CNN

ゼロから作る Deep learning 7章

藤本 09.10.20

CNNの構造

- CNN (convolutional neural network)

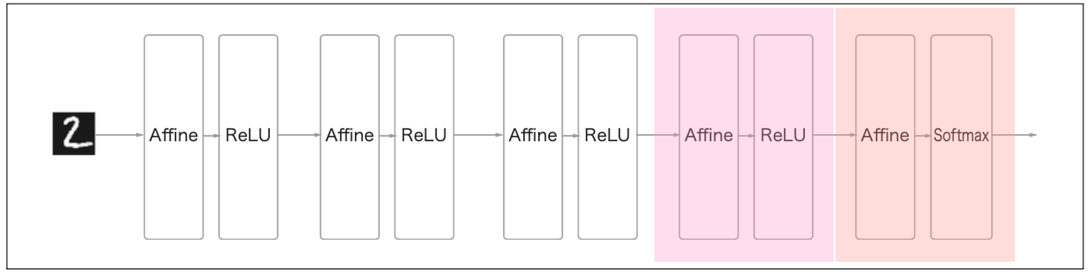

図7-1　全結合層（Affine レイヤ）によるネットワークの例

Affine - ReLU -> Convolution - ReLU - (Pooling)

出力に近い層　最後の出力層

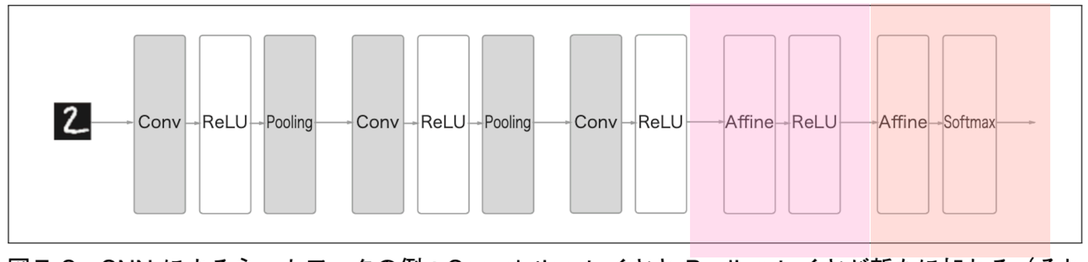

図7-2　CNN によるネットワークの例：Convolution レイヤと Pooling レイヤが新たに加わる（それぞれ背景が灰色の矩形で描画）

- 今までは、隣接する層の全てのニューロン間で結合があった（**全結合**）
  全部Affineレイヤで結合
- ConvolutionレイヤとPoolingレイヤを追加

全結合の問題点

- データの形状が無視されてしまう。
  (Affineレイヤでは、画像のような3次元データ (縦、横、チャンネル)を1列に並べて入力)

