Python与深度学习基础

张越一　zhyuey@ustc.edu.cn

大作业

- 与网络相关，使用API
- 把结果用适当方式展示出来 (GUI界面，网页）
- 上交内容：github/gitee库的公开地址，报告，小视频
- Deadline: 2019年3月31日23：59分
- 上交方式：发给两位助教，分配方式待定
- 参考API：
  - 空气质量
    - http://pm25.in/api_doc
  - 百度
    - http://ai.baidu.com/docs/#/
  - 聚合API
    - https://www.juhe.cn/docs
  - 地震
    - https://www.programmableweb.com/category/earthquakes/api

- 其他要求：
- 至少要设计一个类
- 至少要用两个文件

- 代码要加注释

- 学号分配

关于上次的小作业

○ Pascal's Triangle

○ Fraction Class

○ Go to see jupyter notebook

About Image

- 2D, 3D and more
- CMOS CCD
- Format: BMP, JPG, PNG, TIFF
- Depth: 8Bit, 16Bit
- Gray, Color, Alpha
- RGB, BGR, YVR
- Compression?

- Resize, Rotation
- Filter

- PIL, pillow
- OpenCV
- Matplotlib

PIL (pillow)

- PIL: Python Image Library
- only support 2.X
- Fredrik Lundh and contributors

- pillow: a friendly fork of PIL
- Alex Clark and contributors
- conda install pillow
- pip install pillow

- Easy to use Imaging library

Numpy

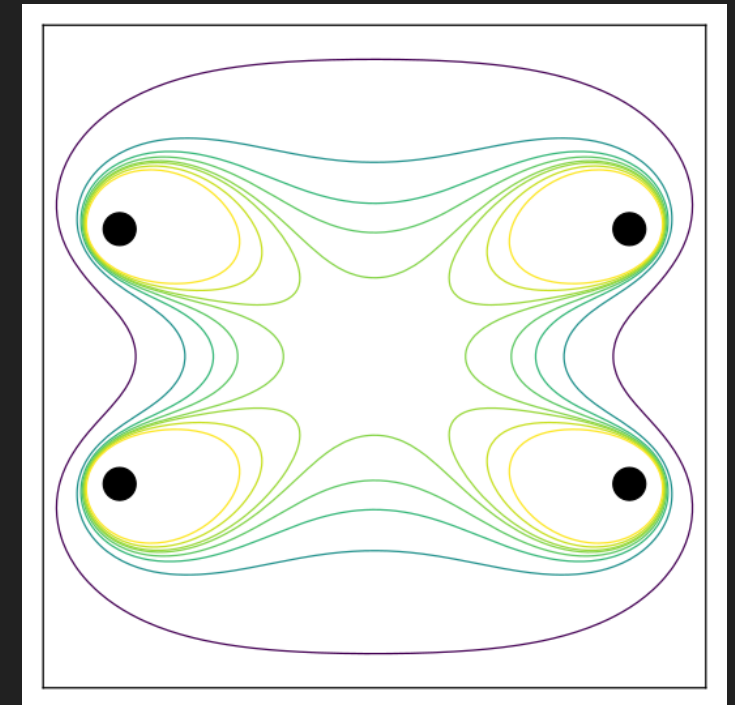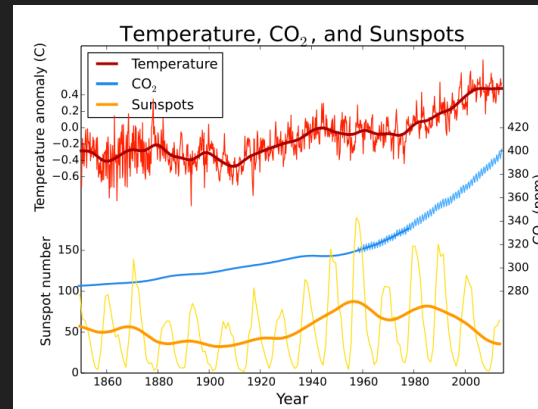- Numpy是python语言的一个扩充程序库
- 支持高端大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库
- 现已成为机器学习的必备模块
- https://docs.scipy.org/doc/numpy-1.15.1/user/quickstart.html

Matplotlib

- **matplotlib** 是Python编程语言及其数值数学扩展包 NumPy的可视化操作界面。它为利用通用的图形用户界面工具包，如Tkinter, wxPython, Qt或GTK+向应用程序嵌入式绘图提供了应用程序接口（API）。

