



## Day 4: Exercises for topology and network dynamics

Before starting the exercises, please update the svn directory by entering the following commands in the terminal:

---

```
you@your_machine:$ cd /path/to/your/svn/directory
you@your_machine:/path/to/your/svn/directory$ svn up
```

---

This will create a new Day4 directory with three skeleton exercise\*.py files that you will use for completing today's exercises.

### Exercise 1: Effect of broad degree distributions on the network dynamics

Implement a recurrent network with  $N = 1,000$  inhibitory neurons and connection probability  $\epsilon = 0.2$ . Use the template exercise1.py that you will find inside Day4 folder.

The goal of this exercise is to study the effect of broadening the degree distributions by alternating between a binomial distribution with parameters  $\mathcal{B}(N, \epsilon)$  and a  $\delta$ -shaped distribution  $\delta(x - \epsilon N)$  for in-degree and out-degree independently. Write your code in such a way that you can easily switch between the following 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.

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?

**Hint:** How to alternate between broad in-degree and broad out-degree. When wiring up your network, you can choose between these two NEST functions:

---

```
nest.RandomDivergentConnect()
nest.RandomConvergentConnect()
```

---

In the first case, out-degree distribution will be a delta function and the in-degree distribution will be a binomial. In the other case, the opposite will happen. In order to implement in and out-degree distributions that are both binomial you will have to write your own code since there is no specific NEST function for it. Use the function `numpy.random.binomial()` and loop through all the neurons to wire them up.



### Exercise 2: Getting started with `nest.topology`

We start with a warm-up exercise, the goal is to implement some interesting connectivity profiles in a 2D network.

1. We assume a layer of excitatory neurons, for which it is sufficient to use the `iaf_neuron` model from NEST. Our layer will be 1mm by 1mm, and have 40x40 neurons.
2. For each of the descriptions provided in the table below, create connection profiles and wire up the networks.
3. For each profile, plot the network and show the post-synaptic population for a single and centrally located neuron. Save each plot as `CDX_conn.png` for  $X = \{1..4\}$ . You may wish to use `matplotlib.savefig`.

Connection dictionary	Description
CD1	Each neurons is observed have a 60-80% chance of forming connections with targets, up to a maximal distance of $250\mu\text{m}$ 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 synapse on neurons that are located between 0.25mm and 0.6mm distance, with a probability of 50%. Weights are found to be distance-dependent, from 0.75mV nearest to soma, to 0.4mV at the furthest edge.

Your plots should look like these:

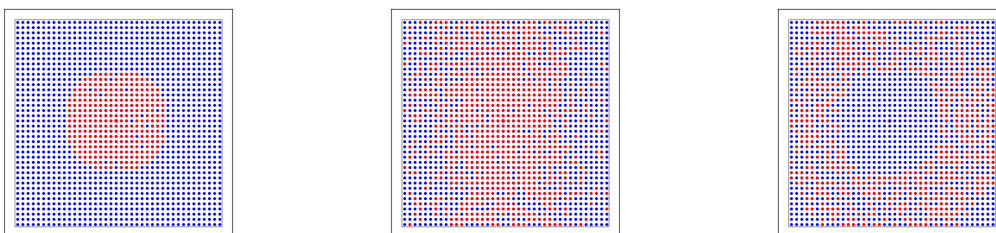


Figure 1: Connectivity profiles

**Hint:** Look into the handout to learn how to use the `nest.topology` module.



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

Implement a locally connected random network of inhibitory neurons. Make use of the template `exercise3.py` that you will find inside `Day4` folder. All the parameters that you will need are contained in the template. The goal of this exercise is to study the effect of different connectivity kernels on the network dynamics.

Create a 1mm by 1mm layer of inhibitory neurons similarly to what you did in `exercise2.py`. This time the neuron model is slightly different, but all the parameters 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 conditions wrap the layer into a torus (i.e., a lentic shape).

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 switch easily from one configuration to the other. The two different connectivity profiles are as follows:

1. Use a circular mask with a radius of 0.15 mm. The connection probability inside this circular region is 0.8. Make sure that neurons receive on average  $\sim 90$  incoming connections (in-degree).
2. Use a doughnut-shaped mask. The connection probability inside the doughnut is 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.

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?

To explore the mechanism underlying the generation of these different dynamics, modify progressively 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 the dynamics observed with the doughnut-shape profile is indistinguishable from the one produced by the circular connectivity profile?