

从零开始深度学习环境配置（Windows）

假如你有一台新电脑，电脑什么都没有。我将带领大家从0开始进行深度学习详细的环境配置教程。

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。
- **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。
- **Jupyter Notebook:** 交互式笔记本。
 - 这个Anaconda自带，无需操作。
- ~~超算并行云：租赁的云服务器的一个软件，主要是用来训练模型的（需要GPU）。~~
 - ~~进入并行超算云下载页面（<https://cloud.paratera.com/>）后下载 `exe` 文件，一步步按照指示安装就好。~~

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。

2. Windows环境配置

- 安装cuda和cudnn

详见: https://blog.csdn.net/qq_45904458/article/details/132144817

- 检查cuda是否安装成功的命令: `nvcc -V`
- 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 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  
-----  
protoply           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  
pyparser           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
```

2. 软件配置

1. Jupyter Notebook

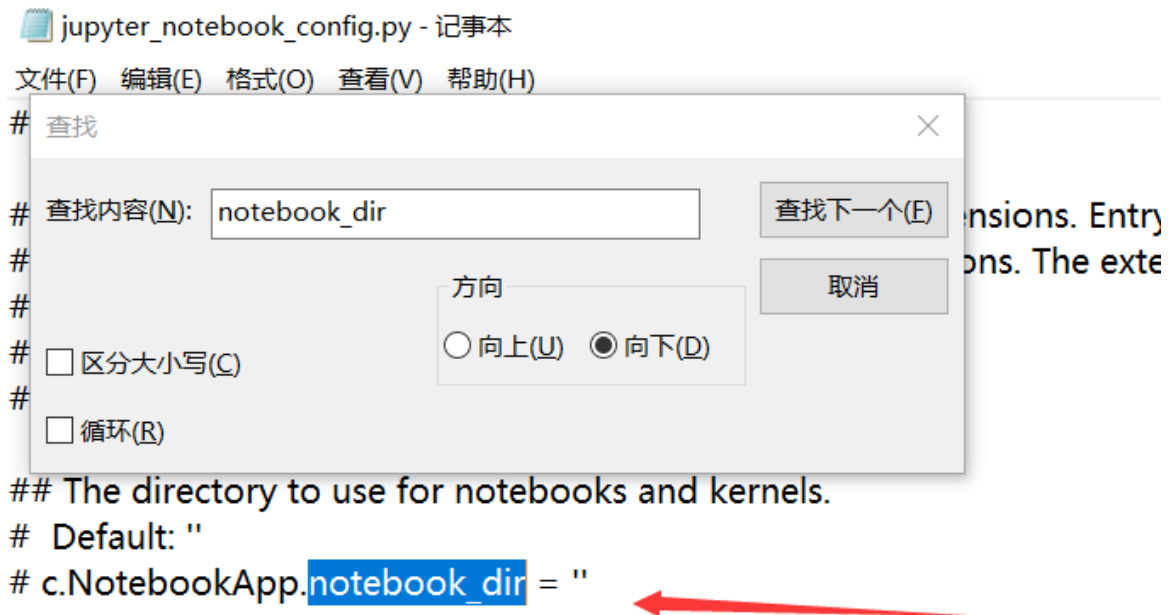
本小节主要是讲解jupyter的工作目录的更换以及内核的添加和切换。

- 打开Anaconda Prompt，执行下面命令

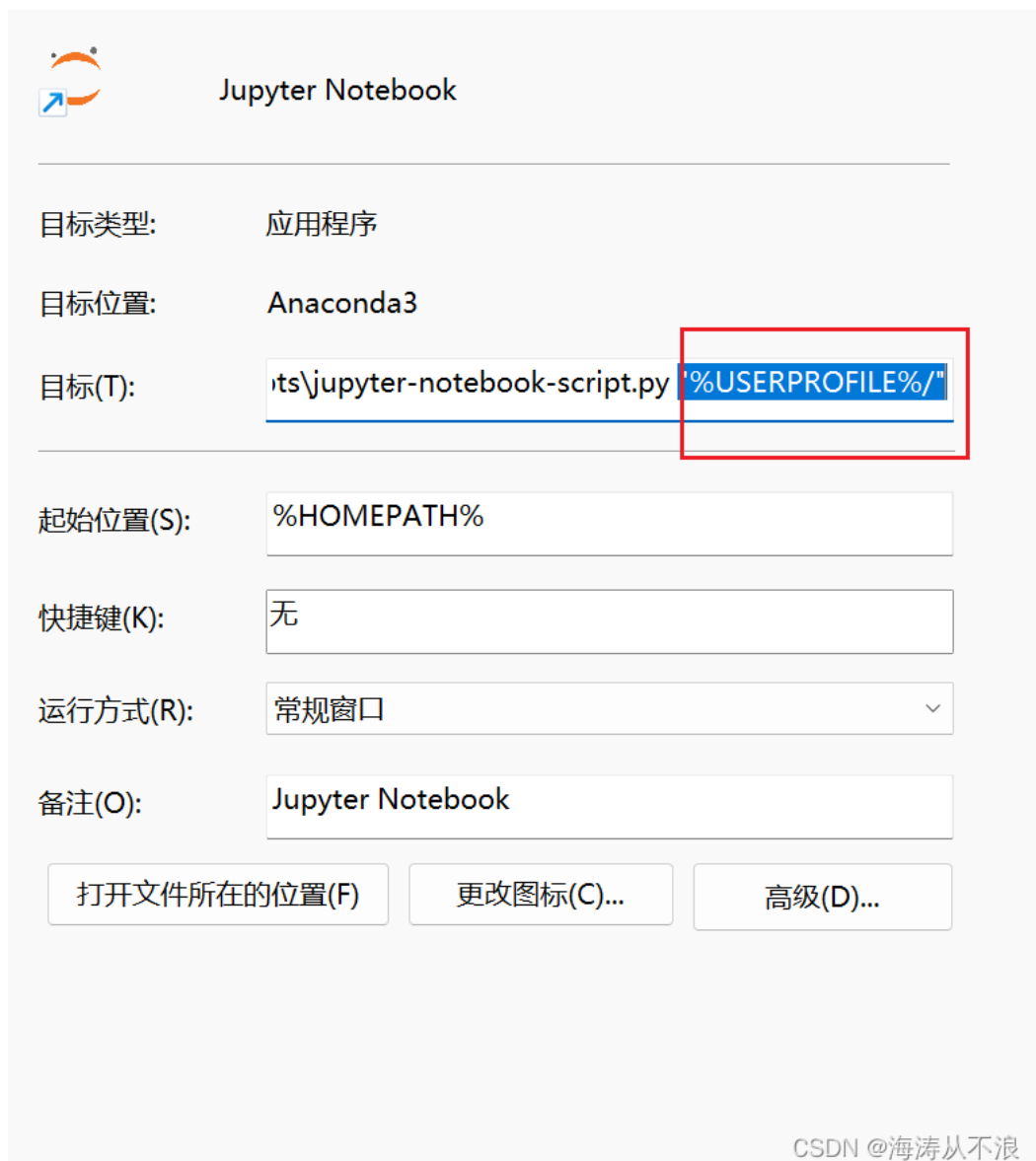
```
# 查看配置文件目录
jupyter notebook --generate-config #
C:\Users\Guocc\.jupyter\jupyter_notebook_config.py
```

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



- 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



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 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 torch --display-
name torch_c
```

