Recurrent Neural Networks

성균관대학교 소프트웨어학과
이 지 형

Sequential Data Modeling

- **Sequential Data**
  - **Most of data are sequential**
  - **Speech, Text, Image, …**
- **Deep Learnings for Sequential Data**
  - **Convolutional Neural Networks (CNN)**
    - Try to find local features from a sequence
  - **Recurrent Neural Networks: LSTM, GRU**
    - Try to capture the feature of the past

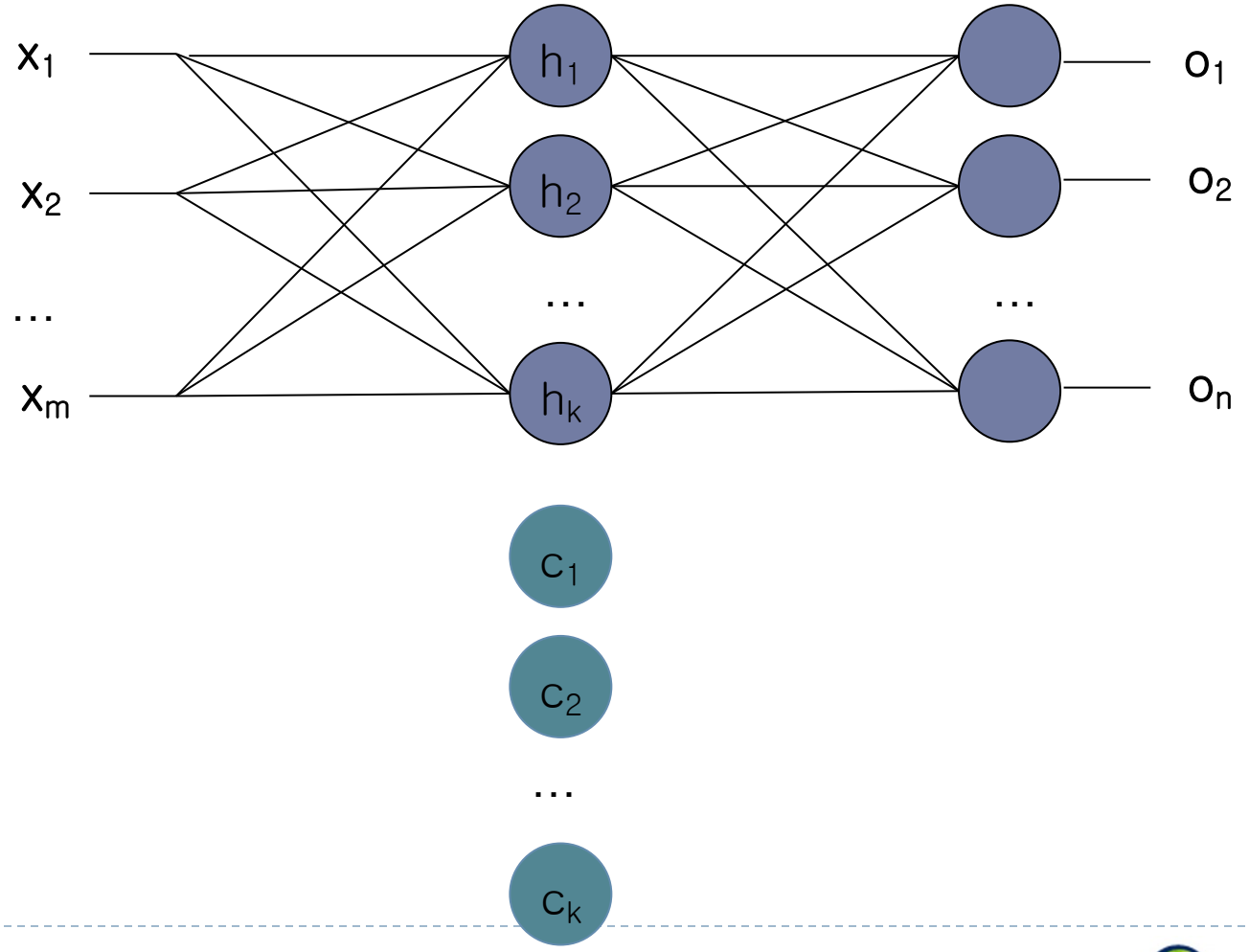
Sequential Data Processing

- What is sequential data?

