

从零开始深度学习环境配置（Windows）

假如你有一台新电脑，电脑未进行任何相关环境的配置。那么，我将带领大家手把手进行深度学习详细的环境配置。

1. 软件安装

我们需要安装的软件是：



- **Anaconda:** 配置不同的虚拟环境
 - 进入Anaconda官网下载页面（<https://www.anaconda.com/download>）后下载 `exe` 文件，一步步按照指示安装就可以。
 - 配置Jupyter Notebook（交互式笔记本）
- **Pycharm:** 运行代码的集成工具（代码补全、语法高亮等）
 - 前往Pycharm官网（<https://www.jetbrains.com.cn/en-us/pycharm/download/?section=windows>）后下载 `exe` 文件，一步步按照指示安装就可以。

2. 环境配置

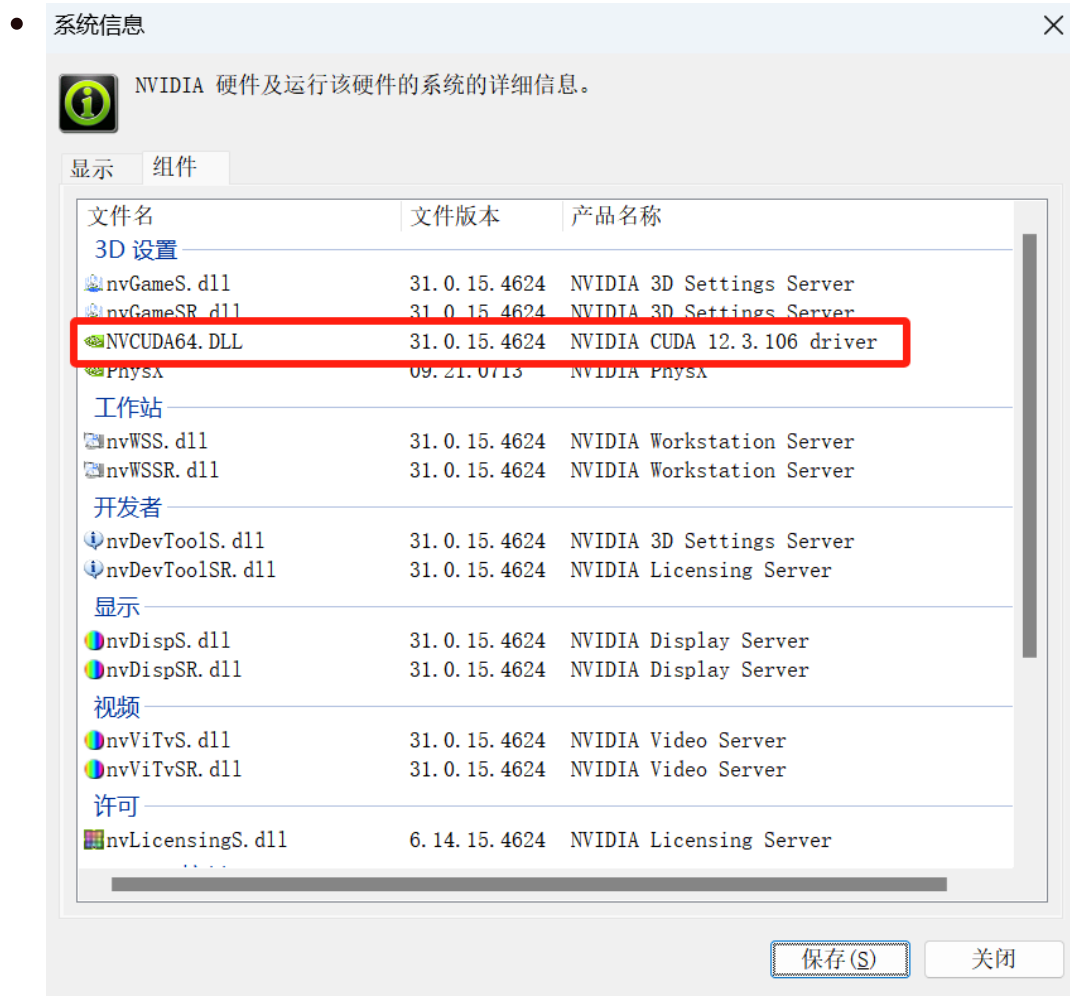
所有的环境都是基于Anaconda配置的，这样可以创建多个虚拟环境，将不同的环境隔离开来。本次以配置torch环境为例子。

