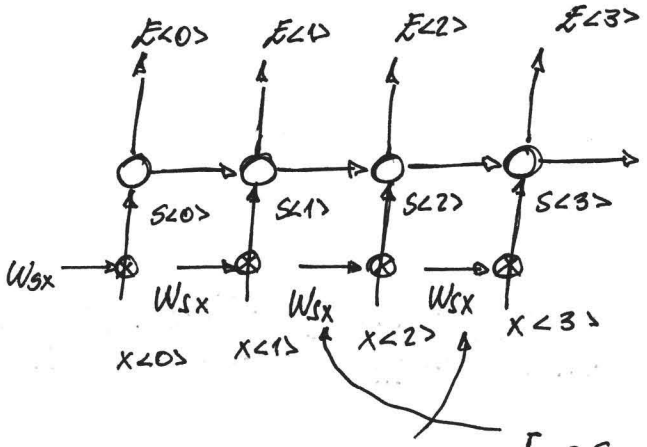
RNN Network Backprop through time



$$\frac{\partial E}{\partial W_{Sx}} = \frac{\partial E^{<3>}}{\partial S^{<3>}} \cdot \frac{\partial S^{<3>}}{\partial W_{Sx}} + \left[\frac{\partial E_3}{\partial S^{<3>}} \cdot \frac{\partial S^{<3>}}{\partial S^{<2>}} \cdot \frac{\partial S^{<2>}}{\partial W_{Sx}} + \frac{\partial E^{<2>}}{\partial S^{<2>}} \cdot \frac{\partial S^{<2>}}{\partial W_{Sx}} \right]$$

$$+ \left[\frac{\partial E^{<3>}}{\partial S^{<3>}} \cdot \frac{\partial S^{<3>}}{\partial S^{<2>}} \cdot \frac{\partial S^{<2>}}{\partial S^{<1>}} \cdot \frac{\partial S^{<1>}}{\partial W_{Sx}} + \frac{\partial E^{<2>}}{\partial S^{<2>}} \frac{\partial S^{<2>}}{\partial S^{<1>}} \cdot \frac{\partial S^{<1>}}{\partial W_{Sx}} + \right.$$

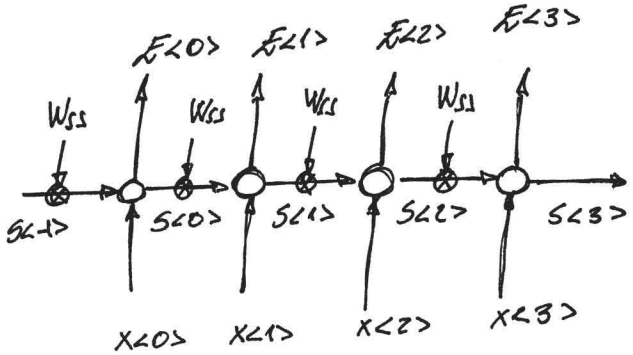
$$\left. + \frac{\partial E^{<1>}}{\partial S^{<1>}} \cdot \frac{\partial S^{<1>}}{\partial W_{Sx}} \right] + \left[\frac{\partial E^{<3>}}{\partial S^{<3>}} \cdot \frac{\partial S^{<3>}}{\partial S^{<2>}} \cdot \frac{\partial S^{<2>}}{\partial S^{<1>}} \cdot \frac{\partial S^{<1>}}{\partial S^{<0>}} \frac{\partial S^{<0>}}{\partial W_{Sx}} + \right.$$

$$+ \frac{\partial E^{<2>}}{\partial S^{<2>}} \cdot \frac{\partial S^{<2>}}{\partial S^{<1>}} \cdot \frac{\partial S^{<1>}}{\partial S^{<0>}} \cdot \frac{\partial S^{<0>}}{\partial W_{Sx}} + \frac{\partial E^{<1>}}{\partial S^{<1>}} \cdot \frac{\partial S^{<1>}}{\partial S^{<0>}} \cdot \frac{\partial S^{<0>}}{\partial W_{Sx}} +$$

$$\left. + \frac{\partial E^{<0>}}{\partial S^{<0>}} \frac{\partial S^{<0>}}{\partial W_{Sx}} \right] = \sum_{k=3}^{3} \frac{\partial E^{<k>}}{\partial S^{<k>}} \cdot \prod_{j=3+1}^{k} \left(\frac{\partial S^{<j>}}{\partial S^{<j-1>}} \right) \cdot \frac{S^{<3>}}{\partial W_{Sx}}$$

$$+ \sum_{k=2}^{3} \frac{\partial E^{<k>}}{\partial S^{<k>}} \cdot \prod_{j=3+1}^{k} \left(\frac{\partial S^{<j>}}{\partial S^{<j-1>}} \right) \cdot \frac{\partial S^{<2>}}{\partial W_{Sx}} + \dots \longrightarrow$$

$$\frac{\partial E}{\partial W_{Sx}} = \sum_{l=0}^{T_x} \sum_{k=l}^{T_y} \frac{\partial E^{<k>}}{\partial S^{<k>}} \prod_{j=l+1}^{k} \left(\frac{\partial S^{<j>}}{\partial S^{<j-1>}} \right) \cdot \frac{\partial S^{<l>}}{\partial W_{Sx}}$$



$$\frac{\partial E}{\partial W_{ss}} = \left[\frac{\partial E\langle 3\rangle}{\partial S\langle 3\rangle}\frac{\partial S\langle 3\rangle}{\partial W_{ss}}\right] + \left[\frac{\partial E\langle 3\rangle}{\partial S\langle 3\rangle}\cdot\frac{\partial S\langle 3\rangle}{\partial S\langle 2\rangle}\cdot\frac{\partial S\langle 2\rangle}{\partial W_{ss}} + \frac{\partial E\langle 2\rangle}{\partial S\langle 2\rangle}\cdot\frac{\partial S\langle 2\rangle}{\partial W_{ss}}\right] +$$

$$+ \left[\frac{\partial E\langle 3\rangle}{\partial S\langle 3\rangle}\cdot\frac{\partial S\langle 3\rangle}{\partial S\langle 2\rangle}\cdot\frac{\partial S\langle 2\rangle}{\partial S\langle 1\rangle}\cdot\frac{\partial S\langle 1\rangle}{\partial W_{ss}} + \frac{\partial E\langle 2\rangle}{\partial S\langle 2\rangle}\frac{\partial S\langle 2\rangle}{\partial S\langle 1\rangle}\cdot\frac{\partial S\langle 1\rangle}{\partial W_{ss}} +\right.$$

$$\left. + \frac{\partial E\langle 1\rangle}{\partial S\langle 1\rangle}\cdot\frac{\partial S\langle 1\rangle}{\partial W_{ss}}\right] + \left[\frac{\partial E\langle 0\rangle}{\partial S\langle 0\rangle}\cdot\frac{\partial S\langle 0\rangle}{\partial W_{ss}} + \frac{\partial E\langle 3\rangle}{\partial S\langle 3\rangle}\cdot\frac{\partial S\langle 3\rangle}{\partial S\langle 2\rangle}\cdot\frac{\partial S\langle 2\rangle}{\partial S\langle 1\rangle}\cdot\frac{\partial S\langle 1\rangle}{\partial S\langle 0\rangle}\right.$$

$$\left. \frac{\partial E\langle 0\rangle}{\partial W_{ss}} + \frac{\partial E\langle 2\rangle}{\partial S\langle 2\rangle}\cdot\frac{\partial S\langle 2\rangle}{\partial S\langle 1\rangle}\cdot\frac{\partial S\langle 1\rangle}{\partial S\langle 0\rangle}\cdot\frac{\partial S\langle 0\rangle}{\partial W_{ss}} + \frac{\partial E\langle 1\rangle}{\partial S\langle 1\rangle}\cdot\frac{\partial S\langle 1\rangle}{\partial S\langle 0\rangle}\cdot\frac{\partial S\langle 0\rangle}{\partial W_{ss}}\right] \Rightarrow$$

$$\frac{\partial E}{\partial W_{ss}} = \sum_{t=0}^{T_x}\sum_{k=t}^{T_y}\frac{\partial E_k}{\partial S\langle k\rangle}\prod_{j=t+1}^{k}\left(\frac{\partial S\langle j\rangle}{\partial S\langle j-1\rangle}\right)\frac{\partial S\langle t\rangle}{\partial W_{ss}}$$

$$\frac{\partial E}{\partial X\langle t\rangle} = \sum_{k=t}^{T_y}\frac{\partial E_k}{\partial S\langle k\rangle}\prod_{j=t+1}^{k}\left(\frac{\partial S\langle j\rangle}{\partial S\langle j-1\rangle}\right)\frac{\partial S\langle t\rangle}{\partial X\langle t\rangle}$$

* These consecutive multiplications may cause vanishing gradient or exploding gradient:

- Exploding gradient $\longrightarrow \frac{\partial E}{\partial \theta} \longrightarrow \left\|\frac{\partial E}{\partial \theta}\right\| > \text{Threshold} \longrightarrow \frac{\partial E}{\partial \theta} = \frac{\text{thres}}{\left\|\frac{\partial E}{\partial \theta}\right\|}\cdot\frac{\partial E}{\partial \theta}$

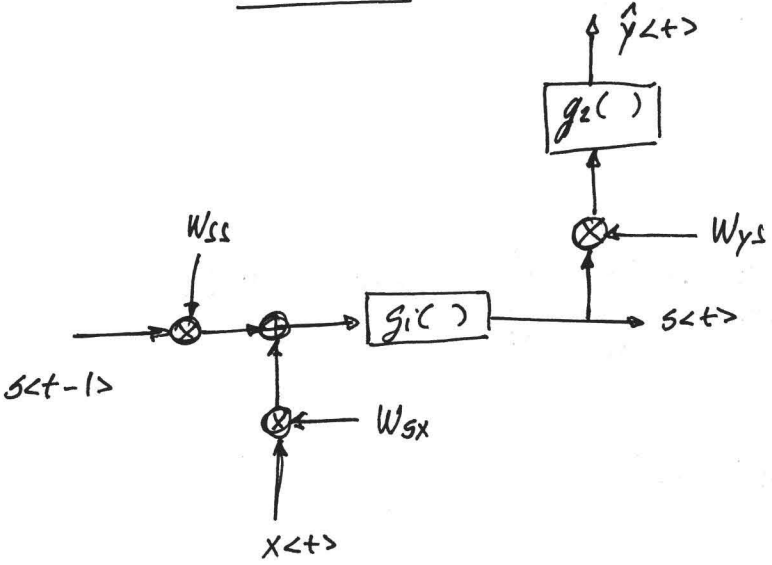
- Vanishing gradient $\longrightarrow$ LSTM, GRU

$$\frac{\partial S_{ki}\langle t\rangle}{\partial S_{ij}\langle t-1\rangle} = \tanh'(z_{ki}\langle t\rangle)\cdot W_{ssKi}\,S_{ij}$$

$0.X \leftarrow z \le -1$, $z \ge 1$ $\longrightarrow$ < 1 ; $0.X^T \longrightarrow$ Vanishing

> 1 ; $1.X^T \longrightarrow$ Exploding

RNN Cell



* Forward →

$$s^{\langle t \rangle} = g_1(z^{\langle t \rangle})$$

$$z^{\langle t \rangle} = W_{ss}\, s^{\langle t-1 \rangle} + W_{sx}\, x^{\langle t \rangle} + b_s$$

$$\hat{y}^{\langle t \rangle} = g_2(p^{\langle t \rangle})$$

$$p^{\langle t \rangle} = W_{ys}\, s^{\langle t \rangle} + b_y$$

Dimensions → $x^{\langle t \rangle} \in \mathbb{R}^{(n_x, m)}$; $s^{\langle t \rangle} \in \mathbb{R}^{(n_s, m)}$; $y^{\langle t \rangle} \in \mathbb{R}^{(n_y, m)}$

$W_{ss} \in \mathbb{R}^{(n_s, n_s)}$; $W_{sx} \in \mathbb{R}^{(n_s, n_x)}$; $W_{ys} \in \mathbb{R}^{(n_y, n_s)}$

$$\frac{\partial s_{ke}^{\langle t \rangle}}{\partial z_{ij}^{\langle t \rangle}} = g_1'(z_{ke}^{\langle t \rangle})\, \delta_{ki}\, \delta_{ej} \quad ; \quad \frac{\partial \hat{y}_{ke}^{\langle t \rangle}}{\partial p_{ij}^{\langle t \rangle}} = g_2'(s_{ke}^{\langle t \rangle})\, \delta_{ki}\, \delta_{ej}$$

$$\frac{\partial z_{ke}^{\langle t \rangle}}{\partial W_{ss_{ij}}} = \delta_{ki}\, s_{je}^{\langle t-1 \rangle} \quad ; \quad \frac{\partial z_{ke}^{\langle t \rangle}}{\partial W_{sx_{ij}}} = \delta_{ki}\, x_{je}^{\langle t \rangle} \quad ; \quad \frac{\partial z_{ke}^{\langle t \rangle}}{\partial s_{ij}^{\langle t-1 \rangle}} = W_{ss_{ki}}\, \delta_{ej}$$

$$\frac{\partial z_{ke}^{\langle t \rangle}}{\partial x_{ij}^{\langle t \rangle}} = W_{sx_{ki}}\, \delta_{ej} \quad ; \quad \frac{\partial \hat{y}_{ke}^{\langle t \rangle}}{\partial W_{ys_{ij}}} = \delta_{ki}\, s_{je}^{\langle t \rangle} \quad ; \quad \frac{\partial p_{ke}^{\langle t \rangle}}{\partial s_{ij}^{\langle t \rangle}} = W_{ys_{ki}}\, \delta_{ej}$$

$$\frac{\partial s_{ke}^{\langle t \rangle}}{\partial x_{ij}^{\langle t \rangle}} = \sum_{r=1}^{n_s} \sum_{s=1}^{m} \frac{\partial s_{ke}^{\langle t \rangle}}{\partial z_{rs}^{\langle t \rangle}} \frac{\partial z_{rs}^{\langle t \rangle}}{\partial x_{ij}^{\langle t \rangle}} = \sum_{r=1}^{n_s} \sum_{s=1}^{m} g_1'(z_{ke}^{\langle t \rangle})\, \delta_{ki}\, \delta_{ej}$$

$$\cdot W_{sx_{ki}}\, \delta_{sj} = g_1'(z_{ke}^{\langle t \rangle}) \cdot W_{ki}\, \delta_{ej} \quad ; \quad \frac{\partial s_{ke}^{\langle t \rangle}}{\partial s_{ij}^{\langle t-1 \rangle}} = g_1'(z_{ke}^{\langle t \rangle}) \cdot W_{ss_{ki}}\, \delta_{ej}$$

$W_{sx_{ki}}$

$$\frac{\partial s_{ke}^{\langle t \rangle}}{\partial W_{ss_{ij}}} = \sum_{r=1}^{n_s} \sum_{s=1}^{m} \frac{\partial s_{ke}^{\langle t \rangle}}{\partial z_{rs}^{\langle t \rangle}} \cdot \frac{\partial z_{rs}^{\langle t \rangle}}{\partial W_{ss_{ij}}} = \sum_{r=1}^{n_s} \sum_{s=1}^{m} g_1'(z_{ke}^{\langle t \rangle})\, \delta_{kr}\, \delta_{es}\, \delta_{ri}\, s_{js}^{\langle t-1 \rangle}$$

$$= g_1'(z_{ke}^{\langle t \rangle})\, s_{je}^{\langle t-1 \rangle}\, \delta_{ki}$$

$s_{je}^{\langle t-1 \rangle}$

$$\frac{\partial s_{re\langle t\rangle}}{\partial w_{xs,ij}} = g_1'(z_{re\langle t\rangle}) \cdot x_{je\langle t\rangle}\, \delta_{ki}$$

$$\frac{\partial \hat{y}_{re\langle t\rangle}}{\partial s_{ij\langle t\rangle}} = \sum_{r=1}^{n_y}\sum_{s=1}^{m} \frac{\partial \hat{y}_{re\langle t\rangle}}{\partial p_{rs\langle t\rangle}} \cdot \frac{\partial p_{rs\langle t\rangle}}{\partial s_{ij\langle t\rangle}} = \sum_{r=1}^{n_y}\sum_{s=1}^{m} g_2'(p_{re\langle t\rangle})\, \delta_{kr}\, \delta_{is}$$

$$w_{ys}{}_{ri}\, \delta_{sj} = g_2'(p_{re\langle t\rangle}) \cdot w_{ys\,ki}\, \delta_{ej}$$

$$\frac{\partial \hat{y}_{re\langle t\rangle}}{\partial w_{ys,ij}} = \sum_{r=1}^{n_y}\sum_{s=1}^{m} \frac{\partial \hat{y}_{re\langle t\rangle}}{\partial p_{rs\langle t\rangle}} \frac{\partial p_{rs\langle t\rangle}}{\partial w_{ys,ij}} = \sum_{r=1}^{n_y}\sum_{s=1}^{m} g_2'(p_{re\langle t\rangle})\, \delta_{kr}\, \delta_{es}$$

$$\cdot\, s_{js\langle t\rangle}\, \delta_{ri} = g_2'(p_{re\langle t\rangle}) \cdot s_{je\langle t\rangle}\, \delta_{ki}$$

$$\frac{\partial \hat{y}_{re\langle t\rangle}}{\partial w_{xs,ij}} = \sum_{r=1}^{n_s}\sum_{s=1}^{m} \frac{\partial \hat{y}_{re\langle t\rangle}}{\partial s_{rs\langle t\rangle}} \frac{\partial s_{rs\langle t\rangle}}{\partial w_{xs,ij}} = \sum_{r=1}^{n_s}\sum_{s=1}^{m} g_2'(p_{re\langle t\rangle}) \cdot w_{ys\,kr}\, \delta_{es}$$

$$\cdot\, g_1'(z_{rs\langle t\rangle}) \cdot x_{js\langle t\rangle}\, \delta_{ri} = g_2'(p_{re\langle t\rangle}) \cdot w_{ys\,ki}\, g'(z_{ie\langle t\rangle})\, x_{je\langle t\rangle}$$

$$\frac{\partial \hat{y}_{re\langle t\rangle}}{\partial s_{ij\langle t-1\rangle}} = \sum_{r=1}^{n_s}\sum_{s=1}^{m} \frac{\partial p_{re\langle t\rangle}}{\partial s_{rs\langle t\rangle}} \cdot \frac{\partial s_{rs\langle t\rangle}}{\partial s_{ij\langle t-1\rangle}} = \sum_{r=1}^{n_a}\sum_{s=1}^{m} g_2'(p_{re\langle t\rangle}) \cdot w_{ys\,kr}\, \delta_{es}$$

$$\cdot\, g_1'(z_{rs\langle t\rangle}) \cdot w_{ss\,ri} \cdot \delta_{sj} =$$

$$= \sum_{r=1}^{n_s} g_2'(p_{re\langle t\rangle}) \cdot w_{ys\,kr} \cdot g_1'(z_{rj\langle t\rangle})\, w_{ss\,ri}\, \delta_{ej}$$

