畳み込み演算

Convolution

Conv2D

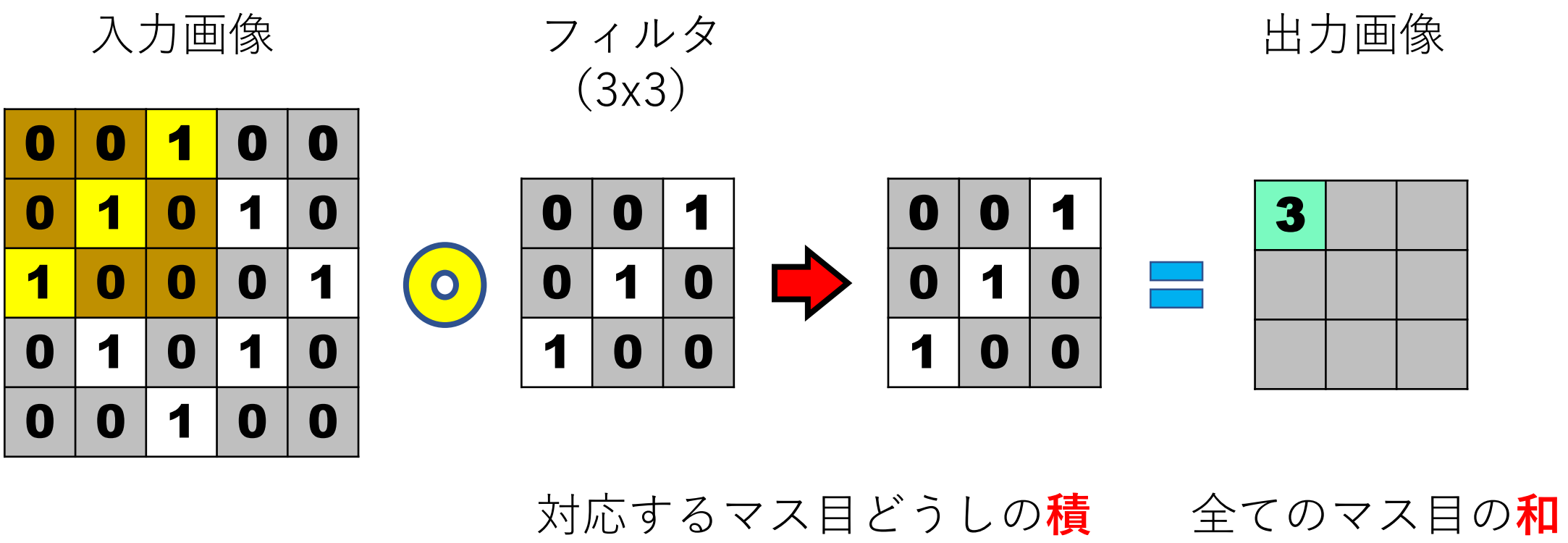
入力画像

0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
0	1	0	1	0
0	0	1	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0

Conv2D


入力画像
フィルタ
(3x3)
出力画像
0 0 1 0 0
0 1 0 1 0
1 0 0 0 1
0 1 0 1 0
0 0 1 0 0
0 0 1
0 1 0
1 0 0
0 0 1
0 1 0
1 0 0
3
対応するマス目どうしの積
全てのマス目の和

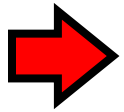
Conv2D

入力画像

0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
0	1	0	1	0
0	0	1	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	0	0
0	0	0

出力画像

3	0	

対応するマス目どうしの**積**

全てのマス目の**和**

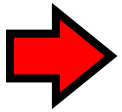
Conv2D

入力画像

0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
0	1	0	1	0
0	0	1	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	1	0
0	0	0

出力画像

3	0	1

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D

入力画像

0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
0	1	0	1	0
0	0	1	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0

0	0	0
0	0	0
0	0	0

出力画像

3	0	1
0		

対応するマス目どうしの**積**

全てのマス目の**和**

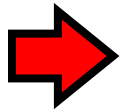
Conv2D

入力画像

0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
0	1	0	1	0
0	0	1	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	1
0	0	0
1	0	0

出力画像

3	0	1
0	2	

対応するマス目どうしの**積**

全てのマス目の**和**

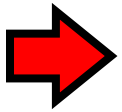
Conv2D

入力画像

0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
0	1	0	1	0
0	0	1	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	0	0
0	0	0

