

Recurrent Neural Networks

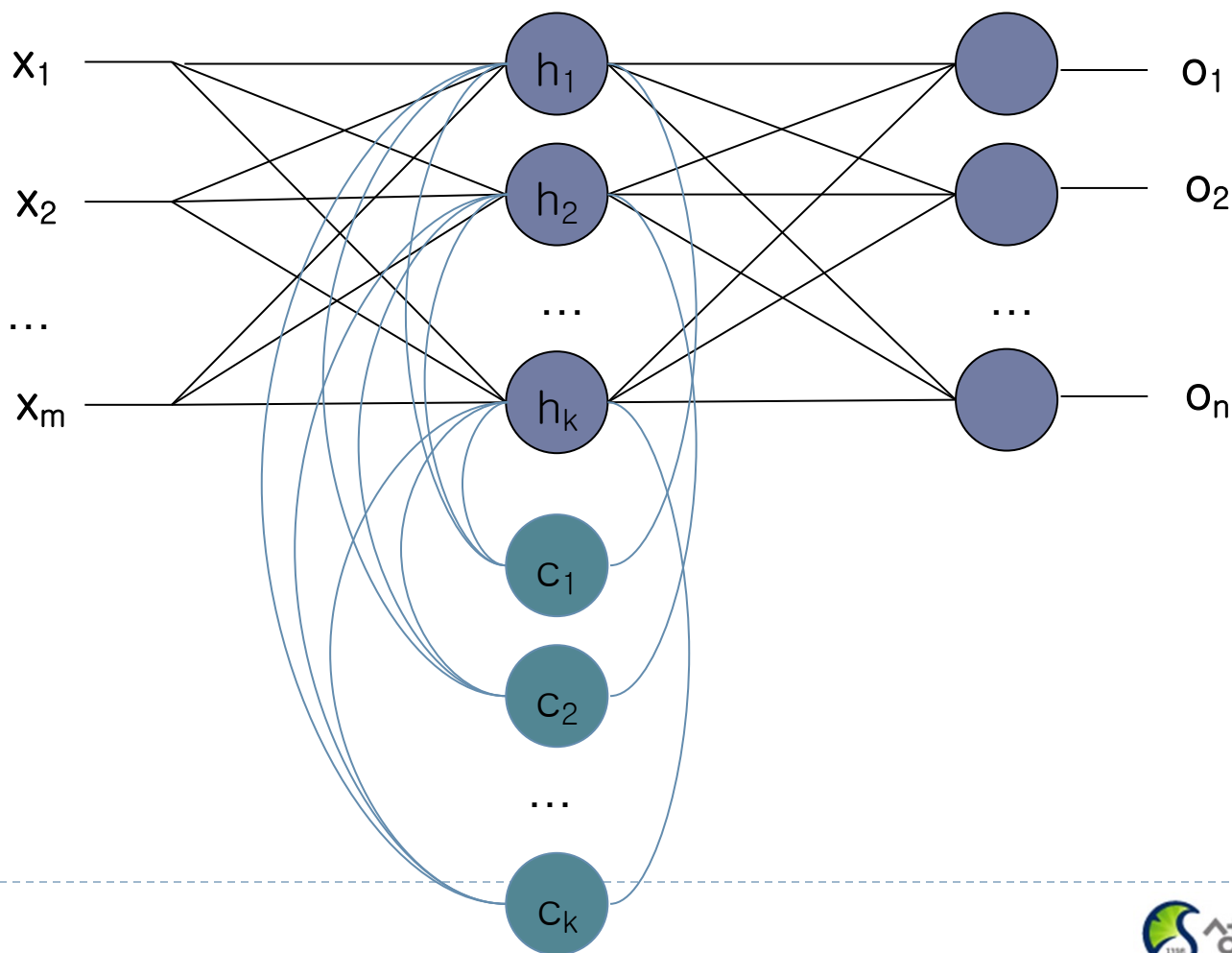
성균관대학교 소프트웨어학과
이 지 형

Contents

- ▶ **Recurrent Neural Networks**
- ▶ **Long Short-Term Memory (LSTM)**
- ▶ **Attention Model**

