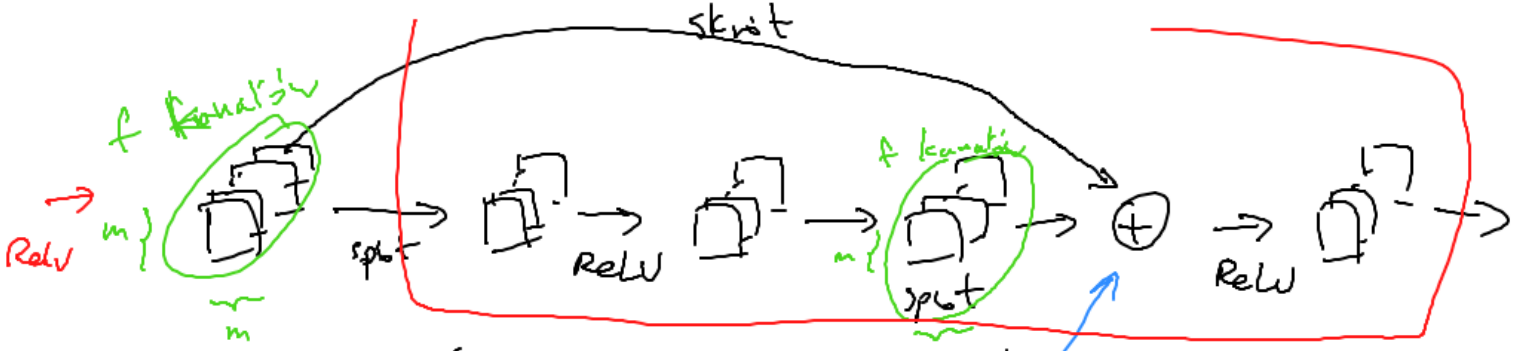
ResNet (residual neural network)



Jeśli dodawanie do wyjściowych było wykonane bez skrótowych połączeń, to wyjściowe warstwy rozpoczynają na zielono muszą być identyczne (Można stosować tę metodę również gdy zielone warstwy nie są identyczne, ale wtedy potrzeba coś dodać).

Pomysł polega na tym, że gdyby wagi w spłotach w części czerwonej były małe, to część czerwona jest (prawie) identycznością.

Łączenie skrótów zmniejsza problem zanikających gradientów.

| Sequential

Optymalizacja:

. SGD (stochastic gradient descent) — to jest to, o co było

jak w regresji gradientowej

metoda opisana dla sieci neuronowych (ogólnych)
(on-line lub batch)

. coby różnić się SGD, to zwykle stopniowo zmniejsza się stałą uczenia

. często różnią się dodatkowo metody „bezwładności" (momentum):

jeśli w i-tym kroku aktualizacyjny pewną wagę w o Δw_i, to

w kroku $i+1$:

$$w := w - \eta \cdot \frac{\partial L}{\partial w} + \alpha \cdot \Delta w_i$$

η — stała uczenia

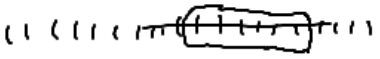
α — stała bezwładności $\in [0,1)$
często $\alpha = 0.9$

- adaptacyjne : szybkość uczenia zależy od kolejnych wagi

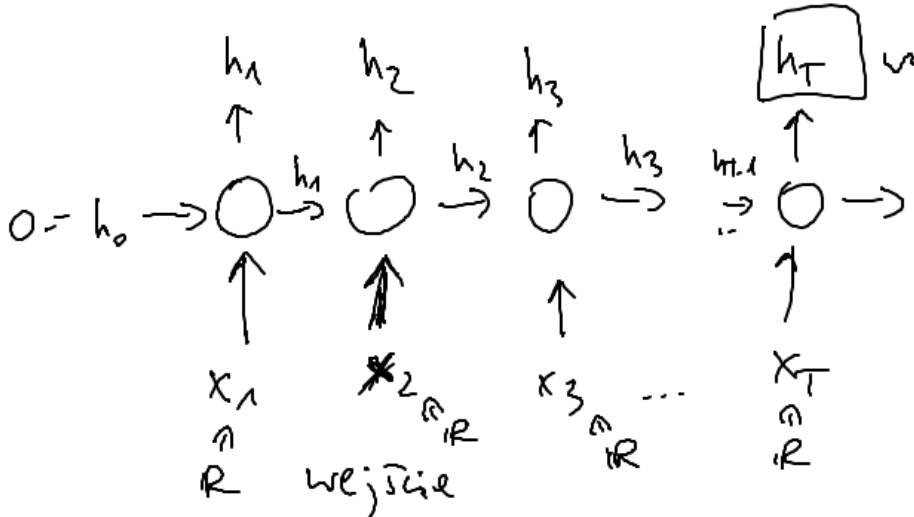
AdaGrad, Ada, RMSprop, Adam

wiele znanych rozwiązań mimo dobrych wyników SGD (+momentum)