1. 安装CUDA及cudnn

如果要使用GPU进行模型的训练，则必须安装CUDA和cudnn

- **CUDA:** 是显卡厂商NVIDIA推出的只能用于自家GPU的并行计算框架。
- **cudnn:** 是针对深度卷积神经网络的加速库。CUDNN需要在有CUDA的基础上进行，CUDNN可以在CUDA基础上加速2倍以上。

1. 检查自己电脑支持的cuda版本



- 方式二：win+r 然后输入cmd 打开命令行窗口->输入 `nvidia-smi`

```

C:\Windows\system32\cmd.e: X + v
Microsoft Windows [版本 10.0.22631.3593]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\Guocc>nvidia-smi
Sat May 25 22:54:30 2024

+-----+
| NVIDIA-SMI 546.24                  Driver Version: 546.24          CUDA Version: 12.3         |
+-----+-----+-----+-----+-----+-----+
| GPU  Name            TCC/WDDM    Bus-Id      Disp.A   Volatile Uncorr. ECC |
| Fan  Temp            Perf         Pwr:Usage/Cap     Memory-Usage   GPU-Util  Compute M. |
|              MIG M. |
+-----+-----+-----+-----+-----+-----+
| 0   NVIDIA GeForce RTX 4060 ... WDDM    00000000:01:00:00 On      N/A       |
| N/A   42C            P8              3W / 140W      1005MiB / 8188MiB      30%      Default  |
+-----+-----+-----+-----+-----+-----+
|                                         N/A       |
+-----+-----+-----+-----+-----+

```

检查发现自己的电脑支持安装CUDA的版本最高为12.3，那么只要我们安装一个不超过12.3的版本即可。

2. 前往官网下载CUDA

- 下载链接：<https://developer.nvidia.com/cuda-toolkit-archive>
- 进入下载页面，选择要下载的版本，我这里选择11.8.0

[CUDA Toolkit 12.4.1 \(April 2024\), Versioned Online Documentation](#)
[CUDA Toolkit 12.4.0 \(March 2024\), Versioned Online Documentation](#)
[CUDA Toolkit 12.3.2 \(January 2024\), Versioned Online Documentation](#)
[CUDA Toolkit 12.3.1 \(November 2023\), Versioned Online Documentation](#)
[CUDA Toolkit 12.3.0 \(October 2023\), Versioned Online Documentation](#)
[CUDA Toolkit 12.2.2 \(August 2023\), Versioned Online Documentation](#)
[CUDA Toolkit 12.2.1 \(July 2023\), Versioned Online Documentation](#)
[CUDA Toolkit 12.2.0 \(June 2023\), Versioned Online Documentation](#)
[CUDA Toolkit 12.1.1 \(April 2023\), Versioned Online Documentation](#)
[CUDA Toolkit 12.1.0 \(February 2023\), Versioned Online Documentation](#)
[CUDA Toolkit 12.0.1 \(January 2023\), Versioned Online Documentation](#)
[CUDA Toolkit 12.0.0 \(December 2022\), Versioned Online Documentation](#)
[CUDA Toolkit 11.8.0 \(October 2022\), Versioned Online Documentation](#)
[CUDA Toolkit 11.7.1 \(August 2022\), Versioned Online Documentation](#)
[CUDA Toolkit 11.7.0 \(May 2022\), Versioned Online Documentation](#)
[CUDA Toolkit 11.6.2 \(March 2022\), Versioned Online Documentation](#)
[CUDA Toolkit 11.6.1 \(February 2022\), Versioned Online Documentation](#)
[CUDA Toolkit 11.6.0 \(January 2022\), Versioned Online Documentation](#)
[CUDA Toolkit 11.5.2 \(February 2022\), Versioned Online Documentation](#)
[CUDA Toolkit 11.5.1 \(November 2021\), Versioned Online Documentation](#)

- 选择版本，下载

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the [CUDA EULA](#).