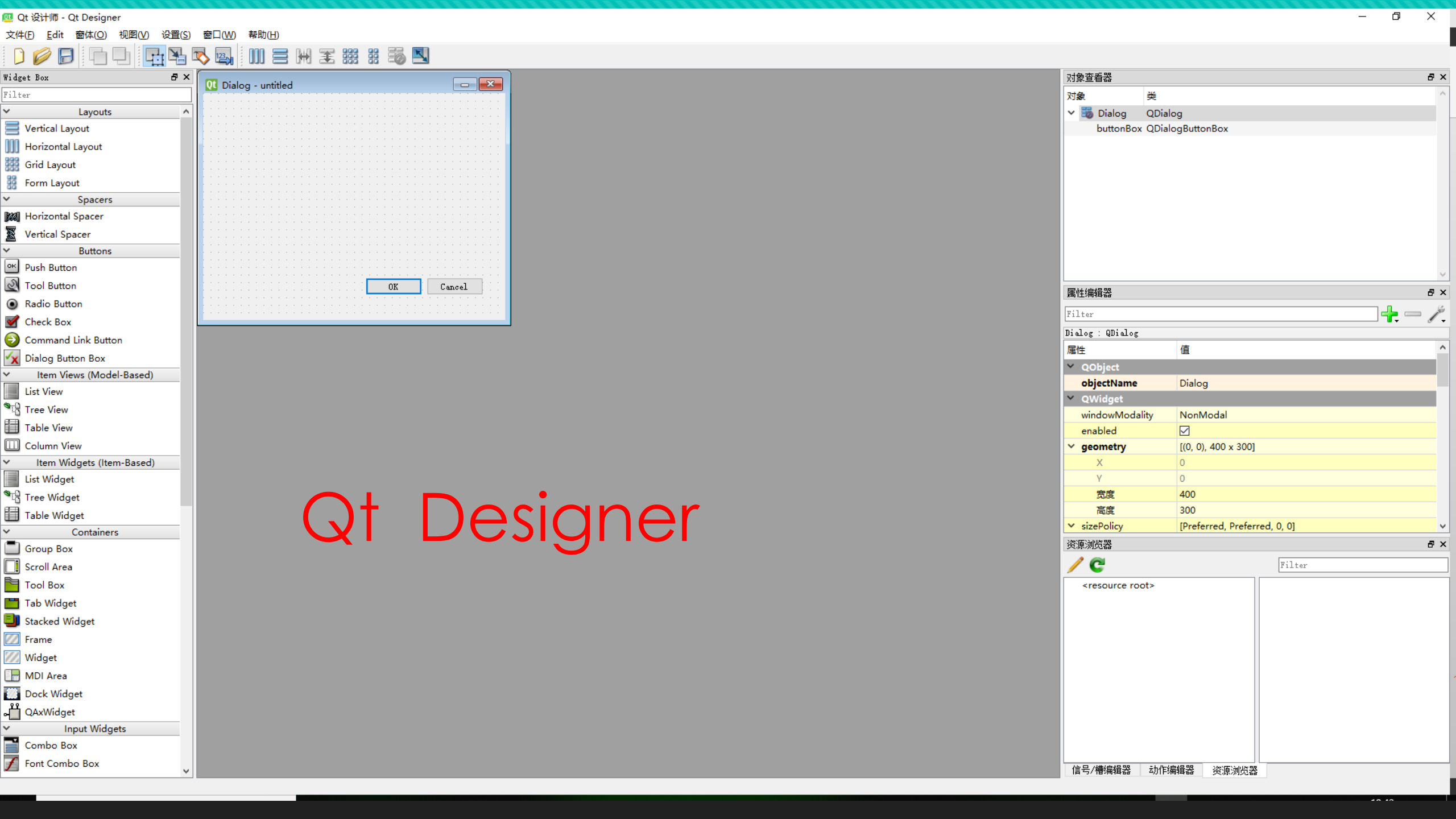- https://matplotlib.org/users/tutorials.html
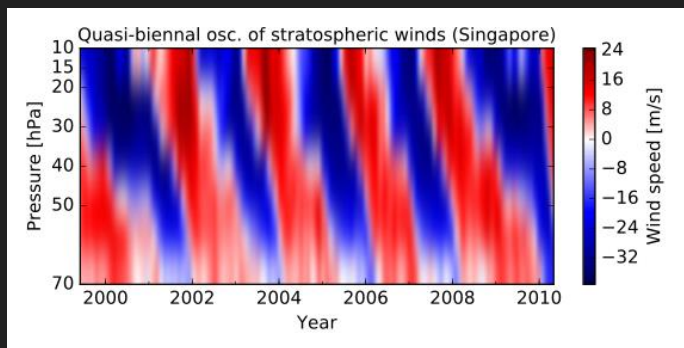
OpenCV

