

MobileNet(v1、v2)网络详解与模型的搭建

首先给出三个链接：

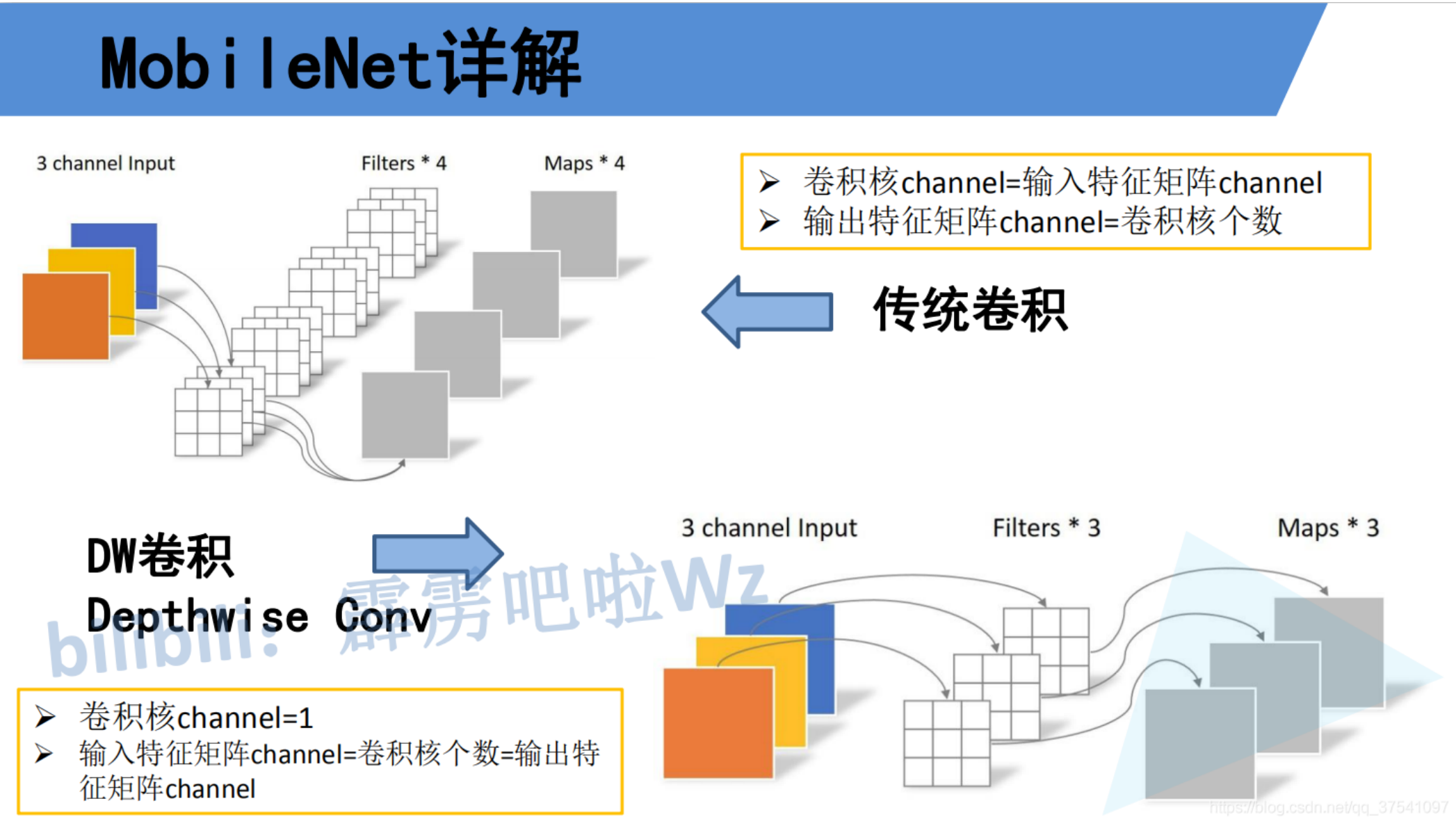
- 1. MobileNet (v1, v2) 网络详解视频
- 2. 使用pytorch搭建mobilenet v2并基于迁移学习训练视频
- 3. 使用tensorflow2搭建mobilenet v2并基于迁移学习训练视频

