Unsupervised Learning

Introduction

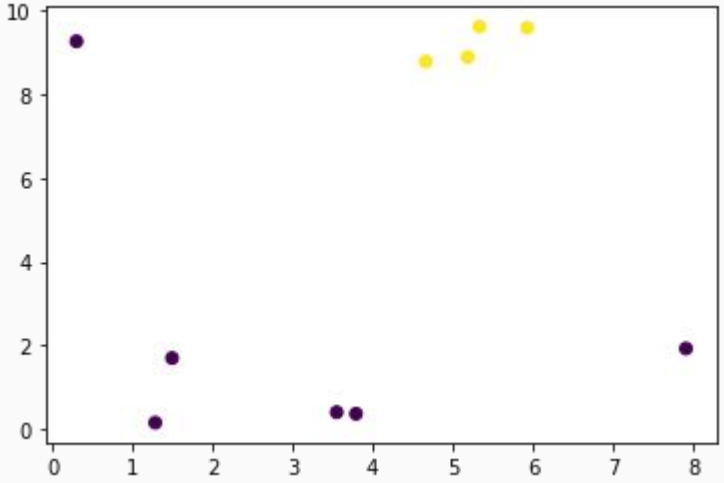
Unsupervised learning is a branch of machine learning in which we draw inferences from a data set which does not have any labels. That is, it has no idea of the result whatsoever.

We try to find patterns or structure within the data.

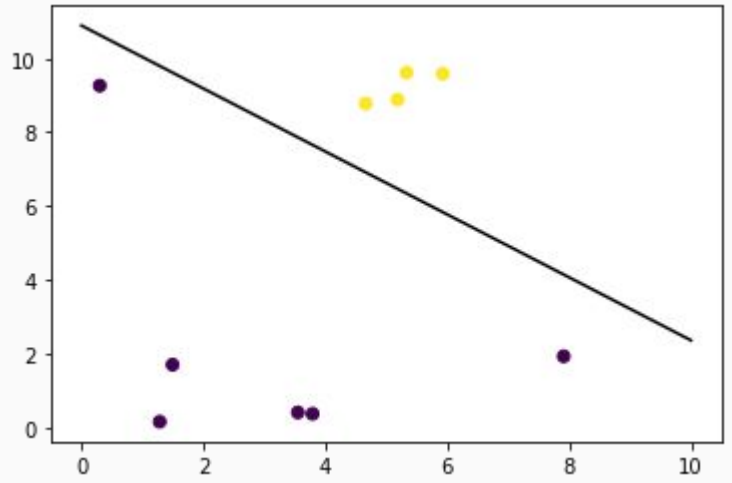
Mostly used for making groups (called clusters) of related data points. Other major use cases include Dimensionality Reduction (3D data to 2D etc), Anomaly Detection, Recommender Systems etc.

