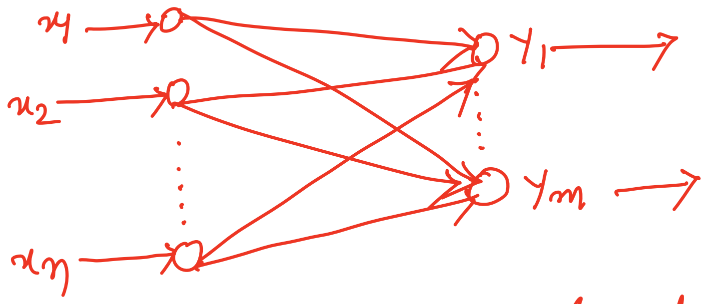
lec-04
24/08/21

→ Perceptron model
→ Adaline model

Perceptron :



→ The perceptron is a computational model of the retina of the eye and hence it is named as "perceptron".

→ The network comes under single layer feed forward network and are named as simple perceptrons.

Perceptron learning algo:

Step 1: Initialize all wts $w_0, w_1 \ldots w_n$.

2: Set the learning rate η s.t $0 < \eta < 1$ and threshold θ.

3: for each training pair $s : t$ do steps 4–8.

4: activate the input units $x_i = s_i$, for $i = 0, \ldots n$

5: Compute the net input to the output unit

$$Y_{in} = \sum_{i=0}^{n} w_i x_i$$

6. Compute the activation of the o/p unit using the function :

$$Y_{out} = \begin{cases} 1 & \text{if } Y_{in} > \theta \\ 0 & \text{if } -\theta \le Y_{in} \le \theta \\ -1 & \text{if } Y_{in} < -\theta \end{cases}$$

7. If there is an error i.e. $Y_{out} \neq T$, then adjust the weight as follows:

$$\boxed{\begin{aligned} w_i(new) &= w_i(old) + \eta \times t \times x_i \\ b(new) &= b(old) + \eta \times t \end{aligned}}$$

If however, no error has occurred, the wts. are kept unchanged.

8. If there is no error i.e. $Y_{out} = T$, for the entire set of training pairs then stop else goto step 2.

$w_1 = 0 + 1 \times 1 \, 1$

$0 + 01 \times 1 \times 1$

(Assume $\theta = 0.2$)
$\eta = 01$

Numerical:

x_0	x_1	x_2	x_3	Y_{in} Net I/P	Net O/P Y_{out}	Target	wts. w_0 w_1 w_2 w_3
							0 0 0 0
1	1	1	1	0	0	1	1 1 1 1
1	-1	1	1	2	1	-1	0 2 0 0
1	1	-1	1	2	01	-1	-1 1 1 -1
1	1	1	-1	2	01	-1	-2 0 0 0

Iteration 1

$w_0 = 0 + 1 \times 1 \times 1$

$w_1 = 0 + 1 \times 1 \times 1$

$w_2 = 0 + 1 \times 1 \times 1$

$w_3 = 0 + 1 \times 1 \times 1$

Iteration 2

Err 01 $= (Y_{out} - T)^2 = 1$

02 $= (1 - (-1))^2 = 4$

03 $= (1 - (-1))^2 = 4$

04 $= (1 - (-1))^2 = 4$

$$M.S.E = \frac{1+4+4+4}{4}$$

$$= \frac{13}{4}$$

After 1st Epoch: $R.M.S.E = \sqrt{13/4} = 1.8$

$rmse < 0.01$

→ The input vectors are allowed to be either binary / or bipolar. However, the O/P must be in bipolar form.

→ The bias wt. w_0 is adjustable but the threshold θ used in the activation function is fixed.

Adaline : (Adaptive linear neural element network)

[By Bernard Widrow of Stanford Univ.].

→ Here there is only one O/P neuron and the O/P are bipolar (−1 or +1).

→ The i/p could be binary, bipolar or real valued.

