Implementation of a Self-Organizing Map

Damien DELPY

ENSEIRB-MATMECA

November 1, 2023

OVERVIEW

What is a SOM ?

Definition

It is a neural network of just one layer : the output layer.

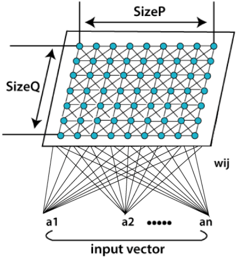


Figure: SOM Architecture (see 2 of Bibliography).

Wikipedia

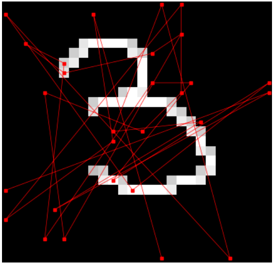
A self-organizing map (SOM) is used to produce a low-dimensional (typically two-dimensional) representation of a higher dimensional data set, while preserving the topological structure of the data.

Motivation

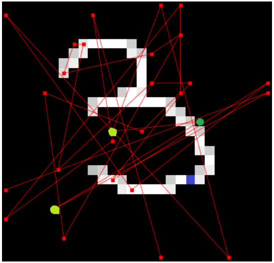
The Self-Organizing Maps permit to :

- Analyse and visualise the data. It represents complex data on a map of only two or three dimensions (see Convergence slide).
- Detect patterns from the data. Clustering (see K-means slide).
- Improve a deep neuronal network by sorting the data at the beginning.

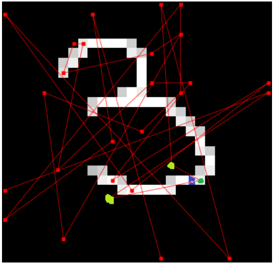
Example



Step 1



Step 2



Step 3

1. Initialize the weight vectors (randomly or not) in red.
2. **Competition** : Select a data vector(blue), then chose the closest weight vector(green) to it.
3. **Adaptation** : Update the winner and its neighbors (all green ones). The neighbors share a single link with the winner.
4. repeat the process till reach max_iteration.

Similarities with the Perceptron

Perceptron

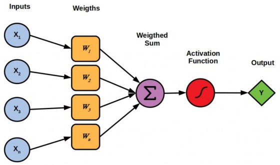


Figure: Diagram of a Perceptron.

It is also a one-layer neuronal network. However, this one is used to separate two different classes. The output is actually a binary one.

This is a supervised learning algorithm.

SOM

The SOM can gather vectors due to their similarities.

The SOM is an **unsupervised learning algorithm**.

Similarities with K-means algorithm

K-means algorithm is an unsupervised learning technique that can automatically gather data by creating **clusters**, which are subsets of data elements that share common characteristics.

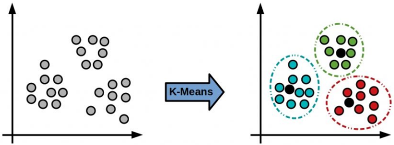


Figure: Process of K-means algorithm

The user must define the number of clusters **K**. However, The SOM does not require this, it guesses the right amount of clusters.

Convergence

The convergence of the SOM algorithm is not guaranteed (1). There are actually 2 errors that can appear.

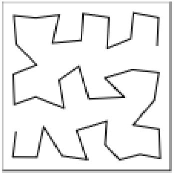


Figure: **Dimension Error**.
It happens when the number of neurons does not fit with the data.
The ideal number of neurons is $5\sqrt{N}$ where N is the number of data vectors.

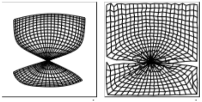


Figure: **Topological Error**.
It happens when a node is created. It looks like a butterfly.

ALGORITHM

Different Phases

Initialisation

The weights m_i or w_i of neuron connections are randomly initialized.

Competition

Chose randomly an input vector X_j.

Search for the neuron with the closest weight vector from X_j.

To realise it, a distance is needed.

All distances can work, let is choose the **Euclidean distance**.

The identity of the **winner** neuron is :

$$\mathrm{k} = \arg_i \min \|X_j - w_i\|$$

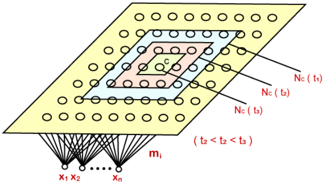


Figure 1. The structure of SOM Network

Cooperation

To prevent from discontinous adaptation by only modifying the winner neuron, all of its neighbors are also modified.

The number of neighbors depends on the topology.

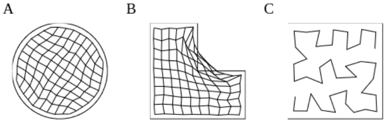


Figure: **Different topologies possible**. A : circular topology; B : rectangular topology; C : linear topology

Here, the grid is the neural network.

Each node of the grid is a neuron.

On the linear topology, each neuron has a maximum number of 2 neighbors.

On the rectangular topology, each neuron has a maximum number of 8 neighbors.

Adaptation

Here is the update of the weight vectors of the winner neuron and its neighbors.

it is at the instant t :

$$w_k(\mathrm{t}+1) = w_k(\mathrm{t}) + \alpha . h_{kj}(\mathrm{t}).(X_j \text{ - } w_k(\mathrm{t}))$$

- X_j is the input vector
- $w_k(\mathrm{t})$ is the weight vector of the neuron k at the instant t
- α is the **learning rate**
- h_{kj} is the **neighborhood** function which smooths the update of the neighbors depending on their proximity with the winner and the iteration t. (see 1 of Bibliography)

We can choose the **Gaussian** function :

$$h_{jk} = \exp\left(\frac{-\| w_{winner} - w_k \|^2}{2\sigma^2}\right)$$

σ is the standard deviation of the Gaussian function (2). It could be a function which decreases with t.

Algorithm

Algorithm 1 Self-Organizing Map Algorithm

Require: input data **X**; maximum number of iterations **IMAX**; the neighborhood function **h**; lattice dimensions **k**x**l**

Ensure: the weight vectors **W**

Randomly initialize weight vectors w_i
for t ← 1 **to** IMAX **do**

    for sample **in** X **do**
        competition
        Find the neuron closest to the sample
        cooperation and adaptation
        Update this neuron and its neighbors
    end for
end for
return W

BIBLIOGRAPHY

1. Self-Organizing Maps - Teuvo Kohonen (2001)

2. https://www.baeldung.com/cs/som-algorithm

3. http://www.pspc.unige.it/ drivsco/Papers/VanHulle_Springer.pdf