Distance Based Neural Network- 2, **Project : 5**

Atsushi oba (s1230128), Hitoshi Ikuta (m5221106)
Chowdhury Md Intisar (m5212106), Yunosuke Teshima (m5221151)

May 17, 2018

1 Kohonen Network and Vector Quantization

A kohonen Network is a self organizing network which encodes a large set of input vectors by finding a smaller set of representatives or prototypes which provides a good approximation to the original input space. Thus if we have an input space of suppose 3-dimension we can encode and project the 3 dimensional space to 2-dimension space. Such network can be used for dimensionality reduction or data compression. This principle well known as the vector quantization which is depicted as follows in the equation

$$D = \sum_x \left\| x - w_{I(x)} \right\|^2 \tag{1}$$

Thus we can see, the difference between the input vector x and the prototype vector or neuron $w_{I(x)}$ is the error. The target is to minimize the error with gradient descent. Miniminzing the error will help to converge to a low dimensional representation of the input space, unless it does not get trapped in the local minima.

2 Self Organizing Feature Map

The SOFM is an extension of Kohonen Network. It is also a topology preserving network which aims in dimension reduction and data compression. Although it is an extension of Kohonen network, there is slight change in the equation which takes into account neighborhood function during the training phase. A mathematical explanation is as follows.

Let us assume that we have input vector x and weight vector w. Unlike other nearest neighbor method we will update our weight vector taking into account neighbour function as in equation 2.

$$w_i = w_i + \eta\phi(i,k)(\xi - w_i) \tag{2}$$

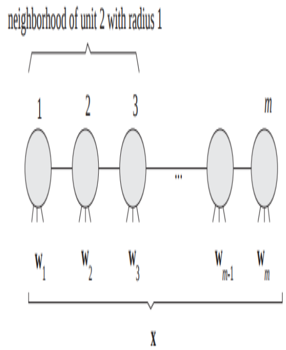
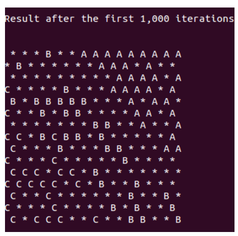
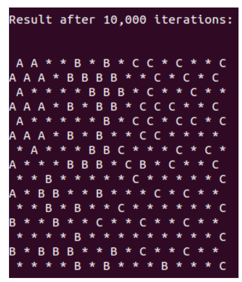


Figure 1: Learning of the prototype



(a) Output after 1000 iteration



(b) Output after 10,000 iteration

Usually the neighborhoods function can be of any shape. In this context let us assume the function is a radius of length ()̊Ṫhen all the unit within neighborhood of **winner** neuron will be updated.

- As th e weight converges every iteration the neighborhood function $\phi(i, k)$ is shrinked.
- The learning rate η is also decreased after every iteration depending on the conve rgence of the weight neuron.

3 Program Output by the given Source Code and Analysis

The provided source code has been modified slightly for input dimension. The given source code was tested for the **Iris Data Set** and projected to a two dimentional space depicted in the figure 2. Both of the image are output of same dataset with different iternation number.

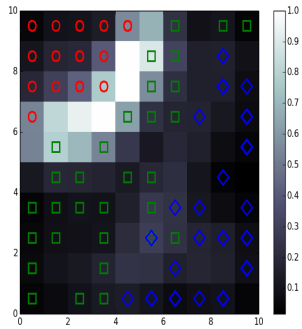


Figure 3: Iris data Visuilized in 2D Plan

We can observe from figure 1(a,b) that the program has projected the 3 class of Iris into there repsective clusteres in two dimentional space. Three of the class are denoted as **A,B and C**.

4 Visuilizing in 2D-plan with Python Visuilization tools

For a better visuilization of the input space and output space we have implemented the proposed problem in python with different dataset such as, **Iris Data**, **digits**. Since python is equipped with better visuilization tools we leveraged it and found some clear idea.