在之前的文章中讲的AlexNet、VGG、GoogLeNet以及ResNet网络，它们都是传统卷积神经网络（都是使用的传统卷积层），缺点在于内存需求大、运算量大导致无法在移动设备以及嵌入式设备上运行。而本文要讲的MobileNet网络就是专门为移动端，嵌入式端而设计。

MobileNet v1

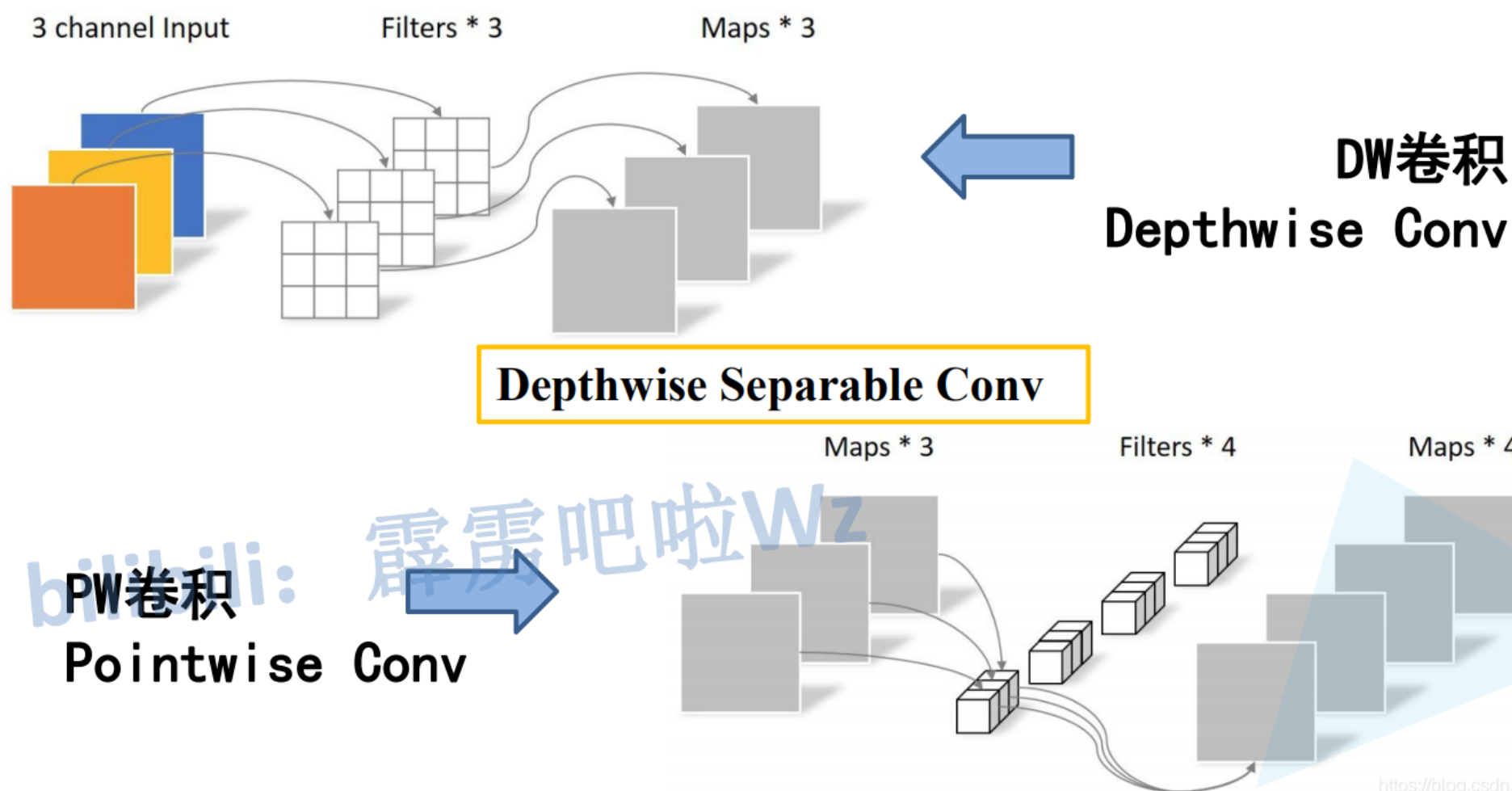
MobileNet网络是由google团队在2017年提出的，专注于移动端或者嵌入式设备中的轻量级CNN网络。相比传统卷积神经网络，在准确率小幅降低的前提下大大减少模型参数与运算量。(相比VGG16准确率减少了0.9%，但模型参数只有VGG的1/32)。

