# （一）Win10-x64下安装tensorflow

## 1 安装前的准备过程

最近打算学习深度学习方面的知识，打算把深度学习和之前研究的双目视觉目标测距结合起来，提高目标测距的鲁棒性和准确性。因为之前没有接触过深度学习的知识，只是简单了解过神经网络控制算法。导师向我推荐了Google旗下的开源深度学习框架tensorflow。于是开始网上寻找资料进行TF的安装。

网上关于TF的安装教程五花八门，看了很多篇之后发现，TF最初的很多个版本是不支持windows系统的，只支持基于Unix内核的操作系统，包括Mac OSX和Linux系统。所以，广大的windows用户为了使用TF，必须创建一个Linux虚拟环境或者下载一个Linux操作系统，安装双系统。过程十分繁琐。我也差点入坑。其中较完整的参考博文如下：

<http://www.leiphone.com/news/201606/ORlQ7uK3TIW8xVGF.html>

<http://blog.csdn.net/zr459927180/article/details/51627910>

还好后面继续了解到，在2016/11/29，官方终于发布了TensorFlow 0.12,从该版本开始，TF添加了对windows版本的支持。原网址如下：

<https://developers.googleblog.com/2016/11/tensorflow-0-12-adds-support-for-windows.html>

于是找到了一篇非常优秀的博文，详细地介绍了win10-x64下TF的安装：

<http://blog.csdn.net/u010099080/article/details/53418159>

虽然安装难度降低了很多，但我还是折腾了两天才把TF成功安装，下面记录一下过程中碰到的问题：

## 2 安装过程中遇到的问题

由于TF开发中主要是使用Python语言。之前并没有接触过Python，于是在其官网下载了Python3.5.3版本进行安装。根据安装博文，可以直接在cmd.exe中利用Python的pip命令：

# GPU版本

pip3 install --upgrade tensorflow-gpu

# CPU版本

pip3 install --upgrade tensorflow

进行TF的在线下载安装。windows版本的TF目前只支持Python3.5.x版本，而文中作者又提到最好安装Anaconda。于是我又从官网上下载了最新版的Anaconda for Python 3.6 version进行安装。好了，接下来是在cmd命令行了直接输入：

pip3 install --upgrade tensorflow-gpu

安装TF了。于是问题就来了：

C:\Users\yanshuang>pip3 install --upgrade tensorflow-gpu  
'pip3' 不是内部或外部命令，也不是可运行的程序  
或批处理文件。

折腾了一阵才发现，好像是直接输入：

pip install --upgrade tensorflow-gpu

而不是“pip3”，我想是不是只安装python3.5.X（没有同时安装python2.X）时就不需要在pip之后单独输入数字‘3’加以区分呢？

好了，解决掉“pip3”的问题之后，输入：

pip install --upgrade tensorflow-gpu

下一个问题出现了：

D:\Python353>pip install --upgrade tensorflow-gpu

Collecting tensorflow-gpu

Could not find a version that satisfies the requirement tensorflow-gpu (from versions: )

No matching distribution found for tensorflow-gpuD:\Python353>

这个问题就是Python版本的问题了，前面我说过，我先单独安装了一个Python3.5.3。然后又安装了Anaconda （for Python 3.6 version）。事实上，Anaconda中就已经包含了Python3.6。Anconda相当于Python的一个扩展，除了包含Python，Anconda 里面还集成了很多关于python科学计算的第三方库。所以，我电脑上有两个Python版本，而且安装Anconda时将Python3.6设置为默认版本。而前面说过了，windows版本的TF目前只支持Python3.5.x版本。于是就出现了以上错误。

解决办法是将电脑上的Python3.5.X和Python3.6（Anconda）全部卸载干净，包括删除系统环境变量的相关路径。重新下载一个包含Python3.5.X的Anconda版本进行安装。我选择的是：Anaconda3-4.2.0-Windows-x86\_64.exe。安装时注意会有两个勾选项，一个是把python加入环境变量的选项，一个是把当前anconda里python版本设置为系统默认python版本的选项，把两个都勾上。

安装完后，在cmd命令行里输入python，查看python版本是否是python3.5。如下：

C:\Users\yanshuang>python

Python 3.5.2 |Anaconda 4.2.0 (64-bit)| (default, Jul 5 2016, 11:41:13) [MSC v.1900 64 bit (AMD64)] on win32Type "help", "copyright", "credits" or "license" for more information

可以看到，Anconda4.2.0里面确实包含了Python3.5.2。好了，现在可以进行TF在线安装了：直接在cmd里输入pip命令：

pip install --upgrade tensorflow-gpu

cmd.exe里记录如下：

D:\Anaconda3>pip install --upgrade tensorflow-gpu

Collecting tensorflow-gpu

Downloading tensorflow\_gpu-1.0.1-cp35-cp35m-win\_amd64.whl (43.1MB)

100% |████████████████████████████████| 43.1MB 18kB/s

Collecting numpy>=1.11.0 (from tensorflow-gpu)

Downloading numpy-1.12.1-cp35-none-win\_amd64.whl (7.7MB)

100% |████████████████████████████████| 7.7MB 54kB/s

Collecting protobuf>=3.1.0 (from tensorflow-gpu)

Downloading protobuf-3.2.0-py2.py3-none-any.whl (360kB)

100% |████████████████████████████████| 368kB 313kB/s

Requirement already up-to-date: wheel>=0.26 in d:\anaconda3\lib\site-packages (from tensorflow-gpu)

