5. Uczenie głębokie (Deep learning) – Dalszy ciąg

Chciałbym omówić drugą metodę (CNN = konwolucja + Pooling), która wzmacnia propagację wsteczną (uczenie głębokie).

Kunihiko Fukushima (1936–)
Neocognitron (1980), the original deep convolutional neural network (CNN) architecture
K. Fukushima, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biological Cybernetics* **36** (1980), 193–202.
W pracy Fukushimy:
Convolution $\longrightarrow$ S-cells (S=Simple)
Pooling $\longrightarrow$ C-cells (C=Complex)
Learning by back propagation $\longrightarrow$ Add if-Silent (AiS)

5.2. CNN (= **Convolutional Neural Network**, nie CNN, BBC, Sky News, ...)

Splotowe sieci neuronowe, konwolucyjne sieci neuronowe

Konwolucja i Pooling są używane i zastosowane w przetwarzaniu obrazów. Dlatego potraktojemy sygnały określone na dwu wymiarowych indeksach:

$$\Omega = \{1, \dots, n\} \times \{1, \dots, n\} = \{(i,j) | i,j = 1, \dots, n\}.$$

Możemy rozumieć, że (i,j) oznacza piksel albo jeden neuron.

Convolution (splatanie, splot, konwolucja) w jednym wymiarze to przekształcenie $(u_i)_{i=-\infty}^{\infty} \mapsto (y_i)_{i=-\infty}^{\infty}$ określone wzorem

$$y_i = \sum_{i'=-\infty}^{\infty} w_{i'} u_{i-i'}.$$

Wersja dwu wymiarowa tego splotu jest

$$y_{(i,j)} = \sum_{i'=-\infty}^{\infty} \sum_{j'=-\infty}^{\infty} w_{(i',j')} u_{(i-i',j-j')}$$

dla przekształcenia

$$(u_{(i,j)})_{i,j=-\infty}^{\infty} \mapsto (y_{(i,j)})_{i,j=-\infty}^{\infty}.$$

Żeby zastosować CNN, rozważmy sygnały na obszarze

$$\Omega = \{1, \dots, n\} \times \{1, \dots, n\} = \{(i,j) | i,j = 1, \dots, n\}.$$

Weźmiemy zbiór nosinków wag $w_{(i,j)}$ następująco:

$$I = \{(i,j) | i,j = -m, -(m-1), \dots, -1, 0, 1, \dots, m-1, m\},$$

przy czym $m \ll n$.

Przekształcenie konwolucji zdefiniujemy wzorem

$$y_{(i,j)} = f\Big(\sum_{(i',j') \in I} w_{(i',j')} u_{(i-i',j-j')} \Big)$$

dla każdego $(i,j) \in \Omega$. Jeśli $(i-i', j-j') \notin \Omega$ dla $(i,j) \in \Omega$ i $(i',j') \in I$, to rozumiemy $u_{(i-i',j-j')} = 0$. Następnie w CNN stosujemy "Pooling"po konwolucji. Są dwie możliwości.
Average pooling

$$x_{(i,j)} = \sum_{(i',j')\in I} \frac{1}{|I|} y_{(i-i',j-j')}$$

Max pooling

$$x_{(i,j)} = \max_{(i',j')\in I} y_{(i-i',j-j')}$$

W sumie w całej strukturze CNN mamy

Convoluttion $\to$ Pooling $\to$ Convoluttion $\to$ Pooling $\to \cdots \to$ Convoluttion $\to$ Pooling.

Convolution przeprowadza ekstrakcję cech obrazów (feature extraction), natomiast Pooling (łączenie) przeprowadza operację próbkowania przestrzennego. Do tej całej sytuacji stosujemy metodę gradientu dla propagacji wstecznej.