

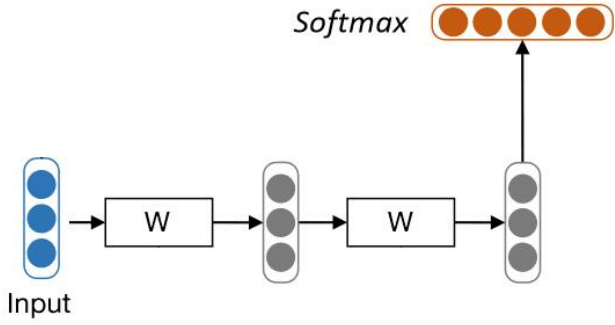
Introduction to deep NLP

Vzorová úloha

Klasifikácia textu - Chceme zaradiť texty do tried, napr. novinové články do kategórií (šport / zahraničie / ...).

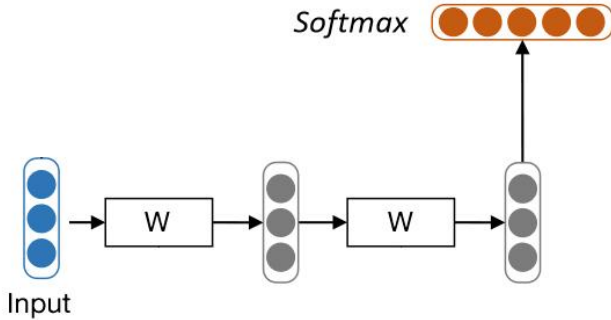
Dataset sú články a o každom vieme do akej kategórie patrí.

Chceme vedieť automaticky zaradovať nové články.



Dopredná neurónová sieť

$$h_1 = \tanh(i \cdot W_1)$$
$$h_2 = \tanh(h_1 \cdot W_2)$$
$$o = \text{softmax}(h_2)$$



Výstup je veľkosti $|C|$, kde C je množina kategórií.

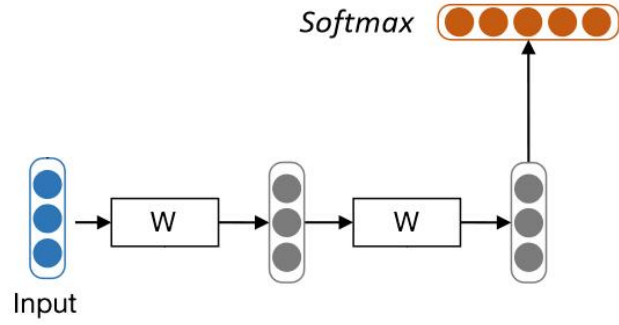
Softmax = Výstup sa dá interpretovať ako pravdepodobnosť.

[0.1; 0.2; 0.7]

10% prvá kategória

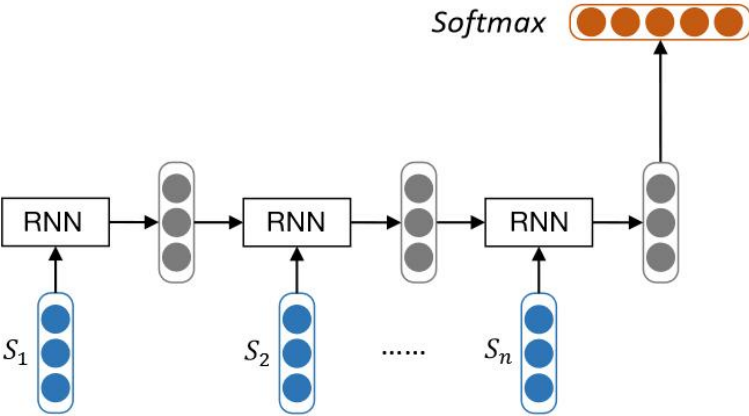
20% druhá kategória

70% tretia



Ako na input vložiť text?

Dá sa text reprezentovať vektorom?



