



华中科技大学

图像卷积运算的理解

许向阳

xuxy@hust.edu.cn





华中科技大学

参考书:

姚天任，数字信号处理，华中科技大学出版社



数字图像中的卷积

卷积

$$\begin{aligned} g(x, y) &= f(x, y) \otimes h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n) \\ &= f(x, y) * h(0, 0) + f(x + 1, y + 1) * h(-1, -1) + f(x - 1, y - 1) * h(1, 1) + \dots \end{aligned}$$

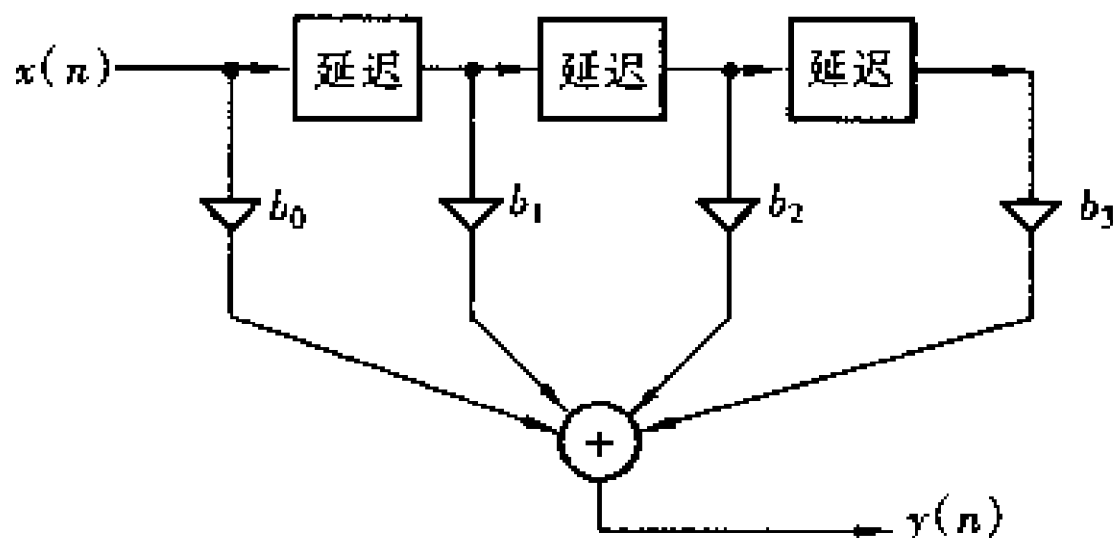
滤波

$$\begin{aligned} g(x, y) &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) * h(m - x, n - y) \\ &= f(x, y) * h(0, 0) + f(x + 1, y + 1) * h(1, 1) + f(x - 1, y - 1) * h(-1, -1) + \dots \end{aligned}$$

FIR数字滤波器

有限冲激响应 (Finite Impulse Response)系统

$$y(n] = b_0 x(n] + b_1 x(n-1] + b_2 x(n-2] + b_3 x(n-3]$$

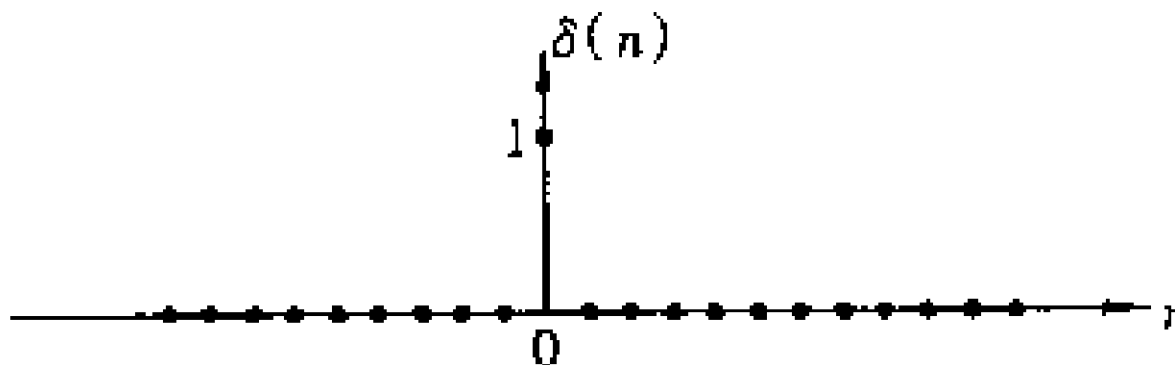


三阶FIR系统的实现

离散时间信号的基本形式

单位取样序列(离散冲激)