Requirement already up-to-date: six>=1.10.0 in d:\anaconda3\lib\site-packages (from tensorflow-gpu)

Collecting setuptools (from protobuf>=3.1.0->tensorflow-gpu)

Downloading setuptools-34.3.3-py2.py3-none-any.whl (389kB)

100% |████████████████████████████████| 399kB 497kB/s

Collecting appdirs>=1.4.0 (from setuptools->protobuf>=3.1.0->tensorflow-gpu)

Downloading appdirs-1.4.3-py2.py3-none-any.whl

Collecting packaging>=16.8 (from setuptools->protobuf>=3.1.0->tensorflow-gpu)

Downloading packaging-16.8-py2.py3-none-any.whl

Collecting pyparsing (from packaging>=16.8->setuptools->protobuf>=3.1.0->tensorflow-gpu)

Downloading pyparsing-2.2.0-py2.py3-none-any.whl (56kB)

100% |████████████████████████████████| 61kB 566kB/s

Installing collected packages: numpy, appdirs, pyparsing, packaging, setuptools, protobuf, tensorflow-gpu

Found existing installation: numpy 1.11.1

Uninstalling numpy-1.11.1:

Successfully uninstalled numpy-1.11.1

Found existing installation: pyparsing 2.1.4

Uninstalling pyparsing-2.1.4:

Successfully uninstalled pyparsing-2.1.4

Found existing installation: setuptools 27.2.0

Cannot remove entries from nonexistent file d:\anaconda3\lib\site-packages\easy-install.pth

You are using pip version 8.1.2, however version 9.0.1 is available.

You should consider upgrading via the 'python -m pip install --upgrade pip' command.

D:\Anaconda3>python -m pip install --upgrade pip

Collecting pip

Downloading pip-9.0.1-py2.py3-none-any.whl (1.3MB)

100% |████████████████████████████████| 1.3MB 18kB/s

Installing collected packages: pip

Found existing installation: pip 8.1.2

Uninstalling pip-8.1.2:

Successfully uninstalled pip-8.1.2

Successfully installed pip-9.0.1

从记录中可以看到，输入pip命令后，安装程序自动在线搜索最新版本的TF（1.0.1）和其他必要组件并下载。最后面黑色部分提示，说我目前的pip版本为8.1.2，可以升级到最新的9.0.1.按照其提供的升级命令，顺利升级了pip版本。

到此为止，还是没有成功安装TF，因为出现了上面的两行红色部分提示。后来在以下这篇博文中找到了解决方法：

<http://blog.csdn.net/fool_frog/article/details/53422460>

原因是因为setuptools版本太低，tensorflow要求29.0.1，当前版本为27.2.0,在更新setuptools版本的时候又找不到easy-install.pth，导致更新失败。

解决方法：

运行：

pip install --upgrade --ignore-installed setuptools

记录如下：

C:\Users\yanshuang>pip install --upgrade --ignore-installed setuptools

Collecting setuptools

Using cached setuptools-34.4.0-py2.py3-none-any.whl

Collecting appdirs>=1.4.0 (from setuptools)

Using cached appdirs-1.4.3-py2.py3-none-any.whl

Collecting packaging>=16.8 (from setuptools)

Using cached packaging-16.8-py2.py3-none-any.whl

Collecting six>=1.6.0 (from setuptools)

Retrying (Retry(total=4, connect=None, read=None, redirect=None)) after connection broken by 'ReadTimeoutError("HTTPSConnectionPool(host='pypi.python.org', port=443): Read timed out. (read timeout=15)",)': /packages/c8/0a/b6723e1bc4c516cb687841499455a8505b44607ab535be01091c0f24f079/six-1.10.0-py2.py3-none-any.whl

Downloading six-1.10.0-py2.py3-none-any.whl

Collecting pyparsing (from packaging>=16.8->setuptools)

Using cached pyparsing-2.2.0-py2.py3-none-any.whl

Installing collected packages: appdirs, six, pyparsing, packaging, setuptools

Successfully installed appdirs-1.4.3 packaging-16.8 pyparsing-2.2.0 setuptools-34.4.0 six-1.10.0

从记录中红色部分可以看到，安装程序找到了最新的setuptools-34.4.0并成功安装。于是再次运行pip安装命令：

C:\Users\yanshuang>pip install --upgrade tensorflow-gpu

Collecting tensorflow-gpu

Using cached tensorflow\_gpu-1.0.1-cp35-cp35m-win\_amd64.whl

Requirement already up-to-date: wheel>=0.26 in d:\anaconda3\lib\site-packages (from tensorflow-gpu)

Requirement already up-to-date: numpy>=1.11.0 in d:\anaconda3\lib\site-packages (from tensorflow-gpu)

Collecting protobuf>=3.1.0 (from tensorflow-gpu)

Using cached protobuf-3.2.0-py2.py3-none-any.whl

Requirement already up-to-date: six>=1.10.0 in d:\anaconda3\lib\site-packages (from tensorflow-gpu)

Requirement already up-to-date: setuptools in d:\anaconda3\lib\site-packages (from protobuf>=3.1.0->tensorflow-gpu)

Requirement already up-to-date: appdirs>=1.4.0 in d:\anaconda3\lib\site-packages (from setuptools->protobuf>=3.1.0->tensorflow-gpu)

