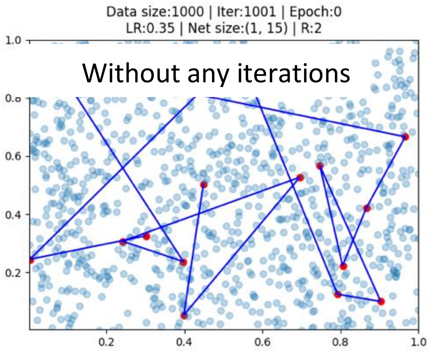
Kohonen (SOM) Algorithm – NN

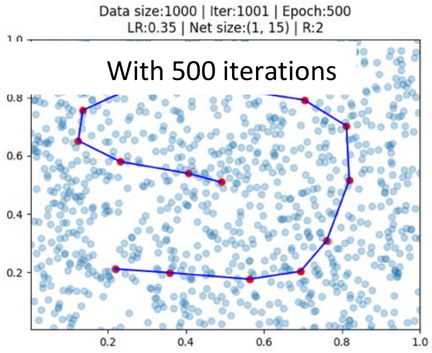
In this exercise, we implement the Kohonen network, which is a type of artificial neural network that is trained using unsupervised learning.

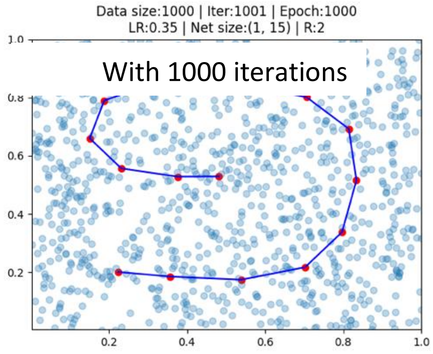
Part 1:

In this part, we will use the data (x, y) such that 0<= x, y<=1, this data is distributed uniformly and we will use 15 and 200 neurons linearly order for the Kohonen level.
First, let's look at the results and then I'll explain them:

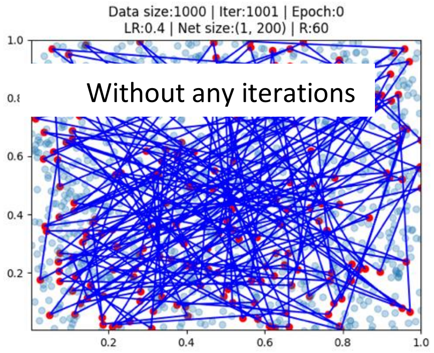
15 neurons:

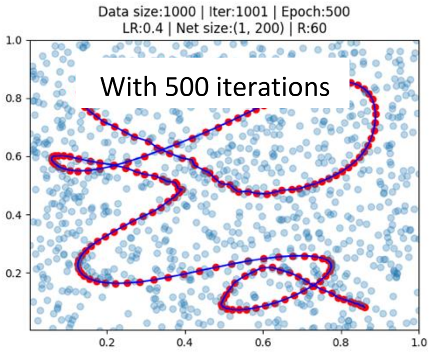


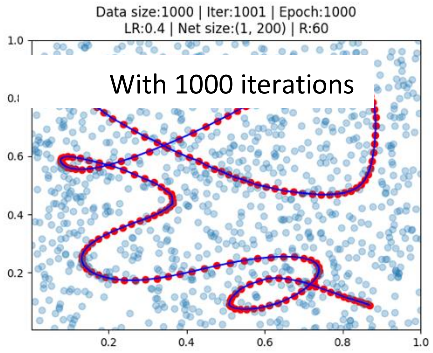




200 neurons:







You can see the different parameters at the top of each graph:
Data size: The size of training data.
Iteration: The number of iterations that the algorithm will do.
Epoch: current iteration.
LR: learning rate.
Net size: the shape of the neuron network.
R: This parameter makes the neurons "connected" more strongly, bigger R make neurons influence this neighbor More:
R = 2∂ (t) ^2
(This YouTube video explains this formula for details).

