lhanchao的博客

[深度学习] Very Deep Convolutional Networks for Large-Scale Image Recognitic （VGGNet）阅读笔记

标签： 深度学习 VGG

2017年03月20日 16:03:35 3278人阅读 评论(0) 收藏

分类： 深度学习（14）

先发发牢骚，最近的日子就是“准备数据集——想改进方法——跑实验——实验结果不好”的循环，熬得情都没有了= =

好了，废话不多说了，这篇VGGNet的论文是两三周之前看的了，而且最近搭的网络结构跟VGGNet很相像出来复习一下吧。

这篇论文是牛津大学的几个人做出来，在ILSVRC 2014中的classification项目的比赛中取得了第2名的成错，第一就是上一篇博客中介绍的GooLeNet），但是实际上这里提出的VGGNet单个网络的识别率是比GooLeNet要好的，下面是正式的介绍。

Abstract

这篇论文中作者探究了一下深度神经网络中网络的层数与网络的分类能力的关系，结果就是网络越深，网络学习能好，分类能力越强。现在看来可能是显而易见的，可是在14年还没几个人搭特别深的网络时，也算个发现成果吧。的主要贡献就是使用了具有小卷积核（3x3大小）的深度神经网络在达到一定深度（16~19层）以后性能会有非常提升。

1. ConvNet Configurations

1.1 Architecture

在网络的训练过程中，作者采用的输入图像为224x224的RGB彩色图像，作者做的唯一的预处理就是计算这三个通均值，在训练时减去这些平均值，这样处理训练时网络可能更快的收敛。输入图像在网络中经过一系列具有3x3大积核的卷积层（说3x3是小卷积核的原因是，3x3大小是可以同时获取上下左右像素信息的最小的卷积核）。在实验作者也采用了1x1大小的卷积核，1x1大小的卷积核不能获取局部区域的信息，只能作为对图像单个像素的一种线性理。

在网络中，所有卷积层的大小都是3x3，而且stride均为1，同时pad也为1，保证卷积层不改变feature map的大小ax pooling层的kernel为2x2，同时stride为2。卷积层以后是两个全连接层，全连接层为4096维，后面是一个100全连接层，最后跟一个softmax层。另外，所有隐层后面都跟一个ReLU激活曾进行非线性处理。

说了这么多看一下作者总结的网络结构表格吧，表格说的更清楚。

Table 1: **ConvNet configurations** (shown in columns). The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as "conv⟨receptive field size⟩-⟨number of channels⟩". The ReLU activation function is not shown for brevity.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

1.2 Configurations

全部都写到上面的表中了，这里需要提一下的是作者的**卷积层的filters数目**，从64开始，每经过一个max pooling
filters数目翻倍。

1.3 Discussion

这一部分作者主要介绍了使用3x3大小的卷积核的原因：我么可以很明显的看出，两个连续的3x3大小的卷积层与一
大小的卷积核具有相同的局部空间，而连续3个3x3大小的卷积核则与一个7x7大小的卷积核具有相同的局部空间。
比使用一个7x7大小的卷积核，3个连续的3x3的卷积核进行了3次非线性处理（7x7的进行了1次），这样在一定程
提高了网络的学习能力；另外，使用3x3的卷积核也降低了参数的数目，假设3x3的的卷积核处理C通道的feature
时，一共有3（3x3xCxC）=27CxC个参数，而7x7的则有7x7xCxC共49CxC个参数；最后这也可以看成是对7x7的
行正则化处理，强迫7x7的卷积核分解为3x3大小的。
后面作者介绍了1x1卷积核的好处，这里我们就不详述了，在GooLeNet的介绍博客中已经很详细的说明了1x1大小
积核的作用。
最后，作者介绍了几个使用小卷积核的工作，证明小卷积核相比大卷积核确实是有一定的优势的。

2. Classification Framework

2.1 训练阶段

这里主要介绍了作者在训练时的一些参数配置：

（1）使用随机梯度下降法（SGD）进行训练；

（2）batch size = 256；

（3）momentum = 0.9；

（4）weight decay = 0.0005；

（5）dropout ratio = 0.5

（6）learning rate = 0.01

（7）learning rate变化方法：当validation的识别率不变化的时候，learning rate降低十倍**（很好奇，怎么用ca
这一点的？）**

以上的设置都与我们平时的设置差不大多，就不详细解释了。

最后作者说明在训练的过程中，比AlexNet收敛的要快一些，原因为：（1）使用小卷积核和更深的网络进行的正则
（2）在特定的层使用了预训练得到的数据进行参数的初始化。

作者提到对于较浅的网络，如上图中的网络A，可以直接使用随机数进行随机初始化，而对于比较深的网络，则使
已经训练好的较浅的网络中的参数值对其前几层的卷积层和最后的全连接层进行初始化。

2.2 测试阶段

本篇论文中值得一提的是作者在测试阶段的操作，在测试阶段中，作者把测试图片的最短边设为Q，注意到这个Q
过程中图片的最短边S不一定相等，而且可以对于同一张测试图片可以rescale成不同的大小，即多个scale的同一张
在都在网络中进行测试，这样可以提升测试效果。在测试阶段，作者参考Sermanet等人的做法（Sermanet的做法
OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks这也是一篇非
的文章，以后再写关于它的阅读笔记）对于网络的全连接层转换为多个卷积层进行处理。

