



Topic 1 安装

Vscode安装、Anaconda安装、 tensorflow环境配置、gpu配置* (Win10)

魏准 (eleweiz@zju.edu.cn)
浙江大学，信息与电子工程学院
2022

内容概述

本课程以**编程实践**和**工程项目训练**为主要目标，重点介绍人工智能中**分类与回归**、**支持向量机**、**主成分分析**、**多层神经网络**和**卷积神经网络**等基本算法，同时结合计算机视觉和机器学习领域，进行**实际的工程项目编程训练**。通过此课程学习，一方面能使**学生掌握基本的算法编程**，另一方面增加**学生实际工程项目的编程能力**。

要求：

考试形式：

（1）实验报告：规范撰写实验报告，依据实验报告内容及实验客观结果综合评定分数。实验报告占60%的分数。（**实验题目、实验结果及分析、附图**）

（2）设计实验答辩：针对最后的大实验“基于深度学习的大规模编程”，开展答辩环节。综合实验结果及答辩表现评定分数。设计实验答辩占40%分数。

参考教材：

- 1、机器学习， 周志华， 清华大学出版社， 2016
- 2、统计学习方法（第二版）， 李航， 清华大学出版社， 2019
- 3、学习OPENCV， G.BRADSKI， 清华大学出版社， 2012
4. 动手学深度学习PYTORCH版， Aston Zhang, 李沫等, 人民邮电出版社， 2020
- 5、机器学习实战， Peter Harrington, 人民邮电出版社， 2013.
- 6、Phil Kim, 2017. MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence.
- 7、机器学习实战-基于Scikit-Learn、Keras和TensorFlow, Aurelien Geron, 机械工业出版社

目录

CONTENTS

- 1 环境配置讲解
- 2 Vscode安装
- 3 Anaconda安装
- 4 Tensorflow环境搭建
- 5 (选) gpu配置

第一部分

环境配置讲解

步骤:

- ◆ 安装vscode
- ◆ 配置tensorflow(后记tf)相关:
 - 安装anaconda (方便的包管理)
 - 使用anaconda建立tf环境
 - 在tf环境中安装对应版本的python&tf以及相关的依赖包
 - 一些系统配置&vscode设置

一些讲解:

目标:

在vscode上, 使用keras进行深度学习代码编写。

Keras:

有易于使用的高级API的工具包, 需要靠'backend'完成实际的计算, 长期以来keras支持多种backends, 而tensorflow逐渐成为主流

Tensorflow:

19年发布的tensorflow2.0相比tensorflow1.0是全新的项目, 未来应该使用全新的2.0系列

Tensorflow2.0:

2.0版本keras成为了tensorflow官方API: tf.keras; 之后多backends的keras停止更新只进行错误维护, 未来应使用能利用tensorflow2.0新特性且持续更新的tf.keras; 不需要另外安装keras工具包。

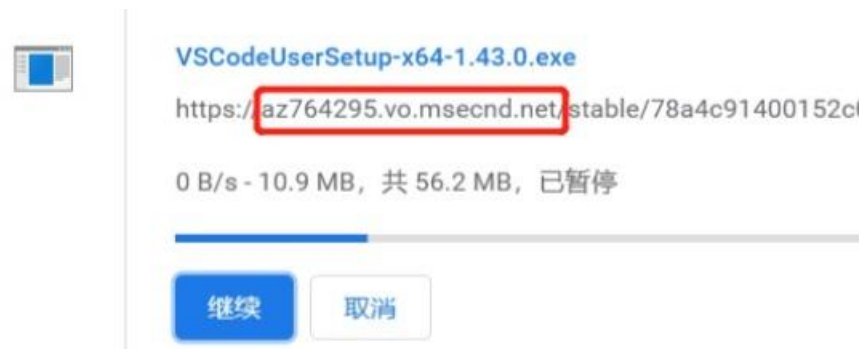
第二部分

Vscode安装

网址:

<https://code.visualstudio.com/Download>
[Download Visual Studio Code - Mac, Linux, Windows](#)

官网下载速度可能较慢，可能100M文件需要几个小时，还可能中断，可以下载时可以将地址替换为国内镜像源：



复制上图所示的下载链接到新页面，将红框部分替换为：vscode.cdn.azure.cn

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



↓ Windows

Windows 7, 8, 10, 11

| | | | |
|------------------|--------|--------|-----|
| User Installer | 64 bit | 32 bit | ARM |
| System Installer | 64 bit | 32 bit | ARM |
| .zip | 64 bit | 32 bit | ARM |



↓ .deb

Debian, Ubuntu

↓ .rpm

Red Hat, Fedora, SUSE

| | | | |
|---------|--------|-----|--------|
| .deb | 64 bit | ARM | ARM 64 |
| .rpm | 64 bit | ARM | ARM 64 |
| .tar.gz | 64 bit | ARM | ARM 64 |

