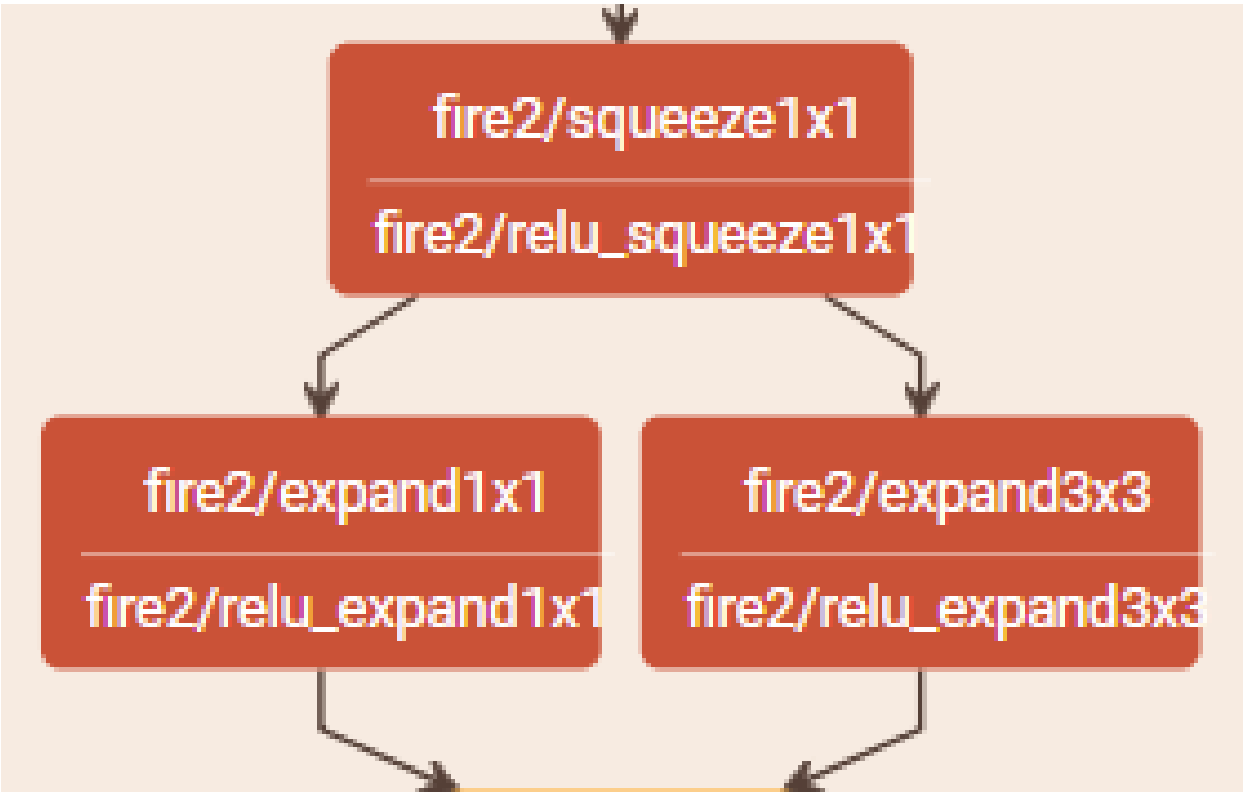
大网络虽然精度高，但是体积太大，不利于部署移动端。于是出现了一些性能好、精度高的轻量级网络。

