Experiment 3: Implement Perceptron for Logical Functions

Aim:

To implement a Perceptron model in Python and use it to simulate basic logical functions such as AND, OR, and NOT.

Apparatus Required:

- Python (Anaconda/Miniconda/Standard Python Installation)
- Jupyter Notebook or any Python IDE
- NumPy Library

Theory:

A perceptron is the simplest type of artificial neural network model and a basic building block for more complex neural networks. It is used for binary classification tasks. The perceptron takes multiple binary inputs, applies weights to them, sums them, and passes the result through an activation function (typically a step function).

The perceptron learning algorithm adjusts the weights based on prediction error using a simple rule. It can be used to implement basic logical gates such as AND, OR, and NOT.

Code:

```
import numpy as np
class Perceptron:
    def __init__(self, input_size):
        self.weights = np.zeros(input_size + 1)  # +1 for bias
    def predict(self, x):
        x = np.insert(x, 0, 1)  # Add bias input
        return 1 if np.dot(self.weights, x) > 0 else 0
    def train(self, X, y, lr=0.1, epochs=10):
        for _ in range(epochs):
            for i in range(len(X)):
                x_i = np.insert(X[i], 0, 1)  # Add bias input
                y_pred = self.predict(X[i])
                self.weights += lr * (y[i] - y_pred) * x_i
# Logical AND function
X = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
y = np.array([0, 0, 0, 1])
p = Perceptron(2)
```

```
p.train(X, y)
print("AND Gate Predictions:", [p.predict(x) for x in X])
# Logical OR function
y_or = np.array([0, 1, 1, 1])
p_or = Perceptron(2)
p_or.train(X, y_or)
print("OR Gate Predictions:", [p_or.predict(x) for x in X])
# Logical NOT function
X_not = np.array([[0], [1]])
y_not = np.array([1, 0])
p_not = Perceptron(1)
p_not.train(X_not, y_not)
print("NOT Gate Predictions:", [p_not.predict(x) for x in X_not])

# Tests
assert and_preds == [0, 0, 0, 1], "AND gate test failed"
assert or_preds == [0, 1, 1, 1], "OR gate test failed"
assert not_preds == [1, 0], "NOT gate test failed"
print("✅ All logic gate tests passed!")
```

Result:

The perceptron model correctly simulated the logical AND, OR, and NOT gates.