要说MobileNet网络的优点，无疑是其中的Depthwise Convolution结构(大大减少运算量和参数数量)。下图展示了传统卷积与DW卷积的差异，在传统卷积中，每个卷积核的channel与输入特征矩阵的channel相等（每个卷积核都会与输入特征矩阵的每一个维度进行卷积运算）。而在DW卷积中，每个卷积核的channel都是等于1的（每个卷积核只负责输入特征矩阵的一个channel，故卷积核的个数必须等于输入特征矩阵的channel数，从而使得输出特征矩阵的channel数也等于输入特征矩阵的channel数）



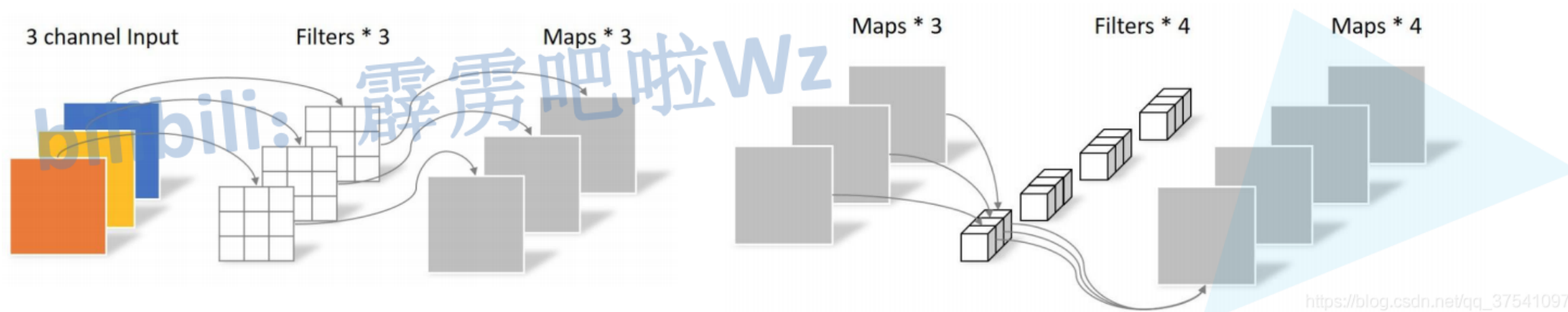
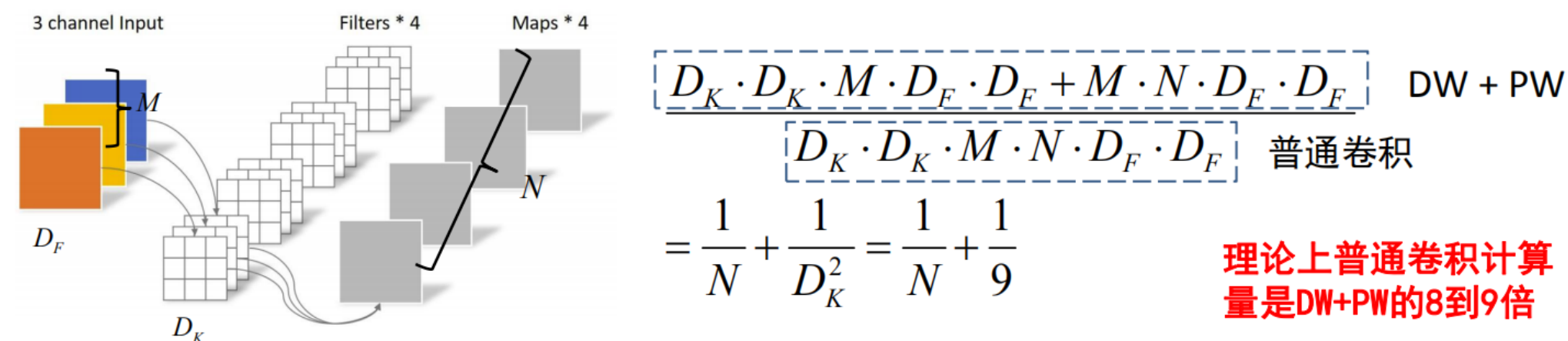
刚刚说了使用DW卷积后输出特征矩阵的channel是与输入特征矩阵的channel相等的，如果想改变/自定义输出特征矩阵的channel，那只需要在DW卷积后接上一个PW卷积即可，如下图所示，其实PW卷积就是普通的卷积而已（只不过卷积核大小为1）。通常DW卷积和PW卷积是放在一起使用的，一起叫做Depthwise Separable Convolution（深度可分卷积）。