Snap Store



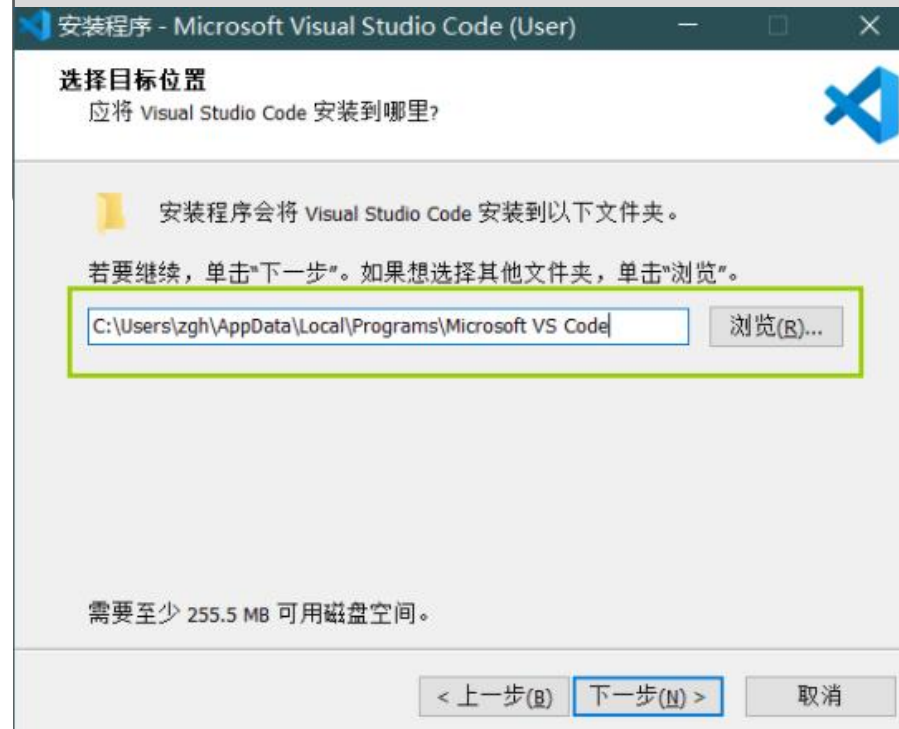
↓ Mac

macOS 10.11+

.zip Universal Intel Chip Apple Silicon

根据自己的机器选择，大多数同学选择
64bit windows；如红框所示

注意根据自己的习惯自定义选择安装路径



第三部分

Anaconda安装

- Anaconda指的是一个开源的Python发行版本，其包含了Python、conda等180多个科学包及其依赖项。
- Anaconda对于python初学者而言及其友好，相比单独安装python主程序，选择Anaconda可以帮助省去很多麻烦，Anaconda里添加了许多常用的功能包，如果单独安装python，这些功能包则需要一条一条自行安装，这不仅繁琐，还涉及到多个功能包互相依赖导致的版本匹配性问题。而使用anaconda则不需要考虑这类问题。
- 下载网址：
- <https://www.anaconda.com/products/individual>



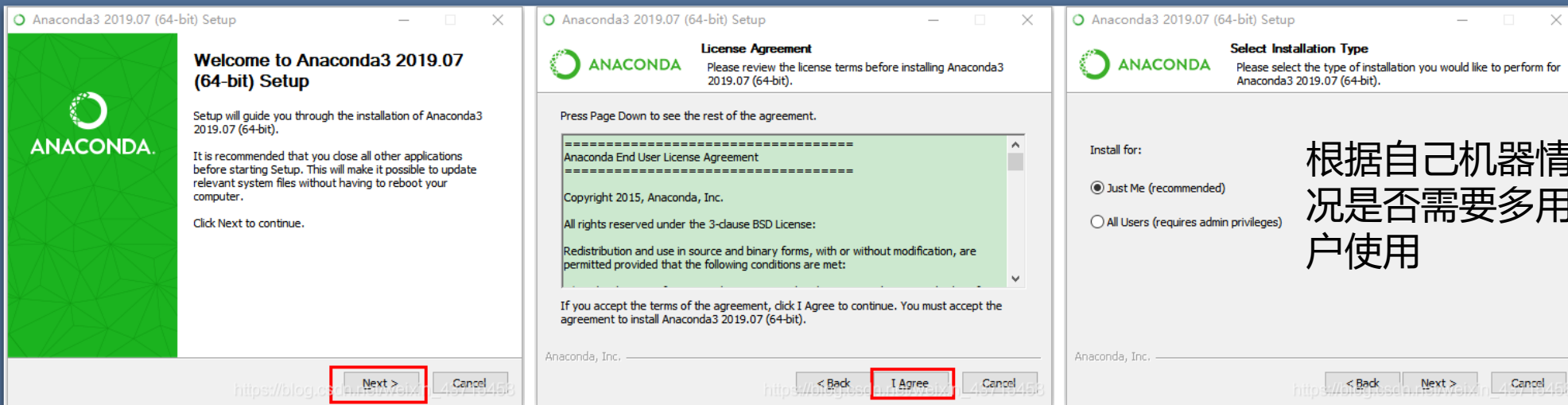
Individual Edition

Your data science toolkit

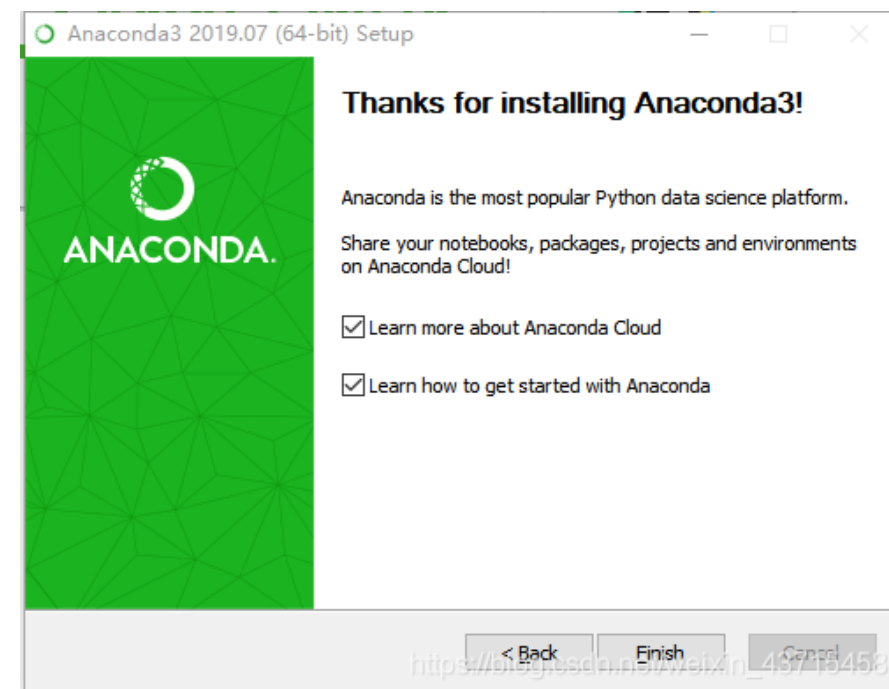
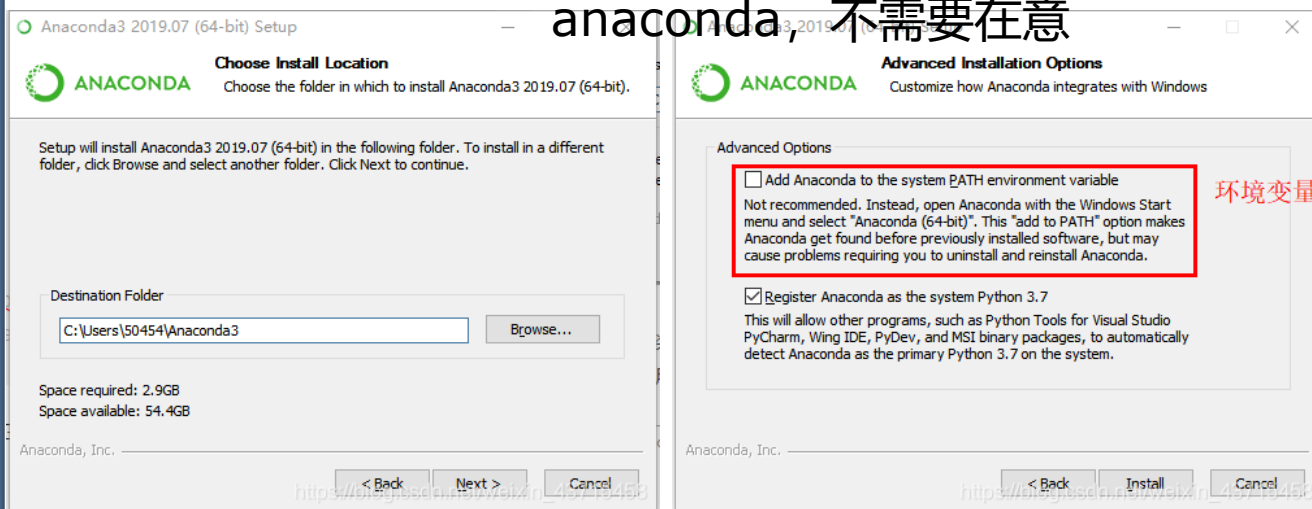
With over 25 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.



