时间：2019/2/26

软件下载

VS2015社区版

Anaconda3-4.2.0-Windows-x86\_64（python3.5）

cuda\_8.0.61\_win10

cuda\_8.0.61.2\_windows(补丁包)

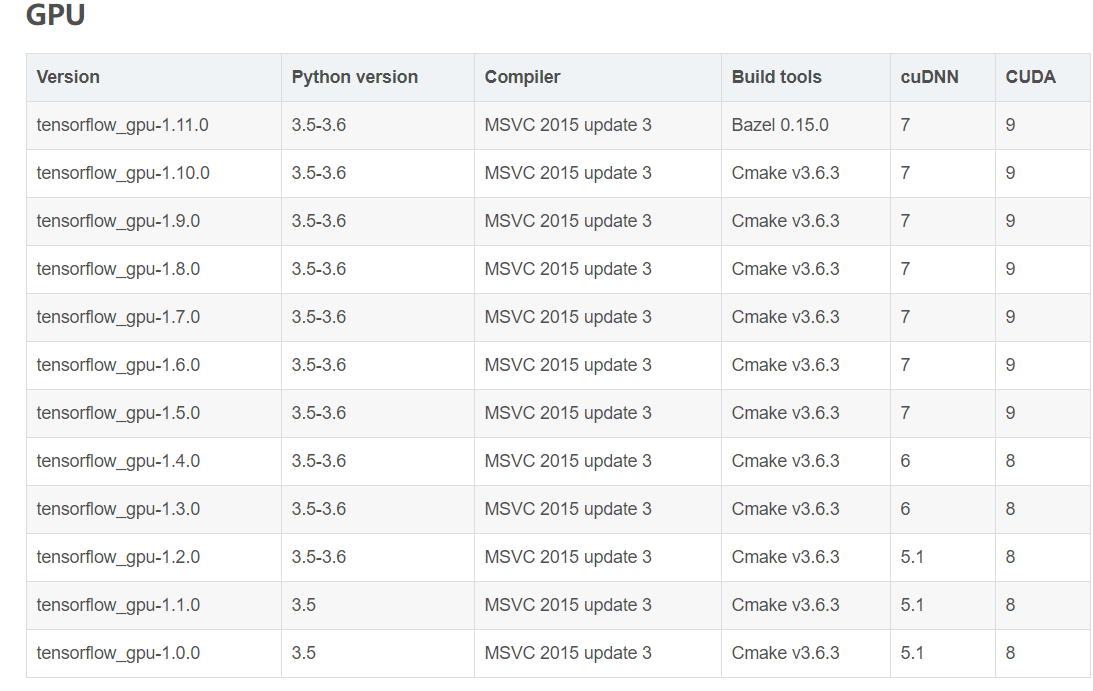
cudnn-8.0-windows10-x64-v5.1

pip-19.0.3.tar

（该配置下若要安装TensorFlow，对应cuda8.0和cudnn5.1以及python3.5的是TensorFlow1.2.0）

pip uninstall tensorflow-gpu==1.12.0

pip install tensorflow-gpu==1.2.0



cmake-3.13.4-win64-x64

安装上述软件，配置环境变量

https://github.com/BVLC/caffe/tree/windows下载的“caffe-windows”压缩包解压到project某一个文件夹中得到caffe文件夹，并在powershell进行编译：

