Unit-5

Unsupervised Learning

Unsupervised learning can be thought of as self learning where the algorithm can find previously unknown patterns in datasets that does not have any sort of labels.

Example :- Suppose you and your friends want to watch the cricket match. but you do not know what cricket is. you don't know anything about cricket, but for your friends, you say yes. Then you start watching the match & analyzing the things. At the end you yourself reached at the Conclusions like

→ There are two teams

→ Different kind of players such as batters, bowlers & so forth.

→ Ball hits wickets or is caught, batsman is ou[t] and so on.

→ unsupervised learning find patterns which were previously unknown.

→ Patterns help in categorization or finding association.

→ They can detect anomalies & defects in the data.

→ They work on unlabeled data.

Types of unsupervised ~~Alg~~ learning

↓ Clustering

↓ Association

<u>Clustering</u> :- It is a type of ~~un~~ unsupervised learning where you find patterns in the data that you are working on. Pattern may be shape, size, Color etc which can be used to group data items to make clusters.

Association :- It is a type of unsupervised learning where you can find the ~~dependencies~~ dependencies of one data item to another data item. & mapped them such that they can help you profit better.

Eg:- Supermarket analysis.

Self Organization Model (SOM)

There are ~~a~~ number of different NN architectures specifically designed for clustering. The most widely known ~~a~~ is SOM.

A SOM is a neural network that has a set of neurons connected to form a topological grid. When some pattern is presented to an SOM, the neuron with closest weight vector is considered a winner & its weights are adapted to the pattern. In this way an SOM naturally finds data clusters.

The term self organization refers to the ability to learn & organize the information without being given the correct answers for input patterns.

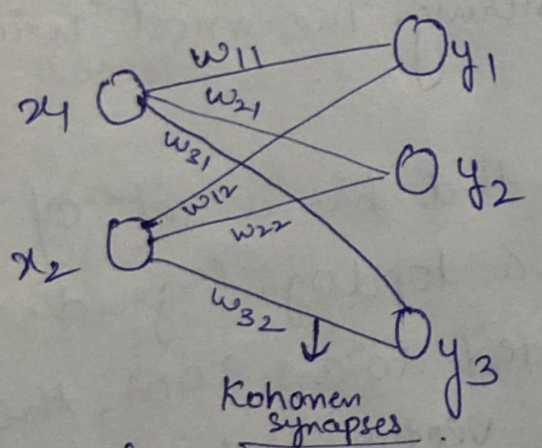
• It is unsupervised learning.

SOM operate in two modes : training and mapping. Training builds the map using input examples. It is a competitive process, also called vector quantization. Mapping automatically classifies a new input vector.

Kohonen Self Organizing Maps (KSOM)

→ It is unsupervised learning neural network

→ It is widely used to reduce dimensionality of input space preserving its topological structure.



→ A typical ~~to~~ kohonen SOM architecture consists of an input layer connected to a output layer (2-D kohonen layer) via Kohonen synapses.

→ Each ~~space~~ neuron in a Kohonen layer is associated with a unique set of co-ordinates in 2D space.

→ The input layer with 'n' input neurons is fed with n-dimensional input data one by one.

→ The output layer organizes itself to represent the inputs.

→ The objective of SOM training is to ensure that different parts of the network respond similarly to similar input vector, so the training mainly involves analysing the behaviour of the network for a training sample & adjusting the weights of Synapses to ensure that the network exhibits a similar behaviour for a similar input.

KSOM Algo

Step 1: Initialize the weights w_{ij} (Random values may be assumed). Initialize the learning rate 'α'.

Step 2: Calculate square of the Euclidean distance, i.e for each $j = 1$ to m.

$$D(j) = \underbrace{\sum_{i=1}^{n}}_{\text{no. of input vectors}} \underbrace{\sum_{j=1}^{m}}_{\text{no of clusters}} (\underbrace{x_i}_{\text{input}} - w_{ij})^2$$

w_{ij} → weight with which the input vector is associated with the clusters.

Step 3: Find ~~training~~ winning unit index J. So that $D(J)$ is minimum.

