**Results report for Mini-Project Number 2.**

**Made by** Ariel Yechezkel, Almog Amiga and Dvir Sadon

**Introduction:**

In this assignment we were asked to implement the Kohonen (Self organizing map - SOM) algorithm and answer a few questions relating to that algorithm.

In this report we will detail the process of implementation (i.e. what we implemented) and the results of the code we wrote. Also, we will display screenshots at different iteration times of the algorithm to show the way it runs.

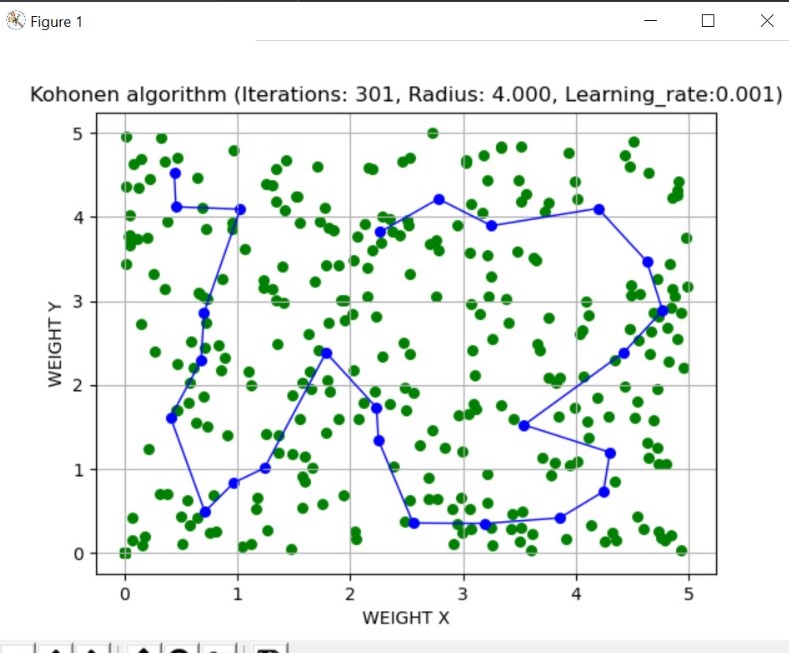
**Question A:**

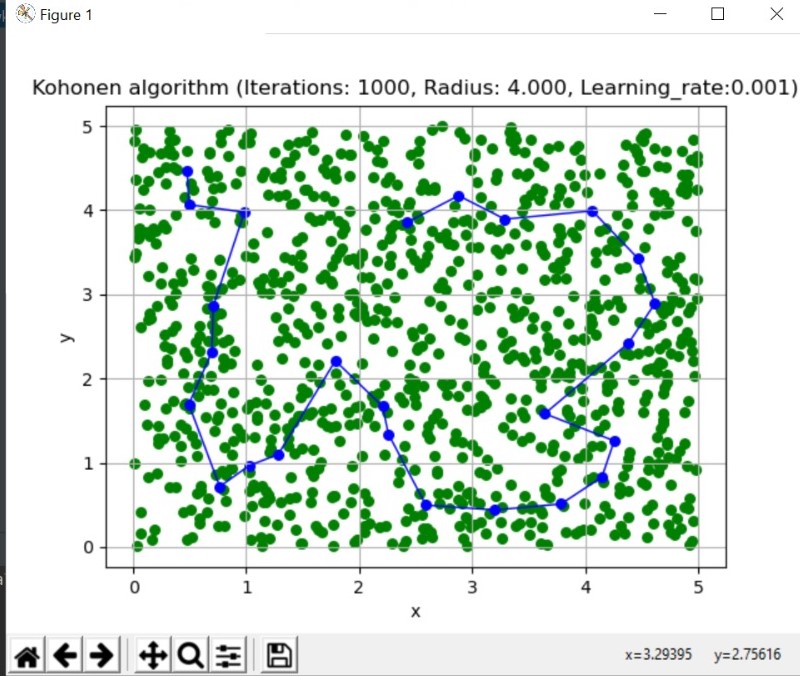
In this question we arranged our 30 neurons in a straight line and took date from data organized in a circle with a radius of 2.

