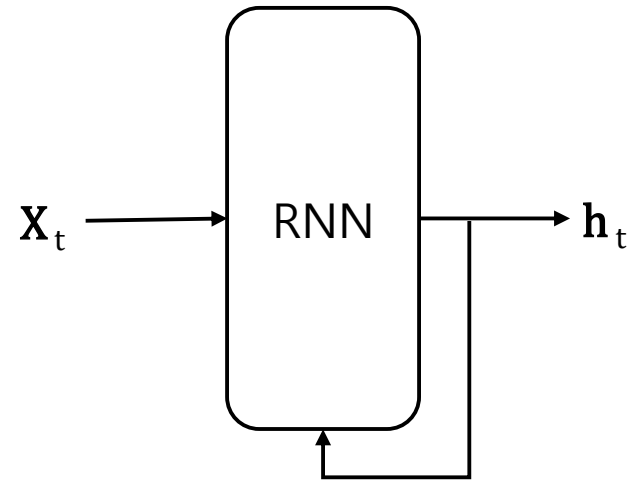


sh951011@gmail.com

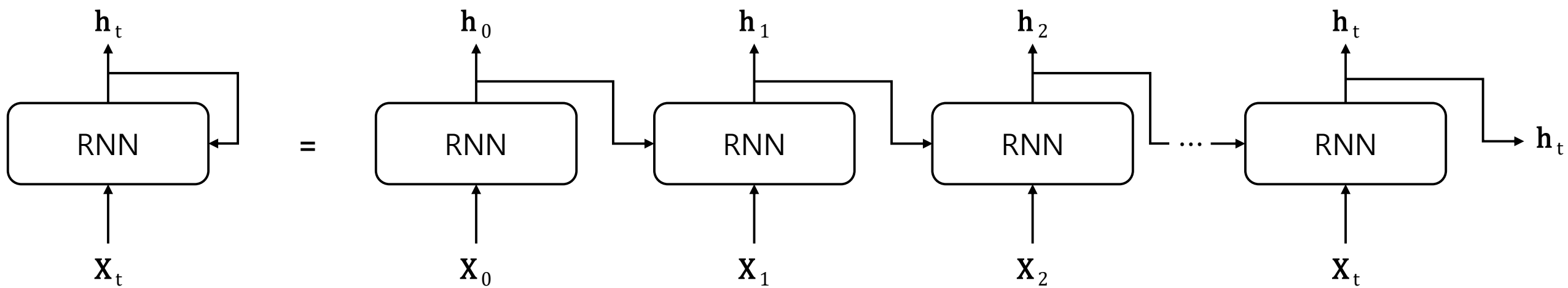
<https://github.com/sh951011>

RNN

Recurrent Neural Network (RNN)

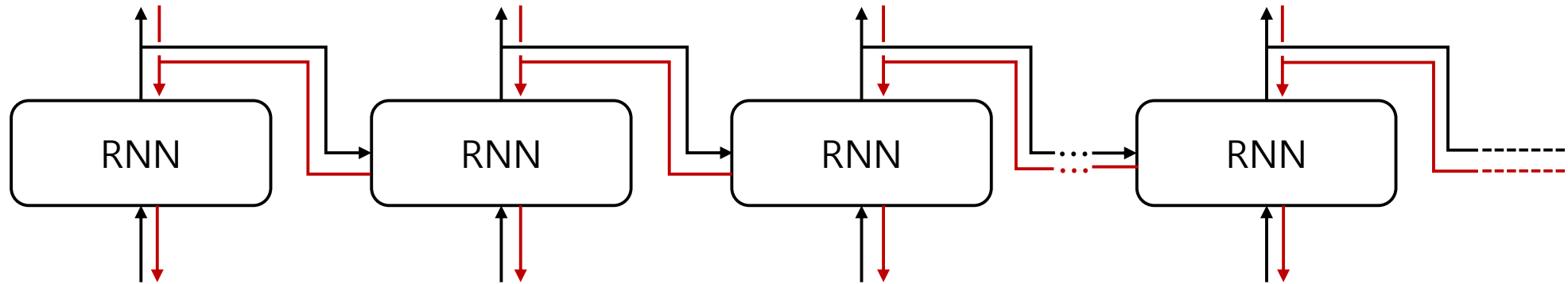


RNN 계층의 순환 구조 펼치기



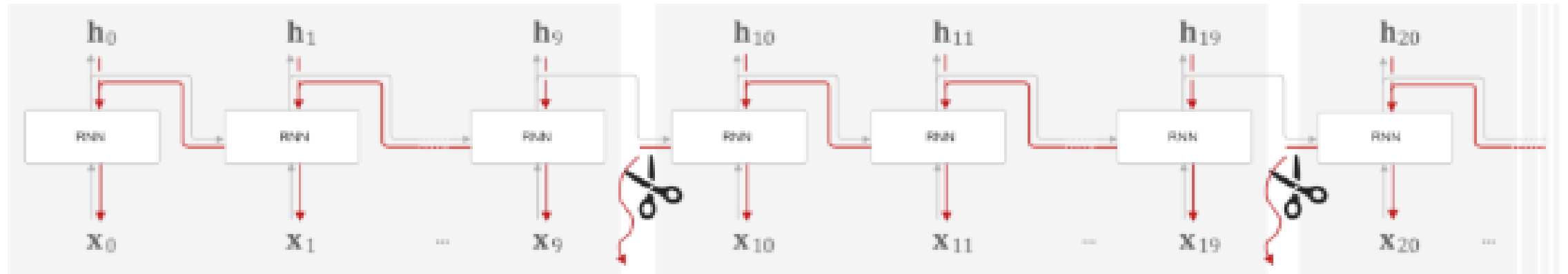
Recurrent Neural Network (RNN)

BPTT (Backpropagation Through Time)



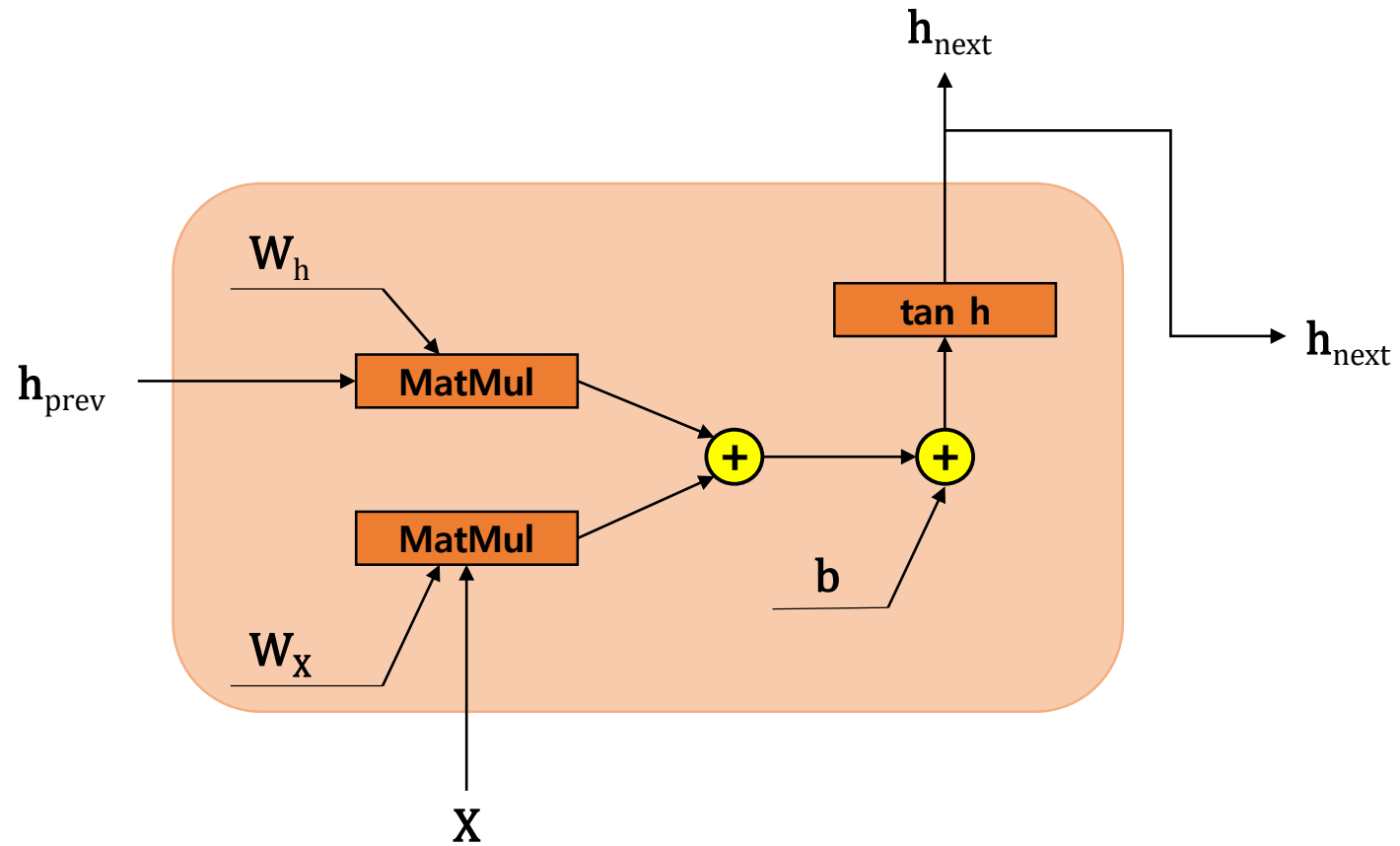
Recurrent Neural Network (RNN)

Truncated BPTT



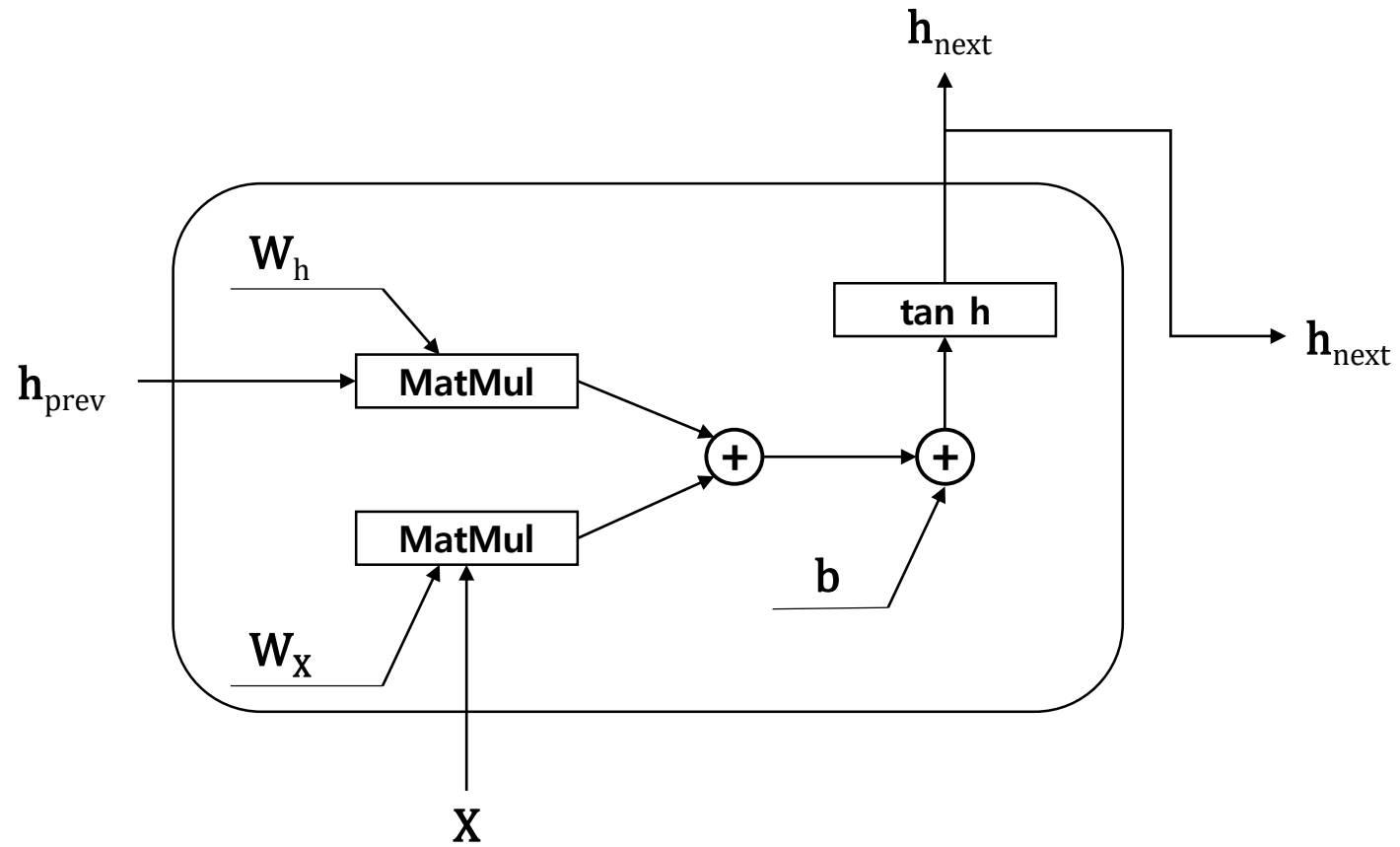
Recurrent Neural Network (RNN)

forward (순전파)



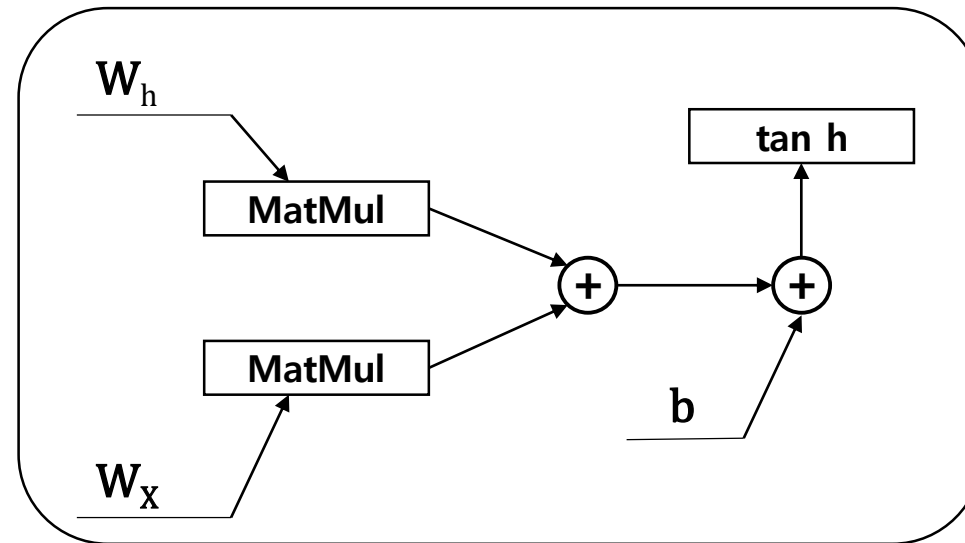
Recurrent Neural Network (RNN)

forward (순전파)



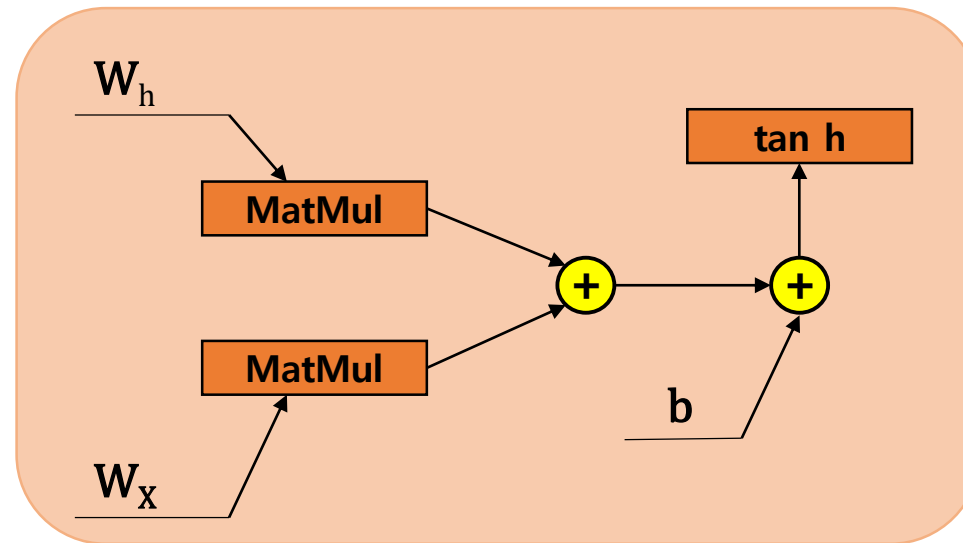
Recurrent Neural Network (RNN)

forward (순전파)



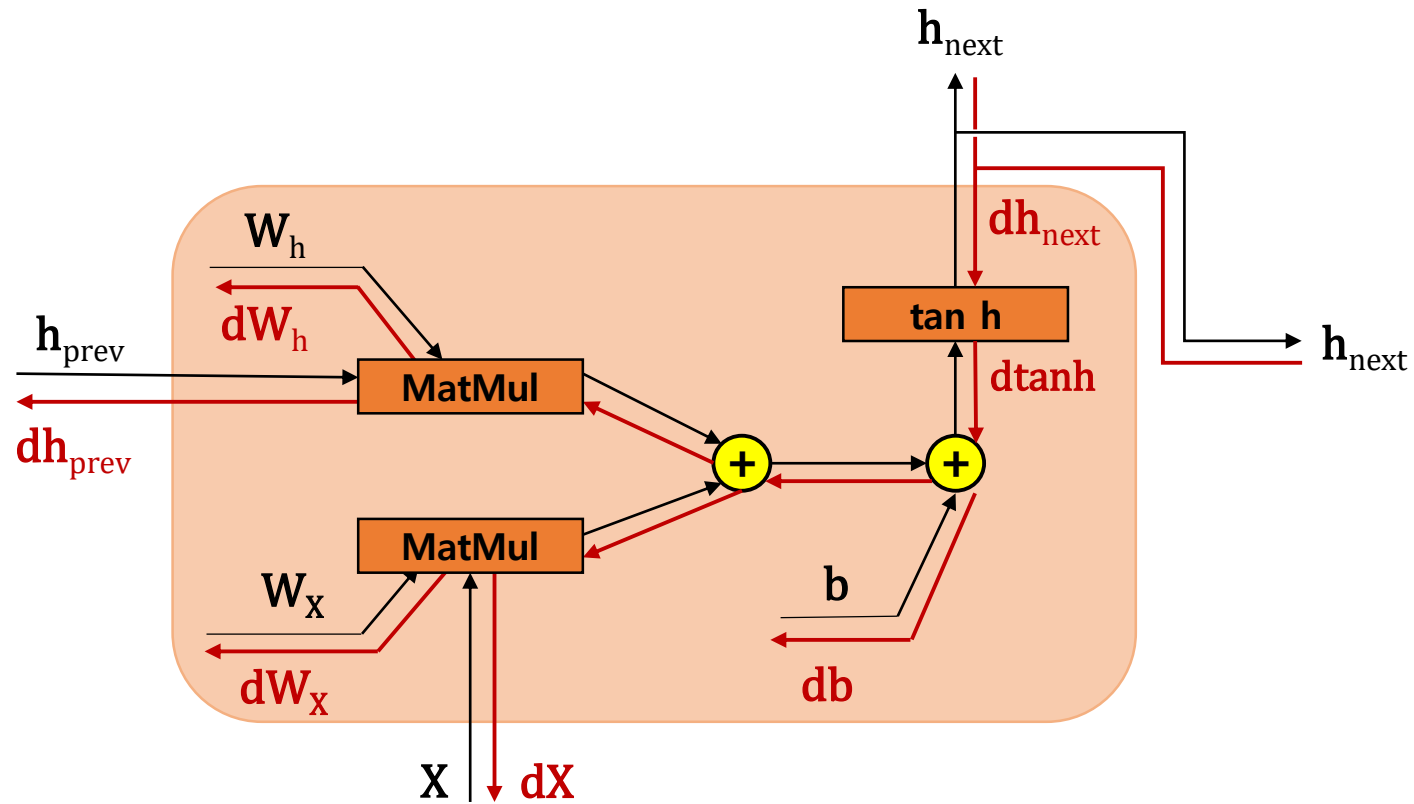
Recurrent Neural Network (RNN)

forward (순전파)



