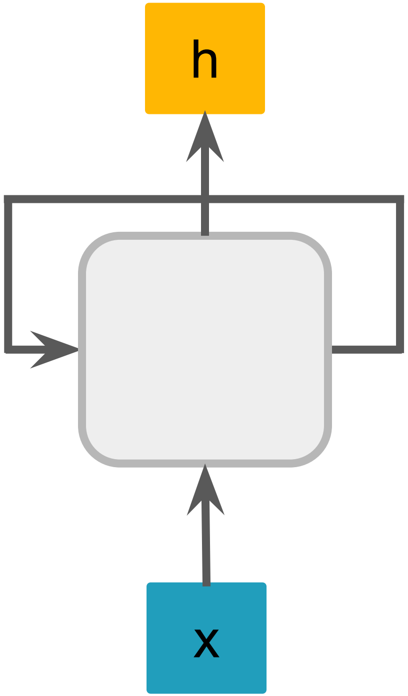
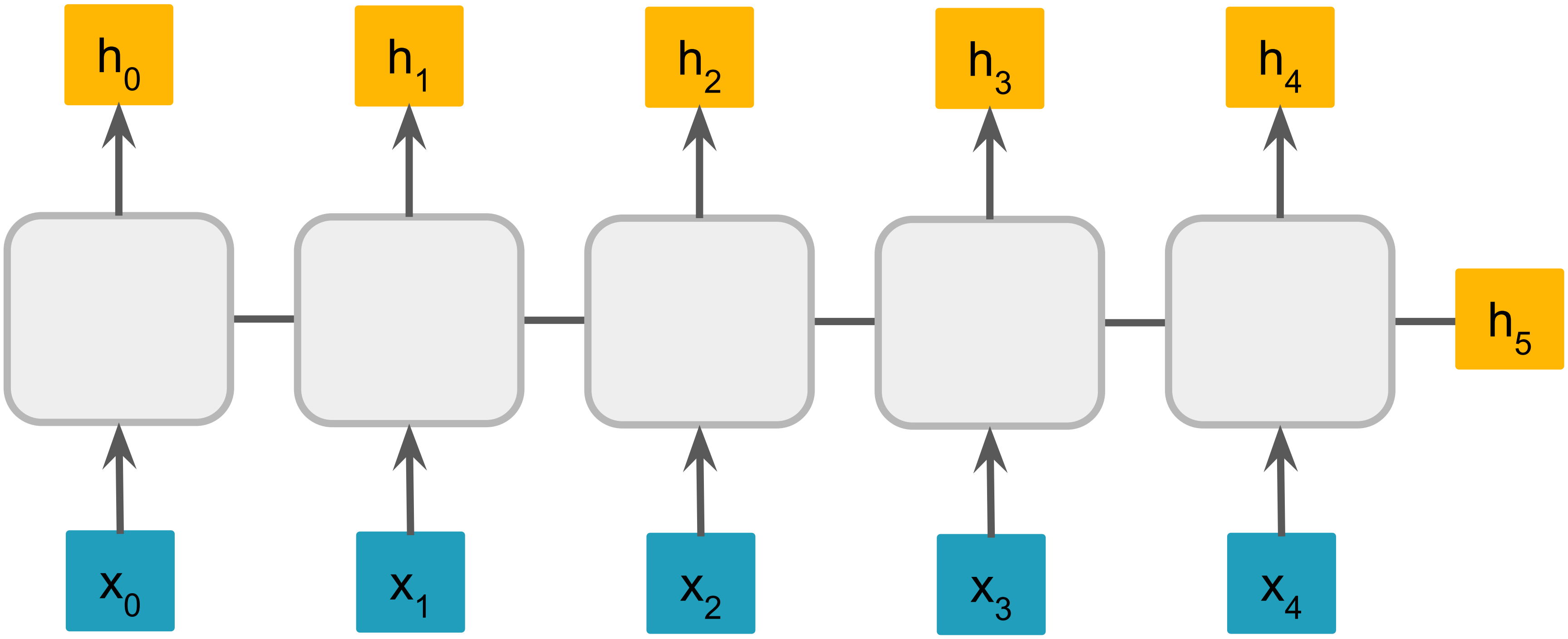
Recurrent Neural Networks II

Dr. Parlett-Pelleriti

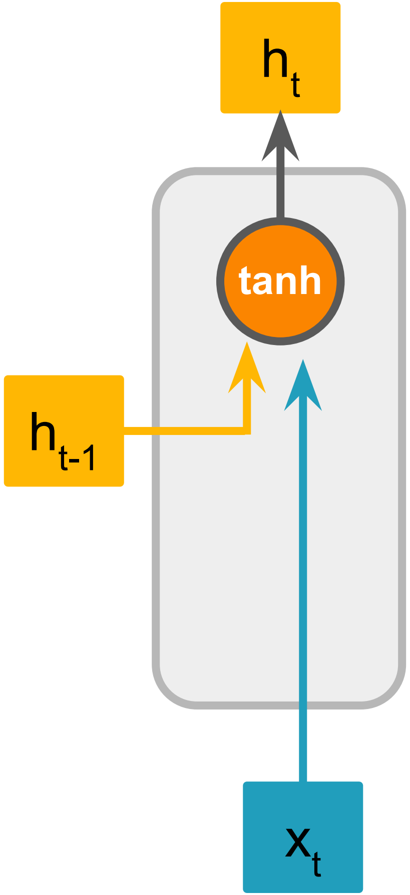
Recurrent Structure Generally



Recurrent Structure Unrolled

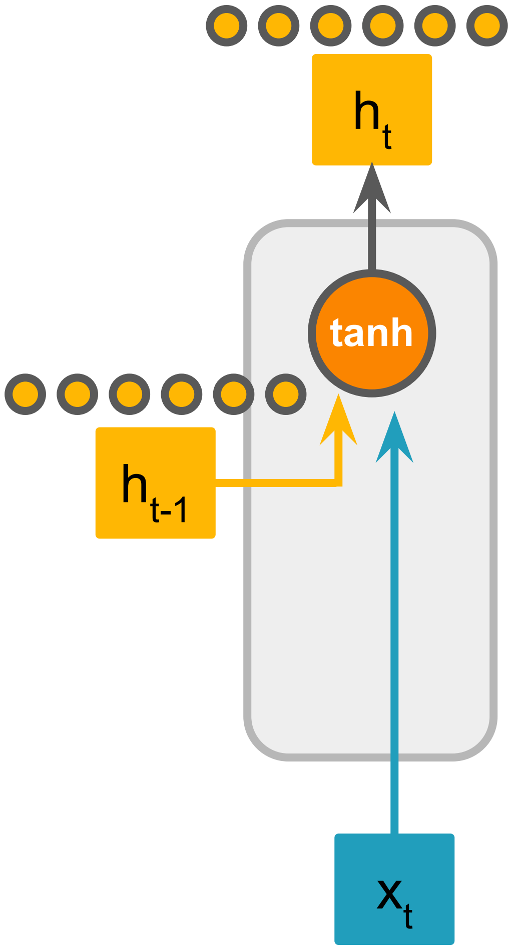


Simple RNN



$$h_t = \tanh(W \cdot [h_{t-1}, x_t] + b)$$

Simple RNN



$$h_t = \tanh(W \cdot [h_{t-1}, x_t] + b)$$

Problem with RNN Architecture

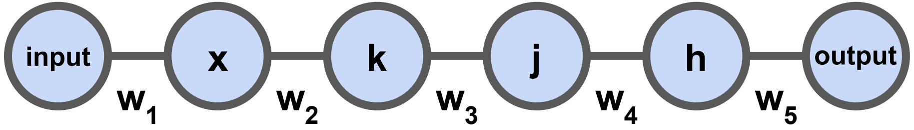
Gradient Descent

$$\overbrace{w_i'}^{\text{new weight}} = \underbrace{w_i}_{\text{old weight}} - \overbrace{\lambda}^{\text{learning rate}} \underbrace{\frac{\partial L}{\partial w_i}}_{\text{derivative of loss w.r.t. weight}}$$

$$w_i' = w_i - \lambda \frac{\partial L}{\partial w_i}$$

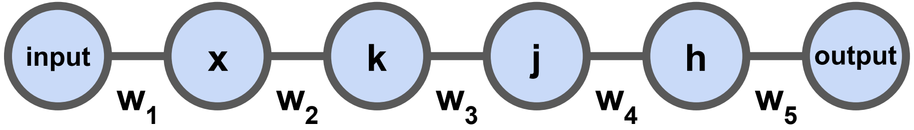
The Vanishing Gradient Problem

What happens when the gradient is small?



