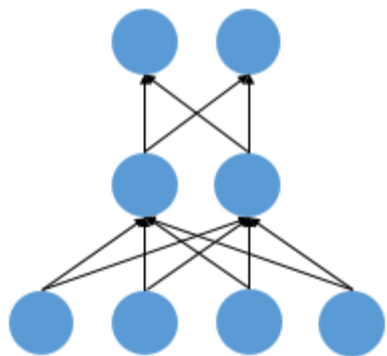


LSTM

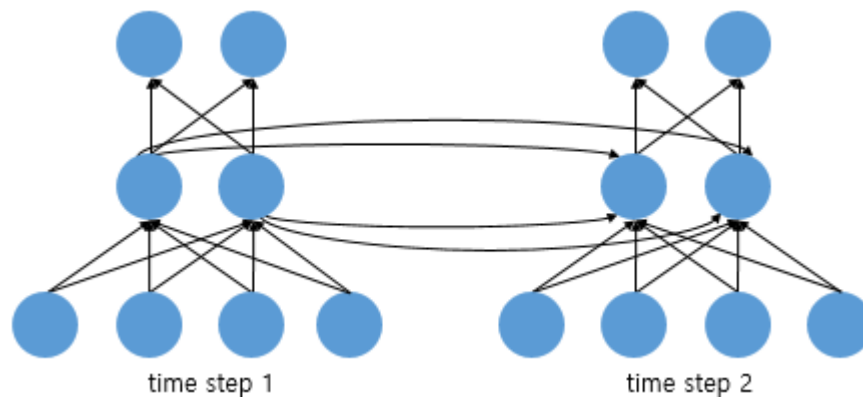
RNN (Recurrent Neural Network)

RNN은 Sequence data를 모델링 하기 위해 나온 신경망 모델

Sequence data: 순서 데이터 ex) 단어(sky), 시간(1분, 2분, ...) 음계 등...

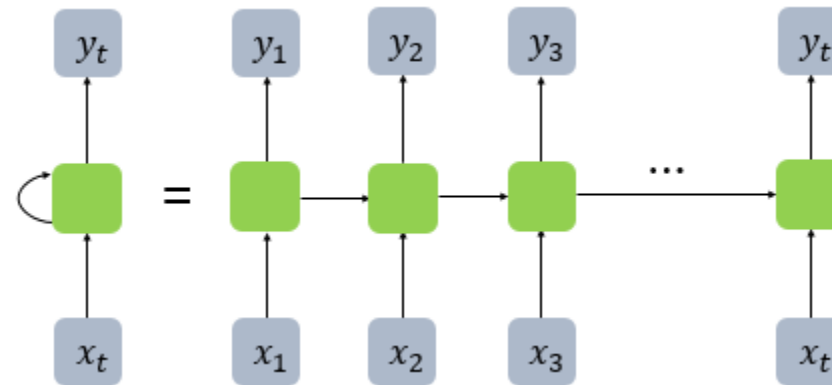
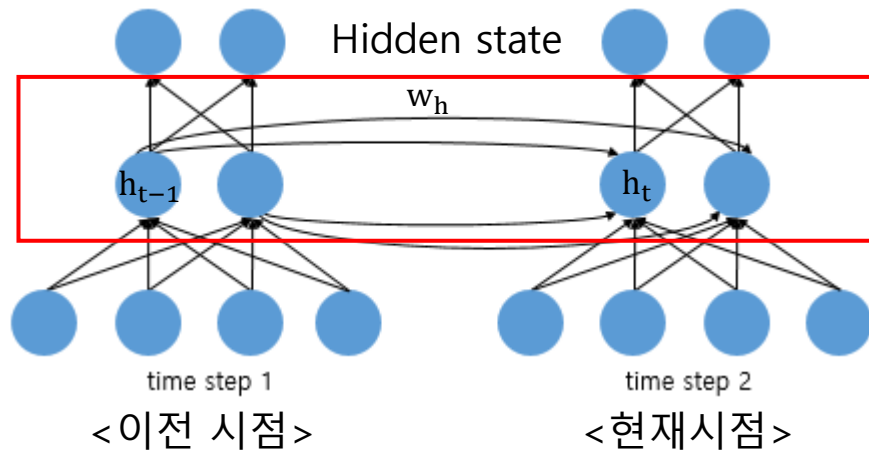


<vanilla Neural Network>



<RNN(Recurrent Neural Network)>

RNN (Recurrent Neural Network)



