SOM visualizations implementation in Python

Visualizing attribute values per unit (coding assignment **E**)

Cristiano Pacini (12206613), Joanna Peloso (12206622), Xavier Pisco (12206635)

Self-Organizing Systems, February 2023

1 Introduction

GitHub repository: https://github.com/cripacz/Self-Organizing-Maps

This report describes our work on the "implementation" option for assignment 3. This assignment was about implementing SOM visualizations in Python. We chose to implement the visualization of attribute values per unit (option e). For this purpose, we plotted the value of selected attributes as bar charts, radar charts, chernoff faces, box plots, and violin plots. We trained a small map (10x10) and a large map (100x60) on two different datasets, the chainlink, and the 10-clusters. We compared the different visualizations of attribute values per unit between the trained and non-trained datasets.

2 Implementation

2.1 Code explanation

The python file "plots.py" contains the implementation, with comments, of the different types of visualizations regarding different attributes for each unit in the SOM maps. These are then tested in the notebook "implementation_final.ipynb", which shows the results of our experiments on the data.

For both datasets (chainlink and 10-clusters), we trained two maps of different sizes (10x10 and 100x60) with the miniSOM library.

2.2 Customizing parameters

For each visualization function, it is possible to select the attributes to view.

The third parameter of each function corresponds to the list of the attribute numbers to be shown.

It is also possible to choose the units displayed by changing the units numbers in the list of the second parameter of each function.

For the barplot function, users have the possibility to choose to see the plots sorted by the real topology of the map or not (fourth argument of the function).

2.3 Barplots

For each selected unit, a barplot is displayed, where the bars' height represents the attribute value in the weight vectors. The user can display:

- the whole map preserving its topology, *i.e.*, with every unit in the correct spot
- only some selected cells, presented sequentially, in order to better visualize the bars for each single cell.

This can be managed through the following parameters:

- *true_vis* is a boolean; if True, the map's units are displayed with no labels and in the correct topology. If False, the selected units are displayed with labels and subsequentially.
- *mapsize* needs to be a tuple and contains the 2D size of the map. It is needed when *true_vis* == True.

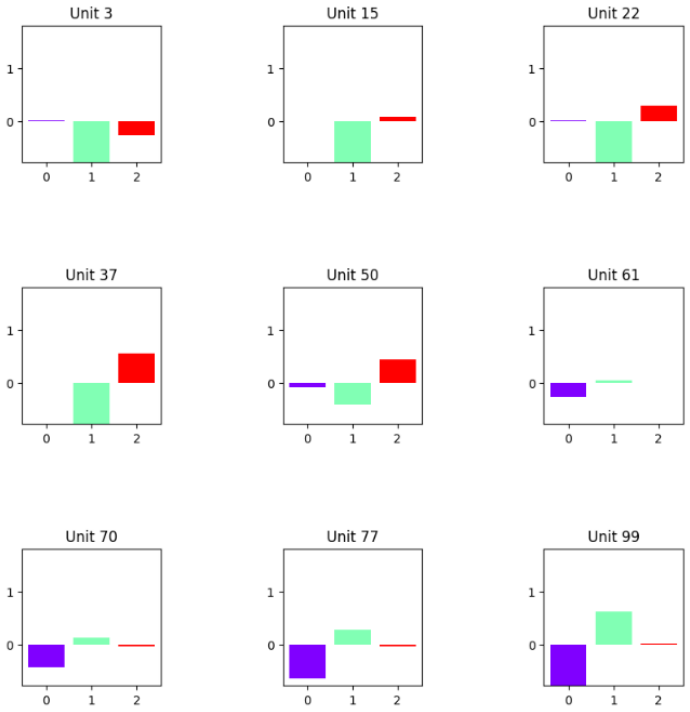


Figure 1: Barpplots for trained map (10x10) on chainlink dataset

2.4 Chernoff

Chernoff faces are a convenient way of representing our data since every attribute is mapped into a facial feature. The variables are the input data, the selected units to show, and the attributes to display. The units are displayed sequentially. In this case, true_vis is not implemented because it is useful for a high number of selected units, and chernoff faces would be hardly interpretable. The implementation relies on the python package "ChernoffFace".

