EfficientNet：再论 CNN 的网络规模

1、概述

一般而言，扩展 CNN 网络规模会得到一定的性能提升，网络模型的扩展主要体现在三个方面上：

- 网络宽度：即在每层的输出通道上进行扩展，增加更多的 filter
- 网络深度：这个很好理解，可以将一些层进行堆叠扩展深度
- 分别率：如 AlexNet 的输入大小为 224×224，Inception-V3 是 299×299，以及更大的感受野

理论上，我们可以在这三个方面任意的进行扩展，然而如果采用手工调参，不仅费时费力，而且得到的结果往往是次优的。所以这篇论文的要解决的一个根本问题是：Is there a principled method to scale up ConvNets that can achieve better accuracy and efficiency?

通过实证研究表明，平衡网络的深度/宽度/分辨率是很重要的，而且这种均衡可以很简单的利用恒定比率去缩放不同的维度实现。

具体的搜索方式采用 neural architecture search (NAS) 方式去搜索，简单的说就是利用一个学习器主动去学习最佳的 CNN 的架构是什么样子的，而不再是像传统网络需要我们人工设计复杂的网络架构。主要涉及到三个方面内容：搜索空间、搜索算法以及搜索目标。

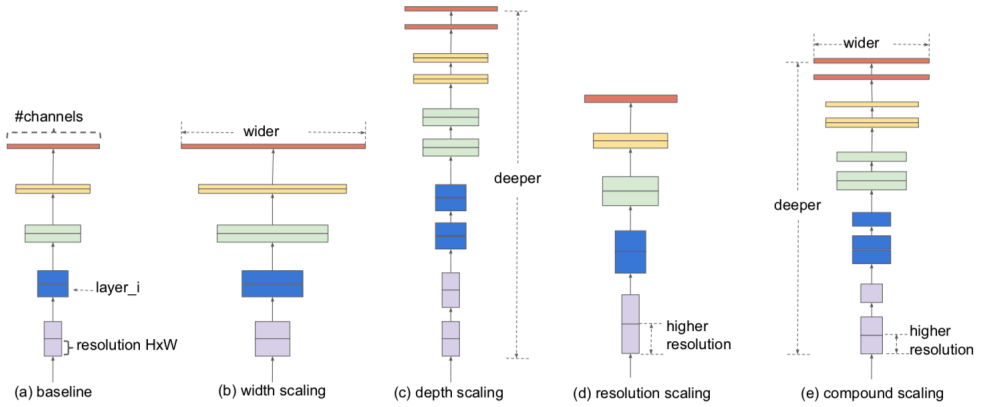

Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

上图演示了不同维度层面的架构扩展示意图。

2、模型缩放详解

2.1 问题是什么

对于一个层 i，它做的一件事就是 $Y_i = \mathcal{F}_i(\mathcal{X}_i)$，这里的 $\mathcal{F}_i$ 代表这层所进行的计算操作，X_i 和 Y_i 分别代表这层的输入和输出，X_i 通常可以表示为 $\langle H_i, W_i, C_i \rangle$（忽略批量维度，$H$ 和 W 表示空间信息，即二维的图像特征，C 是通道数）。

所以整个 CNN 可以描述为 $\mathcal{N} = \mathcal{F}_k \bigodot \cdots \bigodot \mathcal{F}_2 \bigodot \mathcal{F}_1(\mathcal{X}_1) = \bigodot_{j=1 \cdots k} \mathcal{F}_j(\mathcal{X}_1)$

由于网络架构中通常存在一些重复堆叠的层，我们可以简单的将其划分为不同的块，例如 ResNet 有 5 个块，每个块内部的所有层有相同的卷积形式。所以可以将公式改写为：

$N = \bigodot_{i=1 \cdots s} F_i^{L_i}(X_{\langle H_i, W_i, C_i \rangle})$

这里的 $F_i^{L_i}$ 表示第 i 个块内的 F_i 层重复堆叠 L_i 次。

通过公式我们发现，我们的扩展是基于深度（L_i）和宽度（C_i） 以及分辨率（H_i, W_i）进行扩展的，与 F_i 无关，所以只要定义一个基线网络（F_i），我们基于这个基线网络就可以做扩展，因此寻找一个良好的基线网络也是非常重要的。

