



黄冈师范学院  
HUANGGANG NORMAL UNIVERSITY

# 人工智能与机器学习

Artificial Intelligence and Machine Learning

章节：实验7-感知机的NumPy实现

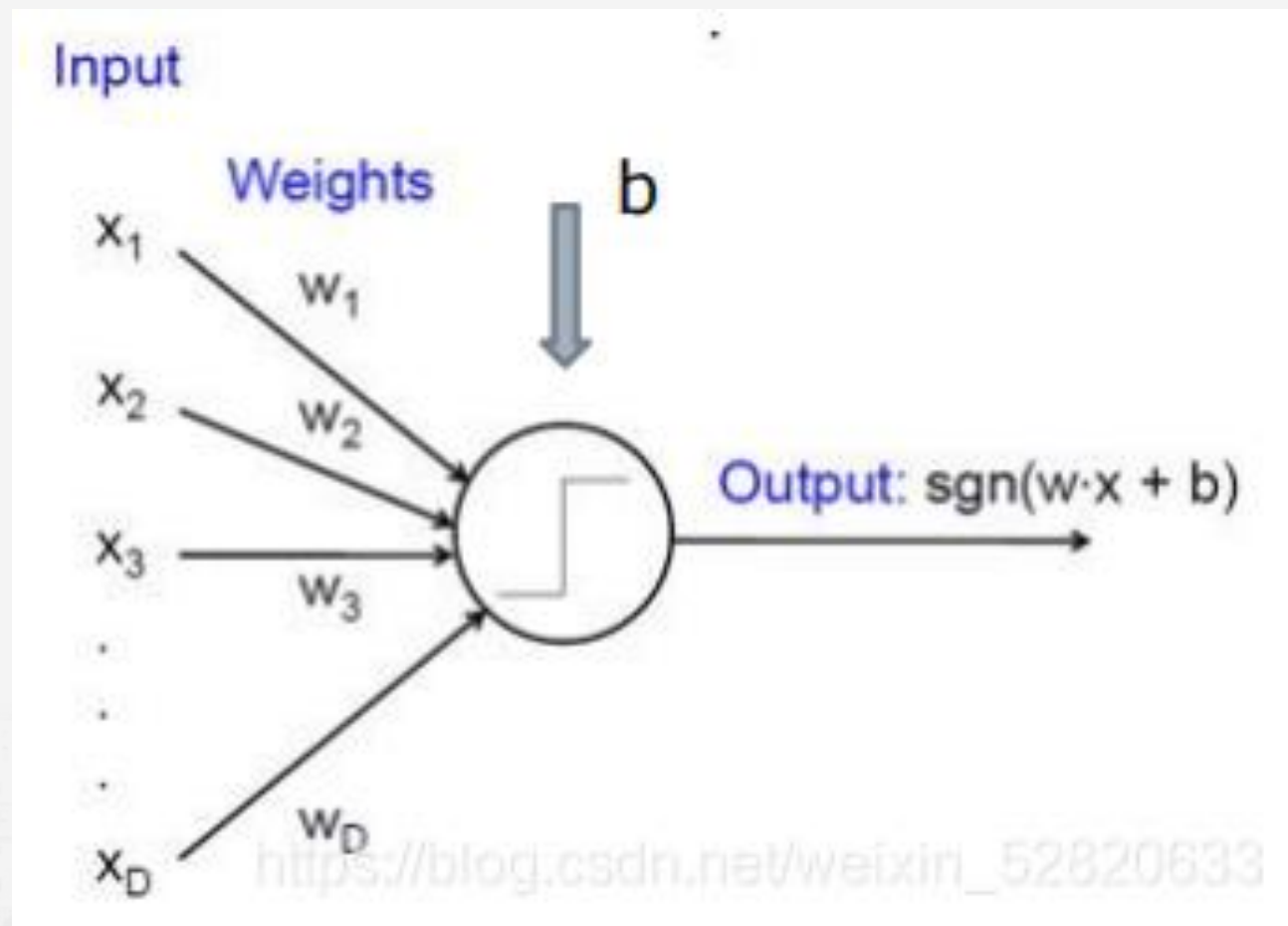
教师：刘重

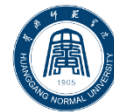
学院：计算机学院

厚德 博学 力行 致远

## 一、实验目的

- (1) 巩固感知机的结构和原理;
- (2) 巩固感知机的训练方法;
- (3) 练习感知机的NumPy实现。





## 二、实验内容

一、感知机推导

二、基于NumPy的感知机实现

1. 定义辅助函数

2. 定义感知机训练过程

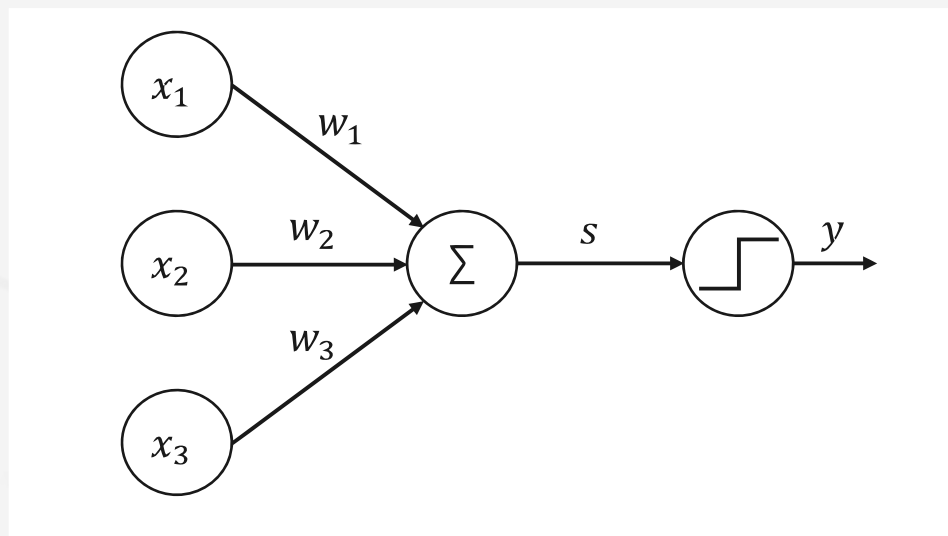
3. 测试数据准备

4. 感知机训练

5. 绘制感知机的线性分隔超平面

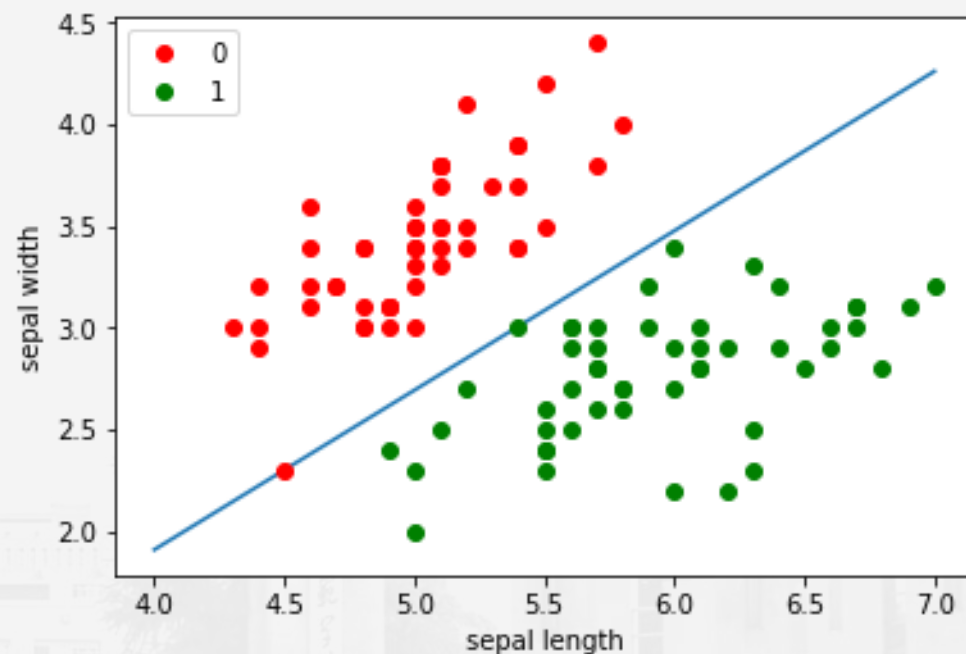
三、实验结果分析

- 感知机是神经网络和支持向量机的理论基础。从数学上来说，感知机就是一个线性模型，旨在建立一个线性超平面对线性可分的数据集进行分类。



