import numpy as np

def sigmoid(x):

  return 1 / (1 + np.exp(- x))

# Arquitetura da MPL 4x3x2

N\_input = 3

N\_hidden = 4

N\_output = 2

#Vetor com valores de entrada aleatórios

X = np.array([1, 2, 3])

#Pesos da Camada Oculta

weigths\_in\_hidden = np.array([[-0.08, 0.08, -0.03, 0.03],

                              [ 0.05, 0.10,  0.07, 0.02],

                              [-0.07, 0.04, -0.01, 0.01]])

#Pesos da Camada de Saída

weigths\_in\_out = np.array([[-0.18, 0.11],

                           [-0.09, 0.05],

                           [-0.04, 0.05],

                           [-0.02, 0.07]])

#Passagem ForWard pela rede

#Camada Oculta

#Calcule a combinação linear de entradas e pesos sinápticos

hidden\_layer\_in = np.dot(X, weigths\_in\_hidden)

#Aplicando a função de ativação

hidden\_layer\_out = sigmoid(hidden\_layer\_in)

#Camada de Saída

#Calcule a combinação linear de entradas e pesos sinápticos

output\_layer\_in = np.dot(hidden\_layer\_out, weigths\_in\_out)

#Aplicando a função de Ativação

output\_layer\_out = sigmoid(output\_layer\_in)

print("As saídas da rede são: ", output\_layer\_out)