

Lab1 Jupyter notebook 实践

实验简介: Anaconda 内置 Jupyter Notebook，它是一款功能强大的 Python 交互 IDE，又称 IPython notebook，支持运行超过 40 种编程语言，能够重现整个分析过程，并能将说明文字、代码、图表、公式、结论整合在一个文档中。

作业要求: 仅提交代码文件（代码格式为“xxx.ipynb”，关键语句和逻辑要有注释），且代码需要运行结果。

任务描述:

1、运行 Jupyter notebook

安装 Anaconda 后，可以在其 Navigator 图形界面中点击启动 Notebook 或在终端命令窗口中输入如下命令启动：jupyter notebook，默认 Web 浏览器被自动打开，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。

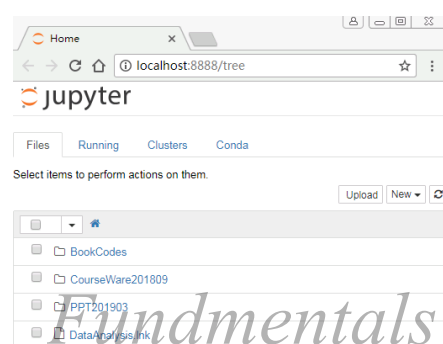


图 3 Jupyter notebook 界面

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。

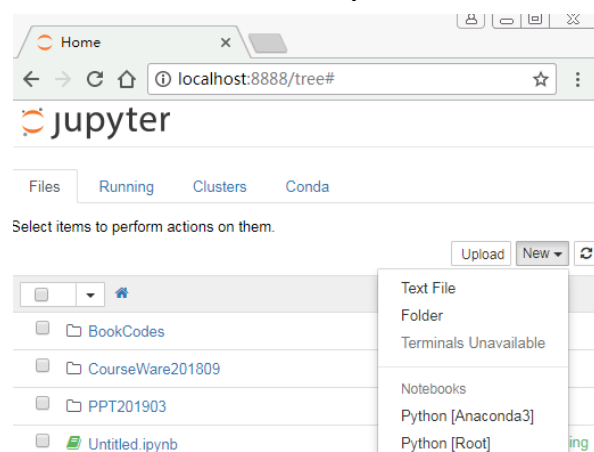


图 4 在 Jupyter notebook 中新建文档

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。

Fundamentals of Machine Learning

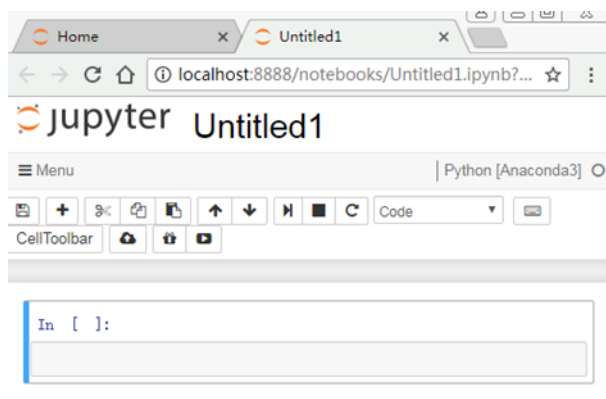


图 5 Jupyter notebook 中的单元格

试着在空白单元格中输入代码：`print('hello python')`，按下 **Shift + Enter** 键，运行结果直接出现在单元格下方，并且在下面又建立了一个新的单元格，如图 6 所示。

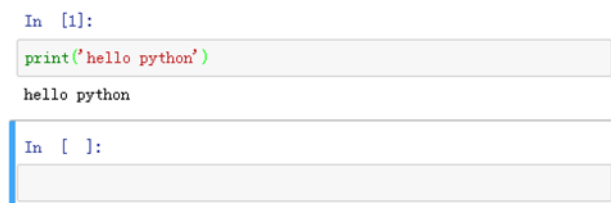


图 6 使用 Jupyter notebook 打印“hello python”

