卒業論文

事前学習が Vision Transformer に与える影響

18A1066　桝田 修慎

指導教員　山口裕 助教

2022 年 2 月

福岡工業大学情報工学部情報工学科

概要

Vision Transformer を用いる.

キーワード　Vision Transformer

目次

第 1 章

序論

1.1　背景

近年，画像認識分野では，機械翻訳で脚光を浴びることになった Transformer[1] をコンピュータビジョンに適応させた Vision Transformer（以下 ViT と称する）が登場した [2]．ViT は，層を深くし畳み込みを行う畳み込みニューラルネットワーク（以下 CNN という）とは違い，畳み込み演算を Attention 機構を用いて代用しており，特に大規模なデータで事前学習を行なったときの，小・中規模の画像認識ベンチマーク（ImageNet，CIFAR-100，VTAB，etc.）では，過去の state-of-the-art の CNN モデルと比べて少ない計算リソースで訓練することができ、更に素晴らしい結果も出している．

本研究では，Vision Transformer が提案された論文「An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale」を参考にし，事前学習やデータ拡張の有無が，学習及び推論に与える影響を検証した．

1.2　本研究の目的

本研究の目的を以下に示す．

1. 一定の条件下での振る舞いを従来のモデル（VGG，ResNet）と比較し，ViT の優れている点・そうではない点を明らかにする．
2. 事前学習やデータ拡張が各モデルに及ぼす影響を調べる．

1.3　深層学習

深層学習とは，脳の神経回路を模したニューラルネットワークをより深くしたものを指し，入力データから有用な特徴量を自動で抽出する手法である．

1.3.1　畳み込みニューラルネットワーク

畳み込みニューラルネットワーク（CNN）は，入力した画像に対して重み行列（カーネル）を移動させながらスカラ積を求めていく畳み込み層を複数重ねているネットワークのことである．カーネルは画像全体に同じものを適用するため，CNN は移動させたカーネルと画像のスカラ積による局所性と，画像全体に渡る同一カーネルの重みの使用による移動不変性を持つ．

1.3.2　VGG[3]

VGG は CNN の一種で，3x3 という小さいカーネルを用いており，2014 年当時では珍しい 16 層及び 19 層の深いネットワークである．畳み込みとプーリング，線形層のシンプルなアーキテクチャであるにも関わらず，2014 年の ImageNet チャレンジのローカリゼーション・クラシフィケーションタスクで 1 位と 2 位を収めている．

1.3.3　ResNet[4]

一般的に，CNN では層を重ねることで，より高次元な特徴を抽出することができるが，層が深くなるにつれて勾配が発散・消失するという問題があった．しかし，ResNet は，畳み込み層を重ねるだけではなく，前の層，もしくはより浅い層の出力を次の層の入力とする残差機構（ショートカット結合）を取り入れることで，より畳み込み層を多く積み重ねながらも，SoTA を達成した．ResNet のブロックの一部を図 1.1 に示す．