Supervised vs Unsupervised

Input for Supervised Learning

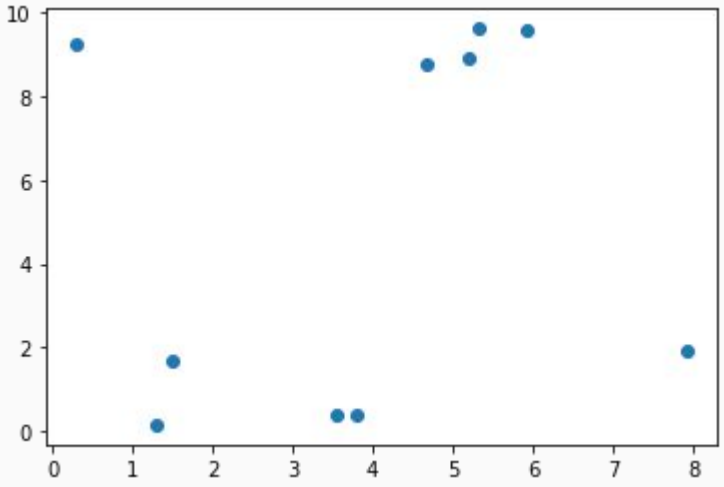


Output for Supervised Learning

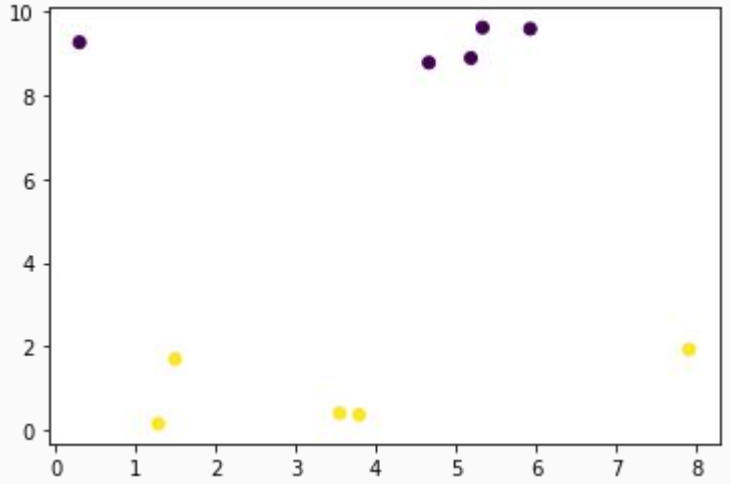


Supervised vs Unsupervised

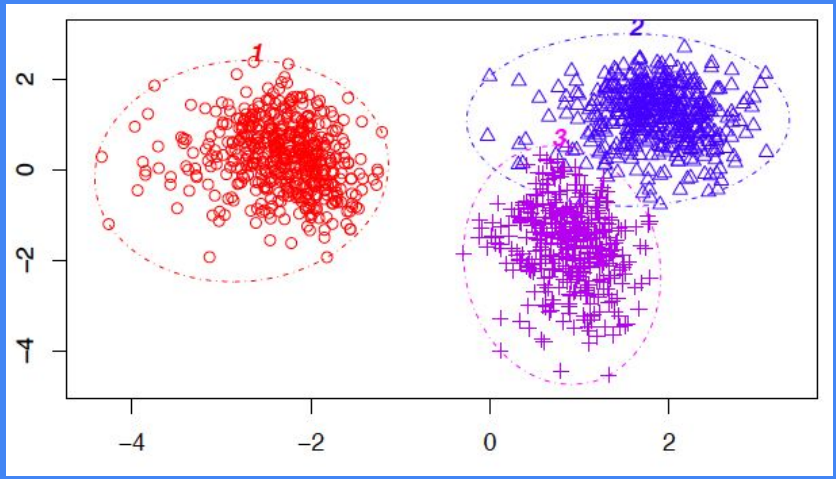
Input for Unsupervised Learning



Output for Unsupervised Learning



K-Means Algorithm



K-Means Algorithm (Clustering)

K-means algorithm identifies k number of centroids (number of classes/clusters), and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible.

Great for discovering underlying patterns in the dataset.

K is the number of centroids we want to find in the dataset. Later, we'll discuss ways to select K.

