Computational Intelligence
Self Organising Maps

Athanasios Theocharis
2961

May 22, 2020

1 Problem Introduction

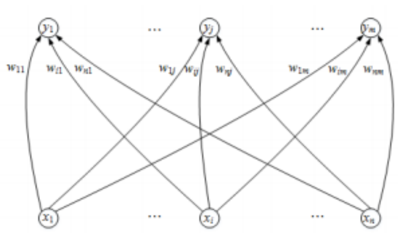
This project in Computational Intelligence aims to visualise and perform clustering on two data sets, in a clear manner, by utilizing Self Organising Maps. The first data set is the infamous IRIS data set[1], where 150 rows containing information on 3 different flowers are given. The second one is the Breast Cancer Wisconsin(Diagnostic)[1] with 569 rows of 32 attributes each.

2 Neural Network - Self Organising Maps

Self Organising Maps are artificial neural networks developed by Teuvo Kohonen. An unsupervised learning algorithm is employed with the main goal being the visualisation of high-dimensional data and dimensionality reducing.

2.1 Learning Algorithm

The Self Organising Maps achieve their goal by using competitive and collaborative learning. The main idea behind the algorithm is that each input vector changes a node (chosen using the minimum euclidean distance as the sole criterion) of a 2D-map and its neighborhood. After the training is complete, a topological map with clusters of the various classes is created.



A basic layout of the map is seen, with every input vector being connected to every node of the map.
Steps:

1. Create a 2D map containing randomly initialized weight vectors
2. Randomize the order of the rows of the data set
3. Repeat for every row in the data set

   Find vector in the map with the *minimum* Euclidean Distance between the vector and the row

   Update that weight vector as well as the neighborhood vectors according to a preset formula
4. Go to step 2 (for a set number of times/epochs)

3 Results

DISCLAIMER: The code in the SOM.py file was written from scratch. It was written in Python, and some libraries were utilized for easier data set loading and data handling. No external libraries were used that aided in the implementation of the SOM algorithm.

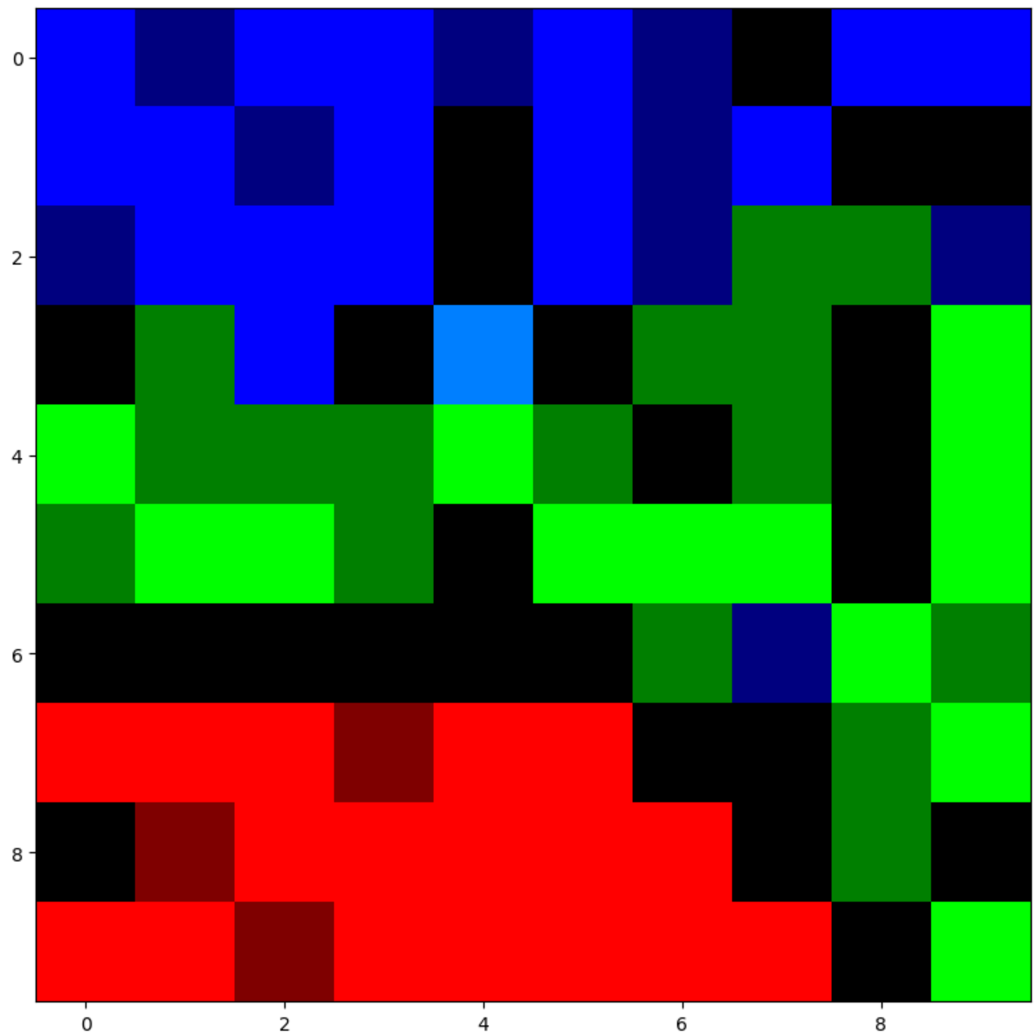
The various parameters that were changed during multiple trials were:

- Number of epochs
- Size of the map
- Neighborhood function
- Learning rate

For both data sets the initial parameters were:

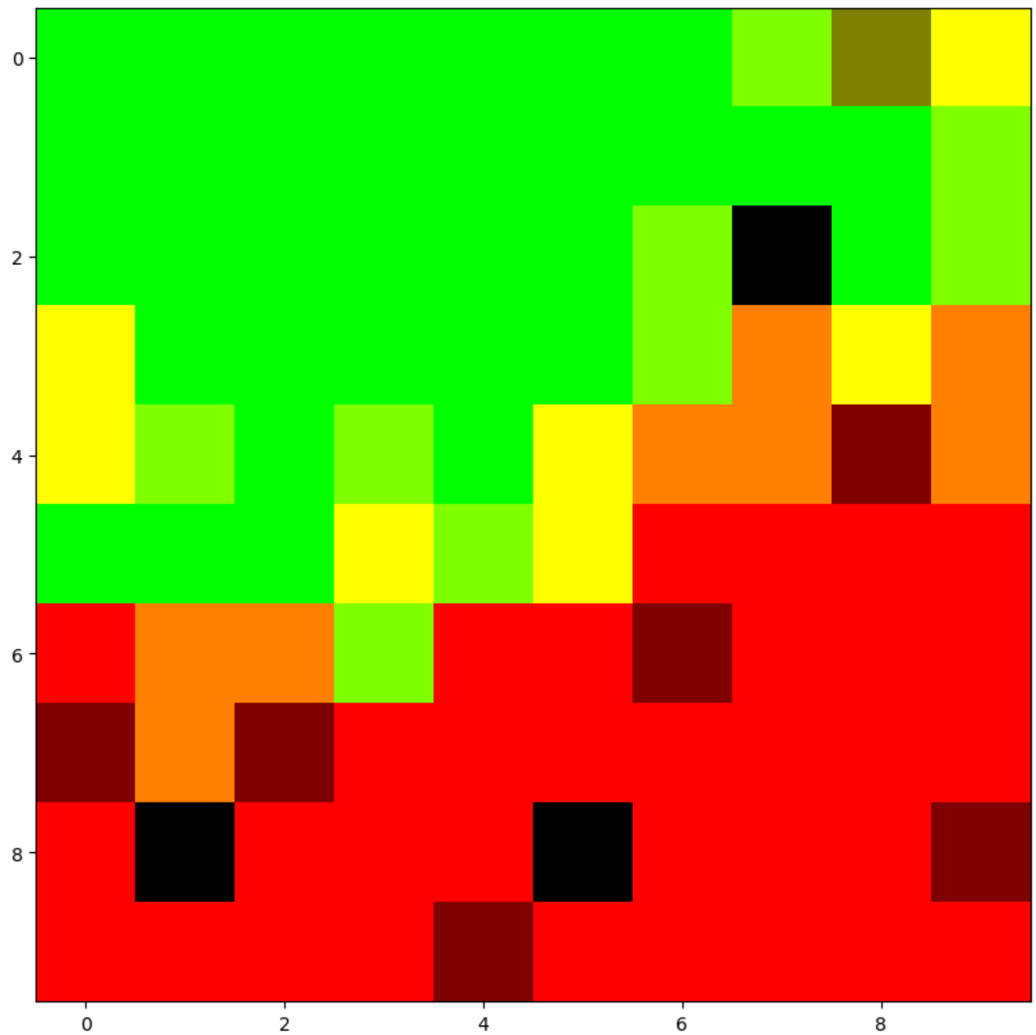
- Number of epochs: 500
- Size of the map: 10x10
- Neighborhood function: 1
- Radius function: $SideOfMap/2 = 5$
  and the updated radius on each iteration is calculated by $radius * e^{-\frac{curr\_epoch}{epochs}}$
- Learning rate: 0.25
  and the updated learning rated is calculated by $learning_rate * (1 - \frac{curr\_epoch}{epochs})$

3.1 IRIS Data Set



In the above image the class differentiation can be easily observed. There are 3 distinct classes, one for each flower species.

3.2 Breast Cancer Data Set



Here the two classes are also observed, whether the cancer is benign or malignant.

4 How to run the code

The *SOM.py* file can be run in any terminal, obviously. The prerequisites include having Python installed as well as some packages (such pandas). The user can choose between the two data sets. Of course, a small deviation from the results produced here is expected, since the random initialization method is used.

References

[1] UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science.