



## CNN中千奇百怪的卷积方式大汇总



Xf Mao · 3 个月前

2017.9.19 更新：

补充了两个新的卷积工作：**Deformable Convolution**以及**Squeeze\_excitation Convolution**，实现的demo样例我后续会贴出来。

推仔力一篇很棒的总结：[支形卷积核、可刀两卷积：卷积神经网络中让人拍案叫绝的卷积。](#)

这篇总结很专业，并且很好的回答了评论中的疑问：

Depthwise Separable Convolution就是卷积加group吧？

这篇文章里是这么说的：

要注意的是，**Group conv**是一种channel分组的方式，**Depthwise +Pointwise**是卷积的方式，只是**ShuffleNet**里面把两者应用起来了。因此**Group conv**和**Depthwise +Pointwise**并不能划等号。

我觉得Depthwise +Pointwise卷积是提供一种把feature map的空间信息（height & width）和通道信息（channel）拆分分别处理的方法，而group卷积只是单纯的通道分组处理，降低复杂度。

对于表述有误的地方欢迎更正！另外目前仅列举了一些比较著名且实用的卷积操作，对于没提到的，希望大家指出与补充

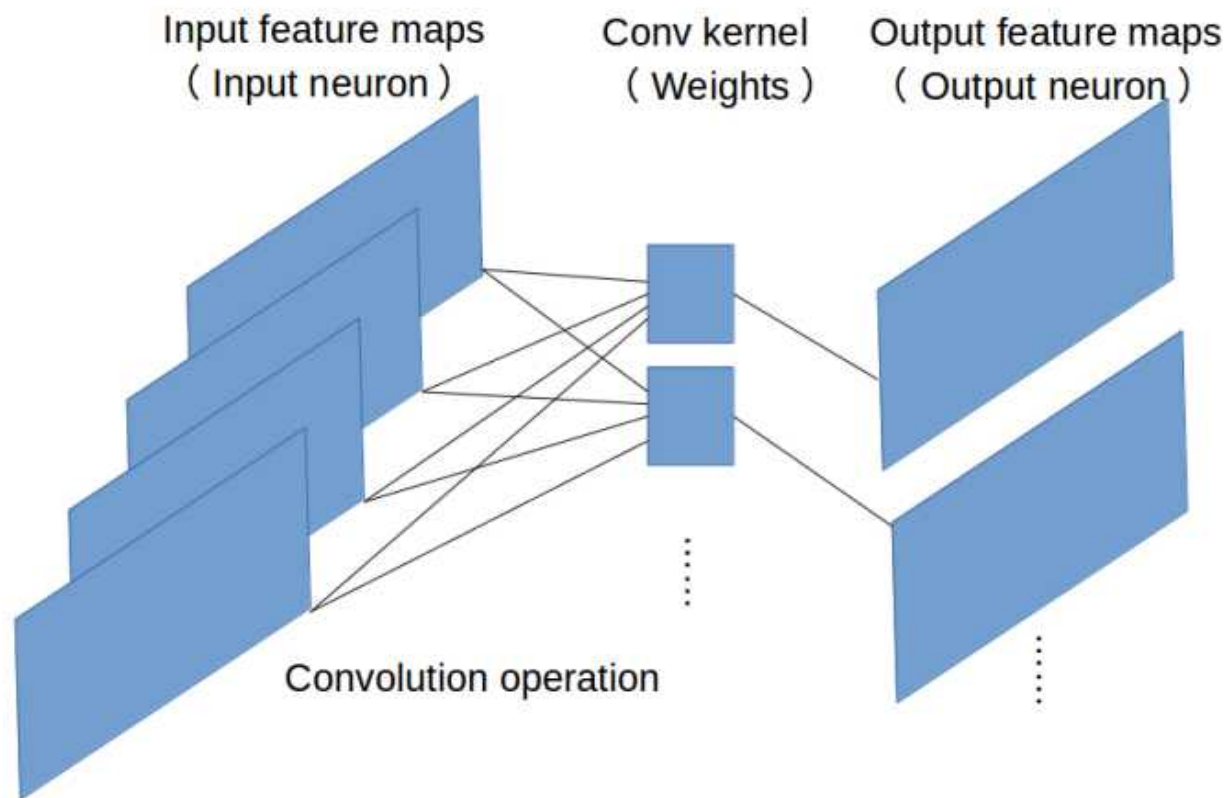
## 1.原始版本

取千的卷积方式还没有任何独长路，那就也没什么好说的了。

见下图，原始的conv操作可以看做一个2D版本的无隐层神经网络。

附上一个卷积详细流程：

【TensorFlow】tf.nn.conv2d是怎样实现卷积的？ - CSDN博客



LeNet：敢于使用stack单卷积+单池化结构的方式，卷积层不做特征提取，池化不做空间下采样

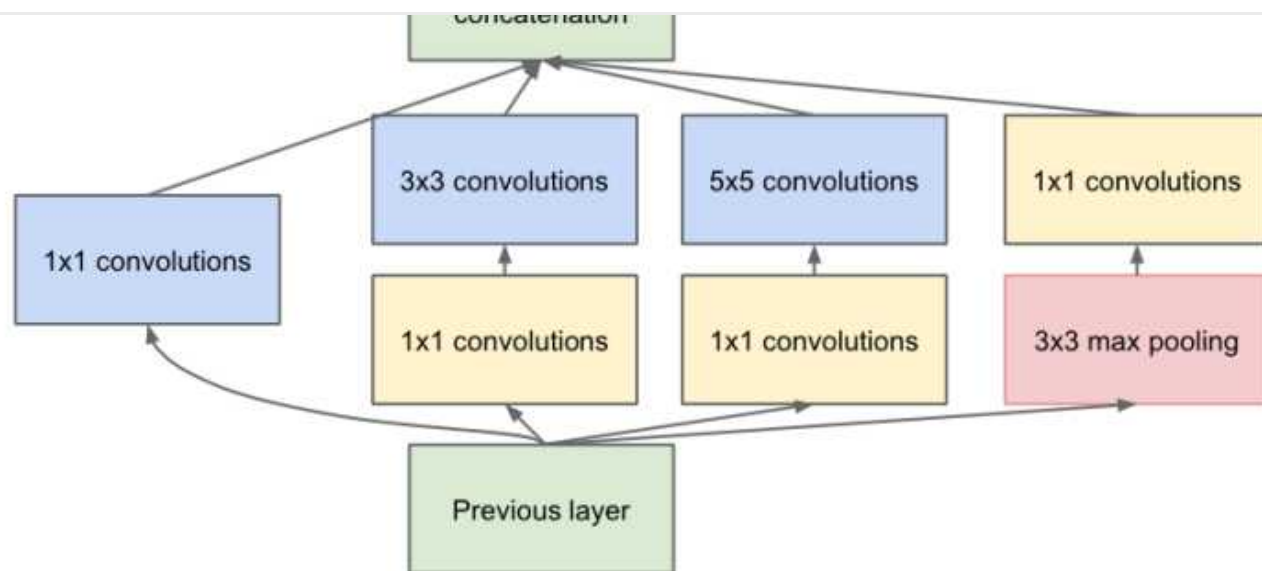
