Assignment 3- Visualizing Data using SOMs.

Review literature for visualization clusters in data using SOMs and briefly describe three such methods (approximately half a page for each + any figures)

Introduction

Self-Organizing Maps (SOMs), also known as Kohonen maps, are a type of artificial neural network introduced by Teuvo Kohonen in the 1980s. SOMs can be used to cluster and visualize data. Just Like the K-means algorithm, SOMs group similar data points together. However, SOMs differ in that they map the data onto a low-dimensional grid, which is often two-dimensional but can have higher dimensions as well.

SOMs can be used to map high-dimensional data onto a grid, enabling visualization of the data in a reduced dimensional space. By assigning each input to a node in the grid, SOMs capture the topological structure of the input data, including clusters and relationships among data points. The advantage of SOMs is that they can preserve the underlying structure of the original data while reducing its dimensionality for visualization purposes.

SOM Basic Knowledge

To get an abstract idea about what a SOM is please refer to the figure taken from Stefanovič, Pavel & Kurasova, Olga. (2011) [2]

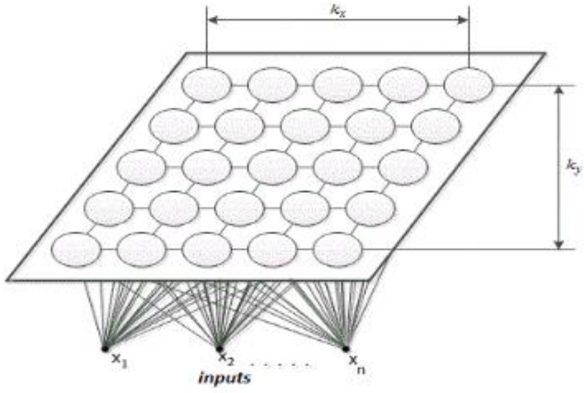

Fig. 1. Two-dimensional SOM (rectangular topology)

As depicted in the graph an SOM is a set of nodes, connected to one another via a rectangular or hexagonal topology [2]. A node is represented by a circle in the diagram.

The connections between input and nodes have weights such that, a set of weights corresponds to each node. We call the vector that is formed by these weights set as a neuron or a codebook vector (M_{ij}), the dimension of this codebook vector equals to number of inputs. Normally SOMs are referred to as Self Organized Neural Networks as well [2]

Next, we will see how SOM works. This is the algorithm for the SOM's internal working.

1. First, each node's weights are initialized.
2. Then a vector is chosen from the training data
3. Next, each node is examined to determine which node's weights are more like the chosen vector. The selected node is called the best matching unit (BMU)
4. Then the neighborhood of the BMU is calculated. The number of neighbors decreases over time.
5. Next, a reward is given to the winning node with becoming more like sample vector. Neighbors also become more like sample vector. If a node is closer to the BMU. It learns better and its weights get much altered. If a node is further from BMU, it leans much less.
6. Steps 2-5 are repeated for a specified number of iterations or until convergence is reached.

To find the BMU we simply calculate the distance from the weight vector to the sample vector. The unit with the smallest distance will be the BMU. Normally distance is measured using Euclidian distance.

Cluster Visualization Methods Using SOMs

Three methods will be discussed under this. They are,

1. U-Matrix
2. Component Planes
3. Vector Fields

1. U-Matrix

U-Matrix (unified distance matrix) can be defined as a representation of a Self-Organizing Map. It is used to visualize the distance between neurons. In other words, it is used to show relationships between neighboring neurons.

$$\textbf{u-matrix} = \begin{pmatrix} u_{11} & u_{11|22} & u_{12} & u_{12|13} & \cdots & u_{1k_y} \\ u_{11|21} & & u_{11|22} & & \cdots & u_{1k_y||2k_y} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{k_x1} & u_{k_x1||k_x2} & u_{k_x2} & u_{k_x2||k_x3} & \cdots & u_{k_xk_y} \end{pmatrix}.$$

Here, values $U_{ij\ |\ i(j+1)}$ and $U_{ij\ |\ (i+1)j}$ are the distance vectors Between the neighboring neurons M_{ij} and $M_{i(j+1)}$ and M_{ij} and $M_{(i+1)j}$ respectively. The values of elements U_{ij} can be the average of neighboring elements of the u-matrix. if the number of the neighbors is smaller, the average will be computed with a smaller number of elements.