出力画像

3	0	1
0	2	0

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D

入力画像

0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
0	1	0	1	0
0	0	1	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0

0	0	0
0	1	0
0	0	0

出力画像

3	0	1
0	2	0
1		

対応するマス目どうしの**積**

全てのマス目の**和**

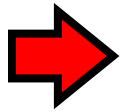
Conv2D

入力画像

0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
0	1	0	1	0
0	0	1	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	0	0
0	0	0

出力画像

3	0	1
0	2	0
1	0	

対応するマス目どうしの**積**

全てのマス目の**和**

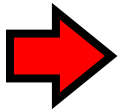
Conv2D

入力画像

0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
0	1	0	1	0
0	0	1	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	1
0	1	0
1	0	0

出力画像

3	0	1
0	2	0
1	0	3

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D

入力画像

0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
0	1	0	1	0
0	0	1	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0

出力画像

3	0	1
0	2	0
1	0	3

Conv2D

入力画像

0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
0	1	0	1	0
0	0	1	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0

出力画像

3	0	1
0	2	0
1	0	3

出力画像が入力画像より**サイズが小さく**なってしまう

出力画像を入力画像と**同じサイズにするオプション**

padding='same'

Conv2D

padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0

Conv2D

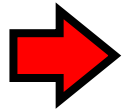
padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	0	0
0	0	0

出力画像

0				

対応するマス目どうしの**積**

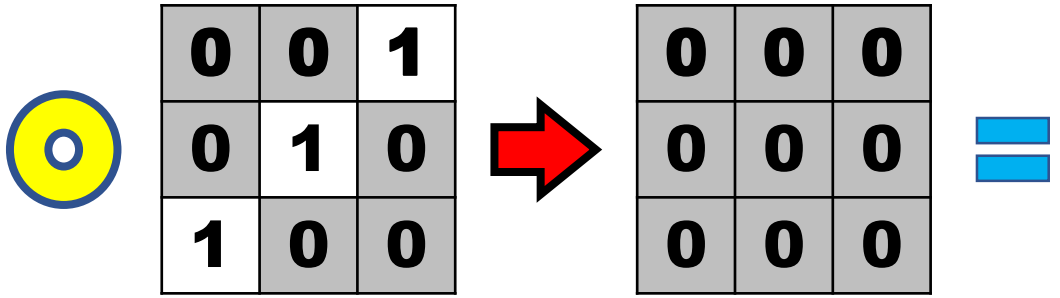
全てのマス目の**和**

Conv2D padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)



出力画像

0	0			

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D

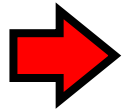
padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	1	0
1	0	0

出力画像

0	0	2		

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D

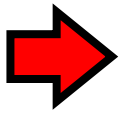
padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	0	0
0	0	0

出力画像

0	0	2	0	

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D

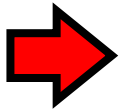
padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	0	0
1	0	0

出力画像

0	0	2	0	1

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D

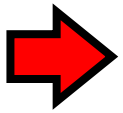
padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	0	0
0	0	0

出力画像

0	0	2	0	1
0				

対応するマス目どうしの**積**

全てのマス目の**和**

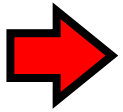
Conv2D　padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	1
0	1	0
1	0	0

出力画像

0	0	2	0	1
0	3			

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D

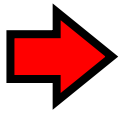
padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	0	0
0	0	0

出力画像

0	0	2	0	1
0	3	0		

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D

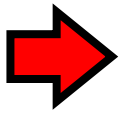
padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	1	0
0	0	0

出力画像

0	0	2	0	1
0	3	0	1	

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D

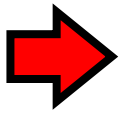
padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	0	0
0	0	0

出力画像

0	0	2	0	1
0	3	0	1	0

対応するマス目どうしの**積**

全てのマス目の**和**

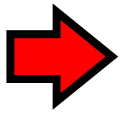
Conv2D padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	1
0	1	0
0	0	0