- Convolutionレイヤでは、形状を維持して入力する。

  特徴マップ (feature map) : Convolutionレイヤの入出力データ

畳み込み演算

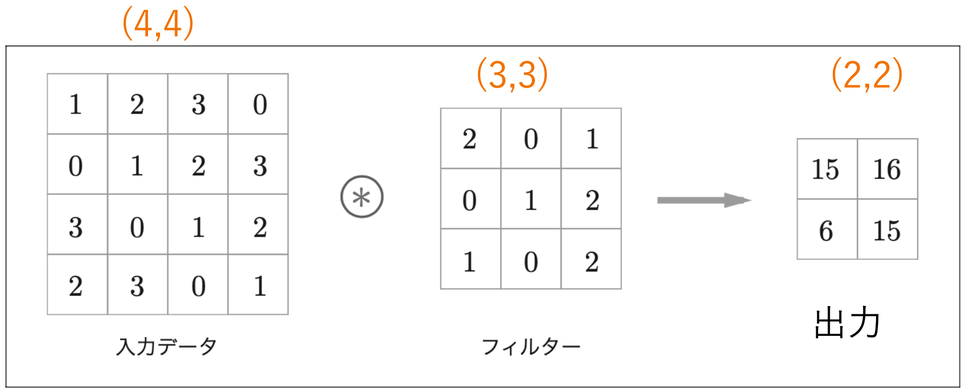

図7-3　畳み込み演算の例：畳み込み演算を「⊛」記号で表記

- フィルター(3,3)をずらしながら積和演算をして、出力の対応する場所に格納する。

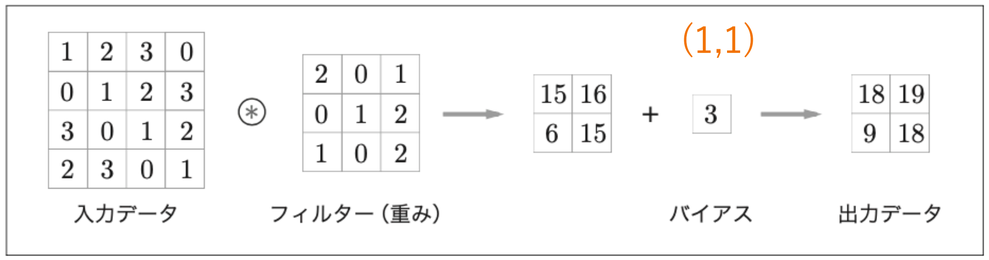

図7-5　畳み込み演算のバイアス：フィルターの適用後の要素に固定の値（バイアス）を加算する

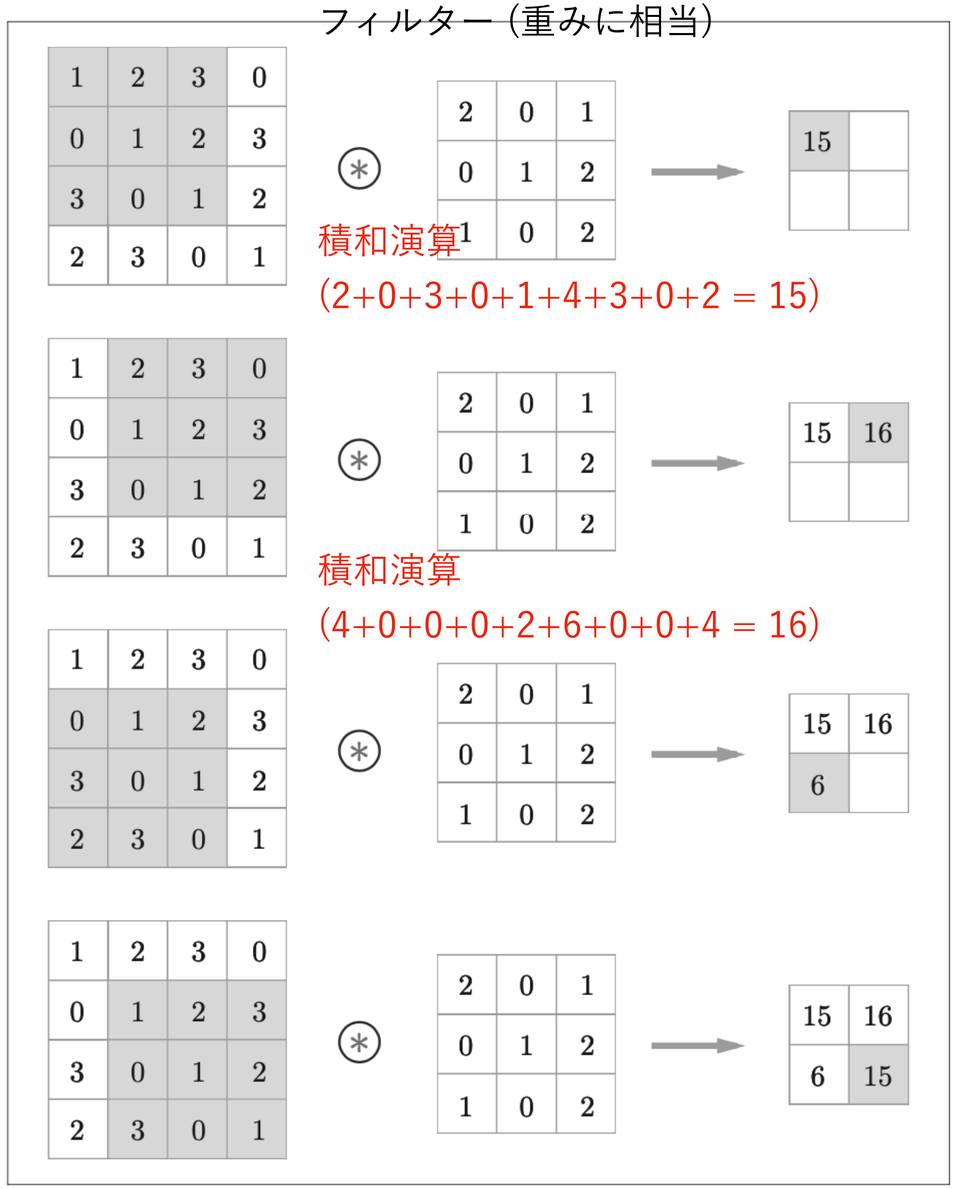

