背景

很多改进网络的做法都是从网络的深度、网络的宽度以及图像的分辨率这三个方面入手（如图 1 所示），提高准确率，像 VGG、ResNet 等图像分类网络都是将输入的图像尺寸固定为 224×224，加深网络的深度来提高性能。虽然可以任意缩放两个或三个维度，但任意缩放需要繁琐的手动调优，并且通常会产生精度和效率不是最优的。而且增加网络的深度虽说可以得到更加丰富的特征，但是随着网络深度的加深会出现梯度消失、训练困难的问题。增加网络的宽度可以获得更高细粒度的特征，但是对于宽度很宽，深度较浅的网络很难提取到更具有判别性的特征。提高输入图像分辨率也可以获取到更高细粒度的特征，但是过高的分辨会增加计算量。从图 2 可以看出，这三个参数增加到一定程度其产生的准确率增益会减小

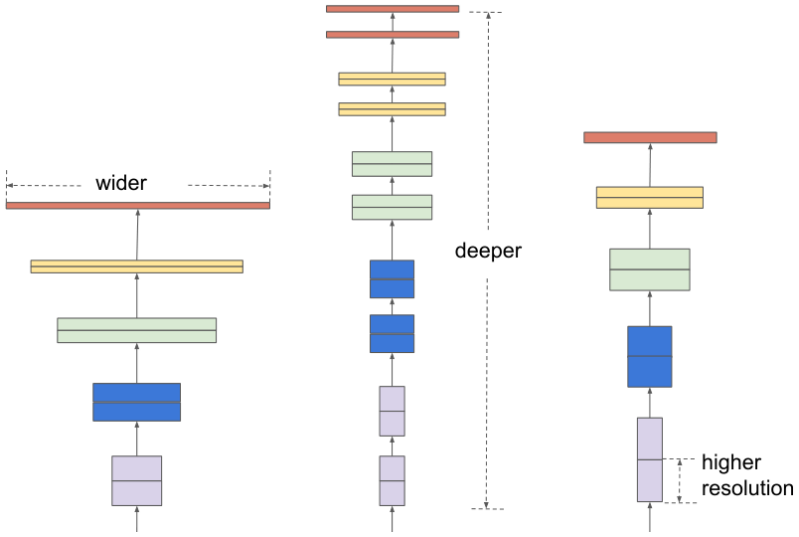

图 1

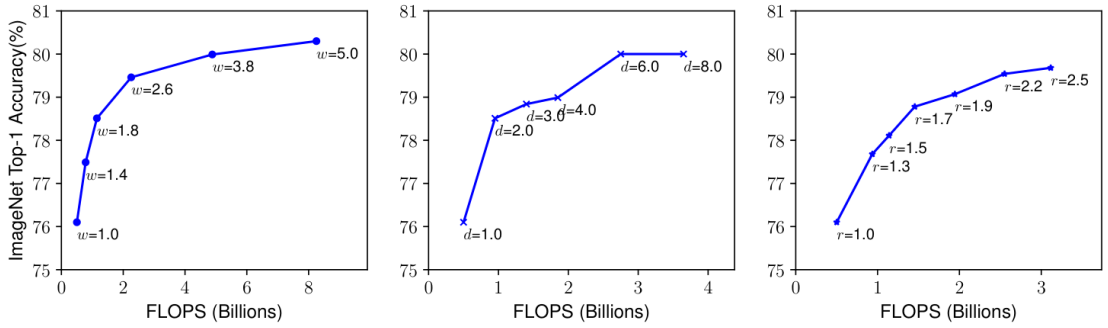

图 2

方法

这篇文章认为平衡网络的深度、网络的宽度以及图像的分辨率很重要，而且这种平衡可以通过简单地用常数比例缩放它们来实现。在此基础上，提出了一种简单有效的复合标度方法。与传统的任意缩放这些因素的做法不同，这里使用一组固定的缩放系数统一缩放网络的深度、网络的宽度以及图像的分辨率。

