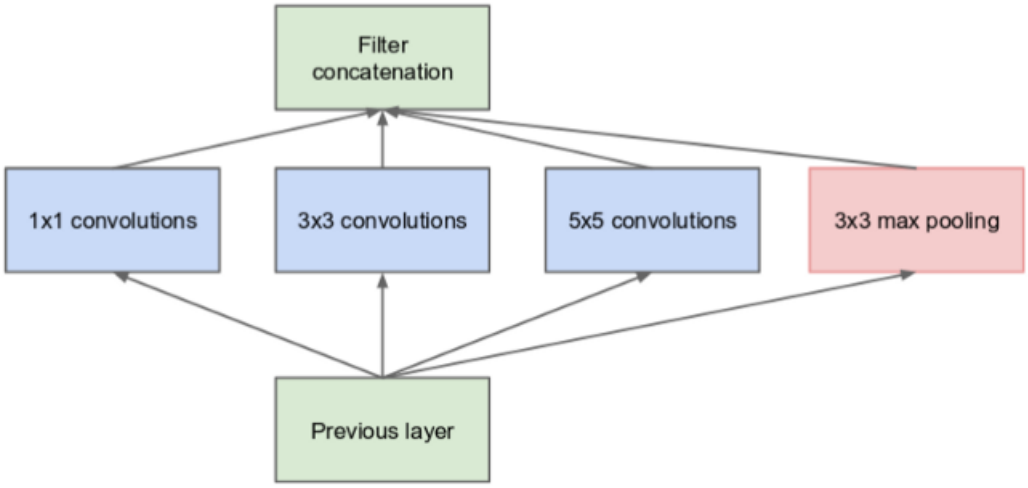
Xception

对于Xception的所有理解都应该建立在对**深度可分离卷积（Depthwise Separable Convolution）**以及对**Inception系列模型**的基本理解上。而实际上，Xception 取自 Extreme Inception，即表示 Xception 是一种极端的 Inception。

Inception基本信息

在Inception的基本模型中，其采取以下方式：



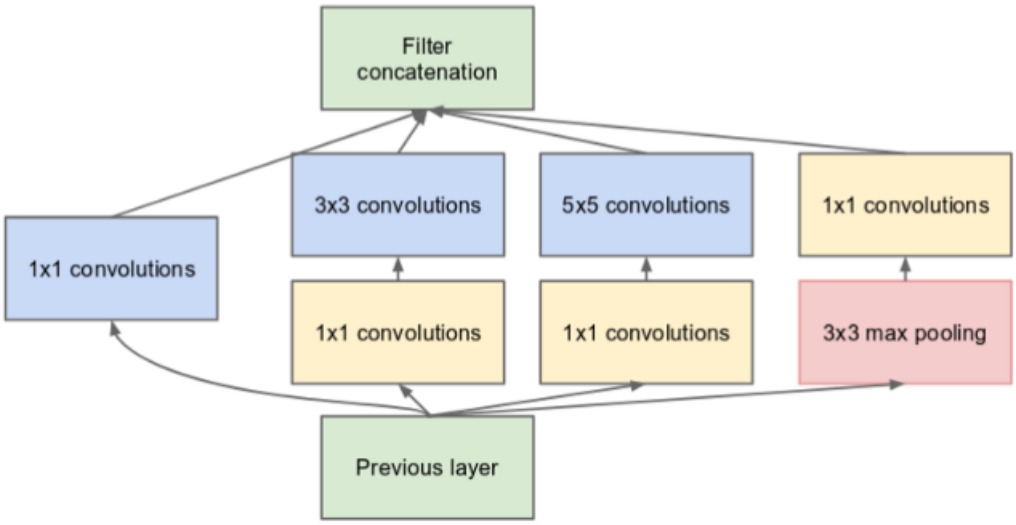
(a) Inception module, naïve version

Inception Module基本组成结构有四个成分。$1*1$卷积，$3*3$卷积，$5*5$卷积，$3*3$最大池化。最后对四个成分运算结果进行通道上的直接黏合，并输入至下一个Inception Module中。在这一模型中，**通过多个卷积核提取图像不同尺度的信息，最后进行融合，可以得到图像更好的表征。**