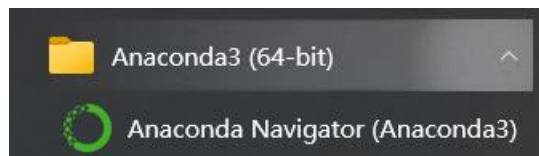
因为anaconda包含了很多python的工具包，文件较大，需要几十分钟的下载。



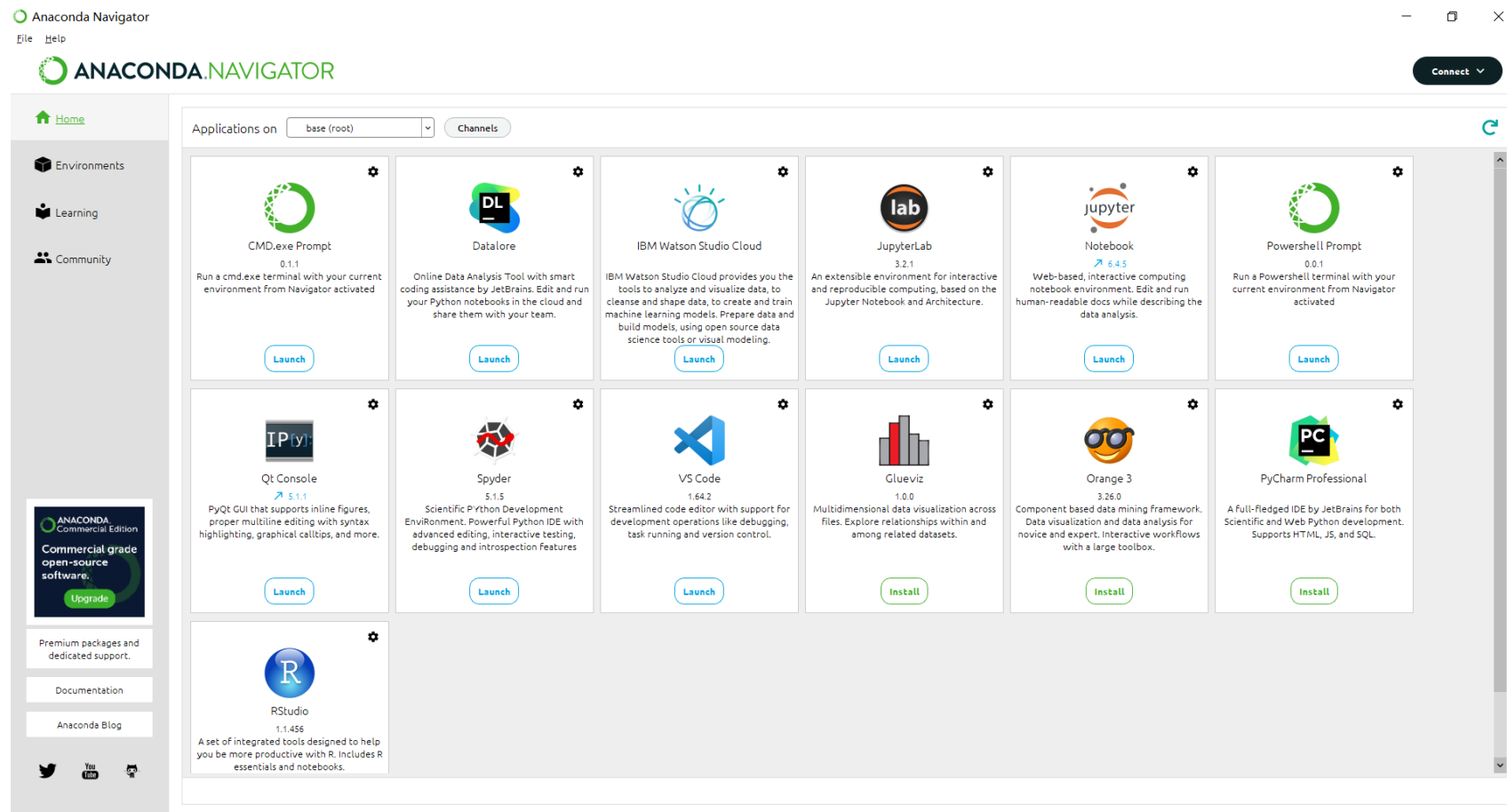
这一个选项是添加环境变量，需要勾选，
变红是指选择后机器上已有程序可以发现
anaconda，不需要在意



