TensorFlow2.0

Installation and Application

Shaokai Yang

yangs9uc@gmail.com

Outline

1. Environment Introduction
2. Python Basic
3. TensorFlow Basic
4. Deep Neural Network
5. Application Examples

1 Introduction to the development environment

Development Environment

- Python 3
- TensorFlow 2.0
- Anaconda 3

Python 3

- User friendly
- Rich libraries
- Powerful ecosystem

TensorFlow Versions

- An efficient open source framework for artificial intelligence
- 2015.11 TensorFlow 0.1
- 2017.02 TensorFlow 1.0.0
- Marking the birth of the stable version
- 2019.03 TensorFlow 2.0 Alpha
- 2019.06 TensorFlow 2.0 Beta
- 2019.10 TensorFlow 2.0 Official version

TensorFlow 2.0

The significance of TensoFlow 2.0 lies in : TensorFlow officially declares that it will focus on the usability permanently

It is simple, clear and easy to expand, which greatly reduces the threshold of deep learning programming

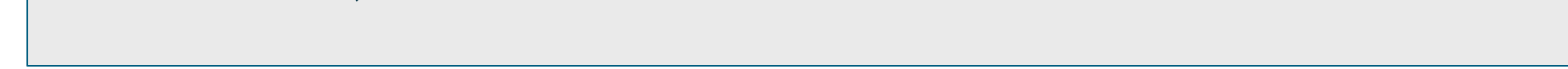
The official introduction video can be found in:

https://www.youtube.com/channel/UC0rqucBdTuFTjJiefW5t-IQ

Anaconda 3

- A python distribution for scientific computing
- Pre-installed Numpy, Matplotlib and other open-source packages and scientific computing tools
- Powerful package management and environment management
  - Package Management: Install, upgrade and uninstall toolkits;
    When the toolkit is installed,the corresponding dependent packages are automatically installed
  - Environment Management: Ability to create several independent virtual enviromments on the same machine;
    Ability to quarantine different versions of the toolkit that are needed for different projects
    Ability to quickly switch between different environments

- **Anaconda official website**



- **Anaconda Navigator**
  - The desktop interface for anaconda
  - Ability to launch applications to manage the conda environment and python module packages
- **Anaconda Navigator**
  - The command execution terminal for anaconda
- **Jupyter notebook**
  - A web-based interactive program editing tool
- **Spyder**
  - An integrated a development environment

1.2 Anaconda Installation

- **VS Code:** Lightweight yet powerful code editor
- **Jupyter Lab:** An extension of Jupyter Notebook, provides a better user experience
- **Glue:** Python repository for exploring relationships between mutual data sets
- **Qt Console:**
  - Graphical interface for IPython
  - Ability to directly display the code generated graphics as well as built-in many useful functions and functions
- **Orang3:** Interactive data visualization software
- **R Studio:** An additive development environment for the R language

Python Operating Mode — Interactive Mode

```
shaokai@shaokai-Yang:~$ python3
Python 3.8.2 (default, Apr 27 2020, 15:53:34)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello World')
Hello World
>>>
```

- Open a Terminal
- Type Python3, to activate Python 3 interactive mode;
  The Python prompt appears >>>
- At prompt >>>, input Python statement
- **Return to gets the result of the Python statement**

Python Operating mode——Script Mode

- Open the editor, create a python file
- Write Python statement , save the script
- Run the script in a terminal by python3 + file name to get the results

print() Function

Print (Expression / String)

```
>>> print('Hello World')
Hello World
>>> print(11*11)
121
>>> print(111*111)
12321
>>> print(1111*1111)
1234321
```

