

Centos 系统安装 tensorflow-gpu

张桐 | | 2019.7.24

手册中出现的\$开头的命令都是在终端进行

python3 安装 (python 版本 3.5.0)

安装

安装必要工具 yum-utils, 它的功能是管理 repository 及扩展包的工具 (主要是针对 repository)

```
$ sudo yum install yum-utils
```

使用 yum-builddep 为 Python3 构建环境, 安装缺失的软件依赖, 使用下面的命令会自动处理。

```
$ sudo yum-builddep python
```

完成后下载 Python3 的源码包, Python 源码包目录:

```
https://www.python.org/ftp/python/
```

```
$ curl -O https://www.python.org/ftp/python/3.5.0/Python-3.5.0.tgz
```

最后一步, 编译安装 Python3, 默认的安装目录是 /usr/local 如果你要改成其他目录可以在编译(make)前使用 configure 命令后面追加参数 "-prefix=/alternative/path" 来完成修改。

```
$ tar xf Python-3.5.0.tgz
```

```
$ cd Python-3.5.0
```

```
$ ./configure
```

```
$ make
```

```
$ sudo make install
```

查看版本

输入 python3, 出现如图信息即为安装成功

```
[root@localhost ~]# python3
Python 3.5.0 (default, Jul 22 2019, 16:06:17)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-36)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

手册中出现的\$开头的命令都是在终端进行

显卡驱动安装 (GeForce RTX 2080 驱动版本 430.34)

查看驱动版本

安装 nvidia-detect 命令,从 ELRepo 源安装

```
$ rpm --import https://www.elrepo.org/RPM-GPG-KEY-elrepo.org
```

```
$ rpm -Uvh http://www.elrepo.org/elrepo-release-7.0-2.el7.elrepo.noarch.rpm
```

安装显卡驱动检查

```
$ yum install nvidia-detect
```

安装完成运行 nvidia-detect -v 查看

```
[root@localhost ~]# nvidia-detect -v
Probing for supported NVIDIA devices...
[10de:1e82] NVIDIA Corporation Device 1e82
This device requires the current 430.34 NVIDIA driver kmod-nvidia
WARNING: Xorg log file /var/log/Xorg.0.log does not exist
WARNING: Unable to determine Xorg ABI compatibility
WARNING: The driver for this device does not support the current Xorg version
```

没找到 GTX2080 不重要, 自己知道显卡是什么就好。没有搜到 1e82 是什么。

注意驱动版本是 430.34 需要下载对应的显卡驱动

下载安装显卡驱动

显卡驱动下载地址

<https://www.geforce.cn/drivers>



手册中出现的\$开头的命令都是在终端进行



下载的同时做如下准备工作

安装前的准备工作

(前两句我没运行)

```
$ yum -y update //注意这是升级系统比较慢
```

```
$ yum -y groupinstall "GNOME Desktop" "Development Tools"
```

```
$ sudo yum install "kernel-devel-uname-r == $(uname -r)"
```

```
$ yum -y install epel-release
```

```
$ yum -y install dkms
```

编辑 grub 文件

```
$ vi /etc/default/grub
```

在"GRUB_CMDLINE_LINUX"中添加

手册中出现的\$开头的命令都是在终端进行

rd.driver.blacklist=nouveau nouveau.modeset=0

```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet rd.driver.blacklist=nouveau nouveau.modeset=0"
GRUB_DISABLE_RECOVERY="true"
```

随后生成配置

\$ grub2-mkconfig -o /boot/grub2/grub.cfg

创建 blacklist

\$ vi /etc/modprobe.d/blacklist.conf

添加 blacklist nouveau (添加到文件内就行，哪行都行)

更新配置

\$ mv /boot/initramfs-\$(uname -r).img /boot/initramfs-\$(uname -r)-nouveau.img

\$ dracut /boot/initramfs-\$(uname -r).img \$(uname -r)

重启系统

\$ shutdown -r now

确认禁用了 nouveau

\$ lsmod | grep nouveau

若无输出则禁用成功

驱动下载完成，传到服务器上开始安装（sh 后边跟的参数为下载的驱动名称 注意：

我下载的这个驱动是 430.34 版本，跟上文中查看的驱动版本对应）

\$ sh NVIDIA-Linux-x86_64-430.34.run

安装完成之后重启

手册中出现的\$开头的命令都是在终端进行

查看安装结果

\$ nvidia-smi

显示如下信息，安装显卡驱动成功

```
[root@localhost ~]# nvidia-smi
Wed Jul 24 10:34:52 2019

+-----+
| NVIDIA-SMI 418.67                Driver Version: 418.67          CUDA Version: 10.1     |
+-----+-----+
| GPU Name      Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+
| 0  GeForce RTX 2080    Off | 00000000:65:00:0 Off |                  N/A |
| 36%   37C   P0      N/A /  N/A |    0MiB /  7951MiB |      0%      Default  |
+-----+-----+

