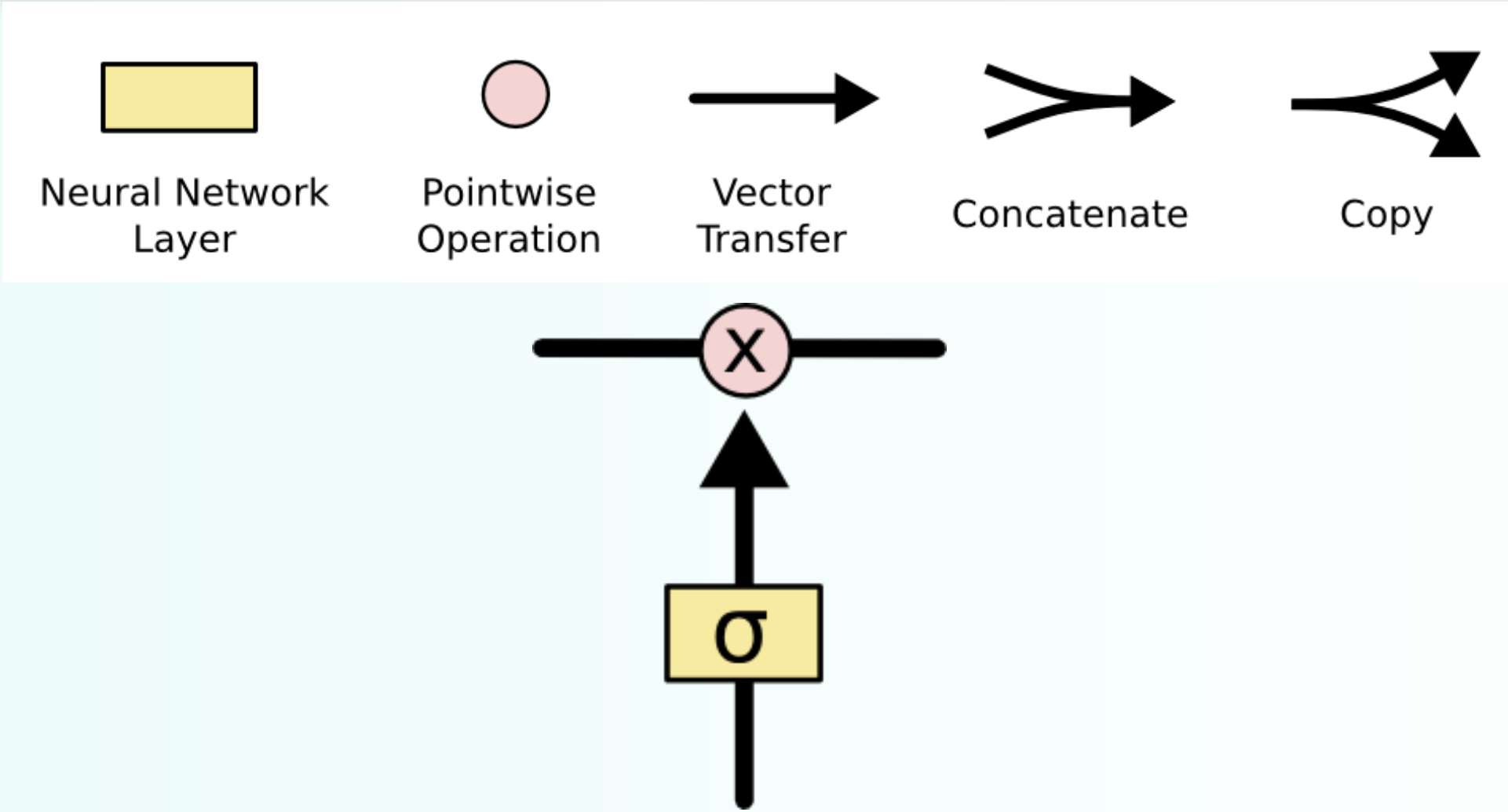
What is the LSTM (Long Short Term Memory)?

LSTM (short for long short-term memory) primarily solves the vanishing gradient problem in backpropagation. LSTMs use a gating mechanism that controls the memoizing process. Information in LSTMs can be stored, written, or read via gates that open and close. These gates store the memory in the analog format, implementing element-wise multiplication by sigmoid ranges between 0-1.



