

# 순환신경망

Recurrent Neural Network

DA 4기 정민화

2023.07.28 10:00 AM

# 1. 왜 RNN을 사용할까?

언어데이터의 특성:

기호로서의 언어는 표현하고자 하는 ‘내용’과 그것을 표현하는 ‘형식’으로 이루어져있다. 둘의 관계는 자의적이며, 이를 언어의 자의성이라고 한다.

# 1. 왜 RNN을 사용할까?

언어데이터의 특성:

기호로서의 언어는 표현하고자 하는 ‘내용’과 그것을 표현하는 ‘형식’으로 이루어져있다. 둘의 관계는 자의적이며, 이를 언어의 자의성이라고 한다.

# 1. 왜 RNN을 사용할까?

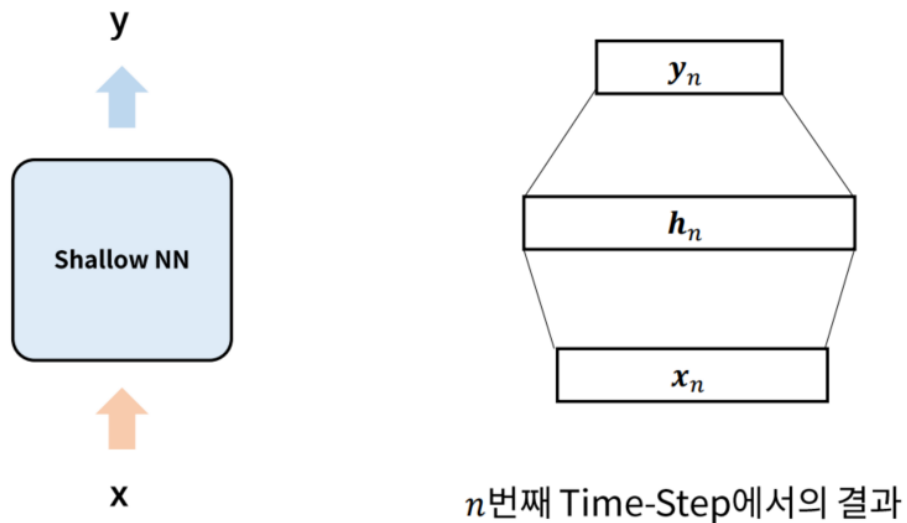
언어데이터의 특성: **순서가 있는 sequence data**

기호로서의 언어는 표현하고자 하는 ‘내용’과 그것을 표현하는 ‘형식’으로 이루어져있다. 둘의 관계는 자의적이며, 이를 언어의 자의성이라고 한다.