Step 4: For all units j within a specific neighbourhood of j & for all i, calculate new weights

Scanned with CamScanner

$$W_{ij}(new) = W_{ij}(old) + \alpha(x_i - W_{ij}(old))$$

After completing 1st round (EPOCH) update α.

Step 5: update learning rate 'α' using the formula

$$\cancel{\alpha(t+1) = 0.5\,\alpha(t)}$$

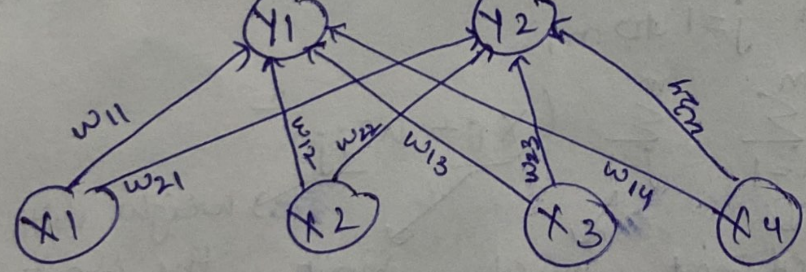
$$\alpha(t+1) = 0.5\,\alpha(t)$$

Numerical

Q. Construct KSOFM to cluster four given vectors [0 0 1 1], [1 0 0 0], [0 1 1 0] and [0 0 0 1], No of clusters to be formed is 2. Assume an initial learning rate of 0.5.

Solⁿ No of input vectors ; $n = 4$

No of clusters $m = 2$



$X1, X2, X3, X4$ = Input

Y_1, Y_2 = clusters.

Initialize weights randomly b/w 0 & 1.

$$W_{ij} = \begin{bmatrix} 0.2 & 0.9 \\ 0.4 & 0.7 \\ 0.6 & 0.5 \\ 0.8 & 0.3 \end{bmatrix}$$

weights of x_1, x_2, x_3, x_4 is connected to y_1

weights of x_1, x_2, x_3, x_4 is connected to y_2

for first input vector i.e

$X = [0 \ 0 \ 1 \ 1]$

(1) Calculate euclidean distance

$$D(j) = \sum (w_{ij} - x_i)^2 \qquad w_{ij} = \begin{bmatrix} 0.2 \to w_{11} \\ 0.4 \to w_{12} \\ 0.6 \to w_{13} \\ 0.8 \to w_{14} \end{bmatrix}$$

for cluster 1, $J=1$

$$D(1) = (0.2-0)^2 + (0.4-0)^2 + (0.6-1)^2 + (0.8-1)^2$$

$$= 0.04 + 0.16 + 0.16 + 0.04$$

$$= 0.4$$

$$w_{ij} = \begin{bmatrix} 0.9 \to w_{21} \\ 0.7 \to w_{22} \\ 0.5 \to w_{23} \\ 0.3 \to w_{24} \end{bmatrix}$$

for cluster 2 $D(2) = \sum (w_{ij} - x_i)^2$

$$= (0.9-0)^2 + (0.7-0)^2 + (0.5-1)^2 + (0.3-1)^2$$

$$= 0.81 + 0.49 + 0.25 + 0.49$$

$$= 2.04$$

(2) Choose the minimum (D(1)) so D(1) is the winning cluster, & now update its weights ($J=1$)

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

$$w_{11}(new) = w_{11}(old) + \alpha[x_i - w_{11}(old)]$$

$$= 0.2 + 0.5[0 - 0.2]$$

$$= 0.1$$

$$w_{12}(new) = w_{12}(old) + \alpha[x_i - w_{12}(old)]$$

$= 0.4 + 0.5(0 - 0.04)$

$= 0.2$

$w_{13}(new) = 0.6 + 0.5[1 - 0.6)$

$= 0.8$

$w_{14}(new) = 0.8 + 0.5(1 - 0.8)$

$= 0.9$

(4) updated weight Matrix

$$\begin{bmatrix} 0.1 & 0.9 \\ 0.2 & 0.7 \\ 0.8 & 0.5 \\ 0.9 & 0.3 \end{bmatrix}$$

