- Kohonen self organizing networks
  - also known as Kohonen feature maps or topology-preserving maps.

  - competition based n/w paradigm for data clustering.

  - N/w of this type impose a neighborhood constraint on the o/p units, such that a certain topological property in the i/p data is reflected in the o/p unit's weights.

  - Similar to competative learning n/w.

  - For Kohonen feature maps, we update not only the winning unit's weights but also all of the weights in a neighborhood around the winning units. The neighborhood's size decreases slowly with each iteration.

Step 1: select the winning o/p unit as the one with the largest similarity measure (or smallest disimilarity measure) betn all weight vectors w_i & the i/p vector x.

If Euclidean distance is chosen as the dissimilarity measure, then the winning unit c satisfies the following eqn.

$$\|x - w_c\| = \min_i \|x - w_i\|,$$

where the index c refers to the winning unit

Step 2: Let NB_c denote a set of index corresponding to a neighborhood around winner c.

The weights of the winner & its neighboring units are then updated by

$$\Delta w_i = \eta (x - w_i), \quad i \in NB_c$$

where $\eta \Rightarrow$ small positive learning rate.

Neighborhood Fun^n $\Omega_c(i)$ arround winning unit c.

(Gaussian function)

$$\boxed{\Omega_c(i) = \exp\left(\frac{-\|p_i - p_c\|^2}{2\sigma^2}\right)}$$

where p_i & $p_c \Rightarrow$ positions of the o/p. units i & c.

$\sigma \Rightarrow$ scope of neighborhood.

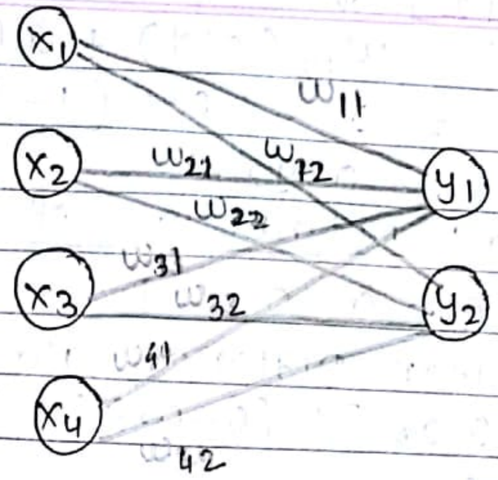
By using neighborhood function, updated formula can be rewritten as.

$$\boxed{\Delta w_i = \eta \Omega_c(i) (x - w_i)}$$

where $i \Rightarrow$ index for all o/p units

Kohonen's Self Organizing maps.

Steps	Action
1	Initialise weights, set max value for R, set learning rate α.
2.	While stopping condition false do steps 3 to 9
3.	For each i/p vector $\underline{x}$ do step 4 to 8.
4.	For each j neuron, compute the Euclidean distance. $D(j) = \sqrt{\sum_{i=1}^{n} (x_i - w_{ij})^2}$
5.	Find the index J such that D(J) is min.
6.	For all neurons j within a specified neighbourhood of J & for all i $w_{ij}(new) = w_{ij}(old) + \alpha(x_i - w_{ij}(old))$
7.	Update learning rate α. It is a decreasing function of the no of epochs.
8.	Reduce radius of topological neighbourhood at specified times.
9.	Test stopping condition. Typically this is a small value of the learning rate with which the weight updates are insignificant.



initial weight matrix.

$$\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \\ w_{41} & w_{42} \end{bmatrix}$$

consider simple e.g in which there are only 4 i/p training patterns

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

x_1	x_2	x_3	x_4
1	1	0	0
0	0	0	1
1	0	0	0
0	0	1	1

Let the learning time rate t+1 be given by

$$\alpha(t+1) = \frac{\alpha(t)}{2}$$

& suppose $\alpha(t=0) = 0.6$.

Let topological radius $R=0$.

- For vector 1100. ~~i.e. x_1~~

using Euclidean distance algorithm,

$$D(1) = (1-0.2)^2 + (1-0.6)^2 + (0-0.5)^2 + (0-0.9)^2$$
$$= 1.86$$

$$D(2) = (1-0.8)^2 + (1-0.4)^2 + (0-0.7)^2 + (0-0.3)^2$$
$$= 0.98$$

$\therefore J = 2$ as $D(1) > D(2)$

$$\begin{bmatrix} 0.08 & 0.92 \\ 0.6 & 0.76 \\ 0.5 & 0.28 \\ 0.9 & 0.12 \end{bmatrix}$$

$\therefore\ w_{12}(new) = w_{12}(old) + \alpha(x_1 - w_{12}(old))$

$= 0.8 + 0.6(1 - 0.8)$

$= 0.92$

Do this for all neighbours of w_{12}

- For ~~x_2~~ ~~ie~~ 0 0 0 1

using Euclidean Distance algorithm,

$D(1) = (0-0.08)^2 + (0-0.6)^2 + (0-0.5)^2 + (1-0.9)^2$

$= 0.66$

$D(2) = (0-0.92)^2 + (0-0.76)^2 + (0-0.28)^2 + (1-0.12)^2$

$= 2.2768$

$\therefore J = 1$ as $D(1) < D(2)$

$\therefore\ w_{21}(new) = w_{21}(\cancel{new}\ old) + \alpha(x_2 - w_{21}(old))$

