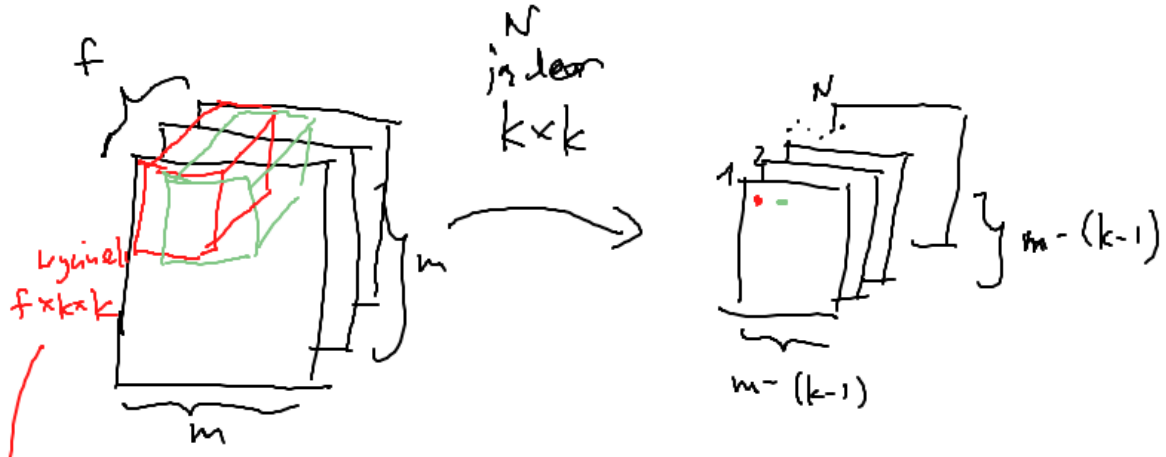


jądra k×k mają w istocie wielkość f×k×k

mnożenie po współrzędnych : suma

Liczba f. aktywacji

wyjście

wag mamy $N \cdot \underbrace{k \cdot k \cdot f}_{\text{w jednym jądrze}}$

w naiwnym algorytmie mamy:

$$k \cdot k \cdot f \cdot (m-(k-1))^2 \cdot N$$

mnożeń

najczęściej ReLU

$$f(x) = x_+ = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases}$$

Dropout – tylko w fazie treningu

→ wejście tego neuronu w fazie testowania to:

$$= \sum_{i=1}^{5} w_i y_i = \sum_{i=1}^{3} w_i y_i + \sum_{i=4}^{5} w_i y_i =$$

$$= \left(\frac{3}{5}\left(\frac{5}{3}\sum_{i=1}^{3} w_i y_i\right) + \frac{2}{5}\left(\frac{5}{2}\sum_{i=4}^{5} w_i y_i\right)\right)$$

to było sumowane w 1. kroku uczenia

...w 2. kroku

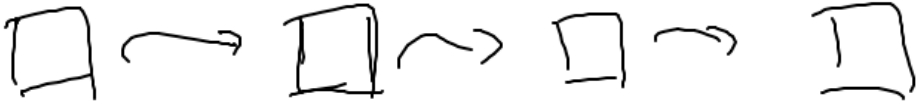
w fazie uczenia wyłączamy niektóre z neuronów w pewnej warstwie, np. zaznaczone dwa. Aby to skompensować, mnożymy wyjście z pozostałych neuronów przez odpowiedni współczynnik, w powyższym przykładzie $\frac{5}{3}$

Powiedzmy, że w kolejnym kroku wyłączymy inne 3 neurony., wyjście pozostałych mnożymy przez $\frac{5}{2}$.

W fazie testowania nie wyłączamy żadnych neuronów.

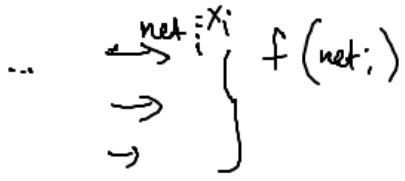
Zwykle wyłącza się 25-50% neuronów.

warstwy są zwykle tworzone tak, aby koszt obliczeń związany z poszczególnymi warstwami był podobny



wiek: 99% 0.5% 0.5%

Softmax



$$y_i = f(x_i) = \frac{e^{x_i}}{\sum_{i=1}^{N} e^{x_i}}$$

$$f \geq 0, \quad \sum_{i=1}^{N} f(x_i) = 1$$

sigmoid

kanoniczna funkcja kosztu

$$L = \frac{1}{2} \sum_{i=1}^{N} (y_i - t_i)^2$$

funkcja kosztu: cross-entropy

$$L(y, t) = -\sum_{i=1}^{N} t_i \log(y_i) = -\log y_j \text{, gdzie } j: t_j = 1$$

wyjście

etykieta (oczekiwane wyjście)

$t = (0, \dots, 1, 0, \dots)$