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

CUDA、CUDNN 安装 (CUDA 10.1,CUDNN 7.4.2)

版本对照关系确认以及下载

CUDA 下载 <https://developer.nvidia.com/cuda-toolkit-archive>

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

由于 CUDA 目前最高支持 418.67。因此下载驱动版本为 418.67 的 CUDA 10.1 版本

The screenshot shows the NVIDIA CUDA Toolkit Archive website. The 'Select Target Platform' section is highlighted with a red box. The 'Operating System' is set to Linux, 'Architecture' to x86_64, 'Distribution' to CentOS, and 'Version' to 7. The 'Installer Type' is set to rpm (network). Below this, the 'Download Installer for Linux CentOS 7 x86_64' section is shown. The 'Base Installer' is selected, and the 'Download [2.5 GB]' button is highlighted with a red arrow. A red handwritten note '418.67' is written next to the 'Base Installer' button. The installation instructions are also visible.

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, ppc64le

Distribution: Fedora, OpenSUSE, RHEL, CentOS, SLES, Ubuntu

Version: 7, 6

Installer Type ⓘ: runfile (local), rpm (local), rpm (network)

Download Installer for Linux CentOS 7 x86_64

The base installer is available for download below.

Base Installer: Download [2.5 GB] ⓘ

Installation Instructions:

- Run `sudo sh cuda_10.1.168_418.67_linux.run`
- Follow the command-line prompts

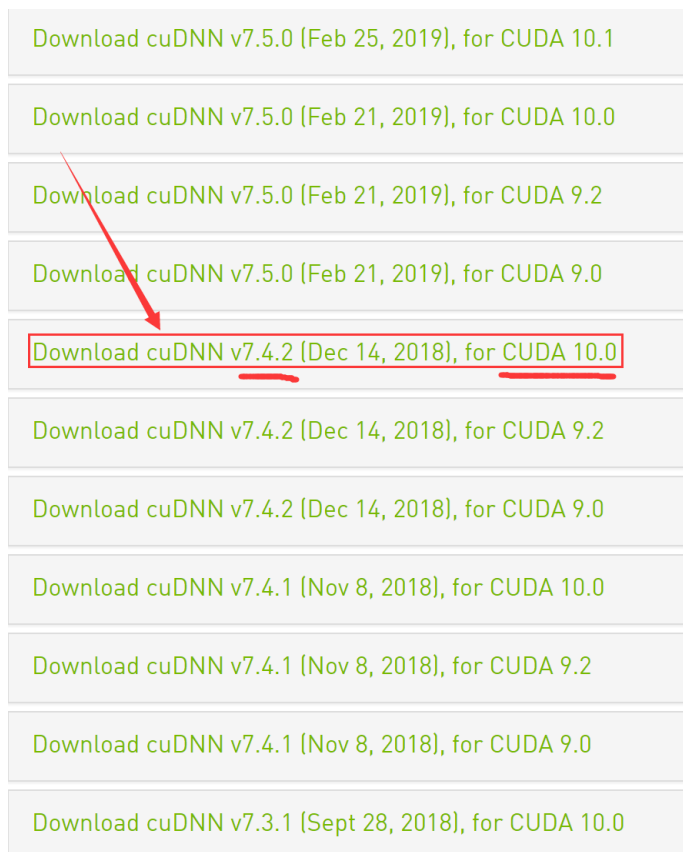
手册中出现的\$开头的命令都是在终端进行

根据 tensorflow 官网[经过测试的构建配置](https://www.tensorflow.org/install/source)对比选择 CUDNN 版本 7.4

<https://tensorflow.google.cn/install/source>

版本	Python 版本	编译器	编译工具	cuDNN	CUDA
tensorflow_gpu-1.13.1	2.7、3.3-3.6	GCC 4.8	Bazel 0.19.2	7.4	10.0
tensorflow_gpu-1.12.0	2.7、3.3-3.6	GCC 4.8	Bazel 0.15.0	7	9
tensorflow_gpu-1.11.0	2.7、3.3-3.6	GCC 4.8	Bazel 0.15.0	7	9
tensorflow_gpu-1.10.0	2.7、3.3-3.6	GCC 4.8	Bazel 0.15.0	7	9
tensorflow_gpu-1.9.0	2.7、3.3-3.6	GCC 4.8	Bazel 0.11.0	7	9
tensorflow_gpu-1.8.0	2.7、3.3-3.6	GCC 4.8	Bazel 0.10.0	7	9
tensorflow_gpu-1.7.0	2.7、3.3-3.6	GCC 4.8	Bazel 0.9.0	7	9
tensorflow_gpu-1.6.0	2.7、3.3-3.6	GCC 4.8	Bazel 0.9.0	7	9
tensorflow_gpu-1.5.0	2.7、3.3-3.6	GCC 4.8	Bazel 0.8.0	7	9
tensorflow_gpu-1.4.0	2.7、3.3-3.6	GCC 4.8	Bazel 0.5.4	6	8
tensorflow_gpu-1.3.0	2.7、3.3-3.6	GCC 4.8	Bazel 0.4.5	6	8
tensorflow_gpu-1.2.0	2.7、3.3-3.6	GCC 4.8	Bazel 0.4.5	5.1	8
tensorflow_gpu-1.1.0	2.7、3.3-3.6	GCC 4.8	Bazel 0.4.2	5.1	8
tensorflow_gpu-1.0.0	2.7、3.3-3.6	GCC 4.8	Bazel 0.4.2	5.1	8

