Introduction to neural networks

Homework No. 2

Elad Tolochinsky, Yossi Solomon

Part 1

We used our back-propagated feed-forward network from HW1 as the basis for this exercise. The network was configured as:

- A network with a single hidden layer.
    - An input layer comprised of 900 neurons.
    - A hidden layer of X neurons.
    - An output layer of 900 neurons.

The number of neurons in the hidden layer (X) was changed by us to find the best possible option. Initially we thought the more neurons we had there the better the final result would be, but it turned out that wasn't true – we think that was because of overfitting. Using a grid-search we tried different options and ended up with 30 neurons at the hidden level.

As an error function we used an RMS function between output and input, then averaging that over all sub-pictures.

After finding the amount of hidden neurons we used the grid search again to determine the optimal learning rate and number EPOCHS. We did this instead of adding a stop condition based on the quality of the network as experimenting with different options was faster this way and easier to implement.

The training time was the fastest for the networks with less hidden neurons which was beneficial to the total run-time.

After running the best network we had on the other two pictures (landscape, person) and the text file we saw we can define a filter based on the error function by defining an error threshold, thus distinguishing between pictures and text in this example.

Part 2

We implemented a Kohonen network as described in the exercise.

For the learning rule, we used a learning rate which decreases in time (from iteration to iteration) and in space (as we get further away from the winning node)

The learning rule we used is:

$$W_t^i = W_{t-1}^i + e^{-d(i,j)/\sigma(t)} * 0.1 * e^{-t/t_{max}}$$

Where

- W_t^i is the weight of the node I at iteration t
- J is the winning node

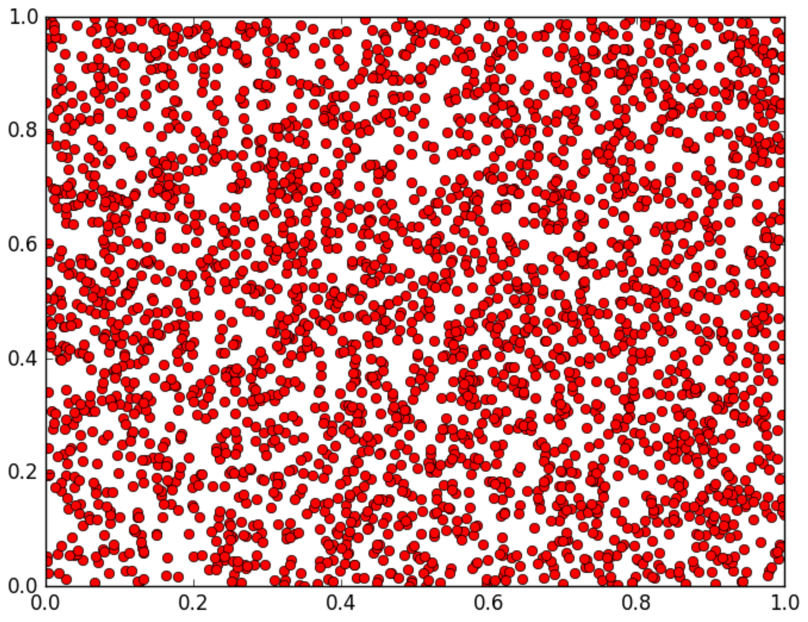
- d(i,j) is the distance between I and j
- $\sigma(t) = 4e^{-tlog(4)/t_{max}}$
- t is the current iteration
- t_{max} is the maximum iteration number

The following figures shows the networks during the training for the various topologies and data.

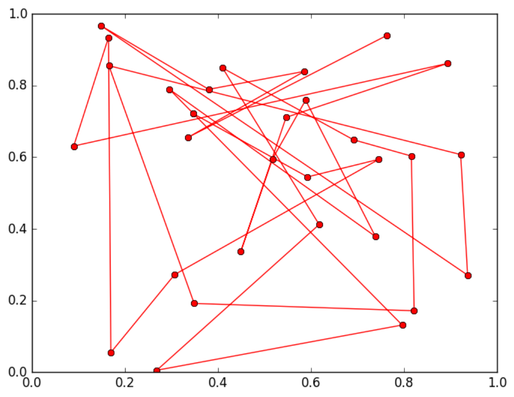
For all the experiments we generated 3000 points and used 30,000 iterations. Snapshots were taken every 10,000 iterations