> Choosing the **right kernel size** for the convolution operation becomes tough. A **larger kernel** is preferred for information that is distributed more **globally**, and a **smaller kernel** is preferred for information that is distributed more **locally.**

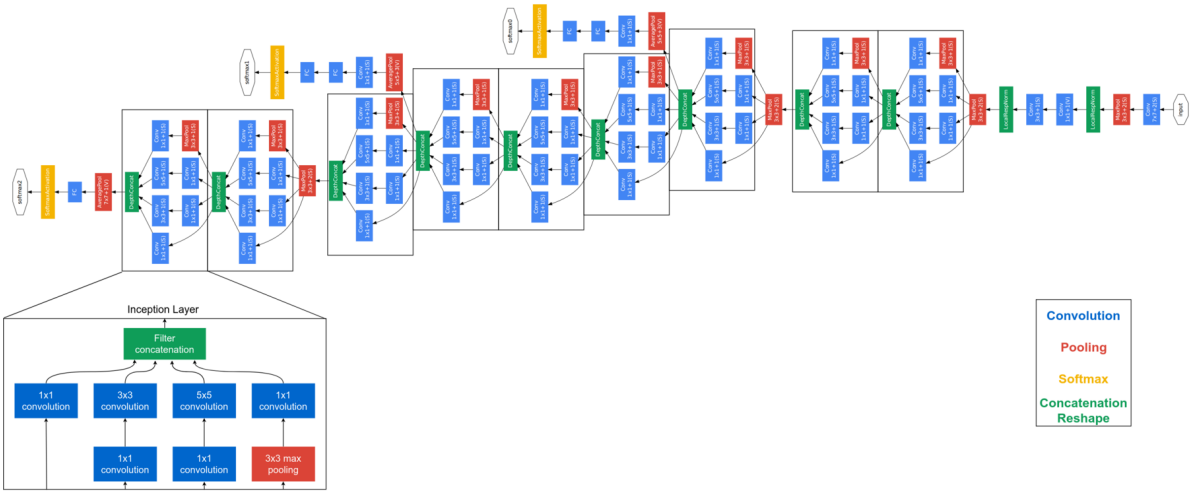
但在这一基本模型下，计算的开销会比较大，因此需要首先对输入进行一定处理来减小计算代价；

在这一过程中，Inception将会使用$1*1$的卷积核进行一次降维操作后再使用新的大卷积核进行操作：



(b) Inception module with dimension reductions

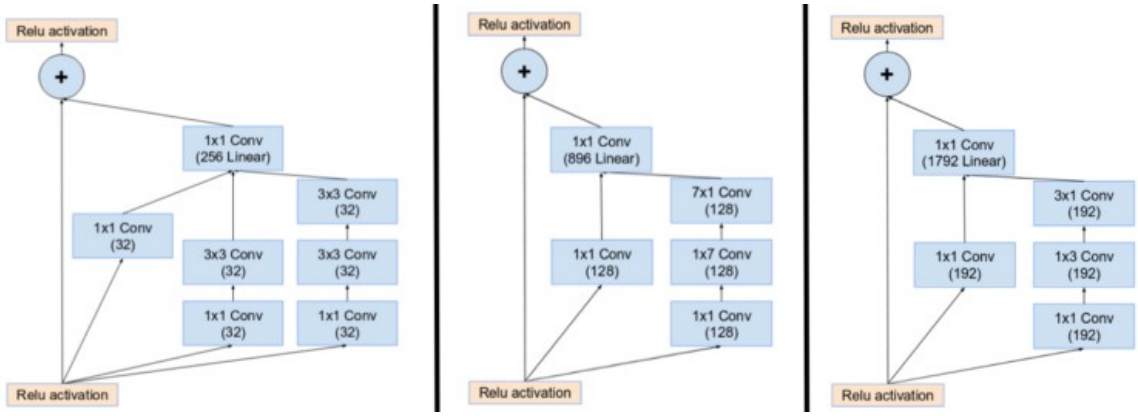
将多个Inception Module进行组合，即可以得到一个GoogLeNet，它的基本模型与信息如下：



