

# AI 개념과 활용사례

한국과학기술정보연구원 이홍석





# 오늘 배울 내용

- 인공지능을 이해하기 위해서는 우리의 뇌를 잘 알아야 한다.
- 기억도 있고, 학습도 있고,
- 뇌에서 학습하는 원리를 이해하면 인공지능은 매우 쉽게 이해할 수 있다.

# 01. 인공신경망 역사

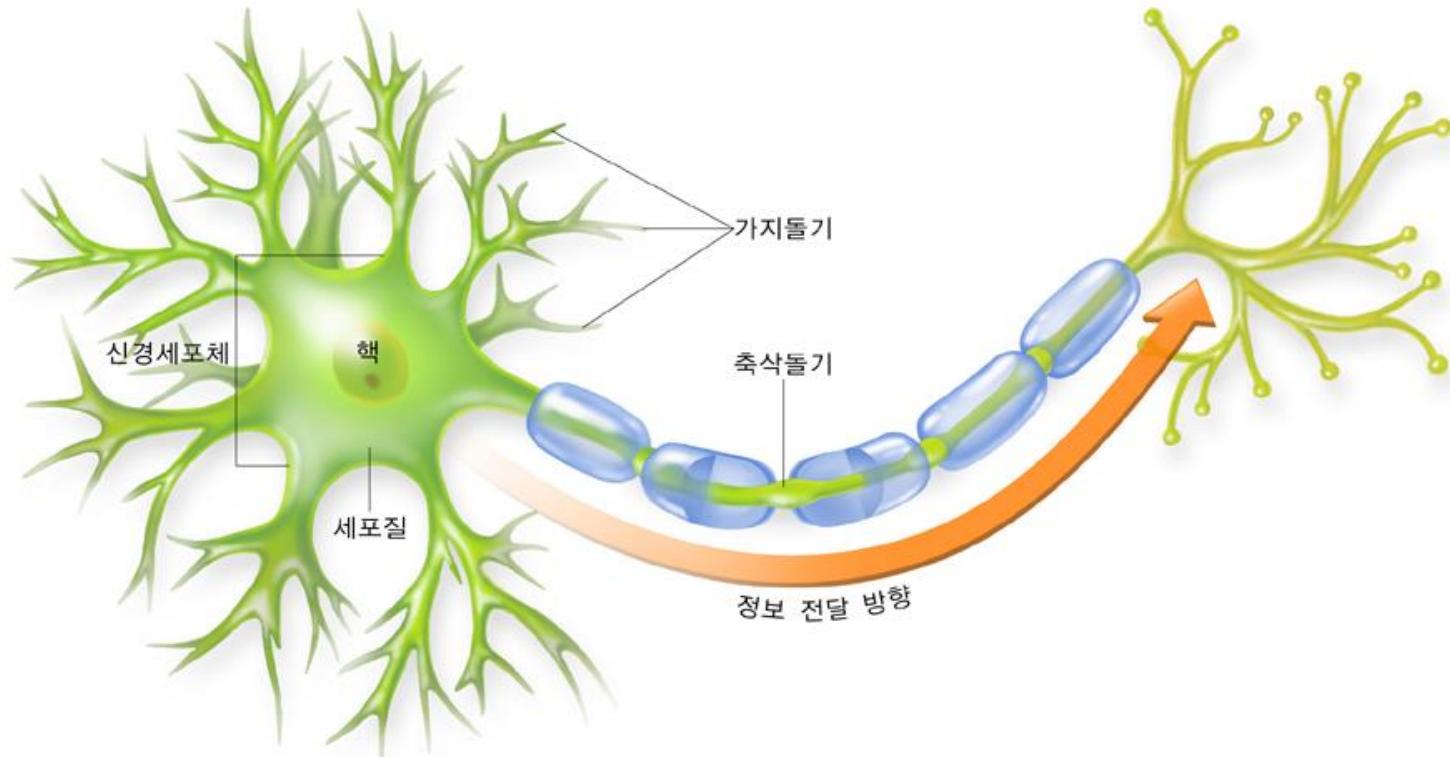
이홍석 (hsyi@kisti.re.kr)





# 인공신경망 이해 (1)

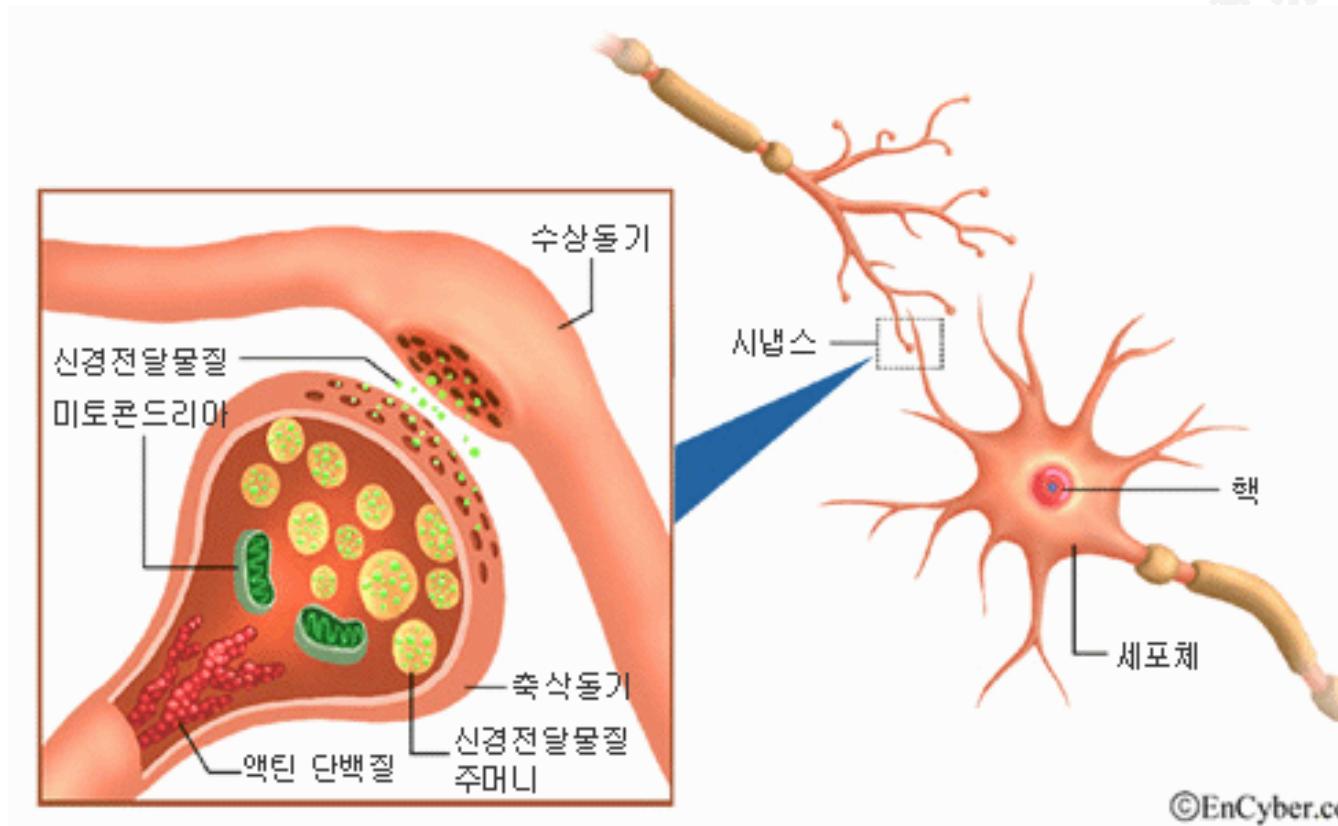
사람의 자극의 전달 과정과 시냅스





# 시냅스 (Synapse): 자극의 전달

뉴런의 집합체로 한 뉴런의 축삭돌기 말단과 다음 뉴런의 수상돌기 사이의 연접 부위





# 신경망의 역사 1950년 대부터

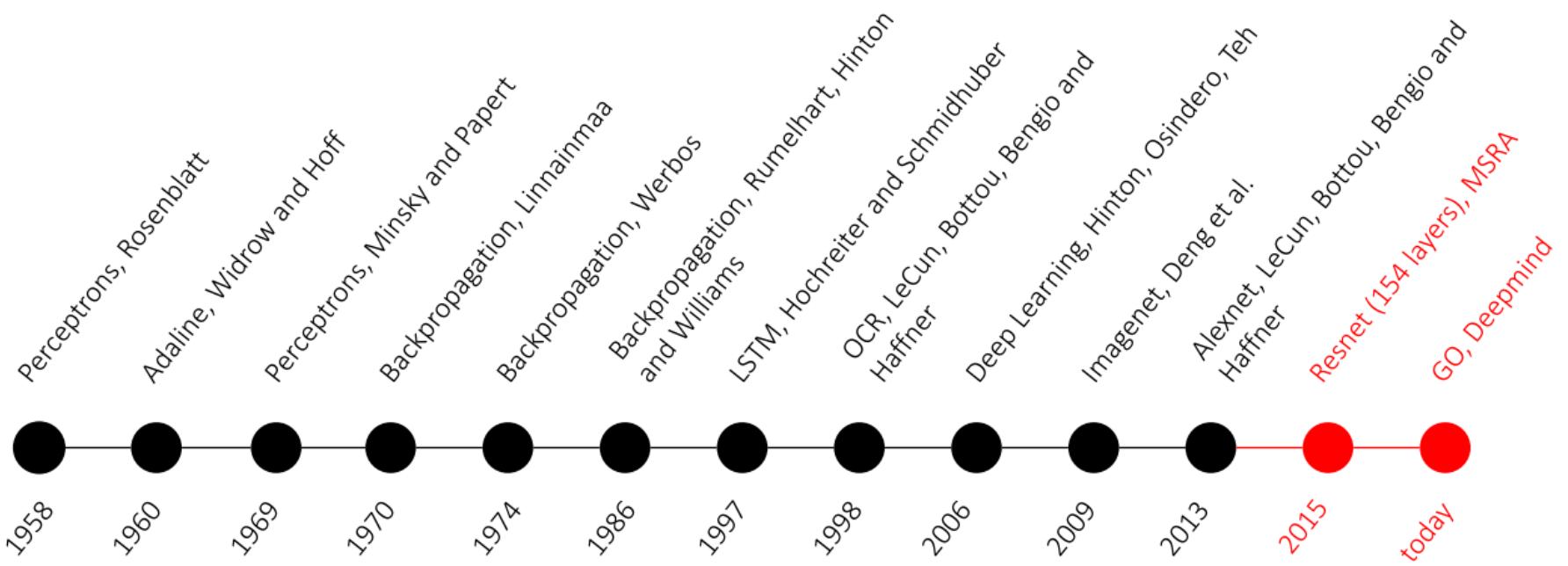
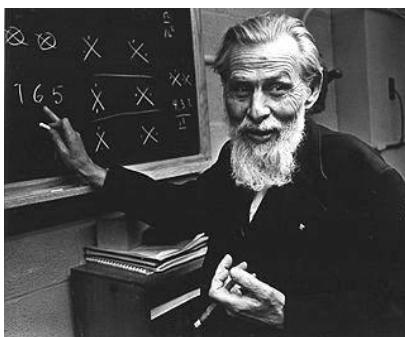


Fig. credit to Efstratios Gavves, Intro. to DL



# 인공지능의 역사 : 1943년

- 인공신경망 개념 최초 제안 : 맥컬록-피츠 신경망 모델 (1943)
  - McCulloch와 Pitts가 1943년에 ANN 최초 논문이 발표
    - “A logical calculus of the ideas immanent in nervous activity”
  - 인간의 신경 구조를 복잡한 스위치들이 연결된 네트워크로 표현할 수 있다고 제안

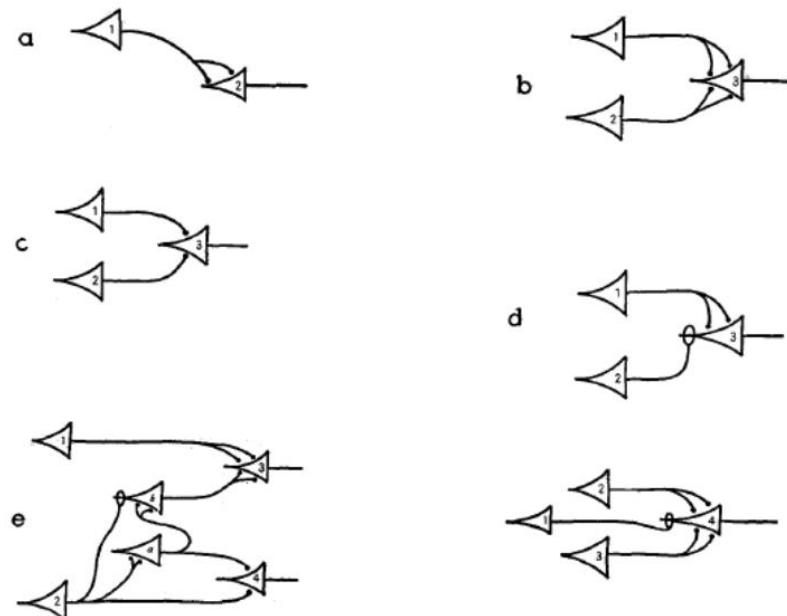


맥컬록



피츠

제안한 최초의 인공신경망 개념

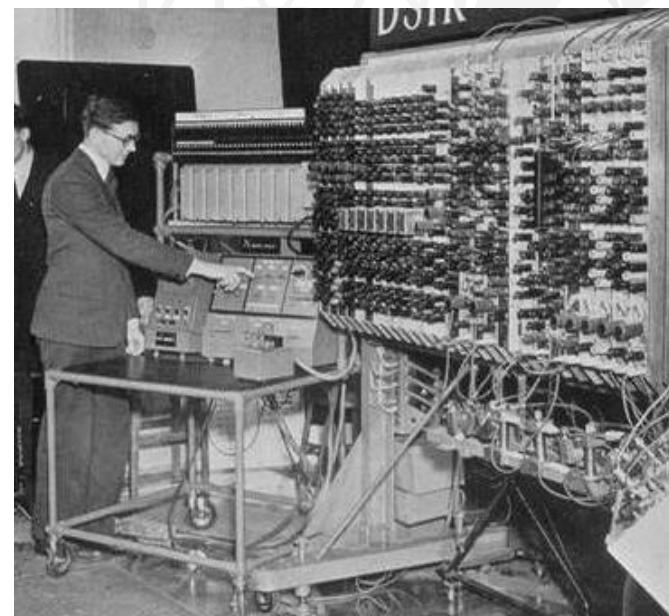
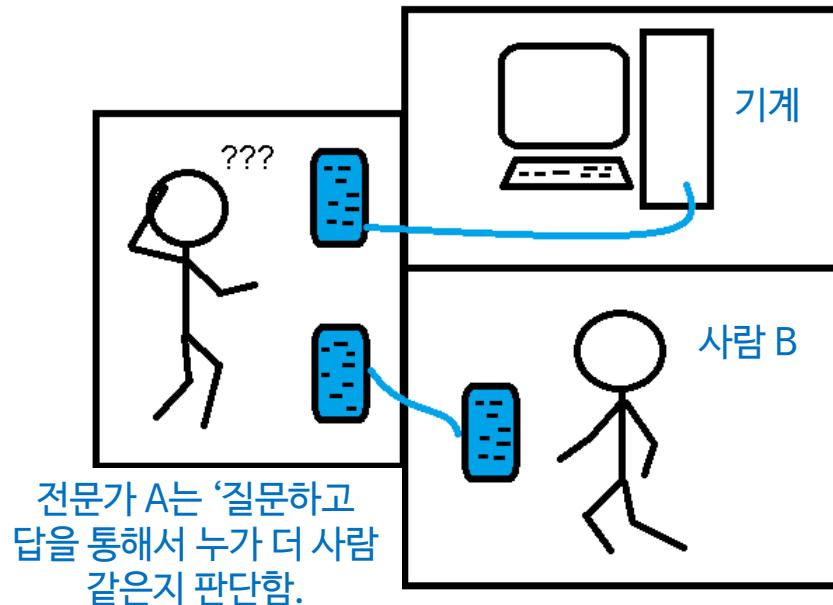




# 인공지능의 역사 (2) : 1950년

## • 튜링 테스트 (1950)

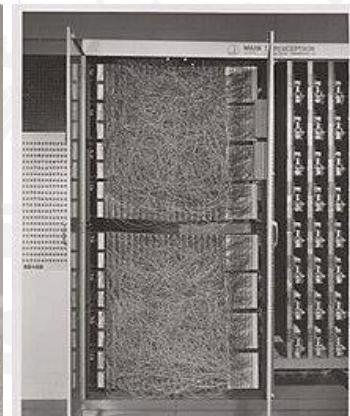
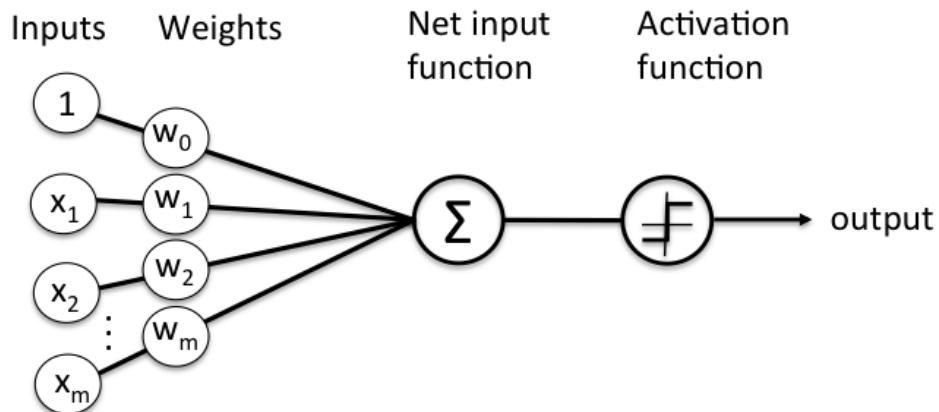
- 앤런 튜링(Turing)은 컴퓨터 과학의 아버지로 불리움.
- 기계가 사람처럼 생각할 수 있다는 것을 아래 그림처럼 테스트 함.
- 사람 A가 상태에서 2명과 대화를 했을 때, 기계(Z)가 더 자연스러움.





# 인공지능의 역사 (3) : 1958년

- ANN은 퍼셉트론 신경망 설명 : 프랑크 로젠블라트, 1958)
  - Frank Rosenblatt는 퍼셉트론(Perceptron)라는 선형분류 피드포워드 신경망
    - “The perceptron: A probabilistic model for information storage and organization in the brain.” 논문에서 제시
    - 입력과 가중치(weight)들의 곱을 모두 더한 뒤 활성함수(계단함수)로 선형 분류기
  - (문제점) 1개 퍼셉트론으로 XOR 선형 분류를 설명할 수 없음.



퍼셉트론은 현대의 딥러닝의 기초이다. 그 당시, 퍼셉트론을 통해서 진짜 인간과 같은 인공지능을 만들 수 있다는 기대가 매우 컸다.