The Vanishing Gradient Problem

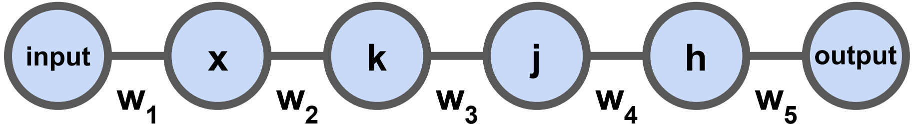
What happens when the gradient is small?



$$\frac{\partial Loss}{\partial w1} = \frac{\partial Loss}{\partial \text{output}} * \frac{\partial \text{output}}{\partial h} * \frac{\partial h}{\partial j} * \frac{\partial j}{\partial k} * \frac{\partial k}{\partial x} * \frac{\partial x}{\partial w_1}$$

The Vanishing Gradient Problem

What happens when the gradient is small?

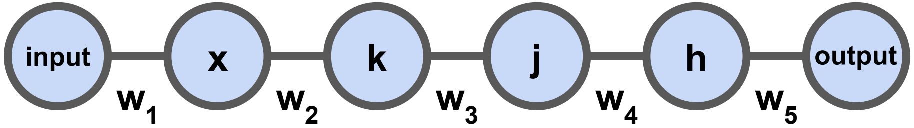


$$\frac{\partial Loss}{\partial w1} = \frac{\partial Loss}{\partial \mathrm{output}} * \frac{\partial \mathrm{output}}{\partial h} * \frac{\partial h}{\partial j} * \frac{\partial j}{\partial k} * \frac{\partial k}{\partial x} * \frac{\partial x}{\partial w_1}$$

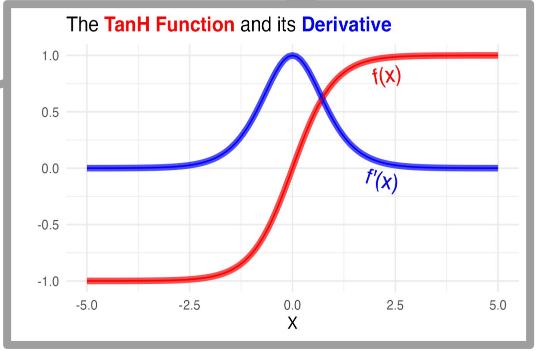
What happens if a lot of these are small?

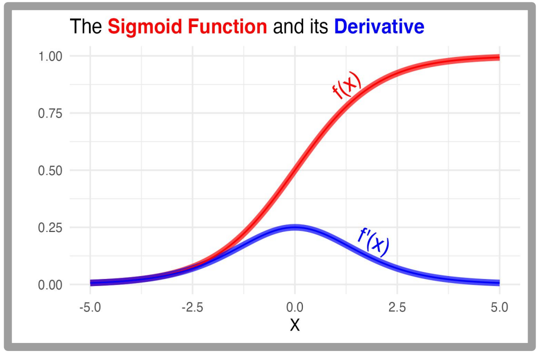
The Vanishing Gradient Problem

What happens when the gradient is small?



$$\frac{\partial Loss}{\partial w1} = \frac{\partial Loss}{\partial \text{output}} * \frac{\partial \text{output}}{\partial h} * \frac{\partial h}{\partial j} * \frac{\partial j}{\partial k} * \frac{\partial k}{\partial x} * \frac{\partial x}{\partial w_1}$$





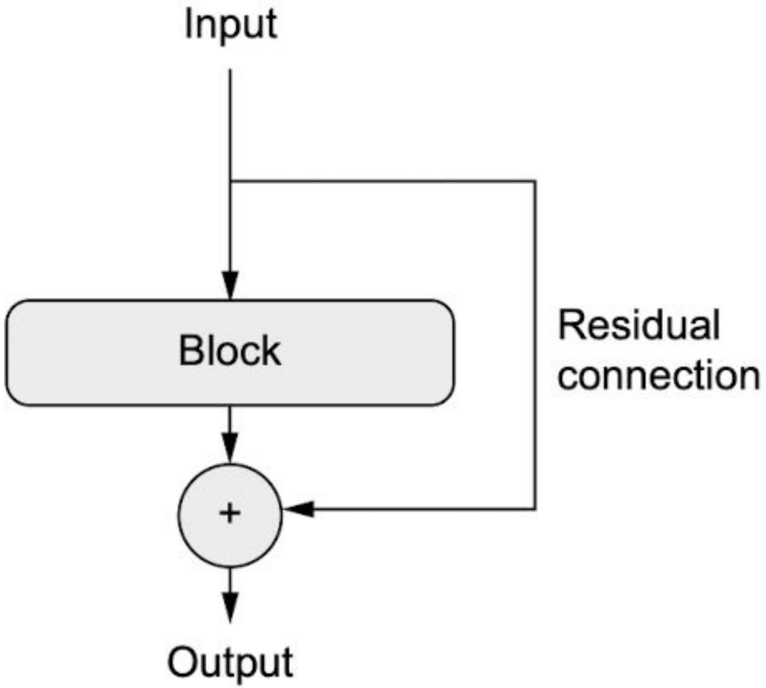
What happens if a lot of these are small?

What would this look like?

Your weights stop updating (much)

Long Short Term Memory

Residual Connections



CNN Flashback

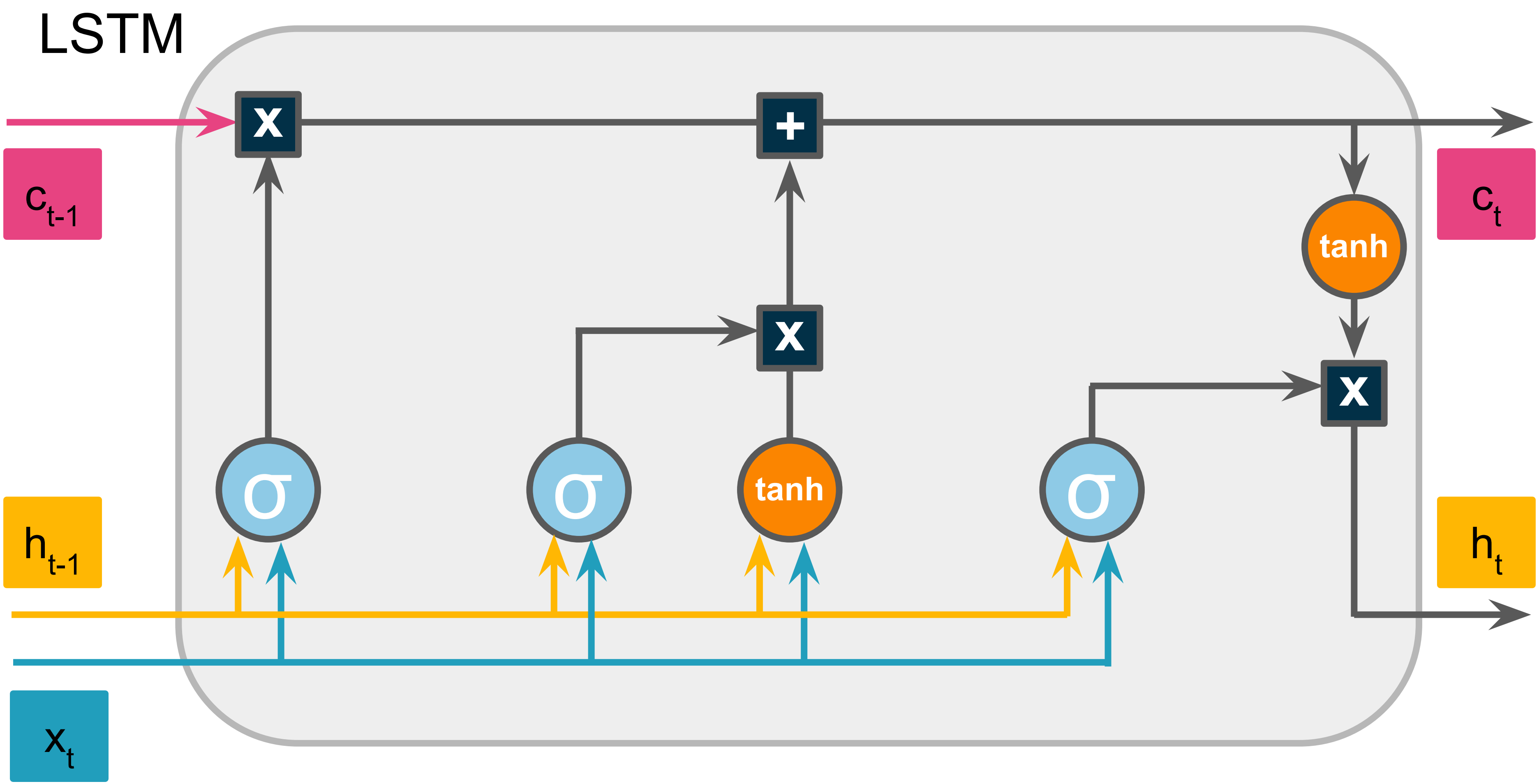
Image from: Deep Learning with Python (Chollet)

