

FReLU を導入した EfficientNet

日大生産工 ○木内 稜久 日大生産工 山内 ゆかり

1. まえがき

近年、画像認識において深層学習の Convolution Neural Network(CNN)がよく用いられている。モデルの精度向上のためCNNのスケールアップが広く研究されている。一般的にモデルのスケールアップの方法は、「広さ」「深さ」「解像度」の3種類あり、これまでの研究は3つの次元のうち1つだけをスケールアップすることが一般的であった。また、2次元や3次元をスケールアップすることは可能だが面倒な手動チューニングが必要で精度と効率が最適以下であった。そこでシンプルかつ効果的な複合スケールリング法の EfficientNet[1]が提案された。以前のモデルと異なり、ネットワークの幅、深さ、解像度を一様にスケールリングすることで効率的な精度の向上を達成している。

本研究では、畳み込み層で使用されている ReLUに代わり、空間的要因を考慮し2次元空間に拡張した活性化関数 FReLU[2]の導入を提案する。本研究は、ハードの性能とC言語で実装という都合によりNASの実装ができず、ベースモデルやブロック数、カーネルサイズの最適化が不可能である。そのため、MobileNet[3]をベースモデルとし EfficientNetの「3次元のスケールリング」 EfficientNetV2[4]の「Fused-MBConvと Progressive Learning」を加えたモデルに FReLUを導入し、これを提案手法とする。提案手法とNAS探索を除いた EfficientNetV2を比較し、効率性を保ちつつさらなる精度の向上を目指す。

2. 従来研究

2.1 MobileNet

MobileNet は畳み込みに Depthwise Convolution と Pointwise Convolution(1×1の畳み込み)のふたつを組み合わせた Depthwise Separable Convolution を使用している。標準的な畳み込みは両方のフィルタを1ステップで畳み込みをしているが、MobileNetでは2ステップで畳み込みをする。Depthwise Convolutionでは1つのフィルタに1つのチャンネルを対応させ、畳み込みを行う。畳み込み処理はチャンネルごとに独立している。Pointwise Convolutionでは単純に1×1の畳み込みを行う。MobileNetは標

準的な畳み込みに対し、パラメータ数が1/8～1/9に削減できる。

2.2 EfficientNet

EfficientNetは、ネットワークの「広さ」「深さ」、入力画像の「解像度」の3次元を一様にスケールリングするモデルである。従来モデルに比べ、パラメータ数は1/8、計算量は1/12の削減に成功している。複合係数 Φ を用いて、すべてのレイヤーの広さ・深さ・解像度の値を定数倍で一様にスケールリングする複合スケールリング法(1)を以下に示す。

Φ はユーザが使えるリソースによって決める値で、 α, β, γ はグリッドサーチによって決まる定数。 $(\alpha \cdot \beta^2 \cdot \gamma^2)^\Phi$ に従ってFLOPSが増えるため、 $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ という条件を決めることで増え方を 2^Φ にした。

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

レイヤー自体の処理は変えないため、ベースラインのモデルには性能が高いモデルを持つことも重要である。従来研究では、NASによるベースラインのモデル構築を行っているが、本研究では行わず MobileNet をベースモデルとする。

2.3 EfficientNetV2

EfficientNetV2は、EfficientNetの問題点を改善させパラメータ数を抑えたまま学習速度を向上させたモデルである。問題点、改善方法について紹介する。

2.3.1 大きな画像サイズの学習が遅い

画像サイズが大きい場合、メモリ使用量が大きく、小さなバッチサイズで学習させるため、学習が遅くなる。そこで、正則化を考慮した Progressive Learning を導入する。学習に小さい画像サイズを用いる。学習が進むにつれ、画像サイズと正則化の強さを段階的に大きくすることで改善する。

Introducing FReLU into EfficientNet

Riku KIUCHI and Yukari YAMAUCHI

2.3.2 深さ方向の畳み込みでは学習が遅い

EfficientNetで用いられるMBConvブロック内のDepthwise畳み込みはパラメータ数削減の役割があるが、初期の層で学習が遅い。1×1畳み込みとDepthwise畳み込みの代わりに3×3畳み込みを用いたFused-MBConvブロックを使用する。その構造をFigure1に示す。

