Ariel Arturo Goubiah Gamboa Vázquez A01749802

Kohonen network

Suppose that we have a Kohonen network with weight matrix W and we create a new set of weights by scaling W by some positive constant c. How would this change affect the network's decisions? Justify your answer and include an example that supports it.

When we multiply the set of weights by an arbitrary constant, it is really hard to know what is going to happen. We can try to analyse the phenomenon by thinking about the geometrical effects this would have, for instance, in the example provided in class, we can clearly see that the neighbourhoods would change dramatically, ergo we should expect the performance of the network to drop.

Learning vector quantization

Given a LVQ network with two inputs and two outputs, provide an example where the simple LVQ training rule will fail to learn how to classify the input (its accuracy will always be zero after the network is trained). Please specify the input data (including the labels), the weights of the network, the labels for the output neurons and a graphical representation of your example. Explain the reasons that make LVQ useless for this particular case (the reasons why it fails to learn) and what changes to the basic LVQ learning method could be implemented in order to properly classify the input.

We could just provide the XOR example because we know that any algorithm that works using distances or some simple heuristic to characterize two labels will fail in this particular problem. Of course, this is a two-output, single input problem, but we could just assign the second output arbitrarily, or maybe the NOT(XOR). XOR and NOT(XOR) are linearly dependants but still, the LVQ can not solve the XOR problem, so this example still works.

Kohonen networks and the XOR problem

Ariel Arturo Goubiah Gamboa Vázquez A01749802

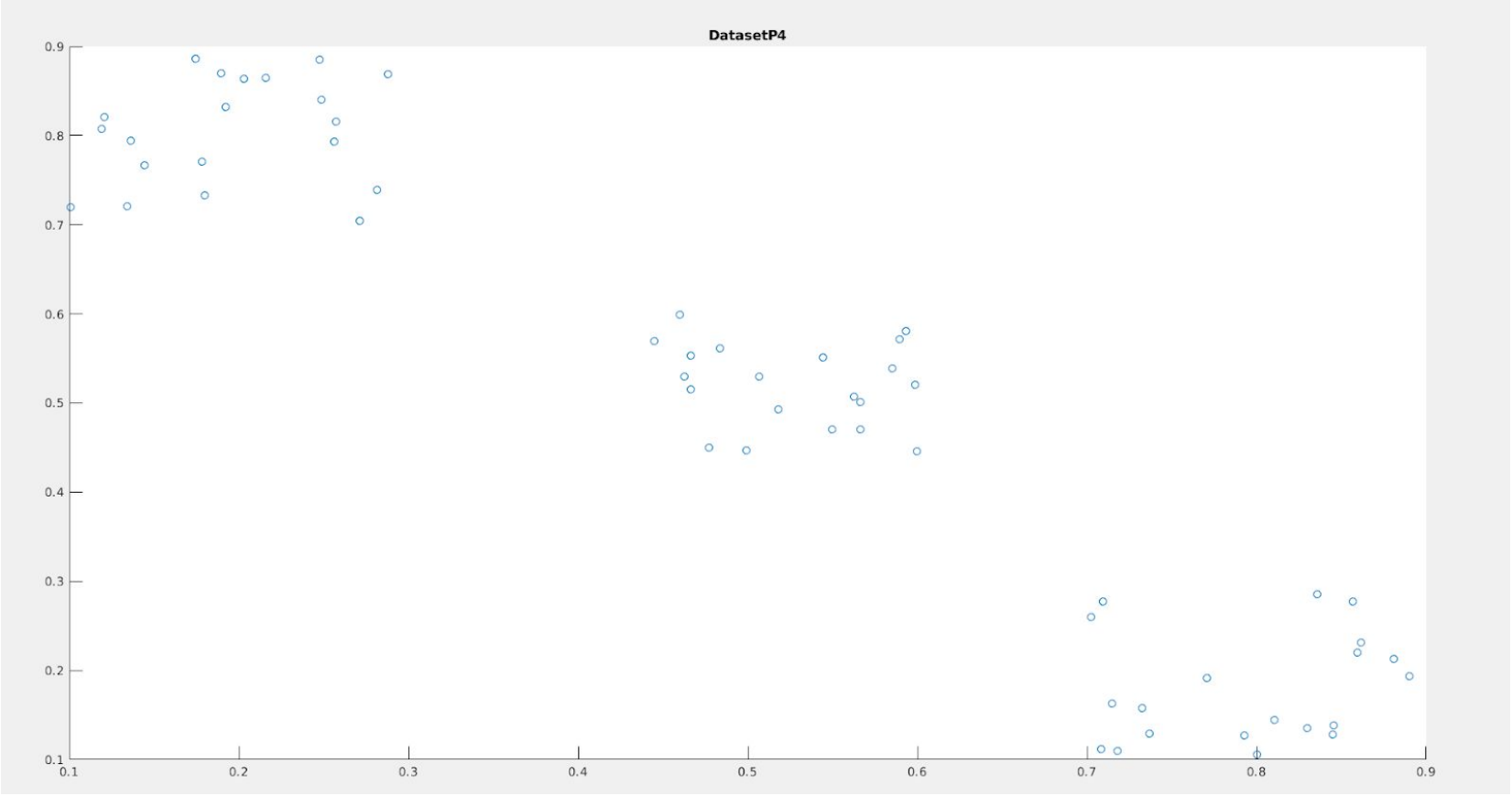
We have previously stated in this course that it is not possible for a single-layered perceptron to solve the XOR problem since it is not linearly separable. Now that you know about self-organizing neural networks, do you think is possible for an LVQ neural network to solve the XOR problem? Justify your answer and include an example that supports it.

