

План

Что такое изображение

Как классифицировать

Свёртка (Convolution)

Отступ (Padding) Шаг (stride), Расширение (Dilation)

Свёрточные нейронные сети (ConvNet, CNN)

Pooling (агрегация, субдискретизация / subsampling)

Какие бывают свёртки

Dropout в свёрточных сетях

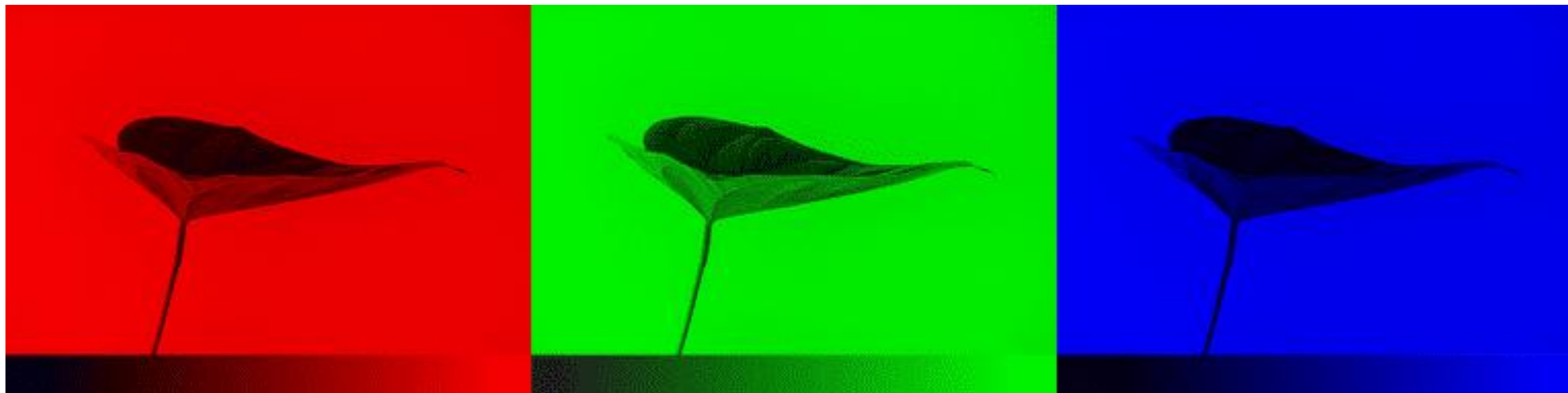
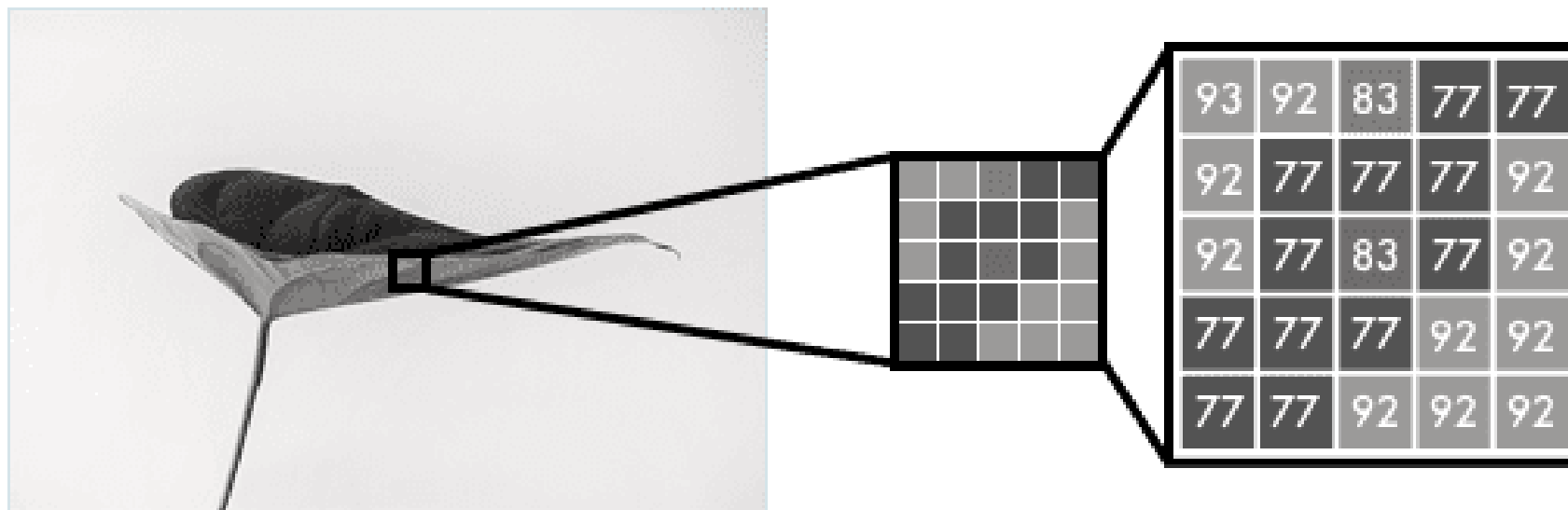
Что такое изображение – $H \times W$ -матрица



62	62	63	64	65	66	67	67	69	70	71	72	72	73	73	73	73	72	72	71	70	69	67	66	66	66	65	63	62	61	60	60
61	62	63	64	66	66	67	68	68	69	70	71	71	72	72	73	72	72	71	71	70	69	68	66	66	65	65	63	62	61	60	60
61	62	63	64	66	66	68	68	69	70	70	71	72	73	73	73	72	72	71	71	69	68	67	66	66	65	65	64	63	62	61	61
61	63	64	64	66	67	68	68	68	69	70	71	71	73	73	74	73	73	73	71	70	69	68	66	66	65	64	63	62	61	61	60
61	63	64	65	67	68	69	69	70	70	71	71	72	55	53	69	72	72	71	71	70	69	68	67	66	65	64	63	62	60	60	60
63	64	65	66	67	68	69	69	70	70	71	72	42	4	5	11	48	72	71	71	69	69	68	67	66	65	64	62	62	60	59	59
63	65	66	66	68	68	69	70	71	71	71	72	18	4	4	7	8	66	71	70	69	68	68	67	66	65	64	63	61	59	59	58
63	65	67	67	68	69	69	70	71	71	72	64	4	27	24	54	33	29	52	64	68	68	67	66	65	64	63	62	61	59	58	58
64	65	66	66	68	69	70	71	41	24	24	12	17	24	48	60	37	43	38	52	66	68	67	66	65	64	63	61	60	59	58	57
65	66	67	67	68	69	71	49	6	6	6	5	34	36	12	47	34	17	29	54	43	63	67	66	65	64	63	62	60	59	58	57
64	65	66	66	68	69	38	6	6	5	5	7	16	19	4	47	44	27	24	40	67	66	66	65	65	64	63	61	60	59	58	57
63	64	65	65	67	30	6	6	5	5	5	6	8	9	20	27	51	78	41	44	66	65	65	65	65	64	63	62	60	59	58	57
63	64	65	65	34	5	5	5	5	5	5	5	4	19	6	7	54	64	20	59	65	65	64	64	64	63	62	61	60	59	57	56
63	64	64	65	14	5	6	5	5	4	5	4	18	7	5	4	19	10	11	65	64	64	64	63	61	66	62	61	60	59	58	56
63	64	64	65	53	7	4	5	6	6	7	10	6	5	5	4	21	24	18	64	64	64	63	62	64	65	62	62	60	59	58	57
64	64	64	64	65	50	4	4	4	5	11	16	6	6	4	6	35	16	26	66	64	64	63	61	72	67	63	62	61	59	58	57
64	64	64	64	65	46	4	4	4	5	6	9	8	5	29	10	43	56	29	57	64	64	63	61	70	67	62	64	65	59	59	57
64	64	64	65	66	27	5	4	4	5	6	6	6	18	68	20	57	60	48	38	75	70	62	61	70	67	62	61	60	59	58	58
49	50	62	65	57	5	5	6	5	6	6	6	6	41	59	28	60	58	44	22	63	71	72	60	69	68	61	60	58	59	59	58
42	52	57	52	26	5	5	5	5	5	5	5	5	70	50	43	61	62	64	39	42	64	60	62	56	63	65	65	67	61	53	53
32	32	32	33	6	5	5	5	5	5	5	6	11	39	21	33	51	50	45	46	18	32	36	33	23	44	70	71	51	42	27	31
50	50	51	39	5	5	5	5	6	5	6	6	42	69	28	34	42	39	43	37	26	29	40	26	29	26	35	42	35	33	18	19
52	53	51	22	5	5	5	5	6	5	6	5	44	56	17	51	54	53	54	56	51	22	54	54	55	55	54	53	53	53	52	52
54	54	53	8	5	5	5	5	6	5	6	13	52	42	21	51	54	51	49	49	50	22	41	45	42	42	41	40	41	44	43	42
52	52	54	36	8	5	5	6	6	5	6	28	55	32	32	54	53	51	51	51	51	44	25	51	51	49	49	50	49	48	46	46
54	54	52	53	30	7	5	6	6	5	6	40	54	29	52	51	53	56	55	52	52	51	38	52	52	50	49	46	46	45	46	47
51	52	51	53	27	14	5	4	5	4	7	47	51	21	39	49	47	49	52	52	52	49	35	31	48	46	47	47	47	46	46	43
48	50	51	53	25	14	17	8	4	4	17	46	40	18	43	47	46	49	52	54	53	53	54	18	50	49	46	47	47	47	47	45
49	49	49	49	22	12	20	24	6	14	35	51	39	48	48	50	51	51	49	51	51	52	50	41	58	48	47	47	47	45	45	46
51	49	50	50	22	13	19	36	13	12	42	50	40	73	50	50	50	49	48	49	49	48	49	45	51	46	44	44	44	42	45	47
47	49	49	47	20	16	26	39	21	15	36	48	42	61	47	48	51	47	50	51	51	51	49	47	47	52	47	47	44	43	45	46
48	50	48	52	19	13	33	36	18	18	36	49	51	54	47	47	49	46	46	49	49	49	49	47	44	53	44	48	44	46	46	45

