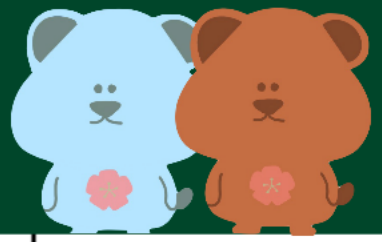
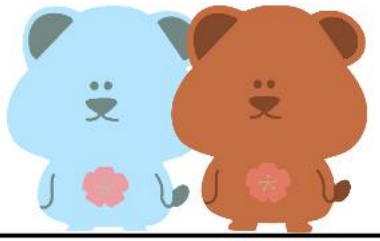


2. 자연어 처리와 딥러닝

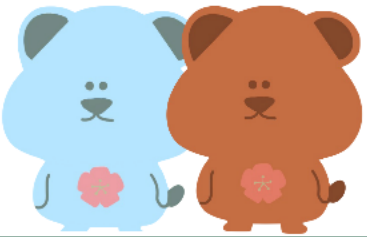
Euron 6기 중급 장서연

목차



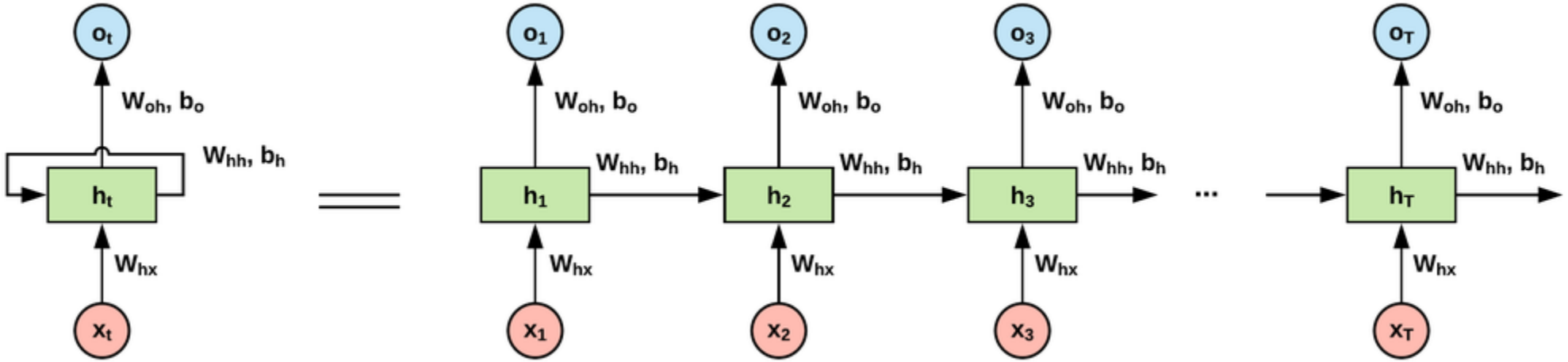
EWHA EURON

#2-1 RNN



EWHA EURON

RNN



RNN은 은닉층에서 나온 결과값이 다시 입력값이 되어 순환한다는 점에서 Recurrent Neural Network 라고 불린다

RNN

RNN은 현재 타임 스텝(h_t)에 대해 이전 스텝(h_(t-1))과 현재 입력값의 정보를 기반으로 예측 값을 산출하는 구조

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

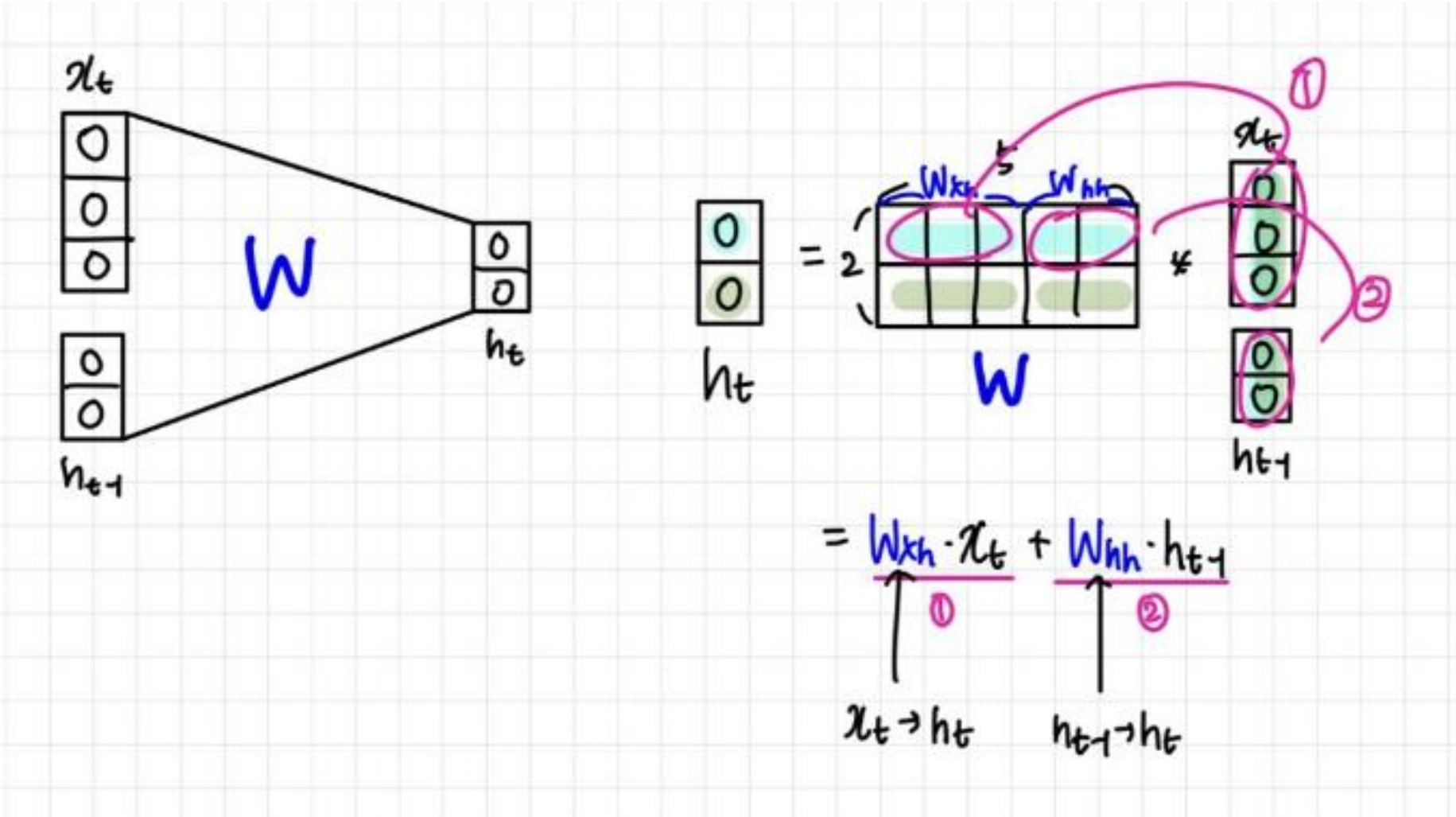
$$\downarrow$$

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

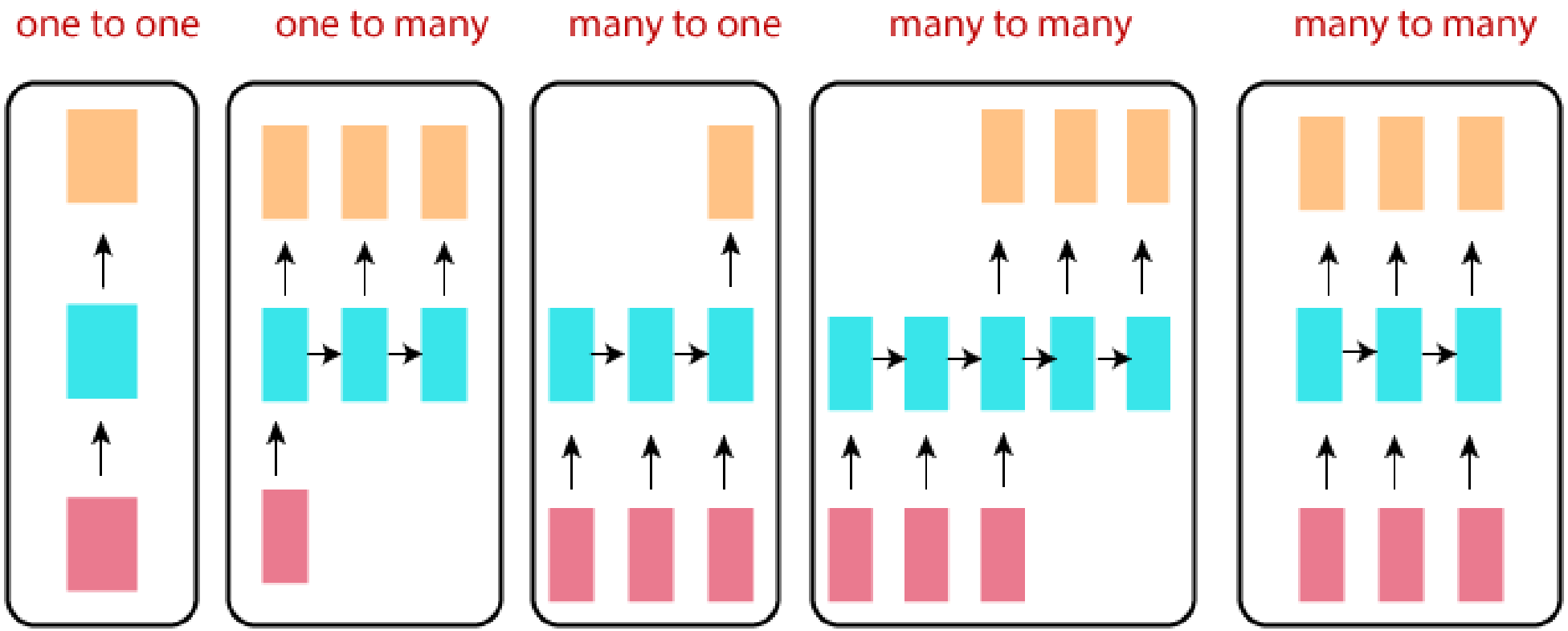
$$y_t = W_{hy}h_t$$

- h_t-1 : old hidden-state vector
- x_t : input vector at some time step
- h_t : new hidden-state vector
- f_w : RNN function with parameters W
- y_t: output vector at time step t

RNN의 hidden layer 연산

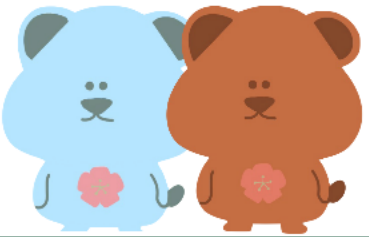


RNN의 종류와 사용 분야



Sequence data X | Image Captioning | 감정 분석 | 기계 번역 | 비디오 분류0.

#2-2 Character-level language Model



EWHA EURON

Character-level Language Model

언어 모델이란 이전에 등장한 문자0.열을 기반으로 다음 단어를 예측하는 것. 그 중에서도 character-level Language Model은 **문자 단위**로 다음에 올 문자를 예측하는 언어 모델.

Training sequence “hello”

1. **사전 구축**

   Vocabulary : [h, e, l, o]