Mob i l eNet详解



那Depthwise Separable Convolution（深度可分卷积）与传统的卷积相比有到底能节省多少计算量呢，下图对比了这两个卷积方式的计算量，其中 D_f 是输入特征矩阵的宽高（这里假设宽和高相等）， D_k 是卷积核的大小， M 是输入特征矩阵的channel， N 是输出特征矩阵的channel，卷积计算量近似等于卷积核的高 \times 卷积核的宽 \times 卷积核的channel \times 输入特征矩阵的高 \times 输入特征矩阵的宽（这里假设stride等于1），在我们mobilenet网络中DW卷积都是使用3x3大小的卷积核。所以理论上普通卷积计算量是DW+PW卷积的8到9倍（公式来源于原论文）：

Mob i l eNet详解



在了解完Depthwise Separable Convolution（深度可分卷积）后在看下mobilenet v1的网络结构，左侧的表格是mobileNetv1的网络结构，表中标Conv的表示普通卷积，Conv dw代表刚刚说的DW卷积，s表示步距，根据表格信息就能很容易的搭建出mobileNet v1网络。在mobilenetv1原论文中，还提出了两个超参数，一个是 α 一个是 β 。 α 参数是一个倍率因子，用来调整卷积核的个数， β 是控制输入网络的图像尺寸参数，下图右侧给出了使用不同 α 和 β 网络的分类准确率，计算量以及模型参数：

Mob i l eNet详解

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32 \text{ dw}$	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64 \text{ dw}$	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128 \text{ dw}$	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128 \text{ dw}$	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256 \text{ dw}$	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256 \text{ dw}$	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1	$3 \times 3 \times 512 \text{ dw}$	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512 \text{ dw}$	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024 \text{ dw}$	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Table 8. MobileNet Comparison to Popular Models

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
GoogLeNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

Multiply-Add
计算量

Table 6. MobileNet Width Multiplier

Width Multiplier	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
0.75 MobileNet-224	68.4%	325	2.6
0.5 MobileNet-224	63.7%	149	1.3
0.25 MobileNet-224	50.6%	41	0.5

α

Width Multiplier

Table 7. MobileNet Resolution

Resolution	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
1.0 MobileNet-192	69.1%	418	4.2
1.0 MobileNet-160	67.2%	290	4.2
1.0 MobileNet-128	64.4%	186	4.2

