

第 9 章 畳み込みネットワーク

眞家 佳悟
Keigo Maie

1 はじめに

本稿では、畳み込みネットワーク (以下, CNN) について述べる。はじめに, CNN の概要を 2 章で述べる。次に, CNN の主要な構成要素である入力データの特徴量を抽出する畳み込み処理について 3 章で, 計算効率の改善を行うプーリングについて 4 章で述べる。また, 畳み込み処理の変種であるタイル型畳み込みとゼロパディングについて 5 章で述べる。最後に, 実際に使用されている CNN から AlexNet と Residual Neural Networks (以下, ResNet) について 6 章で紹介する。

2 CNN

CNN は, 格子状データの処理に利用する NN である。また, CNN は, 物体認識や自動運転などの基本技術として注目されており, 特に画像処理系の応用に適性がある。基本的に CNN の主要な構成要素である畳み込み層は, 畳み込み, 非線形活性化関数, プーリングで構成される。ここで, 非線形活性化関数は, 同書の 6.3 節にて詳細に取り扱われているため, 本稿では割愛する¹⁾。

3 畳み込み処理

3.1 仕組み

畳み込み処理とは, 実数の引数を 2 つ持つ関数に対する処理と定義されている数学的な処理の 1 つである。本稿では, 入力に対してカーネルを適用することで特徴マップを出力する手法を指す。ここで, カーネルとはフィルタと称されることもある学習アルゴリズムにより学習したパラメータの多次元配列である。ここで, 2 次元データにおける畳み込みの式を式 (1) に, 2 次元データにおける畳み込み処理のイメージ図を Figure 1 に示す。

$$s(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n) \quad (1)$$

I : 入力の各要素

K : カーネルのパラメータ

i, j : 出力のピクセル

m, n : 入力のピクセル

Figure 1 に示すように入力データに対してカーネルを赤から青, 青から緑に移動させながら演算を行う。この移動の間隔をストライドという。具体的な演算方法は, カーネルの要素と入力データの対応する要素を乗算し積の和を求め, 対応する場所に格納する。また, 入力がカーネルと類似した大小パターンを持つ場合, 出力の値が大きくなり, 特徴抽出が可能となる。

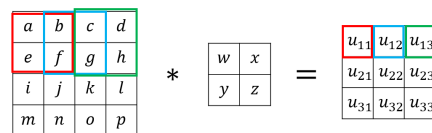


Figure 1 2次元データにおける畳み込み処理のイメージ

3.2 畳み込み処理の考え方

畳み込みは, 疎な結合, パラメータ共有, 等価表現と呼ばれる 3 つの考え方を利用している。

● 疎な結合

NN は全ての出力ユニットが全ての入力ユニットから影響を受ける。しかし, CNN においては, 入力より小さなカーネルを利用することで, より少ない計算処理で意味のある特徴を抽出可能である。

● パラメータ共有

NN はある層の出力を計算するときカーネルの各要素を一度のみ使用するが, パラメータ共有はあるモデルに関する複数の関数で同じパラメータを利用することでモデルの保存に必要なメモリ容量を削減する。CNN においては, カーネルの各要素が全ての入力要素で利用される。

● 等価表現

畳み込みの場合, パラメータ共有によってその層が移動に対する等価性という性質を持つ。等価とは入力に変化した場合に出力が同じように変化することである。 $f(g(x)) = g(f(x))$ が成立する場合, 関数 $f(x)$ は g と等価である。この性質は, 近接する少数のピクセルの何らかの関数を, 複数の入力位置に適用することが役に立つと知っているときに有用である。例えば, 画像を処理する場合, 同じエッジは多かれ少なかれ画像のほとんどの場所で現れるため, 画像全体でパラメータを共有することは実用的である。

4 プーリング

プーリングは, ある場所におけるネットワークの出力を周辺出力の要約統計量に置換する処理である。代表的なプーリングに Max プーリングと Average プーリングがある。Max プーリングのイメージ図を Figure 2 に示す。Figure 2 に示すように 4×4 の行列を 2×2 の行列 4 つに分割し最大値を出力とする。Average プーリングの場合は, 最大値ではなく平均値を出力する。このような処理により, 出力の特徴量が入力の特徴量に比べて減少するため,

計算効率が向上する。また、入力が多変化した場合においてプーリングを通過した出力はほとんど変化しない。そのため、局所的な範囲において特徴の位置より特徴が存在するかに注目することが可能である。

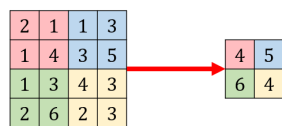


Figure 2 Max プーリングのイメージ図

5 畳み込み処理の変種

5.1 ゼロパディング

ゼロパディングは、各層において入力の周りをゼロで囲み、入力の幅を広げることで、CNN の多層化を可能にした処理の 1 つである。ゼロパディングのイメージ図を Figure 3 に示す。ゼロパディングを用いない畳み込み処理を行った場合、 4×4 から 3×3 に出力サイズを縮小する。これを連続して行くと出力サイズが 1×1 となり畳み込み処理が実行不可能となる。ここで、出力サイズの縮小はより小さなカーネルを利用することで緩和されるが、小さなカーネルでは表現力が低下する。そこで、ゼロパディングを用いることで出力サイズの縮小を軽減することが可能となる。ゼロパディングを用いて畳み込み処理を行う場合、入力の周りにゼロを加え、入力サイズを 6×6 とすることで出力サイズが 5×5 となり、出力サイズの縮小を緩和する。また、ゼロパディングを行うことで入力の端に対する畳み込み回数が増加するため、端の特徴を考慮可能になる。