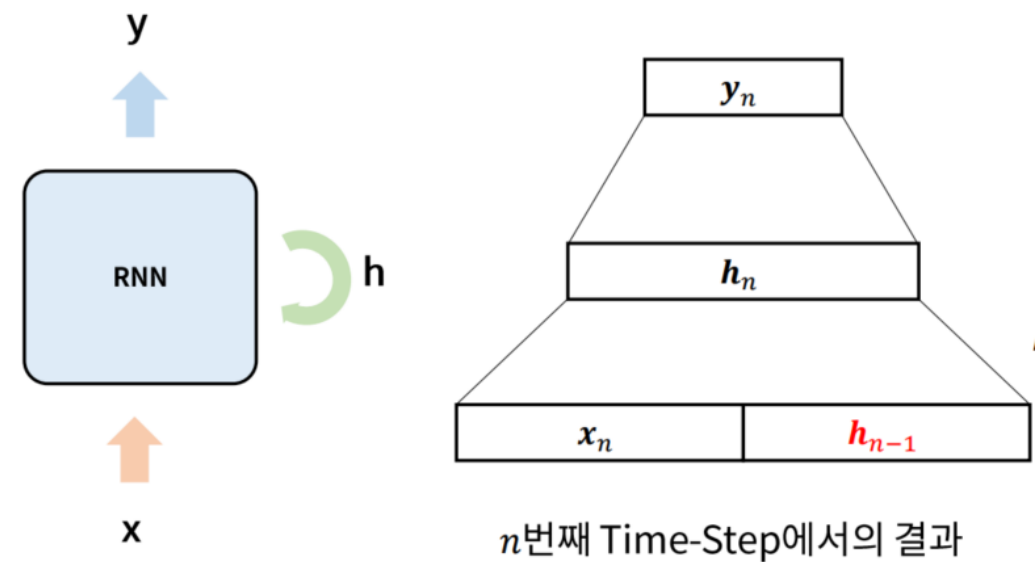
→ 순서가 의미가 있고, 순서가 달라질 경우 의미가 손상되는 데이터

## 2. Sequence는 RNN에 어떻게 반영될까?

얕은 신경망의 경우,  
출력층의 결과가 이전 입력에 영향을  
받지 않음

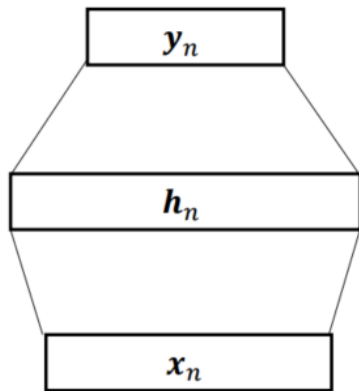


순환 신경망의 경우,  
얕은 신경망 구조에 '순환'이 추가



## 2. Sequence는 RNN에 어떻게 반영될까?

얇은 신경망의 경우,  
출력층의 결과가 이전 입력에 영향을  
받지 않음

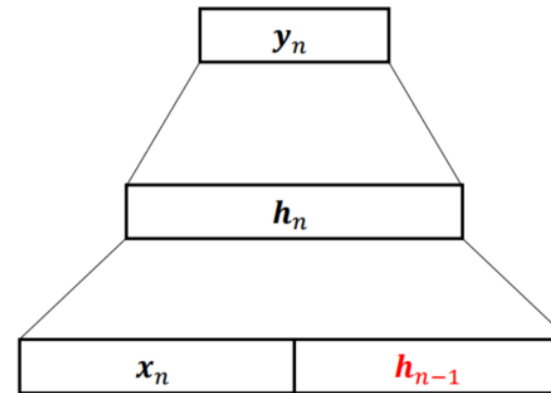


$$y_n = a(W_{hy}h_n + b_y)$$

$$h_n = a(W_{xh}x_n + b_h)$$

$n$ 번째 Time-Step에서의 결과

순환 신경망의 경우,  
얇은 신경망 구조에 ‘순환’이 추가

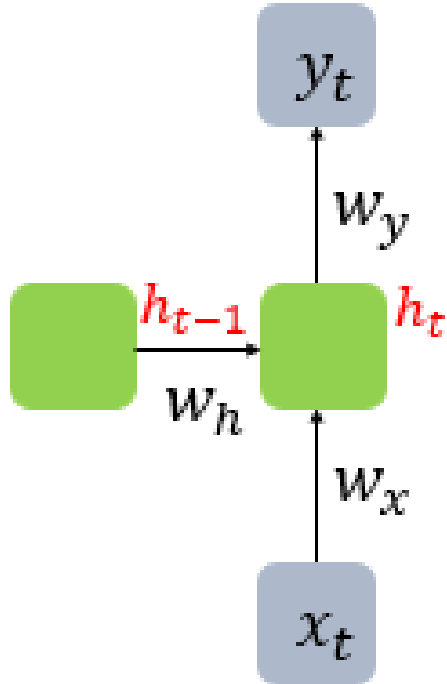


$$y_n = \tanh(W_{hy}h_n + b_y)$$

$$h_n = \tanh(W_{xh}x_n + W_{hh}h_{n-1} + b_h)$$

$n$ 번째 Time-Step에서의 결과

## 2-1) RNN의 구조



t시점(t번째 단어)에서

입력층:  $x_t$

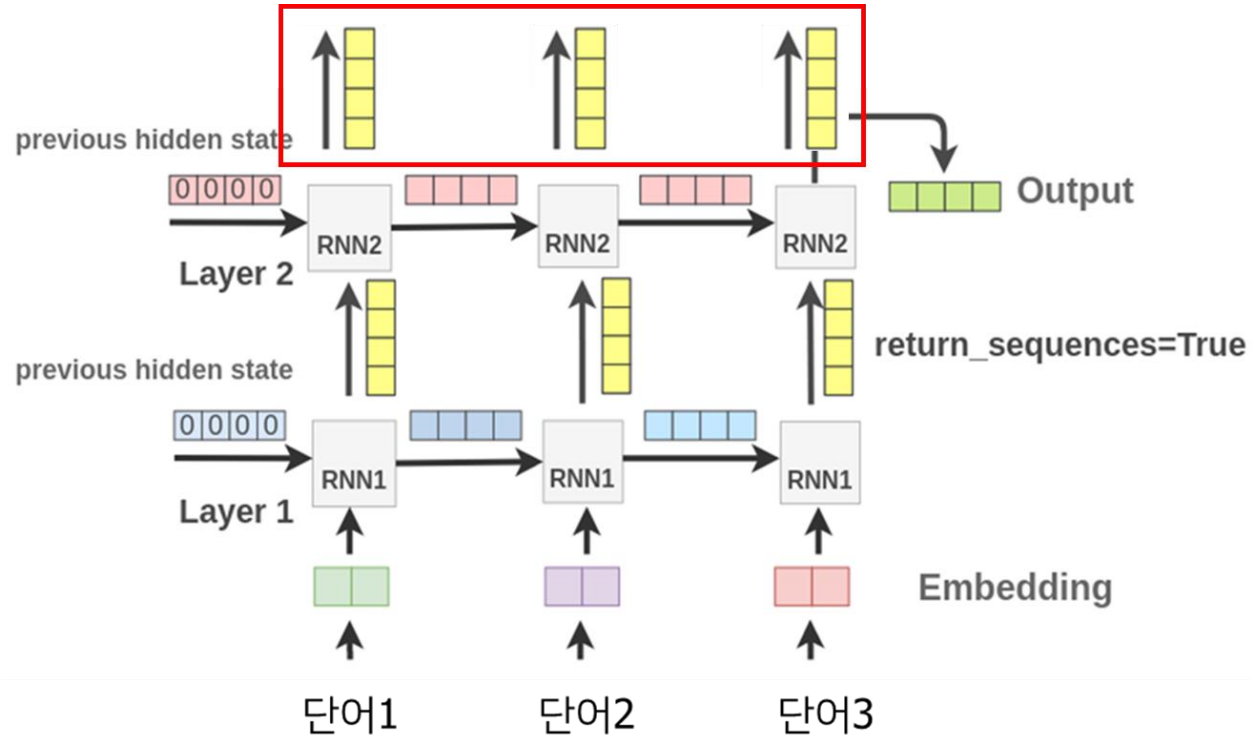
은닉층:  $h_t = \tanh(W_x x_t + W_h h_{t-1} + b)$

출력층:  $y_t = f(W_y h_t + b)$

(f는 활성화함수 중 하나)

\*Vanilla RNN은 활성화함수로  
하이퍼볼릭 탄젠트 함수를 활용

## 2-2) Stacked RNN

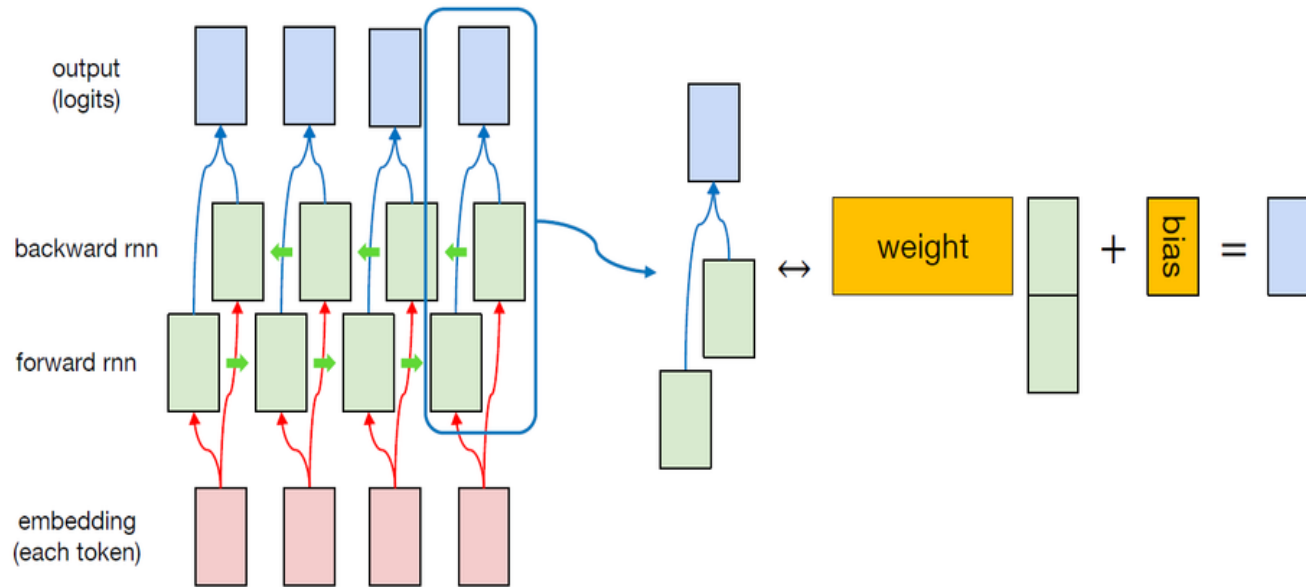


- RNN레이어를 쌓아 보다 모델에서 보다 깊은 신경망을 구성하는 방법
- **return\_sequences = True**로 설정하여 이전 rnn레이어의 은닉층 값이 다음 rnn레이어의 입력값이 될 수 있도록 해야함