$$\frac{\partial E^{\langle t+1:T_y\rangle}}{\partial s_{ij}^{\langle t-1\rangle}} = \sum_{k=1}^{n_s}\sum_{c=1}^{m} \frac{\partial E^{\langle t+1:T_y\rangle}}{\partial s_{kc}^{\langle t\rangle}} \cdot \frac{\partial s_{kc}^{\langle t\rangle}}{\partial s_{ij}^{\langle t-1\rangle}} = \sum_{k=1}^{n_s}\sum_{c=1}^{m} \frac{\partial E^{\langle t+1:T_y\rangle}}{\partial s_{kc}^{\langle t\rangle}} \cdot g_1'(z_{kc}^{\langle t\rangle}) \cdot W_{ss_{ki}} s_{cj}$$

$$= \sum_{k=1}^{n_s} \frac{\partial E^{\langle t+1:T_y\rangle}}{\partial s_{kj}^{\langle t\rangle}} \cdot g_1'(z_{kj}^{\langle t\rangle}) \cdot W_{ss_{ki}} \longrightarrow$$

vectorized implementation $\longrightarrow$

$$\frac{\partial E^{\langle t+1:T_y\rangle}}{\partial s_{ij}^{\langle t-1\rangle}} = W_{ss}^{T} \cdot \left(\frac{\partial E^{\langle t+1:T_y\rangle}}{\partial s^{\langle t\rangle}} * g_1'(z^{\langle t\rangle}) \right)$$

$$\frac{\partial E^{\langle t+1:T_y\rangle}}{\partial x_{ij}^{\langle t\rangle}} = W_{sx}^{T} \left(\frac{\partial E^{\langle t+1:T_y\rangle}}{\partial s^{\langle t\rangle}} * g_1'(z^{\langle t\rangle}) \right)$$

$$\frac{\partial E^{\langle t+1:T_y\rangle}}{\partial W_{ss_{ij}}} = \sum_{k=1}^{n_s}\sum_{c=1}^{m} \frac{\partial E^{\langle t+1:T_y\rangle}}{\partial s_{kc}^{\langle t\rangle}} \cdot \frac{\partial s_{kc}^{\langle t\rangle}}{\partial W_{ss_{ij}}} = \sum_{k=1}^{n_s}\sum_{c=1}^{m} \frac{\partial E^{\langle t+1:T_y\rangle}}{\partial s_{kc}^{\langle t\rangle}} \cdot g_1'(z_{kc}^{\langle t\rangle}) \cdot s_{jc}^{\langle t-1\rangle} \cdot \delta_{ki}$$

$$= \sum_{c=1}^{m} \frac{\partial E^{\langle t+1:T_y\rangle}}{\partial s_{ic}^{\langle t\rangle}} \cdot g_1'(z_{ic}^{\langle t\rangle}) \cdot s_{jc}^{\langle t-1\rangle}$$

vectorized implementation $\longrightarrow$

$$\frac{\partial E^{\langle t+1:T_y\rangle}}{\partial W_{ss}} = \left(\frac{\partial E^{\langle t+1:T_y\rangle}}{\partial s^{\langle t\rangle}} * g_1'(z^{\langle t\rangle}) \right) \cdot s^{T\langle t-1\rangle}$$

$$\frac{\partial E^{\langle t+1:T_y\rangle}}{\partial W_{sx}} = \left(\frac{\partial E^{\langle t+1:T_y\rangle}}{\partial s^{\langle t\rangle}} * g_1'(z^{\langle t\rangle}) \right) \cdot x^{T\langle t\rangle}$$

$$\frac{\partial E^{<t>}}{\partial s_{ij}^{<t>}} = \sum_{k=1}^{n_y} \sum_{c=1}^{m} \frac{\partial E^{<t>}}{\partial \hat{y}_{kc}^{<t>}} \cdot \frac{\partial \hat{y}_{kc}^{<t>}}{\partial s_{ij}^{<t>}} = \sum_{k=1}^{n_y} \sum_{c=1}^{m} \frac{\partial E^{<t>}}{\partial \hat{y}_{kc}^{<t>}} \cdot g_2'(p_{kc}^{<t>}) \cdot W_{ys_{ki}} \cdot \delta_{ej}$$

$$= \sum_{k=1}^{n_y} \frac{\partial E^{<t>}}{\partial \hat{y}_{kj}^{<t>}} \cdot g_2'(p_{kj}^{<t>}) \cdot W_{ys_{kl}} -$$

$$\frac{\partial E^{<t>}}{\partial W_{ys_{ij}}} = \sum_{k=1}^{n_y} \sum_{c=1}^{m} \frac{\partial E^{<t>}}{\partial \hat{y}_{kc}^{<t>}} \cdot \frac{\partial \hat{y}_{kc}^{<t>}}{\partial W_{ys_{ij}}} = \sum_{k=1}^{n_y} \overbrace{\sum_{c=1}^{m}}^{} \frac{\partial E^{<t>}}{\partial \hat{y}_{kc}^{<t>}} \cdot g_2'(p_{kc}^{<t+1>}) \cdot s_{jc}^{<t>} \delta_{ki}$$

$$= \sum_{c=1}^{m} \frac{\partial E^{<t>}}{\partial \hat{y}_{ic}^{<t>}} \cdot g_2'(p_{ic}^{<t>}) \cdot s_{jc}^{<t>}$$

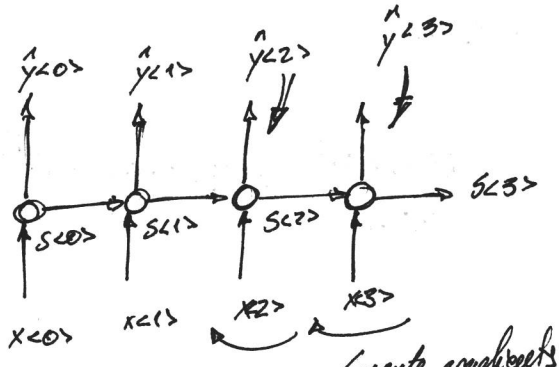
Vectorized Implementation $\longrightarrow$

$$\frac{\partial E^{<t>}}{\partial s^{<t>}} = W_{ys}^{T} \left(\frac{\partial E^{<t>}}{\partial \hat{y}^{<t>}} * g_2'(p^{<t>}) \right)$$

$$\frac{\partial E^{<t>}}{\partial W_{ys}} = \left(\frac{\partial E^{<y>}}{\partial \hat{y}^{<t>}} * g_2'(p^{<t>}) \right) \cdot s^{T<t>}$$

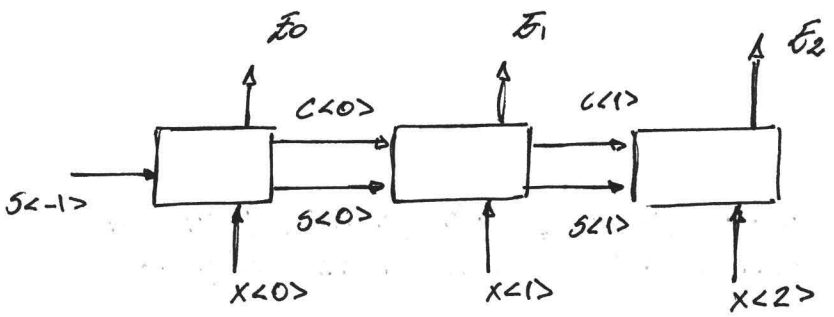
$$\frac{\partial E}{\partial s^{<t-1>}} = \left[W_{ys}^{T} \left(\frac{\partial E^{<t>}}{\partial \hat{y}^{<t>}} * g_2'(p^{<t>}) \right) * f_1'(z^{<t>}) \right] + W_{ss}^{T} \left(\frac{\partial E^{<t+1:T_y>}}{\partial s^{<t>}} * g_1' z^{<t>} \right)$$

W_{ss}^{T}



Implementation.

Compute gradients, Iterate across time & accumulate grad.



LSTM Network gradient through time

$$\frac{\partial E}{\partial W_f} = \left[\frac{\partial E_2}{\partial S_{<2>}} \cdot \frac{\partial S_{<2>}}{\partial C_{<2>}} \cdot \frac{\partial C_{<2>}}{\partial W_f}\right] + \left[\frac{\partial E_2}{\partial S_{<2>}} \cdot \frac{\partial S_{<2>}}{\partial C_{<2>}} \cdot \frac{\partial C_{<2>}}{\partial C_{<1>}} \cdot \frac{\partial C_{<1>}}{\partial W_{ax}} +\right.$$

$$\left. + \frac{\partial E_2}{\partial S_{<2>}} \cdot \frac{\partial S_{<2>}}{\partial S_{<2>}} \cdot \frac{\partial C_{<2>}}{\partial S_{<1>}} \cdot \frac{\partial S_{<1>}}{\partial C_{<1>}} \cdot \frac{\partial C_{<1>}}{\partial W_f} + \frac{\partial E_1}{\partial S_{<1>}} \cdot \frac{\partial S_{<1>}}{\partial C_{<1>}} \cdot \frac{\partial C_{<1>}}{\partial W_f}\right] +$$

$$+ \left[\frac{\partial E_2}{\partial S_{<2>}} \cdot \frac{\partial S_{<2>}}{\partial C_{<2>}} \cdot \frac{\partial C_{<2>}}{\partial C_{<1>}} \cdot \frac{\partial C_{<1>}}{\partial C_{<0>}} \cdot \frac{\partial C_{<0>}}{\partial W_f} +\right.$$

$$\left. + \frac{\partial E_2}{\partial S_{<2>}} \cdot \frac{\partial S_{<2>}}{\partial C_{<2>}} \cdot \frac{\partial C_{<2>}}{\partial C_{<1>}} \cdot \frac{\partial C_{<1>}}{\partial S_{<0>}} \cdot \frac{\partial S_{<0>}}{\partial C_{<0>}} \cdot \frac{\partial C_{<0>}}{\partial W_f}\right.$$

$$\left. + \frac{\partial E_2}{\partial S_{<2>}} \cdot \frac{\partial S_{<2>}}{\partial C_{<2>}} \cdot \frac{\partial C_{<2>}}{\partial S_{<1>}} \cdot \frac{\partial S_{<1>}}{\partial C_{<1>}} \cdot \frac{\partial C_{<1>}}{\partial S_{<0>}} \cdot \frac{\partial S_{<0>}}{\partial C_{<0>}} \cdot \frac{\partial C_{<0>}}{\partial W_f}\right.$$

$\partial S_{<1>}/\partial C_{<1>}$

$$\left. + \frac{\partial E_2}{\partial S_{<2>}} \cdot \frac{\partial S_{<2>}}{\partial C_{<2>}} \cdot \frac{\partial C_{<2>}}{\partial S_{<1>}} \cdot \frac{\partial S_{<1>}}{\partial S_{<0>}} \cdot \frac{\partial S_{<0>}}{\partial W_f} +\right.$$

$$\left. + \frac{\partial E_2}{\partial S_{<2>}} \cdot \frac{\partial S_{<2>}}{\partial C_{<2>}} \cdot \frac{\partial C_{<2>}}{\partial S_{<1>}} \cdot \frac{\partial S_{<1>}}{\partial C_{<1>}} \cdot \frac{\partial C_{<1>}}{\partial S_{<0>}} \cdot \frac{\partial S_{<0>}}{\partial C_{<0>}} \cdot \frac{\partial C_{<0>}}{\partial W_f}\right.$$

$$\left. + \frac{\partial E_1}{\partial S_{<1>}} \cdot \frac{\partial S_{<1>}}{\partial C_{<1>}} \cdot \frac{\partial C_{<1>}}{\partial C_{<0>}} \cdot \frac{\partial C_{<0>}}{\partial W_f} + \frac{\partial E_1}{\partial S_{<1>}} \cdot \frac{\partial S_{<1>}}{\partial C_{<1>}} \cdot \frac{\partial C_{<1>}}{\partial C_{<0>}} \cdot \frac{\partial C_{<0>}}{\partial W_f}\right.$$

$$\left. + \frac{\partial E_0}{\partial S_{<0>}} \cdot \frac{\partial S_{<0>}}{\partial C_{<0>}} \cdot \frac{\partial C_{<0>}}{\partial W_f}\right.$$

$$= \frac{\partial E_2}{\partial s^{\langle 2\rangle}} \cdot \frac{\partial s^{\langle 2\rangle}}{\partial c^{\langle 2\rangle}} \cdot \frac{\partial c^{\langle 2\rangle}}{\partial W_f} +$$

$$+ \left[\frac{\partial E_2}{\partial s^{\langle 2\rangle}} \cdot \frac{\partial s^{\langle 2\rangle}}{\partial c^{\langle 2\rangle}} \left(\frac{\partial c^{\langle 2\rangle}}{\partial c^{\langle 1\rangle}} + \frac{\partial c^{\langle 2\rangle}}{\partial s^{\langle 1\rangle}} \cdot \frac{\partial s^{\langle 1\rangle}}{\partial c^{\langle 1\rangle}} \right) \frac{\partial c^{\langle 1\rangle}}{\partial W_f} + \frac{\partial E_1}{\partial s^{\langle 1\rangle}} \frac{\partial s^{\langle 1\rangle}}{\partial c^{\langle 1\rangle}} \cdot \frac{\partial c^{\langle 1\rangle}}{\partial W_f} \right] +$$

$$+ \left[\frac{\partial E_2}{\partial s^{\langle 2\rangle}} \cdot \frac{\partial s^{\langle 2\rangle}}{\partial c^{\langle 2\rangle}} \left(\frac{\partial c^{\langle 2\rangle}}{\partial c^{\langle 1\rangle}} + \frac{\partial c^{\langle 2\rangle}}{\partial s^{\langle 1\rangle}} \cdot \frac{\partial s^{\langle 1\rangle}}{\partial c^{\langle 1\rangle}} \right) \left(\frac{\partial c^{\langle 1\rangle}}{\partial c^{\langle 0\rangle}} + \frac{\partial c^{\langle 1\rangle}}{\partial s^{\langle 0\rangle}} \cdot \frac{\partial s^{\langle 0\rangle}}{\partial c^{\langle 0\rangle}} \right) \frac{\partial c^{\langle 0\rangle}}{\partial W_f} \right.$$

$$+ \frac{\partial E_1}{\partial s^{\langle 1\rangle}} \cdot \frac{\partial s^{\langle 1\rangle}}{\partial c^{\langle 1\rangle}} \left(\frac{\partial c^{\langle 1\rangle}}{\partial c^{\langle 0\rangle}} + \frac{\partial c^{\langle 1\rangle}}{\partial s^{\langle 0\rangle}} \cdot \frac{\partial s^{\langle 0\rangle}}{\partial c^{\langle 0\rangle}} \right) \frac{\partial c^{\langle 0\rangle}}{\partial W_f} +$$

$$\left. + \frac{\partial E_1}{\partial s^{\langle 0\rangle}} \cdot \frac{\partial s^{\langle 0\rangle}}{\partial c^{\langle 0\rangle}} \cdot \frac{\partial c^{\langle 0\rangle}}{\partial W_f} \right] \Rightarrow$$

$$\frac{\partial E}{\partial W_f} = \sum_{l=0}^{T_x} \sum_{k=l}^{T_y} \frac{\partial E_k}{\partial s^{\langle k\rangle}} \cdot \frac{\partial s^{\langle k\rangle}}{\partial c^{\langle k\rangle}} \left[\prod_{j=l+1}^{k} \left(\frac{\partial c^{\langle j\rangle}}{\partial c^{\langle j-1\rangle}} + \frac{\partial c^{\langle j\rangle}}{\partial s^{\langle j-1\rangle}} \cdot \frac{\partial s^{\langle j-1\rangle}}{\partial c^{\langle j-1\rangle}} \right) \right] \frac{\partial c^{\langle l\rangle}}{\partial W_f}$$

$$\frac{\partial E}{\partial W_u} = \sum_{l=0}^{T_x} \sum_{k=l}^{T_y} \frac{\partial E_k}{\partial s^{\langle k\rangle}} \cdot \frac{\partial s^{\langle k\rangle}}{\partial c^{\langle k\rangle}} \left[\prod_{j=l+1}^{k} \left(\frac{\partial c^{\langle j\rangle}}{\partial c^{\langle j-1\rangle}} + \frac{\partial c^{\langle j\rangle}}{\partial s^{\langle j-1\rangle}} \cdot \frac{\partial s^{\langle j-1\rangle}}{\partial c^{\langle j-1\rangle}} \right) \right] \frac{\partial c^{\langle l\rangle}}{\partial W_u}$$

$$\frac{\partial E}{\partial W_c} = \sum_{l=0}^{T_x} \sum_{k=l}^{T_y} \frac{\partial E_k}{\partial s^{\langle k\rangle}} \cdot \frac{\partial s^{\langle k\rangle}}{\partial c^{\langle k\rangle}} \left[\prod_{j=l+1}^{k} \left(\frac{\partial c^{\langle j\rangle}}{\partial c^{\langle j-1\rangle}} + \frac{\partial c^{\langle j\rangle}}{\partial s^{\langle j-1\rangle}} \cdot \frac{\partial s^{\langle j-1\rangle}}{\partial c^{\langle j-1\rangle}} \right) \right] \frac{\partial c^{\langle l\rangle}}{\partial W_c}$$

$$\frac{\partial E}{\partial W_o} = \sum_{l=0}^{T_x} \sum_{k=l}^{T_y} \frac{\partial E_k}{\partial s^{\langle k\rangle}} \cdot \prod_{j=l+1}^{k} \left(\frac{\partial s^{\langle j\rangle}}{\partial s^{\langle j-1\rangle}} \right) \frac{\partial s^{\langle l\rangle}}{\partial W_o}$$

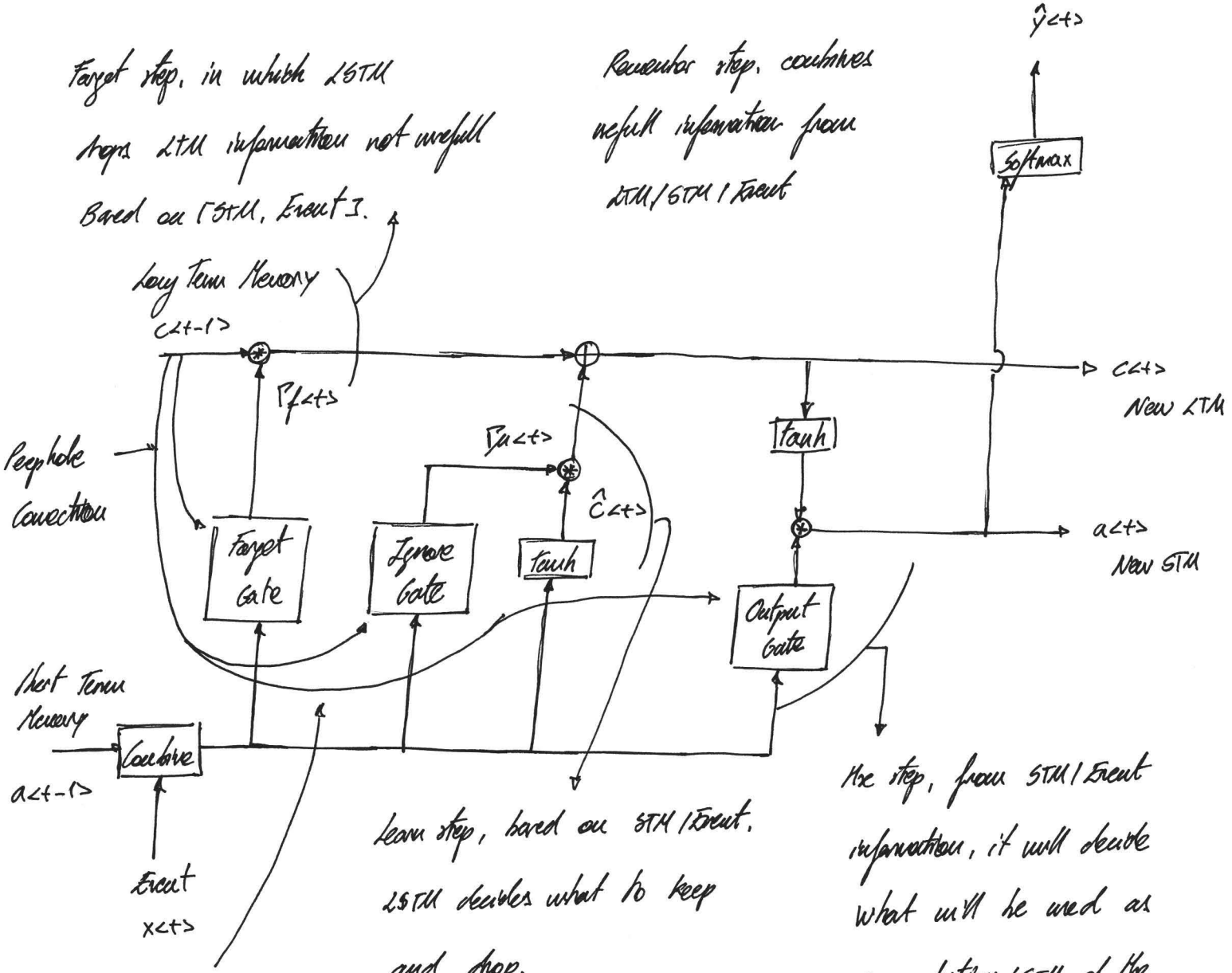
$$\frac{\partial c^{\langle j\rangle}}{\partial c^{\langle j-1\rangle}} = \Gamma_f^{\langle t\rangle} \Longrightarrow \prod \frac{\partial c^{\langle j\rangle}}{\partial c^{\langle j-1\rangle}}$$

[illegible] $\Longrightarrow$ Value (0,1) depending if it had importance $\Longrightarrow$ If enabled and $E\uparrow$, enabled update.