   *단어의 개수만큼의 차원을 가지는 one-hot vector로 나타냄

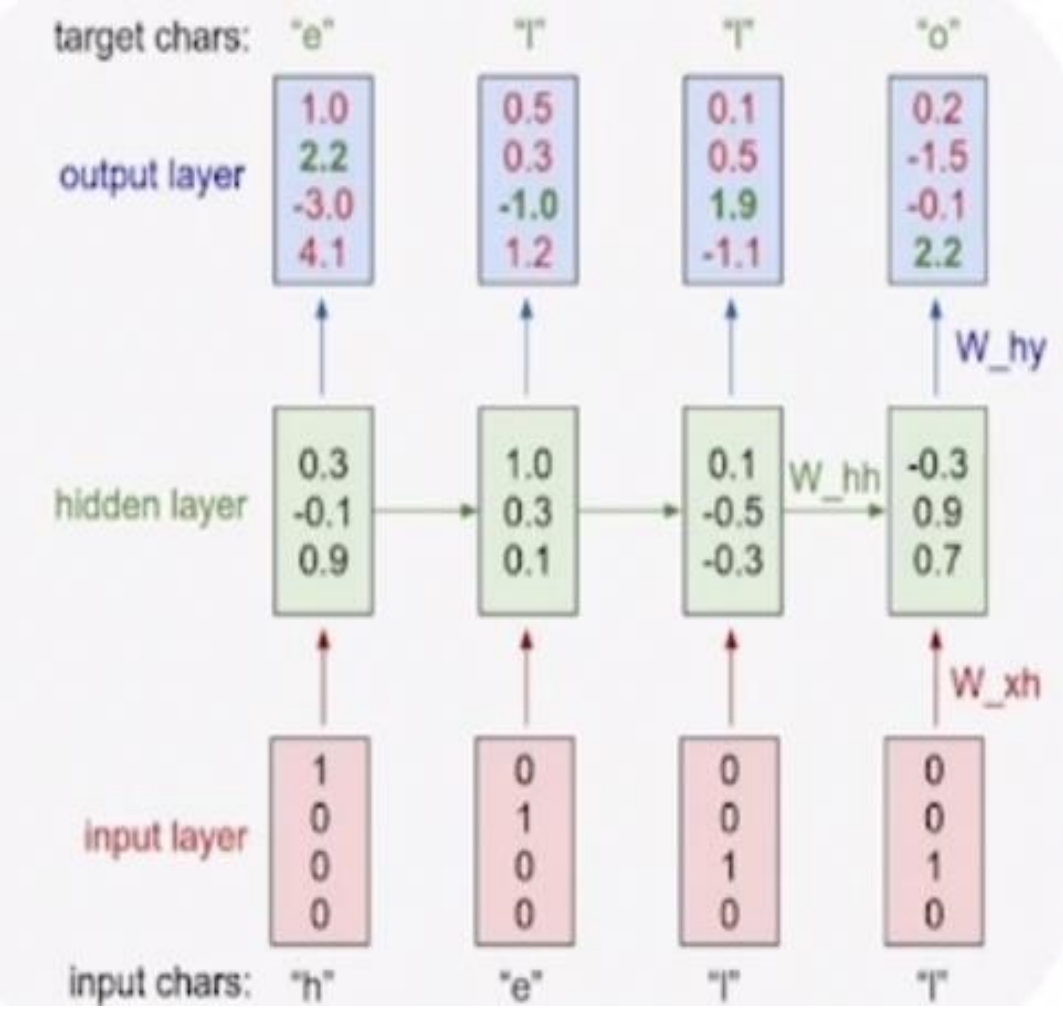
2. **h_t**

   h_t-1의 값과 현재 입력값 x의 선형 결0합 후 비선형 활성화함수(tanh)를 적용하여 h_t를 만든다.

   - 이때, h0는 모두 값이 0 인 벡터로 설정(default)

3. 결과로 나온 output vector를 **softmax layer**를 통과시켜 확률 값으로 변환 (multi-class classification)

Training sequence "hello"

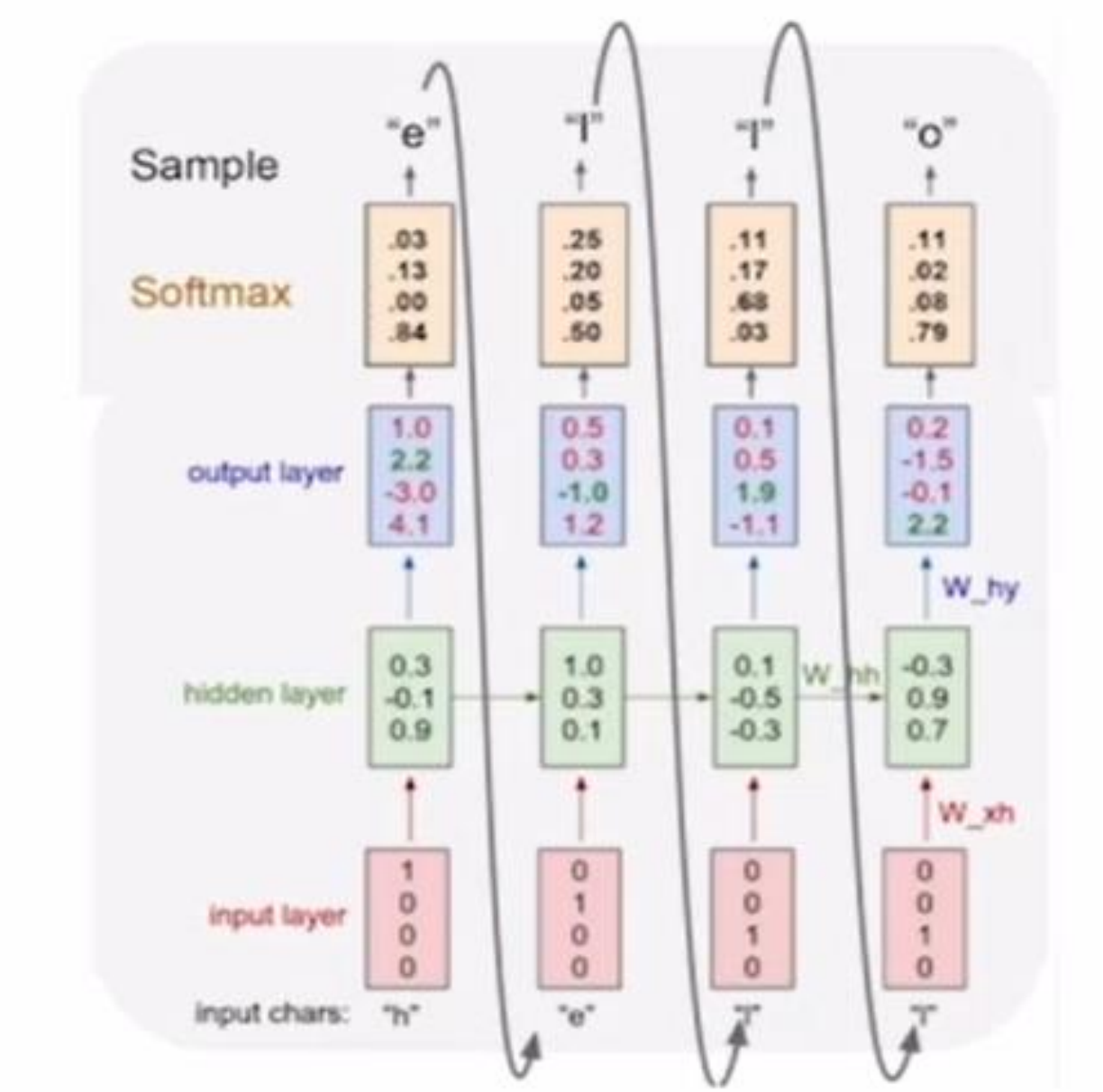


h를 입력으로 넣었을 때, 다음문자로 o를 예측하는 결과가 나온다.

(= o의 확률값이 가장 높음)

이를 e값의 확률이 높아지도록 계속 학습

Inferencing “hello”



예측을 통해 얻은 값 ‘e’가 다음 time step의 입력값이 된다.

다양한 언어모델의 예시

문단에 대한 예측 모델.

