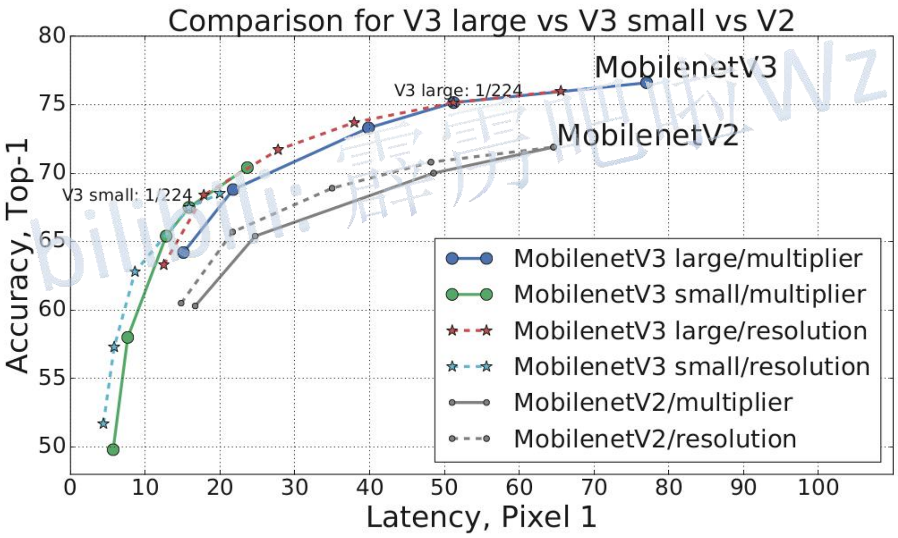
深度学习-图像分类篇

作者：神秘的wz

MobileNet V3

Searching for MobileNetV3

Andrew Howard[1] Mark Sandler[1] Grace Chu[1] Liang-Chieh Chen[1] Bo Chen[1] Mingxing Tan[2]
Weijun Wang[1] Yukun Zhu[1] Ruoming Pang[2] Vijay Vasudevan[2] Quoc V. Le[2] Hartwig Adam[1]
[1]Google AI, [2]Google Brain
{howarda, sandler, cxy, lcchen, bochen, tanmingxing, weijunw, yukun, rpang, vrv, qvl, hadam}@google.com



- 更新Block(bneck)
- 使用NAS搜索参数
  （Neural Architecture Search）
- 重新设计耗时层结构

github上有论文下载链接

MobileNet V3

Atrous Spatial Pyramid Pooling (LR-ASPP). We achieve new state of the art results for mobile classification, detection and segmentation. MobileNetV3-Large is 3.2% more accurate on ImageNet classification while reducing latency by 20% compared to MobileNetV2. MobileNetV3-Small is 6.6% more accurate compared to a MobileNetV2 model with comparable latency. MobileNetV3-Large detection is over 25% faster at roughly the same accuracy as MobileNetV2 on COCO detection. MobileNetV3-Large LR-ASPP is 34% faster than MobileNetV2 R-ASPP at similar accuracy for Cityscapes segmentation.

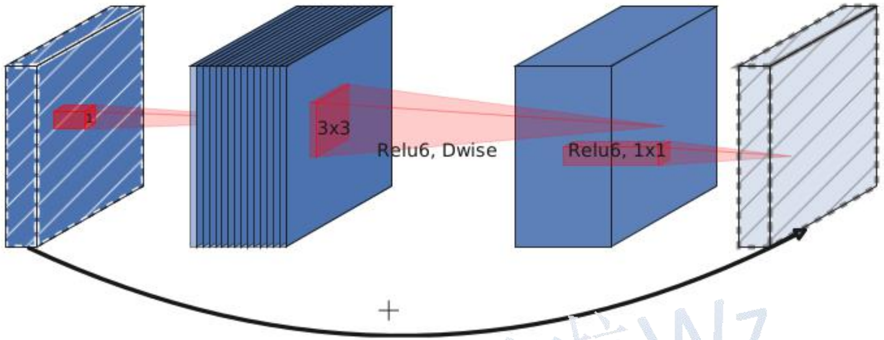
更准确，更高效

Network	Top-1	MAdds	Params	P-1	P-2	P-3
V3-Large 1.0	**75.2**	**219**	5.4M	**51**	**61**	**44**
V3-Large 0.75	73.3	155	4.0M	39	46	40
MnasNet-A1	75.2	315	3.9M	71	86	61
Proxyless[5]	74.6	320	4.0M	72	84	60
V2 1.0	72.0	300	3.4M	64	76	56
V3-Small 1.0	**67.4**	**56**	2.5M	**15.8**	**19.4**	**14.4**
V3-Small 0.75	65.4	44	2.0M	12.8	15.6	11.7
Mnas-small [43]	64.9	65.1	1.9M	20.3	24.2	17.2
V2 0.35	60.8	59.2	1.6M	16.6	19.6	13.9

