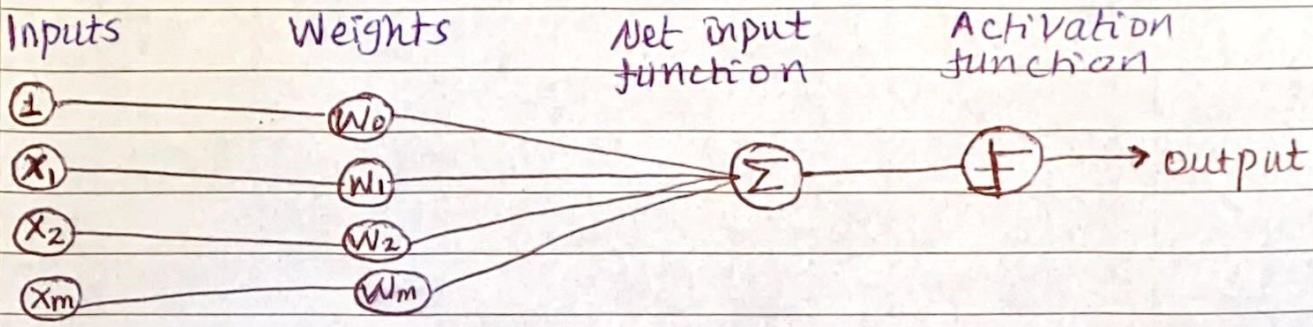
Implementing neural network from scratch

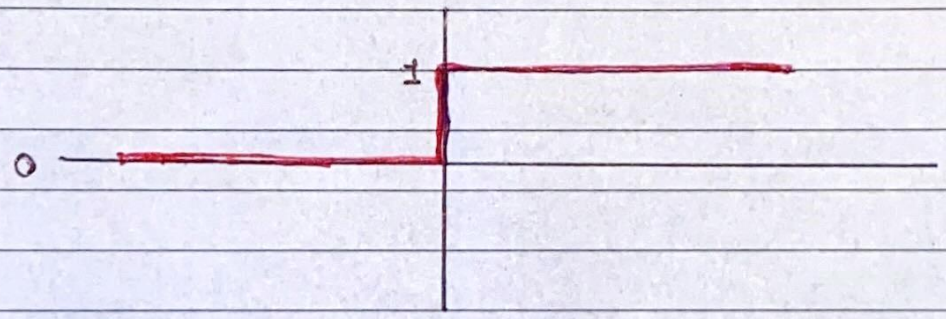


- Linear Model

$$f(w,b) = w^T x + b$$

- Activation Function
  - Unit step function

$$g(z) = \begin{cases} 1 & \text{if } z \geq \theta \\ 0 & \text{otherwise} \end{cases}$$



- Approximation

$$\hat{y} = g(f(w,b)) = g(w^T x + b)$$

- Perceptron update rule.

-For each training sample x_i :

$$w := w + \Delta w$$

$$\Delta w := \alpha \cdot (y_i - \hat{y}_i) \cdot x_i \qquad \alpha : \text{learning rate in } [0,1]$$

- Update rule explanation

y	$\hat{y}$	$y - \hat{y}$
1	1	0
1	0	1
0	0	0
0	1	-1

* Weights are pushed towards positive or negative target class in case of missclassification.