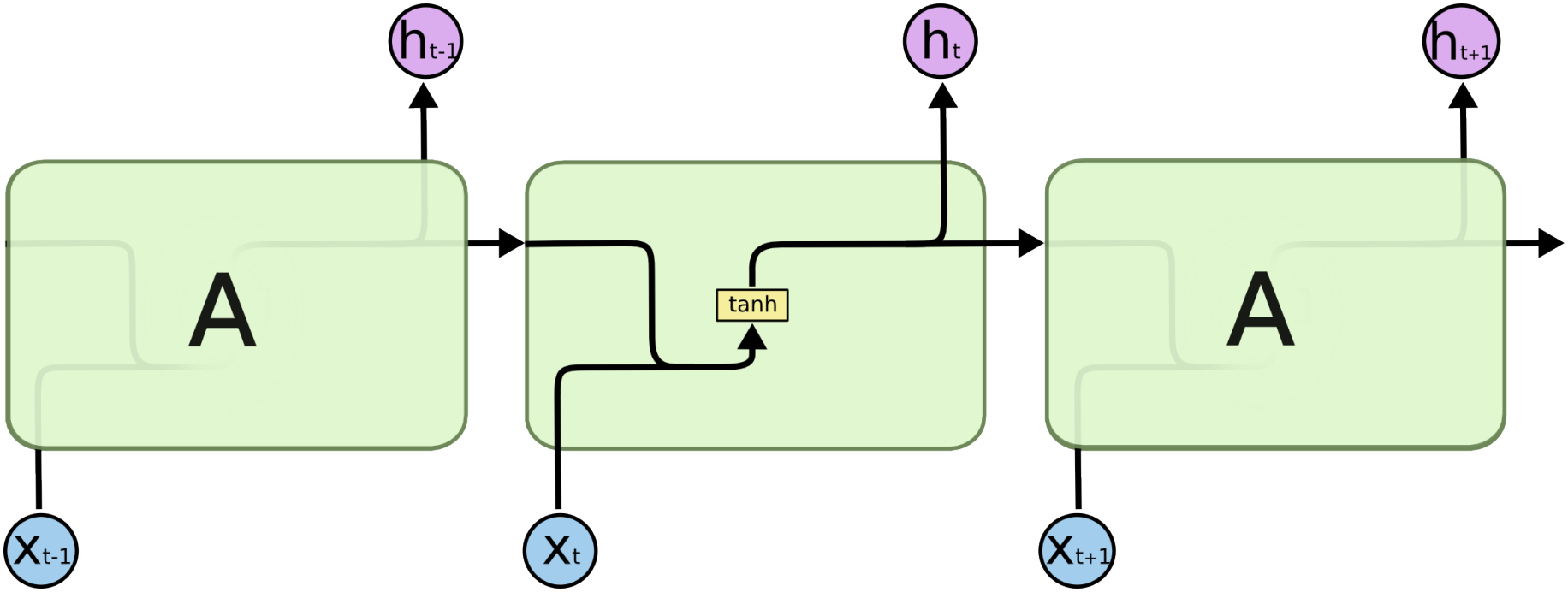
Recurrent Neural Nets
Long Short-Term Memory Units