→ For this gradient, it won't vanish because of the path until previous block $\frac{\partial c^{\langle t+1\rangle}}{\partial a^{\langle t\rangle}} \cdot \frac{\partial a^{\langle t\rangle}}{\partial W_o}$

$$\frac{\partial E}{\partial x_{\langle l \rangle}} = \sum_{k=l}^{T_y} \frac{\partial E_k}{\partial s_{\langle k \rangle}} \cdot \frac{\partial s_{\langle k \rangle}}{\partial c_{\langle k \rangle}} \left[\prod_{j=l+1}^{k} \left(\frac{\partial c_{\langle j \rangle}}{\partial c_{\langle j-1 \rangle}} + \frac{\partial c_{\langle j \rangle}}{\partial s_{\langle j-1 \rangle}} \cdot \frac{\partial s_{\langle j-1 \rangle}}{\partial c_{\langle j-1 \rangle}} \right) \right] \frac{\partial c_{\langle l \rangle}}{\partial x_{\langle l \rangle}}$$

Long Short Term Memory



Learn step, based on STM / Event. LSTM decides what to keep and drop. Useful information of STM / Event.

Use step, from STM / Event information, it will decide what will be used as a prediction / STM of the combination LTM / STM / Event.

Peephole connections: LSTM variant, includes LTM in gate decisions.

$$\Gamma f\langle t\rangle = \sigma(W_f[a\langle t-1\rangle, x\langle t\rangle] + b_f)$$

$$\Gamma u\langle t\rangle = \sigma(W_u[a\langle t-1\rangle, x\langle t\rangle] + b_u)$$

$$\hat{c}\langle t\rangle = \tanh(W_c[a\langle t-1\rangle, x\langle t\rangle] + b_c)$$

$$c\langle t\rangle = c\langle t-1\rangle * \Gamma f\langle t\rangle + \Gamma u\langle t\rangle * \hat{c}\langle t\rangle$$

$$\Gamma o\langle t\rangle = \sigma(W_o[a\langle t-1\rangle, x\langle t\rangle] + b_o)$$

$$a\langle t\rangle = \Gamma o\langle t\rangle * \tanh(c\langle t\rangle)$$

cell backprop →

$\hat{y}^{\langle t \rangle} = \text{softmax}(z^{\langle t \rangle})$; $z^{\langle t \rangle} = W_y a^{\langle t \rangle} + b_y$

* cross-entropy gradient → $\frac{\partial E}{\partial \hat{y}_{ij}} = -\frac{1}{m} \cdot \frac{y_{ij}}{\hat{y}_{ij}}$

* softmax gradient → $\frac{\partial A_{rc}}{\partial z_{ij}} = \left[(A_{rc} - A_{rc})^2 \delta_{ki} - A_{rc} A_{ic} (1 - \delta_{ki}) \right] \delta_{cj}$

$\frac{\partial E^{\langle t \rangle}}{\partial z_{ij}^{\langle t \rangle}} = \frac{1}{m} \left[\hat{y}_{ij} - y_{ij} \right]$; $\frac{\partial z_{rc}^{\langle t \rangle}}{\partial a_{ij}^{\langle t \rangle}} = W_{y k,i} \, \delta_{cj}$

$$\frac{\partial E^{\langle t \rangle}}{\partial a_{ij}^{\langle t \rangle}} = \sum_{k=1}^{n_y} \sum_{s=1}^{m} \frac{\partial E^{\langle t \rangle}}{\partial z_{ks}^{\langle t \rangle}} \cdot \frac{\partial z_{rs}^{\langle t \rangle}}{\partial a_{ij}^{\langle t \rangle}} = \sum_{k=1}^{n_y} \sum_{s=1}^{m} \frac{1}{m} \left[\hat{y}_{rs} - y_{rs}^{\langle t \rangle} \right] \cdot W_{y n,i} \, \delta_{s,j}$$

$$= \sum_{k=1}^{n_y} \frac{1}{m} \left[\hat{y}_{rj} - y_{rj}^{\langle t \rangle} \right] \cdot W_{y n,j}$$

Vectorized → $\frac{\partial E^{\langle t \rangle}}{\partial a^{\langle t \rangle}} = \frac{1}{m} W_y^T \left[\hat{y}^{\langle t \rangle} - y^{\langle t \rangle} \right]$

$$\frac{\partial E^{\langle t \rangle}}{\partial W_y} = \frac{1}{m} \left[\hat{y}^{\langle t \rangle} - y^{\langle t \rangle} \right] \cdot a^{T\langle t \rangle}$$

$\frac{\partial C_{rc}^{\langle t \rangle}}{\partial C_{ij}^{\langle t-1 \rangle}} = \left(\ell + \Gamma_{f rc}^{\langle t \rangle} \right) \delta_{ki} \, \delta_{cj}$; $\frac{\partial a_{rc}^{\langle t \rangle}}{\partial C_{ij}^{\langle t \rangle}} = \Gamma_{o rc}^{\langle t \rangle} \left(1 - \tanh^2 \left(C_{rc}^{\langle t \rangle} \right) \right) \delta_{ki} \, \delta_{cj}$

$$\frac{\partial E^{\text{recurrent}}}{\partial C_{ij}^{\langle t \rangle}} = \frac{\partial E^{\langle t+1:T_y \rangle}}{\partial C_{ij}^{\langle t \rangle}} + \sum_{r=1}^{n_c} \sum_{s=1}^{m} \left(\frac{\partial E^{\langle t+1:T_y \rangle}}{\partial a_{rs}^{\langle t \rangle}} + \frac{\partial E^{\langle t \rangle}}{\partial a_{rs}^{\langle t \rangle}} \right) \Gamma_{o rs}^{\langle t \rangle} \left(1 - \tanh^2 \left(C_{rs}^{\langle t \rangle} \right) \right) \delta_{ri} \, \delta_{sj}$$

$$= \frac{\partial E^{\langle t+1:T_y \rangle}}{\partial C_{ij}^{\langle t \rangle}} + \left[\frac{\partial E^{\langle t+1:T_y \rangle}}{\partial a_{ij}^{\langle t \rangle}} + \frac{\partial E^{\langle t \rangle}}{\partial a_{ij}^{\langle t \rangle}} \right] \Gamma_{o ij}^{\langle t \rangle} \left(1 - \tanh^2 \left(C_{ij}^{\langle t \rangle} \right) \right)$$

Vectorized → $\frac{\partial E}{\partial C^{\langle t \rangle}} = \frac{\partial E^{\langle t+1:T_y \rangle}}{\partial C^{\langle t \rangle}} + \left[\frac{\partial E^{\langle t+1:T_y \rangle}}{\partial a^{\langle t \rangle}} + \frac{\partial E^{\langle t \rangle}}{\partial a^{\langle t \rangle}} \right] * \Gamma_o^{\langle t \rangle} * \left(1 - \tanh^2 \left(C^{\langle t \rangle} \right) \right)$

$$\frac{\partial E}{\partial c_{ij}^{<t-1>}} = \sum_{k=1}^{n_c}\sum_{c=1}^{m} \frac{\partial E}{\partial c_{kc}^{<t>}} \cdot \underbrace{\Gamma_{fk}^{<t>}\,\delta_{ki}\,\delta_{cj}}_{\partial c_{kc}^{<t>}/\partial c_{ij}^{<t-1>}} = \frac{\partial E}{\partial c_{kj}^{<t>}} \cdot \Gamma_{fij}^{<t>}$$

Vectorized $\longrightarrow$ $\frac{\partial E}{\partial c^{<t-1>}} = \frac{\partial E}{\partial c^{<t>}} * \Gamma_f^{<t>}$

$$\frac{\partial E}{\partial \Gamma_{fij}^{<t>}} = \frac{\partial E}{\partial c_{ij}^{<t>}} \cdot c_{ij}^{<t-1>} \;;\; \frac{\partial E}{\partial \Gamma_{u(ij)}} = \frac{\partial E}{\partial c_{ij}^{<t>}} * \hat{c}_{ij}^{<t>} \;;\; \frac{\partial E}{\partial \hat{c}_{ij}^{<t>}} = \frac{\partial E}{\partial c_{ij}^{<t>}} * \Gamma_{u,ij}^{<t>}$$

Vectorized $\longrightarrow$ $\frac{\partial E}{\partial \Gamma_f^{<t>}} = \frac{\partial E}{\partial c^{<t>}} * c^{<t-1>} \;;\; \frac{\partial E}{\partial \Gamma_u^{<t>}} = \frac{\partial E}{\partial c^{<t>}} * \hat{c}^{<t>} \;;\; \frac{\partial E}{\partial \hat{c}^{<t>}} = \frac{\partial E}{\partial c^{<t>}} * \Gamma_u^{<t>}$

$$\frac{\partial E}{\partial \Gamma_{o,ij}^{<t>}} = \sum_{r=1}^{n_c}\sum_{s=1}^{m}\left[\frac{\partial E^{<t>}}{\partial a_{rs}^{<t>}} + \frac{\partial E^{<t+1:T_y>}}{\partial a_{rs}^{<t>}}\right] \cdot \tanh\left(c_{ij}^{<t>}\right)\,\delta_{ri}\,\delta_{sj}$$

Vectorized $\longrightarrow$ $\frac{\partial E}{\partial \Gamma_o^{<t>}} = \frac{\partial E}{\partial a^{<t>}} * \tanh\left(c^{<t>}\right) \;;\; \frac{\partial E}{\partial W_o} = \left[\frac{\partial E}{\partial a^{<t>}} * \tanh\left(c^{<t>}\right)\right]\left[a^{<t-1>}, x^{<t>}\right]^T$

$$\frac{\partial E}{\partial [a^{<t-1>}, x^{<t>}]} = \sum_{r=1}^{n_c}\sum_{s=1}^{m}\Bigg[\frac{\partial E}{\partial \Gamma_{frs}^{<t>}} \cdot W_{frj}\,\delta_{sj} + \frac{\partial E}{\partial \Gamma_{urs}^{<t>}} \cdot W_{urj}\,\delta_{sj} + \frac{\partial E}{\partial \hat{c}^{<t>}}\,W_{crj}\,\delta_{sj}$$

$$+ \frac{\partial E}{\partial \Gamma_{org}^{<t>}} \cdot W_{orj} \cdot \delta_{sj}\Bigg] = \sum_{r=1}^{n_c}\Bigg[\frac{\partial E}{\partial \Gamma_{frj}^{<t>}} \cdot W_{fri} + \frac{\partial E}{\partial \Gamma_{urj}} \cdot W_{uri} +$$

$$+ \frac{\partial E}{\partial \Gamma_{orj}^{<t>}} W_{ori} + \frac{\partial E}{\partial \hat{c}_{nj}^{<t>}} W_{cri}\Bigg]$$

Vectorized

$$\frac{\partial E}{\partial [a^{<t-1>}, x^{<t>}]} = \frac{\partial E}{\partial \Gamma_f^{<t>}} \cdot W_f^T + W_u^T \cdot \frac{\partial E}{\partial \Gamma_u^{<t>}} + W_c^T \frac{\partial E}{\partial \hat{c}^{<t>}} + W_o^T\left(\frac{\partial E}{\partial \Gamma_o^{<t>}} * \Gamma_o^{<t>} * \left(1-\Gamma_o^{<t>}\right)\right)$$

$\leftarrow * \Gamma_f(1-\Gamma_f)$ (sigmoids) $\quad * \Gamma_u(1-\Gamma_u)$ (sigmoids) $\quad (1-\tanh^2(\cdot))\ (1-\hat{c}^{<t>})$ $\quad \sigma'(\ldots)$

$[a^{<t-1>}, x^{<t>}] \equiv \text{data}\,(n_a + n_x, m)$

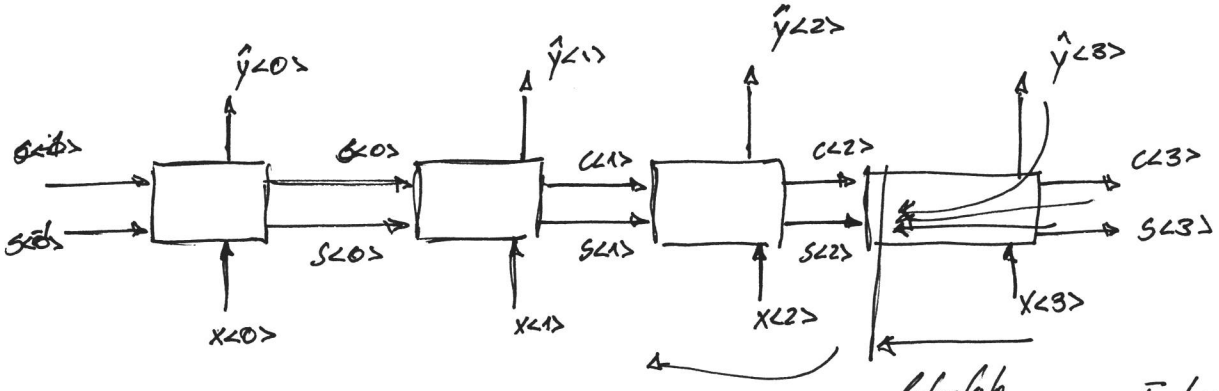
$$\longrightarrow \frac{\partial E}{\partial a^{<t-1>}} = \frac{\partial E}{[\partial a^{<t-1>}, \partial x^{<t>}]}[:n_a, m] \;;\; \frac{\partial E}{\partial x^{<t>}} = \frac{\partial E}{[\ldots]}[n_a : n_a + n_x, m]$$

Vorwärts →

$$\frac{\partial E}{\partial W_f} = \left[\frac{\partial E}{\partial \Gamma_f^{<t>}} * \Gamma_f^{<t>} * (1 - \Gamma_f^{<t>}) \right] \cdot [a^{<t-1>}, x^{<t>}]^T$$

$$\frac{\partial E}{\partial W_u} = \left[\frac{\partial E}{\partial \Gamma_u^{<t>}} * \Gamma_u^{<t>} * (1 - \Gamma_u^{<t>}) \right] \cdot [a^{<t-1>}, x^{<t>}]^T$$

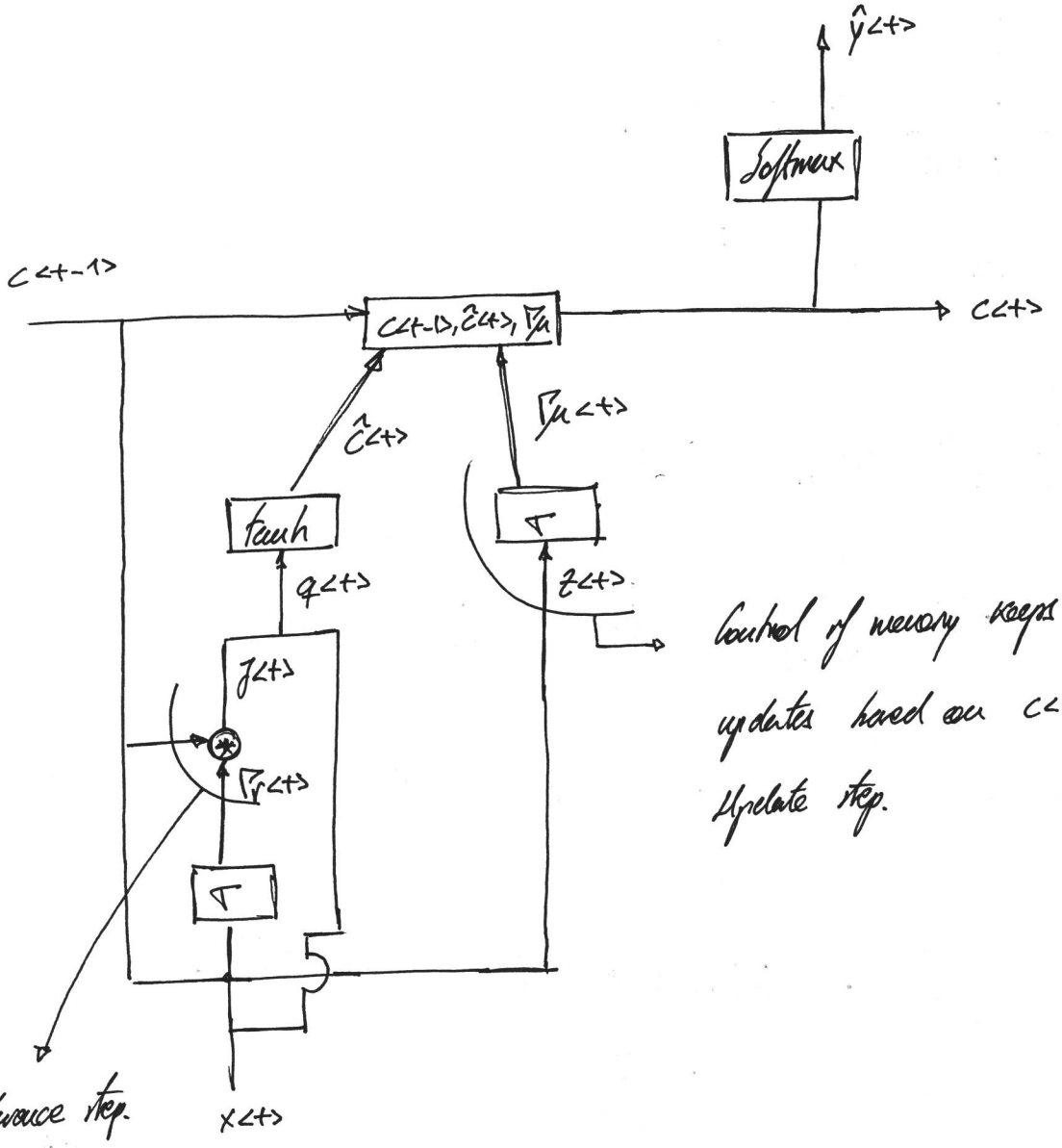
$$\frac{\partial E}{\partial W_c} = \left[\frac{\partial E}{\partial \tilde{c}^{<t>}} * (1 - \hat{c}^{2<t>}) \right] \cdot [a^{<t-1>}, x^{<t>}]^T$$



Calculate grad; Time step une.

