

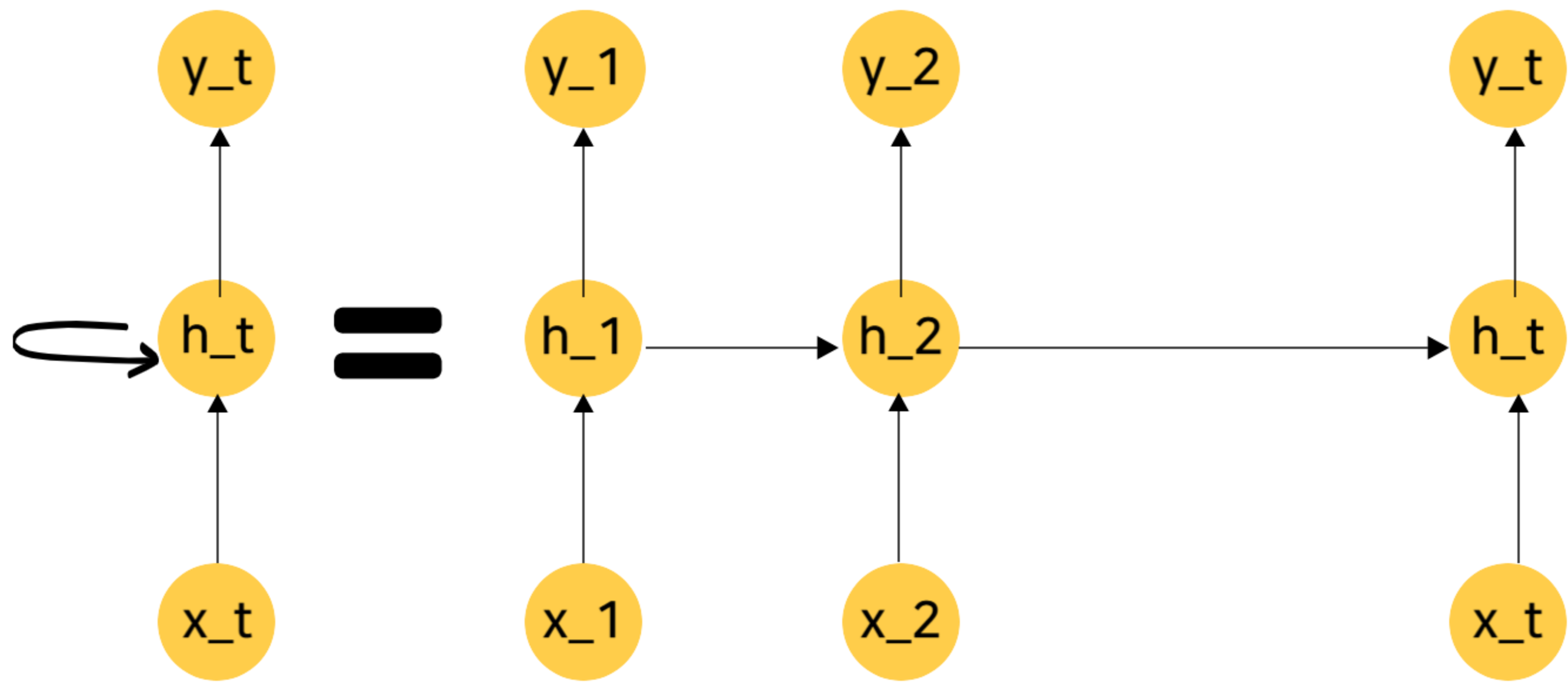
🔍 두산로보틱스 로키부트캠프 ▼

스터디 7조 신준식



순환 신경망(RNN)

RNN(Recurrent Neural Network)은 시간적으로 연속성이 있는 데이터를 처리하려고 고안된 인공 신경망입니다. RNN의 'Recurrent(반복되는)'는 이전 은닉층이 현재 은닉층의 입력이 되면서 '반복되는 순환 구조를 갖는다'는 의미입니다. RNN이 기존 네트워크와 다른 점은 '기억'을 갖는다는 것입니다. 이때 기억은 현재까지 입력데이터를 요약한 정보라고 생각하면 됩니다. 따라서 새로운 입력이 네트워크로 들어올 때마다 기억은 조금씩 수정되며, 결국 최종적으로 남겨진 기억은 모든 입력 전체를 요약한 정보가 됩니다.





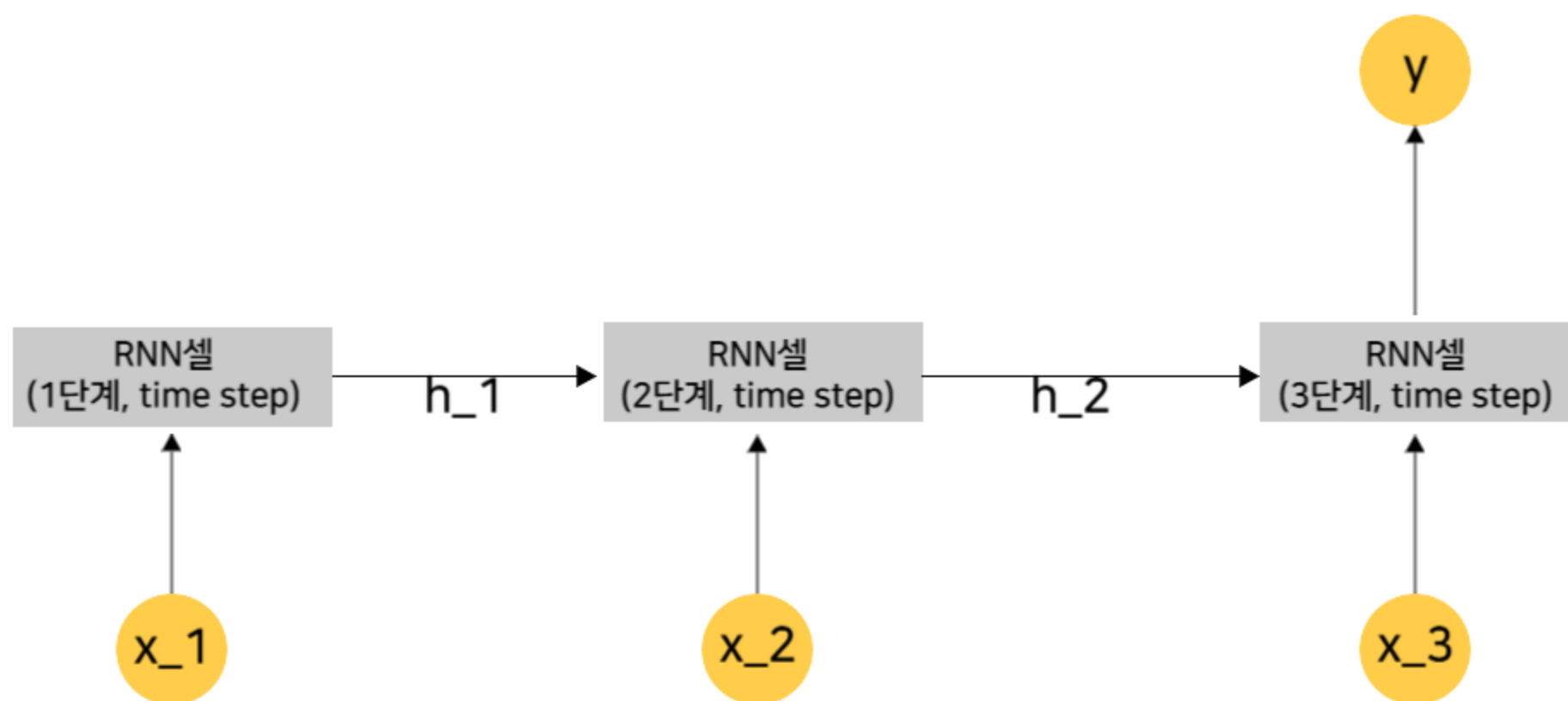
일대일 : 순환이 없기 때문에 RNN이라고 말하기 어려우며 , 순방향 네트워크가 대표적 사례

일대다 : 입력이 하나이고, 출력이 다수인 구조. 이미지를 입력해서 이미지에 대한 설명을 문장으로 출력하는 이미지 캡션이 대표적 사례

다대일 : 입력이 다수이고, 출력이 하나인 구조로, 문장을 입력해서 긍정/부정을 출력하는 감성 분석기에서 사용됨.

