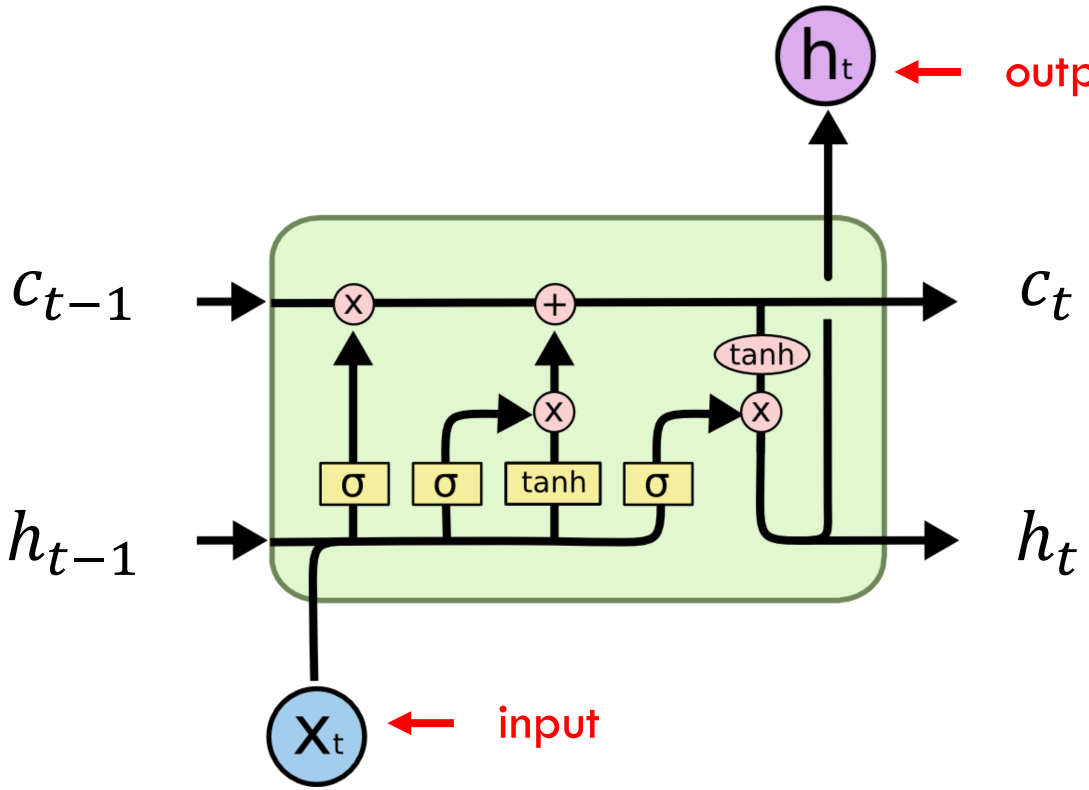
LSTM diagram



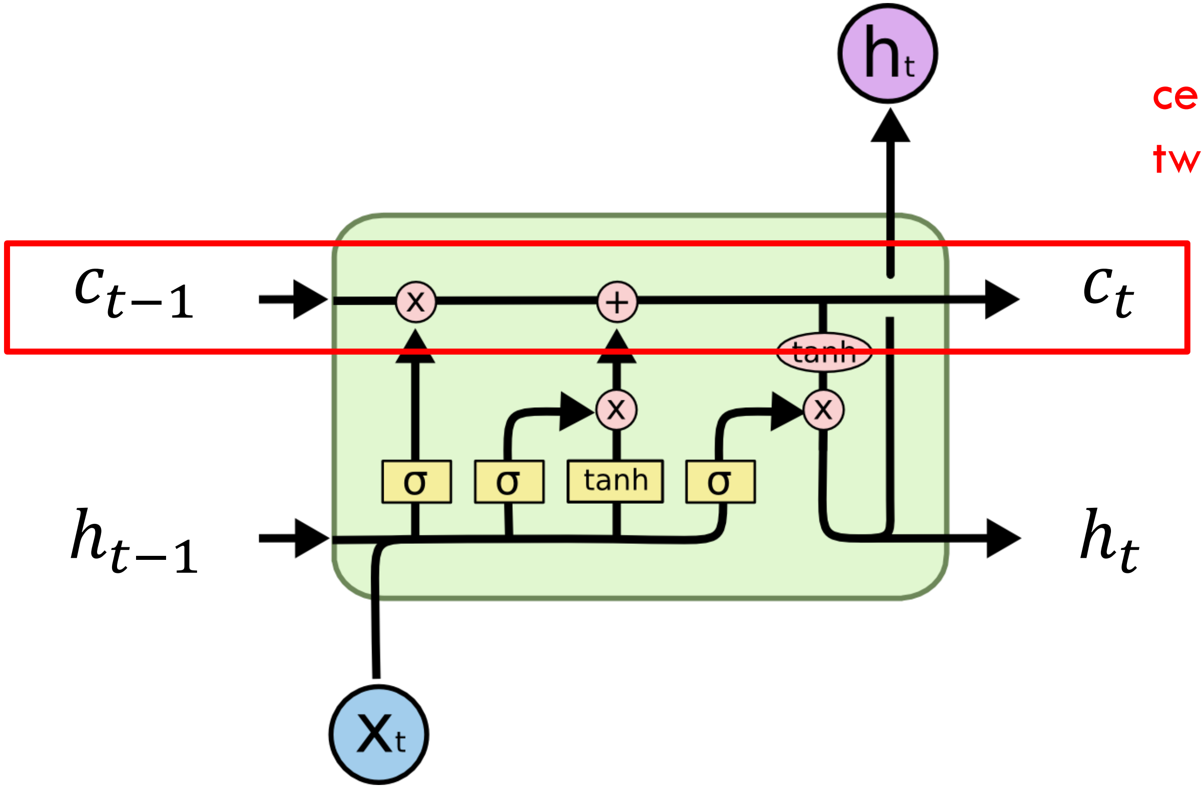
output

cell state

output (which also feeds into next unit)

