**北京邮电大学软件学院**

**2018－2019学年第一学期实验报告**

**课程名称： 云计算数据中心**

**项目名称： 机器学习基础**

**项目完成人：**

**姓名：\_\_\_\_朱岩\_\_\_\_学号：\_\_\_\_2016522039\_\_\_\_**

**指导教师： \_\_管皓\_\_\_\_\_\_\_\_\_\_\_**

**日 期： 2018 年 12 月 28 日**

目录

[一、实验目的 3](#_Toc534390854)

[二、实验内容 3](#_Toc534390855)

[三、实验环境 3](#_Toc534390856)

[四、实验结果 4](#_Toc534390857)

[4.1 请可视化训练样本点。 4](#_Toc534390858)

[4.2 编程实现感知机算法。 4](#_Toc534390859)

[4.3 可视化训练完成的感知机。 5](#_Toc534390860)

[4.4 对5个未知样本点进行分类。 6](#_Toc534390861)

[五、遇到问题及解决方案 6](#_Toc534390862)

[5.1 遇到问题1： 6](#_Toc534390864)

[5.2 遇到问题2： 7](#_Toc534390865)

[六、实验总结： 7](#_Toc534390866)

[七、附录 8](#_Toc534390867)

[7.1 选用Python原因： 8](#_Toc534390870)

[7.2 流程图 9](#_Toc534390871)

[7.3 思路及实现: 10](#_Toc534390872)

[图表 1-可视化训练样本点 4](#_Toc534390816)

[图表 2-感知机算法重要代码 4](#_Toc534390817)

[图表 3-训练结果 5](#_Toc534390818)

[图表 4-可视化训练完成的感知机 5](#_Toc534390819)

[图表 5-测试数据可视化分类 6](#_Toc534390820)

[图表 6-txt文件读入 7](#_Toc534390821)

[图表 7-csv文件读入 7](#_Toc534390822)

[图表 8-流程图 10](#_Toc534390823)

[图表 9-训练数据 10](#_Toc534390824)

[图表 10-数据可视化部分代码 11](#_Toc534390825)

[图表 11-可视化原数据 11](#_Toc534390826)

[图表 12-设置初始参数值部分代码 11](#_Toc534390827)

[图表 13-参数更新代码 12](#_Toc534390828)

[图表 14-训练部分代码 12](#_Toc534390829)

[图表 15-两点确定一条直线 13](#_Toc534390830)

[图表 16-训练结果可视化 13](#_Toc534390831)

[图表 17-读取测试数据并可视化代码 13](#_Toc534390832)

[图表 18-测试数据可视化 14](#_Toc534390833)

[图表 19-测试数据代码 14](#_Toc534390834)

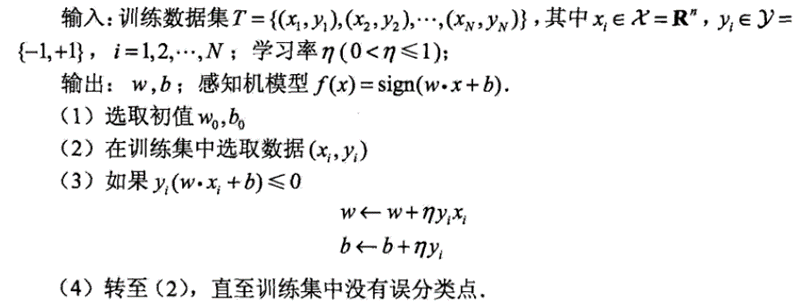
[图表 20-测试结果可视化 15](#_Toc534390835)

