

# Recurrent Neural Networks

성균관대학교 소프트웨어학과  
이 지 형

# Contents

---

- ▶ **Sequential Data Processing**
- ▶ **Recurrent Neural Networks**
- ▶ **Long Short-Term Memory (LSTM)**

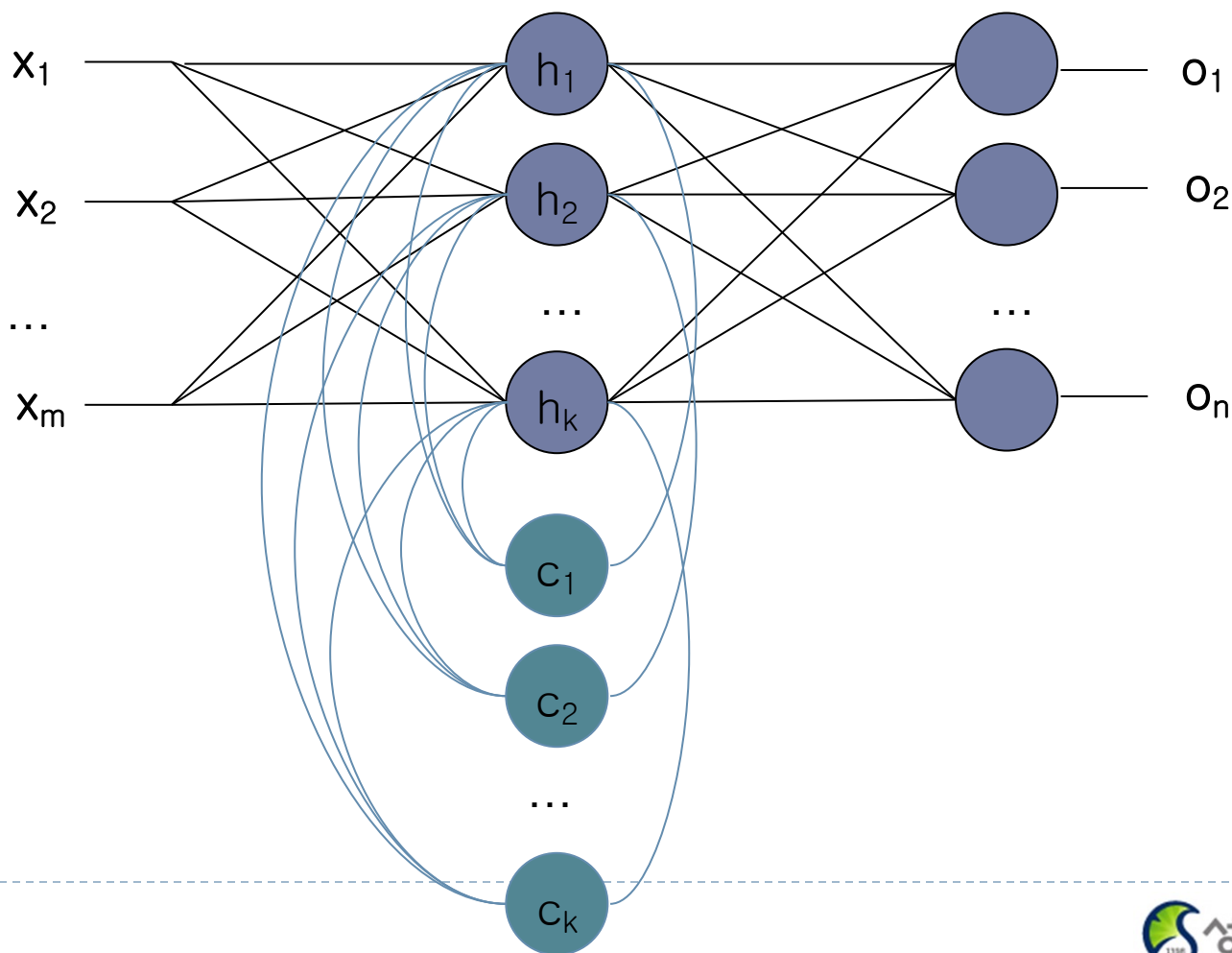
# Sequential Data Processing

---

- ▶ What is sequential data?
- ▶ What do we have to consider for sequential data processing?