The results we were able to achieve:

1. Learning rates: 0.001
2. Number of iterations until convergence: 1000

Screenshots:





**Question B:**

In this question we arranged our 30 neurons in a circle and took date from data organized in a circle with a radius of 2.

The results we were able to achieve:

1. Learning rates: 0.001
2. Number of iterations until convergence:1000

Screenshots:

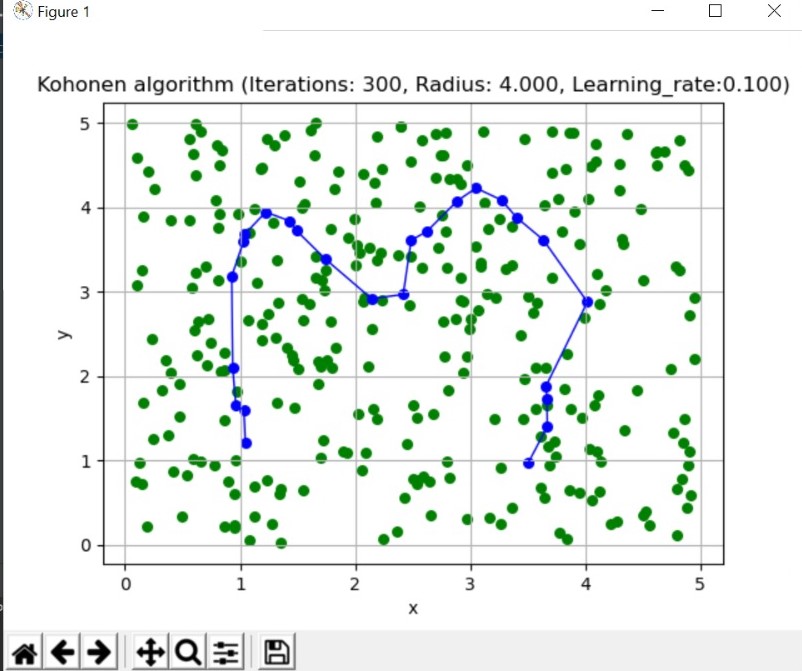
**Question C:**

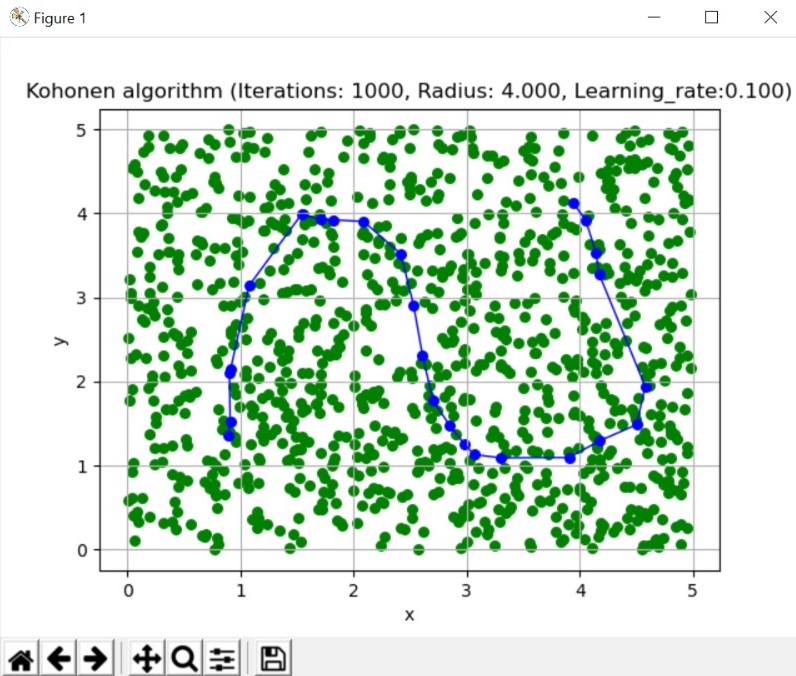
In this question we arranged our 30 neurons in a circle and took date from data organized in a circle with a radius of 2.

