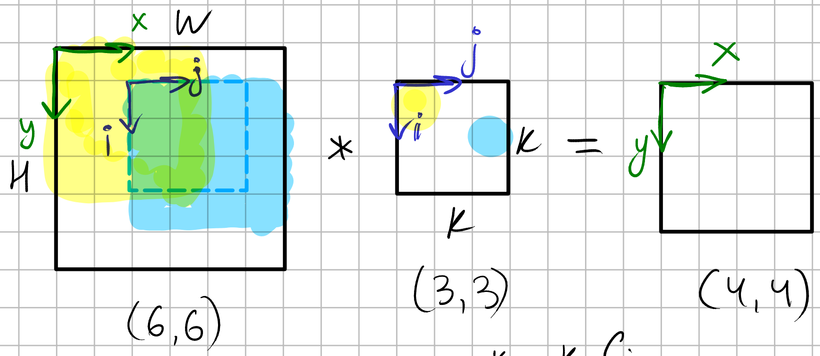
# 1. Алгоритм свёртки (stride = 1, padding = 0, dilation = 1)

$$X \in \mathbb{R}^{N \times H \times W \times C_{in}} * F \in \mathbb{R}^{K \times K \times C_{in} \times C_{out}} = Y \in \mathbb{R}^{N \times (H-K+1) \times (W-K+1) \times C_{out}}$$

$(N, H, W, C_{in})$  $(K, K, C_{in}, C_{out})$  $(N, H-K+1, W-K+1, C_{out})$

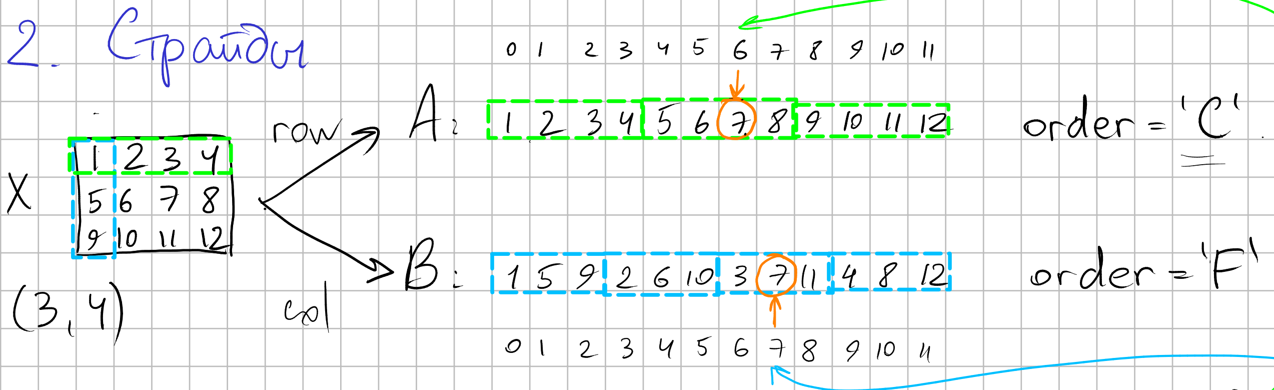



2d

$$Y[y,x] = \sum_{i=1}^{K}\sum_{j=1}^{K} X[y+i, x+j] \times F[i,j]$$

4d: $$Y[n,y,x,c_{out}] = \sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{c_{in}=1}^{C_{in}} X[n, y+i, x+j, c_{in}] \times F[i, j, c_{in}, c_{out}]$$

# 2. Страйды




A.strides: $(4, 1)$  $A[1,2] = A.data[\overbrace{1 \times 4 + 2 \times 1}^{=6}] = 7$

B.strides: $(1, 3)$  $B[1,2] = B.data[\overbrace{1 \times 1 + 2 \times 3}^{=7}] = 7$

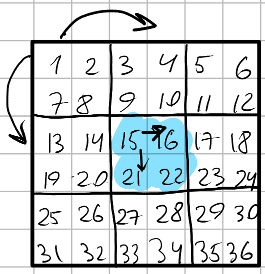
$$X[\mathbf{i},\mathbf{j}] = X.data[\mathbf{i} \times X.strides[0] + \mathbf{j} \times X.strides[1]]$$

