Assignment 3: Self Organizing Maps

Francesco De Nunzio
12302615

Roberto Carminati
12302504

Poremba Pascal
11809911

ABSTRACT

The objective of this assignment is to perform data analysis using SOM, in order to learn how to use and interpret them along with their visualizations, combining various parameter settings.
Self-Organizing Maps (SOMs) are unsupervised learning models that map high-dimensional data onto a 2-dimensional grid. Useful for clustering and visualization, SOMs arrange similar data points closer to each other, identifying patterns and structures.

The main libraries used are MiniSom[1] and PySOMVis[2]. MiniSom is a minimalistic Python implementation of self-organizing maps designed to be lightweight, while PySOMVis is a library used to build useful visualizations.

The study emphasizes the significance of multiple visualizations to comprehend patterns in data. While individual visualizations provide insights, a comprehensive understanding is derived from diverse visual perspectives. The exploration of different parameter configurations reveals the relevance of SOM size, with emergent SOMs showing better performance. Also other parameters such as learning rate and neighborhood radius are crucial.

The importance of graphically visualizing parameter changes is highlighted, since other ways to find the better combination, such as with grid search, turns out to be disappointing.

In this scenario we have the labels of the classes, which is very helpful in some visualizations, and in the calculation of the final metrics as the confusion matrix. The final result will be surprisingly good.

[1] https://github.com/JustGlowing/minisom
[2] https://github.com/smnishko/PySOMVis

DATASET

Finding a dataset that met all the project requirements proved challenging, but we finally selected the Dry Bean Dataset[3]. This dataset originates from research which aimed at developing a computer vision system capable of distinguishing seven distinct registered varieties of dry beans, despite their closely aligned physical characteristics. Through high-resolution imaging, the system captured detailed features from 13,611 grains spanning these seven bean varieties.

The dataset comprises 18 distinct features extracted from the bean images, including geometric properties such as area, perimeter, major and minor axis lengths, aspect ratio, eccentricity, convex area, equivalent diameter, extent, solidity, roundness, compactness, and four shape factors. These features were used to create a classification model to identify and classify the seven varieties of dry beans: Seker, Barbunya, Bombay, Cali, Dermosan, Horoz, and Sira.

Our project's goal is to classify different types of dry beans using Self-Organizing Maps (SOM). The aim is to visually and analytically explore the behavior of these models through the parameter tuning process to achieve an optimal SOM for classification.

PREPROCESSING

The dataset contains only numerical values, and does not have missing values, so there is no need for imputation.

Taking a look at the variable correlations, the feature "ConvexArea'' is removed because a has a correlation of 1 with the variable"Area ".

When dealing with SOMs, it is very important to check the feature ranges because they’re distance based models. Investigating the distributions of variables using Boxplots (Figure 1), we decided to log-transform the variables "Perimeter," "Area,"

[3] https://www.kaggle.com/datasets/sansuthi/dry-bean-dataset

"EquivDiameter," "MajorAxisLenght" and "MinorAxisLenght", because they were really asymmetric with many outliers.

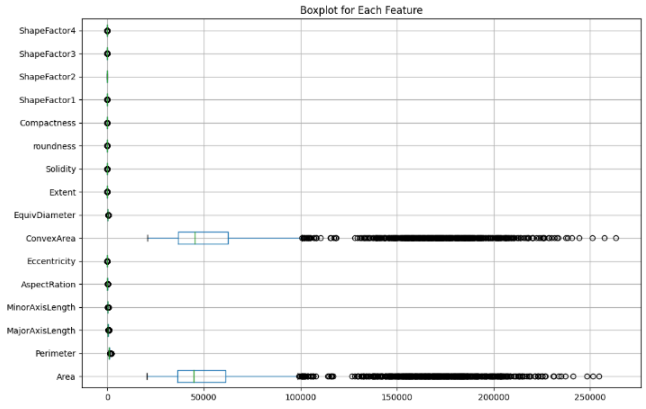


Figure 1: Boxplot of the variables before the log-transformation and Z scaling

Then, it is very important to scale the data, so z-score scaling has been applied, since it is not sensitive to outliers:

$x = (x - \mu)/\sigma$.

Where μ is the mean and σ is the variance of the variable.

At the end of this preprocessing, the variables are distributed as shown in Figure 2.

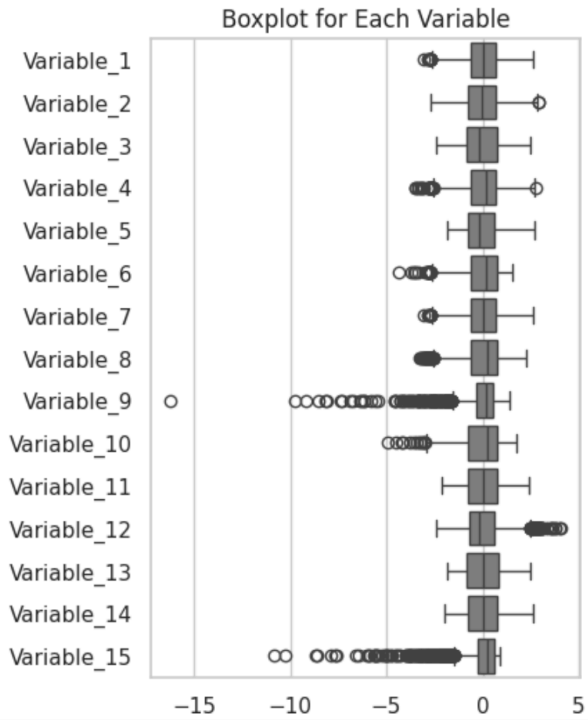
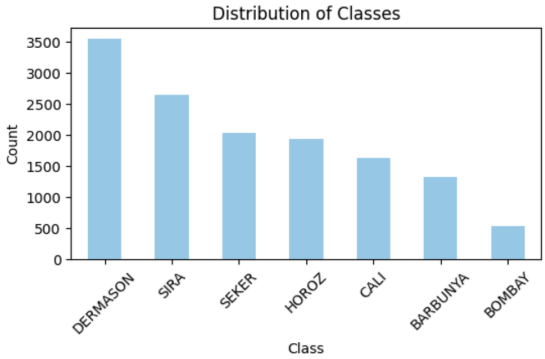


Figure 2: Boxplot of the variables after the log-transformation and Z scaling

After examining the frequencies of the class labels (Seker, Barbunya, Bombay, Cali, Dermosan, Horoz, and Sira), we decided to first focus on the four classes with the highest number of observations: Dermosan, Sira, Horoz, and Seker. Visualizing seven distinct classes in SOMs visualizations can be confusing, so by removing the less frequent classes, we aim to achieve clearer visualizations.



We've performed undersampling to balance the frequencies of the four classes, ensuring each type of bean now has 705 observations. While this approach may not be the most efficient for prediction purposes, it aligns with our primary goal of enhancing our analysis and understanding of cluster formations.

Later, the Optimal SOM will then be applied on the dataset containing all the classes. Undersampling will still be applied, having 500 observations of each of the 7 classes.

Finally, we rearranged our data to use MiniSom and the PySOMVis, transforming it into numpy arrays, and mapping the classes.

SOM TRAINING AND ANALYSIS

1. Train a reasonably sized „regular“ SOM

The dataset now contains 2820 datapoints, and a reasonable resolution for the SOM could be around 10. We decided then to build a 15x15 SOM with a total of 225 units, obtaining a resolution of ~12.
It is very interesting to visualize the class distribution in the SOM to have an idea of how clusters may be divided. It is possible to see the predominant classes in each unit (with interpolation) thanks to Chessboard visualization (Figure 4).

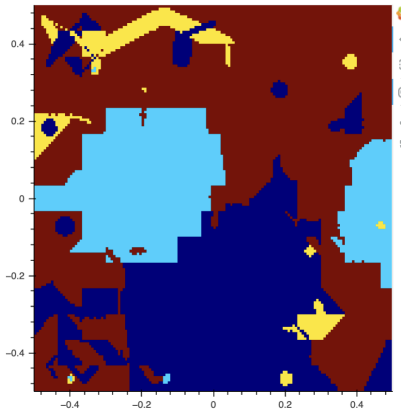


Figure 4: Chessboard 15x15 SOM with default parameters

As we can see there are areas with a predominance of the light blue class and dark blue class. Then there is the brown class which is all around the SOM, meaning that many beans with different characteristics are still belonging to the same class. Finally, the yellow class is not well defined in the SOM and its spread around the units.

It is very helpful to apply clustering algorithms to the SOM units to find groups of similar units, and compare them to the class distribution (Figure 5).

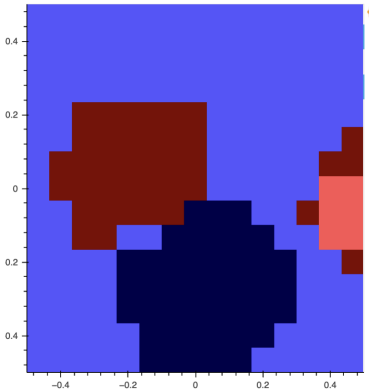


Figure 5: Cluster of Units (KMeans) 15x15 SOM with default values

Note that the colors unfortunately don't correspond to the same classes.
We can see that even clustering the units shows the same structure of the classes, highlighting a macro cluster in light blue, a cluster in dark red divided into 2 subclusters. Then the light red cluster on the right is a subcluster of the dark red, and finally a well defined dark blue cluster (bottom).

It is possible to notice the border effect of SOMs, which means that all the borders of the SOM pull the center toward itself, making the SOM able to “specialize” the unit weights better on the border.

It can be shown how data is distributed across the SOM thanks to Hit Histogram on the left and P-matrix on the right (Figure 6). Both plots show concentration of data in the corners, but P-matrix also uses a hypersphere to show how input data is mapped in the units.

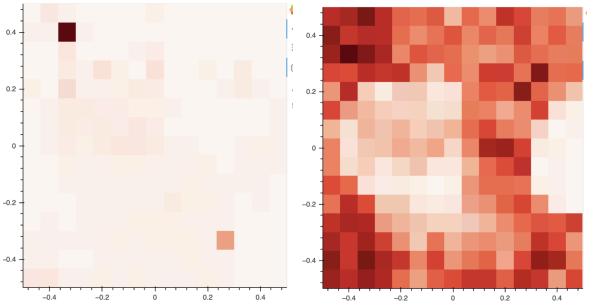


Figure 6: Hit histogram on the left and P-matrix on the right

It can be interesting visualizing the Quantization Error of the units, meaning the error between each datapoint and the weights of the BMU (figure 7). As we can see the error is equally distributed, other than couple cells with a higher error. The total value of quantization error is 1.28, which can be later compared to other errors of different SOMs.

