

denny的学习专栏

[博客园](#) [首页](#) [新随笔](#) [联系](#) [管理](#) [订阅](#) [XML](#)

随笔- 115 文章- 0 评论- 190

Caffe学习系列(13)：数据可视化环境（python接口）配置

caffe程序是由c++语言写的，本身是不带数据可视化功能的。只能借助其它的库或接口，如opencv, python或matlab。大部分人使用python接口来进行可视化，因为python出了个比较强大的东西：ipython notebook, 现在的最新版本改名叫jupyter notebook，它可将python代码搬到浏览器上去执行，以富文本方式显示，使得整个工作可以以笔记的形式展现、存储，对于交互编程、学习非常方便。

python环境不能单独配置，必须先编译好caffe，才能编译python环境。

python环境的配置说起来简单，做起来非常复杂。在安装的过程中，可能总是出现这样那样的问题。因此强烈建议大家用anaconda来进行安装，anaconda把很多与python有关的库都收集在一起了，包括numpy, scipy等等，因此，我们只需要下载对应系统，对应版本的anaconda来安装就可以了。

如果你想通过anaconda来安装，请跳过第一、二步，直接进入第三步开始：

一、安装python和pip

一般linux系统都自带python，所以不需要安装。如果没有的，安装起来也非常方便。安装完成后，可用version查看版本

```
# python --version
```

pip是专门用于安装python各种依赖库的，所以我们这里安装一下pip1.5.6

先用链接下载安装包 <https://pypi.python.org/packages/source/p/pip/pip-1.5.6.tar.gz>，然后解压，里面有一个setup.py的文件，执行这个文件就可以安装pip了

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

有些电脑可能会提示 no module name setuptools 的错误，这是没有安装setuptools的原因。那就需要先安装一下setuptools，到<https://pypi.python.org/packages/source/s/setuptools/setuptools-19.2.tar.gz> 下载安装包setuptools-19.2.tar.gz，然后解压执行

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

就要以安装setuptools了，然后再回头去重新安装pip。执行的代码都是一样的，只是在不同的目录下执行。

二、安装python接口依赖库

在caffe根目录的python文件夹下，有一个requirements.txt的清单文件，上面列出了需要的依赖库，按照这个清单安装就可以了。

在安装scipy库的时候，需要fortran编译器（gfortran），如果没有这个编译器就会报错，因此，我们可以先安装一下。

首先回到caffe的根目录，然后执行安装代码：

```
# cd ~/caffe
# sudo apt-get install gfortran
# for req in $(cat requirements.txt); do sudo pip install $req; done
```

安装完成以后，我们可以执行：

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

就会看到，安装成功的，都会显示Requirement already satisfied, 没有安装成功的，会继续安装。

在安装的时候，也许问题会有一大堆。这时候你就知道anaconda的好处了。

三、利用anaconda来配置python环境

如果你上面两步已经没有问题了，那么这一步可以省略。

如果你想简单一些，利用anaconda来配置python环境，那么直接从这一步开始，可以省略上面两步。

先到<https://www.continuum.io/downloads> 下载anaconda, 现在的版本有python2.7版本和python3.5版本，下载好对应版本、对应系统的anaconda，它实际上是一个sh脚本文件，大约280M左右。我下载的是linux版的python 2.7版

昵称：[denny402](#)园龄：[5年10个月](#)粉丝：[74](#)关注：[2](#)[+加关注](#)

< 2016年5月 >						
日	一	二	三	四	五	六
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

搜索

<input type="text"/>	<input type="button" value="找找看"/>
<input type="text"/>	<input type="button" value="谷歌搜索"/>

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)
[更多链接](#)

我的标签

[python\(26\)](#)
[caffe\(25\)](#)
[opencv3\(10\)](#)
[matlab\(9\)](#)
[mvc\(9\)](#)
[MVC3\(8\)](#)
[ajax\(7\)](#)
[geos\(6\)](#)
[opencv\(6\)](#)
[ml\(5\)](#)
[更多](#)

随笔分类

[caffe\(26\)](#)
[GDAL\(2\)](#)
[GEOS\(6\)](#)
[matlab\(11\)](#)
[opencv\(19\)](#)
[Python\(25\)](#)

