EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

Жижин Петр

НИУ ВШЭ

12.03.2020

Аннотация

- Показываем, как надо тюнить один параметр чтобы увеличивать качество нейросетей
- Добились нового State-of-the-Art
  - 8.4x меньше параметров
  - 6.1x быстрее
  - Не требует нескольких видеокарт и сложных пайплайнов

Аннотация

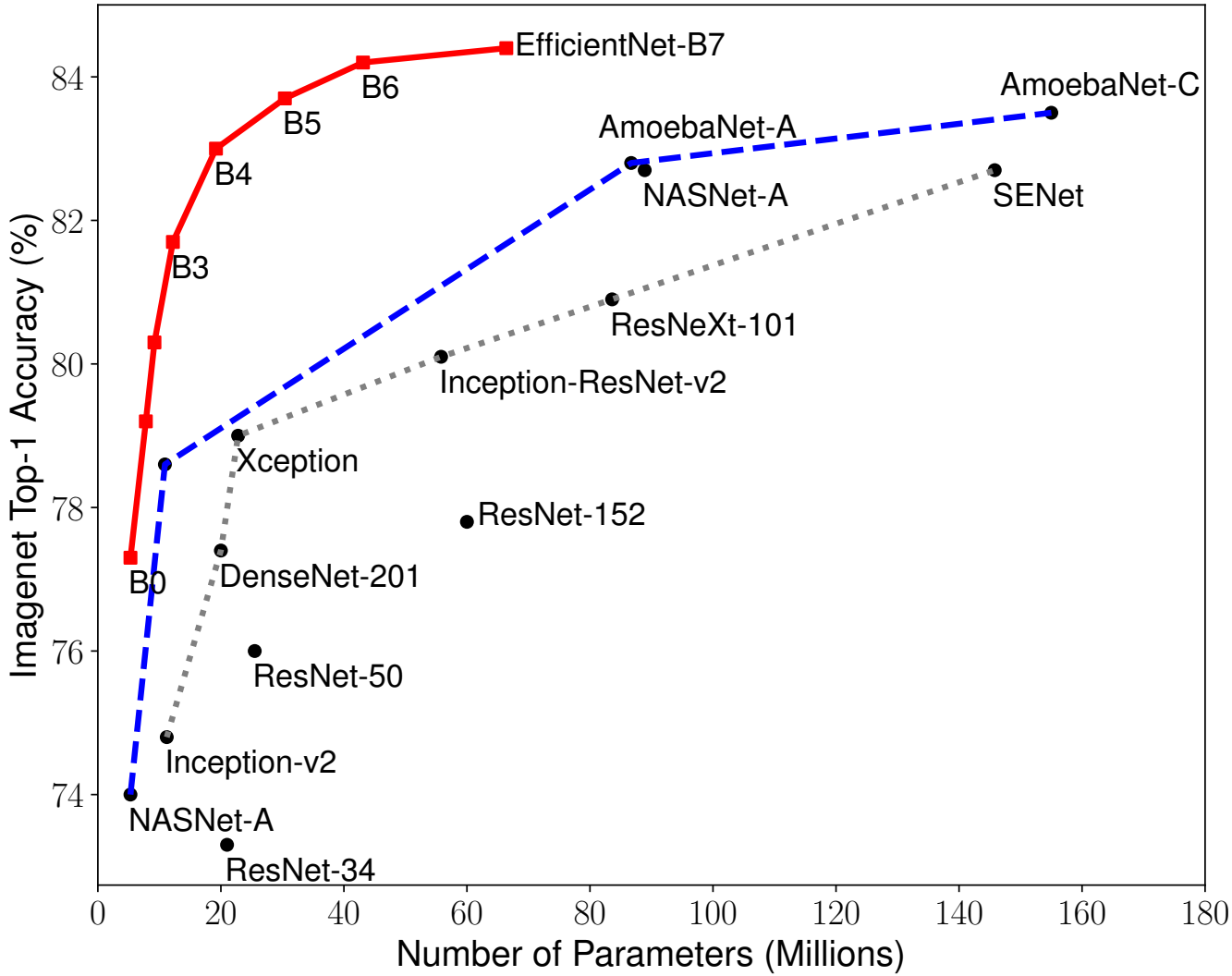

Рис. 1: Размер модели против точности. EfficientNet лучше всех на всех вычислительных бюджетах

Введение

Как масштабируют сети?

- Пусть у вас есть свёрточная сеть
- Вы учите её на любимом датасете
- Качество недостаточно высокое, модель не переобучается. Что делать?

Введение

Как масштабируют сети?

- Пусть у вас есть свёрточная сеть
- Вы учите её на любимом датасете
- Качество недостаточно высокое, модель не переобучается. Что делать?
  - Stack more layers (ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152)

Введение

Как масштабируют сети?