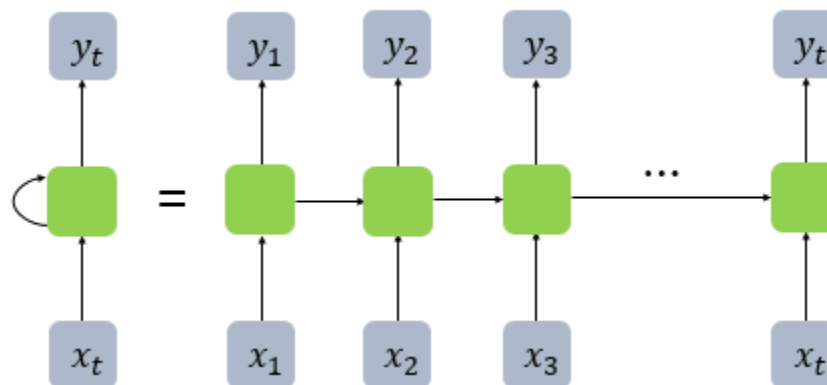
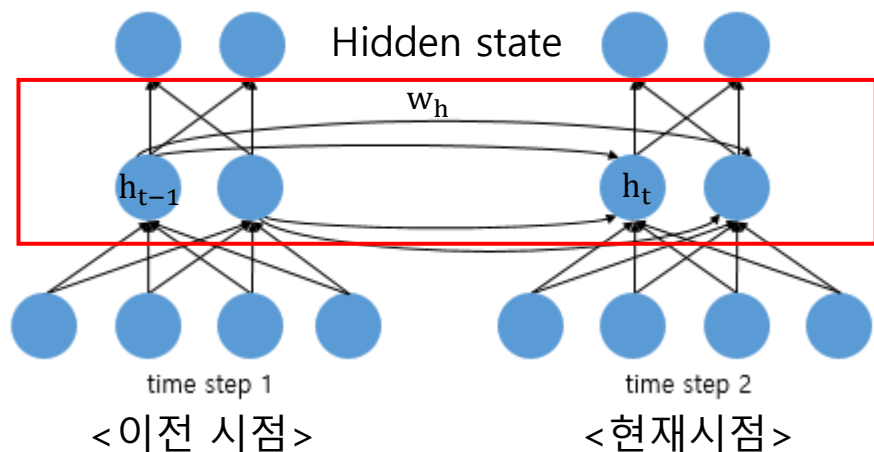
메모리 셀이 출력층 방향으로 또는 다음 시점 $t+1$ 의 자신에게 보내는 값을 은닉 상태(hidden state)라고 한다.

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

출력층: $y_t = \tanh(W_y h_t + b)$

RNN (Recurrent Neural Network)

