

5.Inception V2 V3

2015年由google团队提出，论文地址[Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift](#),
[Rethinking the Inception Architecture for Computer Vision](#)

问题：representatial bottleneck，特征描述瓶颈是指中间某层对特征空间维度进行较大比例的压缩（比如使用pooling），**导致很多特征丢失**，虽然pooling是CNN结构中必须的功能，但我们可以通过一些优化方法来减少pooling造成的损失。特征数目越多收敛的越快。

一些特征的结构

1.卷积分解

将 5×5 的卷积分解成2个 3×3 的卷积运算以提升计算速度。

在本文中，作者将 $n \times n$ 的卷积核尺寸分解为 $1 \times n$ 和 $n \times 1$ 两个卷积。

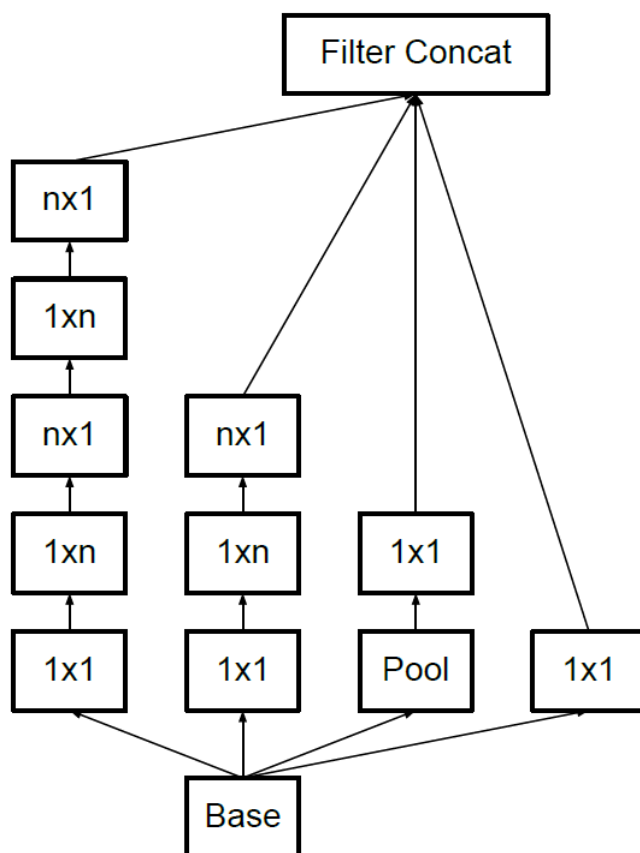


图8 卷积分解

模块中的滤波器组被扩展（变得更宽而不是更深），以解决特征瓶颈。如果该模块没有被扩展宽度，而是变得更深，那么维度会过多减少，造成信息损失。

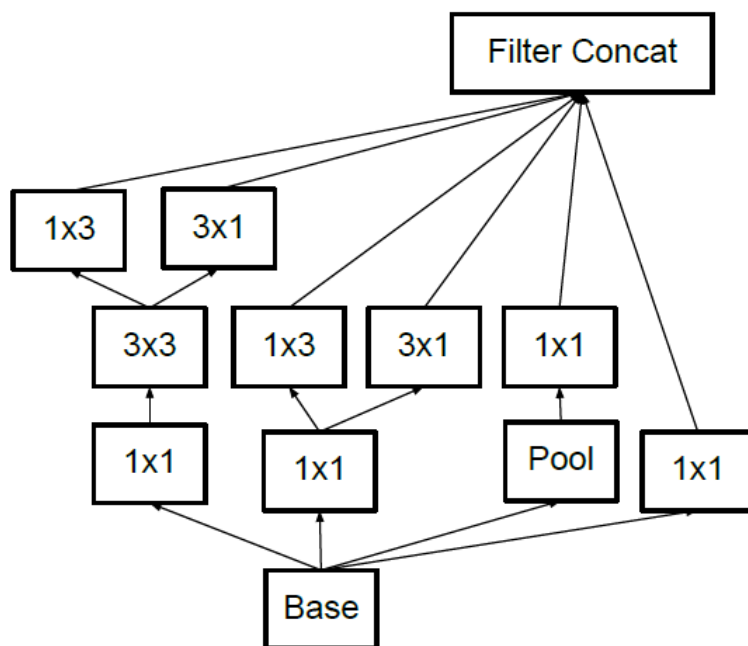


图9 特征扩展

2.辅助分类器的效用

辅助分类器在训练前期并没有起什么作用，到了训练的后期才开始在精度上超过没有辅助分类器的网络，并达到稍微高的平稳期。并且，在去除这两个辅助分类器后并没有不利的影响，因此在Inception V1中提到的帮助低层网络更快的训练是有问题的。如果这两个分支有BN或Dropout住分类器的效果会更好，这是BN可充当正则化器的一个微弱证据。