↓ updated weight ↓ Previous weight

Take 2nd input vector

$x = [1 \ 0 \ 0 \ 0]$

(1) Calculate Euclidean distance

$D(1) = (0.1-1)^2 + (0.2-0)^2 + (0.8-0)^2 + (0.9-0)^2$

$= 2.3$

$D(2) = (0.9-1)^2 + (0.7-0)^2 + (0.5-0)^2 + (0.3-0)^2$

$= 0.84$

(2) $D(1) > D(2)$ so choose $D(2)$ & update weight

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

$w_{21} = 0.9 + 0.5(1-0.9)$
$= 0.95$

$w_{22} = 0.7 + 0.5(0-0.5)$
$= 0.35$

$w_{23} = 0.5 + 0.5(0-0.5)$
$= 0.25$

$w_{24} = 0.3 + 0.5(0-0.3)$
$= 0.15$

(4) updated weight matrix

$$\begin{bmatrix} 0.1 & 0.95 \\ 0.2 & 0.35 \\ 0.8 & 0.25 \\ 0.9 & 0.15 \end{bmatrix}$$

Take 3rd input vector

$x = [0\ 1\ 1\ 0]$

(1) Calculate Euclidean distance

$D(1) = (0.1-0)^2 + (0.2-1)^2 + (0.8-1)^2 + (0.9-0)^2$
$= 1.5$

$D(2) = (0.95-0)^2 + (0.35-1)^2 + (0.25-1)^2 + (0.15-0)^2$
$= 1.91$

(2) Choose winning cluster i.e ($D(1) < D(2)$)

$D(1)$ is " "

(3) weight updation

$w_{11} = 0.05$

$w_{12} = 0.6$

$w_{13} = 0.9$

$w_{14} = 0.45$

(4) updated weight matrix

$$\begin{bmatrix} 0.05 & 0.95 \\ 0.6 & 0.35 \\ 0.9 & 0.25 \\ 0.45 & 0.15 \end{bmatrix}$$

Fourth Input vector

$x = [0 \quad 0 \quad 0 \quad 1]$

(1) Calculate Euclidean distance

$D(1) = 1.475$, $D(2) = 1.81$

(2) winning cluster, $D(1)$

(3) update weight

$w_{11} = 0.025$

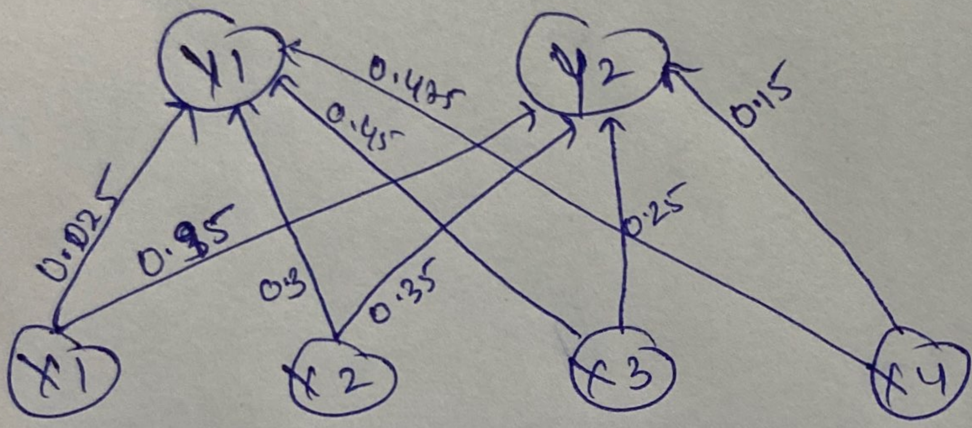
$w_{12} = 0.3$

$w_{13} = 0.45$

$w_{14} = 0.475$

④ updated Weight Matrix

$$W_{ij} = \begin{bmatrix} 0.025 & 0.95 \\ 0.3 & 0.35 \\ 0.45 & 0.25 \\ 0.475 & 0.15 \end{bmatrix}$$



Finally update the learning rate (α) & repeat all the steps again.

winner take all learning