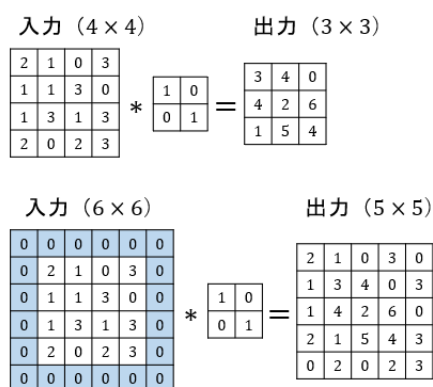


Figure 3 ゼロパディング未使用の場合（上）と使用した場合（下）のイメージ図

5.2 タイル型畳み込み

タイル型畳み込みは、近傍の場所は異なるカーネルを持ち、これを入力全体に繰り返し利用する畳み込み関数である。タイル型畳み込みの例と畳み込みの例を Figure 4 に示す。タイル型畳み込みは t 種類の異なるカーネルを有する。Figure 4 では、 $t=2$ とした場合を示しており、一方の

カーネルは”w” と”x” のラベルを割り振ったエッジを持ち、他方のカーネルは”y” と”z” のラベルを割り振ったエッジを持つ。出力を 1 ピクセルずらすごとに別のカーネルを利用して、出力の近傍ユニットは異なるパラメータを持つことになる。利用可能な t 個のカーネルを使用後、再び最初のカーネルに戻って繰り返し利用する。畳み込みは、”w” と”x” でラベル付けされた重み付きのカーネルを使用する。カーネルは 1 種類のみを使用し、すべての場所で適用される。よって、畳み込みは $t=1$ のタイル型畳み込みと同一である。

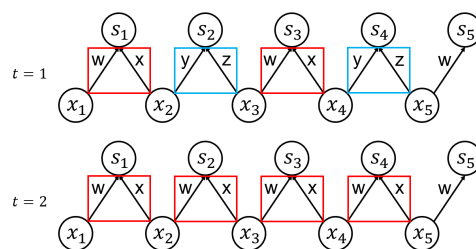


Figure 4 タイル型畳み込み（上）と畳み込み（下）の例

6 CNN の紹介

6.1 AlexNet

AlexNet は、ImageNet で物体認識精度を競う「IS-LVRC 2012」にて圧倒的優勝を果たした CNN である²⁾。具体的には、2 位よりも 10.8% 以上低いエラー率 15.3% で優勝した。主な特徴は、以下の 3 点である。

- GPU での学習
- 活性化関数に ReLU を利用
- ドロップアウトの利用

ここで、ReLU は同書の 6.3 節、ドロップアウトは同書の 7.12 節にて詳細に取り扱われているため、本稿では詳細を割愛する¹⁾。

6.2 ResNet

ResNet は、多層化を行った際に生じる勾配消失問題を残差ブロックにより解決した CNN の 1 つである³⁾。ここで、勾配消失問題とは、層が深くなるにつれ、誤差が減衰し、勾配が 0 に近づいてしまうことで学習が停滞する問題である。残差ブロックのイメージ図を Figure 5 に示す。Figure 5 において、入力を x 、畳み込み層の出力を $F(x)$ とすると、ここで、残差ブロックでは、 x を出力に直接加算するため、最終的な出力 $H(x)$ は $F(x) + x$ となる。また、 L を損失関数とすると、Figure 5 に示す残差ブロックに伝わってきた誤差は $\frac{\partial L}{\partial H}$ であり、残差ブロックで伝播する誤差は $\frac{\partial L}{\partial x}$ である。 $\frac{\partial L}{\partial x}$ を連鎖率を用いて変形すると式 (2) になる。

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial H} \frac{\partial F}{\partial x} + \frac{\partial L}{\partial H} \quad (2)$$

x : 入力

F : 畳み込み層の出力

H : ブロック出力

L : 損失関数

$\frac{\partial L}{\partial H}$ は伝播してきた誤差であるため、式 (2) より、誤差が減衰することなく伝播されていることが分かる。この仕組みにより、勾配消失問題を解決し、ResNet は「ILSVRC 2015」にて前述した AlexNet より 11.7% 低いエラー率 3.6% で優勝を果たした。

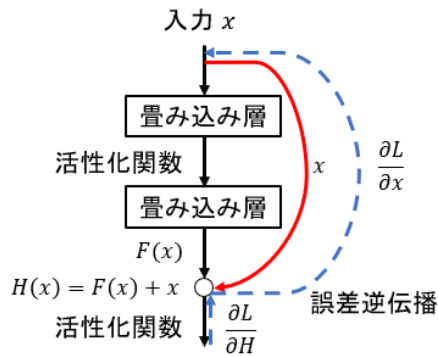


Figure 5 残差ブロックのイメージ図

7 まとめ

本稿では、畳み込みネットワークについて述べた。はじめに、CNN の概要を 2 章で述べた。次に、CNN の主要な構成要素である入力データの特徴量を抽出する畳み込み処理について 3 章で、計算効率の改善を行うプーリングについて 4 章で述べた。また、畳み込み処理の変種であるタイル型畳み込みとゼロパディングについて 5 章で述べた。最後に、実際に使用されている CNN から AlexNet と ResNet について 6 章で紹介した。

参考文献

- 1) Ian Goodfellow et al. 深層学習. 岩澤有祐ほか監訳. 株式会社ドワンゴ, 2018, 582p.
- 2) Krizhevsky, Alex, Ilya Sutslever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
- 3) He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.