一、SqueezeNet



SqueezeNet的特点就是先squeeze，再expand。

即先降低channel数量，再分两路扩大channel数量，最后进行concat拼接。

模型大小不到5M。

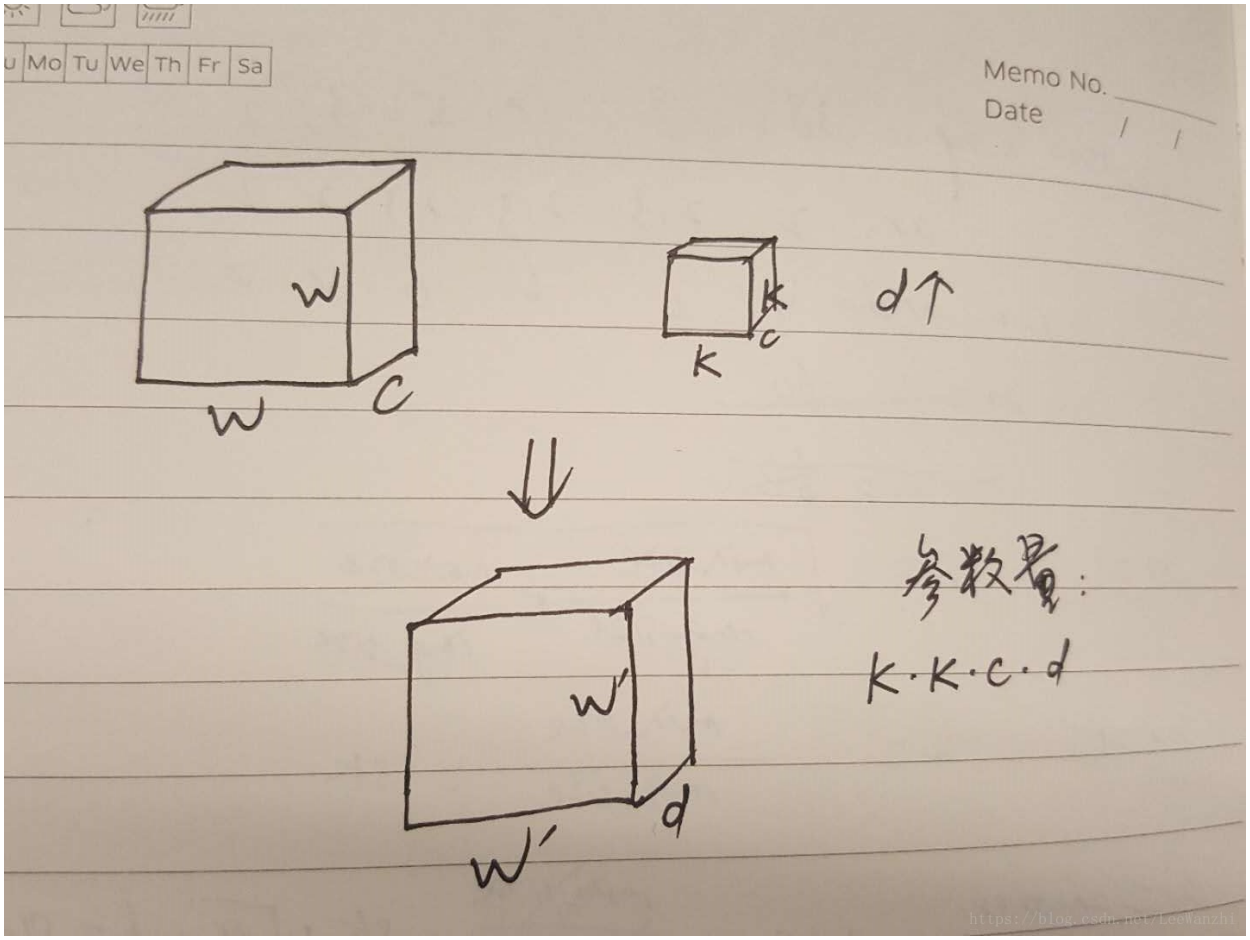
我将SqueezeNet作为base_network用到faster rcnn上，检测一下自己的任务，最终实现的mAP仅达到88.2%。(VGG(47M)，ResNet(68M)分别为90.7%、90.53%)。但胜在模型只有4.2M。