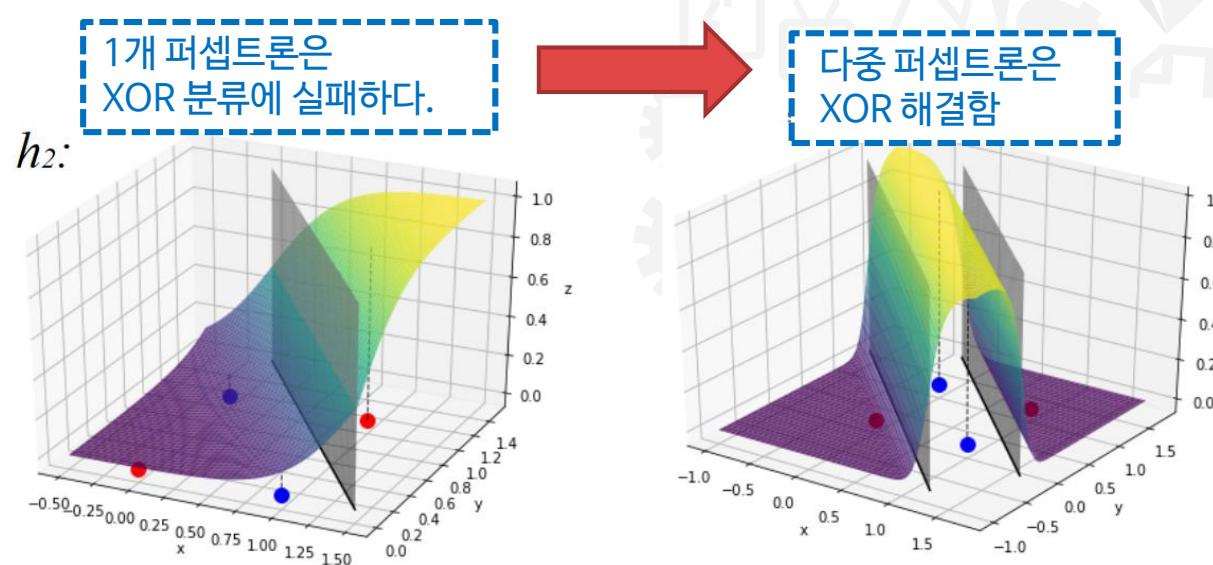
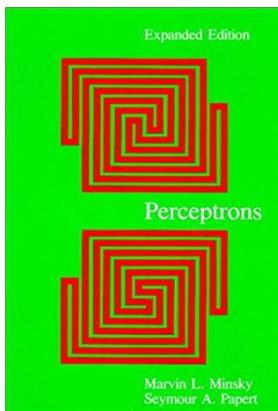
로젠블라트

IBM 퍼셉트론 계산기



# ANN의 역사 (4): 1차 격울 (1969)

- 퍼셉트론의 무용론 등장으로 1차 인공지능 겨울이 시작된다.
  - 1969년 마빈 민斯基는 ‘퍼셉트론’은 단순 선형 분류기이며, ‘XOR’ 분류도 할 수 없는 미미한 선형분류기라는 것을 수학적으로 증명함
  - 퍼셉트론 인기가 사그라들면서 인공지능 1차 암흑기가 도래한다.

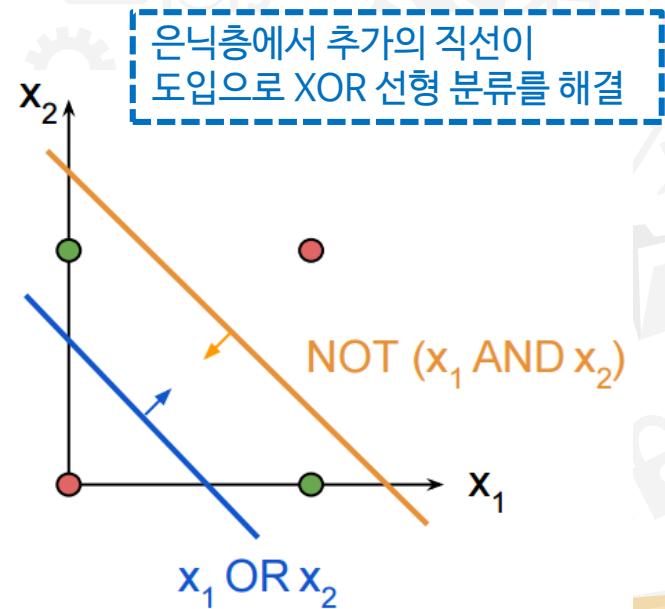
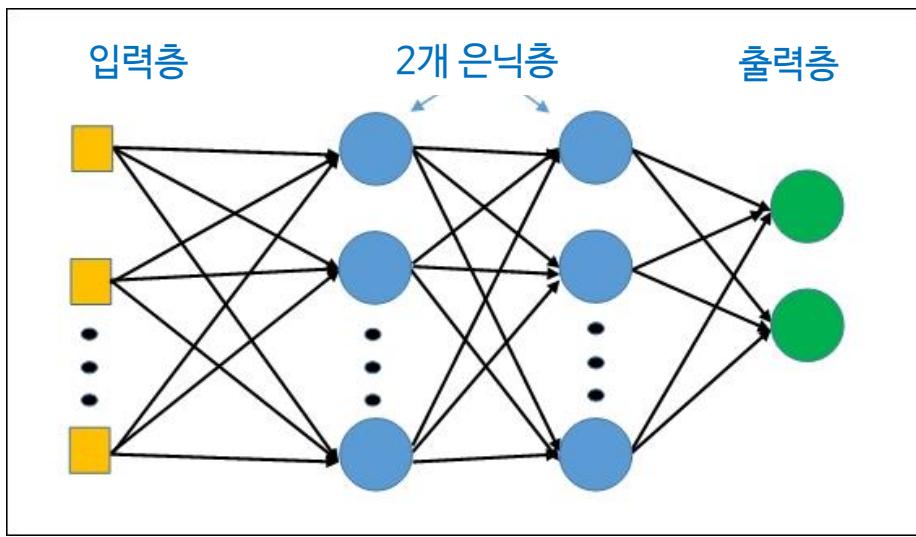


1개의 퍼셉트론은 XOR 문제에서 빨강과 파랑색을 구분하는 초평면을 만들수 없다.



# ANN의 역사 (5): 다시 봄(1986)

- **다층 퍼셉트론과 역전파 알고리즘 등장**
  - 다층 퍼셉트론은 전방향(feed-forward) 신경망으로 중간에 은닉층을 추가함
    - 기존의 퍼셉트론이 선형 분류기라는 한계에 의해 XOR 문제를 해결할 수 없었다면,
    - 다층 퍼셉트론은 은닉층(hidden layer)라는 중간 레이어를 추가로 XOR 문제를 해결
  - (문제점) 다층 퍼셉트론은 은닉층의 추가로 신경망을 훈련에 많은 어려움이 있다.

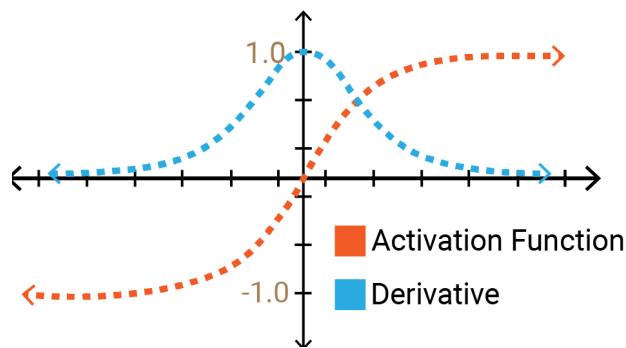




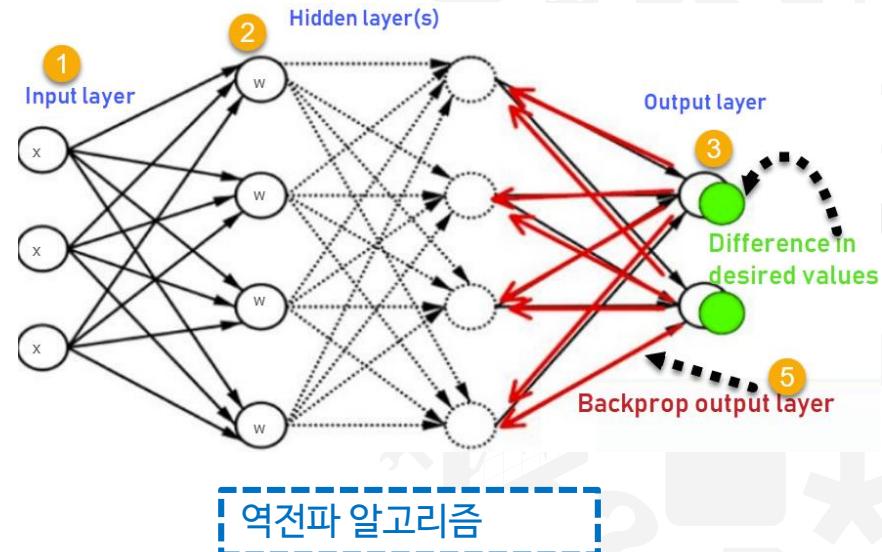
# ANN의 역사 (6): 다시 봄(1986)

- **다층 퍼셉트론과 역전파 알고리즘 등장**

- 1986년 McClelland, James L., David E. Rumelhart, and Geoffrey E. Hinton은 Backpropagation Algorithm을 제안해서 이 문제를 해결
  - 오류 역전파 알고리즘은 Feedforward 연산 이후, 오차를 후방(Backward)으로 다시 보내 줌으로써, 많은 노드를 가진 MLP라도 최적의 가중치와 Bias를 학습할 수 있다.



활성함수로 Sigmoid 함수와  
미분 가능함.

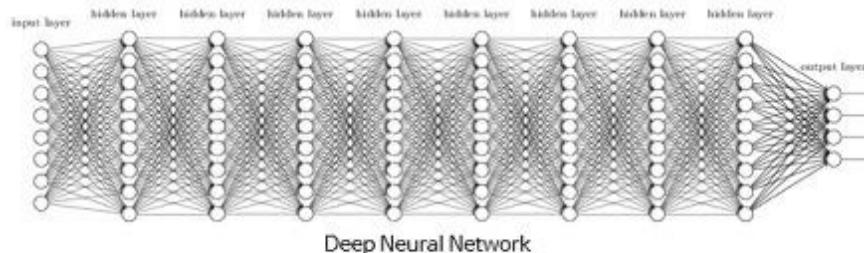


역전파 알고리즘



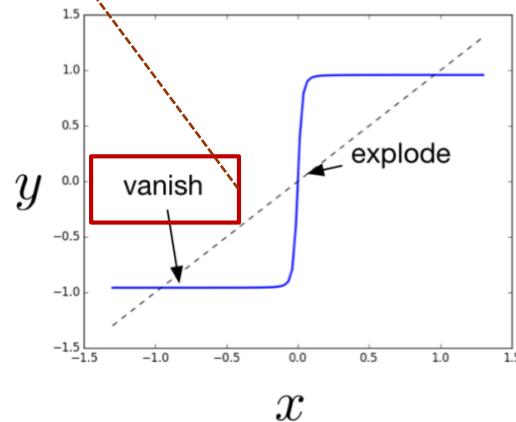
# ANN의 역사 (7): 2차 겨울(1993~)

- **다층 퍼셉트론의 경사 발산과 소멸 문제 등장(1997)**
  - RNN에서 VGP(Vanishing Gradient Problem) 문제 (1993)
  - LSTM(Long Short Term Memory)로 VGP 해결(1997), Hochreiter
  - 이시기는 기계학습 SVM, Random Forest 등이 큰 인기가 있음.



다층 신경망에서 경사값 사라짐.

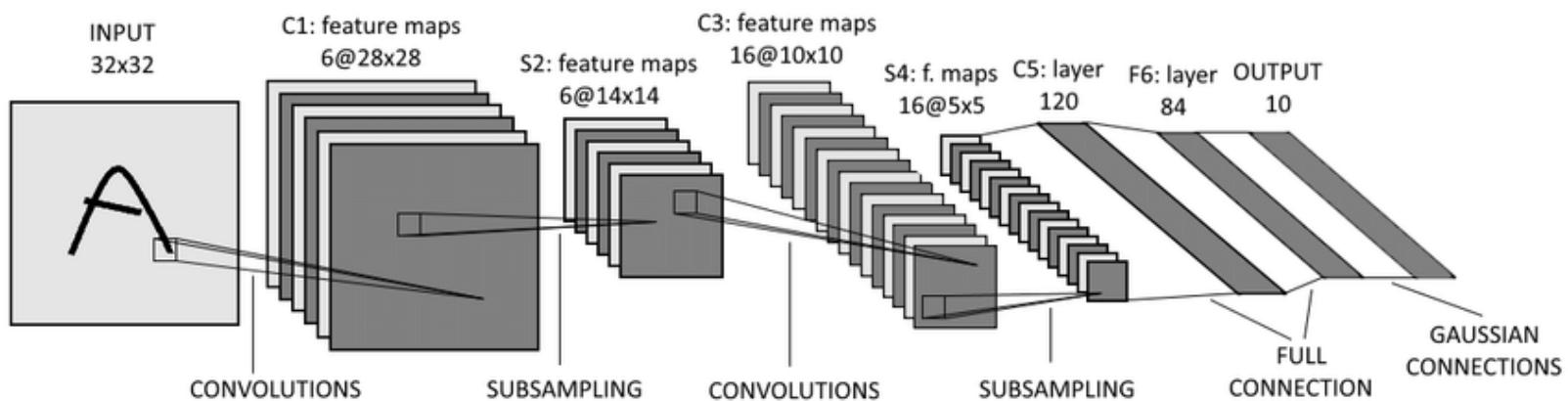
(문제의 원인) 시그모이드는  
미분값이 사라짐과 발산함





# ANN의 역사 (8): 2차 겨울 극복 노력

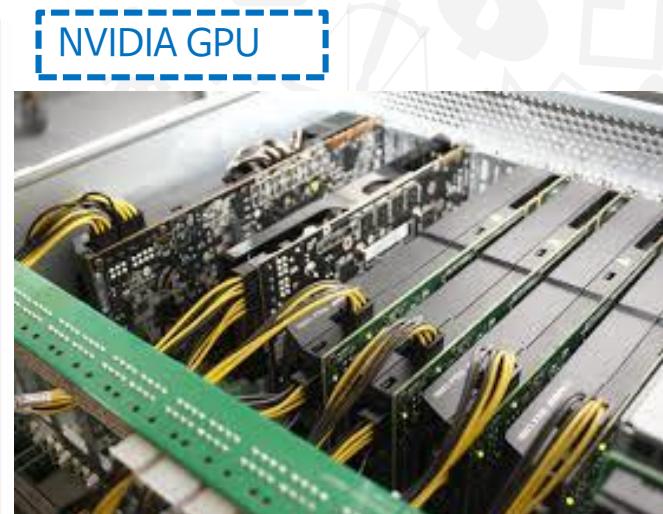
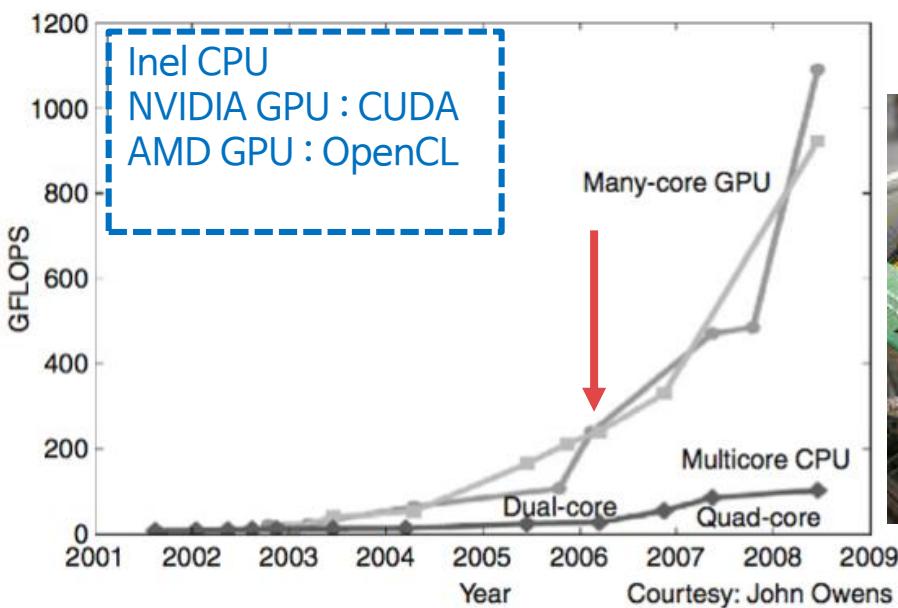
- 초기 합성곱 신경망(CNN)인 LeNet 신경망 성과(1998)
  - 1998년 얀 루큰(Lecun)은 MLP와 역전파를 MNIST 이미지에 적용 성공
  - LeNet-5라는 현대의 CNNs의 시초 제안





# ANN의 역사 (9): 2차 겨울 지속

- 인공지능 2차 겨울 10년간 지속됨 (1993~2006)
  - 과적합문제(Overfitting)
  - Vanishing Gradient (다층 신경망 역전파 과정에서 기울기 값이 사라짐)
  - 계산이 너무 느리다 (Too slow)
    - GPU, TPU, Xeon Phi 등 가속기 컴퓨팅 등장이 아직 안됨

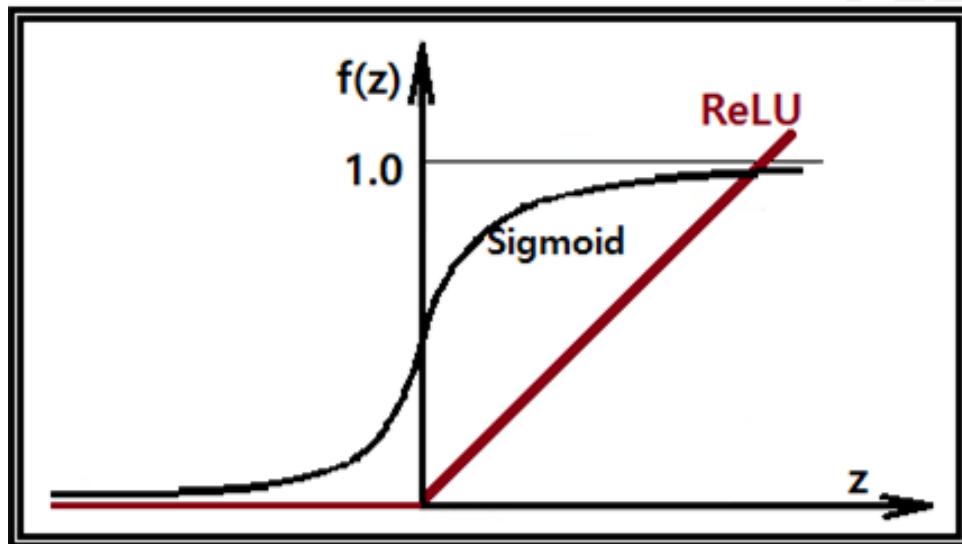


Courtesy: John Owens



# ANN의 역사 (10): 다시 봄(2006)

- 신경망 대신 딥러닝으로 인공지능 분야의 부활 (2006~)
  - 드롭아웃 층(dropout layer) 도입으로 과적합 문제를 해결
  - ReLU(Rectified Linear Unit) 활성함수 도입으로 기울기 사라짐 문제 해결
  - GPU 컴퓨팅과 고속 최적화 알고리즘 등장
  - 2006년 제프리 힌튼은 DBN(Deep Brief Network)을 적용
    - ILSVRC에서 오류율을 15%로 획기적으로 낮춤

