**AI FOR LLM- CSA1704**

**16. Write the python program to implement Feed forward neural Network**

**CODE:**

import numpy as np

# Sigmoid activation function

def sigmoid(x):

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

# Feed Forward Neural Network

class FeedForwardNN:

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

# Initialize weights and biases randomly

self.W1 = np.random.randn(input\_size, hidden\_size) # Input → Hidden

self.b1 = np.random.randn(hidden\_size) # Hidden bias

self.W2 = np.random.randn(hidden\_size, output\_size) # Hidden → Output

self.b2 = np.random.randn(output\_size) # Output bias

def forward(self, X):

# Hidden layer

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

self.a1 = sigmoid(self.z1)

# Output layer

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

self.a2 = sigmoid(self.z2)

return self.a2

# Example usage

if \_\_name\_\_ == "\_\_main\_\_":

np.random.seed(42) # reproducibility

# Create network: 3 input neurons, 4 hidden neurons, 2 output neurons

nn = FeedForwardNN(input\_size=3, hidden\_size=4, output\_size=2)

# Example input (3 features)

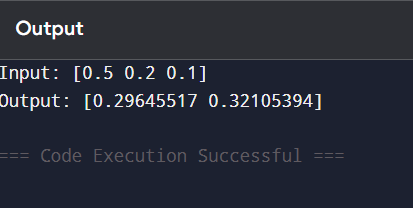
X = np.array([0.5, 0.2, 0.1])

output = nn.forward(X)

print("Input:", X)

print("Output:", output)

**OUTPUT:**

****