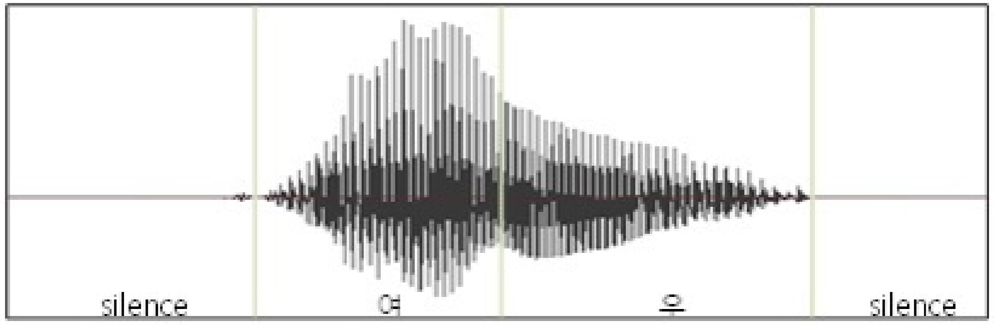
신입생 Deep Learning 기초 교육

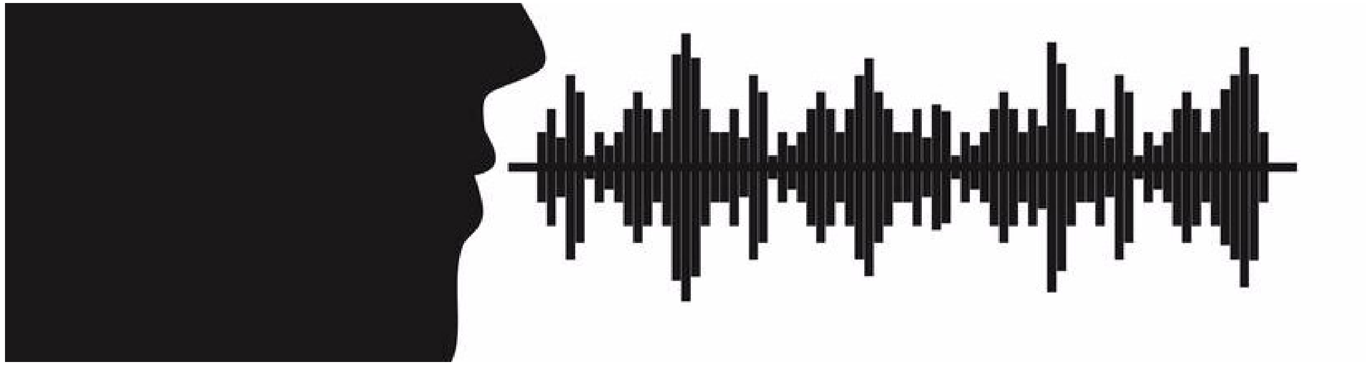
4회: Recurrent neural networks

Multimodal Language Cognition Lab,
Kyungpook National University

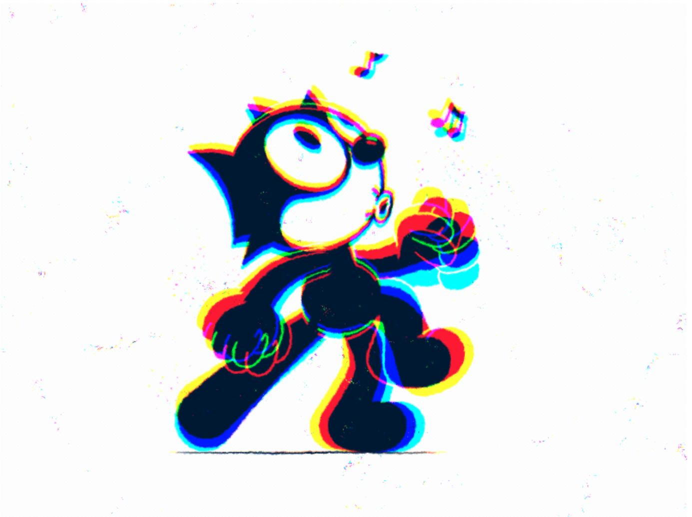
