



RNN and LSTM

(Oct 12, 2016)

YANG Jiancheng



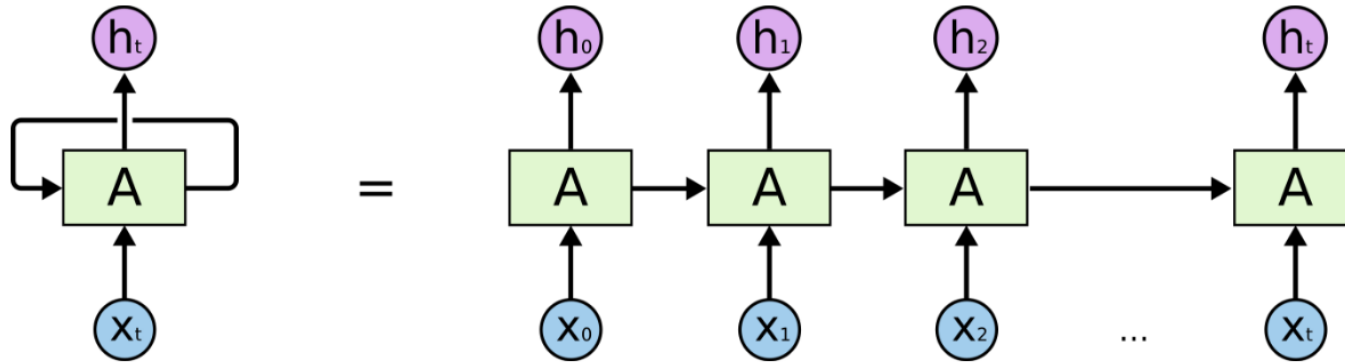
Outline

- **I. Vanilla RNN**
- **II. LSTM**
- **III. GRU and Other Structures**



- I. Vanilla RNN

GREAT Intro: Understanding LSTM Networks

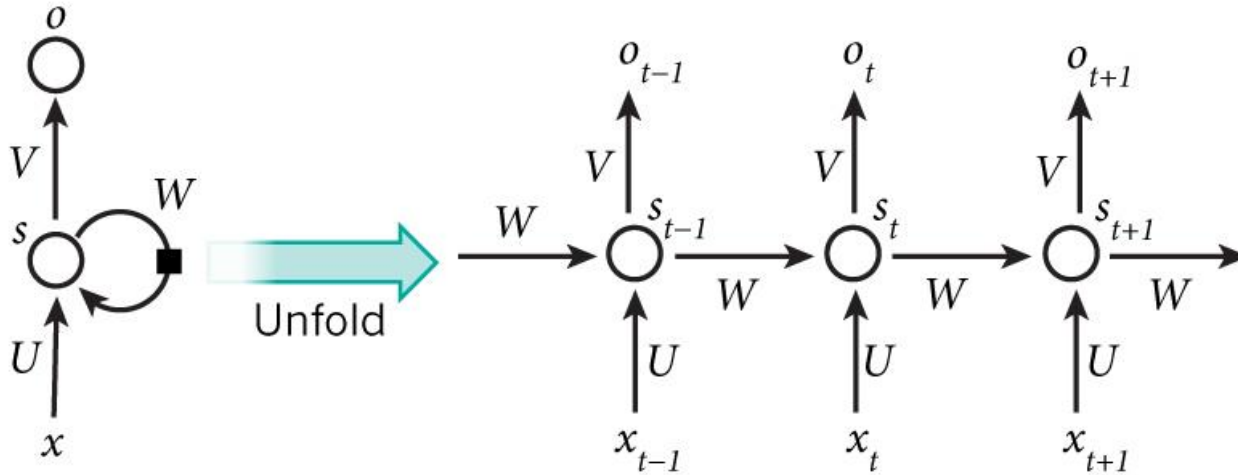


An unrolled recurrent neural network.

In theory, RNNs are absolutely capable of handling such “long-term dependencies.” A human could carefully pick parameters for them to solve toy problems of this form. Sadly, **in practice**, RNNs don’t seem to be able to **learn** them.