Requirement already up-to-date: packaging>=16.8 in d:\anaconda3\lib\site-packages (from setuptools->protobuf>=3.1.0->tensorflow-gpu)

Requirement already up-to-date: pyparsing in d:\anaconda3\lib\site-packages (from packaging>=16.8->setuptools->protobuf>=3.1.0->tensorflow-gpu)

Installing collected packages: protobuf, tensorflow-gpu

Successfully installed protobuf-3.2.0 tensorflow-gpu-1.0.1

红色部分可看出，TensorFlow-gpu-1.0.1安装成功。

接下来是在Python环境下载入TF测试是否成功，载入之后出现了如下的问题：

>>> import tensorflow as tf

……

Ic:\tf\_jenkins\home\workspace\release-win\device\gpu\os\windows\tensorflow\stream\_executor\dso\_loader.cc:126] Couldn't open CUDA library cudnn64\_5.dll

Ic:\tf\_jenkins\home\workspace\release-win\device\gpu\os\windows\tensorflow\stream\_executor\cuda\cuda\_dnn.cc:3517] Unable to load cuDNN DSO

……

显示无法打开cudnn.dll库。

又折腾了一阵，终于在TF官网上发现了相同问题的答案，原因原来是出在我下载的cudnnV6.0版本太高，因为我最开始使用的是安装博文提供的cuda8.0和cudnnV5.1，安装cuda时出现了和显卡的兼容性问题，于是我就到官网去分别下载了最新的cudaV8.0.61和cudnnV6.0。后来一直没注意cudnn版本的问题，刚才在TF官网下发现以下answer：

In addition to the answers above make sure that you've downloaded the supported version of cuDNN. Currently TensorFlow supports the older cuDNN v.5.1 while there is a newer cuDNN 6.0 available on Nvidia site. I had such errors with 6.0. When I rolled back to 5.1 everything worked.

参考网址：<http://stackoverflow.com/questions/41007279/tensorflow-on-windows-couldnt-open-cuda-library-cudnn64-5-dll>

把cudnn版本换成V5.1之后，终于能正常import tensorflow进行测试了。

## 3 小结（TensorFlow-gpu）

下面总结一下正确安装TF的过程：

### **（1）需要准备的软件包**

Python3.5.X或者是Anconda4.2.0、CUDA8.0.x、CudnnV5.1。

### （2）安装Python3.5.X或者Anconda（两者二选一）

注意，不能下载最新版的Python3.6，因为TF官网的安装教程明确说了：



网址：<https://www.tensorflow.org/install/install_windows>

另外，最好是安装一个Anconda4.2.0。里面就已经包含了Python3.5.2。不用再单独安装一个Python。这样可以省去很多麻烦，比如单独安装Python之后还要安装各种小插件之后才能执行pip安装命令。而Anconda则不用，一步到位。

提示：

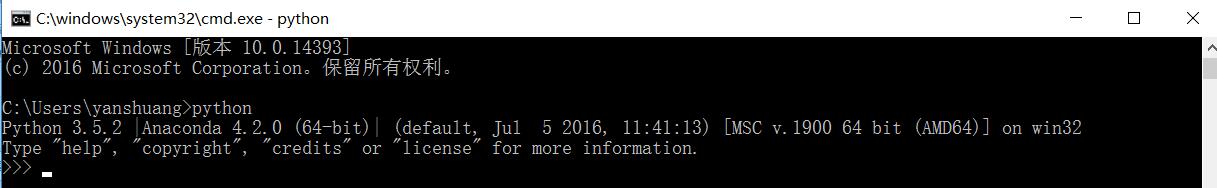
Anconda下载网址：<https://www.continuum.io/downloads/>。一般首页只会提供最新版的Anconda（目前是Python3.6 version），但是我们需要的是包含Python3.5.x的Anconda。可以在下面找到：



里面就有历史版本，在里面找到：Anaconda3-4.2.0-Windows-x86\_64.exe。

另外在安装Anconda时会有两个勾选项，记得都要打上勾（其实默认是打钩的）。一个是把python加入环境变量的选项，一个是把当前anconda里python版本设置为系统默认python版本。

安装完成后，按“win+R”快捷键，输入“cmd”，进入命令提示符窗口，输入“python”：



可以看到Anconda以及其对应的Python的版本信息都显示出来，表示Anconda安装成功。

### （3）利用pip命令安装TF

按“win+R”快捷键，输入“cmd”，进入命令提示符窗口，直接输入以下pip命令：

#若是安装cpu版本，则输入：pip install --upgrade tensorflow。

#若是安装gpu版本，则输入：pip install --upgrade tensorflow-gpu。

（提醒：如果没有安装Anconda而直接安装Python，输入的应该是“pip3 install…”）

回车。若是前面的Anconda已正确安装，则系统安装程序就会自动在线搜索最新的TF版本进行在线下载，同时还包括很多与TF有关的小插件，安装过程请保证计算机正常联网。

补充：

a. 安装TF时的pip版本如果过低，请在cmd输入：

python -m pip install --upgrade pip

进行pip版本升级。

b. 如果安装过程最后出现红色错误提示：

Found existing installation: setuptools 27.2.0

Cannot remove entries from nonexistent file d:\anaconda3\lib\site-packages\easy-install.pth

