

# 普兒的技术传送门

计算机视觉菜鸟一枚.

[首页](#) [新随笔](#) [联系](#) [订阅](#) [管理](#)

随笔 - 62 文章 - 0 评论 - 213

## Caffe提取任意层特征并进行可视化

现在Caffe的Matlab接口 (matcaffe3) 和python接口都非常强大, 可以直接提取任意层的feature map以及parameters, 所以本文仅仅作为参考, 更多最新的信息请参考:

<http://caffe.berkeleyvision.org/tutorial/interfaces.html>

原图

昵称：[普兒](#)  
园龄：[7年3个月](#)  
粉丝：[66](#)  
关注：[2](#)  
[+加关注](#)

搜索

找找看

谷歌搜索

我的标签

[ubuntu\(11\)](#)  
[Caffe\(4\)](#)  
[matlab\(2\)](#)  
[nvidia\(2\)](#)  
[error\(2\)](#)  
[oj\(2\)](#)  
[opencv\(2\)](#)  
[sicity\(2\)](#)  
[驱动\(2\)](#)



conv1层可视化结果（96个filter得到的结果）

算法(2)

更多

## 随笔分类

---

C/C++(3)

CF(1)

CSS(1)

Sicily(9)

TeX(2)

环境配置(18)

机器学习(3)

计算机视觉(2)

网站开发(2)

## 随笔档案

---

2016年9月 (1)

2016年7月 (1)

2016年4月 (1)

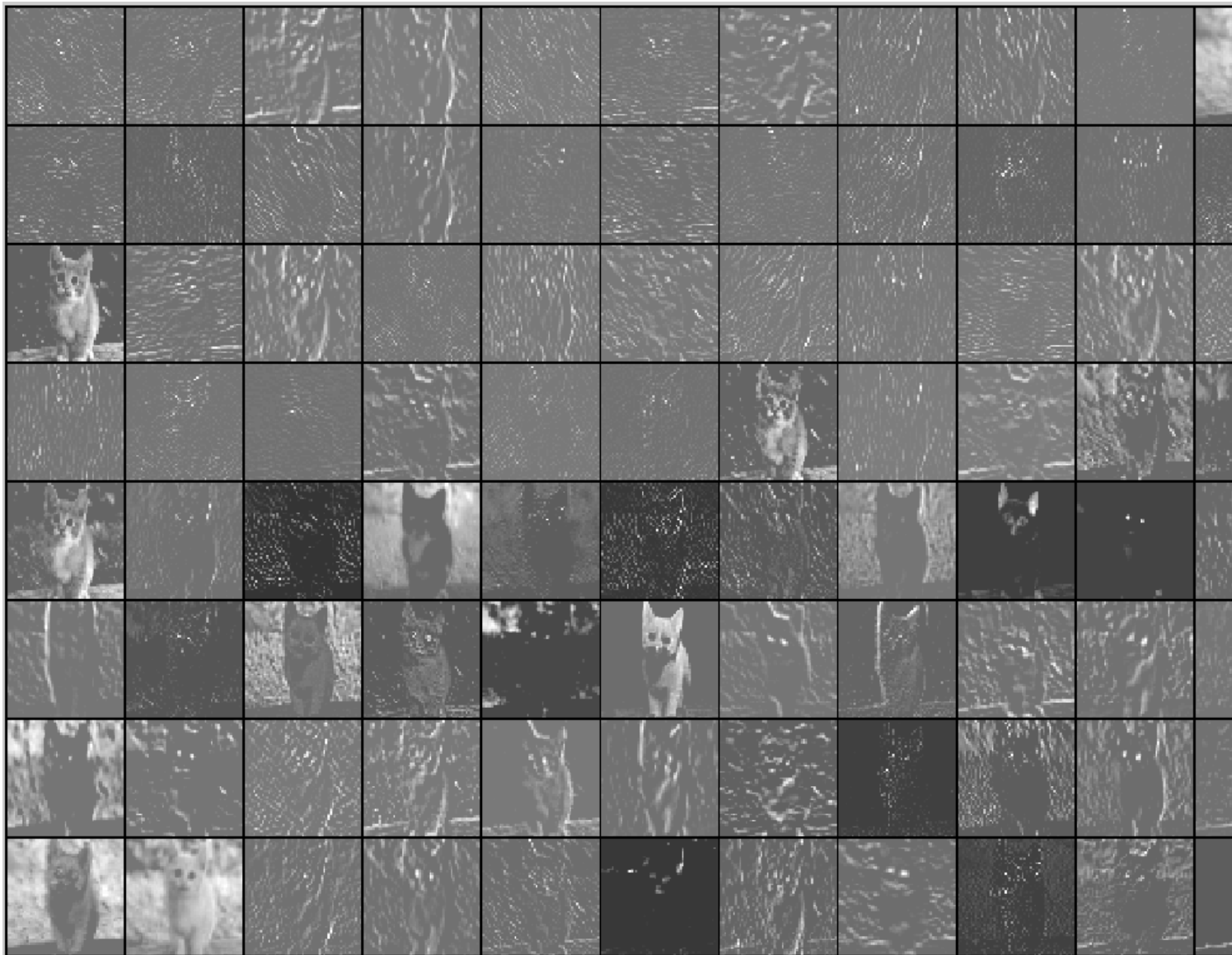
2016年3月 (1)

