

# 畳み込み実装2

shun sato

# 畳み込みニューラルネットワーク(CNN)での活用

畳み込みカーネルの値をトレーニングで学習させる

- チャンネル
  - RGBなどの**複数チャンネル**に対応
- パディング
  - 画像周りの**0埋め**して画像サイズの維持
- スライド
  - カーネルを**ステップ**させる

入力データ (5x5x3)

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

カーネル (3x3)

1	2	3
4	5	6
7	8	9

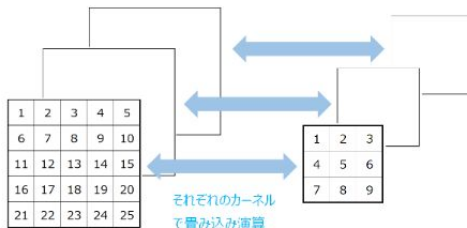
\*

この場合だと3x3x3なので27個のパラメータになります。

そして、チャンネルごとに畳み込み演算を行います。

入力データ (5x5x3)

カーネル (3x3)



		20	20	20
	10	10	10	20
411	456	501	10	20
636	681	726	10	
861	906	951	10	

# Strideの考え方

Strides = 2

5	4	3	2	1
5	5	3	2	1
5	4	5	2	1
5	4	3	5	1
5	4	5	5	1

Input Image(x)  
(n x n)  
5 x 5

\*

1	1	1
0	0	0
1	1	1

Filter  
(f x f)  
3 x 3

=

26	14
28	19

Output Image (y1)  
 $(\frac{n-f}{s} + 1) \times (\frac{n-f}{s} + 1)$   
 $(\frac{5-3}{2} + 1) \times (\frac{5-3}{2} + 1)$   
2 x 2

出力画像のサイズが小さくなる

# 実装してみよう

追加の機能を畳み込みカーネルに実装する

- 複数チャンネル
  - RGBをそれぞれ処理できるように
- パディング処理
  - 出力画像が入力画像と同じサイズになるように
- スライド
  - できればやる(少し難しいです)
  - x,y方向で別々のスライドを指定できるように

# 出力画像のサイズ計算

- ストライドとパディングがある時の出力サイズの計算式

$$\text{output size} = \left\lfloor \frac{(\text{input size}) + 2 * \text{padding} - (\text{kernel size} - 1) - 1}{\text{stride}} + 1 \right\rfloor$$

<https://makeyourownneuralnetwork.blogspot.com/2020/02/calculating-output-size-of-convolutions.html>