\*\* time-step에서는 파라미터 공유, 레이어마다 다른 파라미터 사용 \*\*



## 2-3) Bidirectional RNN



- 전방향(forward)과 후방향(backward) 각각에 대한 hidden layer가 있어 인풋값에 대한 양방향 정보를 고려할 수 있는 모델

- 출력값은 모든 input이 적용된 후 전방향과 후방향 hidden layer값을 concat하여 연산함

$$\vec{h}_t = \sigma(W_{x\vec{h}}x_t + W_{\vec{h}\vec{h}}\vec{h}_{t-1} + b_{\vec{h}})$$

$$\overleftarrow{h}_t = \sigma(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}})$$

$$y_t = W_{\vec{h}y}\vec{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y$$

# 3. 실습: 한국어 데이터

## 활용 데이터: KLUE TC(Topic Classification)

- 자연어 이해 평가 데이터셋(Korean Language Understanding Evaluation)
- 2016년 1월부터 2020년 12월까지 네이버 뉴스에 올라간 연합뉴스의 헤드라인을 수집한 데이터.  
헤드라인이 어떤 토픽에 해당하는지 모델을 통해 분류

토픽종류: 7개 (**정치**, **경제**, 사회, 생활문화, 세계, IT과학, **스포츠**)

\*\* 사전 전처리 \*\*

- 7개의 토픽 중 3개(정치,경제,스포츠) 선정 → 다중분류 문제로 변환
- 한자가 포함되어있어 [hanja]라이브러리를 통해 한글로 변환

# Reference

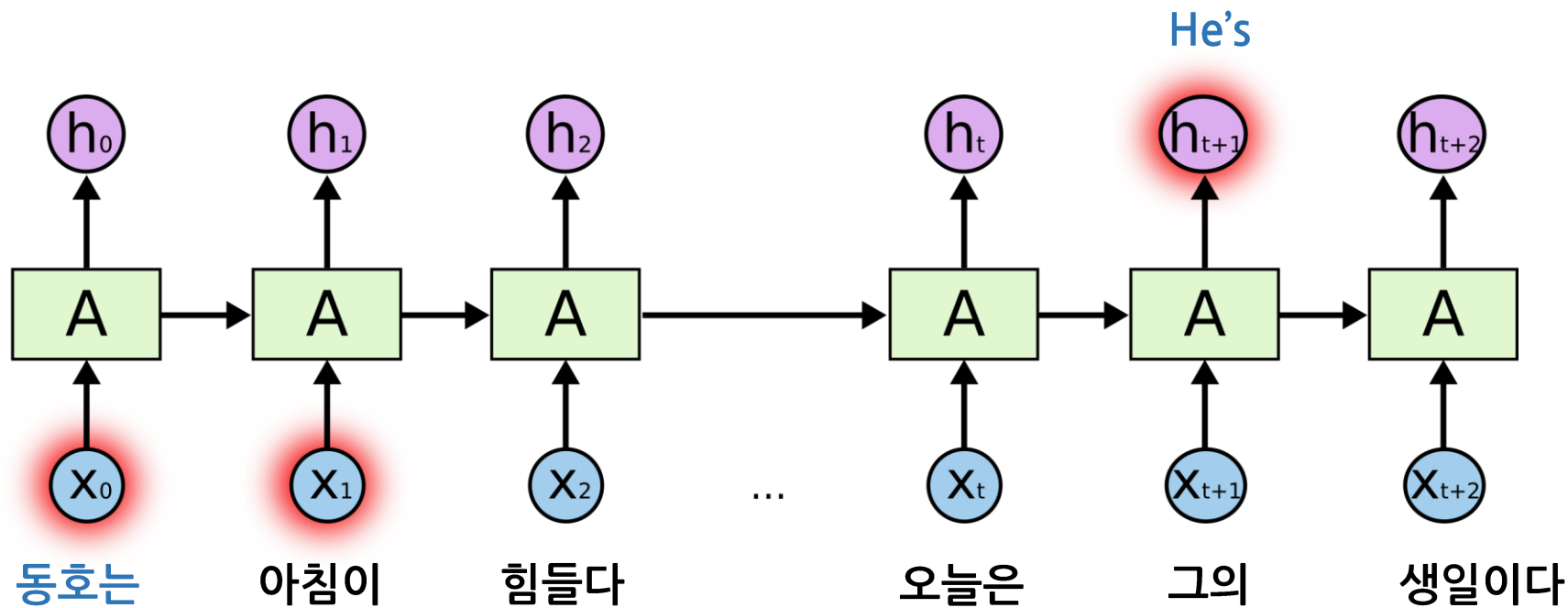
- <https://wikidocs.net/22886>
- <https://yeong-jin-data-blog.tistory.com/entry/Stacked-RNN>