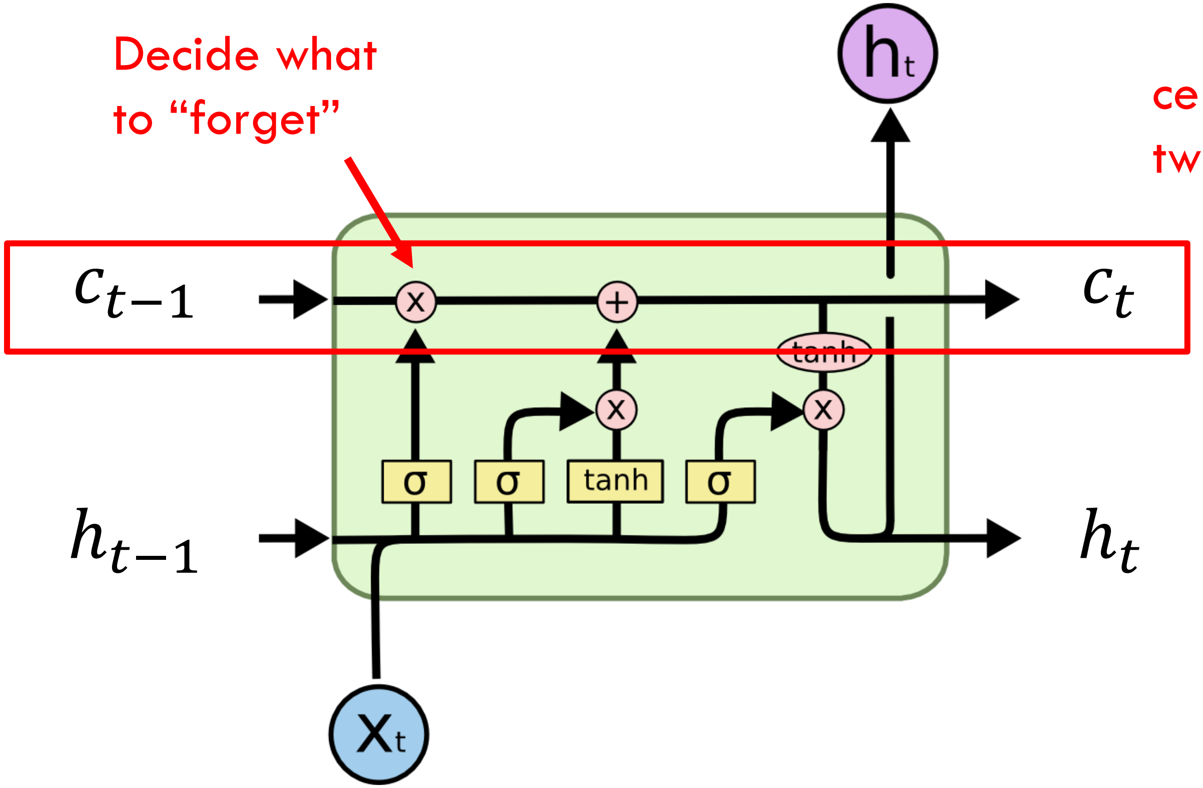
input

LSTM diagram



cell state gets updated in two stages

LSTM diagram



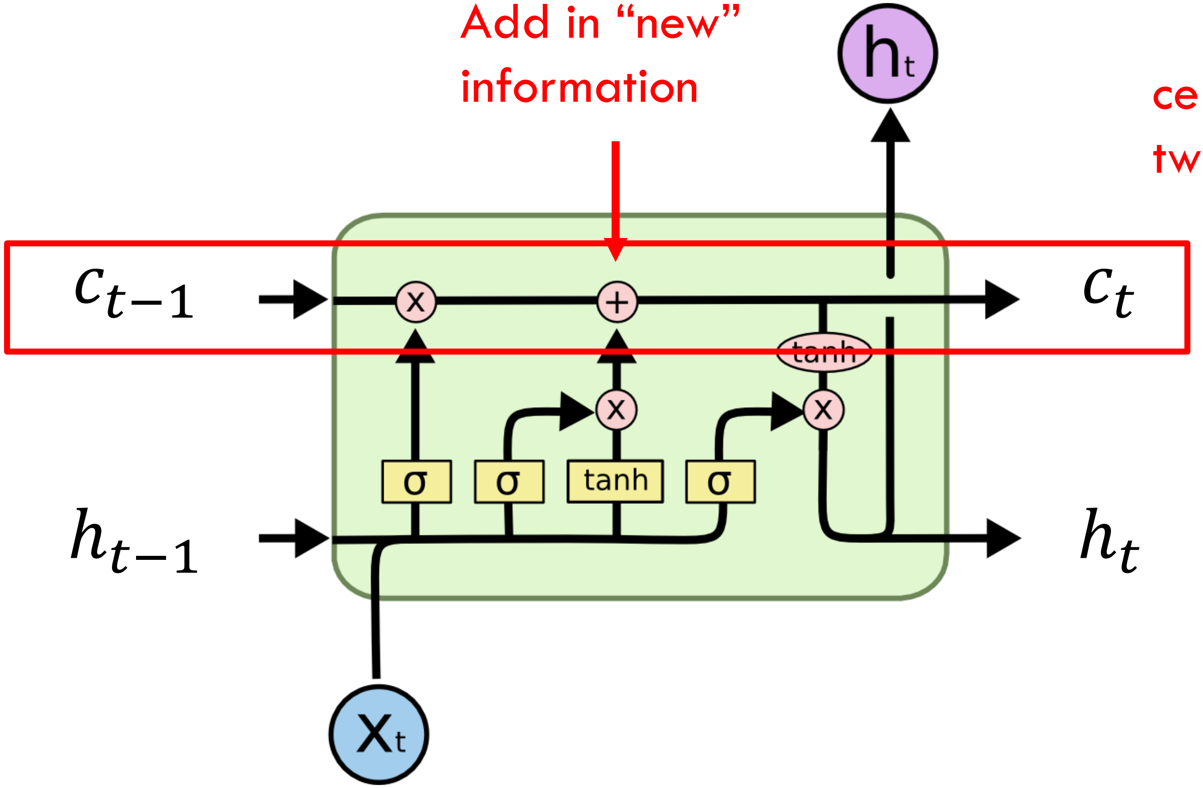
Decide what to “forget”