As we have stated in problem two, we can not solve the XOR problem by simple heuristics such as distance, so we can confidently say that the LVQ can not solve the XOR problem as well. Furthermore, I will provide an example: if we train an LCQ using only the 0,0 and the 1,1 that correspond to the same label under the XOR problem, and if we train the 1,0 and 0,1, we would get the same weights for the Kohonen network, which is 0.5,0.5.

Ariel Arturo Goubiah Gamboa Vázquez A01749802

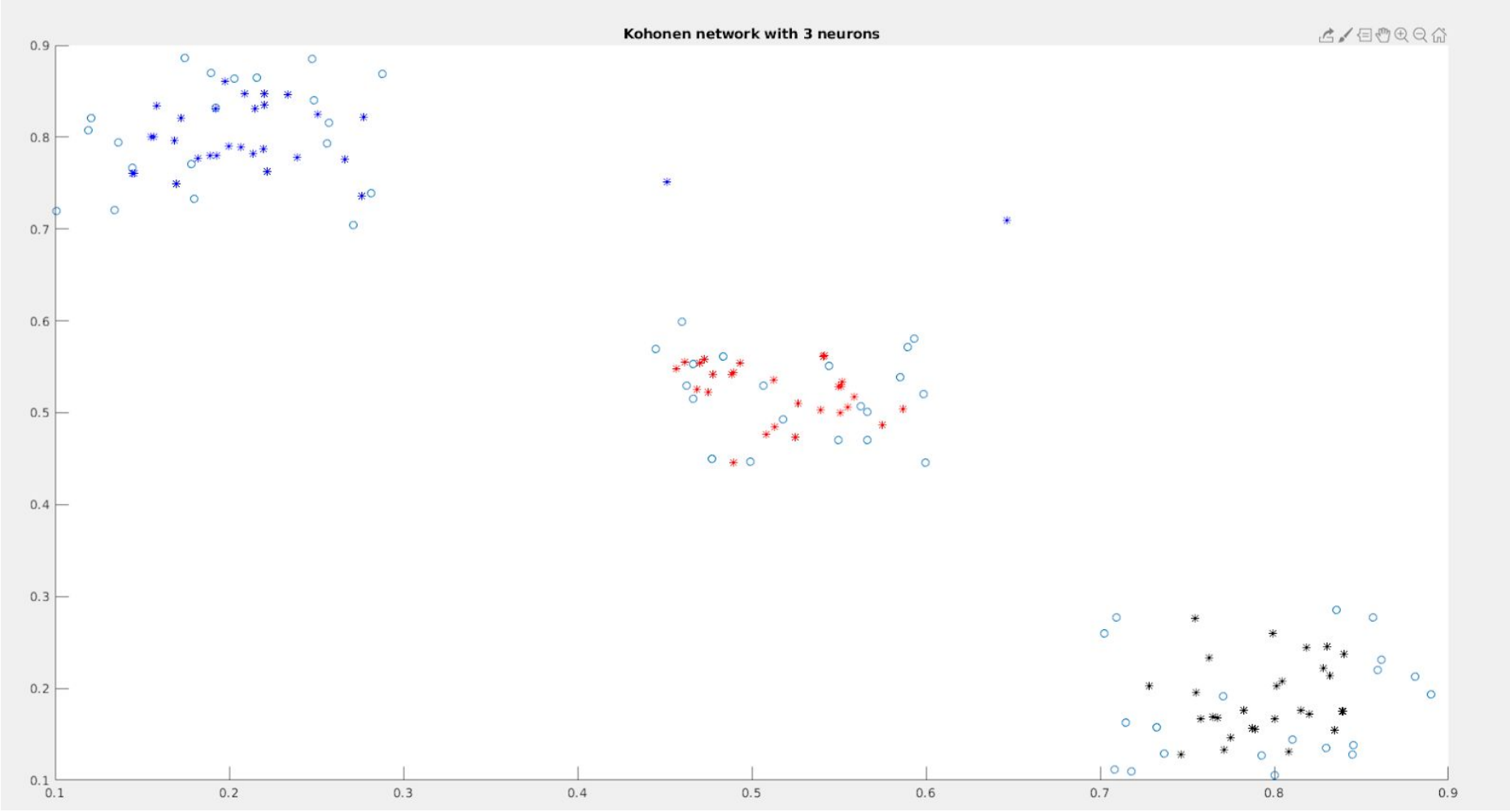
Clustering

- Plot the data and, by inspection, propose a set of centroids that properly cluster the data.
    - Good centroids would be [0.2, 0,8], [0.5, 0.5], [0.8, 0.2]