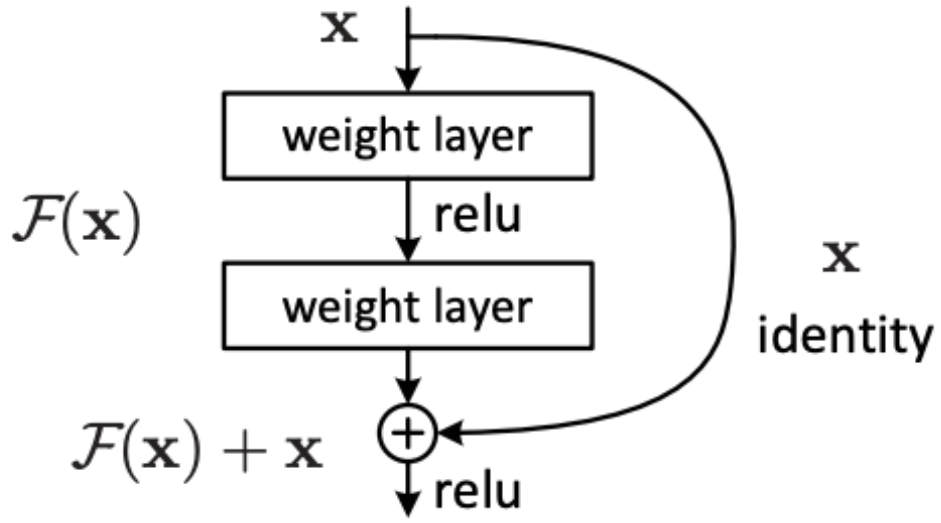

図 1.1. Residual learning

1.3.4　Transformer

Transformer は，それまで機械翻訳モデルで多く使われてきた畳み込みニューラルネットワーク・再帰ニューラルネットワークのような複雑なアーキテクチャを持つネットワークとは違い Attention 機構のみを用いて構成されているエンコーダ・デコーダモデルである．Transformer のアーキテクチャを図 1.2 に示す．

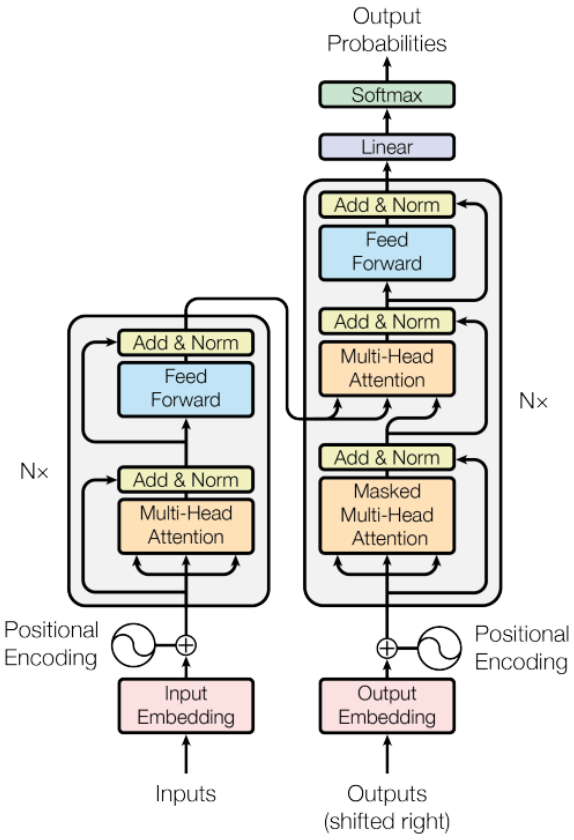

図 1.2. Transformer のアーキテクチャ

1.3.5　Vision Transformer

Vision Transformer は，機械翻訳で用いられていた Transformer をコンピュータビジョンに適応させたモデルであり，画像を複数のパッチに分割してそれぞれをベクトルとして埋め込み，平坦化して入力とする特徴がある．Vision Transformer のアーキテクチャを図 1.3 に示す．

