Tensorflow完整安装手册

|  |  |
| --- | --- |
| 制作小组 | BIG DATA GROUP |
| 制作日期 | 2018.09.19 |

Contents

[第一章 Windows系统下安装tensorflow 3](#_Toc525143667)

[1软件支持 3](#_Toc525143668)

[1.1 Python 3](#_Toc525143669)

[1.2 CUDA和CuDNN 3](#_Toc525143670)

[1.3 TensorFlow 3](#_Toc525143671)

[2安装过程 4](#_Toc525143672)

[2.1. 安装Anaconda3 5.1 4](#_Toc525143673)

[2.2 安装CUDA® Toolkit 9.0+cuDNN v7.1 4](#_Toc525143674)

[1）CUDA9.0安装 4](#_Toc525143675)

[2）CuDNN7.1安装 4](#_Toc525143676)

[2.3 安装tensorflow1.7 5](#_Toc525143677)

[2.3.1 激活并进入环境 5](#_Toc525143678)

[2.3.2 升级pip到最新版 5](#_Toc525143679)

[2.3.3 安装tensorflow1.7.0及相应依赖包 5](#_Toc525143680)

[2.4验证 5](#_Toc525143681)

[第二章 Linux系统下安装tensorflow 6](#_Toc525143682)

[1. 第1版安装TensorFlow 1.5 6](#_Toc525143683)

[2. 第2版安装TensorFlow 1.9 8](#_Toc525143684)

[2.1 For Tensorflow 1.9 without TensorRT 8](#_Toc525143685)

[2.2 For Tensorflow 1.9 with TensorRT (need python 3.5 because TensorRT don’t support python 3.6) 9](#_Toc525143686)

# Windows系统下安装tensorflow

# 1软件支持

 Win10上搭建TensorFlow的开发环境需要至少需要安装3个软件，分别为：1.python，2.CUDA和CuDNN，3.TensorFlow（GPU版）。安装前一定要确认好每个软件的版本是否相互支持。

## 1.1 Python

相比于作为一个过渡版本的Python 2.6，笔者选择的是Python 3.6。在安装时，选择了Anaconda（一个开源的Python发行版本）的最新版本。Anaconda包含了conda、Python等180多个科学包及其依赖项，功能十分强大。  
Anaconda下载地址：[https://www.anaconda.com/download/](https://www.anaconda.com/download/" \t "_blank)

## 1.2 CUDA和CuDNN

CUDA是NVIDIA推出的运算平台，CuDNN是专门针对Deep Learning框架设计的一套GPU计算加速方案。在安装之前要查询下最新TensorFLow发行版支持到了哪个版本。笔者在安装TensorFLow时，CUDA已经到了9.1版本，但是TensorFLow1.7只支持到9.0版。另外，也要确认CUDA版本是否支持自己的显卡。笔者电脑的MX150只有CUDA9.0及以上的版本才支持。基于以上两个条件，笔者选择了CUDA9.0，并下载了对应的CuDNN版本。另外，笔者在后续的编程中发现tensorflow1.7只支持7.0的CuDNN。  
1）显卡型号支持：[https://developer.nvidia.com/cuda-gpus](https://developer.nvidia.com/cuda-gpus" \t "_blank)

2）CUDA下载地址：[https://developer.nvidia.com/cuda-toolkit-archive](https://developer.nvidia.com/cuda-toolkit-archive" \t "_blank)

3）CuDNN下载地址：[https://developer.nvidia.com/rdp/cudnn-download](https://developer.nvidia.com/rdp/cudnn-download" \t "_blank)

## 1.3 TensorFlow

TensorFlow的版本信息可以在Github，tensorflow中文社区以及pypi上查看。

    Github：[https://github.com/tensorflow/tensorflow/releases](https://github.com/tensorflow/tensorflow/releases" \t "_blank)

    tensorflow社区：[https://tensorflow.google.cn/versions/](https://tensorflow.google.cn/versions/" \t "_blank)

    pypi：[https://pypi.org/project/tensorflow/#history](https://pypi.org/project/tensorflow/" \l "history" \t "_blank)

