实验一：构建感知机模型实现鸢尾花数据的分类

主讲教师：严资林

实验教师：匡慈维

目录

1 实验任务

2 数据说明

3 预备知识

4 实验步骤

5 实验要求

本学期实验总体安排

本学期实验课程共 10 个学时，5 个实验项目，总成绩为 20 分。

实验项目	一	二	三	四	五
学时	2	2	2	2	2
实验内容	感知机模型	决策树模型	K近邻模型	支持向量机模型	聚类模型
分数	3	4	4	4	5
上课时间(地点)	第11周 周四 (T2102)	第12周 周六 (T2102)	第14周 周四 (T2102)	第16周 周二 (T2102)	第17周 周四 (T2102)
检查方式	提交实验截图文档	提交实验报告、工程文件			

5-6节 3&4班； 7-8节 1&2班

线上腾讯会议：848-8762-6539

实验任务

- 鸢尾花分类是机器学习中比较经典的**入门式**教学课程。
- 构建一个**感知机**模型，根据鸢尾花的花萼和花瓣大小将其分为三种不同的品种。
- **任务一**：用Python自编程实现鸢尾花分类。
- **任务二**：用Sklearn库内的Perceptron分类器实现鸢尾花分类。
- **思考题：**见后面PPT



数据说明

◆ 数据集

- 总共包含150行数据
- 每一行数据由 4 个特征值及1个目标值组成。
- 4 个特征值分别为：

  花萼长度、花萼宽度、花瓣长度、花瓣宽度
- 目标值为三种不同类别的鸢尾花，分别为:

  Iris Setosa

  Iris Versicolour

  Iris Virginica

```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
5.4,3.7,1.5,0.2,Iris-setosa
```

预备知识

Numpy库介绍

- Numpy（Numerical Python的简称）是高性能科学计算和数据分析的基础包;
- Numpy 最重要的一个特点是其 N 维数组对象 **ndarray**，它是一系列同类型数据的集合，以 0 下标为开始进行集合中元素的索引;
- ndarray对象的内容可以通过**索引或切片**来访问和修改，与 Python 中 list 的切片操作一样。

- 在Windows系统中的安装命令 pip install numpy
- 在python代码中导入numpy库 import numpy as np

预备知识

Numpy库的基本操作

- 查看数组属性
- np.size、np.shape、np.ndim、 np.dtype

- 索引
- b= np.arange(5,20,2)、b[5]、b[2:5]

- 切片
- arr_slice = b[2:5]、 arr_slice[1] = 12345、 arr_slice[:]=64

更多学习请参考numpy中文网：https://www.numpy.org.cn/

预备知识

Pandas库介绍

- Pandas是Python第三方库，提供高性能易用数据类型和分析工具
- Pandas中有两大核心数据结构：Series（一维数据） 和 DataFrame（多特征数据，既有行索引，又有列索引）
- 安装命令 pip install pandas
- 导入pandas库 import pandas as pd

Series

index	value
0	12
1	4
2	7
3	9

Series的数据结构为"键值对"的形式

键—>可以重复

DataFrame

index	writer	title	price
0	mark	cookbook	23.56
1	barket	HTML5	50.70
2	tom	Python	12.30
3	job	Numpy	28.00

DataFrame可以进行行索引列索引

是Pandas中重要的数据结构

预备知识

Pandas库的基本操作

- 查看数据
  df.describe()、df.index 、df.columns 、df.values
- 选取特定列和行的数据
  df[col]、df[[col1,col2]]、df.iloc[[row1, row2], [col1, col2]]
- 增加、删除、修改列的值
  df.append( ) 、del pd[col] 、df.pop
- 导入导出文件
  df.read_csv、df.to_csv