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$



离散时间信号的基本形式

单位阶跃序列

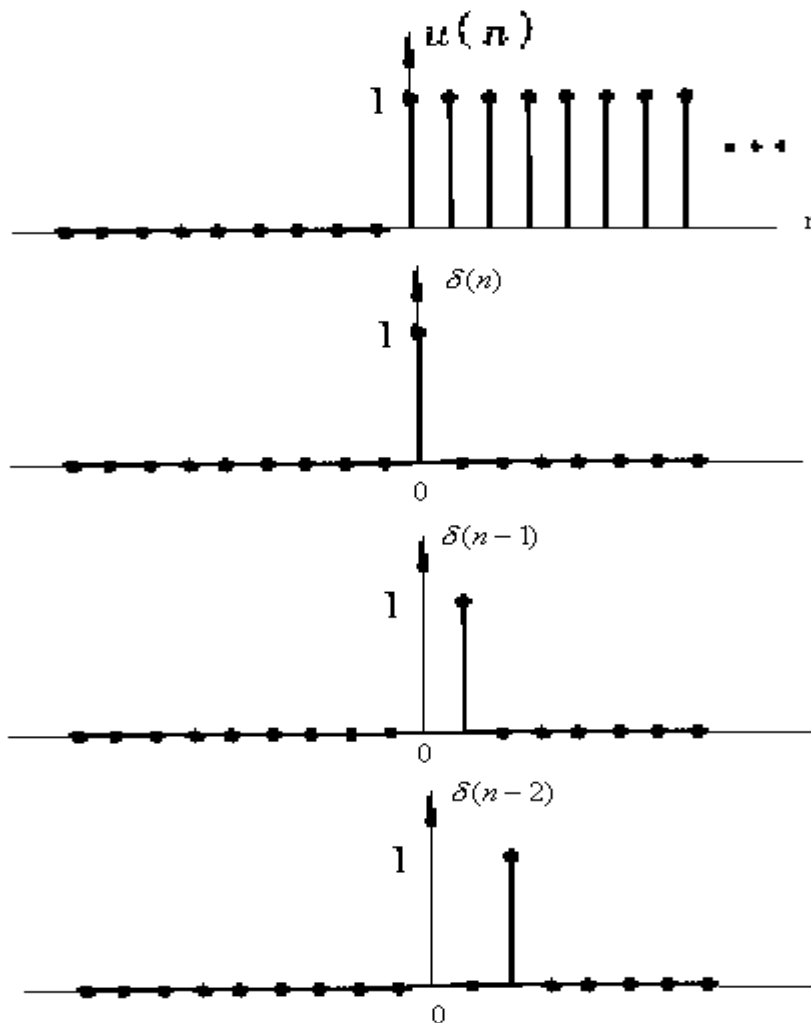
$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$



离散时间信号的基本形式

单位阶跃序列

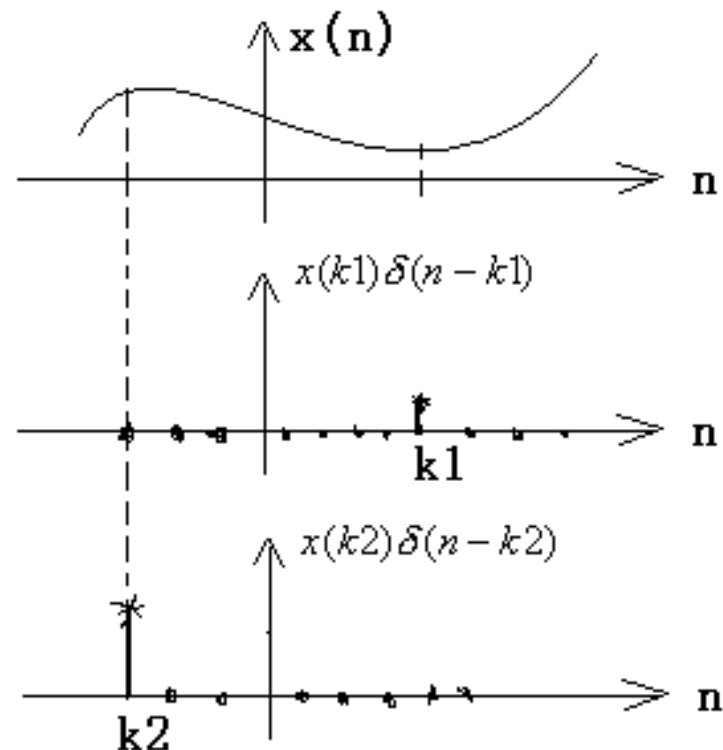
$$u(n) = \sum_{k=0}^{\infty} \delta(n-k)$$