RNN의 한계점

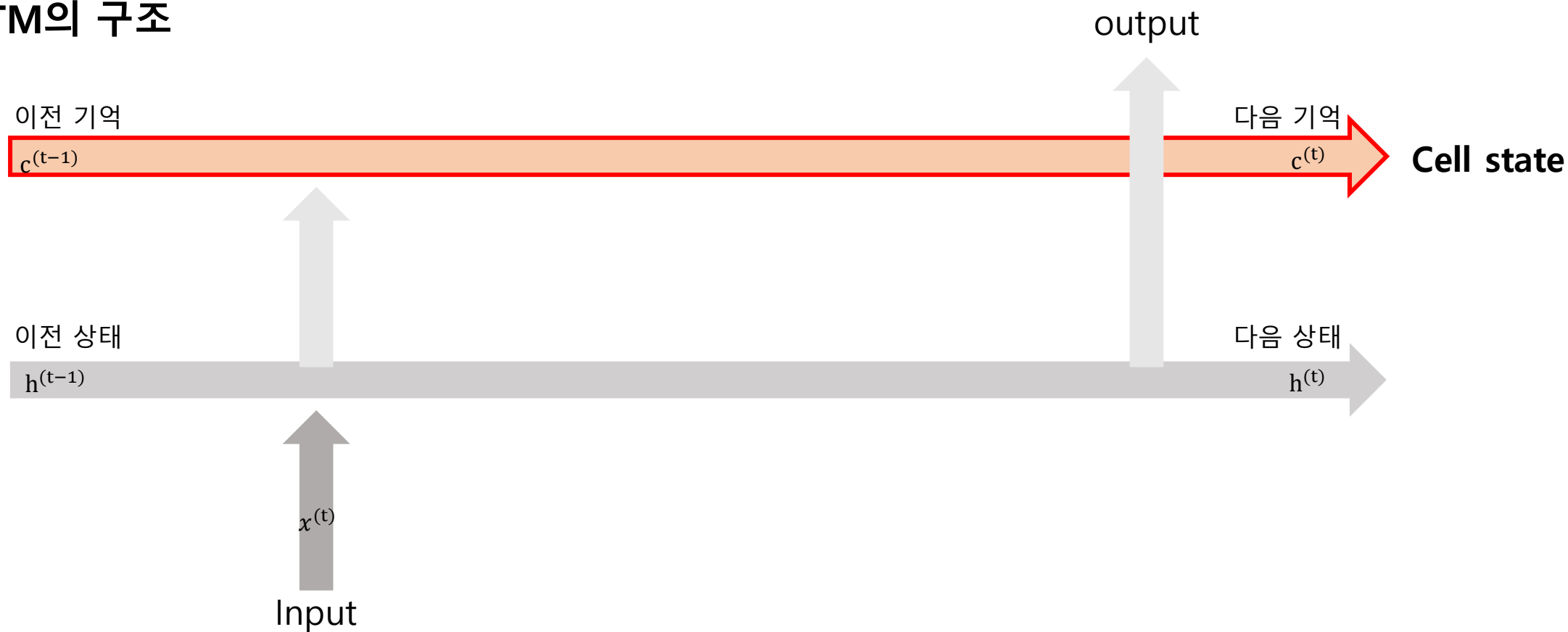


이론적으로는 RNN은 긴 시간의 데이터를 완벽하게 다룰 수 있다고는 하지만 실제로는 term이 길 경우 성능이 좋지 않은 문제점이 있음.

-> 장기 의존성 문제(은닉층의 과거의 정보가 마지막까지 전달되지 못하는 현상)

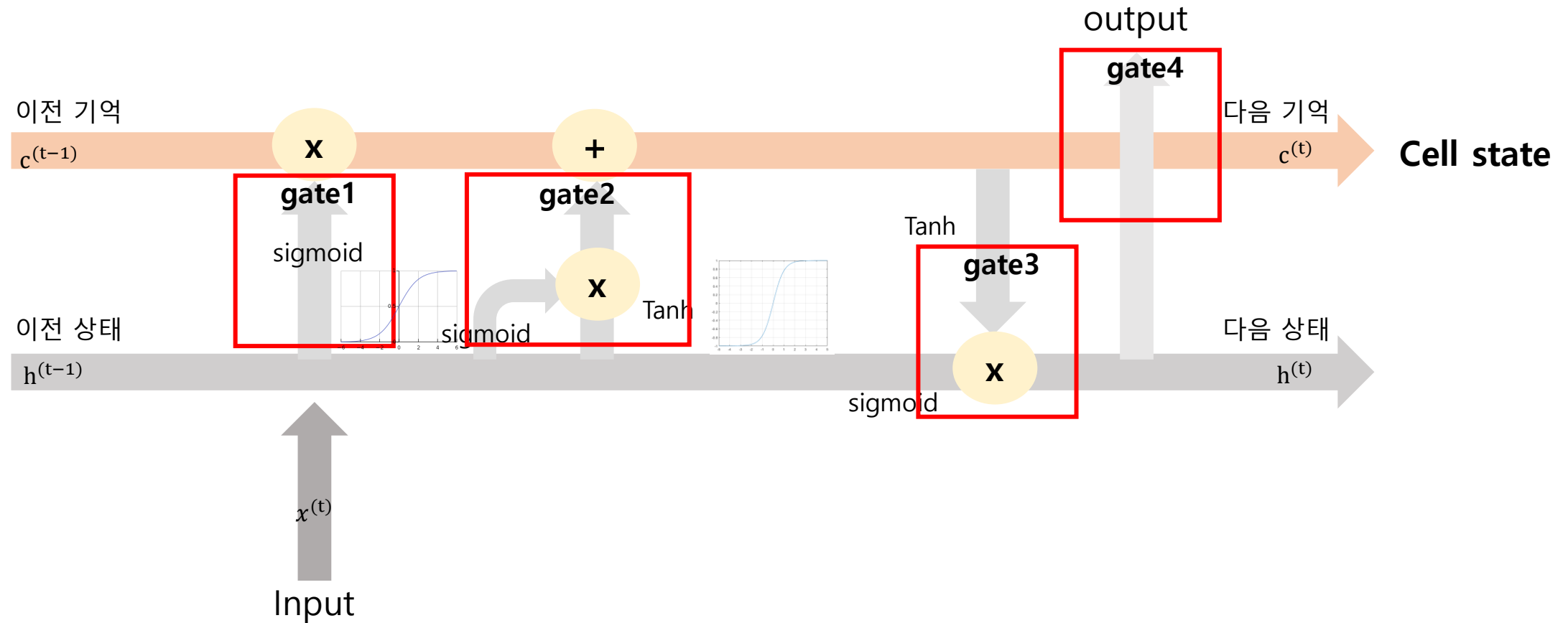
LSTM

LSTM의 구조



이전 단계의 정보를 메모리 셀이라는 공간에 저장해서 현재의 단계를 계산할 때 반영을 한다.

