FINAL
TAKE-HOME EXAM

Artificial Intelligence

Fall 2019

CMSC 409

Due Dec. 12 at noon

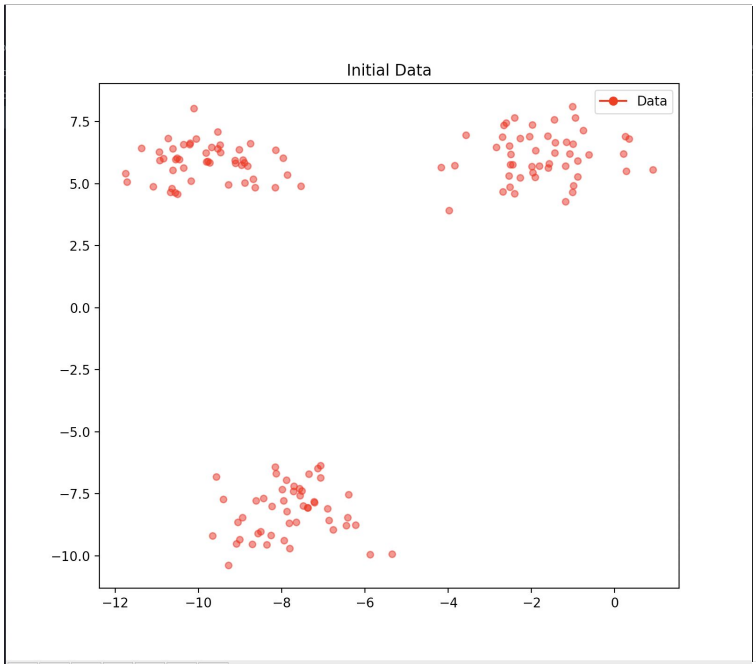
Michael Poblacion - V00852051
Name & ID

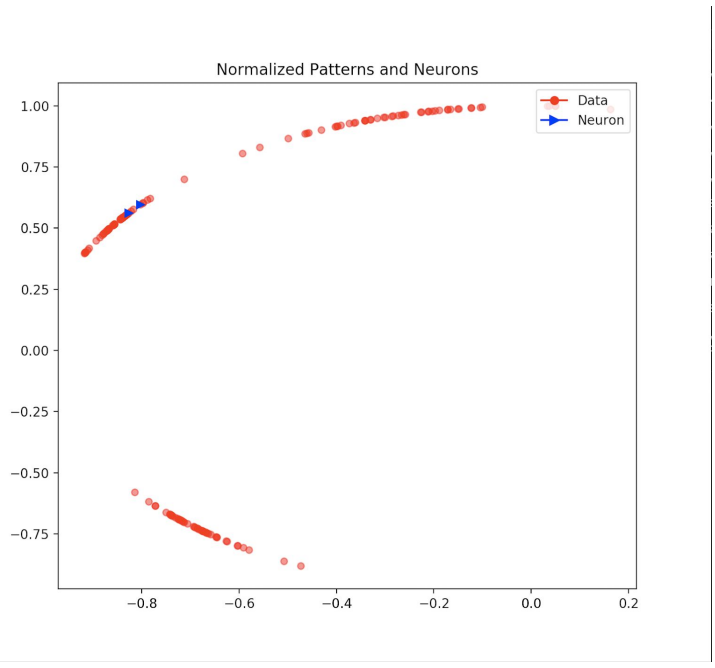
Student certification:
Please read and sign the following statement before you begin:
I fully understand that I am on my honor to do my own work on this examination. I will not share exam or any portion of it with the rest of the group that may be taking the exam after me. Further, I certify that I have neither given nor received any aid on this test.
If I violate this confidence, I may receive a letter grade of "F" for this exam, or for the course, or be expelled from the program.

Print Name: Michael Poblacion *Date:* 12/11/2019
Signed : _____ *(you can sign/scan or use e-signature)*

Ex 2.1 Competitive Learning (Kohenen's Winner Take All)