2023.02.08

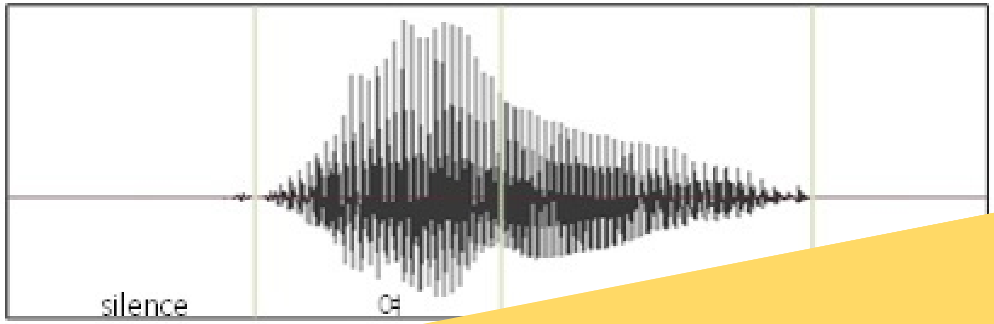
MLCL Kyungpook National University


silence
여
우
silence



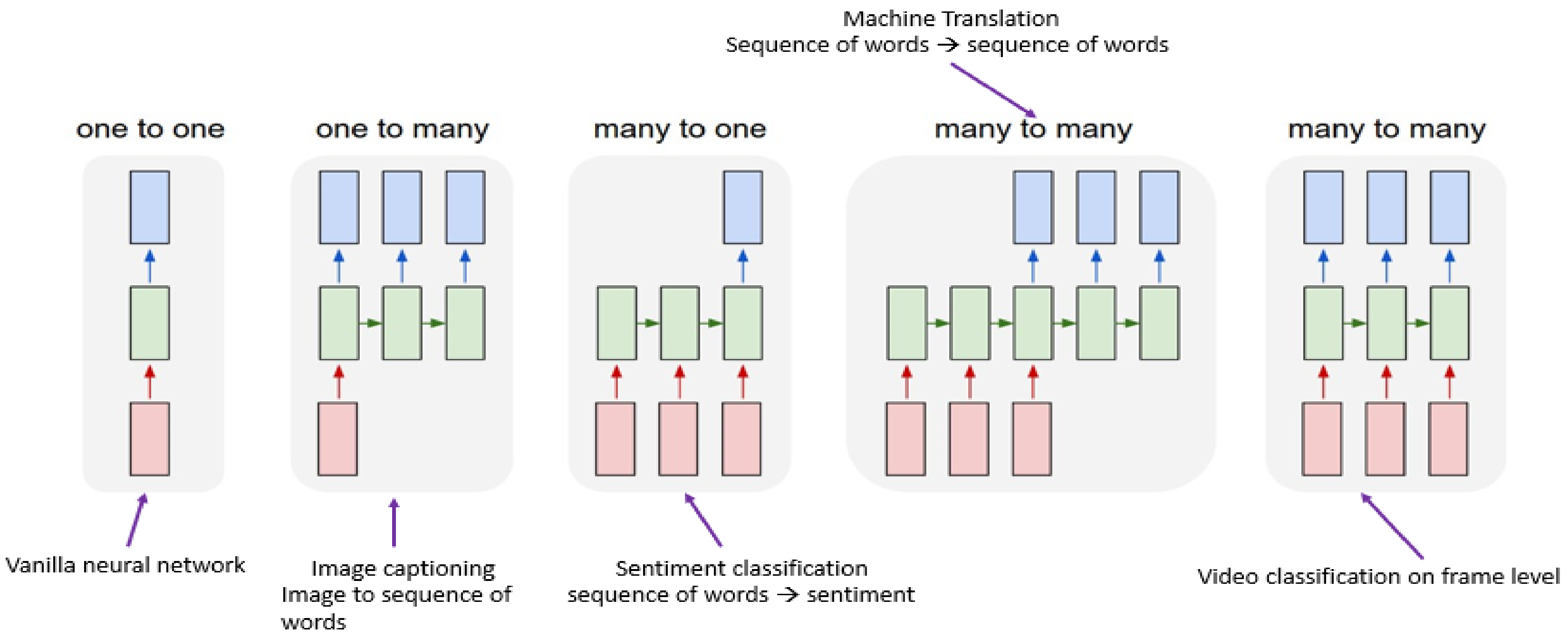
This sentence is a sequence of words…





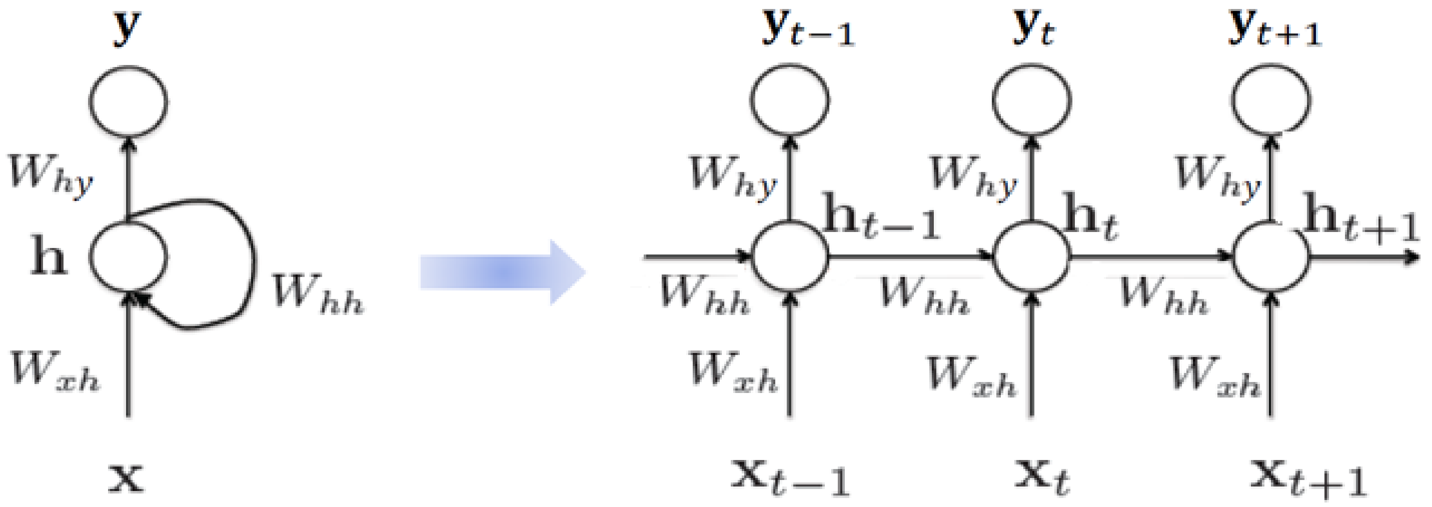
Sequential data !