Rekurentná neurónová sieť

Na vstupe máme sekvenciu vektorov - text je sekvencia rozličných jednotiek.

$$h_1 = \tanh(W_i \cdot S_1)$$

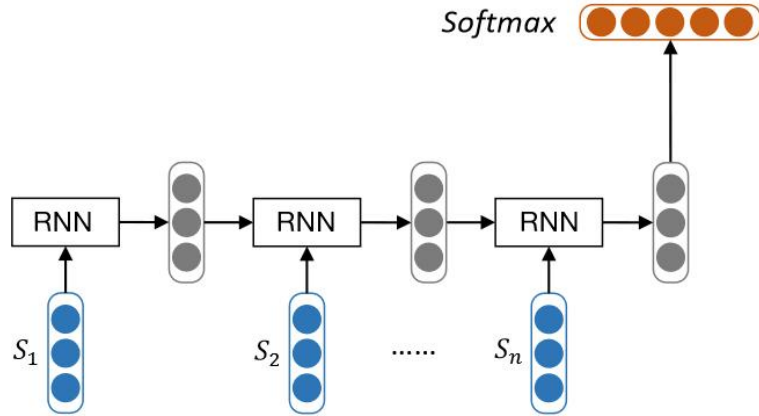
$$h_2 = \tanh(W_i \cdot S_2 + W_h \cdot h_1)$$

$$h_t = \tanh(W_i \cdot S_t + W_h \cdot h_{t-1})$$

$$o = \text{softmax}(h_t)$$

Skrytá vrstva je reprezentáciou všetkých predošlých vstupov.

Posledná vrstva je reprezentáciou celého textu.



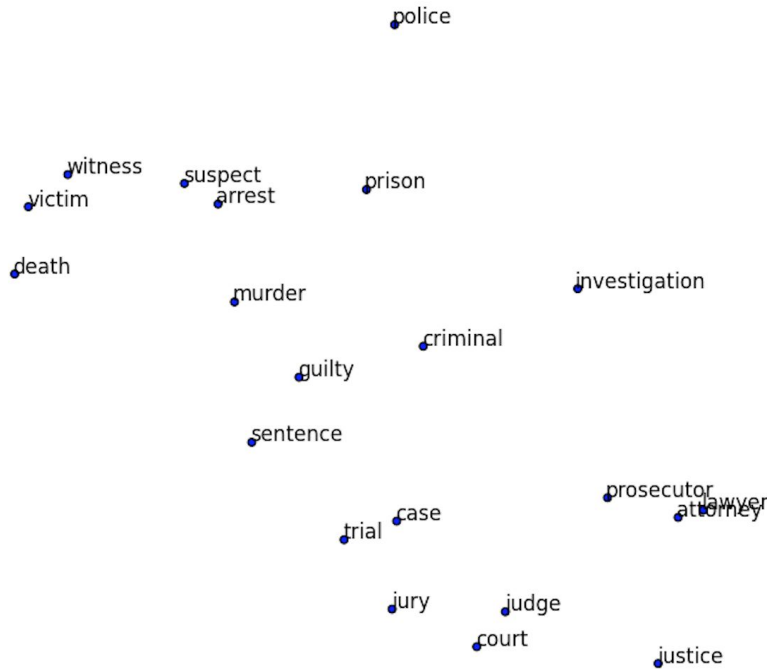
Hlavná otázka: Ako reprezentovať slová?

Používajú sa tzv. *word embeddings*

Iné názvy: word vectors, distributed representations

Modely: **word2vec** (skip-gram, CBOW), GloVe, fastText

Slovensky: vektory latentných črt, distribuované reprezentácie slov



Word embeddings

Slová majú priradené krátke vektory - dĺžka rádovo v stovkách.

Tieto vektory “šifrujú”, aké má slovo sémantiku.

Slová s podobným významom sú blízko pri sebe.

Block 1
The fast cat
wears no hat.

Block 2
The cat in
the hat ran
fast.