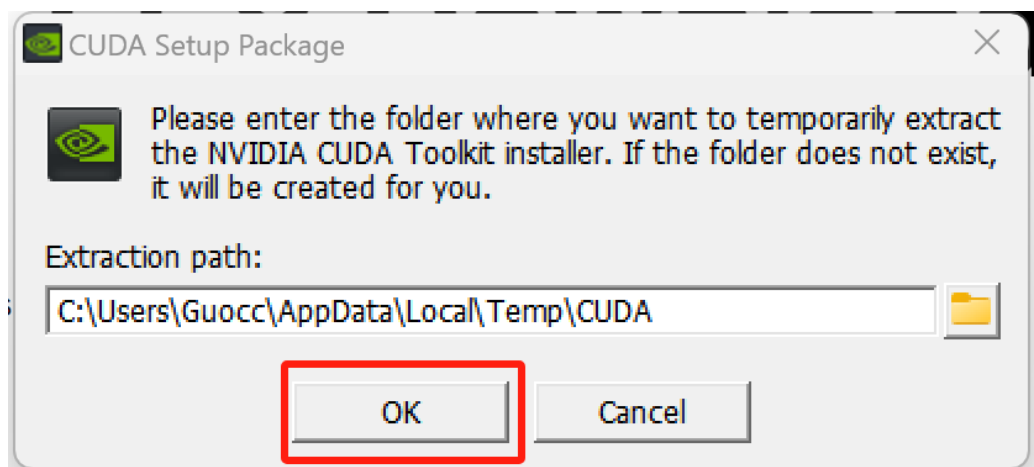
Operating System	<input type="button" value="Linux"/>	<input checked="" type="button" value="Windows"/>
Architecture	<input checked="" type="button" value="x86_64"/>	
Version	<input type="button" value="10"/>	<input checked="" type="button" value="11"/>
	<input type="button" value="Server 2016"/>	<input type="button" value="Server 2019"/>
		<input type="button" value="Server 2022"/>
Installer Type	<input type="button" value="exe (local)"/>	<input checked="" type="button" value="exe (network)"/>

Download Installer for Windows 11 x86_64

The base installer is available for download below.

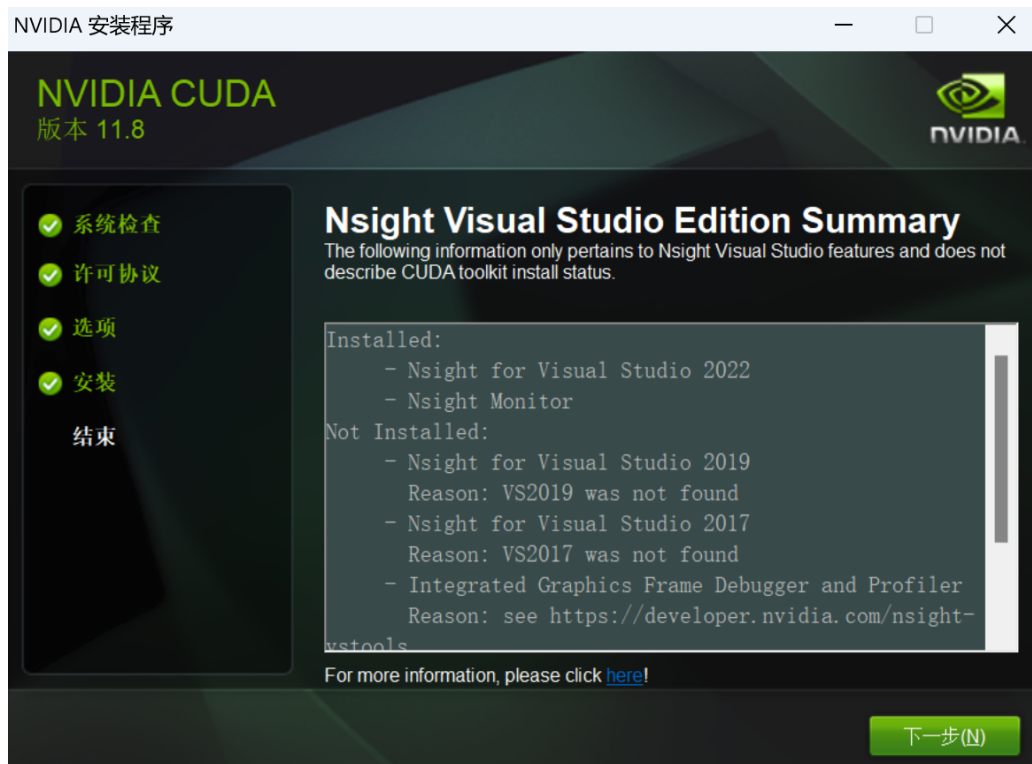
> Base Installer	<input checked="" type="button" value="Download (29.1 MB)"/>
-------------------------------------	--

