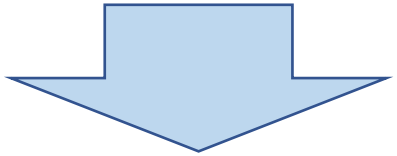
EfficientNetの紹介

- AlexNet以降は、**CNNモデルを大規模にスケールアップすることで精度を改善**するアプローチが主流となった（例：ResNet-18からResNet-200まである）
- 従来では、以下の変数（幅、深さ、解像度）を「適当」にスケールアップ
  - **幅**：1レイヤーのサイズ(ニューロンの数)
  - **深さ**：レイヤーの数
  - **解像度**：入力画像の大きさ
- 精度を向上できたものの、**モデルが複雑で高コスト**



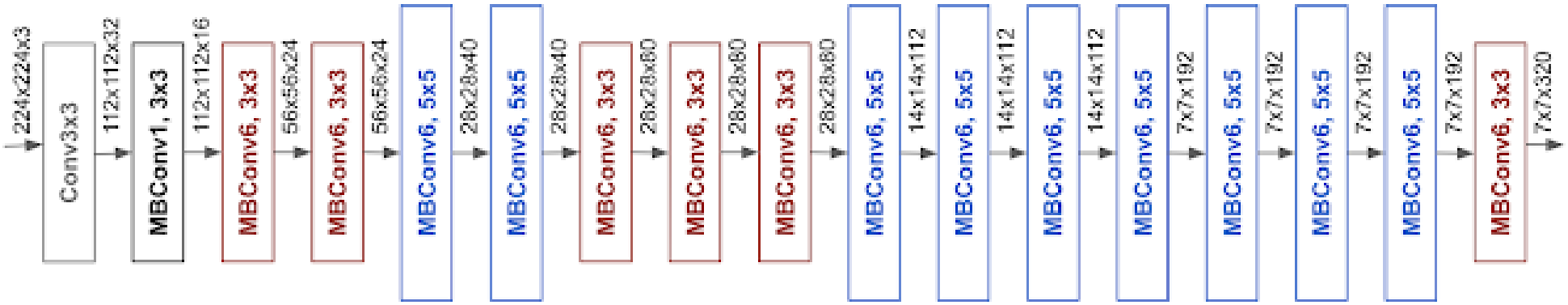
2019年に開発された**EfficientNet** モデル群は、効率的なスケールアップの規則を採用することで、**開発当時の最高水準の精度を上回り、同時にパラメータ数を大幅に減少**
（詳細は次頁）

EfficientNetとCNNモデルのスケールアップ

- ICML2019の論文で、新たなモデルスケーリングの「法則」が提案された
- 幅、深さ、解像度などを何倍増やすかは、**複合係数（Compound Coefficient）**を導入することで最適化

参考論文「EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks」解説; https://arxiv.org/abs/1905.11946

- EfficientNetではCompound Coefficientに基づいて、深さ・広さ・解像度を最適化したことにより、「小さなモデル」かつ高い精度を達成
- モデルが小さい（パラメータ数が少ない）→効率化（小型化と動作の高速化）



