Questions：

1. 膨胀卷积实现的方法有哪些？

2. 膨胀卷积和转置卷积有什么关系？

3. Spatial Pyramis Pooling的作用？

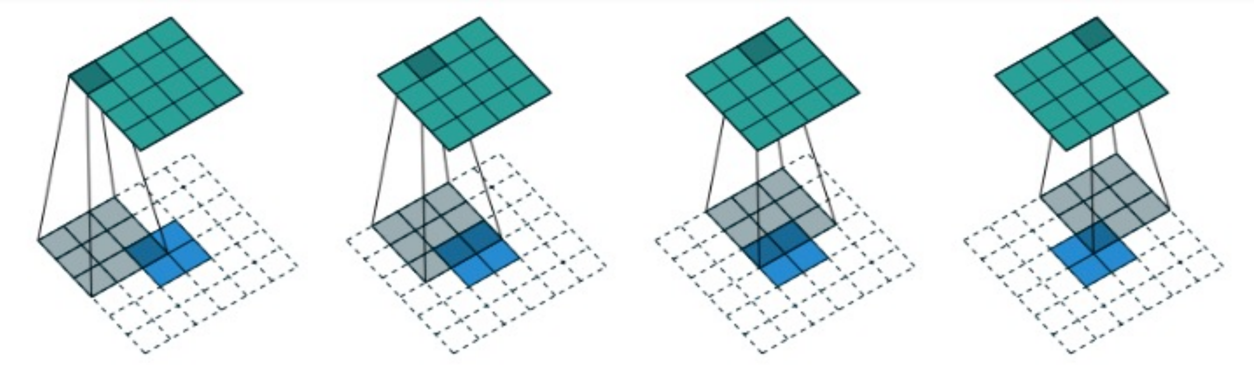
4. FCN 和 Deeplabv3+的区别是什么？

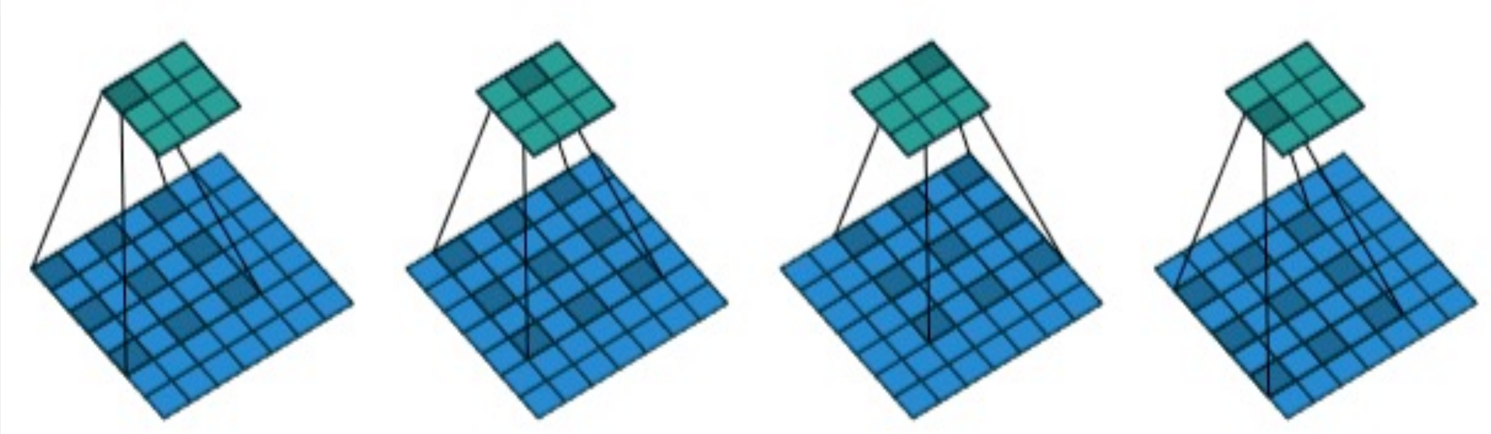
Question：

1. 膨胀卷积实现的方法有哪些？

A：（1）在kernel当中插入0，感受野变大，参数数量没有改变，但是效率比较低，
（2）输入等间隔采样

2. 膨胀卷积和转置卷积有什么关系？





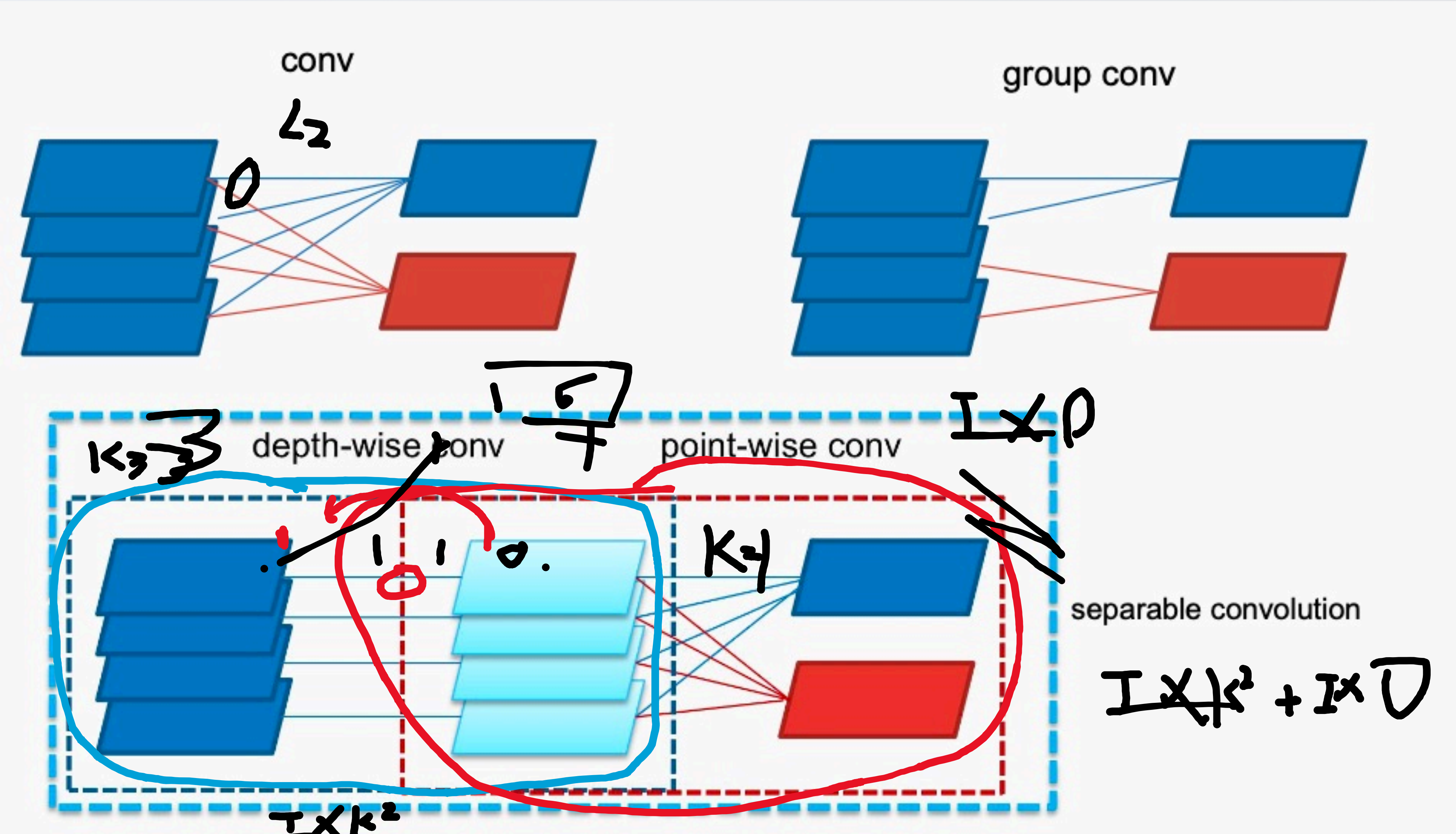
3. Spatial Pyramis Pooling的作用？

A：（1）多尺度融合
