Dr. Mohammad Nurul Huda
Machine Learning
Lecture - IV

Self Organizing Maps (SOMs):

An artificial neural network (ANN) model important aspect is whether it needs guidance in learning or not.

Two learning categories
- Supervised learning
- Unsupervised learning

Supervised learning:

Training phase:
- Each input needs a desired output

Labelled data
Teacher data

Example: Multi layer perceptron

Unsupervised learning:

- No target results for the input data vectors

Example: Self organizing map (SOM)

Why SOM is needed?

- SOM can be used for clustering the input data
- SOM can be used to detect features inherent to the problem.

Properties of SOM:

- Developed by Professor Kohonen.
- Competitive learning networks.
- Unsupervised learning
- Used for clustering input data
- Used to detect features inherent to the problem.
- Can recognize or characterize inputs it has never encountered before.

SOM

- A categorization method
- A neural network technique
- Unsupervised

Input and Output

- Training data: vectors, X
  - Vectors of length n

$$\left.\begin{matrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ x_{31} & x_{32} & \cdots & x_{3n} \\ \vdots & & & \\ x_{p1} & x_{p2} & \cdots & x_{pn} \end{matrix}\right\} \text{p distinct training vectors}$$

  - Vector components are real numbers

- Outputs
  - A vector, Y, of length m:

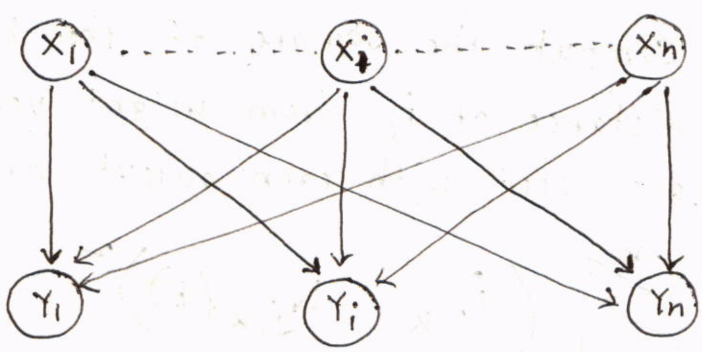
    $Y_1 \; Y_2 \; Y_3 \ldots Y_m$

  - Each of the p vectors in the training data is classified as falling in one of m clusters or categories
  - Which category does the training vector fall into?

- Generalization
  - For a new vector: $(x_1, x_2, x_3, \ldots x_n)$
  - Which of the m categories (clusters) does it fall into?

Network Architecture

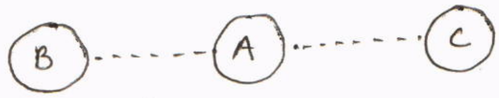
- Two layers of units
  - Input: n units (length of training vectors)
  - Output: m units (number of categories)
- Input units full connected with weights to output units.
- Intra-layer (lateral) connections
  - Within output layer
  - No weight between these connections, but used in algorithm for updating weights.



[There is one weight vector of length n associated with each output unit]

Output Layer Topology

- View output in spatial manner
  - a 1D or 2D arrangement
- 1D arrangement
  - Topology defines which output layer units are neighbors with which others.
  - Have a function, D(t), which gives output unit neighborhood as a function of time (iterations) of the training algorithm.
  - 3 output units



$D(t)=1$ means update weight B and A if input maps onto B.

SOM Algorithm

step1: Initialize random small values to weights.

step2:

$t=1$

While computational bounds are not exceeded do

i) Select input vector i_l

ii) Comput the square of the Euclidean distance of i_l from weight vectors (W_j) associated with each output node

$$\sum_{k=1}^{n}\left(i_{l,k}-W_{j,k}(t)\right)^2$$

iii) Select output node j^* that has weight vector with minimum value from step 2

iv) Update weights to all nodes within a topological distance given by $D(t)$ from j^*, using the weight update rule

$$W_j(t+1)=W_j(t)+\eta(t)\left(i_l-W_j(t)\right)$$

v) Increment t

end while

** Learning rate generally decreases with time:

$$0<\eta(t)<\eta(t-1)\leqslant 1$$

Example:

Training samples

i1: (1, 1, 0, 0)

i2: (0, 0, 0, 1)

i3: (1, 0, 0, 0)

i4: (0, 0, 1, 1)

