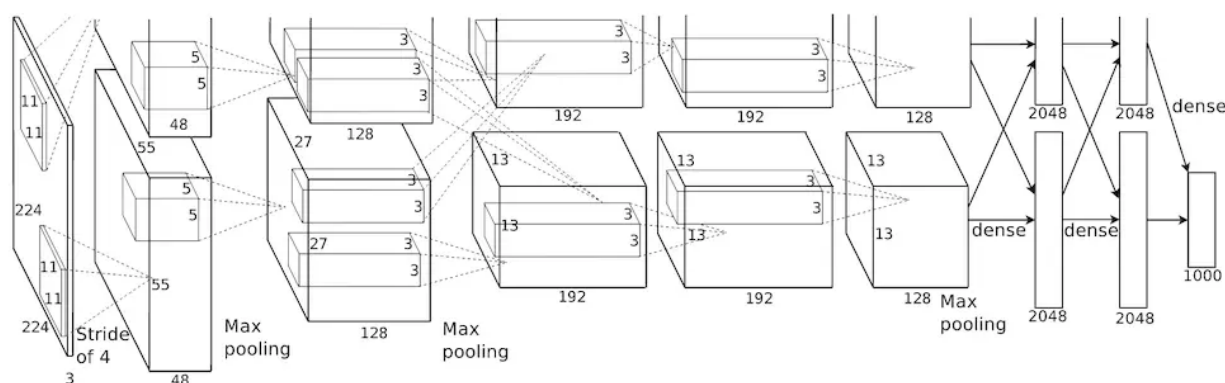


- 1.卷积只能在同一组进行吗? -Group convolution
- 2.卷积核一定越大越好? -3x3卷积核 1x1卷积核
- 3.每层卷积只能用一种尺寸的卷积核? -Inception结构
- 4.怎样才能减少卷积层参数量? --Bottleneck
- 5.越深的网络越难训练吗? -resnet
- 6.卷积操作时所有通道都只能用同一个过滤器吗? -Depthwise操作
- 7.分组卷积能否对通道进行随机分组? -ShuffleNet
- 8.通道间的特征都是平等的吗? -SENet
- 9.能否让固定大小的卷积核看到更大范围的区域? -Dilated convolution
- 10.卷积核形状一定是矩形吗? -Deformable convoluton可变形卷积

启发与思考

1.卷积只能在同一组进行吗? -Group convolution

分组卷积，最早在Alexnet中出现，作者把feature maps分给多个gpu分别进行处理，最后把多个gpu的结果进行融合。



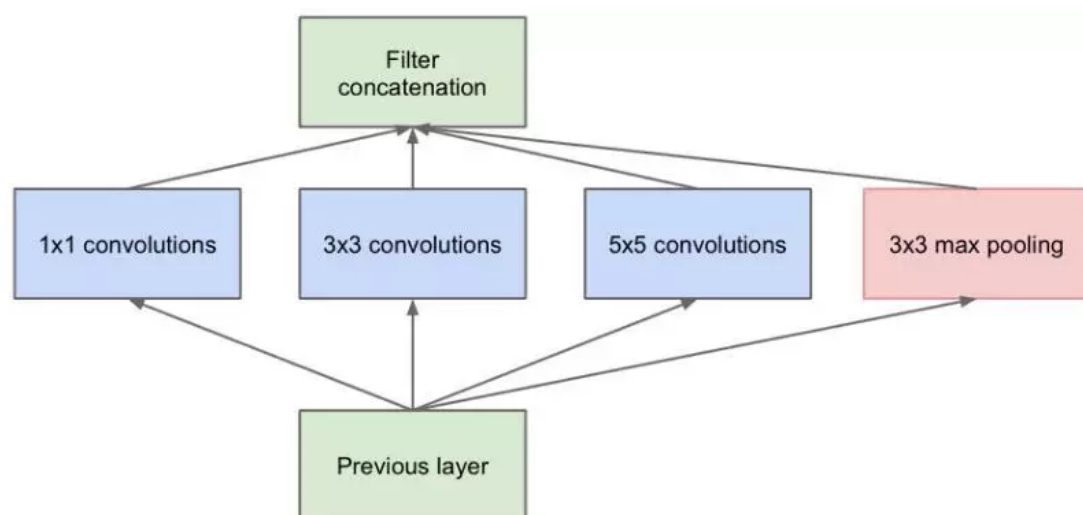
分组卷积的思想影响比较深远，当前一些轻量级的SOTA（State Of The Art）网络，都用到了分组卷积的操作，以节省计算量。