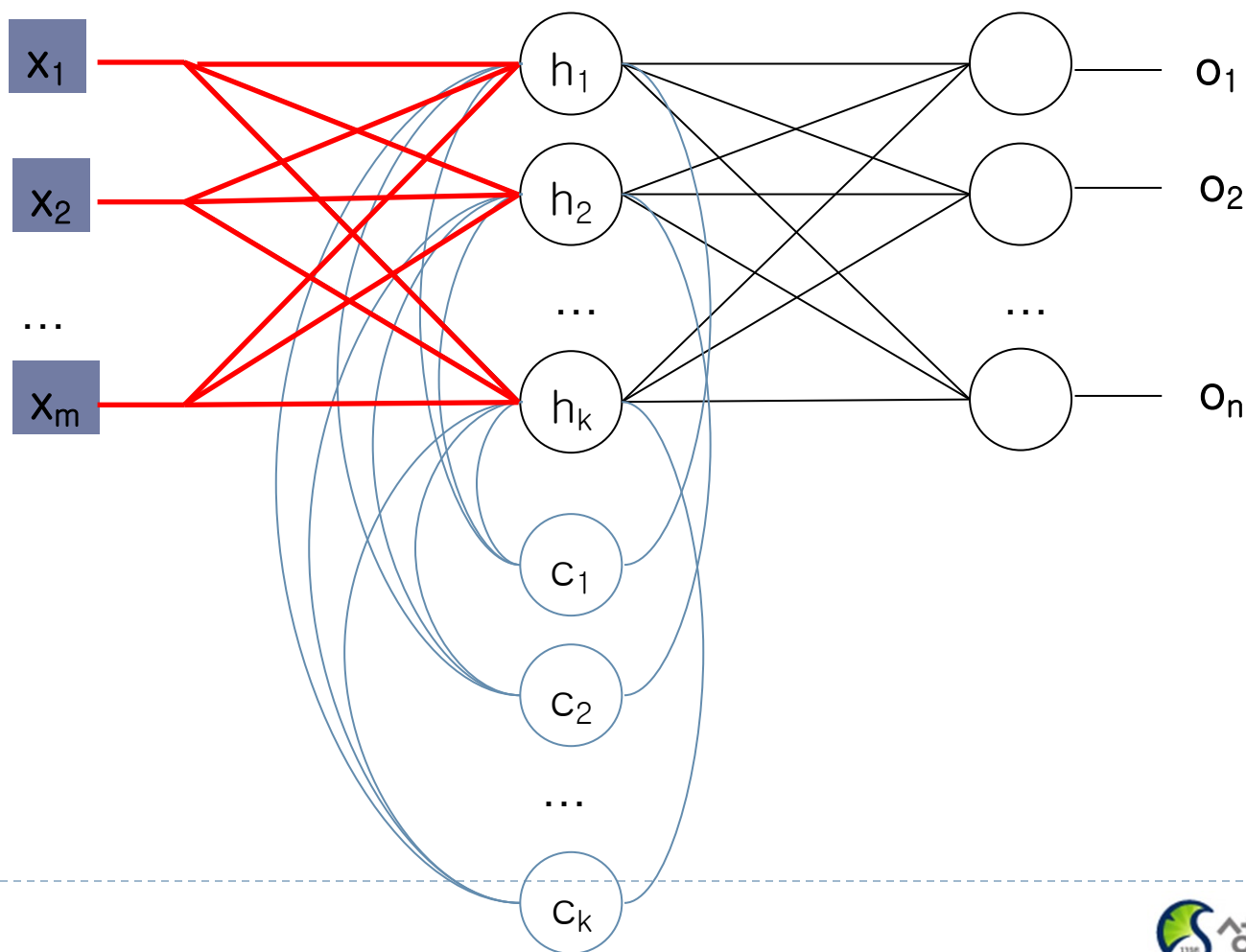
Recurrent Neural Networks

► Connections form cycles



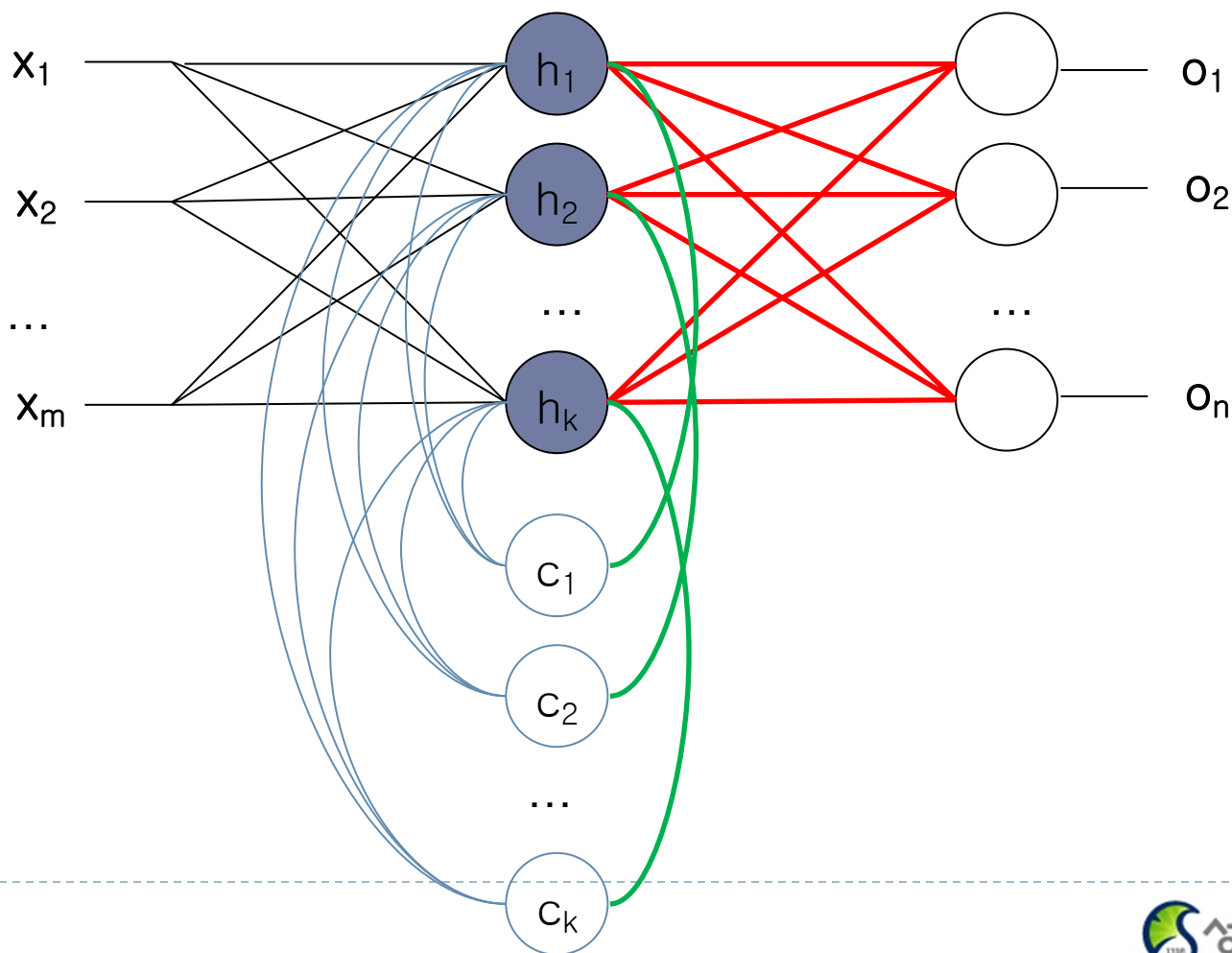
Recurrent Neural Networks

► Connections form cycles



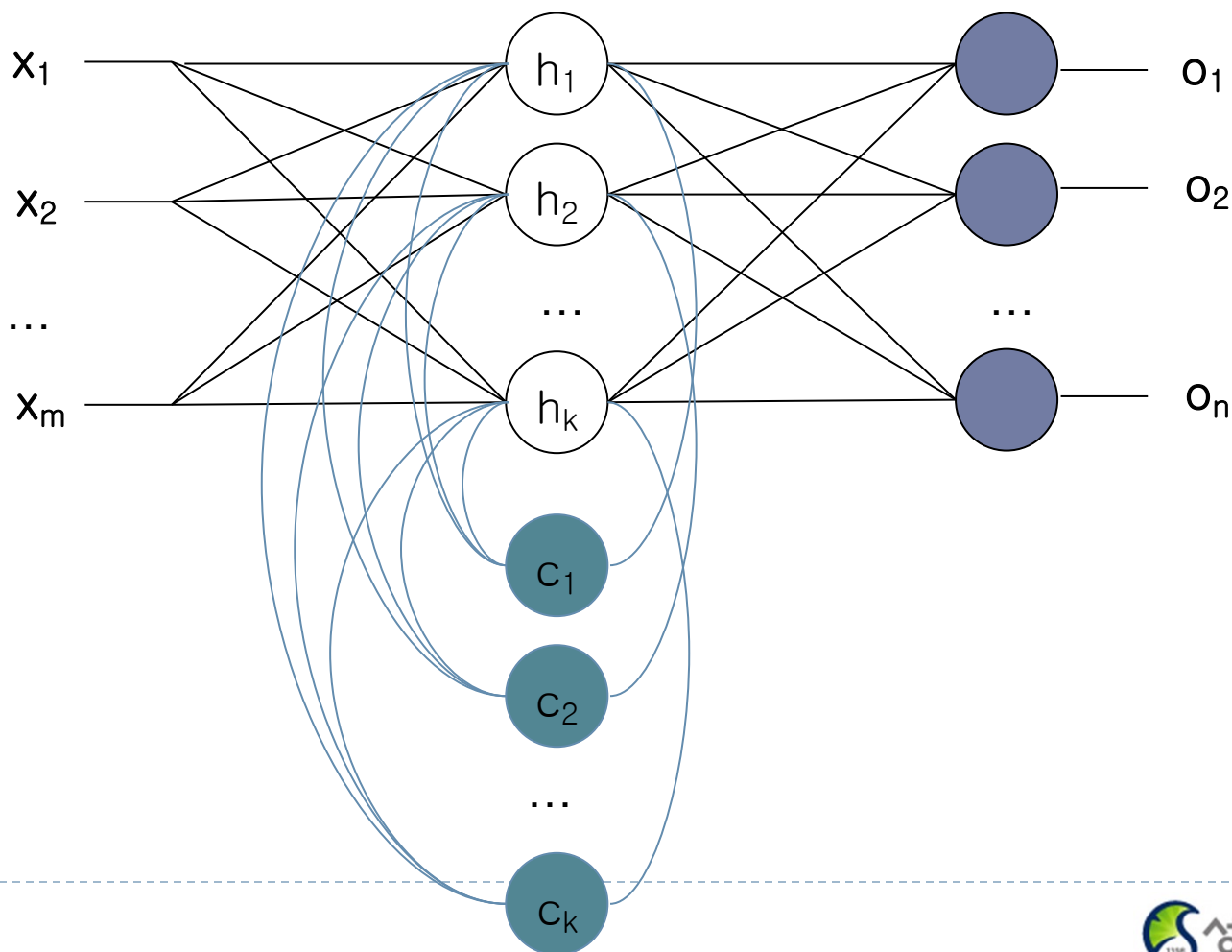
Recurrent Neural Networks

► Connections form cycles



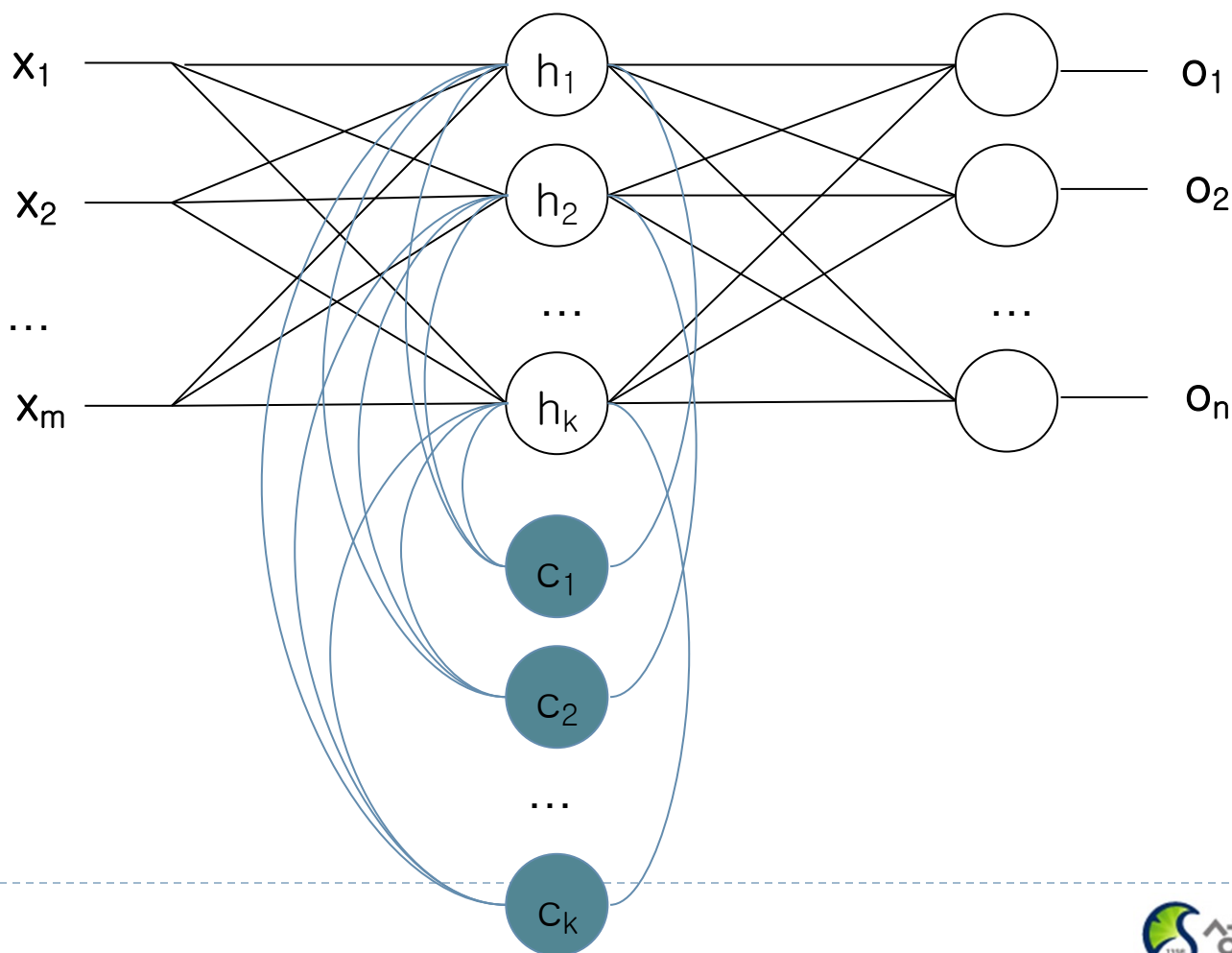
Recurrent Neural Networks

► Connections form cycles



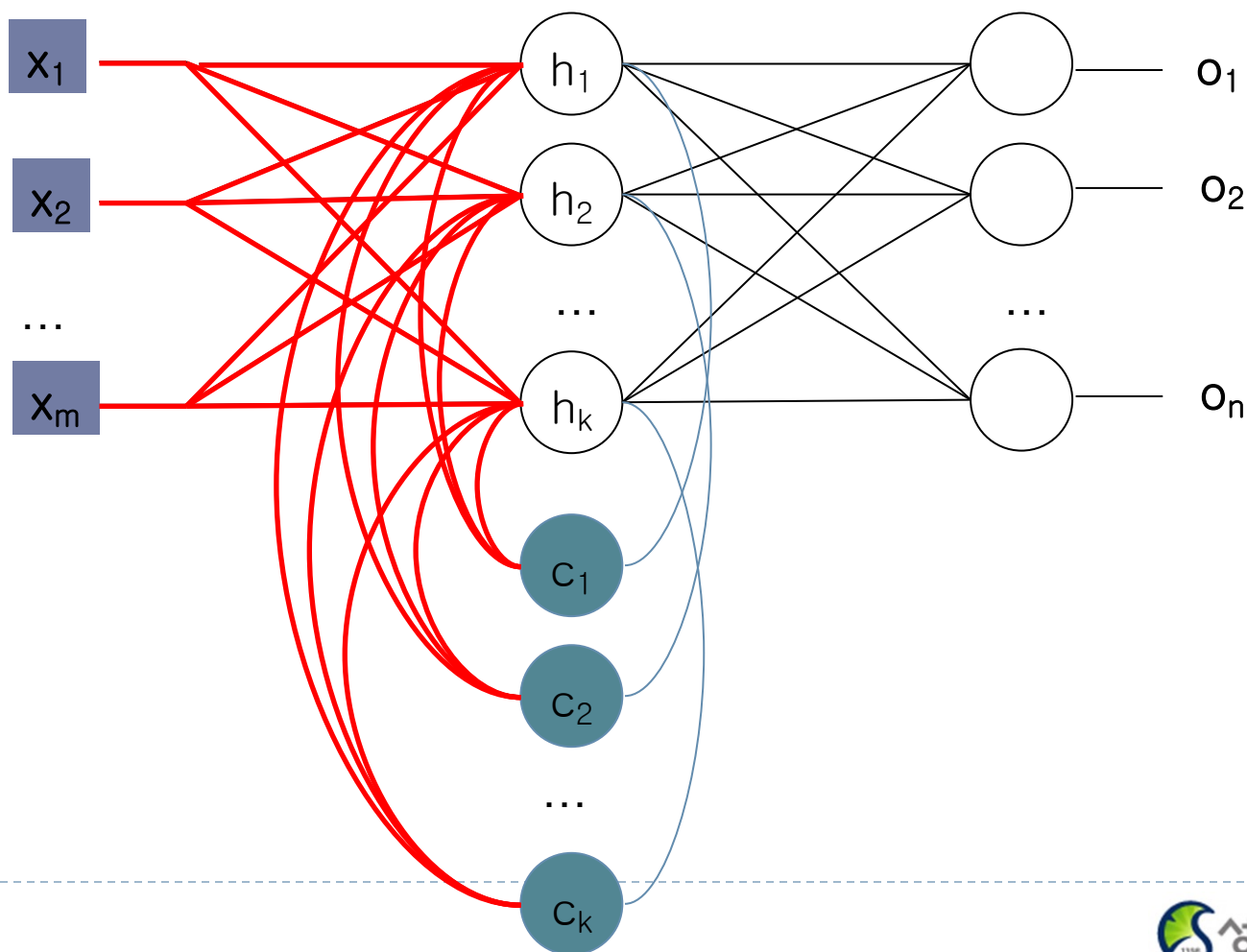
Recurrent Neural Networks

► Connections form cycles



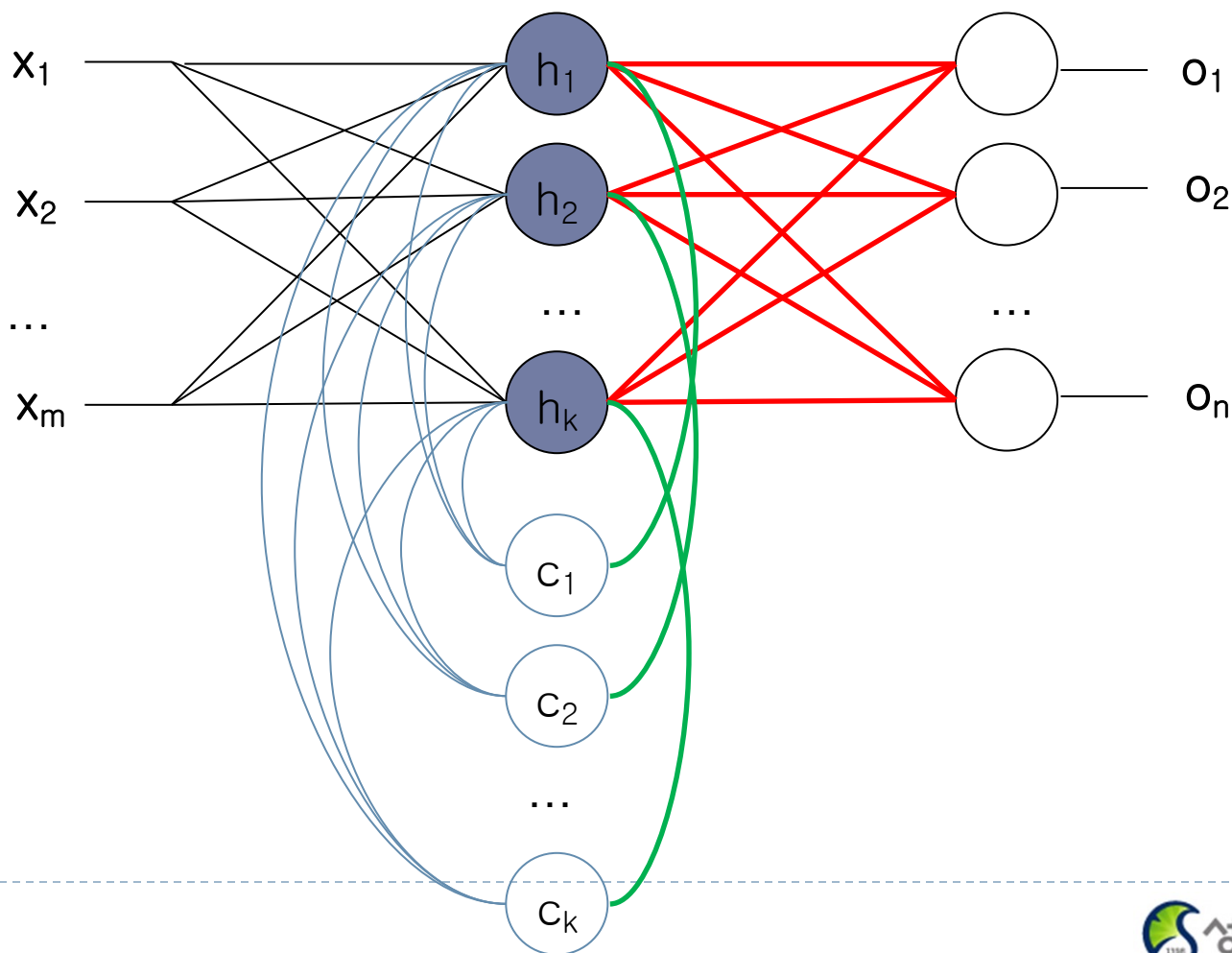
Recurrent Neural Networks

► Connections form cycles



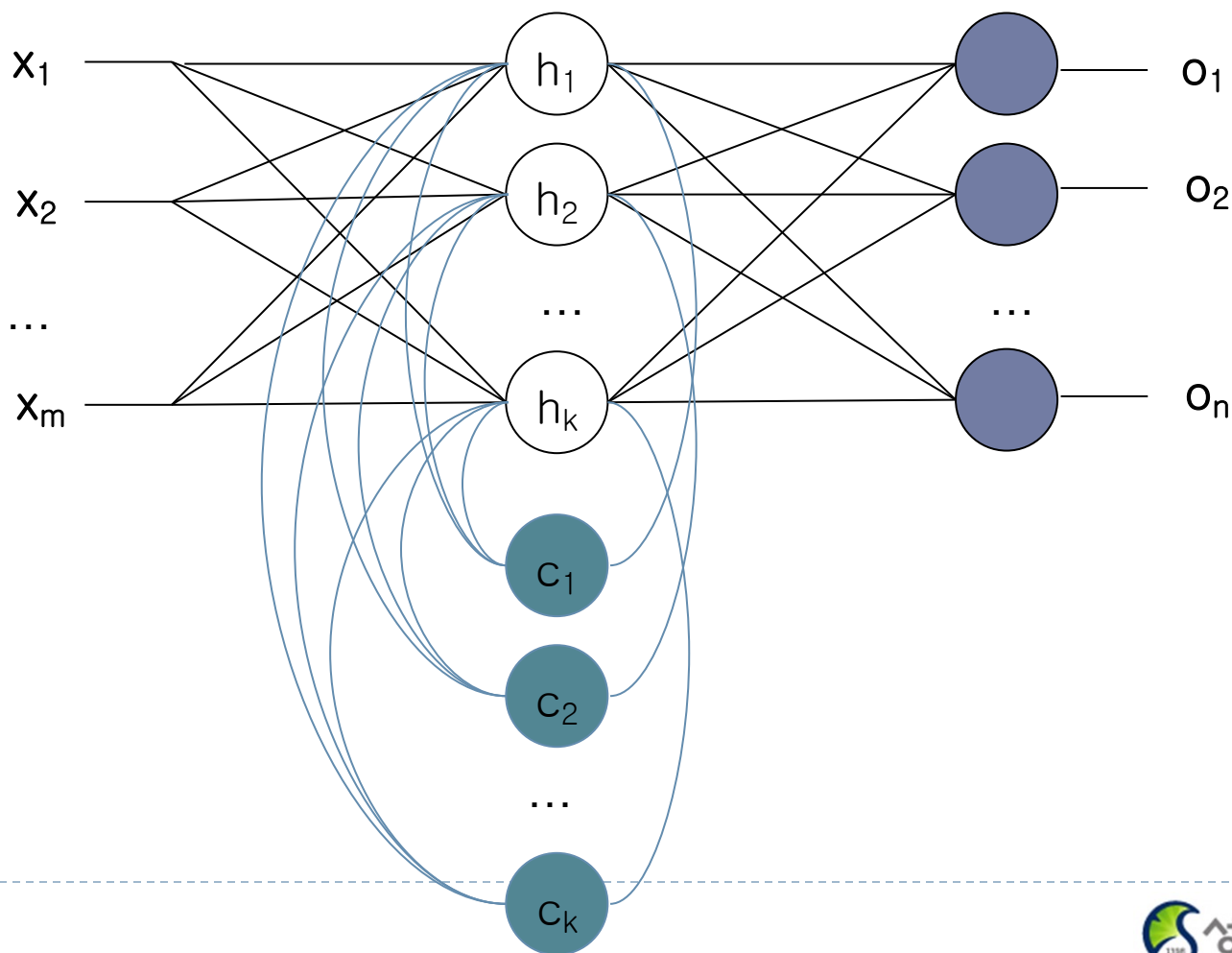
Recurrent Neural Networks

► Connections form cycles



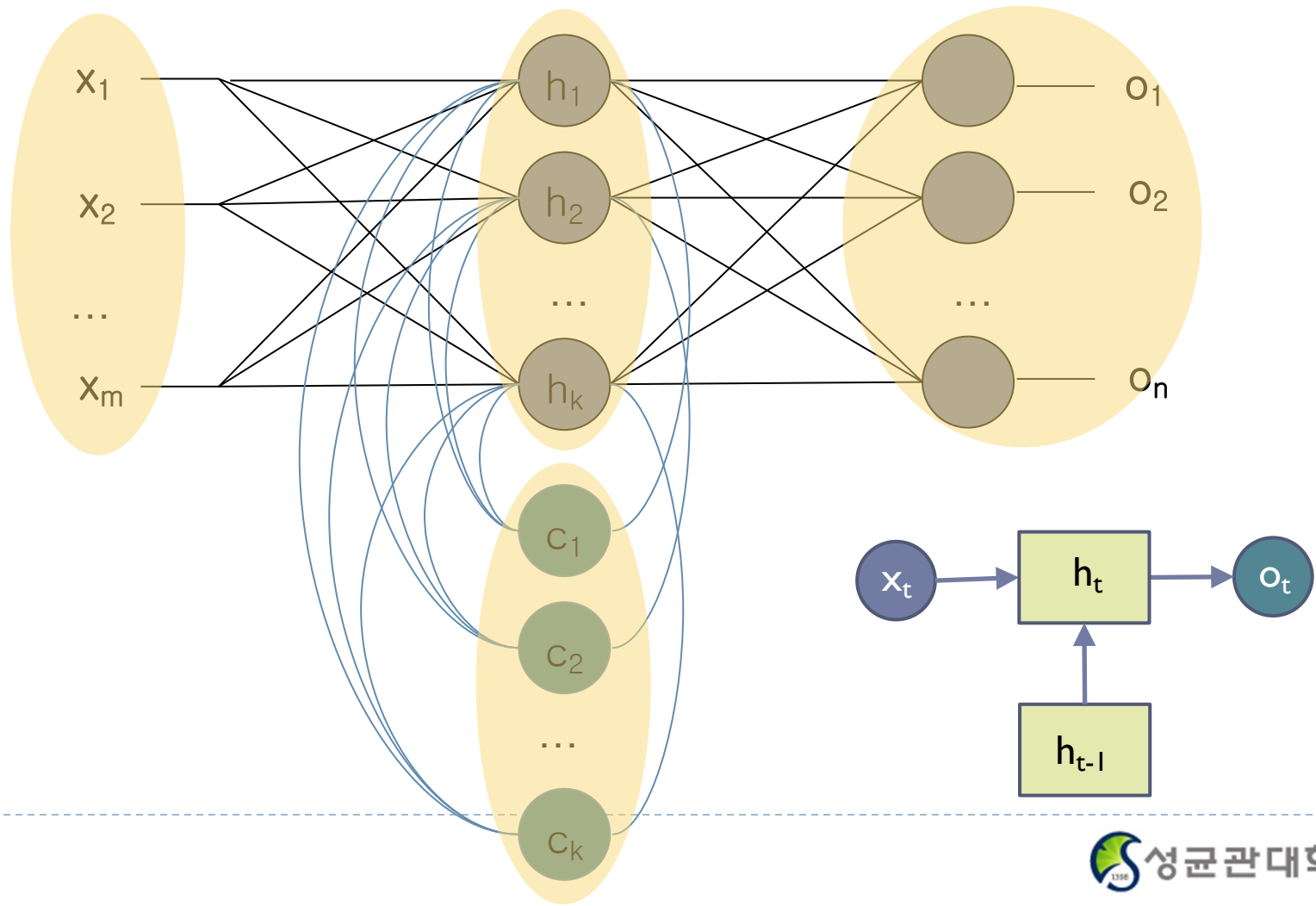
Recurrent Neural Networks