- Пусть у вас есть свёрточная сеть
- Вы учите её на любимом датасете
- Качество недостаточно высокое, модель не переобучается. Что делать?
  - Stack more layers (ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152)
  - Увеличить количество фильтров, размер полносвязного слоя (параметр α в MobileNet)

Введение

Как масштабируют сети?

- Пусть у вас есть свёрточная сеть
- Вы учите её на любимом датасете
- Качество недостаточно высокое, модель не переобучается. Что делать?
  - Stack more layers (ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152)
  - Увеличить количество фильтров, размер полносвязного слоя (параметр α в MobileNet)
  - Увеличить разрешение исходной картинки (ДЗ по DeepLearning)

Введение

Как масштабируют сети?

- Пусть у вас есть свёрточная сеть
- Вы учите её на любимом датасете
- Качество недостаточно высокое, модель не переобучается. Что делать?
  - Stack more layers (ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152)
  - Увеличить количество фильтров, размер полносвязного слоя (параметр α в MobileNet)
  - Увеличить разрешение исходной картинки (ДЗ по DeepLearning)
  - Поменять структуру сети, использовать более умные блоки: ResNet, MobileNet

Введение

Как масштабируют сети?

- Пусть у вас есть свёрточная сеть
- Вы учите её на любимом датасете
- Качество недостаточно высокое, модель не переобучается. Что делать?
  - Stack more layers (ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152)
  - Увеличить количество фильтров, размер полносвязного слоя (параметр α в MobileNet)
  - Увеличить разрешение исходной картинки (ДЗ по DeepLearning)
  - Поменять структуру сети, использовать более умные блоки: ResNet, MobileNet
- Как правильно масштабировать (блоки не меняем)?

Введение

Как масштабируют сети?

- Пусть у вас есть свёрточная сеть
- Вы учите её на любимом датасете
- Качество недостаточно высокое, модель не переобучается. Что делать?
  - Stack more layers (ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152)
  - Увеличить количество фильтров, размер полносвязного слоя (параметр α в MobileNet)
  - Увеличить разрешение исходной картинки (ДЗ по DeepLearning)
  - Поменять структуру сети, использовать более умные блоки: ResNet, MobileNet
- Как правильно масштабировать (блоки не меняем)?
  - Совместно надо увеличивать масштаб

Формализуем масштабирование

- Начальная сеть:

$$\mathcal{N} = \bigodot_{i=1\ldots s} \mathcal{F}_i^{L_i}\left(X_{\langle H_i, W_i, C_i\rangle}\right)$$

  $\mathcal{N}$ – сеть, s – количество стадий, $\mathcal{F}_i^{L_i}$ – блок на стадии i (например ResNet или MobileNet блок), повторённый L_i раз, H_i, W_i, C_i – размер изображения на входе стадии сети.

- Отмасштабированная сеть. Глубина в d раз, ширина в w раз, разрешение в r раз.

$$\mathcal{N}(d, w, r) = \bigodot_{i=1\ldots s} \mathcal{F}_i^{d\cdot L_i}\left(X_{\langle r\cdot H_i, r\cdot W_i, w\cdot C_i\rangle}\right)$$

- Обычно в статьях меняется один из параметров d, w, r, а остальные незначительно.

