前言

首先要搭建好python开发环境，Anaconda 是一个非常不错的环境管理器和包管理器。相比pip 安装第三放模块有时需要安装多个依赖包，Anaconda实在是要方便很多。例如，安装pandas模块适，pip中要依次安装NumPy, dateutil, pytz ,setuptools, 而Anaconda 中只要安装一个pandas包其他依赖包就会自动安装。 conda可以管理不同的运行环境，方便我们在不同版本的 python 间切换使用 。非常好的一项功能就是共享环境。

## 目录

[**目录** 2](#_Toc58662027)

[1. Python 的安装 5](#_Toc58662028)

[1.1 Windwos 5](#_Toc58662029)

[1.2 Linux 5](#_Toc58662030)

[1.2.1 yum更新yum源 5](#_Toc58662031)

[1.2.2 安装Python 5](#_Toc58662032)

[1.2.3 创建python虚拟环境 6](#_Toc58662033)

[2. Anaconda 的安装和下载 7](#_Toc58662034)

[2.1 Windows 安装 7](#_Toc58662035)

[2.2 3.2 Linux下安装： 8](#_Toc58662036)

[2.2.1 下载 8](#_Toc58662037)

[2.2.2 安装 8](#_Toc58662038)

[2.2.3 配置环境变**量** 8](#_Toc58662039)

[3. conda安装及使用 8](#_Toc58662040)

[3.1 conda 的安装 8](#_Toc58662041)

[3.1.1 换国内源 9](#_Toc58662042)

[3.1.2 临时使用镜像 10](#_Toc58662043)

[3.1.3 版本导致的问题 10](#_Toc58662044)

[3.2 Anaconda 的使用 10](#_Toc58662045)

[3.2.1 Anaconda安装后开始菜单配置问题 10](#_Toc58662046)

[3.2.2 管理包 11](#_Toc58662047)

[3.2.3 管理环境 11](#_Toc58662048)

[3.2.4 共享环境 12](#_Toc58662049)

[3.2.5 jupyter notebook 的配置 12](#_Toc58662050)

[3.2.6 matplotlib 中文字体配置 13](#_Toc58662051)

[4. pip的安装使用 14](#_Toc58662052)

[4.1 pip的安装 14](#_Toc58662053)

[4.1.1 windows下 14](#_Toc58662054)

[4.1.2 Linux下 15](#_Toc58662055)

[4.1.3 安装包时临时配置 15](#_Toc58662056)

[4.2 pip 的使用 16](#_Toc58662057)

[4.2.1 安装包 16](#_Toc58662058)

[4.2.2 查看所有版本 16](#_Toc58662059)

[4.2.3 指定版本 16](#_Toc58662060)

[4.2.4 卸载包 16](#_Toc58662061)

[4.2.5 查看安装包信息 16](#_Toc58662062)

[4.2.6 列出所有安装包 16](#_Toc58662063)

[4.2.7 检测更新 17](#_Toc58662064)

[4.2.8 更新包 17](#_Toc58662065)

[4.2.9 环境导出导入 17](#_Toc58662066)

[5. TensorFlow&Pytorch GPU版本安装 17](#_Toc58662067)

[5.1 TesnsorFlow 17](#_Toc58662068)

[5.2 Pytorch 17](#_Toc58662069)

# Python 的安装

## Windwos

如果不想用Anaconda 可以到官网直接下载python

Python 官网下载地址 <https://www.python.org/downloads/>

安装时添加环境变量

安装后确保环境变量中有如下的地址

根据自己的实际路径

D:\programs\python3.8\Scripts\

D:\programs\python3.8\

## Linux

### yum更新yum源

yum update

* 安装Python 3.7所需的依赖否则安装后没有pip3包

yum install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel readline-devel tk-devel libffi-devel gcc make

