原文链接：<https://blog.csdn.net/ccnucb/article/details/79873460>

**用GPU加速深度学习: Windows安装CUDA+TensorFlow教程**

2018年04月09日 21:56:06

阅读数：226

**雷锋网按：本文作者阿萨姆，本文首发于作者的知乎专栏《[数据说](https://zhuanlan.zhihu.com/aiinsight" \t "_blank)》，雷锋网(公众号：雷锋网)获其授权发布。**

**背景**

在Windows上使用GPU进行深度学习一直都不是主流，我们一般都首选Linux作为深度学习操作系统。**但很多朋友如果只是想要了解深度学习，似乎没有必要专门装双系统或者改用Linux。现实生活中，很多使用学校或者公司电脑的朋友也没有操作权限改换系统。**那么到底是否可以在Windows系统上设置深度学习框架，开发深度学习模型呢？

好消息是越来越多的深度学习框架开始支持Windows，这使得在Windows上使用GPU加速学习过程也变成了可能。**很多朋友虽然没有一块很强劲的显卡，但也可以以较低的代价来了解在GPU上运行深度学习模型的过程。值得欣喜的是，大部分Nvidia GeForce系列的显卡都可以使用CUDA，大部分有独显的笔记本理论上都可以使用GPU来“深度学习”。即使加速效果不明显，但很多入门级的显卡依然可以用于TensorFlow，这至少可以帮助大家了解和熟悉这个框架。**

**请不要让缺少预算或者系统不兼容成为探索路上的拦路虎和借口。**

本着探索的精神，我试着在两台Windows电脑上安装了CUDA(用于调用GPU)和TensorFlow(谷歌开发的深度学习工具库)，把这个过程记录下来和大家分享。

在开始之前想再次提醒大家，在Windows上安装CUDA和TensorFlow有很多坑，对于各种软件版本以及系统设置的要求可谓非常刁钻，很容易就“误入陷阱”。**请大家严格对照我的操作流程，以防出错。多图长文预警！！！使用教程时建议在电脑端阅读，可以放大图片。**

**准备步骤**

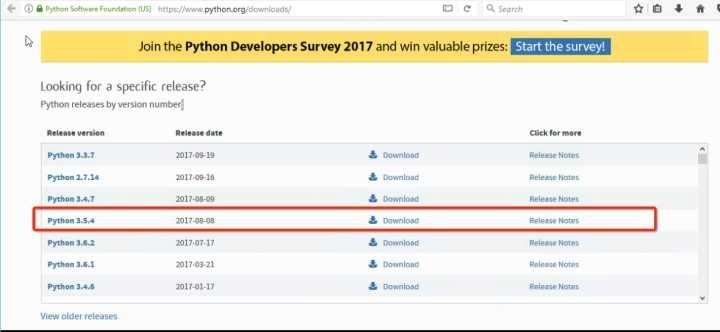
**请按照介绍下载以下软件备用，具体安装流程在第二部分介绍。**

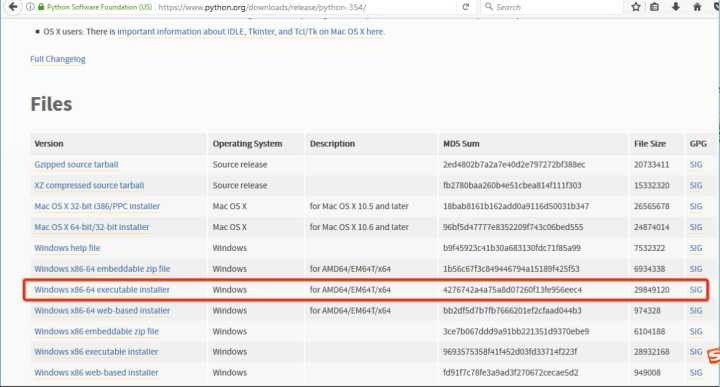
**简单的说，我们需要 Windows 7/8/10, Server 2012/2016 + Python 3.5 + Visual Studio 2015 + CUDA 8.0 + CuDnn 6.0。**