► Connections form cycles



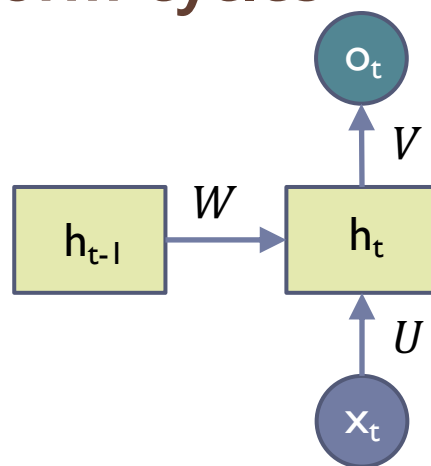
Recurrent Neural Networks

► Connections form cycles



Recurrent Neural Networks

► Connections form cycles



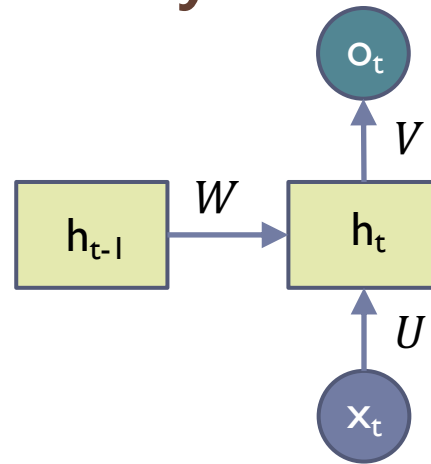
$$h_t = f(Ux_t + Wh_{t-1})$$

$$o_t = g(Vh_t)$$

- 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



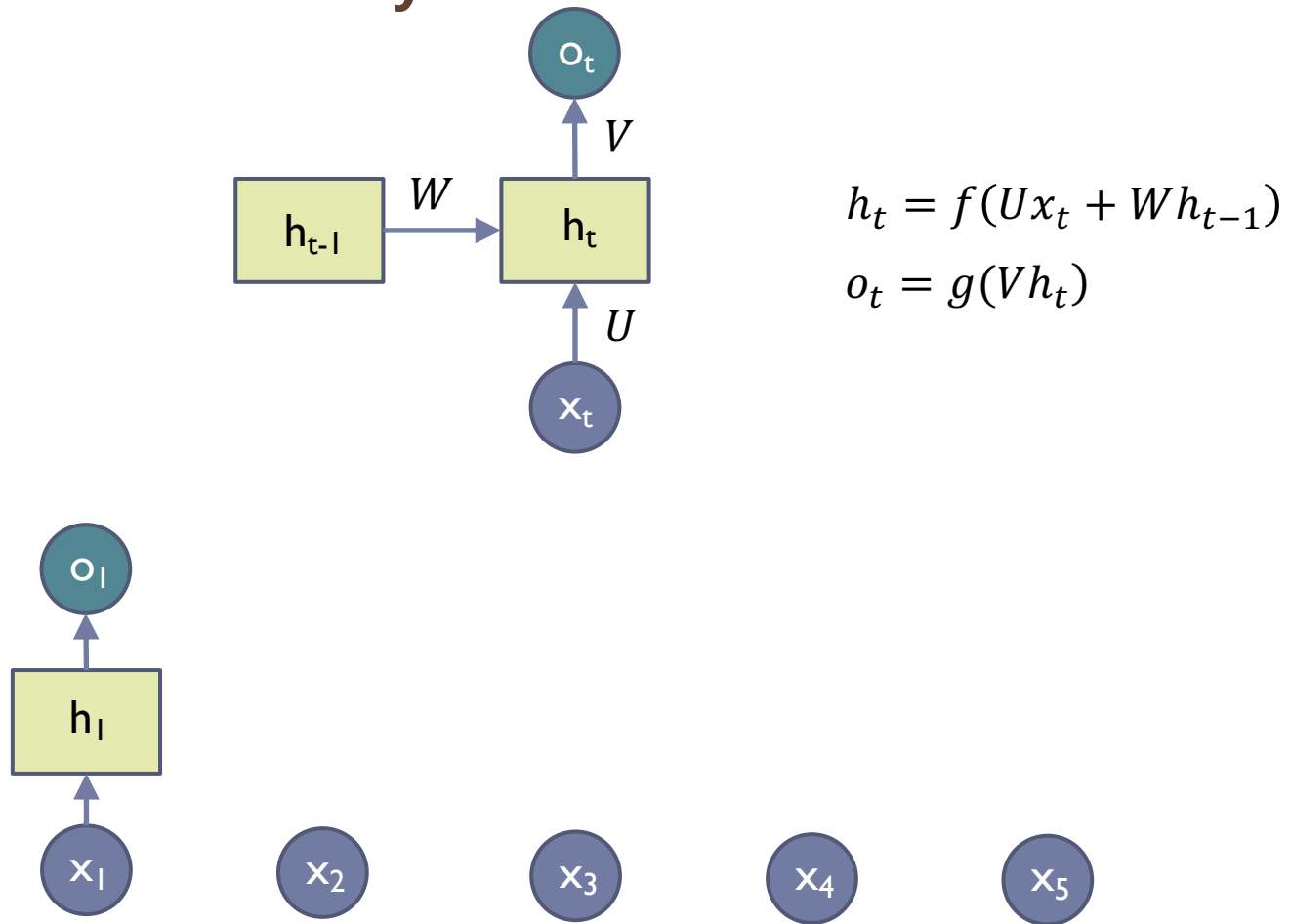
$$h_t = f(Ux_t + Wh_{t-1})$$

$$o_t = g(Vh_t)$$



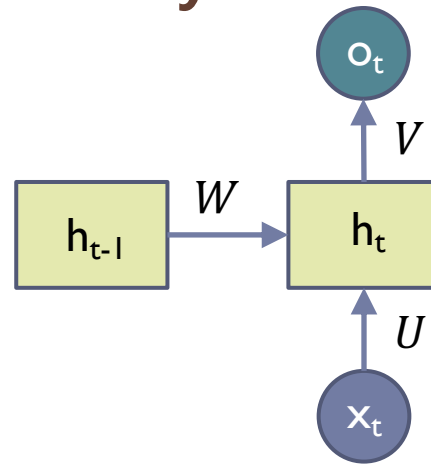
Recurrent Neural Networks

► Connections form cycles

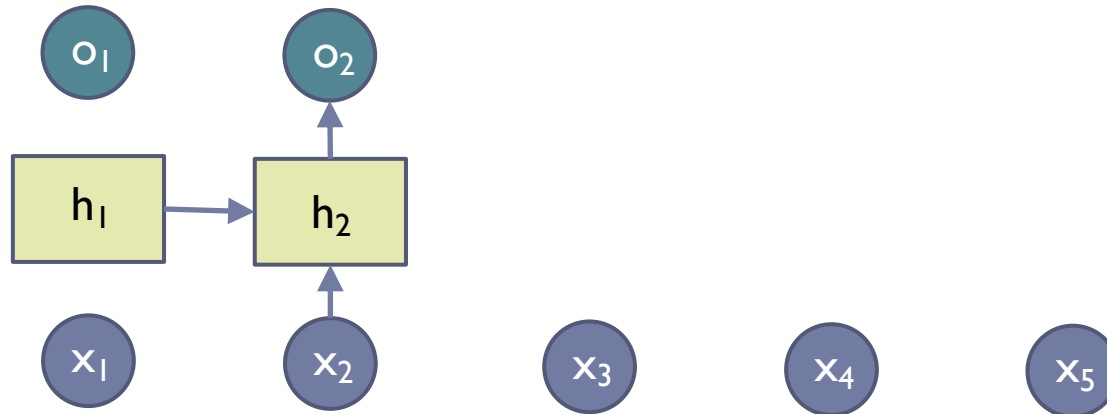


Recurrent Neural Networks

► Connections form cycles

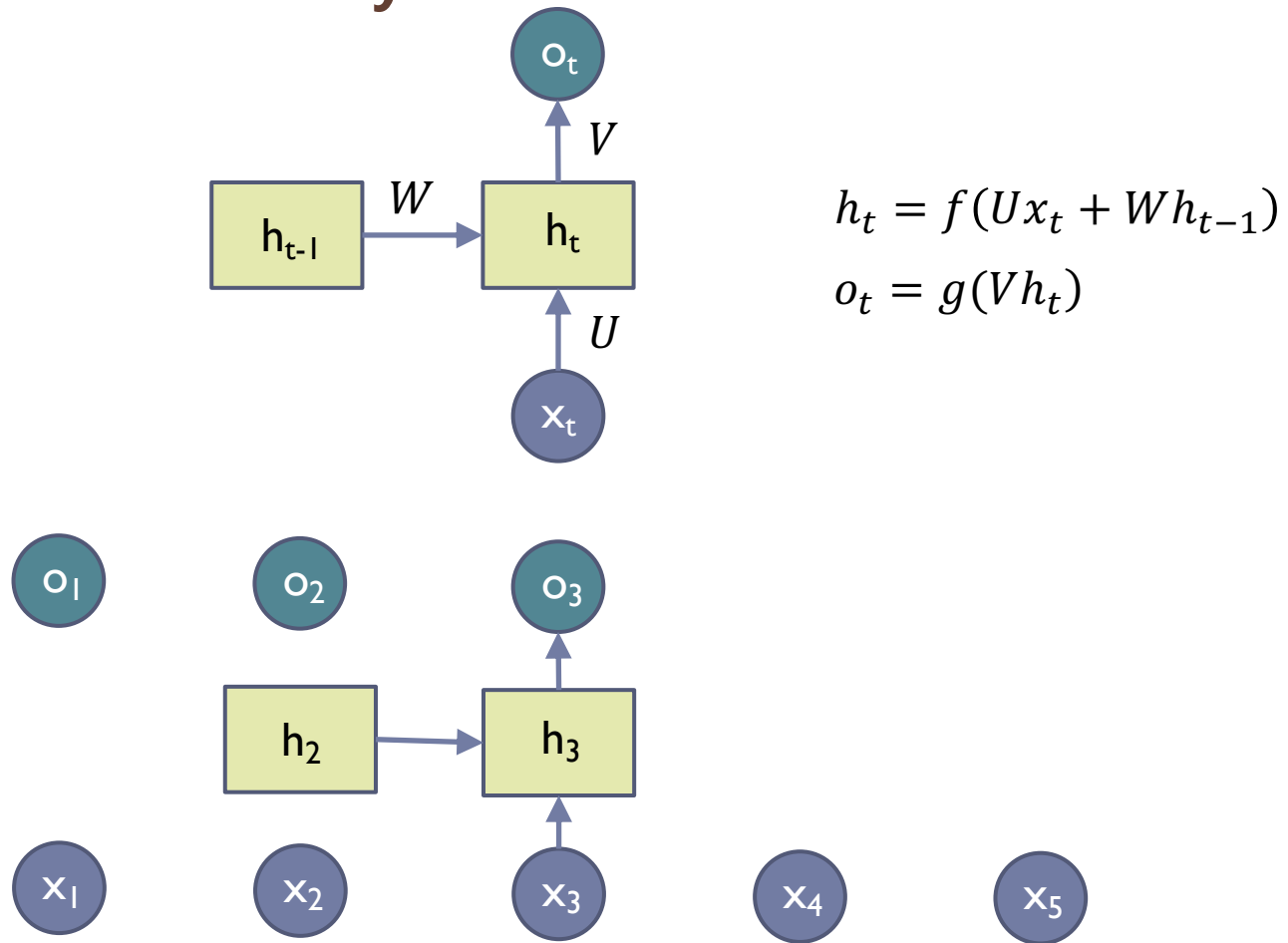


$$h_t = f(Ux_t + Wh_{t-1})$$
$$o_t = g(Vh_t)$$



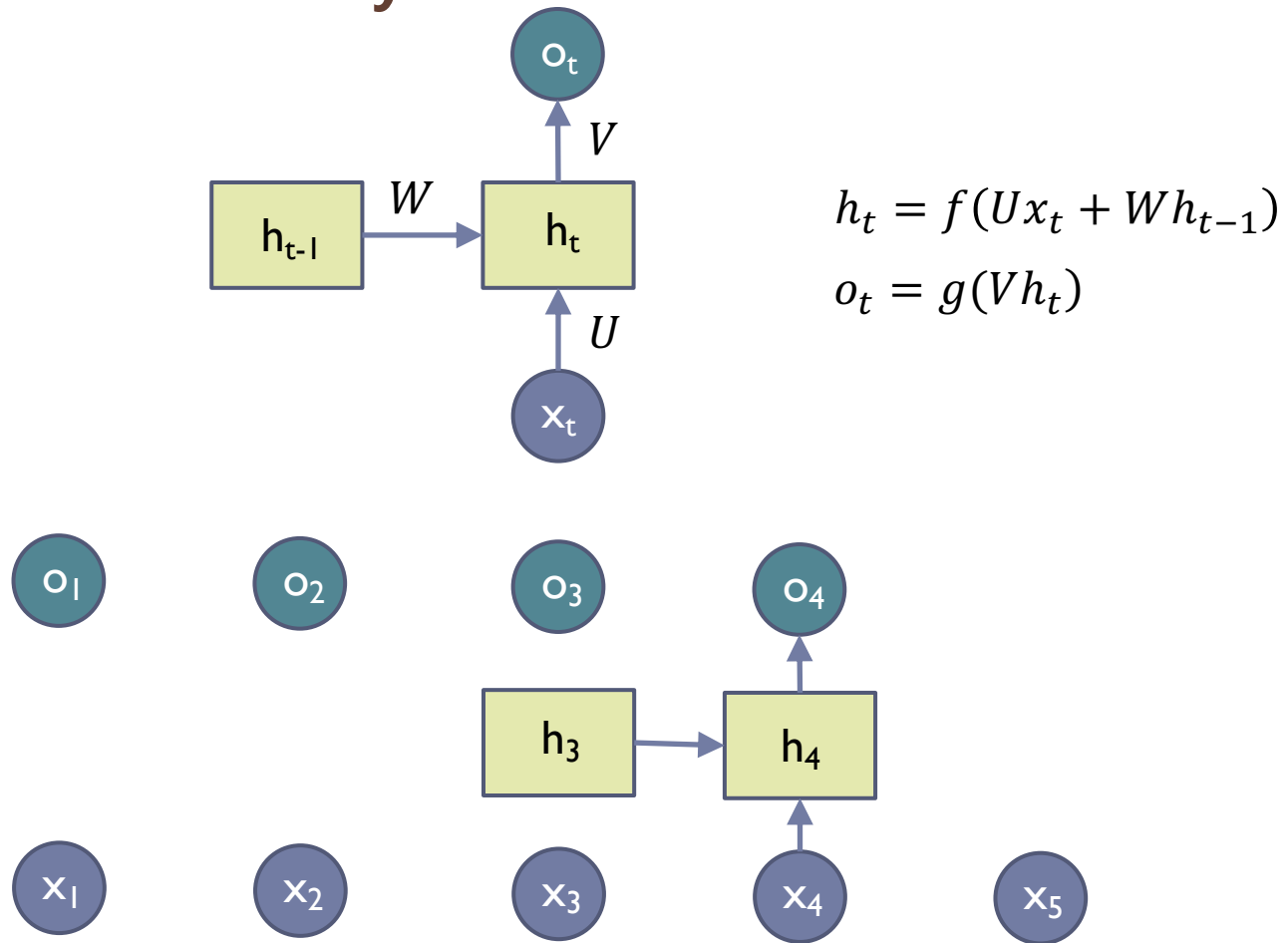
