Thực hành Nguyên Lý Máy Học
Buổi 5: Perceptron + ôn tập

Mục tiêu:

- Viết hàm tìm trọng số w cho dữ liệu khả tách tuyến tính
- Sử dụng Sklearn để áp dụng giải thuật perceptron cho dữ liệu không khả tách
- Kiểm thử và đánh giá
- Ôn tập các giải thuật đã học

A. HƯỚNG DẪN THỰC HÀNH PERCEPTRON

1. Dữ liệu khả tách tuyến tính
    - Khởi tạo ngẫu nhiên các w
    - Đưa từng mẫu học qua perceptron và quan sát giá trị đầu ra
    - Nếu giá trị đầu ra khác với giá trị mong muốn, cập nhật lại các trọng số theo công thức:

$$w_j = w_j + \eta \cdot (y_i - o_i) \cdot x_{ij}, \forall j = 0..n$$

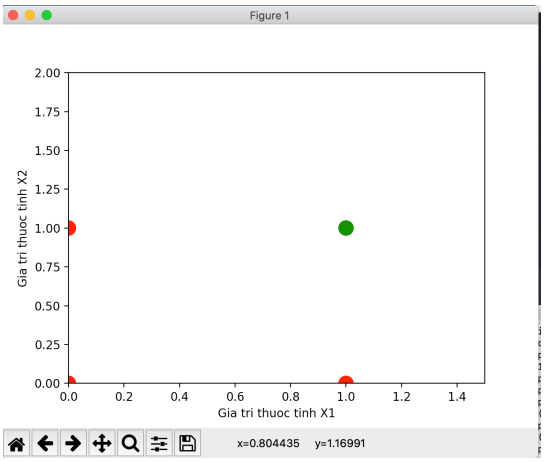
- ***Dữ liệu huấn luyện***

x1	x2	Y
0	0	0
0	1	0
1	0	0
1	1	1

```
import numpy as np
import matplotlib.pyplot as plt

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

X1  = np.array([[0,0],[0,1],[1,0],[1,1]])
X1

Y = np.array([0,0,0,1])
Y
```

- ***Biểu diễn dữ liệu để kiểm tra dữ liệu có khả tách hay không***

```
import matplotlib.pyplot as plt
colormap = np.array(['red','green'])
plt.axis([0,1.5,0,2])
plt.scatter(X[:,0],X[:,1], c=colormap[Y],s=150)
plt.xlabel("Gia tri thuoc tinh X1")
plt.ylabel("Gia tri thuoc tinh X2")
plt.show()
```



- ***Khởi tạo giá trị w0 và các w theo độ lớn của biến X***

  Với các trọng số w0 = -0.2,

  w1 =0.5, w2=0.5

  Tốc độ học: eta = 0.15

- ***Cài đặt giải thuật cập nhật các trọng số w0, w1, w2 dựa vào dữ liệu huấn luyện***

```
def my_perceptron(X, y, eta, lanlap):
    n = len(X[0,])
    m = len(X[:,0])
    print ("m =",m, " n =", n)
    w0 = -0.2 # kiểm tra kết quả theo bài học lý thuyết
    w = (0.5,0.5) # kiểm tra kết quả theo bài học lý thuyết
    print (" w0 =", w0)
    print (" w =", w)
    for t in range(0,lanlap):
        print("lanlap ____", t+1)
        for i in range(0,m):
            gx = w0 + sum(X[i,]*w)
            print ("gx = ", gx)
            if (gx>0):
                output = 1
            else:
                output = 0
            w0 = w0 + eta*(y[i]-output)
            w = w + eta*(y[i]-output)*X[i,]
            print (" w0 =", w0)
            print (" w =", w)
    return (np.round(w0,3), np.round(w,3))

my_perceptron(X, Y, 0.15, 2)
```

2. Dữ liệu không khả tách tuyến tính (sklearn)
   Cho tập dữ liệu có dạng:

	X1	X2	X3	X4	X5	Y
0	42000	5850	3	1	2	0
1	38500	4000	2	1	1	0
2	49500	3060	3	1	1	0
3	60500	6650	3	1	2	1
4	61000	6360	2	1	1	1
5	66000	4160	3	1	1	1
6	66000	3880	3	2	2	1
7	69000	4160	3	1	3	1
8	83800	4800	3	1	1	1
9	88500	5500	3	2	4	1
10	90000	7200	3	2	1	1

   Anh/chị hãy thực hiện các yêu cầu sau:
   - Sử dụng nghi thức hold-out để phân chia tập dữ liệu huấn luyện và kiểm tra
   - Sử dụng thư viện sklean để huấn luyện mô hình bằng giải thuật perceptron
   - Mô hình có bao nhiêu giá trị trọng số? Ghi lại các giá trị trọng số của mô hình mà các anh chị huấn luyện.
   - Dự đoán giá trị y cho các phần tử trong tập kiểm tra
   - Đánh giá độ chính xác của giải thuật cho các phần tử trong tập kiểm tra

Hướng dẫn

a. Đọc dữ liệu
   Đọc tập dữ liệu từ file data_per.csv (có sẵn trên elcit)
   read_csv() (thư viện pandas)
b. Phân chia dữ liệu
   train_test_split()
c. Sklearn cho giải thuật perceptron
   http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Perceptron.html

```
from sklearn.linear_model import Perceptron
net = Perceptron()
net.fit(X_train,y_train)
print(net)
```

   Net.coef_ net.intercef_ net.n_iter_
d. Đánh giá độ chính xác
   from sklearn.metrics import accuracy_score
   accuracy_score()

B. Bài tập:

1. **Tự viết hàm tương tự hàm my_perceptron() ở câu 1, với thay đổi là dữ liệu đầu vào có số lượng thuộc tính bất kỳ thay vì 2 thuộc tính. Thử nghiệm với tập dữ liệu data_per.csv**

Gợi ý:

Sử dụng hàm Random của thư viện numpy để tạo ngẫu nhiên các giá trị w
Ví dụ bên dưới giúp tạo ngẫu nhiên 5 giá trị.

```
>>> np.random.rand(5)
array([0.05487985, 0.84602334, 0.41868244, 0.47315742, 0.90349042])
>>> np.random.randn(5)
array([-0.3611232 , -1.01753306,  0.130839  , -1.34120705, -0.53243104])
```

2. Sử dụng hàm có sẵn của Sklearn với dữ liệu đầu vào là tập dữ liệu iris, thay đổi các giá trị tham số max_iter (5,50,100,1000), eta0 (0.002, 0.02, 0.2). Tính độ chính xác cho mỗi lần thay đổi tham số (In giá trị của các tham số và độ chính xác tương ứng).