**1.操作系统要求和硬件要求:**

* Windows版本：Windows 7，Windows 8，Windows 10，Windows Server 12/16
* 显卡版本：请对照英伟达提供的支持CUDA的显卡列表，建议查看英文版。不知道自己显卡版本的可以通过“设备管理器”查看，或者使用第三方软件GPU-Z查看。请注意，AMD的显卡不可以使用英伟达开发的CUDA...
* 中文列表：[CUDA - 支持CUDA的GPU - NVIDIA(英伟达)](http://t.cn/RjHSpCY" \t "_blank)
* 英文列表：[CUDA GPUs](http://t.cn/zjYolU1" \t "_blank)

**2.Python版本：**64位版本的Python 3.5。注意Python3.6和2.7都不可以。地址和版本如下图所示。如果使用错误的Python版本，将无法安装TensorFlow！





**3.Visual Studio版本：**

* 我们使用的CUDA8.0**不支持Visual Studio 2017，使用VS2017会报错。**
* Visual Studio 2015的话现在最新的版本是**Visual Studio 2015 with Update 3**，网上有人说不能用Update 3这个版本，但根据我的亲测可以使用VS2015 with Update 3。下载免费的Community版本即可:<https://imagine.microsoft.com/en-us/Catalog/Product/101>。这个链接是ISO版本，大家也可选择EXE版本安装。其他VS2015的版本，如企业版(Enterprise)或者专业版(Professional)也可以，事实上我测试的时候安装的是企业版。
* Visual Studio 2013根据文档介绍也可以，但我并没有实际测试，不推荐使用。

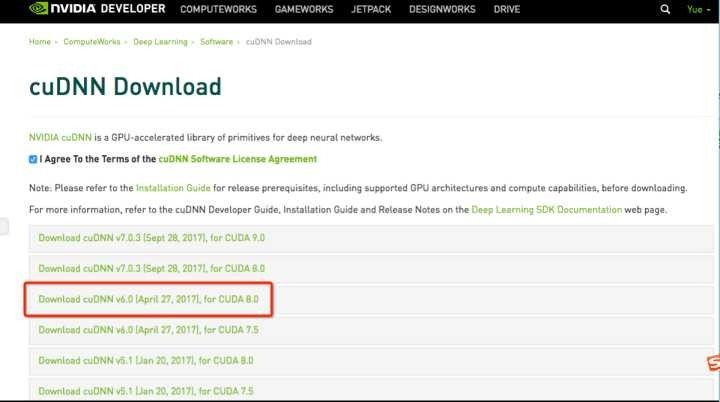
如果你的电脑上已经安装了VS2017或者VS2013，可以同时下载安装VS2015，并不存在冲突。

**4.CUDA版本：**TensorFlow的支持的CUDA版本截止到今天（2017年10月1日）还是 CUDA 8.0。**千万要注意英伟达官网上的默认版本是CUDA 9.0，请不要下载安装这个版本。**

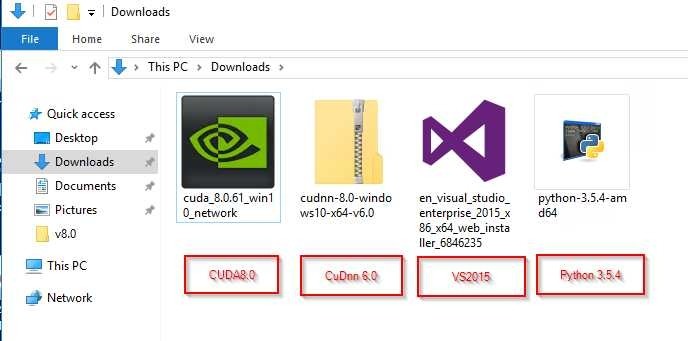
正确的8.0版本下载地址: [CUDA Toolkit 8.0 - Feb 2017](http://t.cn/ROVLbvu) 。网络比较好的朋友建议下载网络安装版本，不要下载本地版。

**5.CuDnn版本：CuDnn 6.0** for CUDA8.0。这是唯一可以使用的版本，下载时请对应的你的操作系统选择合适的版本。

请特别注意下载CuDnn前必须注册英伟达社区的[会员](http://t.cn/RYIJenh" \t "_blank)，虽然说是英文的但非常简单。 正确的下载版本如下图所示：



