Imperial College London

COURSEWORK

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Computational Neurodynamics

Author:
Emanuele Elie Debach, Cesare Bergossi (CID: 02547611, 06011559)

Date: November 17, 2024

README

The goal of the project is to generate small-world modular networks of Izhikevich neurons. The network consists of excitatory and inhibitory neurons which are divided into modules. These have both intra-module and inter-module connections. The algorithm aims to simulate the connectivity structure of the network, trying to understand how firing patterns and mean firing rates are affected by the rewiring probability p, which controls the randomness of the connections between modules. We analyse these results through the help of connectivity matrices, raster plots and mean firing rate plots.

Note: The project requires:

- `Python 3` to run the code.
- `numpy` and `matplotlib` libraries for matrix operations and visualisation purposes.
- `iznetwork` custom package containing the `IzNetwork` class, developed in Exercise 2.

Code Structure

`ModularNetwork` Class

The main class (`ModularNetwork`) includes modelling of the modular network, simulation execution and visualisation. It contains the following methods:

- `__init__`: Constructor initialising the network structure and parameters, with adjustable rewiring probability p.
- `set_neuron_parameters`: Sets Izhikevich model parameters (`a`, `b`, `c`, `d`) for excitatory and inhibitory neurons. These parameters were set in advance to model regular spiking behaviour for excitatory neurons and fast spiking behaviour for inhibitory neurons.
- `generate_modular_connectivity`: Creates intra- and inter-module connections between neurons, specifically:
  - Random excitatory-to-excitatory connections within modules.
  - Rewiring of inter-module connections based on `p`.
  - Excitatory-to-inhibitory focal connections within modules.
  - Diffuse inhibitory-to-excitatory and inhibitory-to-inhibitory connections.
- `run_simulation`: Simulates the network for `T` time steps, recording firing rates for each module and membrane potentials for all neurons.
- `connectivity_matrix`: Visualises the excitatory connectivity matrix as a scatter plot and saves it as a PDF.
- `plot_firing_and_mean_firing_rate`: Plots raster plot of neuron firing activity and mean firing rates per module in a sliding window, then saves the plots as PDF files.

`main()` Function

The `main()` function simulates the network for different values of rewiring probability `p`. It:

- Initialises a `ModularNetwork` instance for each `p`.
- Generates the connectivity matrix.
- Plots firing activity and mean firing rates.

Output Plots

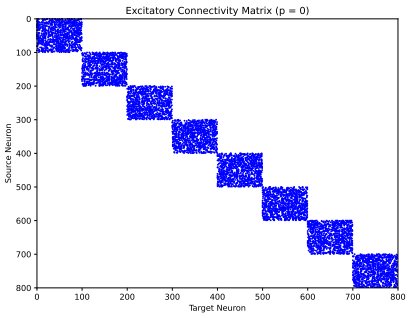

Figure 1: Connectivity Matrix ($p = 0$)

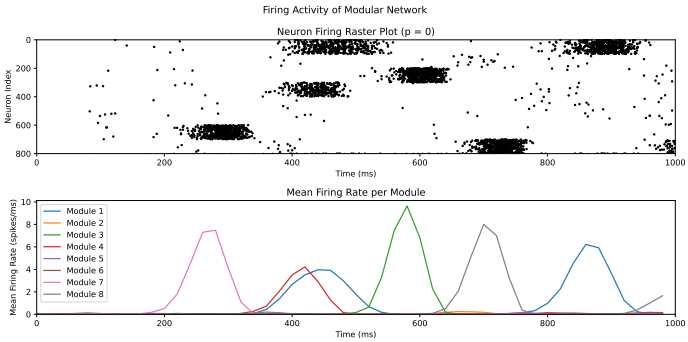

Figure 2: Raster Plot and Firing Rate ($p = 0$)

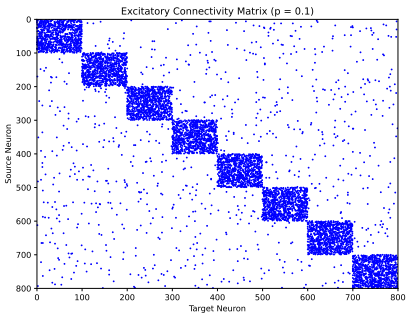

Figure 3: Connectivity Matrix ($p = 0.1$)

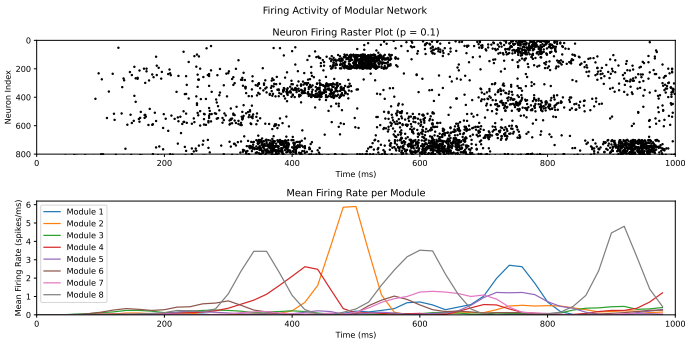

Figure 4: Raster Plot and Firing Rate ($p = 0.1$)