2015年8月 (1)

2015年5月 (1)

2015年4月 (1)

2015年3月 (2)



2015年1月 (2)  
2014年12月 (3)  
2014年11月 (6)  
2014年10月 (1)  
2014年9月 (2)  
2014年8月 (1)  
2014年4月 (1)  
2014年1月 (1)  
2013年12月 (3)  
2013年7月 (1)  
2013年5月 (1)  
2012年12月 (1)  
2012年9月 (2)  
2012年5月 (2)  
2011年8月 (1)  
2011年3月 (4)  
2010年12月 (2)  
2010年11月 (11)  
2010年10月 (1)  
2010年8月 (7)

练习

更多上机练习

数据模型与准备

安装好Caffe后，在examples/images文件夹下有两张示例图像，本文即在这两张图像上，用Caffe提供的预训练模型，进行特征提取，并进行可视化。

1. 进入caffe根目录，创建临时文件夹，用于存放所需要的临时文件

```
mkdir examples/_temp
```

2. 根据examples/images文件夹中的图片，创建包含图像列表的txt文件，并添加标签（0）

```
find `pwd`/examples/images -type f -exec echo {} \; > examples/_temp/temp.txt  
sed "s/$/ 0/" examples/_temp/temp.txt > examples/_temp/file_list.txt
```

3. 执行下列脚本，下载imagenet12图像均值文件，在后面的网络结构定义prototxt文件中，需要用到该文件（data/ilsrvrc212/imagenet\_mean.binaryproto）

```
data/ilsrvrc12/get_ilsrvrc_aux.sh
```

4. 将网络定义prototxt文件复制到\_temp文件夹下

```
cp examples/feature_extraction/imagenet_val.prototxt examples/_temp
```

## 提取特征

1. 创建 src/yourname/ 文件夹，存放我们自己的脚本

```
mkdir src/yourname
```

2. caffe的 extract\_features 将提取出的图像特征存为leveldb格式，为了方便观察特征，我们将利用下列两个python脚本将图像转化为matlab的.mat格式（请先安装caffe的python依赖库）

