1.在win10上安装bazel

在https://github.com/bazelbuild/bazel/releases上下载bazel-0.19.2-windows-x86\_64.exe（对应tensorflow-gpu 1.13.0版本），速度有时很慢，无法下载，有时快，可以下载，不行的话多试几次。

早期参考的资料：<https://blog.csdn.net/qq_24800377/article/details/82857018>

<https://blog.csdn.net/robothn/article/details/86157724>的bazel部分

里面的一些细节都跟我实际的操作有点出入，但大体的过程是一样的

2.下载tensorflow源码

在https://github.com/tensorflow/tensorflow上下载各种版本的tensorflow源码。

我下载了好几个版本，最终编译成功了2个版本，一个是tensorflow-1.14.0的cpu版本，对应的bazel版本是bazel-0.25.2-windows-x86\_64.exe；一个是tensorflow-1.13.0的gpu版本，对应的bazel版本就是不很容易下载下来的bazel-0.19.2-windows-x86\_64.exe。

1. 编译过程

打开cmd

cd E:\ML\_DL\tensorflow\_C++\_API\tensorflow-r1.14

python ./configure.py

bazel build --config opt //tensorflow:libtensorflow\_cc.so

bazel build --config=opt --config=monolithic //tensorflow:libtensorflow\_cc.so

cpu版本库的指令（任选一个）

最开始编译大概等待了半个小时吧

其中编译tensorflow-1.14.0的cpu版本时碰到一个下载错误的问题，再重新编译可能下载就又好了。

接着再编译tensorflow-1.14.0的gpu版本时编译出错，报的错误在网上查不到相关资料，然后果断换版本编译，就换到tensorflow-1.13.0版本来编译gpu版的库，很顺利一下就编译成功。

cd E:\ML\_DL\tensorflow\_C++\_API\tensorflow-r1.13

python ./configure.py

bazel build --config=opt --config=cuda //tensorflow:libtensorflow\_cc.so （编译成功）

bazel build --config=opt --config=cuda --copt=-nvcc\_options=disable-warnings //tensorflow:libtensorflow\_cc.so（编译成功）

gpu版本库的指令（任选一个，优选下面的那个）

不知道是不是因为上面编译过程中已经下载了一些需要的文件，这次要快很多，大概等待几分钟就显示编译成功了。

参考资料：

<https://tensorflow.google.cn/install/source_windows>

<https://blog.csdn.net/robothn/article/details/86157724>的bazel部分

<https://blog.csdn.net/jin739738709/article/details/86705735>的bazel部分

部分细节跟我的实际操作不一样。

资料中也提到可以直接下载一些编译好的库，但是由于各种版本的对应问题，不容易找到跟自己电脑已有的环境完全相配的，如果能自己编译成功就最好，实在编译不成功可以考虑寻找已经编译好的库直接下载拿来试着使用。

1. 上面编译出来的是.so的库，在查找VS C++如何调用这个库的时候发现了bazel也可以编

译出lib和dll的库，参考下面资料中的方法

<https://blog.csdn.net/qq_35975447/article/details/91636625>

<https://blog.csdn.net/sinat_19758811/article/details/90896639>

cd E:\ML\_DL\tensorflow\_C++\_API\tensorflow-r1.13

python ./configure.py

bazel build --config=opt --config=cuda //tensorflow/tools/lib\_package:libtensorflow（这个编译成功）

bazel build --config=opt --config=cuda --copt=-nvcc\_options=disable-warnings //tensorflow/tools/lib\_package:libtensorflow（这个编译开始报关于nvcc的错，第二天再编译又是成功的）

