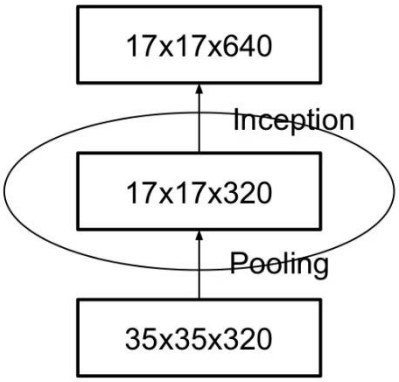
背景

提出了四个通用的网络设计准则和优化思路：

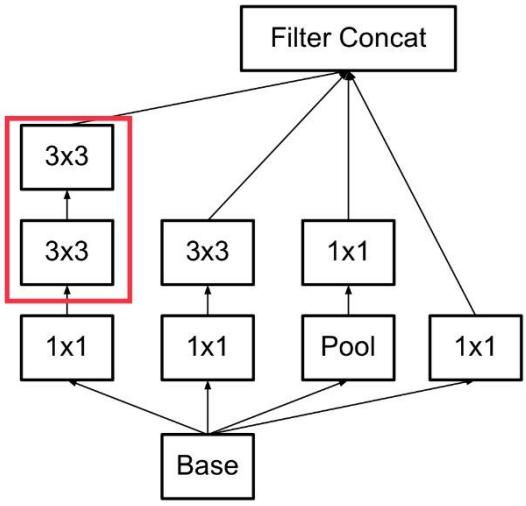
①从输入到输出，特征图的尺寸应该是逐渐变小的，不然会引起特征表达瓶颈；



如上图所示的结构就背离了第一原则。

②高维信息更容易被网络处理。增加卷积后的激活函数的次数，得到高维稀疏特征；

③可以先将具有大感受野的将卷积映射到低维空间，不会引起很多，甚至不会引起任何在特征表达能力上的损失；



如上图利用两个 3×3 卷积代替一个 5×5 的卷积。

④均衡考虑网络的深度和宽度。

方法

①分解大卷积核

邻近的激活区域是高度关联的，意味着输出的特征图里有冗余的部分，因此考虑将大卷积特征聚合之前，将其降维处理，且保留非线性激活函数，增加非线性的同时也提高了特征表达能力。

将大卷积分解减少了计算量。假设输入特征图尺寸为(C,H,W)，卷积个数都为 c 个经过一个 5×5 的卷积，计算量为：

$$H\times W\times C\times 5\times 5\times c=25HWCc$$

经过两个 3×3 卷积，计算量为：

$$(H\times W\times C\times 3\times 3\times c)\times 2=18HWCc$$

②非对称分解

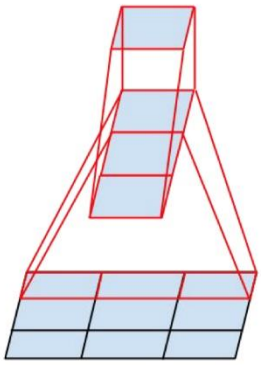


Figure 3. Mini-network replacing the 3×3 convolutions. The lower layer of this network consists of a 3×1 convolution with 3 output units.

如上图所示，将一个 3×3 卷积核分解为一个 3×1 卷积和一个 1×3 卷积。

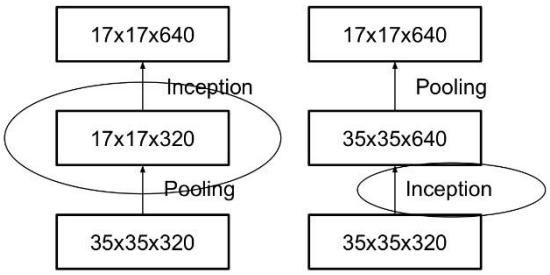
更通俗地来讲可以将一个 n×n 的卷积分解为一个 n×1 卷积和一个 1×n 卷积。在特征尺寸介于 12~20 之间使用这种卷积分解效果最好。

