Introduction to Perceptron

Mr Frank Rosenblatt 19^{th} Century

- Building block of ANN

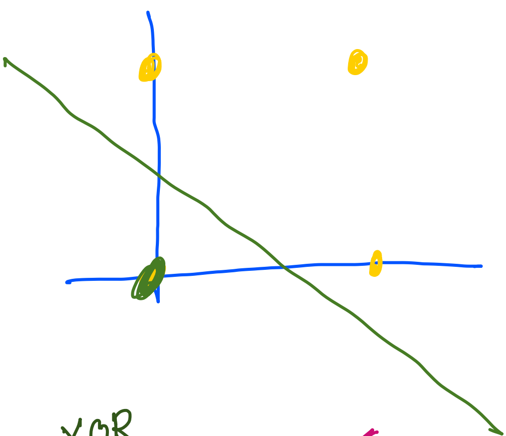
Supervised Learning.

AND OR Gates.

x_1	x_2	Y
0	0	0 ✓
1	0	1
0	1	1
1	1	1

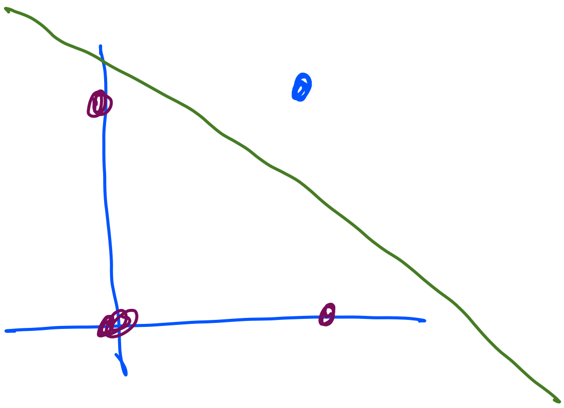
x_1	x_2	Y
0	0	0
0	1	0
1	0	0
1	1	1 ✓

OR



Linearly Separable

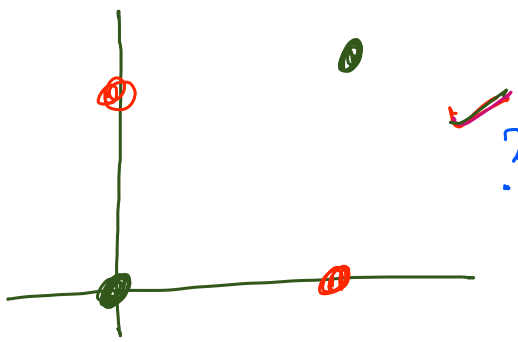
AND



XOR

x_1	x_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

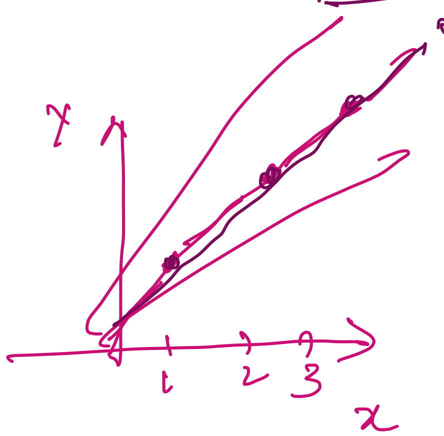
$x_1\bar{x}_2 + \bar{x}_1 x_2$



? No

Can we have a linear decison surface for XOR?

$$\boxed{y = 2x}$$



x	1	2	3	.	10
y	2	4			20

Linear

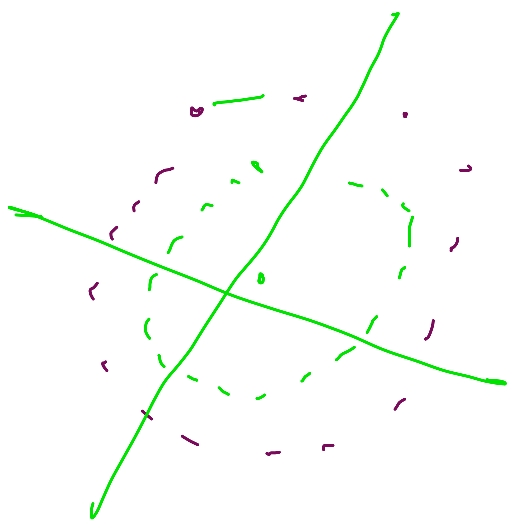
(1, 2)

