1.简介

2.网络结构

2.1 Deep-wise卷积

2.2 网络结构

2.3 训练细节

2.4 宽度因子和分辨率因子

1.简介

论文地址：https://arxiv.org/abs/1704.04861

深度学习在图像分类，目标检测和图像分割等任务表现出了巨大的优越性。

但是伴随着模型精度的提升是计算量，存储空间以及能耗方面的巨大开销，对于移动或车载应用都是难以接受的。

之前的一些模型小型化工作是将焦点放在模型的尺寸上。

因此，在小型化方面常用的手段有：

（1）卷积核分解，使用1×N和N×1的卷积核代替N×N的卷积核

（2）使用bottleneck结构，以SqueezeNet为代表

（3）以低精度浮点数保存，例如Deep Compression

（4）冗余卷积核剪枝及哈弗曼编码

MobileNet进一步深入的研究了depthwise separable convolutions使用方法后设计出MobileNet，depthwiseseparable convolutions的本质是冗余信息更少的稀疏化表达。在此基础上给出了高效模型设计的两个选择：宽度因子（width multiplier）和分辨率因子（resolutionmultiplier）；通过权衡大小、延迟时间以及精度，来构建规模更小、速度更快的MobileNet。Google团队也通过了多样性的实验证明了MobileNet作为高效基础网络的有效性。

2.网络结构

2.1 Deep-wise卷积

MobileNet使用了一种称之为deep-wise的卷积方式来替代原有的传统3D卷积，减少了卷积核的冗余表达。在计算量和参数数量明显下降之后，卷积网络可以应用在更多的移动端平台。

传统的3D卷积使用一个和输入数据通道数相同的卷积核在逐个通道卷积后求和最后得出一个数值作为结果，计算量为

$$M \times D_k \times D_k$$

其中M为输入的通道数，D_k为卷积核的宽和高

一个卷积核处理输入数据时的计算量为（有Padding）：

$$D_k \times D_k \times M \times D_F \times D_F$$

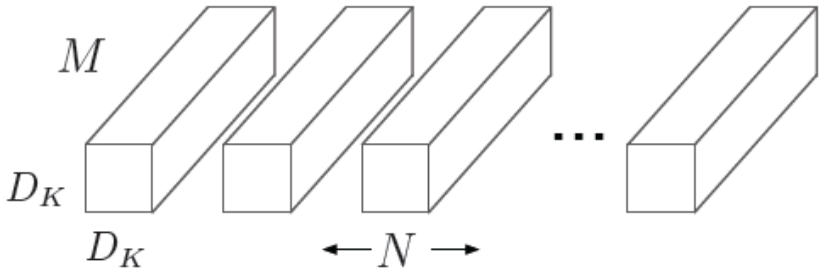
其中D_F为输入的宽和高

在某一层如果使用N个卷积核，这一个卷积层的计算量为：

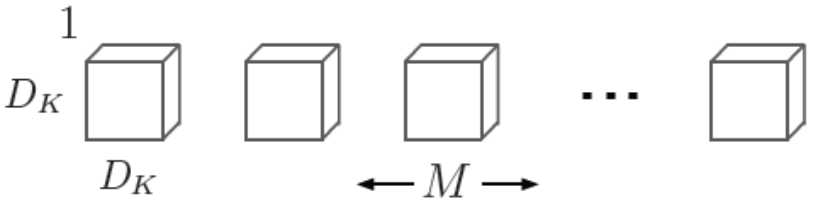
$$D_k \times D_k \times N \times M \times D_F \times D_F$$

如果使用deep-wise方式的卷积核，我们会首先使用一组二维的卷积核，也就是卷积核的通道数为1，每次只处理一个输入通道的，这一组二维卷积核的数量是和输入通道数相同的。

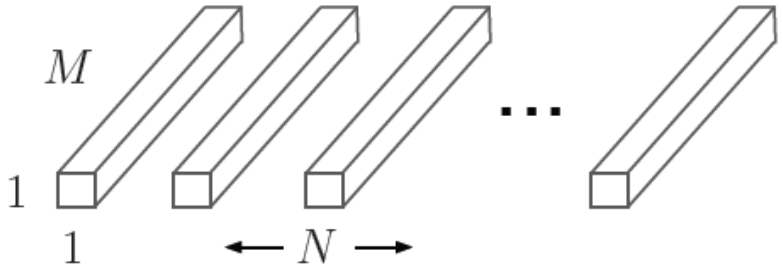
在使用逐个通道卷积处理之后，再使用3D的1*1卷积核来处理之前输出的特征图，将最终输出通道数变为一个指定的数量，论文原图说明的比较到位，请看



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

图a中的卷积核就是最常见的3D卷积，替换为deep-wise方式：一个逐个通道处理的2D卷积（图b）结合3D的1*1卷积（图c）

从理论上来看，一组和输入通道数相同的2D卷积核的运算量为：

$$D_k \times D_k \times M \times D_F \times D_F$$

3D的1*1卷积核的计算量为：

$$N \times M \times D_F \times D_F$$

因此这种组合方式的计算量为：

$$D_k \times D_k \times M \times D_F \times D_F + N \times M \times D_F \times D_F$$

deep-wise方式的卷积相比于传统3D卷积计算量为：

$$\frac{D_k \times D_k \times M \times D_F \times D_F + N \times M \times D_F \times D_F}{D_k \times D_k \times N \times M \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_k^2}$$