随笔档案

[2016年1月 \(33\)](#)
[2015年12月 \(29\)](#)
[2015年11月 \(10\)](#)
[2015年7月 \(7\)](#)
[2014年10月 \(4\)](#)

Anaconda for Linux

PYTHON 2.7	PYTHON 3.5
Linux 64-bit 288M	Linux 64-bit 285M
Linux 32-bit 279M	Linux 32-bit 277M

下载成功后，在终端执行（2.7版本）：

```
# bash Anaconda2-2.4.1-Linux-x86_64.sh
```

或者3.5 版本：

```
# bash Anaconda3-2.4.1-Linux-x86_64.sh
```

在安装的过程中，会问你安装路径，直接回车默认就可以了。有个地方问你是否将anaconda安装路径加入到环境变量（.bashrc）中，这个一定要输入yes

安装成功后，会有当前用户根目录下生成一个anaconda2的文件夹，里面就是安装好的内容。

输入conda list 就可以查询，你现在安装了哪些库，常用的numpy, scipy名列其中。如果你还有什么包没有安装上，可以运行

conda install *** 来进行安装，

如果某个包版本不是最新的，运行 conda update *** 就可以了。

四、编译python接口

首先，将caffe根目录下的python文件夹加入到环境变量

打开配置文件bashrc

```
# sudo vi ~/.bashrc
```

在最后面加入

```
export PYTHONPATH=/home/xxx/caffe/python:$PYTHONPATH
```

注意 /home/xxx/caffe/python 是我的路径，这个地方每个人都不同，需要修改

保存退出，更新配置文件

```
# sudo ldconfig
```

然后修改编译配置文件Makefile.config. 我的配置是：

```
## Refer to http://caffe.berkeleyvision.org/installation.html
# Contributions simplifying and improving our build system are welcome!

# cuDNN acceleration switch (uncomment to build with cuDNN).
USE_CUDNN := 1

# CPU-only switch (uncomment to build without GPU support).
CPU_ONLY := 1

# uncomment to disable IO dependencies and corresponding data layers
# USE_OPENCV := 0
# USE_LEVELDB := 0
# USE_LMDB := 0

# uncomment to allow MDB_NOLOCK when reading LMDB files (only if necessary)
#   You should not set this flag if you will be reading LMDBs with any
#   possibility of simultaneous read and write
# ALLOW_LMDB_NOLOCK := 1

# Uncomment if you're using OpenCV 3
```

2014年7月 (4)

2013年10月 (3)

2013年8月 (5)

2013年7月 (7)

2013年6月 (6)

2011年4月 (4)

2010年6月 (3)

最新评论

1. Re:Caffe学习系列(23)：如何将别人训练好的model用到自己的数据上

您好，看到您的教程学到很多，我没用digs t，直接用的命令操作，但是由于电脑原因，我在做图片的lmdb的时候吧图片设置成125~125的，然后运行的时候就出现了错误错误内容是：Check failed.....

--weichang88688

2. Re:Caffe学习系列(4)：激活层 (Activiation Layers)及参数

给博主赞一个，对入门小白帮助真大！另外可以请问下你，为什么sigmoid层是另建一层，然后将自己输出，relu则本地操作不添加新的层，而后面的TanH,absolute value, power等都是.....

--MaiYatang

3. Re:Caffe学习系列(13)：数据可视化环境 (python接口)配置

在哪个路径下Make Clear 呢？Caffe的编译会被清除么？

--TonyFaith

4. Re:Caffe学习系列(13)：数据可视化环境 (python接口)配置

@TonyFaith清除以前的编译可以make clear，再重新编译就可以了。缺少python.h我不知道是什么原因...

--denny402

5. Re:Caffe学习系列(17)：模型各层数据和参数可视化

@weichang88688卷积层的输出数据就是net.blobs[conv1].data[0]，用一个变量保存起来就可以了c1=net.blobs[conv1].data[0]...