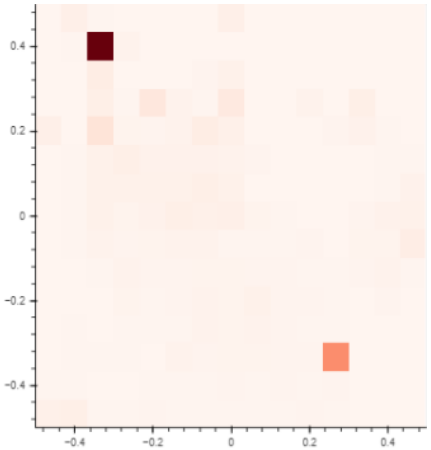


Figure 7: Quantization error of a 15x15 SOM with default values

Finally, the topographic error is a very interesting metric that gives information about how the SOM respects the topology. This value depends on the number of times the BMU is not a neighbor of the second BMU for a datapoint. Figure 8 shows the distribution of this measure across the SOM.

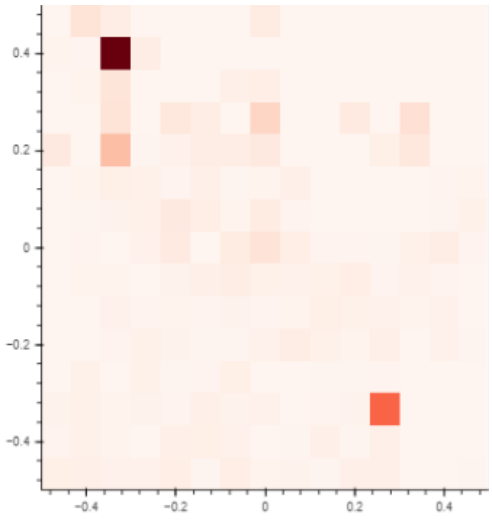
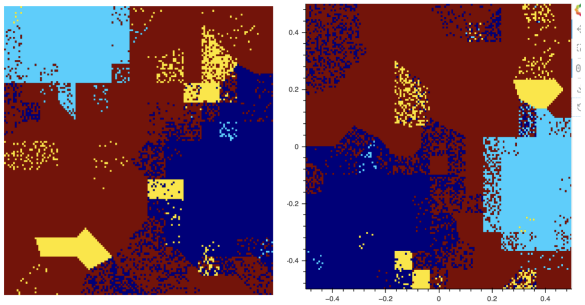


Figure 8: Topographic error of a 15x15 SOM with default values

As we see the topographic error is equally distributed, other than two specific units with a higher value. The topographic error is 0.42.
It's interesting to notice that the errors are distributed in the same way, and it's higher in the most dense areas of the map.

2. Analyze different initializations of the SOM

Later we explored the differences when using different types of initialization by changing the seeds. The most useful plots in finding similarities and differences are chessboard and clustering the units (Figure 9).



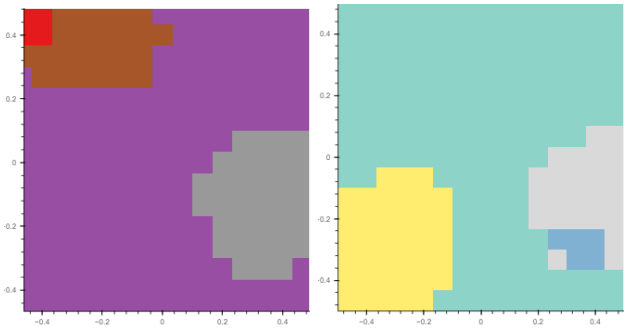


Figure 9: Chessboards and Cluster of Units with 2 different initializations of 15x15 SOM with default parameters

It is possible to see that the clusters mostly have the same shape, but they're shifted in the SOM. It is still evident the big cluster all around the map, and the subclusters on the borders. In addition, both SOMs show a cluster with a smaller subcluster inside.

Density visualizations, as Hit Histogram in fig 10, confirm that the clusters are shifted: for example in the chessboard left plot of fig 9 the top-left cluster (brown) is shifted (in the right chessboard visualization) on bottom right (grey), and it is possible to see the same shift in the data densities of figure 10.

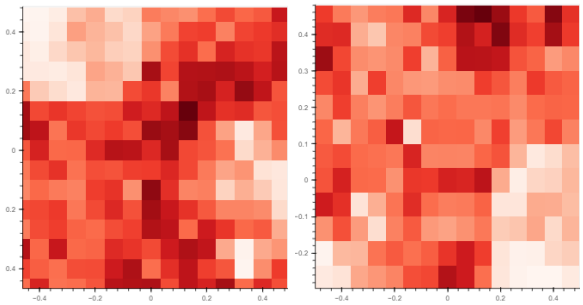


Figure 10 : Hit histograms of a 15x15 SOM with default values with 2 different initializations

3. Analyze different map sizes

Now 2 additional SOMs varying the size (very small / very large), and with a large neighborhood radius and high learning rate are trained.
The very large SOM is an emergent SOM, meaning that the number of units is higher than the number of datapoints. The dimension is 100x100 meaning 10000 units, while just 2880 data points (resolution of 3). The small SOM is instead a 7x7, half size of the regular SOM.
In the SOM previously built, the learning rate and the neighborhood radius had default values of 1 and 1, while now they are respectively 1.5 and 2.5

First the large som (100x100) is analyzed. An interesting plot is the P matrix, which shows for each unit the number of points in the input space in a predefined range. It is possible to see that two areas with most of the input points are top-right and bottom-left.

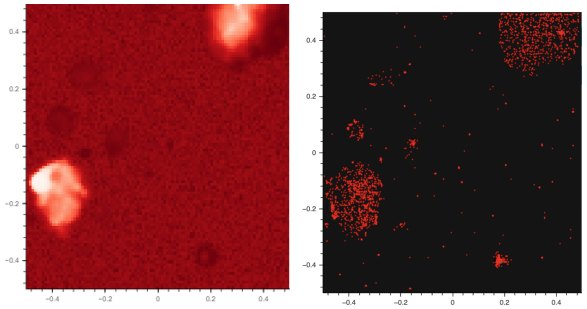


Figure 11: P-Matrix on the left and Sky Metaphor on the right of a 100x100 SOM with neighborhood radius of 2.5 and learning rate of 1.5

In addition it is more evident the datapoint distribution with Sky Metaphor visualization (figure 11 on the right).

In figure 12 classes distribution and cluster of units are shown.

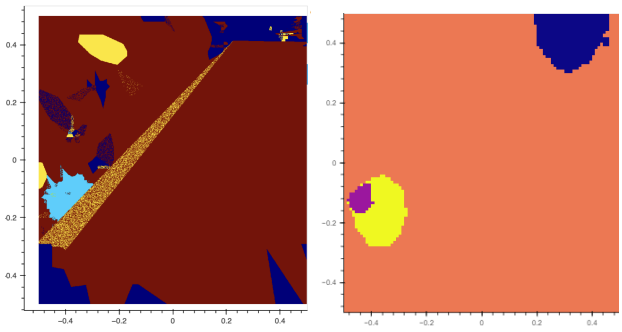


Figure 12: Chessboard and clustering of the units of a 100x100 SOM

Chessboard plot isn't very clear showing cluster boundaries, but it is possible to see a cluster top-right (dark blue), another one (light blue) on the left. There is then a big cluster all around the map, and the yellow class spreads between many different units. On the right of fig 12, thanks to clustering of units, it is easier to see what cluster boundaries should be. Unfortunately the colors don't match the same area, but it is evident a blue cluster top-right, and a yellow one with a subcluster on the left. Finally the rest of the area represents the last cluster.
Here it is evident how border effects let the SOM units specialize the most and represent the data better in the borders.

Quantization error has the value of 1.02, and is shown in figure 13. It is very low because of the many units, which allow the SOM to represent the data very well.

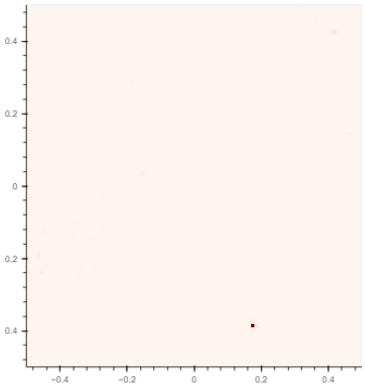


Figure 13:Quantization Error of a 100x100 SOM

Then, Topographic error is shown in figure 14.

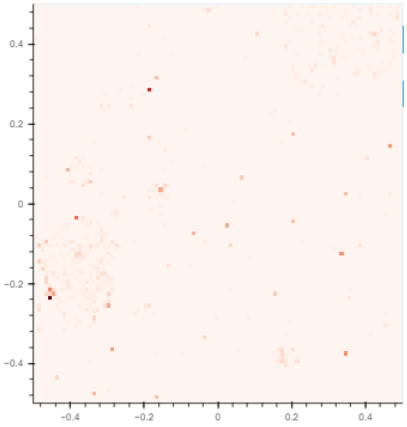


Figure 14:Topographic Error of a 100x100 SOM

Then the small SOM (7x7) is analyzed. In figure 15 Sky Metaphor visualization is plotted, which shows the input data is homogeneously mapped on the SOM.

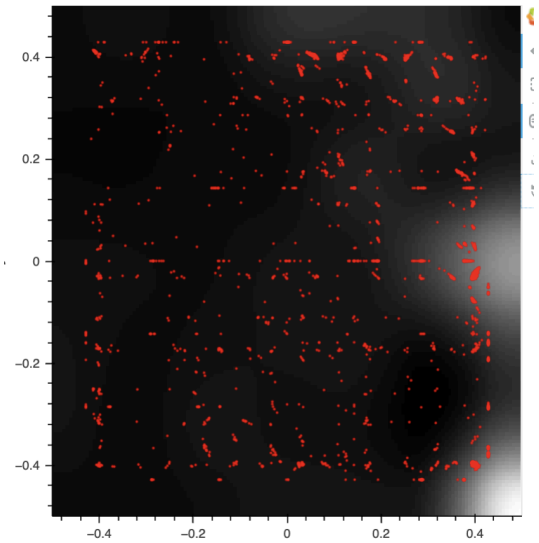


Figure 15: Sky Metaphor of the 7x7 SOM

Then in figure 16 the Chessboard and the Clustering of Units are shown.

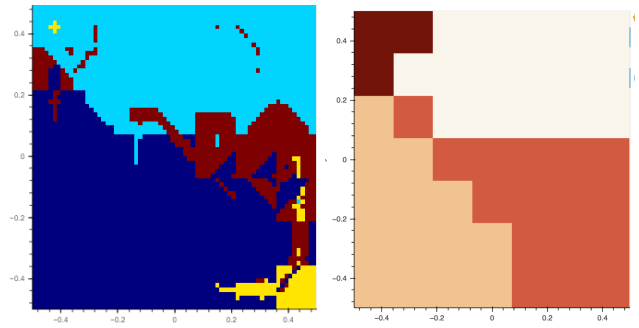


Figure 16: Chessboard and cluster of the units for a 7x7 SOM

Classes are not perfectly divided in the chessboard, but it is still possible to see a structure in the data. It is much easier to see clusters as suggested from Clustering on the right.