# 一、实验目的

学生通过编程，初步掌握机器学习中的感知机算法原理，为进一步研究更高级的智能算法（神经网络与深度学习）打下基础。

# 二、实验内容

**概述：**

感知机算法是近代神经网络的重要基础，它是一种二类分类的线性分类模型。其基础的算法流程如下：

**题目：**

**现有二维空间的样本点 (x,y),样本分为两类(类别1，类别-1)。现提供了20个数据样本及其标签供使用（train\_data）。有5个未知样本待分类(test\_data)。**

1. **请可视化训练样本点。**
2. **编程实现感知机算法。**
3. **可视化训练完成的感知机。**
4. **对5个未知样本点进行分类。**

# 三、实验环境

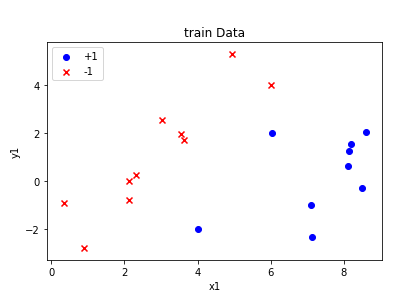
Windows操作系统

编程语言：python

编程工具：Spyder 3.7

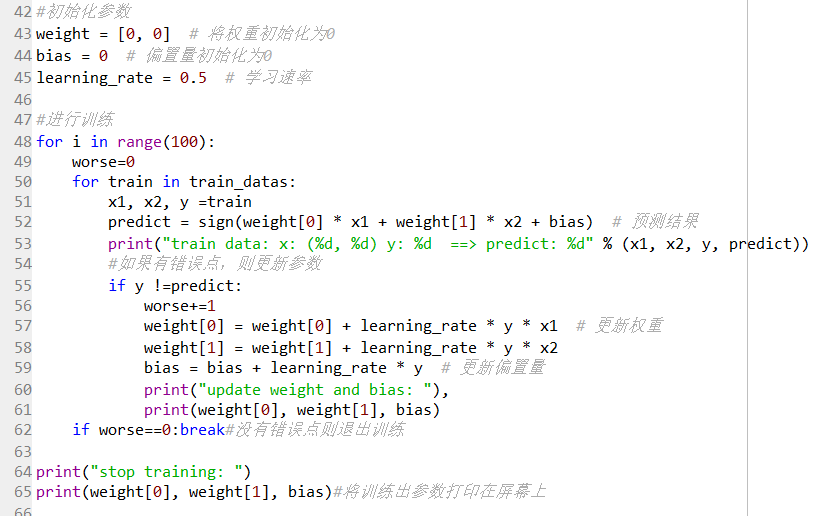
# 四、实验结果

## 请可视化训练样本点。

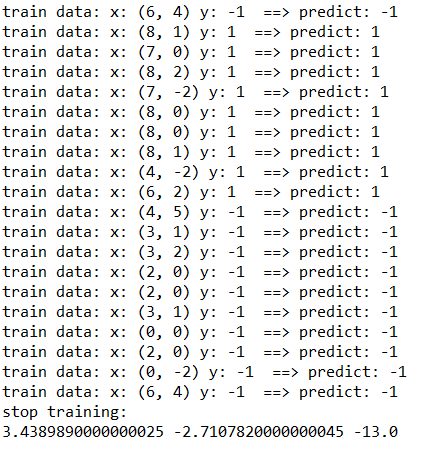


图表 1-可视化训练样本点

## 编程实现感知机算法。

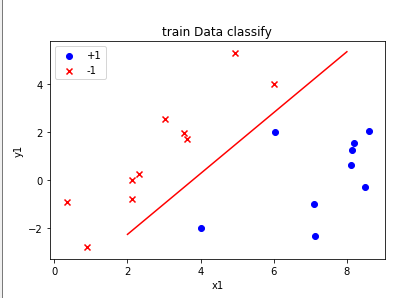


图表 2-感知机算法重要代码



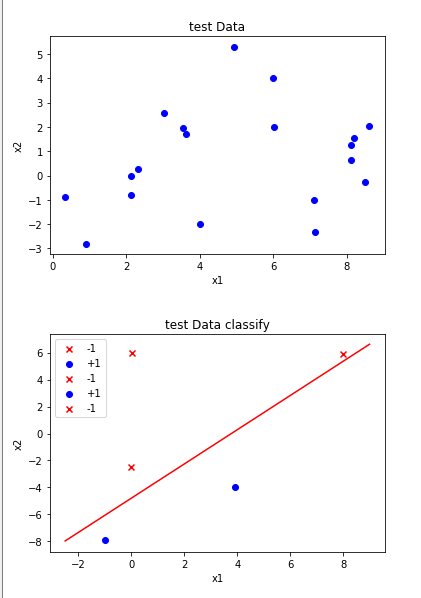
图表 3-训练结果

## 可视化训练完成的感知机。



图表 4-可视化训练完成的感知机

## 对5个未知样本点进行分类。



图表 5-测试数据可视化分类

# 五、遇到问题及解决方案

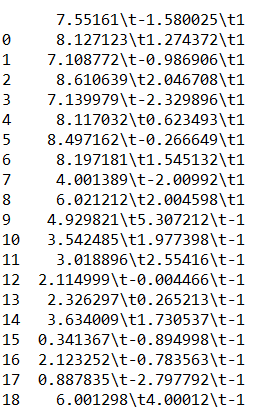


## 遇到问题1：

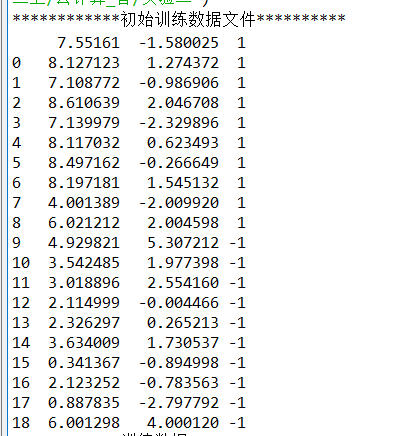
使用pandas读取txt文件，后续对文件按列进行拆分时出现错误。