（2）可变输入变成固定长度，最后输出的长度:1*256+4*256+16*256 (256为channel个数）

4. FCN 和 Deeplabv3+的区别是什么？

A：在图像的后面，将普通的conv替换成为膨胀卷积，这导致了
（1）分辨率不减小
（2）感受野一致

卷积之间的联系和区别

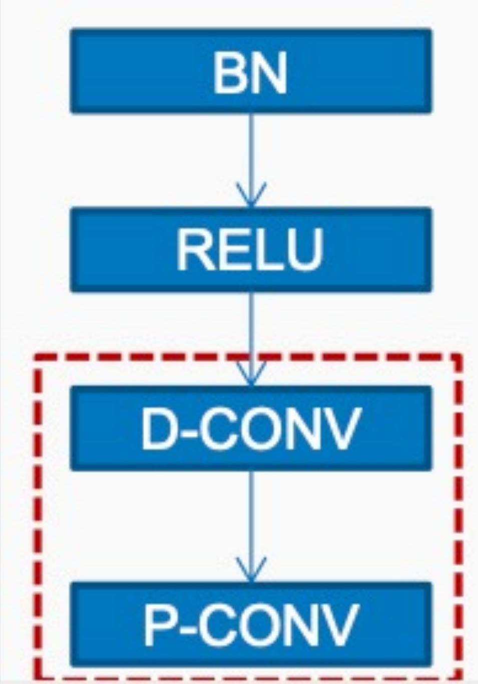

conv
group conv
depth-wise conv
point-wise conv
separable convolution
K=3
I×O
I×K² + I×O
I×K²

Depth-wise conv

Depth-wise conv一般与Point –wise conv 联合使用。在其使用时，可以看成把传统卷积变成空间卷积与通道卷积分开进行运算，解耦操作之间的相关性，减少了参数量和计算量。