The sigmoid layer outputs numbers between 0-1 determine how much each component should be let through. Pink X gate is point-wise multiplication.

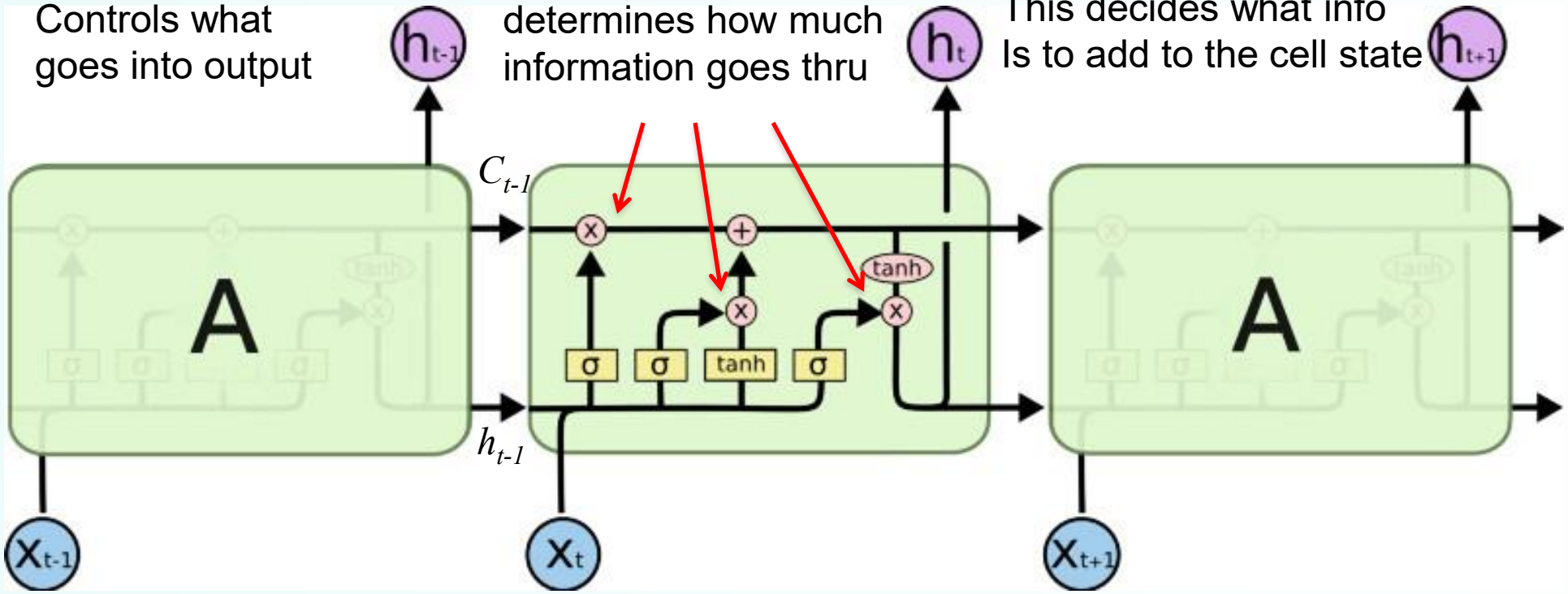
LSTM

Why sigmoid or tanh:
Sigmoid: 0,1 gating as switch.
Vanishing gradient problem in LSTM is handled already.
ReLU replaces tanh.

Output gate
Controls what goes into output

This sigmoid gate determines how much information goes thru

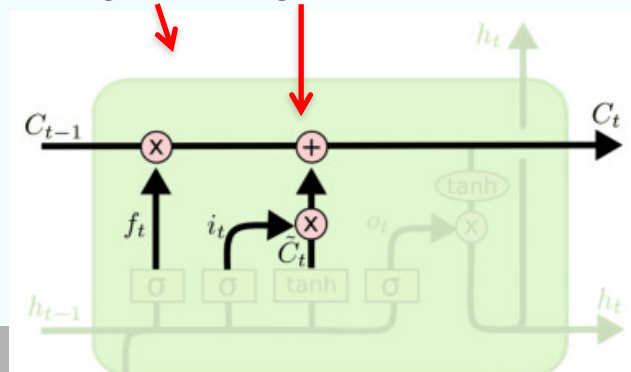
This decides what info
Is to add to the cell state



