

# **From RNN to LSTM**

**Basic concepts of RNN & LSTM**

Hyungjin Kim

# RNN

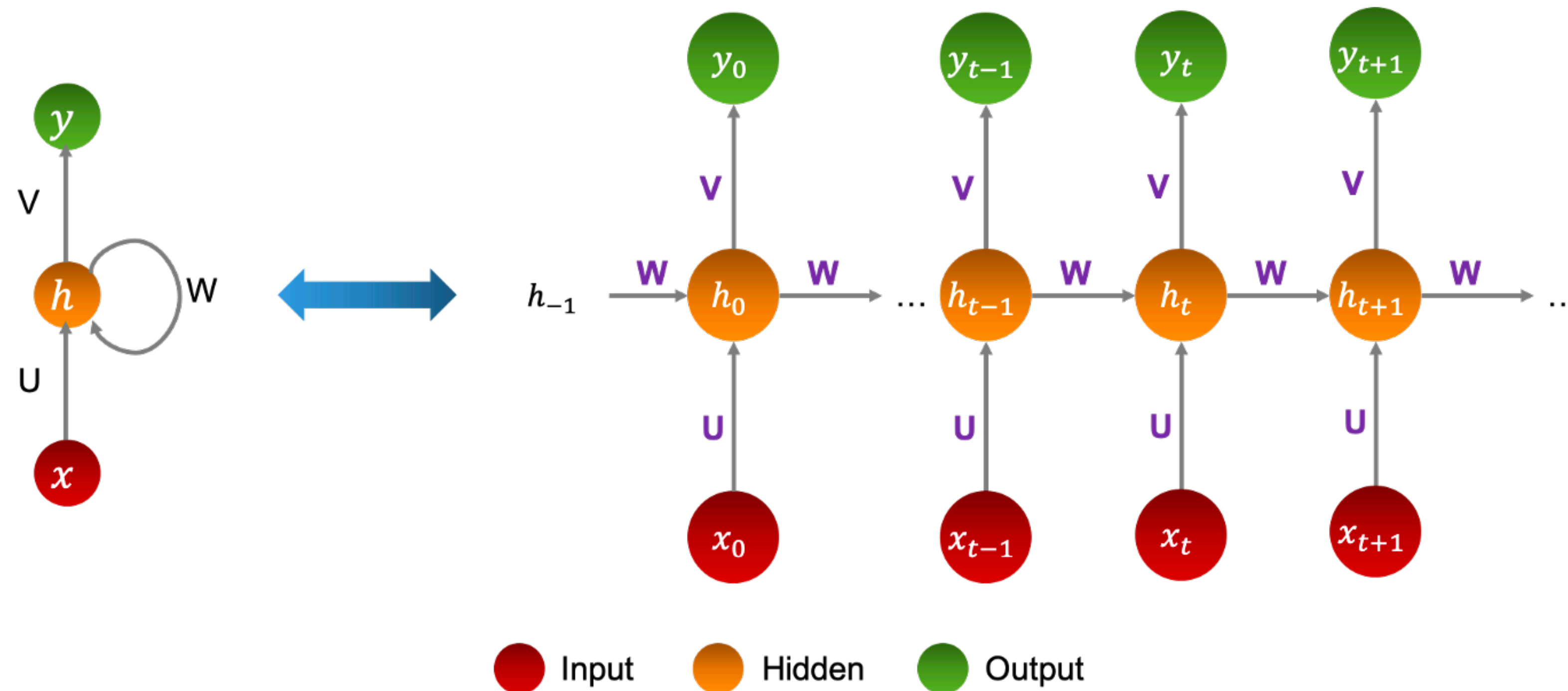
What is RNN?



**R**ecurrent **N**eural **N**etworks

# RNN

How it works?



