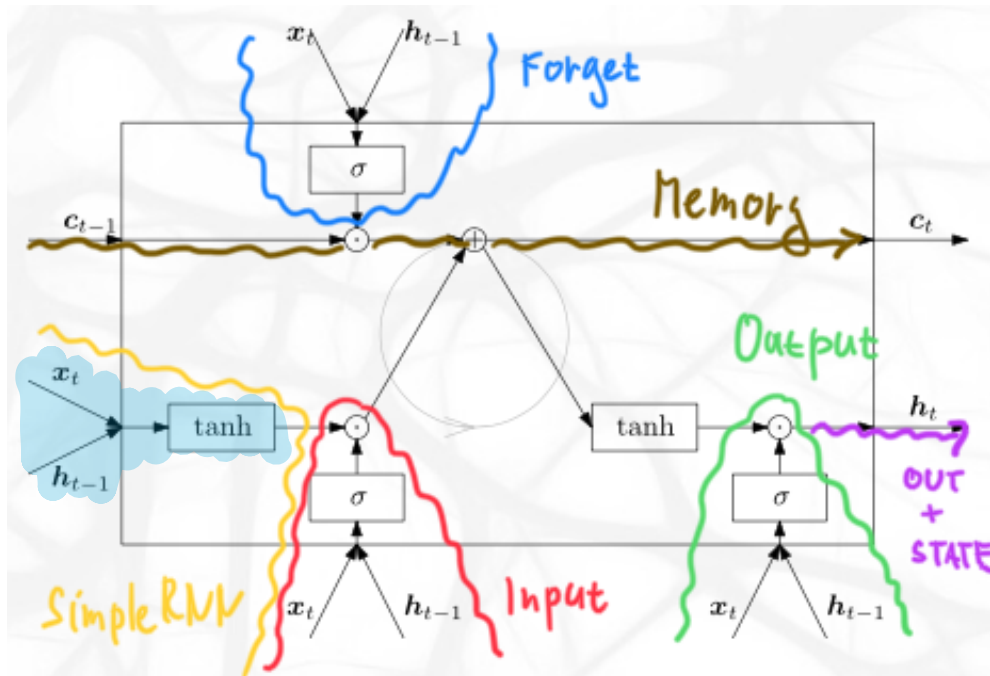


Lekce 8 [45]

Write down how the Long Short-Term Memory (LSTM) cell operates, including the explicit formulas. Also mention the forget gate bias. [10]