図7-4　畳み込み演算の計算手順

Padding

幅1のpaddingで入力サイズは(6,6)になる

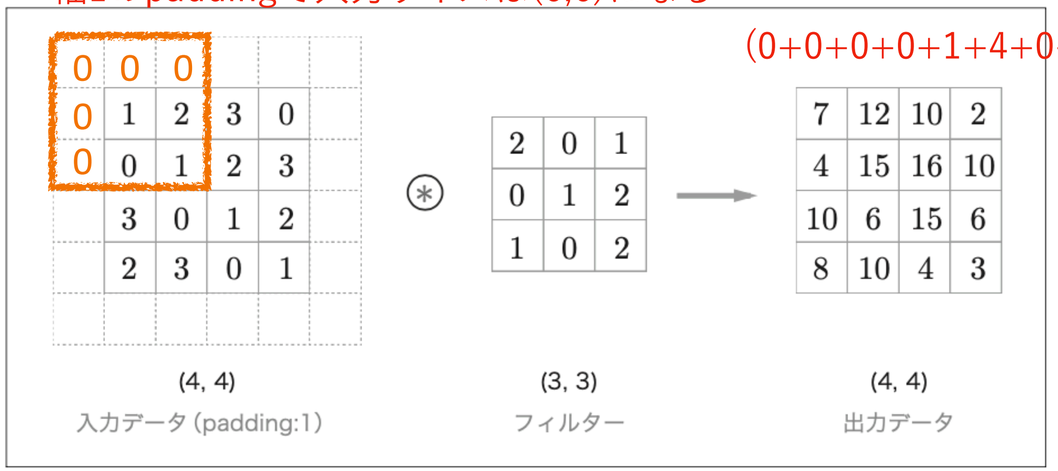

図7-6　畳み込み演算のパディング処理：入力データの周囲に 0 を埋める（図ではパディングを破線で表し、中身の「0」の記載は省略する）

- 出力サイズを調整するために、固定の値 (0とか)で周囲を埋める

- Paddingしなかったら出力が (2,2) になって、
  入力サイズから縮小されてしまう
  -> 空間的なサイズを一定にしたまま次の層にデータを渡す。
  　　繰り返すとそのうち(1,1)になるのを防ぐ