Иллюстрация типов масштабирования

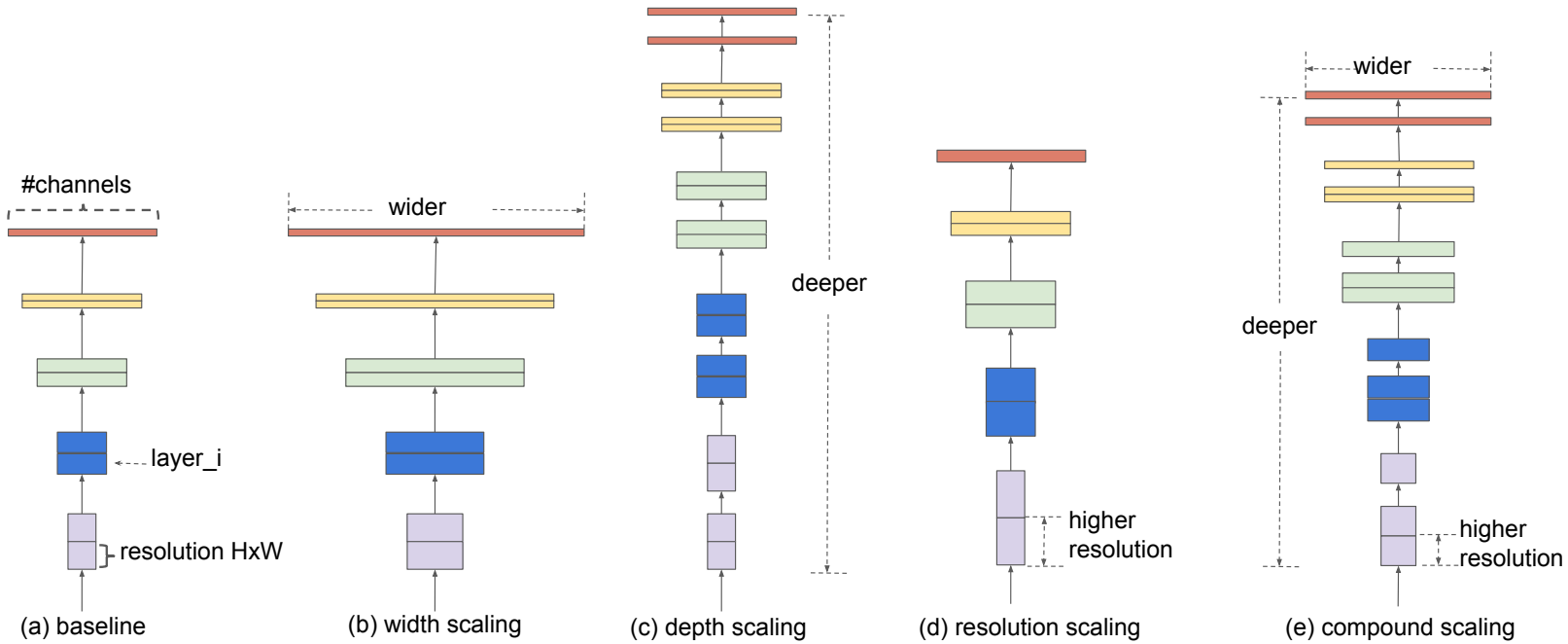

Рис. 2: Разные типы масштабирования. (a) – начальная модель, (b-d) – масштабирование по одной из осей, (e) – предложенный метод совместного масштабирования

Важное дополнение

1. Глубина за счёт количества последовательных блоков. Больше блоков ResNet, не более глубокие блоки.
2. Если блок одной архитектуры повторяется несколько раз, то при увеличении глубины – кратно увеличивается количество повторений блока.
3. L_i – количество слоёв на этапе i (количество повторений блока).

До масштабирования

Stage i	Operator $\hat{\mathcal{F}}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	32	1
2	Res Net block, depth 3, k3x3	16	1
3	Res Net block, depth 2, k3x3	24	2
4	Res Net block, depth 3, k5x5	40	2
5	Conv1x1 & Pooling & FC	1280	1

После масштабирования в 2 раза по глубине

Stage i	Operator $\hat{\mathcal{F}}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	32	2
2	Res Net block, depth 3, k3x3	16	2
3	Res Net block, depth 2, k3x3	24	4
4	Res Net block, depth 3, k5x5	40	4
5	Conv1x1 & Pooling & FC	1280	2