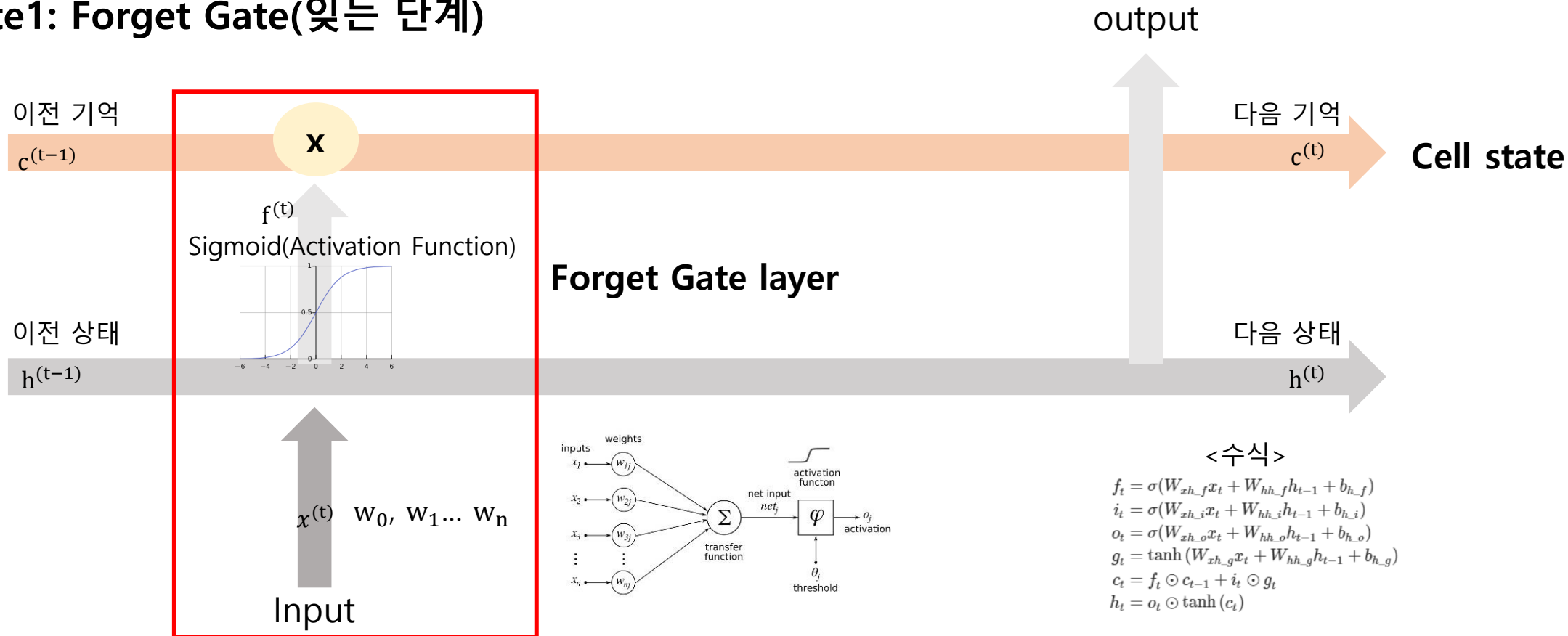
LSTM



LSTM은 4개의 gate를 이용해서 메모리셀을 제어한다.
Forget Gate, Input Gate, Update Gate, Output Gate

