**轻量化神经网络模型总结**

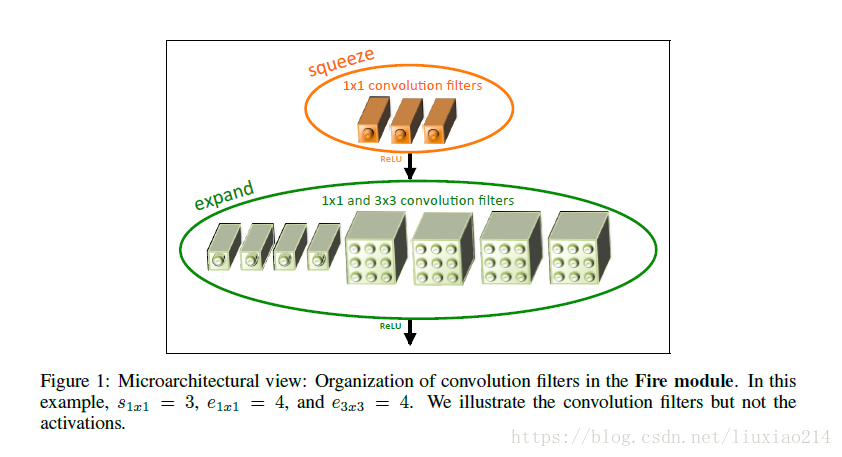
**一、SqueezeNet**

在ImageNet上实现了与Alexnet相似的效果，参数只有其1/50， 模型是0.5MB，占其1/510。

SqueezeNet核心内容有以下几点：

1. 使用1x1卷积核代替3x3卷积核，减少参数量；
2. 通过squeeze layer限制通道数量，减少参数量；
3. 借鉴inception思想，将1x1和3x3卷积后结果进行concat；为了使其feature map的size相同，3x3卷积核进行了padding；
4. 减少池化层，并将池化操作延后，给卷积层带来更大的激活层，保留更多地信息，提高准确率；
5. 使用全局平均池化代替全连接层；

上述1-3是通过fire module实现的，fire module主要分为两部分，如下图所示

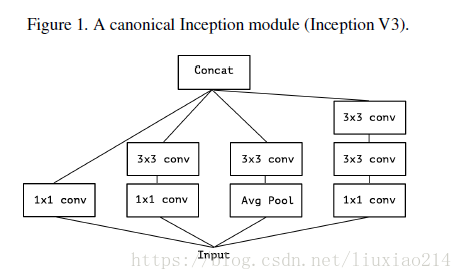


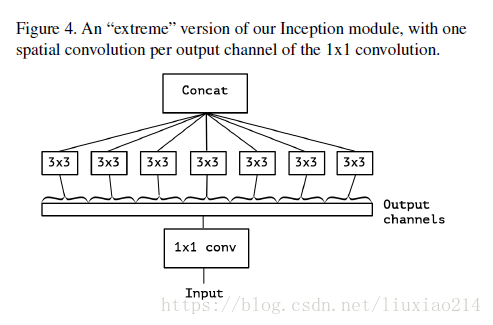
* squeeze：1x1卷积核，参数s\_1x1表示卷积核数量
* expand：1x1卷积核和3x3卷积核，参数e\_1x1和e\_3x3分别表示两种卷积核的数量

该模块一共三参数s\_1x1、e\_1x1、e\_3x3，关系保持s\_1x1< e\_1x1+e\_3x3

**二、Xception**

虽然本文中方法可以降低参数量，但是论文加宽了网络结构，因此这篇论文不在于压缩模型，旨在于提高性能，与同等参数量的inception v3相比，效果更好。 首先是inception v3的一系列延伸，见下图：





然后在xception中，主要采用depthwise separable convolution思想。

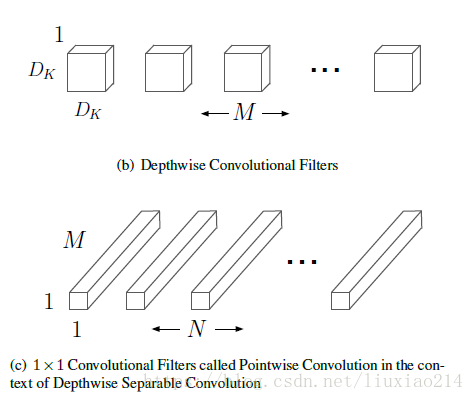
首先xception类似于图4，但是区别有两点：

1. xception中没有relu激活函数；
2. 图4是先1x1卷积，后通道分离；xception是先进行通道分离，即depthwise separable convolution，然后再进行1x1卷积。
3. 此外，进行残差连接时，不再是concat，而是采用加法操作。

**三、MobileNet**

主要是两个策略： 1. 采用depthwise separable convolution，就是分离卷积核；

1. 设置宽度因子width multipler和分辨率因子resolution multiplier；
2. depthwise separable convolution（分通道卷积）



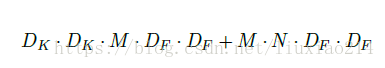
将卷积核进行拆解，分为两步，首先用M个D\_K \* D\_K卷积核在feature map进行卷积，计算量为

IMG_256

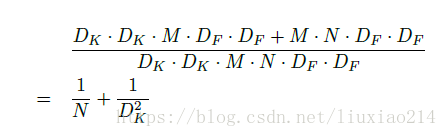
然后再使用N个1 \* 1 \* M卷积核在前面得到的结果上进行feature map，计算量为：

IMG_256

所以，进行分解后的总计算量为：



计算量比较



可以看到，随着卷积核个数的增加，即通道数变多，feature map的大小，传统方式的计算量比分解要大得多

1. 宽度因子和分辨率因子

怎么才能使网络进一步压缩呢？可以进一步减少feature map的通道数和size，通过宽度因子减少通道数，分辨率因子减少size。

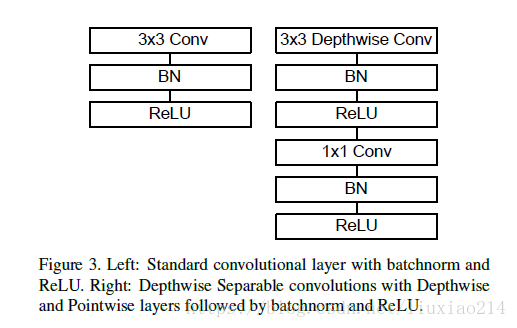
* 宽度因子α

IMG_256

* 分辨率因子β

IMG_256

两个参数都属于(0,1]之间，当为1时则是标准mobileNet。



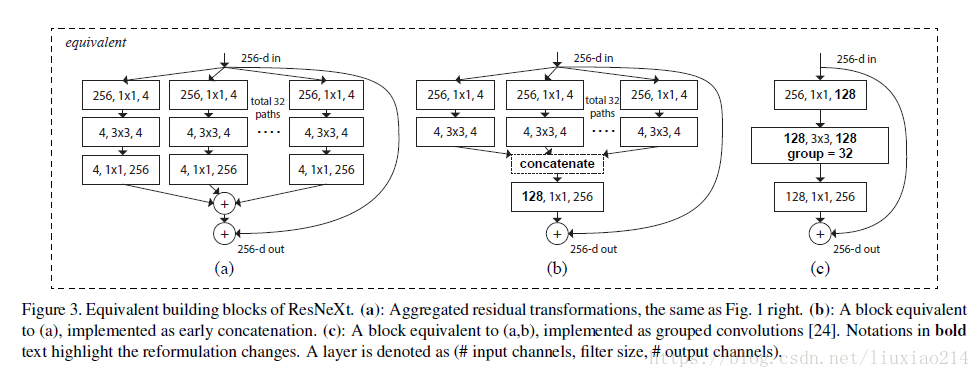
**四、核心思想**

核心思想有两点：

* 借鉴resnext分组卷积思想，但不同的是采用1x1卷积核；
* 进行通道清洗，加强通道间的信息流通，提高信息表示能力。
* 此外本篇论文中也采取了mobilenet的depthwise separasable convolution的方式。

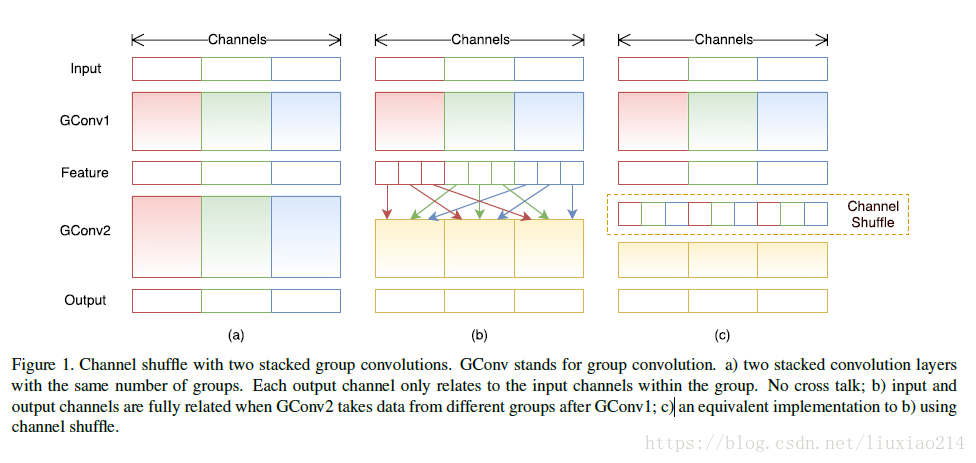
1. 逐点群卷积pointwise group convolution

这个就是采用resnext的思想，将通道分组，每组分别进行卷积操作，然后再把结果进行concat。但是不同于resnext的是，shufflenet采用的是1x1卷积核。

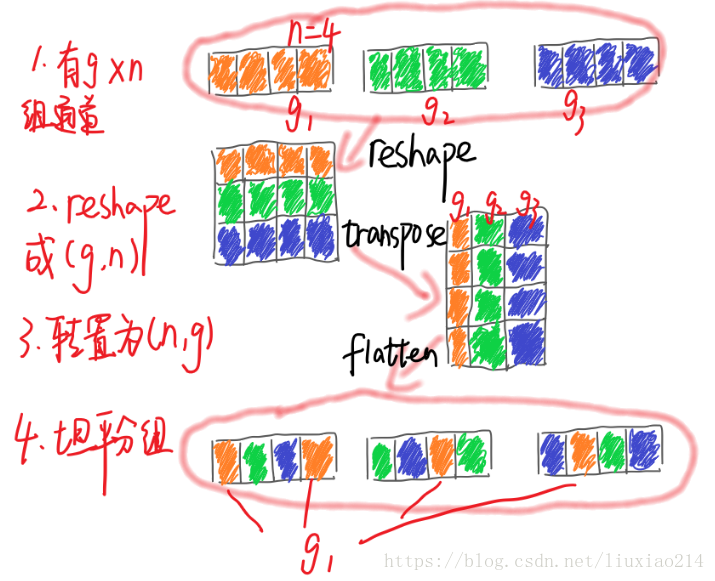


1. 通道清洗channel shuffle

什么是通道shuffle，就是在分组卷积后得到的feature map不直接进行concat，先将每组feature map按通道打乱，重新concat，如下图所示：

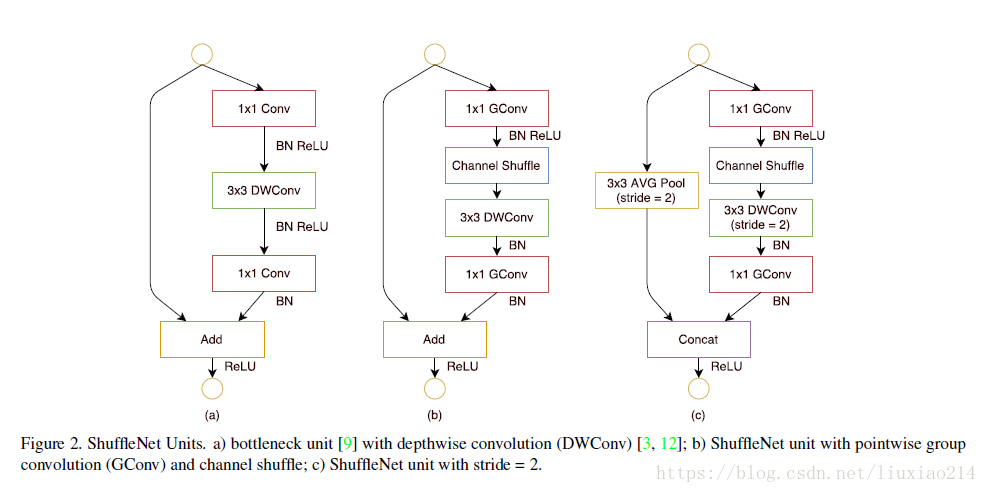
 对于一个卷积层分为g组，

1. 卷积后一共得到g×n个输出通道的feature map；
2. 将feature map 进行 reshape为(g,n);
3. 进行转置为(n,g)；
4. 对转置结果flatten，再分回g组作为下一层的输入。



1. shuffle unit

下图中，a是标准的残差结构，不过是3x3卷积核使用了mobilenet中的depthwise convolution操作； b是在a的基础上加了本文的通道shuffle操作，先对1x1卷积进行分组卷积操作，然后进行channel shuffle； c是在旁路加了一步长为2的3x3的平均池化，并将前两者残差相加的操作改为了通道concat，增加了通道数量！

参考文献：<https://blog.csdn.net/liuxiao214/article/details/81875251>