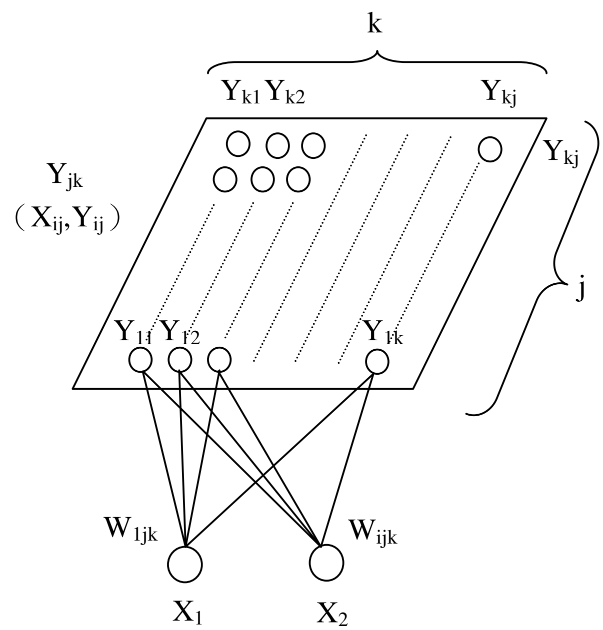
SOM（Self Organization Map）

1. It’s proposed by Kohonen in 1980.
2. SOM is an unsupervised two layered network that can organize a topological map from a random starting point.
   SOM is also called Kohnen’s self organizating feature map.
   The resulting map shows the natural relationships among the patterns that are given to the network.
3. The application is good for clustering analysis.
4. Structure：
   －One input layer
   －One competitive layer which is usually a 2-Dim grid



Input layer：f(x)：x
Output layer：competitive layer with topological map relationship
Weights：randomly assigned

Neighbor hood

①Center：the winning node C is the center.

②Distance： $r_j = \sqrt{(N_x - C_x)^2 + (N_y - C_y)^2}$

N is the node is output N to C

r_j is the distance from N to C

③R Factor（鄰近係數）：RF_j：f（r_j,R）=$e^{(-rj/R)}$

R：the radius of the neighborhood

r_j：distance from N to C

$e^{(-rj/R)}$ $\longrightarrow$ 1 when r_j=∮

$\longrightarrow$ ∮ when $r_j=\infty$

$\longrightarrow$ 0.368 when r_j=R

The longer the distance, the smaller the neighborhood area.

④R Factor Adjustment：Rn=R-rate · R^{n-1}

R-rate<1.0

Learning：

1. Setup network
2. Randomly assign weights to W
3. Set the coordinate value of the output layer N（x,y）
4. Input a training vector X
5. Compute the winning node
6. Update weight $\triangle$W with R factor
7. $\eta^{n}=\eta$ -rate・η^{n-1}

   R^{n}=R-rate・η^{n-1}
8. Repeat from 4 to 7 until converge

Reuse the network：

1. Setup the network
2. Read the weight matrix
3. Set the coordriate value of the output layer $N_{(x,y)}$
4. Read input vector
5. Compute the winning node
6. Output the clustering result Y.

Computation process

1. Setup network

   X_1~Xn （Input Vector）

   N_{jk}（output）
2. Compute winning node

   $$net_{jk} = \sqrt{\sum_i \left(X_i - W_{ijk}\right)^2}$$

   $$net_{j^*k^*} = \min_{j\bullet k}\left\lfloor net_{jk} \right\rfloor$$
3. $$Yjh= \begin{cases} 1 & \text{if } j=j^*\&k=k^* \\ \phi & \text{others} \end{cases}$$

4. Update Weights

   $\triangle W_{ijk}=\eta$（x_i-W_{ijk}）・RF_{jk}

   when $RF_{JK} = e^{\left(-r_{jk}/R\right)}$

   $r_{jk}=(N_{jk} - N_{j^*k^*})^2 = \left(j - j^*\right)^2 + (k - k^*)^2$

   $W_{ijk}=\triangle W_{ijk}+W_{ijk}$
5. Repeat 1-4 for all input
6. $\eta^{n}=\eta$ -rate・ η^{n-1}