二、MobileNet_v2

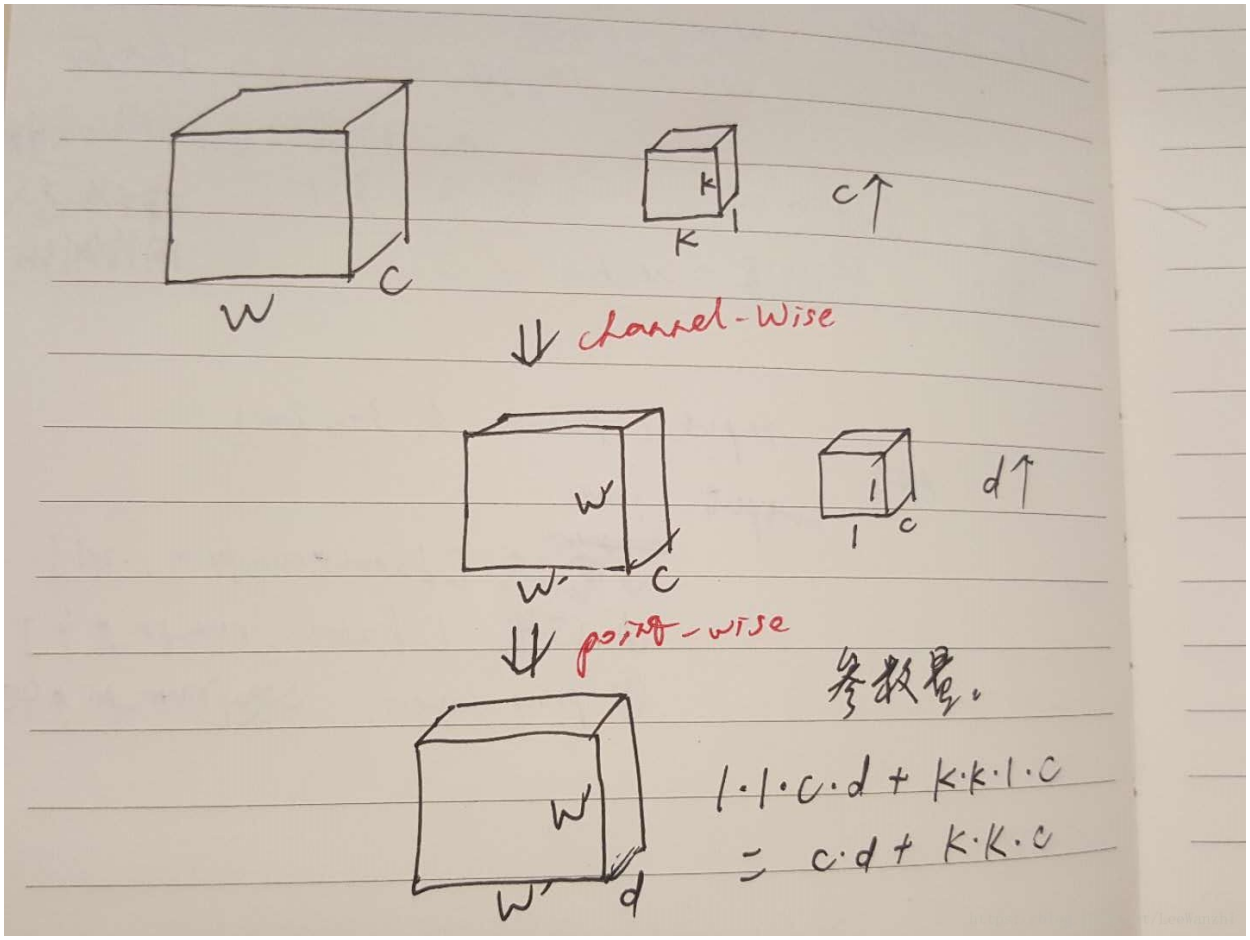
首先说下v1：

v1的核心是对通道进行**group**操作，也称为**depth_wise**或 **channel_wise**。所以caffe中实现，既可以用group，也可以自定义一个layer来实现。

先来看传统的conv是怎么操作的，传统的conv操作如下:



v1的操作则不同:



v1涉及到了group卷积。

group卷积是：比如32个通道，我分成2组，则将每一个卷积核**按照通道**分为2组(每组的通道数=32/2= 16)，第1组卷积核对第1组通道进行卷积 ，第2组卷积核对第2组通道进行卷积。然后concat。

通过上图可以看出，**v1采用的group数量，等于上一层feature map的通道数C，也就是分成C组。**

这一步我们称为depth_wise或者channel_wise。

但这也存在一个问题，就是单个通道与单个通道之间的信息不流通。

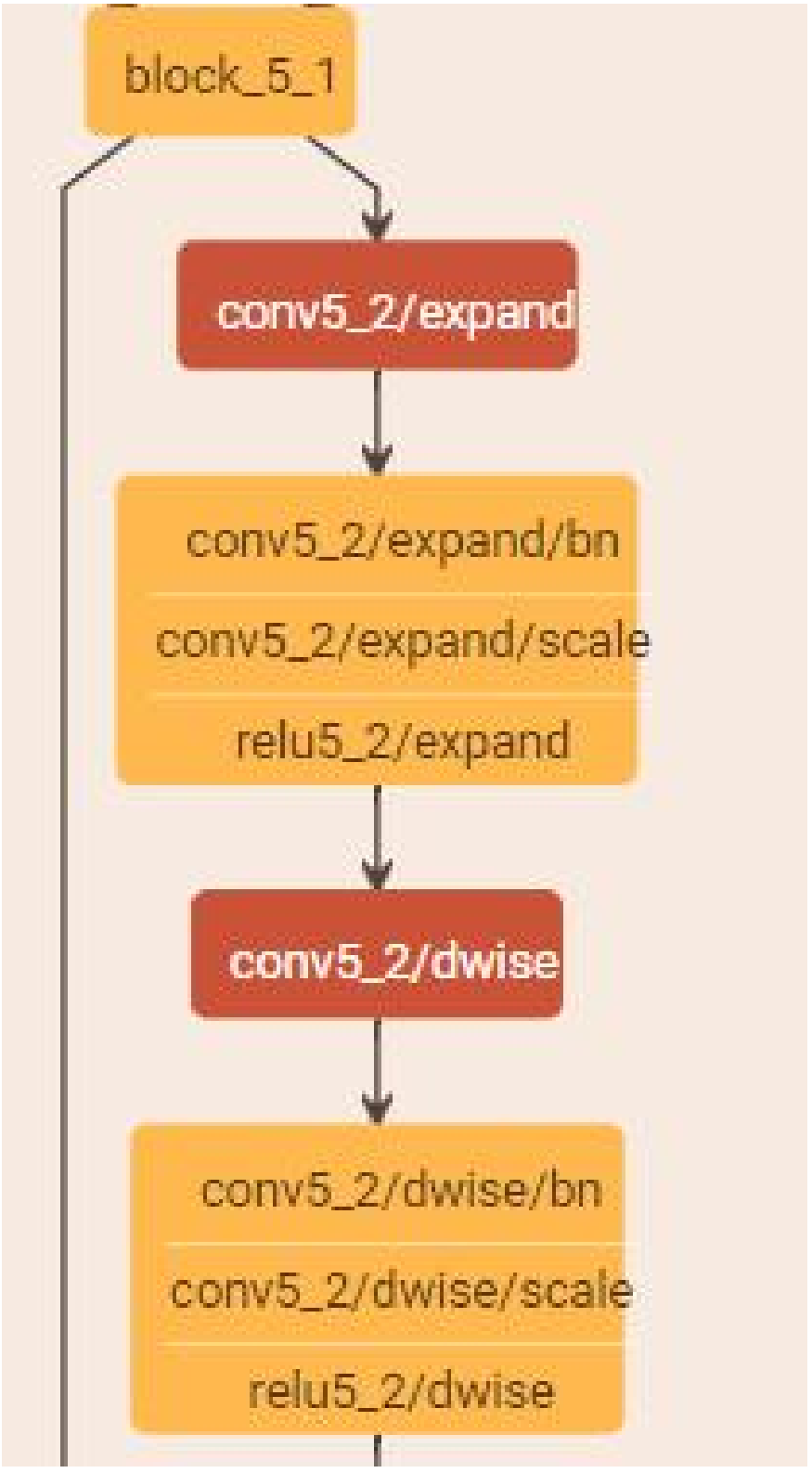
于是，进行depth_wise后，还需接上一个point_wise，即用一个1x1的卷积核正常的卷积一遍，这样就可以把通道的信息联系起来。