- 双击下载好的文件，按照下面的流程进行









- 检验是否安装成功：打开cmd窗口执行 `nvcc -V`

```
Microsoft Windows [版本 10.0.22631.3593]  
(c) Microsoft Corporation。保留所有权利。  
  
C:\Users\Guocc>nvcc -V  
nvcc: NVIDIA (R) Cuda compiler driver  
Copyright (c) 2005-2022 NVIDIA Corporation  
Built on Wed_Sep_21_10:41:10_Pacific_Daylight_Time_2022  
Cuda compilation tools, release 11.8, V11.8.89  
Build cuda_11.8.r11.8/compiler.31833905_0
```

3. 安装cudnn

- 网址链接: <https://developer.nvidia.com/rdp/cudnn-archive>
- 选择对应版本（在下载时可能会要求你注册账号，自行注册一下就好）

cuDNN Archive

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

Download cuDNN v8.9.7 (December 5th, 2023), for CUDA 12.x
Download cuDNN v8.9.7 (December 5th, 2023), for CUDA 11.x
Download cuDNN v8.9.6 (November 1st, 2023), for CUDA 12.x
Download cuDNN v8.9.6 (November 1st, 2023), for CUDA 11.x
Download cuDNN v8.9.5 (October 27th, 2023), for CUDA 12.x

Download cuDNN v8.9.7 (December 5th, 2023), for CUDA 11.x

Local Installers for Windows and Linux, Ubuntu(x86_64, armsbsa)

Local Installer for Windows (Zip)

Local Installer for Linux x86_64 (Tar)

Local Installer for Linux PPC (Tar)

Local Installer for Linux SBSA (Tar)

Local Installer for Debian 11 (Deb)







Local Installer for Ubuntu18.04 x86_64 (Deb)

Local Installer for Ubuntu20.04 x86_64 (Deb)

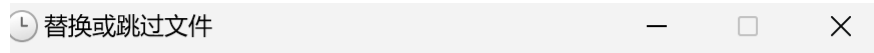
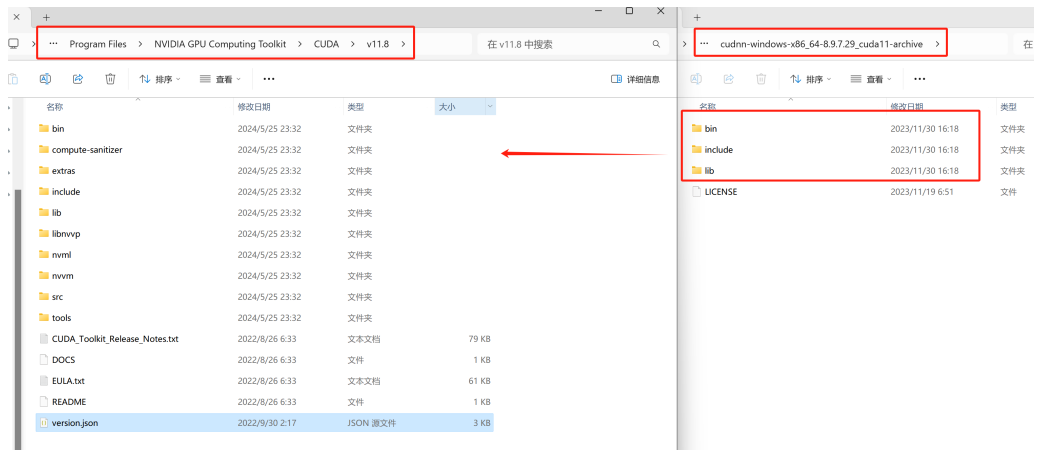
Local Installer for Ubuntu22.04 x86_64 (Deb)

