

## Содержание

1. СБОРКА CAFFE ДЛЯ РАБОТЫ С PYTHON 3.....	1
1.1 Настройка конфигурации для Python 3.....	1
1.2 Сборка Caffe.....	2
1.3 Борьба с проблемами.....	3
2. ЗАПУСК DEEP DREAM С PYTHON 3.....	4
2.1 Блок импортов:.....	4
2.2 Блок Loading DNN model:.....	4
2.3 Блок Producing dreams.....	4

## 1. СБОРКА CAFFE ДЛЯ РАБОТЫ С PYTHON 3

**Перед сборкой:** boost и openCV нужно собирать с правильным Python! Полный список зависимостей и наиболее удобный способ их установки находится по [ССЫЛКЕ](#) на официальном сайте. Инструкция для Ubuntu подходит также и для Debian. Получить исходный код Caffe можно по [ссылке на github](#). Я брала прямо из ветки master, а не релиз-кандидатов.

*Указанная ниже инструкция работает на следующей конфигурации:*

**Процессор:** Intel® Core™ i3-2350M CPU @ 2.30GHz × 2

**Графика:** GeForce GT 520M/PCIe/SSE2

**ОС:** Debian GNU/Linux 8 (jessie) 64-бит

**CUDA:** 5.5, cuDNN 4.0.4

**Python:** anaconda3, Python 3.5.1, Numpy 1.11

**Прочее:** GNU CXX Compiler 4.6, Boost 1.55, Atlas (blas), openCV 2.4

### 1.1 Настройка конфигурации для Python 3

1. По умолчанию Caffe ищет Python 2. Поэтому в `CMakeLists.txt`, в блоке **Options** (примерно 33 строка) заменить `python_version "2"` на `python_version "3.5.1"`:

```
set(python_version "3.5.1" CACHE STRING "Specify which Python version to use")
```

**Внимание!** Изменять Python по умолчанию с помощью `update-alternatives` строго не рекомендуется. В Debian 8 по дефолту используется Python 2.7, поэтому некоторые пакеты зависят именно от этой версии и установленных в её каталоги модулей. Замена может привести к очень большим проблемам. В моём случае рухнули `apt-get`, `terminator` и некоторые другие утилиты. В процессе возвращения в «как было» заодно убился Gnome.

2. Чтобы убедиться, что при сборке нужный Python найдется, а все зависимости будут разрешены, выполнить:

```
mkdir build
cd build
cmake ..
```

Если `cmake` упорно видит не тот Python, который нужен, то нужно вычистить `build` и повторить шаг 2 заново. Если `cmake` падает с ошибками, нужно свериться со [ССЫЛКОЙ](#) на пререквизиты и доставить недостающие зависимости — `cmake` сообщит, чего не хватает.

**Замечание:** Сборку `cmake` не рекомендую, поскольку по неизвестным мне причинам возникали ошибки, которых не возникало при сборке с помощью `make`. Подобные комментарии встречала в Сети.

## 1.2 Сборка Caffe

1. В `caffe-master` положить `Makefile.config`. У меня он такой:

```
USE_CUDNN := 1
USE_OPENCV := 1
USE_LEVELDB := 1
USE_LMDB := 1
OPENCV_VERSION := 2
CUSTOM_CXX := /usr/bin/g++

CUDA_DIR := /usr/local/cuda-5.5-power8
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 \

BLAS := atlas

ANACONDA_HOME := $(HOME)/anaconda3
PYTHON_INCLUDE := $(ANACONDA_HOME)/include \
$(ANACONDA_HOME)/include/python3.5m \
$(ANACONDA_HOME)/lib/python3.5/site-packages/numpy/core/include
PYTHON_LIB := $(ANACONDA_HOME)/lib
PYTHON_LIBRARIES := boost_python3 python3.5m
WITH_PYTHON_LAYER := 1

INCLUDE_DIRS := $(PYTHON_INCLUDE) /usr/local/include
LIBRARY_DIRS := $(PYTHON_LIB) /usr/local/lib /usr/lib
BUILD_DIR := build
DISTRIBUTE_DIR := distribute

DEBUG := 1
TEST_GPUID := 0
Q ?= @
```

2. Если `cmake` завершается успешно, из каталога `build` запустить `make -j 4` (или 8 — сколько Ваша машина потянет параллельных процессов, чтобы ускорить сборку). Сборка занимает длительное время. Если все закончилось успешно, выполнить:

```
make all
make install
make runtest
```

3. Альтернатива 2, если с `cmake` возникли проблемы, можно попробовать другой способ: из `caffe-master` выполнить:

```
make all
make pycaffe
make test
make runtest
```

4. Чтобы `caffe` был доступен для Python, перейти в каталог `caffe-master/python` и оттуда выполнить:

```
export PYTHONPATH=`pwd`${PYTHONPATH:+:${PYTHONPATH}}
python -c "import caffe;print (caffe.__version__)"
```

## 1.3 Борьба с проблемами

...Но у меня в процессе сборки то ли дело возникали проблемы, поэтому привожу способы решения.