(2, 4)

(3, 6)

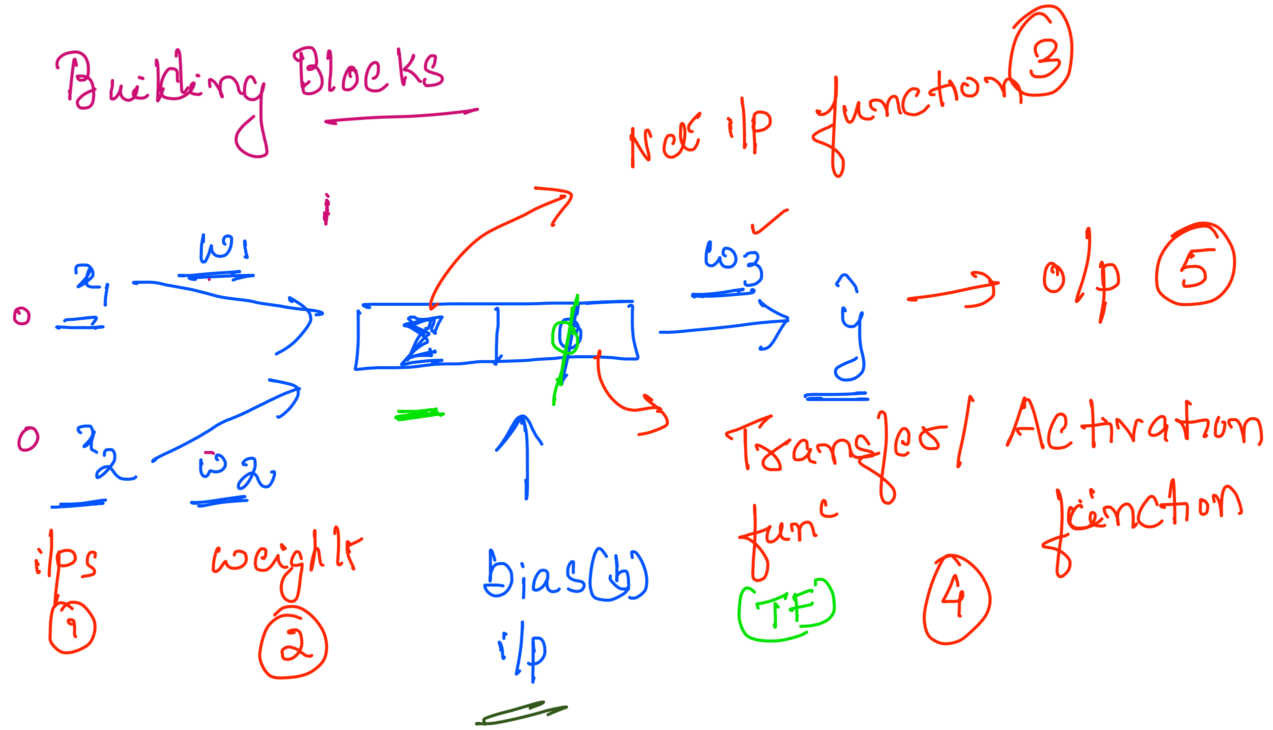
Non-linearity

$$y = 2x + c$$



Single Perceptron

Building Blocks



Σ = $x_1 w_1 + x_2 w_2 + b$

weighted summation

$\phi(\Sigma) \simeq \phi(x_1 w_1 + x_2 w_2 + b)$

↓

y' ← TF

↓

$w_3 y'$

↓

$\hat{y}$

$w_3 \Sigma$

↓

y' o/p

bias = 0

Why TF is required?