Implementation

Gated Recurrent Unit



Control of memory keeps new values or updates based on $c^{<t-1>}, x^{<t>}$.
Update step.

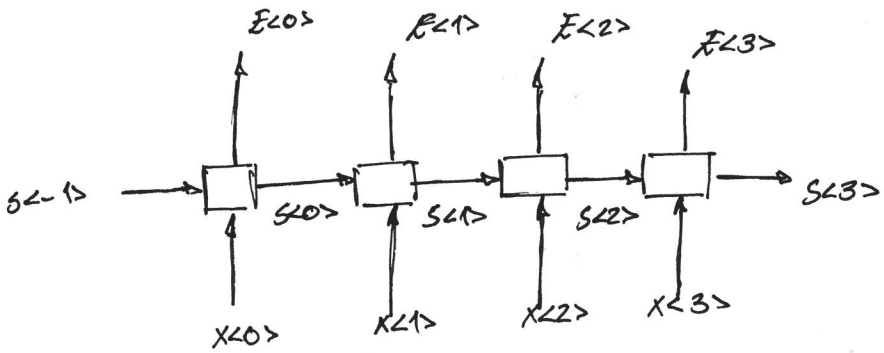
Decides which features to of memory we want to keep.

$$\Gamma_r^{<t>} = \sigma\left(W_r\left[c^{<t-1>}, x^{<t>}\right] + b_r\right)$$

$$\hat{c}^{<t>} = \tanh\left(W_c\left[\Gamma_r^{<t>} * c^{<t-1>}, x^{<t>}\right] + b_c\right)$$

$$\Gamma_u^{<t>} = \sigma\left(W_u\left[c^{<t-1>}, x^{<t>}\right] + b_u\right)$$

$$c^{<t>} = \Gamma_u^{<t>} * \hat{c}^{<t>} + \left(1 - \Gamma_u^{<t>}\right) * c^{<t-1>}$$



$$\frac{\partial E}{\partial W_a} = \sum_{l=0}^{T_x} \sum_{k=l}^{T_y} \frac{\partial E_k}{\partial S\langle k\rangle} \prod_{j=l+1}^{k} \left(\frac{\partial S\langle j\rangle}{\partial S\langle j-1\rangle} \right) \frac{\partial S\langle l\rangle}{\partial W_a}$$

$$\frac{\partial E}{\partial X\langle l\rangle} = \sum_{k=l}^{T_y} \frac{\partial E\langle k\rangle}{\partial S\langle k\rangle} \prod_{j=l+1}^{k} \left(\frac{\partial S\langle j\rangle}{\partial S\langle j-1\rangle} \right) \frac{\partial S\langle l\rangle}{\partial X\langle l\rangle}$$

Same as RNN (overall architecture), but $\partial S\langle j\rangle / \partial S\langle j-1\rangle$ has a linear path to prevent vanishing gradient $\longrightarrow$

$(1 - \Gamma_u\langle t\rangle)$ [illegible]

If enable or disable (0,1)

If enabled and $E\uparrow$, path to carry update. If disable on $E\uparrow$, W_a will change to minimize $\longrightarrow$ carry gradient when enabled.

GRU cell Backprop →

$\hat{y}^{<t>} = \text{softmax}\ ;\ z^{<t>} = W_y a^{<t>} + b_y$

$$\frac{\partial \mathcal{E}^{<t>}}{\partial a_{ij}^{<t>}} = \sum_{r=1}^{n_y} \sum_{s=1}^{m} \frac{\partial \mathcal{E}^{<t>}}{\partial z_{rs}^{<t>}} \cdot \frac{\partial z_{rs}^{<t>}}{\partial a_{ij}^{<t>}} = \sum_{r=1}^{n_y} \sum_{s=1}^{m} \frac{1}{m} \left[\hat{y}_{rs}^{<t>} - y_{rs}^{<t>} \right] \cdot W_{y_{ri}}\, \delta_{sj}$$

$$= \sum_{k=1}^{n_y} \frac{1}{m} \left[\hat{y}_{rj}^{<t>} - y_{rj}^{<t>} \right] \cdot W_{y_{ri}}$$

Vectorized → $\dfrac{\partial \mathcal{E}^{<t>}}{\partial a^{<t>}} = \dfrac{1}{m} W_y^T \left[\hat{y}^{<t>} - y^{<t>} \right]$

$$\frac{\partial \mathcal{E}^{<t>}}{\partial W_y} = \frac{1}{m} \left[\hat{y}^{<t>} - y^{<t>} \right] \cdot a^{T<t>}$$

$$\frac{\partial \mathcal{E}}{\partial c^{<t>}} = \frac{\partial \mathcal{E}^{<t>}}{\partial c^{<t>}} + \frac{\partial \mathcal{E}^{<t+1:T_y>}}{\partial c^{<t>}}$$

$$\frac{\partial c_{ke}^{<t>}}{\partial \Gamma_{\mu_{ij}}^{<t>}} = \hat{c}_{ke}^{<t>}\, \delta_{ki}\, \delta_{ej} \ ;\ \frac{\partial c_{ke}^{<t>}}{\partial \hat{c}_{ij}^{<t>}} = \Gamma_{\mu_{ij}}^{<t>}\, \delta_{ki}\, \delta_{ej}$$

$$\frac{\partial \mathcal{E}}{\partial \Gamma_{\mu}^{<t>}} = \frac{\partial \mathcal{E}}{\partial c^{<t>}} * \hat{c}^{<t>} \ ;\ \frac{\partial \mathcal{E}}{\partial \hat{c}^{<t>}} = \frac{\partial \mathcal{E}}{\partial c^{<t>}} * \Gamma_{\mu}^{<t>} \quad \leftarrow \text{Vectorized.}$$

$$\frac{\partial \mathcal{E}}{\partial z_{ij}^{<t>}} = \sum_{r=1}^{n_c} \sum_{s=1}^{m} \left(\frac{\partial \mathcal{E}}{\partial \Gamma_{\mu}^{<t>}} * \Gamma_{\mu_{rs}}^{<t>} \cdot \left(1 - \Gamma_{\mu_{rs}}^{<t>}\right) \right) \cdot W_{\mu_{ri}}\, \delta_{sj}$$

$$\frac{\partial \mathcal{E}}{\partial z^{<t>}} = W_{\mu}^T \left[\frac{\partial \mathcal{E}}{\partial \Gamma^{<t>}} * \Gamma_{\mu}^{<t>} * \left(1 - \Gamma_{\mu}^{<t>}\right) \right] \quad \leftarrow \text{Vectorized.}$$

$$\frac{\partial \mathcal{E}}{\partial a^{<t>}} = W_c^T \left[\frac{\partial \mathcal{E}}{\partial \hat{c}^{<t>}} * \left(1 - \hat{c}^{2<t>}\right) \right]$$

$J^{<t>} = q^{<t>}[:n_c,:]$ ⟶ Just elements corresponding to $c^{<t-1>}$

$$\frac{\partial E}{\partial J^{<t>}} = \frac{\partial E}{\partial q^{<t>}}[:n_c,:]$$

$$J^{<t>} = \Gamma_r^{<t>} * c^{<t-1>} \longrightarrow \frac{\partial J^{(t)ke}}{\partial \Gamma_{r_{ij}}^{<t>}} = c_{ke}^{<t-1>}\,\delta_{ki}\,\delta_{jl}\,;\quad \frac{\partial J_{ke}^{<t>}}{\partial c_{ij}^{<t-1>}} = \Gamma_{r_{ke}}^{<t>}\,\delta_{ki}\delta_{ij}$$

$$\frac{\partial E}{\partial p^{<t>}} = (\cancel{\partial E})\, W_r^{T}\left[\frac{\partial E}{\partial J^{<t>}} * c_k^{<t>} * \Gamma_r^{<t>} * (1-\Gamma_r^{<t>})\right]$$

$$\frac{\partial E}{\partial x^{<t>}} = \frac{\partial E}{\partial p^{<t>}}[n_c:,:] + \frac{\partial E}{\partial z^{<t>}}[n_c:,:] + \frac{\partial E}{\partial q^{<t>}}[n_c:,:]$$

$$\frac{\partial E}{\partial c^{<t-1>}} = \frac{\partial E}{\partial c^{<t>}} * (1-\Gamma_u^{<t>}) + \frac{\partial E}{\partial J^{<t>}} * \Gamma_r^{<t>} + \frac{\partial E}{\partial z^{<t>}}[:n_c,:]$$

← Vectorization.

Sequence Models

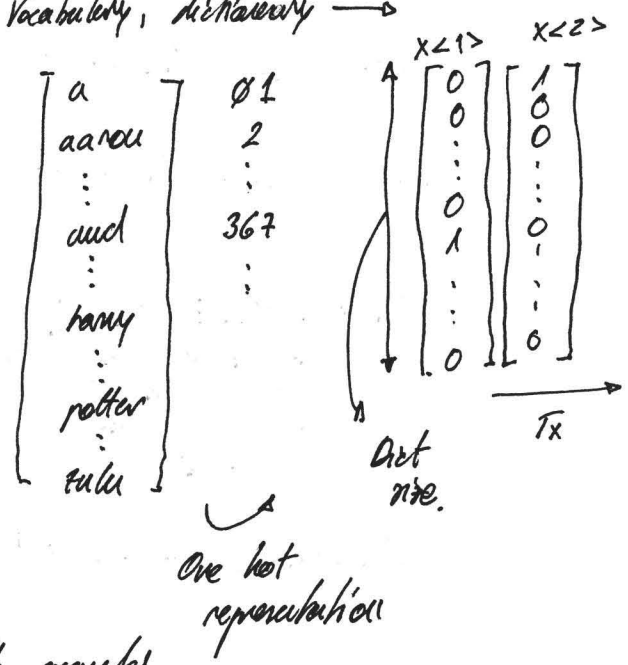
* Notation →

$x^{<t>}$ ≡ Sample t on the temporal space.

T_x ≡ Length of the sequence.

$x^{(i)<t>}$ ≡ Sample t on the i^{th} sample set.

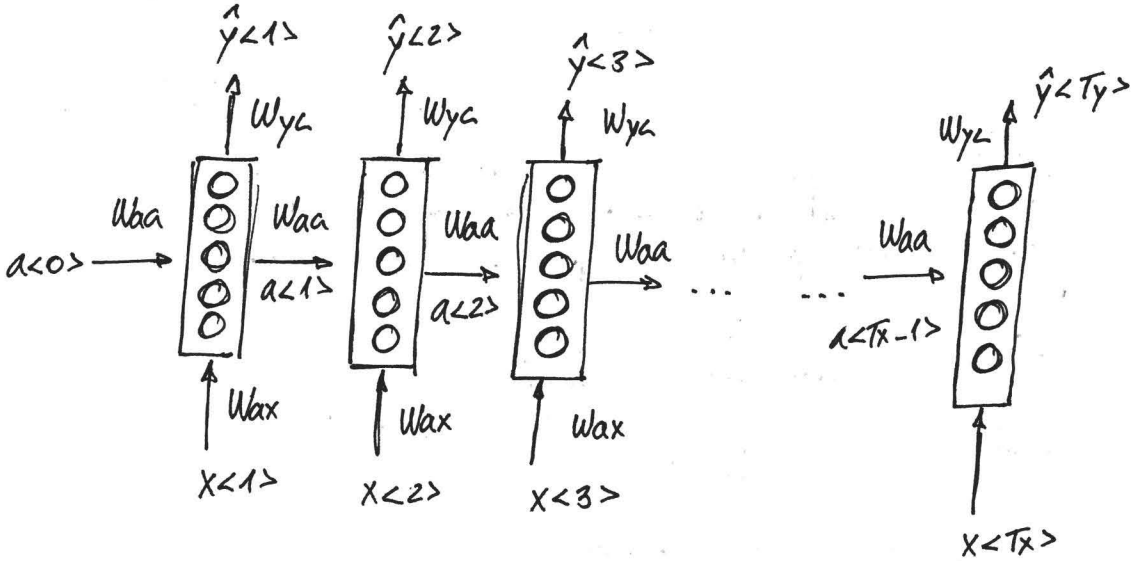
Vocabulary, dictionary →



* RNN model →

Reasons not to use DNN →

- Input or outputs can have different length examples.
- They don't share features learned across different portions of the sequence.
- If you want to use the whole set of words layer become huge.



$$a^{<t>} = g_1(W_{aa}\, a^{<t-1>} + W_{ax}\, x^{<t>} + b_a) \leftarrow \text{Tanh / ReLU}$$

$$\hat{y}^{<t>} = g_2(W_{ya}\, a^{<t>} + b_y) \leftarrow \text{sigmoid}$$

↳ Usually, not necessarily.

- Simplifying equations →

$$a^{<t>} = g_1(W_{aa}\, a^{<t-1>} + W_{ax}\, x^{<t>} + b_a)$$

$$a^{<t>} = g_1(W_a\, [a^{<t-1>}, x^{<t>}] + b_a)$$

$W_a = [W_{aa} : W_{ax}]$ ↕ 100 → Stacked horizontally.

100 (for W_{aa}), 10000 (for W_{ax}), 10100 (for W_a)

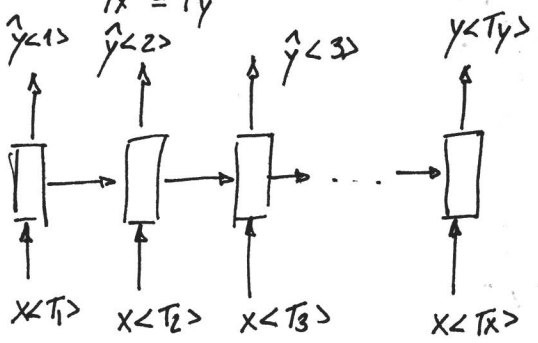
$$[a^{<t-1>}, x^{<t>}] = \begin{bmatrix} a^{<t-1>} \\ x^{<t>} \end{bmatrix} \begin{matrix} \updownarrow 100 \\ \updownarrow 10000 \end{matrix} \quad \updownarrow 10100$$

$$[W_{aa} : W_{ax}] \begin{bmatrix} a^{<t-1>} \\ x^{<t>} \end{bmatrix} = W_{aa}\, a^{<t-1>} + W_{ax}\, x^{<t>}$$

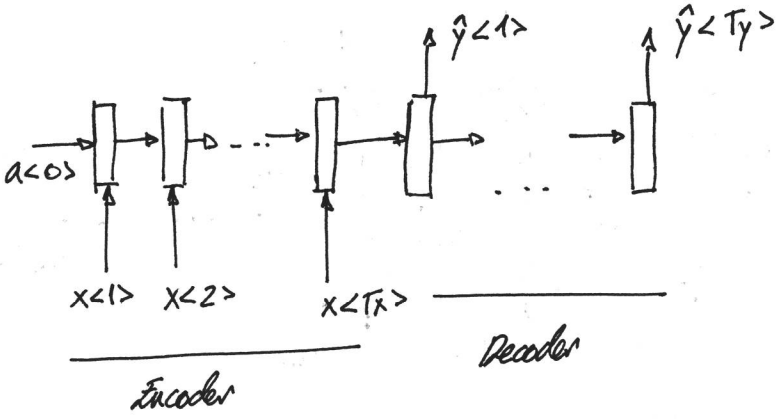
* Different RNNs →

Many-to-Many

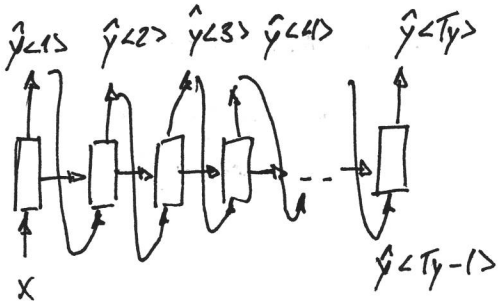
$T_x = T_y$



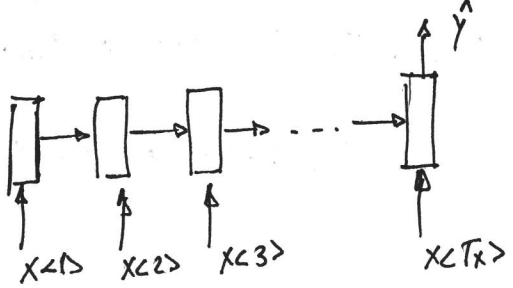
Many-to-Many



One-to-Many



Many-to-one



* Language model sequence generation →

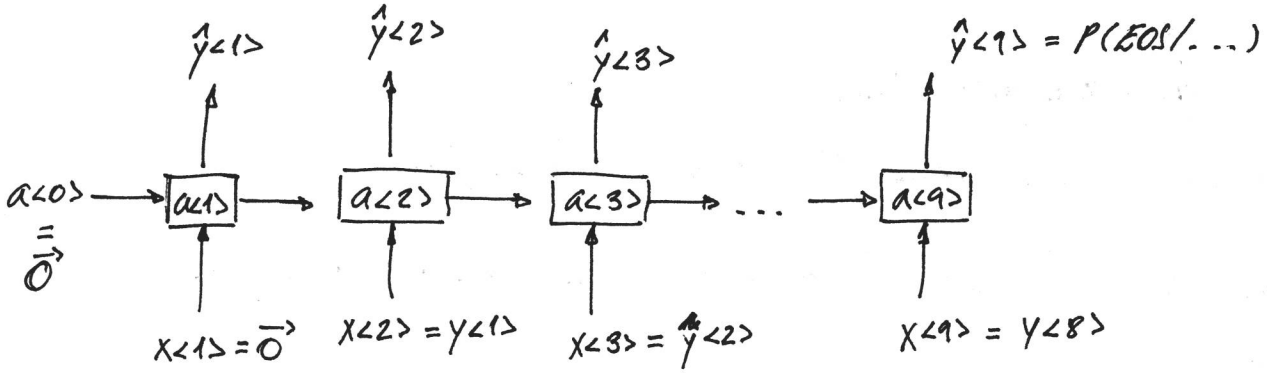
- Does probability assignation to word sequence → sentence.

$$P(y^{<1>}, y^{<2>}, y^{<3>}, \ldots, y^{<T_y>})$$

"The Egyptian Mau is a breed of a cat" → <EOS>

↑ Not in dict, <UNKS.