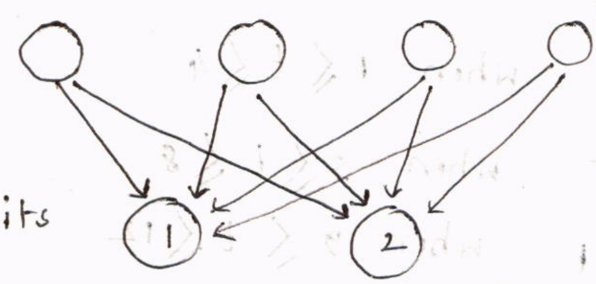
Number of training vectors N = 4

input dimension, n = 4

classify two classes

Network Architecture:

Input units



Output units

Categorization by Euclidean distance:

	i1	i2	i3	i4
i1	0			
i2	3	0		
i3	1	2	0	
i4	4	1	3	0

class-I: (i_1, i_3)

class-II: (i_2, i_4)

Categorization by SOM :

- Training samples

  i1: (1, 1, 0, 0)

  i2: (0, 0, 0, 1)

  i3: (1, 0, 0, 0)

  i4: (0, 0, 1, 1)

- Let neighborhood, $D(t) = 0$
  - Only update weights associated with winning output unit (cluster) at each iteration.

- Learning rate

  $\eta(t) = 0.6$ when $1 \leqslant t \leqslant 4$

  $\eta(t) = 0.5$ when $5 \leqslant t \leqslant 8$

  $\eta(t) = 0.1$ when $9 \leqslant t \leqslant 12$

- Initial weight matrix (random values between 0 and 1)

$$\begin{array}{r} \text{Unit1:} \\ \text{Unit2:} \end{array} \begin{bmatrix} 0.2 & 0.6 & 0.5 & 0.9 \\ 0.8 & 0.4 & 0.7 & 0.3 \end{bmatrix}$$

input dim, n=4

- $d^2 = (\text{Euclidean distance})^2 = \sum_{k=1}^{n} \left(i_{l,k} - W_{j,k}(t)\right)^2$

- Weight update :

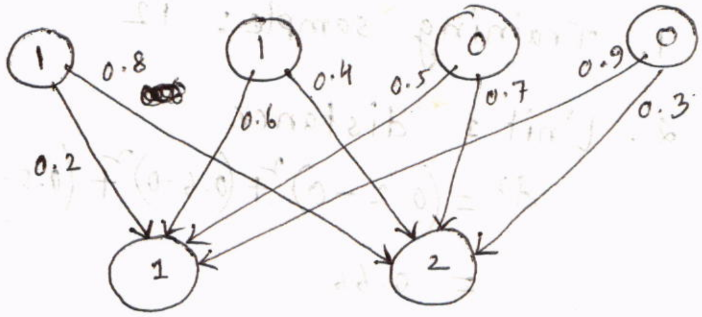
$$W_j(t+1) = W_j(t) + \eta(t)\left(i_l - W_j(t)\right)$$

First Weight Update :

Weight matrix :

$$\begin{array}{c} \text{Unit 1} \\ \text{Unit 2} \end{array} \begin{bmatrix} 0.2 & 0.6 & 0.5 & 0.9 \\ 0.8 & 0.4 & 0.7 & 0.3 \end{bmatrix}$$

* i1: (1, 1, 0, 0)
i2: (0, 0, 0, 1)
i3: (1, 0, 0, 0)
i4: (0, 0, 1, 1)



1. Training sample : i1
2. Unit 1 distance

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

Unit 2 distance

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

3. Since $1.86 > 0.98$, Unit2 wins

4. Update unit 2 weight vector of weight matrix

$$[0.8 \quad 0.4 \quad 0.7 \quad 0.3] + \eta * [[1 \quad 1 \quad 0 \quad 0] - [0.8 \quad 0.4 \quad 0.7 \quad 0.3]]$$
$$= [0.8 \quad 0.4 \quad 0.7 \quad 0.3] + 0.6 * [0.2 \quad 0.6 \quad -0.7 \quad -0.3]$$
$$= [0.8 \quad 0.4 \quad 0.7 \quad 0.3] + [0.12 \quad 0.36 \quad -.42 \quad -0.18]$$
$$= [0.92 \quad 0.76 \quad 0.28 \quad 0.12]$$

New weight matrix :

$$\begin{array}{c} \text{Unit 1} \\ \text{Unit 2} \end{array} \begin{bmatrix} 0.2 & 0.6 & 0.5 & 0.9 \\ 0.92 & 0.76 & 0.28 & 0.12 \end{bmatrix}$$