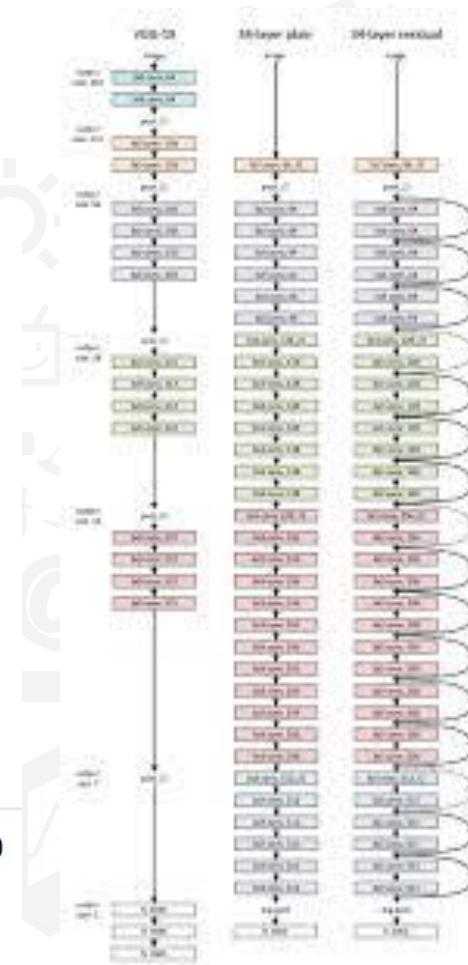
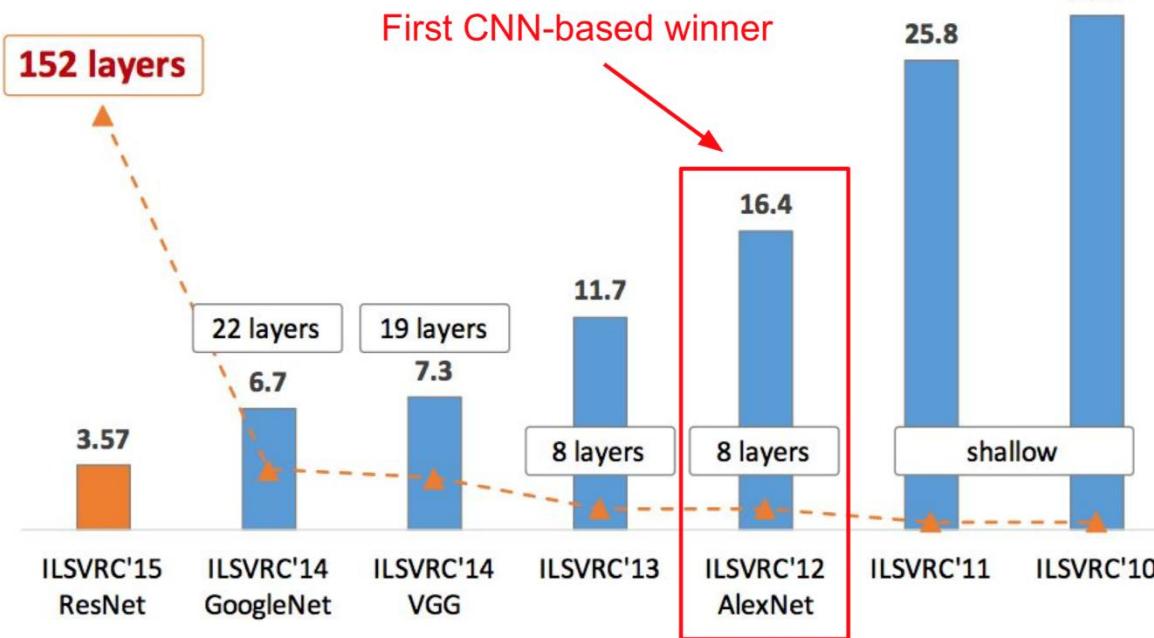




# ANN의 역사 (11): AI 활용 (2015)

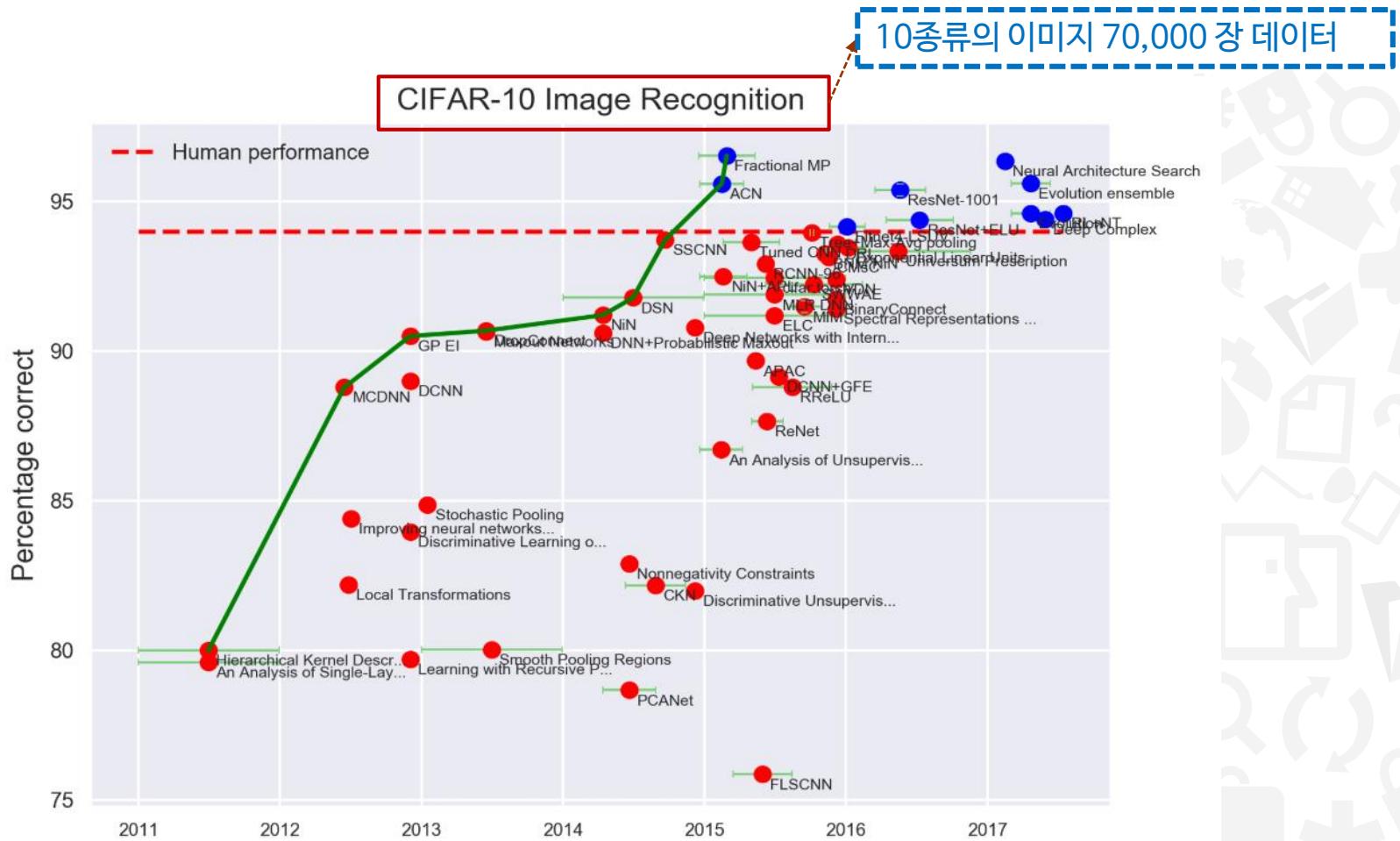
- 인간 수준 이상의 ResNet 성공

- Resnet-152는 이미지넷 챌린지에서 3.57% 오류률
  - 인간의 오류율은 대체적으로 5%정도





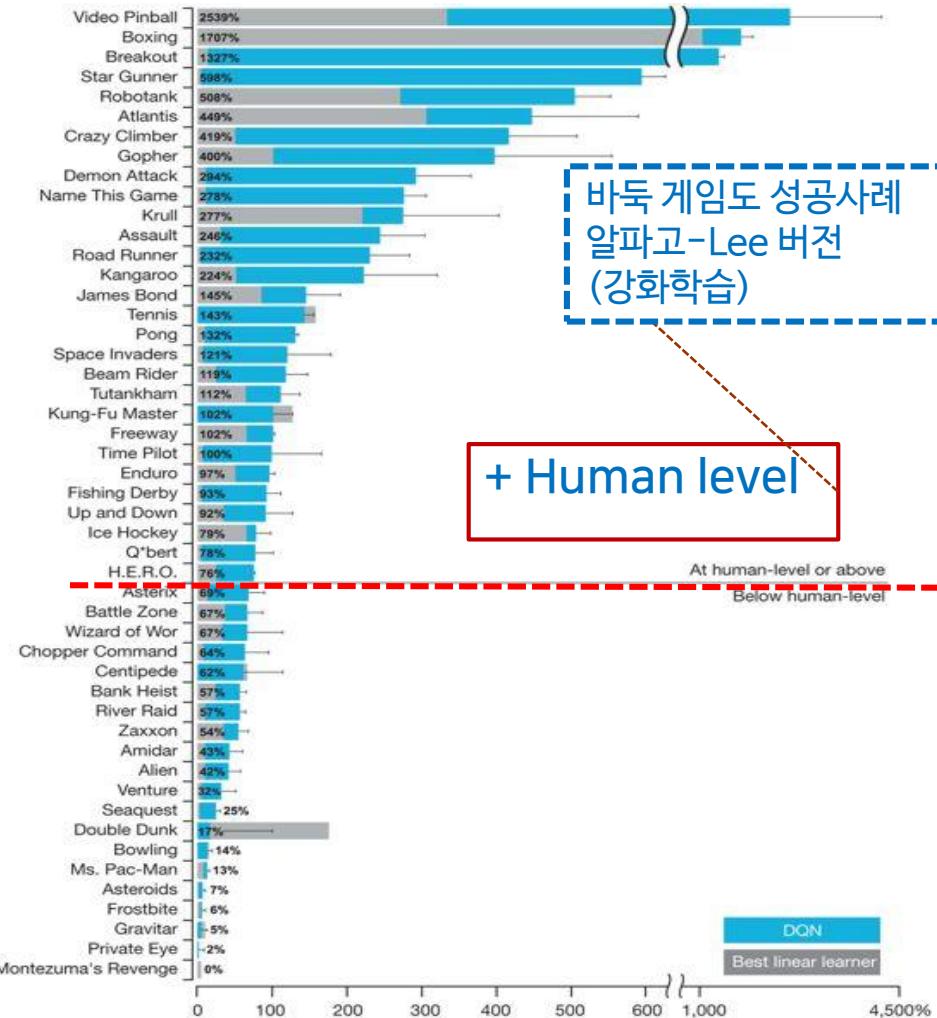
# ANN의 역사 (12): 이미지 및 비전



<https://www.eff.org/ai/metrics>



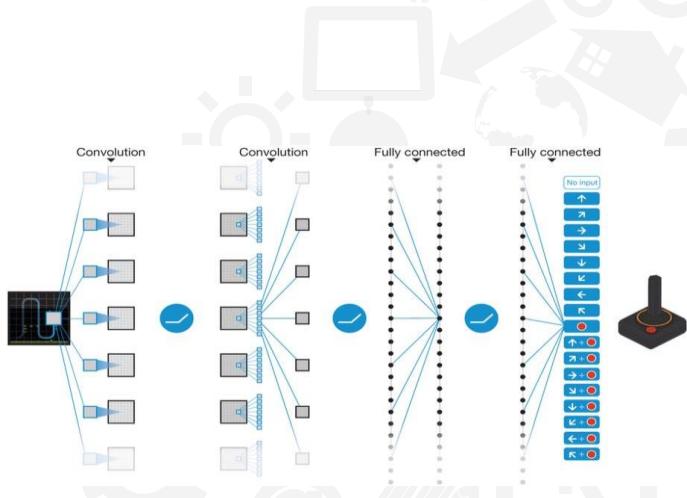
# ANN의 역사 (13): AI 활용 (2015)



바둑 게임도 성공사례  
알파고-Lee 버전  
(강화학습)

+ Human level

심화 강화학습을 게임에 적용하면,  
사람보다 우수한 성능을 보임.



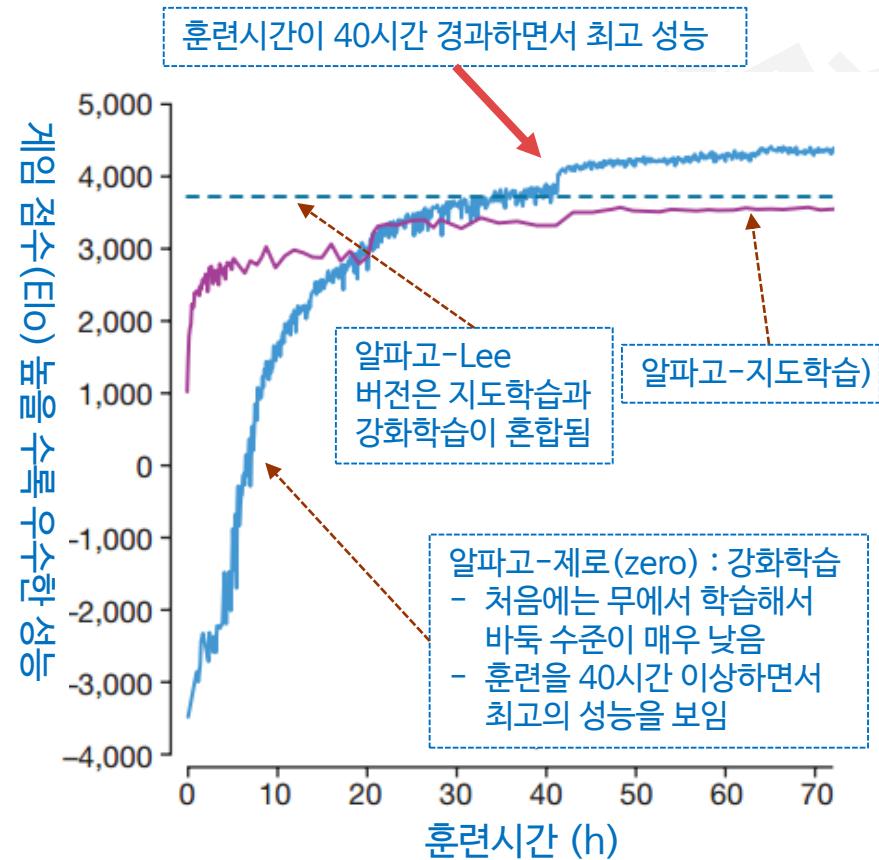
구글 딥마인드, Nature 2015



# ANN의 역사 (14): 알파고 (2016)

알파고(AlphaGo)는 3개 버전

- ① 알파고-지도학습(바둑 기보)
- ② 알파고-리 (이세돌 9단)
- ③ 알파고-제로 (강화학습)



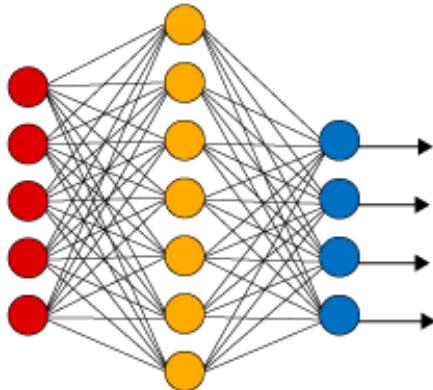
자료: Mastering the game of Go without human knowledge , David Silver, et al. Nature(2017)



# ANN의 역사 (13): Deep vs Shallow

딥러닝은 은닉층 2개 이상으로 구성된 인공신경망으로 정의함.

A. 단순 인공신경망(Shallow)  
1개의 은닉층과 7개의 뉴런으로 구성

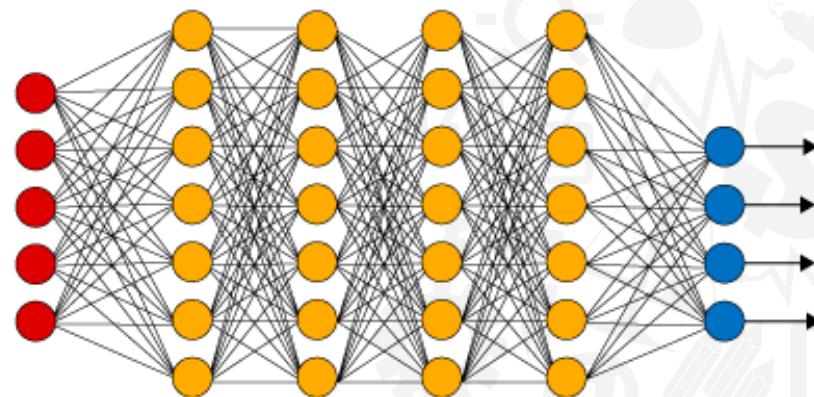


● Input Layer

● Hidden Layer

● Output Layer

B. 딥러닝 구조  
4개의 은닉층과 4x7(28)개의 뉴런으로 구성



자료: <https://www.xenonstack.com/blog/data-science/log-analytics-deep-machine-learning-ai/>

(퀴즈) 단순 인공신경망 II(Shallow)

만일, 1개의 은닉층과 28개의 뉴런으로 구성되었다면, B와 비교하면 무엇이 다를까?

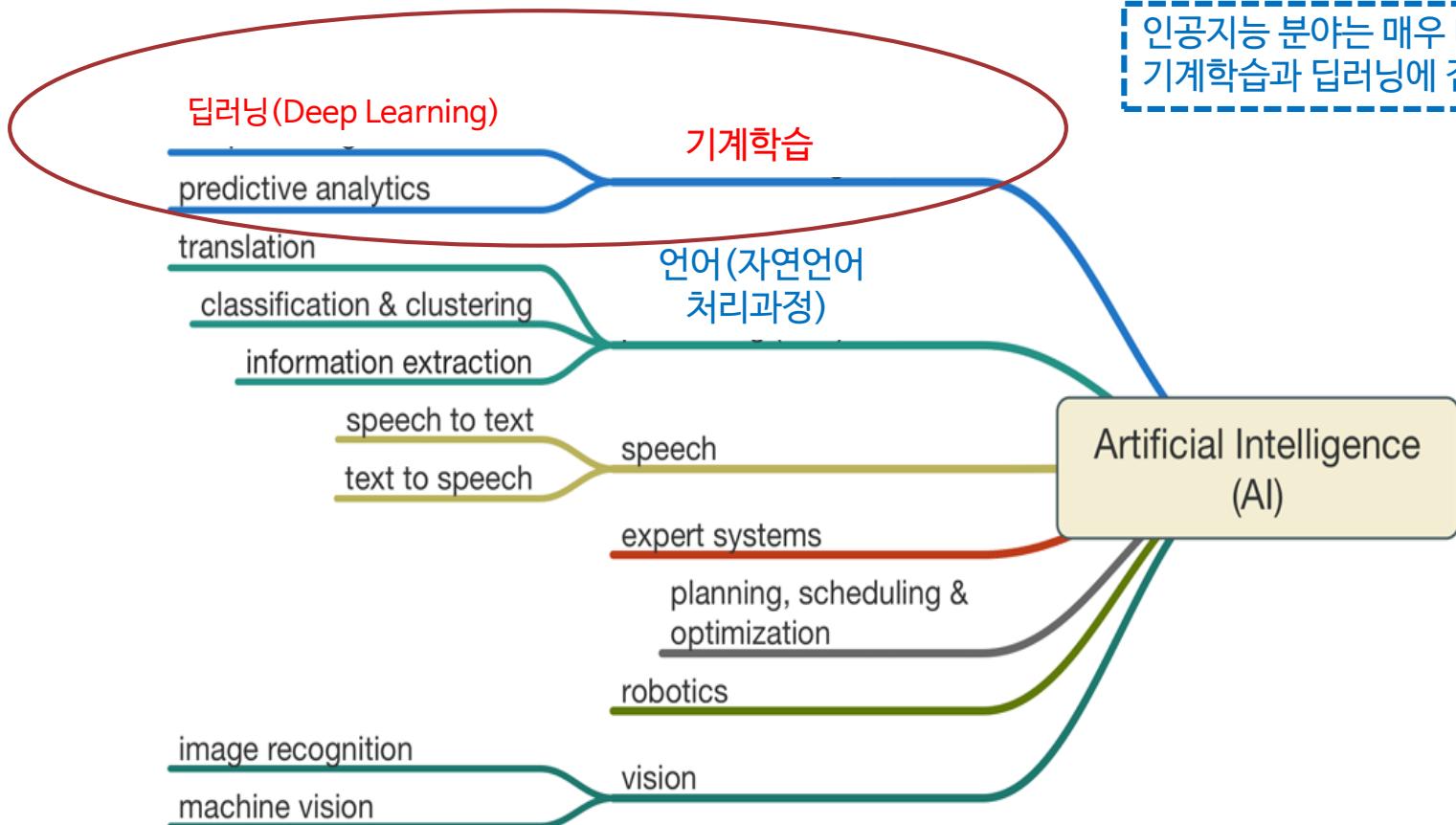
## 02. 인공지능 활용 분야

이홍석 (hsyi@kisti.re.kr)





