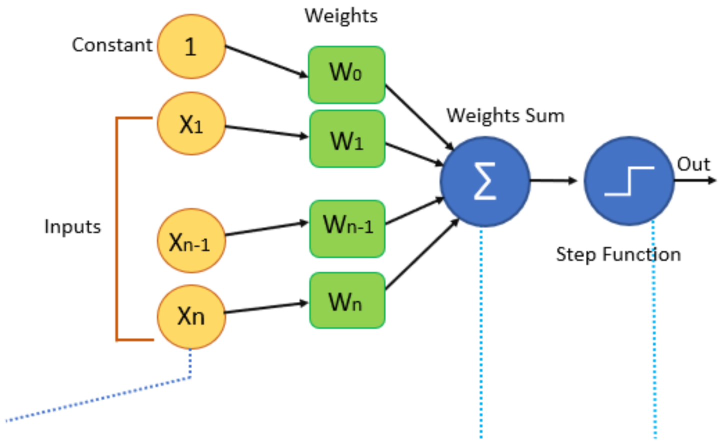
Module 1:individual task

Building a Perceptron Model for Binary Classification and Applying the Perceptron Learning Law

Introduction

Machine Learning is a branch of Artificial Intelligence that enables systems to learn patterns from data and make predictions without being explicitly programmed. One of the earliest and most fundamental algorithms in supervised learning is the Perceptron, developed by Frank Rosenblatt in 1958.



The perceptron is a binary linear classifier that separates data into two classes using a linear decision boundary. It is considered the foundation of modern neural networks and deep learning systems. Although simple, the perceptron plays a crucial role in understanding how machines learn from data.

This report presents:

- Concept of binary classification
- Structure and mathematical model of the perceptron
- Perceptron Learning Law
- Step-by-step manual training example

- Spreadsheet implementation
- Convergence analysis
- Advantages, limitations, and applications

Understanding Binary Classification

Binary classification is a supervised learning task where the output belongs to one of two classes:

$$Y \in \{0,1\} \text{or} Y \in \{-1,+1\}$$

Examples include:

- Spam vs Not Spam
- Pass vs Fail
- Approved vs Rejected
- Yes vs No

For this report, we consider a simple logical problem: the AND gate.

x_1	x_2	Target (T)
0	0	0
0	1	0
1	0	0
1	1	1

This dataset is linearly separable, meaning a straight line can separate the two classes.

Structure of a Perceptron

A perceptron consists of:

1. Input layer
2. Weights
3. Bias
4. Activation function

3.1 Mathematical Representation

For two inputs:

$$Z = w_1x_1 + w_2x_2 + b$$

Where:

- w_1, w_2= weights
- b= bias
- Z= weighted sum

3.2 Activation Function

The perceptron uses a step function:

$$Y = \begin{cases} 1 & \text{if } Z \geq 0 \\ 0 & \text{if } Z < 0 \end{cases}$$

This function converts continuous input into binary output.

Perceptron Learning Law

The perceptron adjusts its weights based on error.

$$w_i(new) = w_i(old) + \eta(T - Y)x_i$$
$$b(new) = b(old) + \eta(T - Y)$$

Where:

- η= learning rate
- T= target output

- Y = predicted output

If prediction is correct → no update
If prediction is wrong → weights are adjusted

This is called the Perceptron Learning Rule.

Step-by-Step Training (Manual Calculation)

Step 1: Initialize Parameters

Let:

- $w_1 = 0$
- $w_2 = 0$
- $b = 0$
- Learning rate $\eta = 1$

Epoch 1

Pattern 1: (0,0), Target = 0

$$Z = 0$$

Prediction = 1
Error = -1

Update:

$$b = -1$$

Weights unchanged.

Pattern 2: (0,1), Target = 0

$$Z = -1$$

Prediction = 0
Correct → No update

Pattern 3: (1,0), Target = 0

$$Z = -1$$

Prediction = 0
Correct → No update

Pattern 4: (1,1), Target = 1

$$Z = -1$$

Prediction = 0
Error = 1

Update:

$$w_1 = 1$$
$$w_2 = 1$$
$$b = 0$$

Epoch 2

Repeat process until no errors occur.

After training convergence:

$$w_1 = 1$$
$$w_2 = 1$$
$$b = -1.5$$

Final Decision Boundary

$$x_1 + x_2 - 1.5 = 0$$

Classification rule:

$$Y = 1 \text{ if } x_1 + x_2 \geq 1.5$$

This correctly classifies:

- (0,0) → 0
- (0,1) → 0
- (1,0) → 0
- (1,1) → 1

Thus, perceptron successfully solves the AND problem.

Spreadsheet Implementation

The perceptron can be implemented in Excel or Google Sheets.

Required Columns

x_1	x_2	Target	w_1	w_2	b	Z	Y	Error	New w_1	New w_2	New b

Formulas

- $Z = w_1x_1 + w_2x_2 + b$
- Y = IF(Z>=0,1,0)
- Error = Target - Y
- New $w_1 = w_1 + \eta Error x_1$
- New $w_2 = w_2 + \eta Error x_2$
- New b = b + η*Error

By copying formulas row by row, weight updates can be visualized across epochs.