4.1 Iris Data Set

About 150 of the data was mapped in the 2d plane. The distance bar depicts how far is the mapped data away from the winner neuron. Also, the number of times winner neuron fires in shown in the figure 3.

4.2 Digits

The experiment with digits was done using 5 different digits. Each digit has 64 features, that is 64 dimensions. We have used the Mini self organizing map library and project these digits in 2 dimensional surface as depicted in figure

```
[[ 4.  3.  2.  2.  0.  0.  3.  3.  3.  3.]
 [ 0.  0.  1.  2.  2.  2.  0.  1.  3.  1.]
 [ 2.  0.  0.  1.  1.  0.  0.  5.  5.  3.]
 [ 1.  1.  2.  1.  0.  1.  0.  3.  3.  4.]
 [ 1.  0.  0.  0.  1.  0.  2.  0.  0. 10.]
 [ 3.  0.  3.  4.  1.  0.  2.  2.  1.  0.]
 [ 3.  1.  1.  1.  0.  0.  1.  1.  1.  1.]
 [ 1.  0.  2.  1.  0.  0.  1.  0.  0.  0.]
 [ 4.  0.  2.  0.  1.  0.  0.  6.  2.  1.]
 [ 0.  4.  3.  1.  0.  6.  4.  1.  0.  2.]]
```

Figure 4: The number of times winner neuron fires for each input

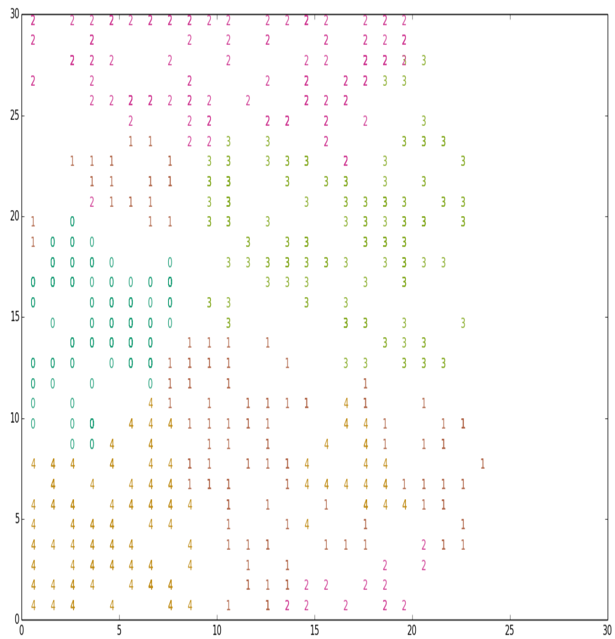


Figure 5: Digits datasets projected on 2D-plane using SOFM

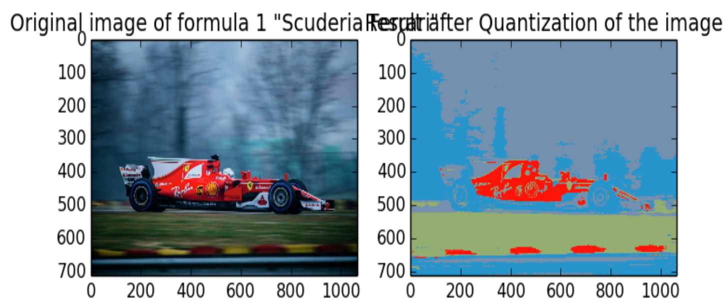


Figure 6: Original Image in the left, and quantized image in the right with the help Self Organizing Map

5 Experiment with Image to demonstrate Learning Vector Quantization

5.1 Learning Vector Quantization

The learning vector qualization is the supervised version of vector qualization. We applied this method for image quantization which alligns with the property of Kohonen network. An image is fed as input. As output we found compressed and clustered element of the image. Although the image is not clear but it is understandable.

We can see from the image that, the Self organizing network quantized the important detail of the image. It learns the similar colors and mapped them in to their respective places.

The source code for the program has been attached with the file.