# RNN Variants

DA 4기 정민화  
2023.07.28

# 1. RNN의 한계

- 장기의존성(long term dependency)문제

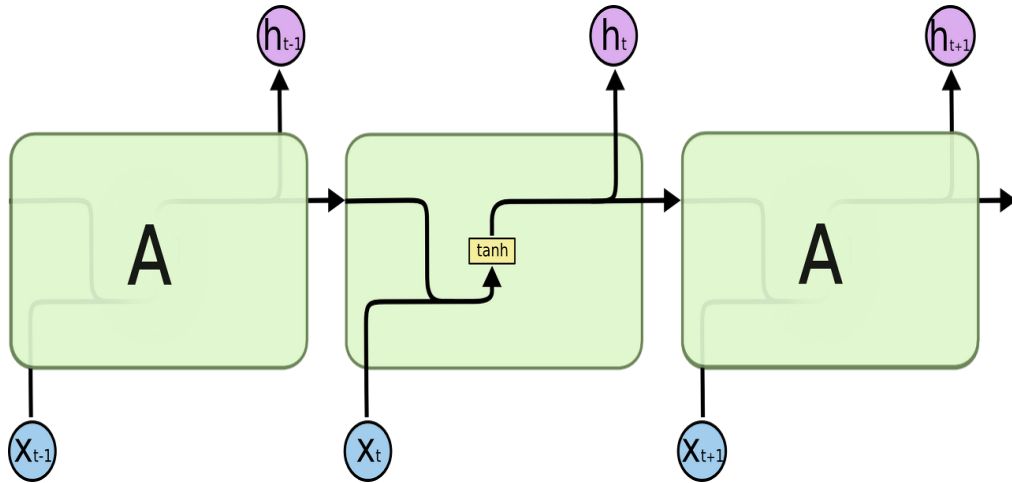


어떤 입력의 정보가 사용되는 시점(time step)이 해당 정보가 입력된 시점과 차이가 많이 날 경우 학습능력이 저하됨

# 1. RNN 과 LSTM의 비교

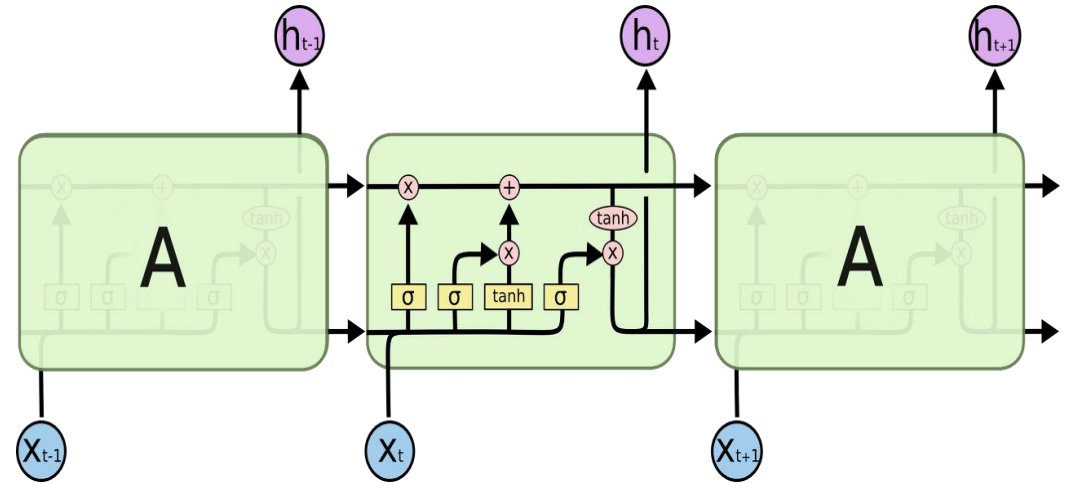
RNN의 경우,

순환되는 신경망이 하나의  
레이어만을 포함함

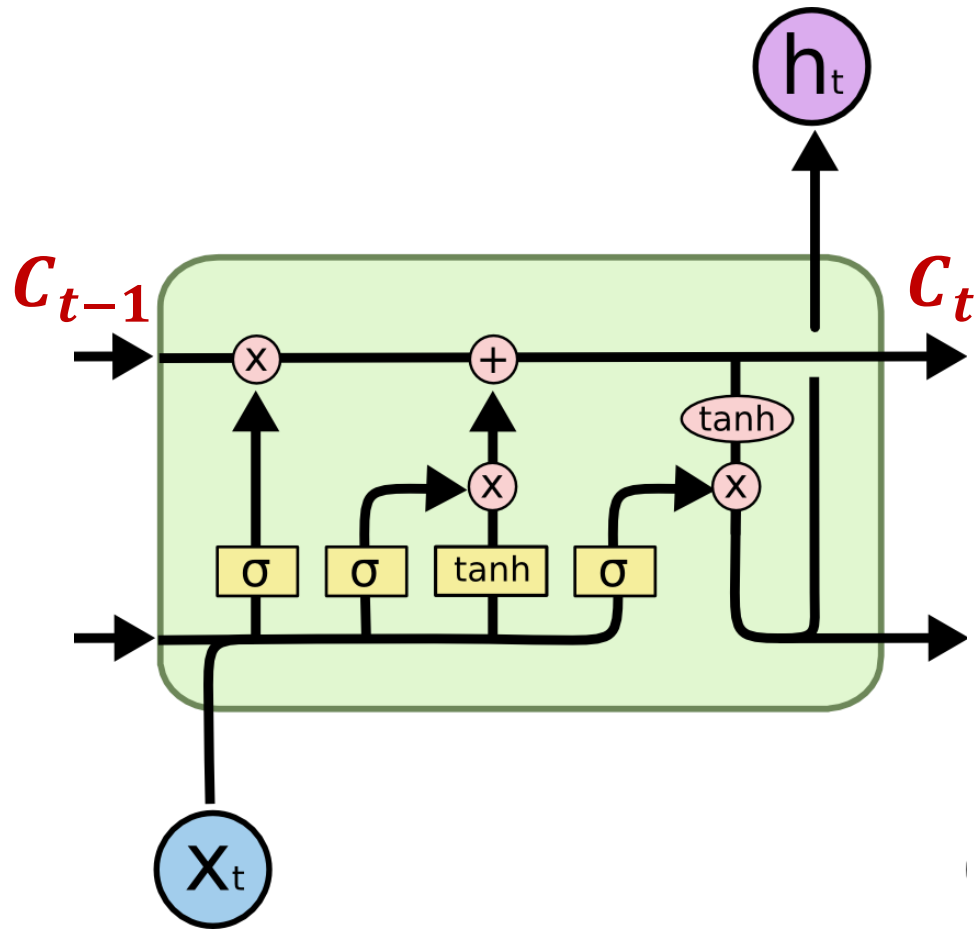


LSTM의 경우,

Gate를 통해 기억을 제거 혹은 더하는  
상호작용 가능



## 2. LSTM의 구조(1997)



### Cell state( $C_t$ )

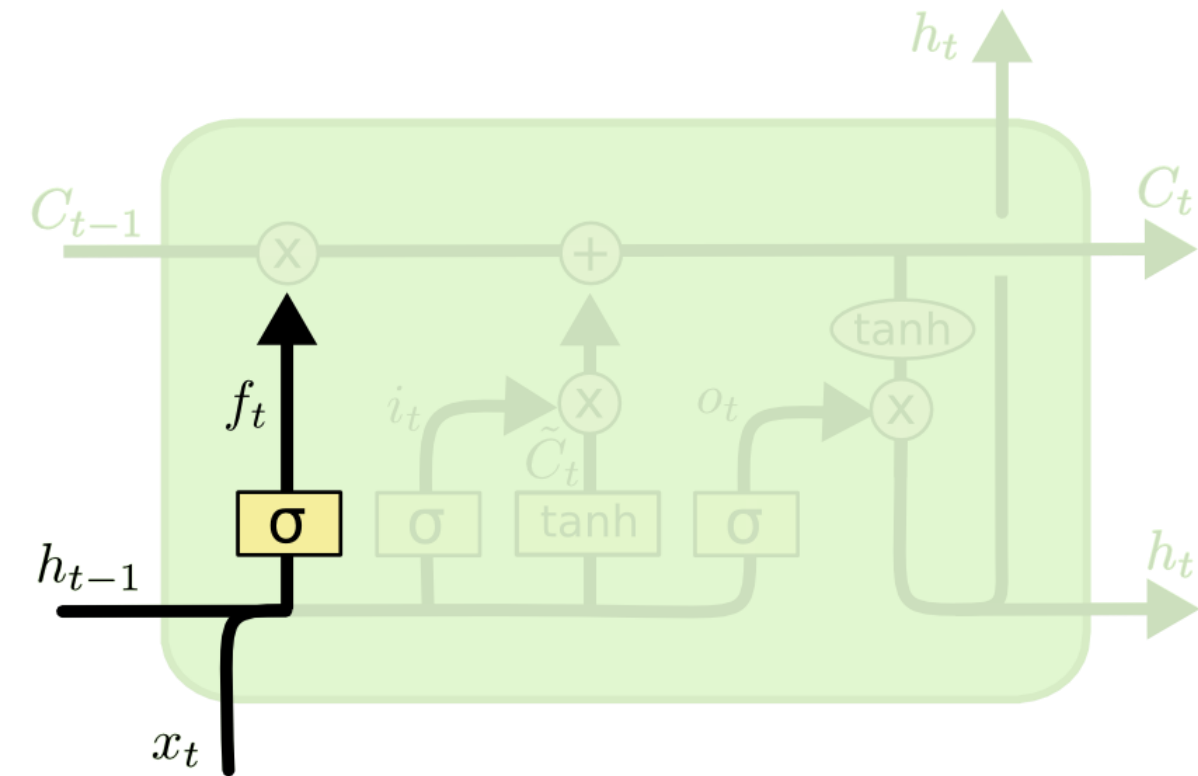
기억을 오랫동안 유지할 수 있는 구조  
새로운 특징을 덧셈연산으로 받음

