Q) a. Use Adaline network to train OR function with bipolar inputs and targets. perform two epochs of training.

b. Explain the training algorithm used in adaptive linear neuron.

x_1	x_2	t
-1	-1	-1
-1	1	1
1	-1	1
1	1	1

Assume,

$w_1 = w_2 = b = 0.1$

$\alpha = 0.1$

$y_{in} = b + w_1 x_1 + w_2 x_2$

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

$b(new) = b(old) + \alpha (t - y_{in})$

EPOCH 1.

case 1 : $x_1 = -1$, $x_2 = -1$, $t = -1$

$y_{in} = b + w_1 x_1 + w_2 x_2 = 0.1 + (0.1 \times -1) + (0.1 \times -1) = -0.1$

Error, $t - y_{in} = -1 - (-0.1) = -0.9$

weight updates,

$w_1(new) = w_1(old) + \alpha(t - y_{in}) x_1 = 0.1 + 0.1(-1 - -0.1) \times -1 = 0.19$

$w_2(new) = w_2(old) + \alpha(t - y_{in}) x_2 = 0.1 + 0.1(-1 - -0.1) \times -1 = 0.19$

$b = b(old) + \alpha(t - y_{in}) = 0.1 + 0.1(-1 - -0.1) = 0.01$

case 2: $x_1 = -1$, $x_2 = 1$, $t = 1$

$Y_{in} = 0.01 + (0.19)(-1) + (0.19)(1) = 0.01$

error $= 1 - 0.01 = 0.99$

weight update:

$w_1 = 0.19 + 0.1(0.99)(-1) = 0.091$

$w_2 = 0.19 + 0.1(0.99)(1) = 0.289$

$b = 0.01 + 0.1(0.99) = 0.109$.

case 3: $x_1 = +1$, $x_2 = -1$, $t = 1$

$y_{in} = 0.109 + (0.091)(1) + (0.289)(-1) = -0.089$

Error $= 1 - (-0.089) = 1.089$

update:

$w_1 = 0.091 + 0.1(1.089)(1) = 0.1999$

$w_2 = 0.289 + 0.1(1.089)(-1) = 0.1801$

$b = 0.109 + 0.1(1.089) = 0.2179$.

case 4: $x_1 = 1$, $x_2 = 1$, $t = 1$

$y_{in} = 0.2179 + 0.1999 + 0.1801 = 0.5979$

Error $= 1 - 0.5979 = 0.4021$.

update:

$w_1 = 0.1999 + 0.1(0.4021) = 0.24011$

$w_2 = 0.1801 + 0.1(0.4021) = 0.22031$

$b = 0.2179 + 0.1(0.4021) = 0.2581$

End of Epoch 1.

$w_1 = 0.24011$, $w_2 = 0.22031$, $b = 0.2581$

EPOCH 2:

$w_1 = 0.24011$, $w_2 = 0.22031$ $b = 0.25811$

case 1: $x_1 = -1$, $x_2 = -1$, $t = -1$

$y_{in} = 0.25811 - 0.24011 - 0.22081 = -0.20231$

Error $= -1 - (-0.20231) = -0.79769$

update:

$w_1 = 0.319879$

$w_2 = 0.300079$

$b = 0.178341$

case 2: $x_1 = -1$, $x_2 = 1$, $t = 1$

$y_{in} = 0.178341 - 0.319879 + 0.300079 = 0.158541$.

Error $= 1 - 0.158541 = 0.841459$

update:

$w_1 = 0.2357331$

$w_2 = 0.3842249$

$b = 0.2624869$

case 3: $x_1 = 1$, $x_2 = -1$, $t = 1$

$y_{in} = 0.2624869 + 0.2357331 - 0.3842249 = 0.1139951$

error $= 1 - 0.1139951 = 0.8860049$.

update:

$w_1 = 0.32483359$

$w_2 = 0.29562441$

$b = 0.35108739$

case 4: $x_1 = 1$, $x_2 = 1$, $t = 1$

$y_{in} = 0.35108739 + 0.32483359 + 0.29562441 = 0.97104539$

$y_{in} = 0.35108789 + 0.32482859 + 0.29562141 = 0.97104589$

error = $1 - 0.97104589 = 0.02895411$.

update:

$w_1 = 0.32722905$

$w_2 = 0.29851981$

$b = 0.35898285$

final weight after 2 epochs : $w_1 = 0.3272$, $w_2 = 0.2985$, $b = 0.3540$.

(5) step 0: weights and bias are set to some random values but $\neq 0$. set the learning rate parameter α (The range can be between 0.1 and 1.0).

step 1: perform step 2-6 when stopping condition is false.

step 2: perform step 3-5 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 and 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 height weight change that occured during training is smaller than a specified tolerence then stop training process, else continue. This is the test for stopping condition of a network.