**Introduction**

In this project, we delve into the practical application of numerical methods, specifically focusing on Euler's method, to simulate neural models and synaptic transmission. Our primary objective is to utilize computational models inspired by biological systems to simulate the behavior of Leaky Integrate-and-Fire (LIF) neurons and alpha synapses. Through these simulations, we aim to gain a deeper understanding of neural processing dynamics and synaptic behavior.

The project underscores the potential of numerical methods in advancing artificial intelligence and neuromorphic computing. By employing numerical approaches to model biological phenomena, we aim to design systems that mirror the adaptive and efficient learning processes observed in natural neural networks. This work enhances our grasp of neural dynamics and lays the groundwork for future research at the intersection of neuroscience and computer science.

**Methods**

The project focuses on simulating Leaky Integrate-and-Fire (LIF) neurons and alpha synapses using Euler's method for numerical integration. Below are the essential components and steps involved in the simulation process.

**Implementation Details**

Tools: The project utilizes Python 3.10, with support from libraries such as NumPy and Matplotlib for numerical computations and data visualization.

Code Structure: The code is modularly organized, with functions dedicated to each simulation component, enhancing ease of use and scalability.

**Execution**

The program accepts command-line arguments to specify the simulation mode ("spike" or "current") and simulation time in milliseconds.

In "current" mode, the program requires an additional input of current in nanoamps.

In "spike" mode, the program requires the input spike rate in Hz.

**Leaky Integrate-and-Fire (LIF) Neuron Model**

The lifFunction function computes the new membrane voltage using Euler’s method.

The lif\_neuron\_sim function simulates a LIF neuron exposed to a constant input current over a specified time period using the lifFunction.

The voltage is updated at each time step, considering whether the neuron is in a refractory period or has reached the spike threshold. If the voltage hits the threshold, a spike is registered, and the voltage is reset to the neuron's resting potential.

**Alpha Synapse Model**

The alpha\_synapse\_sim function simulates a LIF neuron receiving input spikes at a provided input spike rate.

Based on the given spike rate and simulation time, input spikes are generated using the generate\_spikes function, creating an array of input spike times throughout the simulation period.

Synaptic current is calculated based on the weight, conductance, and potential difference across the synapse, as well as the time since the last input spike relative to the decay time.

Using the computed synaptic current, the membrane voltage is updated with the lifFunction at each time step, factoring in whether the neuron is in a refractory period or has reached the spike threshold. If the voltage reaches the threshold, a spike is registered, and the voltage is reset to the neuron's resting potential.

**Visualization**

Plotting Results: Upon completing the simulations, the membrane potential over time is plotted for both LIF neuron and alpha synapse simulations. This visualization provides valuable insights into the behavior of the neuron and synaptic transmission processes.

**Results**

“In this section, you should use your code to perform the following experiments and answer questions about them.”

|  |
| --- |
| For experiments 1, 2, and 3, the following parameter values should be used in your configuration. |
| For Euler’s Method: |

Experiment 1 Run the following command:

python neuro\_sim.py current 250 --current 0.003

Q1: Show the plot of the membrane voltage, , for the duration of the experiment (250 s).

Q2: How many spikes were produced over the course of the experiment?

Experiment 2 Run the following command:

python neuro\_sim.py current 6 --current 0.003

Q1: Show the plot of the membrane voltage, , for the duration of the experiment (6 s).

Experiment 3 Run the following command:

python neuro\_sim.py spike 100 --spike\_rate 50

Q1: Show the plot of the membrane voltage, , for the duration of the experiment (100 s).

Q2: How many spikes were produced over the course of the experiment?

Experiment 4 Run the following command, but change the Euler’s method time step to :

python neuro\_sim.py current 250 --current 0.003

Q1: Show the plot of the membrane voltage, , for the duration of the experiment (100 s).

Q2: How many spikes were produced over the course of the experiment?

Q3: How does this compare to the result from Experiment 1?

Experiment 5: Repeat experiment 3, but replace the exponential term in Equation 3, , with a 10th order Taylor Series approximation of , centered at 0.

Q1: Show the plot of the membrane voltage, , for the duration of the experiment (100 s).

Q2: How many spikes were produced over the course of the experiment?

Q3: How does this compare to your result with Experiment 3?

Conclusion

In this project, we simulated Leaky Integrate-and-Fire (LIF) neurons and alpha synapses using Euler's method for numerical integration. By implementing command line arguments to accept various inputs such as simulation mode, time, spike rate, and current, we demonstrated how different excitation methods influence neuronal behavior. Through the simulation, we explored how neurons respond to different input patterns and observed changes in membrane potential and spike patterns over time. The results showed how Euler's method can effectively simulate neuronal dynamics, highlighting its suitability for modeling simple neural networks. This project offered valuable insights into the computational modeling of neural processes and set a foundation for more complex simulations and explorations in the field of computational neuroscience.