Cell State

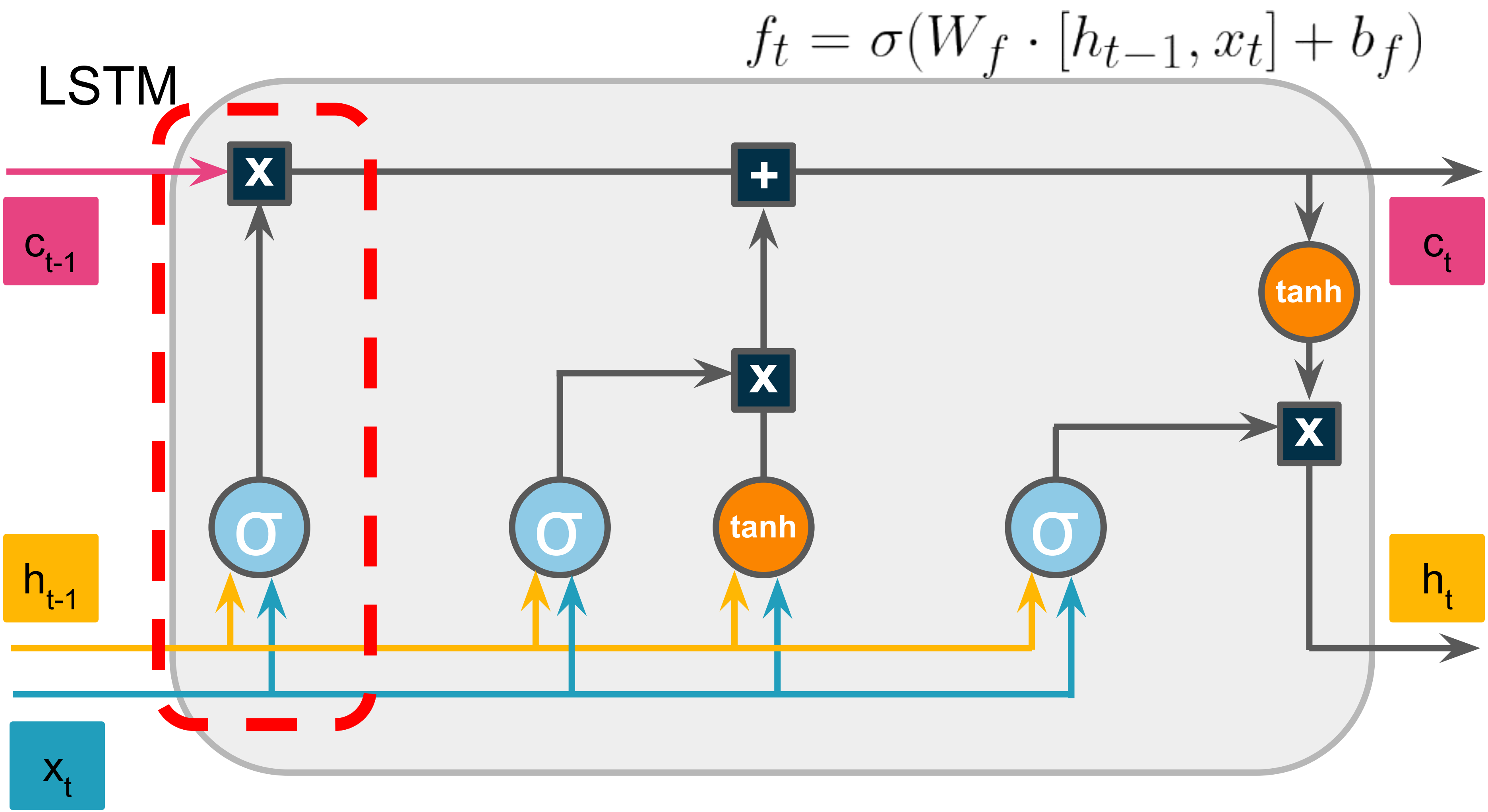
My friends are going to the market today to see Bob, he is a really cool butcher.

Cell State

My friends are going to the market today to see Bob, he is a really cool butcher.