出力画像

0	0	2	0	1
0	3	0	1	0
2				

対応するマス目どうしの**積**

全てのマス目の**和**

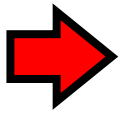
Conv2D padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	0	0
0	0	0

出力画像

0	0	2	0	1
0	3	0	1	0
2	0			

対応するマス目どうしの**積**

全てのマス目の**和**

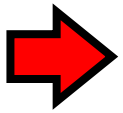
Conv2D padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	1
0	0	0
1	0	0

出力画像

0	0	2	0	1
0	3	0	1	0
2	0	2		

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D

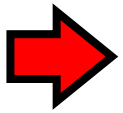
padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	0	0
0	0	0

出力画像

0	0	2	0	1
0	3	0	1	0
2	0	2	0	

対応するマス目どうしの**積**

全てのマス目の**和**

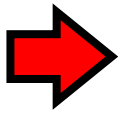
Conv2D padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	1	0
1	0	0

出力画像

0	0	2	0	1
0	3	0	1	0
2	0	2	0	2

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D

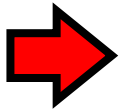
padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	0	0
0	0	0

出力画像

0	0	2	0	1
0	3	0	1	0
2	0	2	0	2
0				

対応するマス目どうしの**積**

全てのマス目の**和**

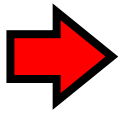
Conv2D padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	1	0
0	0	0

出力画像

0	0	2	0	1
0	3	0	1	0
2	0	2	0	2
0	1			

対応するマス目どうしの**積**

全てのマス目の**和**

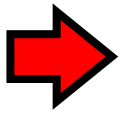
Conv2D padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	0	0
0	0	0

出力画像

0	0	2	0	1
0	3	0	1	0
2	0	2	0	2
0	1	0		

対応するマス目どうしの**積**

全てのマス目の**和**

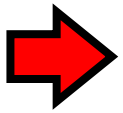
Conv2D padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	1
0	1	0
1	0	0

出力画像

0	0	2	0	1
0	3	0	1	0
2	0	2	0	2
0	1	0	3	

対応するマス目どうしの**積**

全てのマス目の**和**

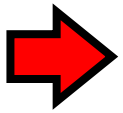
Conv2D padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	0	0
0	0	0

出力画像

0	0	2	0	1
0	3	0	1	0
2	0	2	0	2
0	1	0	3	0

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D

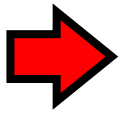
padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	1
0	0	0
0	0	0

出力画像

0	0	2	0	1
0	3	0	1	0
2	0	2	0	2
0	1	0	3	0
1				

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0

0	0	0
0	0	0
0	0	0

出力画像

0	0	2	0	1
0	3	0	1	0
2	0	2	0	2
0	1	0	3	0
1	0			

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0

0	0	1
0	1	0
0	0	0

出力画像

0	0	2	0	1
0	3	0	1	0
2	0	2	0	2
0	1	0	3	0
1	0	2		

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D

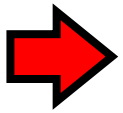
padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0



0	0	0
0	0	0
0	0	0

出力画像

0	0	2	0	1
0	3	0	1	0
2	0	2	0	2
0	1	0	3	0
1	0	2	0	

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D

padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0

0	0	0
0	0	0
0	0	0

出力画像

0	0	2	0	1
0	3	0	1	0
2	0	2	0	2
0	1	0	3	0
1	0	2	0	0

対応するマス目どうしの**積**

全てのマス目の**和**

Conv2D padding='same'

入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

0	0	1
0	1	0
1	0	0

出力画像

0	0	2	0	1
0	3	0	1	0
2	0	2	0	2
0	1	0	3	0
1	0	2	0	0

フィルタの特徴を反映した画素の強度が高くなっている

次は自分で手計算してみよう！

Conv2D

padding='same'