* 在官网下载所需版本，这里用的是3.8.5版本

wget https://www.python.org/ftp/python/3.8.5/Python-3.8.5.tgz

### 安装Python

* 解压

tar -xvf Python-3.8.5.tgz

* 配置编译

cd Python-3.8.5

* 配置编译的的路径（这里--prefix是指定编译安装的文件夹）

./configure --prefix=/usr/local/python3

* 执行该代码后，会编译安装到 /usr/local/bin/ 下，且不用添加软连接或环境变量

./configure --enable-optimizations

make && make install

* 添加软连接

ln -s /usr/local/python3/bin/python3 /usr/bin/python3

ln -s /usr/local/python3/bin/pip3 /usr/bin/pip3

* 将/usr/local/python3/bin加入PATH

vim /etc/profile

* 然后在文件末尾添加

export PATH=$PATH:/usr/local/python3/bin

* 修改完后，还需要让这个环境变量在配置信息中生效，执行命令

source /etc/profile

### 创建python虚拟环境

* 安装virtualenv

　　　　　　yum install python-virtualenv

* 创建python虚拟环境

　　　　　　virtualenv env

* 执行后，在本地会生成一个与虚拟环境同名的文件夹
* 如果你的系统里安装有不同版本的python，可以使用--python参数指定虚拟环境的python版本：

　　virtualenv --python=/usr/local/python3/bin/python3 env

* 启动虚拟环境

　　　source bin/activate

* 退出虚拟环境

　　　deactivate #

# Anaconda 的安装和下载

## Windows 安装

官网地址：

<https://docs.conda.io/projects/conda/en/latest/user-guide/install/index.html>

官网下载地址：

<https://www.anaconda.com/products/individual>

官网下载慢的同学可移步到清华镜像源：

<https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/>

官网默认的版本一般是稳定版本，大家可以参考官网的版本到清华镜像源下载对应的版本

安装步骤参考附件：Anaconda的下载和使用.pdf

## 3.2 Linux下安装：

### 下载

Wget https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2020.11-Linux-x86\_64.sh

### 安装

bash Anaconda3-2020.11-Linux-x86\_64.sh

### 配置环境变**量**

vi /etc/profile

在文件最后加入如下语句（路径需要根据自己的安装位置更改）

PATH**=**$PATH: /root/anaconda3/bin #路径名跟自己实际情况而定

export PATH

按住shift键+:键，输入wq，保存文件并退出。最后使用如下命令刷新环境变量即可：

source /etc/profile

echo $PATH

### Docker 安装 Anaconda

https://docs.anaconda.com/anaconda/user-guide/tasks/docker/

# conda安装及使用

## conda 的安装

如果用户从来没有使用过conda config 命令，就不会有配置文件，当用户第一次运行 conda config命令时，将会在用户的家目录创建该文件，即一个名为.condarc的文本文件，一般表示 conda 应用程序的配置文件，在用户的家目录之下：

windows：C:\users\username\.condarc

Linux：/home/username/.condarc）

注意：condarc配置文件，是一种可选的（optional）运行期配置文件，其默认情况下是不存在的，但当用户第一次运行 conda config命令时，才会在用户的家目录创建该文件。我可以通过conda config 命令来配置该文件，也完全可以自己手动编辑也可以。

手动编辑

channels:

- http://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/

- <http://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/>

- http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/

- http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/msys2/

ssl\_verify: true

### 换国内源

* 查看源：conda config --show-sources



* 添加清华源：conda config --add channels http://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/



* 是否显示channel的来源

conda config --set show\_channel\_urls yes/no

如果是：conda config --set show\_channel\_urls yes 则配置文件中为 show\_channel\_urls: True。这表示在使用conda search package或者是conda install package的时候会显示这个包是来自于哪一个镜像源。

当然我也可以不显示，则为：conda config --set show\_channel\_urls no 则配置文件中为 show\_channel\_urls: False

* 移除源

conda config --remove channels http://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/

### 临时使用镜像

conda install -c 镜像源 包名

例如：

conda install -c <http://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/> pandas

### 版本导致的问题

Anaconda3 2020.11(Python 3.8.5)之前的版本镜像源使用https 开头

<https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/>

## Anaconda 的使用

### Anaconda安装后开始菜单配置问题

win+R 运行cmd，输入一下命令

conda update menuinst #更新菜单栏 出现 conda install -f console\_shortcut ipython ipython-notebook ipython-qtconsole launcher sp

### 管理包

conda list #查看已安装内容

conda upgrade --all #更新所有包

conda insatll package\_name #安装包

conda search search\_term 模糊查找包 如查找numpy可输入conda search numpy

python setup.py install #进入下载好的第三方库路径下运行

conda isntall numpy=1.10 #指定所需的包版本

conda remove package\_name #卸载包

conda update package\_name #更新包

### 管理环境

* 创建环境和包

conda create -n env\_name package\_names

simple: conda create -n py2 pandas

* 指定版本

conda create -n py2 python=2

* 进入环境
* Windows 上

conda activate my\_env

* OSX/Linux上

source activate my\_env

* 在环境中安装包的命令与前面的一样：

conda install package\_name

* 离开环境
* Windows上

conda deactivate

* OSX/Linux上

source deactivate

### 共享环境

* 当前命令窗口路径下输出环境中所有包的名称

conda env export>environment.yaml

* 进入你的环境，

activate py3

* 使用以下命令更新你的环境

conda env update -f=/path/to/environment.yaml

-f表示你要导出的文件在本地的路径 所以/path/to/environment.yaml要更换成你本地的实际路径

* 列出环境

conda env list

* 删除指定环境

conda env remove -n env\_name

### jupyter notebook 的配置

* conda 环境中安装jupyter noterbook

conda insatll jupyter notebook

* 安装环境自动关联包（目前只支持到python3.5）

conda install nb\_conda

conda install -n python\_env ipykernel

进入notebook

jupyter notebook

* 安装自动补全代码包

conda install pyreadline

jupyter notebook 中使用Tab键

* + 为jupyter notebook 添加目录

conda install -c conda-forge jupyter\_contrib\_nbextensions

方法：运行Jupyter Notebook, 在打开的Notebook界面里, 你会发现多了一个Nbextensions,点击这个tab勾选Table of Contents (有的版本是toc2). 然后创建或者打开一个Jupter Notebook

