import numpy as np

class FeedForwardNN:

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

# Initialize weights

self.weights\_input\_hidden = np.random.rand(input\_size, hidden\_size)

self.weights\_hidden\_output = np.random.rand(hidden\_size, output\_size)

self.bias\_hidden = np.random.rand(hidden\_size)

self.bias\_output = np.random.rand(output\_size)

def sigmoid(self, x):

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

def forward(self, x):

# Input to hidden layer

self.hidden\_input = np.dot(x, self.weights\_input\_hidden) + self.bias\_hidden

self.hidden\_output = self.sigmoid(self.hidden\_input)

# Hidden to output layer

self.output\_input = np.dot(self.hidden\_output, self.weights\_hidden\_output) + self.bias\_output

self.output = self.sigmoid(self.output\_input)

return self.output

# Example usage

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

# Create a Feed Forward Neural Network

nn = FeedForwardNN(input\_size=3, hidden\_size=5, output\_size=2)

input\_data = np.array([0.5, 0.2, 0.1])

output\_data = nn.forward(input\_data)

print("Output of the Feed Forward Neural Network:", output\_data)