As expected the Quantization Error is higher than before (1.84) because of the far fewer units (figure 17).

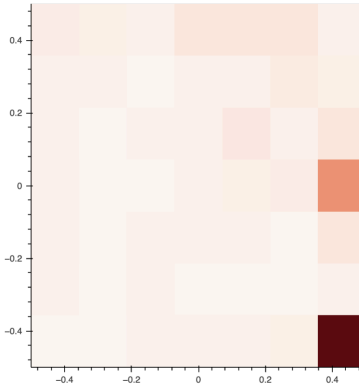


Figure 17:Quantization Error of a 7x7 SOM

Topographic error is shown in figure 18, and its value of 0.047 is as expected much lower because of the fewer units.

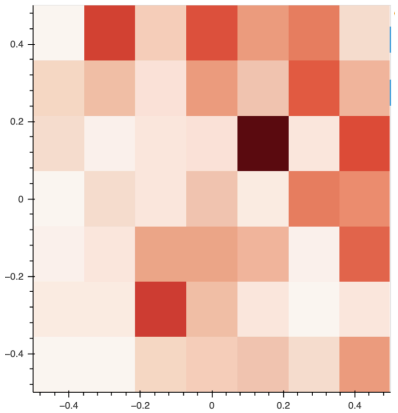


Figure 18:Topographic Error of a 7x7 SOM

4. Analyze different initial neighborhood radius settings

Now two SOMs are trained („regular“ / very large) with much too large / much too small neighborhood radius. Since the standard value is 1, the analysis is performed with 0.4 and 8, in order to have a much smaller, and much larger neighborhood radius.
It is important to specify that the neighborhood function used is Gaussian: it assigns weights to neighboring points based on their distance from a central point, with closer points receiving higher weights, following the bell-shaped curve characteristic of a Gaussian distribution.

First regular som (15x15) with small neighborhood radius (0.4) is analyzed. Chessboard visualization on the left and Clustering of Units in figure 19.

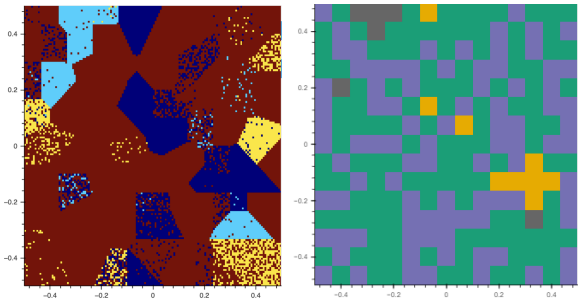


Figure 19: Chessboard and cluster of the units for a 15x15 SOM with a small neighborhood radius

There is a class that is mostly around the whole SOM, and the other classes are spread in small areas in different points of the SOM. There is no evident structure of the data.

Topographic Error and Quantization Error (fig 20) are similarly distributed, with a higher error on the same units.

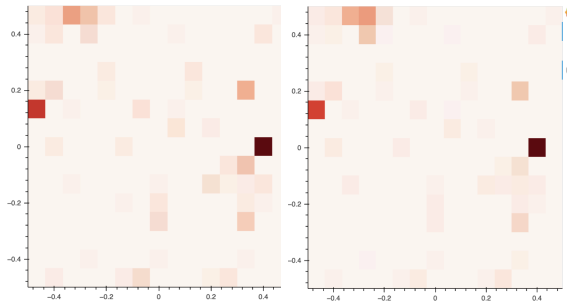


Figure 20: Quantization Error (left) and Topographic Error (right) for a 15x15 SOM with a small neighborhood radius

Comparing the Hit Histograms (Figure 21) and the errors (Figure 20), it is interesting to notice the error is higher where the area is more dense.

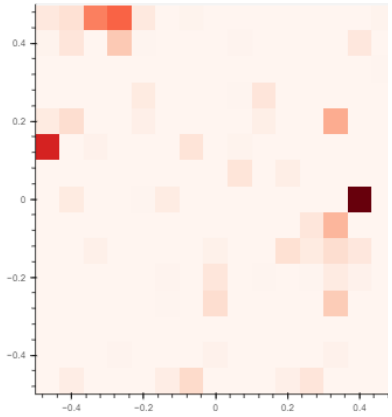


Figure 21: Hit histogram for a 15x15 SOM with a small neighborhood radius

Then regular SOM (15x15) is trained with a high neighborhood radius (8).
Chessboard visualization is shown on the left and Clusters of Units on the right of figure 22.

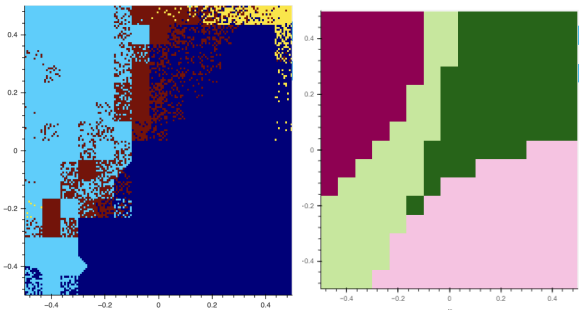


Figure 22: Chessboard and cluster of the units for a 15x15 SOM with a high neighborhood radius

Surprisingly a very large value of neighborhood radius shows very well defined clusters even on the regular SOM.

In figure 23 Quantization Error, which is equally distributed, and Topographic Error which is not homogenous.

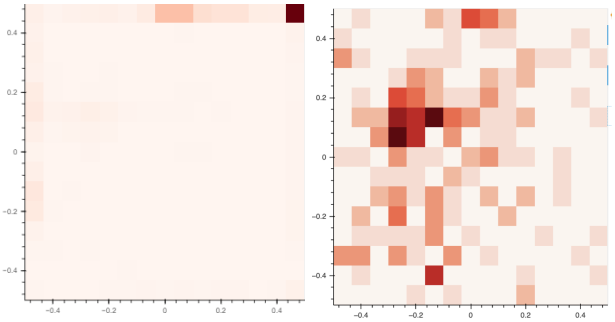


Figure 23: Quantization Error (left) and Topographic Error (right) for a 15x15 SOM with a high neighborhood radius

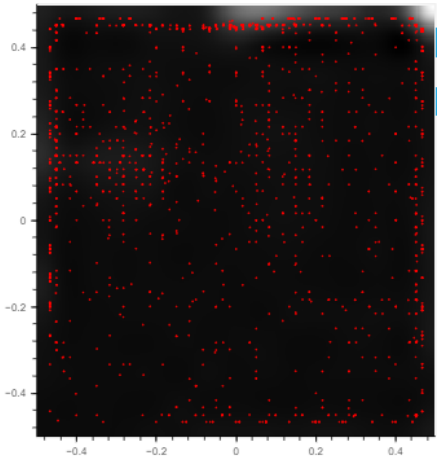


Figure 24: Sky metaphor for a 15x15 SOM with a high neighborhood radius

The Sky Metaphor (Figure 24) clearly shows that the area at the top of the SOM is the most dense, and as expected, it corresponds with the area with higher errors.

Now the large SOM (100x100) with small (0.4) neighborhood radius is analyzed. Chessboard visualization and clusters of units is shown in figure 25.

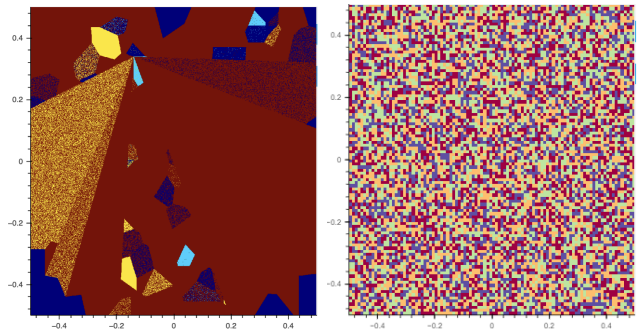


Figure 25: Chessboard and cluster of the units for a 100x100 SOM with a small neighborhood radius

As in the case with a regular SOM, when the neighborhood radius is too small, it is almost impossible to find clusters in the data.

Quantization error and Topographic Error are shown in figure 26.

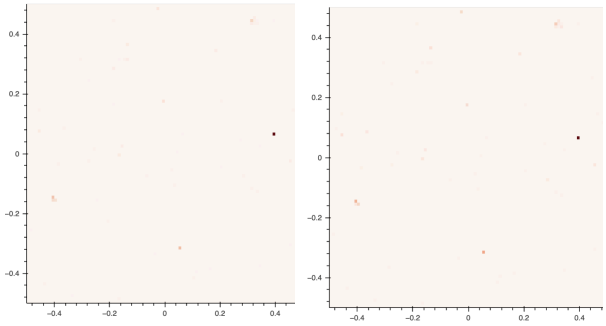


Figure 26:Quantization Error (left) and Topographic Error (right) for a 100x100 SOM with a small neighborhood radius

As mentioned before, the Hit Histograms (Figure 27) shows that the errors (Figure 26) are higher where the area is more dense.

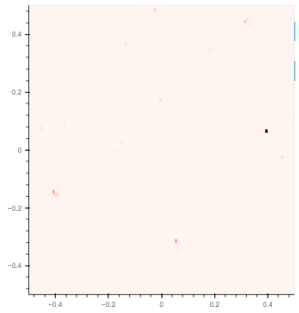


Figure 27: Hit Histogram for a 100x100 SOM with a small neighborhood radius

Now the high value of neighborhood radius (8) is used. Chessboard visualization and Clusters of units are shown in figure 28.

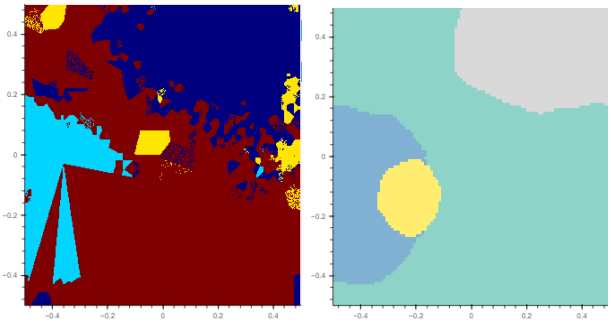


Figure 28: Chessboard and cluster of the units for a 100x100 SOM with a high neighborhood radius

The improvement of the SOM in visualizing the clusters and their boundaries is evident: this SOM shows the same structure, but upper scaled, as in figure 12. Using a higher neighborhood radius in a very large SOM seems to upscale the SOM.