Stride

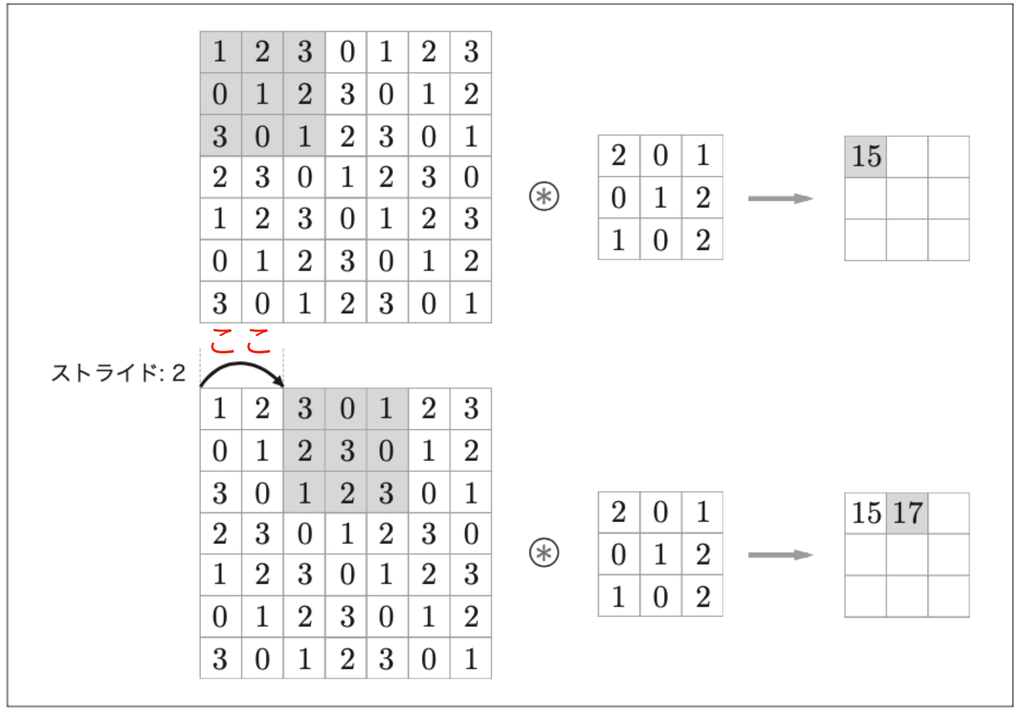

図7-7　ストライドが２の畳み込み演算の例

- フィルターをどれだけずらすか
- Strideを大きくすると、出力サイズは小さくなる。

Padding & Stride

入力 Height　Filter Height
Padding
出力 Height

$$OH = \frac{H + 2P - FH}{S} + 1$$

Stride
入力 Width　Filter Width
出力 Weight

$$OW = \frac{W + 2P - FW}{S} + 1$$

- 注意：割り切れるように値を設定する。
  割り切れない場合はエラーを出力するようにする。

3次元の場合

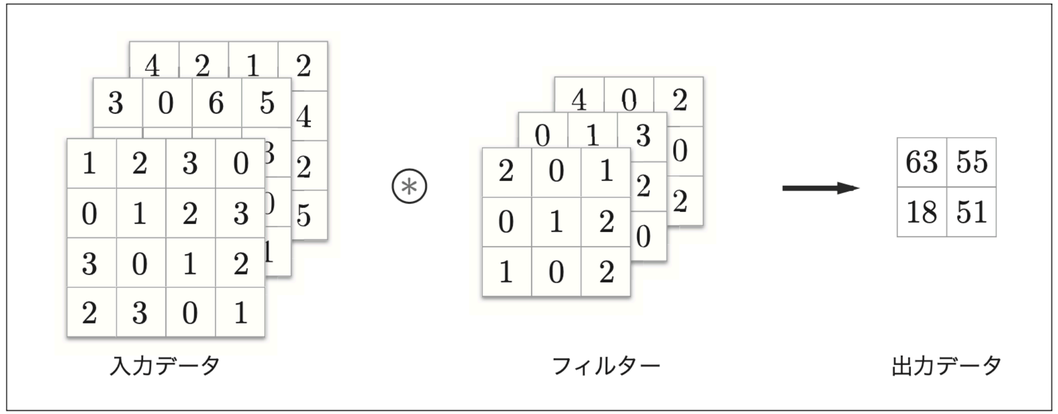

図7-8　3 次元データに対する畳み込み演算の例

- チャンネル方向も加えて、3次元
- チャンネル方向に複数の特徴マップがある場合、
  チャンネルごとに畳み込み演算をして、それらの結果を加算する。
- 入力データのチャンネル数とフィルターのチャンネル数は同じ。

