

15강

인공지능

# 딥러닝(2)

컴퓨터과학과 이병래교수

# 학습목차

1 심층합성곱 신경망

2 순환 신경망

3 트랜스포머

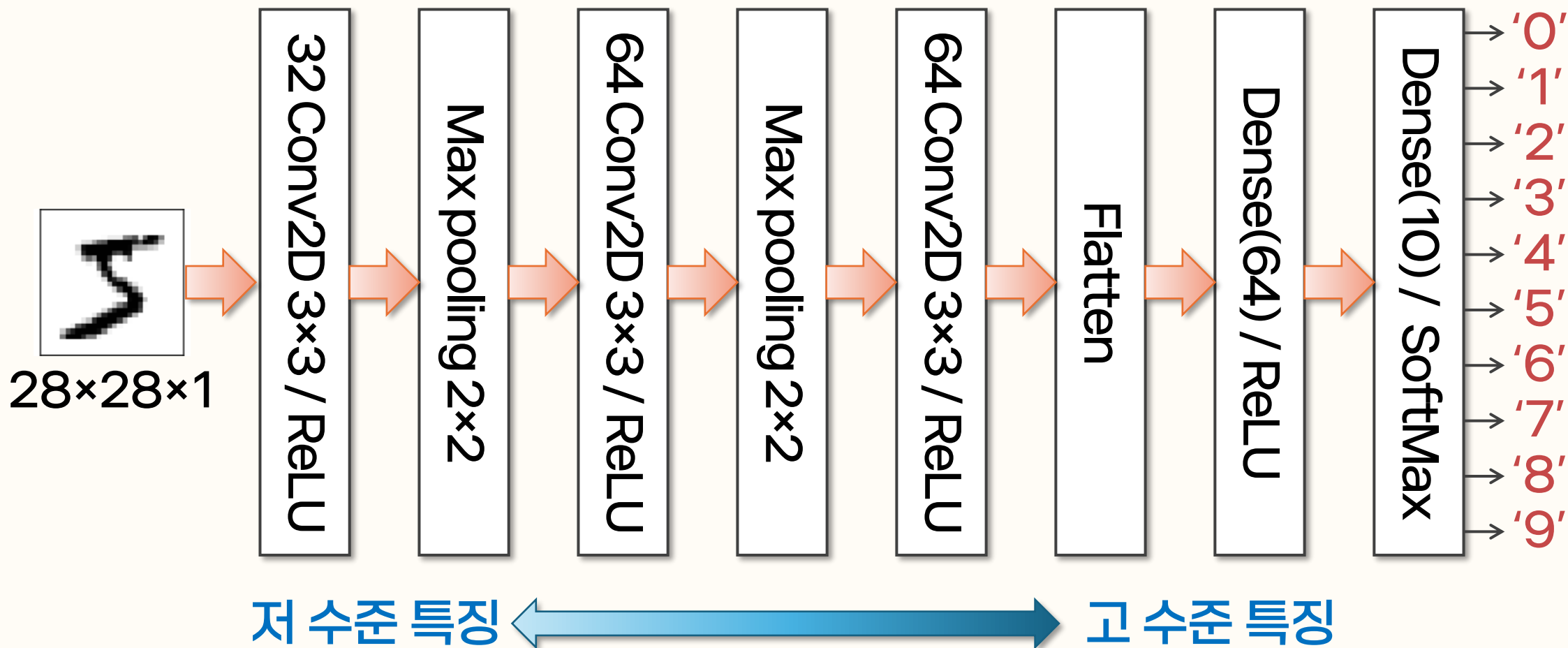




# 심층 합성곱 신경망

# 1. 심층 신경망의 층 수와 성능

## ■ 신경망의 층과 특징 검출



# 1. 심층 신경망의 층 수와 성능

## ■ 신경망의 층과 특징 검출

- 저층: 낮은 수준의 특징 추출
- 고층: 높은 수준의 복합적인 특징 추출



층 수가 많을수록 특징의 수준을 풍부하게 함으로써  
높은 표현력을 가질 수 있음

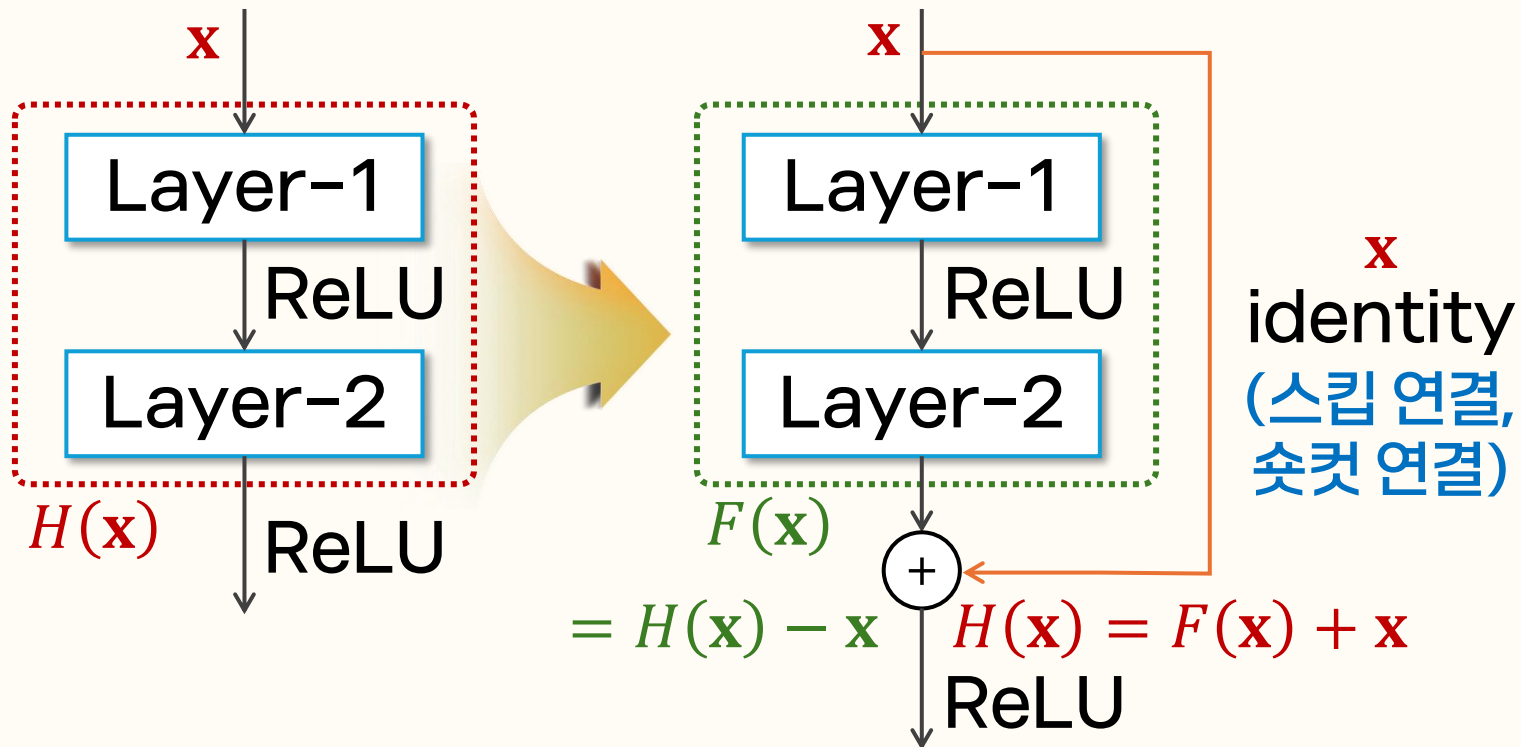


## 층을 더 쌓는 것 만큼 잘 학습시키는 것 역시 쉬운가?

- ➡ 층이 매우 많아지면 오히려 학습 품질이 떨어지는 문제가 발생할 수 있음
- ➡ Kaiming He: 잔차 블록(residual block)으로 구성되는 ResNet(Residual Network) 모델을 제안함

## 2. ResNet의 개념

### 잔차 블록(residual block)

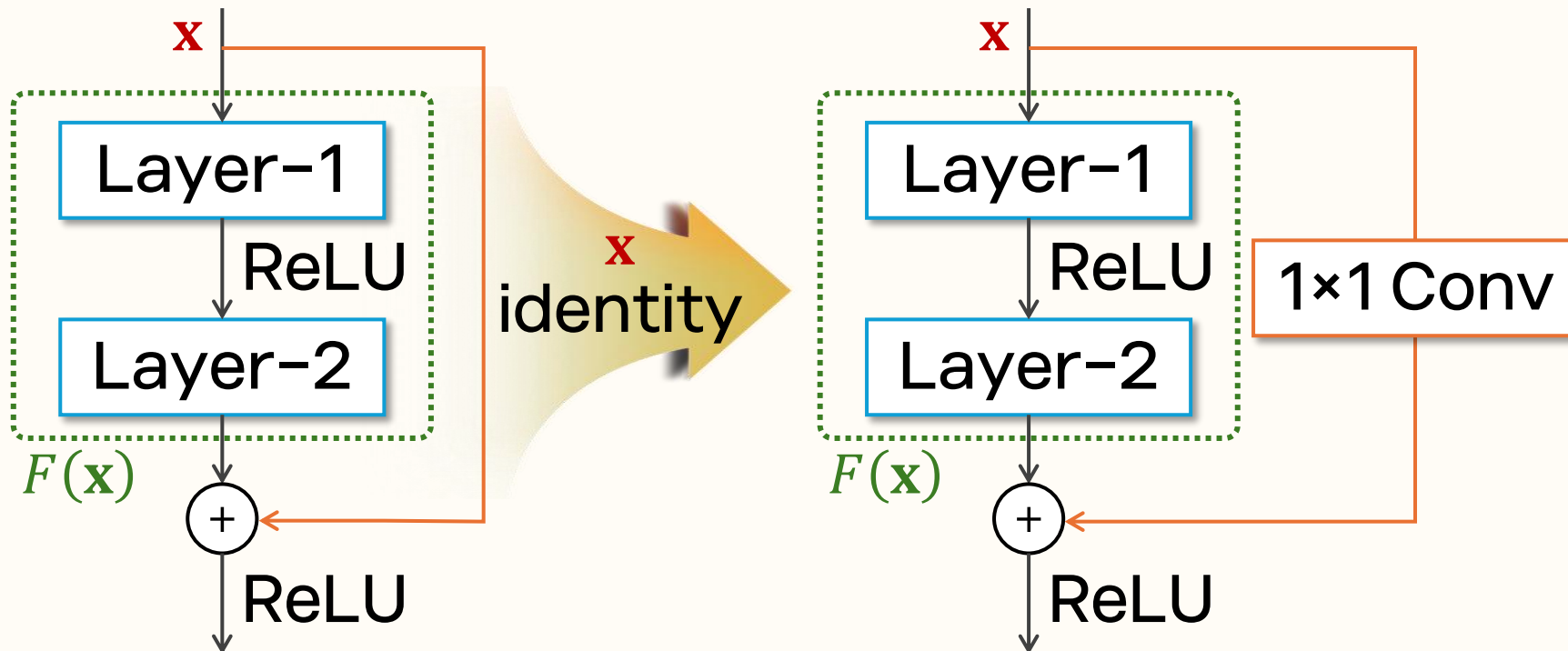


## 2. ResNet의 개념

### 잔차 블록(residual block)

- Layer-2의 출력( $F(x)$ )과  $x$ 의 규격이 같아야 함

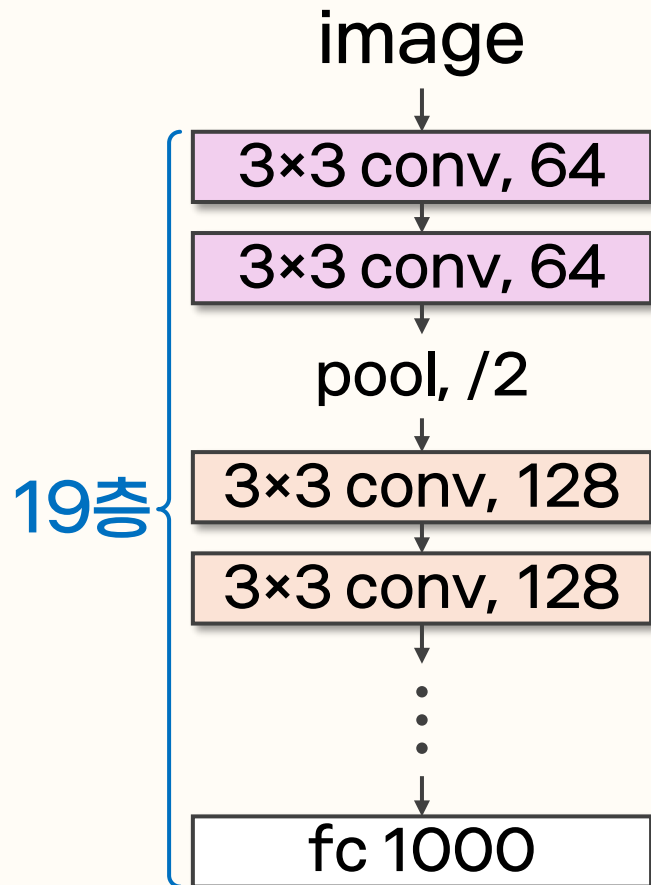
→ 채널 수가 다르다면  $1 \times 1$  합성곱을 사용하여 맞출 수 있음



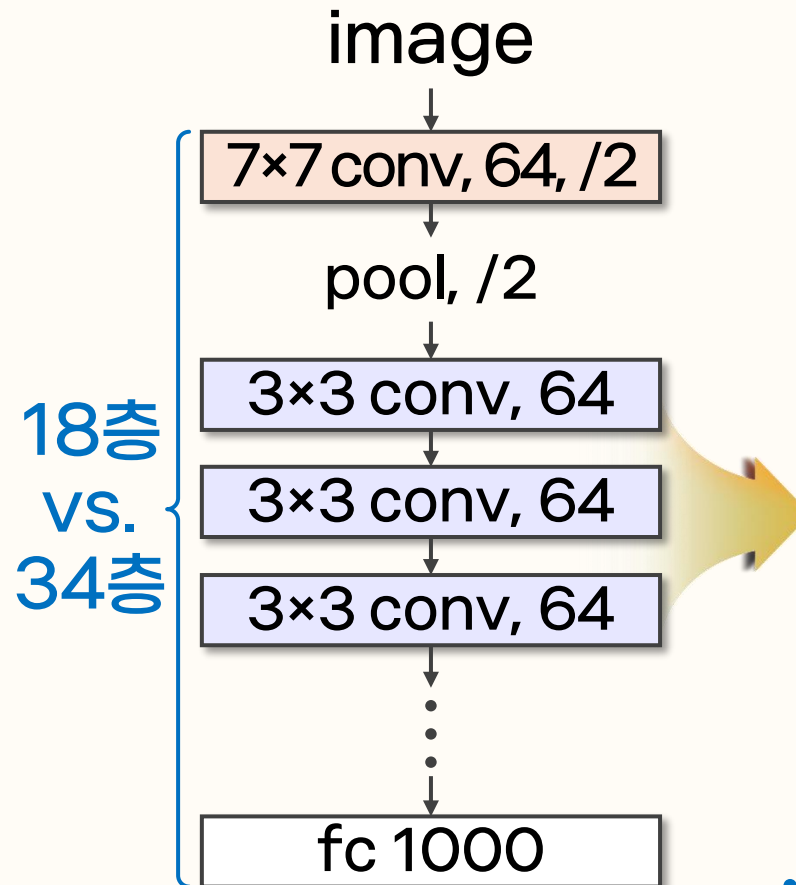
### 3. 심층 ResNet을 이용한 영상 분류


#### ❑ VGG model

##### VGG-19



##### 34-layer plain



 **Top-1 error**  
(ImageNet Classification)

18층: 27.94%

34층: 28.54%

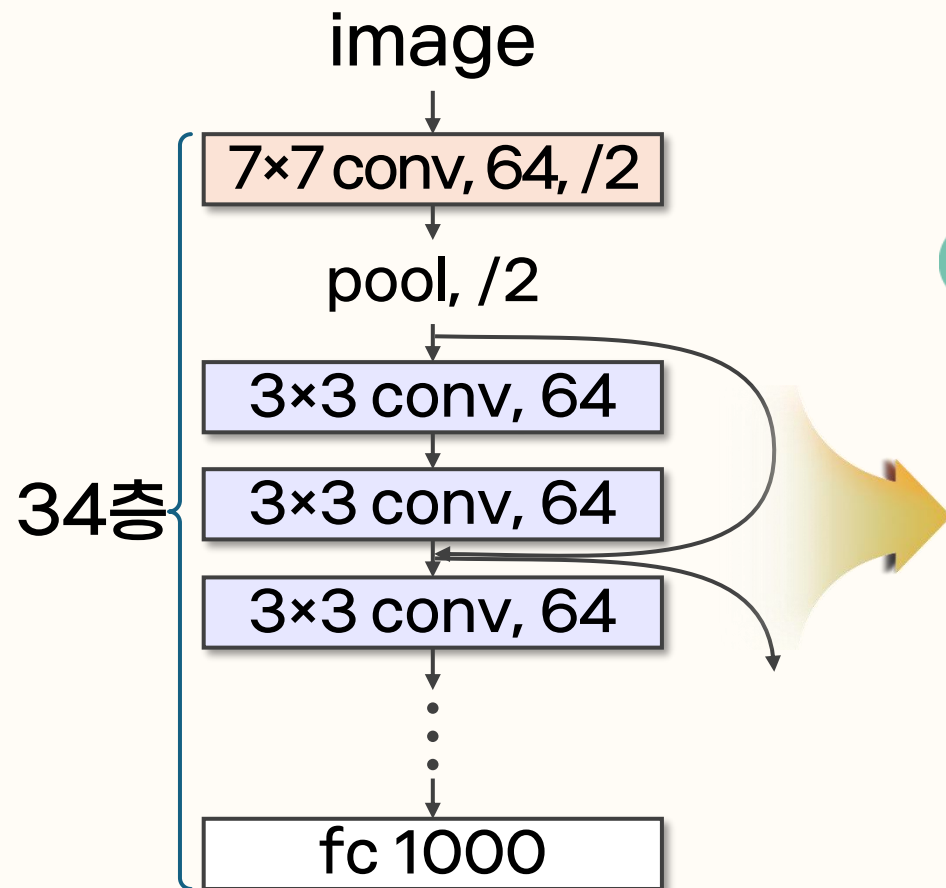
• Kaiming He, et al(2016)



### 3. 심층 ResNet을 이용한 영상 분류

#### ■ ResNet의 구성

#### 34-layer residual



#### Top-1 error (ImageNet Classification)

18층: 27.88%

34층: 25.03%

50층: 22.85%

152층: 21.43%

• Kaiming He, et al(2016)



## 순환 신경망

# 1. 순환 신경망의 개념

## ■ 순환 신경망(Recurrent Neural Network: RNN)이란?

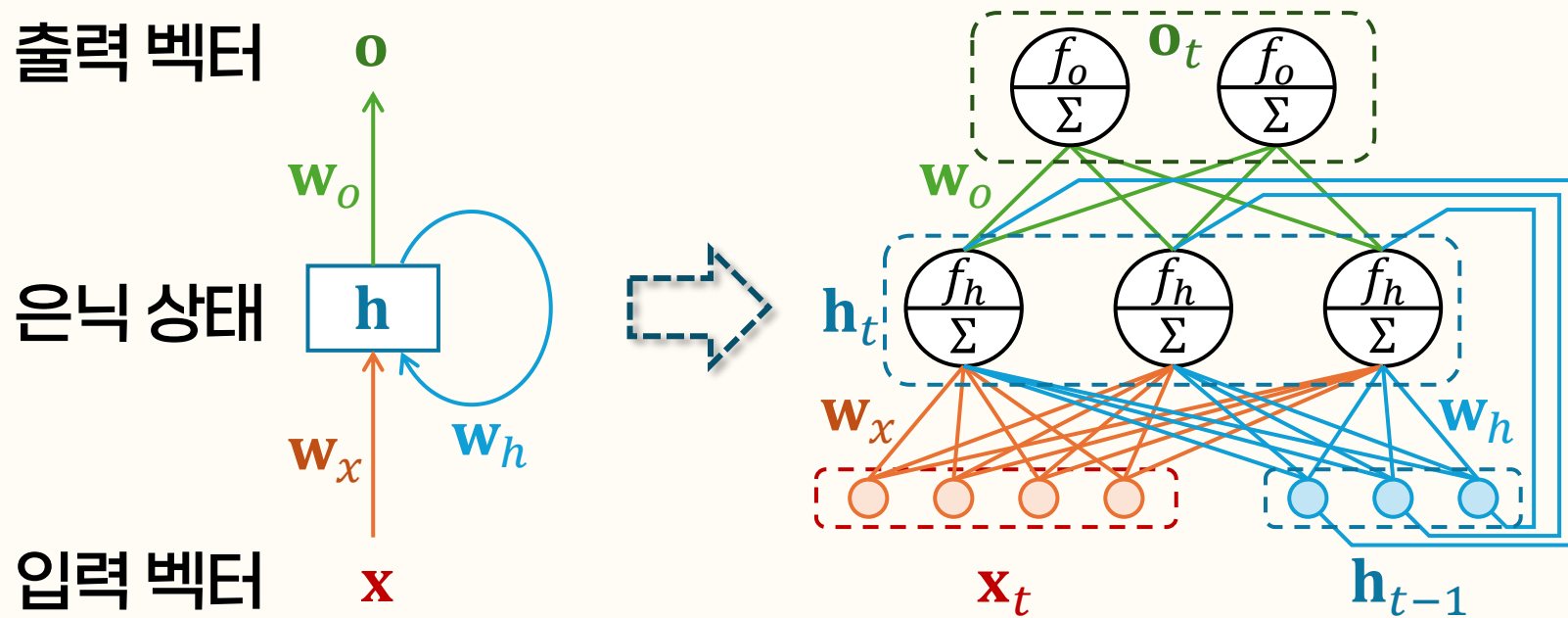
### 목적

- 연속적으로 발생하는 데이터(sequence, 시퀀스)의 처리
  - 시계열 데이터: 주가, 날씨 등 시간에 따라 변화하는 데이터
  - 자연언어의 문장: 순서에 따라 나열된 단어
- 과거 시퀀스의 패턴을 학습한 지식을 바탕으로 미래의 데이터를 예측함

### 구조

- 사이클을 형성하는 순환연결을 포함함
  - ➡ 이전 입력에 대한 '기억'을 유지할 수 있음

## 2. RNN의 구조

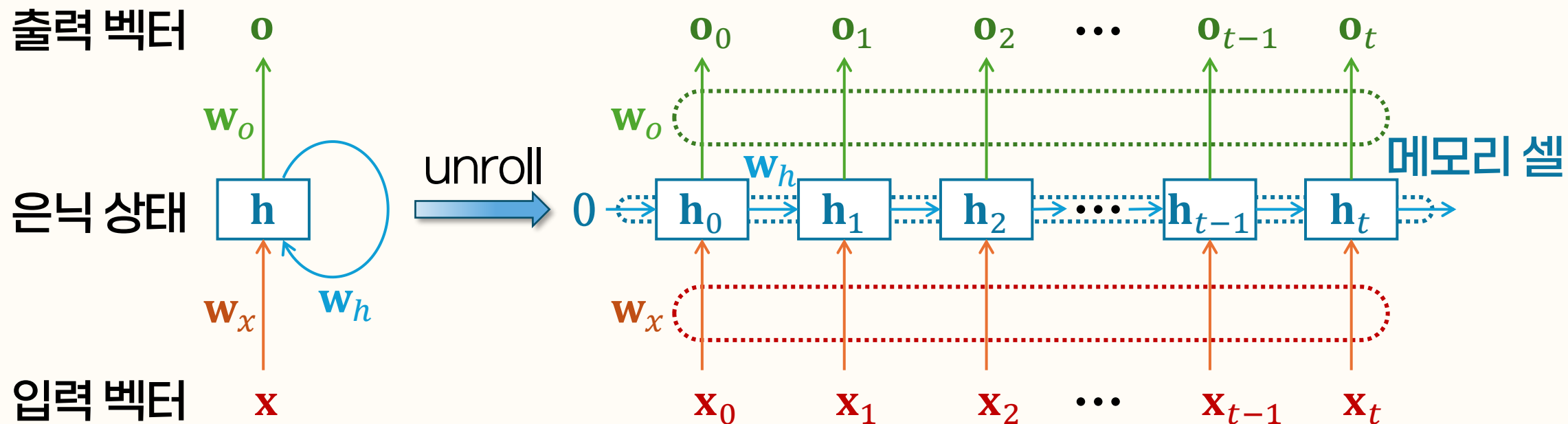


$$\mathbf{h}_t = f_h(\mathbf{w}_x \mathbf{x}_t + \mathbf{w}_h \mathbf{h}_{t-1} + \mathbf{b}_h)$$

$$\mathbf{o}_t = f_o(\mathbf{w}_o \mathbf{h}_t + \mathbf{b}_o)$$

여기서  $\mathbf{b}$ : 바이어스,  $f$ : 활성화함수

## 2. RNN의 구조



$$\mathbf{h}_t = f_h(\mathbf{w}_x \mathbf{x}_t + \mathbf{w}_h \mathbf{h}_{t-1} + \mathbf{b}_h)$$

$$\mathbf{o}_t = f_o(\mathbf{w}_o \mathbf{h}_t + \mathbf{b}_o)$$

여기서  $\mathbf{b}$ : 바이어스,  $f$ : 활성화함수

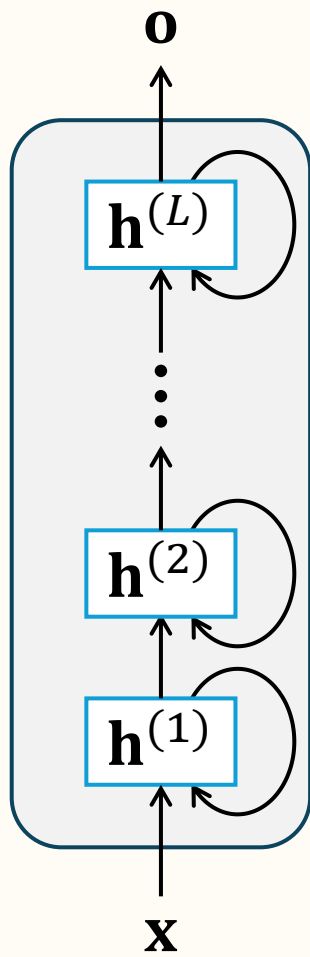
## 2. RNN의 구조

### ■ 심층 RNN

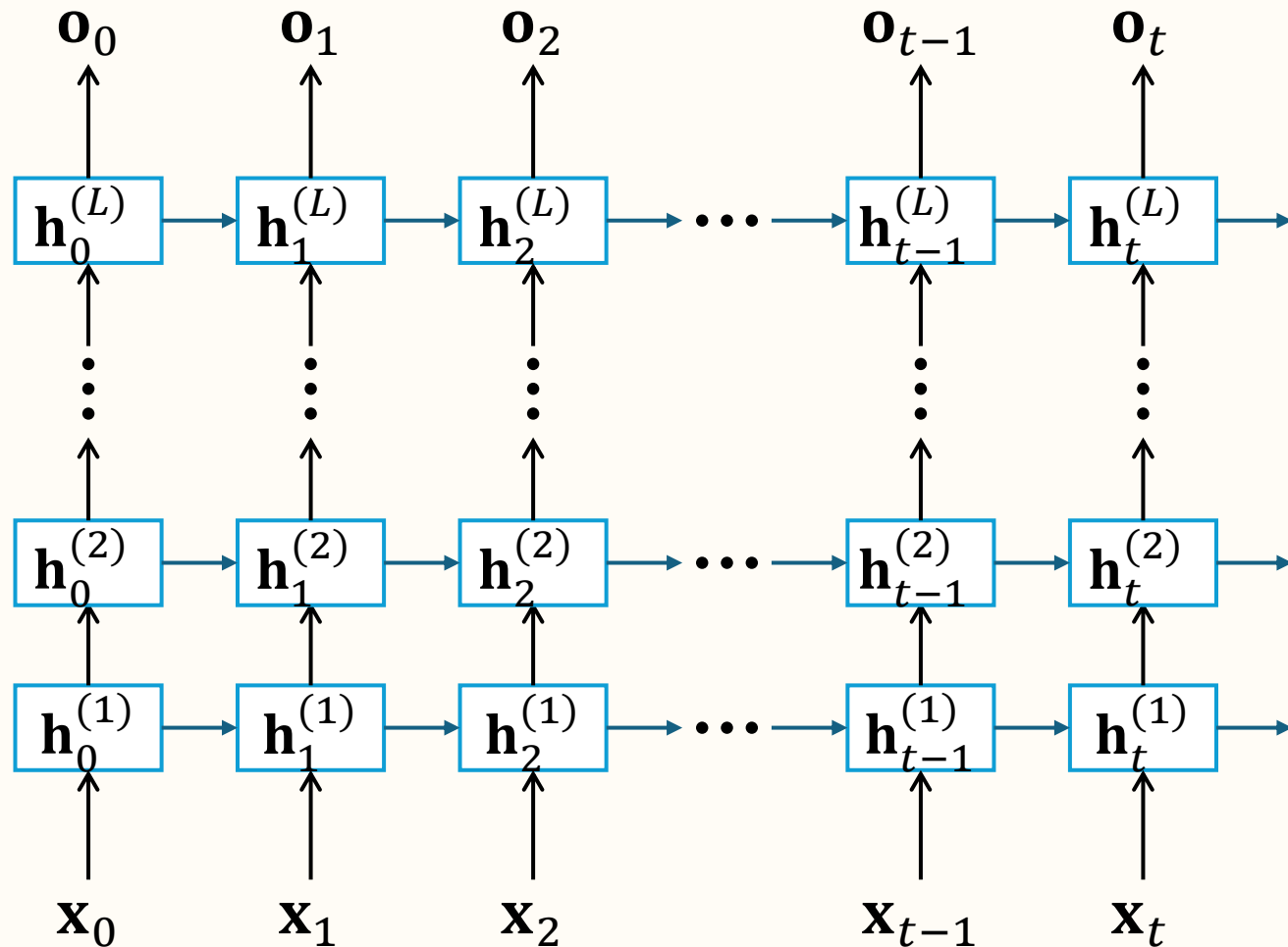
출력 벡터

은닉 상태

입력 벡터



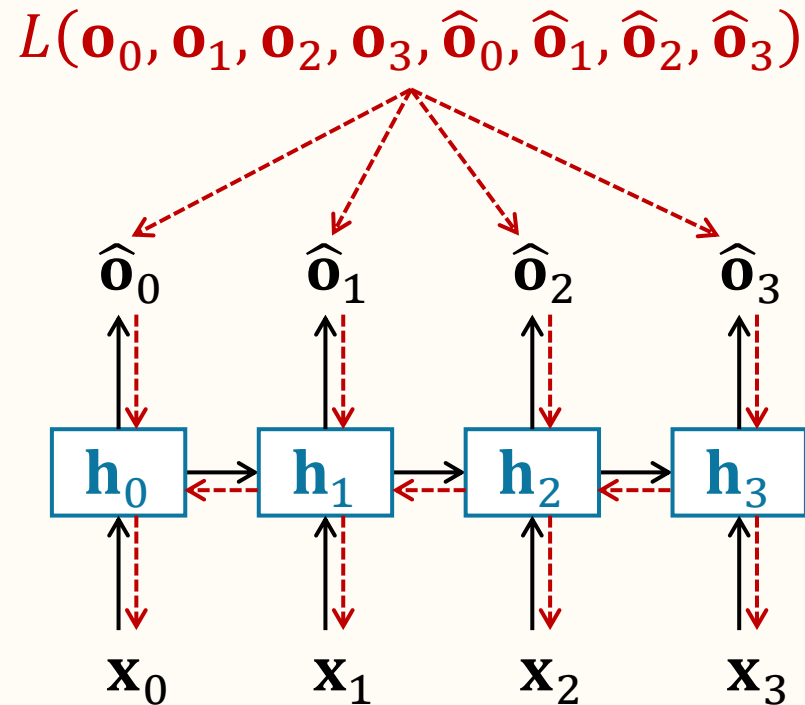
unroll



### 3. RNN의 훈련

#### ■ BPTT(BackPropagation Through Time)

- 손실함수를 계산한 후, 역전파 방식으로  $w$ 와  $b$ 를 업데이트함
- 시간 단계의 역순으로 경사가 전달됨



## 4. 개선된 RNN 모델

### ■ 기본 RNN 모델의 문제점

- 메모리 셀은 한정된 길이의 패턴을 기억할 수 있음
  - ➡ 시퀀스의 길이가 길어지면 최근 정보만 남고, 오래된 정보가 손실되어 모델의 성능 저하 발생

### ■ 개선 방법

- 중요한 정보는 보존하고 덜 중요한 정보는 잊게 하는 구조
  - LSTM(Long Short-Term Memory)
  - GRU(Gated Recurrent Unit)



## 4. 개선된 RNN 모델

### ■ LSTM(Long Short-Term Memory)

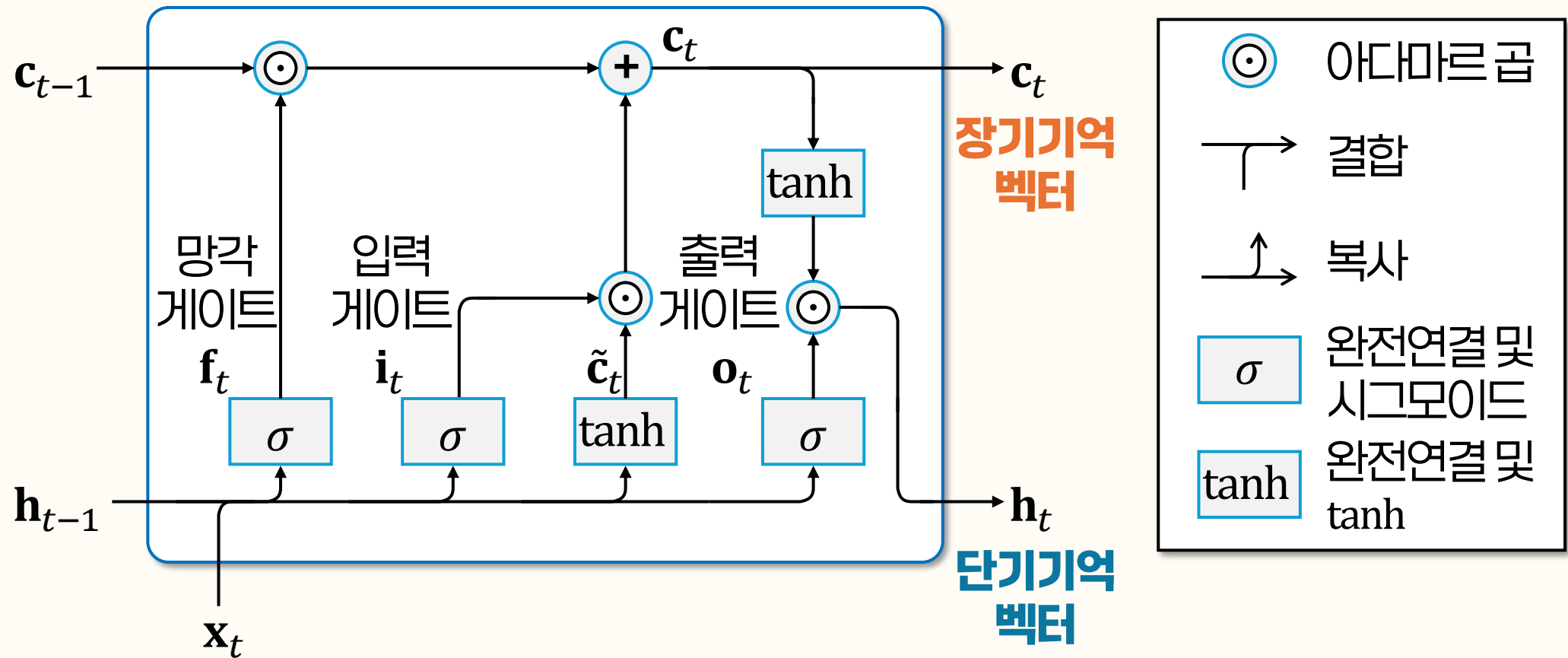
- 어떤 정보를 유지하고 어떤 정보를 지워야 하는지를 제어하는 메커니즘을 사용

➡ 은닉 상태 벡터를 **장기 기억 벡터**와 **단기 기억 벡터**로 표현함

- 정보의 유지 및 활용을 위한 요소 도입
  - 망각 게이트(forget gate): 장기 기억에서 어떤 정보를 잊어야 할지 결정함
  - 입력 게이트(input gate): 입력 정보 중 어떤 것이 중요하여 기억해야 할지 결정함
  - 출력 게이트(output gate): 출력을 만들 때 어떤 정보만 남겨야 할지 결정함

# 4. 개선된 RNN 모델

## ■ LSTM(Long Short-Term Memory)



## 4. 개선된 RNN 모델

### ■ GRU(Gated Recurrent Unit)

- LSTM을 단순화한 구조
  - LSTM의 장기기억 및 단기기억 벡터를 통합하여 하나의 상태 벡터  $h$ 만 사용함
  - 하나의 게이트 제어기가 망각 및 입력 게이트를 동시에 제어
    - 망각 게이트가 열리면 입력 게이트는 닫히고, 망각 게이트가 닫히면 입력 게이트가 열림
  - 출력 게이트는 제거
- LSTM과 유사한 성능을 내면서도 좀더 빠르게 동작함

## 5. 자연언어 번역과 RNN

### ■ 인코더-디코더(encoder-decoder) 구조

#### 인코더

- 하나의 언어로 표현된 문장을 문맥 벡터(context vector)로 인코딩함



#### 디코더

- 문맥 벡터를 입력으로 하여 번역된 문장을 생성함

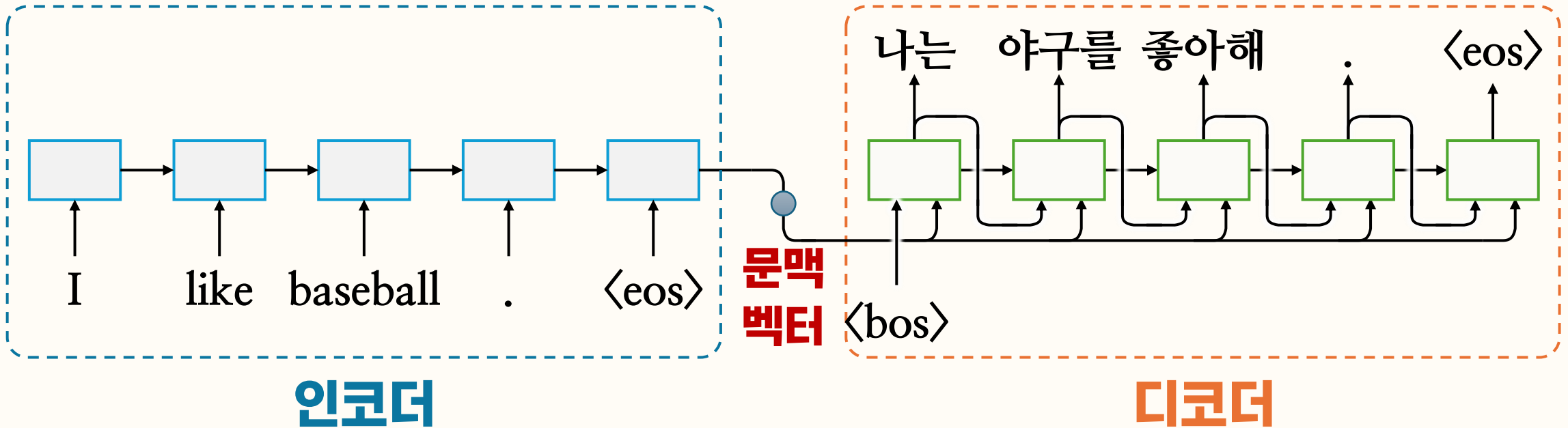


인코더와 디코더는 기본 RNN, LSTM, GRU등이 사용될 수 있음

## 5. 자연언어 번역과 RNN

### 인코더-디코더(encoder-decoder) 구조

- seq2seq: 시퀀스를 입력으로 받아 다른 시퀀스로 변환하는 인코더-디코더 구조



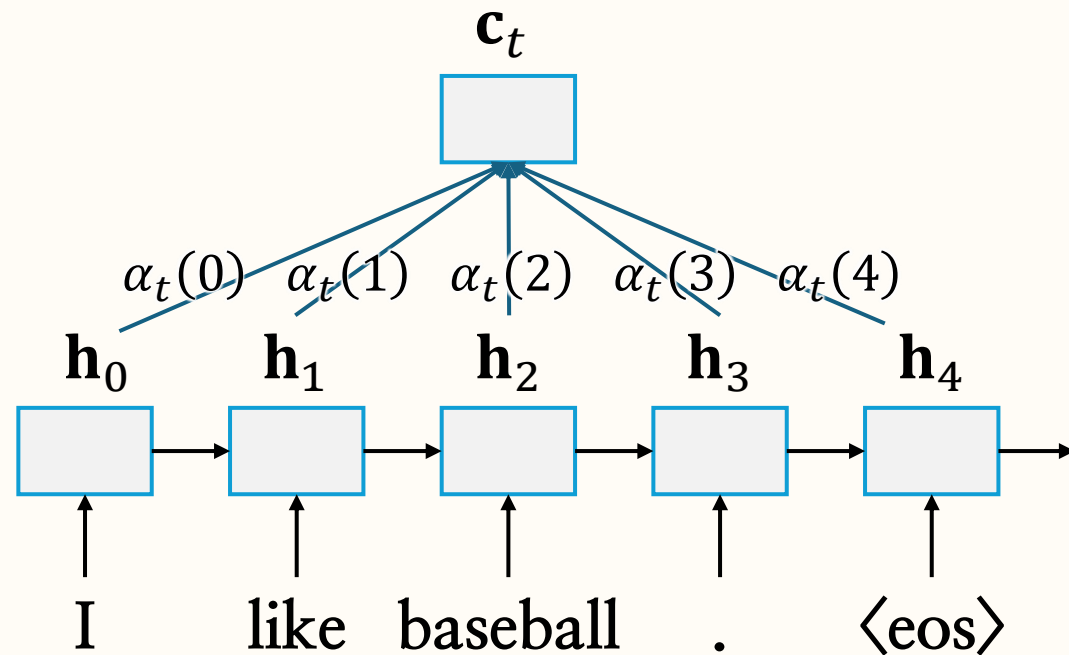
 디코더의 모든 시점에 동일한 문맥 벡터가 사용됨

 **문제점:** 시퀀스가 긴 경우 앞쪽의 정보가 문맥 벡터에 남지 못함

## 5. 자연언어 번역과 RNN

### ■ 어텐션(attention) 메커니즘

- 디코더의 시점에 따라 인코더의 각 시점에서 은닉 상태의 가중 평균으로 문맥 벡터를 만들어 디코딩하는 방식
  - 디코더의 각 시점에서 가장 관계가 깊은 입력에 주의를 기울일 수 있음
- 가중치는 모델의 손실을 줄일 수 있도록 훈련하여 결정





트랜스포머

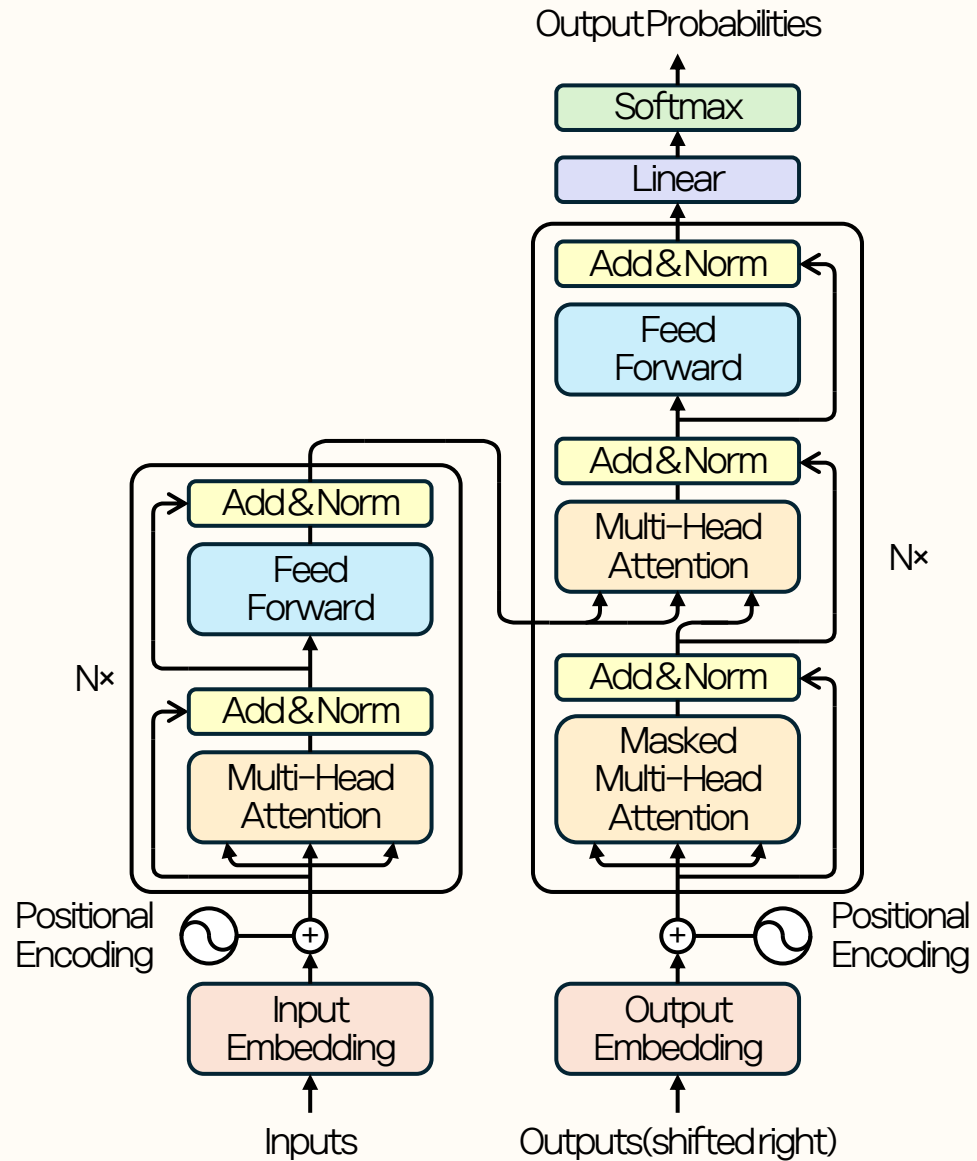
# 1. 트랜스포머의 개념

## ■ 트랜스포머(transformer)의 핵심 아이디어

- 인코더와 디코더를 구현하는데 RNN 대신 셀프 어텐션(self-attention) 메커니즘을 사용함
  - 인코더: 입력 문장의 각 토큰에 대해 문맥에 맞는 표현을 생성
  - 디코더: 출력 문장을 생성
- 시퀀스를 순차적으로 처리하지 않고 전체 시퀀스를 한 단위로 처리하기 때문에 훈련과정을 병렬화할 수 있음
- 멀리 떨어진 단어 사이의 관계를 잘 표현할 수 있음
- 거대 언어 모델(Large Language Model: LLM)에 적합함
  - GPT, BERT 등에 활용



## 2. 트랜스포머의 구조



# 정리하기

- ✓ 심층 신경망은 층 수가 많을수록 특징의 수준을 풍부하게 함으로써 높은 표현력을 가질 수 있으나, 학습 품질이 오히려 떨어지는 문제가 발생할 수 있다.
- ✓ ResNet의 잔차 블록은 항등 스킵 연결을 통해 입력을 더함으로써, 입력과 학습하려는 출력의 차를 학습하게 한다.
- ✓ 잔차 블록을 여러 층 순차적으로 연결한 ResNet은 깊은 심층망을 구성하더라도 일반적인 심층망에 비해 최적화가 잘 이루어지는 결과를 보였다.
- ✓ 순환 신경망(RNN)은 사이클을 형성하는 순환연결을 포함하며, 이에 따라 이전 입력에 대한 기억을 유지할 수 있다.

# 정리하기

- ✓ LSTM이나 GRU는 기본 RNN에서 오래된 정보에 손실이 발생하는 문제를 개선할 수 있다.
- ✓ seq2seq는 시퀀스를 입력으로 받아 다른 시퀀스로 변환하는 인코더-디코더 구조이다.
- ✓ 어텐션 메커니즘은 디코더의 각 시점에서 가장 관계가 깊은 입력에 주의를 기울일 수 있게 한다.
- ✓ 트랜스포머는 인코더와 디코더를 구현하는 데 RNN 대신 셀프 어텐션(self-attention) 메커니즘을 사용한다. RNN에 비해 계산과정을 병렬화할 수 있으며, 멀리 떨어진 단어 사이의 관계를 잘 표현할 수 있다.

인공지능 ▶▶▶

한 학기동안  
수고많이하셨습니다.