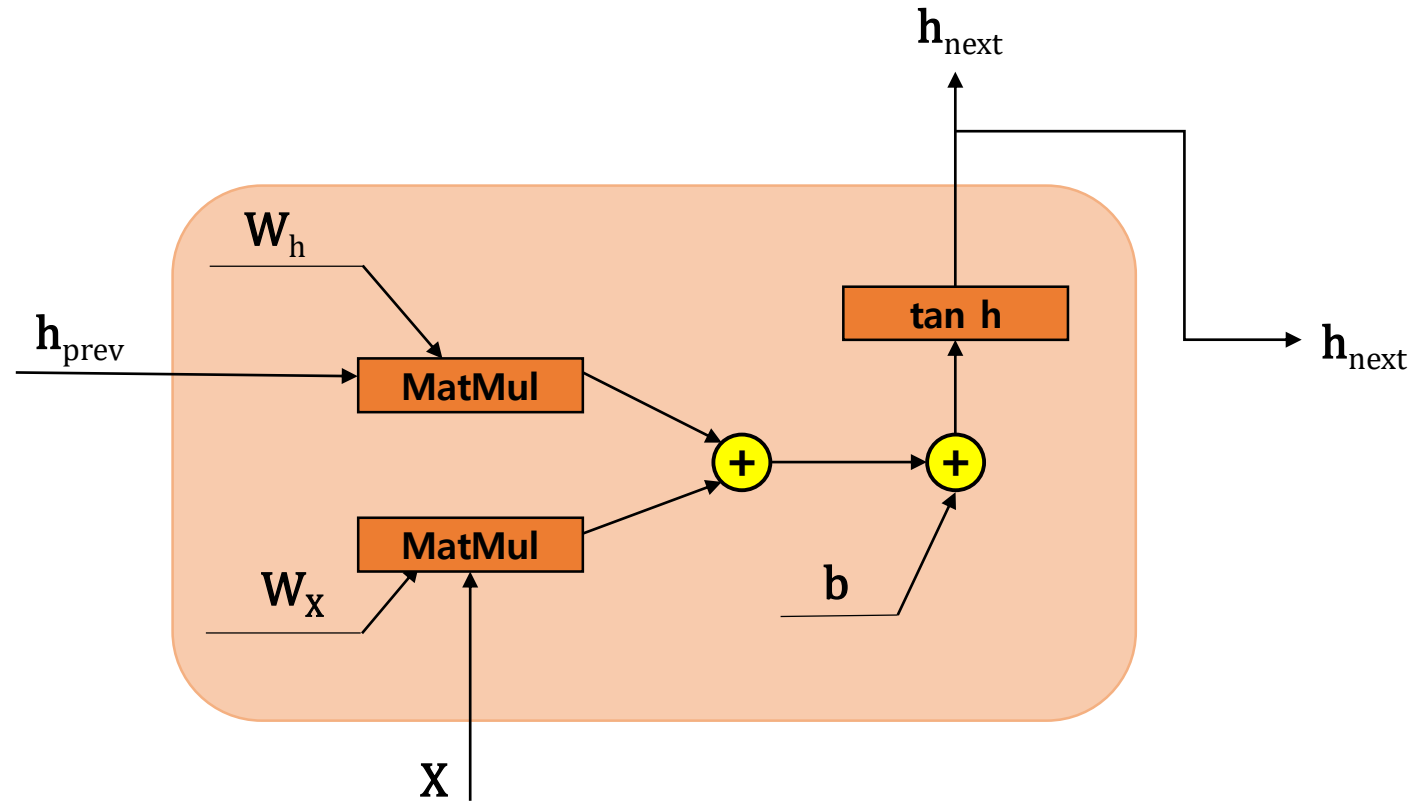
Recurrent Neural Network (RNN)

backward (역전파)



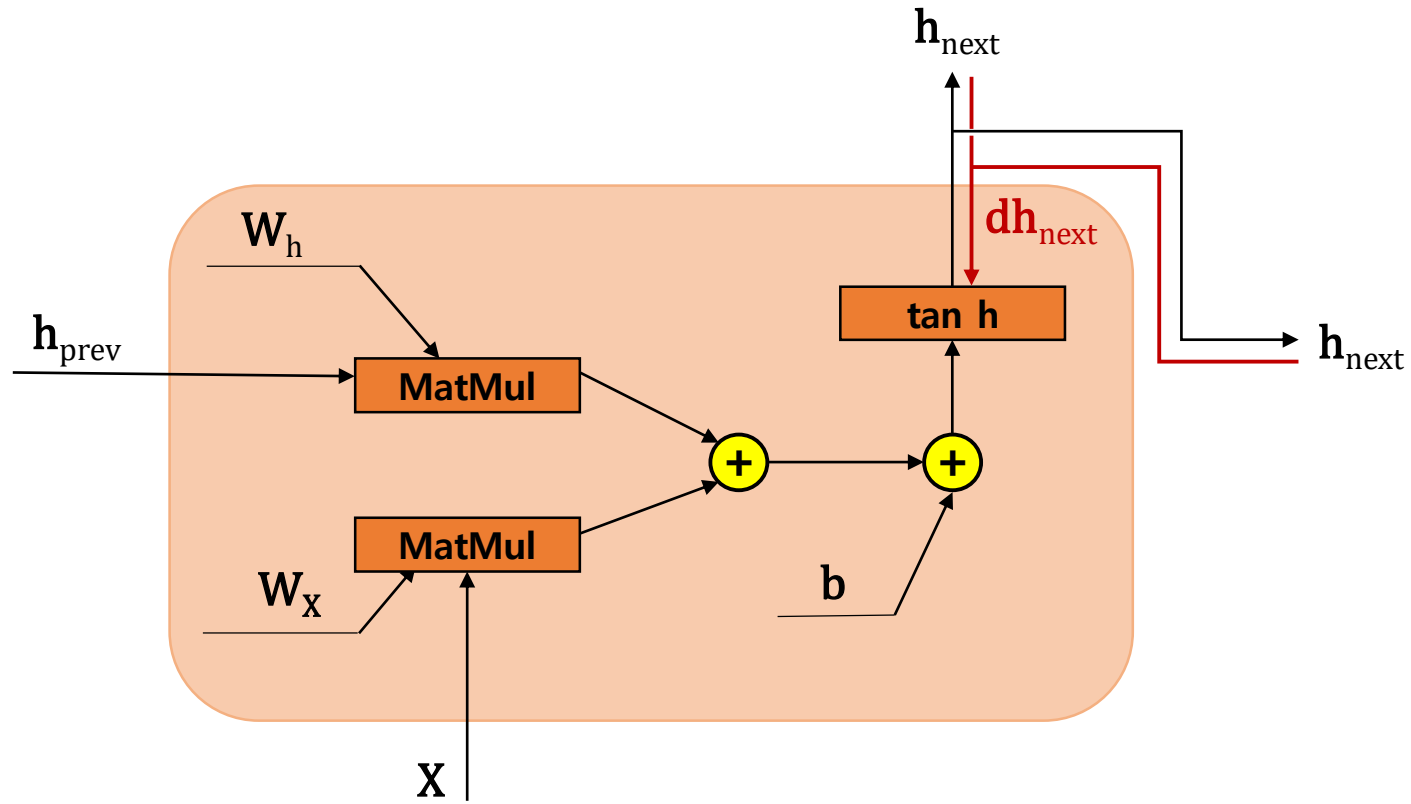
Recurrent Neural Network (RNN)

backward (역전파) - 시작



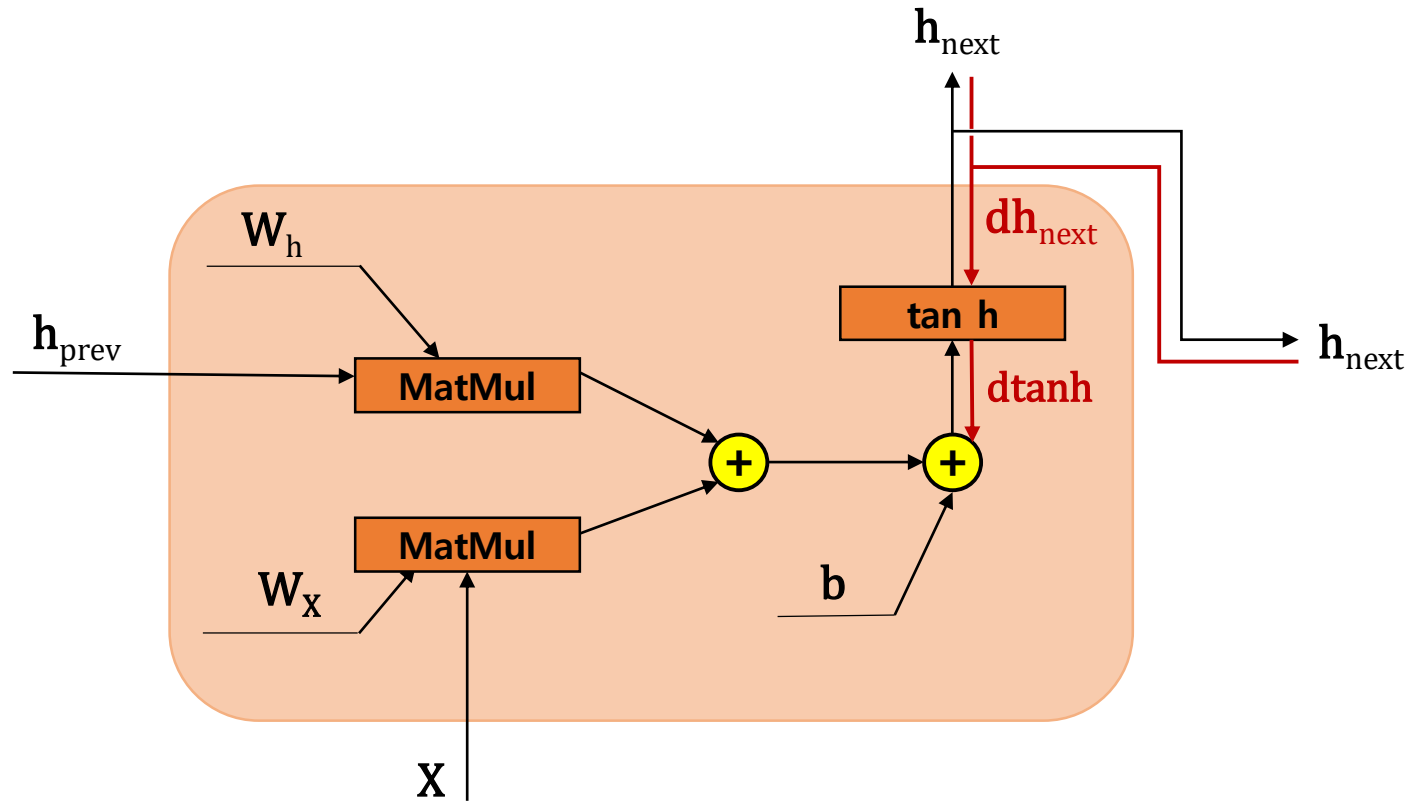
Recurrent Neural Network (RNN)

backward (역전파) - (1) dh_{next}



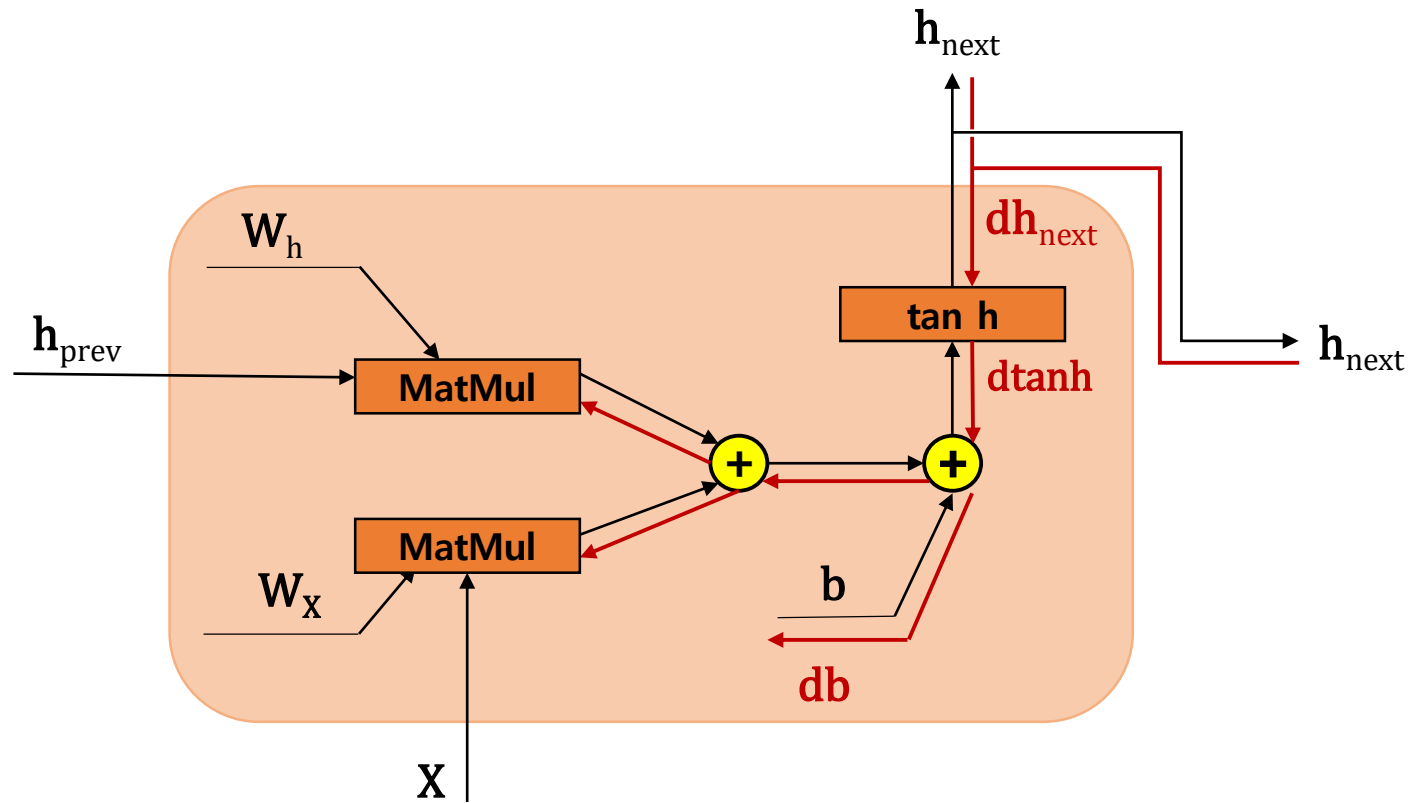
Recurrent Neural Network (RNN)

backward (역전파) - (2) dtanh



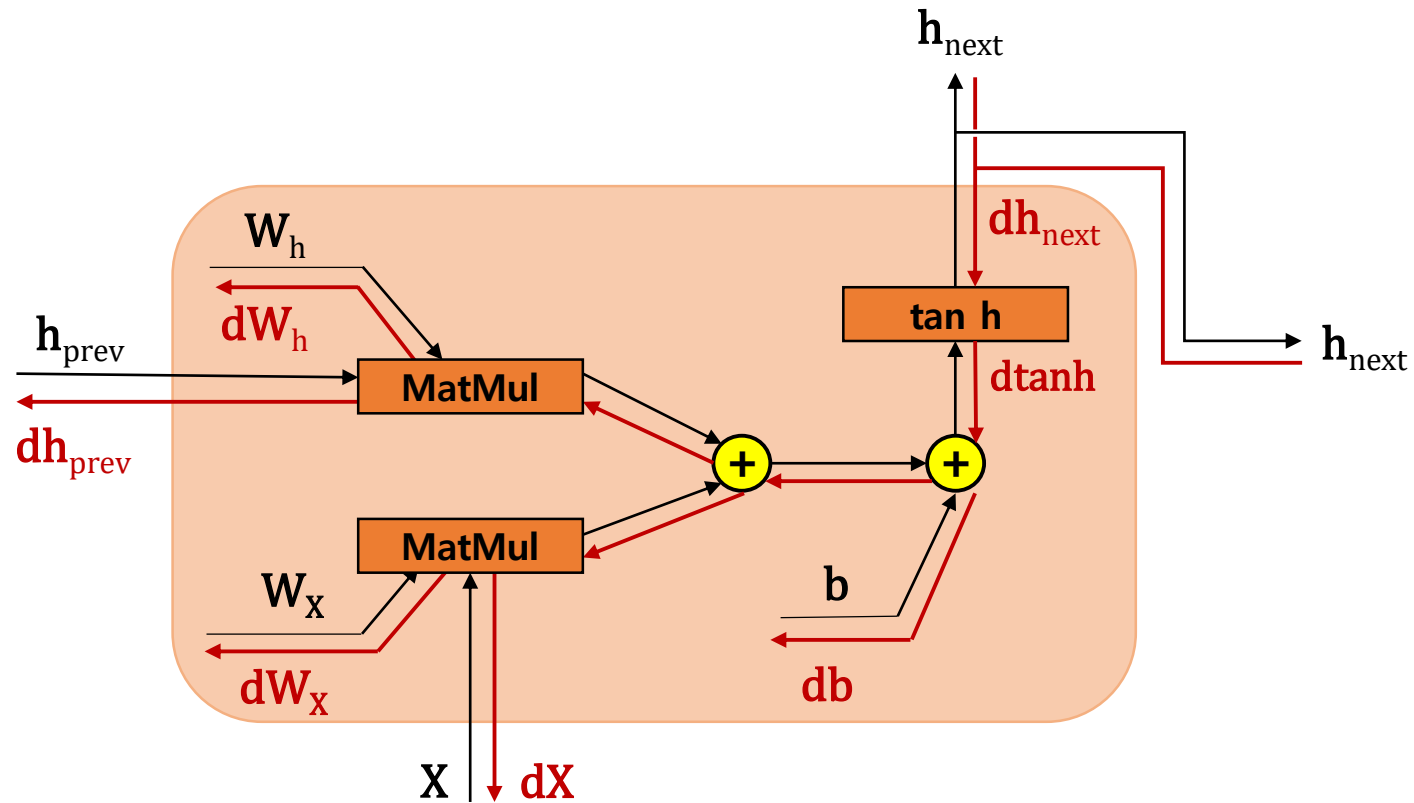
Recurrent Neural Network (RNN)

backward (역전파) - (3) 덧셈 노드



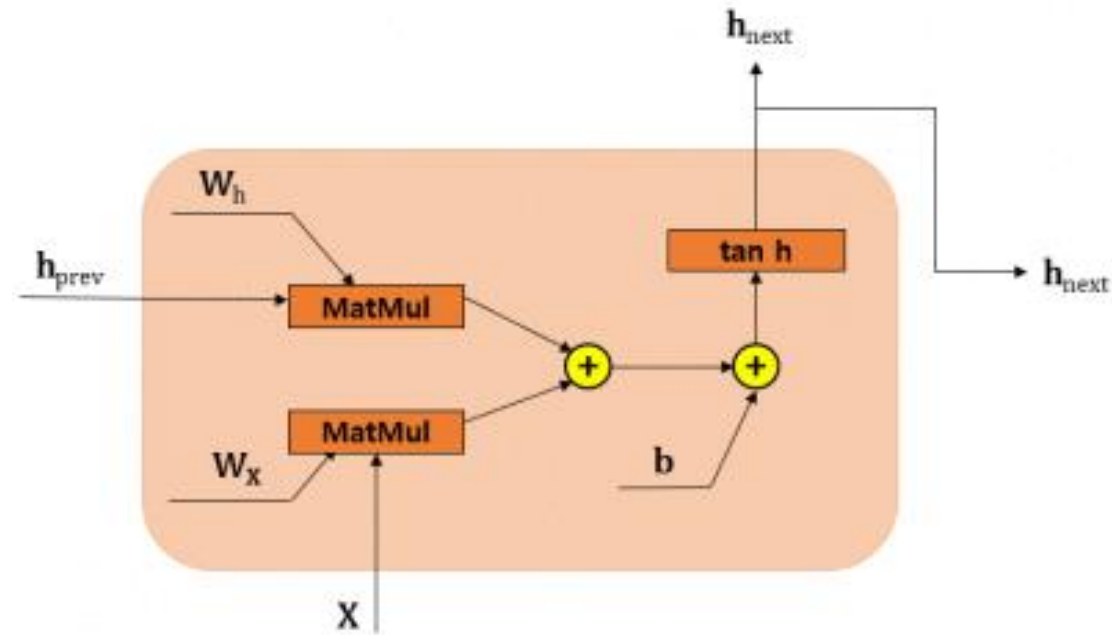
Recurrent Neural Network (RNN)