# 인공지능의 분류 (1)

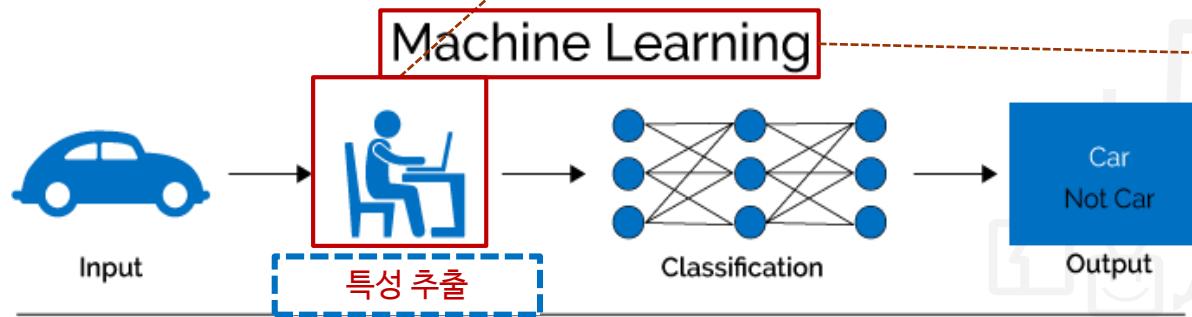


인공지능 분야는 매우 다양하다.  
기계학습과 딥러닝에 집중함



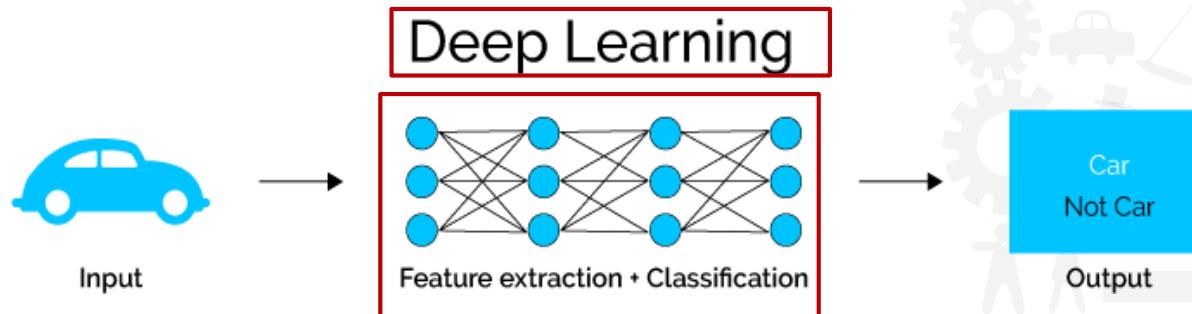
# 인공지능의 분류 (2)

머신러닝은 입력 데이터의 특성(피처)를 선택과 추출을 통해서 데이터를 훈련 시킴.



머신러닝과 딥러닝  
중에 무엇을  
사용해야 하나?

판단하는 기준은  
무엇인가?

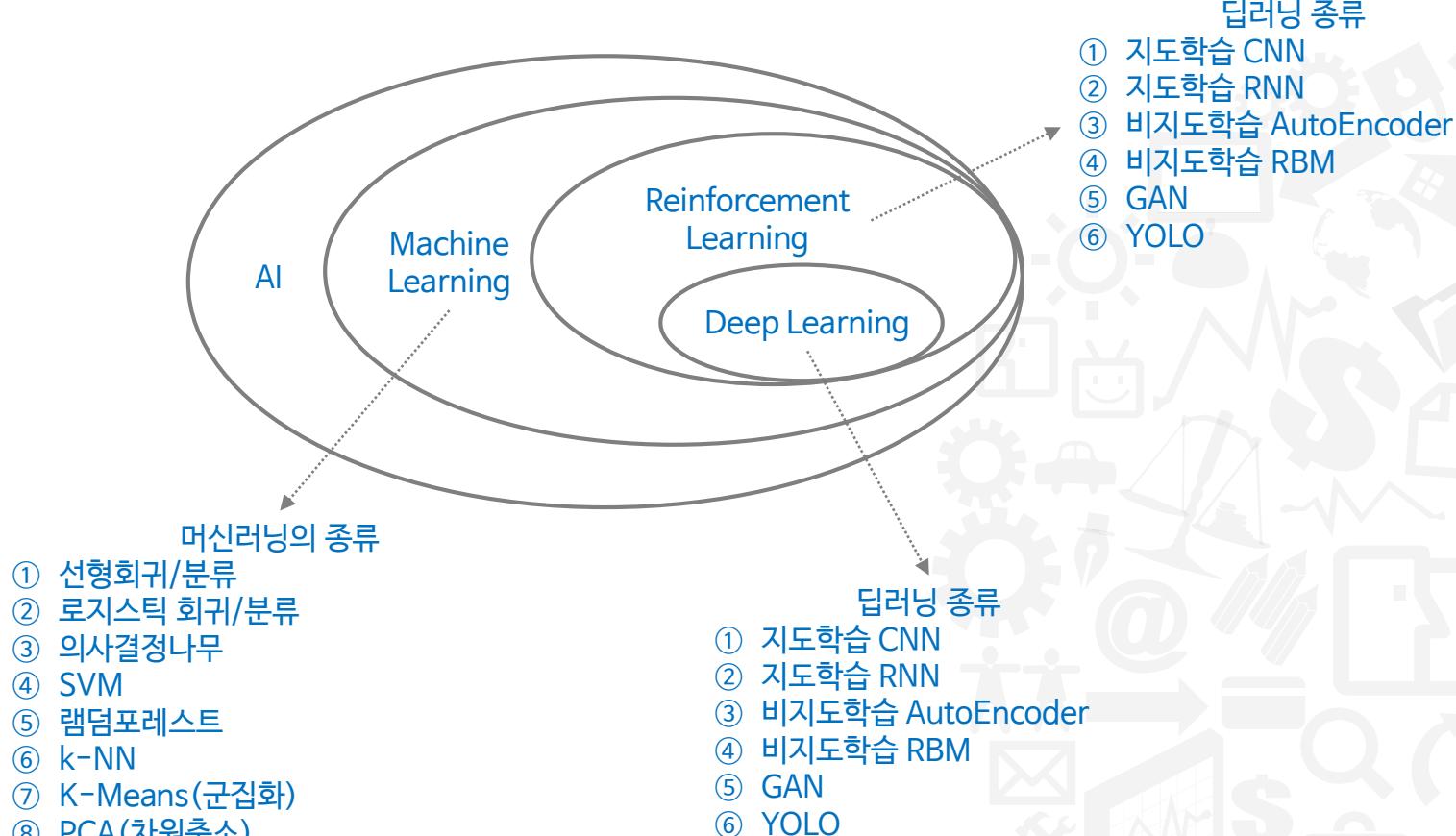


자료: <https://www.xenonstack.com/blog/data-science/log-analytics-deep-machine-learning-ai/>

딥러닝은 입력 데이터의 특성을 선택이나 추출하는 대신에, 수 많은 은닉층(필터)을 사용하여, 입력데이터에 숨겨져있는 패턴을 학습하는 방법.



# 인공지능의 분류 (3)

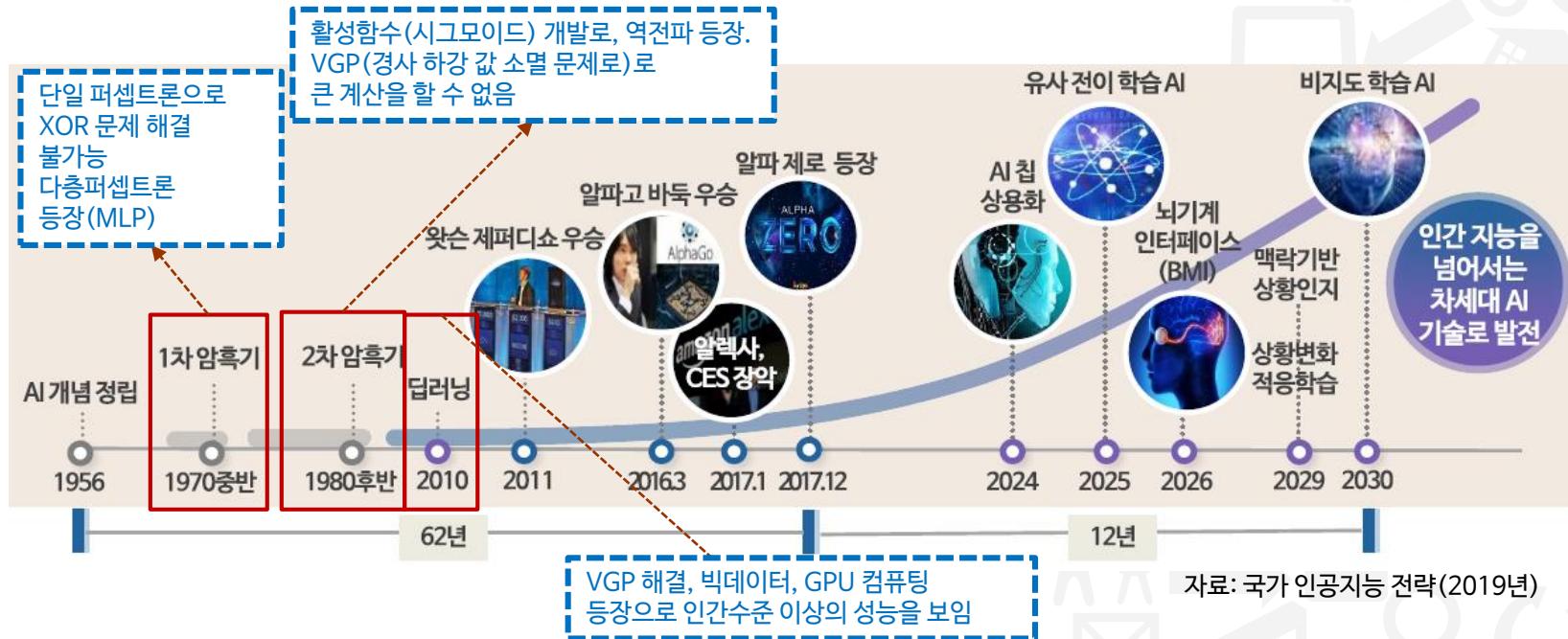


scikit-learn 패키지를 주로 사용하고 있음



# 국가 중점 인공지능 R&D 계획

두 번의 AI암흑기에도 불구, 데이터, 컴퓨팅 파워, 딥러닝 진화 등으로 AI 부흥기



- \* 1차 ('70년대) : 메모리·처리속도 등이 구축 미비로 실패함. AI에 대한 지원 중단
- \* 2차 ('00년대) : 전문가시스템의 고유지비용·업데이트 한계·오류 등으로 AI회의론 확산

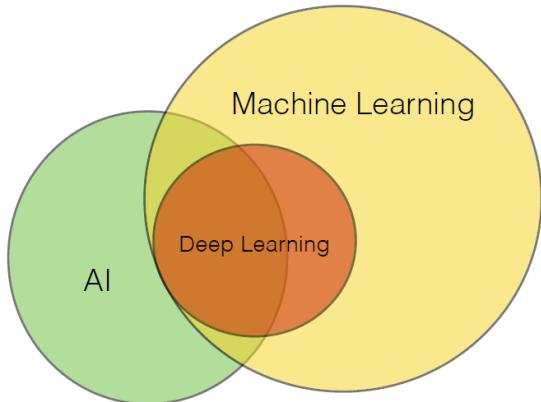
### 03. 지도학습의 이해

이홍석 ([hsyi@kisti.re.kr](mailto:hsyi@kisti.re.kr))





# 인공지능 분류 (0)



## Supervised Learning

- Labeled data
- Direct feedback
- Predict outcome/future

## Unsupervised Learning

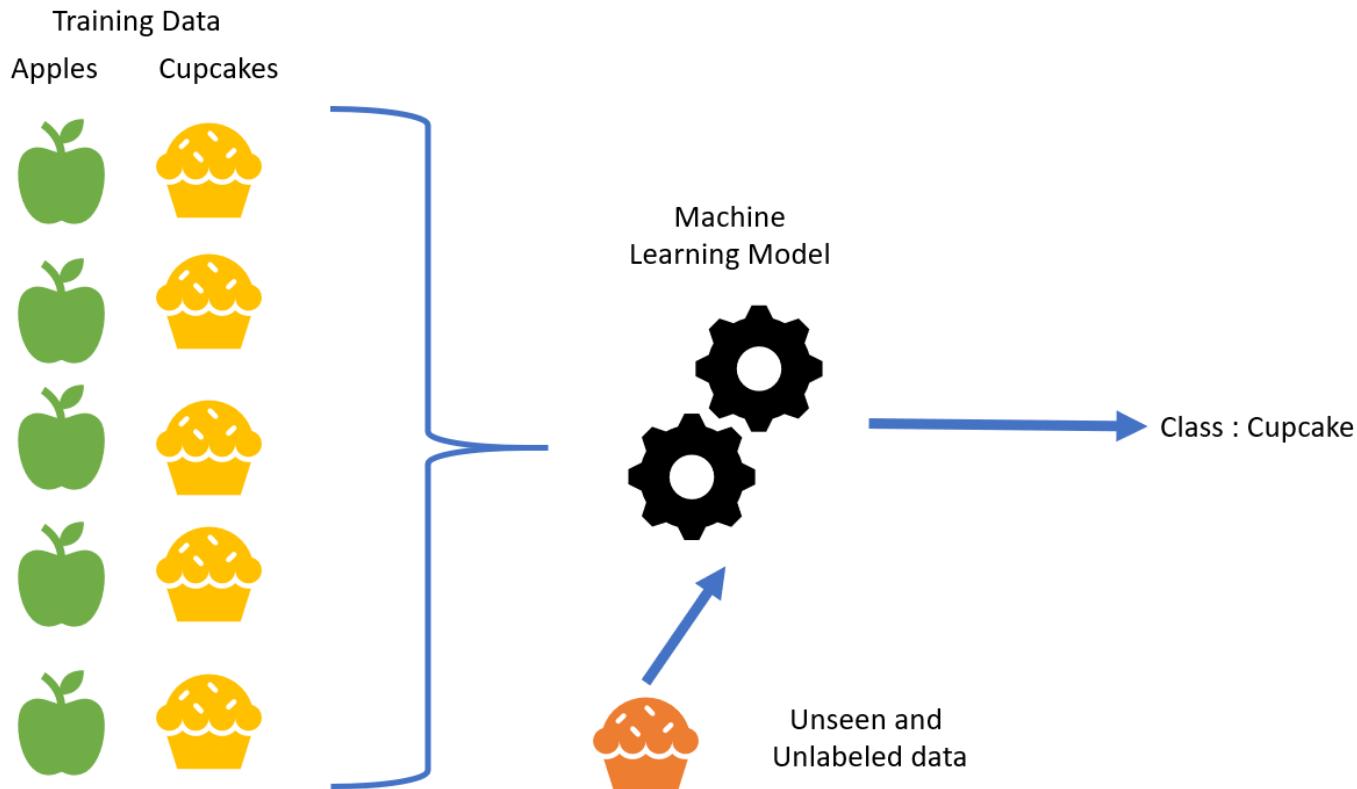
- No labels/targets
- No feedback
- Find hidden structure in data

## Reinforcement Learning

- Decision process
- Reward system
- Learn series of actions

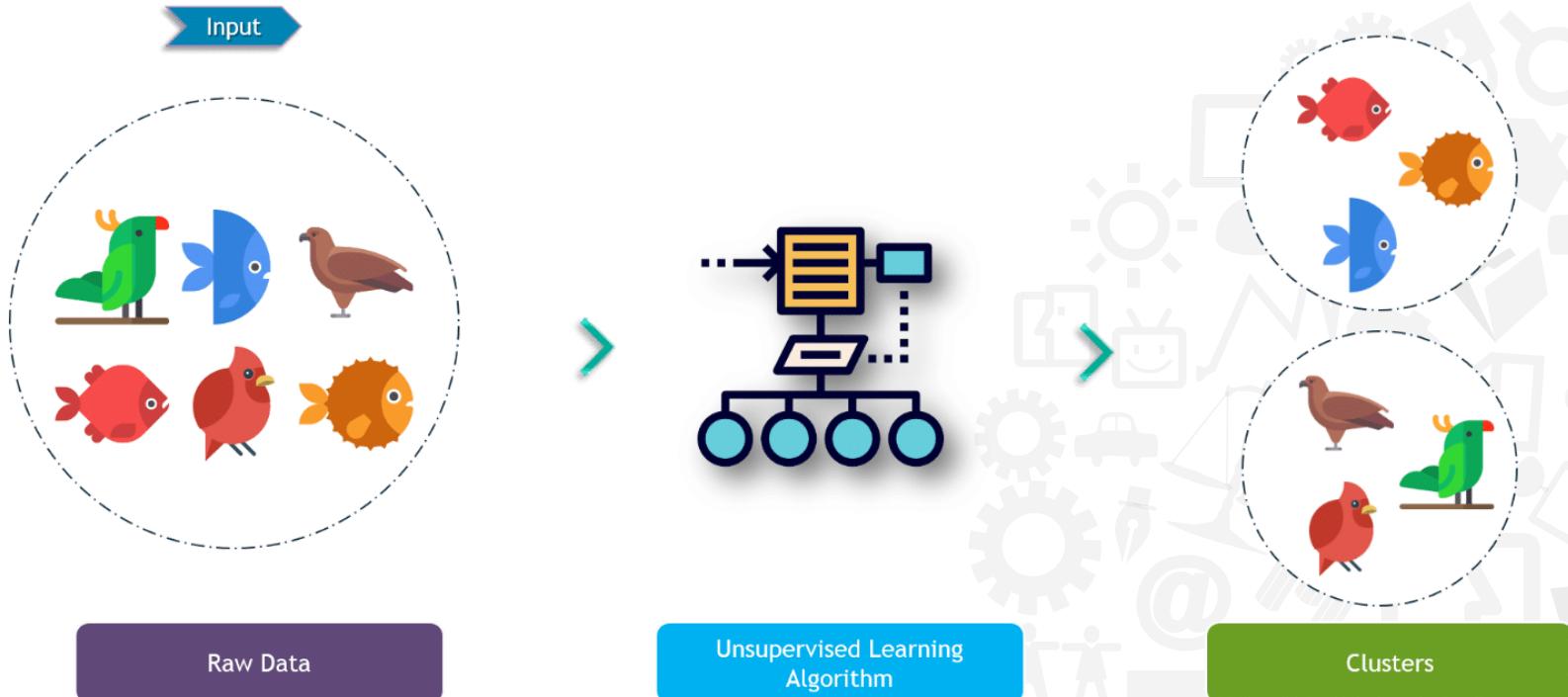


# 지도학습





# 비지도학습





# 강화학습 예 : 스타크래프트 //

House (environment)

CAT (Agent)



Sitting

State (Action)

© guru99.com

(Reward)