β

Resolution Multiplier

depthwise部分的卷积核容易费掉，即卷积核参数大部分为零。

MobileNet v2

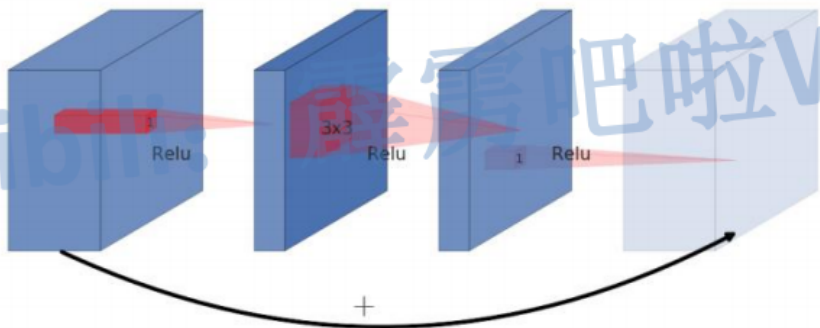
在MobileNet v1的网络结构表中能够发现，网络的结构就像VGG一样是个直筒型的，不像ResNet网络有shortcut之类的连接方式。而且有人反映说MobileNet v1网络中的DW卷积很容易训练废掉，效果并没有那么理想。所以我们接着看下MobileNet v2网络。

MobileNet v2网络是由google团队在2018年提出的，相比MobileNet V1网络，准确率更高，模型更小。刚刚说了MobileNet v1网络中的亮点是DW卷积，那么在MobileNet v2中的亮点就是Inverted residual block（倒残差结构），如下下图所示，左侧是ResNet网络中的残差结构，右侧就是MobileNet v2中的到残差结构。在残差结构中是1x1卷积降维->3x3卷积->1x1卷积升维，在倒残差结构中正好相反，是1x1卷积升维->3x3DW卷积->1x1卷积降维。为什么要这样做，原文的解释是高维信息通过ReLU激活函数后丢失的信息更少（注意倒残差结构中基本使用的都是ReLU6激活函数，但是最后一个1x1的卷积层使用的是线性激活函数）。