Thang Vu

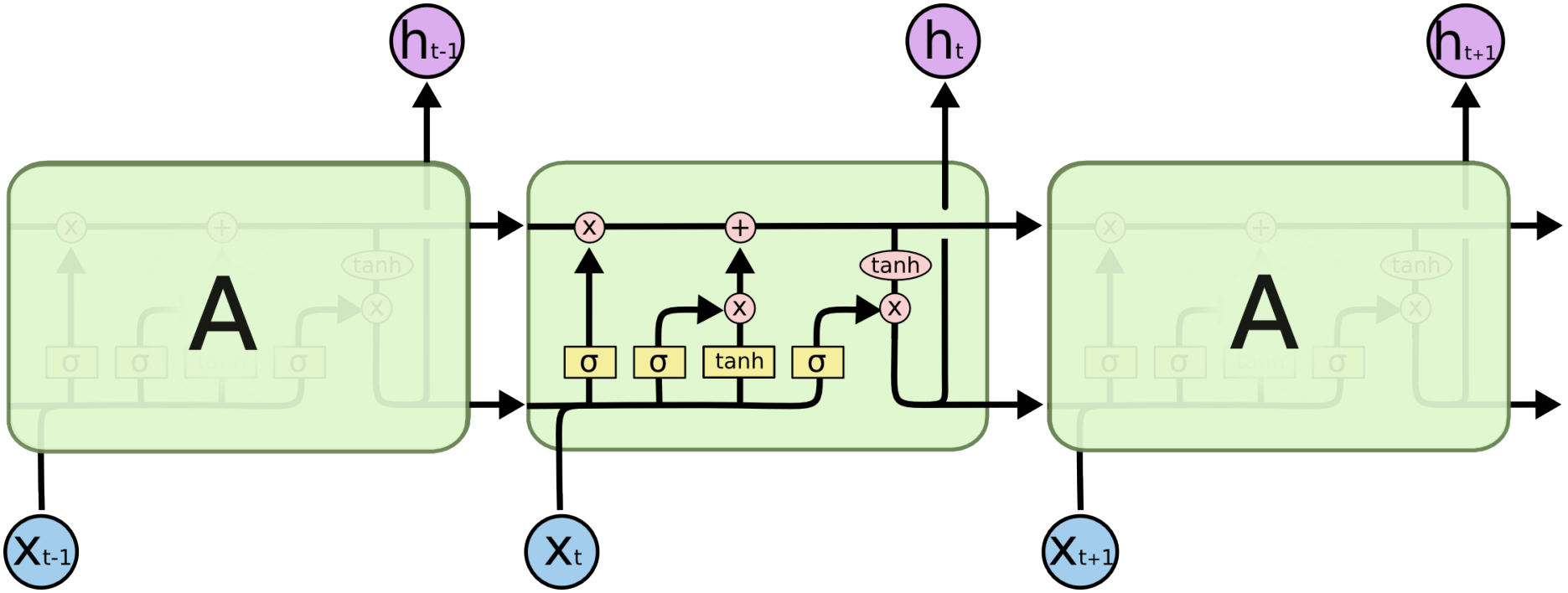
Long short term memory (LSTM)

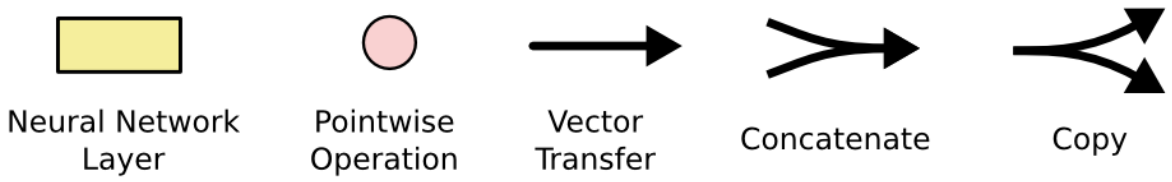
- Long short term memory RNN (Hochreiter and Schmidhuber 1997)
- LSTM is a special neuron which allows us:
  - To store information over time
  - To controll which information should be stored and which should be forgotten
- Recently, LSTM RNNs won several international pattern recognitions on large and complex data sets