- **I. Vanilla RNN**



$$s_t = \tanh(Ux_t + Ws_{t-1})$$

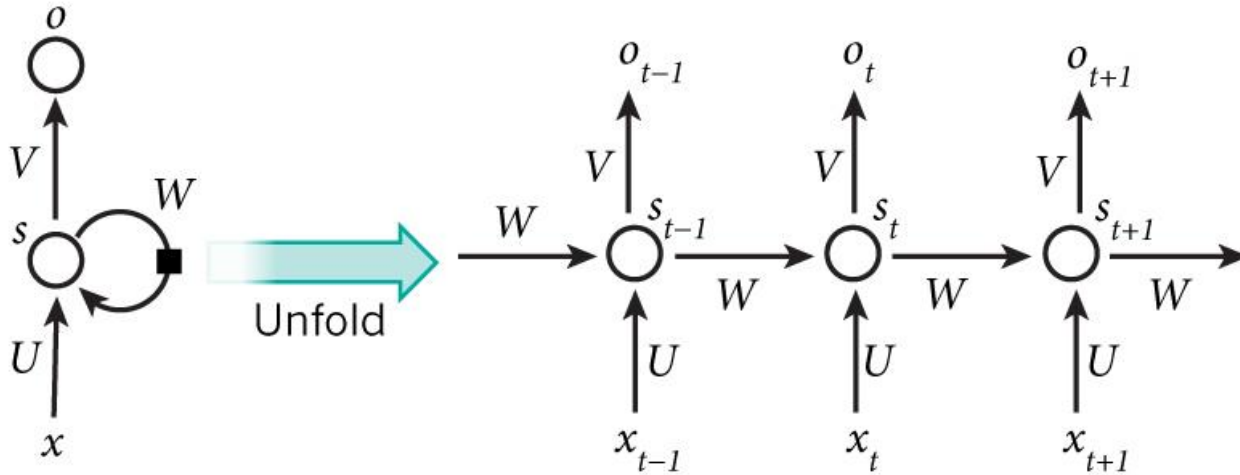
$$\hat{y}_t = \text{softmax}(Vs_t)$$

[WILDML](#) has a series of articles to introduce RNN (4 articles, 2 GitHub repos).



• I. Vanilla RNN

• Back Prop Through Time (BPTT)



$$\begin{aligned}
 \frac{\partial E_3}{\partial V} &= \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial V} \\
 &= \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial z_3} \frac{\partial z_3}{\partial V} \\
 &= (\hat{y}_3 - y_3) \otimes s_3
 \end{aligned}$$

$$\frac{\partial E_3}{\partial W} = \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \frac{\partial s_3}{\partial W}$$