cell state gets updated in two stages

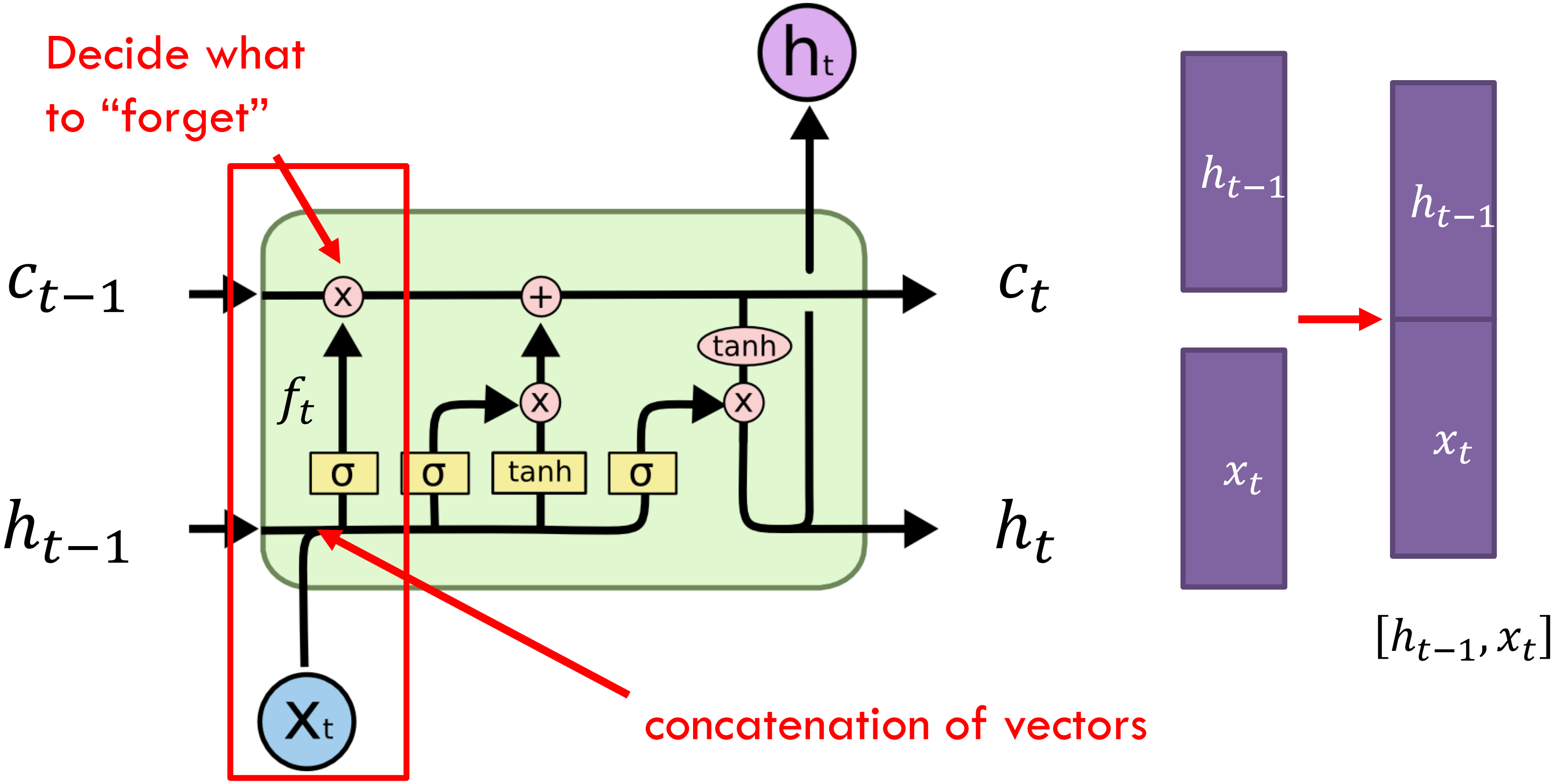
LSTM diagram

Add in “new” information

cell state gets updated in two stages



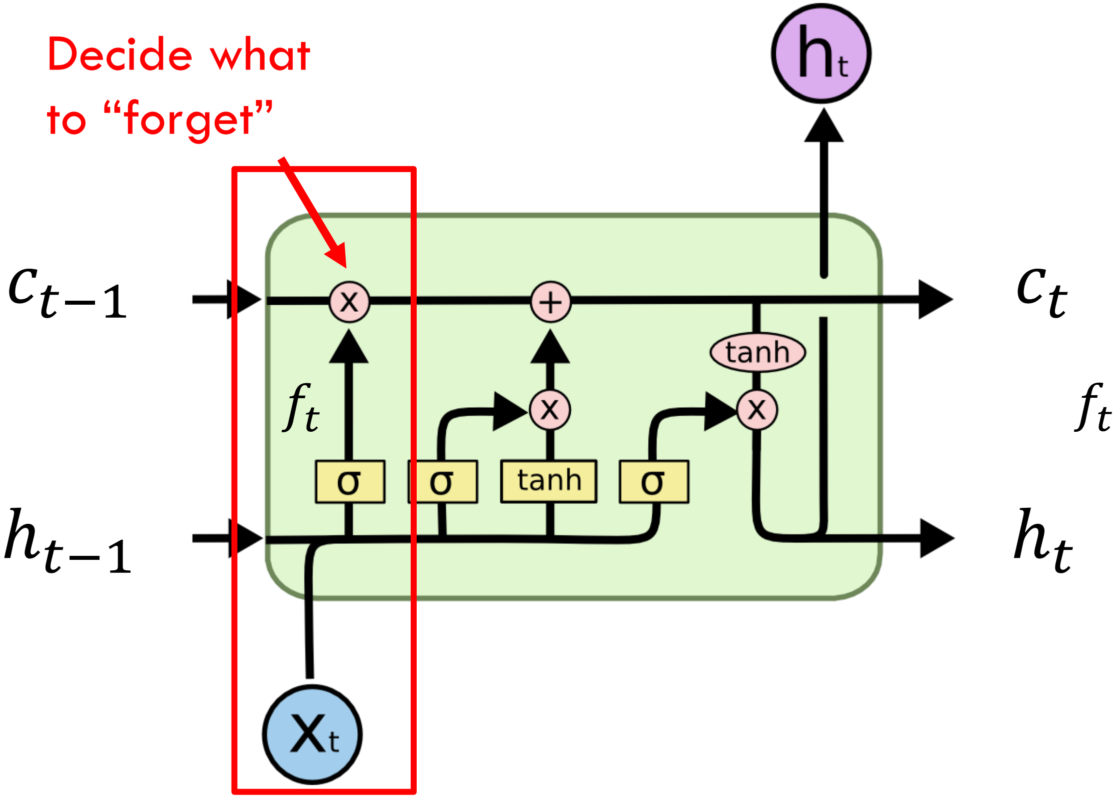
LSTM diagram



Decide what to “forget”

