all

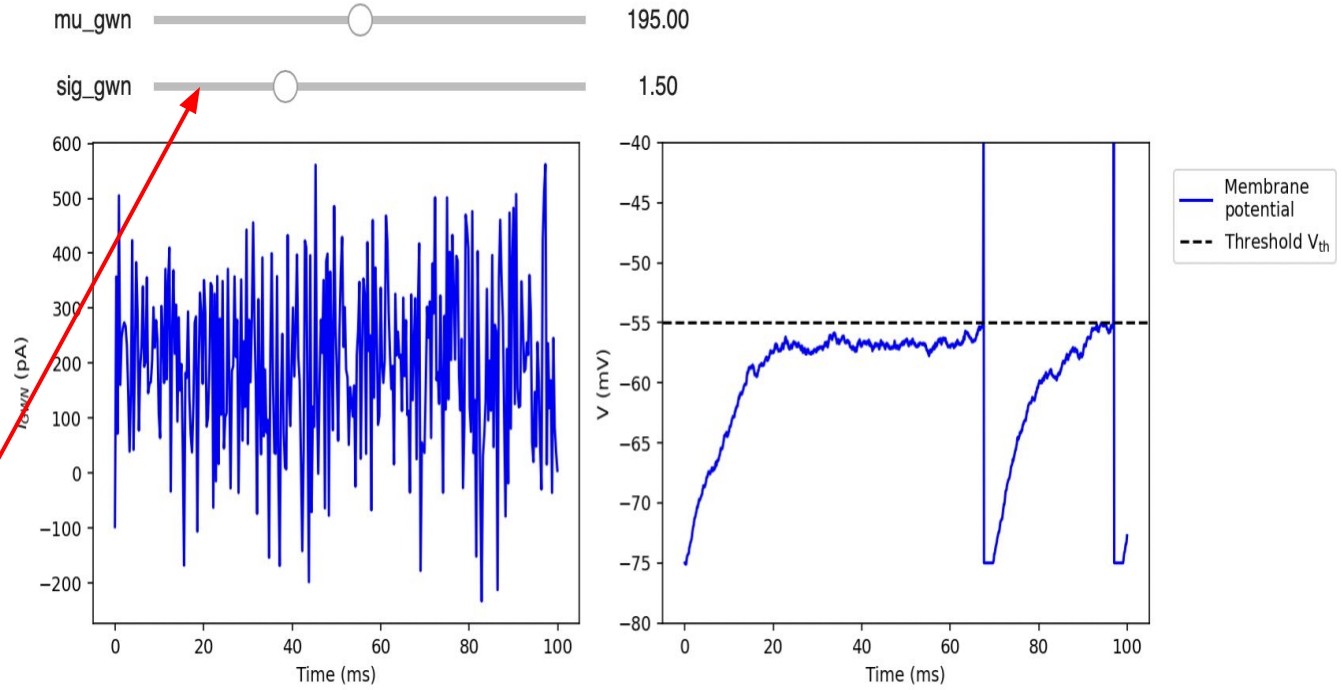
```python
def default_pars(**kwargs):
  pars = {}

# typical neuron parameters#
  pars['V_th'] = -55.     # spike threshold [mV]
  pars['V_reset'] = -75.  # reset potential [mV]
  pars['tau_m'] = 10.     # membrane time constant [ms]
  pars['g_L'] = 10.       # leak conductance [nS]
  pars['V_init'] = -75.   # initial potential [mV]
  pars['E_L'] = -75.      # leak reversal potential [mV]
  pars['tref'] = 2.       # refractory time (ms)

# simulation parameters #
  pars['T'] = 500.  # Total duration of simulation [ms]
  pars['dt'] = .1   # Simulation time step [ms]

  for k in kwargs:
    pars[k] = kwargs[k]

  pars['range_t'] = np.arange(0, pars['T'], pars['dt'])

  return pars


pars = default_pars()
print(pars)
```

Change the threshold and the parameters
for different results

Firing rate of neuron with membrane potential and threshold

Firing rate with Gaussian White Noise





Use the widgets to change the value of the variable and the change in firing

- By doing the above mentioned steps you will be able to understand how the Leaky Integrate and Fire Neuron model works.
- By changing the parameter of the neuron and adjusting the widgets which are provided in the code and explained in the above steps you will be able to understand the firing of neuron with and without Gaussian White Noise
- By the Application of Gaussian White Noise you can understand the difference between the behaviour of an Ideal Neuron and the real life Neuron