Thus, we calculate the distance between adjacent neurons and present the findings with different colors between adjacent nodes. If the color is dark between the neurons, that means the distance is large and thus shows that there is a gap between the codebook values in the input space. Inversely light coloring depicts that the codebook vectors are close. Therefore, we can think of light areas as clusters and dark areas as cluster separators.

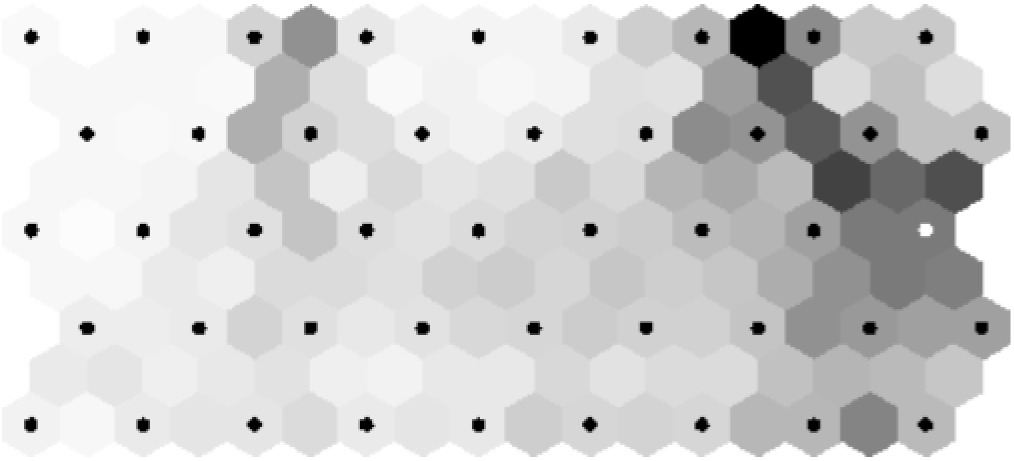

Figure 2.8: U-matrix representation of the Self-Organizing Map

For example, look at the above figure, according to what we discussed earlier we can get see the signs of a separate cluster in the upper right corner. Please note that U-matrix visualization does not have to be grey-scale. Other color schemes can be applied.

2. Component Planes

Component planes can be defined as plots that show the value of the weight vectors of the SOM in each dimension. The component planes are very helpful in analyzing the effect of a certain variable on the output [3].

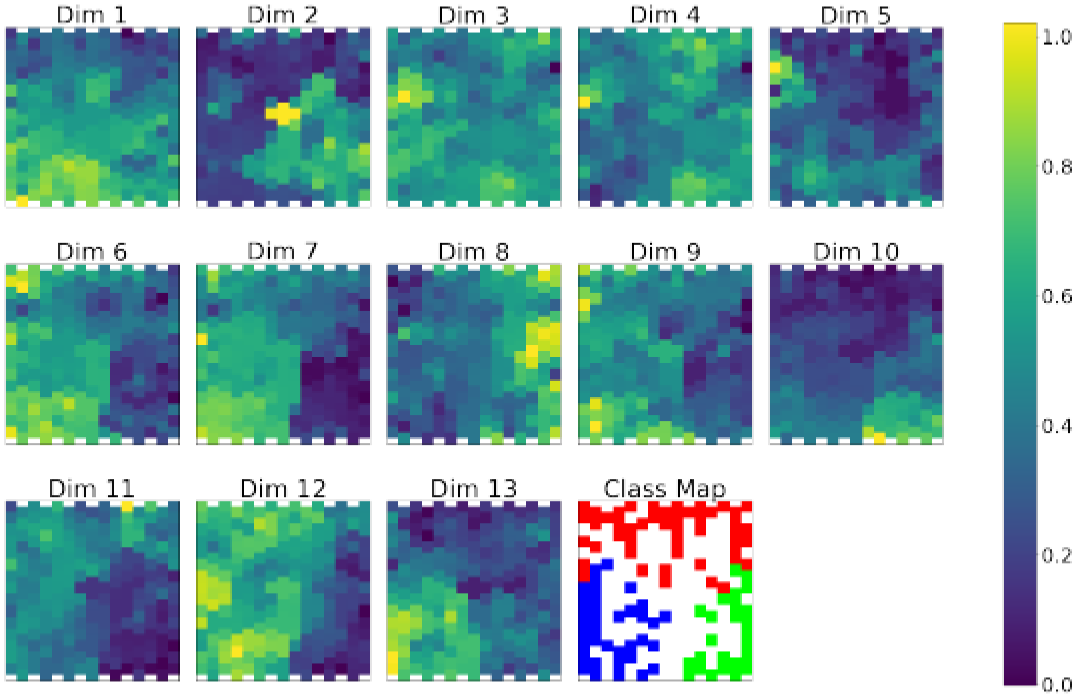

Figure 14: Component Planes and Class Map for the 13-D Wine Dataset of a 20 × 20 SOM