backward (역전파) - (4) 곱셈 노드

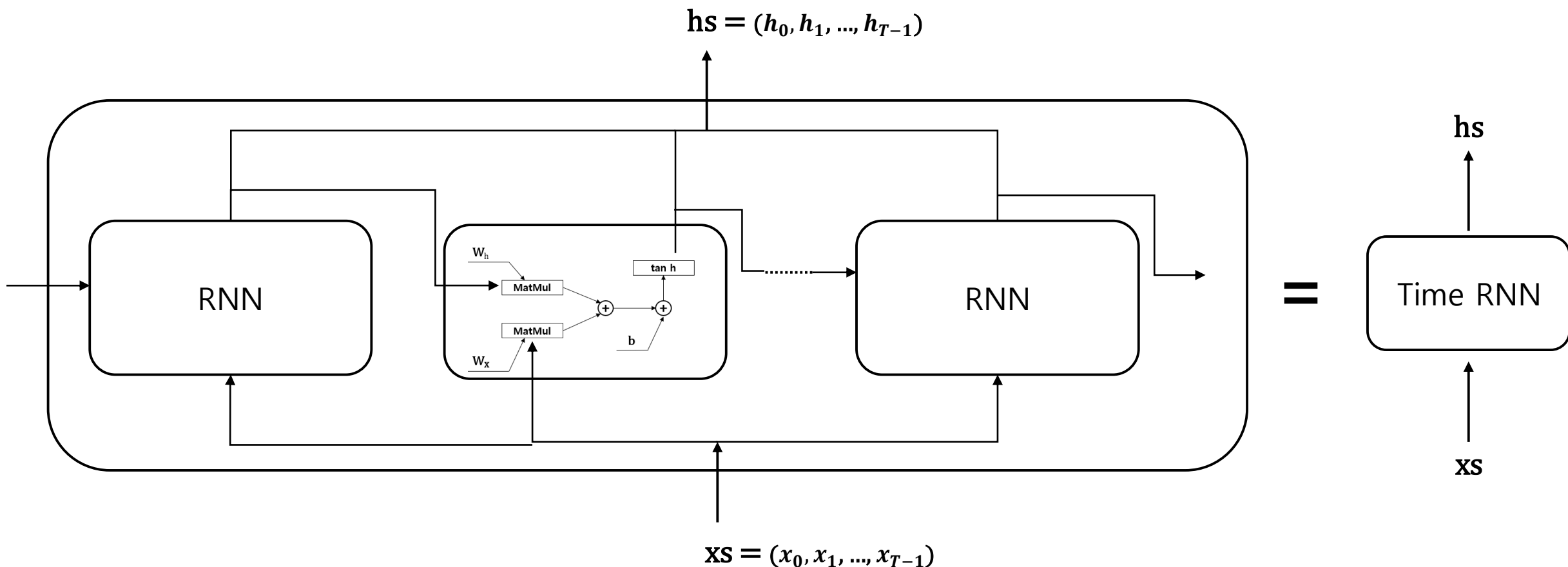


Recurrent Neural Network (RNN)

