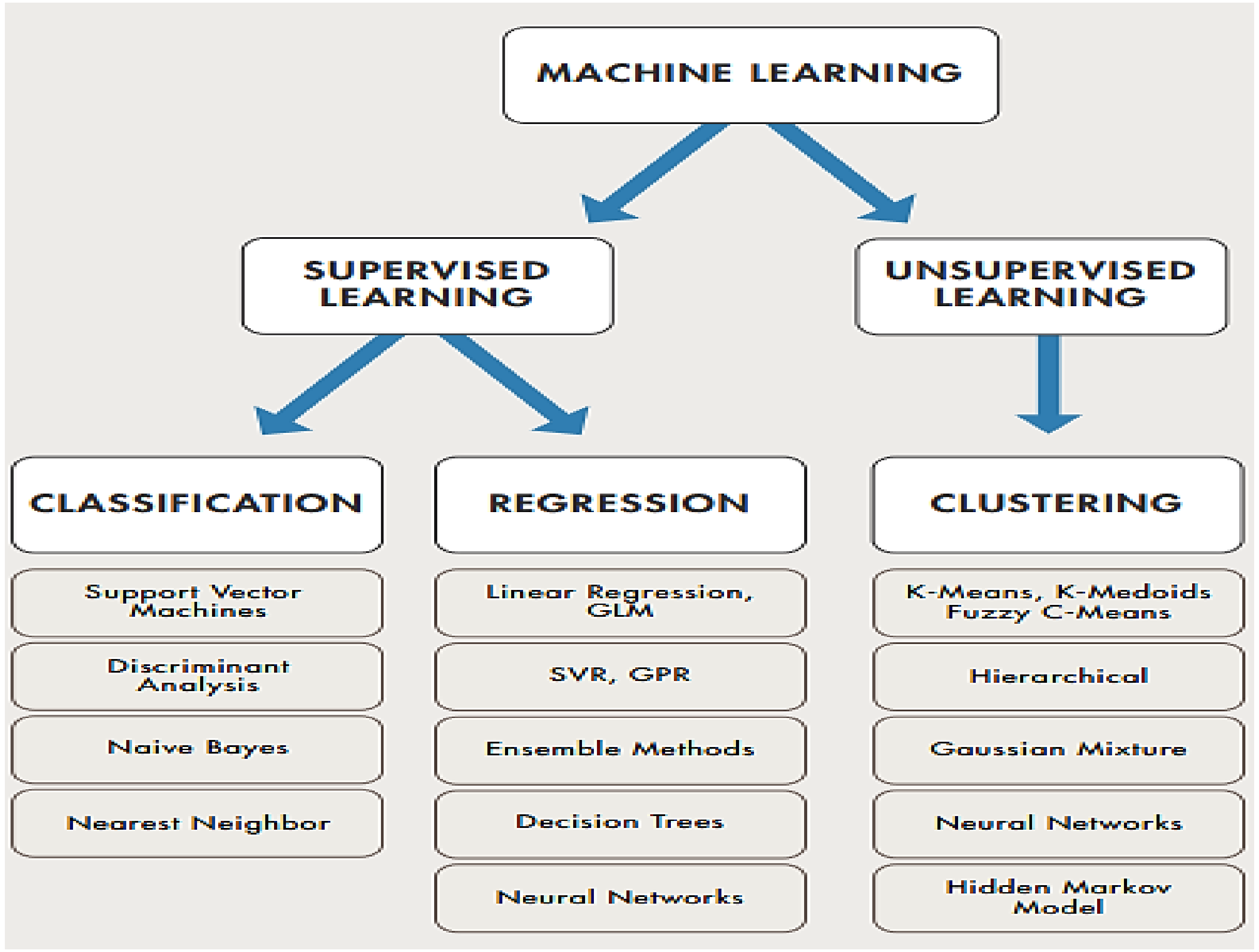
CLUSTERING

K-Means,SOM

Learning

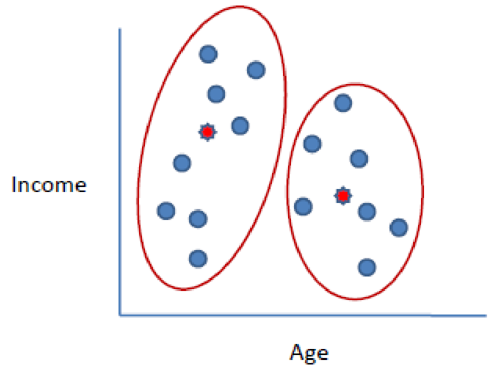

MACHINE LEARNING
SUPERVISED LEARNING
UNSUPERVISED LEARNING
CLASSIFICATION
REGRESSION
CLUSTERING
Support Vector Machines
Discriminant Analysis
Naive Bayes
Nearest Neighbor
Linear Regression, GLM
SVR, GPR
Ensemble Methods
Decision Trees
Neural Networks
K-Means, K-Medoids Fuzzy C-Means
Hierarchical
Gaussian Mixture
Neural Networks
Hidden Markov Model

K-Means Clustering

- K-means clustering is one of the simplest and popular unsupervised machine learning algorithms.
- Typically, unsupervised algorithms make inferences from datasets using only input vectors without referring to known, or labelled, outcomes

K-Means Algorithm

- Clusters the data into *k* groups where *k* is predefined.
- Select *k* points at random as cluster centers.
- Assign objects to their closest cluster center according to the *Euclidean distance* function.
- Calculate the centroid or mean of all objects in each cluster.
- Repeat steps 2, 3 and 4 until the same points are assigned to each cluster in consecutive rounds.



Example

- **Suppose we want to group the visitors to a website using just their age (one-dimensional space) as follows:**

***n* = 19**

- **15,15,16,19,19,20,20,21,22,28,35,40,41,42,43,44,60,61,65**

- Initial clusters (random centroid or average):

$$k = 2$$

$$c_1 = 16$$
$$c_2 = 22$$

$$\mathit{Distance}\ 1 = |x_i - c_1|$$

$$\mathit{Distance}\ 2 = |x_i - c_2|$$

Iteration 1:

$c_1 = 15.33$
$c_2 = 36.25$

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	16	22	1	7	1	**15.33**
15	16	22	1	7	1	
16	16	22	0	6	1	
19	16	22	3	3	2	**36.25**
19	16	22	3	3	2	
20	16	22	4	2	2	
20	16	22	4	2	2	
21	16	22	5	1	2	
22	16	22	6	0	2	
28	16	22	12	6	2	
35	16	22	19	13	2	
40	16	22	24	18	2	
41	16	22	25	19	2	
42	16	22	26	20	2	
43	16	22	27	21	2	
44	16	22	28	22	2	
60	16	22	44	38	2	
61	16	22	45	39	2	
65	16	22	49	43	2	

Iteration 2:

$c_1 = 18.56$

$c_2 = 45.90$

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	15.33	36.25	0.33	21.25	1	
15	15.33	36.25	0.33	21.25	1	
16	15.33	36.25	0.67	20.25	1	
19	15.33	36.25	3.67	17.25	1	
19	15.33	36.25	3.67	17.25	1	**18.56**
20	15.33	36.25	4.67	16.25	1	
20	15.33	36.25	4.67	16.25	1	
21	15.33	36.25	5.67	15.25	1	
22	15.33	36.25	6.67	14.25	1	
28	15.33	36.25	12.67	8.25	2	
35	15.33	36.25	19.67	1.25	2	
40	15.33	36.25	24.67	3.75	2	
41	15.33	36.25	25.67	4.75	2	
42	15.33	36.25	26.67	5.75	2	
43	15.33	36.25	27.67	6.75	2	**45.9**
44	15.33	36.25	28.67	7.75	2	
60	15.33	36.25	44.67	23.75	2	
61	15.33	36.25	45.67	24.75	2	
65	15.33	36.25	49.67	28.75	2	

Iteration 3:

$c_1 = 19.50$
$c_2 = 47.89$

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	18.56	45.9	3.56	30.9	1	**19.50**
15	18.56	45.9	3.56	30.9	1	
16	18.56	45.9	2.56	29.9	1	
19	18.56	45.9	0.44	26.9	1	
19	18.56	45.9	0.44	26.9	1	
20	18.56	45.9	1.44	25.9	1	
20	18.56	45.9	1.44	25.9	1	
21	18.56	45.9	2.44	24.9	1	
22	18.56	45.9	3.44	23.9	1	
28	18.56	45.9	9.44	17.9	1	
35	18.56	45.9	16.44	10.9	2	**47.89**
40	18.56	45.9	21.44	5.9	2	
41	18.56	45.9	22.44	4.9	2	
42	18.56	45.9	23.44	3.9	2	
43	18.56	45.9	24.44	2.9	2	
44	18.56	45.9	25.44	1.9	2	
60	18.56	45.9	41.44	14.1	2	
61	18.56	45.9	42.44	15.1	2	
65	18.56	45.9	46.44	19.1	2	

Iteration 4:

$c_1 = 19.50$

$c_2 = 47.89$

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	19.5	47.89	4.50	32.89	1	**19.50**
15	19.5	47.89	4.50	32.89	1	
16	19.5	47.89	3.50	31.89	1	
19	19.5	47.89	0.50	28.89	1	
19	19.5	47.89	0.50	28.89	1	
20	19.5	47.89	0.50	27.89	1	
20	19.5	47.89	0.50	27.89	1	
21	19.5	47.89	1.50	26.89	1	
22	19.5	47.89	2.50	25.89	1	
28	19.5	47.89	8.50	19.89	1	
35	19.5	47.89	15.50	12.89	2	**47.89**
40	19.5	47.89	20.50	7.89	2	
41	19.5	47.89	21.50	6.89	2	
42	19.5	47.89	22.50	5.89	2	
43	19.5	47.89	23.50	4.89	2	
44	19.5	47.89	24.50	3.89	2	
60	19.5	47.89	40.50	12.11	2	
61	19.5	47.89	41.50	13.11	2	
65	19.5	47.89	45.50	17.11	2	

No change between iterations 3 and 4 has been noted. By using clustering, 2 groups have been identified 15-28 and 35-65.

K-Means Clustering for 2 attributes

subject	A	B
1	1	1
2	1.5	2
3	3	4
4	5	7
5	3.5	5
6	4.5	5
7	3.5	4.5

Algorithm

- Choose the number of clusters you want to create
- As per the number of clusters, choose initial seeds to start.
- Calculate distance of every value now from the initial seeds.
- Now every row will be assigned to one cluster as per its minimum distance.
- After one iteration, mean is calculated of each cluster and again same steps are performed until convergence.

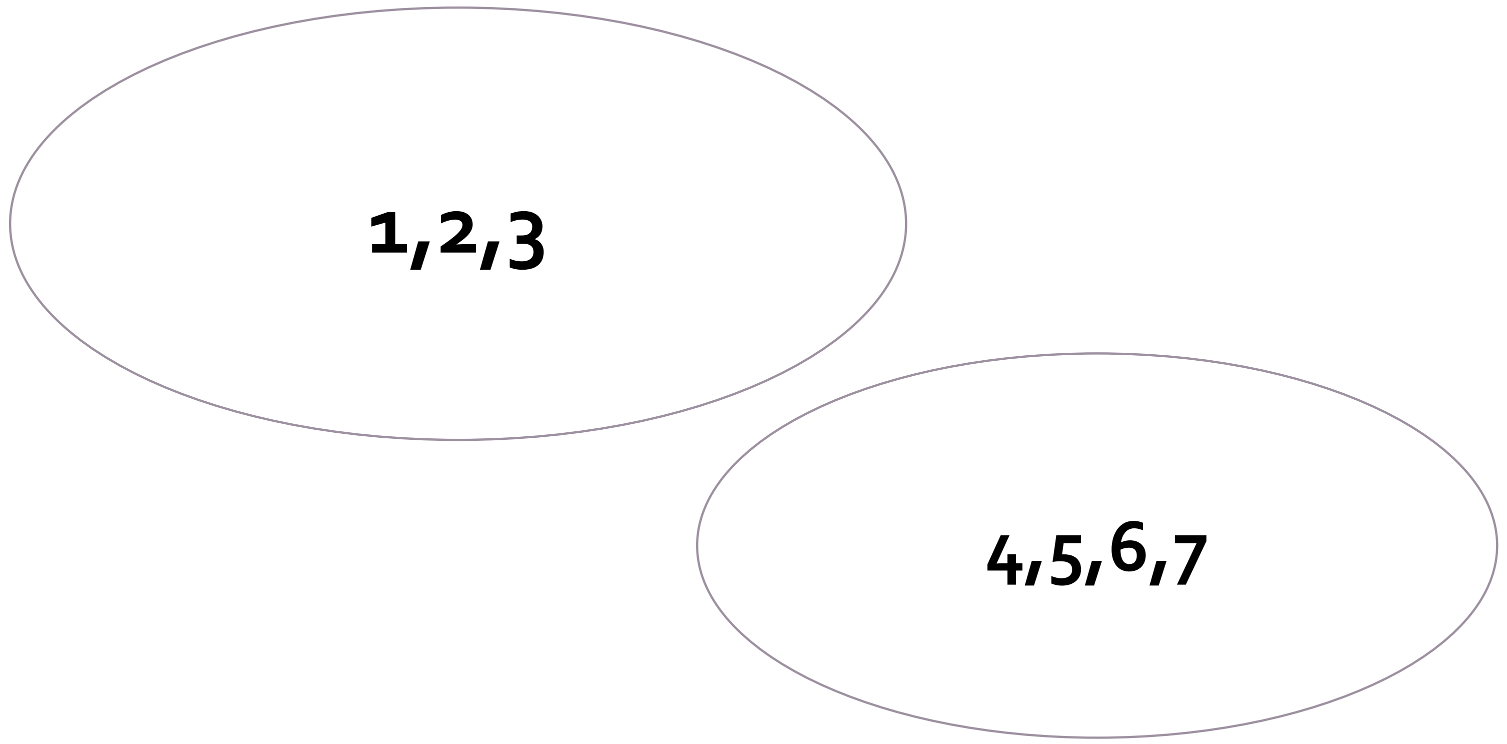
First iteration

	Seed 1	Seed 2
1	0	7.21
2	1.12	6.10
3	3.61	3.61
4	7.21	0
5	4.72	2.5
6	5.32	2.06
7	4.30	2.92

Initial seeds had been (1,1) and (5,7)
Now using these distances calculated we will put each instance into one cluster

1 complete cycle

subject	A	B	Seed 1	Seed 2	Group
1	1	1	0	7.21	C1
2	1.5	2	1.12	6.10	C1
3	3	4	3.61	3.61	C1
4	5	7	7.21	0	C2
5	3.5	5	4.72	2.5	C2
6	4.5	5	5.32	2.06	C2
7	3.5	4.5	4.30	2.92	C2



