**Win10使用VS2017安装Caffe详细总结**

1 前言

  本文将在Win10系统上安装和配置caffe。与网络上大多数文章不同，本文将在VS2017环境下编译CUDA9.0版本的caffe。事实上，本文配置流程对于更高版本的CUDA其安装流程是类似的，读者可进行参考。

2 符号约定

  本文中用%CAFFE\_DIR%代表准备安装caffe的目录路径，请读者自行替换为实际路径，如D:\Caffe\。

  本文中用%USER\_NAME%代表Windows用户名，因此Windows用户文件夹路径为C:\Users\%USER\_NAME%,请读者自行替换为实际路径。

3 系统环境

OS：Windows 10 x64

CPU：Intel Core i7-8700K 3.70GHz

GPU：Nvidia GEFORCE RTX 2080 Ti

Memory RAM：32GB

设置环境变量：

Boost\_INCLUDE\_DIR=D:\boost\boost

CUDA\_BIN\_PATH=C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\bin

CUDA\_LIB\_PATH=C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\lib\x64

CUDA\_PATH=C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0

CUDA\_PATH\_V10\_0=C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0

CUDA\_SDK\_BIN\_PATH=C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.0\bin\win64

CUDA\_SDK\_LIB\_PATH=C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.0\common\lib\x64

CUDA\_SDK\_PATH=C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.0

OpenCV\_DIR=D:\opencv3.4.1\build\x64\vc15\lib

4 预安装

Microsoft Visual Studio Community 2017 version 15.8

Anaconda3 version 4.3.30 with Python 3.6.5

Nvidia CUDA 10.0

cuDNN v7.5 for CUDA 10.0

Git v2.19.1

OpenCV 3.4.1

MatLab 2018a

5 caffe的获取

  在准备安装caffe的目录下运行以下命令

git clone https://github.com/BVLC/caffe.git

cd caffe

git checkout windows

6 caffe的编译

  本节是最为繁琐的一步。

6.1 VC++ 2017 版本 15.4 v14.11工具集

  新的VS2017 15.8自带的VC++工具集版本与CUDA 10.0有严重的兼容性问题，直接使用该版本的工具集编译CUDA 10.0会出现许多莫名其妙的错误，且在网上很难找到解决方法。但我们并不需要因此重新安装旧版本的VS2017，而只需安装并使用旧的VC++ 2017 版本 15.4 v14.11工具集编译CUDA 10.0即可。

6.1.1 安装VC++工具集

  打开VS Installer，修改安装，在安装选项里面的单个组件选择VC++ 2017 版本 15.4 v14.11 工具集安装。

6.1.2 配置VC++工具集

  在VS2017安装路径中查找vcvars64.bat所在目录，参考路径为C:\Program Files (x86)\Microsoft Visual Studio\2017\Enterprise\VC\Auxiliary\Build\vcvars64.bat。

6.2 配置build\_win.cmd

  编辑文件%CAFFE\_DIR%\caffe\scripts\build\_win.cmd，做如下修改后保存。

6.2.1 修改70~99行

  根据自己的需要修改70~99行的内容对编译进行配置。个人配置如下，供参考。

:: Change the settings here to match your setup

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

:: 【使用VS2017，MSVC\_VERSION=15】

if NOT DEFINED MSVC\_VERSION set MSVC\_VERSION=15

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

:: 【不使用Ninja】

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

:: Change to 1 to build caffe without CUDA support

:: 【使用GPU加速】

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)

:: 【在Anaconda配多版本python，这里用python3.5编译，PYTHON\_VERSION=3】

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

:: Change these options for your needs.

