**Assignment No-6**

**Name-**Hule Samiksha Anil

**Roll No**-24

**Title-** Artificial Neural Network trining process of Forward Propagation, Back Propagation .

**Program:**

import numpy as np

class NeuralNetwork:

def \_init\_(self, input\_size, hidden\_size, output\_size):

self.W1 = np.random.randn(input\_size, hidden\_size)

self.b1 = np.zeros((1, hidden\_size))

self.W2 = np.random.randn(hidden\_size, output\_size)

self.b2 = np.zeros((1, output\_size))

def sigmoid(self, x):

return 1 / (1 + np.exp(-x))

def sigmoid\_derivative(self, x):

return x \* (1 - x)

def forward\_propagation(self, X):

self.z1 = np.dot(X, self.W1) + self.b1

self.a1 = self.sigmoid(self.z1)

self.z2 = np.dot(self.a1, self.W2) + self.b2

y\_hat = self.sigmoid(self.z2)

return y\_hat

def backward\_propagation(self, X, y, y\_hat):

self.error = y - y\_hat

self.delta2 = self.error \* self.sigmoid\_derivative(y\_hat)

self.a1\_error = self.delta2.dot(self.W2.T)

self.delta1 = self.a1\_error \* self.sigmoid\_derivative(self.a1 self.W2 += self.a1.T.dot(self.delta2)

self.b2 += np.sum(self.delta2, axis=0, keepdims=True)

self.W1 += X.T.dot(self.delta1)

self.b1 += np.sum(self.delta1, axis=0)

def train(self, X, y, epochs, learning\_rate=0.1)

for i in range(epochs):

y\_hat = self.forward\_propagation(X)

self.backward\_propagation(X, y, y\_hat)

if i % 1000 == 0:

print("Error at epoch", i, ":", np.mean(np.abs(self.error)))

def predict(self, X):

return self.forward\_propagation(X)

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

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

nn = NeuralNetwork(input\_size=2, hidden\_size=4, output\_size=1)

nn.train(X, y, epochs=10000)

predictions = nn.predict(X)

print("\nPredictions:\n", predictions)

**Output:**

[[5.55111512e-16]

[6.66666667e-01]

[6.66666667e-01]

[6.66666667e-01]]