

[HOME](#)[PYTHON](#)[TENSORFLOW](#)[OPENCV](#)[MACHINE LEARNING](#)[GUEST POST](#)[WRITE TO US](#)

How to install Tensorflow GPU with CUDA 10.0 for python on Ubuntu

THIS TUTORIAL IS FOR BUILDING TENSORFLOW 1.12 GPU FROM SOURCE ALONG WITH CUDA 10 AND CUDNN 7.3.1

🕒 2018-10-13 👤 Arun Mandal 💬 48

This is going to be a tutorial on how to install **tensorflow 1.12 GPU version**. We will also be installing **CUDA 10.0** and **cuDNN 7.3.1** along with the GPU version of tensorflow 1.12. At the time of writing this blog post, the latest version of tensorflow is 1.12. This tutorial is for building tensorflow from source. If you want to use the official pre-built pip package instead, I recommend another post, [How to install Tensorflow using official pip package](#).

[Tensorflow](#) is an open source software library developed and used by Google that is fairly common among students, researchers, and developers for deep learning applications such as neural networks. It has both the CPU as well as GPU version available and although the CPU version works quite well, realistically, if you are going for deep learning, you will need GPU. In order to use the GPU version of TensorFlow, you will need an [NVIDIA GPU with a compute capability > 3.0](#).

Using latest version of Tensorflow provides you latest features and optimization, using latest CUDA Toolkit provides you speed improvement with latest gpu support and using latest CUDNN greatly improves deep learning training time.

There must be 64-bit python installed tensorflow does not work on 32-bit python installation.

Step 1: Update and Upgrade your system:

```
sudo apt-get update
```

[SUBSCRIBE TO US](#)

Name

Email *

[SUBSCRIBE](#)[RECENT POSTS](#)

python data types, interactive help, and built-in functions

🕒 2018-12-28 💬 0



Yearly Review - 2018

🕒 2018-12-26 💬 0



Top 10 reasons why you should learn python

🕒 2018-11-07 💬 0

```
sudo apt-get upgrade
```

Step 2: Verify You Have a CUDA-Capable GPU:

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

Note GPU model. eg. GeForce 840M

If you do not see any settings, update the PCI hardware database that Linux maintains by entering update-pciids (generally found in /sbin) at the command line and rerun the previous lspci command.

If your graphics card is from NVIDIA then goto <http://developer.nvidia.com/cuda-gpus> and verify if listed in CUDA enabled gpu list.

Note down its Compute Capability. eg. GeForce 840M 5.0

Step 3: Verify You Have a Supported Version of Linux:

To determine which distribution and release number you're running, type the following at the command line:

```
uname -m && cat /etc/*release
```

The x86_64 line indicates you are running on a 64-bit system which is supported by cuda 9.1

Step 4: Install Dependencies:

Required to compile from source:

```
sudo apt-get install build-essential
```

```
sudo apt-get install cmake git unzip zip
```

```
sudo apt-get install python-dev python3-dev python-pip python3-pip
```

Step 5: Install linux kernel header:

Goto terminal and type:

```
uname -r
```

You can get like "4.15.0-36-generic". Note down linux kernel version.

To install linux header supported by your linux kernel do following:

```
sudo apt-get install linux-headers-$(uname -r)
```



Python 3.7 download and install for windows

🕒 2018-10-24 💬 0



python3 print function

🕒 2018-10-20 💬 3



python data types,
interactive help, and built-in functions

Yearly Review – 2018

Top 10 reasons why you should learn python

Python 3.7 download and install for windows

python3 print function

How to install Tensorflow GPU with CUDA 10.0 for python on Windows

How to install Tensorflow GPU with CUDA 10.0 for python on Ubuntu

Wrangling Data with Pandas

Introduction to programming using python3

How to use MTCNN face detection to monitor your background

Step 6: Install NVIDIA CUDA 10.0:

Remove previous cuda installation:

```
sudo apt-get purge nvidia*

sudo apt-get autoremove

sudo apt-get autoclean

sudo rm -rf /usr/local/cuda*
```

Install cuda :

For Ubuntu 16.04 :

```
sudo apt-key adv --fetch-keys
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/
x86_64/7fa2af80.pub

echo "deb
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/
/x86_64/" | sudo tee /etc/apt/sources.list.d/cuda.list
```

For Ubuntu 18.04 :

```
sudo apt-key adv --fetch-keys
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/
x86_64/7fa2af80.pub

echo "deb
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/
/x86_64/" | sudo tee /etc/apt/sources.list.d/cuda.list
```

For Both Options:

```
sudo apt-get update

sudo apt-get -o Dpkg::Options::="--force-overwrite" install cuda-
10-0 cuda-drivers
```

You can also install cuda toolkit following instructions from [here](#) and it is recommended to use deb[network].

Step 7: Reboot the system to load the NVIDIA drivers.

Step 8: Go to terminal and type:

```
echo 'export PATH=/usr/local/cuda-10.0/bin${PATH:+:${PATH}}' >>
~/.bashrc
```

```
echo 'export LD_LIBRARY_PATH=/usr/local/cuda-10.0/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}' >> ~/.bashrc
```

```
source ~/.bashrc
```

```
sudo ldconfig
```

```
nvidia-smi
```

Check driver version probably Driver Version: 396.26

(not likely) If you got nvidia-smi is not found then you have unsupported linux kernel installed. Comment your linux kernel version noted in step 5.

You can check your cuda installation using following sample:

```
cuda-install-samples-10.0.sh ~
```

```
cd ~/NVIDIA_CUDA-10.0_Samples/5_Simulations/nbody
```

```
make
```

```
./nbody
```

1 2



CUDA 10

HOW TO

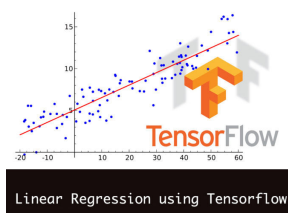
INSTALL

TENSORFLOW

TENSORFLOW-GPU

UBUNTU

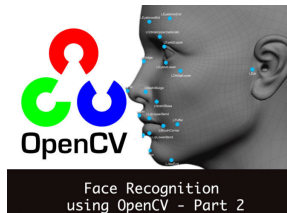
RELATED ARTICLES



Getting Started with Tensorflow (Linear Regression)



Python 3.6 download and install for windows



Face Recognition using OpenCV – Part 2



HOME

PYTHON

TENSORFLOW

OPENCV

MACHINE LEARNING

GUEST POST

WRITE TO US

How to install Tensorflow GPU with CUDA 10.0 for python on Ubuntu

THIS TUTORIAL IS FOR BUILDING TENSORFLOW 1.12 GPU FROM SOURCE ALONG WITH CUDA 10 AND CUDNN 7.3.1

🕒 2018-10-13 👤 Arun Mandal 💬 48

Step 9: Install cuDNN 7.3.1:

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

Goto <https://developer.nvidia.com/cudnn> and download Login and agreement required

After login and accepting agreement.

Download the following:

cuDNN v7.3.1 Library for Linux [cuda 10.0]

Goto downloaded folder and in terminal perform following:

```
tar -xvf cudnn-10.0-linux-x64-v7.3.1.20.tgz
```

```
sudo cp -R cuda/include/* /usr/local/cuda-10.0/include
```

```
sudo cp -R cuda/lib64/* /usr/local/cuda-10.0/lib64
```

Step 10: Install NCCL 2.3.5:

SUBSCRIBE TO US

Name

Email *

SUBSCRIBE

RECENT POSTS



python data types,
interactive help,
and built-in
functions

🕒 2018-12-28 💬 0



Yearly Review –
2018

🕒 2018-12-26 💬 0



Top 10 reasons why
you should learn
python

🕒 2018-11-07 💬 0

NVIDIA Collective Communications Library (NCCL) implements multi-GPU and multi-node collective communication primitives that are performance optimized for NVIDIA GPUs

Go to <https://developer.nvidia.com/nccl/nccl-download> and attend survey to download Nvidia NCCL.