Влияние размера модели на качество

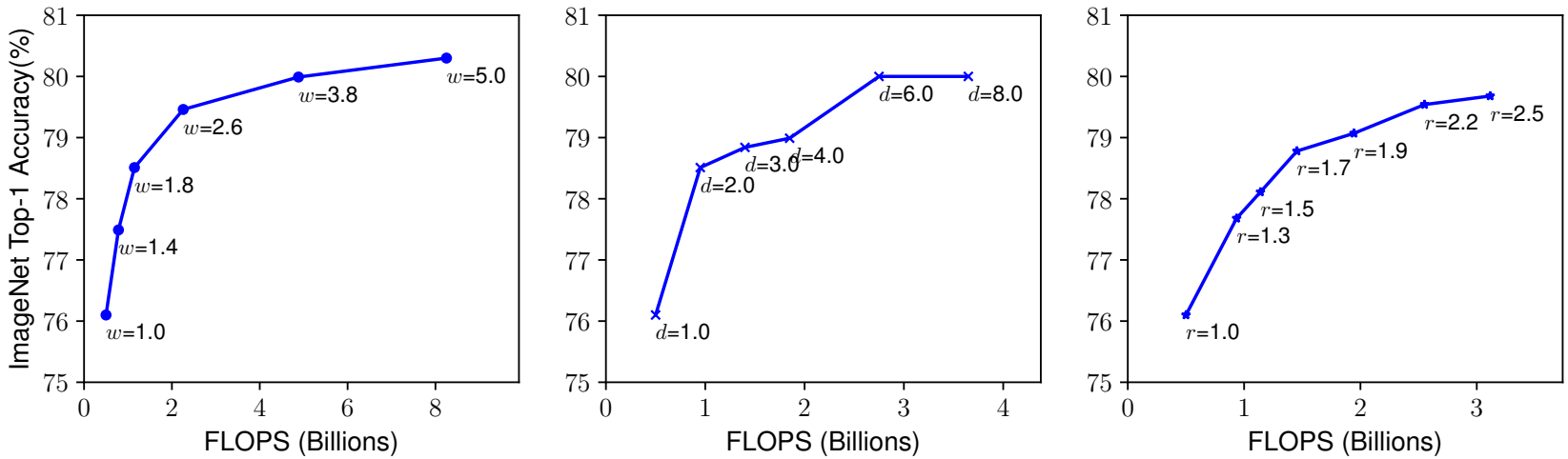

Рис. 3: Качество масштабирования по одному параметру: ширине, глубине, разрешению

Влияние размера модели на качество

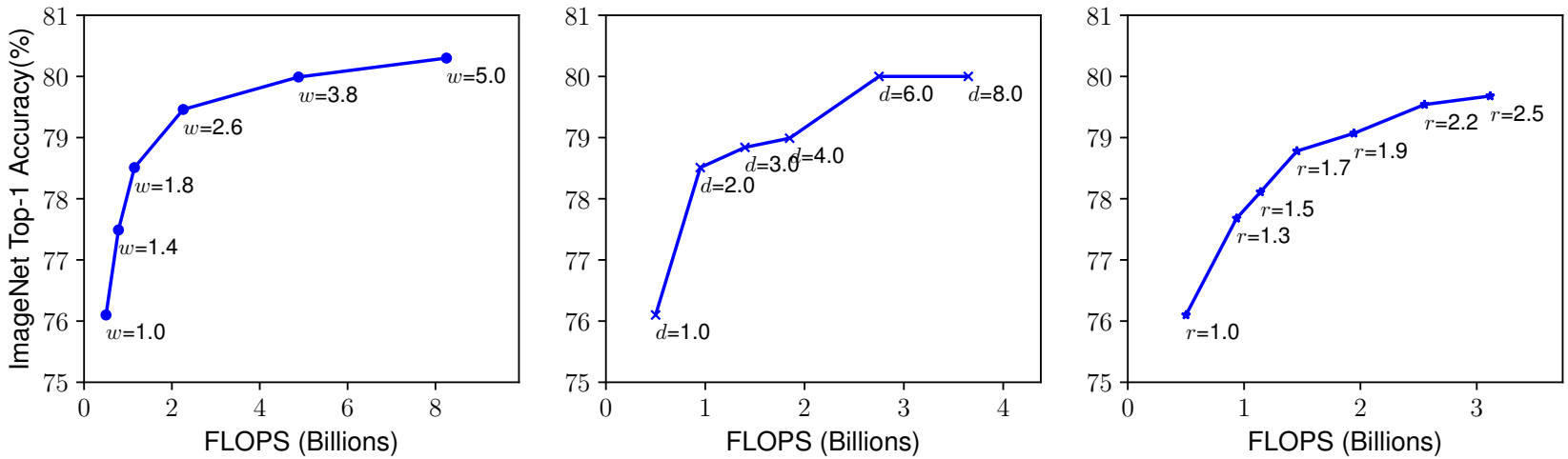

Рис. 3: Качество масштабирования по одному параметру: ширине, глубине, разрешению

- Вывод 1: Масштабирование сетей по одному из параметров увеличивает качество, однако с определённого значения выходит на плато.