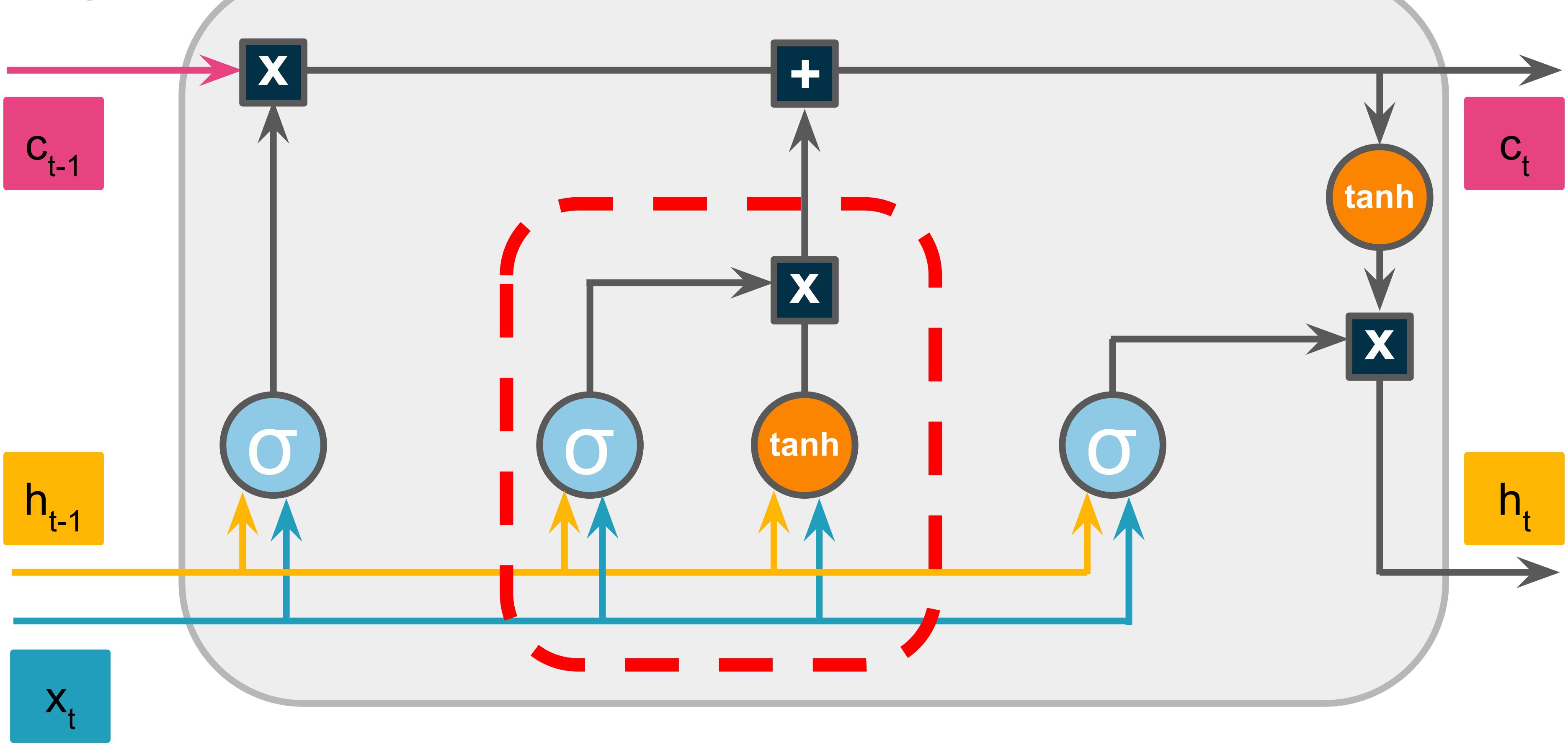


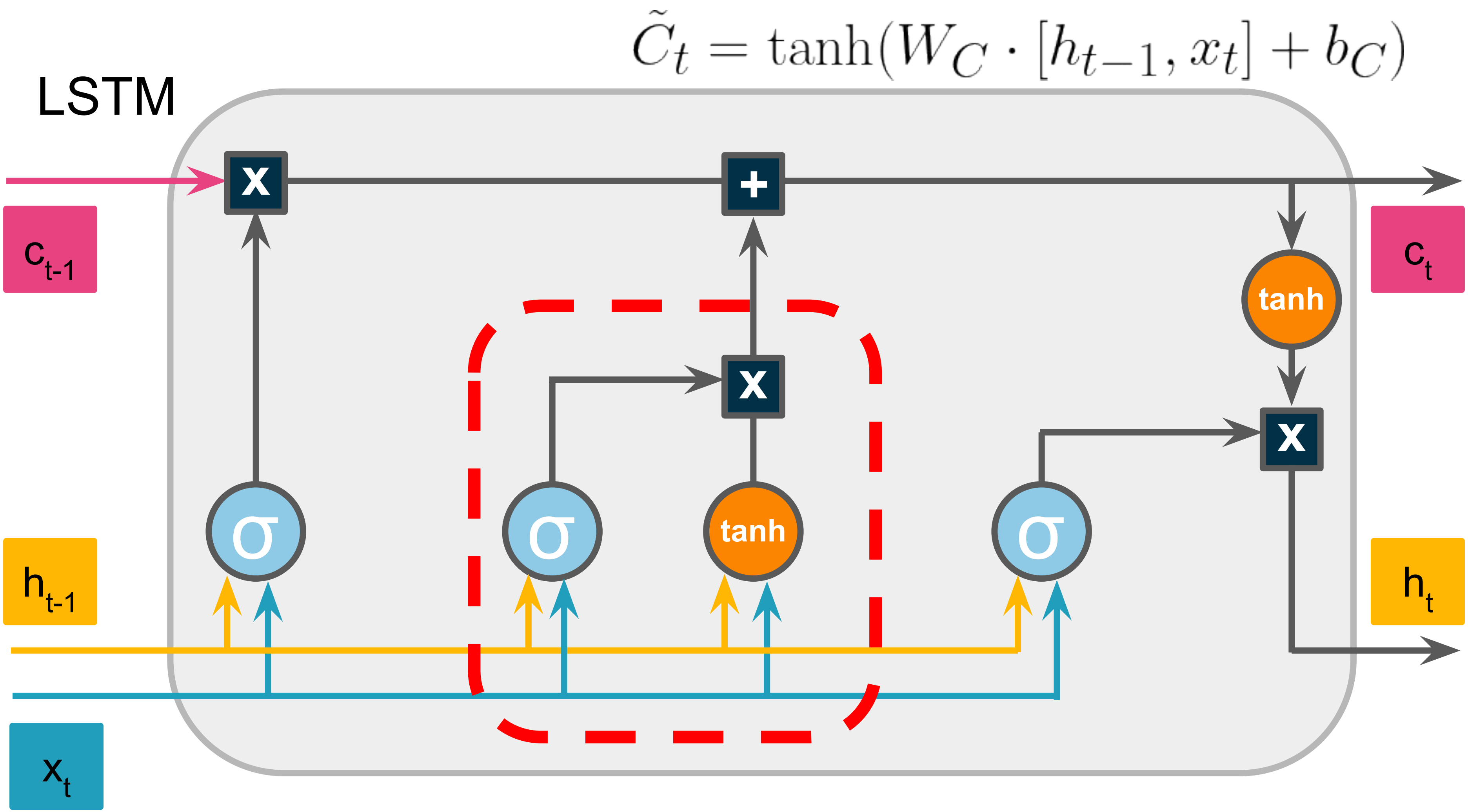
LSTM
x
+
c_{t-1}
c_t
tanh
x
x
σ
σ
tanh
σ
h_{t-1}
h_t
x_t

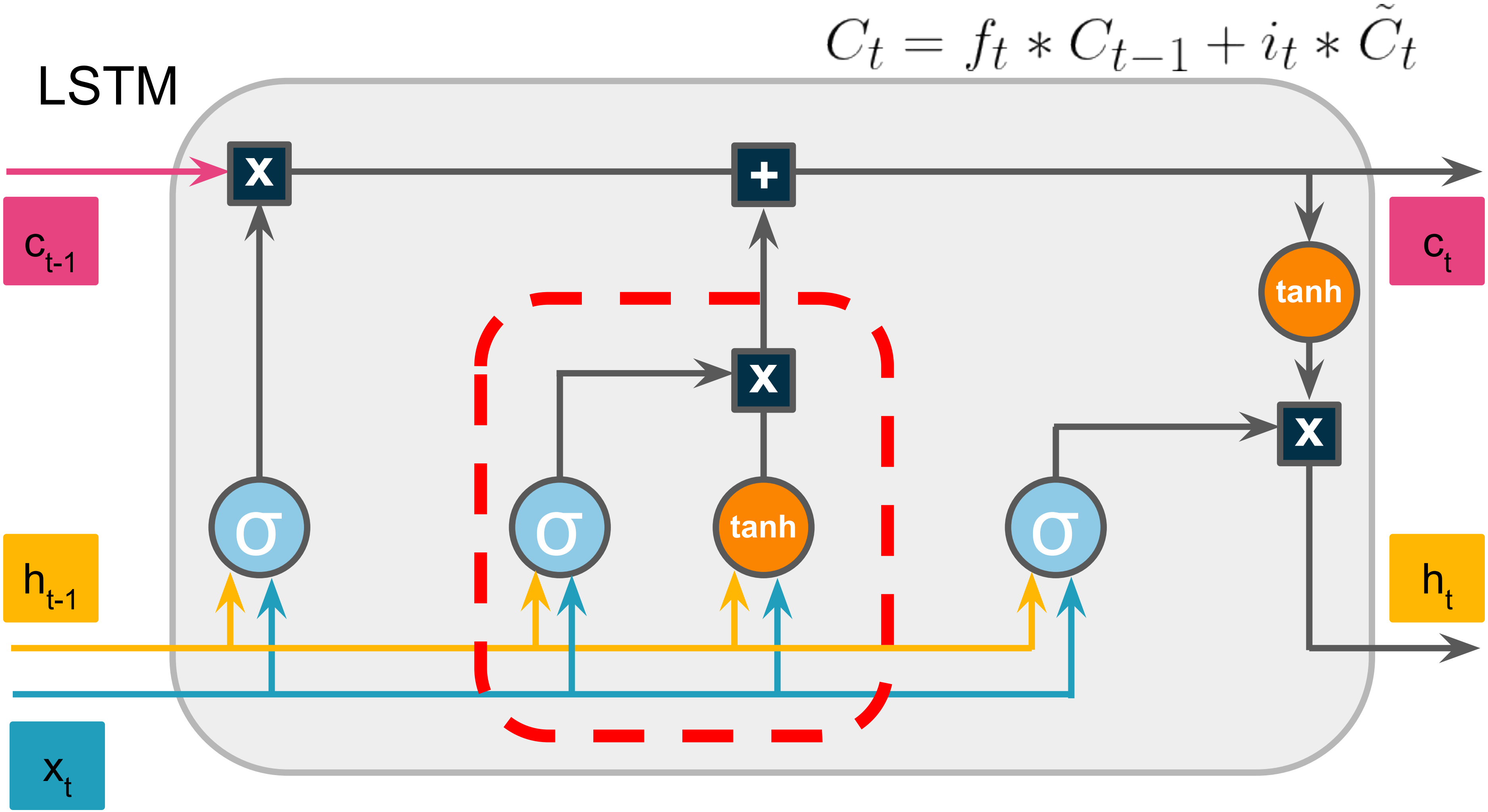

$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$
LSTM
x
+
x
x
tanh
σ
σ
tanh
σ
c_{t-1}
h_{t-1}
x_t
c_t
h_t