**6.所有必备文件：**

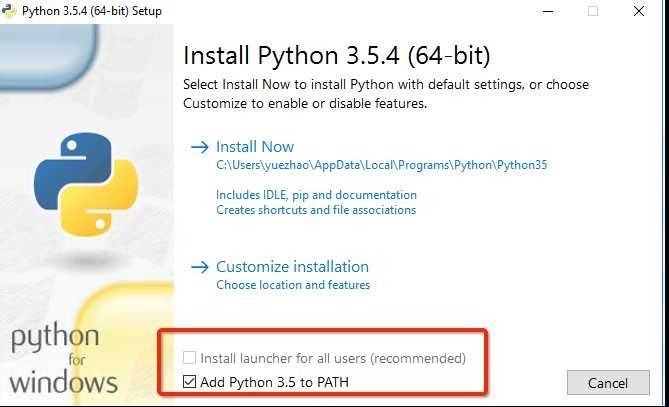


**安装步骤(建议使用管理员权限账号)**

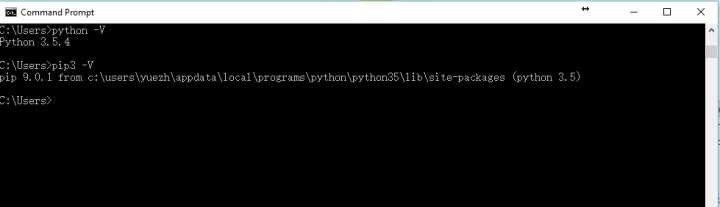
**请尽量按照本文的安装顺序进行安装，否则有可能运行失败！最重要的是，安装CUDA必须在安装Visual Studio之后，这两个顺序不可以调换。**

**1.安装Python**

建议直接用Python的原生安装包，安装时建议勾选把Python加到系统路径当中去。



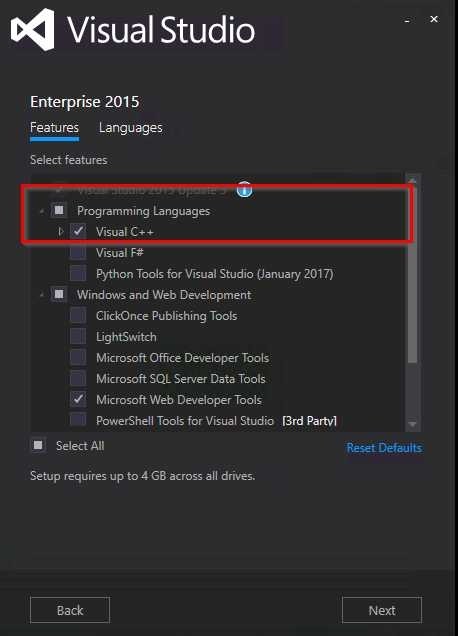
安装完成后检查Python的版本和PiP3的版本，那就是系统路径中的默认Python是3.54，且Pip3的版本大于8.01。



检测方法如上图，打开命令行分别输入 "python -V" 和 “pip3 -V”即可，正确的输出如上图。

**2.安装Visual Studio 2015**

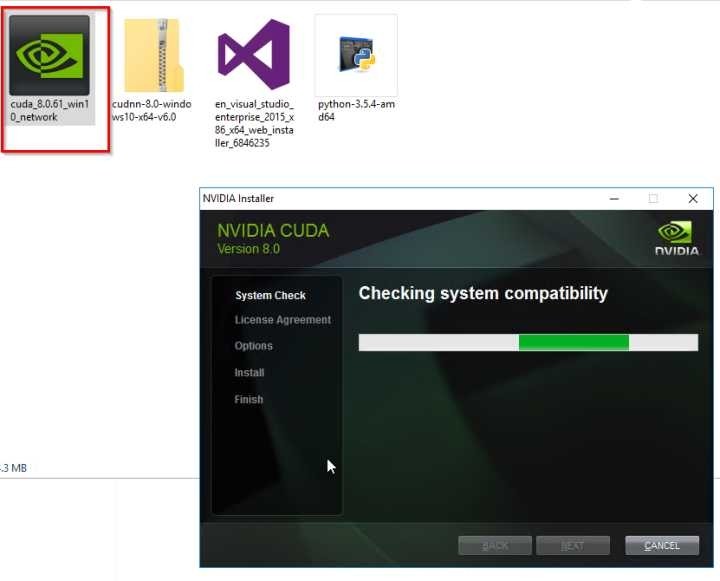
安装VS2015没什么难度，**只有一个选项需要自定义，别的选项都可以使用默认值**。VS2015的默认安装不包括C++的编译器，必须手动勾选Visual C++，不然会面临后续的CUDA编译错误。