concatenation of vectors

LSTM diagram



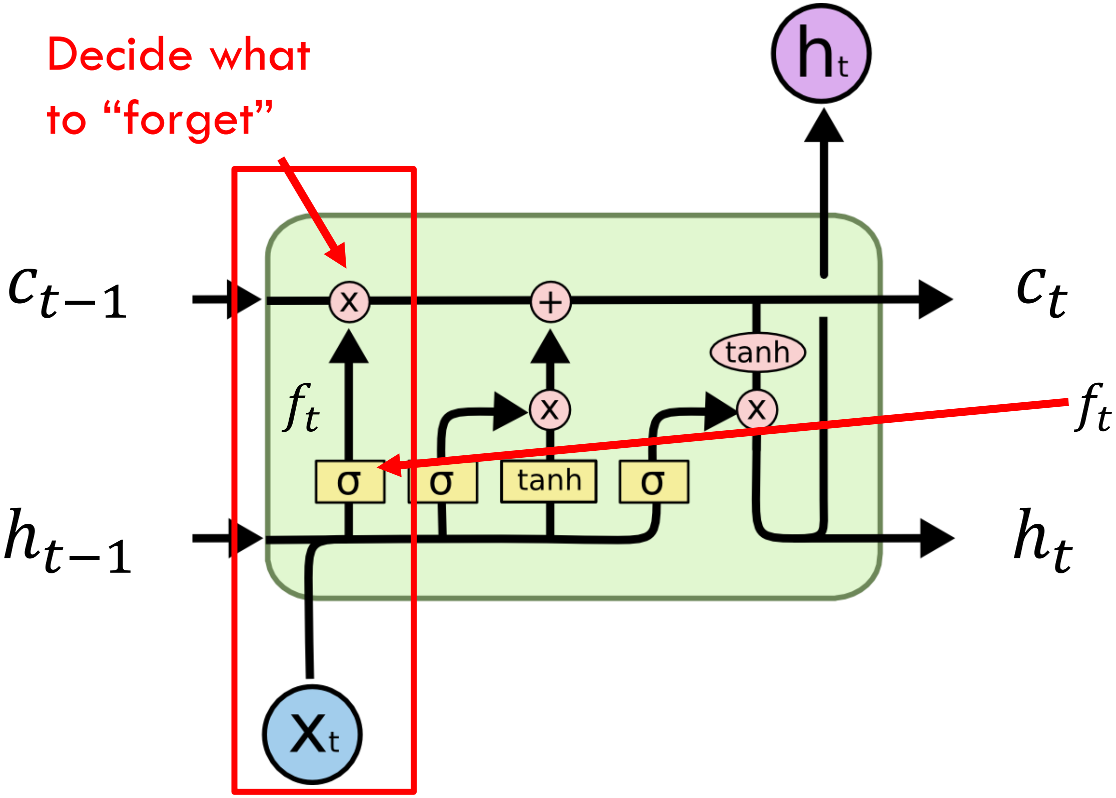
Decide what to “forget”

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

based on previous output and current input

LSTM diagram

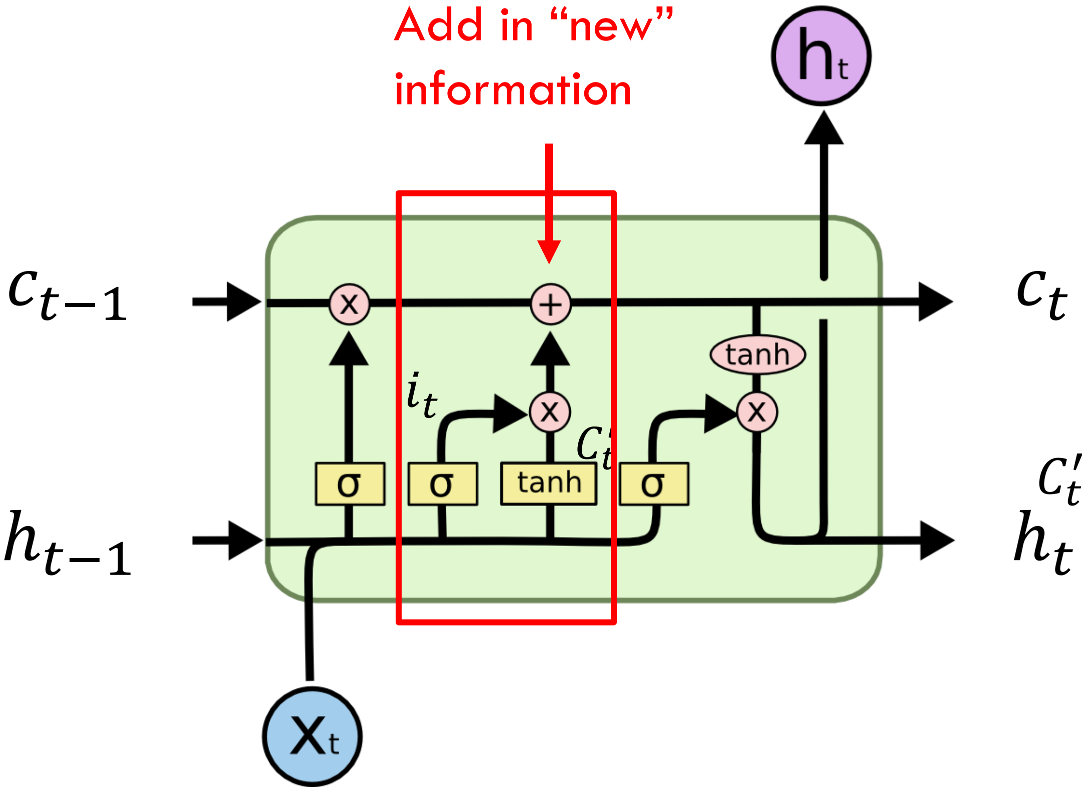
Decide what
to “forget”