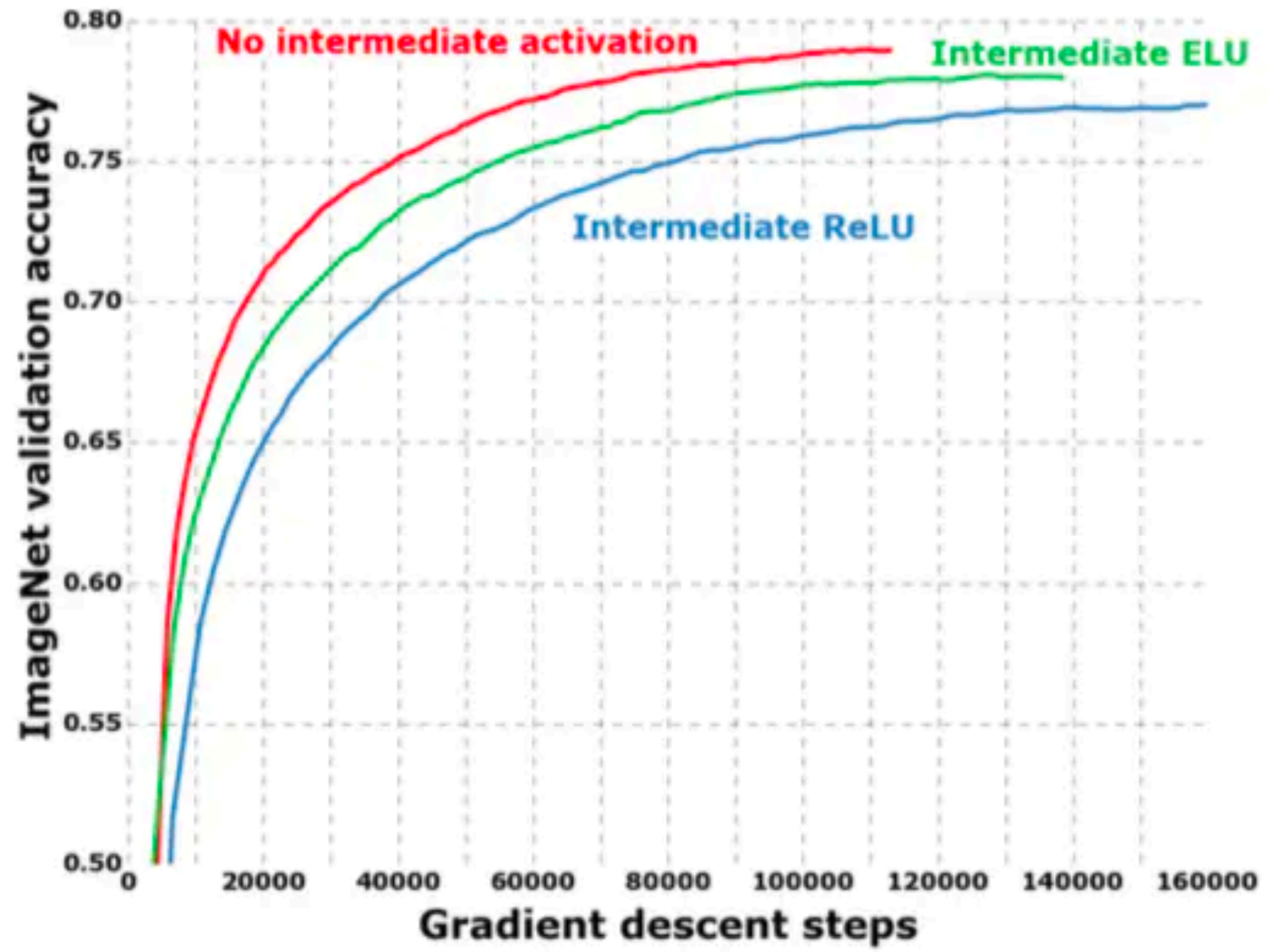
Depth-wise的conv的权重不加L2的约束

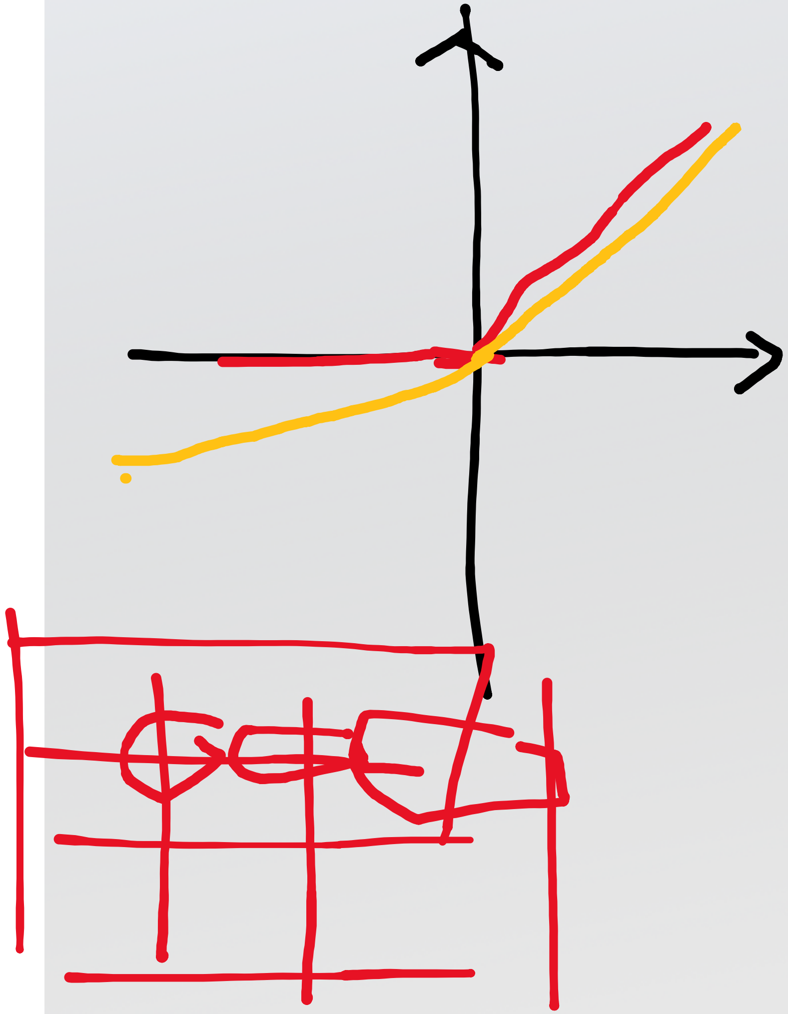
先用3*3的kernel建立空间之间的相关性（depth-wise conv）
再用1*1的kernel建立通道之间的相关性（point-wise conv）



