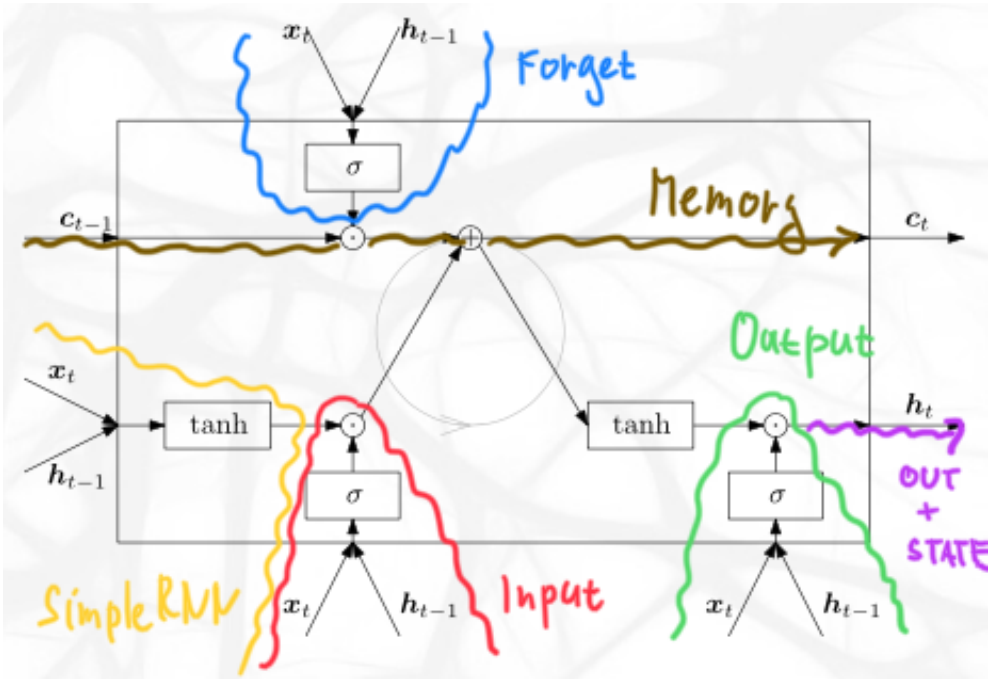
Lekce 8 [45]