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

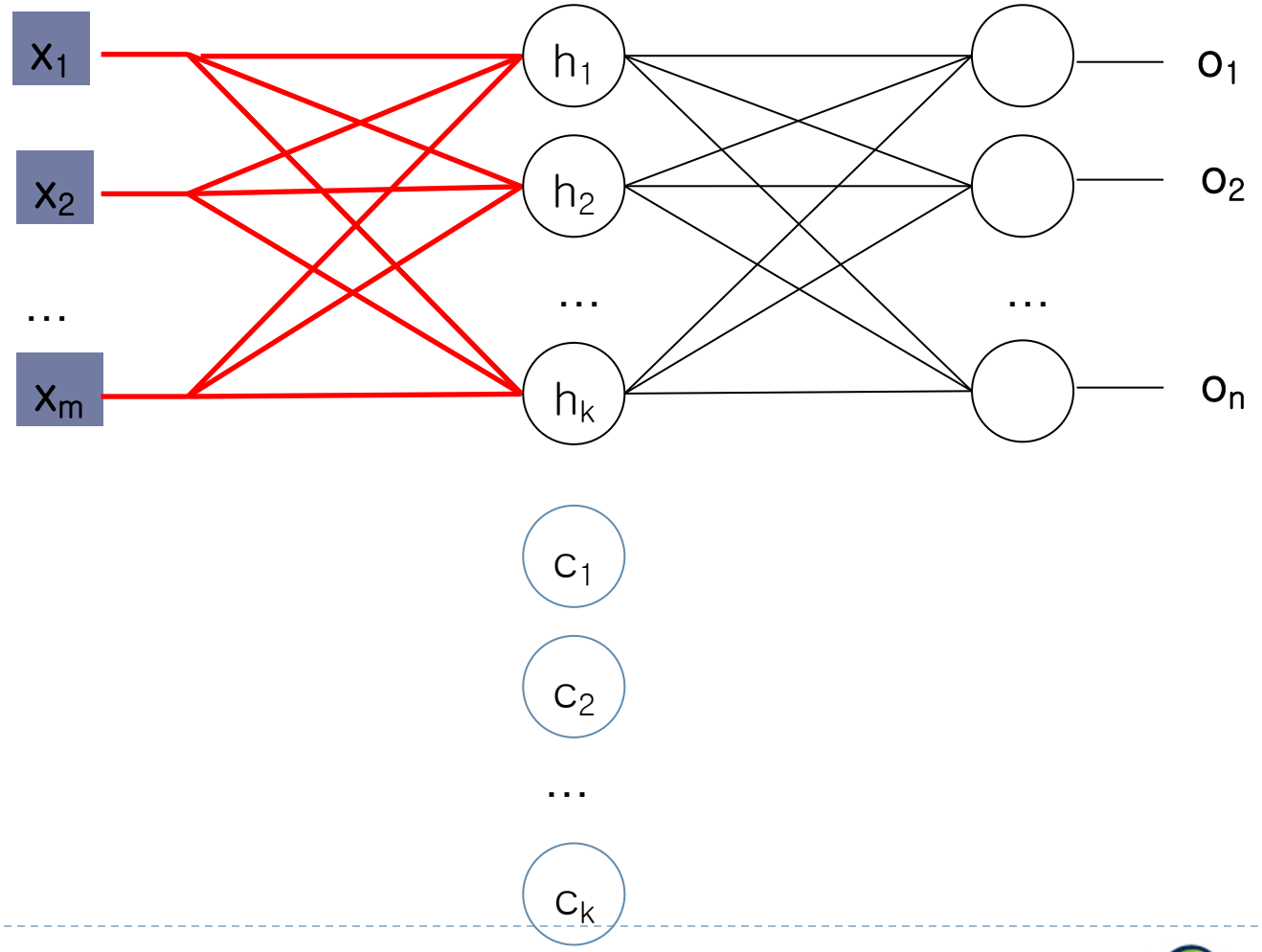
Recurrent Neural Networks

- **Connections form cycles**



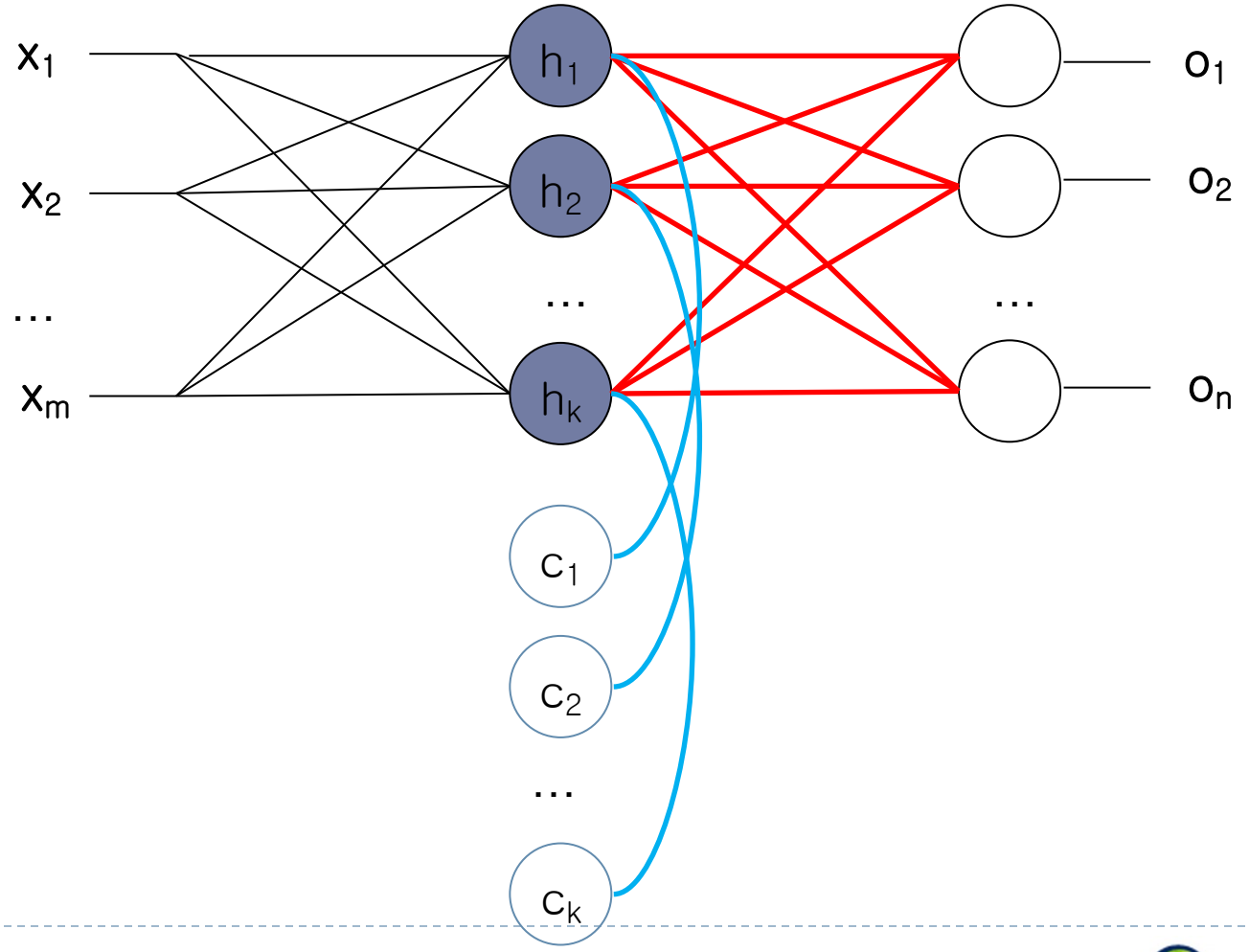
Recurrent Neural Networks

- Connections form cycles



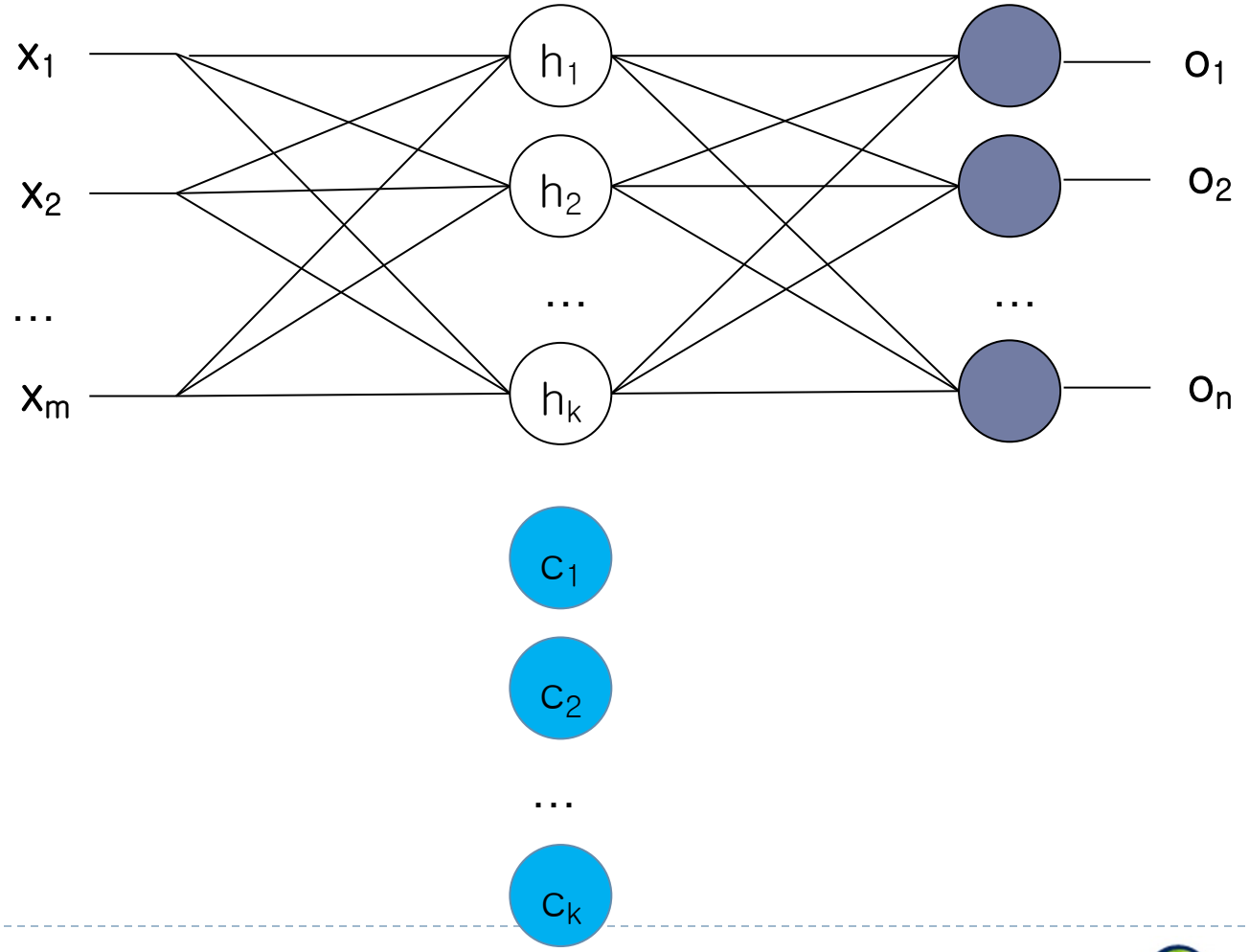
Recurrent Neural Networks

- Connections form cycles



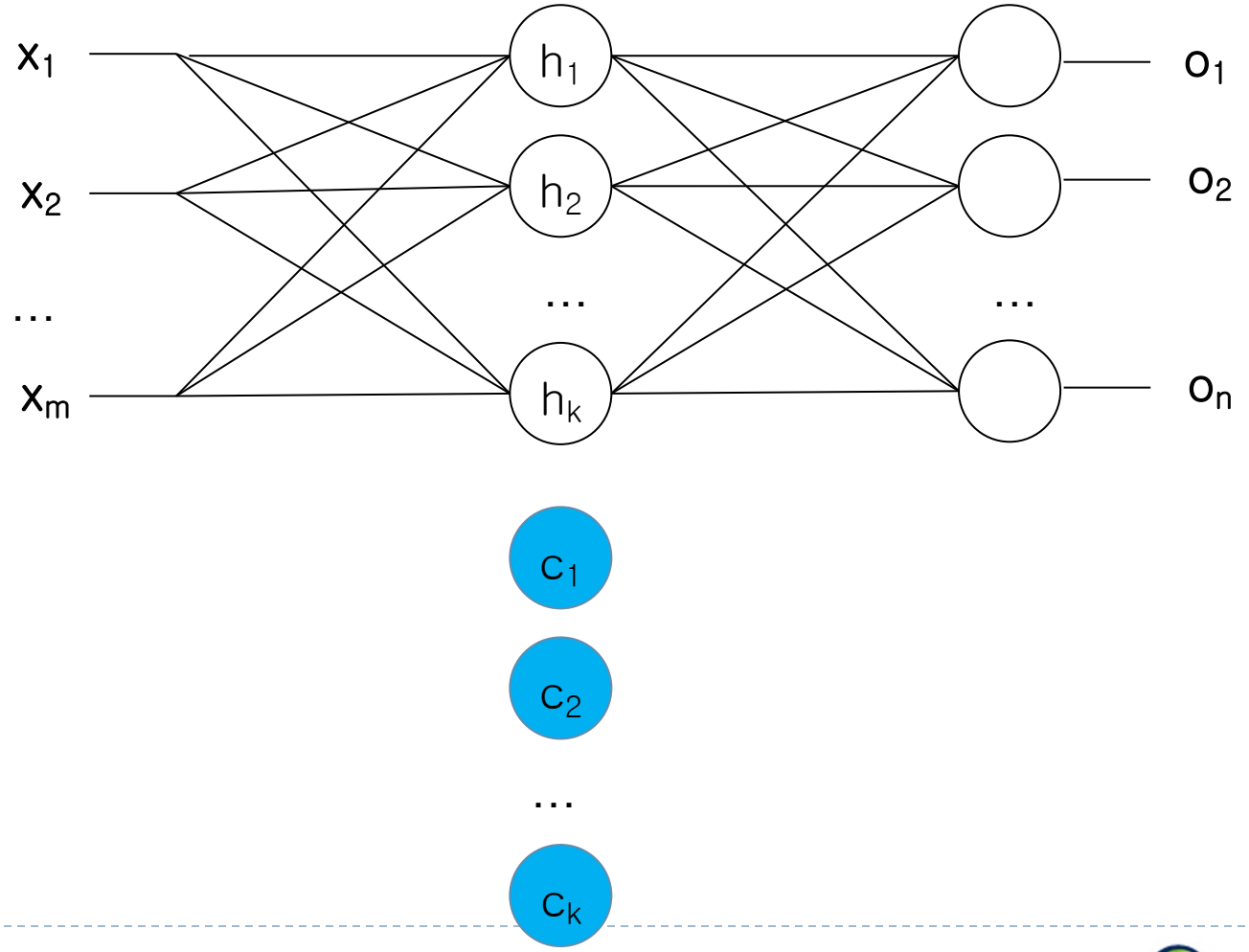
Recurrent Neural Networks

- **Connections form cycles**



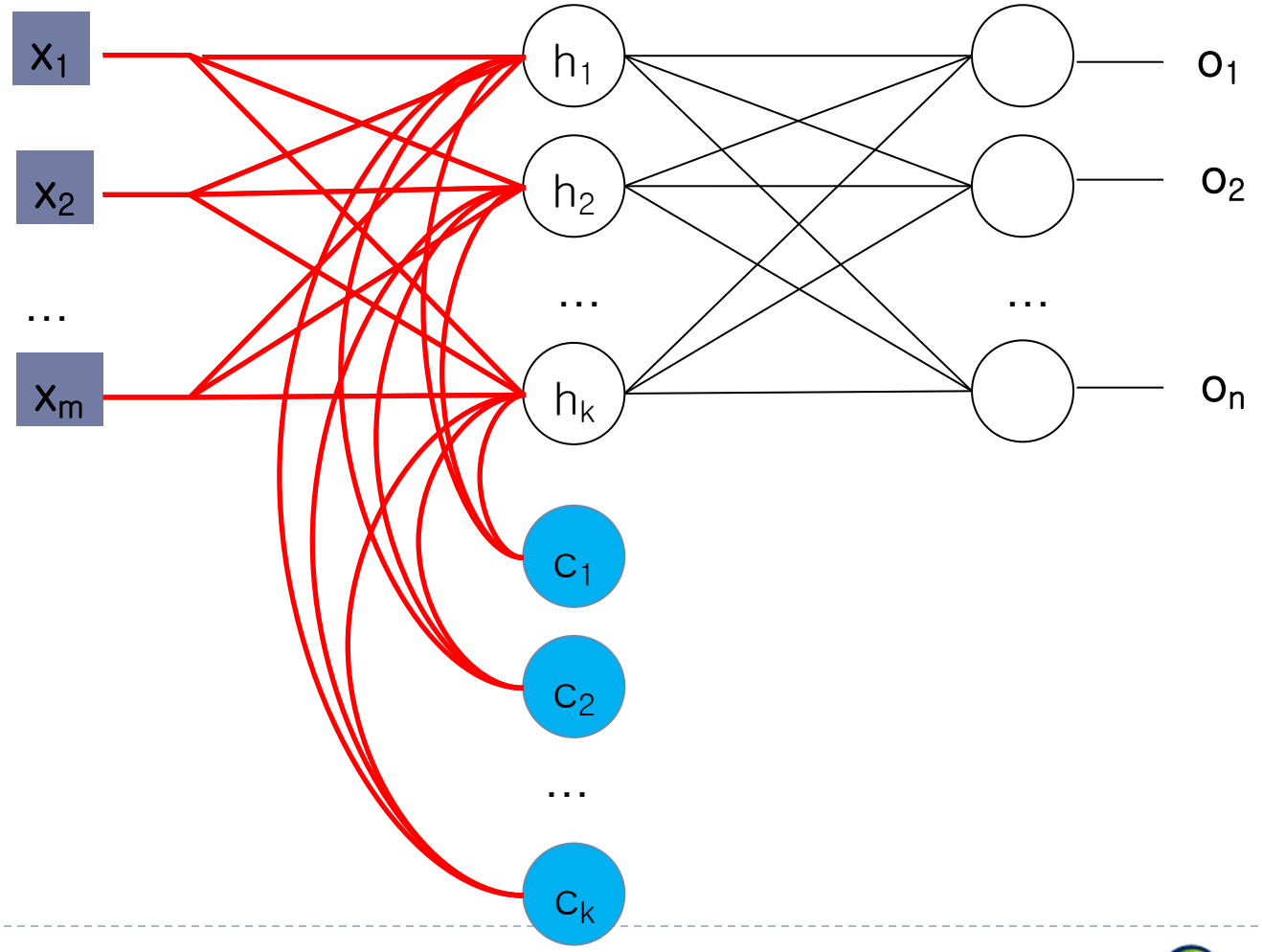
Recurrent Neural Networks

- **Connections form cycles**



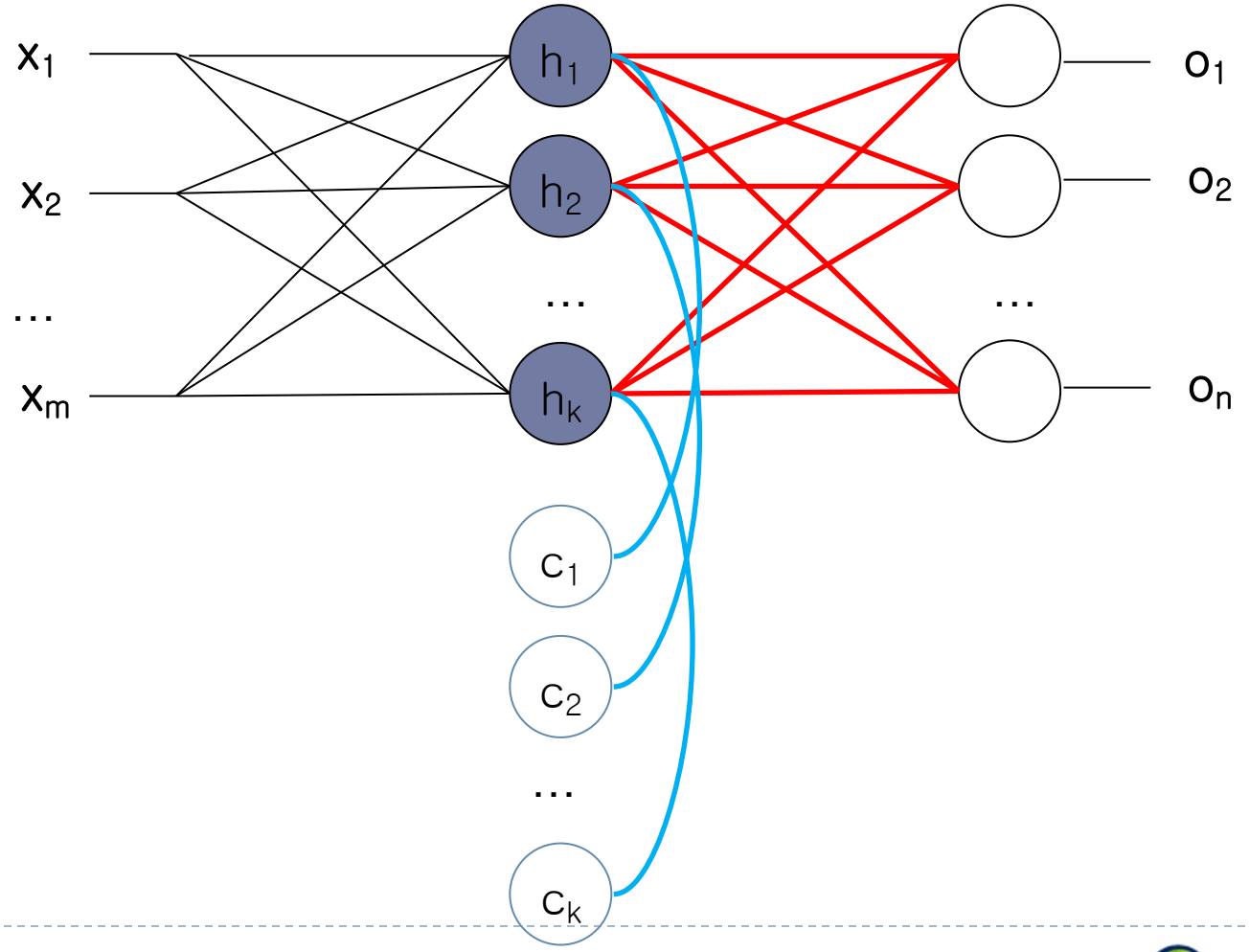
Recurrent Neural Networks

- Connections form cycles



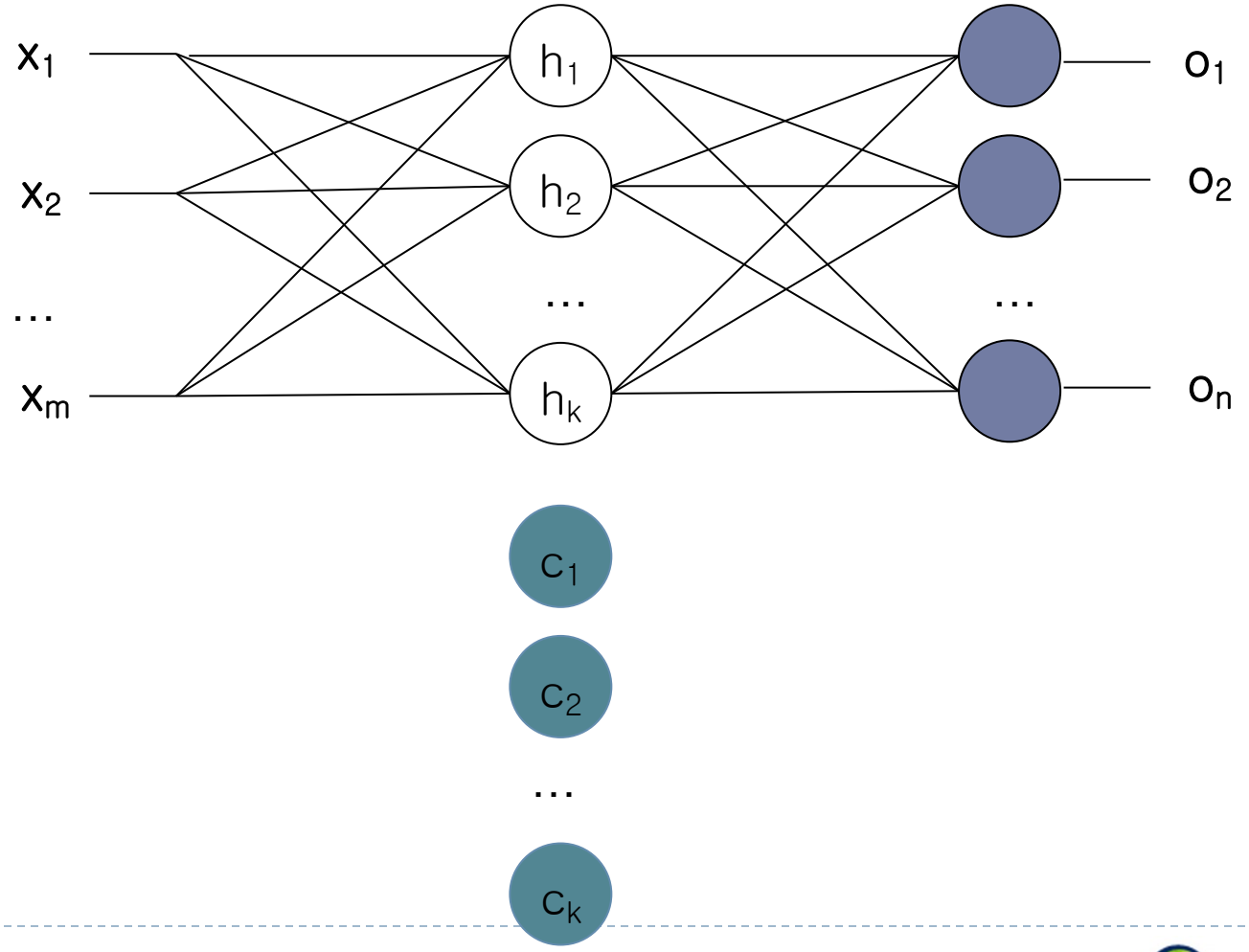
Recurrent Neural Networks

- Connections form cycles



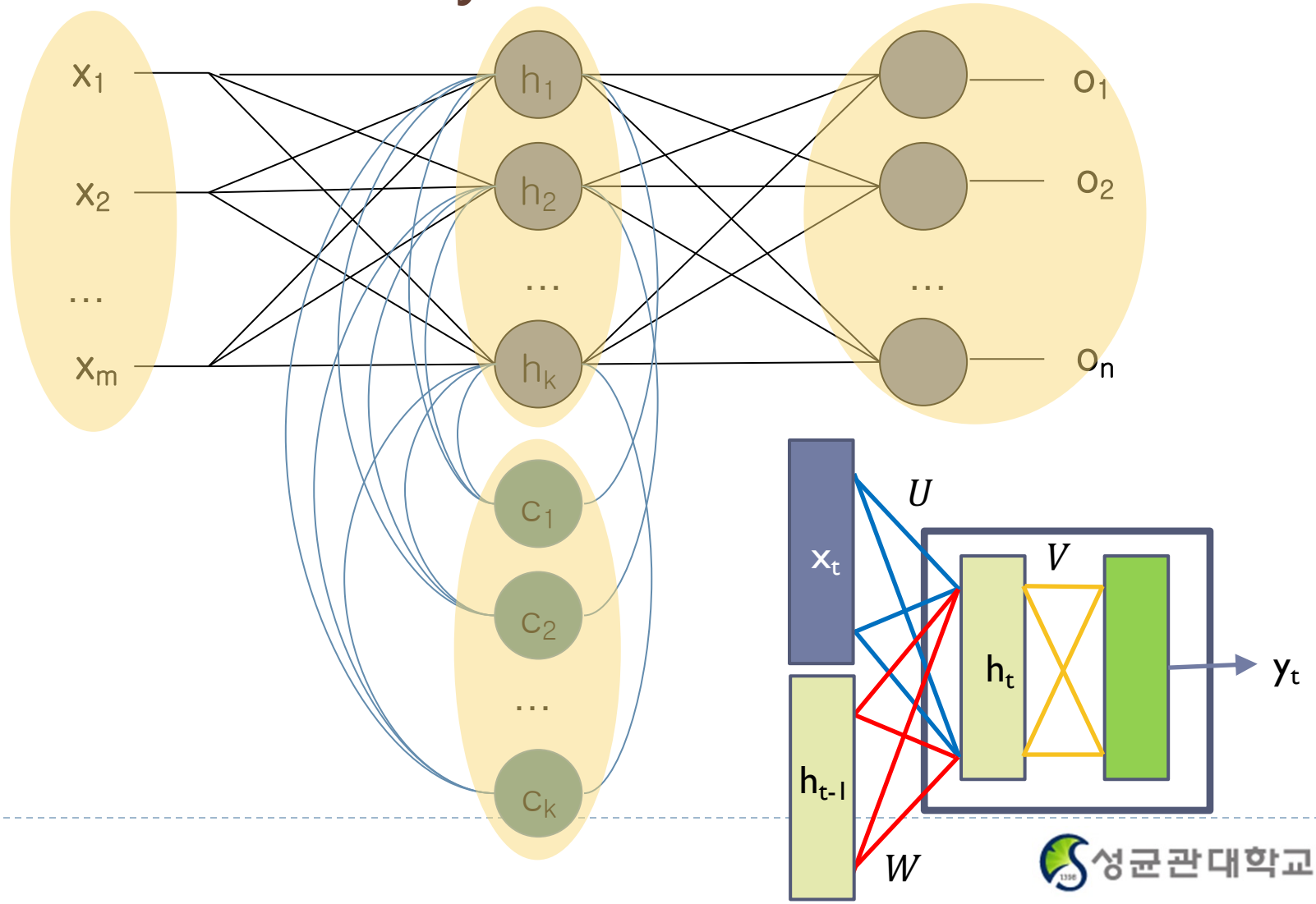
Recurrent Neural Networks

- **Connections form cycles**



Recurrent Neural Networks

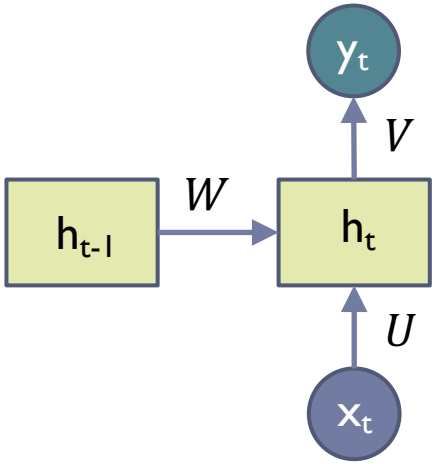
- Connections form cycles

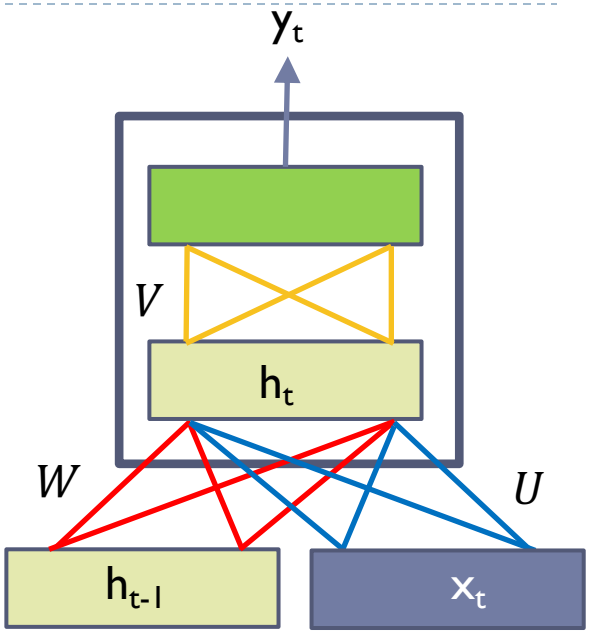


Recurrent Neural Networks

- **Connections form cycles**

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





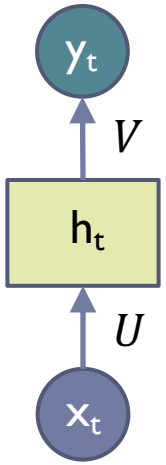
- x_t: input at time t
- h_t: hidden state at time t
- f: is an activation function
- U, V, W: network parameters