AlexNet：后来发现单卷积提取到的特征不是很丰富，于是开始stack多卷积+单池化的结构

VGG ([1409.1556] [Very Deep Convolutional Networks for Large-Scale Image Recognition](#))：结构没怎么变，只是更深了

## 2.多隐层非线性版本

这个版本是一个较大的改进，融合了Network In Network的增加隐层提升非线性表达的思想，于是有了这种先用 $1 \times 1$ 的卷积映射到隐空间，再在隐空间做卷积的结构。同时考虑了多尺度，在单层卷积层中用多个不同大小的卷积核来卷积，再把结果concat起来。

这一结构，被称之为“Inception”



代表模型：

Inception-v1 ( [\[1409.4842\] Going Deeper with Convolutions](#) )：stack以上这种Inception结构

Inception-v2 ( [Accelerating Deep Network Training by Reducing Internal Covariate Shift](#) )：  
加了BatchNormalization正则，去除5\*5卷积，用两个3\*3代替

Inception-v3 ( [\[1512.00567\] Rethinking the Inception Architecture for Computer Vision](#) )：  
7\*7卷积又拆成7\*1+1\*7

### 3.空洞卷积

Dilation卷积，通常译作空洞卷积或者卷积核膨胀操作，它是解决pixel-wise输出模型的一种常用的卷积方式。一种普遍的认识是，pooling下采样操作导致的信息丢失是不可逆的，通常的分类识别模型，只需要预测每一类的概率，所以我们不需要考虑pooling会导致损失图像细节信息的问题，但是做像素级的预测时（譬如语义分割），就要考虑到这个问题了。

所以就要有一种卷积代替pooling的作用（成倍的增加感受野），而空洞卷积就是为了做这个的。通过卷积核插“0”的方式，它可以比普通的卷积获得更大的感受野，这个idea的motivation就介绍到这里。具体实现方法和原理可以参考如下链接：

[如何理解空洞卷积（dilated convolution）？](#)

[膨胀卷积--Multi-scale context aggregation by dilated convolutions](#)

我在博客里面又做了一个空洞卷积小demo方便大家理解

[【Tensorflow】tf.nn.atrous\\_conv2d如何实现空洞卷积？ - CSDN博客](#)

代表候望。

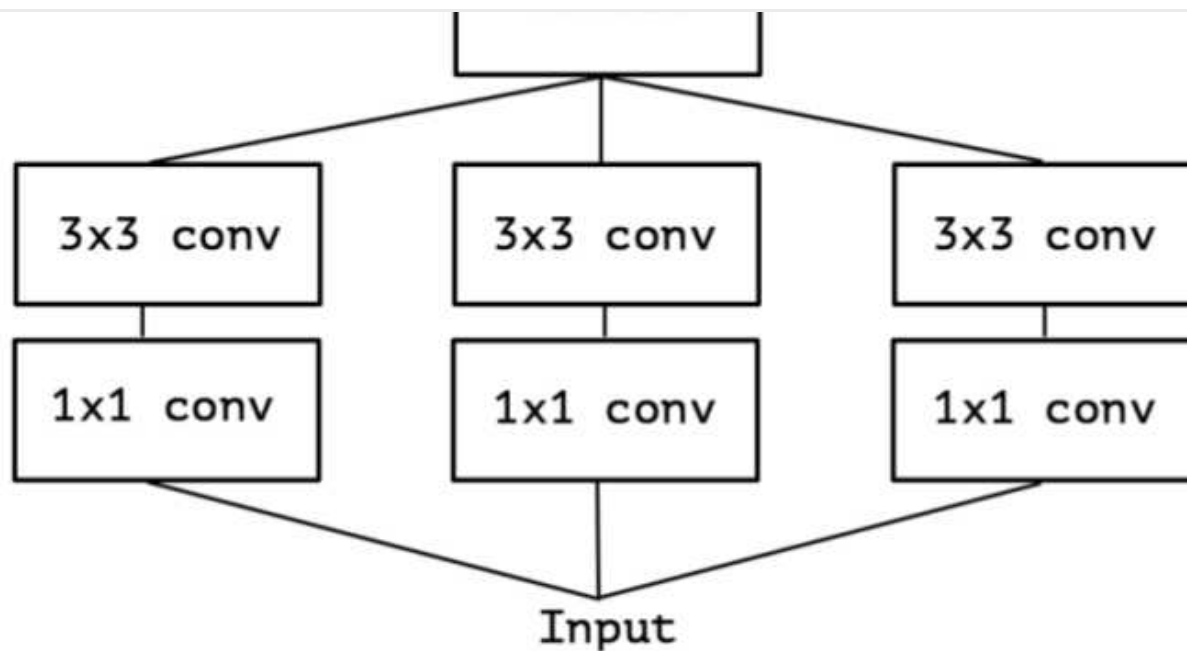
FCN ( [\[1411.4038\] Fully Convolutional Networks for Semantic Segmentation](#) ) : Fully convolutional networks , 顾名思义 , 整个网络就只有卷积组成 , 在语义分割的任务中 , 因为卷积输出的feature map是有spatial信息的 , 所以最后的全连接层全部替换成了卷积层。