CUDNN 下载界面如图（选择 7.4.1 或 7.4.2 都可以，我选择的是 7.4.2）



手册中出现的\$开头的命令都是在终端进行

选择 CUDNN Library for Linux

[Download cuDNN v7.4.2 \(Dec 14, 2018\), for CUDA 10.0](#)

Library for Windows, Mac, Linux, Ubuntu and RedHat/Centos(x86_64 architecture)

[cuDNN Library for Windows 7](#)

[cuDNN Library for Windows 10](#)

[cuDNN Library for Linux](#)

[cuDNN Library for OSX](#)

[cuDNN Runtime Library for Ubuntu18.04 \(Deb\)](#)

[cuDNN Developer Library for Ubuntu18.04 \(Deb\)](#)

[cuDNN Code Samples and User Guide for Ubuntu18.04 \(Deb\)](#)

[cuDNN Runtime Library for Ubuntu16.04 \(Deb\)](#)

[cuDNN Developer Library for Ubuntu16.04 \(Deb\)](#)

[cuDNN Code Samples and User Guide for Ubuntu16.04 \(Deb\)](#)

[cuDNN Runtime Library for Ubuntu14.04 \(Deb\)](#)

[cuDNN Developer Library for Ubuntu14.04 \(Deb\)](#)

[cuDNN Code Samples and User Guide for Ubuntu14.04 \(Deb\)](#)

[cuDNN Runtime Library for RedHat/Centos 7.3 \(RPM\)](#)

[cuDNN Developer Library for RedHat/Centos 7.3 \(RPM\)](#)

[cuDNN Code Samples and User Guide for RedHat/Centos 7.3 \(RPM\)](#)

CUDA 安装

将下载的安装包上传到 Centos 服务器，使用命令进行安装

```
$ sh cuda_10.1.168_418.67_linux.run
```

按照命令行提示操作

Accept 然后 都是默认 选择 install 等待结果

安装成功如图（这张图网上找的 类似界面为安装成功）

```
Installing the CUDA Toolkit in /usr/local/cuda-9.0 ...

=====
= Summary =
=====

Driver:   Not Selected
Toolkit:  Installed in /usr/local/cuda-9.0
Samples:  Not Selected

Please make sure that
- PATH includes /usr/local/cuda-9.0/bin
- LD_LIBRARY_PATH includes /usr/local/cuda-9.0/lib64, or, add /usr/local/cuda-9.0/lib64 to /etc/ld.so.conf

To uninstall the CUDA Toolkit, run the uninstall script in /usr/local/cuda-9.0/bin

Please see CUDA_Installation_Guide_Linux.pdf in /usr/local/cuda-9.0/doc/pdf for detailed information

***WARNING: Incomplete installation! This installation did not install the CUDA Driver. A driver of
To install the driver using this installer, run the following command, replacing <CudaInstaller> with
    sudo <CudaInstaller>.run -silent -driver

Logfile is /tmp/cuda_install_9979.log
```


手册中出现的\$开头的命令都是在终端进行

配置环境变量

```
$ vi /etc/profile
```

末尾添加

```
export PATH=/usr/local/ cuda-10.1/bin:$PATH export
```

```
LD_LIBRARY_PATH=/usr/local/ cuda-10.1/lib64:$LD_LIBRARY_PATH
```