K Means Algorithm

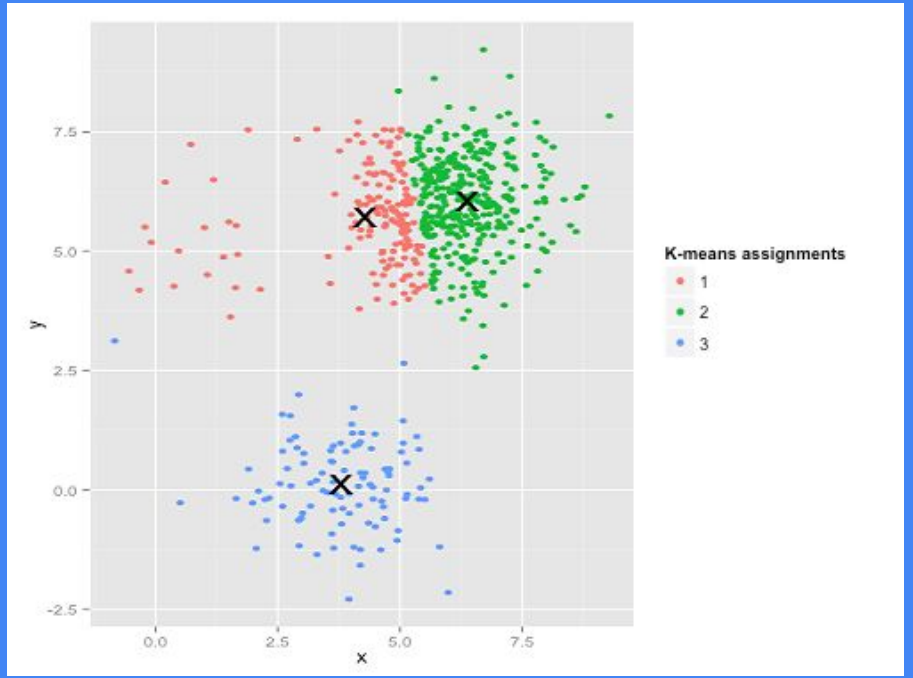
1. **Initialization :** Randomly initialize “K” centroids.
2. **Cluster Assignment :** Calculate distance between each centroid and datapoint. Assign the data point to the centroids group closest to it.
3. **Centroid Relocation :** For each centroid group, take average of all the points in a group. This is the new location of the centroid.

If the new location is not equal to the centroid location, repeat from step 2.

Does centroid initialization affect the clusters?

Mostly yes, depending on the dataset. Often random initialization of centroid works fine but sometimes it can get stuck on the local optimum.

Example



So, we need something to evaluate the classifier right?

Cost Function (also called Distortion Function)

This is not like the cost function in supervised learning where we compare the predictions to the labels.
We calculate the distance (distortion) of the data points from their cluster centroids.

- Say, we have “m” data points. $X^{(i)}$ is the i^{th} datapoint $i \in \{1,m\}$.
- We have “k” centroids. $C^{(j)}$ is the j^{th} datapoint $i \in \{1,k\}$.
- We have a vector μ, which represents the centroid a datapoint belongs to.
  EG : $\mu^{(3)} = 2$, means 3rd datapoint belongs to cluster number 2.

$$\textbf{Cost =: } J(C,\mu) = (1/m) * \sum_{i=1 \text{ to } m} ||X^{(i)} - \mu^{(i)}||^2$$

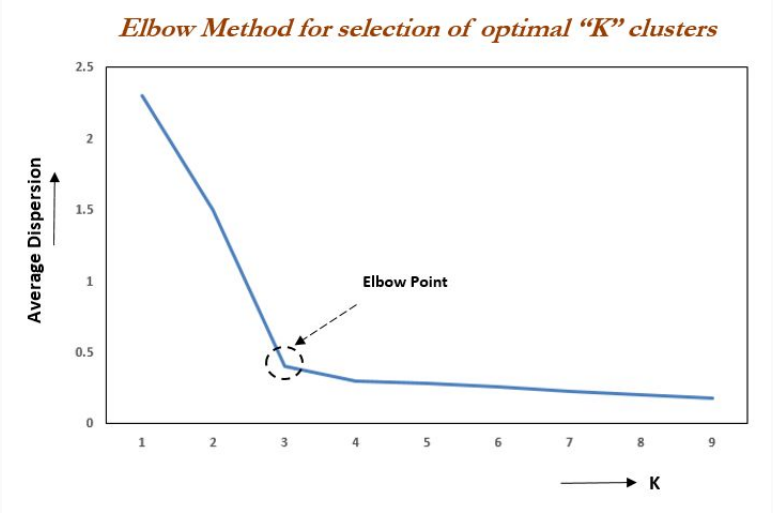
How to decide the value of K?

Elbow Method

It consists of plotting the distortion (cost) vs the number of clusters. We pick the elbow of the curve as the number of clusters to use.