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

LSTM

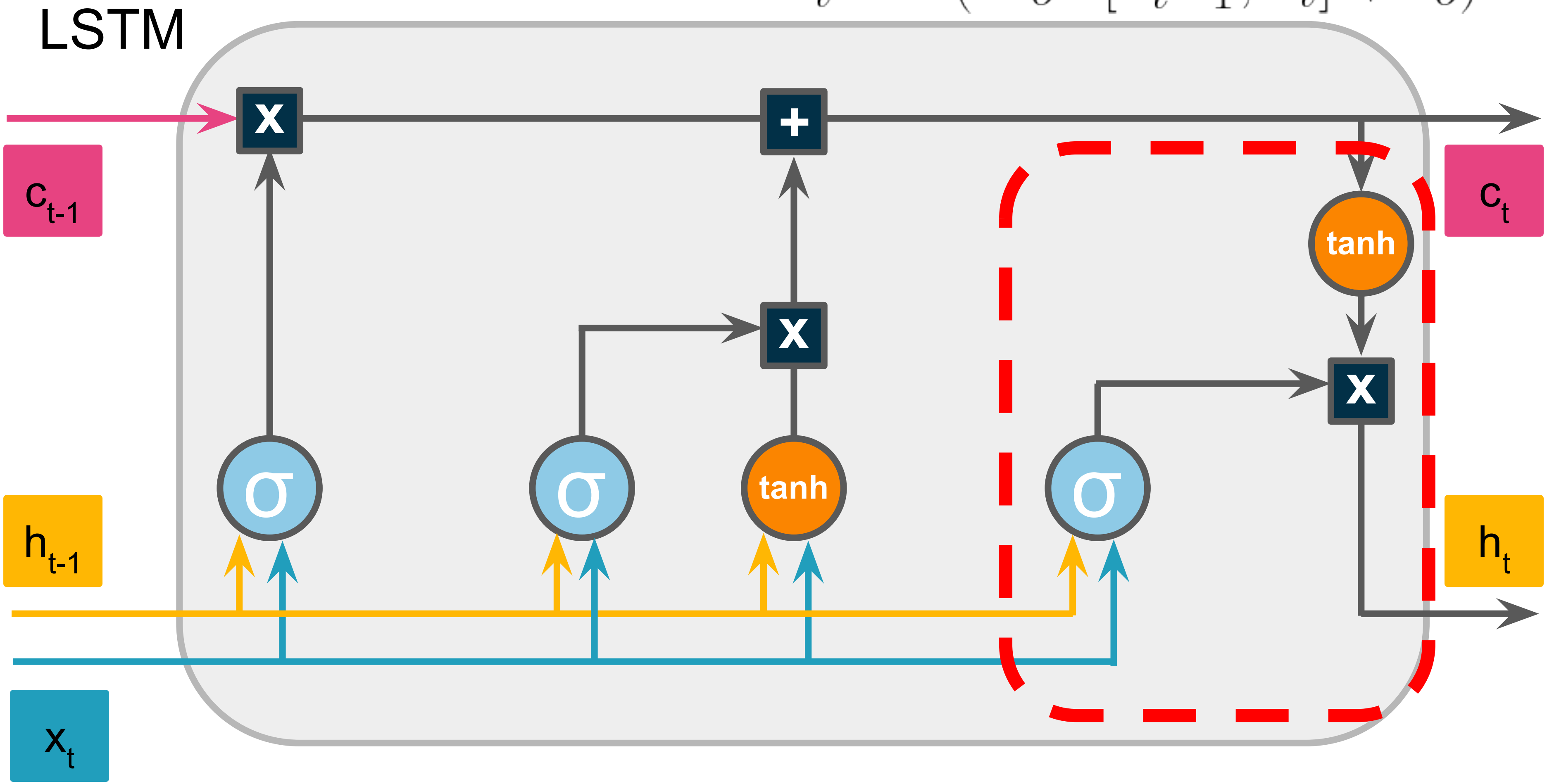


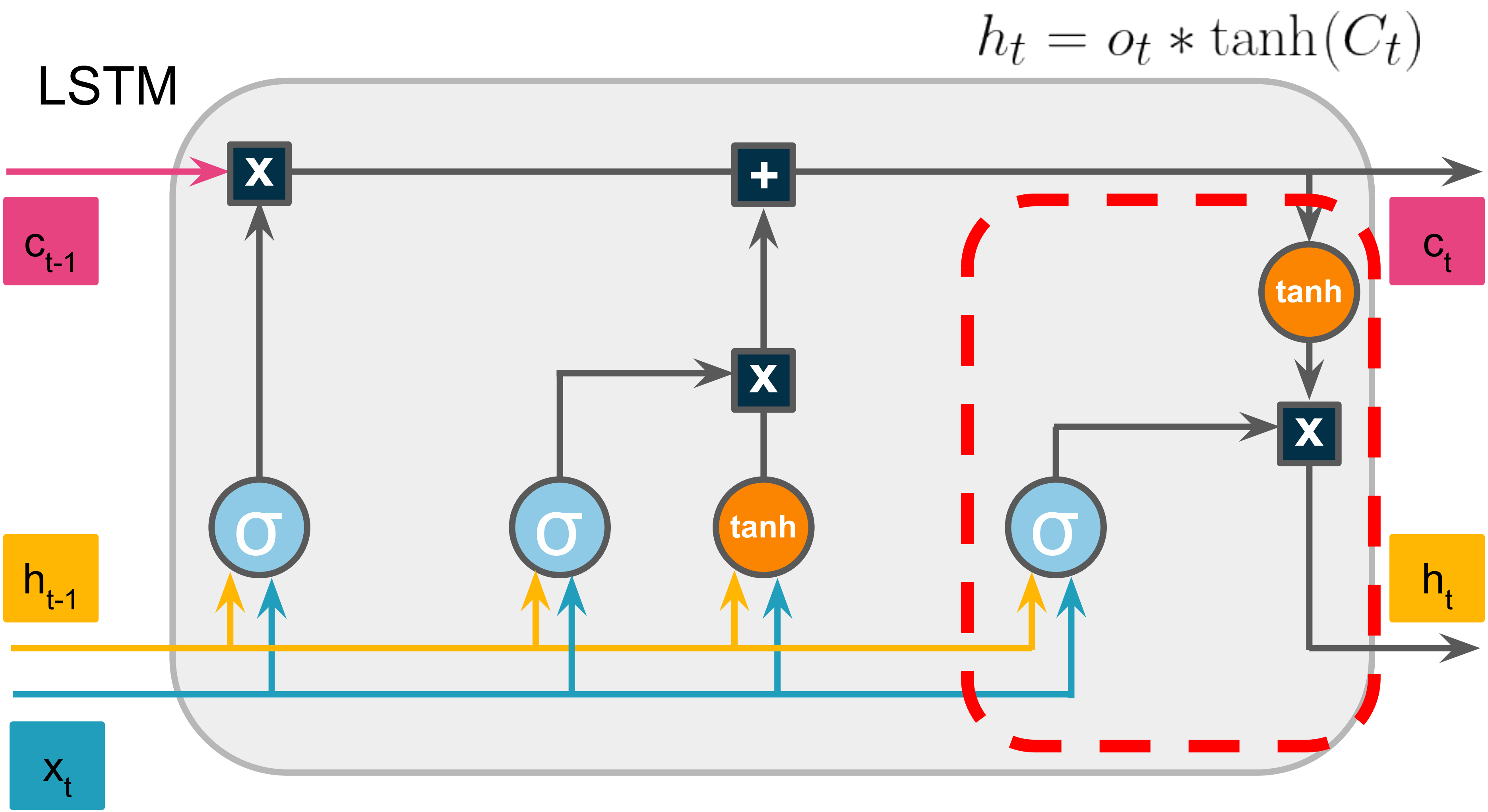

$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$
LSTM
x
+
c_{t-1}
c_t
tanh
x
x
σ
σ
tanh
σ
h_{t-1}
h_t
x_t



$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$
LSTM
c_{t-1}
h_{t-1}
x_t
c_t
h_t
x
+
x
x
σ
σ
tanh
σ
tanh

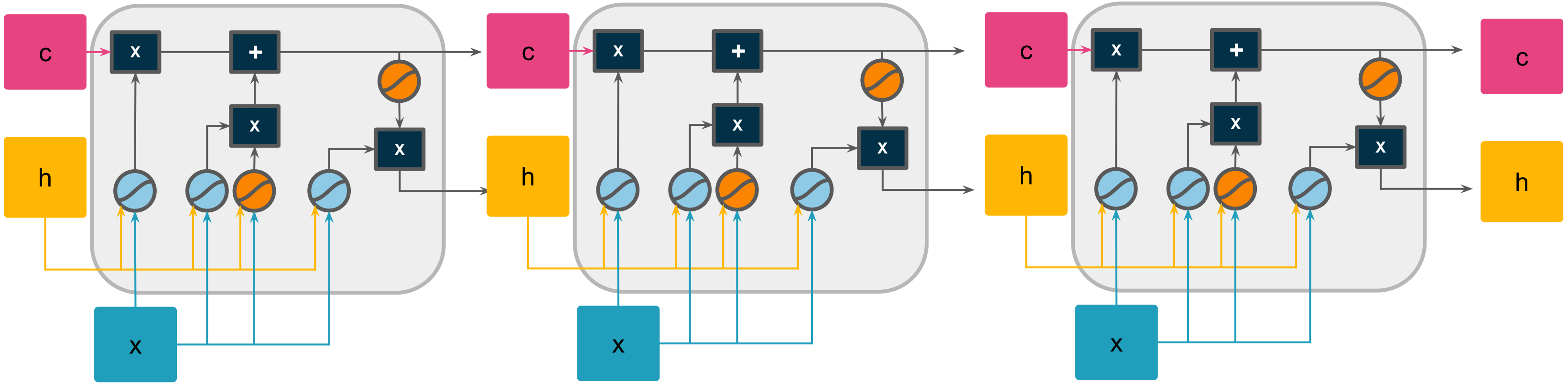
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$





$h_t = o_t * \tanh(C_t)$
LSTM
x
+
x
x
c_{t-1}
c_t
tanh
tanh
σ
σ
σ
h_{t-1}
h_t
x_t