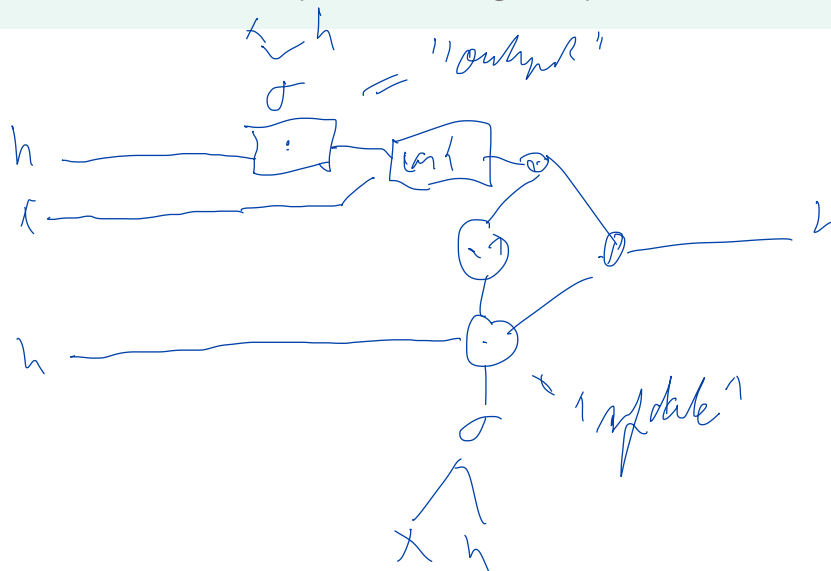


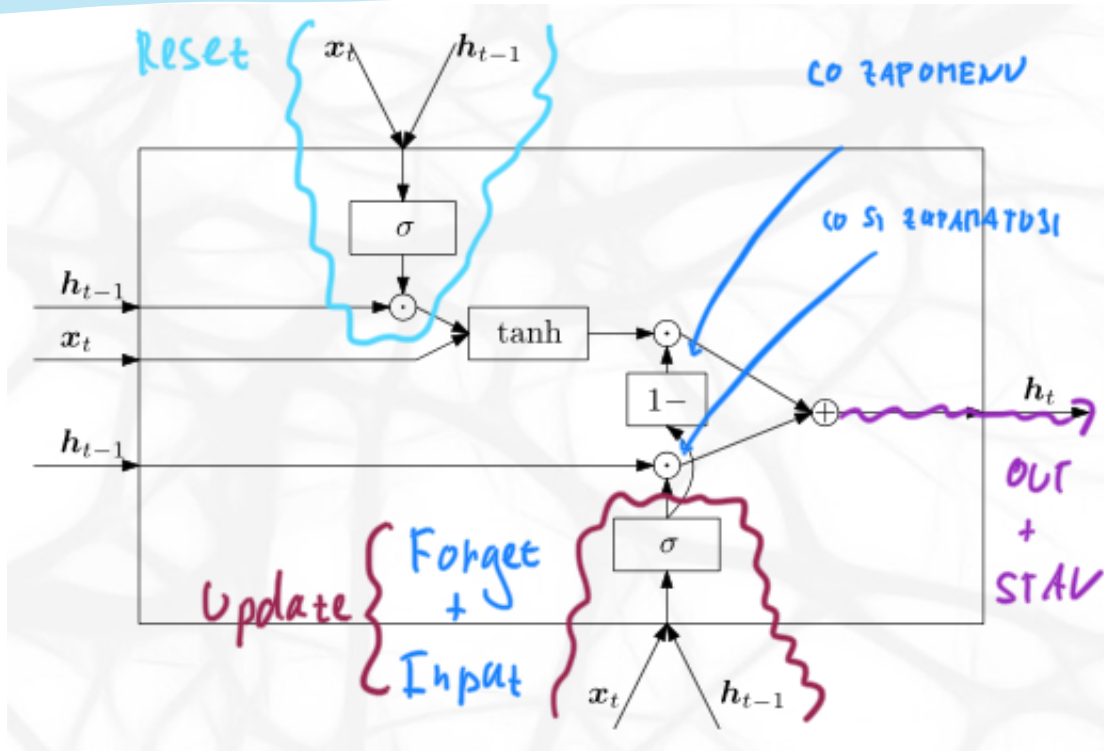
LSTM je vlastně Simple RNN buňka (tj tanh aktivovaná lineární kombinace vstupu a minulého stavu) rozšířená o paměť c a tři řídicí brány: input, output a v rozšířené verzi i forget.

$$\begin{aligned} i_t &\leftarrow \sigma(W^i x_t + V^i h_{t-1} + b^i) \\ f_t &\leftarrow \sigma(W^f x_t + V^f h_{t-1} + b^f) \\ o_t &\leftarrow \sigma(W^o x_t + V^o h_{t-1} + b^o) \\ c_t &\leftarrow f_t \cdot c_{t-1} + i_t \cdot \tanh(W^y x_t + V^y h_{t-1} + b^y) \\ h_t &\leftarrow o_t \cdot \tanh(c_t) \end{aligned}$$

Aktivace \tanh se používá schválně, aby nám nemohly explodovat gradienty (gradient je < 1).

Write down how the Gated Recurrent Unit (GRU) operates, including the explicit formulas. [10]





GRU je LSTM upravené tak, aby mělo méně parametrů. Oproti LSTM má:

- Reset gate, který je takovým posunutým Output gatem z LSTM. Slouží k určení, jakou část minulého stavu potřebujeme v současném výpočtu
- Update gate, který se stará o to, kterou část minulého stavu chceme zapomenout a nahradit něčím novým

$$\begin{aligned}
 r_t &\leftarrow \sigma(W^r x_t + V^r h_{t-1} + b^r) \\
 u_t &\leftarrow \sigma(W^u x_t + V^u h_{t-1} + b^u) \\
 \hat{h}_t &\leftarrow \tanh(W^h x_t + V^h (r_t \cdot h_{t-1}) + b^h) \\
 h_t &\leftarrow u_t \cdot h_{t-1} + (1 - u_t) \cdot \hat{h}_t
 \end{aligned} \tag{1}$$

Describe Highway network computation. [5]

Vypadá hodně podobně jako GRU, potažmo reziduální spojení. Do původní FC vrstvy

$$y \leftarrow H(x, W_H) \tag{2}$$

přidáme přes trénovatelný gating i původní vstup

$$y \leftarrow H(x, W_H) \cdot T(x, W_T) + x \cdot (1 - T(x, W_T)), \tag{3}$$

kde většinou $T(x, W_T) \leftarrow \sigma(W_T x + b_T)$.

Why the usual dropout cannot be used on recurrent state? Describe how can the problem be alleviated with variational dropout. [5]

Když budeme naivně náhodně dropovat ze stavu, nakonec nám tam moc dlouhodobých informací nezbyde — ale to je celý point RNN. Dropoutuje se ale běžně na vstupech i výstupech.