We run it for K = 1,2,3….. and then plot something like,

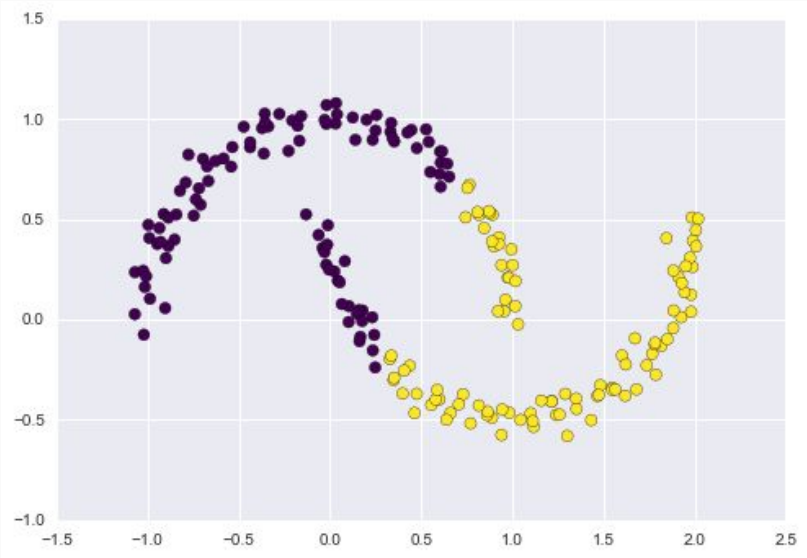
The elbow point is taken as the value for K.



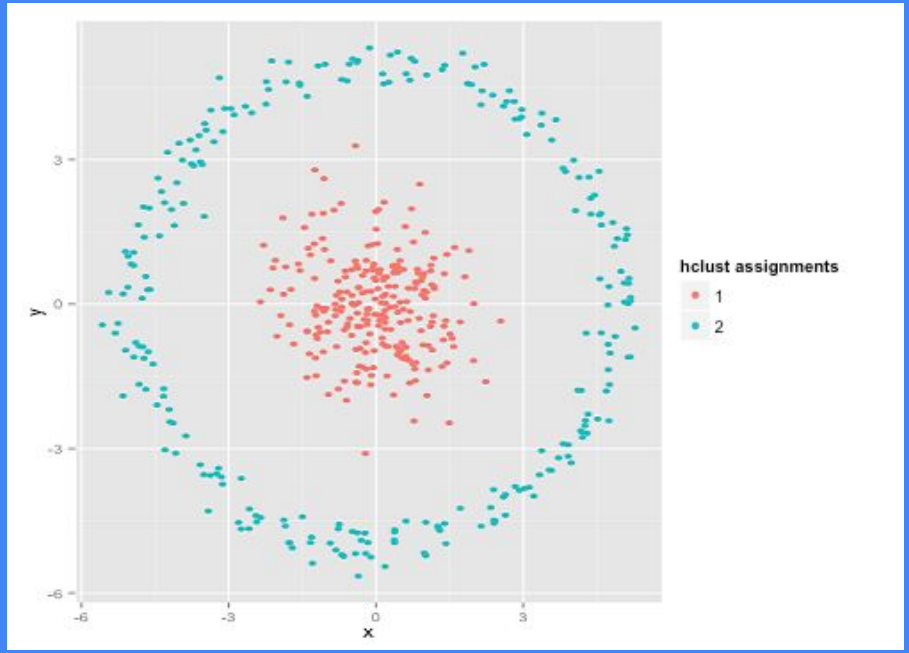
Drawbacks of K Means

- Works on linearly separable classes only
- We need to define “K” before hand
- Sensitive to outliers : An abnormal datapoint is also considered while making the clusters
- Performs hard clustering. Every data point is alloted a group. There is no confidence score. It can not perform well where data points belong to two classes.