通过实证研究发现，不同尺度之间并不是相互独立的。比如说：

对于更高分辨率的图像，增加网络深度，这样更大的接受域可以帮助捕获在更大的图像中包含更多像素的相似特征；

当分辨率较高时，我们也应该增加网络宽度，以便在高分辨率图像中捕获更多像素的更细粒度的模式。

因此需要平衡不同的缩放维度，而不是传统的一维缩放。

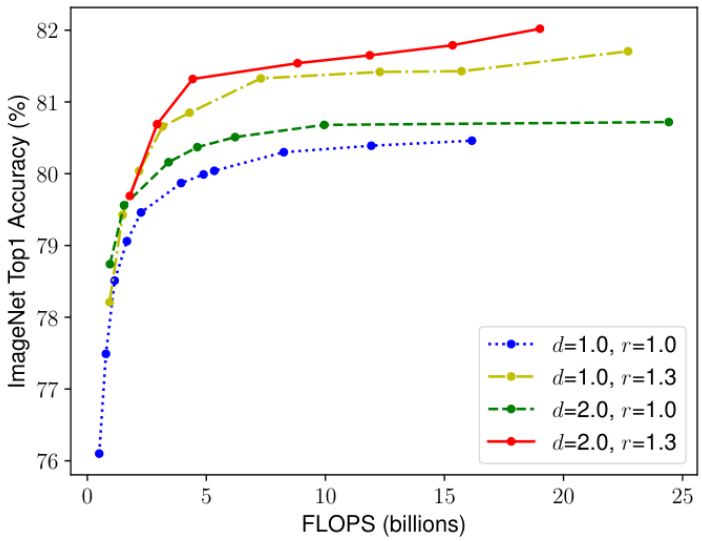

图 3

图 3 是采用不同深度和分辨率组合，然后改变网络的宽度得到的实验结果，可以发现在相同的 FLOPs 下，同时增加网络的深度和分辨率效果最好。

问题公式化

作者先将卷积层 i 定义为一个函数：$Y_i = F_i(X_i)$，而一个卷积网络是用 duo 个卷积层堆叠而成的，那么卷积网络就可以表示为 $N = F_k \odot \cdots \odot F_1(X_1)$，卷积网络通常可以划分为多个阶段，每个阶段中的所有层都共享相同的架构，因此每个

阶段中的所有层都具有相同的卷积类型，所以我们可以将卷积网络定义为 $N = \bigodot_{i=1,\ldots,s} F_i^{L_i}(X)$，其中 $F_i^{L_i}$ 表示在 stage 中 $F_i(X_i)$ 运算被重复执行 L_i 次。为了探究深度、宽度、分辨率对网络性能的影响，将深度(d)、宽度(w)、分辨率(r)带入公式中，将最优化问题抽象为了公式(1)：

$$\begin{aligned} \max_{d,w,r} \quad & Accuracy\big(\mathcal{N}(d,w,r)\big) \\ s.t. \quad & \mathcal{N}(d,w,r) = \bigodot_{i=1\ldots s} \hat{\mathcal{F}}_i^{d\cdot\hat{L}_i}\big(X_{\langle r\cdot\hat{H}_i, r\cdot\hat{W}_i, w\cdot\hat{C}_i\rangle}\big) \\ & \text{Memory}(\mathcal{N}) \le \text{target\_memory} \\ & \text{FLOPS}(\mathcal{N}) \le \text{target\_flops} \end{aligned} \tag{1}$$

d 用来缩放深度 $\hat{L}_i$，r 用来缩放分辨率，即直接对 $\hat{H}_i$、$\hat{W}_i$ 产生影响，w 用来缩放特征矩阵的通道数 $\hat{C}_i$，target_memory 和 target_flops 为对内存和 FLOPs 的限制。

混合缩放法

混合缩放方法是利用混合因子 ϕ 对网络宽度、深度和分辨率进行统一缩放，具体的计算公式如下：

$$\begin{aligned} \text{depth: } & d = \alpha^\phi \\ \text{width: } & w = \beta^\phi \\ \text{resolution: } & r = \gamma^\phi \\ \text{s.t. } & \alpha\cdot\beta^2\cdot\gamma^2 \approx 2 \\ & \alpha \ge 1, \beta \ge 1, \gamma \ge 1 \end{aligned} \tag{2}$$