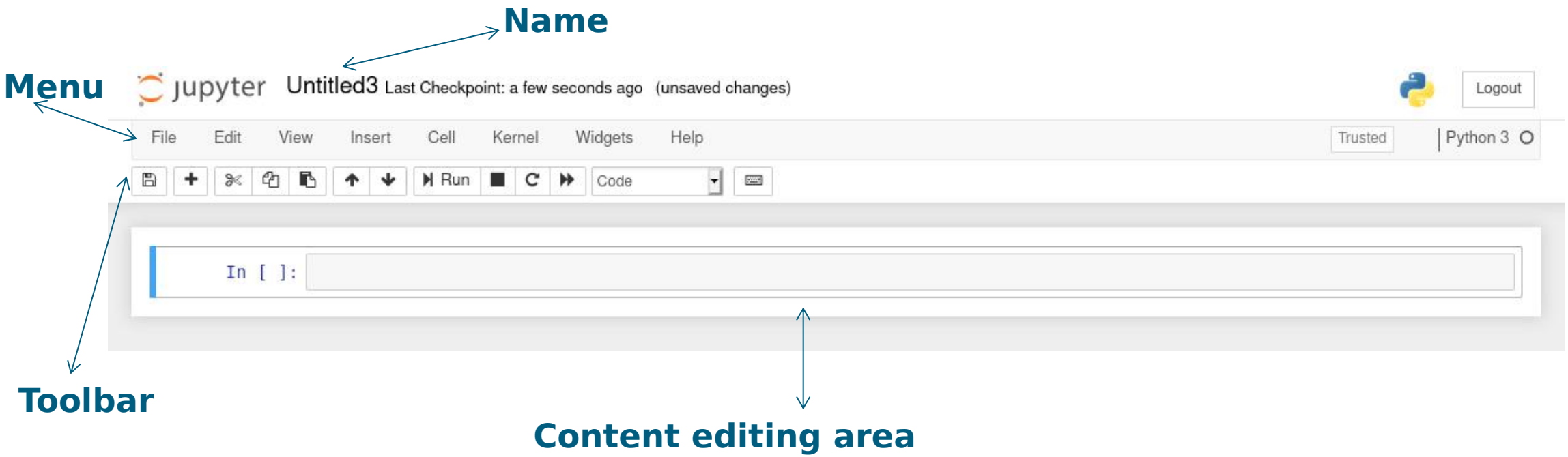
Exit the Python interactive

- At the Python command prompt,enter exit() or quit();
- You can also type ctrl+Z to stop;

Jupyter Notebook

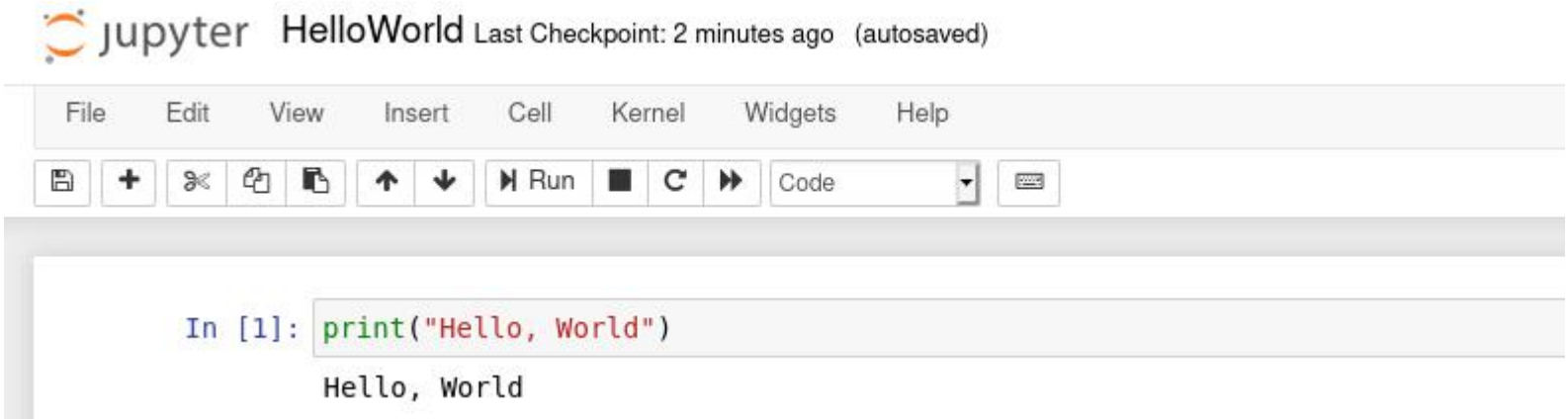
- An interactive tool for editing programs accessed from the web
- The invention is convenient for the remote access of the program, shields the difference between different systems, and is convenient for the code sharing
- Ability to integrate the codes, images, comments, graphics and operating results in the same document in order to prapare a beautiful interactive document