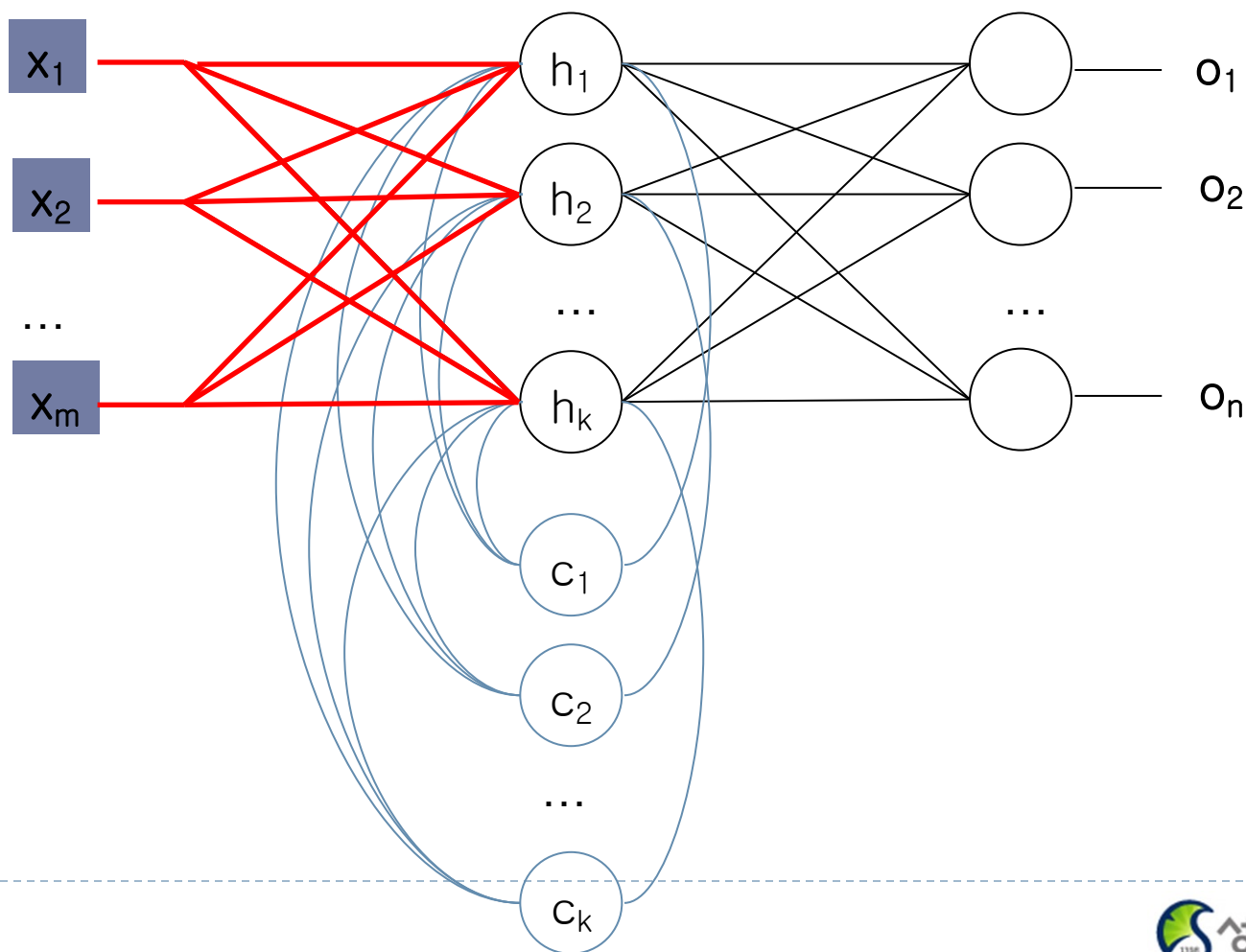
# Recurrent Neural Networks

## ► Connections form cycles



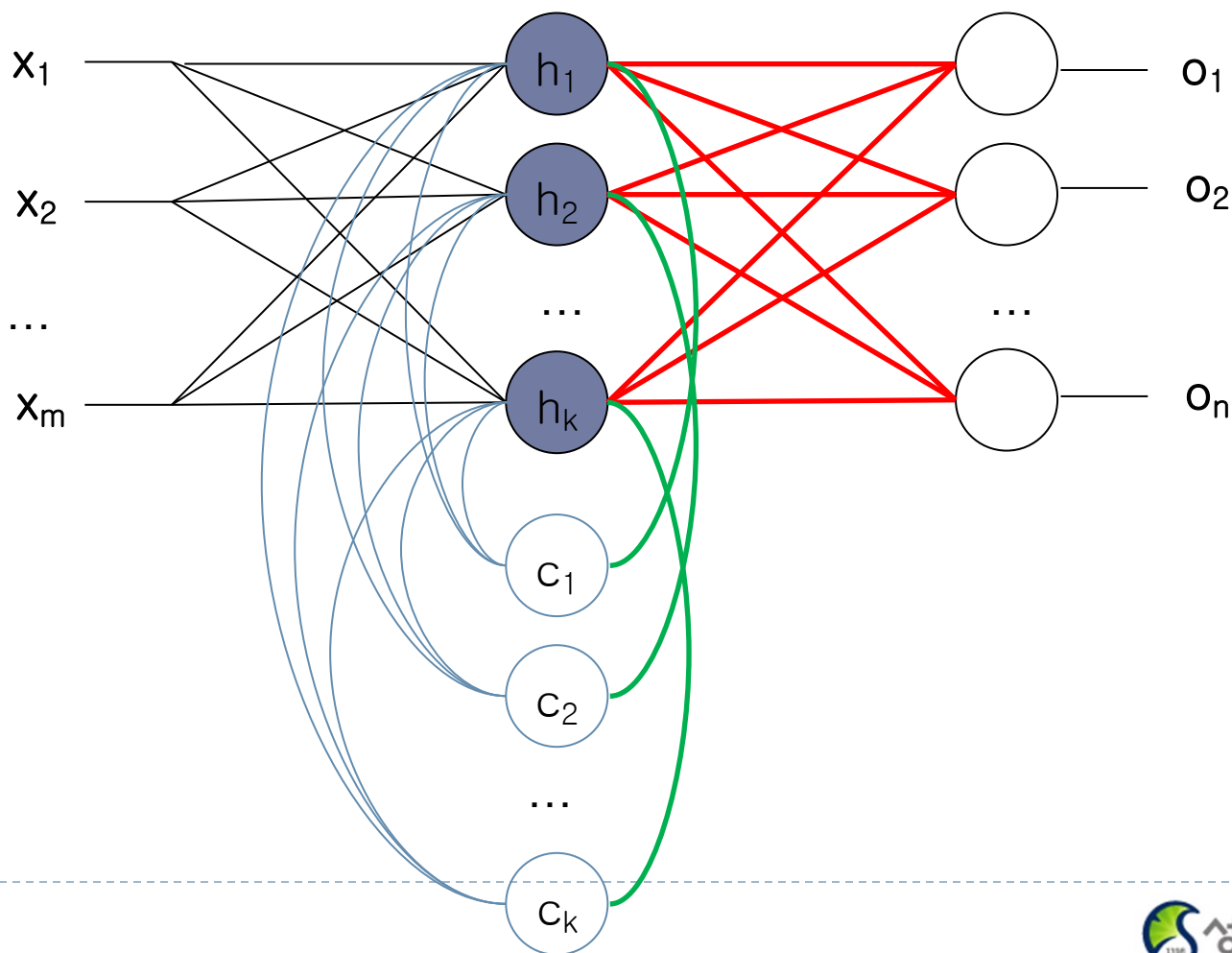
# Recurrent Neural Networks

## ► Connections form cycles



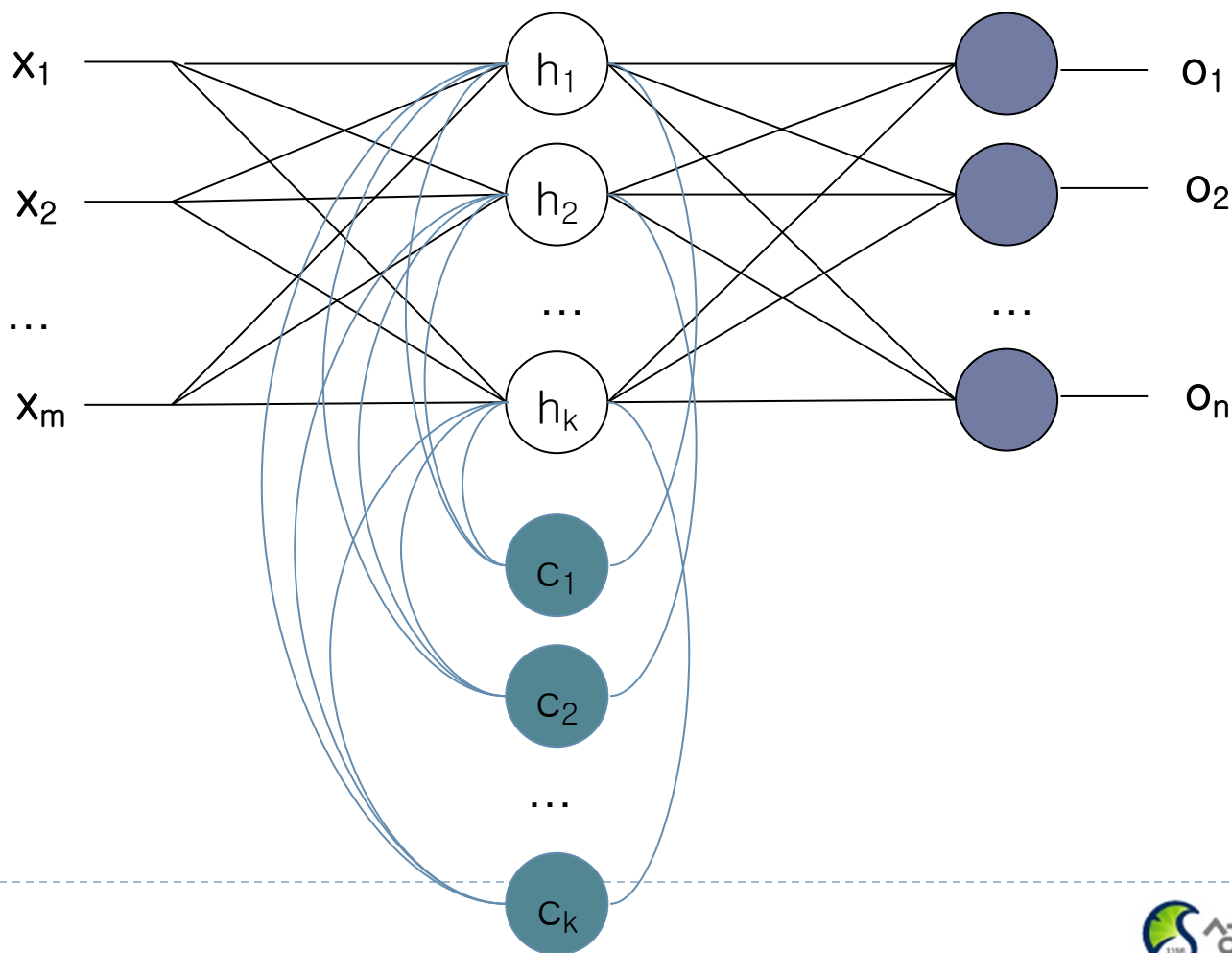
# Recurrent Neural Networks

## ► Connections form cycles



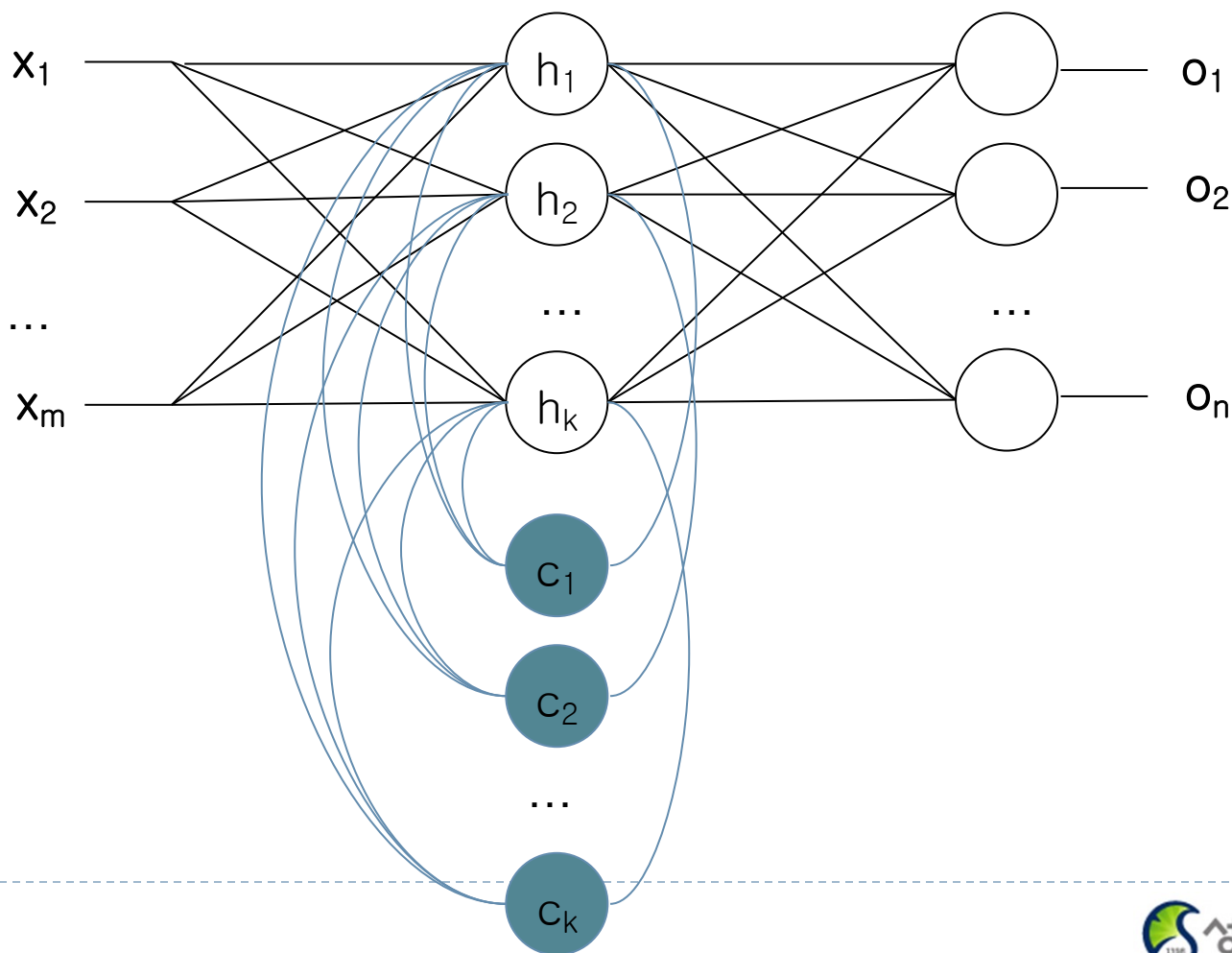
# Recurrent Neural Networks

## ► Connections form cycles



# Recurrent Neural Networks

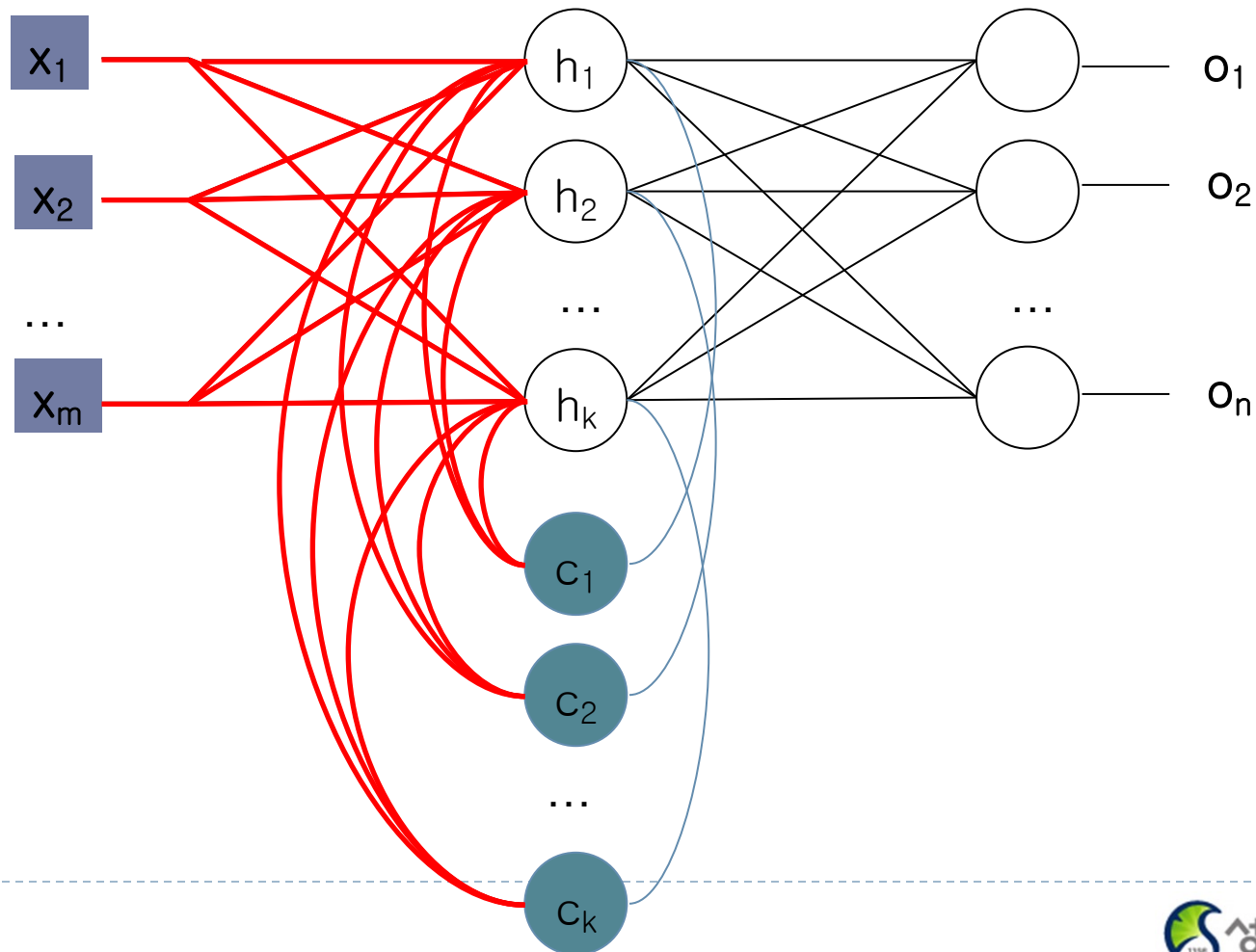
## ► Connections form cycles





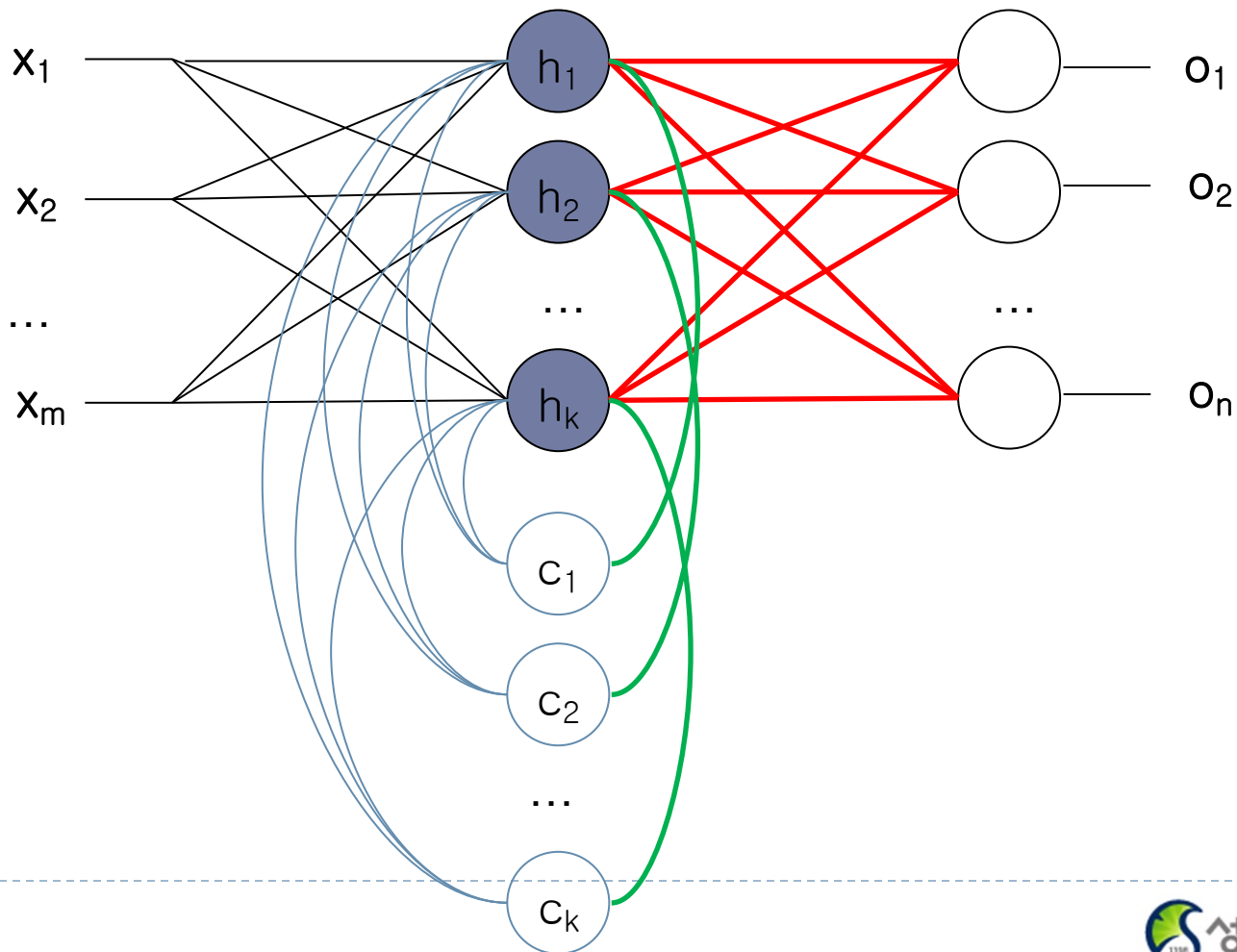
# Recurrent Neural Networks

## ► Connections form cycles



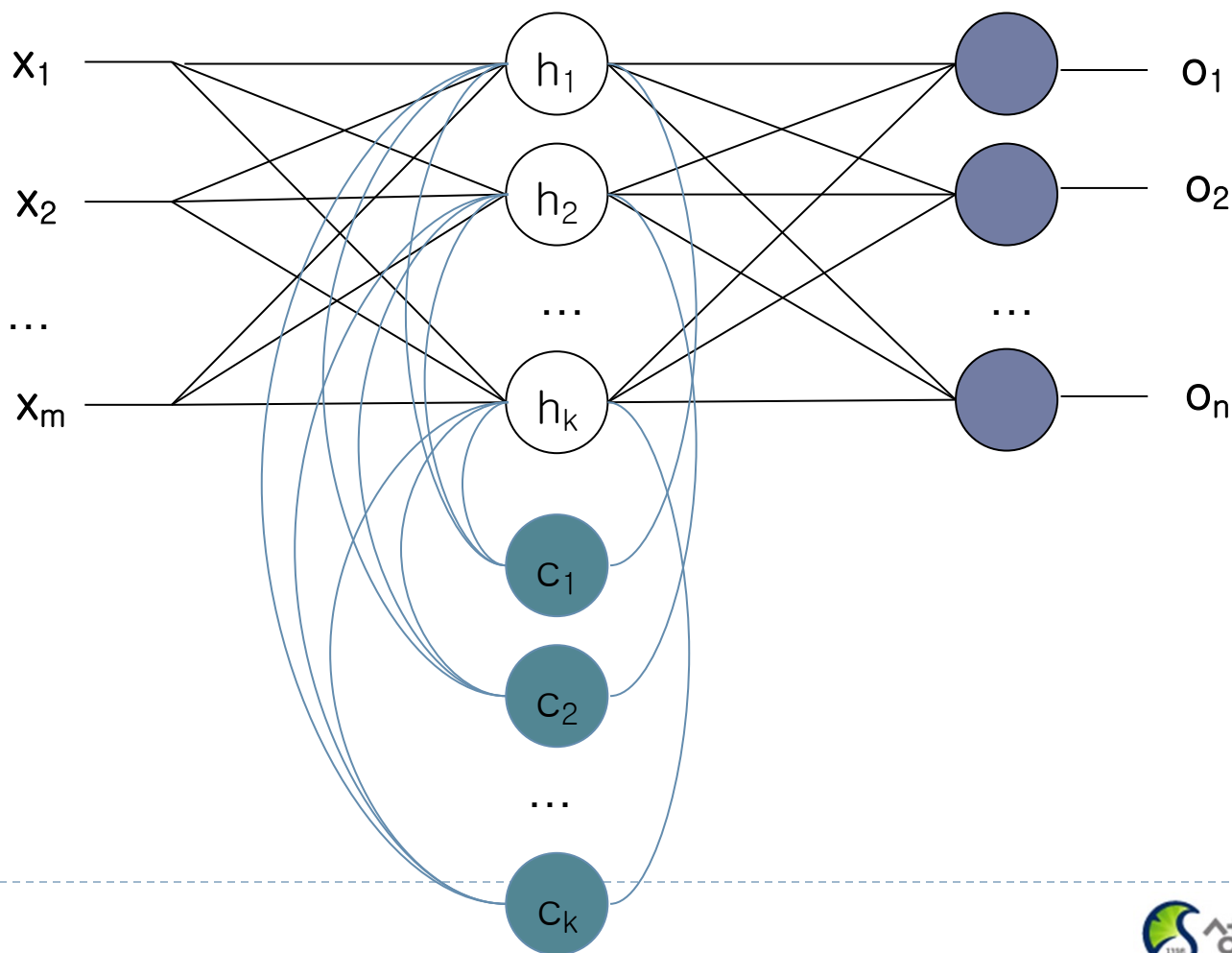