离散时间信号的基本形式

任意的输入信号

$$x(n] = \sum_{k=-\infty}^{+\infty} x(k) \delta(n - k)$$



离散时间系统

离散时间系统，就是将输入序列变成输出序列的系统



满足叠加原理的系统称为**线性系统**。

若系统的响应与输入信号施加于系统的时刻无关，称该系统为**非移变系统（非时变）**。

$$T[x(n-k)] = y(n-k)$$

线性非移变系统

输入 $x(n)$ ，输出 $y(n)$

当输入为 $\delta(n)$

$$y(n) = T[\delta(n)] = h(n)$$

输出 $h(n)$ ，称为单位冲激（取样）响应

线性非移变系统

$$\begin{aligned} y(n) &= T[x(n)] = T\left[\sum_{k=-\infty}^{+\infty} x(k)\delta(n-k)\right] \\ &= \sum_{k=-\infty}^{+\infty} T[x(k)\delta(n-k)] \\ &= \sum_{k=-\infty}^{+\infty} x(k)T[\delta(n-k)] \\ &= \sum_{k=-\infty}^{+\infty} x(k)h(n-k) \\ &= x(n) \otimes h(n) \end{aligned}$$

信号的叠加

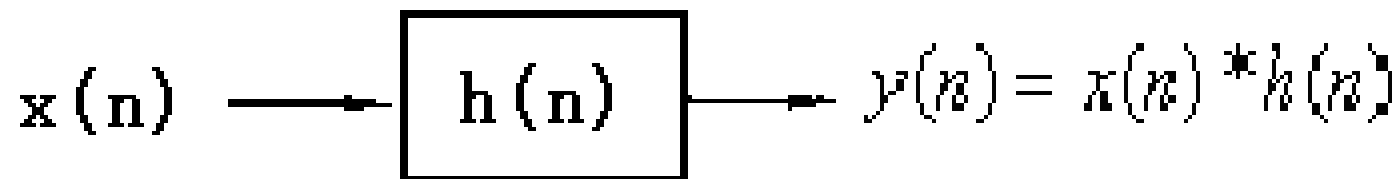
线性系统

线性系统

非移变

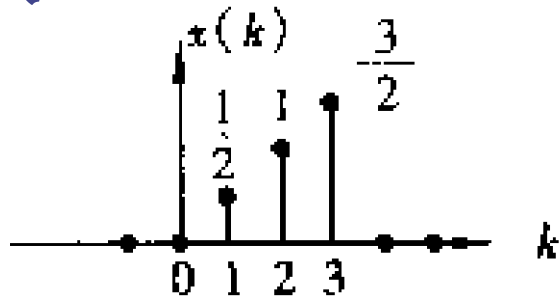
卷积

线性非移变系统

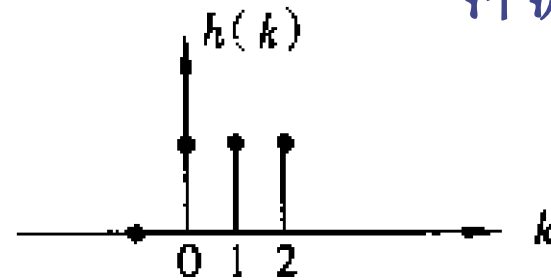


线性非移变系统

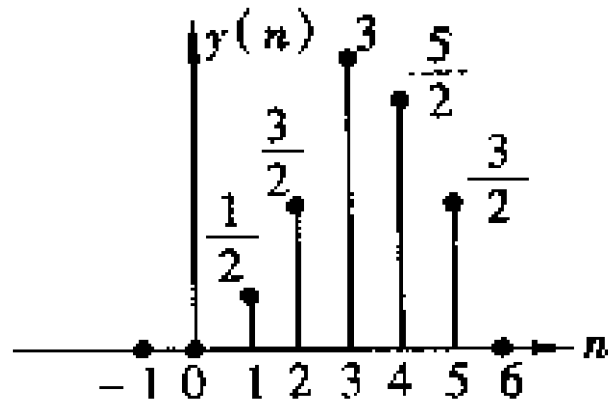