2.卷积核一定越大越好? -3x3卷积核 1x1卷积核

之前人们的观念是，卷积核越大，感受野越大，看到的图片信息越多，因此获得的特征越好。但是大的卷积核会导致计算量的暴增，不利于模型深度的增加。在Inception网络中，作者提出用2个3x3卷积核的组合比1个5x5卷积核的效果更佳，同时参数量（ $3 \times 3 \times 2 + 1 < 5 \times 5 + 1$ ）降低，后来该方法被广泛应用。

3.每层卷积只能用一种尺寸的卷积核？ -Inception结构

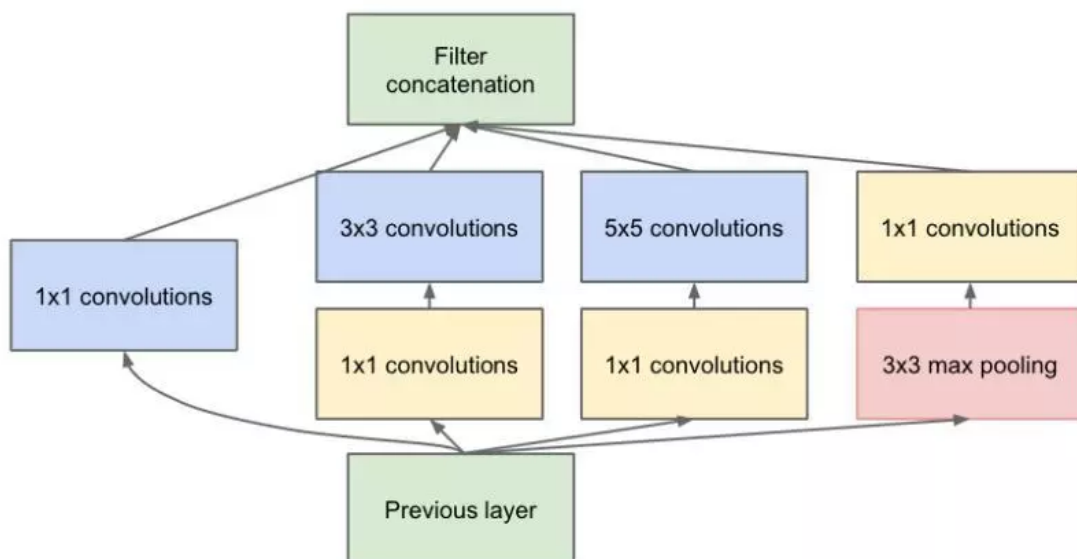
传统的层叠式网络，基本上都是一个卷积层的堆叠，每层只用一个尺寸的卷积核，例如VGG结构中使用了大量的 3×3 卷积层。事实上，同一层feature map可以分别使用多个不同尺寸的卷积核，以获得不同尺度的特征，再把这些特征结合起来，得到的特征往往比使用单一卷积核的要好，因此谷歌发明的GoogleNet，或者说Inception系列的网络，就使用了多个卷积核的结构：



如上图所示，一个输入的feature map分别同时经过 1×1 、 3×3 、 5×5 的卷积核的处理，得出的特征再组合起来，获得更佳的特征。但这个结构会存在一个严重的问题：参数量比单个卷积核要多很多，如此庞大的计算量会使得模型效率低下。这就引出了一个新的结构：