主要原因是VS2015在安装时并没有默认安装C++的编译器，也就是CL.exe。在后文中我会提到如果你没装Visual C++系统会如何报错。

**3.安装CUDA**

**安装CUDA前请务必确认VS2015安装成功！**这个时候你需要安装CUDA，双击我们下载的安装文件即可，一切都选择默认即可。

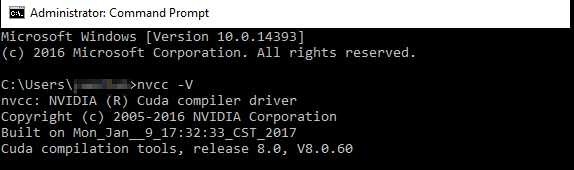


安装成功后会看到如下图：



**4.验证CUDA安装成功:**

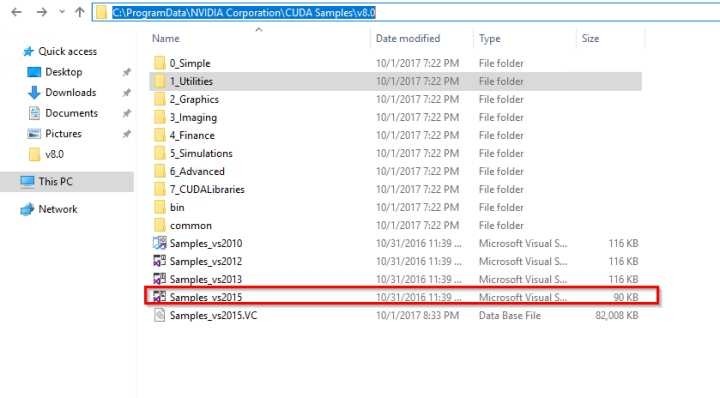
**4.1. 打开命令行**，也就是cmd然后输入“nvcc -V”，如果安装正确的话你应该看到这样的输出：



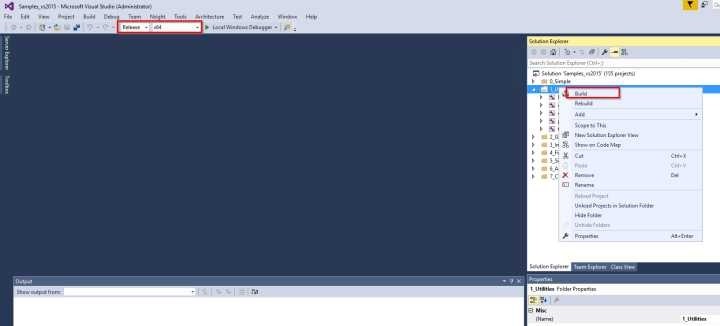
输出中显示了CUDA的版本是release 8.0。

**4.2. 使用VS和CUDA编译测试文件**

进入“C:\ProgramData\NVIDIA Corporation\CUDA Samples\v8.0”文件夹，双击打开 “Samples vs2015”这个文件。



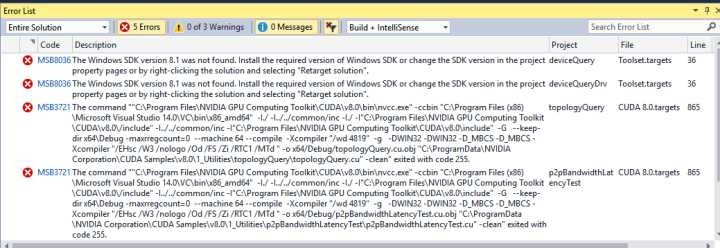
选择编译生成1\_Utilities中所有的文件。具体操作就是在1\_Utilities上右键选择Build，注意红框部分的64位和Release：