  - RNN shares the same parameters across all time steps
- g: activation function for the output layer

Recurrent Neural Networks

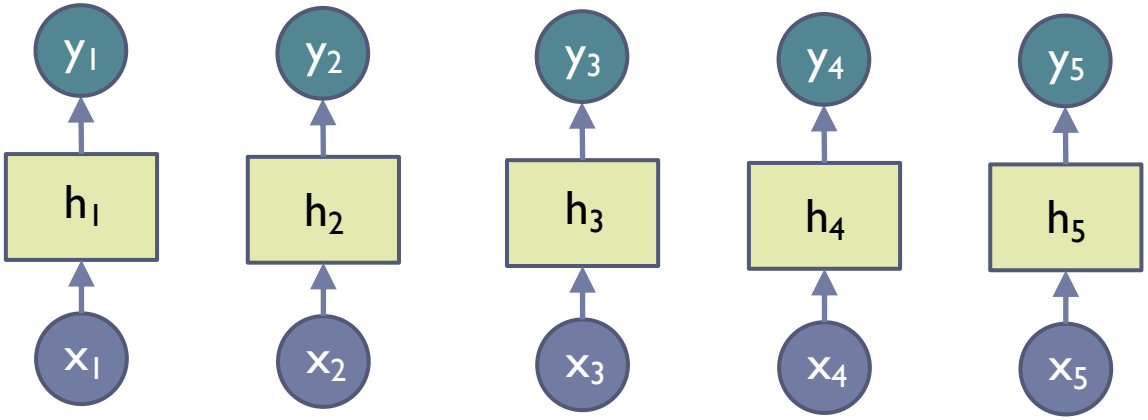
- **Recap: Regular NN**

Training Data

(x_1, y_1)
(x_2, y_2)
(x_3, y_3)
(x_4, y_4)
(x_5, y_5)



$$h_t = f(Ux_t)$$
$$y_t = g(Vh_t)$$

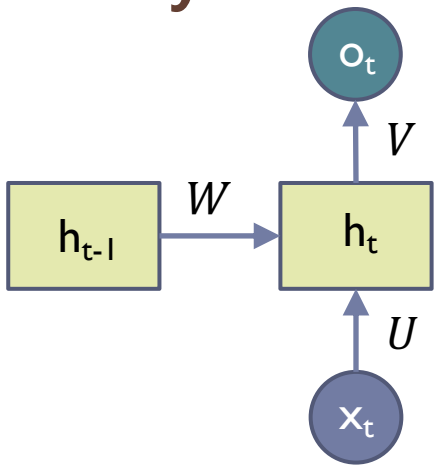


Recurrent Neural Networks

- **Connections form cycles**

Training Data

(x_1, y_1)
(x_2, y_2)
(x_3, y_3)
(x_4, y_4)
(x_5, y_5)



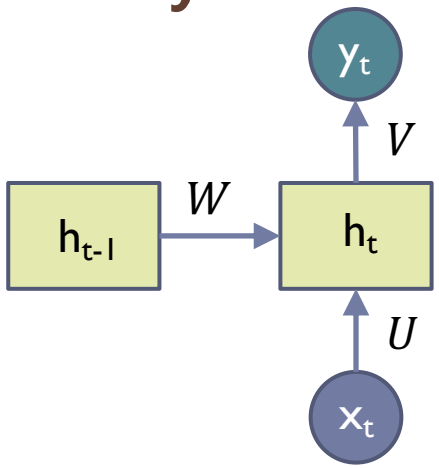
$$h_t = f(Ux_t + Wh_{t-1})$$
$$o_t = g(Vh_t)$$

Recurrent Neural Networks

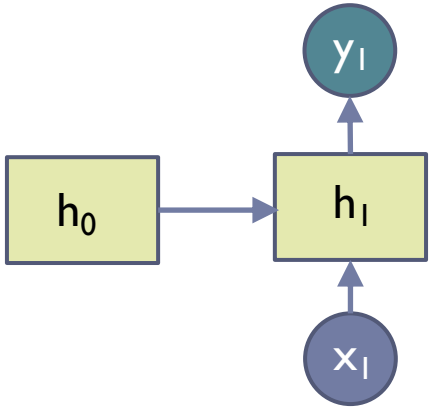
- Connections form cycles

Training Data

(x_1, y_1)
(x_2, y_2)
(x_3, y_3)
(x_4, y_4)
(x_5, y_5)



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



Recurrent Neural Networks

- **Connections form cycles**

Training Data

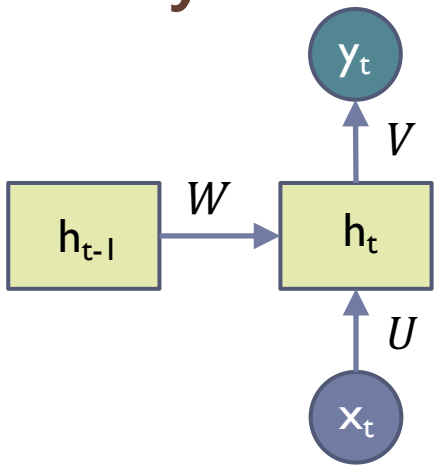
(x_1, y_1)

(x_2, y_2)

(x_3, y_3)

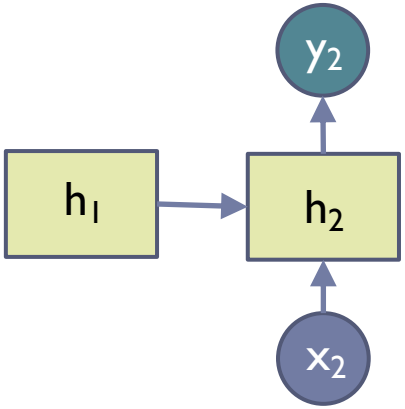
(x_4, y_4)

(x_5, y_5)



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

$$y_t = g(Vh_t)$$

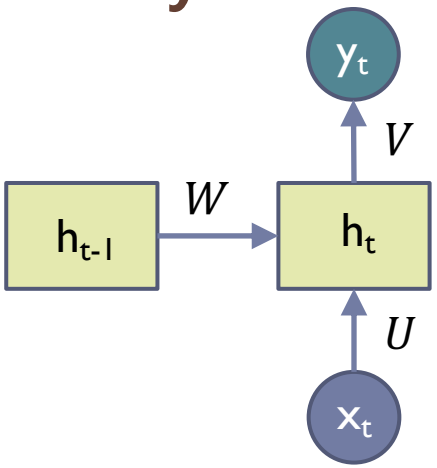


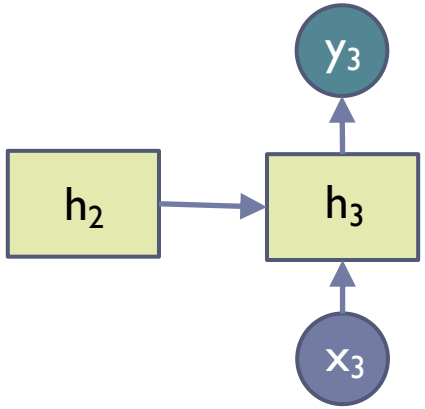
Recurrent Neural Networks

- **Connections form cycles**

Training Data

(x_1, y_1)
(x_2, y_2)
(x_3, y_3)
(x_4, y_4)
(x_5, y_5)