Rekurencyjne sieci neuronowe (RNN)



zaczniemy od prostego przypadku pojedynczego wejścia, warstwy, komórki



wagi (GIR), które dopasowujemy

prosta komórka RNN: $h_t = \sigma\left(W \cdot x_t + U \cdot h_{t-1} + b\right)$, $t = 1, 2, 3, \ldots$

gdzie
- $W \cdot x_t$ – wejście
- $U \cdot h_{t-1}$ – poprzednie wyjście („historia")

σ – funkcja aktywacji, np. $\sigma(x) = \frac{1}{1+e^{-x}}$

Zastosowania:

- rozpoznawanie wskazanego słowa przy wpisywaniu SMS

Dzień w

dobry

- modelowanie sieregów czasowych

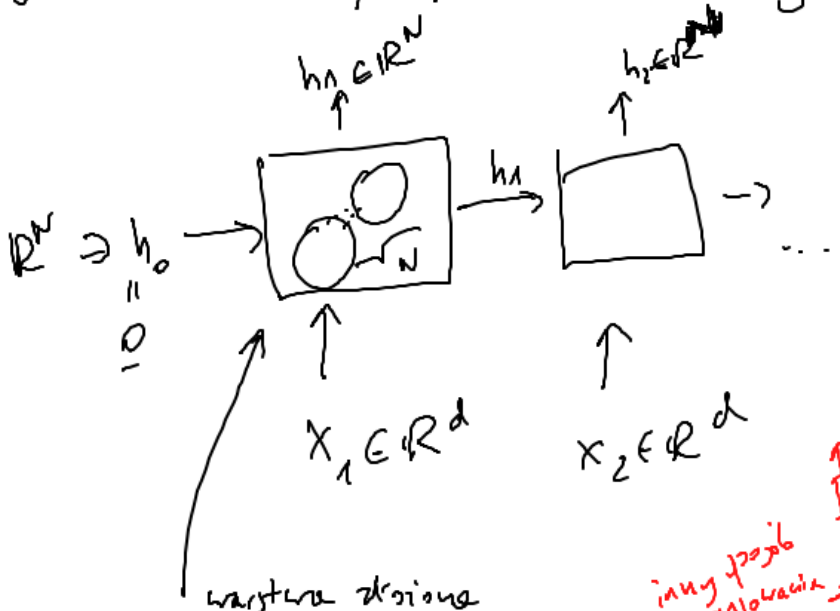
- przetwarzanie języka naturalnego

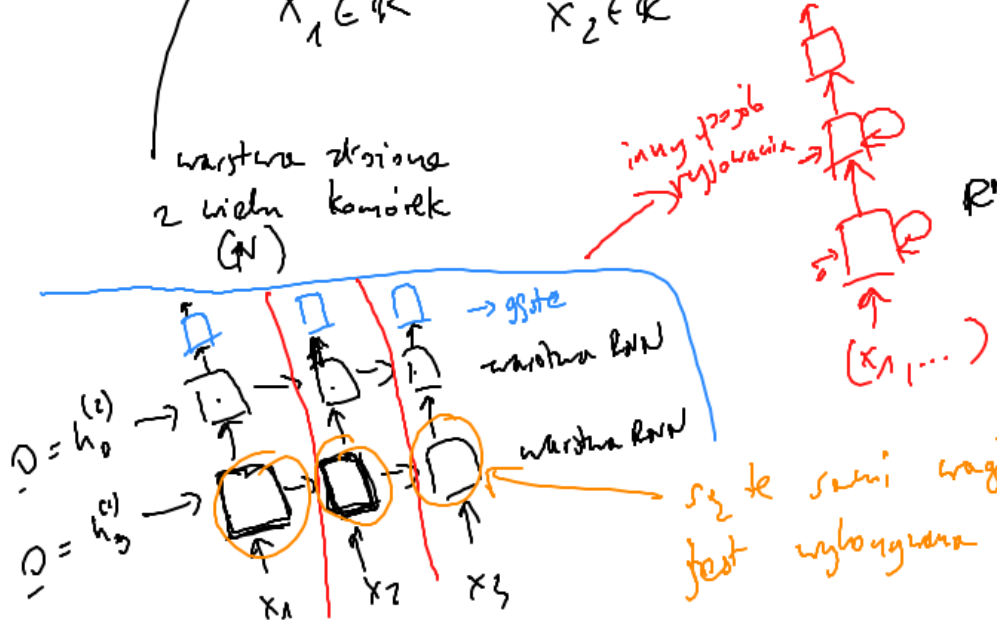
→0→ ..→0 ⊢ ... wejście: zdanie po polsku

wejście: zdanie po angielsku

Uogólnienie na przypadek wielowymiarowy:

Liczba parametrów: $Nd + N^2 + N$





są te same wagi!
Jest wykonywana operacja tego samego typu, ale dla innych wejść

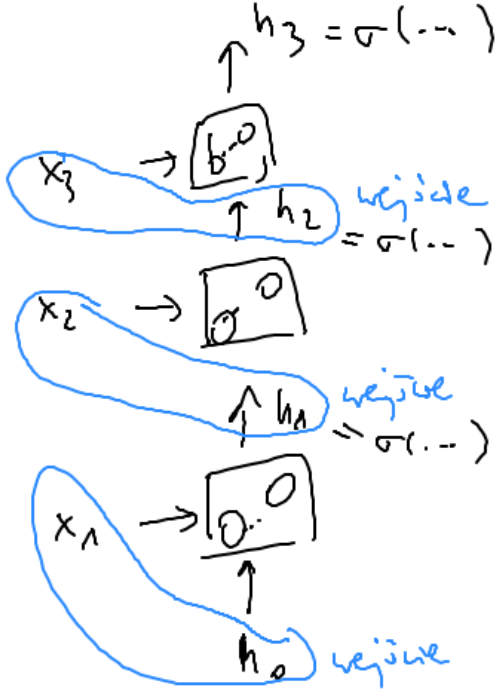
wagi, które są dopasowywane w procesie uczenia

(działa na współrzędnych)

$$h_t = \sigma\left(W \cdot x_t + U \cdot h_{t-1} + b\right) = \begin{bmatrix} \sigma(\dots) \\ \sigma(\dots) \\ \vdots \\ \sigma(\dots) \end{bmatrix} \in \mathbb{R}^N$$

$h_t \in \mathbb{R}^N$, W: $N \times d$, x_t: $d \times 1$, U: $N \times N$, $h_{t-1} \in \mathbb{R}^N$, $b \in \mathbb{R}^N$

pojednicza warstwa RNN dla T=3



LSTM (Long - short term memory)

95 - 97 - 99 Hochreiter, Schmidhuber

warstwa z N komórek

parametrów jest $4(Nd + N^2 + N)$

$$f_t = \sigma_g\left(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f\right) \in [0,1]^N \quad \text{forget gate}$$

$$i_t = \sigma_g\left(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i\right) \in [0,1]^N \quad \text{input gate}$$

$$o_t = \sigma_g\left(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o\right) \in [0,1]^N \quad \text{output gate}$$

$$\tilde{c}_t = \sigma_c\left(W_c \cdot x_t + U_c h_{t-1} + b_c\right)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \in \mathbb{R}^N \leftarrow \text{memory cell}$$

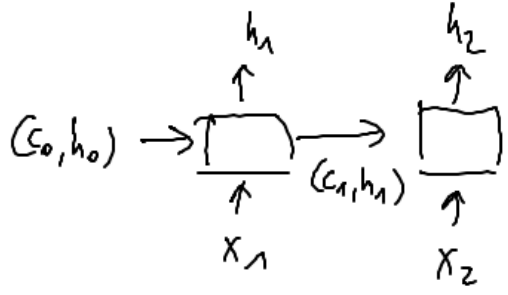
$$h_t = o_t \odot \sigma_h(c_t) \leftarrow \text{wyjście i kolejny ukryty stan}$$

$\mathbb{R}^N$ po wyjściu poprzednich; $\mathbb{R}^N$ mnożenie po wyjściu poprzednich

$$(a_1, \dots, a_N) \odot (b_1, \dots, b_N) = (a_1 b_1, a_2 b_2, \dots, a_N b_N)$$

$$\sigma_g(x) = \frac{1}{1+e^{-x}}$$

$$\sigma_c(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



Dalsza modyfikacja: GRU (gated recurrent unit)

prosta wersja jedna bramka mniej niż w LSTM

ale zwykle podobne działanie