Computational Neurodynamics Coursework

Dynamical Complexity

Authors:

Asia Belfiore, Ginevra Cepparulo, Vincent Lefeuve

Imperial College London

November 17, 2024

Readme

About

This project simulates a Small World Modular Network of *Izhikevich Neurons*. It follows the Network creation and rewiring procedures covered in Lectures 8 (Modular Networks) and 9 (Dynamical Complexity). Information and specifications about the class parameters and functions can be found in the *coursework.py* file.

Getting Started

To get started, create an object from the *SmallWorldModularNet* class and set the required parameters, then simulate the network for a desired amount of time. By default, the class generates a Network of 1000 Izhikevich Neurons organized into 8 Communities, each with 100 *Excitatory Neurons* and 1000 random intra-edges, and a core of 200 *Inhibitory Neurons* connected to every Network Neuron.
Here is an example use of the class to create a default Network with no rewiring, and simulate it for 1000ms:

```
from iznetwork import IzNetwork
from coursework import SmallWorldModularNet

net = SmallWorldModularNet(p=0, n=1000, C=8, m=1000)
net.plot_weights("Matrix Connectivity")
net.simulate_net(T=1000) # simulate network activity for 1 second
net.plot_raster(T=1000) # generate raster plot
net.plot_rolling_mean_per_module(T=1000) # firing rate plot
```

Dependencies

The project requires the following packages and files in order to work:

python: The used programming language (any version $\geqq 3.9$).

iznetwork: The python file for the IzNetwork class that simulates the Izhikevich Neurons (provided alongside coursework specifications). It needs to be in the same directory as the *coursework.py* file for the network simulations to run.

numpy: This package is used throughout the project to perform all of the vector/matrix operations.

matplotlib: This package is used to generate the raster plots.

seaborn: This package is used to generate the heatmap of the network matrix connectivity.

Functions

The *SmallWorldModularNet* class is the main class for the project and is comprised of the following:

set_excitatory(), set_inhibitory(), set_delays(), rewire() and set_params() : private methods called when the object is initialised in order to set up the network with the desired parameters.

set_up_plotting() : private method used to set plotting parameters.

assert_connections() : public method that verifies that the network is set up correctly.

simulate_net() : public method that simulates network activity for some desired amount of milliseconds.

plot_weights(), plot_raster(), and plot_rolling_mean_per_module() : public methods to generate plots to show (respectively) the connectivity matrix, the spikings over time, and the firing rates of the network modules throughout the simulation.

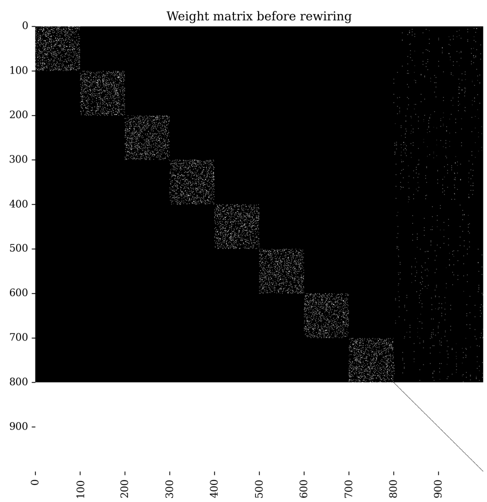
Please note that more precise descriptions are available in the python file documentation.

Question 1

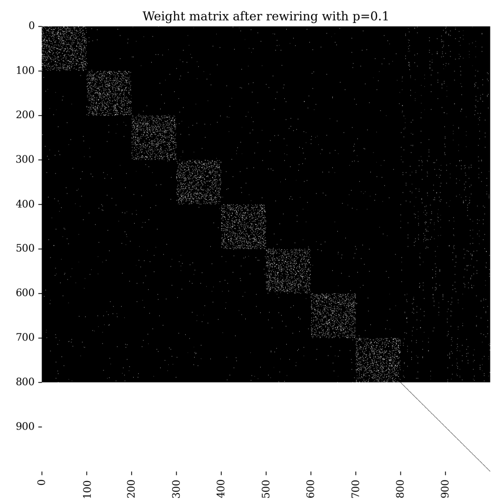
All plots below are grouped by question task (a,b,c) for all *Rewiring Probability* values p: p=0, p=0.1, p=0.2, p=0.3, p=0.4 and p=0.5.