Recurrent Neural Networks

► Connections form cycles



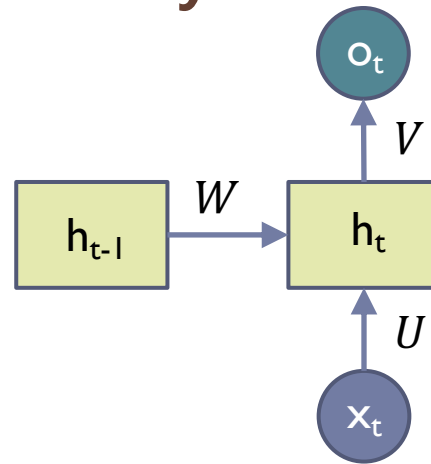
Recurrent Neural Networks

► Connections form cycles

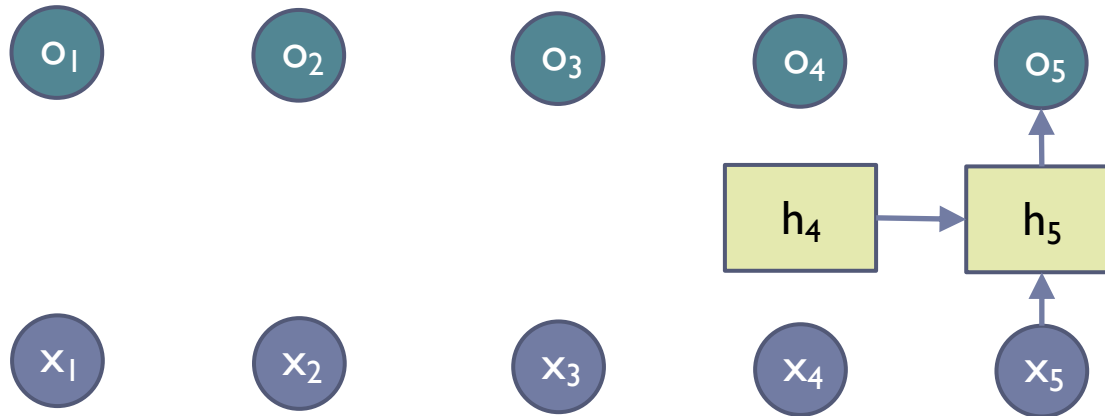


Recurrent Neural Networks

► Connections form cycles

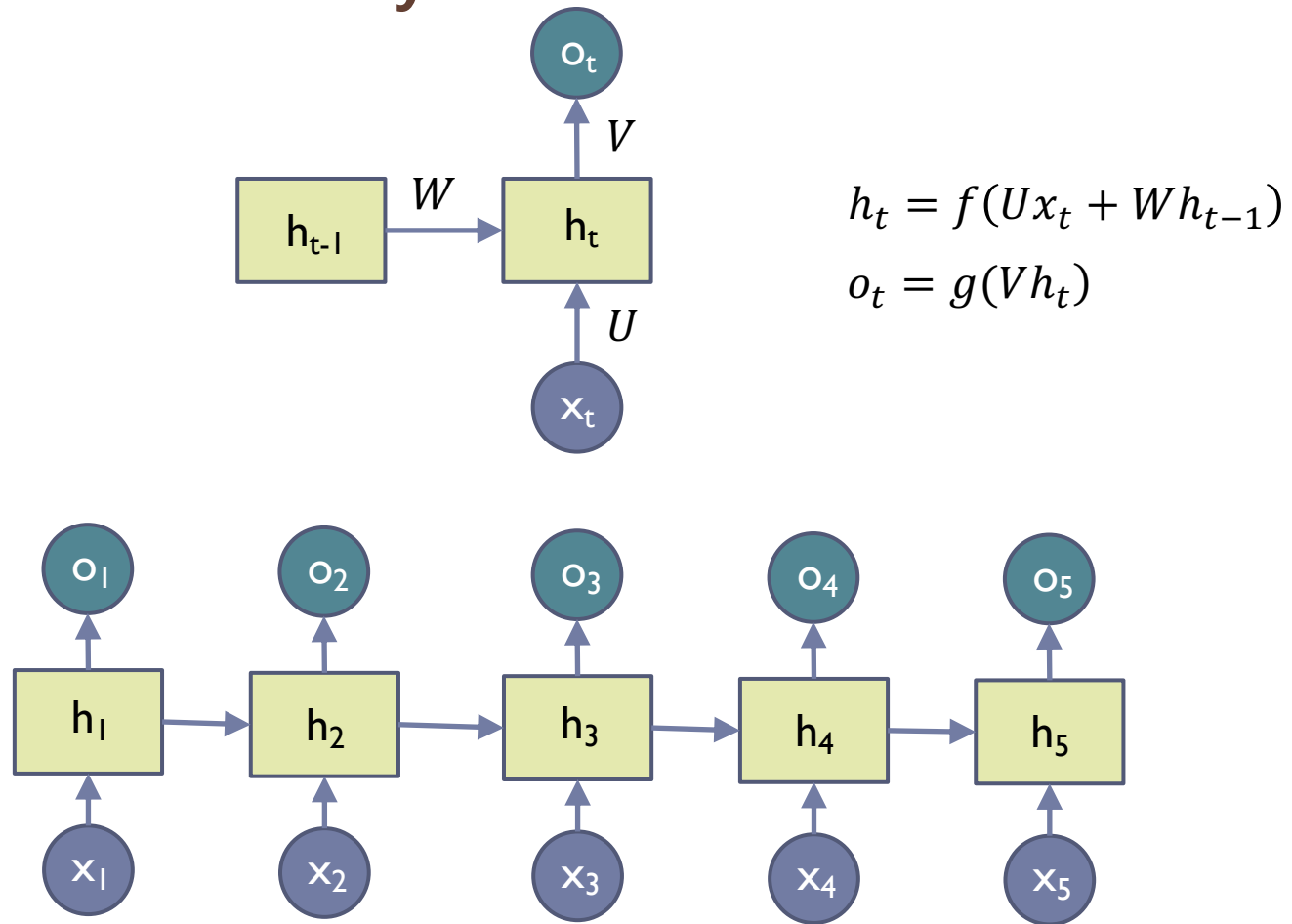


$$h_t = f(Ux_t + Wh_{t-1})$$
$$o_t = g(Vh_t)$$



Recurrent Neural Networks

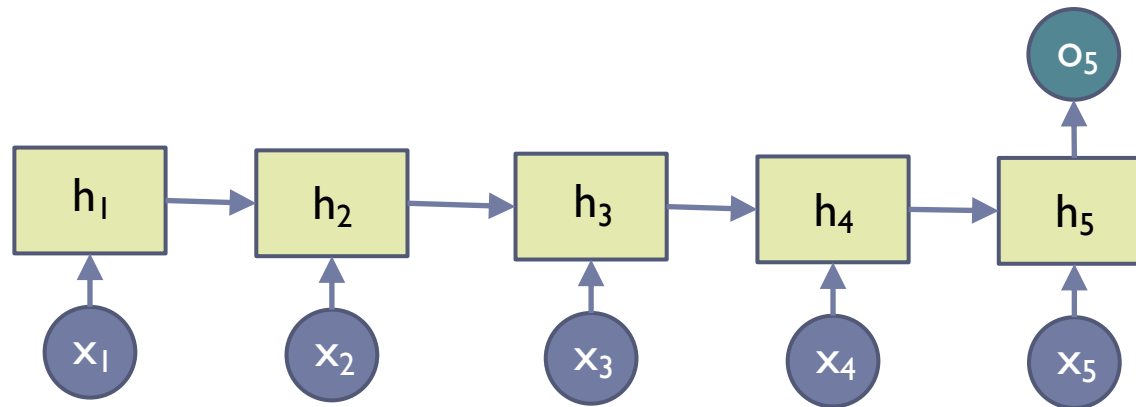
► Connections form cycles



Recurrent Neural Networks

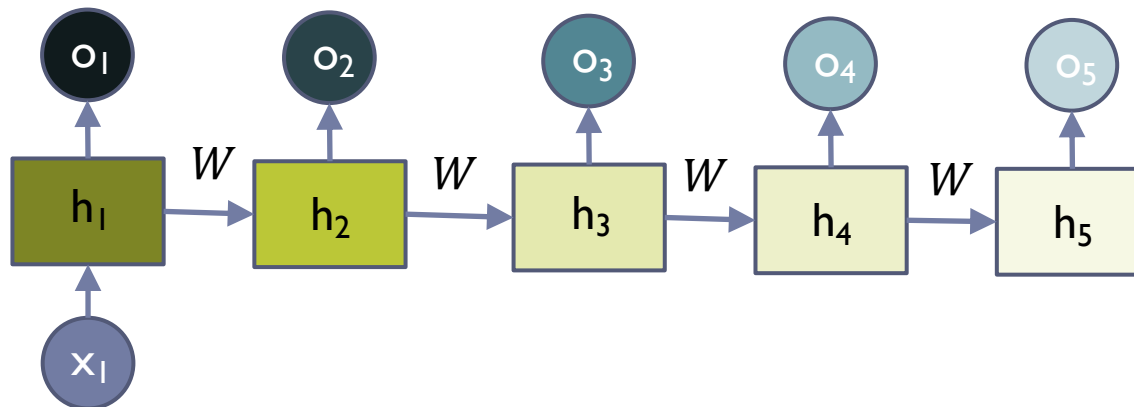
▶ Long Term Dependency

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



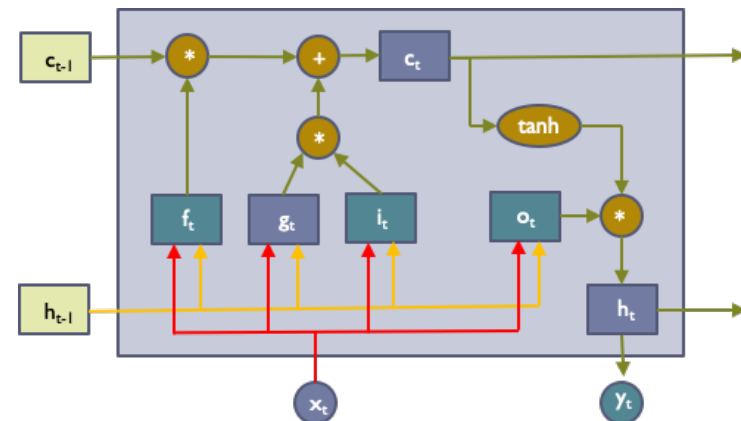
Recurrent Neural Networks

- ▶ **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)

