

Lab1 实验环境搭建

1、实验目的

Windows 下搭建上机实验环境，主要包括 Python, Jupyter Notebooks, pandas, Tensorflow。

2、实验简介

Windows 下使用 Anaconda 发行版来同时安装 Python 和 Jupyter Notebooks, 并包含了数据科学和机器学习社区中常用的许多软件包, 例如, Numpy、Scipy、Matplotlib、Pandas、Scikit-Learn 等 180 多个科学包及其依赖项。Tensorflow 等第三方库需要单独安装。

3、安装环境

安装 Anaconda

根据操作系统，下载相应的 anaconda 版本：

<https://www.anaconda.com/download/>

安装 anaconda: 双击安装文件，进行安装，全部默认选项：

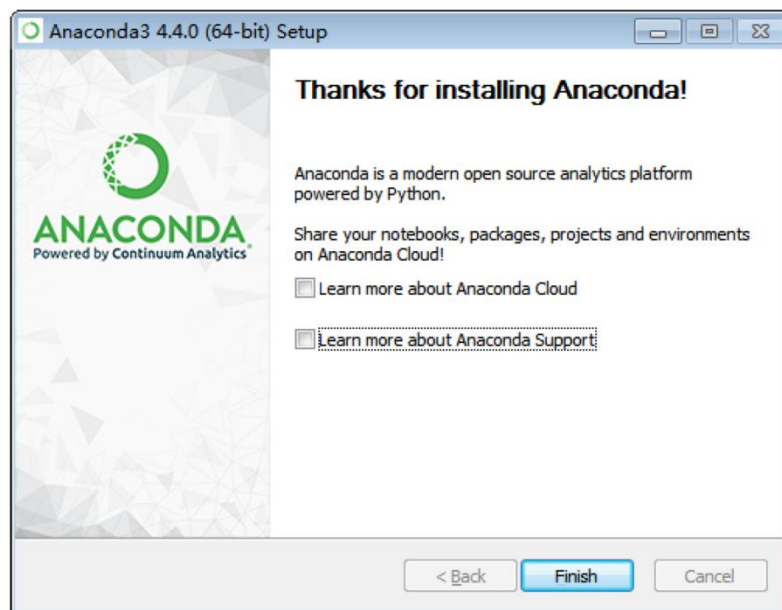


图 1 Anaconda 安装界面

正常启动 Anaconda:

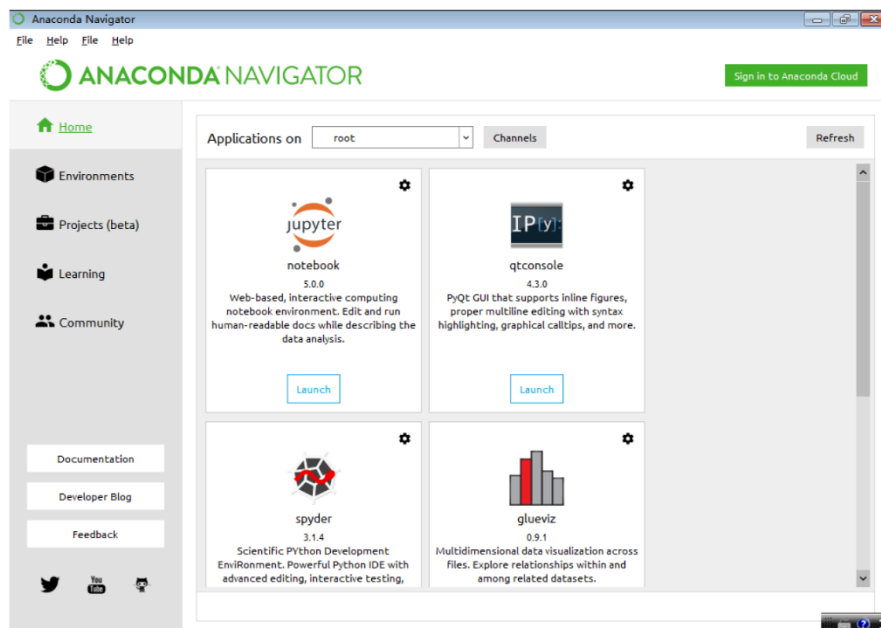


图 2 Anaconda Navigator

安装 Tensorflow

安装 Tensorflow（确保 path 中有相应的环境变量，如 C:\Users\Anaconda3\Scripts）

在命令窗口 cmd 中，执行：

```
> python -m pip install --upgrade pip
```

```
> pip install tensorflow
```