Recurrent Neural Networks(RNN)



- RNN is a type of neural network that is specialized for processing time series or sequential data
- Key ideas
  - **Recurrence architecture** to exploit the past information of data
  - **Weight sharing** over time for efficient computation

Recurrent Neural Networks(RNN)



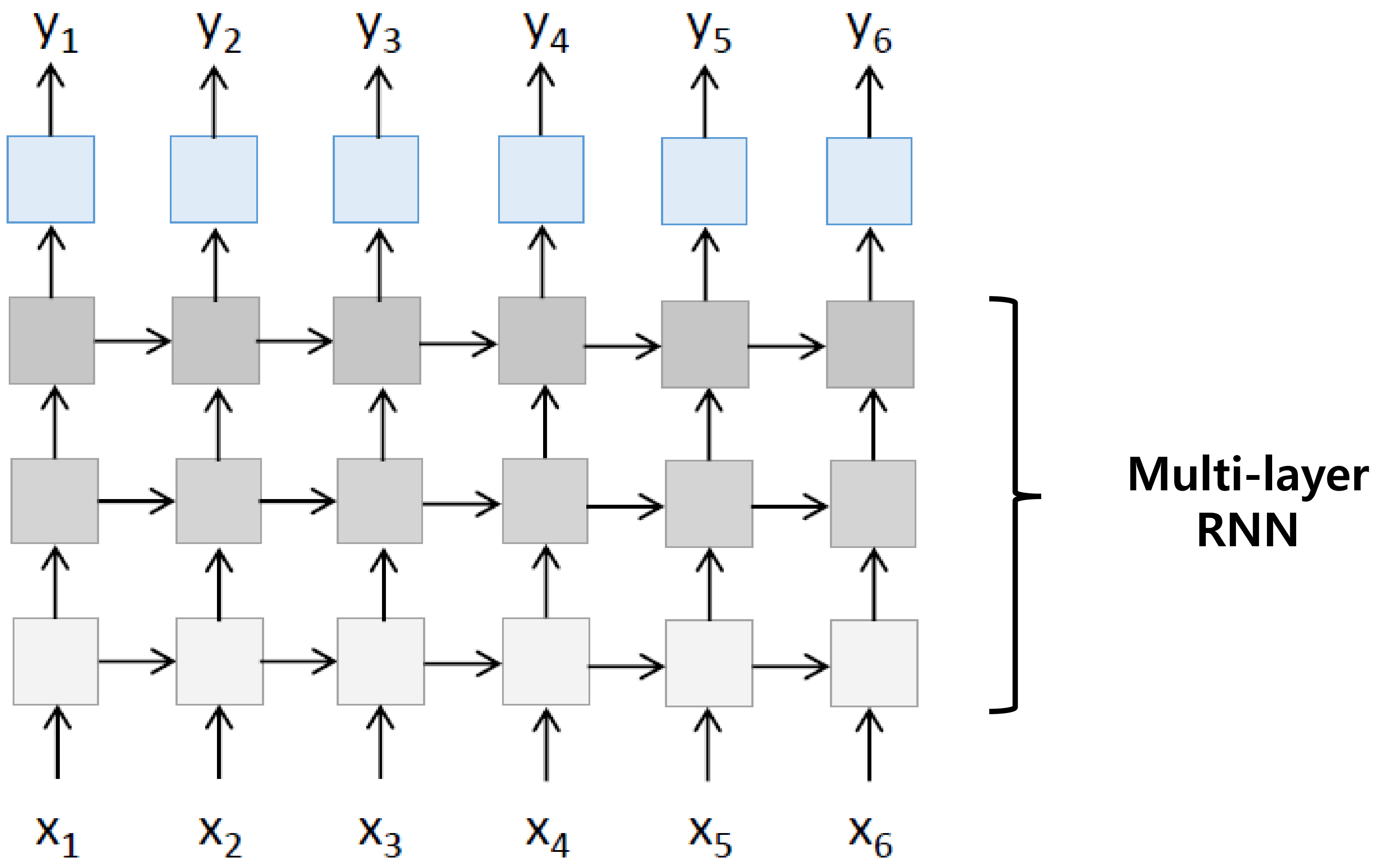
$$h_t = f_W(h_{t-1}, x_t)$$

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

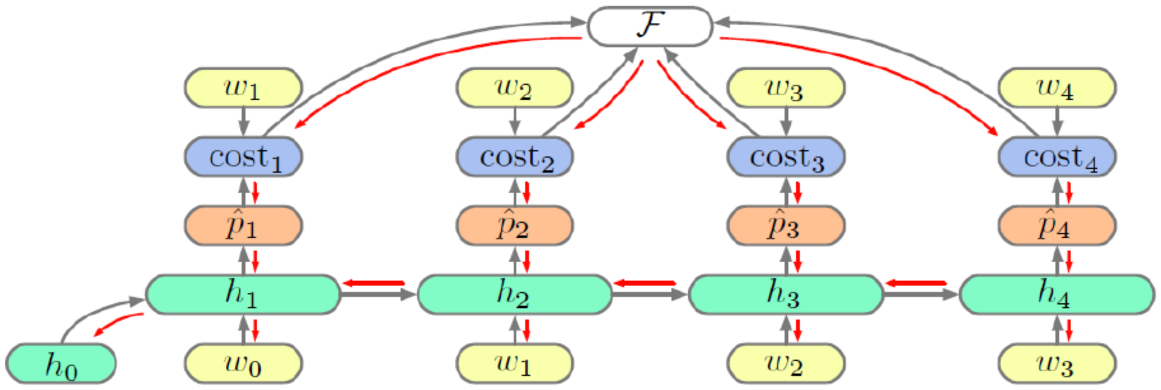
- W_{xh} = Weights connecting the input layer and hidden layer
- W_{hh} = Weights connecting the hidden layer and hidden layer
- W_{hy} = Weights connecting the hidden layer and output layer
- **Weight sharing**
  - The amount of parameters in the model is reduced
  - Independent of the length of the feature vector T