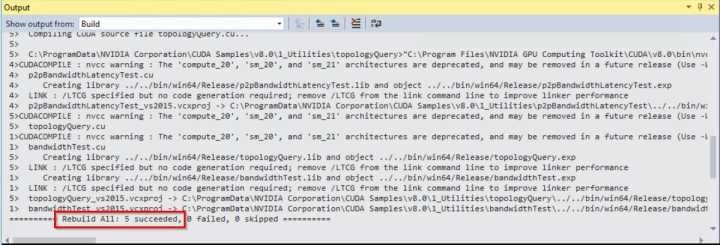
这个时候，你可能遇到了编译错误，如果你发现错误提示是无法找到 CUDA.prop那么就是CUDA安装错误，建议检查CUDA版本重新安装。

如果你看到这两个错误:

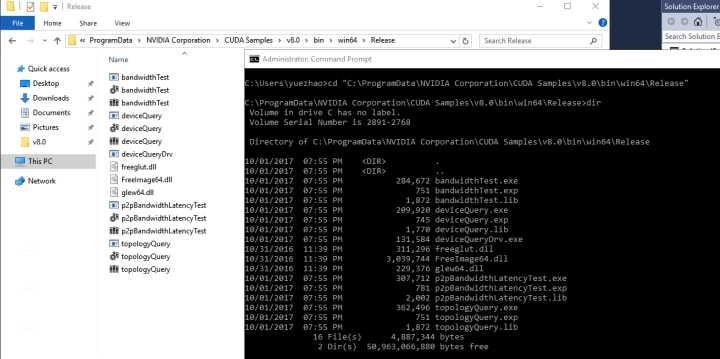
* **TRK0005:** Failed to locate: "CL.exe". The system cannot find the file specified.系统找不到CL.exe，无法编译文件。**面对这个错误那就是你没选Visual C++，需要重新安装时选上这个组件。**
* **MSB8036:** Windows SDK 8.1 was not found.如果你同时看到了这个错误说你没有安装 Windows SDK 8.1，**那也可以通过安装C++ Redistribution这个来解决。**



如果你没有遇到编译错误，那么应该看到下图提示5个文件编译成功：

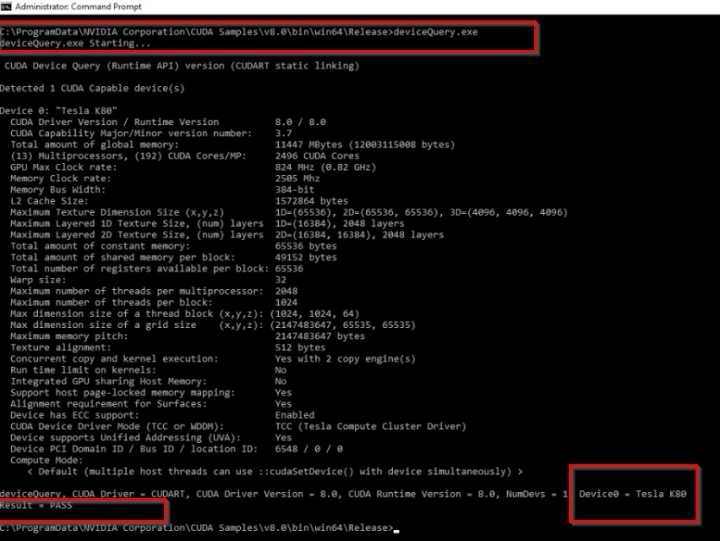


在成功后，你会发现在“C:\ProgramData\NVIDIA Corporation\CUDA Samples\v8.0\bin\win64\Release”文件夹中出现了一堆文件，我们主要需要deviceQuery和bandwidthTest。