Variational dropout používá jednu masku na vstupy, jednu masku na výstupy, a jednu masku na stav; tím pádem se masky nemění v čase, a na některých místech nám dlouhodobá informace zůstane.

Describe layer normalization and write down an algorithm how it is used during training and an algorithm how it is used during inference. [5]

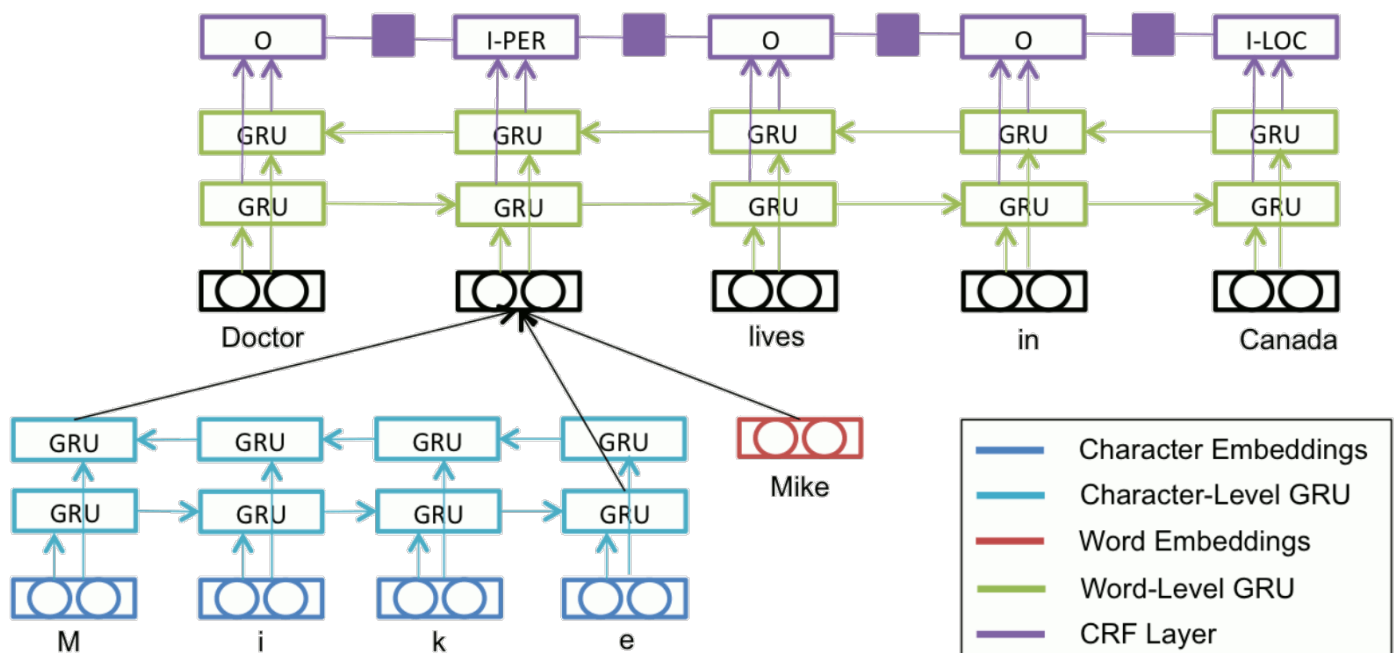
Pro RNN lepší než batchnorm. Funguje stejně jako batchnorm, s tím rozdílem, že normalizuje úplně celou vrstvu v rámci jednoho exemplu. Tedy pro vstupy x_i dané vrstvy v rámci jednoho example

$$\begin{aligned}\mu &\leftarrow \frac{1}{m} \sum_{i=1}^m \mathbf{x}^{(i)} \\ \sigma^2 &\leftarrow \frac{1}{m} \sum_{i=1}^m (\mathbf{x}^{(i)} - \mu)^2 \\ \hat{\mathbf{x}}^{(i)} &\leftarrow (\mathbf{x}^{(i)} - \mu) / \sqrt{\sigma^2 + \varepsilon} \\ \mathbf{y}^{(i)} &\leftarrow \gamma \hat{\mathbf{x}}^{(i)} + \beta\end{aligned}\tag{4}$$

Během inference si už nemusíme pamatovat předpočítané hodnoty μ a σ^2 , prostě je spočítáme z konkrétních vstupů, které dostaneme.

Sketch a tagger architecture utilizing word embeddings, recurrent character-level word embeddings and two sentence-level bidirectional RNNs with a residual connection. [10]

Nejbližší je tento slide, i když na konci by měla být ještě jedna vrstva.



S dvěma vrstvami na konci s reziduálními spojeními by to mohlo vypadat nějak takto.