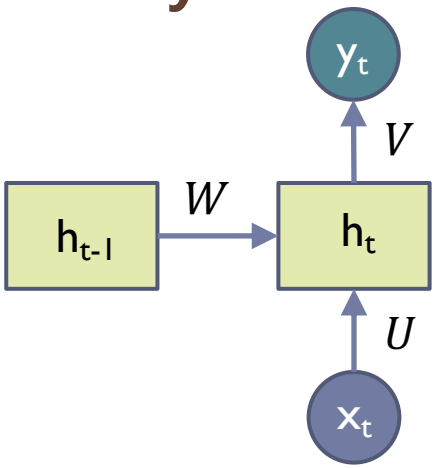
$$h_t = f(Ux_t + Wh_{t-1})$$
$$y_t = g(Vh_t)$$

Recurrent Neural Networks

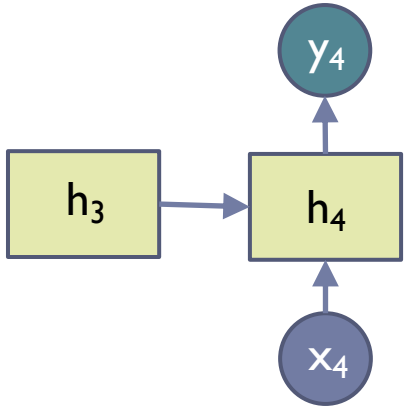
- **Connections form cycles**

Training Data

(x_1, y_1)
(x_2, y_2)
(x_3, y_3)
(x_4, y_4)
(x_5, y_5)



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

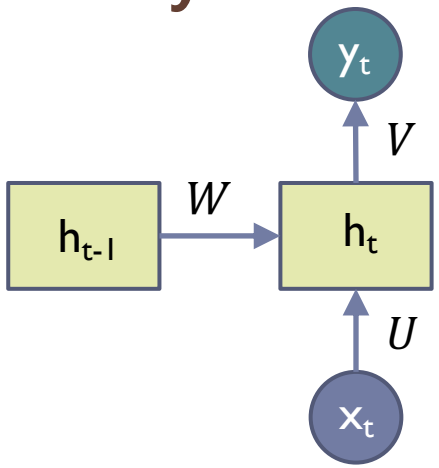


Recurrent Neural Networks

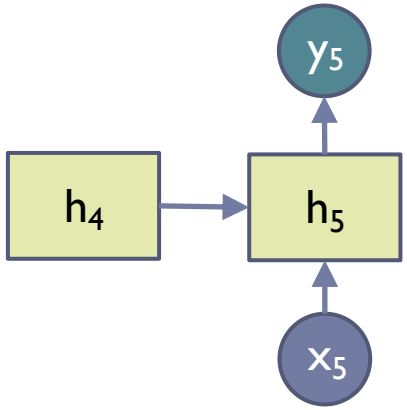
- **Connections form cycles**

Training Data

(x_1, y_1)
(x_2, y_2)
(x_3, y_3)
(x_4, y_4)
(x_5, y_5)



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

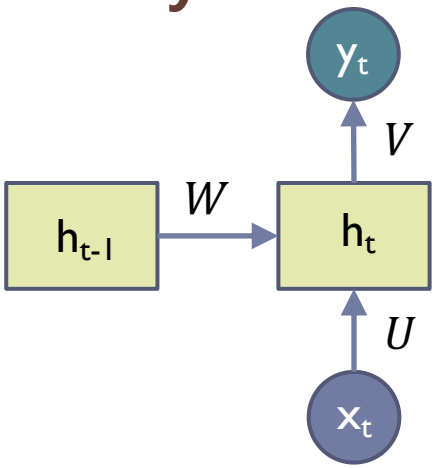


Recurrent Neural Networks

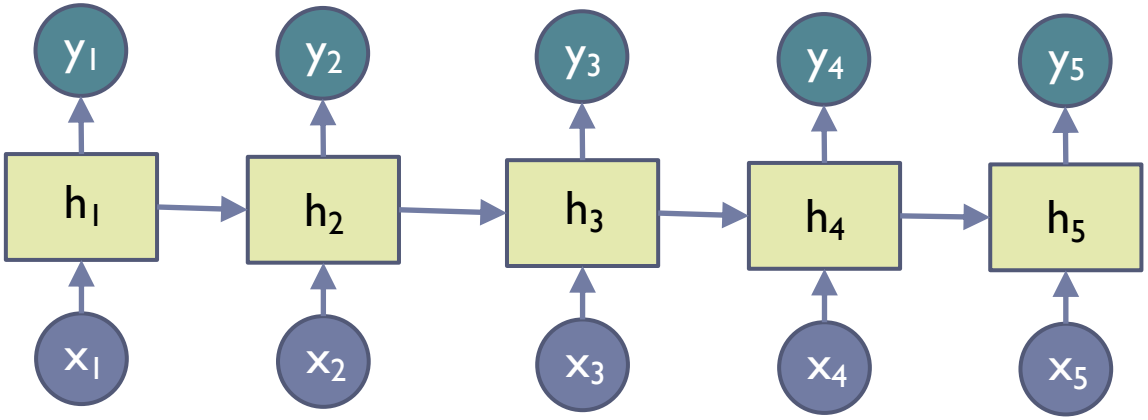
- **Connections form cycles**

Training Data

(x_1, y_1)
(x_2, y_2)
(x_3, y_3)
(x_4, y_4)
(x_5, y_5)

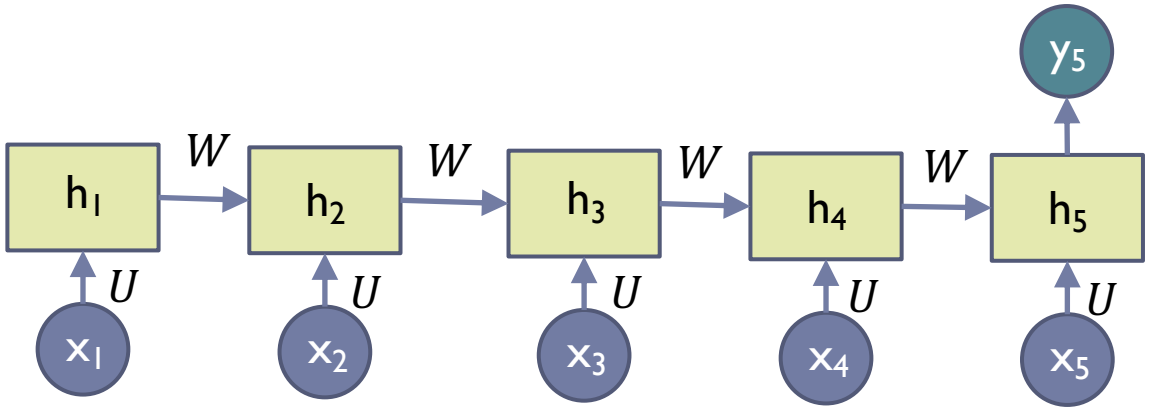


$$h_t = f(Ux_t + Wh_{t-1})$$
$$y_t = g(Vh_t)$$



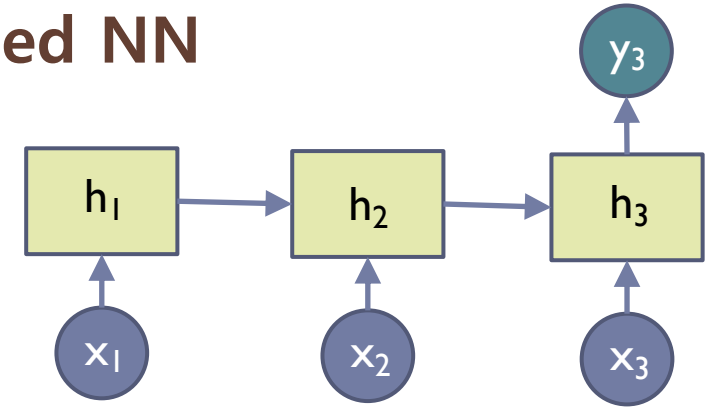
Recurrent Neural Networks

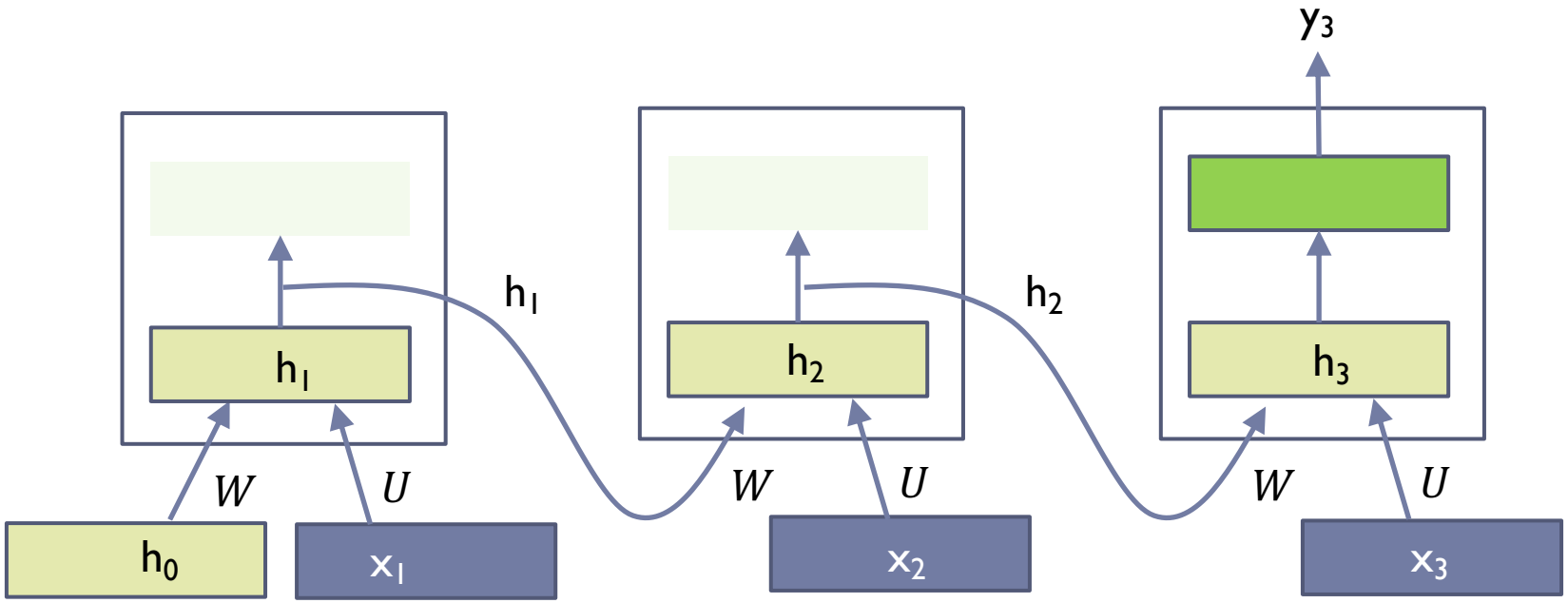
- **Long Term Dependency**
  - $x_1 \sim x_{t-1}$ are encoded into h_{t-1}
  - h_{t-1} has the information on the past
  - It is a context to process x_t



