Neural Networks

Lecture 18

Unsupervised Learning Network (2)

Md. Mijanur Rahman, Prof. Dr.
Dept. of Computer Science and Engineering, Jatiya Kabi Kazi Nazrul Islam University, Bangladesh.
Web: www.mijanrahman.com | Email: mijanjkkniu@gmail.com; mijan@jkkniu.edu.bd

Chapter Contents

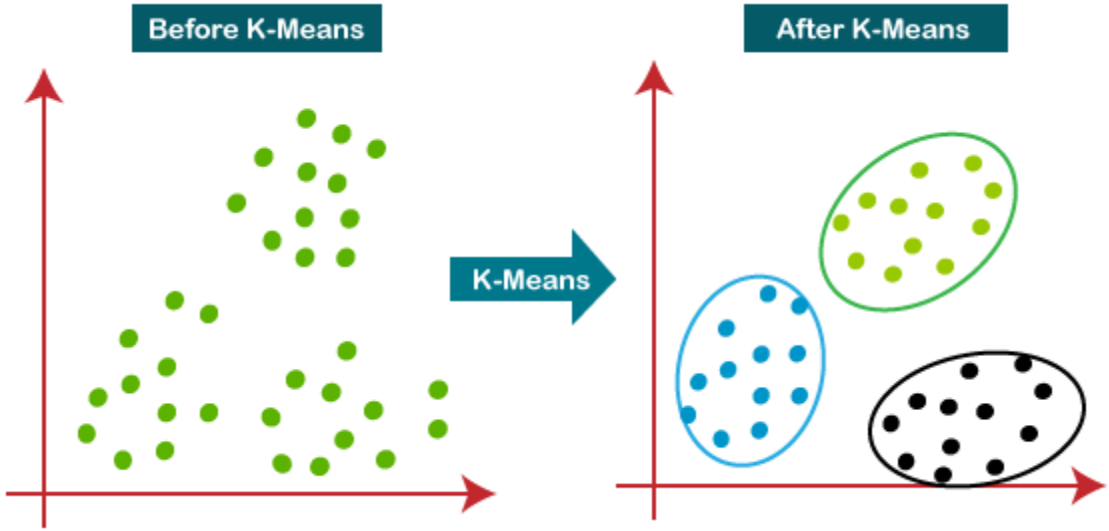
- **This chapter covers the following topics:**
  - Introduction to Unsupervised Learning Networks
  - Hamming Network
  - Max Net
  - Competitive Learning in ANN
  - **K-means Clustering Algorithm**
  - **Self- Organizing Feature Map**

K-means Clustering Algorithm…

- K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning.
- K-Means Clustering groups the unlabeled dataset into different clusters. It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.
- The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.
- The goal of clustering is to determine the intrinsic grouping in a set of unlabelled data.

K-means Clustering Algorithm…

- The k-means clustering algorithm mainly performs two tasks:
  1. Determines the best value for K center points or centroids by an iterative process.
  2. Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.
- The below diagram explains the working of the K-means Clustering Algorithm:



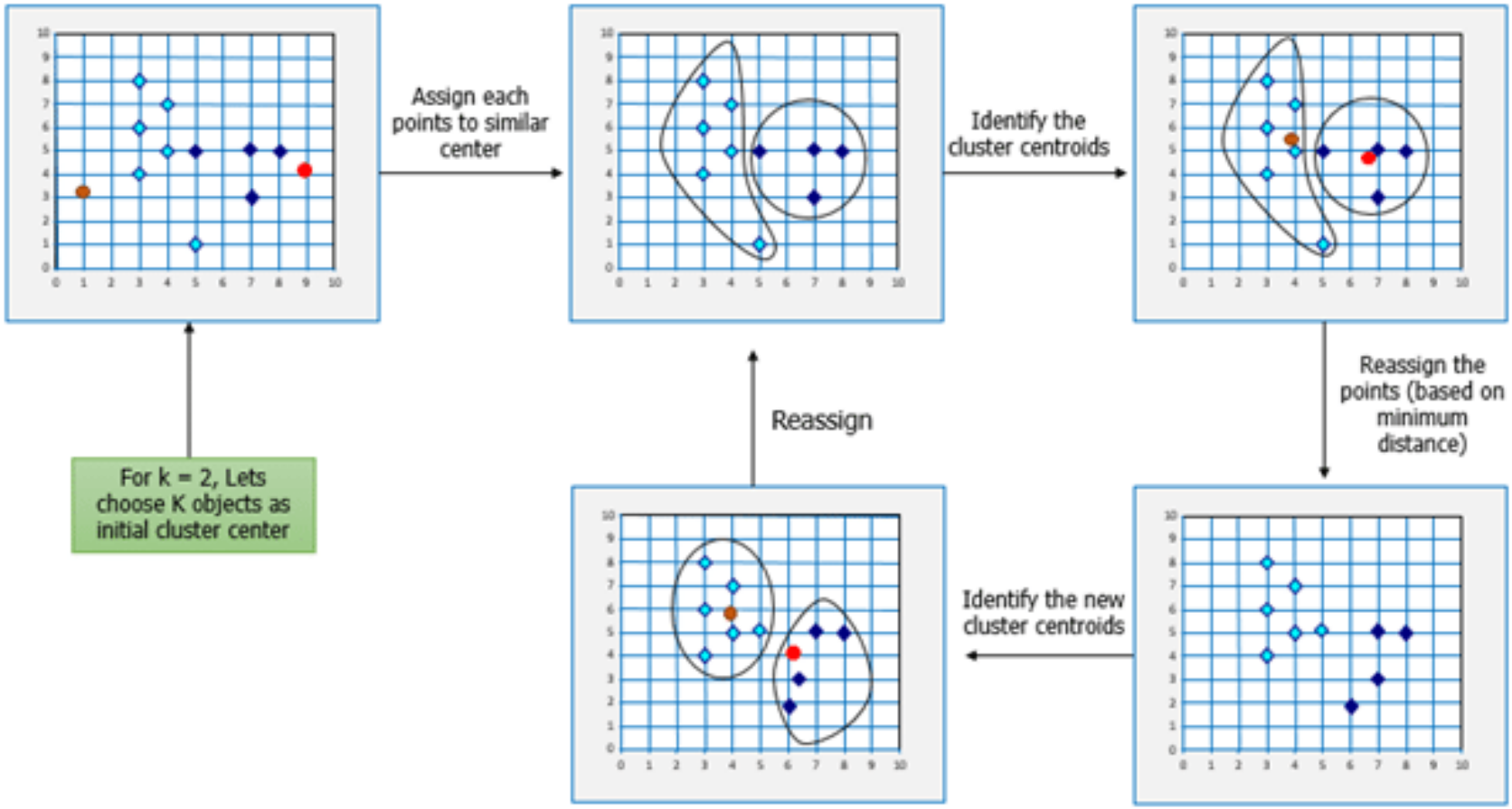
K-means Clustering Algorithm…

How does the K-Means Algorithm Work?

- The working of the K-Means algorithm is explained in the below steps:
  - **Step-1:** Select the number K to decide the number of clusters.
  - **Step-2:** Select random K points or centroids. (It can be other from the input dataset).
  - **Step-3:** Assign each data point to their closest centroid, which will form the predefined K clusters.
  - **Step-4:** Calculate the variance and place a new centroid of each cluster.
  - **Step-5:** Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.
  - **Step-6:** If any reassignment occurs, then go to step-4 else go to FINISH.

K-means Clustering Algorithm

- **The step by step process:**



Self-Organizing Feature Maps (SOM)

- A paradigm of unsupervised learning neural networks, which maps an input space by its fixed topology and thus independently looks for similarities.
- Based on this principle and exploring the question of how biological neural networks organize themselves, Dr. Teuvo Kohonen developed the **self-organizing feature maps** (shortly referred to as self-organizing maps or SOMs) in 1982.
- This as a paradigm of neural networks where the output is the state of the network, which learns completely unsupervised, i.e. without a teacher.

Self-Organizing Feature Maps (SOM)

Characteristics:

- The notable characteristic of this algorithm is that the input vectors that are close — similar — in high dimensional space are also mapped to nearby nodes in the 2D space.

- Thus, it is in essence a method for dimensionality reduction, as it maps high-dimension inputs to a low (typically two) dimensional representation;
- And it conserves the underlying structure of its input space.

Self-Organizing Feature Maps (SOM)

Characteristics:

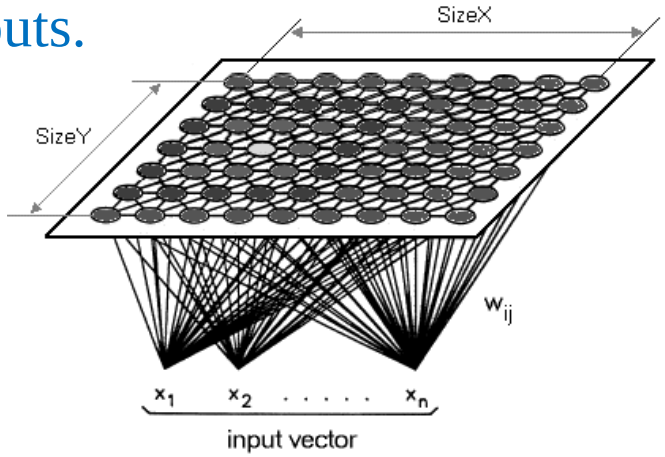
- A valuable detail is that the entire learning occurs without supervision i.e. the nodes are *self-organising*.
- They are also called *feature maps*, as they are essentially retraining the features of the input data, and simply grouping themselves according to the *similarity* between one another.