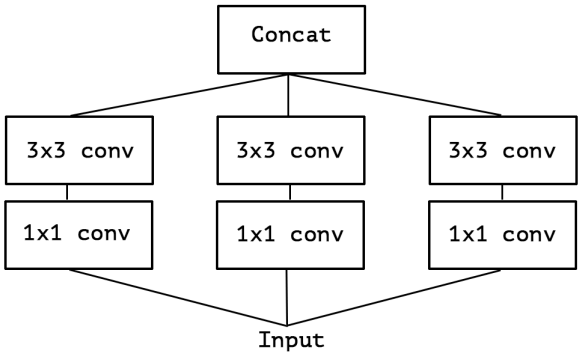
在这一模型中，一共会有9个Inception Module进行堆叠，基于此，其共有27个层，再最后一个层中，它会使用平均池化代替全连接层，这有利于提高finetune的效率。

也可以注意到：在模型中**另外增加了两个辅助的softmax分支**，其作用有两点，一是避免了梯度消失，用于向前传导梯度。反向传播时如果有一层求导为0，链式求导结果则为0。二是将中间某一层输出用作分类，起到模型融合作用。最后的$loss = loss_2 + 0.3 * loss_1 + 0.3 * loss_0$。实际测试时，这两个辅助softmax分支会被去掉。

在后续的版本中，Inception的改进主要集中于引入残差网络，以及拆分卷积核等操作；



基于此，可以认为Inception的基本Module概况如下：



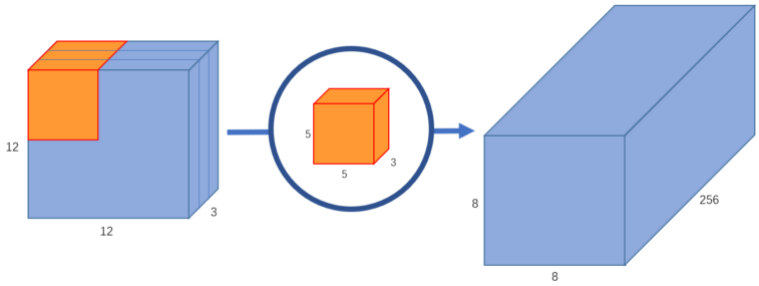
卷积运算

在实际的矩阵运算中，$1*1$矩阵总有着特殊的意义，即：**其主要目的是用于对输入的矩阵进行加权并对其进行一次维度的转化：**

> A $1*1$ kernel — or rather, n $1*1*m$ kernels where **n is the number of output channels** and **m is the number of input channels** — can be used outside of separable convolutions. **One obvious purpose of a 1x1 kernel is to increase or reduce the depth of an image** ..If you find that your convolution has too many or too little channels, a 1x1 kernel can help balance it out.

在这种情况下，某个卷积层应该有目标输出个大小为$1*1$，层数为输入维度的卷积核；这一性质实际上也是所谓**Depthwise Separable Convolutions**的基础：

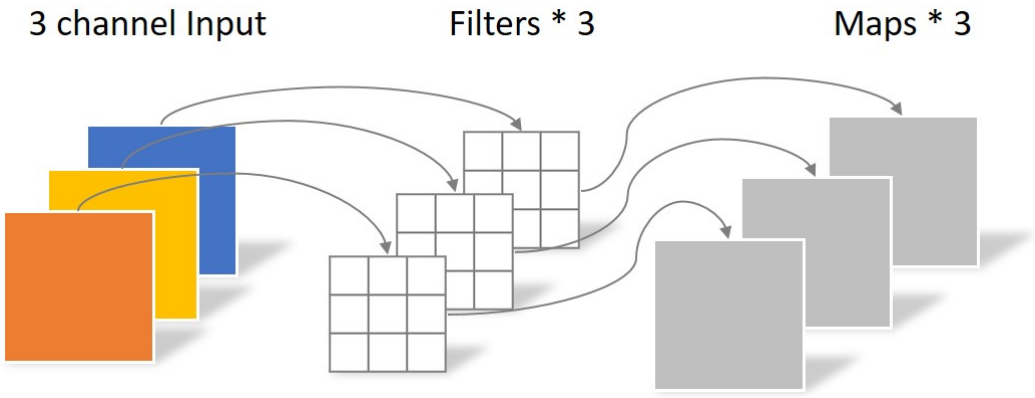
在一般的卷积操作中，将会使用卷积核与对应的输入层进行卷积后叠加得到一个channel为1的输出，在这一情况下，每一个卷积层的输出channel数目将会由该层的卷积核个数决定，即：



> we can create 256 kernels to create 256 8x8x1 images, then stack them up together to create a $8*8*256$ image output.