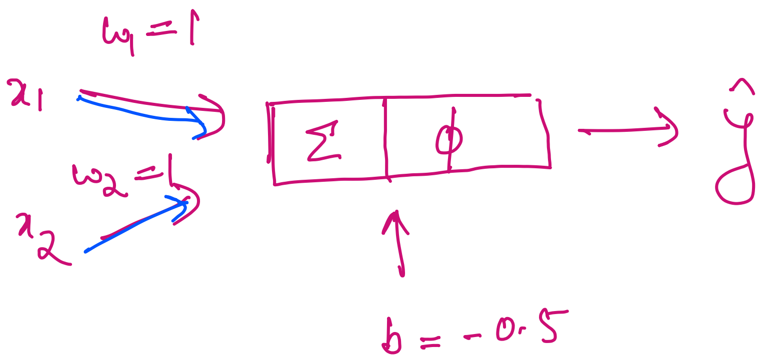
↓ Bring non-linearity

$$P = x_1 w_1 + x_2 w_2 + b$$

Activation Function

$$\phi(P) = \begin{cases} 0 & \Sigma xw + b \leq 0 \\ 1 & \Sigma xw + b > 0 \end{cases}$$

Perceptron for OR Gates



x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

$$f(x, w) = x_1 w_1 + x_2 w_2 + \dots + x_n w_n$$

1st Case

$x_1 = 0$ $\quad$ $x_2 = 0$

$$P = x_1 w_1 + x_2 w_2 + b$$
$$= 0.1 + 0.1 + (-0.5)$$
$$= -0.5$$

$$\phi(P) = \phi(-0.5) = 0 \quad \hat{y}$$

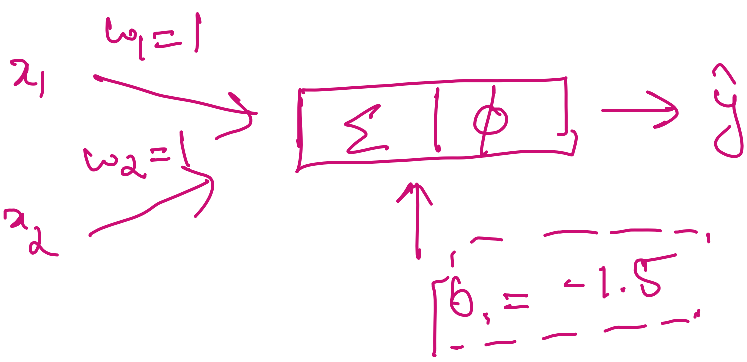
2nd Case

$x_1 = 0$ $\quad$ $x_2 = 1$

$$\phi(0.5) = 1 \ (\hat{y})$$

$$P = x_1 w_1 + x_2 w_2 + b$$
$$= 0.1 + 1.1 + (-0.5)$$
$$= 0.5$$

Perceptron for AND gate



x_1	x_2	y
0	0	0
0	1	0
1	0	0 ✓
1	1	1 ✓

$$\phi = \begin{cases} 0 & w.x + b \le 0 \\ 1 & w.x + b > 0 \end{cases}$$

3rd Case

$x_1 = 1 \qquad x_2 = 0$

$$P = x_1 w_1 + x_2 w_2 + b$$
$$= 1 \times 1 + 0 \times 1 + (-1.5)$$
$$= 1 - 1.5$$
$$= -0.5$$

$$\phi(P) = \phi(-0.5)$$
$$= 0 \quad \underline{\hat{y}}$$

4th Case

$x_1 = 1 \qquad x_2 = 1$

$$P = x_1 w_1 + x_2 w_2 + b$$
$$= 1 \times 1 + 1 \times 1 + (-1.5)$$
$$= 2 - 1.5$$
$$= 0.5$$