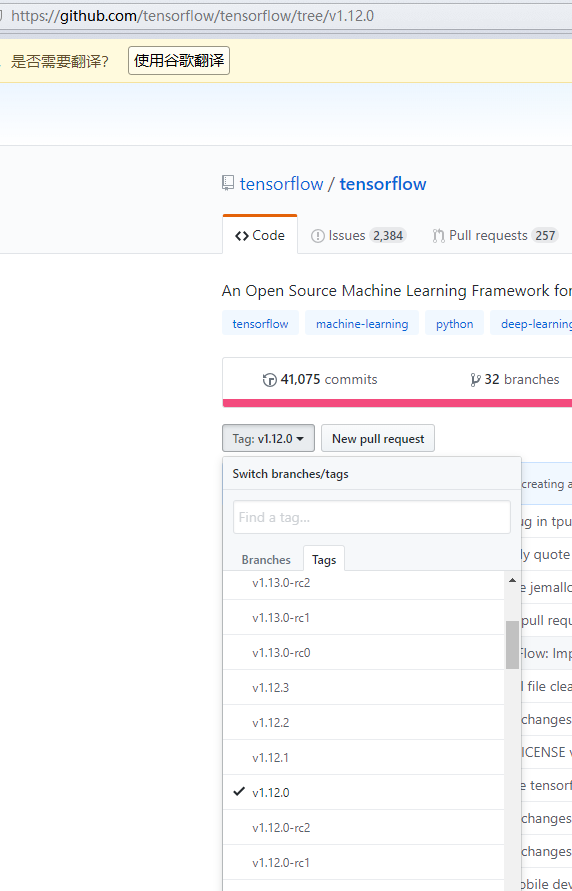
上面直接修改.so文件为.dll文件的方法最后验证还时不行，在编译时不能生成解决方案，报错26个无法解析的外部符号，如： 严重性 代码 说明 项目 文件 行 禁止显示状态 错误 LNK2019 无法解析的外部符号 "public: \_\_cdecl tensorflow::GraphDef::GraphDef(void)" (??0GraphDef@tensorflow@@QEAA@XZ)，该符号在函数 main 中被引用 cjTensorflowTest E:\ML\_DL\C++调用keras模型\cjTensorflowTest\cjTensorflowTest\cjTensorflowTest.obj 1

1. 在上述作者的文章下留言疑问，作者快速回复后参考他的另一篇文章

<https://blog.csdn.net/qq_35975447/article/details/91986142>

[https://tensorflow.google.cn//install/source\_windows?hl=zh-cn](https://tensorflow.google.cn/install/source_windows?hl=zh-cn) --tensorflow中文官网

选tags下载源码，不是Branches



打开cmd 在E:\ML\_DL\tensorflow\_C++\_API\tensorflow-1.12.0下执行命令：

python -m venv venv 创建一个虚拟环境；以管理员方式打开Windows PowerShell，进行编译：

（更多的可以参考配置虚拟环境<https://www.jb51.net/article/144859.htm>）

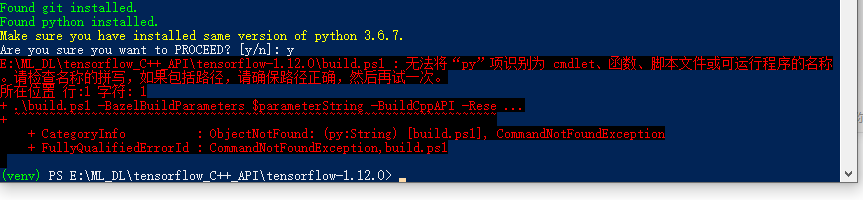
cd E:\ML\_DL\tensorflow\_C++\_API\tensorflow-1.12.0

$parameterString = "--config=opt --config=cuda --define=no\_tensorflow\_py\_deps=true --copt=-nvcc\_options=disable-warnings //tensorflow:libtensorflow\_cc.so --verbose\_failures"

.\build.ps1 -BazelBuildParameters $parameterString -BuildCppAPI -ReserveSource -ReserveVenv