LSTM Unrolled



Deep LSTMs

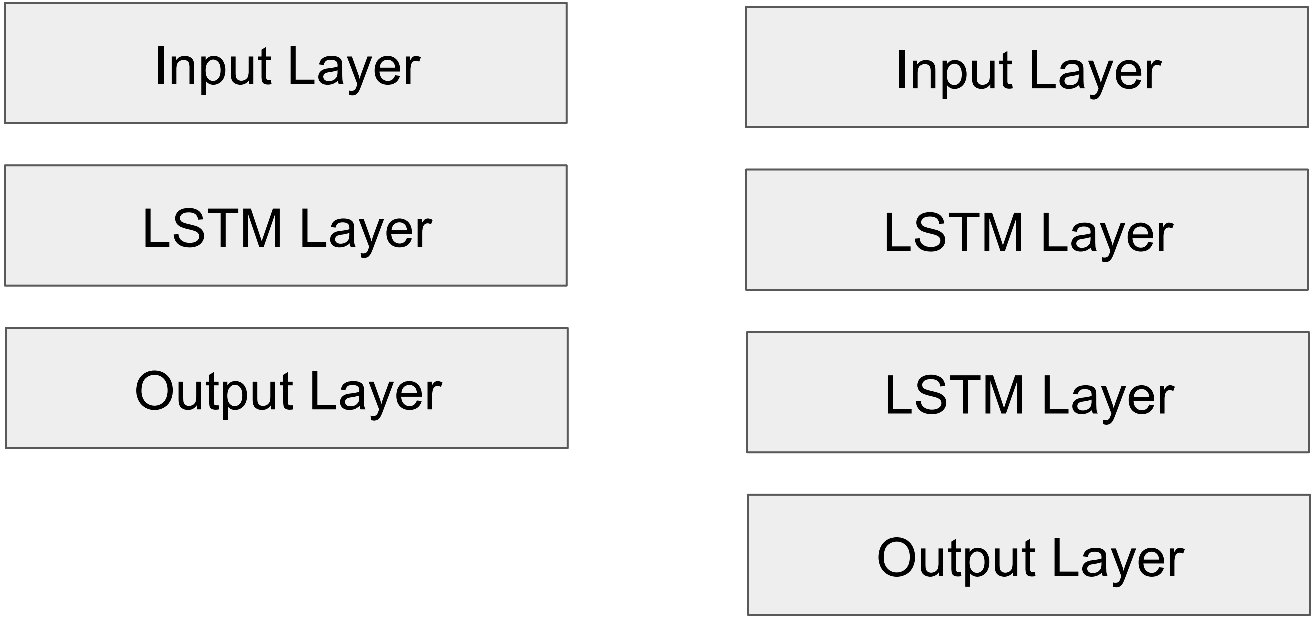
Deep LSTMs

Input Layer

LSTM Layer

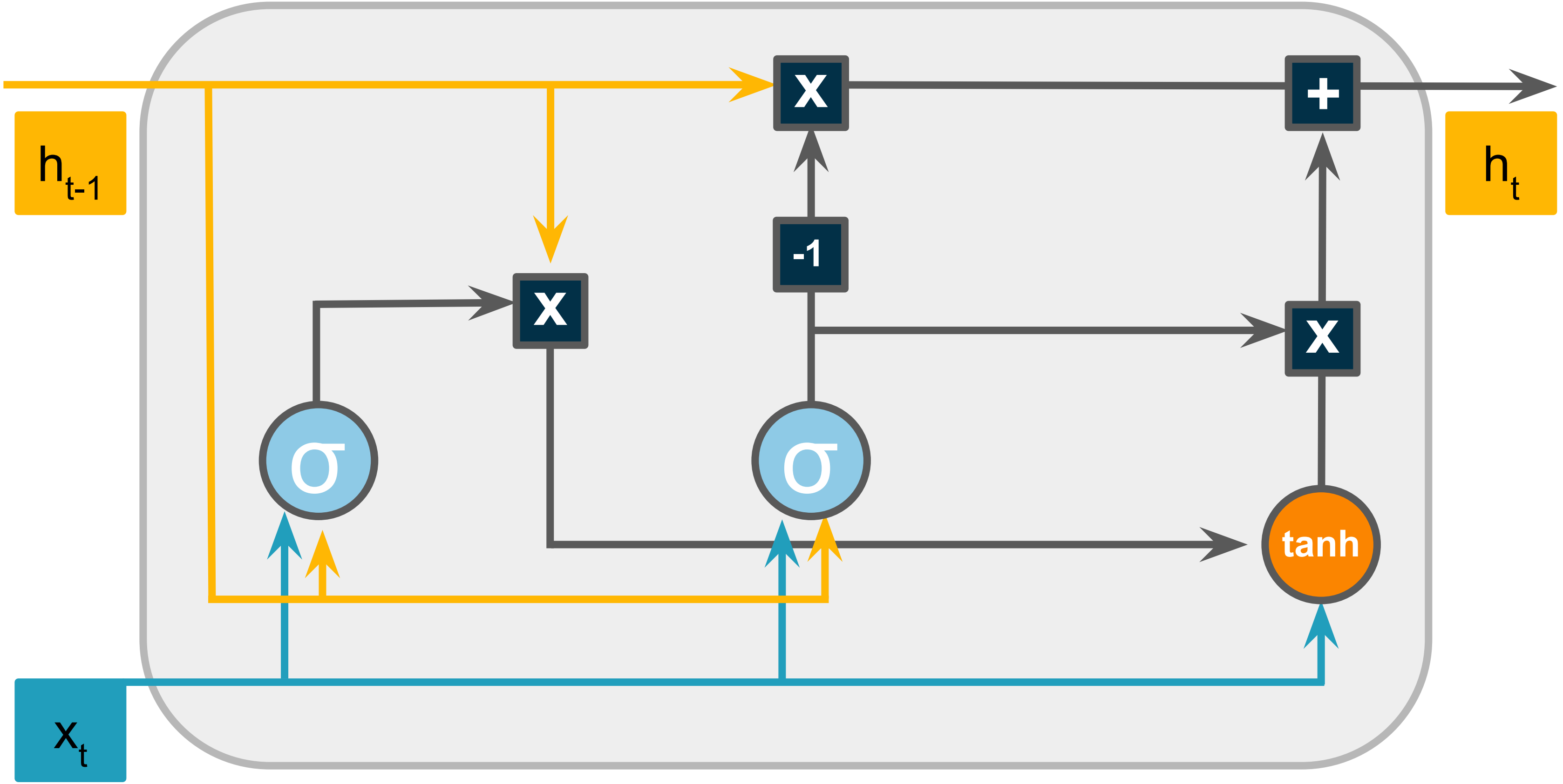
Output Layer

Deep LSTMs


Input Layer
LSTM Layer
Output Layer
Input Layer
LSTM Layer
LSTM Layer
Output Layer

