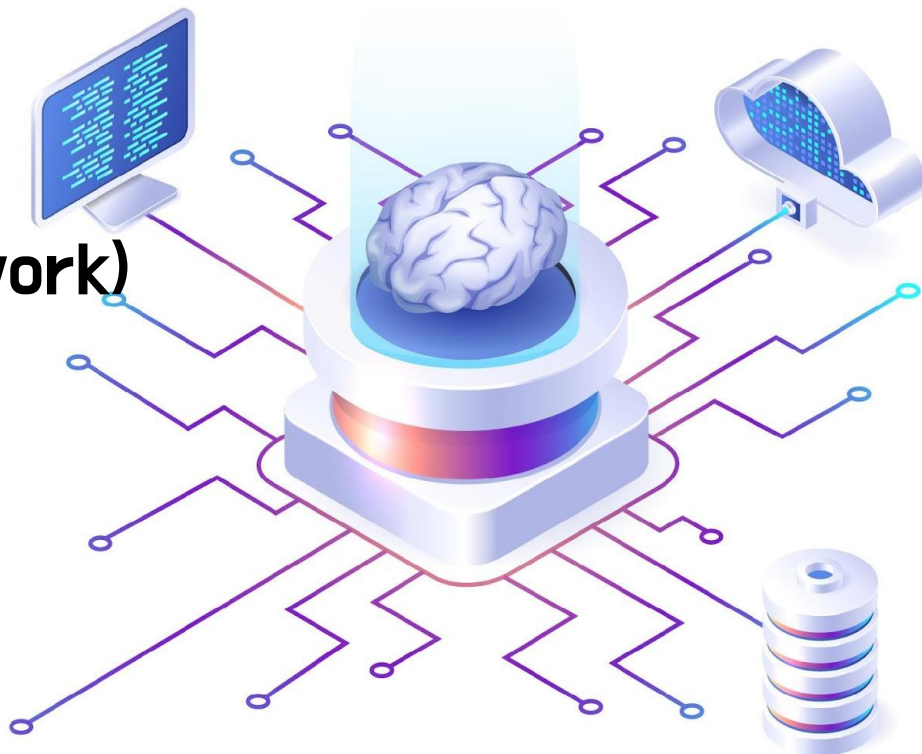


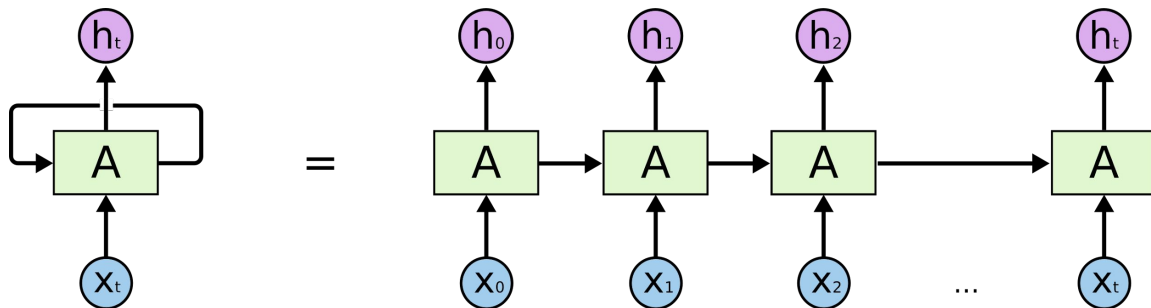
RNN (Recurrent Neural Network)

실무형 인공지능 자연어처리



RNN (Recurrent Neural Network)

- 순환 인공 신경망 (RNN)은 인공 신경망의 한 종류
- 유닛간의 연결이 순환적 구조
- 이러한 구조는 시계열 특징을 모델링 할 수 있음



https://en.wikipedia.org/wiki/Recurrent_neural_network

RNN 활용 (1)

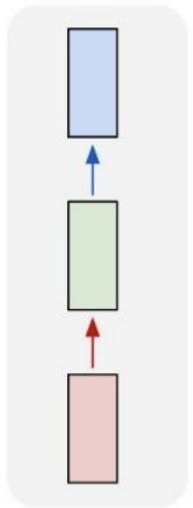
주식과 같은 연속적인(sequential) 시계열(time series) 데이터에 적합

- 언어 모델링(Mikolov et al., 2010, 2011; Sutskever et al., 2011)
- 기계번역(Liu et al., 2014; Auli et al., 2013; Sutskever et al., 2014)
- 음성인식(Robinson et al., 1996; Graves et al., 2013; Graves and Jaitly, 2014; Sak et al., 2014)
- 이미지 캡셔닝(Karpathy and Fei-Fei, 2015)

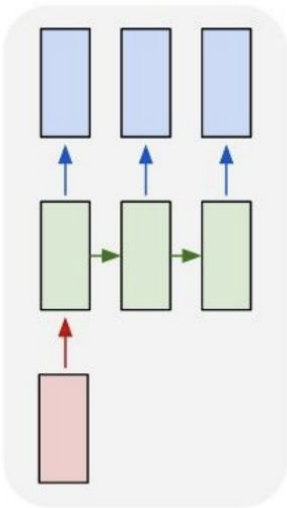
RNN 활용 (2)