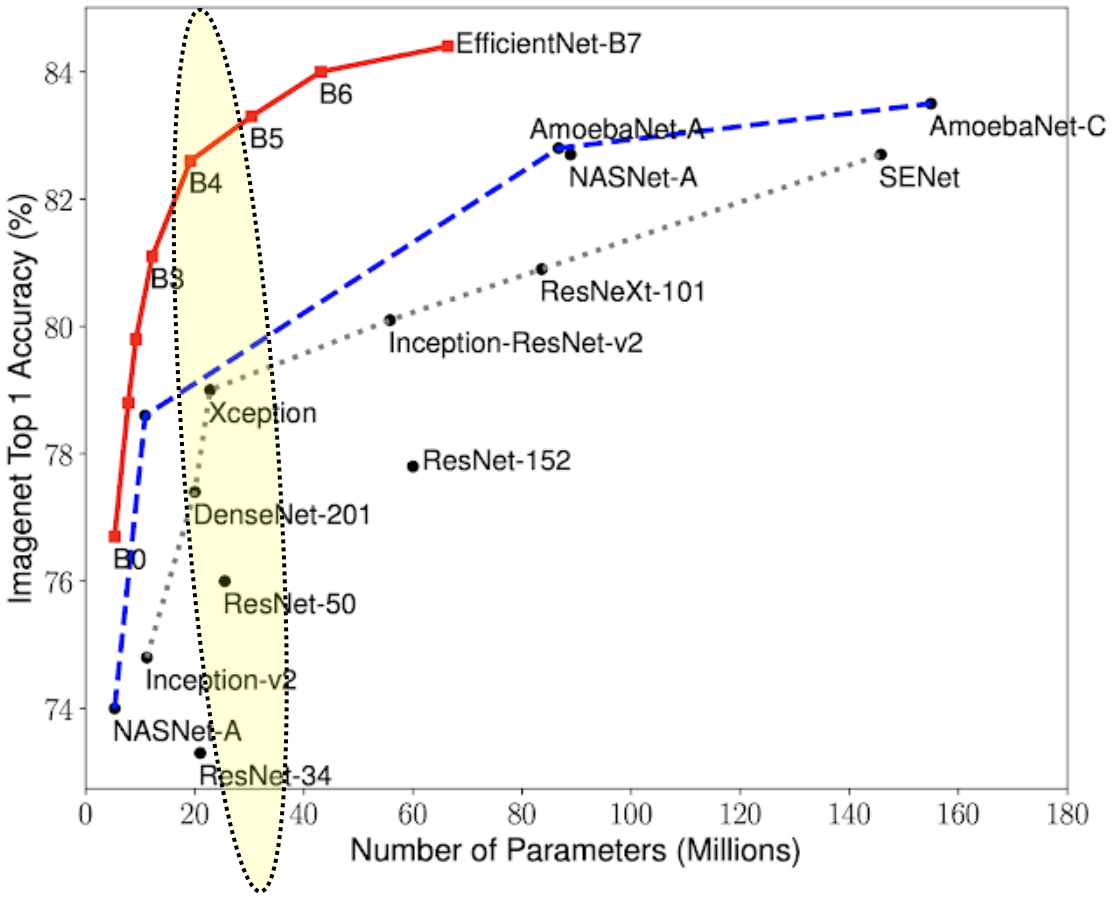
（右図）EfficientNet-B0 のアーキテクチャ
画像引用：Google AI Blog, https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html

EfficientNetの性能

（右図）ImageNetデータを用いた、既存のCNNとEfficientNet系モデルを比較した結果

- EfficientNetは精度と効率の両側面で優れている
- パラメータの数と計算量は数倍～１桁減少
- ResNet-50に比べてEfficientNet-B4は同程度の処理速度と計算量で精度が6.3%改善

- EfficientNetは、**転移学習でも性能を発揮**（シンプルかつ簡潔な構造、汎用性が高い）



https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html

EfficientNetモデルはオープンソースとして公開
（参考）EfficientNet の実装：
https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet

Compound Scaling Method（複合スケーリング手法）の詳細

以下のどれも、ある程度まで増やすと精度の向上は横ばい

- Depth (d) :
  - ネットワークの層を深くすることで、表現力を高くし、複雑な特徴表現を獲得できる
- Width (w) :
  - ユニット数を増やすことでより細かい特徴表現を獲得し、学習を高速化できる
  - しかし、深さに対してユニット数が大きすぎると、高レベルな特徴表現を獲得しにくくなる
- Resolution (r) :
  - 高解像度の入力画像を用いると、画像中の詳細なパターンを見出せる
  - 時代が進むにつれてより大きいサイズの入力画像が使われるようになってきた

- 畳み込み演算の**演算量FLOPSはd, w^2, r^2に比例**
（例：depthが2倍になるとFLOPSも2倍、widthとresolutionが2倍になるとFLOPSは4倍になる）

Compound Scaling Method（複合スケーリング手法）の詳細

- 3つのパラメータDepth、width、resolutionは、**単一の係数φで一様にスケーリング可能**
- α,β,γはグリッドサーチで求める定数
- φはユーザー指定パラメータ、モデルのスケーリングに使用できる計算リソースを制御する役割