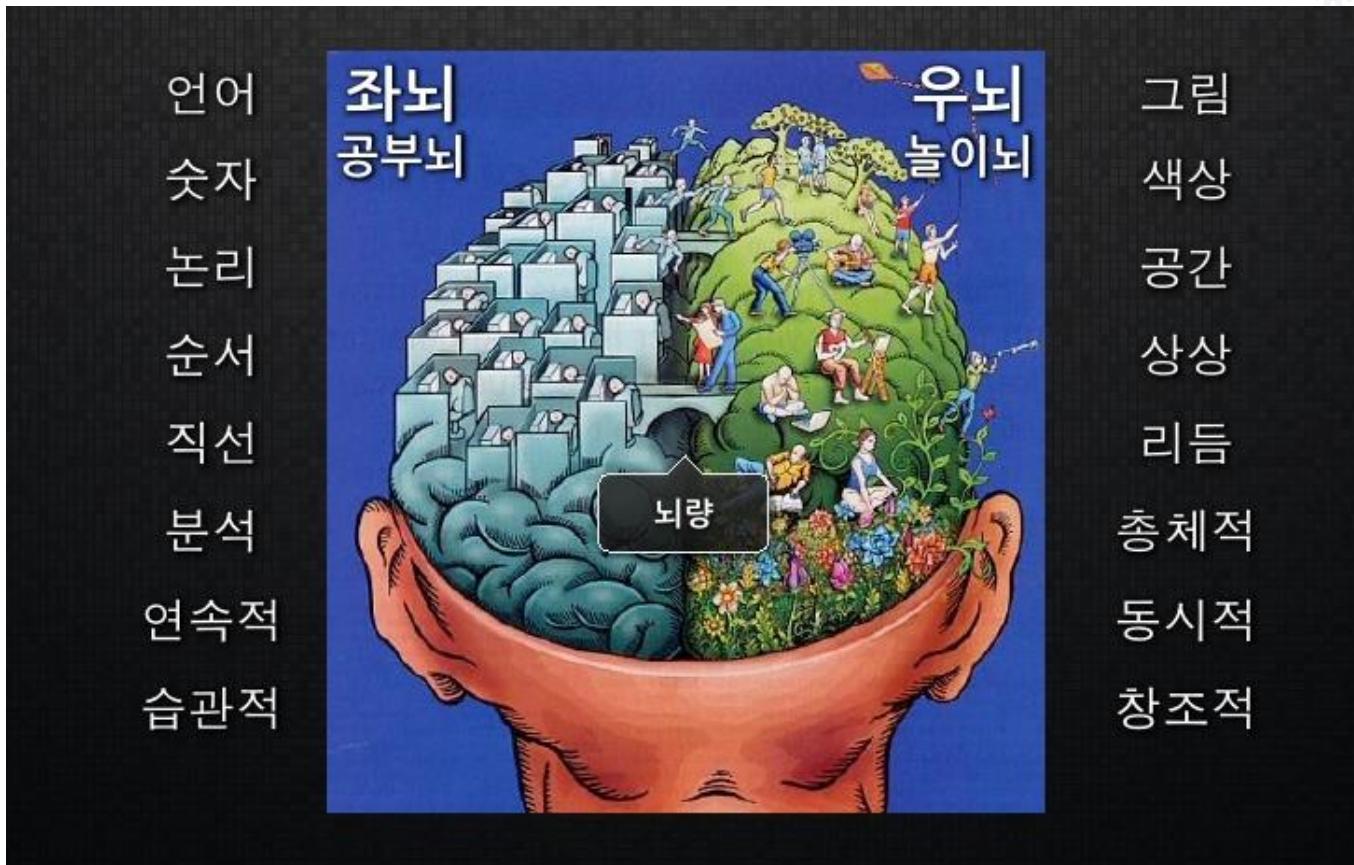


Walk



# 뇌(brain)와 기억(Memory) (1)

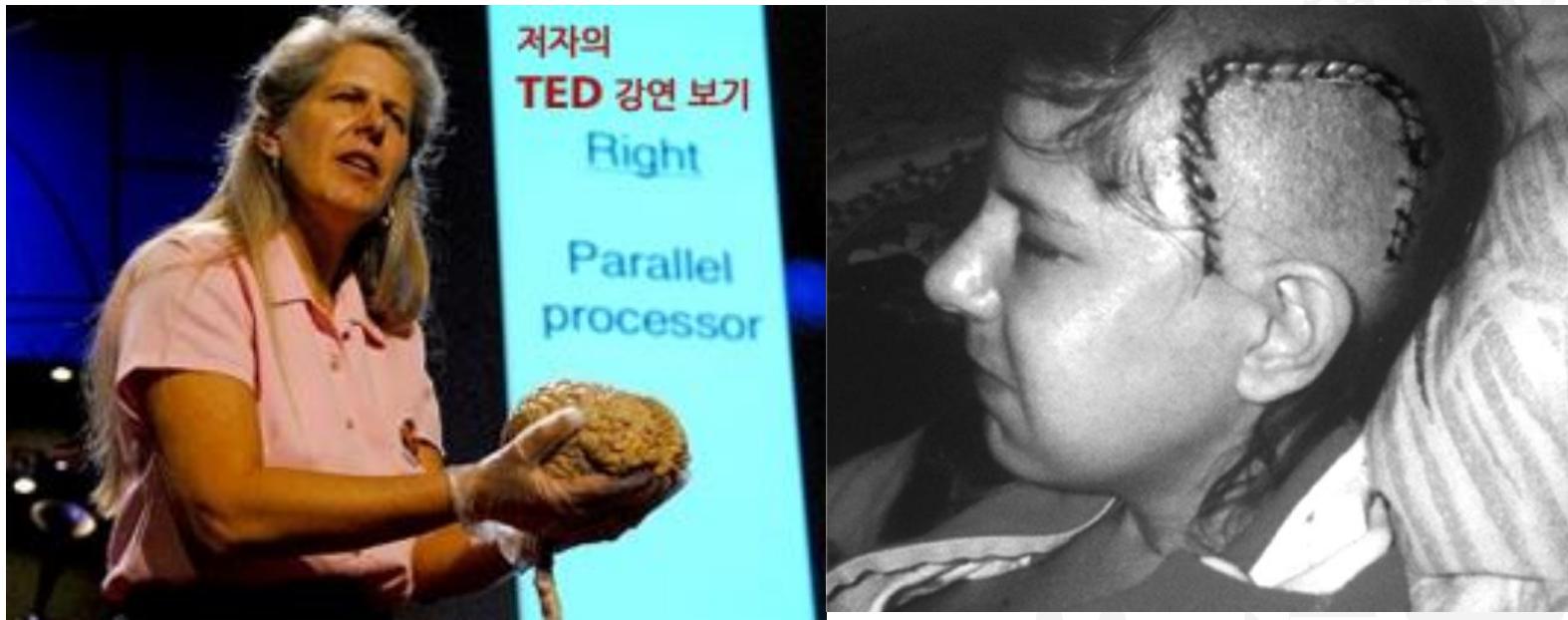
뇌량(Corpus Callosum)은 좌뇌와 우뇌의 신경세포들을 서로 연결해 주는 신경다발





# 뇌와 기억 (2) : 긍정의 뇌

J. 테이러(Taylor) 교수: 하버드대 뇌과학자 뇌출중 체험기





# 학습(훈련) 그리고 메모리(기억) (1)

좌뇌와 우뇌의 차이는 왜 생길까?

- 상과 별

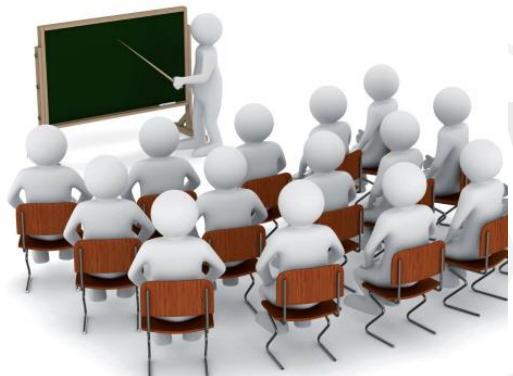
「지도학습의 예:」

지도를 받는다.  
대부분 정답이  
있는(라벨) 것을 학습한다.



비지도학습의 예:

지도학습은 정답이 있는 것을  
가르친다.  
하지만, 비지도학습은 예를들면,  
시험을 보고, 답안지를 주지 않으면  
어떨까?  
또 다시 다른 시험을 본다고 한다면,  
물론 답안지는 없다.

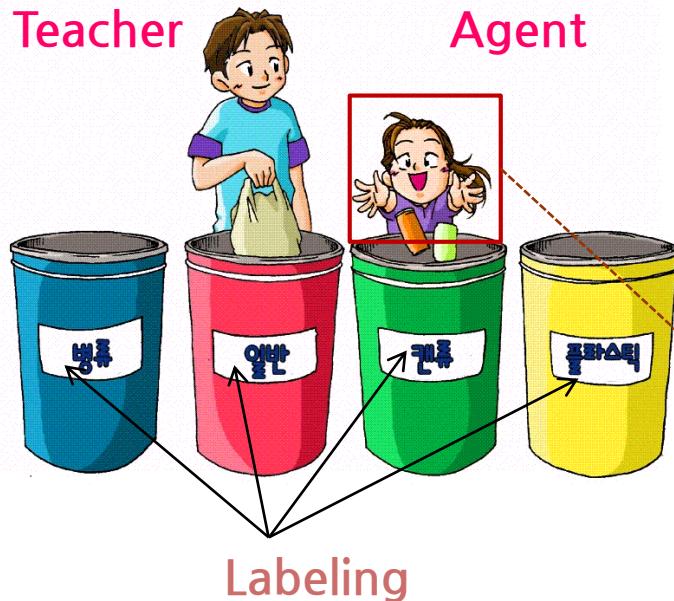




# 지도학습과 레이블(Label) (1)

## Supervised Learning(지도학습)

- ① 입력(x), 출력(y)
- ② Labeling
- ③ Prediction



빅데이터 : 원본 데이터  
훈련데이터+테스트 데이터로 나눔  
비율은 7:3정도가 일반적임.



### 훈련데이터와 테스트 데이터

기존 훈련 데이터로 학습을 완료한 이후에,  
새로운 테스트 데이터(쓰레기)를 누구의  
도움도 없이 혼자서 분류를 한다면, 분류의  
정확도는 얼마나 될까요?



# 지도학습과 레이블(Label) (2)

대표적인 딥러닝(머신러닝) 기반 지도학습을 위한 기초 데이터 제공함.  
MNIST 데이터는 훈련용으로 60,000장 이미지와, 테스트용으로 10,000장

label = 5



label = 0



label = 4



label = 1



label = 9



label = 2



label = 1



label = 3



label = 1



label = 4



label = 3



label = 5



label = 3



label = 6



label = 1



label = 7



label = 2



label = 8



label = 6



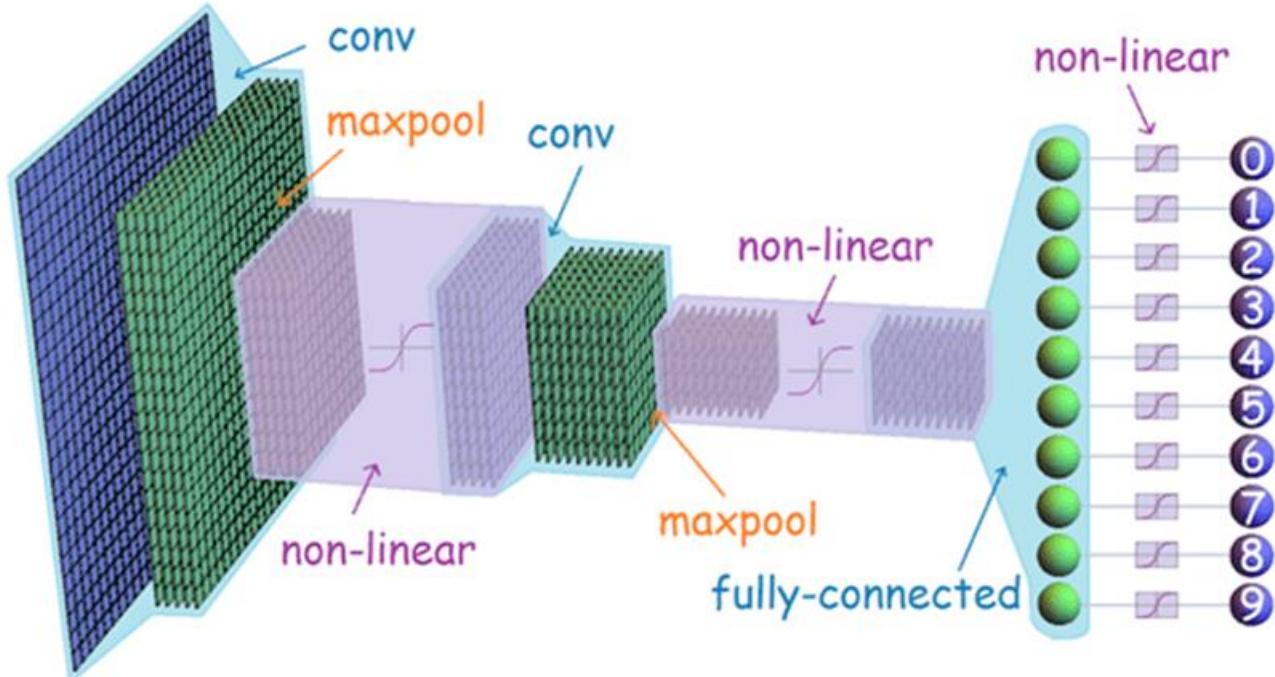
label = 9



모든 손글씨 이미지  
마다 라벨링이 되어  
있음.  
이미지 9와 라벨 '9'는  
항상 쌍으로 저장됨



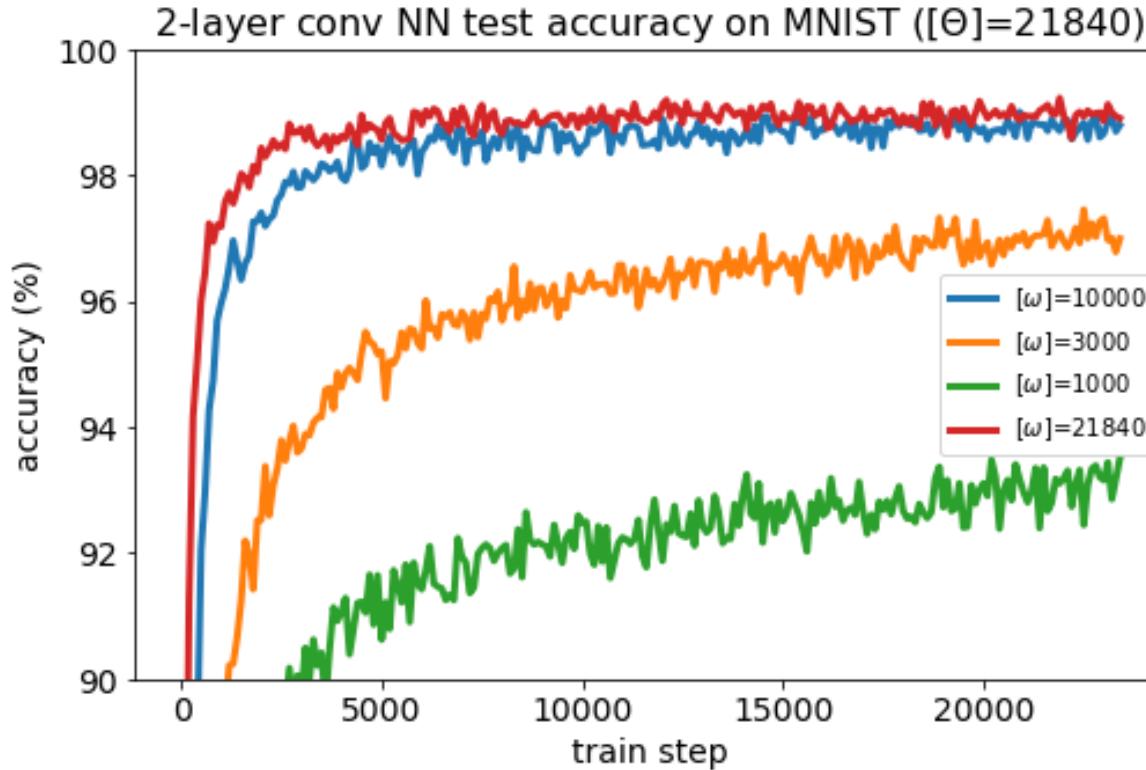
# 지도학습과 레이블(Label) (3)



<https://sefiks.com/2017/11/05/handwritten-digit-recognition-using-cnn-with-keras/>