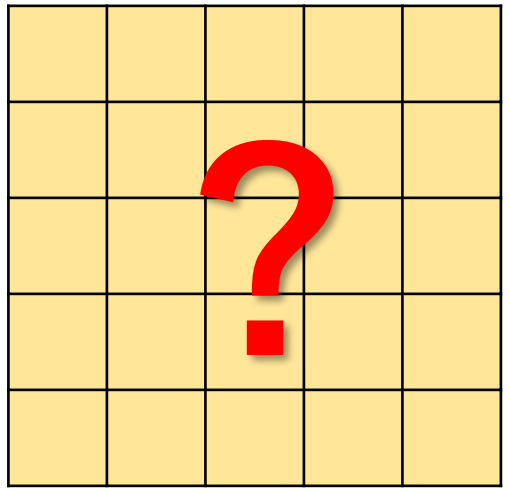
入力画像

0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

フィルタ
(3x3)

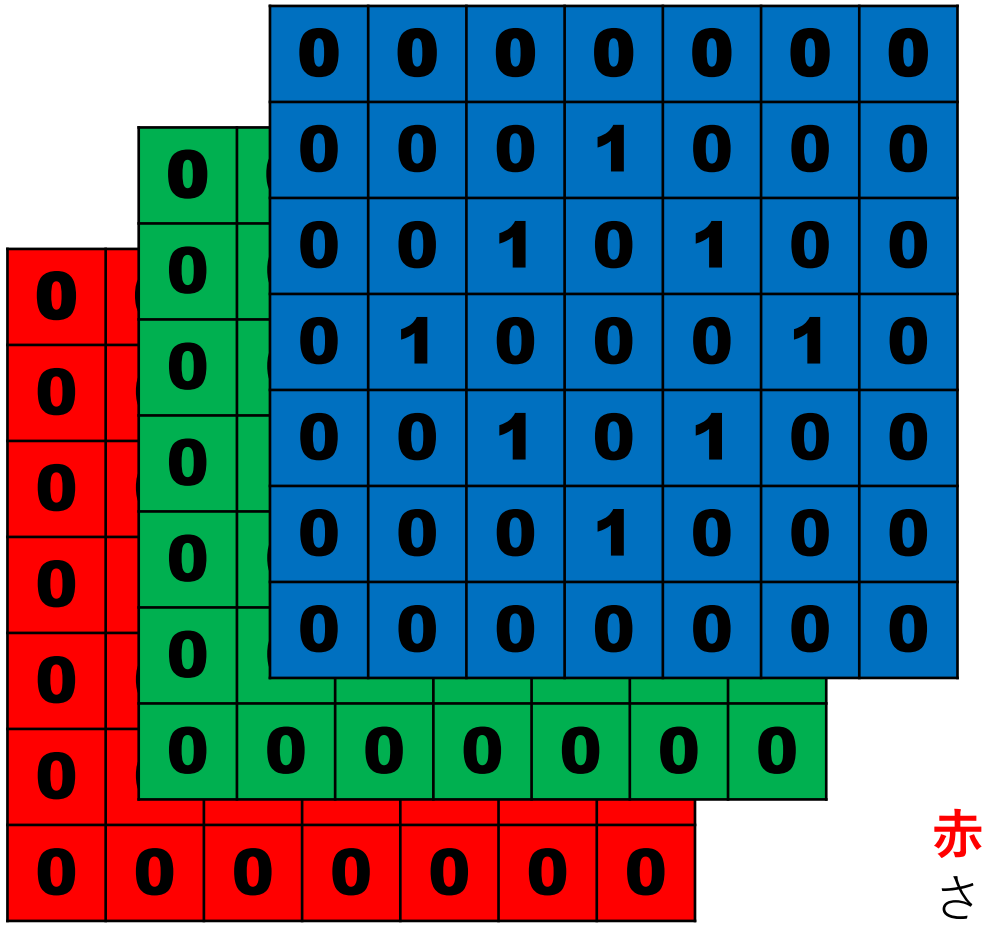
1	0	0
0	1	0
0	0	1

出力画像

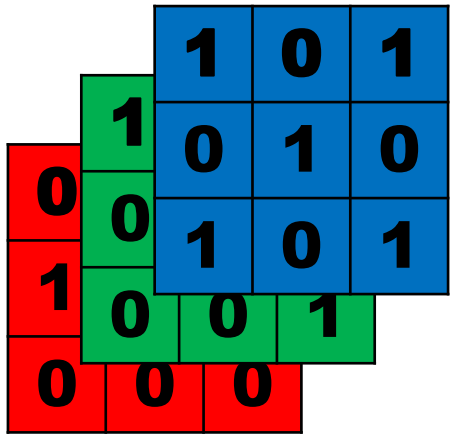


Conv2D　padding='same'