4.怎样才能减少卷积层参数量？ --Bottleneck

如果仅仅引入多个尺寸的卷积核，会带来大量的额外的参数，为了解决这个问题，他们往Inception结构中加入了一些 1×1 的卷积核，如图所示：

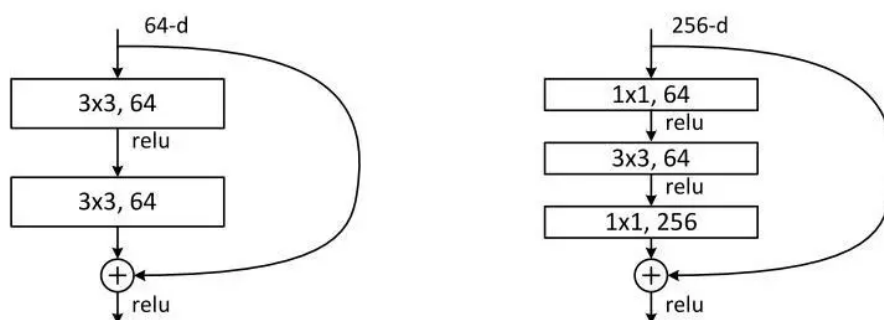


根据上图，我们看着 $1 \times 1 \rightarrow 3 \times 3$ 这段通路，来做个计算，假设输入 feature map 的维度为256维，要求输出维度也是256维。有以下两种操作：

1. 256维的输入直接经过一个 $3 \times 3 \times 256$ 的卷积层，输出一个256维的 feature map，那么参数量为： $256 \times 3 \times 3 \times 256 = 589,824$
2. 256维的输入先经过一个 $1 \times 1 \times 64$ 的卷积层，再经过一个 $3 \times 3 \times 64$ 的卷积层，最后经过一个 $3 \times 3 \times 256$ 的卷积层，输出256维，参数量为：
 $256 \times 1 \times 1 \times 64 + 64 \times 3 \times 3 \times 64 + 64 \times 1 \times 1 \times 256 = 69,632$ 。足足把第一种操作的参数量降低到九分之一！