# 지도학습과 레이블 (5) : 정확도

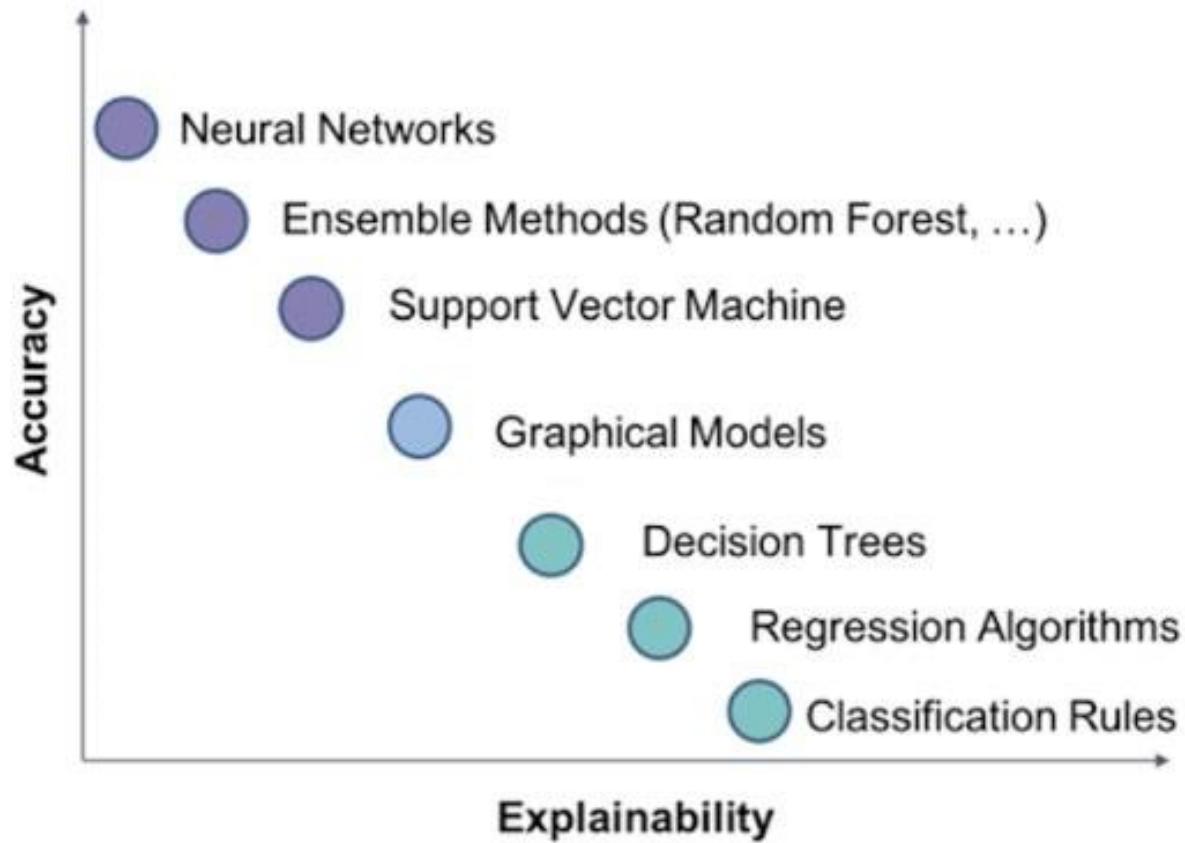


epochs: 100  
learning rate : 0.003  
batch size: 256  
max accuracy: 99.0%

자료: <https://greydanus.github.io/2017/10/30/subspace-nn/>



# 지도학습과 레이어 (6) : 정확도



자료: <https://www.kdnuggets.com/2019/01/explainable-ai.html>

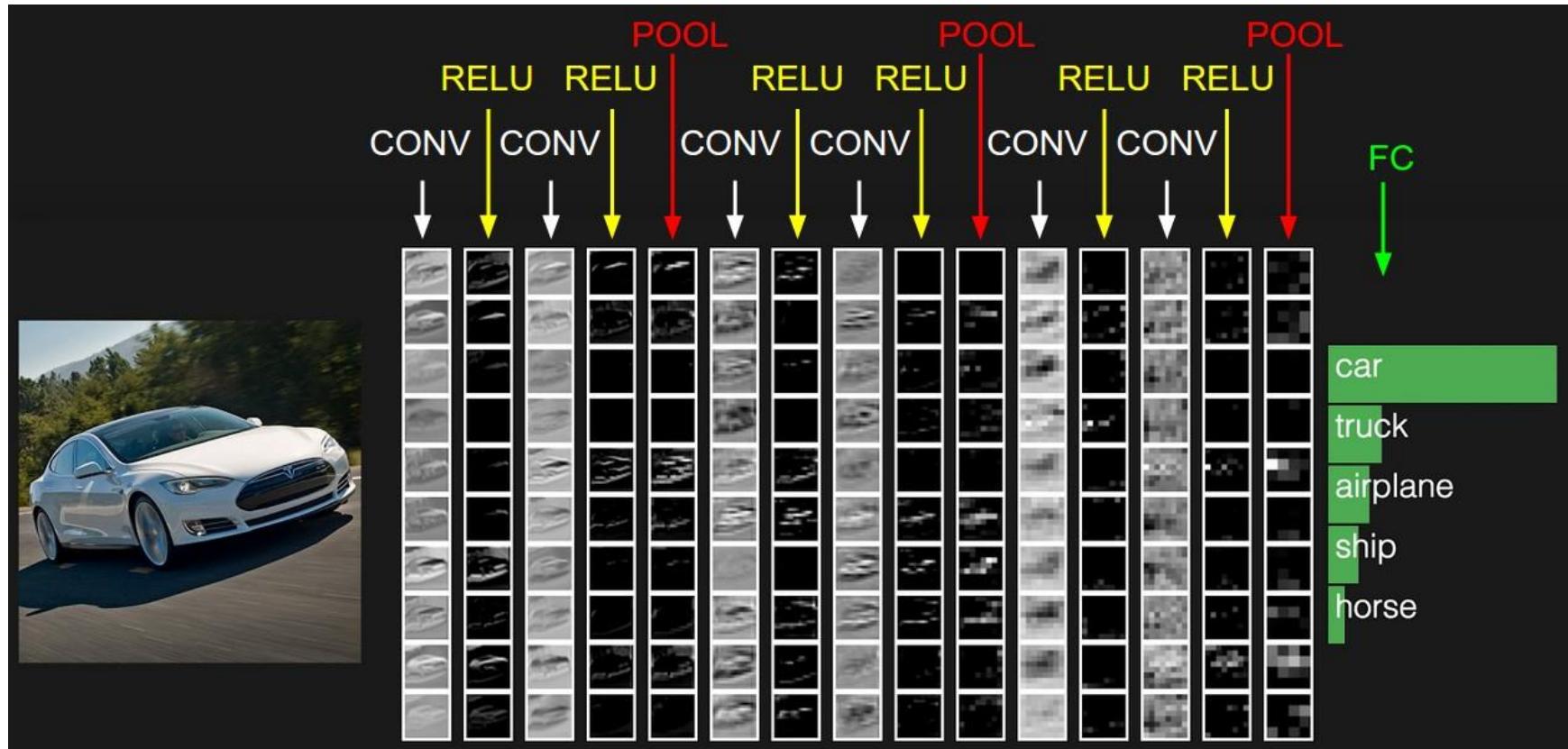
## 04. CNN 기반 지도학습

이홍석 ([hsyi@kisti.re.kr](mailto:hsyi@kisti.re.kr))



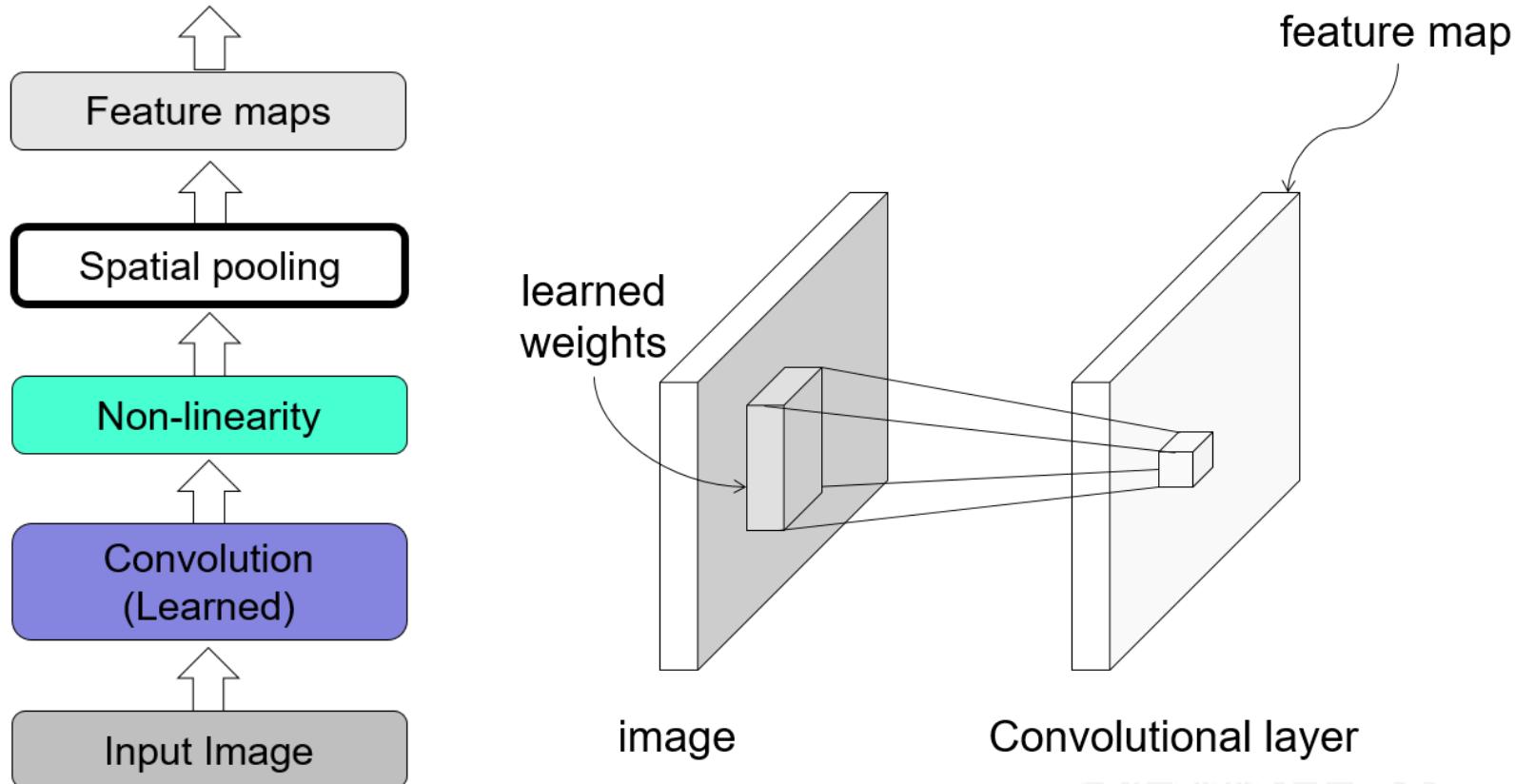


# CNN 소개 (1)



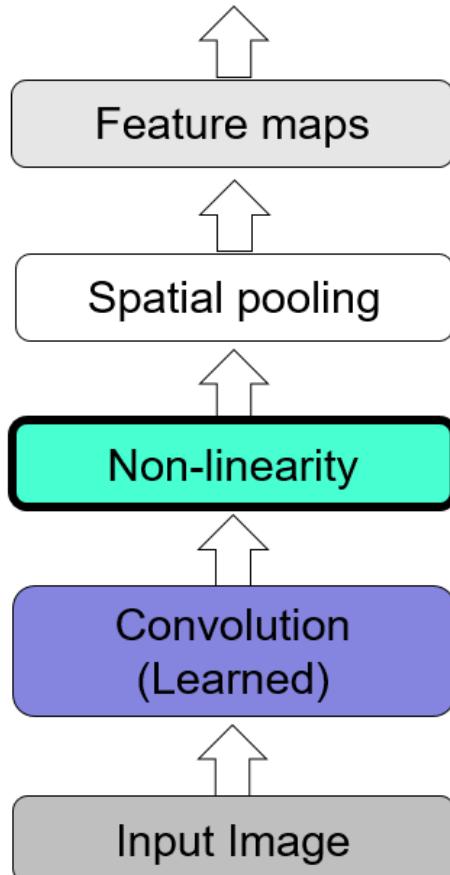


# CNN 소개 (2) : 콘볼루션

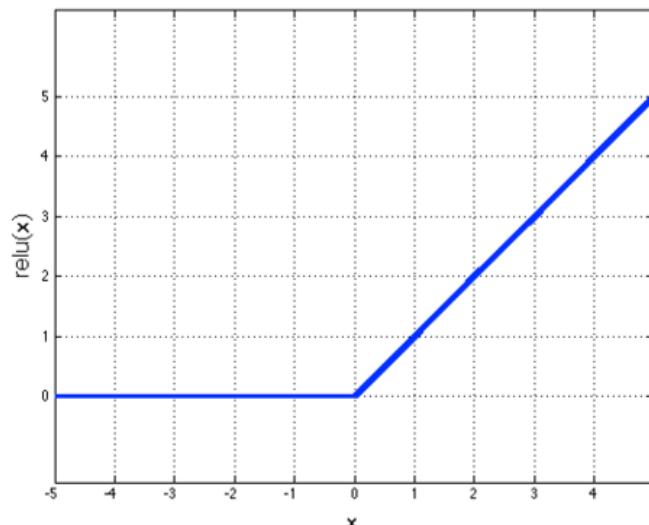




# CNN 소개 (4) : 활성함수 (ReLU)

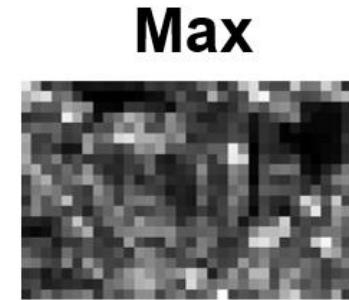
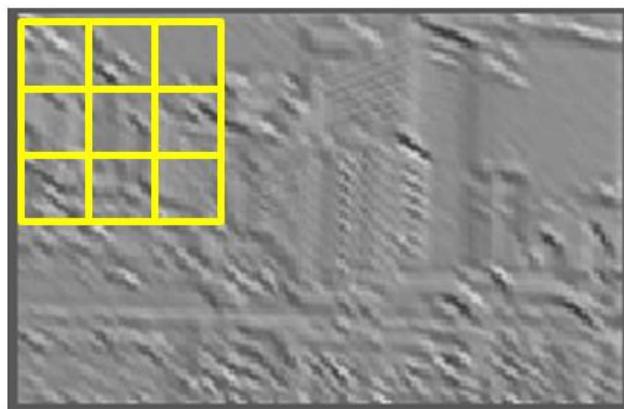
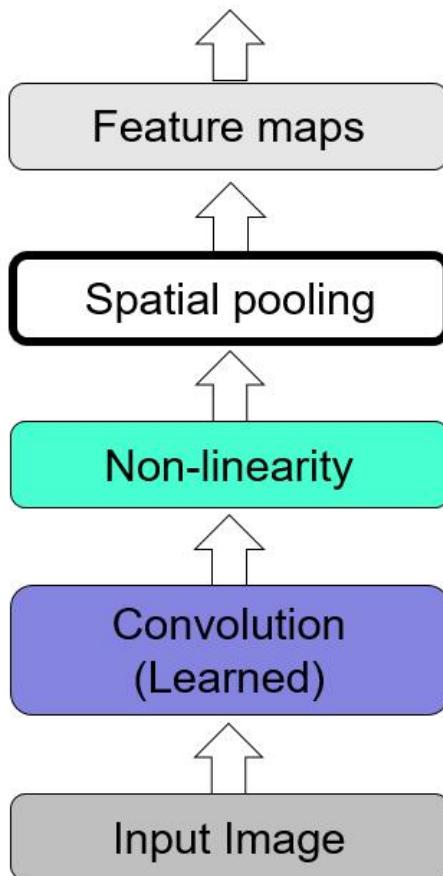


Rectified Linear Unit (ReLU)