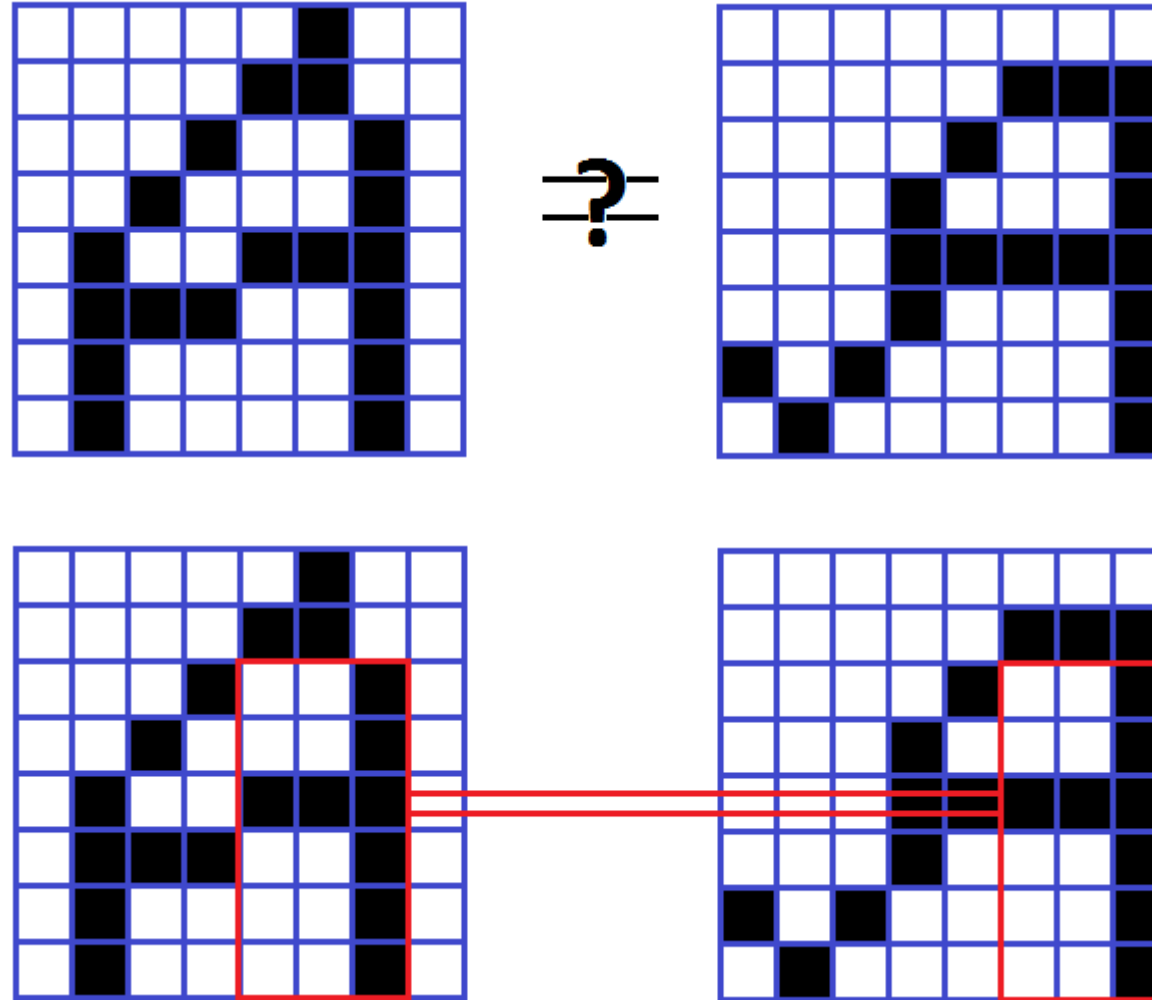
«чёрно-белое» (в градациях серого) – целочисленная матрица

Что такое изображение – трёхмерный $C \times H \times W$ -тензор



цветное – 3-х мерная целочисленная матрица (тензор)

Как сравнивать изображения



хотим устойчивость к сдвигам / небольшим поворотам / сжатиям-растяжениям
хотим нахождения паттернов

Что такое свёртка в математике

$$(f * g)(x) = \int_{\mathbb{R}^n} f(y)g(x - y)dy$$

Дискретный случай:

$$(f * g)(i) = \sum_{j \in \mathbb{Z}} f(j)g(i - j)$$

Случай, когда одна функция равна нулю за пределами окна...

свёртка коммутативна, а после этого предположения появляется асимметрия

$$(f * g)(i) = \sum_{j=-k}^k f(j)g(i - j) = f(-k)g(i - k) + \dots + f(0)g(i) + \dots + f(+k)g(i + k)$$

[https://ru.wikipedia.org/wiki/Свёртка_\(математический_анализ\)](https://ru.wikipedia.org/wiki/Свёртка_(математический_анализ))

Что такое 1-D свёртка (Convolution)

пусть $I = (i_1, \dots, i_n) \in \mathbb{R}^n$ – сигнал / массив,

$K = (k_1, \dots, k_r) \in \mathbb{R}^r$ – ядро свёртки, тогда свёртка:

$$I * K = (i_1 k_1 + \dots + i_r k_r, i_2 k_1 + \dots + i_{r+1} k_r, \dots, i_{n-r+1} k_1 + \dots + i_n k_r) \in \mathbb{R}^{n-r+1}$$

-1	0	1							
1	2	3	4	5	5	4	3	2	1
	3 - 1								

	-1	0	1						
1	2	3	4	5	5	4	3	2	1
	2	4 - 2							

							-1	0	1
1	2	3	4	5	5	4	3	2	1
	2	2	2	1	-1	-2	-2	1 - 3	

Отступ (Padding)

Нулевой

-1	0	2	0	-1				
0	0	1	2	3	4	5	0	0
		-1	0	0	6	7		

Константный

-1	0	2	0	-1				
1	1	1	2	3	4	5	5	5
		-2	-1	0	1	2		

Зеркальный

-1	0	2	0	-1				
2	1	1	2	3	4	5	5	4
		-3	-1	0	1	3		

Циклический

-1	0	2	0	-1				
4	5	1	2	3	4	5	1	2
		-5	-5	0	5	5		

позволяет получать вектор нужной длины

Шаг (Stride)

свёртка с шагом 3

-1	0	1							
1	2	3	4	5	5	4	3	2	1
	3 - 1								

			-1	0	1				
1	2	3	4	5	5	4	3	2	1
	3 - 1			5 - 4					

						-1	0	1	
1	2	3	4	5	5	4	3	2	1
	2			1			-4 + 2		

2	1	-2
---	---	----

позволяет получать вектор небольшой длины
и делать его элементы менее коррелированными

Расширение (Dilation)

-1	0	1
----	---	---

свёртка с расширением 3

-1			0			1			
1	2	3	4	5	5	4	3	2	1
			4 - 1						

	-1			0			1		
1	2	3	4	5	5	4	3	2	1
			3	3 - 2					

			-1			0			1
1	2	3	4	5	5	4	3	2	1
			3	1	-1	-3			

позволяет увеличить область действия свёртки («рецептивную зону»)

Минутка кода: свёртка – слой сети

```
import torch

x = torch.Tensor([1, 2, 3, 4, 5, 5, 4, 3, 2, 1]).view(1,1,10)
c = torch.nn.Conv1d(in_channels=1, out_channels=1,
                    kernel_size=3, bias=False)
c.weight = torch.nn.Parameter(torch.Tensor([-1, 0, 1]).view(1,1,3))
c.weight.data.copy_(torch.tensor([-1, 0, 1])) # лучше так
c(x)
tensor([[[[ 2.,  2.,  2.,  1., -1., -2., -2., -2.]]]], grad_fn=<SqueezeBackward1>)

c = torch.nn.Conv1d(in_channels=1, out_channels=1,
                    kernel_size=3, bias=False, stride=3)
c(x)
tensor([[[[ 2.,  1., -2.]]]], grad_fn=<SqueezeBackward1>)

c = torch.nn.Conv1d(in_channels=1, out_channels=1,
                    kernel_size=3, bias=False, dilation=3)
c(x)
tensor([[[[ 3.,  1., -1., -3.]]]], grad_fn=<SqueezeBackward1>)
```

2-D свёртка (Convolution)

$$(I * K)_{xy} = \sum_{i=1}^h \sum_{j=1}^r K_{ij} I_{x+i-1, y+j-1}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 1 & 0 & 2 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} & \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} \\ \begin{bmatrix} 3 & 4 \\ 1 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} & \begin{bmatrix} 4 & 5 \\ 0 & 2 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 5 & 3 \end{bmatrix}$$

может быть немного другая индексация

хорошее объяснение: Vincent Dumoulin, Francesco Visin «A guide to convolution arithmetic for deep learning» <https://arxiv.org/pdf/1603.07285.pdf>

Свёртка (Convolution)

