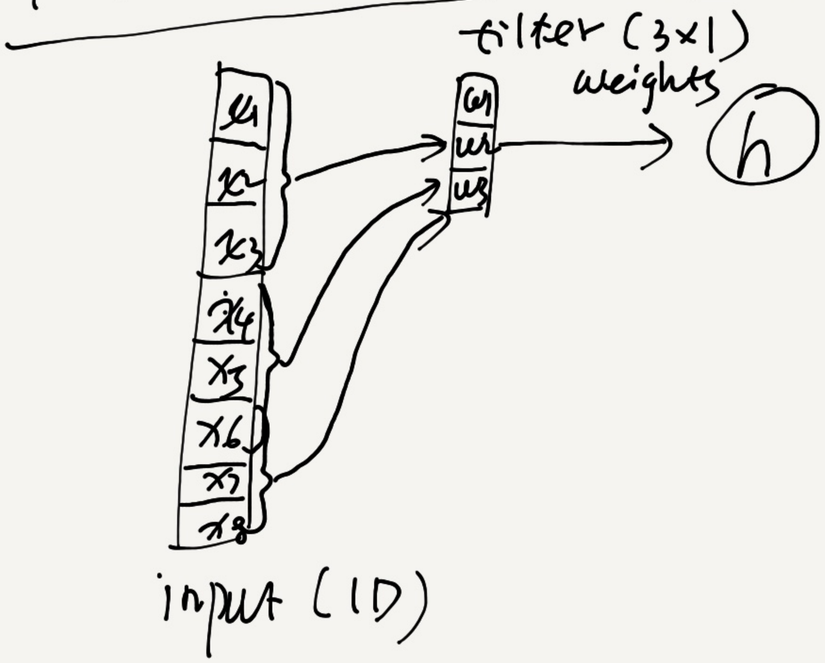
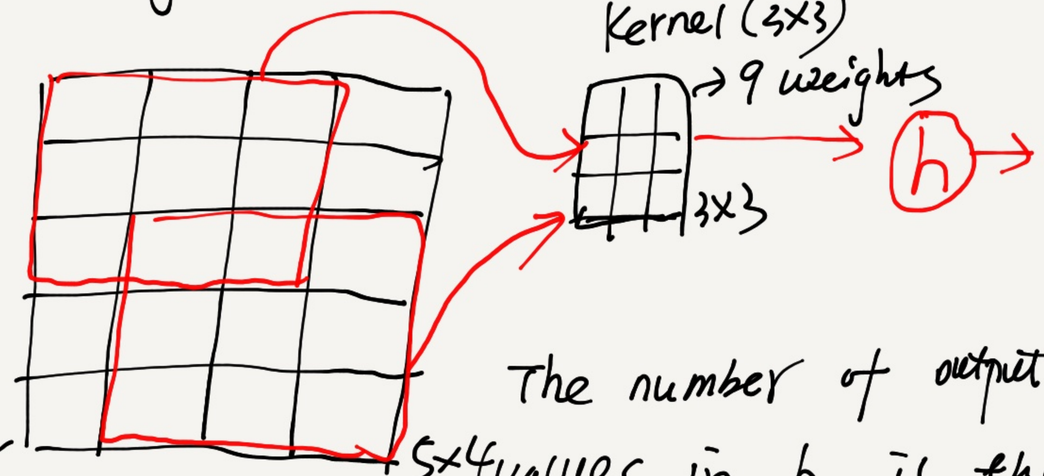
CNN Operations (Leeture 19)

1. CNNs reduce the number weights by sharing weights spatially and perform calculation locally. (in the sliding window).

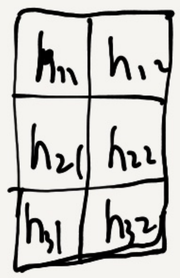




The number of output is the # of sliding windows.

↳ how many sliding windows?

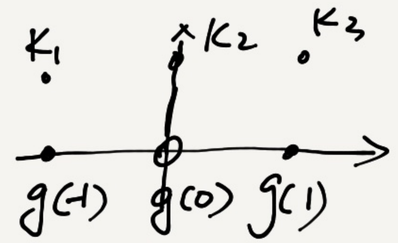
step size = 1. → 6 windows.

h →  2×3.

2. Convolution operations in CNNs.

① Convolution : multiplication between two sequences of signals.

↳ signal processing.



$f(i)$: $x_1, x_2, \cdots, x_{10}$

$g(i)$: K_1, K_2, K_3

$f * g(i) = \sum_{u=-S}^{S} f(i-u) \cdot g(u)$. (* → convolution)

↳ window.

$u \in [-S, S]$

$f * g(1) = \sum_{u=-S}^{S} f(1-u) \cdot g(u)$

f: $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}$

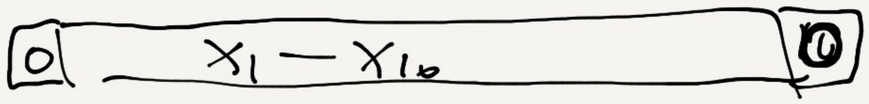
g: K_1, K_2, K_3

$f*g$: $y_2 = x_1K_1 + x_2K_2 + x_3K_3$, $y_3 = x_2K_1 + x_3K_2 + x_4K_3$, ⋯ $y_9 = x_8K_1 + x_9K_2 + x_{10}K_3$

$y_2, y_3, y_4, \cdots, y_8, y_9$ → 8 values

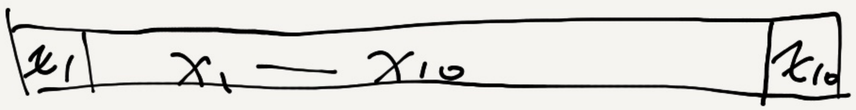
2) Keep the size of features.