那是setuptools的版本太低，请在cmd输入：

pip install --upgrade --ignore-installed setuptools

对其进行在线升级。升级之后在用pip命令安装TF应该就能成功了。我安装成功后的最后两句提示是：

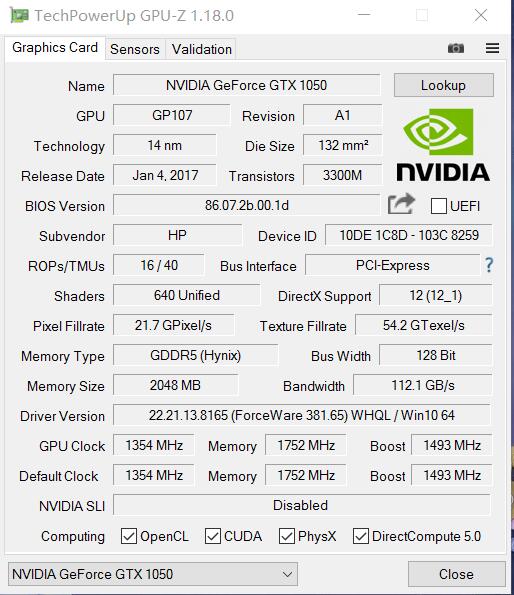
Installing collected packages: protobuf, tensorflow-gpu

Successfully installed protobuf-3.2.0 tensorflow-gpu-1.0.1

c. CPU版本和GPU版本的TF的区别主要是计算速度。特别是进行模型训练时，好的显卡下，GPU应该会节约很多倍的时间。

### （4）安装CUDA8.0和CudnnV5.1（针对GPU版本）

注意，TF的GPU版本主要是针对英伟达系列的显卡，CUDA8.0和CudnnV5.1也是NVIDIA开发的用于利用显卡进行加速计算的软件。请确认自己电脑上的显卡是否支持CUDA，可以下载一个小程序GPU-Z进行检查，它用于检查显卡的很多信息，比如我的显卡：



如图所示，我的计算机显卡是GTX1050，支持CUDA，目前安装的显卡驱动版本是381.65。此外，还要求显卡的计算能力不小于3.0。满足以上要求，就可以下载相应的CUDA和Cudnn了。

CUDA下载地址：<https://developer.nvidia.com/cuda-downloads>

根据对应的windows系统，下载相应CUDA8.0.x版本。我下载的版本是：

cuda\_8.0.61\_win10.exe

Cudnn下载地址：<https://developer.nvidia.com/rdp/cudnn-download>

下载前需要注册登录，关键点：请选择下载CudnnV5.1版本。不要选择最新的V6.0版本。否则import tensorflow时会出错。我下载的Cudnn版本是：

cudnn-8.0-windows10-x64-v5.1.zip

补充：

a. 安装时CUDA时可能会遇到显卡驱动版本不符的问题，比如，因为CUDA安装前会自动进行系统兼容性检查，其实也就是寻找电脑上的显卡版本，看看是否兼容，CUDA安装时其中有一个安装组件就是显卡驱动；但如果电脑的显卡比较新，则进行系统兼容性检查时就不能识别显卡硬件，就会提示警告，我的显卡TGX1050比较新，就遇到了这个问题，我的显卡对应的驱动版本是381.56，但CUDA8.0自带的显卡驱动版本只有373.15。这时不用管它，继续安装完成，一般不用更改安装目录，一切默认完成。最后需要做一件事情：下载安装GeForce Experience，进行一下显卡更新就好。

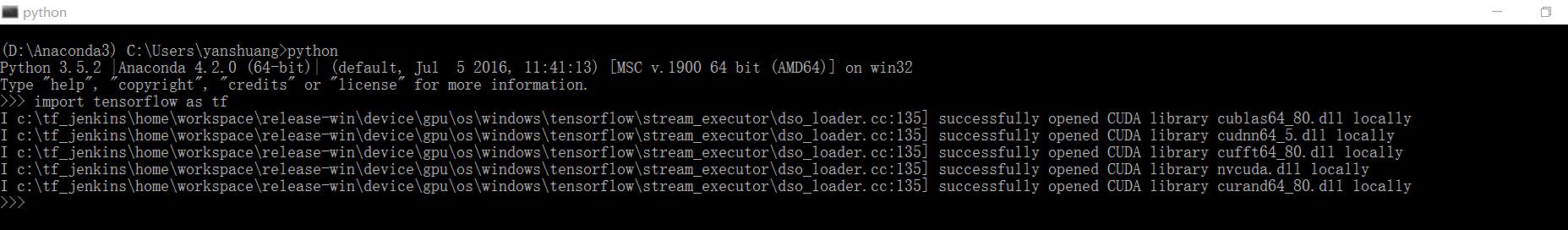
b. Cudnn下载下来后不用安装，将其解压，里面其实就只有三个文件，分别在bin/include/lib文件夹中，将这三个文件夹的上一级文件夹cuda放置到比如c盘下。这时需要做一件事：

将路径“C:\cuda\bin”添加到系统环境变量中。除了这种办法，还可以将cudnn下三个文件cudnn64\_5.dll、cudnn.h、cudnn.lib分别复制到CUDA安装目录下的bin/include/lib文件夹中。比如CUDA下bin文件夹的路径默认是：C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v8.0\bin。或者两个方法都执行。

### （5）完成测试及参考目录

以上步骤都正确执行之后，TF应该就顺利装上了。接下来在Python环境下执行测试。

打开“Anconda Prompt”窗口，输入“python”，即切换到了python环境。然后输入“import tensorflow as tf”，如果一切顺利应该会出现以下结果：



