EfficientNet: Repensando o dimensionamento de modelos para redes neurais convolucionais

Bronzeado Mingxing [1] **Quoc V. Le** [1]

Abstrato

Redes Neurais Convolucionais (ConvNets) são comumente desenvolvido com um orçamento de recursos fixo, e depois ampliado para melhor precisão se mais recursos estão disponíveis. Neste artigo, estudamos sistematicamente o escalonamento do modelo e identificamos que equilibrar cuidadosamente a profundidade, largura e resolução da rede pode levar a um melhor desempenho. Baseado nesta observação, propomos uma nova escala método que dimensiona uniformemente todas as dimensões do profundidade/largura/resolução usando um método simples, mas altamente coeficiente composto efetivo. Nós demonstramos a eficácia deste método na ampliação MobileNets e ResNet.

Para ir ainda mais longe, usamos a pesquisa de arquitetura neural para projetar uma nova rede de linha de base e ampliá-lo para obter uma família de modelos, chamadas EfficientNets, que alcançam muito melhor precisão e eficiência do que anteriores ConvNets. Em particular, nosso EfficientNet-B7 alcança precisão de última geração de 84,3% top-1 no ImageNet, sendo **8,4x menor** e **6,1x mais rápido** na inferência do que o melhor existente ConvNet. Nossos EfficientNets também transferem bem e obtenha precisão de última geração no CIFAR-100 (91,7%), Flores (98,8%) e 3 outras transferências conjuntos de dados de aprendizagem, com uma ordem de magnitude menos parâmetros. O código fonte está em https://github.com/tensorflow/tpu/tree/master/modelos/oficial/eficientenet.

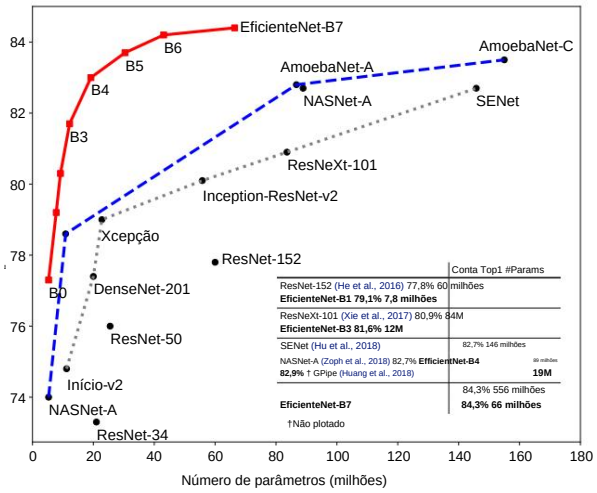

Figura 1. **Tamanho do modelo versus precisão do ImageNet.** Todos os números são para monocultura, modelo único. Nossas EfficientNets superam significativamente outras ConvNets. Em particular, o EfficientNet-B7 alcança nova precisão de última geração de 84,3% top-1, mas sendo 8,4x menor e 6,1x mais rápido que o GPipe. EfficientNet-B1 é 7,6x menor e 5,7x mais rápido que o ResNet-152. Os detalhes estão nas Tabelas 2 e 4.

1. Introdução

A ampliação de ConvNets é amplamente utilizada para obter melhor precisão . Por exemplo, ResNet (He et al., 2016) pode ser dimensionado passando de ResNet-18 para ResNet-200 usando mais camadas; Recentemente, o GPipe (Huang et al., 2018) alcançou 84,3% de precisão do ImageNet top-1 ao ampliar um modelo de linha de base quatro tempo maior. No entanto, o processo de ampliação de ConvNets nunca foi bem compreendido e atualmente existem muitos maneiras de fazer isso. A maneira mais comum é aumentar as ConvNets por sua profundidade (He et al., 2016) ou largura (Zagoruyko & Komodakis, 2016). Outro menos comum, mas cada vez mais popular, o método é ampliar modelos por resolução de imagem (Huang et al., 2018). Em trabalhos anteriores, é comum dimensionar apenas uma das três dimensões – profundidade, largura e imagem tamanho. Embora seja possível dimensionar duas ou três dimensões arbitrariamente, o dimensionamento arbitrário requer ajuste manual tedioso e ainda muitas vezes produz precisão e eficiência abaixo do ideal.

Neste artigo, queremos estudar e repensar o processo de ampliar ConvNets. Em particular, investigamos o questão central: existe um método de princípios para ampliar ConvNets que podem alcançar melhor precisão e eficiência? Nosso estudo empírico mostra que é fundamental equilibrar todos dimensões de largura/profundidade/resolução da rede e, surpreendentemente, esse equilíbrio pode ser alcançado simplesmente dimensionando cada deles com razão constante. Com base nesta observação, podemos propor um método de escalonamento composto simples, mas eficaz. Ao contrário da prática convencional que dimensiona arbitrariamente esses fatores , nosso método dimensiona uniformemente a largura, profundidade e profundidade da rede.

1Google Research, Brain Team, Mountain View, CA. Correspondência para: Mingxing Tan <tanmingxing@google.com>.

Anais da 36ª Conferência Internacional sobre Máquinas Aprendizagem, Long Beach, Califórnia, PMLR 97, 2019.

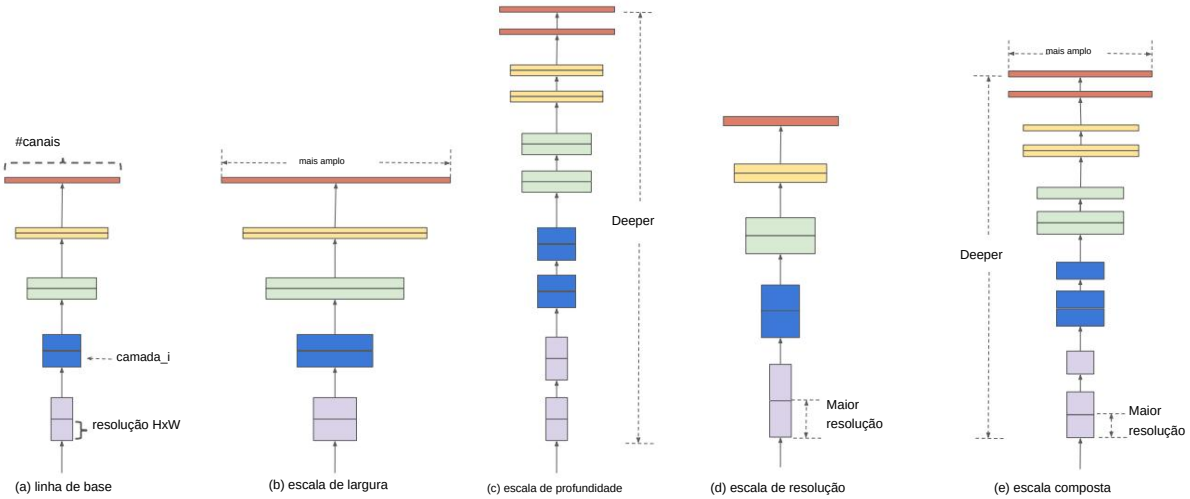

Figura 2. **Dimensionamento do modelo.** (a) é um exemplo de rede de referência; (b)-(d) são escalonamentos convencionais que aumentam apenas uma dimensão da rede largura, profundidade ou resolução. (e) é o nosso método de escala composto proposto que escala uniformemente todas as três dimensões com uma proporção fixa.

e resolução com um conjunto de coeficientes de escala fixos. Para por exemplo, se quisermos usar 2 ^{N} vezes mais computacional recursos, então podemos simplesmente aumentar a profundidade da rede em ^{Nÿ}, largura por ÿ ^{N} , e tamanho da imagem por ÿ ^{N} , onde ÿ, ÿ, ÿ são coeficientes constantes determinados por uma pequena pesquisa em grade em o pequeno modelo original. A Figura 2 ilustra a diferença entre nosso método de escalonamento e métodos convencionais.