1×1 卷积核也被认为是影响深远的操作，往后大型的网络为了降低参数量都会应用上 1×1 卷积核。

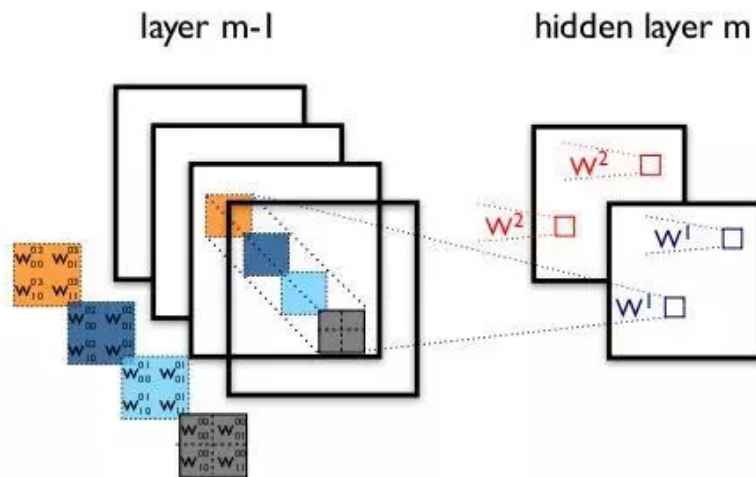
5.越深的网络越难训练吗？ -resnet



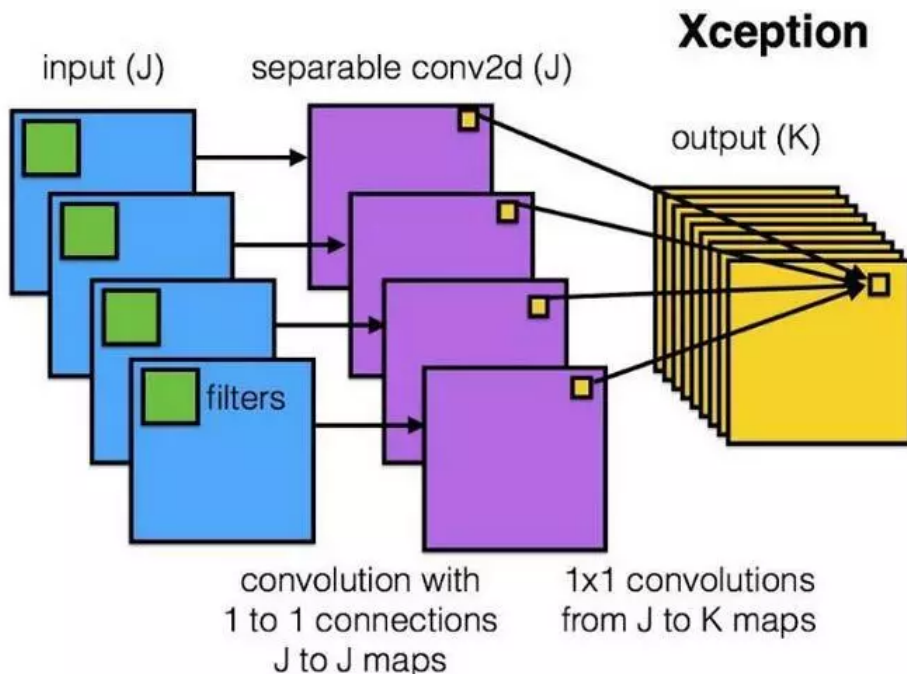
传统的卷积层层叠网络会遇到一个问题，当层数加深时，网络的表现越来越差，很大程度上原因是因为当层数加深时，梯度消散得越来越严重，以至于反向传播很难训练到

浅层的网络。为了解决这个问题，何凯明大神想出了一个“残差网络”，使得梯度更容易地流动到浅层的网络当中去。

6.卷积操作时所有通道都只能用同一个过滤器吗？ - Depthwise操作



标准的卷积过程可以看上图，一个 2×2 的卷积核在卷积时，对应图像区域中的所有通道均用同一个过滤器，问题在于，为什么一定要同时考虑图像区域和通道？我们为什么不能每个通道分开考虑？



Xception网络就是基于以上的问题发明而来。我们首先对每一个通道进行各自的卷积操作，有多少个通道就有多少个过滤器。得到新的通道feature maps之后，这时再对这批新的通道feature maps进行标准的 1×1 跨通道卷积操作。这种操作被称为“DepthWise convolution”，缩写“DW”。

这种操作是相当有效的，在imagenet 1000类分类任务中已经超过了InceptionV3的表现，而且也同时减少了大量的参数，我们来算一算，假设输入通道数为3，要求输出通道数为256，两种做法：

1. 直接接一个 $3 \times 3 \times 256$ 的卷积核，参数量为： $3 \times 3 \times 3 \times 256 = 6,912$
2. DW操作，分两步完成，参数量为： $3 \times 3 \times 3 + 3 \times 1 \times 1 \times 256 = 795$ ，又把参数量降低到九分之一！

因此，一个depthwise操作比标准的卷积操作降低不少的参数量，同时得到更好的效果，因为它对每一个通道都进行了学习（每个通道对应一个不同的过滤器），而不是所有通道对应同一个过滤器，得到的特征质量更佳！

7.分组卷积能否对通道进行随机分组？-ShuffleNet

在AlexNet的Group Convolution当中，特征的通道被平均分到不同组里面，最后再通过两个全连接层来融合特征，这样一来，就只能在最后时刻才融合不同组之间的特征，对模型的泛化性是相当不利的。为了解决这个问题，ShuffleNet在每一次层叠这种Group conv层前，都进行一次channel shuffle，shuffle过的通道被分配到不同组当中。进行完一次group conv之后，再一次channel shuffle，然后分到下一层组卷积当中，以此循环。

