LSTM

RNN input at time t is soon forgotten.

$$\vec{y}^{\,t+100}\left(\vec{x}^{\,t+100}, \vec{h}^{\,t+99}\left(\vec{x}^{\,t+99}\right)\right)$$

Theoretically, $\vec{y}^{\,t+100}$ should depend on $\vec{x}^{\,t}$.

Practically, it does not.

There is no long-term dependency of outputs on previous inputs. LSTM

Because of W''.

$$\vec{y}^{\,t+100}\left(\vec{x}^{\,t+100}, \vec{h}^{\,t+99}\left(\vec{x}^{\,t+99}\right)\right)$$

left(left(right) right) — RNN

$$\vec{h}^{\,t} = \tanh\left(W''\vec{h}^{\,t-1} + W^{o}_{x}\vec{x}^{\,t} + b\right)$$

Identity

$$= W''\left(W''\vec{h}^{\,t-2} + W^{o}_{x}\vec{x}^{\,t-1}\right) + W^{o}_{x}\vec{x}^{\,t}$$

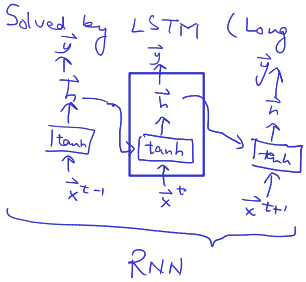
$$\vdots$$

$$= (W'')^{t}\ldots + \ldots + (W'')^{t}W^{o}_{x}\vec{x}^{\,0}$$

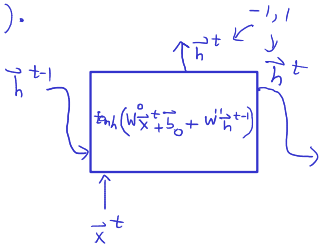
vector

Magnitude depends on largest eval. of W''

Solved by LSTM (Long Short-term Memory).



RNN



RNN "Cell"

$M\vec{x} + \vec{b}$ = Affine transformation

$M\vec{x}$ = linear transformation of $\vec{x}$

LSTM Cell

$$\vec{v}^{\,t} = \begin{bmatrix} \vec{h}^{\,t-1} \\ \vec{x}^{\,t} \end{bmatrix}$$

$(M+D)\times 1$

① $\sigma_f\left(W_f\vec{v}^{\,t} + \vec{b}_f\right)$ — $M\times(M+D)$, $(M+D)\times 1$, $M\times 1$ → $M\times 1$

$$\vec{f}^{\,t}$$

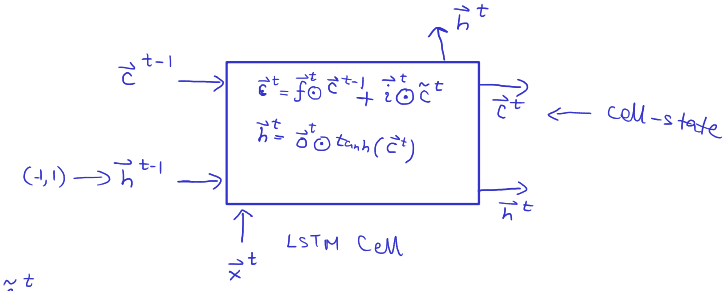
$M\times 1$ vector with value between 0 and 1

② $\sigma\left(W_i\vec{v}^{\,t} + \vec{b}_i\right)$ → $\vec{i}^{\,t}$ ($M\times 1$, $0-1$)

③ $\sigma\left(W_o\vec{v}^{\,t} + \vec{b}_o\right)$ → $\vec{o}^{\,t}$ ($M\times 1$, $0-1$)

④ $\tanh\left(W_c\vec{v}^{\,t} + \vec{b}_c\right)$ → $\tilde{c}^{\,t}$ ($M\times 1$, $-1,1$, $\tanh_c$)

$\tilde{c}^{\,t}$ Potentially new cell state



$\odot$ = Hadamard product = element-wise multiplication

Roles of $\vec{f}^{\,t}, \vec{i}^{\,t}, \vec{o}^{\,t}$ and $\tilde{c}^{\,t}$

Between 0 and 1

$$\vec{f}^{\,t}_{j} = \begin{cases} 0 \Rightarrow \vec{c}^{\,t-1}_{j} \text{ will be removed/forgotten in the new cell state } \vec{c}^{\,t}_{j} \\ 1 \Rightarrow \text{——— retained fully ———} \end{cases}$$

Forget gate on $\vec{c}^{\,t-1}$.

$$\vec{i}^{\,t}_{j} = \begin{cases} 0 \Rightarrow \text{no new information will be add to cell state } \vec{c}^{\,t}_{j} \\ 1 \Rightarrow \text{add the potential cell state } \tilde{c}^{\,t}_{j} \text{ completely (irrespective of forgetness level)} \end{cases}$$

Input gate on $\tilde{c}^{\,t}$.

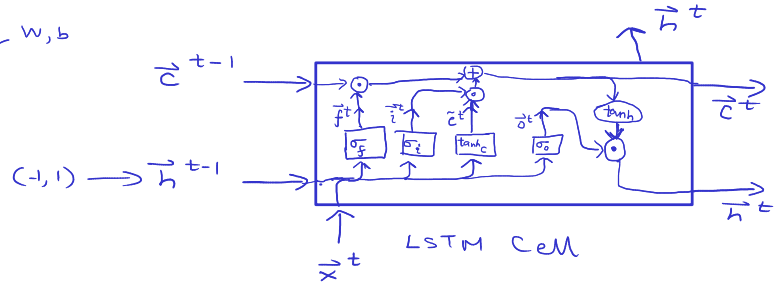
$$\vec{o}^{\,t}_{j} = \begin{cases} 0 \Rightarrow \text{completely hide cell state } \vec{c}^{\,t}_{j} \\ 1 \Rightarrow \text{set } \vec{h}^{\,t}_{j} = \tanh\left(\vec{c}^{\,t}_{j}\right). \text{ Cell state } \vec{c}^{\,t}_{j} \text{ is fully exposed in space and time.} \end{cases}$$

Output gate on $\vec{c}^{\,t}$.

Deep Learning: Initialization; Batch norm; $F(x)+x$ (0); Flow of info. ensure.

□ = parameterised transformation (W, b)

⬭ = operation



$\vec{c}^{\,t} = \vec{f}^{\,t}\odot\vec{c}^{\,t-1} + \vec{i}^{\,t}\odot\tilde{c}^{\,t}$

$\vec{h}^{\,t} = \vec{o}^{\,t}\odot\tanh(\vec{c}^{\,t})$

- Notice that depending on $\vec{f}^{\,t}$ and $\vec{i}^{\,t}$, $\vec{c}^{\,t}$ can be an exact copy of previous cell state $\vec{c}^{\,t-1}$. "Flowing cell state".
- If $\vec{c}^{\,t}_{i} = 1$, it can imply that a bracket is currently open. It can remain 1 for a long time until a closing bracket is observed at input.
- This gives long term memory.