Wavenet ( [WaveNet: A Generative Model for Raw Audio](#) ) : 用于语音合成。

## 4.深度可分离卷积

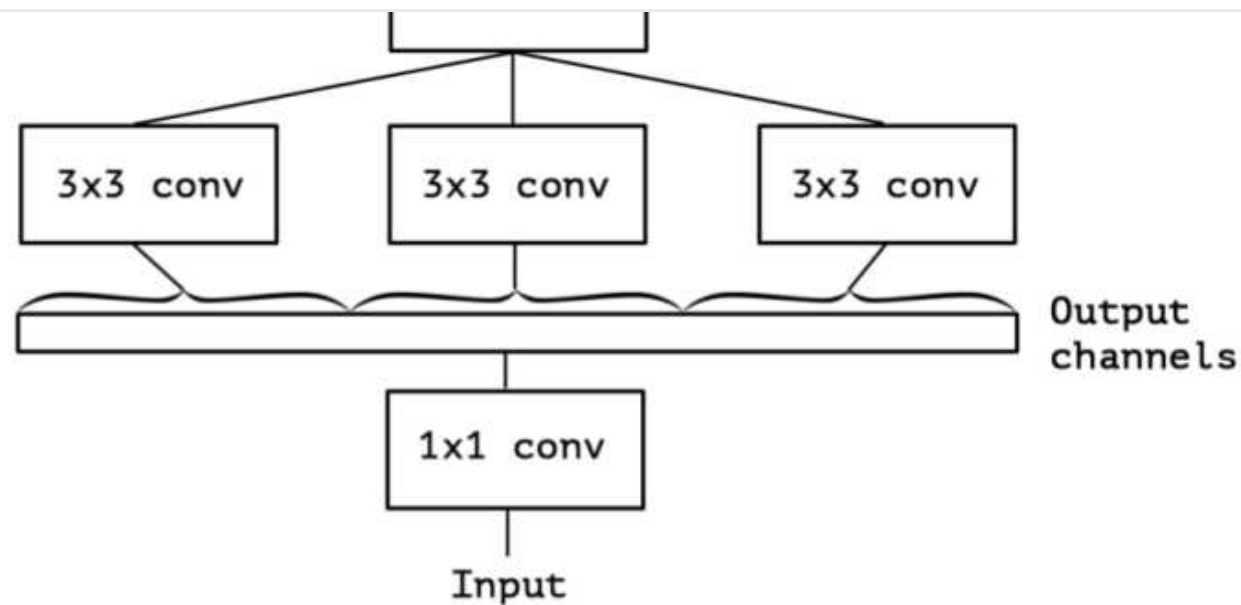
Depthwise Separable Convolution , 目前已被CVPR2017收录 , 这个工作可以说是Inception的延续 , 它是Inception结构的极限版本。

为了更好的解释 , 让我们重新回顾一下Inception结构 ( 简化版本 ) :

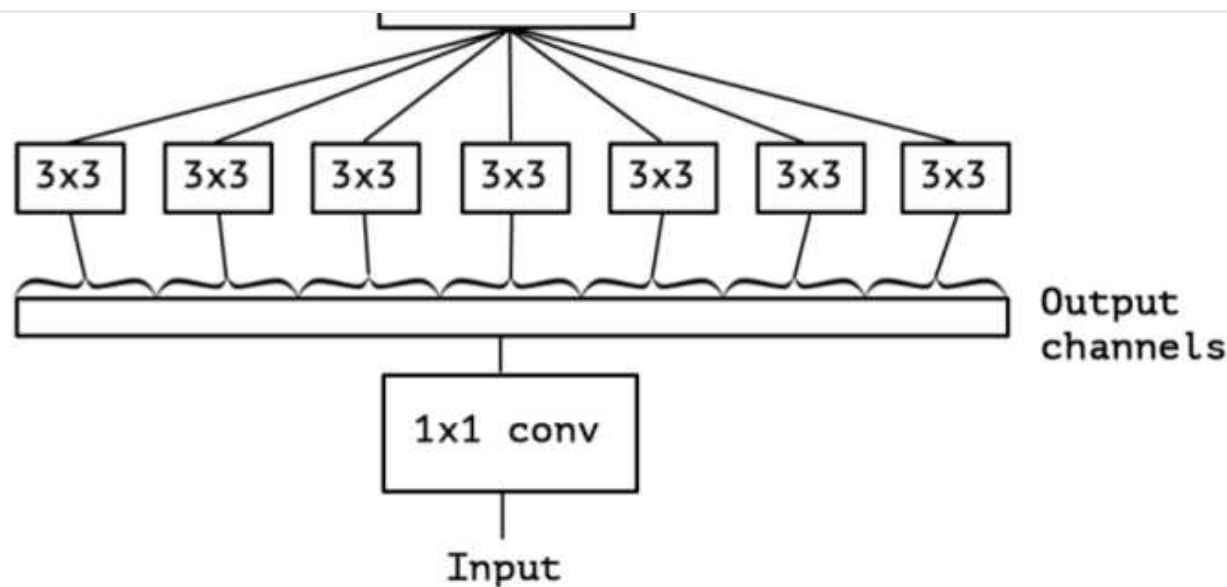


上面的简化版本，我们又可以看做，把一整个输入做1\*1卷积，然后切成三段，分别3\*3卷积后相连，如下图，这两个形式是等价的，即Inception的简化版本又可以用如下形式表达：