虽然，我们不修改 F_i，这样大大缩小了我们的搜索空间，但是 L_i, C_i, H_i, W_i 的选择范围依然很大。为了进一步缩小搜索范围，这篇论文规定所有的层的缩放需要有一个一致的缩放比率。

整个问题演变为下面的优化问题：

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

这里的 w, d, r 是网络的宽度、深度、分辨率缩放系数，$\hat{F}_i, \hat{L}_i, \hat{H}_i, \hat{W}_i, \hat{C}_i$ 都是基线网络的参数。这里的基线网络是参照 MNASNet 网络利用强化学习搜索出来的，可以简单认为这里的基线网络就是 MNASNet，只是在搜索目标上，由于 MNASNet 是采用真实的手机硬件测试的模型推理延迟时间，所以在上面公式中，对于 MNASNet 而言，他们的 FLOPS 是真的拿到了手机上测试的延迟时间，而 EfficientNet 不针对特定硬件，因此搜索目标中仍然采用 FLOPS。

> FLOPS 是个什么玩意儿?
>
> 是每秒钟的浮点数计算量，简单的理解是你的网络计算量大小。计算量越大，在预测阶段的耗时越长，所以理想目标是降计算量同时保证精度，这样一来上面的公式就很好理解了。求最高的精度，同时模型的参数量（内存占用）及计算量都要比给定的目标值要小。

2.2 不同维度方面的扩展

2.2.1 深度

加深网络的深度通常可以取得更高的预测精度，然而，更深的网络由于存在梯度消失问题更不易训练，尽管存在一些诸如跳跃连接以及批量正则等方式可以缓解一些，但是更深的架构获利不大，例如 ResNet-1000 和 ResNet-101 差不太多。

2.2.2 宽度

宽度的扩展一般用于小规模的网络上，宽度的扩展一般能够捕获更细粒度的特征（因为宽度扩展意味着使用了更多的特征提取器），也更加容易训练，但是一些深度较浅的网络很难去捕获高级特征（高级特征一般需要在更高层组合出，似乎也说明了宽度的扩展需要深度维度加深辅助）

2.2.3 分辨率

这个毫无疑问，在早期的架构中一般使用 224×224（如 AlexNet），后期使用 299×299（如Inception-V3），或 331×331 （如 NASNet、PNASNet等）、以及 GPipe 中使用 480×480。增加分辨率确实可以带来性能提升，但是需要非常高的分辨率。

下图展示了三个不同维度下单独测试结果：

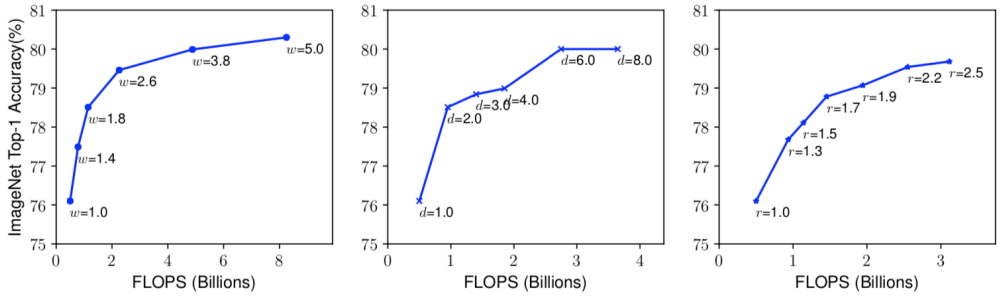

Figure 3. **Scaling Up a Baseline Model with Different Network Width (w), Depth (d), and Resolution (r) Coefficients.** Bigger networks with larger width, depth, or resolution tend to achieve higher accuracy, but the accuracy gain quickly saturate after reaching 80%, demonstrating the limitation of single dimension scaling. Baseline network is described in Table 1.