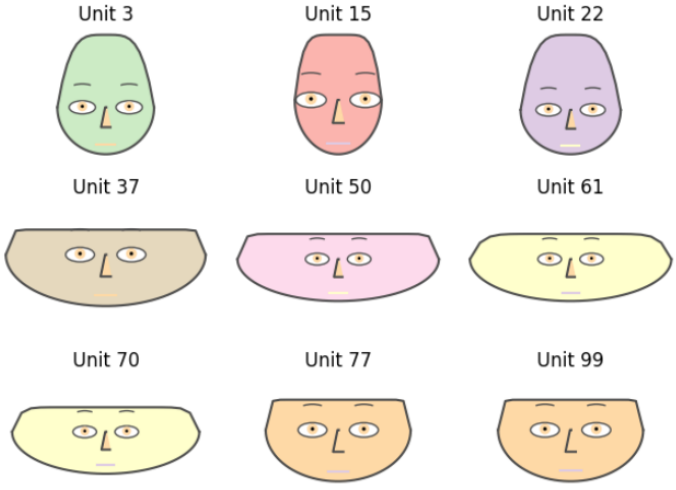


Figure 2: Chernoff faces for trained map (10x10) on chainlink dataset (example units).

2.5 Radarplots

Radar plots allow the representation of data in a polygonal/circular frame. The variables are the input data, the selected units to show, and the attributes to display. The units are displayed sequentially, and the number of rows/cols is computed by rounding the square root of the total number of selected cells in order to mimic the shape of an nXn matrix as much as possible. In this case, true_vis is not implemented because it is useful for a high number of selected units, and radar plots would be hardly interpretable.

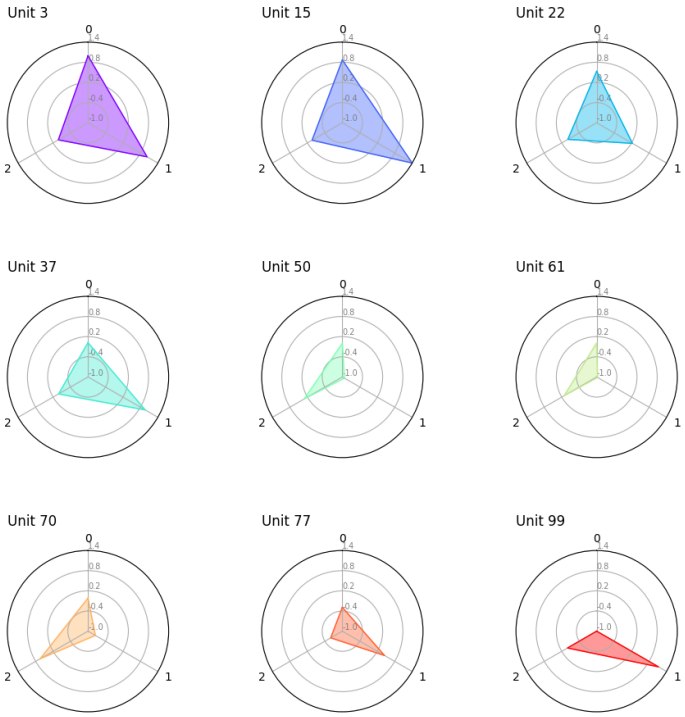
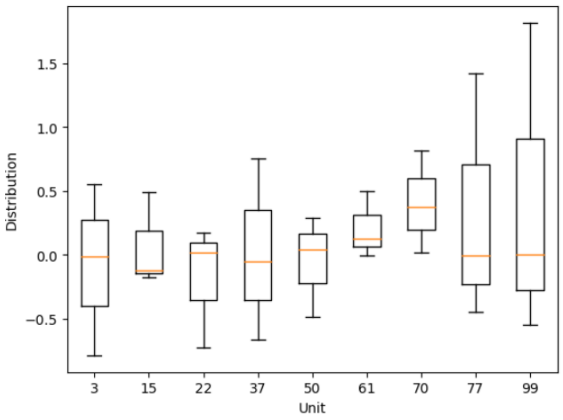