若不成功，则根据出现的问题尝试其他方法。

4、Jupyter notebook 操作

Anaconda 自带了 Jupyter Notebook。Jupyter Notebook 是功能强大的 Python 交互 IDE，又称 IPython notebook，是一个交互式展现笔记本，支持运行超过 40 种编程语言，能重现整个分析过程，并将说明文字、代码、图表、公式、结论整合在一个文档中。

4.1 运行 Jupyter notebook

安装 Anaconda 后，可以在其 Navigator 图形界面中点击启动 Notebook，也可以在终端命令窗口中输入如下命令启动：jupyter notebook

这时默认 Web 浏览器被自动打开，浏览器会自动开启一个页面显示用户当前文件目录，即 Jupyter notebook 的初始界面，如图 3 所示。

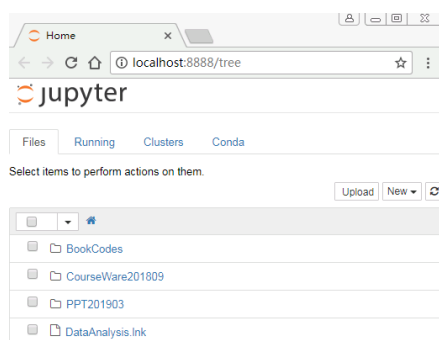


图3 Jupyter notebook 界面

4.2 创建 Notebook 文件

启动 Jupyter notebook 之后，首先要创建一个 Notebook 文件，通过点击右上角的 new 按钮，在出现的菜单中选择 Python3，如图 4 所示。

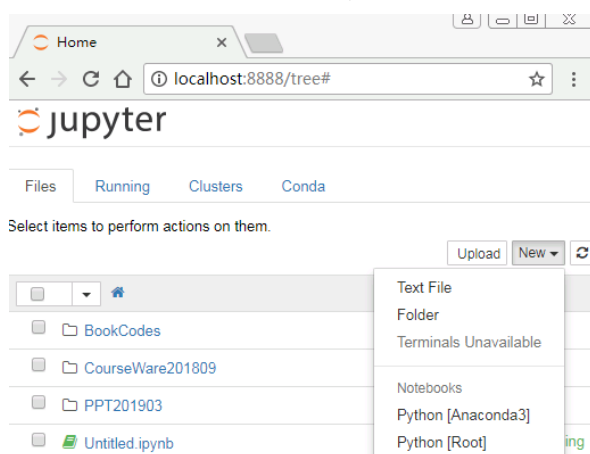


图4 在 Jupyter notebook 中新建文档

之后 Jupyter notebook 会自动打开新建文档，并出现一个空白的单元格 (Cell)，如图 5 所示。

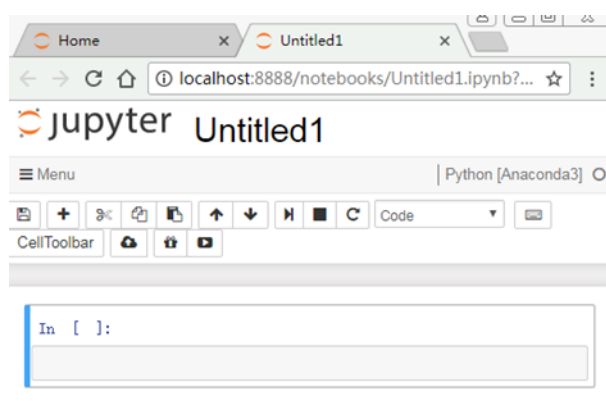


图5 Jupyter notebook 中的单元格

试着在空白单元格中输入如下代码： `print('hello python')`

按下 `Shift + Enter` 键，运行结果直接出现在单元格下方，并且在下面有建立了一个新的单元格，如图 6 所示。



图 6 使用 Jupyter notebook 打印 “hello python”

当按下 **Ctrl + Enter** 键运行代码则不会进入下一个单元格。

现将这个文档重新命名为 “hello python”：在 Jupyter notebook 的 File 菜单中找到 **Rename** 选项，如图 7 所示。之后在弹出的对话框中输入新的名称 “hello python”，单击 **Rename** 确认按钮，则完成重命名操作。