# Recurrent Neural Networks

## ► Connections form cycles



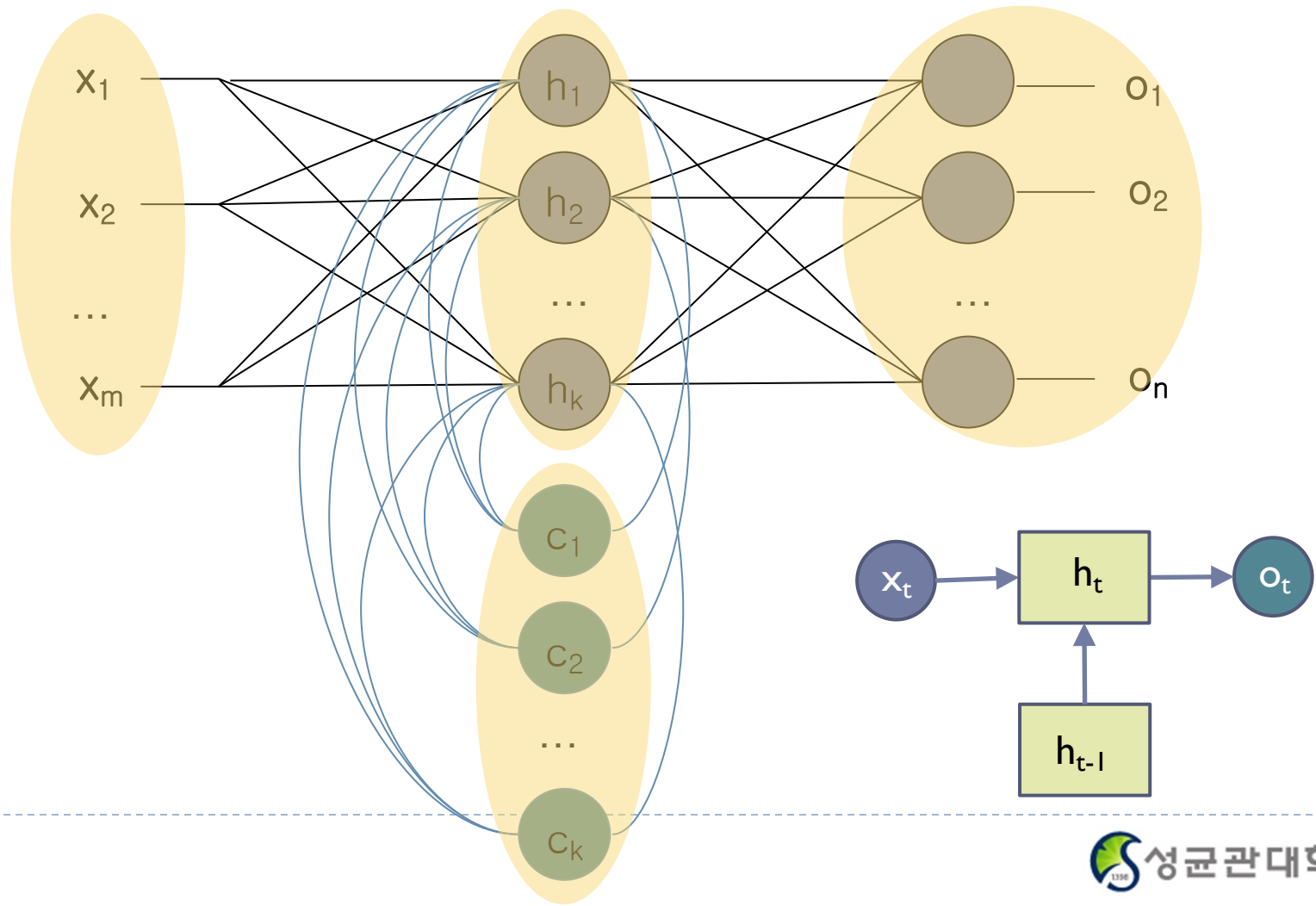
# Recurrent Neural Networks

## ► Connections form cycles



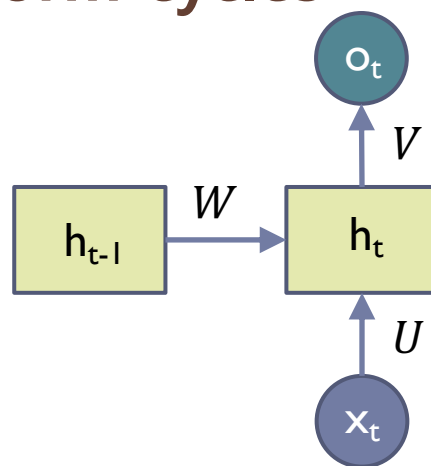
# Recurrent Neural Networks

## ► Connections form cycles



# Recurrent Neural Networks

## ► Connections form cycles



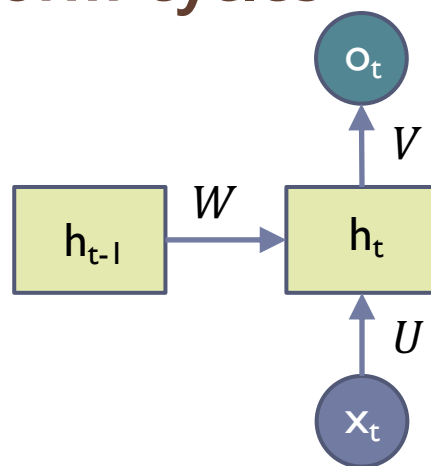
$$h_t = f(Ux_t + Wh_{t-1})$$

$$o_t = g(Vh_t)$$

- $x_t$ : input at time  $t$
- $h_t$ : hidden state at time  $t$
- $f$ : is an activation function
- $U, V, W$ : network parameters
  - RNN shares the same parameters across all time steps
- $g$ : activation function for the output layer