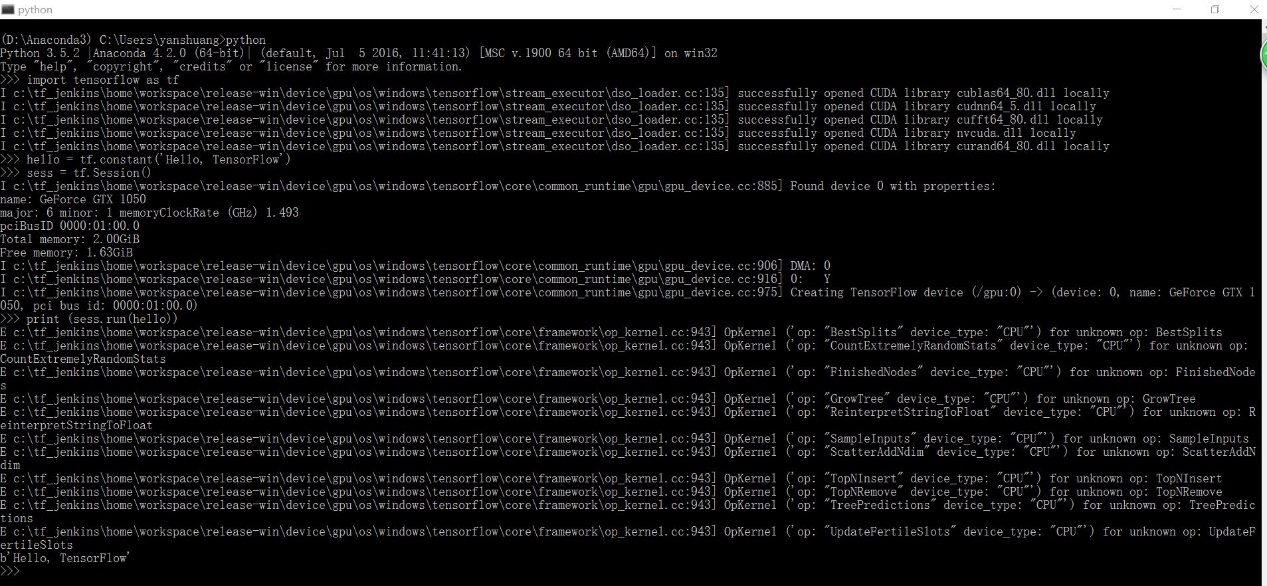
显示出所有CUDA库都成功打开。接着继续输入：

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

sess = tf.Session()

print (sess.run(hello))

会得到以下结果：



可以看到，程序找到了我的显卡，并最后成功输出了“Hello TensorFlow”。

TF安装工作全部完成。

有价值的参考目录：

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

<http://blog.csdn.net/u010099080/article/details/53418159>

<http://stackoverflow.com/questions/41007279/tensorflow-on-windows-couldnt-open-cuda-library-cudnn64-5-dll>

<http://blog.csdn.net/infovisthinker/article/details/54705826>

<http://blog.csdn.net/infovisthinker/article/details/55095816>

如果是安装的Python3.5.x（没有安装Anconda）TF-cpu版本请参考：

<http://blog.csdn.net/jonsen_hb/article/details/55657259>

TFr0.12版本的各种安装方式（pip installation/virtualenv installation/Anconda installation/Docker installation/Installing from source）请参考：

<https://www.tensorflow.org/versions/r0.12/get_started/os_setup#pip-installation-on-windows>