공백이나 콤마, 특수문자도 하나의 vocabulary 로 기록되어, 문단이어도 하나의 벡터로 나타낼 수 있다

by William Shakespeare

Let me not to the marriage of true minds
Admit impediments. Love is not love
Which alters when it alteration finds,
Or bends with the remover to remove:
O no! it is an ever-fixed mark
That looks on tempests and is never shaken;
It is the star to every wandering bark,
Whose worth's unknown, although his height be taken.
Love's not Time's fool, though rosy lips and cheeks
Within his bending sickle's compass come:
Love alters not with his brief hours and weeks,
But bears it out even to the edge of doom.
If this be error and upon me proved,
I never writ, nor no man ever loved.

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

train more

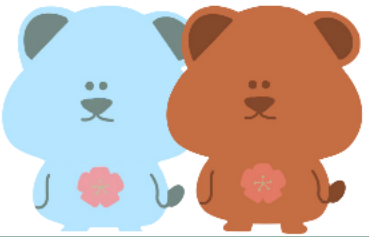
```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

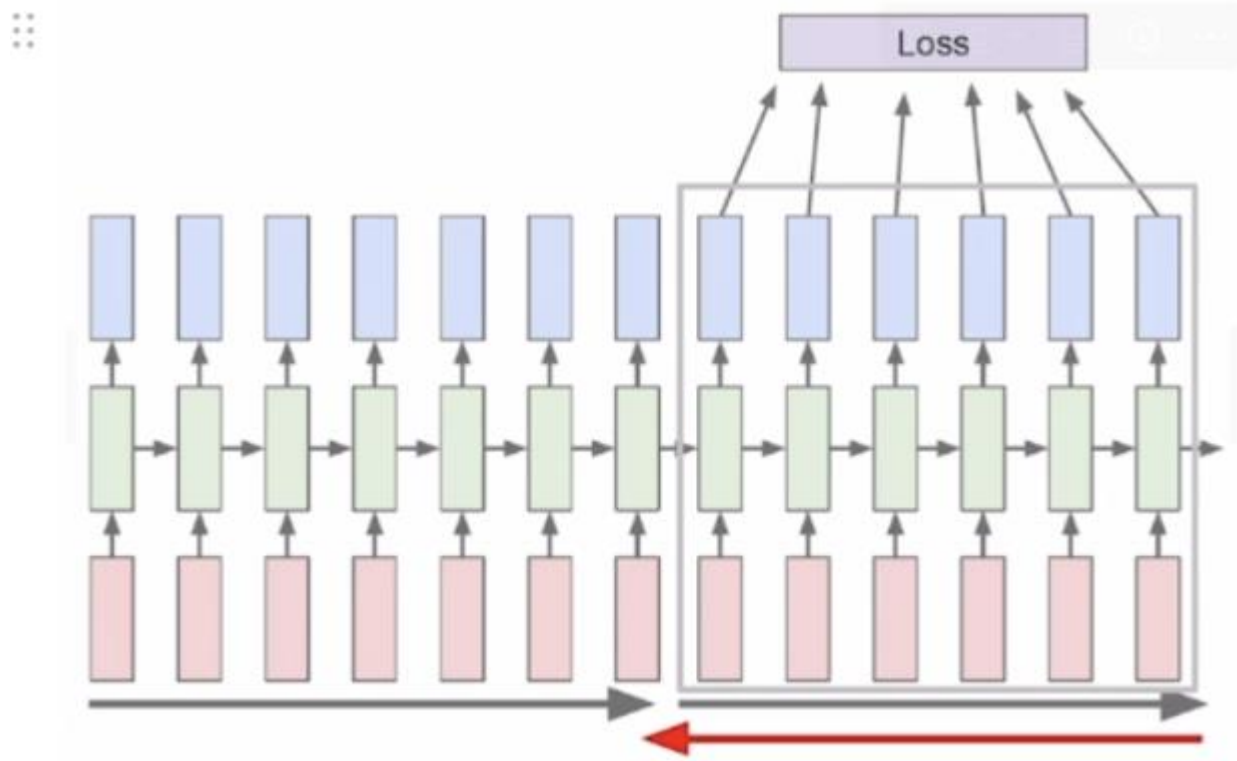
첫번째 iteration에서는 제대로 예측하지 못하는 것을 알 수 있으나, 반복할수록 성능이 향상됨을 알 수 있음

#2-3 BPTT and Long-Term-Dependency



EWHA EURON

Truncation



제한된 리소스(메모리) 내에서 모든 시퀀스를 학습할 수 없기 때문에 위와 같이 나눠서 학습에 사용하는 것

Hidden state vector h_t

이전의 데이터에 대한 정보는 ht(hidden state vector) 에 담겨져야 함.

특정 hidden state를 시각화한 그림이다.
크기가 어떻게 변화하는지를 나타낸 것. 어느 위치의 값이 유의미한지 분석하여 역추적할 수 있다.

Quote Detection Cell

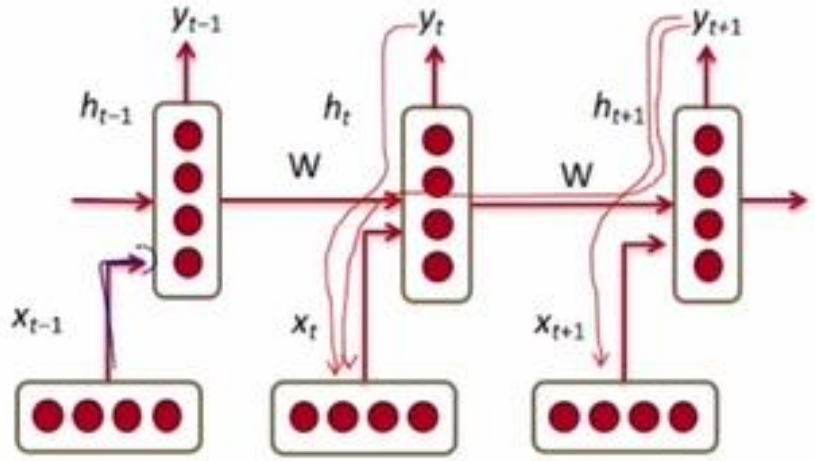
"You mean to imply that I have nothing to eat out of.... On the contrary, I can supply you with everything even if you want to give dinner parties," warmly replied Chichagov, who tried by every word he spoke to prove his own rectitude and therefore imagined Kutuzov to be animated by the same desire.

Kutuzov, shrugging his shoulders, replied with his subtle penetrating smile: "I meant merely to say what I said."

If Statement Cell