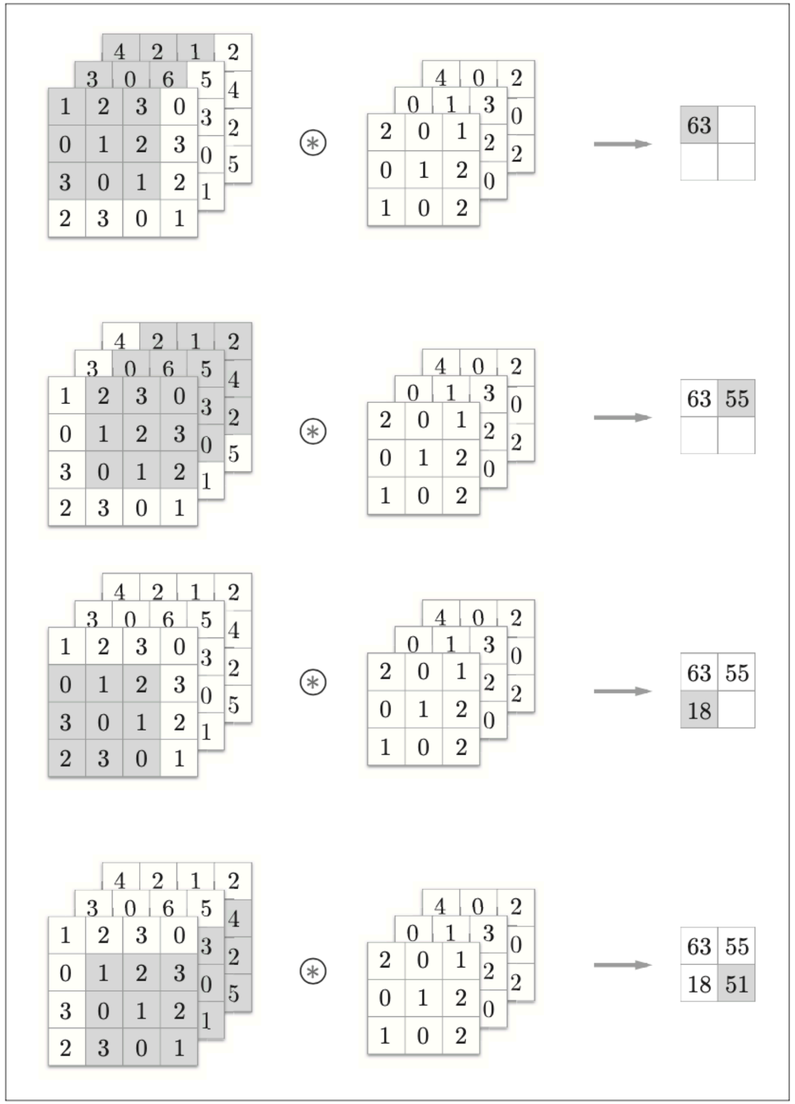

図7-9　3 次元データに対する畳み込み演算の計算手順

3次元の場合：ブロックで考える

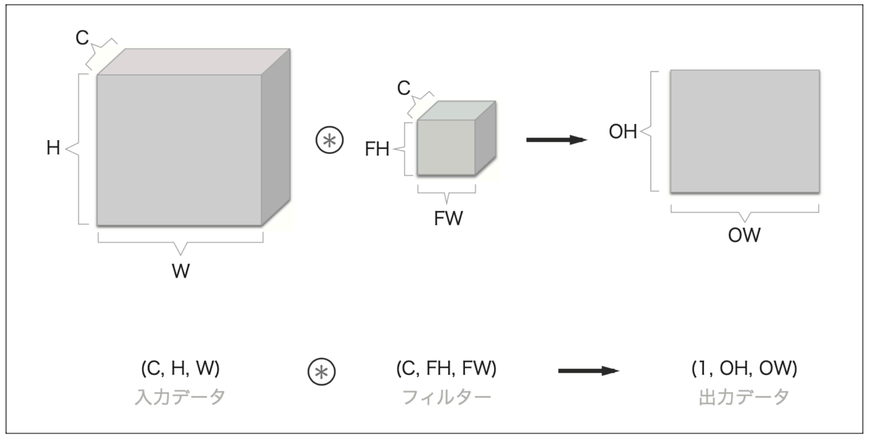

図7-10　畳み込み演算をブロックで考える。ブロックの形状に注意

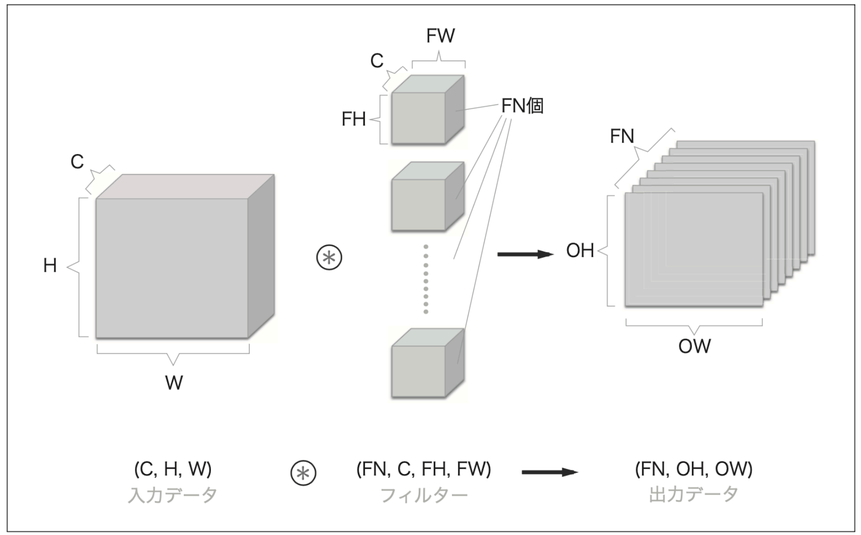