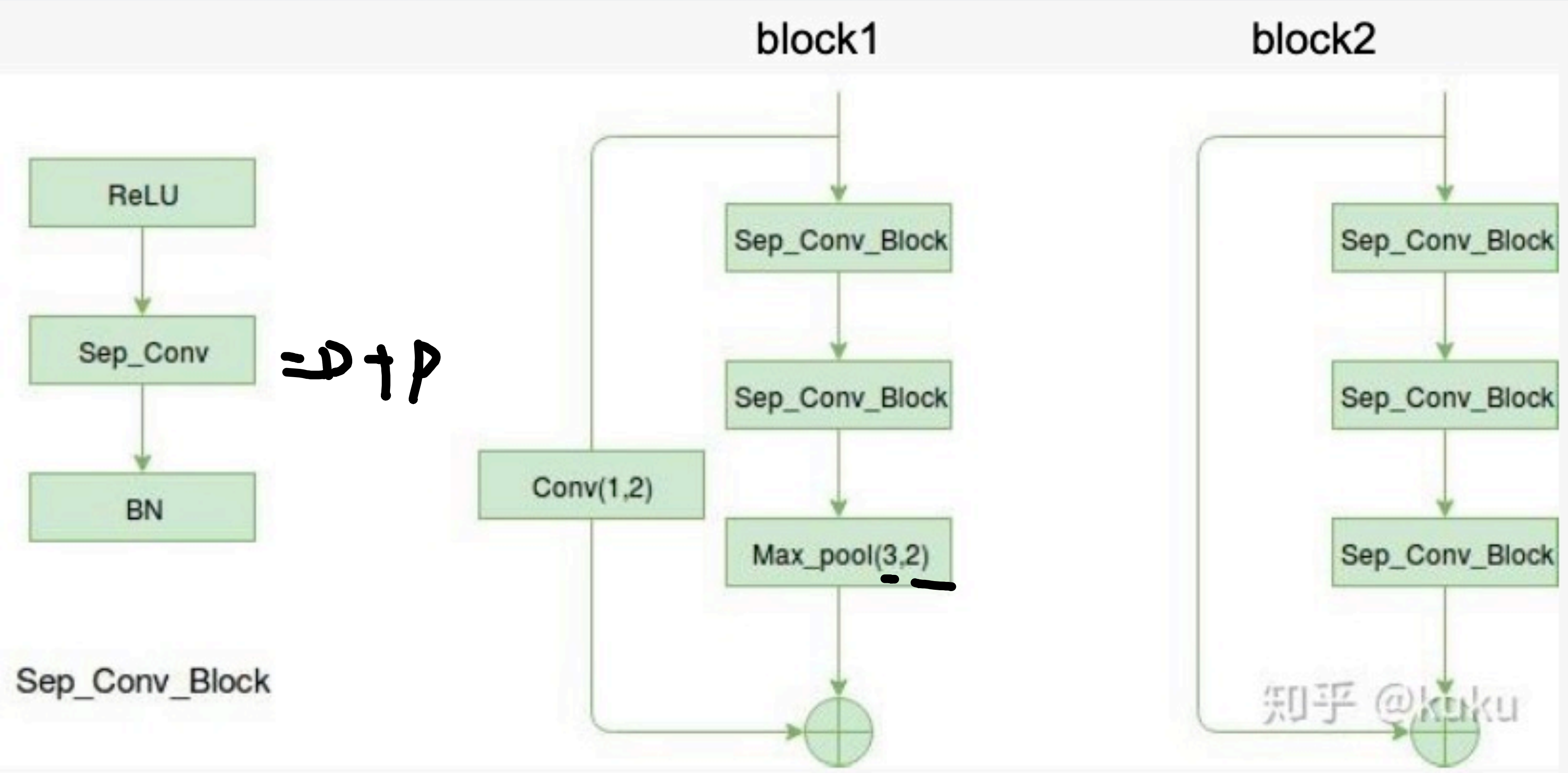
× ReLU × BN

Figure 10. Training profile with different activations between the depthwise and pointwise operations of the separable convolution layers.



