Tools & Simulation Structure (Before Coding)

Thursday, 4 December 2025 3:29 PM

1. Why Numpy is needed (fast math)

Python alone is slow for:

- arrays
- loops
- math on thousands of time steps
- large vectors

Numpy lets you do:

✓ fast vector operations
✓ handle voltage arrays
✓ handle spike arrays
✓ simulate in milliseconds
✓ run thousands of dt steps instantly

Without Numpy, our simulation would be painfully slow.

2. Why Matplotlib is used (graph plotting)

Neuroscience = LOTS of graphs

We need:

- Voltage vs time graphs
- Spike raster plots
- Emotional patterns
- Firing rates
- Feature visualizations

Matplotlib makes these super easy, clean and scientific looking

3. Simulation Loop Logic (concept)

Every simulation has this loop:

```
for t in range(total_steps):
    I = input_function(t)
    V = update_voltage(V, I)
    if V crosses threshold:
        record spike
        reset voltage
    store results
```

This loop runs every dt

This is the entire heart of our simulation

4. Data structure to store Voltage & Spikes

Voltage (V)

```
voltage = np.zeros(total_steps)
```

Each index = voltage at a time step

Spikes

```
spikes = np.zeros(total_steps)
```

when neuron fires:

```
spikes[t] = 1
```

This becomes spike train dataset

5. Why Jupyter Notebooks are Useful

✓ We can see graphs instantly
✓ We can run cell-by-cell
✓ We can keep notes + diagrams next to code
✓ We can debug more easily

6. Understanding Basic Project folder layout

Now the simulation files will be organized:

```
neuron_models/
    lif_single_neuron.ipynb
    lif_single_neuron.py
    multi_neuron_circuit.ipynb
emotional_simulations/
    spike_trains/
    spike_plots/
```

* Summary

- Numpy → fast math & arrays
- Matplotlib → graphs
- simulation loop = core of neuron modeling
- Voltage stored in arrays
- Spikes stored in binary arrays
- Jupyter = perfect for scientific simulations
- folder layout keeps project clean.