図7-11　複数のフィルターによる畳み込み演算の例

- 出力は channel方向1の特徴マップ
- 出力を channel方向も複数にしたかったら、
  フィルター(重み)を複数にする
- あとバイアスも

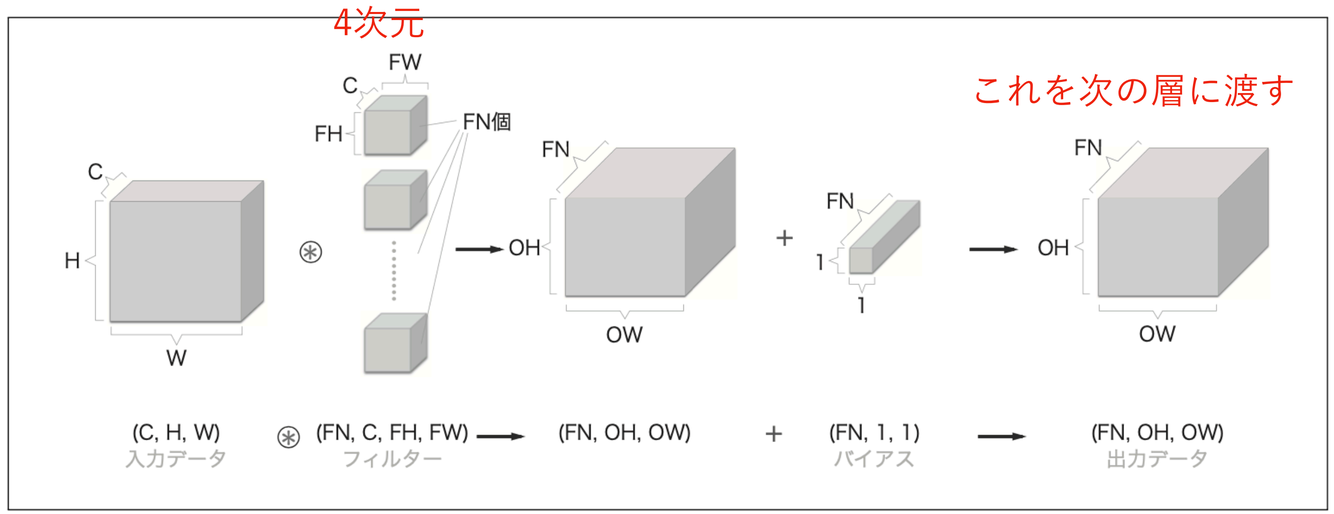

図7-12　畳み込み演算の処理フロー（バイアス項も追加）