- RNN model →



$\hat{y}^{<1>}$ ≡ Softmax prediction of a word, $x^{<1>}$. ⟹ dim of dictionary.

$\hat{y}^{<2>} = P(\text{Egyptian} / y^{<1>} = \text{the})$

- Cost function →

$$\mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>}) = -\sum_i y_i^{<t>} \log \hat{y}_i^{<t>}$$

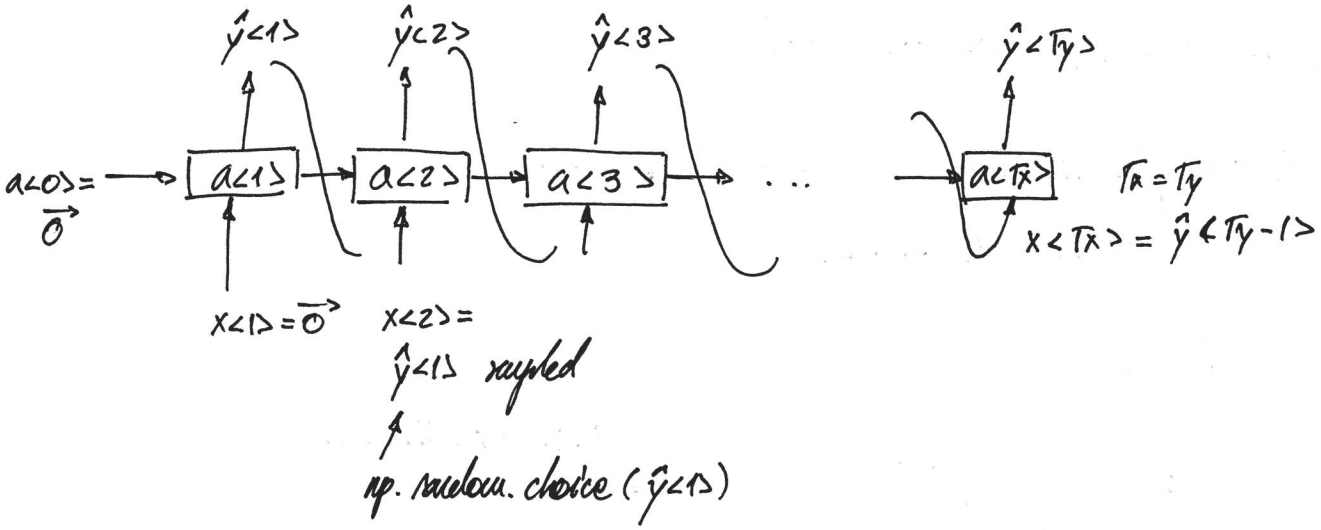
$$\mathcal{L}(\hat{y}, y) = \sum_t \mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>})$$

- Given new sentence; you can determine the probability →

$$P(y^{<1>}, y^{<2>}, y^{<3>}) = P(y^{<1>})\, P(y^{<2>} / y^{<1>})\, P(y^{<3>} / y^{<2>}\, y^{<1>})$$

* Sampling novel sequences →

The models represent the chance of having sequence words.



* Randomly sample across the softmax distribution ŷ<1>

* You can sample until the sentence has x words or when you hit EOS.

* Character level. →

vocabulary = [a, b, c, ..., z, ., ,, ;, 0, 1, ..., 9, A, ..., Z]

- Same idea as before → ŷ<1> = char ; E.g. = 'a'.

- This setup is able to assign P=0 to unknown words such as Mau.

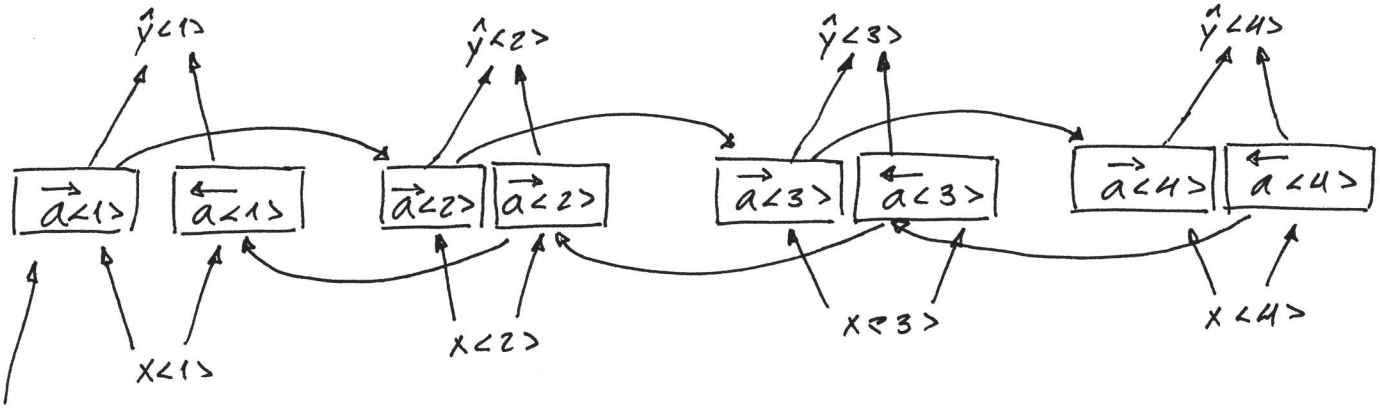
* Vanishing gradient →

- Long sequences → Problem with the gradients as in RNNs.

- Exploding gradients can also happen →

NaN ; gradient clipping → rescale gradient.

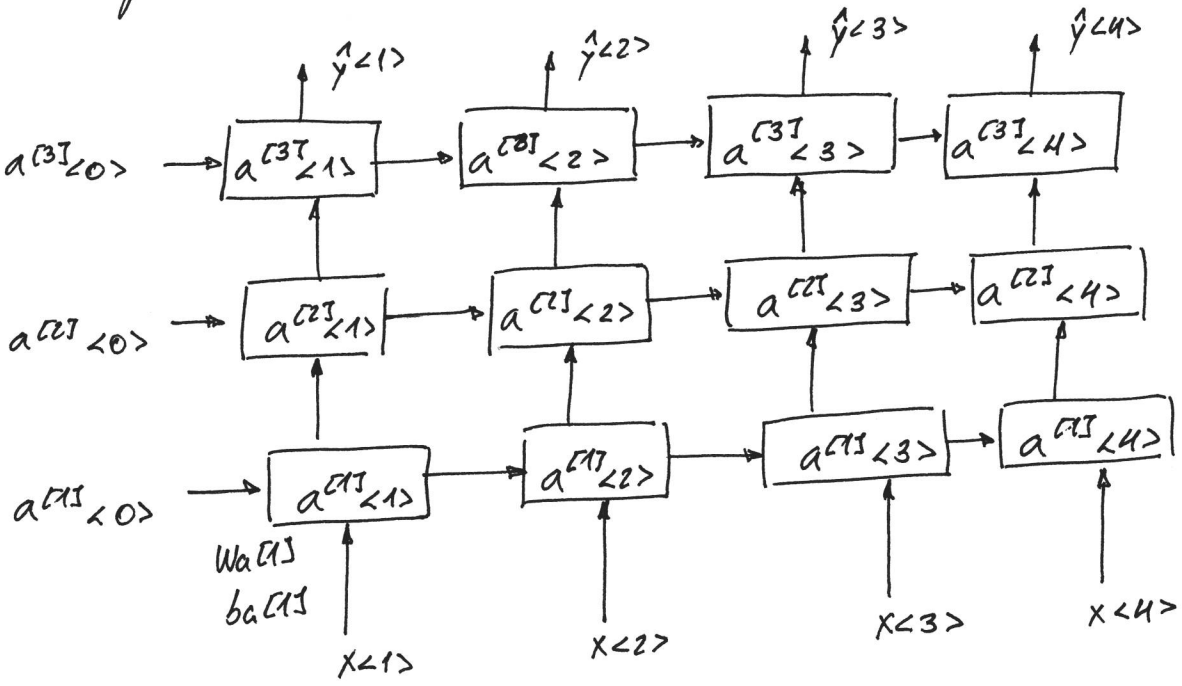
* Bidirectional RNNs →



RNN; GRU; LSTM.

$$\hat{y}^{<t>} = g(W_y [\vec{a}^{<t>}, \overleftarrow{a}^{<t>}] + b_y)$$

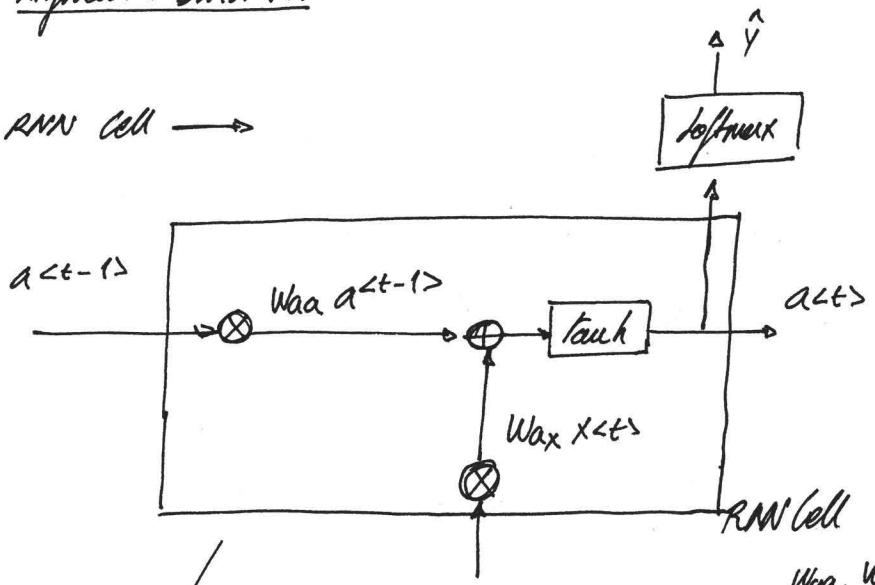
* Deep RNN →



$$a^{[2]<3>} = g(Wa^{[2]} [a^{[2]<2>}, a^{[1]<3>}] + ba^{[2]})$$

- Argument: Build RNN →

* RNN cell →



$a^{\langle t \rangle} = \tanh(W_{ax}x^{\langle t \rangle} + W_{aa}a^{\langle t-1 \rangle} + b_a)$

$\hat{y}^{\langle t \rangle} = \text{softmax}(W_{ya}\,a^{\langle t \rangle} + b_y)$

$x^{\langle t \rangle} \rightarrow n_x$ vector.

$a^{\langle t \rangle} \rightarrow n_a$ vector

$\hat{y}^{\langle t \rangle} \rightarrow n_y$ vector.

$W_{ax} \rightarrow (n_a, n_x)$ $W_{ya} \rightarrow (n_y, n_a)$

$W_{aa} \rightarrow (n_a, n_a)$

W_{aa}, W_{ax}, W_{ya} & b_a, b_y

cache backprop → $(a^{\langle t \rangle}, a^{\langle t-1 \rangle}, x^{\langle t \rangle}, \text{params})$

"Whole" →

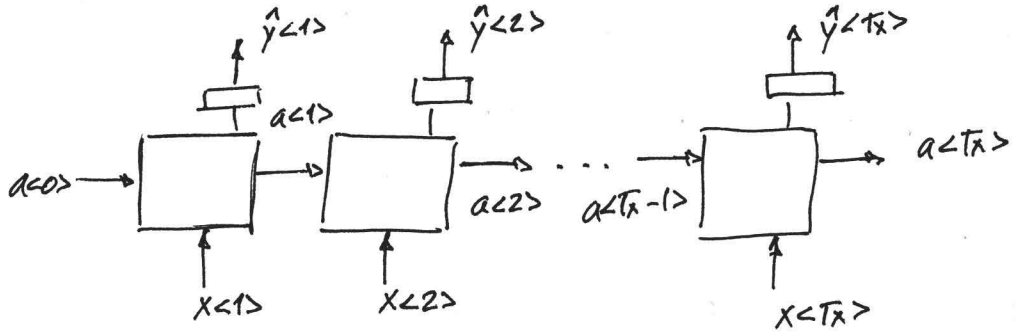
$a \rightarrow (n_a, m, T_x)$

$y \rightarrow (n_y, m, T_x)$

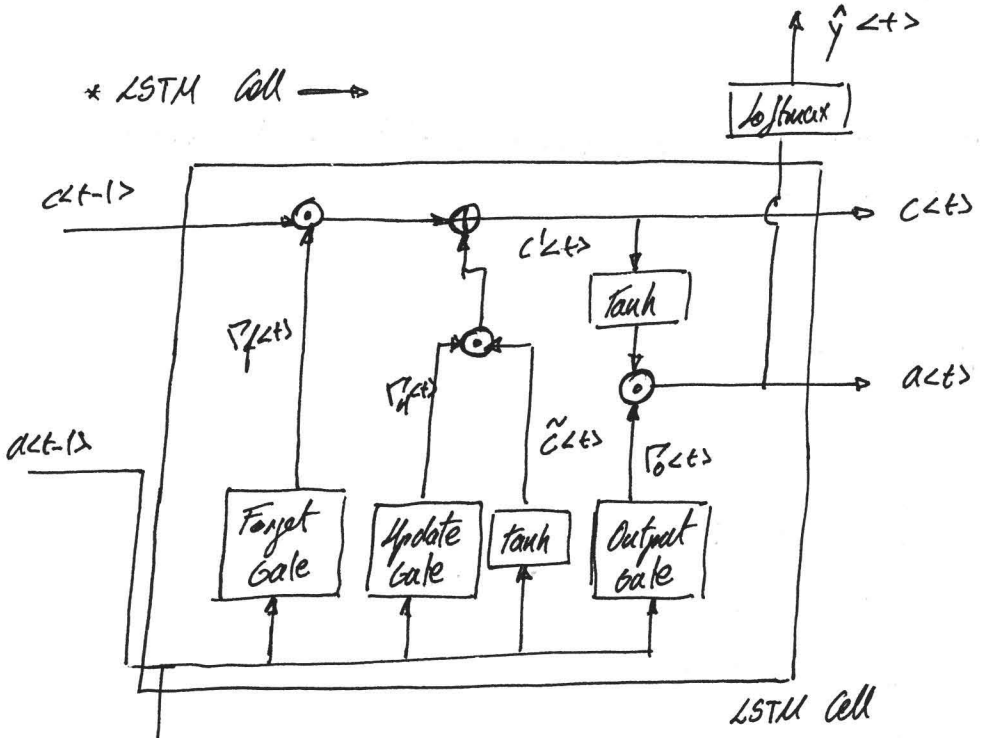
number samples

fine-tuning

* Forward pass →



* LSTM Cell →



$\Gamma_f^{\langle t \rangle} = \sigma(W_f[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_f)$

$\Gamma_u^{\langle t \rangle} = \sigma(W_u[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u)$

$\tilde{c}^{\langle t \rangle} = \tanh(W_c[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c)$

$c^{\langle t \rangle} = c^{\langle t-1 \rangle} * \Gamma_f^{\langle t \rangle} + \Gamma_u^{\langle t \rangle} * \tilde{c}^{\langle t \rangle}$

$\Gamma_o^{\langle t \rangle} = \sigma(W_o[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_o)$

$a^{\langle t \rangle} = \Gamma_o^{\langle t \rangle} * \tanh(c^{\langle t \rangle})$

- Cache → $(a^{\langle t \rangle}, a^{\langle t-1 \rangle}, c^{\langle t \rangle}, c^{\langle t-1 \rangle}, \Gamma_f, \Gamma_u, \tilde{c}^{\langle t \rangle}, \Gamma_o, x^{\langle t \rangle})$, parameters.

* Comments about gates →

- Forget gate →

  * E.g: piece of text, keep track of grammatical structures, such as subject is singular or plural. If it changes from singular to plural need a way to get rid of previous value.

- Update gate →

  * Once we forget the subject, we need to update the value so it reflects the new plural.

- Updating the cell →

  * To update the new subject, we need to create a new vector that we can add to our previous cell state.

- Output gate →

  * To decide which of the outputs we will use.