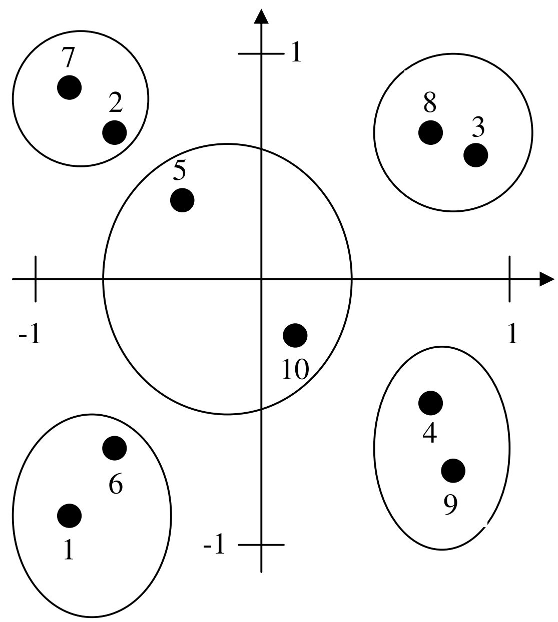
$R^n = R\text{-rate} \cdot R^{n-1}$

7. Repeat until converge

EX：Let there be one 2-Dim clustering problem.

The vector space include 5 different clusters and each has 2 sample sets.

	X1	X2
1	-0.9	-0.8
2	-0.8	0.6
3	0.9	0.6
4	0.7	-0.4
5	-0.2	0.2
6	-0.7	-0.6
7	-0.9	0.8
8	0.7	0.6
9	0.8	-0.8
10	0.1	-0.2



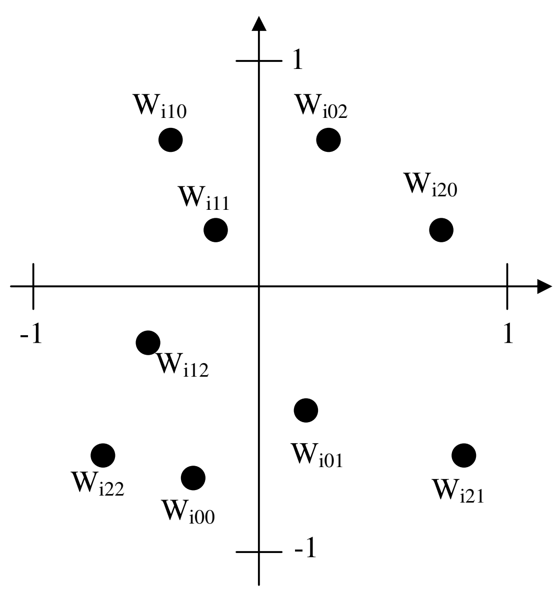
Sol：①setup an 2X9 network

②randomly assign weights

③Let R=2.0 η =1.0 $RF_{jk}=e^{\frac{-rj}{R}}$

④代入第一個 pattern[-0.9, -0.8]

	X1	X2
W_{i00}	-0.2	-0.8
W_{i01}	0.2	-0.4
W_{i02}	0.3	0.6
W_{i10}	-0.4	0.6
W_{i11}	-0.3	0.2
W_{i12}	-0.6	-0.2
W_{i20}	0.7	0.2
W_{i21}	0.8	-0.6
W_{i22}	-0.8	-0.6



$net_{00}=[(-0.9+0.2)^2+(-0.8+0.8)^2]=0.49$

$net_{01}=[(-0.9-0.2)^2+(-0.8+0.4)^2]=1.37$

$net_{02}=[(-0.9+0.3)^2+(-0.8-0.6)^2]=2.32$
$net_{10}=[(-0.9+0.4)^2+(-0.8-0.6)^2]=1.71$
$net_{11}=[(-0.9+0.3)^2+(-0.8-0.2)^2]=1.36$
$net_{12}=[(-0.9+0.6)^2+(-0.8+0.2)^2]=0.45$
$net_{20}=[(-0.9+0.7)^2+(-0.8-0.2)^2]=1.04$
$net_{21}=[(-0.9-0.8)^2+(-0.8+0.6)^2]=2.93$
$net_{22}=[(-0.9+0.8)^2+(-0.8+0.6)^2]=0.05$【MIN】

$\min_{j,k}[net_{jk}] \longrightarrow net_{22}$

j*2 k*2

---

⑤修正 weight

j*=2 k*=2

$r_{00}=\sqrt{(0-2)^2+(0-2)^2}=2.282$ $RF_{00}=0.243$

$r_{01}=\sqrt{(0-2)^2+(1-2)^2}$ $RF_{00}=$

$r_{02}=\sqrt{(0-2)^2+(2-2)^2}$

$r22=\sqrt{(2-2)^2+(2-2)^2}=\phi$ $RF_{00}=1$

$\triangle W_{00}=\eta(X_1-W_{00})RF_{00}$
1.0（-0.9+0.2）×（0.243）=－0.17