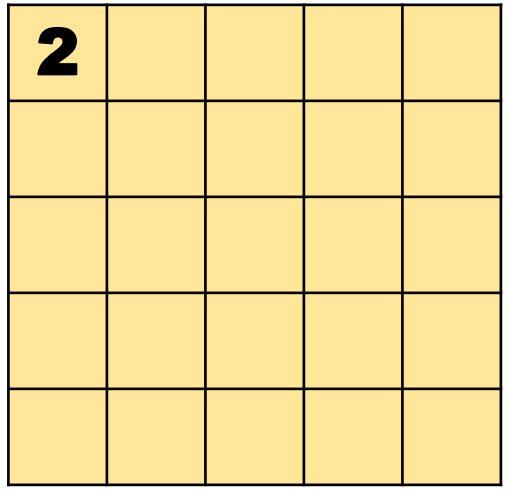
入力画像（ＲＧＢ）



特徴フィルタ **1個分**
(3x3)



出力画像 １ＣＨ



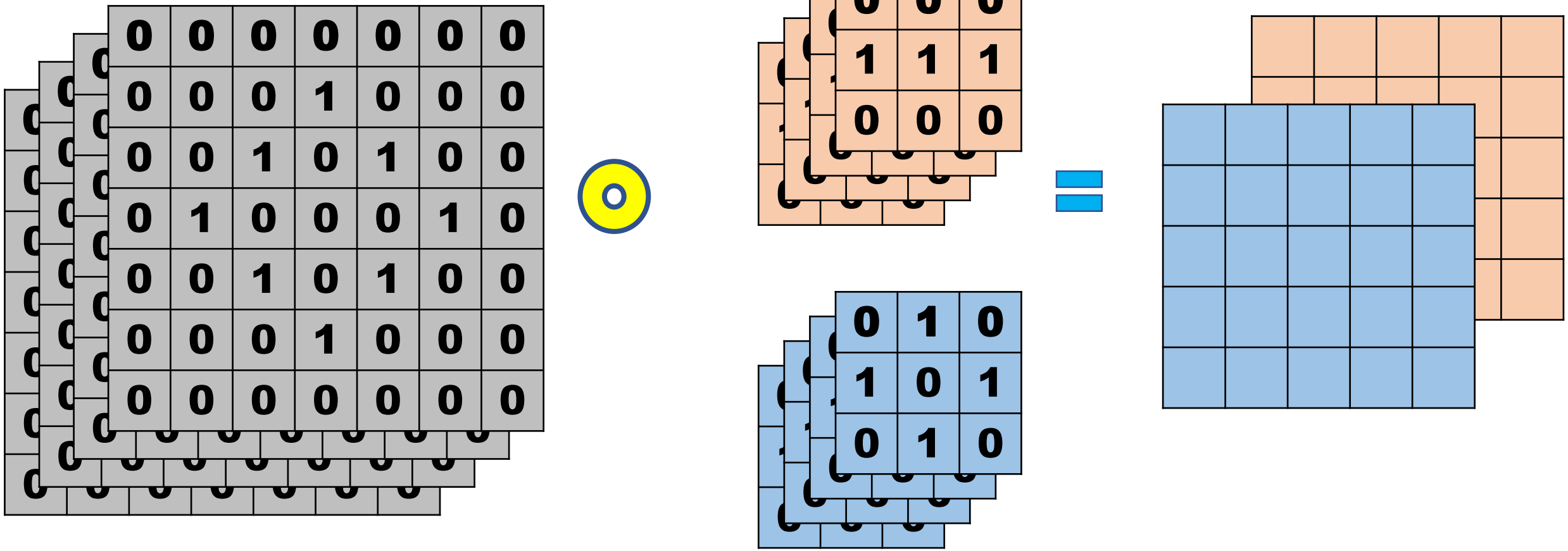
赤・**緑**・**青**のそれぞれについて**積和計算**をして、
さらに**３つの和**を計算した値を出力画像のマス目に記入する。
例えば、**特徴フィルタ**が**4個**ある場合、**出力画像**は **4CH** となる。