Jupyter Notebook Interface



In Jupyter Notebook, the basic unit is the **Cell**, the content-editing area in the above graph

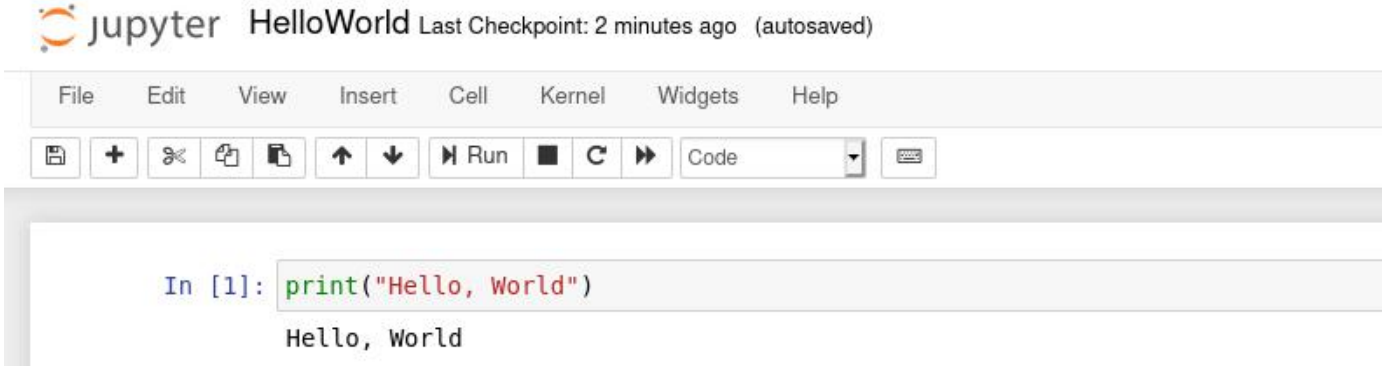
Edit the code



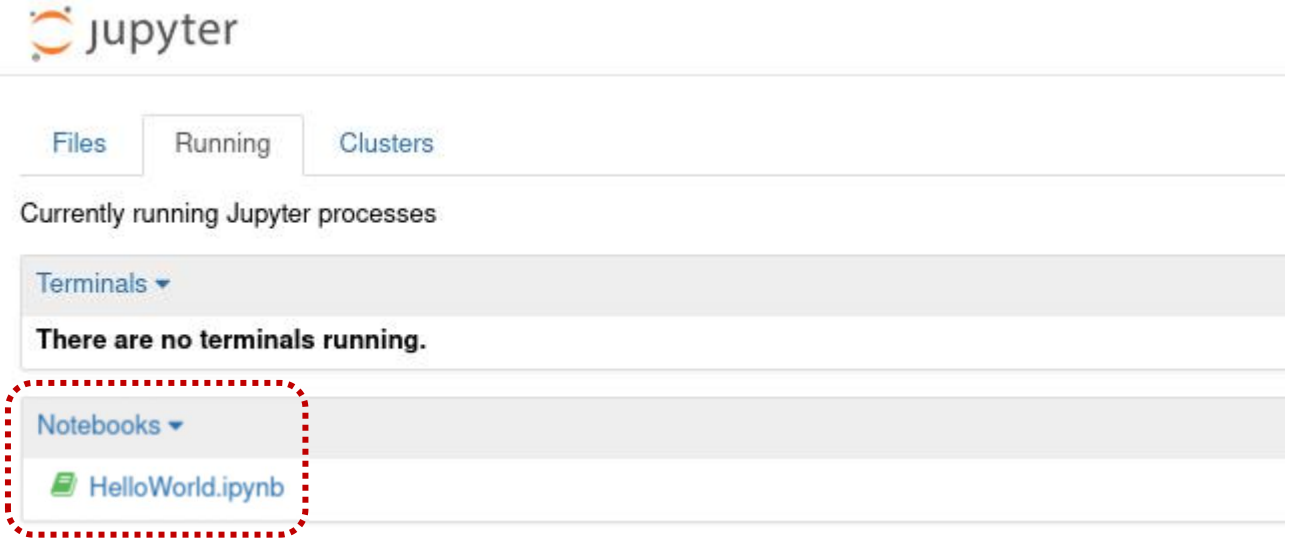
a) Modify the file name in the name field
b) In the content editing area enter the code
c) Click on the Run button to get the results

After saved the HelloWorld.ipynb, a file is generated in the working directory and appears in the file list of the home page

- Edit the code



- Save the file



Package Managenment: install, uninstall, update, etc

conda command -- installation

```
conda install <libraries' names>
```

Specify the packages to install:

```
conda install numpy
```

Installing multiple packages at once:

```
conda install numpy matplotlib
```

Specifies the version installed:

```
conda install numpy=1.17.4
conda install numpy==1.17
```

When using conda installation, the corresponding dependencies can be automatically installed

conda command

Installing package	conda install <Library's name>
Uninstall package	conda remove <Library's name>
Updating package	conda update <Library's names>
Searching for package	conda search <Library's name>
Checking all packages	conda list
Upgrading package versions	conda update<Library's name>

Pip command

- **Installing package** pip install <Libraries' names>
- **Uninstall package** pip uninstall <Libraries' names>

```
pip install numpy

pip install matplotlib

pip install tensorflow==2.0.0
```

Evironmental Management

Create the environment

when the full name is used for the argument, double minus signs are used in front

```
conda create env  - -name <Environment name><Libraries' names>