$$y = \text{sign}(w \cdot x + b)$$

$$\text{sign}(x) = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$



## 感知机的数学原理

- 感知机的学习目标是建立一个**线性超平面**，以将训练数据正例和负例完全分开，我们可以通过**最小化损失函数来确定模型参数 $w$ 和 $b$** 。如何定义感知机的损失函数呢？一个方法是定义**误分类点到线性超平面的总距离**。假设输入空间中任意一点 $x_0$ 到超平面的距离为：

$$\frac{1}{\|w\|} w \cdot x_0 + b$$

- 对于任意一误分类点 $(x_i, y_i)$ ，当 $w \cdot x_i + b > 0$ 时， $y_i = -1$ ；当 $w \cdot x_i + b < 0$ 时， $y_i = +1$ ，因而都有 $-y_i(w \cdot x_i + b)$ 成立。所以误分类点到超平面的距离 $S$ 为：

$$-\frac{1}{\|w\|} y_i (w \cdot x_0 + b)$$

- 感知机的损失函数可以表示为：

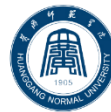
$$L(w, b) = - \sum_{x_i \in M} y_i (w \cdot x_0 + b) \quad \Rightarrow \quad \begin{cases} \frac{\partial L(w, b)}{\partial w} = - \sum_{x_i \in M} y_i x_i \\ \frac{\partial L(w, b)}{\partial b} = - \sum_{x_i \in M} y_i \end{cases} \quad \Rightarrow \quad \begin{cases} w = w + \alpha y_i x_i \\ b = b + \alpha y_i \end{cases}$$

$\alpha$ 为学习步长，也就是神经网络训练调参中的学习率

- 当一个实例被误分类时，即实例位于分离超平面的错误一侧时，需要调整参数 $w$ 和 $b$ 的值，使得分离超平面向该误分类点的一侧移动，以缩短该误分类点与超平面的距离，直到超平面越过该误分类点使其能够被正确分类。