### Hidden state( $h_t$ )

계층의 출력/다음 타임 스텝으로 넘기는 정보

LSTM은 RNN과 달리 cell state가 있어 '기억'에  
대한 부분을 전담

# LSTM - forget gate



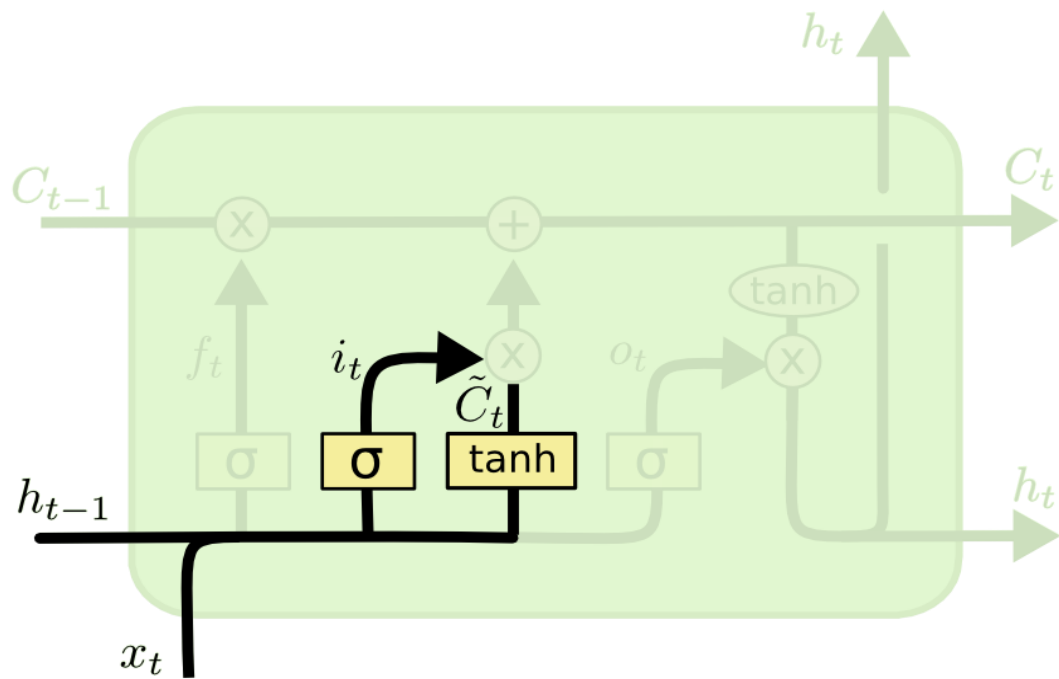
- 출력값이 활성화함수로 sigmoid를 사용하기 때문에 0~1
- Forget gate의 출력값은 cell state에 곱해져 이전의 정보를 ‘얼만큼 잊을지’ 결정함

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

$$f_t = \sigma(W_{hf}x_t + W_{xf}h_{t-1} + b_f)$$



## 2. LSTM - input gate



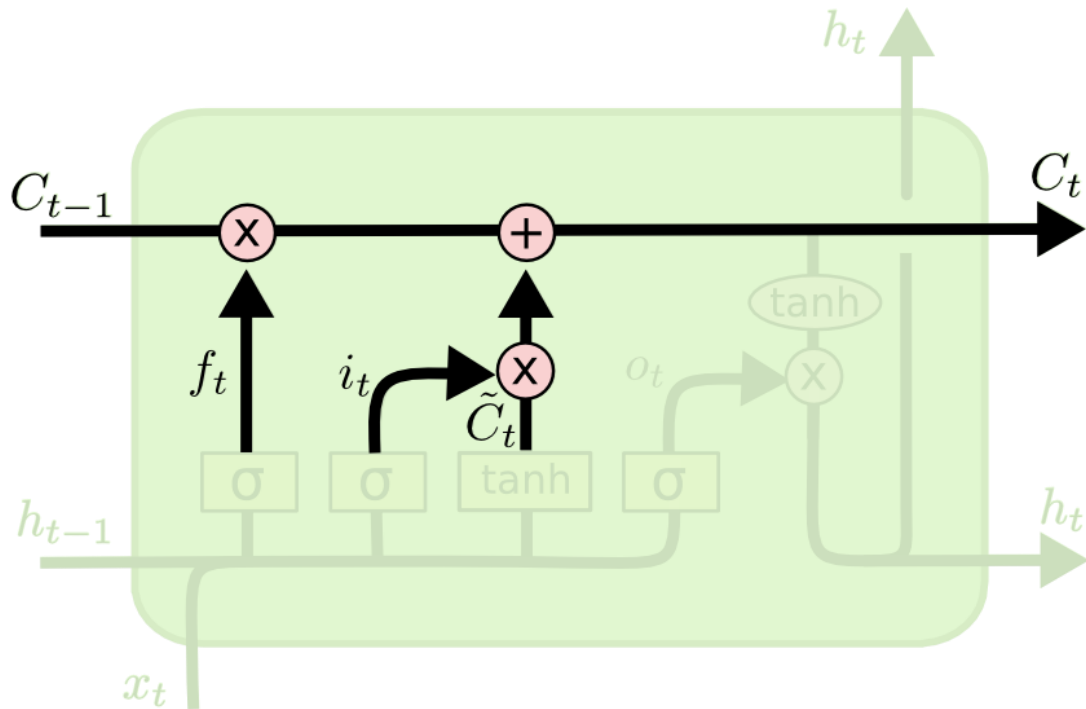
- Output gate와 마찬가지로 sigmoid 함수를 활성화함수로 사용(출력값 0~1)
- 새롭게 추출한 특징을 ‘얼만큼 사용할지’ 결정

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

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

\*\*  $\tilde{C}_t$ 는 RNN의 hidden state와 같은 연산을 수행

## 2. LSTM - cell state



- ‘기억’을 총괄
- Forget gate, input gate의 값을 종합하여 cell state를 업데이트함
- 아다마르곱 연산을 통해 각 gate값 반영