RNN

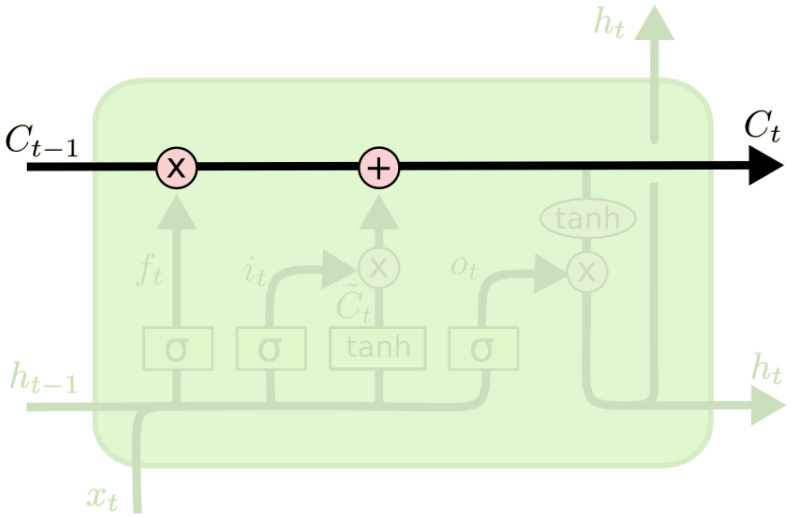


RNN - LSTM


h_{t-1}
h_t
h_{t+1}
A
A
σ
σ
tanh
σ
tanh
X_{t-1}
X_t
X_{t+1}

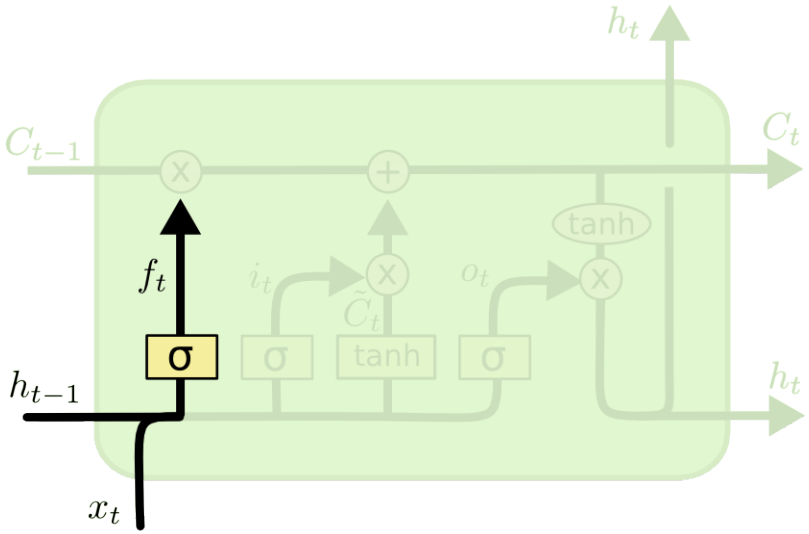

Neural Network Layer
Pointwise Operation
Vector Transfer
Concatenate
Copy

LSTM Core Idea



- LSTM allows to add (‚remember‘) or remove (‚forget‘) information after each time step
- The sigmoid layer ouputs values between 0 and 1 which indicates how much information should be let through

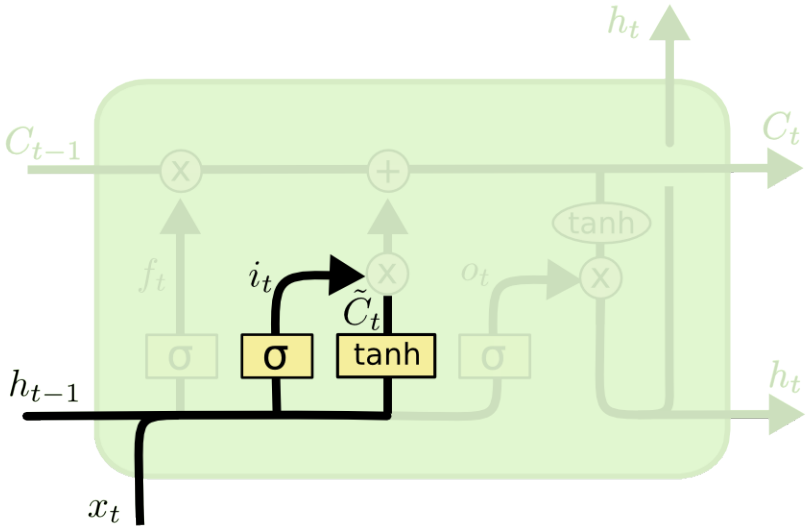
LSTM – Step by Step



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

- The first step is to decide how much information will be let through with a ‚forget gate‘
- 1: completely keep this and 0: completely forget it
- It takes into account the output of the previous time step and the input of the current time step

LSTM – Step by Step

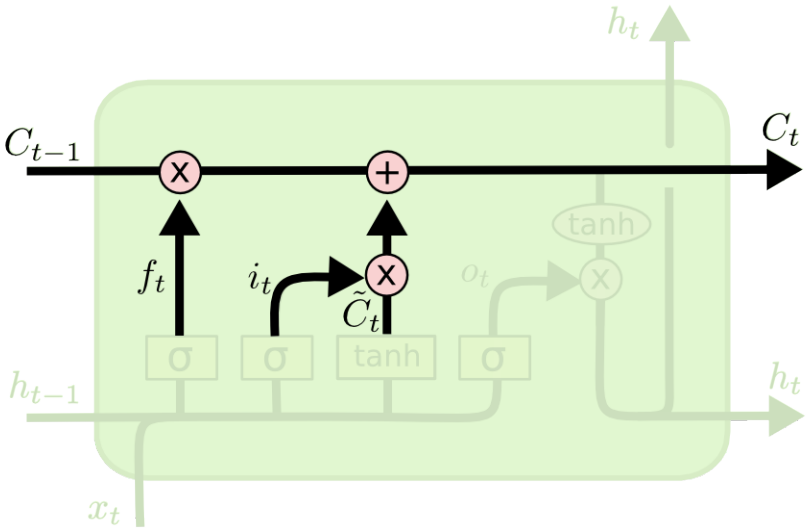


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

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

- The second step determines which new information should be stored
  - A sigmoid layer (‚input gate‘) decides which value should be updated
  - A tanh layer creates a vetor of new candidate values