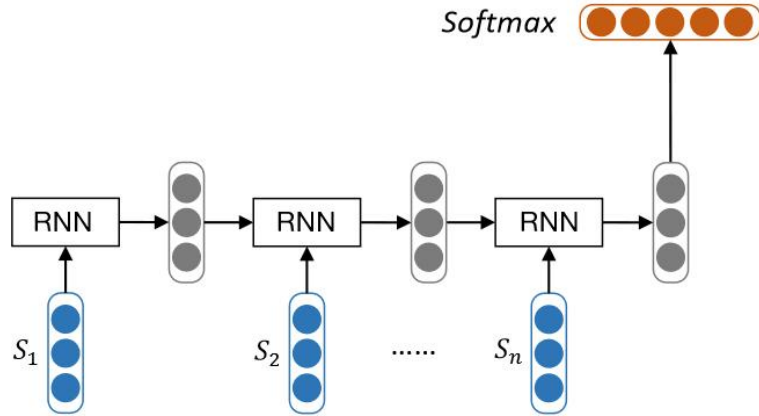


	cat	fast	hat	in	no	ran	the	wears
cat	0							
fast	2	0						
hat	2	2	0					
in	1	1	1	0				
no	1	1	1	0	0			
ran	1	1	1	1	0	0		
the	3	3	3	2	1	2	1	
wears	1	1	1	0	1	0	1	0

Word embeddings

Vznikajú na základe informácie o spoluvýskyte slov.

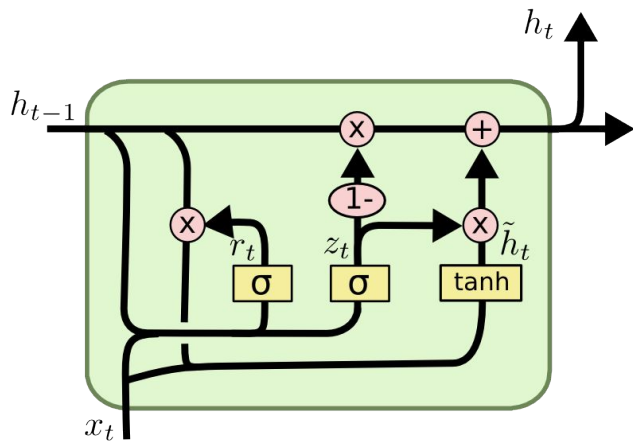
Takáto matica spoluvýskytu je neprakticky veľká - word embeddings sú v podstate jej kompresiou.



Takéto jednoduché RNN zabúdajú.

Riešením je LSTM rekurentná neurónová sieť
(alternatívne GRU, QRNN)

RNN, ktorá si dokáže lepšie pamatáť, čo videla
v minulosti.



LSTM

Riešením je LSTM rekurentná neurónová sieť (alternatívne GRU, QRNN)

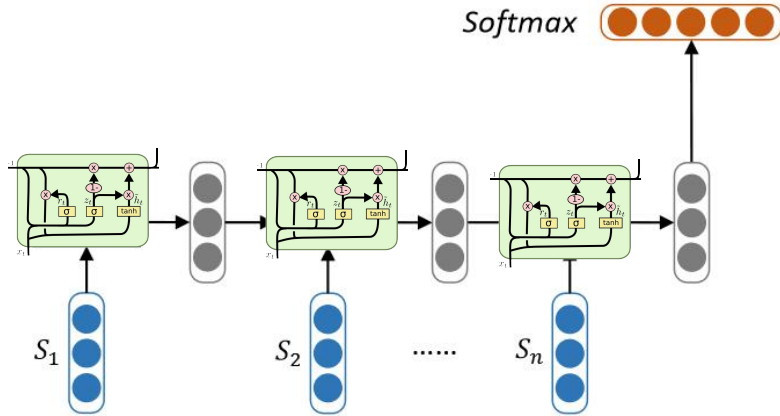
RNN, ktorá si dokáže lepšie pamätať, čo videla v minulosti.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$



LSTM

Riešením je LSTM rekurentná neurónová sieť (alternatívne GRU, QRNN)