Conv2D

入力画像 4 CH

特徴フィルタ 2 個
（3x3）

出力画像 2 枚



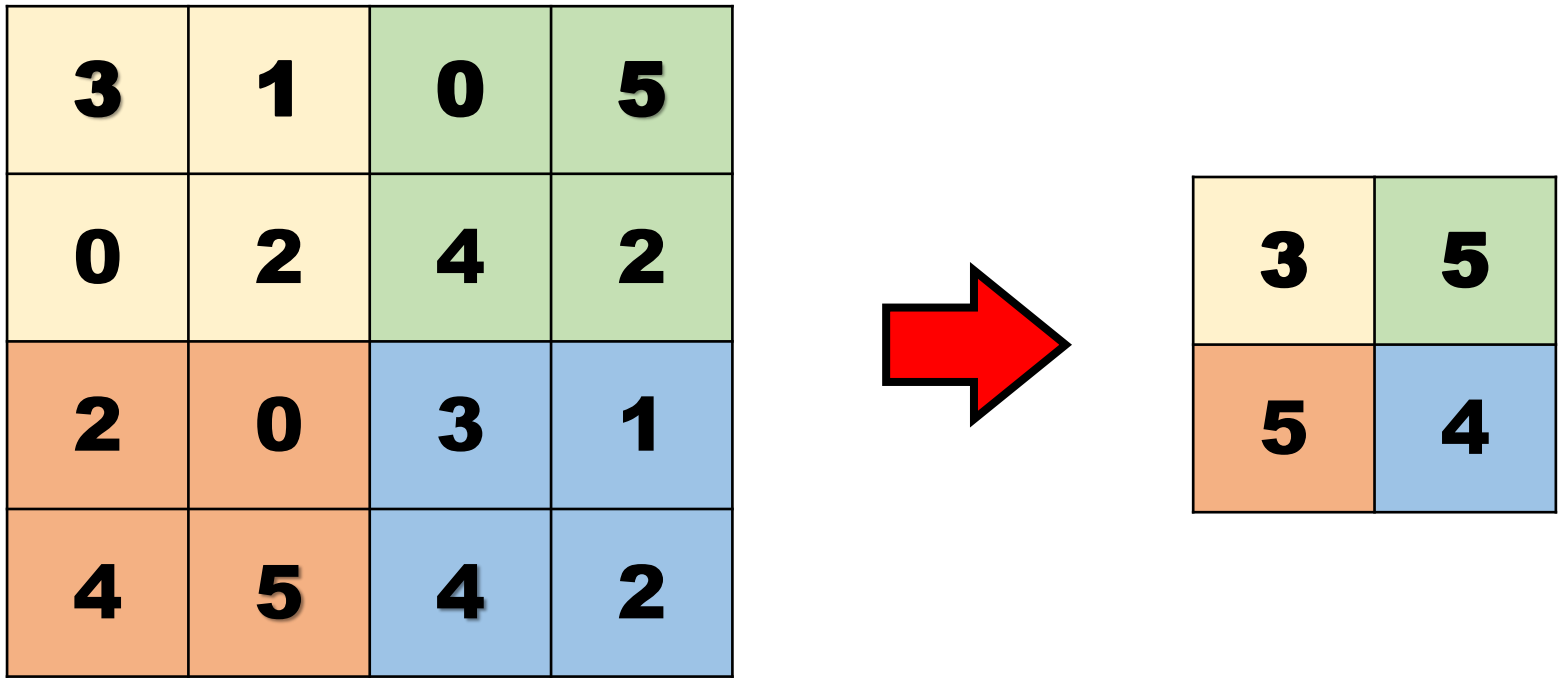
プーリング演算

MaxPooling

ダウンサイジング効果

MaxPooling2D

pool_size=(2, 2)

