

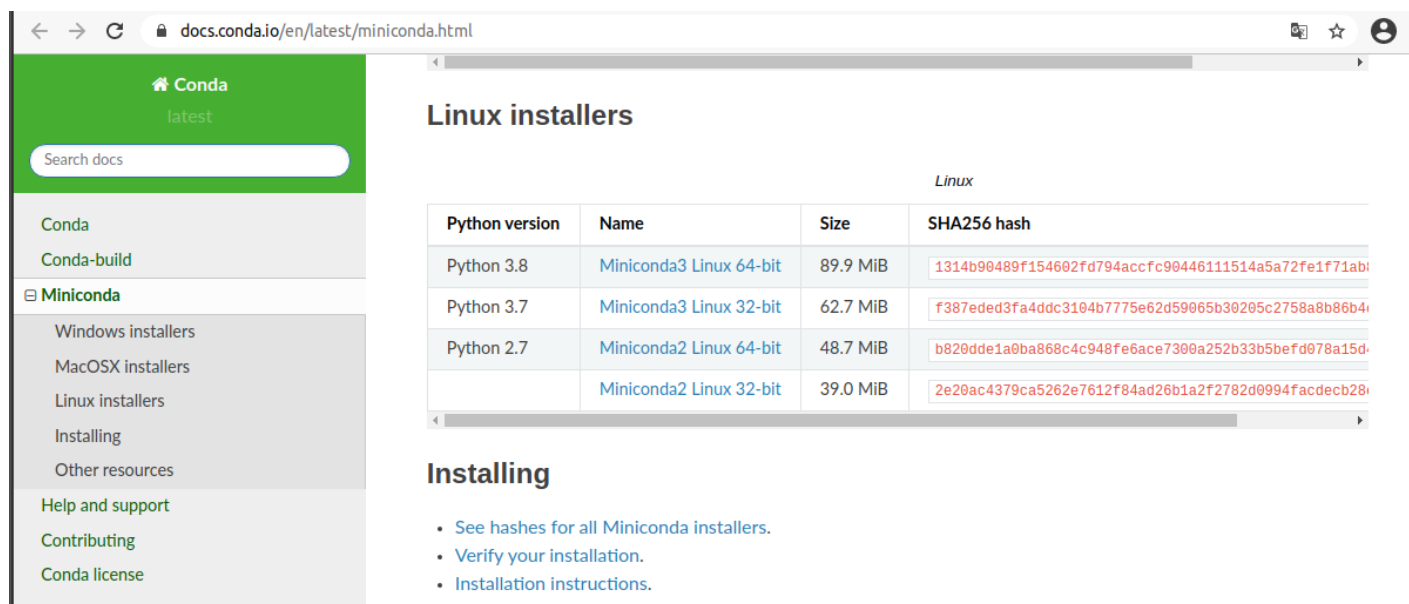
Anaconda + CUDA 基础环境安装

Anaconda 安装配置

安装 conda 可以方便的对我们系统中的 python 不同环境进行操作,避免因为 python 版本等问题导致系统环境混乱.

miniconda 安装

先去 conda 官网下载对应系统的[最新版本 miniconda](#), linux 操作系统下载.sh 的执行文件



The screenshot shows the Conda website's Linux installers page. The left sidebar contains a search bar and a navigation menu with links to Conda, Conda-build, Miniconda, Windows installers, MacOSX installers, Linux installers, Installing, Other resources, Help and support, Contributing, and Conda license. The main content area is titled 'Linux installers' and features a table of installers for different Python versions. Below the table, there is an 'Installing' section with links to see hashes, verify installation, and installation instructions.

Python version	Name	Size	SHA256 hash
Python 3.8	Miniconda3 Linux 64-bit	89.9 MiB	1314b90489f154602fd794accfc90446111514a5a72fe1f71abi
Python 3.7	Miniconda3 Linux 32-bit	62.7 MiB	f387eded3fa4ddc3104b7775e62d59065b30205c2758a8b86b4
Python 2.7	Miniconda2 Linux 64-bit	48.7 MiB	b820dde1a0ba868c4c948fe6ace7300a252b33b5befd078a15d
	Miniconda2 Linux 32-bit	39.0 MiB	2e20ac4379ca5262e7612f84ad26b1a2f2782d0994facdec28

Installing

- [See hashes for all Miniconda installers.](#)
- [Verify your installation.](#)
- [Installation instructions.](#)