$= 0.6 + 0.6(0 - 0.6)$

$= 0.24$

$$\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \\ w_{41} & w_{42} \end{bmatrix} = \begin{bmatrix} 0.08 & 0.92 \\ 0.24 & 0.76 \\ 0.20 & 0.28 \\ 0.96 & 0.12 \end{bmatrix}$$

$w_{31}(new) = w_{31}(old) + \alpha(x_3 - w_{31}(old))$

$= 0.5 + 0.6(0 - 0.5)$

$= 0.20$

$w_{41}(new) = w_{41}(old) + \alpha(x_4 - w_{41}(old))$

$= 0.9 + 0.6(1 - 0.9)$

$= 0.96$

$w_{11}(new) = w_{11}(old) + \alpha(x_1 - w_{11}(old))$

$= 0.2 + 0.6(0 - 0.2)$

$= 0.08$

- For vector 1000 $\cancel{x_3}$

using Euclidean distance method.

$D(1) = (1-0.08)^2 + (0-0.24)^2 + (0-0.20)^2 + (0-0.9$
$= 1.8656$

$D(2) = (1-0.92)^2 + (0-0.76)^2 + (0-0.28)^2 + (0-0.12)$
$= 0.6768$

$\because D(2) < D(1), \quad J = 2$

$\therefore w_{12}(new) = w_{12}(old) + \alpha(x_1 - w_{12}(old))$
$= 0.92 + 0.6(1-0.92)$
$= 0.968$

$w_{22}(new) = w_{22}(old) + \alpha(x_2 - w_{22}(old))$
$= 0.76 + 0.6(0-0.76)$
$= 0.304$

$w_{32}(new) = w_{23}(old) + \alpha(x_3 - w_{23}(old))$
$= 0.28 + 0.6(0-0.28)$
$= 0.112$

$w_{42}(new) = w_{24}(old) + \alpha(x_4 - w_{24}(old))$
$= 0.12 + 0.6(0-0.12)$
$= 0.048$

$$\begin{bmatrix} w_{11} & w_{21} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \\ w_{41} & w_{42} \end{bmatrix} = \begin{bmatrix} 0.08 & 0.968 \\ 0.24 & 0.304 \\ 0.20 & 0.112 \\ 0.96 & 0.048 \end{bmatrix}$$

- For vector 0011 i.e x_4

$D(1) = (-0.08)^2 + (-0.24)^2 + (1-0.20)^2 + (1-0.96)^2$
$= 0.7056$

$D(2) = (-0.968)^2 + (-0.304)^2 + (1-0.112)^2 + (1-0.048)^2$
$= 2.724$

$\because D(1) < D(2) \quad \therefore J = 1$

$w_{11}(new) = w_{11}(old) + \alpha(x_1 - w_{11}(old))$
$= 0.08 + 0.6(0 - 0.08)$
$= 0.032$

$w_{21}(new) = w_{21}(old) + \alpha(x_2 - w_{21}(old))$
$= 0.24 + 0.6(0 - 0.24)$
$= 0.096$

$w_{31}(new) = w_{31}(old) + \alpha(x_3 - w_{31}(old))$
$= 0.20 + 0.6(1 - 0.20)$
$= 0.68$

$w_{41}(new) = w_{41}(old) + \alpha(x_4 - w_{41}(old))$
$= 0.96 + 0.6(1 - 0.96)$
$= 0.984$

$$\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \\ w_{41} & w_{42} \end{bmatrix} = \begin{bmatrix} 0.032 & 0.968 \\ 0.096 & 0.304 \\ 0.68 & 0.112 \\ 0.984 & 0.048 \end{bmatrix}$$

Step 2:

Now reduce learning rate

$$\alpha(1) = \frac{\alpha(0)}{2} = \frac{0.6}{2} = 0.3.$$

It can be shown that after 100 presentations of all the i/p vector, the final weight matrix is

$$\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \\ w_{41} & w_{42} \end{bmatrix} = \begin{bmatrix} 6.7\times10^{-17} & 1 \\ 2\times10^{-16} & 0.49 \\ 0.51 & 2.3\times10^{-16} \\ 1 & 1\times10^{-16} \end{bmatrix}$$

This matrix seems to converge to .

$$\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \\ w_{41} & w_{42} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0.5 \\ 0.5 & 0 \\ 1 & 0 \end{bmatrix}$$

cluster1 cluster2

Test n/w.

Suppose the i/p patter is 1100. then.

x_1	x_2	x_3	x_4	Cluster
1	1	0	0	2
0	0	0	1	1
1	0	0	0	2
0	0	1	1	1

$$D(1) = (0-1)^2 + (0-1)^2 + (0.5-0)^2 + (1-0)^2$$
$$= 3.25$$

$$D(2) = (1-1)^2 + (0.5-1)^2 + (0-0)^2 + (0-0)^2$$
$$= 0.25.$$

$\therefore$ neuron 2 is winner.