```
import numpy as np


def sigmoid(x):
    result = (1+np.exp(-x)) # 1 + (e^ -x)
    return 1/result         # 1 / (1 + (e^ -x))


X = np.array([3, 4, 2])

weights_in_hidden = np.array([[-0.07,  0.04, -0.05, 0.07],
                              [ 0.04,  0.10,  0.02, 0.01],
                              [-0.03,  0.04, -0.11, 0.06]])

out = np.array([[-0.18,  0.11],
                       [-0.09,  0.05],
                       [-0.04,  0.05],
                       [-0.02,  0.07]])


weights_hidden_out = np.array([[-0.10,  0.09],
                               [-0.04,  0.12],
                               [-0.02,  0.04],
                               [-0.01,  0.09]])



#Calcule a combinação linear de entradas e pesos sinápticos
hidden_layer_in = np.dot(X, weights_in_hidden)

#Aplicado a função de ativação
hidden_layer_out = sigmoid(hidden_layer_in)

#Calcule a combinação linear de entradas e pesos sinápticos
output_layer_in = np.dot(hidden_layer_out, weights_hidden_out)

#Aplicado a função de ativação
output_layer_out = sigmoid(output_layer_in)

print('O input da camada oculta é:',hidden_layer_in)

print('O output da camada oculta é:',hidden_layer_out)


print('O input da camada de output é:',output_layer_in)

print('As saídas da rede são',output_layer_out)
```