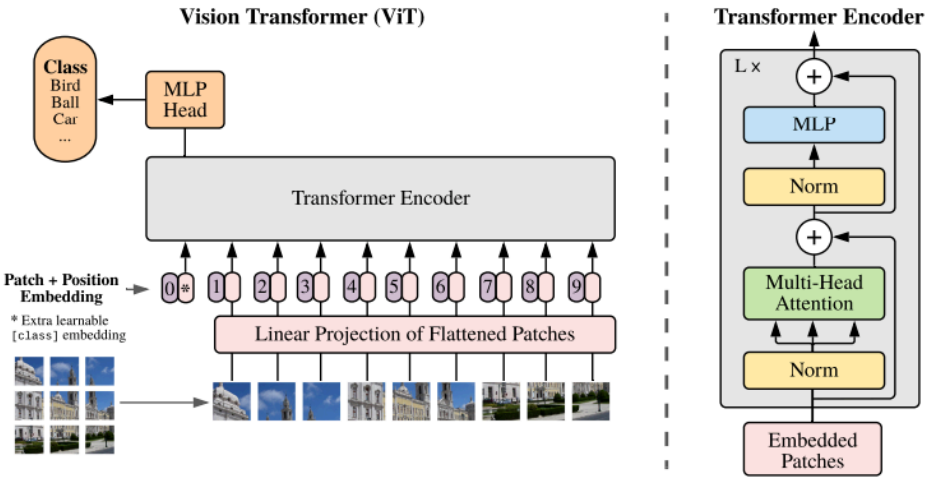

図 1.3. Vision Transformer のアーキテクチャ

1.4　使用ツール・実験環境

1.4.1　Python

Python は 1990 年代の始め，オランダにある Stichting Mathematisch Centrum で Guido van Rossum によって ABC と呼ばれる言語の後継言語として生み出された．Python はコードを簡潔に書くことができ，数値計算の NumPy，データ解析の Pandas など，専門的なライブラリが充実していることから，機械学習の研究開発をはじめとしたさまざまな分野で使用されている言語である．簡単に Python を始めるディストリビューションとして Anaconda がよく使われている．

1.4.2　PyTorch

PyTorch はディープラーニング・プロジェクトの構築を容易にする，Python のライブラリである．柔軟性を重視した設計であり，さらに，ディープラーニングモデルを Python の慣用的なクラスや関数の取り扱い方で実装できるようになっている．

1.4.3　Pytorch Image models

Pytorch Image Models（timm）は Ross Wightman によって作成されたディープラーニングライブラリであり，コンピュータビジョンの最先端のモデルが集められている．数行の記述でモデルを呼び出すことができ，必要に応じて書き換えることで，さまざまなタスクに適用できる．

1.4.4　Albumentations

Albumentations は Python のためのデータ拡張のライブラリである．モデルの汎化性能を上げるために行うデータ拡張のメソッドを多数揃えており，パイプラインを構成するとデータを効率的に拡張できる．

1.4.5　ImageNet

ImageNet はディープラーニング研究のために無償で利用できる大規模なデータセットのことで，モデルの性能を測るためのベンチマークとして使われることが多い．

1.4.6　Kaggle

Kaggle は世界規模のデータサイエンスのプラットフォームであり，世界中のデータサイエンティストが技術を競う場である．本実験では Kaggle が提供している Notebook の GPU を利用して学習及び推論を行う．

1.4.7　Plant Pathology 2021 - FGVC8

本実験では Kaggle 上で提供されているデータセットの Plant Pathology2021 を用いる．このデータセットの構成を表 1.1 に示す．

表 1.1. データセットの構成

データセットの構成	説明
test_images	3 枚の画像
train_images	18,632 枚の画像
sample_submission.csv	提出用のサンプル csv ファイル
train.csv	image，labels の 2 カラムの csv ファイル

1.5　論文の構成

論文の構成を書く．

第 2 章

実験モデル

2.1　ネットワークモデル

ViT の入力は画像である．画像サイズを H，W，チャネル数を C とすると，入力 x は式 (2.1) のように表せる．

$$x \in \mathbb{R}^{H \times W \times C} \tag{2.1}$$

また，二次元画像を扱うために，二次元の平坦化したパッチに整形する．式 (2.2) に示す．

$$x_p \in \mathbb{R}^{N \times (P^2 \times C)} \tag{2.2}$$

ここで，N はパッチ数であり，P はパッチサイズである．また，N は H，W，P を用いて式 (2.3) のように表せる．

$$N = \frac{HW}{P^2} \tag{2.3}$$

ViT のエンコーダに入力するには，x_p をさらに埋め込む必要がある．長さ $P^2 \times C$ を D 次元のベクトルとして，線形投影したものを埋め込みパッチ z_0 とする．また，入力データの先頭には [class] トークンを付与し，埋め込みパッチ $z_0^0 = x_{class}$ とする．式 (2.4) に示す．

$$z_0 = [x_{class};\ x_p^1 E;\ x_p^2 E;\cdots;\ x_p^N E] + E_{pos}, \qquad E \in \mathbb{R}^{(P^2 \times C) \times D},\ E_{pos} \in \mathbb{R}^{(N+1)) \times D} \tag{2.4}$$