```
static int __dequeue_signal(struct sigpending *pending, sigset_t *
        siginfo_t *info)
{
  int sig = next_signal(pending, mask);
  if (sig) {
    if (current->notifier) {
      if (sigismember(current->notifier_mask, sig)) {
        if (!(current->notifier)(current->notifier_data)) {
          clear_thread_flag(TIF_SIGPENDING);
          return 0;
        }
      }
    }
    collect_signal(sig, pending, info);
  }
  return sig;
}
```

Long Term Dependency



W_hh가 반복적으로 곱해지므로 (등비수열과 같은) 기울기가 기하급수적으로 커지거나 값이 작아짐.

따라서 유의미한 값을 먼 time step까지 전달할 수 없게 됨.

Vanishing/Exploding Gradient in RNN

Toy Example

- $h_t = \tanh(w_{xh}x_t + w_{hh}h_{t-1} + b), t = 1,2,3$
- For $w_{hh} = 3, w_{xh} = 2, b = 1$

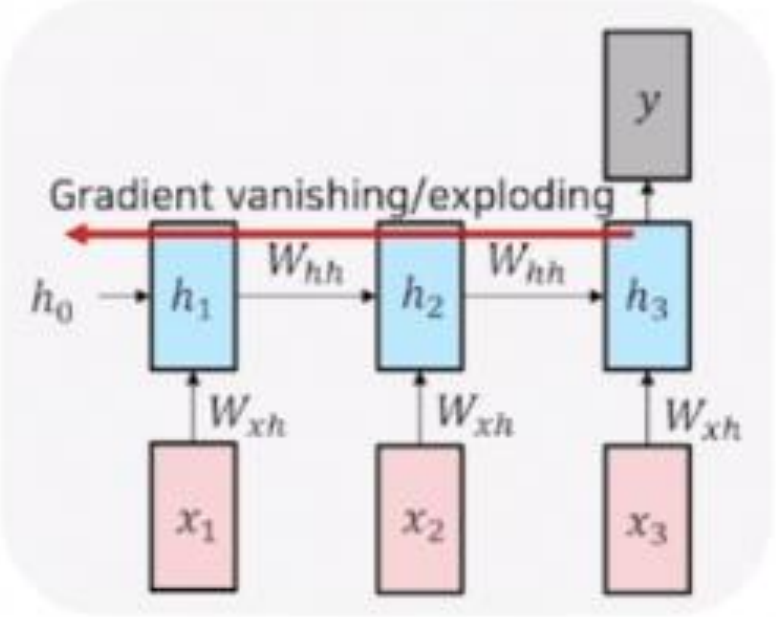
$$h_3 = \tanh(2x_3 + 3h_2 + 1)$$

$$h_2 = \tanh(2x_2 + 3h_1 + 1))$$

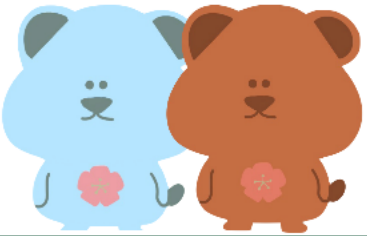
$$h_1 = \tanh(2x_1 + 3h_0 + 1))$$

...

$$h_3 = \tanh(2x_3 + 3\tanh(2x_2 + 3h_1 + 1) + 1)$$



#2-4 LSTM



EWHA EURON

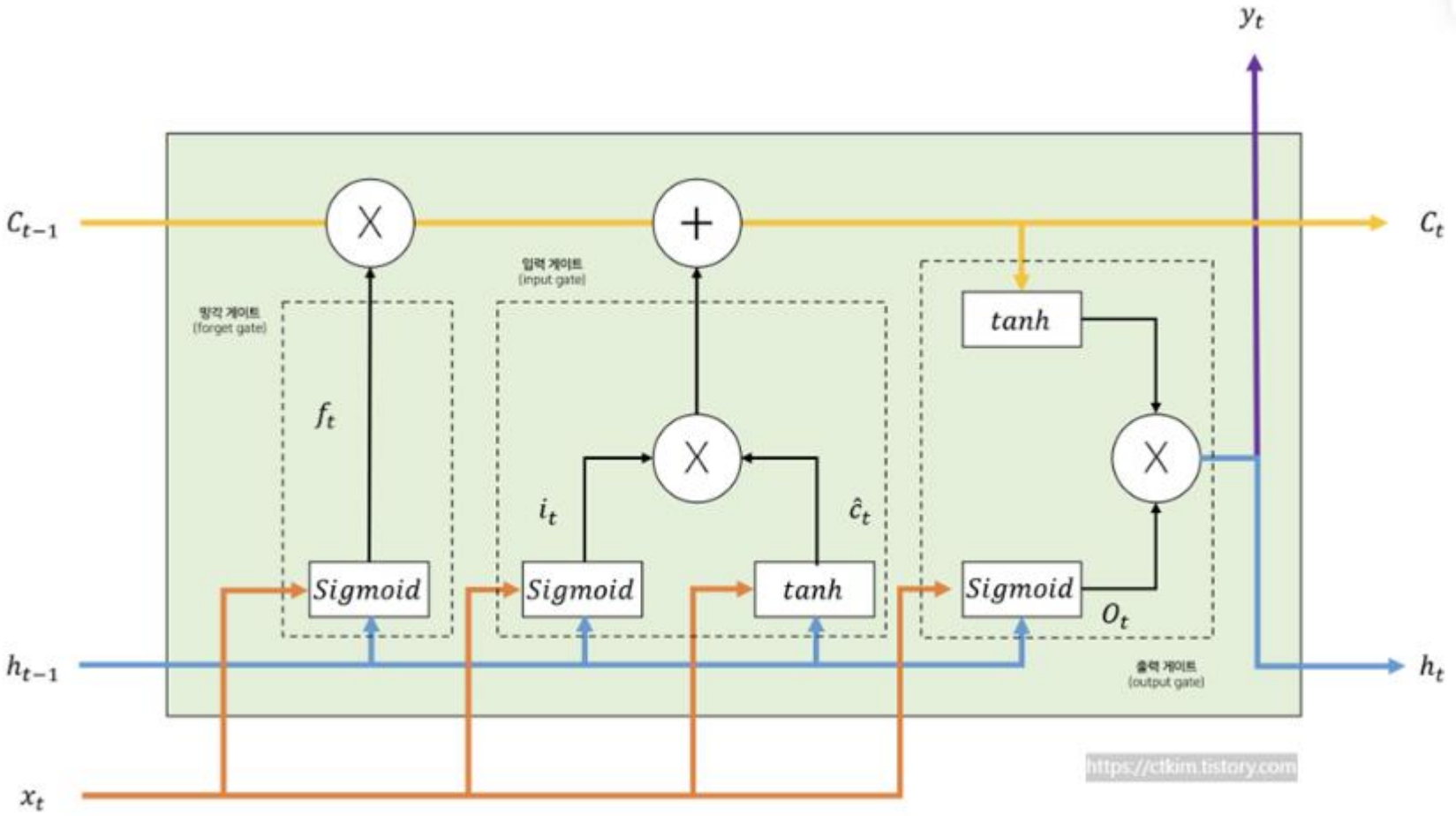
LSTM

Long Short-Term Memory

RNN의 gradient 문제를 해결하기 위해 고안된 모델

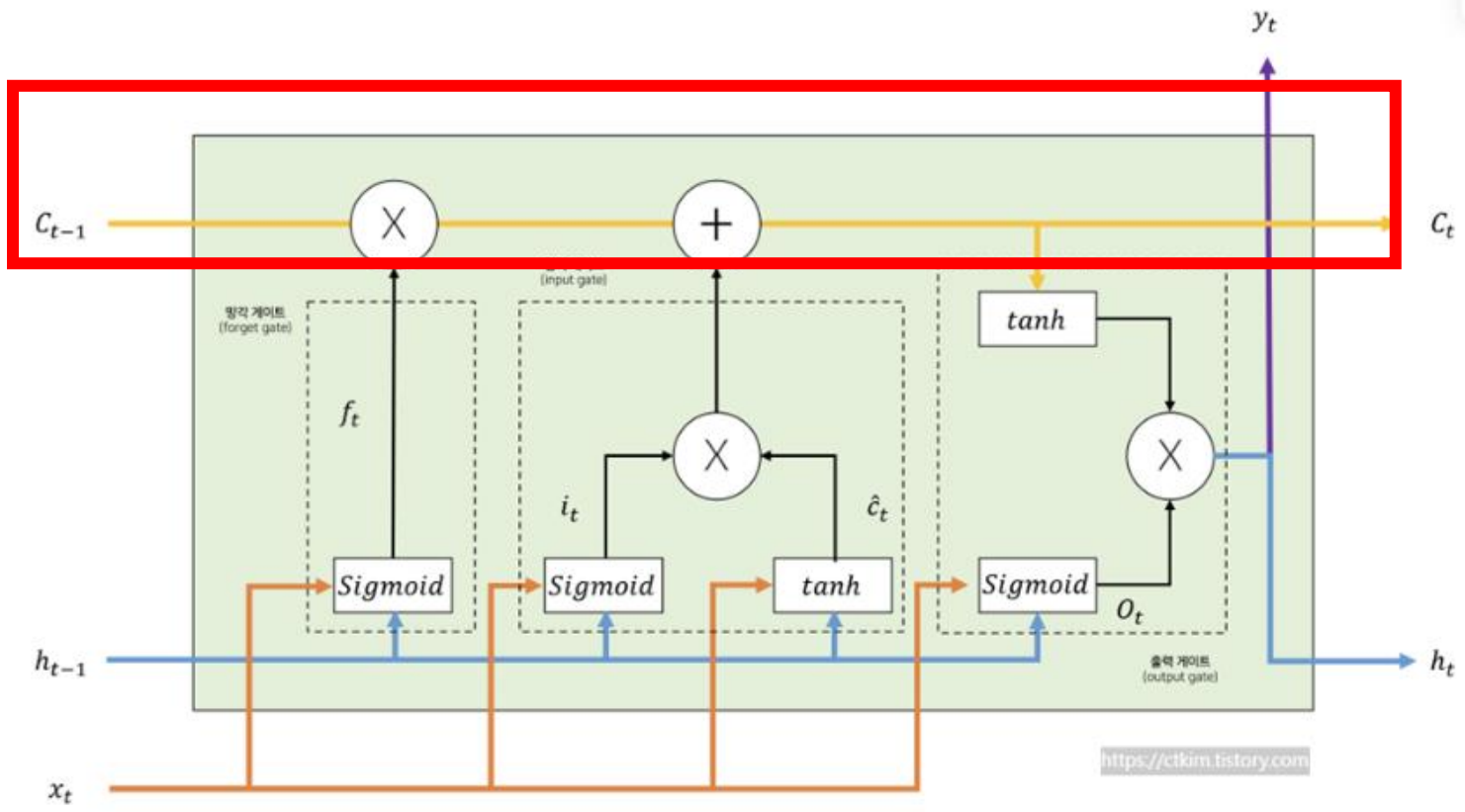
Cell state에는 핵심 정보들을 모두 담아두고, 필요할 때마다 Hidden state를 가공해 time step에 필요한 정보만 노출하는 형태로 정보가 전파.
=> Cell의 값을 얼마나 기억할지 결정하는 **게이트**가 있기 때문에 필요한 정보만 기억하도록 제어

LSTM


y_t
C_{t-1}
C_t
망각 게이트
(forget gate)
