Practical - 3 Assignment

① Perceptron

Assume weight vector of initial decision boundary $W^T X = 0$

$W = [1, 1]$

$\Rightarrow x_1 + x_2 = 0$

$b = 0$

$y_{in} = w_i^T x_i + b = w_1 x_1 + w_2 x_2 + b$

Assume learning rate α 1

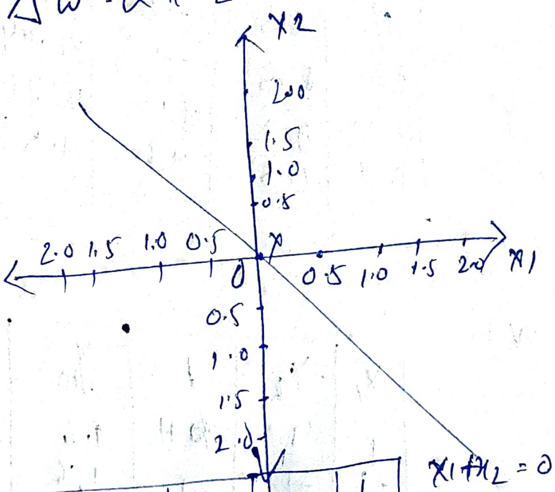
$y = \begin{cases} 1, & \text{if } y_n > 0 \\ 0, & y_n = 0 \\ -1, & y_n < 0 \end{cases}$

$\Delta w_1 = \alpha t x_1$

$\Delta b = \alpha t$

$\Delta w = \alpha t x_2$

(i)



(i)

x_1	x_2	class (t)	y_{in}	y	Δw_1	Δw_2	Δb	w_1	w_2	b
+1	-1	+1	2	+1	0	0	0	1	1	0
-1	0.5	-1	-2	-1	0	0	0	1	1	0
0	0.5	-1	0.5	+1	0	-0.5	-1	1	0.5	-1
0.7	0.2	-1	-0.05	-1	0	0	0	1	0.5	-1
0.2	1	+1	-0.7	-1	0.2	0.2	1	1.2	0.7	0
0.9	1.8	+1	1.43	+1	0	0	0	1.2	0.7	0

(ii)

x_1	x_2	t	y_{in}	y	Δw_1	Δw_2	Δb	w_1	w_2	b
1	1	+1	1.1	+1	0	0	0	1.2	0.7	0
-1	-1	-1	-1.9	-1	0	0	0	1.2	0.7	0
0	0.5	-1	0.35	+1	0	-0.5	-1	1.2	0.2	-1
0.1	0.5	-1	-0.78	-1	0	0	0	1.2	0.2	-1
0.2	0.2	+1	-0.72	-1	0.2	0.2	1	1.2	0.4	0
0.9	0.5	+1	1.46	-1	0	0	0	1.4	0.4	0
				+1				1.7		

(iii)

x_1	x_2	t	y_{in}	y	Δw_1	Δw_2	Δb	w_1	w_2	b
1	1	+1	1.8	+1	0	0	1.40	1.4	0.4	0
-1	-1	-1	-1.8	-1	0	0	1.10	1.4	0.4	0
0	0.5	-1	0.2	+1	0	-0.5	-1	1.4	-0.1	-1
0.1	0.5	-1	-0.81	-1	0	0	0	1.4	-0.1	-1
0.2	0.2	+1	0.74	-1	0.2	0.2	1	1.6	0.1	0
0.9	0.5	+1	1.94	+1	0	0	0	1.6	0.1	0

(iv)

x_1	x_2	t	y_{in}	y	Δw_1	Δw_2	Δb	w_1	w_2	b
1	1	+1	1.7	+1	0	0	0	1.6	0.7	0
-1	-1	-1	-1.7	-1	0	0	0	1.6	0.7	0
0	0.5	-1	0.05	+1	0	-0.5	-1	1.6	-0.4	-1
0.1	0.5	-1	-1.04	-1	0	0	0	1.6	-0.4	-1
0.2	0.2	+1	0.76	-1	0.2	0.2	1	1.8	-0.2	0
0.9	0.5	+1	1.52	+1	0	0	0	1.8	-0.2	0

(v)

x_1	x_2	t	y_{in}	y	Δw_1	Δw_2	Δb	w_1	w_2	b
1	1	+1	1.6	+1	0	0	0	1.8	-0.2	0
-1	-1	-1	-1.6	-1	0	0	0	1.8	-0.2	0
0	0.5	-1	-0.1	-1	0	0	0	1.8	-0.2	0
0.1	0.5	-1	0.08	+1	-0.1	-0.5	-1	1.7	-0.7	-1
0.2	0.2	+1	-0.8	-1	0.2	0.2	1	1.9	-0.5	0
0.9	0.5	+1	1.46	+1	0	0	0	1.9	-0.5	0

(vi)

x_1	x_2	t	y_{in}	y	Δw_1	Δw_2	Δb	w_1	w_2	b
1	1	+1	1.4	+1	0	0	0	1.9	-0.5	0
-1	-1	-1	-1.4	-1	0	0	0	1.9	-0.5	0
0	0.5	-1	-0.25	-1	0	0	0	1.9	-0.5	0
0.1	0.5	-1	-0.06	-1	0	0	0	1.9	-0.5	0
0.2	0.2	+1	0.28	+1	0	0	0	1.9	-0.5	0
0.9	0.5	+1	1.46	+1	0	0	0	1.9	-0.5	0