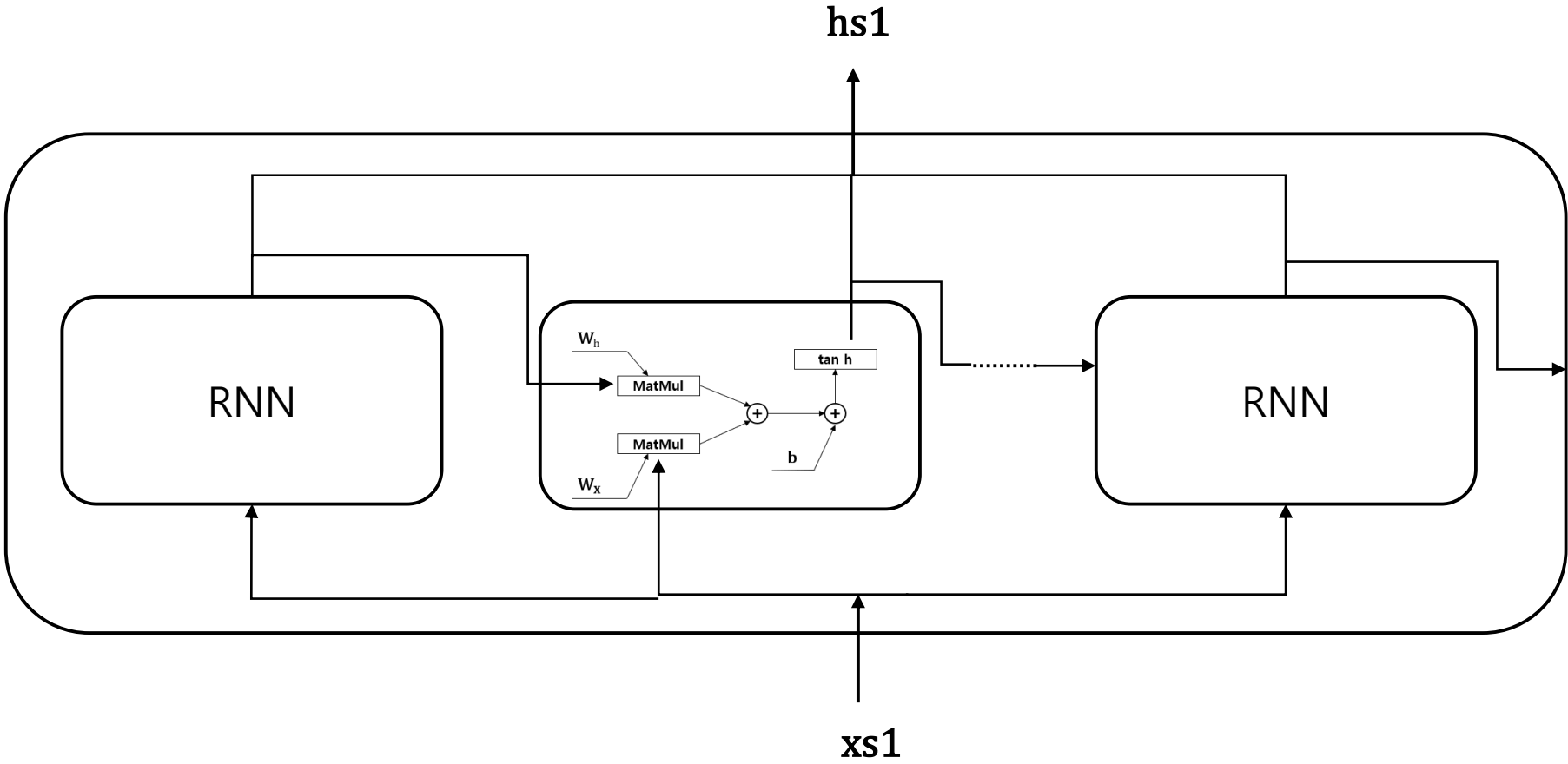
backward (역전파) – gif



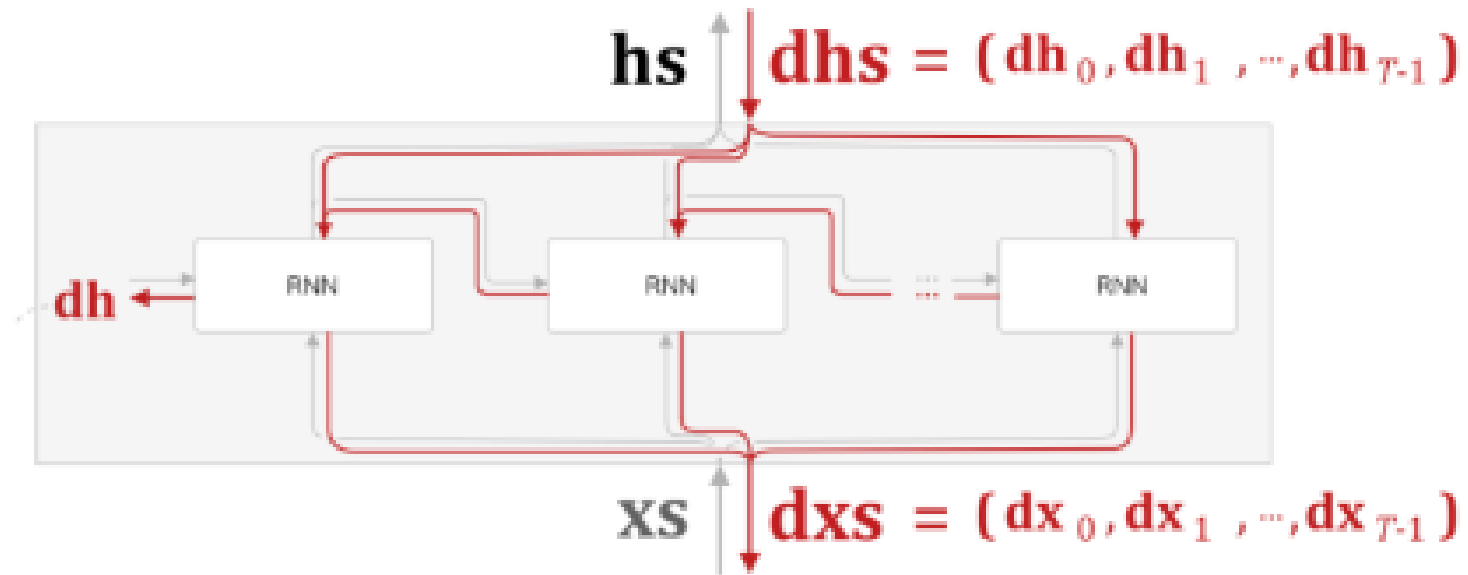
Time RNN 계층과 RNN 계층



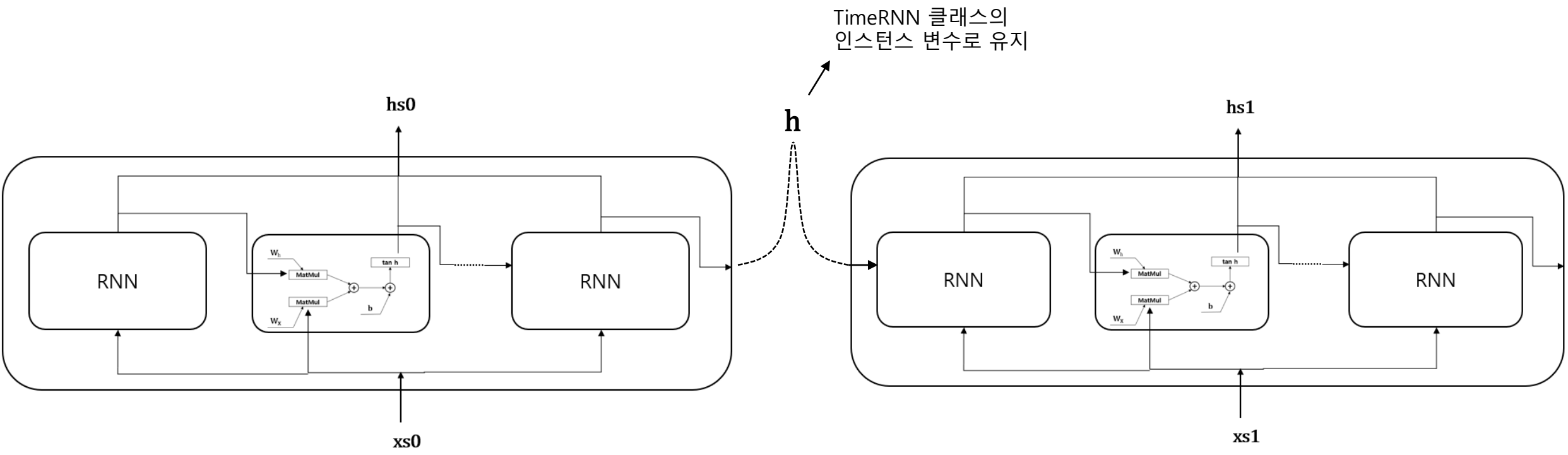
Time RNN 계층과 RNN 계층



Time RNN 계층의 역전파



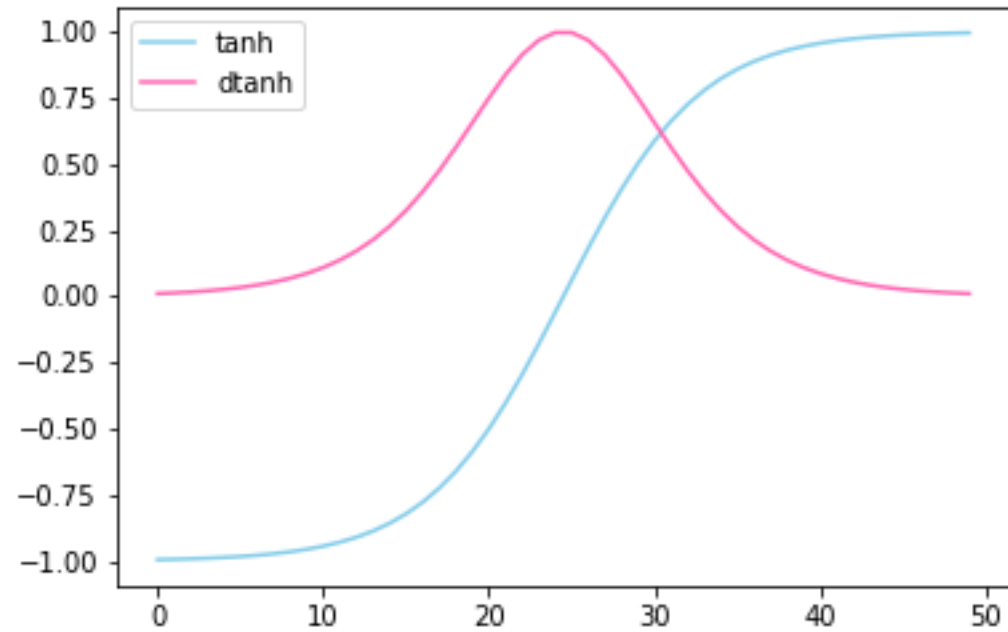
Time RNN 계층과 RNN 계층



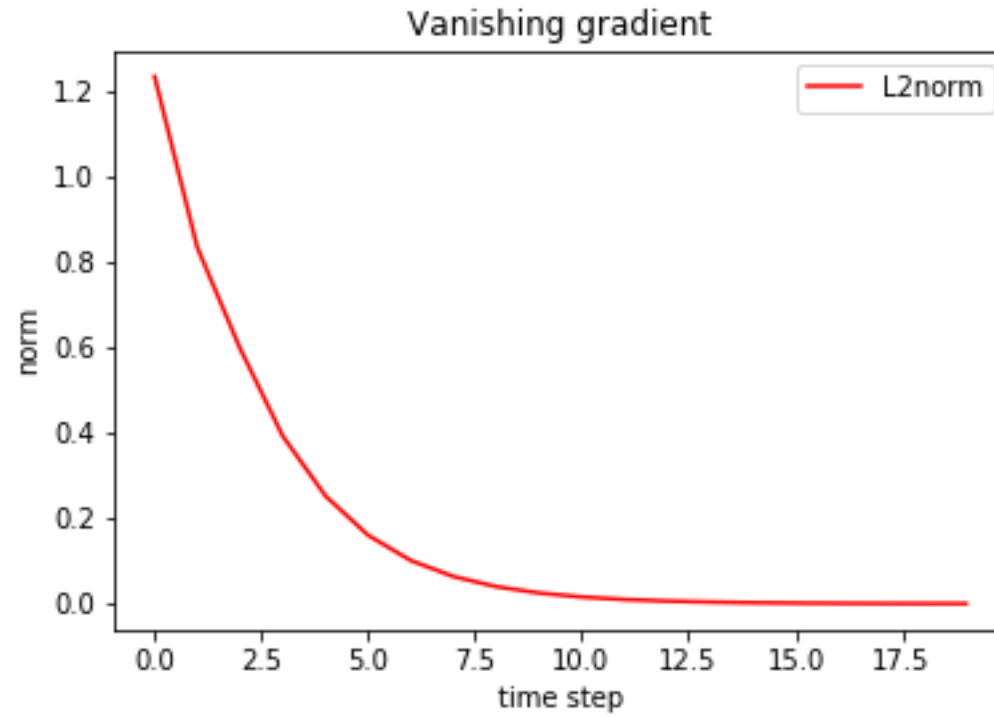
LSTM

Hyperbolic tangent

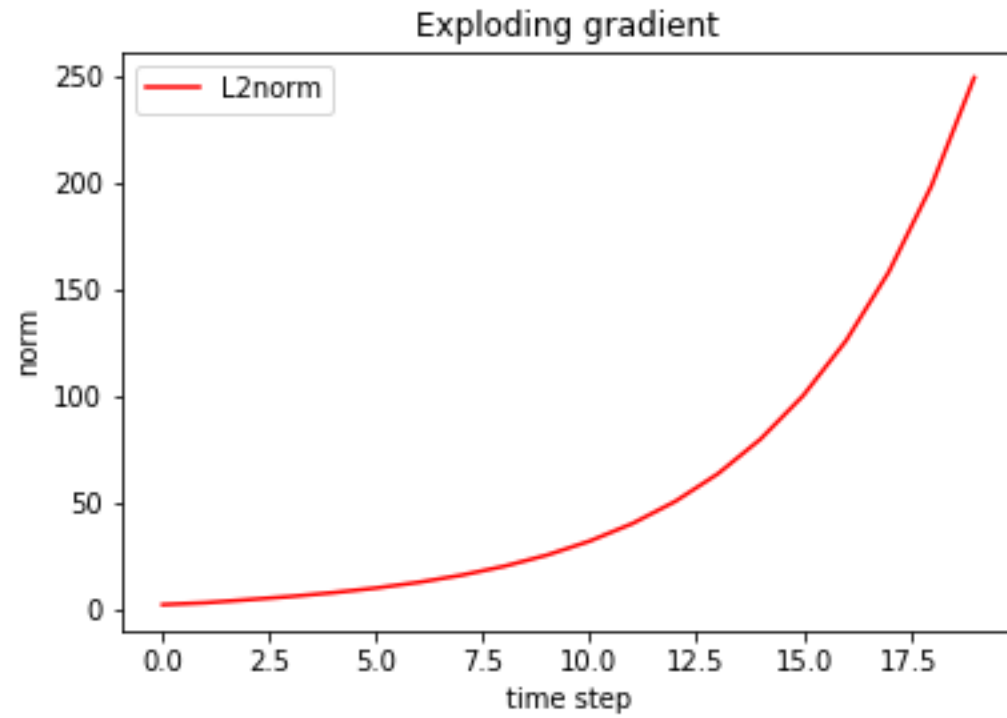
\tanh & $\text{d}\tanh$



Vanishing gradient

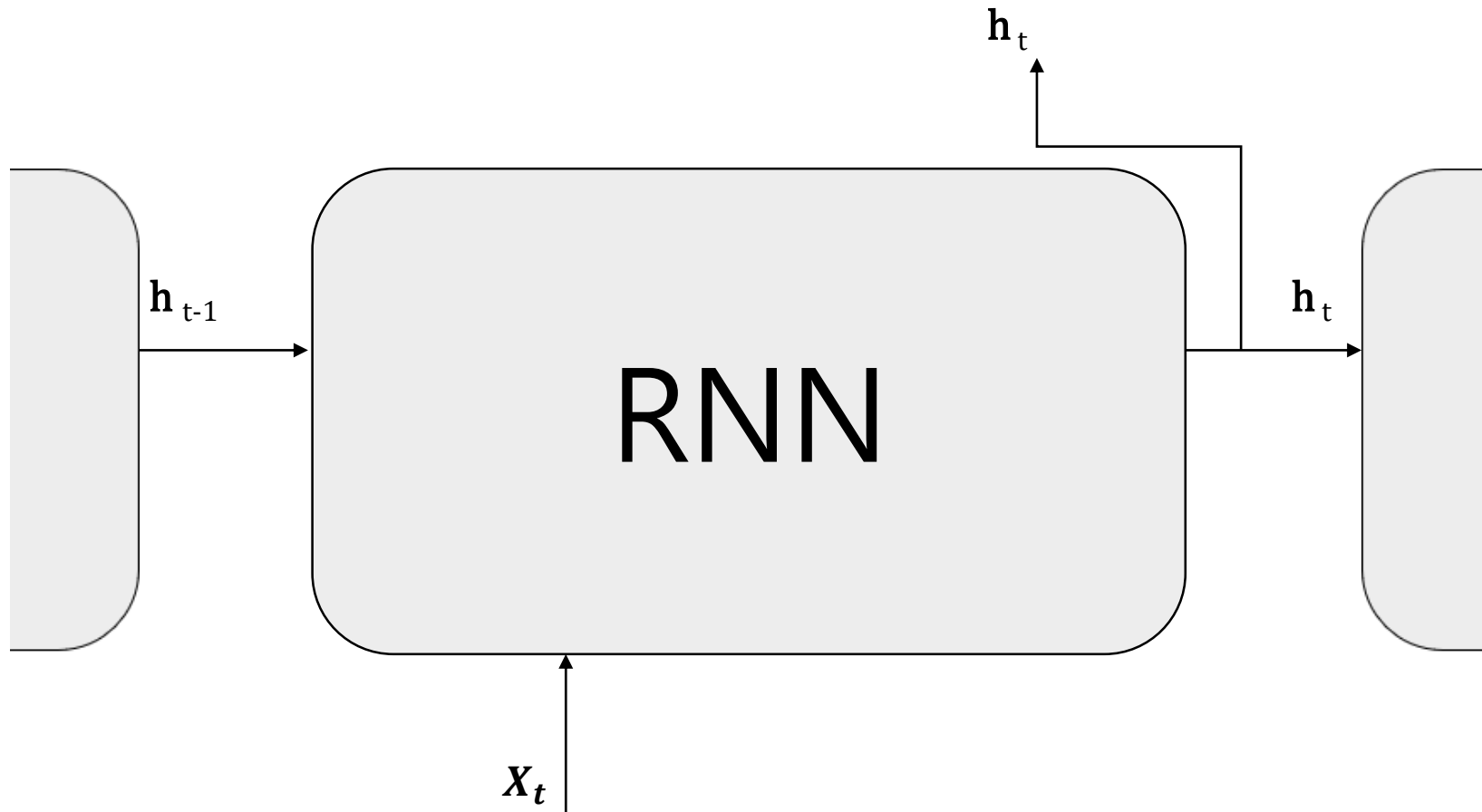


Exploding gradient

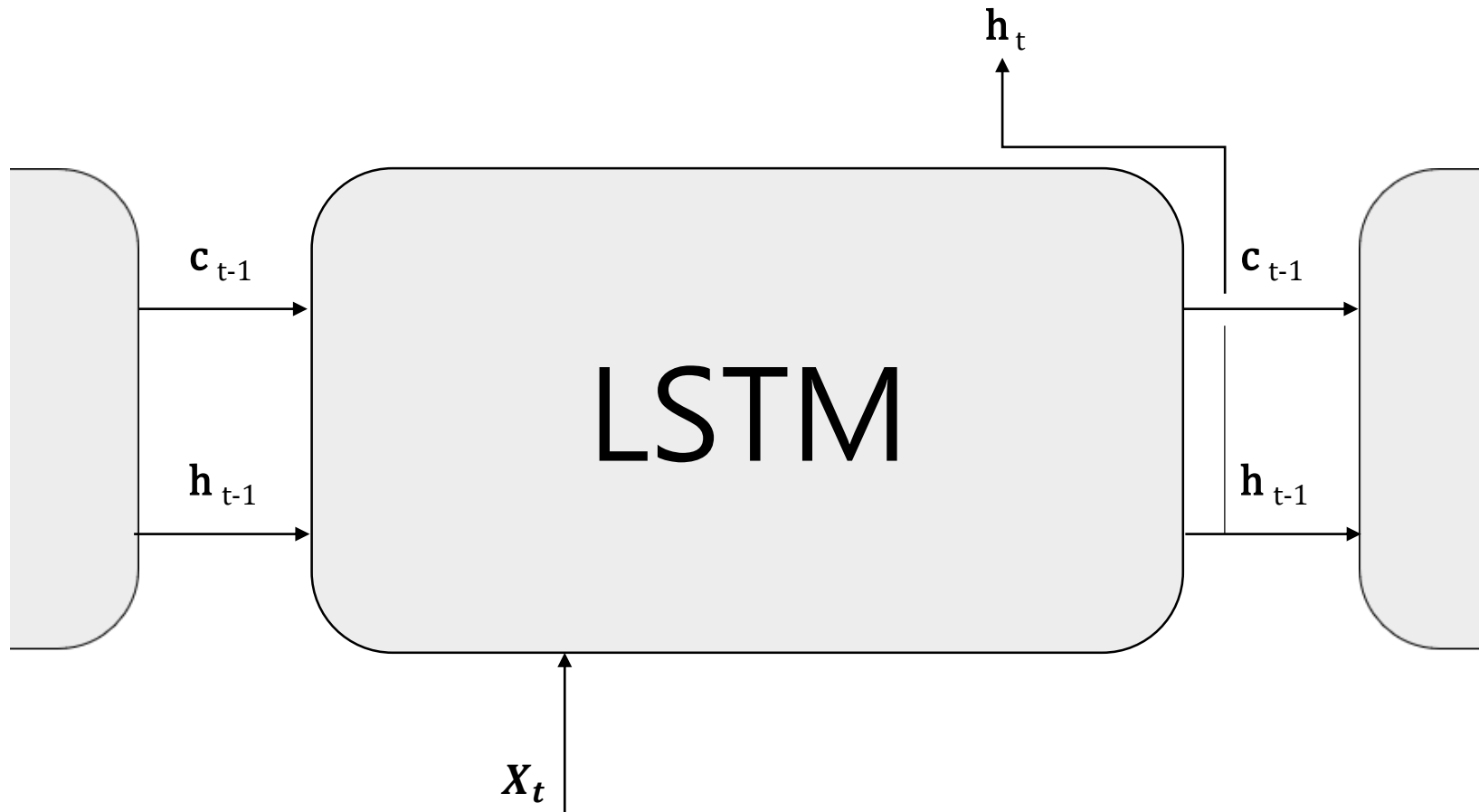


Recurrent Neural Network (RNN)

Interface

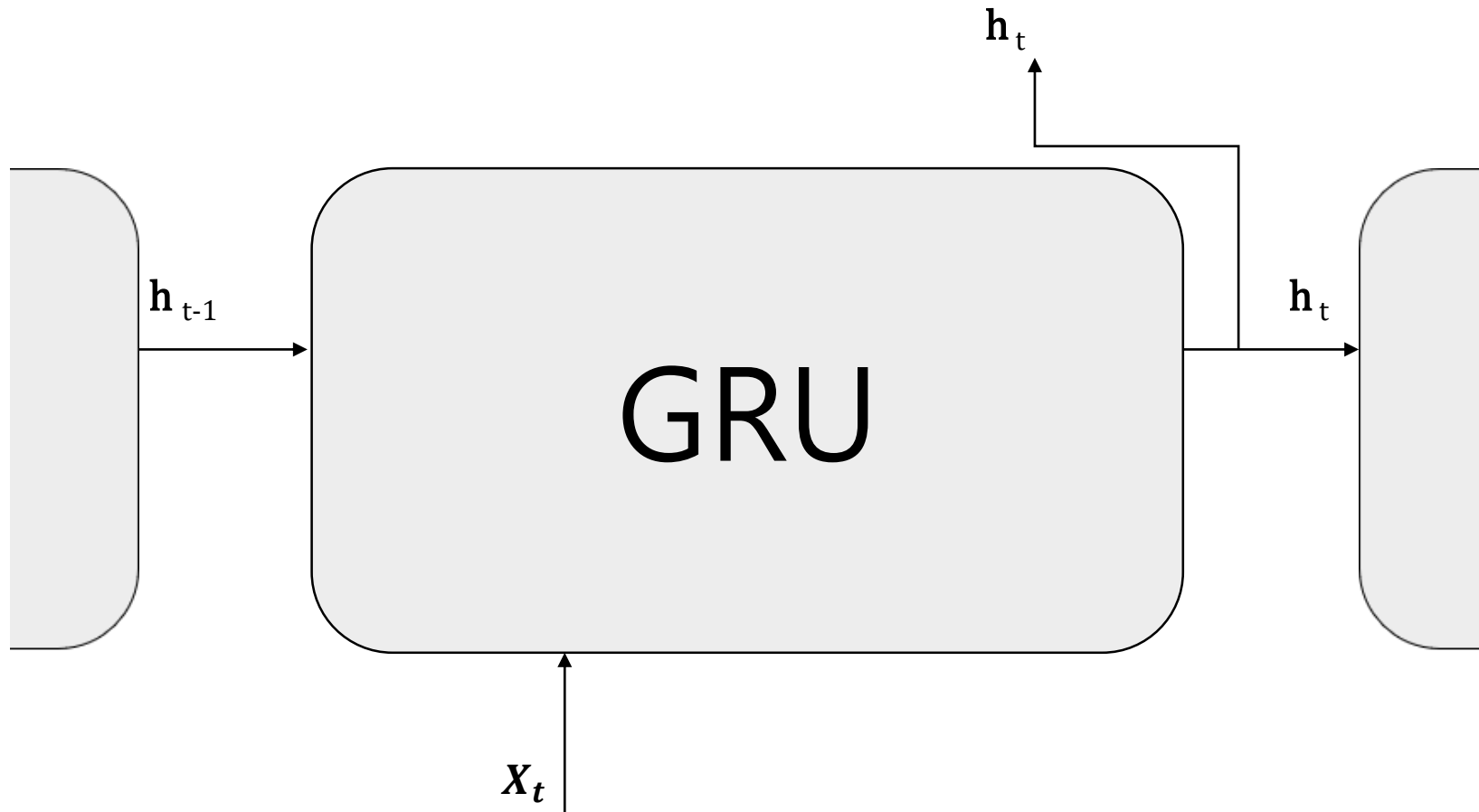


Long Shor Term Memory (LSTM) Interface

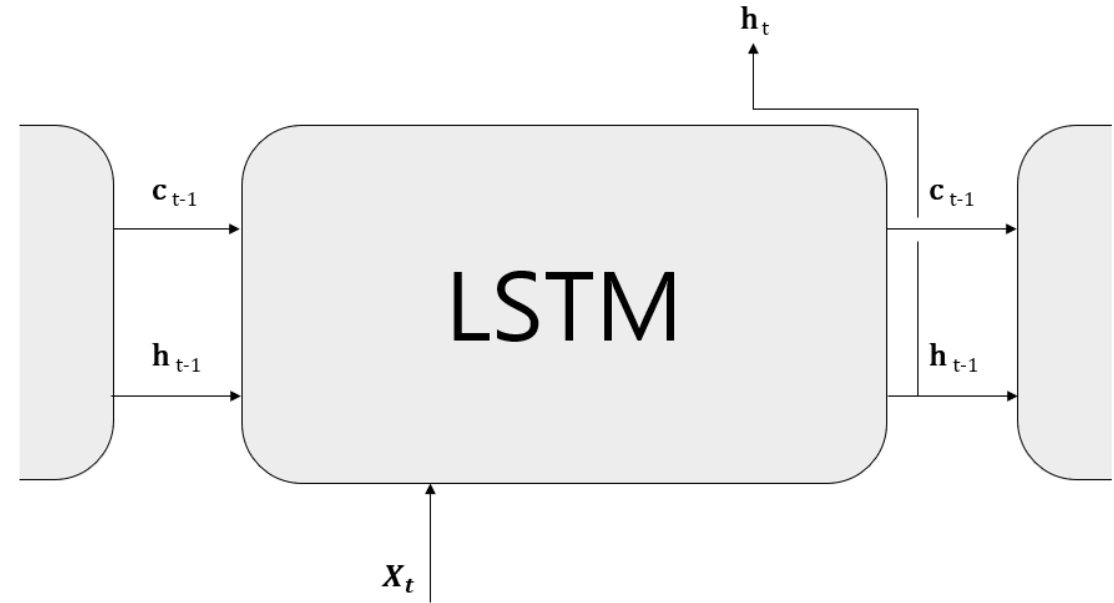
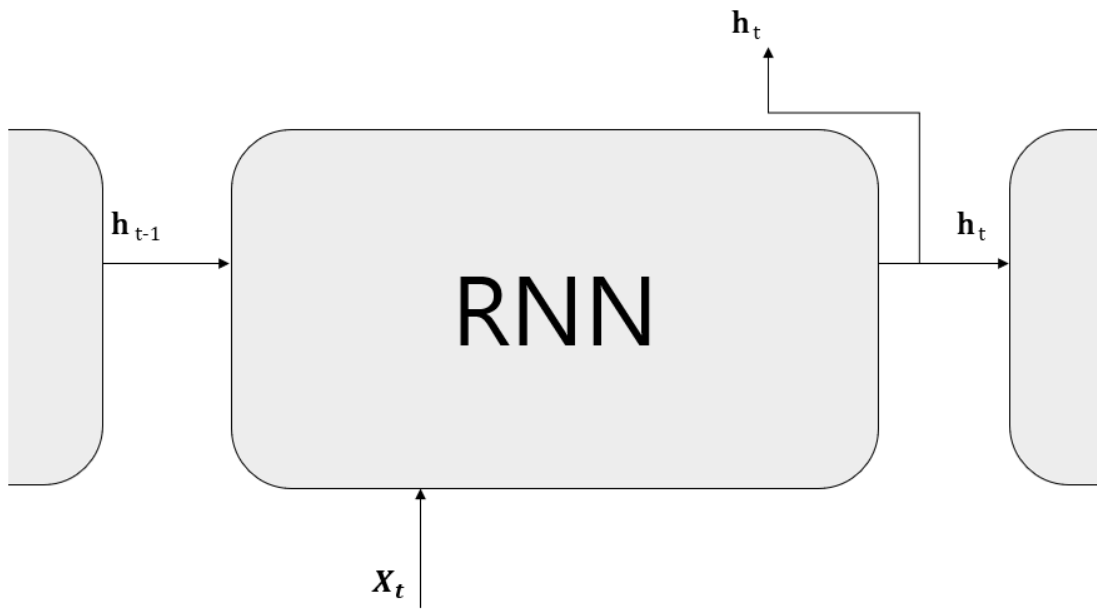


Gate Recurrent Unit (GRU)

Interface

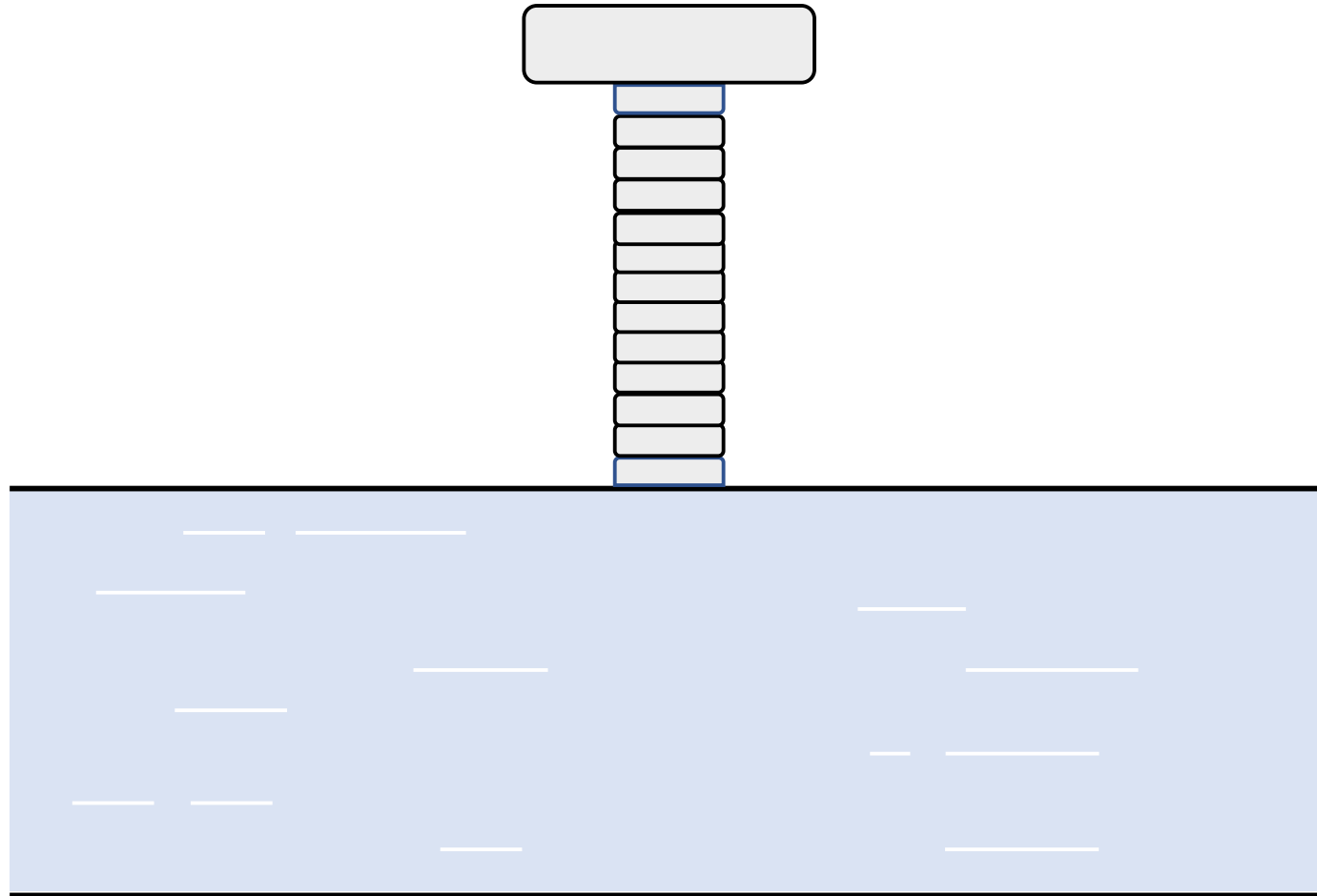


RNN과 LSTM Interface 비교



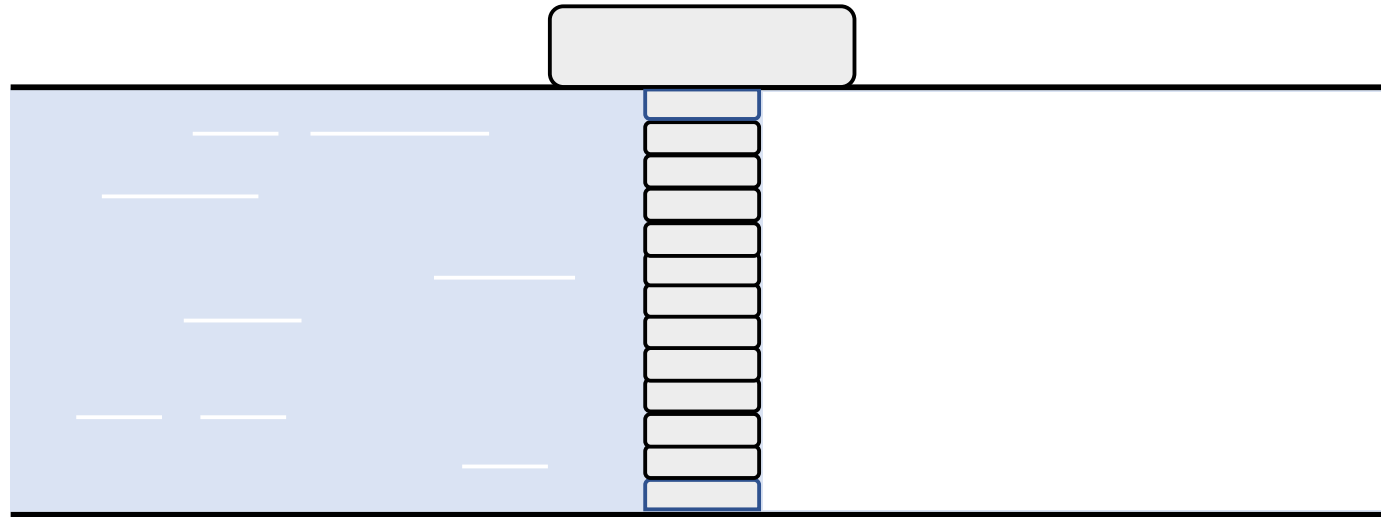
Long Shor Term Memory (LSTM)