解决方案：经输出中间结果发现pandas将txt文件中的制表符也当作数据存储下来，并不是根据制表符分隔，而是根据逗号分隔的，故将原数据转化为csv文件。

问题成功解决



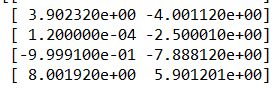
图表 6-txt文件读入



图表 7-csv文件读入

## 遇到问题2：

Pandas读数据时原本是五组数据但只显示了4行



解决方案：Pandas在读取文件时是认为第一行为表头，正式数据从第二行开始，故将自己将数据加上表头即可

# 六、实验总结：

通过本次实验，使我更加深入了解了感知机算法的原理及其实现，刚开始看到题目时真的无从下手，通过查找相关资料，和自己不断地尝试。最终理解并实现了感知机二分类算法。我感觉这个就像中学时期的数学题给定一个已知直线y=ax+b，问你某点是在直线上，还是在直线下。而感知机算法就是通过不断的猜测a和b这些参数将未知的分界线找出，然后根据点与线的位置关系判断该点属于哪类。如果数据集是二维的则分界线可能为直线或各种曲线，推广到三维，分界线可能为各种曲面，再向高维扩展就超出我的认知水平了，此时应该可以在预处理时应用降维技术来减少后续计算量。

# 七、附录



## 选用Python原因：

由于本实验主要是关于机器学习方面问题的，且目前主流的用于机器学习方面开发最主流的语言就是python，python对这类应用的支持度也相比其他语言支持度更高，相关的库更全，使得其更适用于本实验。

## 流程图



图表 8-流程图

## 思路及实现:

* + 1. **将excel文件转化为csv文件**
    2. **读取txt文件**

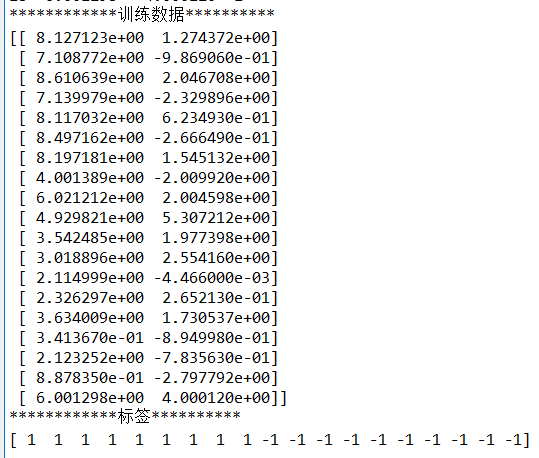
首先引用pandas库，方便后续对数据进行切片处理

import pandas as pd

data = pd.read\_csv('./train\_data. CSV')