Image Captioning
image → sequence of words

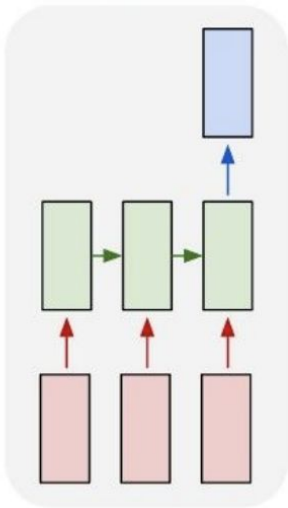
one to one



one to many

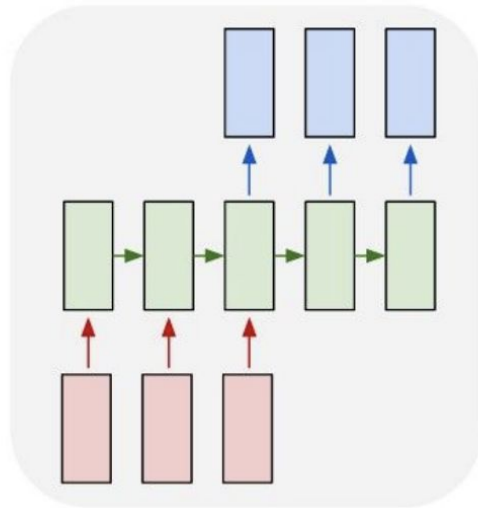


many to one

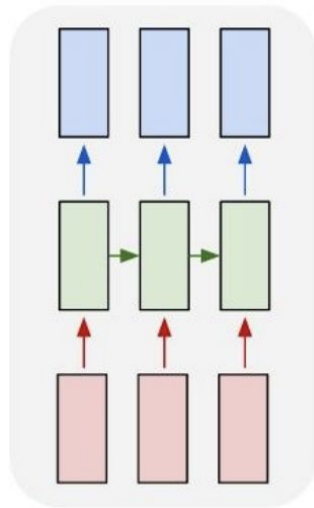


Machine Translation
seq of words → seq of words

many to many



many to many



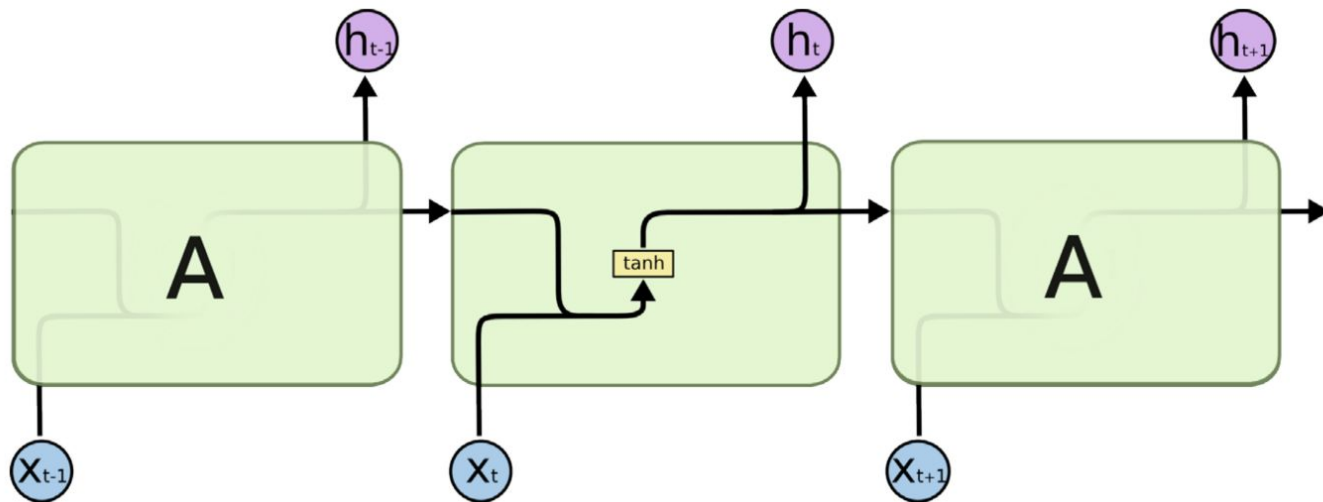
Vanilla Neural Networks

Sentiment Classification
sequence of words → sentiment

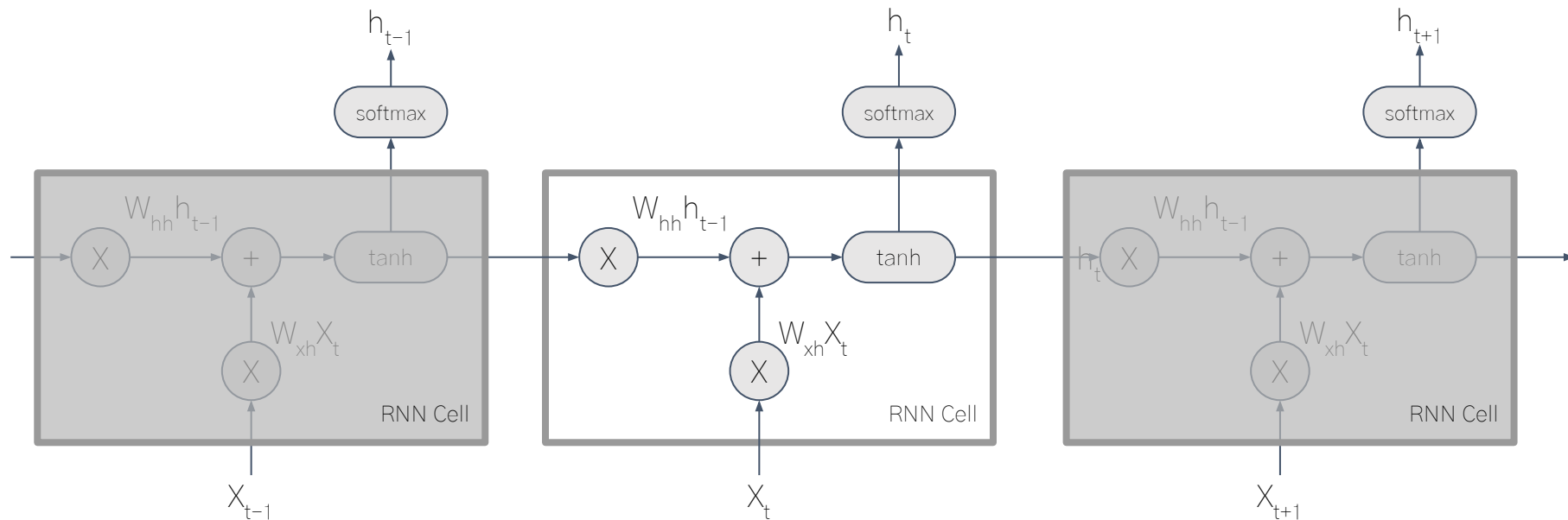
Video classification on frame level

http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf

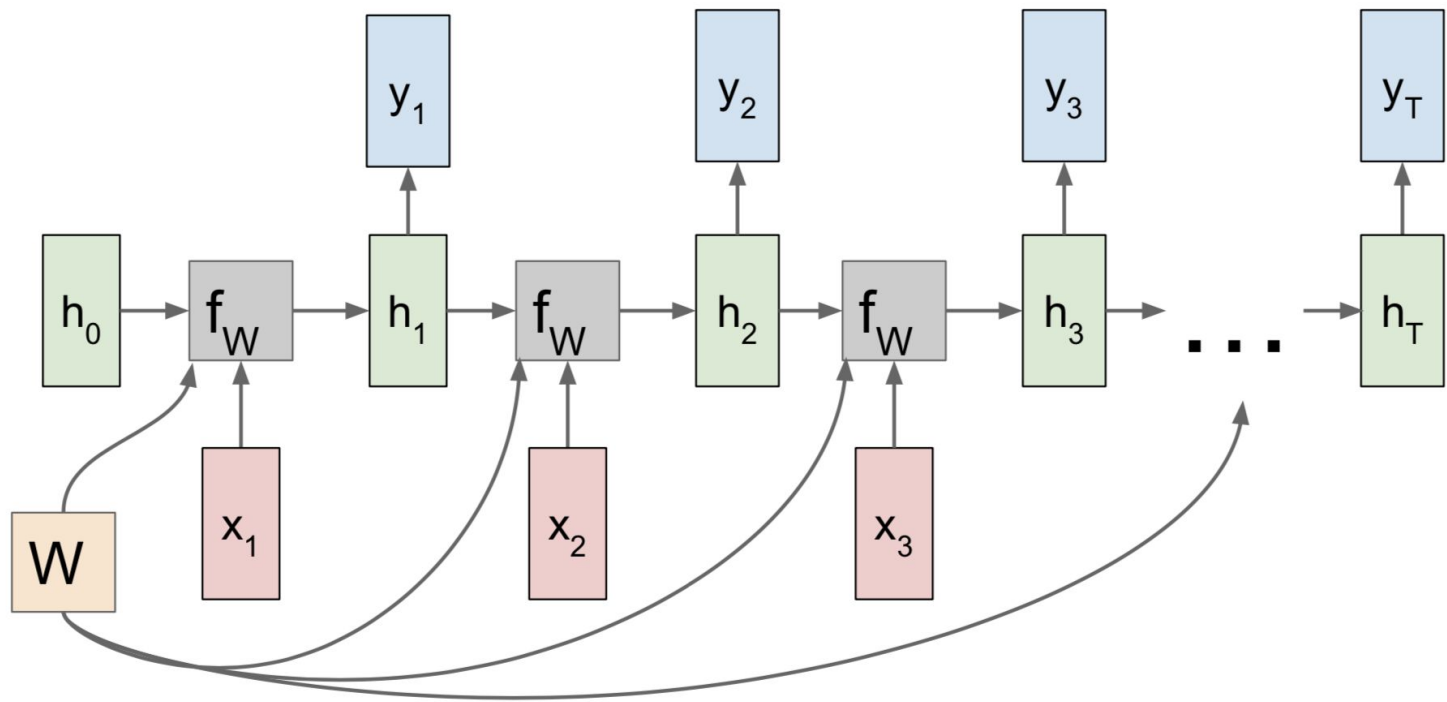
RNN (Recurrent Neural Network)



RNN (Recurrent Neural Network)



RNN (Recurrent Neural Network)



RNN (Recurrent Neural Network)

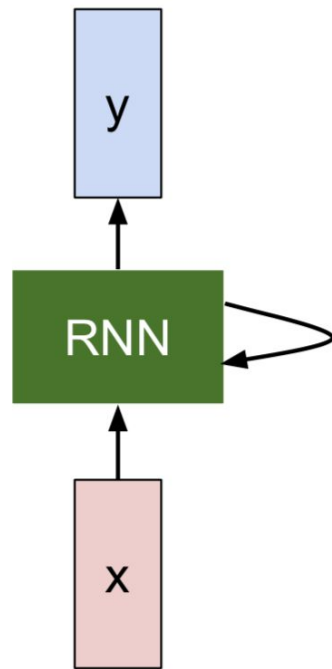
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

some function with parameters W

old state

input vector at some time step



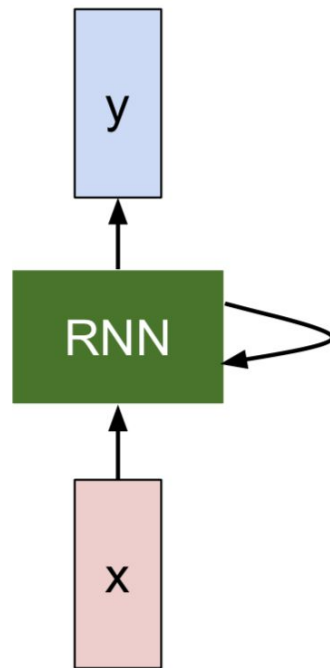
RNN (Recurrent Neural Network)