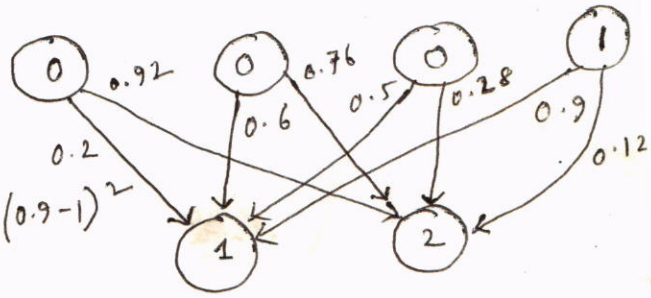
Second Weight Update :

Weight Matrix :

$$\begin{array}{l} \text{Unit 1} \\ \text{Unit 2} \end{array} \begin{bmatrix} 0.2 & 0.6 & 0.5 & 0.9 \\ 0.92 & 0.76 & 0.28 & 0.12 \end{bmatrix}$$

i1: (1, 1, 0, 0)
*i2: (0, 0, 0, 1)
i3: (1, 0, 0, 0)
i4: (0, 0, 1, 1)

1. Training sample: i2



2. Unit 1 distance

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

$$= 0.66$$

Unit 2 distance

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

$$= 2.28$$

3. Since $2.28 > 0.66$, Unit 1 wins.

4. Update Unit 1 weight vector of weight matrix

$$[0.2 \;\; 0.6 \;\; 0.5 \;\; 0.9] + \eta \, [[0 \;\; 0 \;\; 0 \;\; 1] - [0.2 \;\; 0.6 \;\; 0.5 \;\; 0.9]]$$

$$= [0.2 \;\; 0.6 \;\; 0.5 \;\; 0.9] + 0.6 \, [-0.2 \;\; -0.6 \;\; -0.5 \;\; 0.1]$$

$$= [0.08 \;\; 0.24 \;\; 0.20 \;\; 0.96]$$

New weight Matrix :

$$\begin{array}{l} \text{Unit 1} \\ \text{Unit 2} \end{array} \begin{bmatrix} 0.08 & 0.24 & 0.20 & 0.96 \\ 0.92 & 0.76 & 0.28 & 0.12 \end{bmatrix}$$

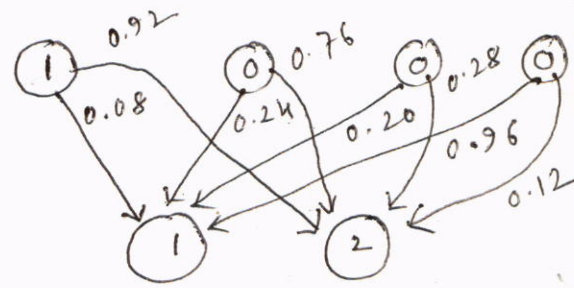
Third Weight Update:

Weight Matrix:

$$\begin{array}{l} \text{Unit 1} \\ \text{Unit 2} \end{array} \begin{bmatrix} 0.08 & 0.24 & 0.20 & 0.96 \\ 0.92 & 0.76 & 0.28 & 0.12 \end{bmatrix}$$

i1: (1, 1, 0, 0)
i2: (0, 0, 0, 1)
* i3: (1, 0, 0, 0)
i4: (0, 0, 1, 1)

1. Training sample: i3



2. Unit 1 distance

$$d^2 = (0.08-1)^2 + (0.24-0)^2 + (0.20-0)^2 + (0.96-0)^2 = 1.87$$

Unit 2 distance

$$d^2 = (0.92-1)^2 + (0.76-0)^2 + (0.28-0)^2 + (0.12-0)^2 = 0.68$$

3. Since $1.87 > 0.68$, Unit 2 wins

4. Update Unit 2 weight vector of Weight Matrix

$$[0.92 \;\; 0.76 \;\; 0.28 \;\; 0.12] + \eta\,[[1,0,0,0] - [0.92 \;\; 0.76 \;\; 0.28 \;\; .12]]$$

$$= [0.92 \;\; 0.76 \;\; 0.28 \;\; 0.12] + 0.6\,[[1,0 \;\; 0 \;\; 0] - [0.92 \;\; 0.76 \;\; 0.28 \;\; .12]]$$

$$= [\overset{0.97}{\cancel{0.08}} \;\; \overset{0.30}{\cancel{0.10}} \;\; \overset{0.11}{\cancel{0.68}} \;\; \overset{0.05}{\cancel{0.98}}]$$