3.高效降低Grid Size

有两种方式：（1）先池化再Inception，（2）先Inception再池化

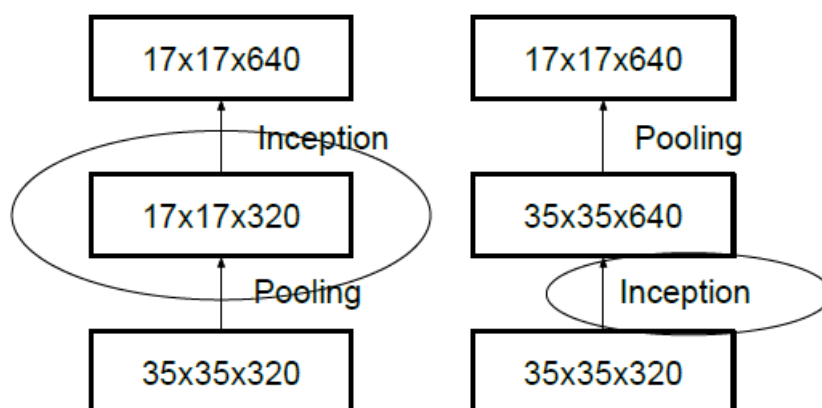


图10 降低Grid Size

左边特征先减少的话容易造成特征丢失，即bottleneck问题；右边会极大增加计算量

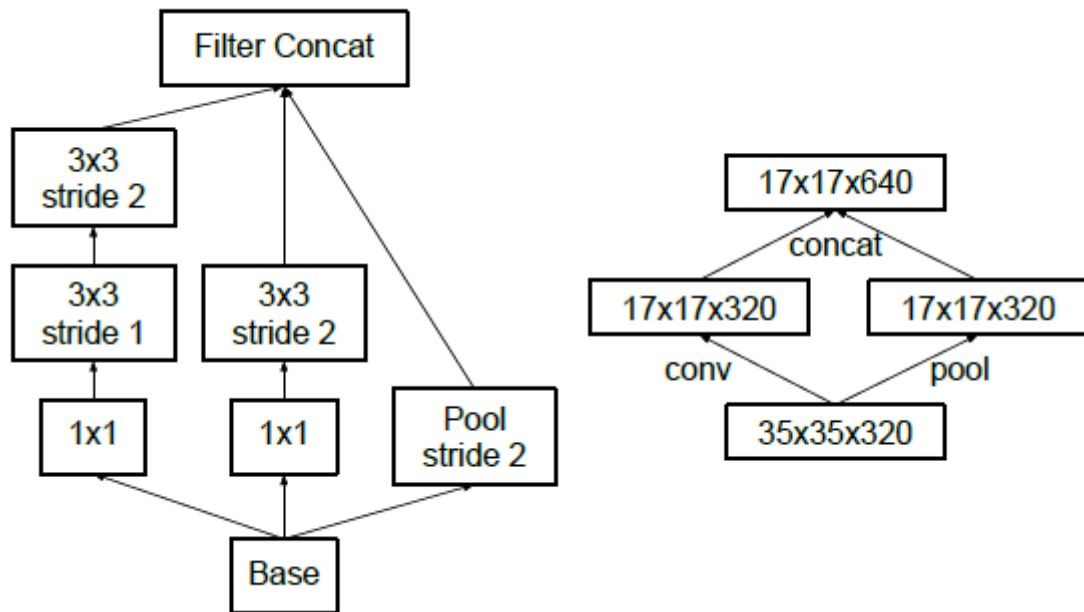


图11 降低Grid Size

并行操作，一边减小特征图尺寸，一边增加特征图数量

辅助分类器：

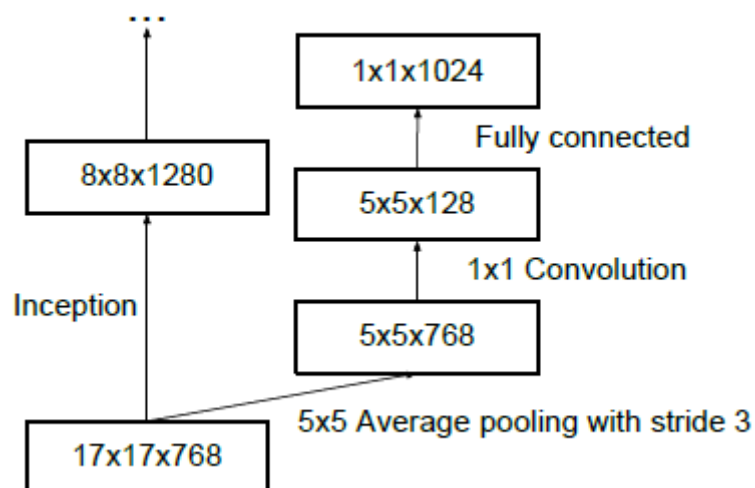


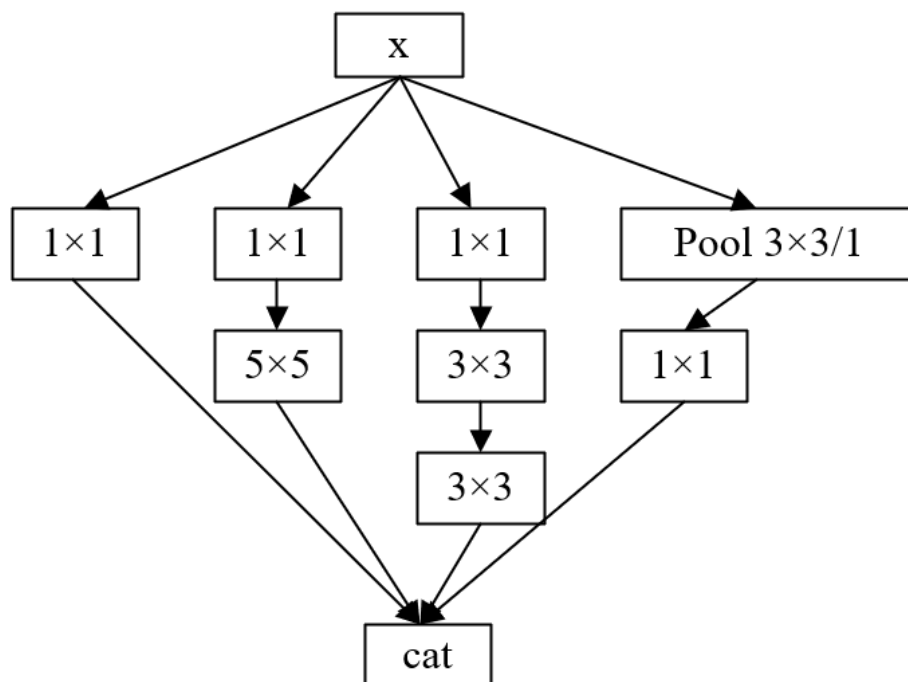
图12 辅助分类器

type	patch size/stride or remarks	input size
