

实验一 Regression

一、说明

- 实验采用 jupyter notebook, 请填写完代码后提交完整的 ipynb 文件
- 文件命名 规则: 班级_姓名_ML2019_HW1. ipynb, 如计科_1701_张三_ML2019_HW1. ipynb
- 提交方式: 采用在线提交至:
<http://pan.csu.edu.cn:80/invitation/8ff7d0cf-1bc4-41aa-9a93-2a33c156cbae>
- 实验提交截至日期: 2019.9.30 23:59

二、实验内容

本实验在一个具体的应用中逐步地指导用户训练出一个Regression模型。
实验使用的训练方法有梯度下降法和正规方程法。

梯度下降法是机器学习中常用的用于训练模型方法。梯度下降法通过使参数往负梯度方向移动的方法不断降低模型的损失函数值, 训练得到拟合训练数据的模型。本实验会实现梯度下降法来解决具体问题, 并直观展示梯度下降的过程。实验中会体现特征的归一化、学习率和梯度下降的迭代次数对训练的影响。

正规方程法是求线性回归模型的另一种方法, 通过具体公式可以直接求出线性回归模型的参数。本实验会实现正规方程法来解决具体问题。

三、实验目标

- 掌握搭建 python 的开发环境, 并能够使用 numpy 工具。
- 掌握特征的归一化处理。
- 掌握梯度下降法的具体过程, 并能够实现梯度下降法, 能够根据具体情况选择适当的超参数、学习率、迭代次数。
- 掌握单变量、多变量线性回归模型具体原理。

四、实验操作步骤

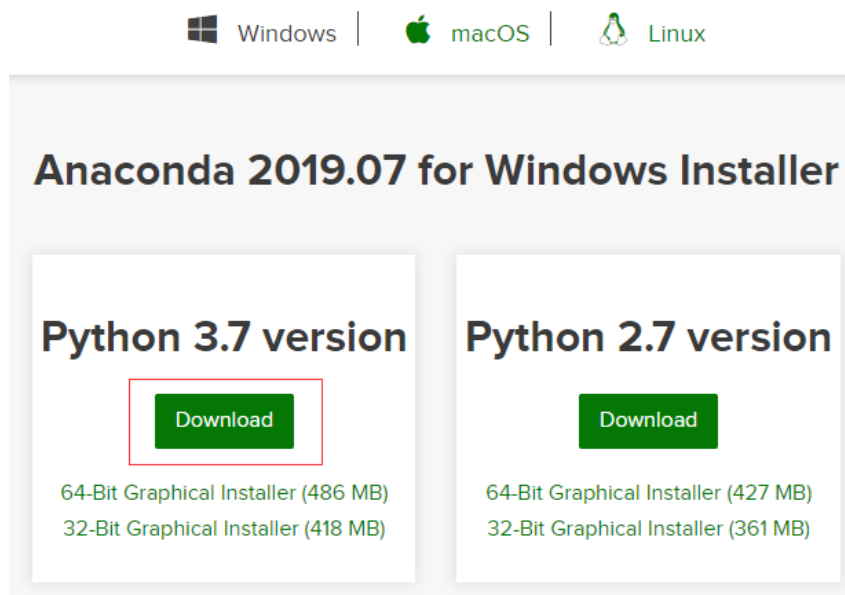
1. 搭建 python 环境

本实验需要用到的 python 环境包括


名称	版本
python	≥ 3.6
numpy	≥ 1.14
matplotlib	≥ 2.2
jupyter	≥ 1.0

推荐安装 anaconda，下载页面 <https://www.anaconda.com/download/>。

根据操作系统选择不同版本，这里演示 Windows 版的安装过程。



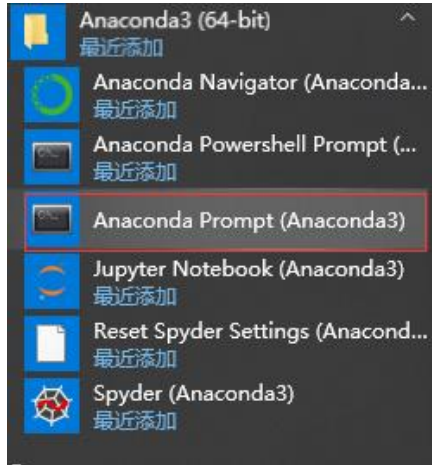
双击运行下载好的文件

 Anaconda3-2019.07-Windows-x86_64.exe

在安装界面一路选择 next，安装目录可以自己选择，比如 C:\Users\HZJ\Anaconda3
安装完毕后，python、numpy、matplotlib 和 jupyter 都安装成功。

