

TEHNICI DE OPTIMIZARE

LABORATOR 8

APRILIE 2020

Departamentul de Matematica si Informatica

FMI – UOC

Conf.dr. Elena PELICAN; email: epelican@univ-ovidius.ro

1. Rulati codul urmatoar care implementeaza un MLP (Multiple-Layer-Perceptron)

```
import numpy as np

def nonlin(x,deriv=False):
    if(deriv==True):
        return x*(1-x)
    return 1/(1+np.exp(-x))

X = np.array([[0,0,1],
              [0,1,1],
              [1,0,1],
              [1,1,1]])

y = np.array([[0],
              [1],
              [1],
              [0]])

np.random.seed(1)

#initializare ponderi
W0 = 2*np.random.random((3,5)) - 1
W1 = 2*np.random.random((5,1)) - 1

for j in range(60000):

    # propagare inainte pt layers 0, 1 si 2
    l0 = X
```

Email: epelican@univ-ovidius.ro

```
l1 = nonlin(np.dot(l0,W0))
l2 = nonlin(np.dot(l1,W1))

# evaluam eroarea dupa o trecere
l2_error = y - l2

if (j% 10000) == 0:
    print ("Error:" + str(np.mean(np.abs(l2_error))))

l2_delta = l2_error*nonlin(l2,deriv=True)

l1_error = l2_delta.dot(W1.T)

l1_delta = l1_error * nonlin(l1,deriv=True)

#actualizam ponderile cu metoda gradient descent
W1 += l1.T.dot(l2_delta)
W0 += l0.T.dot(l1_delta)
print(l2)
```

Cerinte:

- Comentati functia seed(). Ce observati?
 - Schimbati functia de activare (cu tangent hiperbolica tan, cu ReLU)
 - Modificati numarul de neuroni de pe nivelul ascuns
 - Modificati numarul de epoci
 - Modificati tipul de eroare
 - Modificati rata de invatare (in cod la mine, alfa=1)
 - Modificati optimizatorul (vezi ex 3)
 - Modificati ca sa avem bias (in codul acesta bias=0)
 - Modificati codul astfel ca sa aveti 2 sau 3 neuroni pe ultimul nivel (nu mai este perceptron in acest caz)
 - Modificati cosul dat de mine astfel ca datele sa fie cele din date.csv pe care le veti citi folosind Pandas (ca in lab 3, cred)
2. Repetati exercitiile anterioare, modificand optimizatorul. Incepeti cu Adams <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>
Continuati cu SGD, Nesterov, Momentum etc
 3. Repetati ex 1, implementand SLP (Single-Layer-Perceptron) pentru modelarea operatorilor logici NOT (operator unar), AND (operator binar) si OR (operator binar).
 4. Repetati ex 2, implementand MLP (Multi-Layer-Perceptron) pentru modelarea operatorului binary XNOR.
 5. Implementati cerintele de la ex 1 si ex 2, de pe pagina mea, TO_Lab8. Pdf, slide-ul 11 (formulele sunt pe slide-urile urmatoare) – sunt metodele de tip directive (de relaxare si de tip gradient, pe cazurile generale).