Name :- Ajinkya Kumbhar

Roll No :- 36

Subject: Artificial Neural Network (SL - II)

Class : TE

Branch: AI & DS

Practical – 5

Problem statement : Implement an Artificial Neural Network training process in Python by using Forward Propagation, and Back Propagation.

Code :

```
import numpy as np
def sigmoid(x):
    return 1 / (1 + np.exp(-x))
def sigmoid_derivative(x):
    return x * (1 - x)
class NeuralNetwork:
    def __init__(self, x, y):
        self.input = x
        self.output = y
        self.hidden_size = 4
        self.weights1 = np.random.randn(self.input.shape[1], self.hidden_size)
        self.weights2 = np.random.randn(self.hidden_size, 1)
    def feedforward(self):
        self.hidden = sigmoid(np.dot(self.input, self.weights1))
        self.predicted_output = sigmoid(np.dot(self.hidden, self.weights2))
    def backpropagate(self):
        output_error = self.output - self.predicted_output
        d_predicted_output = output_error * sigmoid_derivative(self.predicted_output)
        hidden_error = d_predicted_output.dot(self.weights2.T)
        d_hidden = hidden_error * sigmoid_derivative(self.hidden)
```

```
        self.weights1 += self.input.T.dot(d_hidden)
        self.weights2 += self.hidden.T.dot(d_predicted_output)
    def train(self, epochs):
        for epoch in range(epochs):
            self.feedforward()
            self.backpropagate()
    def predict(self, x):
        hidden = sigmoid(np.dot(x, self.weights1))
        predicted_output = sigmoid(np.dot(hidden, self.weights2))
        return predicted_output
X = np.array([[0, 0, 1], [0, 1, 1], [1, 0, 1], [1, 1, 1]])
y = np.array([[0], [1], [1], [0]])
nn = NeuralNetwork(X, y)
nn.train(10000)
x_test = np.array([[0, 0, 0], [1, 0, 0]])
for x in x_test:
    print("Input:", x)
    print("Output:", nn.predict(x))
```

OUTPUT :