ISSN 2186-5647

P-22

MLP-Mixer におけるパッチ拡張手法の提案

日大生産工　○加藤　暢弥　　日大生産工　山内　ゆかり

1. まえがき

近年、画像認識領域では、畳み込みニューラルネットワーク（CNN）やアテンション機構（Transformer）を用いたネットワークが優れた性能を発揮している。一方、MLP-Mixer[1]は畳み込みやアテンション機構を使わず、多層パーセプトロン（MLP）のみで画像を分割し、空間方向（トークン）とチャンネルごとに学習する層を追加することで、高い性能を示すアーキテクチャとして提案された。しかし、MLP-Mixerは誘導バイアスが弱いため、少量データや小規模モデルでの一般化と安定性に課題があり、さらにトークン数に比例して計算量とメモリ使用量が増大するという問題がある。

本研究では、従来のパッチング手法を拡張し、MLP-Mixerの精度・安定性・効率の向上を目指す。評価はMNISTで行い、ベースラインのMLP-Mixerと比較して性能がどう変化するかについて報告する。

2. 従来研究

2.1 多層パーセプトロン（MLP）

多層パーセプトロン（MLP: Multi-Layer Perceptron）は、複数のニューロン層から構成される順伝播型のニューラルネットワークである。その基本構造は、外部からのデータを受け取る入力層、最終的な処理結果を出力する出力層、そしてそれらの間に位置する一つ以上の隠れ層からなる。MLPの最大の特徴は、隣接する層のニューロン間が全結合している点にあり、これにより複雑な非線形問題への適応が可能となる。各ニューロンでは、前層からの入力値に重み付けを行い、バイアスを加えた後、活性化関数を適用して次層へと出力する。この一連の処理を順伝播と呼び、モデルの出力と正解ラベルとの誤差に基づき重みを更新するプロセスを誤差逆伝播法（Backpropagation）と呼ぶ。

2.2 MLP-Mixer

MLP-Mixer[1]は、MLPのみで構成されたアーキテクチャである。画像を大きさP×Pのパッチ（トークン）に分割し、各パッチを全結合層で埋め込みベクトルに変換し、チャンネルごとにパッチ数分複数の要素を混ぜる層（Token Mixing MLP層）と各トークン内で複数のチャンネル情報を混ぜる層（Channel Mixing MLP層）の２種類の層からなるMixer層を主な層として構成している。さらに、Mixer層に突入する前にパッチごとに全結合をする層、Mixer層を通過した後にチャンネルごとに平均を出して要素数をP×Pにする層（Global Average Pooling を行う層）、それをもとに全結合をする層、画像認識をする層からなる。Figure 1にMLP-mixerのアルゴリズムの図を示す。

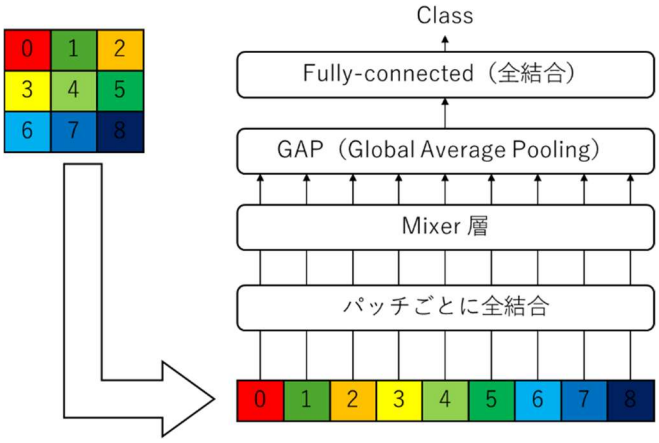

Figure 1　MLP-mixerのアルゴリズム

2.3 Mixer層

Mixer層は、Token Mixing MLP層とChannel Mixing MLP層を主体として構成されていて、さらに、転置する層、レイヤー正規化をする層、スキップ接続、ドロップアウトも含まれている。入力特徴量をXとすると、Mixer層の処理を式に表すと以下のようになる。

$$U_{*,i} = X_{*,i} + W_2\sigma\left(W_1 LayerNorm(X)_{*,i}\right), \quad (1)$$

$$\text{for } i = 1 \dots C$$

$$Y_{j,*} = U_{j,*} + W_4\sigma\left(W_3 LayerNorm(U)_{j,*}\right), \quad (2)$$

$$\text{for } i = 1 \dots S$$

式(1)は、トークン間の空間情報を混ぜ合わせるToken Mixingを、式(2)は各トークン内部の特徴量を混ぜ合わせるChannel Mixingをそれぞれ表している。具体的には、式(1)では入力層から受け取ったデータXをレイヤー正規化してパッチを基準にデータを分ける。その後、重みW_1と掛け合わせて活性化関数σを適用し、重みW_2と掛け合わせたものを先ほどのXと足し合わせることで、Uが出力されることを表している。また、式(2)は先ほどのデータUを正規化した後、

Improving MLP-Mixer with Dilated Patching
Nobuya KATO and Yukari YAMAUCHI

今度はチャンネルを基準に同様のMLP処理（W_3W_4）を行い、入力Uと足し合わせることで、最終的な出力Yを生成されることを表している ここで、Sはパッチ数（トークン数）、Cはチャンネル数、σは活性化関数GELU、Wは全結合層の重みを示している。