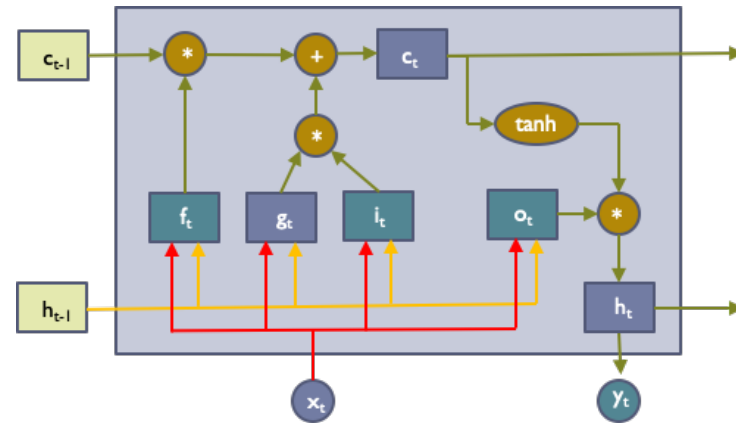
- ▶ **Capable of learning long-term dependencies.**
 - ▶ LSTM networks introduce a new structure called a memory cell
 - ▶ An LSTM can learn to bridge time intervals in excess of 1000 steps
 - ▶ **Gate units that learn to open and close access to the past**
 - ▶ Input gate
 - ▶ Forget gate
 - ▶ Output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



Long Short-Term Memory (LSTM)

► Equations

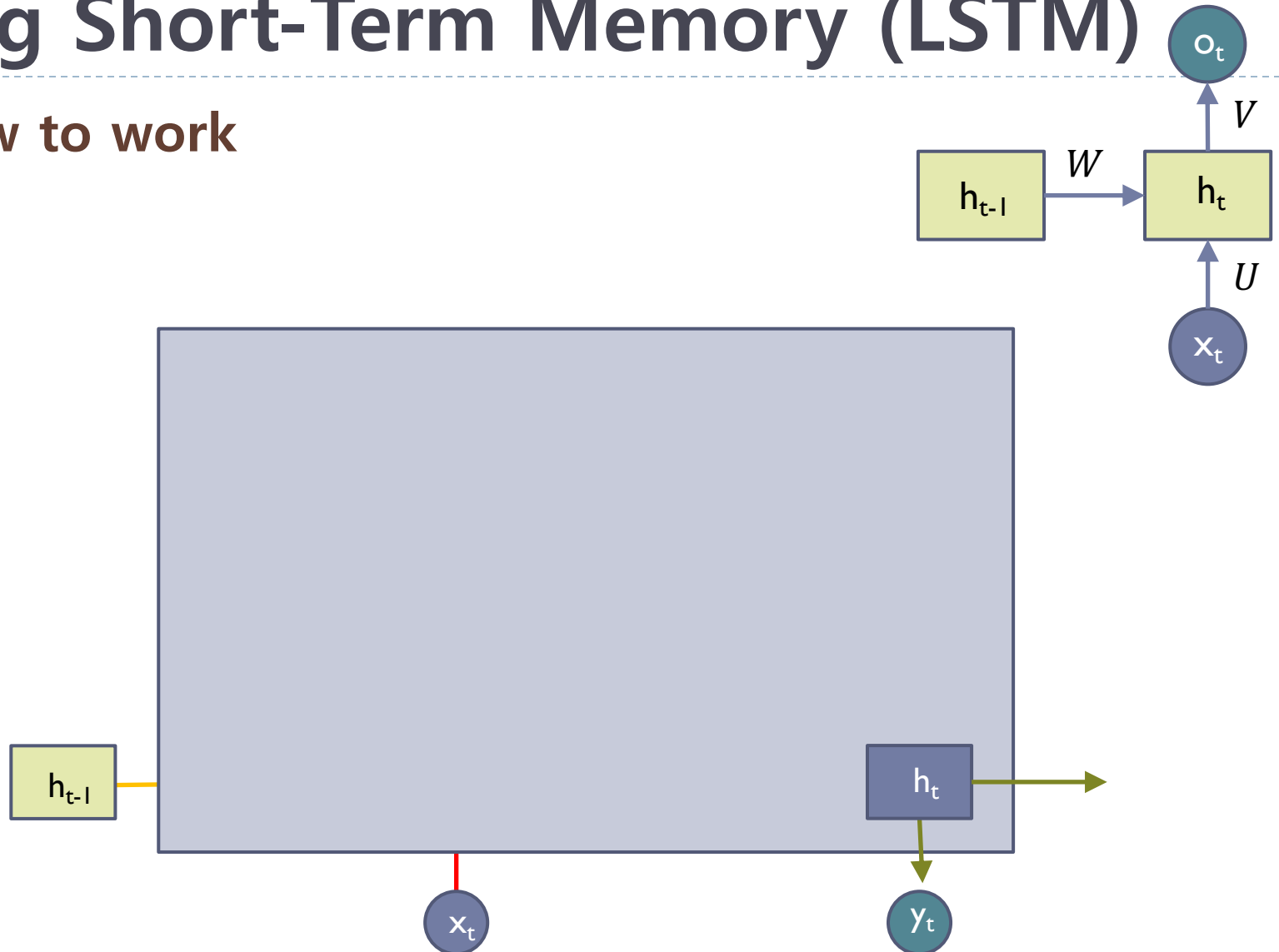
- i : input gate
- f : forget gate
- o : output gate
- g : self-recurrent
- c_t : internal memory
- h_t : hidden state
- y : final output



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

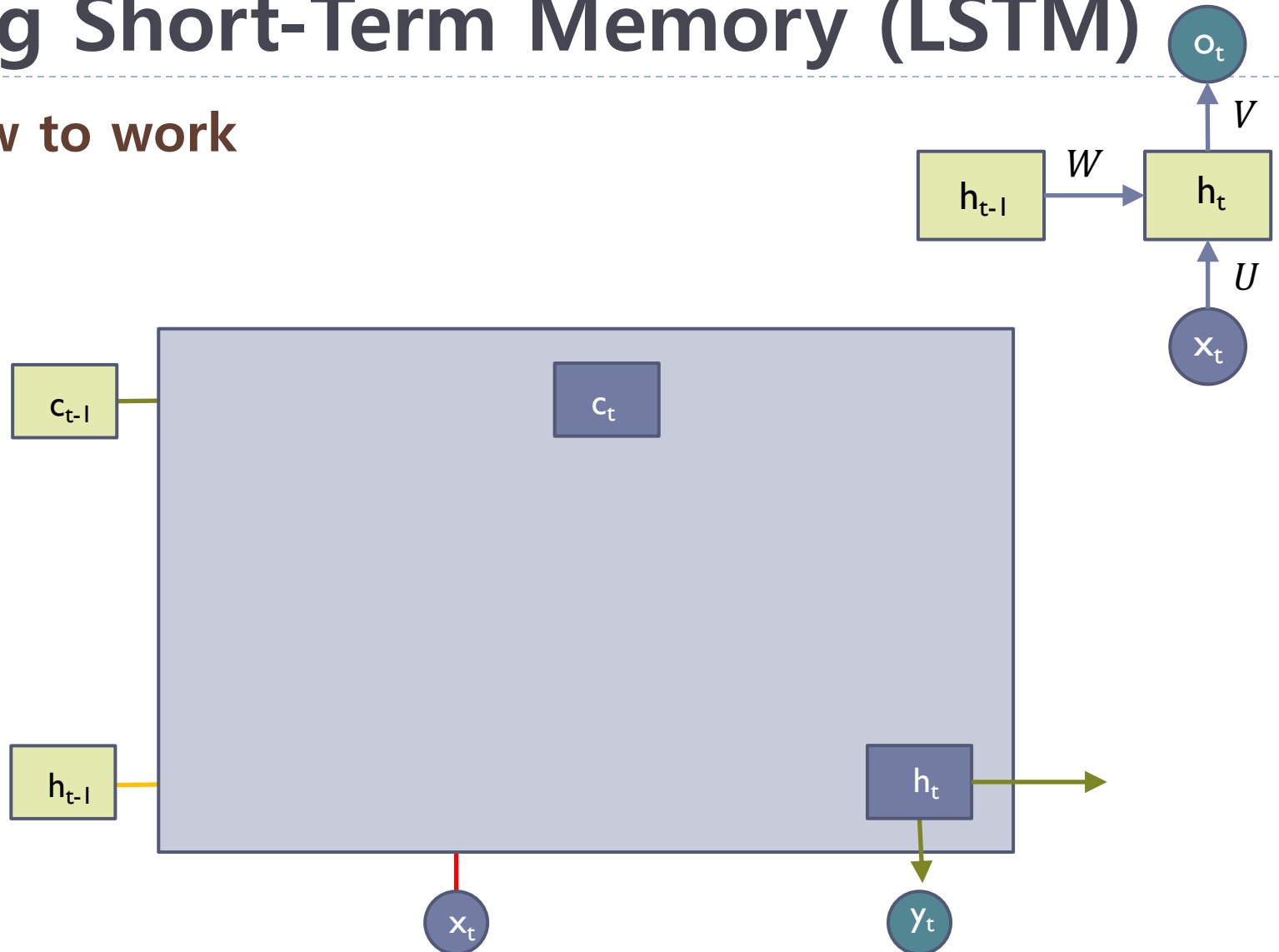
Long Short-Term Memory (LSTM)