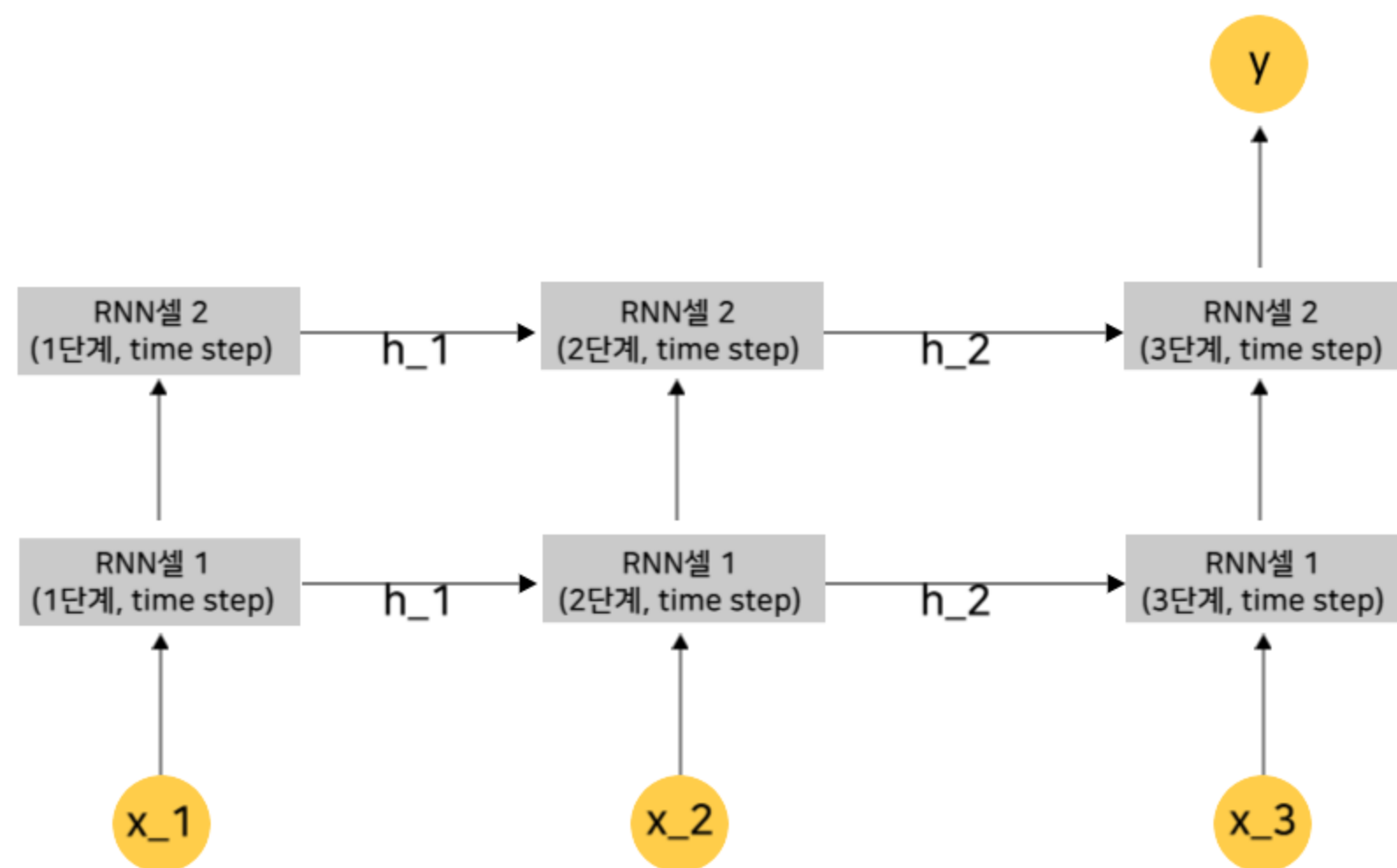
다대다 : 입력과 출력이 다수인 구조로, 언어를 번역하는 자동 번역기 등이 대표적 사례

동기화다대다 : 다대다의 유형처럼 입력과 출력이 다수인 구조입니다. 문장에서 다음에 나올 단어를 예측하는 언어모델, 즉 프레임 수준의 비디오 분류가 대표적 사례



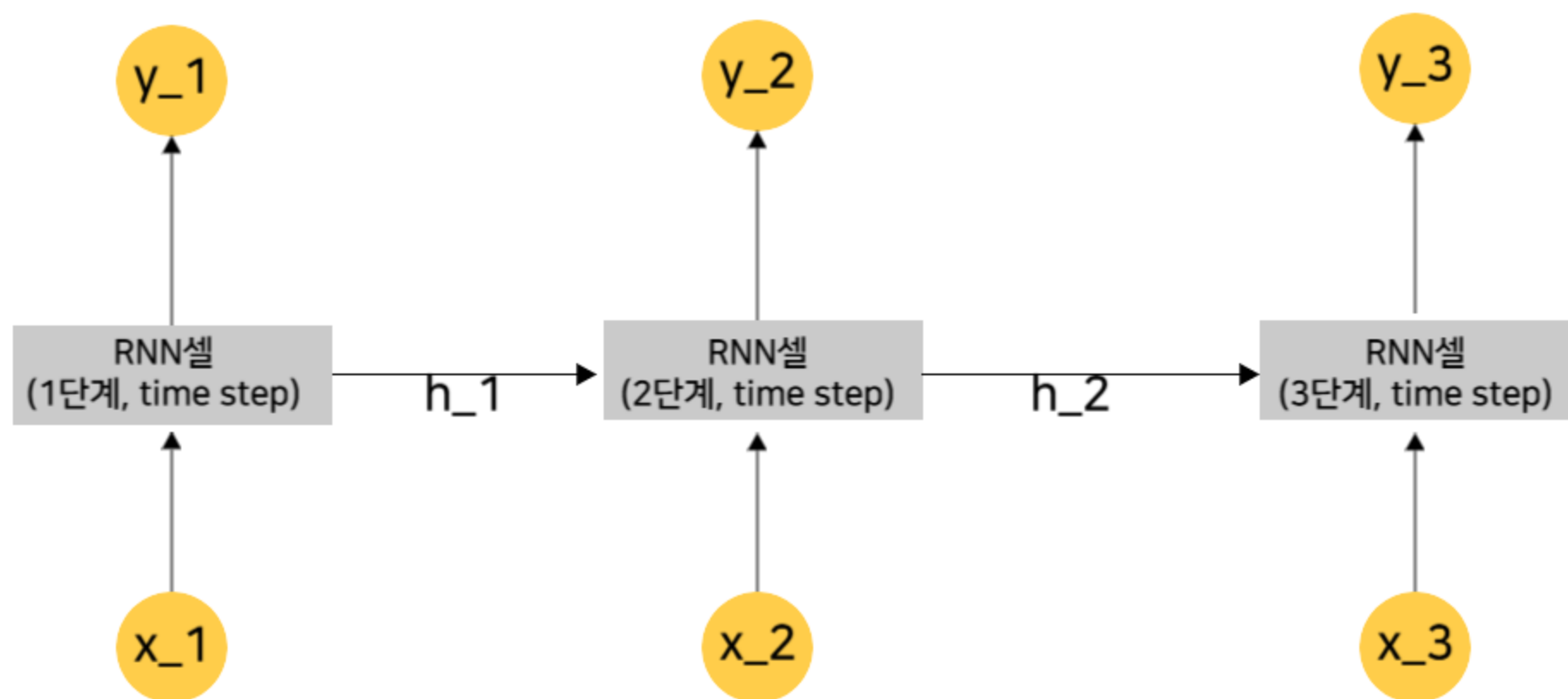
다대일 : 입력이 다수이고, 출력이 하나인 구조로, 문장을 입력해서 긍정/부정을 출력하는 감성 분석기에서 사용됨.

```
1 In_layer = tf.keras.input(shape=(3,1), name='input')
2 RNN_layer = tf.keras.layers.SimpleRNN(100, name='RNN')(In_layer)
3 Out_layer = tf.keras.layers.Dense(1, name='output')(RNN_layer)
```