Local Installer for Ubuntu20.04 aarch64sbsa (Deb)

- 将下载的zip文件解压

 cudnn-windows-x86_64-8.9.7.29_cuda11-ar...	2024/5/25 23:55
 cuda_11.8.0_windows_network.exe	2024/5/25 23:15
 pycharm-professional-2024.1.1.exe	2024/5/24 22:25
 Anaconda3-2024.02-1-Windows-x86_64.exe	2024/5/24 22:23
 course	2024/5/24 23:04
 cudnn-windows-x86_64-8.9.7.29_cuda11-ar...	2023/11/30 16:18

- 文件替换



正在将 34 个项目从 cudnn-windows-x86_64-8.9.7.29_c... 复制到 v11.8
目标包含 5 个同名文件

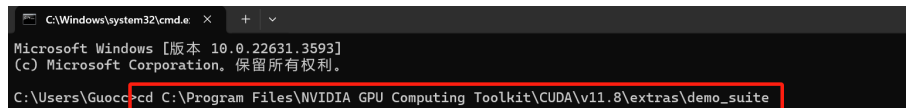
✓ 替换目标中的文件(R)

↪ 跳过这些文件(S)

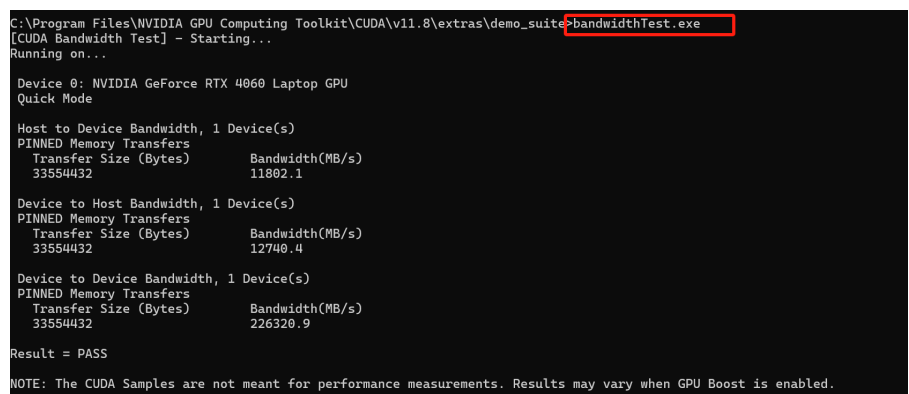
🔄 让我决定每个文件(L)

^ 简略信息

- 打开cmd窗口，检验是否安装成功。
 - 进到指定目录，我的是C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.8\extras\demo_suite