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



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

$$y_t = W_{hy}h_t$$



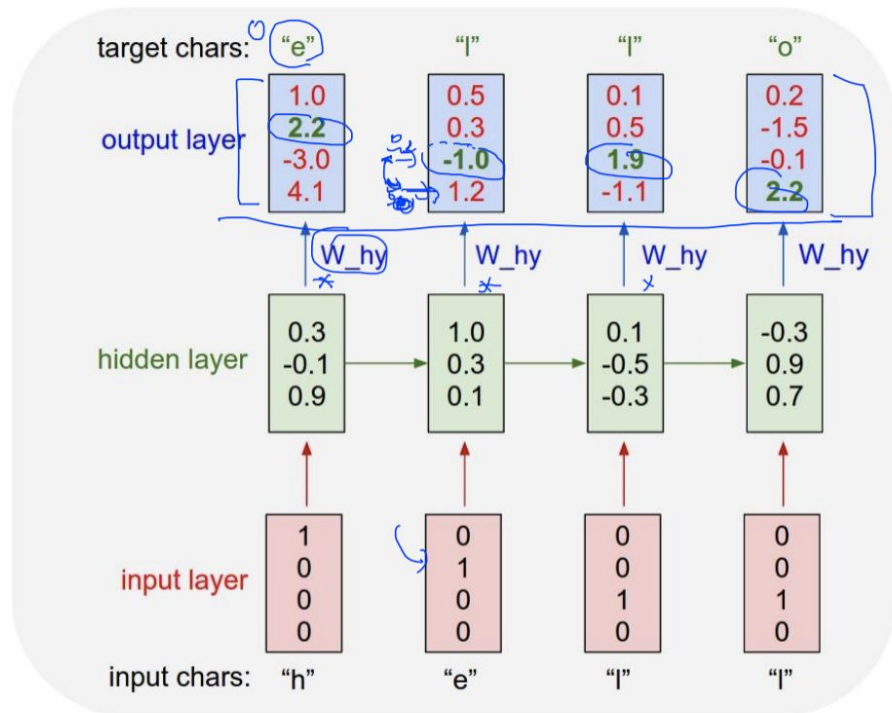
RNN 예시

Character-level language model example

$$y_t = W_{hy}h_t$$

Vocabulary:
[h,e,l,o]

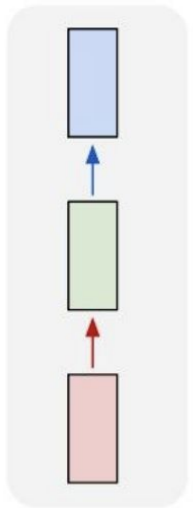
Example training sequence:
“hello”



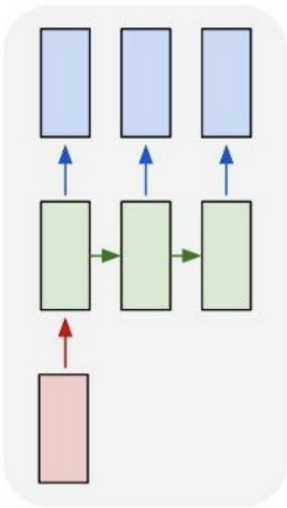
RNN 활용 (2)

Image Captioning
image → sequence of words

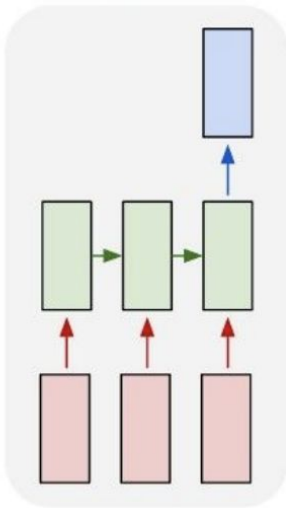
one to one



one to many

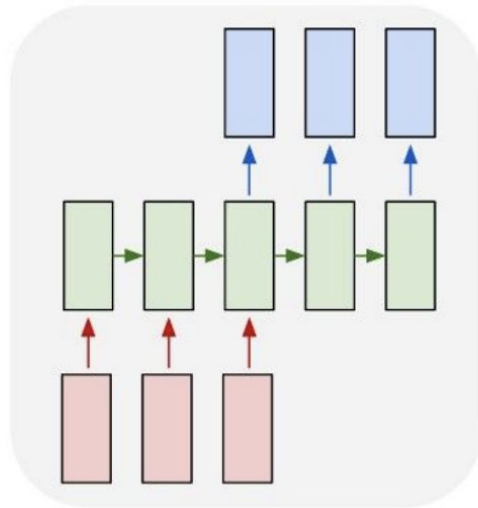


many to one

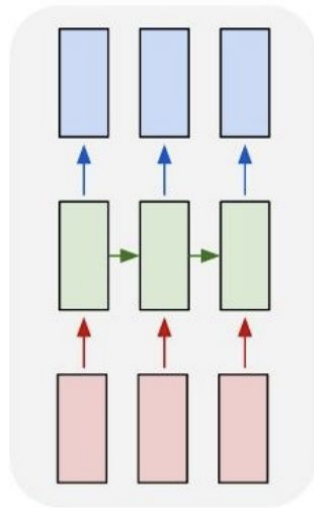


Machine Translation
seq of words → seq of words

many to many



many to many



Vanilla Neural Networks

Sentiment Classification
sequence of words → sentiment

Video classification on frame level

http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf

RNN (Recurrent Neural Network)

