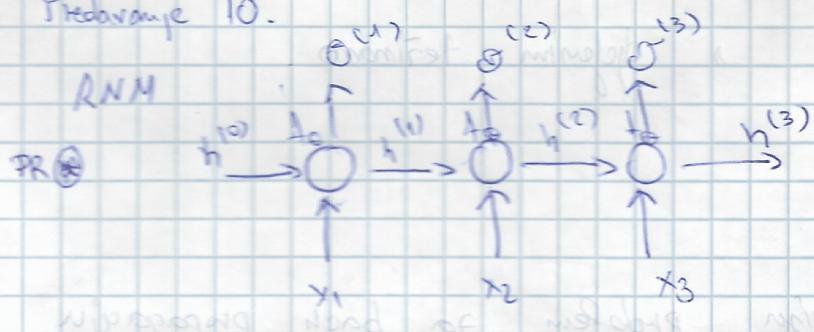


Predavanje 10.

RNM



$$f_0(h^{(+-)}, x^{(\pm)}) = (h^{(+)}, 0^{(+)})$$

Formula za to implementaciju funkcije to

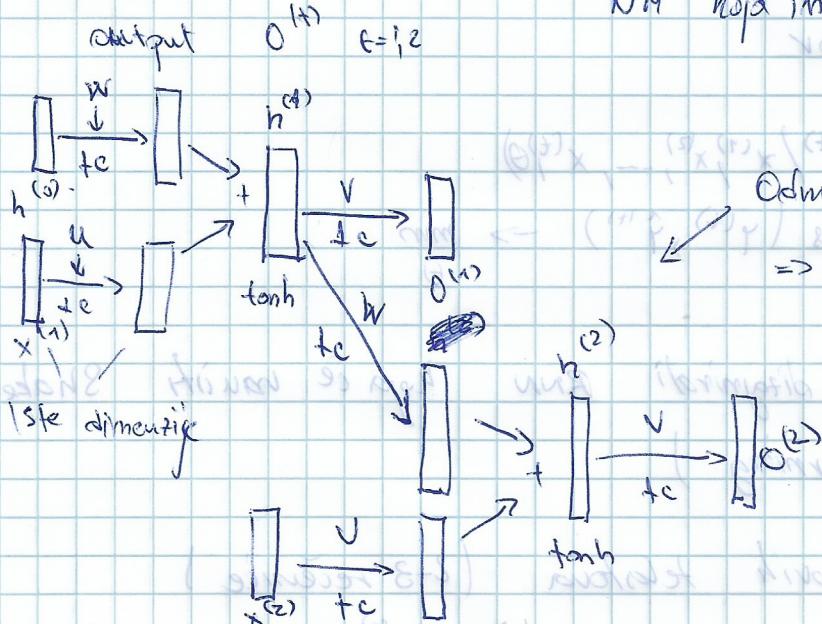
$$h^{(t)} = \tanh \left(W \cdot h^{(t-1)} + U \cdot x^{(t)} \right) \quad \text{sto in parante} \\ O^{(t)} = V \cdot h^{(t)} \quad \text{semplicemente}$$

↳ To \mathbf{w} matrice
 $\mathbf{w}, \mathbf{v}, \mathbf{y}$

RNM je samo neutralna mreža

Primer input $x^{(+)}$ $t=12$

NN koja ima objekt ("ihated") weights



Razumenje građevinarstva nm \rightarrow definicija teorije

$$\frac{\partial y}{\partial w_{ij}} = \frac{\partial y}{\partial b_{ij}^{(1)}} + \frac{\partial y}{\partial v_{ij}^{(1)}}$$

Lemma: Dijeljenje teorije nizom problem za back propagaciju

Implementacija loših funkcija. Rješenje problema kroz klasične

$$(x^{(i)}, y^{(i)}) \quad i=1, \dots, m$$

stavimo $x^{(i)} \in \mathbb{R}^n$ $y^{(i)} \in \{0, 1\} \in \mathbb{R}$
 tada jedan vektor

Predstavljamo da u prijemu tvrdiličavim (\Rightarrow) smez pi MSA

$$\hat{y}(+) = \text{softmax}(o^{(+)})$$

\Rightarrow vjeratnostni vektor

$$\text{loss}(y^{(+)}, \hat{y}(+)) = p(y^{(+)}) / x^{(1)}, x^{(2)}, \dots, x^{(t)}, 0)$$

$$J(\theta) = \sum_{t=1}^T \text{loss}(y^{(+)}, \hat{y}^{(+)}) \rightarrow \min_{\theta}$$

Primer

Problem 1. Želimo dizajnirati RNN koja će transkribovati Shakespearia

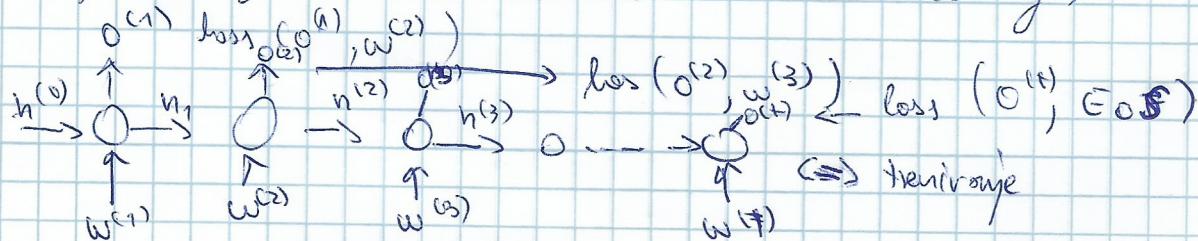
(NN and Deep learning)