- 执行 bandwidthTest.exe



- 执行 deviceQuery.exe


```
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.8\extras\demo_suite>deviceQuery.exe
deviceQuery.exe Starting...

CUDA Device Query (Runtime API) version (CUDA static linking)

Detected 1 CUDA Capable device(s)

Device 0: "NVIDIA GeForce RTX 4060 Laptop GPU"
CUDA Driver Version / Runtime Version      12.3 / 11.8
CUDA Capability Major/Minor version number: 8.9
Total amount of global memory:              8188 MBytes (8585216000 bytes)
MapSMtoCores for SM 8.9 is undefined. Default to use 128 Cores/SM
MapSMtoCores for SM 8.9 is undefined. Default to use 128 Cores/SM
(24) Multiprocessors, (128) CUDA Cores/MP: 3072 CUDA Cores
GPU Max Clock rate:                        2370 MHz (2.37 GHz)
Memory Clock rate:                         8801 MHz
Memory Bus Width:                          128-bit
L2 Cache Size:                             33554432 bytes
Maximum Texture Dimension Size (x,y,z)      1D=(131072), 2D=(131072, 65536), 3D=(16384, 16384, 16384)
Maximum Layered 1D Texture Size, (num) layers 1D=(32768), 2048 layers
Maximum Layered 2D Texture Size, (num) layers 2D=(32768, 32768), 2048 layers
Total amount of constant memory:            20 bytes
Total amount of shared memory per block:    20 bytes
Total number of registers available per block: 65536
Warp size:                                   32
Maximum number of threads per multiprocessor: 1536
Maximum number of threads per block:        1024
Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
Max dimension size of a grid size (x,y,z):  (2147483647, 65535, 65535)
Maximum memory pitch:                       20 bytes
Texture alignment:                          20 bytes
Concurrent copy and kernel execution:        Yes with 1 copy engine(s)
Run time limit on kernels:                  Yes
Integrated GPU sharing Host Memory:          No
Support host page-locked memory mapping:     Yes
Alignment requirement for Surfaces:          Yes
Device has ECC support:                     Disabled
CUDA Device Driver Mode (TCC or WDDM):       WDDM (Windows Display Driver Model)
Device supports Unified Addressing (UVA):     Yes
Device supports Compute Preemption:          Yes
Supports Cooperative Kernel Launch:          Yes
Supports MultiDevice Co-op Kernel Launch:    No
Device PCI Domain ID / Bus ID / Location ID: 0 / 1 / 0
Compute Mode:
    < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 12.3, CUDA Runtime Version = 11.8, NumDevs = 1, Device0 = NVIDIA GeForce RTX 4060 Laptop GPU
Result = PASS
```

2.创建虚拟环境并配置torch

- 在开始菜单栏点击Anaconda Prompt
- 了解基础命令：

```
# 查看有哪些虚拟环境
conda env list

# 激活指定虚拟环境
conda activate 虚拟环境名
```

- 创建虚拟环境

```
# conda create -n 环境名 python=3.x
# 一般建议安装python版本为3.8，兼容性高一点
conda create -n course_torch python=3.8
```

- 激活虚拟环境

在安装torch之前一定要激活虚拟环境

```
# conda activate $虚拟环境名称$
conda activate torch
```

- 安装torch

Pytorch命令复制网址：<https://pytorch.org/get-started/previous-versions/>

```
# 安装的torch版本必须和python版本对应，不过3.8版本的python基本和1.4
及以上的都匹配
# 首先打开电脑的cmd，使用命令nvidia-smi查看CUDA的版本号，在pytorch官
网找到对应的版本（不对应好像也行，但还是建议对应）。
# 我的是12.0，我打算安装低版本，为了这个环境能够通用
# conda命令，非常不建议使用！！！！因为很容易安装的是CPU版本的torch
# 建议一定使用对应的pip安装方式
pip install torch==1.13.1+cu116 torchvision==0.14.1+cu116
torchaudio==0.13.1 --extra-index-url
https://download.pytorch.org/whl/cu116
```

- 如果你用conda方式没有安装成功，可以尝试用pip的方式覆盖安装。
- 检验


主要检验安装GPU版本的torch是否成功。

- 方法一：

```
# 使用命令 pip list
pip list
```

我们可以看到下面箭头所指有+cuxxx就证明安装的是GPU版本的torch。
（下图1.13.1代表torch的版本，cu116代表对应CUDA11.6版本的）

```
(torch) C:\Users\Guocc>pip list
Package                Version
-----
protlipy               0.7.0
certifi                2023.5.7
cffi                   1.15.1
charset-normalizer     2.0.4
cryptography           39.0.1
idna                   3.4
mkl-fft                1.3.6
mkl-random             1.2.2
mkl-service            2.4.0
numpy                  1.24.3
Pillow                 9.4.0
pip                    23.1.2
pycparser              2.21
pyOpenSSL              23.0.0
PySocks                1.7.1
requests               2.29.0
setuptools             67.8.0
torch                  1.13.1+cu116
torchaudio             0.13.1
torchvision            0.14.1+cu116
typing_extensions      4.6.3
urllib3                1.26.16
wheel                  0.38.4
win-inet-pton          1.1.0
```