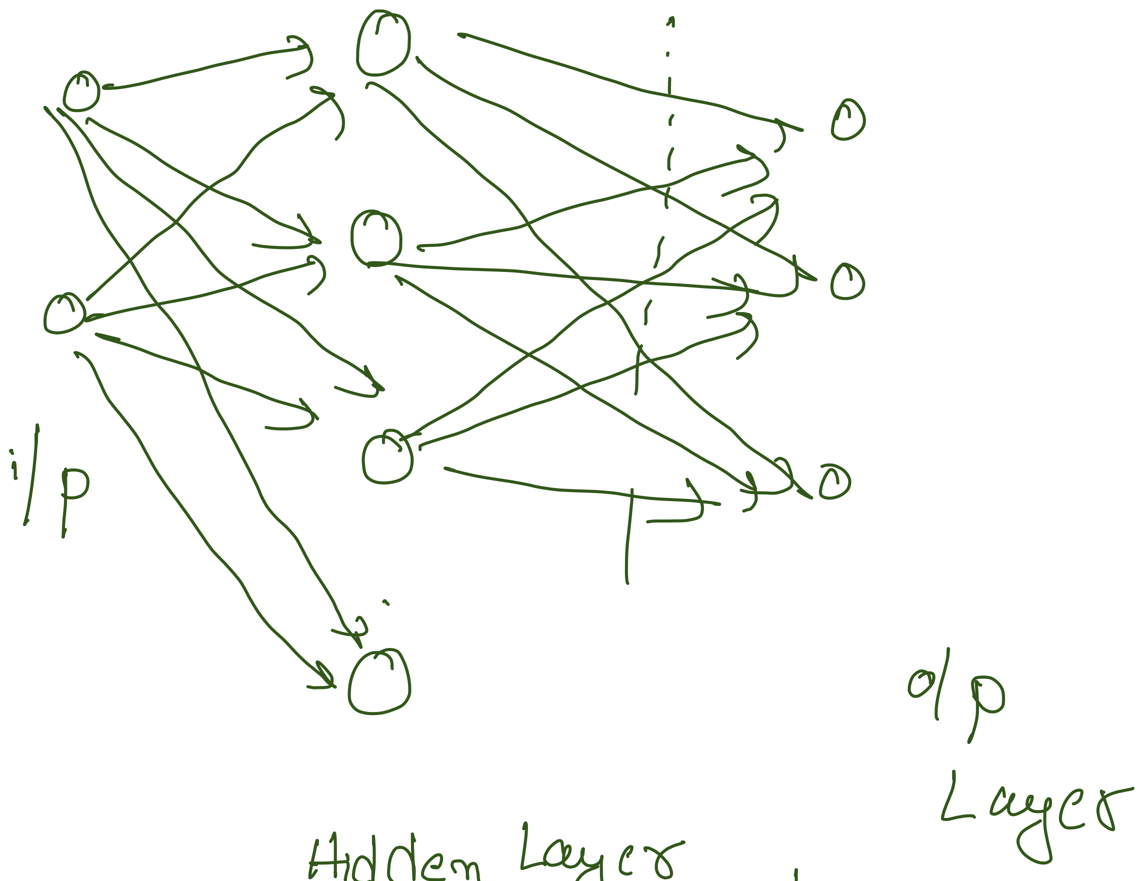
$$\phi(0.5) = 1 \quad \hat{y}$$

XOR

1 Single neuron enough ?

Single Layer Single Neuron ✓

Single Layer Multiple Neuron. ✓

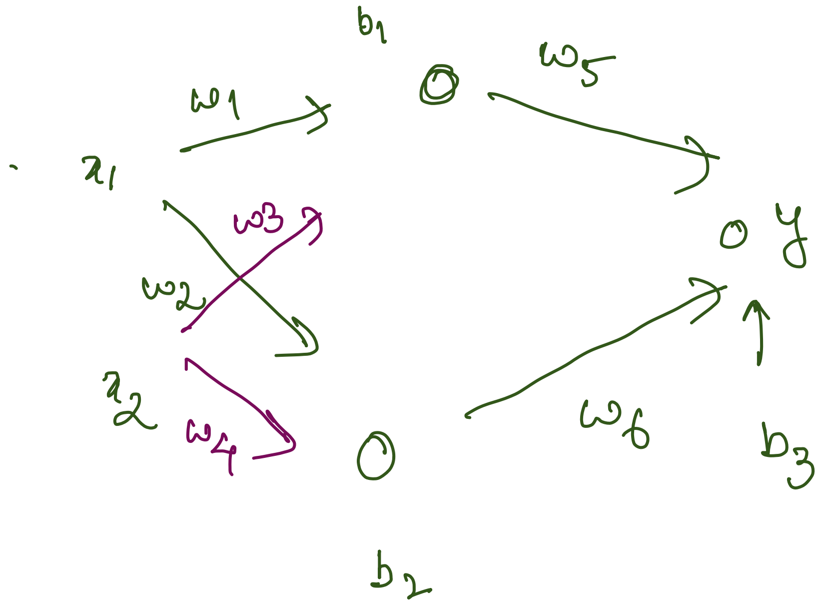


Hidden Layer =1

Fully Connected N/W

Multi Layer

→ Hidden Layer = more than 1



x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Single Layer 2 neurons

1. If we start from any arbitrary weights then is it possible that n/w is able to tune the weights?

Yes Learning

2. weights that are fixed but is this the only representation or there could be other representation which results the same?