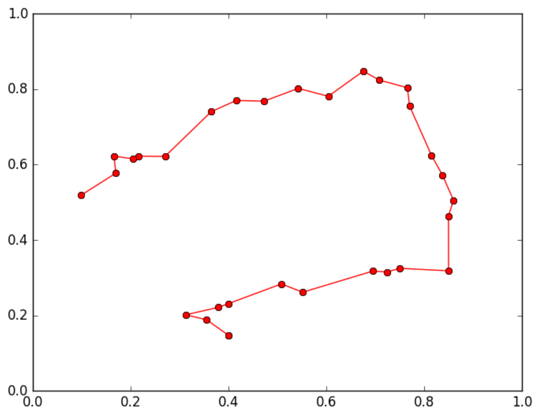
<u>Points in the Square</u>

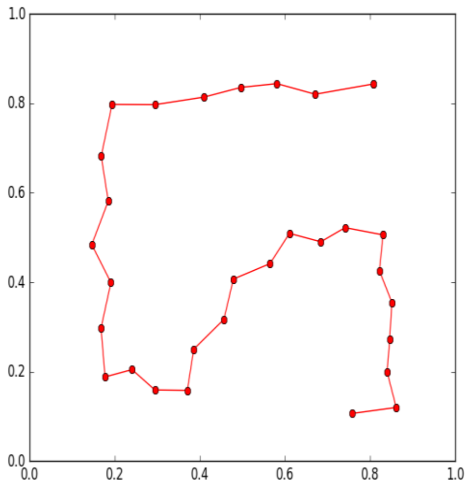
<u>Uniform sampling data</u>

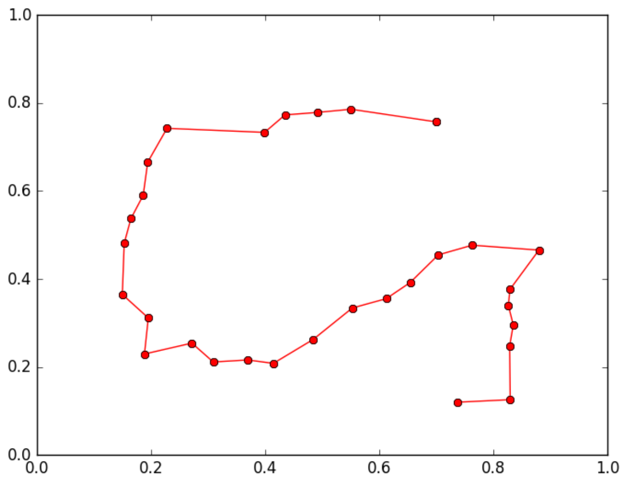


<u>Line Topology</u>

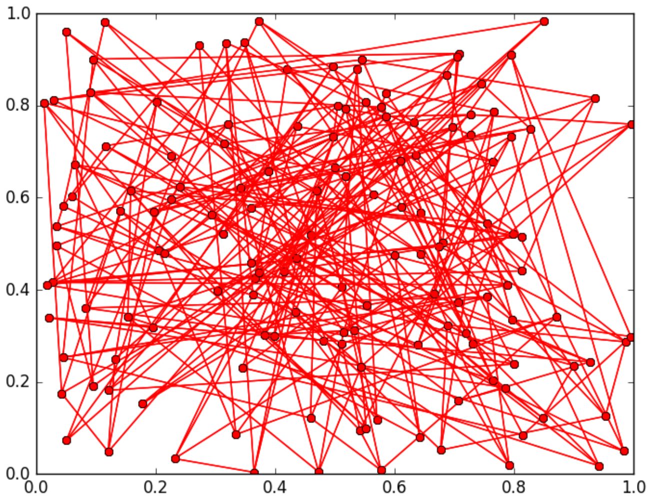


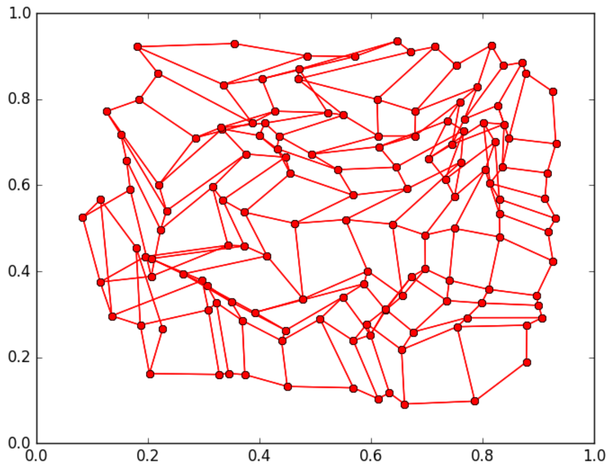