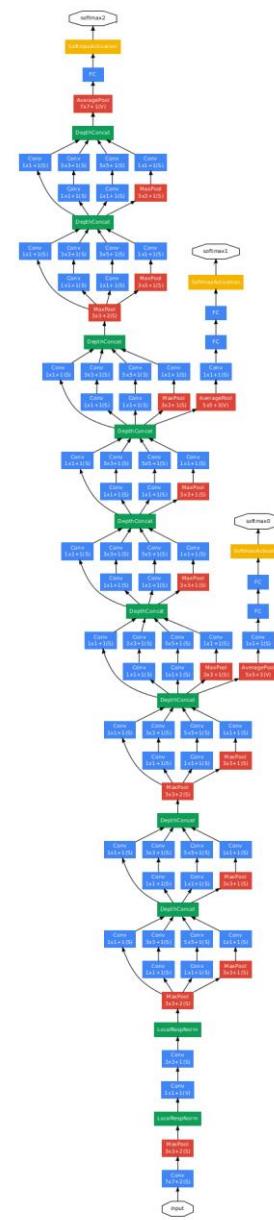


# CNN 소개 (5) : 풀링(pooling)





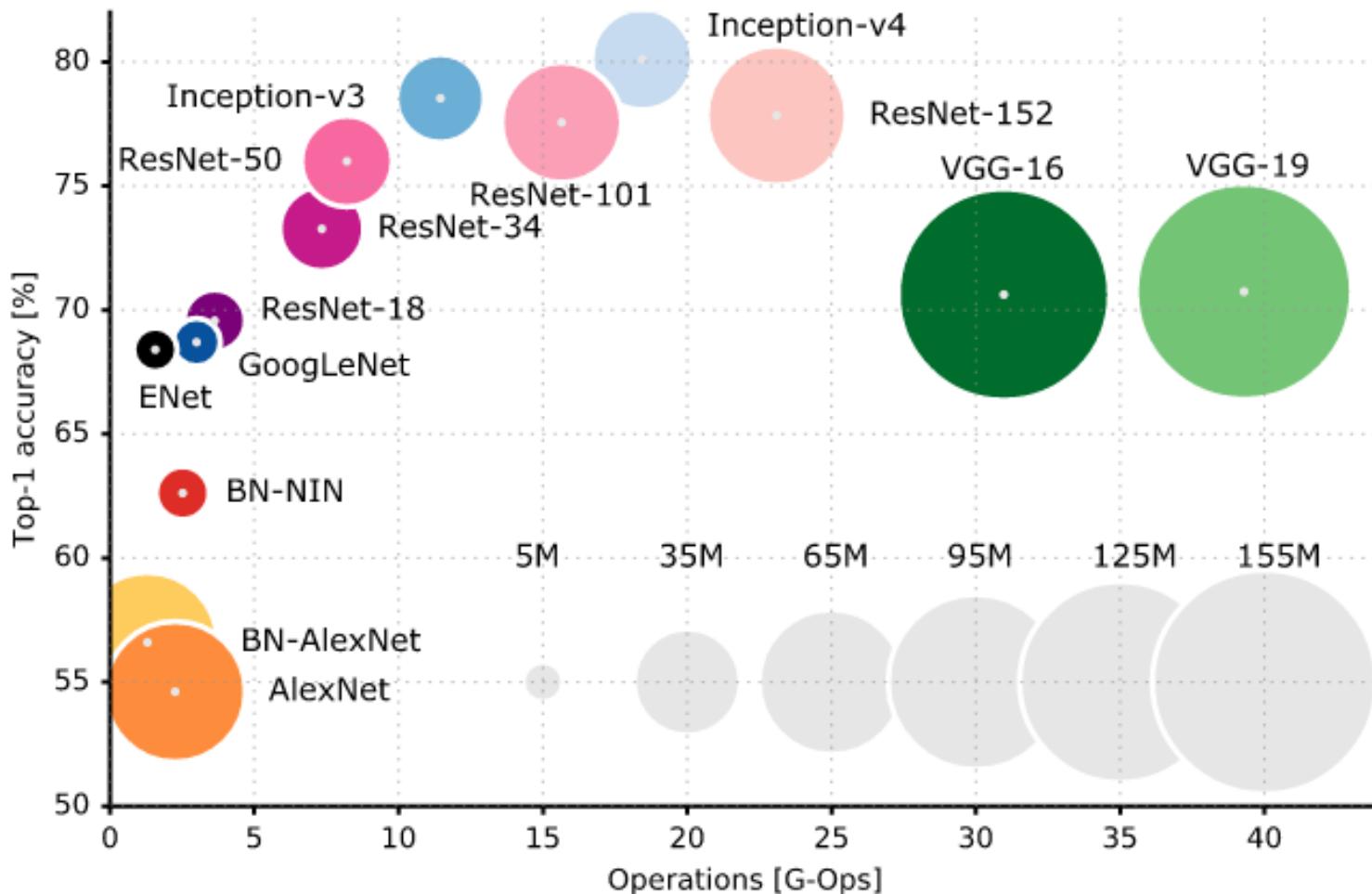
# ImageNet Challenge 2014



# GoogleNet 2015



# CNN Accuracy vs. efficiency



<https://culurciello.github.io/tech/2016/06/04/nets.html>



# Objection Detection (1)

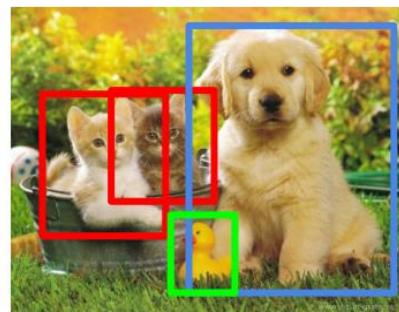
**Classification**



**Classification + Localization**



**Object Detection**



**Instance Segmentation**



CAT

CAT

CAT, DOG, DUCK

CAT, DOG, DUCK

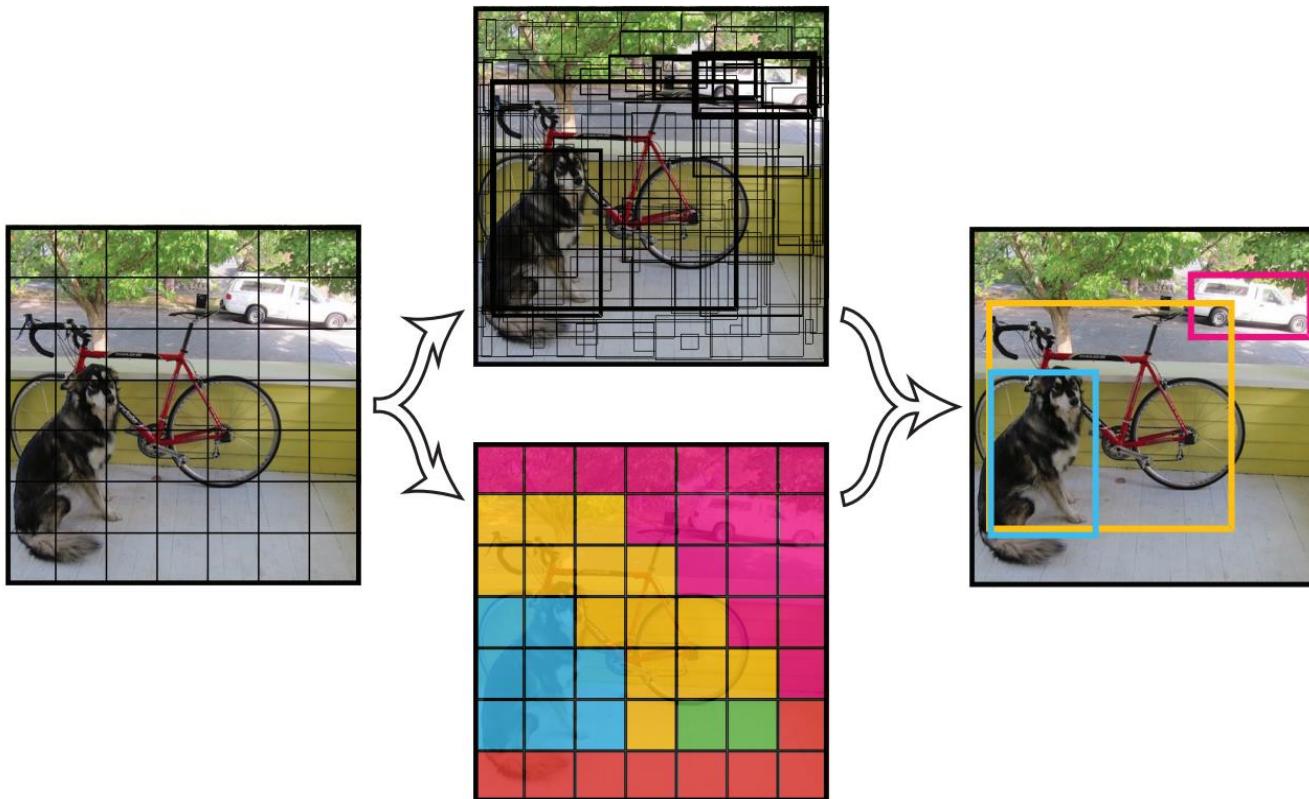
Single object

Multiple objects



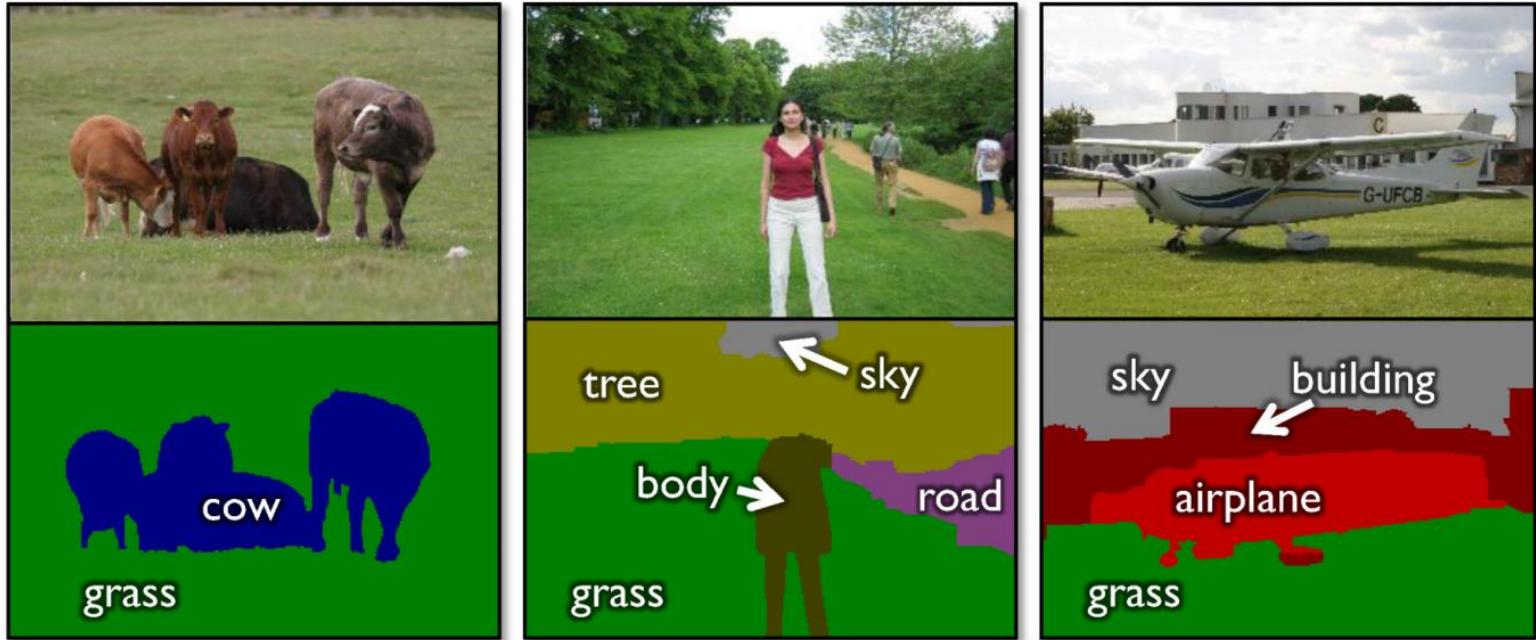
# Objection Detection (2) :YOLO

YOLO (You Only Look Once) : Unified, real-time object detection





# Semantic Segmentation (1)



object classes	building	grass	tree	cow	sheep	sky	airplane	water	face	car
bicycle	flower	sign	bird	book	chair	road	cat	dog	body	boat



# Semantic Segmentation (2)

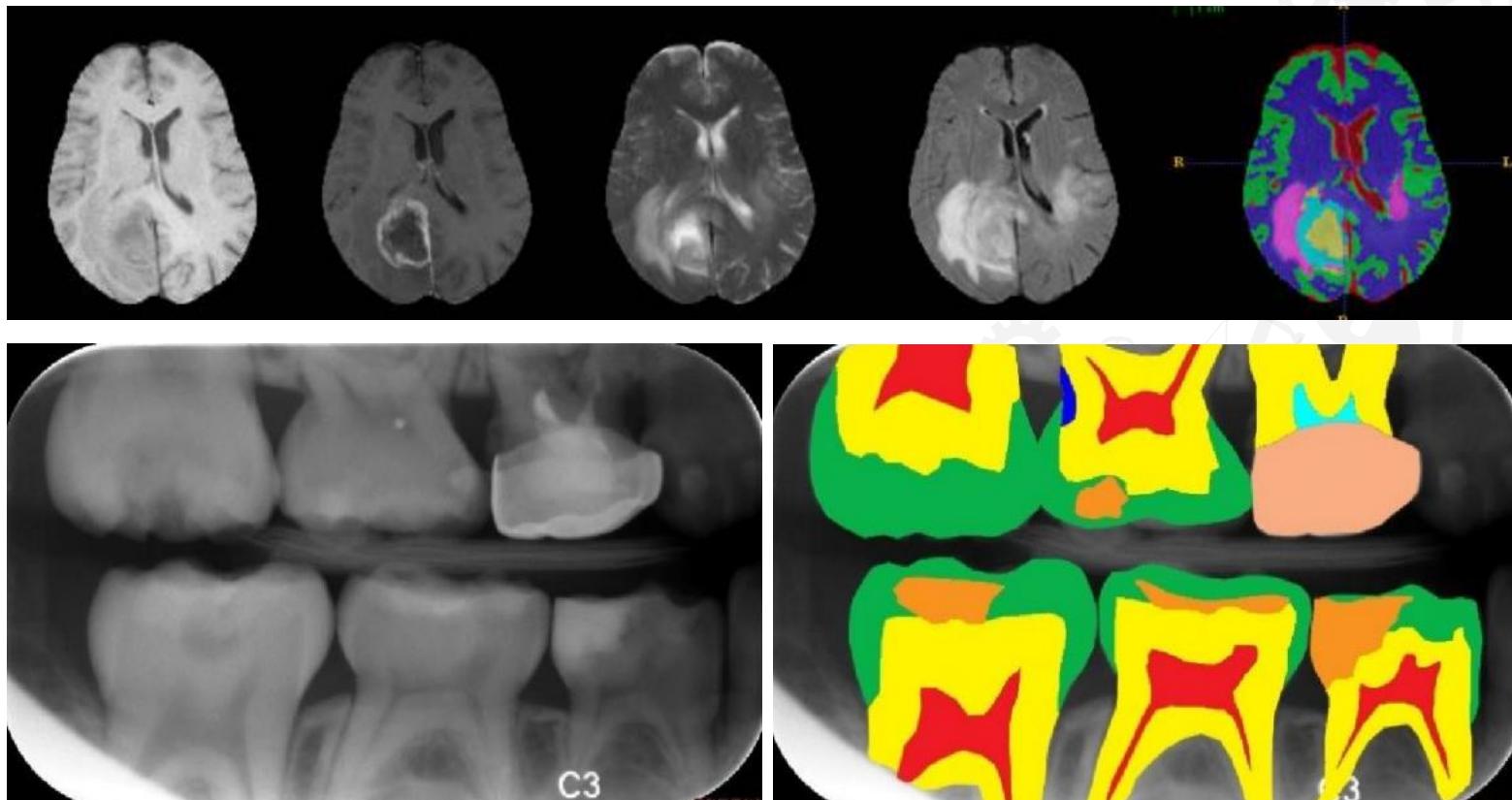
Self-Driving Car와 자율주행 드론에 응용





# Semantic Segmentation (3)

의료 이미지 분석에 응용



## 05. RNN 기반 지도학습

이홍석 ([hsyi@kisti.re.kr](mailto:hsyi@kisti.re.kr))





# 시계열 데이터 : 주식 (3)

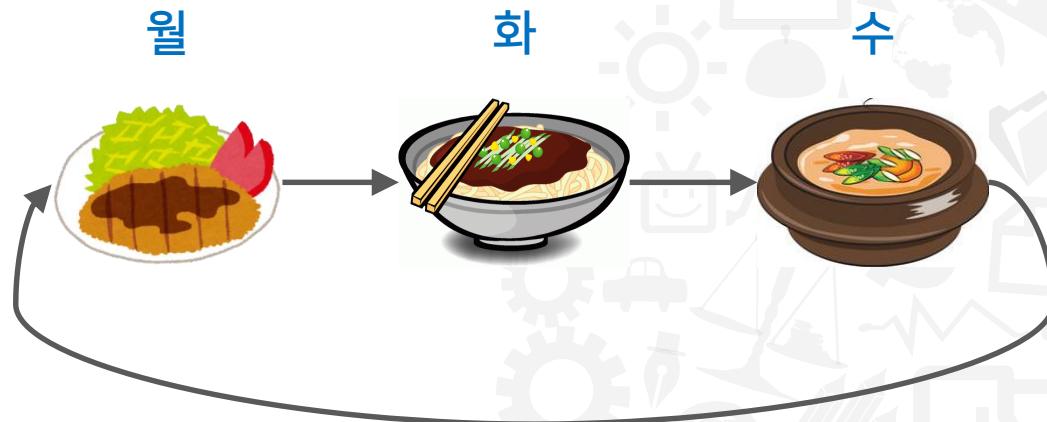
This Is What A Transition From A Bear To A Bull Market Looks Like  
S&P 500 Index (2000-2008)



For illustrative purposes only. Use at own risk.



# 간단한 순환신경망(RNN) (1)



RNN = Recurrent Neural Network



# 간단한 순환신경망(RNN) (2)

날씨에 따라 패턴이 벗어난다

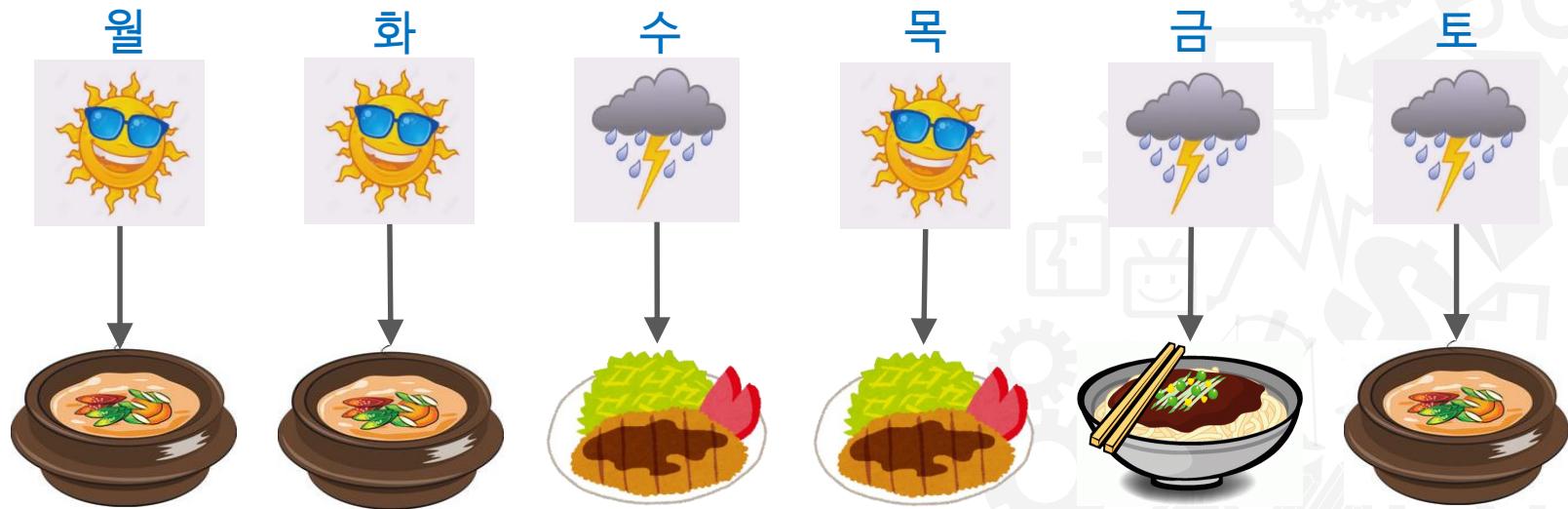


어제 먹었던 음식

원래 RNN 패턴



# 날씨에 따른 순환신경망(RNN) (3)



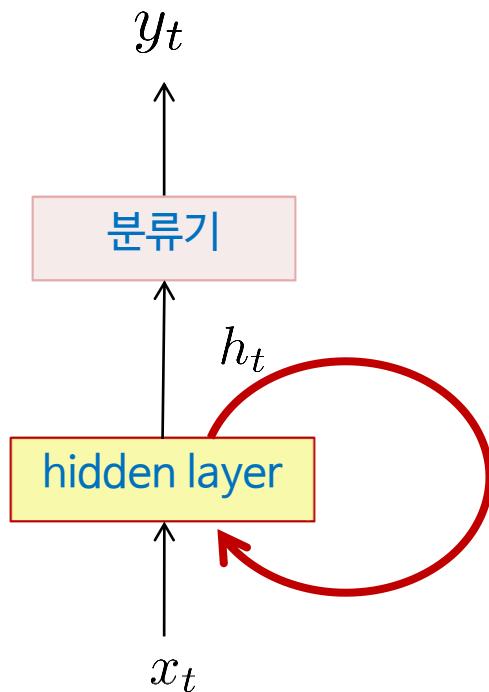


# 순환신경망 (RNN)

출력(t)

은닉층(t)

입력(t)



Recurrence:

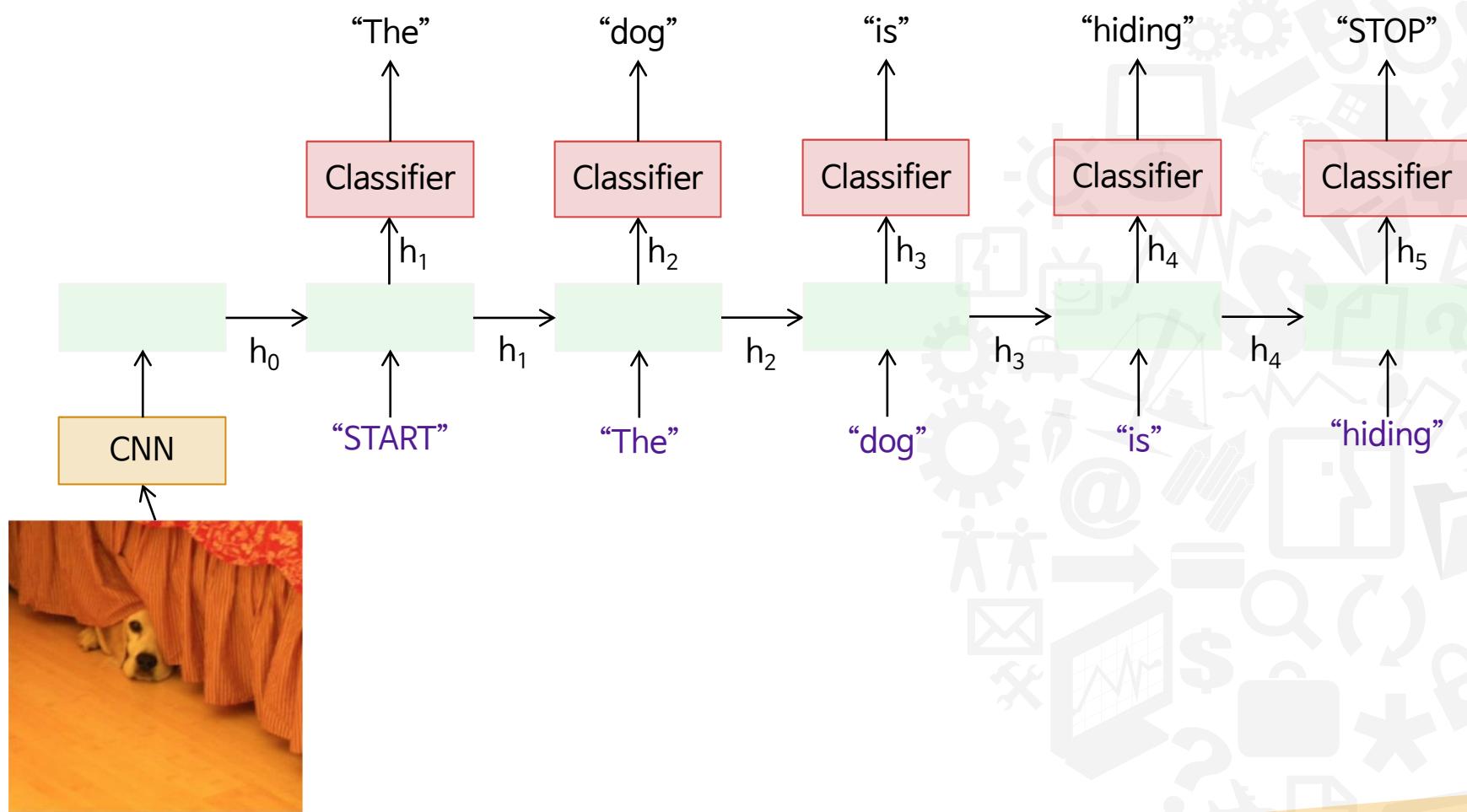
$$h_t = f_W(x_t, h_{t-1})$$

new state      function of W    input at time t    old state



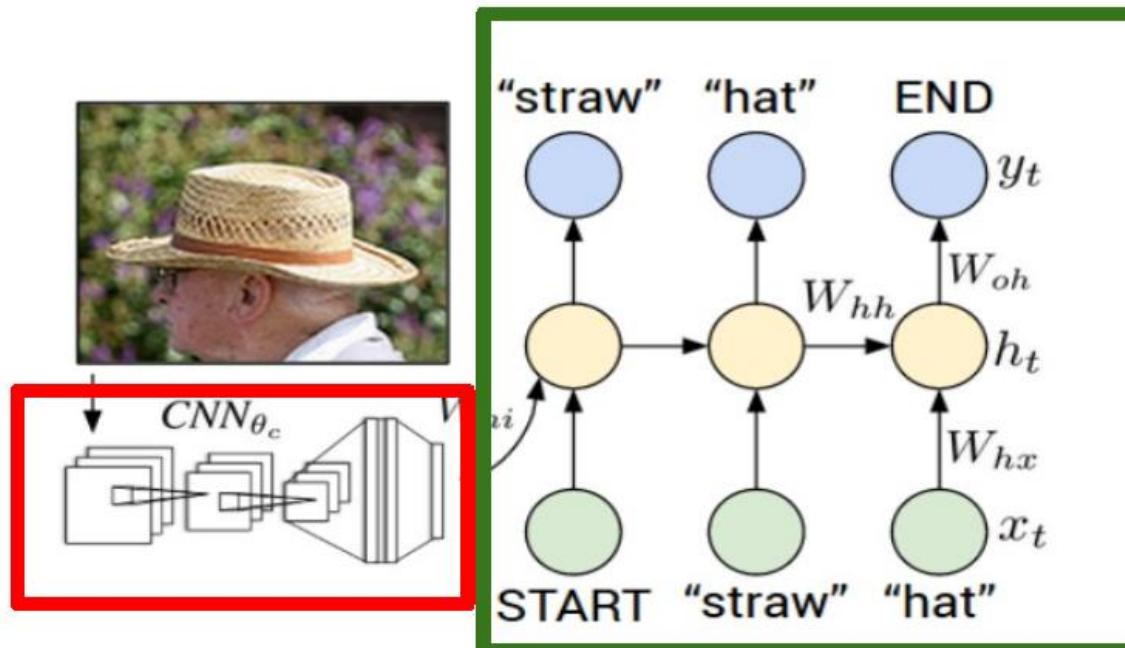
# RNN 응용 : 이미지 캡션 생성

“The dog is hiding”