Влияние размера модели на качество

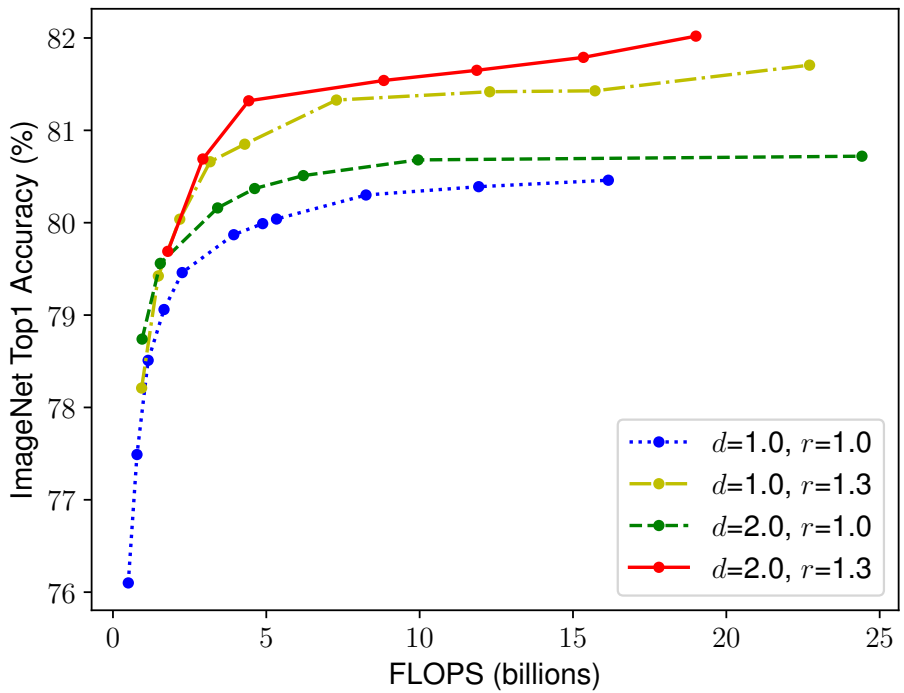

Рис. 4: Качество масштабирования по нескольким параметрам

Влияние размера модели на качество

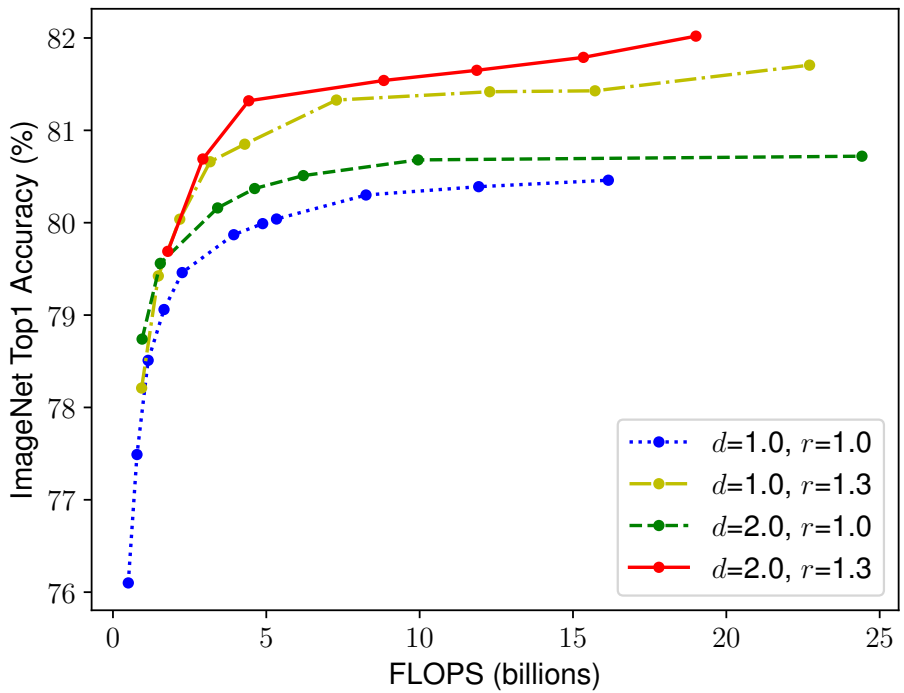

Рис. 4: Качество масштабирования по нескольким параметрам

- Вывод 2: Для оптимальной производительности надо одновременно увеличивать количество слоёв в сети, количество фильтров и разрешение в каком-то соотношении.

Compound scaling

- Главная формула и идея статьи. Сети надо масштабировать вот так:

Compound scaling

- Главная формула и идея статьи. Сети надо масштабировать вот так:

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

Compound scaling

- Главная формула и идея статьи. Сети надо масштабировать вот так:

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

- Что?

Compound scaling

- Главная формула и идея статьи. Сети надо масштабировать вот так:

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

- Что?
  - Хотим масштабировать сеть одним параметром (тремя сложно)

Compound scaling

- Главная формула и идея статьи. Сети надо масштабировать вот так:

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

- Что?
  - Хотим масштабировать сеть одним параметром (тремя сложно)
  - Как увеличится вычислительная сложность при увеличении глубины в 2 раза?

Compound scaling

- Главная формула и идея статьи. Сети надо масштабировать вот так:

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

- Что?
  - Хотим масштабировать сеть одним параметром (тремя сложно)
  - Как увеличится вычислительная сложность при увеличении глубины в 2 раза? (В 2 раза)

Compound scaling

- Главная формула и идея статьи. Сети надо масштабировать вот так:

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

- Что?
  - Хотим масштабировать сеть одним параметром (тремя сложно)
  - Как увеличится вычислительная сложность при увеличении глубины в 2 раза? (В 2 раза)
  - При увеличении ширины в 2 раза?