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

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

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

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

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

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

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

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

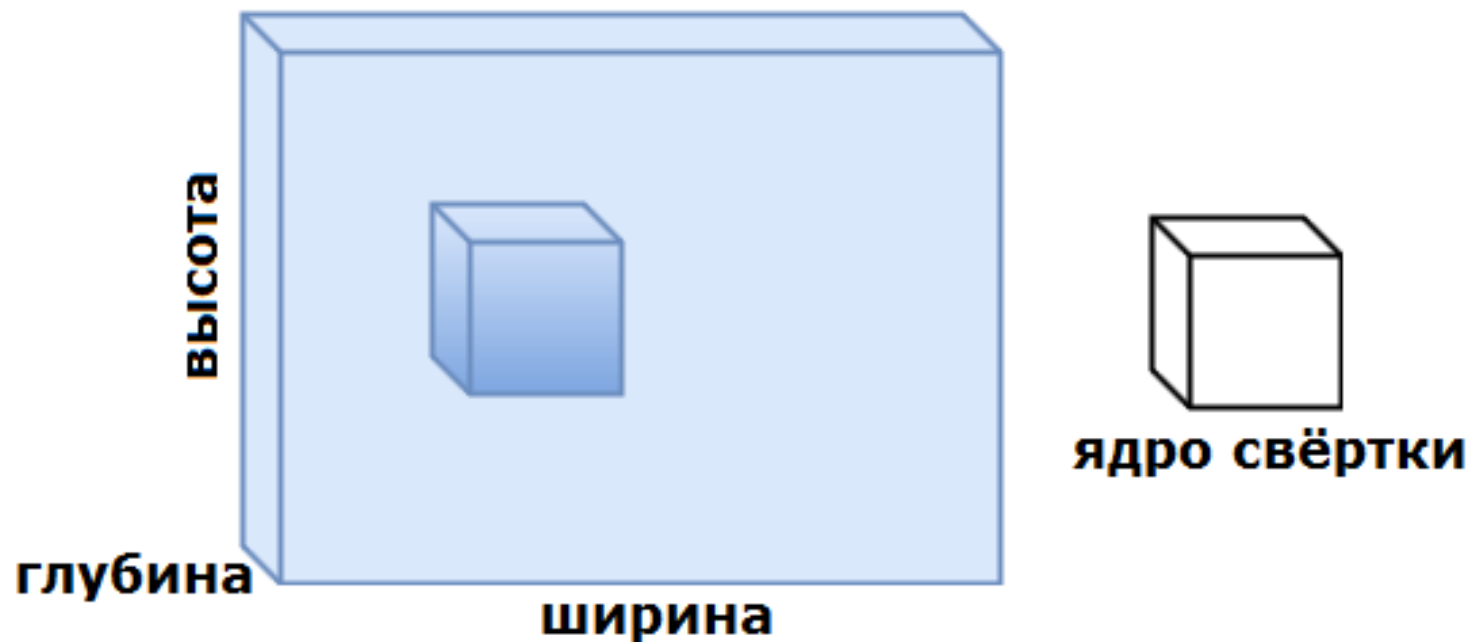
3	3	2	1	0
0	0	1	3	1
3	1 ₀	2 ₁	2 ₂	3
2	0 ₂	0 ₂	2 ₀	2
2	0 ₀	0 ₁	0 ₂	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

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

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Свёртка (Convolution)



Глубина (depth) / число каналов

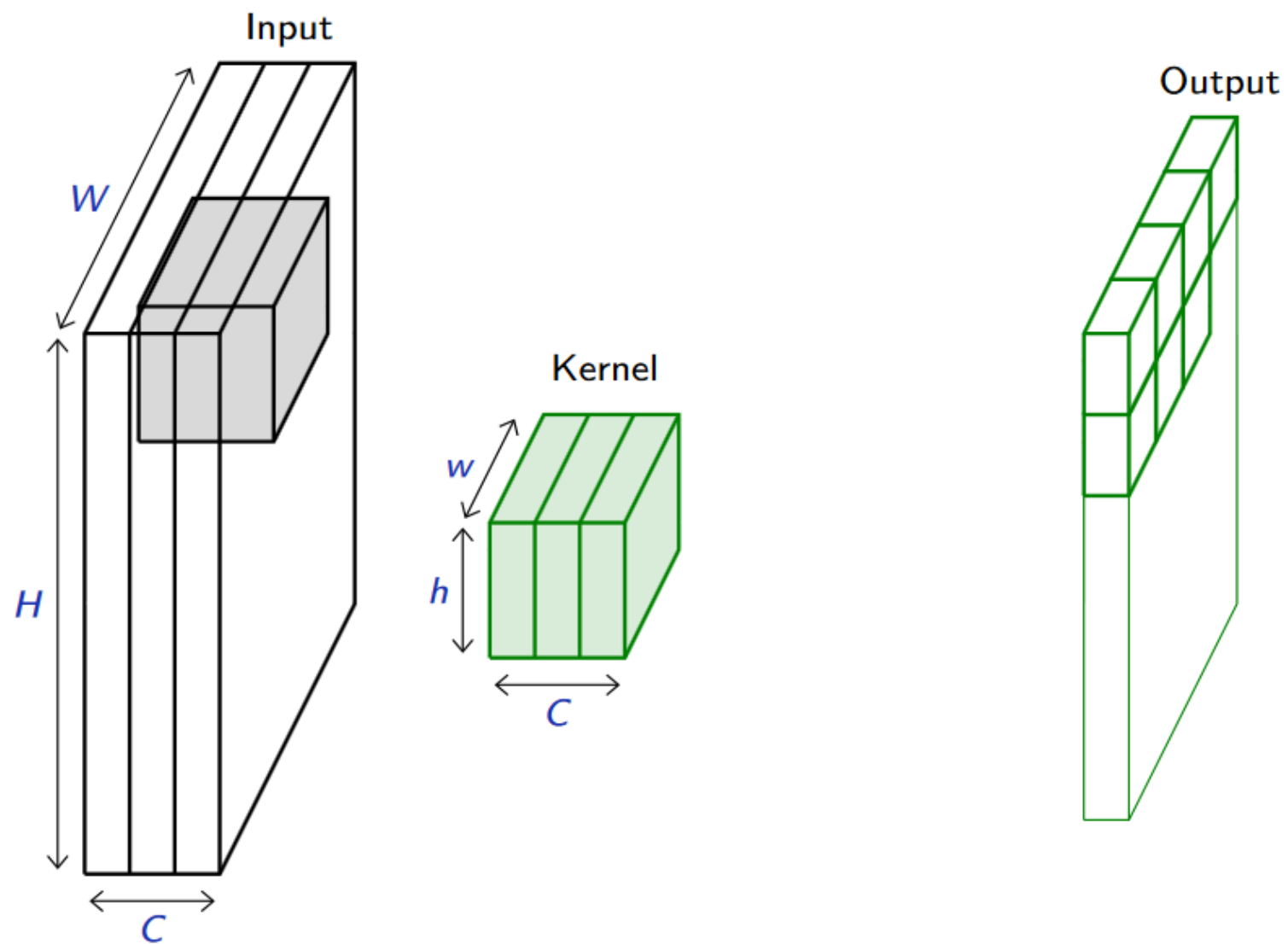
Высота (height) и ширина (width) тензора (изображения) / ядра

Шаг (stride) — на сколько смещается ядро при вычислении свёрток (чем больше, тем меньше размер итогового изображения)

Отступ (padding) – для дополнения изображения нулями по краям

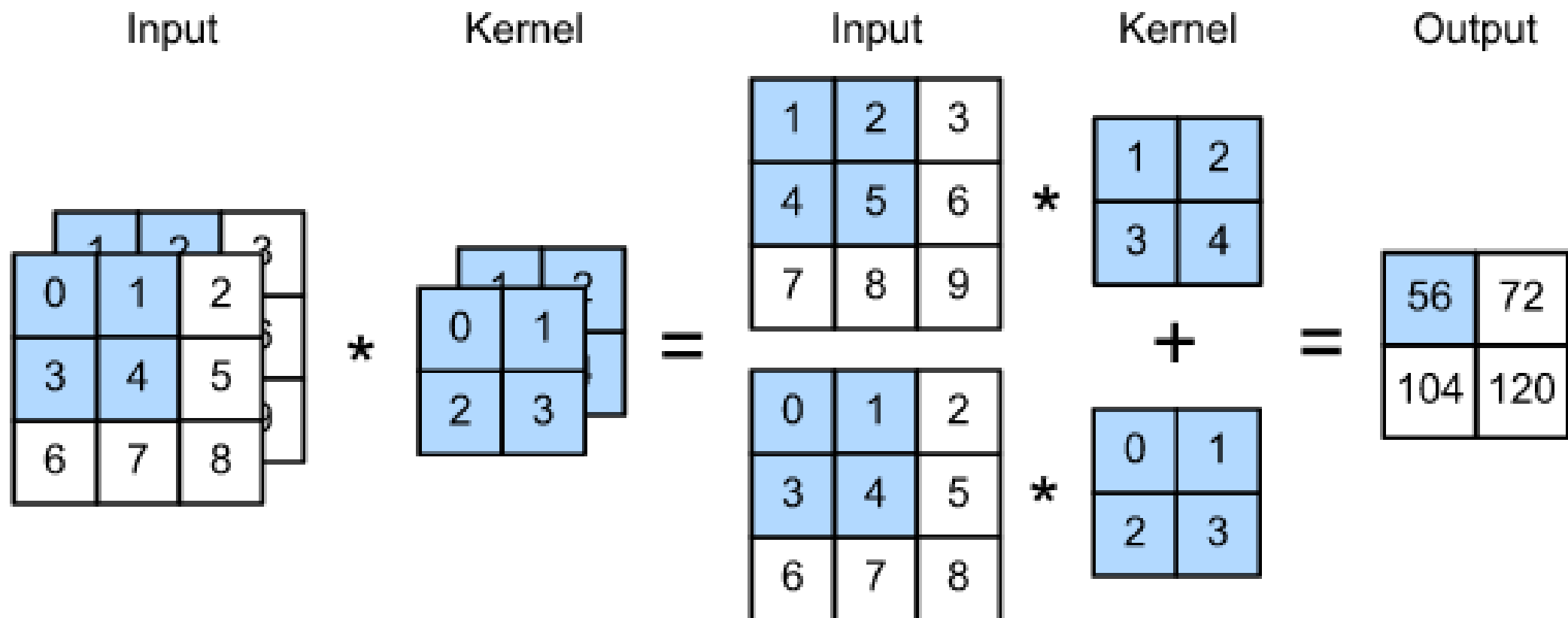
Ядро (kernel) или фильтр (filter) – размерность как у предыдущего тензора; в 3D длина и ширина меньше (глубина совпадает)