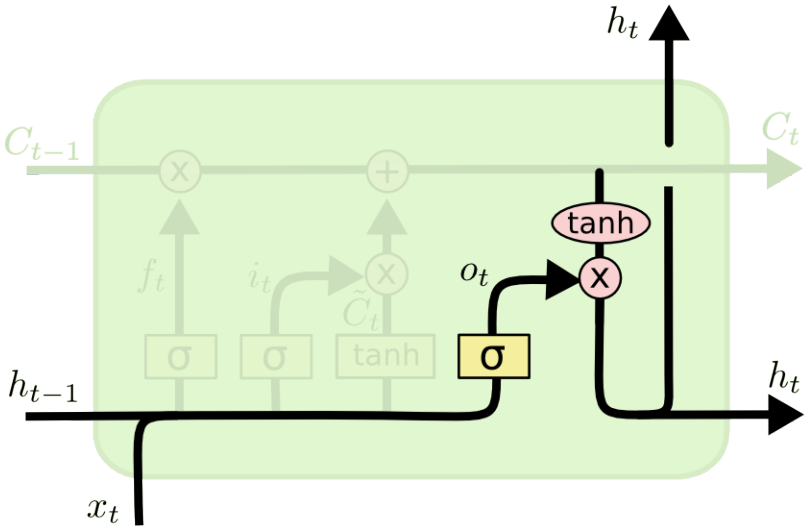
LSTM – Step by Step



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

- At this step, the memory will be updated with the information saved in the previous time step and the new information