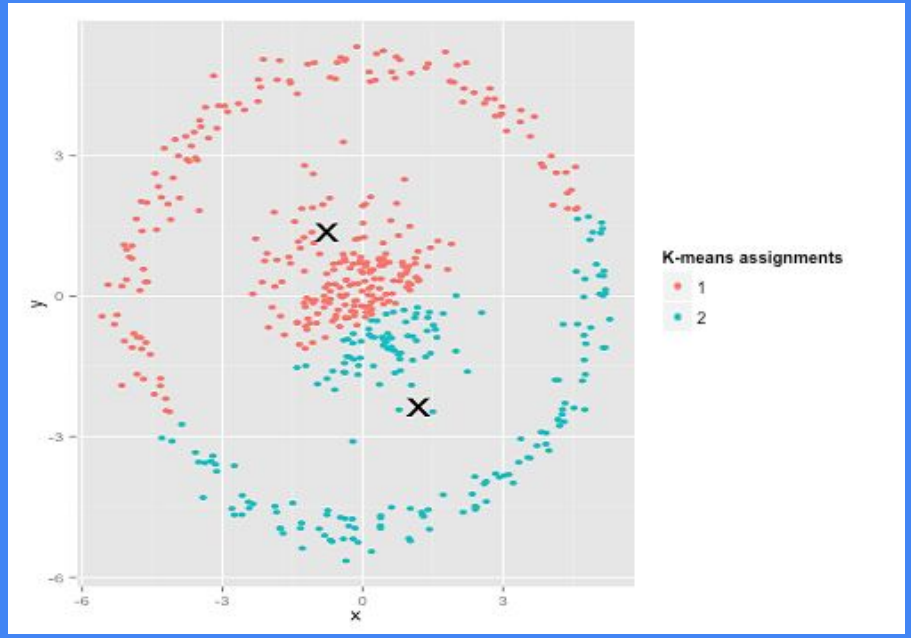
Linear boundary limitation



Desired output



K-Means stuck at local optima



Let’s code

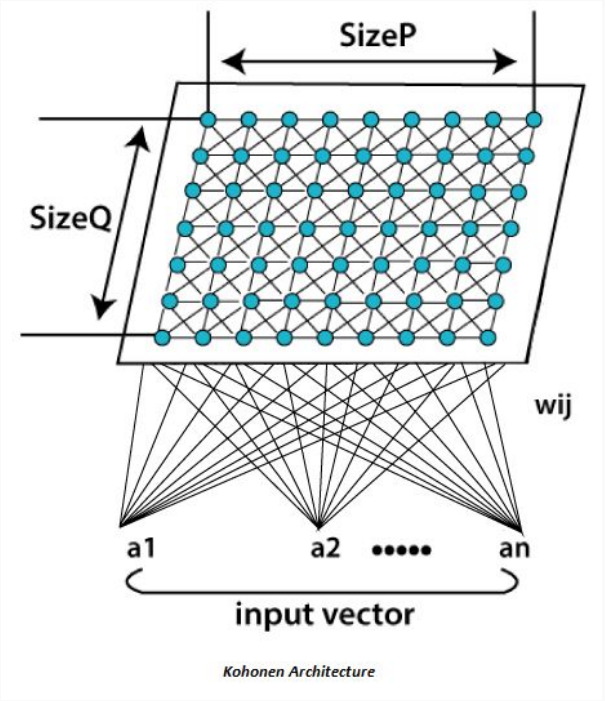
Self Organizing Maps

A competitive learning algorithm that finds clusters by finding the closest approximation to the inputs.

It’s a soft clustering algorithm, as opposed to K-Means which is a hard clustering algorithm.