# Recurrent Neural Networks

## ► Connections form cycles



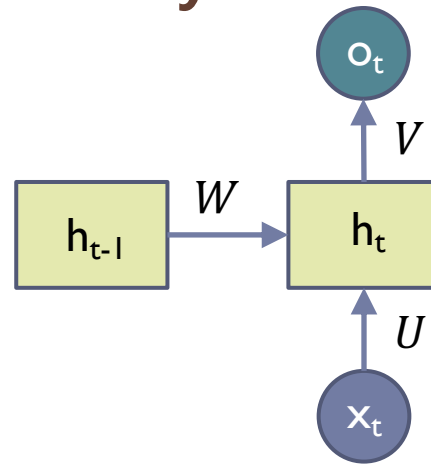
$$h_t = f(Ux_t + Wh_{t-1})$$

$$o_t = g(Vh_t)$$



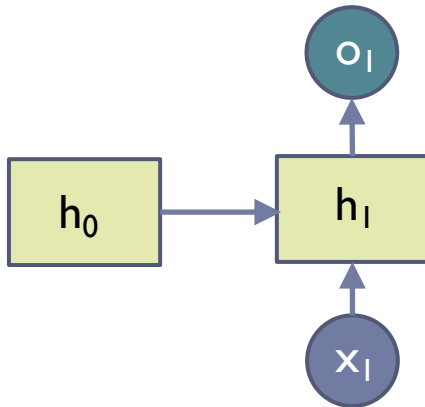
# Recurrent Neural Networks

## ► Connections form cycles



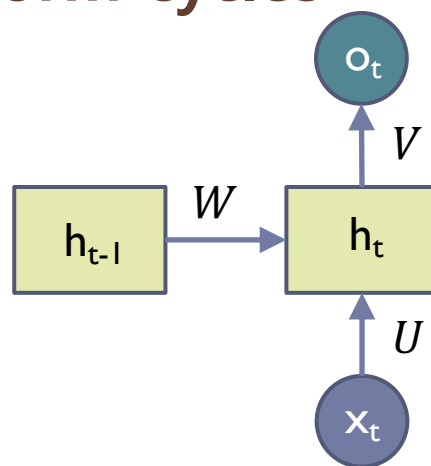
$$h_t = f(Ux_t + Wh_{t-1})$$

$$o_t = g(Vh_t)$$



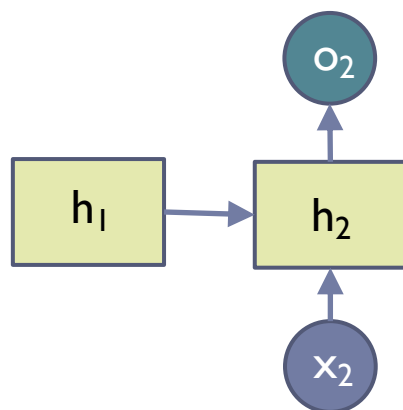
# Recurrent Neural Networks

## ► Connections form cycles



$$h_t = f(Ux_t + Wh_{t-1})$$

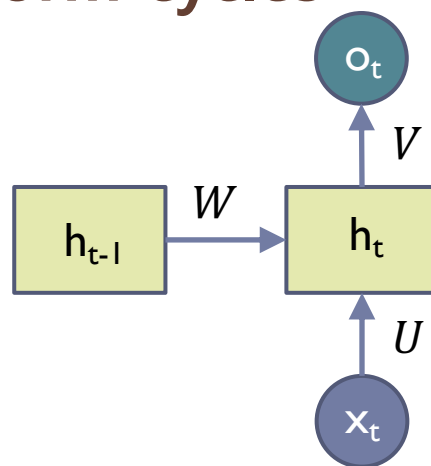
$$o_t = g(Vh_t)$$





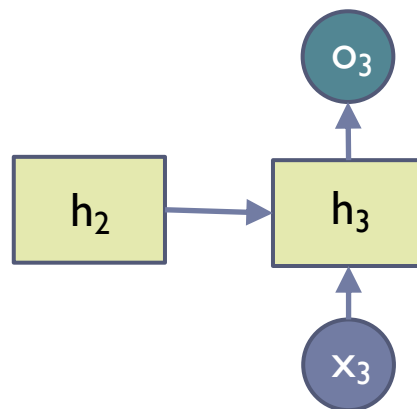
# Recurrent Neural Networks

## ► Connections form cycles



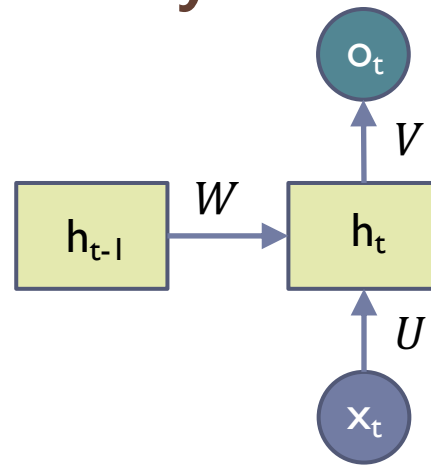
$$h_t = f(Ux_t + Wh_{t-1})$$

$$o_t = g(Vh_t)$$

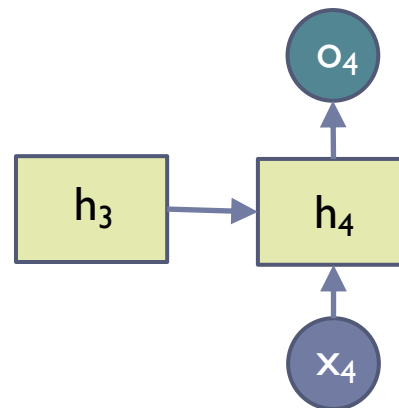


# Recurrent Neural Networks

## ► Connections form cycles

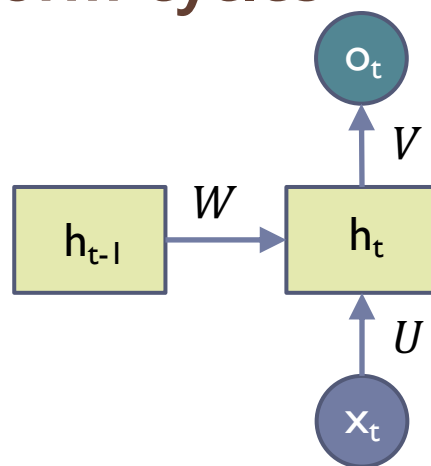


$$h_t = f(Ux_t + Wh_{t-1})$$
$$o_t = g(Vh_t)$$



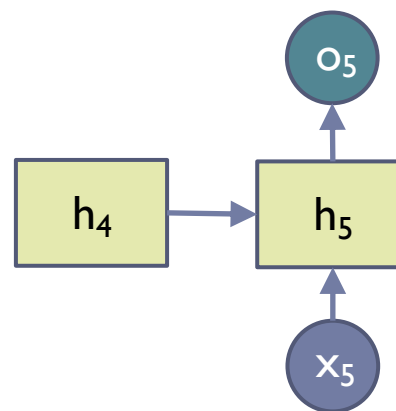
# Recurrent Neural Networks

## ► Connections form cycles



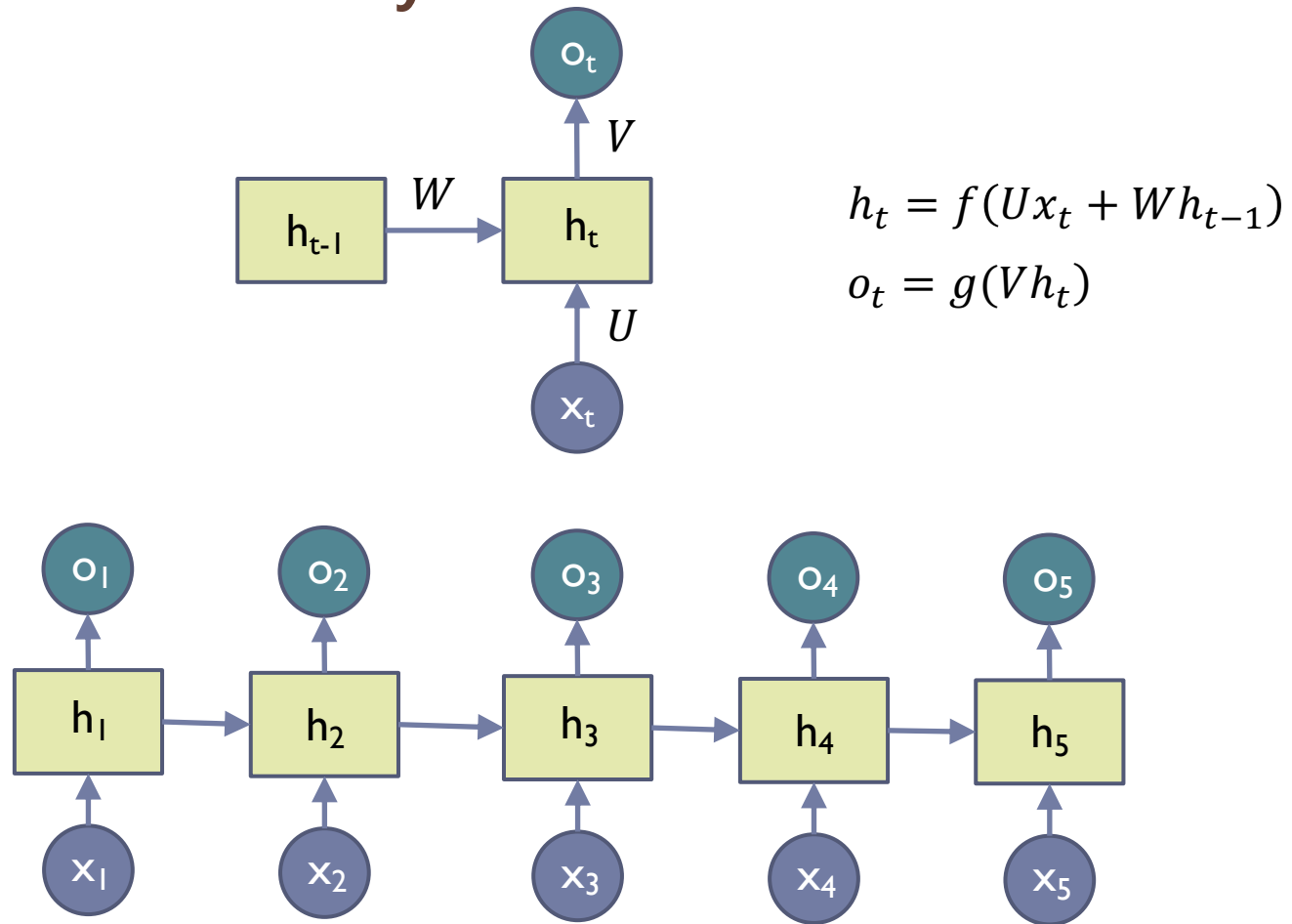
$$h_t = f(Ux_t + Wh_{t-1})$$

$$o_t = g(Vh_t)$$



# Recurrent Neural Networks

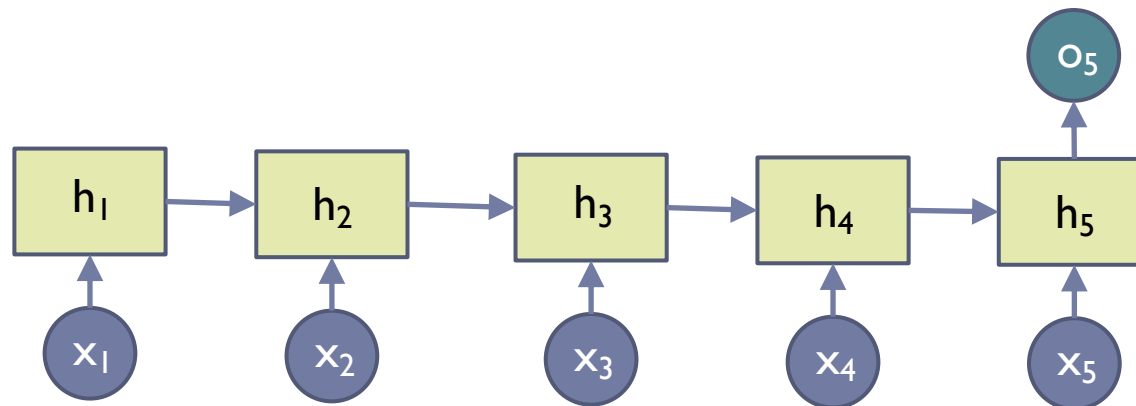
## ► Connections form cycles



# Recurrent Neural Networks

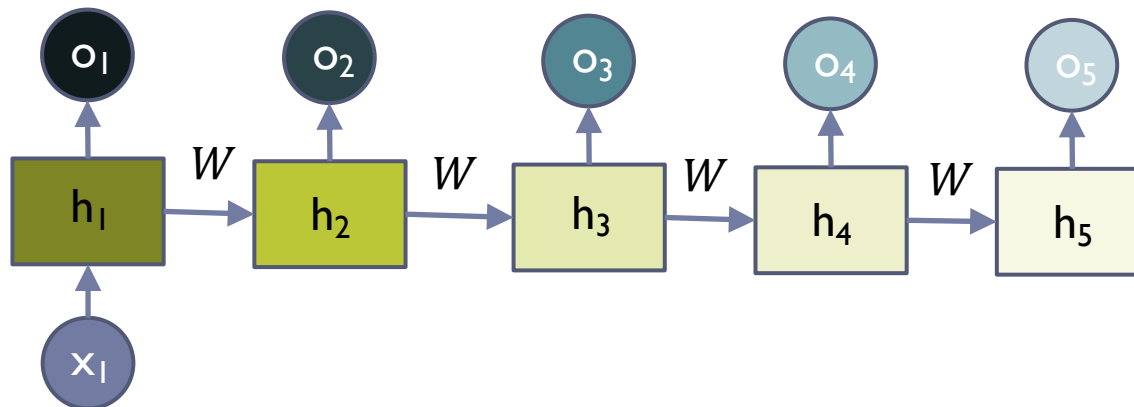
## ▶ Long Term Dependency

- ▶  $x_1 \sim x_{t-1}$  are encoded into  $h_{t-1}$
- ▶  $h_{t-1}$  has the information on the past
- ▶ It is a context to process  $x_t$



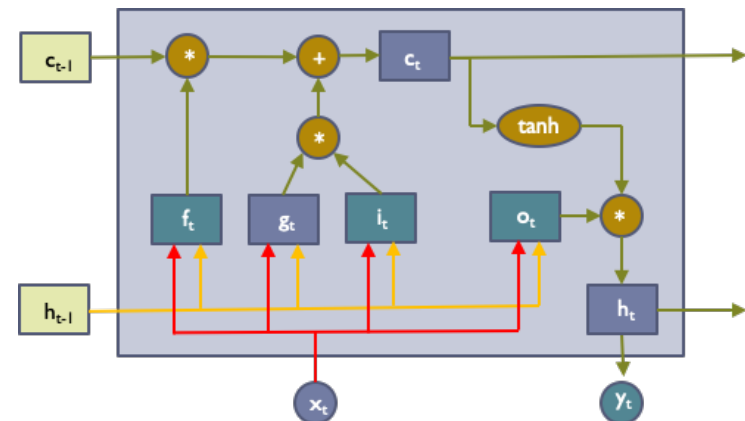
# Recurrent Neural Networks

- ▶ **Long Term Dependency of Standard RNN**
  - ▶ However, it may exponentially decay or grow
  - ▶ Usually it is limited to 10 steps



# Long Short-Term Memory (LSTM)

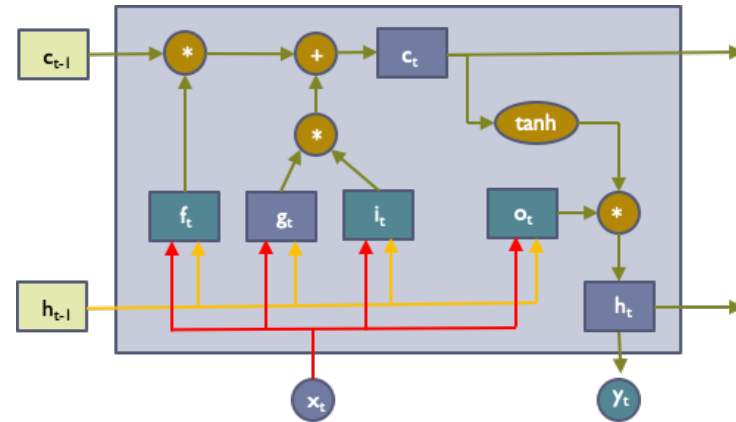
- ▶ **Capable of learning long-term dependencies.**
  - ▶ LSTM networks introduce a new structure called a memory cell
    - ▶ An LSTM can learn to bridge time intervals in excess of 1000 steps
  - ▶ **Gate units that learn to open and close access to the past**
    - ▶ Input gate
    - ▶ Forget gate
    - ▶ Output gate
    - ▶ Neuron with a self-recurrent



