

深度学习 caffe 安装与测试

官网的安装说面：

http://caffe.berkeleyvision.org/install_ap.html

首先，安装依赖的一些包和库：

```
sudo apt-get install libprotobuf-dev libleveldb-dev libsnappy-dev libopencv-dev  
libhdf5-serial-dev protobuf-compiler  
  
sudo apt-get install --no-install-recommends libboost-all-dev
```

直接使用 `apt-get install` 安装即可，如果安装失败的话 `update` 一下继续装即可。

然后安装 CUDA7.5 和英伟达驱动，这里不做概述。

接下来安装 ATLAS 直接执行 `apt-get install libatlas-base-dev`

如果客户没有要求装 `opneBLAS` 或者 `MKL` 的情况下这里直接安装 `atlas` 即可。

执行：`apt-get install libatlas-base-dev` 等待安装完毕即可。

接下来配置 `python` 环境，这里我也做了更改不需要手动安装 `anaconda`，直接 `apt-get install python-dev` 等待安装完毕即可，不要修改任何环境变量，等待安装完毕即可。

然后需要安装 `CUDNN`，把压缩包拷到 `/home` 下，
执行

```
tar xvf cudnn*.tgz

cd cuda

sudo cp */*.h /usr/local/cuda/include/

sudo cp */libcudnn* /usr/local/cuda/lib64/

sudo chmod a+r /usr/local/cuda/lib64/libcudnn*
```

然后这里需要执行一步：

```
Sudo nano /etc/ld.so.conf
```

写入： /usr/local/cuda/lib64

保存退出 然后执行：sudo ldconfig

这里安装完毕之后，还需要安装一些依赖包，具体如下：

```
apt-get install libgflags-dev libgoogle-glog-dev liblmdb-dev
```

等待安装完毕以后继续往下。

安装 protobuf

1. 下载 protobuf 源代码（当前最新版本为：2.5.0）

```
#cd /opt
```

```
#wget https://protobuf.googlecode.com/files/protobuf-2.5.0.tar.gz
```

2. 解压，编译，安装

```
#tar zxvf protobuf-2.5.0.tar.gz
```

```
#cd protobuf-2.5.0
```

```
#./configure
```

```
#make
```

```
#make check
```

```
#make install
```

3. 继续安装 protobuf 的 python 模块(如果不用 python, 可跳过这一步)

```
#cd ./python
```

```
#python setup.py build
```

```
#python setup.py test
```

```
#python setup.py install
```

4. 安装完成, 验证 Linux 命令

```
#protoc --version
```

5. 验证 Python 模块是否被正确安装

```
#python
```

```
>>>import google.protobuf
```

如果没有报错, 说明安装正常。

-

为 Scipy 安装一些常用工具

-

```
sudo apt-get install -y libfreetype6-dev libpng12-dev pip install -U  
matplotlib ipython[all] jupyter pandas scikit-image  
sudo apt-get install libopenblas-dev python-dev python-pip python-nose python-numpy  
python-scipy
```

Caffe

```
sudo apt-get install libprotobuf-dev libleveldb-dev libsnpappy-dev  
libopencv-dev libhdf5-serial-dev protobuf-compiler
```

```
sudo apt-get install --no-install-recommends libboost-all-dev

sudo apt-get install libgflags-dev libgoogle-glog-dev liblmdb-dev

Sudo apt-get install python-numpy python-scipy python-matplotlib python-sklearn
python-skimage python-h5py python-protobuf python-leveldb python-networkx
python-nose python-pandas python-gflags Cython ipython
```

等待安装完毕后装 CAFFE,

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

1 Cd caffe

```
cp Makefile.config.example Makefile.config
```

然后编辑 Makefile.config 文件

2 只需要修改两个地方

```
# cuDNN acceleration switch (uncomment to build with cuDNN).

USE_CUDNN := 1
```

```
# CUDA directory contains bin/ and lib/ directories that we need.

CUDA_DIR := /usr/local/cuda

# On Ubuntu 14.04, if cuda tools are installed via
```

然后保存退出！

```
sudo pip install -r python/requirements.txt
```

这步比较慢，看网络情况。有没装成功的一定要重新装，保证全部装成功。

等待安装完毕以后编译

编译

```
1. make all -j4 #4 代表 cpu 核心，这个值可以改
2. make test -j4
3. make runtest -j4
4. make distribute -j4
5. make pycaffe -j4
```

上述如果没有任何报错，都操作正常的话，这里是完全可以 make 成功的。这 3 步执行完之后。我们来验证 caffe。

首先不要切换目录，就在我们 **caffe** 的目录里输入：

```
echo 'export CAFFE_ROOT=$(pwd)' >> ~/.bashrc

echo 'export PYTHONPATH=$CAFFE_ROOT/python:$PYTHONPATH' >>
~/.bashrc
source .bashrc
```