Default: shape $= (n_1, n_2, \dots, n_{d-1}, n_d)$
(row-major) strides $= (n_2 \times \dots \times n_d, n_3 \times \dots \times n_d, \dots, n_d, 1)$
||
Compact

shape: $(3, 4, 5)$

stride: $(20, 5, 1)$

## 3. Плиточное разбиение (Tiling)

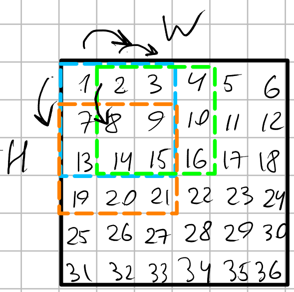


TILE = 2 : $(6, 6) \longrightarrow (3, 3, 2, 2)$

$(M, N) \longrightarrow \left(\frac{M}{TILE}, \frac{N}{TILE}, TILE, TILE\right)$

Strides : $(12, 2, 6, 1)$

## 4. Im2Col (2D)




SHAPE

$(H, W) \xrightarrow{im2col} (H-K+1, W-K+1, K, K) \xrightarrow{reshape} ((H-K+1)\times(W-K+1), K^2)$

A      B      C

matrix mul.

$C @ W.reshape(K^2) : ((H-K+1)\times(W-K+1), )$

↓ reshape

$(H-K+1, W-K+1)$

B[0, 0] =

B[0, 1] =

B[1, 1] =

$(6, 1, 6, 1).$

$B.strides = (W, 1, W, 1)$

## 5. Im2Col (4D)

$(N, H, W, C_{in}) \xrightarrow{im2col} (N, H-K+1, W-K+1, K, K, C_{in}) \xrightarrow{reshape}$

$\xrightarrow{reshape} (N\times(H-K+1)\times(W-K+1), C_{in}\times K^2) \xrightarrow[(C_{in}\times K^2, C_{out})]{matmul} (N\times(H-K+1)\times(W-K+1), C_{out}) \xrightarrow{reshape}$

$\xrightarrow{reshape} (N, H-K+1, W-K+1, C_{out})$

$A.strides = (H\times W\times C_{in}, W\times C_{in}, C_{in}, 1)$

$B.strides = (H\times W\times C_{in}, W\times C_{in}, C_{in}, W\times C_{in}, C_{in}, 1)$

## 6. ConvTranspose

$$\underset{X}{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}} *^{T} \underset{F}{\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}} = \underset{Y}{\begin{bmatrix} 1 & 0 & 2 & 0 \\ 0 & -1 & 0 & -2 \\ 3 & 0 & 4 & 0 \\ 0 & -3 & 0 & -4 \end{bmatrix}}$$

stride = 2

padding = 0

output_padding = 0

$K = 2$

1) Вставляем нули с шагом (stride − 1) – dilation

$$\hat{X} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 3 & 0 & 4 \end{bmatrix}$$

2) Применяем padding к $\hat{X}$ размером output_padding

$$\tilde{\hat{X}} = \hat{X}$$

3) Разворачиваем ядро F на 180° – flip(−1, −2)

$$\tilde{F} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

4) Свертка со $\widetilde{stride} = 1$

$$\widetilde{padding} = \overset{2}{K} - 1 - \overset{0}{padding} + \overset{0}{output\_padding} = 1 \geq 0$$

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

Обратите внимание, что $Y * F = X \cdot \|F\|^2$

Почему эта свёртка называется транспонированной?

Рассмотрим след. пример

$$X = \begin{pmatrix} 1 & 2 & -3 & -2 \\ 5 & 1 & 4 & -4 \\ 0 & -2 & 0 & -5 \\ 3 & 0 & -1 & 1 \end{pmatrix}$$

X.shape

$$F = \begin{pmatrix} 2 & 1 & 4 \\ 0 & 3 & -5 \\ -3 & 1 & -2 \end{pmatrix} \longrightarrow \hat{F} = \left(\begin{array}{cccc|cccc|cccc|cccc} 2 & 1 & 4 & & 0 & 3 & -5 & & -3 & 1 & -2 & & & & & \\ & 2 & 1 & 4 & & 0 & 3 & -5 & & -3 & 1 & -2 & & & & \\ & & & & 2 & 1 & 4 & & 0 & 3 & -5 & & -3 & 1 & -2 & \\ & & & & & 2 & 1 & 4 & & 0 & 3 & -5 & & -3 & 1 & -2 \end{array}\right)$$

params

$$Y = X * F = \begin{pmatrix} -27 & 41 \\ 14 & 12 \end{pmatrix}$$ — stride = 1, padding = 0

$$Z = Y *^{T} F = \begin{pmatrix} -54 & 55 & -67 & 164 \\ 28 & -43 & 326 & -157 \\ 81 & -108 & 61 & -142 \\ -42 & -22 & -16 & -24 \end{pmatrix}$$ — stride = 1, padding = 0, output_p_ = 0

Можно убедиться, что $\hat{F}$ @ X.flatten() == Y.flatten()

$\hat{F}^T$ @ Y.flatten() == Z.flatten()