- Now instance 1,2,3 will be in cluster 1 and the rest will be in cluster 2

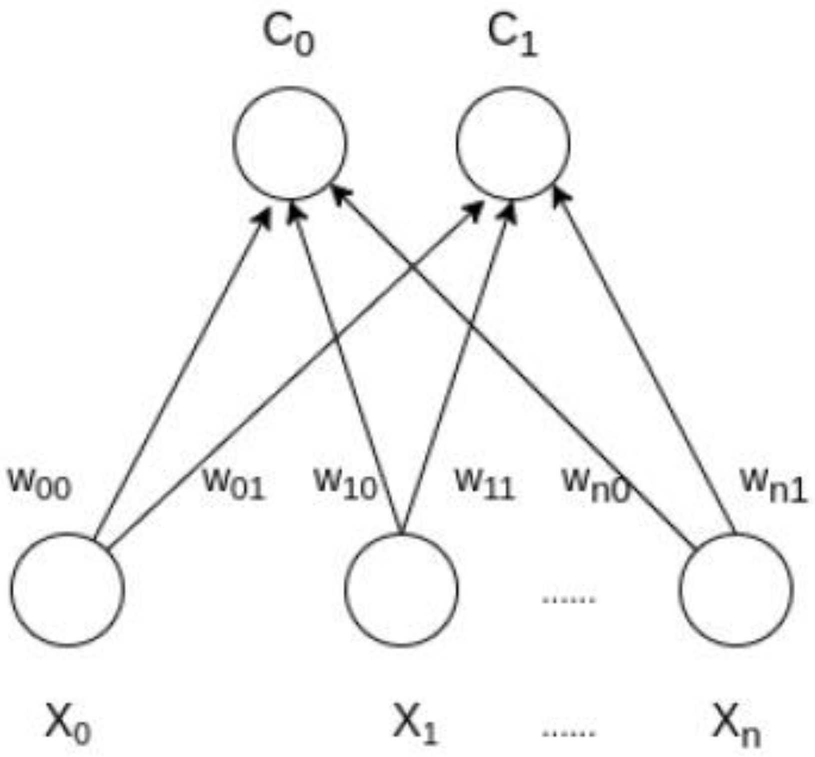
Second iteration

- Now calculate centroid of each cluster and then re calculate everything done in iteration 1.
- This will be done until no change happens.
- After iteration 3 you will observe that no changes will occur from iteration 2 to 3 thereby we stop.
- In iteration 2 instance 3 will move to cluster 2.

2: Self-Organizing Map (SOM)

- A self-organizing map is a type of artificial neural network that attempts to build a two-dimensional map of some problem space.
- It follows an unsupervised learning approach and trained its network through a competitive learning algorithm.
- SOM is used for clustering and mapping (or dimensionality reduction) techniques to map multidimensional data onto lower-dimensional which allows people to reduce complex problems for easy interpretation.
- SOM has two layers, one is the Input layer and the other one is the Output layer.

Architecture of the Self Organizing Map with two clusters



Steps

- 1. Each node’s weights are initialized.
- 2. A vector is chosen at random from the set of training data and presented to the lattice.
- 3. Every node is examined to calculate which ones weights are most like the input vector. The winning node is commonly known as the Best Matching Unit (BMU).
- 4. The radius of the neighborhood of the BMU is now calculated. This is a value that starts large, typically set to the ‘radius’ of the lattice, but diminishes each time-step. Any nodes found within this radius are deemed to be inside the BMU’s neighborhood.
- 5. Each neighboring node’s (the nodes found in step 4) weights are adjusted to make them more like the input vector. The closer a node is to the BMU; the more its weights get altered.
- 6. Repeat step 2 for N iterations.

Best matching unit

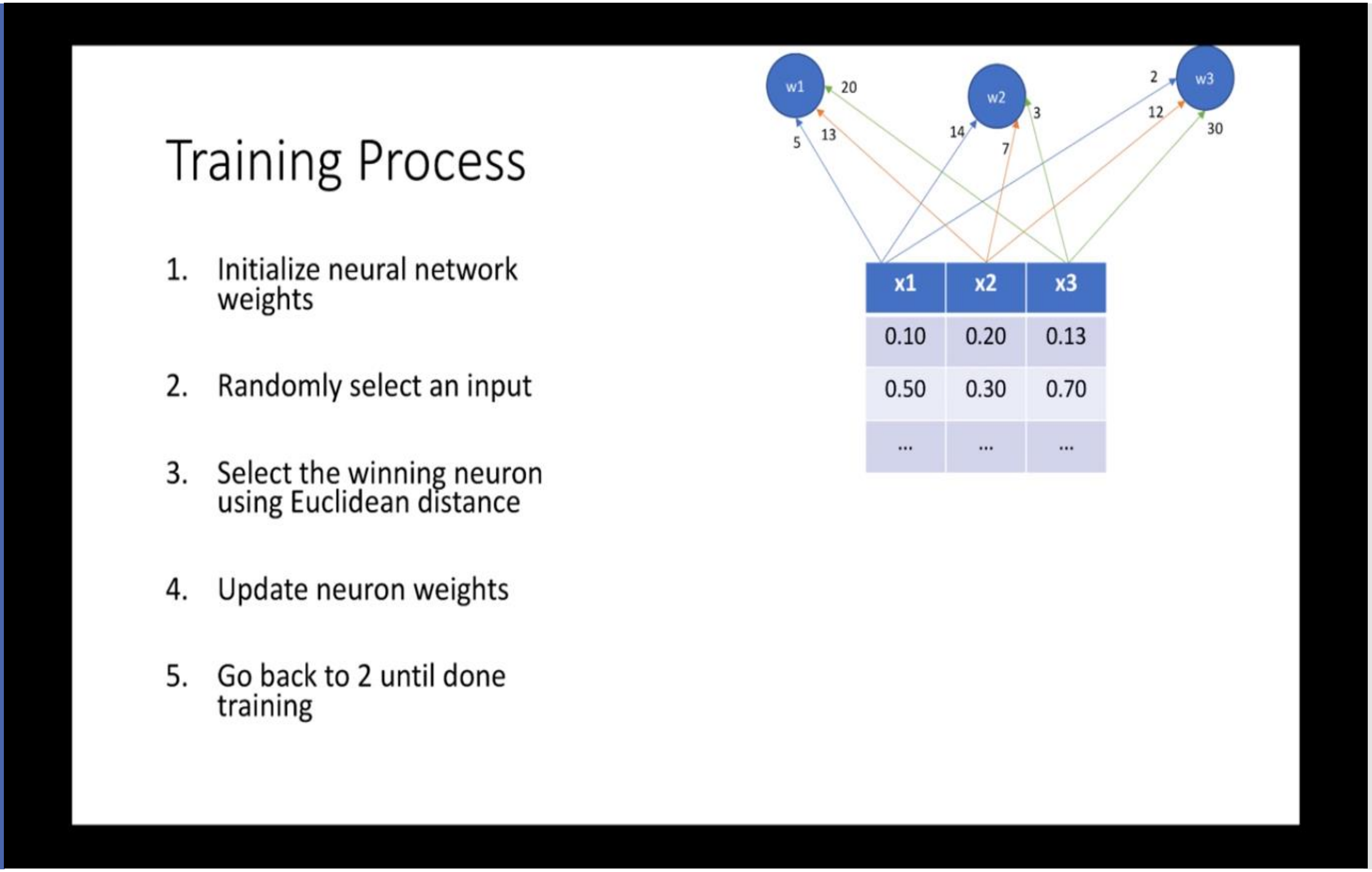
- To determine the best matching unit, one method is to iterate through all the nodes and calculate the Euclidean distance between each node's weight vector and the current input vector. The node with a weight vector closest to the input vector is tagged as the BMU.

The Euclidean distance is given as:

Where X is the current input vector and W is the node's weight vector.