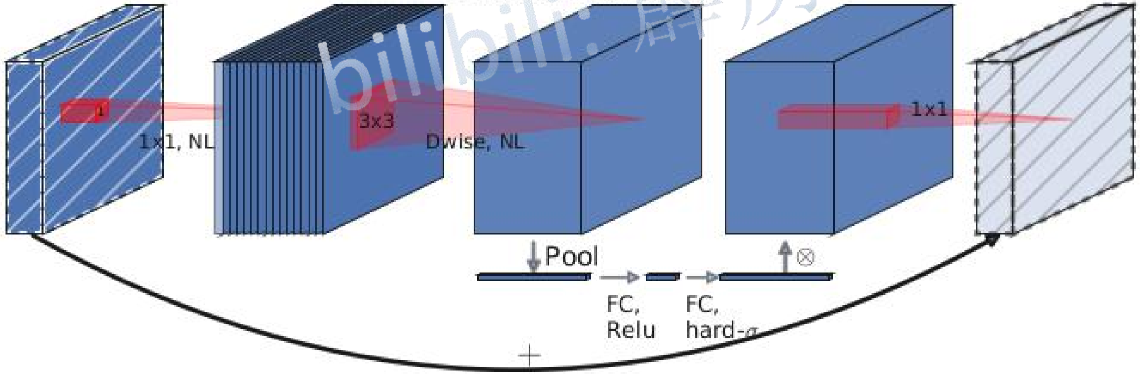
MobileNet V3

更新Block

1. 加入SE模块
2. 更新了激活函数



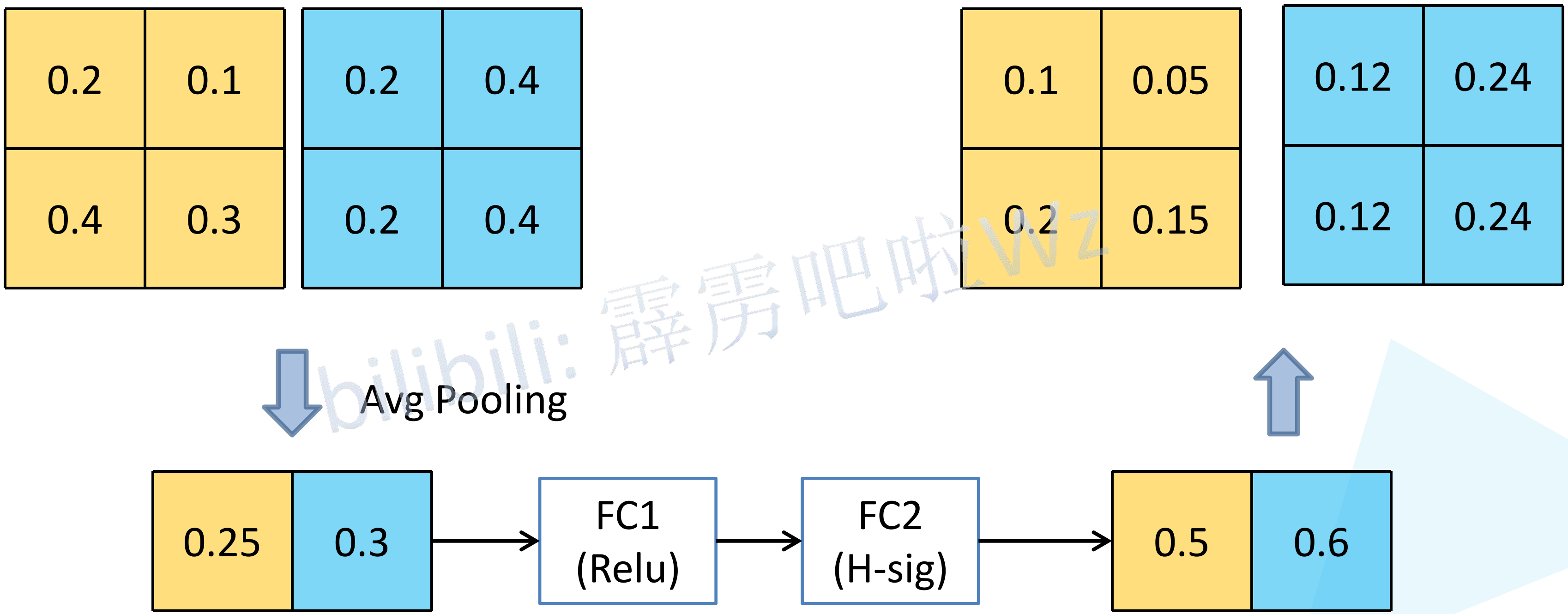
当stride == 1且
input_c == output_c
才有shortcut连接