减少了计算量而且 n×1 卷积和一个 1×n 卷积之间的非线性激活函数增加了网络的表征能力。

③辅助分类器

在 Inception_v1 里提到的辅助分类器可以使模型收敛地更快，其实通过实验发现用不用辅助分类器差别不大，只是使用辅助分类器的精度会相对高一些。而且去掉浅层的辅助分类器也不会有什么副作用。如果辅助分类器中使用了 BN 或者 dropout 等正则化方法，得到的效果更好。

④高效降低尺寸（使特征图变小变厚）

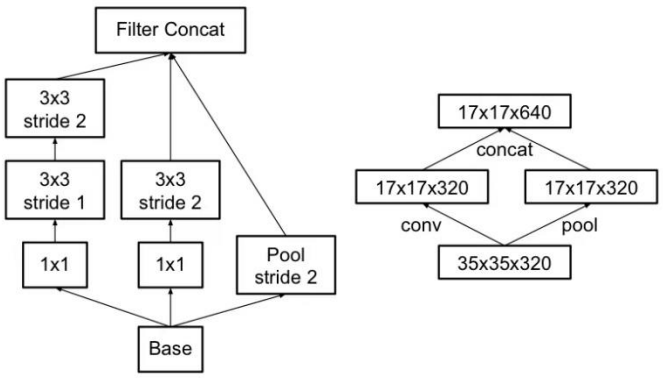


上图中，左图利用步长为 2 的卷积下采样，造成信息丢失，带来特征表达瓶颈问题，而右图存在的问题是计算量大。

假设输入特征图尺寸为(d,d,k)，输出的特征图尺寸为$(\frac{d}{2},\frac{d}{2},2k)$，如果先升

维(卷积核为 n，卷积核个数为 2k)再池化，其计算量有：

$$n\times n\times k\times d\times d\times 2k$$



采用并行的方式解决上述问题，左边两路卷积操作，右边一路池化操作，即避免了特征瓶颈又减少了计算量。

⑤标签平滑操作

通常的分类任务是将图像的真实标签值转换成 one-hot 编码，然后根据 softmax 值计算交叉熵，优化任务就是使得交叉熵函数最小。这样存在一个问题，就是当 softmax 得到的置信度越大时，交叉熵越小，因此最小交叉熵会使得正确类别的分数趋于无穷，使得模型对于预测结果过于自信，泛化能力差。

因此引入了标签平滑参数 ε 以及标签先验分布 $u(k)$ 来解决这些问题。

原先的标签函数分布：

$$q(k)=\begin{cases}1 & k=y\\ 0 & k\neq y\end{cases}$$

可以写作 $q(k)=\delta_{k,y}$

标签平滑后变为：

$$q'(k)=(1-\varepsilon)\delta_{k,y}+\frac{\varepsilon}{K}$$

交叉熵可以表示为：

$$H(q',p)=-\sum_{k=1}^{K}\log p(k)q'(k)=(1-\epsilon)H(q,p)+\epsilon H(u,p)$$

$H(q,p)$ 为预测值与真实标签值的交叉熵，$H(u,p)$ 为预测值与先验分布的交叉熵。相当于在真是标签上增加噪声，让预测值不要过度集中于概率较高的类别

Inception_v2 指的是使用了新提出的技术中的一种或多种的 Inception 模块。而 Inception_v3 指的是这些技术全用了的 Inception 模块

总结

相比于 Inception_v1 的结构针对于不同深度设计了不同的结构，不是像 v1 简单地重复相同的模块结构。也解决了 v1 里存在的特征瓶颈问题，以更柔和的

方式让特征图的变小变厚。

Inception 结构感觉就是在保证表达能力的同时减少参数量。将原本的乘积操作以并行的方式分散为加和操作，以后自己在设计网络时，如果参数量大的话，可以考虑利用这种并行的方式减少计算量。