其中 α，β，γ 分别指定如何将这些额外的资源分配给网络宽度、深度和分辨率。假设网络的深度增加一倍，那么 FLOPs 也会增加一倍；网络的深度增加一倍，且假设输入输出的特征图尺寸不变，那么因为网络深度增加了一倍，输入输出特征图的通道数也会增加一倍，那么 FLOPs 也会增加四倍，相当于网络增加一倍，FLOPs 增加平方倍；输入图像的分辨率增加一倍，那么 FLOPs 增加四倍。总的 FLOPs 可以近似用 $(\alpha\beta^2\gamma^2)^\phi$ 表示。

网络结构

使用 NAS 搜索技术得到 EfficientNet-B0 的结构，其结构图如下图所示：

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

图 4

可以看到网络的 Stage2~ Stage8 是由 MBConv 堆叠得到的，MBConv1、MBConv6 指的是 MBConv 中第一个 1×1 卷积层将输入特征图的通道数扩充 1 或者 6 倍，k3×3、k5×5 表示 MBConv 中深度方向卷积所采用的卷积核的大小。这里的 MBConv 就是 MobileNetV3 中使用的反向残差块。

然后又在 EfficientNet-B0 的基础上使用 NAS 搜索 α，β，γ 这三个参数：

首先固定混合因子 ϕ，然后使用公式(1)和公式(2)进行搜索，发现当 $\alpha=1.2$，$\beta=1.1$，$\gamma=1.15$ 时 EfficientNet-B0 的性能最佳；然后固定刚刚得到的三个参数，改变混合因子 ϕ 得到 EfficientNet-B1~ EfficientNet-B7。

背景

EfficientNetV1 关注的时准确率、参数数量以及 FLOPs，而在 EfficientNetV2 中关注的则是模型的训练速度，除此之外针对 EfficientNetV1 中存在的问题：

①训练图像的尺寸很大时，训练速度很慢

②在浅层网络中使用深度方向的卷积速度会很慢

③同等放大每个 Stage 中的深度和宽度得到的结果不是最优的

做出了改进。

方法

针对训练图像分辨率大引起的训练速度慢做出的改变

从下图可以看出当训练图像的尺寸为 380 时增加 batch_size 大小可以提高训练速度，但是当图像分辨率变大时，训练速度下降了，这时再使用较大的 batch_size 的话显存就不够了。那么针对这一问题最直接的方法就是降低训练图像的尺寸，不仅加快了训练速度，而且还能使用更大的 batch_size，没有必要像 EfficientNetV1 执着地增加分辨率。

		TPUv3 imgs/sec/core		V100 imgs/sec/gpu	
	Top-1 Acc.	batch=32	batch=128	batch=12	batch=24
train size=512	84.3%	42	OOM	29	OOM
train size=380	84.6%	76	93	37	52

图 5

Fused-MBConv

虽然说理论上使用深度方向的卷积计算量会很小，但是在实际中使用的时候因为无法充分利用现有的加速器，所以速度较慢，因此引入了 Fused-MBConv 结构。

图 6 是 MBConv 和 Fused-MBConv 的结构示意图。可以看出来 Fused-MBConv 是直接将 MBConv 中第一个 1×1 的卷积层和深度方向的卷积直接换为一个 3×3 的卷积层。它们的实验效果如图 7 所示，使用 MBConv 时其准确率可以达到 82.8%，速度是每块 GPU 每秒训练 155 张图片，使用 Fused-MBConv 将原来

EfficientNetV1 中 stage2~4 的卷积模块换为 Fused-MBConv 可以发现准确率和训练速度都有所提升。

将原来 EfficientNetV1 中 stage2~6 的卷积模块或者将所有的 stage 换为 Fused-MBConv，通过图 7 的对比可以发现，还是将 stage2~4 的 MBConv 进行替换是最优的。随后作者也使用了 NAS 搜索技术去寻找将哪些 stage 的卷积模块进行替换道道的效果是最好的。