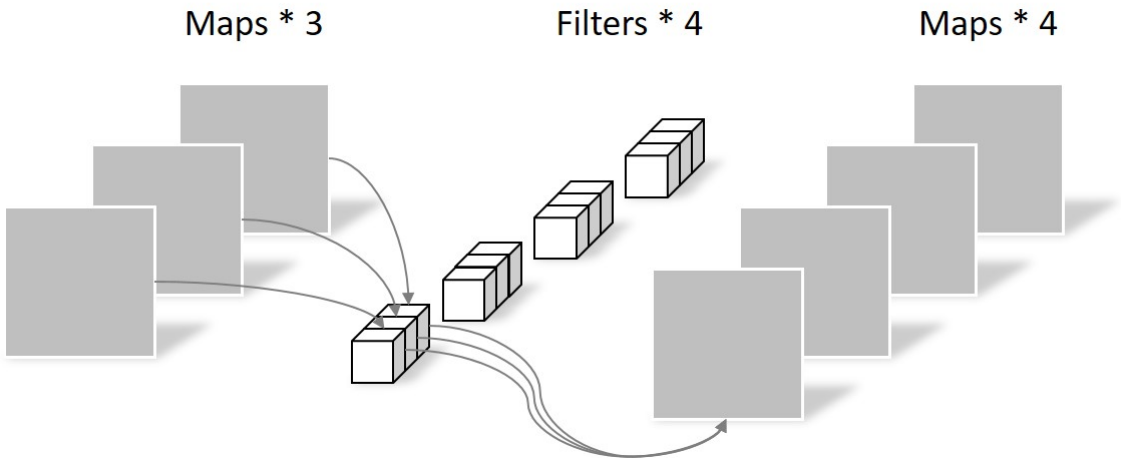
而**Separable Convolution**则希望将一个完整的卷积运算分解为两步进行，分别为`Depthwise Convolution`与`Pointwise Convolution`。

在`Depthwise Convolution`中，**卷积核的个数固定为输入的图像的层数，卷积核的channel固定为1；**即：



在这一情况下：`Depthwise Convolution`完成后的**Feature map数量与输入层的depth相同，但是这种运算对输入层的每个channel独立进行卷积运算后就结束了，没有有效的利用不同map在相同空间位置上的信息。因此需要增加另外一步操作来将这些map进行组合生成新的Feature map，**即接下来的`Pointwise Convolution`。

在`Pointwise Convolution`中，实际上就是利用了$1*1$矩阵的加权以及调节维度的性质，**这里的卷积运算会将上一步的map在深度方向上进行加权组合，生成新的Feature map。有几个Filter就有几个Feature map。**