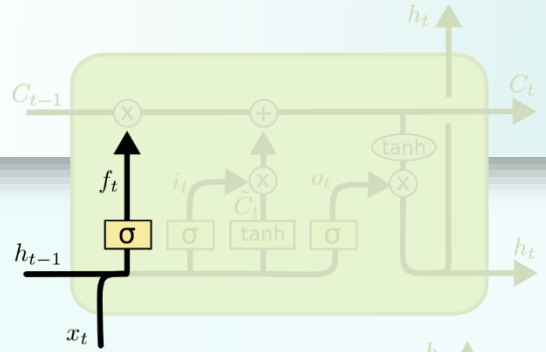
The core idea is this cell state C_t, it is changed slowly, with only minor linear interactions. It is very easy for information to flow along it unchanged.

Forget gate

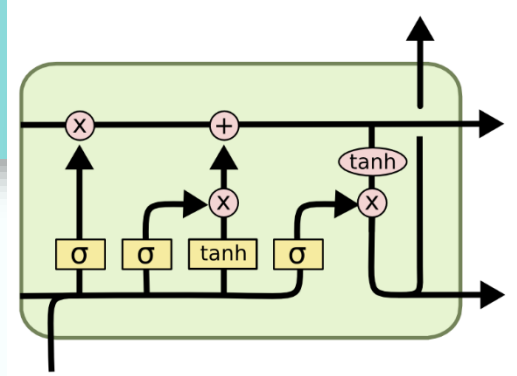
input gate

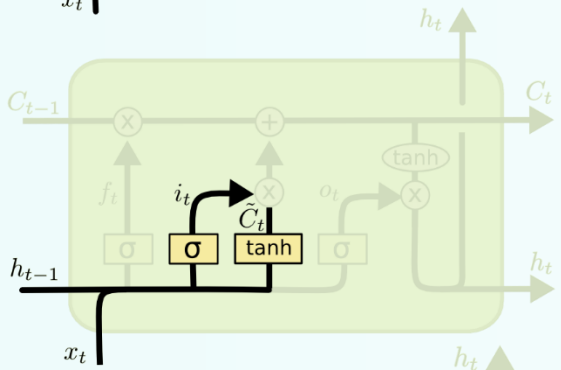


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



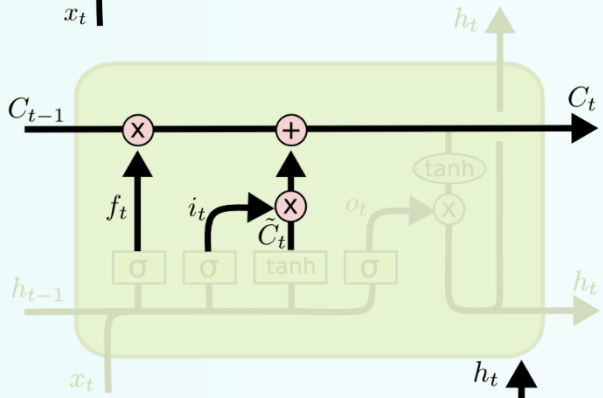
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$





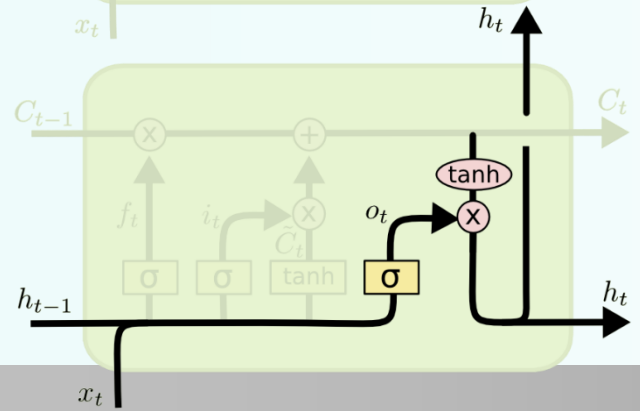
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