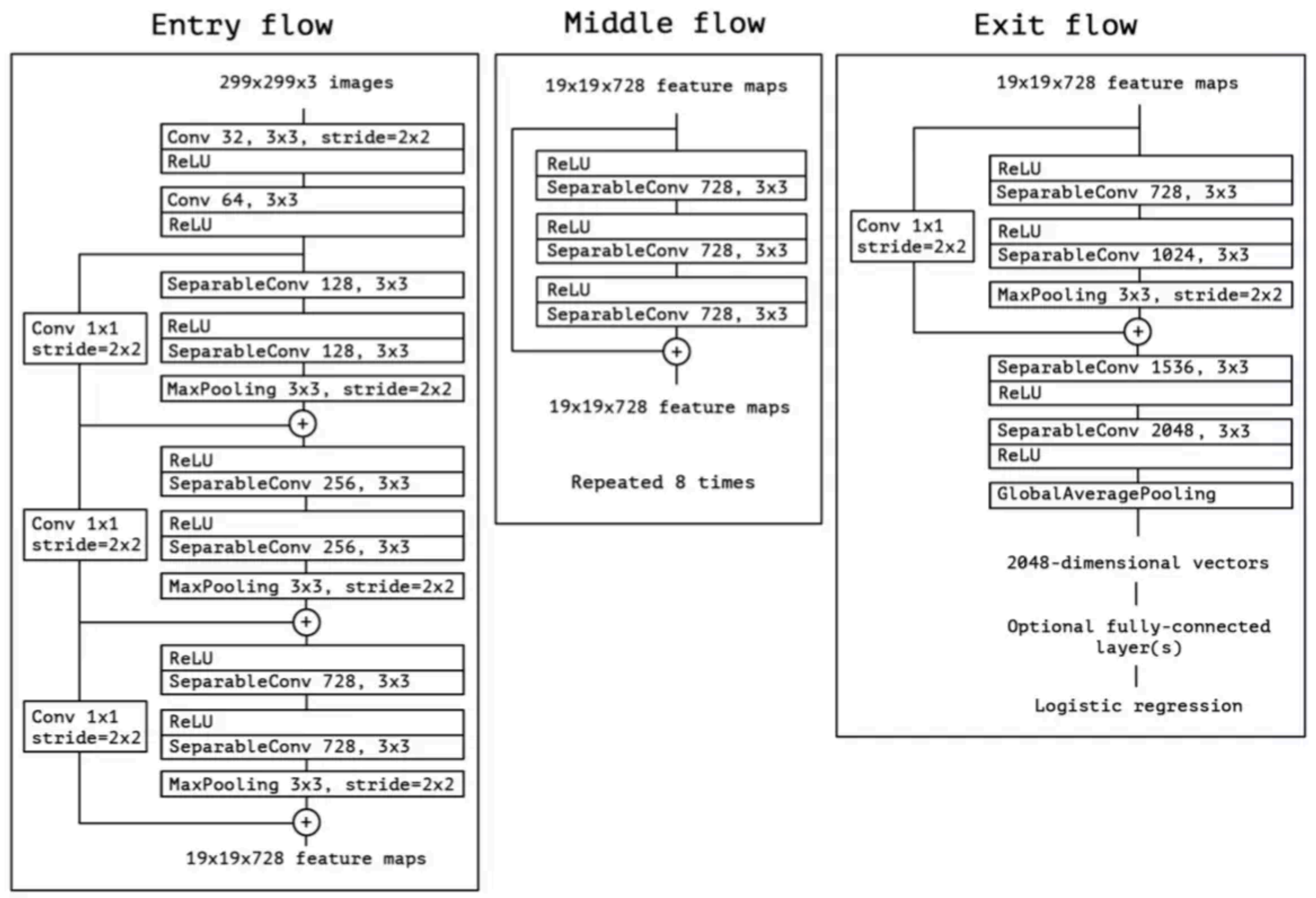


Xception 中尺寸变化时的block单元与中间的block操作单元



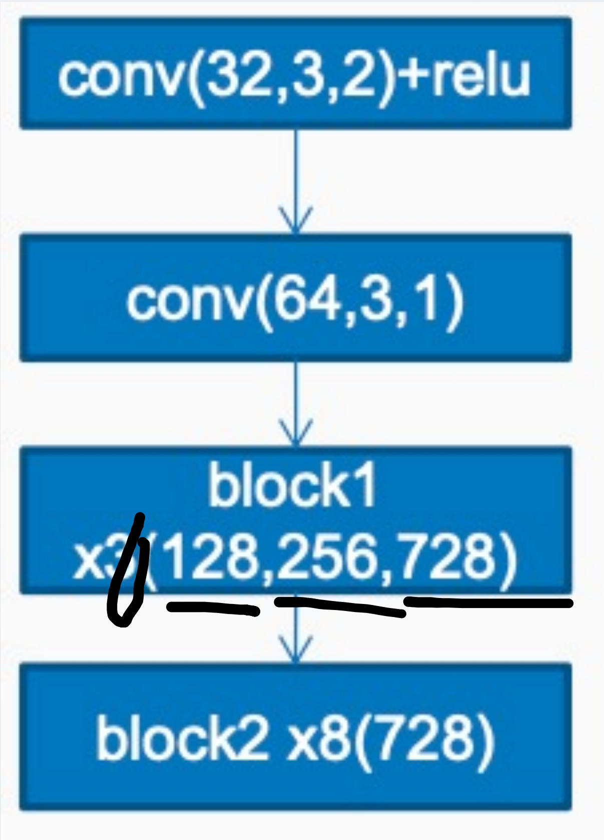
From： @kuku

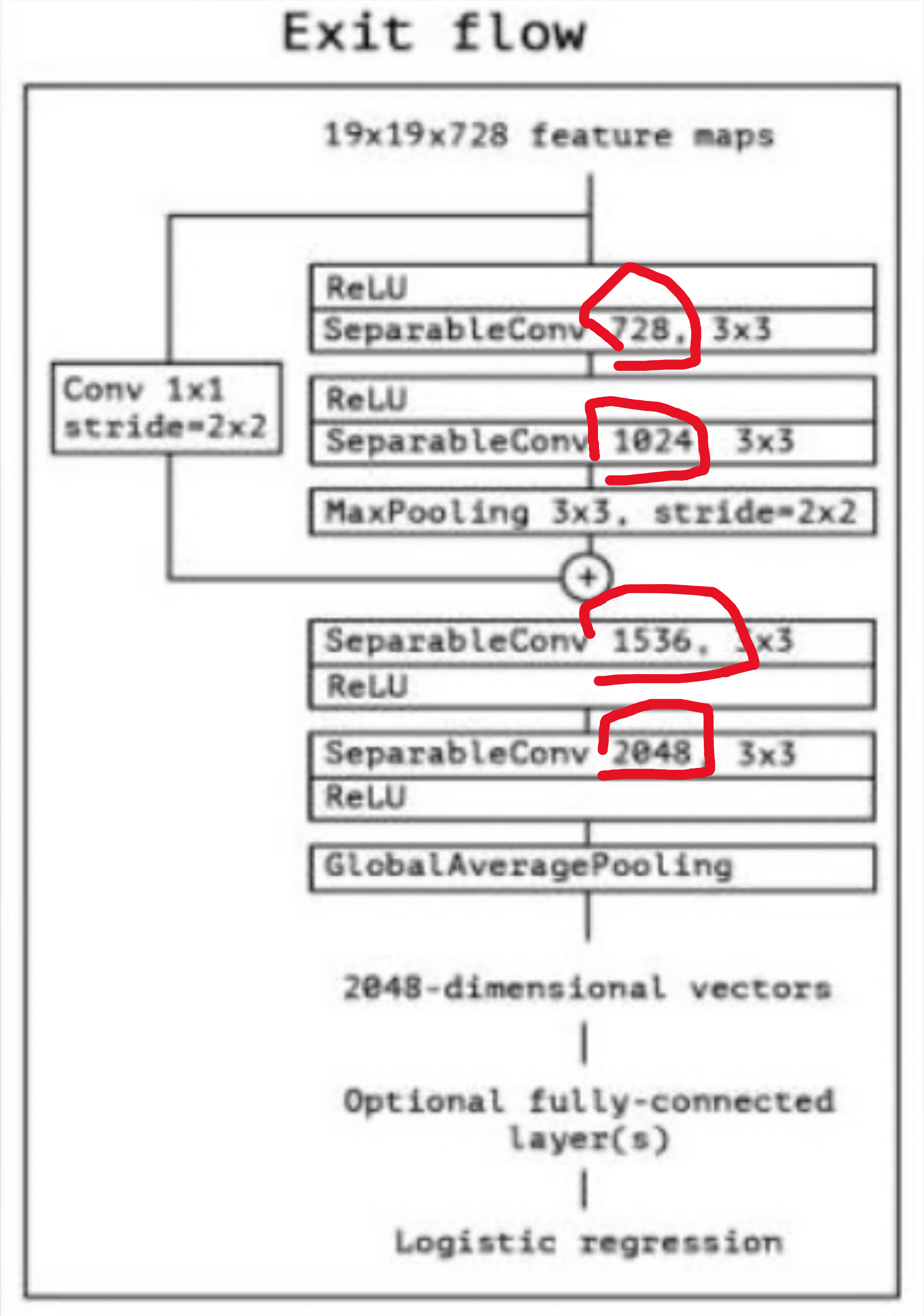
Figure 5. The Xception architecture: the data first goes through the entry flow, then through the middle flow which is repeated eight times, and finally through the exit flow. Note that