Recurrent Neural Networks

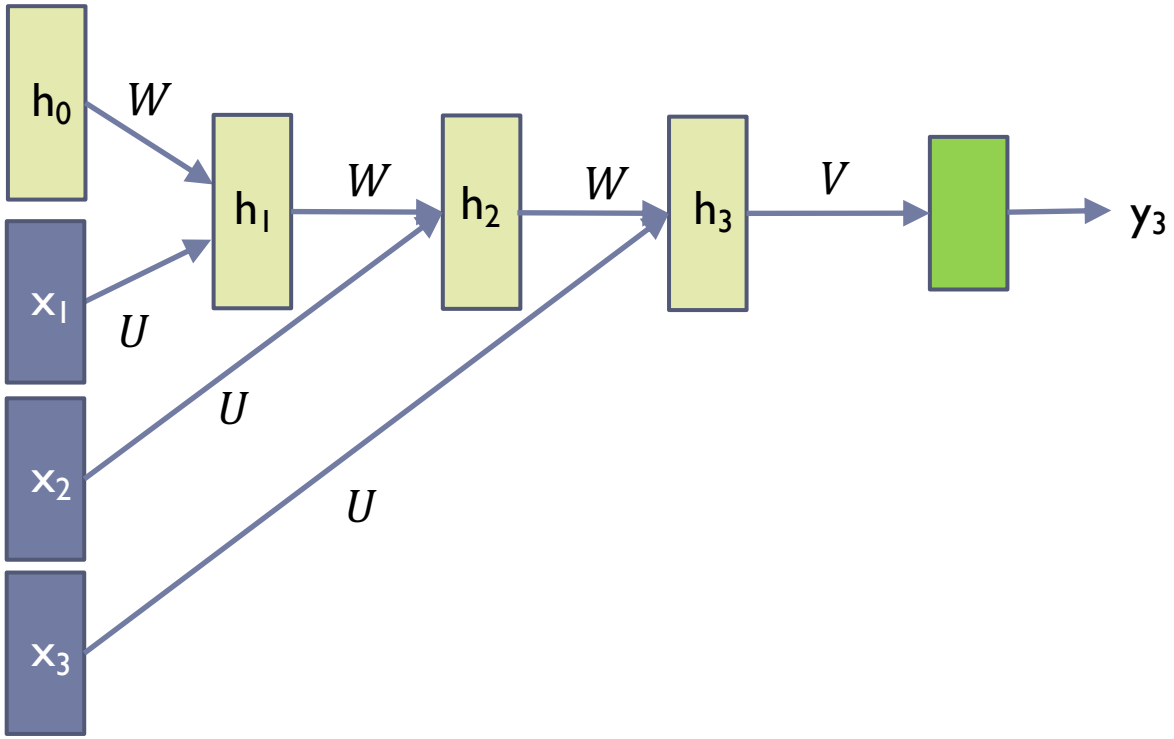
- **Fully-Connected NN**





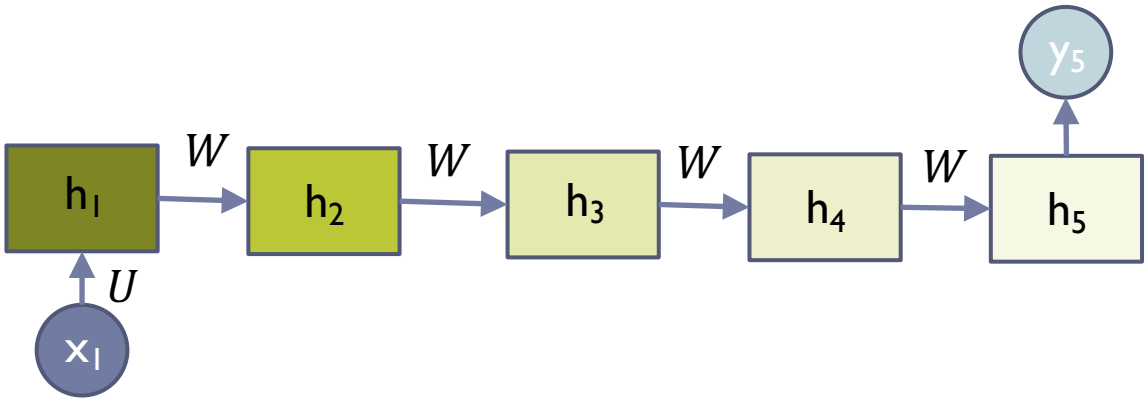
Recurrent Neural Networks

- **Fully-Connected NN**



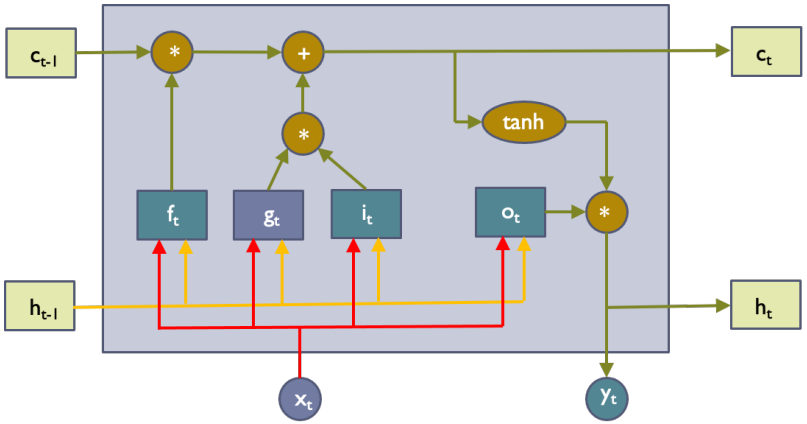
Recurrent Neural Networks

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



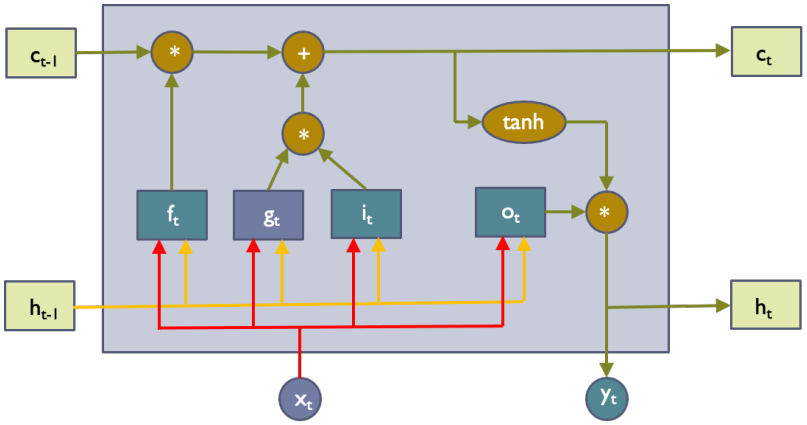
Long Short-Term Memory (LSTM)

- **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



$$i = \sigma(x_t U^i + h_{t-1} W^i)$$

$$f = \sigma(x_t U^f + h_{t-1} W^f)$$

$$o = \sigma(x_t U^o + h_{t-1} W^o)$$

$$g = \tanh(x_t U^g + h_{t-1} W^g)$$

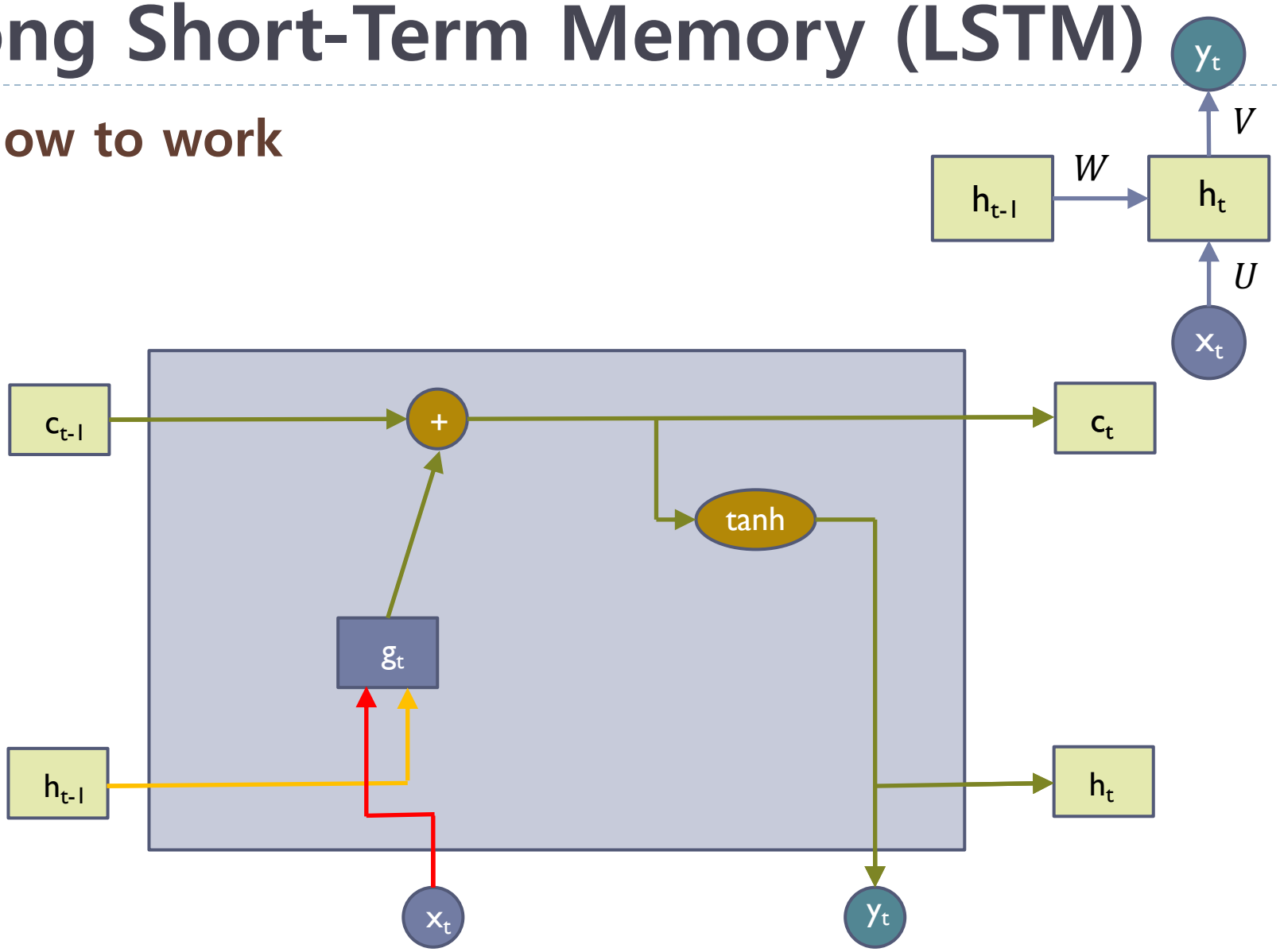
$$c_t = c_{t-1} \circ f + g \circ i$$

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

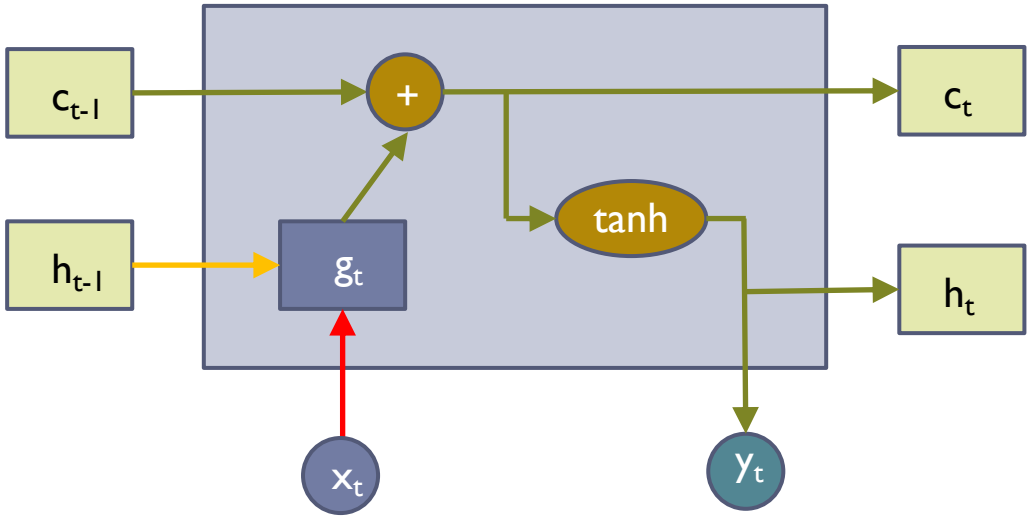
$$y = softmax(V h_t)$$

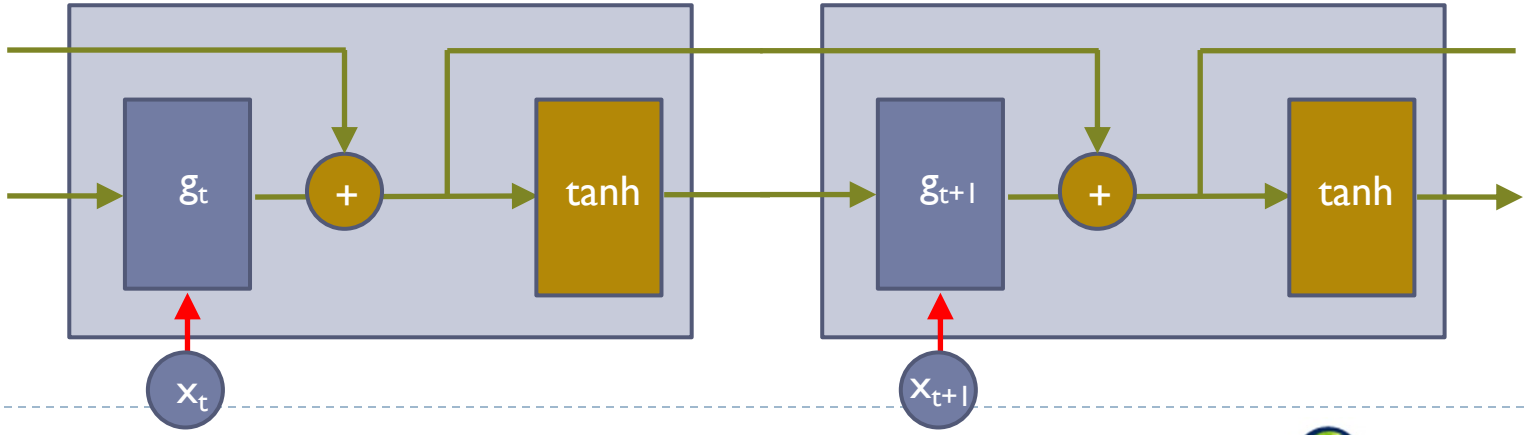
Long Short-Term Memory (LSTM)

- How to work



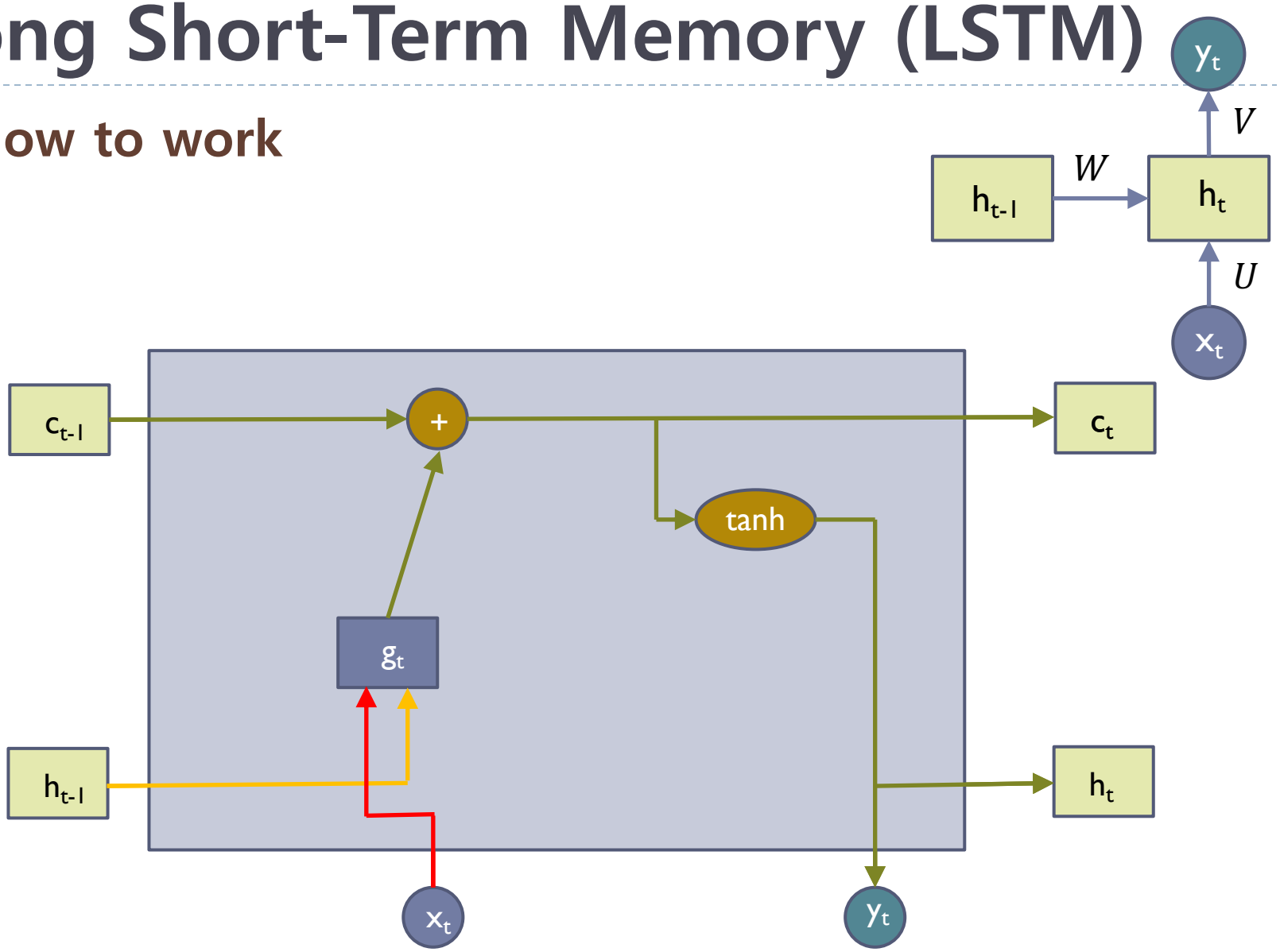
Long Short-Term Memory (LSTM)





