

3D U-Net CNN with Keras(Demo)

本文主要介绍代码如何实现以及代码的内容，代码的理解。

首先是代码的安装环境配置：

1. 训练：

训练数据： 下载 BRATS 2017 [GBM](#) (HGG) 和 [LGG](#) 数据，放在解压的文件夹下 (brats/data/original)



2. 安装依赖环境： nibabel, keras, pytables, nilearn, SimpleITK, nipy

```
pip install -r requirements.txt
```

当然，也可以用 pip 一个个装。

3. 安装 ANTs N4BiasFieldCorrection (下载到路径 "/home/自己的用户名/" 目录下)，然后将 ANTs 库的位置放到环境变量的路径下。

a. 首先安装 CMAKE

```
wget https://cmake.org/files/v3.12/cmake-3.12.1-Linux-x86_64.sh
```

下载完安装包之后，在安装包的路径下调出终端，输入

```
sh cmake-3.12.1-Linux-x86_64.sh
```

解压完成之后还需要输入：

```
sudo apt-get install cmake-curses-gui
```

安装完成，可以使用命令 ccmake。

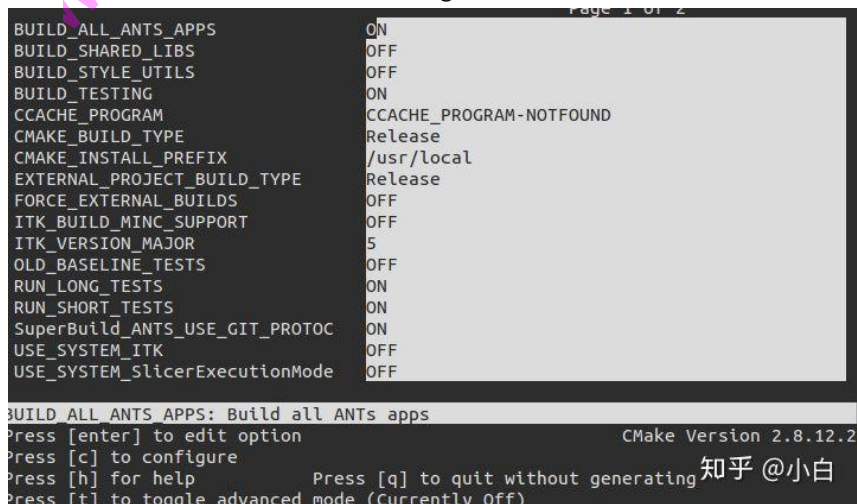
b. 新建一个文件夹 ANTs, 用来存放编译之后的文件 (xxx 是自己的用户名)

```
mkdir -p /home/xxx/ANTs
```

```
cd /home/xxx/ANTs
```

```
ccmake /home/xxx/ANTs-2.3.1 (这是之前下载好的安装包)
```

在跳出来的编译界面，按 c，Cmake 会做一些检查和给出选择界面，再次按 c，做另一轮的 configuration。如果没有报错的话，就可以按 g 来生成生成 make files。



完成之后，会返回终端命令行。输入如下命令来编译。

```
make
```

当然，这个命令编译的时间比较长。如果想要节省时间，可以使用下面的命令：

```
make -j 2
```

系统会编译 ITK 然后编译 ANTs，使用默认的设置，需要编译 40 分钟。想要加速的话，需要在 configuration 时，关掉 RUN_LONG_TESTS，或者关掉整个测试。

编译完成之后，类似于这样：

```
[100%] Linking CXX executable antsRegistrationTestDriver
[100%] Built target antsRegistrationTestDriver
Scanning dependencies of target antsApplyTransformsTestDriver
[100%] Building CXX object Examples/TestSuite/CMakeFiles/antsApplyTransformsTestDriver.dir/antsApplyTransformsTestDriver.cxx.o
[100%] Building CXX object Examples/TestSuite/CMakeFiles/antsApplyTransformsTestDriver.dir/antsApplyTransformsTestDriver.cxx.o
[100%] Linking CXX executable antsApplyTransformsTestDriver
[100%] Built target antsApplyTransformsTestDriver
[ 94%] No install step for 'ANTS'
[100%] Completed 'ANTS'
[100%] Built target ANTS
```