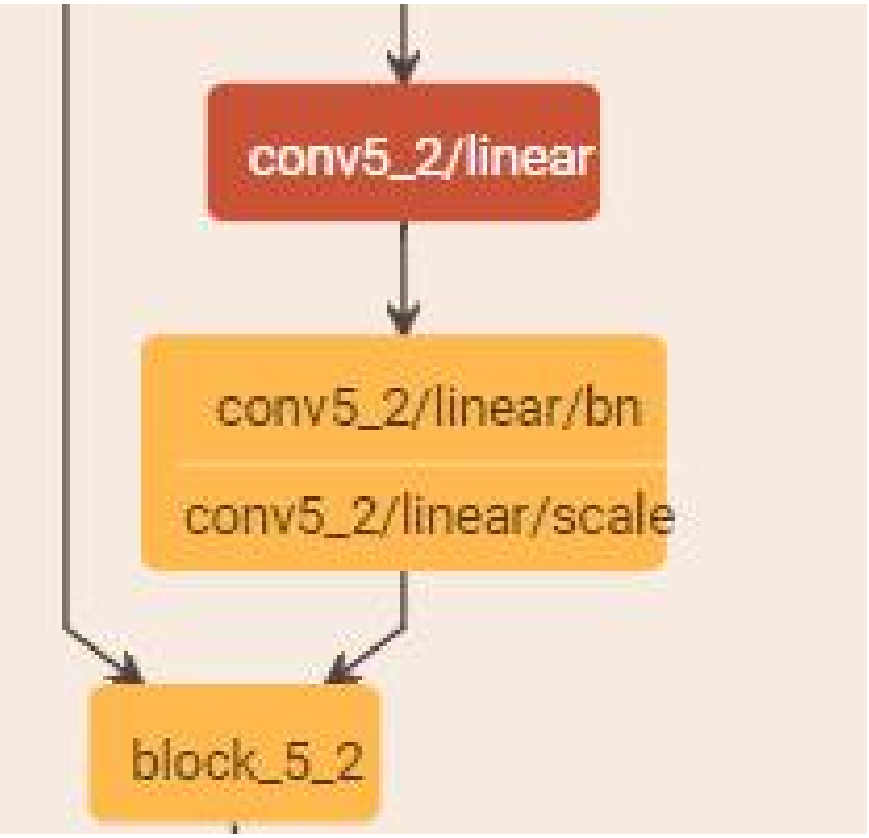
然后说v2：

v2有两大特点：

(1) Inverted residuals：

我们正常的残差块，都是先降通道数，再升，最后进行element_wise操作。v2则是先升后降，因为如果要做depth_wise，通道被压缩后提取的特征也会变少，所以先expand，再depth_wise。