all Convolution and SeparableConvolution layers are followed by batch normalization [7] (not included in the diagram). All SeparableConvolution layers use a depth multiplier of 1 (no depth expansion).



后厂理工学院
houchangtech.com

From: Xception-fig5

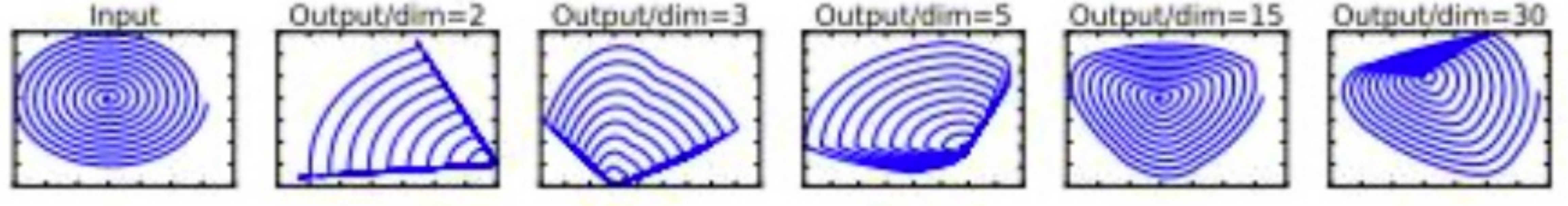

conv(32,3,2)+relu
conv(64,3,1)
block1
x3(128,256,728)
block2 x8(728)