5.3. Large squeeze-and-excite

In [43], the size of the squeeze-and-excite bottleneck was relative the size of the convolutional bottleneck. Instead, we replace them all to fixed to be 1/4 of the number of channels in expansion layer. We find that doing so increases the accuracy, at the modest increase of number of parameters, and no discernible latency cost.

MobileNet V3

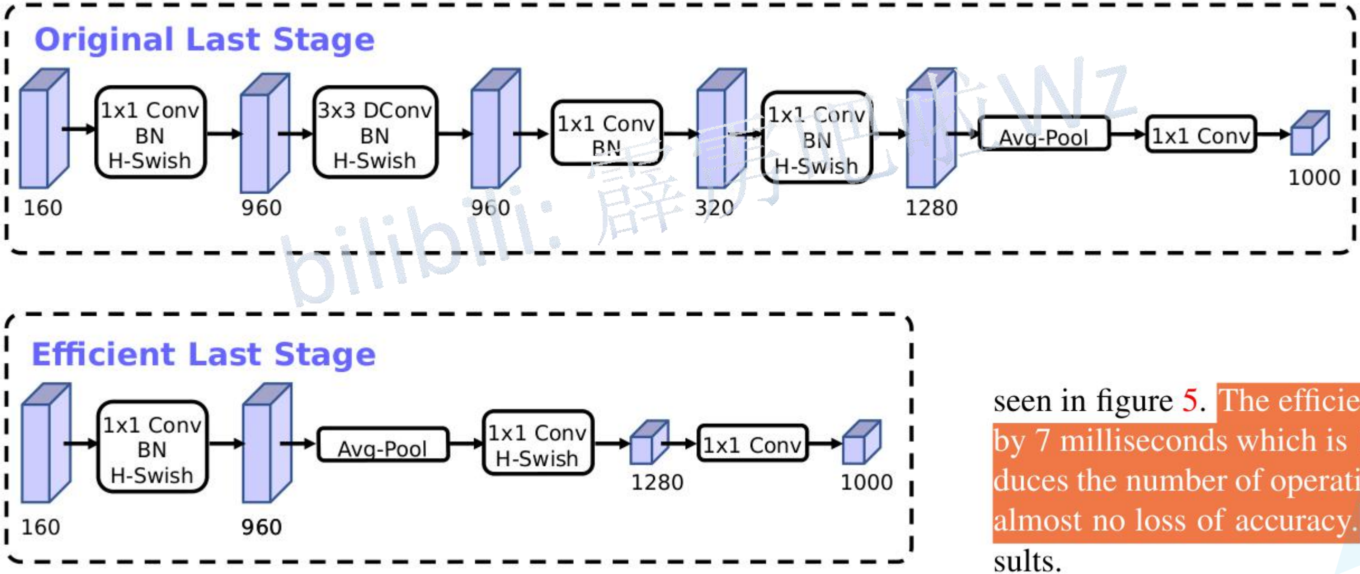


MobileNet V3

重新设计耗时层结构

1. 减少第一个卷积层的卷积核个数(32->16)
2. 精简Last Stage

different nonlinearities to try and reduce redundancy. We settled on using the hard swish nonlinearity for this layer as it performed as well as other nonlinearities tested. We were able to reduce the number of filters to 16 while maintaining the same accuracy as 32 filters using either ReLU or swish. This saves an additional 2 milliseconds and 10 million MAdds.



seen in figure 5. The efficient last stage reduces the latency by 7 milliseconds which is 11% of the running time and reduces the number of operations by 30 millions MAdds with almost no loss of accuracy. Section 6 contains detailed results.