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

based on
previous output
and current
input

LSTM diagram

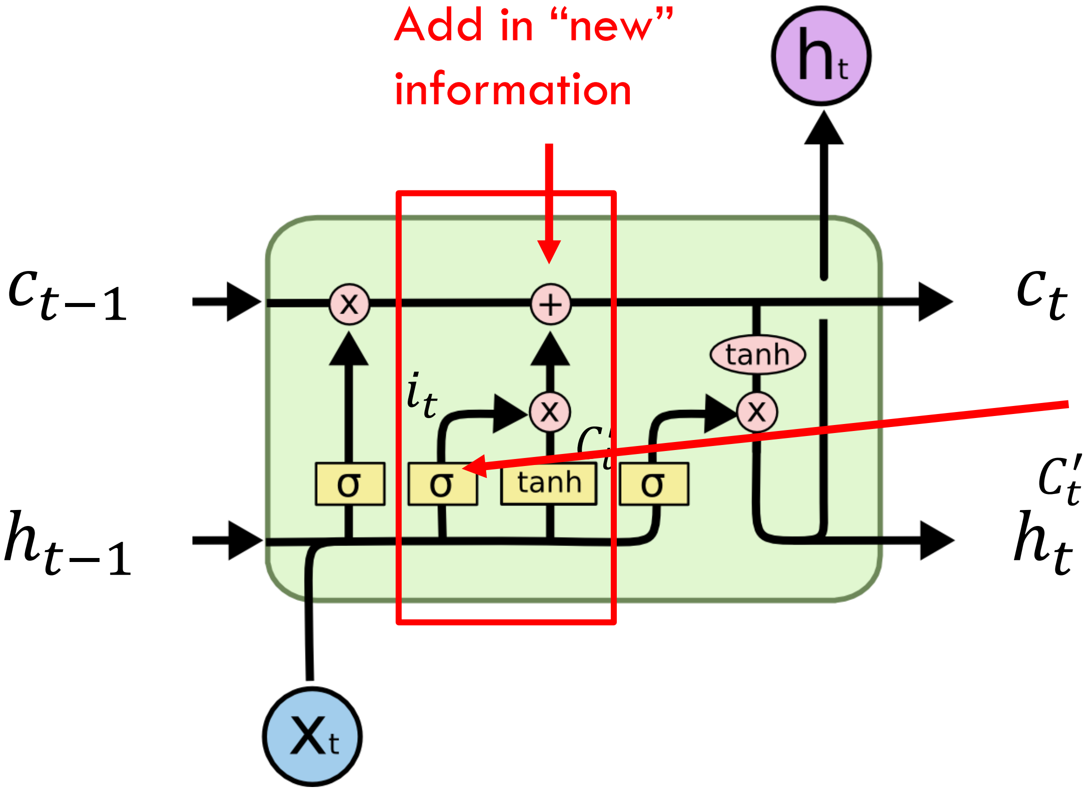


Add in “new” information

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

$$C'_t = tanh(W_C[h_{t-1}, x_t] + b_C)$$

LSTM diagram

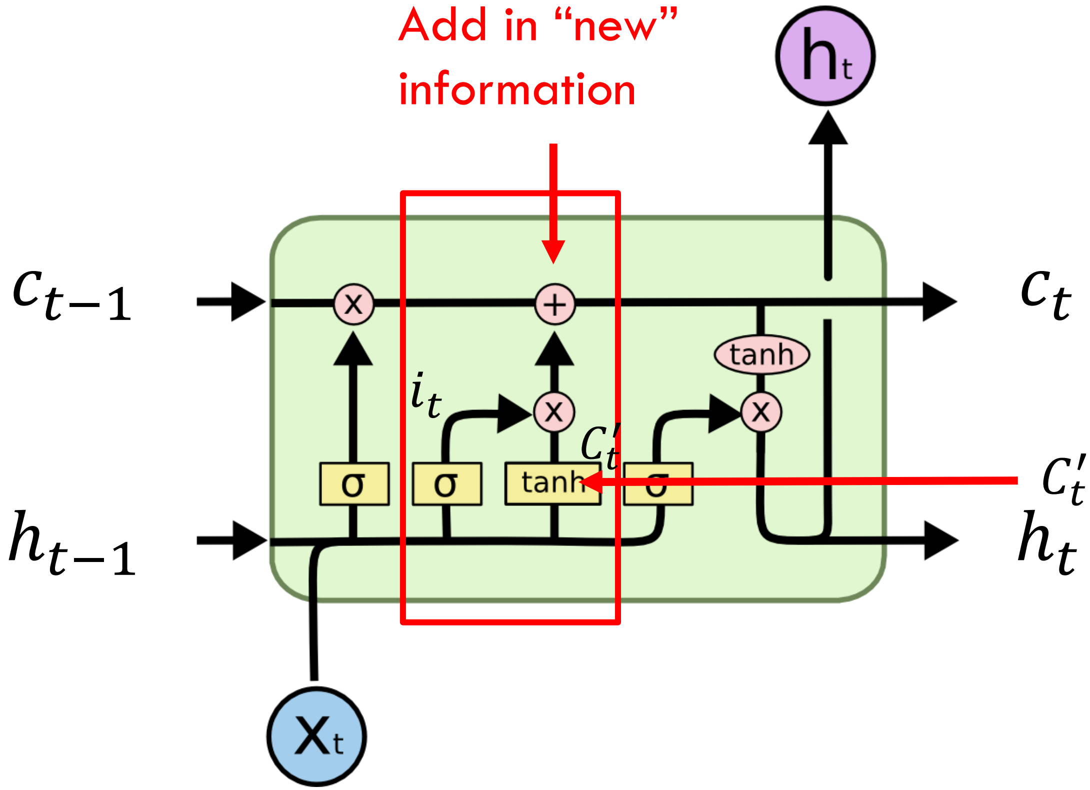


Add in “new” information

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