# Long Short-Term Memory (LSTM)

## ► Equations

- $i$ : input gate
- $f$ : forget gate
- $o$ : output gate
- $g$ : self-recurrent
- $c_t$ : internal memory
- $h_t$ : hidden state
- $y$ : final output

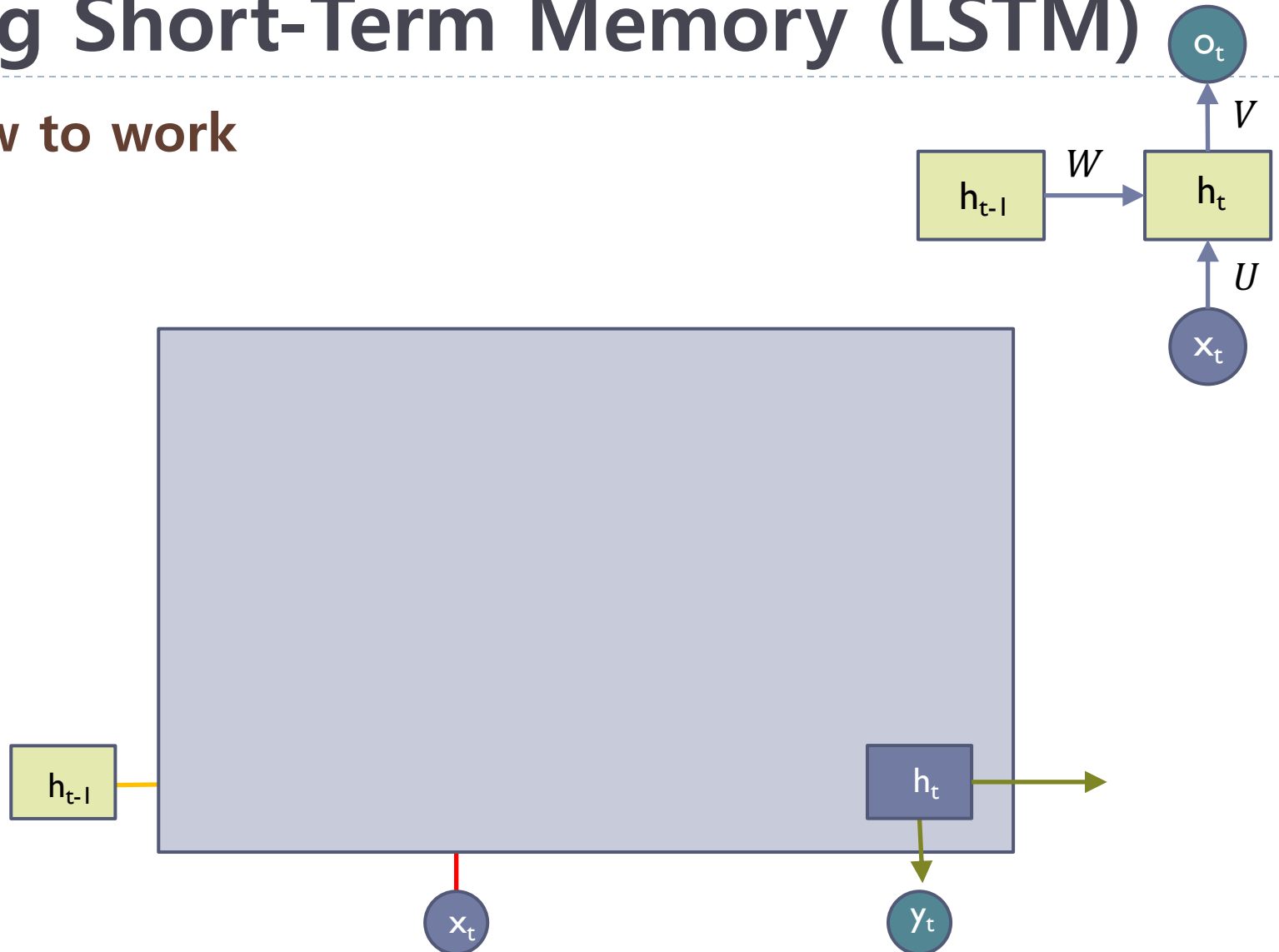


$$\begin{aligned}i &= \sigma(x_t U^i + h_{t-1} W^i) \\f &= \sigma(x_t U^f + h_{t-1} W^f) \\o &= \sigma(x_t U^o + h_{t-1} W^o) \\g &= \tanh(x_t U^g + h_{t-1} W^g) \\c_t &= c_{t-1} \circ f + g \circ i \\h_t &= \tanh(c_t) \circ o \\y &= \text{softmax}(V h_t)\end{aligned}$$



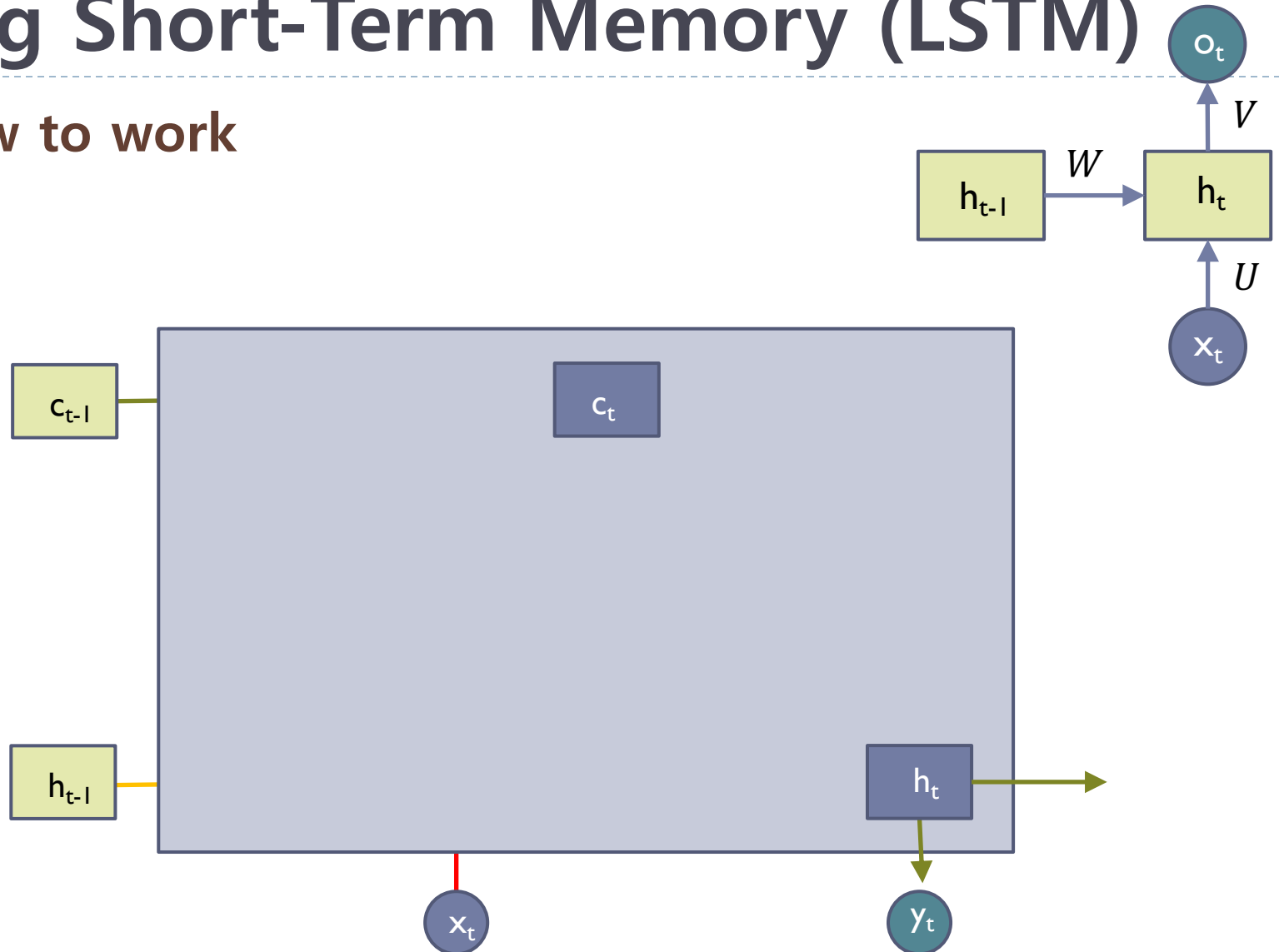
# Long Short-Term Memory (LSTM)

## ▶ How to work



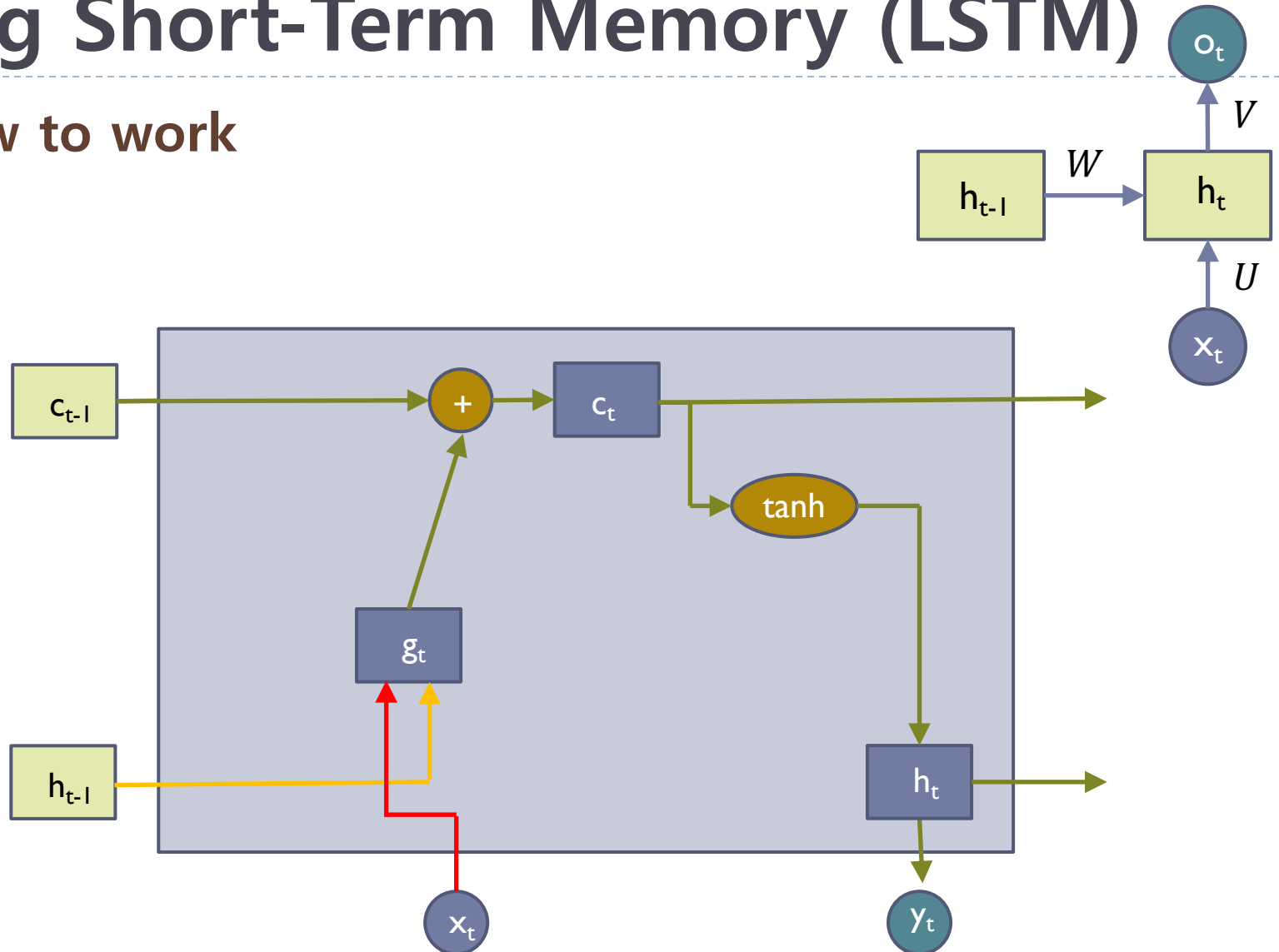
# Long Short-Term Memory (LSTM)