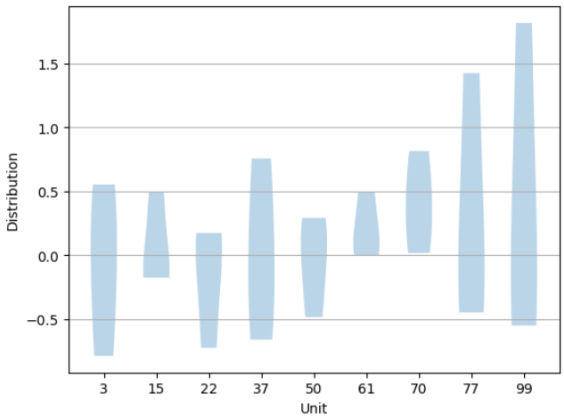
Figure 3: Chernoff for trained map (10x10) on chainlink dataset (example units). We can compare the results with the ones obtained in the Chernoff faces' plots. It is clear from the radar plots above how units 50, 61, 70 have a small value for component 1, in contrast to the other selected units. This is translated in the Chernoff faces in the width of the forehead, which is narrow for the 3 mentioned units, while large for the others. On the other hand, component 2 is mapped into the height of the forehead, as shown by the comparison of units 70, 77, 99 in the radar plot and in Chernoff faces.

2.6 Boxplots / Violinplots

For each of the selected units, a box plot contains the distribution of each attribute for the data instances mapped onto a certain unit among the selected ones, which are the data samples for which the unit is the best-matching unit.



(a) Boxplot for trained map (10x10) on chainlink dataset.



(b) Violinplot for trained map (10x10) on chainlink dataset.

3 Experimentations

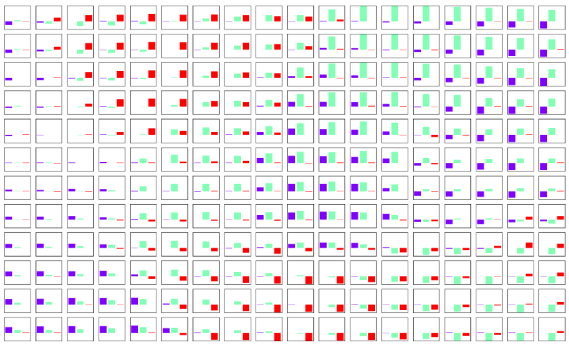
3.1 Comparison between trained and pre-trained data

3.1.1 Barplot

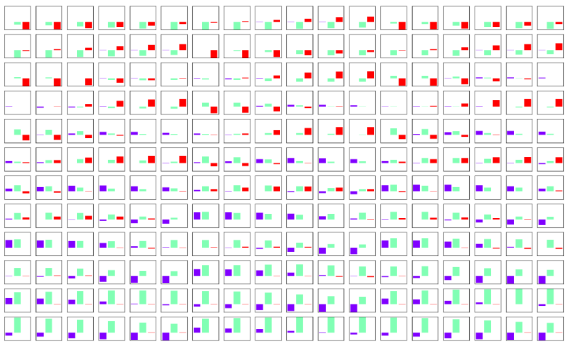
In order to validate the correctness of our implementation, we compared the visualization of the trained map and the pre-trained data. We used the barplot to visualize the differences between trained and untrained data.

Chainlink dataset

Figures 5a and 5b show the barplot visualizations for the chainlink dataset before and after training (on a 12x18 map).



(a) Barplot for trained map on chainlink dataset



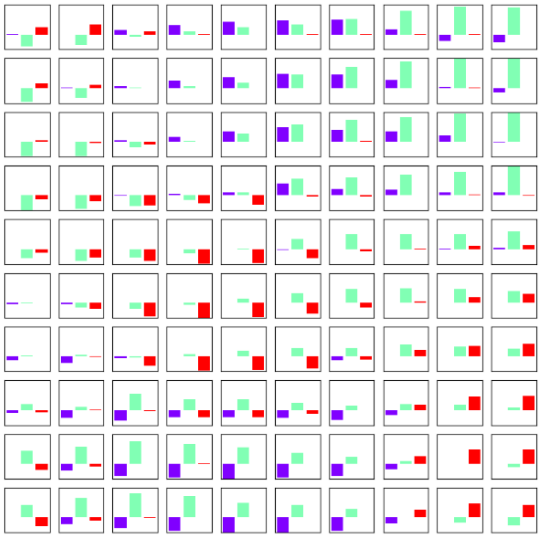
(b) Barplot for untrained chainlink dataset

In Figure 5b, we can see that the second attribute (green) is higher on the bottom units of the graph, while the third attribute (red) is close to zero. In contrast, the highest units of the graph present the first attribute (purple) close to zero. This graph seems to split horizontally into two parts.