비유하자면 게이트는 물의 흐름을 제어한다



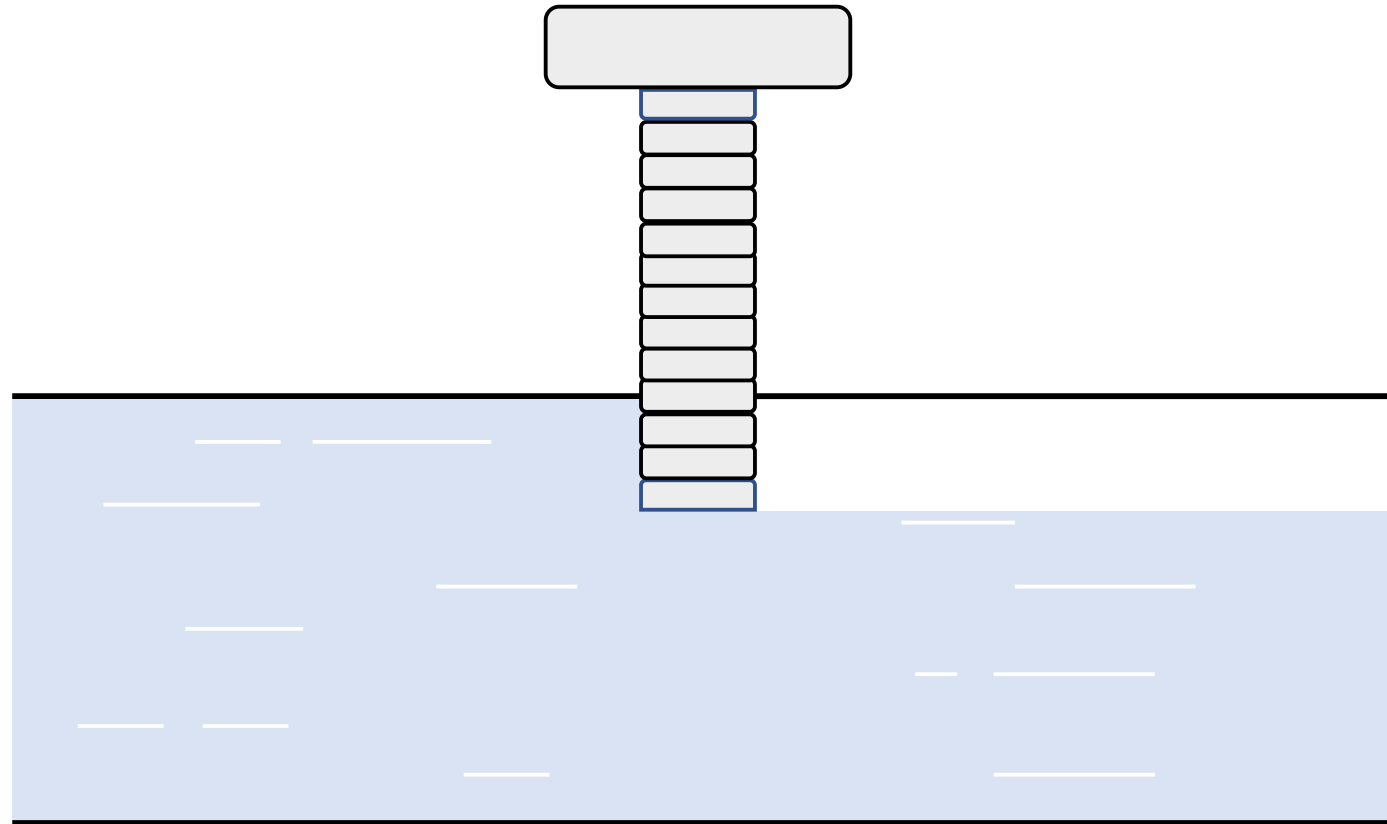
Long Shor Term Memory (LSTM)

비유하자면 게이트는 물의 흐름을 제어한다



Long Shor Term Memory (LSTM)

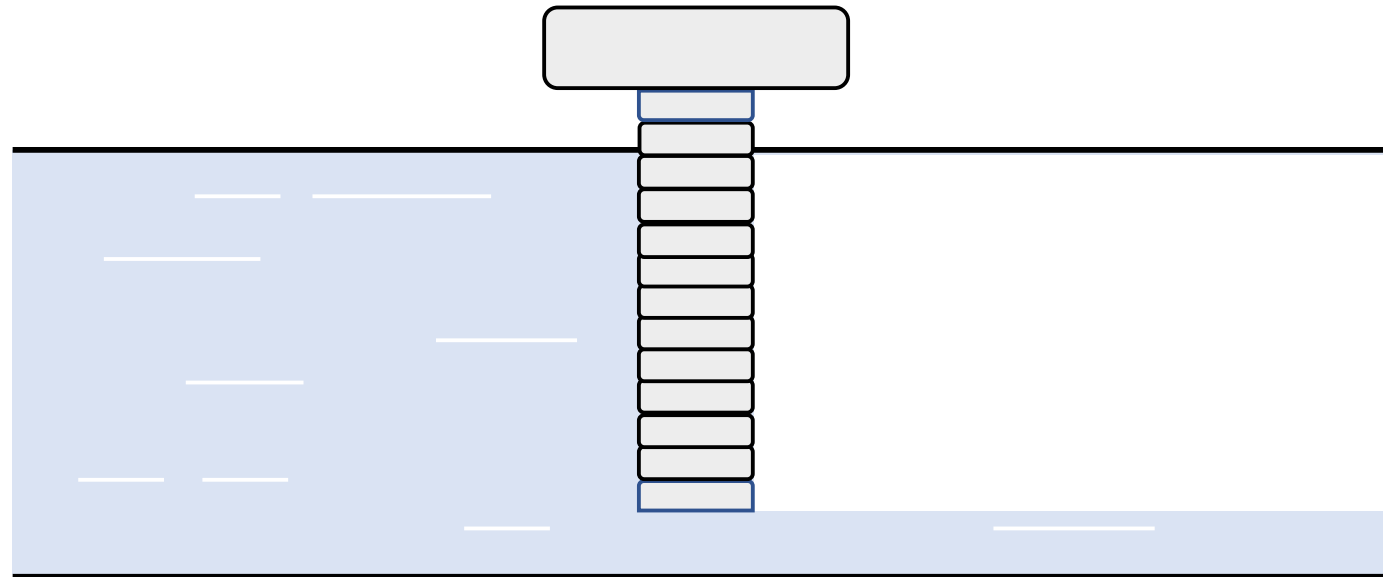
물이 흐르는 양을 0.0 ~ 1.0 범위에서 제어한다.



0.7 (70%)

Long Short Term Memory (LSTM)

물이 흐르는 양을 0.0 ~ 1.0 범위에서 제어한다.



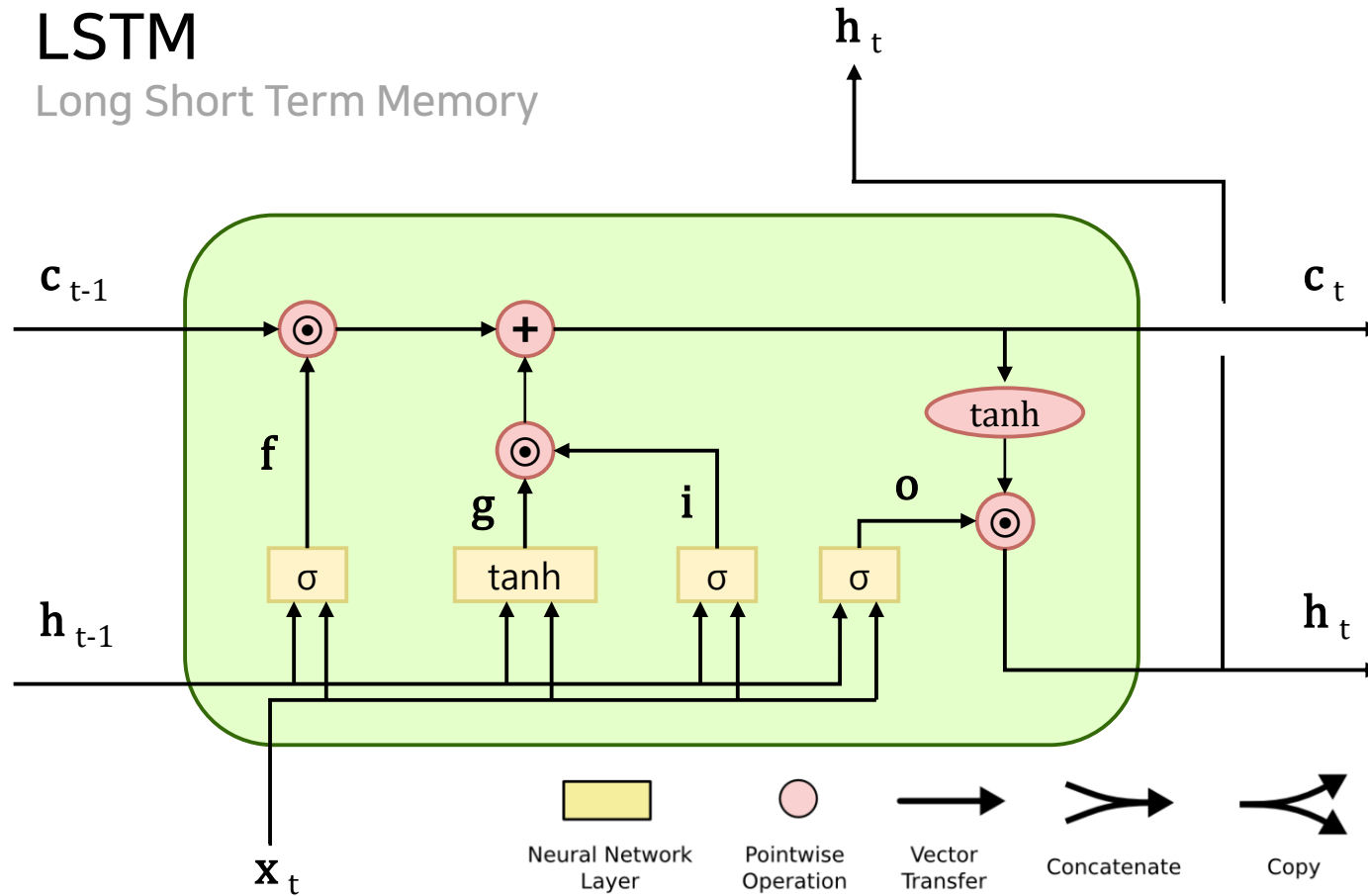
0.2 (20%)

Long Short Term Memory (LSTM)

LSTM의 계산 그래프

LSTM

Long Short Term Memory



$$f = \sigma(\mathbf{x}_t \mathbf{W}_x^{(f)} + \mathbf{h}_{t-1} \mathbf{W}_h^{(f)} + \mathbf{b}^{(f)})$$

$$g = \tanh(\mathbf{x}_t \mathbf{W}_x^{(g)} + \mathbf{h}_{t-1} \mathbf{W}_h^{(g)} + \mathbf{b}^{(g)})$$

$$i = \sigma(\mathbf{x}_t \mathbf{W}_x^{(i)} + \mathbf{h}_{t-1} \mathbf{W}_h^{(i)} + \mathbf{b}^{(i)})$$

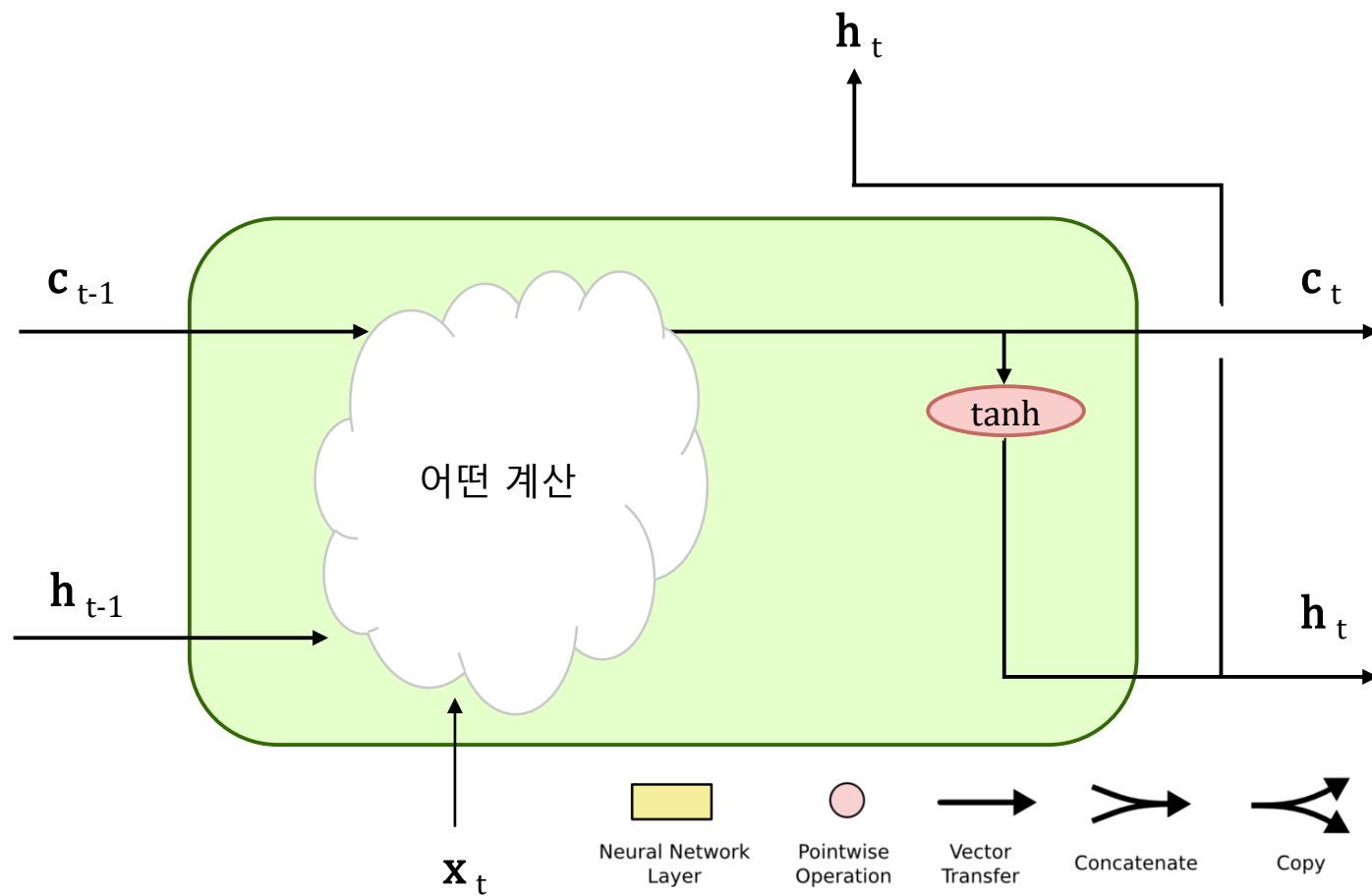
$$o = \sigma(\mathbf{x}_t \mathbf{W}_x^{(o)} + \mathbf{h}_{t-1} \mathbf{W}_h^{(o)} + \mathbf{b}^{(o)})$$

$$\mathbf{c}_t = f \odot \mathbf{c}_{t-1} + g \odot i$$

$$\mathbf{h}_t = o \odot \tanh(\mathbf{c}_t)$$

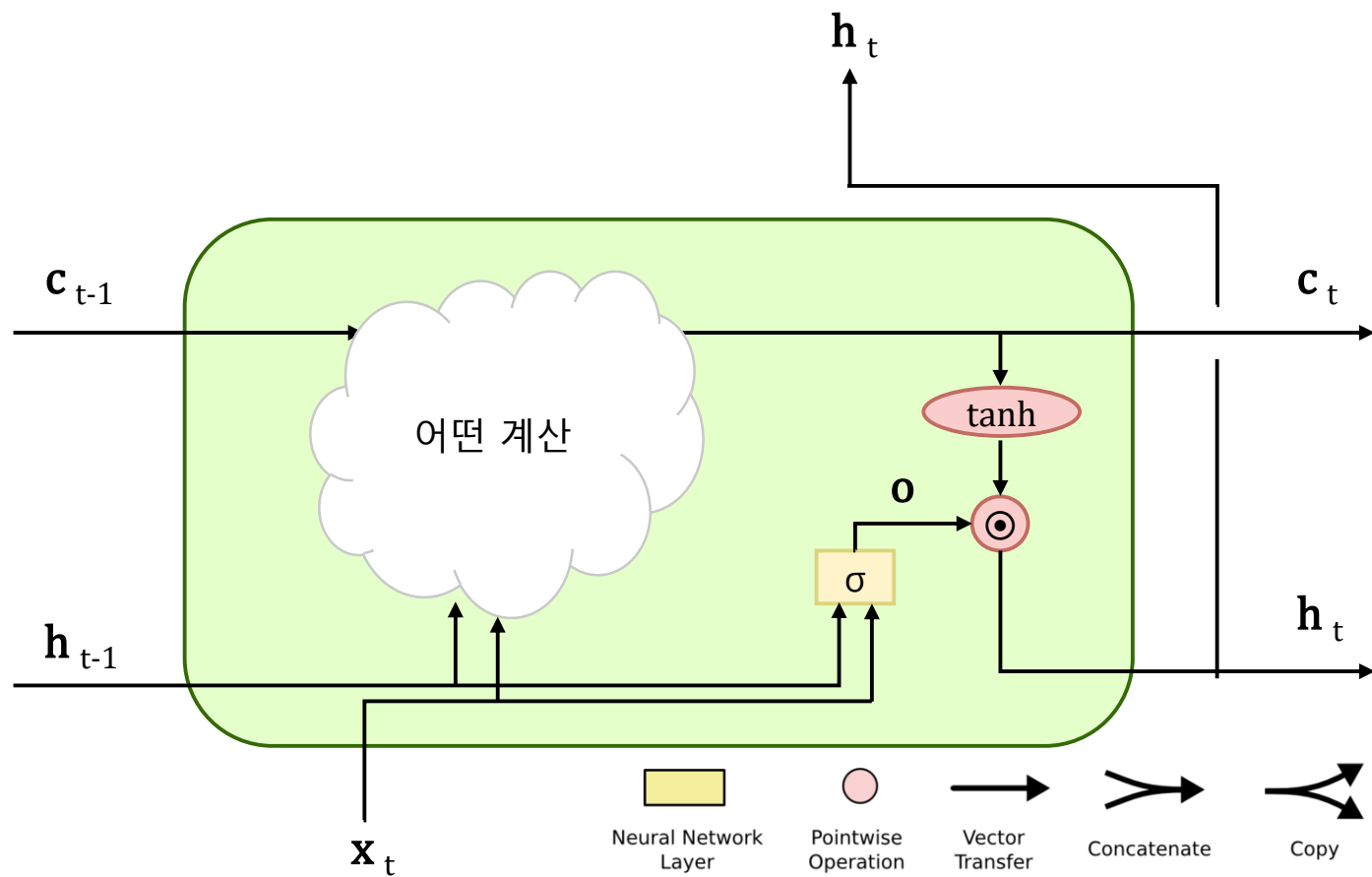
Long Short Term Memory (LSTM)

