

CO542 Neural Networks and Fuzzy Systems

Lab 06 - Self Organizing Maps (SOM)

Department of Computer Engineering, Faculty of Engineering, University of Peradeniya

November 03, 2023

1 Objectives

- Implement and train Self Organizing Maps (SOM) using python libraries.
- Identify the usage of SOM for real world applications.
- Practice different data visualizations associated with SOM.

2 Self-Organizing Maps

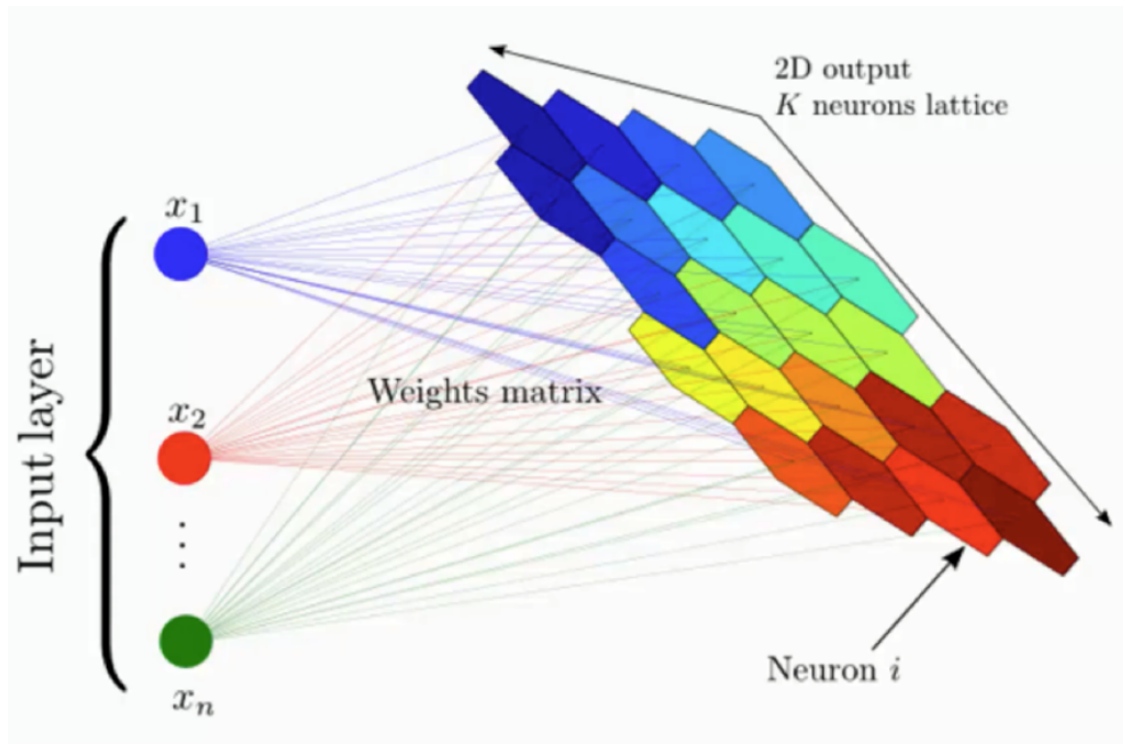


Figure 1: Self-organizing map (Source: <https://www.superdatascience.com/blogs/the-ultimate-guide-to-self-organizing-maps-soms>)

3 Minisom

MiniSom is a minimalistic and Numpy based implementation of the Self Organizing Maps (SOM). SOM is a type of Artificial Neural Network that is able to convert complex, nonlinear statistical relationships between high-dimensional data items into simple geometric relationships on a low-dimensional display. Minisom is

designed to allow researchers to easily build on top of it and to give students the ability to quickly grasp its details.

3.1 Installation

- *Using pip*

```
pip install minisom
```

- Or Download MiniSom to a directory of your choice and use the setup script

```
git clone https://github.com/JustGlowing/minisom.git  
python setup.py install
```

3.2 Usage

- In order to use MiniSom you need your data organized as a Numpy matrix where each row corresponds to an observation or as list of lists like the following:

```
data = [[ 0.80, 0.55, 0.22, 0.03],  
        [ 0.82, 0.50, 0.23, 0.03],  
        [ 0.80, 0.54, 0.22, 0.03],  
        [ 0.80, 0.53, 0.26, 0.03],  
        [ 0.79, 0.56, 0.22, 0.03],  
        [ 0.75, 0.60, 0.25, 0.03],  
        [ 0.77, 0.59, 0.22, 0.03]]
```

- Then you can train MiniSom as follows:

```
from minisom import MiniSom  
#initialization of 6x6 SOM  
som = MiniSom(6, 6, 4, sigma=0.3, learning_rate=0.5)  
# trains the SOM with 100 iterations  
som.train(data, 100)
```

- You can obtain the position of the winning neuron on the map for a given sample as follows (E.g. For sample data [0]) :

```
som.winner (data [0] )
```

For more information refer to the,

- GitHub - JustGlowing/minisom: MiniSom is a minimalistic implementation of the Self Organizing Maps.
- minisom/examples at master · JustGlowing/minisom · GitHub (Provides an overview of all the features implemented in minisom. Important: Follow these examples as a reference to complete the following tasks).

4 Exercise

The Pima Indians Diabetes Database contains several medical predictor variables and one target variable, 'Outcome'. Predictor variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

1. Load the data set as a pandas' data frame and generate the feature matrix by selecting the appropriate features in the data set.
2. Use `sklearn.preprocessing.MinMaxScaler` normalize the data between 0 and 1.
3. Initialize the weights using `MiniSom.pca_weights_init` function (Also identify the usage of `random_weights_init` method and mention an advantage of using `pca_weights_init`).
4. Identify the difference between `Minisom.train_batch` and `Minisom.train_random` methods available in `minisom` and use one of the models to train your self-organizing map.
5. Study about the distance map and mention what does it indicate related to the self-organizing maps.
6. Use `Minisom.distance_map` function to visualize the results obtained from the training process and use appropriate markers to indicate individual samples matched into each cell (according to the classes; patient who has diabetes/ do not have diabetes define two markers).
7. Generate another grid to indicate how often each neuron is activated using, `MiniSom.activation_response`.
8. Visualize the proportion of samples per class falling in a specific neuron using the, `matplotlib.gridspec.GridSpec` and `matplotlib.patches.Patch`

For further study

[Dynamic self-organizing maps with controlled growth for knowledge discovery. IEEE Transactions on Neural Networks](#)

5 Submission

Submit a single folder containing,

- All the required .py files or. ipynb files.
- A PDF containing all required outputs (E.g., Graphs, maps, etc...) and explanations before the deadline (announced in the course page).

Note: Please submit all the files (pdfs, code files etc.) in a single zipped file renamed as EXXYYY_Lab06, where XX is your batch ID, and YYY is your E No.

6 Reference

- GitHub - JustGlowing/minisom: MiniSom is a minimalistic implementation of the Self Organizing Maps