**ML EXPERIMENT 10**

import numpy as np

class Perceptron:

    def \_\_init\_\_(self, input\_size, learning\_rate=0.1, epochs=10):

        self.weights = np.zeros(input\_size + 1)

        self.learning\_rate = learning\_rate

        self.epochs = epochs

    def activation(self, x):

        # Step function as activation function

        return 1 if x >= 0 else 0

    def predict(self, x):

        z = np.dot(x, self.weights[1:]) + self.weights[0]

        return self.activation(z)

    def fit(self, X, y):

        for epoch in range(self.epochs):

            print(f"Epoch {epoch + 1} weights: {self.weights}")

            for inputs, label in zip(X, y):

                prediction = self.predict(inputs)

                error = label - prediction

                self.weights[1:] += self.learning\_rate \* error \* inputs

                self.weights[0] += self.learning\_rate \* error

            print(f"Updated weights: {self.weights}")

# Example for AND operation

X = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])

y = np.array([0, 0, 0, 1])

# Initialize and train the perceptron

perceptron = Perceptron(input\_size=2)

perceptron.fit(X, y)