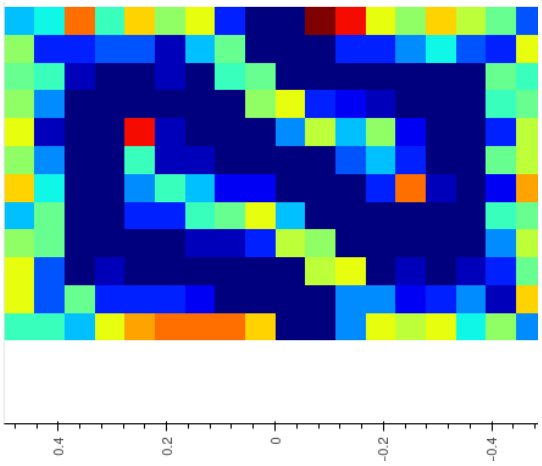
Figure 5a shows the same dataset on an 18x12 trained map. We can see four different parts in that graph:

- at the top right of the graph : the first attribute (purple) is negative, the second one (green) is positive, and the third one (red) is close to zero ;
- at the top left of the graph : the first attribute (purple) is close to zero, the second one (green) is also close to zero, and the third one (red) is positive ;
- at the bottom right of the graph : the first attribute (purple) is close to zero, the second one (green) is negative, and the third one (red) is negative ;
- at the bottom left of the graph : the first attribute (purple) is positive, the second one (green) is positive, and the third one (red) is close to zero.

This arrangement is closer to the true topology of the chainlink than the untrained data, as we can notice by the comparison with the hit histogram, Figure 6b.