Topological Neighborhood: $T_{j,I(x)}(t) = \exp\left(-\frac{S^2_{j,I(x)}}{2\sigma(t)^2}\right)$

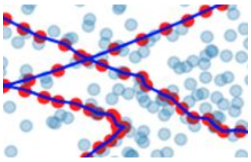
The result is as expected, etch data point "pull" the nearest neuron and his neighbors so we expect this kind of shape that tries to spreads evenly in the field.

At 15 neurons net, we spread widely the neurons and allowed R to be small because there is a lot of space that 15 neurons need to fill.

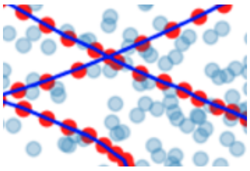
At 200 neurons net, we tighten the neurons and make R big (60) because as much as the R-value is lowest the neuron disperse inconsistently; this weakens the connection between the neurons and impairs our ability to conclude the network.

We can see that as the number of iteration increase the line is smoother: this is happening because the LR is getting smaller in etch iteration so the changes are gentler.

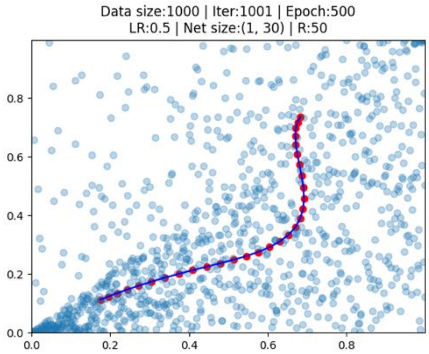
500



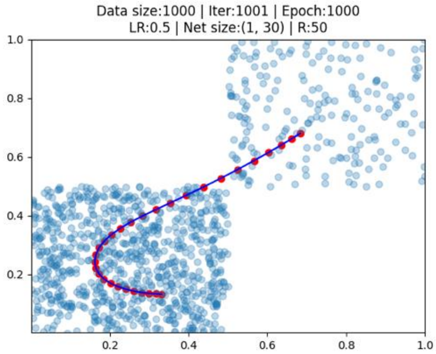
1000



Now let's look at non-uniform distribute data:



At the first data, the likelihood to pick point between 0 to X is 80% and from X to 1 is 20%:

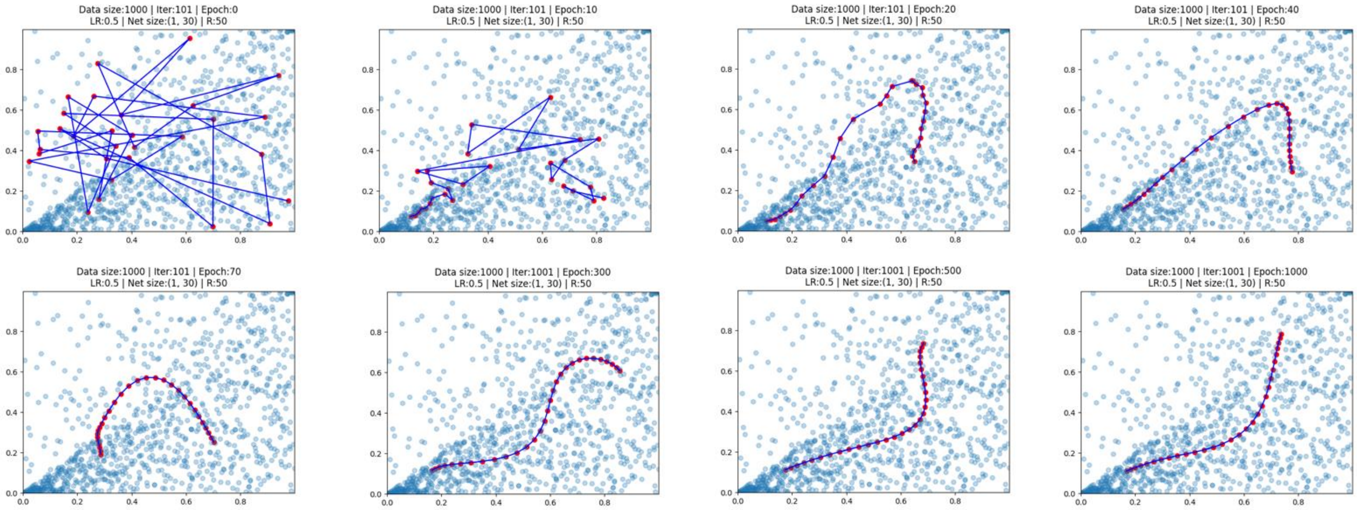


At the second data, the likelihood to pick point in range (0, 0.5) is 80% and from 0.5 to 1 is 20% (for x and y):