2. 启动 jupyter notebook

在 **Windows 系统** 中，在启动目录中选择 Anaconda > Anaconda Prompt

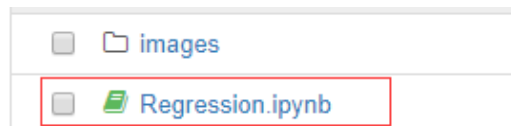


在弹出的命令行中 `cd` 到实验目录 打开 `jupyter notebook` 比如实验目录在 `"D:\experiment1\"` 输入下列命令

```
(base) C:\Users\HZJ>D:
(base) D:\>cd D:\experiment1\
(base) D:\experiment1>jupyter notebook
```

然后弹出 `jupyter notebook` 的页面。

在弹出的 `jupyter notebook` 页面，可以看到有文件 `Regression.ipynb`（如下图），单击打开它。



在新的页面中可以看到具体的实验内容。

A screenshot of the Jupyter Notebook interface. The top bar shows the notebook name 'jupyter Regression_answer (自动保存)' and a 'Logout' button. Below the top bar is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. A toolbar contains various icons for file operations and execution. The main content area displays the notebook's title '线性回归' (Linear Regression) and a paragraph of text describing a problem: '某城市的电网系统需要升级，以应对日益增长的用电需求。电网系统需要考虑最高温度对城市的峰值用电量的影响。项目负责人需要预测明天城市的峰值用电量，他搜集了以往的数据。现在，负责人提供了他搜集到的数据，并请求你帮他训练出一个模型，这个模型能够很好地预测明天城市的峰值用电量。' Below the text is a section header '1- 准备' (1- Preparation) and a sub-header '先导入必要的python包' (First import necessary python packages). A code cell is shown with the following code:

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import time
%matplotlib inline
```

在 Regression.ipynb 中 有许多任务, 每个任务需要实现相应代码。有 `#### START CODE HERE ####` 的标记说明这里是要填写代码 , `#### END CODE HERE ####`说明到这里终止。一般情况下只需要填写在 **START CODE HERE** 里有 **None** 的代码。例如

```
#### START CODE HERE ####  
  
h_theta = None  
loss = None  
  
#### END CODE HERE ####
```

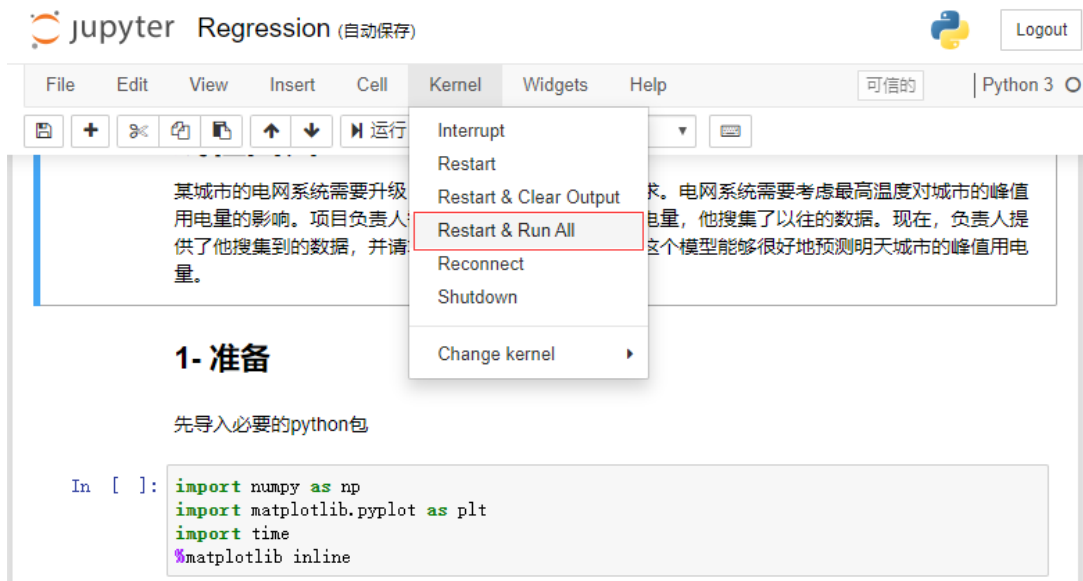
填写完代码并运行这个任务, 会有一些结果输出, 比如

```
data = np.loadtxt('data.txt')  
#data 第一列为温度信息 第二列为人口信息  
X = data[:,0].reshape(-1,1)  
#data 第三列为用电量信息  
Y = data[:,2].reshape(-1,1)  
plt.xlabel('High temperature')  
plt.ylabel('Peak demand')  
plt.scatter(X,Y)  
print('X shape:',X.shape)  
print('Y shape:',Y.shape)  
print('some X:',X[:5])  
print('some Y:',Y[:5])
```