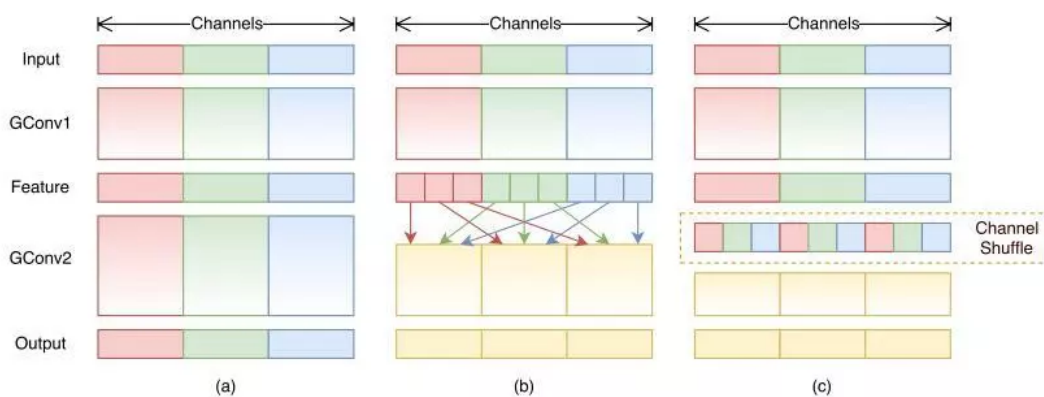
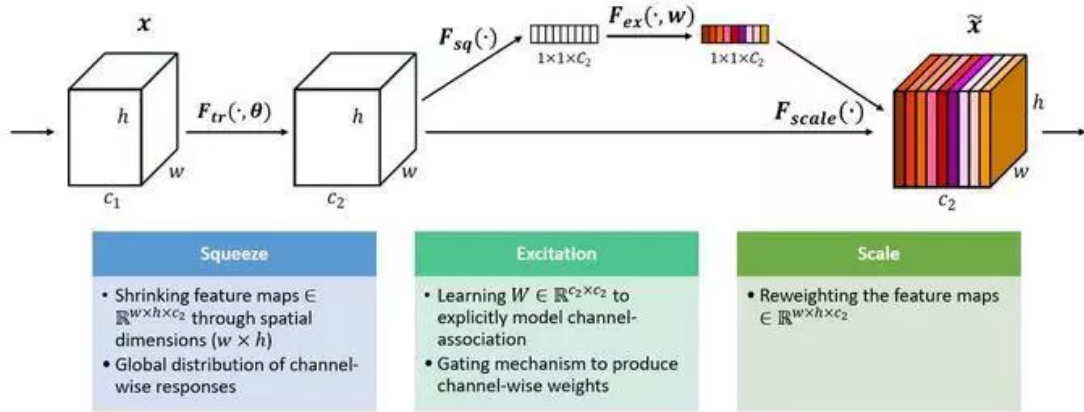


Figure 1: Channel shuffle with two stacked group convolutions. GConv stands for group convolution. a) two stacked convolution layers with the same number of groups. Each output channel only relates to the input channels within the group. No cross talk; b) input and output channels are fully related when GConv2 takes data from different groups after GConv1; c) an equivalent implementation to b) using channel shuffle.

经过channel shuffle之后，Group conv输出的特征能考虑到更多通道，输出的特征自然代表性就更高。另外，AlexNet的分组卷积，实际上是标准卷积操作，而在ShuffleNet里面的分组卷积操作是depthwise卷积，因此结合