:: 【使用python接口】

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

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

:: 【使用matlab接口】

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=0

### 6.2.2 插入第112行

  在112行插入如下代码。

if "%MSVC\_VERSION%"=="15" (

set CMAKE\_GENERATOR=Visual Studio 15 2017 Win64

)

### 6.2.3 修改第158行

  将158行的batch\_file值修改在6.1.2节中修改的文件vcvars64\_VC1411.bat的绝对路径，如将原来的

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

修改为

set batch\_file=C:\Program Files (x86)\Microsoft Visual Studio\2017\Enterprise\VC\Auxiliary\Build\vcvars64.bat

## 6.3 修改WindowsDownloadPrebuiltDependencies.cmake

  编辑%CAFFE\_DIR%\caffe\cmake\WindowsDownloadPrebuiltDependencies.cmake文件，在第14行插入

set(DEPENDENCIES\_URL\_1916\_35 "${DEPENDENCIES\_URL\_BASE}/v${DEPENDENCIES\_VERSION}/${DEPENDENCIES\_NAME\_1900\_35}${DEPENDENCIES\_FILE\_EXT}")

set(DEPENDENCIES\_SHA\_1916\_35 "f060403fd1a7448d866d27c0e5b7dced39c0a607")

在这里，1916是MSVC编译器版本号，对应VC++ 2017 版本 15.4 v14.11工具集，35是python版本号。

## 6.4 修改Cuda.cmake

  编辑%CAFFE\_DIR%\caffe\cmake\Cuda.cmake文件，做如下修改并保存。

### 6.4.1 注释90~104行

  将90~104行注释，如下

# if(${CUDA\_ARCH\_NAME} STREQUAL "Fermi")

# set(\_\_cuda\_arch\_bin "20 21(20)")

# elseif(${CUDA\_ARCH\_NAME} STREQUAL "Kepler")

# set(\_\_cuda\_arch\_bin "30 35")

# elseif(${CUDA\_ARCH\_NAME} STREQUAL "Maxwell")

# set(\_\_cuda\_arch\_bin "50")

# elseif(${CUDA\_ARCH\_NAME} STREQUAL "Pascal")

# set(\_\_cuda\_arch\_bin "60 61")

# elseif(${CUDA\_ARCH\_NAME} STREQUAL "All")

# set(\_\_cuda\_arch\_bin ${Caffe\_known\_gpu\_archs})

# elseif(${CUDA\_ARCH\_NAME} STREQUAL "Auto")

# caffe\_detect\_installed\_gpus(\_\_cuda\_arch\_bin)

# else() # (${CUDA\_ARCH\_NAME} STREQUAL "Manual")

# set(\_\_cuda\_arch\_bin ${CUDA\_ARCH\_BIN})

# endif()

### 6.4.2 插入第104行

  在第104行插入

set(\_\_cuda\_arch\_bin "75")

在这里，75代表的是我的RTX2080Ti显卡的GPU计算能力。可以在https://developer.nvidia.com/cuda-gpus中查找Nvidia所有显卡的计算能力，修改为你的显卡对应计算能力数值即可。

6.5 Anaconda配置python环境

  由于项目需要，本文所安装的caffe在python 3.5环境下编译，但本机安装的Anaconda3默认安装的是python 3.7，因此需要利用创建一个python 3.5的虚拟环境。这里仅介绍用到的Anaconda的命令，其他用于管理虚拟环境的命令请读者自行搜索。

6.5.1 创建python 3.5虚拟环境

  命令行运行

conda create -n py35 python=3.5

激活成功后命令行前会显示(py35)字样。注意，下文提到的命令行命令均在这个虚拟环境中运行。

### 6.5.3 安装pycaffe所需包

  本文需要安装caffe的python接口，因此需要先安装pycaffe所需的包。命令行运行

conda install --yes numpy scipy protobuf six scikit-image pyyaml pydotplus graphviz

等待安装结束。

## 6.6 运行build\_win.cmd

  命令行运行文件%CAFFE\_DIR%\caffe\scripts\build\_win.cmd，则自动下载caffe依赖库，但会报openCV配置错误，错误打印片段如下

...

Call Stack (most recent call first):

cmake/Dependencies.cmake:114 (find\_package)

CMakeLists.txt:80 (include)

CMake Error at cmake/Dependencies.cmake:114 (find\_package):

Found package configuration file:

C:/Users/%USER\_NAME%/.caffe/dependencies/libraries/OpenCVConfig.cmake

but it set OpenCV\_FOUND to FALSE so package "OpenCV" is considered to be

NOT FOUND.