Treniranje

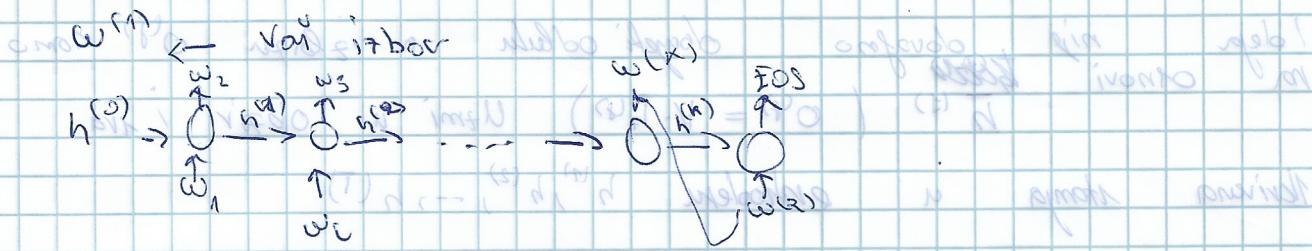
kroz paragrafi Shakespearovih tekstova (2-3 recenice)

Neka je paragraf takođe od narednih ($w^{(1)}, w^{(2)}, \dots, w^{(t)}$)

- napraviti rijeci u vektore (one-hot encoding)



testiranjem RNN generira sljedeća riječ



Problem 2

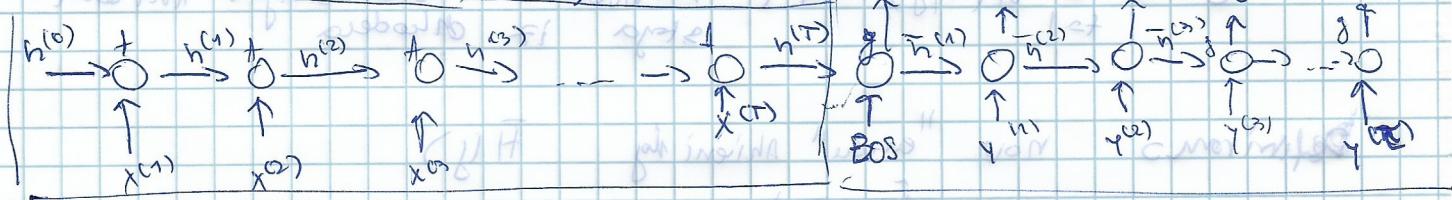
Zelimo AI učitavati koji prevedi HR na ENG i odgovarajući prijevod

Training poohvat rečenica i odgovarajući prijevod

Rečenica na HR $x^{(1)}, x^{(2)}, \dots, x^{(T)}$
Prijevod na ENG $y^{(1)}, y^{(2)}, \dots, y^{(T)}$

Ideja početi prevesti tek kada se čuo u cijelosti rečenicu

RNN model (proces treniranja)



RNN - Encoder

RNN - Decoder

Proces testiranja : za domu HR. Rečenici $x^{(1)}, \dots, x^{(T)}$



Attention mechanism

U problemu 2.

Ideja nije dovoljno
množi se s vektorm $\bar{h}^{(t)}$ ($O^{(t)} = V \cdot \bar{h}^{(t)}$) uzmim obzir i uvaži
iskrivena stanja u entzodenu $h^{(1)}, h^{(2)}, \dots, h^{(T)}$

Implementacija

za ikvieno stanje $\bar{h}^{(t)}$ definisano je "attention" parametarski

$$a(i, t) = \frac{\exp(h^{(i)\top} \cdot \bar{h}^{(t)})}{\sum_{j=1}^J \exp(h^{(j)\top} \cdot \bar{h}^{(t)})} \quad \forall i = 1, \dots, T$$

$(a(1, t), a(2, t), \dots, a(T, t))$ vjerojatnosti vektor

Definirano je kontekstni vektor $\tilde{h}^{(t)}$

$$\tilde{h}^{(t)} = \sum_{i=1}^J a(i, t) \circ h^{(i)} \rightarrow \text{kontekstna kombinacija ih učinkova}$$

Definiramo nov "eksterni" vektor $\bar{H}(t)$

$$\bar{H}(t) = \tanh \left(M \begin{bmatrix} \tilde{h}^{(t)} \\ \tilde{c}^{(t)} \end{bmatrix} \right)$$

novi parametri

$$\Rightarrow O^{(t)} = V \cdot \bar{H}(t)$$

Implementacijski detalji

→

Dubina mreže u direktnoj mreži, definiranu važećim mrežama
problem računanja gradijenta (numerički problem) u okviru mrežama
(vanishing, exploding gradient)

Različite implementacije ($h^{(t)}, o^{(t)} = f_{\Theta}(h^{(t-1)}, x^{(t)})$)

- LSTM implementacija f_{Θ}

- GRU implementacija f_{Θ}

↳ gated recurrent units