## 四、问题描述



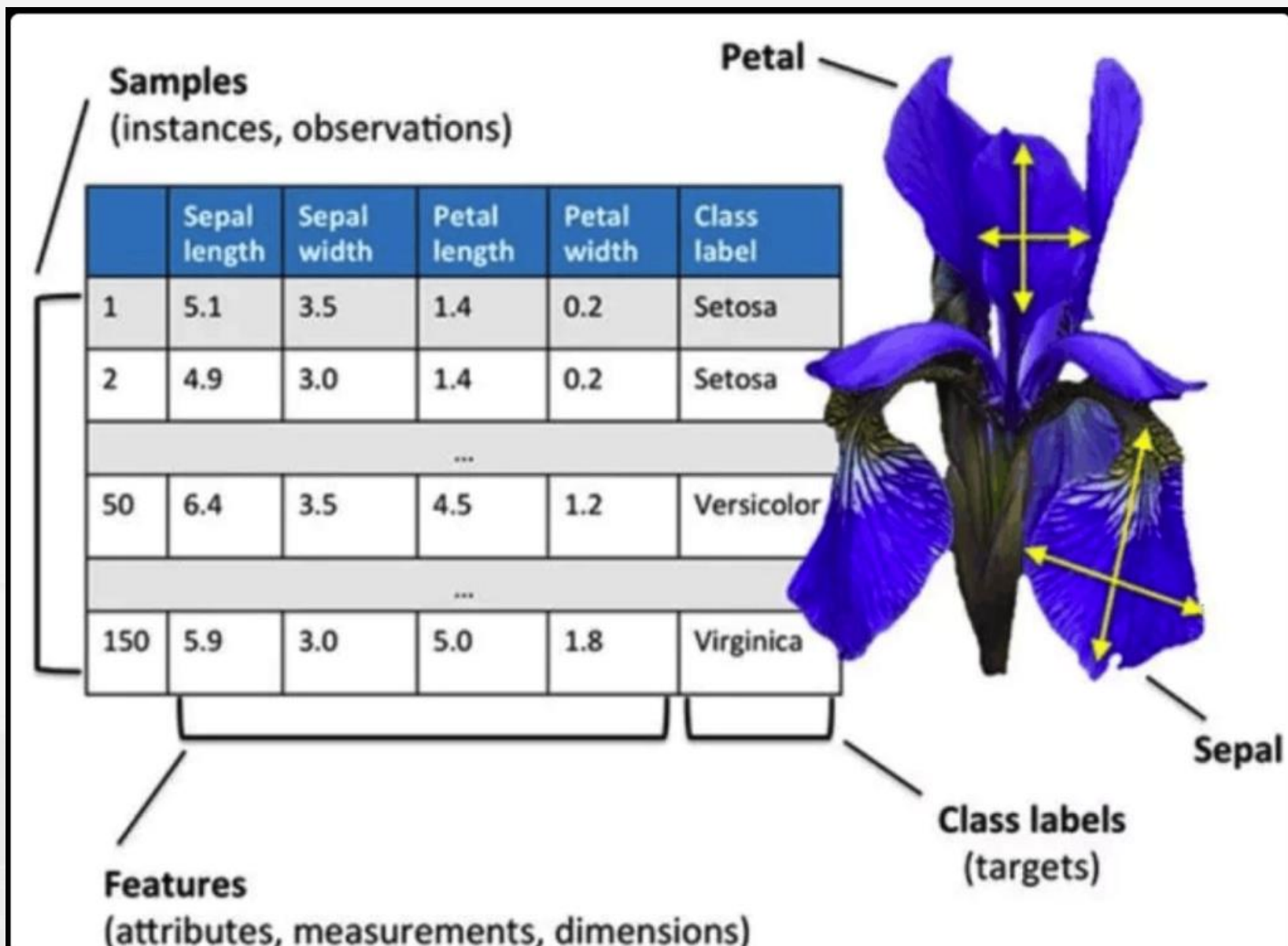
鸢尾花的四个属性分别是

- ❑ Sepal length--萼片长度;
- ❑ Sepal width--萼片宽度;
- ❑ Petal length--花瓣长度
- ❑ Petal Bwidth--花瓣宽度

花的3个类别分别是

- ✓ setosa (山鸢尾花) ;
- ✓ versicolor (杂色鸢尾花) ;

采用感知机模型实现鸢尾花的二值分类



## 感知机的Numpy实现

- 1、先定义感知机  
符号函数和参数  
初始化函数

```

8  # 导入相关库
9  import pandas as pd
10 import numpy as np
11 from sklearn.datasets import load_iris
12 import matplotlib.pyplot as plt
13
14 # 定义sign符号函数
15 def sign(x, w, b):
16     '''
17     ... 输入:
18     ... x: 输入实例
19     ... y: 权重系数
20     ... b: 偏置系数
21     ... 输出: 符号函数值
22     ... '''
23     return np.dot(x, w) + b
24
25 # 定义参数初始化函数
26 def initialize_parameters(dim):
27     '''
28     ... 输入:
29     ... dim: 输入数据维度
30     ... 输出:
31     ... w: 初始化后的权重系数
32     ... b: 初始化后的偏置参数
33     ... '''
34     w = np.zeros(dim, dtype=np.float32)
35     b = 0.0
36     return w, b
    
```