Intuitivamente, o método de escala composta faz sentido porque se a imagem de entrada for maior, então a rede precisa mais camadas para aumentar o campo receptivo e mais canais para capturar padrões mais refinados na imagem maior. Em fato, teórico anterior (Raghu et al., 2017; Lu et al., 2018) e resultados empíricos (Zagoruyko & Komodakis, 2016) ambos mostrar que existe certa relação entre rede largura e profundidade, mas até onde sabemos, somos o primeiro a quantificar empiricamente a relação entre todos os três dimensões de largura, profundidade e resolução da rede.

Demonstramos que nosso método de escalonamento funciona bem em MobileNets existentes (Howard et al., 2017; Sandler et al., 2018) e ResNet (He et al., 2016). Notavelmente, a eficácia o dimensionamento do modelo depende muito da rede de linha de base; para ir ainda mais longe, usamos a pesquisa de arquitetura neural (Zoph & Le, 2017; Tan et al., 2019) para desenvolver uma nova linha de base rede e ampliá-la para obter uma família de modelos, chamada Redes Eficientes. A Figura 1 resume o desempenho do ImageNet, onde nossos EfficientNets superam significativamente outras ConvNets. Em particular, o nosso EfficientNet-B7 supera a melhor precisão GPipe existente (Huang et al., 2018), mas usando 8,4x menos parâmetros e executando 6,1x mais rápido na inferência . Em comparação com o ResNet-50 amplamente utilizado (He et al., 2016), nosso EfficientNet-B4 melhora a precisão top-1 de 76,3% para 83,0% (+6,7%) com FLOPS semelhantes. Além do mais ImageNet, EfficientNets também transferem bem e alcançam estados precisão de última geração em 5 dos 8 conjuntos de dados amplamente utilizados, enquanto reduzindo os parâmetros em até 21x do que os ConvNets existentes.

2. Trabalho relacionado

Precisão ConvNet: Desde AlexNet (Krizhevsky et al., 2012) venceu a competição ImageNet de 2012, ConvNets tornar-se cada vez mais preciso ao aumentar: enquanto o vencedor do ImageNet de 2014, GoogleNet (Szegedy et al., 2015) atinge 74,8% de precisão top-1 com cerca de 6,8 milhões de parâmetros, o vencedor do ImageNet de 2017, SENet (Hu et al., 2018) alcança 82,7% de precisão top-1 com parâmetros de 145 milhões. Recentemente, GPipe (Huang et al., 2018) impulsiona ainda mais o estado da arte Precisão de validação top-1 do ImageNet para 84,3% usando 557M parâmetros: é tão grande que só pode ser treinado com um biblioteca especializada de paralelismo de pipeline, particionando o rede e espalhando cada parte para um acelerador diferente . Embora esses modelos sejam projetados principalmente para ImageNet, estudos recentes mostraram que modelos ImageNet melhores também apresentam melhor desempenho em uma variedade de conjuntos de dados de aprendizagem por transferência. (Kornblith et al., 2019) e outras tarefas de visão computacional como detecção de objetos (He et al., 2016; Tan et al., 2019). Embora maior precisão seja crítica para muitas aplicações, já atingimos o limite de memória do hardware e, portanto, maior ganho de precisão precisa de melhor eficiência.

Eficiência ConvNet: ConvNets profundos costumam ser superparametrizados. Compressão de modelo (Han et al., 2016; He e outros, 2018; Yang et al., 2018) é uma forma comum de reduzir o tamanho do modelo, trocando precisão por eficiência. À medida que os telefones celulares se tornam onipresentes, também é comum criar ConvNets eficientes de tamanho móvel, como SqueezeNets. (Iandola et al., 2016; Gholami et al., 2018), MobileNets (Howard et al., 2017; Sandler et al., 2018) e ShuffleNets

(Zhang et al., 2018; Ma et al., 2018). Recentemente, a pesquisa de arquitetura neural tornou-se cada vez mais popular no projeto de ConvNets móveis eficientes (Tan et al., 2019; Cai et al., 2019) e alcança uma eficiência ainda melhor do que ConvNets móveis feitos à mão, ajustando extensivamente a largura da rede. , profundidade, tipos e tamanhos de kernel de convolução. No entanto, não está claro como aplicar essas técnicas para modelos maiores que possuem espaço de design muito maior e custos de ajuste muito mais caros . Neste artigo, pretendemos estudar a eficiência do modelo para ConvNets supergrandes que superam a precisão do estado da arte . Para atingir este objetivo, recorremos ao escalonamento do modelo.

Dimensionamento do modelo: Existem muitas maneiras de dimensionar um Con-vNet para diferentes restrições de recursos: ResNet (He et al., 2016) pode ser reduzido (por exemplo, ResNet-18) ou aumentado (por exemplo, ResNet-200) ajustando a rede profundidade (#layers), enquanto WideResNet (Zagoruyko & Komodakis, 2016) e Mo-bileNets (Howard et al., 2017) podem ser dimensionados pela largura da rede (#channels). Também é reconhecido que um tamanho maior da imagem de entrada ajudará na precisão com a sobrecarga de mais FLOPS. Embora estudos anteriores (Raghu et al., 2017; Lin & Jegelka, 2018; Sharir & Shashua, 2018; Lu et al., 2018) tenham mostrado que a profundidade e a largura da rede são importantes para o poder expressivo dos ConvNets, ainda permanece um questão em aberto sobre como dimensionar efetivamente uma ConvNet para obter melhor eficiência e precisão. Nosso trabalho estuda sistemática e empiricamente o escalonamento da ConvNet para todas as três dimensões de largura, profundidade e resolução da rede.

3. Dimensionamento do modelo composto

Nesta seção, formularemos o problema de escalonamento, estudaremos diferentes abordagens e proporemos nosso novo método de escalonamento.

3.1. Formulação de problema

Uma camada ConvNet i pode ser definida como uma função: Yi = Fi (Xi), onde Fi é o operador, Yi é o tensor de saída, Xi é o tensor de entrada, com forma de tensor Hi onde Hi e Wi são dimensão espacial e Ci é o canal dimensão.[1]

Uma ConvNet N pode ser representada por uma lista de camadas compostas : N = Fk ... F2 F1(X1) = Fj (X1). Na prática, as camadas ConvNet são frequentemente particionadas em múltiplos estágios e todas as camadas em cada estágio compartilham a mesma arquitetura : por exemplo, ResNet (He et al., 2016) tem cinco estágios, e todas as camadas em cada estágio têm a mesma arquitetura convolucional. tipo , exceto que a primeira camada executa a redução da amostragem. Portanto, podemos definir um ConvNet como:

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

onde $F_i^{L_i}$ denota que a camada Fi é repetida Li vezes no estágio i, $\langle H_i, W_i, C_i \rangle$ denota a forma do tensor de entrada X da camada eu. A Figura 2 (a) ilustra um ConvNet representativo, onde a dimensão espacial é gradualmente reduzida, mas a dimensão do canal é expandida ao longo das camadas, por exemplo, do formato de entrada inicial 224, 224, 3 até o formato de saída final 7, 7, 512.