更多学习请参考pandas中文网： https://www.pypandas.cn/

预备知识

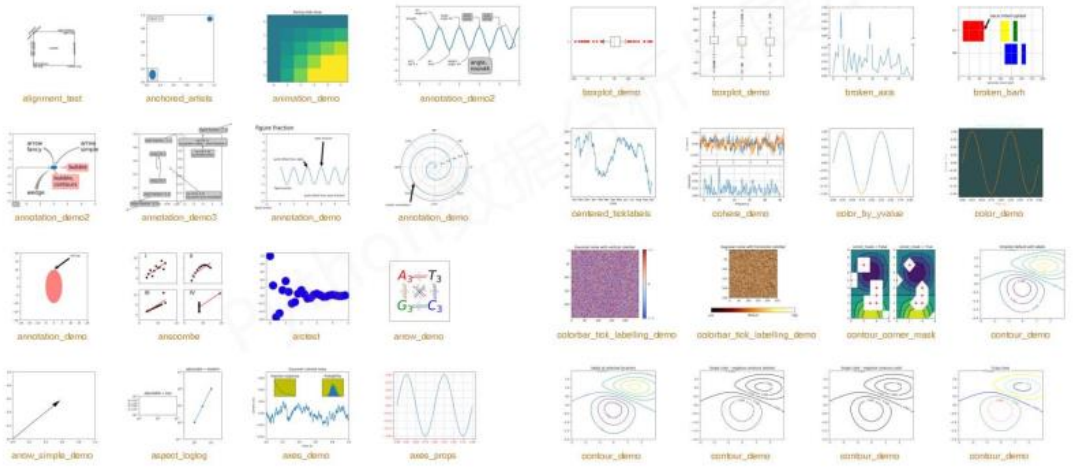
Matplotlib库介绍

- Matplotlib库由各种可视化类构成，内部结构复杂
- matplotlib.pylot是绘制各类可视化图形的命令字库，相当于快捷方式
- 安装命令 pip install matplotlib
- 导入命令matplotlib.pylot依赖包

```
import matplotlib.pyplot as plt
```

Matplotlib库的效果

http://matplotlib.org/gallery.html



预备知识

Sklearn库介绍

Sklearn (scikit-learn) 是基于 Python 语言的机器学习工具。在 sklearn 里面有六大任务模块分别是：

- Classification 分类　　Regression 回归
- Clustering 聚类　　Dimensionality reduction 数据降维
- Model Selection 模型选择　　Preprocessing 数据与处理

- 功能：只需要调用几行sklearn 库里的API即可实现一个复杂的机器学习算法
- 安装命令 pip install scikit-learn
- **更多学习请参考**scikit-learn**官网：** https://scikit-learn.org/stable/

任务一：实验步骤

◆实验步骤 （python自编程）

1、准备数据

✓读取数据，提取特征；

2、定义模型

3、训练模型

4、绘制图像

主函数代码参考如下：

```
if __name__ == '__main__':

    # 加载训练集
    df = create_df()
    # 可视化数据集
    show_image(df)

    #数据切片
    [data, X, y] = data_slice(df)

    #构造感知机对象，对数据集进行训练，得出模型参数
    perceptron = Model(data)
    perceptron.fit(X, y)

    #绘制图像
    show(data)
```

任务一：实验步骤

1、准备数据

◆ 导入必要的包

```python
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris
import matplotlib.pyplot as plt
```

◆ 加载数据集

```python
iris = load_iris()
print(iris.data.shape)  # data对应了样本特征
print(iris.target.shape)  # target对应了样本的类别（目标属性）
print(iris.target)  # 显示所有样本的目标属性
print(iris.target_names)    # 显示所有样本的目标属性名称
print(iris.feature_names) # 显示样本中的4个特征名称
```

输出结果：

```
(150, 4)
(150,)
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
['setosa' 'versicolor' 'virginica']
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
```

◆ 将列表式的数据转化为转换为DataFrame

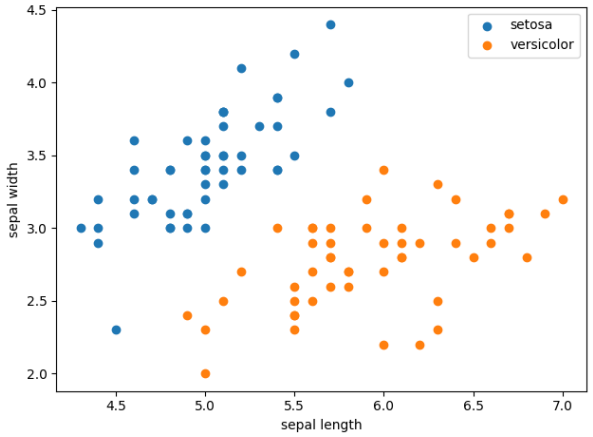
```python
# 将鸢尾花4个特征，以4列存入pandas的数据框架
df = pd.DataFrame(iris.data, columns=iris.feature_names)
# 在最后一列追加 加入（目标值）列数据
df['label'] = iris.target
# 显示df每一行的标签
df.columns = ['sepal length', 'sepal width', 'petal length', 'petal width', 'label']
print(df)
```

