Day 3: Exercises for topology and dynamics

It is recommended to work through Part 4 of *Introduction to PyNEST* before starting.

Exercise 1: The effect of broad degree distribution on network dynamics

a) In-degrees and out-degrees

Implement a densely connected (connection probability $\epsilon = 0.2$) recurrent inhibitory network of N current based integrate and fire neurons using the template `exercise1.py`. We will study the effect of broad degree distributions by alternating between a binomial with parameters $B(N, \epsilon)$ and a delta function distribution of the form $\delta(x - \epsilon N)$ for in-degree and out-degree independently. Write your code in such a way that you can easily switch between these three different connectivity schemes:

1. Out-degree distribution is binomial and in-degree distribution is a delta function.
2. Out-degree distribution is a delta function and in-degree distribution is binomial.
3. In and out degree distributions are both binomial.

Hint: How to alternate between broad in-degree and broad out-degree When wiring up your network, use normal Connect routines but use a rule in the connection dict to fix either the in-degree or out-degree.

```
conn_dict = {"rule": "fixed_indegree", "indegree": N}
nest.Connect(A, B, conn_dict, syn_dict)
```

Fixing the in-degree or out-degree has the effect that the non-fixed degree will be binomially distributed. In order to implement in- and out-degree distributions that are both binomial, use the connection rule `pairwise_bernoulli`. More information can be found at http://www.nest-simulator.org/connection_management/

b) Analysis of network dynamics

Follow the instructions in `exercise1.py` to implement the following data analysis routines:

- Plot all membrane potential traces and the mean membrane potential trace
- Plot a histogram of the neuron firing rates
- **Bonus:** Plot a raster of the spiking activity ordered by neuron rate

c) Effect of connectivity on dynamics

Observe the effect of broadening the in-degree and out-degree distributions alternatively. How does the broadening of degree distributions affect the dynamics of the network? How does it affect the distribution of firing rates?

Note: if you want to learn more detailed analysis about the effect of broad degree distribution on neuronal network dynamics read Roxin (2011).

Exercise 2: Connection profiles

Complete the template found in `exercises2.py` to define the following connection profiles. A handy plotting routine is provided for you to check your results on a 1mm by 1mm layer with 40x40 neurons `iaf_neurons`.

Connection dictionary	Description
CD1	Each neurons is observed have a 60-80% chance of forming connections with targets, up to a maximal distance of 250μm from the soma. Intracellular recordings revealed that all postsynaptic neurons had a fixed EPSP size of 2mV, although delays increased linearly with distance from 2ms for neurons adjacent to the soma, up to 5ms for neurons located at the most distant edge.
CD2	The target population of neurons forms a cigar-shaped band, with approximate values of 0.25mm and 0.5mm for the width and height respectively, in which the presynaptic neuron was centrally located. Neurons do not connect to all possible targets; instead, it connects to all neurons in the immediate vicinity, and has a 65% chance of forming a connection to neurons at the edge of this profile.
CD3	Neurons did not form connections onto neurons in the immediate vicinity; instead they synpase on neurons that are located between 0.25mm and 0.6mm distant, with a probabilty of 50%. Weights are found to be distance-dependent, from 0.75mV nearest to soma, to 0.4mV at the furthest edge.

Exercise 3: Local inhibition vs. local excitation

In this exercise we will study the effect of local connectivity on the distribution of pairwise correlation coefficients by completing the template `exercise3.py`. The template already contains almost all the necessary parameters.

a) Create the layers

Use the provided layer dictionary to create two populations of integrate and fire neurons, one inhibitory and one excitatory. Extract the ids of the nodes in each population for later use.

b) Connect the layers

To investigate the effects of connectivity, your model must be able to alternate between random and local connectivity for each population. Write your code in a way that you can alternate easily between the following configurations:

1. Both populations are randomly wired.
2. The excitatory population is randomly wired but the inhibitory population connects locally.
3. The Inhibitory population is randomly wired but the excitatory population connects locally.
4. Both populations connect locally.

When implementing the random connectivity use the function `Connect()` with a fixed_outdegree rule . The local connectivity kernel has to have the following properties:

1. The number of connections has to be roughly the same as in the random case.
2. Use a circular mask to add a cutoff to kernel.
3. The kernel is Gaussian with the parameters given in the template.

4. Use divergent connections.

In order to check that the connectivity is the one that you were looking for, make use of the plotting functions of the topology module to visualize the network.

c) Analyse the activity

Use the functions that you implemented in exercise 1 to visualize the activity of the network. It is sufficient for the purpose of this exercise to record from the excitatory population only. In addition:

- add a population stimulus time histogram (PSTH) to observe the summed activity of the whole population
- Use the function `getccs()` to get the list of pairwise correlation coefficients and plot its histogram.

d) Effect of local and random connectivity

Investigate the effect of the different connection configurations on the network activity. What do you observe? Does the mean of the distribution change with the different configurations? What happens to the spread of the distribution?

If you want to learn more detailed analysis about the effect of broad degree distribution on neuronal network dynamics read Pernice et al. (2011).