Recurrent Neural Networks(RNN)

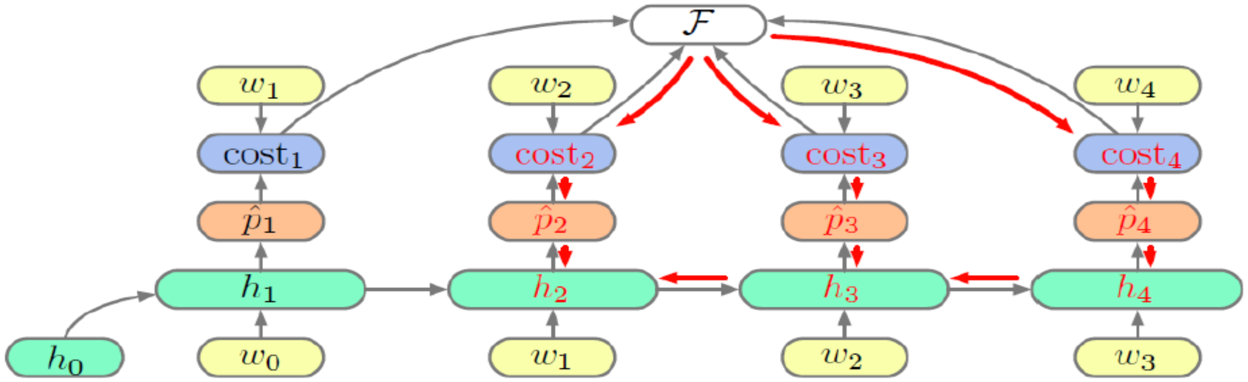


Recurrent Neural Networks(RNN)

$$\mathcal{F} = -\frac{1}{4}\sum_{n=1}^{4} \mathrm{cost}_n(w_n, \hat{p}_n)$$



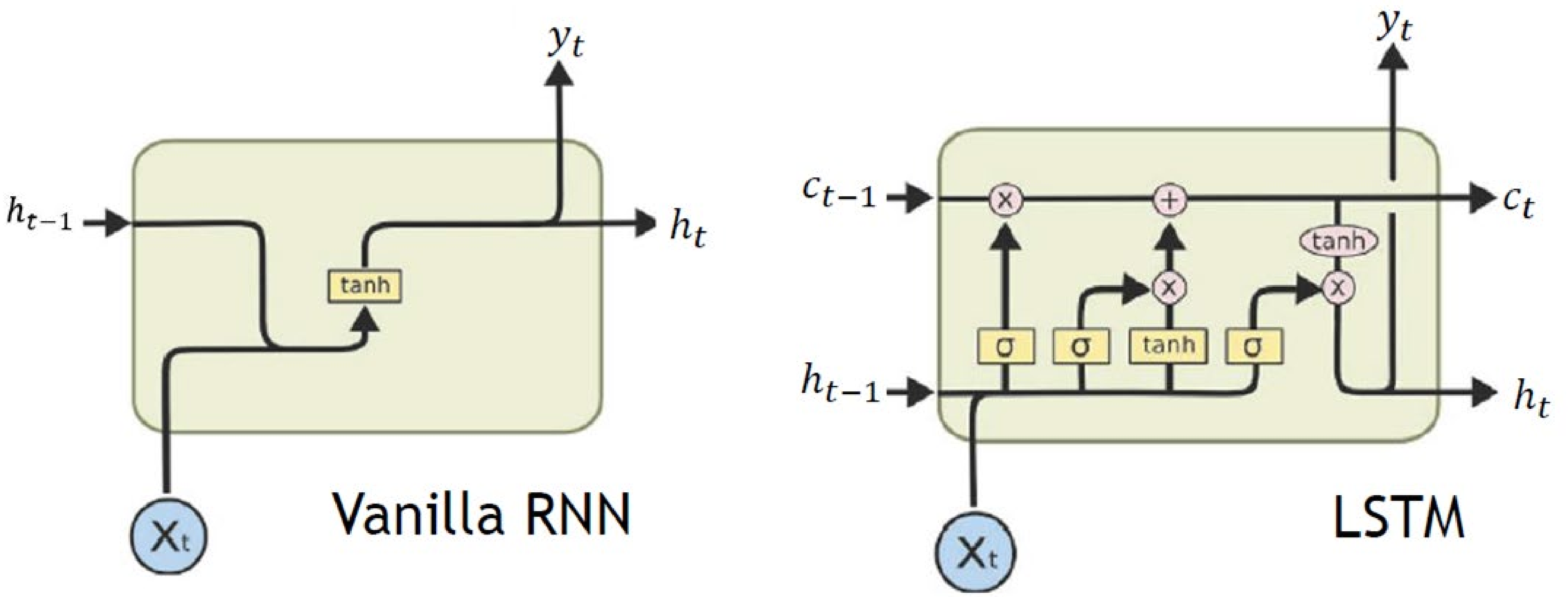
$$\frac{\partial \mathcal{F}}{\partial h_2} = \frac{\partial \mathcal{F}}{\partial \mathrm{cost}_2}\frac{\partial \mathrm{cost}_2}{\partial \hat{p}_2}\frac{\partial \hat{p}_2}{\partial h_2} + \frac{\partial \mathcal{F}}{\partial \mathrm{cost}_3}\frac{\partial \mathrm{cost}_3}{\partial \hat{p}_3}\frac{\partial \hat{p}_3}{\partial h_3}\frac{\partial h_3}{\partial h_2} + \frac{\partial \mathcal{F}}{\partial \mathrm{cost}_4}\frac{\partial \mathrm{cost}_4}{\partial \hat{p}_4}\frac{\partial \hat{p}_4}{\partial h_4}\frac{\partial h_4}{\partial h_3}\frac{\partial h_3}{\partial h_2}$$