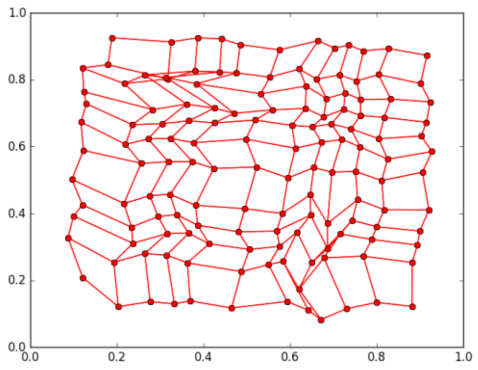


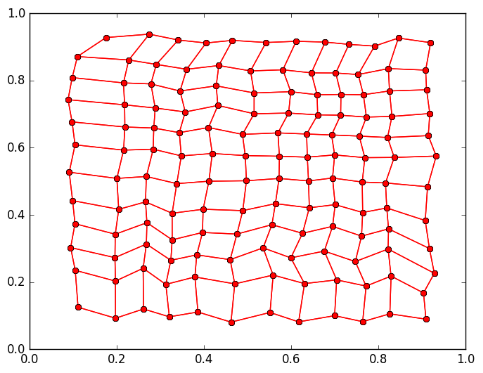


Grid Topology



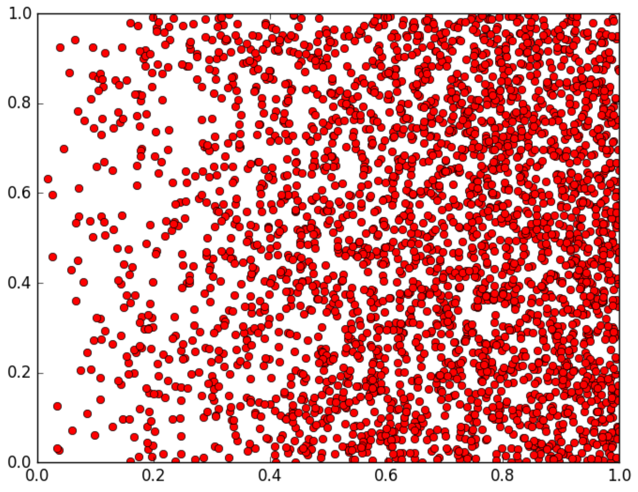




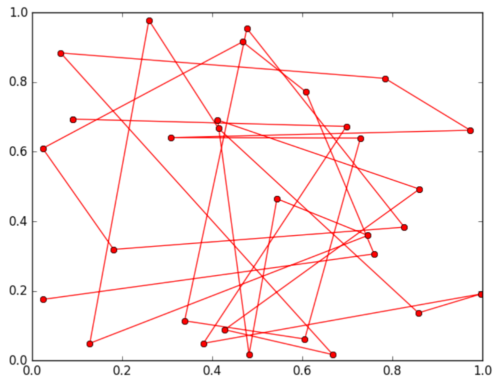
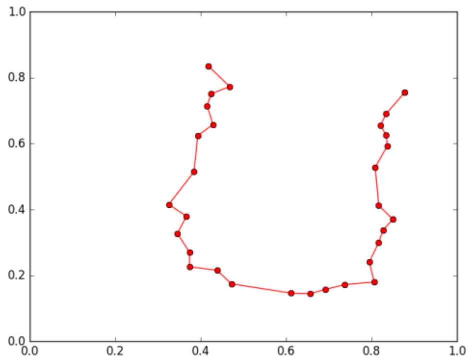
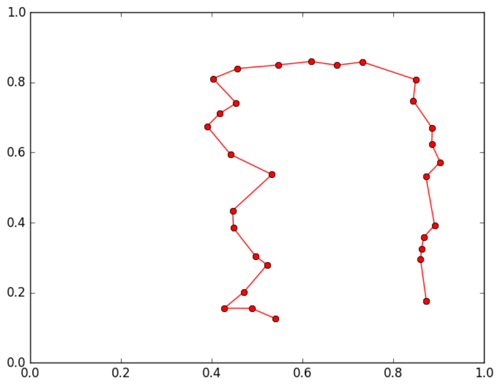
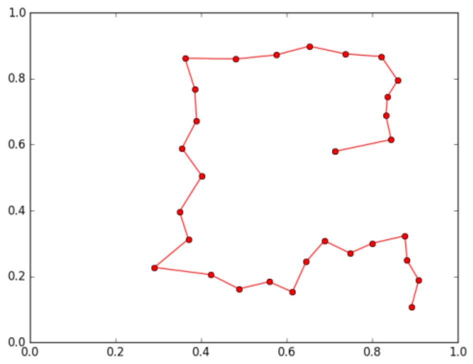