* Dimensions →

$W \equiv (n_a, n_x + n_a)$ $\quad x \equiv (n_x, m)$ $\quad a \equiv (n_a, m)$ $\quad \Big| \longrightarrow [a^{<t-1>}, x^{<t>}] \equiv (n_x + n_a, m)$

$c \equiv (n_a, m)$

$y \equiv (n_y, m)$

$\Gamma \Rightarrow \sigma(W[a^{<t-1>}, x^{<t>}] + b)$

$$\begin{pmatrix} W_{11} & \dots & W_{1\,n_x+n_a} \\ \vdots & & \vdots \\ W_{n_a 1} & \dots & W_{n_a\,n_x+n_a} \end{pmatrix} \begin{pmatrix} a_{11} & \dots & a_{1m} \\ \vdots & & \vdots \\ a_{n_a 1} & \dots & a_{n_a m} \\ \hline x_{11} & \dots & x_{1m} \\ \vdots & & \vdots \\ x_{n_x 1} & \dots & x_{n_x m} \end{pmatrix} \longrightarrow (n_a, m)$$

* LSTM ≡ long Short Term Memory ⟶

$$\hat{c}^{\langle t \rangle} = \tanh\left(W_c \left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_c\right)$$

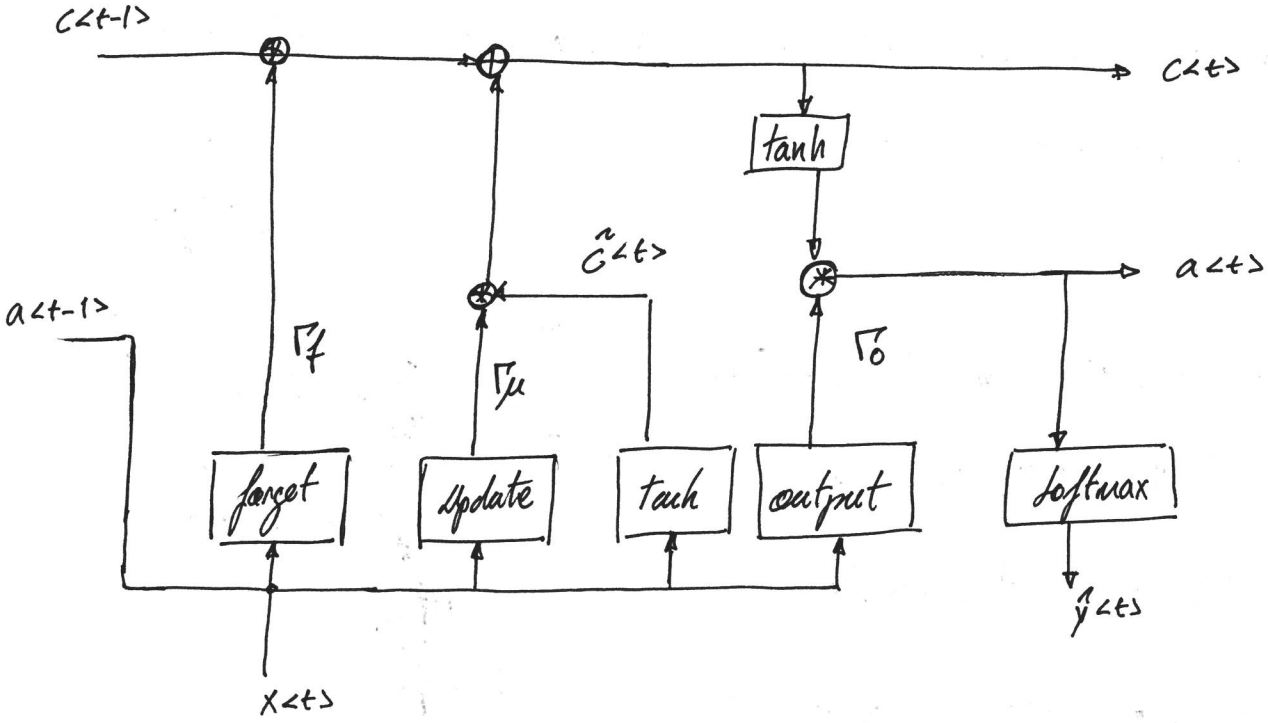
$$\Gamma_u = \sigma\left(W_u \left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_u\right) \quad ; \text{ update.}$$

$$\Gamma_f = \sigma\left(W_f \left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_f\right) \quad ; \text{ forget.}$$

$$\Gamma_o = \sigma\left(W_o \left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_o\right) \quad ; \text{ output.}$$

$$c^{\langle t \rangle} = \Gamma_u * \hat{c}^{\langle t \rangle} + \Gamma_f * c^{\langle t-1 \rangle}$$

$$a^{\langle t \rangle} = \Gamma_o * \tanh\left(c^{\langle t \rangle}\right)$$



- Possible variant ⟶ Peephole connection.

* Gate values depend on previous memory values.

$$\Gamma_o, \Gamma_f, \Gamma_u = \sigma\left(W \left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}, c^{\langle t-1 \rangle}\right] + b\right)$$

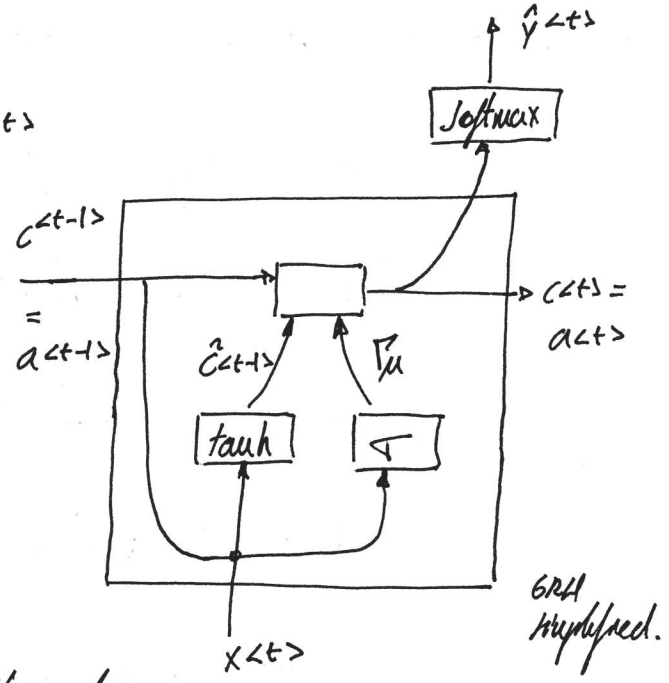
* GRU ≡ Gated Recurrent Unit ⟶

- Simplified ⟶ $c \equiv$ Memory Cell ; $c^{\langle t \rangle} = a^{\langle t \rangle}$

$$\hat{c}^{\langle t \rangle} = \tanh\left(W_c \left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_c\right)$$

$$\Gamma_u = \sigma\left(W_u \left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_u\right)$$

$$c^{\langle t \rangle} = \Gamma_u * \hat{c}^{\langle t \rangle} + (1 - \Gamma_u) * c^{\langle t-1 \rangle}$$



GRU simplified.

* $c^{\langle t \rangle}$, $\hat{c}^{\langle t \rangle}$, Γ_u ≡ have dimension.

* Γ_u ≡ Update gate, the network can get values close to 1, 0 for high, low values.

* Γ_u will control if $c^{\langle t \rangle}$ will update its value depending on new value $x^{\langle t \rangle}$ or previous $c^{\langle t-1 \rangle}$.

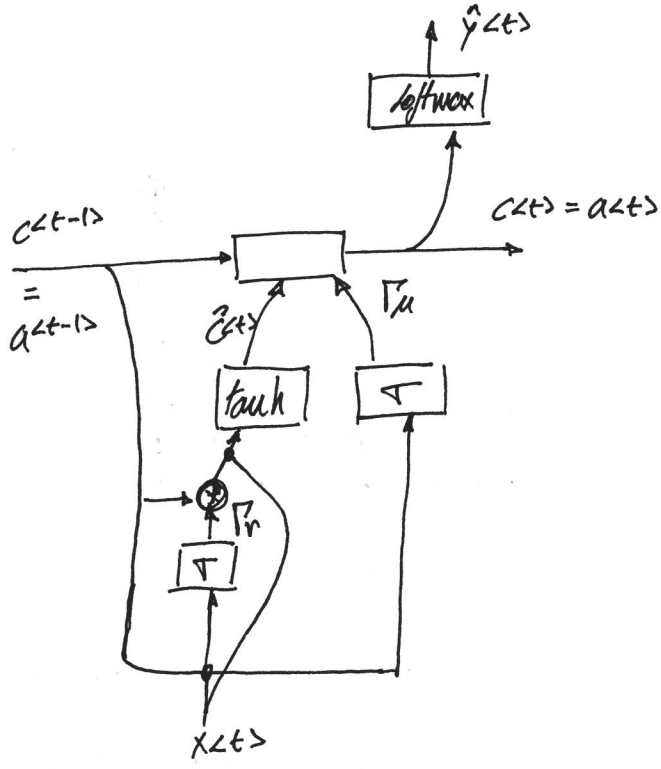
- Full GRU ⟶

$$\hat{c}^{\langle t \rangle} = \tanh\left(W_c \left[\Gamma_r * c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_c\right)$$

$$\Gamma_u = \sigma\left(W_u \left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_u\right)$$

$$\Gamma_r = \sigma\left(W_r \left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_r\right)$$

$$c^{\langle t \rangle} = \Gamma_u * \hat{c}^{\langle t \rangle} + (1 - \Gamma_u) * c^{\langle t-1 \rangle}$$



* Introduced relevance gate ≡ Γ_r

NLP & Word Embeddings →

* Word representation →

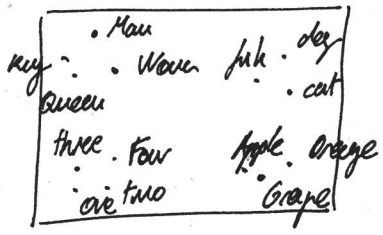
- One-hot representation per word has limitations, tell nothing about meaning or relationship between them.

Features	Man 5391	Woman 9853	King 4914	Queen 7157	Apple 456	Orange 6257
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.01
Food	0.04	0.01	0.02	0.01	0.95	0.97

↓ 300 features.

↓ e^{5391}

- Visualization →



t-SNE

300D → 2D

non-linear.

- Using word embedding makes the RNN learn meaning based on features →

If RNN was trained with "Sally Johnson is an orange farmer", given "Robert Lin is a durian cultivator", durian features will tell that it is a fruit and cultivator similar to farmer.

* Transfer learning →

1. Learn word embeddings from large textset (1-100B words) (or download pre-trained model).

2. Train embeddings to a new task with smaller training set → 100K words.

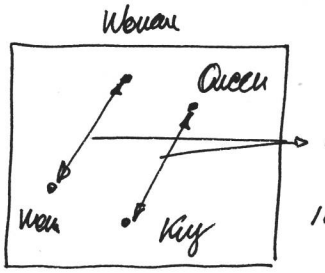
3. Continue to finetune the word embeddings with new data. (Optional).

* Properties →

- Analogy →

$e^{man} - e^{woman} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

$e^{king} - e^{queen} \approx \begin{bmatrix} -2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ → vector difference is gender.



vector that represents difference in gender

t-SNE might not show this difference, since it is a non-linear mapping

* Similarity function →

$$sim(e_w, e_{king} - e_{queen} + e_{woman})$$

$$sim(u, v) = \frac{u^T v}{\|u\|_2 + \|v\|_2} \quad ; \quad \|u\|_2 = \sqrt{\sum_{i=1}^{n} u_i^2}$$

- You could also use the square distance, it will work out but cosine works better.

* Embedding matrix →

Vocabulary size

word1 word2 ... word350 ... word9999 <UNK>

300 features

E

Orange vector

O_{6257}

$$\begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow 6257$$

$$E \cdot O_{6257} = e_{6257}$$

$$(300, 10000)(10000, 1) = (300, 1)$$

$$E \cdot O_j = e_j$$

Embedding for word j

- Objective → learn vectors e_j
- Initialize randomly, do gradient descent on it.
- Keras → embedding layer (Multiplication is inefficient).

* Learning word embeddings →

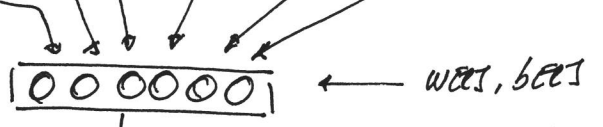
"I want a glass of orange ____" ← target.

O_{4543} O_{9665} O_1 O_{3852} O_{6163} O_{6257}

E E E E E E

e_{4543} e_{9665} e_1 e_{3852} e_{6163} e_{6257}

- What context?
- Necessity to find a context - target.



← $w[1], b[1]$

softmax ← $w[2], b[2]$

$\hat{y}$ prediction target.

* Word 2 Vec ⟶

- Skip-gram ⟶

"I want a glass of orange juice to go along with my cereal."

Context	Target
orange	juice
orange	glass
orange	any

↑ Pick randomly — ↑ Based on target, choose randomly.

- Model ⟶

Vocabulary size = 10.000

orange ⟶ target

$O_c \rightarrow E \rightarrow e_c \Longrightarrow$ softmax $\rightarrow \hat{y}$

$e_c = E \cdot O_c$

$$\Rightarrow p(\text{target} \mid \text{context}) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10000} e^{\theta_j^T e_c}}$$

$$\mathcal{L}(\hat{y}, y) = -\sum_{i=1}^{10000} y_i \log \hat{y}_i$$

This is really expensive.

For each on the words of the dictionary.

$$y_i = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \rightarrow \text{one hot vector.}$$

* Negative sampling →

Context	Word	Target
Orange	juice	1
Orange	King	0
Orange	book	0
Orange	the	0
Orange	of	0

Input. Output

→ K random samples from dictionary.

Small data sets → K = 5/20

Large data sets → K = 2/5

- Now it becomes a logistic regression problem →

$$P(y=1 / \text{Context}, \text{Target}) = \sigma(\theta_t^T e_c)$$

- It is now a 10 000 binary logistic regression classifier.

- Instead we only train K+1

- Instead of a 10K softmax → 10K binary classification for K+1.

- Selecting negative examples →

  * Sample according to the freq words, but you want to take the one not freq or do it uniformly → anymore.

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=1}^{10k} f(w_i)^{3/4}}$$

* Glove ≡ Global Word Vector ⟶

"I want a glass of ..."

X_{ij} ≡ # times that word i appears in context of word j"

j ≡ Context word ; i ≡ Target word.

$$\text{minimize}\left(\sum_{i=1}^{10000}\sum_{j=1}^{10000}\left[\theta_i^T e_j + b_i + b_j - \log X_{ij}\right]^2\right)$$

relationship between 2 words

$f(X_{ij}) \Rightarrow = 0$ if $X_{ij} = 0$ (Around $\log 0$)

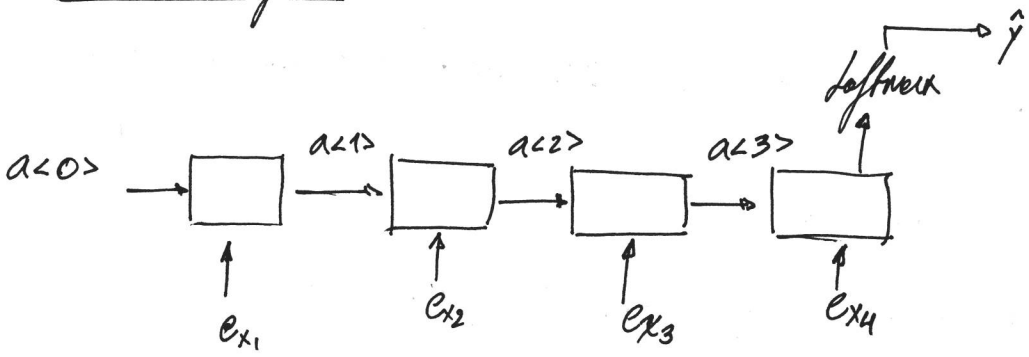
It can also be handle to treat fuy words such as

the, of, a, ...

- θ_i, e_j are simetric ; initialize random uniform distribution.

- θ_i, e_j gradient descent.

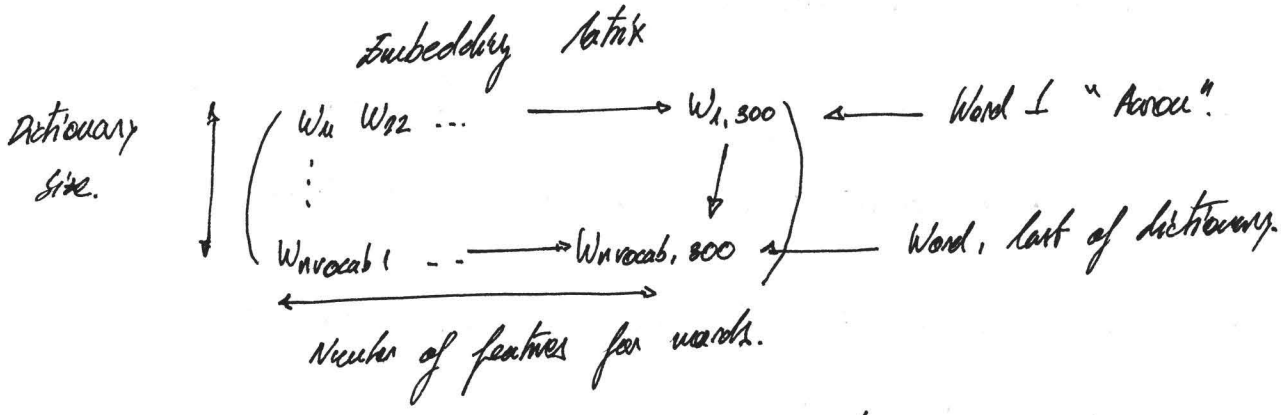
- Sentiment classification ⟶

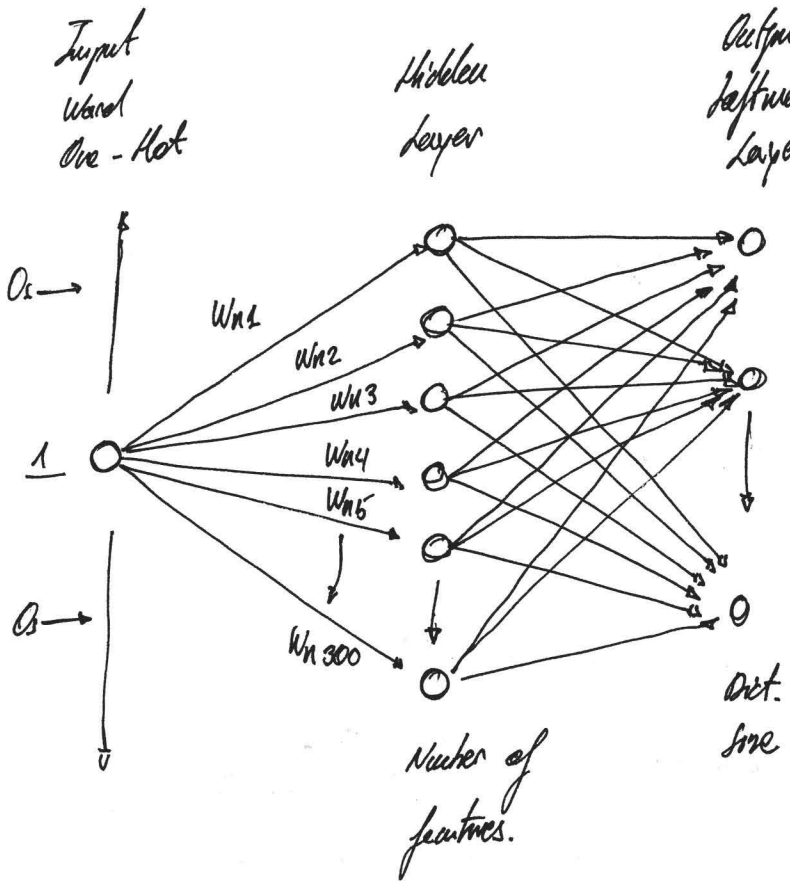


Many to one.

Word Embedding

* Skip-gram implementation ⟶
  - like a MLP with a softmax classifier
  - build semantic relationship between words based on context.





$$P(\text{Target} \mid \text{Context}) = \hat{y}_i = \frac{e^{\theta_c^T \cdot e_c}}{\sum_{j=1}^{10K} e^{\theta_j^T e_c}}$$

$$\text{Loss} = -\sum_{i=1}^{10K} y_i \log \hat{y}_i$$

* Per input only one row of the embedding matrix will be enabled ⟶
  No multiplication, just indexing.

* Calculating softmax and loss is really expensive. needs improvement.

Improvements on top →

* Word pairs & phrases →

  - "Boston Globe" as a word, not "Boston" and "Globe".

  - Count number of times two words appear together in the training set →

    * Equation decides to make phrases out of those pairs based on the phrase count and the individual occurances.

    * Favors phrases from infrequent words, to avoid things like "This is" "and the".

