

## Kohonen (SOM) Algorithm - part B – NN

In this exercise, we used SOM from the Sklearn library to fit our needs we copied their "fit" function and modified it so that we can print the result at every iteration and change the data without resetting the weights.

To create "monkey hand" data we use a binary matrix of 20X20.

In sklearn\_SOM, every epoch is iteration over all the data points randomly.

In addition, each data point is sent to the step function, which updates the weights.

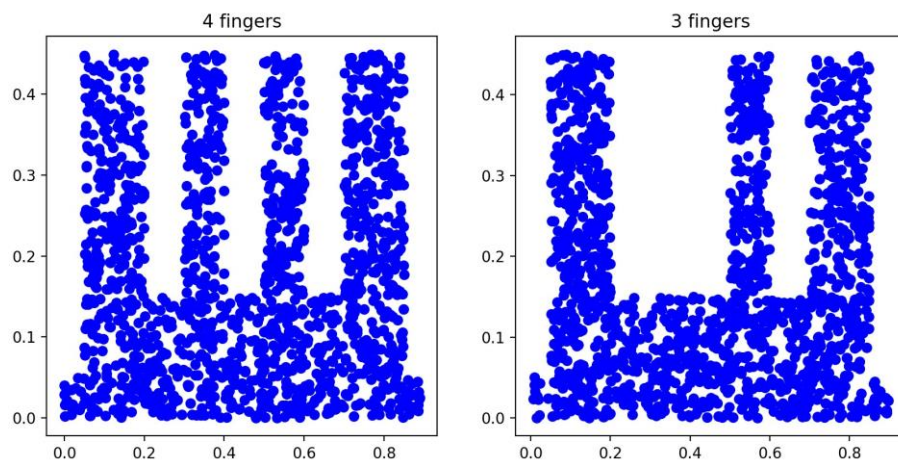
The step function is straightforward:

- Get an index of the best-matching unit.
- Find the square distance from each weight to the BMU.
- Compute update neighborhood
- Update weights

You can read all about Sklearn\_SOM [here](#).

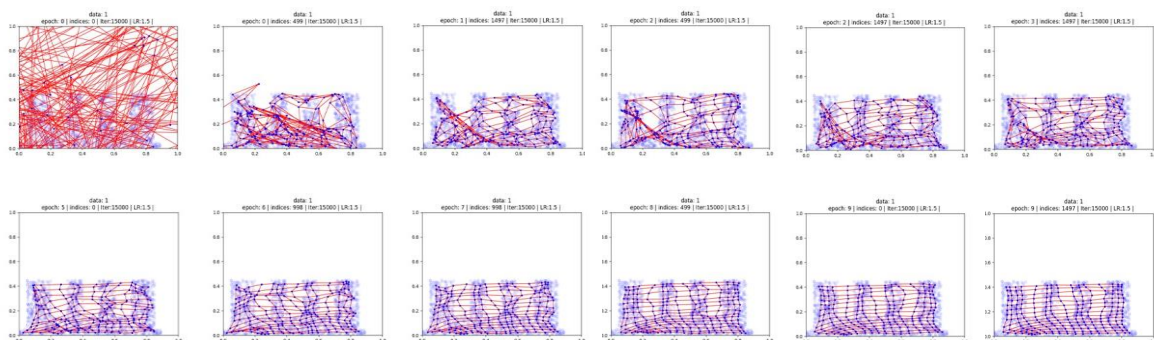
### Part 2:

1. Our data is (x, y) s.t  $0 \leq x, y \leq 1$ , in a shape of a hand:



We run sklearn SOM on this data with a 15X15 neuron net (225).

On these data, we can see how the neurons spread over the blue surface:



At the top of every picture, we wrote the parameter of every step.

**Epoch** – iteration over all of the data ones.

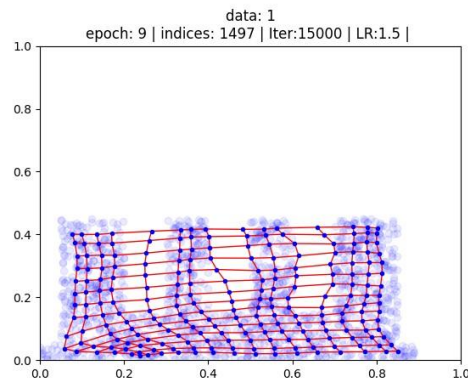
**Indices** – iteration over each data point (at the data we have 1500 data points).

**Iteration** – Max iteration (every epoch is 1500 iterations).

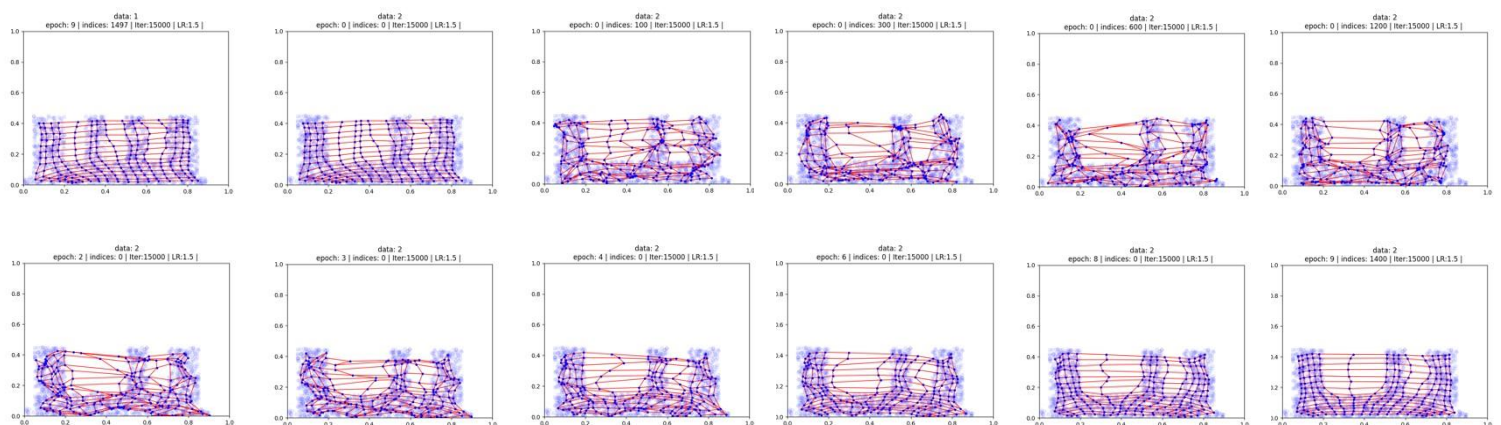
**LR** – learning rate.

It is noticeable that in the first iteration the neurons were spared randomly. Already in the 500 iterations, it can be seen that it begins to conform to the form of the hand, from the 3 epochs there are only small adjusts.

**Result:**



2. After fitting the net to "4 fingers" we will change the data to "3 fingers" and observe how the network rearranged:



The first image is the result from the first data set we can see how the network rearranged itself to the new data set, from the beginning the LR is very small because it is a direct continuation of the previous LR.

**Result:**