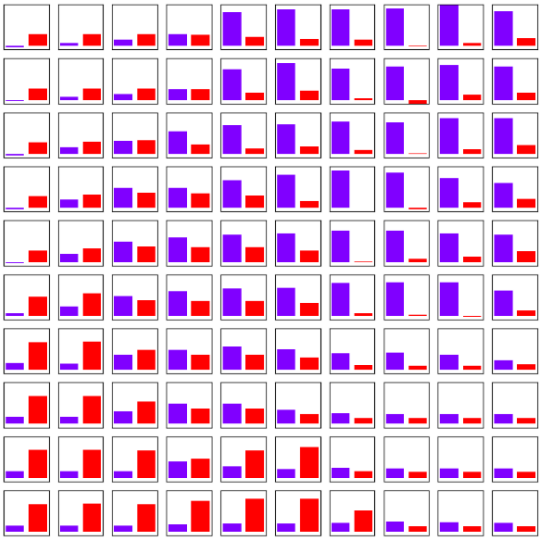


(a) Barplot for trained map (10x10) on the chainlink dataset

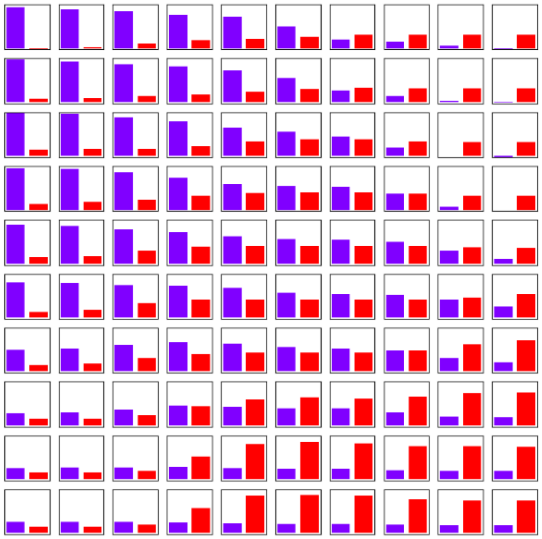


(b) Hit histogram for chainlink dataset

10 clusters dataset



(a) Barplot (Attribute 1 and 4) for trained map on 10-clusters dataset



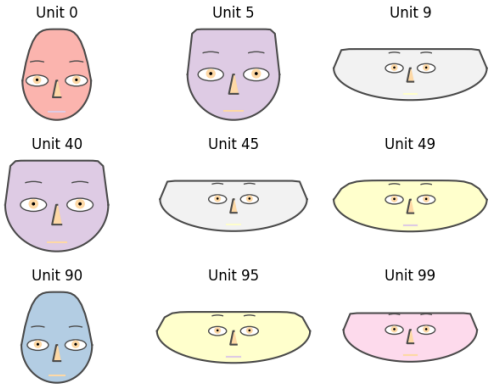
(b) Barplot (Attribute 1 and 4) for untrained 10-clusters dataset

Figures 7a and 7b show the first and fourth attributes for the 10 clusters dataset before and after training. We can notice that areas of both maps are mirrored, for instance, the top right of the trained map and the top left of the untrained map.

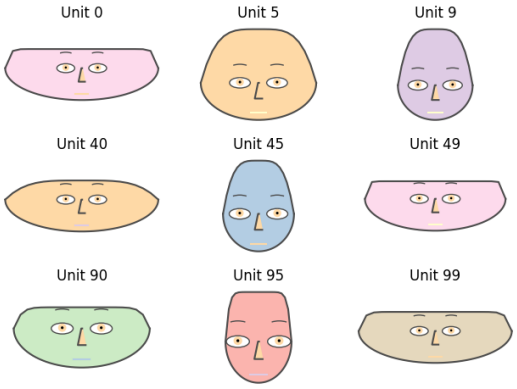
3.1.2 Chernoff and Radarplot

To compare the chernoff plot, radar plot, boxplot, and violin plot, we chose to use only nine elements, three from the first row (0, 5, and 9), the middle (50, 55, 59), and the last (90, 95, 99).

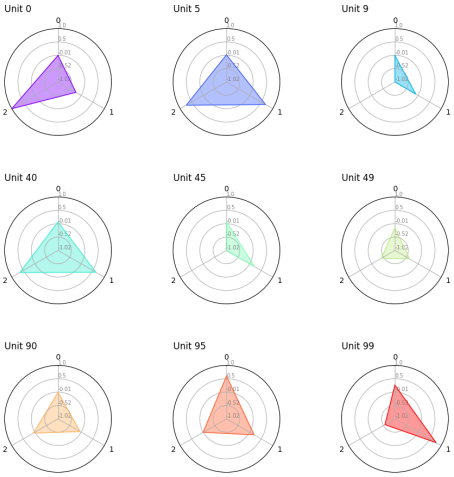
Chainlink dataset



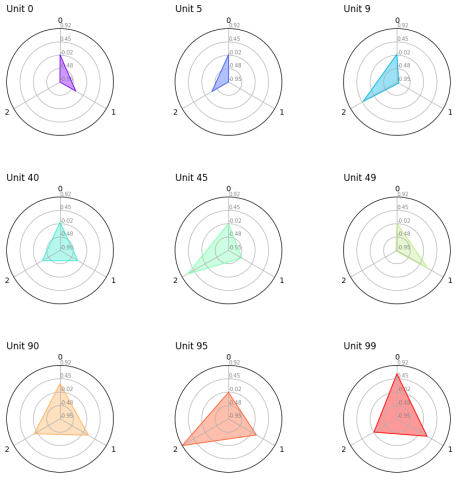
(a) Chernoffplot (All 3 attributes) for trained map on chainlink dataset.



(b) Chernoffplot (All 3 attributes) for untrained chainlink dataset.

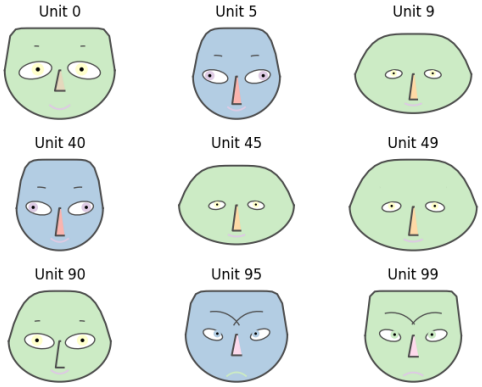


(a) Radarplot (All 3 attributes) for trained map on chainlink dataset.