Ao contrário dos projetos ConvNet regulares que se concentram principalmente em encontrar a melhor arquitetura de camada Fi , o escalonamento do modelo tenta expandir o comprimento da rede (Li), a largura (Ci) e/ou a resolução (Hi Wi) sem alterar o Fi predefinido no rede básica . Ao corrigir Fi , o escalonamento do modelo simplifica o problema de design para novas restrições de recursos, mas ainda permanece um grande espaço de design para explorar diferentes $\langle H_i, W_i, C_i \rangle$ Li Wi para cada camada. Para reduzir ainda mais o espaço de design, restringimos que todas as camadas sejam dimensionadas uniformemente com proporção constante. Nosso objetivo é maximizar a precisão do modelo para quaisquer restrições de recursos, o que pode ser formulado como um problema de otimização:

$$\begin{aligned} &\max_{d,w,r} \quad \text{Precisão } N(d, w, r) \\ &\text{st } N(d, w, r) = \bigodot_{i=1...s} \hat{F}_i^{d \cdot \hat{L}_i}(X_{\langle r \cdot \hat{H}_i, r \cdot \hat{W}_i, w \cdot \hat{C}_i \rangle}) \\ &\quad \text{Memória } (N) \ddot{y} \text{ memória alvo} \\ &\quad \text{FLOPS}(N) \ddot{y} \text{ flops alvo} \end{aligned} \quad (2)$$

onde w, d, r são coeficientes para dimensionar largura, profundidade e resolução da rede; Medidores $\hat{F}_i, \hat{L}_i, \hat{H}_i, \hat{C}_i$ são parâmetros na rede de linha de base (ver Tabela 1 como exemplo).

3.2. Dimensionando Dimensões

A principal dificuldade do problema 2 é que os d, w, r ótimos dependem um do outro e os valores mudam sob diferentes restrições de recursos. Devido a esta dificuldade, os métodos convencionais escalam principalmente ConvNets em uma destas dimensões:

Profundidade (d): O dimensionamento da profundidade da rede é a forma mais comum usada por muitos ConvNets (He et al., 2016; Huang et al., 2017; Szegedy et al., 2015; 2016). A intuição é que o ConvNet mais profundo pode capturar recursos mais ricos e complexos e generalizar bem em novas tarefas. No entanto, redes mais profundas também são mais difíceis de treinar devido ao problema do gradiente evanescente (Zagoruyko & Komodakis, 2016). Embora várias técnicas, como pular conexões (He et al., 2016) e normalização de lote (Ioffe & Szegedy, 2015), aliviem o problema de treinamento, o ganho de precisão de redes muito profundas diminui: por exemplo, ResNet-1000 tem precisão semelhante ao ResNet-101, embora tenha muito mais camadas. A Figura 3 (meio) mostra nosso estudo empírico sobre o dimensionamento de um modelo de linha de base com diferentes coeficientes de profundidade d, sugerindo ainda a diminuição do retorno de precisão par

Largura (w): O dimensionamento da largura da rede é comumente usado para modelos de tamanho pequeno (Howard et al., 2017; Sandler et al., 2018;

[1] Por uma questão de simplicidade, omitimos a dimensão do lote.

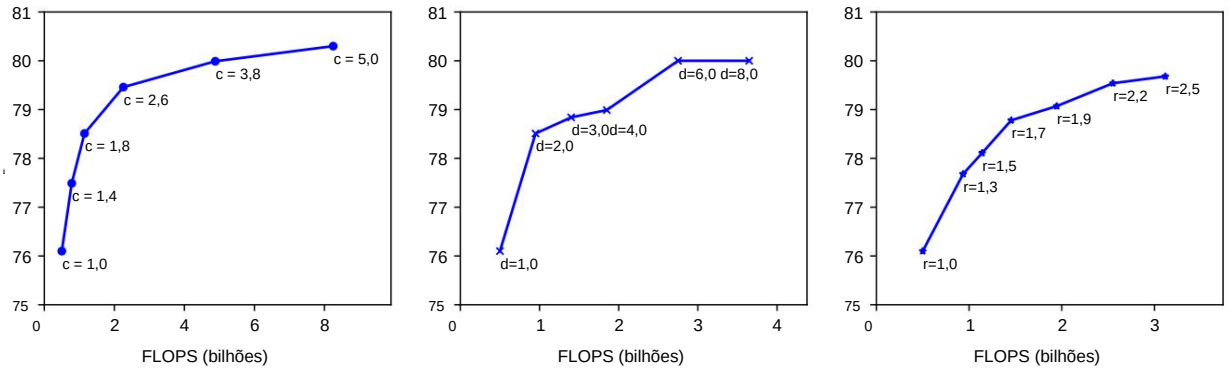

Figura 3. **Ampliando um modelo de linha de base com diferentes coeficientes de largura de rede (w), profundidade (d) e resolução** (r) . Redes maiores com maior largura, profundidade ou resolução tendem a atingir maior precisão, mas o ganho de precisão satura rapidamente após atingir 80%, demonstrando a limitação do dimensionamento unidimensional. A rede de linha de base está descrita na Tabela 1.

Tan et al., 2019) [2]. Conforme discutido em (Zagoruyko & Komodakis , 2016), redes mais amplas tendem a ser capazes de capturar recursos mais refinados e são mais fáceis de treinar. No entanto, redes extremamente amplas, mas superficiais, tendem a ter dificuldades em capturar características de nível superior. Nossos resultados empíricos na Figura 3 (esquerda) mostram que a precisão satura rapidamente quando as redes se tornam muito mais largas com w maiores.

Resolução (r): Com imagens de entrada de resolução mais alta, os ConvNets podem potencialmente capturar padrões mais refinados. A partir de 224x224 nos primeiros ConvNets, os ConvNets modernos tendem a usar 299x299 (Szegedy et al., 2016) ou 331x331 (Zoph et al., 2018) para melhor precisão. Recentemente, o GPipe (Huang et al., 2018) alcançou precisão ImageNet de última geração com resolução de 480x480. Resoluções mais altas, como 600x600, também são amplamente utilizadas em ConvNets de detecção de objetos (He et al., 2017; Lin et al., 2017). A Figura 3 (direita) mostra os resultados do dimensionamento de resoluções de rede, onde de fato resoluções mais altas melhoram a precisão, mas o ganho de precisão diminui para resoluções muito altas (r = 1,0 denota resolução 224x224 e r = 2,5 denota resolução 560x560).

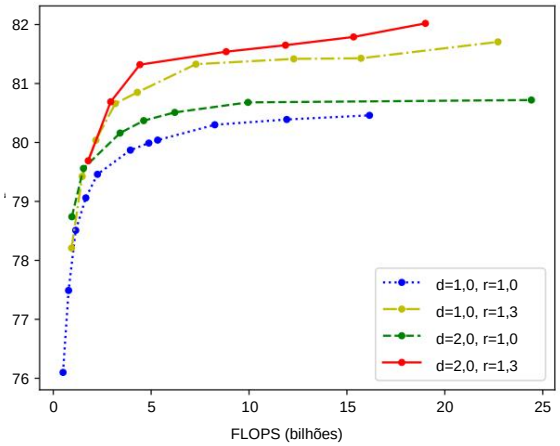

Figura 4. **Dimensionamento da largura da rede para diferentes redes de base.** Cada ponto em uma linha denota um modelo com coeficiente de largura (w) diferente. Todas as redes de linha de base são da Tabela 1. A primeira rede de linha de base (d=1,0, r=1,0) possui 18 camadas convolucionais com resolução 224x224, enquanto a última linha de base (d=2,0, r=1,3) possui 36 camadas com resolução 299x299.

As análises acima nos levam à primeira observação:

Observação 1 – Aumentar qualquer dimensão de largura, profundidade ou resolução da rede melhora a precisão, mas o ganho de precisão diminui para modelos maiores.

3.3. Escala Composta

Observamos empiricamente que diferentes dimensões de escala não são independentes. Intuitivamente, para imagens de resolução mais alta, deveríamos aumentar a profundidade da rede, de modo que os campos receptivos maiores possam ajudar a capturar características semelhantes que incluam mais pixels em imagens maiores. Da mesma forma, também devemos aumentar a largura da rede quando a resolução for maior, em para capturar padrões mais refinados com mais pixels em imagens de alta resolução. Essas intuições sugerem que precisamos coordenar e equilibrar diferentes dimensões de escala, em vez da escala convencional de dimensão única.