Sampling Proportional to X

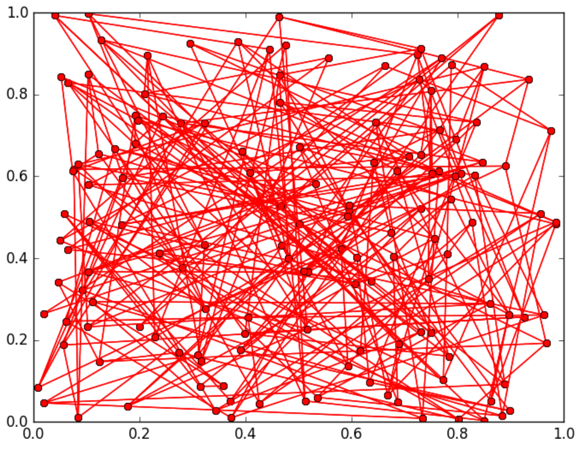
Data

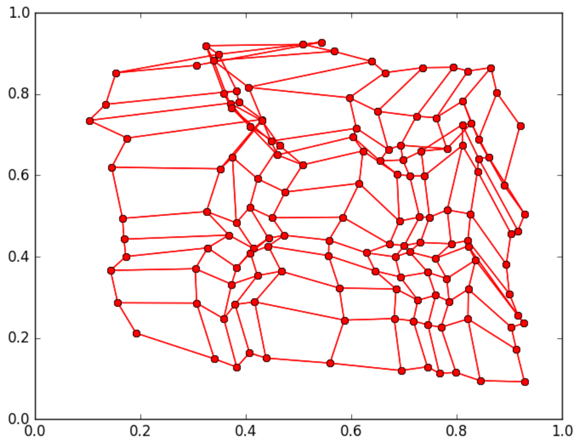


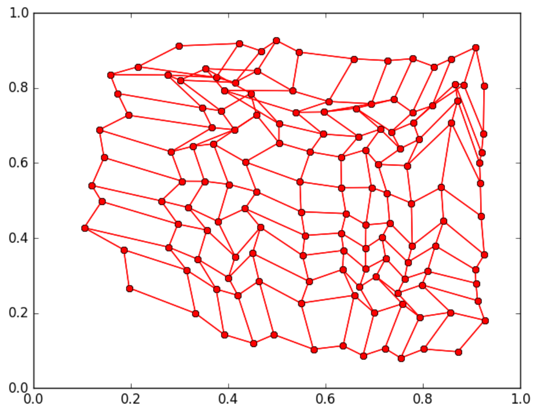
Line Topology

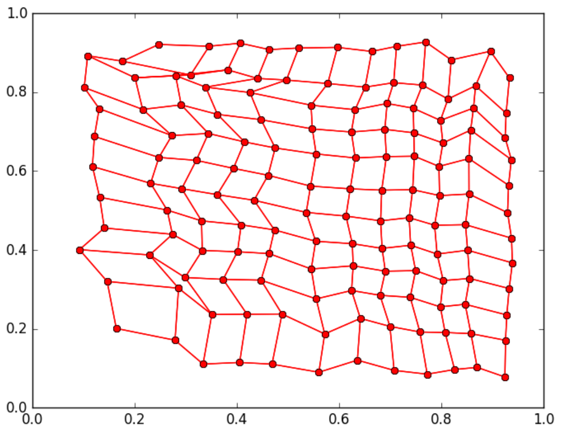





Grid topology



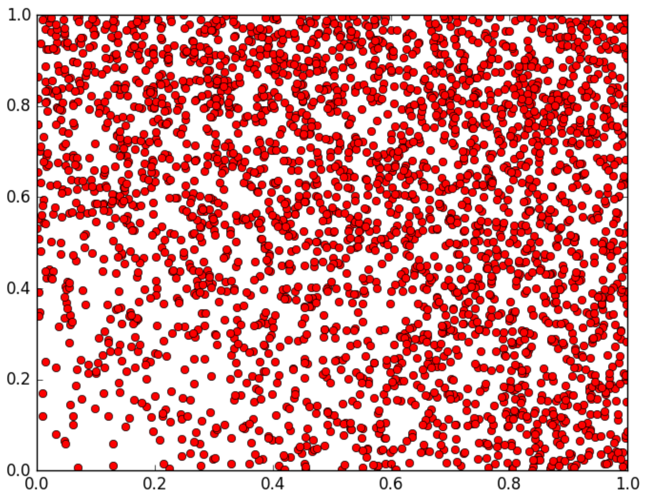