TensorFlow可以直接在Anaconda Prompt的命令行中用指令：“conda install tensorflow-gpu”直接安装。

# 2安装过程

## 2.1. 安装Anaconda3 5.1

 Anaconda过程安装过程简单，打开安装包后选择好路径后就能安装

安装启动前若不勾选上第一项，则需在软件安装完成手动添加环境变量

## 2.2 安装CUDA® Toolkit 9.0+cuDNN v7.1

### 1）CUDA9.0安装

运行cuda\_9.0.176\_win10.exe。安装软件会先运行一个系统检查，如果没有软硬件不兼容的情况就能继续进行下一步。若有不兼容情况，系统检查则会报错，无法进行下一步。如果报错了，就最先考虑下是不是自己的显卡不被该版本支持。在选项着一栏选择自定义，否则安装程序会一股脑把里面所有软件包都给你装上。在选择安装项时一般不安装GeForce Experience，CUDA是核心组件必须勾上，剩下两个选项的当前版本如果比新版本低的话也可以勾选上。接着点击下一步就开始安装了。

### 2）CuDNN7.1安装

解压压缩包cudnn-9.0-windows10-x64-v7.0.zip，得到三个文件夹:bin , include , lib.

将这三个文件拷贝到CUDA9.0的安装路径根文件夹下.

## 2.3 安装tensorflow1.7

打开Anaconda Prompt，进入Anaconda命令行管理界面。配置清华仓库镜，输入指令：  
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/  
conda config --set show\_channel\_urls yes  
创建运行环境，输入指令：  
conda create -n tensorflow-gpu python=3.6  
新建一个名字叫“tensorflow-gpu”，python版本为3.6的运行环境，此环境与Anaconda中其它环境隔离。红框中的软件包也会随之安装，输入“y“和回车后开始安装。

### 2.3.1 激活并进入环境

使后续指令在激活的环境中生效，输入指令：conda activate tensorflow-gpu

### 2.3.2 升级pip到最新版

防止稍后的安装时出现错误，输入指令：

python -m pip install --upgrade pip

### 2.3.3 安装tensorflow1.7.0及相应依赖包

输入指令：pip install --ignore-installed --upgrade tensorflow-gpu  
如果需要在本次安装环境并作为主环境，则不需要进行conda环境设置，直接利用上面pip进行安装。

安装出现Access is denied，在命令后面添加 –user可以解决

至此，tensorflow的安装完成。

## 2.4验证

在命令行中，进入python，并输入以下代码：  
import tensorflow as tf  
hello = tf.constant('Hello, TensorFlow!')  
sess = tf.Session()  
print(sess.run(hello))

输出：  
b’ Hello, TensorFlow!

## 2.5 Sklearn windows 安装

方式一：

在命令行中输入：pip install --upgrade scikit-learn

方式二：

命令行中输入：conda install scikit-learn

# 第二章 Linux系统下安装tensorflow

## 第1版安装TensorFlow 1.5

* install CUDA 9.1 <https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html#ubuntu-installation>
* install cuDNN 7 [https://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html](https://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html#troubleshoot)
* install anaconda
* conda create -n tensorflow-1.5.0 pip python=3.6
* source activate tensorflow-1.5.0
* cd ~/AI\_software/tensorFlow
* git clone <https://github.com/tensorflow/tensorflow>
* cd tensorflow
* git checkout r1.0
* install bazel
  + sudo apt-get install bazel
* install libcupti
* pip install numpy dev pip wheel
* ./configure
  + Do you wish to build TensorFlow with CUDA support? [y/N] Y
  + Please specify the CUDA SDK version you want to use, e.g. 7.0. [Leave empty to default to CUDA 9.0]: 9.1
  + Do you wish to build TensorFlow with MPI support? [y/N] #TODO []try Y later
* export LD\_LIBRARY\_PATH=/usr/local/cuda-X.X/lib64/:/usr/local/cuda-X.X/extras/CUPTI/lib64
* bazel build --config=opt --config=cuda //tensorflow/tools/pip\_package:build\_pip\_package --cxxopt="-D\_GLIBCXX\_USE\_CXX11\_ABI=0"
* bazel-bin/tensorflow/tools/pip\_package/build\_pip\_package /tmp/tensorflow\_pkg
* Save the whl package at /tmp/tensorflow\_pkg for other user install
* pip install /tmp/tensorflow\_pkg/tensorflow-1.5.0-cp36-cp36m-linux\_x86\_64.whl # the whl file name might change
* python
  + **# Python**

