

# Thực hành Nguyên Lý Máy Học

## Buổi 5: Perceptron

### 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

### HƯỚNG DẪN THỰC HÀNH PERCEPTRON

#### 1. Dữ liệu khả tách tuyến tính

##### *Dữ liệu huấn luyện*

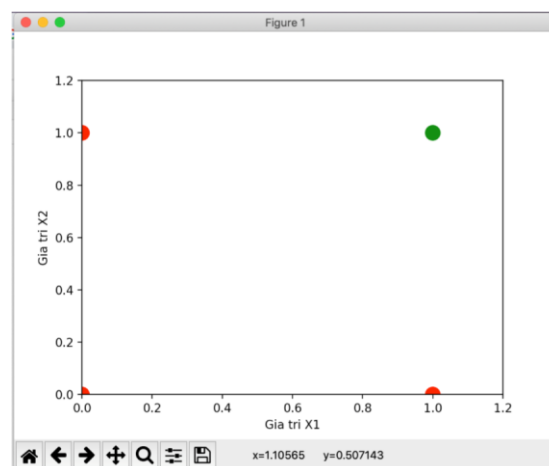
x1	x2	Y
0	0	0
0	1	0
1	0	0
1	1	1

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

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

##### *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([0,1.2,0,1.2])
plt.scatter(X[:,0],X[:,1], c=colormap[Y],s=150)
plt.xlabel("Gia trị X1")
plt.ylabel("Gia trị X2")
plt.show()
```



### **Khởi tạo giá trị $w_0$ và các $w$ theo độ lớn của biến $X$**

Với các trọng số  $w_0 = -0.2$ ,

$w_1 = 0.5, w_2 = 0.5$

Tốc độ học:  $\eta = 0.15$

### **Cài đặt giải thuật cập nhật các trọng số $w_0, w_1, w_2$ dựa vào dữ liệu huấn luyện**

```
import random
import numpy as np

def my_perceptron(X, y, eta, lanlap):
    n = len(X[0,])
    m = len(X[:,0])
    print ("m =",m, "và n =", n)
    #w0= random.random() # khởi tạo ngẫu nhiên w0
    #w = np.random.random(n) # khởi tạo ngẫu nhiên các giá trị w
    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)
        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 (w0, w)

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 sklearn 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
- 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 elcitr)*

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](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.intercept\_      net.n\_iter\_

d. Đánh giá độ chính xác

from sklearn.metrics import accuracy\_score

accuracy\_score()

### Bài tập

1. Cập nhật hàm my\_perceptron ở phần 1 với dữ liệu đầu vào có số lượng thuộc tính bất kỳ (thay vì 2 thuộc tính). Kiểm tra hàm vừa xây dựng trên 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 tạo ngẫu nhiên 5 giá trị sử dụng **rand** hoặc **randn**

```
>>> np.random.rand(5)
```

```
>>> np.random.randn(5)
```

<https://numpy.org/doc/stable/reference/random/generated/numpy.random.rand.html>

<https://numpy.org/doc/stable/reference/random/generated/numpy.random.randn.html>

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 kèm giá trị tham số và độ chính xác tương ứng)