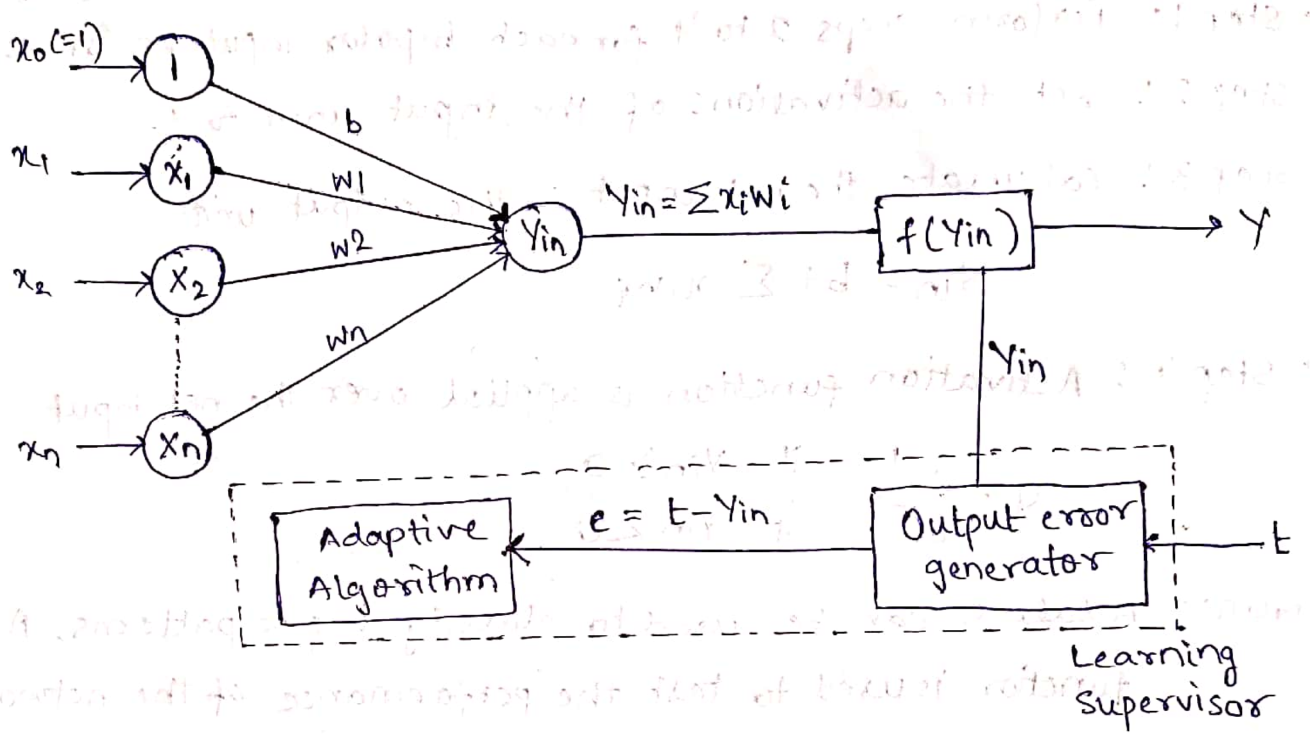
Q.1 With a neat architecture, write the training algorithm and testing algorithm of Adaline network.

A.1 (I) Architecture of ADALINE



(II) Training Algorithm of ADALINE

- Step 0: Weights & bias are set to some random values but not to zero.
- Step 1: Perform step 2 to 6 when stopping condition is false
- Step 2: Perform steps 3 to 5 for each bipolar training pair s:t
- Step 3: Set activations for input units i=1 to n ($x_i = s_i$)
- Step 4: Calculate the net input to the output unit

$$Y_{in} = b + \sum_{i=1}^{n} x_i w_i$$

- Step 5: Update the weights & bias for i=1 to n

$$w_i(new) = w_i(old) + \alpha(t - Y_{in})x_i$$
$$b(new) = b(old) + \alpha(t - Y_{in})$$

- Step 6: If the highest weight change that occurred during training is smaller than a specified tolerance then stop the training process, else continue.

NOTE: The learning rate can be between 0.1 & 1.0.

(III) Testing Algorithm for ADALINE

- Step 0: Initialize the weights (obtained from the training algorithm)
- Step 1: Perform steps 2 to 4 for each bipolar input vector x.
- Step 2: Set the activations of the input units to x.
- Step 3: Calculate the net input to the output unit

$$y_{in} = b + \sum x_i w_i$$

- Step 4: Activation function is applied over the net input.

$$y = \begin{cases} 1 & \text{if } y_{in} \geq 0 \\ -1 & \text{if } y_{in} < 0 \end{cases}$$

NOTE: Adaline can be used to classify input patterns. A step function is used to test the performance of the network.