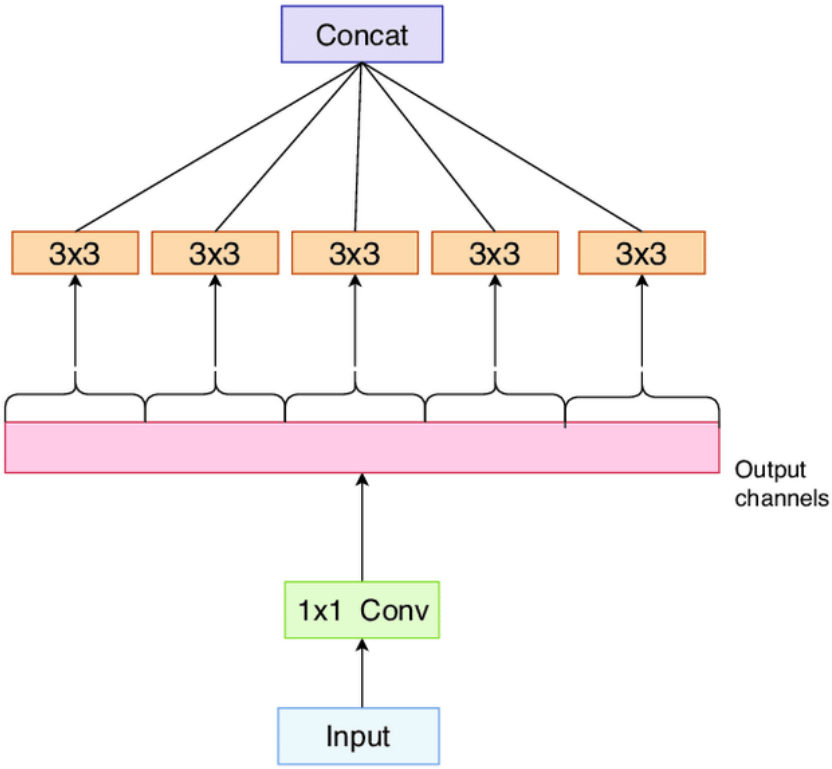


这样的计算减少了需要被训练的参数，因此提高了模型的一些性能。

Xception的基本性质

实际上，在Inception模组中，其认为可以通过一些跨通道的与一些空间的自相关关系可以足够地，分别地表示图像的基本信息而并不需要将它们连接在一起：

> In effect, the fundamental hypothesis behind Inception is that cross-channel correlations and spatial correlations are sufficiently decoupled that it is preferable not to map them jointly



基于此，Xception所提出的猜想即是：**首先使用一个$1*1$的卷积核对跨通道的各图像作出一次映射后，分别地对每一组输出的矩阵分别进行处理，最后使用一个卷积核分别在每一层输出上做卷积计算；**

这一种处理方式其实也是基于所谓`Depthwise Convolution`的一些基本处理思想，但其实际操作顺序实际上发生了变化：

> An “extreme” version of an Inception module, based on this stronger hypothesis, would first use a 1x1 convolution to map cross-channel correlations, and would then separately map the spatial correlations of every output channel.

> The order of the operations: depthwise separable convolutions as usually implemented (e.g. in TensorFlow) perform first channel-wise spatial convolution and then perform 1x1 convolution, whereas Inception performs the $1x1$ convolution first.

此外，还有一个细节，即：**在Inception Module中，对于经过$1*1$的卷积核计算得到的输出层，会在经过一个non-linearity层进行处理后输入大卷积核，而在Basic Xception Module中并没有进行这一项操作**，其原因也会在之后进行说明。