Para validar nossas intuições, comparamos a escala de largura sob diferentes profundidades e resoluções de rede, conforme mostrado na Figura 4. Se dimensionarmos apenas a largura da rede w sem alterar a profundidade (d = 1,0) e a resolução (r = 1,0), a precisão saturará rapidamente. Com resolução mais profunda (d = 2,0) e maior (r = 2,0), o dimensionamento de largura atinge uma precisão muito melhor com o mesmo custo de FLOPS. Esses resultados nos levam à segunda observação:

Observação 2 – Para buscar melhor precisão e eficiência, é fundamental equilibrar todas as dimensões de largura, profundidade e resolução da rede durante o dimensionamento do ConvNet.

[2] Em alguma literatura, o escalonamento do número de canais é chamado de “profundidade multiplicador”, que significa o mesmo que nosso coeficiente de largura w.

Na verdade, alguns trabalhos anteriores (Zoph et al., 2018; Real et al., 2019) já tentaram equilibrar arbitrariamente a largura da rede e profundidade, mas todos eles exigem um ajuste manual tedioso.

Neste artigo, propomos um novo **método de escalonamento composto,** que usam um coeficiente composto ÿ para escalar uniformemente largura, profundidade e resolução da rede de maneira baseada em princípios:

$$\begin{aligned} &\text{profundidade: } d = \ddot{y}^{\ddot{y}} \\ &\text{largura: } w = \ddot{y} \quad ^{\ddot{y}} \\ &\text{resolução: } r = \ddot{y} \quad ^{\ddot{y}} \\ &\text{st } \ddot{y} \cdot \ddot{y} \quad ^{2} \cdot \ddot{y} \, ^{2} \, \ddot{y}2 \\ &\ddot{y} \, \ddot{y} \, 1, \ddot{y} \, \ddot{y} \, 1, \ddot{y} \, \ddot{y} \, 1 \end{aligned} \tag{3}$$

onde ÿ, ÿ, ÿ são constantes que podem ser determinadas por um pesquisa em grade pequena. Intuitivamente, ÿ é um coeficiente especificado pelo usuário que controla quantos recursos a mais estão disponíveis para dimensionamento do modelo, enquanto ÿ, ÿ, ÿ especificam como atribuir esses recursos extras para largura, profundidade e resolução da rede, respectivamente . Notavelmente, os FLOPS de uma operação de convolução regular é proporcional a d, w ^{2}, R^{2}, ou seja, duplicando a profundidade da rede duplicará os FLOPS, mas duplicar a largura ou a resolução da rede aumentará os FLOPS em quatro vezes. Desde a convolução as operações geralmente dominam o custo de computação em ConvNets, dimensionar um ConvNet com a equação 3 aumentará aproximadamente o total de FLOPS em ÿ · ÿ ^{2} · ÿ ^{2ÿ} · Neste artigo, nós restrição ÿ · ÿ ^{2} $_{2 \cdot ÿ}$ ÿ 2 tal que para qualquer novo ÿ, o total FLOPS aumentará aproximadamente[3] em 2 ÿ .

4. Arquitetura EficienteNet

Como o dimensionamento do modelo não altera os operadores de camada F$\hat{}$ na rede de linha de base, ter uma boa rede de linha de base é também crítico. Avaliaremos nosso método de escalonamento usando ConvNets existentes, mas para melhor demonstrar a eficácia do nosso método de escalonamento, também desenvolvemos uma nova linha de base para dispositivos móveis, chamada EfficientNet.

Inspirados por (Tan et al., 2019), desenvolvemos nossa rede de base aproveitando uma arquitetura neural multiobjetivo pesquisa que otimiza a precisão e os FLOPS. Especificamente, usamos o mesmo espaço de busca de (Tan et al., 2019), e use ACC(m)×[F LOP S(m)/T] w como a otimização objetivo, onde ACC(m) e F LOP S(m) denotam a precisão e FLOPS do modelo m, T é o FLOPS alvo e w = -0,07 é um hiperparâmetro para controlar o trade-off entre precisão e FLOPS. Ao contrário (Tan et al., 2019; Cai et al., 2019), aqui otimizamos FLOPS em vez de latência, uma vez que não temos como alvo nenhum dispositivo de hardware específico. Nossa busca produz uma rede eficiente, que nome EfficientNet-B0. Como usamos o mesmo espaço de busca como (Tan et al., 2019), a arquitetura é semelhante a Mnas-Net, exceto que nosso EfficientNet-B0 é um pouco maior devido a o alvo maior de FLOPS (nosso alvo de FLOPS é 400M). A Tabela 1 mostra a arquitetura do EfficientNet-B0. Seu principal O alicerce é o gargalo invertido móvel MBConv (Sandler et al., 2018; Tan et al., 2019), ao qual também adicionamos otimização de compressão e excitação (Hu et al., 2018).

Tabela 1. **Rede de linha de base EfficientNet-B0** – Cada linha descreve um estágio i com camadas L$\hat{}$i , com resolução de entrada H$\hat{}$ı, W$\hat{}$ e saída canais C$\hat{}$i. As notações são adotadas da equação 2.

Estágio eu	Operador F$\hat{}$	Resolução H$\hat{}$eu × W$\hat{}$	#Canais C$\hat{}$	#Camadas EU$\hat{}$
	Conv3x3	224 × 224	32	1
1	MBConv1, k3x3	112 × 112	16	1
2	MBConv6, k3x3	112 × 112	24	2
3	MBConv6, k5x5	56 × 56	40	2
4	MBConv6, k3x3	28 × 28	80	3
5	MBConv6, k5x5	14 × 14	112	3
6	MBConv6, k5x5	14 × 14	192	4
7	MBConv6, k3x3	7 × 7	320	1
8 9	Conv1x1 e Pool e FC	7 × 7	1280	1

A partir da linha de base EfficientNet-B0, aplicamos nossa método de escalonamento composto para aumentá-lo em duas etapas:

- PASSO 1: primeiro fixamos ÿ = 1, assumindo duas vezes mais recursos disponíveis, e fazemos uma pequena pesquisa na grade de ÿ, ÿ, ÿ com base nas Equações 2 e 3. Em particular, encontramos os melhores valores para EfficientNet-B0 são ÿ = 1,2, ÿ = 1,1, ÿ = 1,15, sob restrição de ÿ · ÿ ^{2} · ÿ $^{2 ÿ 2.}$
- PASSO 2: fixamos ÿ, ÿ, ÿ como constantes e aumentamos rede de linha de base com ÿ diferente usando a Equação 3, para obter EfficientNet-B1 a B7 (Detalhes na Tabela 2).

Notavelmente, é possível alcançar um desempenho ainda melhor procurando por ÿ, ÿ, ÿ diretamente em torno de um modelo grande, mas o o custo de pesquisa torna-se proibitivamente mais caro em grandes modelos. Nosso método resolve esse problema apenas fazendo pesquisas uma vez na rede de linha de base pequena (etapa 1) e, em seguida, use os mesmos coeficientes de escala para todos os outros modelos (etapa 2).

5. Experimentos

Nesta seção, primeiro avaliaremos nosso método de escalonamento em ConvNets existentes e as novas EfficientNets propostas.

5.1. Ampliando MobileNets e ResNets

Como prova de conceito, primeiro aplicamos nosso método de escalonamento às amplamente utilizadas MobileNets (Howard et al., 2017; Sand-dler et al., 2018) e ResNet (He et al., 2016). Tabela 3 mostra os resultados do ImageNet ao escalá-los em diferentes caminhos. Em comparação com outros métodos de dimensionamento unidimensional, nosso método de escala composto melhora a precisão em todos esses modelos, sugerindo a eficácia de nossa proposta método de escalonamento para ConvNets gerais existentes.