バッチ処理

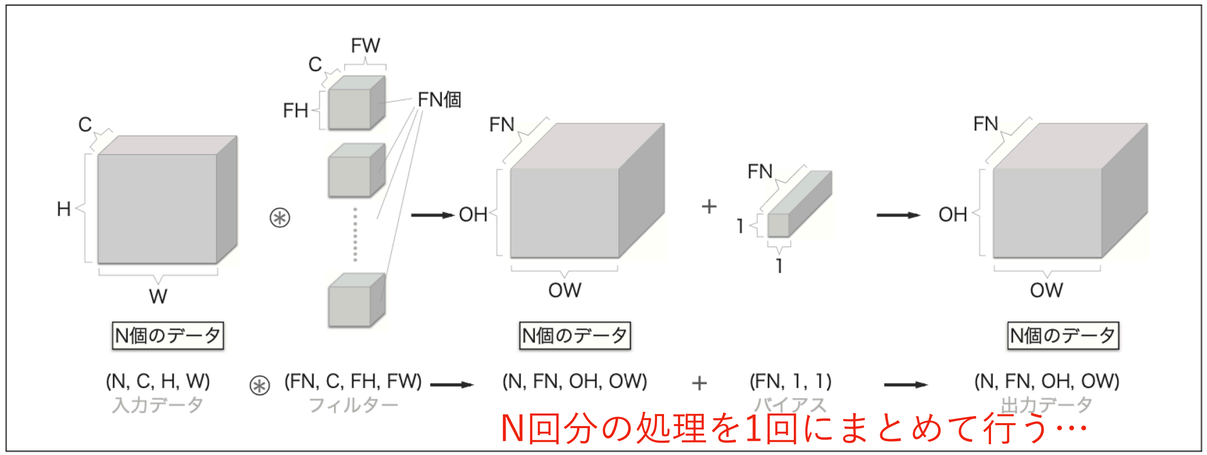

図7-13　畳み込み演算の処理フロー（バッチ処理）

- バッチ処理もやるよ
- (batch_num , channel, height, width)の4次元でデータを流す

Pooling 層

2 ✖ 2の Max pooling (stride 2 の場合)