## ▶ How to work



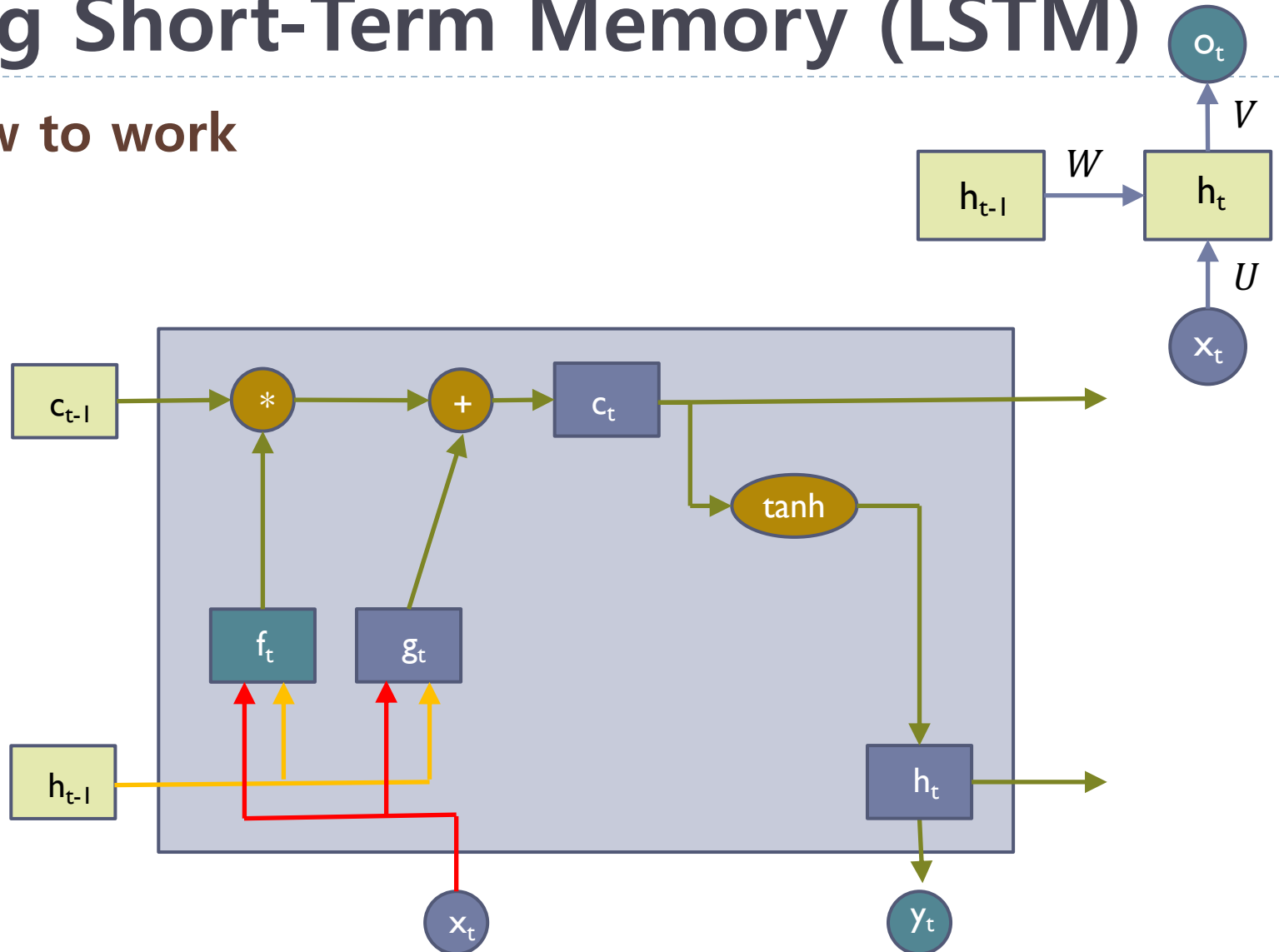
# Long Short-Term Memory (LSTM)

## ► How to work



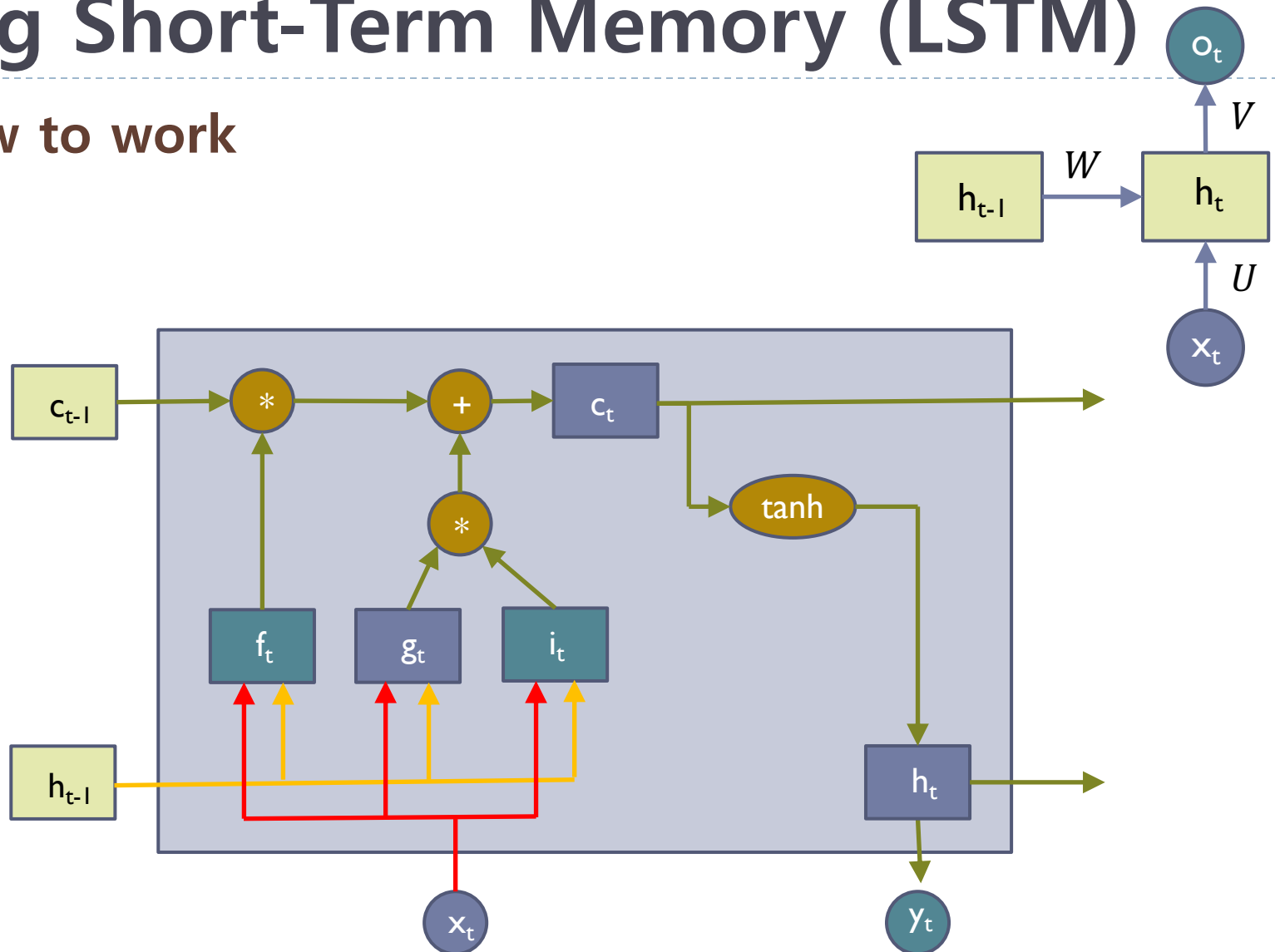
# Long Short-Term Memory (LSTM)

## ► How to work



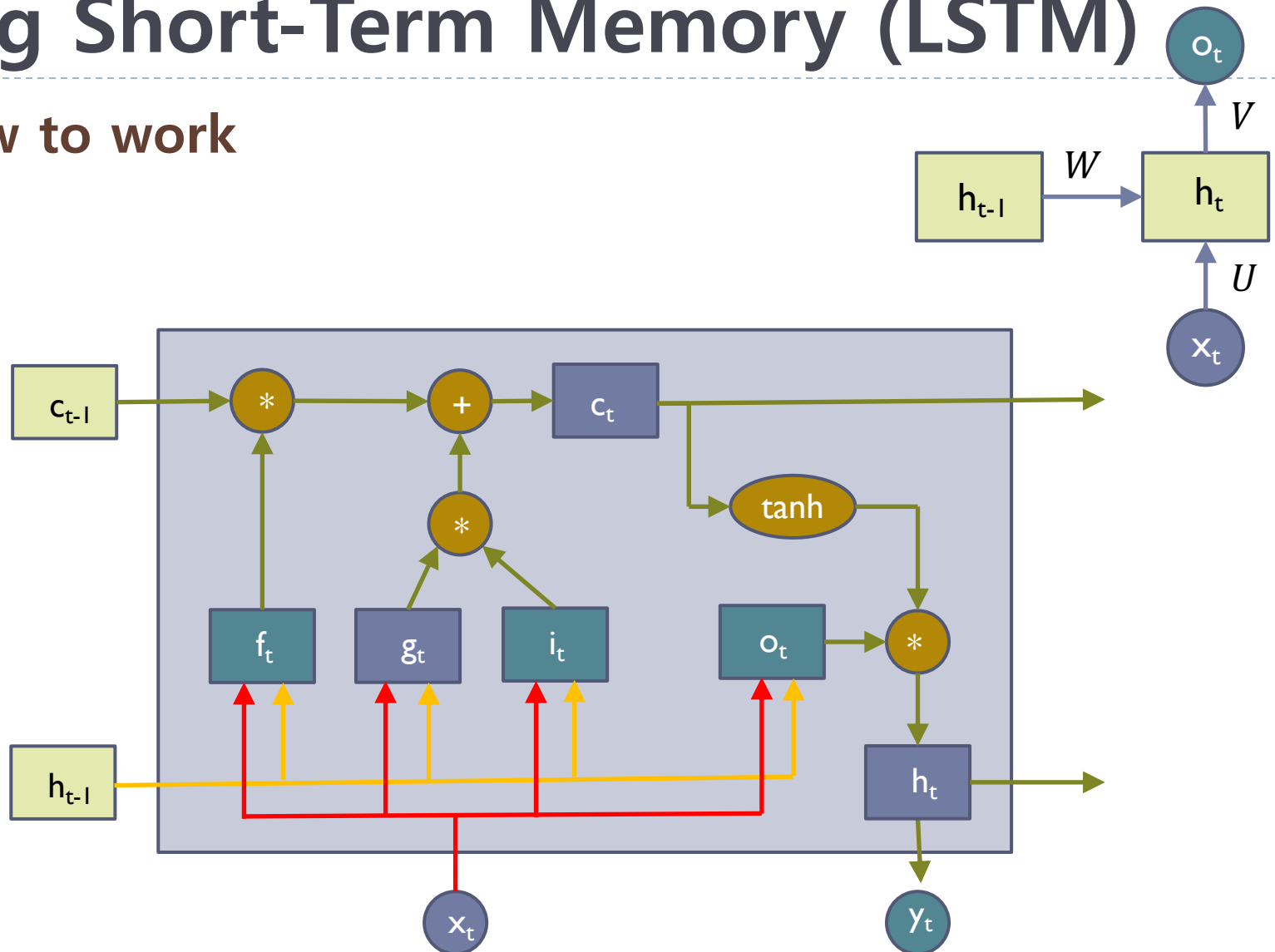
# Long Short-Term Memory (LSTM)