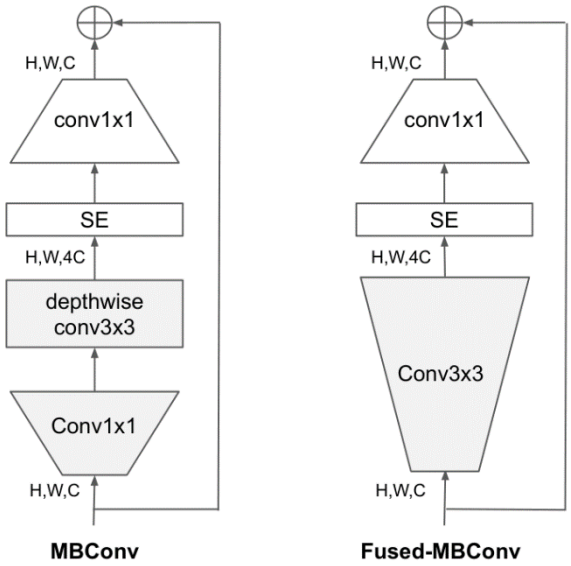

图 6

	Params (M)	FLOPs (B)	Top-1 Acc.	TPU imgs/sec/core	V100 imgs/sec/gpu
No fused	19.3	4.5	82.8%	262	155
Fused stage1-3	20.0	7.5	83.1%	362	216
Fused stage1-5	43.4	21.3	83.1%	327	223
Fused stage1-7	132.0	34.4	81.7%	254	206

图 7

采用非均匀缩放

EfficientNetV1 中同等地放大了每个 Stage 中的深度和宽度，由于每个 Stage 对于网络训练速度以及参数数量的贡献不是均等的，因此将其均匀缩放所得到的结果不是最优的，所以在这里，采用非均匀缩放的方式来进行模型深度和宽度的缩放调整，逐渐地添加更多的层到后面的几个阶段，并且修改了缩放策略，将图

像的最大尺寸限制为较小的值。

Training-aware NAS 搜索

因为现在有很多模型设计的选择，现在采用的不是原来的 NAS 搜索技术，而是基于 NAS 结合了新的优化对象，例如准确率、参数有效性和训练效率三个维度，模型设计的空间包含了卷积模块、层数、卷积核尺寸以及卷积模块中第一个卷积涉及到的扩展率。

EfficientNetV2 结构

网络结构框图如下所示：

Stage	Operator	Stride	#Channels	#Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	256	15
7	Conv1x1 & Pooling & FC	-	1280	1

图 8

除了在较浅的层使用 Fused-MBConv 外，使用的卷积模块的扩展率不同于 V1 中的 1 或 6，选择使用较小的扩展率，可以减少内存访问成本。而且不同于 V1 中使用了 3×3 和较多的 5×5 卷积核，V2 选择全部使用较小的 3×3 卷积核，又由于卷积核尺寸的减小，也就意味着感受野变小，所以每一个卷积模块堆叠的次数也变多了。

渐进学习策略

由于训练图像的分辨率对网络的训练速度有很大的影响，有人做实验发现在网络训练的一开始使用较小的图像分辨率，然后再加大分辨率，这样得到的准确率通常会下降，那么在这里，作者猜测，准确率的降低时不平衡的正则化导致的。

接下来使用前面提到的 training-awareNAS 搜索技术，在该搜索空间中采样并训练模型，在训练过程中使用不同的图像分辨率以及不同的数据增强手段。

	Size=128	Size=192	Size=300
RandAug magnitude=5	**78.3** ±**0.16**	81.2 ±0.06	82.5 ±0.05
RandAug magnitude=10	78.0 ±0.08	**81.6** ±**0.08**	82.7 ±0.08
RandAug magnitude=15	77.7 ±0.15	81.5 ±0.05	**83.2** ±**0.09**

图 9

图 9 的红色框可以看出，当使用很少的图像增强手段，且图片分辨率较小的时候，网络的效果时最好的。从蓝色的框中可以看出，当选用较大的图像分辨率时，使用更多的数据增强方法达到下效果是最好的。

基于这些实验结果，提出了渐进学习策略（如图 10 所示），也就是在训练早期阶段使用较小的图像分辨率以及弱正则化方法，让网络学习到一些简单的特征表达，然后随着训练进程的加深，加大图像的分辨率的同时也增强正则化手段。

图 10