输入



冲激响应



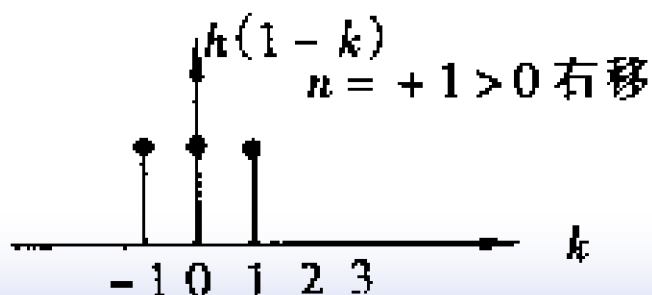
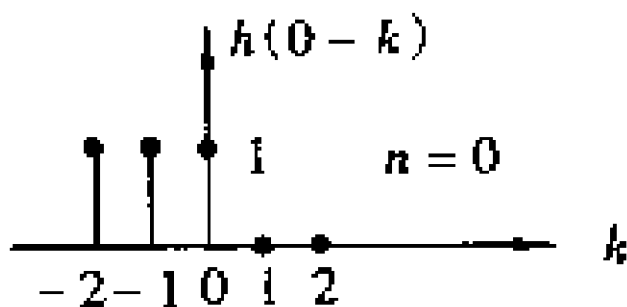
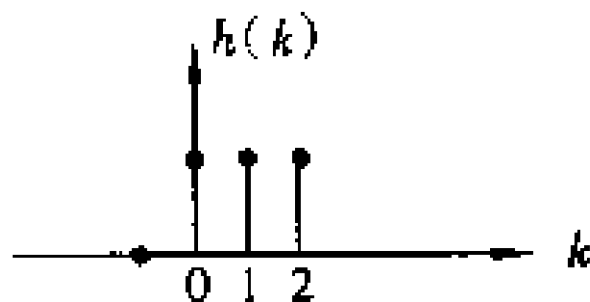
输出结果



线性非移变系统

线性卷积的计算方法

- 折叠
- 移位
- 相乘
- 相加





线性非移变系统

线性卷积的计算方法

➤ 折叠

$p(k) = h(-k)$ 关于原点反转

➤ 移位

$q(k) = p(k - n)$ 向右平移 n
 $= h(n - k)$

➤ 相乘

➤ 相加

$$y(n) = \sum_{k=-\infty}^{+\infty} x(k)h(n-k)$$





卷积的性质

a =

1 2 3 4 5 6

>> b

b =

1 2 3

>> conv(a,b)

ans =

1 4 10 16 22 28 27 18

>> conv(b,a)

ans =

1 4 10 16 22 28 27 18

$$\text{conv}(a,b) = \text{conv}(b,a)$$





卷积的性质

$$\text{conv}(\text{conv}(a, b), c) = \text{conv}(a, \text{conv}(b, c))$$

```
c =
```

```
1 3
```

```
>> conv(conv(a, b), c)
```

```
ans =
```

```
1 7 22 46 70 94 111 99 54
```

```
>> conv(a, conv(b, c))
```

```
ans =
```

```
1 7 22 46 70 94 111 99 54
```

卷积性质的应用

$$\text{conv}(\text{conv}(a, b), c) = \text{conv}(a, \text{conv}(b, c))$$

用一个二维高斯模板对图像进行平滑，等价于 先用一维的高斯逐行（或逐列）对图像进行平滑，然后再用一维的高斯逐列（或逐行）对新图像进行平滑。

$$g1(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

$$g2(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} = g1(x) * g1(y)$$

$$\text{conv2}(I, g_2(x, y)) = \text{conv2}(\text{conv2}(I, g_1(x)), g_1^T(x))$$

$$g2(x, y) = \text{conv2}(g_1(x), g_1'(x));$$



卷积性质的应用

```
g1 = fspecial('gaussian',[1 7]);
```

	1	2	3	4	5	6	7
1	1.1979e-08	2.6387e-04	0.1065	0.7866	0.1065	2.6387e-04	1.1979e-08

```
g2 = fspecial('gaussian',[7 7]);
```

	1	2	3	4	5	6	7
1	1.4351e-16	3.1610e-12	1.2752e-09	9.4227e-09	1.2752e-09	3.1610e-12	1.4351e-16
2	3.1610e-12	6.9625e-08	2.8089e-05	2.0755e-04	2.8089e-05	6.9625e-08	3.1610e-12
3	1.2752e-09	2.8089e-05	0.0113	0.0837	0.0113	2.8089e-05	1.2752e-09
4	9.4227e-09	2.0755e-04	0.0837	0.6187	0.0837	2.0755e-04	9.4227e-09
5	1.2752e-09	2.8089e-05	0.0113	0.0837	0.0113	2.8089e-05	1.2752e-09
6	3.1610e-12	6.9625e-08	2.8089e-05	2.0755e-04	2.8089e-05	6.9625e-08	3.1610e-12
7	1.4351e-16	3.1610e-12	1.2752e-09	9.4227e-09	1.2752e-09	3.1610e-12	1.4351e-16

卷积性质的应用

$$g2 = g1' * g1;$$

```
>> g2
```

```
g2 =
```

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0002	0.0000	0.0000	0.0000
0.0000	0.0000	0.0113	0.0837	0.0113	0.0000	0.0000
0.0000	0.0002	0.0837	0.6187	0.0837	0.0002	0.0000
0.0000	0.0000	0.0113	0.0837	0.0113	0.0000	0.0000
0.0000	0.0000	0.0000	0.0002	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
>> g1'*g1
```

```
ans =
```

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0002	0.0000	0.0000	0.0000
0.0000	0.0000	0.0113	0.0837	0.0113	0.0000	0.0000
0.0000	0.0002	0.0837	0.6187	0.0837	0.0002	0.0000
0.0000	0.0000	0.0113	0.0837	0.0113	0.0000	0.0000
0.0000	0.0000	0.0000	0.0002	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000



卷积性质的应用

$$g3 = g1' * g1;$$

```
>> g1'*g1
```

```
ans =
```

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0002	0.0000	0.0000	0.0000
0.0000	0.0000	0.0113	0.0837	0.0113	0.0000	0.0000
0.0000	0.0002	0.0837	0.6187	0.0837	0.0002	0.0000
0.0000	0.0000	0.0113	0.0837	0.0113	0.0000	0.0000
0.0000	0.0000	0.0000	0.0002	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

$$g2 = \text{conv2}(g1, g1');$$

```
>> conv2(g1,g1')
```

```
ans =
```

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0002	0.0000	0.0000	0.0000
0.0000	0.0000	0.0113	0.0837	0.0113	0.0000	0.0000
0.0000	0.0002	0.0837	0.6187	0.0837	0.0002	0.0000
0.0000	0.0000	0.0113	0.0837	0.0113	0.0000	0.0000
0.0000	0.0000	0.0000	0.0002	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000





卷积性质的应用

Q: 用二次一维高斯平滑来代替一次的二维高斯平滑，有何优点？

➤ **计算量变小。**

假设模板是 $7 * 7$ ；对结果图像中的每一个点 有**49**次运算；

先用 $1 * 7$ 的模板，对中间图像的每一个点 有运算 7 次运算；再用 $7 * 1$ 的模板平滑，每个点又有 7 次运算，共计 **14** 次运算。

➤ **参数量变少**

在卷积神经网络中，一个 $7 * 7$ 的核是 **49** 个参数，而 $1 * 7$ 的核只有 **7** 个参数。





卷积性质的应用

Q: 对一个图像先用二维高斯平滑，然后在 x 方向上求一阶导数，如何优化实现？

$$\begin{aligned} & \text{conv2}(\text{conv2}(I, g_2(x,y)), [1 \ 0 \ -1]) \\ &= \text{conv2}(I, \text{conv2}(g_2(x,y), [1 \ 0 \ -1])) \\ &= \text{conv2}(I, \partial g_2(x,y) / \partial x) \end{aligned}$$

$$g_2(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} = g_1(x) * g_1(y)$$

$$\frac{\partial}{\partial x} g_2(x, y) = \frac{\partial g_1(x)}{\partial x} * g_1(y)$$

先对图像用一维高斯在列方向平滑，然后用一维高斯的导数模板在行方向上平滑。





卷积的补充说明

$C = \text{conv}(A, B)$ convolves vectors A and B.

$C = \text{conv}(A, B, \text{SHAPE})$ returns a subsection of the convolution with size specified by SHAPE:

SHAPE : 'full'、 'same' 'valid'

$C = \text{conv2}(A, B)$ performs the 2-D convolution of matrices A and B

convn N-dimensional convolution.





卷积运算的实现方法

$$k = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 4 \\ 3 & 4 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 0 \\ 0 & 5 & 6 & 7 & 8 & 0 \\ 0 & 9 & 10 & 11 & 12 & 0 \\ 0 & 13 & 14 & 15 & 16 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

傅里叶变换 与反变换

```
conv2(A, k) => conv2(A, k, 'same')  
FR= fft2(A).* psf2otf(k, [6 6]);  
ifft2(FR) = conv2(A, k, 'same')
```

注意： **A** 矩阵的外围都是 **0**，
傅里叶变换要求周期性，通过扩展 **0**，
解决了缠绕问题。





卷积运算的实现方法

用矩阵乘法
实现卷积

$$k = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 4 \\ 3 & 4 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

1	2	3	0	2	1	4	0	3	4	1	0	0	0	0	0
0	1	2	3	0	2	1	4	0	3	4	1	0	0	0	0
0	0	0	0	1	2	3	0	2	1	4	0	3	4	1	0
0	0	0	0	0	1	2	3	0	2	1	4	0	3	4	1

K 重排后的矩阵
卷积矩阵

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	

*

=





反卷积

matlab 中的反卷积

deconv

```
C = conv(a, b);  
a = deconv(c,b);
```

```
>> a  
  
a =  
  
    1    2    3    4    5    6  
  
>> b  
  
b =  
  
    1    2    3  
  
>> c=conv(a,b)  
  
c =  
  
    1    4   10   16   22   28   27   18  
  
>> p=deconv(c,b)  
  
p =  
  
    1    2    3    4    5    6
```



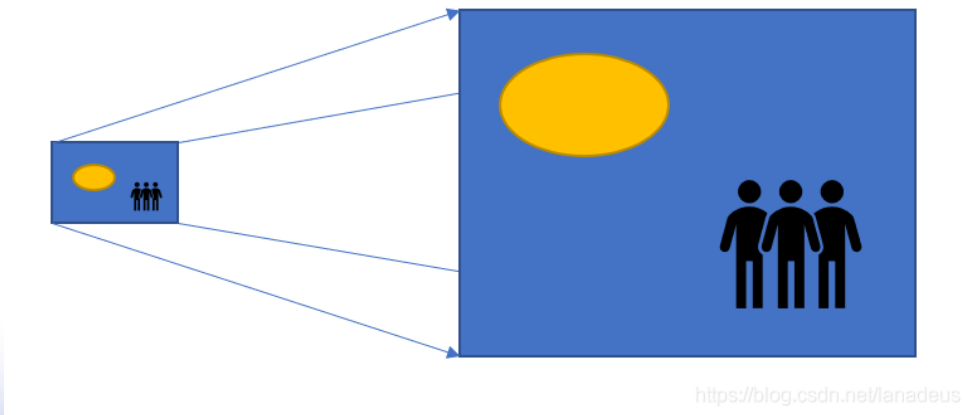
反卷积

假设卷积核 为 3×3 ,

卷积操作：将输入矩阵的 **9** 个值 通过卷积核映射为 **1** 个值

反卷积：将某矩阵的 **1** 个值 映射为另外一个矩阵的 **9** 个值。

上采样(up-sampling)：提高低分辨率图像的分辨率



反卷积

$$A1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$A2 = \begin{bmatrix} 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 3 & 3 & 4 & 4 \\ 3 & 3 & 4 & 4 \end{bmatrix}$$

A1 放大 2倍，得到A2

用向量表示矩阵

1	1	0	0	0	0
1	1	0	0	0	0
2	0	2	2	0	0
2	0	2	2	0	0
1	1	0	0	0	0
1	1	0	0	0	0
2	0	2	2	0	0
2	0	2	2	0	0
3	0	0	0	3	0
3	0	0	0	3	0
4	0	0	0	0	4
4	0	0	0	0	4
3	0	0	0	3	0
3	0	0	0	3	0
4	0	0	0	0	4
4	0	0	0	0	4

=

1	1	0	0	0	0
1	1	0	0	0	0
2	0	2	2	0	0
2	0	2	2	0	0
1	1	0	0	0	0
1	1	0	0	0	0
2	0	2	2	0	0
2	0	2	2	0	0
3	0	0	0	3	0
3	0	0	0	3	0
4	0	0	0	0	4
4	0	0	0	0	4
3	0	0	0	3	0
3	0	0	0	3	0
4	0	0	0	0	4
4	0	0	0	0	4

+

0	0	0	0	0	0
0	0	0	0	0	0
2	2	0	0	0	0
2	2	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
2	2	0	0	0	0
2	2	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

+

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
3	3	0	0	0	0
3	3	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

+

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
3	3	0	0	0	0
3	3	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

反卷积

转置卷积
(transposed
convolution)

卷积矩阵 ->
转置

$$k = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

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

*

1
2
3
4

=

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

反卷积

反射:

1个值 对应一个向量

可视为映射 为 向量中的非 0 元素

$$k = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

$\alpha 1 =$

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

$$[\alpha 1 \quad \alpha 2 \quad \alpha 3 \quad \alpha 4] *$$

1
2
3
4

=

$$1 * \alpha 1 + 2 * \alpha 2 + 3 * \alpha 3 + 4 * \alpha 4$$

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

反卷积

$$4 * k = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad A2 = \begin{bmatrix} 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 3 & 3 & 4 & 4 \\ 3 & 3 & 4 & 4 \end{bmatrix}$$

卷积，移动步长 (stride) 为 2

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

$$\begin{array}{|c|} \hline 1 \\ \hline 1 \\ \hline 2 \\ \hline 2 \\ \hline 1 \\ \hline 1 \\ \hline 2 \\ \hline 2 \\ \hline 3 \\ \hline 3 \\ \hline 3 \\ \hline 3 \\ \hline 4 \\ \hline 4 \\ \hline 3 \\ \hline 3 \\ \hline 4 \\ \hline 4 \\ \hline \end{array}
 *
 \begin{array}{|c|} \hline 1 \\ \hline 1 \\ \hline 2 \\ \hline 2 \\ \hline 1 \\ \hline 1 \\ \hline 2 \\ \hline 2 \\ \hline 3 \\ \hline 3 \\ \hline 3 \\ \hline 3 \\ \hline 4 \\ \hline 4 \\ \hline 3 \\ \hline 3 \\ \hline 4 \\ \hline 4 \\ \hline \end{array}
 = 4
 \begin{array}{|c|} \hline 1 \\ \hline 2 \\ \hline 3 \\ \hline 4 \\ \hline \end{array}$$



卷积与反(转置)卷积

- 转置卷积和普通卷积本质相同，建立一些值之间的关系
转置卷积是1对多，而普通卷积是多对1；
- 可以使用转置卷积来进行上采样；
- 转置卷积中的权重是可以被学习的；
- 转置卷积并非是拿一个已有的卷积矩阵的转置来作为权重矩阵 进行转置卷积操作；
- 转置卷积其实并不算卷积.但是可以把input矩阵中的某些位置填上0并进行普通卷积，来获得和转置卷积相同的output矩阵。