$$\text{Distance} = \sqrt{\sum_{i=0}^{i=n} (X_i - W_i)^2}$$

SOM Example

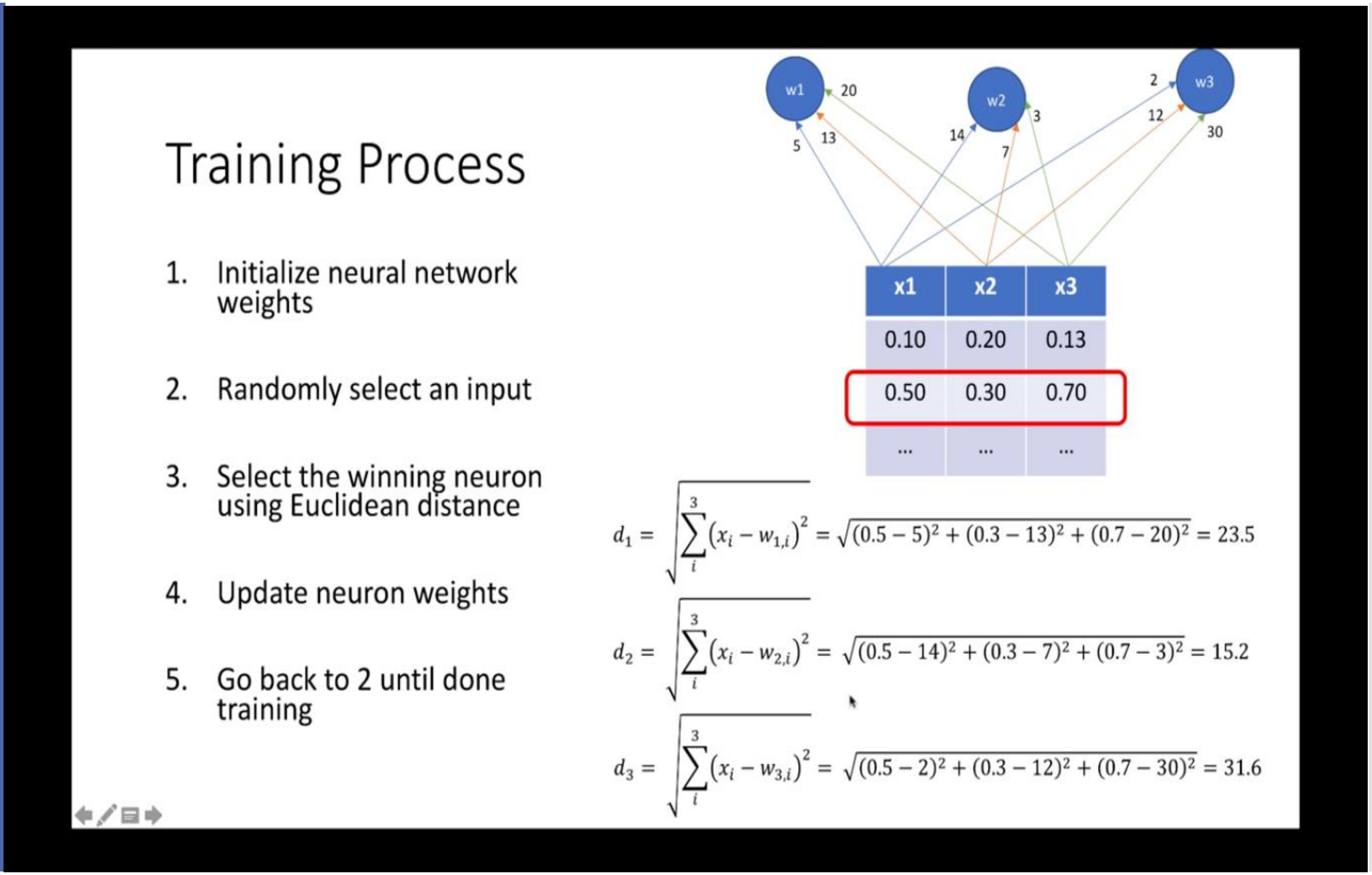


Training Process

1. Initialize neural network weights
2. Randomly select an input
3. Select the winning neuron using Euclidean distance
4. Update neuron weights
5. Go back to 2 until done training

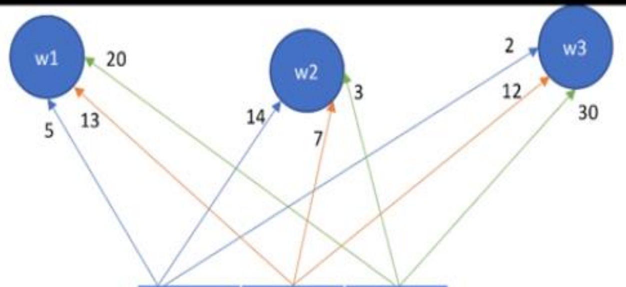
x1	x2	x3
0.10	0.20	0.13
0.50	0.30	0.70
...	...	...

Example..



Training Process

1. Initialize neural network weights
2. Randomly select an input
3. Select the winning neuron using Euclidean distance
4. Update neuron weights
5. Go back to 2 until done training



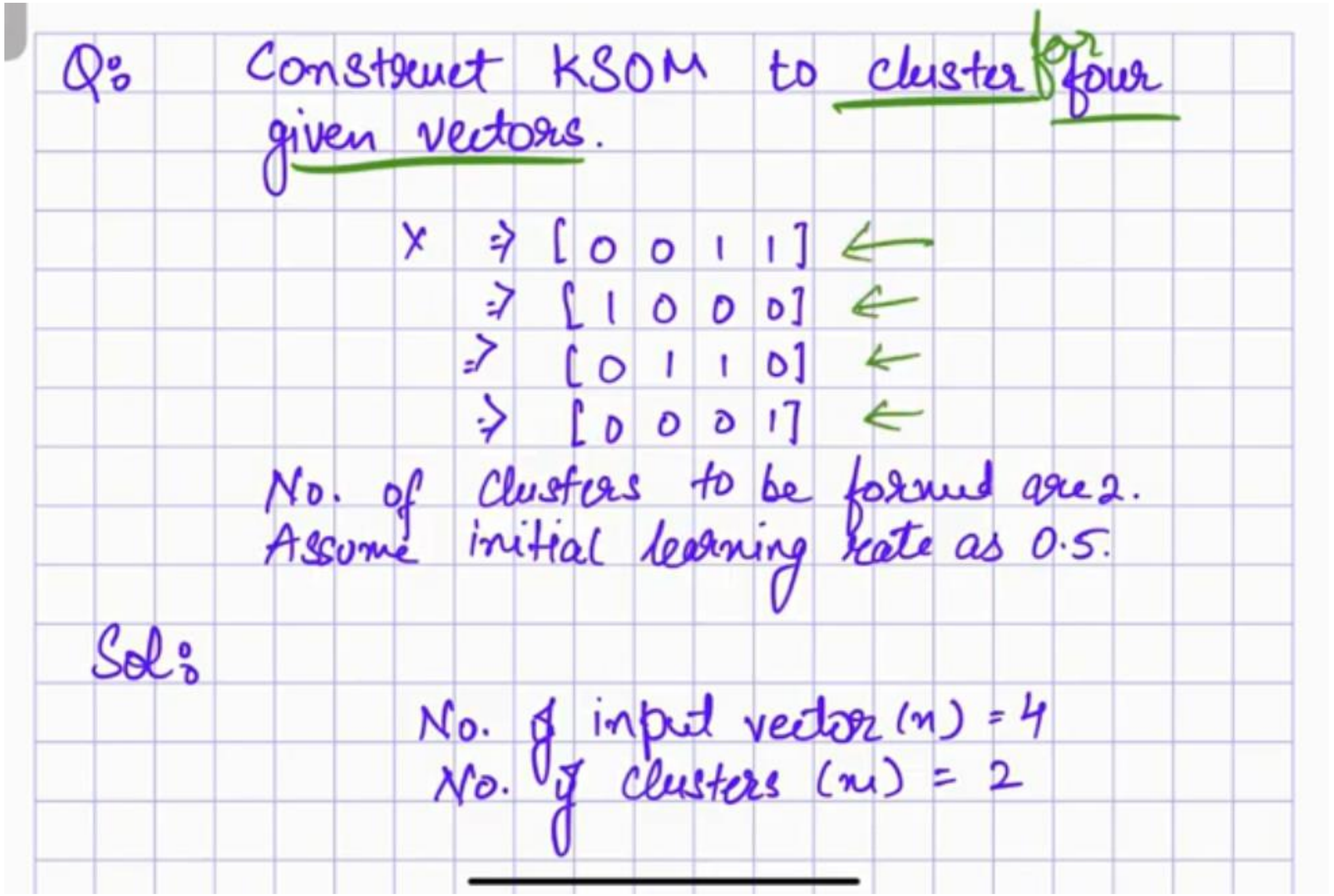
x1	x2	x3
0.10	0.20	0.13
0.50	0.30	0.70
...	...	...

$$d_1 = \sqrt{\sum_i^3 (x_i - w_{1,i})^2} = \sqrt{(0.5-5)^2 + (0.3-13)^2 + (0.7-20)^2} = 23.5$$

$$d_2 = \sqrt{\sum_i^3 (x_i - w_{2,i})^2} = \sqrt{(0.5-14)^2 + (0.3-7)^2 + (0.7-3)^2} = 15.2$$

$$d_3 = \sqrt{\sum_i^3 (x_i - w_{3,i})^2} = \sqrt{(0.5-2)^2 + (0.3-12)^2 + (0.7-30)^2} = 31.6$$

Scenario



Q: Construct KSOM to cluster for four given vectors.