▶ How to work



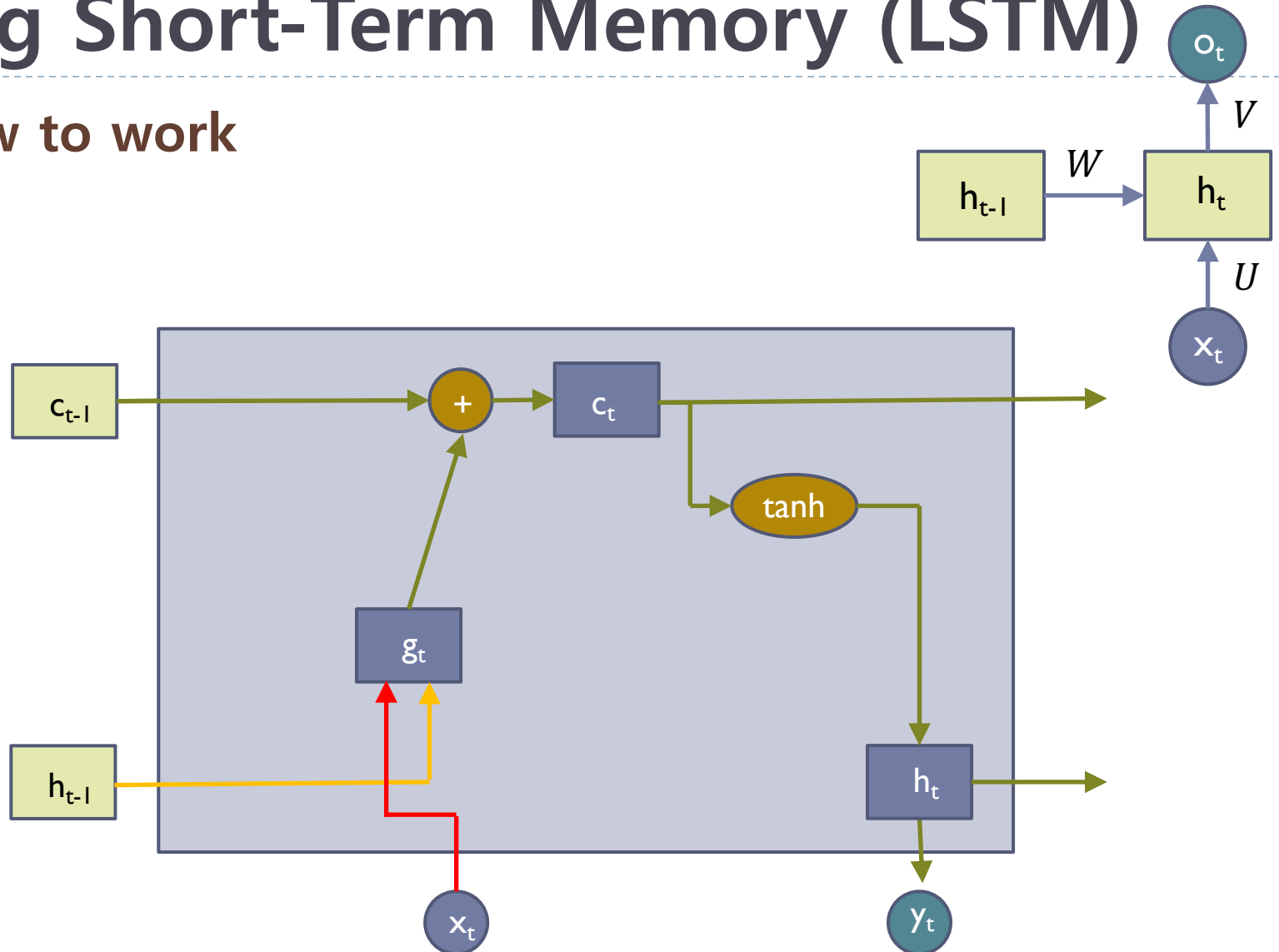
Long Short-Term Memory (LSTM)

▶ How to work



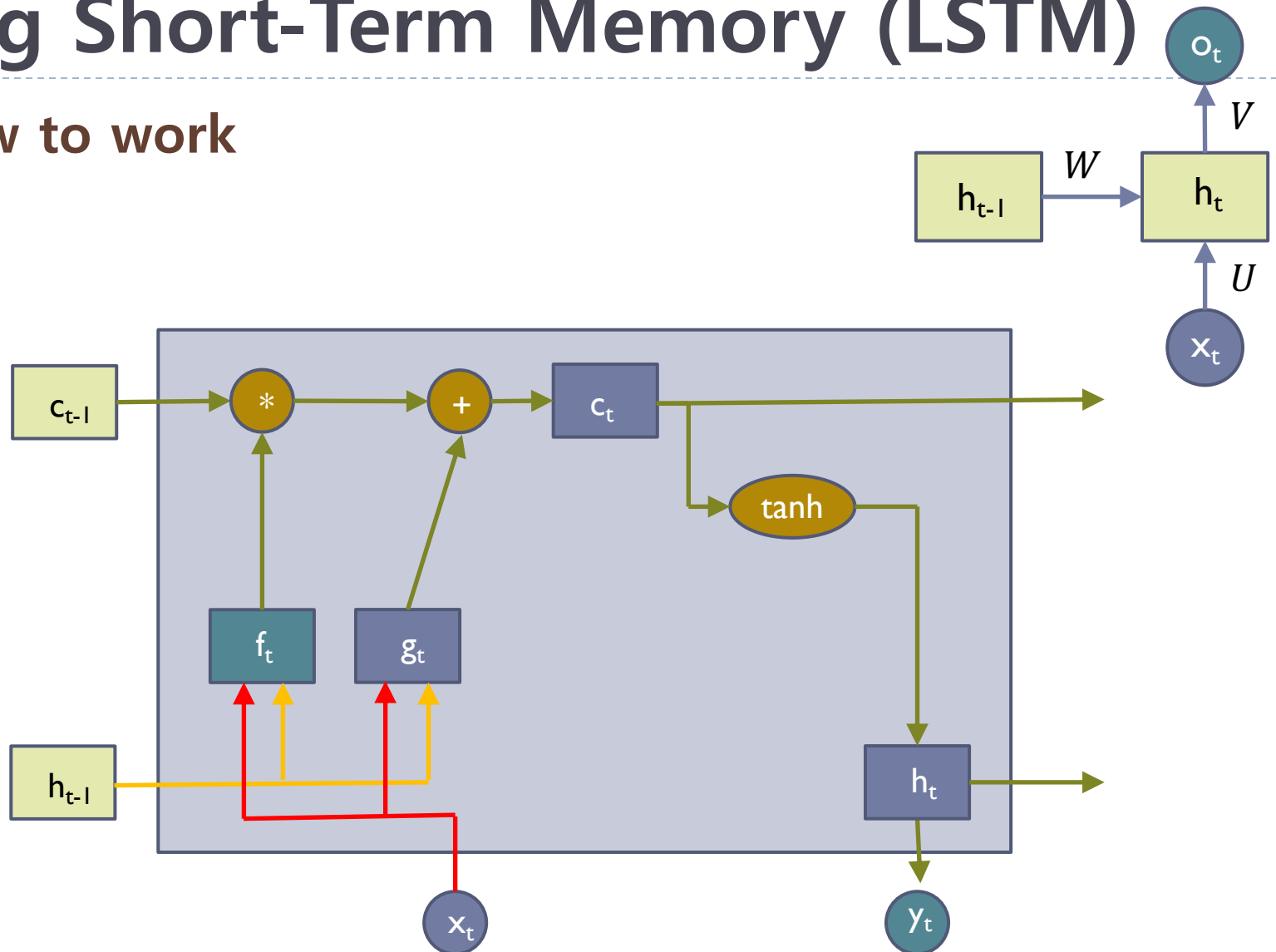
Long Short-Term Memory (LSTM)

► How to work



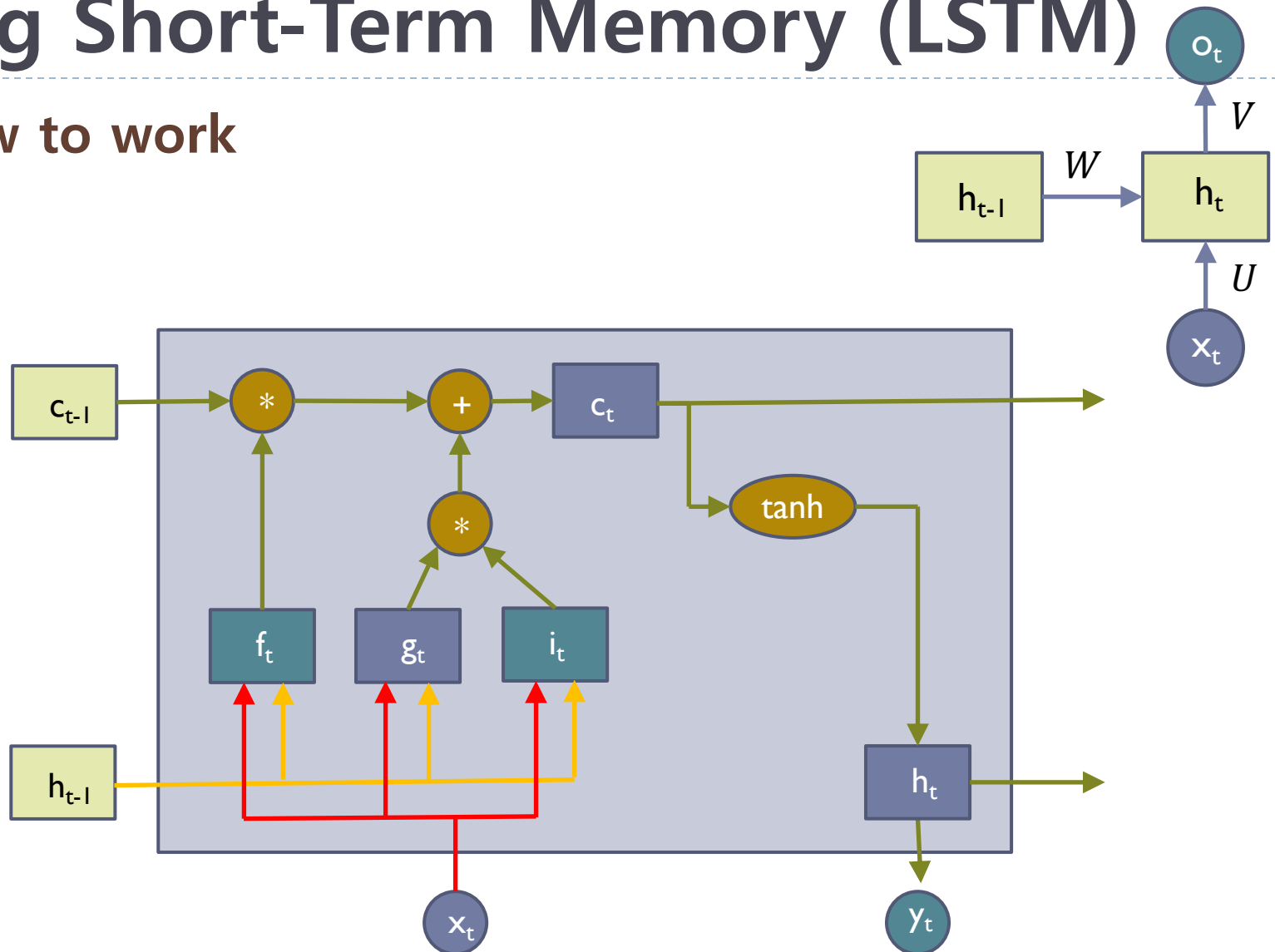
Long Short-Term Memory (LSTM)