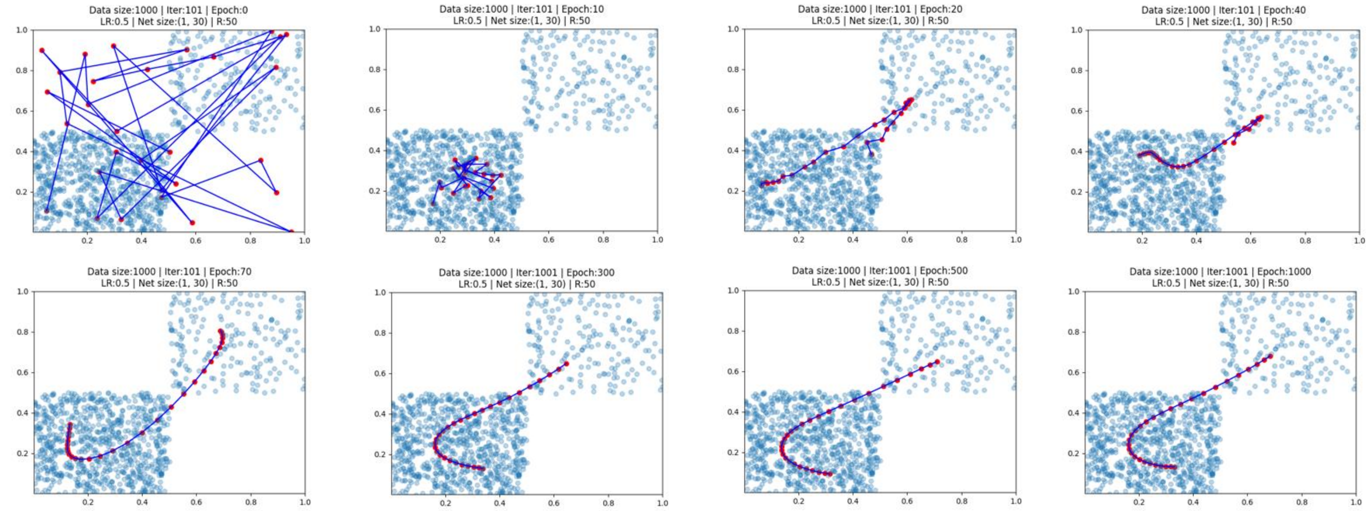
It can be seen that the neurons tend strongly towards the higher the likelihood!

Kohonen map evolution:

Data set 1:



Data set 2:



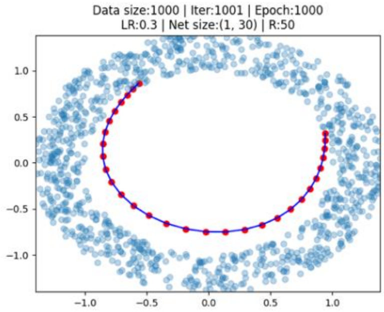
Data set 1: You can see how the neurons barely cross the main diagonal; already in the 40th iteration, you can see a clear shape of the network. In addition, as we progress with the iterations the more the web will turn towards the likelihood.

Data set 2: In the tenth iteration something interesting happens and all the neurons are between 0 - 0.5, the net is repaired quickly and already in the twentieth iteration, you can see a leak in the 1st direction. In the end, most of the neurons are between 0-0.5 and there is a little after 0.6.