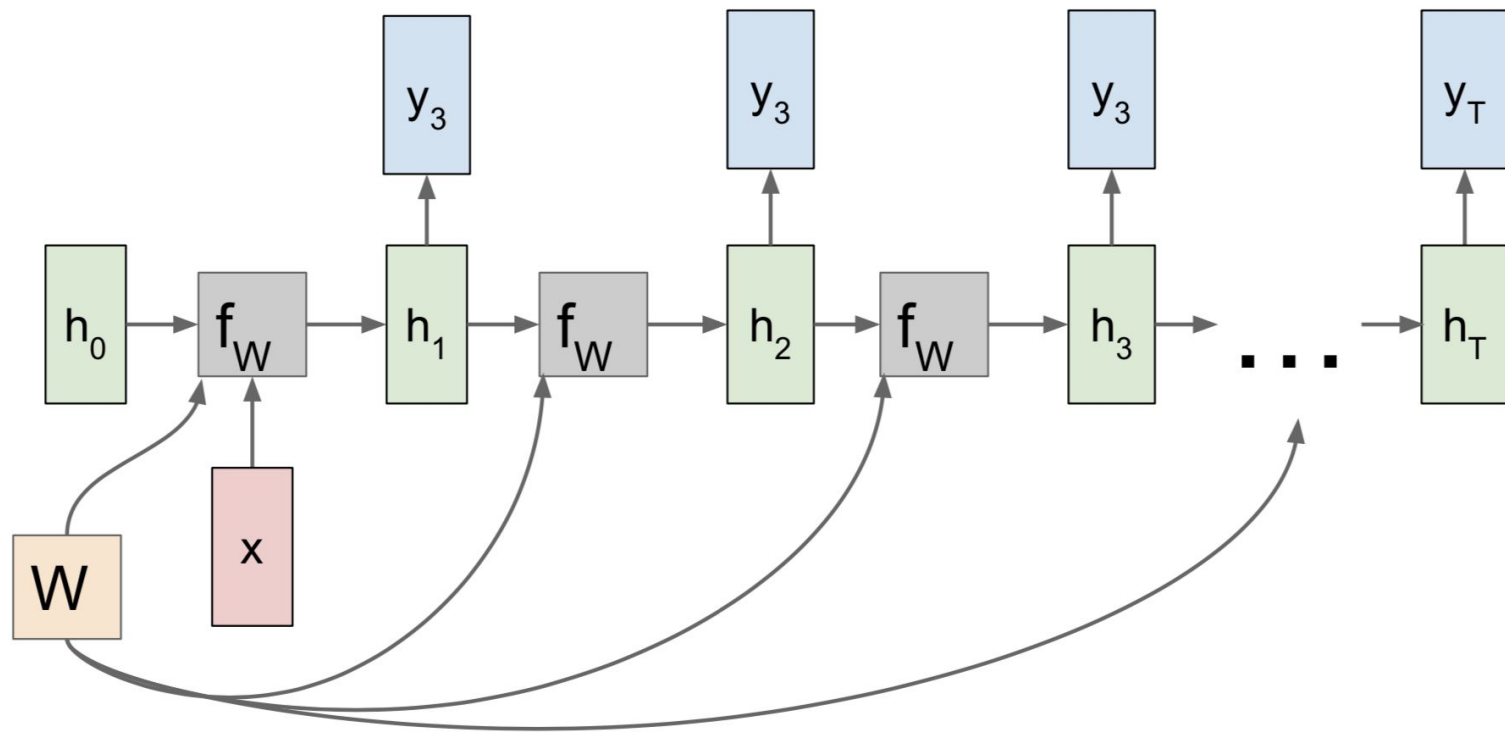
딥러닝기반 자연어 처리

1

RNN - One to Many

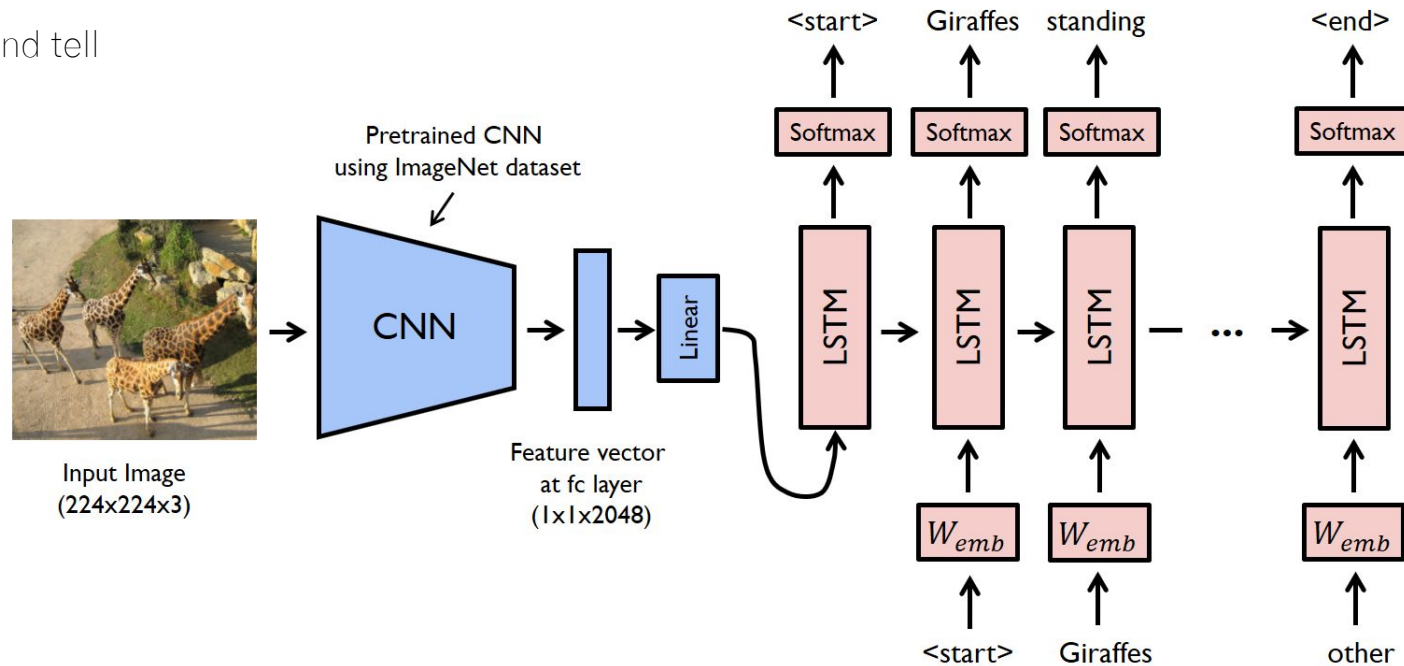


RNN : One to Many



RNN : One to Many

show and tell



RNN (Recurrent Neural Network)

딥러닝기반 자연어 처리

2

RNN - Many to One



RNN : Many to One (1)

Sequence classification

eg. classify polarity of sentence

sequence : sentence, tokens : word

['This movie is good']

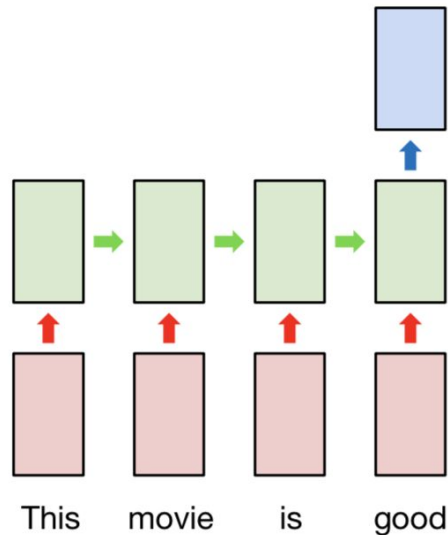
↓ *Tokenization*

['This', 'movie', 'is', 'good']

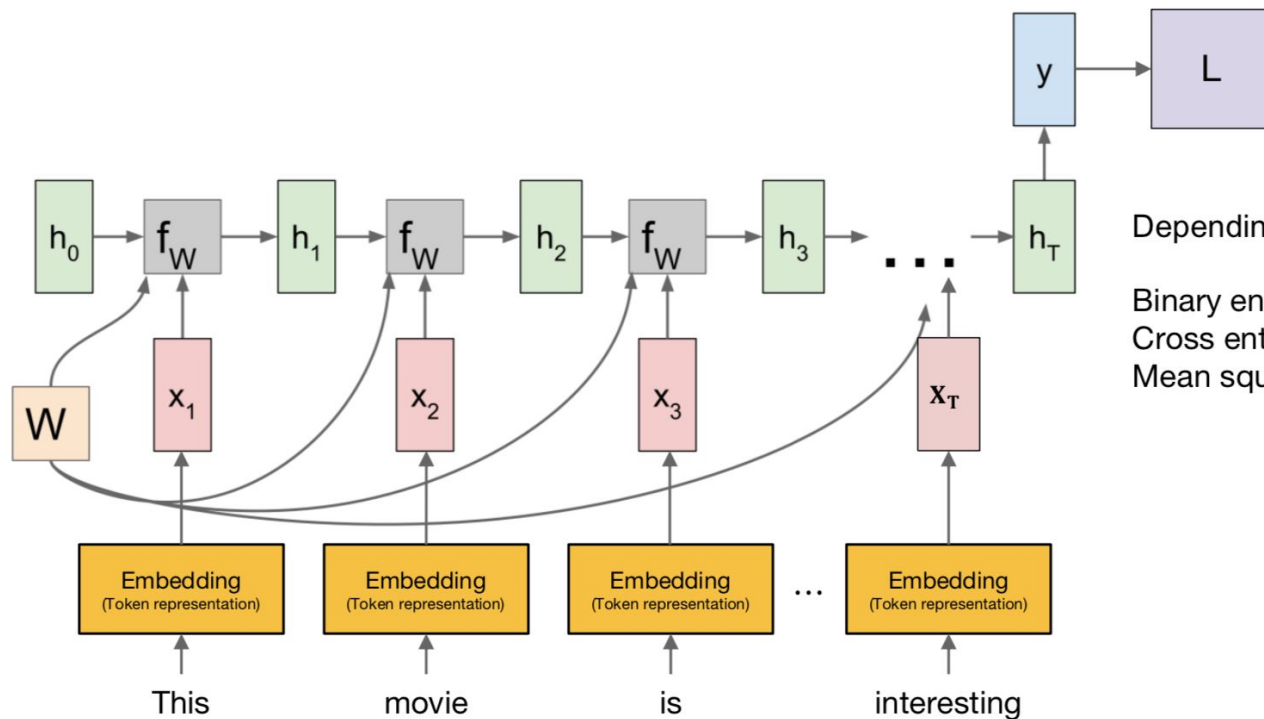
↓ *Classification*

Positive

Classification : **Positive** or **negative**?



RNN : Many to One (2)



Depending on the task...

Binary entropy loss (binary classifier)
Cross entropy loss (softmax classifier)
Mean squared loss (regression)

RNN (Recurrent Neural Network)

딥러닝기반 자연어 처리

3

RNN - Many to One Stacking



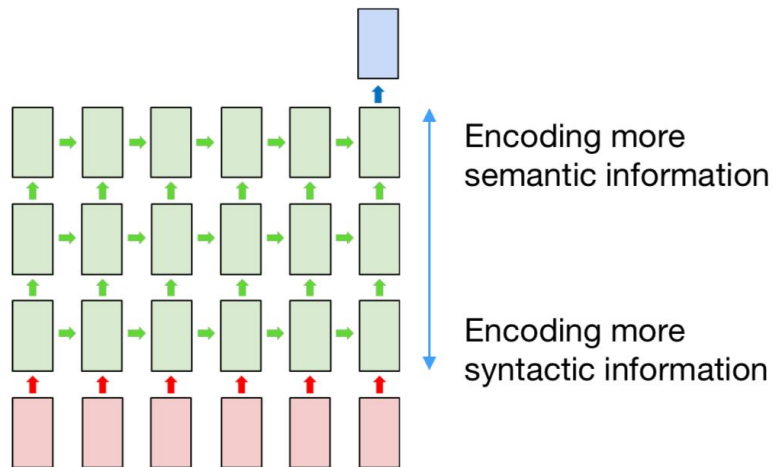
RNN : Many to One Stacking (1)

- What is “stacking”?
- many to one stacking
- Example : sentence classification
 - Preparing dataset
 - Creating and training model
 - Checking performance

RNN : Many to One Stacking (2)

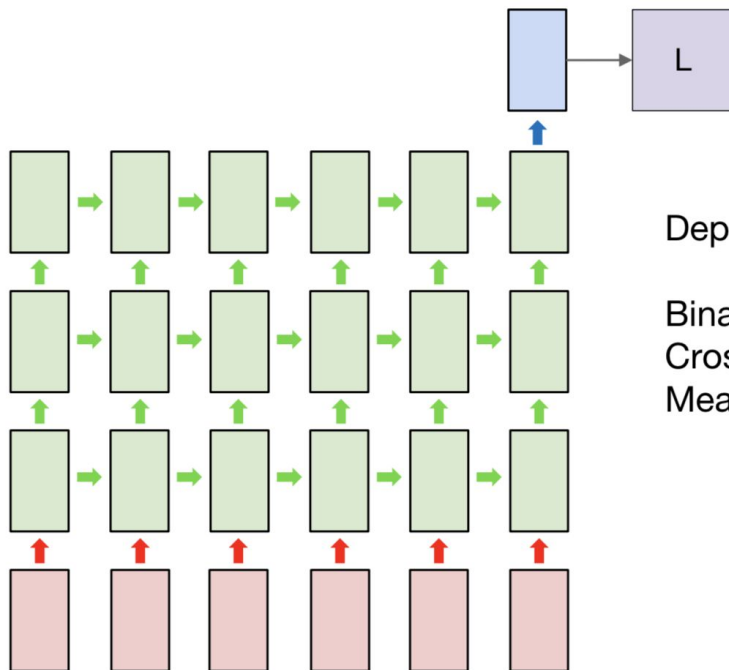
What is “stacking”?