- **OpenCV**的全称是Open Source Computer Vision Library，是一个跨平台的计算机视觉库。
- OpenCV是由英特尔公司发起并参与开发，以BSD许可证授权发行，可以在商业和研究领域中免费使用。
- OpenCV可用于开发实时的图像处理、计算机视觉以及模式识别程序。该程序库也可以使用英特尔公司的IPP进行加速处理。
- 原生的OpenCV是用C\C++编写，但提供了python接口。
- https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html

About Qt

- a free and open-source widget toolkit for creating graphical user interfaces

- cross-platform applications that run on various software and hardware platforms

- Qt is currently being developed by The Qt Company, a publicly listed company

- the Qt Project under open-source governance, involving individual developers and organizations working to advance Qt

- Qt Creator and Qt Designer

- Developer:

- Trolltech (1991–2008)

- Nokia (2008–2011)

- Qt Project (2011–present)

- Digia (2012–2014)

- The Qt Company (2014–present)

文件(F)　Edit　窗体(O)　视图(V)　设置(S)　窗口(W)　帮助(H)

Widget Box

Filter

Layouts
Vertical Layout
Horizontal Layout
Grid Layout
Form Layout

Spacers
Horizontal Spacer
Vertical Spacer

Buttons
Push Button
Tool Button
Radio Button
Check Box
Command Link Button
Dialog Button Box

Item Views (Model-Based)
List View
Tree View
Table View
Column View

Item Widgets (Item-Based)
List Widget
Tree Widget
Table Widget

Containers
Group Box
Scroll Area
Tool Box
Tab Widget
Stacked Widget
Frame
Widget
MDI Area
Dock Widget
QAxWidget

Input Widgets
Combo Box
Font Combo Box

Dialog - untitled

OK　Cancel

Qt Designer

对象查看器

对象	类
Dialog	QDialog
buttonBox	QDialogButtonBox

属性编辑器

Filter

Dialog : QDialog

属性	值
QObject	
objectName	Dialog
QWidget	
windowModality	NonModal
enabled	☑
geometry	[(0, 0), 400 x 300]
X	0
Y	0
宽度	400
高度	300
sizePolicy	[Preferred, Preferred, 0, 0]

资源浏览器

Filter

\<resource root\>

信号/槽编辑器　动作编辑器　资源浏览器

PyQt

- PyQt is a Python binding of the cross-platform GUI toolkit Qt

- PyQt is free software developed by the British firm Riverbank Computing.

- PyQt supports Microsoft Windows as well as various flavours of UNIX, including Linux and MacOS (or Darwin).

- https://build-system.fman.io/pyqt5-tutorial

- http://zetcode.com/gui/pyqt5/

- https://likegeeks.com/pyqt5-tutorial/

Signal and Slot

- 信号和槽机制是QT的核心机制
- 信号和槽是一种高级接口，应用于对象之间的通信
- 当一个特定事件发生的时候，一个信号被发射
- Qt有很多预定义的信号
- 槽就是一个对应于特定信号的被调用的函数
- Qt有很多预定义的槽

widget.signal.connect(slot_method)

button.clicked.connect(self.slot_method)

```python
from PyQt5.QtWidgets import (QApplication, QComboBox, QDialog,
QDialogButtonBox, QFormLayout, QGridLayout, QGroupBox, QHBoxLayout,
QLabel, QLineEdit, QMenu, QMenuBar, QPushButton, QSpinBox, QTextEdit,
QVBoxLayout)

import sys

class Dialog(QDialog):

    def slot_method(self):
        print('slot method called.')

    def __init__(self):
        super(Dialog, self).__init__()

button=QPushButton("Click")
button.clicked.connect(self.slot_method)

mainLayout = QVBoxLayout()
mainLayout.addWidget(button)

self.setLayout(mainLayout)
self.setWindowTitle("Button Example - pythonspot.com")

if __name__ == '__main__':
    app = QApplication(sys.argv)
    dialog = Dialog()
    sys.exit(dialog.exec_())
```