Download following after completing survey.

Download NCCL v2.3.5, for CUDA 10.0 -> NCCL 2.3.5 O/S agnostic and CUDA 10.0

Goto downloaded folder and in terminal perform following:

```
tar -xf nccl_2.3.5-2+cuda10.0_x86_64.tgz
```

```
cd nccl_2.3.5-2+cuda10.0_x86_64
```

```
sudo cp -R * /usr/local/cuda-10.0/targets/x86_64-linux/
```

```
sudo ldconfig
```

Step 11: Install Dependencies

Use following if not in active virtual environment.

```
pip install -U --user pip six numpy wheel mock
```

```
pip3 install -U --user pip six numpy wheel mock
```

```
pip install -U --user keras_applications==1.0.5 --no-deps
```

```
pip3 install -U --user keras_applications==1.0.5 --no-deps
```

```
pip install -U --user keras_preprocessing==1.0.3 --no-deps
```

```
pip3 install -U --user keras_preprocessing==1.0.3 --no-deps
```

Use following if in active virtual environment.

```
pip install -U pip six numpy wheel mock
```

```
pip install -U keras_applications==1.0.5 --no-deps
```

```
pip install -U keras_preprocessing==1.0.3 --no-deps
```

Step 12: Configure Tensorflow from source:

Download bazel:



**Python 3.7
download and
install for windows**

🕒 2018-10-24 💬 0



**python3 print
function**

🕒 2018-10-20 💬 3



python data types,
interactive help, and built-
in functions

Yearly Review – 2018

Top 10 reasons why you
should learn python

Python 3.7 download and
install for windows

python3 print function

How to install Tensorflow
GPU with CUDA 10.0 for
python on Windows

How to install Tensorflow
GPU with CUDA 10.0 for
python on Ubuntu

Wrangling Data with
Pandas

Introduction to
programming using
python3

How to use MTCNN face
detection to monitor your
background

```
cd ~/

wget
https://github.com/bazelbuild/bazel/releases/download/0.17.2/bazel-
0.17.2-installer-linux-x86_64.sh

chmod +x bazel-0.17.2-installer-linux-x86_64.sh

./bazel-0.17.2-installer-linux-x86_64.sh --user

echo 'export PATH="$PATH:$HOME/bin"' >> ~/.bashrc
```

Reload environment variables

```
source ~/.bashrc
sudo ldconfig
```

Start the process of building TensorFlow by downloading latest tensorflow 1.12 .

```
cd ~/

git clone https://github.com/tensorflow/tensorflow.git

cd tensorflow

git checkout r1.12

./configure
```

Give python path in

```
Please specify the location of python. [Default is
/usr/bin/python]: /usr/bin/python3
```

Press enter two times

```
Do you wish to build TensorFlow with Apache Ignite support? [Y/n]:
Y
```

```
Do you wish to build TensorFlow with XLA JIT support? [Y/n]: Y
```

```
Do you wish to build TensorFlow with OpenCL SYCL support? [y/N]: N
```

```
Do you wish to build TensorFlow with ROCm support? [y/N]: N
```

```
Do you wish to build TensorFlow with CUDA support? [y/N]: Y
```

```
Please specify the CUDA SDK version you want to use. [Leave empty
to default to CUDA 9.0]: 10.0
```

```
Please specify the location where CUDA 10.0 toolkit is installed.
Refer to README.md for more details. [Default is /usr/local/cuda]:
```

```
/usr/local/cuda-10.0
```

Please specify the cuDNN version you want to use. [Leave empty to default to cuDNN 7]: 7.3.1

Please specify the location where cuDNN 7 library is installed. Refer to README.md for more details. [Default is /usr/local/cuda-10.0]: /usr/local/cuda-10.0

Do you wish to build TensorFlow with TensorRT support? [y/N]: N

Please specify the NCCL version you want to use. If NCCL 2.2 is not installed, then you can use version 1.3 that can be fetched automatically but it may have worse performance with multiple GPUs. [Default is 2.2]: 2.3.5

Now we need compute capability which we have noted at step 1 eg. 5.0

Please note that each additional compute capability significantly increases your build time and binary size. [Default is: 5.0] 5.0

Do you want to use clang as CUDA compiler? [y/N]: N

Please specify which gcc should be used by nvcc as the host compiler. [Default is /usr/bin/gcc]: /usr/bin/gcc

Do you wish to build TensorFlow with MPI support? [y/N]: N

Please specify optimization flags to use during compilation when bazel option "--config=opt" is specified [Default is -march=native]: -march=native

Would you like to interactively configure ./WORKSPACE for Android builds? [y/N]:N

Configuration finished

Step 13: Build Tensorflow using bazel

The next step in the process to install tensorflow GPU version will be to build tensorflow using bazel. This process takes a fairly long time.

To build a pip package for TensorFlow you would typically invoke the following command:

```
bazel build --config=opt --config=cuda  
//tensorflow/tools/pip_package:build_pip_package
```

Note:-

add "--config=mkl" if you want Intel MKL support for newer intel cpu for faster training on cpu

add "--config=monolithic" if you want static monolithic build (try this if build failed)

add "--local_resources 2048,.5,1.0" if your PC has low ram causing Segmentation fault or other related errors

This process will take a lot of time. It may take 3- 4 hours or maybe even more.

Also if you got error like Segmentation Fault then try again it usually worked.

The bazel build command builds a script named build_pip_package. Running this script as follows will build a .whl file within the tensorflow_pkg directory:

To build whl file issue following command:

```
bazel-bin/tensorflow/tools/pip_package/build_pip_package
tensorflow_pkg
```

To install tensorflow with pip:

```
cd tensorflow_pkg
```

for existing virtual environment:

```
pip install tensorflow*.whl
```

With a new virtual environment using virtualenv:

```
sudo apt-get install virtualenv
virtualenv tf_1.12_cuda10.0 -p /usr/bin/python3
source tf_1.12_cuda10.0/bin/activate
pip install tensorflow*.whl
```

for python 2: (use sudo if required)

```
pip2 install tensorflow*.whl
```

for python 3: (use sudo if required)

```
pip3 install tensorflow*.whl
```

Note : if you got error like unsupported platform then make sure you are running correct pip command associated with the python you used while configuring tensorflow build.

You can check pip version and associated python by following command

```
pip -V
```

Step 14: Verify Tensorflow installation

Run in terminal

```
python
```

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

If the system outputs the following, then you are ready to begin writing TensorFlow programs:

Success! You have now successfully installed tensorflow 1.12 on your machine. If you are on Windows OS, you might want to check out our other post here, [How to install Tensorflow 1.7.0 GPU with CUDA 9.1 and cuDNN 7.1.2 for Python 3 on Windows OS](#). Cheers!!

For prebuilt wheels go to [this](#) link .

1

2



CUDA 10

HOW TO

INSTALL

TENSORFLOW

TENSORFLOW-GPU

UBUNTU

RELATED ARTICLES



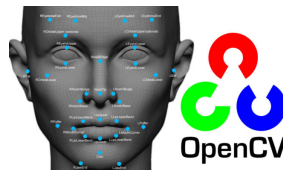
TensorFlow 1.8 CUDA 9.2

**How to install
Tensorflow GPU
with CUDA 9.2 for
python on Ubuntu**



Install TensorFlow GPU on windows

**How to install
Tensorflow GPU on
Windows**

Face Recognition
using OpenCV - Part 1

**Face Recognition
using OpenCV –
Part 1**

**18 COMMENTS ON HOW TO INSTALL TENSORFLOW GPU WITH
CUDA 10.0 FOR PYTHON ON UBUNTU**