feat\_helper\_pb2.py



```
# Generated by the protocol buffer compiler.  DO NOT EDIT!
```

```
from google.protobuf import descriptor
from google.protobuf import message
from google.protobuf import reflection
from google.protobuf import descriptor_pb2
# @@protoc_insertion_point(imports)


DESCRIPTOR = descriptor.FileDescriptor(
    name='datum.proto',
    package='feat_extract',

    serialized_pb='\\n\\x0b\\x64\\x61tum.proto\\x12\\x0c\\x66\\x65\\x61t_extract\\\"i\\n\\x05\\x44\\x61tum\\x12\\x10\\n\\x08\\x63hannels\\x18\\x01 \\x01(\\x05\\x12\\x0e\\n\\x06height\\x18\\x02 \\x01(\\x05\\x12\\r\\n\\x05width\\x18\\x03 \\x01(\\x05\\x12\\x0c\\n\\x04\\x64\\x61ta\\x18\\x04 \\x01(\\x0c\\x12\\r\\n\\x05label\\x18\\x05 \\x01(\\x05\\x12\\x12\\n\\nfloat_data\\x18\\x06 \\x03(\\x02\\')

    _DATUM = descriptor.Descriptor(
        name='Datum',
        full_name='feat_extract.Datum',
        filename=None,
        file=DESCRIPTOR,
        containing_type=None,
        fields=[
            descriptor.FieldDescriptor(
                name='channels', full_name='feat_extract.Datum.channels', index=0,
                number=1, type=5, cpp_type=1, label=1,
                has_default_value=False, default_value=0,
                message_type=None, enum_type=None, containing_type=None,
                is_extension=False, extension_scope=None,
```

```
options=None),
descriptor.FieldDescriptor(
    name='height', full_name='feat_extract.Datum.height', index=1,
    number=2, type=5, cpp_type=1, label=1,
    has_default_value=False, default_value=0,
    message_type=None, enum_type=None, containing_type=None,
    is_extension=False, extension_scope=None,
    options=None),
descriptor.FieldDescriptor(
    name='width', full_name='feat_extract.Datum.width', index=2,
    number=3, type=5, cpp_type=1, label=1,
    has_default_value=False, default_value=0,
    message_type=None, enum_type=None, containing_type=None,
    is_extension=False, extension_scope=None,
    options=None),
descriptor.FieldDescriptor(
    name='data', full_name='feat_extract.Datum.data', index=3,
    number=4, type=12, cpp_type=9, label=1,
    has_default_value=False, default_value="",
    message_type=None, enum_type=None, containing_type=None,
    is_extension=False, extension_scope=None,
    options=None),
descriptor.FieldDescriptor(
    name='label', full_name='feat_extract.Datum.label', index=4,
    number=5, type=5, cpp_type=1, label=1,
    has_default_value=False, default_value=0,
    message_type=None, enum_type=None, containing_type=None,
    is_extension=False, extension_scope=None,
    options=None),
descriptor.FieldDescriptor(
    name='float_data', full_name='feat_extract.Datum.float_data', index=5,
    number=6, type=2, cpp_type=6, label=3,
    has_default_value=False, default_value=[],
```

```
        message_type=None, enum_type=None, containing_type=None,
        is_extension=False, extension_scope=None,
        options=None),
    ],
    extensions=[
    ],
    nested_types=[],
    enum_types=[
    ],
    options=None,
    is_extendable=False,
    extension_ranges=[],
    serialized_start=29,
    serialized_end=134,
)
```

```
DESCRIPTOR.message_types_by_name['Datum'] = _DATUM
```

```
class Datum(message.Message):
    __metaclass__ = reflection.GeneratedProtocolMessageType
    DESCRIPTOR = _DATUM

    # @@protoc_insertion_point(class_scope:feat_extract.Datum)

    # @@protoc_insertion_point(module_scope)
```



leveldb2mat.py



```
import leveldb
import feat_helper_pb2
import numpy as np
import scipy.io as sio
import time

def main(argv):
    leveldb_name = sys.argv[1]
    print "%s" % sys.argv[1]
    batch_num = int(sys.argv[2]);
    batch_size = int(sys.argv[3]);
    window_num = batch_num*batch_size;

    start = time.time()
    if 'db' not in locals().keys():
        db = leveldb.LevelDB(leveldb_name)
        datum = feat_helper_pb2.Datum()

    ft = np.zeros((window_num, int(sys.argv[4])))
    for im_idx in range(window_num):
        datum.ParseFromString(db.Get('%d' % (im_idx)))
        ft[im_idx, :] = datum.float_data

    print 'time 1: %f' % (time.time() - start)
    sio.savemat(sys.argv[5], {'feats':ft})
    print 'time 2: %f' % (time.time() - start)
    print 'done!'

    #leveldb.DestroyDB(leveldb_name)

if __name__ == '__main__':
    import sys
    main(sys.argv)
```





3. 创建脚本文件extract\_feature.sh，并执行，将在examples/\_temp文件夹下得到leveldb文件（features\_conv1）和.mat文件（features.mat）



```
#!/usr/bin/env sh
# args for EXTRACT_FEATURE
TOOL=../../build/tools
MODEL=../../examples/imagenet/caffe_reference_imagenet_model #下载得到的caffe model
PROTOTXT=../../examples/_temp/imagenet_val.prototxt # 网络定义
LAYER=conv1 # 提取层的名字，如提取fc7等
LEVELDB=../../examples/_temp/features_conv1 # 保存的leveldb路径
BATCHSIZE=10

# args for LEVELDB to MAT
DIM=290400 # 需要手工计算feature长度
OUT=../../examples/_temp/features.mat #.mat文件保存路径
BATCHNUM=1 # 有多少哥batch，本例只有两张图，所以只有一个batch

$TOOL/extract_features.bin $MODEL $PROTOTXT $LAYER $LEVELDB $BATCHSIZE
python leveldb2mat.py $LEVELDB $BATCHNUM $BATCHSIZE $DIM $OUT
```