命令行模式下给下载的文件执行权限,并执行

```
sudo chmod +x filename.sh
# 执行当前目录下的filename.sh 脚本
./filename.sh
```

conda 环境变量配置

根据提示安装好 conda 后, 检查 `~/.bashrc` 中是否有conda环境,

没有则需要手动把 conda 加入环境变量中即可

linux 环境变量配置的地方有几处如 `~/.bashrc` 和 `~/.profile`, 个人建议装在 `~/.profile` 中配置自己的环境变量。

```
$ echo 'export PATH="$~/miniconda3/bin:$ PATH"' >> ~/.profile
$ source ~/.profile
# 一般conda默认安装在~/minicondaX 里, 我这边修改后的示例为
# export PATH=$PATH:~/software/miniconda3/bin
```

```
# 之后执行 source指令激活修改后的环境变量
source ~/.profile
# 测试是否正常
conda env list
```

conda 镜像加速

为了加速之后 conda 的下载速度，建议修改 conda channel 到国内的镜像源

```
# 首先生成conda配置文件
conda config
```

此时 conda 会在 自己用户目录下生成 `/usr/login_name/.condarc` 也就是 `~/.condarc` 文件
将配置文件里的内容修改为：

```
show_channel_urls: true
channels:
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/
  - https://mirrors.bfsu.edu.cn/anaconda/pkgs/main/
  - https://mirrors.bfsu.edu.cn/anaconda/pkgs/free/
  - http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/msys2/
  - http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge
  - http://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
ssl_verify: true
```

此时 conda 的配置源基本处理结束，可以安装自己想用的内容了

conda 的常用基本命令

"Conda的基本操作"

Conda 里也可以配置 cudatoolkit

CUDA 安装配置

安装 cuda 主要为了 gpu 可以参加深度学习的运算。

NVIDIA 驱动

驱动安装前检测

对于 CUDA 安装首先是要确认机器是否可以安装 CUDA，使用以下命令显示自己的 nvidia 显卡信息，并对比显卡是否对应

```
lspci | grep -i nvidia  
nvidia-smi
```

如果没有任何显示，或者在软件与更新中检查自己是否安装 nvidia 驱动,则更新下自己的驱动



可以直接使用 `software&update` 这个内置应用来更新驱动,也可以直接使用 `apt-get install` 的方式安装 nvidia 驱动,由于比较简单,这里介绍一个比较复杂的方式.

如果你曾经安装过 nvidia 驱动,删除后重新安装驱动

```
sudo apt-get remove --purge nvidia-*
```

```
sudo apt autoremove
```

```
dpkg -l | grep nvidia
```

安装 nvidia 驱动

这里推荐手动下载 nvidia 驱动,[驱动官网](https://www.nvidia.com/download/index.aspx?lang=en-us):<https://www.nvidia.com/download/index.aspx?lang=en-us>



Nvidia 驱动安装需要用到 make 命令

```
sudo apt-get install make
```

下载之后 通过 Ctrl+Alt+F3 切换到文字界面安装驱动, 执行对应的驱动文件即可

```
sudo sh /path/to/driver.run -no-x-check -no-nouveau-check -no-opengl-files
```

只有禁用 opengl 这样安装才不会出现循环登陆的问题

- -no-x-check: 安装驱动时关闭 X 服务
- -no-nouveau-check: 安装驱动时禁用 nouveau
- -no-opengl-files: 只安装驱动文件, 不安装 OpenGL 文件

安装会像这样

NVIDIA Accelerated Graphics Driver for Linux-x86_64 (460.32.03)

An alternate method of installing the NVIDIA driver was detected. (This is usually a package provided by your distributor.) A driver installed via that method may integrate better with your system than a driver installed by nvidia-installer.

Please review the message provided by the maintainer of this alternate installation method and decide how to proceed:

Continue installation

Abort installation

The NVIDIA driver provided by Ubuntu can be installed by launching the "Software & Updates" application, and by selecting the NVIDIA driver from the "Additional Drivers" tab.

Information about the alternate installation method

All

基本全部默认即可,直接正常安装.

如果遇到 gcc 版本检测失败,建议退出重新安装 gcc 以保证编译核心与系统兼容.