LSTM – Step by Step

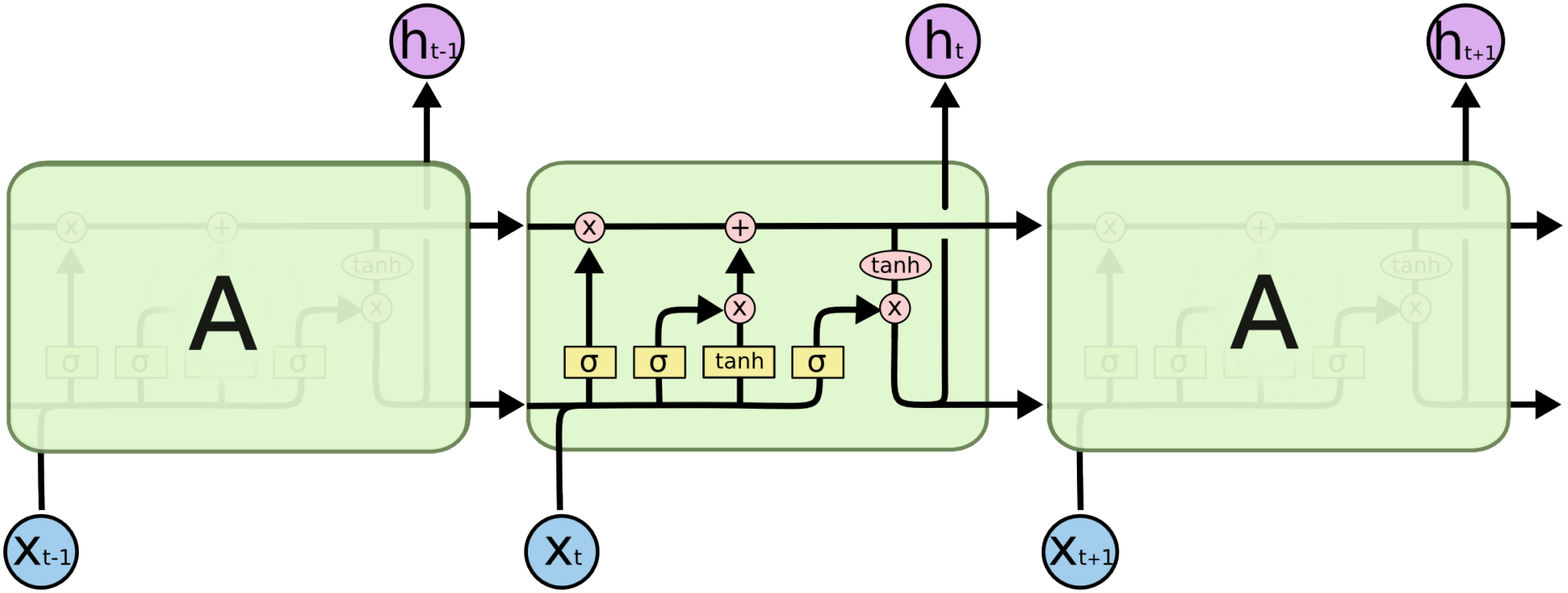


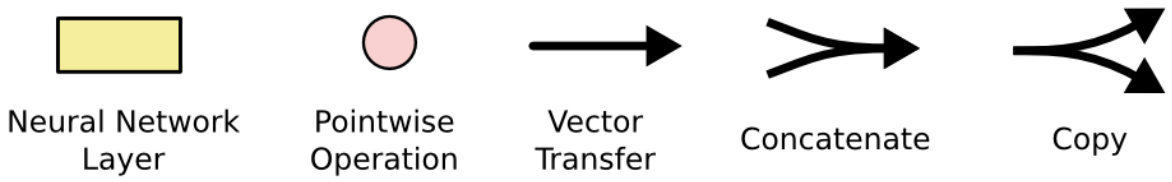
$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

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

- Finally, the output can be computed and then passed to the next step along with the memory
- Again the output of the sigmoid layer is multiplied with the tanh of the memory cell to pass only the information which the network decided to store

RNN - LSTM

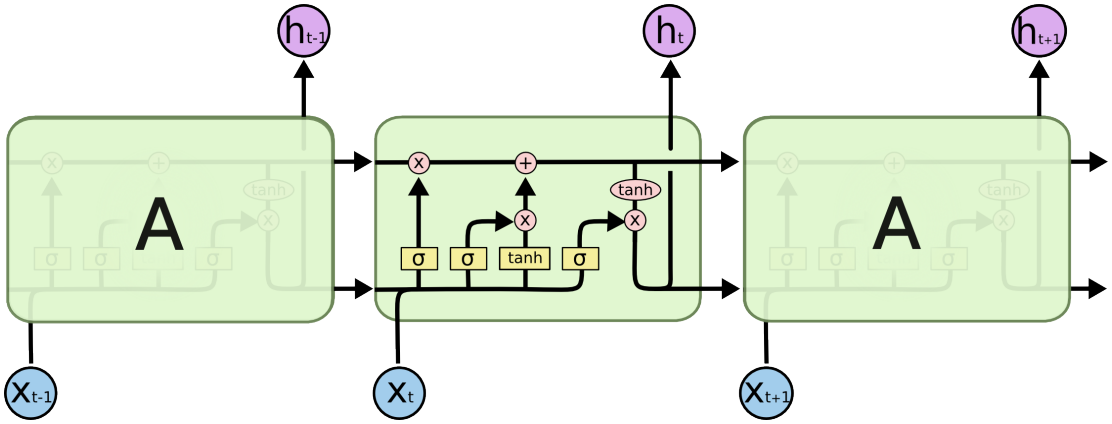

h_{t-1}
h_t
h_{t+1}
A
A
σ
σ
tanh
σ
tanh
X_{t-1}
X_t
X_{t+1}


Neural Network Layer
Pointwise Operation
Vector Transfer
Concatenate
Copy

Why can LSTM handle Vanishing Gradients?

Computing the Gradients

$$\frac{\delta C_k}{\delta W_h} = \sum_{k=1}^{T} \frac{\delta C_k}{\delta W}$$

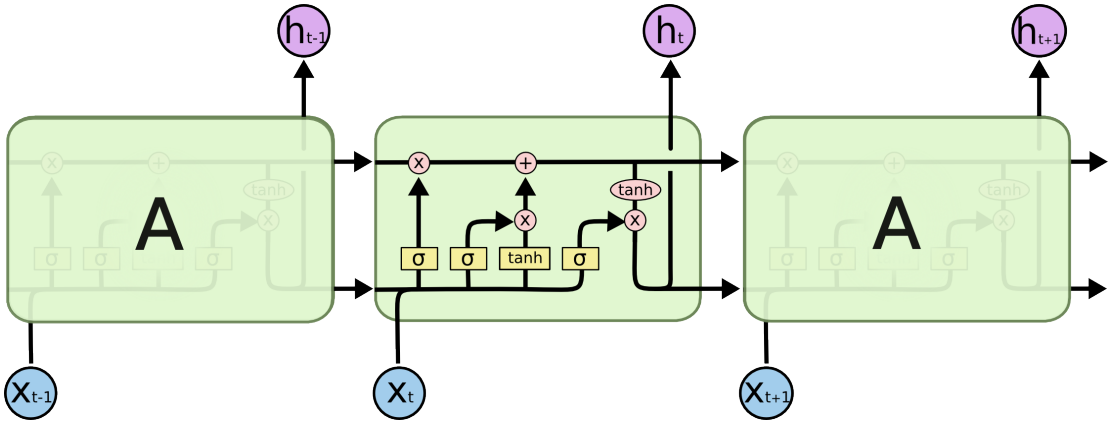


$$\frac{\delta C_k}{\delta W_h} = \frac{\delta C_k}{\delta h_k}\frac{\delta h_k}{\delta m_k} \cdots \frac{\delta m_2}{\delta m_1}\frac{\delta m_1}{\delta W}$$