[3] FLOPS pode diferir do valor teórico devido a arredondamentos.

Tabela 2. **Resultados de desempenho do EfficientNet no ImageNet** (Russakovsky et al., 2015). Todos os modelos EfficientNet são dimensionados a partir de nosso linha de base EfficientNet-B0 usando coeficiente composto diferente ÿ na Equação 3. ConvNets com precisão top-1/top-5 semelhante são agrupados juntos para comparação de eficiência. Nossos modelos EfficientNet dimensionados reduzem consistentemente parâmetros e FLOPS em uma ordem de grandeza (redução de parâmetros de até 8,4x e redução de FLOPS de até 16x) do que os ConvNets existentes.

Modelo	Contas principais 1 5 principais contas.		#Relação de Params para Rede Eficiente		#Relação de FLOPs para Rede Eficiente	
EfficientNet-B0	**77,1%**	**93,3%**	**5,3M**	**1x**	**0,39B**	**1x**
ResNet-50 (Ele 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	**79,1%**	**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
Xcepção (Chollet, 2017)	79,0%	94,5%	23 milhões	3,0x	8,4B	12x
EfficientNet-B2	**80,1%**	**94,9%**	**9,2 milhões**	**1x**	**1,0B**	**1x**
Inception-v4 (Szegedy et al., 2017)	80,0%	95,0%	48 milhões	5,2x	13B	13x
Inception-resnet-v2 (Szegedy et al., 2017)	80,1%	95,1%	56 milhões	6,1x	13B	13x
EficienteNet-B3	**81,6%**	**95,7%**	**12 milhões**	**1x**	**1,8B**	**1x**
ResNeXt-101 (Xie et al., 2017)	80,9%	95,6%	84 milhões	7,0x	32B	18x
PolyNet (Zhang et al., 2017)	81,3%	95,8%	92 milhões	7,7x	35B	19x
EficienteNet-B4	**82,9%**	**96,4%**	**19M**	**1x**	**4.2B**	**1x**
SENet (Hu et al., 2018)	82,7%	96,2%	146 milhões	7,7x	42B	10x
NASNet-A (Zoph et al., 2018)	82,7%	96,2%	89 milhões	4,7x	24B	5,7x
AmoebaNet-A (Real et al., 2019)	82,8%	96,1%	87 milhões	4,6x	23B	5,5x
PNASNet (Liu et al., 2018)	82,9%	96,2%	86 milhões	4,5x	23B	6,0x
EfficientNet-B5	**83,6%**	**96,7%**	**30 milhões**		**9,9B**	**1x**
AmoebaNet-C (Cubuk et al., 2019)	83,5%	96,5%	155 milhões	**1x5,2x**	41B	4,1x
EficienteNet-B6	**84,0%**	**96,8%**	**43 milhões**	**1x**	**19B**	**1x**
GPipe **EfficientNet-**	**84,3%**	**97,0%**	**66 milhões**		**37B**	**1x**
B7 (Huang et al., 2018)	84,3%	97,0%	557 milhões	**1x8,4x**	-	-

Omitimos modelos de conjunto e multi-corte (Hu et al., 2018), ou modelos pré-treinados em imagens 3,5B do Instagram (Mahajan et al., 2018).

Tabela 3. **Ampliação de MobileNets e ResNet.**

Modelo	FLOPS	Top-1 Acc.
Linha de base MobileNetV1 (Howard et al., 2017)	0,6B	70,6%
Dimensione o MobileNetV1 por largura (w=2)	2,2B	74,2%
Dimensione o MobileNetV1 por resolução (r=2)	2,2B	72,7%
escala composta (d=1,4, w=1,2, r=1,3)	**2,3B**	**75,6%**
Linha de base MobileNetV2 (Sandler et al., 2018)	0,3B	72,0%
Dimensione o MobileNetV2 por profundidade (d=4)	1,2B	76,8%
Dimensione o MobileNetV2 por largura (w=2)	1,1B	76,4%
Dimensione o MobileNetV2 por resolução (r=2)	1,2B	74,8%
Escala composta MobileNetV2	**1,3B**	**77,4%**
Linha de base ResNet-50 (He et al., 2016)	4.1B	76,0%
Dimensione o ResNet-50 por profundidade (d=4)	16,2B	78,1%
Dimensione o ResNet-50 por largura (w=2)	14,7B	77,7%
Dimensione o ResNet-50 por resolução (r=2)	16,4B	77,5%
Escala composta ResNet-50	**16,7B**	**78,8%**

Tabela 4. **Comparação de latência de inferência** – A latência é medida com tamanho de lote 1 em um único núcleo da CPU Intel Xeon E5-2690.

	Ac. @ Latência		Ac. @ Latência
ResNet-152	77,8% a 0,554s	GPipe	84,3% às 19,0s
EfficientNet-B1	78,8% a 0,098s	EfficientNet-B7	84,4% a 3,1s
Acelerar	**5,7x**	**Acelerar**	**6,1x**

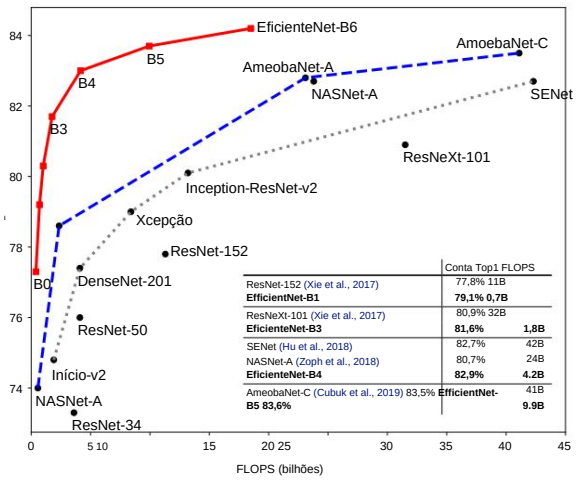

Figura 5. **Precisão FLOPS vs. ImageNet –** semelhante à Figura 1 exceto que compara FLOPS em vez do tamanho do modelo.

5.2. Resultados do ImageNet para EfficientNet

Treinamos nossos modelos EfficientNet no ImageNet usando configurações semelhantes a (Tan et al., 2019): otimizador RMSProp com decaimento 0,9 e momento 0,9; momento da norma do lote 0,99;

Tabela 5. **Resultados de desempenho do EfficientNet em conjuntos de dados de aprendizagem por transferência.** Nossos modelos EfficientNet dimensionados alcançam precisão de última geração para 5 de 8 conjuntos de dados, com 9,6x menos parâmetros em média.

	Comparação com os melhores resultados disponíveis ao público			Comparação com os melhores resultados relatados		
	Modelo	Acc.	#Param(ratio)	Modelo	Acc.	#Param
CIFAR-10	NASNet-A	98,0%	85M (21x)	EficienteNet-B0	98,1%	4M (21x)
CIFAR-100	NASNet-A	87,5%	85M (21x)	EficienteNet-B0	88,1%	4M (21x)
Foto de pássaro	Inception-v4	81,8%	41M (1,5x)	EficienteNet-B5	82,0%	28M (1,5x)
Carros de Stanford	Inception-v4	93,4%	41M (1,5x)	EficienteNet-B3	93,6%	10M (4,1x)
Flores	V4	98,5%	41M (1,5x)	EFABIFIFITYNET-B5	98,5%	28M (1,5x)
Aeronave FGVC	DAT	97,7%	EFIFIFITENTENET-B7	**98,8%**	EFIFIFIFITORY-B7	**92,9%**
Oxford-IIIT Pets	41m	DAT	92,9%	GPipe **95,9%**	556M	EfficientNet-B6
Food-101	ResNet-152	94,5%	58M	EfficientNet-B4	94,8%	17M (5,6x)
	Inception-v4	90,8%	41M	EfficientNet-B4	91,5%	17M (2,4x)
Média geográfica			**(4.7x)**			

	Comparação com os melhores resultados relatados		
	Modelo	Acc.	#Param
CIFAR-10	†Gpipe **99,0%**	556M	EfficientNet-B7
CIFAR-100	Gpipe	91,3%	556M EfficientNet-B7 **91,7%** 64M (8,7x)
Foto de pássaro	GPipe	83,6%	556M EfficientNet-B7 **84,3%** 64M (8,7x)
Carros de Stanford	‡DAT **94,8%**	-	EficienteNET-B7 94,7% Inception- -
Flores			-
Aeronave FGVC			-
Oxford-IIIT Pets			-
Food-101	GPipe	93,0%	556M EfficientNet-B7 **93,0%** 64M (8,7x)
Média geográfica			**(9,6x)**

†GPipe (Huang et al., 2018) treina modelos gigantes com biblioteca especializada de paralelismo de pipeline.