### matplotlib 中文字体配置

notebook 中输入

import matplotlib

matplotlib.matplotlib\_fname()

查找配置文件路径

在font 字体文件夹中的ttf文件夹中添加SimHei.ttf文件

修改matplotlibrc文件

font.family 去掉注释#

font.sans-serif去掉注释# 添加SimHei

axes.unicode\_minus : False 改为True

删除C：\User\用户名下的matplotlib缓存

重启jupyter Notebook

# pip的安装使用

## pip的安装

### windows下

* 查看当前镜像源

pip config list

* 更换镜像源

pip config set global.index-url <https://pypi.tuna.tsinghua.edu.cn/simple>

pip config set install.trusted-host pypi.tuna.tsinghua.edu.cn

有时候镜像源找不到包可以更换镜像源

pip config set global.index-url <https://pypi.douban.com/simple>

pip config set install.trusted-host pypi.douban.com

网易<https://mirrors.163.com/pypi/simple/>

阿里https://mirrors.aliyun.com/pypi/simple/

* 查看镜像源配置地址以及当前使用的源

pip config -v list

pip使用配置文件的搜索路径优先级是按照上述 list 的 “从下往上” 进行的

如果下面的最下面的pip.ini文件不存在就从倒数第二个开始

命令行默认配置的是类似C:\Users\Nan\AppData\Roaming\pip\pip.ini



* 移除镜像源

pip config unset global.index-url <https://pypi.tuna.tsinghua.edu.cn/simple>

pip config unset install.trusted-host [tuna.tsinghua.edu.cn/simple](https://pypi.tuna.tsinghua.edu.cn/simple)

或按照下面的方法

直接在user目录中创建一个pip目录，再新建文件pip.ini。（例如：C:\Users\WQP\pip\pip.ini）内容同上。

注意：一定要确保系统环境变量中的有pip.ini文件所在的路径

### Linux下

在用户目录之下： ~/.pip/pip.conf (没有就创建一个文件夹及文件。文件夹要加“.”，表示是隐藏文件夹)

添加如下内容：

[global]

iindex-url = https://pypi.douban.com/simple #豆瓣源，可以换成其他的源

trusted-host = pypi.douban.com #添加豆瓣源为可信主机，要不然可能报错

disable-pip-version-check = true #取消pip版本检查，排除每次都报最新的

pip timeout = 120

### 安装包时临时配置

pip install -i 原地址 包名称

pip install -i https://pypi.tuna.tsinghua.edu.cn/simple numpy

常见的源有

豆瓣：http://pypi.douban.com/simple/

清华：<https://pypi.tuna.tsinghua.edu.cn/simple>

## pip 的使用

### 安装包

pip install package\_name

### 查看所有版本

pip install package\_name==

### 指定版本

pip install package\_name ==1.0.4

### 卸载包

pip uninstall package\_name

### 查看安装包信息

pip show package\_name

### 列出所有安装包

pip list

### 检测更新

pip list -outdated

### 更新包

pip install --upgrade package\_name

### 环境导出导入

对于不使用conda的用户，可以使用这段命令 将一个txt文件导出并包括在其中

pip freeze >environment.txt

以管理员身份运行cmd下的命令，安装你刚导出来的environment.txt /path/environment.txt导出的文件在本地的实际路径

pip install -r /path/environment.txt

# TensorFlow&Pytorch 安装

## 显卡配置

TensorFlow和Pytorch GPU 版本安装前还要要根据自己的显卡安装对应的cuda 和CUDNN版本

CUDA是深度学习的sdk CUDNN是神经网络的sdk

另外要注意1.0版本和2.0版本支持的cuda版本也不同

NVIDIA 官网： <https://developer.nvidia.com/cuda-gpus>

NVIDIA 驱动 <https://www.nvidia.com/Download/index.aspx>



**CUDA** 对应的驱动版本：[https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html](https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html#)