Свёртка (Convolution): глубина

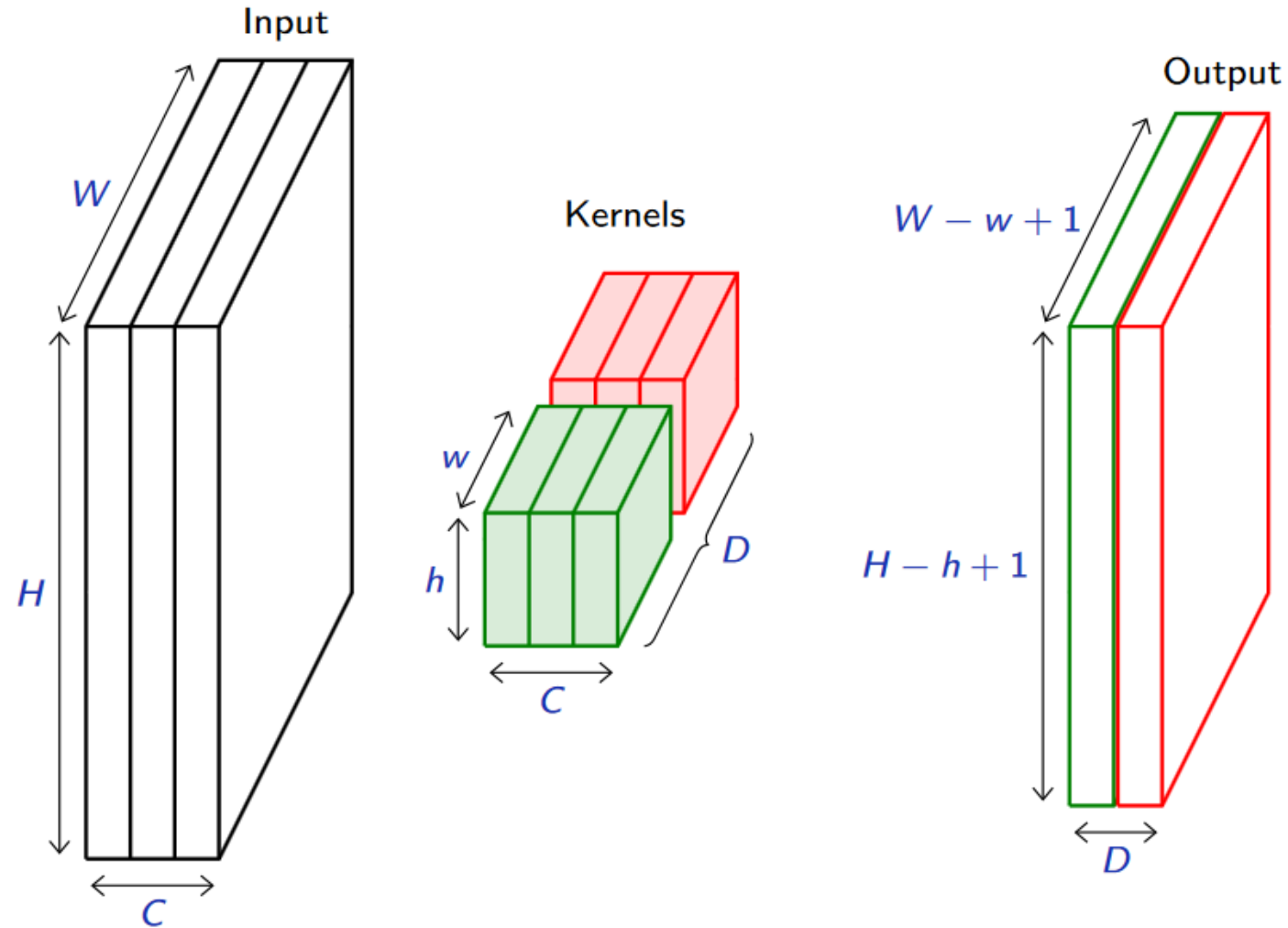


глубина тензора (число каналов) = глубина свёртки

Свёртка (Convolution): глубина



Свёртка (Convolution): применение нескольких свёрток

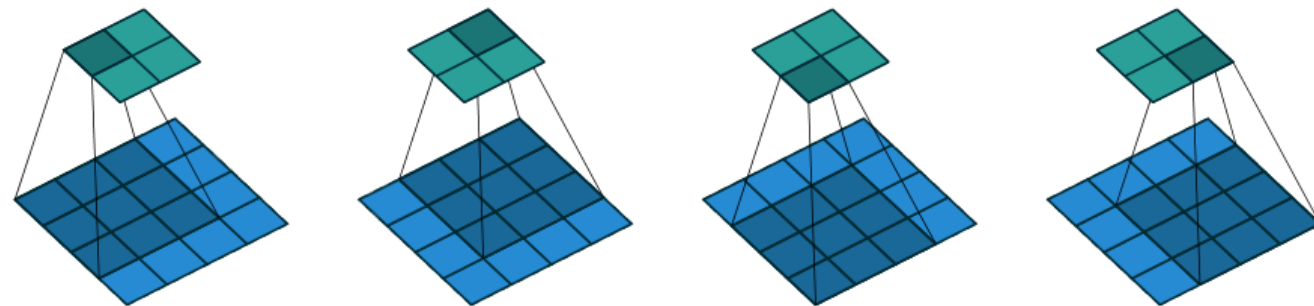


**каждая свёртка – 1 «лист» на выходе, k свёрток – k -канальный выход
получаем на выходе тензор, глубина = число применяемых свёрток
свёрточный слой (для картинок) – 4D-массив $C_{out} \times C_{in} \times h \times w$**

Свёртка (Convolution): отступы (padding) – чтобы сохранялись размеры изображения

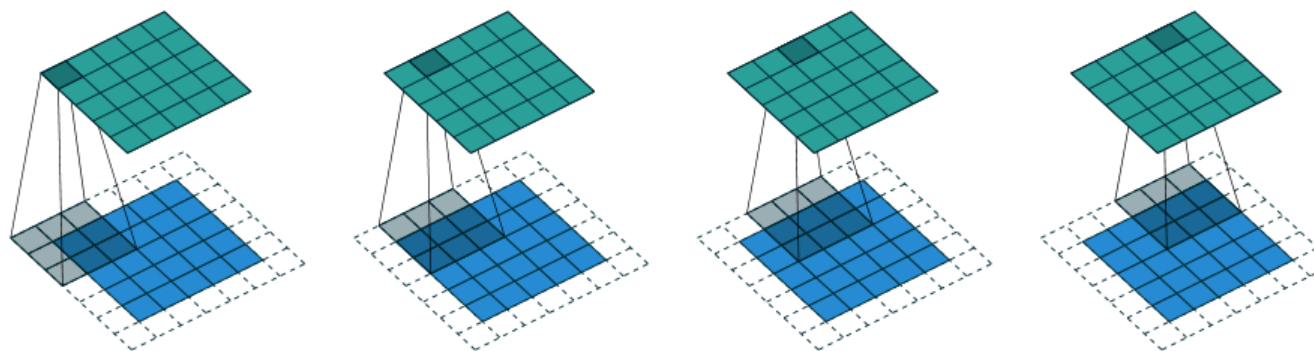
без отступов

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 1 & 0 & 2 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 5 & 3 \end{bmatrix}$$



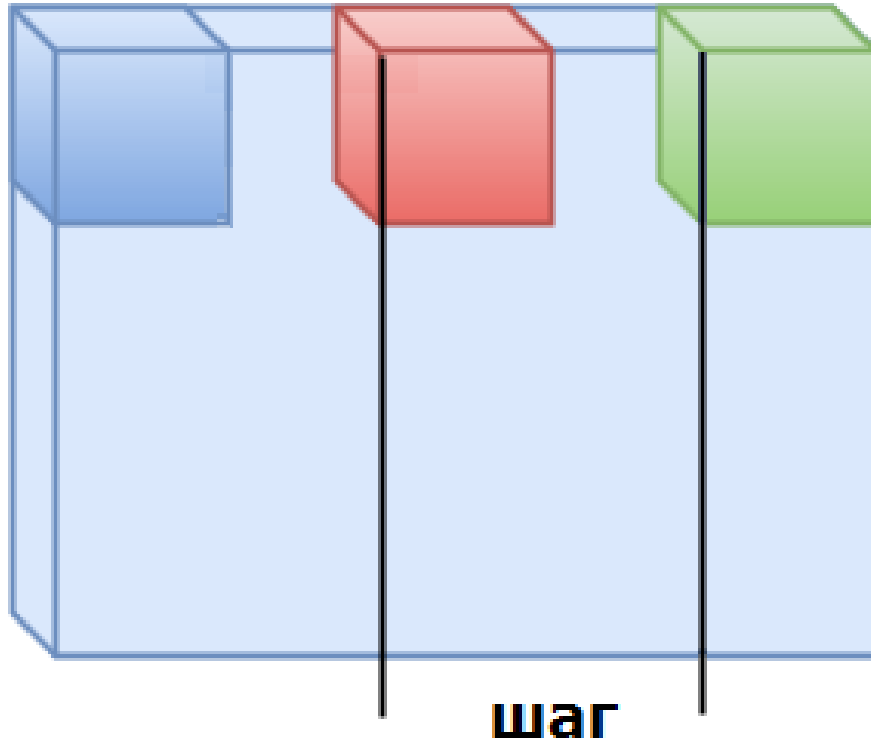
с отступами

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 1 & 0 & 2 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} \equiv$$
$$\equiv \begin{bmatrix} 1 & 2 & 3 & 0 \\ 3 & 4 & 5 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 5 \\ 5 & 3 & 4 \\ 0 & 2 & 0 \end{bmatrix}$$