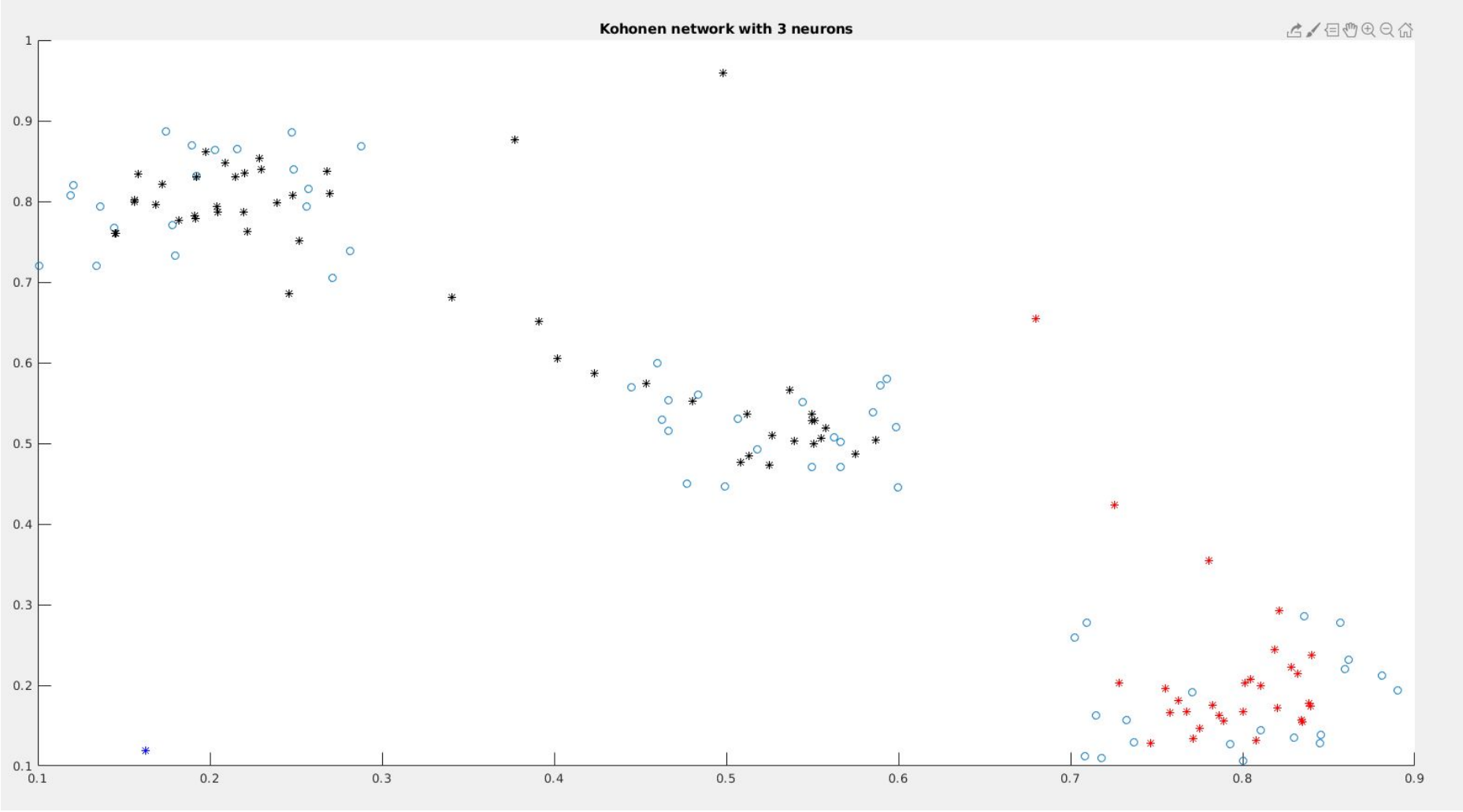


- Write the code to train a Kohonen network and train it for the dataset by using 3, 4, and 5 output neurons. Because the weights are randomly initialized, you will likely run your training process more than once. Among the different runs of the training process, what is the best configuration you obtained? How is the performance of the network affected by the number of output neurons?
    - The best configuration is with three, because it is the number of clusters that are actually linearly separable.
    - The code for this exercise can be found in https://github.com/iamerroralpha/ComputationalIntelligenceHW8

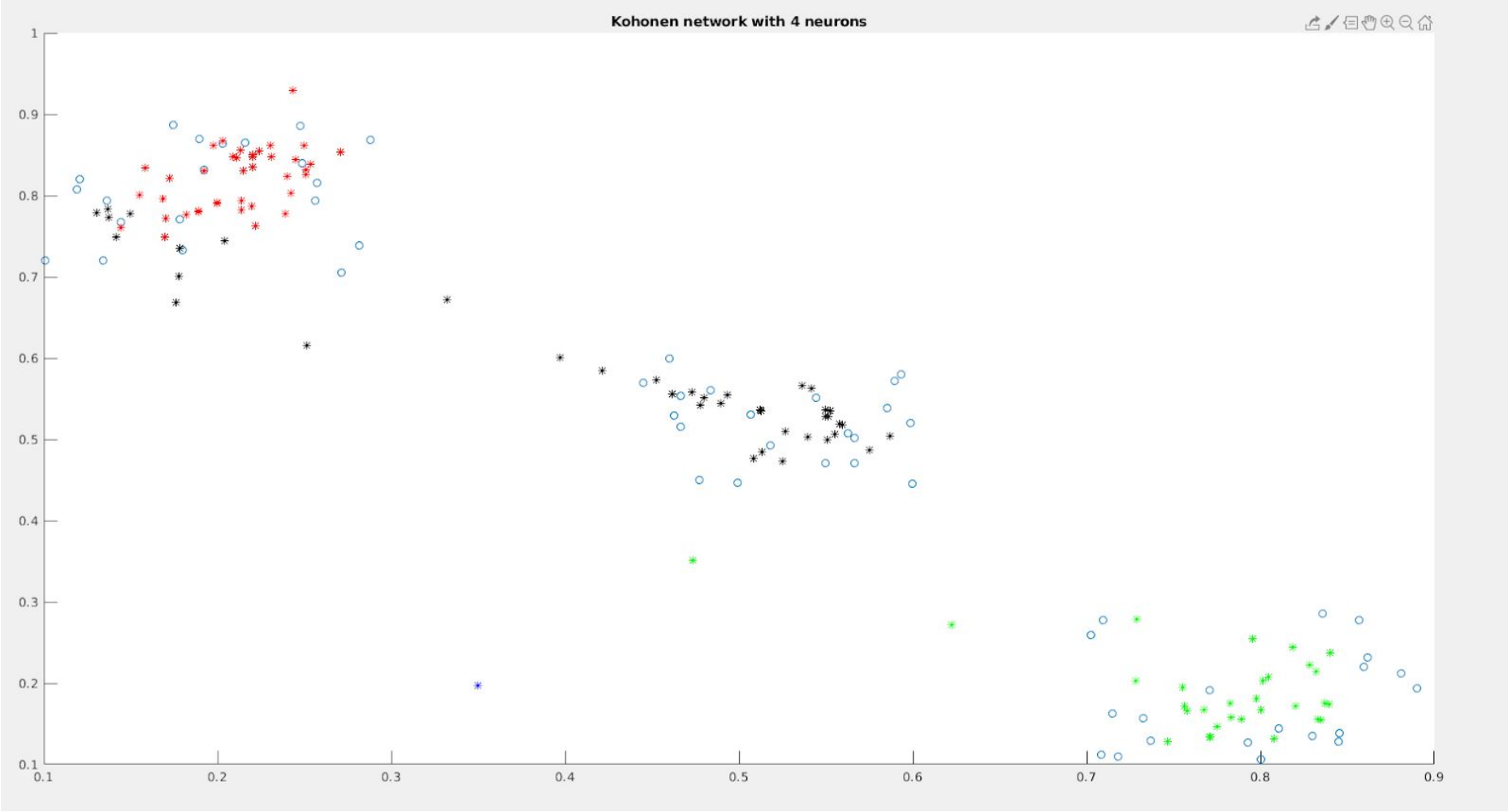
In the following example, the network works fine, the blue, red and black dots represent the evolution of the weights of the randomly inicialized neurnons.



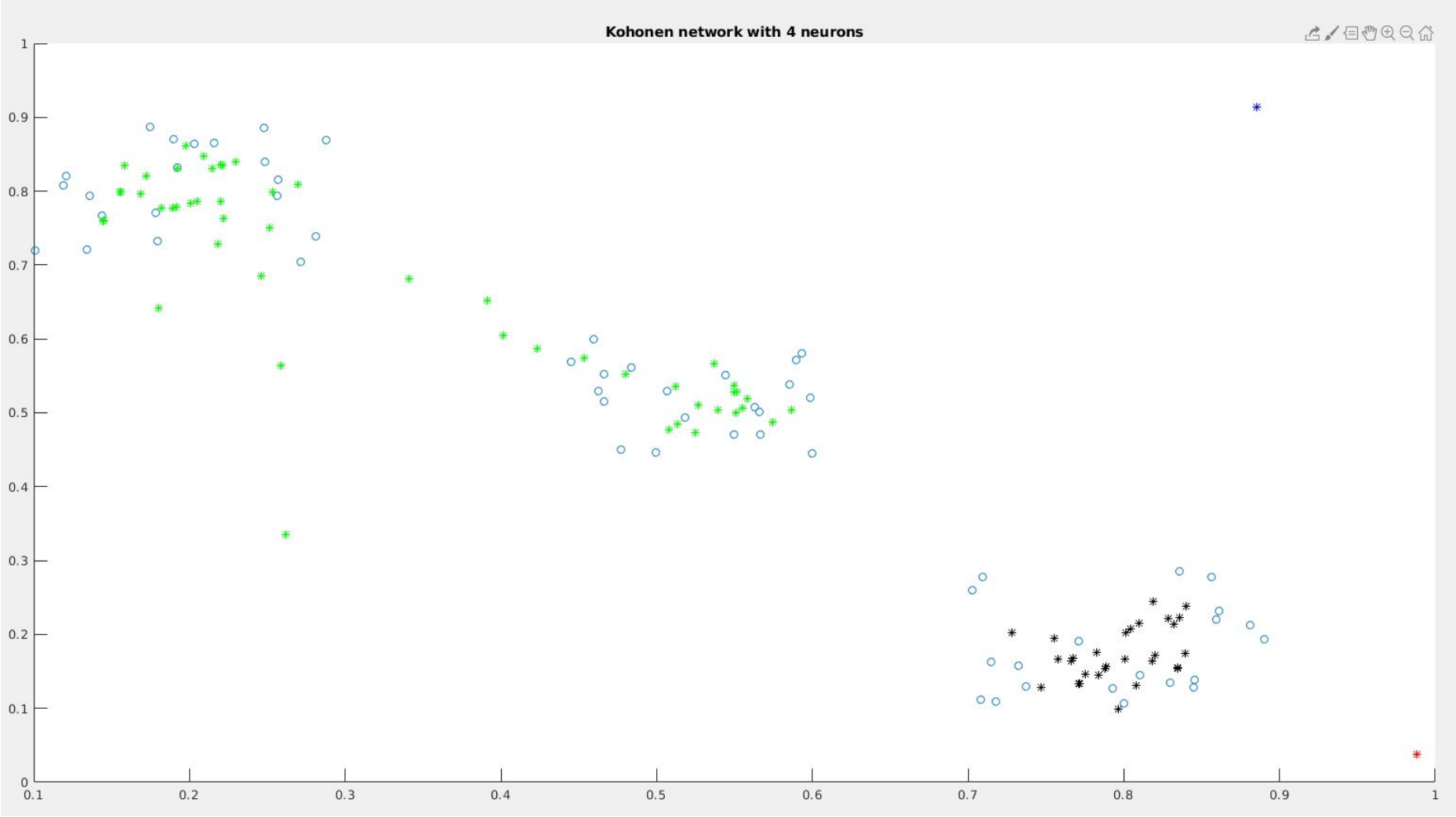
In this other one, the network does not solve the problem, because it is just too far to be relevant to the data points and compete, the blue neuron does not compete and stays far



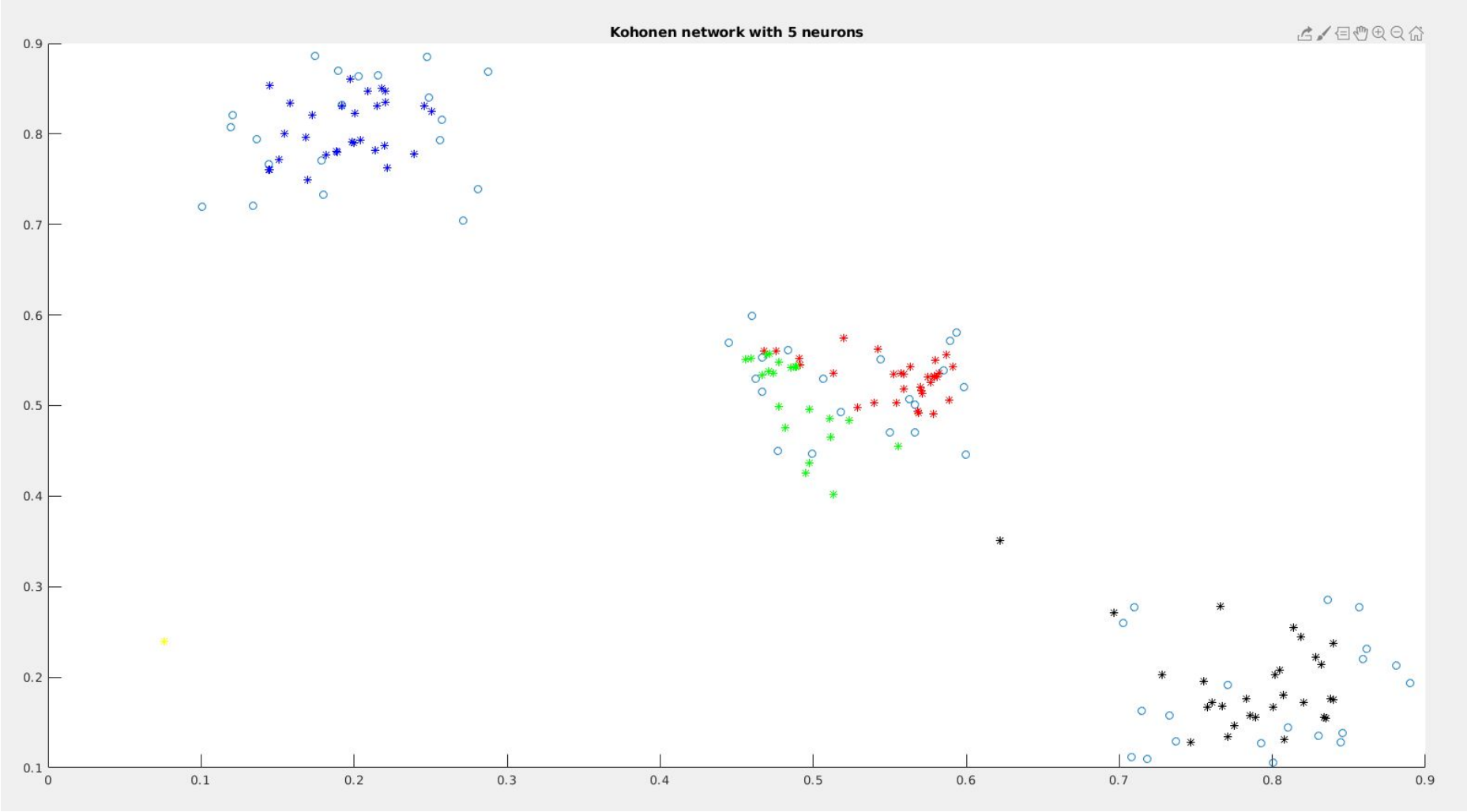
Opposite to what I thought, when we have more neurons, the network performs better, and this is because the network becomes more competitive, i.e. there is a higher chance of having neurons that correctly describe the three groups



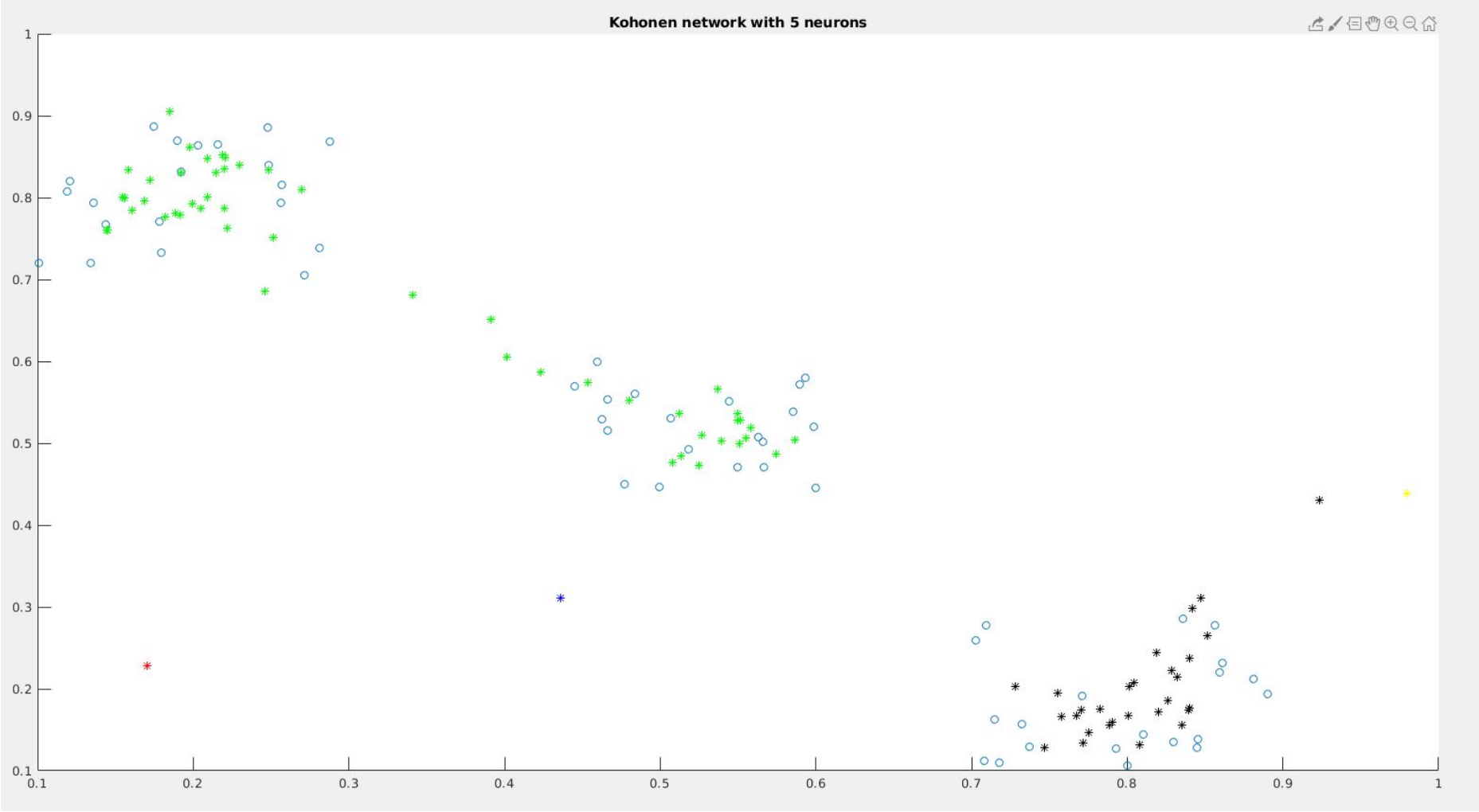
There is also a case where the neurons can not describe all the three groups properly



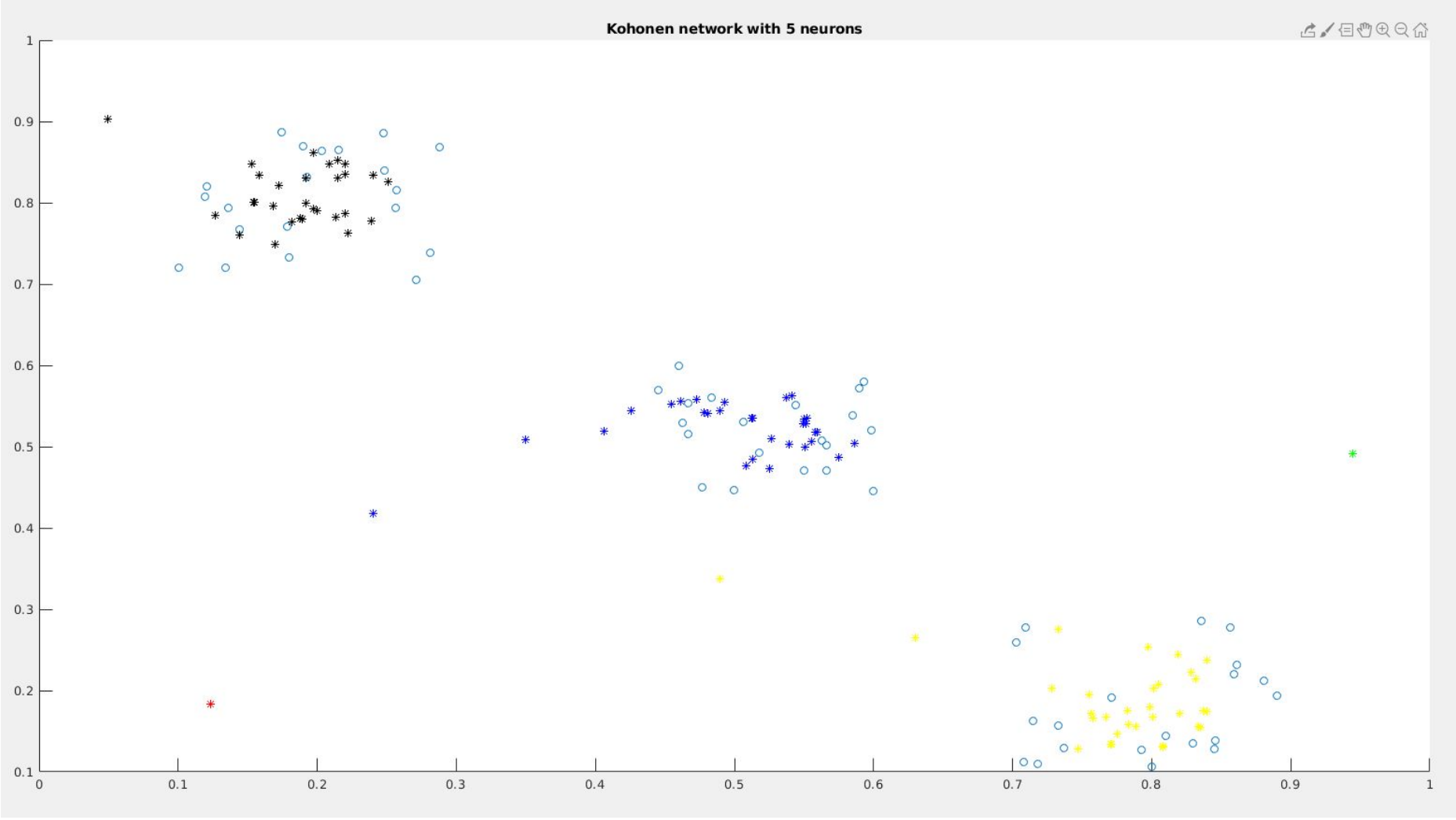
With five neurons, interesting things happen, for example, I found two neurons representing the same group



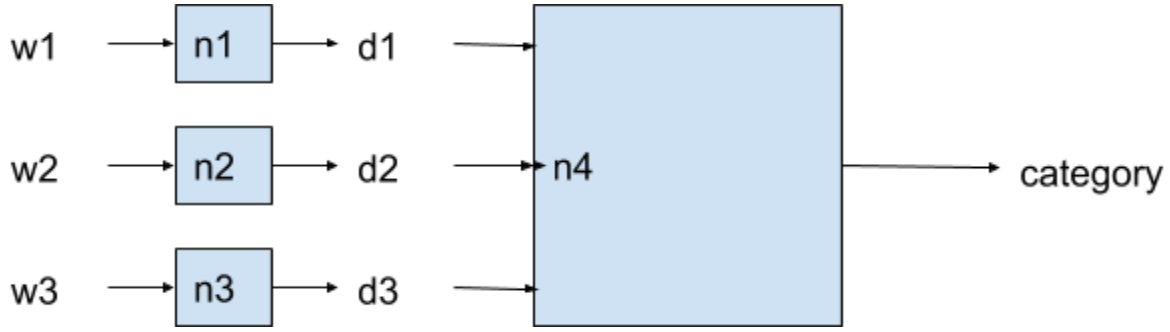
I also found some cases where the network does not classify properly.



There is also the case where the network classifies just as good as the other two cases.



- Can the Kohonen network properly cluster the dataset? In case your answer is no, justify your answer.

- In case the Kohonen network successfully clusters all the points in the dataset, provide the values of the weights of such a network as well as the diagram of the network architecture.
    - The diagram will always be the same, as it is explained in the classes' slides. Where n1, n2 and n3 are the neurons that have a better performance for a group in a given instance of the problem
    - 

    - The weights are always close to the values originally proposed, for instance [0.222, 0.76], [0.52, 0.509], [0.8594, 0.8055]
- How close is the best distribution of the centroids you obtained through the Kohonen network to the distribution you proposed by inspection?
    - It is fairly close.

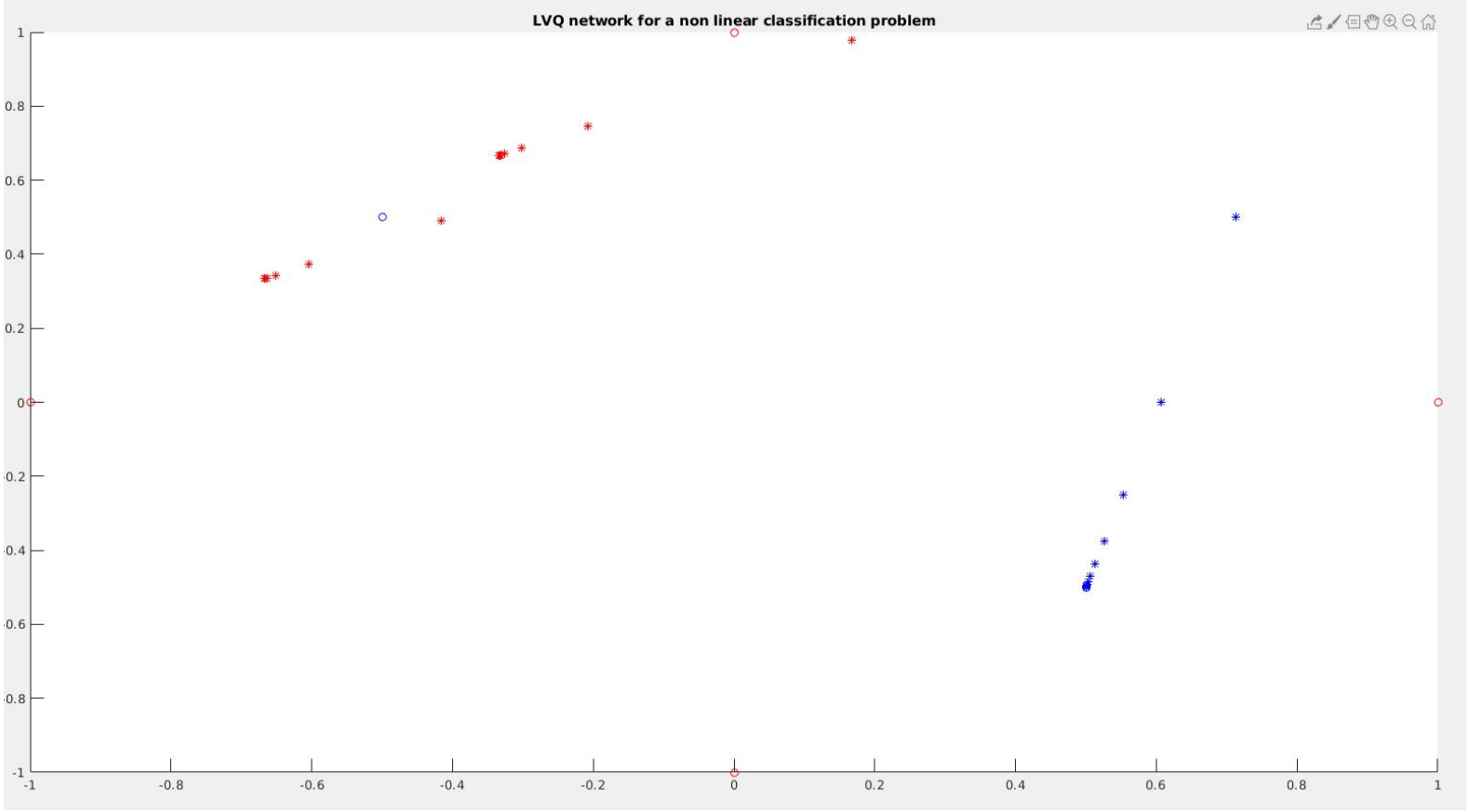
Ariel Arturo Goubiah Gamboa Vázquez A01749802

Classification

- Write the code to train an LVQ neural network. Remember that it is basically a Kohonen network but the learning rule changes so it only updates the weight of the winning neuron if it corresponds to the expected class.

  The code can be found in https://github.com/iamerroralpha/ComputationalIntelligenceHW8/blob/master/LVQ.m

  A result of the code is the following image



- Train the LVQ network on the dataset. Because the weights are randomly initialized, you will likely run your training process more than once. Among the different runs of the training process, what is the highest accuracy level you obtained?

  We can not get an accuracy higher than the 50%, because of the heuristic to determine a class, we calculate the distance between a data point and a neuron, so we can not expect to classify both labels correctly more than 50% of the times

- Can the LVQ network learn the patterns in the dataset? In case your answer is no, justify your answer. In case the LVQ neural network successfully classifies all the patterns in the dataset, provide the values of the weights of such a network as well as the diagram of the network architecture.

  The network is unable to properly classify both classes as stated above.