기억 셀 c_t 를 바탕으로 은닉상태 h_t 를 계산하는 LSTM 계층



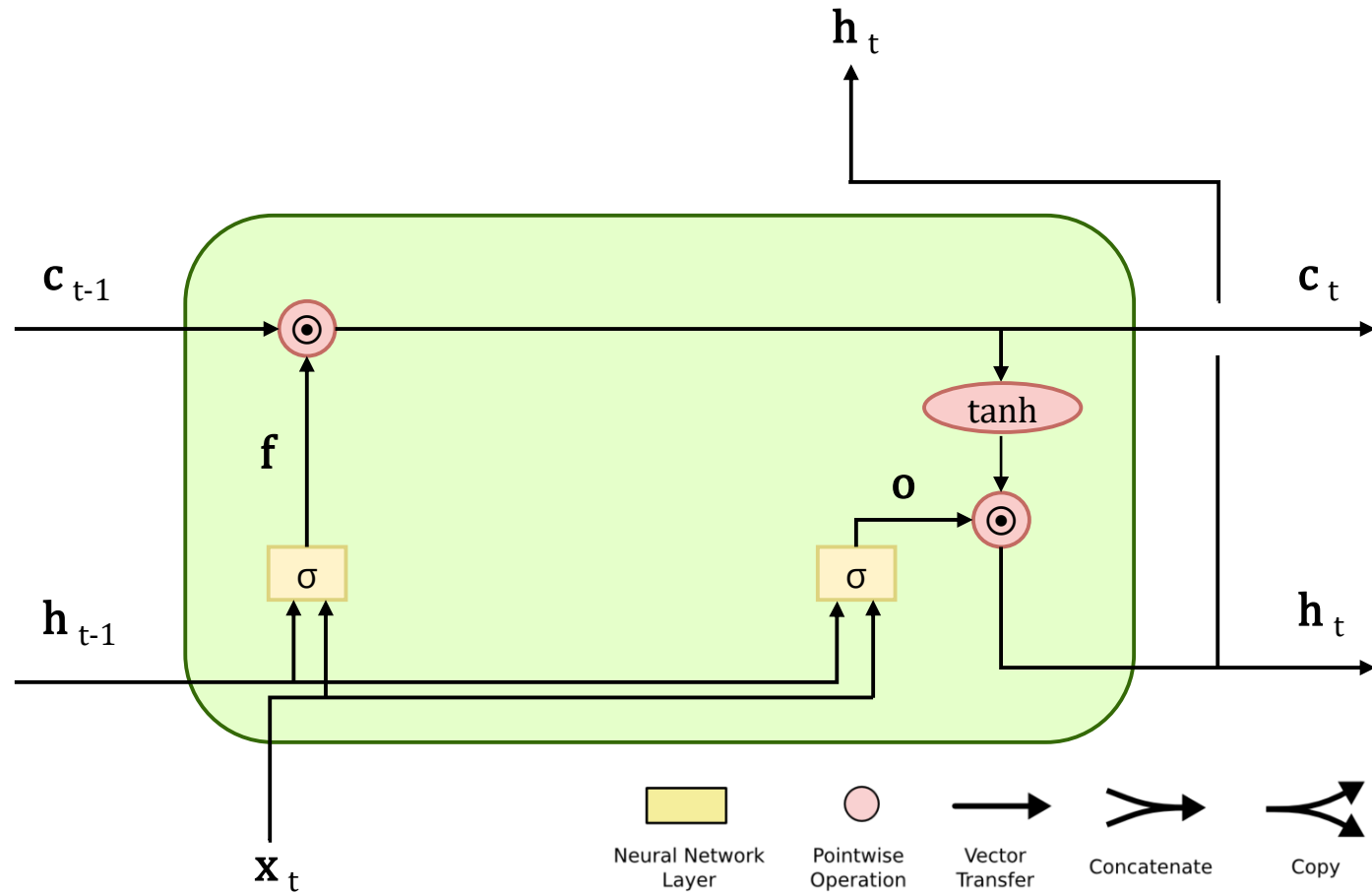
Long Short Term Memory (LSTM)

output 게이트 추가 (o gate)



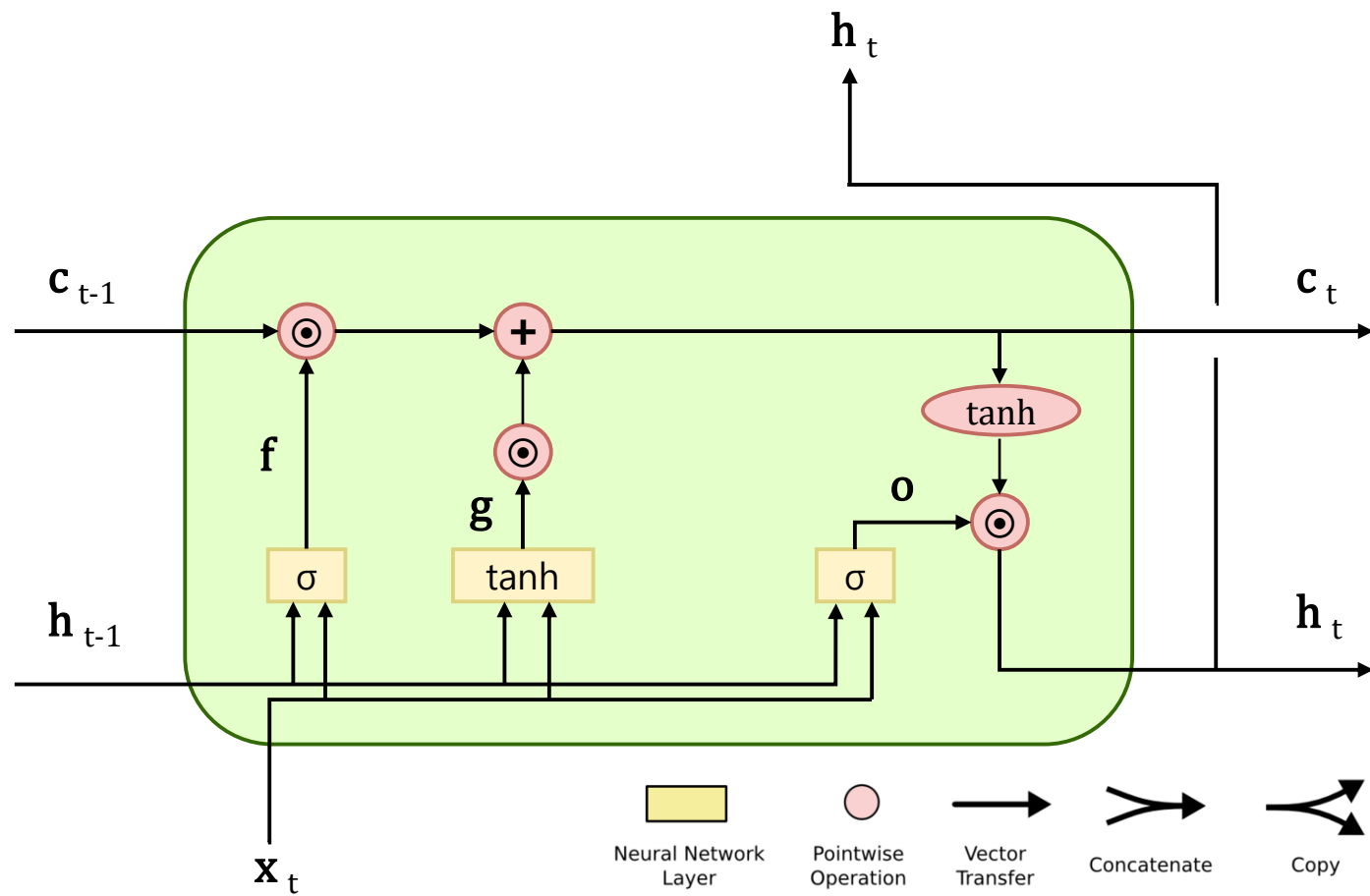
Long Short Term Memory (LSTM)

forget 게이트 추가 (f gate)



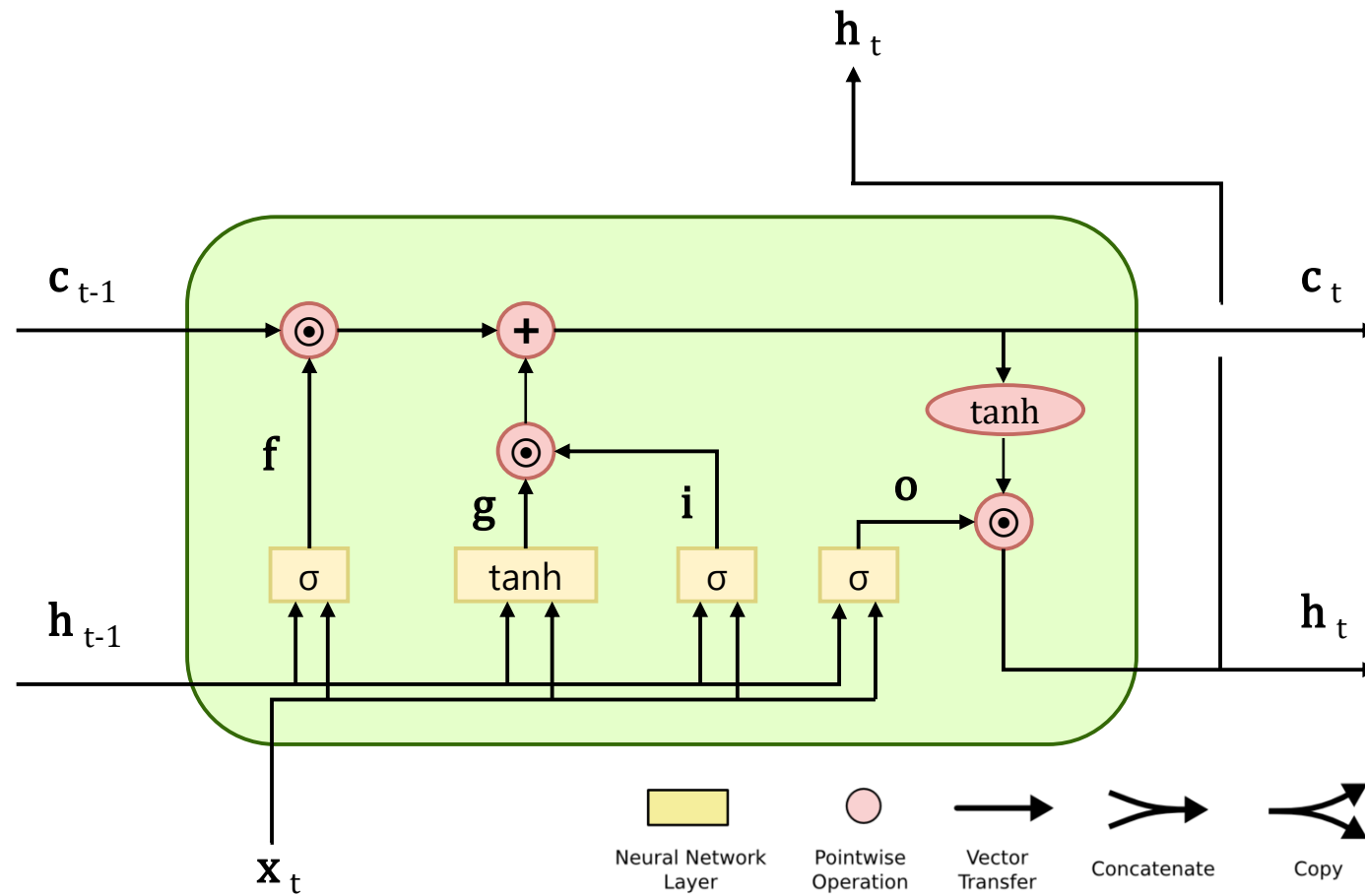
Long Short Term Memory (LSTM)

새로운 기억 셀에 필요한 정보를 추가 (g gate)

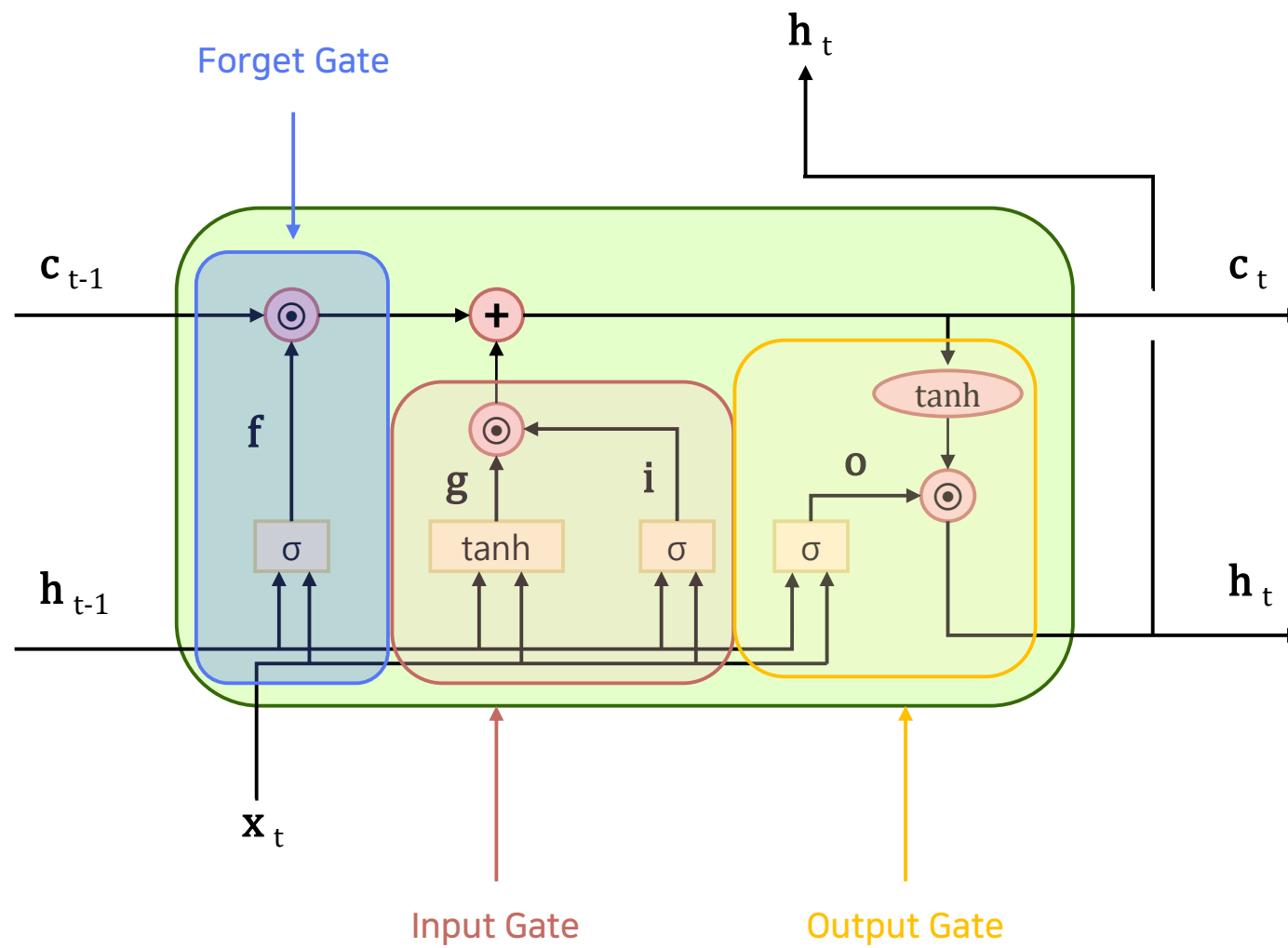


Long Short Term Memory (LSTM)

Input 게이트 추가 (i gate)

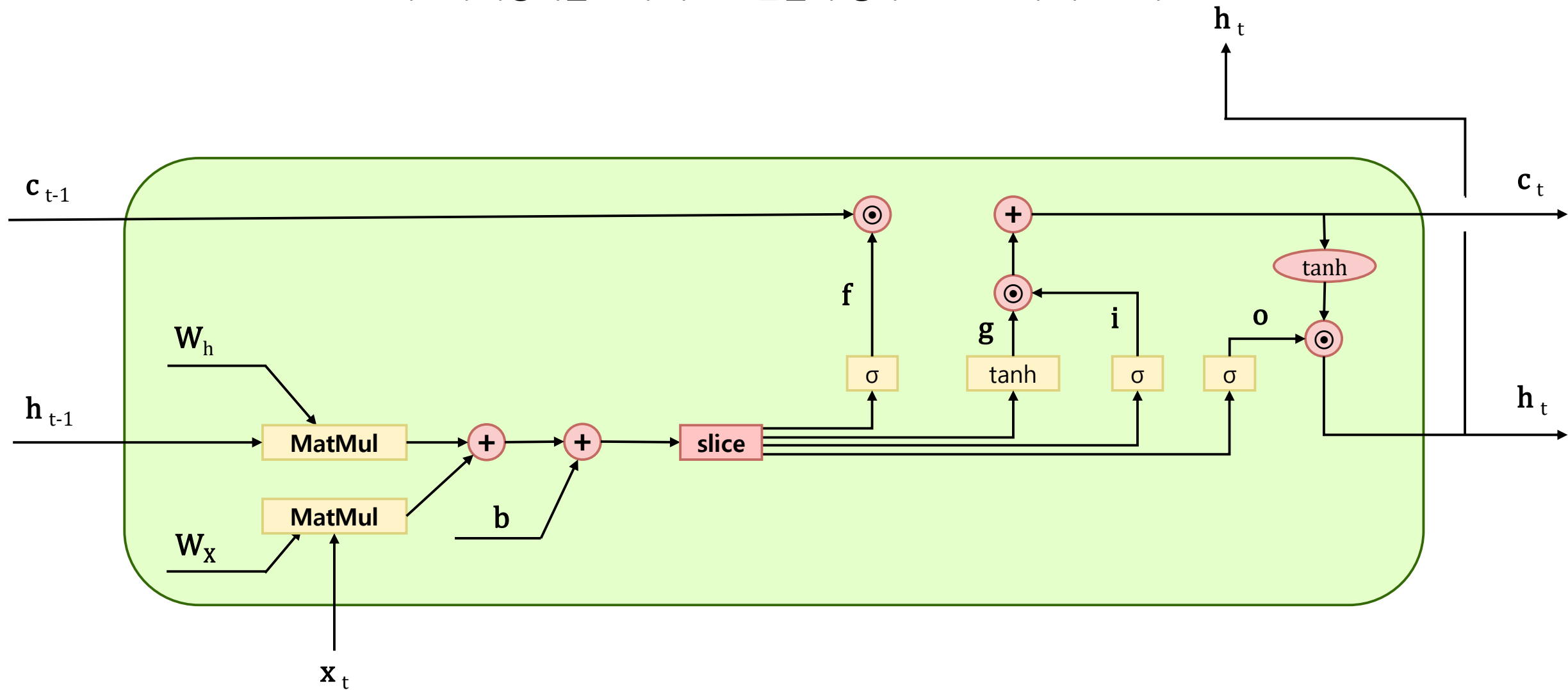


Long Short Term Memory (LSTM)

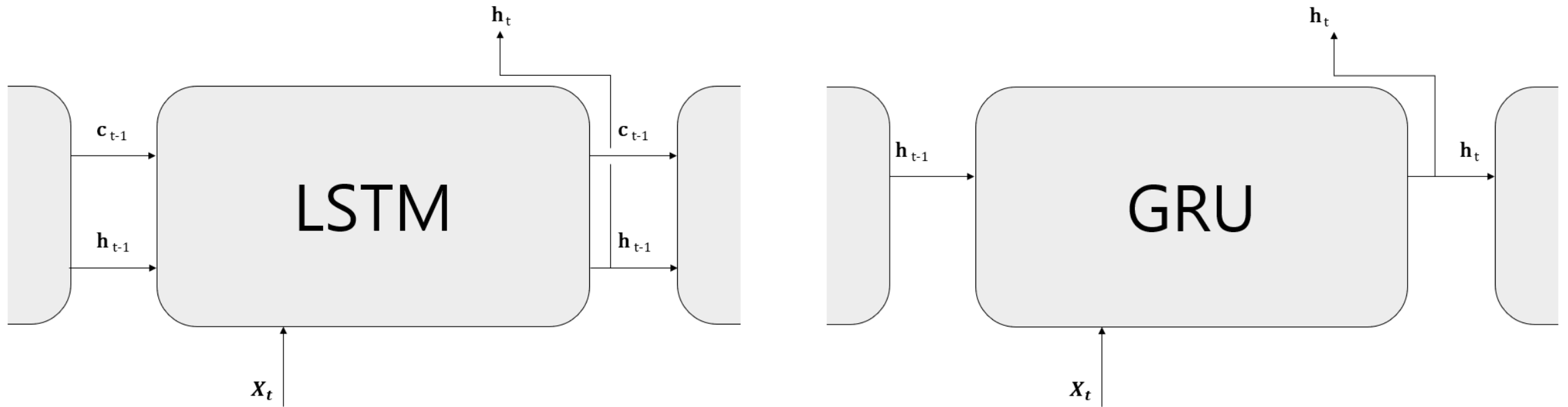


Long Short Term Memory (LSTM)