LSTM

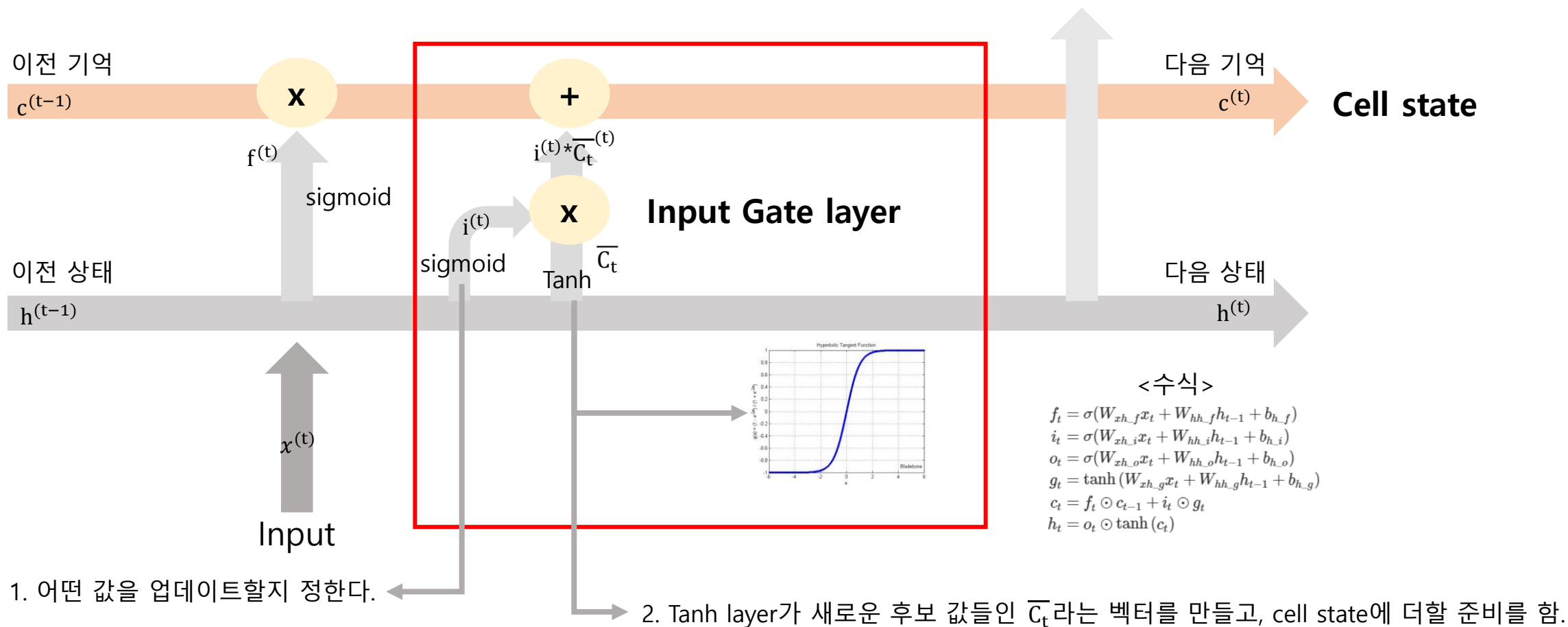
Gate1: Forget Gate(잊는 단계)



Activation Function으로 나온 값이 1에 가깝다면 정보를 많이 활용할 것이고,
0에 가깝다면 과거 정보를 삭제함.

