**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：李春昊\_\_\_\_\_\_\_\_学号：\_\_2016522049\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

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

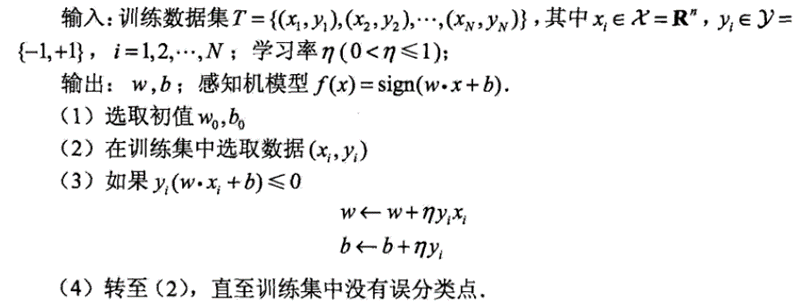
**日 期： 2019 年 1 月 10 日**

1. **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。

1. **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：**

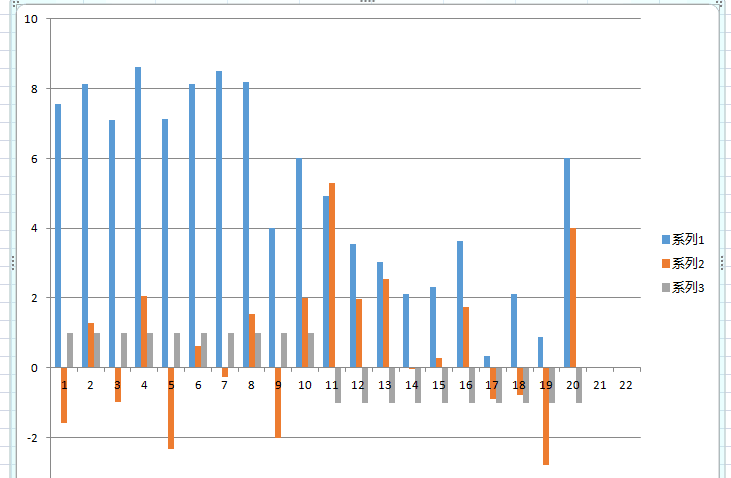
**现有二维空间的样本点 (x,y),样本分为两类(类别1，类别-1)。现提供了20个数据样本及其标签供使用（train\_data）。有5个未知样本待分类(test\_data)。**