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

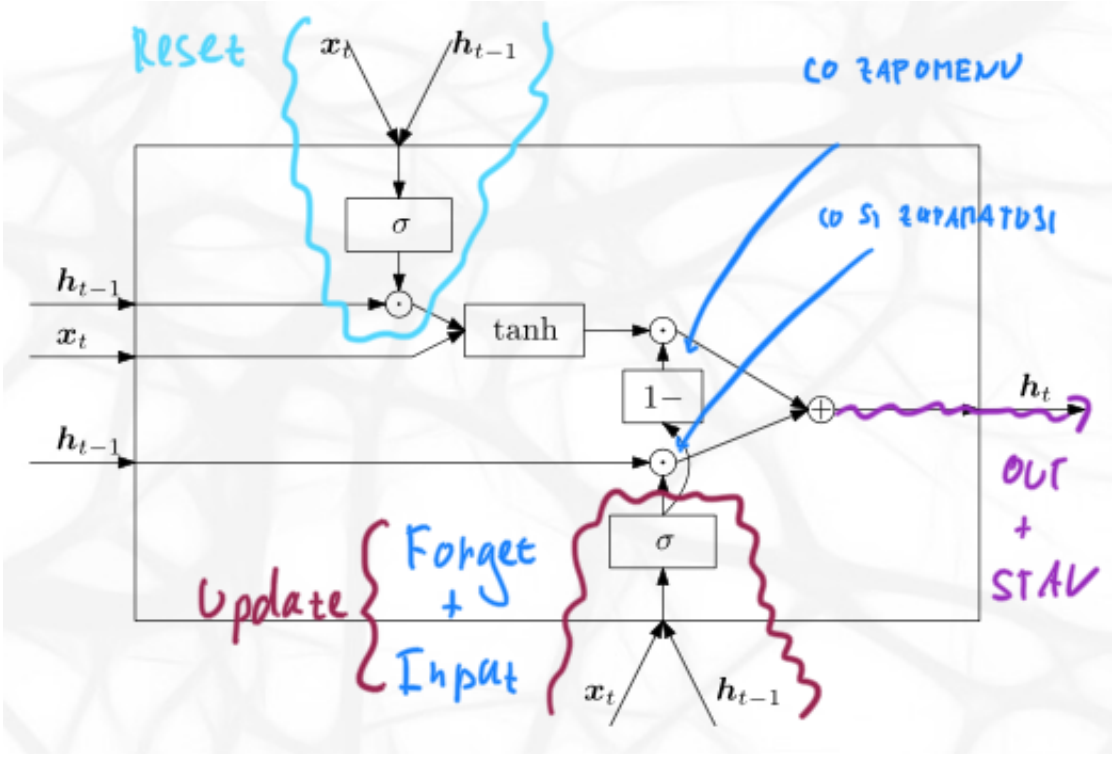


LSTM je vlastně Simple RNN buňka (tj $\tanh$ aktivovaná lineární kombinace vstupu a minulého stavu) rozšířená o paměť c a tři řídící brány: input, output a v rozšířené verzi i forget.

$$
\begin{aligned}
\boldsymbol{i}_t &\leftarrow \sigma\left(\boldsymbol{W}^i \boldsymbol{x}_t + \boldsymbol{V}^i \boldsymbol{h}_{t-1} + \boldsymbol{b}^i\right) \\
\boldsymbol{f}_t &\leftarrow \sigma\left(\boldsymbol{W}^f \boldsymbol{x}_t + \boldsymbol{V}^f \boldsymbol{h}_{t-1} + \boldsymbol{b}^f\right) \\
\boldsymbol{o}_t &\leftarrow \sigma\left(\boldsymbol{W}^o \boldsymbol{x}_t + \boldsymbol{V}^o \boldsymbol{h}_{t-1} + \boldsymbol{b}^o\right) \\
\boldsymbol{c}_t &\leftarrow \boldsymbol{f}_t \cdot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \cdot \tanh(\boldsymbol{W}^y \boldsymbol{x}_t + \boldsymbol{V}^y \boldsymbol{h}_{t-1} + \boldsymbol{b}^y) \\
\boldsymbol{h}_t &\leftarrow \boldsymbol{o}_t \cdot \tanh(\boldsymbol{c}_t)
\end{aligned}
$$

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

$$\begin{aligned}
\boldsymbol{r}_t &\leftarrow \sigma\left(\boldsymbol{W}^r \boldsymbol{x}_t + \boldsymbol{V}^r \boldsymbol{h}_{t-1} + \boldsymbol{b}^r\right) \\
\boldsymbol{u}_t &\leftarrow \sigma\left(\boldsymbol{W}^u \boldsymbol{x}_t + \boldsymbol{V}^u \boldsymbol{h}_{t-1} + \boldsymbol{b}^u\right) \\
\hat{\boldsymbol{h}}_t &\leftarrow \tanh\left(\boldsymbol{W}^h \boldsymbol{x}_t + \boldsymbol{V}^h \left(\boldsymbol{r}_t \cdot \boldsymbol{h}_{t-1}\right) + \boldsymbol{b}^h\right) \\
\boldsymbol{h}_t &\leftarrow \boldsymbol{u}_t \cdot \boldsymbol{h}_{t-1} + (1 - \boldsymbol{u}_t) \cdot \hat{\boldsymbol{h}}_t
\end{aligned} \tag{1}$$

> Describe Highway network computation. [5]