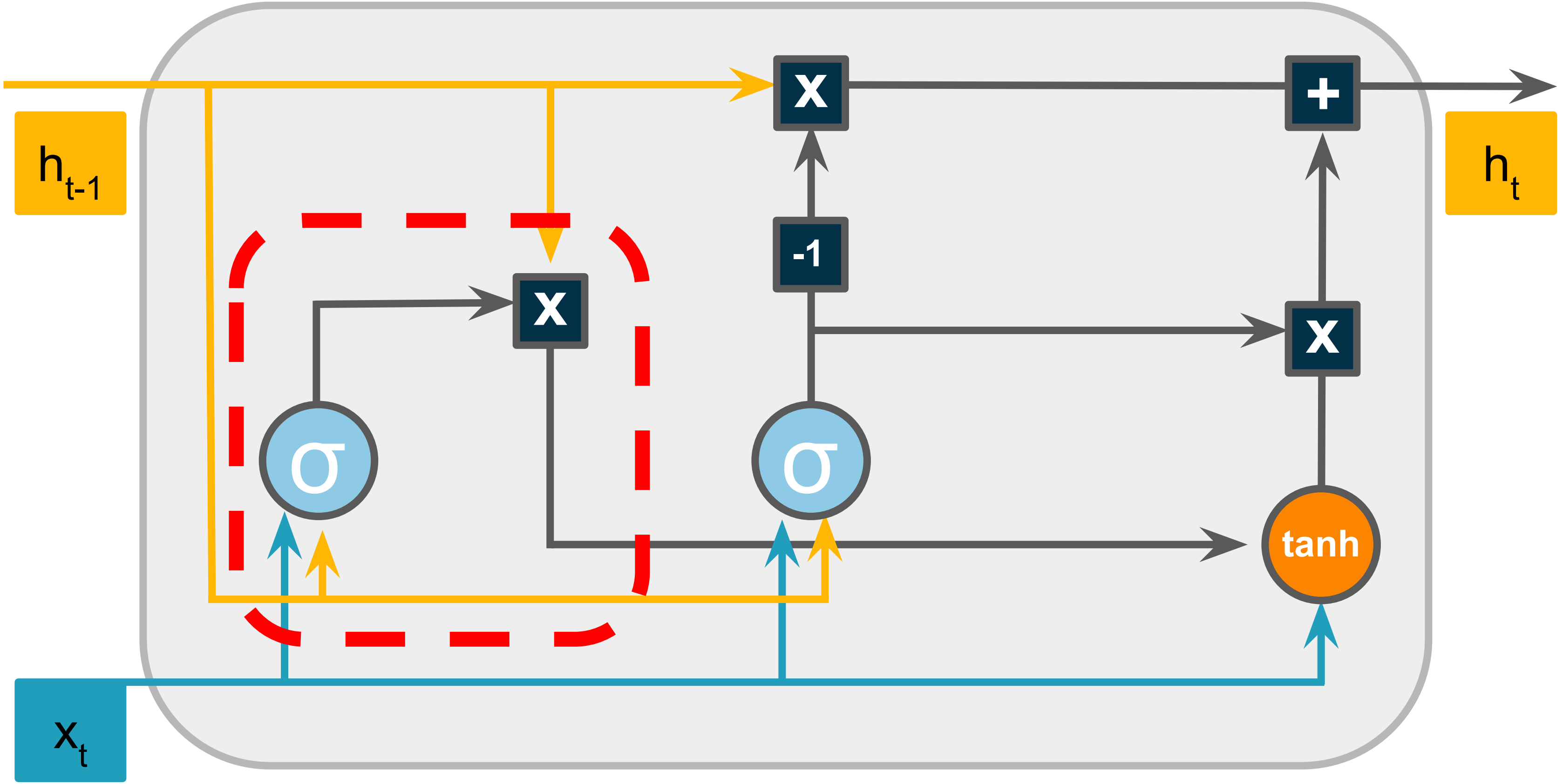
Gated Recurrent Units

Gated Recurrent Unit



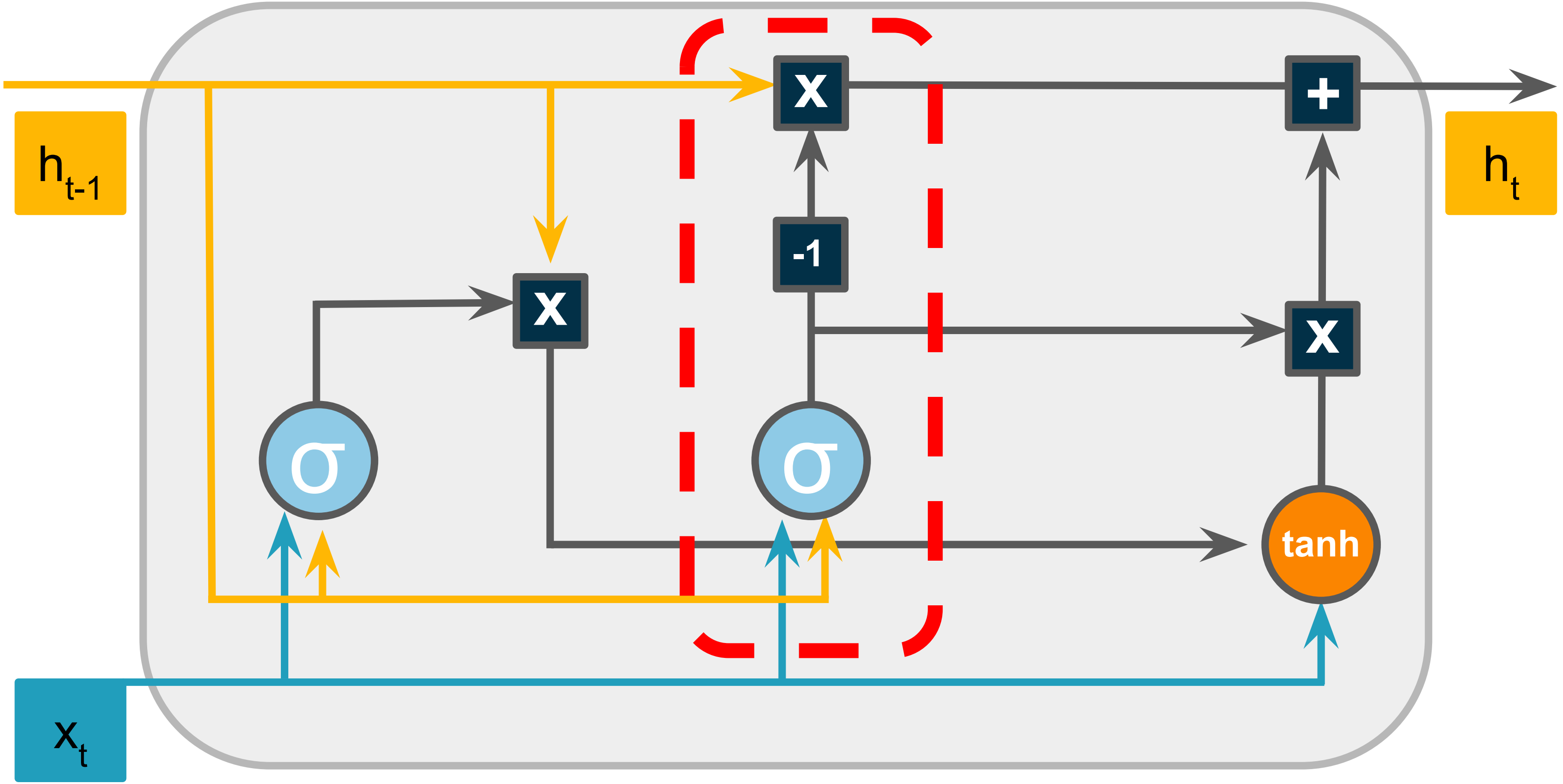
Gated Recurrent Unit

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$$

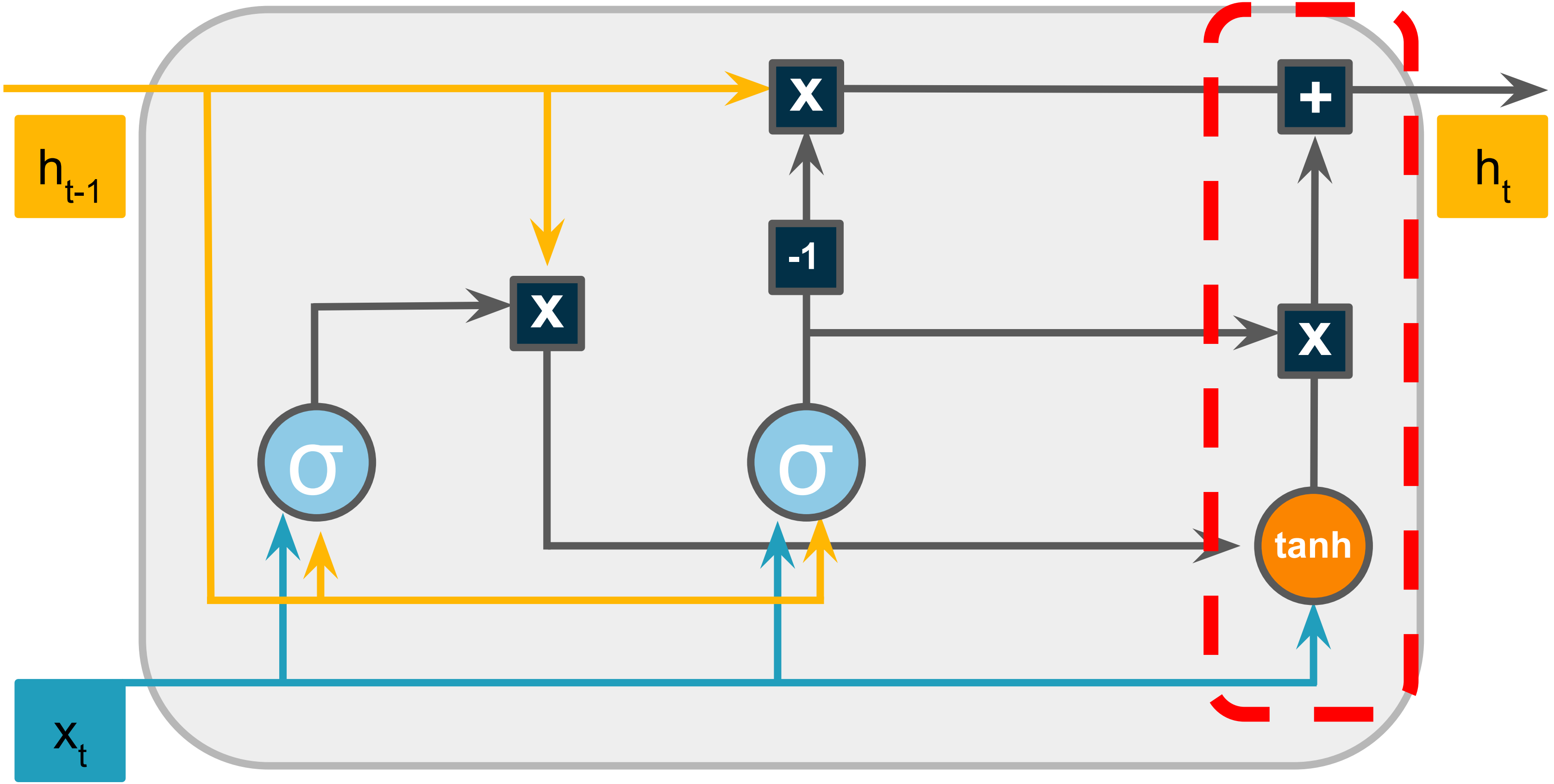


Gated Recurrent Unit

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$$

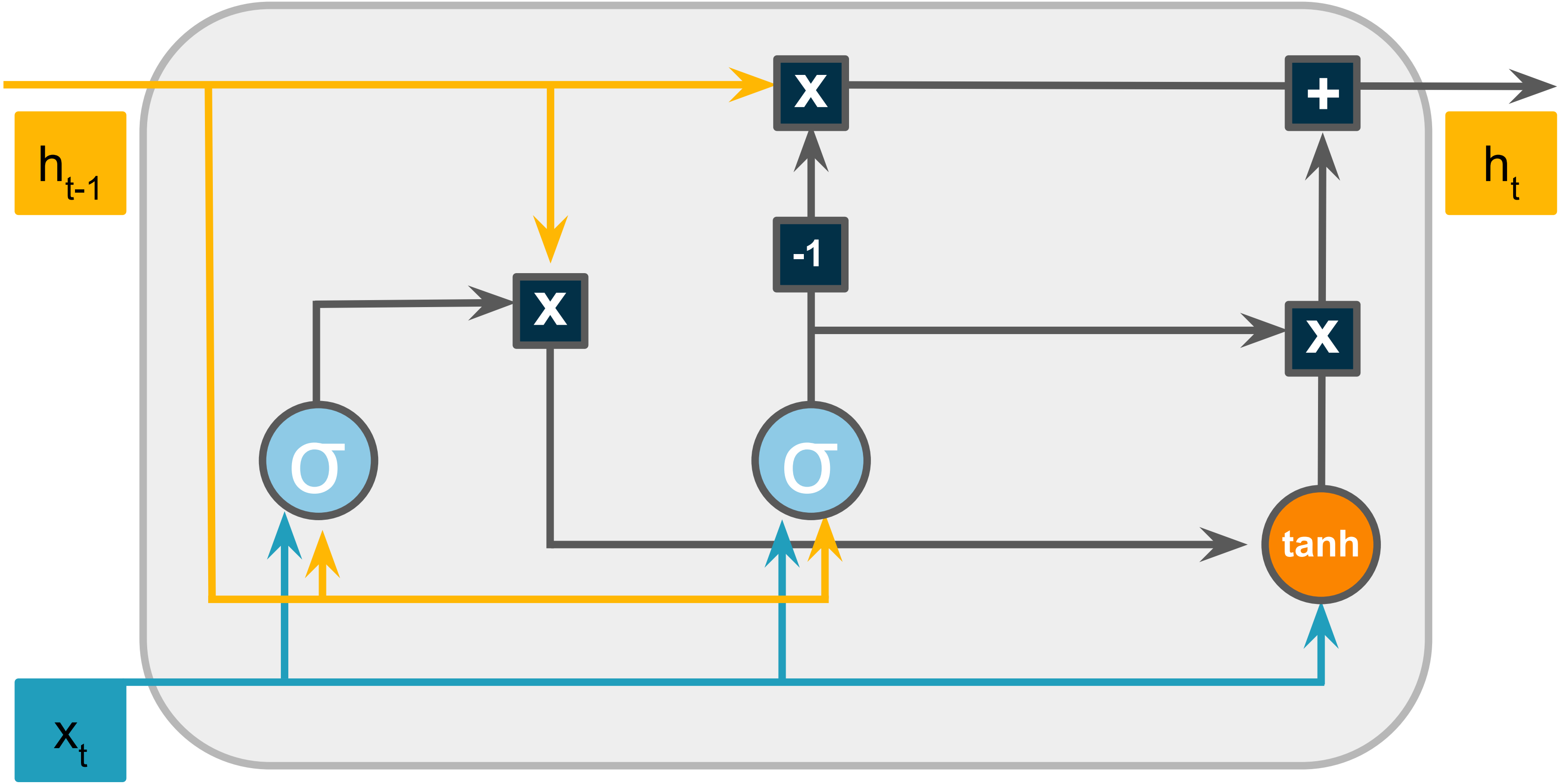


Gated Recurrent Unit $\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t] + b)$

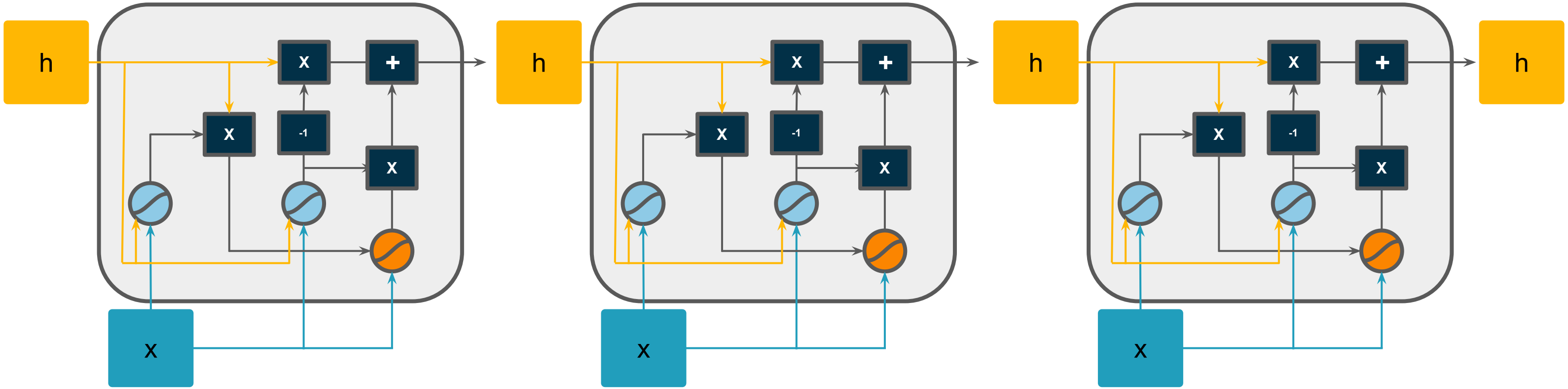


Gated Recurrent Unit

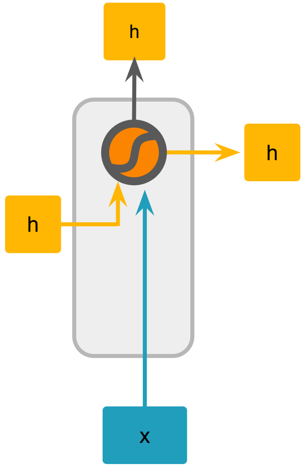
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

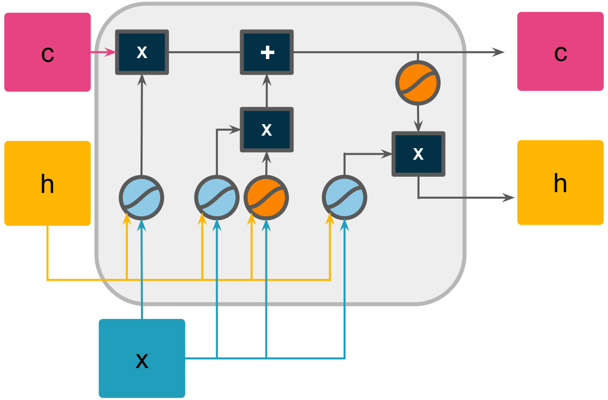