- $\widehat{p_t}$ = Current step t output as probability value
- High computational complexity to compute gradients
- Exploding and **vanishing gradients**
- Difficult to realize the parallel computation due to recurrence

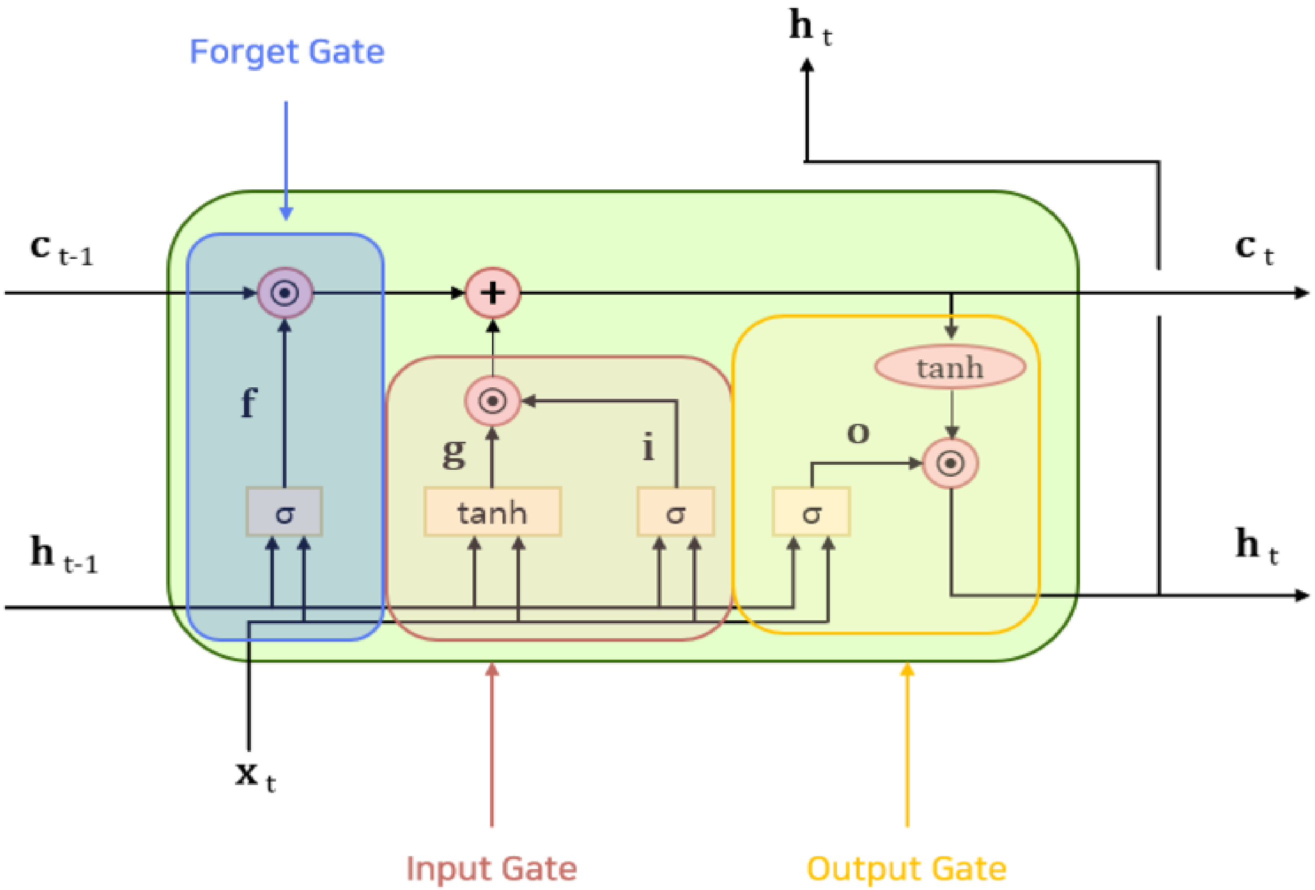
Vanilla RNN vs LSTM

- hidden state(h_t): output state at timer t (short-term memory)
- cell state(c_t): internal state at time t (long-term memory)
- memory control gates(forget gate, input gate) and output control gate