$$C'_t = tanh(W_C[h_{t-1}, x_t] + b_C)$$

LSTM diagram

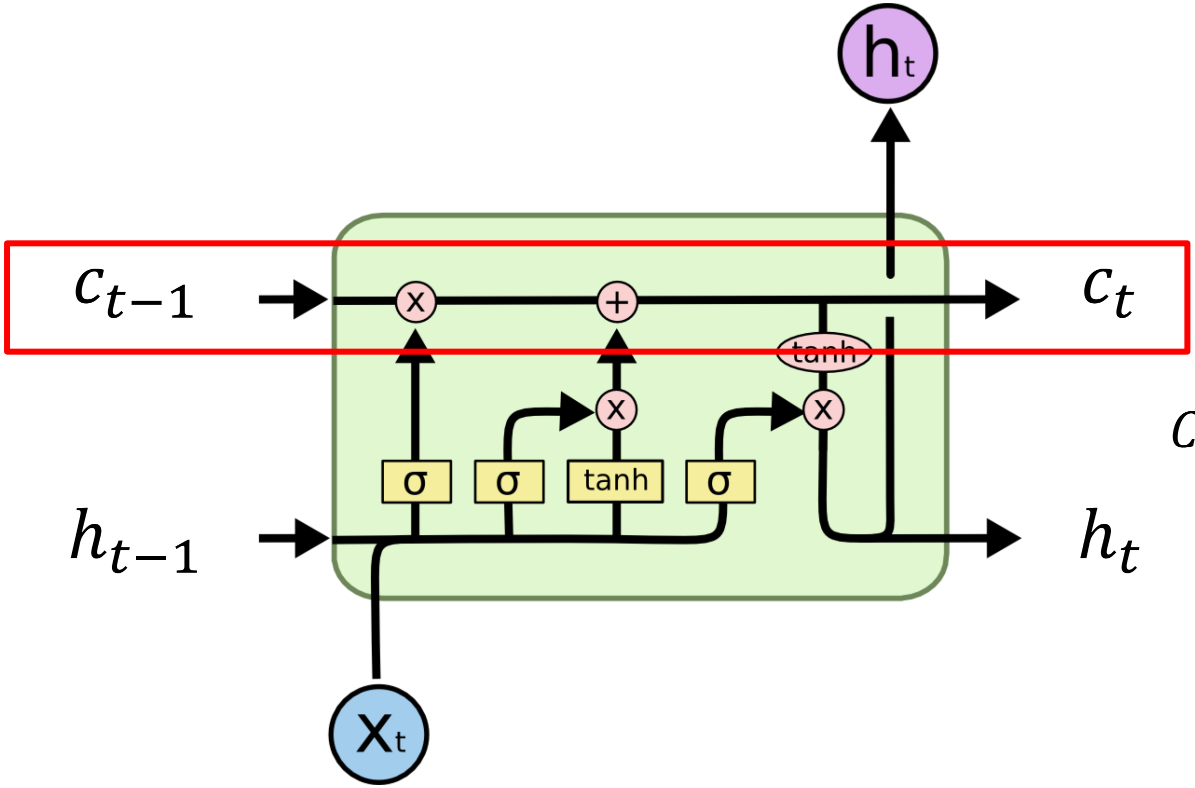


Add in “new” information

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

$$C'_t = tanh(W_C[h_{t-1}, x_t] + b_C)$$

LSTM diagram



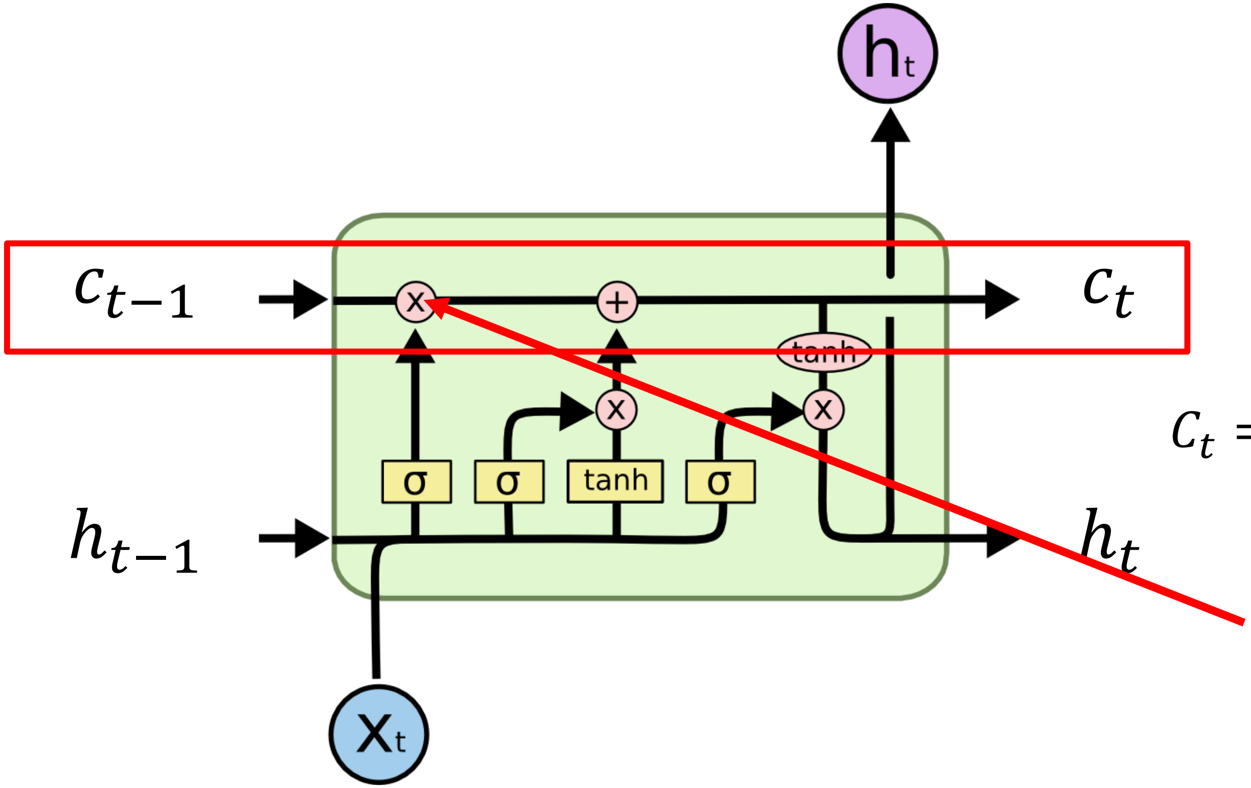
Note: '∗' represents element-wise multiplication

$$C_t = f_i * C_{t-1} + i_t * C'_t$$

forget the old (or not)

add the new (or not)