根据上图，我们得到：

- 增大网络宽度、深度或分辨率的任何维度都会提高一定的精度，但对于较大的模型，精度增益会减少，这三个图增长到一定程度都开始变得平缓
- 为了追求更好地精度和效率，在 ConvNet 缩放过程中要均衡所有的维度，这一点很重要。

2.3 混合扩展策略

因为在深度层面，将一个网络架构翻倍，相当于计算两遍，这是一个线性增长的速度，而宽度和分辨率则是以平方速度增长的。

> 宽度上主要体现在 C 上，例如原始网络的通道数是 $3 \rightarrow 4 \rightarrow 8 \rightarrow 16$ 扩展2倍现在变成了 $3 \rightarrow 8 \rightarrow 16 \rightarrow 32$ 计算量增长了基本是 4 倍（除了输入层到第一隐层是2倍）

论文中分别用 α, β, γ 表示深度、宽度、分辨率上的维度，而且希望 $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$，这样可以保证各个维度上的扩展基本是一致的，这里的 2 没有什么特别的，因为整体的扩展是通过 $(\alpha \cdot \beta^2 \cdot \gamma^2)^{\phi}$ 定义的，相当于我们的模型缩放尺度永远是 2^{ϕ} 倍。通过小的网络搜索技术求解上面的值得 $\alpha = 1.2, \beta = 1.1, \gamma = 1.15$

通过控制 ϕ 的值，会基于基线模型扩展不同的大小，分别搜索了 B1-B7 7个不同尺度的模型。

Table 2. **EfficientNet Performance Results on ImageNet** (Russakovsky et al., 2015). All EfficientNet models are scaled from our baseline EfficientNet-B0 using different compound coefficient ϕ in Equation 3. ConvNets with similar top-1/top-5 accuracy are grouped together for efficiency comparison. Our scaled EfficientNet models consistently reduce parameters and FLOPS by an order of magnitude (up to 8.4x parameter reduction and up to 16x FLOPS reduction) than existing ConvNets.

Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPS	Ratio-to-EfficientNet
EfficientNet-B0	**76.3%**	**93.2%**	**5.3M**	**1x**	**0.39B**	**1x**
ResNet-50 (He et al., 2016)	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169 (Huang et al., 2017)	76.2%	93.2%	14M	2.6x	3.5B	8.9x
EfficientNet-B1	**78.8%**	**94.4%**	**7.8M**	**1x**	**0.70B**	**1x**
ResNet-152 (He et al., 2016)	77.8%	93.8%	60M	7.6x	11B	16x
DenseNet-264 (Huang et al., 2017)	77.9%	93.9%	34M	4.3x	6.0B	8.6x
Inception-v3 (Szegedy et al., 2016)	78.8%	94.4%	24M	3.0x	5.7B	8.1x
Xception (Chollet, 2017)	79.0%	94.5%	23M	3.0x	8.4B	12x
EfficientNet-B2	**79.8%**	**94.9%**	**9.2M**	**1x**	**1.0B**	**1x**
Inception-v4 (Szegedy et al., 2017)	80.0%	95.0%	48M	5.2x	13B	13x
Inception-resnet-v2 (Szegedy et al., 2017)	80.1%	95.1%	56M	6.1x	13B	13x
EfficientNet-B3	**81.1%**	**95.5%**	**12M**	**1x**	**1.8B**	**1x**
ResNeXt-101 (Xie et al., 2017)	80.9%	95.6%	84M	7.0x	32B	18x
PolyNet (Zhang et al., 2017)	81.3%	95.8%	92M	7.7x	35B	19x
EfficientNet-B4	**82.6%**	**96.3%**	**19M**	**1x**	**4.2B**	**1x**
SENet (Hu et al., 2018)	82.7%	96.2%	146M	7.7x	42B	10x
NASNet-A (Zoph et al., 2018)	82.7%	96.2%	89M	4.7x	24B	5.7x
AmoebaNet-A (Real et al., 2019)	82.8%	96.1%	87M	4.6x	23B	5.5x
PNASNet (Liu et al., 2018)	82.9%	96.2%	86M	4.5x	23B	6.0x
EfficientNet-B5	**83.3%**	**96.7%**	**30M**	**1x**	**9.9B**	**1x**
AmoebaNet-C (Cubuk et al., 2019)	83.5%	96.5%	155M	5.2x	41B	4.1x
EfficientNet-B6	**84.0%**	**96.9%**	**43M**	**1x**	**19B**	**1x**
EfficientNet-B7	**84.4%**	**97.1%**	**66M**	**1x**	**37B**	**1x**
GPipe (Huang et al., 2018)	84.3%	97.0%	557M	8.4x	-	-