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

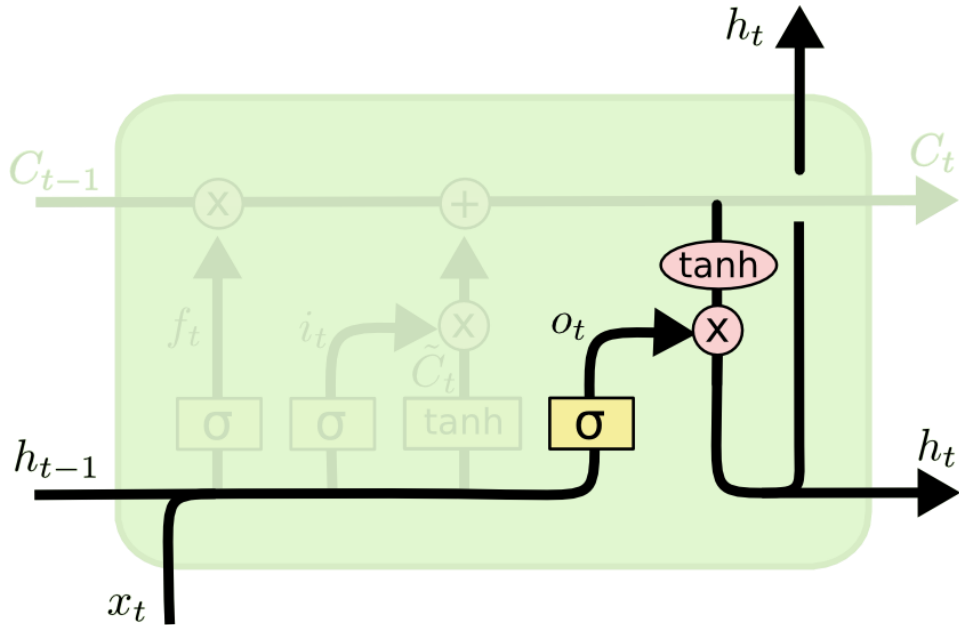
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

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

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

이전의 기억을 얼마나 잊을지, 새로운 기억을 얼마나 기억할지

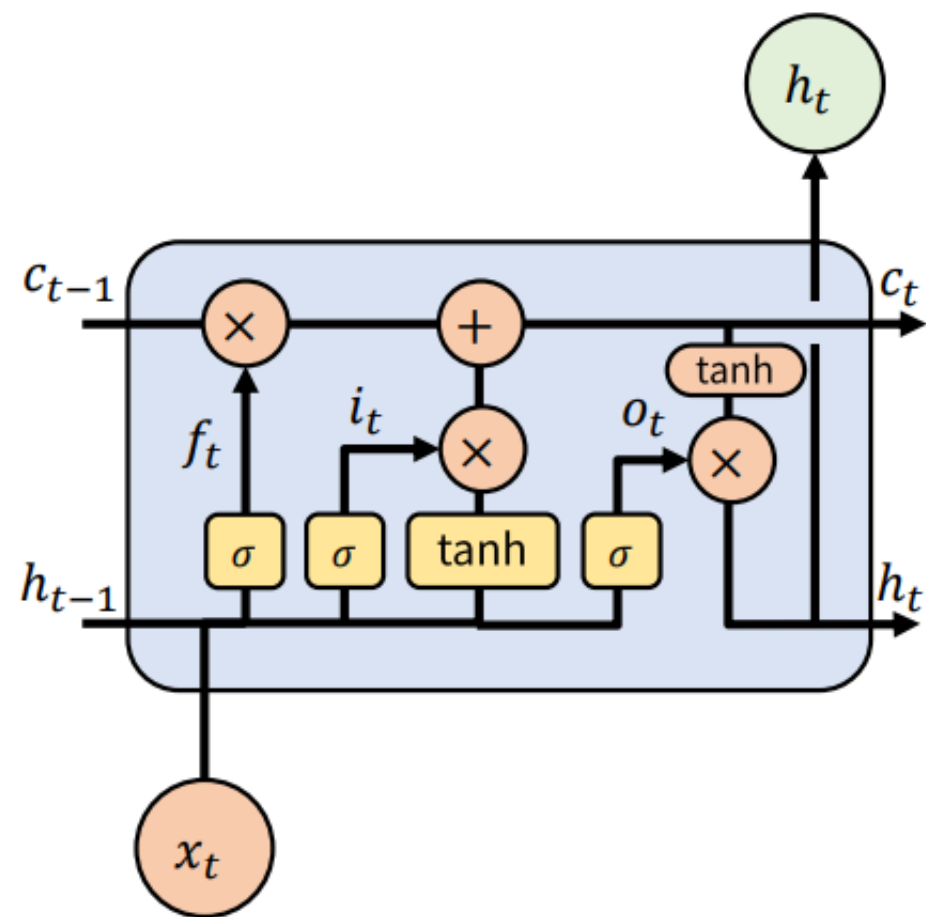
## 2. LSTM - output gate, hidden state



- Cell state 중 어떤 특징을 출력할지 결정
- 아다마르곱을 통해 cell state의 출력값을 반영

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

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



$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$

$$g_t = \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g)$$

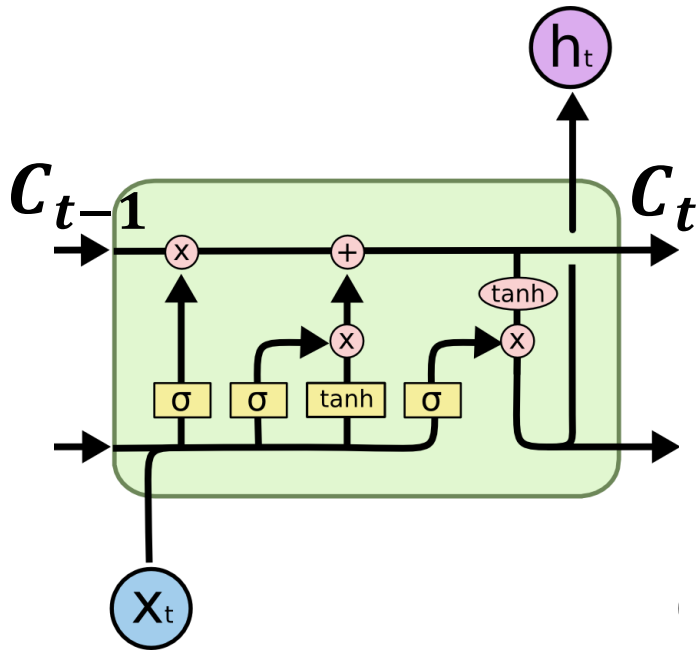
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$