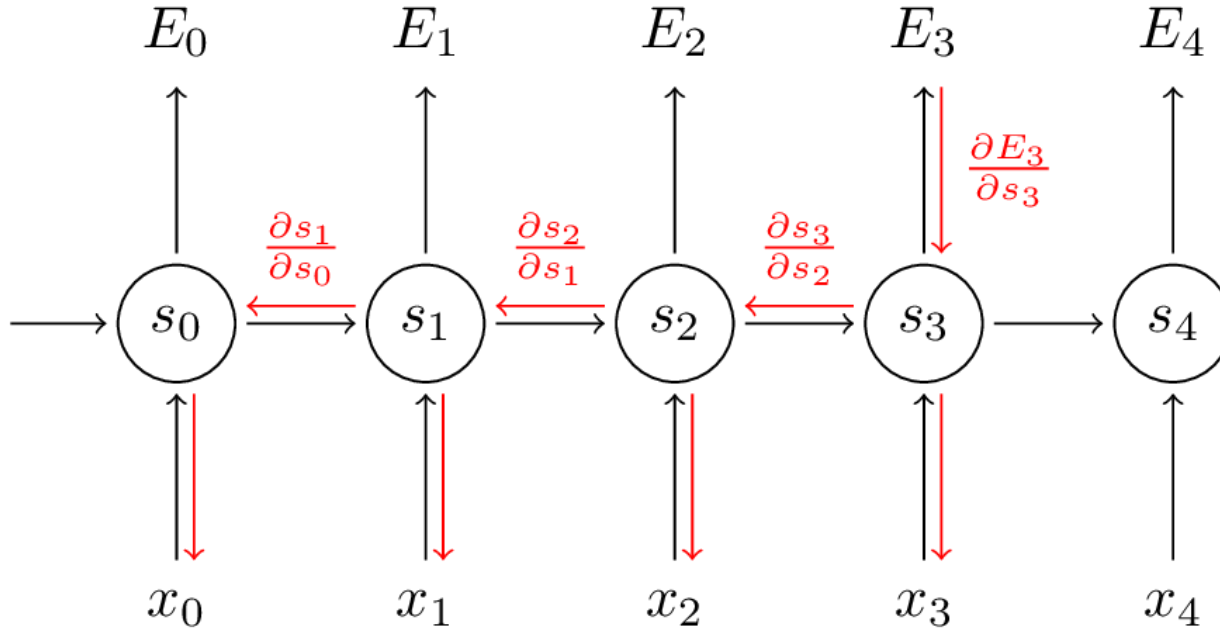
$$s_3 = \tanh(Ux_t + Ws_2)$$

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \frac{\partial s_3}{\partial s_k} \frac{\partial s_k}{\partial W}$$



• I. Vanilla RNN

- Back Prop Through Time (BPTT)



$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \frac{\partial s_3}{\partial s_k} \frac{\partial s_k}{\partial W}$$

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \left(\prod_{j=k+1}^3 \frac{\partial s_j}{\partial s_{j-1}} \right) \frac{\partial s_k}{\partial W}$$

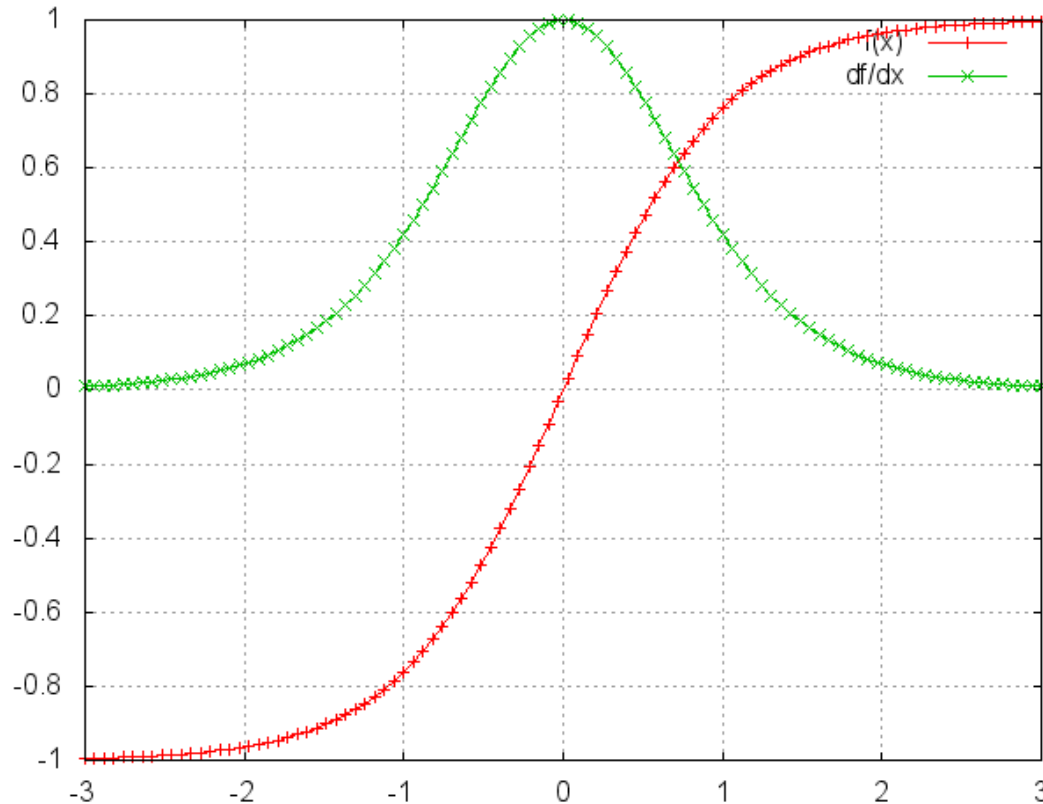


• I. Vanilla RNN

- Gradient Vanishing Problem

RNNs tend to be very deep

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \left(\prod_{j=k+1}^3 \frac{\partial s_j}{\partial s_{j-1}} \right) \frac{\partial s_k}{\partial W}$$