OK，现在我们想，如果不是分成三段，而是分成5段或者更多，那模型的表达能力是不是更强呢？于是我们就切更多段，切到不能再切了，正好是Output channels的数量（极限版本）：



于是，就有了深度卷积（depthwise convolution），深度卷积是对输入的每一个channel独立的用对应channel的所有卷积核去卷积，假设卷积核的shape是[filter\_height, filter\_width, in\_channels, channel\_multiplier]，那么每个in\_channel会输出channel\_multiplier那么多个通道，最后的feature map就会有in\_channels \* channel\_multiplier个通道了。反观普通的卷积，输出的feature map一般就只有channel\_multiplier那么多个通道。

具体的过程可参见我的demo：

【Tensorflow】tf.nn.depthwise\_conv2d如何实现深度卷积？ - CSDN博客

离的通道做了个融合。

这两个过程合起来，就称为Depthwise Separable Convolution了：

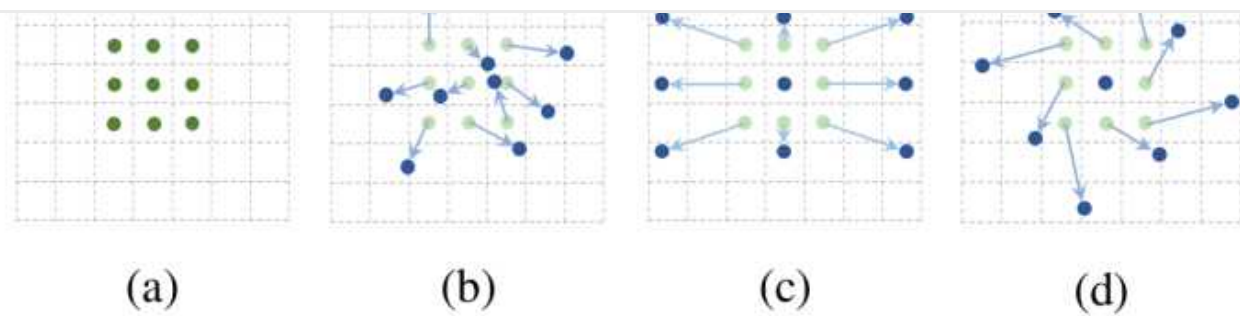
[【Tensorflow】tf.nn.separable\\_conv2d如何实现深度可分卷积? - CSDN博客](#)

代表模型：Xception ( [Xception: Deep Learning with Depthwise Separable Convolutions](#) )