MobileNet V3

重新设计激活函数

5.2. Nonlinearities

In [36, 13, 16] a nonlinearity called *swish* was introduced that when used as a drop-in replacement for ReLU, that significantly improves the accuracy of neural networks. The nonlinearity is defined as

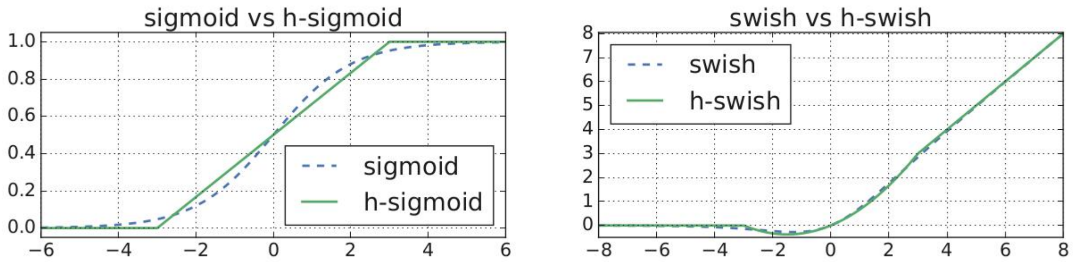
$$\text{swish}\, x = x \cdot \sigma(x)$$

计算、求导复杂，
对量化过程不友好

While this nonlinearity improves accuracy, it comes with non-zero cost in embedded environments as the sigmoid function is much more expensive to compute on mobile devices. We deal with this problem in two ways.

1. We replace sigmoid function with its piece-wise linear hard analog: $\frac{\text{ReLU6}(x+3)}{6}$ similar to [11, 44]. The minor difference is we use ReLU6 rather than a custom clipping constant. Similarly, the hard version of swish becomes

$$\text{h-swish}[x] = x\frac{\text{ReLU6}(x+3)}{6}$$



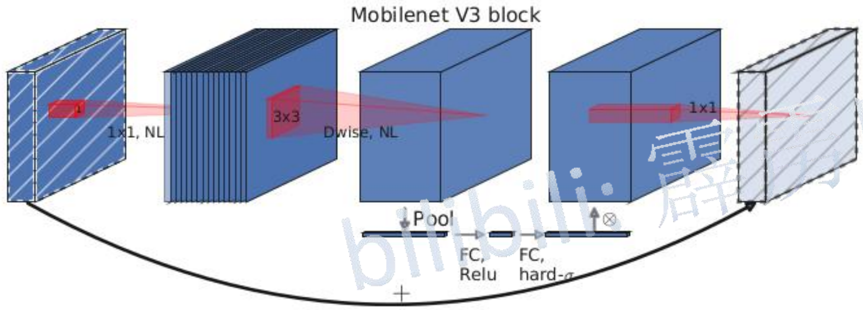
$$\sigma(x) = \frac{1}{1+e^{-x}}$$

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

$$\text{h-sigmoid} = \frac{\text{ReLU6}(x+3)}{6}$$

MobileNet V3

MobileNetV3-Large



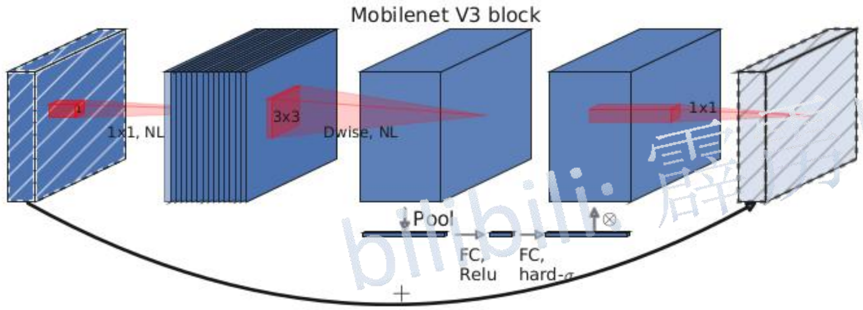
当**stride == 1**且
input_c == output_c
才有shortcut连接

注意：

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

MobileNet V3

MobileNetV3-Small



Input	Operator	exp size	$\#out$	SE	NL	s
$224^2 \times 3$	conv2d, 3x3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	✓	RE	2
$56^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5x5	96	40	✓	HS	2
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	120	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	144	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	288	96	✓	HS	2
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	conv2d, 1x1	-	576	✓	HS	1
$7^2 \times 576$	pool, 7x7	-	-	-	-	1
$1^2 \times 576$	conv2d 1x1, NBN	-	1024	-	HS	1
$1^2 \times 1024$	conv2d 1x1, NBN	-	k	-	-	1

当**stride == 1**且
input_c == output_c
才有shortcut连接

沟通方式

1.github

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

2.bilibili

https://space.bilibili.com/18161609/channel/index

3.CSDN

https://blog.csdn.net/qq_37541097/article/details/103482003