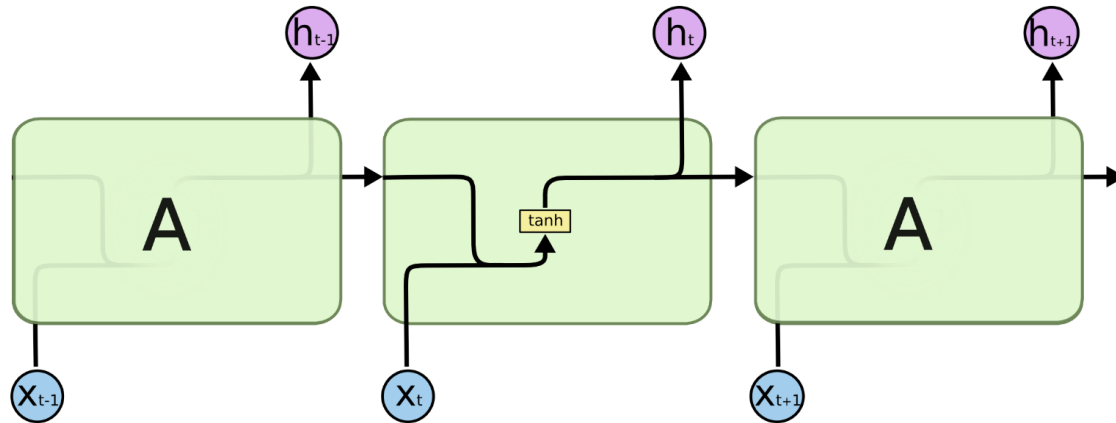


tanh and derivative. Source: <http://nn.readthedocs.org/en/rtd/transfer/>

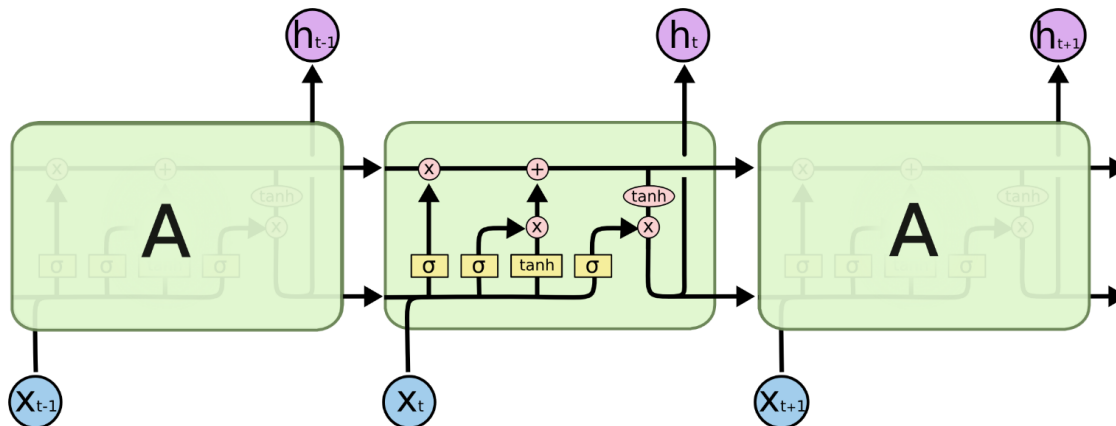


• II. LSTM

- Differences of LSTM and Vanilla RNN



The repeating module in a standard RNN contains a single layer.