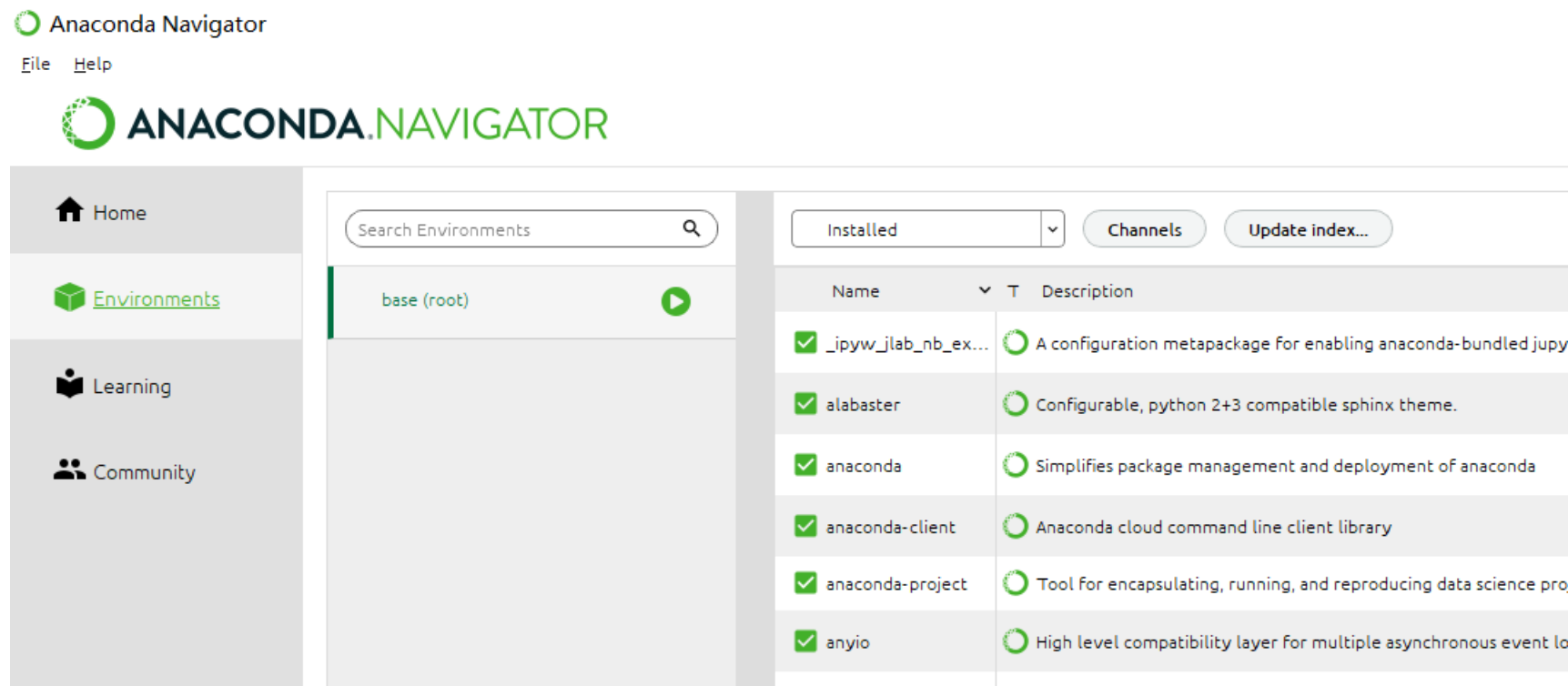
下载完毕后开始界面可以看到：



打开后如下：



点击左侧选项的'Environment'可以看到自动创建的'base'环境。



而接下来实际上就是创建一个新的'tensorflow'环境，这样的好处在于彼此独立，各个工作项目区分开易于管理，和机器上已经有的各种工具包也不会冲突，或者发生不同版本同时存在引起的各种问题



第四部分

Tensorflow环境搭建

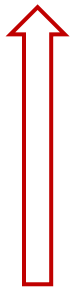
在创建环境之前首先需要明确python版本和tensorflow版本的对应关系，版本不匹配会产生很多错误

| Framework | Env name (-env parameter) | Description | Docker Image | Packages and Nvidia Settings |
|----------------|---------------------------|---|-------------------------------------|--------------------------------|
| TensorFlow 2.2 | tensorflow-2.2 | TensorFlow 2.2.0 + Keras 2.3.1 on Python 3.7. | floydhub/tensorflow | TensorFlow-2.2 |
| TensorFlow 2.1 | tensorflow-2.1 | TensorFlow 2.1.0 + Keras 2.3.1 on Python 3.6. | floydhub/tensorflow | TensorFlow-2.1 |
| TensorFlow 2.0 | tensorflow-2.0 | TensorFlow 2.0.0 + Keras 2.3.1 on Python 3.6. | floydhub/tensorflow | TensorFlow-2.0 |

无论机器是否有GPU，tensorflow2.0系列版本安装时不需要额外安装gpu版本

这里统一安装python 3.7 & tensorflow-2.2.0版本

注：tensorflow，根据是否支持gpu，分为tensorflow-cpu&tensorflow-gpu，其中关系大致上：



| 环境 | tensorflow==1.x | tensorflow-gpu==1.x |
|------------------|-----------------|---------------------|
| 只有CPU | cpu运行 | 和tensorflow一样运行 |
| 有GPU且装Cuda和Cudnn | cpu运行 | gpu运行 |
| 有GPU未装Cuda或Cudnn | cpu运行 | 和tensorflow一样运行 |