$$= \frac{\delta C_k}{\delta h_k}\frac{\delta h_k}{\delta m_k}\left(\prod_{t=2}^{k} \frac{\delta m_t}{\delta m_{t-1}}\right)\frac{\delta m_1}{\delta W}$$

Computing the Gradients

$$\frac{\delta C_k}{\delta W_h} = \sum_{k=1}^{T} \frac{\delta C_k}{\delta W}$$

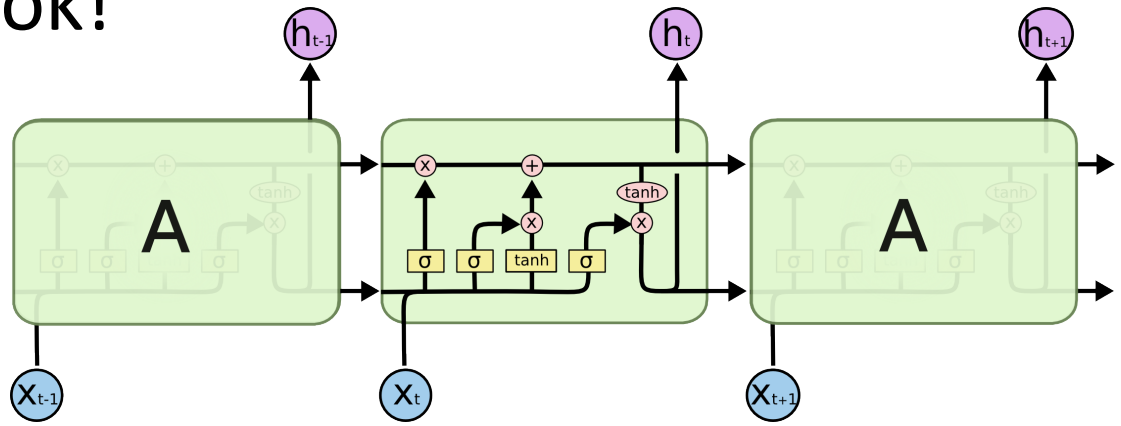


$$\frac{\delta C_k}{\delta W_h} = \frac{\delta C_k}{\delta h_k}\frac{\delta h_k}{\delta m_k} \cdots \frac{\delta m_2}{\delta m_1}\frac{\delta m_1}{\delta W}$$

$$= \frac{\delta C_k}{\delta h_k}\frac{\delta h_k}{\delta m_k}\left(\prod_{t=2}^{k} \frac{\delta m_t}{\delta m_{t-1}}\right)\frac{\delta m_1}{\delta W}$$

Computing the Gradients

- Let's take a closer look!



$$m_t = m_{t-1} \otimes f_t \oplus C_t \otimes i_t$$

$$\frac{\delta m_t}{\delta m_{t-1}} = \frac{\delta}{\delta m_{t-1}} [m_{t-1} \otimes f_t \oplus C_t \otimes i_t]$$

Computing the Gradients

- Let's take a closer look!

$$\frac{\delta m_t}{\delta m_{t-1}} = \frac{\delta}{\delta m_{t-1}} [m_{t-1} \otimes f_t \oplus C_t \otimes i_t]$$

$$= \frac{\delta}{\delta m_{t-1}} \left[m_{t-1} \otimes f_t] + \frac{\delta}{\delta m_{t-1}} [C_t \otimes i_t \right]$$

Computing the Gradients

- Let's take a closer look!

$$\frac{\delta m_t}{\delta m_{t-1}} = \frac{\delta}{\delta m_{t-1}}[m_{t-1} \otimes f_t \oplus C_t \otimes i_t]$$

$$= \frac{\delta}{\delta m_{t-1}}\left[m_{t-1} \otimes f_t] + \frac{\delta}{\delta m_{t-1}}[C_t \otimes i_t\right]$$

$$= \frac{\delta f_t}{\delta m_{t-1}}.m_{t-1} + \frac{\delta m_{t-1}}{\delta m_{t-1}}.f_t + \frac{\delta i_t}{\delta m_{t-1}}.C_t + \frac{\delta C_t}{\delta m_{t-1}}.i_t$$