‡DAT denota aprendizagem de transferência adaptativa de domínio (Ngiam et al., 2018). Aqui comparamos apenas os resultados da aprendizagem por transferência baseada no ImageNet.

A precisão de transferência e #params para NASNet (Zoph et al., 2018), Inception-v4 (Szegedy et al., 2017), ResNet-152 (He et al., 2016) são de (Kornblith et al., 2019).

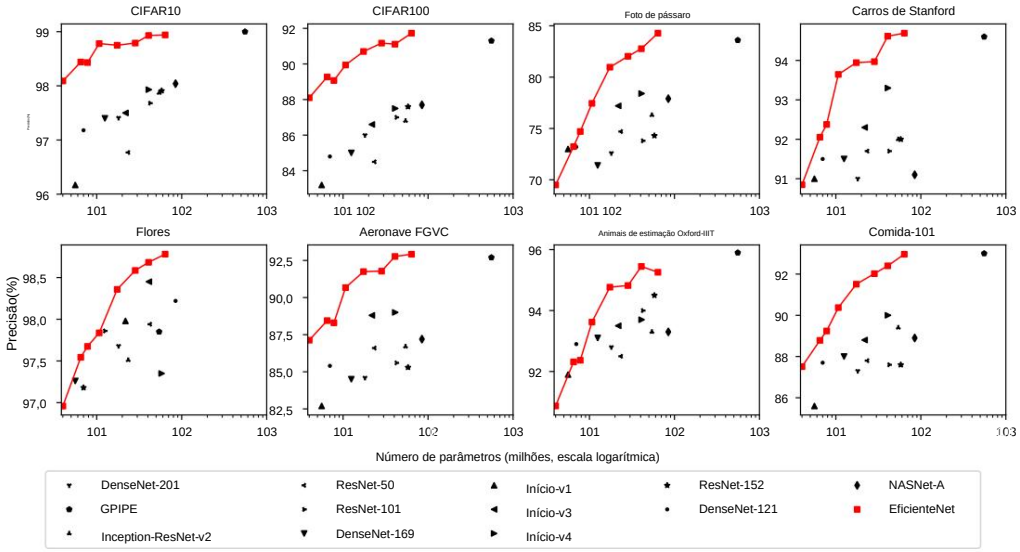

Figura 6. **Parâmetros do modelo versus precisão do aprendizado por transferência –** Todos os modelos são pré-treinados no ImageNet e ajustados em novos conjuntos de dados.

queda de peso 1e-5; taxa de aprendizagem inicial 0,256 que decai em 0,97 a cada 2,4 épocas. Também usamos a ativação SiLU (Swish-1) (Ramachandran et al., 2018; Elfwing et al., 2018; Hendrycks & Gimpel, 2016), AutoAugment (Cubuk et al., 2019) e profundidade estocástica (Huang et al., 2016) com probabilidade de sobrevivência 0,8. Como é comumente sabido que modelos maiores precisam de mais regularização, aumentamos linearmente o abandono (Srivastava et al., 2014) proporção de 0,2 para EfficientNet-B0 para 0,5 para B7. Reservamos 25 mil imagens escolhidas aleatoriamente de o conjunto de treinamento como um conjunto minival e executar antecipadamente parando neste minival; em seguida, avaliamos o ponto de verificação interrompido antecipadamente no conjunto de validação original como relatar a precisão da validação final.

A Tabela 2 mostra o desempenho de todos os modelos EfficientNet que são dimensionados a partir da mesma linha de base EfficientNet-B0. Nosso Os modelos EfficientNet geralmente usam uma ordem de grandeza menos parâmetros e FLOPS do que outros ConvNets com precisão semelhante. Em particular, o nosso EfficientNet-B7 alcança 84,3% de precisão top1 com parâmetros de 66M e 37B FLOPS, sendo mais preciso, mas **8,4x menor** que o anterior melhor GPipe (Huang et al., 2018). Esses ganhos vêm melhores arquiteturas, melhor dimensionamento e melhor treinamento configurações personalizadas para EfficientNet.

A Figura 1 e a Figura 5 ilustram a precisão dos parâmetros e curva de precisão FLOPS para ConvNets representativos, onde nossos modelos EfficientNet dimensionados alcançam melhor precisão com muito menos parâmetros e FLOPS do que outros ConvNets. Notavelmente, nossos modelos EfficientNet não são apenas pequeno, mas também computacionalmente mais barato. Por exemplo, nosso O EfficientNet-B3 atinge maior precisão do que o ResNeXt- 101 (Xie et al., 2017) usando **18x menos FLOPS.**

Para validar a latência, também medimos a inferência latência para alguns CovNets representativos em uma CPU real como mostrado na Tabela 4, onde relatamos a latência média ao longo 20 corridas. Nosso EfficientNet-B1 funciona **5,7x mais rápido** que o ResNet-152 amplamente utilizado, enquanto EfficientNet-B7 roda cerca de **6,1x mais rápido** que o GPipe (Huang et al., 2018), sugerindo que nosso EfficientNets são realmente rápidos em hardware real.

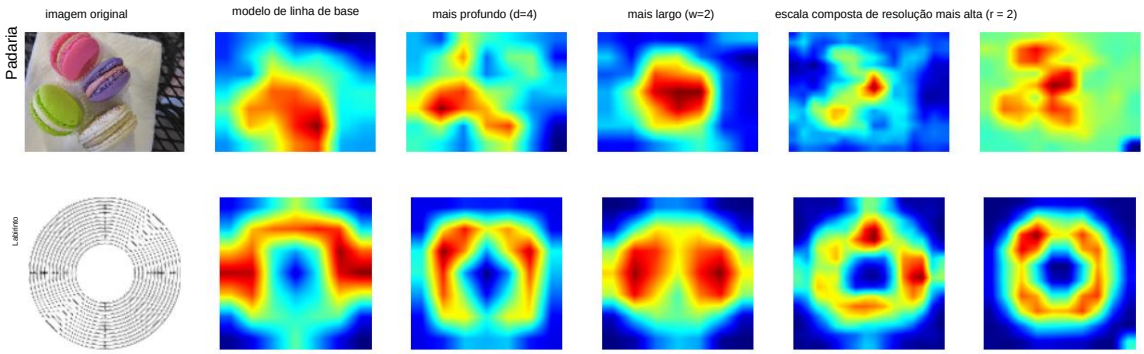

Figura 7. **Mapa de ativação de classe (CAM) (Zhou et al., 2016) para modelos com diferentes métodos de escalonamento -** Nosso método de escalonamento composto permite que o modelo em escala (última coluna) se concentre em regiões mais relevantes com mais detalhes do objeto. Os detalhes do modelo estão na Tabela 7.