# RNN

Tasks we could solve using RNN

the clouds are in the sky

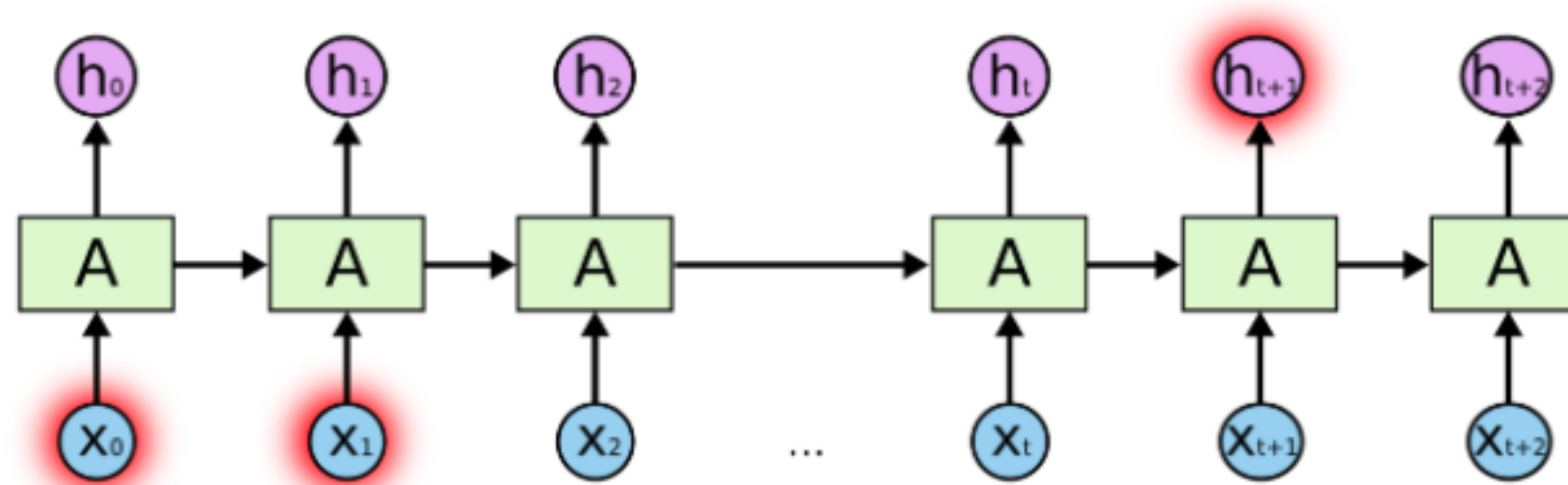
the clouds are in the

# RNN

## Drawbacks

Vanishing Gradient Problem

### Long-Term Dependency



# **RNN**

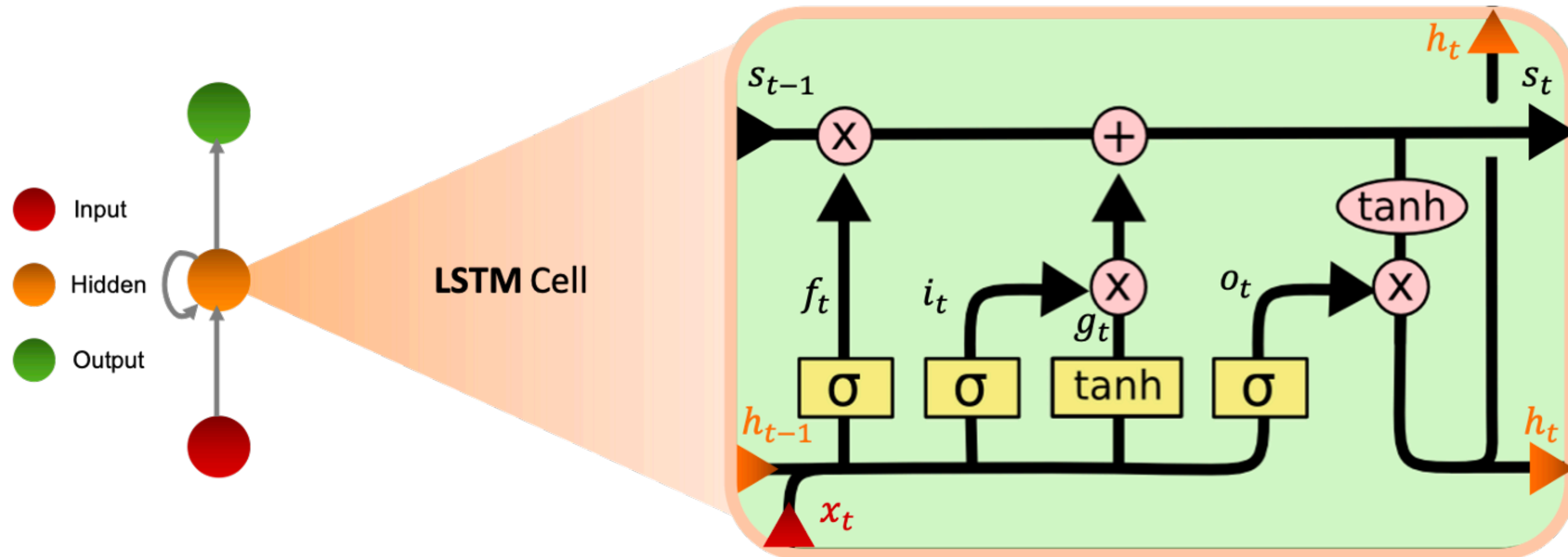
**Solutions**

# **LSTM**

(Hochreiter & Schmidhuber, 1997)

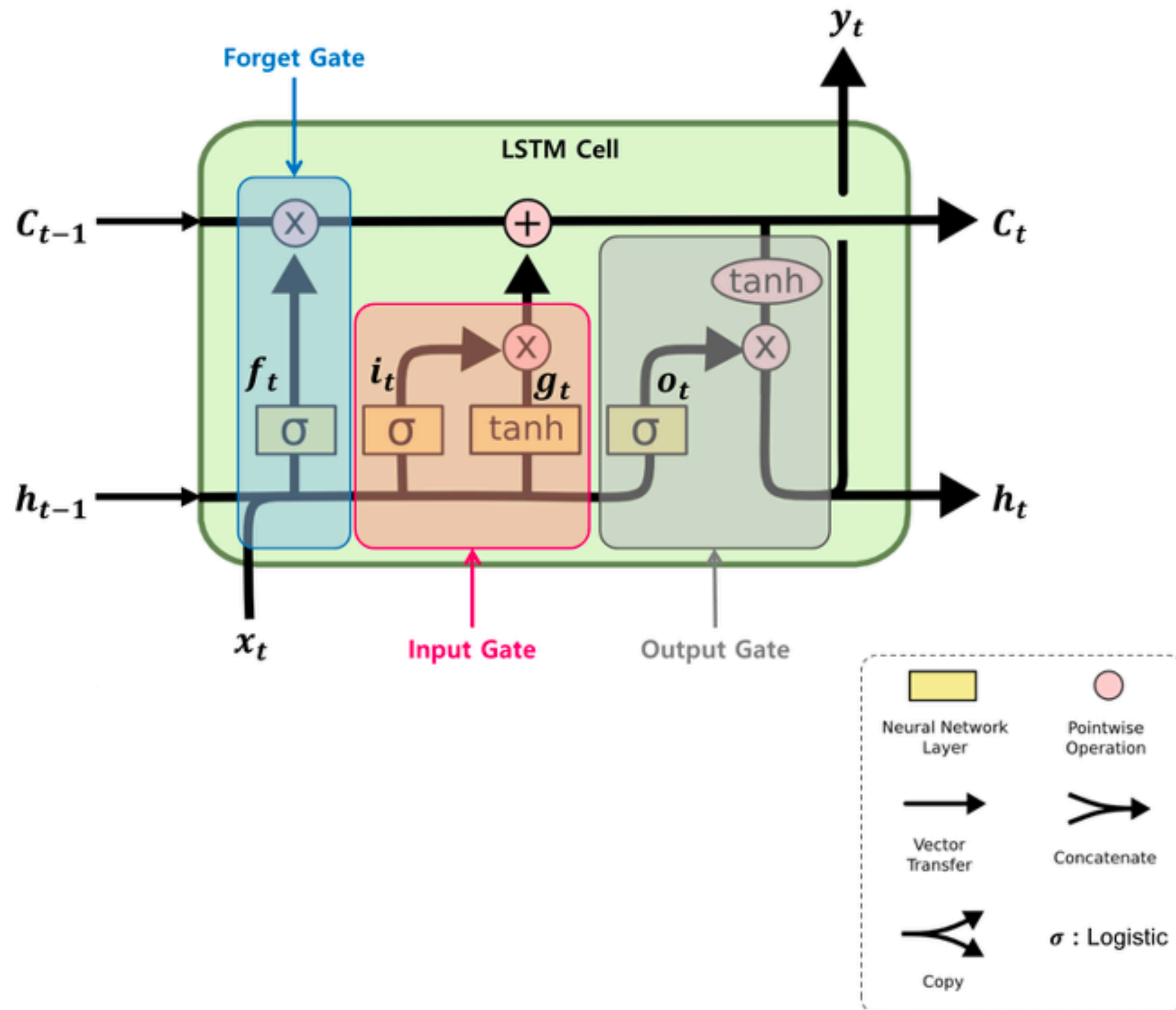
# LSTM

How it works?



# LSTM

How it works?



Input Gate

controls the flow of incoming information

Forget Gate

controls the amount of information from the previous memory cells

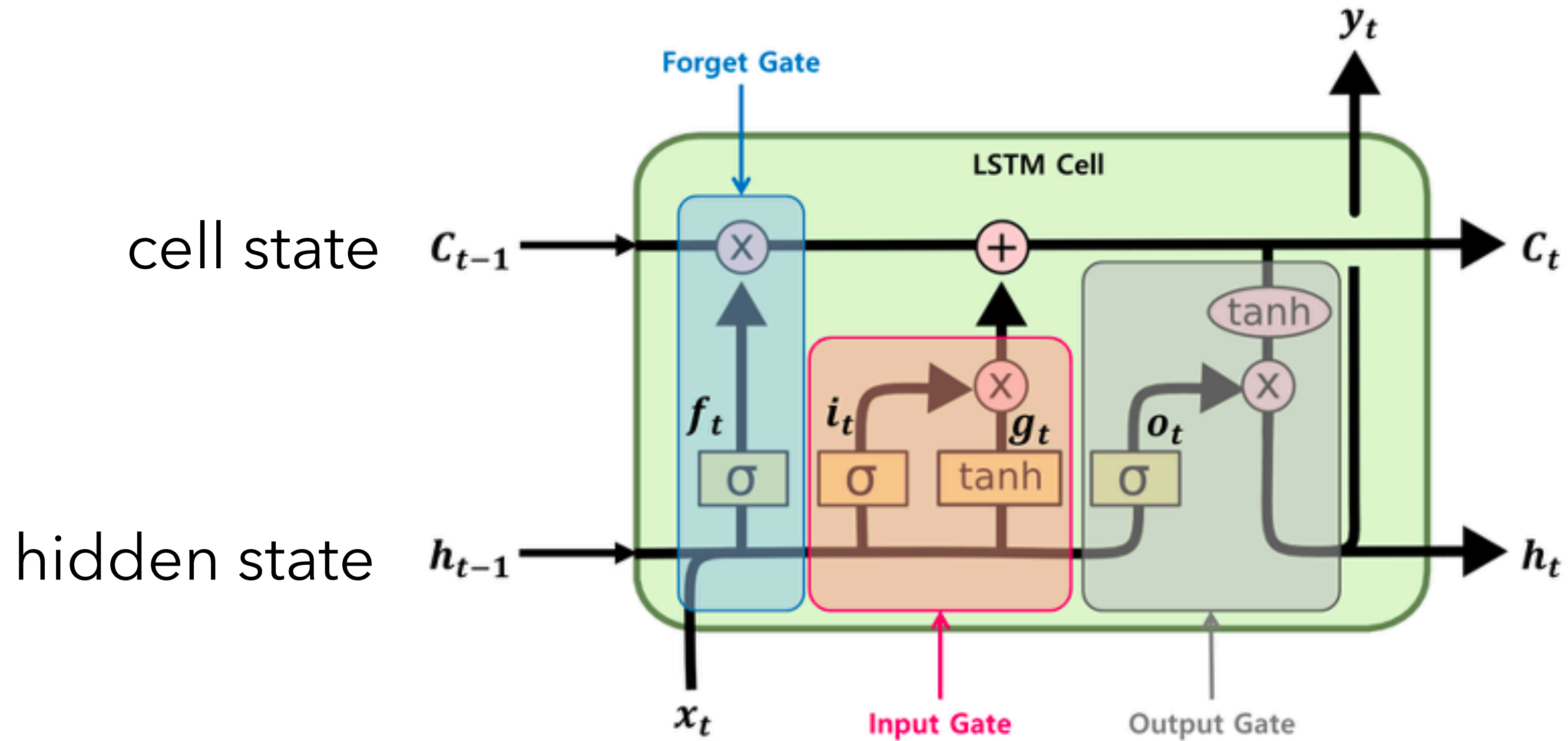
Output Gate

controls the flow of outgoing information



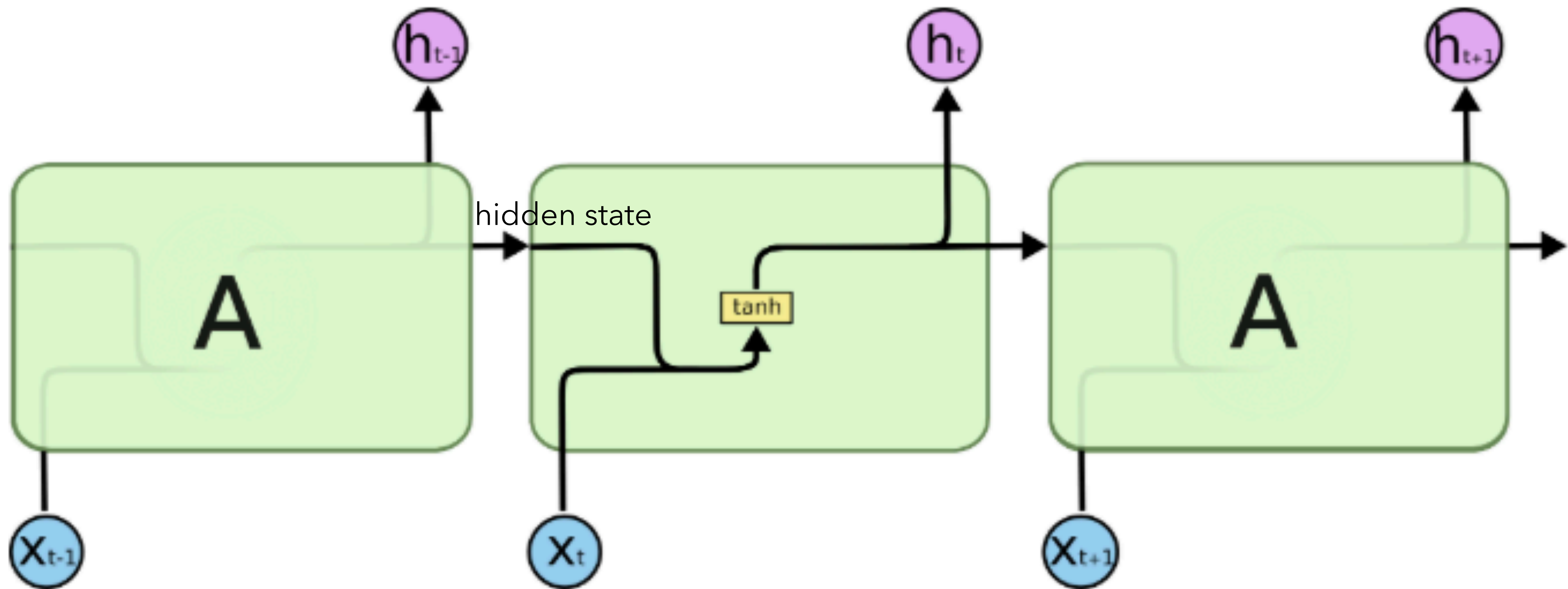
# LSTM

How it works?



