

# Sieci rekurencyjne

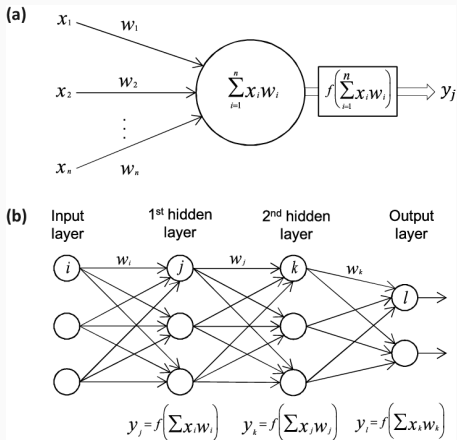
---

mgr inż. Dawid Wiśniewski - Przetwarzanie języka naturalnego

13 April 2021

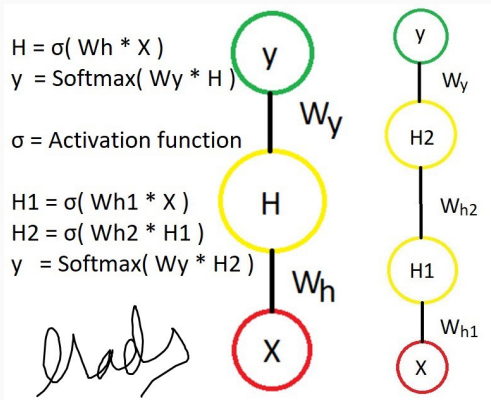
# Sieci rekurencyjne - RNN i

W poprzednim odcinku, sieci typu feedforward:



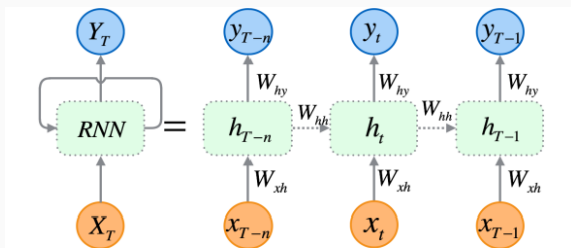
# Sieci rekurencyjne - RNN ii

Graficzna, uproszczona wizualizacja sieci feedforward + wzory



# Sieci rekurencyjne - RNN iii

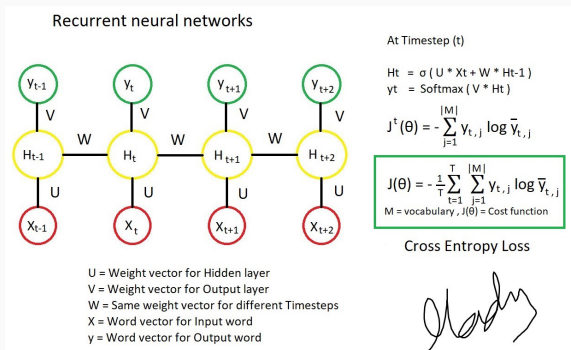
Sieci rekurencyjne - jedno słowo w danym momencie



1 Standard RNN architecture and an unfolded structure with T time

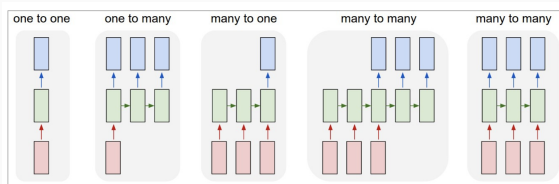
# Sieci rekurencyjne - RNN iv

## Sieci rekurencyjne - wzory



# Sieci rekurencyjne - RNN v

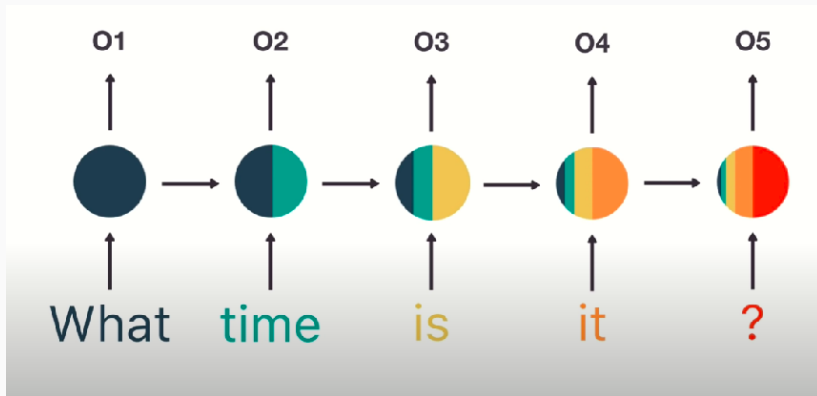
Typy sieci rekurencyjnych:



- one-to-one: klasyczne feedforward (np. dokument -> etykieta)
- one-to-many: generowanie tekstu/muzyki
- many-to-one: wykrywanie sentymentu
- many-to-many: automatyczne tłumaczenie
- many-to-many (synchroniczne): tagowanie części mowy

# Sieci rekurencyjne - RNN vi

Czy cała historia słów rozkłada się po równo?



Kolejne slajdy będą mocno inspirowane”(:) ściągą z:  
<https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>



Funkcja kosztu - analogiczna do klasycznej, ale z uwzględnieniem czasu

□ **Loss function** — In the case of a recurrent neural network, the loss function  $\mathcal{L}$  of all time steps is defined based on the loss at every time step as follows:

$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}(\hat{y}^{<t>}, y^{<t>})$$

Uczenie propagacją wsteczną "przez czas":

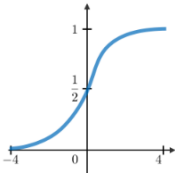
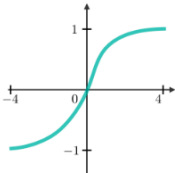
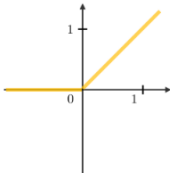
□ **Backpropagation through time** — Backpropagation is done at each point in time. At timestep  $T$ , the derivative of the loss  $\mathcal{L}$  with respect to weight matrix  $W$  is expressed as follows:

$$\frac{\partial \mathcal{L}^{(T)}}{\partial W} = \sum_{t=1}^T \frac{\partial \mathcal{L}^{(t)}}{\partial W} \Big|_{(t)}$$

# Sieci rekurencyjne - RNN x

Popularne funkcje aktywacji w sieciach rekurencyjnych:

□ **Commonly used activation functions** — The most common activation functions used in RNN modules are described below:

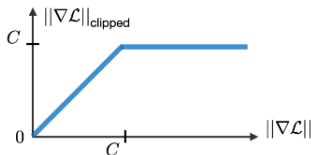
Sigmoid	Tanh	RELU
$g(z) = \frac{1}{1 + e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$
		

Długie sekwencje, długie rozwinięcia sieci, dużo mnożeń w propagacji wstecznej - same kłopoty.

□ **Vanishing/exploding gradient** — The vanishing and exploding gradient phenomena are often encountered in the context of RNNs. The reason why they happen is that it is difficult to capture long term dependencies because of multiplicative gradient that can be exponentially decreasing/increasing with respect to the number of layers.

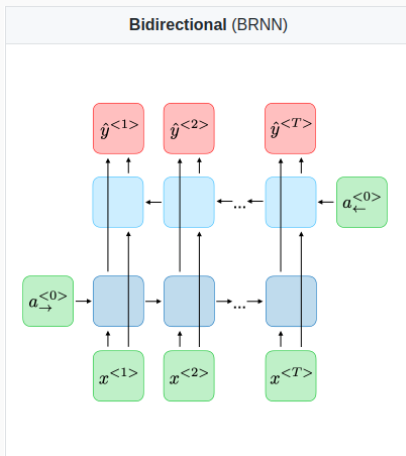
Ucieczka od eksplodującego gradientu:

□ **Gradient clipping** — It is a technique used to cope with the exploding gradient problem sometimes encountered when performing backpropagation. By capping the maximum value for the gradient, this phenomenon is controlled in practice.



## Sieci rekurencyjne - RNN xiii

Bardziej skomplikowane architektury RNN - sieci dwukierunkowe.



# Sieci rekurencyjne - RNN xiv

Bardziej skomplikowane architektury RNN - sieci na stosie.