任务一：实验步骤

◆ **原数据可视化**

```
plt.scatter(df[:50]['sepal length'], df[:50]['sepal width'], label='setosa')
plt.scatter(df[50:100]['sepal length'], df[50:100]['sepal width'], label='versicolor')
plt.xlabel('sepal length')
plt.ylabel('sepal width')
plt.legend()
plt.show()
```

画图结果：



现象： sepal length和sepal width两个变量沿着一条“瘦”直线排列，所以是强相关的

结论：选取这两个变量对setosa和versicolor两类鸢尾花实现能线性分类。

◆ **数据切片（**例如：选取前100行，为setosa和versicolor两类鸢尾花数据**)**

```
data = np.array(df.iloc[:100, [0, 1, -1]])
X, y = data[:, :-1], data[:, -1]
for i in range(len(data)):
    if data[i,-1] == 0:
        data[i,-1] = -1
```

任务一：实验步骤

2&3、 定义与训练模型

◆ 先定义一个Model类

```
# 数据线性可分，二分类数据
class Model:
    def __init__(self,data):
        self.w = np.ones(len(data[0]) - 1, dtype=np.float32)
        self.b = 0
        self.l_rate = 0.1


    def sign(self, x, w, b):
        y = np.dot(x, w) + b
        return y

    # 随机梯度下降法
    def fit(self, X_train, y_train):
        """请根据右侧的伪代码，自行编写程序，计算出w和b"""
```

◆ 后调用

```
#构造感知机对象，对数据集进行训练，得出模型参数
perceptron = Model(data)
perceptron.fit(X, y)
```

伪代码

输入：训练数据集T={$(x_1, y_1)...(x_n, y_n)$}

(1) 选出初始值w_0, b_0以及学习率 η;

(2) 在训练数据集中选取数据(x_i, y_i)

(3) 如果$y_i(wx_i+b)\leq 0$:

$$w=w+\eta y_i x_i$$

$$b=b+\eta y_i$$

(4) 转至（2），直到训练集中没有误分类点

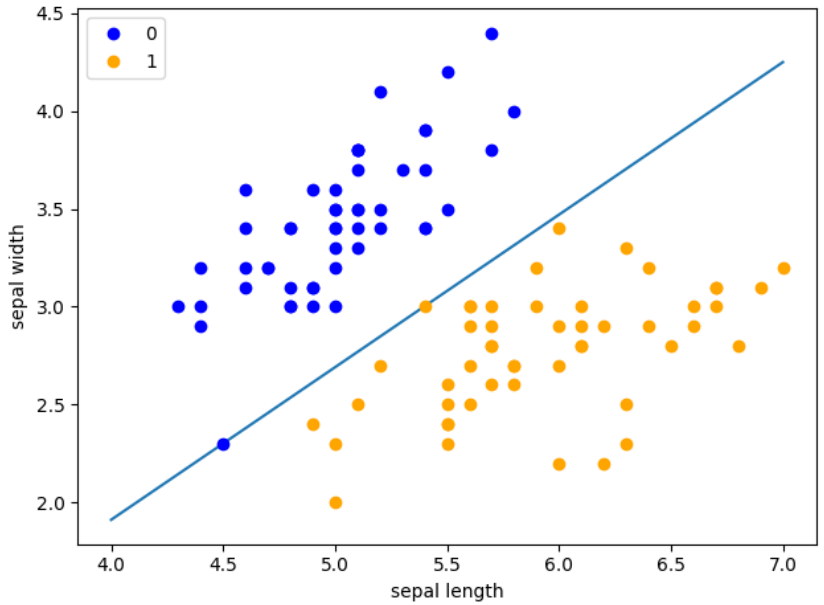
任务一：实验步骤

4、 绘制图像

```
x_points = np.linspace(4, 7, 10)
y_ = -(perceptron.w[0] * x_points + perceptron.b) / perceptron.w[1]
plt.plot(x_points, y_)

plt.plot(data[:50, 0], data[:50, 1], 'bo', color='blue', label='0')
plt.plot(data[50:100, 0], data[50:100, 1], 'bo', color='orange', label='1')
plt.xlabel('sepal length')
plt.ylabel('sepal width')
plt.legend()
plt.show()
```