Свёртка (Convolution): шаг (stride)

сдвигаемся при вычислении свёртки (можно в каждой её размерности)



с шагом 2

$$\begin{bmatrix} 1 & 2 & 3 & 1 & 0 \\ 3 & 4 & 5 & 2 & 1 \\ 1 & 0 & 2 & 3 & 4 \\ 0 & 1 & 3 & 1 & 2 \\ 1 & 2 & 4 & 2 & 3 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ -5 & 2 \end{bmatrix}$$

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

6.0	17.0	3.0
8.0	17.0	13.0
6.0	4.0	4.0

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

6.0	17.0	3.0
8.0	17.0	13.0
6.0	4.0	4.0

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

6.0	17.0	3.0
8.0	17.0	13.0
6.0	4.0	4.0

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

6.0	17.0	3.0
8.0	17.0	13.0
6.0	4.0	4.0

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

6.0	17.0	3.0
8.0	17.0	13.0
6.0	4.0	4.0

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

6.0	17.0	3.0
8.0	17.0	13.0
6.0	4.0	4.0

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

6.0	17.0	3.0
8.0	17.0	13.0
6.0	4.0	4.0

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

6.0	17.0	3.0
8.0	17.0	13.0
6.0	4.0	4.0

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

6.0	17.0	3.0
8.0	17.0	13.0
6.0	4.0	4.0

Свёртка (Convolution): минутка кода

```
torch.nn.Conv2d(in_channels: int,
                out_channels: int,
                kernel_size: Union[T, Tuple[T, T]],
                stride: Union[T, Tuple[T, T]] = 1,
                padding: Union[T, Tuple[T, T]] = 0,
                dilation: Union[T, Tuple[T, T]] = 1,
                groups: int = 1,
                bias: bool = True,
                padding_mode: str = 'zeros') # 'reflect', 'replicate', 'circular'
input = torch.randn(20, 16, 50, 100)
m = nn.Conv2d(16, 33, (3, 5), stride=(2, 1), padding=(4, 2),
              dilation=(3, 1))
output = m(input)
```

in_channels, out_channels – количество каналов на входе и выходе – должны делиться на groups!

kernel_size – размеры ядра

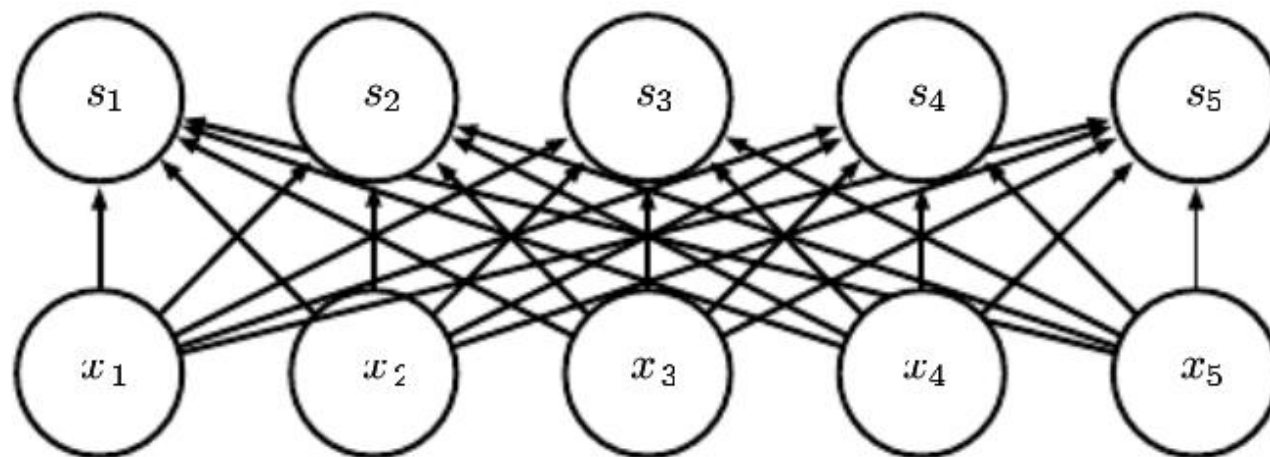
stride – смещение (можно понижать разрешение)

padding – отступы

dilation – расстояние между точками ядра (увеличивает область зависимости)

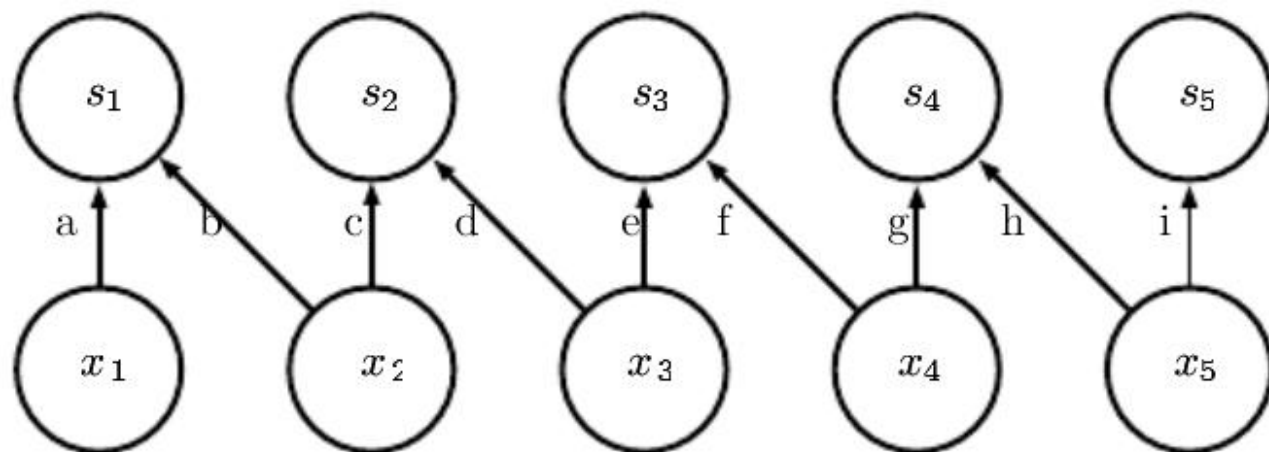
groups – опр. связи между входом и выходом

Полная связность (full connections)



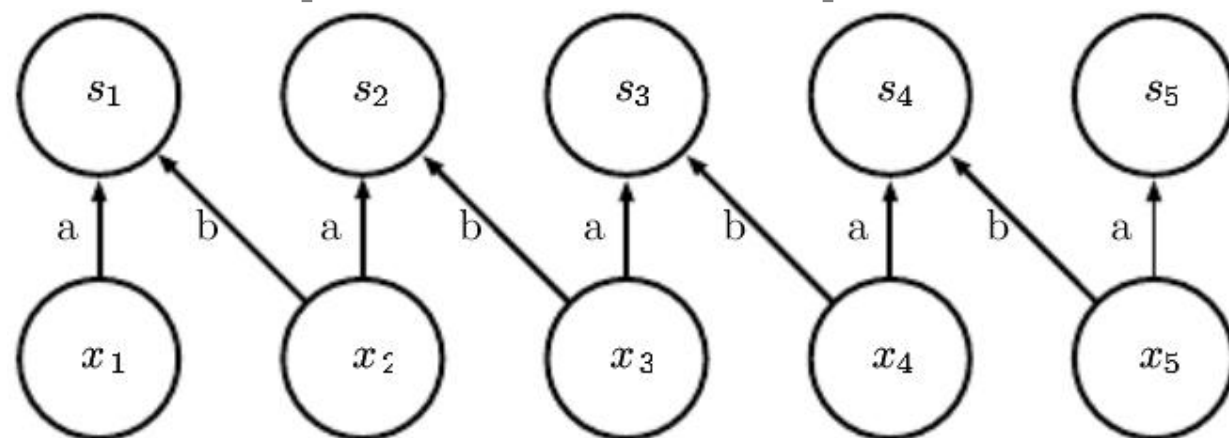
Локальная связность (local connections)

неразделяемая свёртка (unshared convolution)