Exit flow
19x19x728 feature maps
ReLU
SeparableConv 728, 3x3
ReLU
SeparableConv 1024 3x3
MaxPooling 3x3, stride=2x2
Conv 1x1
stride=2x2
SeparableConv 1536, 3x3
ReLU
SeparableConv 2048 3x3
ReLU
GlobalAveragePooling
2048-dimensional vectors
Optional fully-connected
layer(s)
Logistic regression

Mobilenet v2

$\begin{bmatrix} X \\ Y \end{bmatrix}$

Linear Bottlenecks

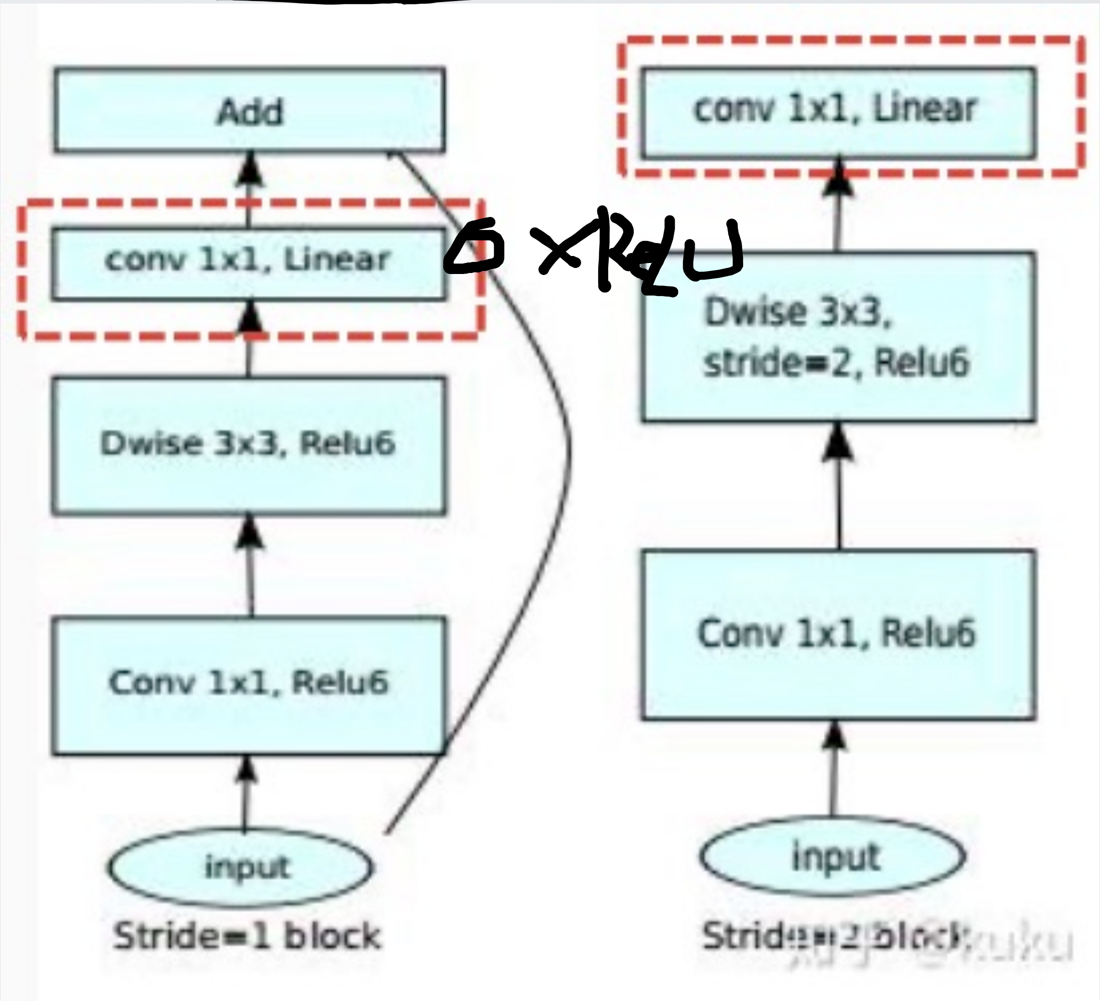