a. Matrix Connectivity

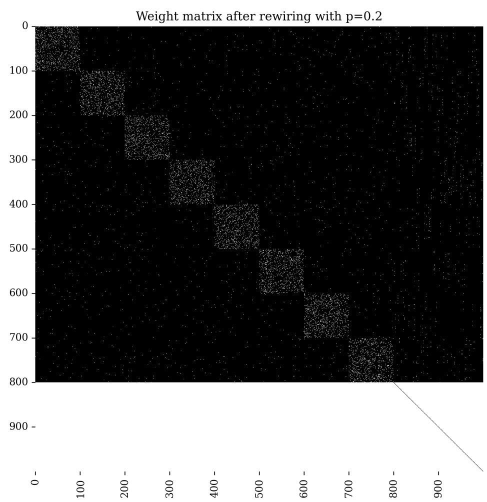
Plots of the Matrix Connectivity of the Network simulated with different *Rewiring Probabilities*. Each white dot in the plot represents a connection between the neuron pair (x,y) on the x-y axis. Every black spot represents the absence of an edge between the two neurons. Note: the white bottom strip represents the one-to-all connections between each inhibitory neuron and all other neurons.



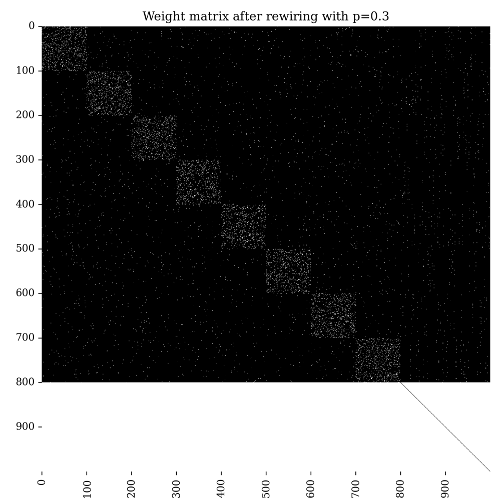
(a) Before Rewiring (p=0).



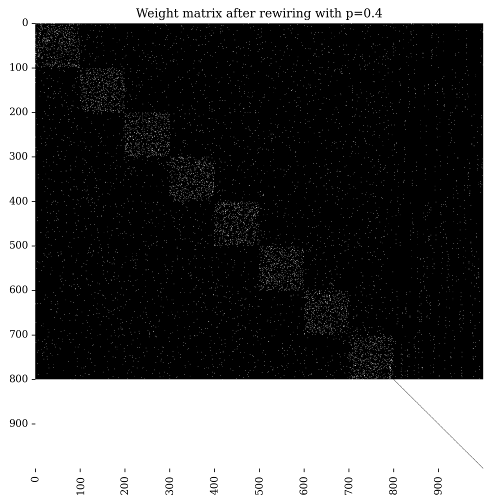
(b) After Rewiring with p=0.1.



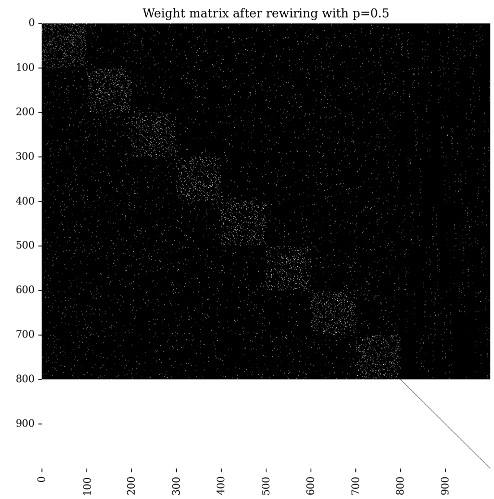
(c) After Rewiring with p=0.2.



(d) After Rewiring with p=0.3.



(e) After Rewiring with p=0.4.

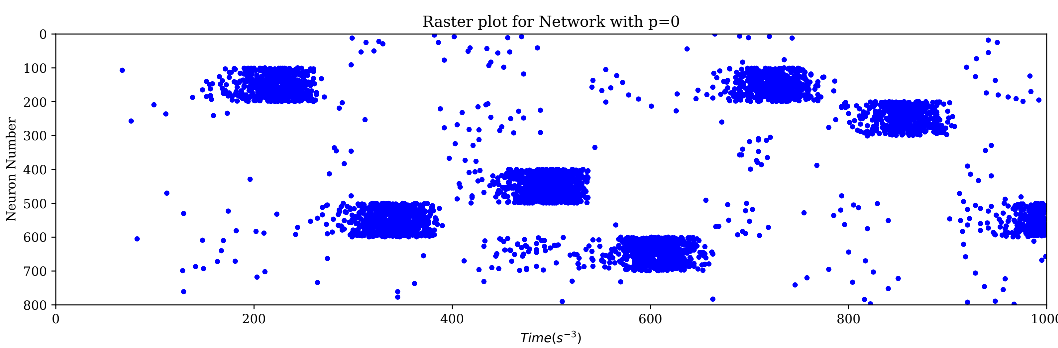


(f) After Rewiring with p=0.5.