Compound scaling

- Главная формула и идея статьи. Сети надо масштабировать вот так:

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

- Что?
  - Хотим масштабировать сеть одним параметром (тремя сложно)
  - Как увеличится вычислительная сложность при увеличении глубины в 2 раза? (В 2 раза)
  - При увеличении ширины в 2 раза? (В 4 раза)

Compound scaling

- Главная формула и идея статьи. Сети надо масштабировать вот так:

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

- Что?
  - Хотим масштабировать сеть одним параметром (тремя сложно)
  - Как увеличится вычислительная сложность при увеличении глубины в 2 раза? (В 2 раза)
  - При увеличении ширины в 2 раза? (В 4 раза)
  - При увеличении разрешения входа в 2 раза?

Compound scaling

- Главная формула и идея статьи. Сети надо масштабировать вот так:

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

- Что?
  - Хотим масштабировать сеть одним параметром (тремя сложно)
  - Как увеличится вычислительная сложность при увеличении глубины в 2 раза? (В 2 раза)
  - При увеличении ширины в 2 раза? (В 4 раза)
  - При увеличении разрешения входа в 2 раза? (В 4 раза)

Compound scaling

- Главная формула и идея статьи. Сети надо масштабировать вот так:

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

- Что?
  - Хотим масштабировать сеть одним параметром (тремя сложно)
  - Как увеличится вычислительная сложность при увеличении глубины в 2 раза? (В 2 раза)
  - При увеличении ширины в 2 раза? (В 4 раза)
  - При увеличении разрешения входа в 2 раза? (В 4 раза)
  - В постановке (1), при увеличении ϕ на 1?

Compound scaling

- Главная формула и идея статьи. Сети надо масштабировать вот так:

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

- Что?
  - Хотим масштабировать сеть одним параметром (тремя сложно)
  - Как увеличится вычислительная сложность при увеличении глубины в 2 раза? (В 2 раза)
  - При увеличении ширины в 2 раза? (В 4 раза)
  - При увеличении разрешения входа в 2 раза? (В 4 раза)
  - В постановке (1), при увеличении ϕ на 1? (В 2 раза. Вся суть.)

Compound scaling

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

- Как выбирать α, β, γ?

Compound scaling

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

- Как выбирать α, β, γ?
  - Перебор по сетке, в статье
  - Оптимизация гиперпараметров: байесовские методы, генетические модели

Compound scaling

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

- Как выбирать α, β, γ?
  - Перебор по сетке, в статье
  - Оптимизация гиперпараметров: байесовские методы, генетические модели
- Общий алгоритм:
  1. Выбираем архитектуру сети. Например, существующую: ResNet, MobileNet.
  2. Фиксируем небольшое ϕ. Например, $\phi = 1$.
  3. Подбираем масштабы α, β, γ для данного ϕ.
  4. Увеличиваем ϕ в зависимости от доступных мощностей.

Сравнение методов масштабирования на существующих архитектурах

Таблица 1: **Scaling Up MobileNets and ResNet.**

Model	FLOPS	Top-1 Acc.
Baseline MobileNetV1	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	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	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%**

EfficientNet-B0

- Что такое NAS (Neural Architecture Search)?

EfficientNet-B0

- Что такое NAS (Neural Architecture Search)?
  - Архитектура сети – параметры для алгоритма оптимизации гиперпараметров.
  - Используются эволюционные алгоритмы, Reinforcement Learning.

EfficientNet-B0

- Что такое NAS (Neural Architecture Search)?
  - Архитектура сети – параметры для алгоритма оптимизации гиперпараметров.
  - Используются эволюционные алгоритмы, Reinforcement Learning.
- Выберем сеть при помощи NAS. Используем специальный таргет:

$$ACC(m) \times [FLOPS(m)/T]^w$$

$ACC(m)$ – Accuracy на ImageNet обученной модели, $FLOPS(m)$ – вычислительная сложность модели в количествах операций, T – цель по сложности, $w = -0.07$ – трейдофф между качеством и сложностью.

EfficientNet-B0

- Что такое NAS (Neural Architecture Search)?
  - Архитектура сети – параметры для алгоритма оптимизации гиперпараметров.
  - Используются эволюционные алгоритмы, Reinforcement Learning.
- Выберем сеть при помощи NAS. Используем специальный таргет:

$$ACC(m) \times [FLOPS(m)/T]^w$$

$ACC(m)$ – Accuracy на ImageNet обученной модели, $FLOPS(m)$ – вычислительная сложность модели в количествах операций, T – цель по сложности, $w = -0.07$ – трейдофф между качеством и сложностью.

- Экспонента! Хотим очень быструю, но не обязательно точную сеть. Зачем нам быстрая сеть?

