

From Beijing with Love

[联系](#)[订阅](#)[管理](#)

在Caffe中实现模型融合

模型融合

有的时候我们手头可能有了若干个已经训练好的模型，这些模型可能是同样的结构，也可能是不同的结构，训练模型的数据可能是同一批，也可能不同。无论是出于要通过ensemble提升性能的目的，还是要设计特殊作用的网络，在用Caffe做工程时，融合都是一个常见的步骤。

比如考虑下面的场景，我们有两个模型，都是基于resnet-101，分别在两拨数据上训练出来的。我们希望把这两个模型的倒数第二层拿出来，接一个fc层然后训练这个fc层进行融合。那么有两个问题需要解决：1) 两个模型中的层的名字都是相同的，但是不同模型对应的权重不同；2) 如何同时在一个融合好的模型中把两个训练好的权重都读取进来。

Caffe中并没有直接用于融合的官方工具，本文介绍一个简单有效的土办法，用融合模型进行ensemble的例子，一步步实现模型融合。

完整例子

模型定义和脚本：

https://github.com/frombeijingwithlove/dlcv_for_beginners/tree/master/random_bonus/multiple_models_fusion_caffe

预训练模型：

随笔分类(44)

[Python\(17\)](#)[不严谨的数学\(3\)](#)[机器学习\(8\)](#)[计算摄像学\(2\)](#)[计算视觉\(9\)](#)[未分类\(5\)](#)

最新评论

1. Re:在Caffe中实现模型融合

@zhitongxiang引用楼主您好，关于fuse_model.py第19行我有个疑问，我觉得应该是fusion.net =，您这里写的是net，难道不应该是融合后的网络的参数等于嘛感谢.....

--達聞西

2. Re:在Caffe中实现模型融合

楼主您好，关于fuse_model.py第19行我有个疑问，我觉得应该是fusion.net =，您

https://github.com/frombeijingwithlove/dlcv_book_pretrained_caffe_models/blob/master/mnist_lenet_odd_iter_30000.caffemodel

https://github.com/frombeijingwithlove/dlcv_book_pretrained_caffe_models/blob/master/mnist_lenet_even_iter_30000.caffemodel

虽然模型只是简单的LeNet-5，但是方法是可以拓展到其他大模型上的。

模型(及数据)准备：直接采用预训练好的模型

本文的例子要融合的是两个不同任务的模型：

对偶数0, 2, 4, 6, 8分类的模型

对奇数1, 3, 5, 7, 9分类的模型

采用的网络都是LeNet-5

直接从上节中提到的本文例子的repo下载预定义的模型和权重。

上一部分第一个链接中已经写好了用来训练的LeNet-5结构和solver，用的是ImageData层，以训练奇数分类的模型为例：

```
name: "LeNet"
layer {
  name: "mnist"
  type: "ImageData"
  top: "data"
  top: "label"
  include {
    phase: TRAIN
  }
  transform_param {
    mean_value: 128
    scale: 0.00390625
  }
  image_data_param {
    source: "train_odd.txt"
    is_color: false
    batch_size: 25
  }
}
layer {
```

这里写的是net，难道不应该是融合后的网络的参数等于嘛

--zhitongxiang

3. Re:漫谈计算摄像学 (一)：直观理解光场 (Light Field)

@zx2017引用@達聞西博主好牛！！学些啦~ 看了好多遍，还有第二篇文章 可是无法用代码重现博主的结果啊。。。不知道从何下手，好着急，能不能给指引个方向啊，或者有没有时间找找源代码供大家参考学习.....

--達聞西

4. Re:漫谈计算摄像学 (一)：直观理解光场 (Light Field)

@達聞西博主好牛！！学些啦~ 看了好多遍，还有第二篇文章 可是无法用代码重现博主的结果啊。。。不知道从何下手，好着急，能不能给指引个方向啊，或者有没有时间找找源代码供大家参考学习下！...