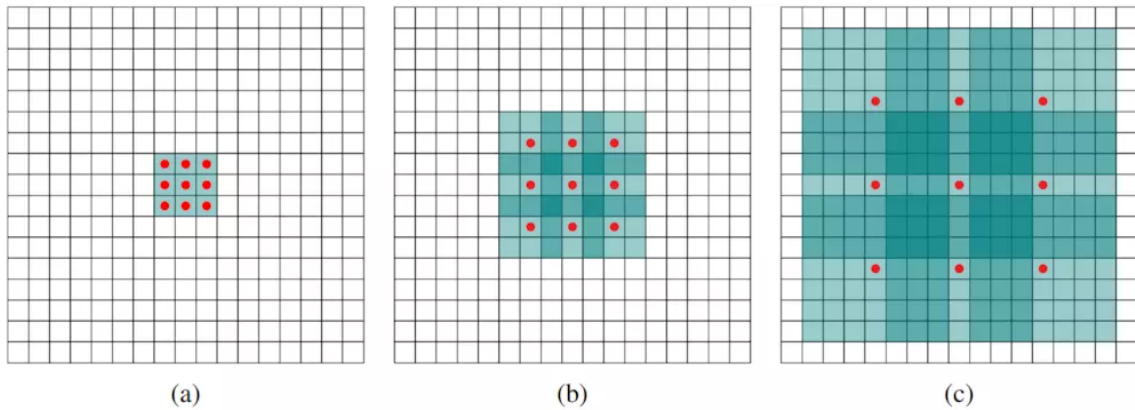
了通道洗牌和分组depthwise卷积的ShuffleNet，能得到超少量的参数以及超越mobilenet、媲美AlexNet的准确率！

8.通道间的特征都是平等的吗？-SENet



一组特征在上一层被输出，这时候分两条路线，第一条直接通过，第二条首先进行Squeeze操作（Global Average Pooling），把每个通道2维的特征压缩成一个1维，从而得到一个特征通道向量（每个数字代表对应通道的特征）。然后进行Excitation操作，把这一列特征通道向量输入两个全连接层和sigmoid，建模出特征通道间的相关性，得到的输出其实就是每个通道对应的权重，把这些权重通过Scale乘法通道加权到原来的特征上（第一条路），这样就完成了特征通道的权重分配。

9.能否让固定大小的卷积核看到更大范围的区域？ - Dilated convolution



上图b可以理解为卷积核大小依然是 3×3 ，但是每个卷积点之间有1个空洞，也就是在绿色 7×7 区域里面，只有9个红色点位置作了卷积处理，其余点权重为0。这样即使卷积核大小不变，但它看到的区域变得更大了

10.卷积核形状一定是矩形吗？-Deformable convolution可变形卷积

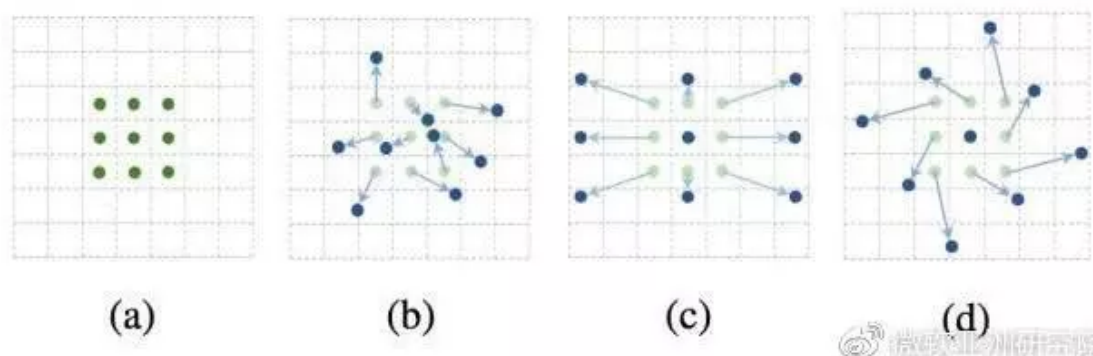


图1：展示了卷积核大小为 3×3 的正常卷积和可变形卷积的采样方式，(a) 所示的正常卷积规律的采样 9 个点（绿点），(b)(c)(d) 为可变形卷积，在正常的采样坐标上加上一个位移量（蓝色箭头），其中(c)(d) 作为 (b) 的特殊情况，展示了可变形卷积可以作为尺度变换，比例变换和旋转变换的特殊情况