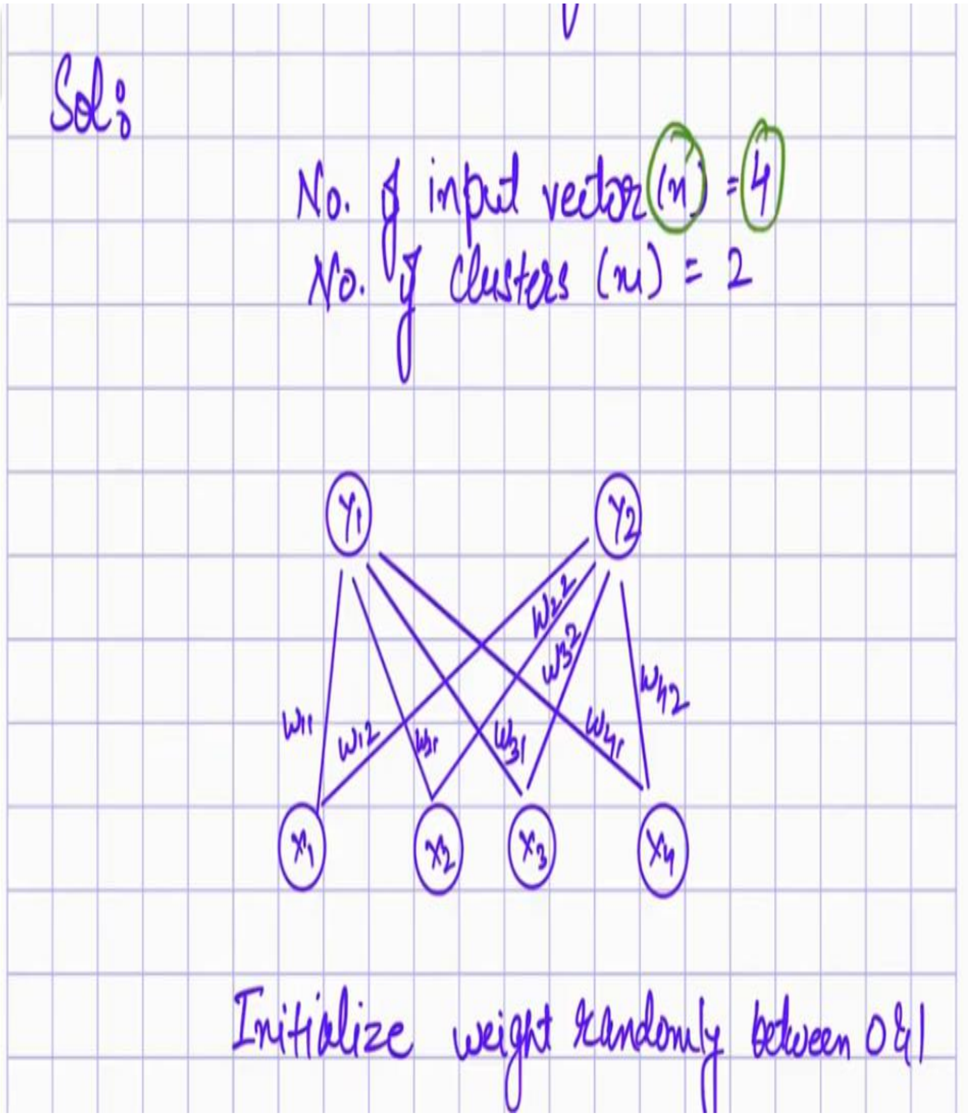
$$X \Rightarrow [0\ 0\ 1\ 1] \leftarrow$$
$$\Rightarrow [1\ 0\ 0\ 0] \leftarrow$$
$$\Rightarrow [0\ 1\ 1\ 0] \leftarrow$$
$$\Rightarrow [0\ 0\ 0\ 1] \leftarrow$$

No. of Clusters to be formed are 2.
Assume initial learning rate as 0.5.

Sol:

No. of input vector $(n) = 4$
No. of clusters $(m) = 2$

Initialize



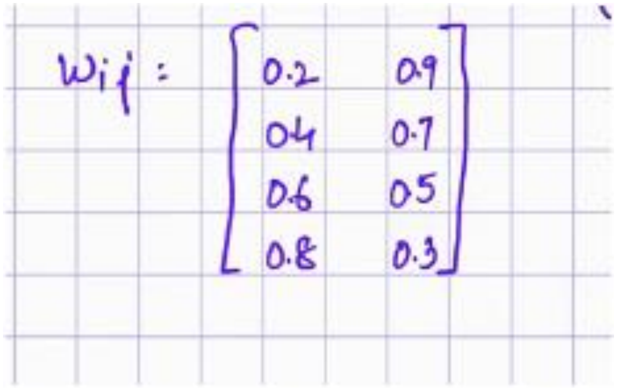
Sol:

No. of input vector (n) = 4

No. of clusters (m) = 2

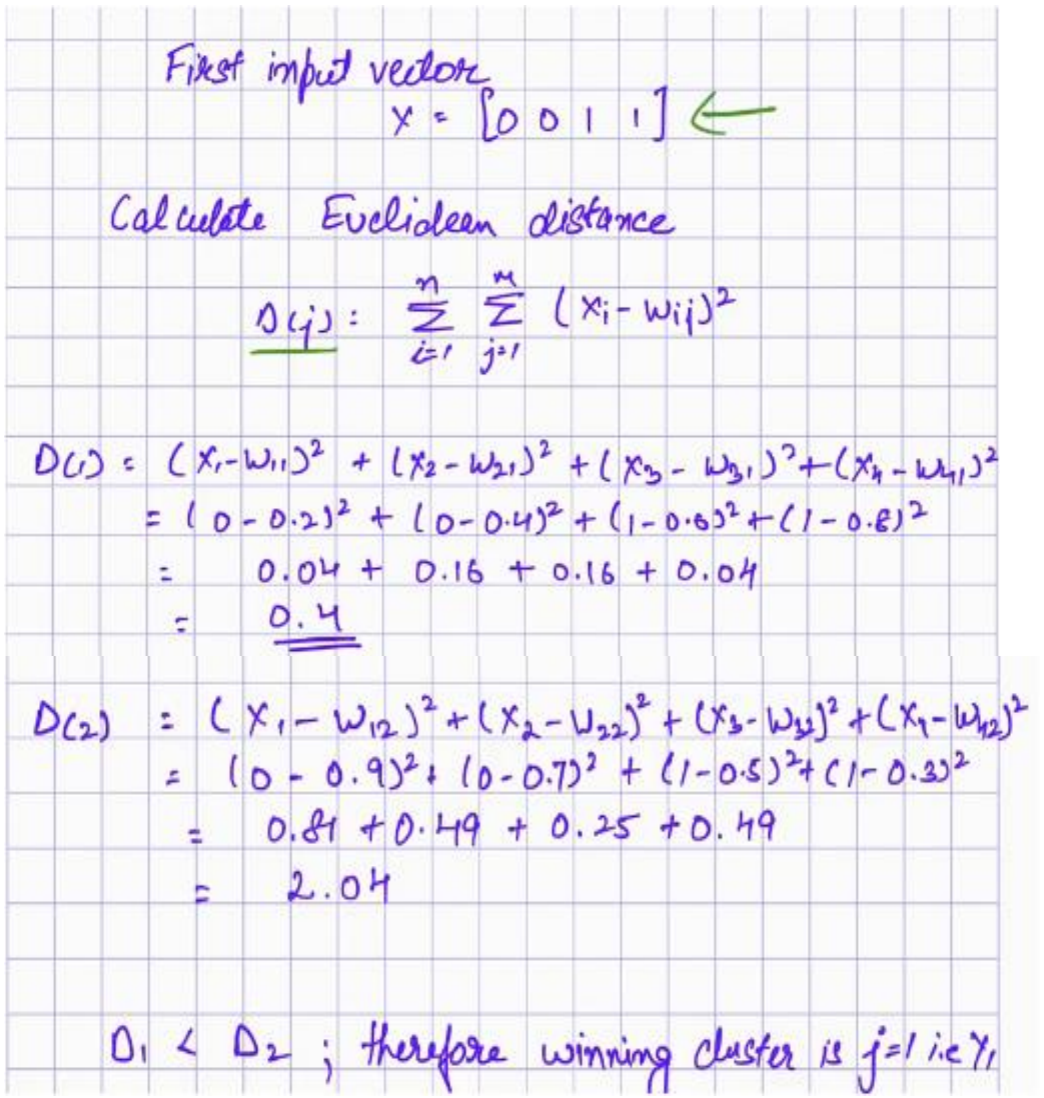
Initialize weight randomly between 0 & 1