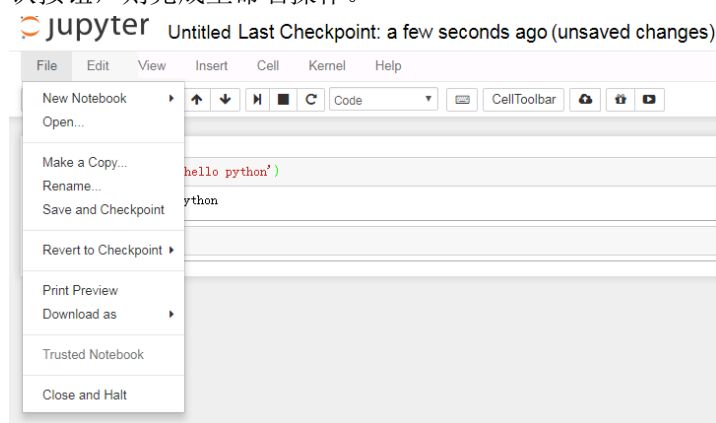


图 7 对文档进行命名操作

由于 Jupyter notebook 会自动保存文档，此时已经可以在初始界面看到新建的 “hello python.ipynb” 文件了，如图 8 所示。

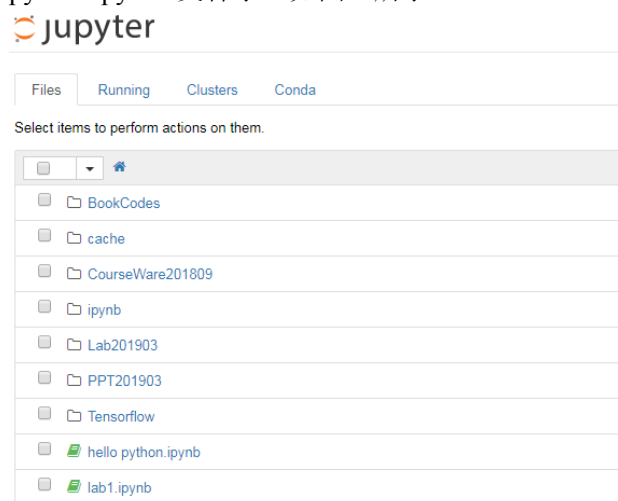


图 8 新建的 “hello python” 文档

4.3 Cell 操作

Notebook 文档（.ipynb 文件）是由一系列单元（Cell）构成，主要有两种形式的单元：

- **代码单元：**编写代码，通过按 **Shift + Enter** 运行代码，其结果显示在本单元下方。代码单元左边有 **In [1]:** 这样的序列标记，方便人们查看代码的执行次序。
- **Markdown 单元：**对文本进行编辑，采用 markdown 的语法规则，可以设置文本格式

式、插入链接、图片甚至数学公式。同样使用 `Shift + Enter` 运行 markdown 单元来显示格式化的文本。

创建 “hello tensorflow.ipynb ”，包含如下代码：

```
#利用 tensorflow 打印"hello, tensorflow!"
import tensorflow as tf
hello = tf.constant("hello, tensorflow!")
sess = tf.Session()
print(sess.run(hello))
```

按 `Shift + Enter` 键运行。

熟悉如下 cell 操作：

- 如果想删除一个 cell，先选中 cell，然后点击 Edit -> Delete Cell；
- 如果想移动一个 cell，点击 Edit -> Move cell [up | down]；
- 如果想剪贴、粘贴 cell，点击 Edit -> Cut Cell，然后再点击 Edit -> Paste Cell [Above | Below]；
- 如果有许多 cells，而只需要执行一次，或者想一次性执行大段代码，可以选择合并多个 cells，点击 Edit -> Merge Cell [Above | below]。
- 执行当前 cell，并自动跳到下一个 cell：Shift + Enter
- 当前的 cell 进入编辑模式：Enter
- 退出当前 cell 的编辑模式：Esc

4.4 Jupyter Notebook 导入代码

4.4.1 将本地.py 文件 load 到一个 cell

使用 new button 可以创建一个可编辑 python 的 test1.ipynb 文件。打开这个文件可以使用 Jupyter Notebook 进行编程，并且可以利用 Markdown 语法在程序单元(cell) 之间加入注解。