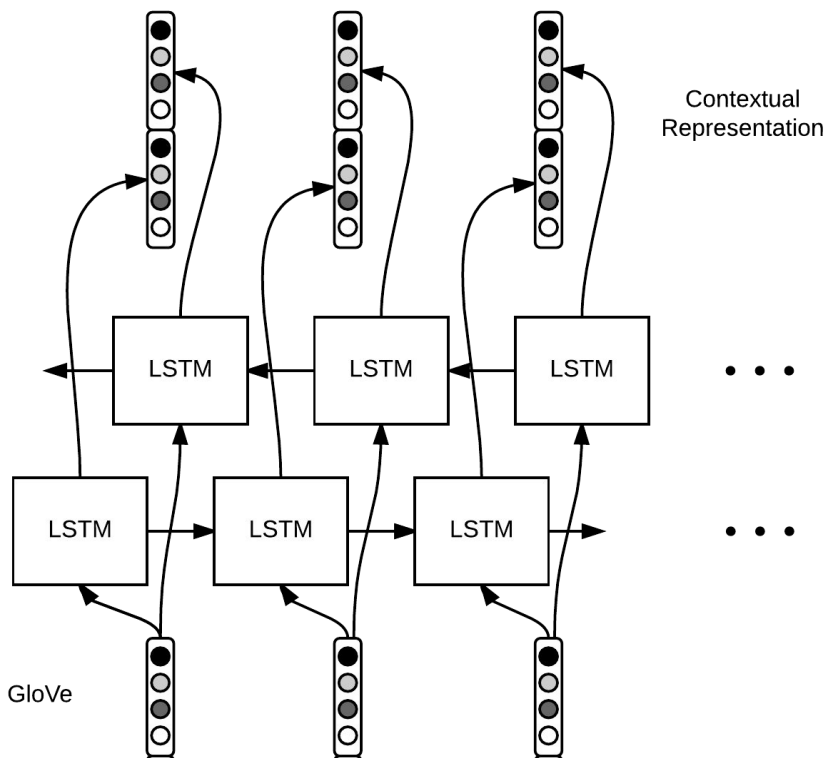
RNN, ktorá si dokáže lepšie pamatáť, čo videla v minulosti.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$



Bi-LSTM

Dve LSTM, jednu spúšťame od prvého slova po posledné, druhú od posledného slova po prvé.

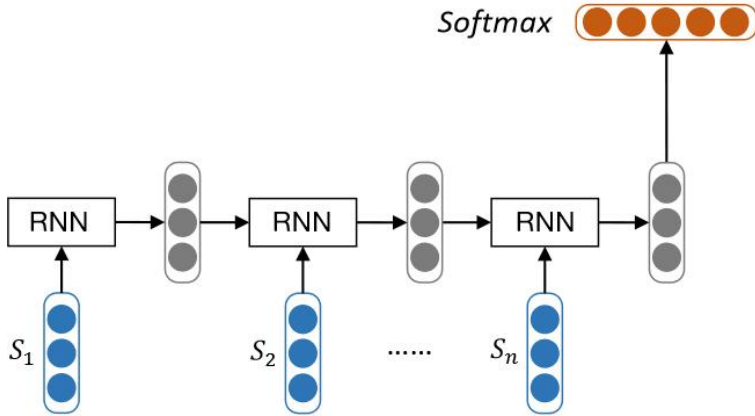
Skryté stavy z oboch spájame.

Zabraňujeme problému so zabúdaním začiatku vety.

