Computational Neurodynamics

Exercise Sheet 4 (**Assessed**)
Dynamical complexity

Question 1

Write Python code that will generate small-world modular networks of Izhikevich neurons. The basic method is the rewiring procedure described in the Modular Networks Topic. But for details of the network construction, follow the description of the experiment in Dynamical Complexity Topic. Note that the networks need a separate inhibitory population. Use the neuron parameters from the earlier lecture notes for excitatory and inhibitory Izhikevich neurons. You can use the `IzNetwork` class from Exercise 2.

Generate a network for each of the following values of p: $p = 0$, $p = 0.1$, $p = 0.2$, $p = 0.3$, $p = 0.4$, and $p = 0.5$. For each of these networks:

a) Generate a plot of the matrix connectivity

b) Generate a raster plot of the neuron firing in a 1000ms run.

c) Generate a plot of the mean firing rate in each of the eight modules for a 1000ms run. Downsample the firing rates to obtain the mean by computing the average number of firings in 50ms windows shifted every 20ms. (This should yield 50 data points for each module.)

Your results should be consistent with those in the slides.

What to Submit

You should submit two files:

- A single Python file containing all the code you have used for the exercise. Suitable software engineering practice is strongly encouraged: encapsulate your code in different functions, add documentation where appropriate, and follow a consistent style. You may assume that `iznetwork.py` is available in Python's import path.
- A PDF file containing all plots in vector graphic format. You may choose to group your plots by sub-questions (a, then b, then c), or by values of p ($p = 0$, then $p = 0.1$, etc). The first page of the PDF should contain a README-style section specifying the structure and dependencies of the functions used.

Working in Groups

You are encouraged to work in groups of up to three students. But you may work in pairs or alone if you prefer.

Frequently Asked Questions

QUESTION: Isn’t the description of how you build a modular small-world network in the Modular Networks slides slightly different from the description in the relevant Dynamical Complexity slides?

ANSWER: Yes! According to the notes from the Modular Networks Topic, to build a modular small-world network with n nodes, m edges, and C communities, you first construct C disconnected communities with n/C nodes each, where each community has m/C random symmetric edges. According to the exercise specification, you should try to reproduce the results in the Dynamical Complexity Topic, where "each module has 1000 randomly assigned one-way excitatory-to-excitatory connections (before the rewiring phase)" and "connections are not symmetrical, so we have a directed network". For the exercise, you should follow the second description.

QUESTION: What scaling factors / weights / conduction delays should we use?

ANSWER: These are all specified on a slide in the Dynamical Complexity Topic.

QUESTION: What libraries can I use?

ANSWER: You can use numpy for any computation, the provided IzNetwork class to simulate the network of Izhikevich neurons and any plotting library of your choice. You may not use any other library.