conv	$3 \times 3 / 2$	$299 \times 299 \times 3$
conv	$3 \times 3 / 1$	$149 \times 149 \times 32$
conv padded	$3 \times 3 / 1$	$147 \times 147 \times 32$
pool	$3 \times 3 / 2$	$147 \times 147 \times 64$
conv	$3 \times 3 / 1$	$73 \times 73 \times 64$
conv	$3 \times 3 / 2$	$71 \times 71 \times 80$
conv	$3 \times 3 / 1$	$35 \times 35 \times 192$
$3 \times$ Inception	As in figure 5	$35 \times 35 \times 288$
$5 \times$ Inception	As in figure 6	$17 \times 17 \times 768$
$2 \times$ Inception	As in figure 7	$8 \times 8 \times 1280$
pool	8×8	$8 \times 8 \times 2048$
linear	logits	$1 \times 1 \times 2048$
softmax	classifier	$1 \times 1 \times 1000$

图13 Inception V2 V3 结构

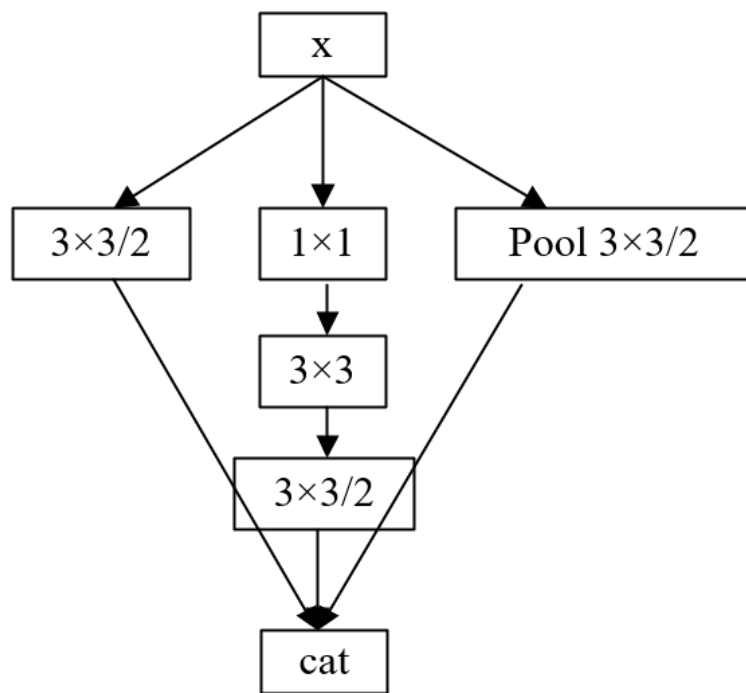
[pytorch实现](#)