cd caffe scripts\build\_win.cmd

**build\_win.cmd：内容如下**

@echo off

@setlocal EnableDelayedExpansion

:: Default values

if DEFINED APPVEYOR (

echo Setting Appveyor defaults

if NOT DEFINED MSVC\_VERSION set MSVC\_VERSION=14

if NOT DEFINED WITH\_NINJA set WITH\_NINJA=1

if NOT DEFINED CPU\_ONLY set CPU\_ONLY=1

if NOT DEFINED CUDA\_ARCH\_NAME set CUDA\_ARCH\_NAME=Auto

if NOT DEFINED CMAKE\_CONFIG set CMAKE\_CONFIG=Release

if NOT DEFINED USE\_NCCL set USE\_NCCL=0

if NOT DEFINED CMAKE\_BUILD\_SHARED\_LIBS set CMAKE\_BUILD\_SHARED\_LIBS=0

if NOT DEFINED PYTHON\_VERSION set PYTHON\_VERSION=2

if NOT DEFINED BUILD\_PYTHON set BUILD\_PYTHON=1

if NOT DEFINED BUILD\_PYTHON\_LAYER set BUILD\_PYTHON\_LAYER=1

if NOT DEFINED BUILD\_MATLAB set BUILD\_MATLAB=0

if NOT DEFINED PYTHON\_EXE set PYTHON\_EXE=python

if NOT DEFINED RUN\_TESTS set RUN\_TESTS=1

if NOT DEFINED RUN\_LINT set RUN\_LINT=1

if NOT DEFINED RUN\_INSTALL set RUN\_INSTALL=1

:: Set python 2.7 with conda as the default python

if !PYTHON\_VERSION! EQU 2 (

set CONDA\_ROOT=C:\ProgramData\Anaconda2

)

:: Set python 3.5 with conda as the default python

if !PYTHON\_VERSION! EQU 3 (

set CONDA\_ROOT=C:\ProgramData\Anaconda3

)

set PATH=!CONDA\_ROOT!;!CONDA\_ROOT!\Scripts;!CONDA\_ROOT!\Library\bin;!PATH!

:: Check that we have the right python version

!PYTHON\_EXE! --version

:: Add the required channels

conda config --add channels conda-forge

conda config --add channels willyd

:: Update conda

conda update conda -y

:: Download other required packages

conda install --yes cmake ninja numpy scipy protobuf==3.1.0 six scikit-image pyyaml pydotplus graphviz

if ERRORLEVEL 1 (

echo ERROR: Conda update or install failed

exit /b 1

)

:: Install cuda and disable tests if needed

if !WITH\_CUDA! == 1 (

call %~dp0\appveyor\appveyor\_install\_cuda.cmd

set CPU\_ONLY=0

set RUN\_TESTS=0

set USE\_NCCL=1

) else (

set CPU\_ONLY=1

)

:: Disable the tests in debug config

if "%CMAKE\_CONFIG%" == "Debug" (

echo Disabling tests on appveyor with config == %CMAKE\_CONFIG%

set RUN\_TESTS=0

)

:: Disable linting with python 3 until we find why the script fails

if !PYTHON\_VERSION! EQU 3 (

set RUN\_LINT=0

)

) else (

:: Change the settings here to match your setup

:: Change MSVC\_VERSION to 12 to use VS 2013

if NOT DEFINED MSVC\_VERSION set MSVC\_VERSION=14

:: Change to 1 to use Ninja generator (builds much faster)

if NOT DEFINED WITH\_NINJA set WITH\_NINJA=0

:: Change to 1 to build caffe without CUDA support

if NOT DEFINED CPU\_ONLY set CPU\_ONLY=0

:: Change to generate CUDA code for one of the following GPU architectures

:: [Fermi Kepler Maxwell Pascal All]

if NOT DEFINED CUDA\_ARCH\_NAME set CUDA\_ARCH\_NAME=Auto

:: Change to Debug to build Debug. This is only relevant for the Ninja generator the Visual Studio generator will generate both Debug and Release configs

if NOT DEFINED CMAKE\_CONFIG set CMAKE\_CONFIG=Release

:: Set to 1 to use NCCL

if NOT DEFINED USE\_NCCL set USE\_NCCL=0

:: Change to 1 to build a caffe.dll

if NOT DEFINED CMAKE\_BUILD\_SHARED\_LIBS set CMAKE\_BUILD\_SHARED\_LIBS=0

:: Change to 3 if using python 3.5 (only 2.7 and 3.5 are supported)

if NOT DEFINED PYTHON\_VERSION set PYTHON\_VERSION=3

:: Change these options for your needs.

if NOT DEFINED BUILD\_PYTHON set BUILD\_PYTHON=1

if NOT DEFINED BUILD\_PYTHON\_LAYER set BUILD\_PYTHON\_LAYER=1

if NOT DEFINED BUILD\_MATLAB set BUILD\_MATLAB=1

:: If python is on your path leave this alone

if NOT DEFINED PYTHON\_EXE set PYTHON\_EXE=python

:: Run the tests

if NOT DEFINED RUN\_TESTS set RUN\_TESTS=0

:: Run lint

if NOT DEFINED RUN\_LINT set RUN\_LINT=0

:: Build the install target

if NOT DEFINED RUN\_INSTALL set RUN\_INSTALL=1

)