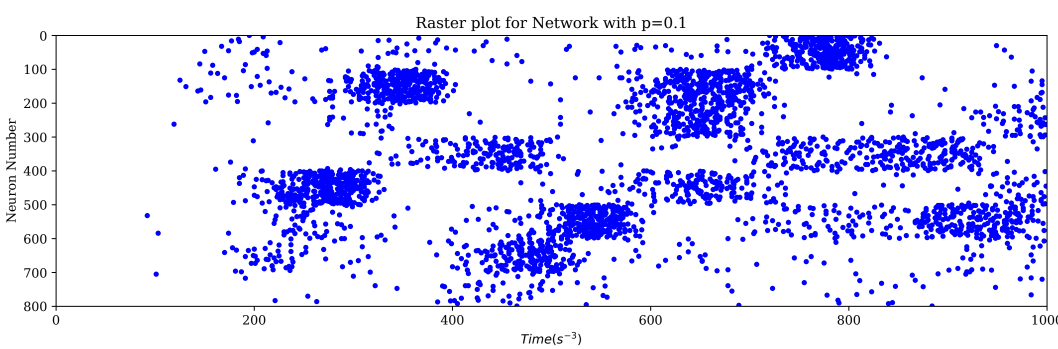
Figure 1: Plots of the Matrix Connectivity of the Network with different Rewiring Probabilities

b. Network Activity Raster Plots

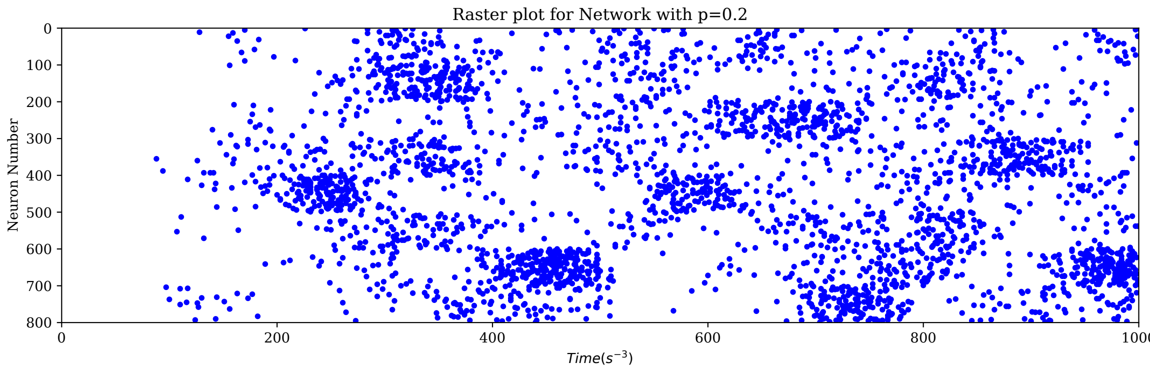
Raster Plots of the Network firing activity simulated for 1000ms with different *Rewiring Probabilities*. Each blue dot represents the firing activity of a neuron y at time x.