安装完毕后,使用命令 `nvidia-smi` 即可看到自己的驱动信息

```
(base) vase@vase:~$ nvidia-smi
```

```
Sat Jan 23 02:10:00 2021
```

```
+-----+
| NVIDIA-SMI 460.32.03      Driver Version: 460.32.03      CUDA Version: 11.2      |
+-----+-----+-----+
| GPU   Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan   Temp   Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                                           MIG M.         |
+-----+-----+-----+
|    0   GeForce GTX 760         Off   | 00000000:0A:00.0 N/A |           N/A       |
| 47%    31C    P8      N/A /  N/A |  47MiB / 1996MiB |      N/A      Default |
|                                           N/A              |
+-----+-----+-----+
```

```
+-----+
| Processes:                                     |
|  GPU   GI    CI          PID    Type    Process name                        GPU Memory |
|          ID    ID                                   Usage                        |
+-----+-----+
| No running processes found |
+-----+
```

更新 nvidia 驱动

当系统更新内核时,可能会导致现在安装的驱动不兼容,需要更新下驱动

```
sudo ubuntu-drivers autoinstall
```

CUDA 安装

安装前准备

一般来说 cuda 均需要 gcc 和 g++ 的支持,所以在执行给定脚本前,先安装好 g++ 和 gcc

```
sudo apt-get install gcc g++
```

值得注意的是 ubuntu20.04 安装的 gcc 默认版本是 9.x, 但是 cuda10.1 并不支持 gcc9.x 版本, 需要自己

->"手动安装低版本的 gcc" <-。

CUDA 脚本下载安装

ubuntu20.04 只支持 cuda11 以上版本 所以目标安装 cuda11.2 最新版本,先是在 nvidia 官网的 [cuda archived](#) 中找到对于的版本, 可以使用 `uname -m && cat /etc/*release` 查看自己的系统信息。



首页 > High Performance Computing > CUDA Toolkit > CUDA Toolkit Archive

CUDA Toolkit Archive

Previous releases of the CUDA Toolkit, GPU Computing SDK, documentation and developer drivers can be found using the links below. Please select the release you want from the list below, and be sure to check www.nvidia.com/drivers for more recent production drivers appropriate for your hardware configuration.

[Download Latest CUDA Toolkit](#)

[Learn More about CUDA Toolkit 11](#)

Latest Release

[CUDA Toolkit 11.2.0](#) (Dec 2020), [Versioned Online Documentation](#)

Archived Releases

[CUDA Toolkit 11.1.1](#) (Oct 2020), [Versioned Online Documentation](#)

[CUDA Toolkit 11.1.0](#) (Sept 2020), [Versioned Online Documentation](#)

[CUDA Toolkit 11.0 Update1](#) (Aug 2020), [Versioned Online Documentation](#)

[CUDA Toolkit 11.0](#) (May 2020), [Versioned Online Documentation](#)

[CUDA Toolkit 10.2](#) (Nov 2019), [Versioned Online Documentation](#)

[CUDA Toolkit 10.1 update2](#) (Aug 2019), [Versioned Online Documentation](#)

并在之后的选择器中选择对于要使用的程序, 建议使用 runfile 本地安装的形式。

选择目标平台

单击指示您目标平台的绿色按钮。系统仅会显示出受支持的平台。下载并使用软件即表示您同意完全遵守 [CUDA EULA](#)

的条款和条件。

操作系统

Linux

Windows

架构

x86_64

ppc64le

sbsa

发行版本

CentOS

Debian

Fedora

OpenSUSE

RHEL

SLES

Ubuntu

WSL-Ubuntu

版本

20.04

18.04

16.04

安装程序类型

runfile (local)

deb (local)

deb (network)

Download Installer for Linux Ubuntu 20.04 x86_64

The base installer is available for download below.

基本安装程序

安装说明:

```
$ wget https://developer.download.nvidia.com/compute/cuda/11.1.0/local_installers/cuda_11.1.0_455.23.05_linux.run
$ sudo sh cuda_11.1.0_455.23.05_linux.run
```

之后就是执行对应的安装脚本了。

```
wget https://developer.download.nvidia.com/compute/cuda/11.1.0/local_installers/cuda_11.1.0_455.23.05_linux.run
sudo sh cuda_11.1.0_455.23.05_linux.run
```

wget 可以使用自己的节点加速下载

使用 `wget -e "http_proxy=http://127.0.0.1"` 下载地址即可

接受 cuda 的 license

```
End User License Agreement
-----

Preface
-----

The Software License Agreement in Chapter 1 and the Supplement
in Chapter 2 contain license terms and conditions that govern
the use of NVIDIA software. By accepting this agreement, you
agree to comply with all the terms and conditions applicable
to the product(s) included herein.

NVIDIA Driver

Description

This package contains the operating system driver and

Do you accept the above EULA? (accept/decline/quit):
accept
```