问题1：

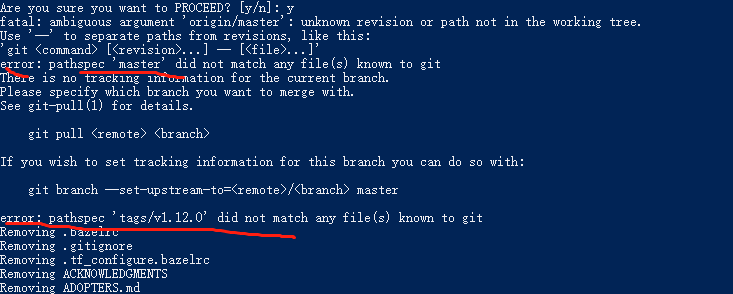


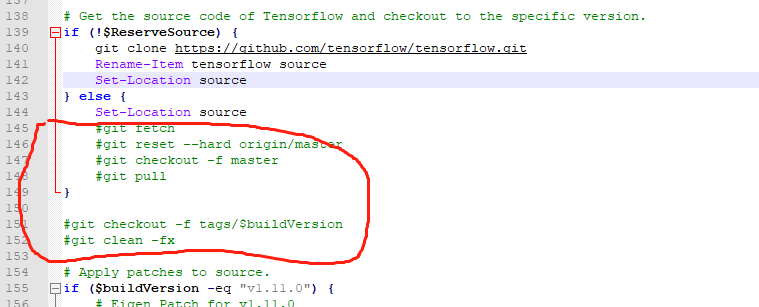
原因：PowerShell找不到python解释器的路径

重新下载个python安装，不用anaconda里的python

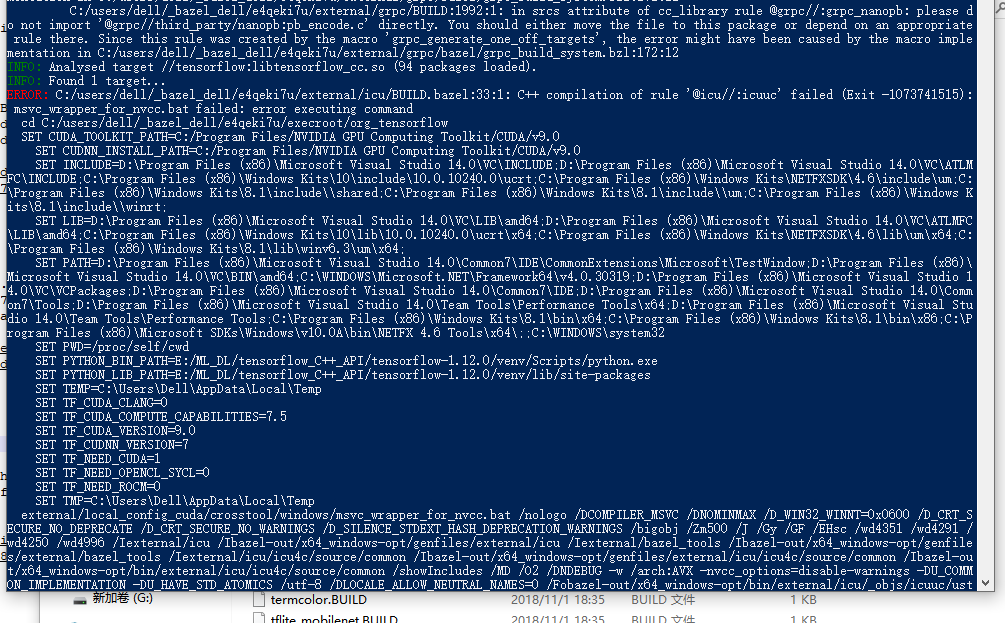
<https://blog.csdn.net/benben513624/article/details/80066136>

