

## CW (CNN)

Konwolucja lub splatanie (*ang.* convolution) i podpróbkowanie (*ang.* “pooling”) w CNN  
(Convolutional Neural Network)

Input (konwolucji)  $\mathbf{u} = (u(i, j))_{i=1, j=1}^{N, N}$ ,  $u(i, j) \in \mathbb{R}$  ( $1 \leq i, j \leq N$ )

Output (konwolucji)  $\mathbf{x} = (x(i, j))_{i=1, j=1}^{N, N}$ ,  $x(i, j) \in \mathbb{R}$  ( $1 \leq i, j \leq N$ )  
( $N = 5$ )

Jądro (*ang.* kernel)  $\mathbf{w} = (w(i', j'))_{i'=-H, j'=-H}^{H, H}$   
 $w(i', j') \in \mathbb{R}$  ( $-H \leq i', j' \leq H$ )  
( $H = 1$ )

Konwolucja (splot)  $\mathbf{u} \mapsto \mathbf{x} = \mathbf{w} * \mathbf{u}$

Konwolucja = ekstrakcja cech obrazów (feature extraction)

$$x(i, j) = f\left(\left\{\sum_{i'=-H}^H \sum_{j'=H}^H w(i', j')u(i+i', j+j')\right\} - \theta\right) \quad (1 \leq i, j \leq N)$$

$f$  funkcja progowa

$$f(x) = \begin{cases} 0 & \text{gdy } x < 0 \\ 1 & \text{gdy } x \geq 0 \end{cases}$$

Uwaga.  $(i+i', j+j') \notin \{1, \dots, N\} \times \{1, \dots, N\} \Rightarrow u(i+i', j+j') = 0$   
(Jak to implementować? Np. rozszerzyć  $(u(i, j))_{i=1, j=1}^{N, N}$  na  $(u(i, j))_{i=0, j=0}^{N+1, N+1}$ .)

“Pooling”  $\mathbf{x} \mapsto \mathbf{y}$

Pooling (łączenie) = operacja próbkowania przestrzennego

Input (“pooling”)  $\mathbf{x} = (x(i, j))_{i=1, j=1}^{N, N}$ ,  $x(i, j) \in \mathbb{R}$  ( $1 \leq i, j \leq N$ )  
( $N = 5$ )

Output (“pooling”)  $\mathbf{y} = (y(i, j))_{i=1, j=1}^{N, N}$ ,  $y(i, j) \in \mathbb{R}$  ( $1 \leq i, j \leq N$ )

$$y(i, j) = f\left(\left\{\frac{1}{(2H+1)^2} \sum_{(i', j') \in P} x(i+i', j+j')\right\} - \theta'\right) \quad (1 \leq i, j \leq N)$$

$$P = \{-H, H\} \times \{-H, H\}$$

$$(H = 1)$$

$f$  funkcja progowa

$$f(x) = \begin{cases} 0 & \text{gdy } x < 0 \\ 1 & \text{gdy } x \geq 0 \end{cases}$$

Uwaga.  $x(i+i', j+j') \notin \{1, \dots, N\} \times \{1, \dots, N\} \Rightarrow x(i+i', j+j') = 0$   
(Jak to implementować? Np. rozszerzyć  $(x(i, j))_{i=1, j=1}^{N, N}$  na  $(x(i, j))_{i=0, j=0}^{N+1, N+1}$ .)

**Dane dla zadania (1)**

$$\mathbf{u}_i = \begin{pmatrix} u(1, 1) & u(1, 2) & u(1, 3) & u(1, 4) & u(1, 5) \\ u(2, 1) & u(2, 2) & u(2, 3) & u(2, 4) & u(2, 5) \\ u(3, 1) & u(3, 2) & u(3, 3) & u(3, 4) & u(3, 5) \\ u(4, 1) & u(4, 2) & u(4, 3) & u(4, 4) & u(4, 5) \\ u(5, 1) & u(5, 2) & u(5, 3) & u(5, 4) & u(5, 5) \end{pmatrix} \quad (1 \leq i \leq 5)$$

■ = 1 ∈ ℝ (float)