:: Set the appropriate CMake generator

:: Use the exclamation mark ! below to delay the

:: expansion of CMAKE\_GENERATOR

if %WITH\_NINJA% EQU 0 (

if "%MSVC\_VERSION%"=="14" (

set CMAKE\_GENERATOR=Visual Studio 14 2015 Win64

)

if "%MSVC\_VERSION%"=="12" (

set CMAKE\_GENERATOR=Visual Studio 12 2013 Win64

)

if "!CMAKE\_GENERATOR!"=="" (

echo ERROR: Unsupported MSVC version

exit /B 1

)

) else (

set CMAKE\_GENERATOR=Ninja

)

echo INFO: ============================================================

echo INFO: Summary:

echo INFO: ============================================================

echo INFO: MSVC\_VERSION = !MSVC\_VERSION!

echo INFO: WITH\_NINJA = !WITH\_NINJA!

echo INFO: CMAKE\_GENERATOR = "!CMAKE\_GENERATOR!"

echo INFO: CPU\_ONLY = !CPU\_ONLY!

echo INFO: CUDA\_ARCH\_NAME = !CUDA\_ARCH\_NAME!

echo INFO: CMAKE\_CONFIG = !CMAKE\_CONFIG!

echo INFO: USE\_NCCL = !USE\_NCCL!

echo INFO: CMAKE\_BUILD\_SHARED\_LIBS = !CMAKE\_BUILD\_SHARED\_LIBS!

echo INFO: PYTHON\_VERSION = !PYTHON\_VERSION!

echo INFO: BUILD\_PYTHON = !BUILD\_PYTHON!

echo INFO: BUILD\_PYTHON\_LAYER = !BUILD\_PYTHON\_LAYER!

echo INFO: BUILD\_MATLAB = !BUILD\_MATLAB!

echo INFO: PYTHON\_EXE = "!PYTHON\_EXE!"

echo INFO: RUN\_TESTS = !RUN\_TESTS!

echo INFO: RUN\_LINT = !RUN\_LINT!

echo INFO: RUN\_INSTALL = !RUN\_INSTALL!

echo INFO: ============================================================

:: Build and exectute the tests

:: Do not run the tests with shared library

if !RUN\_TESTS! EQU 1 (

if %CMAKE\_BUILD\_SHARED\_LIBS% EQU 1 (

echo WARNING: Disabling tests with shared library build

set RUN\_TESTS=0

)

)

if NOT EXIST build mkdir build

pushd build

:: Setup the environement for VS x64

set batch\_file=!VS%MSVC\_VERSION%0COMNTOOLS!..\..\VC\vcvarsall.bat

call "%batch\_file%" amd64

:: Configure using cmake and using the caffe-builder dependencies

:: Add -DCUDNN\_ROOT=F:\cuda ^::cudnn解压目录

:: below to use cuDNN

cmake -G"!CMAKE\_GENERATOR!" ^

-DBLAS=Open ^

-DCMAKE\_BUILD\_TYPE:STRING=%CMAKE\_CONFIG% ^

-DBUILD\_SHARED\_LIBS:BOOL=%CMAKE\_BUILD\_SHARED\_LIBS% ^

-DBUILD\_python:BOOL=%BUILD\_PYTHON% ^

-DBUILD\_python\_layer:BOOL=%BUILD\_PYTHON\_LAYER% ^

-DBUILD\_matlab:BOOL=%BUILD\_MATLAB% ^

-DCPU\_ONLY:BOOL=%CPU\_ONLY% ^

-DCOPY\_PREREQUISITES:BOOL=1 ^

-DINSTALL\_PREREQUISITES:BOOL=1 ^

-DUSE\_NCCL:BOOL=!USE\_NCCL! ^

-DCUDA\_ARCH\_NAME:STRING=%CUDA\_ARCH\_NAME% ^

-C %cd%\libraries\caffe-builder-config.cmake ^

"%~dp0\.."

