

```

# -*- coding: utf-8 -*-
"""
Created on Fri Jun 26 17:07:54 2020

@author: Barbara
"""

##Red neuronal

import numpy as np
import math
entrada=[4,6,1,5]

#inicializacion de variables
oculta=np.zeros(3)
salida=np.zeros(2)
y_oculta=np.zeros(3)

#pesos de la capa oculta
w_e_o=np.array([[0.1,0.3,0.2,0.1],
                [0.2,0.1,0.4,0.5],
                [0.5,0.3,0.5,0.2]])

#pesos de la capa de salida
w_o_s=np.array([[0.2,0.1,0.1],
                [0.4,0.3,0.1]])

#Función sigmoide
def sigmoid(x):
    return 1 / (1 + math.exp(-x))

for j in range(len(oculta)): #se itera sobre las neuronas de la capa oculta
    for i in range(len(entrada)): #se itera sobre las neuronas de la capa de
        #entrada
        oculta[j]+=entrada[i]*w_e_o[j,i] #se suma el cociente de la entrada
        #por el peso al de la iteracion anterior para obtener el sumatorio
    y_oculta[j]=math.tanh(oculta[j]) #salida de la capa oculta, con la funcion
    #de activación hiperbolica
    for k in range(len(salida)): #se itera sobre la capa de salida
        salida[k]+=y_oculta[j]*w_o_s[k,j] #se suma el cociente de la salida de
        #la capa oculta por el peso a la iteracion anterior para obtener el
        #sumatorio

sg=np.vectorize(sigmoid) #se vectoriza la función
y=sg(salida) #se calcula la función de activacion de la capa de salida

print(y)

```