- This has a pragmatic value for visualizing complex or large quantities of high dimensional data and representing the relationship between them into a low, typically two-dimensional, field to see if the given unlabelled data has any structure to it.

Self-Organizing Feature Maps (SOM)

Kohonen Architecture:

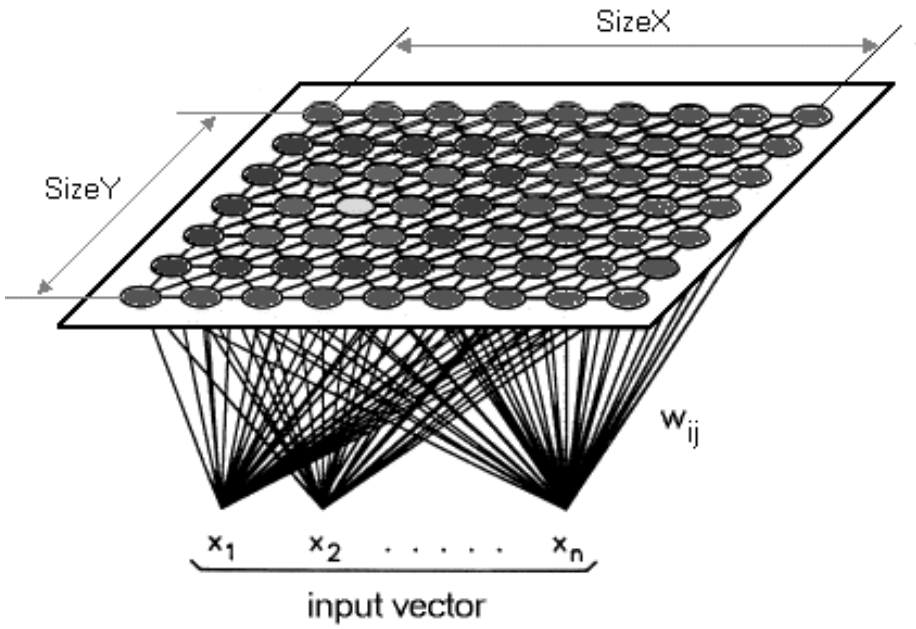
- A Self-Organizing Map (SOM) differs from typical ANNs both in its architecture and algorithmic properties.
- Firstly, its structure comprises of a single-layer linear 2D grid of neurons, instead of a series of layers.
- All the nodes on this grid are connected directly to the input vector, *but not to one another*, meaning the nodes do not know the values of their neighbours, and only update the weight of their connections as a function of the given inputs.



Self-Organizing Feature Maps (SOM)

Kohonen Architecture:

- **The grid *itself is the map* that organizes itself at each iteration as a function of the input of the input data.**
- As such, after clustering, each node has its own *(i,j)* coordinate, which allows one to calculate the Euclidean distance between 2 nodes by means of the **Pythagorean theorem**.

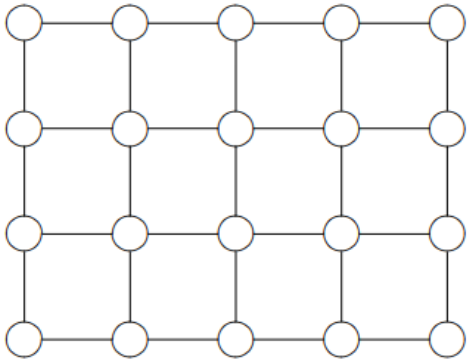


In mathematics, the Pythagorean theorem or Pythagoras' theorem **is a fundamental relation in Euclidean geometry between the three sides of a right triangle**.