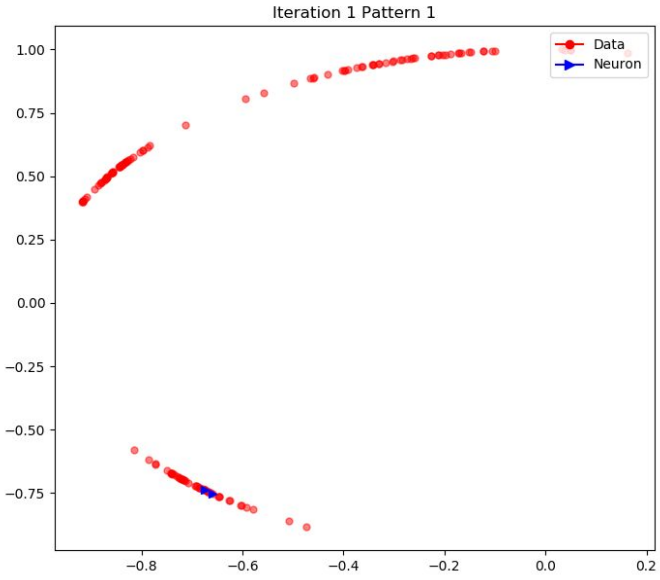
Source code located in Source directory

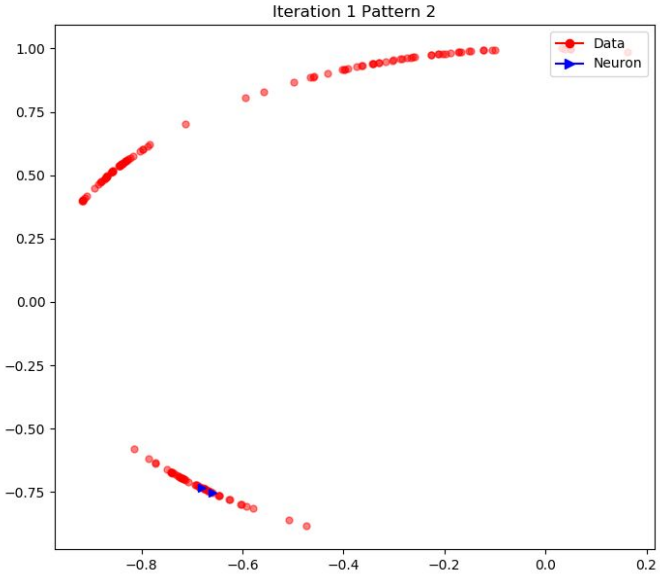


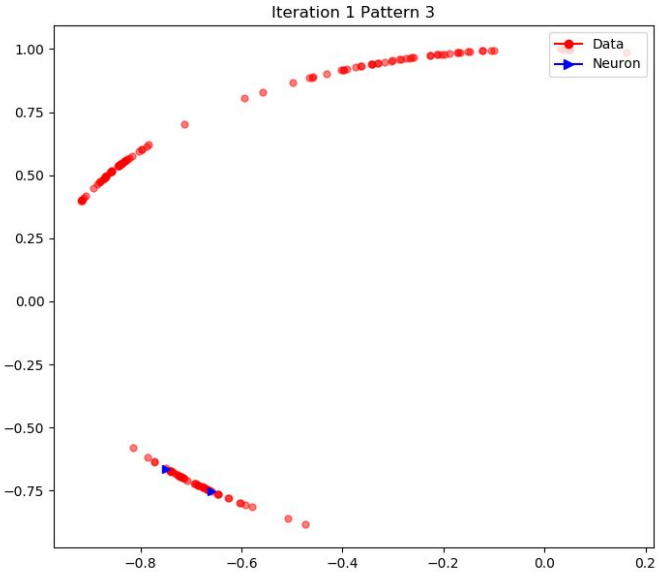


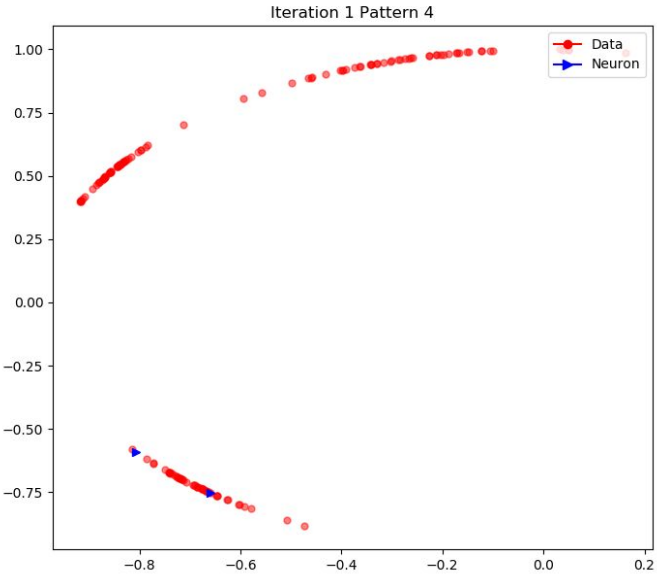
2 Neurons:

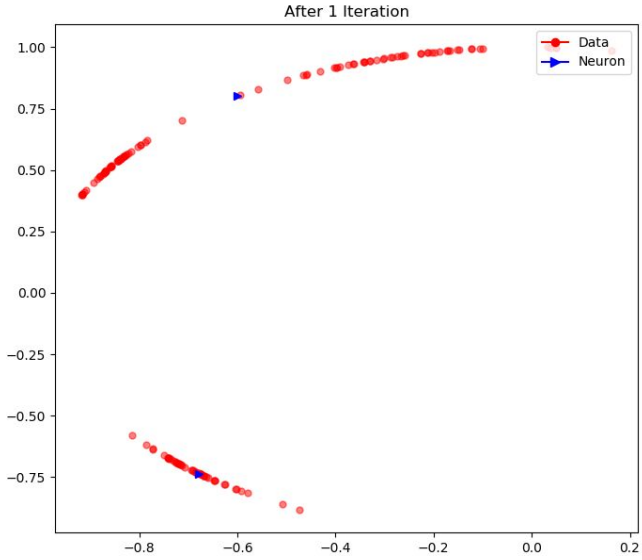
1. Random weights chosen
2. Number of iterations: 10
3. Learning Constant = .1
4. Speed of convergence: Depended on the initialization. However the average convergence was < 5 iterations.
5. Other Lessons: The initialization of weights plays a role of how quickly the neurons will converge.
6. Found 2 clusters every time.