$$h_t = o_t \odot \tanh(c_t)$$

### 3. GRU(Gated Recurrent Unit), 2014

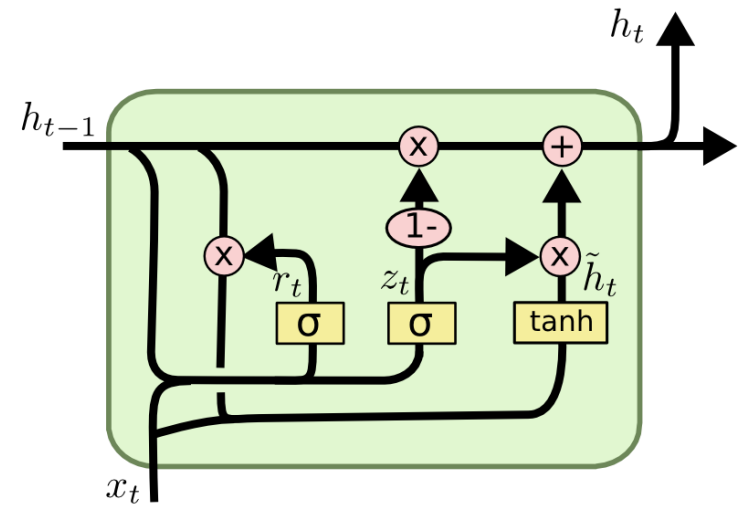
**LSTM**의 경우,

- Forget/input/output gate로 구성
- Cell state, hidden state로 구성

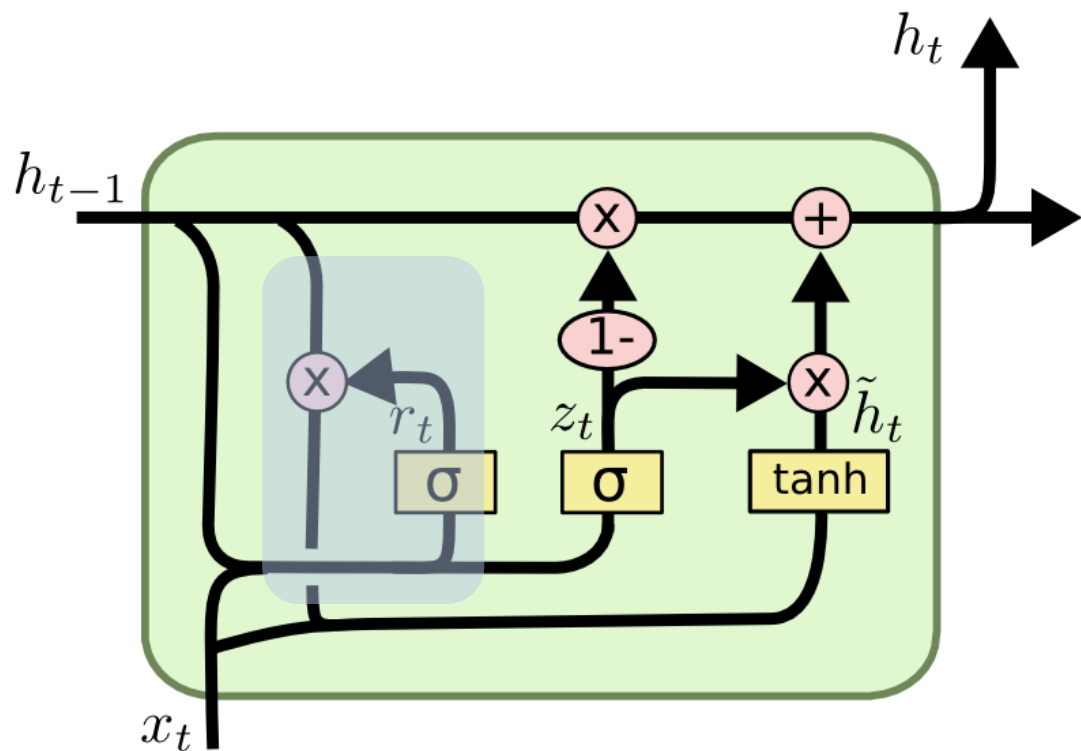


**GRU**의 경우,

- Reset gate, update gate로 구성 (update gate가 forget/input겸함)
- Hidden state로 합쳐짐



# GRU - reset gate



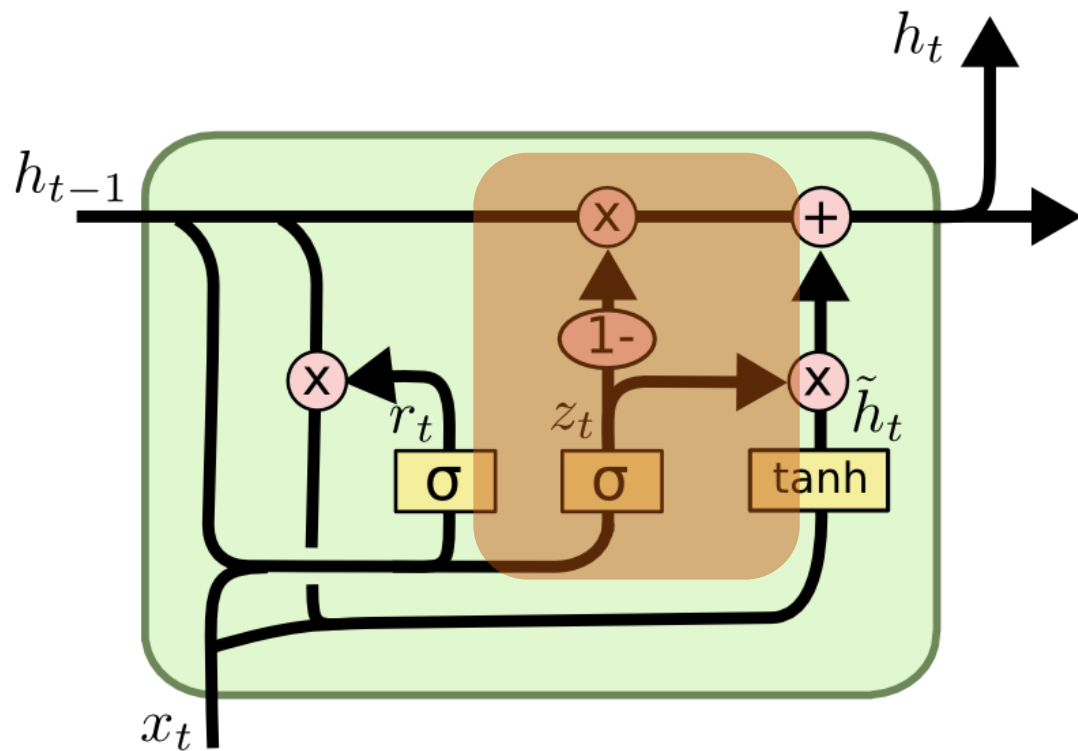
- 이전 hidden state( $h_{t-1}$ )값을 얼마나 잊어야하는지 결정