## ► How to work



# Long Short-Term Memory (LSTM)

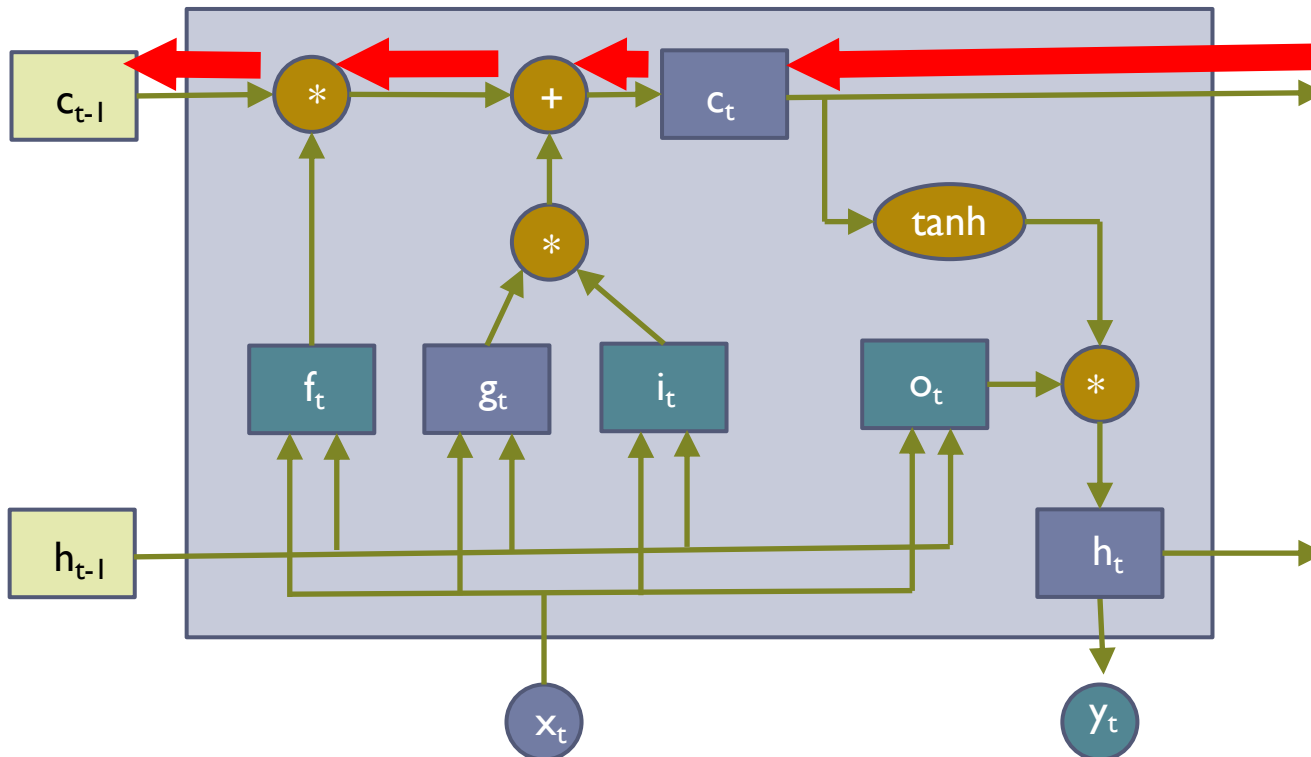
## ► How to work



# Long Short-Term Memory (LSTM)

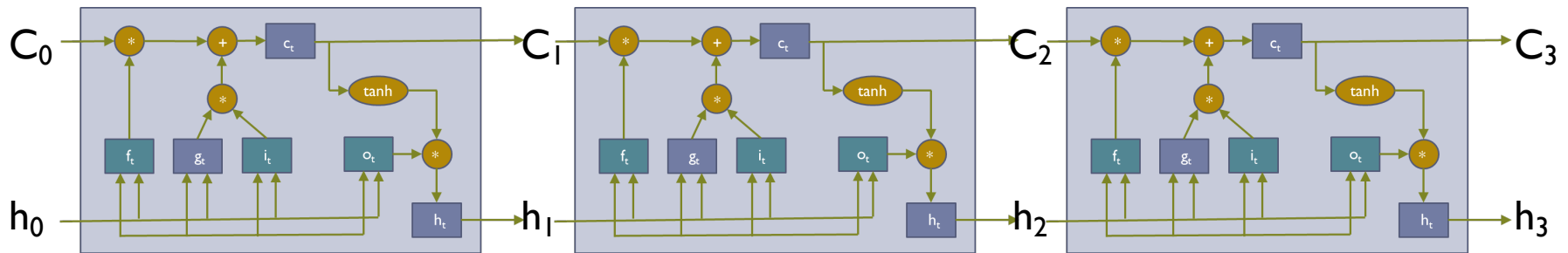
## ► Gradient Flow

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



# Long Short-Term Memory (LSTM)

## ► Gradient Flow





## ► Gradient Flow

The diagram illustrates the internal structure of a recurrent neural network (RNN) unrolled over three time steps, labeled  $t=0$ ,  $t=1$ , and  $t=2$ . A red arrow at the top indicates the direction of the hidden state sequence from left to right.

At each time step  $t$ , the hidden state  $h_{t-1}$  is fed into the network along with the current input vector  $x_t$ . The input vector  $x_t$  is split into three components:  $f_t$ ,  $g_t$ , and  $i_t$ . These components are used to calculate the forget gate output  $f_t$ , the gate output  $g_t$ , and the input gate output  $i_t$ . The forget gate output  $f_t$  is multiplied by the previous hidden state  $h_{t-1}$  (indicated by the  $*$  node). The gate output  $g_t$  and the input gate output  $i_t$  are multiplied together (indicated by the  $*$  node) and then added to the result of the forget gate operation (indicated by the  $+$  node). The result is then passed through a  $\tanh$  activation function (indicated by the  $\tanh$  node) to produce the candidate hidden state  $h_t$ . The candidate hidden state  $h_t$  is then multiplied by the forget gate output  $f_t$  (indicated by the  $*$  node) to produce the final hidden state  $h_t$ .

The hidden state  $h_t$  is also used to calculate the output  $o_t$  (indicated by the  $o_t$  node). The output  $o_t$  is then passed through a  $\tanh$  activation function (indicated by the  $\tanh$  node) to produce the final output  $y_t$ .

The diagram shows the flow of information from the inputs  $x_t$  to the hidden states  $h_t$  and the outputs  $y_t$  across the three time steps. The hidden state sequence is represented by a red arrow at the top, and the input sequence is represented by a red arrow at the bottom.

The diagram illustrates the VGG-16 architecture, showing the sequence of layers from input to output. The layers are color-coded and labeled as follows:

- Input:** A gray box labeled "Input".
- Pool:** A blue box labeled "Pool".
- 1x7 conv 64/12:** An orange box labeled "1x7 conv 64/12".
- 3x3 conv 64:** A series of three maroon boxes labeled "3x3 conv 64".
- 3x3 conv 64:** A series of three maroon boxes labeled "3x3 conv 64".
- 3x3 conv 128:** A series of three purple boxes labeled "3x3 conv 128".
- 3x3 conv 128:** A series of three purple boxes labeled "3x3 conv 128".
- 3x3 conv 256:** A series of three light blue boxes labeled "3x3 conv 256".
- 3x3 conv 256:** A series of three light blue boxes labeled "3x3 conv 256".
- 3x3 conv 512:** A series of three green boxes labeled "3x3 conv 512".
- 3x3 conv 512:** A series of three green boxes labeled "3x3 conv 512".
- FC1000:** A red box labeled "FC1000".
- Pool:** A blue box labeled "Pool".

Arrows indicate the flow of data between layers, with skip connections shown as curved arrows. A red arrow at the top points from right to left, indicating the direction of the data flow.

# Gated Recurrent Units

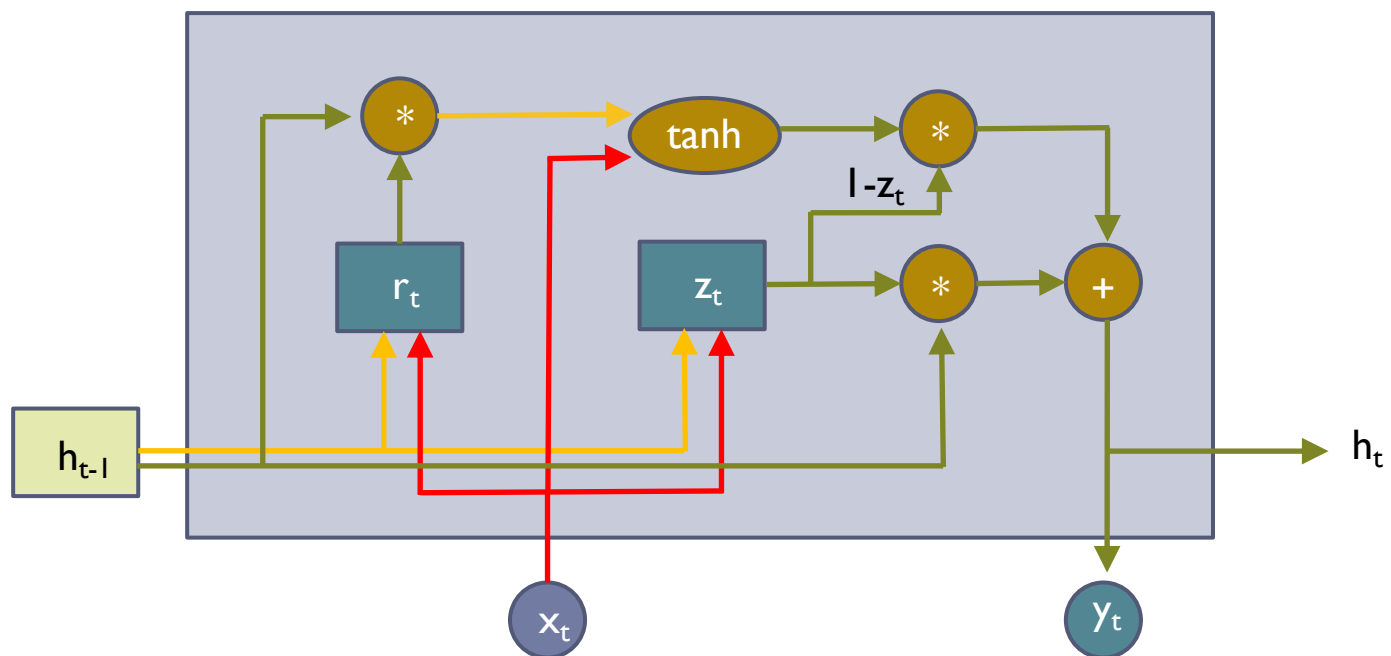
## ► Structure

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r)$$

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z)$$

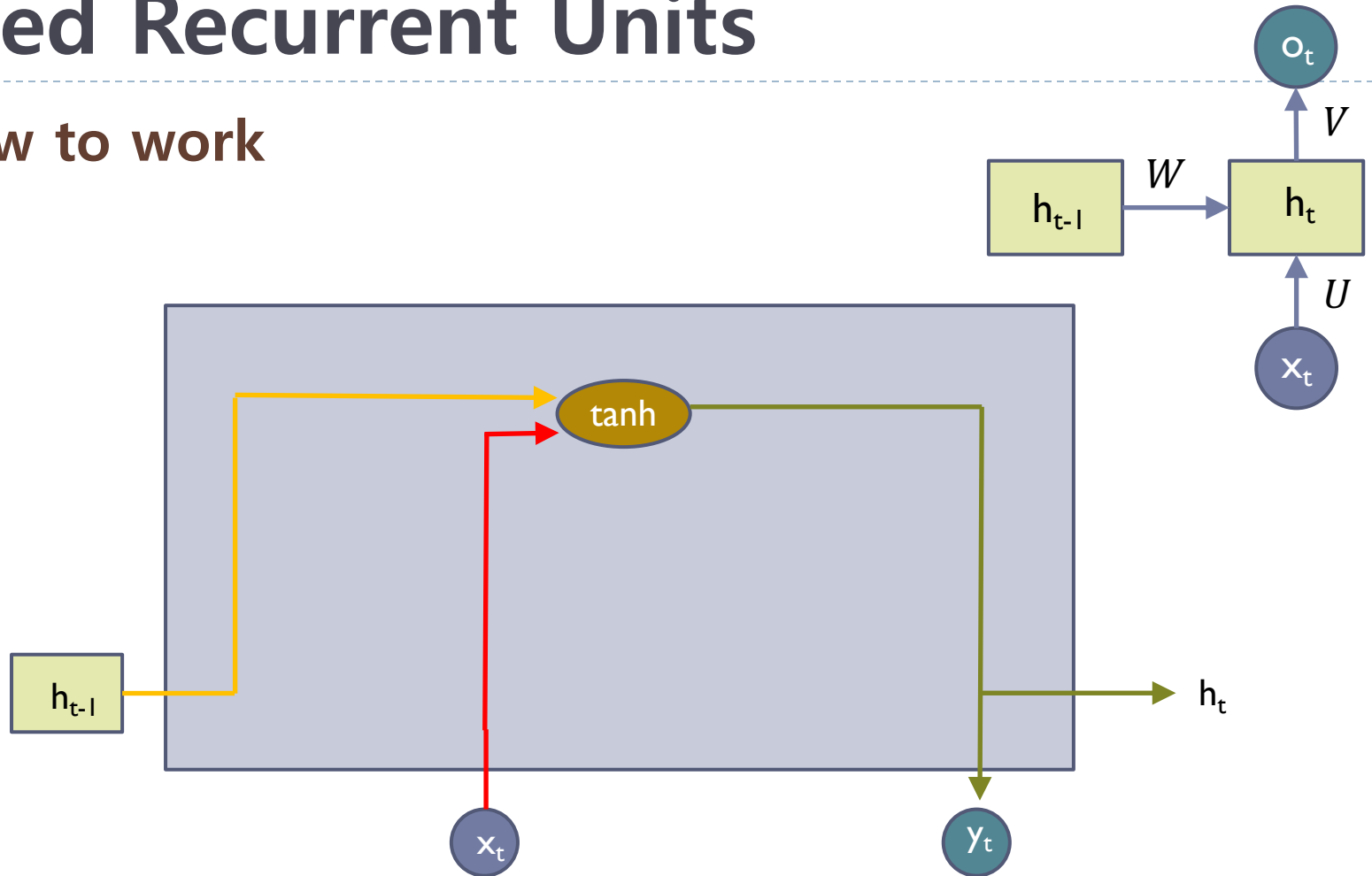
$$\tilde{h}_t = \tanh(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h)$$