LSTM diagram



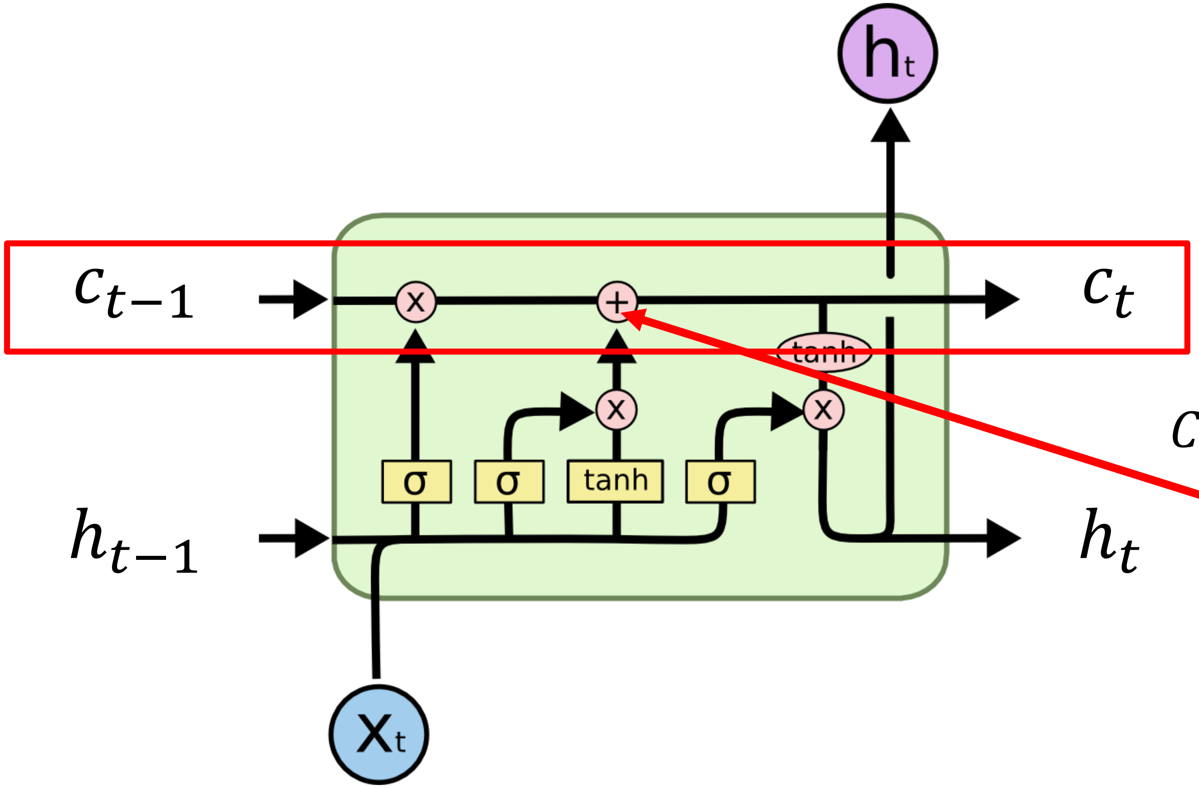
Note: ’*‘ represents element-wise multiplication

$$C_t = f_i * C_{t-1} + i_t * C_t'$$

forget the old (or not)

add the new (or not)

LSTM diagram



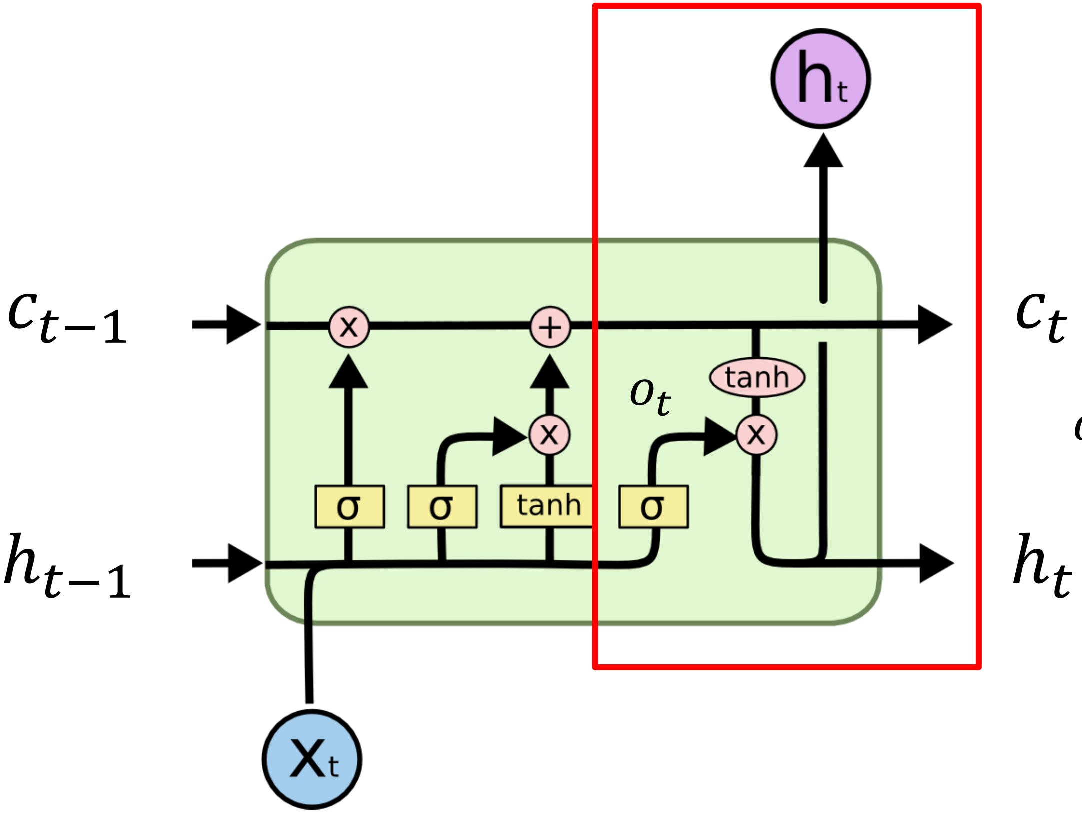
Note: '*' represents element-wise multiplication

$$C_t = f_i * C_{t-1} + i_t * C'_t$$

forget the old (or not)

add the new (or not)