--zx2017

5. Re:在Caffe中实现模型融合

@達聞西你好，博主在生成融合模型的时候为什么选择的是caffe.TEST模式呀。我是融合两个ResNet50网络，我是把train和val分成两个文件的，所以我选择的是caffe.TRAIN模式。

--浪溢泽沉

```
name: "mnist"
type: "ImageData"
top: "data"
top: "label"
include {
  phase: TEST
}
transform_param {
  mean_value: 128
  scale: 0.00390625
}
image_data_param {
  source: "val_odd.txt"
  is_color: false
  batch_size: 20
}
}
layer {
  name: "conv1"
  type: "Convolution"
  bottom: "data"
  top: "conv1"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  convolution_param {
    num_output: 20
    kernel_size: 5
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
```

```
}  
layer {  
  name: "pool1"  
  type: "Pooling"  
  bottom: "conv1"  
  top: "pool1"  
  pooling_param {  
    pool: MAX  
    kernel_size: 2  
    stride: 2  
  }  
}  
layer {  
  name: "conv2"  
  type: "Convolution"  
  bottom: "pool1"  
  top: "conv2"  
  param {  
    lr_mult: 1  
  }  
  param {  
    lr_mult: 2  
  }  
  convolution_param {  
    num_output: 50  
    kernel_size: 5  
    stride: 1  
    weight_filler {  
      type: "xavier"  
    }  
    bias_filler {  
      type: "constant"  
    }  
  }  
}  
layer {  
  name: "pool2"  
  type: "Pooling"  
  bottom: "conv2"
```

```
    top: "pool2"
    pooling_param {
      pool: MAX
      kernel_size: 2
      stride: 2
    }
  }
  layer {
    name: "ip1"
    type: "InnerProduct"
    bottom: "pool2"
    top: "ip1"
    param {
      lr_mult: 1
    }
    param {
      lr_mult: 2
    }
    inner_product_param {
      num_output: 500
      weight_filler {
        type: "xavier"
      }
      bias_filler {
        type: "constant"
      }
    }
  }
}
layer {
  name: "relu1"
  type: "ReLU"
  bottom: "ip1"
  top: "ip1"
}
layer {
  name: "ip2"
  type: "InnerProduct"
  bottom: "ip1"
  top: "ip2"
```

```
param {
  lr_mult: 1
}
param {
  lr_mult: 2
}
inner_product_param {
  num_output: 5
  weight_filler {
    type: "xavier"
  }
  bias_filler {
    type: "constant"
  }
}
}
layer {
  name: "accuracy"
  type: "Accuracy"
  bottom: "ip2"
  bottom: "label"
  top: "accuracy"
  include {
    phase: TEST
  }
}
layer {
  name: "loss"
  type: "SoftmaxWithLoss"
  bottom: "ip2"
  bottom: "label"
  top: "loss"
}
```

训练偶数分类的prototxt的唯一区别就是ImageData层中数据的来源不一样。

模型(及数据)准备：Start From Scratch

当然也可以自行训练这两个模型，毕竟只是个用于演示的小例子，很简单。方法如下：

第一步 下载MNIST数据

直接运行download_mnist.sh这个脚本

第二步 转换MNIST数据为图片

运行convert_mnist.py，可以从mnist.pkl.gz中提取所有图片为jpg

```
import os
import pickle, gzip
from matplotlib import pyplot

# Load the dataset
print('Loading data from mnist.pkl.gz ...')
with gzip.open('mnist.pkl.gz', 'rb') as f:
    train_set, valid_set, test_set = pickle.load(f)

imgs_dir = 'mnist'
os.system('mkdir -p {}'.format(imgs_dir))
datasets = {'train': train_set, 'val': valid_set, 'test': test_set}
for dataname, dataset in datasets.items():
    print('Converting {} dataset ...'.format(dataname))
    data_dir = os.sep.join([imgs_dir, dataname])
    os.system('mkdir -p {}'.format(data_dir))
    for i, (img, label) in enumerate(zip(*dataset)):
        filename = '{:0>6d}_{}.jpg'.format(i, label)
        filepath = os.sep.join([data_dir, filename])
        img = img.reshape((28, 28))
        pyplot.imshow(filepath, img, cmap='gray')
        if (i+1) % 10000 == 0:
            print('{} images converted!'.format(i+1))
```

第三步 生成奇数、偶数和全部数据的列表

运行gen_img_list.py，可以分别生成奇数、偶数和全部数据的训练及验证列表：

```
import os
import sys

mnist_path = 'mnist'
```

```
data_sets = ['train', 'val']

for data_set in data_sets:
    odd_list = '{}_odd.txt'.format(data_set)
    even_list = '{}_even.txt'.format(data_set)
    all_list = '{}_all.txt'.format(data_set)
    root = os.sep.join([mnist_path, data_set])
    filenames = os.listdir(root)
    with open(odd_list, 'w') as f_odd, open(even_list, 'w') as f_even, open(all_list, 'w') as f_all:
        for filename in filenames:
            filepath = os.sep.join([root, filename])
            label = int(filename[:filename.rfind('.')].split('_')[1])
            line = '{} {} \n'.format(filepath, label)
            f_all.write(line)

            line = '{} {} \n'.format(filepath, int(label/2))
            if label % 2:
                f_odd.write(line)
            else:
                f_even.write(line)
```