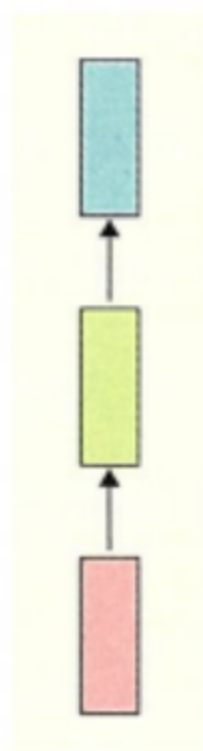
다대일 : 입력이 다수이고, 출력이 하나인 구조로, 문장을 입력해서 긍정/부정을 출력하는 감성 분석기에서 사용됨.

```
1 In_layer = tf.keras.Input(shape=(3,1), name='input')
2 RNN_layer0 = tf.keras.layers.SimpleRNN(100, name='RNN1')(In_layer)
3 RNN_layer1 = tf.keras.layers.SimpleRNN(100, name='RNN2')(RNN_layer0)
4 Out_layer = tf.keras.layers.Dense(1, name='output')(RNN_layer1)
```

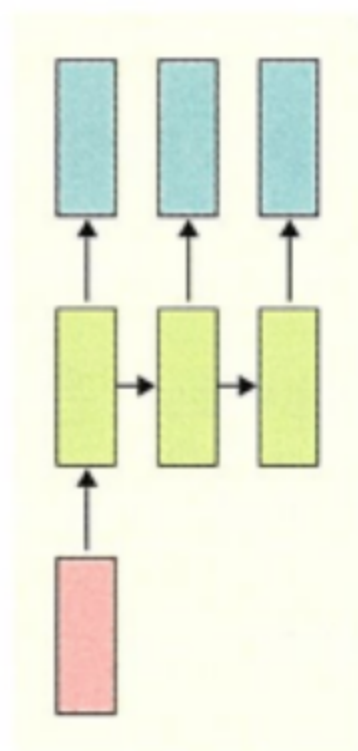


다대다 : 입력과 출력이 다수인 구조로, 언어를 번역하는 자동 번역기 등이 대표적 사례

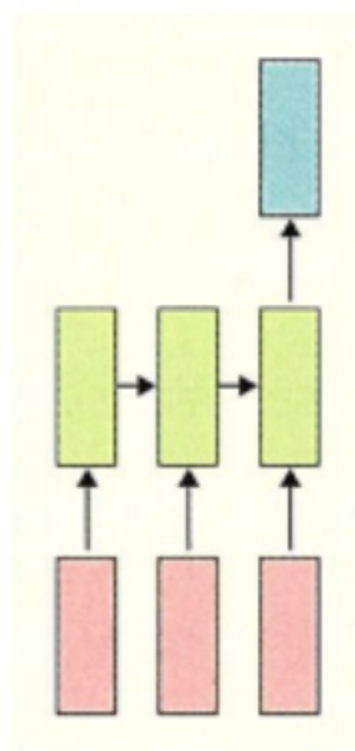
```
1 In_layer = tf.keras.input(shape=(3,1), name='input')
2 RNN_layer = tf.keras.layers.SimpleRNN(100, return_sequences=True, name='RNN')(In_layer)
3 Out_layer = tf.keras.layers.TimeDistributed(keras.layers.Dense(1), name='output')(RNN_layer)
```