Self-Organizing Feature Maps (SOM)

Kohonen Topology:

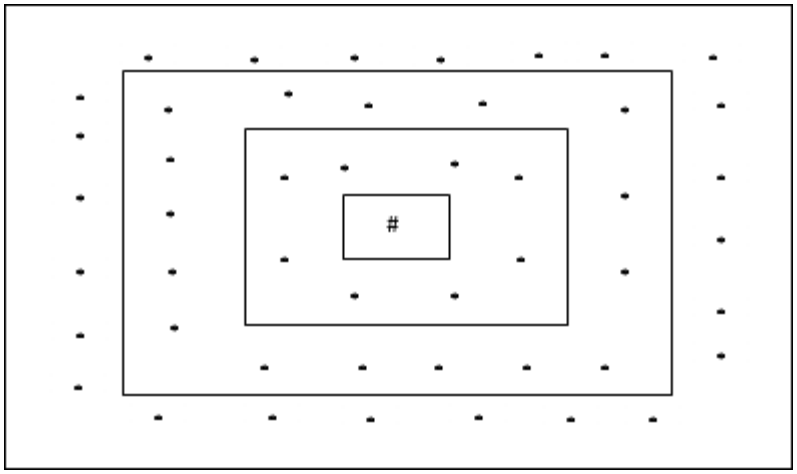
- The neurons are interconnected by neighborhood relationships. These neighborhood relationships are called **topology**.
- **The training of a SOM is highly influenced by the topology**. It is defined by the topology function ***h***(i, k, t), where i is the winner neuron, k the neuron to be adapted (which will be discussed later) and t the timestep. The dimension of the topology is referred to as G.
- Kohonen network's nodes can be in **a rectangular or hexagonal topology**.
- **The following are example topologies of a self-organizing map: a) a one-dimensional topology, and b) a two-dimensional one.**

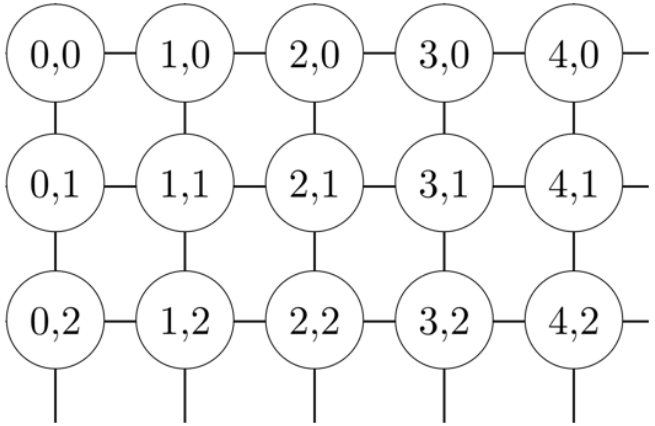


Self-Organizing Feature Maps (SOM)

Rectangular Grid Topology:

- This topology has 24 nodes in the distance-2 grid, 16 nodes in the distance-1 grid, and 8 nodes in the distance-0 grid, which means the difference between each rectangular grid is 8 nodes.
- The winning unit is indicated by #.

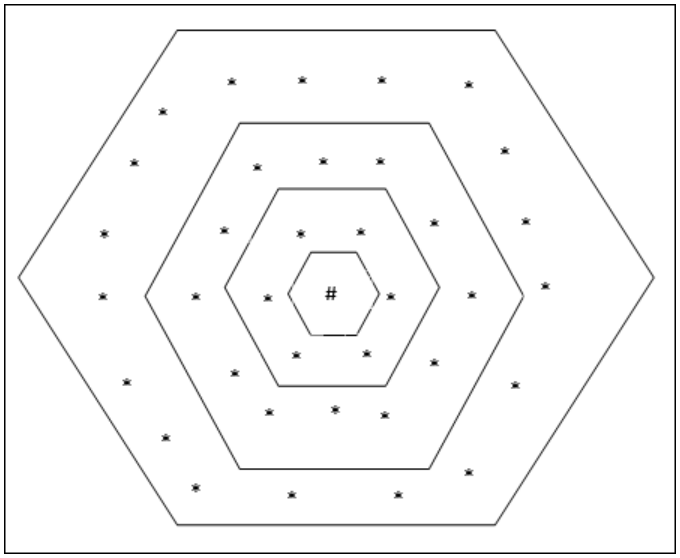


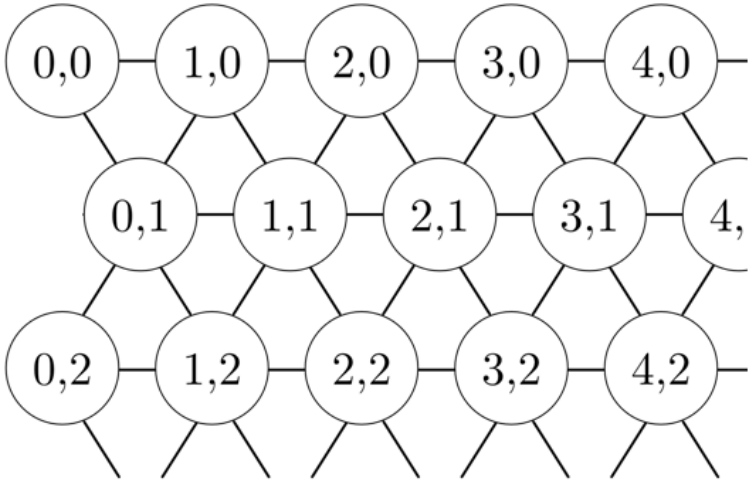


Self-Organizing Feature Maps (SOM)