# （二）TensorBoard可视化工具的使用

## （#）Windows平台下如何正确打开TensorBoard

以训练MNIST数据集的工程为例（使用softmax回归算法），程序中，将summary汇总结果保存到了如下路径：



程序运行之后，生成了如下文件：

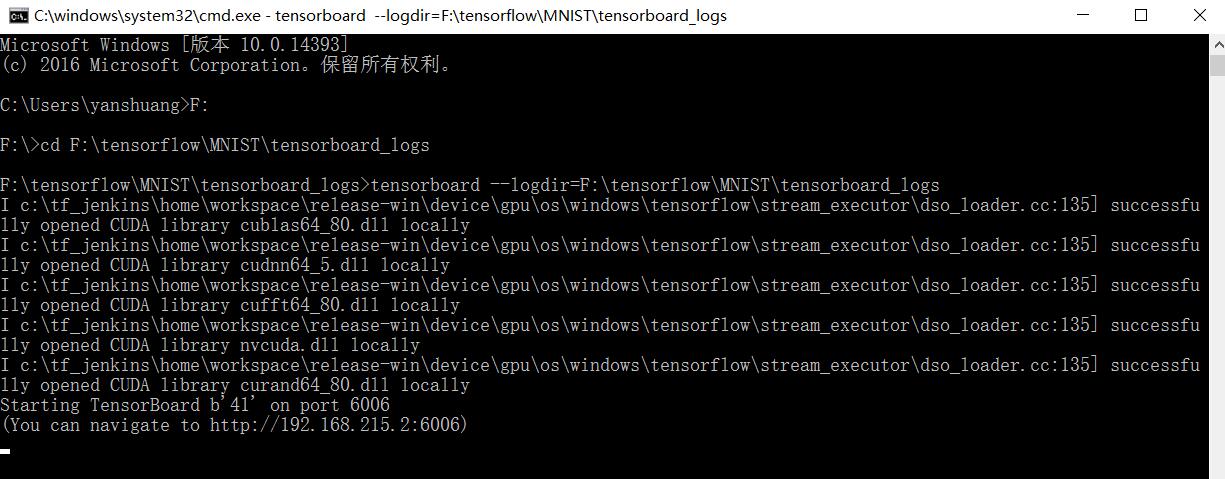


接下来执行以下步骤：

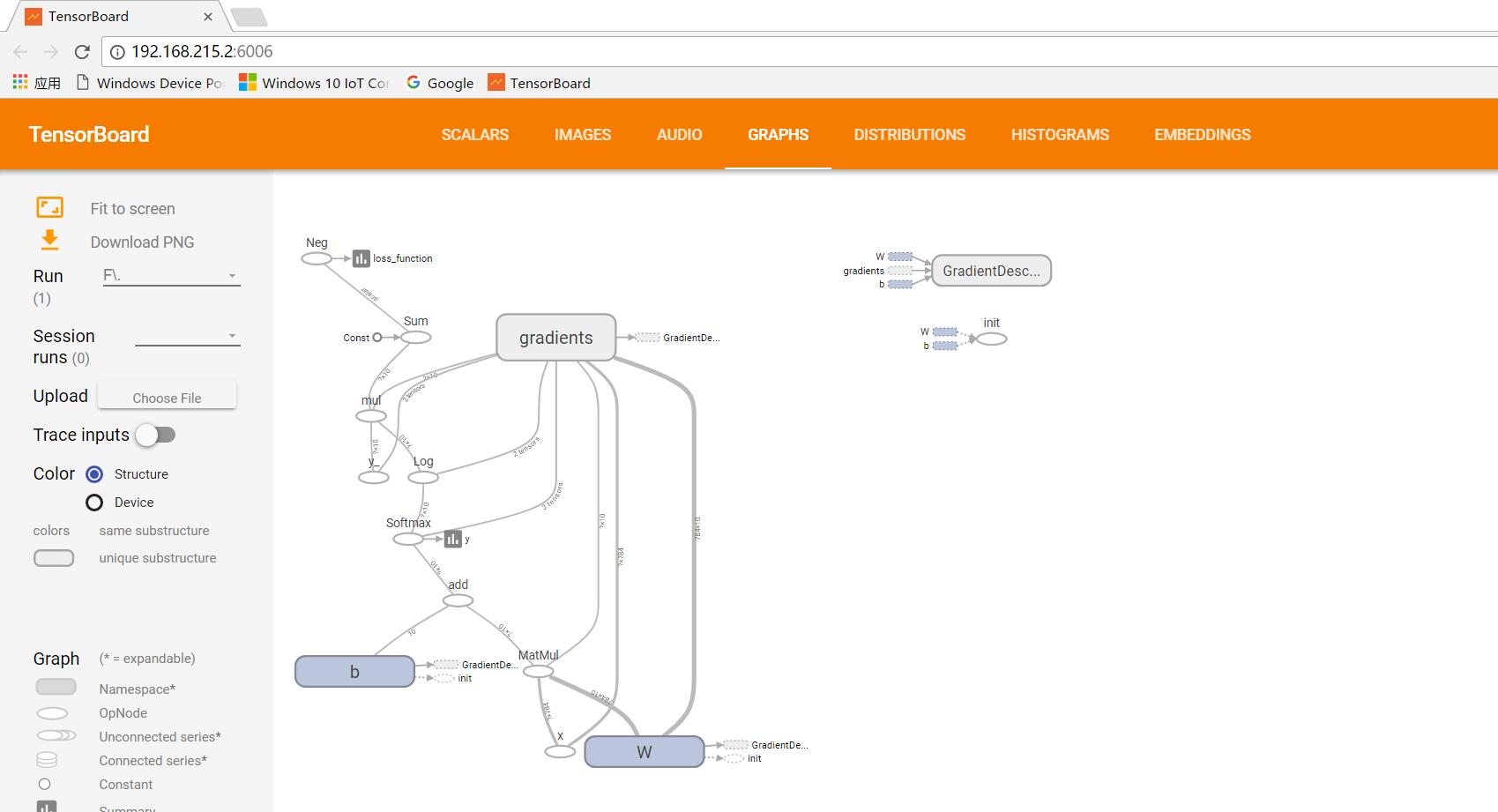
（1）打开CMD命令行终端；

（2）CD到路径“F:\tensorflow\MNIST\tensorboard\_logs”下，也就是存放汇总结果的文件夹。具体操作是：先输入“F:”+回车，进入F盘；接着输入“cd F:\tensorflow\MNIST\tensorboard\_logs”+回车，就进入了指定文件夹；（注意cd后有空格符）

（3）输入“tensorboard --logdir=F:\tensorflow\MNIST\tensorboard\_logs”+回车，得到对应的浏览器中查看tensorboard结果的网址。如下图所示：