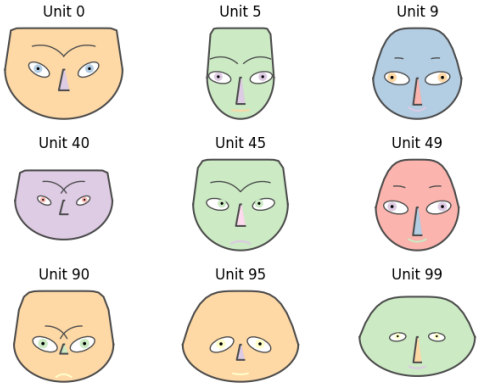


(b) Radarplot (All 3 attributes) for untrained chainlink dataset.

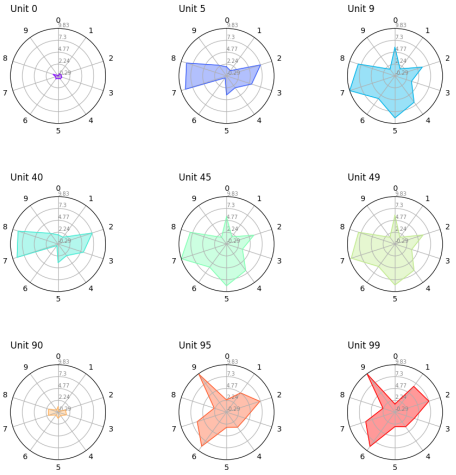
10 clusters dataset



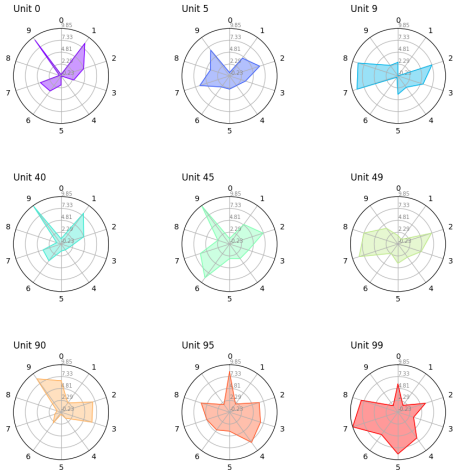
(a) Chernoffplot (All 10 attributes) for trained map on 10-clusters dataset.



(b) Chernoffplot (All 10 attributes) for untrained 10-clusters dataset.



(a) Radarplot (All 10 attributes) for trained map on 10-clusters dataset.

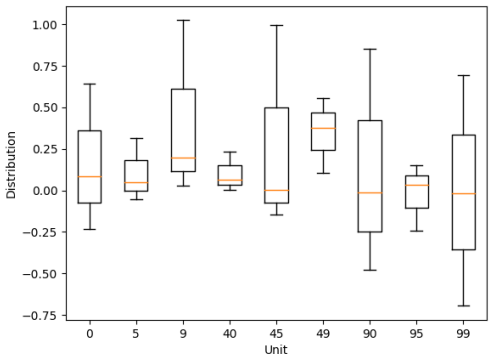


(b) Radarplot (All 10 attributes) for untrained 10-clusters dataset.

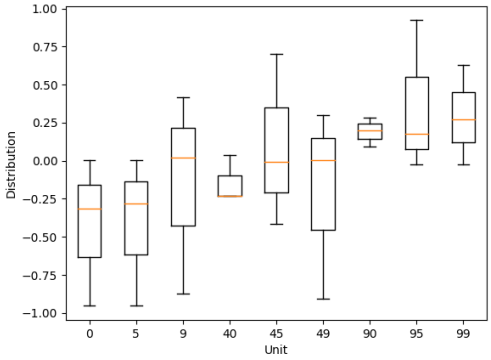
3.1.3 Boxplot and Violinplot

Due to the similarities between these two types of plots, we decided to compare them and, for that, we chose to use 9 different units and all the attributes.