画图结果：



（4.5 2.3）、（5.4 3.1）

注意：这两个点只是看起来近似在分类线上，实际上代入前面感知机模型，分别得到的是<0，>0，所以是能线性分类的。

任务一：实验要求

任务一：

1、对鸢尾花**setosa**和**virginica**两个品种做分类；

2、使用matplotlib画图做分析，选取2个合适的特征；

3、根据伪代码编写梯度下降法程序；

4、采用函数式编写python代码，重要代码加上注释；

5、结果绘图（带分类线）。

任务二：实验步骤

◆实验步骤 （使用Sklearn库来编程）

1、准备数据

✓读取数据，提取特征；

✓将数据分割为训练集和测试集

2、配置模型

3、训练模型

4、评估模型

✓计算权重矩阵和超平面截距

✓计算模型的准确率/精度

主函数代码参考如下：

```
if __name__ == '__main__':
    #加载数据集
    df = create_df()
    # 原数据可视化
    show_image(df)
    #数据切片
    data=create_data(df)
    #数据分割
    # X是除最后一列外的所有列，y是最后一列
    X, y = data[:, :-1], data[:, -1]
    # 调用sklearn的train_test_split方法，将数据随机分为训练集和测试集
    X_train, X_test, y_train, y_test = train_test_split(X,  #被划分的样本特征集
                                                        y,  #被划分的样本目标集
                                                        test_size=0.3, #测试样本占比
                                                        random_state=1) #随机数种子
    # 定义感知机
    clf = Perceptron(fit_intercept=False, max_iter=1000, shuffle=False)
    # 使用训练数据进行训练
    clf.fit(X_train, y_train)
    #计算模型的权重、截距、迭代次数
    print("特征权重：", clf.coef_)  # 特征权重 w
    print("截距（偏置）:", clf.intercept_)  # 截距 b
    print("迭代次数:", clf.n_iter_)
    #评价模型
    print(clf.score(X_test, y_test))
    #绘制图形，观察分类结果
    show(clf, data)
```

任务二：实验步骤

1、准备数据

◆ 导入必要的包

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.linear_model import Perceptron
from sklearn.model_selection import train_test_split
```

◆ 加载数据集

```
iris = load_iris()
print(iris.data.shape)  # data对应了样本特征
print(iris.target.shape)  # target对应了样本的类别（目标属性）
print(iris.target)  # 显示所有样本的目标属性
print(iris.target_names)  # 显示所有样本的目标属性名称
print(iris.feature_names) # 显示样本中的4个特征名称
```

输出结果：

```
(150, 4)
(150,)
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
['setosa' 'versicolor' 'virginica']
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
```

任务二：实验步骤

- **将列表式的数据转化为转换为DataFrame**
- **原数据可视化** （目的：找到两个强相关的变量）
- **数据切片**（例如：选取前100行，为setosa和versicolor两类鸢尾花数据）
- **数据分割**

```
# X是除最后一列外的所有列，y是最后一列
X, y = data[:, :-1], data[:, -1]
# 调用sklearn的train_test_split方法，将数据随机分为训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X,   #被划分的样本特征集
                                                    y,   #被划分的样本目标集
                                                    test_size=0.3, #测试样本占比
                                                    random_state=1) #随机数种子
```

注：test_size和random_state的值是可调整的

任务二：实验步骤

2、配置模型 （构造感知机分类器）

```
#clf = Perceptron()  #定义感知机
clf = Perceptron(fit_intercept=False, max_iter=1000, shuffle=False)
```

序号	部分重要参数	默认值	可选值
1	fit_intercept(计算模型的截距)	True	为False时，则数据中心化处理
2	max_iter(迭代次数)	1000	如果tol不为None，则为1000
3	tol (终止条件)	None	(previous_loss -loss)<tol, 比如 tol=1e-3
4	shuffle(每次迭代后清洗训练数据)	True	False
5	eta(学习率)	1	（0,1]
6	penalty (正则化项)	None	‘l2’or ‘l1’ or ‘elasticnet’
7	alpha (正则化系数)	0.0001	

更多参数学习请参考： https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Perceptron.html

任务二：实验步骤

3、训练模型

```
clf.fit(X_train, y_train)    #使用训练数据进行训练
```

4、评估模型

◆ 调用方法，计算模型的准确率

```
#计算模型的权重、截距、迭代次数
print("特征权重：", clf.coef_)  # 特征权重 w
print("截距（偏置）:", clf.intercept_)  # 截距 b
print("迭代次数:", clf.n_iter_)
#评价模型
print(clf.score(X_test, y_test))
```

输出结果：