i_t decides what component is to be updated.
C'_t provides change contents



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

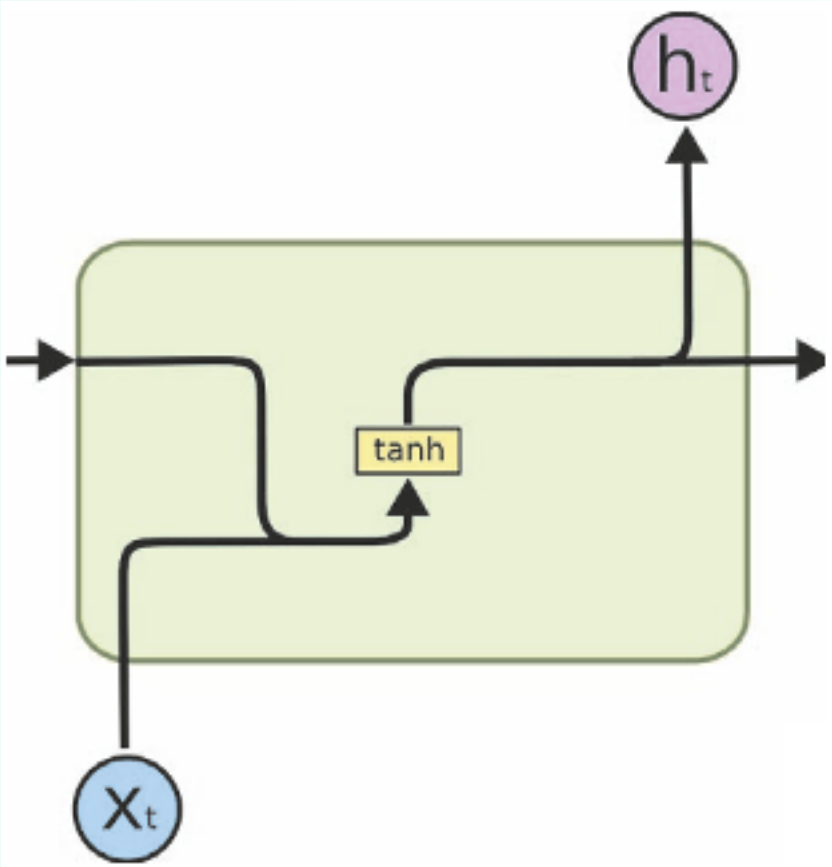
Updating the cell state



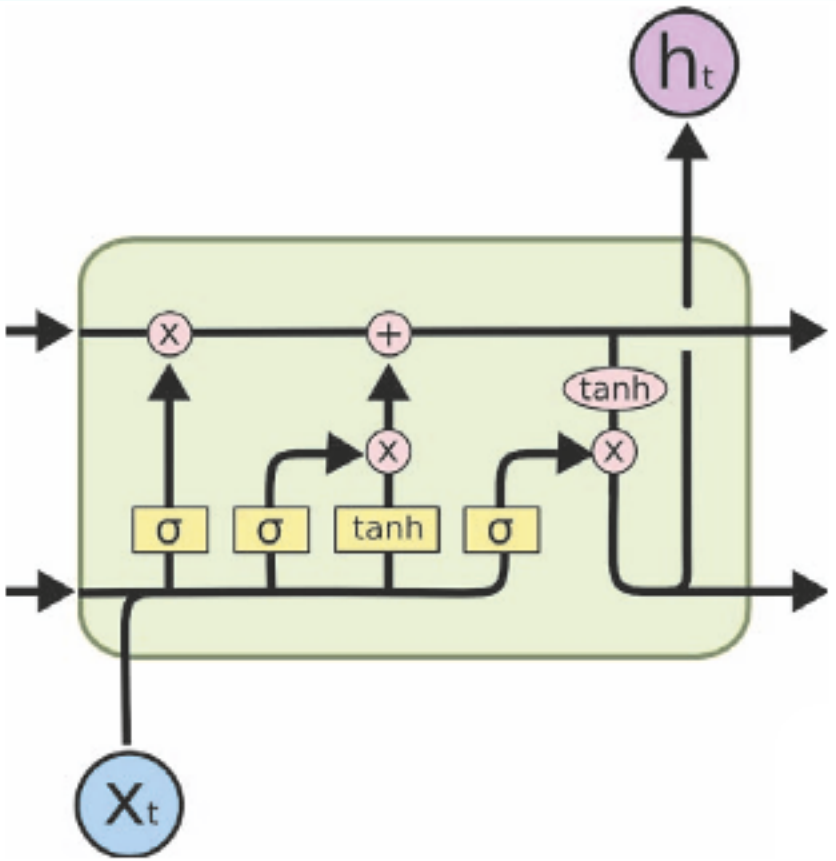
$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Decide what part of the cell state to output