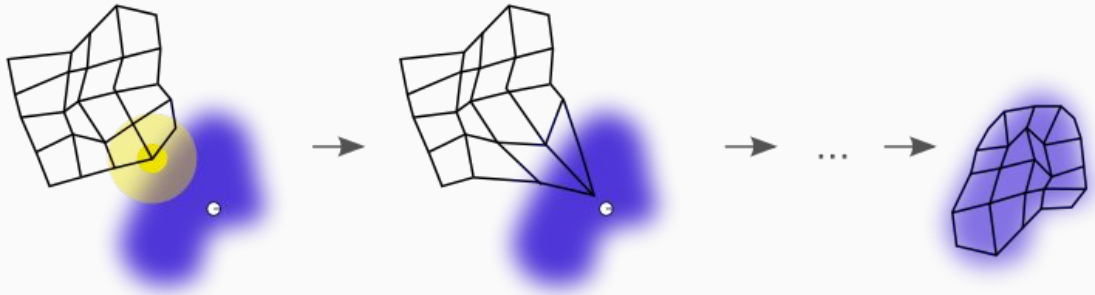
SOM Structure

SOMs can be 2D maps, 3D maps or even higher dimensional maps. Most practical use-cases make use of 2D maps.



Procedure

1. **Randomly initialize** weight vectors with small values.
2. **Compete** to find closest vector for any given input.
3. **Update weights** in the neighborhood of the BMU.

