Brian Matejevich

Problem 1:

Solve a 10-city TSP problem

The below results are after using the hyperparameters learning rate = .8, pivot = 10, sigma = 6, and epochs = 1100,1200,1250,2000 (initial, fair, good, best)

A close up of a map

Description automatically generatedA close up of a map

Description automatically generatedA close up of a map

Description automatically generatedA close up of a map

Description automatically generated

The report should include details on :

● What you learned through this assignment.

When training iterations are lower some points may be completely lost/ignored, see plot 1 (1,100 epochs)

● Implementation of your Self Organizing Map.

See code

● Your results for each question and their analysis.

As the training iterations increase the learned model become better and better by reaching the goal location in the straightest and closest pattern.

Problem 2:

Project the training data onto a 2D discrete space of size n x n using a Kohonen’s SOM.

High density is white and low density is black

A screenshot of a cell phone

Description automatically generated

Label each cluster (Kohonen Neuron) with the wine class.

A screenshot of a cell phone

Description automatically generated

Teal = Class 2, Purple = Class 1 & Yellow = Class 3

4. Comment on how well each class is separated.

In my opinion the SOM did a great job of classifying 13 dimensional data in a 2D plane. There are a few ‘misplaced’ nodes however with only 10,000 epochs it seems to have done a strong job. Note: 10,000 epochs, LR = .7, pivot = 0, sigma = 5, n = 15

I learned that SOM simply project N-dimensional data onto a 2D plane for easy classification.

Problem 3:

Base: N = 6990 LR = .9

A screenshot of a cell phone

Description automatically generated

Changing N: LR = .9 N = 3495 and N = 1398

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Changing LR: N = 6990 LR = .4 and LR = 1.2

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Allowing N to be high initially and gradually shrink is important so that more and more nodes are included in the final map. Shrinking the N also allows for connections between nodes to be straight lines as you can see lowering N reduces the ‘sharpness’ of the plot. Increasing the LR finds a more efficient path and lowering it finds a less efficient path.