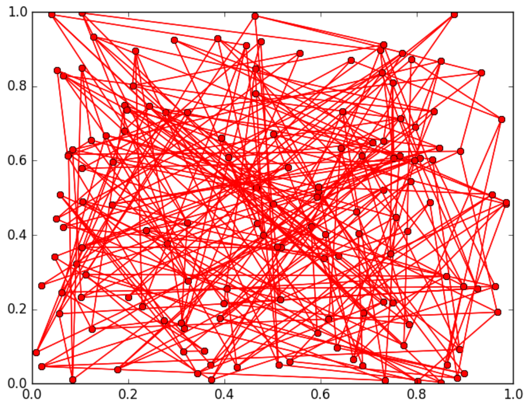


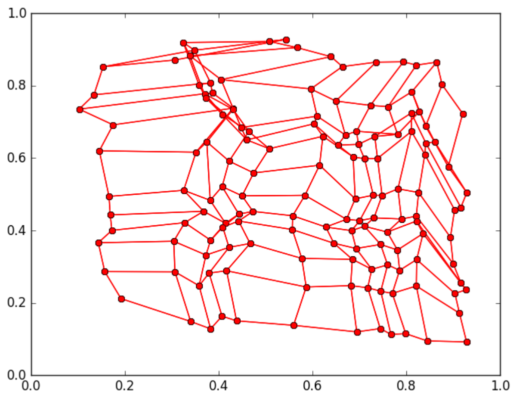
Sampling proportional to norm

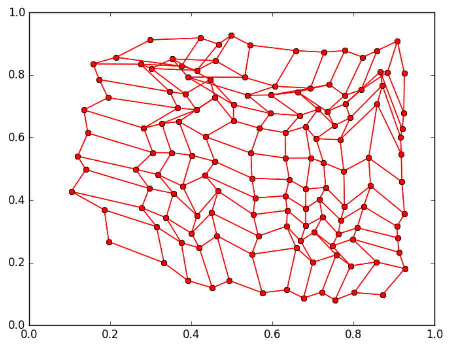
Data

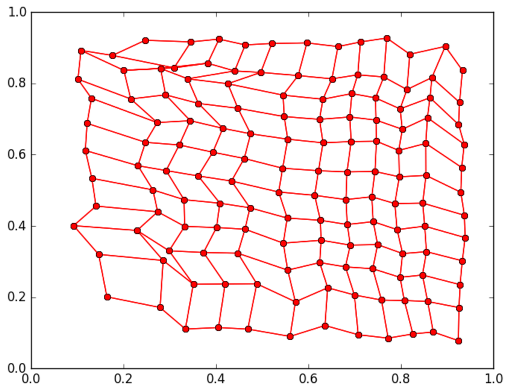


Grid Topology

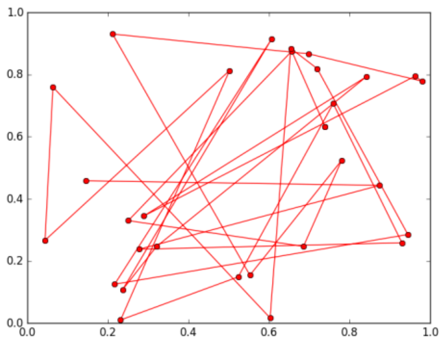


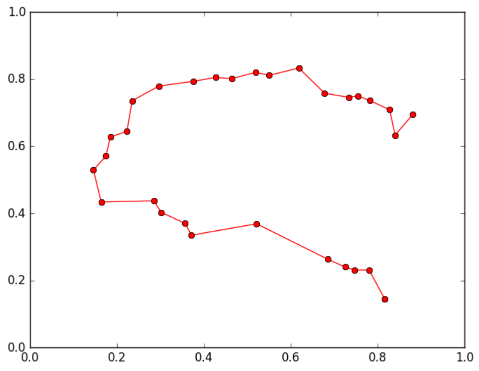