Hexagonal Grid Topology:

- This topology has 18 nodes in the distance-2 grid, 12 nodes in the distance-1 grid, and 6 nodes in the distance-0 grid, which means the difference between each rectangular grid is 6 nodes.
- The winning unit is indicated by #.

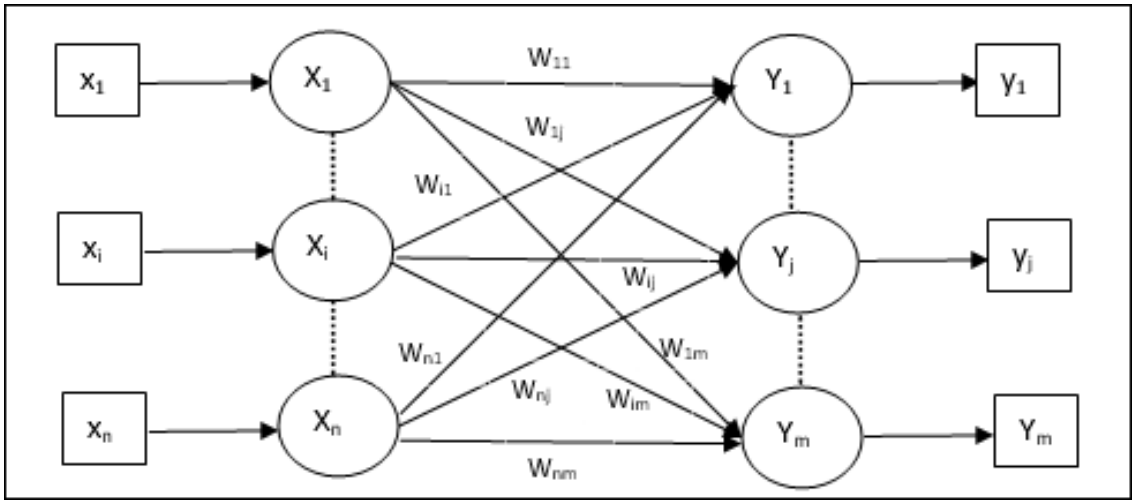




Self-Organizing Feature Maps (SOM)

KSOM Architecture:

- The architecture of KSOM or SOM is similar to that of the competitive network. With the help of neighborhood schemes, the training can take place over the extended region of the network.
- It has two layers: (i) input layer, and (ii) output layer. There are ***n*** units in the input layer, and ***m*** units in the output layer.



Self-Organizing Feature Maps (SOM)

Algorithm for Training:

- **[Training makes the SOM topology cover the input space]** The training of a SOM is as straightforward, and basically, it is structured into five steps.

(1) Initialization: The network starts with the selection of random values for the initial weights, w_{ij}.

(2) Sample Input Pattern: Take a sample training vector, $x = \{x_1, x_2, \ldots, x_n\}$ from the inputs. This training vector is entered into the network.

(3) Distance Measurement and Finding Winner Neuron: Find the winner neuron from the output layer which has the smallest distance to the input vector (i.e., weight vector closest to the input vector). It can be calculated by taking the Square of Euclidean distance for each output unit and find the output unit that has minimum Euclidean distance from the input vector.

Square of Euclidean Distance for **j = 1 to m**

$$D(j) = \sum_{i=1}^{n} \sum_{j=1}^{m} (x_i - w_{ij})^2$$

Self-Organizing Feature Maps (SOM)

Algorithm for Training:

(4) New Weight Calculation: Find the new weights between input vector sample and winning output unit (neuron), by the following relation:

$$w_{ij}(new) = w_{ij}(old) + \alpha[x_i - w_{ij}(old)]$$

Update the learning rate by the following relation:

$$\alpha(t + 1) = 0.5\alpha t$$

(5) Continuation: Repeat steps 2 to 4 until weight updation is negligible (i.e., new weights are similar to old weights) or feature map stop changing..

Assignment

Prepare an assignment on the following topics:

- **K-Means Clustering: Numerical Examples**
- **Self-Organizing Maps: Algorithms and Applications**

UNSUPERVISED LEARNING
NETWORK PARADIGMS

The End