如需要将 test.py 文件载入到 jupyter 的一个 cell，其内容如下：

```
import tensorflow as tf
hello = tf.constant("hello, tensorflow!")
sess = tf.Session()
print(sess.run(hello))
```

方法步骤：

(1) 在需要导入该段代码的 cell 中输入

`%load test.py` #test.py 是当前路径下的一个 python 文件，若没有可以创建

(2) 运行该 cell

利用快捷键“Shift+Enter”

(3) 运行后，看到，`%load test.py` 被自动加入了注释符号#，test.py 中的所有代码都被 load 到了当前的 cell 中

如何从网络 load 代码到 jupyter？在 cell 中输入`%load http://.....`，然后运行该 cell，就会将 load 后面所对应地址的代码 load 到当前的 cell 中。

4.4.2 Jupyter 运行 python 文件

利用 jupyter 的 cell 可以运行 python 文件，即在 cell 中运行如下代码：

```
%run test.py
```

test.py 为要运行的 python 程序，结果会显示在该 cell 中

5、Jupyter Notebook 进阶（扩展）

5.1 使用 Magics

Jupyter Notebook 有预定义的 magic functions，使工作更方便和更具交互性。可以运行以下命令来查看这些函数的列表：`%lsmagic`

看到许多选项，有相对熟悉的 `% matplotlib inline` 函数。

magic 有两种运行方式：

- 逐行运行（Line-wise）：给定的命令都必须以 `%` 字符开始
- 逐块运行（Cell-wise）：给定的命令都必须以 `%%` 开头

操作练习常用的 Magics：

- 目录管理：查看当前工作目录 `%pwd`，更改工作目录 `%cd`，查看目录文件列表 `%ls`
- 文件管理：写入文件 `%%writefile`，运行脚本 `%run`，加载文件 `%load`
- 变量管理：查看当前变量 `%whos`，清除变量 `%reset`
- 测算运行时间：`%time`，`%%timeit`

<http://ipython.readthedocs.io/en/stable/interactive/magics.html>

5.2 数学公式编辑

在 Jupyter Notebook 的 markdown 单元中可以使用 LaTeX 的语法来插入数学公式。

- 行中公式(放在文中与其它文字混编)可以用如下方法表示：`$ 数学公式 $`
- 独立公式可以用如下方法表示：`$$ 数学公式 $$`。比如下面公式表示：
$$\frac{x}{y}$$
`$$ \frac{x}{y} $$`

5.3 幻灯片制作

Jupyter Notebook 擅长展示数据分析过程，除了通过网页形式分享外，也可以将其制作成幻灯片的形式。首先在 notebook 的菜单栏，选择 **View > Cell Toolbar > Slideshow**，这时在文档的每个单元右上角显示了 **Slide Type** 的选项。通过设置不同的类型，来控制幻灯片格式。有如下 5 种类型：

Slide：主页面，通过按左右方向键进行切换。

Sub-Slide：副页面，通过按上下方向键进行切换。

Fragment：一开始是隐藏的，按空格键或方向键后显示，实现动态效果。

Skip：在幻灯片中不显示的单元。

Notes：作为演讲者的备忘录，也不在幻灯片中显示。

幻灯片设置：

当编写好幻灯片形式的 notebook，使用 `nbconvert`：

`jupyter nbconvert notebook.ipynb --to slides --post serve` 演示。

在命令行中敲入上述代码后，浏览器会自动打开相应的幻灯片。

6、作业

(1) 探索如下操作的快捷方式？

- 为当前的 cell 加入 line number:
- 将当前的 cell 转化为具有一级标题的 markdown:
- 将当前的 cell 转化为具有二级标题的 markdown:
- 将当前的 cell 转化为具有三级标题的 markdown:
- 为一行或者多行添加/取消注释: `ctrl + ?`
- 撤销对某个 cell 的删除:
- 浏览器的各个 Tab 之间切换: `Ctrl PgUp` 和 `Ctrl PgDn`
- 快速跳转到首个 cell:
- 快速跳转到最后一个 cell:
- 查询函数的注解: `shift + tab`:

(2) 操作比较 `%time range(10)` 和 `%%timeit range(10)`。

(3) 如何在 notebook 中输入如下公式，并显示：

$$L(a,b) = \sum_{i=1}^n (y_i - (ax_i + b))^2$$

(4) 学习编写幻灯片形式的 notebook，使用 `nbconvert`。