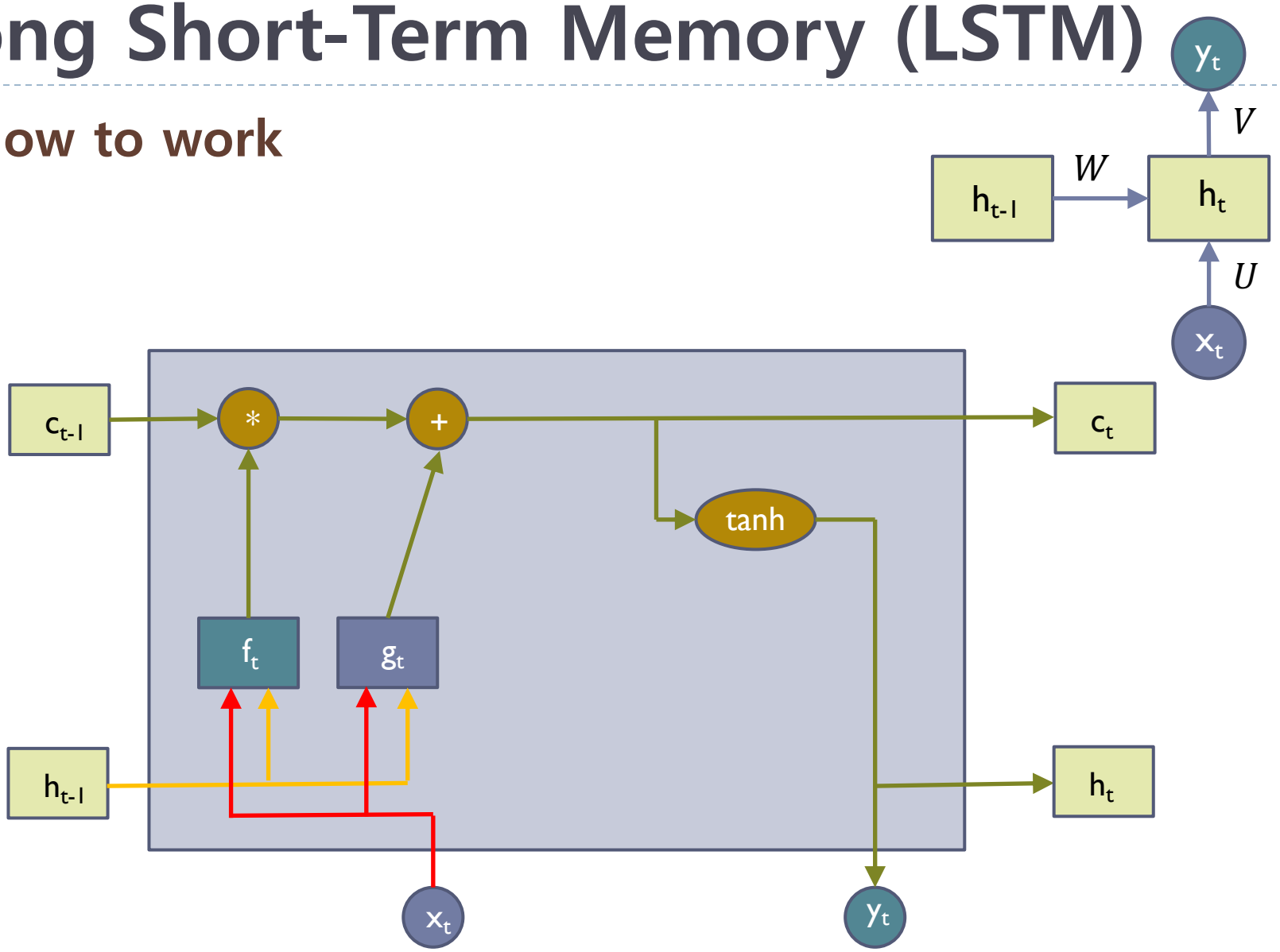
Long Short-Term Memory (LSTM)

- How to work



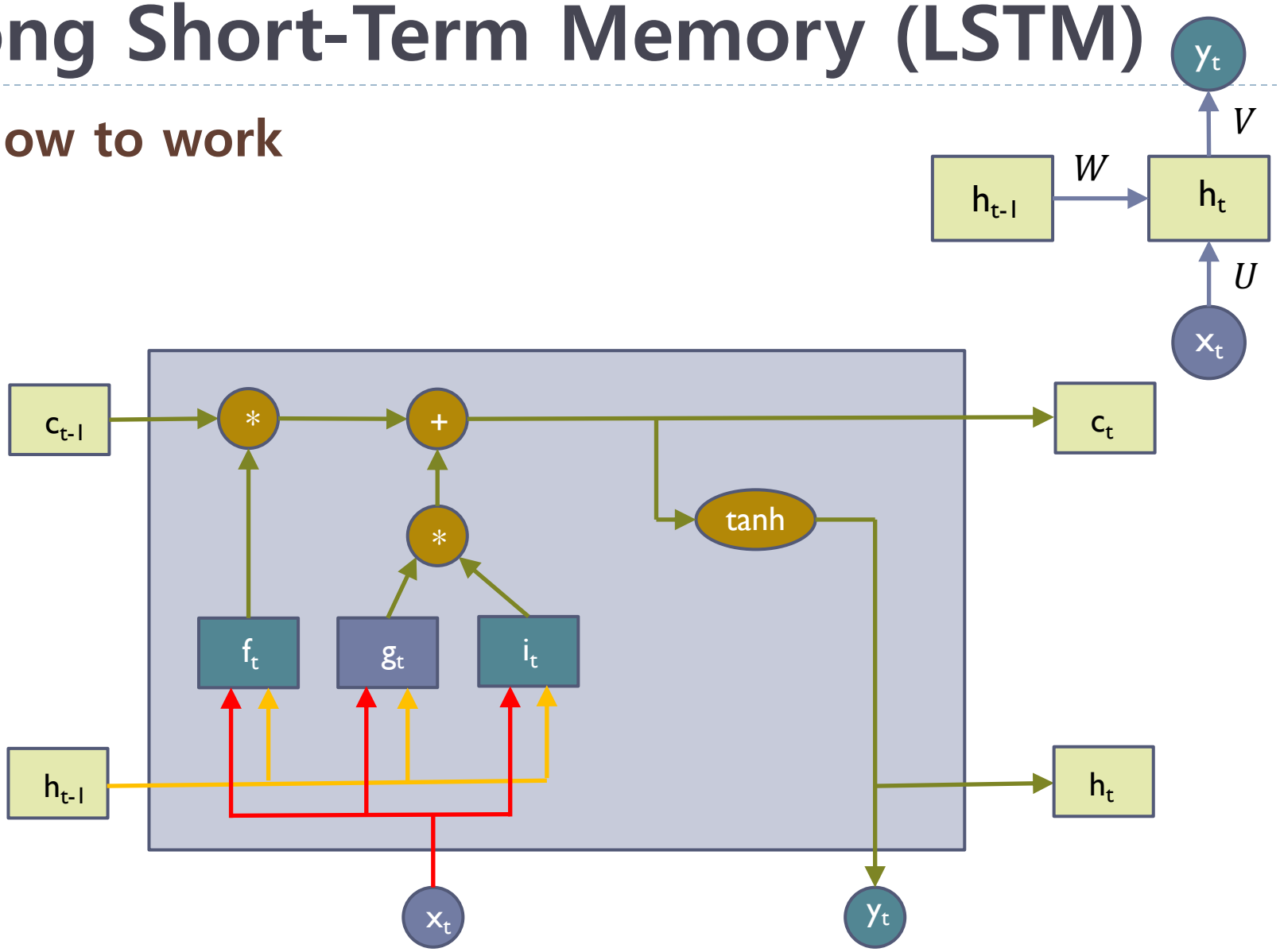
Long Short-Term Memory (LSTM)

- How to work



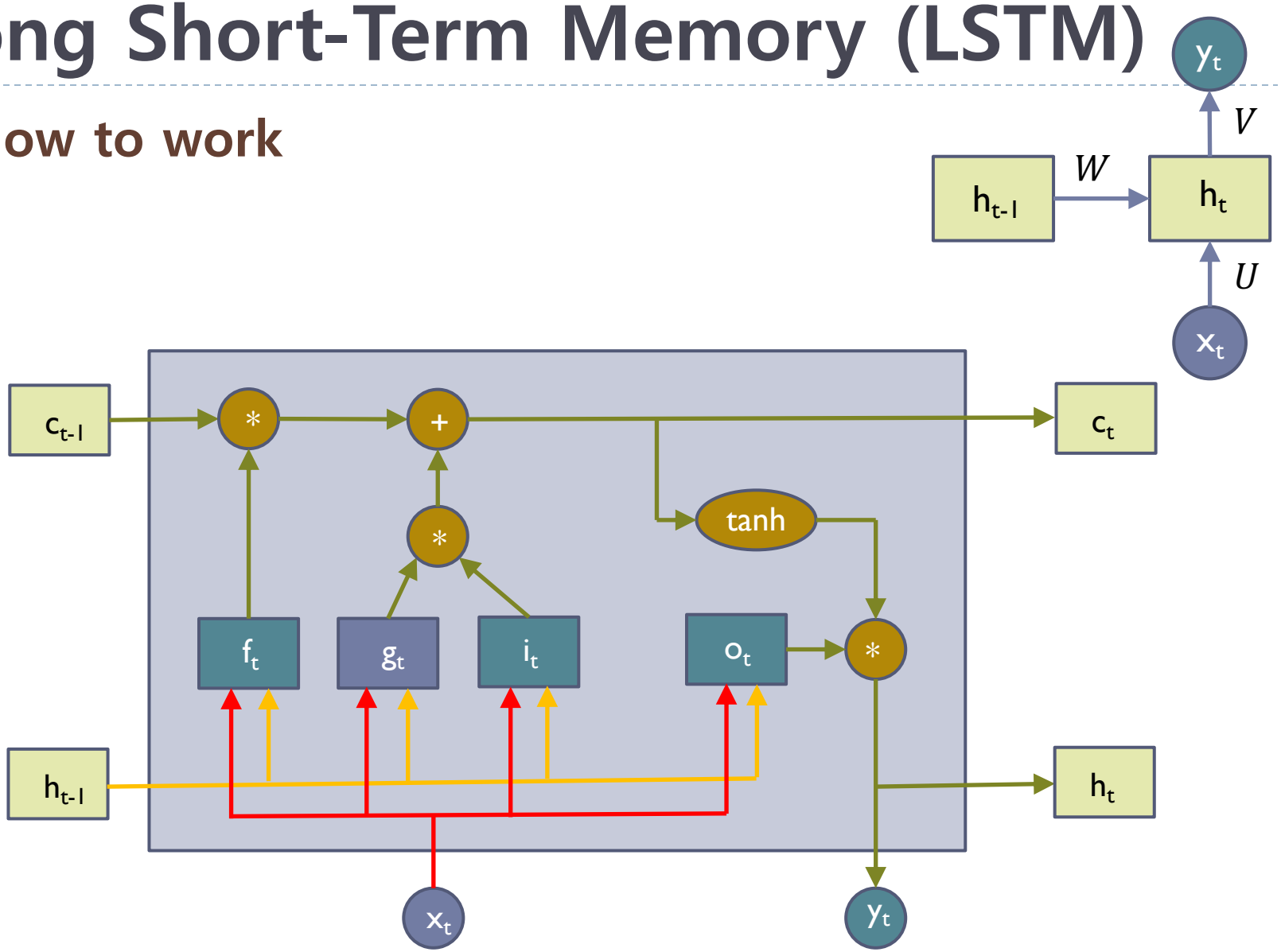
Long Short-Term Memory (LSTM)