Computing the Gradients

- Let's take a closer look!

$$\frac{\delta m_t}{\delta m_{t-1}} = \frac{\delta}{\delta m_{t-1}}[m_{t-1} \otimes f_t \oplus C_t \otimes i_t]$$

$$= \frac{\delta}{\delta m_{t-1}}\left[m_{t-1} \otimes f_t] + \frac{\delta}{\delta m_{t-1}}[C_t \otimes i_t\right]$$

$$= \frac{\delta f_t}{\delta m_{t-1}}.m_{t-1} + \frac{\delta m_{t-1}}{\delta m_{t-1}}.f_t + \frac{\delta i_t}{\delta m_{t-1}}.C_t + \frac{\delta C_t}{\delta m_{t-1}}.i_t$$

$$= \frac{\delta f_t}{\delta m_{t-1}}.m_{t-1} + f_t + \frac{\delta i_t}{\delta m_{t-1}}.C_t + \frac{\delta C_t}{\delta m_{t-1}}.i_t$$

Computing the Gradients

- Let's take a closer look!

$$\frac{\delta m_t}{\delta m_{t-1}} = \frac{\delta}{\delta m_{t-1}}[m_{t-1} \otimes f_t \oplus C_t \otimes i_t]$$

$$= \frac{\delta}{\delta m_{t-1}}\left[m_{t-1} \otimes f_t] + \frac{\delta}{\delta m_{t-1}}[C_t \otimes i_t\right]$$

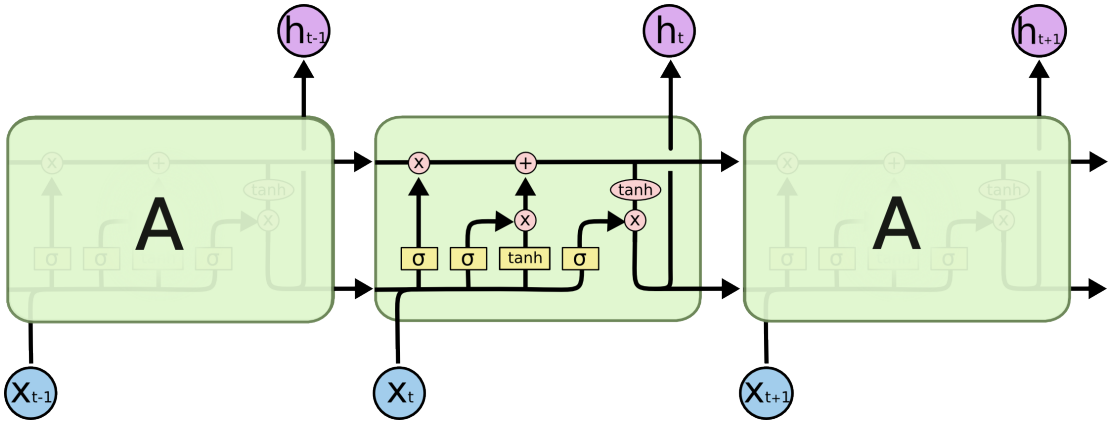
$$= \frac{\delta f_t}{\delta m_{t-1}}.m_{t-1} + \frac{\delta m_{t-1}}{\delta m_{t-1}}.f_t + \frac{\delta i_t}{\delta m_{t-1}}.C_t + \frac{\delta C_t}{\delta m_{t-1}}.i_t$$

$$= \frac{\delta f_t}{\delta m_{t-1}}.m_{t-1} + f_t + \frac{\delta i_t}{\delta m_{t-1}}.C_t + \frac{\delta C_t}{\delta m_{t-1}}.i_t$$