The results we were able to achieve:

1. Learning rates: 0.1
2. Number of iterations until convergence: 1000

Screenshots:





**Question D:**

In this question we repeated what we did in the last 3 questions, **but** the data was chosen randomly with a probability proportional to the distance from the center.

The results we were able to achieve:

1. Learning rates:
2. Number of iterations until convergence:

Screenshots:

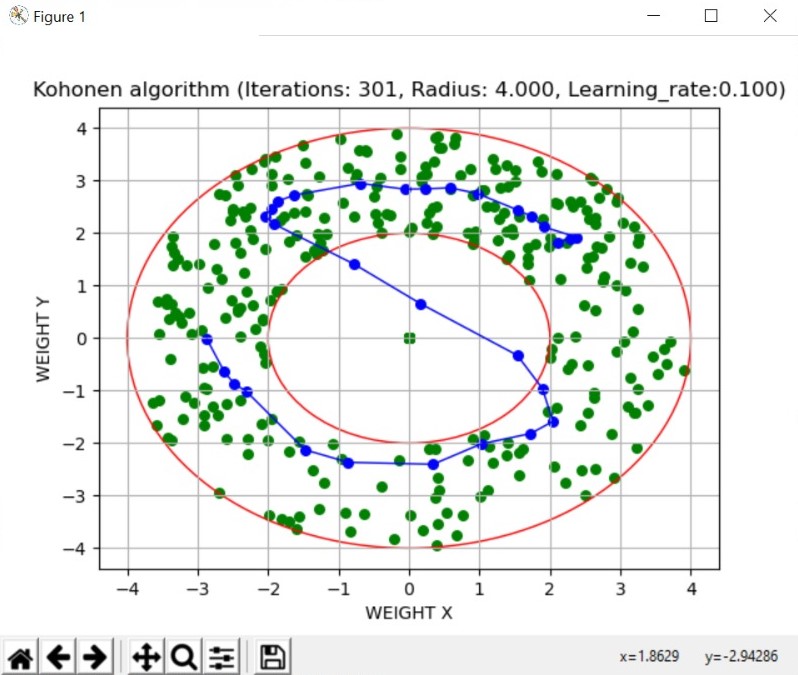
**Question E:**

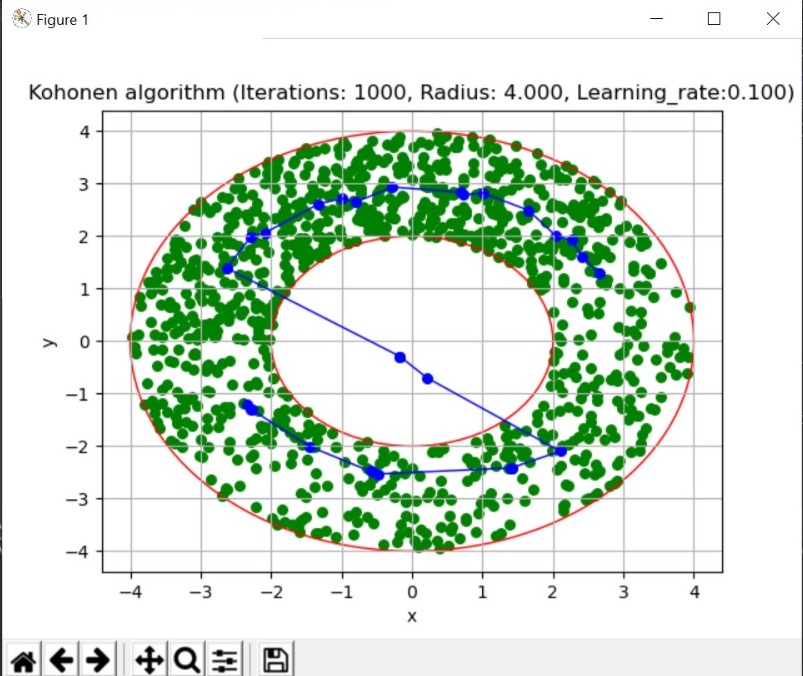
In this question we repeated what we did in questions A and B but here we have two circles of data: with a radius of 2 and a radius of 4 from the center. The chosen data is from the ring created from subtraction the circle with 4 radius the one with a radius of 2 (the ring between radius 2 and 4).