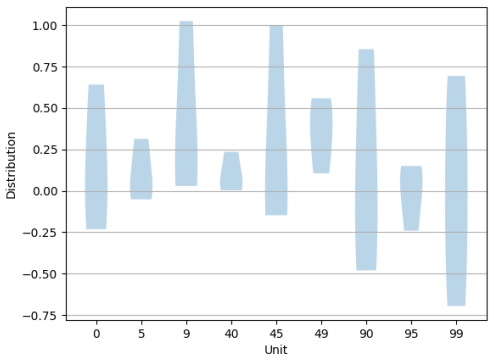
Chainlink dataset



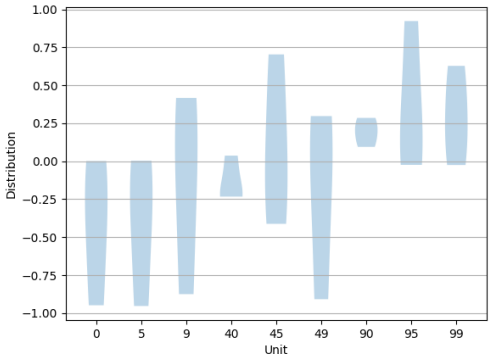
(a) Boxplot (All 3 attributes) for trained map on chainlink dataset.



(b) Boxplot (All 3 attributes) for untrained chainlink dataset.



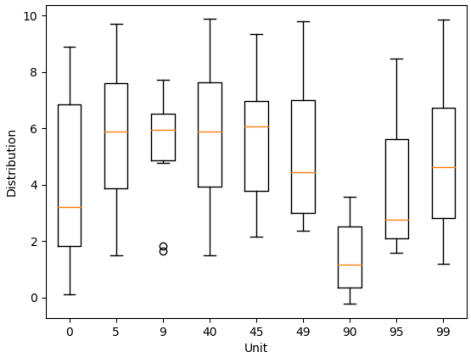
(a) Violin (All 3 attributes) for trained map on chailink dataset.



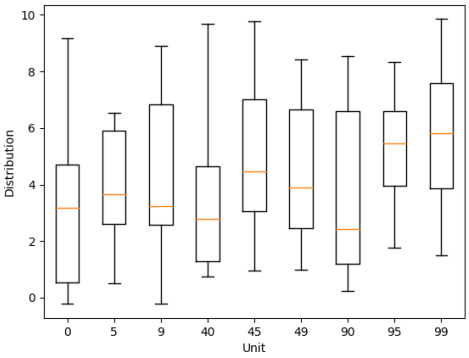
(b) Violin (All 3 attributes) for untrained chailink dataset.

In these 4 plots, we can see that in the trained maps that some of the units have their attributes weights more concentrated while in the untrained most have a wide range of weights.

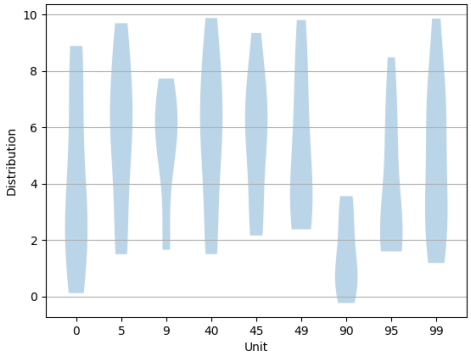
10 clusters dataset



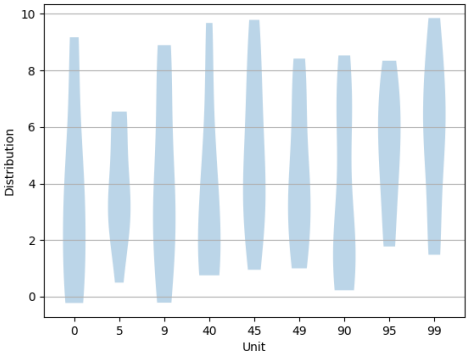
(a) Boxplot (All 10 attributes) for trained map on 10-clusters dataset.



(b) Boxplot (All 10 attributes) for untrained 10-clusters dataset



(a) Violin (All 10 attributes) for trained map on 10-clusters dataset.



(b) Violin (All 10 attributes) for untrained 10-clusters dataset.

