一维离散（np.convolve）

> 满足交换律；conv(x, h) == conv(h, x)

1. `np.convolve`

- https://numpy.org/doc/stable/reference/generated/numpy.convolve.html
- `convolve(x, h, mode='full')`
  - x：signal，h： kernel/filter;
  - **Reversed** h，sweeping over x
- 三种 mode，return length
  - valid: `max(M, N) - min(M, N) + 1`，length 较小的全部参与 计算
    - M == N 时，return 的长度为 1；
  - same: `max(M, N)`
  - full: `(N+M-1,)` default;

2. examples

```
h = [4, 3, 2]    # reversed h: [2, 3, 4]
x = [1, 1, 5, 5]
```

- `full`

```
>> np.convolve(x, h, mode='full')
array([ 4,  7, 25, 37, 25, 10])

# full: (4+3)-1 = 6
2, 3, 4
      1, 1, 5, 5    =  4    (boundary effects)
```

```
 7
 8 2, 3, 4
 9    1, 1, 5, 5       = 7     (boundary effects)
10
11    2, 3, 4
12    1, 1, 5, 5       = 25
13
14       2, 3, 4
15    1, 1, 5, 5       = 37
16
17          2, 3, 4
18    1, 1, 5, 5       = 25  (boundary effects)
19
20             2, 3, 4
21    1, 1, 5, 5       = 10  (boundary effects)
```

- `valid`

```
 1 >> np.convolve(x, h, mode='valid')
 2 array([25, 37])
 3
 4 # valid: (4-3)+1
 5 2, 3, 4
 6       1, 1, 5, 5    =  4    (boundary effects)
 7
 8 2, 3, 4
 9    1, 1, 5, 5       = 7     (boundary effects)
10
11    2, 3, 4
12    1, 1, 5, 5       = 25
13
14       2, 3, 4
15    1, 1, 5, 5       = 37
16
17          2, 3, 4
18    1, 1, 5, 5       = 25  (boundary effects)
19
20             2, 3, 4
21    1, 1, 5, 5       = 10  (boundary effects)
```

- `same`

```
1 >> np.convolve(x, h, mode='same')
```

```
array([ 7, 25, 37, 25])

# same: same as the length of signal
# same:
    # same left
    # same center
    # same right

2, 3, 4
      1, 1, 5, 5    =  4    (boundary effects)

2, 3, 4
   1, 1, 5, 5       = 7     (boundary effects)

   2, 3, 4
   1, 1, 5, 5       = 25

      2, 3, 4
   1, 1, 5, 5       = 37

         2, 3, 4
   1, 1, 5, 5       = 25  (boundary effects)

            2, 3, 4
   1, 1, 5, 5       = 10  (boundary effects)
```