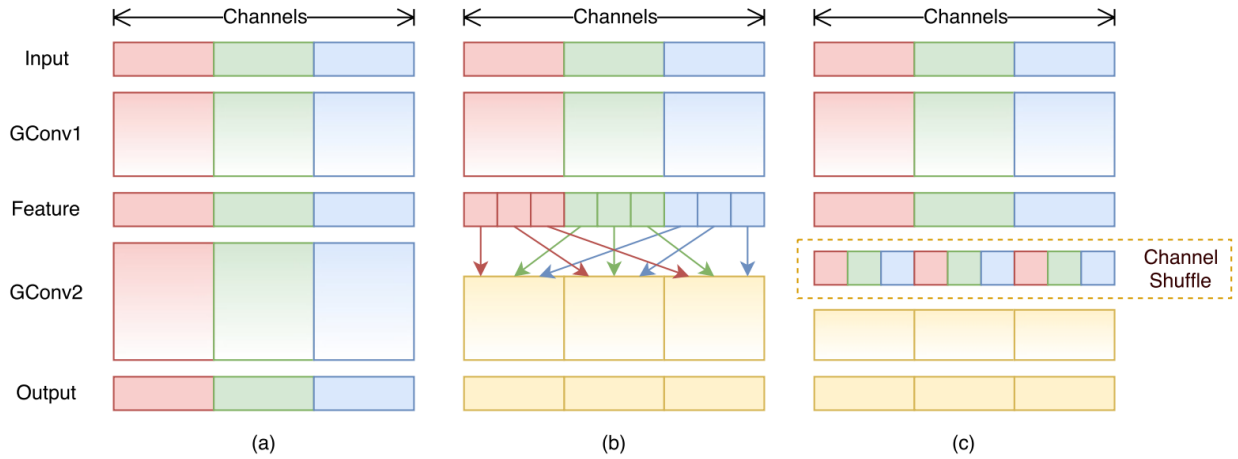
(2)linear bottlenecks： 由于relu会将小于0的数置为0，但这些信息也许有用，所以在point_wise操作后，弃用非线性的激活函数。因此这块point_wise操作称为linear，好记。

虽然mobileNet_v2作为轻量化网络，不过，我将mobileNet_v2作为faster rcnn的base_network训练时，一张12G的显卡竟然带不动。。。。

三、ShuffleNet_v2

先说v1

这里我引用某位博主的一张图[1]:



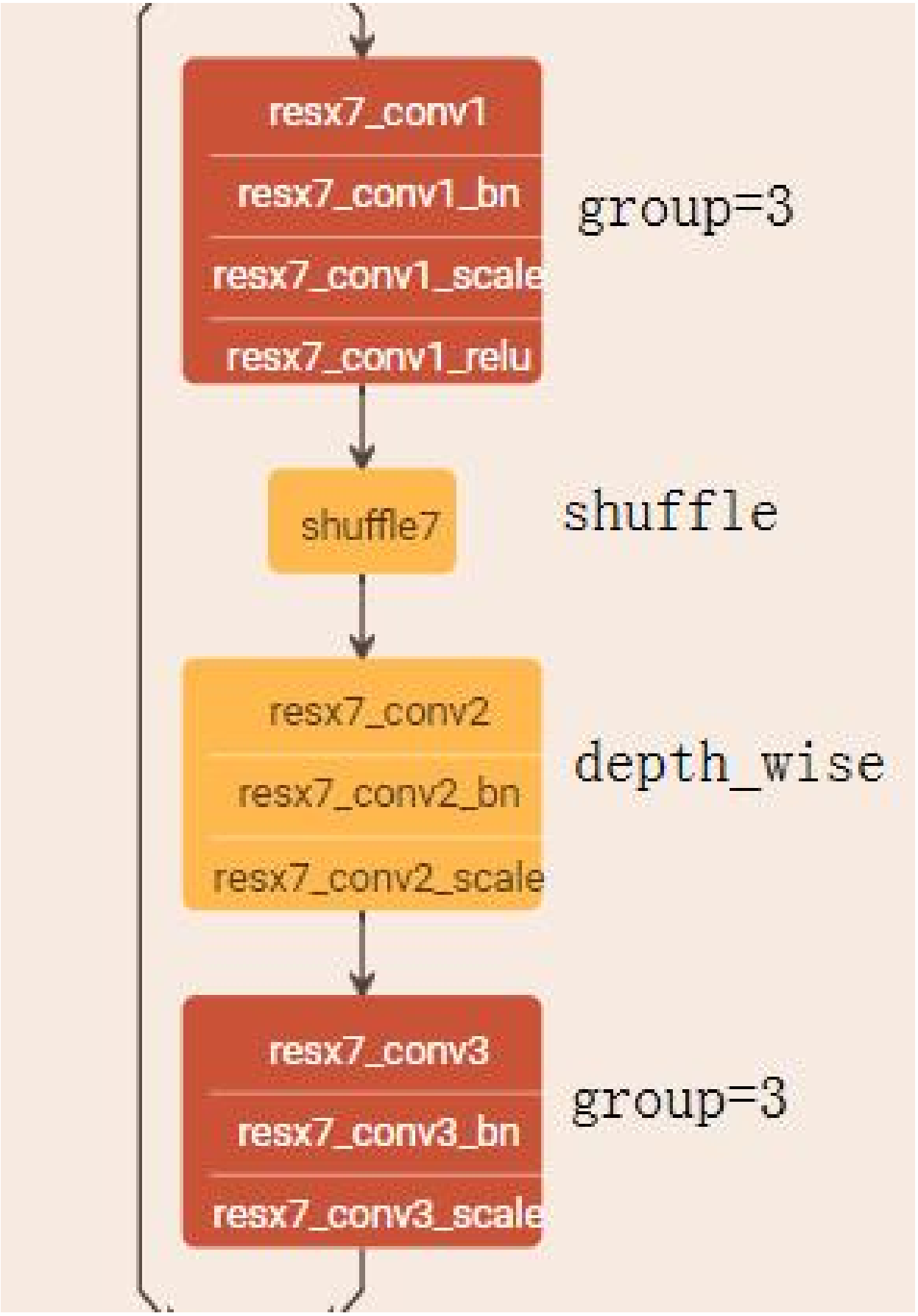
图(a)为group操作，图(b)为shuffleNet，图©为shuffleNet详细描述。