Quantization Error and Topographic Error (fig 29) have similarities in the cells where error is higher.

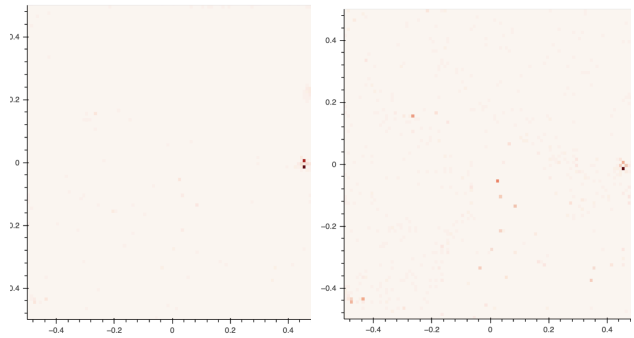


Figure 29: Quantization Error (left) and Topographic Error (right) for a 100x100 SOM with a high neighborhood radius

5. Analyze different initial learning rates

Now both "regular" and large dimensions of the SOM are trained, employing a very large and a very small learning rate.

First we trained a “regular” SOM with a very low learning rate of 0.1.
In the chessboard visualization, figure 30, we can see that the clusters are not clearly visible. It is still possible to distinguish the blue and light blue clusters, while the yellow one is spread around the SOM, and the brown one is dominating the remaining area. Applying KMeans of the units (on the right) it is easier to see that the green cluster is a sub-cluster of the yellow one, while the purple one is well-shaped and far from the others. Also the border effect can be seen here: the clusters specialized at the borders.
It’s important to remember that the colors do not represent the same cluster in the 2 visualizations..

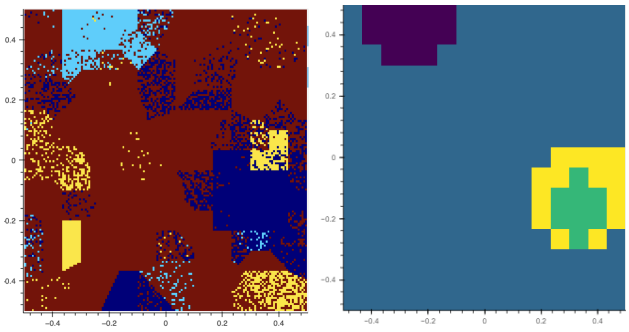


Figure 30: Chessboard and cluster of the units for a 15x15 SOM with a small initial learning rate

Visualizations of the Quantization Error and Topographic error in figure 31.

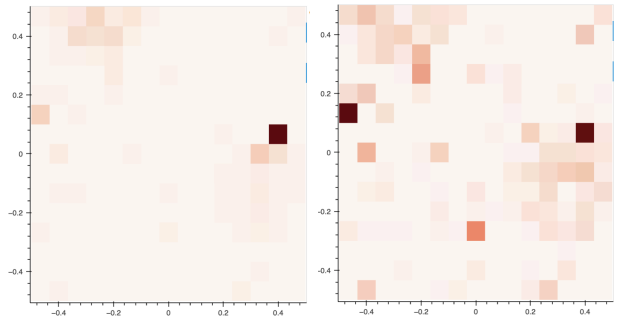


Figure 31:Quantization Error (left) and Topographic Error (right) for a 15x15 SOM with a small initial learning rate

The Hit Histogram is plotted in figure 32.

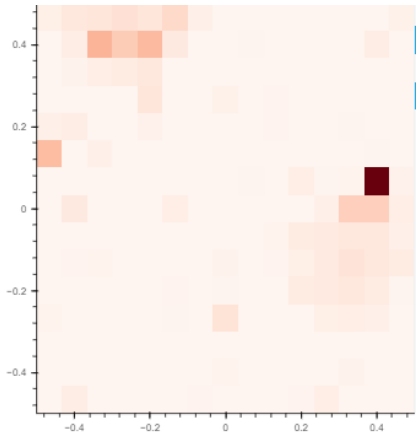


Figure 32: Hit Histogram for a 15x15 SOM with a small initial learning rate

A “regular” SOM with a very high learning rate of 2 is then trained.
In the chessboard visualization (figure 33 on the left), it is possible to see that the clusters are sufficiently separated, other than the yellow one which doesn’t have a defined structure.
On the right, Clusters of units are shown, and it is possible to see that they are defined and separated, other than the pink one which is a subcluster of the green one.

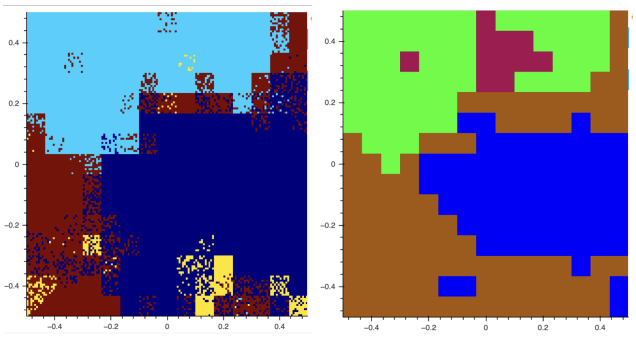


Figure 33: Chessboard and cluster of the units for a 15x15 SOM with a high initial learning rate

In figure 34 Topographic and Quantization errors are shown.

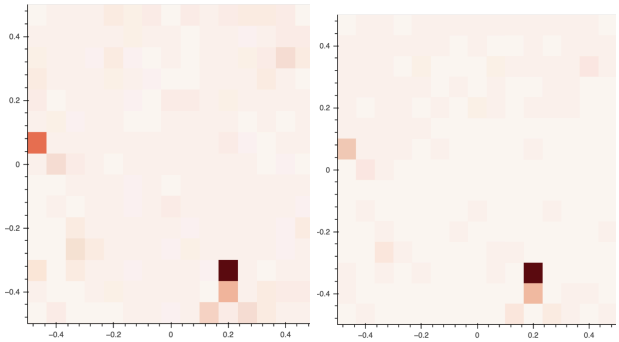


Figure 34:Quantization Error (left) and Topographic Error (right) for a 15x15 SOM with a high initial learning rate

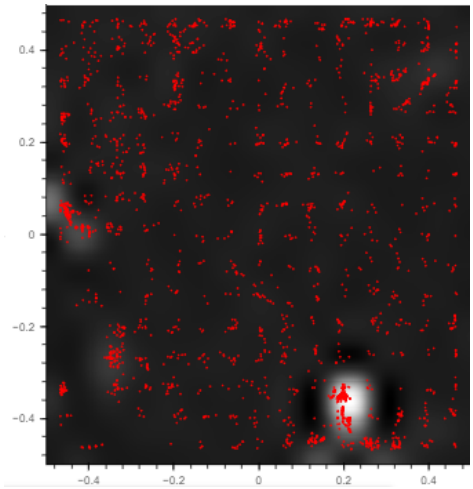


Figure 35:Quantization Error (left) and Topographic Error (right) for a 15x15 SOM with a high initial learning rate

As expected, the error is higher on the units where the most input data is mapped.

It is interesting to see that there is an area where both quantization and topographic error is higher, which is where the yellow class is (check fig 33 left).

Now we switch to the large SOMs trained with a low learning rate of 0.1. The Chessboard visualization (figure 36, left) and the Cluster of the Units (figure 36, right) show that the SOM failed to realize clusters.

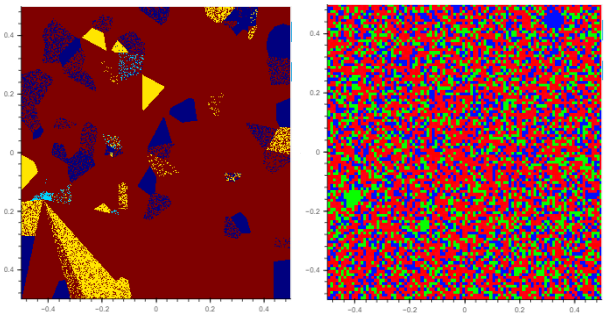


Figure 36: Chessboard and cluster of the units for a 100x100 SOM with a low initial learning rate

In both visualizations we can see that there is no clear structure in the data and the clusters are not defined.

In the following figures Quantization Error and Topographic Error is shown, which are very similar in the units.

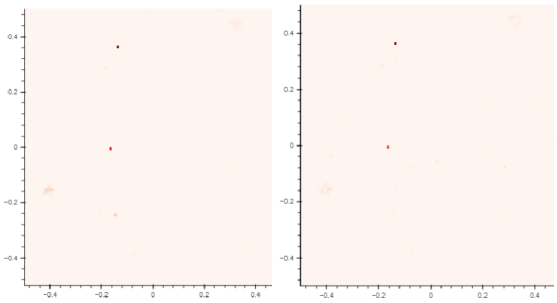


Figure 37:Quantization Error (left) and Topographic Error (right) for a 100x100 SOM with a low initial learning rate

The Hit Histogram is plotted (Figure 38).

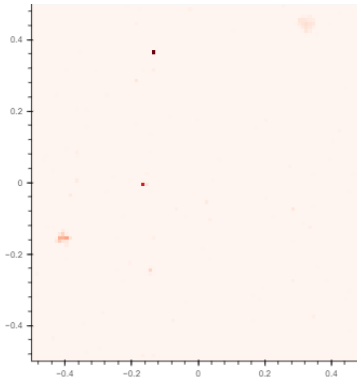


Figure 38: Hit Histogram for a 100x100 SOM with a low initial learning rate

Finally a SOM 100x100 with a high learning rate of 2 is trained. Both visualizations (Chessboard and Clusters of Units in fig 39) show a mixture of classes; there is no decent cluster structure. It is weird compared to the SOM of the same size (100x100) trained with learning rate = 1 in paragraph 3.

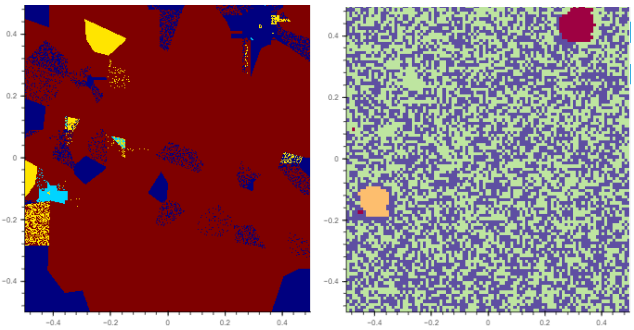


Figure 39: Chessboard and cluster of the units for a 100x100 SOM with a high initial learning rate

Quantization Error and Topographic Error (fig 40) are again very similar and homogenous, other than a specific unit with higher value.