X = data.iloc[:,:2].values，将前两列数据存入X中

y = data.iloc[:,2].values 将第三列标签存入y中



图表 9-训练数据

* + 1. **将原数据可视化**

**plt.scatter(X[:9, 0], X[:9, 1], color='blue', marker='o', label='+1')**

**将标签为+1的数据用蓝色圈圈表示出来**

**plt.scatter(X[9:, 0], X[9:, 1], color='red', marker='x', label='-1')**

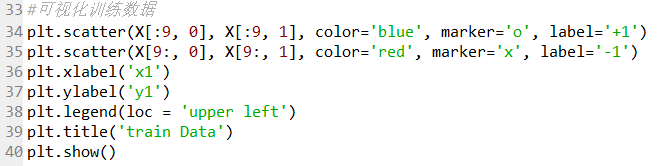
**将标签为-1的数据用蓝色圈圈表示出来**

**plt.xlabel('x1')//横轴**

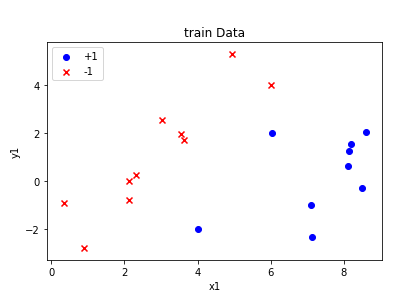
**plt.ylabel('y1')//纵轴**

**plt.title('Original Data')#图表的标题**

**plt.show()#将上述图表显示出来**



图表 10-数据可视化部分代码



图表 11-可视化原数据

* + 1. **对感知机的认识**

**公式：**



其中x是实例的特征向量

W是权重，维度与x维度相等

B是偏置

F(x)是预测值

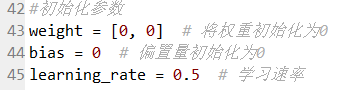
Y为真实值

* + 1. **编写感知机训练部分代码**
       1. 设置初始参数值

使用数组weight来存放权重

Weight[0]是x1的权重

Weight[1]是x2的权重



图表 12-设置初始参数值部分代码

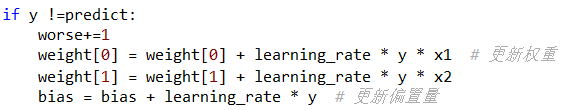
* + - 1. 进行训练

1. 设置最大迭代次数为100
2. 开始训练
3. 设置一个局部变量worse初始化为0,作用：标记是否有判断错误的点
4. If 所有数据训练完成 goto 6

else对每组数据进行预测sign(weight[0] \* x1 + weight[1] \* x2 + bias)//sign函数将括号中内容变为-1/1，方便比较