conda create - - name <Environment name><Libraries' names>

conda create env  - n <Environment name><Libraries' name>

conda create -n <Environment name><Tool package lists >
```

Library name

Libraries names

```
conda create - n py3 python=3.8  pandas
```

Activate the environment

Windows system activate <Environment name>

Linux system source activate <Environment name>

Exit the current environment

Windows system deactivate <Environment name>

Linux system source deactivate <Environment name>

Displays all evironments

View general information about the environments that are currently created by the system

```
conda env list
```

Delete evironment

```
conda env  remove- - name <Environment name>
conda env remove  - name <Environment name>--all

conda env remove  - n <Environment name>--all
conda remove-n  <Environment name>--all
```

Export evironment

When sharing the codes, it also needs to share the environment in which the code runs

```
conda env export > environment.yml
```

Environment configuration files

Configure the evironment

When you run someone else's codes, you need to configure the corresponding environment ,so you can use the other party's shared YAML document to creat the same operating environment

```
conda env  creat-f environment.yml
```

1.6 Install TensorFlow 2.0

- Creat the environment and activate it
  - conda create -n tf2 python=3.6
  - conda activate tf2
- Install the appropriate libraries
  - conda install jupyter
  - conda install scipy
  - conda install scikit-learn
  - conda install matplotlib
  - conda install pandas
- Install TensorFlow 2.0
  - pip install --upgrade tensorflow
- Test TensorFlow 2.0
  - import tensorflow as tf
  - tf.__version__