- 方法二:

```
# 首先进入python交互命令行
python
# 导入torch
import torch
# 查看torch版本
torch.__version__ # 我们会看到1.13.1+cu116，证明了安装正确
# 查看cuda是否可用
torch.cuda.is_available() # True证明可以
```

3. 软件配置

1. Jupyter Notebook

本小节主要是讲解Jupyter的工作目录的更换以及内核的添加和切换。

- 打开Anaconda Prompt，执行下面命令

查看配置文件目录

```
jupyter notebook --generate-config #
```

```
C:\Users\Guocc\.jupyter\jupyter_notebook_config.py
```

如果执行上述命令提示命令不存在，则需要把jupyter notebook 添加到环境变量中（具体请去百度，可以参考：Jupyter Notebook环境变量jupyter notebook环境变量配置失眠软糖12的博客-CSDN博客：https://blog.csdn.net/weixin_37862609/article/details/116050777

- 打开配置文件jupyter_notebook_config.py，搜索关键字notebook_dir

jupyter_notebook_config.py - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)




```
## The directory to use for notebooks and kernels.
```

```
# Default: "
```

```
# c.NotebookApp.notebook_dir = "
```

- 将引号内添加你指定的工作目录并把前面的#删除（也就是去除注释）。
- 最后删除下图中的部分

 Jupyter Notebook

目标类型:

应用程序

目标位置:

Anaconda3

目标(T):

its\jupyter-notebook-script.py '%USERPROFILE%/"

起始位置(S):

%HOMEPATH%

快捷键(K):

无

运行方式(R):

常规窗口

备注(O):

Jupyter Notebook

打开文件所在的位置(F)

更改图标(C)...

高级(D)...

CSDN @海涛从不浪

至此，工作目录设置完毕！接下来设置内核。

- 打开Anaconda Prompt，执行下面命令

```
# 查看目前可用的内核
jupyter kernelspec list
```

- 接下来把前面创建的torch添加到jupyter的内核

```
# 激活虚拟环境
conda activate course_torch
# 安装ipykernel
conda install ipykernel
# 将环境写入Notebook的kernel中，其中torch为虚拟环境名称，torch_c为
jupyter显示的名称
python -m ipykernel install --user --name course_torch --
display-name course_torch
```