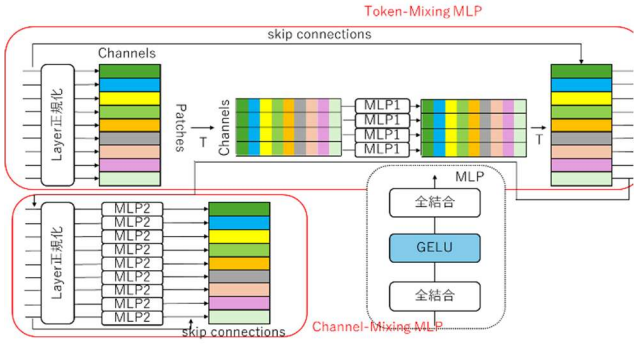

Figure 2　Mixer層のアルゴリズム

2.4　レイヤー正規化

レイヤー正規化（Layer Normalization）は、ニューラルネットワークの学習を安定化させるための手法である。バッチ正規化がミニバッチ内のサンプル間で統計量を計算するのに対し、レイヤー正規化は単一のサンプル内で、全チャンネルにわたって平均と分散を算出し正規化を行う。これにより、バッチサイズに依存することなく安定した学習が可能となり、また、試練時間を短縮し精度を向上させることができる。

2.5　GELU

GELU（Gaussian Error Linear Unit）[2]は、Transformerベースのモデルで標準的に利用される高性能な活性化関数であり、以下のように定義される。

$$GELU(x) = x \cdot \frac{1}{2}(1 + \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)) \qquad (3)$$

erf(x)は、以下の積分で定義される誤差関数である。

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt \qquad (4)$$

この誤差関数は、標準正規分布の累積分布関数を近似計算する役割を担っている。ReLUと比較して、GELUは全領域で微分可能で滑らかな関数であるため、勾配ベースの最適化がより安定しやすい。このような特性から、活性化関数にGELUを用いると、ReLUを用いたときよりもモデルの精度が向上するという特徴がある。

3.　提案手法

MLP-mixerは、優れた性能を発揮しているが、問題点が大きく分けて２つある。１つ目は、誘導バイアスが弱いという点である。このことによって、パッチ間の空間情報をゼロから学習する必要があり、特に少量データでの学習が不安定になりやすいという問題がある。２つ目は、計算コストが増大するという点である。このことによって、高解像度の画像や、細かいパッチ分割を行うと、計算量とメモリ使用量が爆発的に増加するという問題がある。そこで、拡張畳み込みを用いたパッチの改良をすることによって、従来手法では、局所的な部分しか一度に見ることができなかったが、離れた部分の位置関係をまとめて学習できるようにする。これによって、パッチの数も削減できるため、計算コストが削減でき、精度の向上に期待できる。

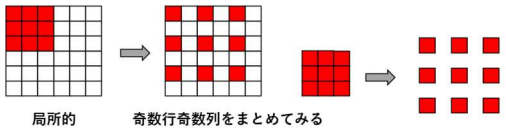

Figure 3 パッチの改良のイメージ図

4.　実験および検討

本研究の有効性を検証するため、提案手法とベースラインであるMLP-Mixerのテスト精度と平均訓練時間の両方で性能の比較を行った。実験には、手書き数字のデータセットであるMNISTを用いた。画像サイズは28×28ピクセルである。モデルの基本的なパラメータは両手法で共通とし、Mixerレイヤー数や隠れ層の次元などを統一した。従来手法では、パッチサイズを4×4とし、画像を一辺あたり7分割することで、49個のパッチを作成した。提案手法では、パッチのサンプリング画素数をベースラインと同じ4×4とし、拡張率（dilation rate）を2に設定した。これにより、1つのパッチが7×7の領域をカバーするため、画像を一辺あたり4分割し、16個のパッチを作成した。訓練は全訓練データのうち10000枚、評価は全テストデータのうち2500枚を用いて行い、10エポック学習させた際の平均訓練時間と最終的なテスト精度を測定する実験を行った結果をTable1に示す。

Table 1　MNISTでの実験結果

	Train ACC	Test ACC	Time
従来手法	0.943	0.868	269
提案手法	0.961	0.877	262

実験結果から、提案手法は従来手法と比較して、Train ACCは1.9%向上し、Test ACCは1.0%向上し、訓練時間が7秒短縮されることが分かる。この結果から、拡張畳み込みを用いたパッチ改良手法は、MLP-Mixerの課題であった誘導バイアスの弱さと計算コストを同時に改善し、精度向上と高速化に有効な手法であるといえる。

5. まとめ

本研究では、MLP-Mixerが持つ弱い誘導バイアスとパッチ数に依存する計算コストという2つの課題を解決するため、拡張畳み込みを用いたパッチの改良を行うことによる精度の向上を提案した。MNISTデータセットを用いた比較実験により、提案手法がベースラインのMLP-Mixerと比較して、テスト精度を向上させつつ、訓練時間を短縮できることを示した。

本研究では、拡張率を2に固定したが、データセットと特性やタスクに応じて最適な拡張率は異なると考えられる。そのため、今後は、複数の拡張率を試し、精度や計算コストがどう変化するかを調査し、実験を行いたい。

参考文献

1) Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., Lucic, M. and Dosovitskiy, A., "MLP-Mixer: An all-MLP Architecture for Vision," arXiv preprint arXiv:2105.01601, 2021.
2) Dan Hendrycks, Kevin Gimpel, “Gaussian Error Linear Units(GELUs)”,Trimmed version of 2016 draft(2020)pp.1