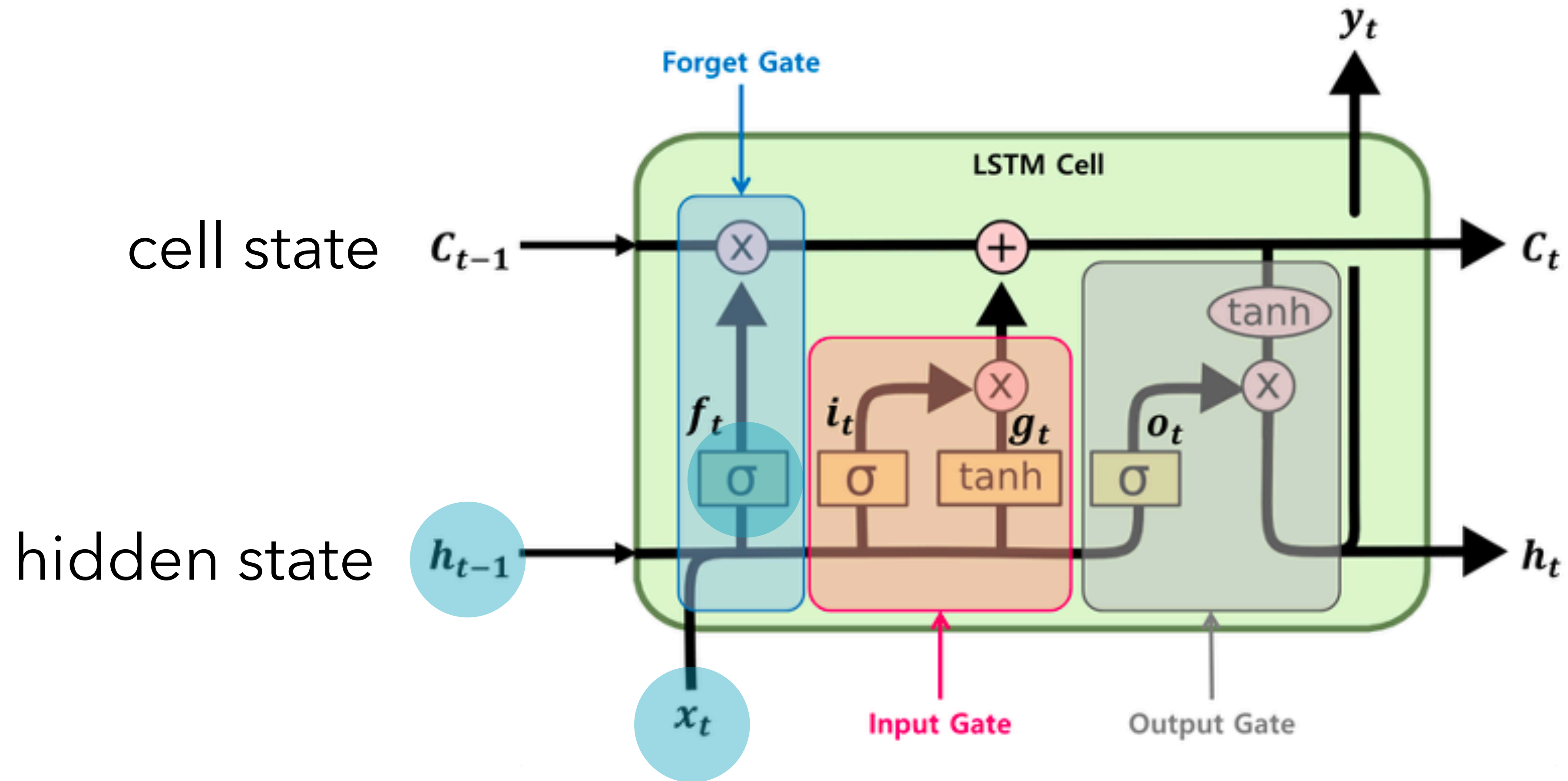
# LSTM

How it works?