（4）复制得到的网址，在Chrome浏览器中粘贴该网站，就进入了tensorboard界面，比如点击到GRAPHS，可以查看到自己构造的模型结构：

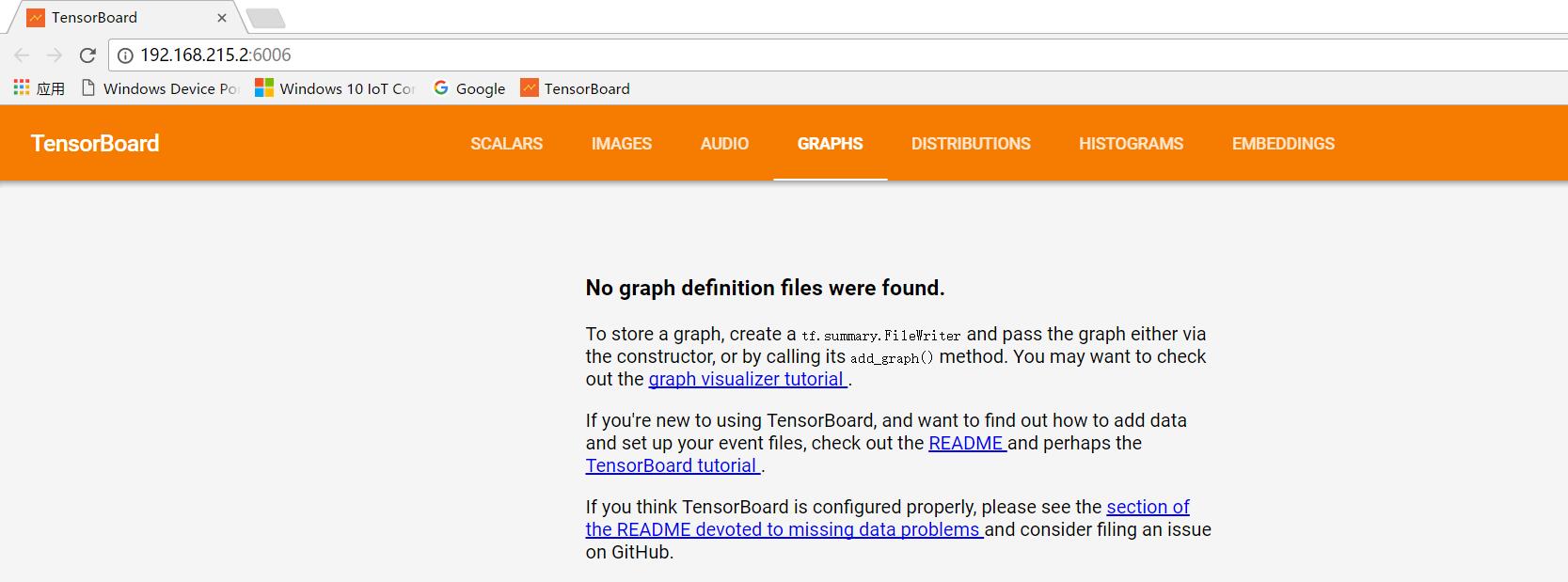


**注意点：**

（1）在cmd中必须先cd到汇总文件所在的文件夹（但是好像上一级文件夹也可以，对于上面的实例即“F:\tensorflow\MNIST”）；然后再输入命令：

tensorboard --logdir=F:\tensorflow\MNIST\tensorboard\_logs

如果打开cmd之后就直接输入以上命令的话，可以得到网址，但是在浏览器中会出现如下结果：



即找不到定义的图文件在哪儿。

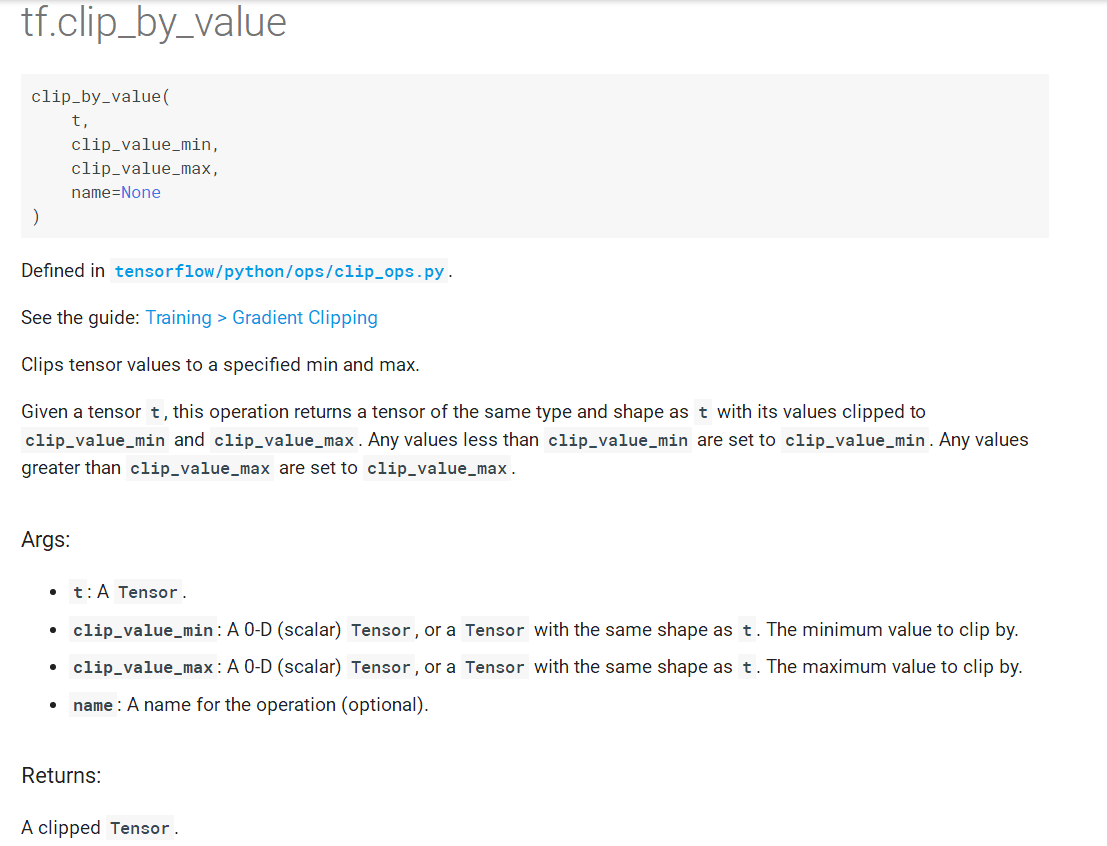
（2）注意一定要用Chrome浏览器，我用过360浏览器和IE浏览器都打不开。

## （##）参考网址

<https://tieba.baidu.com/p/5037861371>

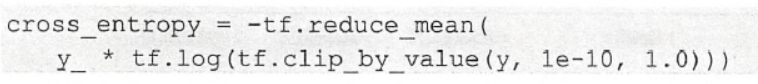
# （3）Tensorflow中比较重要的函数

## 1. tf.clip\_by\_value()函数



函数功能：

将一个tensor值限定在某个范围里面。比如在分类问题中经常用到的损失函数交叉熵：cross\_entropy时就可以用到这个函数：

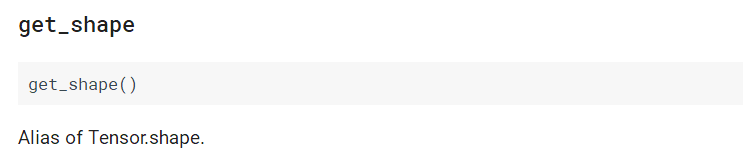