Besides, many works have shown that different layers of deep RNNs encode different types of information.



RNN : Many to One Stacking (3)

many to one stacking



Depending on the task...

Binary entropy loss (binary classifier)
Cross entropy loss (softmax classifier)
Mean squared loss (regression)

RNN (Recurrent Neural Network)

딥러닝기반 자연어 처리

4

RNN - Many to Many



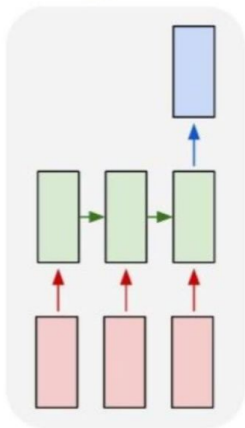
RNN : Many to Many (1)

- What is “many to many”?
- Example : part of speech tagging
 - Preparing dataset
 - Creating and training model
 - Checking performance

RNN : Many to Many (1)

What is “many to many”?

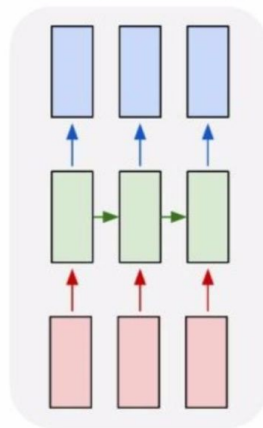
many to one



producing an output for
final input it reads in.

VS

many to many



producing an output for
each input it reads in.

RNN : Many to Many (2)

What is “many to many”?

Sequence tagging

eg. part of speech tagging

sequence : sentence, tokens : word

['tensorflow is very easy']

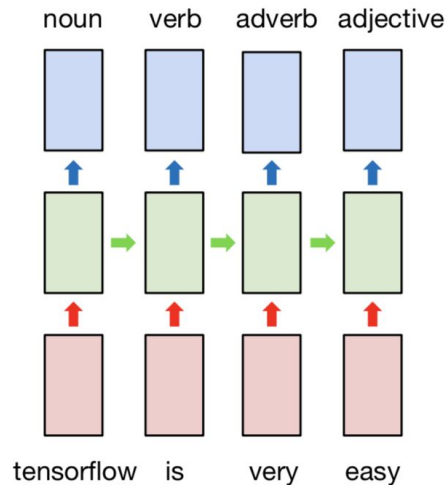
↓ Tokenization

['tensorflow', 'is', 'very', 'easy']

↓ Tagging

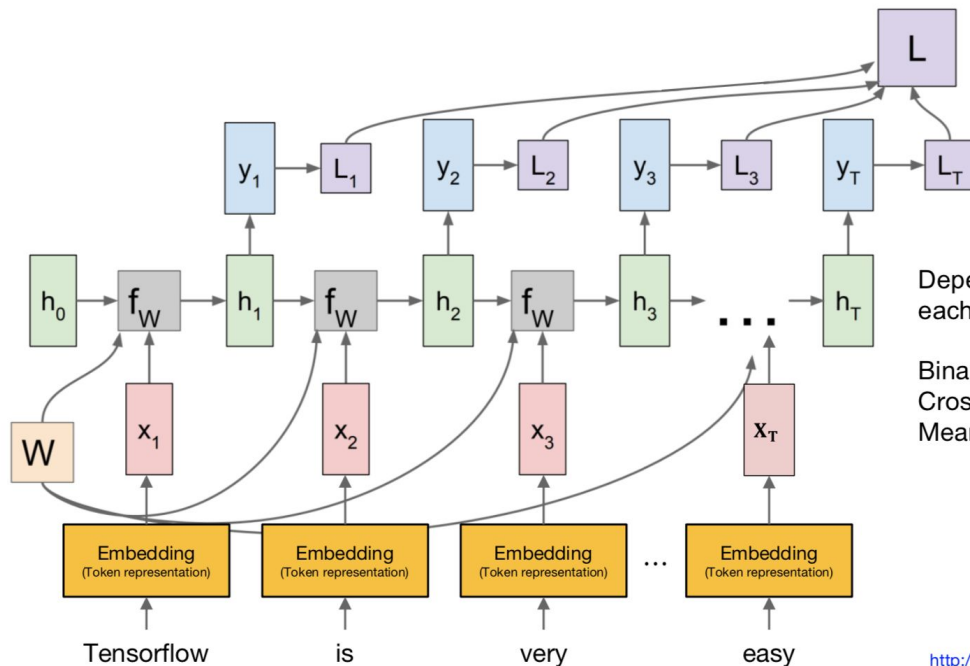
['noun', 'verb', 'adverb', 'adjective']

classification (each time step)



RNN : Many to Many (3)

What is “many to many”?



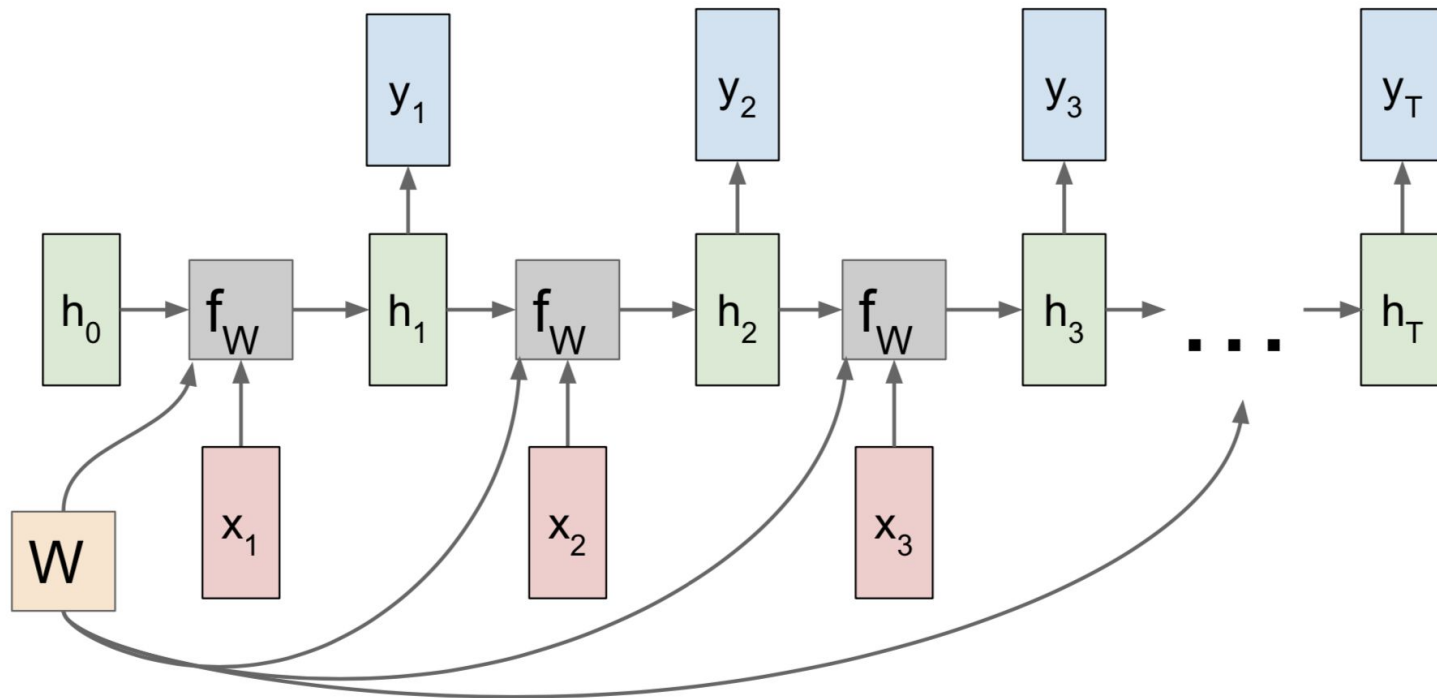
Depending on the task with regard to each time step...