if ERRORLEVEL 1 (

echo ERROR: Configure failed

exit /b 1

)

:: Lint

if %RUN\_LINT% EQU 1 (

cmake --build . --target lint --config %CMAKE\_CONFIG%

)

if ERRORLEVEL 1 (

echo ERROR: Lint failed

exit /b 1

)

:: Build the library and tools

cmake --build . --config %CMAKE\_CONFIG%

if ERRORLEVEL 1 (

echo ERROR: Build failed

exit /b 1

)

:: Build and exectute the tests

if !RUN\_TESTS! EQU 1 (

cmake --build . --target runtest --config %CMAKE\_CONFIG%

if ERRORLEVEL 1 (

echo ERROR: Tests failed

exit /b 1

)

if %BUILD\_PYTHON% EQU 1 (

if %BUILD\_PYTHON\_LAYER% EQU 1 (

:: Run python tests only in Release build since

:: the \_caffe module is \_caffe-d is debug

if "%CMAKE\_CONFIG%"=="Release" (

:: Run the python tests

cmake --build . --target pytest

if ERRORLEVEL 1 (

echo ERROR: Python tests failed

exit /b 1

)

)

)

)

)

if %RUN\_INSTALL% EQU 1 (

cmake --build . --target install --config %CMAKE\_CONFIG%

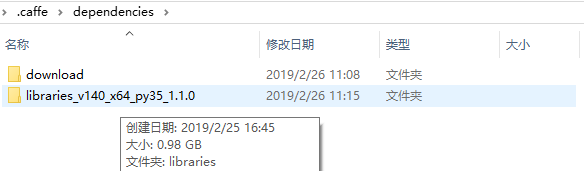
)

popd

@endlocal

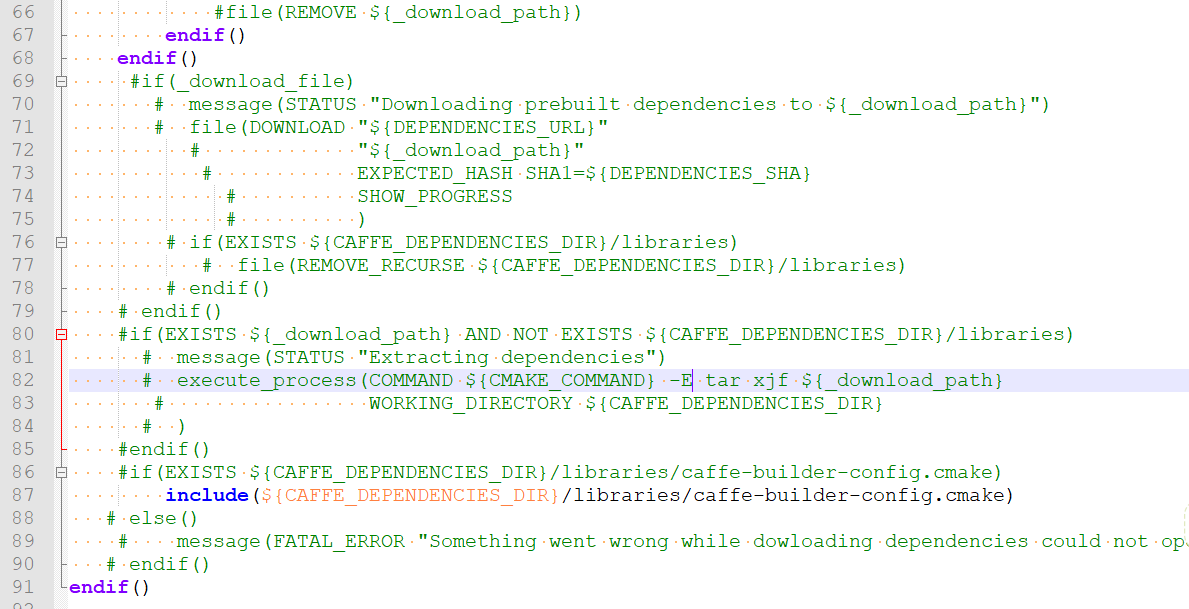
caffe\scripts文件下按shift加鼠标右键打开powershell窗口，将build\_win.cmd拖进窗口回车编译，会出现在下载“libraries\_v140\_x64\_py35\_1.1.0.tar.bz2”，源网站不存在，需要在百度云找资源下载或者在<https://github.com/willyd/caffe-builder/releases>下载