► How to work



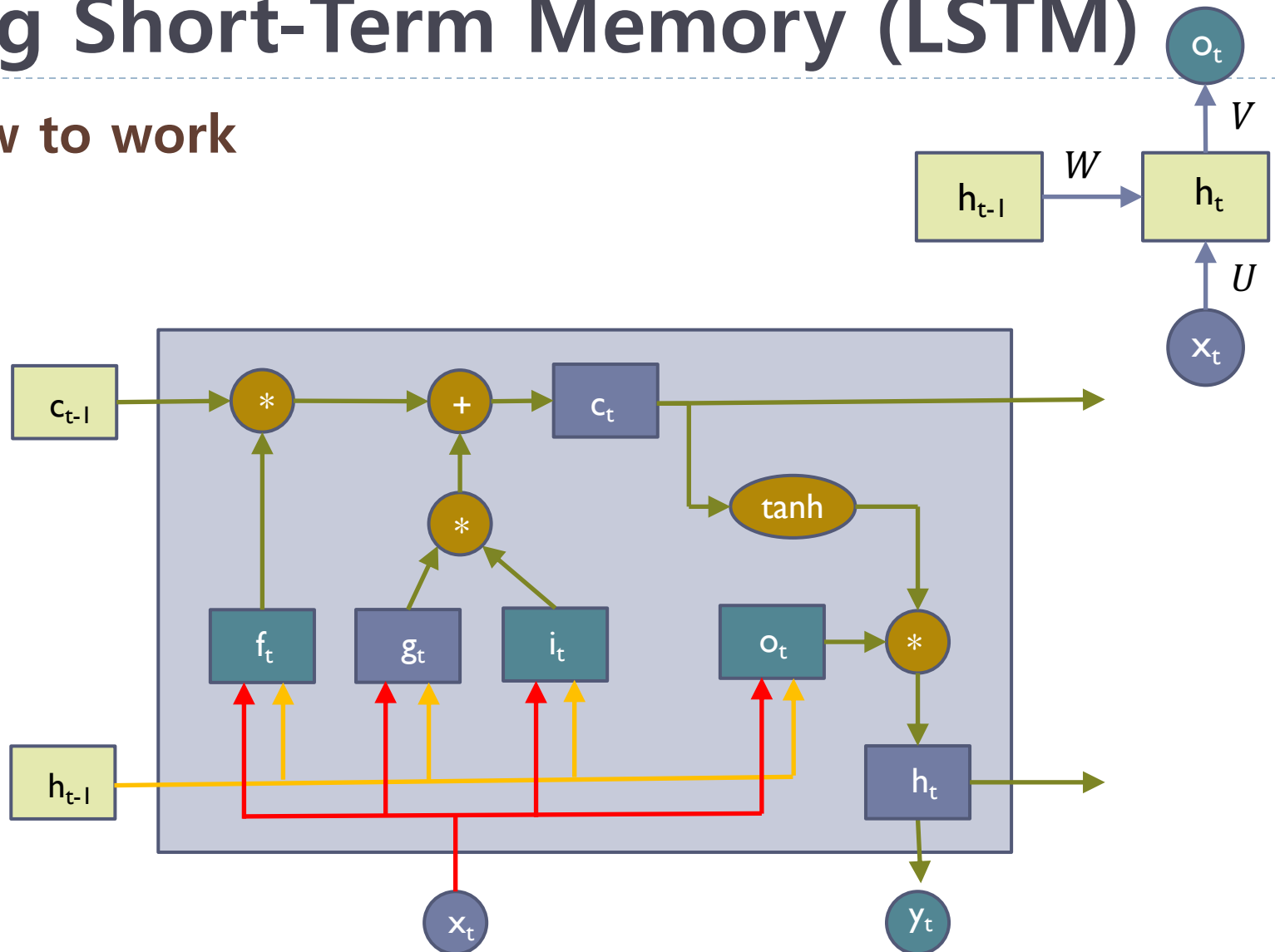
Long Short-Term Memory (LSTM)

► How to work



Long Short-Term Memory (LSTM)

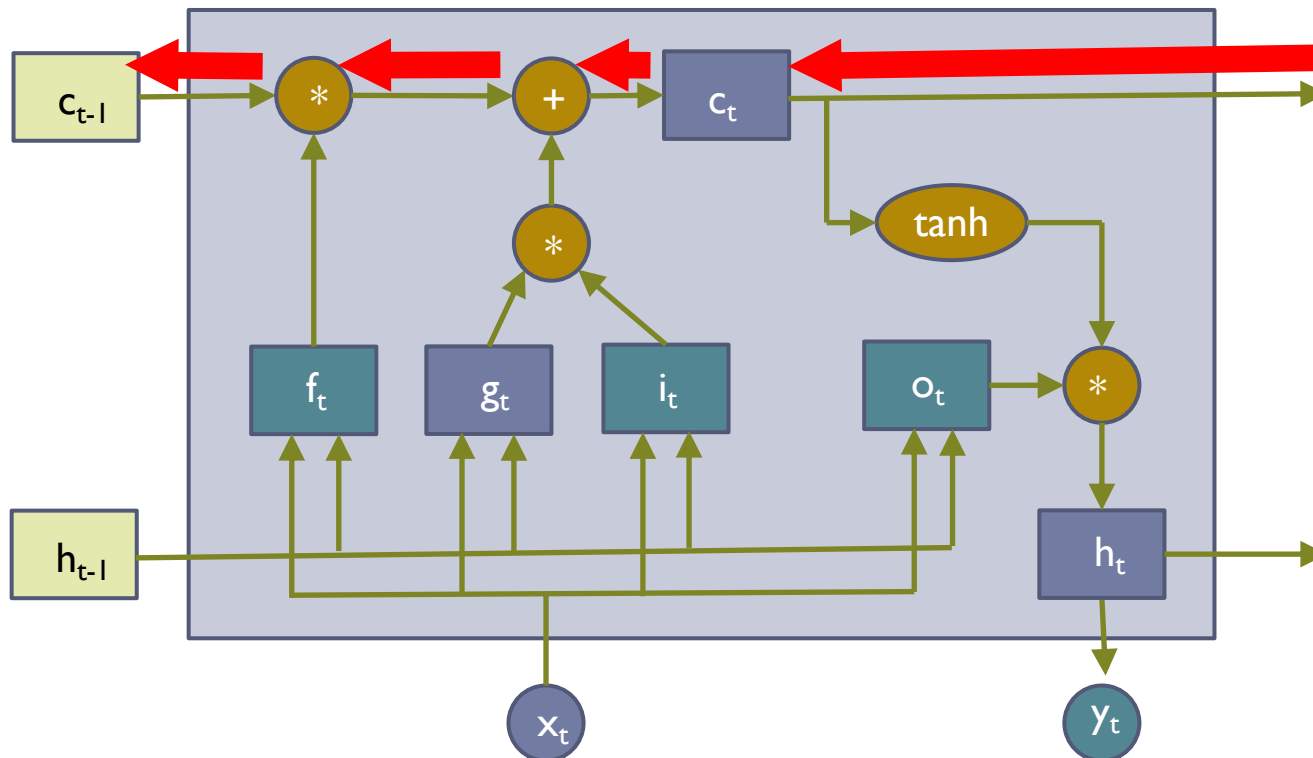
► How to work



Long Short-Term Memory (LSTM)

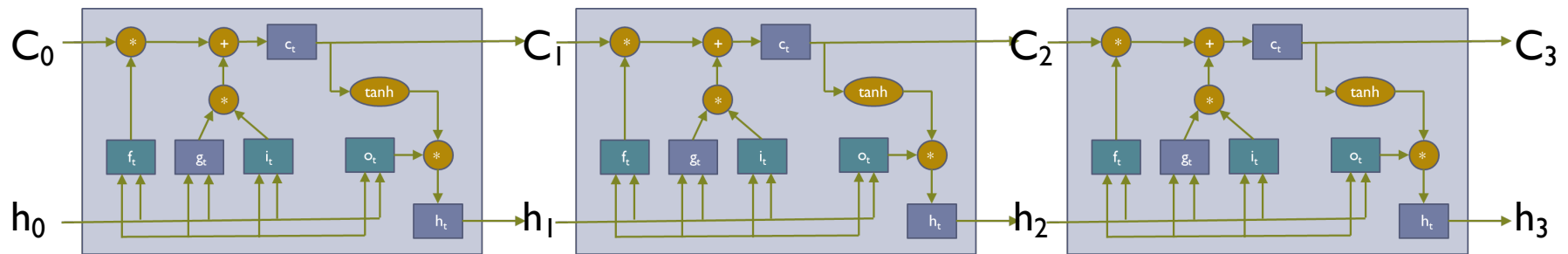
► Gradient Flow

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 C_0 , C_1 , and C_2 . A red arrow at the top indicates the direction of the hidden state sequence from left to right.

Each time step C_t receives an input vector x_t (represented by f_t, g_t, i_t) and the previous hidden state h_{t-1} . The hidden state is updated to h_t using a recurrent neural network structure. The diagram shows the flow of information from inputs to hidden states and outputs, with a red arrow indicating the direction of the hidden state sequence.

The hidden state h_t is updated using a recurrent neural network structure. The diagram shows the flow of information from inputs to hidden states and outputs, with a red arrow indicating the direction of the hidden state sequence.

The diagram illustrates the FC1000 architecture. It starts with an 'Input' layer (grey), followed by a 'Pool' layer (blue), and then a '1x7 conv 64/12' layer (orange). The main body of the network consists of multiple stages of convolutional layers, each represented by a colored block (green, blue, or purple) containing three '3x3 conv 64' units. These stages are connected sequentially, with an ellipsis indicating intermediate stages. The final stage is a 'Pool' layer (blue) at the top, which is connected to the 'FC1000' layer (red) at the bottom. A red arrow on the left points downwards, indicating the flow of data from input to output.

Gated Recurrent Units

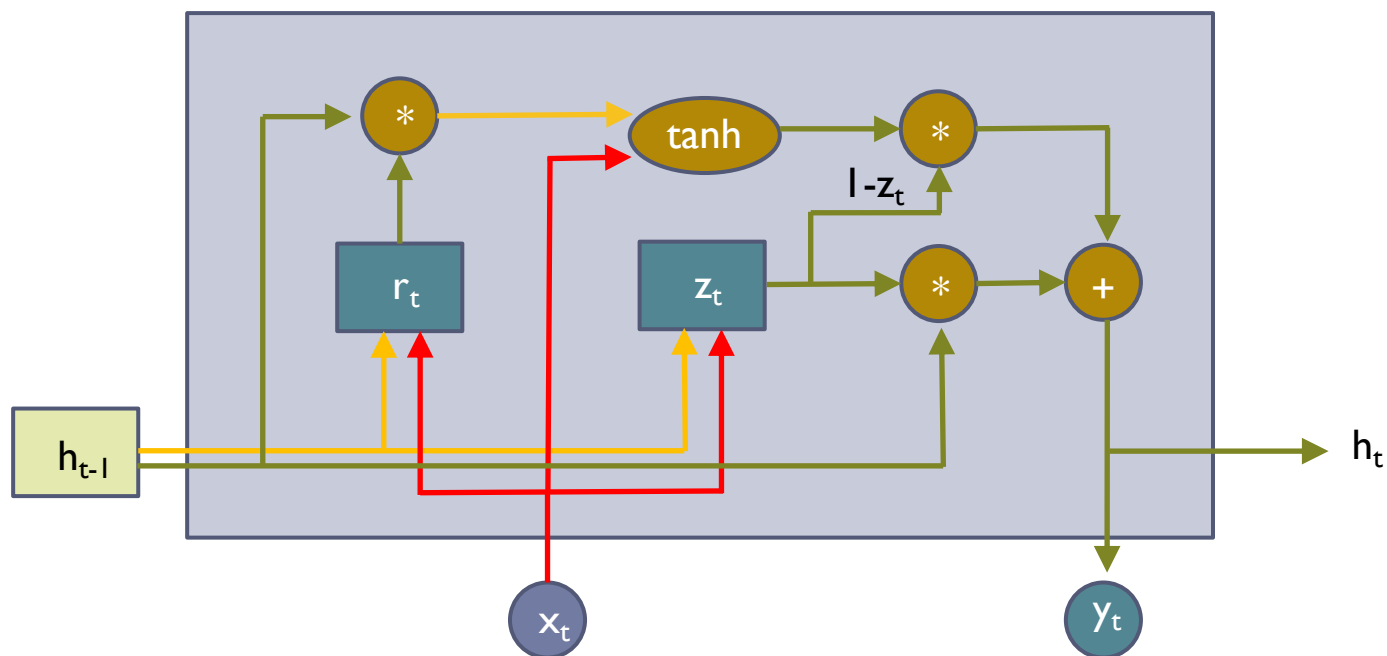
► Structure

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

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

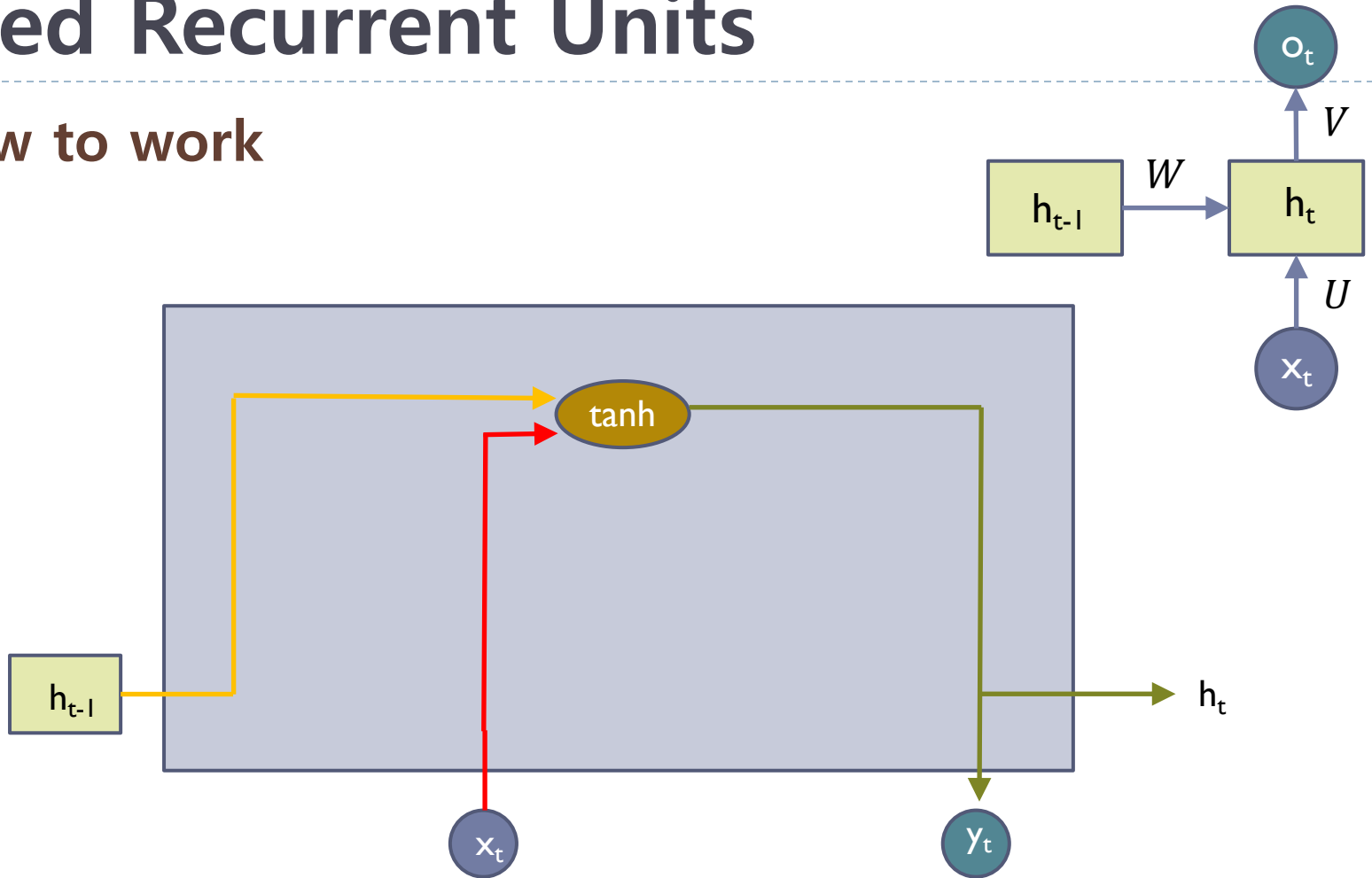
$$\tilde{h}_t = \tanh(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h)$$