第四步 训练两个不同的模型

就直接训练就行了。Solver的例子如下：

```
net: "lenet_odd_train_val.prototxt"
test_iter: 253
test_initialization: false
test_interval: 1000
base_lr: 0.01
momentum: 0.9
weight_decay: 0.0005
lr_policy: "step"
gamma: 0.707
stepsize: 1000
display: 200
max_iter: 30000
snapshot: 30000
```



```
snapshot_prefix: "mnist_lenet_odd"
solver_mode: GPU
```

注意到test_iter是个奇怪的253，这是因为MNIST的验证集中奇数样本多一些，一共是5060个，训练随便取个30个epoch，应该是够了。

制作融合后模型的网络定义

前面提到了模型融合的难题之一在于层的名字可能是相同的，解决这个问题非常简单，只要把名字改成不同就可以，加个前缀就行。按照这个思路，我们给奇数分类和偶数分类的模型的每层前分别加上odd/和even/作为前缀，同时我们给每层的学习率置为0，这样融合的时候就可以只训练融合的全连接层就可以了。实现就是用Python自带的正则表达式匹配，然后进行字符串替换，代码就是第一部分第一个[链接](#)中的rename_n_freeze_layers.py：

```
import sys
import re

layer_name_regex = re.compile('name:\s*"(.*)"')
lr_mult_regex = re.compile('lr_mult:\s*\d+\.\s*\d*')

input_filepath = sys.argv[1]
output_filepath = sys.argv[2]
prefix = sys.argv[3]

with open(input_filepath, 'r') as fr, open(output_filepath, 'w') as fw:
    prototxt = fr.read()
    layer_names = set(layer_name_regex.findall(prototxt))
    for layer_name in layer_names:
        prototxt = prototxt.replace(layer_name, '{}/{}'.format(prefix, layer_name))

    lr_mult_statements = set(lr_mult_regex.findall(prototxt))
    for lr_mult_statement in lr_mult_statements:
        prototxt = prototxt.replace(lr_mult_statement, 'lr_mult: 0')

    fw.write(prototxt)
```

这个方法虽然土，不过有效，另外需要注意的是如果确定不需要动最后一层以外的参数，或者原始的训练prototxt中就没有lr_mult的话，可以考虑用Caffe的propagate_down这个参数。把这个脚本分别对奇数和偶数模型执行，并记住自己设定的前缀even和odd，然后把数据层到ip1层的定义复制并粘贴到一个文件中，然后把ImageData层和融合层的定义也写入到这个文件，注意融合前需要先用Concat层把特征拼接一下：

```
name: "LeNet"
layer {
  name: "mnist"
  type: "ImageData"
  top: "data"
  top: "label"
  include {
    phase: TRAIN
  }
  transform_param {
    mean_value: 128
    scale: 0.00390625
  }
  image_data_param {
    source: "train_all.txt"
    is_color: false
    batch_size: 50
  }
}
layer {
  name: "mnist"
  type: "ImageData"
  top: "data"
  top: "label"
  include {
    phase: TEST
  }
  transform_param {
    mean_value: 128
    scale: 0.00390625
  }
  image_data_param {
    source: "val_all.txt"
```

```
    is_color: false
    batch_size: 20
  }
}
...
### rename_n_freeze_layers.py 生成的网络结构部分 ###
...
layer {
  name: "concat"
  bottom: "odd/ip1"
  bottom: "even/ip1"
  top: "ip1_fused"
  type: "Concat"
  concat_param {
    axis: 1
  }
}
layer {
  name: "ip2"
  type: "InnerProduct"
  bottom: "ip1_fused"
  top: "ip2"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  inner_product_param {
    num_output: 10
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
```

```
name: "accuracy"
type: "Accuracy"
bottom: "ip2"
bottom: "label"
top: "accuracy"
include {
  phase: TEST
}
}
layer {
  name: "loss"
  type: "SoftmaxWithLoss"
  bottom: "ip2"
  bottom: "label"
  top: "loss"
}
```