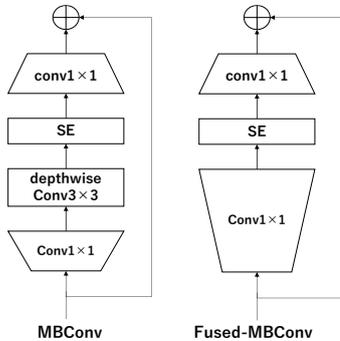


Figure 1 Fused-MBConv ブロックの構造

2.3.3 モデルを均等にスケールアップすることは最適でない

- ・ 画像サイズは最大で480
- ・ 層の数は後半の学習で増やす

これらの制限を追加して複合スケールリングを行う。[4]から、EfficientNetやNFNetと比較して短い学習時間で高い分類精度を出すことが分かる。

3. 提案手法

本研究では、EfficientNet内で使用されているReLU関数に代わり、FReLU関数を導入する。FReLU関数は空間的要因を考慮し、ReLUやPReLUを2次元空間に拡張した活性化関数である。ReLUと比較したFReLUの構造をFigure2に示す。

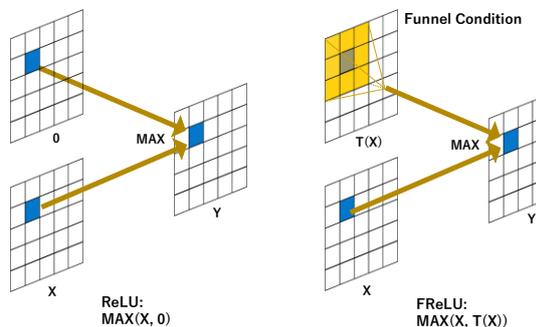


Figure 2 ReLU と FReLU の構造

FReLUは(2)により求めることができる。 $x_{e,i,j}$ は特徴マップにおける c 番目のチャンネル、位置 (i,j) のピクセル値を指す。 T はDepthwise畳み込みを行う2次元空間の関数である。

$$y = \max(x_{c,i,j}, T(x_{c,i,j})) \quad (2)$$

FReLU関数は(入力 x , x の畳み込み)の大きい数を出力する関数である。各ピクセルをそのまま用いるか、周辺情報を一緒に取り込むか、ピクセルごとの特徴を捉えて複雑な画像に対応できるようにする。

4. 実験および検討

データセットはMNISTを用いる。NASによって探索する部分を実装しておらず、従来手法と比較することが困難なため、複数回の比較を行い、それぞれの段階で比較する。比較項目は精度と計算時間とする。(0)MobileNet (1)MobileNetに対して複合スケールリング法を適用する。(2)Fused-MBConvブロックとProgressive Learningを適用する。(3)提案手法FReLUを適用する。

5. まとめ

本研究では、NAS探索を除いたEfficientNet V2に空間的要因を考慮し2次元空間に拡張した活性化関数FReLUを導入することを提案した。一般的な活性化関数に代わり、FReLUを適用することで、適用前と比較してEfficientNetV2で獲得した効率性を保ち、精度を高めることが予想される。NAS探索を実装できないため、[4]から精度が低下すると考えられるが、その制限下での比較を行う。今回の実験では、FReLUの適用で計算量が増えるため、実装時には処理時間と精度のトレードオフを調べ、FReLUを適用させるブロックを調整する必要がある。

参考文献

- [1] Mingxing Tan, Quoc V. Le., “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”, arXiv:1905.11946v5(2020)
- [2] Ningning Ma, Xiangyu Zhang, Jian Sun, “Funnel Activation for Visual Recognition”, arXiv:2007.11824v2(2020)
- [3] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”, arXiv:1704.04861v1(2017)
- [4] Mingxing Tan, Quoc V. Le., “EfficientNet V2: Smaller Models and Faster Training”, arXiv:2104.00298v3(2021)