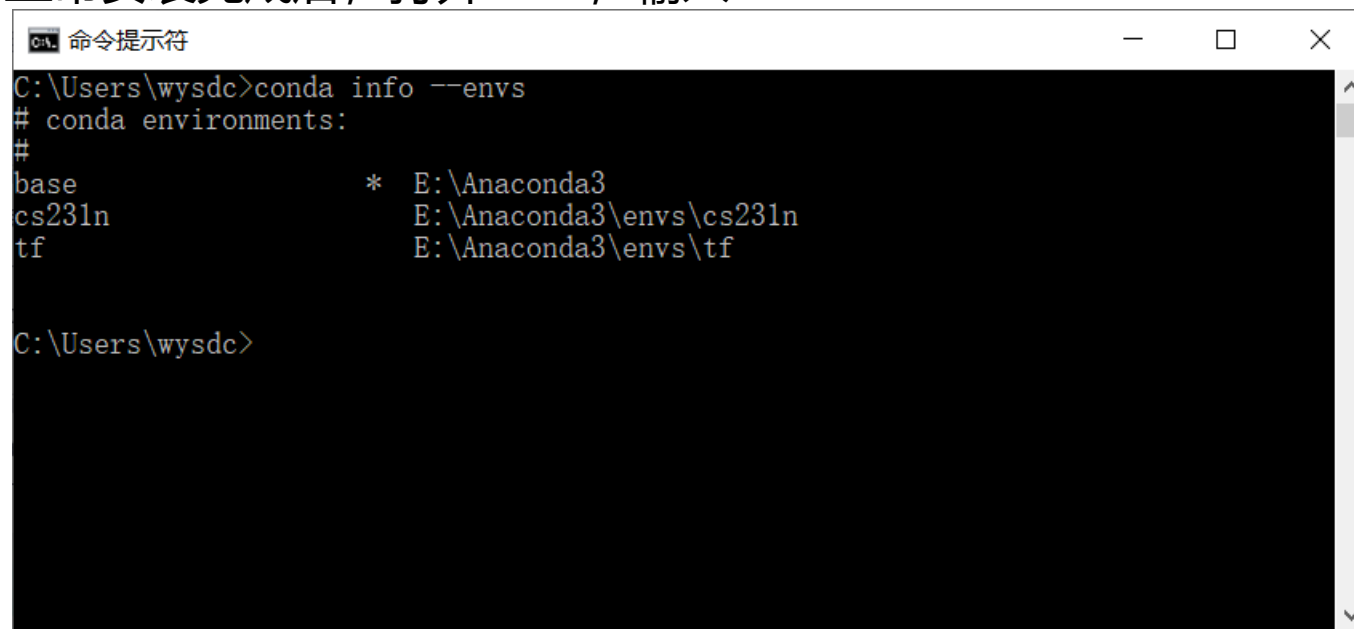
| 环境 | tensorflow-cpu==2.x | tensorflow==2.x |
|------------------|---------------------|-----------------|
| 只有CPU | cpu运行 | cpu运行 |
| 有GPU且装Cuda和Cudnn | cpu运行 | gpu运行 |
| 有GPU未装Cuda或Cudnn | cpu运行 | cpu运行 |

在windows下安装好anaconda后，如果在cmd中输入conda出现报错：

`conda'不是内部或外部命令，也不是可运行的程序或批处理文件

原因是上一步没有正常添加环境变量，此时可以将文件目录下的scripts添加到环境变量即可

正常安装完成后，打开cmd，输入：



```
命令提示符
C:\Users\wysdc>conda info --envs
# conda environments:
#
base                * E:\Anaconda3
cs231n              E:\Anaconda3\envs\cs231n
tf                  E:\Anaconda3\envs\tf

C:\Users\wysdc>
```

可以查看已有的环境

接下来创建tensorflow环境，（这里命名可以自行发挥，比如tf之类，方便易区分即可），并指定python版本为3.7

```
命令提示符 - "E:\Anaconda3\condabin\conda.bat" create --name tensorflow python=...
C:\Users\wysdc>conda create --name tensorflow python==3.7
Collecting package metadata (current_repodata.json): done
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): /
```

按要求输入 y 然后enter即可

```
命令提示符 - "E:\Anaconda3\condabin\conda.bat" create --name tensorflow python=...
The following NEW packages will be INSTALLED:

certifi      pkgs/main/win-64::certifi-2021.10.8-py37haa95532_2
pip          pkgs/main/win-64::pip-21.2.4-py37haa95532_0
python       pkgs/main/win-64::python-3.7.0-hea74fb7_0
setuptools   pkgs/main/win-64::setuptools-58.0.4-py37haa95532_0
vc           pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel        pkgs/main/noarch::wheel-0.37.1-pyhd3eb1b0_0
wincertstore pkgs/main/win-64::wincertstore-0.2-py37haa95532_2

Proceed ([y]/n)? y

Downloading and Extracting Packages
python-3.7.0      | 16.6 MB | #####5 | 86%
```

之后显示如下,

```
命令提示符
done
#
# To activate this environment, use
#
#   $ conda activate tensorflow
#
# To deactivate an active environment, use
#
#   $ conda deactivate
#

C:\Users\wysdc>
```



这里两条指令就是从base环境切换到指定名字环境、以及返回base环境的命令

使用上述命令行切换到新建立的python3.7版本的环境'tensorflow', 可以看到()内显示为tensorflow说明切换成功, 然后进行tensorflow2.2版本的安装:

```
命令提示符 - "E:\Anaconda3\condabin\conda.bat" activate tensorflow - "E:\Anaconda3...
C:\Users\wysdc>conda info --envs
# conda environments:
#
base                * E:\Anaconda3
cs231n              E:\Anaconda3\envs\cs231n
tensorflow           E:\Anaconda3\envs\tensorflow
tf                  E:\Anaconda3\envs\tf

C:\Users\wysdc>conda activate tensorflow
找不到 C:\Windows\Temp\tmp18mgd9c3.bat

(tensorflow) C:\Users\wysdc>
```

运行下述指令：

```
命令提示符 - "E:\Anaconda3\condabin\conda.bat" activate tensorflow - "E:\Anaconda3...  
(tensorflow) C:\Users\wysdc>pip install tensorflow==2.2.0  
Collecting tensorflow==2.2.0  
  Downloading tensorflow-2.2.0-cp37-cp37m-win_amd64.whl (459.2 MB)  
    | 604 kB 13 kB/s eta 9:32:37
```