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

$$\boldsymbol{y} \leftarrow H\left(\boldsymbol{x}, \boldsymbol{W}_H\right) \tag{2}$$

přidáme přes trénovatelný gating i původní vstup

$$\boldsymbol{y} \leftarrow H\left(\boldsymbol{x}, \boldsymbol{W}_H\right) \cdot T\left(\boldsymbol{x}, \boldsymbol{W}_T\right) + \boldsymbol{x} \cdot \left(1 - T\left(\boldsymbol{x}, \boldsymbol{W}_T\right)\right), \tag{3}$$

kde většinou $T\left(\boldsymbol{x}, \boldsymbol{W}_T\right) \leftarrow \sigma\left(\boldsymbol{W}_T \boldsymbol{x} + \boldsymbol{b}_T\right)$.

> 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]

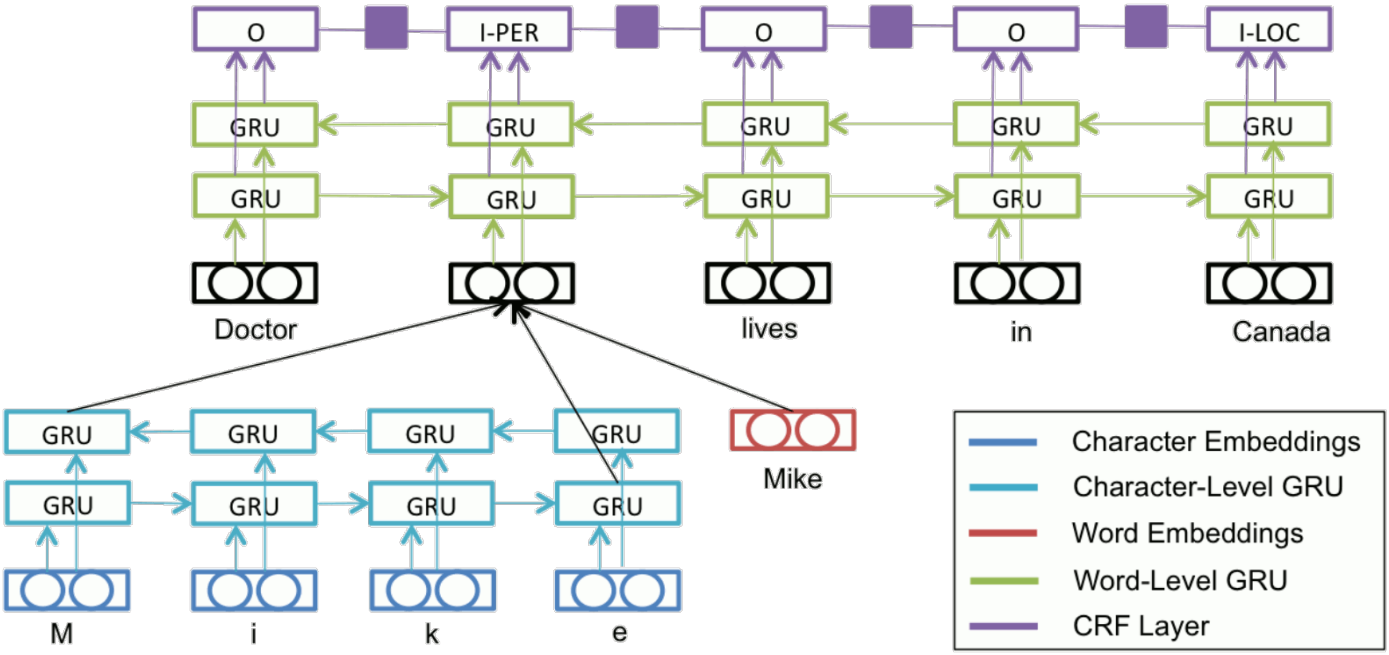
Pro RNN lepší než batchnorm. Funguje stejně jako batchnorm, s tím rozdílem, že normalizuje úplně celou vrstvu v rámci jednoho examplu. Tedy pro vstupy x_i dané vrstvy v rámci jednoho example