ここで，E はパッチを D 次元のベクトルへの埋め込みを示し，E_{pos} は各パッチの位置が一意に定まるように情報を付加する位置エンコーディングを表す．上式で得られた入力 z_0 は Multi-Head Attention（以下，MSA と称する）に入力される．MSA では，Attention を求める計算を複数回行う．Attention の算出式を式 (2.5) に示す．

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.5}$$

初めに，入力ベクトルから Q，K，V ベクトルを生成する．このとき，Q，K の次元は d_k で，V の次元は d_k である．計算時は，Q，K の要素積を $\sqrt{d_k}$ で除算し，softmax 関数で 0 から 1 の値にする．最後に V をかけると Attention スコアが算出できる．Attention の算出過程を図 2.1 に示す．

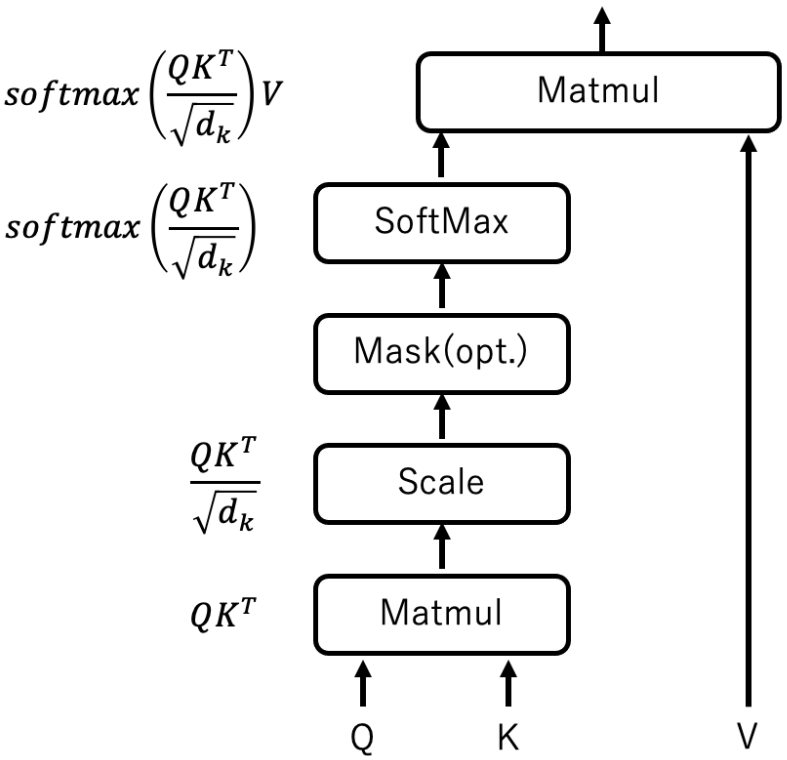

図 2.1. Attention の算出過程

2.2　手順

実験の条件を表 2.1 に示す.

表 2.1. 実験の条件

条件	事前学習	データ拡張
条件 1	なし	なし
条件 2	なし	あり
条件 3	あり	あり

- 条件 1: 事前学習なし・データ拡張なし
- 条件 2: 事前学習なし・データ拡張あり
- 条件 3: 事前学習あり・データ拡張あり

実験手順を以下に示す.

1. ステップ 1
2. ステップ 2

第 3 章

実験結果

実験結果をに示す.
条件ごとの結果を表 3.1 に示す.

表 3.1. 条件ごとの実験結果

条件	事前学習	データ拡張
条件 1	なし	なし
条件 2	なし	あり
条件 3	あり	あり

第 4 章

議論

議論を書く.

第 5 章

結論

結論を書く.

謝辞

謝辞を書く．

参考文献

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[3] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

付録 A

実験結果の図

付録があればここに書く.