下载速度通常很慢，可以使用清华的镜像源，
在原指令后面加上' -i <https://pypi.tuna.tsinghua.edu.cn/simple/>

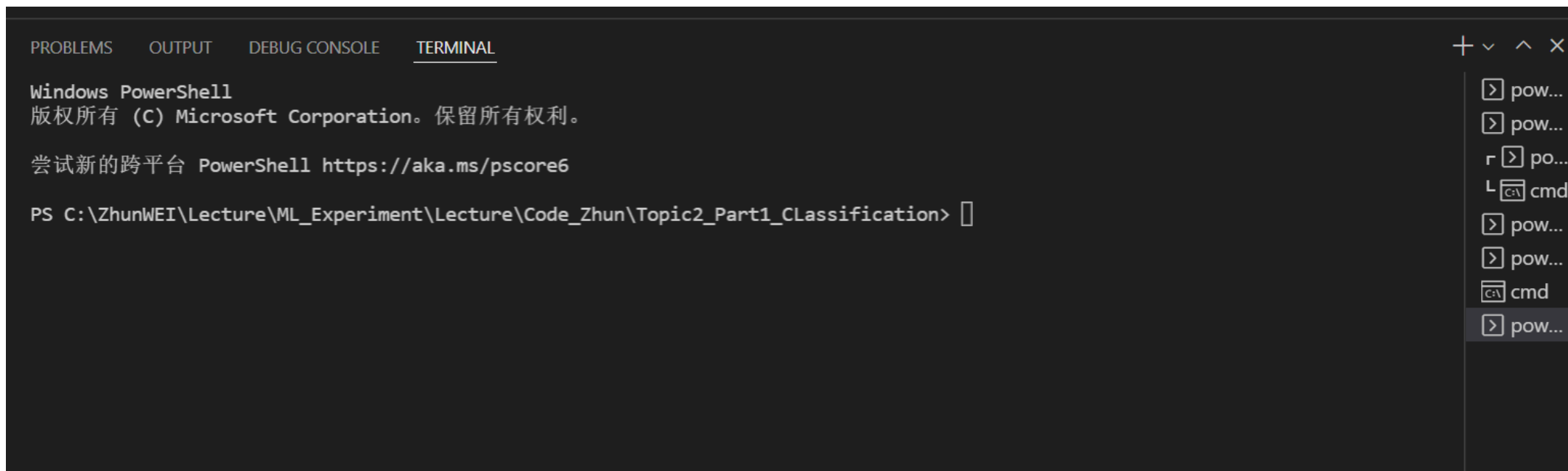
```
命令提示符 - "E:\Anaconda3\condabin\conda.bat" activate tensorflow  
(tensorflow) C:\Users\wysdc>pip install tensorflow==2.2.0 -i https://pypi.tuna.tsinghua.edu.cn/simple/  
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple/  
Collecting tensorflow==2.2.0
```

使用pip list查看当前环境下的工具包，可以看到安装好的tenorflow2.2.0，以及h5py, numpy等等相关的工具包

```
命令提示符 - "E:\Anaconda3\condabin\conda.bat" activate tensorflow
h5py                2.10.0
idna                 3.3
importlib-metadata  4.11.0
isort                5.10.1
Keras-Preprocessing 1.1.2
lazy-object-proxy    1.7.1
Markdown             3.3.6
mccabe               0.6.1
numpy                1.21.5
oauthlib             3.2.0
opt-einsum           3.3.0
pip                  21.2.4
platformdirs         2.5.0
protobuf             3.19.4
pyasn1               0.4.8
pyasn1-modules       0.2.8
pylint               2.12.2
requests             2.27.1
requests-oauthlib    1.3.1
rsa                  4.8
scipy                1.4.1
setuptools           58.0.4
six                  1.16.0
tensorboard          2.2.2
tensorboard-plugin-wit 1.8.1
tensorflow            2.2.0
tensorflow-estimator 2.2.0
termcolor            1.1.0
```

这样'tensorflow'这个独立的环境内部需要的工具包就安装好了，以后要继续安装新的工具包，需要进入这个环境，再安装。

接下来打开安装好的vscode，左上角选择Terminal> New Terminal



The screenshot shows the Visual Studio Code interface with the 'TERMINAL' tab selected. The terminal window displays the following text:

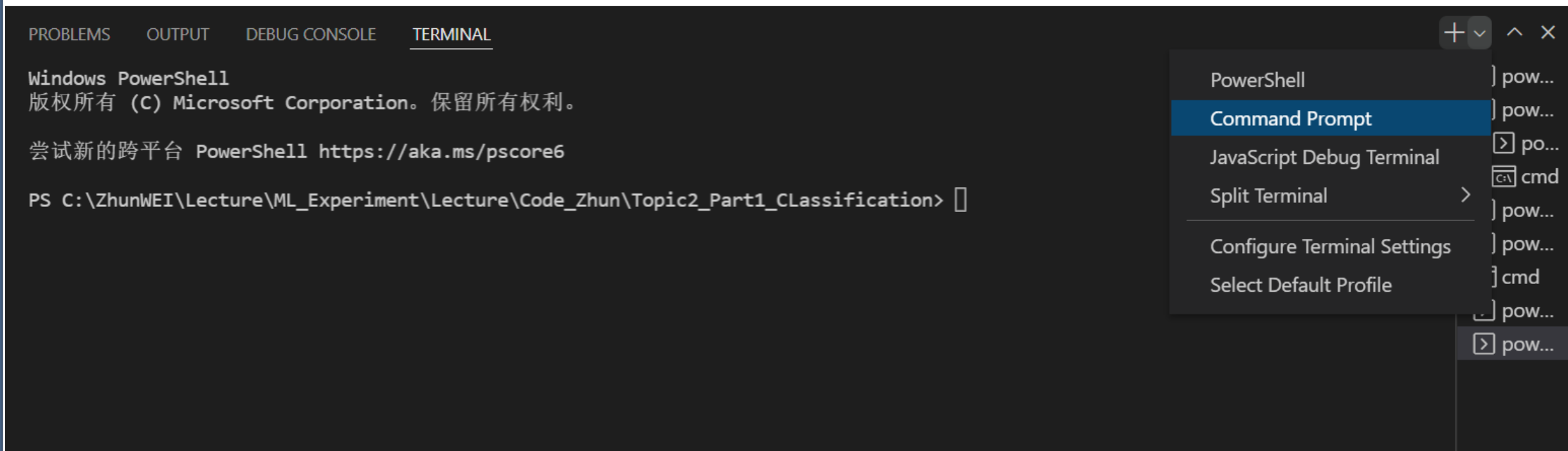
```
Windows PowerShell  
版权所有 (C) Microsoft Corporation。保留所有权利。  
  
尝试新的跨平台 PowerShell https://aka.ms/powershell  
  
PS C:\ZhunWEI\Lecture\ML_Experiment\Lecture\Code_Zhun\Topic2_Part1_Classification>
```