MLCL Kyungpook National University

LSTM



- **Cell state, Hidden state**

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

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

- **Forget gate**

$$f = \sigma\left(x_t W_x^f + h_{t-1} W_h^f + b^f\right)$$

- **Input gate**

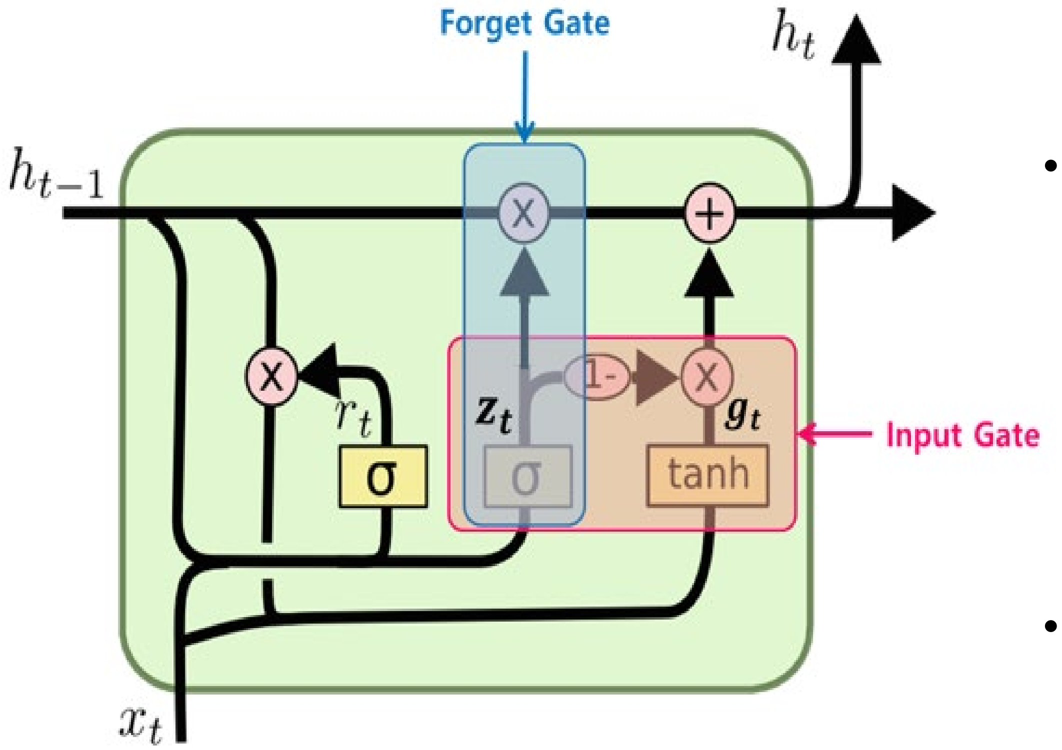
$$g = tanh\left(x_t W_x^g + h_{t-1} W_h^g + b^g\right)$$

$$i = \sigma\left(x_t W_x^i + h_{t-1} W_h^i + b^i\right)$$

- **Output gate**

$$o = \sigma\left(x_t W_x^o + h_{t-1} W_h^o + b^o\right)$$

GRU



- **Hidden state**

  $h_t = (1 - z_t) * g_t + z_t * h_{t-1}$

- **Reset gate**

  $r_t = \sigma(x_t W_x^r + h_{t-1} W_h^r + b^r)$

  $g_t = tanh(x_t W_x^n + r_t * (h_{t-1} W_h^n) + b^n)$

  Reset the information of the previous hidden state (h_{t-1})

- **Update gate**

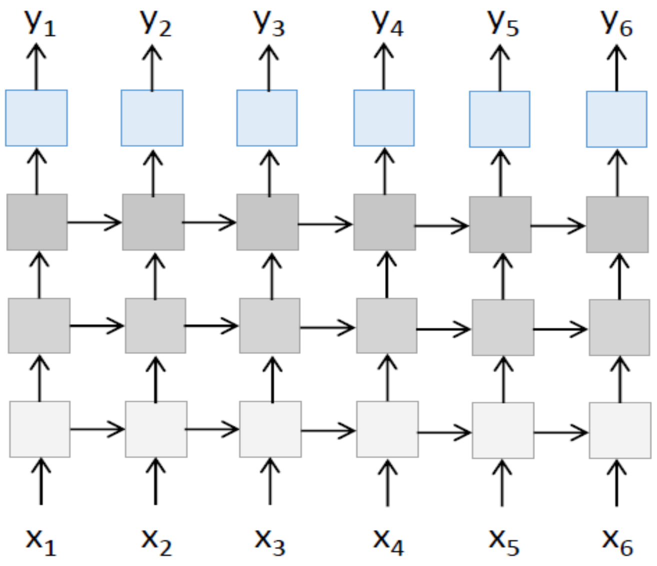
  $z_t = \sigma(x_t W_x^z + h_{t-1} W_h^z + b^z)$