1. if预测值等于标签，则进行下一组数据，goto 4

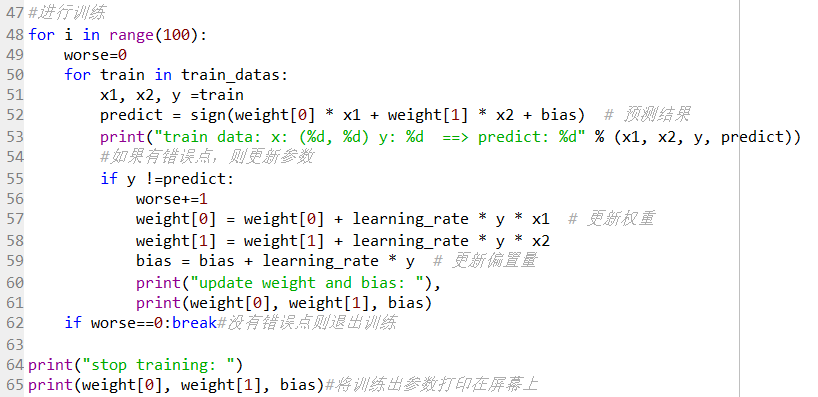
else 对参数进行更新 goto 2



图表 13-参数更新代码

1. if worse>0说明训练未完成，goto 2

else 说明我们已经拿到了我们想要的模型 break



图表 14-训练部分代码

* + 1. **将训练结果可视化**

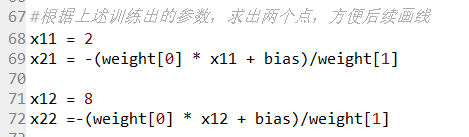
1. 将训练数据显示出来，与7.3.3相同
2. 将训练出的模型可视化（即将那条线显示出来）

0=weight[0]\*x1+weight[1]\*x2+bias

两点确定一条直线

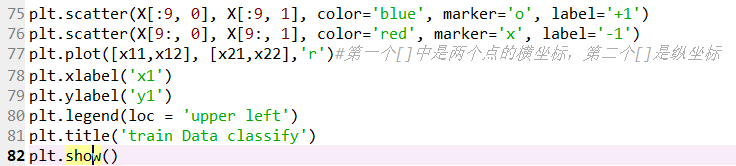
给定x1

则x2= - ( weight[0]\*x1+ bias)/ weight[1]



图表 15-两点确定一条直线

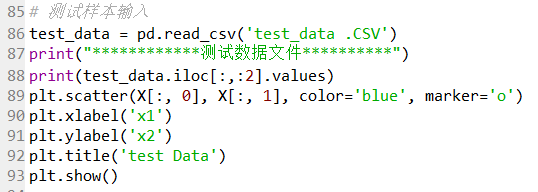
1. 可视化



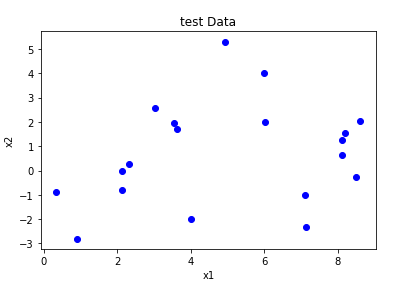
图表 16-训练结果可视化

* + 1. **对测试数据进行测试**

1. 读取测试数据（与7.3.2中读取训练数据基本相同）不同点就是测试数据只需读取两列即可，并将其展示在屏幕上



图表 17-读取测试数据并可视化代码



图表 18-测试数据可视化

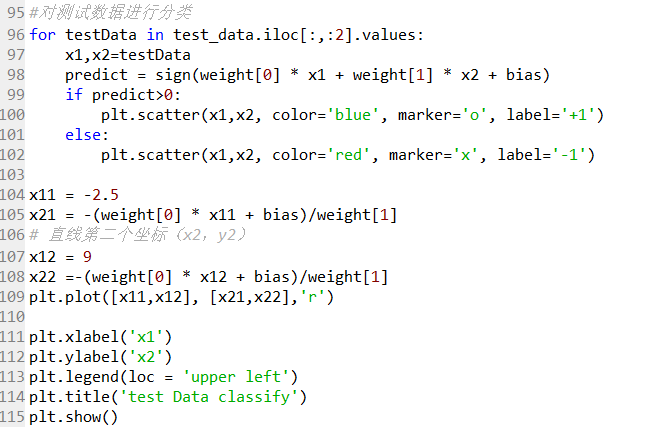
1. 对测试数据进行分类

使用公式sign(weight[0] \* x1 + weight[1] \* x2 + bias)

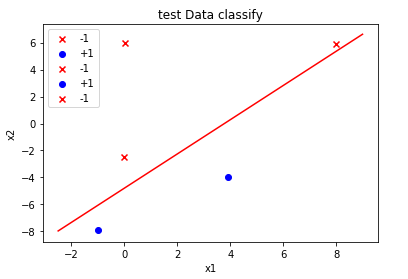
大于0即为1

小于0即为-1

1. 测结果可视化（与训练结果可视化基本相同）



图表 19-测试数据代码



图表 20-测试结果可视化