group操作我在mobileNet中已经说了，但group会造成通道间信息不流通，mobileNet给出的方案的接一层point_wise。

shuffleNet给出的方案则是打乱通道，并均匀重组。

比如我有32个通道，group=2，我将每个group再分成2份，这样有4份(1，2，3，4)，然后重组，1和3为第一个group，2和4为第2个group。然后对重组后的feature map进行depth_wise卷积。

下图为shuffleNet中的某一块，不仅用到了group，还用到了shuffle，还用到了group的特殊情况depth_wise。



然后说v2

论文中针对高性能网络提出了4点准则：很重要:

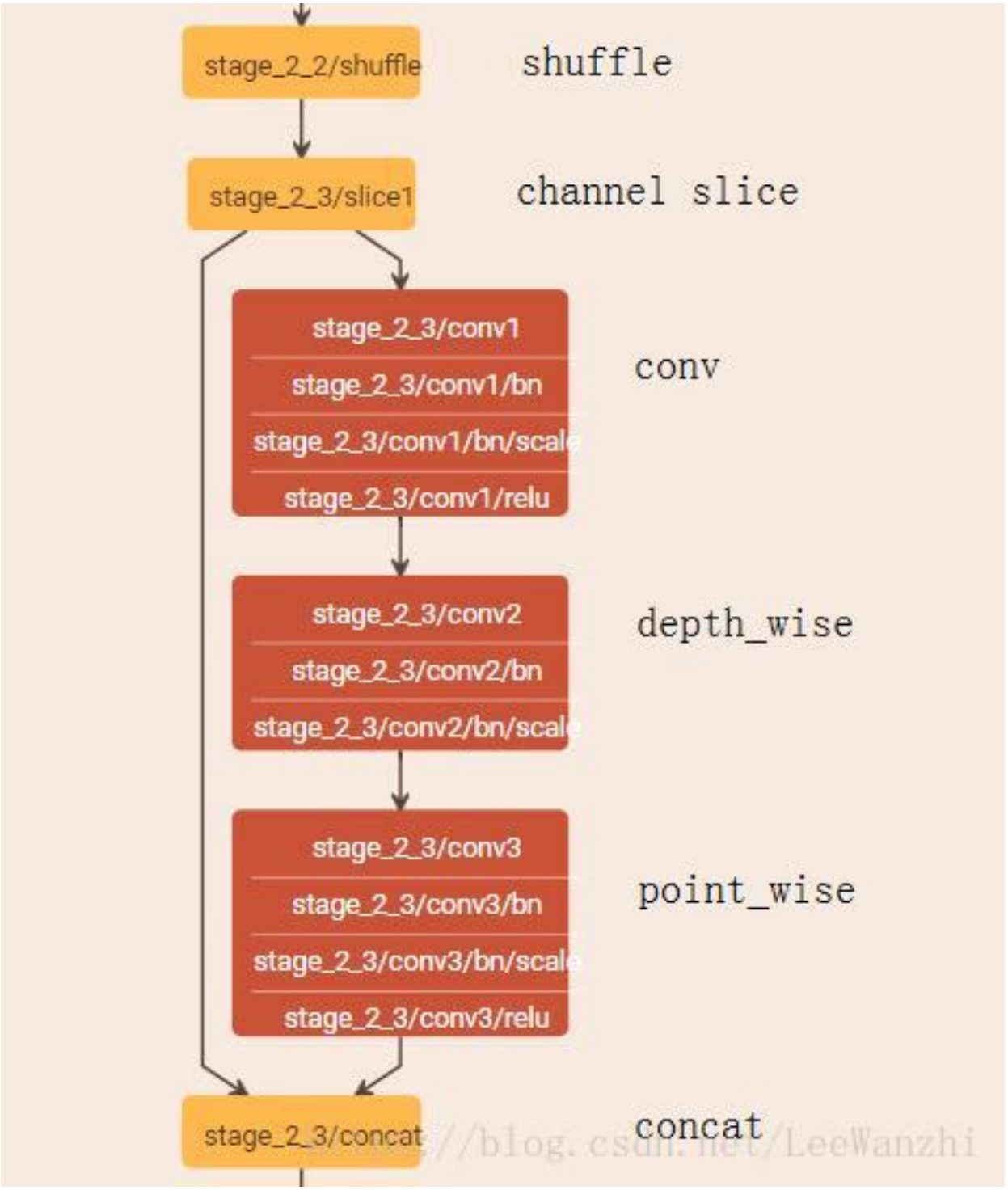
(1)使用相同的通道宽度的卷积

(2) **不要过度使用组卷积，**这会增加内存访问成本。

(3)**降低碎片化程度**((比如Inception中的多路径)