**import tensorflow as tf**

**hello = tf.constant('Hello, TensorFlow!')**

**sess = tf.Session()**

print(sess.run(hello))

* install ipython
  + conda install notebook
  + ipython kernel install --name tensorflow-1.5.0
  + source deactivate
  + source activate tensorflow-1.5.0
* run mnist test finished in 83.750 seconds
  + cd /home/Shared/Softwares/mnist/
  + python -m cProfile mnist\_deep.py --data\_dir MNIST\_data |& tee cProfile.log

#### problems remains:

1 solved, caused by error format in /etc/environment

sudo apt install elinks # or any other packages

Setting up install-info (6.1.0.dfsg.1-5) ...

/usr/sbin/update-info-dir: 2: /etc/environment: /usr/local/cuda/bin: Permission denied

dpkg: error processing package install-info (--configure):

subprocess installed post-installation script returned error exit status 126

Errors were encountered while processing:

install-info

E: Sub-process /usr/bin/dpkg returned an error code (1)

2 jupyter notebook

8888 port not open? ask Longbo

#### note:

# get the public ip address

curl <http://ipecho.net/plain>

**# for new user to setup the env**

bash /home/Shared/Softwares/Anaconda3-5.0.1-Linux-x86\_64.sh

conda create -n tensorflow

source activate tensorflow

pip install numpy dev wheel

cp -r /home/Shared/Softwares/tensorFlow/tensorflow  ~/

cd ~/tensorflow

./configure

**Please specify the CUDA SDK version you want to use, e.g. 7.0. [Leave empty to default to CUDA 9.0]: 9.1**

bazel build --config=opt --config=cuda //tensorflow/tools/pip\_package:build\_pip\_package --cxxopt="-D\_GLIBCXX\_USE\_CXX11\_ABI=0"

bazel-bin/tensorflow/tools/pip\_package/build\_pip\_package ~

pip install ~/tensorflow-1.5.0-cp36-cp36m-linux\_x86\_64.whl

#### ref:

<https://www.tensorflow.org/install/install_sources>

==================================================================================================================

## 第2版安装TensorFlow 1.9

* install CUDA 9.2 <https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html#ubuntu-installation>
* install cuDNN 7.1.4 [https://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html](https://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html#troubleshoot)
* No need to change the symlink /usr/local/cuda (still point to the old /usr/local/cuda-9.1)

### 2.1 For Tensorflow 1.9 without TensorRT

* conda create -n tensorflow-1.9 pip python=3.6
* source activate tensorflow-1.9
* cd /home/Shared/Softwares/tensorFlow/tensorflow
* git pull
* git checkout r1.9
* pip install numpy dev pip wheel
* bazel clean
* ./configure
  + AWS S3 support: Y
  + Do you wish to build TensorFlow with CUDA support? [y/N] Y
  + Please specify the CUDA SDK version you want to use, e.g. 7.0. [Leave empty to default to CUDA 9.0]: 9.2
  + CUDA Path: /usr/local/cuda-9.2
  + cuDNN: 7.1.4
* export LD\_LIBRARY\_PATH=/usr/local/cuda-9.2/lib64/:/usr/local/cuda-9.2/extras/CUPTI/lib64
* bazel build --config=opt --config=cuda //tensorflow/tools/pip\_package:build\_pip\_package --cxxopt="-D\_GLIBCXX\_USE\_CXX11\_ABI=0"
* bazel-bin/tensorflow/tools/pip\_package/build\_pip\_package /tmp/tensorflow\_pkg
* Save the whl package at /tmp/tensorflow\_pkg to /home/Shared/Softwares/tensorFlow for other user install without recomplie
* cd ..
* pip install tensorflow-1.9.0-cp36-cp36m-linux\_x86\_64.whl
* python
  + **# Python**