然后 cd 到 **caffe-master/python** 目录下执行 **python**

```
>>>
```

在>>> 这个后面输入 **import caffe** 导入成功则 OK。

导入成功：即没有任何报错

Caffe 上配置和运行 MNIST (caffe 测试)

Caffe 默认情况会安装在 CAFFEROOT，就是解压到的那个目录，例

如： **/home/username/caffe-master**，所以下面的工作，默认已经切换到了该工作目录。

下面的工作主要是，用于测试 Caffe 是否工作正常，不做详细评估。具体设置请参考官网：

<http://caffe.berkeleyvision.org/gathered/examples/mnist.html>

1. 数据预处理

可以用下载好的数据集，也可以重新下载，网速快可以直接下载，具体操作如下：

```
$ cd data/mnist
```

```
$ sudo sh ./get_mnist.sh
```

2. 重建 LDB 文件，就是处理二进制数据集为 Caffe 识别的数据集，以后所有的数据，包括 jpe 文件都要处理成这个格式

```
$ cd examples/mnist
```

```
$ sudo sh ./create_mnist.sh
```

生成 mnist-train-leveldb/ 和 mnist-test-leveldb/ 文件夹, 这里包含了 LDB 格式的数据集

PS: 这里可能会遇到一个报错信息:

```
Creating lmdb...
```

```
./create_mnist.sh: 16: ./create_mnist.sh:
```

```
build/examples/mnist/convert_mnist_data.bin: not found
```

解决方法是, 直接到 **caffe-master** 的根目录执行, 新版的 **caffe**, 基本上都得从根目录执行。

```
~/caffe-master$ sudo sh examples/mnist/create_mnist.sh
```

确保新生成的 2 个文件 mnist-train-lmdb 和 mnist-test-lmdb 位于 create_mnist.sh 同目录下

3. 训练 mnist

```
~/caffe-master$ sudo sh examples/mnist/train_lenet.sh
```

屏幕显示如下:

.....

```
I0329 16:43:03.739568 4155 solver.cpp:403] Iteration 9800, lr = 0.00599102
```

```
I0329 16:43:03.968662 4155 solver.cpp:191] Iteration 9900, loss = 0.00557684
```

```
I0329 16:43:03.968705 4155 solver.cpp:206] Train net output #0: loss =  
0.00557684 (* 1 = 0.00557684 loss)
```

```
I0329 16:43:03.968732 4155 solver.cpp:403] Iteration 9900, lr = 0.00596843
```

```
I0329 16:43:04.198199 4155 solver.cpp:317] Snapshotting to  
examples/mnist/lenet_iter_10000.caffemodel
```

```
I0329 16:43:04.202082 4155 solver.cpp:324] Snapshotting solver state to  
examples/mnist/lenet_iter_10000.solverstate
```

```
I0329 16:43:04.205214 4155 solver.cpp:228] Iteration 10000, loss = 0.00457862
```

```
I0329 16:43:04.205276 4155 solver.cpp:247] Iteration 10000, Testing net (#0)
```

```
I0329 16:43:04.307497 4155 solver.cpp:298] Test net output #0: accuracy =  
0.9906
```

```
I0329 16:43:04.307528 4155 solver.cpp:298] Test net output #1: loss = 0.02885  
(* 1 = 0.02885 loss)
```

```
I0329 16:43:04.307541 4155 solver.cpp:233] Optimization Done.
```

```
I0329 16:43:04.307549 4155 caffe.cpp:121] Optimization Done.
```

运行完结果如下:

生成四个新文件

```
lenet_iter_10000.caffemodel
```

```
lenet_iter_10000.solverstate
```

```
lenet_iter_5000.caffemodel
```

```
lenet_iter_5000.solverstate
```

备注：执行 `create_mnist.sh` 和 `train_lenet.sh` 前最好先确认这两个 .sh 文件有可执行的权限，否则在执行后可能会遇到如下图所示的报错(增加权限命令：`chmod`)

到这里全部 OK 即安装完毕。这个方法是官网安装方法，如果要求装 openBLAS 安装完 caffe 的 `makefile.config` 文件里 修改 `blas=open` 即可。如果要求装 openCV，先按照我的说明装完 caffe，再装 openCV. 不影响任何使用。也省去很多检查报错的时间。

如有深度学习服务器产品咨询请联系：

温巍

13511075577

北京思腾合力科技有限公司

电话： 010-51627561 转 807

QQ 号： 499686668

地址： 北京市海淀区北三环西路 48 号 1 号楼 A 座 15C

邮箱： wenwei@aiserver.cn