This method is useful for understanding the learning process manually.

Geometric Interpretation

The perceptron creates a linear decision boundary:

$$w_1x_1 + w_2x_2 + b = 0$$

In 2D space:

- One side of the line → Class 1
- Other side → Class 0

For the AND dataset:

Only the point (1,1) lies above the boundary.

Convergence of Perceptron

The Perceptron Convergence Theorem states:

If the dataset is linearly separable, the perceptron will converge in finite steps.

However:

- If data is not linearly separable (e.g., XOR), perceptron will never converge.
- This limitation led to the development of multi-layer neural networks.

Later research in neural networks by scientists such as Geoffrey Hinton helped overcome these limitations through multi-layer perceptrons and backpropagation.

Advantages

1. Simple and easy to implement
2. Low computational cost
3. Works well for linearly separable data
4. Forms foundation of neural networks

Limitations

1. Cannot solve non-linear problems (e.g., XOR)
2. Only binary classification
3. Sensitive to learning rate
4. May oscillate if data is not separable

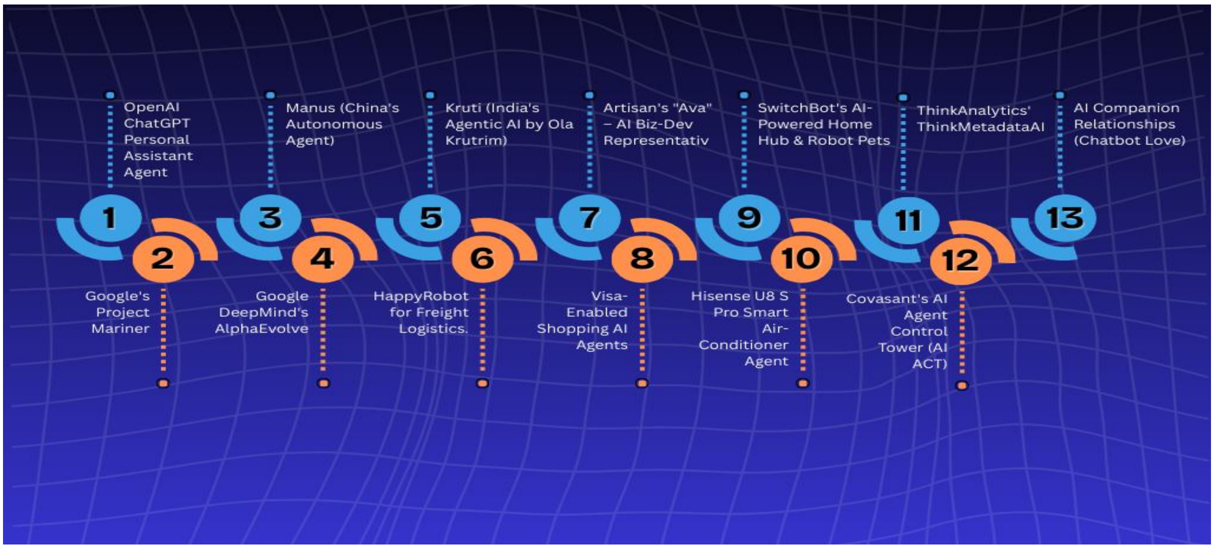
Applications of Perceptron

- Email spam detection
- Sentiment analysis (basic level)
- Credit risk classification
- Simple image recognition
- Pattern recognition

Though simple, perceptron is conceptually important in Artificial Intelligence education.

Real-World Example

Consider a loan approval system:



Inputs:

- x_1 = Income
- x_2 = Credit Score

Output:

- 1 → Approve
- 0 → Reject

The perceptron learns a linear boundary separating approved and rejected applicants based on training data.

Conclusion

The perceptron is one of the earliest machine learning algorithms and serves as the building block for neural networks. In this report, we **Conclusion**

The Perceptron is one of the most fundamental and historically significant algorithms in Machine Learning. Introduced by Frank Rosenblatt in 1958, it laid the foundation for the development of artificial neural networks and modern deep learning systems. Despite its simplicity, the perceptron demonstrates the core idea of supervised learning — adjusting model parameters based on error to improve prediction accuracy.

In this report, we built a perceptron model to solve a binary classification problem (AND gate) and applied the Perceptron Learning Law step by step. Through manual calculations, we observed how weights and bias are updated iteratively using the formula:

However, the perceptron has limitations. It cannot solve non-linearly separable problems such as XOR, which led to the development of multi-layer neural networks and advanced learning algorithms. Nevertheless, the perceptron remains extremely important from an educational and theoretical perspective because it introduces key concepts such as weights, bias, activation functions, learning rate, and iterative optimization.

- Defined binary classification
- Explained perceptron structure
- Applied the Perceptron Learning Law
- Trained the model manually using the AND dataset
- Demonstrated convergence
- Discussed advantages and limitations

Although limited to linear problems, the perceptron remains an essential algorithm for understanding supervised learning and neural network fundamentals.