4개분의 가중치를 모아 아핀 변환을 수행하는 LSTM의 계산그래프



LSTM과 GRU Interface 비교

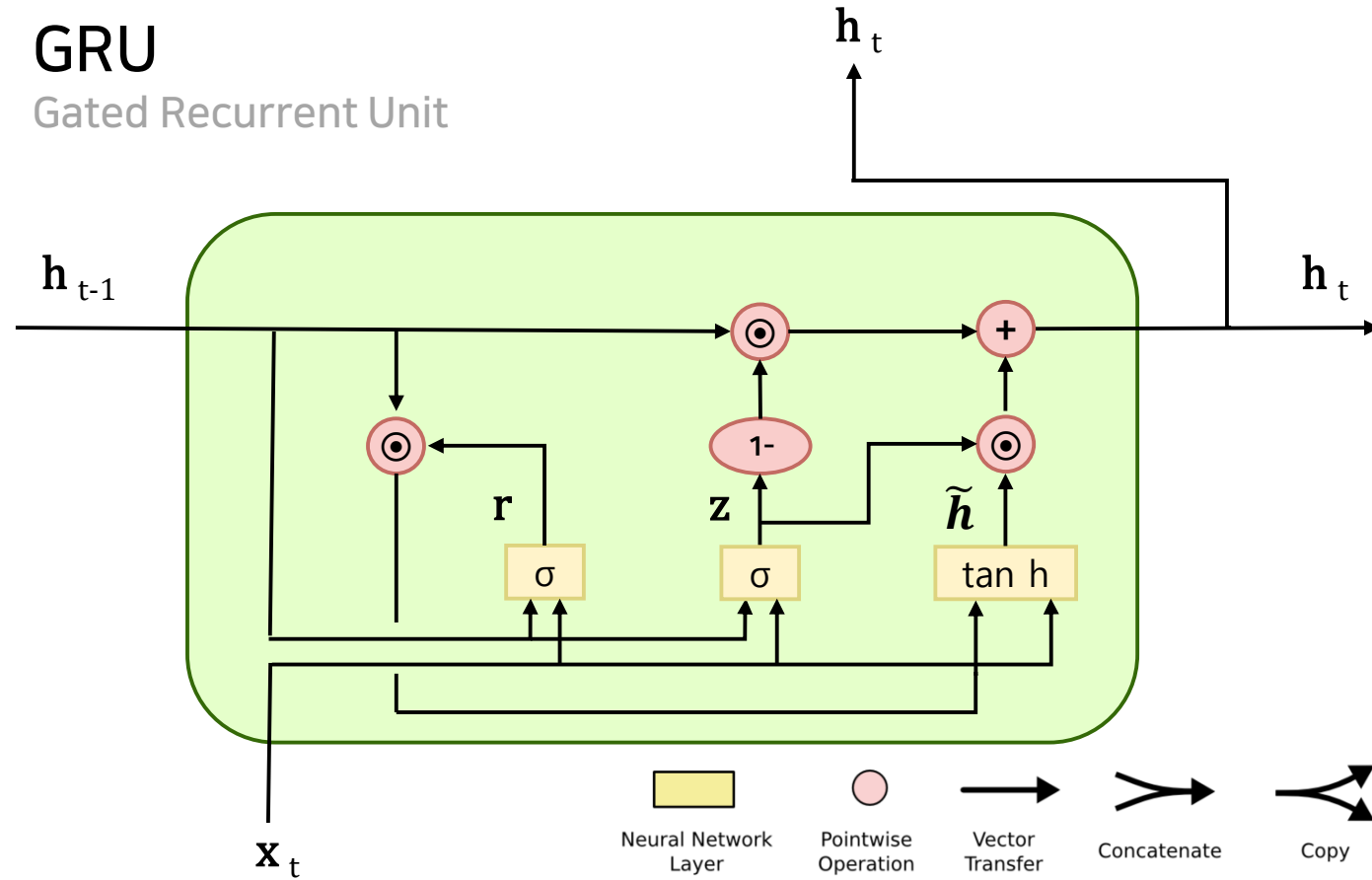


Gated Recurrent Unit (GRU)

GRU의 계산 그래프

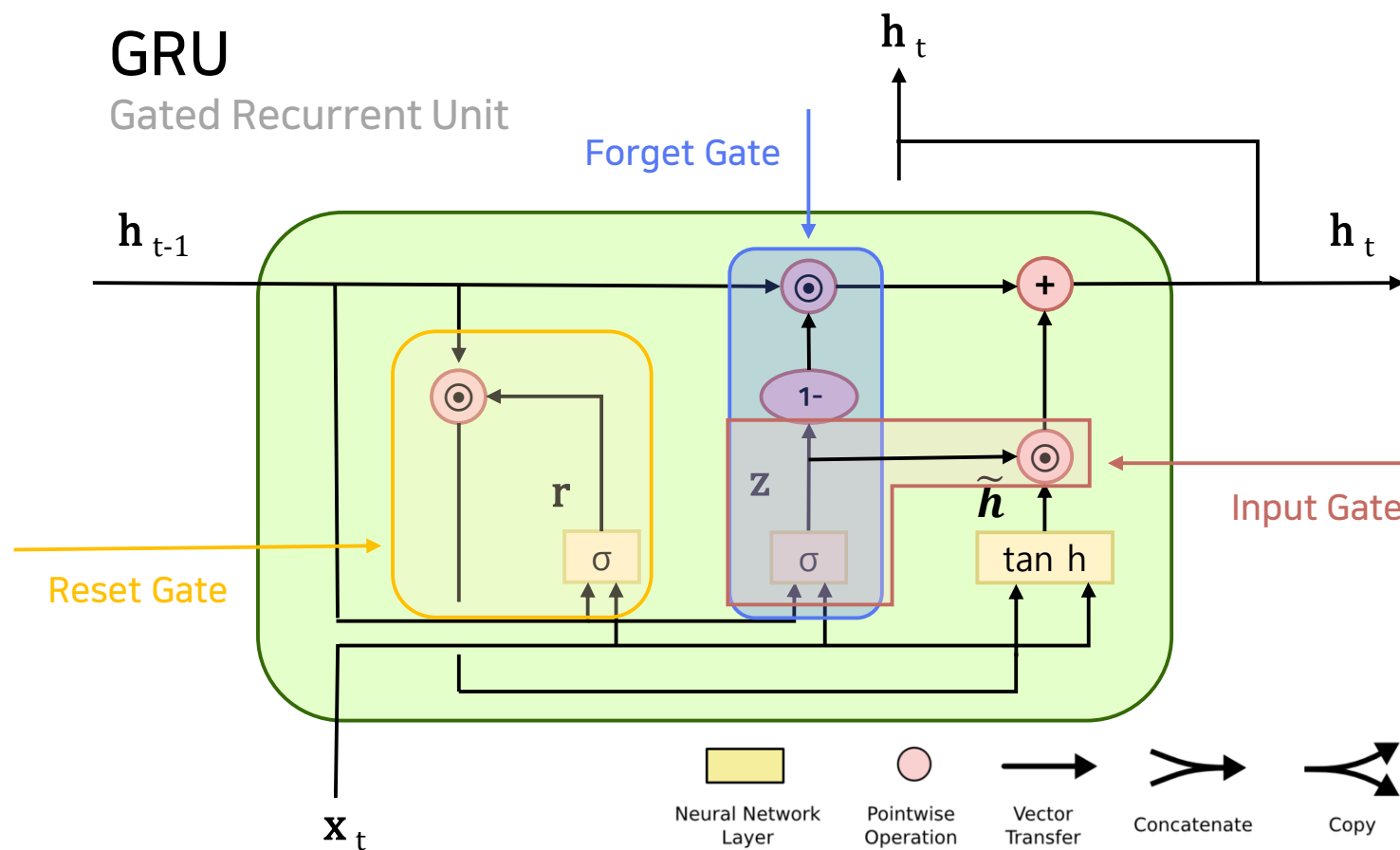
GRU

Gated Recurrent Unit



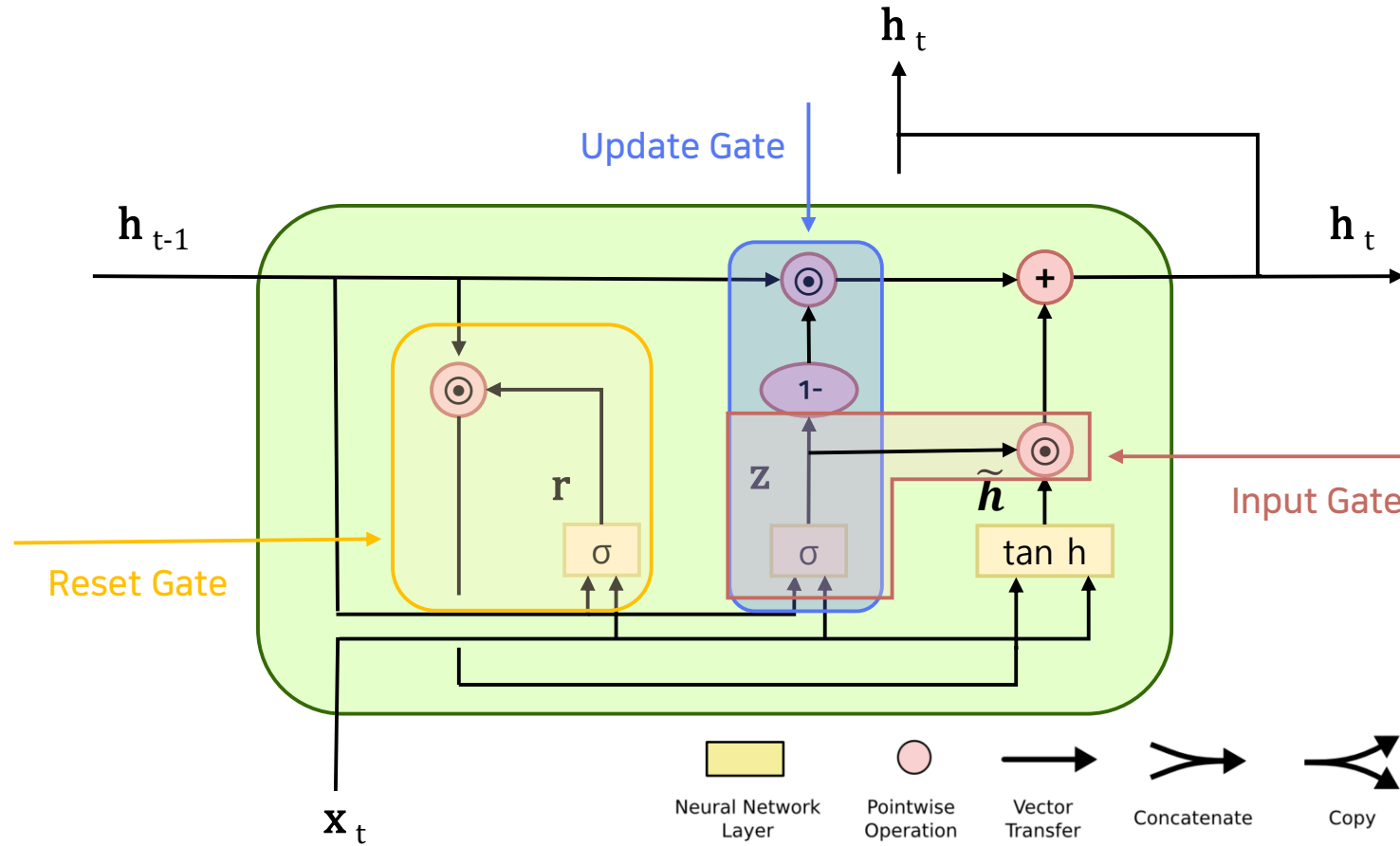
Gated Recurrent Unit (GRU)

GRU의 Forget Gate 와 Input Gate



Gated Recurrent Unit (GRU)

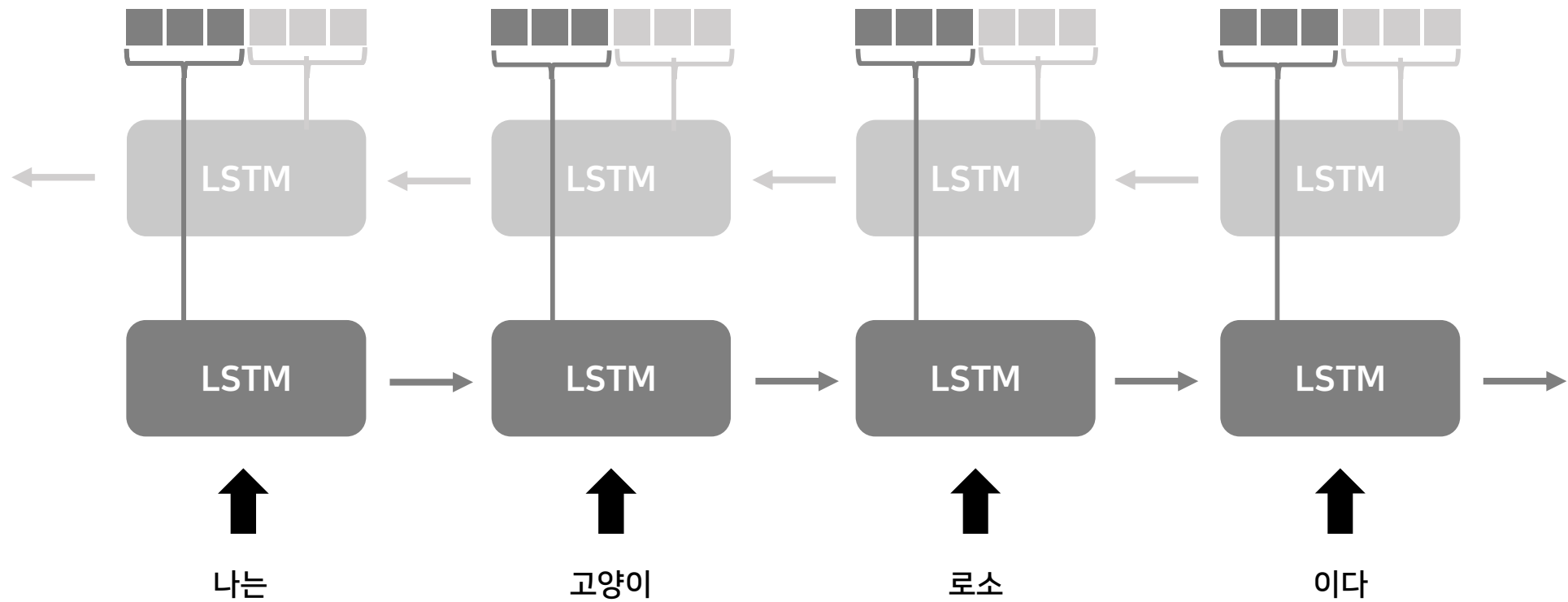
GRU의 Forget Gate 와 Input Gate



BLSTM

Bidirection LSTM

BLSTM



seq2seq

Encoder와 Decoder가 음성인식을 수행하는 예

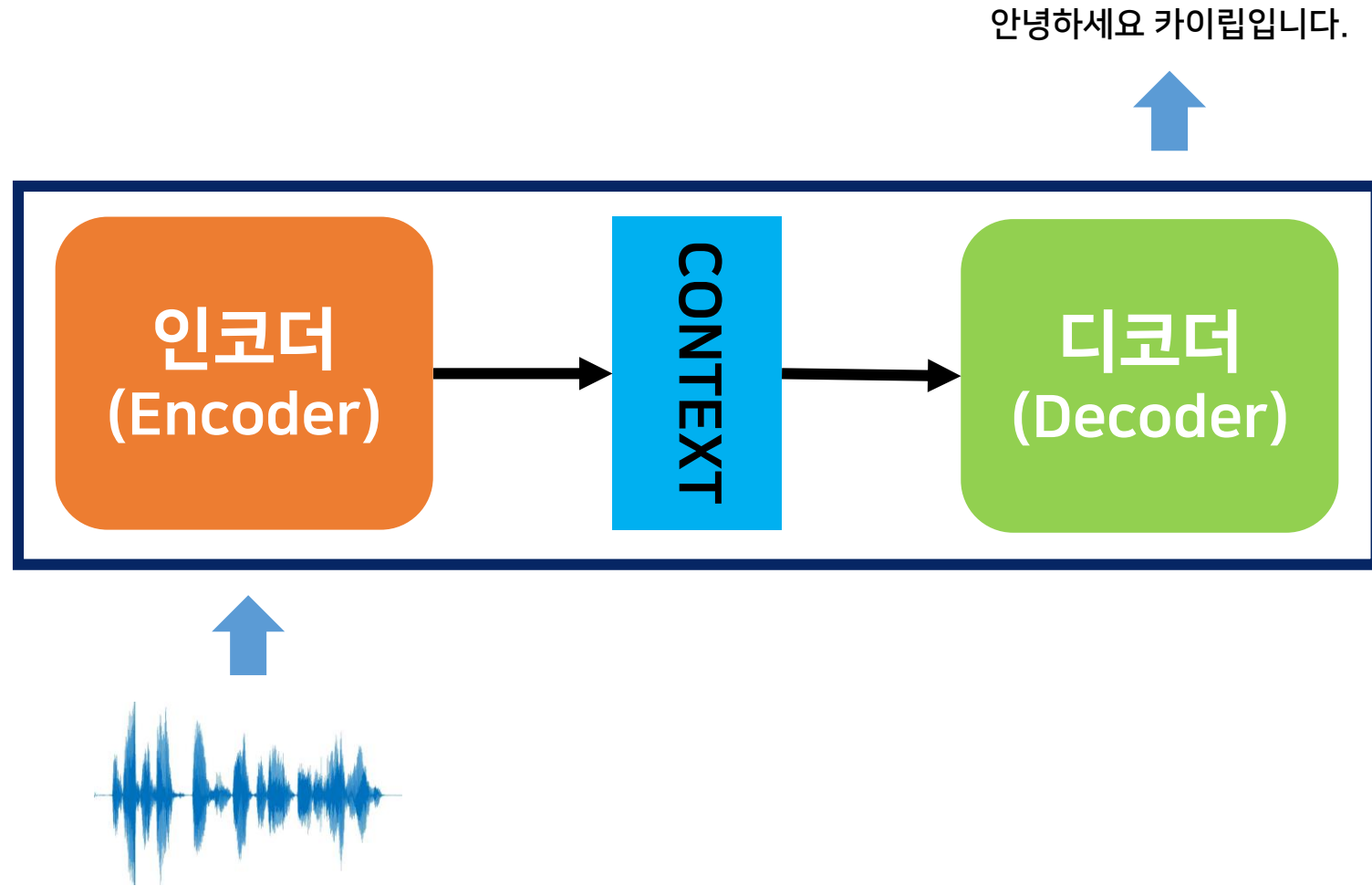
안녕하세요 카이립입니다.