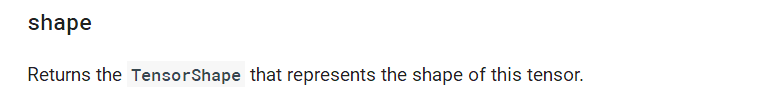
目的是防止当预测值y=0时，log(y)无效，这样可以避免一些难以发现的运算错误。

## 2 Tensor.get\_shape().as\_list()

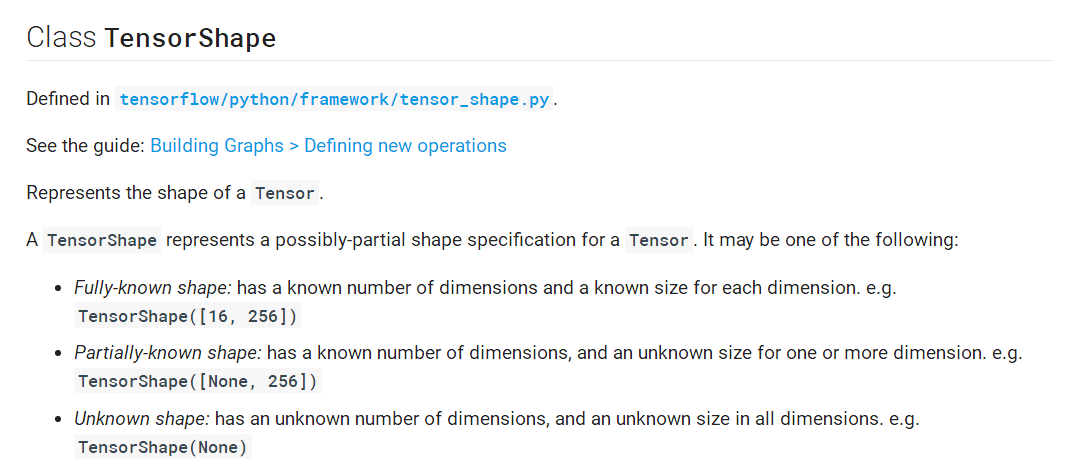
在卷积神经网络模型中，最后一个卷积层的输出(比如名为pool)是一个4D的tensor（batch,height,width,channels），下一个层是全连接层，需要将4D的pool展开成一个2D的tensor（比如名为reshape）（batch,height\*width\*channels），如何reshape?

一个好用的方法是调用pool.get\_shape().as\_list()。其中，pool.get\_shape()返回值是一个tf.TensorShape类，如下图所示：

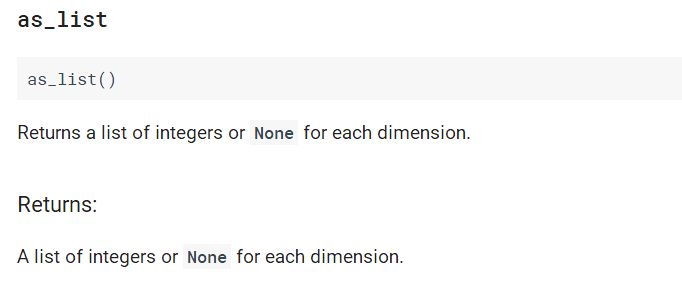




下面是tf.TensorShape类的简介：



此时，继续调用tf.TensorShape类的as\_list()方法，就可以方便地返回原来的pool每个维度的大小了：



该方法返回的是一个整数的list，此时：

reshape=tf.reshape(pool,[pool\_shape[0], pool\_shape[1] \* pool\_shape[2] \* pool\_shape[3]])

这样就不必准确地记住pool的每个维度的具体数值了。

## 3