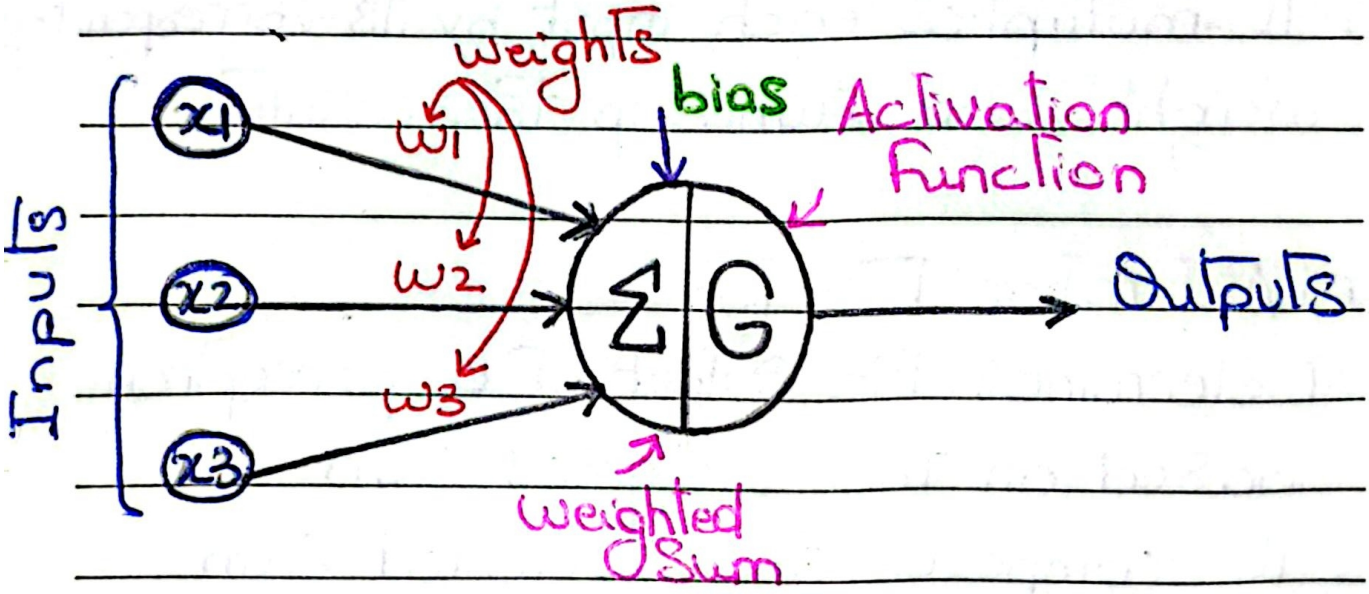
# Perceptron

## → What is a Perceptron

- Simplest type of neural network (only have a single neuron)
- Used to classify between linearly separable classes but not complex classification
- Used for binary classification




### 1) Perceptron Inputs

- Perceptron receives one or more input
- Inputs are called nodes

### 2) Node Weights

- Values assigned to each input
- Shows the importance of each input
- Higher weight means that the input has a stronger influence of the output

## 3) Bias

- Represents focus on a specific part of the environment

## 4) Weighted Sum

$\sum$ weights . input + bias

- The perceptron calculates the weighted sum of its inputs
- It multiplies each input by its corresponding weight and sums up the results

## 5) Activation Function

- Determines the output of a perceptron based on the weighted sum
- It compares the weighted sum to a threshold value and decide how to trigger the output

- In a perceptron we often use "Binary Step" Activation Function

↳ $y = \text{Sign}(\sum w_i x_i + b)$

$$y = \begin{cases} 1 \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases}$$

- There's other types of Activation Functions based on the use case

→ Steps to train a perceptron

Repeat until convergence (or for a fixed number of iterations)

1. Initialize weights and biases (assign a random value usually 0)

2. For each training example $(x_1, x_2, \ldots, x_n)$ with label $y$:
   - Compute the prediction
     $\hat{y} = \text{Sign}(\sum_i x_i w_i + b)$
   - if the prediction is incorrect ($\hat{y} \neq y$ or $\hat{y} \cdot y \leq 0$):
     → update each weight: $w_i = w_i + x_i \cdot y$
     → update the bias: $b = b + y$