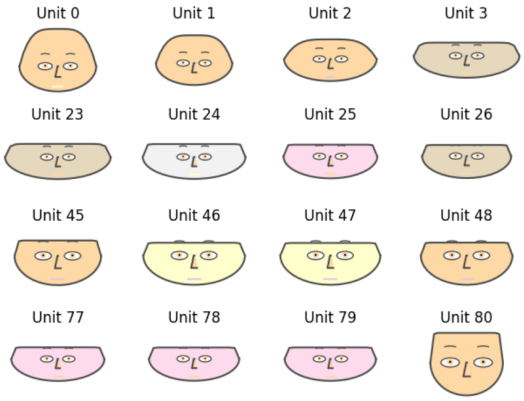
Looking at the 4 plots above we can see the similarities between boxplots and violin plots and, in the trained ones, we can see how outliers are shown in both cases, and how the trained map has units with more distinct attributes.

3.2 Comparison according to the map size

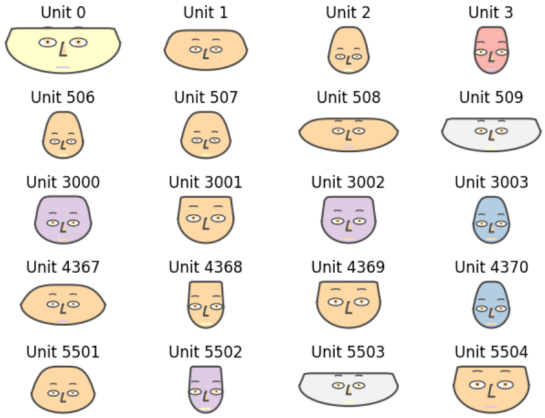
We trained for both datasets a small map (10x10) and a large map (100x60). This section aims to compare the visualization attributes for different map sizes.

Figures 16a and 16b present the chernoff plots for select units of a 10x10 and a 100x60 maps for the chainlink dataset and Figures 17a and 17b for the 10-clusters dataset. On each row, we have the attributes representation for four consecutive units. We can observe similarities between consecutive units, especially for small maps (Figures 16a and 16b).

Chainlink dataset

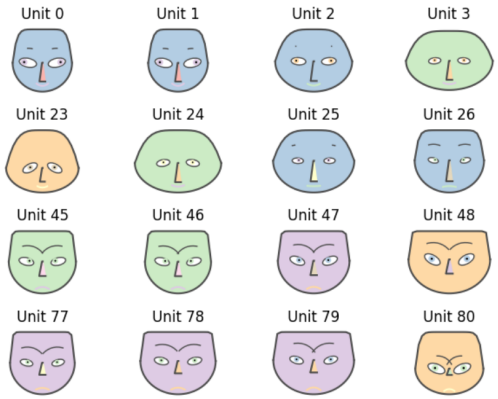


(a) Chernoff for selected units of a small trained map (10x10) on thechainlink dataset.

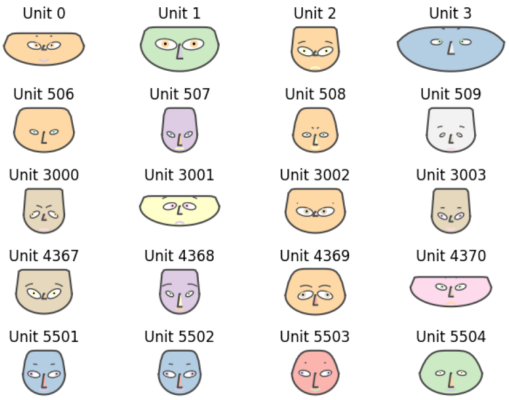


(b) Chernoff for selected units of a large trained map (100x60) on the chainlink dataset.

10-clusters dataset



(a) Chernoff for selected units of a small trained map (10x10) on 10 clusters dataset.



(b) Chernoff for selected units of alarge trained map (100x60) on 10 clusters dataset.

We noticed that attributes for consecutive units are closer for the small map than for the large map. That could be explained be the fact that, in the large map, there are many more units (6 000) than the number of initial data (1 000 in the chainlink dataset / 850 in the 10 clusters dataset).