Tabela 6. **Transferência de conjuntos de dados de aprendizagem.**

Conjunto de dados	Tamanho do trem	Tamanho do teste	#Classes
CIFAR-10 (Krizhevsky & Hinton, 2009)	50.000	10.000	10
CIFAR-100 (Krizhevsky & Hinton, 2009)	50.000	10.000	100
Birdsnap (Berg et al., 2014)	47.386	2.443	500
Carros de Stanford (Krause et al., 2013)	8.144	8.041	196
Flores (Nilsback & Zisserman, 2008)	2.040	6.149	102
Aeronaves FGVC (Maji et al., 2013)	6.667	3.333	100
Animais de estimação Oxford-IIIT (Parkhi et al., 2012)	3.680	3.369	37
Comida-101 (Bossard et al., 2014)	75.750	25.250	101

5.3. Transferir resultados de aprendizagem para EfficientNet

Também avaliamos nosso EfficientNet em uma lista de conjuntos de dados de aprendizagem por transferência comumente usados, conforme mostrado na Tabela 6. Pegamos emprestadas as mesmas configurações de treinamento de (Kornblith et al., 2019) e (Huang et al., 2018), que levam ImageNet pontos de verificação pré-treinados e ajuste fino em novos conjuntos de dados.

A Tabela 5 mostra o desempenho da aprendizagem por transferência: (1) Em comparação com modelos públicos disponíveis, como NASNet-A (Zoph et al., 2018) e Inception-v4 (Szegedy et al., 2017), nossos modelos Ef-ficientNet alcançam melhor precisão com média de 4,7x (até 21x) redução de parâmetros. (2) Em comparação com modelos de última geração, incluindo DAT (Ngiam et al., 2018) que sintetiza dinamicamente dados de treinamento e GPipe (Huang et al., 2018) que é treinado com paralelismo de pipeline especializado , nossos modelos EfficientNet ainda superam sua precisão em 5 de 8 conjuntos de dados, mas usando 9,6x menos parâmetros

A Figura 6 compara a curva de parâmetros de precisão para uma variedade de modelos. Em geral, nossos EfficientNets consistentemente alcançar melhor precisão com uma ordem de magnitude menos parâmetros do que os modelos existentes, incluindo ResNet (He et al., 2016), DenseNet (Huang et al., 2017), Início (Szegedy et al., 2017) e NASNet (Zoph et al., 2018).

6. Discussão

Para desemaranhar a contribuição da nossa escala proposta método da arquitetura EfficientNet, a Figura 8 compara o desempenho do ImageNet de diferentes métodos de escalonamento

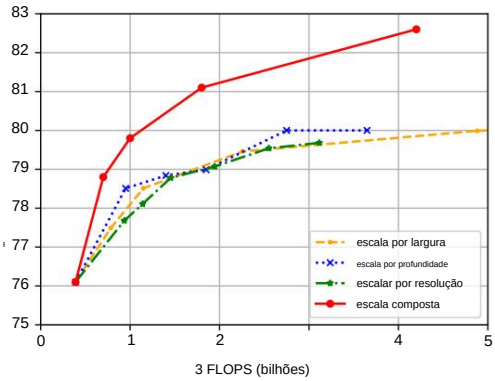

Figura 8. **Ampliando o EfficientNet-B0 com diferentes métodos.**

Tabela 7. **Modelos em escala usados na Figura 7.**

Modelo	FLOPS	Top-1 Acc.
Modelo de linha de base (EfficientNet-B0)	0,4B	77,3%
Modelo em escala por profundidade (d=4)	1,8B	79,0%
Modelo em escala por largura (w=2)	1,8B	78,9%
Modelo em escala por resolução (r=2)	1,9B	79,1%
Escala Composta (d=1,4, w=1,2, r=1,3)	**1,8B**	**81,1%**

ods para a mesma rede de linha de base EfficientNet-B0. Em geral , todos os métodos de escalonamento melhoram a precisão com o custo de mais FLOPS, mas nosso método de escalonamento composto pode melhorar ainda mais a precisão, em até 2,5%, do que outros métodos de dimensionamento unidimensional, sugerindo a importância de nossa escala composta proposta.

Para entender melhor por que nosso escalonamento composto método é melhor que outros, a Figura 7 compara a classe mapa de ativação (Zhou et al., 2016) para alguns representantes modelos com diferentes métodos de escala. Todos esses modelos são dimensionados a partir da mesma linha de base, e suas estatísticas são mostradas na Tabela 7. As imagens são escolhidas aleatoriamente no ImageNet conjunto de validação. Conforme mostrado na figura, o modelo com escala composta tende a focar em regiões mais relevantes com mais detalhes do objeto, enquanto outros modelos carecem de detalhes do objeto ou incapaz de capturar todos os objetos nas imagens.

7. Conclusão

Neste artigo, estudamos sistematicamente o dimensionamento da ConvNet e identificamos que equilibrar cuidadosamente a largura, a profundidade e a resolução da rede é uma peça importante, mas que falta, e nos impede de obter melhor precisão e eficiência. Para resolver esse problema, propomos um método de escalonamento composto simples e altamente eficaz, que nos permite escalar facilmente uma ConvNet de linha de base para quaisquer restrições de recursos alvo de uma forma mais baseada em princípios, mantendo a eficiência do modelo. Alimentados por este método de escala composto, demonstramos que um modelo EfficientNet de tamanho móvel pode ser ampliado de forma muito eficaz, superando a precisão do estado da arte com uma ordem de magnitude menos parâmetros e FLOPS, tanto no ImageNet quanto em cinco comumente usou conjuntos de dados de aprendizagem por transferência.

Reconhecimentos

Agradecemos a Ruoming Pang, Vijay Vasudevan, Alok Aggarwal, Barret Zoph, Hongkun Yu, Jonathon Shlens, Raphael Gontijo Lopes, Yifeng Lu, Daiyi Peng, Xiaodan Song, Samy Bengio, Jeff Dean e à equipe do Google Brain pela ajuda.

Apêndice

Desde 2017, a maioria dos artigos de pesquisa apenas relata e compara a precisão da validação do ImageNet; este artigo também segue esta convenção para melhor comparação. Além disso, também verificamos a precisão do teste enviando nossas previsões sobre as 100 mil imagens do conjunto de testes para http://image-net.org; os resultados estão na Tabela 8. Como esperado, a precisão do teste está muito próxima da precisão da validação.

Tabela 8. **Validação ImageNet vs. Teste de Precisão Top-1/5.**

	B0	B1	B2	B3	B4	B5	B6	B7
Val top1	77,11	79,13	80,07	81,59	82,89	83,60	83,95	84,26
Teste top1	77,23	79,17	80,16	81,72	82,94	83,69	84,04	84,33
Val top5	93,35	94,47	94,90	95,67	96,37	96,71	96,76	96,97
Teste top5	93,45	94,43	94,98	95,70	96,27	96,64	96,86	96,94

Referências

Berg, T., Liu, J., Woo Lee, S., Alexander, ML, Jacobs, DW e Belhumeur, PN Birdsnap: Categorização visual refinada de pássaros em grande escala. CVPR, pp .

Bossard, L., Guillaumin, M., e Van Gool, L. Food-101 – mineração de componentes discriminativos com florestas aleatórias. ECCV, pp.

Cai, H., Zhu, L. e Han, S. Proxylessnas: Pesquisa direta de arquitetura neural na tarefa e hardware alvo. ICLR, 2019.

Chollet, F. Xception: Aprendizado profundo com convoluções separáveis em profundidade. CVPR, pp.

Cubuk, ED, Zoph, B., Mane, D., Vasudevan, V., e Le, QV Autoaugment: Aprendendo políticas de aumento a partir de dados. CVPR, 2019.