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

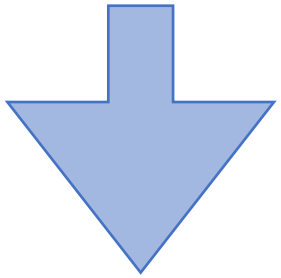
出典：https://arxiv.org/pdf/1905.11946.pdf

- **CNNでは畳み込み演算が計算コストを占領**するので、wとrが2乗のオーダーで効いてくる
  → d,w^2,r^2に比例する**FLOPSは～ $(\alpha \cdot \beta^2 \cdot \gamma^2)^\varphi$ 倍にスケール**する
  ※原論文では **$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$** という制約を設けており、**FLOPSは～$2^\varphi$で増加**すると近似できる

Compound Scaling Method（複合スケーリング手法）の詳細

・最適化問題とは
「与えられた制約の中である目的関数を大きく、または小さくする解を求めること」

・Compound Scaling Methodの目標
与えられたリソース制約に対してモデルの精度を最大化すること



最適化問題として定式化することができる

Compound Scaling Method（複合スケーリング手法）の詳細

$$
\begin{aligned}
\max_{d,w,r} \quad & Accuracy\big(\mathcal{N}(d,w,r)\big) \\
s.t. \quad & \mathcal{N}(d,w,r) = \bigodot_{i=1\ldots s} \hat{\mathcal{F}}_i^{d\cdot\hat{L}_i}\big(X_{\langle r\cdot\hat{H}_i, r\cdot\hat{W}_i, w\cdot\hat{C}_i\rangle}\big) \\
& \text{Memory}(\mathcal{N}) \leq \text{target\_memory} \\
& \text{FLOPS}(\mathcal{N}) \leq \text{target\_flops}
\end{aligned}
\tag{2}
$$

目標が精度を最大化すること（一番上の行）

Compound Scaling Method（複合スケーリング手法）の詳細

$$
\begin{aligned}
\max_{d,w,r} \quad & Accuracy\big(\mathcal{N}(d,w,r)\big) \\
s.t. \quad & \mathcal{N}(d,w,r) = \bigodot_{i=1\ldots s} \hat{\mathcal{F}}_i^{d\cdot\hat{L}_i}\big(X_{\langle r\cdot\hat{H}_i, r\cdot\hat{W}_i, w\cdot\hat{C}_i\rangle}\big) \\
& \text{Memory}(\mathcal{N}) \leq \text{target\_memory} \\
& \text{FLOPS}(\mathcal{N}) \leq \text{target\_flops}
\end{aligned}
\tag{2}
$$

$\mathcal{N}(d,w,r)$ はネットワークを表している
d, w , r はそれぞれ前説明した層の深さ、ユニット数、解像度をスケーリングするための係数

Compound Scaling Method（複合スケーリング手法）の詳細

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

・この式はResNetなどの畳み込みニューラルネットワークを定義している
いくつかのステージに分割され、各ステージのアーキテクチャは共通していることが多い

・i は何番目のステージかを表していて、Fは畳み込み層を表し、Lはステージの中で何回Fを繰り返すかを表している

・Hは入力時のheight、Wは入力時のwidth、Cはチャンネル数である

Compound Scaling Method（複合スケーリング手法）の詳細

$$\bigodot_{i=1\ldots s} \hat{\mathcal{F}}_i^{d\cdot\hat{L}_i}\left(X_{\langle r\cdot\hat{H}_i, r\cdot\hat{W}_i, w\cdot\hat{C}_i\rangle}\right)$$

・先程の式に、今回の最適化問題を解くためにd, w, rを導入する

・Lにdをかけて層の深さを調整している

・H, Wにrをかけて解像度を調整している

・Cにwをかけてチャンネル数を調整している

Compound Scaling Method（複合スケーリング手法）の詳細

$$
\begin{aligned}
\max_{d,w,r} \quad & Accuracy\big(\mathcal{N}(d,w,r)\big) \\
s.t. \quad & \mathcal{N}(d,w,r) = \bigodot_{i=1\ldots s} \hat{\mathcal{F}}_i^{d\cdot\hat{L}_i}\big(X_{\langle r\cdot\hat{H}_i, r\cdot\hat{W}_i, w\cdot\hat{C}_i\rangle}\big) \\
& \text{Memory}(\mathcal{N}) \leq \text{target\_memory} \\
& \text{FLOPS}(\mathcal{N}) \leq \text{target\_flops}
\end{aligned}
\tag{2}
$$

そして、メモリと畳み込み演算の計算量が設定した値以下で、
なおかつ高い精度を達成できるように最適化していく