Binary entropy loss (binary classifier)
Cross entropy loss (softmax classifier)
Mean squared loss (regression)

aka Sequence loss!

http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture10.pdf

RNN : Many to Many (4)



RNN (Recurrent Neural Network)

딥러닝기반 자연어 처리

5

RNN - Many to Many Bidirectional

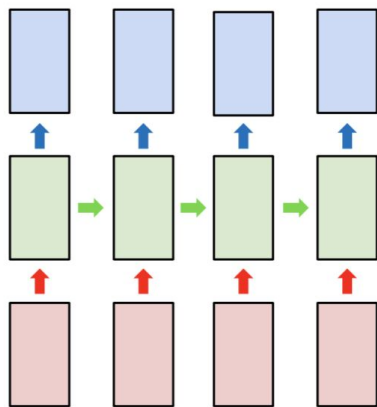


RNN : Many to Many Bidirectional

- What is “bidirectional”?
- many to many bidirectional
- Example : part of speech tagging
 - Preparing dataset
 - Creating and training model
 - Checking performance

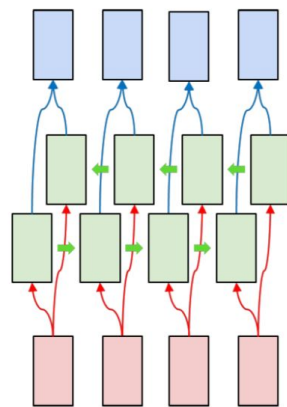
RNN : Many to Many Bidirectional (1)

What is “bidirectional”?



There is **imbalance** in the amount of information seen by the hidden states at different time steps.

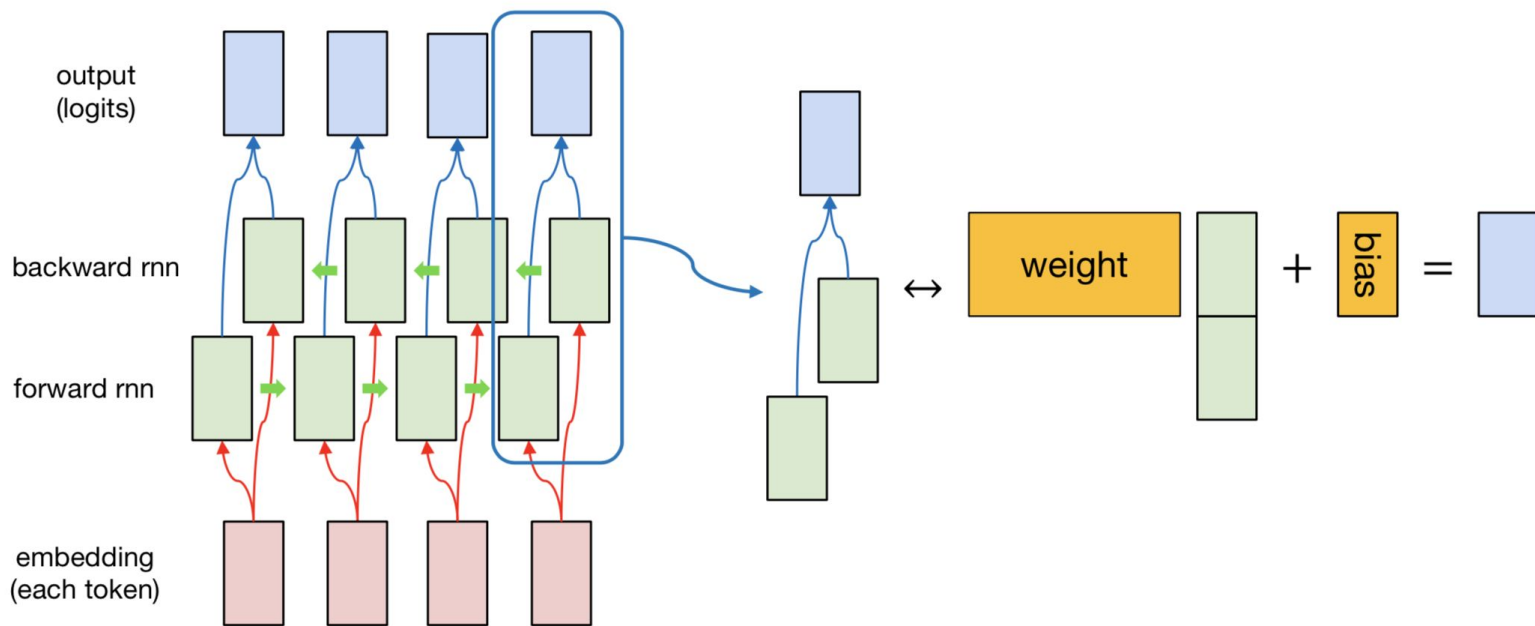
VS



There is **balance** in the amount of information seen by the hidden states at different time steps.

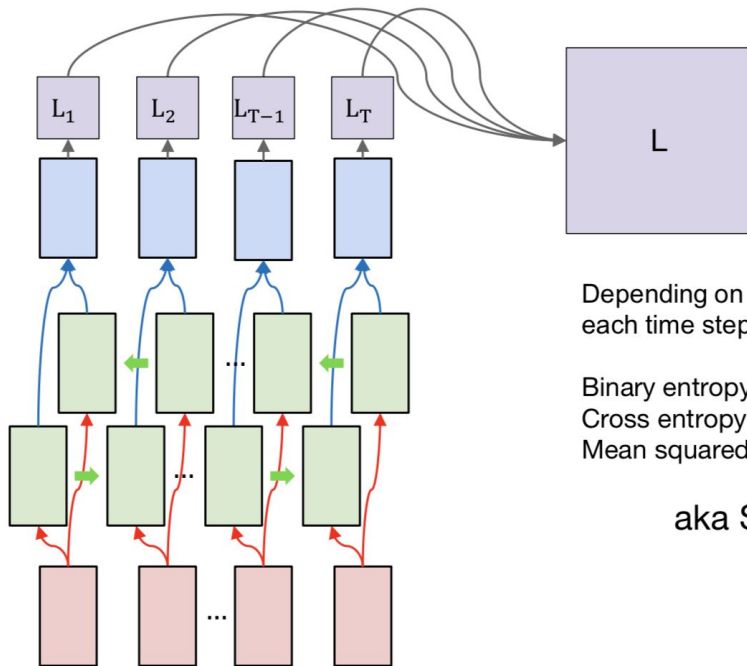
RNN : Many to Many Bidirectional (2)

What is “bidirectional”?



RNN : Many to Many Bidirectional (3)

many to many bidirectional



Depending on the task with regard to each time step...

Binary entropy loss (binary classifier)
Cross entropy loss (softmax classifier)
Mean squared loss (regression)

aka Sequence loss!

RNN (Recurrent Neural Network)

딥러닝기반 자연어 처리

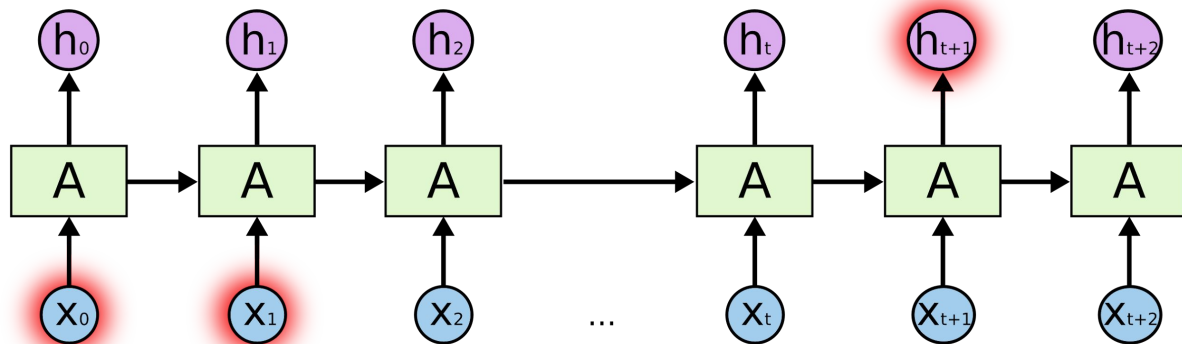
6

LSTM (Long short-term memory)



RNN의 문제점

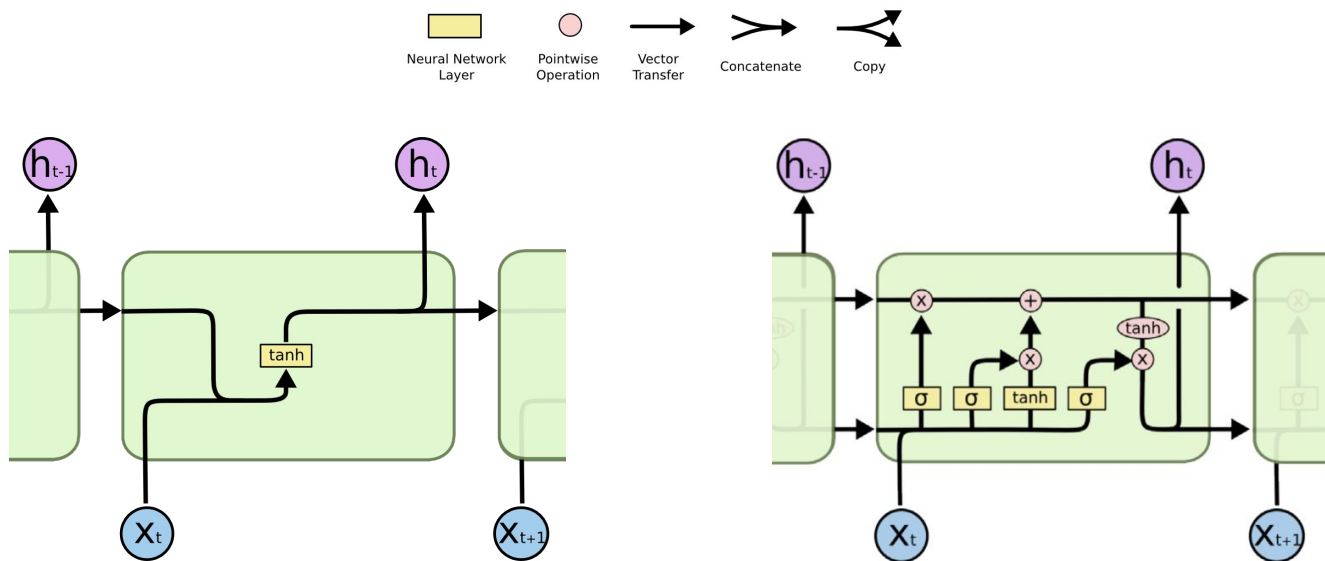
- RNN은 이전 셀의 hidden state를 전달 받음으로써 시계열 데이터 처리에 적합
- 장기 의존성(Long term dependency) 문제