--denny402

阅读排行榜

1. SqlDataReader的关闭问题(9287)
2. 索引超出范围。必须为非负值并小于集合大小。(4655)
3. Caffe学习系列(1)：安装配置ubuntu14.04+cuda7.5+caffe+cuda(3167)
4. Caffe学习系列(12)：训练和测试自己的图片(2919)
5. Caffe学习系列(2)：数据层及参数(2492)

评论排行榜

1. Caffe学习系列(12)：训练和测试自己的图片(38)
2. SqlDataReader的关闭问题(22)
3. caffe windows 学习第一步：编译和安装 (vs2012+win 64)(15)
4. Caffe学习系列(23)：如何将别人训练好的model用到自己的数据上(15)
5. Caffe学习系列(3)：视觉层 (Vision Layers)及参数(11)

推荐排行榜

1. SqlDataReader的关闭问题(5)
2. Caffe学习系列(12)：训练和测试自己的图片(4)
3. Caffe学习系列(11)：图像数据转换成db (leveldb/lmdb)文件(2)


```
# OPENCV_VERSION := 3

# To customize your choice of compiler, uncomment and set the following.
# N.B. the default for Linux is g++ and the default for OSX is clang++
# CUSTOM_CXX := g++

# 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
# "sudo apt-get install nvidia-cuda-toolkit" then use this instead:
# CUDA_DIR := /usr

# CUDA architecture setting: going with all of them.
# For CUDA < 6.0, comment the *_50 lines for compatibility.
CUDA_ARCH := -gencode arch=compute_20,code=sm_20 \
    -gencode arch=compute_20,code=sm_21 \
    -gencode arch=compute_30,code=sm_30 \
    -gencode arch=compute_35,code=sm_35 \
    -gencode arch=compute_50,code=sm_50 \
    -gencode arch=compute_50,code=compute_50

# BLAS choice:
# atlas for ATLAS (default)
# mkl for MKL
# open for OpenBlas
BLAS := atlas
# Custom (MKL/ATLAS/OpenBLAS) include and lib directories.
# Leave commented to accept the defaults for your choice of BLAS
# (which should work)!
# BLAS_INCLUDE := /path/to/your/blas
# BLAS_LIB := /path/to/your/blas

# Homebrew puts openblas in a directory that is not on the standard search path
# BLAS_INCLUDE := $(shell brew --prefix openblas)/include
# BLAS_LIB := $(shell brew --prefix openblas)/lib

# This is required only if you will compile the matlab interface.
# MATLAB directory should contain the mex binary in /bin.
# MATLAB_DIR := /usr/local
# MATLAB_DIR := /Applications/MATLAB_R2012b.app

# NOTE: this is required only if you will compile the python interface.
# We need to be able to find Python.h and numpy/arrayobject.h.
# PYTHON_INCLUDE := /usr/include/python2.7 \
    /usr/lib/python2.7/dist-packages/numpy/core/include
# Anaconda Python distribution is quite popular. Include path:
# Verify anaconda location, sometimes it's in root.
ANACONDA_HOME := $(HOME)/anaconda2
PYTHON_INCLUDE := $(ANACONDA_HOME)/include \
    $(ANACONDA_HOME)/include/python2.7 \
    $(ANACONDA_HOME)/lib/python2.7/site-packages/numpy/core/include \

# We need to be able to find libpythonX.X.so or .dylib.
# PYTHON_LIB := /usr/lib
PYTHON_LIB := $(ANACONDA_HOME)/lib

# Homebrew installs numpy in a non standard path (keg only)
# PYTHON_INCLUDE += $(dir $(shell python -c 'import numpy.core;
print(numpy.core.__file__)')/include
# PYTHON_LIB += $(shell brew --prefix numpy)/lib

# Uncomment to support layers written in Python (will link against Python libs)
WITH_PYTHON_LAYER := 1

# Whatever else you find you need goes here.
INCLUDE_DIRS := $(PYTHON_INCLUDE) /usr/local/include
LIBRARY_DIRS := $(PYTHON_LIB) /usr/local/lib /usr/lib

# If Homebrew is installed at a non standard location (for example your home directory)
and you use it for general dependencies
# INCLUDE_DIRS += $(shell brew --prefix)/include
# LIBRARY_DIRS += $(shell brew --prefix)/lib

# Uncomment to use `pkg-config` to specify OpenCV library paths.
# (Usually not necessary -- OpenCV libraries are normally installed in one of the above
$LIBRARY_DIRS.)
```

```
# USE_PKG_CONFIG := 1

BUILD_DIR := build
DISTRIBUTE_DIR := distribute

# Uncomment for debugging. Does not work on OSX due to
# https://github.com/BVLC/caffe/issues/171
# DEBUG := 1

# The ID of the GPU that 'make runtest' will use to run unit tests.
TEST_GPUID := 0

# enable pretty build (comment to see full commands)
Q ?= @
```



