

```
#tarefa 4
#Jóvio L. Giacomolli

import numpy as np

#função sigmoide
def sigmoid(x):
    return 1/(1 + np.exp(-x))

#arquitetura da MPL
n_input = 3
n_hidden = 4
n_output = 2

#vetor dos valores de entrada(aleatoria)
x = np.array([1, 2, 3])

#pesos camada oculta
weights_in_hidden = np.array([[0.2, 0.1, -0.9, 0.03],
                               [0.6, -0.8, 0.9, 0.02],
                               [0.5, -0.6, 0.1, 0.01]])

#pesos camada de saida
weights_hidden_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
#entrada camada oculta
hidden_layer_in = np.dot(x, weights_in_hidden)
#saída camada oculta
hidden_layer_out = sigmoid(hidden_layer_in)

#camada de saida

output_layer_in = np.dot(hidden_layer_out, weights_hidden_out)
#aplicar a função de ativação
output_layer_out = sigmoid(output_layer_in)

print('As saídas da rede são {}'.format(output_layer_out))
```