# 感知机的代码实

## 感知机的Numpy实现

- 2、定义感知机训练过程

```
38 # 定义感知机训练函数
39 def train(X_train, y_train, learning_rate):
40     '''
41     ... 输入: X_train: 训练输入; y_train: 训练标签; learning_rate: 学习率
42     ... 输出: params: 训练得到的参数
43     ... '''
44     ... w, b = initialize_parameters(X_train.shape[1]) ... # 参数初始化
45     ... is_wrong = False # 初始化误分类状态
46     ... while not is_wrong: # 当存在误分类点时
47         ... wrong_count = 0 # 初始化误分类点的计数
48         ... for i in range(len(X_train)): ... # 遍历训练数据
49             ... X = X_train[i]
50             ... y = y_train[i]
51             ... if y * sign(X, w, b) <= 0: ... # 如果存在误分类点
52                 ... # 更新参数
53                 ... w = w + learning_rate * np.dot(y, X)
54                 ... b = b + learning_rate * y
55                 ... wrong_count += 1 ... # 误分类点+1
56         ... # 直到没有误分类点
57         ... if wrong_count == 0:
58             ... is_wrong = True
59             ... print('There is no missclassification!')
60         ...
61         ... # 保存更新后的参数
62         ... params = {'w': w, 'b': b}
63     ... return params
```