Call Stack (most recent call first):

CMakeLists.txt:80 (include)

-- Configuring incomplete, errors occurred!

See also "%CAFFE\_DIR%/caffe/build/CMakeFiles/CMakeOutput.log".

See also "%CAFFE\_DIR%/caffe/build/CMakeFiles/CMakeError.log".

ERROR: Configure failed

...

这是因为自动生成的opencv及其他库的配置文件与VS2017不合导致的，我们需要手动修改。

6.7 修改OpenCVConfig.cmake

  编辑C:\Users\%USER\_NAME%\.caffe\dependencies\libraries\OpenCVConfig.cmake，在86行插入保存

elseif(MSVC\_VERSION EQUAL 1916)

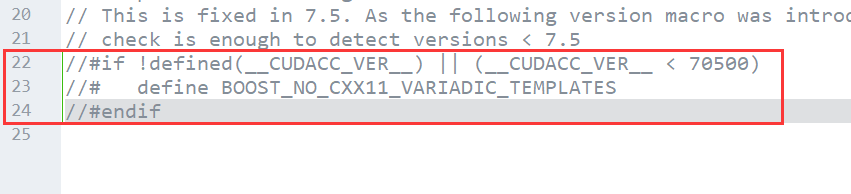
set(OpenCV\_RUNTIME vc14)

环境变量：(一定要添加这个环境变量)

OpenCV\_DIR = D:\opencv3.4.1\opencv\build\x64\vc15\lib

## 6.8 修改nvcc.hpp

  编辑C:\Users\%USER\_NAME%\.caffe\dependencies\libraries\include\boost-1\_61\boost\config\compiler\nvcc.hpp，注销掉这3行：



## 6.9 再次运行build\_win.cmd

  在命令行中运行%CAFFE\_DIR%\caffe\scripts\build\_win.cmd，顺利进行编译。编译过程中会输出大量的警告，其中包含很多乱码，类似下面

...

d:\nvida gpu computing toolkit\cuda\v9.0\include\crt\math\_functions.h(4417): warning C4819: 锟斤拷锟侥硷拷锟斤拷锟斤拷锟斤拷锟斤拷锟节碉拷前锟

斤拷锟斤拷页(936)锟叫憋拷示锟斤拷锟街凤拷锟斤拷锟诫将锟斤拷锟侥硷拷锟斤拷锟斤拷为 Unicode 锟斤拷式锟皆凤拷止锟斤拷锟捷讹拷失 [%CAFFE\_DIR%\caffe\scripts\build\src\caffe\caffe.vcx

proj]

d:\nvida gpu computing toolkit\cuda\v9.0\include\crt\math\_functions.h(5319): warning C4819: 锟斤拷锟侥硷拷锟斤拷锟斤拷锟斤拷锟斤拷锟节碉拷前锟

斤拷锟斤拷页(936)锟叫憋拷示锟斤拷锟街凤拷锟斤拷锟诫将锟斤拷锟侥硷拷锟斤拷锟斤拷为 Unicode 锟斤拷式锟皆凤拷止锟斤拷锟捷讹拷失 [%CAFFE\_DIR%\caffe\scripts\build\src\caffe\caffe.vcx

proj]

...

  这是由于编码字符问题造成的，和其他非乱码警告一样，并不影响编译过程，无需处理。大约经过10分钟，编译结束，如果错误数为0则编译成功。

编译出来的文件在D:\caffe\scripts\build\tools\Release\目录

7 caffe的python接口

  将%CAFFE\_DIR%\caffe\python路径下的caffe文件夹复制到Anaconda中虚拟环境py35的site-packages路径下，参考路径D:\Anaconda3\_64\envs\py35\Lib\site-packages。

  在py35虚拟环境下，运行命令进入python命令编辑模式。

Python

运行命令

import caffe

如果没有报错则说明成功安装。