

Instruction to set-up Tensorflow for deep learning with CUDA (Ubuntu 14.04 64bit.)

1, install CUDA

a, From the link below download CUDA toolkit 8.0:

<https://developer.nvidia.com/cuda-downloads>

b, follow the instructions provided at the download page.

Learn more about CUDA Toolkit 8.0:

- Read the **Introduction to CUDA C and C++** Parallel Forall Blog Post.
- Sign up for a live walkthrough: **What's New in CUDA 8** webinar on Thursday, October 13th.
- Review the CUDA 8 Performance Overview **Webinar, Slides**

For Linux users upgrading from previous versions of the CUDA Toolkit, click to see instructions in this section before proceeding.

For developers on x86-64 systems using Tesla P100 GPUs, click to see instructions before installation..

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	16.04	14.04				
Installer Type ⓘ	runfile (local)	deb (local)	deb (network)	cluster (local)		

Related Links

- CUDA Quick Start Guide
- Release Notes
- EULA
- Online Documentation
- CUDA Toolkit Overview
- Installer Checksums
- Open Source Packages
- Legacy CUDA Toolkits

Download Installer for Linux Ubuntu 14.04 x86_64

The base installer is available for download below.

> Base Installer

Download [1.9 GB] ⬇

Installation Instructions:

1. ``sudo dpkg -i cuda-repo-ubuntu1404-8-0-local_8.0.44-1_amd64.deb``
2. ``sudo apt-get update``
3. ``sudo apt-get install cuda``

c, add path to “bashrc” file.

(Type in command line “gedit ~/.bashrc”, and then append the following to this bashrc file.

add cuda libraries to library path

```
if [[ "${LD_LIBRARY_PATH}" != "" ]]
```

then

```
    export LD_LIBRARY_PATH=/usr/local/cuda/lib64:${LD_LIBRARY_PATH}
```

else

```
    export LD_LIBRARY_PATH=/usr/local/cuda/lib64
```

```
fi
```

Test CUDA installation by typing `nvcc --version` in terminal, you should see something similar:

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2016 NVIDIA Corporation
Built on Sun_Sep__4_22:14:01_CDT_2016
Cuda compilation tools, release 8.0, V8.0.44
```

2, Install CuDNN

a, register and download CuDNN from following link.

<https://developer.nvidia.com/cudnn>

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

☒ I Agree To the Terms of the **cuDNN Software License Agreement**

Please check your framework documentation to determine the recommended version of cuDNN.

If you are using cuDNN with a Pascal (GTX 1080, GTX 1070), version 5 or later is required.

[Download cuDNN v5.1 \(Jan 20, 2017\), for CUDA 8.0](#)

[cuDNN User Guide](#)

[cuDNN Install Guide](#)

[cuDNN v5.1 Library for Linux](#)

b, after download the cuDNN v5.1 Library, unzip the file and copy to CUDA libraries by doing

```
sudo cp lib64/* /usr/local/cuda/lib64/
sudo cp include/* /usr/local/cuda/include/
sudo chmod a+r /usr/local/cuda/lib64/libcudnn*
(make sure the package is downloaded completely and unzipped successfully.)
```

3, Install Tensorflow (I only show pip install which is the easiest one)

(https://www.tensorflow.org/get_started/os_setup#pip_installation)

Ubuntu/Linux 64-bit, CPU only, Python 2.7 (if you do not need GPU enabled, optional)

```
$ export TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-0.12.1-cp27-none-linux_x86_64.whl
```

If you do not have CUDA, skip the following 4 lines and continue on page 3.

Ubuntu/Linux 64-bit, GPU enabled, Python 2.7

Requires CUDA toolkit 8.0 and CuDNN v5.

```
$ export TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow_gpu-0.12.1-cp27-none-linux_x86_64.whl
```

sudo pip install --upgrade \$TF_BINARY_URL

pip install tensorflow (if no CUDA enabled)

or

pip install tensorflow-gpu

check GPU is enabled:

Open python environment by typing python in terminal, and type in import tensorflow as tf, you will see

```
>>> import tensorflow as tf
I tensorflow/stream_executor/dso_loader.cc:128] successfully opened CUDA library libcublas.so locally
I tensorflow/stream_executor/dso_loader.cc:128] successfully opened CUDA library libcudnn.so locally
I tensorflow/stream_executor/dso_loader.cc:128] successfully opened CUDA library libcufft.so locally
I tensorflow/stream_executor/dso_loader.cc:128] successfully opened CUDA library libcuda.so.1 locally
I tensorflow/stream_executor/dso_loader.cc:128] successfully opened CUDA library libcurand.so locally
>>> 
```

Done!