举一个具体的例子，给定输入图像的为3通道的224x224的图像，VGG16网络的第3个卷积层conv2_1输入的是尺寸为112的特征图，通道数为64，卷积核尺寸为3，卷积核个数为128，传统卷积运算量就是

$$3 \times 3 \times 128 \times 64 \times 112 \times 112 = 924844032$$

如果将传统3D卷积替换为deep-wise结合1x1方式的卷积，计算量为：

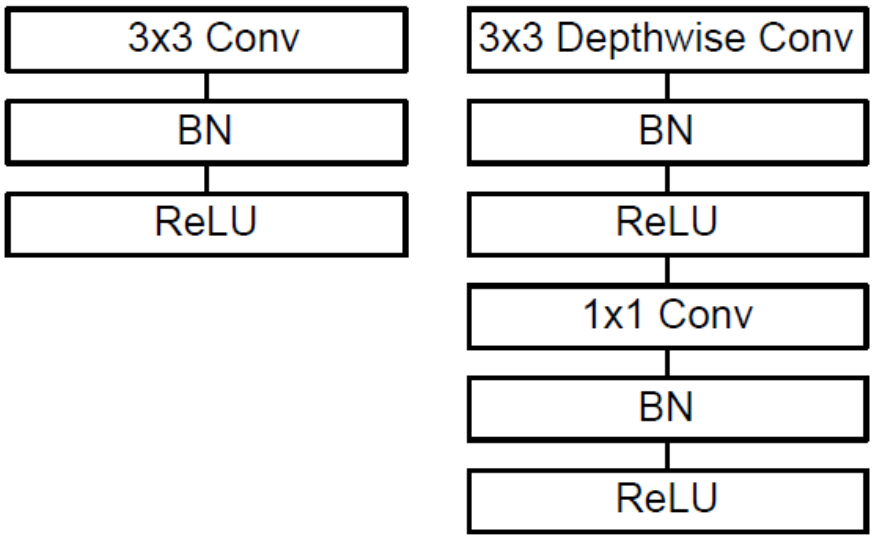
$$3 \times 3 \times 64 \times 112 \times 112 + 128 \times 64 \times 112 \times 112 = 109985792$$

可见在这一层里，MobileNet所采用卷积方式的计算量与传统卷积计算量的比例为：

$$\frac{109985792}{924844032} = 0.1189$$

2.2 网络结构

传统的3D卷积常见的使用方式如下图左侧所示，deep-wise卷积的使用方式如下图右边所示。



2.3 训练细节

作者基于TensorFlow训练MobileNet，使用RMSprop算法优化网络参数。考虑到较小的网络不会有严重的过拟合问题，因此没有做大量的数据增强工作。在训练过程中也没有采用训练大网络时的一些常用手段，例如：辅助损失函数，随机图像裁剪输入等。

deep-wise卷积核含有的参数较少，作者发现这部分最好使用较小的weight decay或者不使用weightdecay。

2.4 宽度因子和分辨率因子

尽管标准的MobileNet在计算量和模型尺寸方面具备了很明显的优势，但是，在一些对运行速度或内存有极端要求的场合，还需要更小更快的模型，如何能够在不重新设计模型的情况下，以最小的改动就可以获得更小更快的模型呢？本文中提出的宽度因子（width multiplier）和分辨率因子（resolutionmultiplier）就是解决这些问题的配置参数。

宽度因子α是一个属于(0,1]之间的数，附加于网络的通道数。简单来说就是新网络中每一个模块要使用的卷积核数量相较于标准的MobileNet比例。对于deep-wise结合1x1方式的卷积核，计算量为：

$$D_k \times D_k \times \alpha M \times D_F \times D_F + \alpha N \times \alpha M \times D_F \times D_F$$

α常用的配置为1,0.75,0.5,0.25；当α等于1时就是标准的MobileNet。通过参数α可以非常有效的将计算量和参数数量约减到α的平方倍。

通过下图可以看出使用α系数进行网络参数的约减时，在ImageNet上的准确率，为准确率，参数数量和计算量之间的权衡提供了参考（每一个项中最前面的数字表示α的取值）。

Width Multiplier	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
0.75 MobileNet-224	68.4%	325	2.6
0.5 MobileNet-224	63.7%	149	1.3
0.25 MobileNet-224	50.6%	41	0.5

分辨率因子β的取值范围在(0,1]之间，是作用于每一个模块输入尺寸的约减因子，简单来说就是将输入数据以及由此在每一个模块产生的特征图都变小了，结合宽度因子α，deep-wise结合1x1方式的卷积核计算量为：

$$D_k \times D_k \times \alpha M \times \beta D_F \times \beta D_F + \alpha N \times \alpha M \times \beta D_F \times \beta D_F$$

下图为使用不同的β系数作用于标准MobileNet时，对精度和计算量以的影响（α固定）

Resolution	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
1.0 MobileNet-192	69.1%	418	4.2
1.0 MobileNet-160	67.2%	290	4.2
1.0 MobileNet-128	64.4%	186	4.2

要注意再使用宽度和分辨率参数调整网络结构之后，都要从随机初始化重新训练才能得到新网络。