安装好驱动后查看驱动版本

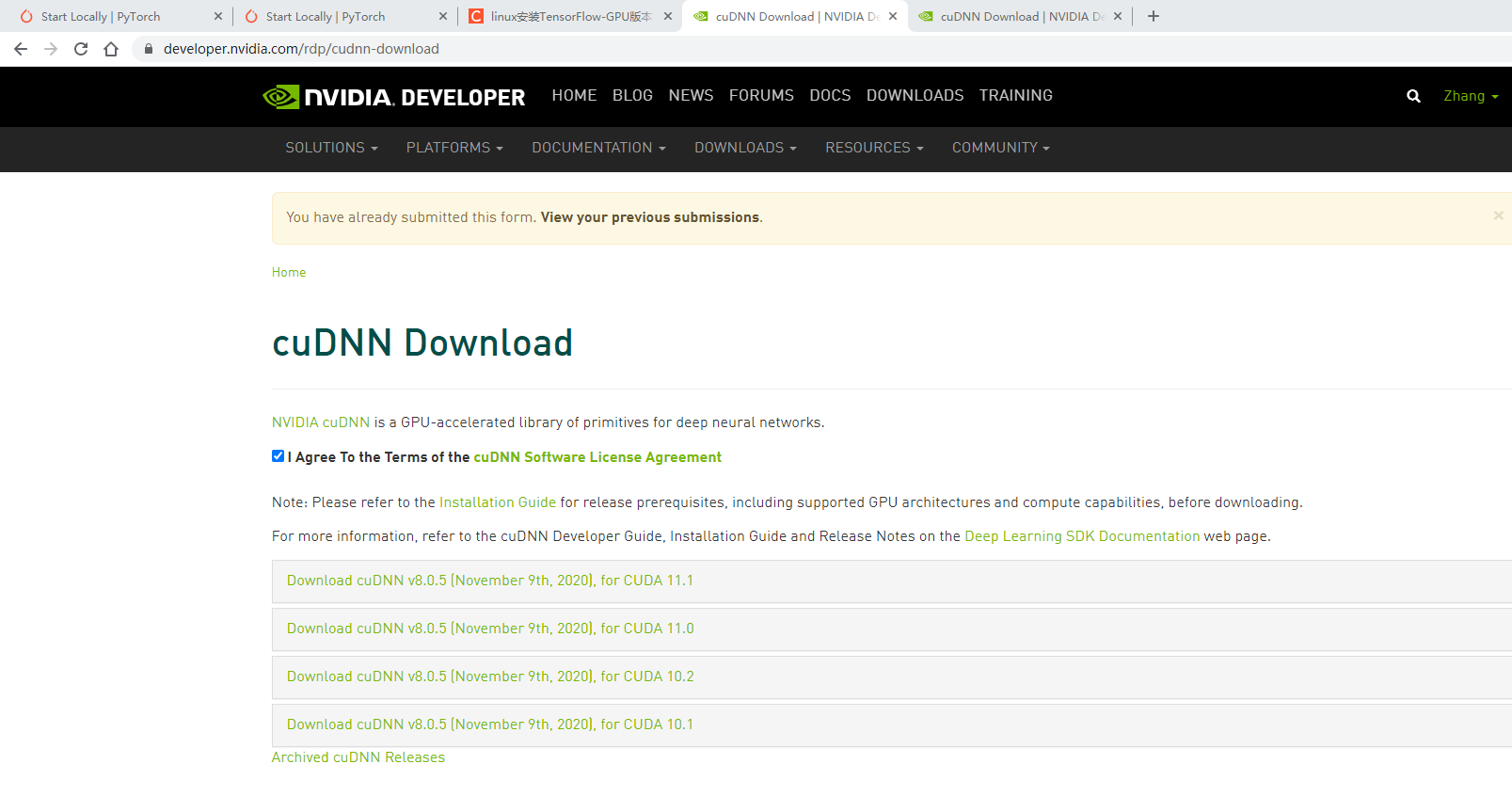


**CUDA Toolkit Archive**: <https://developer.nvidia.com/cuda-toolkit-archive>



**cuDNN** : <https://developer.nvidia.com/rdp/cudnn-download>





解压cuda对应的cudnn



将以上的文件移到你的cuda安装目录下

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.1



添加环境变量

类似的：

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.1

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.1\bin

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.1\lib\x64

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.1\libnvvp



查看GPU状态

nvidia-smi

# 查看cuda 版本

nvcc --version

## TesnsorFlow

官网介绍： <https://www.tensorflow.org/install/pip>



对应版本信息查看<https://tensorflow.google.cn/install/source>





安装好后可在python 编译器里查看是否可以运行

import tensorflow as tf

tf.test.is\_gpu\_available()

返回True 证明成功

## Pytorch

官网介绍：<https://pytorch.org/get-started/locally/>

根据自己的配置选择对用的版本

如果下载慢的话可以直接移步官网用迅雷下载



例如：



安装后python 编译器输入

import torch

print(torch.cuda.is\_available())

返回True 证明成功