Exercises: topology and dynamics

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

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 select between 'random' divergent or convergent connections.

```
conn_dict = {"rule": "fixed_indegree", "indegree": C}
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

The data analysis routines are in `helper_functions.py`. They are mostly already implemented, you should complete the function `plot_Vm_traces()` to plot all membrane potential traces and the mean membrane potential trace in one figure.

c) Effect of connectivity on dynamics

Perform the data analysis for all three set-ups. How do they differ from each other? Note: if you want to learn more detailed analysis about the effect of broad degree distribution on neuronal network dynamics read Roxin (2011).

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. 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	Neurons did not form connections onto neurons in the immediate vicinity; instead they target neurons that are located between 0.25mm and 0.6mm distanct, 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.

At the end of the exercise, your plots should look like these ones:

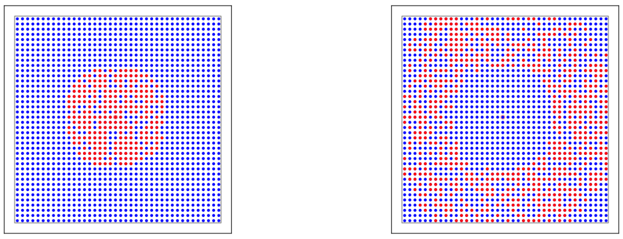
Figure 1: Connectivity profiles

Exercise 3: Effect of the connectivity profile on the network dynamics

The goal of this exercise is to study the effect of different connectivity kernels on the network dynamics. Implement a locally connected random network of inhibitory neurons using the template `exercise3.py`. All the parameters that you will need are contained in the template, as are the data analysis routines.

a) Create the layers

Create a 1mm by 1mm layer of inhibitory neurons similar the one created in `exercise2.py`. This time, the neuron model is slightly different, but all the parameters are provided in the template. Implement two different versions of the layer:

1. One in which the neurons are located in a regular grid.
2. Another in which the neurons are distributed randomly.

In order to avoid boundary effects, wrap the layer forming a torus (i.e., a doughnut shape).

b) Connect the layers

The next step is to create the connectivity dictionaries. You should start by creating two different connectivity profiles. Write your code in a way that you can easily switch from one configuration to the other. The two different connectivity profiles are as follows:

1. Circular mask with a radius of 0.15 mm, with a connection probability of 0.8.
2. Doughnut-shaped mask with a connection probability of 0.8. It is very important that you choose your parameters such that the number of connections is approximately the same as with the circular profile. You can achieve this by creating a doughnut-shaped mask that has the same area as the circular mask.

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

Run your code and observe the dynamics that emerge in the network in each case. How are they different? What is the effect of randomizing the position of the neurons?

d) Effect of connectivity profile

To explore the mechanism underlying the generation of these different dynamics, progressively modify the doughnut-shape connectivity profile and make it more circular, i.e., reduce the area without connections in the center. For which value of inner radius is the dynamics observed with the doughnut-shape profile indistinguishable from the one produced by the circular connectivity profile?