4. 得到.mat文件后，需要对其进行可视化，这里用了UFLDL里的display\_network函数，由于可视化出来结果进行了翻转，因此对原代码的67, 69, 83, 85行进行了修改

display\_network.m 存放在 src/yourname文件夹下



```
function [h, array] = display_network(A, opt_normalize, opt_graycolor, cols, opt_colmajor)
% This function visualizes filters in matrix A. Each column of A is a
% filter. We will reshape each column into a square image and visualizes
% on each cell of the visualization panel.
```

```
% All other parameters are optional, usually you do not need to worry
% about it.
% opt_normalize: whether we need to normalize the filter so that all of
% them can have similar contrast. Default value is true.
% opt_graycolor: whether we use gray as the heat map. Default is true.
% cols: how many columns are there in the display. Default value is the
% squareroot of the number of columns in A.
% opt_colmajor: you can switch convention to row major for A. In that
% case, each row of A is a filter. Default value is false.
warning off all

if ~exist('opt_normalize', 'var') || isempty(opt_normalize)
    opt_normalize= true;
end

if ~exist('opt_graycolor', 'var') || isempty(opt_graycolor)
    opt_graycolor= true;
end

if ~exist('opt_colmajor', 'var') || isempty(opt_colmajor)
    opt_colmajor = false;
end

% rescale
A = A - mean(A(:));

if opt_graycolor, colormap(gray); end

% compute rows, cols
[L M]=size(A);
sz=sqrt(L);
buf=1;
if ~exist('cols', 'var')
```

```
if floor(sqrt(M))^2 ~= M
    n=ceil(sqrt(M));
    while mod(M, n)~=0 && n<1.2*sqrt(M), n=n+1; end
    m=ceil(M/n);
else
    n=sqrt(M);
    m=n;
end
else
    n = cols;
    m = ceil(M/n);
end

array=-ones(buf+m*(sz+buf), buf+n*(sz+buf));

if ~opt_graycolor
    array = 0.1.* array;
end

if ~opt_colmajor
    k=1;
    for i=1:m
        for j=1:n
            if k>M,
                continue;
            end
            clim=max(abs(A(:,k)));
            if opt_normalize
                array(buf+(i-1)*(sz+buf)+(1:sz), buf+(j-1)*(sz+buf)+(1:sz))=reshape(A(:,k), sz, sz)'/clim;
            else
                array(buf+(i-1)*(sz+buf)+(1:sz), buf+(j-1)*(sz+buf)+(1:sz))=reshape(A(:,k), sz, sz)'/max(abs(A(:)));
            end
        end
        k=k+1;
    end
end
```

```
        end
        k=k+1;
    end
end
else
    k=1;
    for j=1:n
        for i=1:m
            if k>M,
                continue;
            end
            clim=max(abs(A(:,k)));
            if opt_normalize
                array(buf+(i-1)*(sz+buf)+(1:sz), buf+(j-1)*(sz+buf)+(1:sz))=reshape(A(:,k), sz, sz)'/clim;
            else
                array(buf+(i-1)*(sz+buf)+(1:sz), buf+(j-1)*(sz+buf)+(1:sz))=reshape(A(:,k), sz, sz)';
            end
            k=k+1;
        end
    end
end

if opt_graycolor
    h=imagesc(array, 'EraseMode', 'none', [-1 1]);
else
    h=imagesc(array, 'EraseMode', 'none', [-1 1]);
end
axis image off


drawnow;

warning on all
```