New weight Matrix:

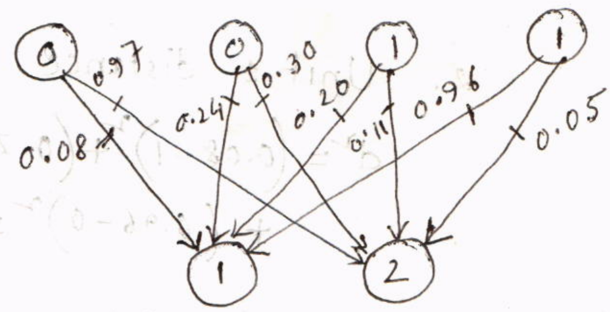
$$\begin{array}{l} \text{Unit 1} \\ \text{Unit 2} \end{array} \begin{bmatrix} 0.08 & 0.24 & 0.20 & 0.96 \\ \overset{0.97}{\cancel{0.08}} & \overset{0.30}{\cancel{0.10}} & \overset{0.11}{\cancel{0.68}} & \overset{0.05}{\cancel{0.98}} \end{bmatrix}$$

Fourth Weight Update

Weight matrix:

$$\begin{array}{l} \text{Unit 1} \\ \text{Unit 2} \end{array} \begin{bmatrix} 0.08 & 0.24 & 0.20 & 0.96 \\ 0.97 & 0.30 & 0.11 & 0.05 \end{bmatrix}$$

i1: (1, 1, 0, 0)
i2: (0, 0, 0, 1)
i3: (1, 0, 0, 0)
*i4: (0, 0, 1, 1)



1. Training sample: i4

2. Unit 1 distance

$$d^2 = (0.08-0)^2 + (0.24-0)^2 + (0.20-1)^2 + (0.96-1)^2$$

$$= 0.71$$

Unit 2 distance

$$d^2 = (0.97-0)^2 + (0.30-0)^2 + (0.11-1)^2 + (0.05-1)^2$$

$$= 2.74$$

3. Since $2.74 > 0.71$, Unit 1 wins

4. Update Unit 1 weight vector of weight matrix

$$[0.08 \quad 0.24 \quad 0.20 \quad 0.96] + \eta\big[[0 \; 0 \; 1 \; 1] - [0.08 \; .24 \; .20 \; .96]\big]$$

$$= [0.08 \quad 0.24 \quad 0.20 \quad 0.96] + 0.6\big[[0 \; 0 \; 1 \; 1] - [0.08 \; .24 \; .20 \; .96]\big]$$

$$= [0.03 \quad 0.10 \quad 0.68 \quad 0.98]$$

New weight Matrix:

$$\begin{array}{l} \text{Unit 1} \\ \text{Unit 2} \end{array} \begin{bmatrix} 0.03 & 0.10 & 0.68 & 0.98 \\ 0.97 & 0.30 & 0.11 & 0.05 \end{bmatrix}$$

~~Fourth Weight Update:~~

Summary:

Training data samples

time, t	i_1	i_2	i_3	i_4	$D(t)$	$\eta(t)$
1	Unit 2				0	0.6
2		Unit 1			0	0.6
3			Unit 2		0	0.6
4				Unit 1	0	0.6

Winning output unit

Weight Matrix:

$$\begin{array}{l} \text{Unit 1} \\ \text{Unit 2} \end{array} \begin{bmatrix} 0.03 & 0.10 & 0.68 & 0.98 \\ 0.97 & 0.30 & 0.11 & 0.05 \end{bmatrix}$$

$$\Downarrow$$

After many iterations (epoches) through the data set

Weight matrix

$$\begin{array}{l} \text{Unit 1} \\ \text{Unit 2} \end{array} \begin{bmatrix} 0 & 0 & 0.5 & 1.0 \\ 1.0 & 0.5 & 0 & 0 \end{bmatrix}$$

Find in which clusters the training data samples fall into?