We omit ensemble and multi-crop models (Hu et al., 2018), or models pretrained on 3.5B Instagram images (Mahajan et al., 2018).

上表是 7 个不同尺度的模型与其他模型的比较结果，还是很厉害的样子。作者也尝试基于其他现有的模型做规模扩展，下图是在 MobileNets 和 ResNet 扩展结果。

Table 3. **Scaling Up MobileNets and ResNet.**

Model	FLOPS	Top-1 Acc.
Baseline MobileNetV1 (Howard et al., 2017)	0.6B	70.6%
Scale MobileNetV1 by width (w=2)	2.2B	74.2%
Scale MobileNetV1 by resolution (r=2)	2.2B	72.7%
compound scale (d=1.4, w=1.2, r=1.3)	**2.3B**	**75.6%**
Baseline MobileNetV2 (Sandler et al., 2018)	0.3B	72.0%
Scale MobileNetV2 by depth (d=4)	1.2B	76.8%
Scale MobileNetV2 by width (w=2)	1.1B	76.4%
Scale MobileNetV2 by resolution (r=2)	1.2B	74.8%
MobileNetV2 compound scale	**1.3B**	**77.4%**
Baseline ResNet-50 (He et al., 2016)	4.1B	76.0%
Scale ResNet-50 by depth (d=4)	16.2B	78.1%
Scale ResNet-50 by width (w=2)	14.7B	77.7%
Scale ResNet-50 by resolution (r=2)	16.4B	77.5%
ResNet-50 compound scale	**16.7B**	**78.8%**

下图是各类型模型在 ImageNet 上的对比图，可以看到 B4 在差不多 5B 的计算量下基本与23B左右的NASNet-A 齐平，相对来说性价比最高了。

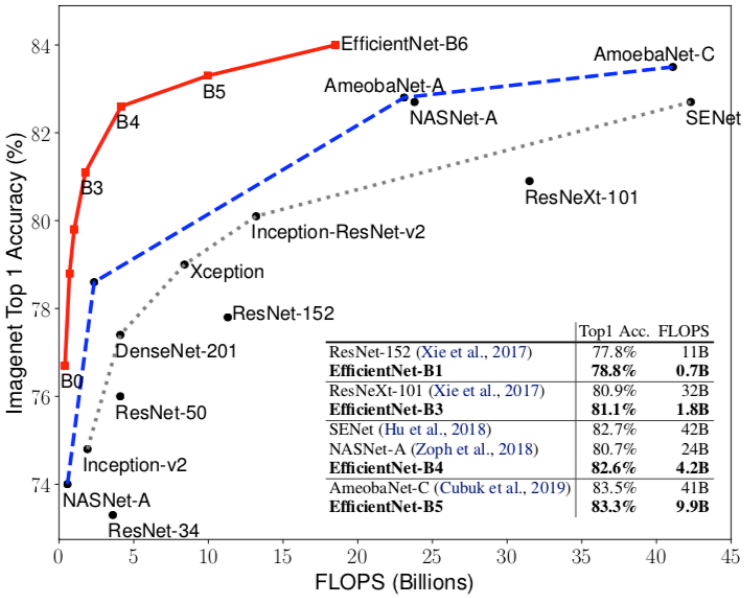

Figure 5. **FLOPS vs. ImageNet Accuracy –** Similar to Figure 1 except it compares FLOPS rather than model size.