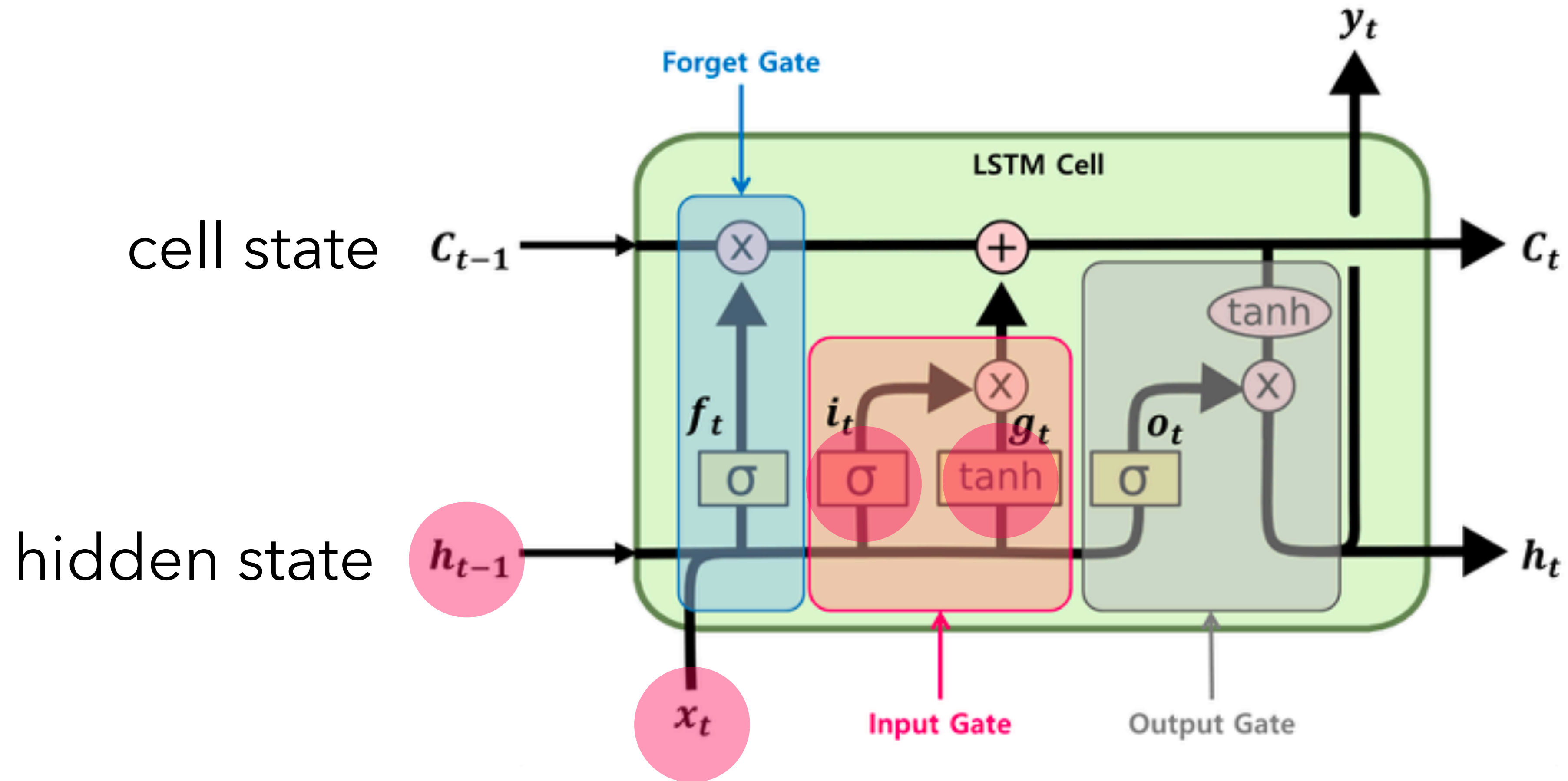
# LSTM

How it works? - forget gate



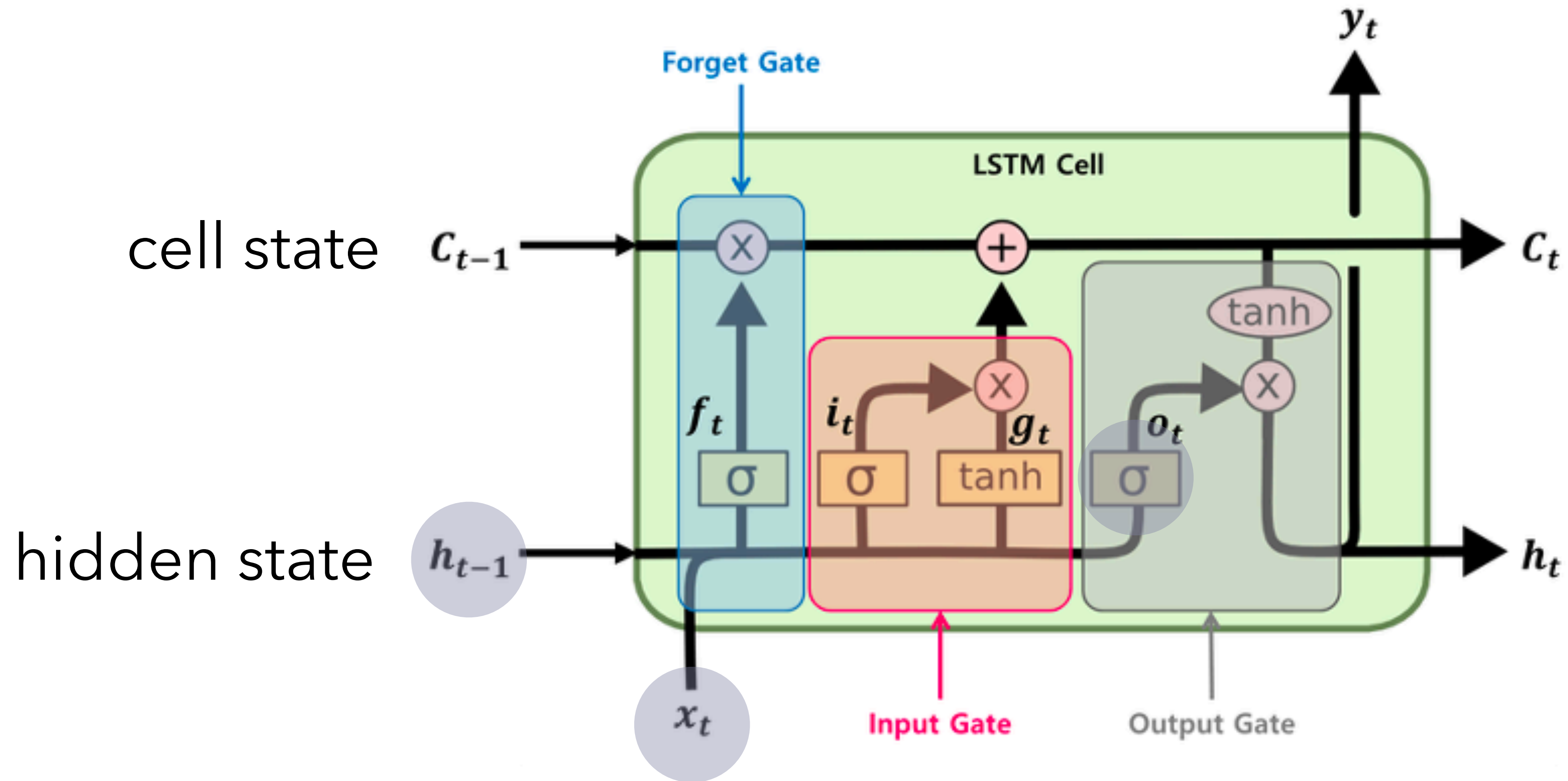
# LSTM

How it works? - input gate



# LSTM

How it works? - output gate

