

```

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
weights_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
weights_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, weights_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, weights_in_out)

#Aplicando a função de Ativação
output_layer_out = sigmoid(output_layer_in)

print("As saídas da rede são: ", output_layer_out)

```