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。**
5. **实验环境**

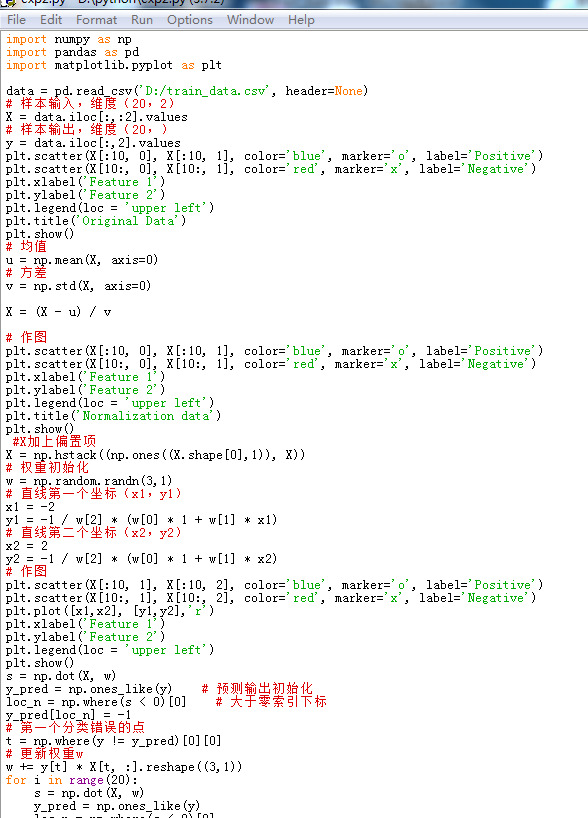
Windows操作系统

1. **实验结果**

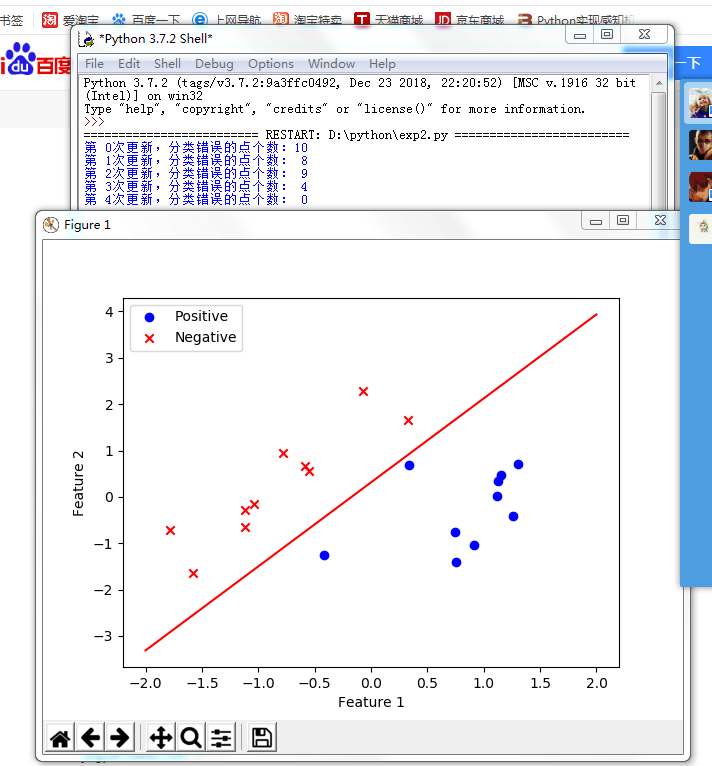
**请可视化训练样本点:**



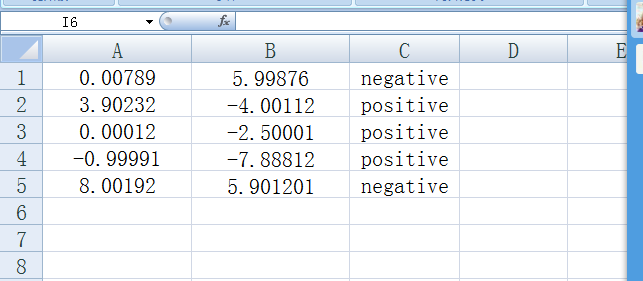
**编程实现感知机算法:**



**可视化训练完成的感知机。**



**对5个未知样本点进行分类。**



1. **附录**

**import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**data = pd.read\_csv('D:/train\_data.csv', header=None)**

**# 样本输入，维度（20，2）**

**X = data.iloc[:,:2].values**

**# 样本输出，维度（20，）**

**y = data.iloc[:,2].values**

**plt.scatter(X[:10, 0], X[:10, 1], color='blue', marker='o', label='Positive')**

**plt.scatter(X[10:, 0], X[10:, 1], color='red', marker='x', label='Negative')**

**plt.xlabel('Feature 1')**

**plt.ylabel('Feature 2')**

**plt.legend(loc = 'upper left')**

**plt.title('Original Data')**

**plt.show()**

**# 均值**

**u = np.mean(X, axis=0)**

**# 方差**

**v = np.std(X, axis=0)**

**X = (X - u) / v**

**# 作图**

**plt.scatter(X[:10, 0], X[:10, 1], color='blue', marker='o', label='Positive')**

**plt.scatter(X[10:, 0], X[10:, 1], color='red', marker='x', label='Negative')**

**plt.xlabel('Feature 1')**

**plt.ylabel('Feature 2')**

**plt.legend(loc = 'upper left')**

**plt.title('Normalization data')**

**plt.show()**

**#X加上偏置项**

**X = np.hstack((np.ones((X.shape[0],1)), X))**

**# 权重初始化**

**w = np.random.randn(3,1)**

**# 直线第一个坐标（x1，y1）**

**x1 = -2**

**y1 = -1 / w[2] \* (w[0] \* 1 + w[1] \* x1)**

**# 直线第二个坐标（x2，y2）**

**x2 = 2**

**y2 = -1 / w[2] \* (w[0] \* 1 + w[1] \* x2)**

**# 作图**

**plt.scatter(X[:10, 1], X[:10, 2], color='blue', marker='o', label='Positive')**

**plt.scatter(X[10:, 1], X[10:, 2], color='red', marker='x', label='Negative')**

**plt.plot([x1,x2], [y1,y2],'r')**

**plt.xlabel('Feature 1')**

**plt.ylabel('Feature 2')**

**plt.legend(loc = 'upper left')**

**plt.show()**

**s = np.dot(X, w)**

**y\_pred = np.ones\_like(y) # 预测输出初始化**

**loc\_n = np.where(s < 0)[0] # 大于零索引下标**

**y\_pred[loc\_n] = -1**

**# 第一个分类错误的点**

**t = np.where(y != y\_pred)[0][0]**

**# 更新权重w**

**w += y[t] \* X[t, :].reshape((3,1))**

**for i in range(20):**

**s = np.dot(X, w)**

**y\_pred = np.ones\_like(y)**

**loc\_n = np.where(s < 0)[0]**

**y\_pred[loc\_n] = -1**

**num\_fault = len(np.where(y != y\_pred)[0])**

**print('第%2d次更新，分类错误的点个数：%2d' % (i, num\_fault))**

**if num\_fault == 0:**

**break**

**else:**

**t = np.where(y != y\_pred)[0][0]**

**w += y[t] \* X[t, :].reshape((3,1))**

**# 直线第一个坐标（x1，y1）**

**x1 = -2**

**y1 = -1 / w[2] \* (w[0] \* 1 + w[1] \* x1)**

**# 直线第二个坐标（x2，y2）**

**x2 = 2**

**y2 = -1 / w[2] \* (w[0] \* 1 + w[1] \* x2)**

**# 作图**

**plt.scatter(X[:10, 1], X[:10, 2], color='blue', marker='o', label='Positive')**

**plt.scatter(X[10:, 1], X[10:, 2], color='red', marker='x', label='Negative')**

**plt.plot([x1,x2], [y1,y2],'r')**

**plt.xlabel('Feature 1')**

**plt.ylabel('Feature 2')**

**plt.legend(loc = 'upper left')**

**plt.show()**

**实验心得：**

**通过这次实验使我了解到了一些关于机械学习的基础知识。通过查阅资料完成了感知机的算法，并且对其训练数据进行了可视化。通过可视化的数据完成了对于测试集的划分。经过这次实验我增强了动手的能力，同时锻炼了我对于python的各种库的了解与安装。使我受益颇深。**