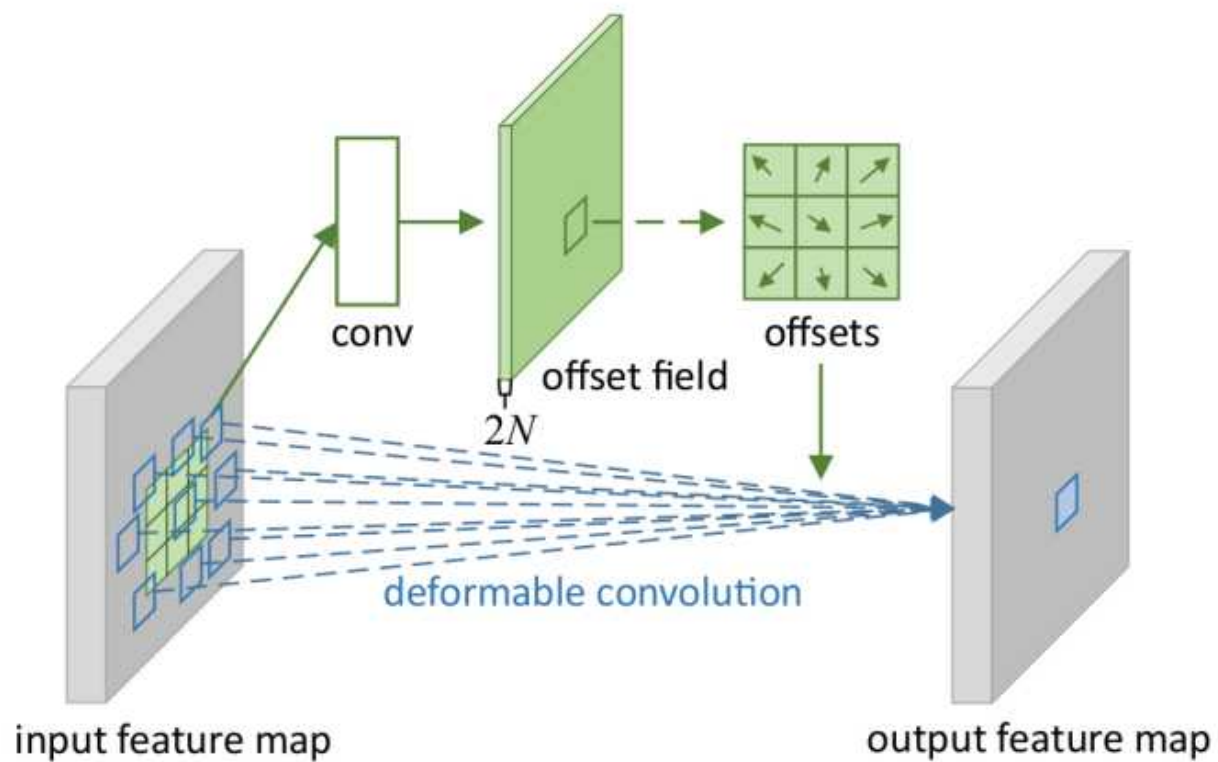
## 5.可变形卷积

可形变卷积的思想很巧妙：它认为规则形状的卷积核（比如一般用的正方形3\*3卷积）可能会限制特征的提取，如果赋予卷积核形变的特性，让网络根据label反传下来的误差自动的调整卷积核的形状，适应网络重点关注的感兴趣的区域，就可以提取更好的特征。

如下图：网络会根据原位置（a），学习一个offset偏移量，得到新的卷积核（b）（c）（d），那么一些特殊情况就会成为这个更泛化的模型的特例，例如图（c）表示从不同尺度物体的识别，图（d）表示旋转物体的识别。



这个idea的实现方法也很常规：



feature map的shape为[batch, height, width, channels]，我们指定输出通道变成两倍，卷积得到的offset field就是[batch, height, width, 2×channels]，为什么指定通道变成两倍呢？因为我们需要在这个offset field里面取一组卷积核的offsets，而一个offset肯定不能一个值就表示的，最少也要用两个值（x方向上的偏移和y方向上的偏移）所以，如果我们的卷积核是3\*3，那意味着我们需要3\*3个offsets，一共需要2\*3\*3个值，取完了这些值，就可以顺利使卷积核形变了。第二处就是使用变形的卷积核来卷积，这个比较常规。（这里还有一个用双线性插值的方法获取某一卷积形变后位置的输入的过程）

这里有一个介绍性的Slide：[prlab.tudelft.nl/sites/...](http://prlab.tudelft.nl/sites/...)

代表模型：Deformable Convolutional Networks（[Deformable Convolutional Networks](#)）：暂时还没有其他模型使用这种卷积，期待后续会有更多的工作把这个idea和其他视觉任务比如检测，跟踪相结合。

## 6.特征重标定卷积

这是ImageNet 2017 竞赛 Image Classification 任务的冠军模型SENet的核心模块，原文叫做“Squeeze-and-Excitation”，我结合我的理解暂且把这个卷积称作“特征重标定卷积”。

式来自动获取到每个特征通道的重要程度，然后依照计算出来的重要程度去提升有用的特征并抑制对当前任务用处不大的特征。

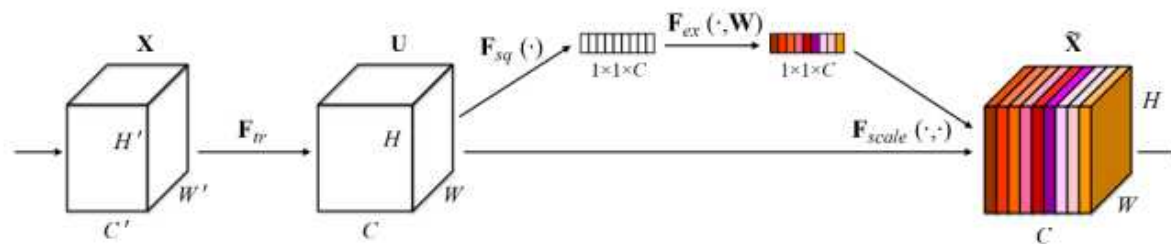


Figure 1. A Squeeze-and-Excitation block.