To understand how to interpret component planes, let's take a look at the above component planes and class map for the 13-D Wine dataset of a 20x20 SOM. We can use the control plane plots to get an idea of what features contribute to what classes. As an example, here if we compare dimension 13 with the Class map, we can clearly see that the blue class region has higher values in dimension 13. So, dimension 13 is a very good indicator in the case of the blue class.

If we think deeply, we can understand the effort we would have to make to do a visual inspection on a single component plane and make assumptions about data in some cases. This will be much harder when we try to correlate a trend from two features to a response variable. A solution for this might be using different color channels to represent different variables. We call such a plot as a 2D component plane [3].

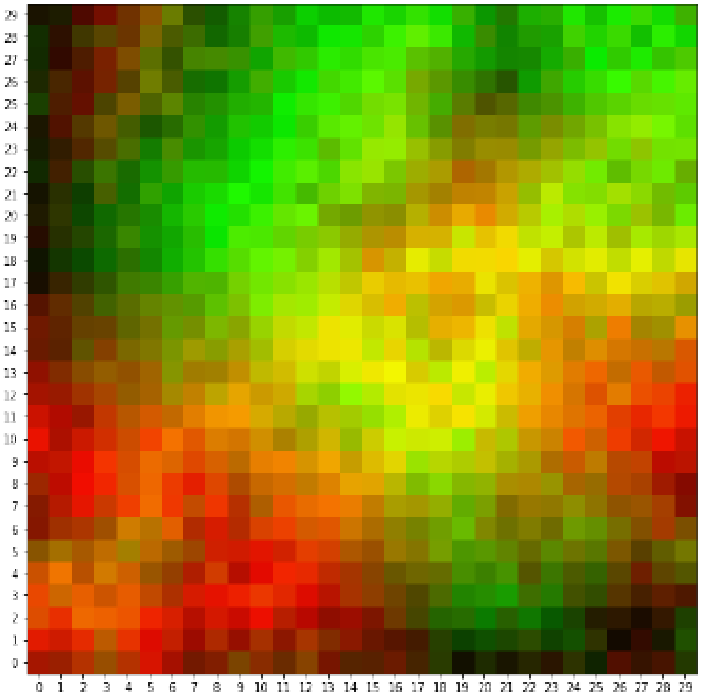

Figure 16: 2-D Feature Map of the ET-5000 dataset for the first two input dimensions.

Given above is the 2-D plane for the first two input dimensions of the ET-5000 dataset. Now we can see that we have represented two components using one feature map unlike the scenario above. This vector field visualization method exploits the neighborhood kernel concept to achieve the visualization objective.

3. Vector Fields

Both U-Matrix and Component plane we discussed earlier take only prototype vectors, not data vectors into account. Apart from those visualizations, there is another form of visualization that is based on vector field plots.

vector fields can be used with SOM to visualize the clusters in the data. This is done by creating a vector for each node in the SOM grid. The direction of the vector is taken by the weight vector of the node, and the magnitude of the vector is taken by the strength of the node's connections to its neighbors.

The vector field allows users to see the patterns and trends in the data and to identify clusters. Also, it is worth noting that vector fields can be visualized using various techniques such as arrows, streamlines, and color coding.