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

- 결과값은 update gate를 반영하기 전의 candidate hidden state에 반영

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

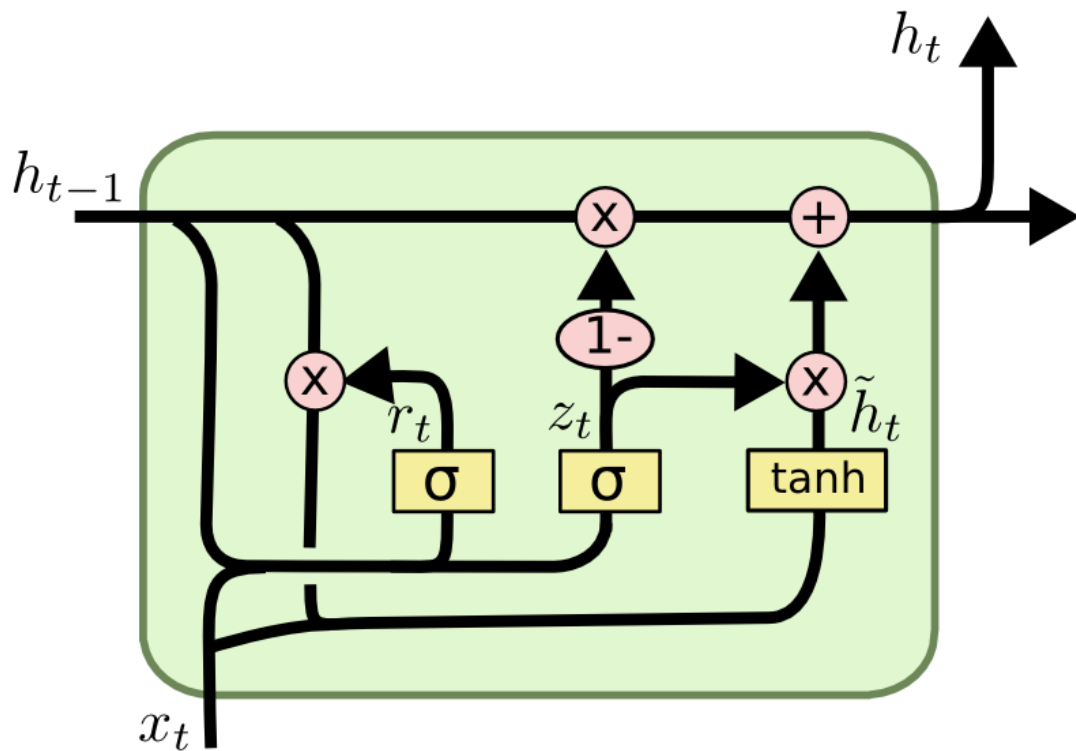
# GRU - update gate



- LSTM의 forget gate와 input gate를 겸함
- 이전의 정보를 얼마나 유지할지를 결정
- 과거와 현재의 정보를 각각 얼마나 반영할지에 대한 비율을 계산

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

# GRU - hidden state



- Update gate의 결과값( $z_t$ )는 새로운 hidden state( $h_t$ )가 이전 hidden state와 같은지 혹은 새로운 candidate hidden state을 많이 반영하는지 결정

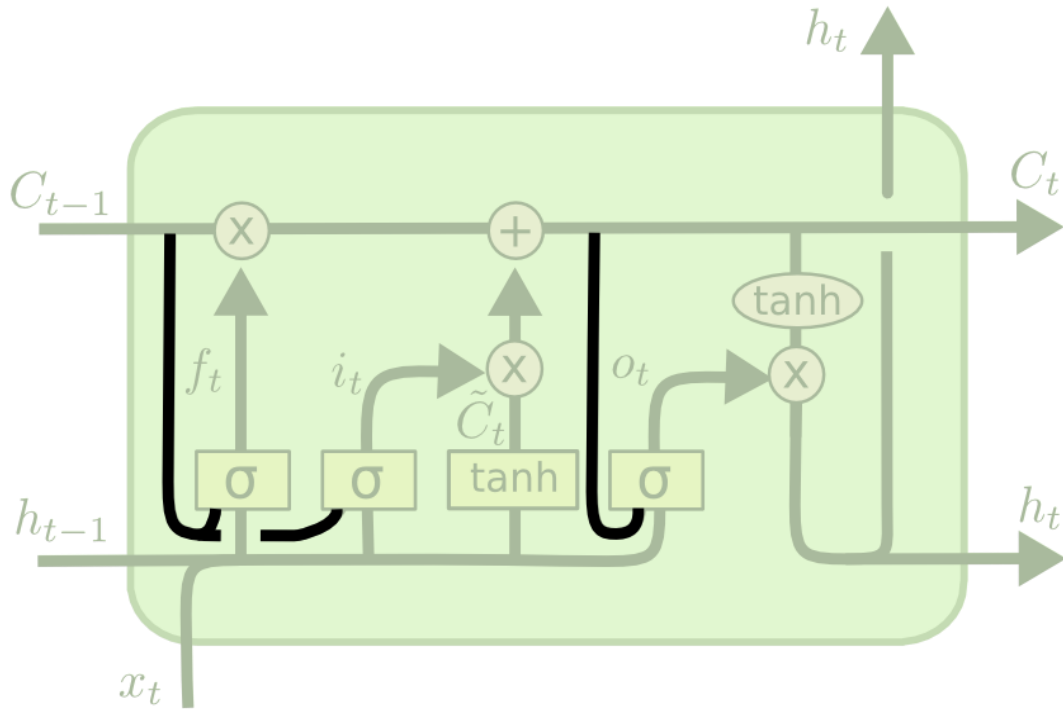
$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

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

과거의 hidden state를 얼마나 잊을지,  
현재의 candidate hidden state를 얼마나 반영할지  
(단, 잊어버린 만큼 채울 수 있게 함)



# 더 변형할 순 없을까?



- Peephole connection(2000)

Gate의 연산에 이전 cell state( $C_{t-1}$ )를 추가함

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

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

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

# 실습: 영어데이터(IMDB활용)

## 활용데이터: IMDB 리뷰 데이터셋

- 영화에 대한 리뷰 텍스트와 그에 대한 감성(긍/부)로 구성
- 훈련데이터와 테스트데이터셋 50:50으로 분할되어있음
- keras.datasets에 내장되어있음

## **\*\* 실습 관련 사항 \*\***

- LSTM과 GRU는 RNN모델에서 레이어만 바꿔주면 되는 간단한 코딩으로 모델 구성 가능
- 이에, return\_state와 return\_sequences인자를 활용하여 모델을 변형하는 실습 수행

# Reference

- LSTM: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/> (다수 참고)
- <https://diane-space.tistory.com/286> (번역본)
- GRU: [https://d2l.ai/chapter\\_recurrent-modern/gru.html](https://d2l.ai/chapter_recurrent-modern/gru.html)
- <https://hyen4110.tistory.com/26> (번역본)