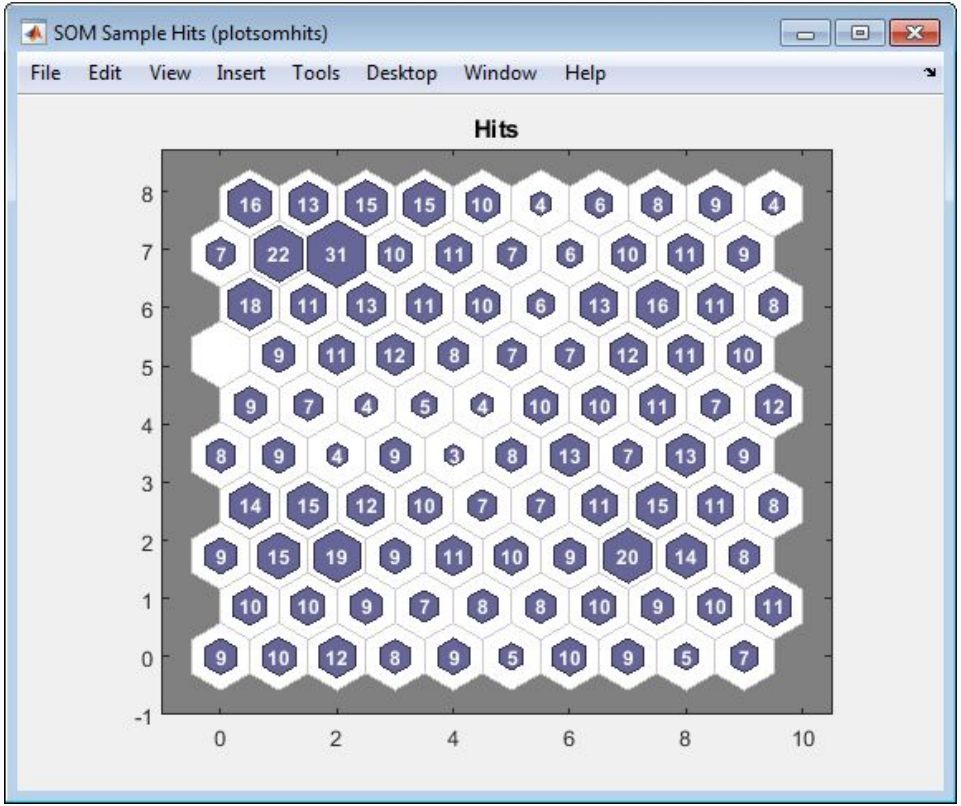


Weight update = Learning

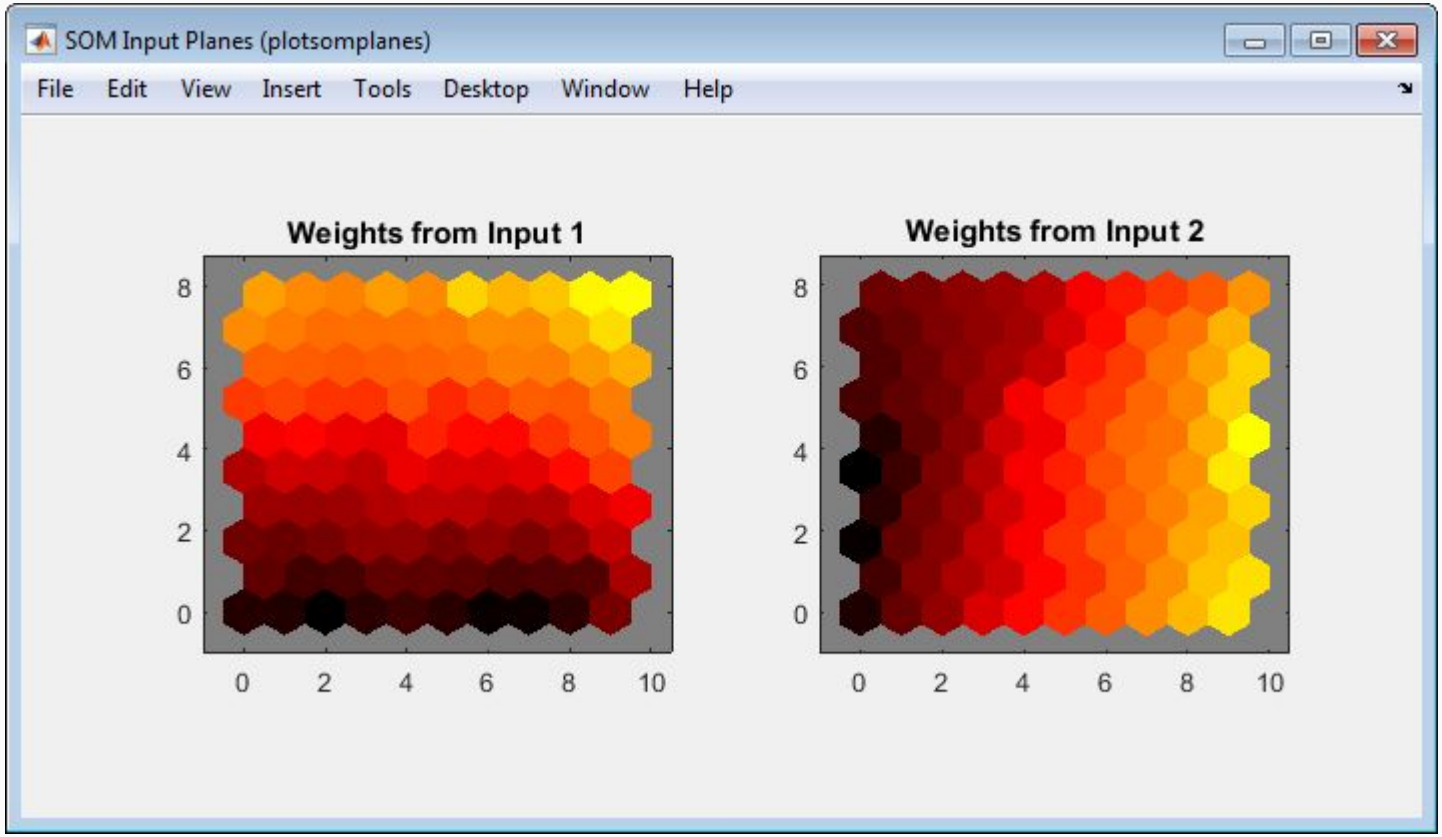
The process of how you update the weight vectors is crucial to the learning. The weight update is done by the following equation

$$W(v, n+1) = W(v, n) + \alpha * \Theta(u) * \{ D(t)-W(v, n) \}$$

Visualization



Weight updates



Resources

- K Means in depth
- Example of Supervised vs Unsupervised
- K Means Implementation
- Self Org Maps with PyTorch
- SOM Intuition