下载后把bz2解压到C:\Users\lhz\.caffe\dependencies，



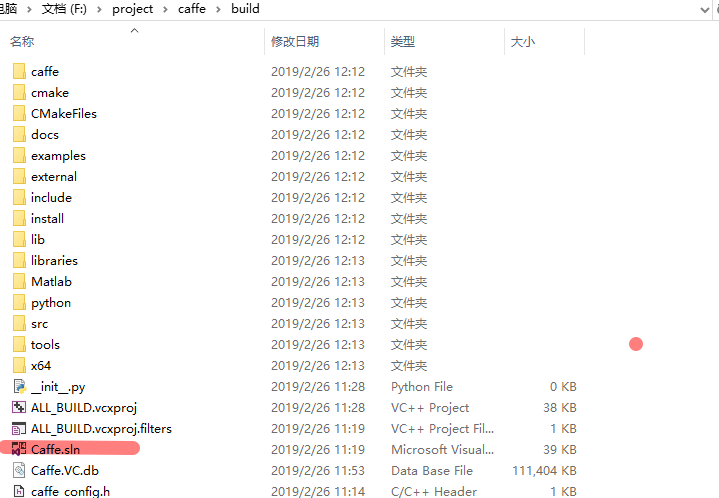
Download中是放置libraries\_v140\_x64\_py27\_1.1.0.tar.bz2的地方。

caffe\cmake中用Notepad++打开WindowsDownloadPrebuiltDependencies.cmake，删除下列代码以避免再次下载“libraries\_v140\_x64\_py27\_1.1.0.tar.bz2”依赖包：



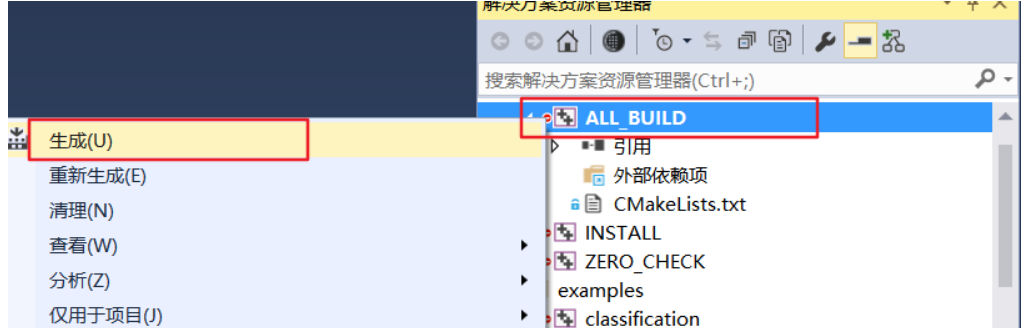
再次编译“build\_win.cmd”，过一段时间，未出错时，

在“…\caffe\build”文件夹下生成Caffe.sln文件



在VS2015中编译Caffe.sln文件

Release编译ALL\_BUILD成功后，注意都是x64，



完成了这一步，那么恭喜你，Caffe配置成功了

**配置Python接口**

打开Anaconda Prompt，如下输入来安装包文件，需要一些时间。

conda config --add channels conda-forge

conda config --add channels willyd

conda install --yes cmake ninja numpy scipy protobuf==3.1.0 six scikit-image pyyaml pydotplus graphviz

---------------------

conda install --yes cmake ninja numpy scipy protobuf==3.1.0 six scikit-image pyyaml pydotplus graphviz

conda install --yes cmake ninja numpy scipy protobuf==3.1.0 six scikit-image pyyaml pydotplus graphviz，管理员身份多执行几次

编译成功后，将“..\caffe\python”添加到环境变量，并把该路径下的caffe文件夹复制到安装Anaconda的site\_packages目录下，在我的电脑上是C:\......\Anaconda3\Lib\site-packages