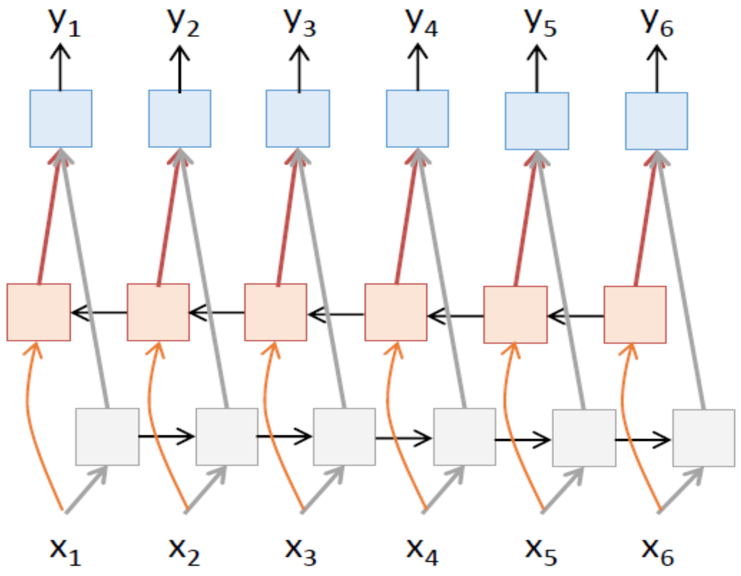
  z_t determines the ratio of **previous information** (forget)

  $(1 - z_t)$ determines the ratio of **current information** (input)

MLCL Kyungpook National University

Bidirectional RNN






```python
@d2l.add_to_class(BiRNNScratch)
def forward(self, inputs, Hs=None):
    f_H, b_H = Hs if Hs is not None else (None, None)
    f_outputs, f_H = self.f_rnn(inputs, f_H)
    b_outputs, b_H = self.b_rnn(reversed(inputs), b_H)
    outputs = [torch.cat((f, b), -1) for f, b in zip(
        f_outputs, reversed(b_outputs))]
    return outputs, (f_H, b_H)
```

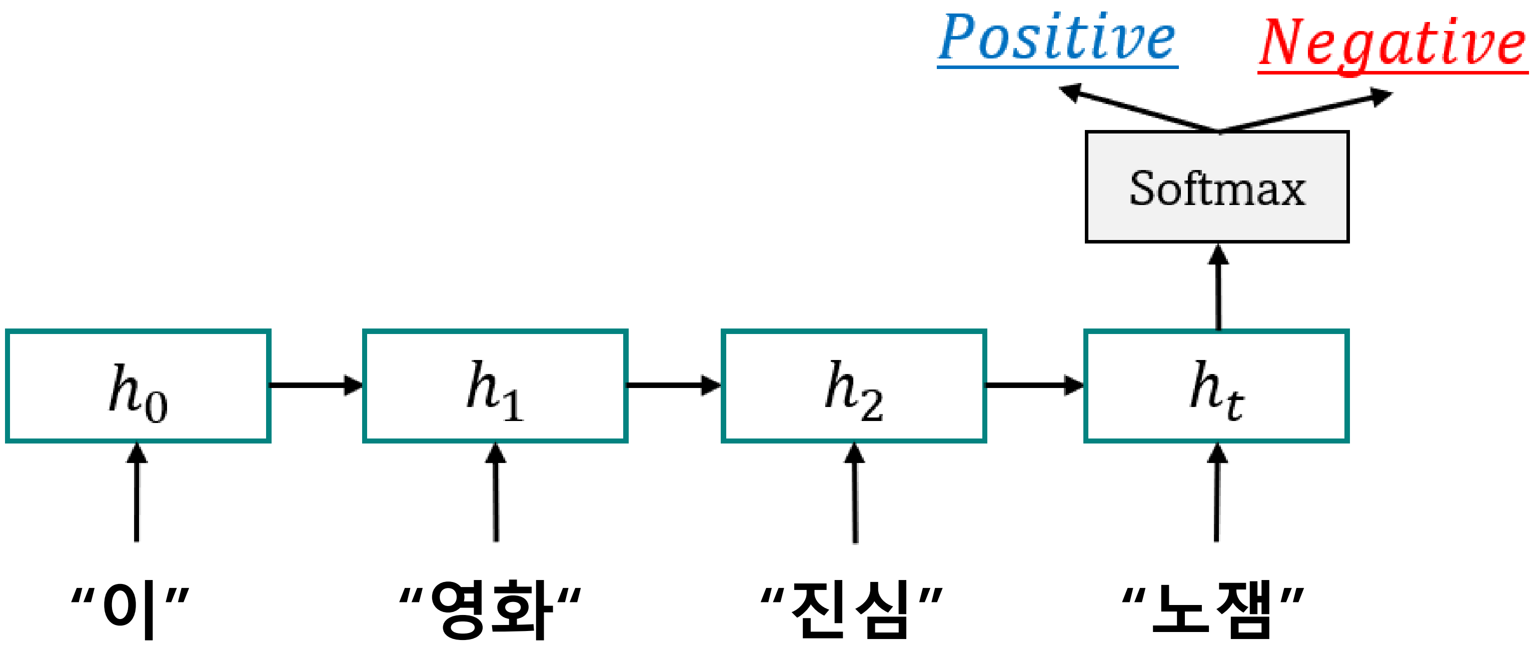
- RNN-based models have past information O, future information X
- Example

