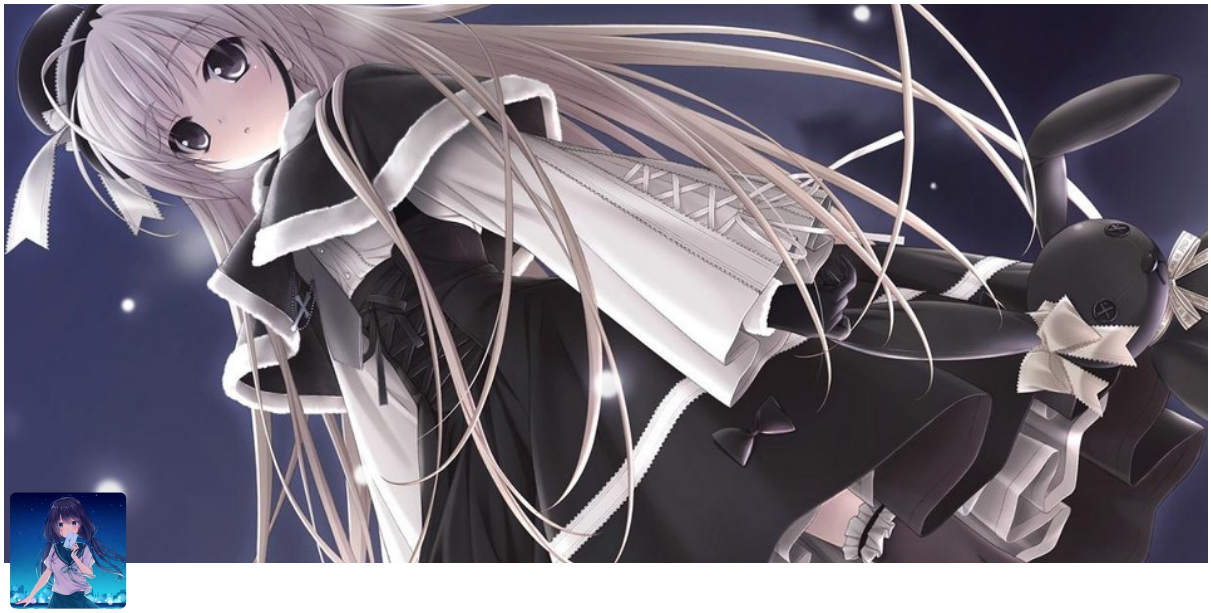


MobileNet网络

MobileNetV1:

网络亮点：

1.Depthwise Convolution(大大减少运算量和参数数量)

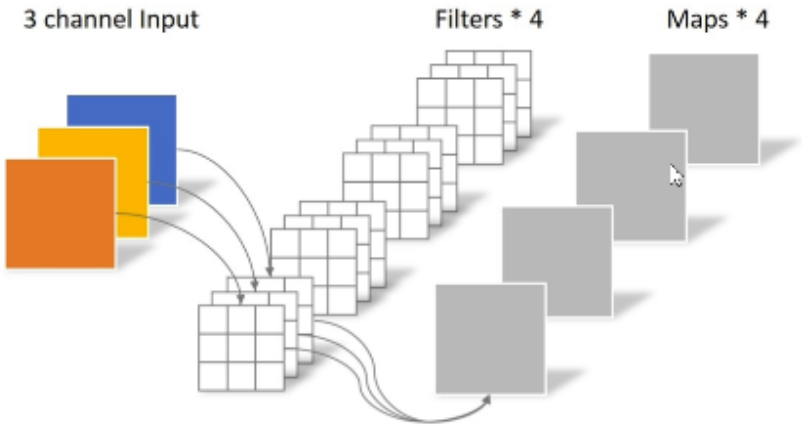
2.增加超参数α、β

传统卷积与DW卷积对比：

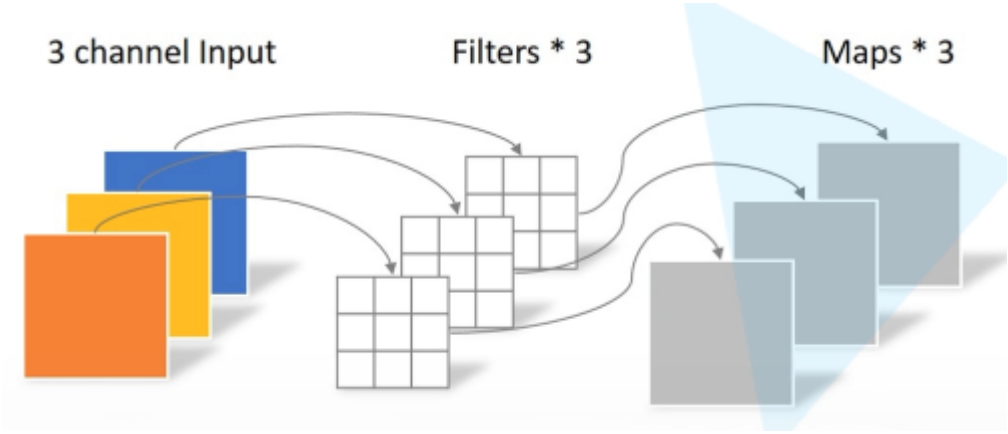
传统卷积：

1.卷积核channel=输入特征矩阵channel

2.输出特征矩阵channel=卷积核个数



Dw卷积：



1.卷积核channel=1

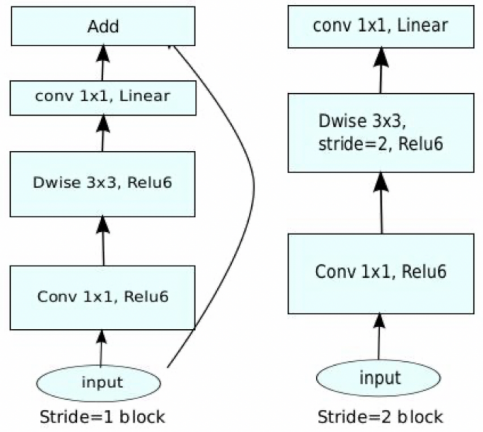
2.输入特征矩阵channel=卷积核个数=输出特征矩阵channel

MobileNekV2:

相对于V1版本，MobileNetV2，准确率更高，模型更小。

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	-	

网络整体结构



bottleneck 结构

网络亮点:

1.Inverted Residuals（倒残差结构)

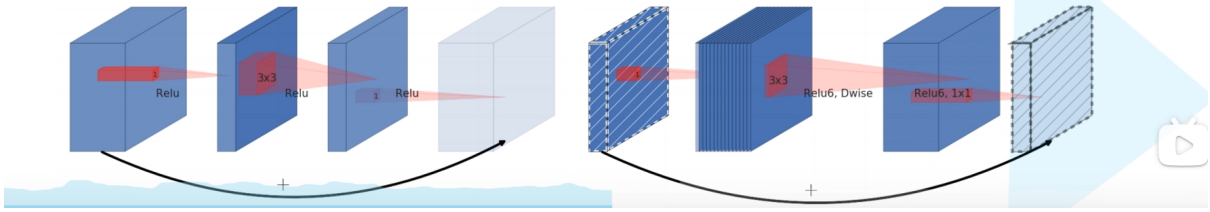
2.Linear Bottlenecks

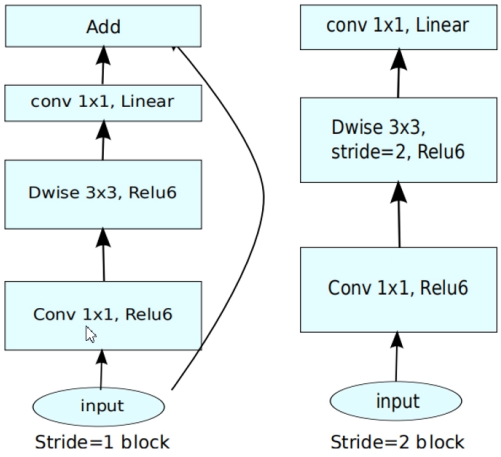
① 1x1 卷积降维
② 3x3 卷积
③ 1x1 卷积升维

① 1x1 卷积升维
② 3x3 卷积 DW
③ 1x1 卷积降维

(a) Residual block

(b) Inverted residual block





(d) Mobilenet V2

当stride=1且输入特征矩阵与输出特征矩阵shape相同时才有shortcut连接。

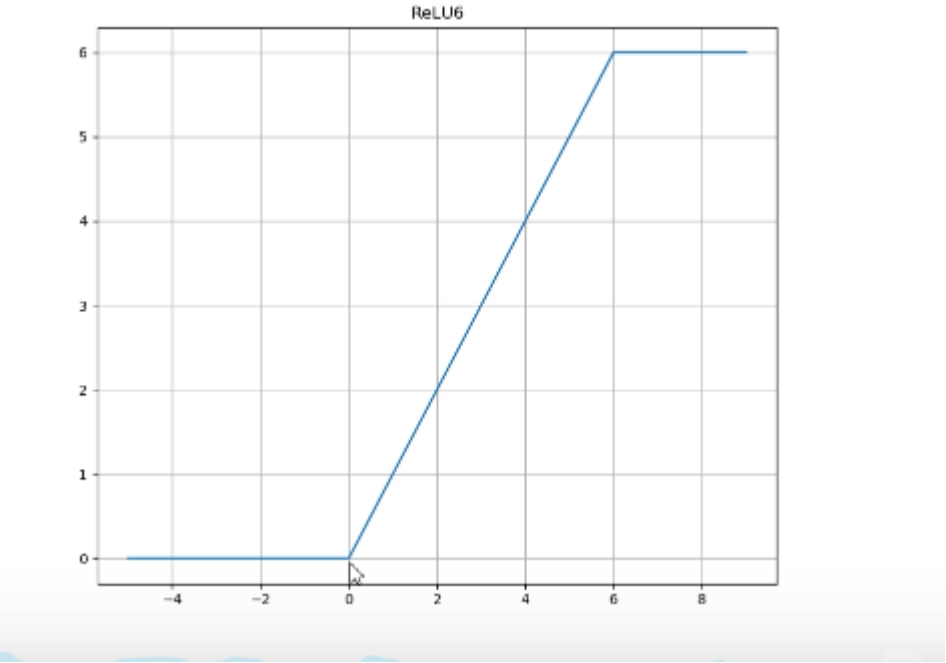
Relu6:

原因：ReLU激活函数对低维特征信息照成大量损失。

公式：

$$y = \text{ReLU6}(x) = \min(\max(x, 0), 6)$$

图像：



MobileNekV3:

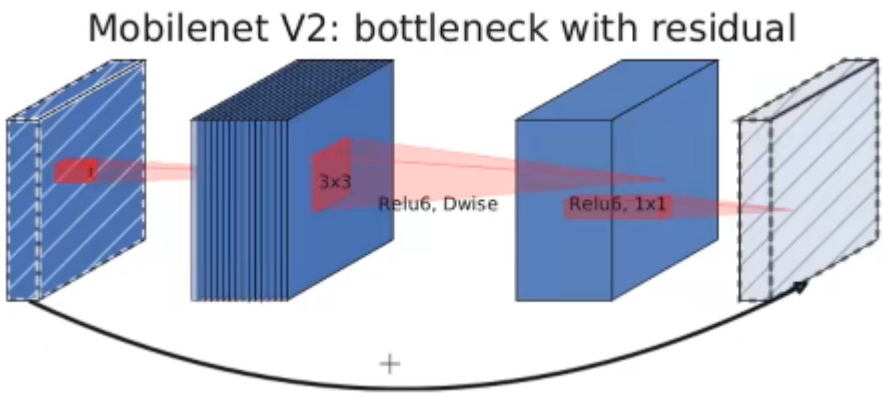
网络亮点：

1.更新了block

2.使用NAS搜索参数（用于参数优化）

3.重新设计耗时层结构

更新block：



- 1×1卷积用于升维和降维
- NL代表使用非线性激活，包含Relu以及h-swish激活函数

- Dwise（Depthwise Conv）代表使用深度可分离卷积（即每个卷积核仅在每个channel上进行卷积操作，卷积个数同通道维数）
- SE结构为（Squeeze-and-Excite）注意力机制（专门有篇论文提出这个网络结构），简单理解就是对C×H×W的特征图，对每一个维度进行全局均值池化，一共得到C个值，然后作为全连接层的输入，然后隐含层设为0.25×C（本文设为0.25），输出成设为C，一共得到C个值作为对应维度的权重与输入的C×H×W进行点乘（每个权重C乘以对应的维度H×W）得到结果进行输出
- 当输入维度和输出维度及大小都相同时，需进行跳跃连接（上图的连线），即对应位置相加

重新设计耗时层结构：

1.减少第一个卷积层的卷积核个数

2.精简 Last stage