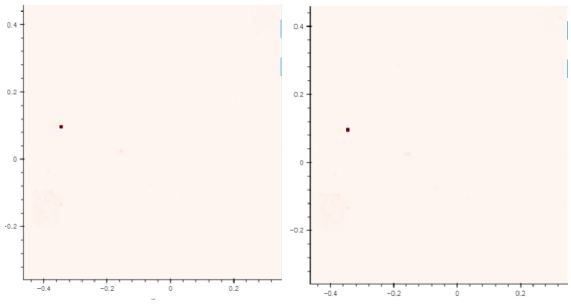


Figure 40:Quantization Error (left) and Topographic Error (right) for a 100x100 SOM with a high initial learning rate

The Hit Histogram is plotted (Figure 41).

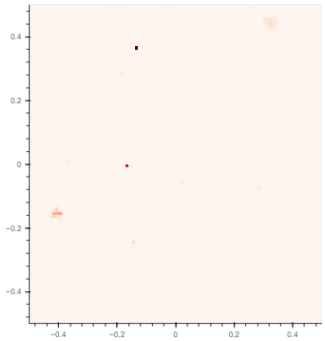


Figure 41: Hit Histogram for a 100x100 SOM with a high initial learning rate

From figure 40 and 41 it's again evident that error is higher where the data is more dense.

6. Analyze different scalings

As mentioned before, since SOM works with distances, it is important to scale the variables as we've done so far. In order to see the importance of this, a SOM is now trained without data normalization.
Chessboard and Cluster of units (KMeans) are shown in figure 42.

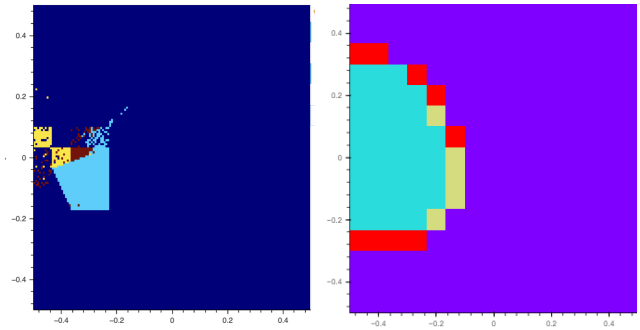


Figure 42: Chessboard and cluster of the units for a 15x15 SOM trained on unnormalized data

As expected the results are bad and it is not possible to distinguish cluster structure in the data. It looks like every cluster is a subcluster of the outer one.

The area where the clusters are coincides with the area where most of the input points are, as shown from the SkyMetaphor visualization in fig 43.

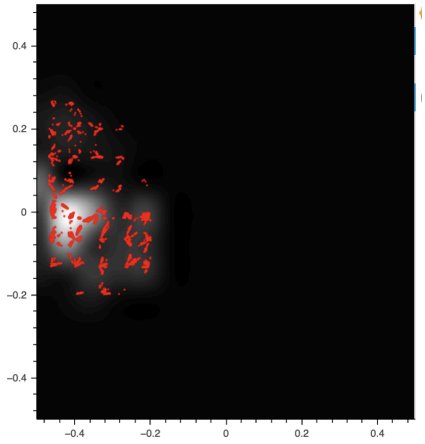


Figure 43: Sky Metaphor for a 15x15 SOM trained on unnormalized data

Quantization error and Topographic error are shown in figures below.

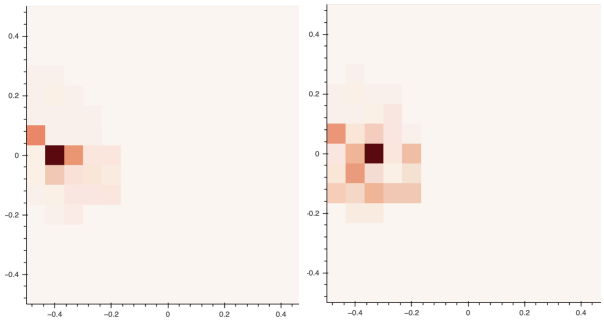


Figure 44:Quantization Error (left) and Topographic Error (right) for a 15x15 SOMtrained on unnormalized data

As expected the error is higher where most of the points are.

7. Analyze different max iterations

Then a 15x15 SOM is trained with default values (learning rate = 1, neighborhood radius = 1) varying the iterations 2, 5, 10, 50, 100, 1000, 5000, 10000.

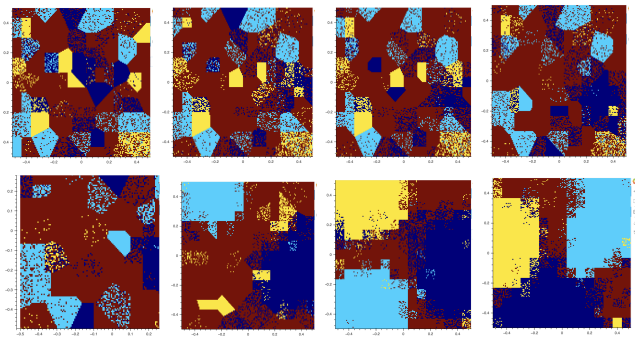


Figure 45: Chessboard for a 15x15 SOM with 2, 5, 10, 50, 100, 1000, 5000, 10000 iterations.

From the chessboards (fig 45), it is possible to observe how the classes are distributed across the 15x15 SOM. It's evident that until the 100 iterations, the SOM's training process fails to define clusters effectively. At 1000 iterations, clusters begin to emerge, and at 5000 and 10000 iterations, they become distinctly separate. Additionally, the border effect is clearly visible: clusters migrate towards the edges of the map.

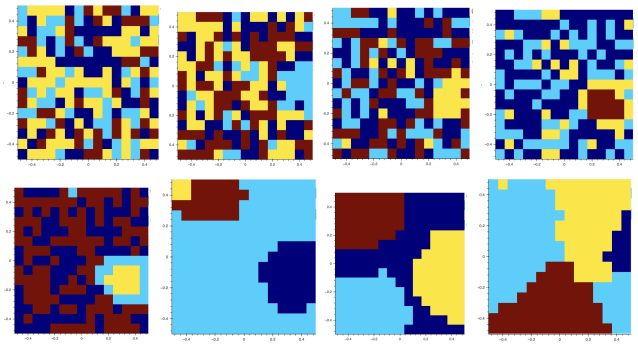


Figure 46: Cluster of the units for a 15x15 SOM with 2, 5, 10, 50, 100, 1000, 5000, 10000 iterations.

When clustering the units, we obtain similar information. With 2, 5, 10, 50 it’s possible to see that the clusters are not well-defined into our map.
When iterations are 1000 clusters start being visible: the 4 clusters are well-defined and it’s possible to see that the yellow is a sub-cluster of the brown one.
Raising the iterations to 5000 there are no subclusters anymore, and the groups are defined. It seems that with 10000 iterations the shape and separation of the cluster is less clear. It seems like increasing the iterations tends to have better results until a threshold value.

In the following figures the Quantization error is shown.

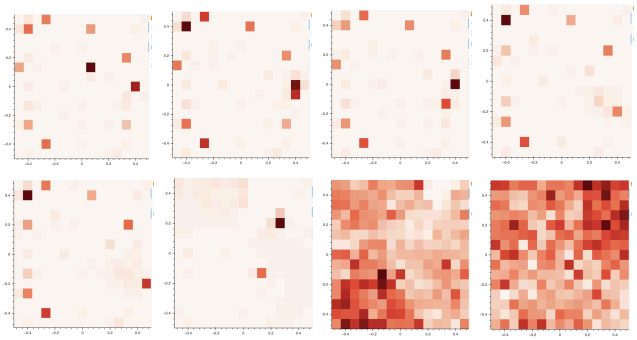


Figure 47: Quantization Error for a 15x15 SOM with 2, 5, 10, 50, 100, 1000, 5000, 10000 iterations.

Finally, the Topographic error is shown in fig 48.

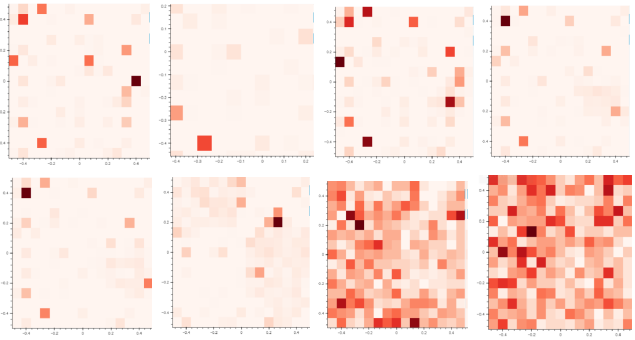


Figure 48:Topology Error for a 15x15 SOM with 2, 5, 10, 50, 100, 1000, 5000, 10000 iterations.

Hit histograms in figure 49.

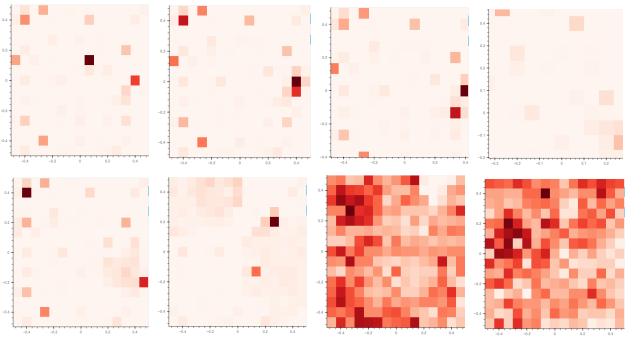


Figure 49: Cluster of the units for a 15x15 SOM with 2, 5, 10, 50, 100, 1000, 5000, 10000 iterations.

As we learnt in the previous paragraphs, there is an evident relationship between the errors and the hit histogram. Where more points are located, the error is higher.
It doesn't make sense to compare the evolution of the errors, because the plots show just a relative measure and we cannot infer about the absolute values but just on how it is distributed along the map.

GRID SEARCH

As we've seen, fine-tuning the parameters for a Self-Organizing Map (SOM) can be a challenging task. For this reason, it can be helpful and faster trying many more combinations performing a grid search with some range of parameters. By systematically exploring SOM sizes, learning rates, neighborhood functions, and iteration counts, this approach aims to uncover combinations that yield good results in capturing the structure of the data. The chosen evaluation metrics, including quantization error, topographic error, and silhouette score, provide a basis for assessing the SOM's performance.
The silhouette metric provides a measure of how well-separated the clusters are in the data. It can be computed for each data point in the dataset and then averaged to obtain an overall score for the entire clustering. It provides a quantitative measure of how well the SOM has organized the data into distinct clusters. The silhouette score ranges from -1 to 1, where higher is better.