对X-Y空间运用随机生成的矩阵T进行变换，然后再接ReLU，再运用T^{-1}反变换回来，观察信息的损失程度

只有当感兴趣的流行信息是在高维空间的子空间里，ReLUctant才能保存所有信息，ReLU在相对低维空间下造成的信息损失比高维空间下大得多

Mobilenet v2

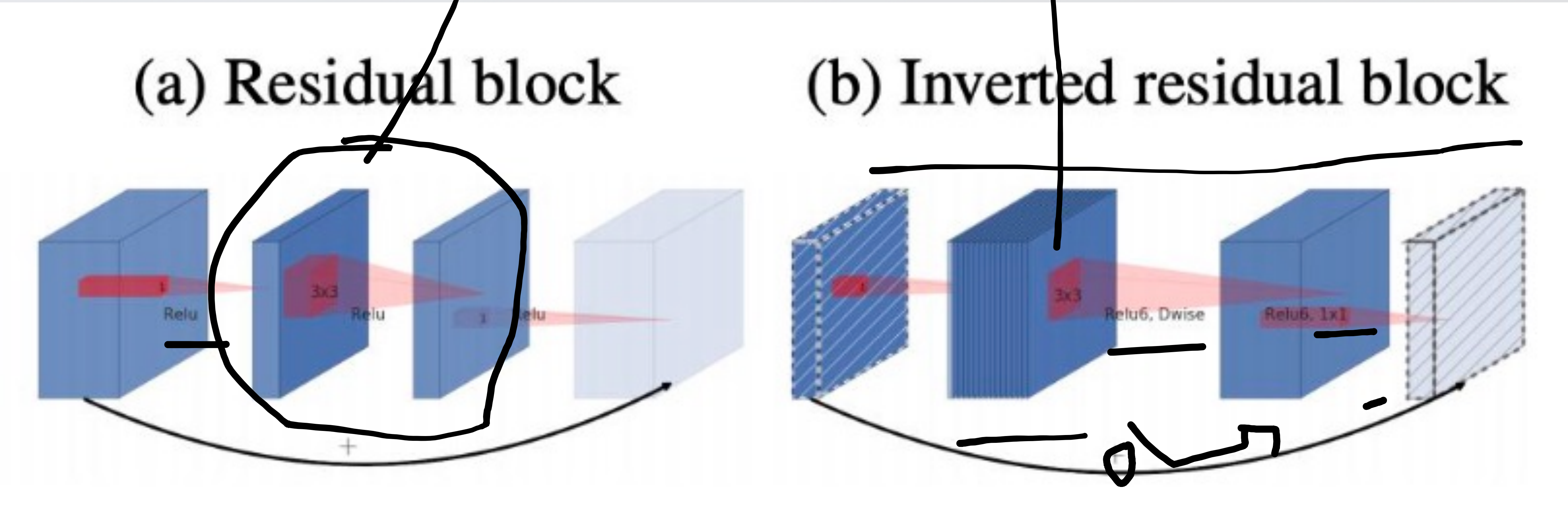
Linear Bottlenecks



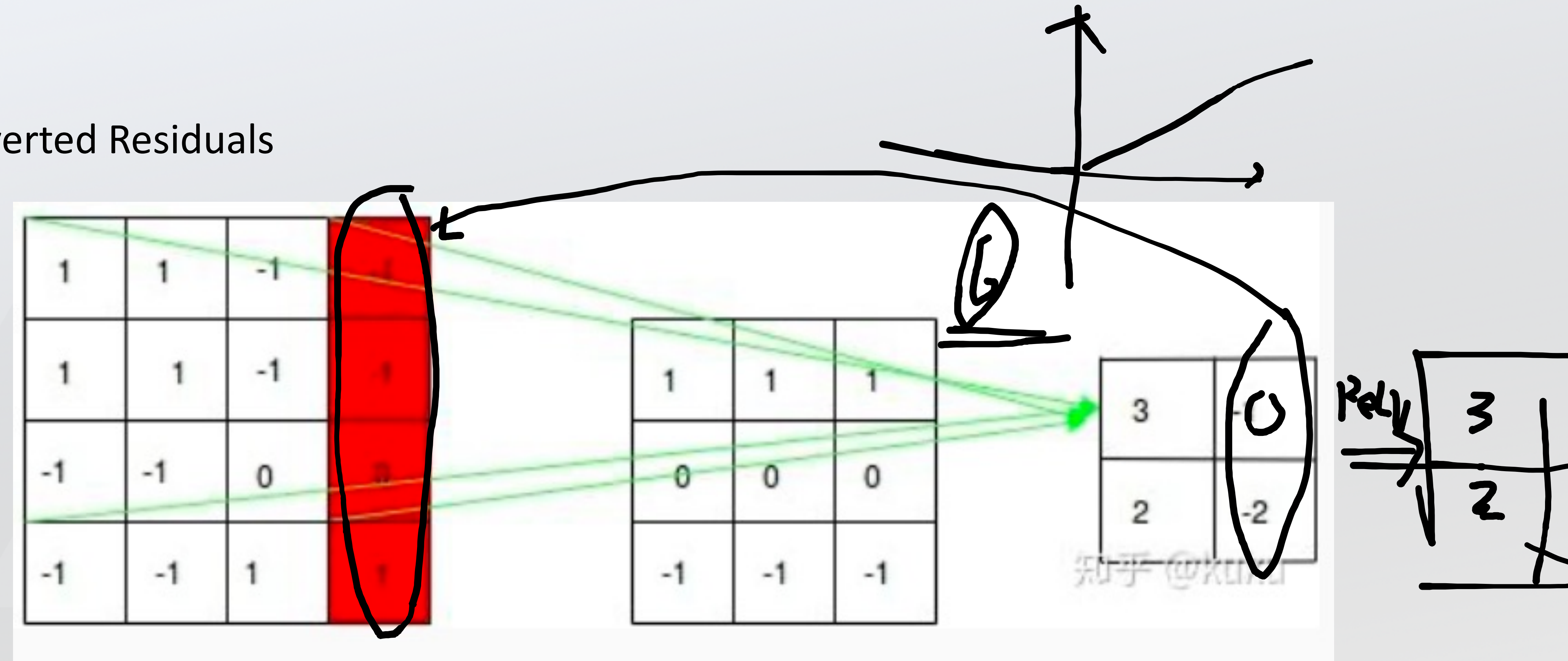
作者在通道变小后，没有使用ReLU6激活函数了

Inverted Residuals

PP × ReLU



Inverted Residuals



t

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

后厂理工学院

houchangtech.com

一所专注前沿互联网技术领域的创新实战大学