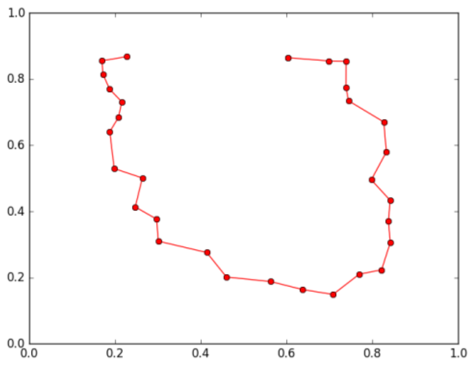


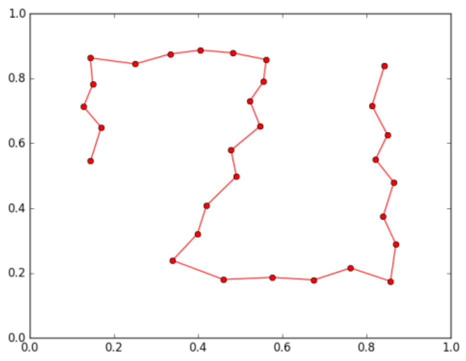


Line topology





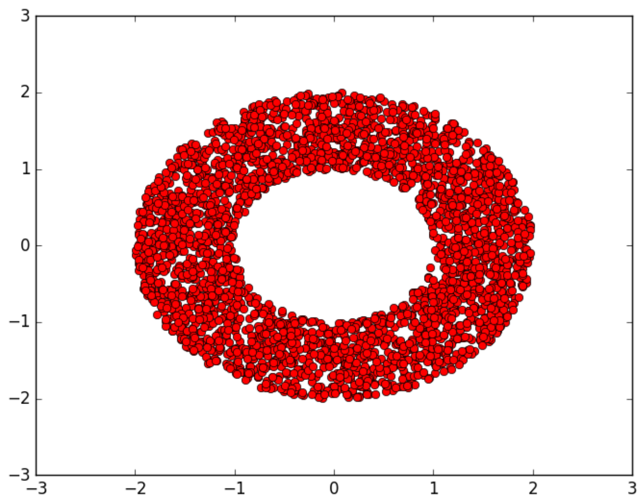




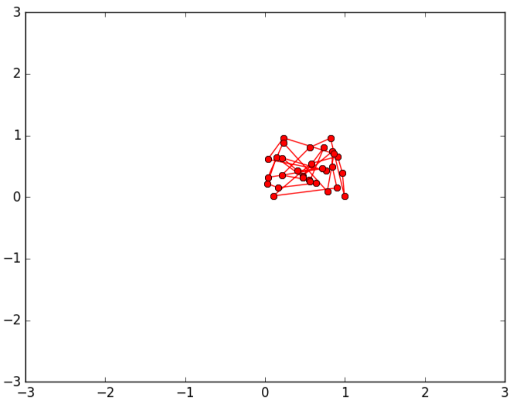
Donut

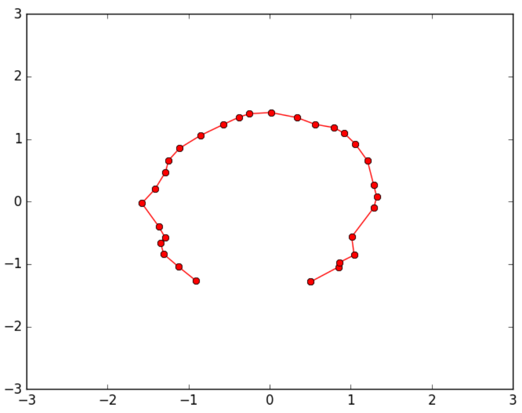
Uniform Sampling