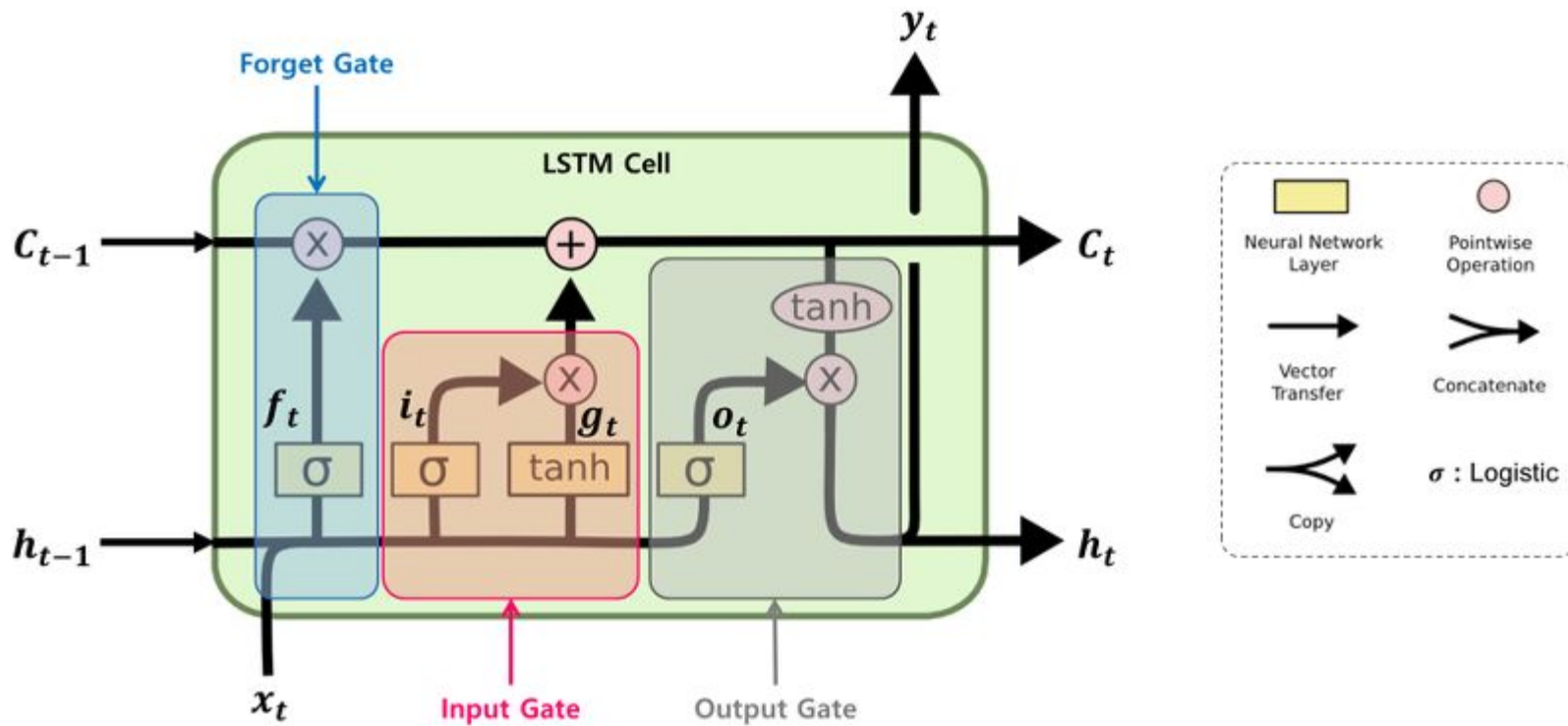
<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

LSTM (Long short-term memory)

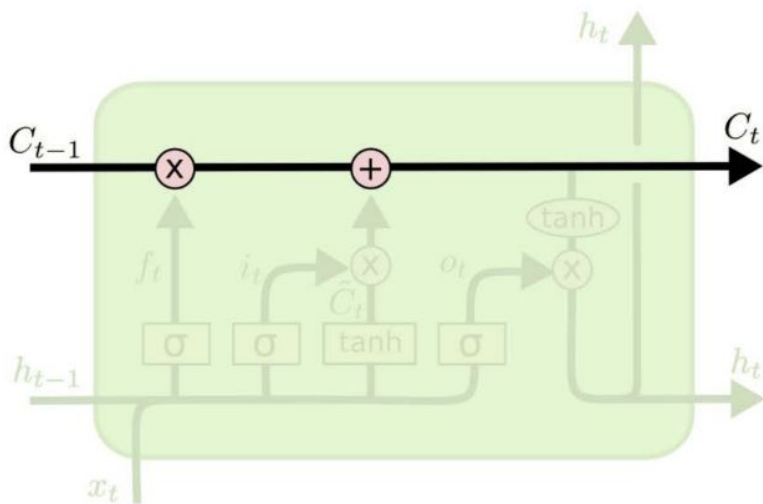
- 장기 단기 메모리 네트워크는 장기적인 종속성을 학습 할 수있는 특수한 종류의 RNN입니다



LSTM 구조 - Cell State

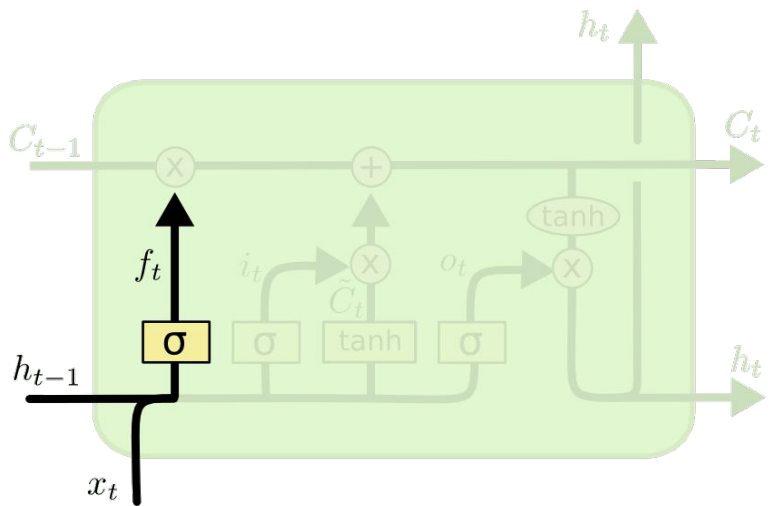


LSTM 구조 - Cell State



- LSTM의 핵심은 Cell state
- 이전 정보를 유지하기 용이
- LSTM은 Input gate를 활용하여 Cell state에 대한 정보를 잊거나 추가 할 수 있음
 - 게이트는 선택적으로 정보를 전달할 수있는 방법
- 시그모이드 레이어는 0에서 1 사이의 숫자를 출력하여 각 구성 요소를 얼마나 많이 통과시켜야하는지 결정
 - 값이 0 : "아무것도 통과하지 않음"
 - 값이 1 : "모든 것을 통과"