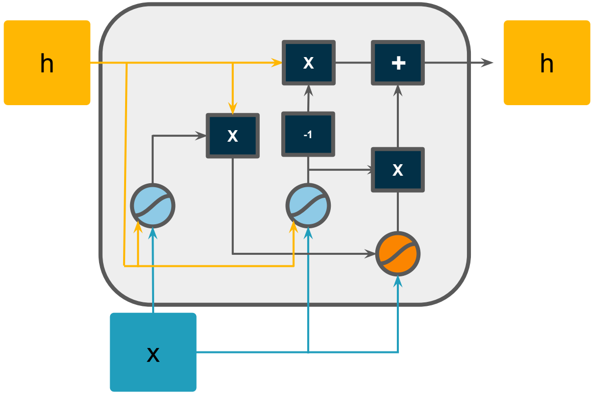
Gated Recurrent Unit Unrolled



Review







Examples of RNN Architecture

Image from: https://www.analyticsvidhya.com/blog/2018/04/solving-an-image-captioning-task-using-deep-learning/

Image from: https://www.forbes.com/sites/cindygordon/2021/12/23/a-market-to-harness-speech-recognition-artificial-intelligence-ai-innovations-on-the-rise

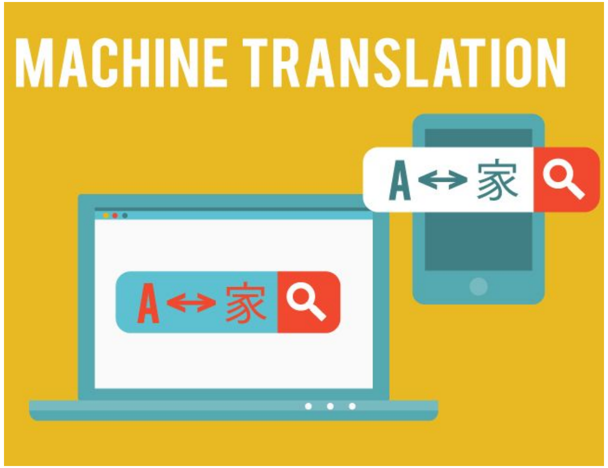


Image from: https://ehlion.com/magazine/machine-translation/

Limitations of Sequential Models

Limitations of Sequential Models

- Difficult to parallelize
- Slow to Train
- Vanishing Gradient/Long Term Memories
- Difficulty with remembering things long term