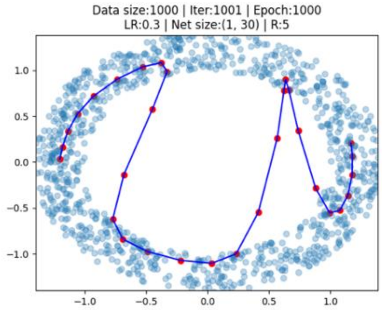
Part 2:

Now I will do the same experiments for fitting a circle of neurons on a "donut" shape with 30 neurons:

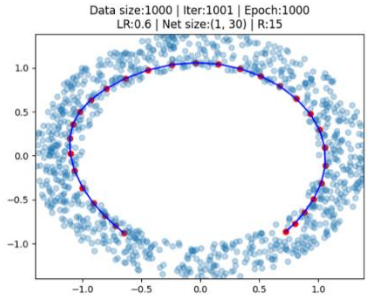
First, I tried it with LR of 0.3 and R of 50, this was not so good because the neurons were too tight to each other and 30 neurons cannot fit the donuts.



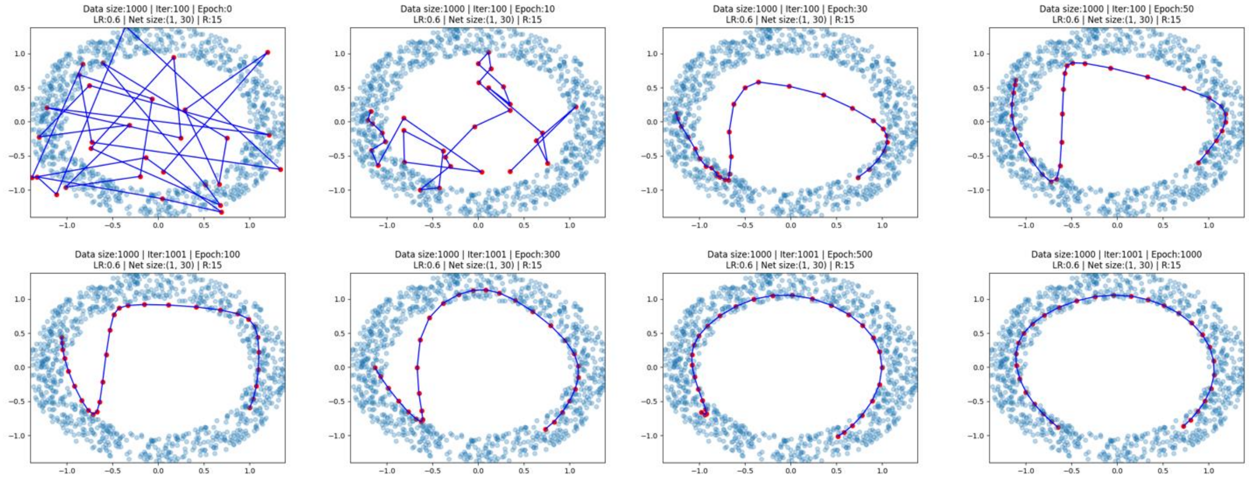
So I tried to decrease R, this also did not work because the connection between the neurons was to week:



After trial and error, I fund that R = 15 and LR = 0.6 is good, the neighbors are spacious enough but still influencing each other:



Kohonen map evolution:


Data size:1000 | Iter:100 | Epoch:0
LR:0.6 | Net size:(1, 30) | R:15
Data size:1000 | Iter:100 | Epoch:10
LR:0.6 | Net size:(1, 30) | R:15
Data size:1000 | Iter:100 | Epoch:30
LR:0.6 | Net size:(1, 30) | R:15
Data size:1000 | Iter:100 | Epoch:50
LR:0.6 | Net size:(1, 30) | R:15
Data size:1000 | Iter:1001 | Epoch:100
LR:0.6 | Net size:(1, 30) | R:15
Data size:1000 | Iter:1001 | Epoch:300
LR:0.6 | Net size:(1, 30) | R:15
Data size:1000 | Iter:1001 | Epoch:500
LR:0.6 | Net size:(1, 30) | R:15
Data size:1000 | Iter:1001 | Epoch:1000
LR:0.6 | Net size:(1, 30) | R:15