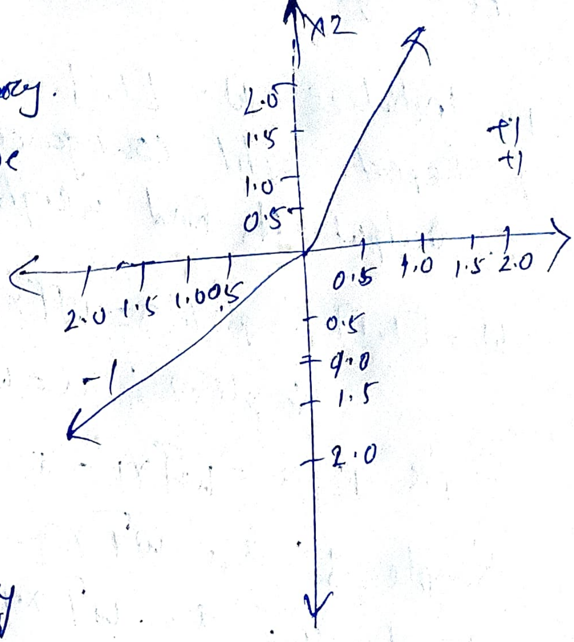
The perceptron learning algorithm converged in 6 steps.

The final weight vector of the decision boundary $w = [1.9\ -0.5]$

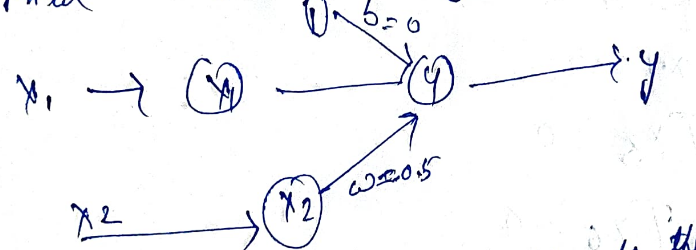
$1.9 x_1 + (-0.5) x_2 = 0 \Rightarrow \boxed{1.9 x_1 - 0.5 x_2 = 0}$

Now plot the final decision boundary.

We can see that $1.9x - 0.5x_2 = 0$ line seperates the two classes correctly.



final decision boundary



Neural network corresponding to the perceptron.

Initial decision boundary $w^T x = 0$

Assuming weight vector of initial decision boundary as $w = [1, 1]$, ~~Solve~~

① Steps perceptron learning algorithm:

x_1	$x_1 = (x_1 - 0)$	x_1^2	x_2	$x_2 = (x_2 - 0.5)$	x_2^2	$x_1 x_2$
1	1	1	1	0.5	0.25	1
-1	-1	1	-1	-0.5	-0.25	1
0	0	0	0.5	0	0	0
0.1	0.1	0.01	0.5	0	0	0.05
0.2	0.2	0.04	0.2	-0.3	-0.09	0.04
0.9	0.9	0.81	0.5	0	0	0.45

The perceptron Algorithm (PLA) updates the weight vector whenever it makes a misclassification on a training example.

$w \leftarrow w + \alpha y_i x_i$

To determine the convergence of the PLA, we need to run the algorithm on the given training samples until all samples are correctly classified by the decision boundary.

The algorithm can be summarized as.

1. Initialize $w = [1,1]$
2. Repeat until convergence
3. Output the final weight vector

$w = [1,1]$ Sample 1: $a = w^T x_1 = 2$
Sample 2: $a = w^T x_2 = 0$ update $w = [0,0]$
Sample 1: $a = w^T x_1 = 0$ Sample 2: $a = w^T x_2 = 0$
Sample 3: $a = w^T x_3 = 0$
Sample 4: $a = w^T x_4 = 0$
" 5: $a = w^T x_5 = 0$
" 6: $a = w^T x_6 = 0$

Convergence reached after 1 update.

Hence the PLA ~~conv~~ converges in 1 step for the given training samples with the initial weight vector of $w = [1,1]$.

(2) To determine final decision boundary, we can apply the perceptron learning Algorithm (PLA) on the given training samples. The algorithm updates the weight vector whenever it makes a misclassification on a training example until all the samples are correctly classified by the decision boundary.

Step-1 Initialize the weight vector $w = [1,1]$
Step-2 For each training example (x_i, y_i).

Compute the activation $a = w^T x$

If the prediction is incorrect ($y a \leq 0$)

Update the weight vector $w = w + \alpha y x$

Repeat step 2 until all training examples are correctly classified by the decision boundary.

Using a learning rate of $\alpha = 1$, the PLA updates the weight vector as follows.

Initial weight vector $w = [1, 1]$

Sample 1: $x = [1, 1]$, $y = +1$ Activation $a = w^T x = 2$ Prediction correct.

Sample 2: $x = [-1, -1]$, $y = -1$ Activation $a = w^T x = 2$ Prediction update.

weight vector $w = [2, 2]$ prediction incorrect. Update

like this $w = [1, 1]$, $[0, 0]$, Incorrect prediction.

$w = [-1, 1]$, $a = w^T x = 2$ prediction correct.

Sample 3

$x = [0, 0.5]$, $y = -1$ Activation $a = w^T x = -0.5$

Prediction incorrect.

weight vector $w = (0, -1)$, $[-1, -0.5]$, $[-2, 0]$, $[-1, -0.5]$

$[-2, 0]$, $[-1, -0.5]$, $[-2, 0]$, $[-1, -0.5]$ $[-1, -0.5]$

prediction incorrect.

~~$w = [2, 0]$~~. (Ans).