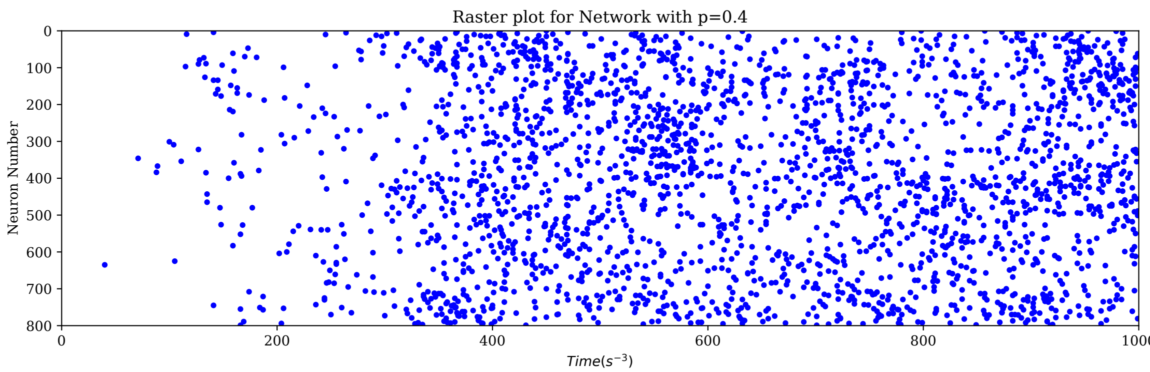
(a) $p = 0$



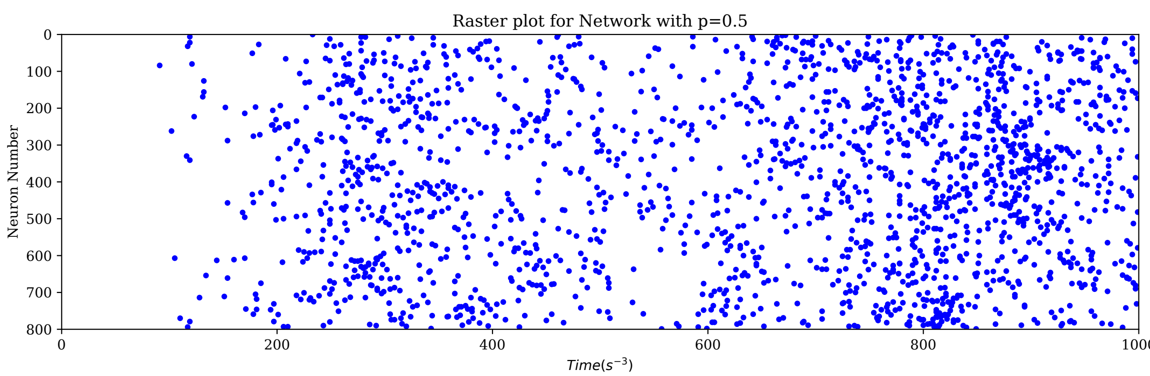
(b) $p = 0.1$



(c) $p = 0.2$



(d) $p = 0.4$

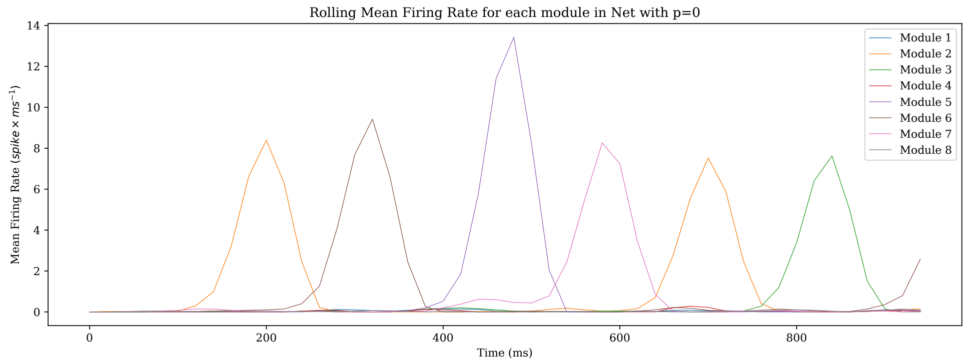


(e) $p = 0.5$

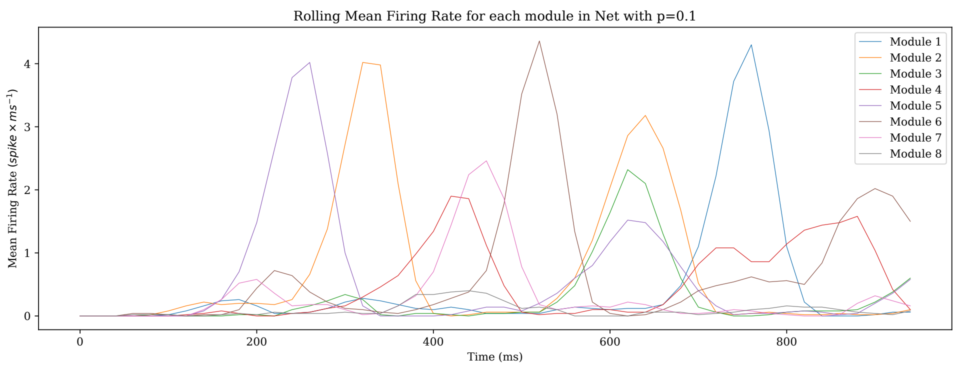
Figure 2: Raster plots of the network simulated for 1000ms with different rewiring probabilities.