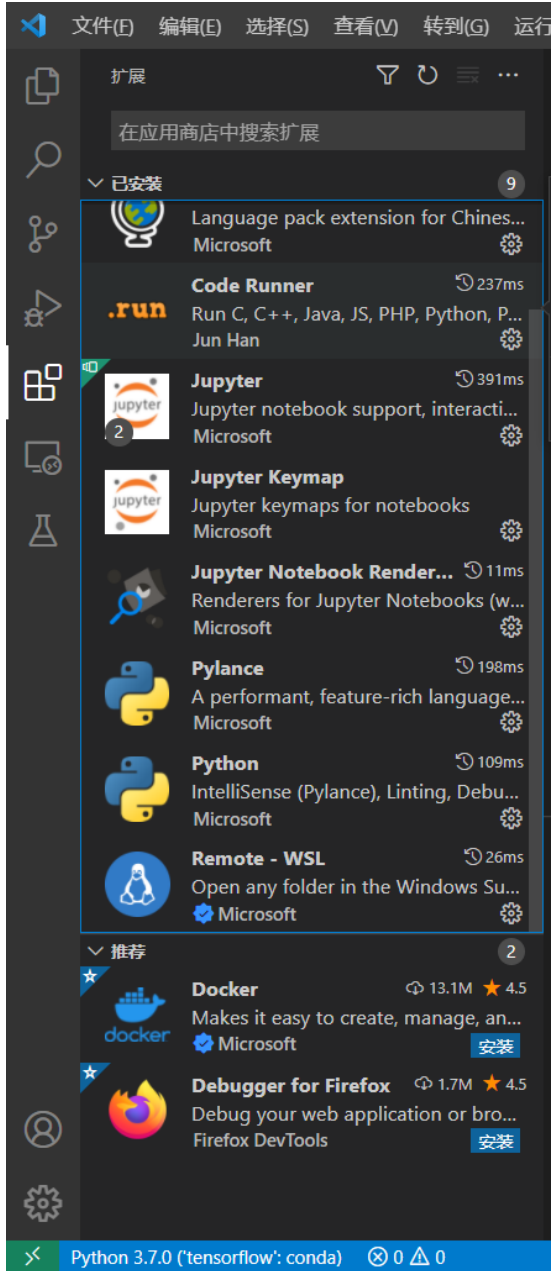
On the right side of the terminal window, there is a list of recent terminal sessions. The sessions are:

- > pow...
- > pow...
- r > po...
- L CA cmd
- > pow...
- > pow...
- CA cmd
- > pow...

右上角选择加号>Command Prompt, 在新的terminal里面输入conda activate tensorflow 即进入之前配置的tensorflow 环境;

在terminal 里面输入 pip install matplotlib 供今后画图使用





Vscode可以方便的添加拓展，这些有用的拓展可以让编写代码更加容易，可以自行了解并安装

对vscode不熟悉的同学可以自学简单的教程



第五部分

(选) GPU配置

Tensorflow2.0系列均支持GPU，而GPU并行计算的特点使得使用GPU进行神经网络训练等计算效率更高。有条件的情况下，可以选择使用GPU

- 以NVIDIA显卡为例，根据tensorflow官网，可以查询到‘计算能力’，通常 ≥ 3.5 可以选择使用GPU

| GPU | Compute Capability |
|---------------------|--------------------|
| GeForce RTX 3090 | 8.6 |
| GeForce RTX 3080 | 8.6 |
| GeForce RTX 3070 | 8.6 |
| NVIDIA TITAN RTX | 7.5 |
| Geforce RTX 2080 Ti | 7.5 |
| Geforce RTX 2080 | 7.5 |
| Geforce RTX 2070 | 7.5 |
| Geforce RTX 2060 | 7.5 |
| NVIDIA TITAN V | 7.0 |
| NVIDIA TITAN Xp | 6.1 |
| NVIDIA TITAN X | 6.1 |
| GeForce GTX 1080 Ti | 6.1 |
| GeForce GTX 1080 | 6.1 |
| GeForce GTX 1070 | 6.1 |
| GeForce GTX 1060 | 6.1 |
| GeForce GTX 1050 | 6.1 |
| GeForce GTX TITAN X | 5.2 |

<https://blog.csdn.net/wbing90>

Tensorflow要使用GPU，还需要安装一些软件工具包

软件要求

必须在系统中安装以下 NVIDIA® 软件：

- [NVIDIA® GPU 驱动程序](#) [🔗](#)：CUDA® 10.1 需要 418.x 或更高版本。
- [CUDA® 工具包](#) [🔗](#)：TensorFlow 支持 CUDA® 10.1（TensorFlow 2.1.0 及更高版本）
- CUDA® 工具包附带的 [CUPTI](#) [🔗](#)。
- [cuDNN SDK 7.6](#) [🔗](#)
- （可选）[TensorRT 6.0](#) [🔗](#)，可缩短用某些模型进行推断的延迟时间并提高吞吐量。

在tensorflow官网中可以查找[CUDA工具包](#)，[cuDNN](#) 和[tensorflow&python](#)的版本匹配关系。

在 Windows 环境中从源代码构建 | TensorFlow (google.cn)
https://tensorflow.google.cn/install/source_windows

| 版本 | Python 版本 | 编译器 | 构建工具 | cuDNN | CUDA |
|----------------------|-----------|-----------|-------------|-------|------|
| tensorflow_gpu-2.6.0 | 3.6-3.9 | MSVC 2019 | Bazel 3.7.2 | 8.1 | 11.2 |
| tensorflow_gpu-2.5.0 | 3.6-3.9 | MSVC 2019 | Bazel 3.7.2 | 8.1 | 11.2 |
| tensorflow_gpu-2.4.0 | 3.6-3.8 | MSVC 2019 | Bazel 3.1.0 | 8.0 | 11.0 |
| tensorflow_gpu-2.3.0 | 3.5-3.8 | MSVC 2019 | Bazel 3.1.0 | 7.6 | 10.1 |
| tensorflow_gpu-2.2.0 | 3.5-3.8 | MSVC 2019 | Bazel 2.0.0 | 7.6 | 10.1 |

