

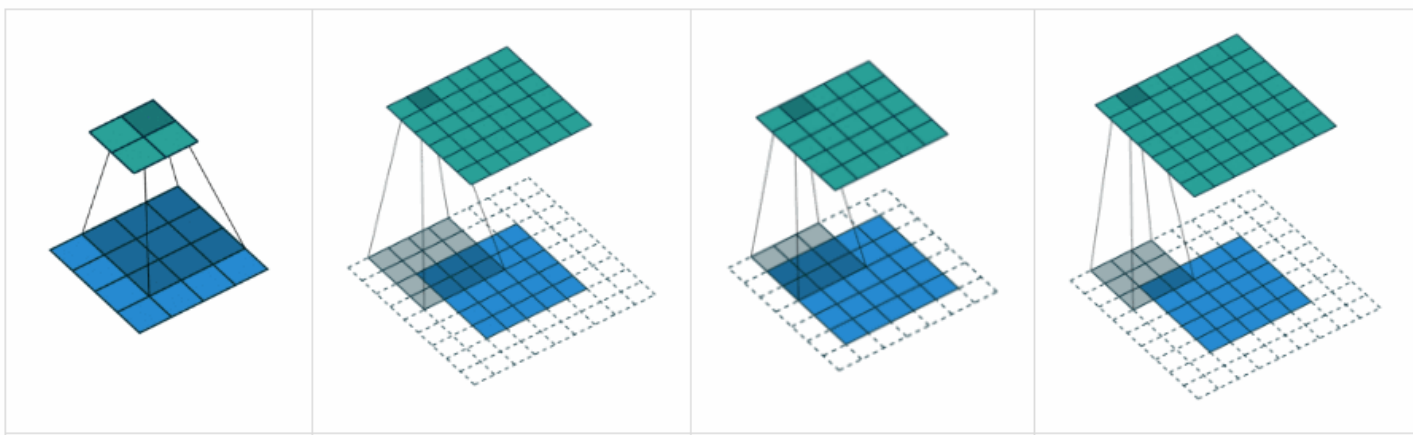
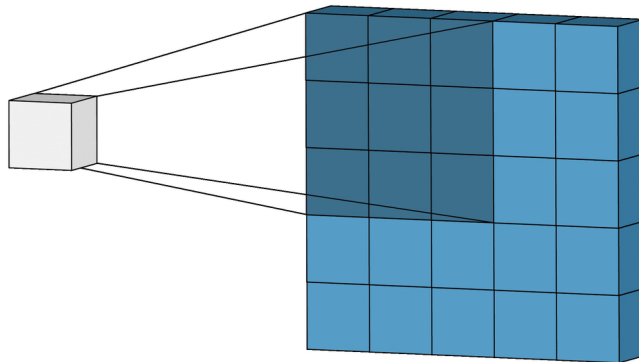
二维图像 (scipy.signal.convolve2d)

mode: x: signal, $n \Rightarrow n \times n$; h: kernel, $m \Rightarrow m \times m$;

- full: $n+m-1, \Rightarrow (n+m-1) \times (n+m-1)$
- same: $n \ (n \geq m) \Rightarrow n \times n$
- valid: $(n-m)+1 \Rightarrow ((n-m)+1) \times ((n-m)+1)$

1. `conv2d(5*5, 3*3, mode='valid')`

- shape: $(5-3)+1 = 3$
 - 3×3



- `1, conv2d(4*4, 3*3, mode='valid')`
 - $(4-3)+1 = 2 \times 2$
- `2, conv2d(9*9, 4*4, mode='valid')`
 - $(9-4)+1 = 6 \times 6$
- `3, conv2d(5*5, 3*3, mode='same')`
 - $\max(5, 3) == 5 \times 5$

- `conv2d(7*7, 3*3, mode='valid')`
 - $7-3+1=5$
- `4, conv2d(5*5, 3*3, mode='full')`
 - $(5+3)-1 == 7*7$

<https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480>

2. `scipy.signal.convolve2d`

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.convolve2d.html>

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

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

```

1 x = [[3, 3, 2, 1, 0],
2       [0, 0, 1, 3, 1],
3       [3, 1, 2, 2, 3],
4       [2, 0, 0, 2, 2],
5       [2, 0, 0, 0, 1]]
6
7 h = [[0, 1, 2],
8       [2, 2, 0],
9       [0, 1, 2]]
10
11 from scipy import signal
12
13 >> signal.convolve2d(x, h, mode='valid')
14
15 array([[18, 20, 19],
16        [10, 9, 17],
17        [11, 8, 14]])
18
19 # 想要得到右侧的结果, 需要对 h 进行一次 reverse
20 h = [[2, 1, 0],
21       [0, 2, 2],
22       [2, 1, 0]]
23
24 >> signal.convolve2d(x, h, mode='valid')
25 array([[12, 12, 17],
26        [10, 17, 19],

```

```
27         [ 9,  6, 14]])  
28
```

<https://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215>