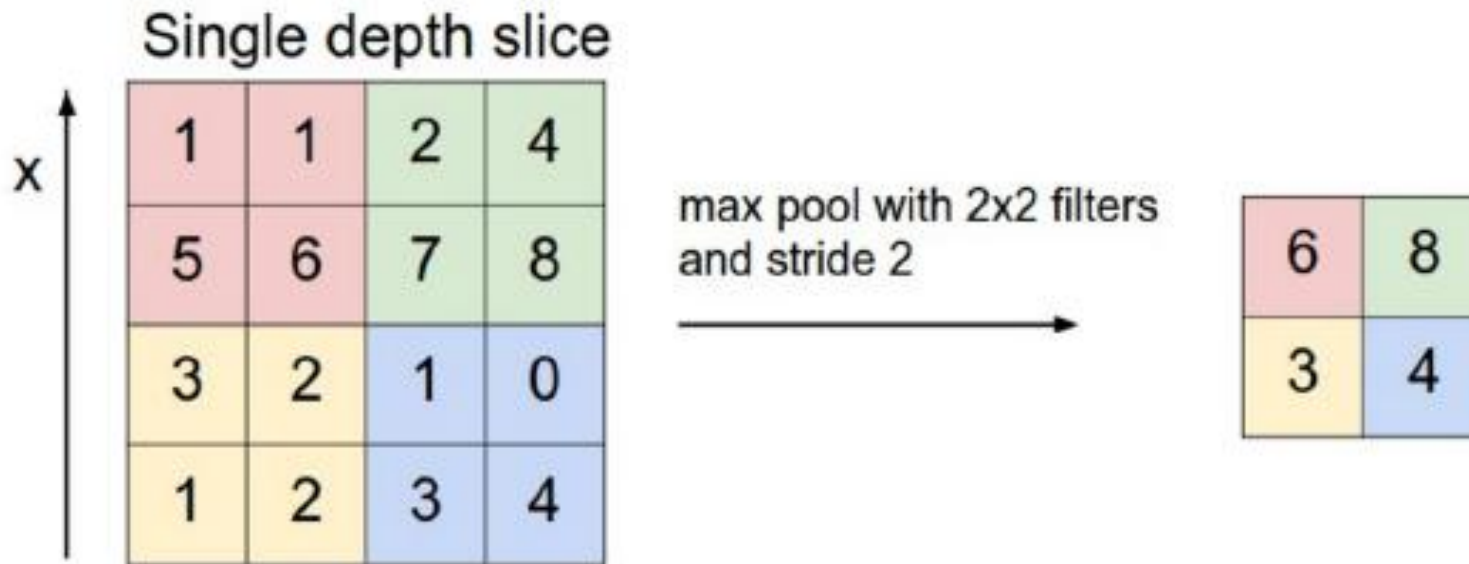
Свёртка (convolution)

разделяемая свёртка



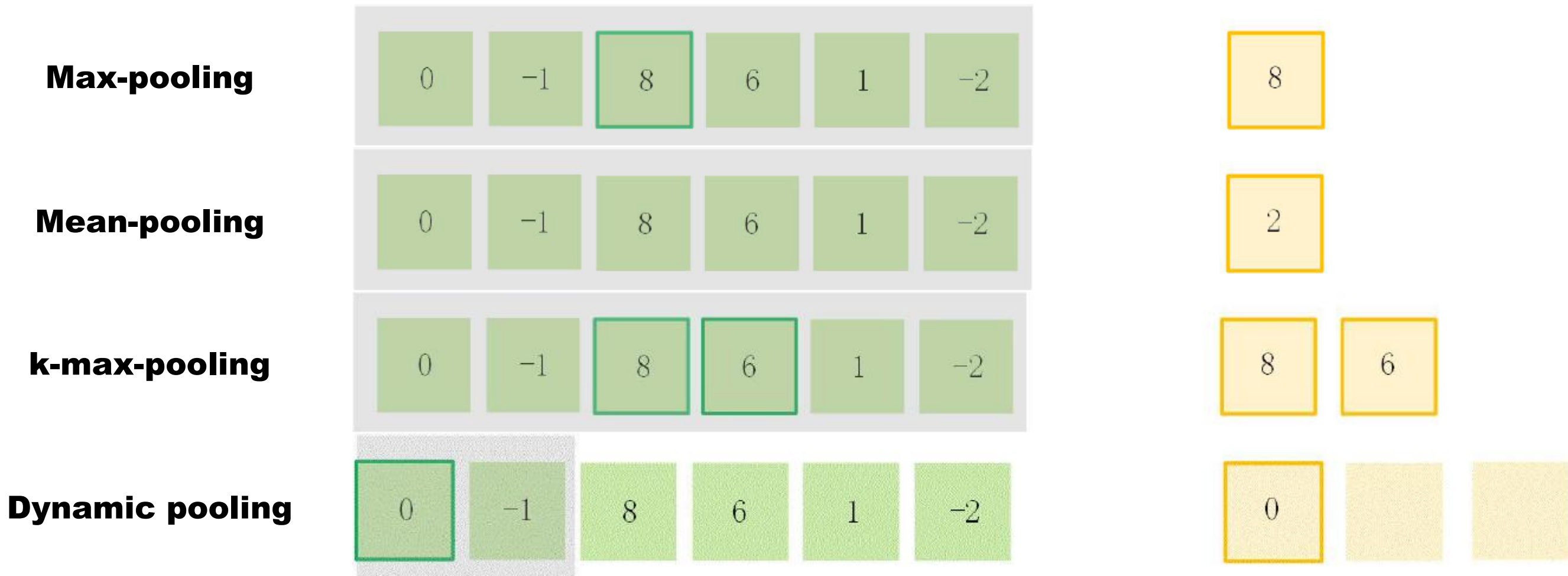
Pooling (агрегация, субдискретизация / subsampling)

для каждого признака канала надо определить,
нашли ли паттерн
используем функцию агрегации (mean, max, ...)



делается независимо по каналам \Rightarrow сохраняет число каналов (глубину тензора)

Агрегация (Pooling): виды пулинга



Динамический – длина может от чего-то зависеть

<http://www.phontron.com/class/nn4nlp2020/schedule.html>

Устройство слоя свёрточной НС:

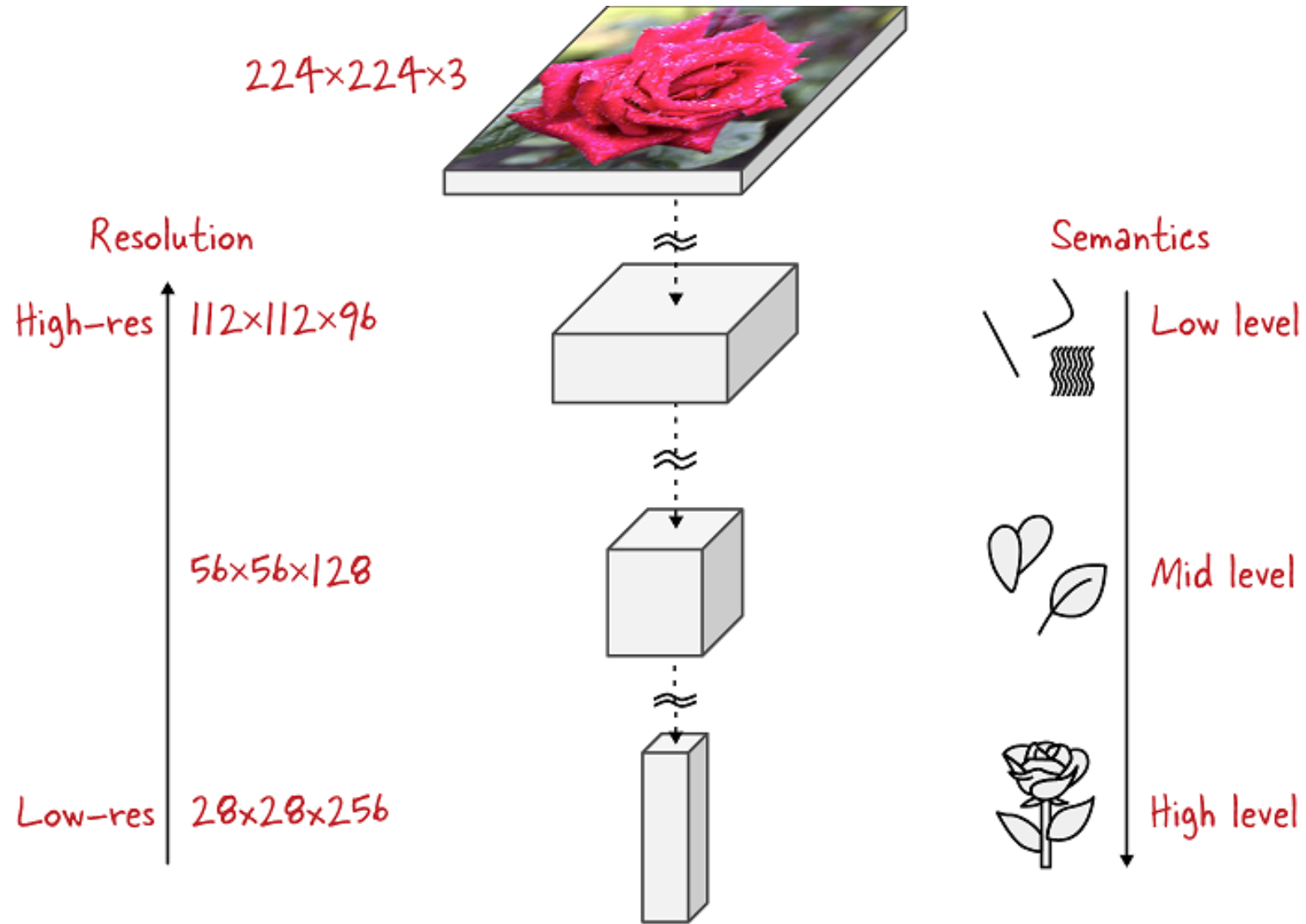
свёрточная часть: [свёртка → нелинейность → пулинг] × k

Мотивация:

- **разреженные взаимодействия (sparse interactions) / локальные признаки**
нет связи нейронов «каждый с каждым»
У свёрточных НС мало весов!!!
- **разделение параметров (parameter sharing)**
одна свёртка используется «по всему изображению»
⇒ мало параметров
- **инвариантные преобразования (equivariant representations)**
инвариантность относительно сдвига

