INDIVIDUAL TASK-1

TASK: Build a Perception model (on paper or spreadsheet) to solve a binary classification problem and apply the Perception learning law.

1. Introduction

Machine Learning involves building models that can learn patterns from data and make predictions. One of the simplest and earliest models used for **binary classification** is the **Perceptron**.

A perceptron works like a basic artificial neuron. It takes input values, multiplies them by weights, adds a bias, and produces an output of either **0 or 1** (or **−1 or +1**) depending on a threshold.

This report explains:

- The perceptron model
- A binary classification problem
- Perceptron learning law
- Step-by-step weight update process

2. Problem Definition (Binary Classification)

Consider a simple problem:

Goal:
Classify whether a student **passes (1)** or **fails (0)** based on:

- x_1= Hours studied
- x_2= Attendance (in %)

This is a **binary classification problem** because the output has only two classes: Pass or Fail.

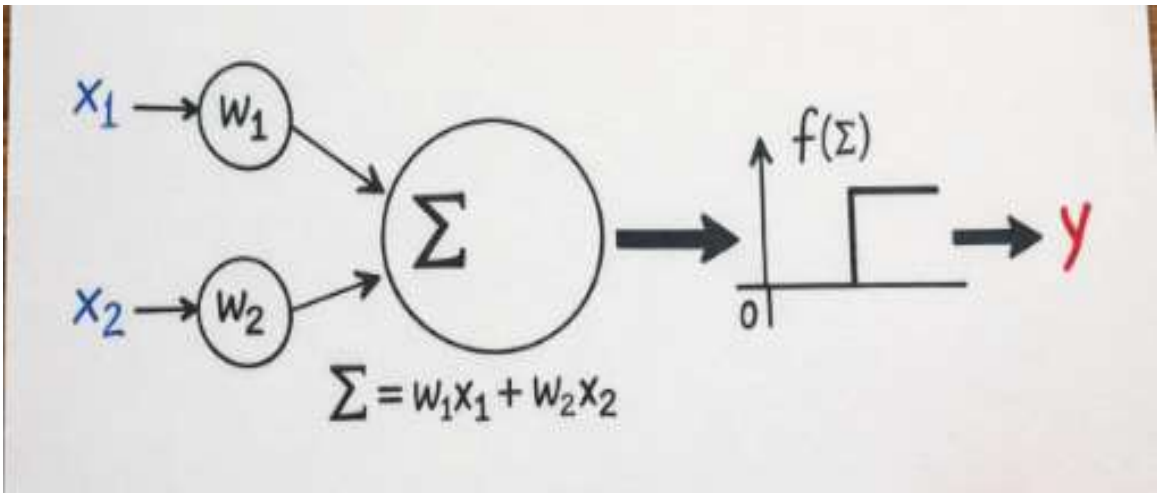
3. Perceptron Model Architecture

The perceptron consists of:

- Input nodes x_1, x_2
- Weights w_1, w_2
- Bias b
- Summation unit

- Activation function (step function)

Diagram: Perceptron Model Architecture



4. Mathematical Model of Perceptron

The net input to the perceptron is:

$$z = w_1x_1 + w_2x_2 + b$$

The output is obtained using the **step activation function**:

$$y = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{if } z < 0 \end{cases}$$

5. Perceptron Learning Law

The perceptron learning law updates weights only when the prediction is wrong.

Weight Update Formula

$$w_i^{new} = w_i^{old} + \eta(t - y)x_i$$
$$b^{new} = b^{old} + \eta(t - y)$$

Where:

- η= learning rate (e.g., 0.1)
- t= target output
- y= predicted output

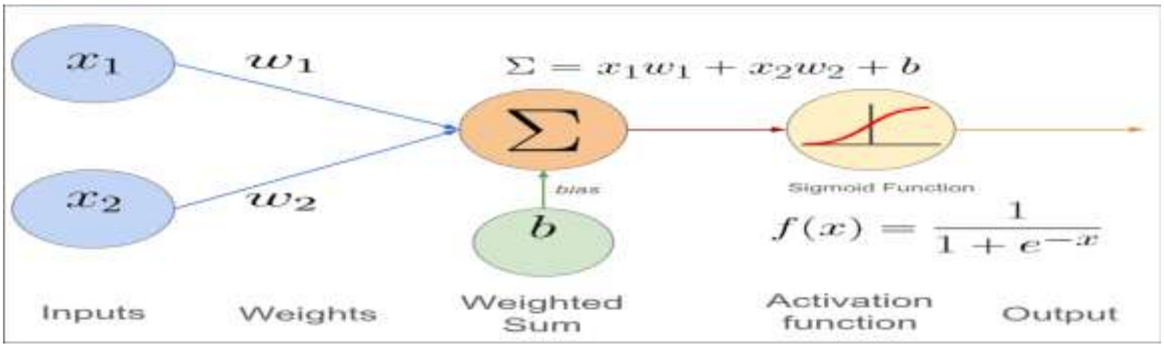
6. Training Dataset

Sample	x_1(Hours)	x_2(Attendance)	Target t
1	2	75	0
2	4	80	1
3	1	60	0
4	5	90	1

7. Initial Values

- $w_1 = 0$
- $w_2 = 0$
- $b = 0$
- Learning rate $\eta = 0.1$

8. Step-by-Step Perceptron Training



Sample 1

$$z = (0)(2) + (0)(75) + 0 = 0$$
$$y = 1$$

Target $t = 0$→ **Error**

Update weights:

$$w_1 = 0 + 0.1(0 - 1)(2) = -0.2$$
$$w_2 = 0 + 0.1(0 - 1)(75) = -7.5$$

$$b = 0 + 0.1(0 - 1) = -0.1$$

Sample 2

$$z = (-0.2)(4) + (-7.5)(80) - 0.1 < 0$$
$$y = 0$$

Target $t = 1$→ **Error**

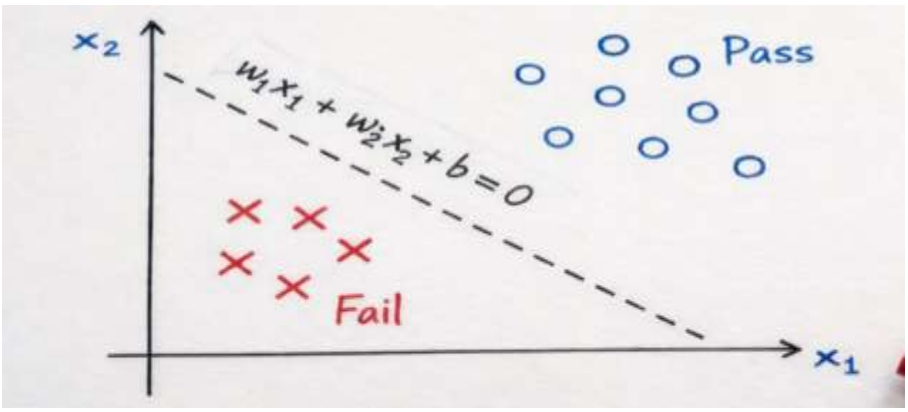
9. Decision Boundary

The perceptron learns a **linear decision boundary**:

$$w_1x_1 + w_2x_2 + b = 0$$

This line separates **Pass** and **Fail** classes.

Diagram: Decision Boundary for Binary Classification



10. Advantages of Perceptron

- **Simple and Easy to Understand:**
  The perceptron is one of the simplest machine learning models. It uses basic mathematical operations such as multiplication, addition, and a threshold (step) function. Because of this simplicity, it is often the **first algorithm taught in neural networks**, making it ideal for beginners and BTech students.

- **Fast Training Process:**
  The perceptron updates its weights using a simple learning rule:

$$w = w + \eta(y - \hat{y})x$$

- Since there is no complex optimization or backpropagation involved, training is computationally efficient and fast, especially for small datasets.

- **Suitable for Linearly Separable Problems**
  When data can be separated by a straight line (or plane in higher dimensions), the perceptron is guaranteed to converge. Examples include simple **Pass/Fail**, **Yes/No**, or **Spam/Not Spam (basic cases)** classifications.

11. Limitations

- **Cannot Solve Non-Linearly Separable Problems**
  The major limitation of the perceptron is that it fails when data is not linearly separable. A classic example is the **XOR problem**, where no straight line can separate the two classes. This limitation led to the development of **multi-layer neural networks**.
- **Sensitive to Noisy Data**
  If the training data contains noise or mislabelled samples, the perceptron may keep updating weights without reaching a stable solution. This makes it less reliable for real-world datasets where noise is common.
- **Uses Only Linear Decision Boundaries**
  The perceptron can only learn linear relationships. Complex patterns, curves, or nonlinear boundaries cannot be modelled using a single-layer perceptron.

12. Conclusion

- In this report, a **perceptron model** was designed to solve a binary classification problem such as Pass/Fail prediction. The model uses weighted inputs, a summation unit, and a step activation function to produce binary outputs.
- The **perceptron learning law** updates weights iteratively based on the prediction error, allowing the model to gradually improve its performance. Through this process, the perceptron learns a **linear decision boundary** that separates two classes.
- Although the perceptron is limited to linearly separable data and simple problems, it plays a **foundational role in machine learning and neural networks**. The ideas introduced by the perceptron form the basis for advanced models such as **multi-layer perceptron's (MLP)** and **deep neural networks**, which are widely used in modern applications like image recognition, speech processing, and medical diagnosis.

13.References Frank Rosenblatt

Rosenblatt, F. (1958). *The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain*. Psychological Review.

- Simon Haykin, **Neural Networks and Learning Machines**, Pearson Education.
- Tom M. Mitchell, **Machine Learning**, McGraw-Hill.
- Christopher M. Bishop, **Pattern Recognition and Machine Learning**, Springer.
- NPTEL Online Courses – *Introduction to Machine Learning* (IITs).