修改完编译配置文件后，最后进行编译：

```
# sudo make pycaffe
```

编译成功后，不能重复编译，否则会提示 Nothing to be done for "pycaffe"的错误。

防止其它意外的错误，最好还编译一下：

```
# sudo make test -j8
# sudo make runtest -j8
```

也许你在编译runtest的时候，会报这样的错误：

.build_release/test/test_all.testbin: error while loading shared libraries: libhdf5.so.10: cannot open shared object file: No such file or directory

这是因为 libhdf5.so的版本问题，你可以进入/usr/lib/x86_64-linux-gnu看一下，你的libhdf5.so.x中的那个x是多少，比如我的是libhdf5.so.7

因此可以执行下面几行代码解决:

```
# cd /usr/lib/x86_64-linux-gnu
# sudo ln -s libhdf5.so.7 libhdf5.so.10
# sudo ln -s libhdf5_hl.so.7 libhdf5_hl.so.10
# sudo ldconfig
```

最终查看python接口是否编译成功：

进入python环境，进行import操作

```
# python
>>> import caffe
```

如果没有提示错误，则编译成功。

五、安装jupyter

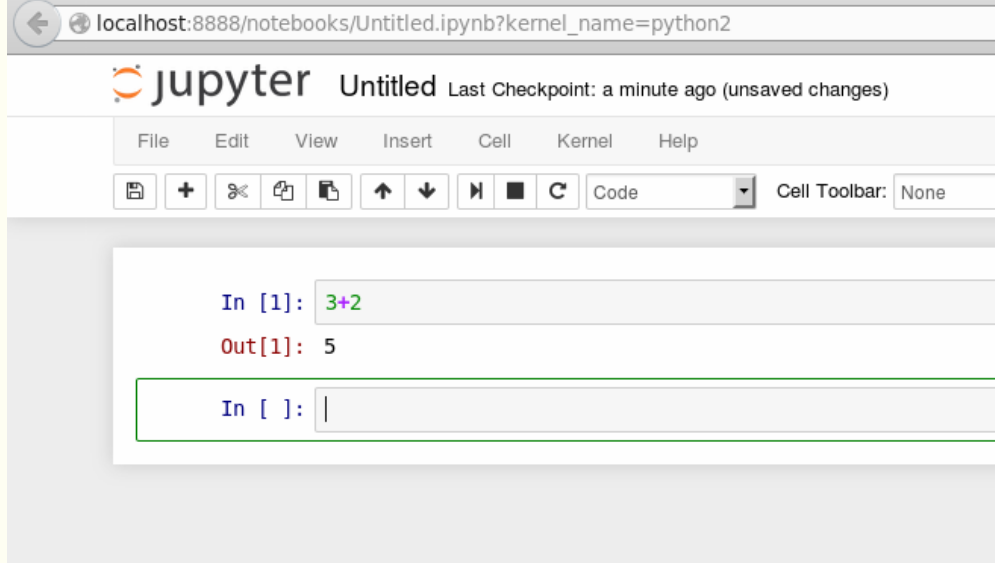
安装了python还不行，还得安装一下ipython，后者更加方便快捷，更有自动补全功能。而ipython notebook是ipython的最好展现方式。最新的版本改名为jupyter notebook,我们先来安装一下。（如果安装了anaconda, jupyter notebook就已经自动装好，不需要再安装）

```
# sudo pip install jupyter
```

安装成功后，运行notebook

```
# jupyter notebook
```


就会在浏览器中打开notebook，点击右上角的New-python2, 就可以新建一个网页一样的文件，扩展名为ipynb。在这个网页上，我们就可以像在命令行下面一样运行python代码了。输入代码后，按shift+enter运行，更多的快捷键，可点击上方的help-Keybaord shortcuts查看，或者先按esc退出编辑状态，再按h键查看。