c. Mean Firing Rate

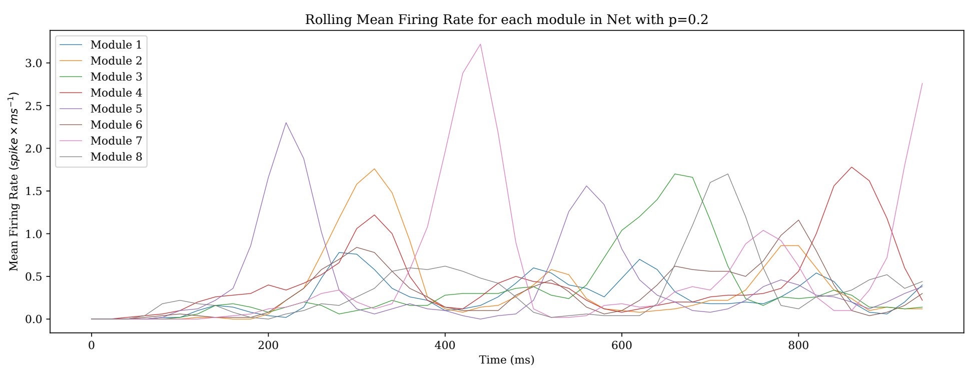
Plots of the mean firing rates of each Network module simulated for 1000ms with different *Rewiring Probabilities*. Each line represents the firing mean of the neurons of a module within the network at each time x.



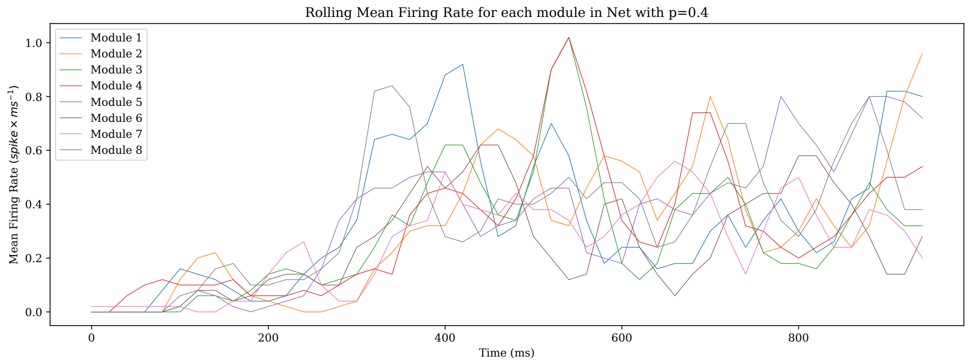
(a) Mean Firing Rate of the network with $p = 0$.



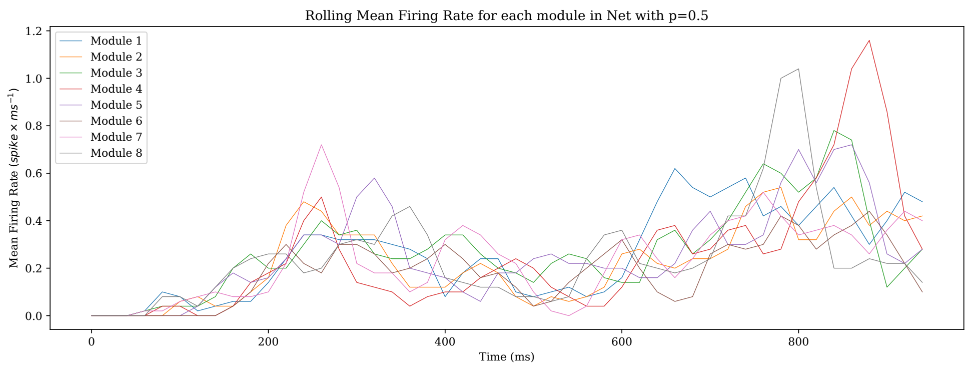
(b) Mean Firing Rate of the network with $p = 0.1$.



(c) Mean Firing Rate of the network with $p = 0.2$.



(d) Mean Firing Rate of the network with $p = 0.4$.



(e) Mean Firing Rate of the network with $p = 0.5$.

Figure 3: Mean Firing Rates of the network simulated for different rewiring probabilities.