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。

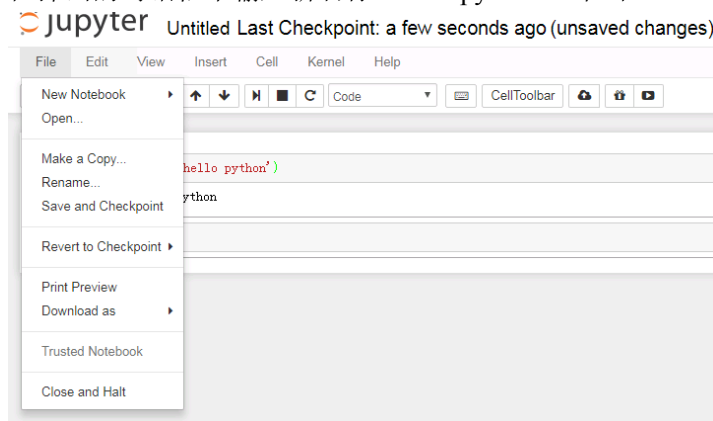


图 7 对文档进行命名操作

由于 Jupyter notebook 会自动保存文档，此时可在初始界面看到新建的“hello python.ipynb”文件，如图 8 所示。

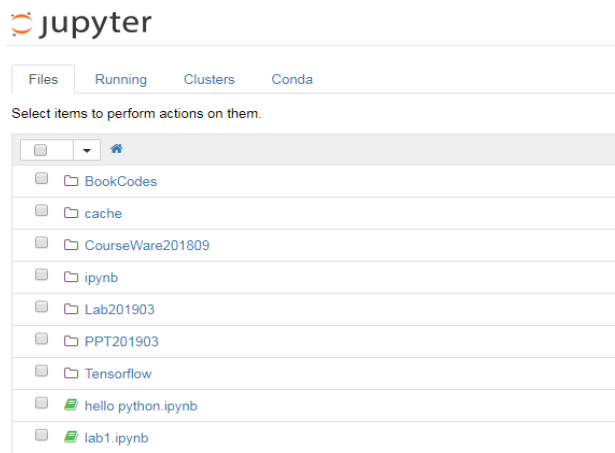


图 8 新建的“hello python”文档

3、Cell 操作

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

Code: 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 [序号]**：这样的序列标记，方便人们查看代码的执行次序。

Markdown: 对文本进行编辑，采用 markdown 的语法规则，可以设置文本格式、插入链接、图片甚至数学公式。同样使用 **Shift + Enter** 运行 markdown 单元显示格式化的文本。

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：

- 删除一个 cell，先选中 cell，然后点击 **Edit -> Delete Cell**；
- 移动一个 cell，点击 **Edit -> Move cell [up | down]**；
- 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**

熟悉如下操作的快捷方式：

按 **enter** 进入编辑模式；按 **esc** 进入命令模式。

- 当前的 cell 加入 line number：命令模式下 **L**
- 当前的 cell 转化为具有一级标题的 markdown：命令模式下 **1**

- 当前的 cell 转化为具有二级标题的 markdown: 命令模式下 2
- 当前的 cell 转化为具有三级标题的 markdown: 命令模式下 3
- 为一行或者多行添加/取消注释: 命令模式下 Ctrl + /
- 撤销对某个 cell 的删除: 命令模式下 z
- 浏览器的各个 tab 之间切换: Ctrl + PgUp 和 Ctrl + PgDn
- 快速跳转到首个 cell: Ctrl+home
- 快速跳转到最后一个 cell: Ctrl + End
- 查询函数的注解: Shift + Tab

4、Jupyter Notebook 导入代码

4.1 导入本地.py 文件

使用 new button 可以创建一个可编辑 python 的 test.ipynb 文件。打开此文件使用 Jupyter Notebook 进行编程, 并且可以利用 Markdown 语法在单元(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
- (2) 按 “Shift + Enter” 运行 cell
- (3) 运行后, %load test.py 被自动加入了注释符号 “#”, test.py 中的所有代码都被 load 入到当前 cell 中

从网络 load 代码到 jupyter: 在当前 cell 中输入 %load <http://.....>, 然后运行该 cell。

4.2 Jupyter 运行 python 文件

利用 jupyter 的 cell 可运行 python 文件, 即在 cell 中运行代码: %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

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。查看 magics 函数列表, 运行命令: %lsmagic, 结果中包含我们熟悉的 %matplotlib inline 函数。

magic 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
- 运行时间：%time，%%timeit

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：
$$\frac{x}{y}$$
 \$\$

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。有五种类型：

- 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：作为演讲者的备忘笔记，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，在命令窗口使用命令演示：

> jupyter nbconvert notebookname.ipynb --to slides --post serve

```
(tensorflow) G:\Anaconda3>jupyter nbconvert ppt.ipynb --to slides --post serve
[NbConvertApp] Converting notebook ppt.ipynb to slides
[NbConvertApp] Writing 250882 bytes to ppt.slides.html
[NbConvertApp] Redirecting reveal.js requests to https://cdn.jsdelivr.net/gh/revealjs/reveal.js@4.0.0
Serving your slides at http://127.0.0.1:8000/ppt.slides.html
Use Control-C to stop this server
WARNING:tornado.access:404 GET /custom.css (127.0.0.1) 0.99ms
WARNING:tornado.access:404 GET /custom.css (127.0.0.1) 0.00ms
WARNING:tornado.access:404 GET /favicon.ico (127.0.0.1) 1.00ms
```

浏览器会自动打开相应的幻灯片进行演示。

6、操作习题

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

(2) 在 notebook 文档中，插入 notebook 链接：

链接格式：[title](URL)

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

(3) 任意截取一张图片，插入 notebook 文档中；

图片格式：![title](URL)

(4) 如何在 notebook 中输入如下公式，并显示：

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

(5) 将编写的 notebook 编写成幻灯片形式，并进行演示。

参考链接：<https://www.jianshu.com/p/f093416d0db0>

Wangbianqin, Public Laboratory & Teaching Center, Guangzhou East Campus, Sun Yat-sen University