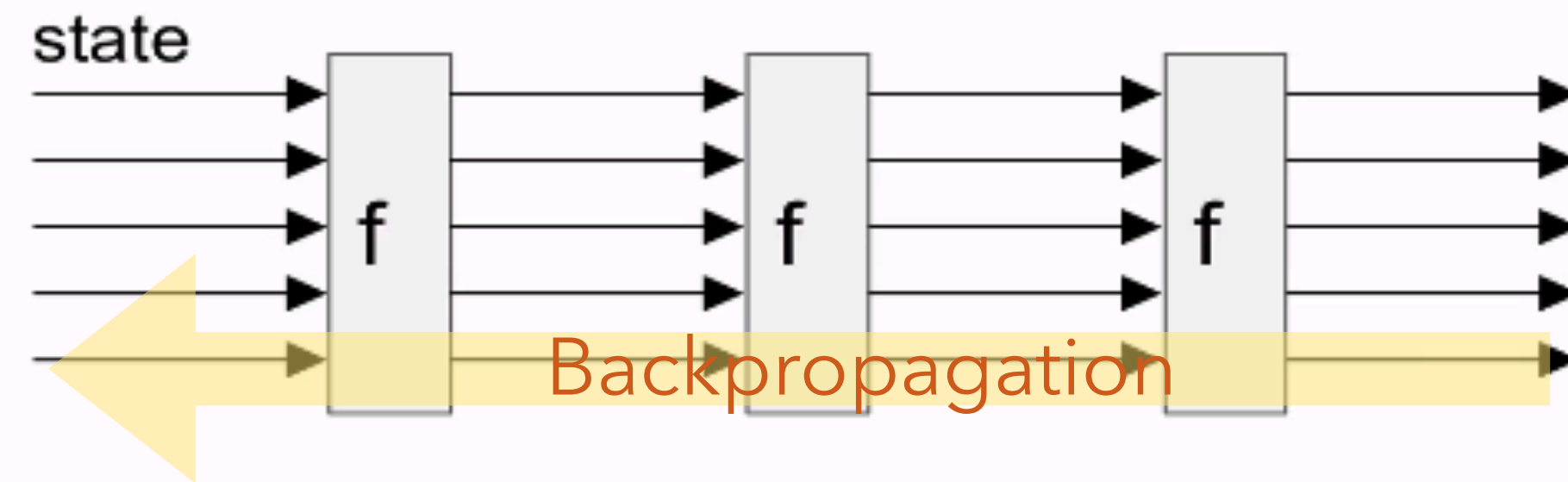




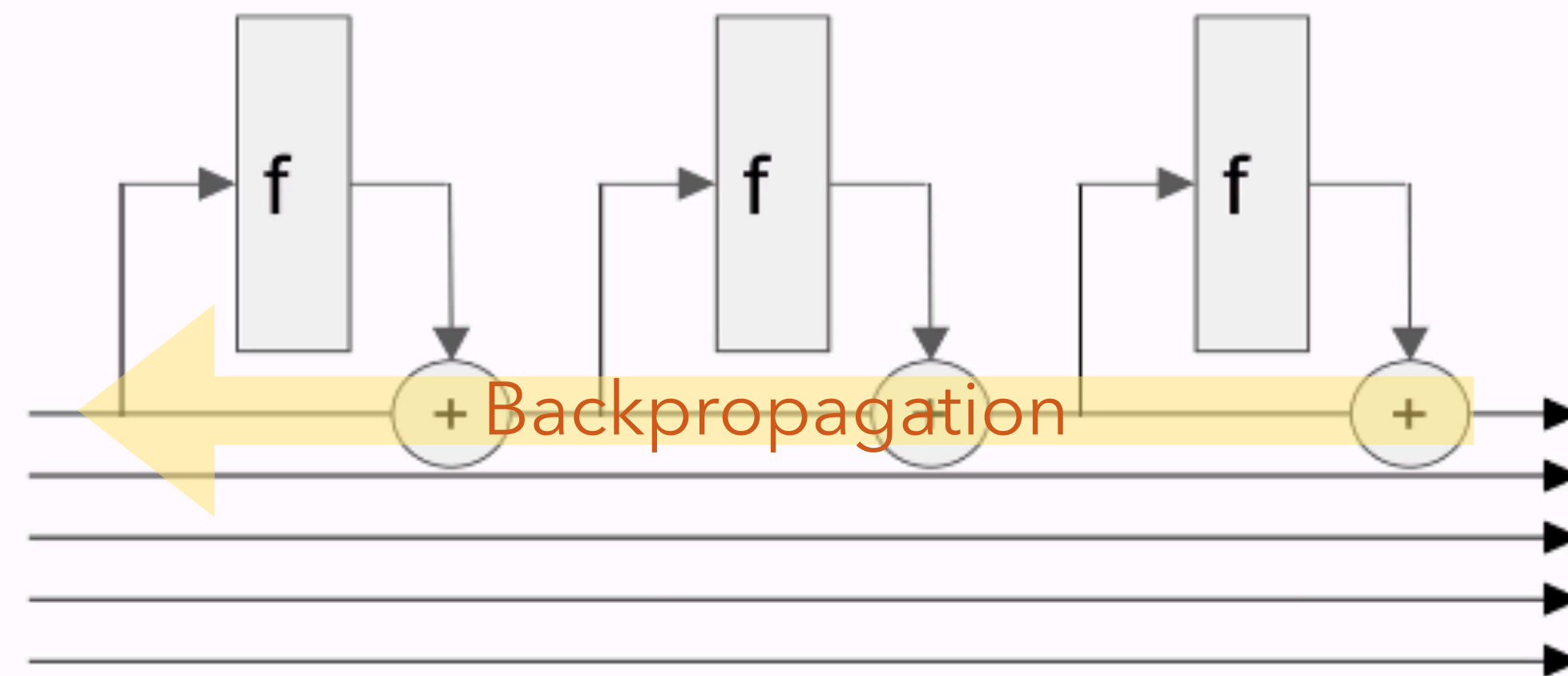
# LSTM

How to prevent vanishing gradient problem

RNN



LSTM  
(ignoring  
forget gates)