LSTM diagram

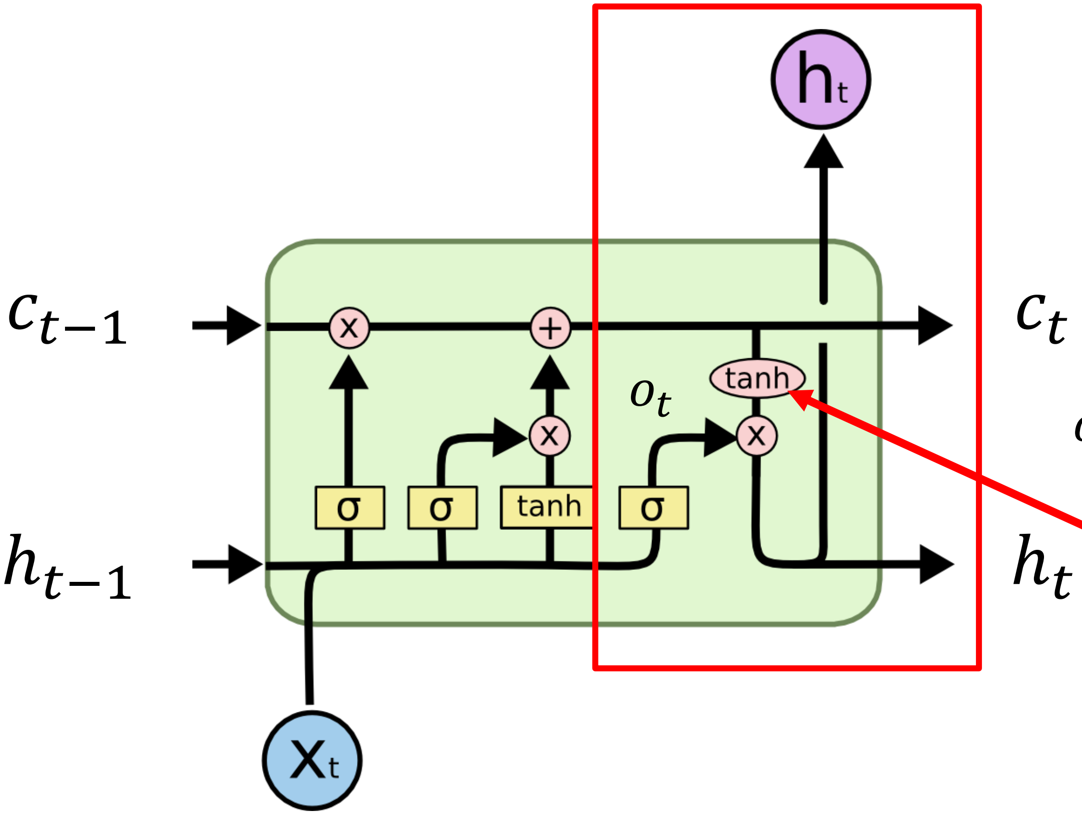


Final stage computes the output

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

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

LSTM diagram



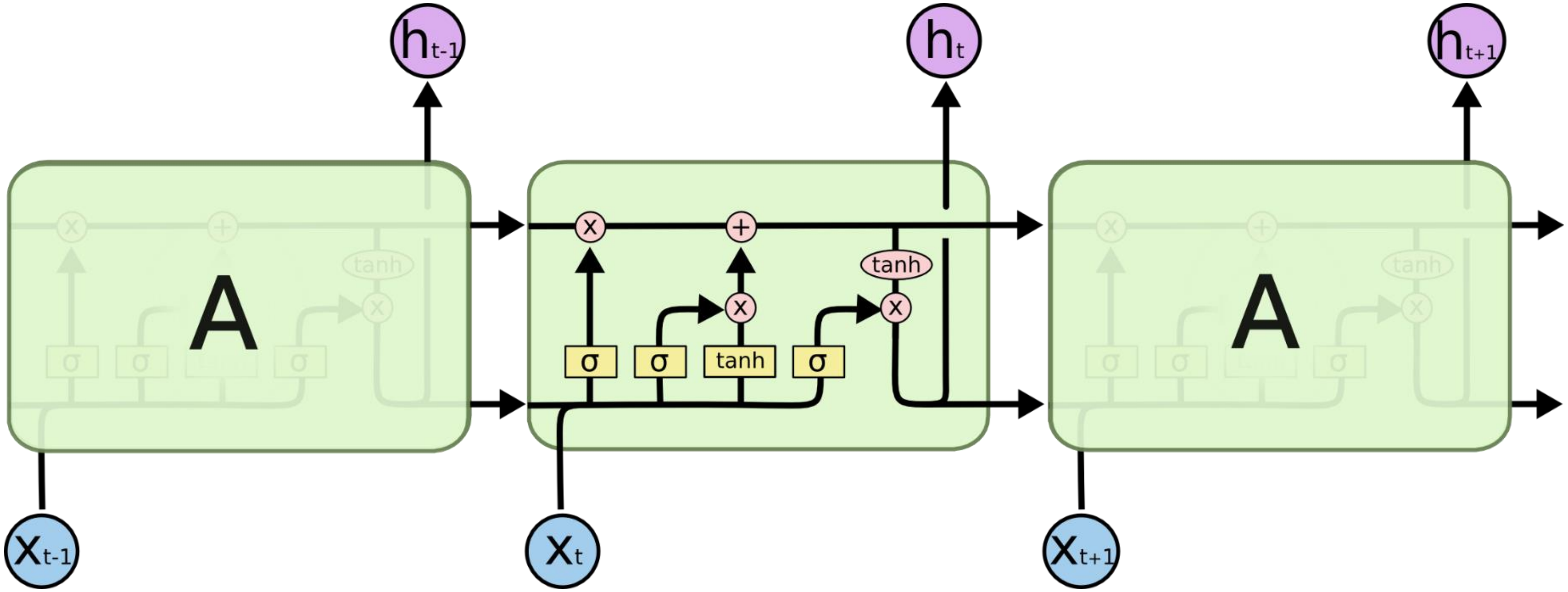
Final stage computes the output

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

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

Note: No weights here

LSTM unrolled



Final Points

- This is the most common version of LSTM, but there are many different “flavors”.
  - Gated Recurrent Unit (GRU)
  - Depth-Gated RNN

- LSTMs have considerably more parameters than plain RNNs.

- Most of the big performance improvements in NLP have come from LSTMs, not plain RNN.