<http://www.deeplearningbook.org/contents/convnets.html>

Визуализация признаков

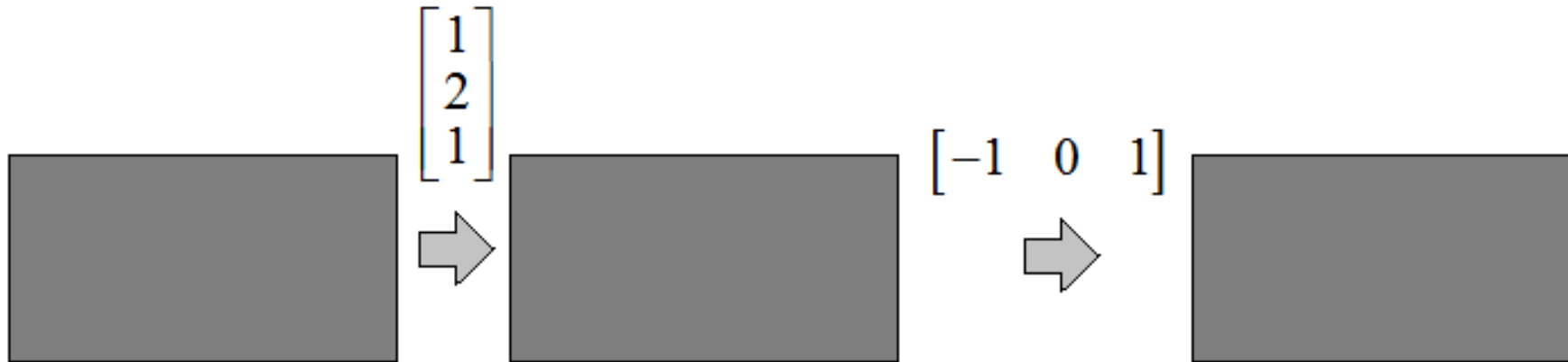


Какие бывают свёртки: Spatial Separable Convolutions

$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

**идея – факторизовать свёртку,
тогда параметров для обучения свёртки не k^2 , а $2k$**

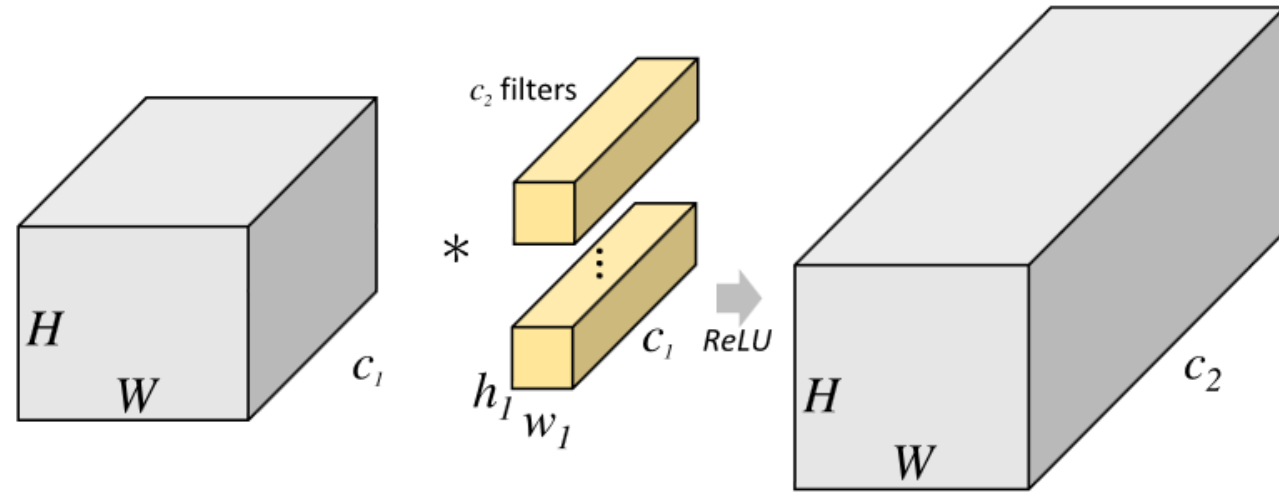
не все свёртки так представимы!



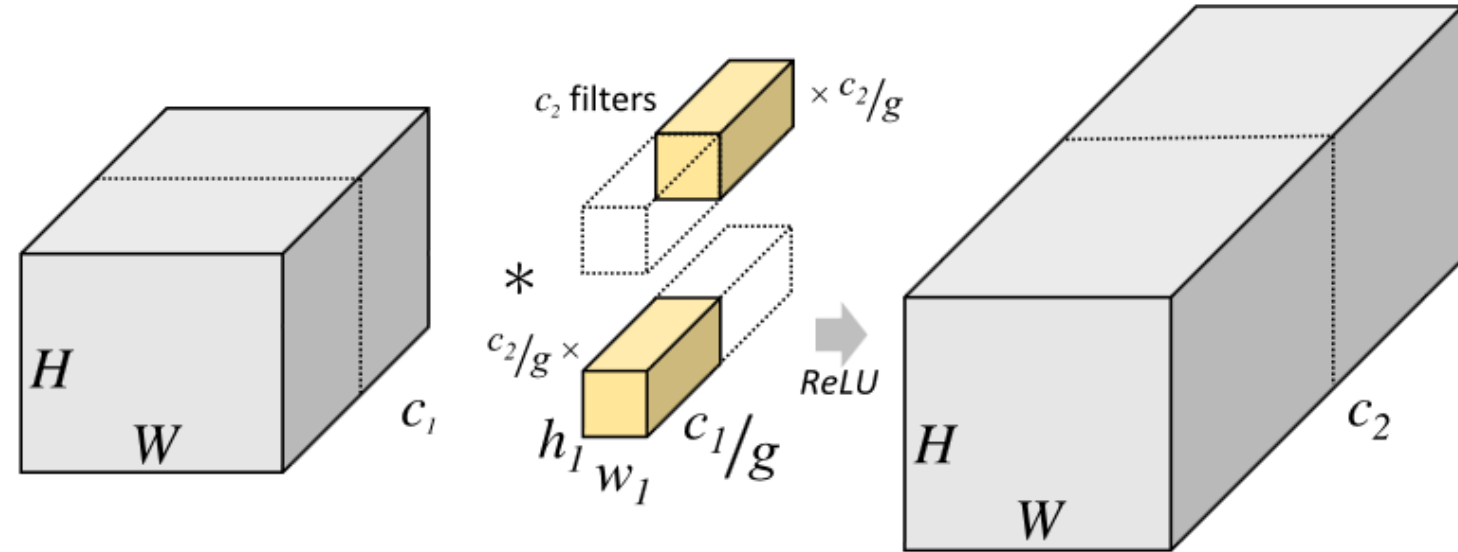
проводим сначала $k \times 1$ -свёртку, а потом $1 \times k$