The grid search generates a variety of SOM instances, each representing a unique combination of parameters. Then the models are ranked based on the metrics, which help us to identify interesting combinations of parameters. It is important to note that since metrics are sensitive to SOM size, this method serves as a qualitative guide, facilitating the selection of SOMs. The objective is not merely to select the SOM with the lowest error but to discern patterns and structures that align with the underlying characteristics of the data.

GridSearch has been performed trying all the combination of the following parameters:
som_sizes = [(7, 7), (15, 15), (10, 20), (100, 100)]
learning_rates = [0.5, 1.0, 2.0, 4.0]
neighborhood_functions = ['gaussian', 'triangle', 'mexican_hat']
neighborhood_radii = [0.5, 1, 2, 5]
iterations = [100, 1000, 5000, 10000]

Then, the SOMs are ranked based on Topographic error, Quantization error, and Silhouette.

```
Top Configs with Least Quantization Err:
Config: [100, 100, 0.5, 'triangle', 10000, 5], Quantization Err:
0.174, Topographic Err: 0.1197, Silhouette 0.028
Config: [100, 100, 2.0, 'triangle', 10000, 2], Quantization Err:
0.197, Topographic Err: 0.734, Silhouette 0.111
Config: [100, 100, 2.0, 'gaussian', 10000, 2], Quantization Err:
0.236, Topographic Err: 0.060, Silhouette 0.027
Config: [100, 100, 4.0, 'gaussian', 10000, 2], Quantization Err:
0.250, Topographic Err: 0.185, Silhouette 0.018
Config: [100, 100, 4.0, 'gaussian', 10000, 1], Quantization Err:
0.254, Topographic Err: 0.677, Silhouette 0.034
Config: [100, 100, 2.0, 'gaussian', 10000, 1], Quantization Err:
0.272, Topographic Err: 0.494, Silhouette 0.127
…
Config: [100, 100, 1.0, 'gaussian', 10000, 0.5], Quantization Err:
0.679, Topographic Err: 0.921, Silhouette 0.143
Config: [100, 100, 2.0, 'mexican_hat', 10000, 0.5], Quantization
Err: 0.692, Topographic Err: 0.999, Silhouette 0.115
Config: [100, 100, 2.0, 'mexican_hat', 10000, 1], Quantization
Err: 0.699, Topographic Err: 1.0, Silhouette 0.111
Config: [15, 15, 0.5, 'gaussian', 10000, 1], Quantization Err:
0.710, Topographic Err: 0.290, Silhouette 0.143
Config: [15, 15, 0.5, 'triangle', 10000, 2], Quantization Err:
0.712, Topographic Err: 0.463, Silhouette 0.148
```

We already expected that bigger SOMs would have had much smaller quantization error, for this reason we focus on the first small SOM with low quantization error (in bold).

Top Configs with Least Topographic Err:

Config: [7, 7, 2.0, 'mexican_hat', 1000, 5], Quantization Err: 3.208, Topographic Err: 0.000354, Silhouette 0.148

Config: [15, 15, 0.5, 'mexican_hat', 1000, 5], Quantization Err: 3.965, Topographic Err: 0.00035, Silhouette 0.122

Config: [15, 15, 0.5, 'mexican_hat', 10000, 5], Quantization Err: inf, Topographic Err: 0.000709, Silhouette 0.163

Config: [7, 7, 0.5, 'mexican_hat', 5000, 5], Quantization Err: 2.615, Topographic Err: 0.001773, Silhouette 0.094

…

Config: [7, 7, 4.0, 'mexican_hat', 100, 5], Quantization Err: 2.982, Topographic Err: 0.005674, Silhouette -0.034

Config: [15, 15, 1.0, 'mexican_hat', 1000, 5], Quantization Err: 3.624, Topographic Err: 0.00564, Silhouette 0.147

Config: [10, 20, 4.0, 'mexican_hat', 1000, 5], Quantization Err: 2.747, Topographic Err: 0.00602, Silhouette 0.061

Config: [7, 7, 0.5, 'gaussian', 1000, 5], Quantization Err: 3.807, Topographic Err: 0.00709, Silhouette 0.019

Config: [100, 100, 0.5, 'gaussian', 10000, 5], Quantization Err: 0.478, Topographic Err: 0.0092, Silhouette -0.003

Config: [7, 7, 0.5, 'mexican_hat', 100, 5], Quantization Err: 2.335, Topographic Err: 0.009574, Silhouette 0.111

Config: [15, 15, 2.0, 'triangle', 10000, 5], Quantization Err: nan, Topographic Err: 0.013121, Silhouette nan

Config: [10, 20, 2.0, 'triangle', 10000, 5], Quantization Err: nan, Topographic Err: 0.013121, Silhouette nan

Config: [7, 7, 0.5, 'mexican_hat', 10000, 5], Quantization Err: 3.855, Topographic Err: 0.014539, Silhouette 0.088

Config: [10, 20, 0.5, 'mexican_hat', 10000, 5], Quantization Err: inf, Topographic Err: 0.014539, Silhouette 0.088

We also expect small SOM to have smaller topographic error, for this reason we focus on the big SOM which still has a small topographic error.

Top Configs with Higher Silhouette:

Config: [10, 20, 4.0, 'mexican_hat', 5000, 2], Quantization: inf, Topographic Err: 1.0, Silhouette 0.695

Config: [10, 20, 4.0, 'gaussian', 5000, 0.5], Quantization: 4.414, Topographic Err: 0.999, Silhouette 0.647

Config: [15, 15, 4.0, 'mexican_hat', 5000, 2], Quantization: inf, Topographic Err: 1.0, Silhouette 0.523

Config: [15, 15, 4.0, 'gaussian', 5000, 0.5], Quantization: 4.366, Topographic Err: 0.900, Silhouette 0.453

Config: [15, 15, 4.0, 'triangle', 10000, 1], Quantization: 4.408, Topographic Err: 1.0, Silhouette 0.430

Config: [7, 7, 2.0, 'mexican_hat', 10000, 1], Quantization: inf, Topographic Err: 1.0, Silhouette 0.414

Config: [7, 7, 1.0, 'mexican_hat', 100, 2], Quantization: 3.536, Topographic Err: 1.0, Silhouette 0.389

Config: [7, 7, 2.0, 'mexican_hat', 1000, 1], Quantization: 2.830, Topographic Err: 1.0, Silhouette 0.388

Config: [10, 20, 2.0, 'mexican_hat', 1000, 2], Quantization: 2.830, Topographic Err: 1.0, Silhouette 0.388

Config: [15, 15, 2.0, 'mexican_hat', 1000, 2], Quantization: 3.254, Topographic Err: 1.0, Silhouette 0.388

Then, as we can see from above, we also focus on a rectangular som which has the best silhouette value.

Finally we could also give a try to a smaller SOM which had low error values and a decent silhouette score.

SOM Size: (7, 7), Learning Rate: 2.0, Neighborhood function: mexican_hat, Iterations: 1000,
Quantization Error: 3.207, Topographic Error: 0.000354, Silhouette Score: 0.147

An easy and fast way to have an idea of the clusters when dealing with SOMs are the Chessboard visualization and Clusters of Units, so these plots will be mostly analysed in order to decide the best parameter combination. In addition also SkyMetaphor will be plotted, to have an idea of the densities.

Configuration 15x15 with learning rate 0.5, neighborhood_function = 'gaussian', 10000 iterations, and neighborhood_radius = 1.

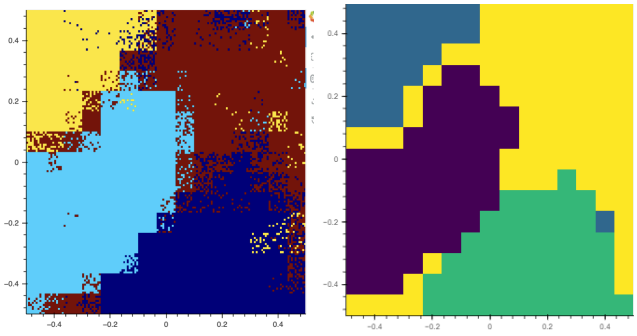


Figure 50: Chessboard and Clustering of the unit

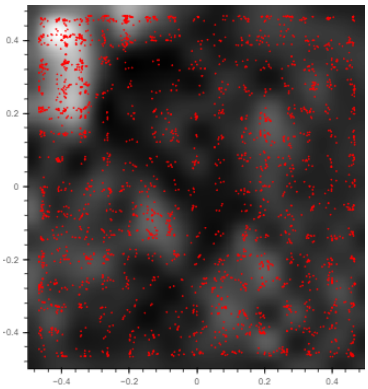


Figure 51: Sky Metaphor

Configuration 100x100 with learning rate 0.5, neighborhood_function = 'gaussian', 10000 iterations, and neighborhood_radius = 5.

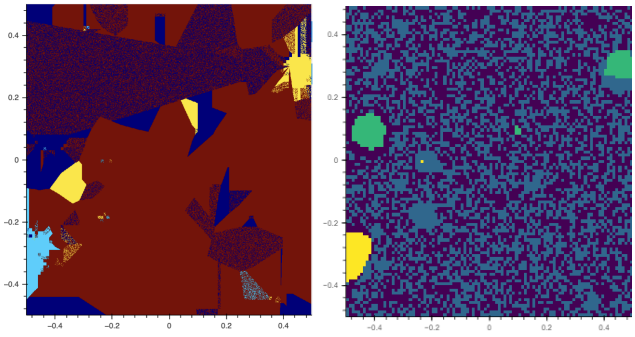


Figure 52: Chessboard and Clustering of the unit

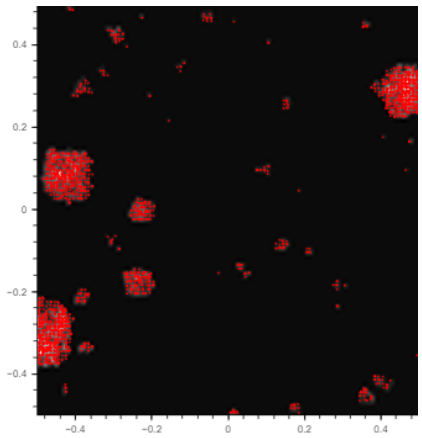


Figure 53: Sky Metaphor

Configuration 10x20 with learning rate 0.5, neighborhood_function = ‘gaussian’, 5000 iterations, and neighborhood_radius = 2.