EfficientNet-B0

- Что такое NAS (Neural Architecture Search)?
  - Архитектура сети – параметры для алгоритма оптимизации гиперпараметров.
  - Используются эволюционные алгоритмы, Reinforcement Learning.
- Выберем сеть при помощи NAS. Используем специальный таргет:

$$ACC(m) \times [FLOPS(m)/T]^w$$

  $ACC(m)$ – Accuracy на ImageNet обученной модели, $FLOPS(m)$ – вычислительная сложность модели в количествах операций, T – цель по сложности, $w = -0.07$ – трейдофф между качеством и сложностью.
- Экспонента! Хотим очень быструю, но не обязательно точную сеть. Зачем нам быстрая сеть?
  - Будем делать перебор α, β, γ. Масштабирование ϕ даст высокую точность потом.

Архитектура EfficientNet-B0

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

- При $\phi = 1$: $\alpha = 1.2, \beta = 1.1, \gamma = 1.15$, $\alpha \cdot \beta^2 \cdot \gamma^2 = 1.92027$
- Масштабирование с разными ϕ даёт сети с EfficientNet-B1 по B7.
- Почему не подбирать α, β, γ на больших сетях?

Архитектура EfficientNet-B0

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

- При $\phi = 1$: $\alpha = 1.2, \beta = 1.1, \gamma = 1.15$, $\alpha \cdot \beta^2 \cdot \gamma^2 = 1.92027$
- Масштабирование с разными ϕ даёт сети с EfficientNet-B1 по B7.
- Почему не подбирать α, β, γ на больших сетях?
  - Слишком большие, подбор займёт слишком много времени.

Результаты разных EfficientNet

Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPS	Ratio-to-EfficientNet
EfficientNet-B0	**77.3%**	**93.5%**	**5.3M**	**1x**	**0.39B**	**1x**
ResNet-50	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169	76.2%	93.2%	14M	2.6x	3.5B	8.9x
EfficientNet-B1	**79.2%**	**94.5%**	**7.8M**	**1x**	**0.70B**	**1x**
ResNet-152	77.8%	93.8%	60M	7.6x	11B	16x
DenseNet-264	77.9%	93.9%	34M	4.3x	6.0B	8.6x
Inception-v3	78.8%	94.4%	24M	3.0x	5.7B	8.1x
Xception	79.0%	94.5%	23M	3.0x	8.4B	12x
EfficientNet-B2	**80.3%**	**95.0%**	**9.2M**	**1x**	**1.0B**	**1x**
Inception-v4	80.0%	95.0%	48M	5.2x	13B	13x
Inception-resnet-v2	80.1%	95.1%	56M	6.1x	13B	13x
EfficientNet-B3	**81.7%**	**95.6%**	**12M**	**1x**	**1.8B**	**1x**
ResNeXt-101	80.9%	95.6%	84M	7.0x	32B	18x
PolyNet	81.3%	95.8%	92M	7.7x	35B	19x
EfficientNet-B4	**83.0%**	**96.3%**	**19M**	**1x**	**4.2B**	**1x**
SENet	82.7%	96.2%	146M	7.7x	42B	10x
NASNet-A	82.7%	96.2%	89M	4.7x	24B	5.7x
AmoebaNet-A	82.8%	96.1%	87M	4.6x	23B	5.5x
PNASNet	82.9%	96.2%	86M	4.5x	23B	6.0x
EfficientNet-B5	**83.7%**	**96.7%**	**30M**	**1x**	**9.9B**	**1x**
AmoebaNet-C	83.5%	96.5%	155M	5.2x	41B	4.1x
EfficientNet-B6	**84.2%**	**96.8%**	**43M**	**1x**	**19B**	**1x**
EfficientNet-B7	**84.4%**	**97.1%**	**66M**	**1x**	**37B**	**1x**
GPipe	84.3%	97.0%	557M	8.4x	-	-

We omit ensemble and multi-crop models, or models pretrained on 3.5B Instagram images.

Результаты на Transfer learning

	Comparison to best public-available results					
	Model	Acc.	#Param	Our Model	Acc.	#Param(ratio)
CIFAR-10	NASNet-A	98.0%	85M	EfficientNet-B0	98.1%	4M (21x)
CIFAR-100	NASNet-A	87.5%	85M	EfficientNet-B0	88.1%	4M (21x)
Birdsnap	Inception-v4	81.8%	41M	EfficientNet-B5	82.0%	28M (1.5x)
Stanford Cars	Inception-v4	93.4%	41M	EfficientNet-B3	93.6%	10M (4.1x)
Flowers	Inception-v4	98.5%	41M	EfficientNet-B5	98.5%	28M (1.5x)
FGVC Aircraft	Inception-v4	90.9%	41M	EfficientNet-B3	90.7%	10M (4.1x)
Oxford-IIIT Pets	ResNet-152	94.5%	58M	EfficientNet-B4	94.8%	17M (5.6x)
Food-101	Inception-v4	90.8%	41M	EfficientNet-B4	91.5%	17M (2.4x)
Geo-Mean						**(4.7x)**

