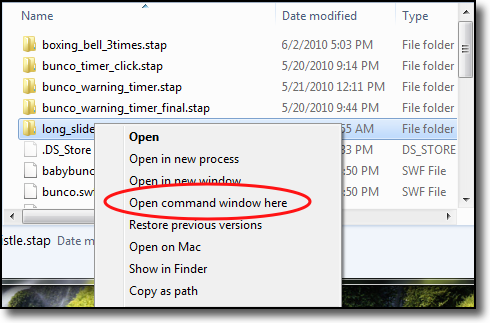
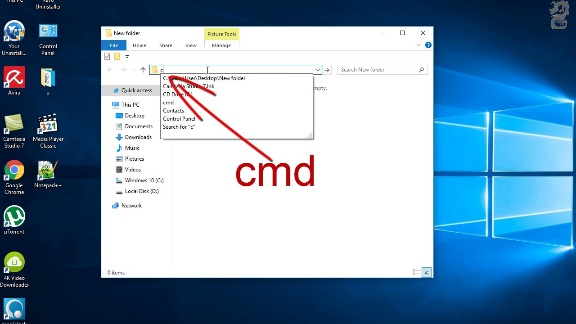
Computational Biology - EX3 – SOM

Eidelman Elli – 306679499

In this task I’ve implemented SOM – Self Organization Map, an un-supervised ANN which help to get a fast glimpse of how the data can be clustered or grouped without any prior knowledge.

The task, writing in python, runs from command line, download it from my GitHub:

Extract the zip, step in folder, right mouse click and choose open command prompt, or in window navigation bar of that folder type cmd

On the cmd window type main.exe

It can also receive the following arguments:

*optional arguments:*

*-h, --help show this help message and exit*

*-f FILE, --file FILE CSV to work with*

*-e EPOCHS, --epochs EPOCHS num of epochs to do*

*--learning\_rate LEARNING\_RATE*

*--radius RADIUS spread of neighborhood*

*--party {3,4,5,6,7,8,9,10,11,12,13,14,15}*

*(3, 'Labour'), (4, 'Yamina'), (5, 'Yahadot Hatora'),*

*(6, 'The Joint Party'), (7, 'Zionut Datit'), (8, 'Kachul Lavan'), (9, 'Israel Betinu'), (10, 'Licod'),*

*(11, 'Merez'), (12, 'Raam'), (13, 'Yesh Atid'), (14, 'Shas'), (15, 'Tikva Hadasha')*

Then I will parse the given csv file, and make a dictionary where the key is the economic cluster and the values are or a vector of all votes per party divided by all votes per settlement, or, if chosen a specific party normalized by all votes per settlement.

Text

Description automatically generated

Then the network will train for num ber of given epochs where the default is 50k, I tried to increase the number to 1M and didn’t saw any improvements the inside of the beehive was a bit different, but the pattern looks the same decreasing this number did affect the training and I got more and more hexagons without any mapping.

Chart

Description automatically generatedChart, bar chart

Description automatically generatedChart, bar chart

Description automatically generatedText

Description automatically generated

After Training session, I’m testing the network on a random 10% of the input data (which I didn’t trained on) and trying to predict the economic cluster of given examples.

When choosing to train only on one party it seems that the network gave me a good glimpse of real life where one can assume that settlements that voted to the Arab parties or the Haredim will come from lower income settlements while Merez voters for example will come from higher income settlements.

Chart, bar chart

Description automatically generated Chart

Description automatically generated

Choosing to train randomly on the data or by the order the data is in file doesn’t change too much the outcome, you can see that I depicted both results and the differences is noticeable just in the location of the color but the clusters are still sort of together just in different location across the honeycomb.

Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generated

Not Random

Random

The distance function between data and cell are L1 norm between the two vectors can be found here:

Text

Description automatically generated

I didn’t try other distance functions; the update of the neighborhood was Gaussian as been asked in exercise where I’m decaying the values for update as the neighbor far from winning cell. I didn’t try to use other functions although I saw usages across the internet of using Mexican-Hat but reported that it didn’t improve the results.

In conclusion:

SOM is a great way to get a good glance and visualize how our data can be summarized and grouped when the data is finite i.e. number of clusters is known. Without any a priori intel I’ve managed to get a nice correlation between the how many people voted to a specific party to their financial status.