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



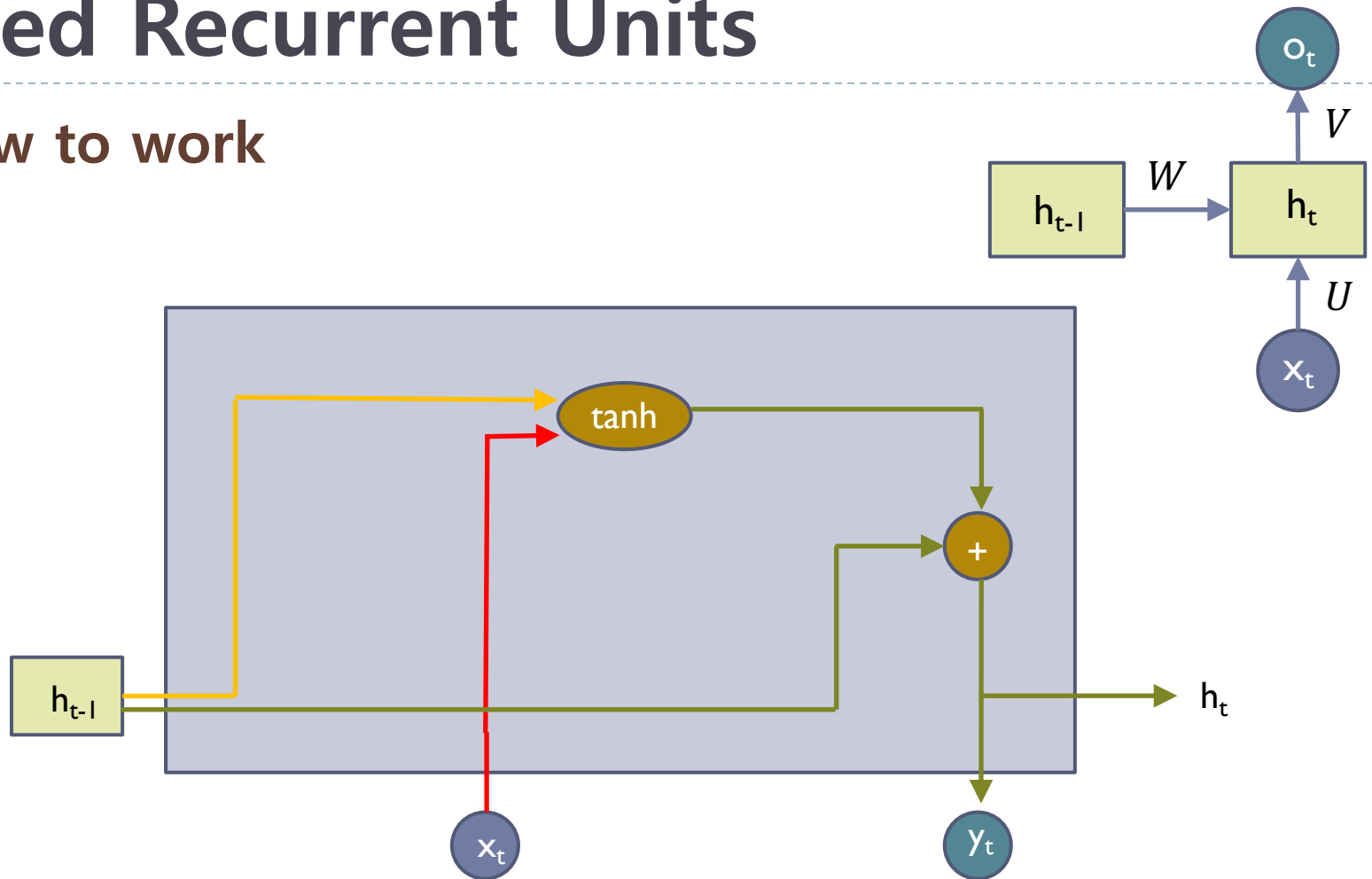
Gated Recurrent Units

▶ How to work



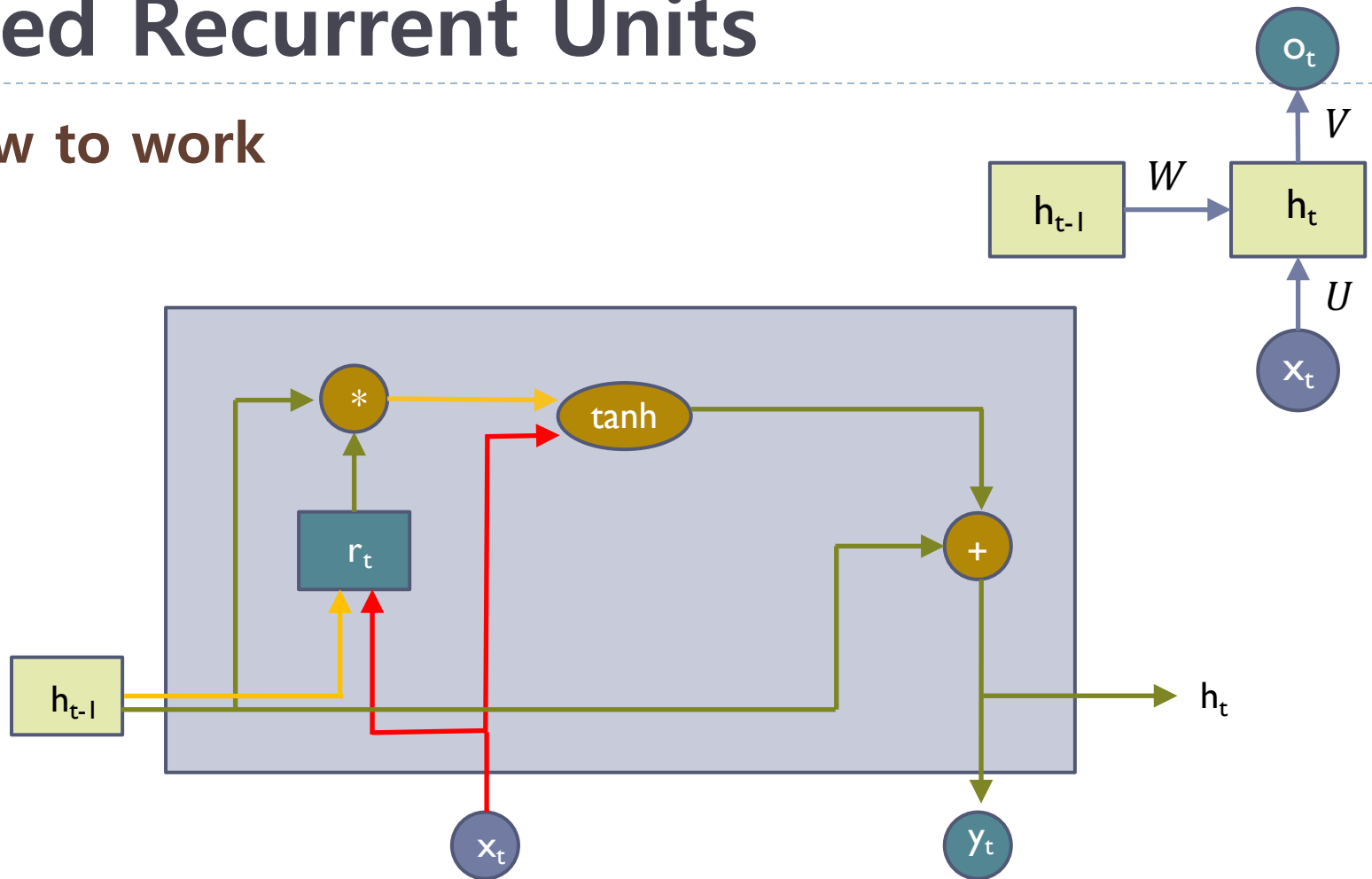
Gated Recurrent Units

▶ How to work



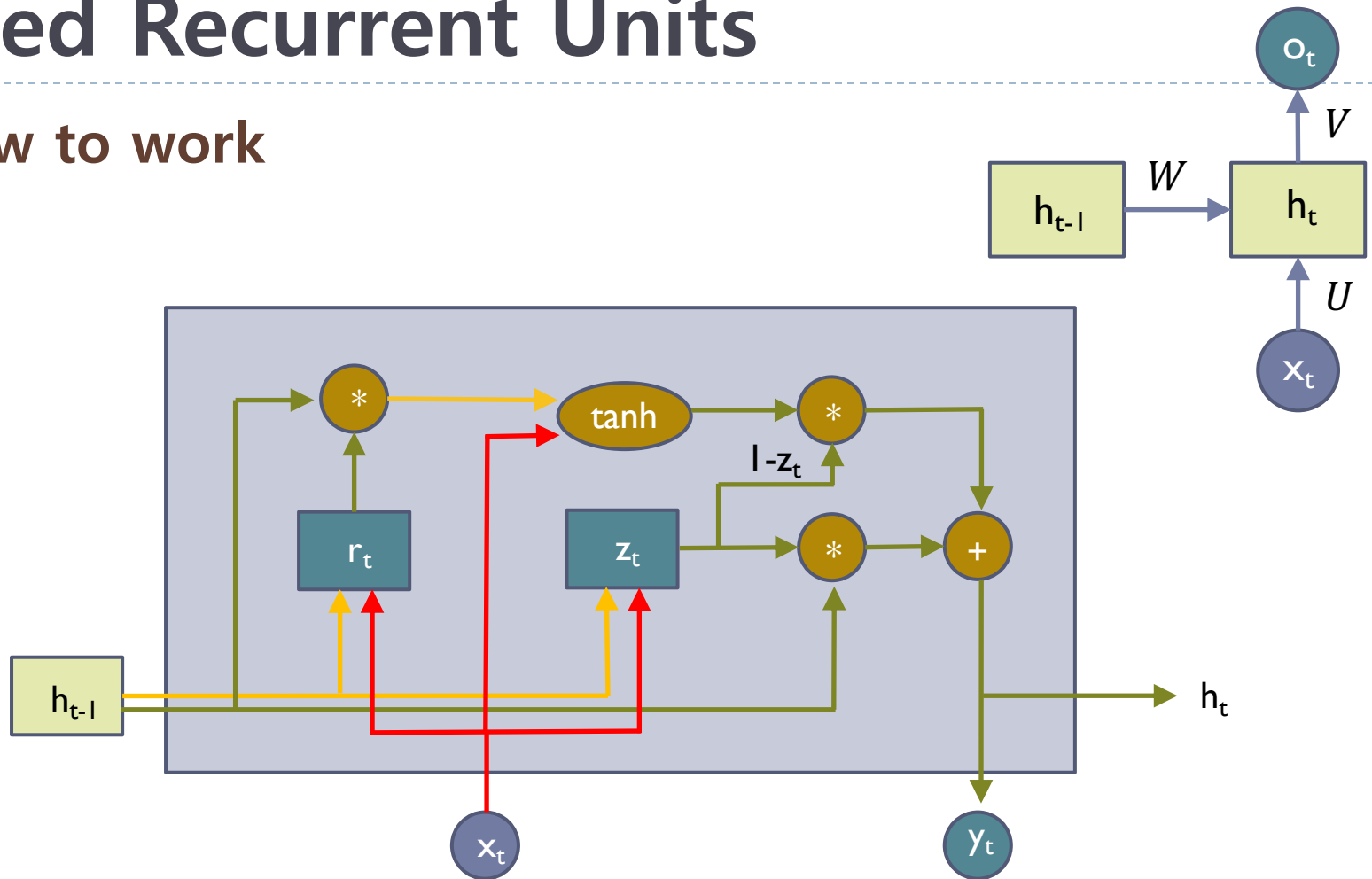
Gated Recurrent Units

► How to work



Gated Recurrent Units

► How to work



Question and Answer