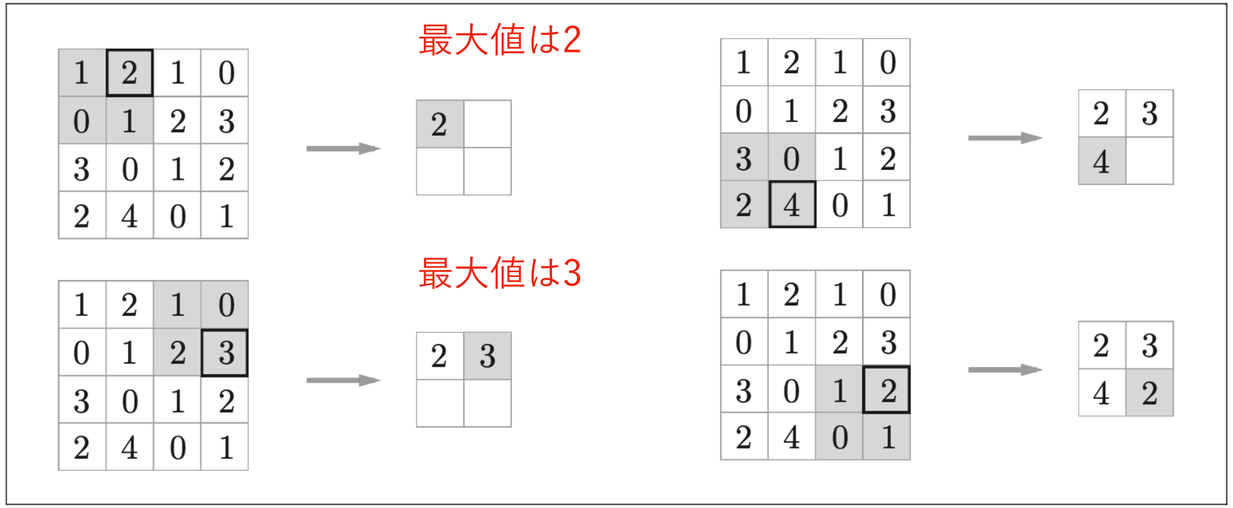

図7-14　Max プーリングの処理手順

- 縦横方向の空間を小さくする演算
  一般的にpoolingのwindowサイズとstrideは同じ値に

Maxプーリング　：最大値をとる

Pooling 層：特徴

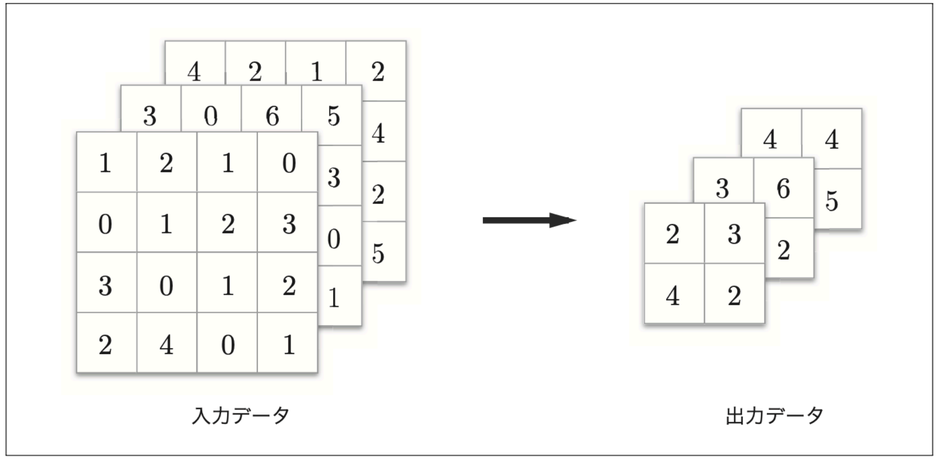

図7-15　プーリングではチャンネル数は変わらない

- 学習するパラメータはない
- チャンネル数は変化しない
- 微小な位置変化に対してロバスト
  （入力データの位置が少しずれても、影響があまりなくなる）

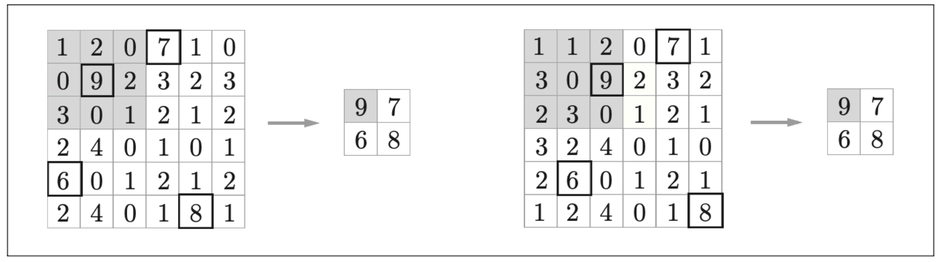

図7-16　入力データが横方向に 1 要素分だけずれた場合でも、出力は同じような結果になる（データによっては同じにならない場合もある）

実装

- 実装部分はまた次回…