分别读取每个模型的权重并生成融合模型的权重

这个思路就是用pycaffe进行读取，然后按照层名字的对对应关系进行值拷贝，最后再存一下就可以，代码如下：

```
import sys
sys.path.append('/path/to/caffe/python')
import caffe

fusion_net = caffe.Net('lenet_fusion_train_val.prototxt', caffe.TEST)

model_list = [
    ('even', 'lenet_even_train_val.prototxt', 'mnist_lenet_even_iter_30000.caffemodel'),
    ('odd', 'lenet_odd_train_val.prototxt', 'mnist_lenet_odd_iter_30000.caffemodel')
]

for prefix, model_def, model_weight in model_list:
    net = caffe.Net(model_def, model_weight, caffe.TEST)

    for layer_name, param in net.params.iteritems():
        n_params = len(param)
        try:
            for i in range(n_params):
```

```
        net.params['{}/{}'.format(prefix, layer_name)][i].data[...] =  
param[i].data[...]  
        except Exception as e:  
            print(e)  
  
fusion_net.save('init_fusion.caffemodel')
```

训练融合后的模型

这个也没什么好说的了，直接训练即可，本文例子的参考Solver如下：

```
net: "lenet_fusion_train_val.prototxt"  
test_iter: 500  
test_initialization: false  
test_interval: 1000  
base_lr: 0.01  
momentum: 0.9  
weight_decay: 0.0005  
lr_policy: "step"  
gamma: 0.707  
stepsize: 1000  
display: 200  
max_iter: 30000  
snapshot: 30000  
snapshot_prefix: "mnist_lenet_fused"  
solver_mode: GPU
```

分类: [Python](#) , [机器学习](#) , [计算视觉](#)

[好文要顶](#)[关注我](#)[收藏该文](#)[達聞西](#)[关注 - 0](#)[粉丝 - 66](#)[+加关注](#)

0

0

« 上一篇：[用Caffe生成对抗样本](#)

» 下一篇：[用GAN生成二维样本的小例子](#)

posted @ 2017-04-09 12:38 達聞西 阅读(2562) 评论(19) 编辑 收藏

评论列表

#1楼 2017-04-11 20:39 [菖蒲花海里的少年谣](#)

博主您好，想问您一个问题：分别训练奇数和偶数的网络分别以奇数和偶数数据集作为输入吗？那么融合之后的训练集和测试集分别是什么呢？是既包括奇数又包括偶数的吗？这样的话如何确定输入图像是通过奇数网络还是偶数网络呢？

支持(0) 反对(0)

#2楼[楼主] 2017-04-12 13:04 [達聞西](#)

@ 菖蒲花海里的少年谣

融合后的训练集是所有数，图片输入之后两个网络都走一遍，然后融合结果，我分开奇数偶数只是举个例子而已。。

支持(0) 反对(0)

#3楼 2017-04-12 15:36 [菖蒲花海里的少年谣](#)

@ 達聞西

谢谢，这里明白了。但是我运行fuse_model.py文件时报错：“Check failed: mdb_status == 0 (2 vs. 0) No such file or directory

*** Check failure stack trace: ***”

想问您一下这个文件我需要修改什么吗？

支持(0) 反对(0)

#4楼[楼主] 2017-04-12 20:59 [達聞西](#)

@ 菖蒲花海里的少年谣

感觉像是数据没有准备好

支持(0) 反对(0)

#5楼 2017-04-13 09:59 菖蒲花海里的少年谣

@

可是我的数据直接训练时没有问题的，一做融合就出现了这个问题。

支持(0) 反对(0)

#6楼[楼主] 2017-04-13 12:43 達聞西

@ 菖蒲花海里的少年谣

有没有可能是融合的prototxt和原来的prototxt路径不一样呢？这个错看上去像是路径错了

支持(0) 反对(0)

#7楼 2017-04-17 11:22 roger18

你好，楼主，请问融合两个model过程中用到的fusion_net = caffe.Net('lenet_fusion_train_val.prototxt', caffe.TEST)中的prototxt有什么要求吗？

我现在有两个网络，想一个当网络前半部分，一个当网络后半部分，对应的model也有，早上在是你给出的程序，提示错误'module' object has no attribute 'TEST'，fuse_net那行，因此我想问问对那个文件有没有什么要求

支持(0) 反对(0)

#8楼[楼主] 2017-04-17 12:43 達聞西

@ roger18

这个错误很奇怪。。是不是你module里有其他的东西也起名叫caffe了？

支持(0) 反对(0)

#9楼 2017-04-17 13:46 roger18

@

没有啊，我就是在你程序基础改了这些文件名，import没改

支持(0) 反对(0)