这里选择号自己要安装的内容,CUDA Toolkit 为必须安装内容,其他的根据自己选择进行安装,一般桌面版的 ubuntu 第一个驱动都不装(因为自己以及装过了)

```
CUDA Installer
- [ ] Driver
  [ ] 418.87.00
+ [X] CUDA Toolkit 10.1
  [X] CUDA Samples 10.1
  [X] CUDA Demo Suite 10.1
  [X] CUDA Documentation 10.1
Options
Install

Up/Down: Move | Left/Right: Expand | 'Enter': Select | 'A': Advanced options
```

[其中 cuda 安装所作的主要是以下的工作:](#)

- 安装 cuda toolkit 到/usr/local/cuda-version
- 创建软连接到/usr/local/cuda

- 安装 cuda 示例到/usr/local/cuda/samples 和 home 文件夹下的 NVIDIA_CUDA-10.1_Samples
- 添加 cuda 的 library path

看到以下则表示安装正常:

```
=====
= Summary =
=====

Driver:    Not Selected
Toolkit:   Installed in /usr/local/cuda-10.1/
Samples:   Not Selected

Please make sure that
- PATH includes /usr/local/cuda-10.1/bin
- LD_LIBRARY_PATH includes /usr/local/cuda-10.1/lib64, or, add /usr/local/cuda-10.1/lib64 to /etc/ld.so.conf and run ldconfig as root

To uninstall the CUDA Toolkit, run cuda-uninstaller in /usr/local/cuda-10.1/bin

Please see CUDA_Installation_Guide_Linux.pdf in /usr/local/cuda-10.1/doc/pdf for detailed information on setting up CUDA.

***WARNING: Incomplete installation! This installation did not install the CUDA Driver. A driver of version at least 418.00 is required for CUDA 10.1 functionality to work.

To install the driver using this installer, run the following command, replacing <CudaInstaller> with the name of this run file:
    sudo <CudaInstaller>.run --silent --driver
```

配置一下环境变量

```
# CUDA 基础命令
echo "export PATH=\"\$PATH:/usr/local/cuda/bin\"" >> ~/.profile

# 一些深度学习框架需要库路径以便找到cuda库 如mxnet
# 如果后续安装cuDNN库,这个路径也是必不可少的
echo "export LD_LIBRARY_PATH=\"/usr/local/cuda/lib64\"" >> ~/.profile
source ~/.profile
```

之后使用 `ncv -V` 检查下 CUDA 环境即可

```
(base) vase@vase:~/software/YesPlayMusic$ nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Sun_Jul_28_19:07:16_PDT_2019
Cuda compilation tools, release 10.1, V10.1.243
```

强制安装低版本 CUDA

正常是无法安装的,为此我们需要先将 gcc,g++ 降级,才可以正常的运行,值得注意的是,在安装 cuda10.1 时,由于未知原因,在安装前要指定好 cuda 的库文件 即通过 `--`