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



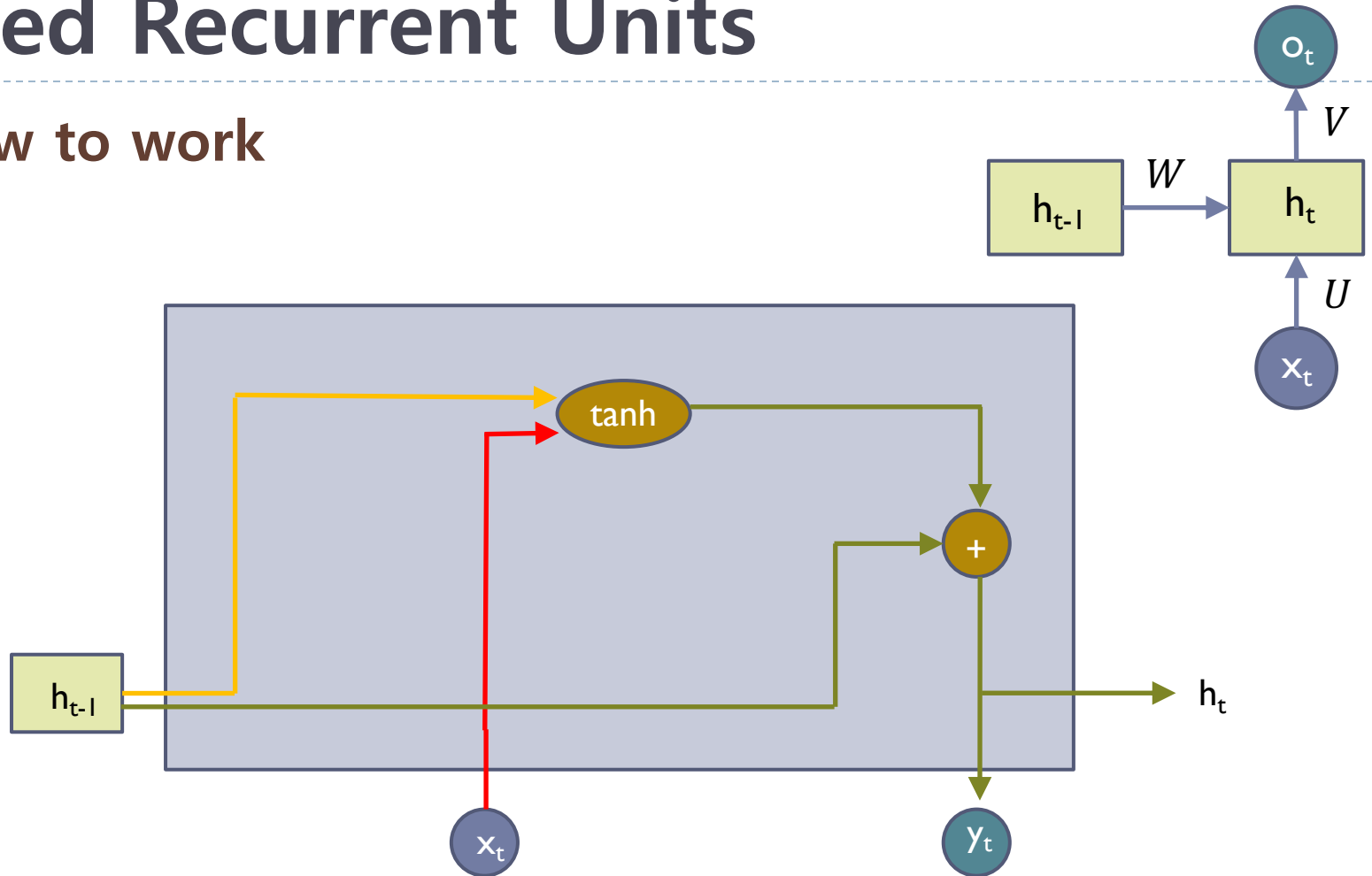
# Gated Recurrent Units

## ▶ How to work



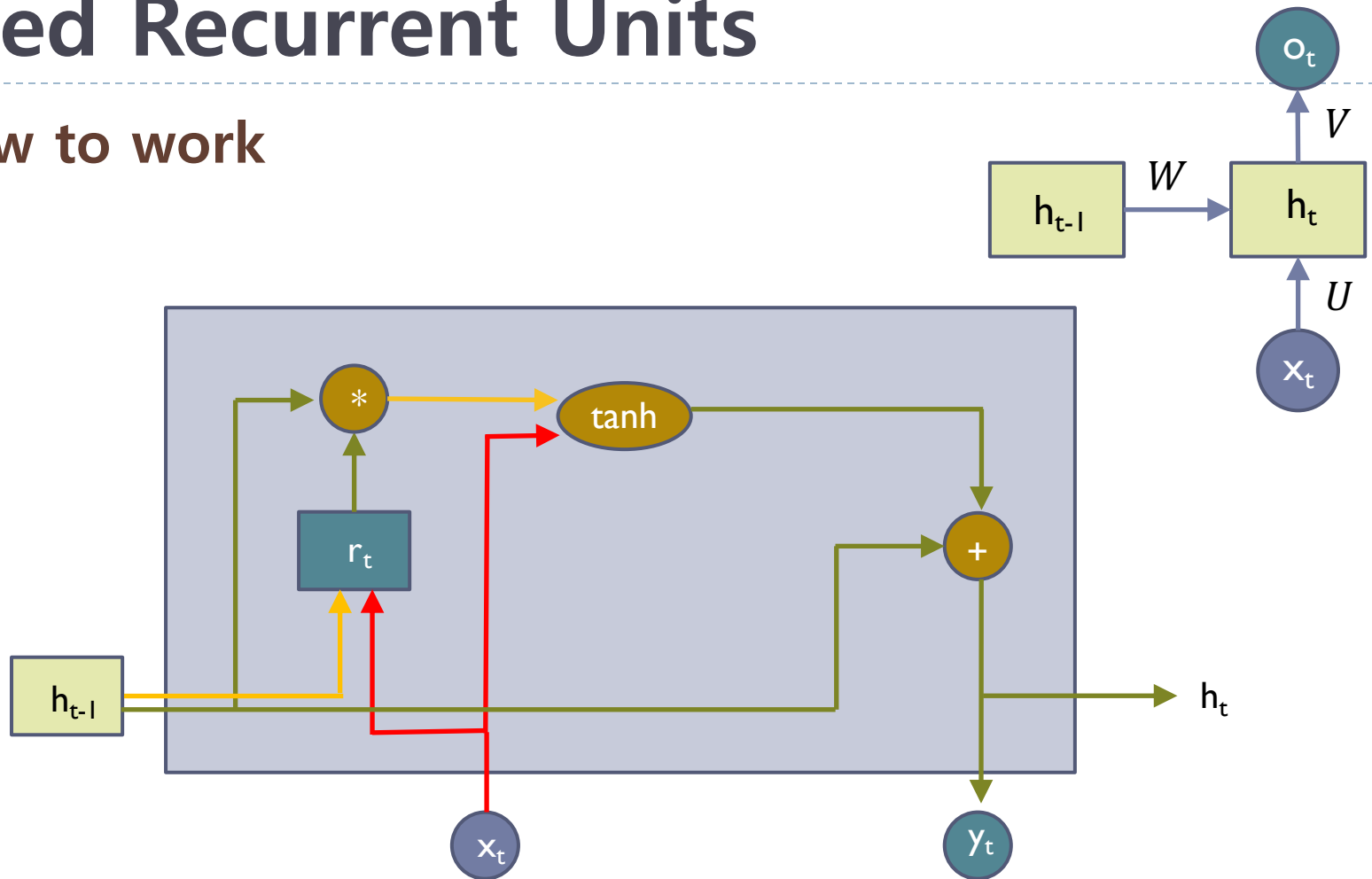
# Gated Recurrent Units

## ▶ How to work



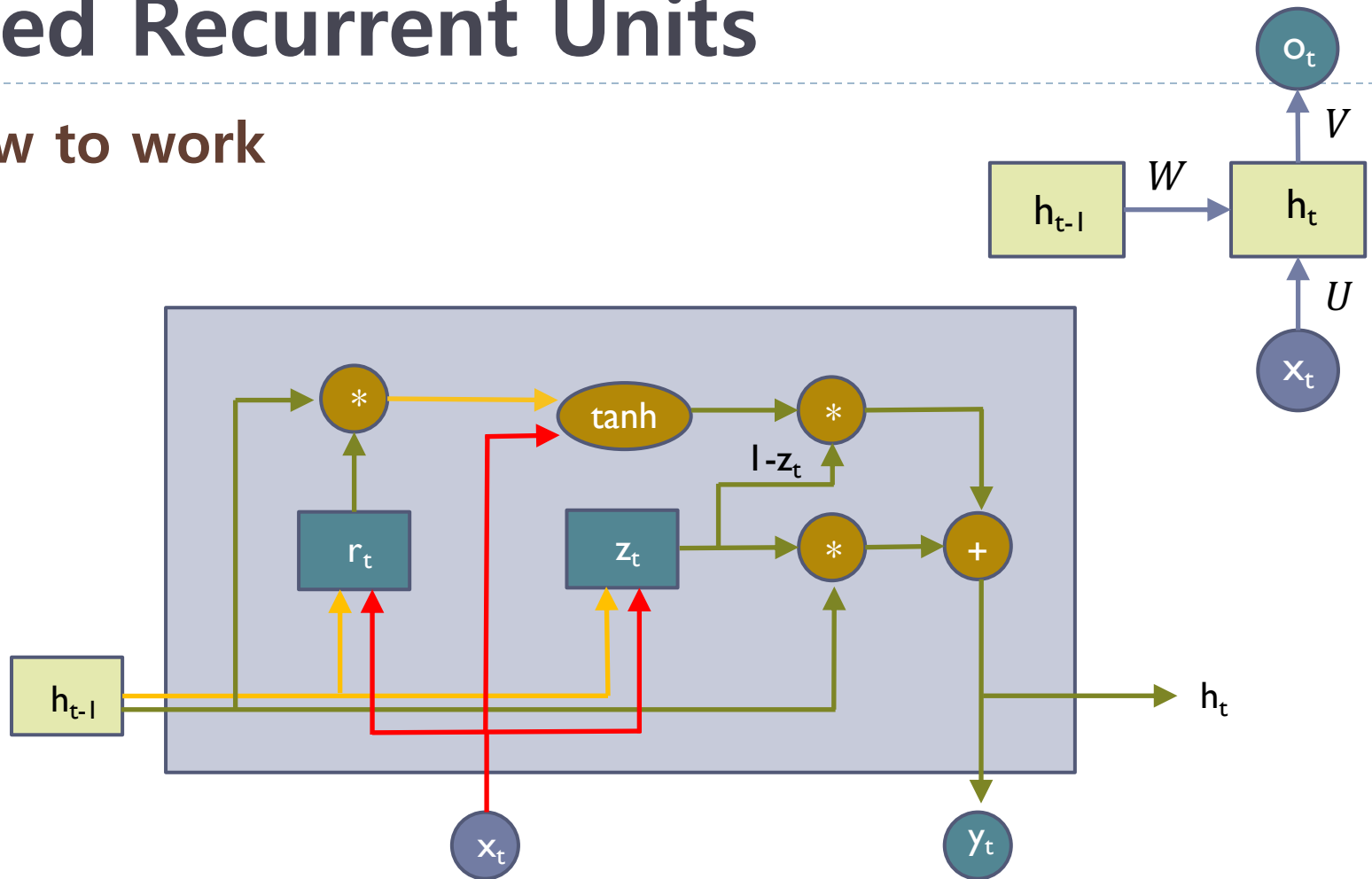
# Gated Recurrent Units

## ► How to work



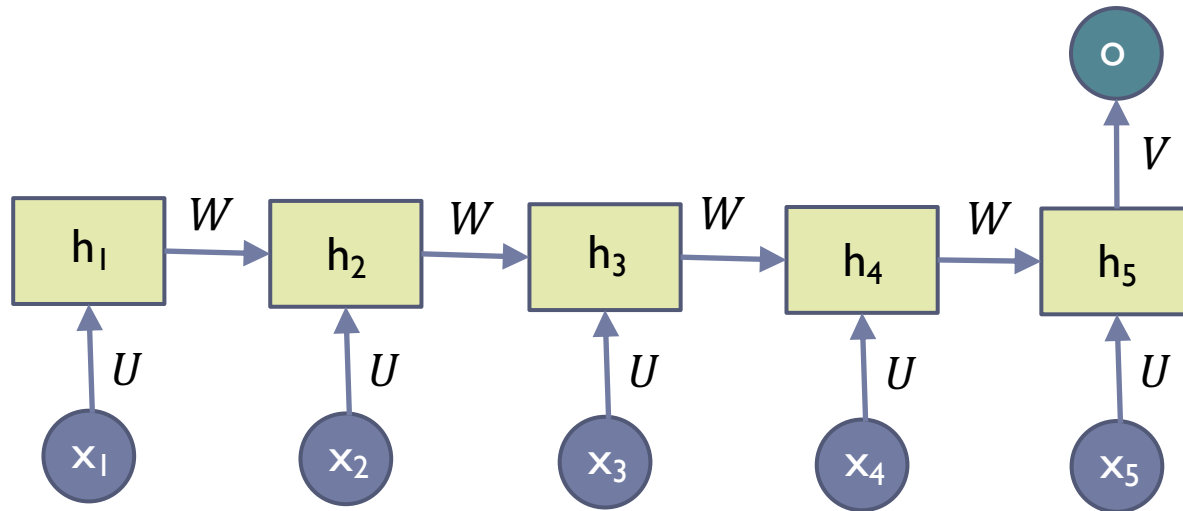
# Gated Recurrent Units

## ► How to work



# Sequence Prediction

$$x_1 x_2 x_3 \cdots x_n \rightarrow y$$



# Question and Answer