LSTM

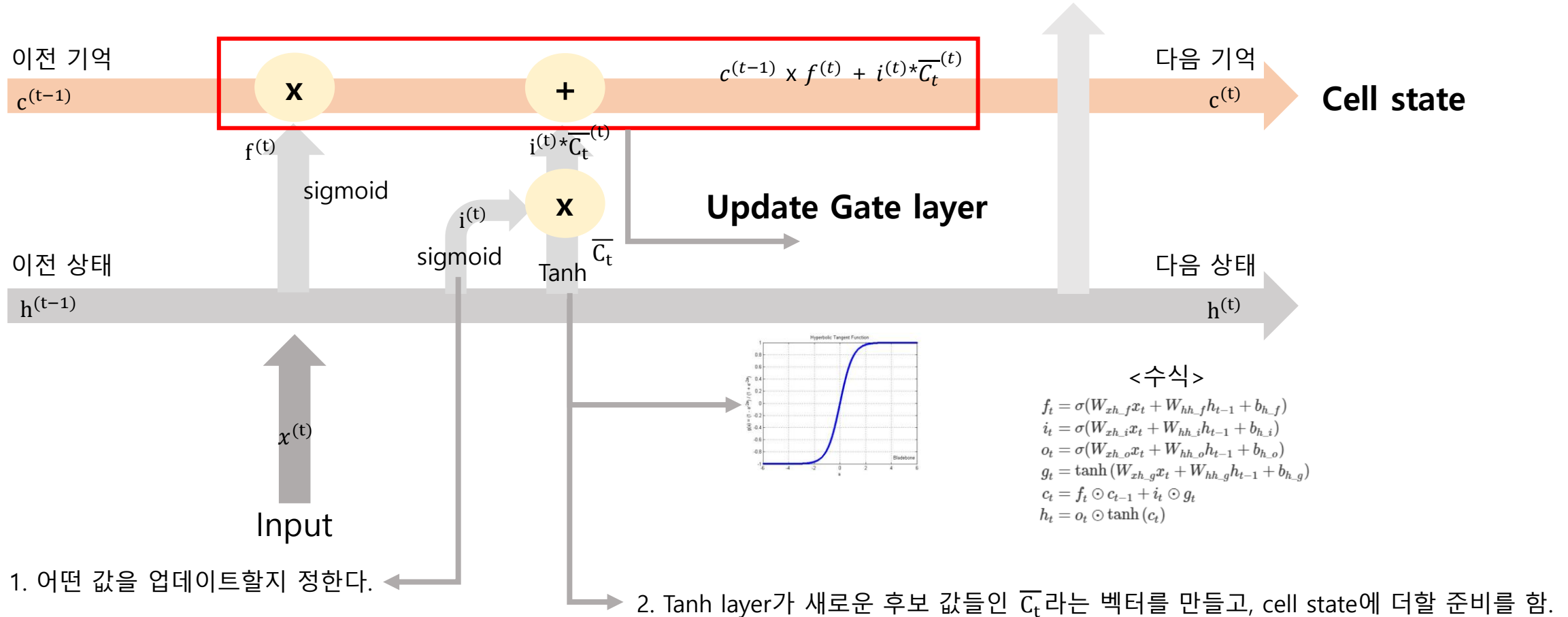
Gate2: Input Gate(기억하는 단계)



Sigmoid로 나온 벡터와 tanh로 나온 벡터를 합쳐서 cell state를 업데이트 할 자료를 만든다.

LSTM

Gate3: Update Gate(메모리셀을 업데이트 하는 단계)