**import tensorflow as tf**

**hello = tf.constant('Hello, TensorFlow!')**

**sess = tf.Session()**

print(sess.run(hello))

* run mnist test passed
  + cd /home/Shared/Softwares/mnist/
  + python -m cProfile mnist\_deep.py --data\_dir MNIST\_data |& tee cProfile.log

### 2.2 For Tensorflow 1.9 with TensorRT (need python 3.5 because TensorRT don’t support python 3.6)

* conda create -n tensorflow-1.9\_tensorRT pip python=3.5
* source activate tensorflow-1.9\_tensorRT
* cd /home/Shared/Softwares/
* Download **TensorRT tar (don’t use deb)** <https://docs.nvidia.com/deeplearning/sdk/tensorrt-install-guide/index.html#gettingstarted>
* tar xzvf TensorRT-4.0.1.6.Ubuntu-16.04.4.x86\_64-gnu.cuda-9.2.cudnn7.1.tar
* mv TensorRT-4.0.1.6  /usr/local/
* cd /usr/local/TensorRT-4.0.1.6/python
* export LD\_LIBRARY\_PATH=/usr/local/cuda-9.2/lib64/:/usr/local/cuda-9.2/extras/CUPTI/lib64:/usr/local/TensorRT-4.0.1.6/lib
* export TENSORRT\_INC\_DIR=/usr/local/TensorRT-4.0.1.6/include
* export TENSORRT\_LIB\_DIR=/usr/local/TensorRT-4.0.1.6/lib
* pip install tensorrt-4.0.1.6-cp35-cp35m-linux\_x86\_64.whl
* cd ..
* pip install uff/uff-0.4.0-py2.py3-none-any.whl
* pip install graphsurgeon/graphsurgeon-0.2.0-py2.py3-none-any.whl
* #install tensorflow 1.9
* cd /home/Shared/Softwares/tensorFlow/tensorflow
* bazel clean
* ./configure
  + AWS S3 support: Y
  + Do you wish to build TensorFlow with CUDA support? [y/N] Y
  + Please specify the CUDA SDK version you want to use, e.g. 7.0. [Leave empty to default to CUDA 9.0]: 9.2
  + CUDA Path: /usr/local/cuda-9.2
  + cuDNN: 7.1.4
  + Do you wish to build TensorFlow with TensorRT support? [y/N]: Y
  + Please specify the location where TensorRT is installed. [Default is /usr/lib/x86\_64-linux-gnu]: /usr/local/TensorRT-4.0.1.6
* bazel build --config=opt --config=cuda //tensorflow/tools/pip\_package:build\_pip\_package --cxxopt="-D\_GLIBCXX\_USE\_CXX11\_ABI=0"
* bazel-bin/tensorflow/tools/pip\_package/build\_pip\_package /tmp/tensorflow\_pkg
* mv /tmp/tensorflow\_pkg/tensorflow-1.9.0-cp35-cp35m-linux\_x86\_64.whl /home/Shared/Softwares/tensorFlow/tensorflow-1.9.0-cp36-cp36m-linux\_x86\_64\_TensorRT.whl
* cd /home/Shared/Softwares/tensorFlow
* pip install tensorflow-1.9.0-cp36-cp36m-linux\_x86\_64\_TensorRT.whl
* **tensorflow-1.9.0-cp36-cp36m-linux\_x86\_64\_tensorRT.whl is not a supported wheel on this platform.**