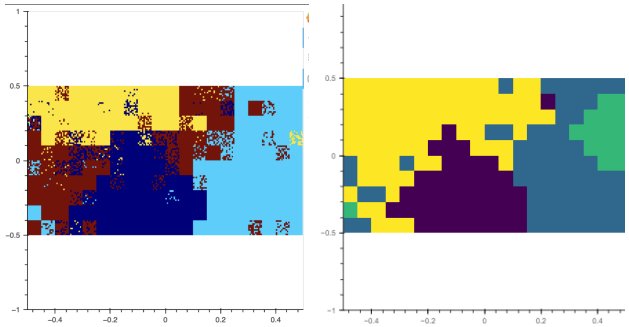


Figure 54: Chessboard and Clustering of the unit

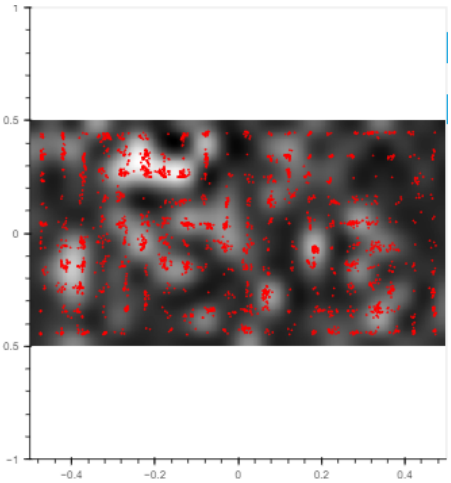


Figure 55: Sky Metaphor

Configuration 7x7 with learning rate 2, neighborhood_function = mexican_hat, 1000 iterations, and neighborhood_radius = 1.

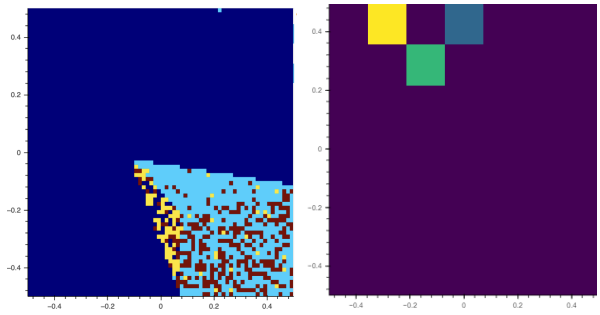


Figure 56: Chessboard and Clustering of the unit

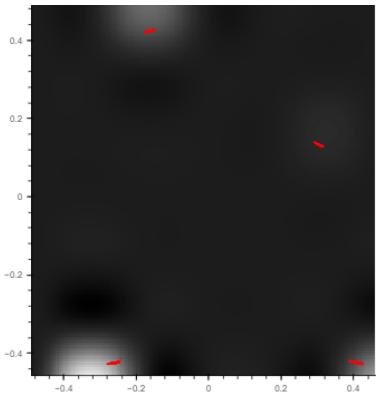


Figure 57: Sky Metaphor

As we can see, even if those SOM parameters were chosen based on metrics such as topographic error and quantization error, many times they don’t show a clear structure of the data. The automatized process is evidently not efficient for this kind of task, human interaction is much more important.

Optimal SOM

Even if from the plots of the previous chapter it may look like the 15x15 SOM works better, it has been decided to proceed the final analysis with a 100x100 SOM. Based on our search, we later realized that when dealing with emergent SOMs, even if the chessboard visualization doesn't define clusters easily, other visualizations show better the structure.

Finally we decided that the best SOM to work on is:
x_som, y_som = 100,100
learning_rate = 1
neighborhood_function = 'gaussian'
iterations = 10000
neighborhood_radius = 6

Looking at the P-Matrix (figure 58) we can clearly see that since the SOM is very large, the data is not distributed equally: there are 3 main areas with most of the input points. Later we’ll see that each of these areas is actually a cluster. The 4th cluster,

instead, corresponds to the big area in the center spread all around the SOM (with low density).

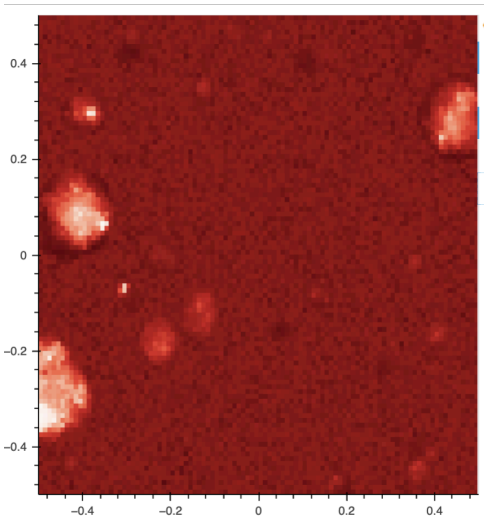


Figure 58: P-matrix of the optimal SOM

Looking at activity histogram (figure 59) it is possible to see that the most distant areas are actually the same areas where clusters were previously found. It's important to keep in mind that this visualization shows distances from a single point, and we are selecting a point in the middle on the map in the 4th big cluster.

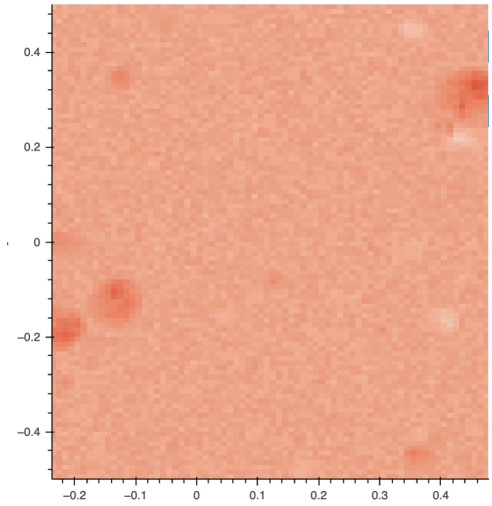


Figure 59: Activity histogram of the optimal SOM

As mentioned before, the chessboard visualization (fig 60) is very confusing and doesn't help finding cluster structure.
Usually the chessboard is very helpful, but when dealing with emergent SOMs, it's important to consider that areas which look very confusing with many classes may have just a few points.

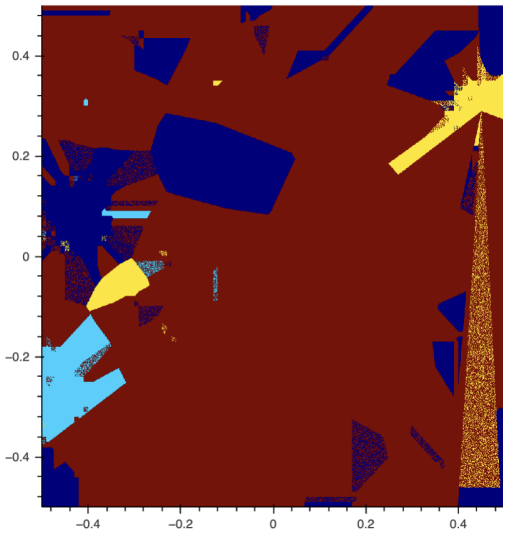


Figure 60: Chessboard of the optimal SOM

We can see much better structures visualizing at the same time the clusters of units (fig 61) and the densities of points with P matrix (fig 58). Three of the clusters are circular, very well separated from each other and the last one cluster is spread all over the map. Even if the size looks very different, as we can see from the P matrix (fig 58), most of the area doesn't contain any point. Between the clusters no hierarchical relations are present, but the cluster covering the larger area is the more general, meanwhile the others specify themself in the borders of the map. In addition, as shown later, the cardinality of all the clusters are similar.

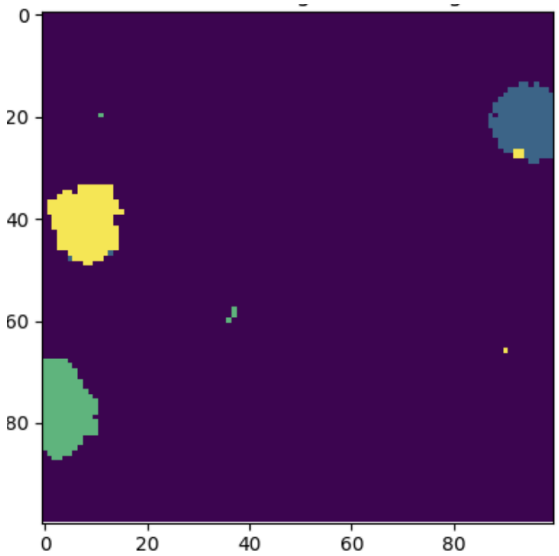


Figure 61: Cluster of the unit of the optimal SOM

In figure 62 the Quantization error and Topographic error are shown. As expected the areas with higher density of points have a higher error.

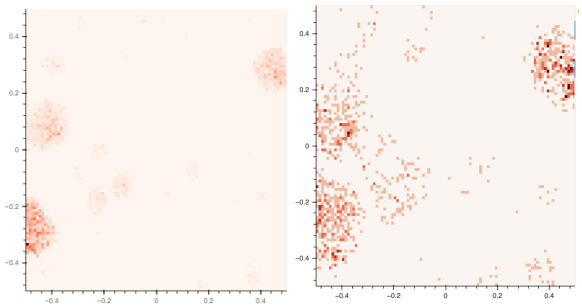


Figure 62: Quantization Error (left) and Topographic Error (right)

While both topographic and quantization errors may look high, they are not highlighting problems in our case. In the emergent SOM's it's obvious to see higher error in the more dense region, although these errors have a low absolute value. The Total Topographic Error of our map is 0.399 and the Total Quantization Error 0.456

Finally it is interesting to find out how good the model actually is. Usually this is not possible when performing clustering, because classes are not given; in this scenario, since classes are provided, it is possible to have some useful metrics.
After building the SOM, and clustering the units, the percentage of each class in each cluster is calculated.

Cluster 1 (830 points):
Class 4: 651 points, 78.43%
Class 1: 137 points, 16.51%
Class 3: 31 points, 3.73%
Class 2: 11 points, 1.33%

Cluster 2 (563 points):
Class 1: 549 points, 97.51%
Class 4: 6 points, 1.07%
Class 2: 4 points, 0.71%
Class 3: 4 points, 0.71%

Cluster 3 (730 points):
Class 2: 690 points, 94.52%
Class 4: 39 points, 5.34%
Class 3: 1 points, 0.14%

Cluster 4 (697 points):
Class 3: 669 points, 95.98%
Class 1: 19 points, 2.73%
Class 4: 9 points, 1.29%

First of all, it is possible to see that each of the clusters have similar cardinalities.
In addition, it is evident that the SOM classifies the data points very well, since each cluster has a very high percentage (mostly even higher than 90%) of a specific class, and very low of the others. Clusters definitely found the structure and the pattern of each class.
In the following table (Table 1) we computed the Confusion Matrix for the 4 classes. The performance is very satisfying with an Accuracy of 90.84%.