Q.2 A Kohonen self-organizing map is shown with weights in Fig. 1

(i) Use the square of the Euclidean distance find the cluster unit C_j that is closest to the input vector (0.7, 0.9)

(ii) Using a learning rate of 0.1, find the new weights for unit C_j.

(iii) Find the new weights for C_{j-1} & C_{j+1} even if they are allowed to learn.

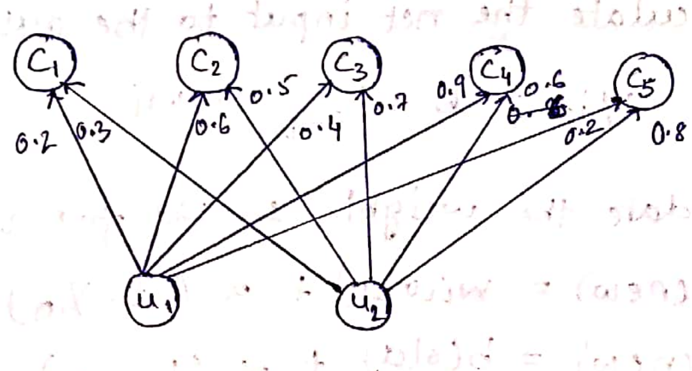


Figure 1

A.2 (i)

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

$n = 2$; $m = 5$

For $x = [0.7 \quad 0.9]$

$D(j) = \Sigma (w_{ij} - x_i)^2$

$D(1) = \Sigma (w_{i1} - x_i)^2$

$= (0.2 - 0.7)^2 + (0.3 - 0.9)^2$

$= 0.61$

$D(2) = \Sigma (w_{i2} - x_i)^2$

$= (0.6 - 0.7)^2 + (0.5 - 0.9)^2$

$= 0.17$

$D(3) = \Sigma (w_{i3} - x_i)^2$

$= (0.4 - 0.7)^2 + (0.7 - 0.9)^2$

$= 0.13$

$D(4) = \Sigma (w_{i4} - x_i)^2$

$= (0.9 - 0.7)^2 + (0.6 - 0.9)^2$

$= 0.13$

$D(5) = \Sigma (w_{i5} - x_i)^2$

$= (0.2 - 0.7)^2 + (0.8 - 0.9)^2$

$= 0.26$

In this case, $D(3) = D(4) = 0.13$ is minimum. Thus, winner unit is the one with the smallest index i.e $D(3)$

$\therefore$ Cluster unit C_3 is closest to input vector $(0.7, 0.9)$

(ii) C_3 i.e $J = 3$ is winner unit with $\alpha = 0.1$. The weight updation formula is given by

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

Substituting $J=3$ in the above equation, we obtain

$$w_{i3}(new) = w_{i3}(old) + \alpha(x_i - w_{i3}(old))$$

For $i = 1, 2$

$$w_{13}(new) = 0.4 + 0.1(0.7 - 0.4) = 0.43$$

$$w_{23}(new) = 0.7 + 0.1(0.9 - 0.7) = 0.72$$

The updated weight matrix for the winning unit is given by,

$$w_{ij} = \begin{bmatrix} 0.2 & 0.6 & 0.43 & 0.9 & 0.2 \\ 0.3 & 0.5 & 0.72 & 0.6 & 0.8 \end{bmatrix}$$

(iii) New weights for C_{J-1} i.e C_2 & C_{J+1} i.e C_4

For C_2 :-

$$w_{i2}(new) = w_{i2}(old) + \alpha(x_i - w_{i2}(old)) \quad \text{For } i=1,2$$

$$w_{12}(new) = 0.6 + 0.1(0.7 - 0.6) = 0.61$$

$$w_{22}(new) = 0.5 + 0.1(0.9 - 0.5) = 0.54$$

updated weight matrix :-

$$w_{ij} = \begin{bmatrix} 0.2 & 0.61 & 0.4 & 0.9 & 0.2 \\ 0.3 & 0.54 & 0.7 & 0.6 & 0.8 \end{bmatrix}$$

For C_4 :-

$$w_{i4}(new) = w_{i4}(old) + \alpha(x_i - w_{i4}(old)) \quad \text{For } i=1,2$$

$$w_{14}(new) = 0.9 + 0.1(0.7 - 0.9) = 0.88$$

$$w_{24}(new) = 0.6 + 0.1(0.9 - 0.6) = 0.63$$

Updated weight matrix :-

$$W_{ij} = \begin{bmatrix} 0.2 & 0.6 & 0.4 & 0.88 & 0.2 \\ 0.3 & 0.5 & 0.7 & 0.63 & 0.8 \end{bmatrix}$$

— × —