입력 게이트
(input gate)
출력 게이트
(output gate)
f_t
i_t
$\hat{c}_t$
O_t
tanh
Sigmoid
Sigmoid
tanh
Sigmoid
h_{t-1}
h_t
x_t
https://ctkim.tistory.com

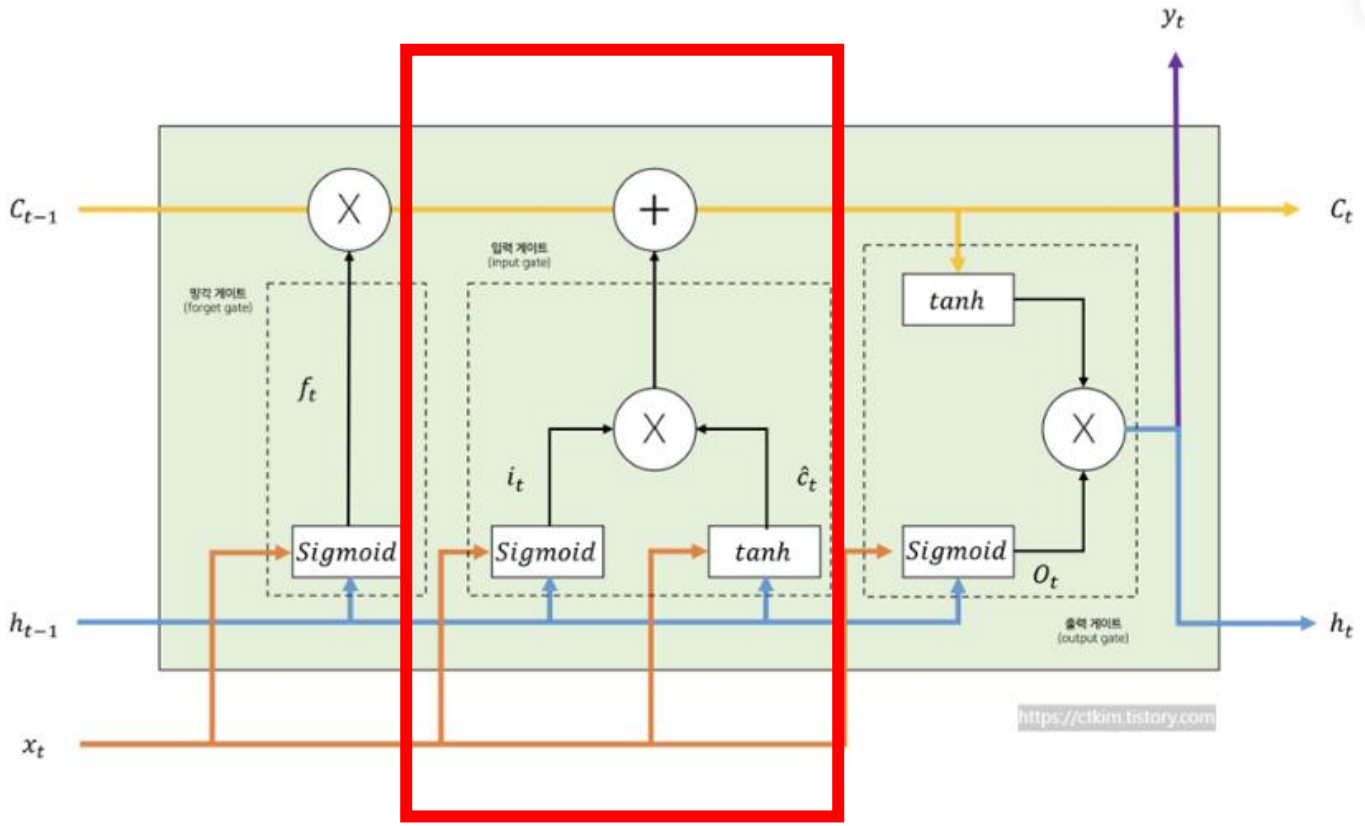
1_ Gate gate (Cell state)

어느 정도로 Cell state에 반영할 지를 -1 ~ 1 사이의 값으로 나타냄



2_ Input Gate

cell 에 쓸지 말지를 결정하는 게이트.
즉, 들어오는 input에 대해서 마지막에 sigmoid를 거쳐 0-1 사이 값으로 표현



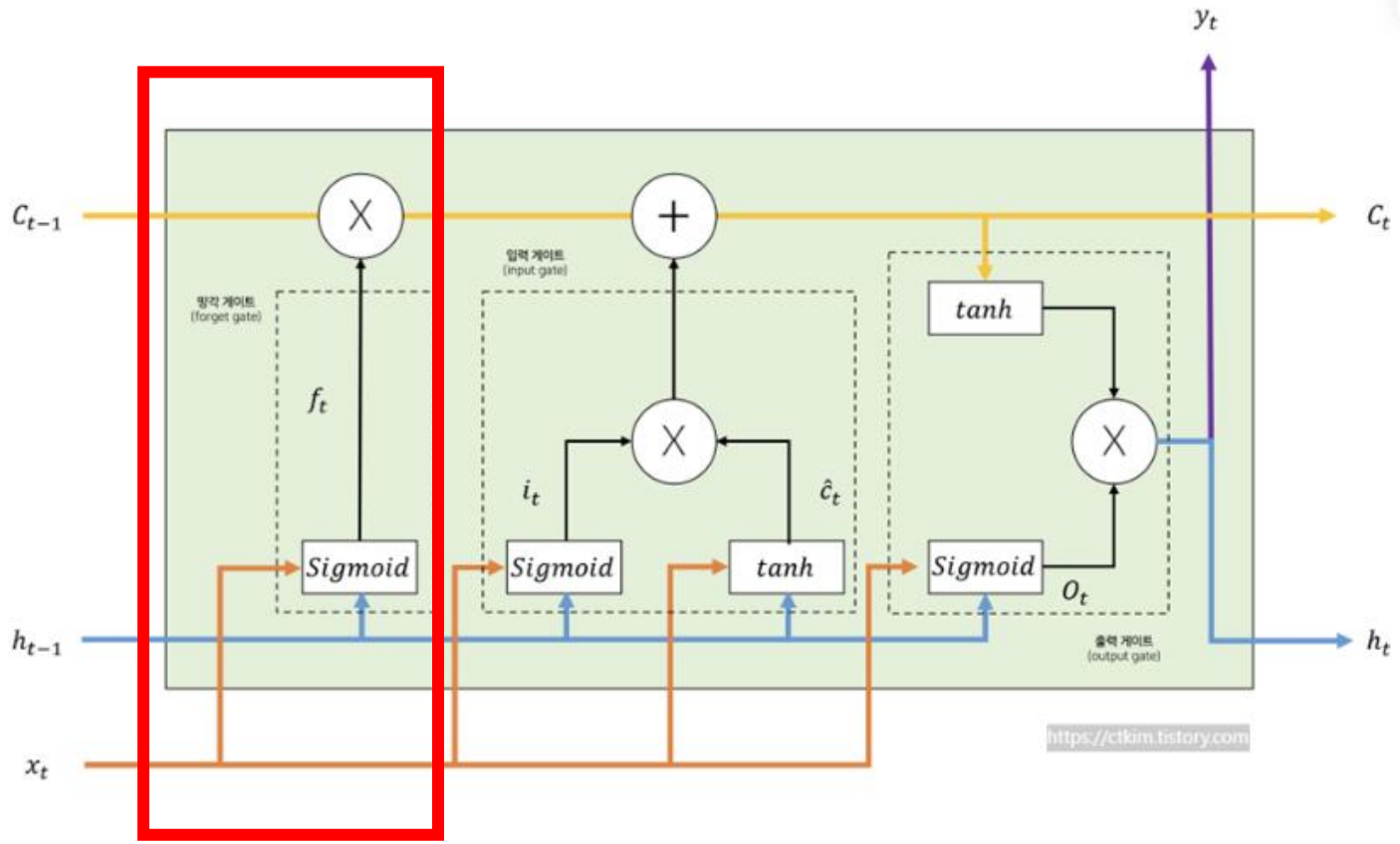
$$i_t = \sigma(W_{hi}h_{t-1} + W_{xi}x_t)$$

$$\hat{C}_t = tanh(W_{hc}h_{t-1} + W_{xc}x_t)$$

$$C_t = c_{tf} \oplus i_t \otimes \hat{c}_t$$

3_ Forget Gate

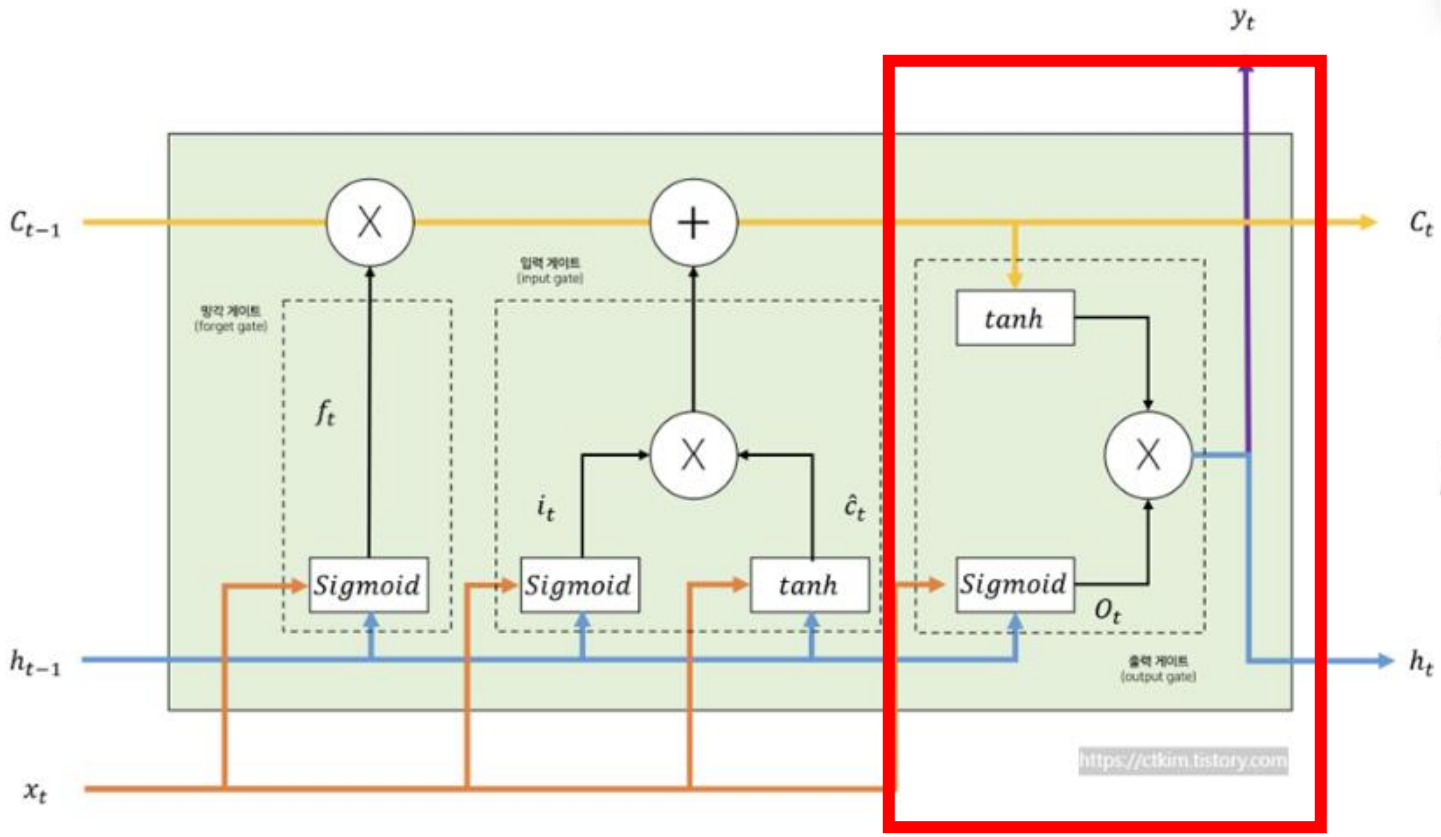
정보를 어느 정도로 지울지를 0~1사이의 값으로 나타냄
0이면 사라지고, 1이면 그대로 전달