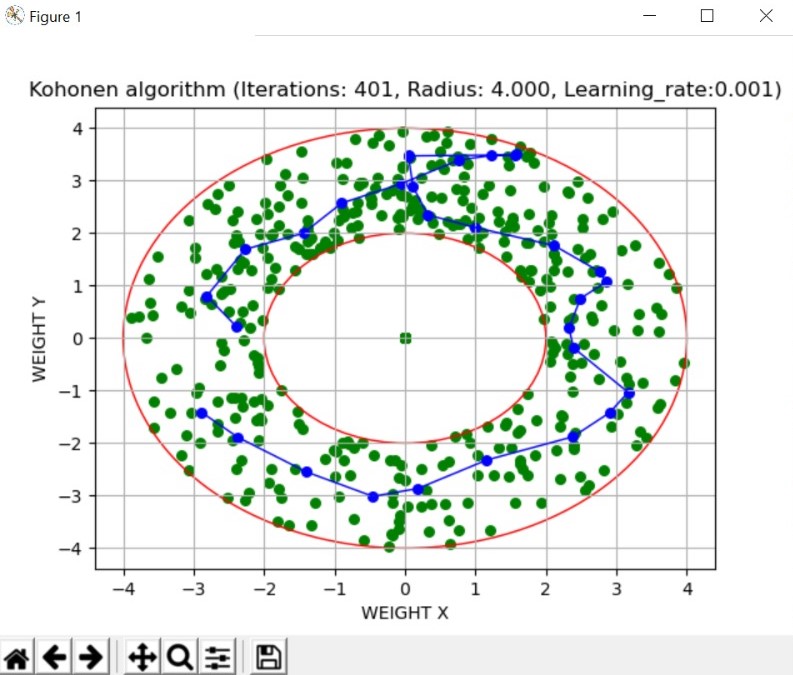
The results we were able to achieve:

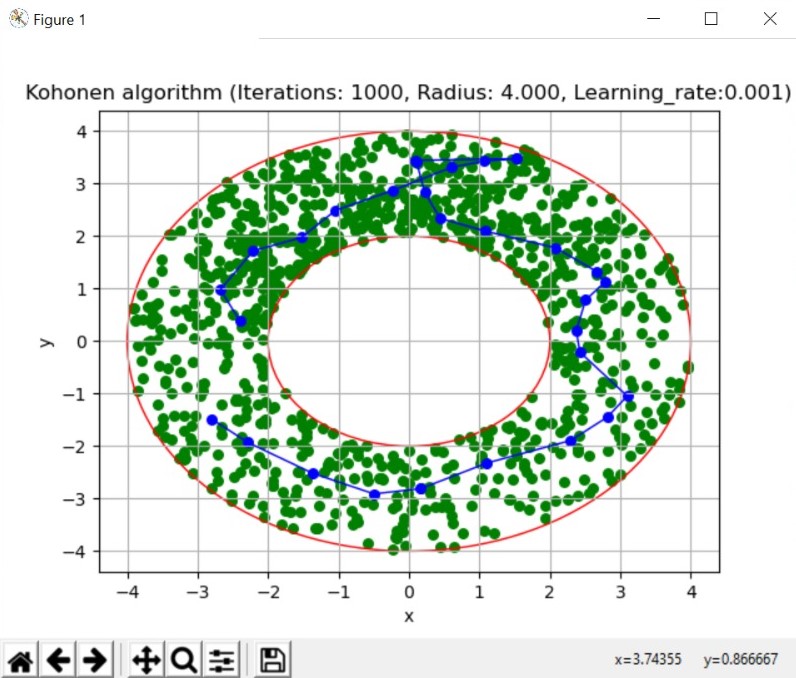
1. Learning rates: 0.001 – Best results
2. Number of iterations until convergence:401

Screenshots:









**Final remarks:**

In testing the algorithm, we ran the algorithms with 100 neurons instead of the specified amount and found that the algorithm works way better and the visualization is a lot clearer that way.

**References:**

**Thank You Very Much for Reading!**