$\rightarrow$



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

Distance Calculation



First input vector

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

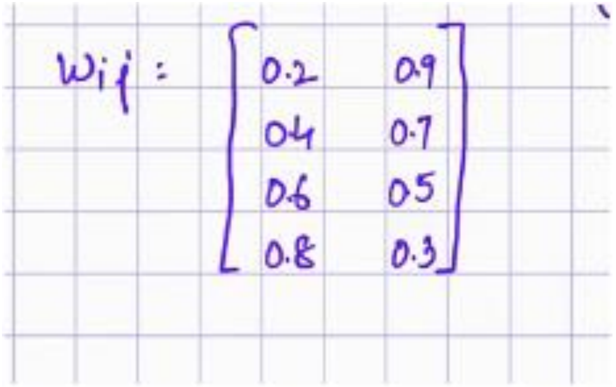
Calculate Euclidean distance

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

$$D(1) = (x_1 - w_{11})^2 + (x_2 - w_{21})^2 + (x_3 - w_{31})^2 + (x_4 - w_{41})^2$$
$$= (0 - 0.2)^2 + (0 - 0.4)^2 + (1 - 0.6)^2 + (1 - 0.8)^2$$
$$= 0.04 + 0.16 + 0.16 + 0.04$$
$$= \underline{\underline{0.4}}$$

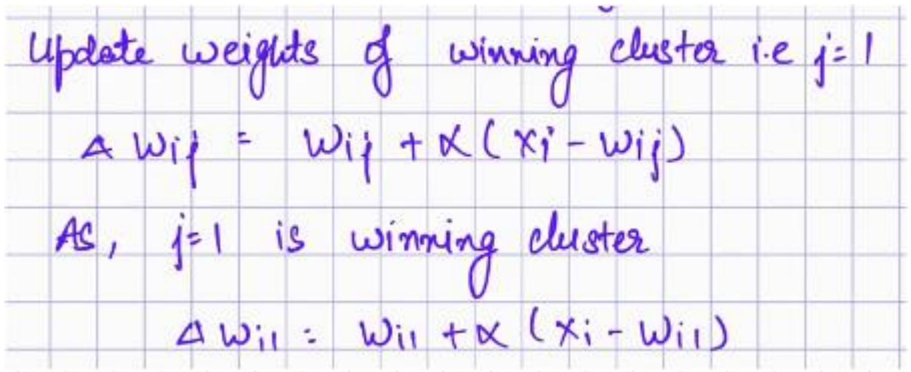
$$D(2) = (x_1 - w_{12})^2 + (x_2 - w_{22})^2 + (x_3 - w_{32})^2 + (x_4 - w_{42})^2$$
$$= (0 - 0.9)^2 + (0 - 0.7)^2 + (1 - 0.5)^2 + (1 - 0.3)^2$$
$$= 0.81 + 0.49 + 0.25 + 0.49$$
$$= 2.04$$

$D_1 < D_2$; therefore winning cluster is $j=1$ i.e Y_1



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

Update weights

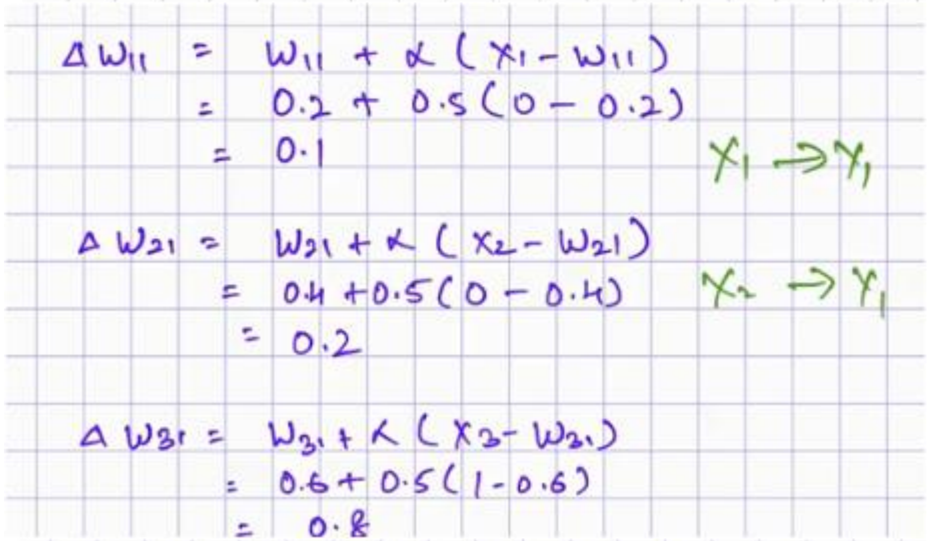


Update weights of winning cluster i.e $j=1$

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

As, $j=1$ is winning cluster

$$\Delta W_{i1} = W_{i1} + \alpha(x_i - W_{i1})$$



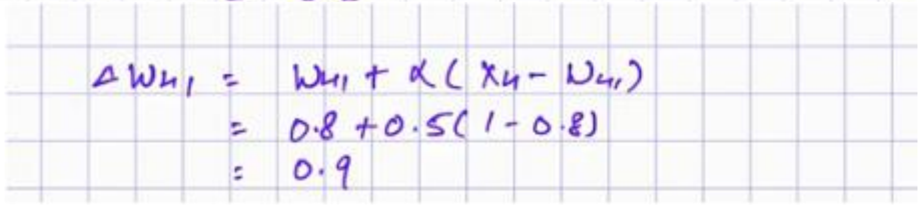
$$\Delta W_{11} = W_{11} + \alpha(x_1 - W_{11}) = 0.2 + 0.5(0 - 0.2) = 0.1$$

$X_1 \rightarrow Y_1$

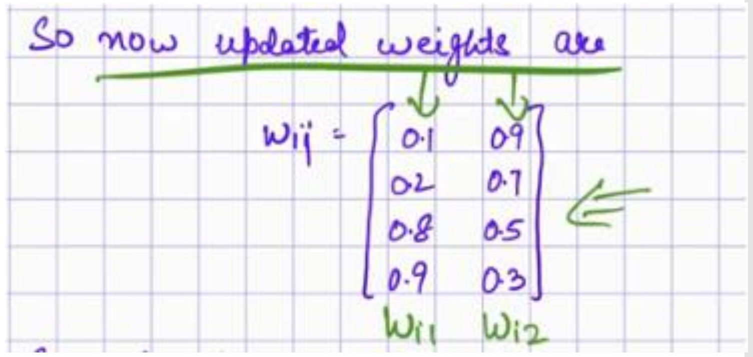
$$\Delta W_{21} = W_{21} + \alpha(x_2 - W_{21}) = 0.4 + 0.5(0 - 0.4) = 0.2$$