Training data samples

i1: (1, 1, 0, 0)

i2: (0, 0, 0, 1)

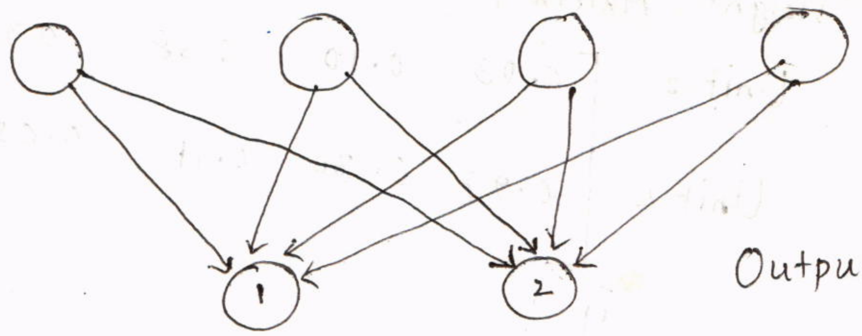
i3: (1, 0, 0, 0)

i4: (0, 0, 1, 1)

Weight Matrix:

$$\begin{array}{l} \text{Unit 1} \\ \text{Unit 2} \end{array} \begin{bmatrix} 0 & 0 & 0.5 & 1.0 \\ 1.0 & 0.5 & 0 & 0 \end{bmatrix}$$

Network:



Input Units

Output Units

For sample i1

Unit 1 distance: $d^2 = (1-0)^2 + (1-0)^2 + (0-0.5)^2 + (0-1.0)^2 = 3.25$

[winner] Unit 2 distance: $d^2 = (1-1)^2 + (1-0.5)^2 + (0-0)^2 + (0-0)^2 = 0.25$

For sample i2

[Winner] Unit 1 distance: $d^2 = (0-0)^2 + (0-0)^2 + (0-0.5)^2 + (1-1.0)^2 = 0.25$

Unit 2 distance: $d^2 = (0-1.0)^2 + (0-.5)^2 + (0-0)^2 + (1-0)^2 = 2.25$

For sample i3

Unit 1 distance: $d^2 = (1-0)^2 + (0-0)^2 + (0-0.5)^2 + (0-1.0)^2 = 2.25$

(winner) Unit 2 distance: $d^2 = (1-1.0)^2 + (0-0.5)^2 + (0-0)^2 + (0-0)^2 = 0.25$

For sample i4

(Winner) Unit 1 distance: $d^2 = (0-0)^2 + (0-0)^2 + (1-0.5)^2 + (1-1.0)^2 = 0.25$

Unit 2 distance: $d^2 = (0-1.0)^2 + (0-0.5)^2 + (1-0)^2 + (1-0)^2 = 3.25$

Output: sample i1, i3 in Unit 2 or class-II

i2, i4 Unit 1 or class-I

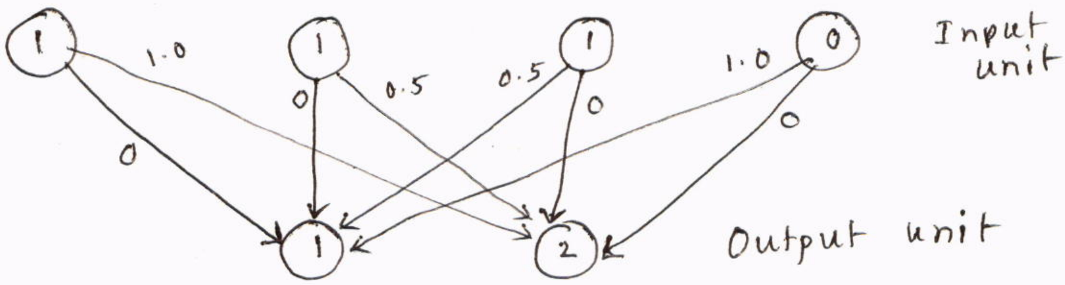
Find in which clusters the new data sample (1, 1, 1, 0) fall into

Data sample: (1, 1, 1, 0)

Weight matrix:

$$\begin{array}{c} \text{Unit 1} \\ \\ \text{Unit 2} \end{array} \begin{bmatrix} 0 & 0 & 0.5 & 1.0 \\ \\ 1.0 & 0.5 & 0 & 0 \end{bmatrix}$$

Network:



Unit 1 distance: $d^2 = (1-0)^2 + (1-0)^2 + (1-0.5)^2 + (0-1.0)^2$

$= 1 + 1 + 0.25 + 1 = 3.25$

Unit 2 distance: $d^2 = (1-1.0)^2 + (1-0.5)^2 + (1-0)^2 + (0-0)^2$

$= 0 + 0.25 + 1 + 0 = 1.25$

Therefore, new data sample (1, 1, 1, 0) is in Unit 2 or class 2