- How to work



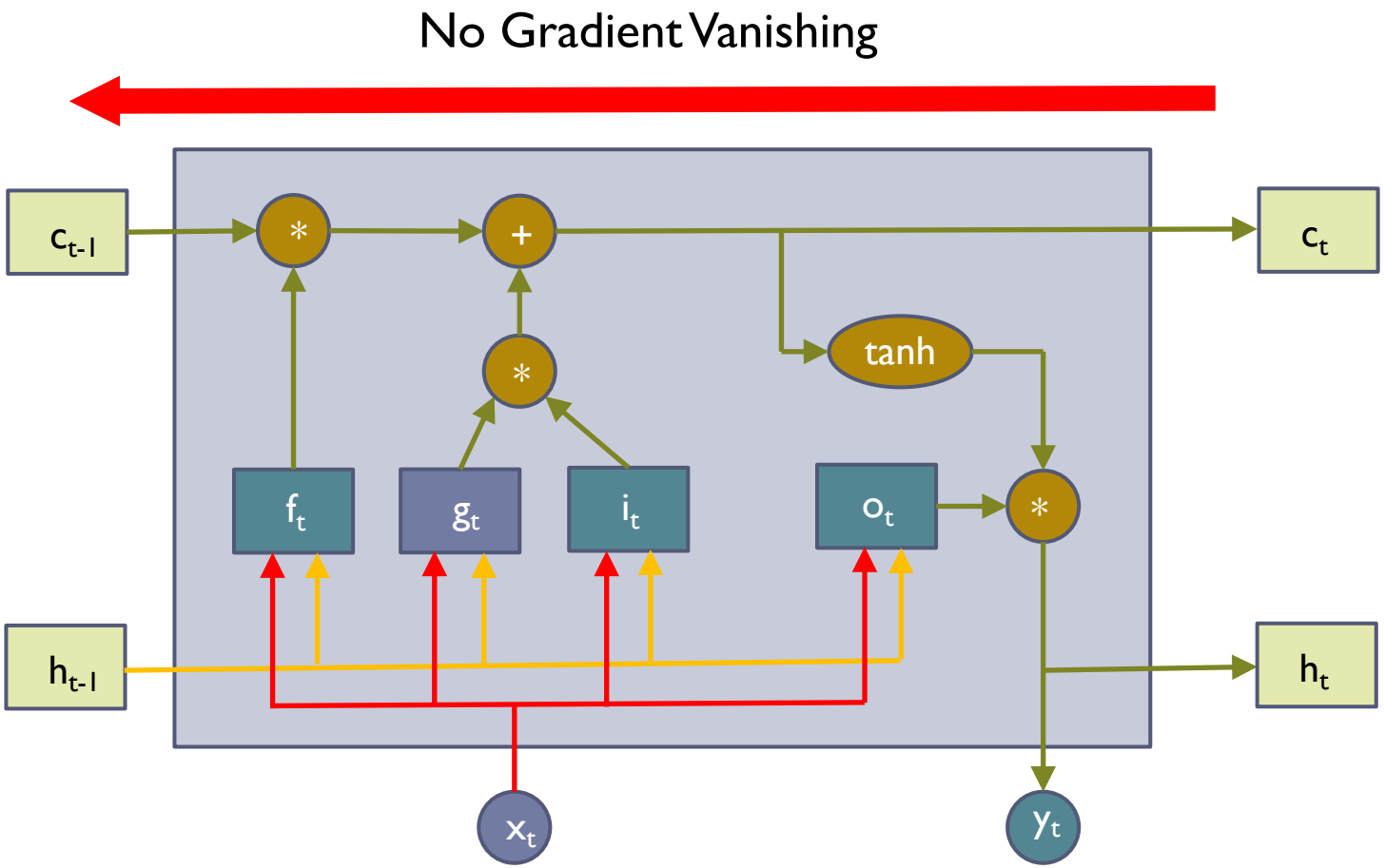
Long Short-Term Memory (LSTM)

- How to work



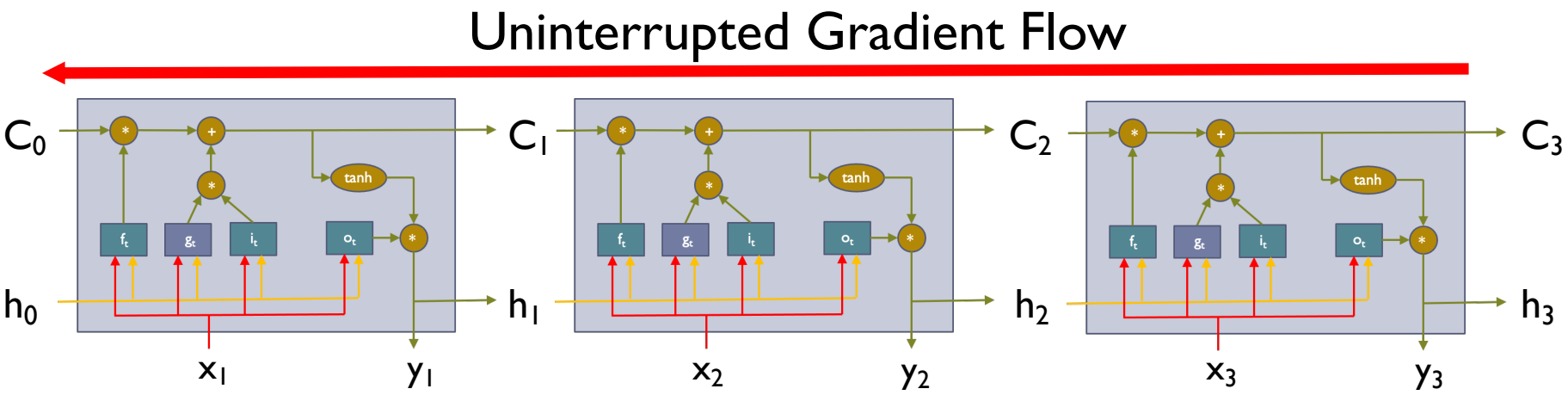
Long Short-Term Memory (LSTM)

- **Gradient Flow**

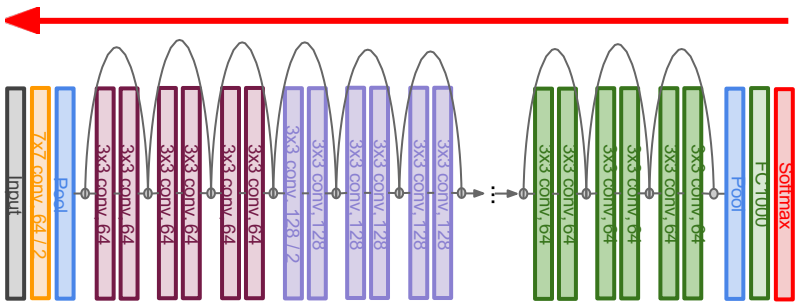


Long Short-Term Memory (LSTM)

- **Gradient Flow**



Similar to ResNet!



Sequence Processing

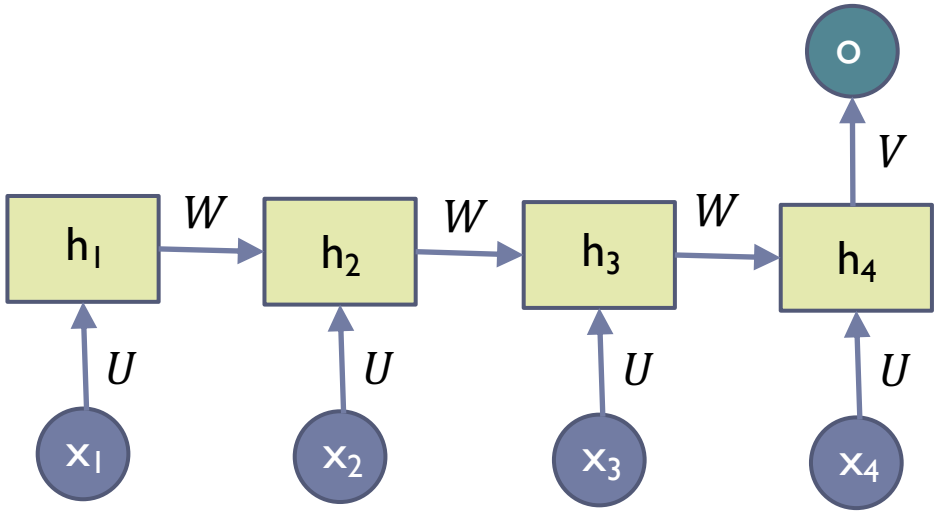
- **Training Data**
  - **Usually, samples are preprocessed in a fixed length**

$$
\begin{array}{l}
(x_1, y_1) \\
(x_2, y_2) \\
(x_3, y_3) \\
(x_4, y_4) \\
(x_5, y_5) \\
(x_6, y_6) \\
(x_7, y_7) \\
(x_8, y_8) \\
\ldots
\end{array}
\quad \Rightarrow \quad
\begin{array}{l}
(x_1 x_2 x_3 x_4, y_4) \\
(x_2 x_3 x_4 x_5, y_5) \\
(x_3 x_4 x_5 x_6, y_6) \\
(x_4 x_5 x_6 x_7, y_7) \\
(x_5 x_6 x_7 x_8, y_8) \\
\ldots
\end{array}
$$

Sequence Processing

- **Training**
  - Samples are trained with a fixed length of RNN

$(x_1 x_2 x_3 x_4, y_4)$
$(x_2 x_3 x_4 x_5, y_5)$
$(x_3 x_4 x_5 x_6, y_6)$
$(x_4 x_5 x_6 x_7, y_7)$
$(x_5 x_6 x_7 x_8, y_8)$
…

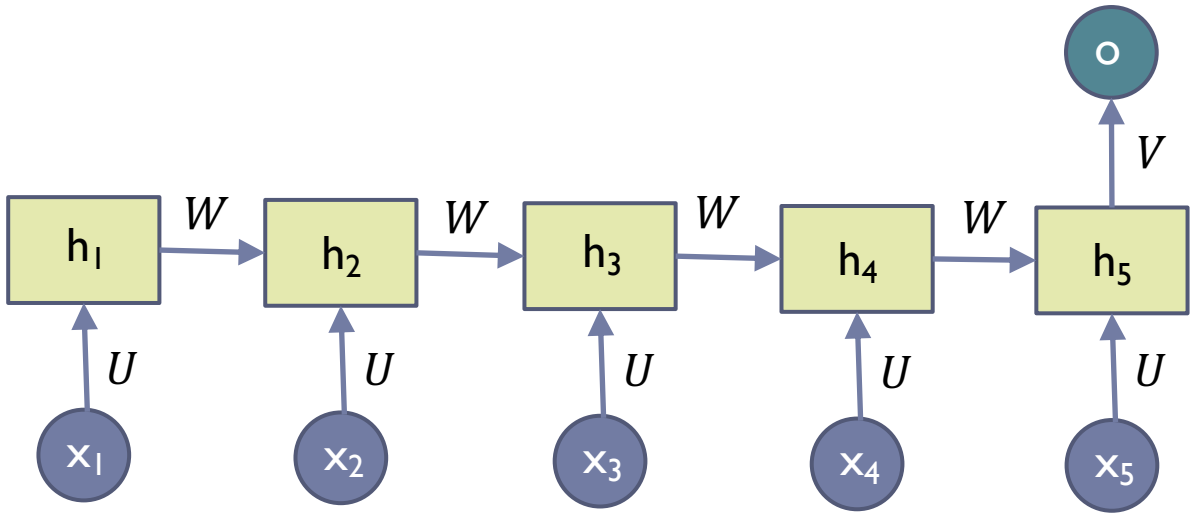


$$E = (y - o)^2$$

Sequence Processing

- **Training**

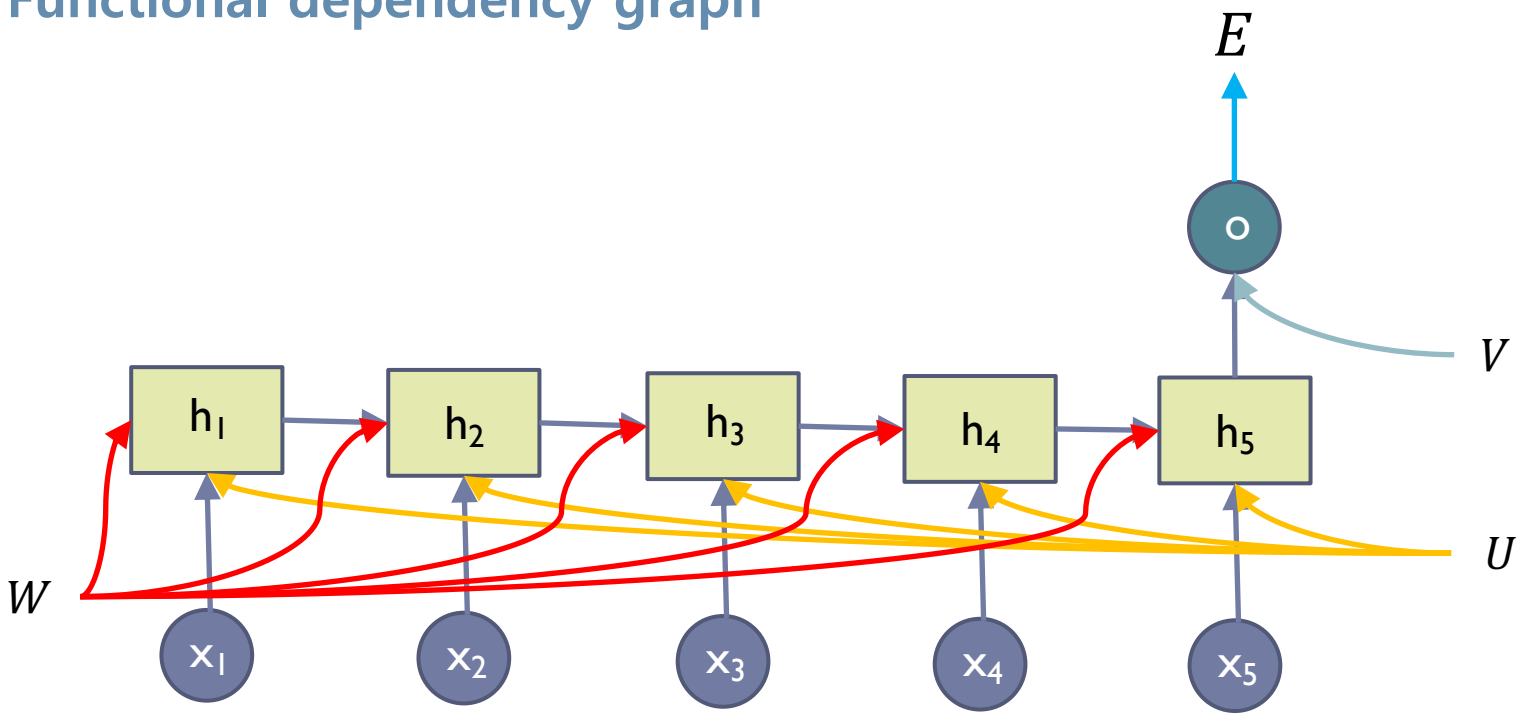
$$x_1 x_2 x_3 \cdots x_n \rightarrow y$$



