

DITA KRISNA

171011400341

07TPLE005

MULTI LAYER PERCEPTRON

```
import numpy as np

ls = np.array([2, 4, 4, 1])

n = len(ls)

W = []

for i in range(n - 1):

    W.append(np.random.randn(ls[i], ls[i + 1]) * 0.1)

B = []

for i in range(1, n):

    B.append(np.random.randn(ls[i]) * 0.1)

O = []

for i in range(n):

    O.append(np.zeros([ls[i]]))

D = []

for i in range(1, n):

    D.append(np.zeros(ls[i]))

A = np.matrix([[0.0, 0.0], [0.0, 1.0], [1.0, 0.0], [1.0, 1.0]])

#Target Vectors (1 row per each)

y = np.matrix([[-0.5], [0.5], [0.5], [-0.5]])

actF = []

dF = []

for i in range(n - 1):

    actF.append(lambda x : np.tanh(x))

    dF.append(lambda y : 1 - np.square(y))
```

```

actF.append(lambda x: x)
dF.append(lambda x : np.ones(x.shape))

a = 0.5

numIter = 250

for c in range(numIter):
    for i in range(len(A)):
        print(str(i))
        t = y[i, :]
        O[0] = A[i, :]
        for j in range(n - 1):
            O[j + 1] = actF[j](np.dot(O[j], W[j]) + B[j])
        print('Out:' + str(O[-1]))
        D[-1] = np.multiply((t - O[-1]), dF[-1](O[-1]))
        for j in range(n - 2, 0, -1):
            D[j - 1] = np.multiply(np.dot(D[j], W[j].T), dF[j](O[j]))
        for j in range(n - 1):
            W[j] = W[j] + a * np.outer(O[j], D[j])
            B[j] = B[j] + a * D[j]
        print('\nFinal weights:')
    for i in range(n - 1):
        print('Layer ' + str(i + 1) + ':\n' + str(W[i]) + '\n')

```