Какие бывают свёртки: Group Convolutions

обычная ситуация



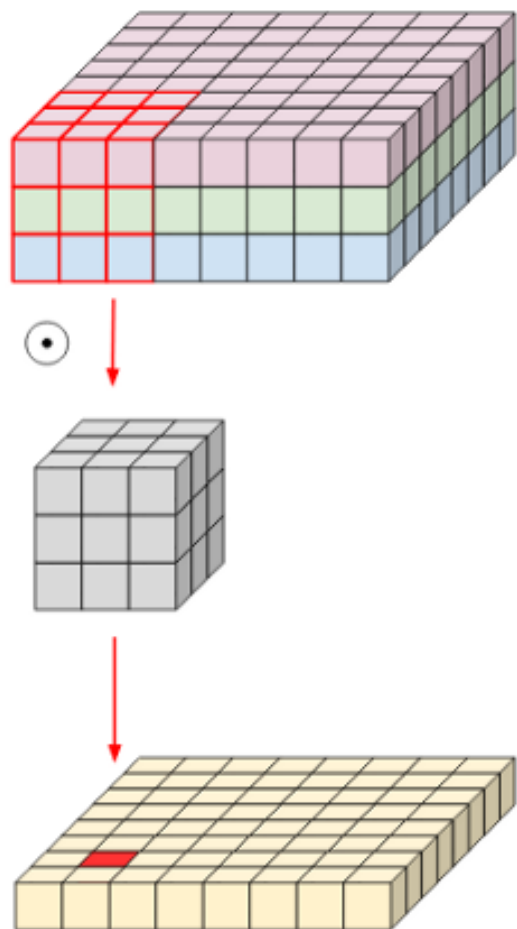
с разбиением на две группы



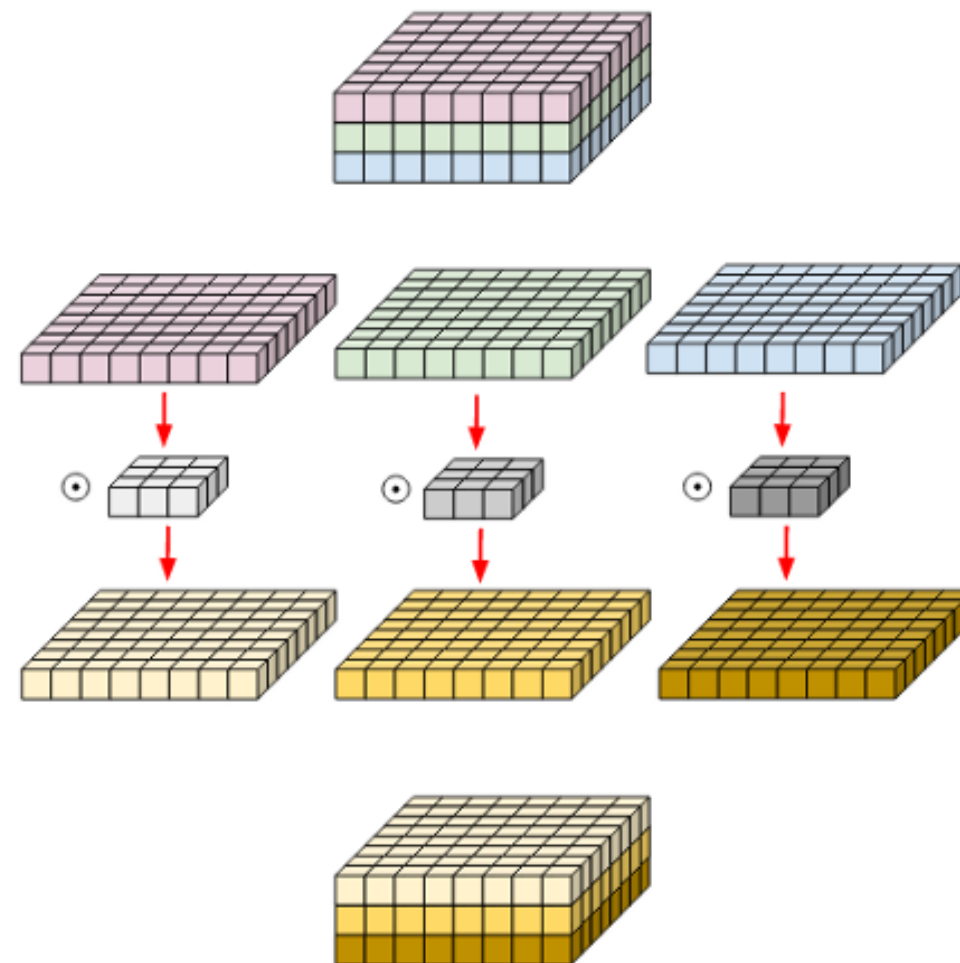
**идея из AlexNet, где были ограничения по памяти
могут быть лучшие (разреженные) признаковые представления
но выходные каналы зависят от узкой группы входных**

Какие бывают свёртки

convolution



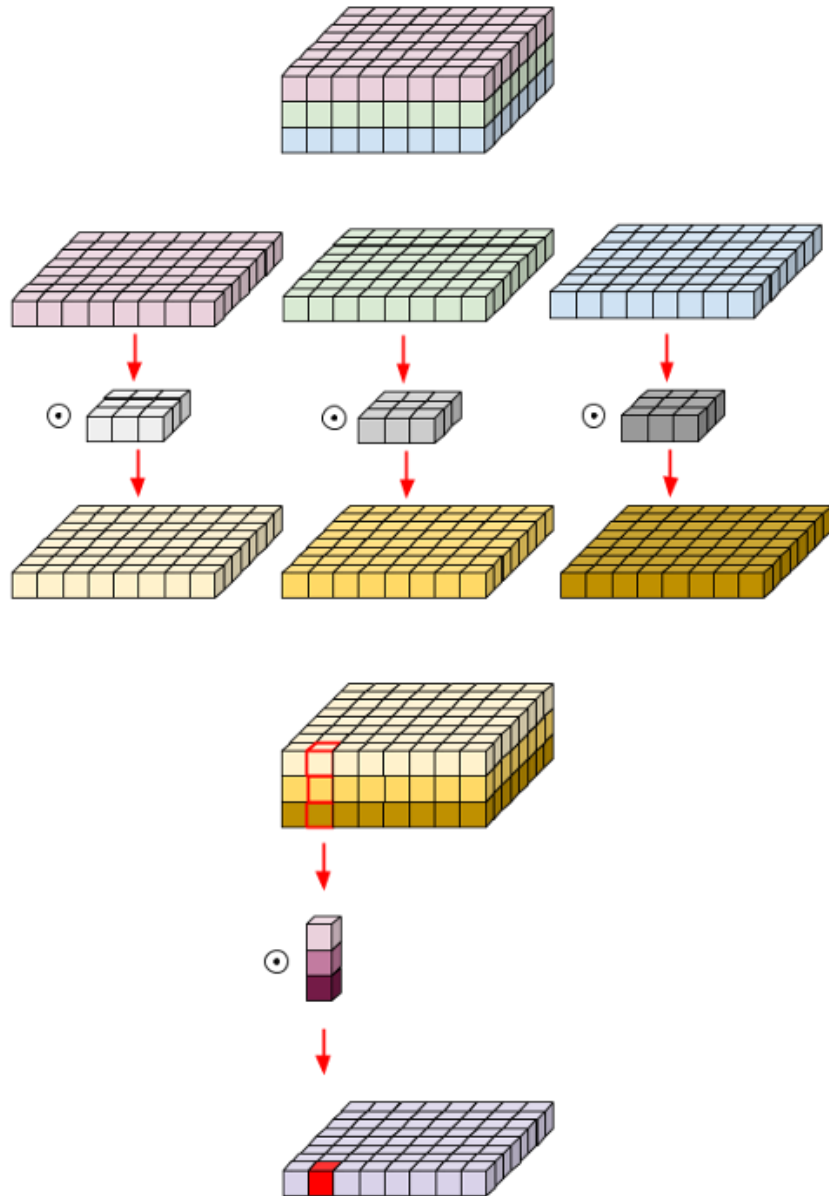
depth-wise convolution



каждый канал «сворачивается» отдельно

<https://eli.thegreenplace.net/2018/depthwise-separable-convolutions-for-machine-learning/>

Какие бывают свёртки: Depth-wise separable convolution



теперь результат зависит от всех каналов

$S=128$, $F=3$, $inC=3$, $outC=16$

Regular convolution:

Parameters:

$$3*3*3*16 = 432$$

Computation cost:

$$3*3*3*128*128*16 = \sim 7e6$$

Depthwise separable convolution:

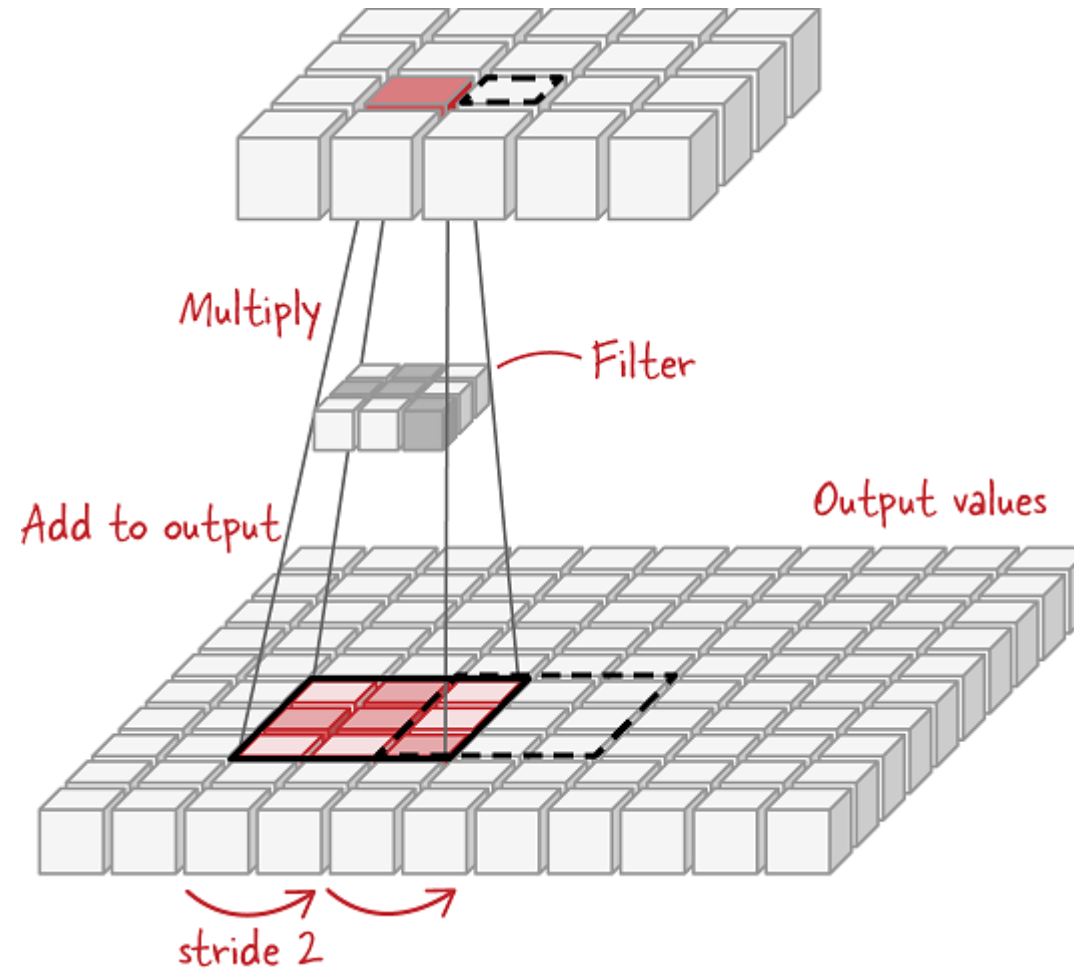
Parameters:

$$3*3*3+3*16 = 75$$

Computation cost:

$$3*3*3*128*128+128*128*3*16 \\ = \sim 1.2e6$$

Transposed convolution (deconvolution / upconvolution / conv-transpose)



термин «deconvolution» считается плохим
~ learnable upsampling operation
[Practical Machine Learning for Computer Vision]