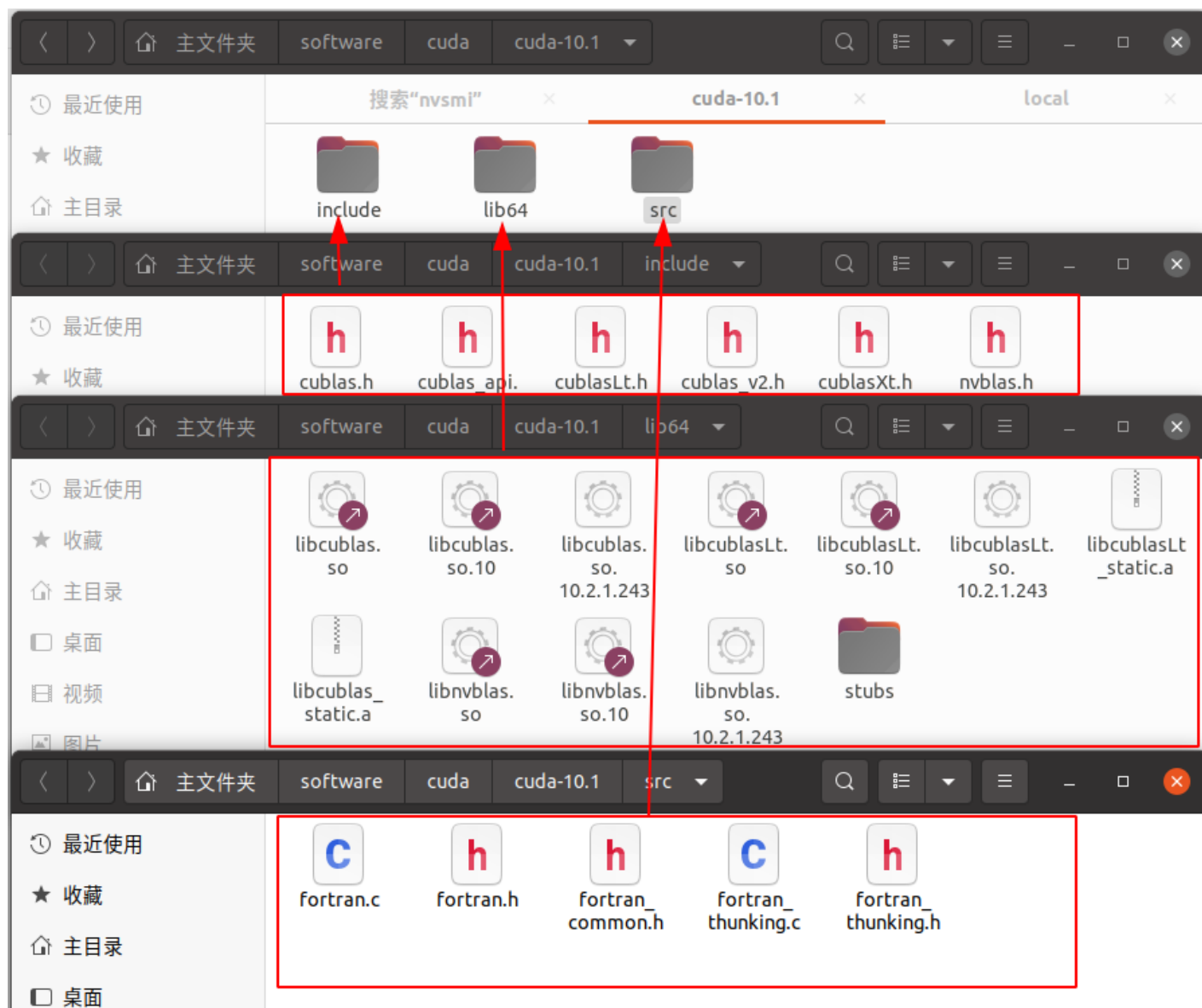
```
librarypath=/usr/local/cuda-10.1
```

即运行指令

```
sudo sh /path/to/cuda_10.1.*.run --librarypath=/usr/local/cuda-10.1
```

这里更改库地址为其他并不会影响 cuda 的安装,但是建议和 cuda 主文件安装在一起

cuda 的库文件如下:



测试

简单在 conda 中安装测试

```
conda create -n gluon python=3.6 jupyter matplotlib pandas  
pip install mxnet-cu101 d2lzh=1.0.0
```

测试内容

```
Python 3.6.12 |Anaconda, Inc.| (default, Sep 8 2020, 23:10:56)  
[GCC 7.3.0] on linux
```

```
Type "help", "copyright", "credits" or "license" for more information.
>>> import mxnet
>>> mxnet.gpu
<function gpu at 0x7f3339576730>
>>>
```

CUDA 卸载

对于 cuda 卸载官方给出过脚本，只需要执行对于 cuda 文件中的 sh 文件即可

一般为 cuda 目录下的 bin 文件夹里的 cuda-uninstaller,即运行

```
cd /usr/local/conda/bin

sudo sh ./cuda-uninstaller
```

对于一些低版本 cuda 可能用的.pl 执行文件,具体为

```
# x.x 为自己的具体版本
sudo /usr/local/cuda-x.x/bin/uninstall_cuda_x.x.pl
```

```
CUDA Uninstaller
[X] CUDA_Demo_Suite_10.1
[X] CUDA_Documentation_10.1
[X] CUDA_Samples_10.1
[X] CUDA_Toolkit_10.1
Done
```

```
Up/Down: Move | 'Enter': Select
```

检查是否卸载干净直接去/usr/local 看看有无残留 cuda 文件即可