Lecture 7: LSTMs

RNNs on Long-Term Dependencies

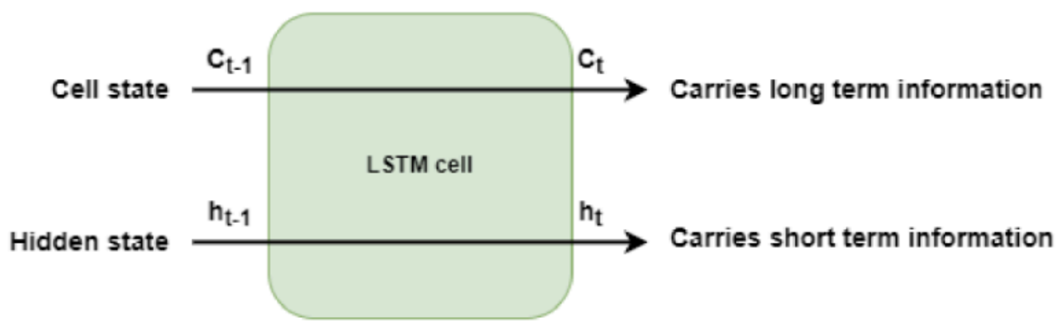
Long-term dependencies occur in text when understanding the context of a token requires us to refer to tokens farther away in the text. RNNs cannot capture these long-term dependencies from the final encoder's hidden state due to vanishing & exploding gradients

Truncated backpropagation through time

In truncated backpropagation through time, the input sequence is split into chunks of specified windows and backpropagation is performed on them. This results in faster training but long-term dependencies are not learnt if they're not in the same window.

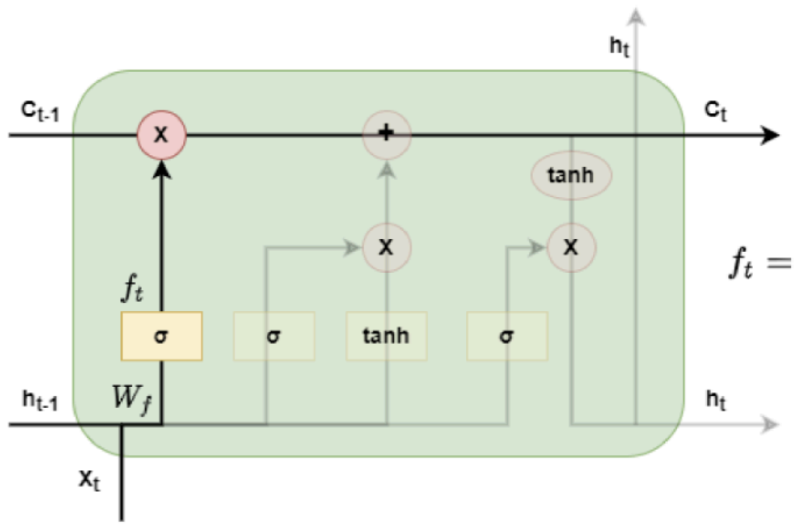
Long Short-Term Memory Networks

LSTM units carry information across different LSTM cells by **hidden state** and **cell state.**



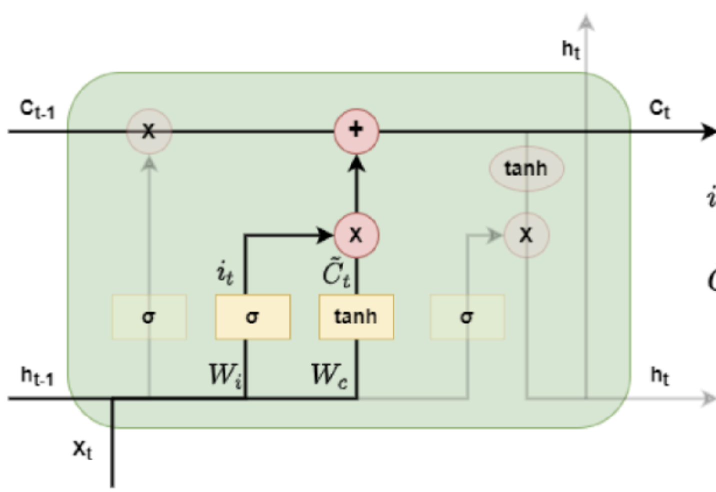
Propagation within an LSTM cell can be broken down into 3 parts

- Forget gate: responsible for deciding what information to forget in cell state, based on current input and hidden states.