	Comparison to best reported results					
	Model	Acc.	#Param	Our Model	Acc.	#Param(ratio)
CIFAR-10	†Gpipe	**99.0%**	556M	EfficientNet-B7	98.9%	64M (8.7x)
CIFAR-100	Gpipe	91.3%	556M	EfficientNet-B7	**91.7%**	64M (8.7x)
Birdsnap	GPipe	83.6%	556M	EfficientNet-B7	**84.3%**	64M (8.7x)
Stanford Cars	‡DAT	**94.8%**	-	EfficientNet-B7	94.7%	-
Flowers	DAT	97.7%	-	EfficientNet-B7	**98.8%**	-
FGVC Aircraft	DAT	92.9%	-	EfficientNet-B7	**92.9%**	-
Oxford-IIIT Pets	GPipe	**95.9%**	556M	EfficientNet-B6	95.4%	41M (14x)
Food-101	GPipe	93.0%	556M	EfficientNet-B7	**93.0%**	64M (8.7x)
Geo-Mean						**(9.6x)**

†GPipe trains giant models with specialized pipeline parallelism library.

‡DAT denotes domain adaptive transfer learning. Here we only compare ImageNet-based transfer learning results.

Почему модель лучше работает?

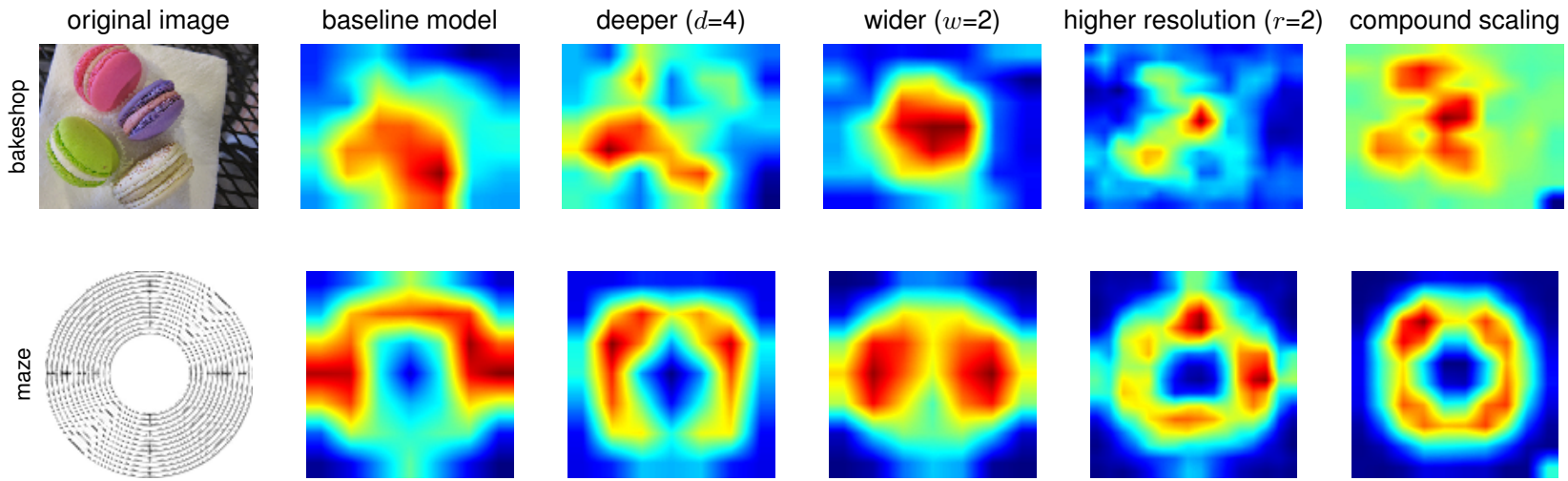

Рис. 5: **Class Activation Map (CAM)** для модели, увеличенной при помощи compound scaling (последняя колонка) показывает, что она фокусируется на релевантных объектах лучше, чем другие методы масштабирования.

Выводы

1. Сети можно эффективно масштабировать на основе compound rule, меняя один параметр:

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

2. Если увеличивать глубину, ширину и разрешение, то это надо делать совместно, иначе выигрыш в качестве быстро затухает.
3. Интересная SotA на ImageNet, SotA на 5 из 8 датасетов transfer learning.