0 ∈ ℝ (float)

$$\mathbf{u}_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \blacksquare & \blacksquare & \blacksquare & 0 \\ 0 & \blacksquare & 0 & \blacksquare & 0 \\ 0 & \blacksquare & \blacksquare & \blacksquare & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{u}_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \blacksquare & \blacksquare & \blacksquare & 0 & 0 \\ \blacksquare & 0 & \blacksquare & 0 & 0 \\ \blacksquare & \blacksquare & \blacksquare & 0 & 0 \end{pmatrix} \quad \mathbf{u}_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \blacksquare & \blacksquare & \blacksquare \\ 0 & 0 & \blacksquare & 0 & \blacksquare \\ 0 & 0 & \blacksquare & \blacksquare & \blacksquare \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{u}_4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \blacksquare & \blacksquare & 0 & 0 \\ 0 & 0 & \blacksquare & 0 & 0 \\ 0 & 0 & \blacksquare & 0 & 0 \\ 0 & 0 & \blacksquare & 0 & 0 \end{pmatrix} \quad \mathbf{u}_5 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \blacksquare & \blacksquare & 0 & 0 & 0 \\ 0 & \blacksquare & 0 & 0 & 0 \\ 0 & \blacksquare & 0 & 0 & 0 \\ 0 & \blacksquare & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{w} = \begin{pmatrix} w(-1, -1) & w(-1, 0) & w(-1, 1) \\ w(0, -1) & w(0, 0) & w(0, 1) \\ w(1, -1) & w(1, 0) & w(1, 1) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

$\theta = 2.5$

## Zadania

**Zadanie (1)** (Konwolucja) Obliczyć i wyświetlić jako obraz  $\mathbf{w} * \mathbf{u}_i$  ( $1 \leq i \leq 5$ ).

## Dane dla zadania (2)

$$\mathbf{x}_i = \begin{pmatrix} x(1, 1) & x(1, 2) & x(1, 3) & x(1, 4) & x(1, 5) \\ x(2, 1) & x(2, 2) & x(2, 3) & x(2, 4) & x(2, 5) \\ x(3, 1) & x(3, 2) & x(3, 3) & x(3, 4) & x(3, 5) \\ x(4, 1) & x(4, 2) & x(4, 3) & x(4, 4) & x(4, 5) \\ x(5, 1) & x(5, 2) & x(5, 3) & x(5, 4) & x(5, 5) \end{pmatrix} \quad (1 \leq i \leq 2)$$

■ = 1 ∈ ℝ (float)

0 ∈ ℝ (float)

$$\mathbf{x}_1 = \begin{pmatrix} 0 & \blacksquare & 0 & 0 & 0 \\ 0 & 0 & \blacksquare & 0 & 0 \\ 0 & \blacksquare & 0 & 0 & 0 \\ 0 & 0 & \blacksquare & 0 & 0 \\ 0 & \blacksquare & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{x}_2 = \begin{pmatrix} 0 & 0 & 0 & \blacksquare & 0 \\ 0 & 0 & 0 & 0 & \blacksquare \\ 0 & 0 & 0 & \blacksquare & 0 \\ 0 & 0 & 0 & 0 & \blacksquare \\ 0 & 0 & 0 & \blacksquare & 0 \end{pmatrix}$$

$$\theta' = \frac{2.5}{9}$$

**Zadanie (2)** (“pooling”) Obliczyć i wyświetlić jako obraz “pooling”  $\mathbf{y}$  dla  $\mathbf{x}_i$  ( $1 \leq i \leq 2$ ).

Convolution przeprowadza ekstrakcję cech obrazów (feature extraction), natomiast Pooling (łączenie) przeprowadza operację próbkowania przestrzennego.

*Notacja* (propozycja)  $i \rightsquigarrow i, j \rightsquigarrow j, i' \rightsquigarrow p, j' \rightsquigarrow q$

$\mathbf{u}_k \rightsquigarrow u\_k[i][j], \mathbf{w} \rightsquigarrow w[p][q], \mathbf{x}_k \rightsquigarrow x\_k[i][j], \mathbf{y} \rightsquigarrow y[i][j]$

$\theta \rightsquigarrow theta\_1, \theta' \rightsquigarrow theta\_2$