RNN vs LSTM

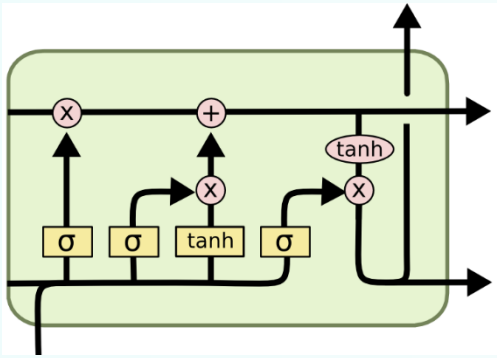


(a) RNN



(b) LSTM

Information Flow in a LSTM



c^{t-1}

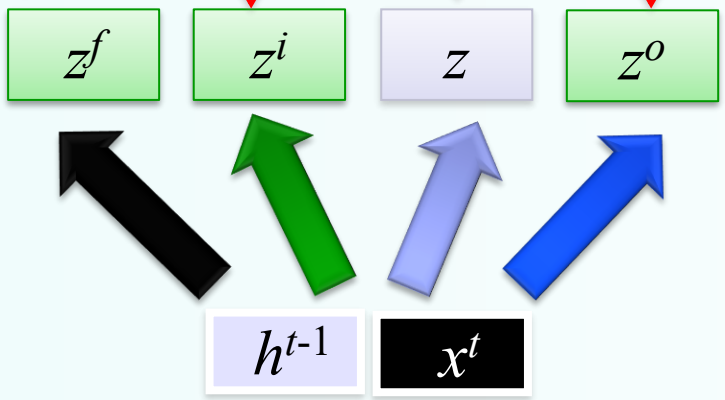
These 4 matrix computation can be done concurrently.

Controls forget gate

Controls input gate

Updating information

Controls Output gate

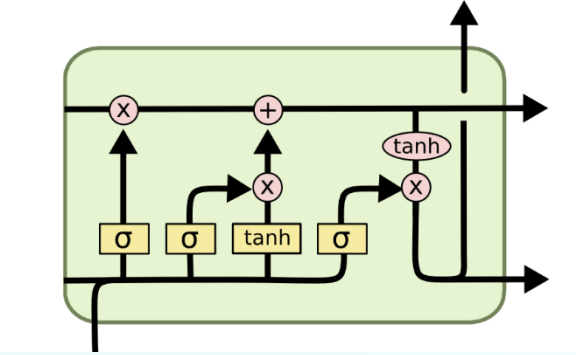


$$z = \tanh\left(W \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix}\right)$$