- 解决jupyter notebook在anaconda prompt命令行无法启动的问题

我的电脑->属性->高级系统设置->环境变量->用户变量->Path->新建变量
D:\Software\Anaconda\Scripts

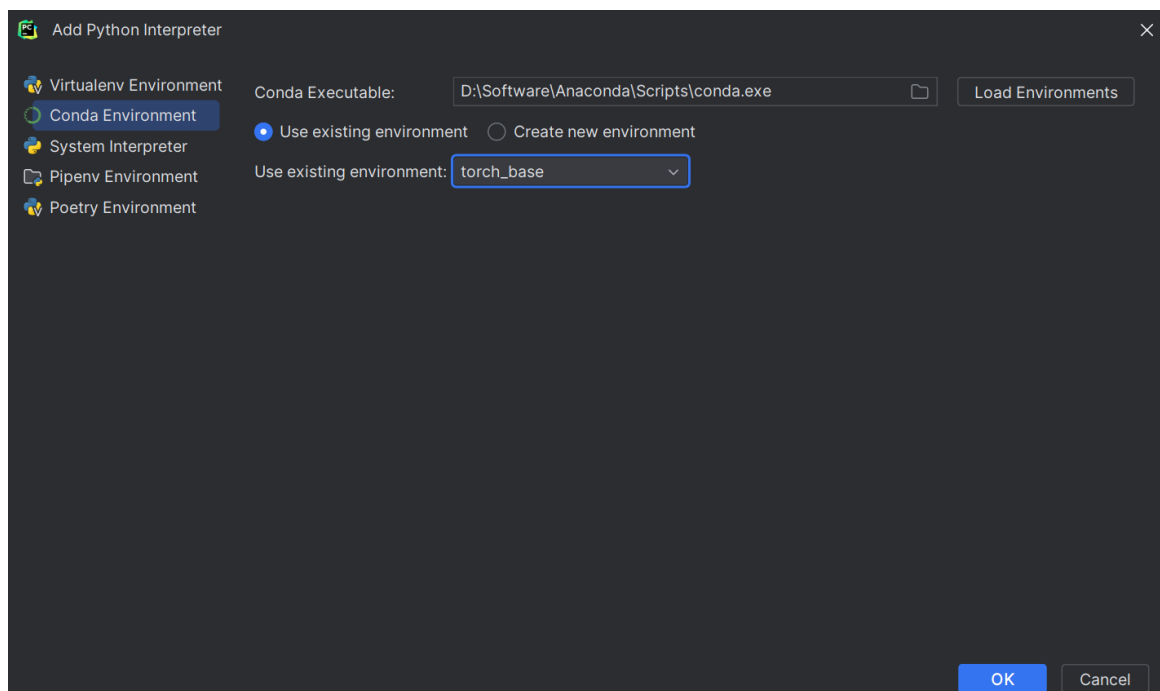
2. Pycharm

- 借助PyCharm在代码中自动添加作者、日期：
 - File（文件）→ Settings（设置）→ Editor（编辑器）→ File and code Templates（文件和代码模板）→ Python Script（Python 脚本）
 - 添加模板

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-
# @Author : Guocc
# @Time : ${DATE} ${TIME}
# @Content :
```

- 为Pycharm添加虚拟环境

注意下图中，添加的是\Anaconda\Scripts\conda.exe路径，然后下面蓝框中选择我的虚拟环境（我创建的虚拟环境名字是torch_base）



4. conda 命令

1. 查看已创建的虚拟环境

```
conda env list
```

2. 激活虚拟环境

```
conda activate [虚拟环境的名称]
```

3. 删除虚拟环境

```
conda deactivate // 确保已经退出当前环境
conda env list // 查看当前存在的所有虚拟环境
conda remove --name [虚拟环境的名称] --all // 删除指定名称的虚拟环境
```

4. 备份虚拟环境

对于一些重要的虚拟环境，我们可能希望进行备份，在系统更换或重装后能够直接恢复。conda提供了一个方便的命令，可以将虚拟环境保存为一个压缩包。

```
conda env export --name [虚拟环境的名称] > environment.yaml
```

这条命令将当前虚拟环境的所有依赖和设置保存到一个名为environment.yaml的文件中。该文件可以作为备份文件进行保存。如果需要恢复这个环境，可以使用以下命令：

```
conda env create --name [新环境的名称] --file environment.yaml
```

5. 重命名虚拟环境

```
conda activate [原虚拟环境的名称]
conda create --name [新虚拟环境的名称] --clone [原虚拟环境的名称]
conda remove --name [原虚拟环境的名称] --all
```

6. 添加镜像源：

```
# 添加清华源
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgsg/free/
```

```
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-
forge/
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/
```

添加阿里云镜像源

```
conda config --add channels
https://mirrors.aliyun.com/anaconda/pkgsg/free/
conda config --add channels
https://mirrors.aliyun.com/anaconda/pkgsg/main/
```

添加中科大源

```
conda config --add channels
https://mirrors.ustc.edu.cn/anaconda/pkgsg/free/
conda config --add channels
https://mirrors.ustc.edu.cn/anaconda/pkgsg/main/
conda config --add channels
https://mirrors.ustc.edu.cn/anaconda/cloud/conda-forge/
conda config --add channels
https://mirrors.ustc.edu.cn/anaconda/cloud/msys2/
conda config --add channels
https://mirrors.ustc.edu.cn/anaconda/cloud/bioconda/
conda config --add channels
https://mirrors.ustc.edu.cn/anaconda/cloud/menpo/
```

设置搜索时显示通道地址

```
conda config --set show_channel_urls yes
```

查看所有源

```
conda config --show-sources
```

删除制定源

```
conda config --remove channels
'https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgsg/free/'
```

删除全部源

```
conda config --remove-key channels
```