问题2：





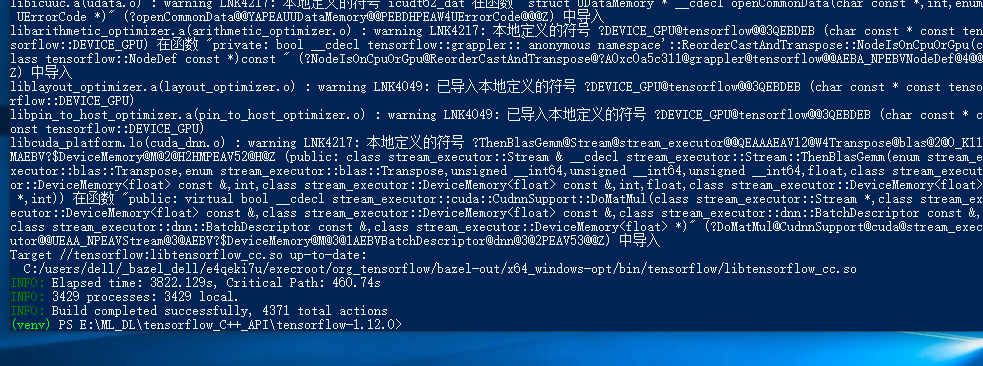
问题3：



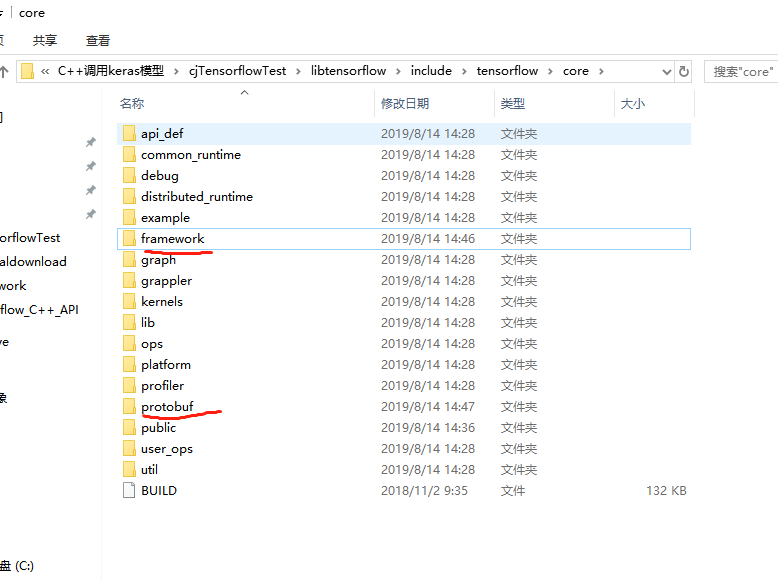
BUILD.bazel:33:1: C++ compilation of rule '@icu//:icuuc' failed (Exit -1073741515): msvc\_wrapper\_for\_nvcc.bat failed: error executing command

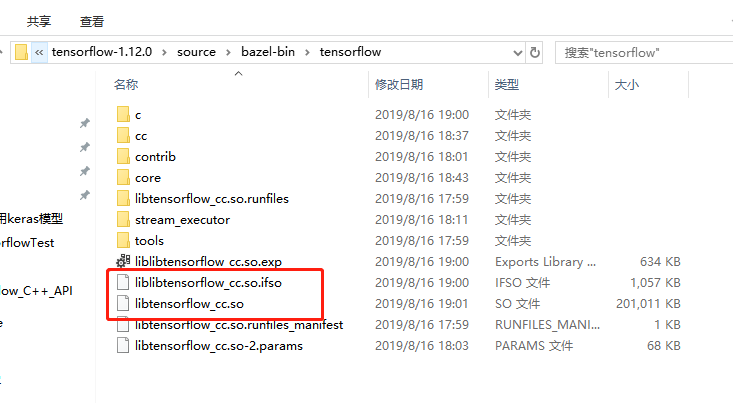
是因为C:\Users\Dell\\_bazel\_Dell\e4qeki7u\external里要下载的内容下载失败，编译过程中会反复多次出现这样的下载失败的问题，再重新编译直到下载OK编译成功；

下面是编译成功的截图



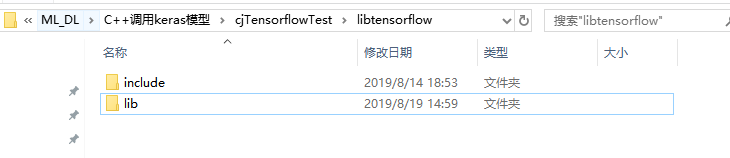
最后在整理库文件中的include文件时，下面划红线的文件夹需要从bazel-genfiles文件夹里再拷贝一些编译时报缺失的文件过来





dll:将上面的libtensorflow\_cc.so复制到下面的lib中并改名为tensorflow\_cc.dll

lib:将上面的liblibtensorflow\_cc.so.ifso复制到下面的lib中并改名为tensorflow\_cc.lib



注意：编译过程中可能需要关闭360杀毒软件，不然可能会自动的把bazel.exe和mysys64里的一些文件当病毒删掉了，导致编译失败