Inception模块



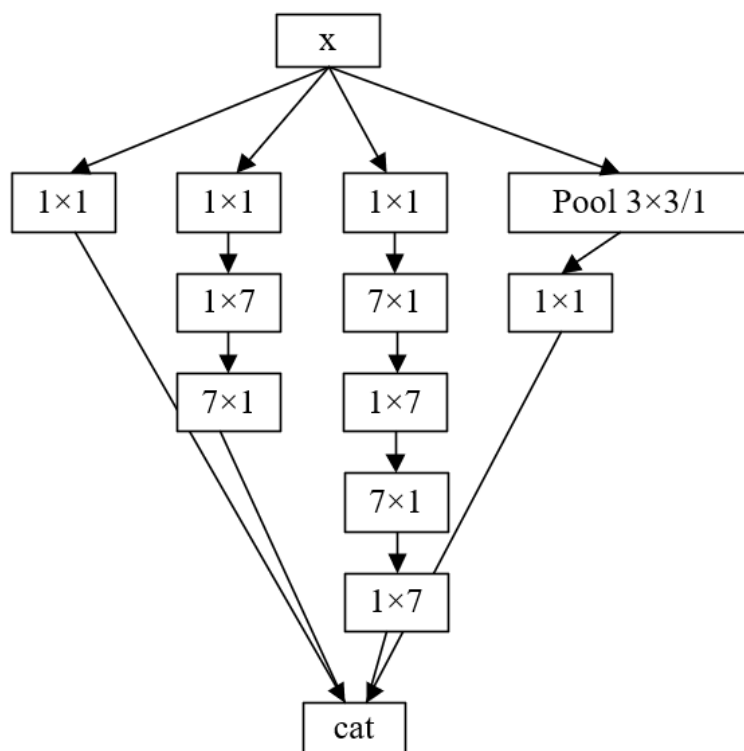
InceptionA

图14 InceptionA



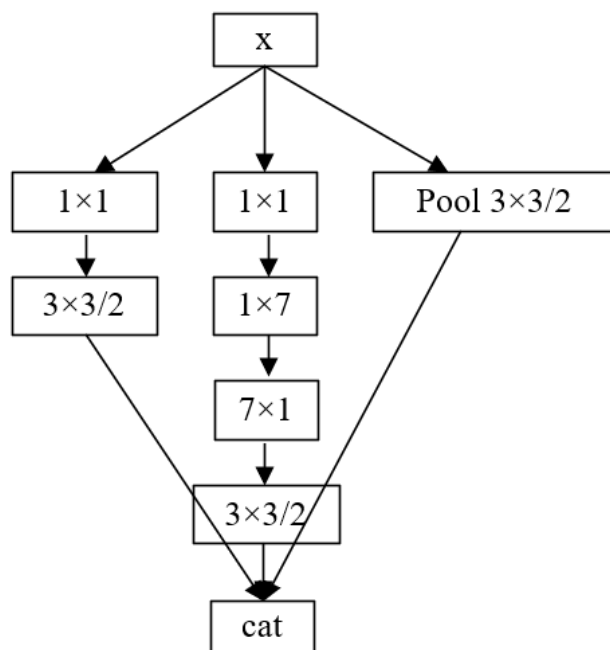
InceptionB

图15 InceptionB



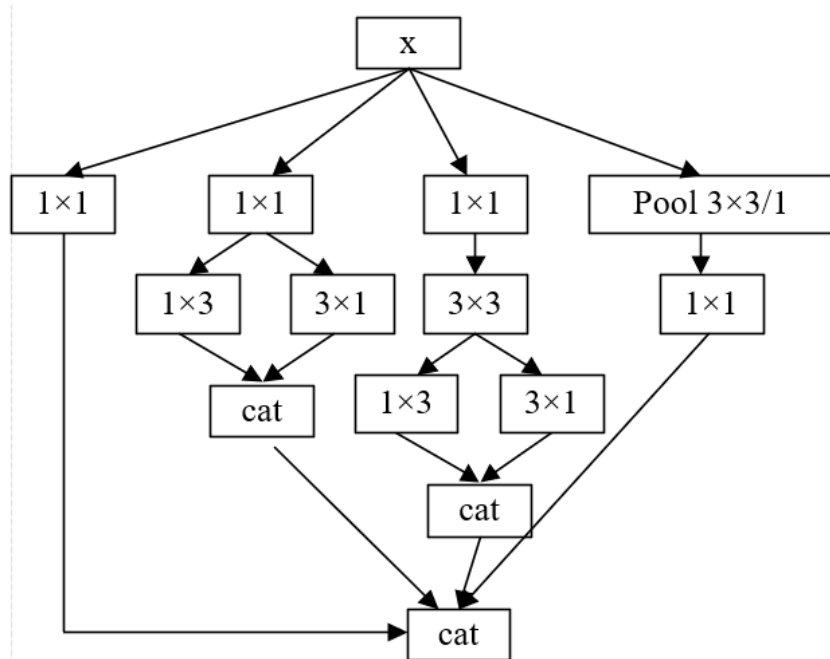
InceptionC

图16 InceptionC



InceptionD

图17 InceptionD



InceptionE

图18 InceptionE