Mob i l eNet详解

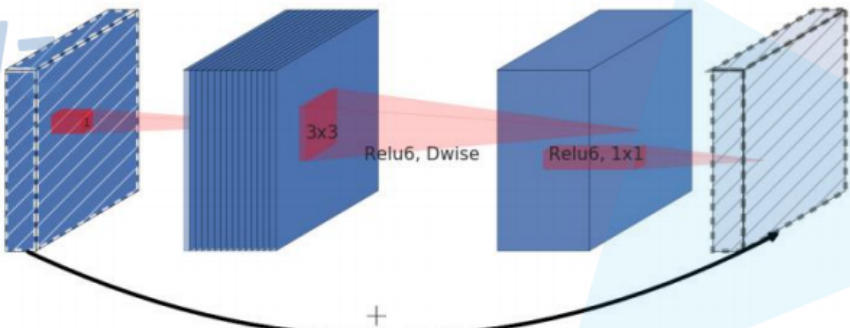
- ① 1x1 卷积降维
- ② 3x3 卷积
- ③ 1x1 卷积升维

(a) Residual block



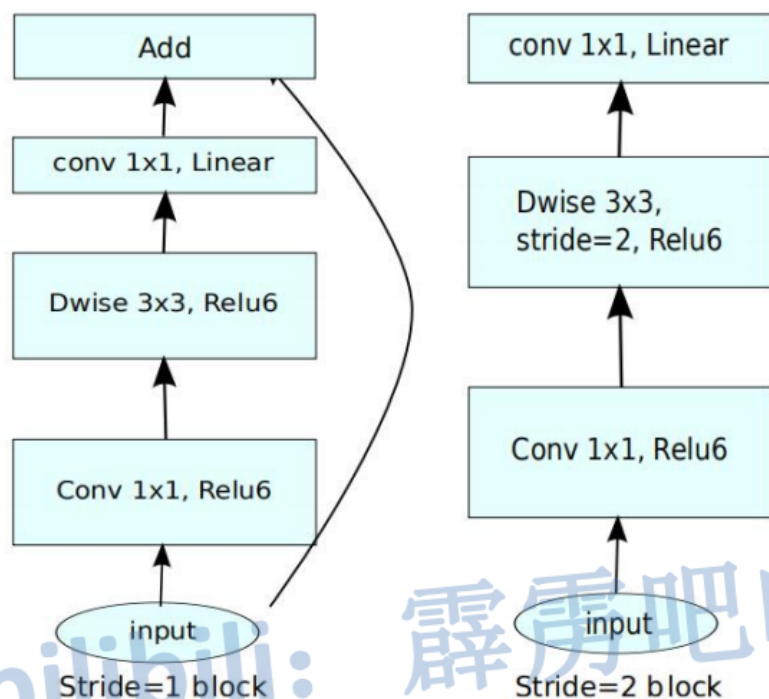
- ① 1x1 卷积升维
- ② 3x3 卷积
- ③ 1x1 卷积降维

(b) Inverted residual block



在使用倒残差结构时需要注意下，并不是所有的倒残差结构都有shortcut连接，只有当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（只有当shape相同时，两个矩阵才能做加法运算，当stride=1时并不能保证输入特征矩阵的channel与输出特征矩阵的channel相同）。

Mob i l eNet详解



Input	Operator	Output
$h \times w \times k$	1x1 conv2d, ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwse s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

Table 1: *Bottleneck residual block* transforming from k to k' channels, with stride s , and expansion factor t .

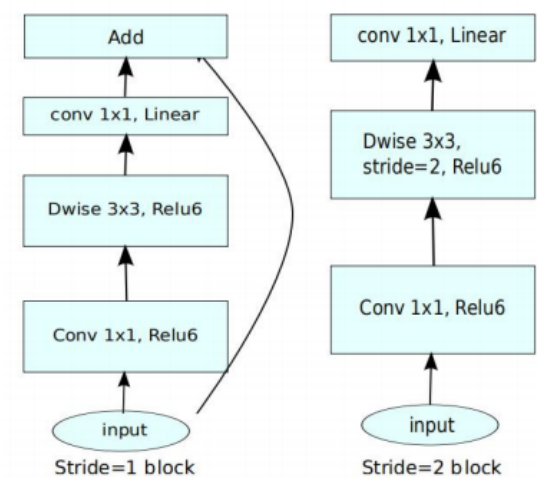
当**stride=1**且输入特征矩阵与输出特征矩阵**shape**相同时才有**shortcut**连接

(d) Mobilenet V2

下图是MobileNet v2网络的结构表，其中t代表的是扩展因子（倒残差结构中第一个1x1卷积的扩展因子），c代表输出特征矩阵的channel，n代表倒残差结构重复的次数，s代表步距（注意：这里的步距只是针对重复n次的第一层倒残差结构，后面的都默认为1）。

Mob i l eNet详解

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	-	-



- t是扩展因子
- c是输出特征矩阵深度channel
- n是bottleneck的重复次数
- s是步距（针对第一层，其他为1）

关于MobileNet V2模型的搭建与训练代码放在我的github中，大家可自行下载使用：

<https://github.com/WZMIAOMIAO/deep-learning-for-image-processing>

pytorch版本在pytorch_learning文件夹中，tensorflow版本在tensorflow_learning文件夹中。

相关推荐

关于我们 招贤纳士 广告服务 开发助手 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 网络110报警服务 中国互联网举报中心 家长监护 Chrome商店下载 ©1999-2021北京创新乐知网络技术有限公司 版权与免责声明 版权申诉 出版物许可证 营业执照