$X_2 \rightarrow Y_1$

$$\Delta W_{31} = W_{31} + \alpha(x_3 - W_{31}) = 0.6 + 0.5(1 - 0.6) = 0.8$$



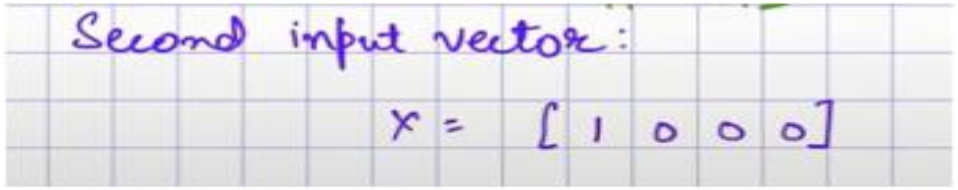
$$\Delta W_{41} = W_{41} + \alpha(x_4 - W_{41}) = 0.8 + 0.5(1 - 0.8) = 0.9$$



So now updated weights are

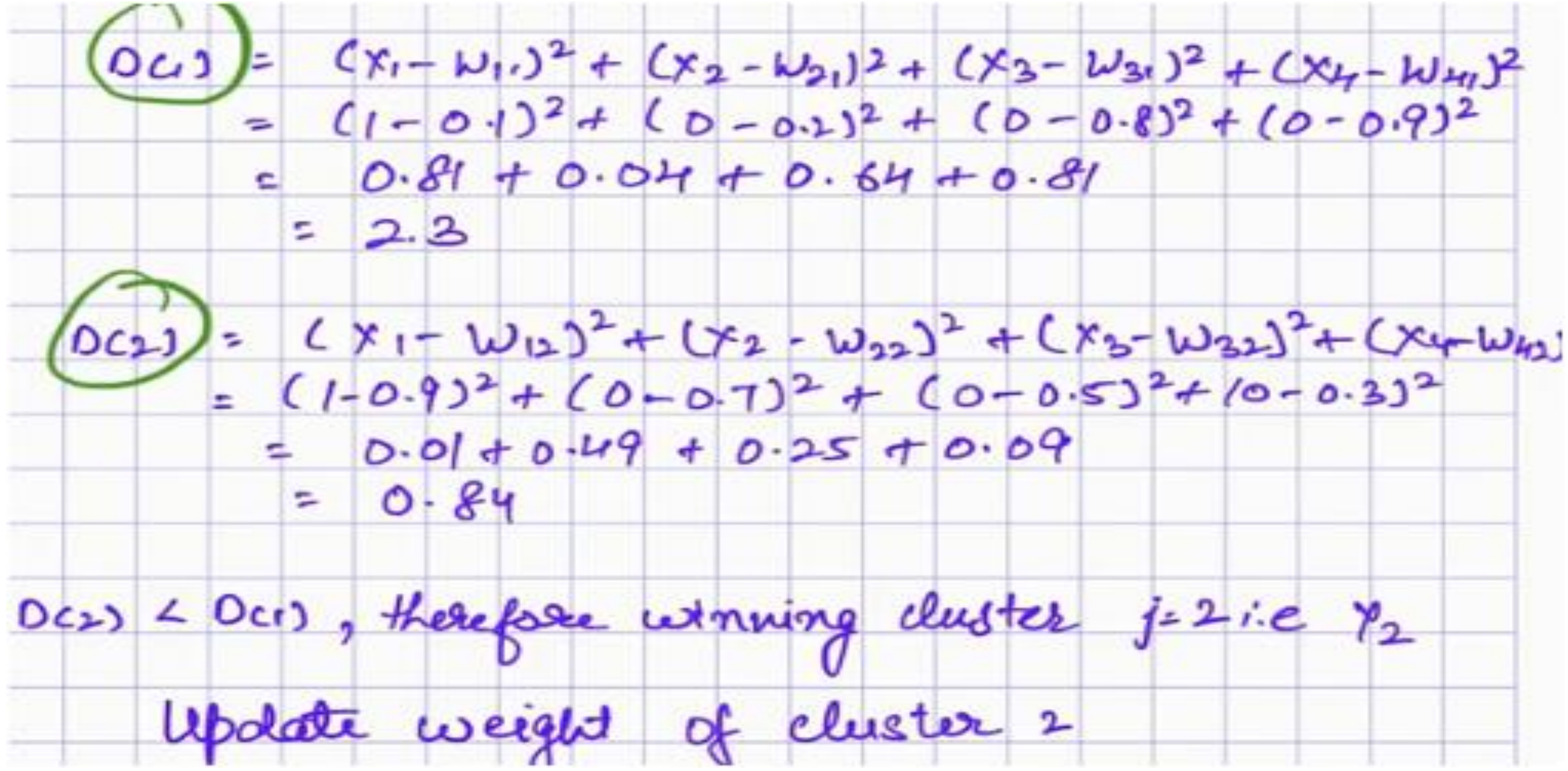
$$W_{ij} = \begin{array}{cc} \downarrow & \downarrow \\ \left[\begin{array}{cc} 0.1 & 0.9 \\ 0.2 & 0.7 \\ 0.8 & 0.5 \\ 0.9 & 0.3 \end{array}\right] \\ W_{i1} & W_{i2} \end{array} \Leftarrow$$

Second vector



Second input vector:

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



$D(1) = (x_1 - w_{11})^2 + (x_2 - w_{21})^2 + (x_3 - w_{31})^2 + (x_4 - w_{41})^2$