(4)减少元素级运算(比如element wise add)

再来看看网络结构：



shuffleNet_v2用到了**通道分割**:: 我们shuffle后，将channel一分为二，一部分不动(**准则3**)，另一部分用于卷积。

为了满足**准则(1)**,涉及到的卷积都是通道不变化的。首先是普通的conv，然后接一个depth_wise，point_size。

这里只进行了一次group conv(**准则2**)。最后concat，而不是eltwise(**准则4**)

我也用shuffleNet v2作为faster rcnn的base_network做了一波检测，可惜mAP只达到80.61%，且测试时单张图片880ms左右(VGG 200ms，mAP=90.7%)，不知道是不是我自己的问题。。。

四、Xception

Xception不是严格意义上的轻量化网络，它只是对Inception v3的改进。

Figure 3. A strictly equivalent reformulation of the simplified Inception module.

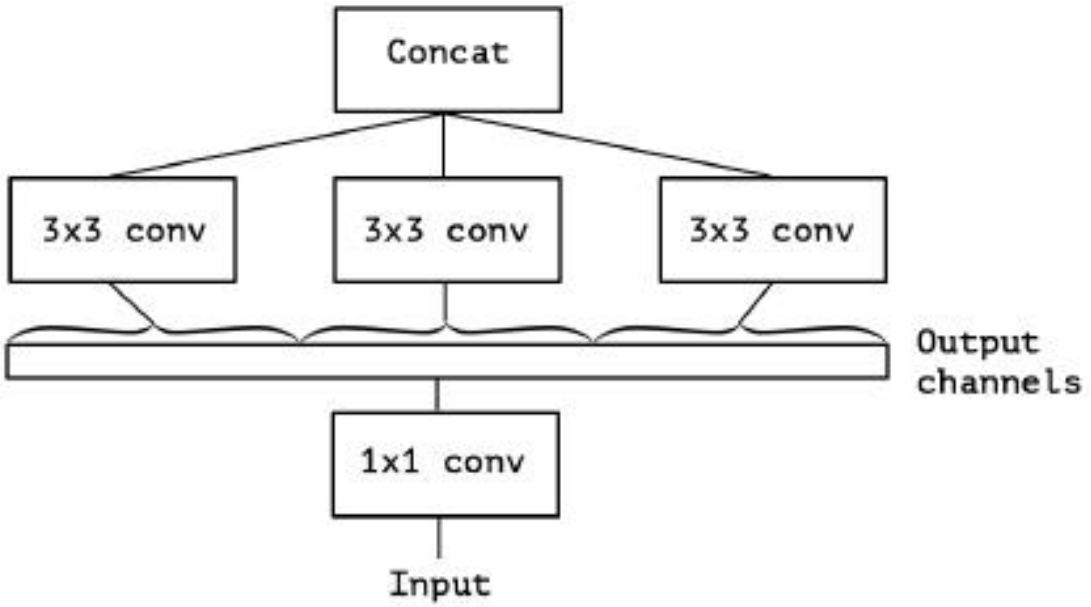
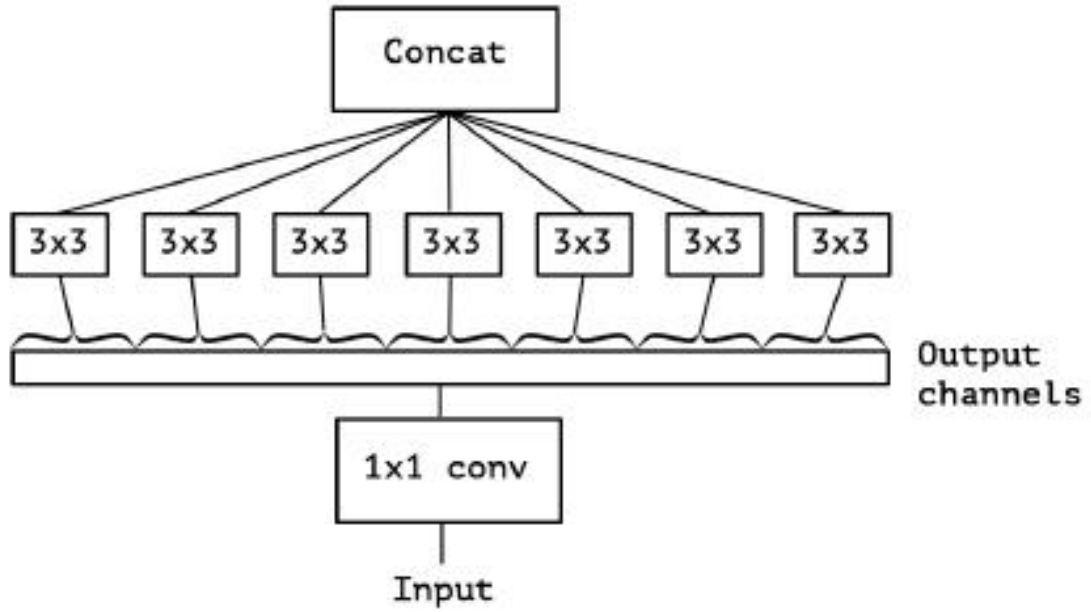
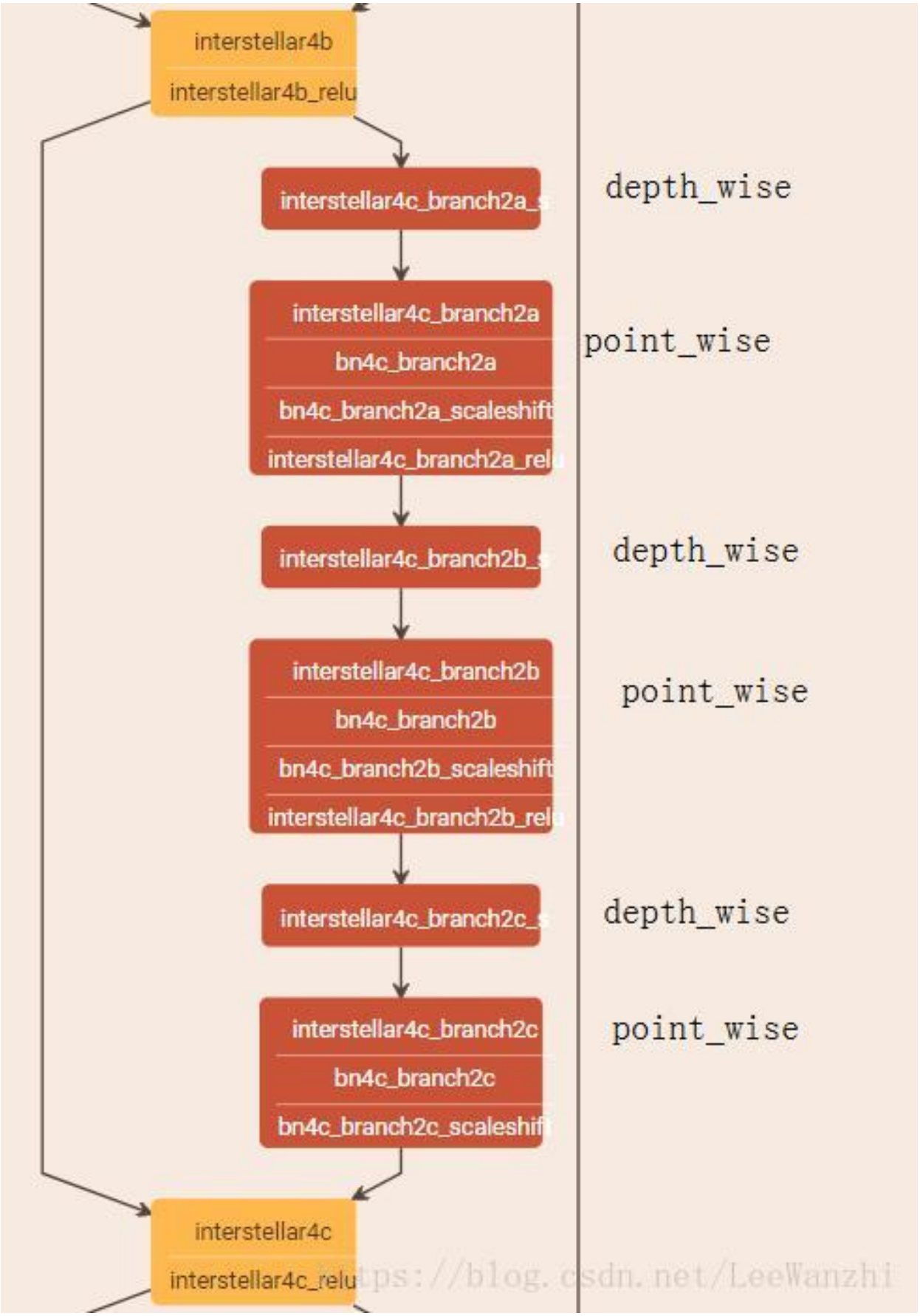


Figure 4. An "extreme" version of our Inception module, with one spatial convolution per output channel of the 1x1 convolution.



Inception路径比较多，Xception将其改进为depth_wise+point_wise结构，最后再concat或eltwise。



##参考文献:##

[1] https://blog.csdn.net/shuzfan/article/details/77141425