

Install TensorFlow gpu on ubuntu18.04



Dhanoop Karunakaran

Follow

Aug 27, 2018 · 3 min read



I wrote this post to simplify the installation process of gpu version of TensorFlow on Ubuntu 18.04. I have described installation steps below and hope that it will be helpful.

NVIDIA driver

Check the nvidia driver is installed by running the command 'nvidia-smi'

Normally, the command will be installed as part of the driver and expected to get the result below.

```

artag@artag:~/untreatedengne_4.19$ nvidia-smi
Mon Aug 27 14:49:09 2018

+-----+
| NVIDIA-SMI 390.77              Driver Version: 390.77          |
+-----+-----+
| 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 GTX 1080    Off   | 00000000:01:00.0  On   |          N/A         |
| 30%   44C   P8      11W / 180W | 569MiB / 8116MiB |      0%      Default  |
+-----+-----+

+-----+
| Processes:                                                       GPU Memory |
|  GPU       PID    Type    Process name                     Usage    |
|====+=====+
| 0          1207    G       /usr/lib/xorg/Xorg                    40MiB   |
| 0          1592    G       /usr/bin/gnome-shell                  49MiB   |
| 0          1773    G       /usr/lib/xorg/Xorg                    269MiB  |
| 0          1917    G       /usr/bin/gnome-shell                  199MiB  |
| 0          2366    G       /usr/lib/firefox/firefox               2MiB   |
| 0          2533    G       /usr/lib/firefox/firefox               2MiB   |
| 0          2584    G       /usr/lib/firefox/firefox               2MiB   |
+-----+

```

If you are not able to see the result like above picture, it may be due to either nvidia driver is not installed or nvidia-smi utility is missing. I had to run the below to install the nvidia driver and was successful with installation.

Only follow below steps if you can't get information like above.

1. Run the below code to uninstall the existing nvidia driver and restart.

```

sudo apt-get purge nvidia*
sudo reboot

```

2. Run the below code to install the nvidia driver 390 at the time of this post writing.

```

sudo apt-get install nvidia-390

```

After these steps, you have installed the nvidia driver and 'nvidia-smi' commad should give you some information about the GPU.

CUDA 9.0 installation

1. Download cuda9.0 installation file from [nvidia site](#).

CUDA Toolkit 9.0 Downloads

Select Target Platform ⓘ
Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System

WindowsLinuxMac OSX

Architecture ⓘ

x86_64ppc64le

Distribution

FedoraOpenSUSERHELCentOSSLESUbuntu

Version

17.0416.04

Installer Type ⓘ

runfile (local)deb (local)deb (network)

Download Installers for Linux Ubuntu 17.04 x86_64
The base installer is available for download below.
There are 4 patches available. These patches require the base installer to be installed first.

> Base Installer

Download [1.6 GB] ⬇

Installation Instructions:
1. Run `sudo sh cuda_9.0.176_384.81_linux.run`
2. Follow the command-line prompts

2. Change the permission of the downloaded file

```
sudo chmod +x cuda_9.0.176_384.81_linux.run
```

3. The run the file

```
./cuda_9.0.176_384.81_linux.run
```

Accept the terms and condition and make sure you type 'no' to the accelerated graphics driver installation question. Also make sure type 'yes' to symoblic link question.

```
You are attempting to install on an unsupported configuration. Do you wish to continue?
(y)es/(n)o [ default is no ]: yes

Install NVIDIA Accelerated Graphics Driver for Linux-x86_64 384.81?
(y)es/(n)o/(q)uit: no

Install the CUDA 9.0 Toolkit?
(y)es/(n)o/(q)uit: yes

Enter Toolkit Location
[ default is /usr/local/cuda-9.0 ]:

/usr/local/cuda-9.0 is not writable.
Do you wish to run the installation with 'sudo'?
(y)es/(n)o: yes

Please enter your password:
Do you want to install a symbolic link at /usr/local/cuda?
(y)es/(n)o/(q)uit: yes

Install the CUDA 9.0 Samples?
(y)es/(n)o/(q)uit: yes
```

CUDANN 7.2.1 installation

Next, Download the file from [nvidia site](#).

cuDNN Download

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

☒ I Agree To the Terms of the [cuDNN Software License Agreement](#)

Note: Please refer to the [Installation Guide](#) for release prerequisites, including supported GPU architectures and compute capabilities, before downloading.