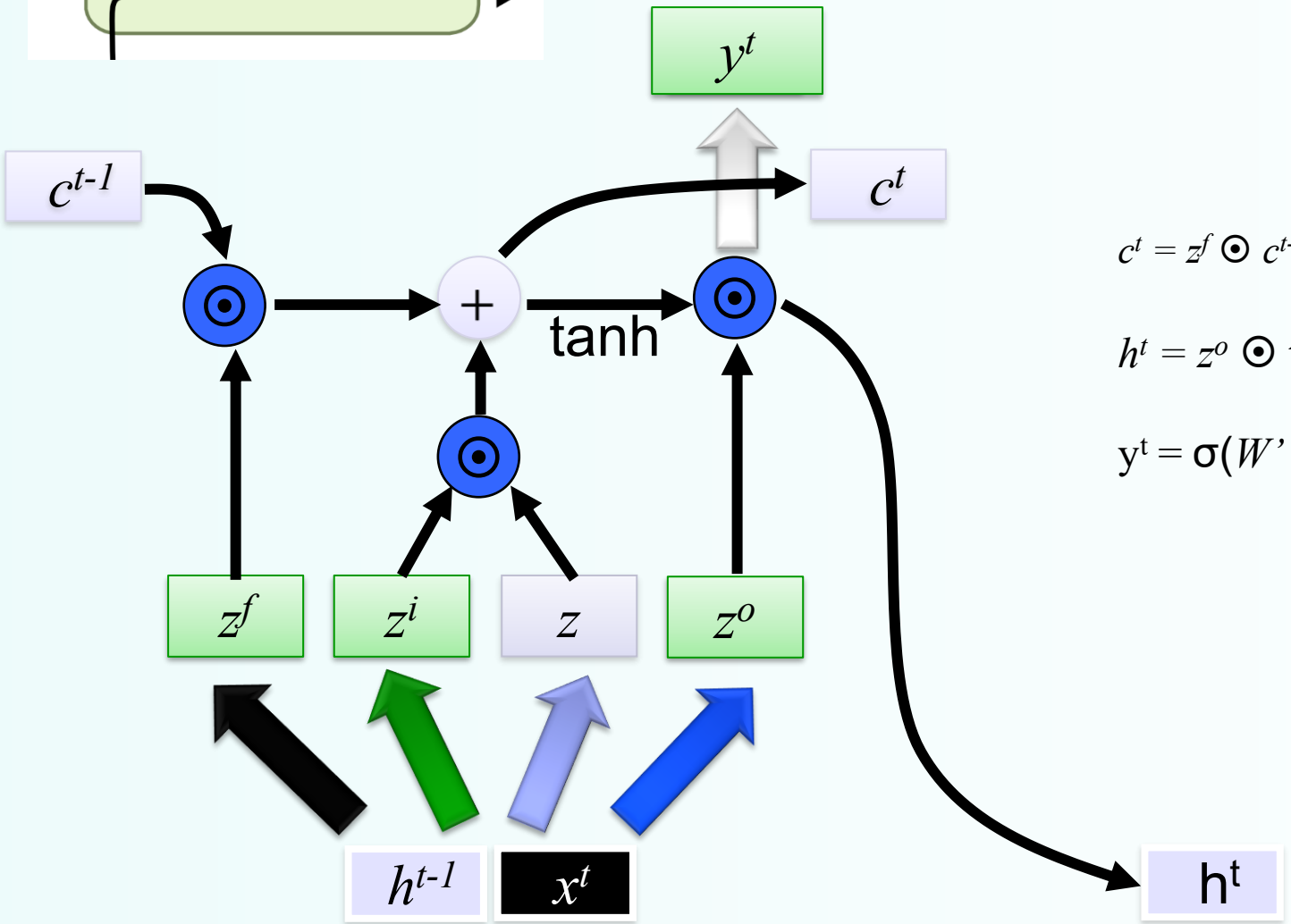
$$z^i = \sigma\left(W^i \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix}\right)$$

$$z^f = \sigma\left(W^f \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix}\right)$$

$$z^o = \sigma\left(W^o \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix}\right)$$