```
X shape: (80, 1)  
Y shape: (80, 1)  
some X: [[38.79]  
[37.53]  
[32.93]  
[25.82]  
[20.89]]
```

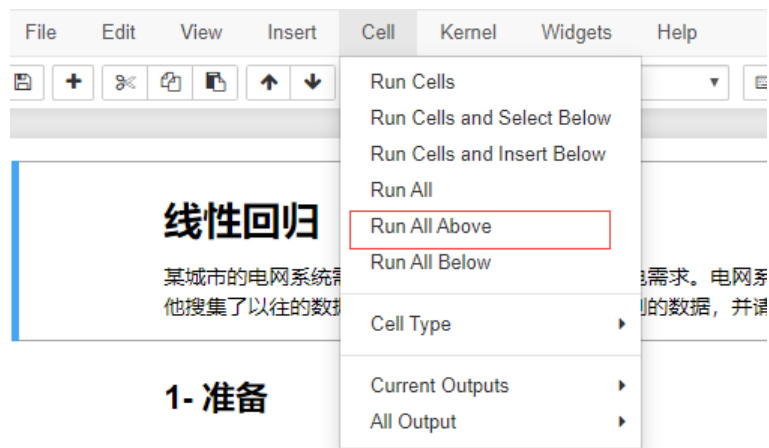
在 jupyter 中每个分隔的区域称为 cell, 一个 cell 可以是 python 代码块, 或者是文字说明块。运行 cell 的方式可以是单击页面头部的菜单栏的运行按钮, 或者使用快捷键“shift+enter”。实验的任务可能包含多个代码 cell 和文字说明 cell, 所以运行任务可能要按照顺序运行多个 cell, 然后才能看到输出结果, 具体输出位置请看输出代码(例如“print”)所在位置。(注意, 新打开的.ipynb 文件要从第一个 cell 开始运行)

做完实验后, 最好重新运行一遍 (Restart & Run All) (如下图的操作)。



注意事项:

因为 jupyter notebook 中的所有 cell 都共享一个 python 环境，同一个变量可能在某个 cell 下面其他 cell 中被重新赋值，那么，在对这个 cell 上面的所有 cell 可能会改变那个变量。建议按顺序从上往下运行。如果变量内容被弄乱了的话，可以先点击某个 cell，然后选择 Run All Above。



3. 完成实验任务

任务 1 在 X 前面加一列 1

```
new X shape: (80, 2)
Y shape: (80, 1)
new X[:10,:]= [[ 1.   38.79]
 [ 1.   37.53]
 [ 1.   32.93]
```

```
[ 1.  25.82]
[ 1.  20.89]]
```

任务 2 初始化参数向量

```
theta shape is (2, 1)
theta = [[0.5488135 ]
         [0.71518937]]
```

任务 3 实现计算损失函数 J

```
first_loss = 145.4723573288773
```

任务 4 计算参数 θ 的梯度

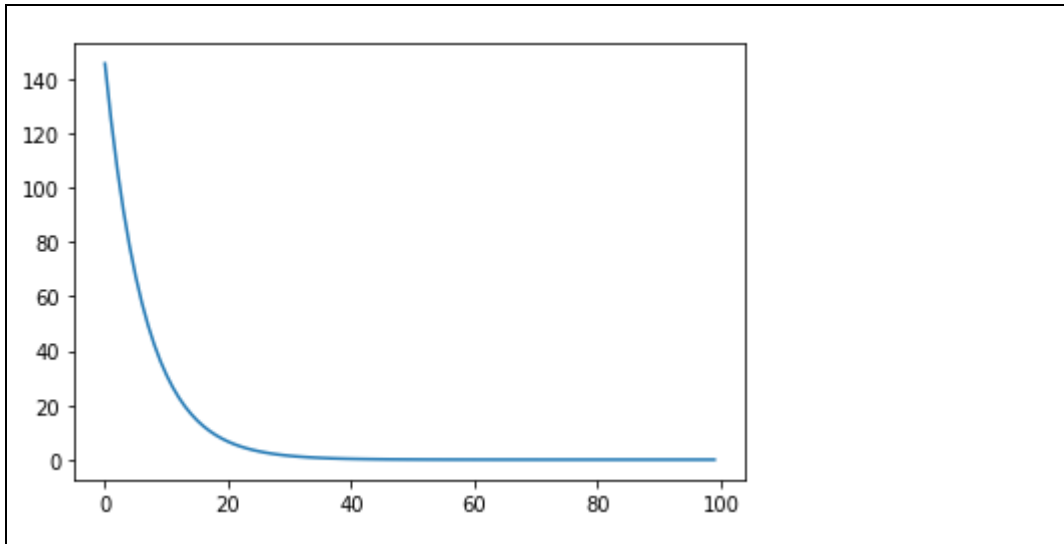
```
gradients_first shape : (2, 1)
gradients_first = [[ 16.10362006]
                  [464.4922825 ]]
```

任务 5 用梯度下降法更新参数 θ