For more information, refer to the cuDNN Developer Guide, Installation Guide and Release Notes on the [Deep Learning SDK Documentation](#) web page.

[Download cuDNN v7.2.1 \(August 7, 2018\), for CUDA 9.2](#)

[Download cuDNN v7.2.1 \(August 7, 2018\), for CUDA 9.0](#)

[cuDNN v7.2.1 Library for Linux](#)

[cuDNN v7.2.1 Library for Linux \(Power8\)](#)

[cuDNN v7.2.1 Library for Windows 7](#)

[cuDNN v7.2.1 Library for Windows 10](#)

[cuDNN v7.2.1 Runtime Library for Ubuntu16.04 \(Deb\)](#)

[cuDNN v7.2.1 Developer Library for Ubuntu16.04 \(Deb\)](#)

[cuDNN v7.2.1 Code Samples and User Guide for Ubuntu16.04 \(Deb\)](#)

[cuDNN v7.2.1 Runtime Library for Ubuntu16.04 & Power8 \(Deb\)](#)

[cuDNN v7.2.1 Developer Library for Ubuntu16.04 & Power8 \(Deb\)](#)

[cuDNN v7.2.1 Code Samples and User Guide for Ubuntu16.04 & Power8 \(Deb\)](#)

[cuDNN v7.2.1 Runtime Library for Ubuntu14.04 \(Deb\)](#)

[cuDNN v7.2.1 Developer Library for Ubuntu14.04 \(Deb\)](#)

[cuDNN v7.2.1 Code Samples and User Guide for Ubuntu14.04 \(Deb\)](#)

[Download cuDNN v7.2.1 \(August 7, 2018\), for CUDA 8.0](#)

[Archived cuDNN Releases](#)

Please make sure you have downloaded the first option *cuDNN v7.2.1 Library for Linux*.

Once you have downloaded, follow the below steps:

1. Extract and copy

```
sudo tar -xzvf
sudo cp cuda/include/cudnn.h /usr/local/cuda/include
sudo cp cuda/lib64/libcudnn* /usr/local/cuda/lib64
sudo chmod a+r /usr/local/cuda/include/cudnn.h
sudo chmod a+r /usr/local/cuda/lib64/libcudnn*
```

2) Copy below commands to ~/.bashrc

```
export
LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/usr/local/cuda/lib64:/usr
/local/cuda/extras/CUPTI/lib64"

export CUDA_HOME=/usr/local/cuda

export PATH="$PATH:/usr/local/cuda/bin"
```

3) Run the below command to source

```
source ~/.bashrc
```

Miniconda installation

At this stage we have installed all the required software for tensorflow-gpu version to work. Now, we need to install the miniconda software that will enable us to create python virtual environments.

```
wget https://repo.continuum.io/miniconda/Miniconda3-latest-
Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh
```

During the installation, it will be asked to append the path to ~/.bahrc and type yes to append them.

Tensorflow-gpu installation

Finally, we are ready to install the gpu version of TensorFlow. We need to run below steps to get there:

1. Create virtual environment.

```
conda create -n tensorflow_gpu
```

2. Get into the environment

```
source activate tensorflow_gpu
```

3. Install tensorflow-gpu

```
pip install tensorflow-gpu
```

4. Check the installation by create a file called 'test.py' and add below commands to it

```
import tensorflow as tf  
  
hello = tf.constant('Hello there!')  
  
print(sess.run(hello))
```

5. Run 'python test. py' and you should be able to get 'Hello there!' message that is he indication gpu version TensorFlow has been installed successfully. Otherwise, Installation failed.

You can see similar write up in [Github repo](#).

If you like my write up, follow me on [Github](#), [Linkedin](#), and/or [Medium](#) profile.