- 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

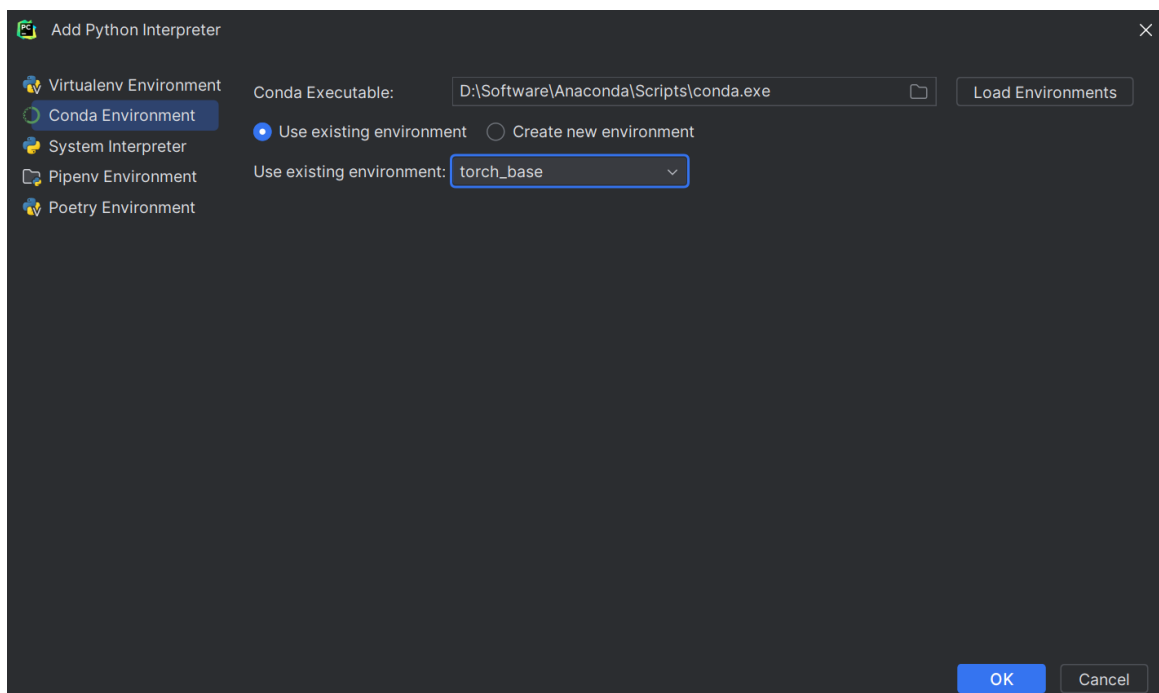
2. Pycharm

- 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）



3. 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
```