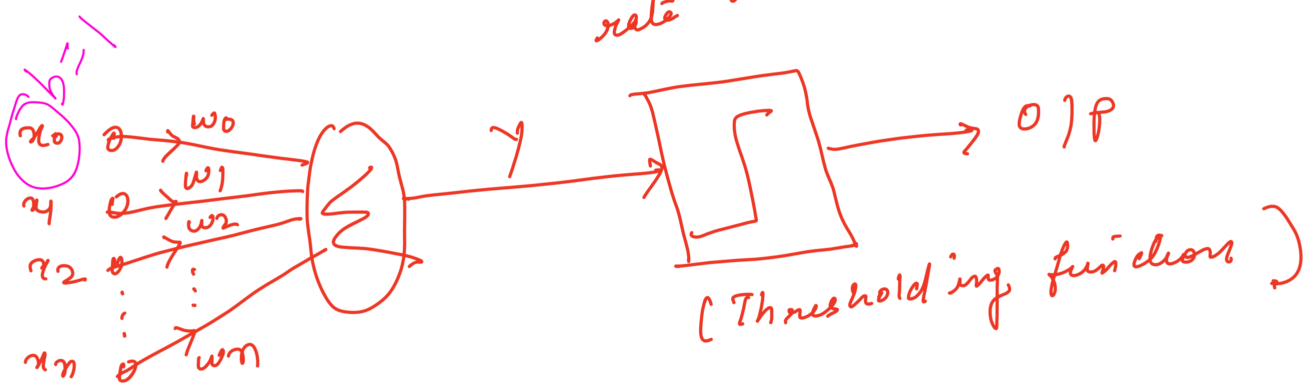
→ If the weighted sum of inputs is greater than or equal to 0, then O/P is +1 else −1.

→ Devised by Widrow Hoff [1960] and the learning algo. is known as Least Mean Square (LMS) or Delta rule :

Imp.

$$\boxed{W_i(new) = w_i(old) + \alpha(t-y)x_i}$$

α → learning rate; t → Target; y → computed O/P; x_i → I/P



Adaline Learning algo :

Step 1 : Set wt. and bias wt. to some random values but not zero. Set the learning parameter α.

2 : Perform step 3–7 till the stopping criteria is not met.

3 : Perform step 4–6 for each bipolar training pair s : t :

4 : Set activation for i/p unit i = 1 to n

$$x_i = s_i$$

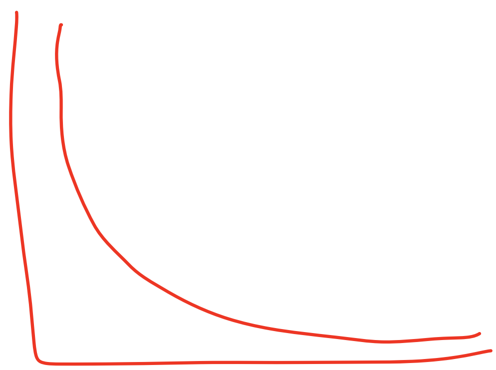
5: Calculate the net i/p to the o/p unit

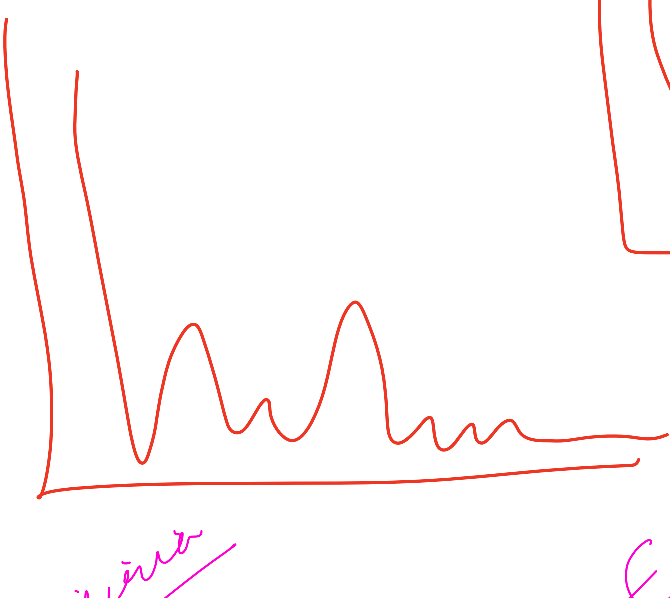
$$y_{in} = b + \sum_{i=1}^{n} w_i x_i$$

6: update the wt. and bias for $i = 1$ to n

$$\boxed{\begin{aligned} w_i(\text{new}) &= w_i(\text{old}) + \alpha(t - y_{in})x_i \\ b(\text{new}) &= b(\text{old}) + \alpha(t - y_{in}) \end{aligned}}$$

7. If the highest wt. change that occurred during training is smaller than a specified tolerance, then stop the training process else continue.





Stopping criteria

1) $t \rightarrow W$

$t+1 \rightarrow W'$

$W - W' \le \epsilon$ OR

2) $rmse \le 0.001$ OR

3) # epochs < 500