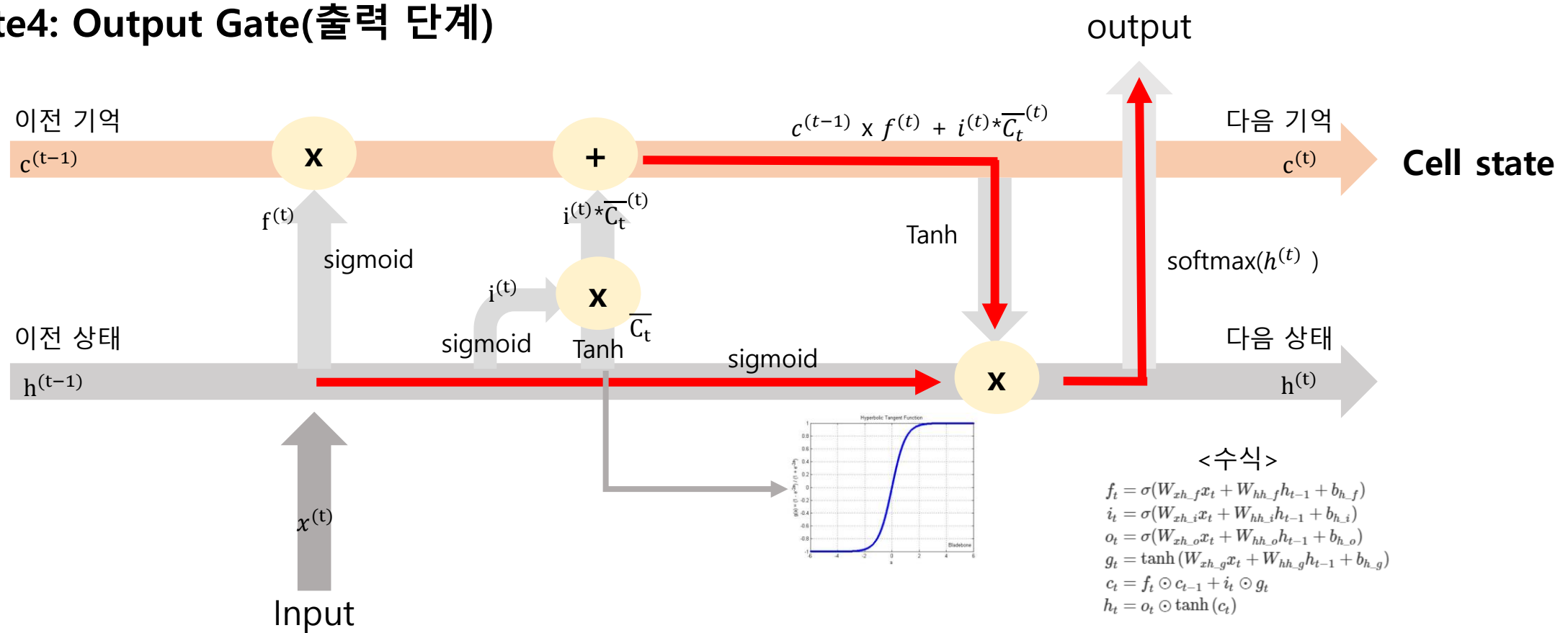


Forget Gate에서 계산된 만큼 잊고, 현시점의 정보 후보에 input gate의 중요도를 weight로 준다.

$$c^{(t)} = c^{(t-1)} \times f^{(t)} + i^{(t)} * \bar{C}_t^{(t)}$$

LSTM

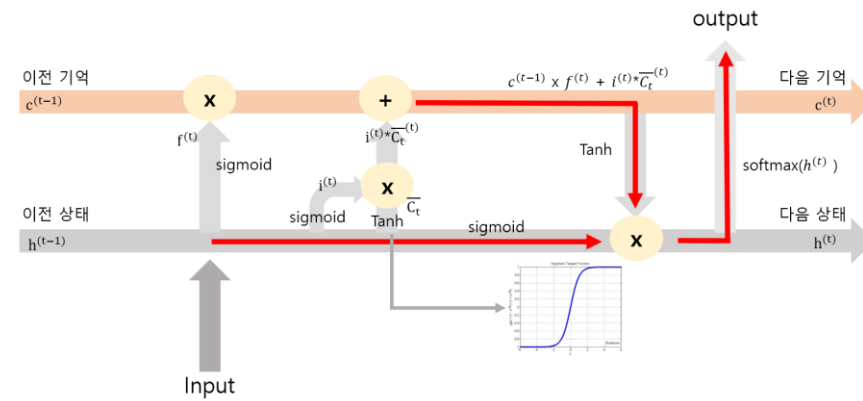
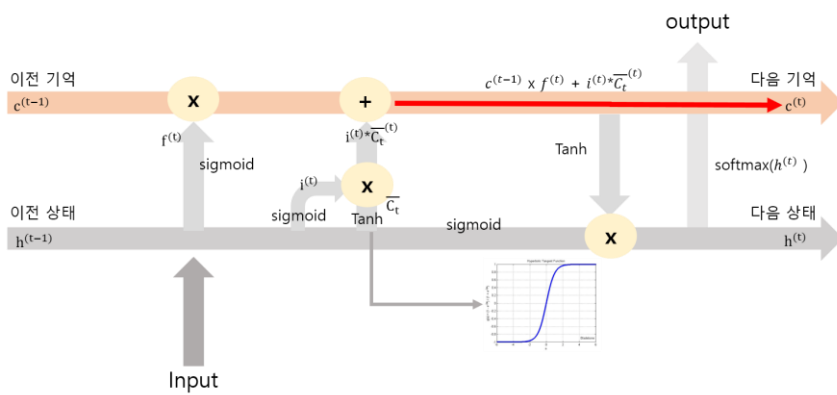
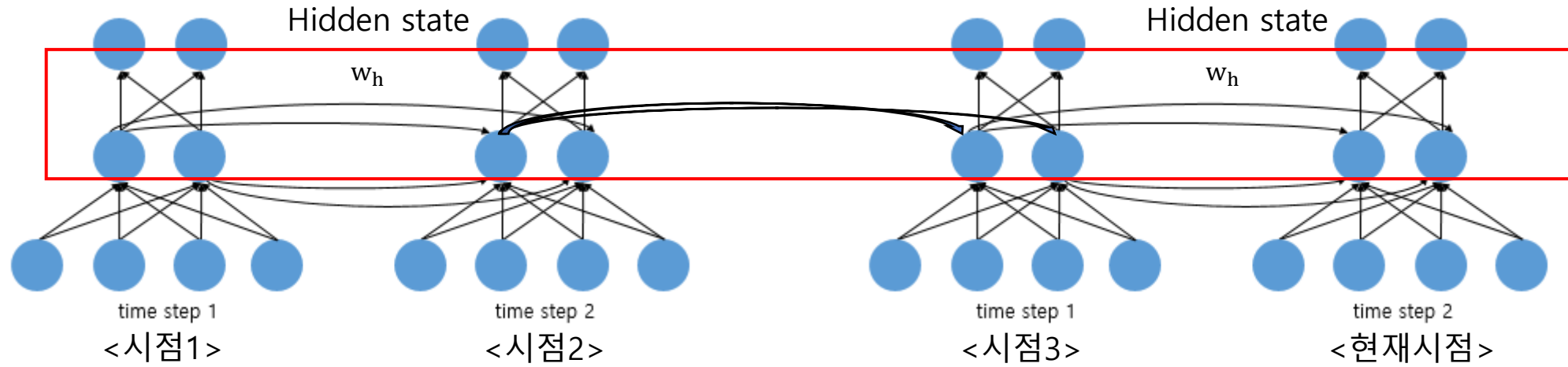
Gate4: Output Gate(출력 단계)