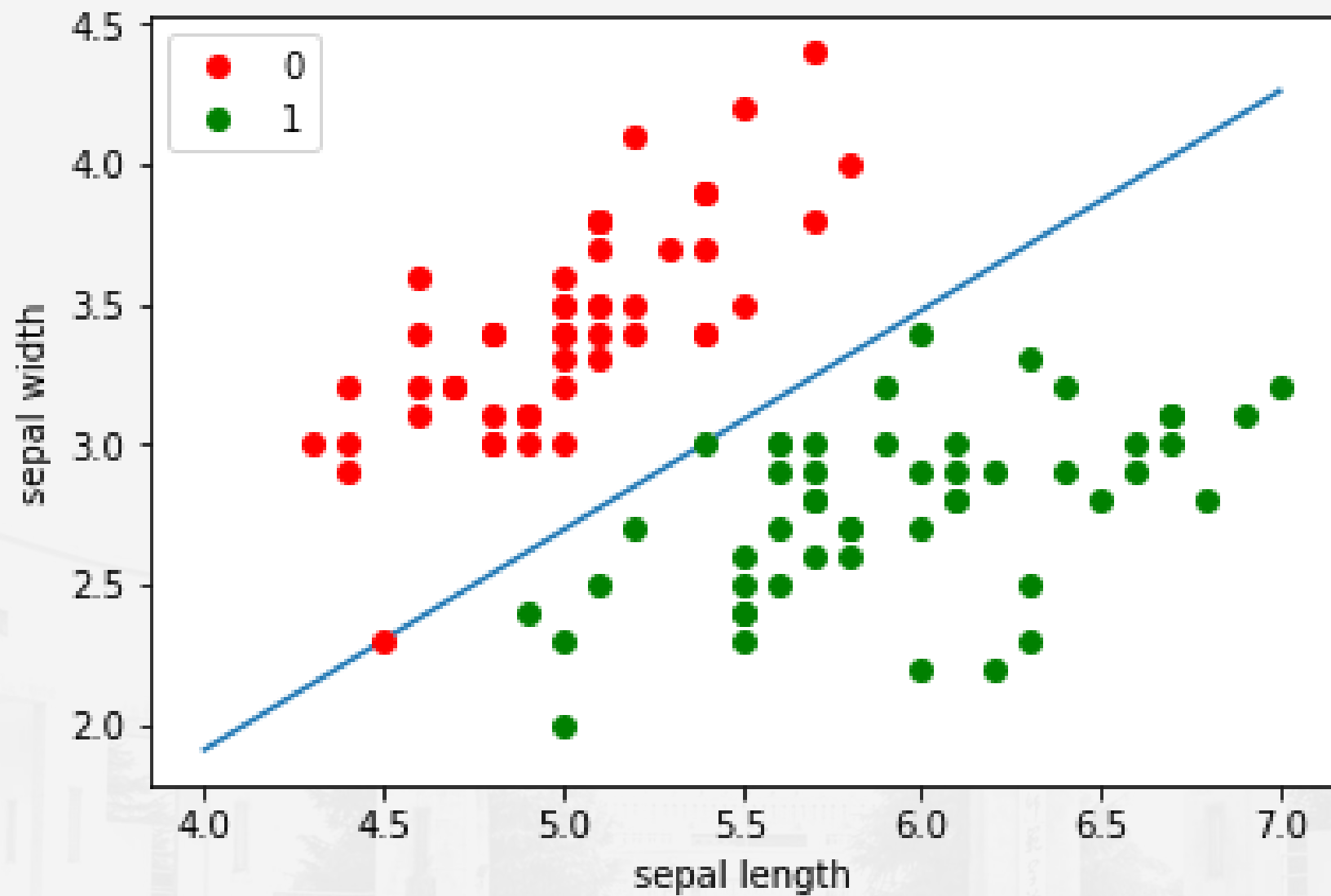
## 感知机的Numpy实现

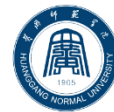
- 3、测试数据准备

```
67 # 生成测试数据
68 # 导入鸢尾花的数据集
69 iris = load_iris()
70 # 转化为Pandas数据框
71 df = pd.DataFrame(iris.data, columns=iris.feature_names)
72 # 数据标签
73 df['label'] = iris.target
74 # 变量重命名
75 df.columns = ['sepal_length', # 花萼长度
76               'sepal_width', # 花萼宽度
77               'petal_length', # 花瓣长度
78               'petal_width', # 花瓣宽度
79               'label'] # 花的种类(标签)
80 # 取前100行数据
81 data = np.array(df.iloc[:100, [0,1,-1]])
82 # 定义训练输入和输出
83 X, y = data[:, :-1], data[:, -1]
84 y = np.array([1 if i == 1 else -1 for i in y])
85 # 输出训练集大小
86 print(X.shape, y.shape)
```

- 4、感知机训练，并绘制感知机的线性分隔超平面

```
90 # 感知机训练
91 params = train(X, y, 0.01)
92 # 输出训练好的模型
93 print(params)
94
95 # 绘制感知机的线性分隔超平面
96 # 输入实例取反
97 x_points = np.linspace(4, 7, 10)
98 # 线性分隔超平面
99 y_hat = -(params['w'][0]*x_points + params['b'])/params['w'][1]
100 # 绘制线性分隔超平面
101 plt.plot(x_points, y_hat)
102
103 # 绘制二分类散点图
104 plt.plot(data[:50, 0], data[:50, 1], color='red', label='0')
105 plt.plot(data[50:100, 0], data[50:100, 1], color='green', label='1')
106 plt.xlabel('sepal length')
107 plt.ylabel('sepal width')
108 plt.legend()
109 plt.show()
```





## 六、实验报告要求

- 1、实验目的
  - 2、实验内容
  - 3、实验原理
  - 4、实验代码
  - 5、运行结果与分析
  - 6、实验小结
- 
- 说明：每个学生都要交电子版的实验报告，命名格式：
  - 01/02-XXXX（学号）-XXX（姓名）



黄冈师范学院  
HUANGGANG NORMAL UNIVERSITY

Q & A

> > > > > > > > > > > > > > > > >

< < < < < < < < < < < < < < < < <