分类: [caffe](#)

标签: [python](#), [caffe](#)



 [denny402](#)
关注 - 2
粉丝 - 74
[+加关注](#)

0 0
推荐 反对

(请您对文章做出评价)

« 上一篇: [Caffe学习系列\(12\) : 训练和测试自己的图片](#)

» 下一篇: [Caffe学习系列\(14\) : 初识数据可视化](#)

posted @ 2015-12-30 12:08 [denny402](#) 阅读(1806) 评论(10) [编辑](#) [收藏](#)

评论

#1楼 2016-02-01 14:23 | [lightbone](#) 




写得很好很详细。

我在编译runtest的时候，也遇到了这个问题，不过用你方法后还是不行。另外，import caffe 时报错：No module named google.protobuf.internal

我先尝试自己解决，如果解决不了，还是得麻烦你帮我看看哪里不对？


[支持\(0\)](#) [反对\(0\)](#)

#2楼 2016-02-01 14:40 | [lightbone](#) 



No module named google.protobuf.internal 的问题解决了，先用root权限，chmod 777 -R 解除anaconda2的文件夹保护，然后anaconda install protobuf，就没有报错了。

[支持\(0\)](#) [反对\(0\)](#)

#3楼 2016-02-01 15:29 | [lightbone](#) 



cd anaconda2

sudo cp /pkgs/hdf5-1.8.15.1-2/lib/libhdf5.so.10 /usr/lib/x86_64-linux-gnu

sudo cp /pkgs/hdf5-1.8.15.1-2/lib/libhdf5_hl.so.10 /usr/lib/x86_64-linux-gnu

完成。不知道为什么，我的库里面找不到libhdf5.so.x

本来x应该是8的。

[支持\(0\)](#) [反对\(0\)](#)

#4楼 2016-03-24 10:19 | [Zoe_only](#) 




楼主您好，请问mac下编译sudo ldconfig时提示

ldconfig: command not found

应该怎么解决呢？

支持(0) 反对(0)

#5楼 2016-03-24 11:16 | Zoe_only 

66

已经解决了，谢谢花时间看我的问题~

支持(0) 反对(0)

#6楼 2016-04-28 00:06 | wwwk9637 

66

@ lightbone

请问如果采用的是第一种方法（没有安装anaconda），怎么解决No module named google.protobuf.internal 的问题呢？

支持(0) 反对(0)

#7楼[楼主 

66

先安装一下google的protobuf库
sudo apt-get install libprotobuf-dev

支持(0) 反对(0)

#8楼 2016-05-07 10:28 | TonyFaith 

66

博主，你好。如果Makefile.config里面的配置配错了，怎么才能重新编译？我之前没有将anaconda2的路径加进去，导致很多功能不能用，现在重新编译提示缺少Python.h文件？

支持(0) 反对(0)

#9楼[楼主 

66

@ TonyFaith

清除以前的编译可以make clear，再重新编译就可以了。缺少python.h我不知道是什么原因

支持(0) 反对(0)

#10楼 2016-05-07 13:22 | TonyFaith 

66

在哪个路径下Make Clear 呢？Caffe的编译会被清除么？

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)



注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

最新IT新闻:

- [Mac笔记本电脑出货量大幅跳水 相比去年跌40%](#)
 - [Slack开放第三方服务使用登入授权机制“Sign in with Slack”](#)
 - [迪士尼财报低于预期，Disney Infinity 电玩产品线断头](#)
 - [高晓松：我现在是wannabe企业家 未来要做真的企业家](#)
 - [八成摄像头存安全隐患 家庭生活或被网上直播](#)
- » [更多新闻...](#)

最新知识库文章:

- [架构漫谈（九）：理清技术、业务和架构的关系](#)
 - [架构漫谈（八）：从架构的角度看如何写好代码](#)
 - [架构漫谈（七）：不要空设架构师这个职位，给他实权](#)
 - [架构漫谈（六）：软件架构到底是要解决什么问题？](#)
 - [架构漫谈（五）：什么是软件](#)
- » [更多知识库文章...](#)