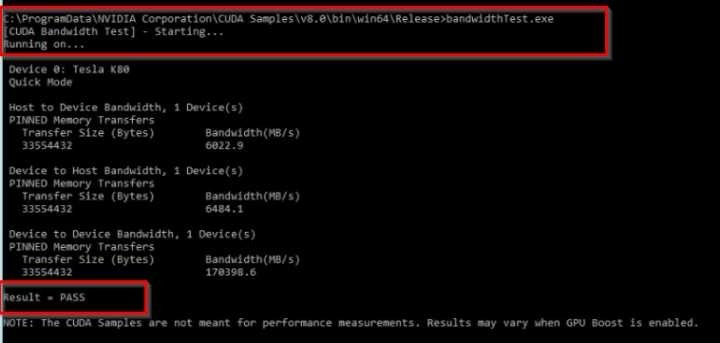


**4.3. deviceQuery和bandwidthTest验证**

运行我们刚才编译出来的deviceQuery.exe，也就是在cmd中运行这个文件，下图中左下的红框显示 result = pass代表安装测试成功，右下的红框是你的显卡型号请确认型号正确。Tesla K80只是我的这台机器型号，这个每个人可能都不一样。

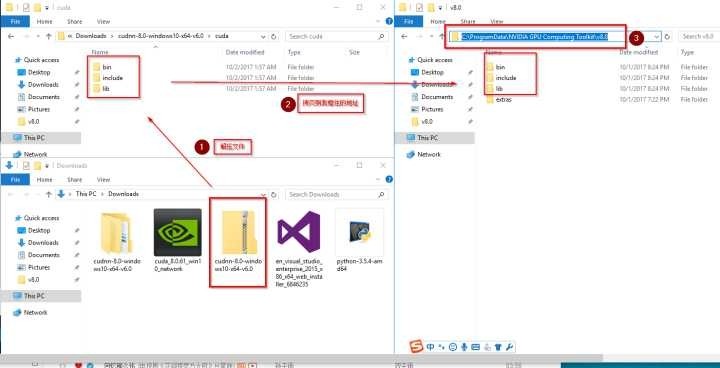


运行我们刚才编译出来的bandwidthTest.exe，方法一样，也是关注是否result = PASS。



**5.安装CuDnn**

解压缩我们下载的CuDnn文件，得到3个文件夹：bin, include, lib。如下图所示，将这个三个文件夹复制到“C:\ProgramData\NVIDIA GPU Computing Toolkit\v8.0”。



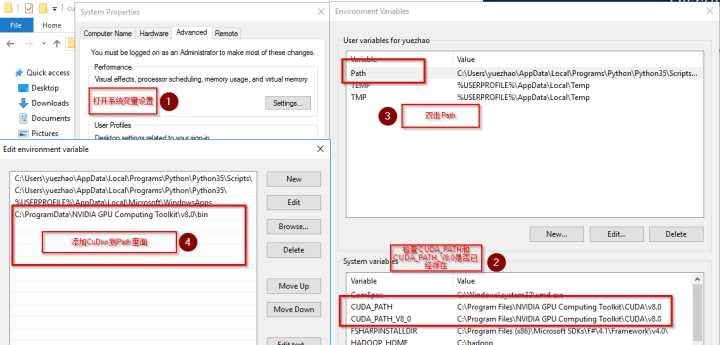
**6.确认系统环境变量(Environment Variables)**

确认Python3.5.4.和Pip3在系统环境变量中，检查的方法上面已经介绍了。

打开系统环境变量设置，以Win10为例，请参考百度经验（http://t.cn/RYIJJht）介绍。

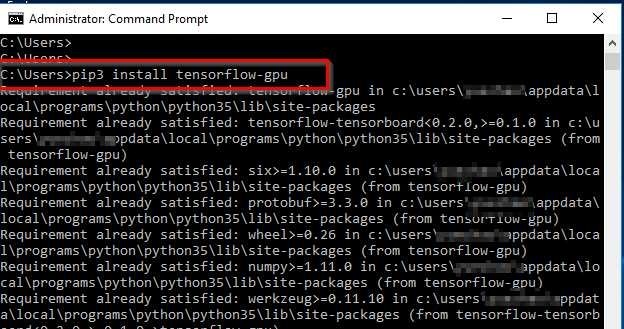
确认CUDA\_PATH和CUDA\_PATH\_V8.0已经存在

**手动添加 “C:\ProgramData\NVIDIA GPU Computing Toolkit\v8.0\bin”到Path里面。**

****

**7.安装TensorFlow的GPU版本**

打开cmd，输入“pip3 install tensorflow-gpu”