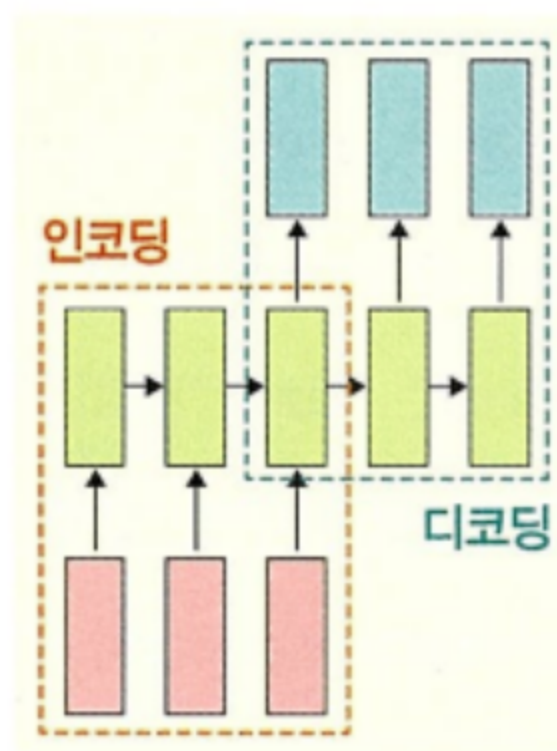
일대일



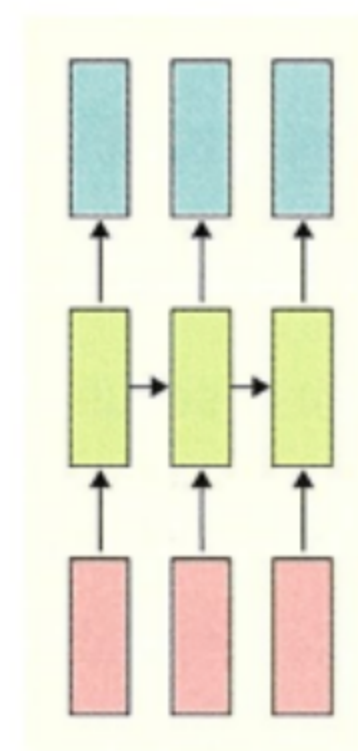
일대다



다대일



다대다



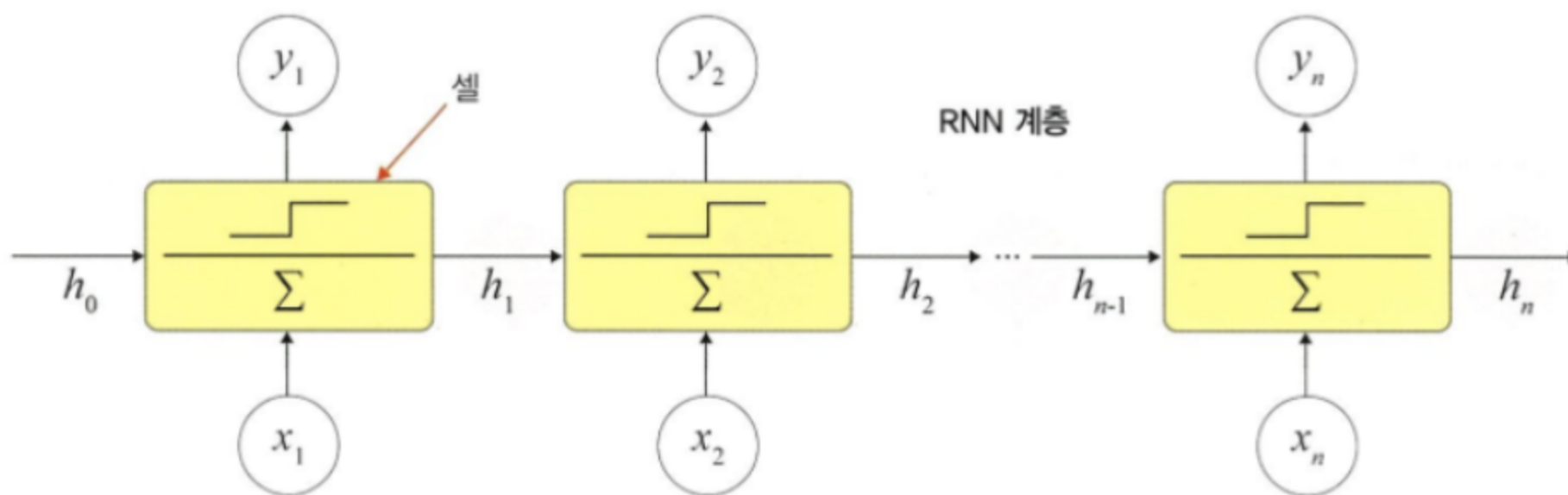
동기화 다대다

RNN 계층과 셀

이제 RNN을 구성하는 RNN 계층(layer)과 RNN 셀(cell)을 살펴보겠습니다.

RNN은 내장된 계층뿐만 아니라 셀 레벨의 API도 제공합니다. RNN 계층이 입력된 배치순서열을 모두 처리하는 것과 다르게 RNN셀은 오직 하나의 단계만 처리합니다. 따라서 RNN 셀은 RNN 계층의 for loop 구문의 내부라고 할 수 있습니다.

RNN 계층은 셀을 래핑하여 동일한 셀을 여러 단계에 적용합니다. 그림에서도 동일한 셀이 x_1, x_2, \dots, x_n 등 전체 RNN 네트워크(계층)에서 사용되고 있습니다. 따라서 셀은 입력 시퀀스에서 반복되고 return_sequence 같은 옵션을 기반으로 출력 값을 계산합니다. 즉, 셀은 실제 계산에 사용되는 RNN 계층의 구성 요소로, 단일 입력과 과거 상태(state)를 가져와서 출력과 새로운 상태를 생성합니다.

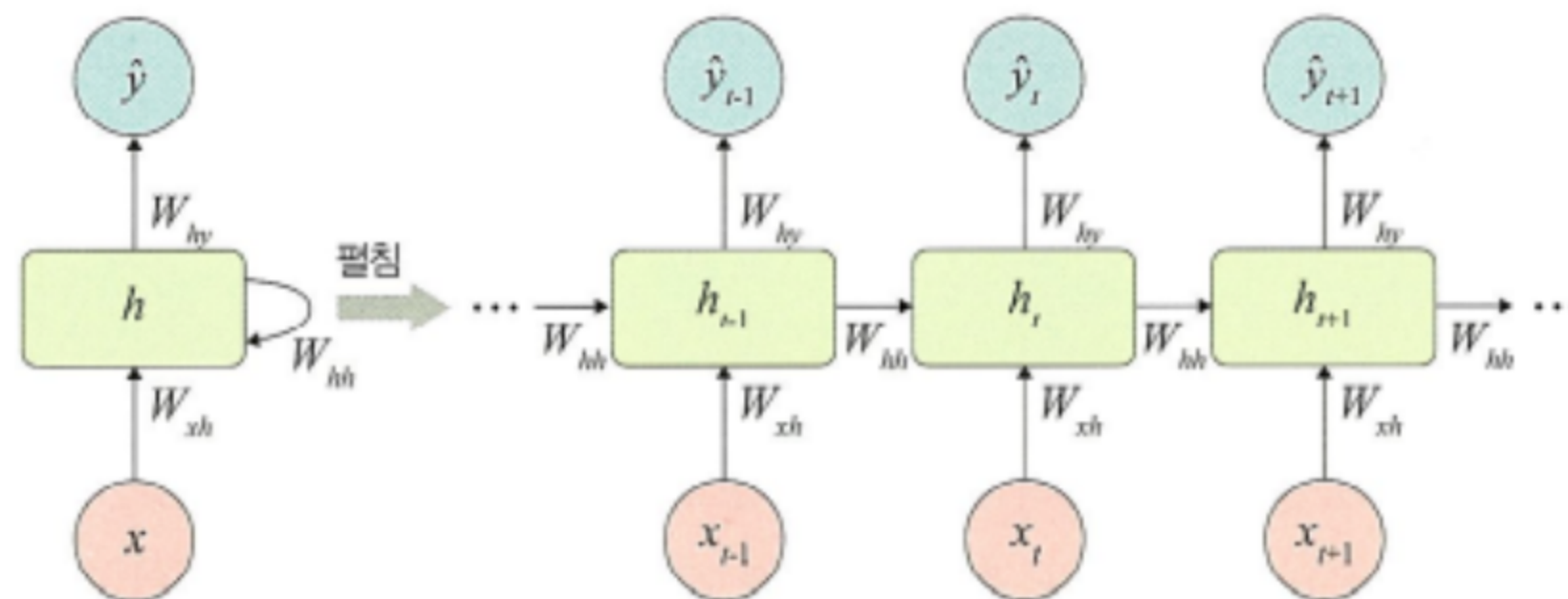


RNN 구조

RNN은 은닉층 노드들이 연결되어 이전 단계 정보를 은닉층 노드에 저장할 수 있도록 구성한 신경망입니다.↵

다음 그림에서 볼 수 있듯이 x_{t-1} 에서 h_{t-1} 을 얻고 다음 단계에서 h_{t-1} 과 x_t 를 사용하여 과거 정보와 현재 정보를 모두 반영합니다. 또한, h_t 와 x_{t+1} 의 정보를 이용하여 과거와 현재 정보를 반복해서 반영하는데, 이러한 구조를 요약한 것이 다음 그림입니다.↵

RNN에서는 입력층, 은닉층, 출력층 외에 가중치를 세 개 가집니다. RNN의 가중치는 W_{xh} , W_{hh} , W_{hy} 로 분류됩니다. W_{xh} 는 입력층에서 은닉층으로 전달되는 가중치이고, W_{hh} 는 t 시점의 은닉층에서 $t+1$ 시점의 은닉층으로 전달되는 가중치입니다. 또한, W_{hy} 는 은닉층에서 출력층으로 전달되는 가중치입니다. 가중치 W_{xh} , W_{hh} , W_{hy} 는 모든 시점에 동일하다는 것에 주의할 필요가 있습니다. 즉, 가중치를 공유하는데 그림과 같이 모든 가중치가 동일한 것을 확인할 수 있습니다. ↵