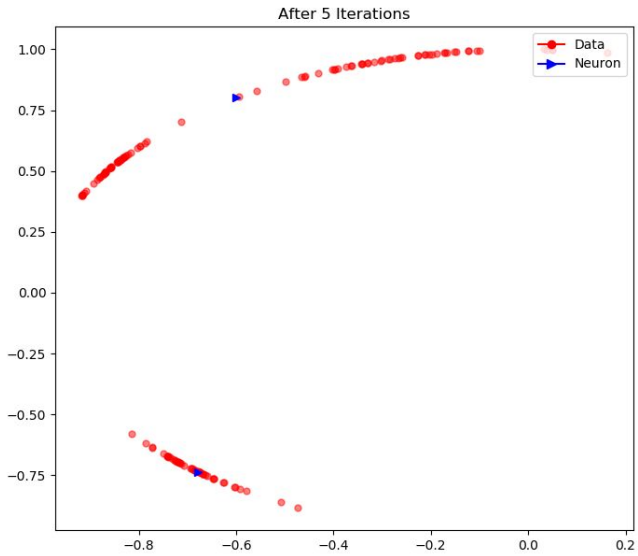


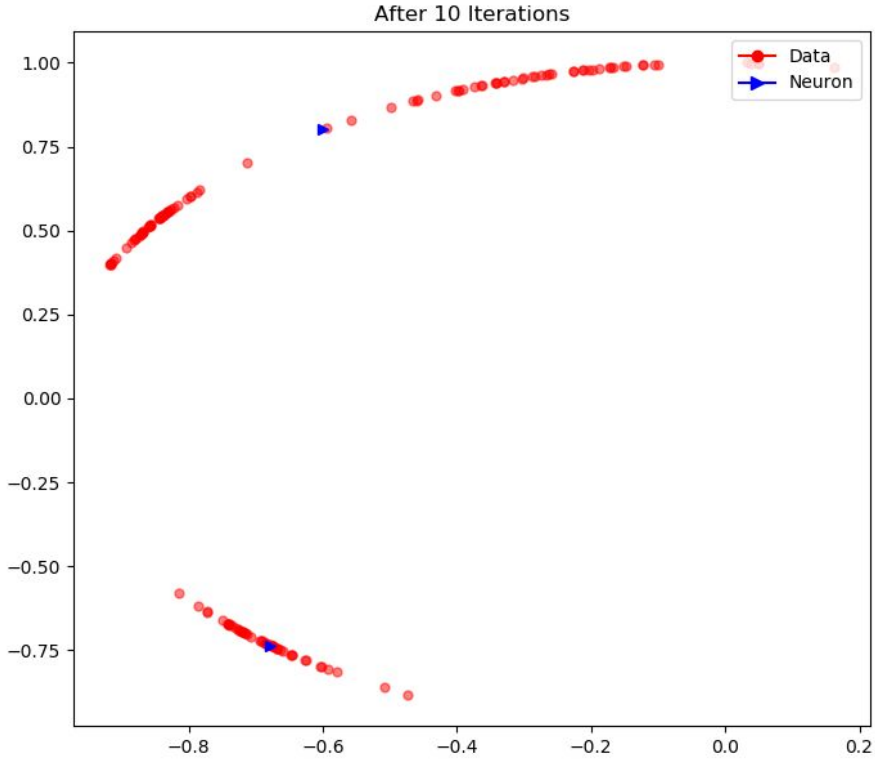







After 1 Iteration
Data
Neuron
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−0.8
−0.6
−0.4
−0.2
0.0
0.2

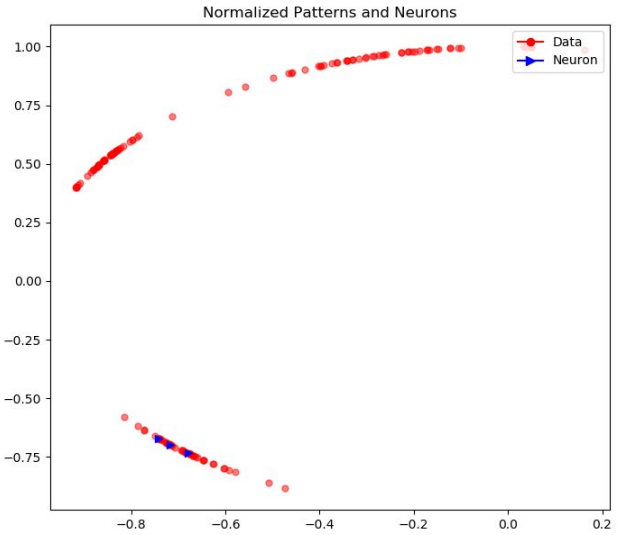

After 5 Iterations
Data
Neuron
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−0.8
−0.6
−0.4
−0.2
0.0
0.2

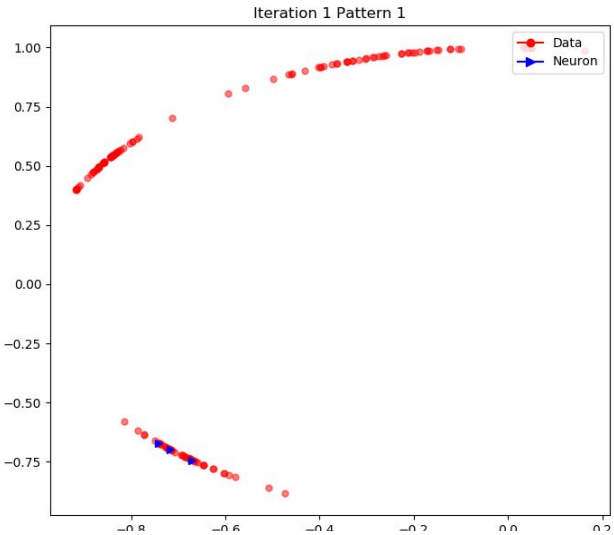

After 10 Iterations
Data
Neuron
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−0.8
−0.6
−0.4
−0.2
0.0
0.2