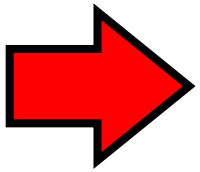


1次元化

Flatten

Flatten

2	1	4
5	3	2
0	2	3



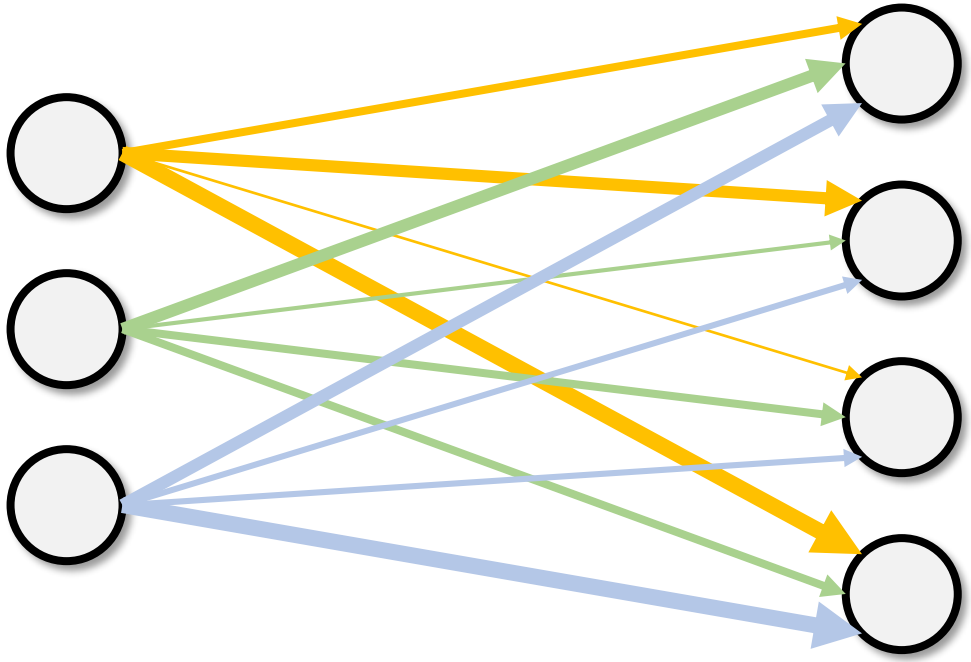
1次元化

2
1
4
5
3
2
0
2
3

全結合

Dense

Dense



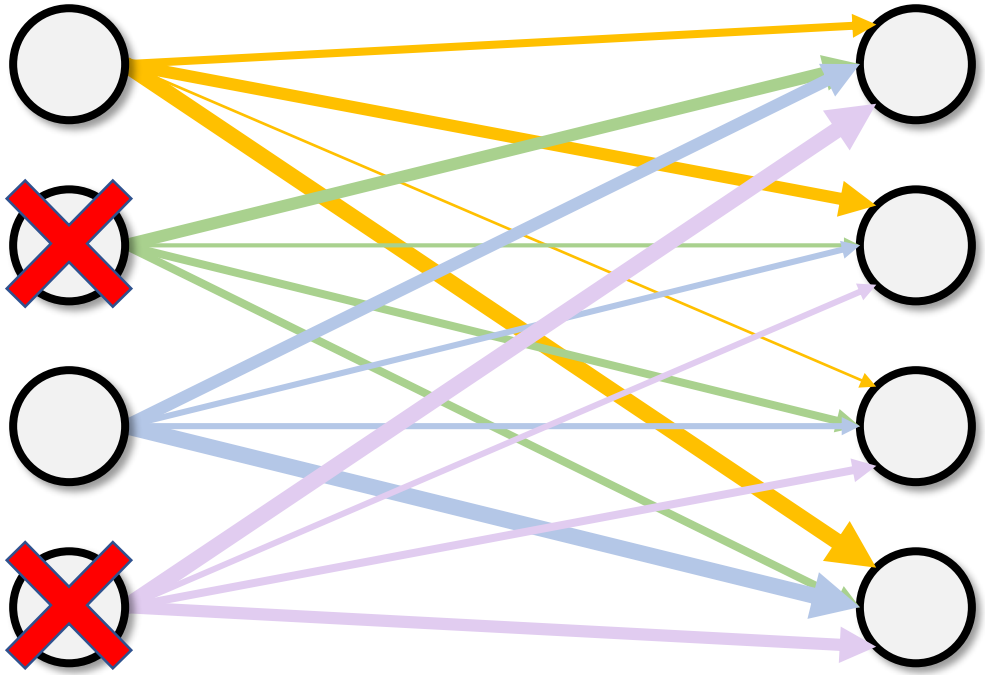
入力層と出力層の素子どうしを**全て結合**する

ドロップアウト

Dropout

Dropout

rate=0.5



入力層の素子の出力を**ランダムに無効化**する