		Actual Values			
		1	2	3	4
Predicted	1	**549**	**4**	**4**	**6**
	2	**0**	**690**	**1**	**39**
	3	**19**	**0**	**699**	**9**
	4	**137**	**11**	**31**	**651**

Table 1: Confusion Matrix of the prediction

Prediction with 7 classes

After finding a good model, we decided to test it with the full dataset containing 7 classes.
First the chessboard visualization is plotted in figure 63, which looks very confusing. The reason is that since it's an emergent SOM, the data is not distributed equally. As it is possible to see from figure 64, the data is more concentrated in areas where the classes are actually more homogeneous.

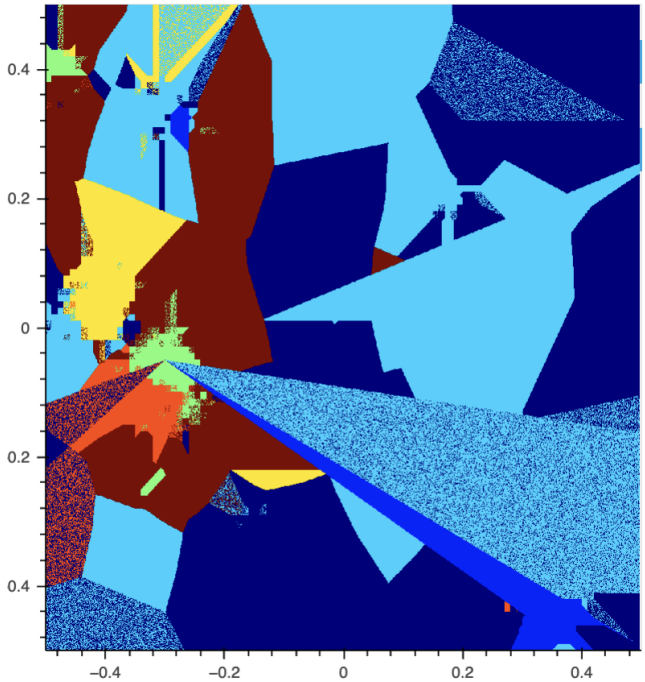


Figure 63: Chessboard visualization with 7 classes

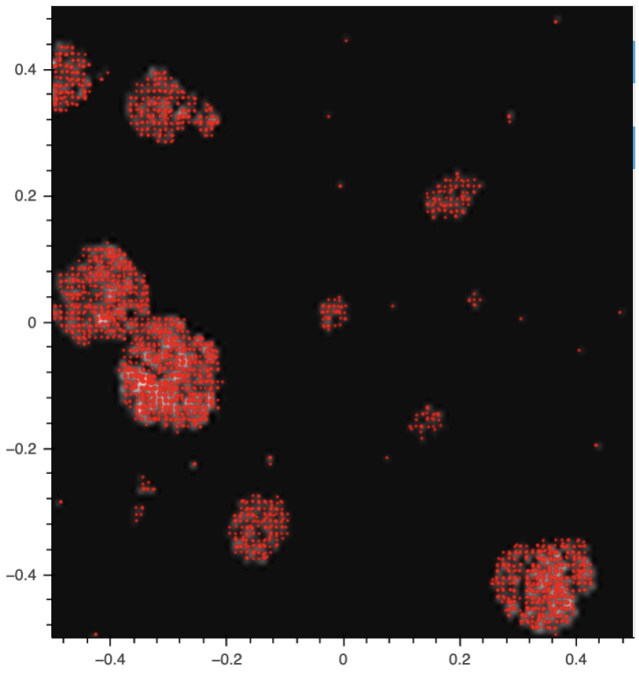


Figure 64: Sky Metaphor

Clustering of Units in figure 65 helps us understand the structure of the data.

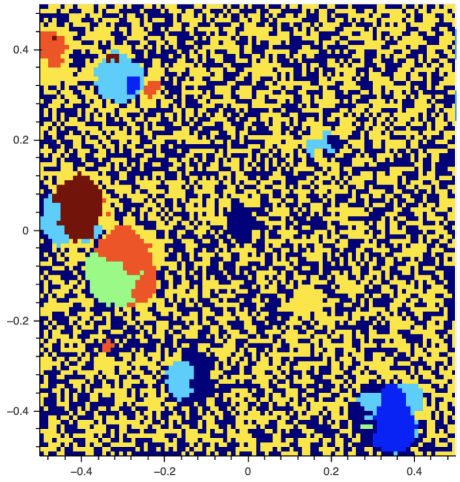


Figure 65: Clustering of Units

It's evident combining the visualizations that areas where the density is higher represent clusters. The two clusters in yellow and dark blue are spread all around the SOM and makes the definition of clusters harder; this isn't a problem since from figure 64 it's possible to see that there are very few data points in that area.

In order to understand how actually the clusters are different, it's important to visualize the distances with Activity Histogram.

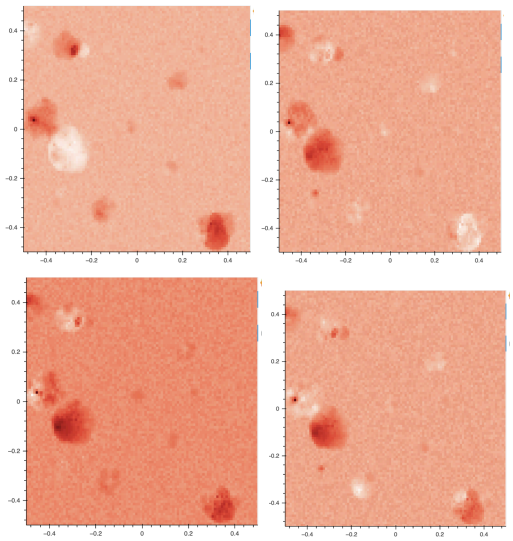


Figure 66: Activity Histograms

In figure 66 there are 4 activity histograms of 4 different points, each one in a different cluster. It's possible to see that, considering each point a representative of its own cluster, it is very different from the other points in the other clusters, meaning clusters are actually different.

As it's possible to see from figure 67, the errors are actually higher where the data points are more dense.

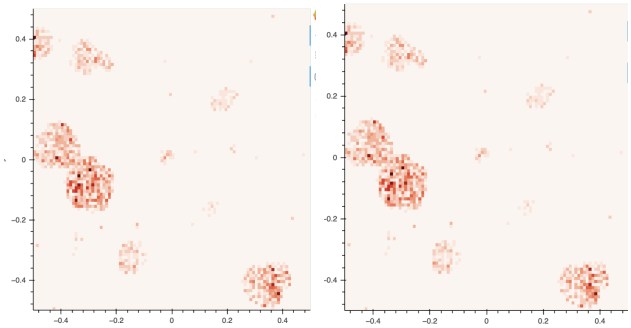


Figure 67: Clustering of Units

Finally, we show for each cluster the percentage of data points of each class.

Cluster 0:
Class **CALI**: 57.31% - Units: 345
Class BARBUNYA: 37.38% - Units: 225
Class HOROZ: 4.15% - Units: 25
Class SEKER: 0.17% - Units: 1
Class SIRA: 1.00% - Units: 6

Cluster 1:
Class **HOROZ**: 96.07% - Units: 464
Class BARBUNYA: 0.41% - Units: 2
Class CALI: 1.45% - Units: 7
Class DERMASON: 0.21% - Units: 1
Class SIRA: 1.86% - Units: 9

Cluster 2:
Class **DERMASON**: 63.09% - Units: 470
Class BARBUNYA: 0.27% - Units: 2
Class HOROZ: 0.67% - Units: 5
Class SEKER: 1.48% - Units: 11
Class SIRA: 34.50% - Units: 257

Cluster 3:
Class **BARBUNYA**: 63.03% - Units: 237
Class CALI: 36.70% - Units: 138
Class HOROZ: 0.27% - Units: 1

Cluster 4:
Class **SIRA**: 79.57% - Units: 222
Class BARBUNYA: 11.11% - Units: 31
Class CALI: 3.23% - Units: 9
Class DERMASON: 0.36% - Units: 1
Class HOROZ: 1.79% - Units: 5
Class SEKER: 3.94% - Units: 11

Cluster 5:
Class **BOMBAY**: 99.80% - Units: 500
Class CALI: 0.20% - Units: 1

Cluster 6:
Class **SEKER**: 92.80% - Units: 477
Class BARBUNYA: 0.58% - Units: 3
Class DERMASON: 5.45% - Units: 28
Class SIRA: 1.17% - Units: 6

As we can see, cardinalities between clusters are very similar, and each cluster actually corresponds to a different class.

The Confusion Matrix is shown in the following table. The performance is worse than the one obtained considering 4 classes with an accuracy of 77,57%, which is still satisfying

		Actual Values						
		1	2	3	4	5	6	7
Predicted Values	1	345	225	25	1	6	0	0
	2	138	237	1	0	0	0	0
	3	7	2	464	0	9	1	0
	4	0	3	0	477	6	28	0
	5	9	31	5	11	222	0	0
	6	0	2	5	11	257	470	0
	7	1	0	0	0	0	0	500

Table 2: Confusion Matrix of the prediction of the 7 classes

Conclusion

In conclusion, our exploration into Self-Organizing Maps (SOM) has highlighted the significance of using multiple visualizations to find patterns in the data. While a single visualization can provide insights, a comprehensive understanding emerges from the synthesis of diverse visual perspectives.
During the assignment many parameter configurations were explored, which showed that the size of the SOM is very relevant. Initially Regular-sized SOMs with resolutions of 10/15 seemed to work well, but emergent SOMs performed even better. It's still important to consider that this kind of size is not realistic when datasets are very large. Also parameters as learning rate and neighborhood radius were important, but default values often performed decently.
It is very important to visualize the SOM when changing the parameters, because as seen in the Grid Search, when delegating the choice of the best parameters to an algorithm based on some metrics, results are usually disappointing.
In terms of visualization effectiveness, the chessboard plot, showcasing class distributions around the SOM, is definitely useful to have an idea of the clusters. When accompanied by the P matrix indicating densities, this combination offered a comprehensive view of the data landscape. Unfortunately, when dealing with unsupervised clustering, chessboard visualization is not possible. Distance visualizations and Clustering of Units, combined with densities visualizations, play a crucial role in defining structures.
Unfortunately some visualizations that were very useful in studying the topology weren't working, as Neighborhood graph, so our analysis missed this kind of information.

Feedback

We personally think the topic of SOM was interesting and new compared to the usual. We liked the simplicity of the idea, and the usefulness of the visualizations that help to really realize what is going on. In the development there were some problems with some visualizations that did not work (such as neighborhood graphs and other topology plots), but they will surely be fixed with time. Perhaps it would have been more interesting to spend more time during the course so that the topic could have been addressed more calmly.

Reference

Andreas Rauber, Slides from VU Self-Organizing Systems, Technische Universität Wien, 2023W