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

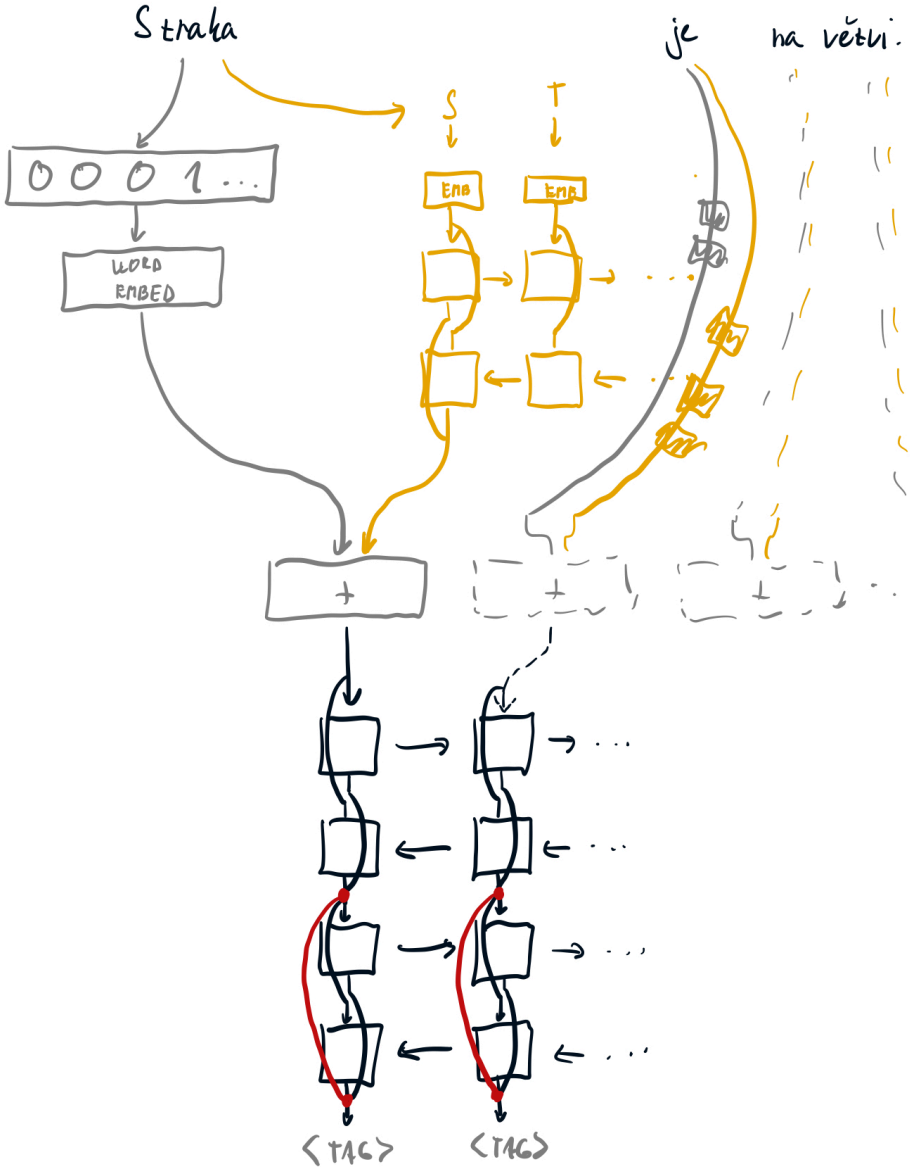
Během inference si už nemusíme pamatovat předpočítané hodnoty μ a σ^2, prostě je spočítáme z konkrétních vstupů, které dostaneme.

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

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



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


Straka
je
na větvi.
S
T
Emb
Emb
0 0 0 1 ...
WORD
EMBED
+
<TAG>
<TAG>