$$h^{(t)} = \text{sigmoid}(W[h^{(t-1)}, (x^{(t)})]) \times \text{Tanh}(c^{(t)})$$

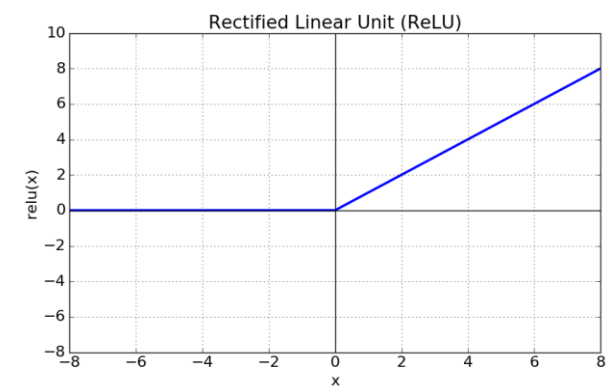
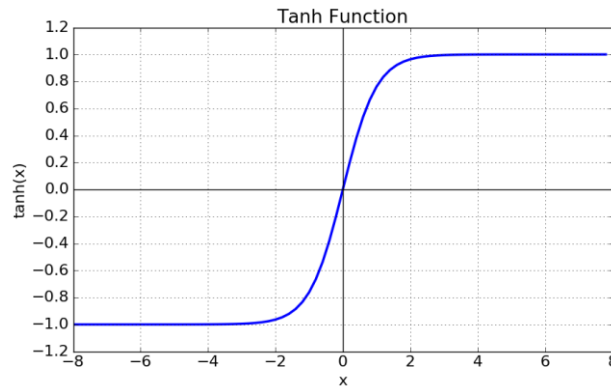
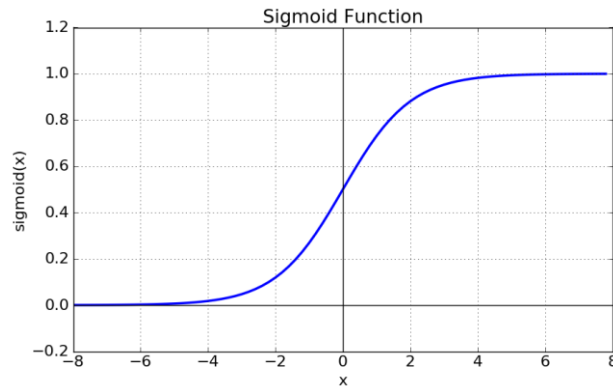
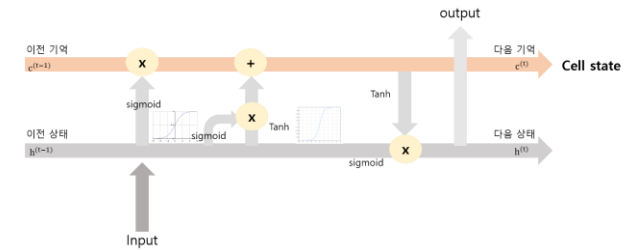
다음 상태 = 현재까지의 상태 x 현재까지 축적된 기억

LSTM



LSTM

왜 RNN 내부에서 Activation Function을 Sigmoid나 tanh를 사용하는가?



RNN 계열은 계속 같은 layer를 여러 번 반복하는 성질을 가지고 있기 때문에 1보다 큰 값이 들어오게 되면 반복하면서 값이 너무 커지는 현상이 발생하기 때문이다.

참고자료

<https://blog.naver.com/brink0/221559324924>

<https://yjjo.tistory.com/17>

<https://dgkim5360.tistory.com/entry/understanding-long-short-term-memory-lstm-kr>

<https://gruuuuu.github.io/machine-learning/lstm-doc/#>

Thank You
