实验1 机器学习实验基础(Numpy)

实验目的

1. 熟悉机器学习实验环境。
2. 了解Python和Numpy的基本用法。
3. 能够使用Jupyter Notebook进行相关实验。

1. 实验环境

序号	环境项	内容	版本
1	开发环境（IDE）	VS Code + Jupyter NoteBook	
2	编程语言	Python 3.0以上	
3	基础类库	Numpy、SciPy、Matplotlib	
4	机器学习类库	Scikit-learn	
5	深度学习框架	MindSpore	

1.1 使用AnaConda安装包

前4项环境安装可以使用AnaConda进行统一安装，下面是安装包的链接。

· windows版本：https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2022.10-Windows-x86_64.exe

· Mac版本https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2022.10-MacOSX-x86_64.sh

可将AnaConda看作手机的应用商店，或者游戏里的steam平台。

1.2 深度学习框架MindSpore的安装

这可以在后面的实验再安装，MindSpore的安装包链接在：https://www.mindspore.cn/install

2. 基本概念介绍

2.1 Python与Jupyter NoteBook

Python 是一种解释型、面向对象、动态数据类型的高级程序设计语言。由 Guido van Rossum 于 1989 年底发明，第一个公开发行版发行于 1991 年。在Web爬虫、数据分析和机器学习等领域取得巨大成功。

- Python教程：https://www.liaoxuefeng.com/wiki/1016959663602400

Jupyter NoteBook是一个基于网页的交互式开发环境，它允许用户创建和分享包含**实时代码**、**markdown文本**、**可视化图表**和**数学公式**的文档。

2.2 Numpy

NumPy(Numerical Python) 是Python 的一种开源的多维数据（矩阵）存取计算的类库。支持大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。是各类机器学习、数据统计类库的基础。

- NumPy 官网 http://www.numpy.org/
- NumPy 源代码：https://github.com/numpy/numpy
- 中文参考：https://www.runoob.com/numpy/numpy-tutorial.html

2.3 Matplotlib

Matplotlib是Python 的绘图库，它能让使用者很轻松地将数据图形化，并且提供多样化的输出格式。可以用来绘制各种静态，动态，交互式的图表。在本课实验中，Matplotlib主要用来绘制线图、散点图、等高线图、3D 图形等。

- Matplotlib 官网：https://matplotlib.org/
- Matplotlib 源代码：https://github.com/matplotlib/matplotlib
- 中文参考：https://www.runoob.com/matplotlib/matplotlib-tutorial.html

2.4 SciPy（第三次实验再介绍）

SciPy是基于Python的开源的科学计算库，提供一系列的最优化、积分、插值、特征值计算、代数方程、微分方程和统计等功能的类库。

本课实验主要使用scipy.optimize.minimize()函数

- SciPy 官网：https://www.scipy.org/
- SciPy 源代码：https://github.com/scipy/scipy
- 中文参考： https://www.runoob.com/scipy/scipy-tutorial.html

3. Numpy实验

3.1 数据读取

使用numpy.loadtxt()读取波特兰市房价数据（ex1data2.txt），并使用ndarray.ndim和ndarrray.shape显示数据的维数和各维的尺寸。

```
import numpy as np
print('Loading data ...')
data = np.loadtxt('./house_price.txt', delimiter=',')

print(data.shape, data.ndim)
print("Size of data is :", data.shape, ", and dimension is", data.ndim)

m = data.shape[0]
n = data.shape[1]
print(m, n)
```

数据集共有47条数据，每条数据有3维（列）。

3.2 数据切片与索引

3.2.1 分别读取不同的列

```
X = data[:, 0:2] # q取[0，2）这个区间的列，不包括2.
X.shape
y = data[:, 2]
y.shape
y = data[:, [2]]
y.shape
```

3.2.2 读取数据

```
print('First 10 examples from the datase(', m ,'): ')
for i in range(10):
    print('x = (', X[i, 0], ',', X[i, 1], '); y = ', y[i, 0] )
```

3.2.3 根据索引读取数据

```
index = y < 200000
print(index)
cheapHouse = X[index,:]
print(cheapHouse)
```

3.3 数组创建

3.3.1 创建空数组numpy.empty()

3.3.2 创建全零数组numpy.zeros()

3.3.3 创建全一数组numpy.ones()

3.3.4 创建随机数组numpy.random.random()

3.4 数组操作

3.4.1 矩阵转置numpy.transpose()和ndarray.T

3.4.2 矩阵拼接numpy.vstack()、numpy.hstack()

numpy.c_() and numpy.r_()的用法

> np.r_是按列连接两个矩阵，就是把两矩阵上下相加，要求列数相等。
>
> np.c_是按行连接两个矩阵，就是把两矩阵左右相加，要求行数相等。

3.4.3 将n维数组拉伸成一维数组ndarray.flatten()

3.4.4 数组改变形状ndarray.reshape()

3.5 向量/矩阵运算

3.5.1 向量点积和矩阵乘 numpy.dot()和@运算符

3.5.2 矩阵求逆 numpy.linalg.inv()

使用numpy.linalg.inv()求解矩阵的逆（inverse）。

3.5.3 矩阵求行列式 numpy.linalg.det()

使用numpy.linalg.det()求解矩阵的行列式（determinant）。

4. 数据可视化

二类数据集的第三列是它们的类别，取值为0或1，首先将数据集分成正例（y==1）和负例（y==0）两部分。

```
1  data = np.loadtxt('./2classes.txt', delimiter=',')
2  X = data[:, 0:2]
3  y = data[:, 2]
4  pos = x[y == 1]
5  neg = x[y == 0]
6  plt.scatter(pos[:,0], pos[:,1], c='k', marker='+')
7  plt.scatter(neg[:,0], neg[:,1], c='y', marker='o', edgecolors='k',
   linewidths=0.5)
8  plt.show
```