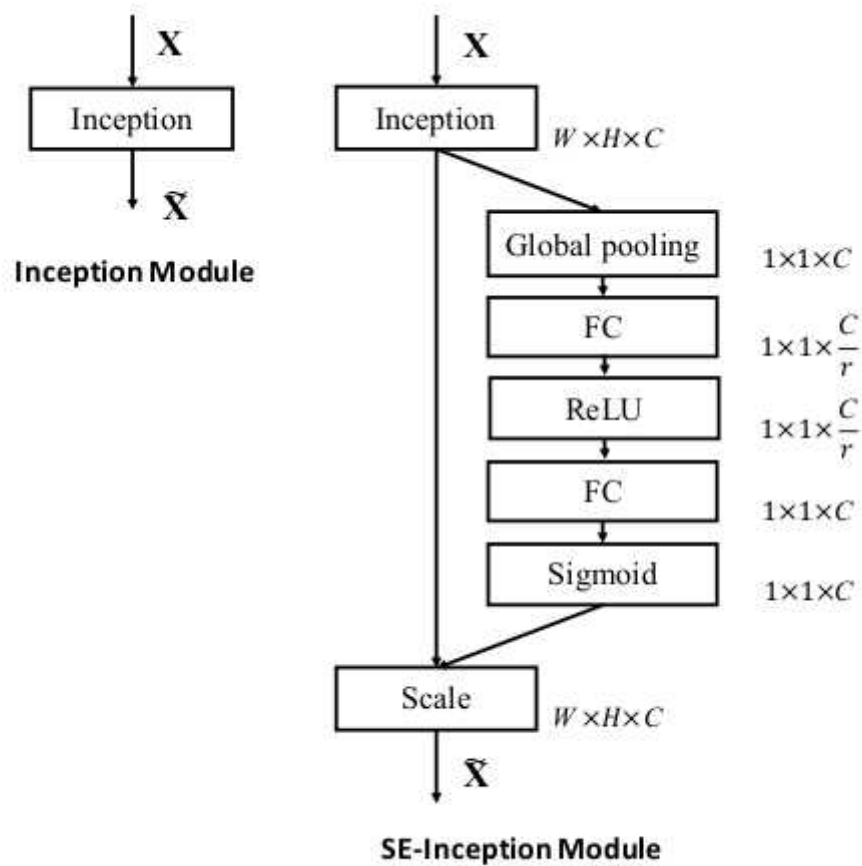
这个想法的实现异常的简单，简单到你难以置信。

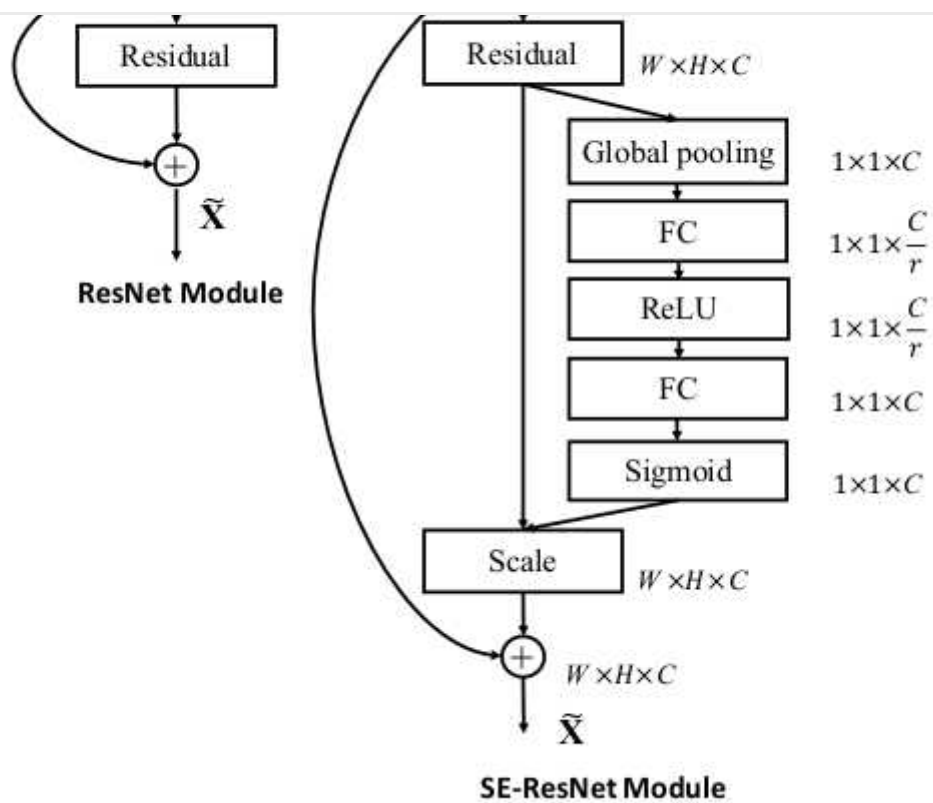
首先做普通的卷积，得到了一个的output feature map，它的shape为[C, H, W]，根据paper的观点，这个feature map的特征很混乱。然后为了获得重要性的评价指标，直接对这个feature map做一个Global Average Pooling，然后我们就得到了长度为C的向量。（这里还涉及到一个额外的东西，如果你了解卷积，你就会发现一旦某一特征经常被激活，那么Global Average Pooling计算出来的值会比较大，说明它对结果的影响也比较大，反之越小的值，对结果的影响就越小）