* Subsampling frequent words →

  - "the" example →

    1. Pairs like "fox", "the" doesn't tell much about the meaning of fox.

    2. Too many examples of "the" to learn.

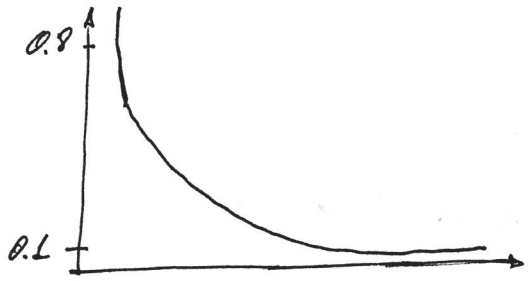
  - Sampling rate →

    * Probability to keep a word in the text. $\equiv P(w_i)$

    * $f(w_i) \equiv$ frequency of a word in the text. $= \frac{\text{counts}}{\text{total words text}}$

    * Parameter $t \equiv$ Parameter, also used to measure probability between 0,1.

$$P(w_i) = \left(\sqrt{\frac{f(w_i)}{t}} + 1\right)\frac{t}{f(w_i)} \quad ; \quad 1 - P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$



$P(w_i) = 1 \longrightarrow z(w_i) < 0.0026$

Only words that will be 0.26% of the text will be subsampled.

$P(w_i) = 0.5 \longrightarrow z(w_i) = 0.00746$

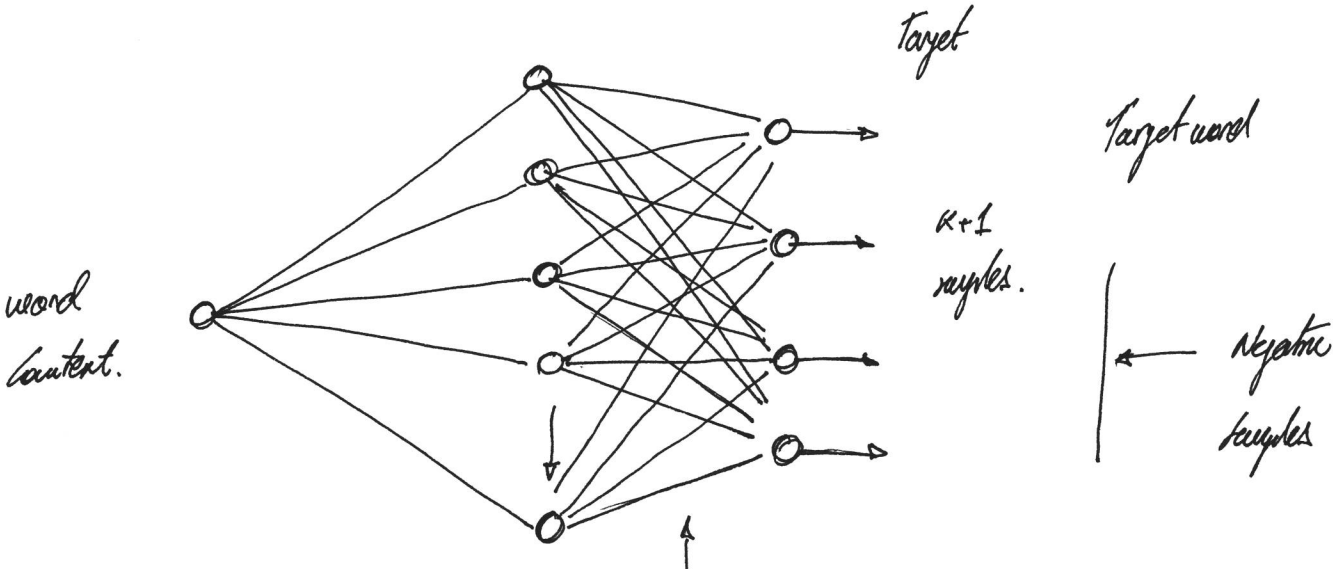
* Point $\longrightarrow$ Lower down the frequency difference for some really common words, like "the", "and", "if."

* No single word should be a large part of the corpus.

- Negative sampling $\longrightarrow$

* softmax step is really expensive $\longrightarrow \sum_{i=1}^{10k} e^{-\theta^T e_i}$; for billions of words.

* One approach $\longrightarrow$



Only the weights for these (K+1) word will be updated in the softmax layer.

* selecting negative samples →

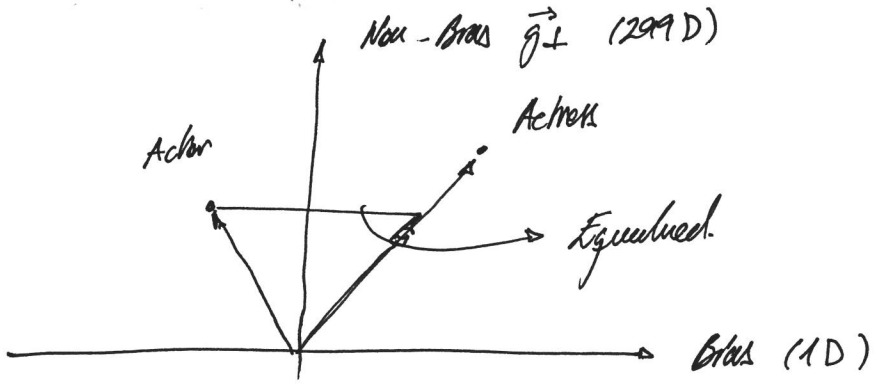
- Chosen only an "unigram distribution"
- Probability of selecting a negative word is proportional to its frequency

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=0}^{n} \left(f(w_i)^{3/4} \right)} \quad ; \quad f(w_i) \equiv \text{word frequency}$$

$3/4 \equiv$ Empirical factor.

* Debiasing word embedding ⟶

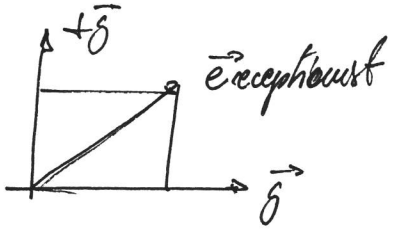
- Word embedding can reflect gender, ethnicity, age, sexual orientation and other biases of the text used for training.

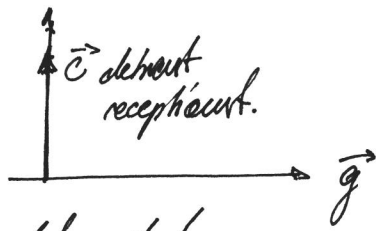


1. Identify bias vector.

2. Neutralize bias ⟶

For every word that is not definitional





$e^{bias\_component} = \frac{e \cdot g}{\|g\|_2^2} * g$ ← dot product.

$e^{debiased} = e - e^{bias\_component}$.

3. Equalize pairs ⟶ Vector Word only difference is bias

$\mu = \frac{e_{w1} + e_{w2}}{2}$; $\mu_B = \frac{\mu \cdot bias\_axis}{\|bias\_axis\|_2^2} * bias\_axis$; $\mu_\perp = \mu - \mu_B$

$e_{w1B} = \frac{e_{w1} * bias\_axis}{\|bias\_axis\|_2^2} * bias\_axis$; $e_{w2B} = \frac{e_{w2} * bias\_axis}{\|bias\_axis\|_2^2} * bias\_axis$

$$e_{w1B}^{corrected} = \sqrt{|1 - 1/\mu_1 u_2^2|} * \frac{e_{w1B} - \mu_B}{|e_{w1} - \mu_1 - \mu_B|} \; ; \; e_{w2B}^{corrected} = \sqrt{|1 - 1/\mu_1 u_2^2|} * \frac{e_{w2B} - \mu_B}{|e_{w2} - \mu_1 - \mu_B|}$$

$$e_1 = e_{w1B}^{corrected} + \mu_1 \; ; \; e_2 = e_{w2B}^{corrected} + \mu_1$$

Sequence to sequence →

- Image captioning →

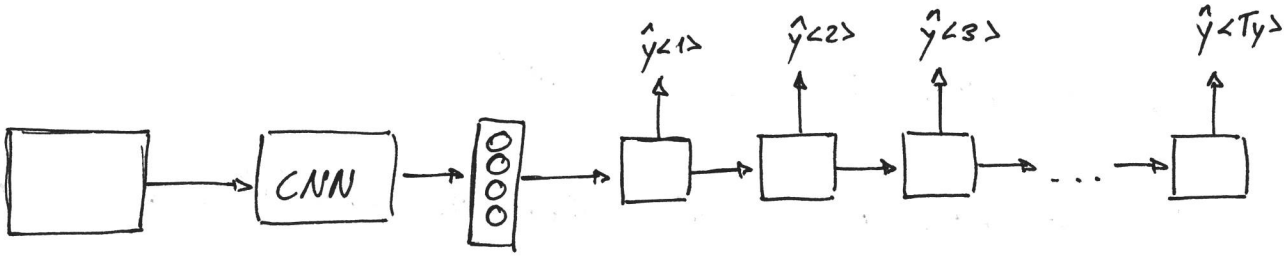


Image of cat sitting on a chair

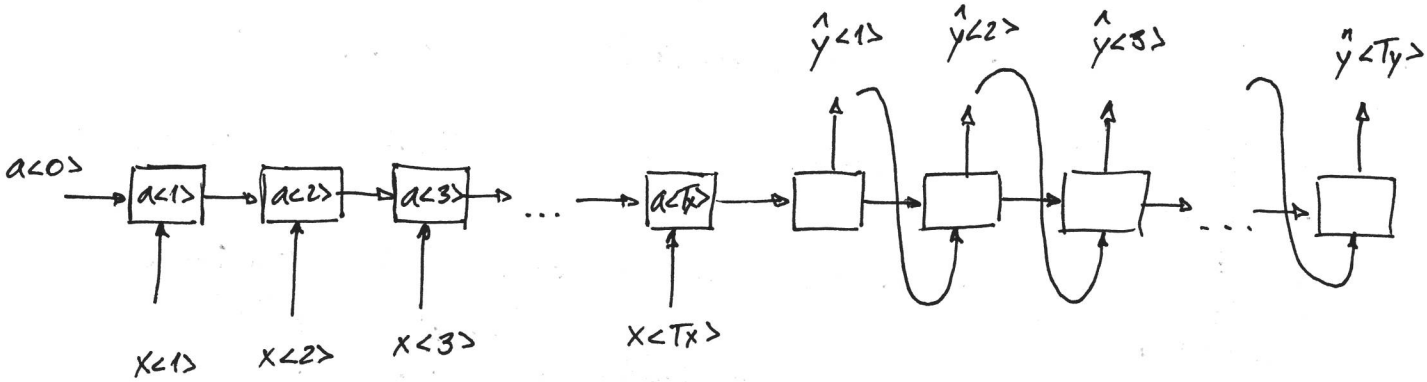
AlexNet shipped softmax

4096 Vector

Encoder / Decoder

- Machine translation →

Jane is visiting Africa in September.



Jane visite l'Afrique en septembre.

Encoder

Decoder ≡ Language model.

« The language model allows to estimate the probability of a sentence

$P(\hat{y}^{<1>}, \ldots, \hat{y}^{<T_y>} / x^{<1>}, \ldots, x^{<T_x>})$

Probability of the english sentence given the french.

* We don't sample in the language model, randomly.

* We don't either do greedy search because $\longrightarrow$

$$P(\hat{y}^{<1>}, \ldots, \hat{y}^{<T_y>} | X) \neq P(\hat{y}^{<1>}) \cdot P(\hat{y}^{<2>}) \ldots P(\hat{y}^{<T_y>})$$

"Jane is visiting Africa in September"

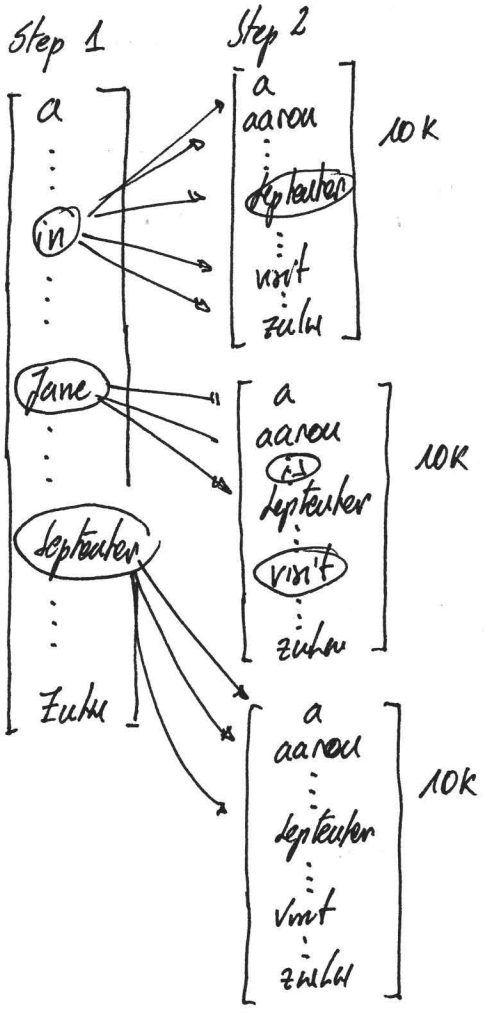
"Jane is going to be visiting Africa in September" $\longrightarrow$ Worse solutions.

$$P(\text{Jane is going} | X) > P(\text{Jane is visiting} | X)$$

- Beam search $\longrightarrow$

* B = Beam width parameter; algorithm will consider 3 need at a time

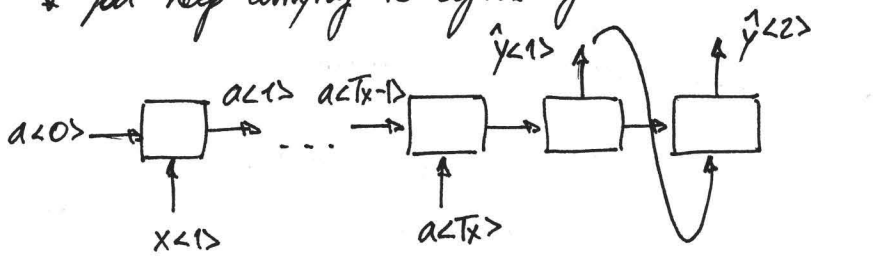
if B = 3. Objective $\longrightarrow$ $\underset{y^{<1>}, \ldots, y^{<T_y>}}{\arg\max}\ P(y^{<1>}, \ldots, y^{<T_y>} | X)$



* For all 30K combinations, pick top B. B = 3

$$P(y^{<1>}, y^{<2>} | X) = P(y^{<1>} | X)\, P(y^{<2>} | X, y^{<1>})$$

* You keep carrying B copies of the network $\longrightarrow$



$\hat{y}^{<1>}$ = in ; $\hat{y}^{<2>}$ = September

$\hat{y}^{<1>}$ = Jane ; $\hat{y}^{<2>}$ = is

$\hat{y}^{<1>}$ = ~~September~~ Jane ; $\hat{y}^{<2>}$ = visit

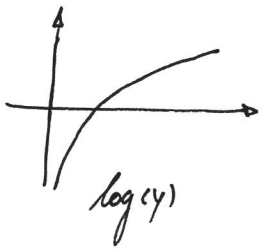
* For each step you carry B copies.

* length normalization ⟶

$$\arg\max_y \prod_{t=1}^{T_y} P(y^{<t>} / x, y^{<1>}, y^{<2>}, \ldots, y^{<t-1>})$$

↓ products of 0.x, result in really small numbers and round up at that low value can result in accuracy loss.

$$\arg\max_y \sum_{y=1}^{T_y} \log P(y^{<t>} / x, y^{<1>}, y^{<2>}, \ldots, y^{<t-1>})$$



- log is an strictly monotonic function, maximizing it should be the same as $\log(P(y/x))$

↓ With longer sentences the $\sum$ tends to be more negative, so it will tend to make shorter sentences.

$$\frac{1}{T_y^{\alpha}} \sum_{t=1}^{T_y} \log P(y^{<t>} / x, y^{<1>}, \ldots, y^{<t-1>}) = \text{score.}$$

↑ Normalizes by number of words in sentence, averaging them.

$\alpha \equiv$ Normalization factor, hyperparameter.

- Check different $T_y = 1, 2, 3, \ldots, 30.$ ⟶

  $B = 3$ ⟶ keep 3 possibilities per T_y

- Look at score for all output and keep highest

* Trade-of between B values →
  - Higher → Better, slower
  - Smaller → Worse, faster.

* Error Analysis →
  - Heuristic algorithm, doesn't guarantee best option.

~~Human reference~~ "Jane visite l'Afrique en septembre"

Human reference "Jane visits Africa in September" y^*

Algorithm "Jane visited Africe in September" $\hat{y}$

- RNN computes $P(y|x)$ (Encoder - decoder part)
- Compute $P(y^*|x)$ and $P(\hat{y}|x)$ →
  * $P(y^*|x) > P(\hat{y}|x)$ → Beam search, not performing well.
  * $P(y^*|x) < P(\hat{y}|x)$ → y^* is a better translation; RNN.

Human	Algorithm	$P(y^*\|x)$	$P(\hat{y}\|x)$	Cause
Sentence 1	T1	2×10^{-11}	1×10^{-11}	B
2	T2	↓	↓	R
3	T3			B
4	T4			B
↓				

% B. % R

- Blue Score ⟶

* Evaluates machine translation performance.

French "le chat est sur le tapis"

Reference 1 "The cat is on the mat"

Reference 2. "There is a cat on the mat"

MT Output: "The cat the cat on the mat"
(1 2 3 4 5 6) — Bi-gram

		Count	Clipped Count	
1, 3	the cat	2	1	⟶ # of times it appears on references 1,2.
2	cat the	1	0	
4	cat on	1	1	
5	on the	1	1	
6	the mat	1	1	

$$\text{Precision} = \frac{\sum_{bigram} \text{Count clip (bigram)}}{\sum_{bigram} \text{Count (bigram)}}$$

$P = 4/6$

$$P_n = \frac{\sum_{n\text{-gram} \in \hat{y}} \text{Count Clip (n-gram)}}{\sum_{n\text{-gram} \in \hat{y}} \text{Count (n-gram)}} \quad ; \quad P_1, P_2, \ldots, P_n = 1.0 \longrightarrow \text{When MT is equal to reference.}$$

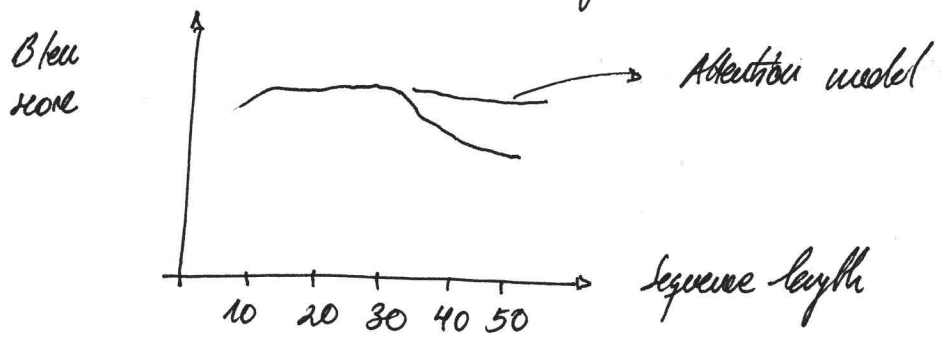
Combined Blue Score ⟶ $BP \cdot \exp\left(\frac{1}{n}\sum_{n=1}^{N} P_n\right)$

BP = Brevity Penalty, comes to get better translations on short sentences.