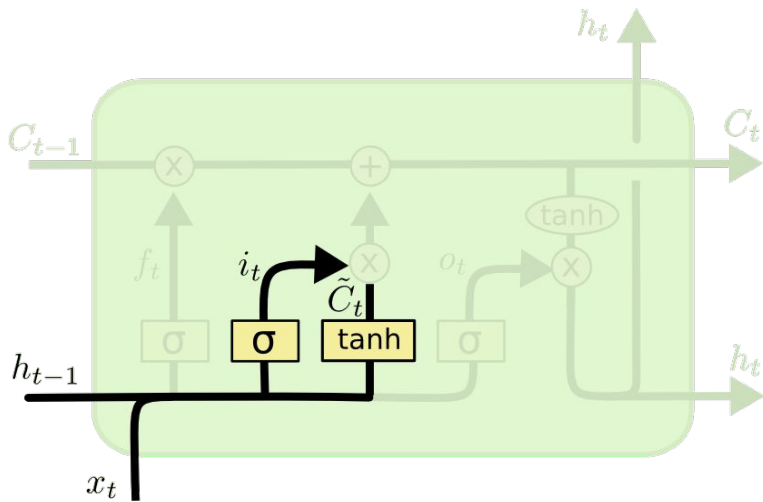
LSTM 구조 - Forget Gate



$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)$$

- Cell state의 정보를 버릴지 유지할 지 결정
- Sigmoid를 활용하여 1의 경우 유지, 0인 경우 제거를 의미

LSTM 구조 - Input Gate

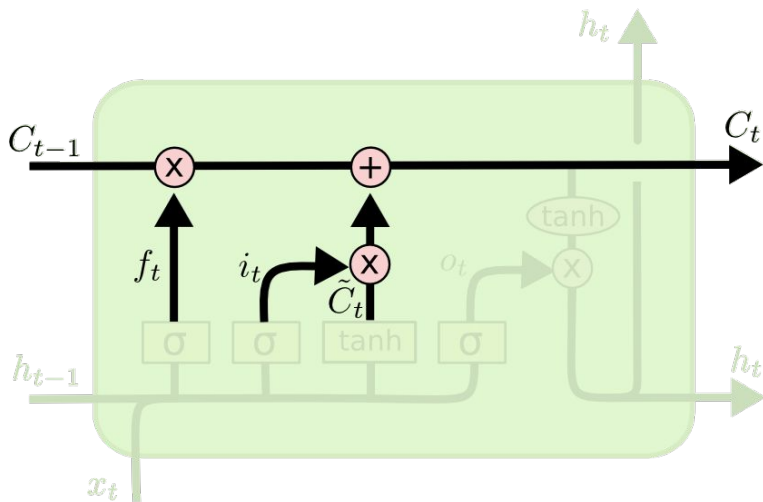


- Cell state에 새로운 정보를 추가할 것인지 결정
- Sigmoid : Cell state에 새로운 정보를 얼마나 추가할 것인지 결정
- Tanh : Cell state에 추가할 새로운 정보

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

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

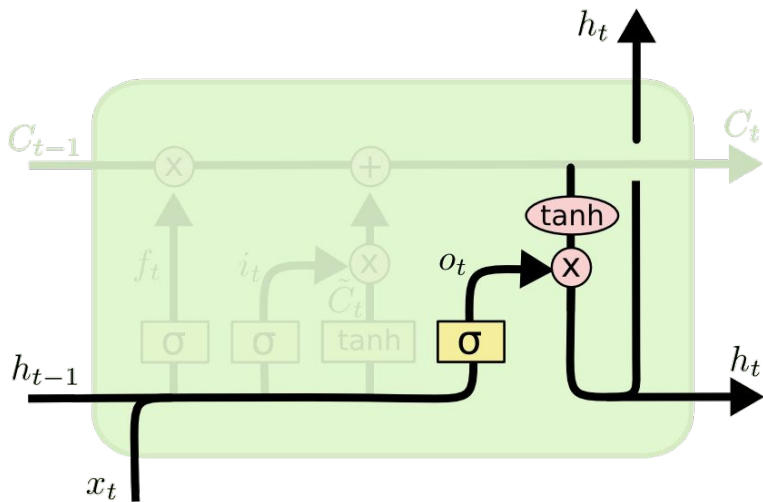
LSTM - Cell State



- Forget gate : 이전 Cell state를 얼마나 유지할지 적용
- Input gate : Cell state에 새로운 정보를 얼마나 추가할지 적용

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

LSTM - Output gate



- Cell state 를 얼마나 정보로 출력할지 sigmoid로 결정

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

$$h_t = o_t * \tanh(C_t)$$

<https://github.com/tensorflow/nmt>

RNN (Recurrent Neural Network)

딥러닝기반 자연어 처리

7

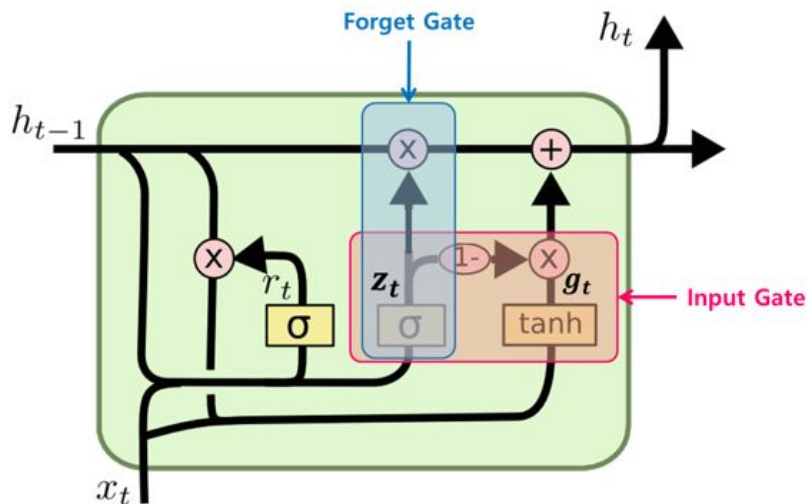
GRU (Gated Recurrent Unit)



GRU (Gated Recurrent Unit)

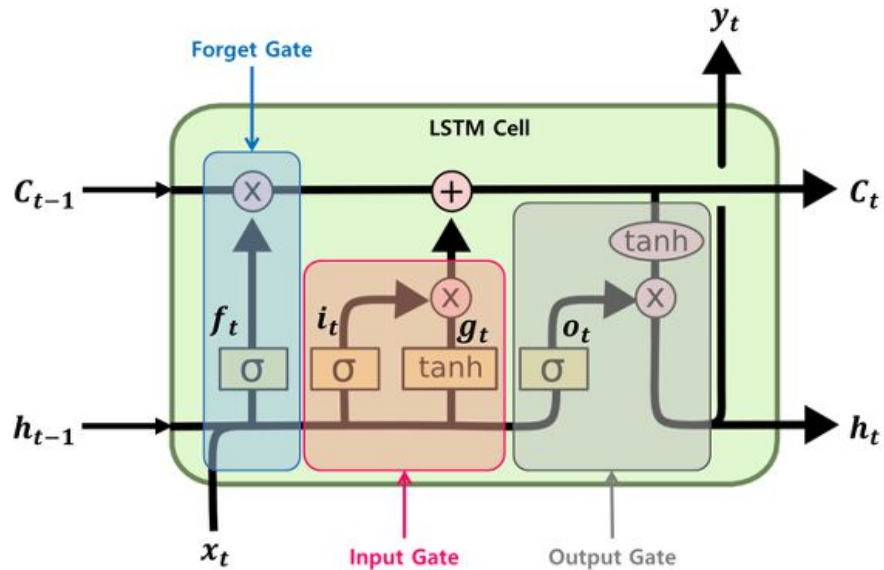
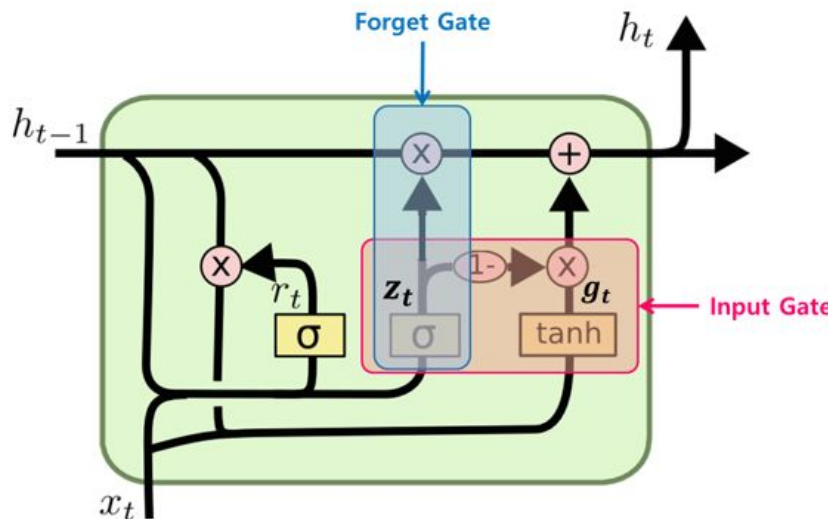
GRU(Gated Recurrent Unit) 셀은 2014년에 K. Cho(조경현) 등에 의해 제안된 LSTM 셀의 간소화된 버전

<https://arxiv.org/pdf/1406.1078v3.pdf>



GRU (Gated Recurrent Unit)

LSTM과 비교하면 유사함을 확인할 수 있음



$$\begin{aligned}
 r_t &= \sigma(W_{xr}^T \cdot x_t + W_{hr}^T \cdot h_{t-1} + b_r) \\
 z_t &= \sigma(W_{xz}^T \cdot x_t + W_{hz}^T \cdot h_{t-1} + b_z) \\
 g_t &= \tanh(W_{xg}^T \cdot x_t + W_{hg}^T \cdot (r_t \otimes h_{t-1}) + b_g) \\
 h_t &= z_t \otimes h_{t-1} + (1 - z_t) \otimes g_t
 \end{aligned}$$

RNN vs LSTM vs GRU