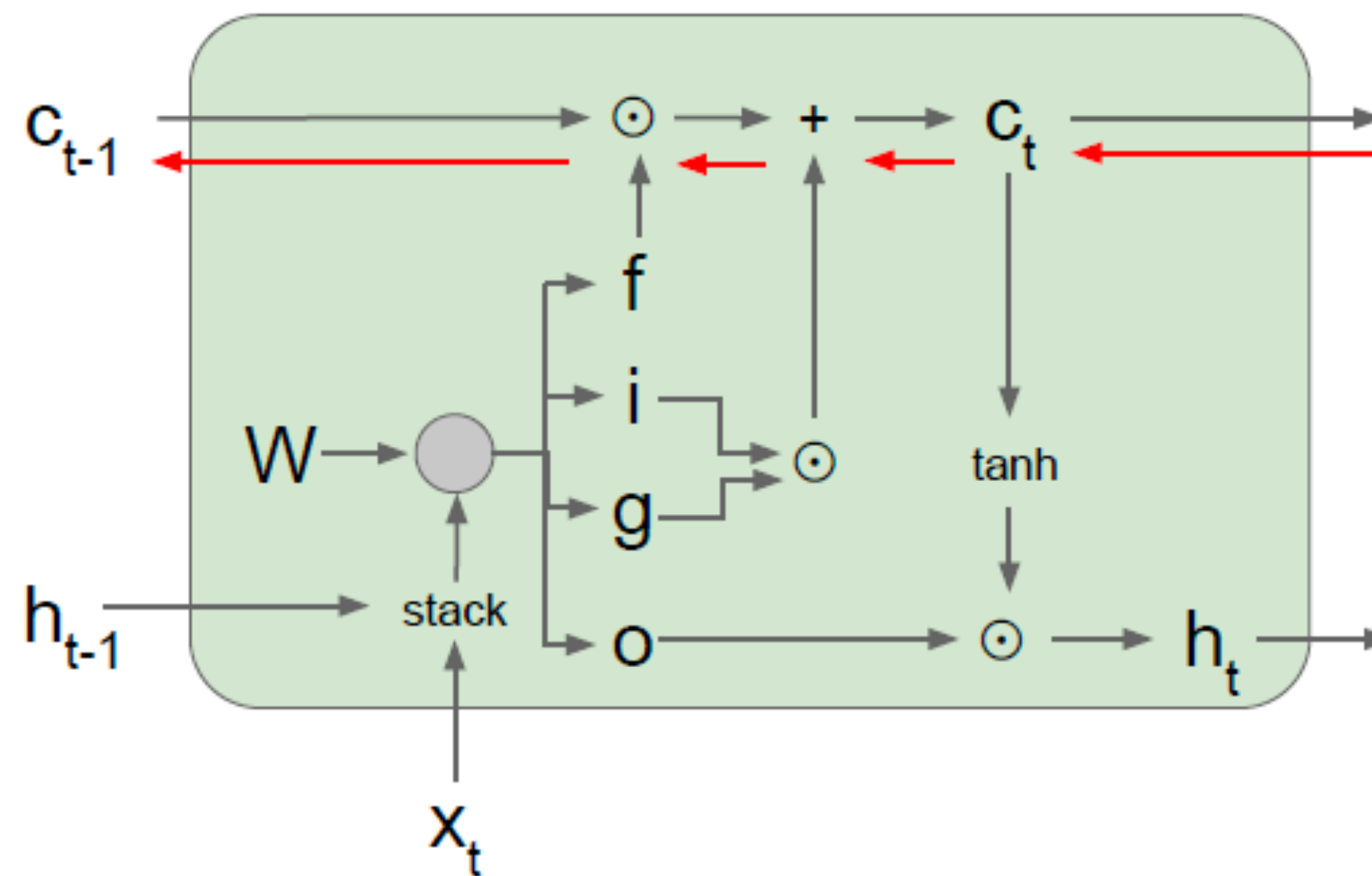


# LSTM

## How to prevent gradient vanishing problem

### Long Short Term Memory (LSTM): Gradient Flow

[Hochreiter et al., 1997]



Backpropagation from  $c_t$  to  $c_{t-1}$  only elementwise multiplication by  $f$ , no matrix multiply by  $W$

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

# Reference

<https://warm-uk.tistory.com/54>

<https://medium.com/tech-break/recurrent-neural-network-and-long-term-dependencies-e21773defd92>

<https://medium.com/@saurabh.rathor092/simple-rnn-vs-gru-vs-lstm-difference-lies-in-more-flexible-control-5f33e07b1e57>

<https://medium.com/analytics-vidhya/rnn-vs-gru-vs-lstm-863b0b7b1573>

<https://towardsdatascience.com/recurrent-neural-networks-b7719b362c65>