其中 cuda-10.1 的文件根据实际位置的文件名决定

使环境变量立即生效 或者 重启系统

```
$ source /etc/profile
```

CUDA 验证

执行以下命令验证是否安装成功

```
$ /usr/local/cuda-10.1/extras/demo_suite/deviceQuery
```

其中 cuda-10.1 的文件根据实际位置的文件名决定

```
[root@localhost ~]# /usr/local/cuda-10.1/extras/demo_suite/deviceQuery
/usr/local/cuda-10.1/extras/demo_suite/deviceQuery Starting...

  CUDA Device Query (Runtime API) version (CUDA static linking)
Detected 1 CUDA Capable device(s)

Device 0: "GeForce RTX 2080"
  CUDA Driver Version / Runtime Version      10.1 / 10.1
  CUDA Capability Major/Minor version number: 7.5
  Total amount of global memory:             7952 MBytes (8338210816 bytes)
  (46) Multiprocessors, ( 64) CUDA Cores/MP: 2944 CUDA Cores
  GPU Max Clock rate:                        1710 MHz (1.71 GHz)
  Memory Clock rate:                         7000 Mhz
  Memory Bus Width:                          256-bit
  L2 Cache Size:                             4194304 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:            65536 bytes
  Total amount of shared memory per block:    49152 bytes
  Total number of registers available per block: 65536
  Warp size:                                 32
  Maximum number of threads per multiprocessor: 1024
  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:                      2147483647 bytes
  Texture alignment:                         512 bytes
  Concurrent copy and kernel execution:      Yes with 3 copy engine(s)
  Run time limit on kernels:                  No
  Integrated GPU sharing Host Memory:         No
  Support host page-locked memory mapping:   Yes
  Alignment requirement for Surfaces:        Yes
  Device has ECC support:                     Disabled
  Device supports Unified Addressing (UVA):   Yes
  Device supports Compute Preemption:        Yes
  Supports Cooperative Kernel Launch:        Yes
  Supports MultiDevice Co-op Kernel Launch:  Yes
  Device PCI Domain ID / Bus ID / location ID: 0 / 101 / 0
  Compute Mode:
    < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 10.1, CUDA Runtime Version = 10.1, NumDevs = 1, Device0 = GeForce RTX 2080
Result = PASS
```

CUDNN 安装

cuDNN 的安装，只需要将压缩包解压，并把文件覆盖到 CUDA 对应的目录中去即可

手册中出现的\$开头的命令都是在终端进行

```
$ tar -zxvf cudnn-10.0-linux-x64-v7.4.2.24.solitairetheme8
```

```
$ cp cuda/include/cudnn.h /usr/local/cuda/include/
```

```
$ cp cuda/lib64/libcudnn* /usr/local/cuda/lib64/
```

Tensorflow-gpu 安装（最终安装版本 1.14.0）

对比上节对照表中 tensorflow 的版本应该为 1.13.0，实际上 1.13.0 并不兼容 CUDA10.1 与 CUDNN7.4.2，兼容版本是 10 与 7.4。因此提升 tensorflow 版本到 1.14.0，成功兼容。

首次安装的 python3 需要先升级 pip

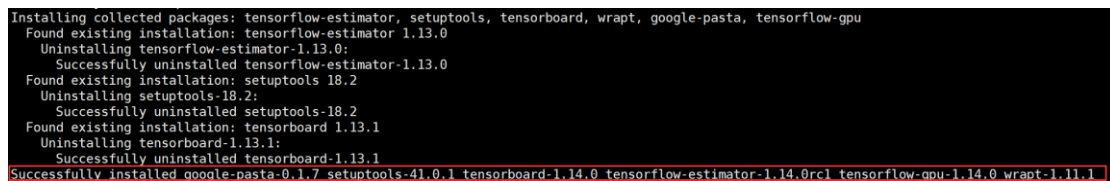
默认 centos 系统自带 python2，因此给 python3 安装库需要使用 pip3，给 python2 安装使用 pip

```
$ pip3 install --upgrade pip
```

安装 tensorflow

```
$ pip3 install tensorflow-gpu==1.14.0 -i  
https://pypi.tuna.tsinghua.edu.cn/simple/
```

安装成功末尾如下图



```
Installing collected packages: tensorflow-estimator, setuptools, tensorboard, wrapt, google-pasta, tensorflow-gpu  
Found existing installation: tensorflow-estimator 1.13.0  
Uninstalling tensorflow-estimator-1.13.0:  
Successfully uninstalled tensorflow-estimator-1.13.0  
Found existing installation: setuptools 18.2  
Uninstalling setuptools-18.2:  
Successfully uninstalled setuptools-18.2  
Found existing installation: tensorboard 1.13.1  
Uninstalling tensorboard-1.13.1:  
Successfully uninstalled tensorboard-1.13.1  
Successfully installed google-pasta-0.1.7 setuptools-41.0.1 tensorboard-1.14.0 tensorflow-estimator-1.14.0rc1 tensorflow-gpu-1.14.0 wrapt-1.11.1
```

测试 tensorflow-gpu 能否使用

进入 python3.5

```
$ python3
```

导入 tensorflow

```
$ import tensorflow
```

手册中出现的\$开头的命令都是在终端进行

不报错即为安装成功

安装 tensorflow 高级库 keras

```
$ pip3 install keras -i https://pypi.tuna.tsinghua.edu.cn/simple/
```

所有安装完成