传统的卷积核一般都是长方形或正方形，但MSRA提出了一个相当反直觉的见解，认为卷积核的形状可以是变化的，变形的卷积核能让它只看感兴趣的图像区域，这样识别出来的特征更佳。

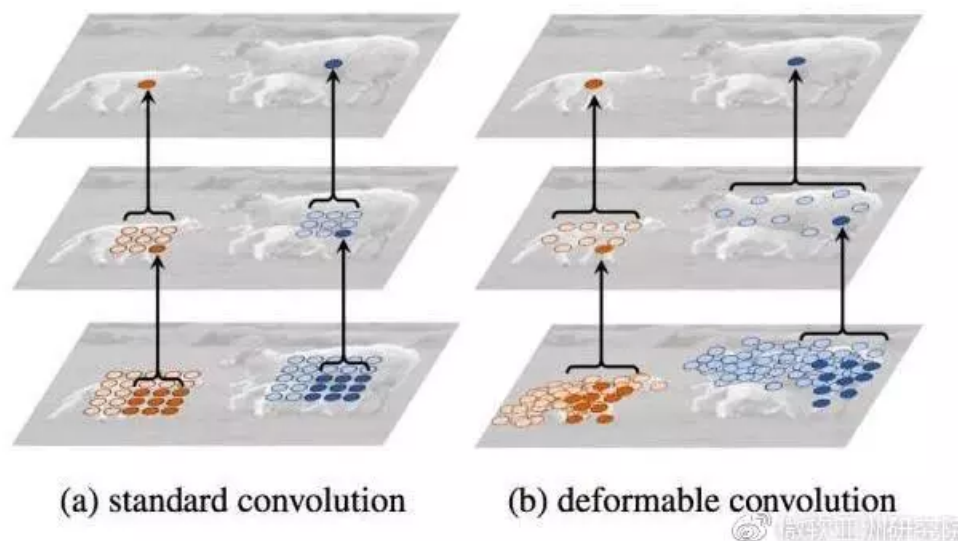


图2：两层 3×3 的标准卷积和可变形卷积的区别。(a) 标准卷积中固定的感受野和卷积核采样点。(b) 可变形卷积中自适应的感受野和卷积核采样点。

要做到这个操作，可以直接在原来的过滤器前面再加一层过滤器，这层过滤器学习的是下一层卷积核的位置偏移量（offset），这样只是增加了一

层过滤器，或者直接把原网络中的某一层过滤器当成学习offset的过滤器，这样实际增加的计算量是相当少的，但能实现可变形卷积核，识别特征的效果更好。

启发与思考

现在越来越多的CNN模型从巨型网络到轻量化网络一步步演变，模型准确率也越来越高。现在工业界追求的重点已经不是准确率的提升（因为都已经很高了），都聚焦于速度与准确率的trade off，都希望模型又快又准。因此从原来AlexNet、VGGnet，到体积小一点的Inception、Resnet系列，到目前能移植到移动端的mobilenet、ShuffleNet（体积能降低到0.5mb!），我们可以看到这样一些趋势：

卷积核方面：

1. 大卷积核用多个小卷积核代替；
2. 单一尺寸卷积核用多尺寸卷积核代替；
3. 固定形状卷积核趋于使用可变形卷积核；
4. 使用 1×1 卷积核（bottleneck结构）。

卷积层通道方面：

1. 标准卷积用depthwise卷积代替；
2. 使用分组卷积；
3. 分组卷积前使用channel shuffle；
4. 通道加权计算。

卷积层连接方面：

1. 使用skip connection，让模型更深；
2. densely connection，使每一层都融合上其它层的特征输出
(DenseNet)

启发

类比到通道加权操作，卷积层跨层连接能否也进行加权处理？bottleneck + Group conv + channel shuffle + depthwise的结合会不会成为以后降低参数量的标准配置？