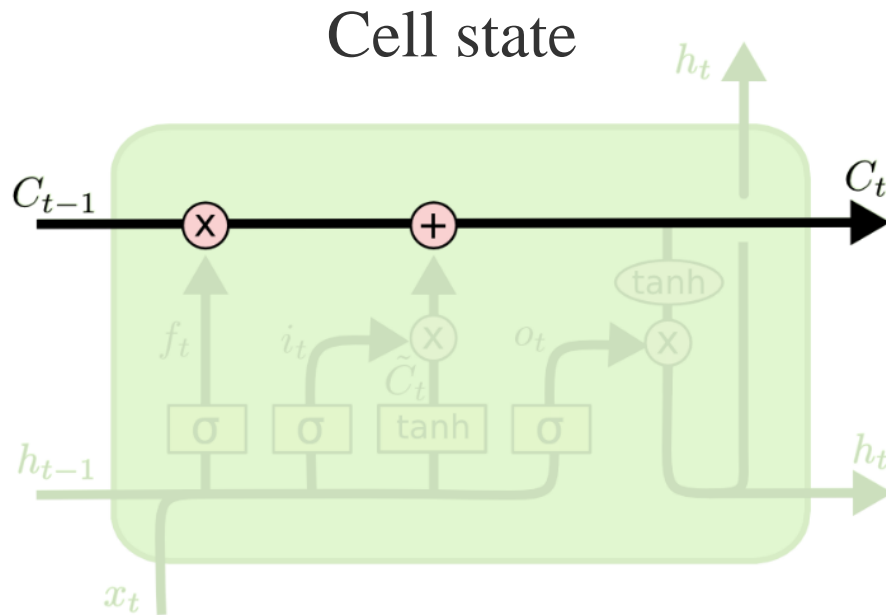


The repeating module in an LSTM contains four interacting layers.

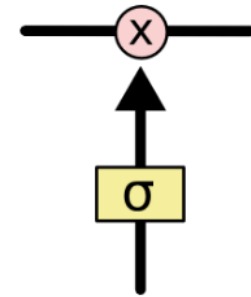


• II. LSTM

- Core Idea Behind LSTMs



Gates





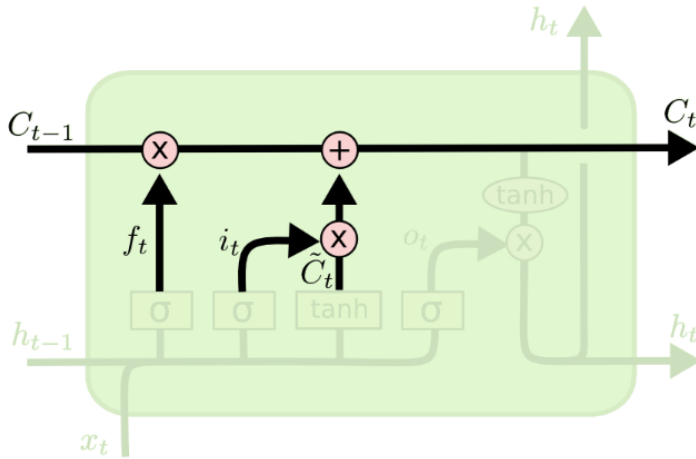
The diagram illustrates the internal structure of an LSTM cell. It shows the flow of information through various gates and operations. The inputs are x_t and the hidden state from the previous time step h_{t-1} . The cell state from the previous time step is C_{t-1} . The diagram shows the calculation of the forget gate output f_t (using a sigmoid function σ), the input gate output i_t (using a sigmoid function σ), and the candidate cell state \tilde{C}_t (using a tanh function). These are then combined to update the cell state C_t using element-wise multiplication (\otimes) and addition ($+$). The output gate o_t (using a sigmoid function σ) is also shown, which will be used to calculate the final hidden state h_t via a tanh activation.