<t 단계에서의 RNN 계산>

1. 은닉층

계산을 위해 x_t 와 h_{t-1} 이 필요합니다. 즉, (이전 은닉층 \times 은닉층 \rightarrow 은닉층 가중치 + 입력층 \rightarrow 은닉층 가중치 \times (현재) 입력 값)으로 계산할 수 있으며, RNN에서 은닉층은 일반적으로 **하이퍼볼릭 탄젠트 활성화 함수**를 사용

$$h_t = \tanh(\hat{y}_t)$$

$$\hat{y}_t = W_{hh} \times h_{t-1} + W_{xh} \times x_t$$

2. 출력층

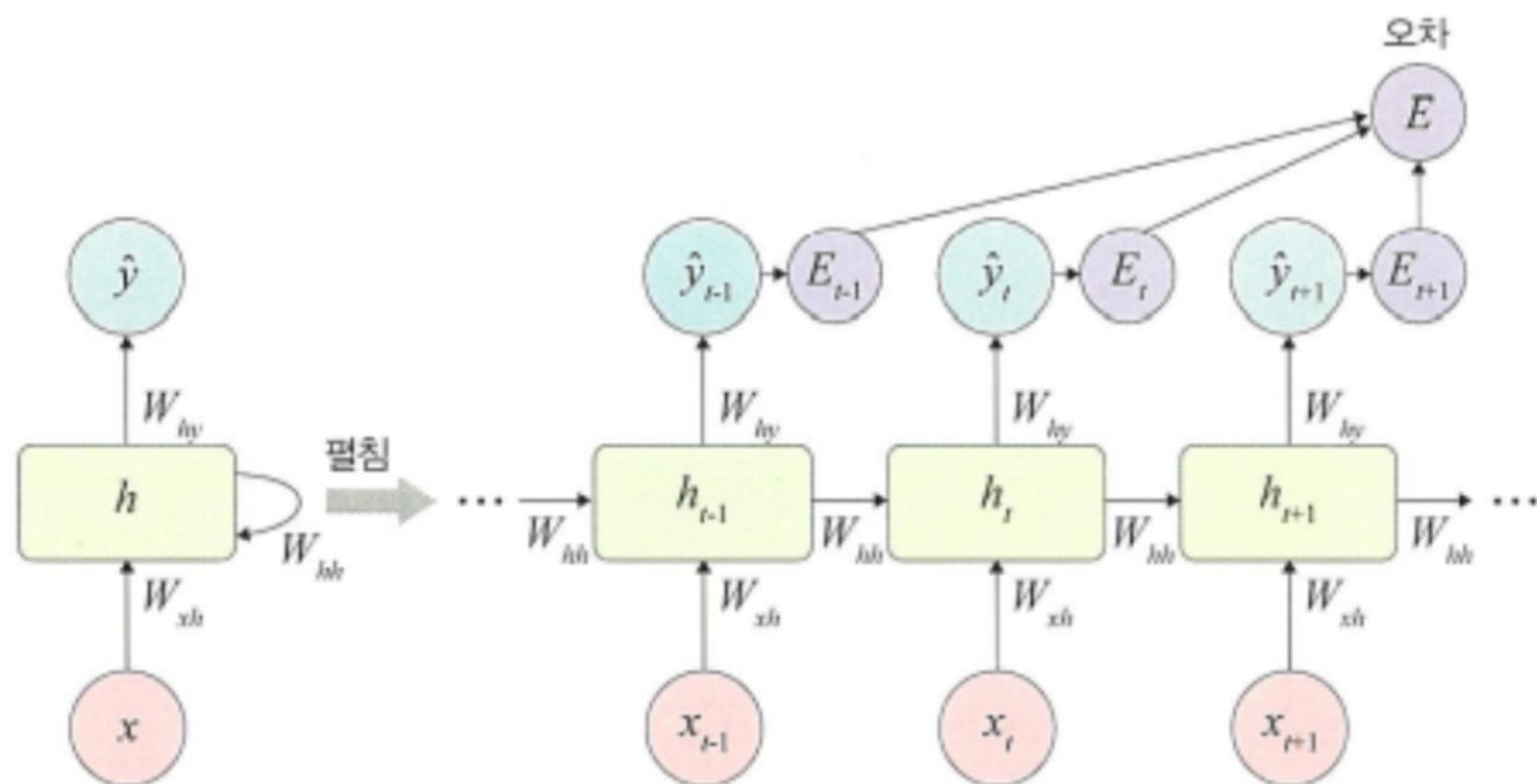
심층 신경망과 계산 방법이 동일합니다. 즉, (은닉층 \rightarrow 출력층 가중치 \times 현재 은닉층)에 **소프트맥스 함수**를 적용

$$\hat{y}_t = \text{softmax}(W_{hy} \times h_t)$$

3. RNN의 오차(E)

심층 신경망에서 전방향(feedforward) 학습과 달리 각 단계(t)마다 오차를 측정합니다.

즉, 각 단계마다 실제 값(y_t)과 예측 값(\hat{y}_t)으로 오차(**평균 제곱 오차(mean square error) 적용**)를 이용하여 측정



생략된-BPTT:
계산량을 줄이기 위해 현재 단계에서 일정 시점까지만(보통 5단계 이전까지만) 오류를 역전파한다.

4.RNN에서 역전파

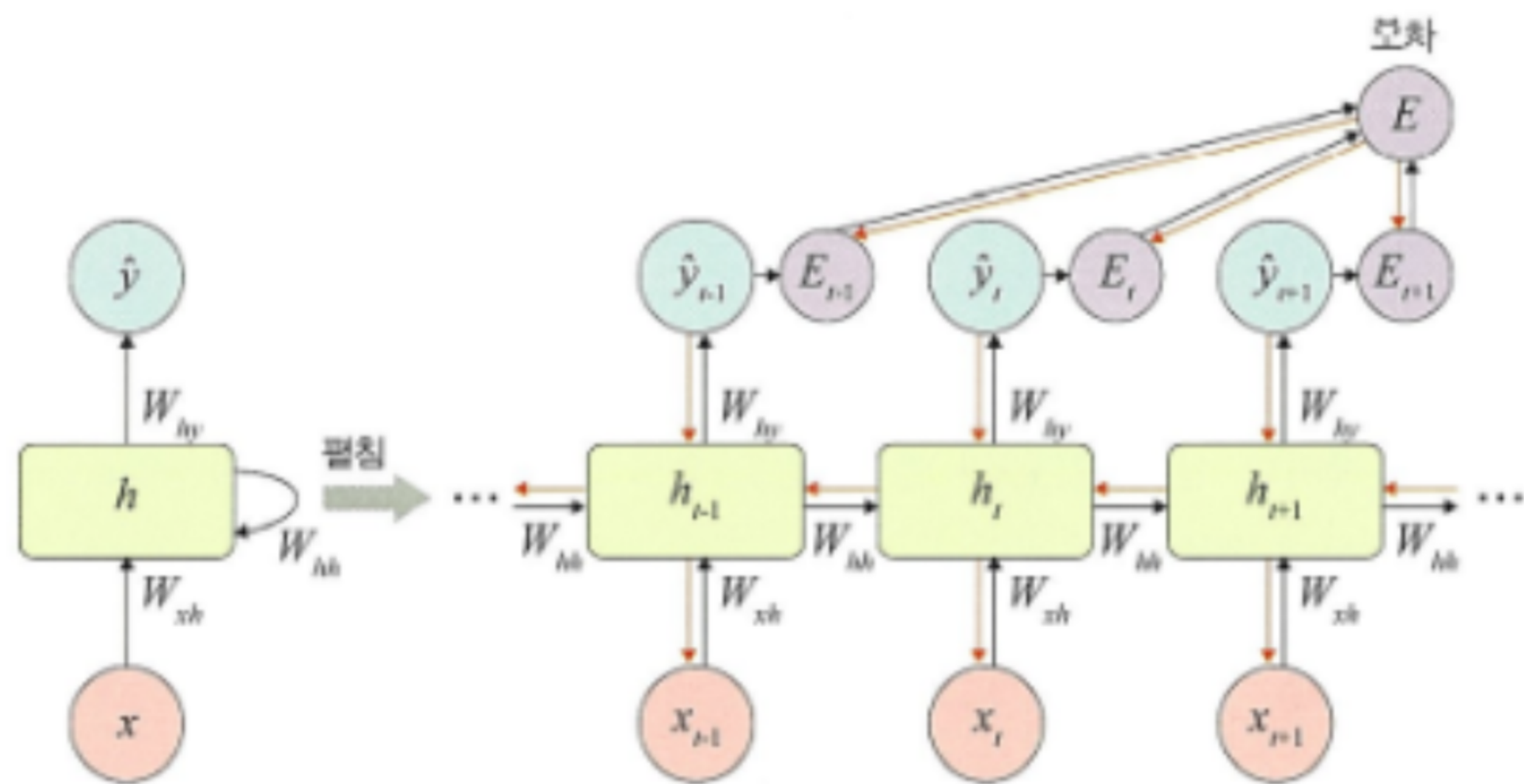
BPTT(BackPropagation Through Time)를 이용하여 모든 단계마다 처음부터 끝까지 역전파

오차는 각 단계(t)마다 오차를 측정하고 이전 단계로 전달되는데, 이것을 BPTT라고 합니다.

즉, 3에서 구한 오차를 이용하여 w_{xh} , w_{hh} , w_{hy} 및 바이어스(bias)를 업데이트 합니다.

이때 BPTT는 오차가 멀리 전파될 때(왼쪽으로 전파) 계산량이 많아지고 전파되는 양이 점차 적어지는 문제점(기울기 소멸 문제(vanishing gradient problem))이 발생합니다.

기울기 소멸 문제를 보완하기 위해 오차를 몇 단계까지만 전파시키는 생략된-BPTT(truncated BPTT)를 사용할 수도 있고, 근본적으로는 LSTM 및 GRU를 많이 사용



RNN의 역방향 학습

LSTM

RNN은 치명적 단점이 있습니다. 앞서 언급했듯이 가중치가 업데이트되는 과정에서 1보다 작은 값이 계속 곱해지기 때문에 기울기가 사라지는 기울기 소멸 문제가 발생합니다. 이를 해결하기 위해 LSTM이나 GRU 같은 확장된 RNN 방식들을 사용합니다.

LSTM 구조는 순전파와 역전파 과정을 살펴보겠습니다.

LSTM 순전파

LSTM은 기울기 소멸 문제를 해결하기 위해 망각 게이트, 입력 게이트, 출력 게이트라는 새로운 요소를 은닉층의 각 뉴런에 추가했습니다.

망각 게이트

망각 게이트는 과거 정보를 어느 정도 기억할지 결정합니다. 과거 정보와 현재 데이터를 입력받아 시그모이드를 취한 후 그 값을 과거 정보에 곱해 줍니다. 따라서 시그모이드의 출력이 0이면 과거 정보는 버리고, 1이면 과거 정보는 온전히 보존합니다.

0과 1사이의 출력 값을 가지는 h_{t-1} 과 x_t 를 입력 값으로 받습니다. 이때 x_t 는 새로운 입력 값이고 h_{t-1} 은 이전 은닉층에서 입력되는 값입니다. 즉, h_{t-1} 과 x_t 를 이용하여 이전 상태 정보를 현재 메모리에 반영할지 결정하는 역할을 합니다.

- 계산한 값이 1이면 바로 직전의 정보를 메모리에 유지
- 계산한 값이 0이면 초기화

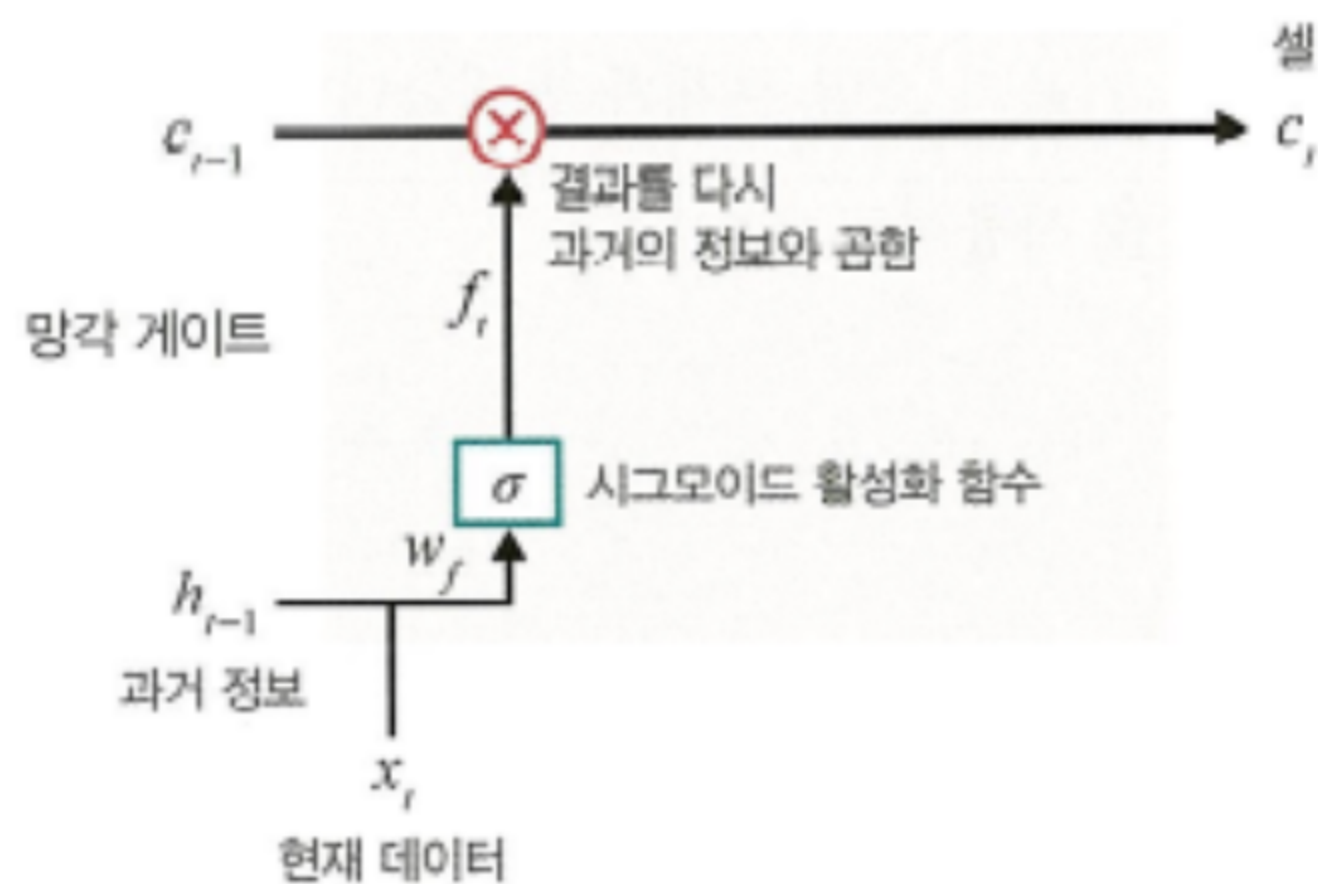
망각 게이트

$$f_t = \sigma(w_f[h_{t-1}, x_t])$$

$$c_t = f_t \cdot c_{t-1}$$

그림으로는 다음과 같이 표현할 수 있습니다.

▼ 그림 7-16 망각 게이트



입력 게이트

입력 게이트는 현재 정보를 기억하기 위해 만들어졌습니다. 과거 정보와 현재 데이터를 입력받아 시그모이드와 하이퍼볼릭 탄젠트 함수를 기반으로 현재 정보에 대한 보존량을 결정합니다.

즉, 현재 메모리에 새로운 정보를 반영할지 결정하는 역할을 합니다.

- 계산한 값이 1이면 x_t 가 들어올 수 있도록 허용 (open)
- 계산한 값이 0이면 차단

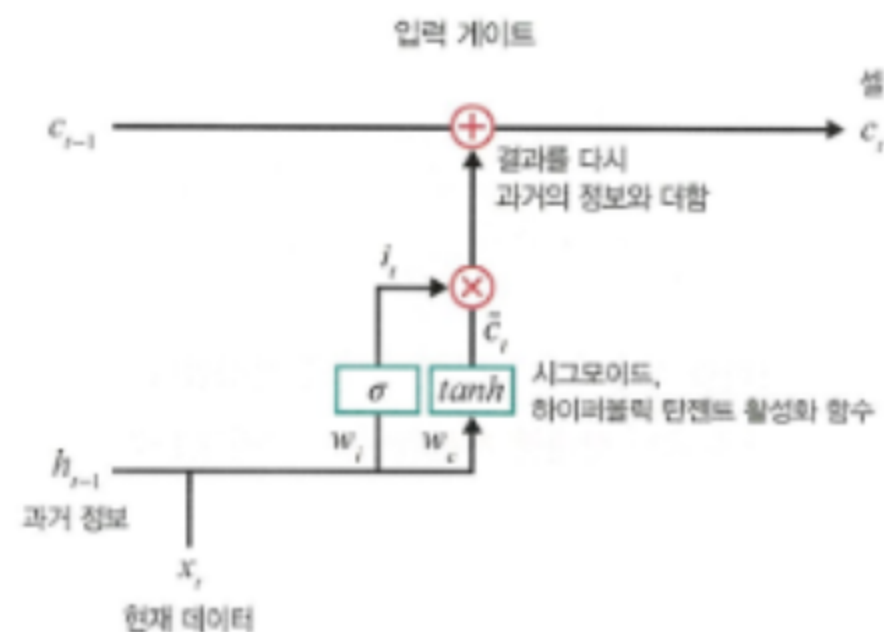
$$i_t = \sigma(w_i[h_{t-1}, x_t])$$

$$\tilde{c}_t = \tanh(w_c[h_{t-1}, x_t])$$

$$c_t = c_{t-1} + i_t \cdot \tilde{c}_t$$

그림으로는 다음과 같이 표현할 수 있습니다.

▼ 그림 7-17 입력 게이트



셀

각 단계에 대한 은닉 노드(hidden node)를 메모리 셀이라고 합니다. '총합'을 사용하여 셀 값을 반영하여, 이것으로 기울기 소멸 문제가 해결됩니다.

셀을 업데이트 하는 방법은 다음과 같습니다.

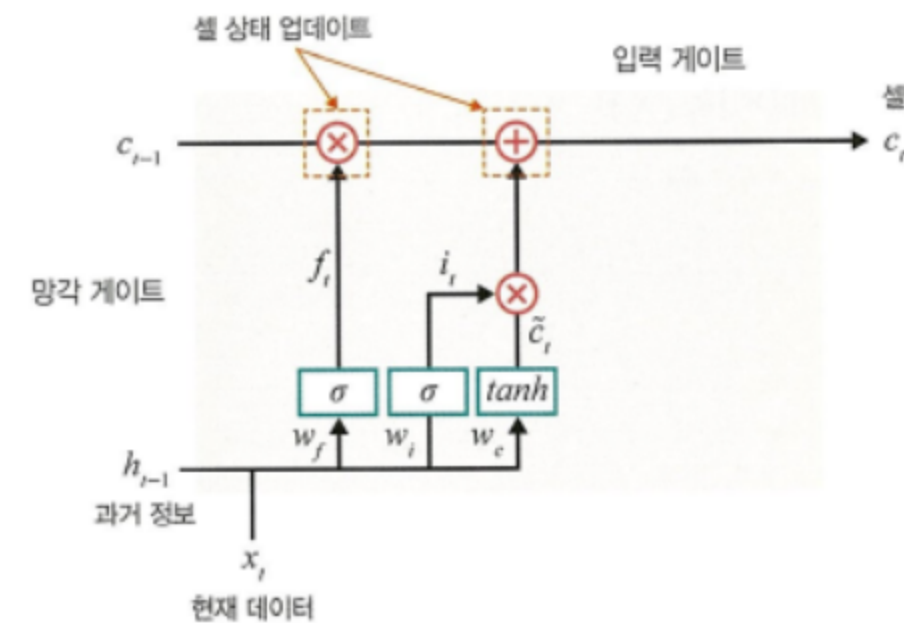
망각 게이트와 입력게이트의 이전 단계 셀 정보를 계산하여 현재 단계의 셀 상태(cell state)를 업데이트 합니다.

$$f_t = \sigma(w_f[h_{t-1}, x_t])$$

$$c_t = c_{t-1} + i_t \cdot \tilde{c}_t$$

그림으로는 다음과 같이 표현할 수 있습니다.

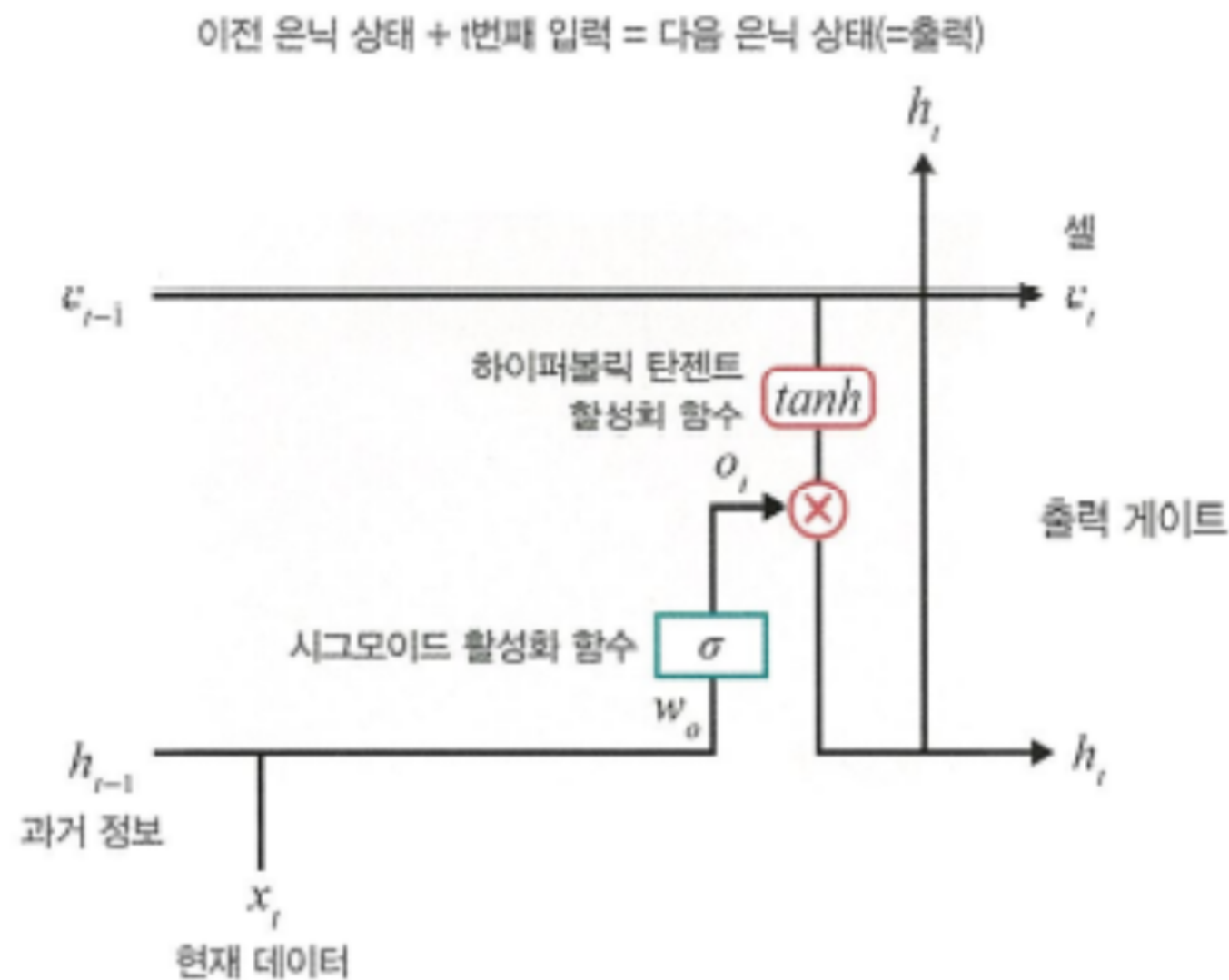
▼ 그림 7-18 셀

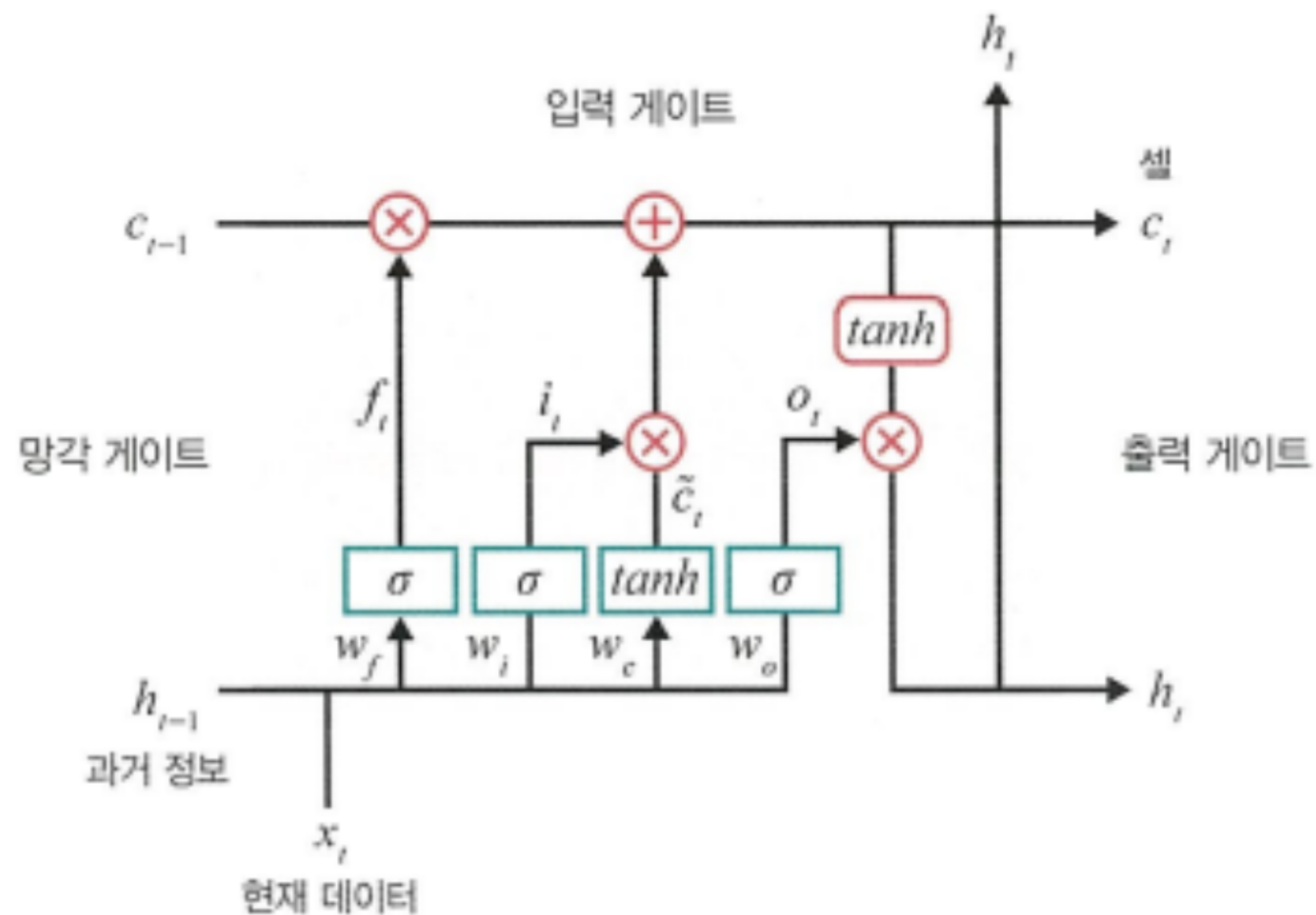


출력 게이트

출력 게이트는 과거 정보와 현재 데이터를 사용하여 뉴런의 출력을 결정합니다. 이전 은닉 상태와 t 번째 입력을 고려해서 다음 은닉 상태를 계산합니다. 그리고 LSTM에서는 이 은닉 상태가 그 시점에서의 출력이 됩니다. 출력 게이트는 생성된 메모리의 출력 값을 제어하는 역할을 합니다.

- 계산한 값이 1이면 의미 있는 결과로 최종 출력
- 계산한 값이 0이면 해당 연산 출력을 하지 않음



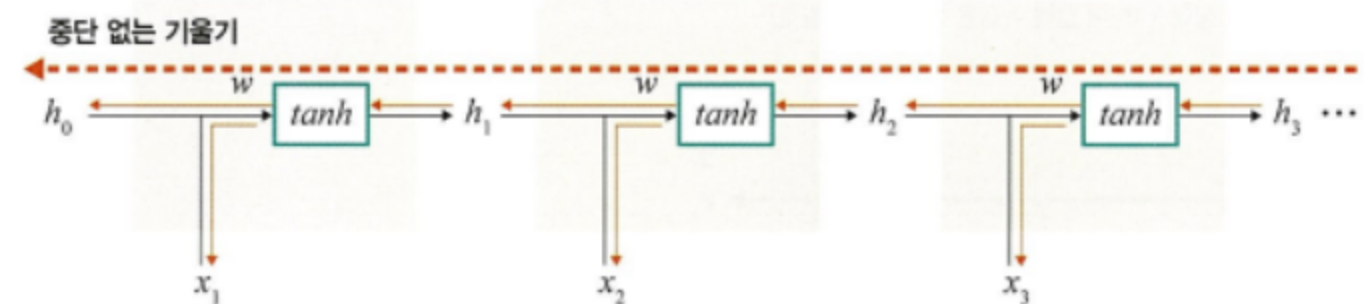


LSTM 역전파

- LSTM 역전파

LSTM은 셀을 통해서 역전파를 수행 하기 때문에 '중단 없는 기울기(uninterrupted gradient flow)'라고도 함

▼ 그림 7-21 LSTM 셀 단위 역전파



다음은 역전파를 수행하기 위한 공식입니다.

$$\begin{aligned}
 t_t &= \tanh(w_{hh}h_{t-1} + w_{xh}x_t) \\
 &= \tanh\left((w_{hh} \quad w_{xh}) \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \\
 &= \tanh\left(w \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)
 \end{aligned}$$

즉, 다음 그림과 같이 최종 오차는 모든 노드에 전파되는데, 이때 셀을 통해서 중단 없이 전파

▼ 그림 7-22 입력층으로의 역전파