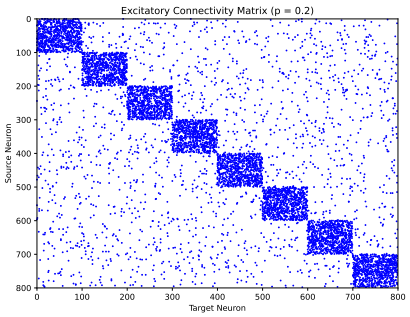

Figure 5: Connectivity Matrix ($p = 0.2$)

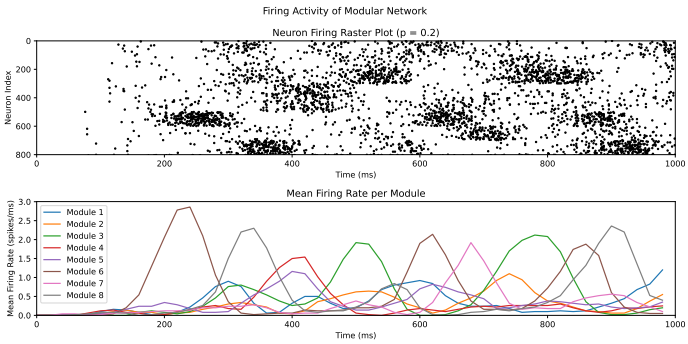

Figure 6: Raster Plot and Firing Rate ($p = 0.2$)

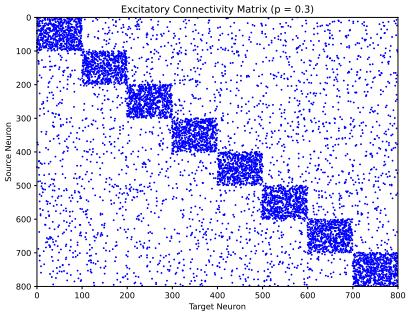

Figure 7: Connectivity Matrix ($p = 0.3$)

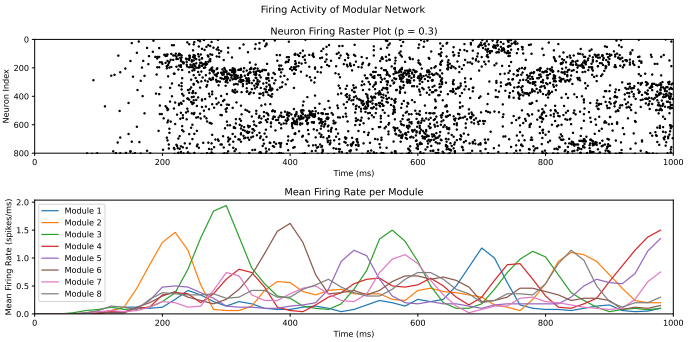

Figure 8: Raster Plot and Firing Rate ($p = 0.3$)

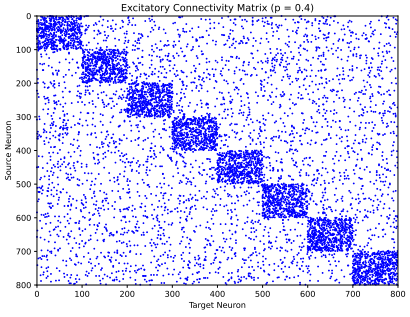

Figure 9: Connectivity Matrix ($p = 0.4$)

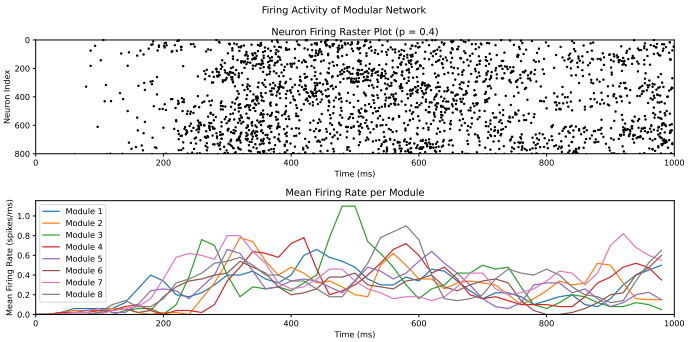

Figure 10: Raster Plot and Firing Rate ($p = 0.4$)

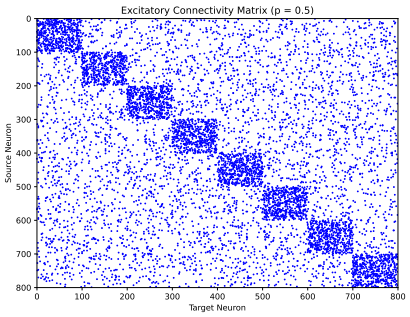

Figure 11: Connectivity Matrix ($p = 0.5$)

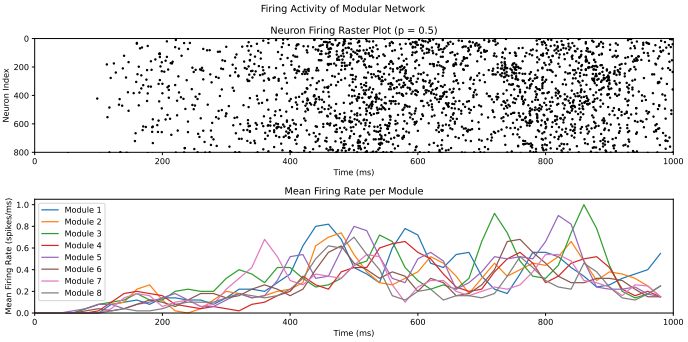

Figure 12: Raster Plot and Firing Rate ($p = 0.5$)