    i) I am happy

    ii) I am very hungry

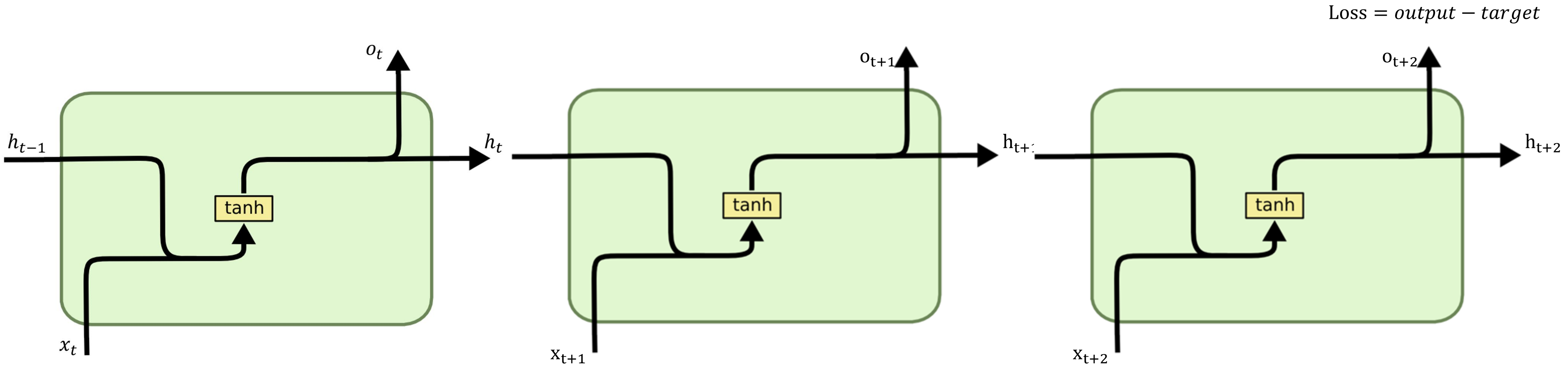
    iii) I am very hungry, and I can eat half a pig

Code



https://colab.research.google.com/drive/1_TXxuBtYv5lPCtKPGidJfFLdvGwN8yAv?usp=share_link

Homework1



$h_t = \tanh(h_{t-1}W_h + x_t W_x + \text{b})$

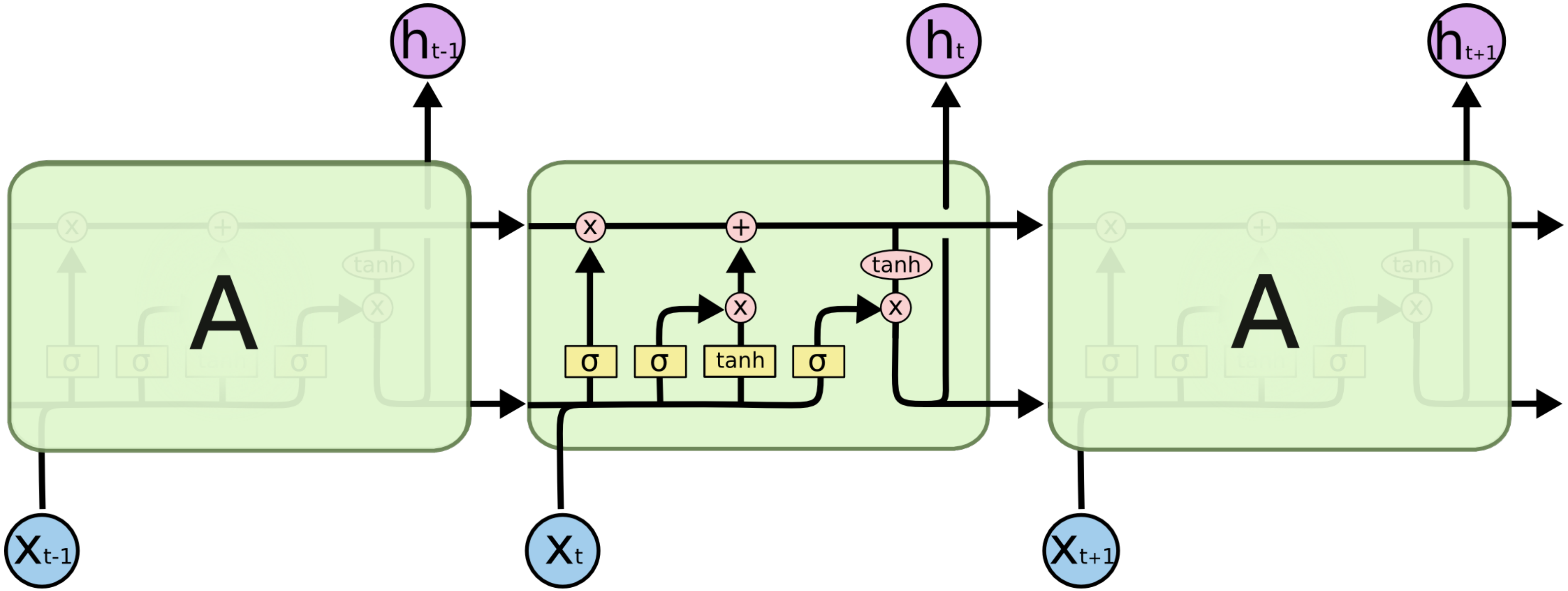
$h_{t+1} = \tanh(h_t W_h + x_{t+1} W_x + \text{b})$

$h_{t+2} = \tanh(h_{t+1} W_h + x_{t+2} W_x + \text{b})$

$$\frac{\partial \text{L}}{\partial W_h} = ?$$

$$\frac{\partial \text{L}}{\partial W_\text{x}} = ?$$

Homework2



LSTM uses the sigmoid and tanh

This is different for each gate

Why?

MLCL Kyungpook National University