知乎 @小白

编译完成之后，如果想要使用 ANTs scripts 的话，需要将 source 文件下 Scripts/ 里面的所有内容拷贝到生成的 ANTs/bin/ 目录下。

c. 设置 PATH 和 ANTSPATH

如果是按照本文建立的话，二进制文件路径应该是/home/xxx/ANTs/bin

这个文件夹是包含了编译好的程序以及刚刚从源文件夹复制过来的 Scripts

对于 Linux 添加路径：

```
cd 到 /home/xxx/ANTs/bin 目录下
```

终端输入：

```
sudo gedit ~/.bash_profile
```

在跳出来的编辑器里面输入：

```
export ANTSPATH=${HOME}/ANTs/bin/
```

```
export PATH=${ANTSPATH}:${PATH}
```

保存之后退出，然后终端输入：

```
source ~/.bash_profile
```

这样环境路径就设置好了。检查一下是否正确，终端输入：

```
which antsRegistration
```

应该输出整个路径：

```
/home/hhtian/ANTs/bin/antsRegistration
```

这是我自己的存放路径。还可以在终端输入：

```
antsRegistrationSyN.sh
```

终端会输出这个 script 的使用介绍。如果一切没有问题，那么 ANTs 安装 OK 了。

4. ANTs 另一种简单安装

解决方案：安装 ANTs 软件，建议源码安装。这是一个编译完成的库，直接放到/usr/bin/下。

注意：下面第一个地址在 wins 上打开下载的是 win 版本，在 linux 上打开对应 linux 版本。

地址：<https://sourceforge.net/projects/advants/files/latest/download>

Github(linux): <https://github.com/MLearning/ANTs-1.9.x-Linux>

5. 将下载的原始数据转换成 nifti 格式的

首先 cd 到 brats 文件夹下，事先要将 3DUnetCNN 目录下的 unet3d 文件夹拷贝到 brats 文件夹中。

接下来在终端输入：

```
python
```

```
>>> from preprocess import convert_brats_data  
>>> convert_brats_data("data/original", "data/preprocessed")
```

只要前面环境配置正确，一般不会遇到错误。即使有也是 hd5f 版本冲突的问题，很好解决。之后就是数据处理，很漫长的一个等待的过程。

```
/tensorflowProject/3DUnetCNN/brats$ python  
Python 3.6.4 |Anaconda, Inc.| (default, Mar 13 2018, 01:15:57)  
[GCC 7.2.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> from preprocess import convert_brats_data  
Using TensorFlow backend.  
>>> convert_brats_data("data/original", "data/preprocessed")
```

知乎 @小白

贴上我自己的处理过程图。(披荆斩棘走到最后)。加油哦~

最后会在 preprocessed 文件夹中生成对应的处理之后的文件数据。

6. 训练

数据处理好之后，直接运行

```
python train.py (使用原始 Unet)
```

```
python train_isensee2017.py
```

如果 GPU 内存比较小，可以修改设置 config['patch_shape'] = (64, 64, 64) (亲测，单卡 NVIDIA Titan Xp GPUs with 12GB 是可以轻松运行的)

7. 测试

```
python predict.py
```

整个网络就可以跑通啦！完美！

8. 参考链接

1. <https://lmb.informatik.uni-freiburg.de/Publications/2016/CABR16/cicek16miccai.pdf> (论文)
2. <https://github.com/zishang33/3DUnetCNN> (代码 1)
3. <https://github.com/MLearning/3d-unet> (代码 2)
4. <https://github.com/ANTsX/ANTs/wiki/Compiling-ANTs-on-Linux-and-Mac-OS> (ANTs)
5. <https://blog.csdn.net/Liangzhaodong/article/details/78275027> (ANTs)
6. <https://sourceforge.net/projects/advants/> (ANTs 下载地址)
7. <https://zhuanlan.zhihu.com/p/43932439> (原文)