5. 调用display\_network 以及提取到的feature进行可视化：

在 examples/\_temp/ 下创建如下matlab脚本， 并执行




```
addpath(genpath(' ../../src/wyang'));

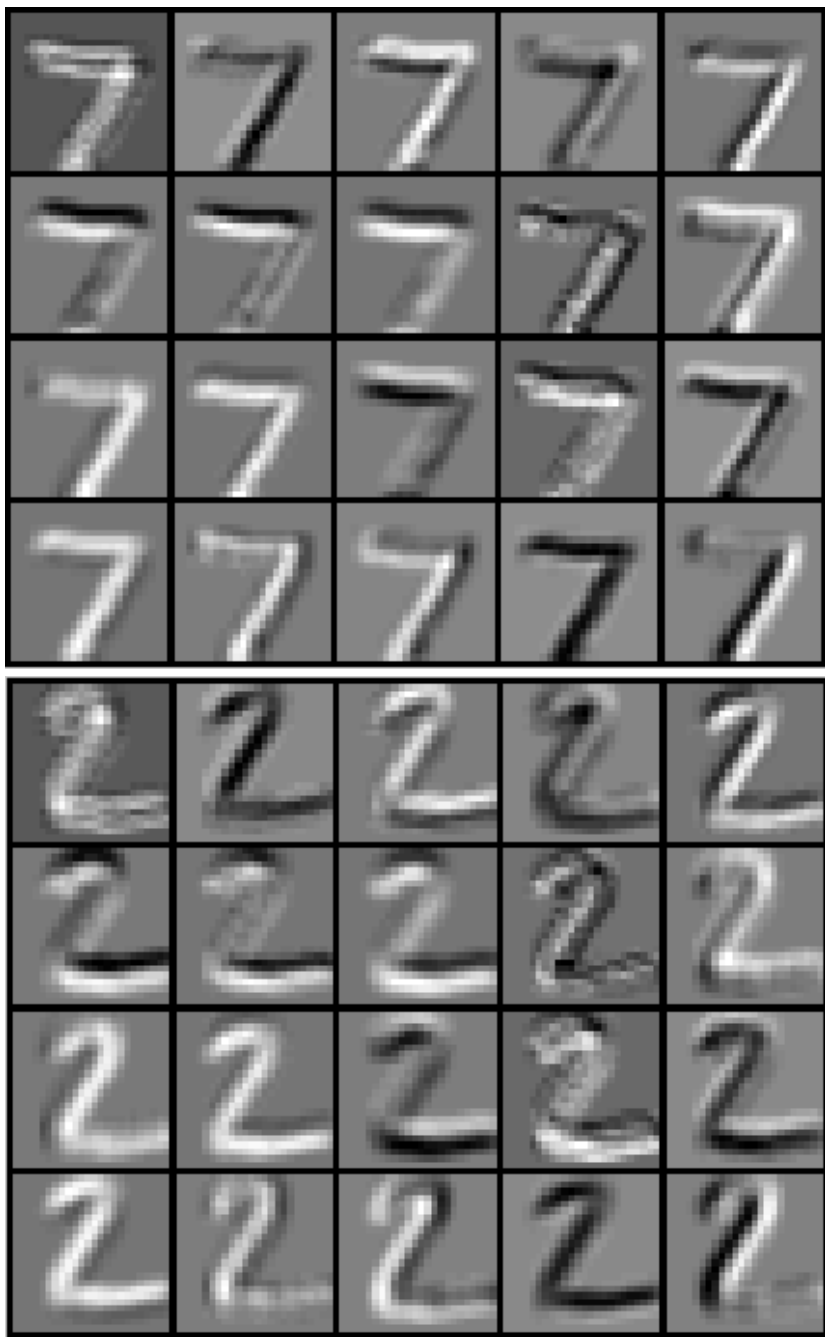
nsample      = 3;
num_output   = 96;

load features.mat
width = size(feats, 2);
nmap   = width / num_output;

for i = 1:nsample
    feat = feats(i, :);
    feat = reshape(feat, [nmap num_output]);
    figure('name', sprintf('image #%d', i));
    display_network(feat);
end
```



下图是在MNIST上用lenet进行conv1层卷积后得到的结果



标签: [Caffe](#), [deep learning](#), [可视化](#)

好文要顶

关注我

收藏该文



普兒

关注 - 2

粉丝 - 66

+加关注

« 上一篇 : [Batch Gradient Descent vs. Stochastic Gradient Descent](#)» 下一篇 : [Caffe + Ubuntu 14.04 64bit + CUDA 6.5 配置说明](#)

2

0

posted @ 2014-09-11 20:54 普兒 阅读(9893) 评论(53) 编辑 收藏

&lt; Prev

1

2

## 评论列表

楼主 #51楼 2016-05-19 17:27 普兒

@ 先疯队

不保证这个帖子现在还能work毕竟是两年前的帖子.

请使用matlab或者python接口.

matlab接口获取参数的例子:

`net.params('conv1', 1).get_data()`

获取feature map的例子

`net.blobs('conv1').get_data()`

支持(0) 反对(0)

---

#52楼 2016-05-20 13:43 [先疯队](#)

@ TJUsxh

引用

@玩的代码

解决了，我的是因为提取特征的时候错了，是extract\_Feature.sh 里的BATCHSIZE=10 乘以 imagenet\_val.prototxt 网络里的batch的个数。

不然提取特征的时候，没算好会越界的。

得到的是个数是什么？我跟你的错误一样

支持(0) 反对(0)

---

#53楼 2016-05-20 14:16 [先疯队](#)

@ \_无声的雨

引用

\$TOOL/extract\_features.bin \$MODEL \$PROTOTXT \$LAYER \$LEVELDB \$BATCHSIZE博主，您好!现在做这个命令提取特征的时候，出现以下反馈信息

E0429 17:21:50.458314 17794 extract\_features.cpp:37] This program takes in a trained network and an input data layer, and then extract features of the input data produced by the net.

Usage: ext...

这个问题怎么解决的？

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](#)。

[【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库](#)

[【推荐】报表开发有捷径：快速设计轻松集成，数据可视化和交互](#)

Copyright ©2017 普兒