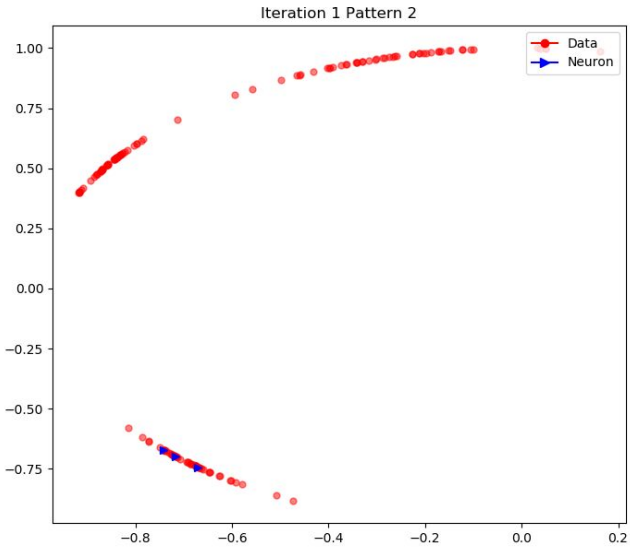
3 Neurons:

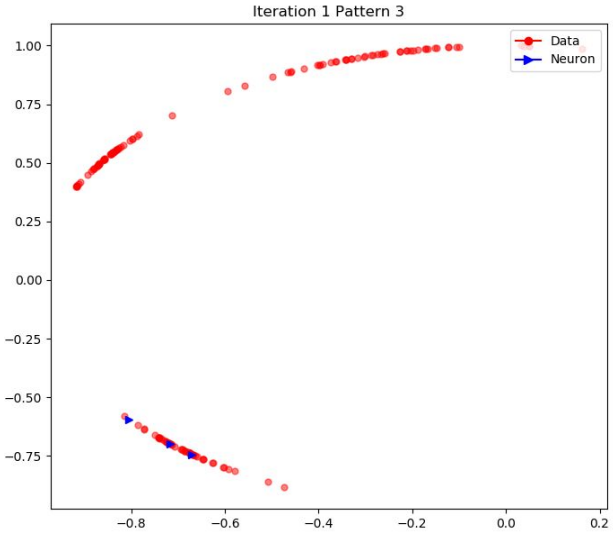
1. Random weights chosen
2. Number of iterations: 10
3. Leaning Constant = .1
4. Speed of convergence: Depended on initialization of neurons. However the average convergence was in < 5 iterations.
5. Other Lessons:
   - 5.1. The initialization proved to be especially important with the initialization of 3 neurons. As shown in the graphs below, if the neurons were initialized to the bottom cluster, the 3 major clusters wouldn't get recognized. This is because the two neurons at the bottom cluster would never "win" once one of the neurons move.
   - 5.2. When the weights were initialized conveniently, all three clusters were recognized.
   - 5.3. Would find 3 clusters, but not always accurate in finding the "most ideal" clusters.

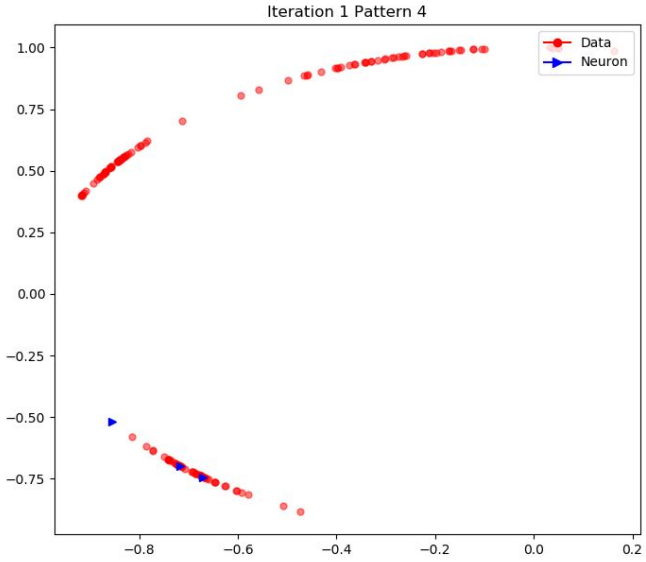
The "Worse" Neuron Initialization:

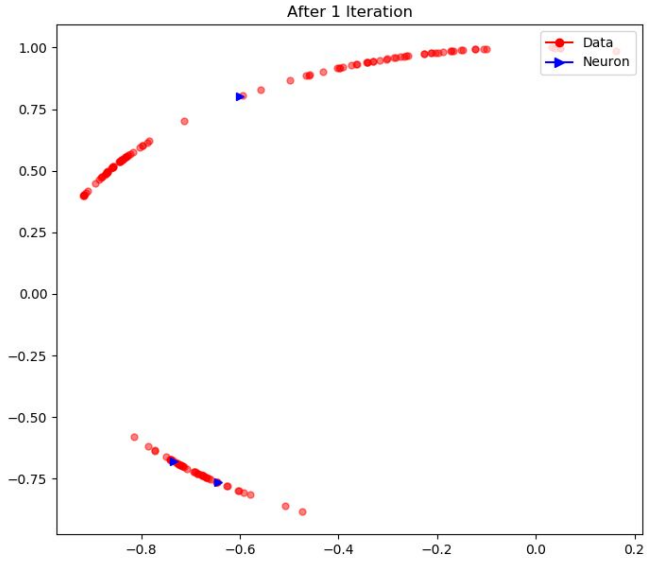


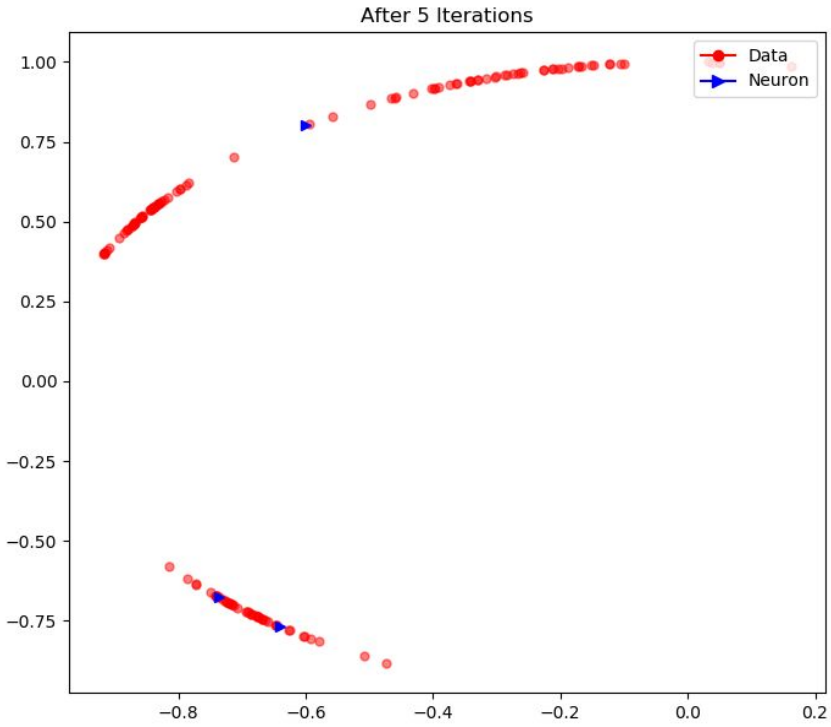




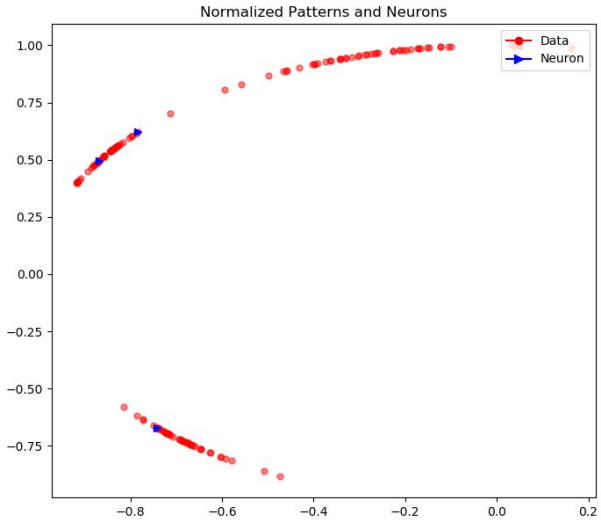


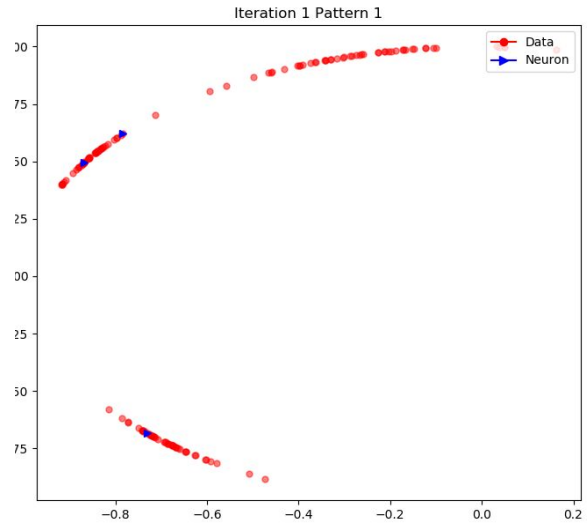

Iteration 1 Pattern 4
Data
Neuron
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−0.8
−0.6
−0.4
−0.2
0.0
0.2

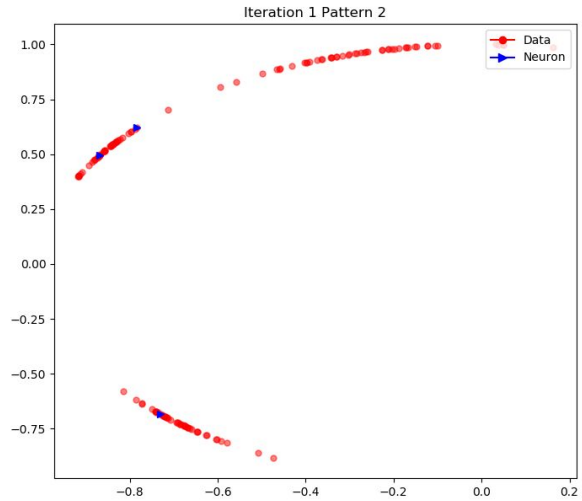

After 1 Iteration
Data
Neuron
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−0.8
−0.6
−0.4
−0.2
0.0
0.2

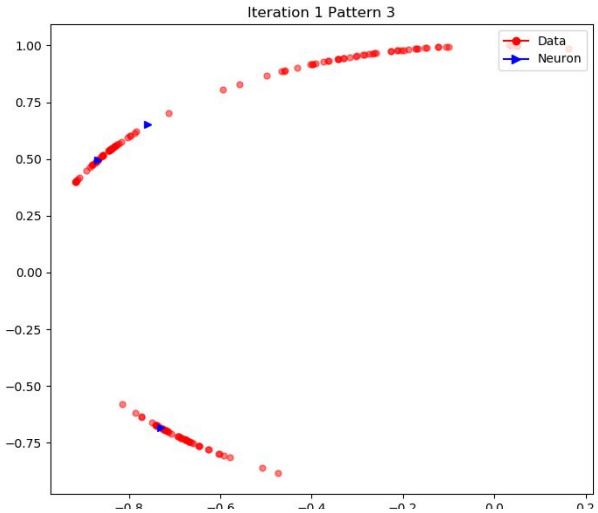

After 5 Iterations
Data
Neuron
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−0.8
−0.6
−0.4
−0.2
0.0
0.2