Element-wise multiply

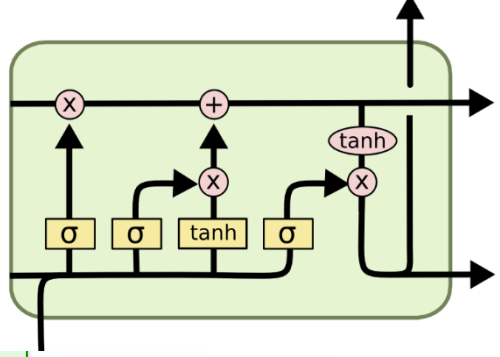


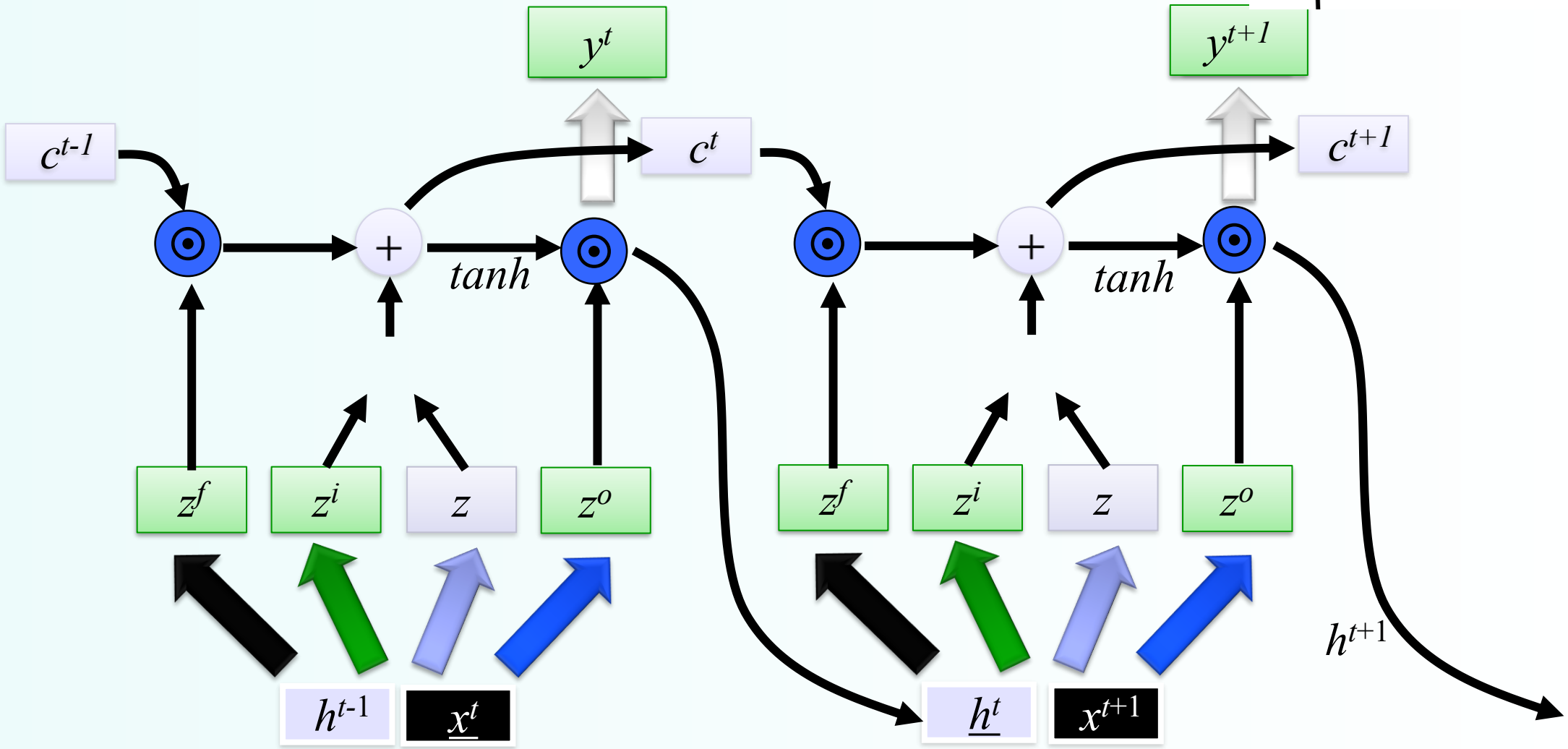
$$c^t = z^f \odot c^{t-1} + z^i \odot z$$

$$h^t = z^o \odot \tanh(c^t)$$

$$y^t = \sigma(W' h^t)$$

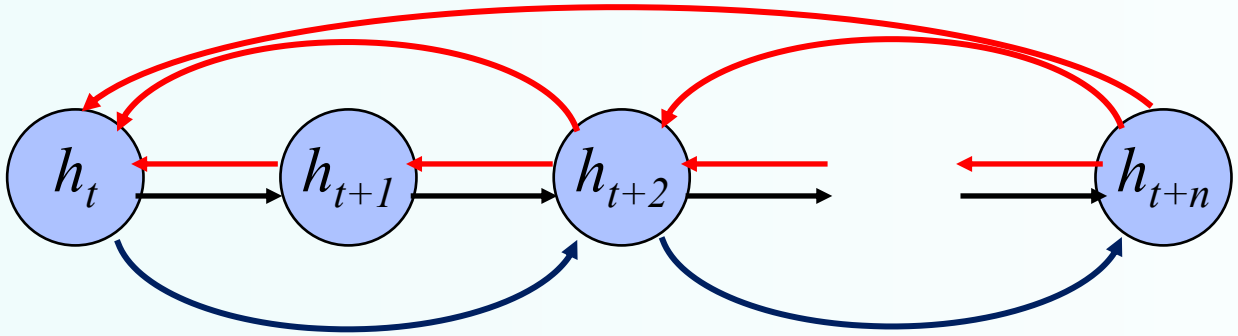
LSTM information flow





Adaptive Shortcut Connections through Gates Mechanism for Neural Network Pruning

- Perhaps we can create *adaptive* shortcut connections.
- Let the net prune unnecessary connections *adaptively*.



- Through the gates mechanism