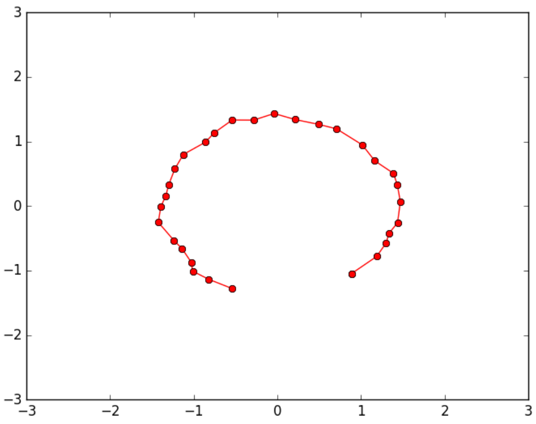
Data

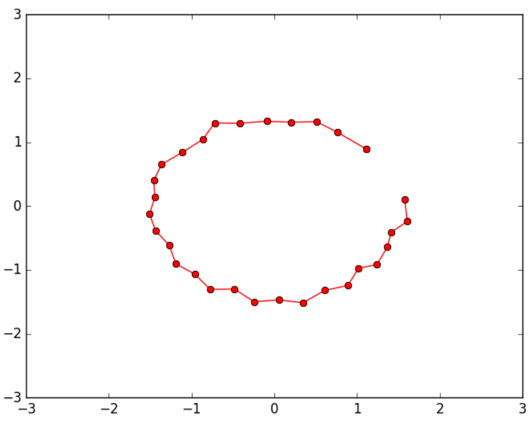


Line Topology

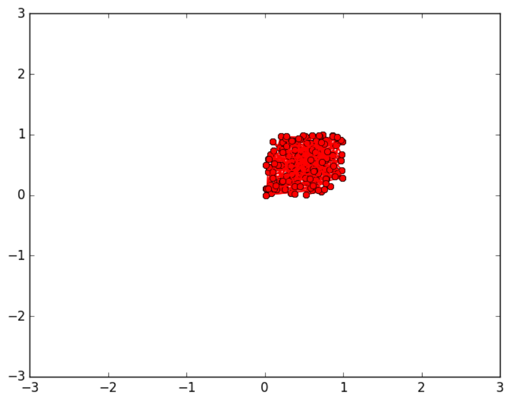


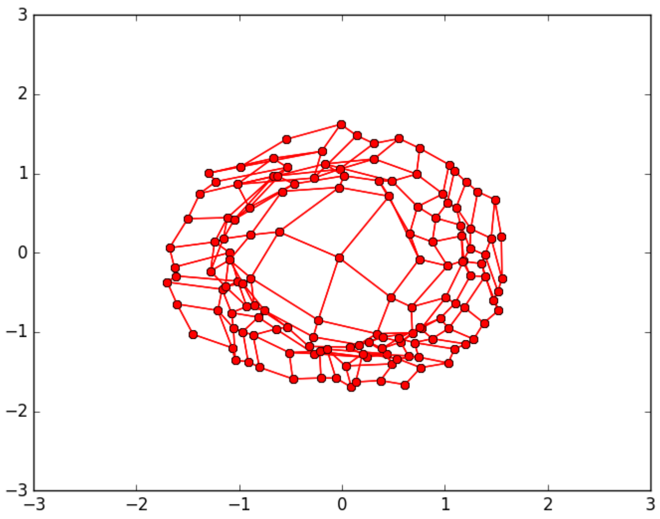


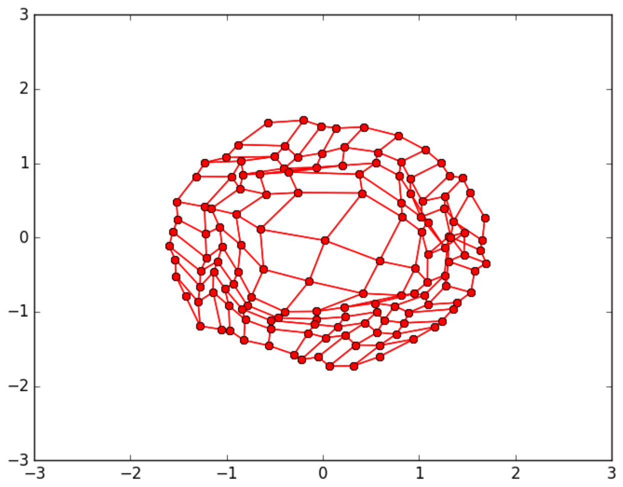


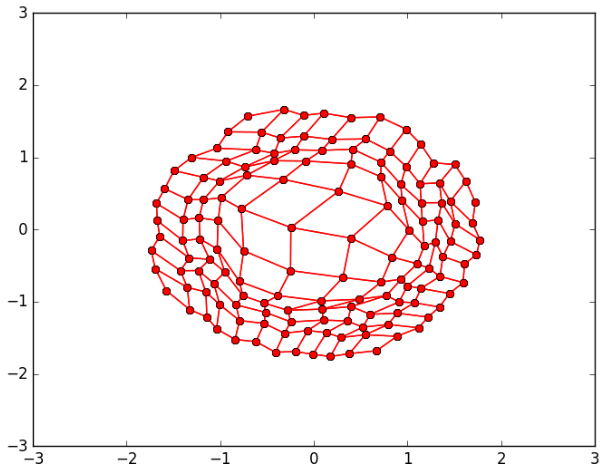


Grid Topology