根据对应关系，统一安装
CUDA10.1以及cuDNN7.6

CUDA安装网址：
<https://developer.nvidia.com/cuda-toolkit-archive>

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System

Windows Linux Mac OSX

Architecture

x86_64

Version

10 8.1 7 Server 2019 Server 2016 Server 2012 R2

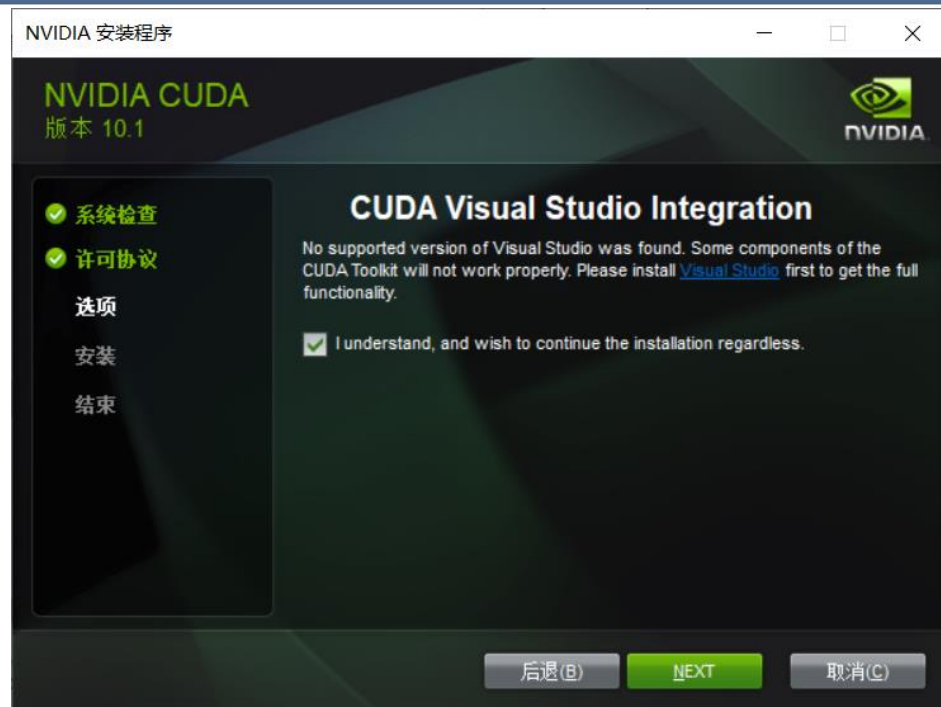
Installer Type

exe (network) exe (local)

- 下载好安装程序后进行安装时务必选择自定义安装：
 - 一方面可以**选择安装路径**
 - 另一方面有一步选择可以勾选使得**无需提前安装 visual studio**，否则若没有提前安装后者（10G大小），会报错（见下页）



选择自定义



勾选

需要注意的是：

显卡驱动的版本也和CUDA版本存在匹配关系，且是向下兼容的，换句话说如果上一步安装失败，可能是机器的显卡以及显卡驱动版本较久，这个匹配关系可以在官网查看

<https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html>

Table 3. CUDA Toolkit and Corresponding Driver Versions

| CUDA Toolkit | Toolkit Driver Version | |
|----------------------|-----------------------------|-------------------------------|
| | Linux x86_64 Driver Version | Windows x86_64 Driver Version |
| CUDA 11.6 GA | >=510.39.01 | >=511.23 |
| CUDA 11.5 Update 2 | >=495.29.05 | >=496.13 |
| CUDA 11.5 Update 1 | >=495.29.05 | >=496.13 |
| CUDA 11.5 GA | >=495.29.05 | >=496.04 |
| CUDA 11.4 Update 4 | >=470.82.01 | >=472.50 |
| CUDA 11.4 Update 3 | >=470.82.01 | >=472.50 |
| CUDA 11.4 Update 2 | >=470.57.02 | >=471.41 |
| CUDA 11.4 Update 1 | >=470.57.02 | >=471.41 |
| CUDA 11.4.0 GA | >=470.42.01 | >=471.11 |
| CUDA 11.3.1 Update 1 | >=465.19.01 | >=465.89 |
| CUDA 11.3.0 GA | >=465.19.01 | >=465.89 |

一些比较旧的显卡以及其相应的驱动程序只支持较低版本的CUDA工具包，
相应的，匹配的tensorflow和python版本都会变化，如果坚持要使用的同学，可以参照前面给出的网址自己查询匹配的版本并重新安装。会花一些时间

个人是不建议的，这种情况就使用cpu就好了

安装cuDNN7.6

安装网址：

<https://developer.nvidia.com/rdp/cudnn-archive>

过程中需要注册NVIDIA developer，根据系统提示即可

NVIDIA Developer Program Membership Required

The file or page you have requested requires membership in the NVIDIA Developer Program. Please either log in or join the program to access this material. You can [learn more](#) about the benefits of the NVIDIA Developer Program here.

Login

Join now

<https://blog.csdn.net/wbing96>

cuDNN7.6以及CUDA10.1安装完成后，需要进行Windows 设置。

官方指南网址：

https://tensorflow.google.cn/install/gpu#hardware_requirements

Windows 设置

请参阅上面列出的[硬件要求](#)和[软件要求](#)，并阅读[适用于 Windows 的 CUDA® 安装指南](#) [🔗](#)。

确保安装的 NVIDIA 软件包与上面列出的版本一致。特别是，如果没有 `cuDNN64_8.dll` 文件，TensorFlow 将无法加载。如需使用其他版本，请参阅在 [Windows 下从源代码构建](#) 指南。

将 CUDA®、CUPTI 和 cuDNN 安装目录添加到 `%PATH%` 环境变量中。例如，如果 CUDA® 工具包安装到 `C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.0`，并且 cuDNN 安装到 `C:\tools\cuda`，请更新 `%PATH%` 以匹配路径：

```
C:\> SET PATH=C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.0\bin;%PATH%
C:\> SET PATH=C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.0\extras\CUPTI\lib64;%PATH%
C:\> SET PATH=C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.0\include;%PATH%
C:\> SET PATH=C:\tools\cuda\bin;%PATH%
```

1、添加环境变量

将CUDA和cuDNN的bin目录添加到系统的环境变量

2、将cuDNN目录下的

bin **include** **lib**

三个文件夹复制到CUDA文件夹下，**粘贴时选择直接覆盖即可**

(在CUDA实际安装时发现虽然修改了路径，文件仍然下载到了默认的C盘路径，复制粘贴的时候需要注意)

由于tensorflow2.0系列默认支持GPU，如果没有进行GPU配置相关工作，在终端运行时会有警告，表示没有cuDNN64_8.dll 文件。此时实际上使用cpu。而配置好GPU之后，运行时tenserflow将会使用GPU，上述警告消失