#10楼[楼主] 2017-04-17 17:48 達聞西

@ roger18

那这个问题很奇怪，是否你在用某个非master branch的caffe？因为caffe.TEST是一个基本的使用时的标签

支持(0) 反对(0)

#11楼 2017-04-18 09:35 roger18

@ 達聞西

好像我换了个caffe的环境就没有这个错，但还是有问题。Unknown bottom blob 'data' (layer 'conv1_1', bottom index 0)

我现在有两个训练好的model，目前想训练的网络结构由这两个训练好的model构成，如果想调用两个model同时finetune有什么解决方法吗？

支持(0) 反对(0)

#12楼[楼主] 2017-04-18 14:15 達聞西

@ roger18

你的数据叫什么名字得定义好和接受数据的layer一致，我这里默认用data作为数据名字了

支持(0) 反对(0)

#13楼 2017-07-03 10:40 iresine

博主您好，请问dlcv_for_beginners这本书全名和作者是谁？搜索不到这本书，谢谢～

支持(0) 反对(0)

#14楼[楼主] 2017-07-17 09:47 達聞西

@ iresine

引用

博主您好，请问dlcv_for_beginners这本书全名和作者是谁？搜索不到这本书，谢谢～

还在审校中.....

支持(0) 反对(0)

#15楼 2017-07-29 20:42 **浪溢泽沉**

你好，问一下，你这个在训练两个网络的时候，每层的名字没有修改，但是融合的时候每层的名字都改了，这样其实还是相当于重新训练，并没有得到之前两个网络训练的参数.....博主求解释

支持(0) 反对(0)

#16楼[楼主] 2017-07-30 10:28 **達聞西**

@ 浪溢泽沉

引用

你好，问一下，你这个在训练两个网络的时候，每层的名字没有修改，但是融合的时候每层的名字都改了，这样其实还是相当于重新训练，并没有得到之前两个网络训练的参数.....博主求解释

是这样的，最后读取的时候不是直接用caffe的Net初始化，而是分别读取每个模型的权重，然后直接赋值给对应的参数，代码是

https://github.com/frombeijingwithlove/dlcv_for_beginners/blob/master/random_bonus/multiple_models_fusion_caffe/fuse_model.py
中的第19行

支持(0) 反对(0)

#17楼 2017-07-30 19:49 **浪溢泽沉**

@ 達聞西

你好，博主在生成融合模型的时候为什么选择的是caffe.TEST模式呀。我是融合两个ResNet50网络，我是把train和val分成两个文件的，所以我选择的是caffe.TRAIN模式。我希望只训练融合后最后一个全连接层，所以在这个layer加了propagate_down: 0.但是最后还是出现out of memory???

支持(0) 反对(0)

#18楼 2017-08-30 16:33 zhitongxiang

楼主您好，关于fuse_model.py第19行我有个疑问，我觉得应该是fusion.net =，您这里写的是net，难道不应该是融合后的网络的参数等于嘛

支持(0) 反对(0)

#19楼[楼主] 2017-08-30 17:50 達聞西

@ zhitongxiang

引用

楼主您好，关于fuse_model.py第19行我有个疑问，我觉得应该是fusion.net =，您这里写的是net，难道不应该是融合后的网络的参数等于嘛

感谢感谢~已修正~

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】腾讯云上实验室 1小时搭建人工智能应用

【推荐】可嵌入您系统的“在线Excel”！SpreadJS 纯前端表格控件

【推荐】阿里云“全民云计算”优惠升级



美团云
MEITUAN OPEN SERVICES

GPU
云主机 5折起

M60 原价1660 / 月 现价830 / 月 P40 原价3400 / 月 现价2100 / 月

[查看详情](#)

最新IT新闻:

- Google将Fuchsia系统的内核重命名为Zircon
 - 微博推出“霸王授权协议”背后，是对优质创作者的抢夺
 - 酷炫！耐克推出能与iPhone互动的NBA球衣
 - 人类用了5400年，才做出了《猩球崛起3》里的大猩猩
 - 凡客的荣辱兴衰，是一场雷军也救不回来的“中年危机”
- » 更多新闻...



JIGUANG | 极光

极光统计 多维洞察用户
增长运营指标

最新知识库文章:

- Google 及其云智慧
 - 做到这一点，你也可以成为优秀的程序员
 - 写给立志做码农的大学生
 - 架构腐化之谜
 - 学会思考，而不只是编程
- » 更多知识库文章...