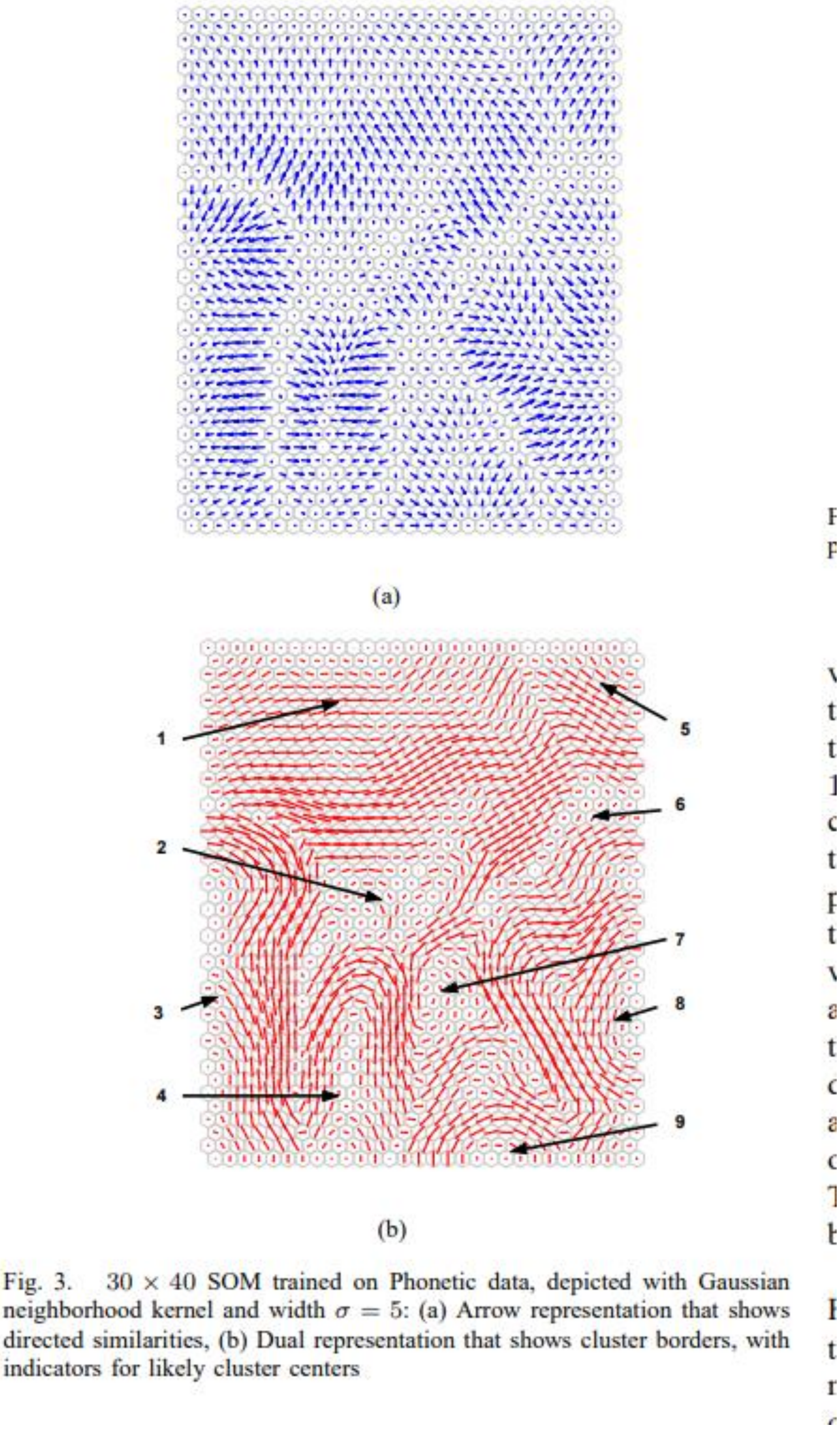

Fig. 3. 30 × 40 SOM trained on Phonetic data, depicted with Gaussian neighborhood kernel and width $\sigma = 5$: (a) Arrow representation that shows directed similarities, (b) Dual representation that shows cluster borders, with indicators for likely cluster centers

This figure is an example of vector field [4]

Pick one of the methods you reviewed in question 1) and use it to visualize the clusters present in the Iris dataset attached herewith (ignore the last column of the dataset as it is the class label). Note that you will have to implement a SOM and show a visualization(figure) for the data.

The selected method was U-Matrix visualization.

Please find the code that I have implemented using this GitHub link. If there is any issue regarding the link kindly notify

Code URL: https://github.com/adheeshagamage/NN-Assignment-3/blob/main/Assignment%203/239316C_SOM.ipynb

Assignment Folder location: https://github.com/adheeshagamage/NN-Assignment-3/tree/main/Assignment%203

These are the results obtained with Iris dataset clustering using SOM with **U-matrix visualization** method. Please note that here I have used a 30x30 U-Matrix SOM