```
特征权重： [[ 31.7 -57.1]]
截距（偏置）: [0.]
迭代次数: 30
0.9666666666666667
```

Perceptron模型的相关信息		
（1）Attributes (属性-变量)		
1	**coef_**	对应w
2	**intercept_**	对应b
3	**n_inter_**	迭代次数
（2）Methods (方法-函数)		
1	**fit**	用于训练数据集
2	**score**	用于评价测试结果

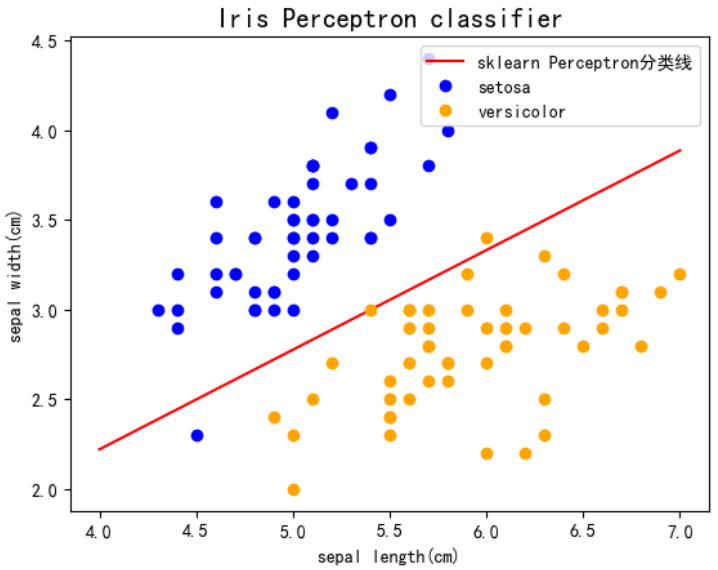
任务二：实验步骤

◆ 绘制图形，观察分类效果

```
x_ponits = np.arange(4, 8)
y_ = -(clf.coef_[0][0] * x_ponits + clf.intercept_) / clf.coef_[0][1]
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.plot(x_ponits, y_, 'r', label='sklearn Perceptron分类线')

plt.plot(data[:50, 0], data[:50, 1], 'bo', color='blue', label='setosa')
plt.plot(data[50:100, 0], data[50:100, 1], 'bo', color='orange', label='versicolor')
plt.xlabel('sepal length(cm)')
plt.ylabel('sepal width(cm)')
plt.title('Iris Perceptron classifier', fontsize=15)
plt.legend()
plt.show()
```

画图结果：



任务二：实验要求

任务二：

1、对鸢尾花**setosa和virginica**两个品种做分类；

2、使用matplotlib画图做分析，选取2个合适的特征；

3、调用sklearn库完成感知机模型的定义与训练；

4、调整感知机模型参数，使得测试结果的准确率为100%；

5、采用函数式编写python代码，重要代码加上注释；

6、结果绘图（带分类线）。

思考题

1、在做数据处理时，为什么要转化为dataFrame格式来处理，不能直接用numpy吗?

2、怎样挑选合适的特征来做分类，理由是什么?

3、为什么要使用随机种子做数据分割?

4、使用sklearn库来编码，学习率对迭代过程和最终结果有无影响？若有/无影响的话，条件是什么?

5、本次实验能对versicolor 和virginica两种鸢尾花做分类吗？能的话，实现出来；不能的话，说明理由。

提交方式

实验报告提交至平台 http://grader.tery.top:8000/#/courses

注意：

- 1、用户名、密码默认均为学号（若之前有修改过密码的，请用新密码登陆）；
- 2、请提交到相应的条目「2022春统计机器学习」课程 - 实验一；
- 3、提交截止时间：下周四 晚24点前；
- 4、文件夹&压缩包命名要求： 学号_姓名_统计机器学习实验一
- 5、提交内容：实验截图文档(pdf格式)，包含两个任务的实验代码和运行结果截图，以及思考题。

助教QQ：573044656　　助教QQ: 1198193933　　匡老师QQ：1197564837

学习资料

- 1、机器学习（李宏毅版本）

  https://aistudio.baidu.com/aistudio/education/group/info/1978

- 2、机器学习（吴恩达版本）

https://www.bilibili.com/video/BV1W34y1i7xK?p=1

- 3、机器学习个人笔记

百度网盘链接: https://pan.baidu.com/s/1SwPCecv6SoyW8T-QNPbVIQ 提取码:t555

统计机器学习实验

同学们，请开始实验吧！