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

4_ Output Gate

Cell 정보를 어느정도 hidden state에서 사용할 지를 0~1사이 값으로 나타냄



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

$$h_t = Ot \otimes \tanh(c_t)$$

LSTM vs RNN

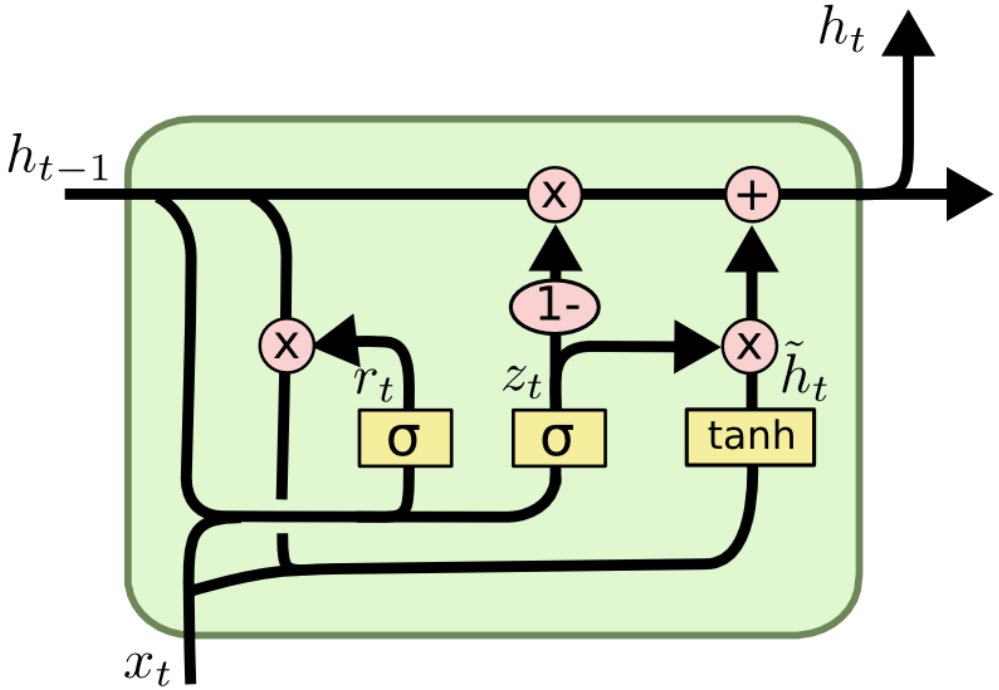
역전파 진행시 RNN은 가중치(W)를 계속해서 **곱해줌**

VS

LSTM은 forget gate를 거친 값에 대해 필요로 하는 정보를 **덧셈**을 통해 연산하여 그레디언트 소실/증폭 문제를 방지

GRU : Gated Recurrent Unit

LSTM과 유사하나, cell state와 hidden state를 합쳐 경량화한 모델



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

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

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

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