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

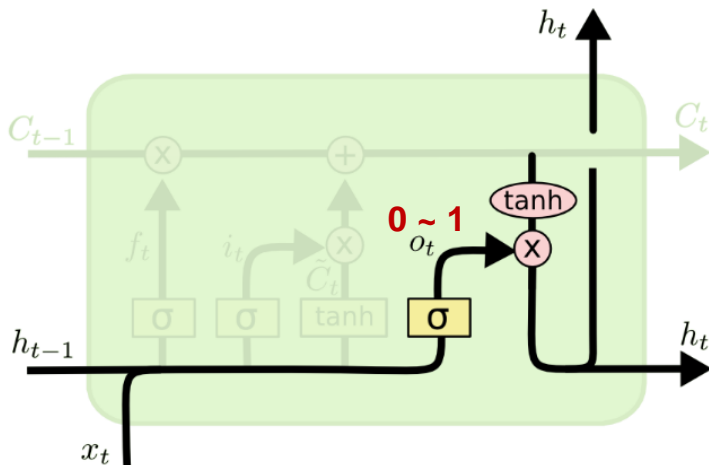


• II. LSTM

• Step-by-Step Walk Through



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



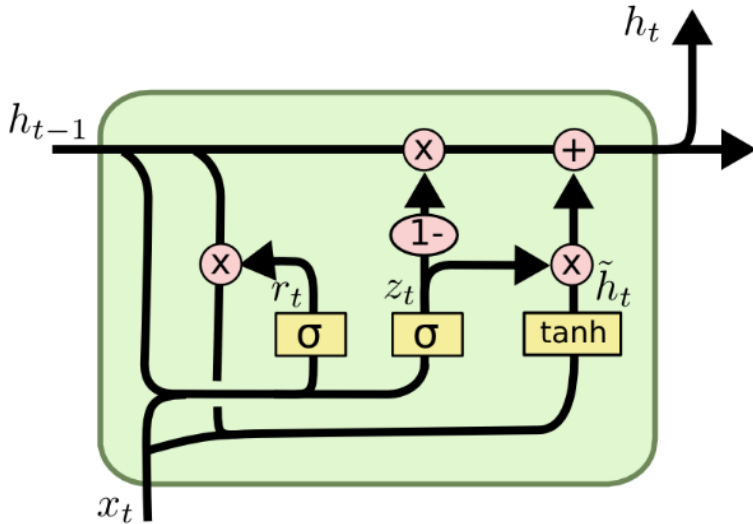
$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh (C_t)$$



• III. GRU and other structures

• Gated Recurrent Unit (GRU)



$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

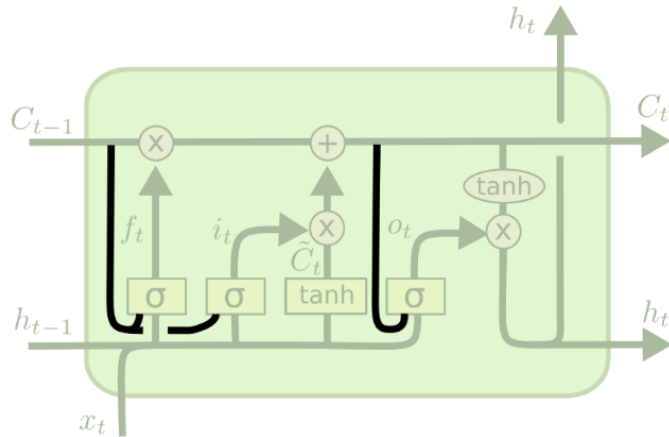
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

- Combines the forget and input gates into a single “update gate.”
- Merges the cell state and hidden state
- Other changes



• III. GRU and other structures

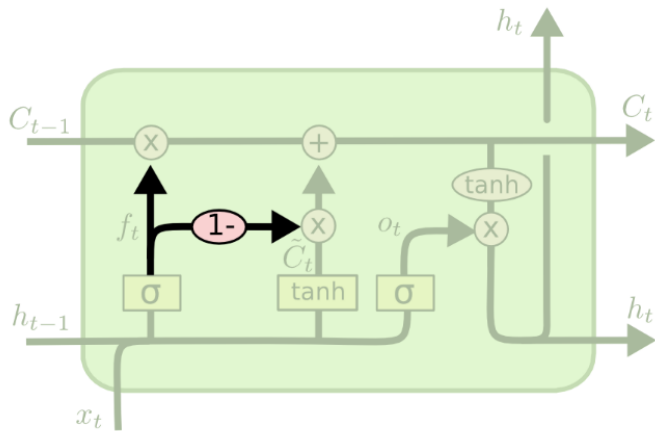
• Variants on Long Short Term Memory



$$f_t = \sigma(W_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [C_{t-1}, h_{t-1}, x_t] + b_i)$$

$$o_t = \sigma(W_o \cdot [C_t, h_{t-1}, x_t] + b_o)$$



$$C_t = f_t * C_{t-1} + (1 - f_t) * \tilde{C}_t$$

[Greff, et al. \(2015\)](#) do a nice comparison of popular variants, finding that they're all about the same.



Bibliography

- [1] [Understanding LSTM Networks](#)
- [2] [Back Propagation Through Time and Vanishing Gradients](#)



Thanks for listening!