然后我们对这个向量加两个FC层，做非线性映射，这俩FC层的参数，也就是网络需要额外学习的参数。

虽然各大框架现在都还没有扩展这个卷积的api，但是我们实现它也就几行代码的事，可谓是简单且实用了。

另外它还可以和几个主流网络结构结合起来一起用，比如Inception和Res：





代表模型：Squeeze-and-Excitation Networks ( [Squeeze-and-Excitation Networks](#) )

## 7.比较

我们把图像（height，width）作为空间维度，把channels做为特征维度。



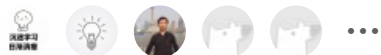
	小不变的情况下)		
多尺度非线性卷积	增加很多	多尺度（空间维度）	$\geq 2$
空洞卷积	不变	感受野（空间维度）	1
深度可分离卷积	减少	空间维度 / 特征维度	$\geq 2$
可变形卷积	增加	空间维度	$\geq 2$
特征重标定卷积	增加	特征维度	$\geq 2$

计算机视觉

深度学习（Deep Learning）

卷积神经网络（CNN）

知

 写文章[登录](#)

## 10 条评论



李鑫

Depthwise Separable Convolution就是卷积加group吧？

3 个月前



小赖sqLai

Depthwise Separable Convolution就是group Convolution吧。我记得最早的Lenet结构里就是这么实现的。

3 个月前




快使用双截棍巴拉

fcnn有用dilated conv？

3 个月前




本三吊打深度 回复 小赖sqLai

 查看对话



**Xf Mao (作者)** 回复 快使用双截棍巴拉


 查看对话

抱歉也有可能是我记岔了，是FCN还是deeplab还是UNet，反正是有一篇做语义分割用到了的

3 个月前



**Xf Mao (作者)** 回复 小赖sqLai


 查看对话

我觉得不太一样，group卷积是sparse connect 但是这里是dense connect

3 个月前



**Xf Mao (作者)** 回复 李鑫

 查看对话

我觉得不太一样，group卷积是sparse connect 但是这里是dense connect，这个问题还要再探究一下。

3 个月前



雨落晴空

骚套路~

3 个月前



清风笑

group conv是当这一层有多个一样结构的stream的时候这么做可以提高效率。depthwise

conv就是输入分成N段，每一段有 $(C2/N) \times H \times W \times (C1/N)$ 的kernel，然后把结果concat，这相当于把多个stream的输出concat。而depthwise separable是每一段有C2个 $H \times W \times (C1/C2)$ ，然后把整个channel范围的信息交换放到了后面的1X1卷积里实现。普通卷积前后都有1X1，inception（V1没有，后续版本有），resnet，densenet等都用了这种bottleneck结构，既先把input feature map过1X1把channel数降下来，然后卷积处理，然后再过1X1融合把channel数升回去。

3 个月前



徐斌

总结的很好，感谢分享，还可以加上每种卷积的效率什么的对比一下～

3 个月前