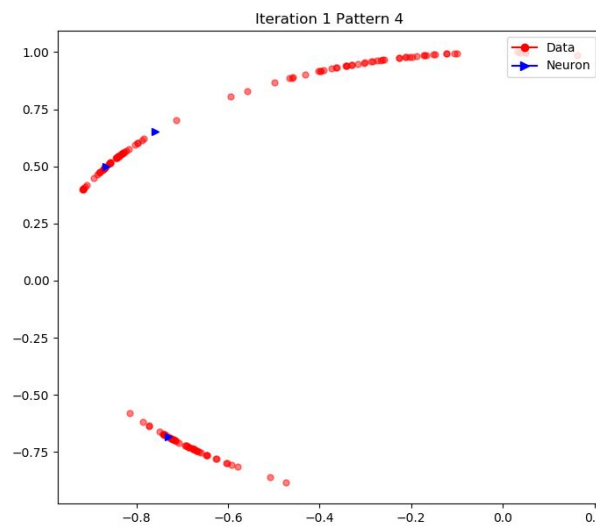
The “Better” 3 Neuron Initialization:

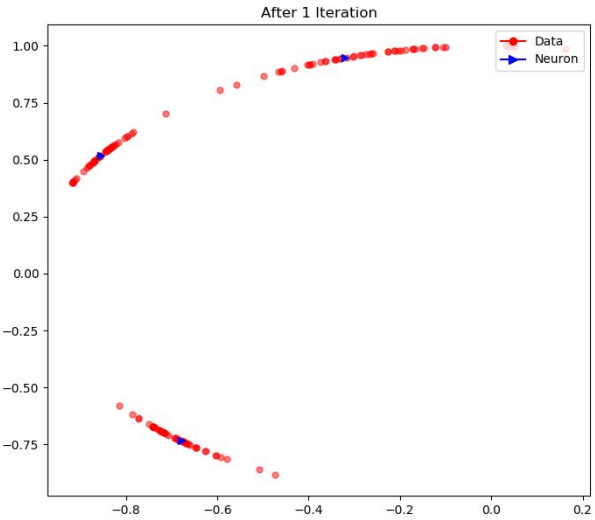


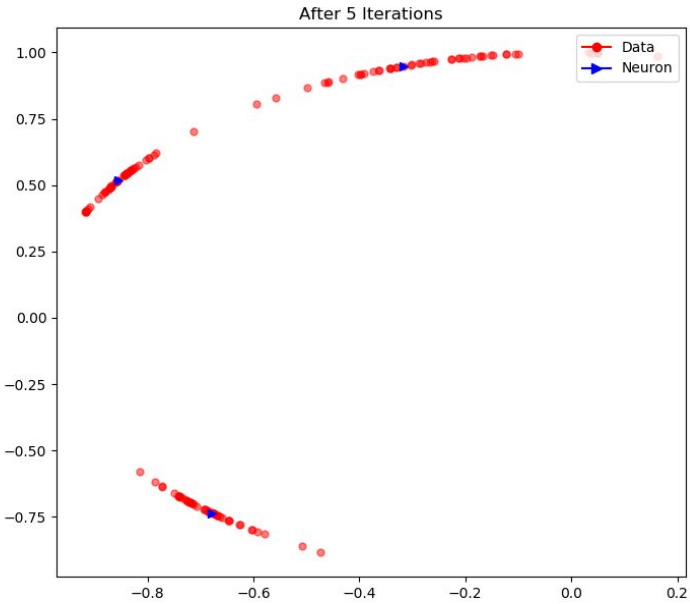




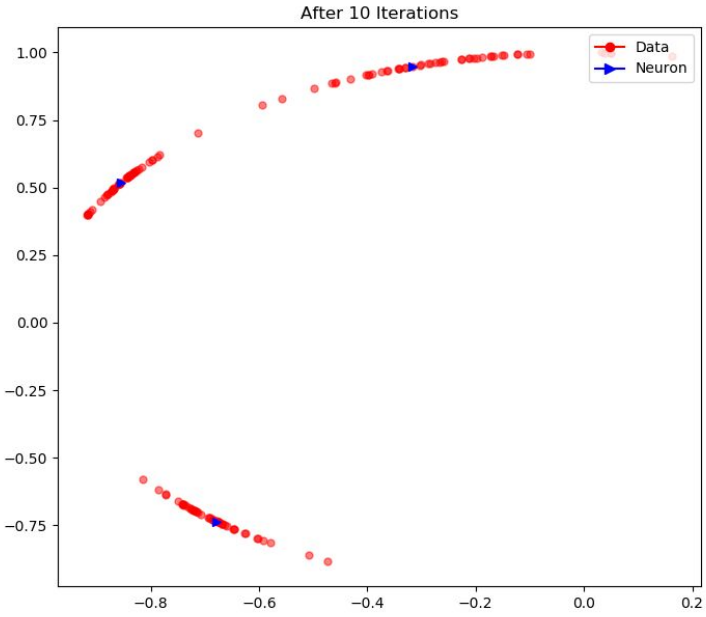


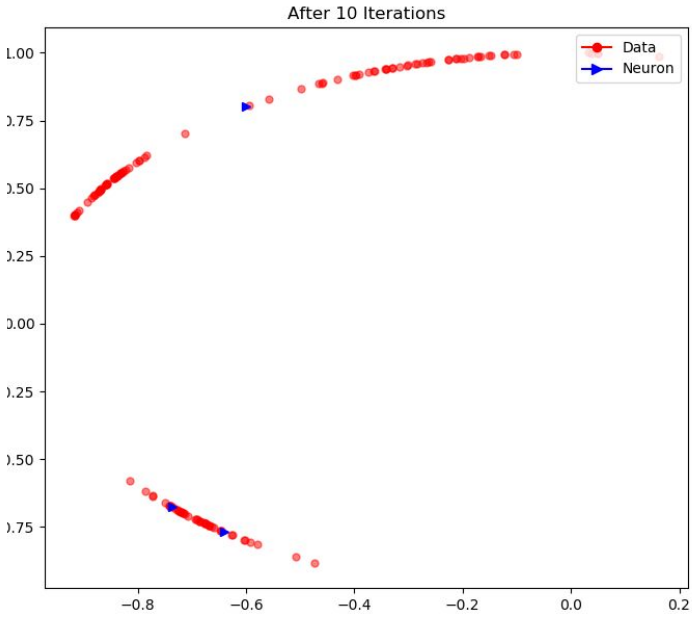






“Better” Neuron After 10 Iterations (Left) vs “Worse” Neuron After 10 Iterations(Right):