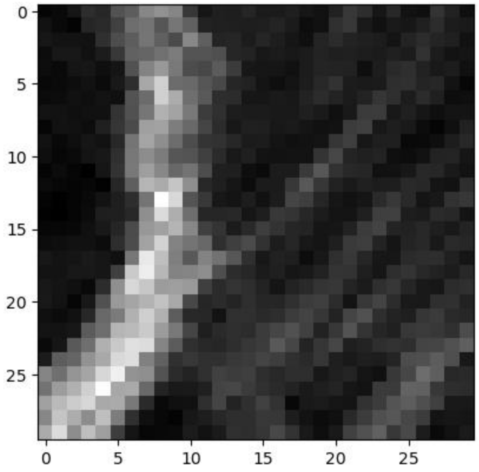

Figure1: U-matrix in grayscale

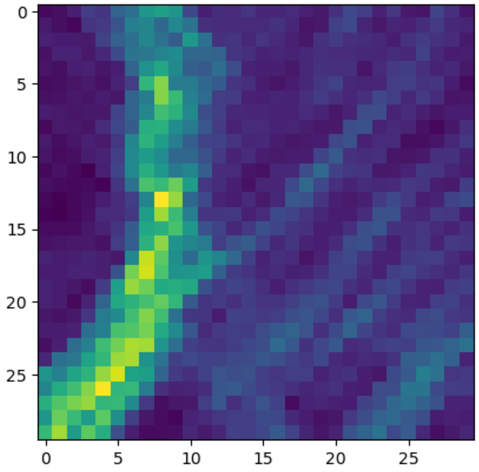

Figure 2: U-matrix in viridis

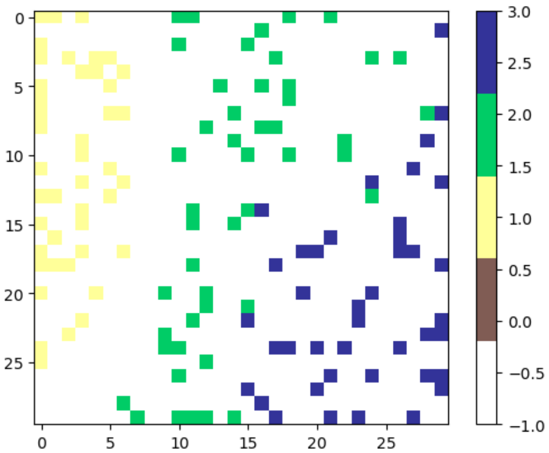

Figure3: Association of each data label with a map node.

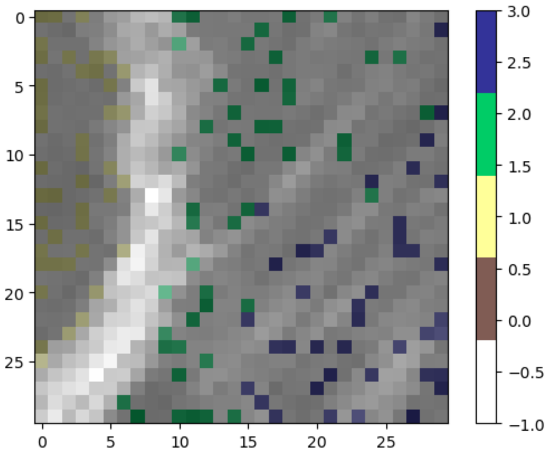

Figure4: Superimposition of U-matrix and Association of each data label with a map node.

When we look at grayscale U-Matrix, black cells indicate data that are similar to each other, and white cells indicate the borders between those groups of similar elements.

References

[1] T. Kohonen, "The self-organizing map," in *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464-1480, Sept. 1990, doi: 10.1109/5.58325.

[2] Stefanovič, Pavel & Kurasova, Olga. (2011). Visual analysis of self-organizing maps. Nonlinear Analysis: Modelling and Control. 16. 10.15388/NA.16.4.14091.

[3] Ponmalai, Ravi, & Kamath, Chandrika. Self-Organizing Maps and Their Applications to Data Analysis. United States. https://doi.org/10.2172/1566795

[4] Polzlbauer, G. & Dittenbach, Michael & Rauber, Andreas. (2005). A visualization technique for Self-Organizing Maps with vector fields to obtain the cluster structure at desired levels of detail. 3. 1558 - 1563 vol. 3. 10.1109/IJCNN.2005.1556110.

[5] Mohd Zin, Zalhan & Khalid, M & Mesbahi, Ehsan & Yusof, Rubiyah. (2012). Data Clustering and Topology Preservation Using 3D Visualization of Self Organizing Maps. Lecture Notes in Engineering and Computer Science. 2198.