我因为已经安装过了，所以显示已经安装。此处要注意三点：

* 要用pip3而不是pip
* 要安装tensorflow-gpu，而不是tensorflow
* 如果安装失败，很有可能你的Python版本不是3.5.，或者pip3版本太低，可以使用"pip3 install --upgrade pip3"来升级pip3

**第一个TensorFlow程序！**

恭喜你，我们离胜利已经一步之遥了:) 让我们来验证我们安装的TensorFlow可以使用GPU！

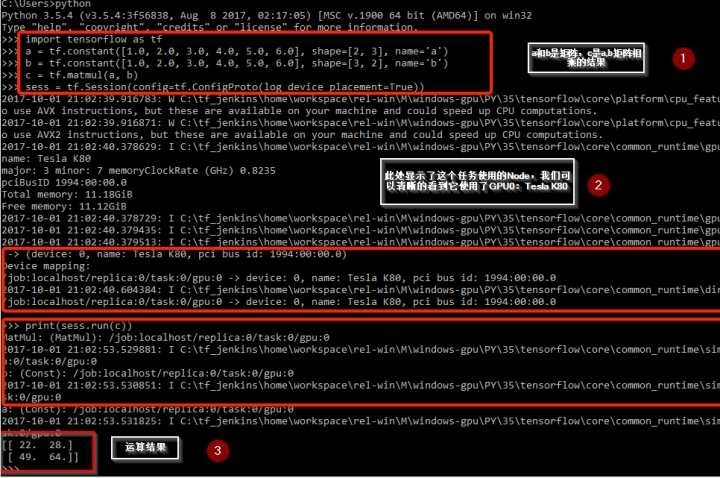
打开cmd，输入以下指令打开python的interactive shell。

首先导入tensorflow:

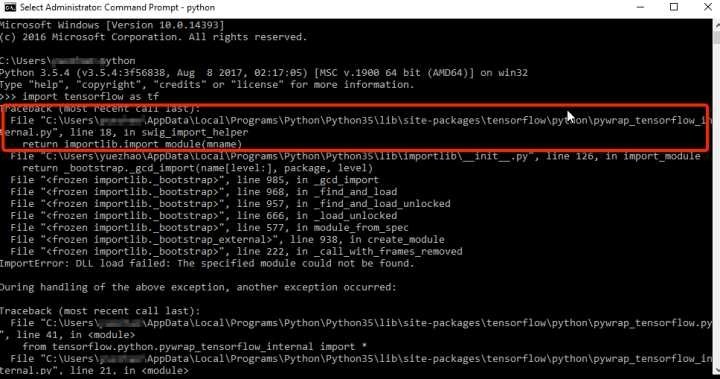
import tensorflow as tf

接着每次输入一行代码，并回车，你应该可以看到下图中表明你的GPU已经开始工作啦~

a = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[2, 3], name='a')  
b = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[3, 2], name='b')  
c = tf.matmul(a, b)  
  
sess = tf.Session(config=tf.ConfigProto(log\_device\_placement=True))  
  
print sess.run(c)



你可能在import TensorFlow时遇到了错误，如下图。这是因为你的CuDnn设置错误，请参考本文确认系统环境变量(Environment Variables)来修复。



**写在最后**

实践是检验真理的唯一标准，时间也是。在所有人都在大力鼓吹深度学习的今天，不管你喜欢还是反对，我都希望你可以亲自试试，感受一下:)

而现实生活中我们都有各种各样的制约，比如系统版本限制、比如有限的显卡预算。**这也是我写这篇教程的目的，我不希望这些原因成为阻碍你探索研究的障碍。**希望我的这篇文章为你的深度学习之路做出了一点微小的贡献。

**欢迎来到深度学习的世界 ʕ•ᴥ•ʔ**