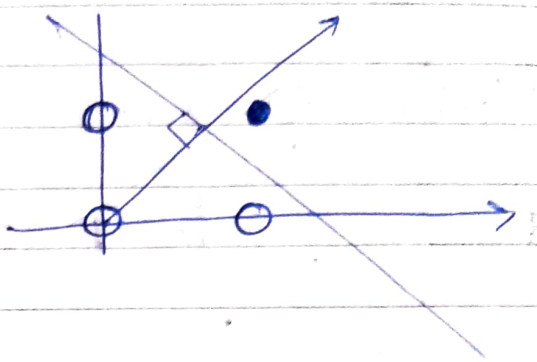
Ex → 0 0 0
0 1 0
1 0 0
1 1 1

① design the decision boundary.



② choose a weight vector that is orthogonal to the decision boundary.

Example:—

			t
p_1 →	1	2	1
p_2 →	−1	2	0
p_3 →	0	−1	0

$$w^* = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix}$$

Start with x_4

$$a = \text{hardlim}(w_1^T x_1) = \text{hardlim}\left(\begin{bmatrix} 1.0 & -0.8 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$$

$$= \text{hardlim}(-0.6) = 0$$

it's output is not correct.

update the weight vector.

$$w^{new} = w^{old} + x$$

$$w_1^{new} = w_1^{old} + x_4 = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 1.2 \end{bmatrix}$$

move on to the next input vector.

DATE: / /

$$a = hardlim(w_1^T x_2) = hardlim\left(\begin{bmatrix}2 & 1.2\end{bmatrix}\begin{bmatrix}-1\\2\end{bmatrix}\right)$$

$$= hardlim(0.4) = 1$$

target associated with x_2 is 0 & the output a is 1. A class 0 vector was misclassified as a 1.

update the weight

if $t = 0$ and $a = 1$ then

$$w_1^{new} = w_1^{old} - x_2$$

$$= \begin{bmatrix}2\\1.2\end{bmatrix} - \begin{bmatrix}-1\\2\end{bmatrix} = \begin{bmatrix}3\\-0.8\end{bmatrix}$$

Now move to next vector

$$a = hardlim\left(\begin{bmatrix}3.0 & -0.8\end{bmatrix}\begin{bmatrix}0\\-1\end{bmatrix}\right) = hardlim(0.8) = 1$$

It is also misclassifies x_3

update weight

$$w_1^{new} = w_1^{old} - p x_3 = \begin{bmatrix}3.0\\-0.8\end{bmatrix} - \begin{bmatrix}0\\-1\end{bmatrix} = \begin{bmatrix}3.0\\0.2\end{bmatrix}$$

continue with iterations you will find that both i/p values will perceptron error $e = target - actual = t - a$ now be correctly classified.

we can update bias

$$b^{new} = b^{old} + e$$ The algo has converged to a solution.

Ex → $w = \begin{bmatrix}0.5\\-1\\0.5\end{bmatrix}$, $b = 0.5$

$x_1 = \begin{bmatrix}+1\\-1\\-1\end{bmatrix}$, $t_1 = [0]$, $p_2 = \begin{bmatrix}1\\1\\-1\end{bmatrix}$, $t_2 = [1]$

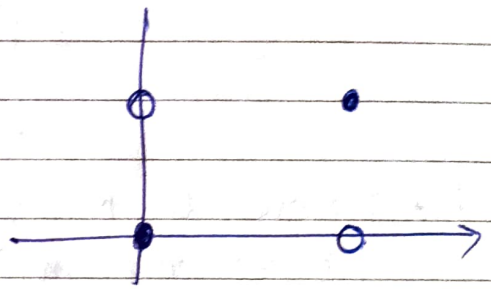
B+

DATE: / /

Limitation :- The perceptron learning rule is guaranteed to converge to a solution in a finite number of steps, so long as a solution exists.

The perceptron can be used to classify input vectors that can be separated by a linear boundary. We call such vectors linearly separable.
Unfortunately, many problems are not linearly separable.
e.g. XOR gate problem.

0	0	0
0	1	1
1	0	1
1	1	0



Ex → $\left\{p_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, t_1 = 0\right\}$

$\left\{p_2 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, t_2 = 1\right\}$

$\left\{p_3 = \begin{bmatrix} -2 \\ 2 \end{bmatrix}, t_3 = 0\right\}$

$\left\{p_4 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, t_4 = 1\right\}$

① Initial weight & bias.

$W(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$b(0) = 0$

② Calculating perceptron output a for first input vector p_1.

③ $W = \begin{bmatrix} -2 \\ -3 \end{bmatrix}, b = 1$

④ Draw Decision Boundary.

B+
Be Positive....