음성 인식기
(SEQUENCE TO SEQUENCE)



인코더의 셀은 주황색 디코더의 색은 초록색으로 표현



인코더 (Encoder)

LSTM

LSTM

LSTM

LSTM

Hi

we

are

Kai Lib

CONTEXT

디코더 (Decoder)

LSTM

LSTM

LSTM

LSTM

안녕

하세요

카이립

입니다

Encoder는 문장을 고정 길이 벡터로 인코딩한다.

h

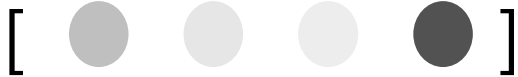
나는 고양iero소이다



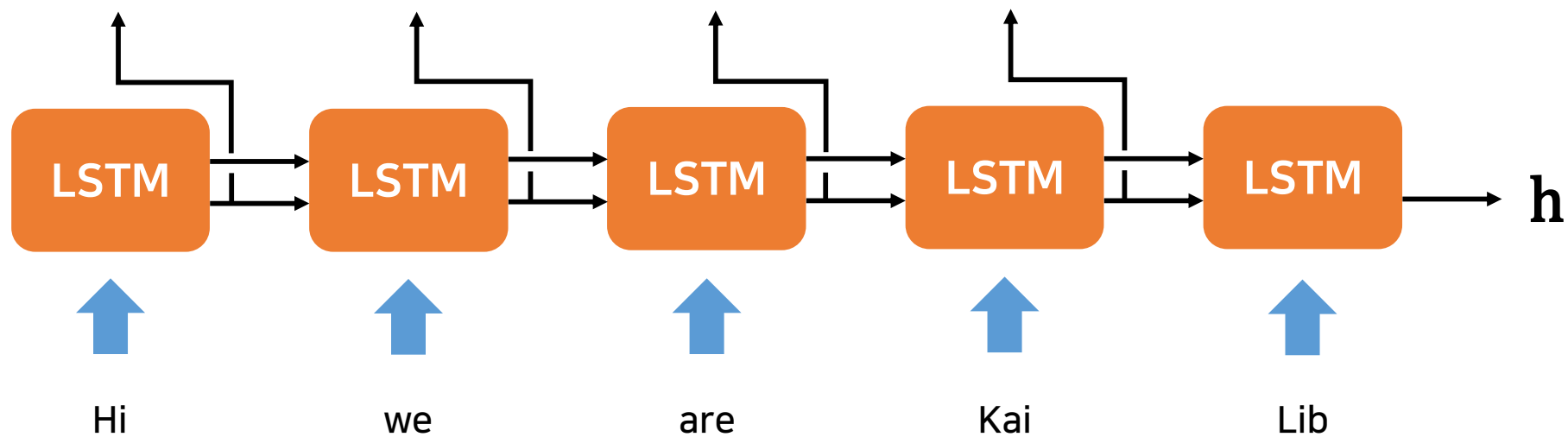
이름은 아직 없어



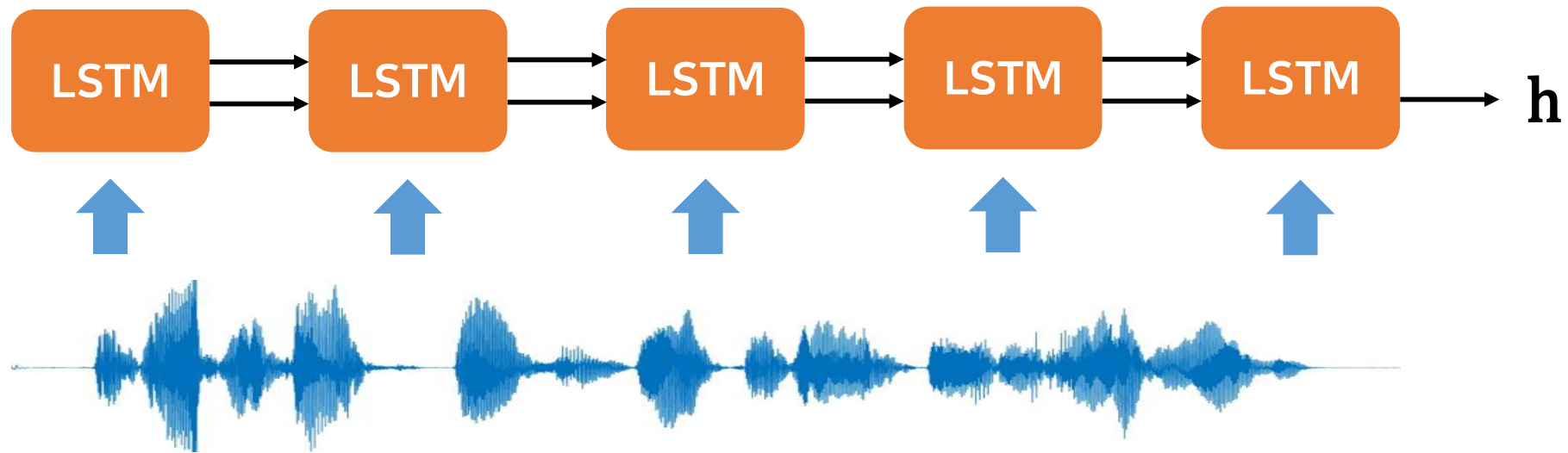
어디서 태어났는지 도무지 모르겠네



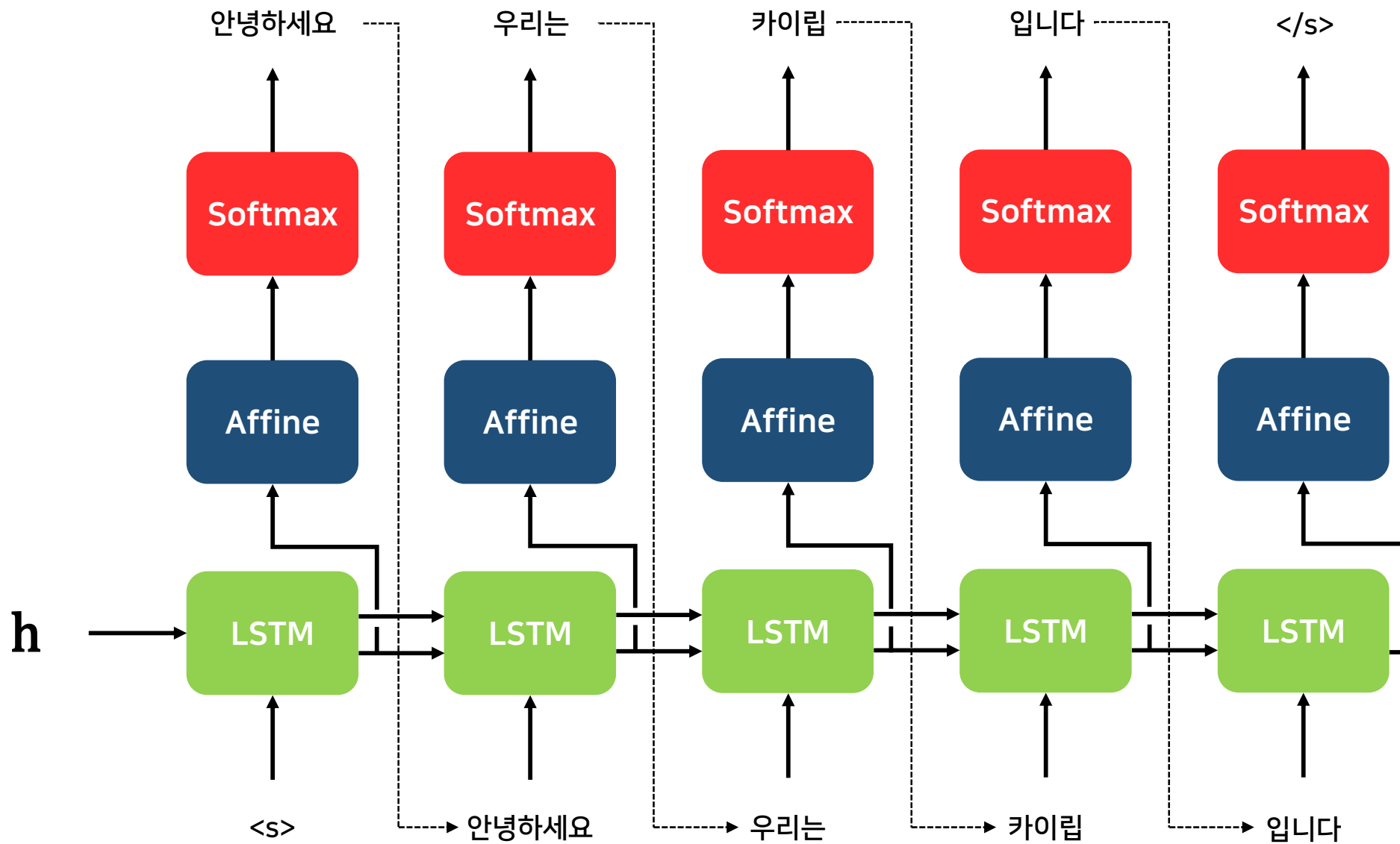
Encoder를 구성하는 계층



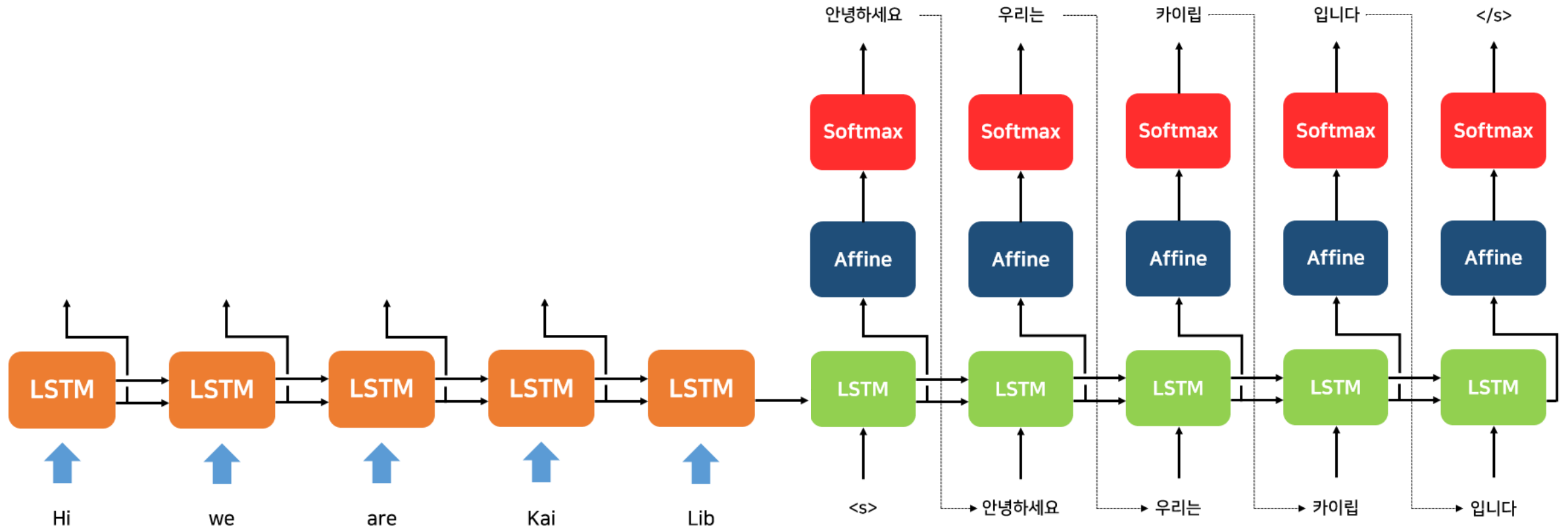
Encoder를 구성하는 계층



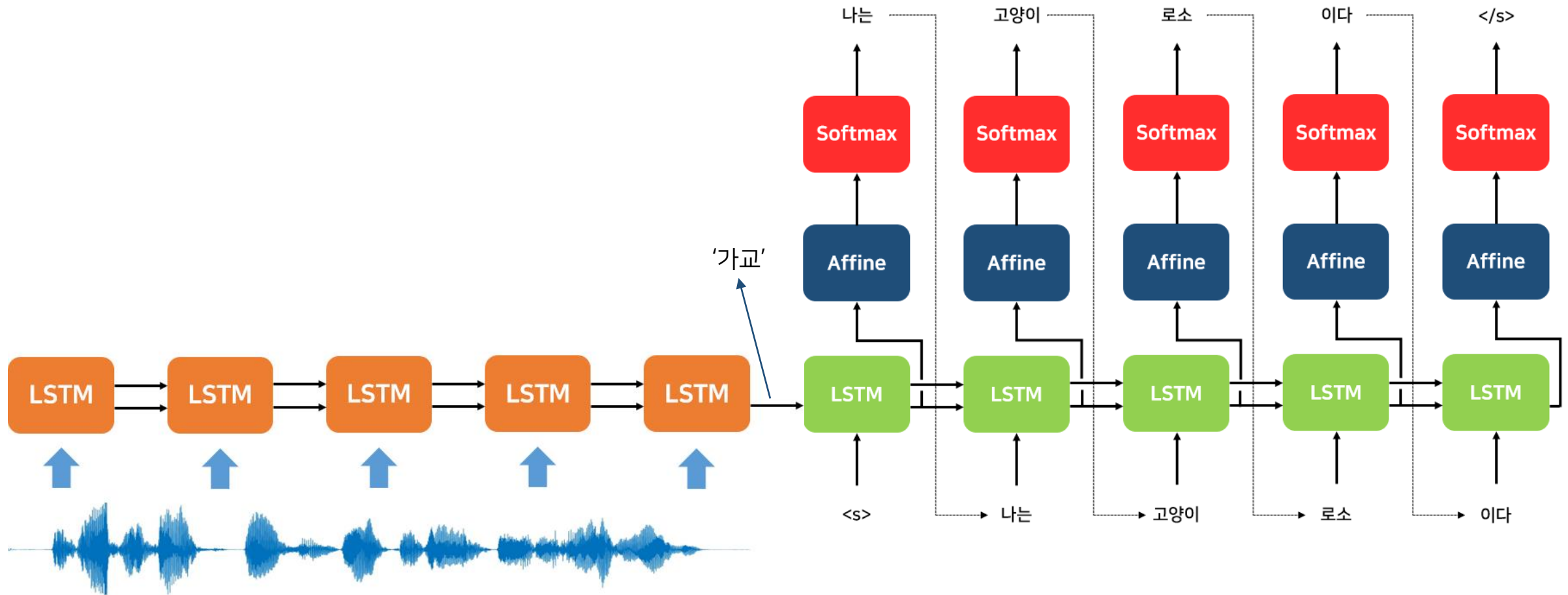
Decoder를 구성하는 계층



Seq2seq의 전체 계층 구성



Seq2seq의 전체 계층 구성



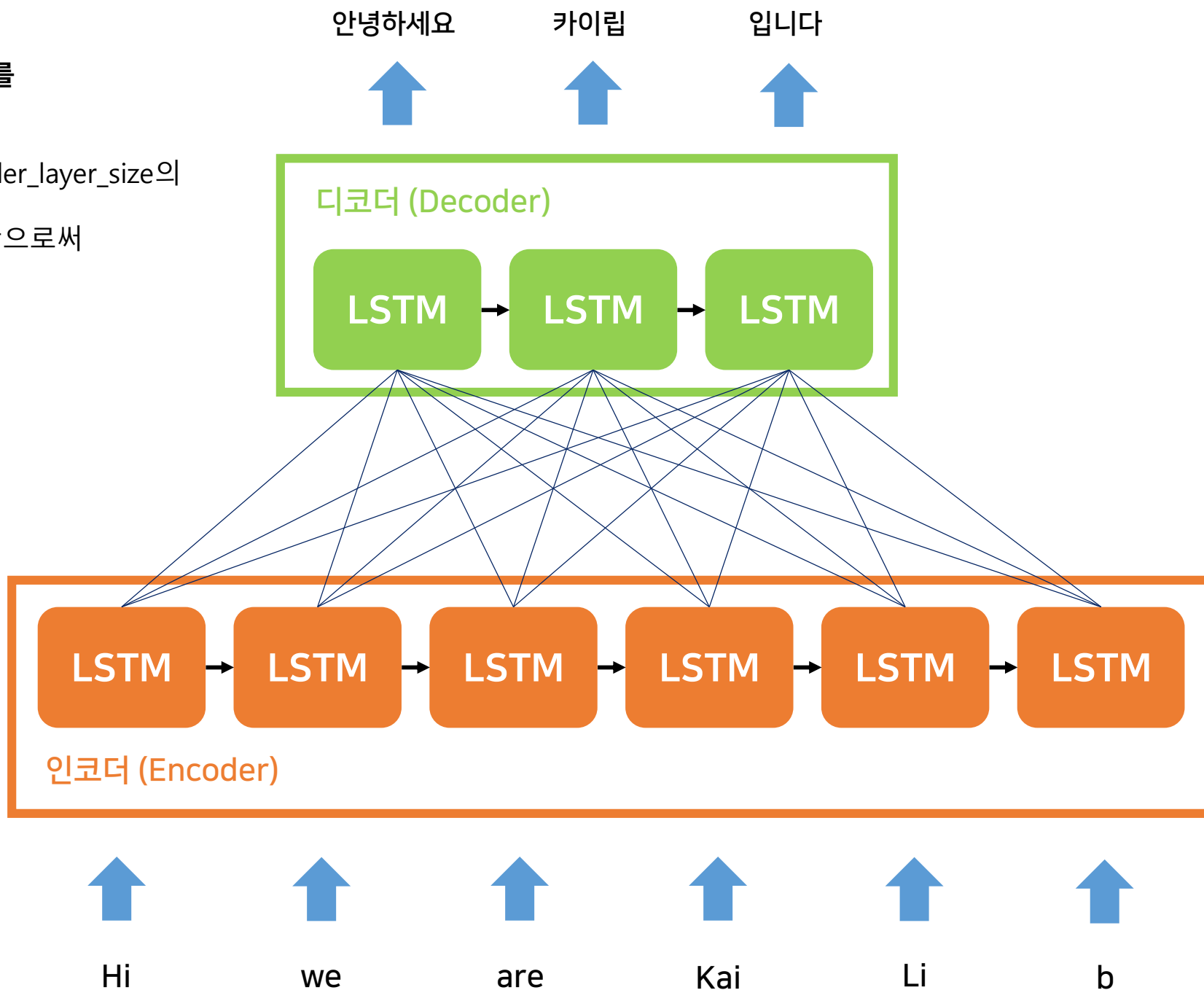
Seq2seq

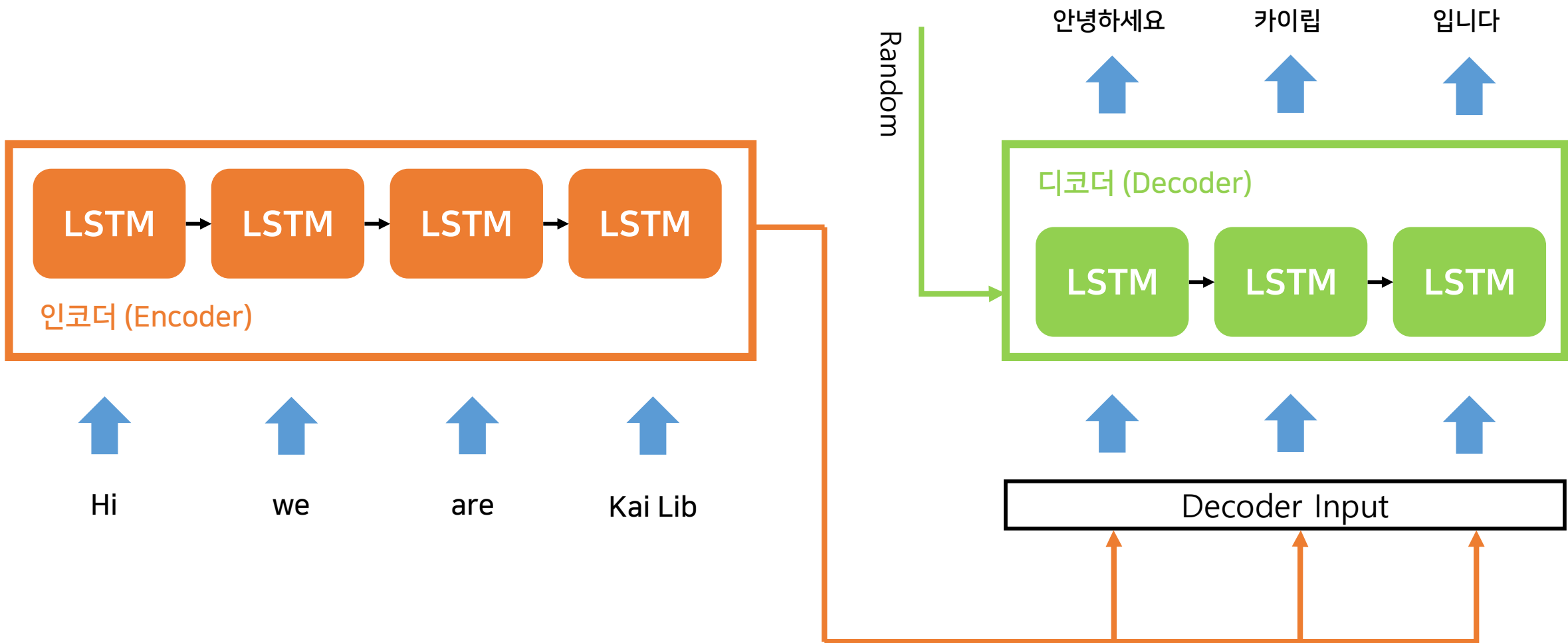
Decoder init state

Single Fully Connected Network를 Encoder와 Decoder 사이에 배치

Encoder_layer_size의 인풋과 decoder_layer_size의 아웃풋을 가지는

Fully Connected Network를 배치함으로써 서로 다른 사이즈의 인코더의 Hidden State를 이용하여 디코더의 Hidden State를 초기화 할 수 있다.





디코더의 Hidden State는 랜덤으로 초기화 한 후,
인코더의 Hidden State Output (Last Hidden State)을
디코더의 인풋에 **concatenate**한다.

Peeky Seq2seq