### 1. error: kernel launches from templates are not allowed in system files

Вот здесь: <http://kvzhao-blog.logdown.com/posts/234264-sip-a-caffe-0-how-do-i-install-caffe> , как и на официальных ресурсах Caffe, предлагают изменить дефолтный GCC. Мне это не помогло. Ситуацию прояснил тред на форуме openCV: <http://code.opencv.org/issues/2843> «This error is related with Thrust library and CUDA toolkit, not OpenCV» - таким образом, вариантов решения может быть два: поставить более новую CUDA и таким образом заодно заменить thrust, поскольку, начиная с версии 1.8 он поставляется вместе с CUDA, или скачать отдельно [thrust-1.8.1.zip](https://github.com/thrust/thrust/releases/tag/1.8.1) с <https://github.com/thrust/thrust/releases/tag/1.8.1> , распаковать архив, перейти в каталог thrust и с правами супер-пользователя скопировать заголовочные файлы в нужное место: `sudo cp -R * /usr/include/thrust/`

Затем собирать надо Caffe с помощью make (с cmake проблема по-прежнему возникает).

**2. То и дело не линковались библиотеки (blas, python, boost\_python3)** — проверить все пути в Makefile.config. В случае с boost убедиться, что в `/usr/lib/x86_64-linux-gnu` действительно лежит `boost_python3.so`:

```
ls /usr/lib/x86_64-linux-gnu | grep boost_python3
```

### 3. Тесты не собираются:

```
error while loading shared libraries: libhdf5_hl.so.10: cannot open shared
object file: No such file or directory
```

HDF5 установлено в `/usr/local/HDF_Group`, там в каталоге `lib` есть много библиотек, однако именно эта находится в `$(HOME)/anaconda3/lib`. Временное решение: создать файл `/etc/ld.so.conf.d/anaconda.conf` и в него прописать путь `$(HOME)/anaconda3/lib`, а затем с правами супер-пользователя выполнить `ldconfig` и перезапустить сборку тестов.

### 4. Тесты собираются, но после запуска падают:

```
cudaconv_layer.cpp:30] Check failed: status == CUDNN_STATUS_SUCCESS (6 vs. 0)
CUDNN_STATUS_ARCH_MISMATCH *** Check failure stack trace: **
```

Моя видеокарта не поддерживается cuDNN. Ознакомиться с поддержкой можно тут: <https://developer.nvidia.com/cuda-gpus> . Поэтому придется пересобирать caffe без cuDNN.

### 5. После выполнения `python -c "import caffe;print (caffe.__version__)"`, ошибка:

```
raise ValueError, "Can't create weekday with n == 0"
```

Старый `dateutil`. Перейти в `caffe-master/python`, исправить в `requirements.txt`: `python-dateutil>=2.0,<3`

Выполнить:

```
for req in $(cat requirements.txt); do sudo $  
(HOME)/anaconda3/bin/pip install $req; done
```

## 2. ЗАПУСК DEEP DREAM С PYTHON 3

Подготовка:

1. Открыть терминал, перейти в `caffe-master/python` и выполнить:

```
export PYTHONPATH=`pwd`${PYTHONPATH:+:${PYTHONPATH}}  
python -c "import caffe;print (caffe.__version__)"
```

Убедиться, что всё работает, и оттуда запустить: `cd && ipython notebook`

2. Выгрузить с гитхаб master-ветку: <https://github.com/google/deepdream> и открыть `dream.ipynb`

### 2.1 Блок импортов:

`from cStringIO import StringIO` заменить на `from io import StringIO`, так как модуль `cStringIO` не поддерживается в python 3.

В функции **showarray** заменить `f = StringIO()` на `f = BytesIO()`.

### 2.2 Блок Loading DNN model:

Нужно перейти в `caffe-master/scripts`, открыть скрипт `download_model_binary.py` и исправить там `import urllib` на `import urllib.request`, а все вызовы `urllib.urlretrieve` на `urllib.request.urlretrieve`. Далее загрузить **googlenet** по инструкции: [http://caffe.berkeleyvision.org/model\\_zoo.html](http://caffe.berkeleyvision.org/model_zoo.html) и убедиться, что в `models/bvlc_googlenet/` появился файл `bvlc_googlenet.caffemodel`.

### 2.3 Блок Producing dreams

Первый блок кода не нужно менять, а в функции **deepdream**:

- добавить скобки в `print (octave, i, end, vis.shape)`
- заменить `xrange` на `range`