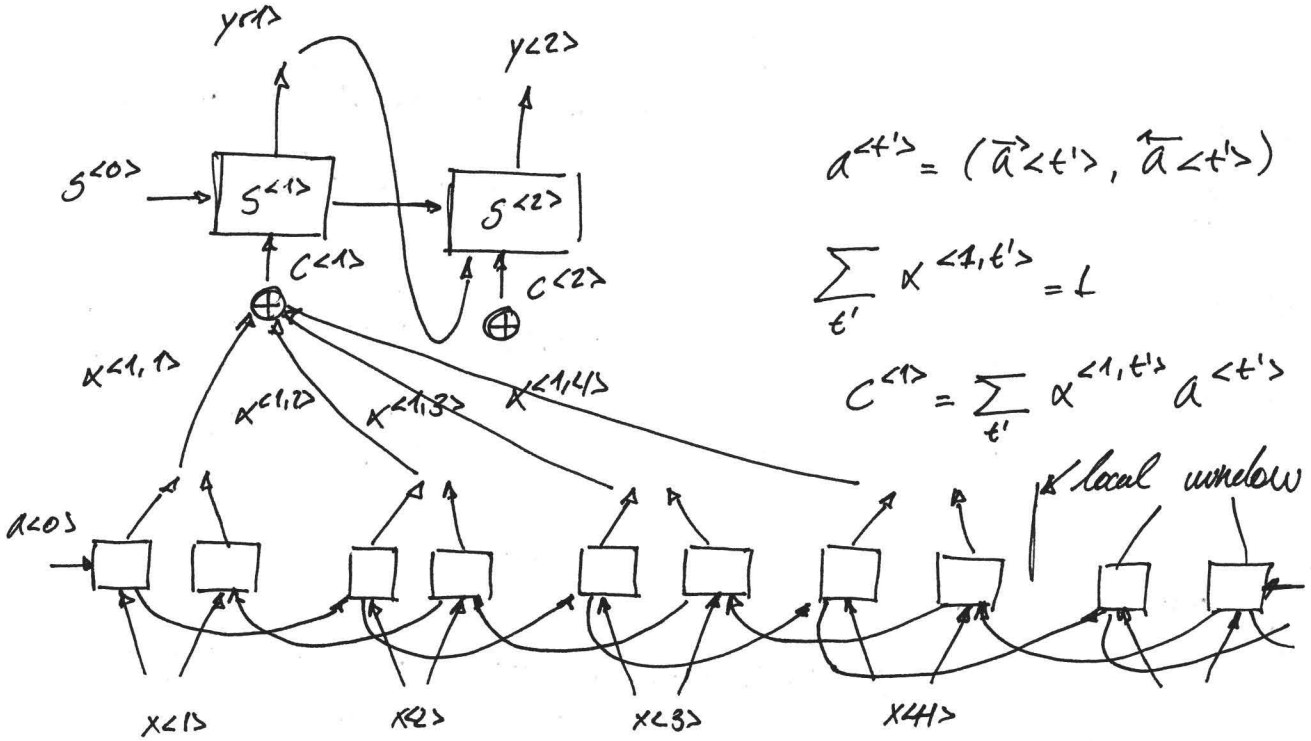
$$BP = \begin{cases} \exp(1 - \text{MT\_length} / \text{Ref\_length}) & \text{Otherwise} \\ 0 & \text{MT\_output\_length} > \text{reference\_length.} \end{cases}$$

- Attention model →

* Problem with long sequence of sentences.



* As a human wouldn't try to translate after memorizing 10-20 sentences, but piece by piece.



$$a^{<t'>} = (\vec{a}^{<t'>}, \overleftarrow{a}^{<t'>})$$

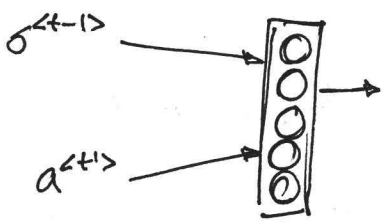
$$\sum_{t'} \alpha^{<1,t'>} = 1$$

$$c^{<1>} = \sum_{t'} \alpha^{<1,t'>} a^{<t'>}$$

$\alpha^{<t,t'>}$ ≡ Amount of attention that $y^{<t>}$ should pay to $a^{<t'>}$

$$\alpha^{<t,t'>} = \frac{\exp(e^{<t,t'>})}{\sum_{t'=1}^{T_x} \exp(e^{<t,t'>})}$$

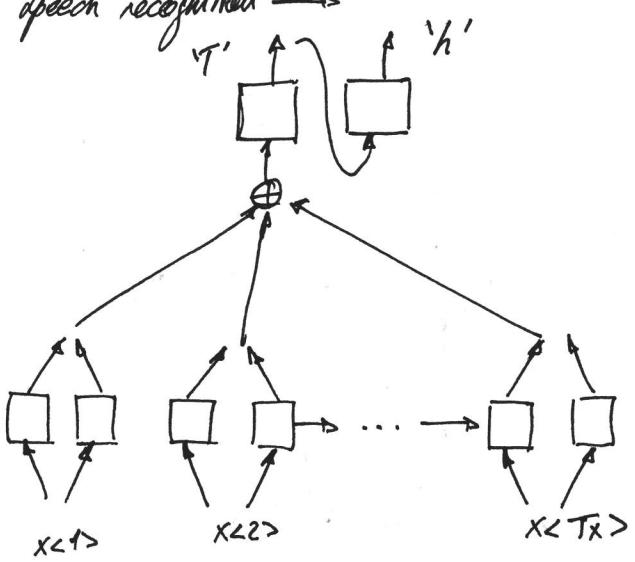
Softmax



$\frac{e^{<t,t'>}}{\alpha^{<t,t'>}}$

function to be learn by the network.

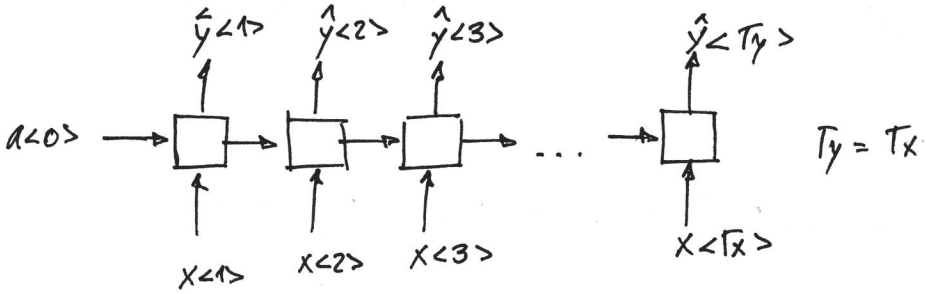
* speech recognition →



* Attention model

* the T_x time frames, e.g: $T_x = 1000$

* CTC cost for speech recognition →



$T_y = T_x$

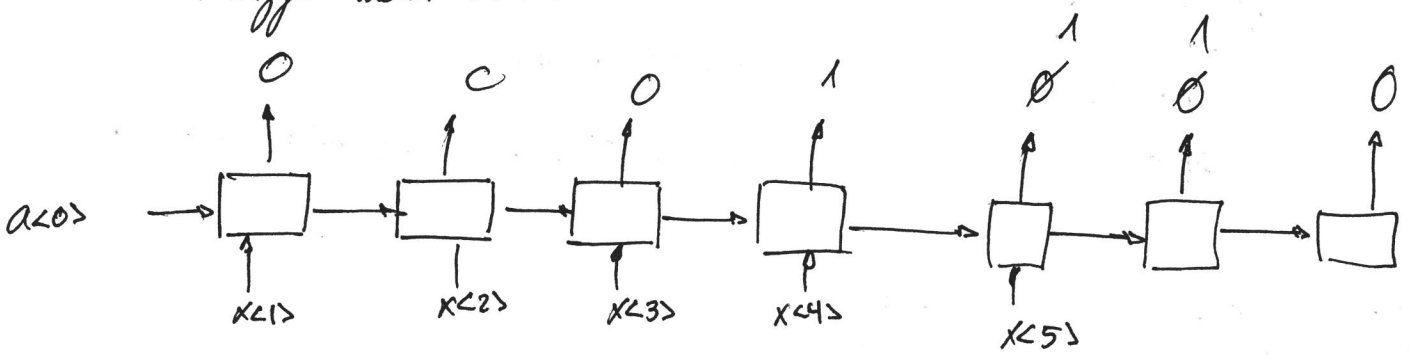
- Represent with regular RNN → Needs to be a bidirectional LSTM/GRU
- For 10sec, at $T_x = 10000$, more input than chars →

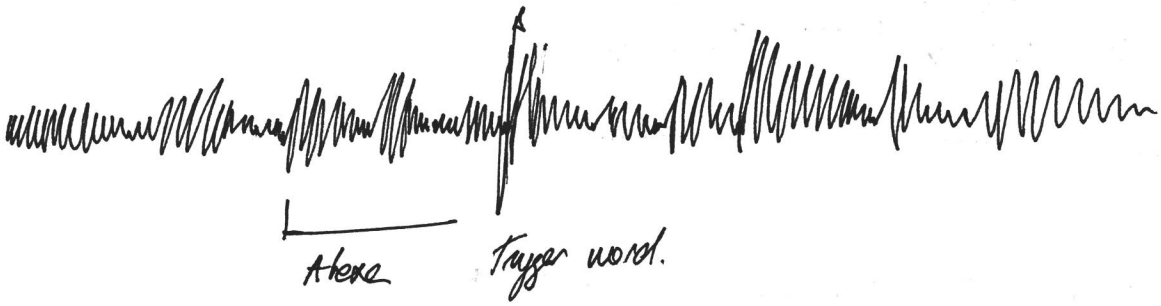
ttt_h_eee___LL___qq_UUUU_I__ ccccc_ KK

"the quick" ← special character blank

- Combined correct output, collapse characters not separated by blank.

* Trigger word detection →

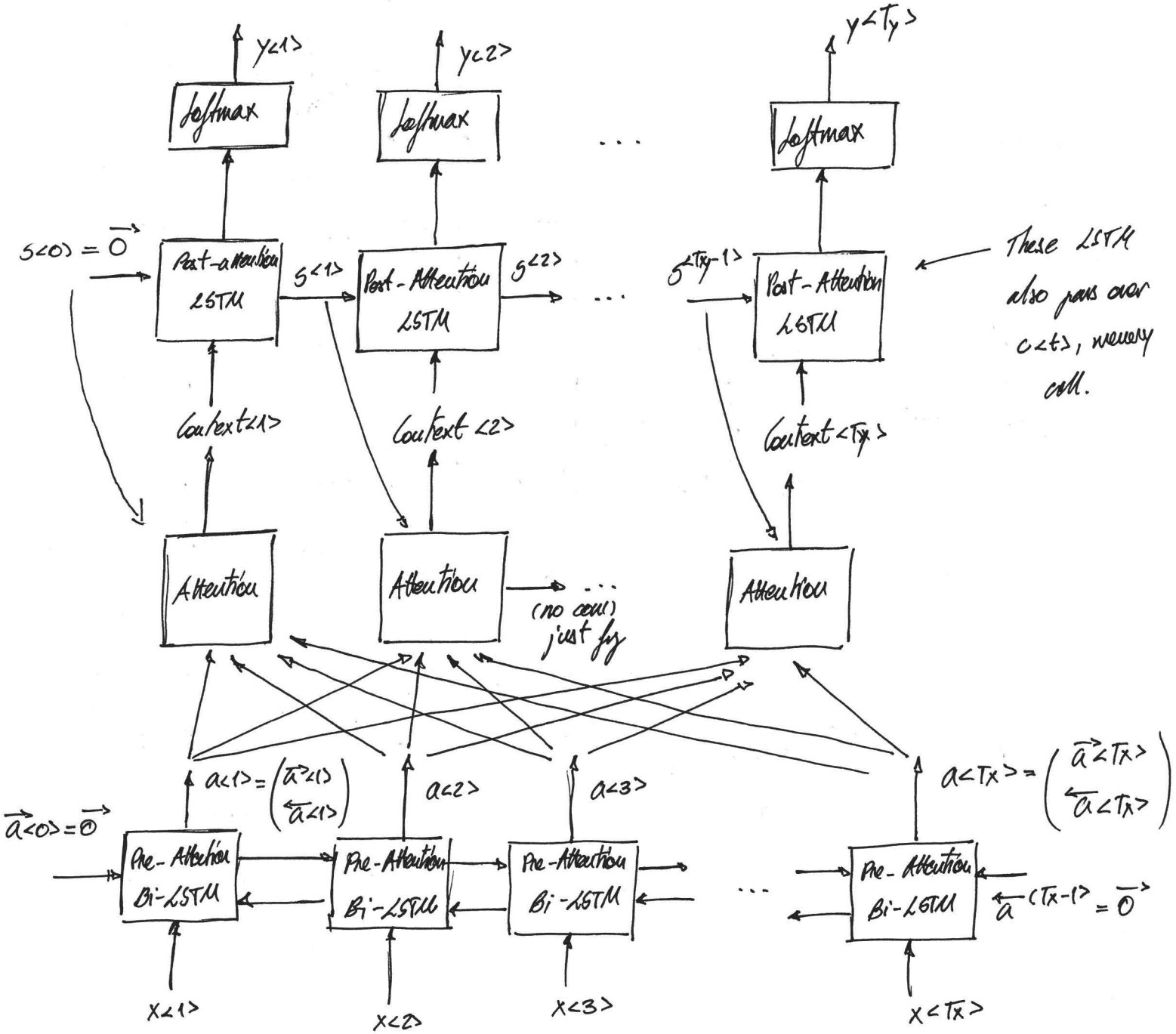




Here Trigger word.

- This sequence will work but the testing set is unbalanced a lot of 0s.
- You can force the output to 1 after a few cycles of the trigger word.

Attention Model

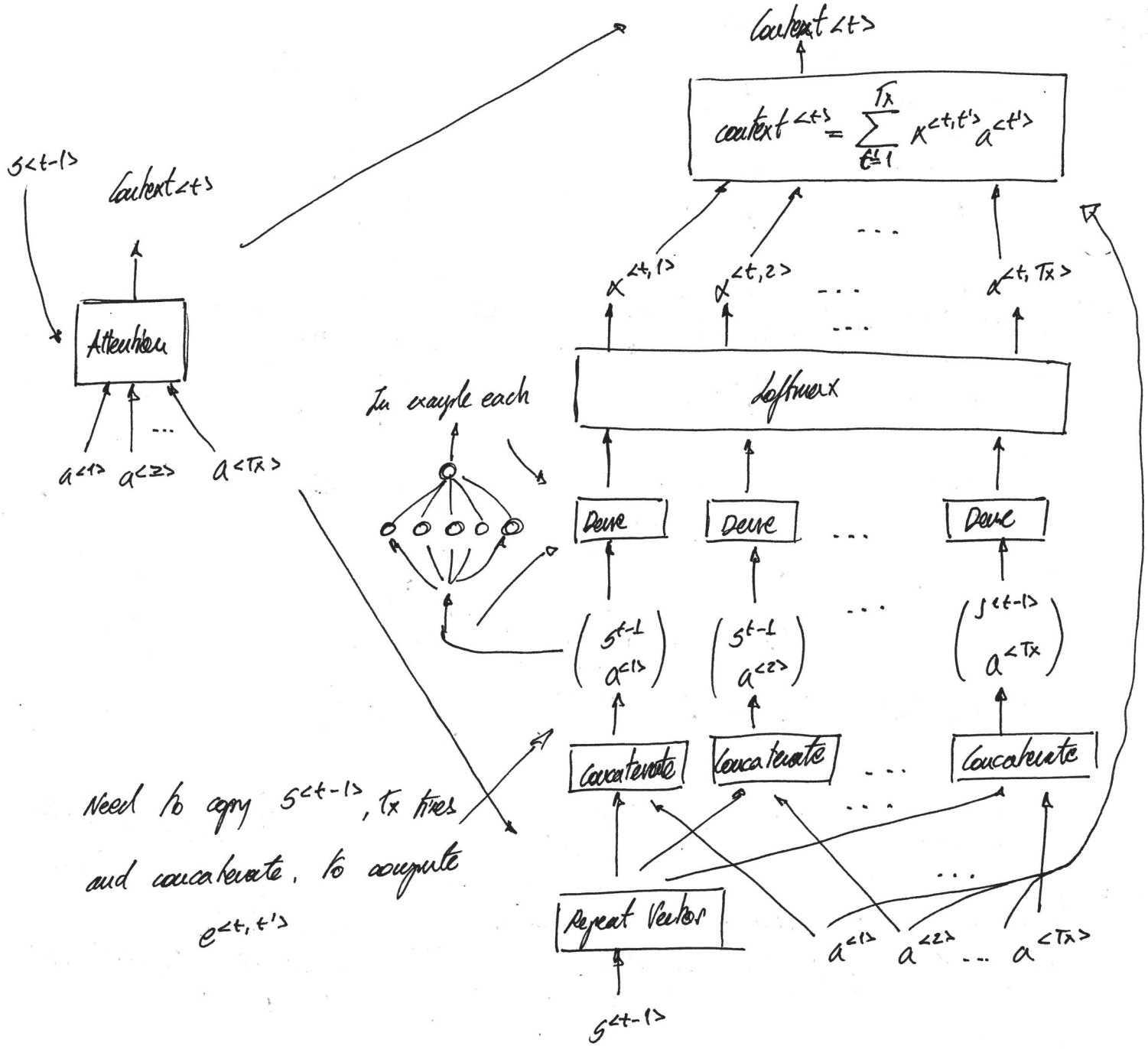


+ Concrete ⟶

- Post-attention LSTM also passes through the memory cell $c^{<t>}$, besides $s^{<t>}$

- Unlike for other purposes (language modeling), in here $y^{<t-1>}$ is not usually seen as an input to the next time step. There no high correlation between previous output and next.

- $a^{<t>} = [\vec{a}^{<t>}, \overleftarrow{a}^{<t>}]$, concatenation of bi-dir activations.


Context<t>
context<t> = Σ_{t'=1}^{Tx} α<t,t'> a<t'>
α<t,1>
α<t,2>
α<t,Tx>
Attention
s<t-1>
Context<t>
a<1> a<2> ... a<Tx>
In example each
LeftToRight
Dense
Dense
Dense
(s<t-1> a<1>)
(s<t-1> a<2>)
(s<t-1> a<Tx>)
Concatenate
Concatenate
Concatenate
Need to copy s<t-1>, Tx times
and concatenate, to compute
e<t,t'>
Repeat Vector
s<t-1>
a<1> a<2> ... a<Tx>