Zero padding.



12 values

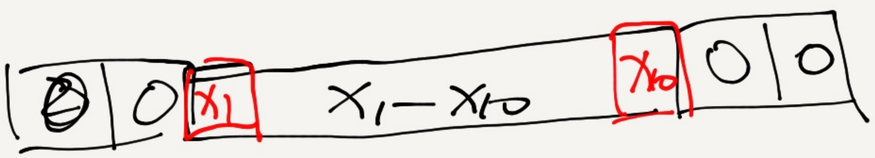
Reflection padding:



12 values.

Kernel size: 3 (weights) $\longrightarrow \left\lfloor \frac{3}{2} \right\rfloor$ at the begining and and at the end.

5 (weights) $\longrightarrow$



$\left\lfloor \frac{5}{2} \right\rfloor$

3) Common filters / Kernels.

① identity filter. [0 | 1 | 0] $\longrightarrow$ $f * g = f$ (with padding)

② averaging filter [1/3 | 1/3 | 1/3] $\longrightarrow$ remove noise / smooth the input features. ③ Gaussian filters

$\frac{1}{N} \cdot$ [1 | 1 | ... | 1] (N weights)

$\frac{1}{4} \cdot$ [1 | 2 | 1]

4) from elementwise multiplication to matrix multiplication

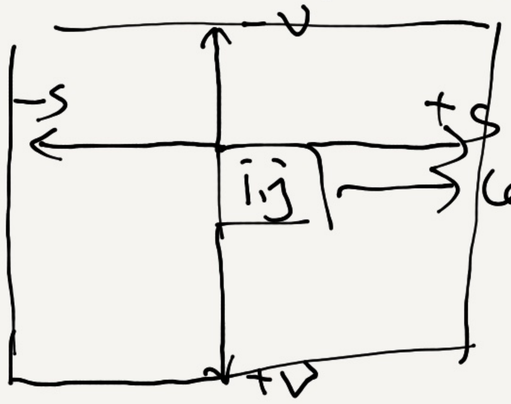
$$y = f * g = K \cdot f$$

$$K = \begin{bmatrix} k_1 & k_2 & k_3 & 0 & 0 \\ 0 & k_1 & k_2 & k_3 & 0 \\ 0 & 0 & k_1 & k_2 & k_3 \end{bmatrix}, \quad f = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}, \quad g = \begin{pmatrix} k_1 \\ k_2 \\ k_3 \end{pmatrix}$$

5). 2D convolutions.

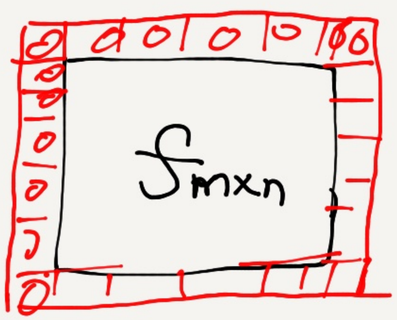
f. g → 2D matrices.

$$f * g(i,j) = \sum_{u=-s}^{s} \sum_{v=-s}^{s} f(i-u, j-v) \cdot g(u,v)$$



center coord. of a sliding window.

padding:

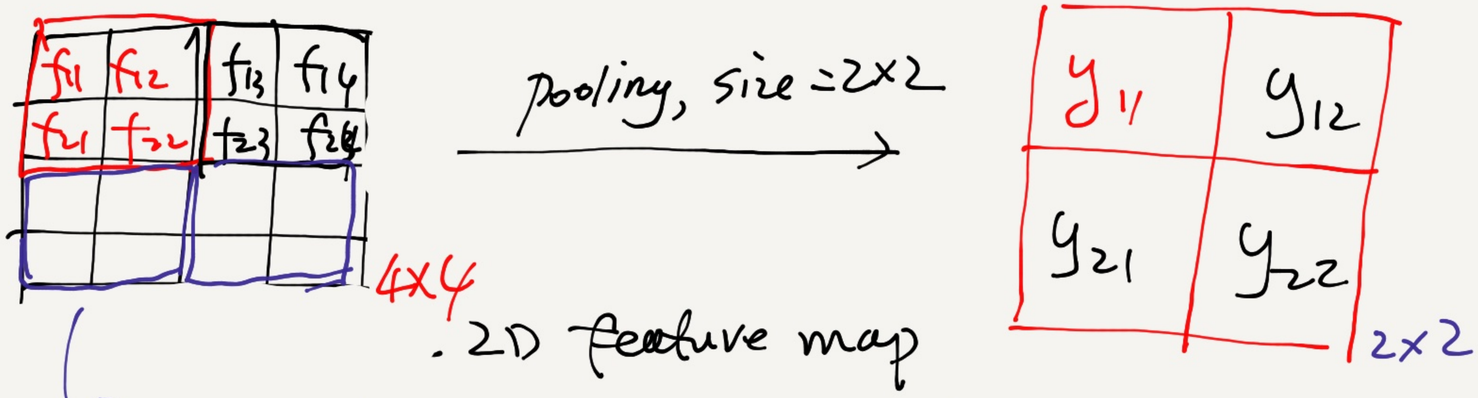


g: 3×3.

# of padding elements:

$(n+2) \times 2 + m \times 2$

3. pooling operations. → downsampling



↳ no overlap among the sliding windows.

max pooling: $y_{11} = \max\{f_{11}, f_{12}, f_{13}, f_{14}\}$

Average pooling. $y_{11} = \frac{1}{4}(f_{11} + f_{12} + f_{13} + f_{14})$

1. reduce the resolution of input feature map

2) focus on more global features.

↳ but local features are also important in some application