A_t B_t C_t D_t

Computing the Gradients

$$\frac{\delta C_k}{\delta W_h} = \sum_{k=1}^{T} \frac{\delta C_k}{\delta W}$$

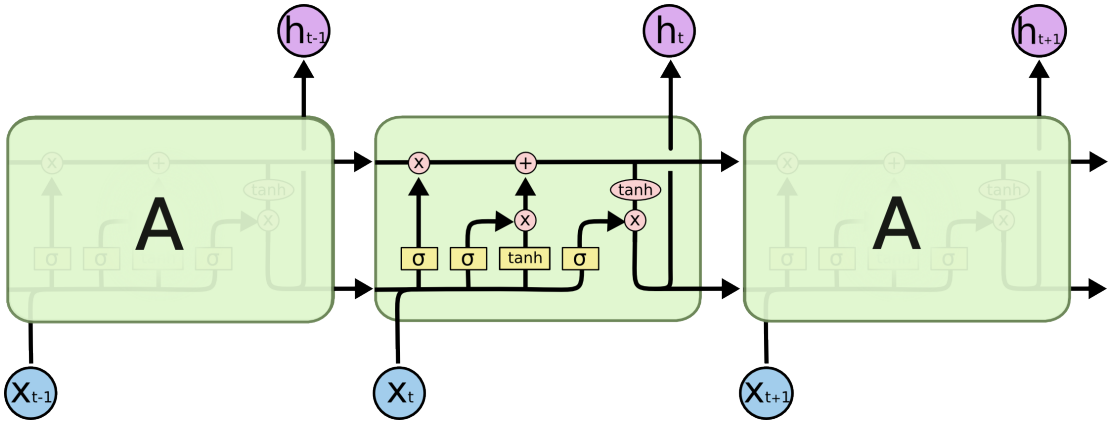


$$\frac{\delta C_k}{\delta W_h} = \frac{\delta C_k}{\delta h_k}\frac{\delta h_k}{\delta m_k} \cdots \frac{\delta m_2}{\delta m_1}\frac{\delta m_1}{\delta W}$$

$$= \frac{\delta C_k}{\delta h_k}\frac{\delta h_k}{\delta m_k}\left(\prod_{t=2}^{k} \frac{\delta m_t}{\delta m_{t-1}}\right)\frac{\delta m_1}{\delta W}$$

Computing the Gradients

$$\frac{\delta C_k}{\delta W} = \sum_{k=1}^{T} \frac{\delta C_k}{\delta W}$$

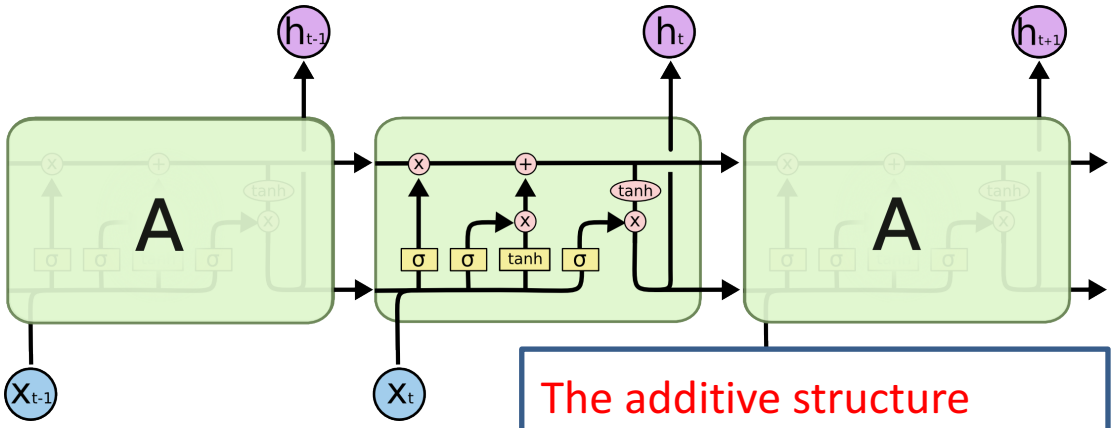


$$\frac{\delta C_k}{\delta W_h} = \frac{\delta C_k}{\delta h_k}\frac{\delta h_k}{\delta m_k} \cdots \frac{\delta m_2}{\delta m_1}\frac{\delta m_1}{\delta W}$$

$$= \frac{\delta C_k}{\delta h_k}\frac{\delta h_k}{\delta m_k}\left(\prod_{t=2}^{k} A_t + B_t + C_t + D_t\right)\frac{\delta m_1}{\delta W}$$

Why can LSTM handle Vanishing Gradients?

$$\frac{\delta C_k}{\delta W_h} = \sum_{k=1}^{T} \frac{\delta C_k}{\delta W}$$



The additive structure allows LSTM to better balance of the gradients. It is less likely that the sum will be vanished.

$$\frac{\delta C_k}{\delta W_h} \ldots \frac{\delta h_2}{\delta h_1} \frac{\delta m_1}{\delta W}$$

The presence of the activations of the forget gate's vector B_t.

$$= \frac{\delta C_k}{\delta h_k} \frac{\delta h_k}{\delta m_k} \left(\prod_{t=2}^{k} A_t + B_t + C_t + D_t \right) \frac{\delta m_1}{\delta W}$$