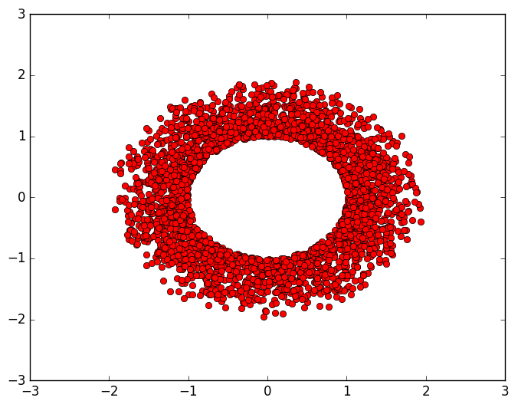




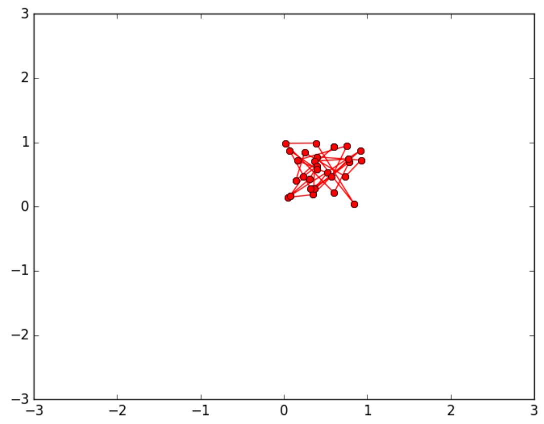


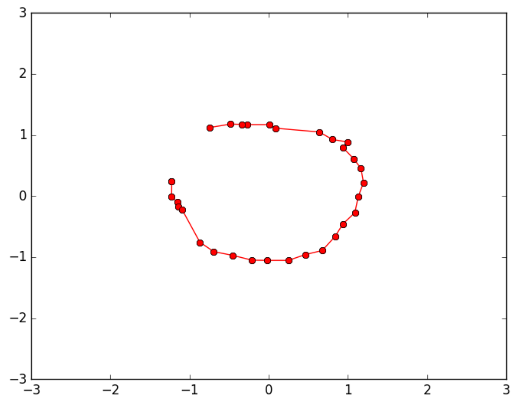
Sampling Inverse Proportional to norm

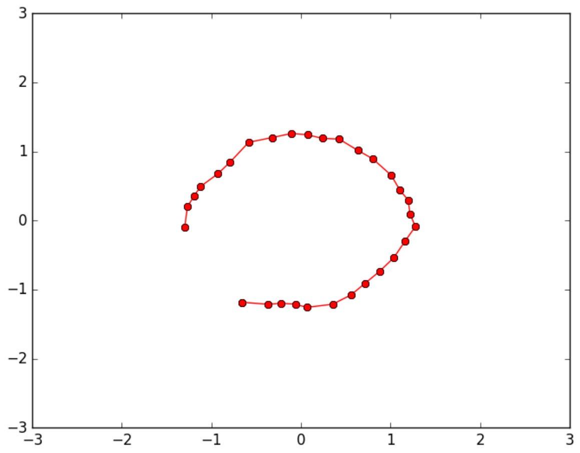
Data

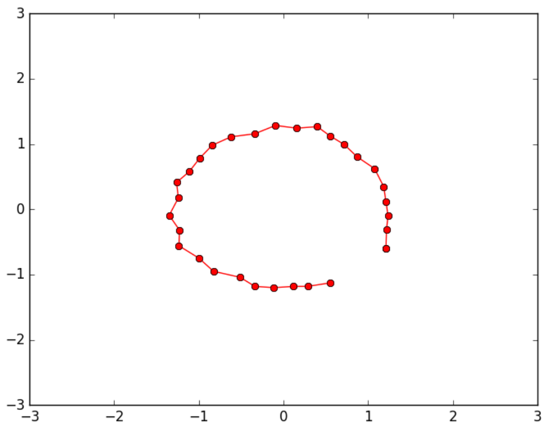


Line Topology









Grid Topology