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

$= 0.81 + 0.04 + 0.64 + 0.81$

$= 2.3$

$D(2) = (x_1 - w_{12})^2 + (x_2 - w_{22})^2 + (x_3 - w_{32})^2 + (x_4 - w_{42}$

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

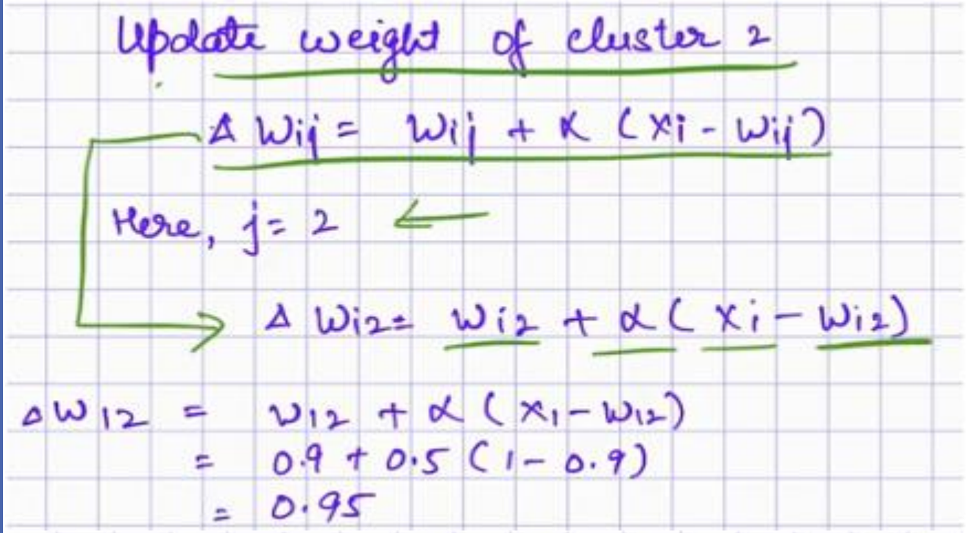
$= 0.01 + 0.49 + 0.25 + 0.09$

$= 0.84$

$D(2) < D(1)$, therefore winning cluster $j = 2$ i.e Y_2

Update weight of cluster 2

update



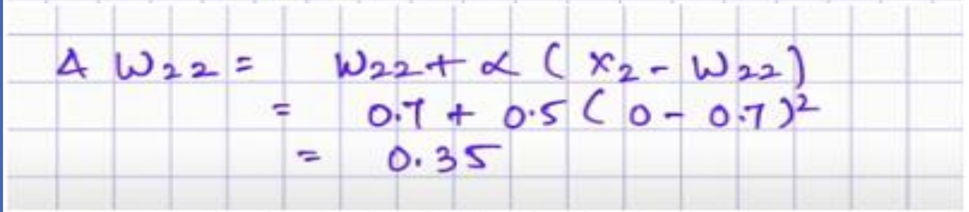
Update weight of cluster 2

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

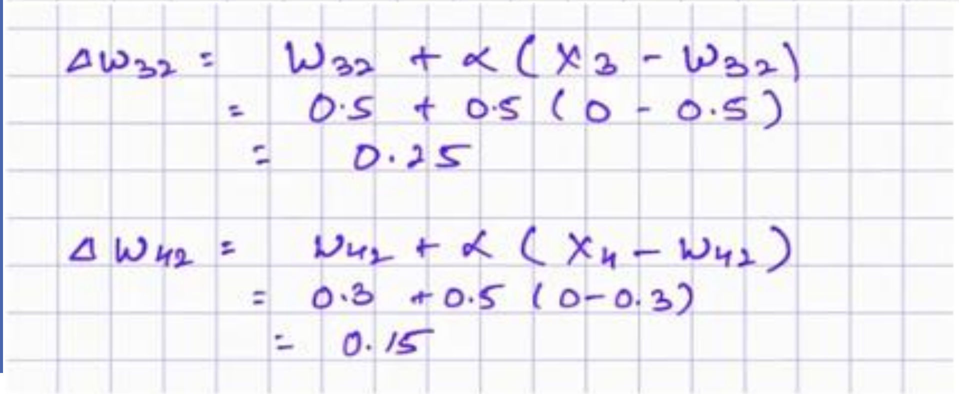
Here, $j = 2$ $\leftarrow$

$\Delta W_{i2} = \underline{W_{i2}} + \underline{\alpha} (\underline{x_i} - \underline{W_{i2}})$

$$\Delta W_{12} = W_{12} + \alpha (x_1 - W_{12})$$
$$= 0.9 + 0.5 (1 - 0.9)$$
$$= 0.95$$

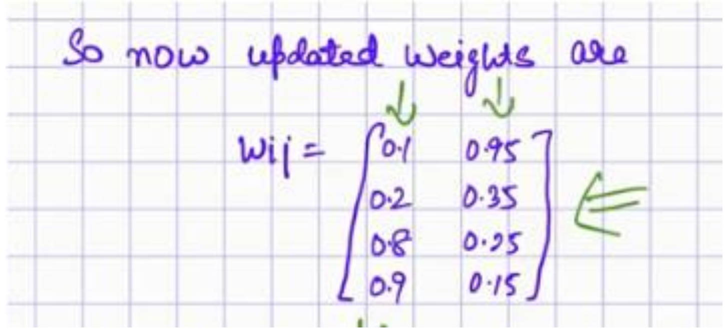


$$\Delta W_{22} = W_{22} + \alpha (x_2 - W_{22})$$
$$= 0.7 + 0.5 (0 - 0.7)^2$$
$$= 0.35$$



$$\Delta W_{32} = W_{32} + \alpha (x_3 - W_{32})$$
$$= 0.5 + 0.5 (0 - 0.5)$$
$$= 0.25$$

$$\Delta W_{42} = W_{42} + \alpha (x_4 - W_{42})$$
$$= 0.3 + 0.5 (0 - 0.3)$$
$$= 0.15$$

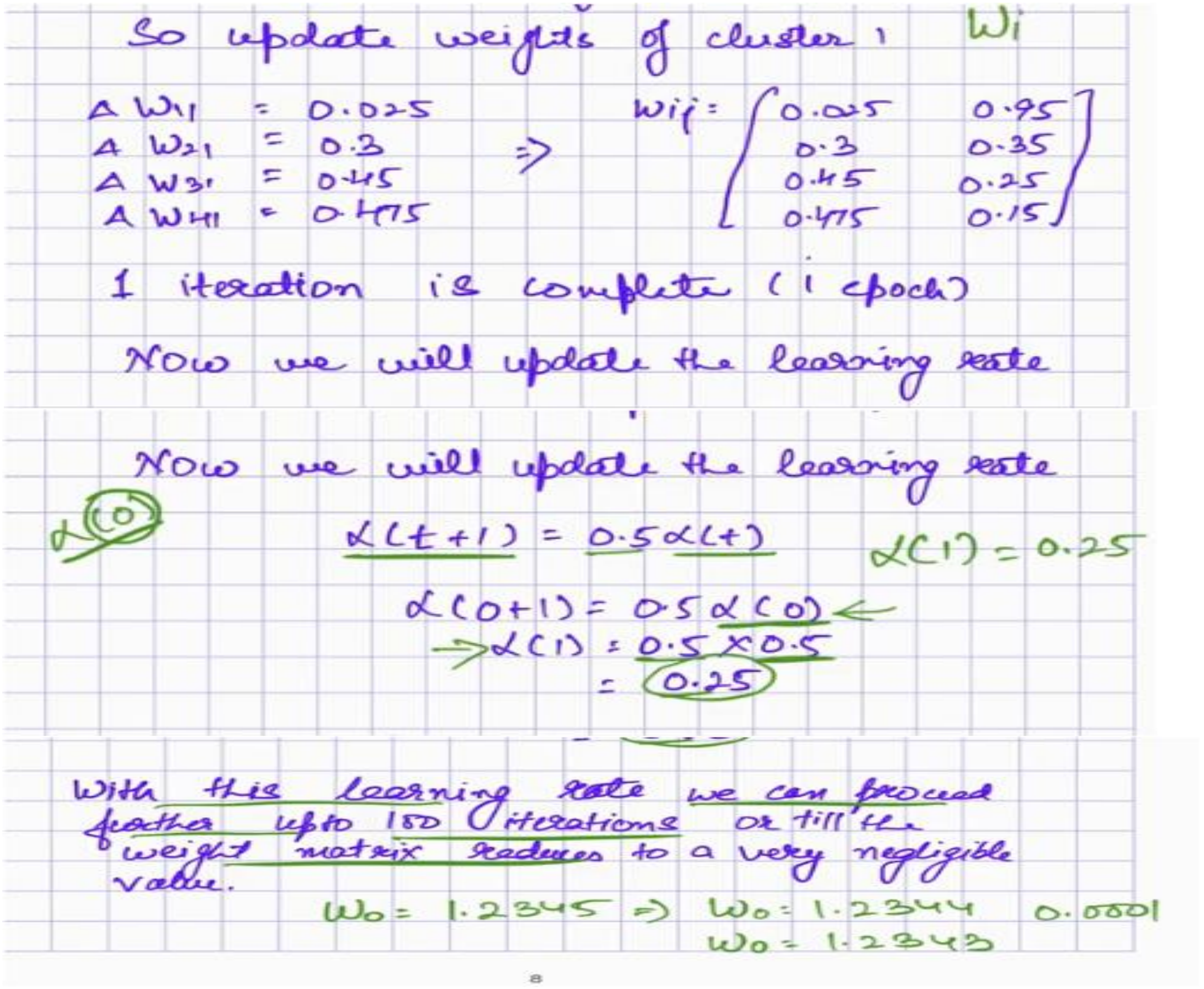


So now updated weights are

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

Calculate the distance and update the weights for Input Vector 3 and 4

Update Learning rate



So update weights of cluster 1 W_i

$\Delta W_{11} = 0.025$
$\Delta W_{21} = 0.3$
$\Delta W_{31} = 0.45$
$\Delta W_{41} = 0.475$

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

1 iteration is complete (1 epoch)

Now we will update the learning rate

Now we will update the learning rate

$\boxed{\alpha(0)}$

$$\underline{\alpha(t+1)} = \underline{0.5\alpha(t)} \qquad \alpha(1) = 0.25$$

$$\alpha(0+1) = 0.5\,\alpha(0) \leftarrow$$
$$\rightarrow \alpha(1) = \underline{0.5 \times 0.5}$$
$$= \boxed{0.25}$$

With this learning rate we can proceed further upto 150 iterations or till the weight matrix reduces to a very negligible value.

$W_0 = 1.2345 \Rightarrow W_0 = 1.2344 \quad 0.0001$
$W_0 = 1.2343$

8