Elfwing, S., Uchibe, E., e Doya, K. Unidades lineares ponderadas em sigmóide para aproximação de função de rede neural na aprendizagem por reforço. Redes Neurais, 107:3–11, 2018.

Gholami, A., Kwon, K., Wu, B., Tai, Z., Yue, X., Jin, P., Zhao, S., e Keutzer, K. Squeezenext: Projeto de rede neural com reconhecimento de hardware . Workshop ECV no CVPR'18, 2018.

Han, S., Mao, H. e Dally, WJ Compressão profunda: Compressão de redes neurais profundas com poda, quantização treinada e codificação huffman. ICLR, 2016.

He, K., Zhang, X., Ren, S., e Sun, J. Aprendizagem residual profunda para reconhecimento de imagem. CVPR, pp .

He, K., Gkioxari, G., Dollar, P. e Girshick, R. Mask r-cnn. 2980–2988, 2017.

Ele, Y., Lin, J., Liu, Z., Wang, H., Li, L.-J., e Han, S. Amc: Automl para compactação e aceleração de modelos em dispositivos móveis. ECCV, 2018.

Hendrycks, D. e Gimpel, K. Unidades lineares de erro gaussiano (gelus). Pré-impressão do arXiv arXiv:1606.08415, 2016.

Howard, AG, Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., e Adam, H. Mobilenets: Redes neurais convolucionais eficientes para aplicações de visão móvel. Pré-impressão do arXiv arXiv:1704.04861, 2017.

Hu, J., Shen, L. e Sun, G. Rede de compressão e excitação funciona. CVPR, 2018.

Huang, G., Sun, Y., Liu, Z., Sedra, D., e Weinberger, KQ Redes profundas com profundidade estocástica. ECCV, pp .

Huang, G., Liu, Z., Van Der Maaten, L., e Weinberger, KQ Redes convolucionais densamente conectadas. CVPR, 2017.

Huang, Y., Cheng, Y., Chen, D., Lee, H., Ngiam, J., Le, QV e Chen, Z. Gpipe: Treinamento eficiente de redes neurais gigantes usando paralelismo de pipeline. Pré-impressão do arXiv arXiv :1808.07233, 2018.

Iandola, FN, Han, S., Moskewicz, MW, Ashraf, K., Dally, WJ e Keutzer, K. Squeezenet: Precisão de nível Alexnet com 50x menos parâmetros e tamanho de modelo <0,5 mb . Pré-impressão do arXiv arXiv:1602.07360, 2016.

Ioffe, S. e Szegedy, C. Normalização em lote: Acelerando o treinamento profundo da rede, reduzindo a mudança interna de covariáveis. ICML, pp. 448–456, 2015.

Kornblith, S., Shlens, J. e Le, QV Os melhores modelos de imagenet transferem melhor? CVPR, 2019.

Krause, J., Deng, J., Stark, M., e Fei-Fei, L. Coletando um conjunto de dados em grande escala de carros refinados. Segundo Workshop sobre Categorização Visual Refinada, 2013.

Krizhevsky, A. e Hinton, G. Aprendendo múltiplas camadas de recursos a partir de imagens minúsculas. Relatório Técnico, 2009.

Krizhevsky, A., Sutskever, I., e Hinton, GE Classificação Imagenet com redes neurais convolucionais profundas. Em NIPS, pp. 1097–1105, 2012.

Lin, H. e Jegelka, S. Resnet com camadas ocultas de um neurônio é um aproximador universal. NeurIPS, pp .

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., e Belongie, S. Apresentam redes de pirâmide para detecção de objetos. CVPR, 2017.

Liu, C., Zoph, B., Shlens, J., Hua, W., Li, L.-J., Fei-Fei, L., Yuille, A., Huang, J., e Murphy, K. Pesquisa progressiva de arquitetura neural. ECCV, 2018.

Lu, Z., Pu, H., Wang, F., Hu, Z. e Wang, L. O poder expressivo das redes neurais: uma visão da largura. NeuroIPS, 2018.

Ma, N., Zhang, X., Zheng, H.-T., e Sun, J. Shufflenet v2: Diretrizes práticas para projeto de arquitetura CNN eficiente. ECCV, 2018.

Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., e van der Maaten, L. Explorando os limites da supervisão fraca Pré treino. Pré-impressão do arXiv arXiv :1805.00932, 2018.

Maji, S., Rahtu, E., Kannala, J., Blaschko, M., e Vedaldi, A. Classificação visual refinada de aeronaves. Pré-impressão do arXiv arXiv :1306.5151, 2013.

Ngiam, J., Peng, D., Vasudevan, V., Kornblith, S., Le, QV e Pang, R. Aprendizagem de transferência adaptativa de domínio com modelos especializados. Pré-impressão do arXiv arXiv:1811.07056, 2018.

Nilsback, M.-E. e Zisserman, A. Classificação automatizada de flores em um grande número de classes. ICVGIP, pp .

Parkhi, OM, Vedaldi, A., Zisserman, A. e Jawahar, C. Gatos e cachorros. CVPR, pp.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., e Sohl-Dickstein, J. Sobre o poder expressivo das redes neurais profundas. CIML, 2017.

Ramachandran, P., Zoph, B., e Le, QV Procurando funções de ativação. Pré-impressão do arXiv arXiv:1710.05941, 2018.

Real, E., Aggarwal, A., Huang, Y., e Le, QV Evolução regularizada para pesquisa de arquitetura de classificador de imagens. AAAI, 2019.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., e outros. Desafio de reconhecimento visual em grande escala da Imagenet . Jornal Internacional de Visão Computacional, 115(3): 211–252, 2015.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. e Chen, L.-C. Mobilenetv2: Resíduos invertidos e gargalos lineares. CVPR, 2018.

Sharir, O. e Shashua, A. Sobre o poder expressivo das arquiteturas sobrepostas de aprendizagem profunda. ICLR, 2018.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., e Salakhutdinov, R. Dropout: uma maneira simples de evitar o overfitting de redes neurais. The Journal of Machine Learning Research, 15(1):1929–1958, 2014.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., e Rabinovich, A. Indo mais fundo nas convoluções. CVPR, pp. 1–9, 2015.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. e Wojna, Z. Repensando a arquitetura inicial para visão computacional. CVPR, pp.

Szegedy, C., Ioffe, S., Vanhoucke, V. e Alemi, AA Inception-v4, inception-resnet e o impacto das conexões residuais na aprendizagem. AAAI, 4:12, 2017.

Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., e Le, QV MnasNet: Pesquisa de arquitetura neural com reconhecimento de plataforma para dispositivos móveis. CVPR, 2019.

Xie, S., Girshick, R., Dollar, P., Tu, Z. e He, K. Transformações residuais agregadas para redes neurais profundas. CVPR, pp.

Yang, T.-J., Howard, A., Chen, B., Zhang, X., Go, A., Sze, V., e Adam, H. Netadapt: Adaptação de rede neural com reconhecimento de plataforma para dispositivos móveis formulários. ECCV, 2018.

Zagoruyko, S. e Komodakis, N. Redes residuais amplas. BMVC, 2016.

Zhang, X., Li, Z., Loy, CC, e Lin, D. Polynet: Uma busca pela diversidade estrutural em redes muito profundas. CVPR, pp .

Zhang, X., Zhou, X., Lin, M., e Sun, J. Shufflenet: Uma rede neural convolucional extremamente eficiente para dispositivos móveis. CVPR, 2018.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., e Torralba, A. Aprendendo recursos profundos para localização discriminativa. CVPR, pp.

Zoph, B. e Le, QV Pesquisa de arquitetura neural com aprendizagem por reforço. ICLR, 2017.

Zoph, B., Vasudevan, V., Shlens, J., e Le, QV Aprendendo arquiteturas transferíveis para reconhecimento de imagem escalonável. CVPR, 2018.