$$E = (y - o)^2$$

Sequence Processing

- Training
  - Functional dependency graph

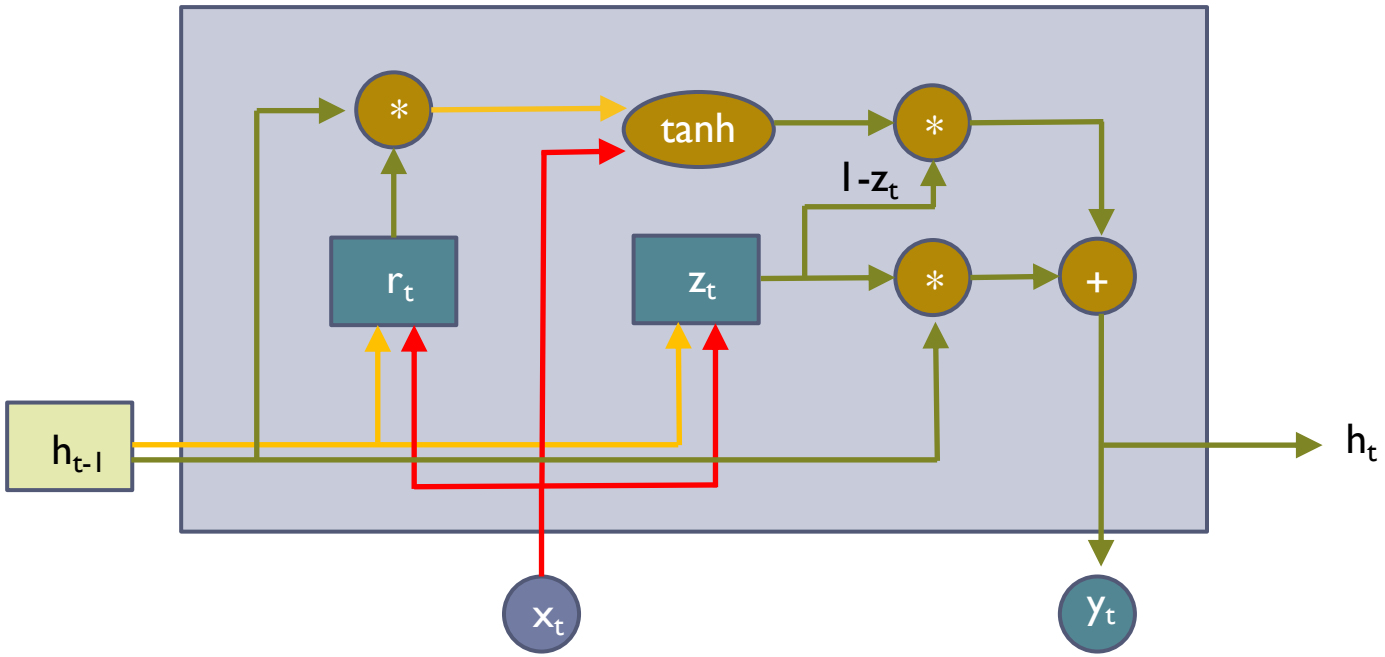


$$\frac{\partial E}{\partial w} = \sum_{i=1}^{n} \frac{\partial E}{\partial h_i} \frac{\partial h_i}{\partial w} \qquad \frac{\partial E}{\partial h_i} = \frac{\partial E}{\partial h_{i+1}} \frac{\partial h_{i+1}}{\partial h_i}$$

Gated Recurrent Units

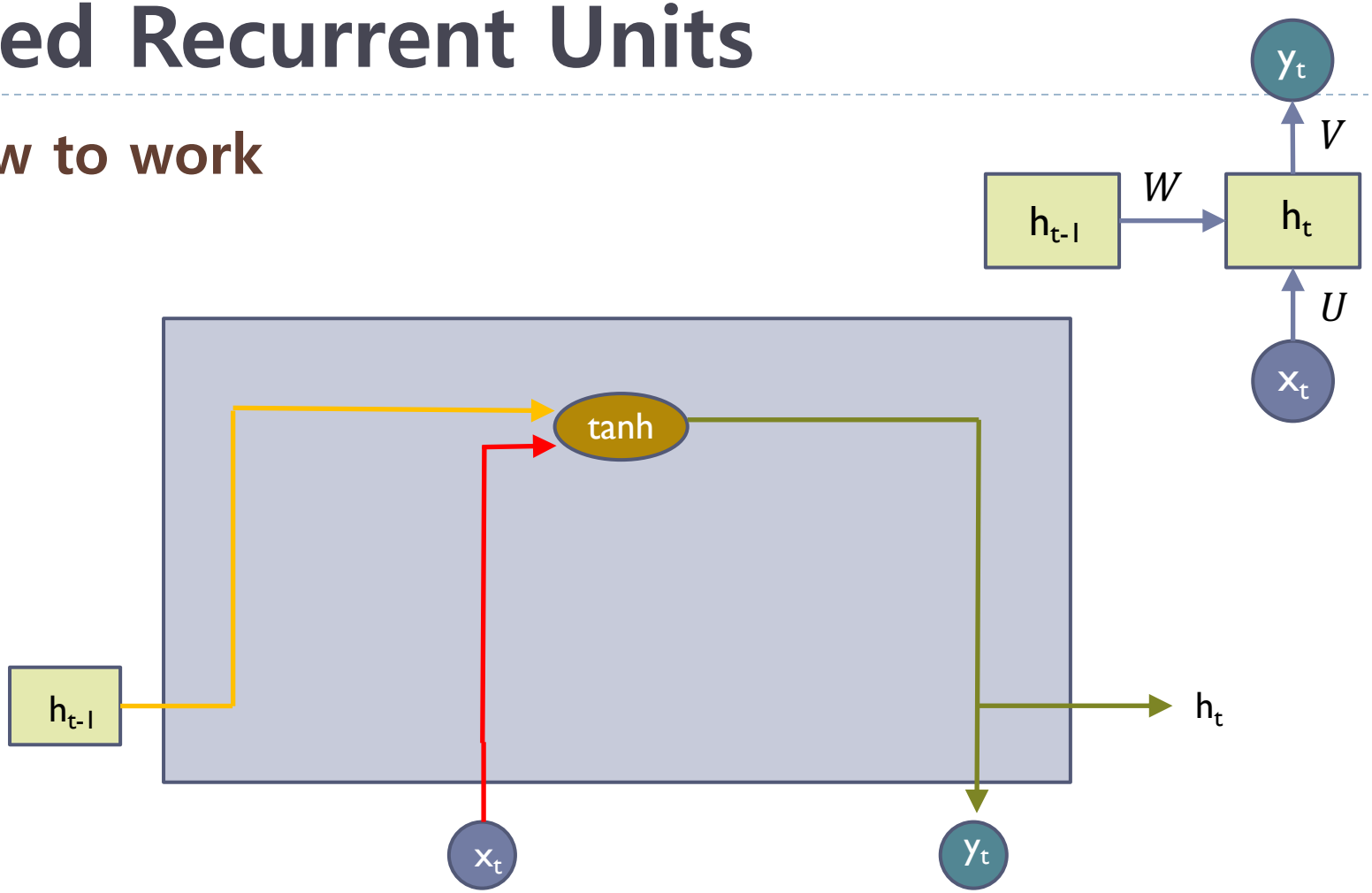
- **Structure**

$$
\begin{aligned}
r_t &= \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \\
z_t &= \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \\
\tilde{h}_t &= \tanh(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h) \\
h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t
\end{aligned}
$$



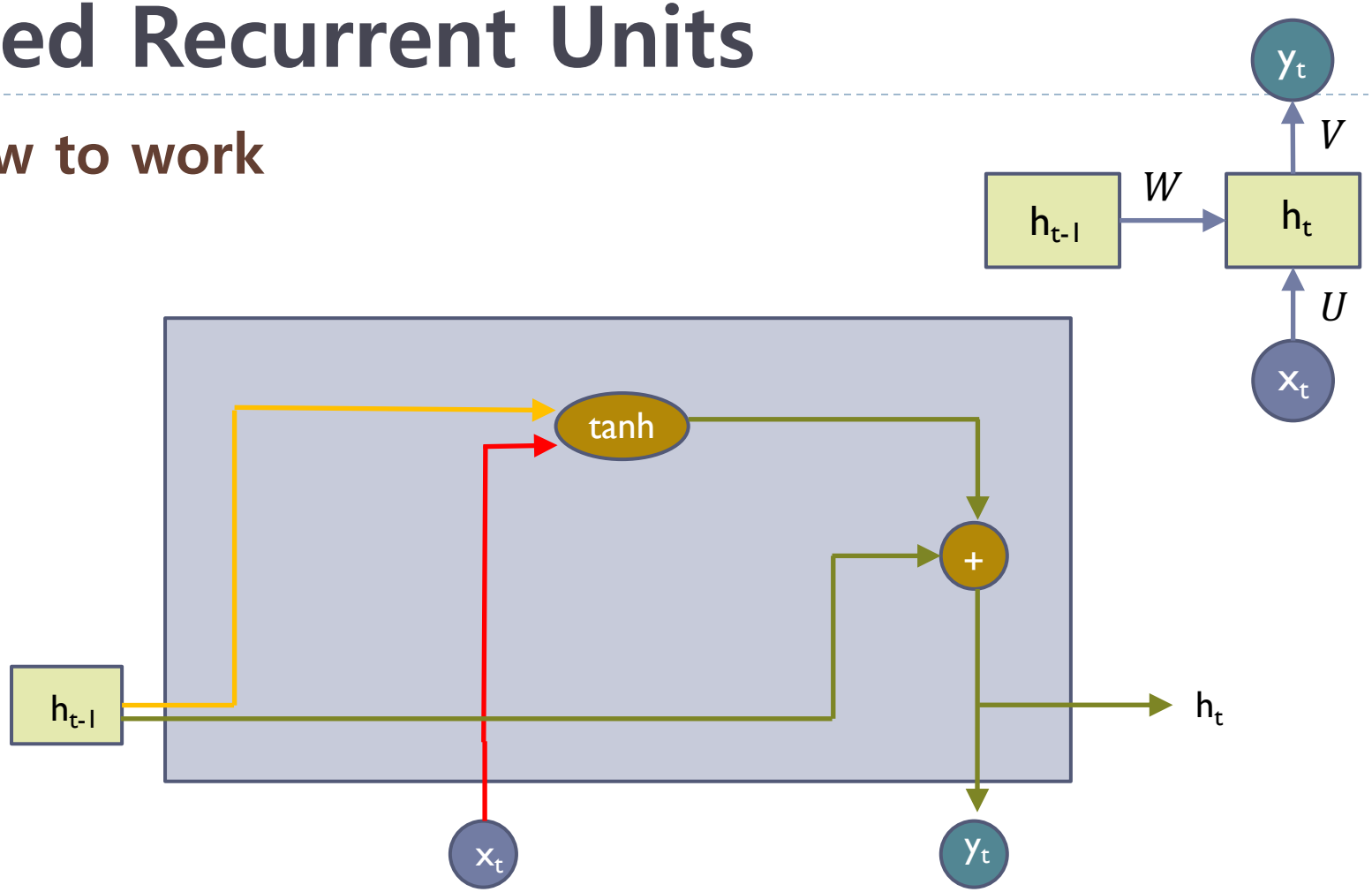
Gated Recurrent Units

- How to work



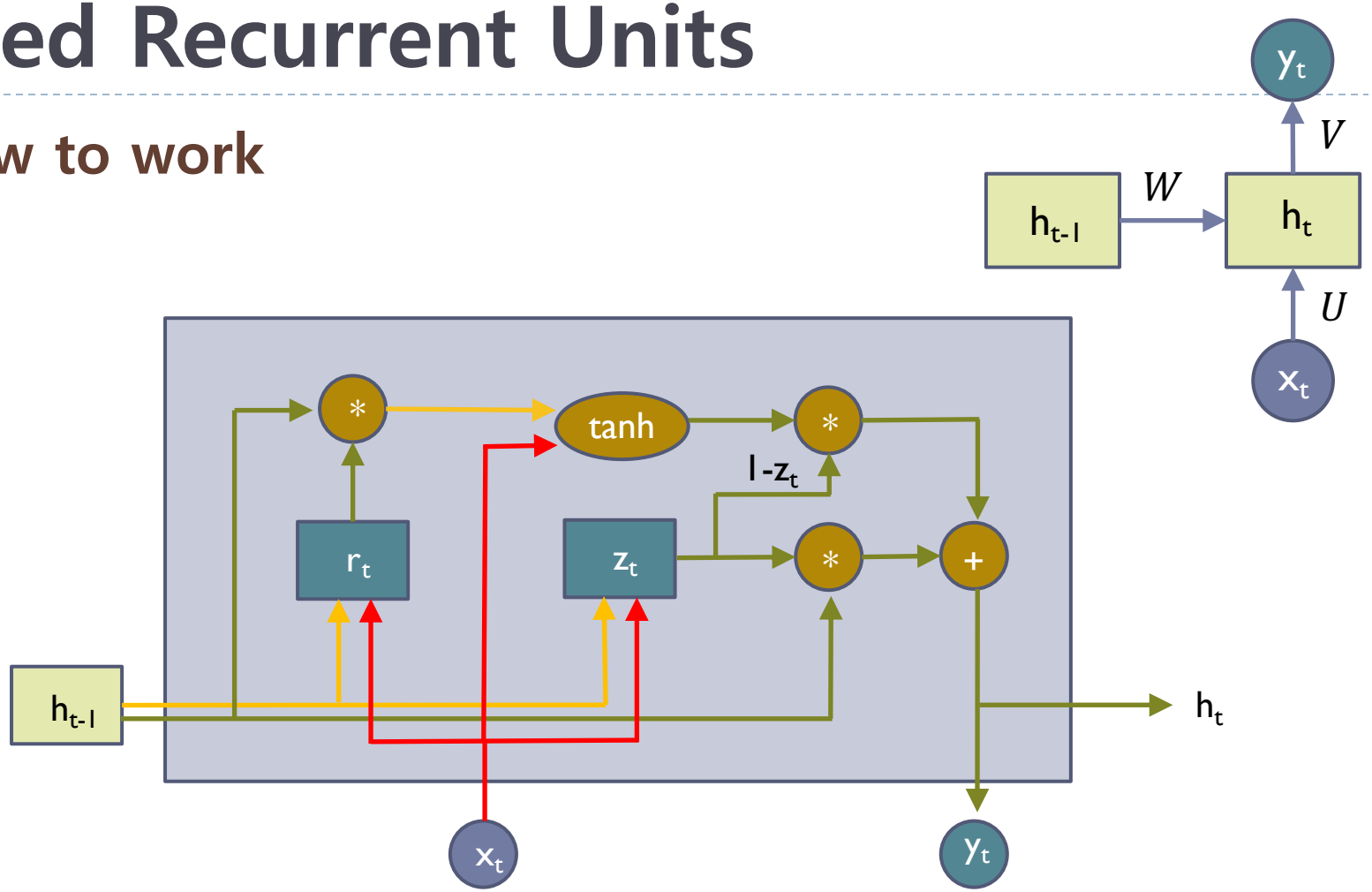
Gated Recurrent Units

- How to work



Gated Recurrent Units

- How to work



Question and Answer