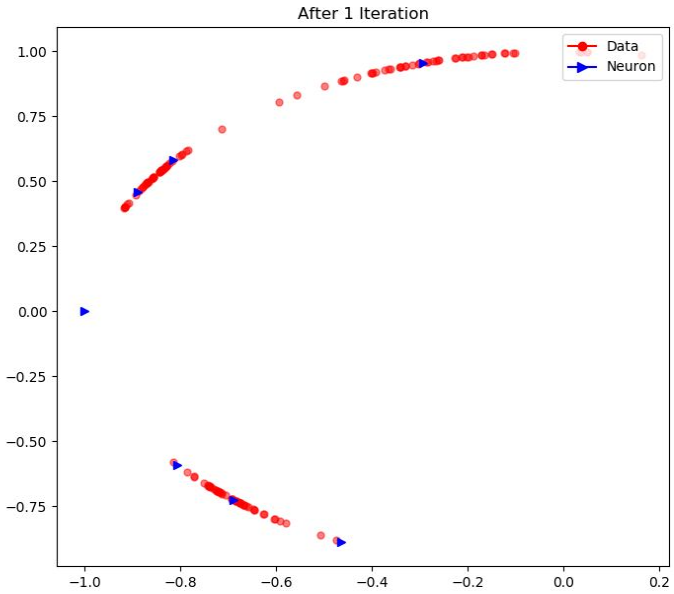


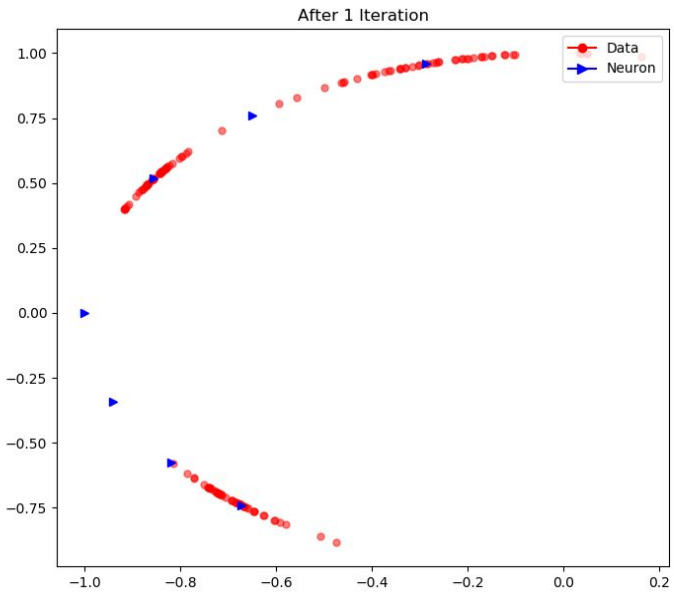


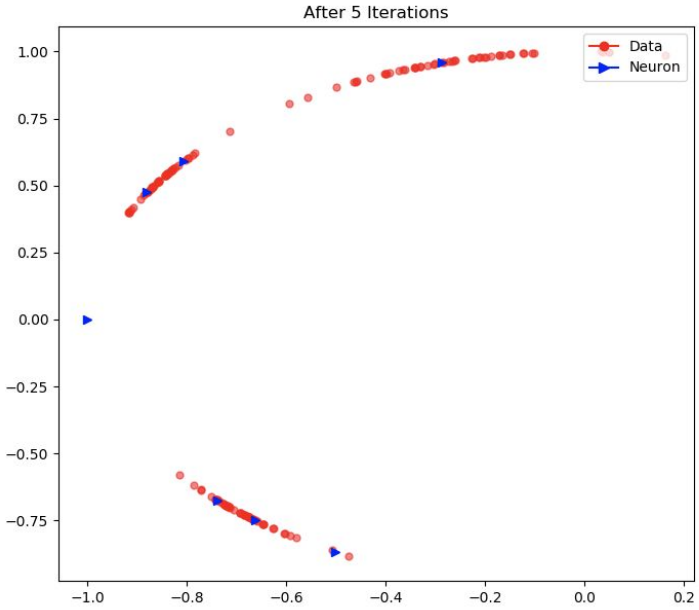
7 Neurons:

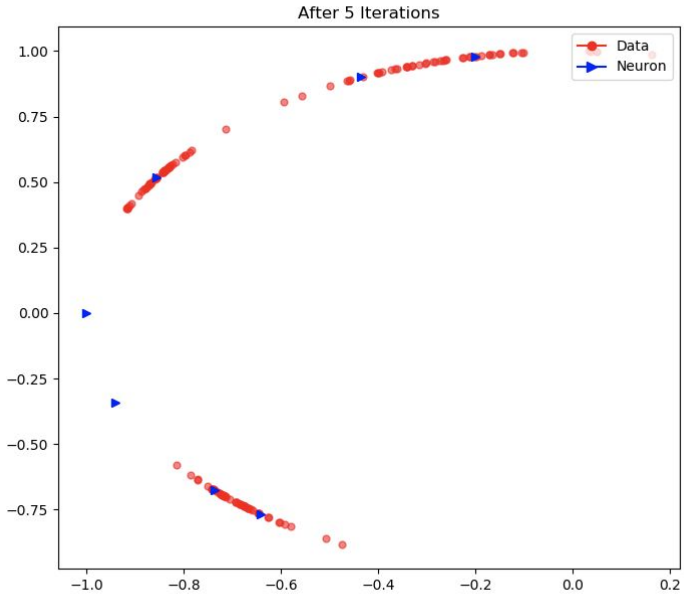
1. Weights manually-randomly chosen
   1.1. Initialized the weights within the range of the data values
2. Number of iterations: 10
3. Leaning Constant = .1
4. Speed of convergence: Depended on initialization of neurons. However the average convergence was in < 5 iterations.
5. Other Lessons:
   5.1. On first iteration, completely random values were used (between 0 and 1). However that would result in only 3 winners/learning neurons. See “random between 1 and zero” figure below.
   5.2. Manually choosing values between the possible range of values typically yielded better results. The results again, were still affected by initialization though.
   5.3. Would find between 5 and 7 clusters depending on initialization (manual random)

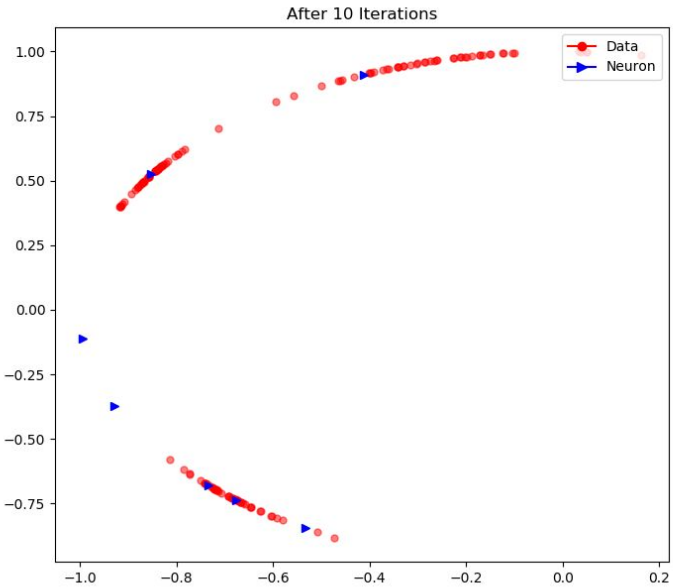
Two Manual-Randomly selected 7 Neuron Networks:

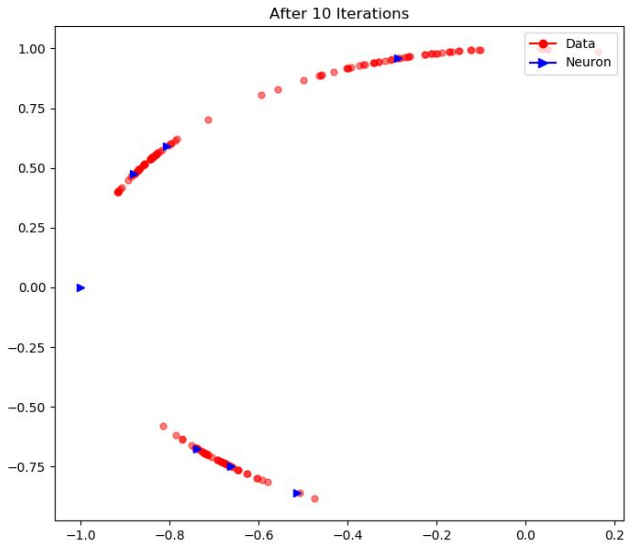




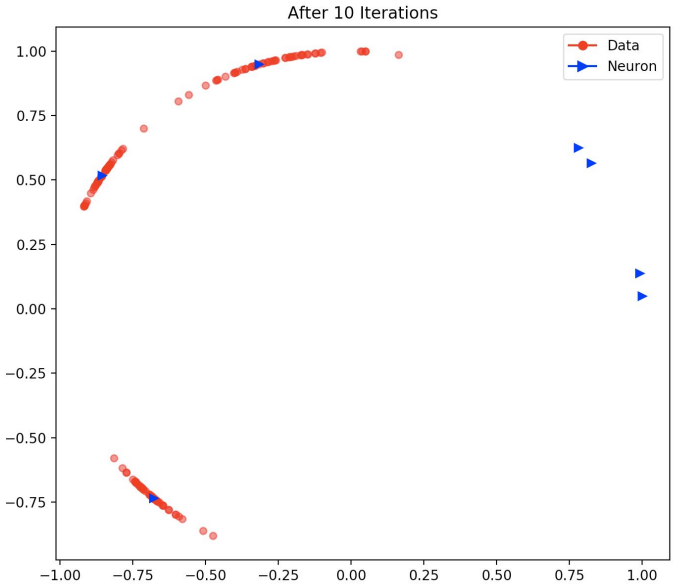

After 5 Iterations
Data
Neuron
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
0.2


After 5 Iterations
Data
Neuron
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
0.2


After 10 Iterations
Data
Neuron
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
0.2


After 10 Iterations
Data
Neuron
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
0.2

Random Between 1 and 0:



Final Thoughts:

The Kohenen network is greatly influenced by the initial conditions of the neurons. If neurons are far enough from data, some neurons will take over and never allow for the others to win. Another influencer of the network is the learning constant. Larger constants cause the neurons to converge much faster.

Note:

The source code is set to work on a seven neuron network architecture with learning rate at .1