Bi-LSTM je de facto NLP štandard

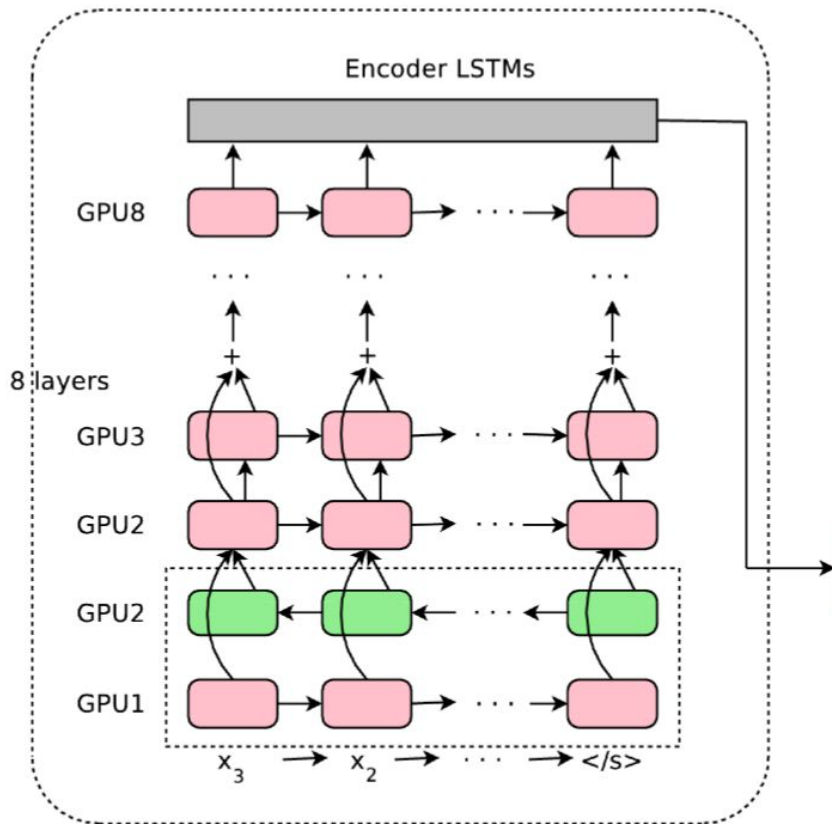
Spoľahlivé

Podpora v NN knižniciach



TensorFlow Tutorial

<https://www.oreilly.com/learning/perform-sentiment-analysis-with-lstms-using-tensorflow>



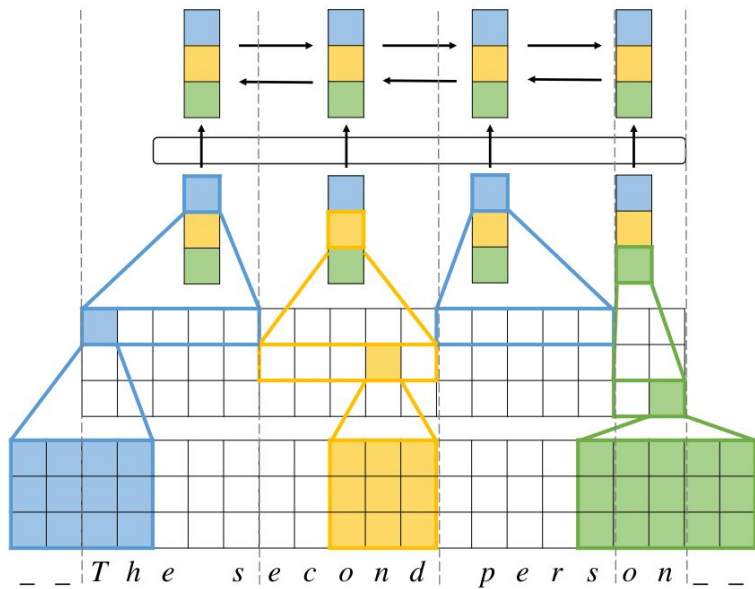
Čo ďalej? Väčšie architektúry

- a) LSTM layers
- b) hierarchické modely

Komplexnejšie modely sú schopné naučiť sa viac.

Potrebujú viac dát a ťažšie sa trénujú.

Wu, Y. (2016) Google's neural machine translation system: Bridging the gap between human and machine translation



Čo ďalej? Práca na úrovni znakov

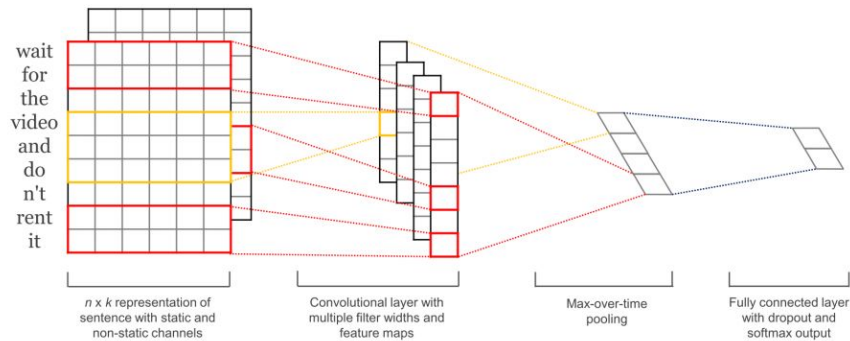
- Skladáme slová zo znakov
- Vôbec neriešime slová

Výrazne menej parametrov.

Veľmi zaujímavé výsledky.

Dokážeme pracovať s menej častými slovami.

Lee, J. et al. (2016). Fully Character-Level Neural Machine Translation without Explicit Segmentation



Čo ďalej? Konvolučná neurónová sieť

Pre niektoré úlohy zaujímavé výsledky.

Rýchlejšia, lepšie sa paralelizuje.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification