



졸지말자 딥러닝

김승일
Seungil Kim



끝지말자 딥러닝



- 인공지능 (Artificial Intelligence)
- 머신러닝 (Machine Learning)
- 딥러닝 (Deep Learning)
- Convolutional Neural Network
- Recurrent Neural Network
- Applications

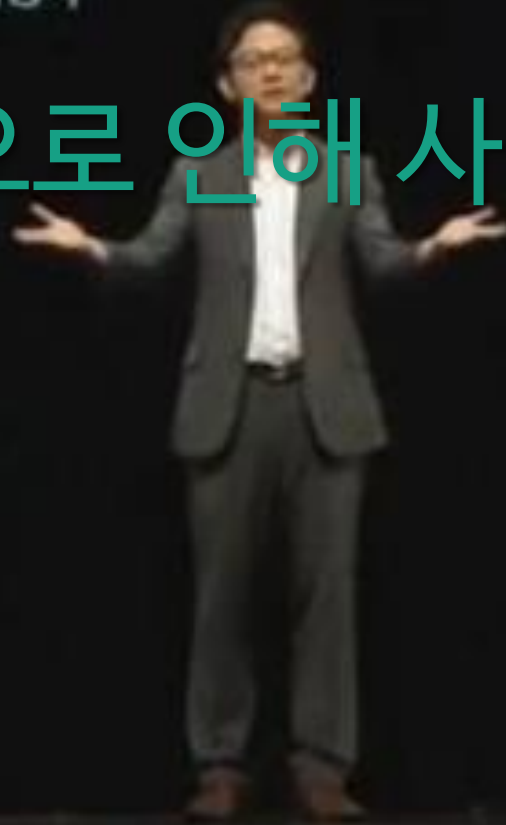
유럽연합(EU) 의회의 'AI로봇 결의안'

구분	주요 내용
법적 지위	• AI 로봇을 '전자인간'으로 인정
'킬 스위치'	• 로봇 작동을 멈추는 버튼 장착
주요 원칙	<ul style="list-style-type: none"> • 로봇은 인간을 위협하면 안 됨 • 늘 인간의 명령에 복종해야 함 • 로봇 역시 자신을 보호해야 함
핵심 권고	<ul style="list-style-type: none"> • EU 안에 AI기술·윤리기구 신설 • 고용 모델 및 조세시스템 개편
최종 의결	• 다음달 본회의 투표 실시 예정

EU의

AI로봇 가이드라인

인공지능으로 인해 사라질 직업들...



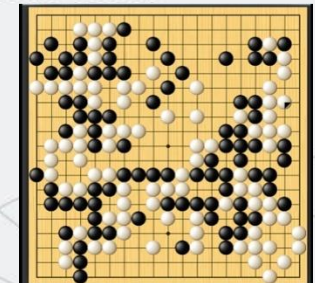
형견만리 인명구조원 카페 직원 은행원 의료기술사 동물 사육사 도서관 사서 KBS1
 택시운전사 미용사 회계사 호텔리어 임상실험가
 전기기계조립자
 의사 변호사
 부동산 중개인 치위생사 판사 이발사
 식당 주인 농산물과학자 항공공학자
 세무사 요리사 약사 자정학자
 신용분석가 배우
 운송업자 관광가이드
 교사 스포츠 심판
 아나운서 전기공학자
 제빵사 보험업자
 건설노동자봉재사
 원자력기술자 제약기술자

알파고를 이긴 유일한 인간



We've been hard at work improving AlphaGo, and over the past few days we've played some unofficial online games at fast time controls with our new prototype version, to check that it's working as well as we hoped. We thank everyone who played our accounts Magister(P) and Master(P) on the Tygem and FoxGo servers, and everyone who enjoyed watching the games too! We're excited by the results and also by what we and the Go community can learn from some of the innovative and successful moves played by the new version of AlphaGo.

Having played with AlphaGo, the great grandmaster Gu Li posted that, "Together, humans and AI will soon uncover the deeper mysteries of Go". Now that our unofficial testing is complete, we're looking forward to playing some official, full-length games later this year in collaboration with Go organisations and experts, to explore the profound mysteries of the game further in this spirit of mutual enlightenment. We hope to make further announcements soon!



인공지능이 두려우신가요?



Artificial Intelligence

인공지능이란 무엇일까요?

인공지능을
단순화해서 생각해봅시다.



인공지능



Dog

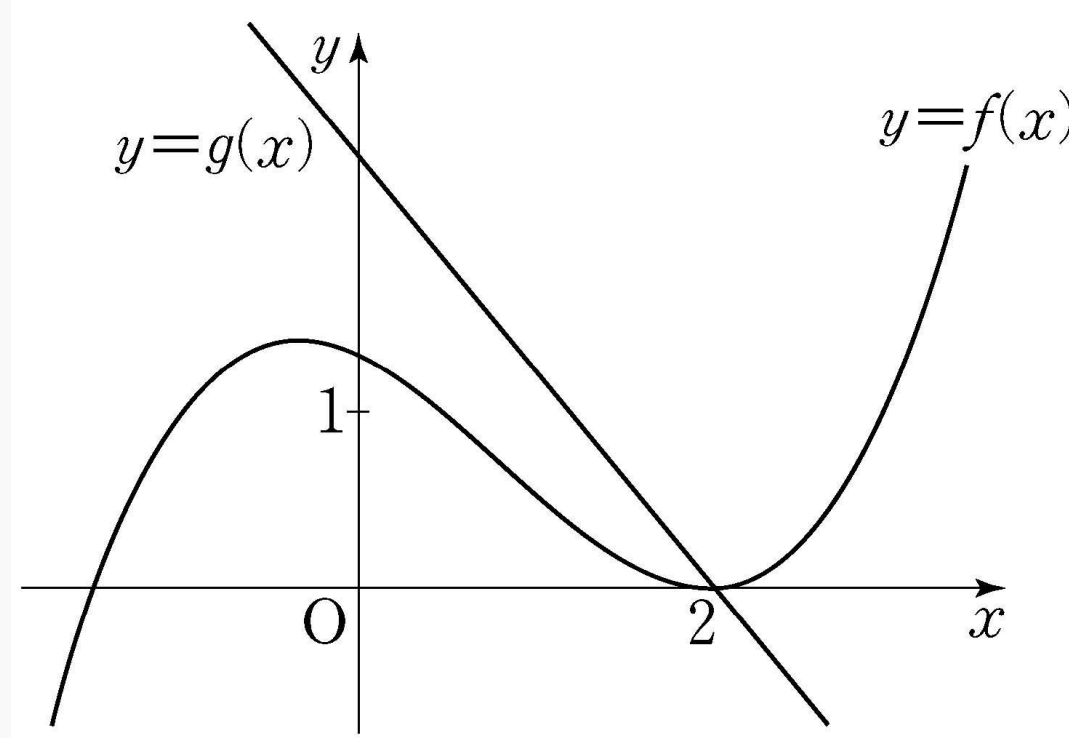


Cat

입력이 주어지면

출력을 내보낸다.

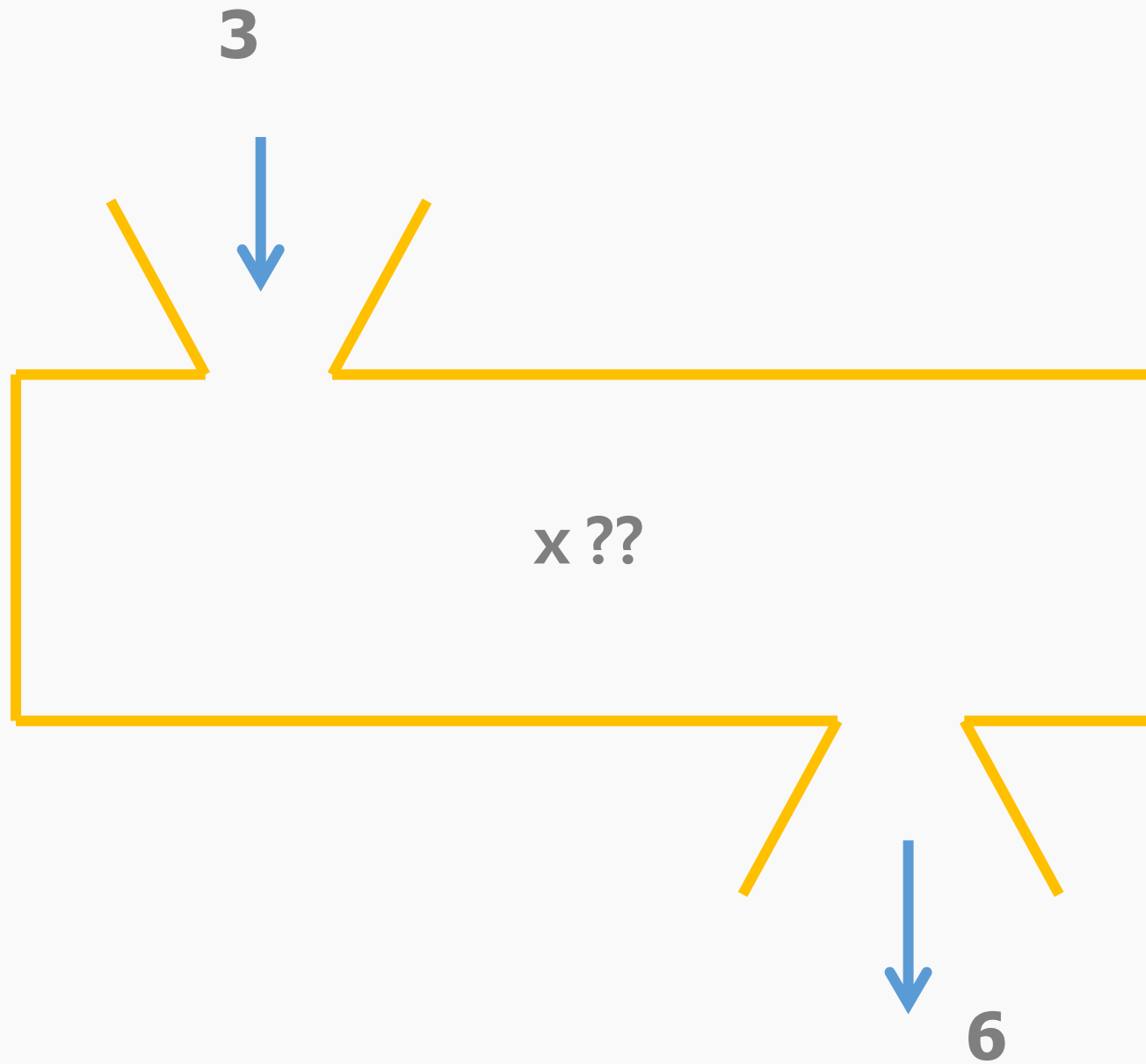
입력이 주어지면, 출력을 내보낸다.
우리 이런 걸 어디서 봤죠?

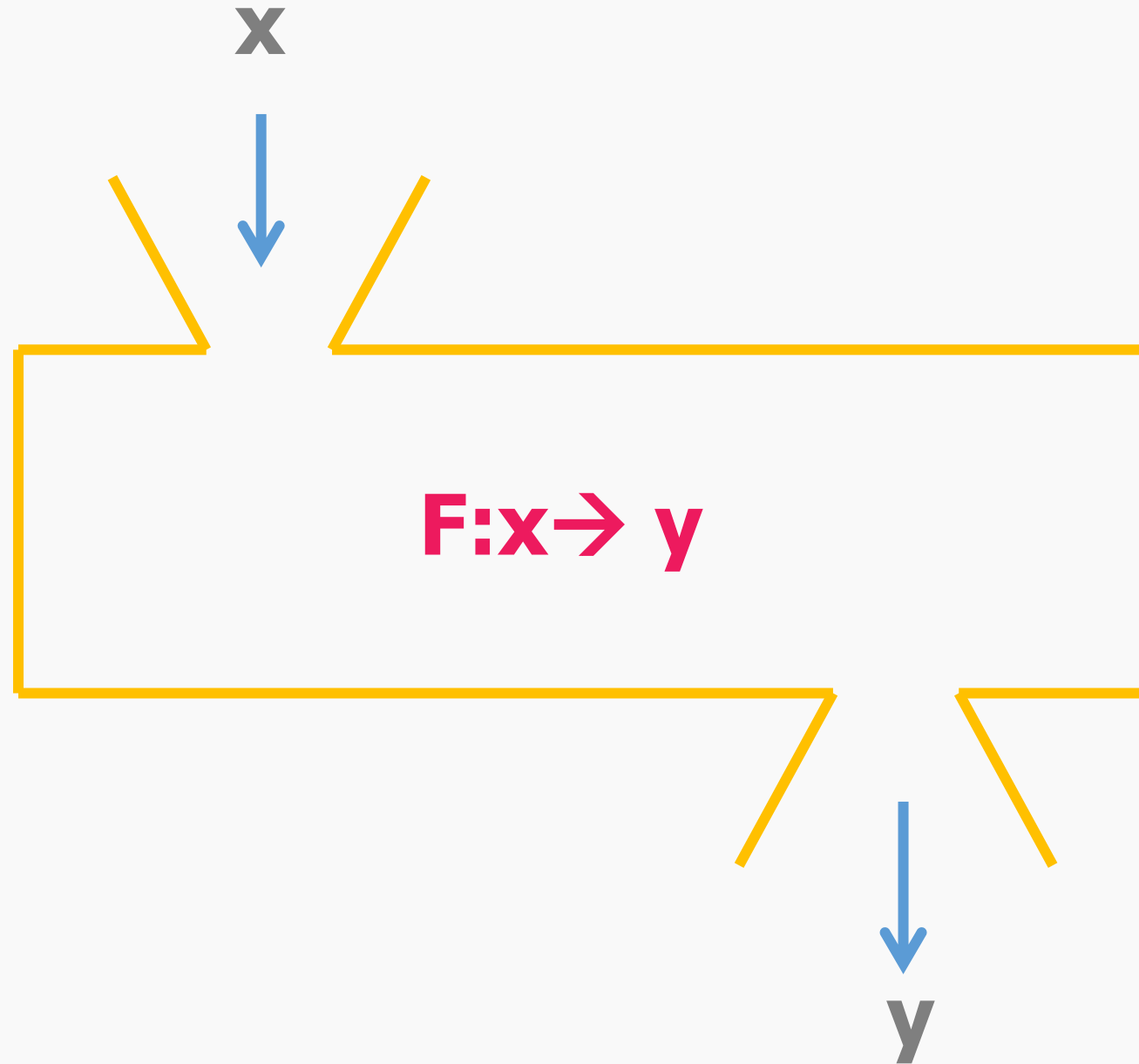


함수 (Functions)

System, Filter 라고도 불립니다.

더 쉽게는
초등학교 때 배운
수수께끼 상자

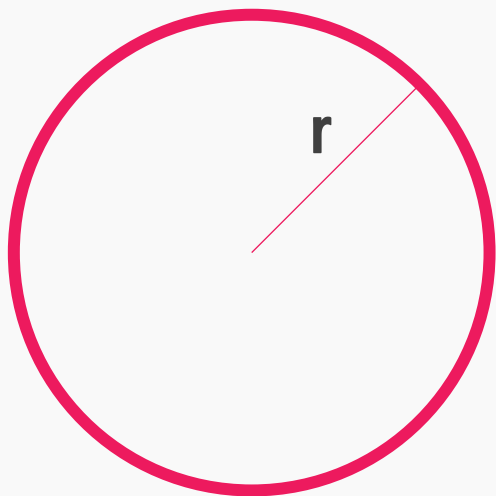




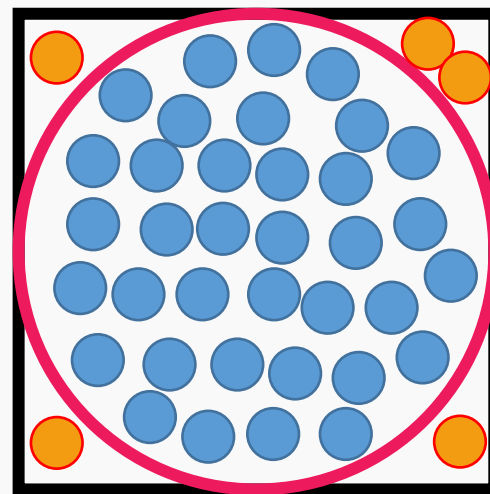


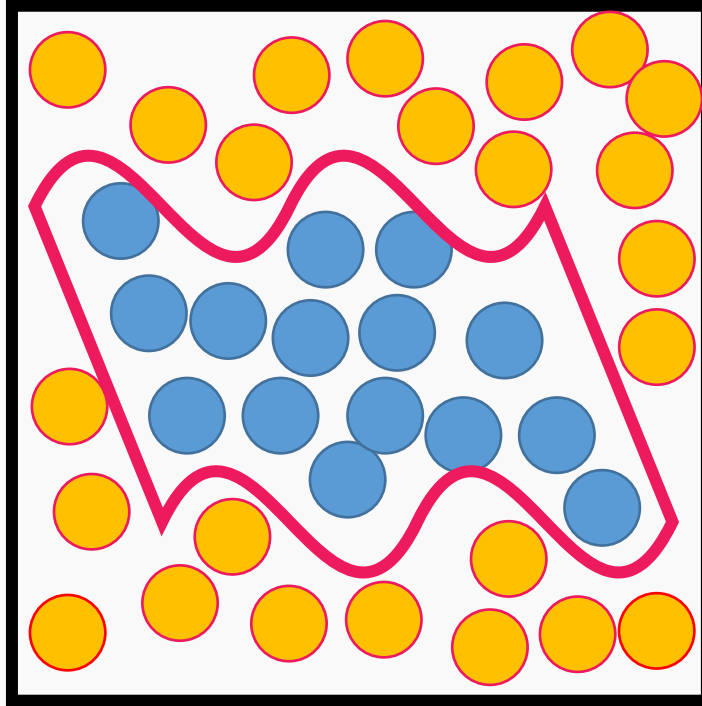
인공지능 별 거 아니죠?
무서울 것 없습니다.
근데 **왜** 갑자기 무서워 졌을까요?

원의 넓이를 구하는 함수는?



$$A = \pi r^2$$





이런 도형의 넓이도?

수학적으로 표현할 수 없었던
복잡한 인간의 두뇌를
데이터를 기반으로 흉내낸다.



끝지말자 딥러닝



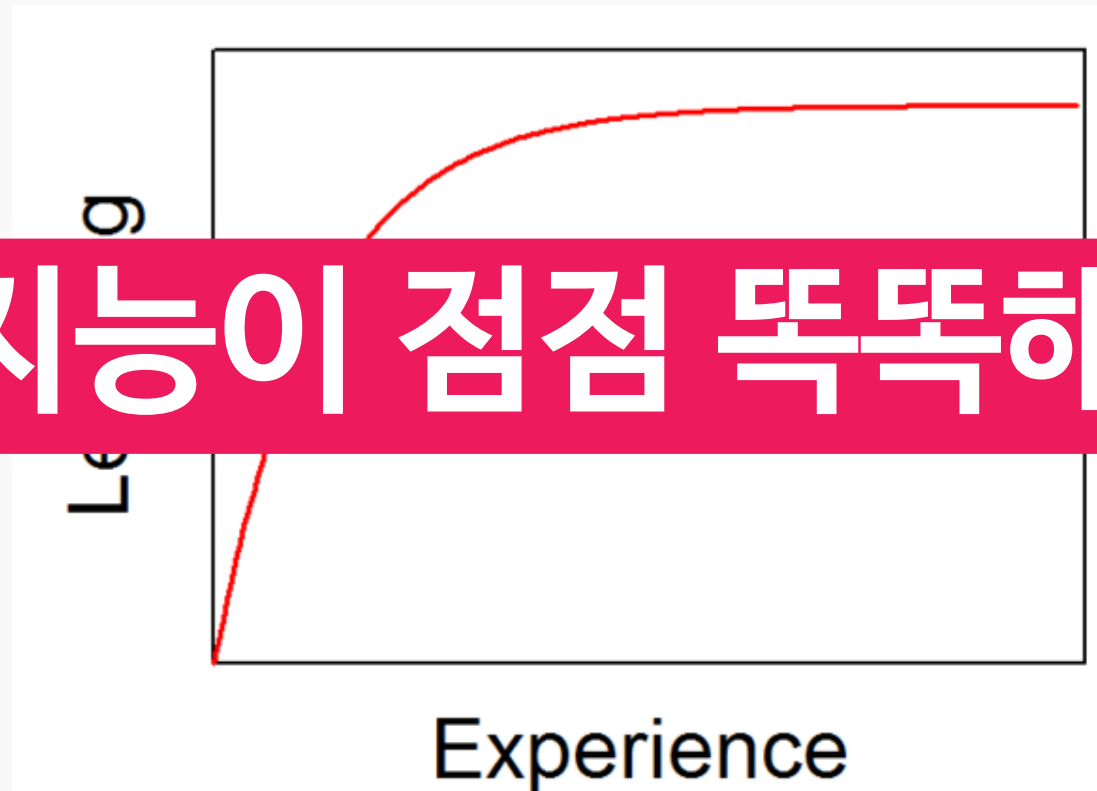
- 인공지능 (Artificial Intelligence)
- 머신러닝 (Machine Learning)
- 딥러닝 (Deep Learning)
- Convolutional Neural Network
- Recurrent Neural Network
- Applications

인공지능? 머신러닝?

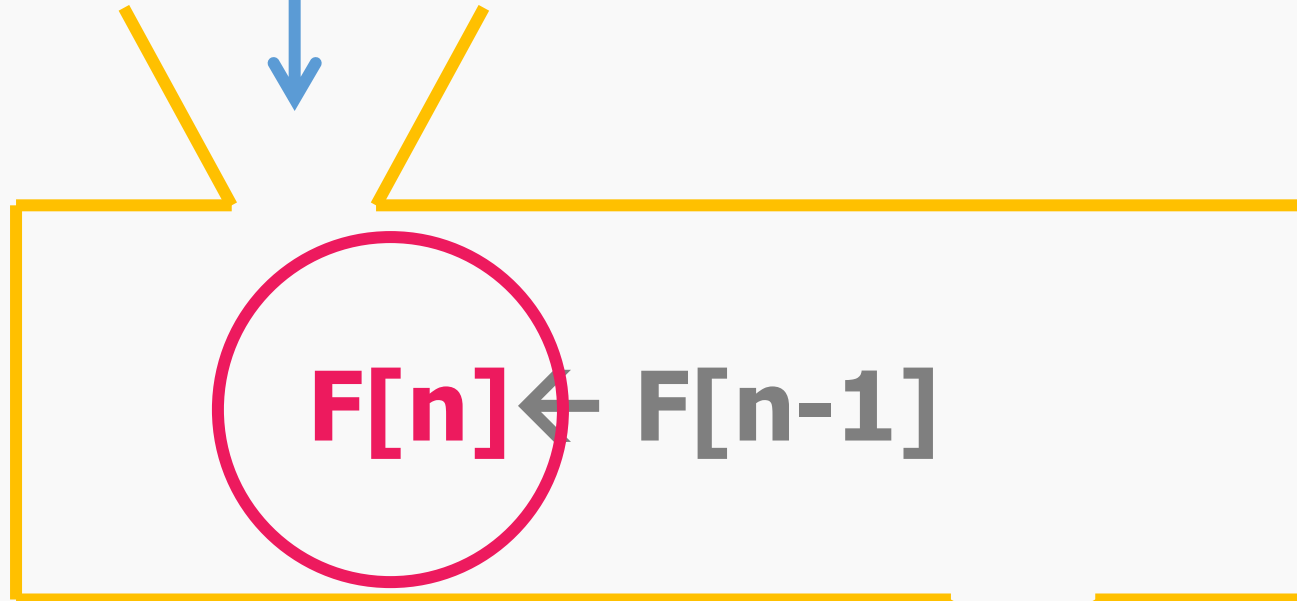
머신러닝은 또 뭘까요?



러닝 (Learning) = 학습
→ Adaptation/Update



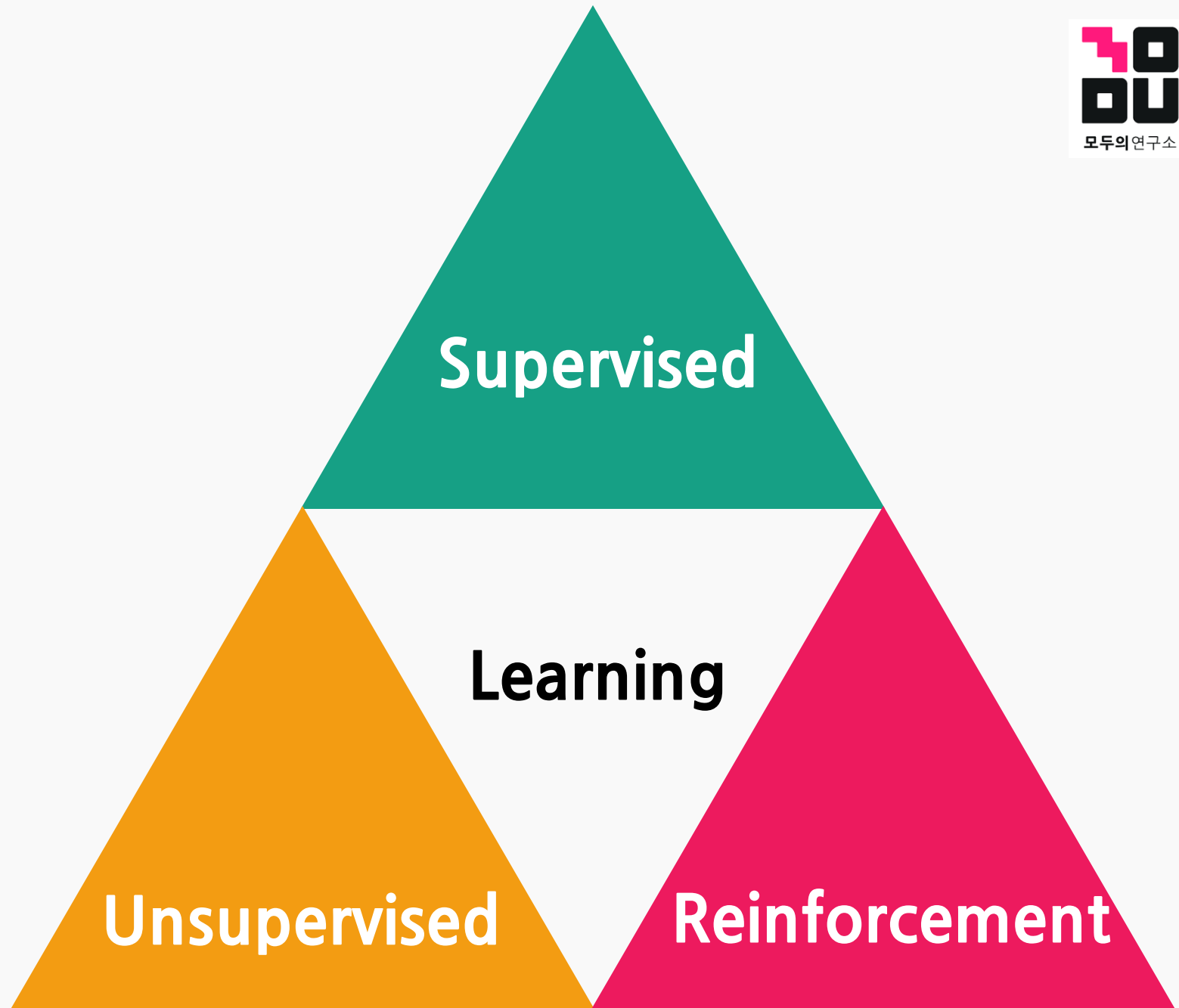
인공지능이 점점 똑똑해진다.

$X[n]$
 n : data index

 $F[n] \leftarrow F[n-1]$

Data가 늘어날수록
점점 인공지능 알고리즘이
학습(Learning)한다.

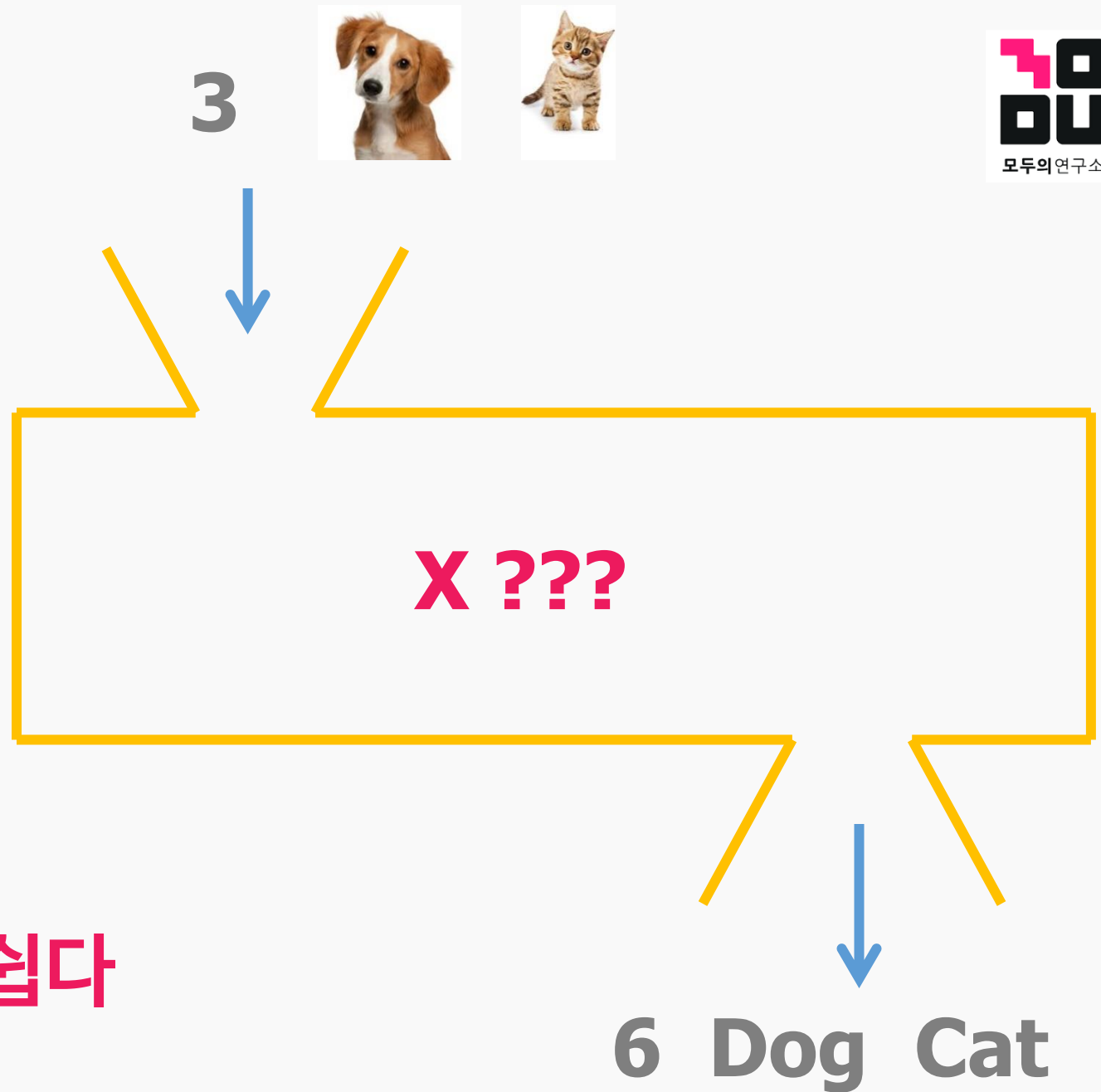

 $Y[n]$

머신러닝의 세가지 타입



Supervised Learning

- 정답이 주어진다.
- (비교적)문제풀이가 쉽다



3



Unsupervised Learning

X ???

미지수 2개, 방정식 1개
풀 수 있나요?



???

$$(x-1)(y-1) = 1,$$

x 와 y 를 구하라.

이것은 풀 수 있나요?

(x, y 는 자연수)

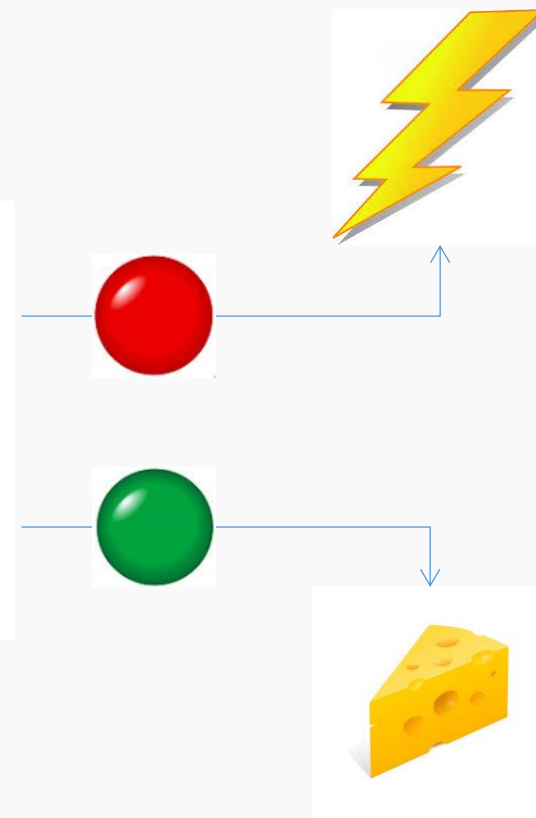
좋은 조건이 주어지거나 잘 찍는 수 밖에...

- 특정 조건이 있을 때만 정답이 주어질 수 있다.
- 기본적으로 문제풀이가 어렵다.

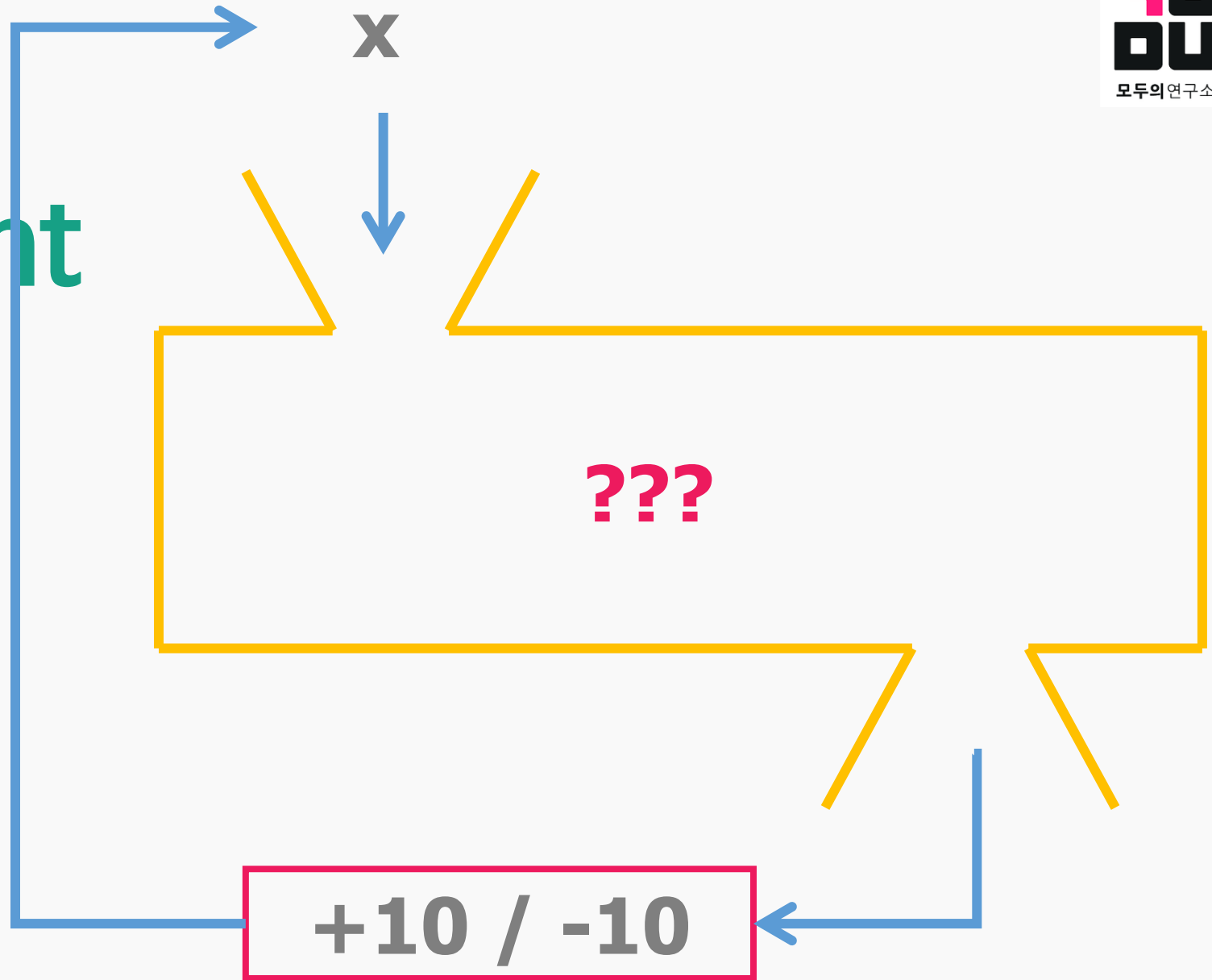


Reinforcement Learning

- (정답이 아닌) reward가 주어진다.



Reinforcement Learning

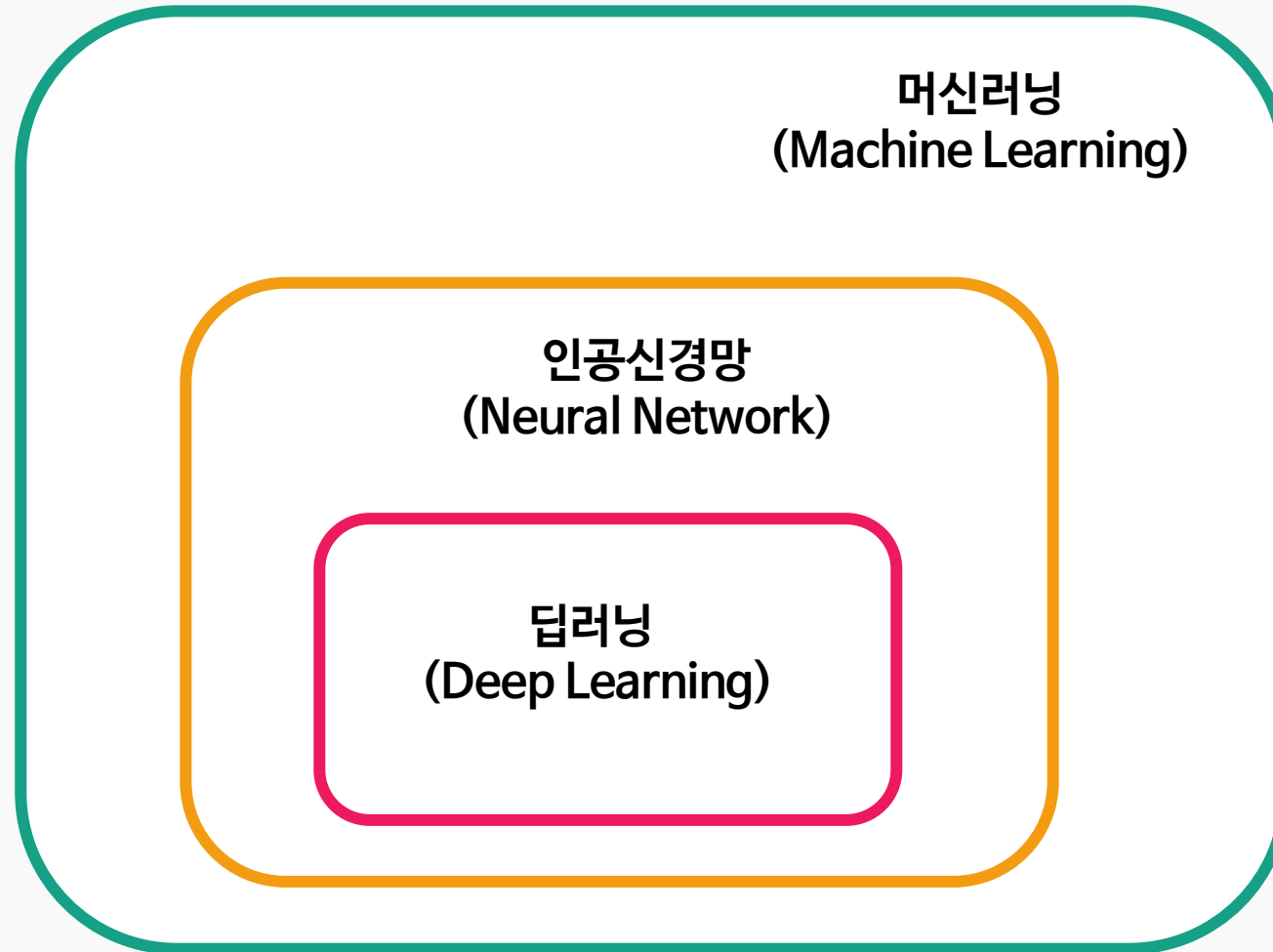


끝지말자 딥러닝

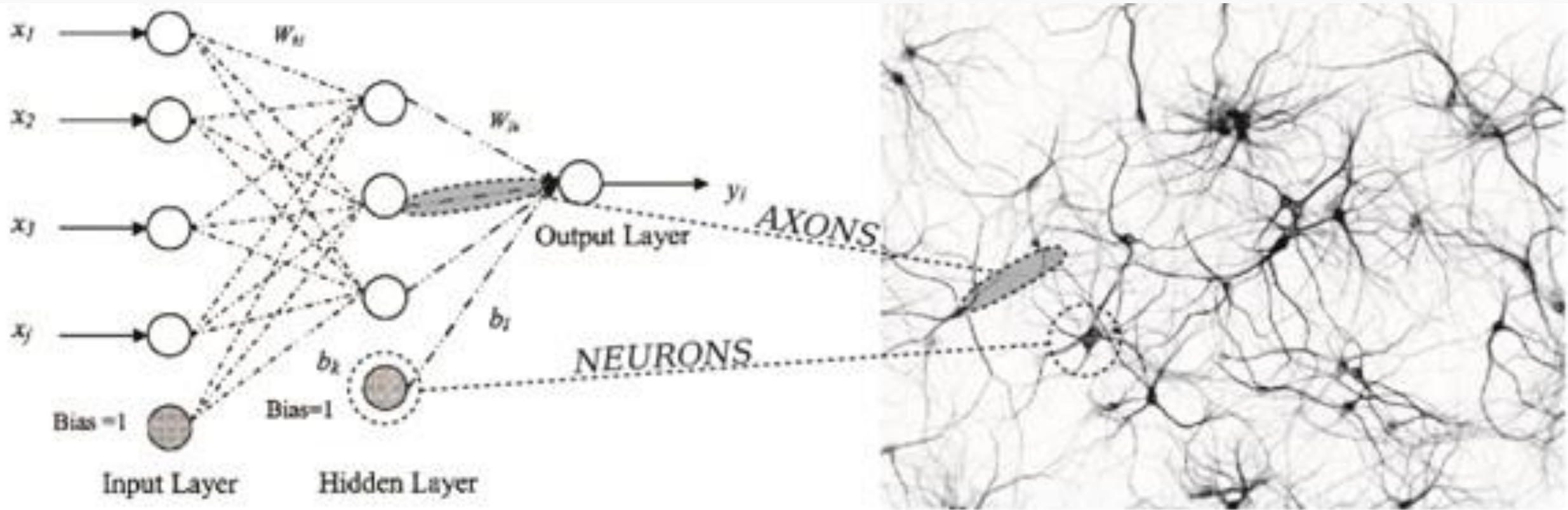


- 인공지능 (Artificial Intelligence)
- 머신러닝 (Machine Learning)
- **딥러닝 (Deep Learning)**
- Convolutional Neural Network
- Recurrent Neural Network
- Applications

딥러닝? 머신러닝?

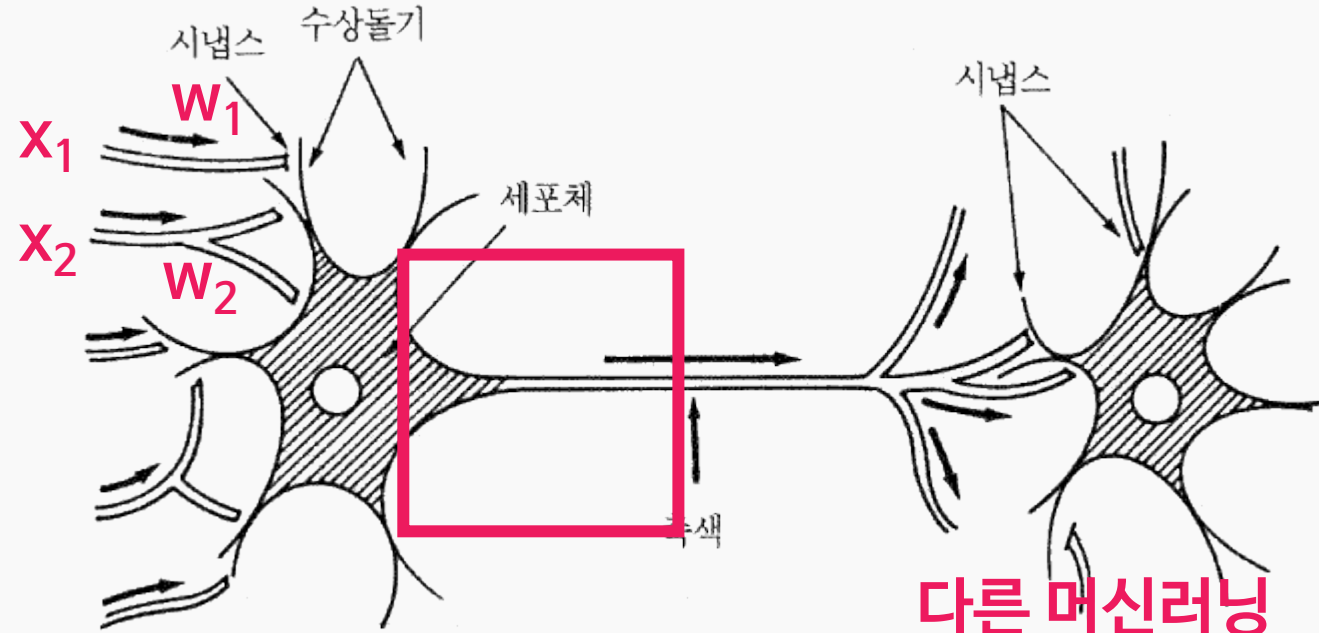


딥러닝을 이해하기 위해서는 인공신경망을 알아야 한다.

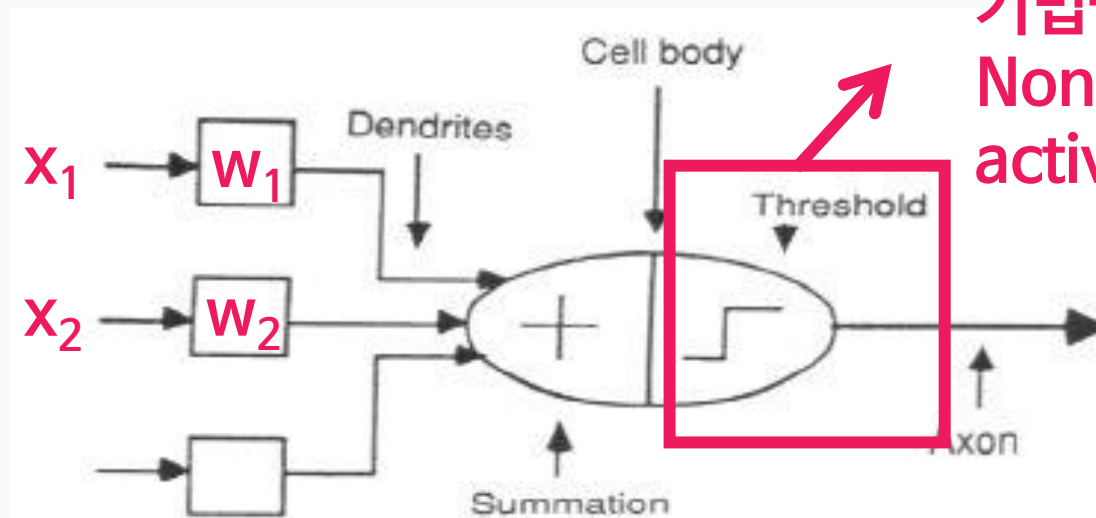


이제부터 조금 어려워집니다.
딥러닝이 원래 진입장벽이 있어요.

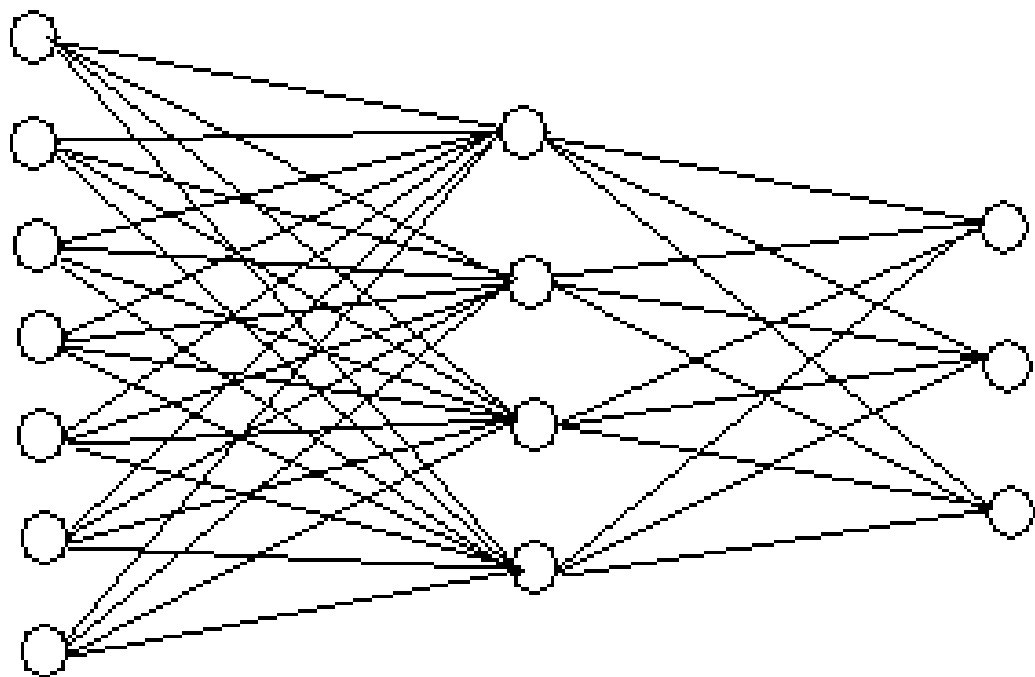
뉴런과 인공뉴런



다른 머신러닝
기법들과의 차이점 1:
Nonlinear (복잡한)
activation function



Schematic Diagram of a Neural Network



Input Units

Hidden Units

Output Units

Flow of Activation

다른 머신러닝
기법들과의 차이점 2 :
Nonlinear function의
Nonlinear function의
Nonlinear function ...



엄청 복잡한 함수
(인공지능)을
만들 수 있다.

다층 레이어 (Multiple Layer)

Hidden layer가 2개이상인 NN를
Deep Learning이라고 부른다.

오~ 그러면 레이어가 많을수록
높은 지능의 인공지능을
(복잡한 함수를)
만들 수 있겠는데!!!

레이어 10000개 만들 수 있을까요?

쉽지 않습니다.

(1) Overfitting (2) Vanishing Gradient

Neural Network이 나온지 오래되었지만,
그동안 이걸 풀지 못해서 암흑기에..
그러나, 이 두가지 문제를 해결하면서
딥러닝의 중흥기가 시작됨.

차근차근 알아봅시다

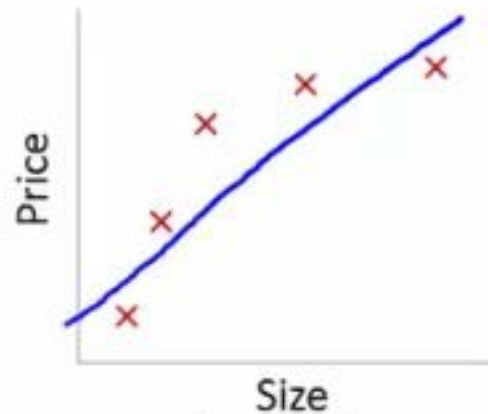
(1) Overfitting

Overfitting 이란?

연습문제는 엄청 잘 푸는데 시험만 보면 망함

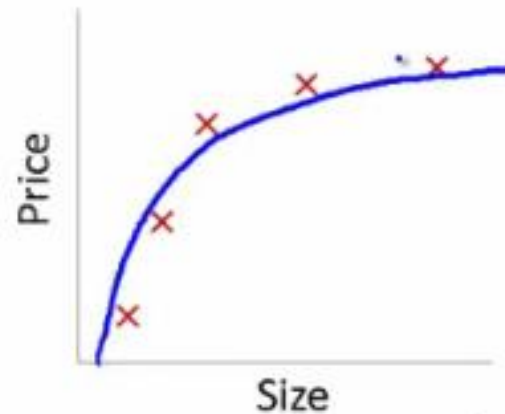
집넓이를 가지고 집값을 맞추는
인공지능(=함수)

오히려 집값이 떨어짐



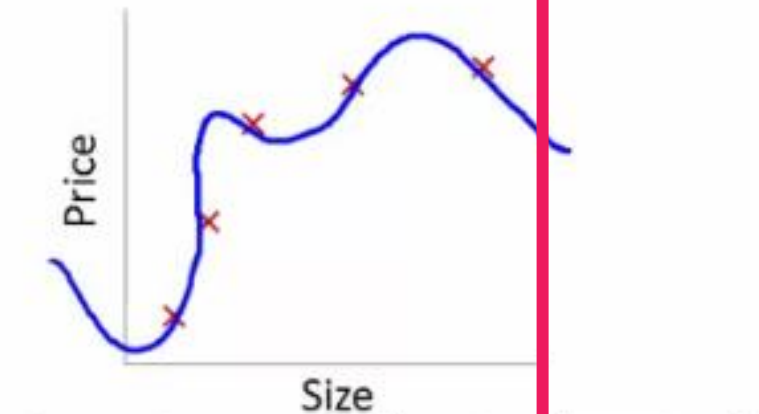
$$\theta_0 + \theta_1 x$$

High bias
(underfit)



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

"Just right"

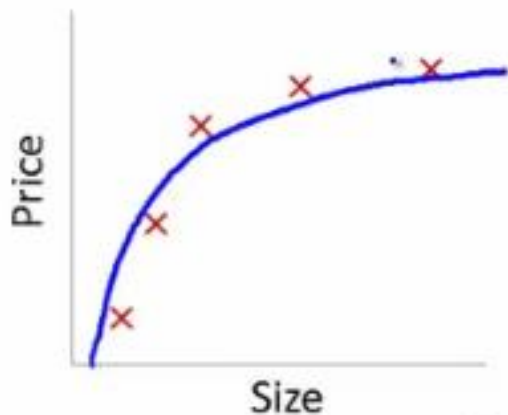


$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

High variance
(overfit)

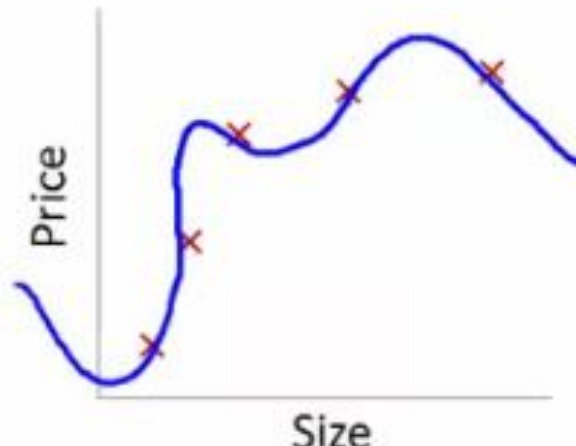
Overfitting은 왜 생길까?

(1) 풀어야 되는 문제는 간단한데, 모델이 너무 복잡...



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

“Just right”



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

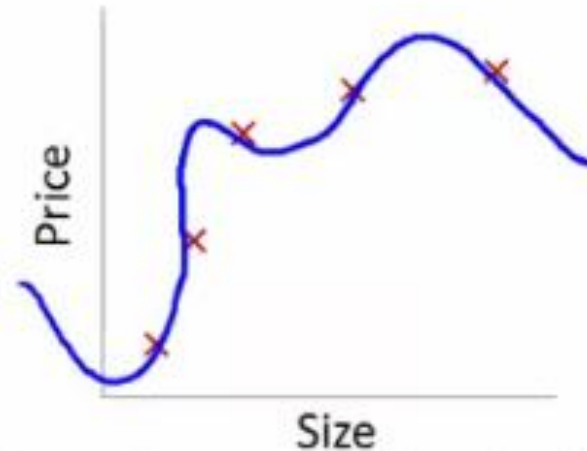
High variance
(overfit)

2차함수면 충분한데 4차함수로 모델링을 했다면?

Overfitting은 왜 생길까?

(2) 주어진 데이터는 몇 개 없는데, 모델이 너무 복잡...

Big Data



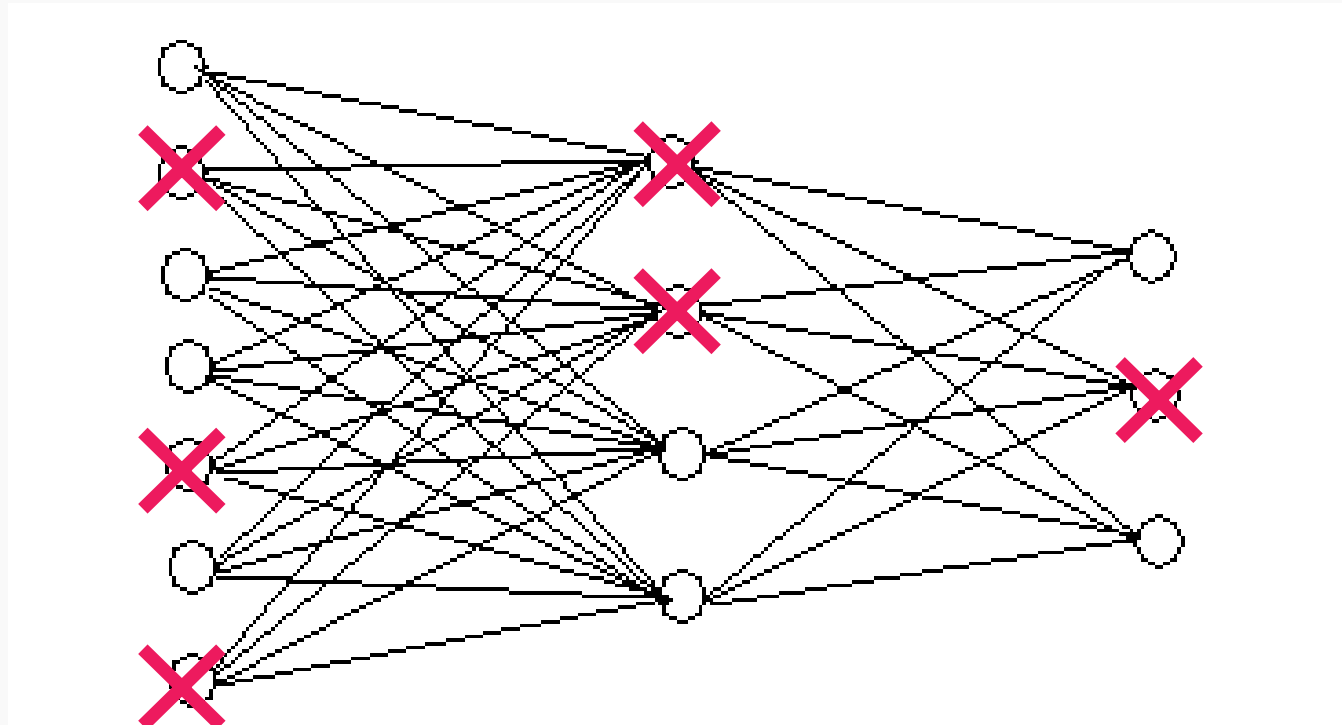
$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

High variance
(overfit)

딥러닝은 모델이 엄청 복잡하다.
수백만개의 파라미터...

Overfitting이 일어나기 쉽다.
어떻게 하면 좋을까???


Dropout



랜덤하게 뉴런을 끊음으로써,
모델을 단순하게 만든다.

차근차근 알아봅시다

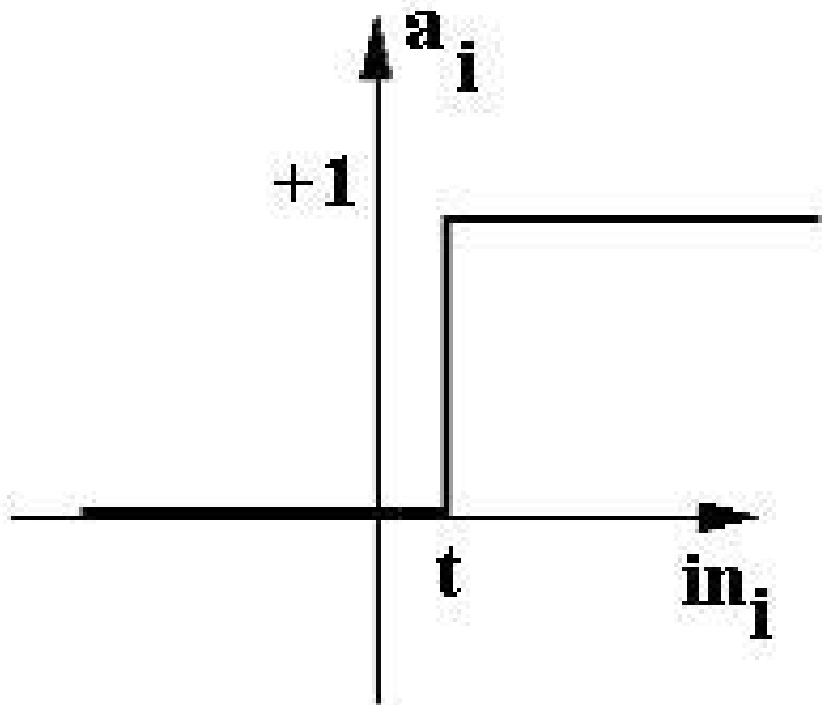
(2) Vanishing Gradient



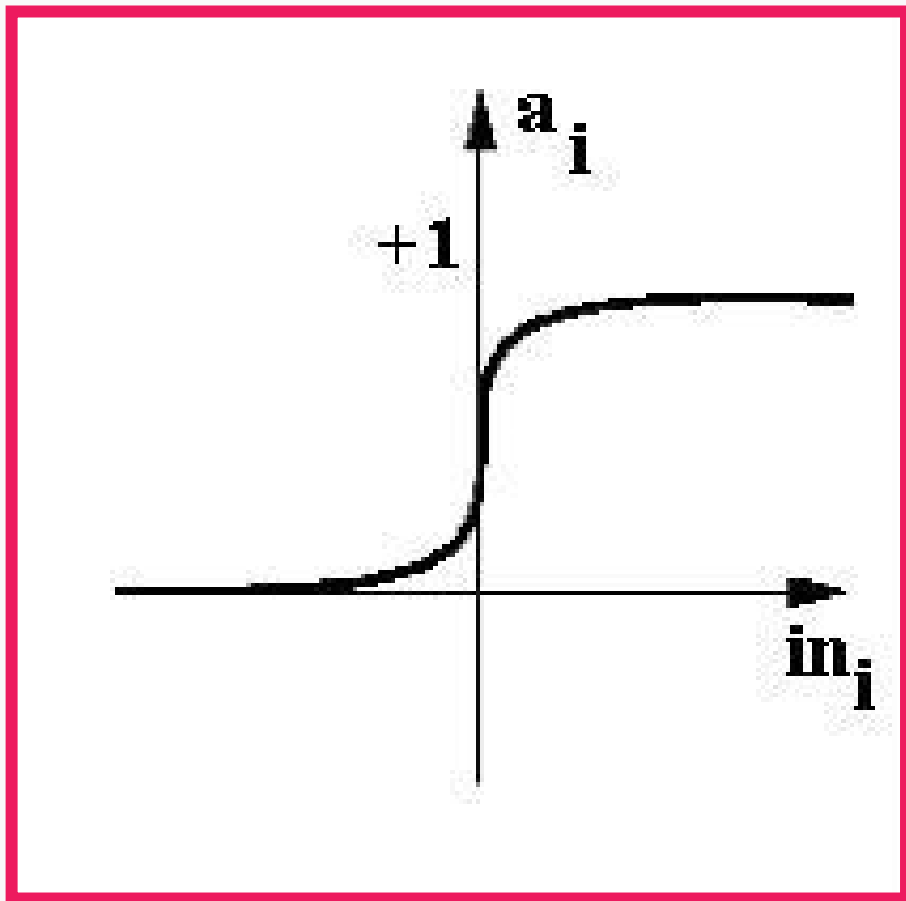
더 이상의
자세한 설명은
생략한다.

(Deep) Learning을 하기 위해서는
여자저차해서 **미분**을 이용합니다.

학습되는 양 = 미분값 * 출력값



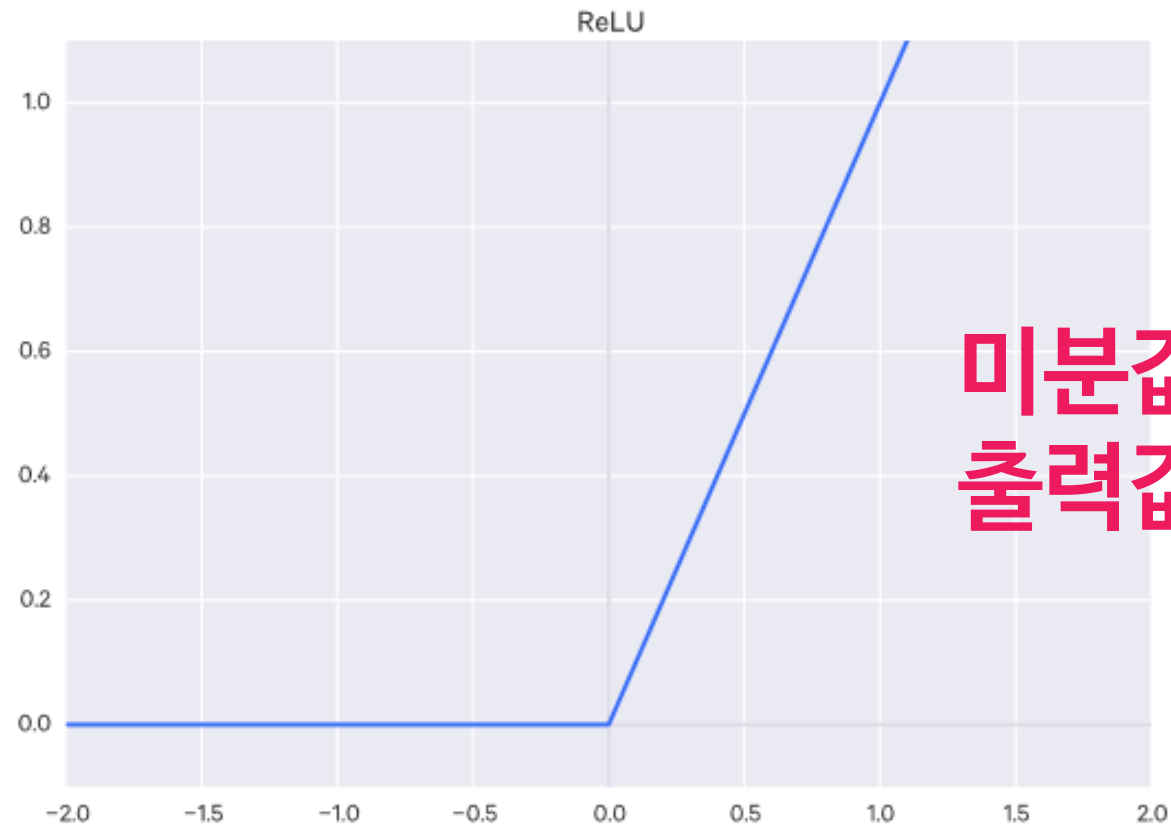
- 미분을 하면 0이 나와서, Learning이 불가함



- (1) 대부분의 영역에서 미분값이 0에 가깝다
- (2) 출력의 최대값이 1이다.
출력값이 0.5라고 했을때, 4개의 레이어만 지나도 $1/16$

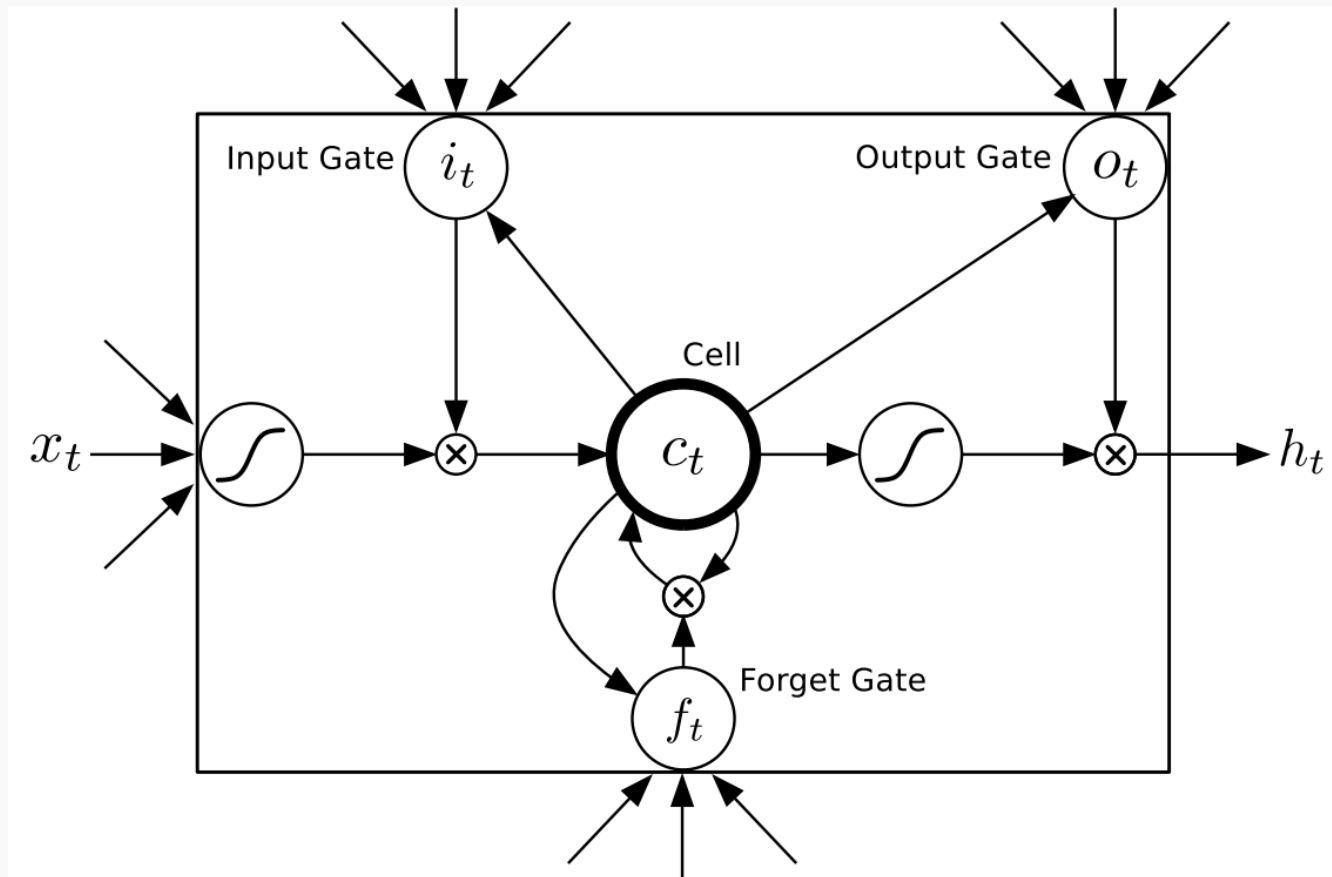
→ Vanishing Gradient

ReLU의 등장

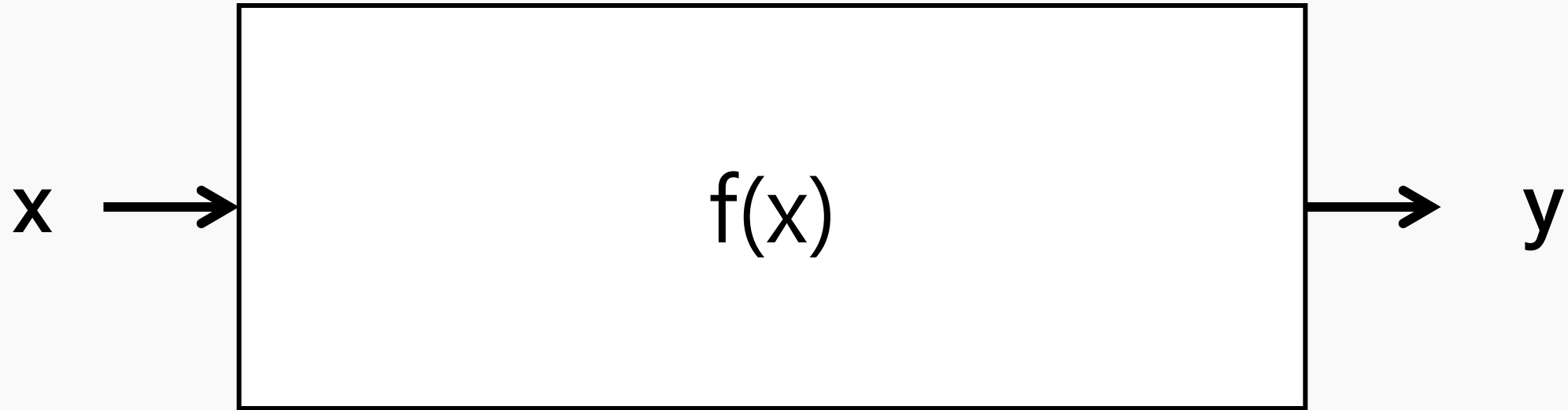


미분값이 1
출력값은 제한 없음

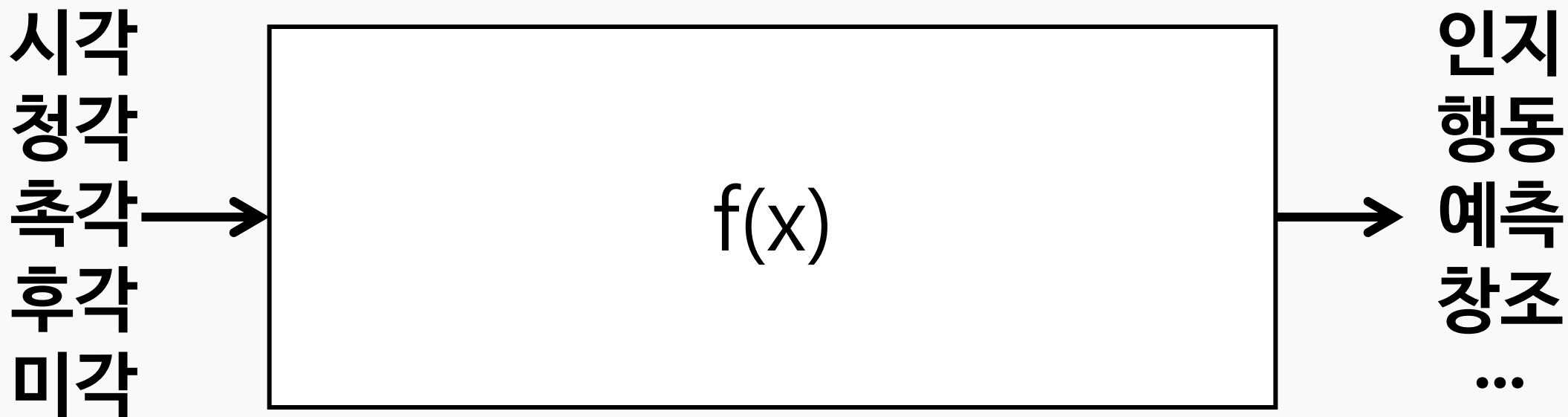
Long Short Term Memory



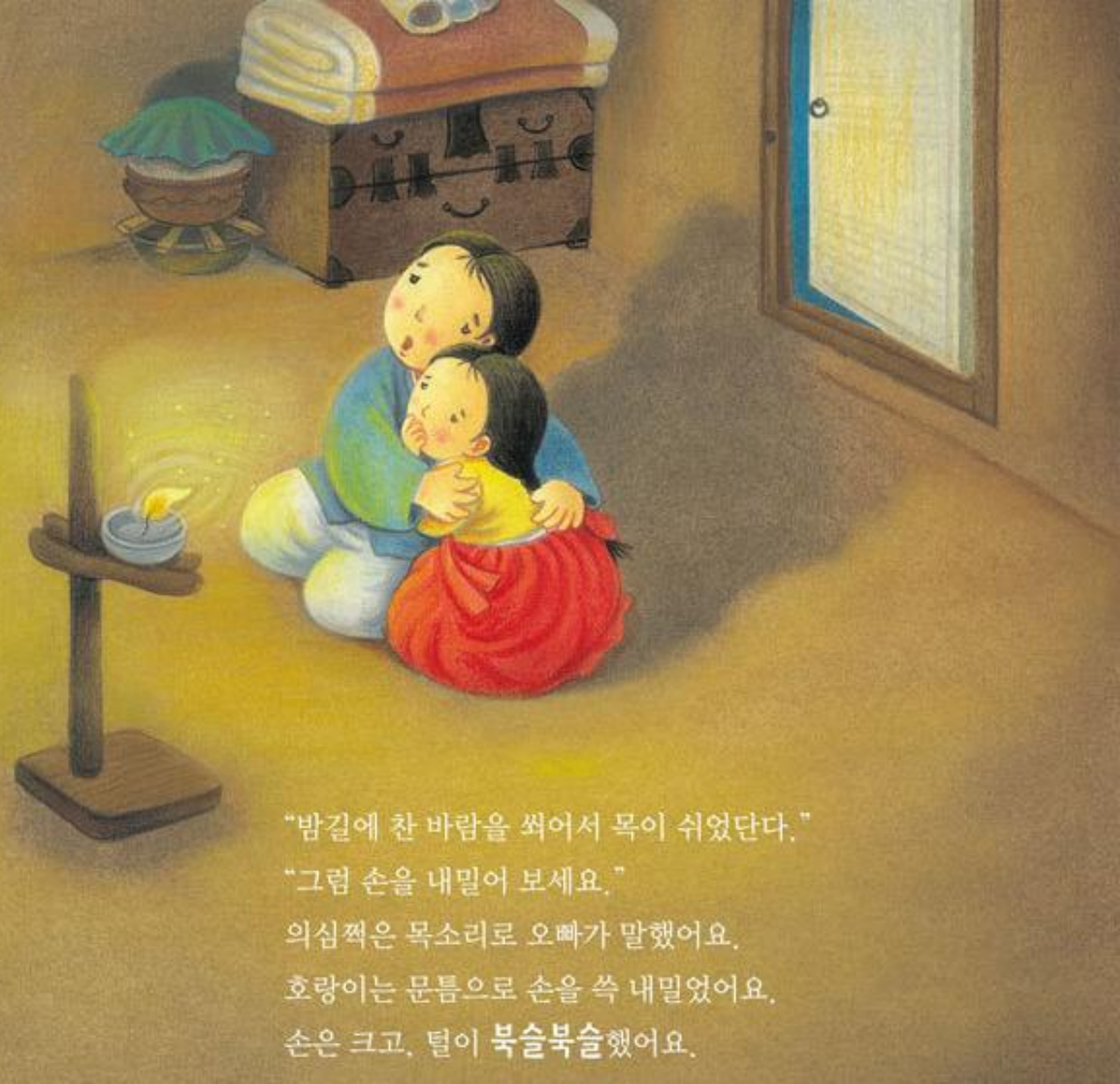
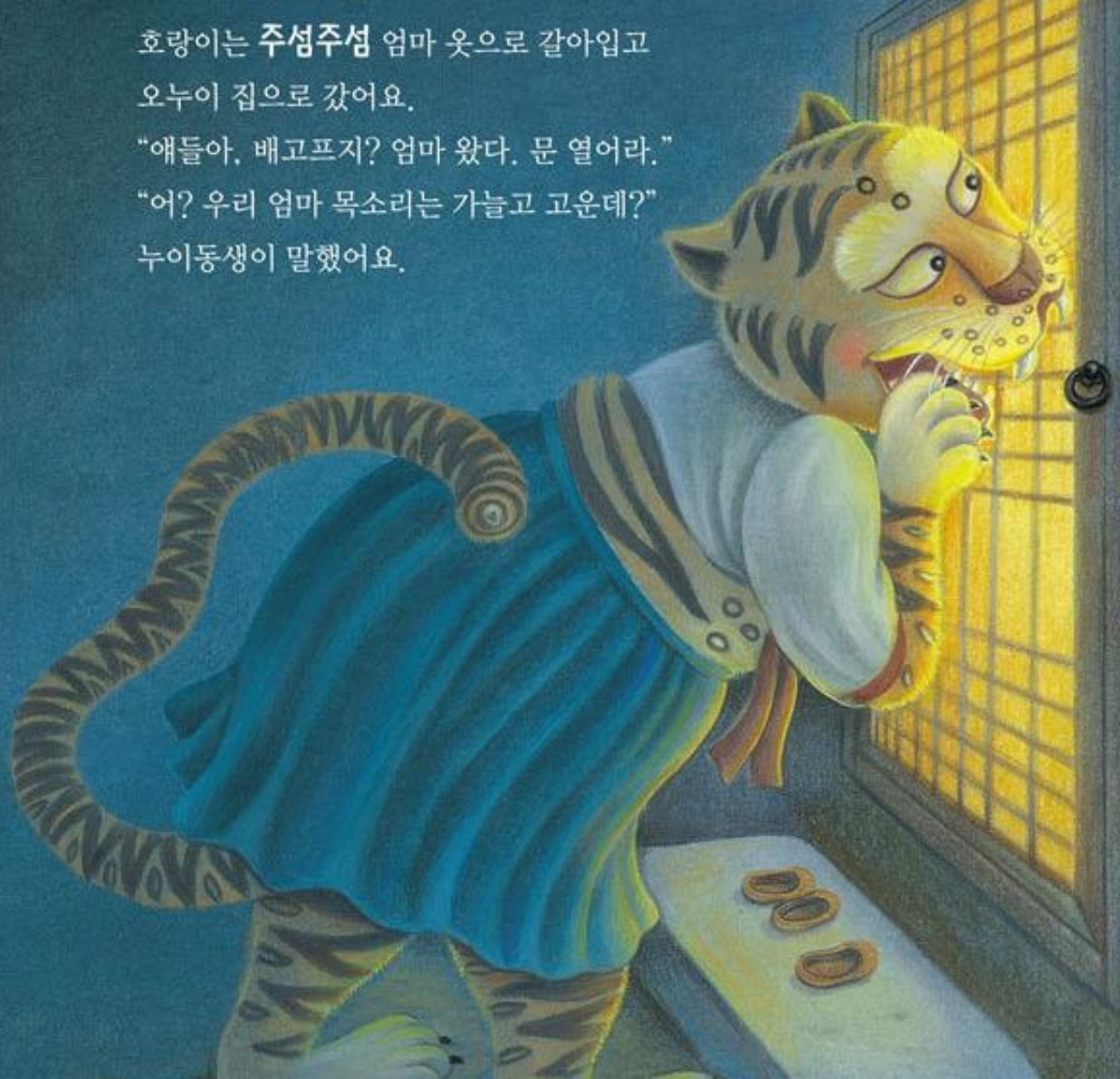
그렇다면 얼마나 큰 (복잡한) 함수를 만들 것인가?



인간의 두뇌만큼 복잡한 함수?



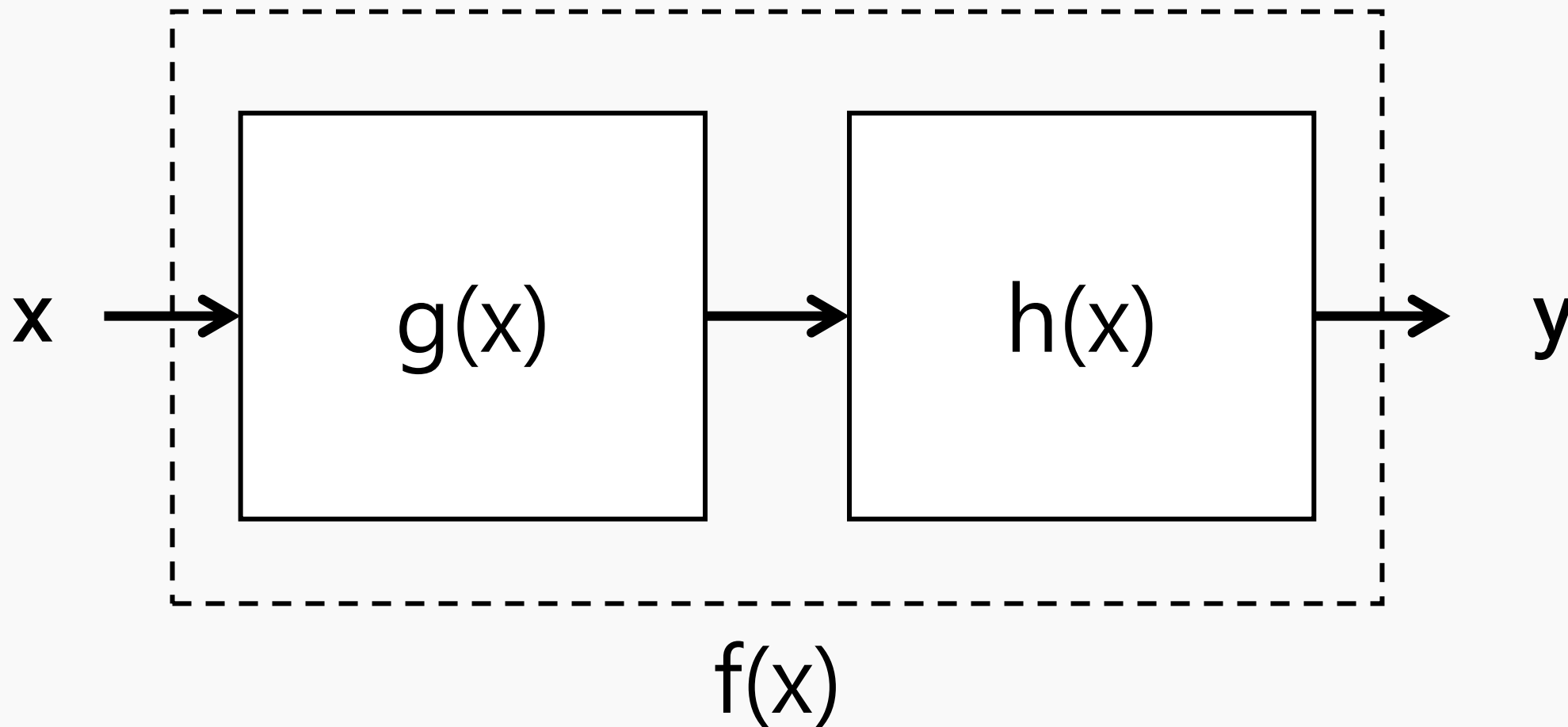
호랑이는 주섬주섬 엄마 옷으로 갈아입고
오누이 집으로 갔어요.
“애들아, 배고프지? 엄마 왔다. 문 열어라.”
“어? 우리 엄마 목소리는 가늘고 고운데?”
누이동생이 말했어요.



“밤길에 찬 바람을 쐬어서 목이 쉬었다.”
“그럼 손을 내밀어 보세요.”
의심쩍은 목소리로 오빠가 말했어요.
호랑이는 문틈으로 손을 쓱 내밀었어요.
손은 크고, 털이 북슬북슬했어요.

소리를 듣고 → 엄마인가?
손을 보고 → 엄마인가?

함수를 쪼갤 수 있다면?



어느 정도의 크기의 함수를 만들지
어떤 입력을 주고
어떤 출력을 얻을지

누가 결정할까요?

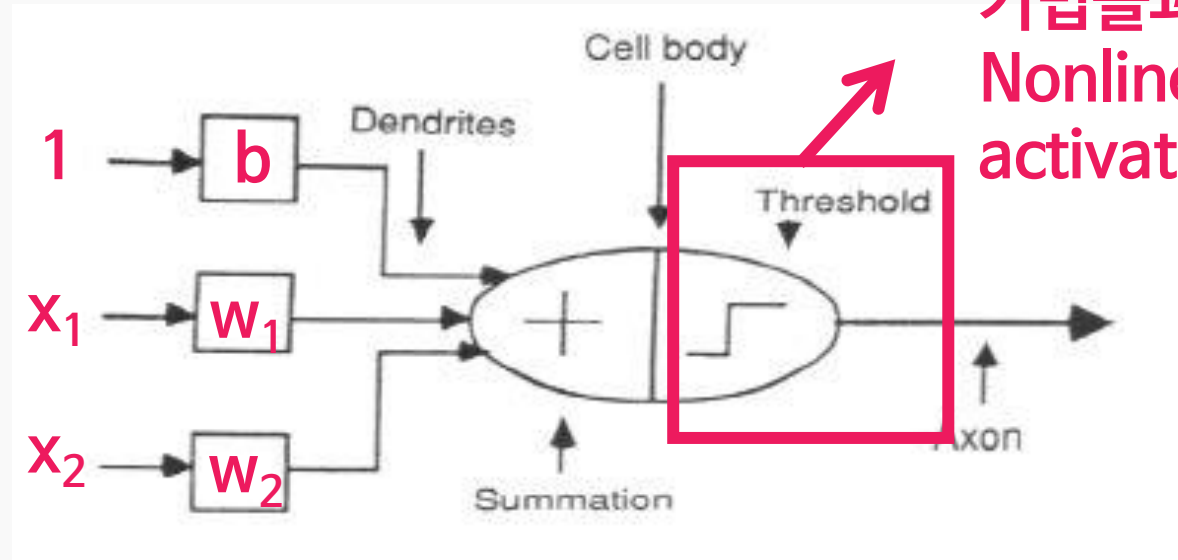
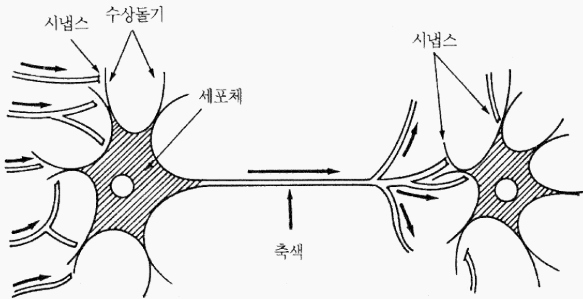
끝지말자 딥러닝



- 인공지능 (Artificial Intelligence)
- 머신러닝 (Machine Learning)
- 딥러닝 (Deep Learning)
- **Convolutional Neural Network**
- Recurrent Neural Network
- Applications

CNN에 들어가기에 앞서
가장 기본적인 Neural Network 부터
공부해봅시다.

인공지능 함수를 가장 간단한 1차함수로 모델링해보자 → (vanilla) neural network



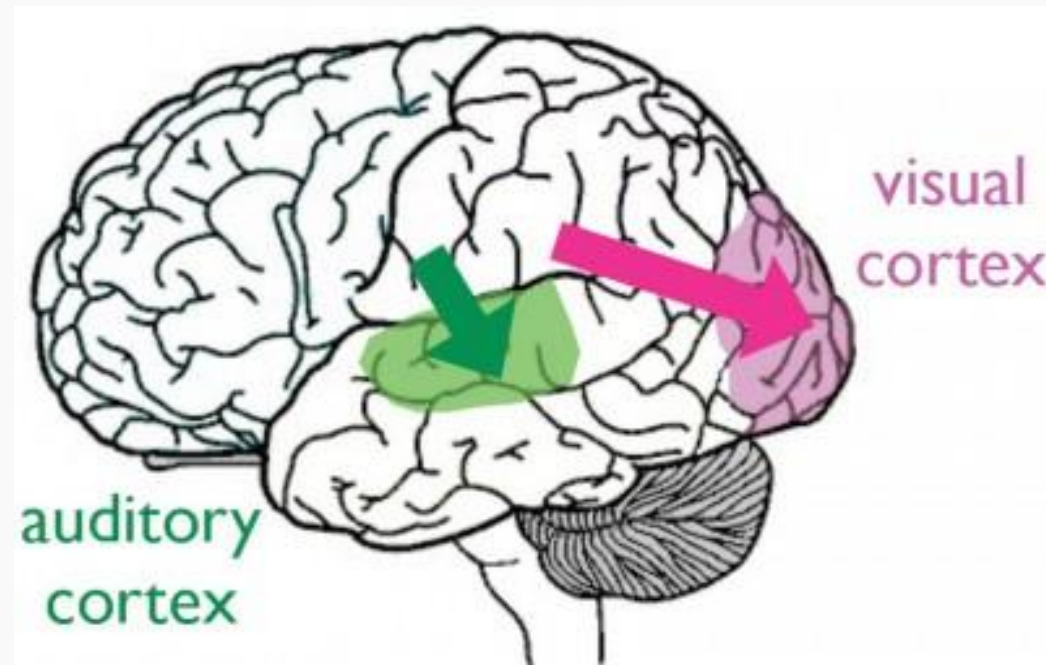
다른 머신러닝
기법들과의 차이점 1:
Nonlinear(복잡한)
activation function $g()$

$$\mathbf{w}^T \mathbf{x} + b$$

$$g(\mathbf{w}^T \mathbf{x} + b)$$

Neural Network를 영상처리에 특화시켜보자!

→ Convolutional Neural Network



Convolution (Operator) 이란?

$$2+3$$

$$9-7$$

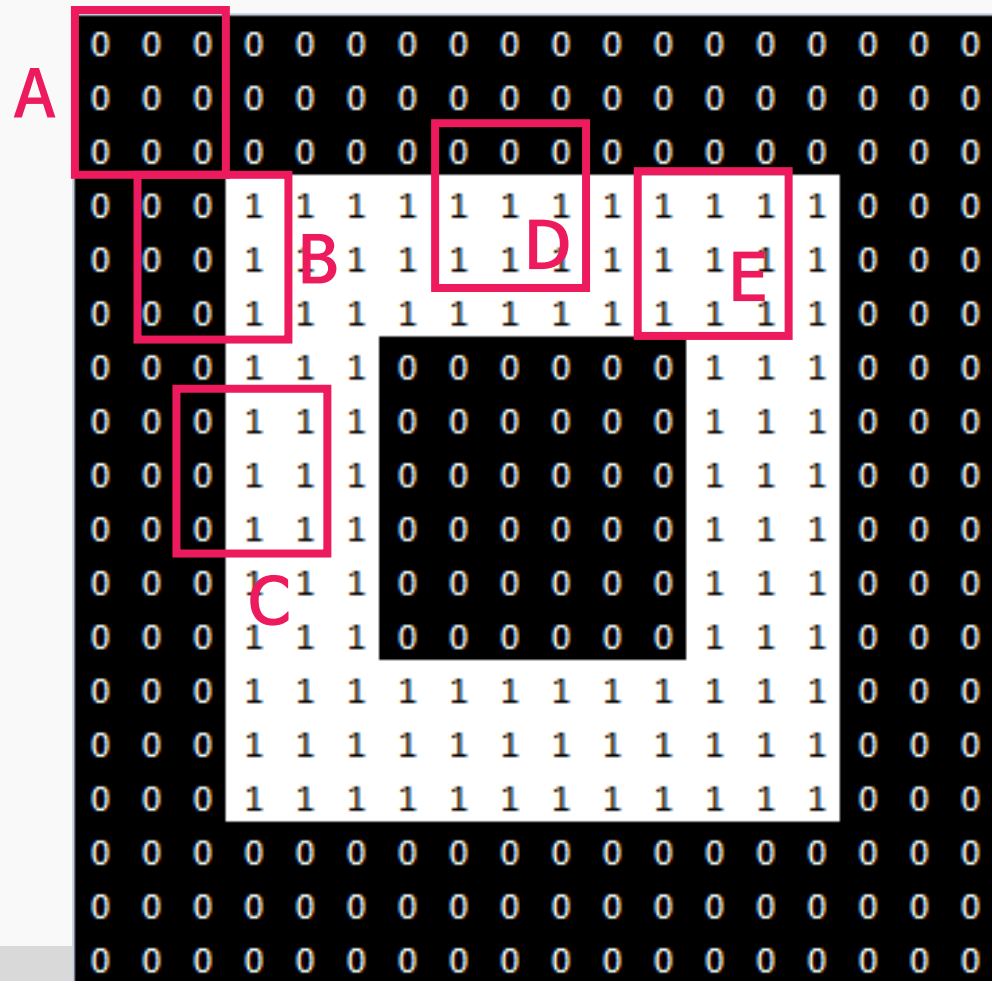
$$7 \times 8$$

$$10/2$$

$$[2 \ 3] * [1 \ 1]$$

영상에서의 Convolution

-1	0	1
-1	0	1
-1	0	1



- (1) A 영역의 Convolution 값은?
- (2) B 영역의 Convolution 값은?
- (3) C 영역의 Convolution 값은?
- (4) D 영역의 Convolution 값은?
- (5) E 영역의 Convolution 값은?
- (6) 전체 이미지의 Convolution 값은?
- (7) 이 filter의 역할은?

Convolution 을 통해
영상의 특징을 뽑아낼 수 있다.

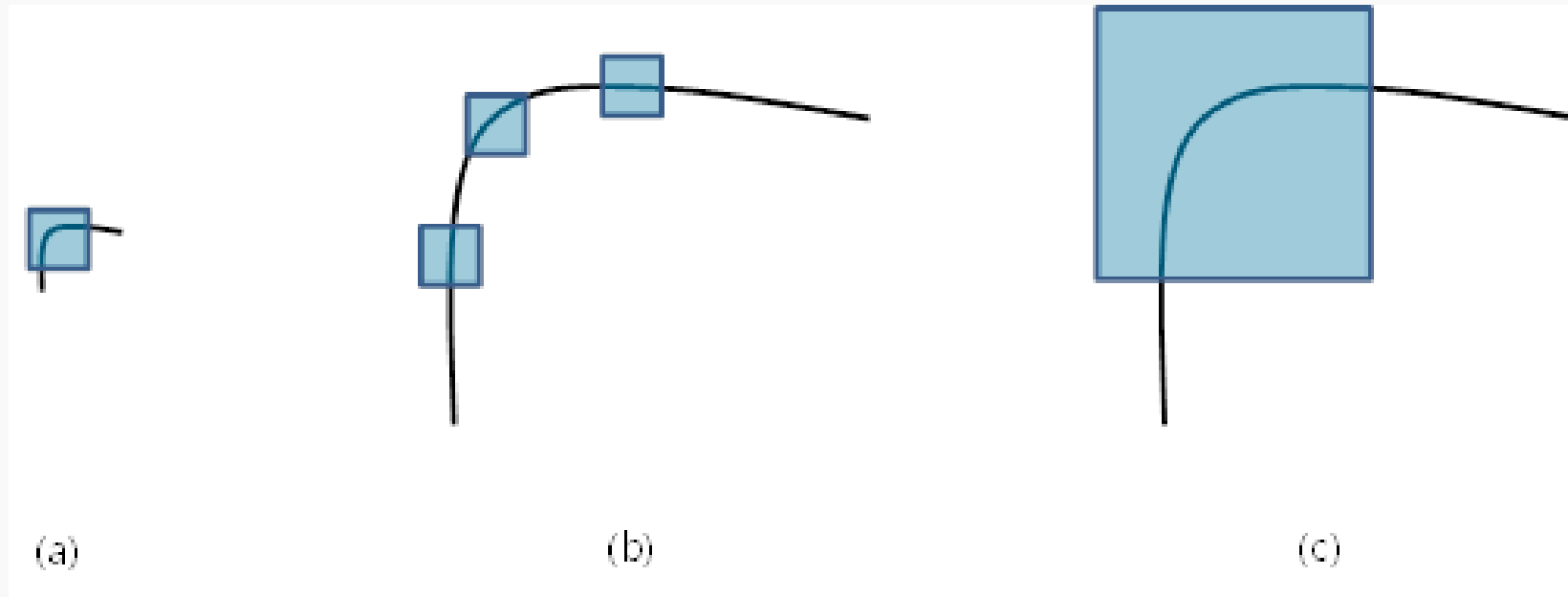
Convolution 연산자의 특징

$(f*g)*h=f*(g*h)$
결합법칙이 성립한다.

[Ex] f : 입력영상,
 g : R/G 통과 필터,
 h : R 통과필터

입력 영상 f 는 어떻게 변했을까?

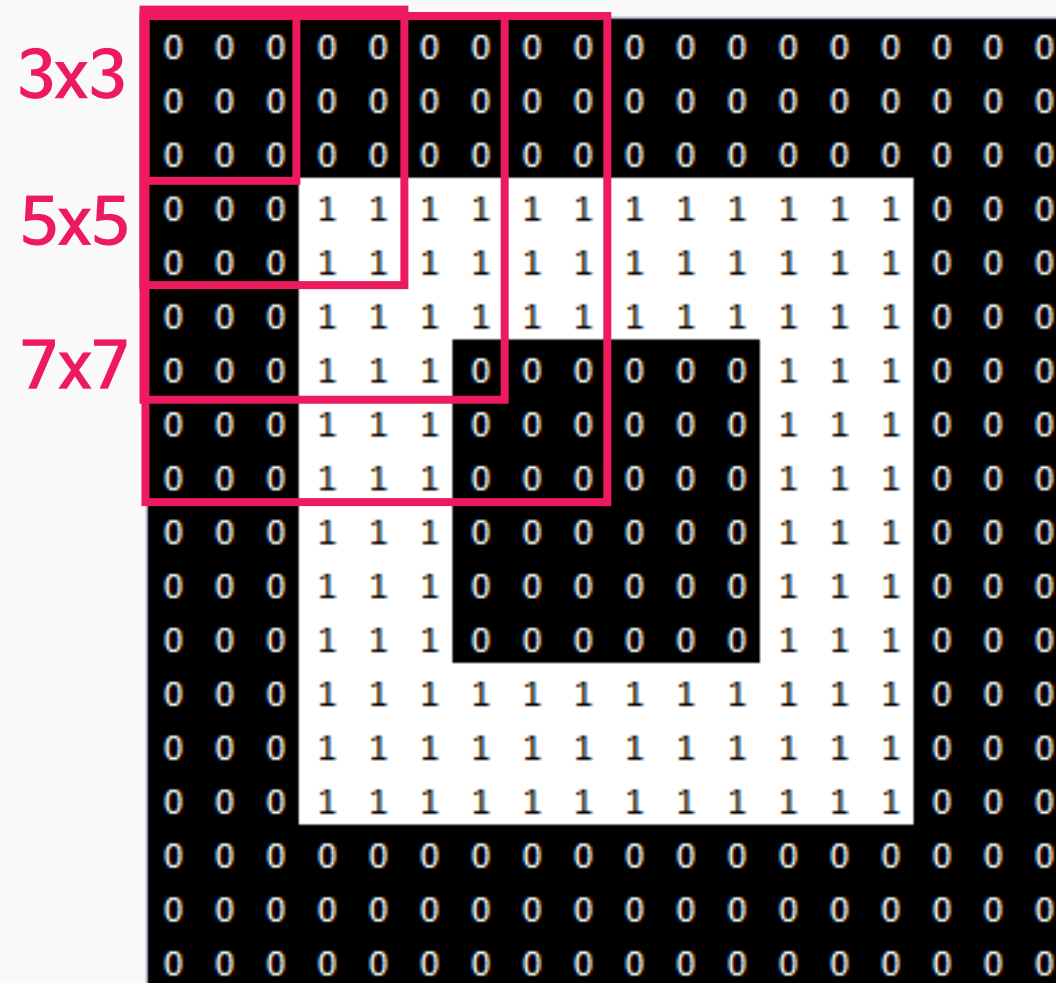
영상 분석의 기초 : 서로 다른 크기의 영상 feature가 필요하다



Corner Detection Example

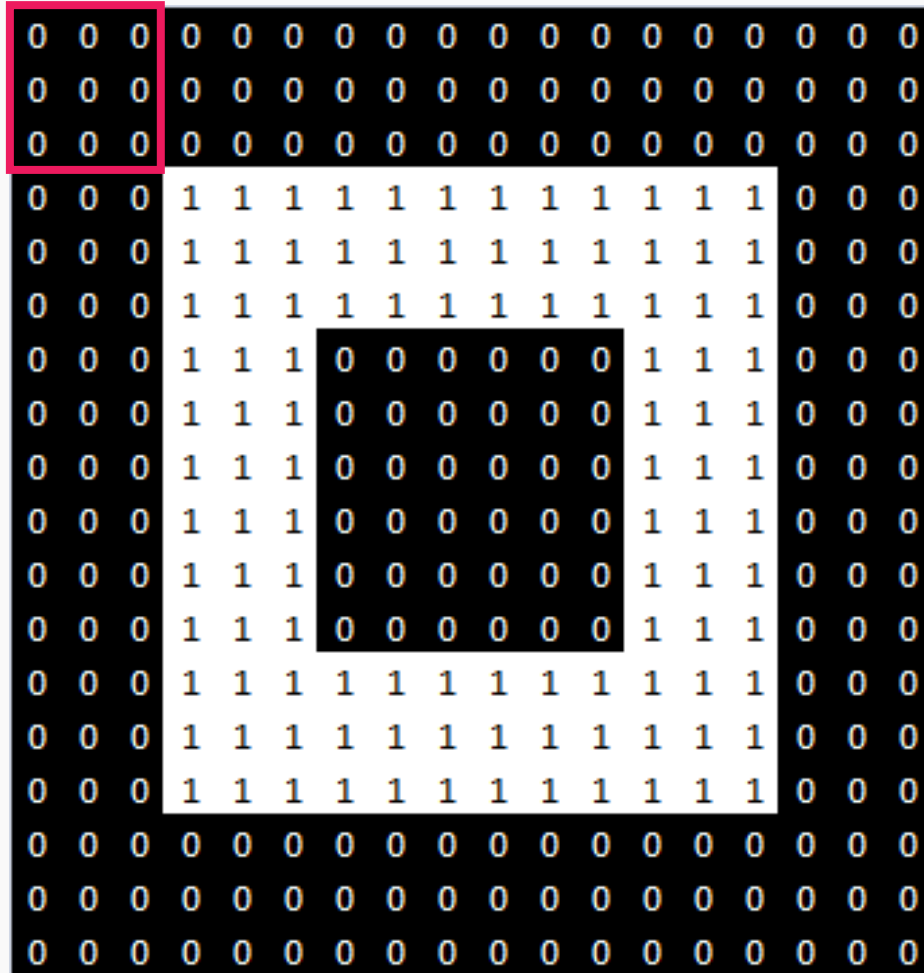
서로 다른 크기의 영상 feature를 만드는 두가지 방법!

1. 필터 사이즈를 점점 크게 만든다.

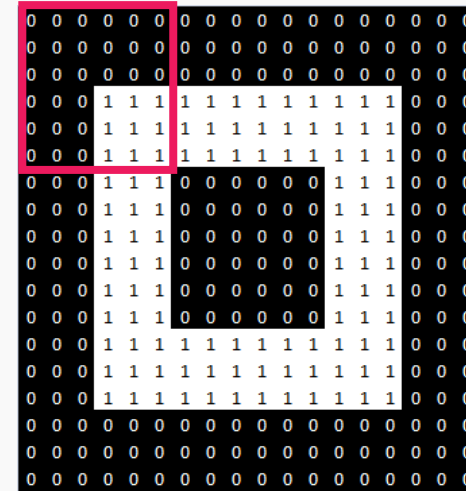


2. 영상 사이즈를 점점 작게 만든다.

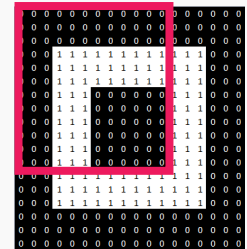
3x3



6x6



12x12



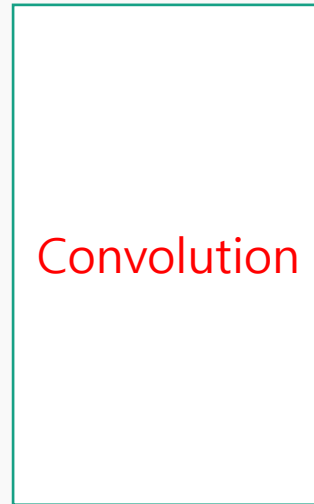
이 두가지를 모두 넣은 게 Convolutional Neural Network!

1. 필터 사이즈를 점점 크게 만든다.

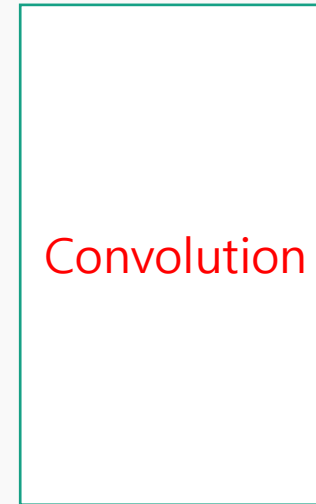
입력 영상 (f)



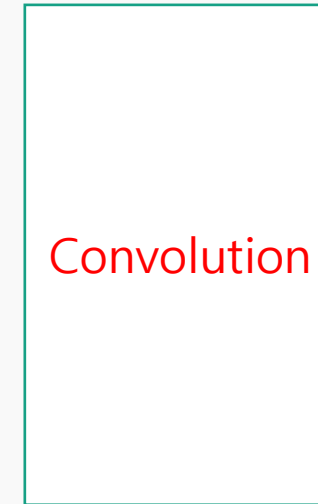
3x3 (g)



3x3 (h)



3x3



← 5x5 (g*h) →

← 7x7 →

$$(f*g)*h=f*(g*h)$$

2. 영상 사이즈를 점점 작게 만든다.

입력 영상



Convolution

Max
Pooling

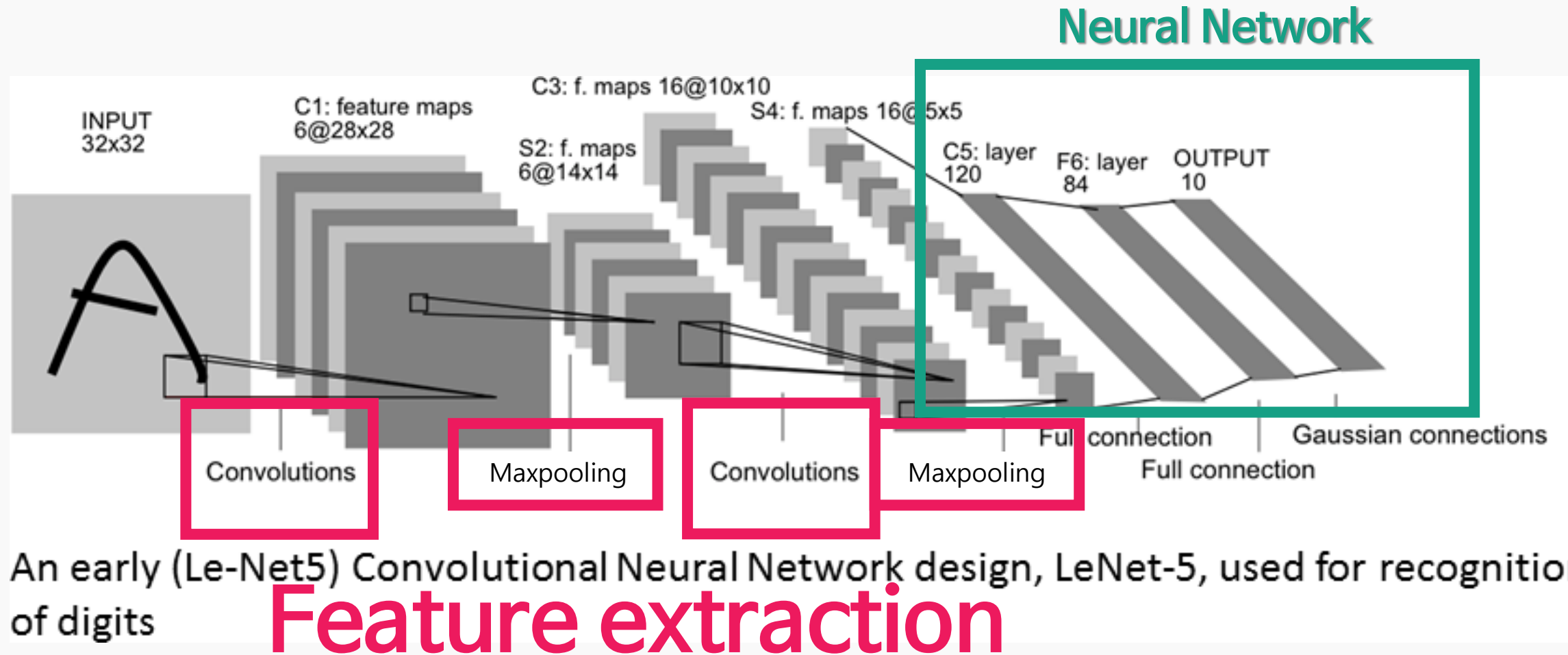
Convolution

Max
Pooling

Convolution



Convolutional Neural Network (CNN)



진짜로 이것만 알면 딥러닝 연구가 가능한가요?

적어도
시작은 가능합니다.

이 정도 지식으로 Artistic Style 논문을 읽을 수 있을까요?



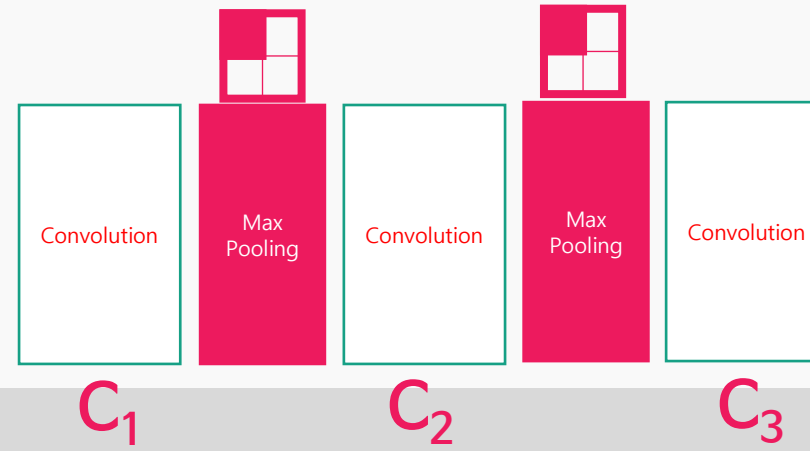
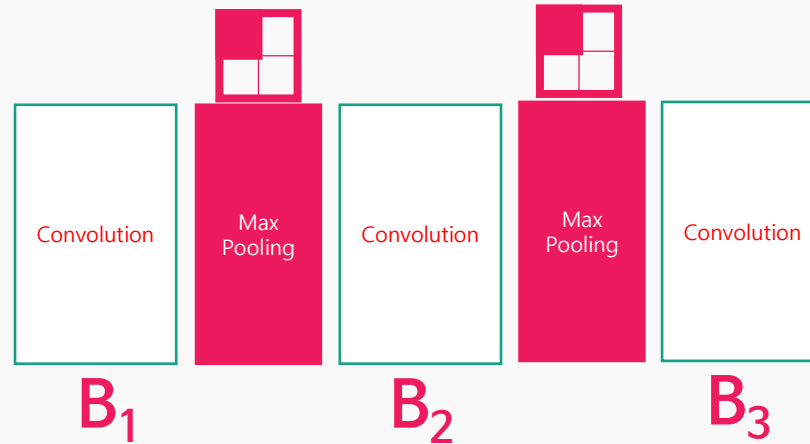
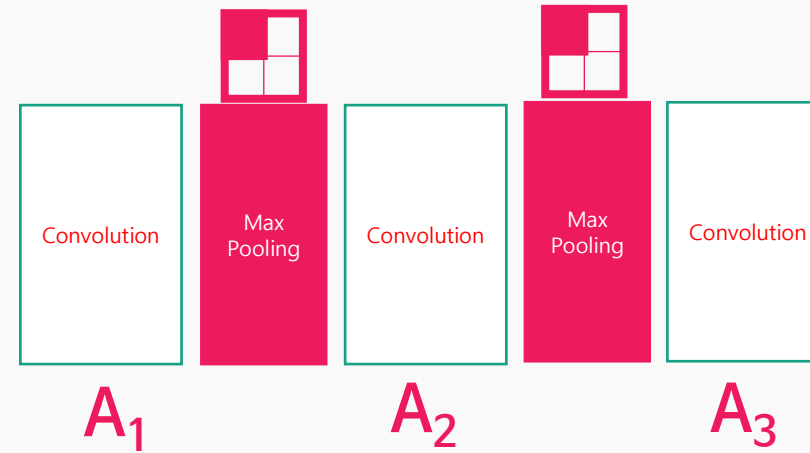
만드는 영상



사진 영상



Art 영상



$$A_1 = B_1$$

$$h(A_1) = h(C_1)$$

하도록 영상을
만들어낸다.

이 정도 지식으로
새로운 알고리즘을 개발할 수 있을까요?

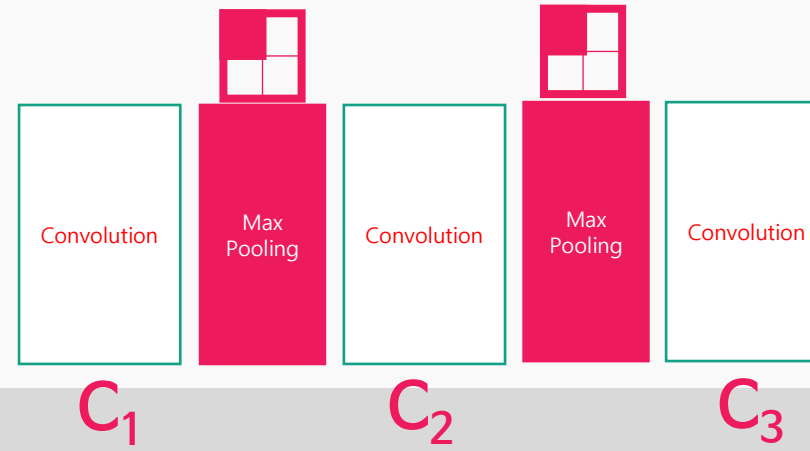
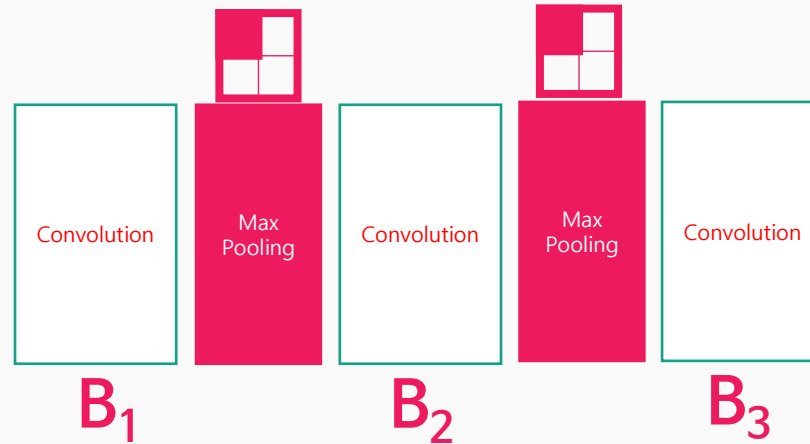
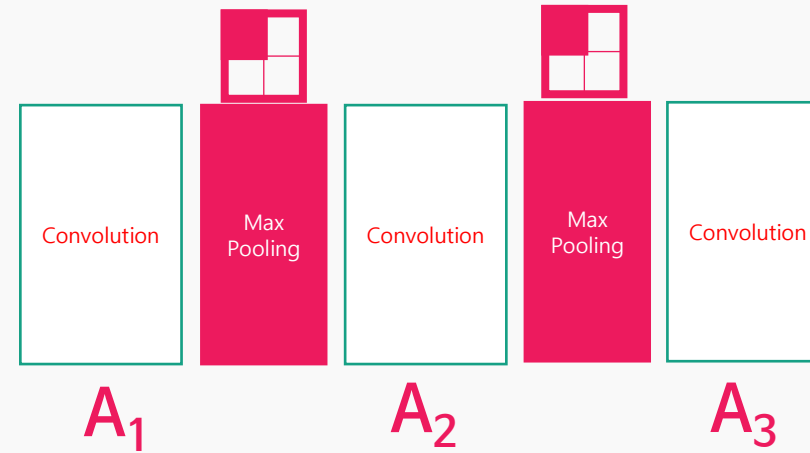
만드는 영상



사진 영상



Art 영상



$$A_1 = B_1$$

$$h(A_1) = h(C_1)$$

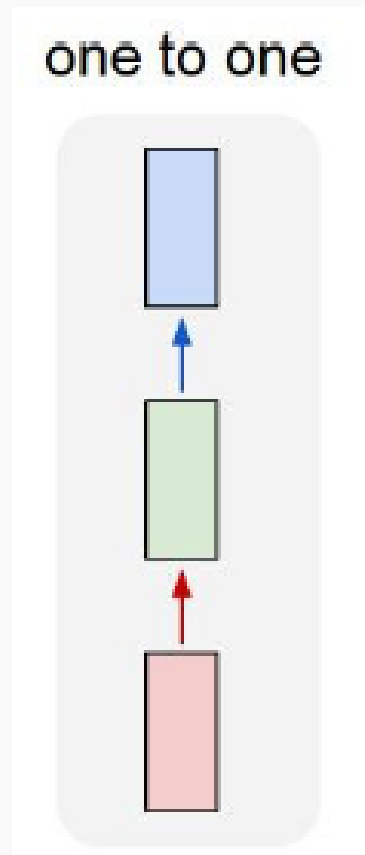
새로운 함수 $h'()$
를 만들어낸다면?

끝지말자 딥러닝



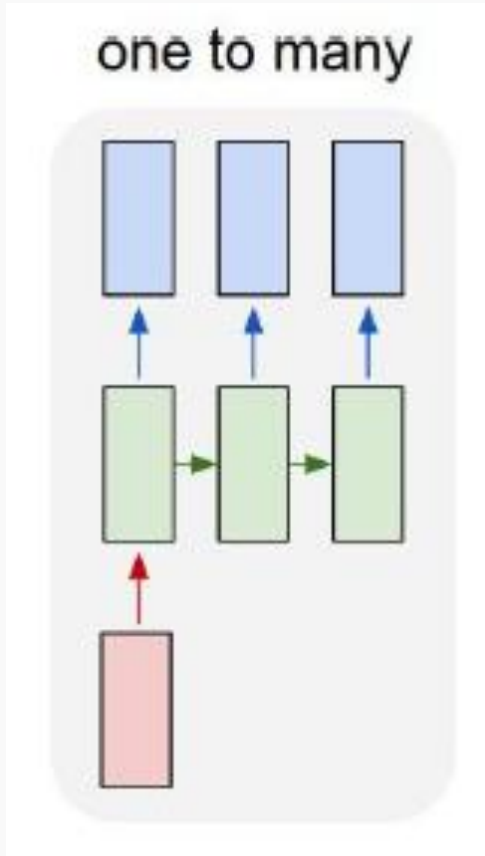
- 인공지능 (Artificial Intelligence)
- 머신러닝 (Machine Learning)
- 딥러닝 (Deep Learning)
- Convolutional Neural Network
- **Recurrent Neural Network**
- Applications

입력 또는 출력에 시간 순서가 있다면?
→ Recurrent Neural Network



(Vanilla)
Neural Network

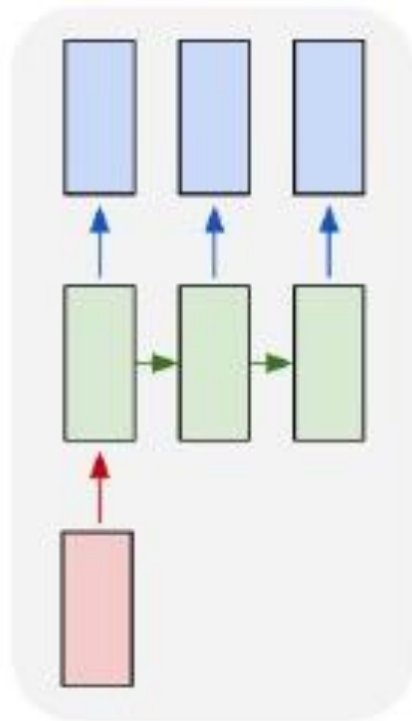
입력이 들어가면
출력이 하나 나온다.



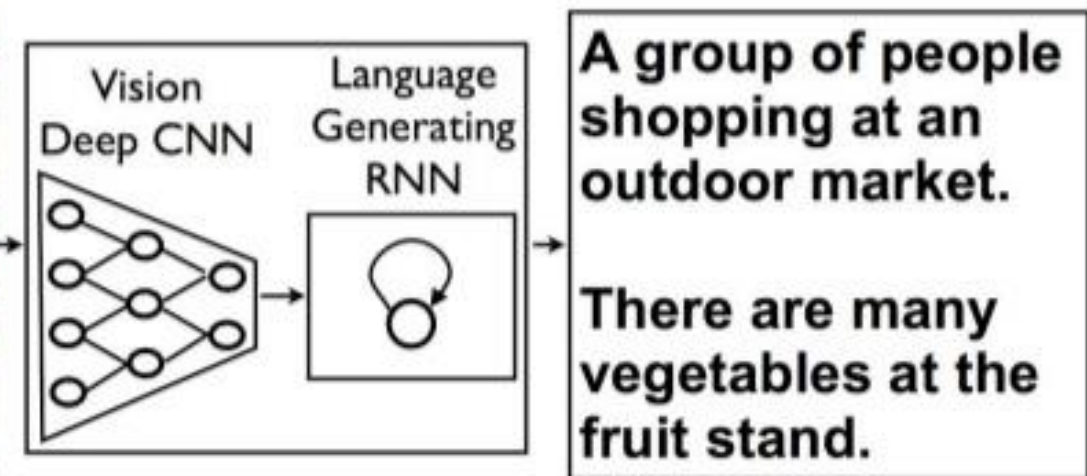
입력이 들어가면
출력이 순차적으로 나온다.

Image Caption Generation

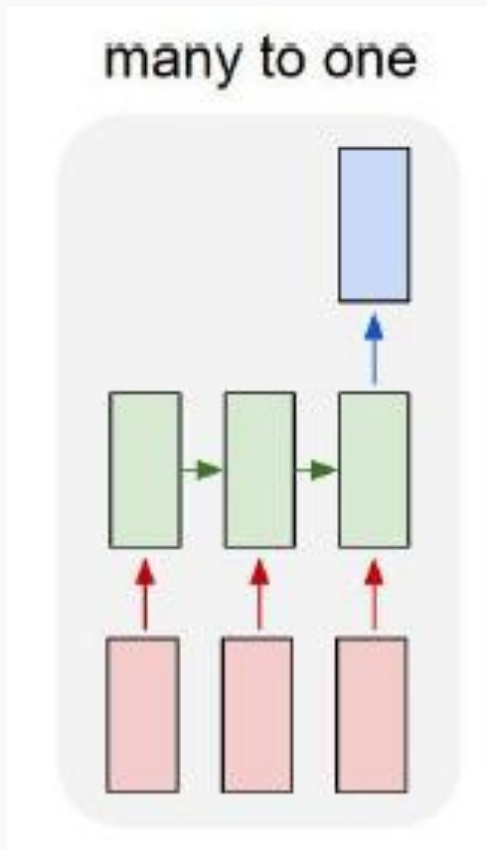
one to many



출력에
시간순서가 있다면?

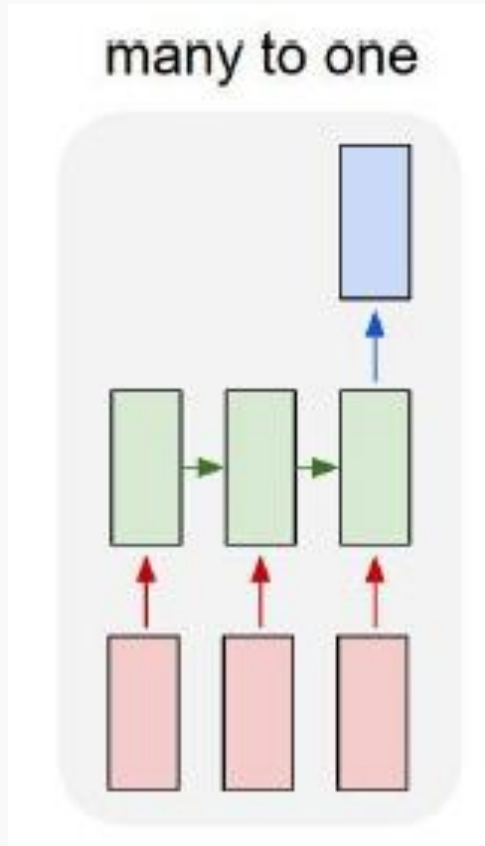


<http://arxiv.org/abs/1411.4555> "Show and Tell: A Neural Image Caption Generator"



입력이 순차적으로 들어가면
출력이 나온다.

Sentiment Classification

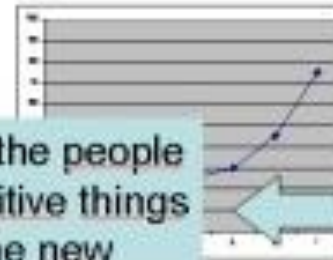


Input of quotes:

"I hate the small screen of my PDA"
"I love my mobile phone"

Sentiment Classification
Using a machine learning technique

75% of the people say positive things about the new mobile phone



Output:

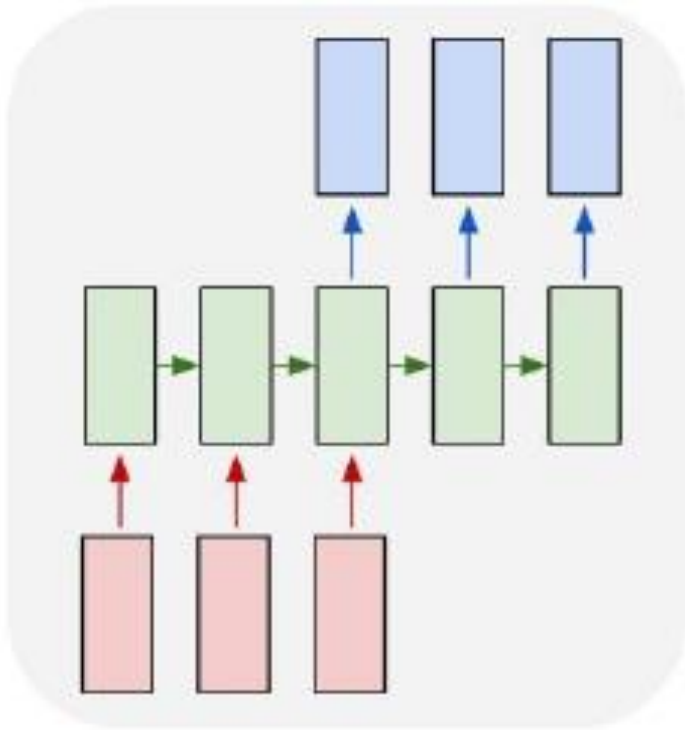
Positive

"I love my mobile phone"

Negative

"I hate the small screen of my PDA"

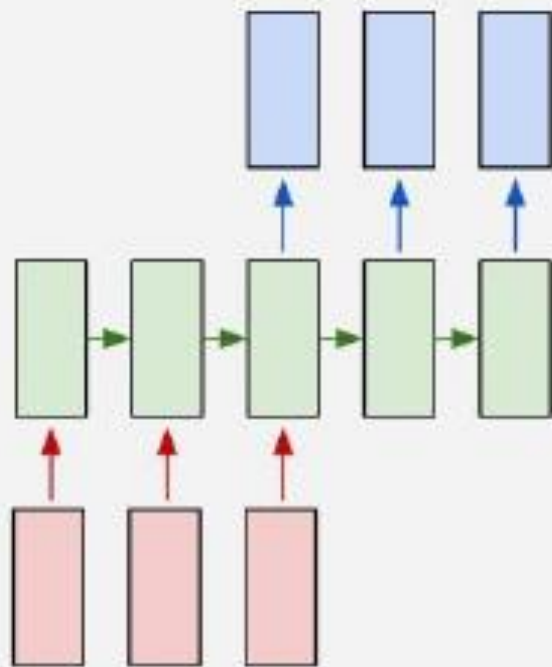
many to many



입력이 순차적으로 들어가면
출력이 순차적으로 나온다.

Machine Translation

many to many



Economic growth has slowed down in recent years .

Das Wirtschaftswachstum hat sich in den letzten Jahren verlangsamt .

Economic growth has slowed down in recent years .

La croissance économique s' est ralentie ces dernières années .

입력 또는 출력에 시간 순서가 있다
→ Sequence

인간은 Sequence를 학습한다.

(마음속으로) 따라 불러 보세요.

손에 손 잡고

하늘 높이 솟는 불
우리들 가슴 고동치게 하네..

·
·
·

(노래를 이미 알고 계신 분들은)
쉽게 따라 부르실 수 있었을 겁니다.

그렇다면, 뒷 쪽에 나오는
이 부분을 불러보세요.

서로서로 사랑하는
한 마음 되자

갑자기 부르려니까 못 부르시겠죠?

인간은 Sequence를 학습한다.

그렇다면,
딥러닝으로 Sequence를 학습시키려면?

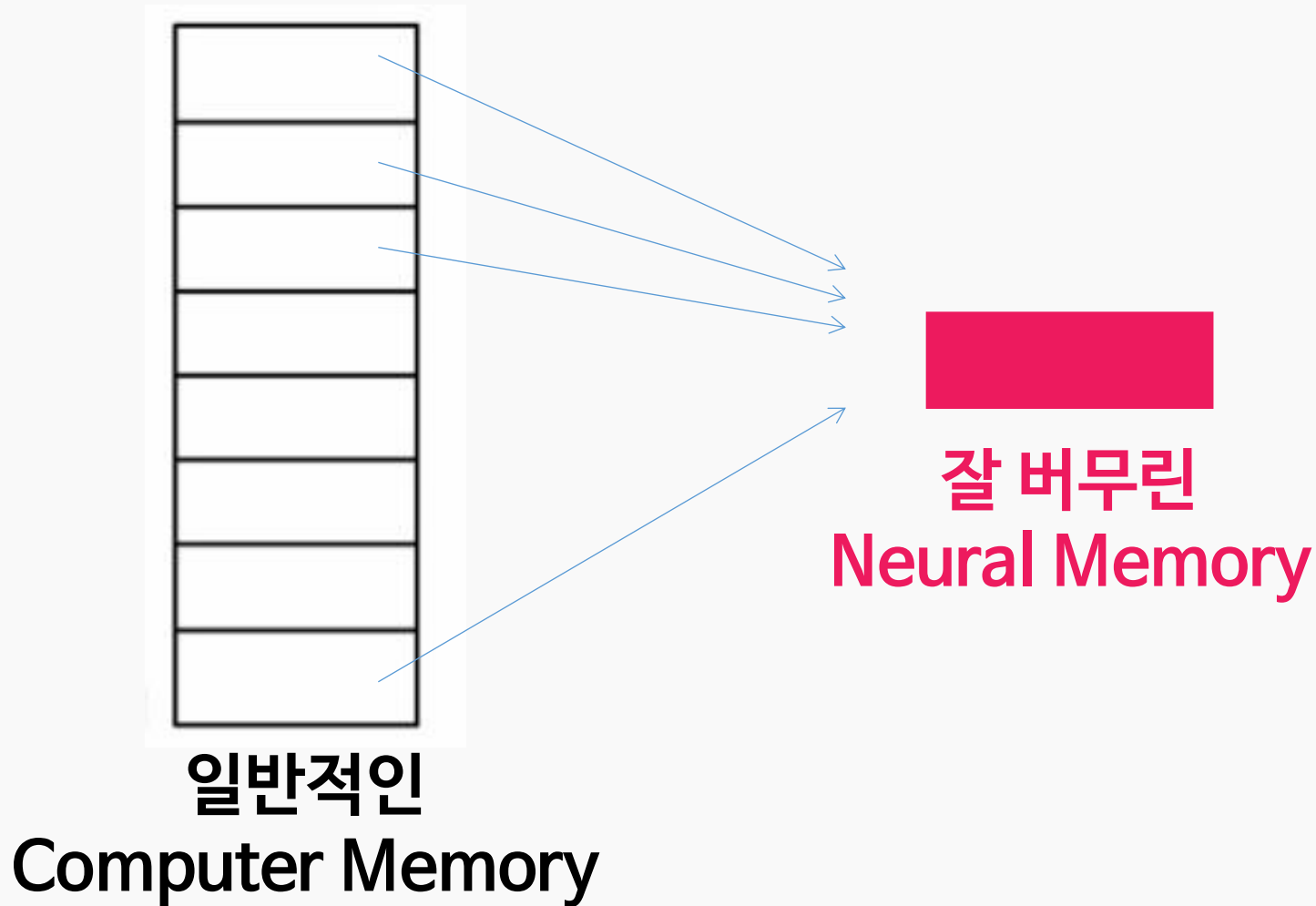
Sequence를 입력으로 다 넣어줄 것인가?

입력 사이즈
(및 Neural Network 사이즈)가
너무 커지게 된다.

입력 사이즈를 유지하면서,
과거의 입력 값들을 반영하는 방법은 없을까?

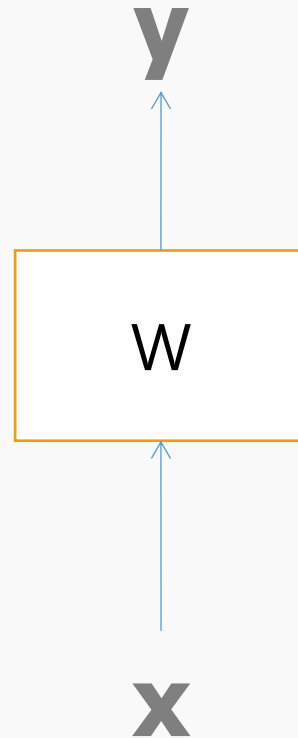
Recurrent Neural Network

과거의 입력 값을
잘 “버무려” 저장해 놓을 수 있는
Memory를 만들어 보자

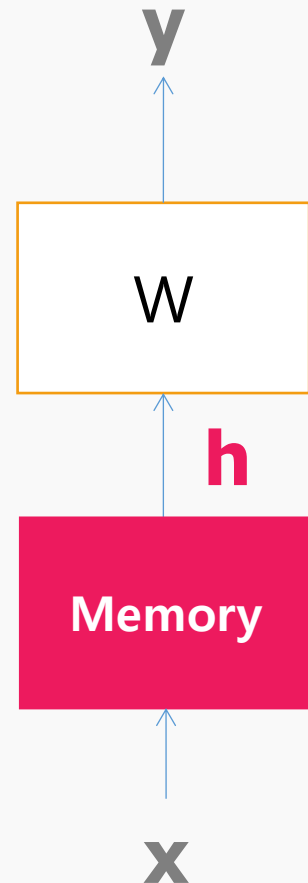


Sequence 를 모두 입력으로 주지 말고,
잘 버무린 Neural Memory만 입력으로 주자.

(Vanilla) Neural Network의 구조



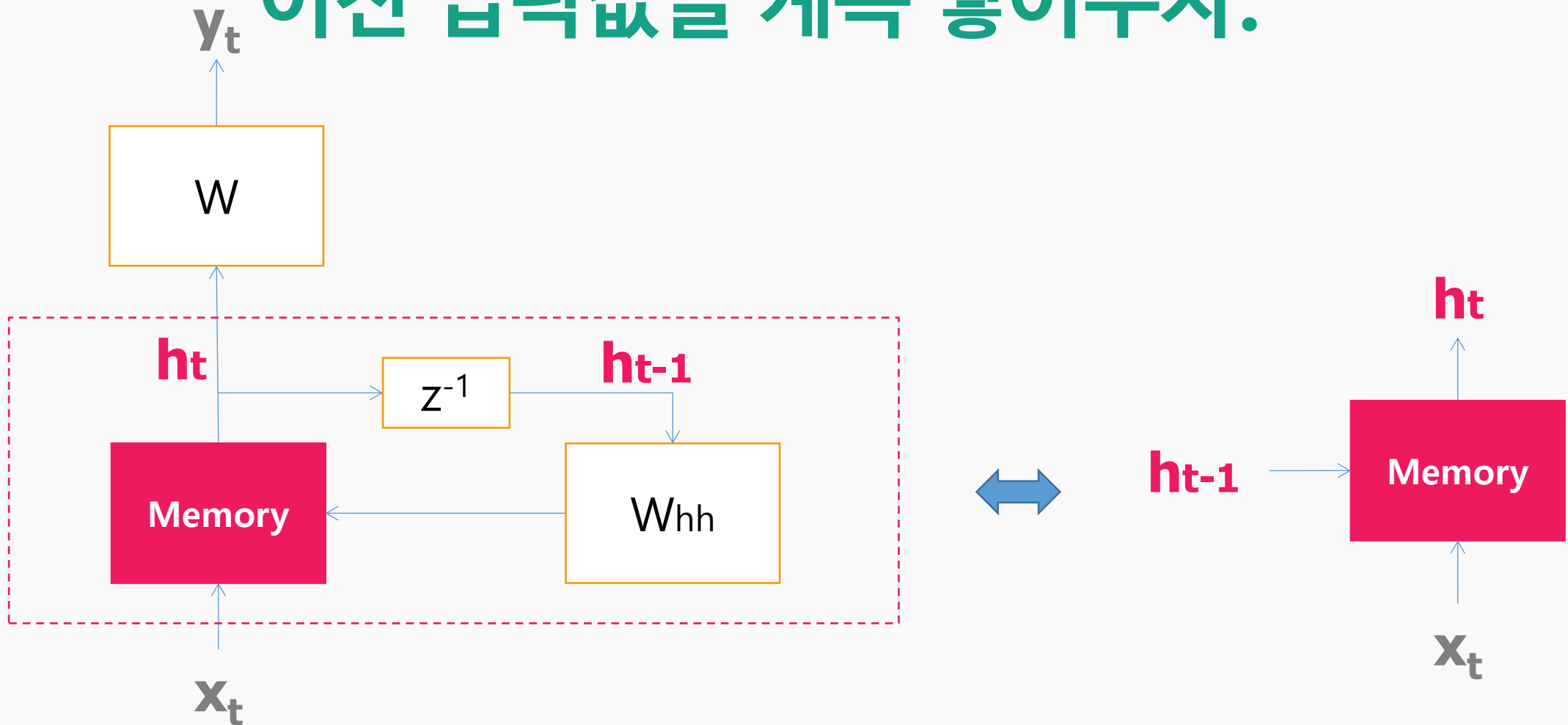
Neural Network Input from Memory



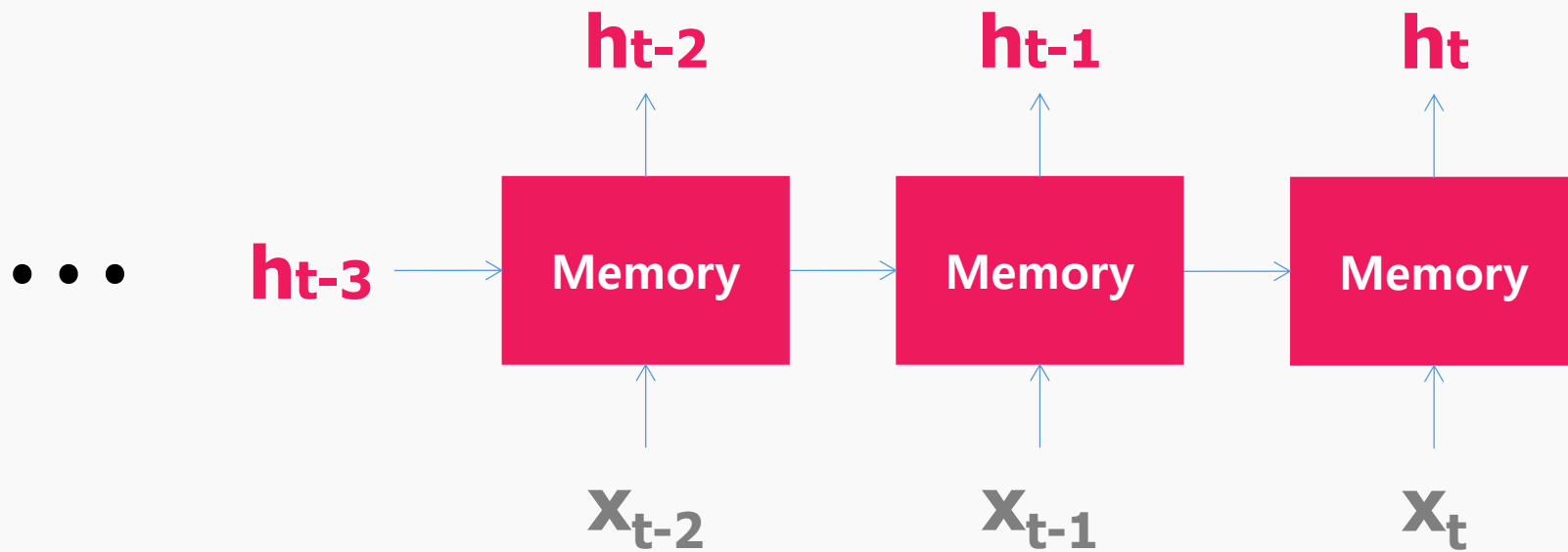
잘 버무려진 입력 값인 h 를
네트워크의 입력으로 넣어주자.

잘 버무리는 방법

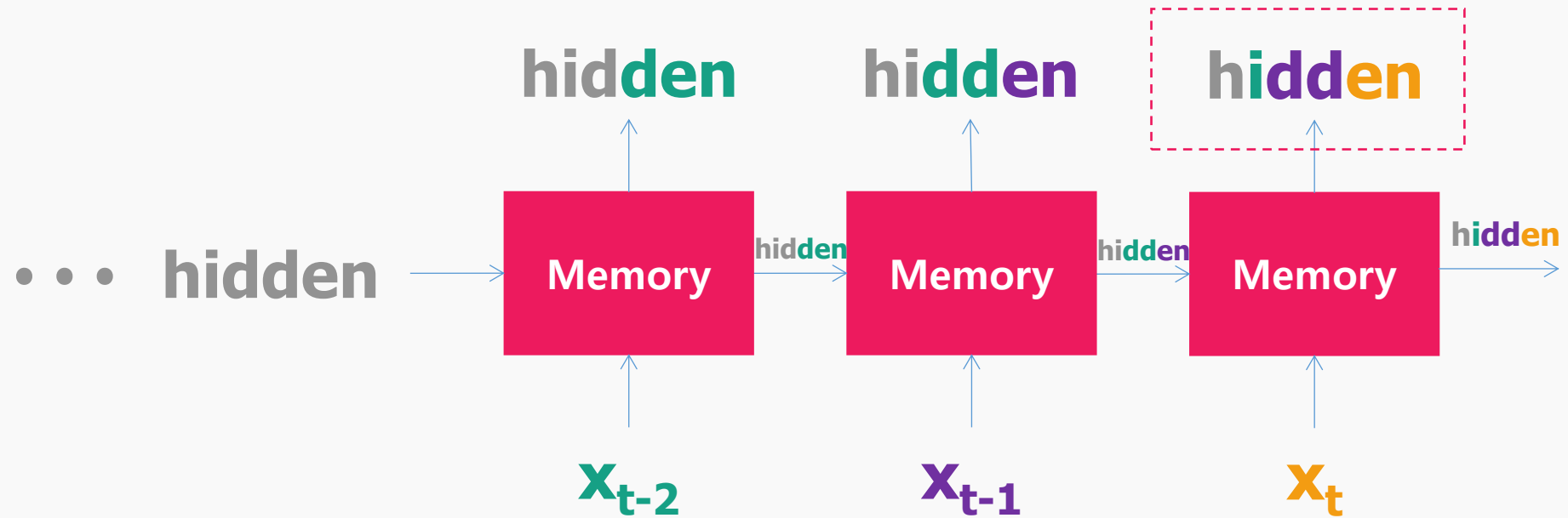
이전 입력값을 계속 넣어주자.



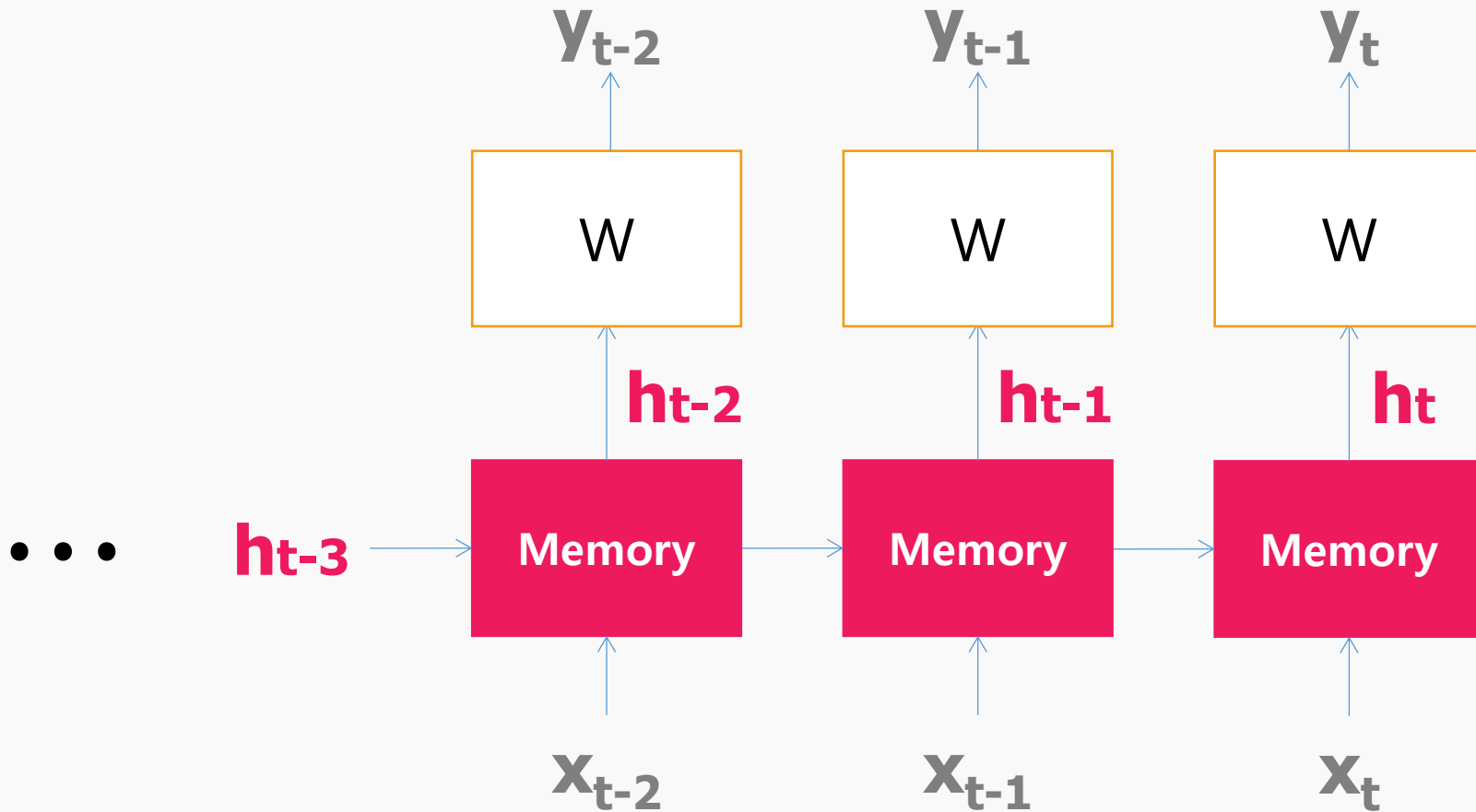
잘 버무리는 방법



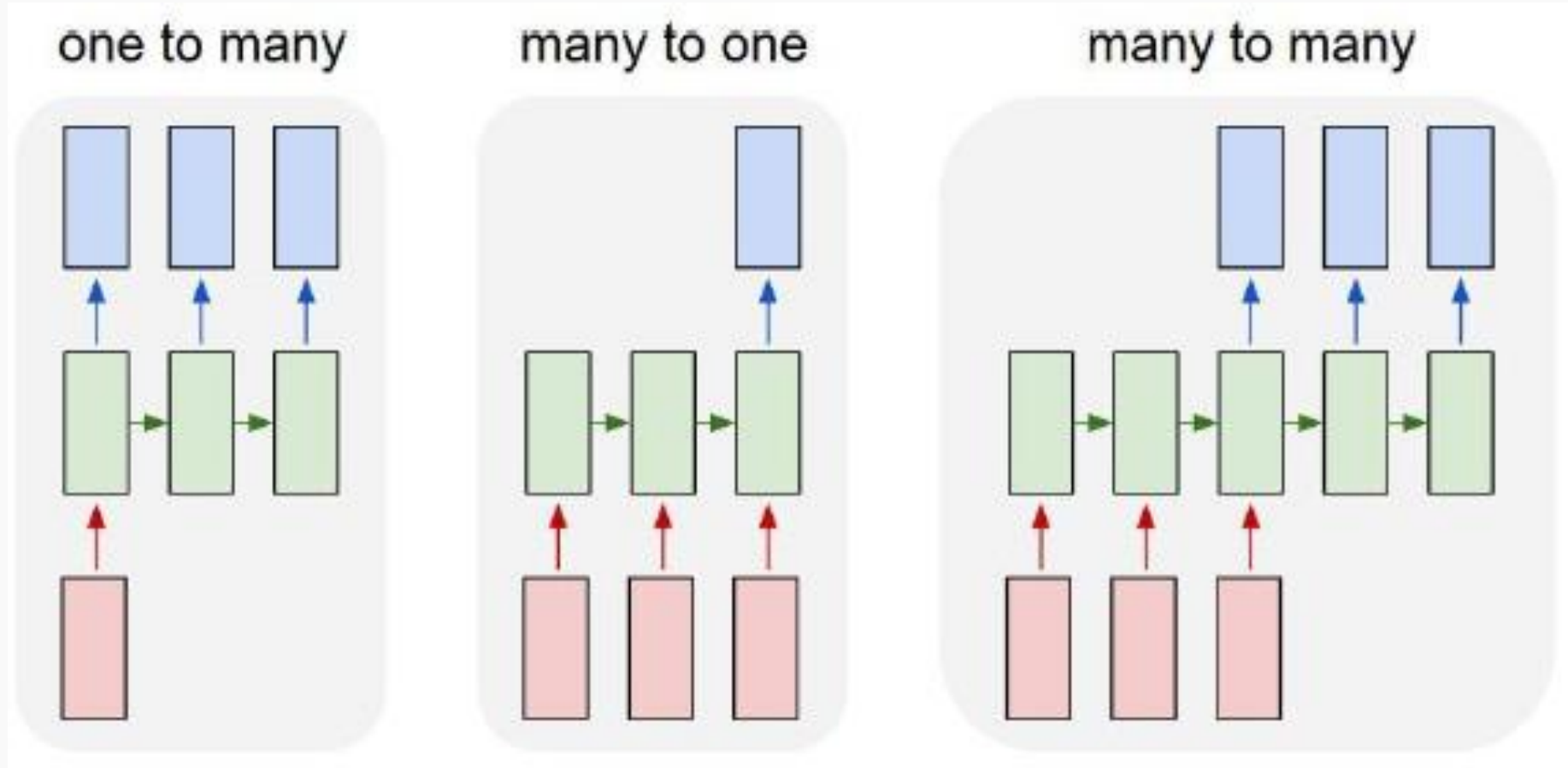
버무려진 후의 메모리



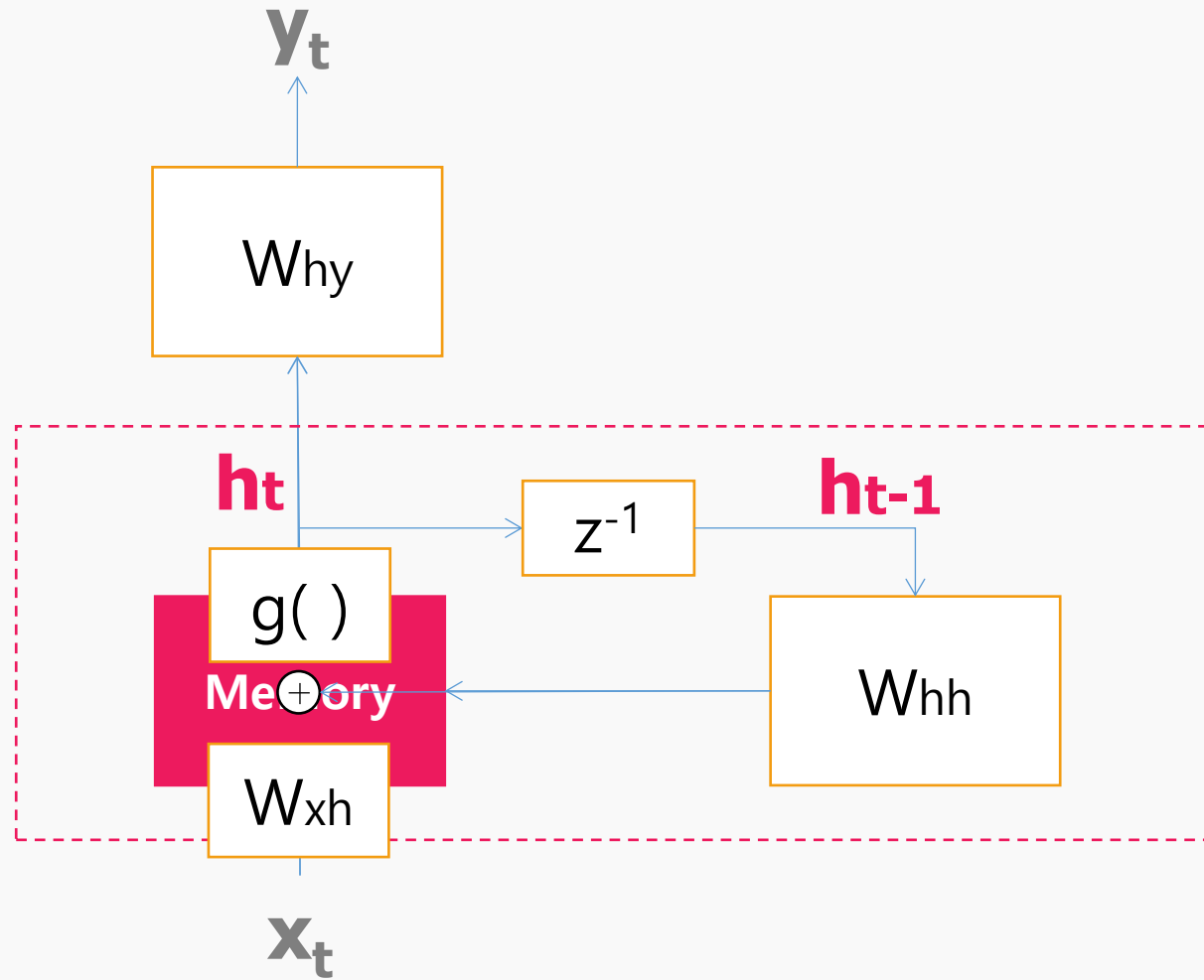
잘 버무린 후에 출력을 내보내면



RNN 응용 구조는 이렇게 만들어진 것들..



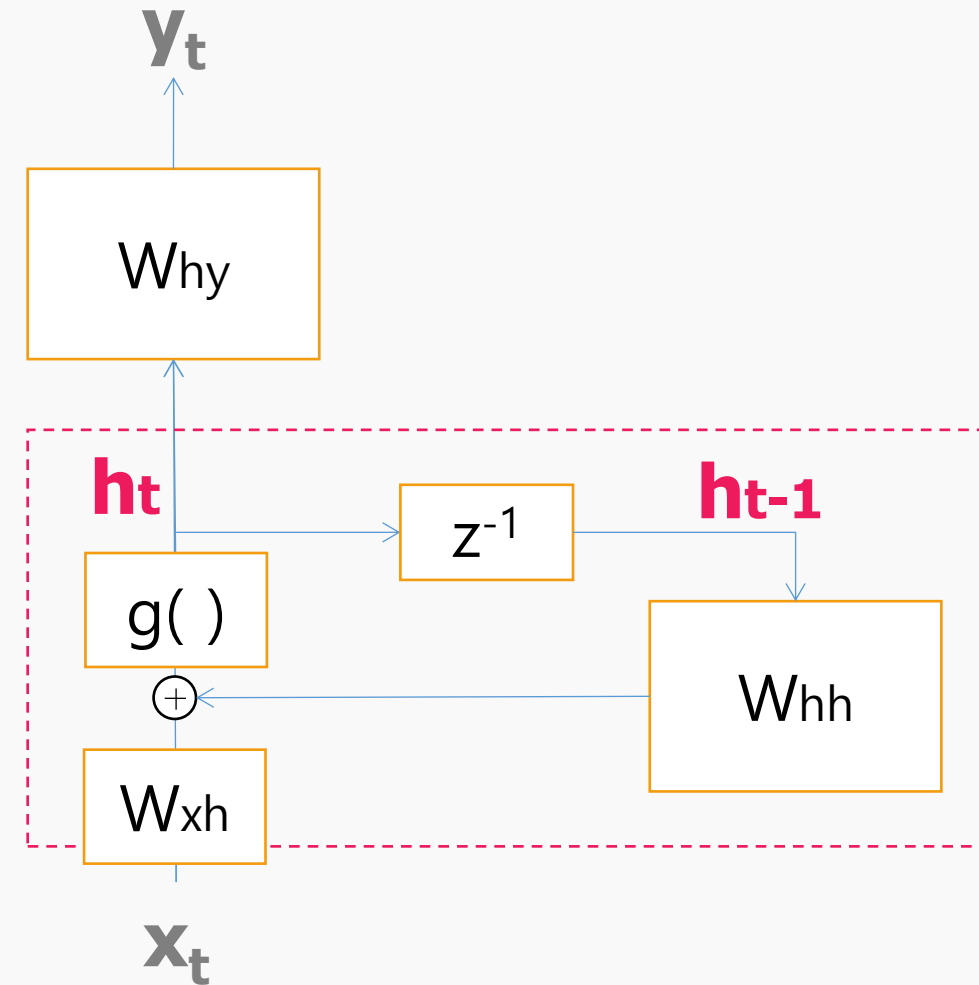
개념적으로 배운 RNN을
수식적으로 정리해 보겠습니다.




RNN

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

$$y_t = W_{hy}h_t$$



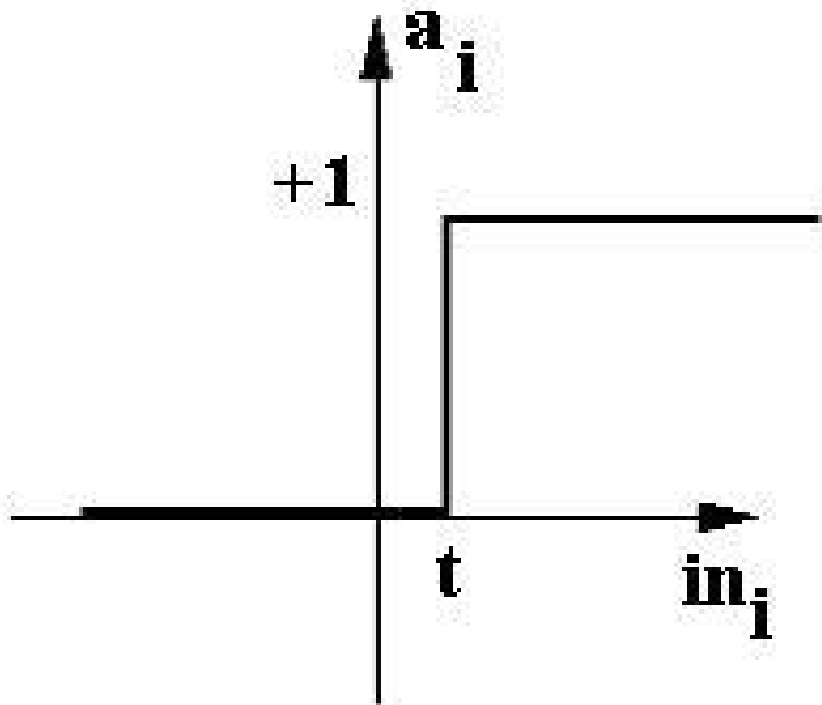
Vanishing Gradient 복습



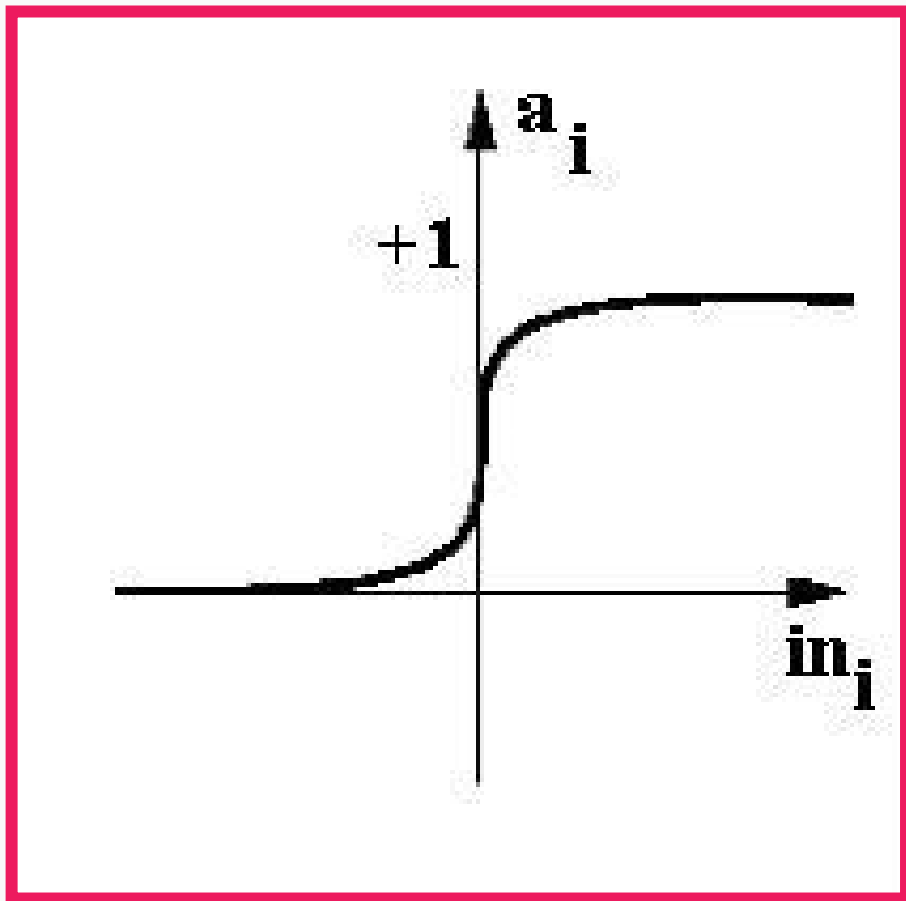
더 이상의
자세한 설명은
생략한다.

(Deep) Learning을 하기 위해서는
여자저차해서 **미분**을 이용합니다.

학습되는 양 = 미분값 * 출력값



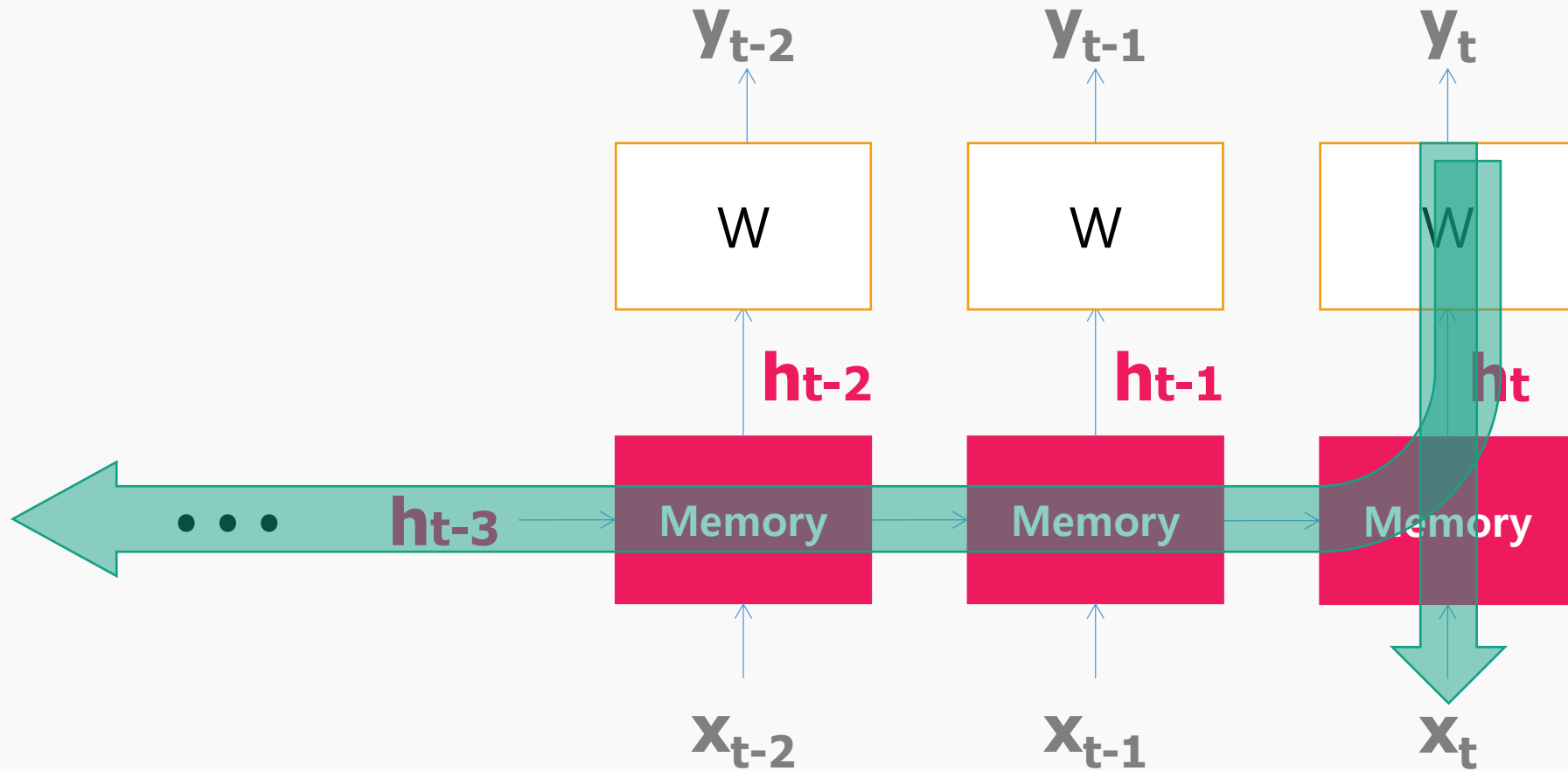
- 미분을 하면 0이 나와서, Learning이 불가함



- (1) 대부분의 영역에서 미분값이 0에 가깝다
- (2) 출력의 최대값이 1이다.
출력값이 0.5라고 했을때, 4개의 레이어만 지나도 1/16

→ Vanishing Gradient

Vanishing Gradient



평균을 구해보자

N개의 sample에 대한
평균 c_N 을 구해보자.

$$\begin{aligned} c_N &= \frac{1}{N} (x_1 + x_2 + x_3 + \cdots + x_N) \\ &= \frac{1}{N} \sum_{i=1}^N x_i \end{aligned}$$

평균을 구하는 또 다른 방법

$$\begin{aligned}
 c_N &= \frac{1}{N} \sum_{i=1}^N x_i \\
 &= \frac{1}{N} \left(\sum_{i=1}^{N-1} x_i + x_N \right) \\
 &= \frac{N-1}{N} \frac{1}{N-1} \sum_{i=1}^{N-1} x_i + \frac{1}{N} x_N \\
 &= \alpha c_{N-1} + (1 - \alpha) x_N \quad 0 < \alpha < 1
 \end{aligned}$$

평균 = 잘 버무렸다...

이 평균을 이용해서
잘 버무린 Memory를 만들어 보자.

RNN의 메모리 출력을 잘 버무린 평균(c_t) 를 구해보자.

$h_t \rightarrow c_t$

$$c_t = f * c_{t-1} + i * h_t$$

얼마만큼 forget 하고,
얼마만큼 input을 받아들일 것인가?

얼마만큼 forget할지,
얼마만큼 input을 받아들일지는
또 다른 Neural Network로 학습시키자.

그래서 만들어진 네트워크가 Input Gate와 Forget Gate

$$i = \text{sigmoid}(W_{hi}h_{t-1} + W_{xi}x_t)$$
$$f = \text{sigmoid}(W_{hf}h_{t-1} + W_{xf}x_t)$$

Q. Input Gate와 Forget Gate에서는
왜 안좋은 sigmoid activation function을 사용할까?

메모리 출력은

$$\cancel{h_{t,LSTM} = c_t}$$

$$(c_t = f * c_{t-1} + i * h_t)$$

LSTM에서는
메모리에서 꺼낼 때
한번 더 네트워크를 거칩니다.

Output Gate

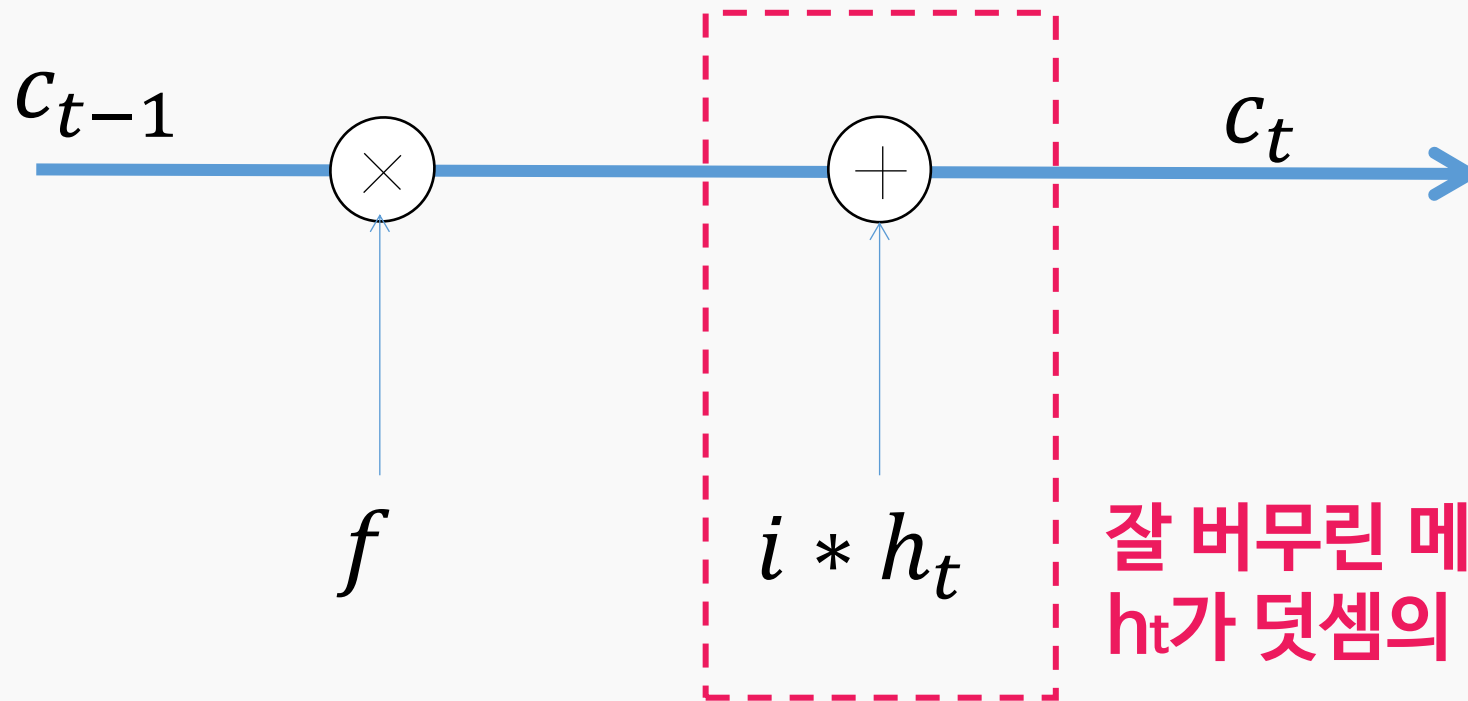
$$o = \text{sigmoid}(W_{ho}h_{t-1} + W_{xo}x_t)$$

최종 메모리 출력은

$$h_{t,LSTM} = o * g(c_t)$$

LSTM에서는 Vanishing Gradient 문제가 해결되나요?

$$c_t = f * c_{t-1} + i * h_t$$



잘 버무린 메모리 c_t 를 만들기 위해,
 h_t 가 덧셈의 형태로 영향을 미친다.

끝지말자 딥러닝



- 인공지능 (Artificial Intelligence)
- 머신러닝 (Machine Learning)
- 딥러닝 (Deep Learning)
- Convolutional Neural Network
- Recurrent Neural Network
- Applications

Artistic Style

2015.8



Artistic Style for Video

Artistic style transfer for video

Manuel
Alex

2016.4

University of Freiburg
Recognition and Image Processing

Prisma



Image Translation

2017.4

AI Fake Porn



네이버웹툰 ‘하일권-마주쳤다’



2018.01



진짜 사진을 찾아라!

DCGAN

2015.11



Image Generation



draw! shuffle! reset!

noise_strength



long_hair

short_hair

very_long_hair

side_ponytail

twintails



braid

ahoge

brown_hair

silver_hair

white_hair



red_hair

orange_hair

yellow_hair

green_hair

aqua_hair

blue_hair



purple_hair

pink_hair

black_eyes

brown_eyes

red_eyes

yellow_eyes



2015.12

<https://github.io/chainer-DCGAN/>

진짜 사진을 찾아라! BEGAN



PGGAN (128 → 1024)

PROGRESSIVE GROWING GAN WITH IMPROVED
QUALITY AND VARIATION

2017.10

Submitted to ICLR 2018

Image Coloring

2017.01



<http://paintschainer.preferred.tech/>

Image Coloring

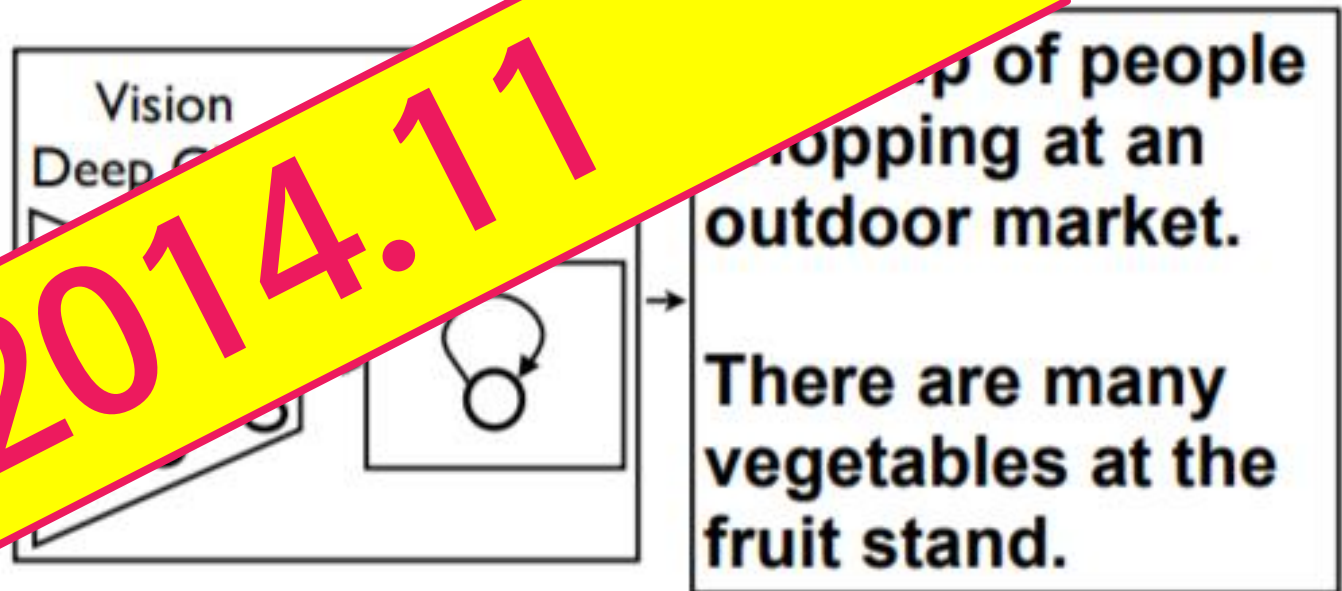


cGAN-based Manga Colorization Using a Single Training Image, <https://arxiv.org/pdf/1706.06918.pdf>

Image Captioning



2014.11



<https://arxiv.org/pdf/1411.4555.pdf>

Image Question and Answering



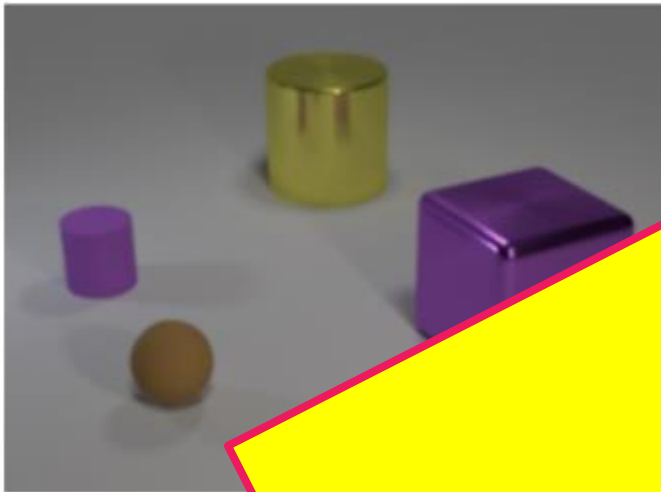
2015.05

Question

<https://github.com/seankim902/imageQA>

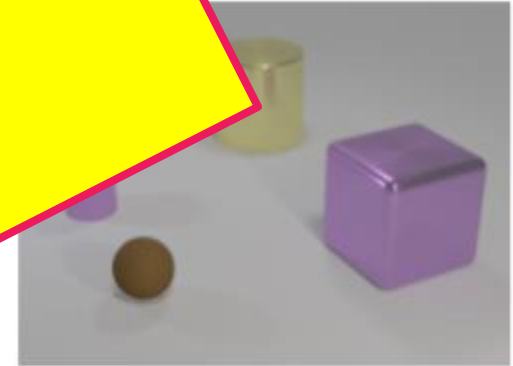
Relational Reasoning

Original Image:



Non-relational question:

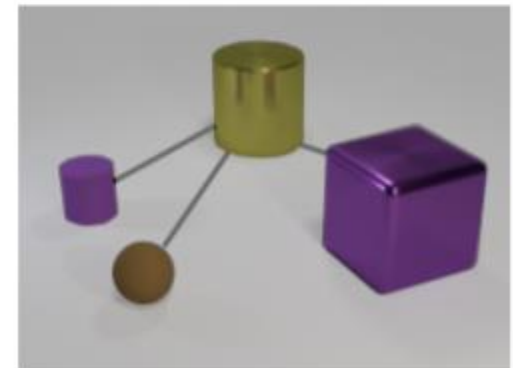
What is the color of the brown sphere?



2017.06

Relational question:

Are there any rubber things that have the same size as the yellow metallic cylinder?



A simple neural network module for relational reasoning <https://arxiv.org/pdf/1706.01427.pdf>

Deep Reinforcement Learning : Game

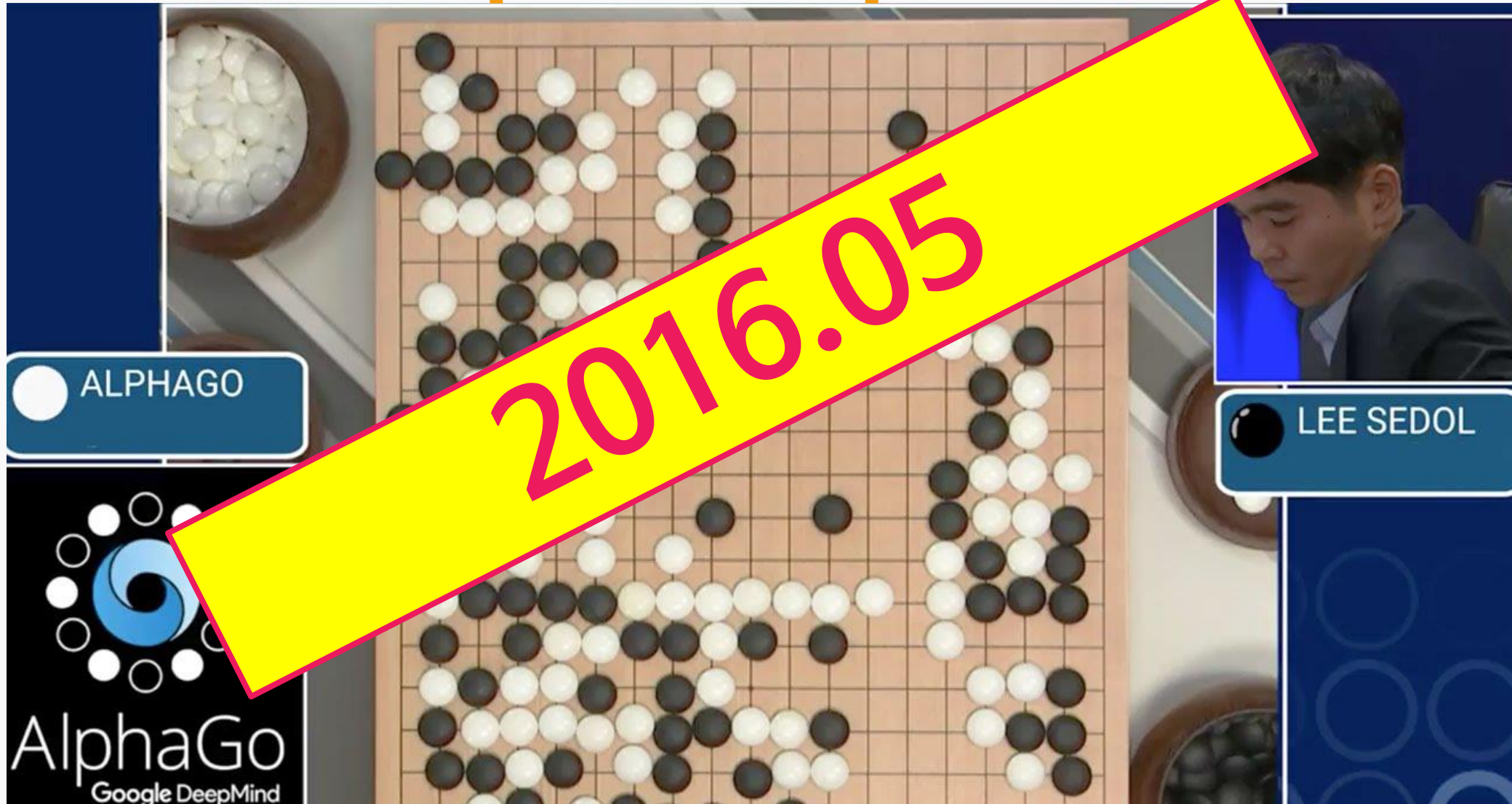
Human-level control
through deep reinforcement

2013.12 - 2015.02

Deep RL : DDPG

2015.09

Deep RL : AlphaGO



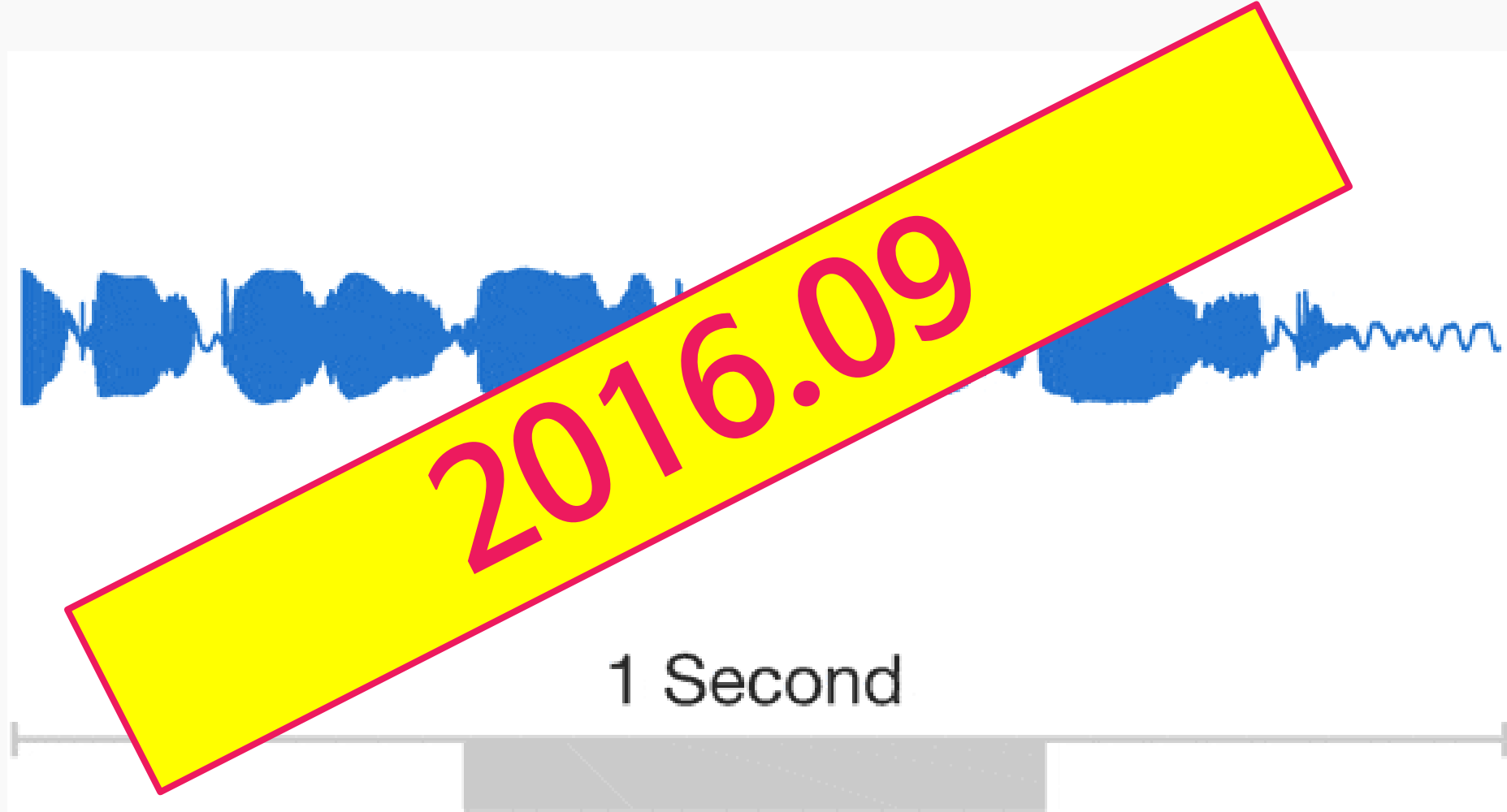
Deep RL : AlphaGO Master



Deep RL : AlphaGO Zero



Voice and Music Generation



<https://deepmind.com/blog/wavenet-generative-model-raw-audio/>

Tacotron: End-to-End Speech Synthesis

Audio samples from "Tacotron: Towards End-to-End Speech Synthesis"

Paper: [arXiv Interspeech 2017](#)

Authors: Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitra, Alexei Baevski, Abdelrahman Mohamed, Andrew Senior, Michael Auli, David S. Saurous

Abstract: A text-to-speech synthesis system typically consists of multiple stages, each requiring extensive domain expertise and may contain brittle design choices. In this paper, we propose a single-stage system that can generate audio from text. In this system, the model can be trained completely from scratch with a large dataset of text-audio pairs. Tacotron achieves a 3.82 subjective 5-scale mean opinion score on the LibriTTS dataset. In terms of inference level, it's substantially faster than sample-level autoregressive models.

All of the below phrases are unseen by Tacotron. [See the Tacotron team.](#)

Tacotron works well on out-of-distribution phrases.

"Generative adversarial networks"

0:00 / 0:05

"Basilar membrane vibrations."

"He reads books." is based on phrase semantics.

(It replaces "read" in the two phrases.)

"He reads books."

0:00 / 0:05

"He reads books."

0:00 / 0:05

Deep Voice 2



Deep Voice 3 / Tacotron 2

2017.10-2017.12



김승일 연구소장

E-mail : si.kim@modulabs.co.kr

Homepage : www.modulabs.co.kr

FB: www.facebook.com/lab4all
www.facebook.com/groups/modulabs

Blog : www.whyDSP.org