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

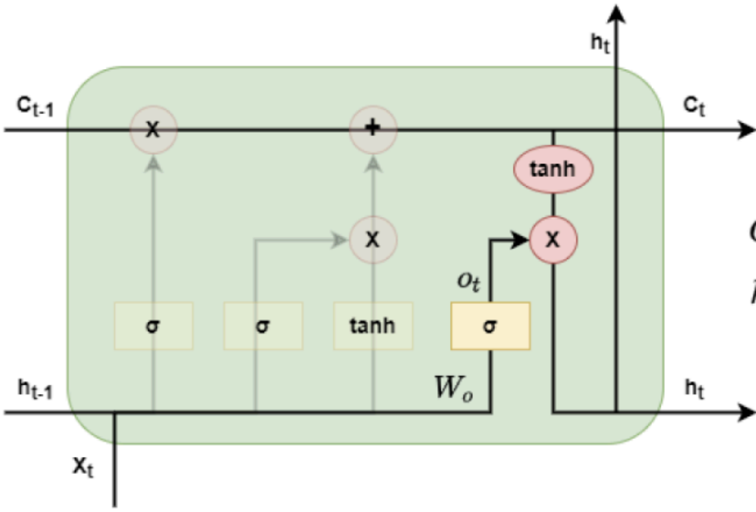
- Input gate: decides what information to add to the cell state, from the input and hidden states.



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

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

- Output gate: decides what relevant information has to be passed from cell state to the next hidden state



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

$$h_t = O_t * tanh(C_t)$$

- The solution to the vanishing & exploding gradients problem: unlike RNNs, gradients of LSTM do not have a fixed pattern and can take any positive value at any time step, thus mitigating the Vanishing and Exploding gradients problem

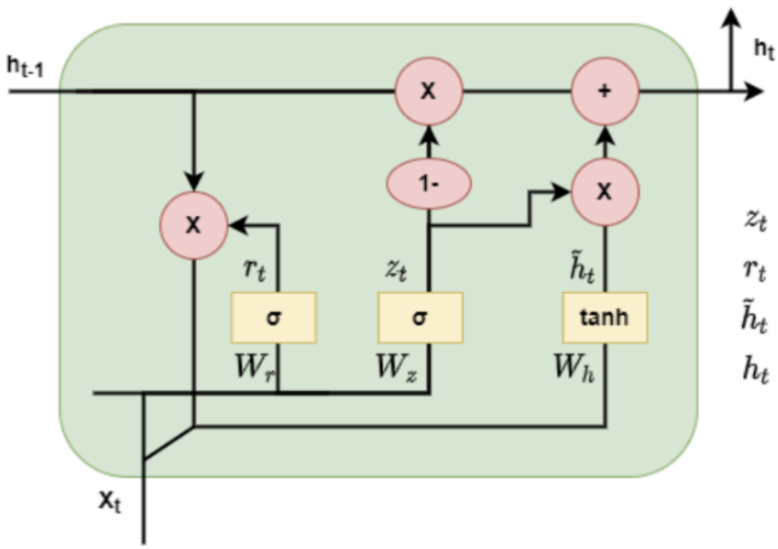
- For example, consider gradients of cell state:

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

LSTM Varients: GRU

GRU architecture has two gates and a single hidden state

- Reset gate: responsible for deciding which information from the hidden state to remove
- Update gate: responsible for deciding which information from inputs to add to the hidden state



$$
\begin{aligned}
z_t &= \sigma(W_z * [h_{t-1}, x_t]) \\
r_t &= \sigma(W_r * [h_{t-1}, x_t]) \\
\tilde{h}_t &= tanh(W_h * [h_{t-1}, x_t]) \\
h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t
\end{aligned}
$$