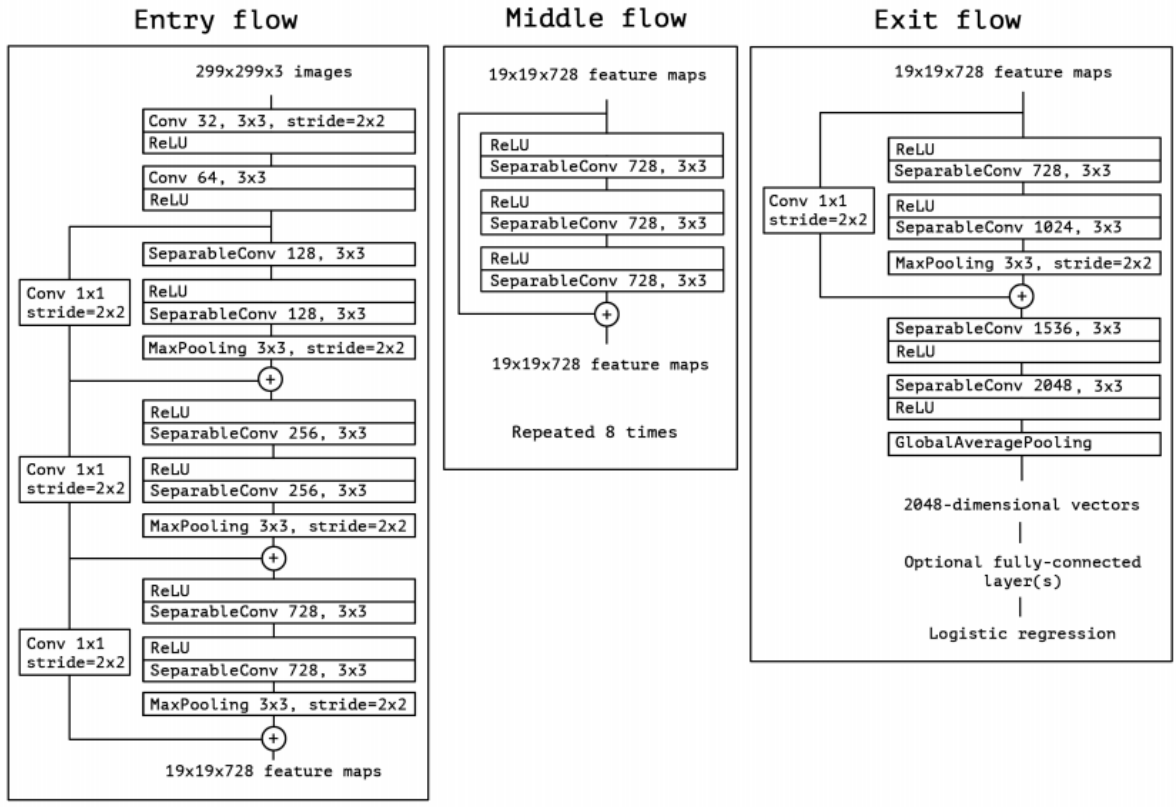
实际上，Xception的思想便来源于某一种折衷，即：一般的卷积操作会把所有的output（往往来自于单张输入）作为对象进行，而Inception实际上是将整个输入以三到四张为一组进行，而Xception实际上就是将每一个输出层作为一组进行卷积操作：

> A regular convolution (preceded by a $1x1$ convolution), at one extreme of this spectrum, corresponds to the single-segment case; a depthwise separable convolution corresponds to the other extreme where there is one segment per channel; Inception modules lie in between, dividing a few hundreds of channels into 3 or 4 segments.

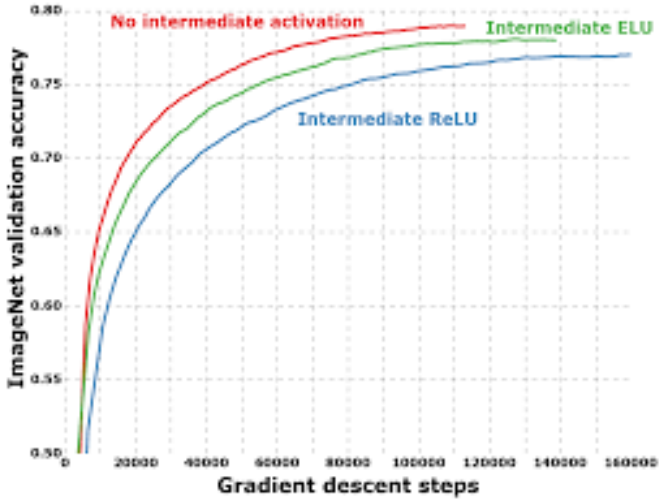
Xception的结构的大致情况如下：

- 用36个卷积层的14个模组完成特征提取操作；
- 模组内部使用残差网络进行连接；
- 可以选择使用全连接层或者逻辑回归层完成最终的输出；

此外，也使用了`Dropout`层与$L2$正则化操作进行了优化。



最后应该回到在是否需要在进行$1*1$卷积后加入一个非线性层进行处理的话题；其得到的基本结果如下：



可以发现，若不使用非线性层，其在ImageNet数据集上将取得更好的效果；这与Inception上的实验结果不同，因此，作者猜想，这是由于非线性层更易于在较深的特征空间中取得好的效果，而在单层的空间中难以实现优化。

总结

在Xception中，提出了将`Depthwise separable convolutions`代替常规的卷积操作作用于图像处理的猜想，实际上，作者也指出，相比于常规的操作，这并非一种优化而更类似于某一种转角，对如何分组进行卷积提出了新的方向：**It may be that intermediate points on the spectrum, lying between regular Inception modules and depthwise separable convolutions, hold further advantages.**

Intro to Inception