

```

# -*- coding: utf-8 -*-
"""
Created on Thu Apr  2 17:39:27 2020
Cap 12
@author: campioni
"""

import matplotlib.pyplot as plt
import math
import random as ran

ran.seed(1)
print()
print('=====  

CALCOLO DI UNA SEMPLICE RETE NEURALE A DUE INGRESSI ED UNA USCITA  

=====')
print()
print('-Dati in ingresso')
print()
eta=float(input('eta (0<eta<1,  valori consigliato 0,1 < eta < 0,2) = '))
print()

iterazioni=int(input('Numero di iterazioni = '))

def sig(t): # calcola la sigmoide
    return 1/(1+math.exp(-t))

def D_sig(t): # calcola la derivata della sigmoide rispetto a t
    return sig(t)*(1-sig(t))

def calc_RP(): #calcola la retro-propagazione (w1 , w2 , b)
    w1=ran.random()
    w2=ran.random()
    b=ran.random()
    costo=[]
    for i in range(iterazioni):
        i1=ran.randint(0,len(dati)-1) #genera un numero casuale intero tra 0 e 9
        serie=dati[i1]
        z=serie[0]*w1+serie[1]*w2+b #definisce la variabile d'appoggio t
        p=sig(z)
        obie=serie[2] #seleziona il terzo dato di ogni serie che rappresenta
l'obiettivo
        costo1=(p-obie)**2 # definisce il costo
        if i%100==0: #lista delcosto ogni 100 valori di i
            costo.append(costo1)

        Dcosto_p=2*(p-obie) # calcola la derivata del costo rispetto a p
        Dp_z=D_sig(z) # calcola la derivata di p rispetto a t
        Dcosto_z=Dcosto_p*Dp_z # calcola la derivata comune alle tre variabili
        Dcosto_w1=Dcosto_z*serie[0] # Calcolo la derivata del costo rispetto ai
pesi e b
        Dcosto_w2=Dcosto_z*serie[1]
        Dcosto_b=Dcosto_z
        w1 -=Dcosto_w1*eta
        w2 -=Dcosto_w2*eta
        b -= Dcosto_b*eta
    return w1,w2,b,costo

dati=[[1.52,1.53,0],[1.23,1.68,1],[1.54,1.56,0],[1.28,1.78,1],[1.50,1.57,0],
[1.21,1.81,1],[1.59,1.53,0],[1.18,1.68,1],[1.60,1.54,0],[1.25,1.78,1]]

w1,w2,b,costo=calc_RP()
predi=[]

```

```

for uomo in dati:          # calcola la propagazione con w1,w2,b aggiornati
    z=uomo[0]*w1+uomo[1]*w2+b
    p=sig(z)
    if p <= 0.5:
        predi.append('di Neanderthal')
    else:
        predi.append('Sapiens')
print()
print('-Dati in uscita')
print('=====')
print()
for i in range(len(dati)):
    print()
    print('Serie n°', i+1 , ' Uomo ' ,predi[i])

x=[]
for i in range (0,iterazioni,100):
    x.append(i)

fig, ax = plt.subplots(facecolor='teal', alpha=0.1)
plt.style.use('ggplot')
plt.tick_params(labelcolor='y')

ax.set_title('Costo in funzione delle iterazioni', color='y')
ax.set_xlabel('Iterazioni - Passo 100', color='y')
ax.set_ylabel('Costo', color='y')

ax.grid('True')
ax.plot(x,costo,'r',label="Tasso d'apprendimento=%.3f"%(eta,))
ax.legend(loc='best')

plt.show()

```