这里多讲一点，把其实全连接层是可以看做卷积层的，比如一个输入为1024个参数，输出为2048个参数的全连接
可以看做输入时1x1大小是的1024通道的feature maps做卷积核为1x1x1024的卷积，输出为1x1大小的2048通道
ure maps。

又比如一个输入为7x7大小，512通道的feature maps，输出为1024个参数的全连接层，就可以看做是输入为7x7
512通道的feature maps做卷积核为7x7x512的卷积，输出为1x1大小的1024通道的feature maps。

其实说白了就是把全连接层中的每个节点看做是一个1x1大小的feature map，而若有n个节点，则看做是1x1的n
eature maps。

不知道我说的清楚不清楚，大家可以参考知乎——全连接层的作用是什么？魏秀参的回答

好的，原归正传，作者在对测试图像进行了类似Sermanet的做法以后，最后得到的就是一个具有n通道的（n为最
分类的类别数目），a x b大小的一个feature map（这里a和b的大小取决于输入的测试图像的大小），最后把不同
的测试图像得到的结果进行了一个average操作，作为真正的分类结果。（其实这里不同scale的测试图像最后得到
道的axb feature map是不同的，Sermanet首先把一个scale中的axbxn的feature map合并为1x1xn大小的向量，
方法就是取这axb个不同向量中同一维（每个向量有n个维度）中最大的那个座位合并结果中该维度的值，最后不同
e的1x1xn的向量再做average操作）

这里有些绕，不知道我说清楚没= =|，有问题可以留言，我尽量说清楚些。

接下来就是介绍为什么使用Sermanet的做法，而不是使用Alexnet的做法，对一个测试图片进行多次crop（取中间
角的crop），因为Sermanet的做法可以一次计算即可获取同一张图片不同的区域的分类结果（上面提到的axb，即
不同的区域），而Alexnet则需要对不同的crop分别进行计算，这样就减少了计算量。另外，作者也提到了，如果
张图片切成大量的的crops，进行测试的话同样也可以提高正确率。但是由于大量的crops需要大量的计算，实际应
由于增加crops增加正确率相比增大的计算量明显不划算，所以实用性肯定不强，但是作为参考作者还是进行了多
的测试。

3. 实验结果

终于可以缓口气，大家来看一下作者的实验结果

Table 3: **ConvNet performance at a single test scale.**

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
A	256	256	29.6	10.4
A-LRN	256	256	29.7	10.5
B	256	256	28.7	9.9
C	256	256	28.1	9.4
	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
D	256	256	27.0	8.8
	384	384	26.8	8.7
	[256;512]	384	25.6	8.1
E	256	256	27.3	9.0
	384	384	26.9	8.7
	[256;512]	384	**25.5**	**8.0**

Table 4: **ConvNet performance at multiple test scales.**

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
B	256	224,256,288	28.2	9.6
C	256	224,256,288	27.7	9.2
	384	352,384,416	27.8	9.2
	[256; 512]	256,384,512	26.3	8.2
D	256	224,256,288	26.6	8.6
	384	352,384,416	26.5	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**
E	256	224,256,288	26.9	8.7
	384	352,384,416	26.7	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**

Table 5: **ConvNet evaluation techniques comparison.** In all experiments the training scale S was sampled from $[256; 512]$, and three test scales Q were considered: $\{256, 384, 512\}$.

ConvNet config. (Table 1)	Evaluation method	top-1 val. error (%)	top-5 val. error (%)
D	dense	24.8	7.5
	multi-crop	24.6	7.5
	multi-crop & dense	**24.4**	**7.2**
E	dense	24.8	7.5
	multi-crop	24.6	7.4
	multi-crop & dense	**24.4**	**7.1**

Table 6: **Multiple ConvNet fusion results.**

Combined ConvNet models	Error		
	top-1 val	top-5 val	top-5 test
ILSVRC submission			
(D/256/224,256,288), (D/384/352,384,416), (D/[256;512]/256,384,512) (C/256/224,256,288), (C/384/352,384,416) (E/256/224,256,288), (E/384/352,384,416)	24.7	7.5	7.3
post-submission			
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), dense eval.	24.0	7.1	7.0
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop	23.9	7.2	-
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop & dense eval.	**23.7**	**6.8**	**6.8**

以上一个图分别是单尺度测试、多尺度测试、大量crops测试、多模型投票的实验结果，可以看出的是深度越深，果越好，多尺度测试比单尺度测试结果要好，大量crops的测试结果可以进一步提高正确率，多模型若何也可以提率。

4. 总结

其实以现在（2017年）的角度来看，我觉得这篇文章介绍的VGGNet的网络结构创新性并不大，说白了就是堆网数。值得称道的是作者这样设计的原因，能够分析小卷积核的作用，总体来说还是大神一样的存在，要不然现在就很多人直接拿VGGNet在自己的任务上训练了。

- 上一篇 Ubuntu 16.04无法安装第三方deb软件的解决方法
- 下一篇 [深度学习]Deep Residual Learning for Image Recognition(ResNet,残差网络)阅读笔记