```
theta_one_iter = [[ 0.3877773 ]
                  [-3.92973346]]
```

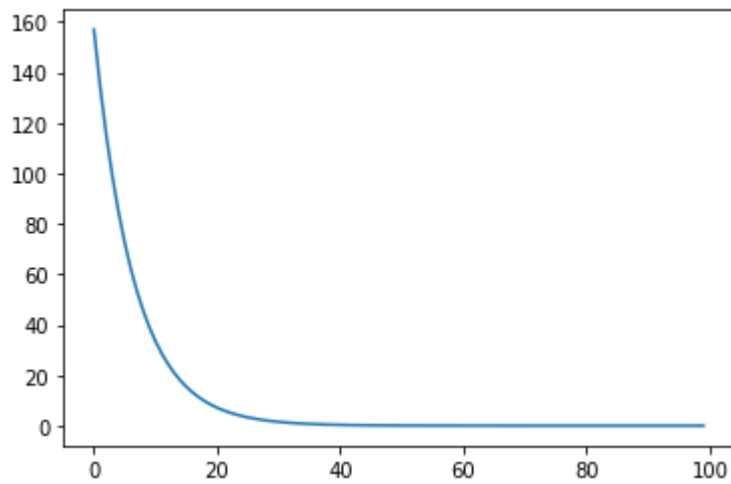
任务 6: 将前面定义的函数整合起来, 实现完整的模型训练函数。

```
theta = [[0.52741588]
         [0.09023805]]
loss = 0.0930181186193844
```



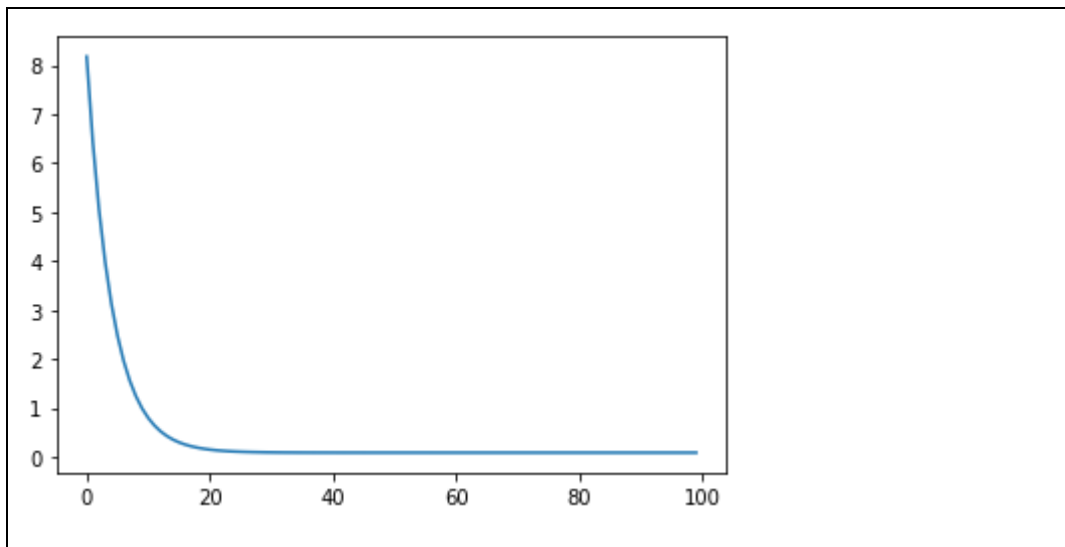
任务 7： 训练一个多变量回归模型。

```
theta = [[0.526022 ]  
         [0.06720419]  
         [0.57591482]]  
loss = 0.10571180408810799
```



任务 8： 对数据进行零均值单位方差归一化处理

```
mu = [25.77175  1.1355 ]  
sigma = [8.82317046 0.35648247]  
theta = [[2.87687827]  
         [0.69766231]  
         [0.03497325]]  
loss = 0.08778900945492227
```



任务 9： 实现正规方程

```
theta = [[2.876875 ]  
         [0.69769608]  
         [0.0349138 ]]
```

任务 10： 预计明天的峰值用电量

预计明天的峰值用电量为： 4.21 GW

任务 11： 训练一个多项式模型