# RNN 응용: Image Captioning (1)



Recurrent  
Neural  
Network

Convolutional Neural Network

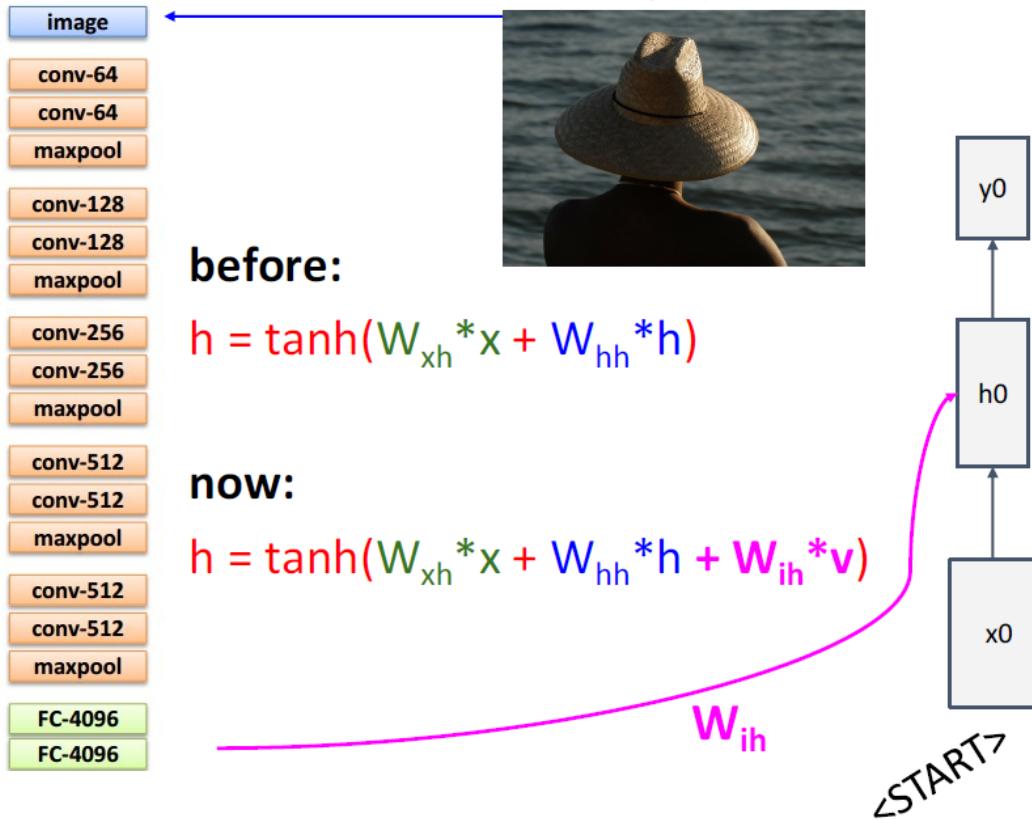


# RNN 응용: Image Captioning (2)



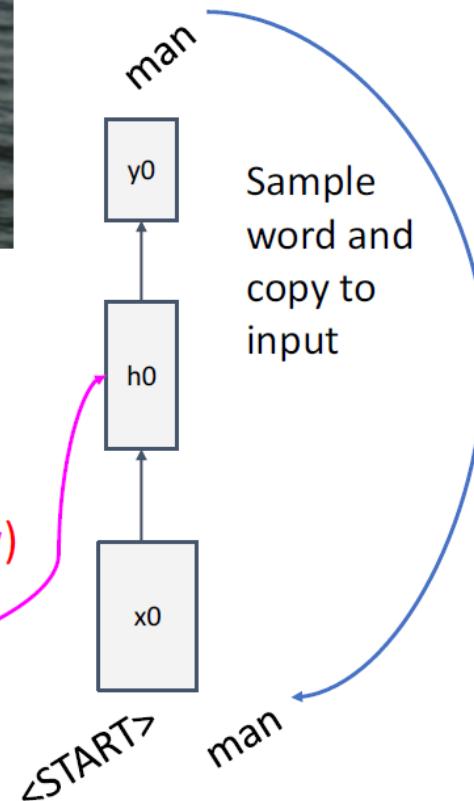
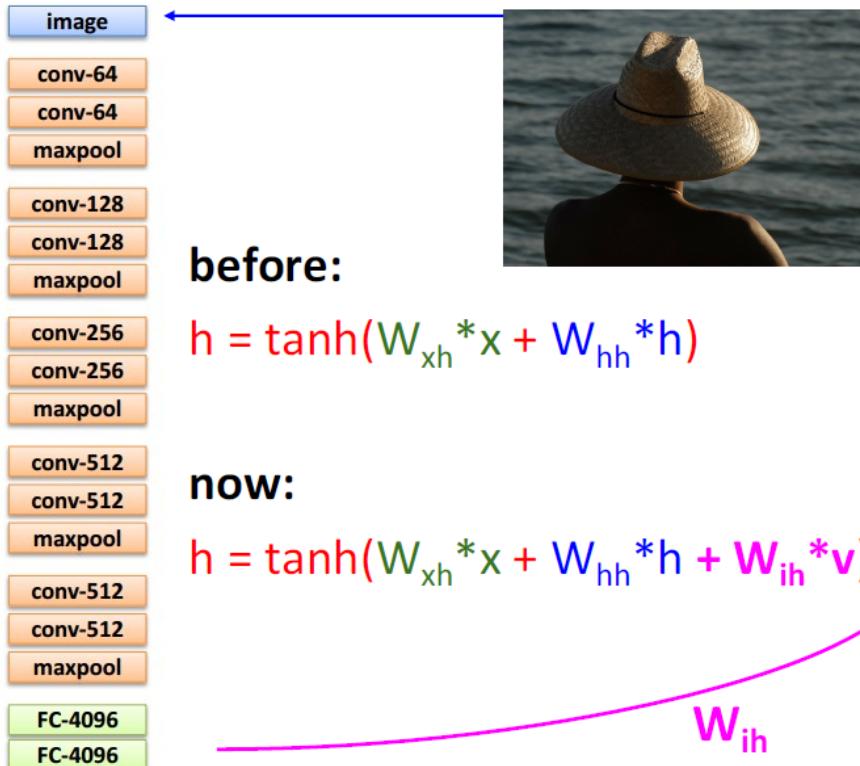


# RNN 응용: Image Captioning (3)





# RNN 응용: Image Captioning (4)





# RNN 응용: Image Captioning (5)



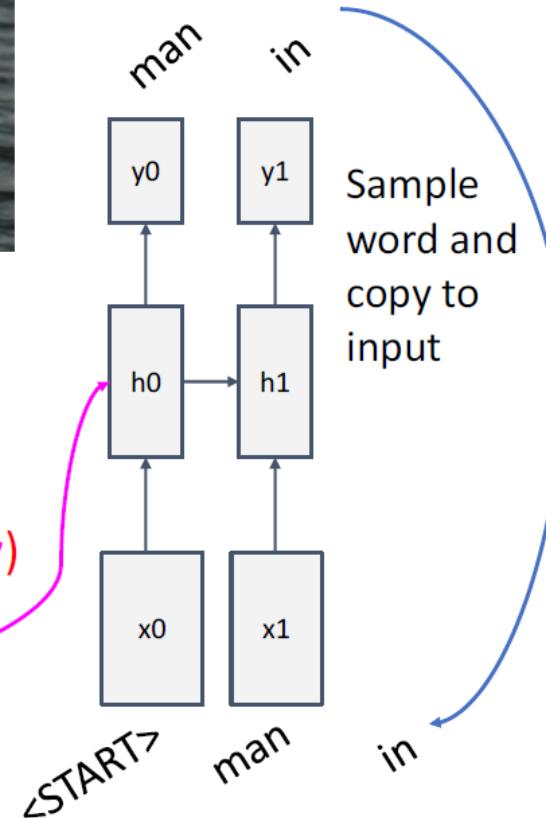
before:

$$h = \tanh(W_{xh} * x + W_{hh} * h)$$

now:

$$h = \tanh(W_{xh} * x + W_{hh} * h + W_{ih} * v)$$

$W_{ih}$





# RNN 응용: Image Captioning (6)



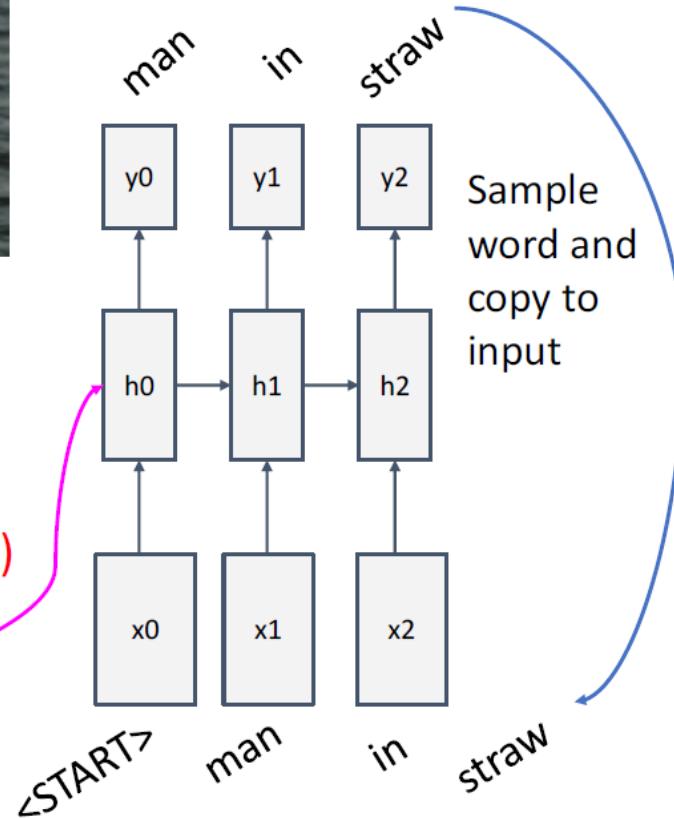
**before:**

$$h = \tanh(W_{xh} * x + W_{hh} * h)$$

**now:**

$$h = \tanh(W_{xh} * x + W_{hh} * h + W_{ih} * v)$$

$W_{ih}$





# RNN 응용: Image Captioning (7)



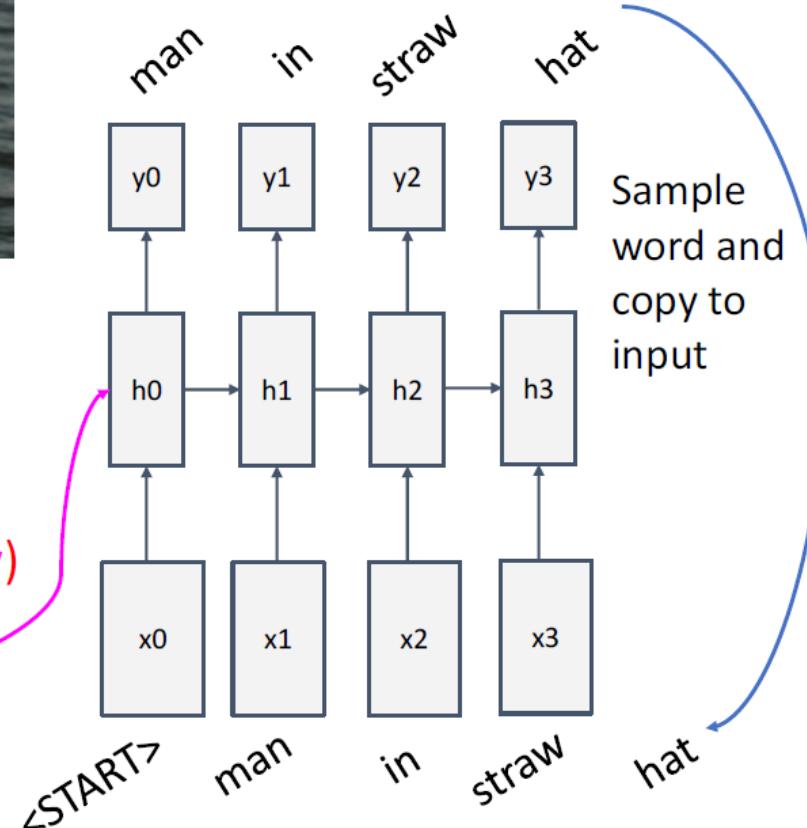
before:

$$h = \tanh(W_{xh} * x + W_{hh} * h)$$

now:

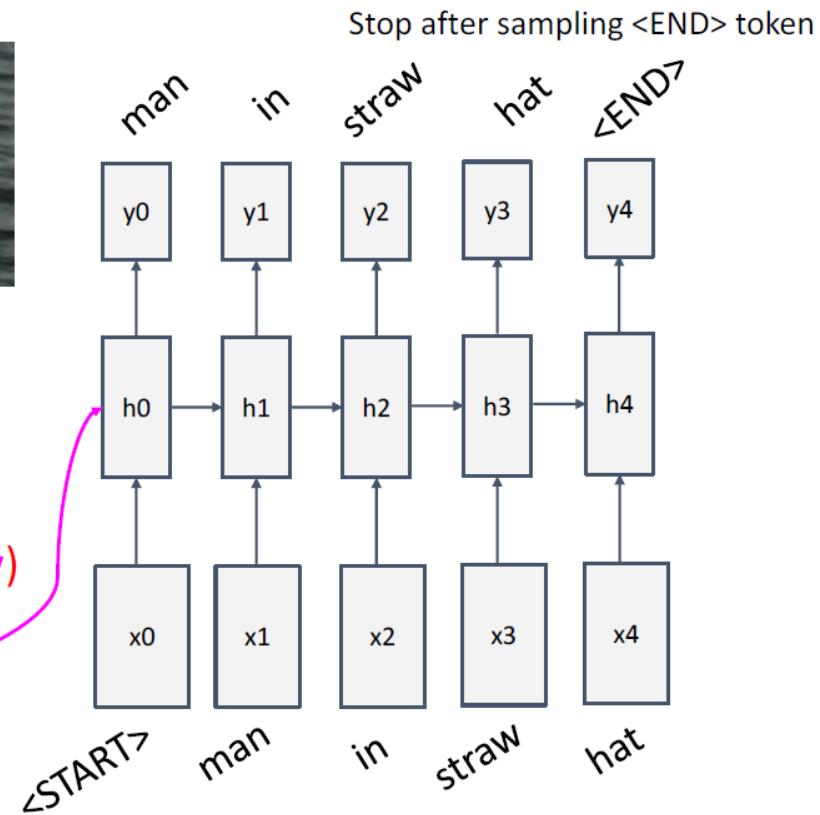
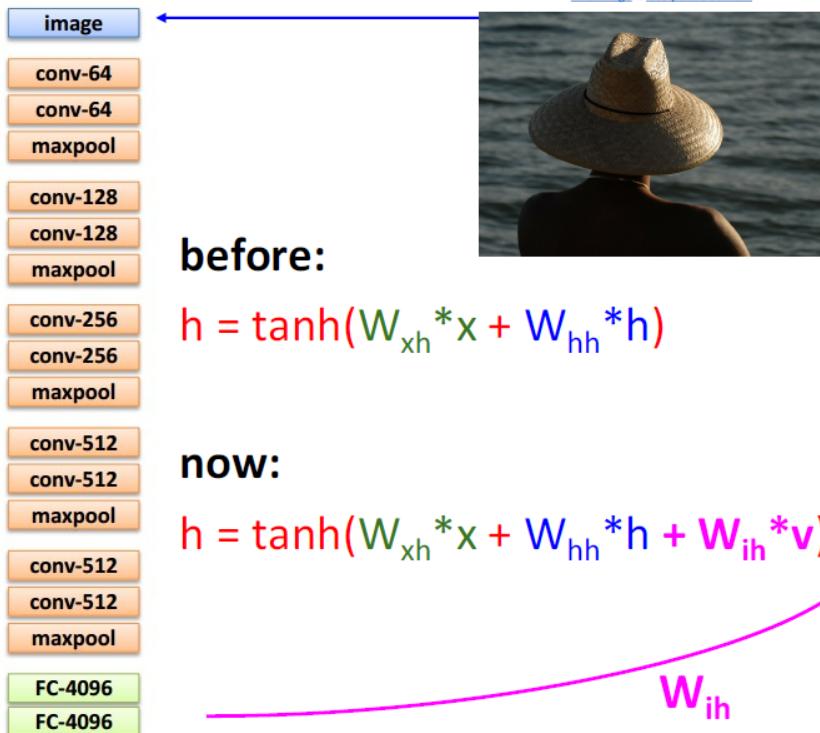
$$h = \tanh(W_{xh} * x + W_{hh} * h + W_{ih} * v)$$

$W_{ih}$





# RNN 응용: Image Captioning (8)





# Image Captioning 성공 사례 (9)

## Image Captioning: Example Results

Captions generated using [neuraltalk2](#)  
All images are [CC0 Public domain](#): [cat](#),  
[suitcase](#), [cat tree](#), [dog](#), [bear](#), [surfers](#),  
[tennis](#), [giraffe](#), [motorcycle](#)



*A cat sitting on a suitcase on the floor*



*A cat is sitting on a tree branch*



*A dog is running in the grass with a frisbee*



*A white teddy bear sitting in the grass*



*Two people walking on the beach with surfboards*



*A tennis player in action on the court*



*Two giraffes standing in a grassy field*



*A man riding a dirt bike on a dirt track*



# Image Captioning 실패 사례 (10)

한 여성은 손에 고양이를 들고 있다.

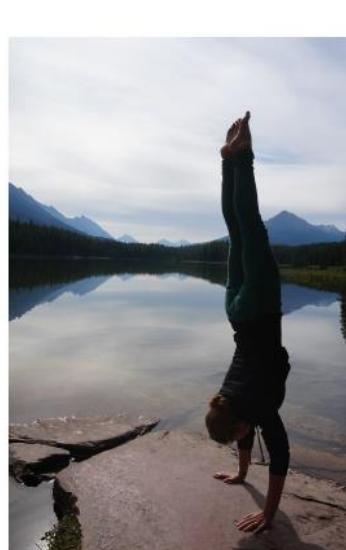


*A woman is holding a cat in her hand*



*A person holding a computer mouse on a desk*

한 사람이 책상에서 마우스를 들고 있다.



*A woman standing on a beach holding a surfboard*

한 여성이 해변에서 서핑보드를 들고 서있다.

새가 나무 가지에 앉았다.



*A bird is perched on a tree branch*

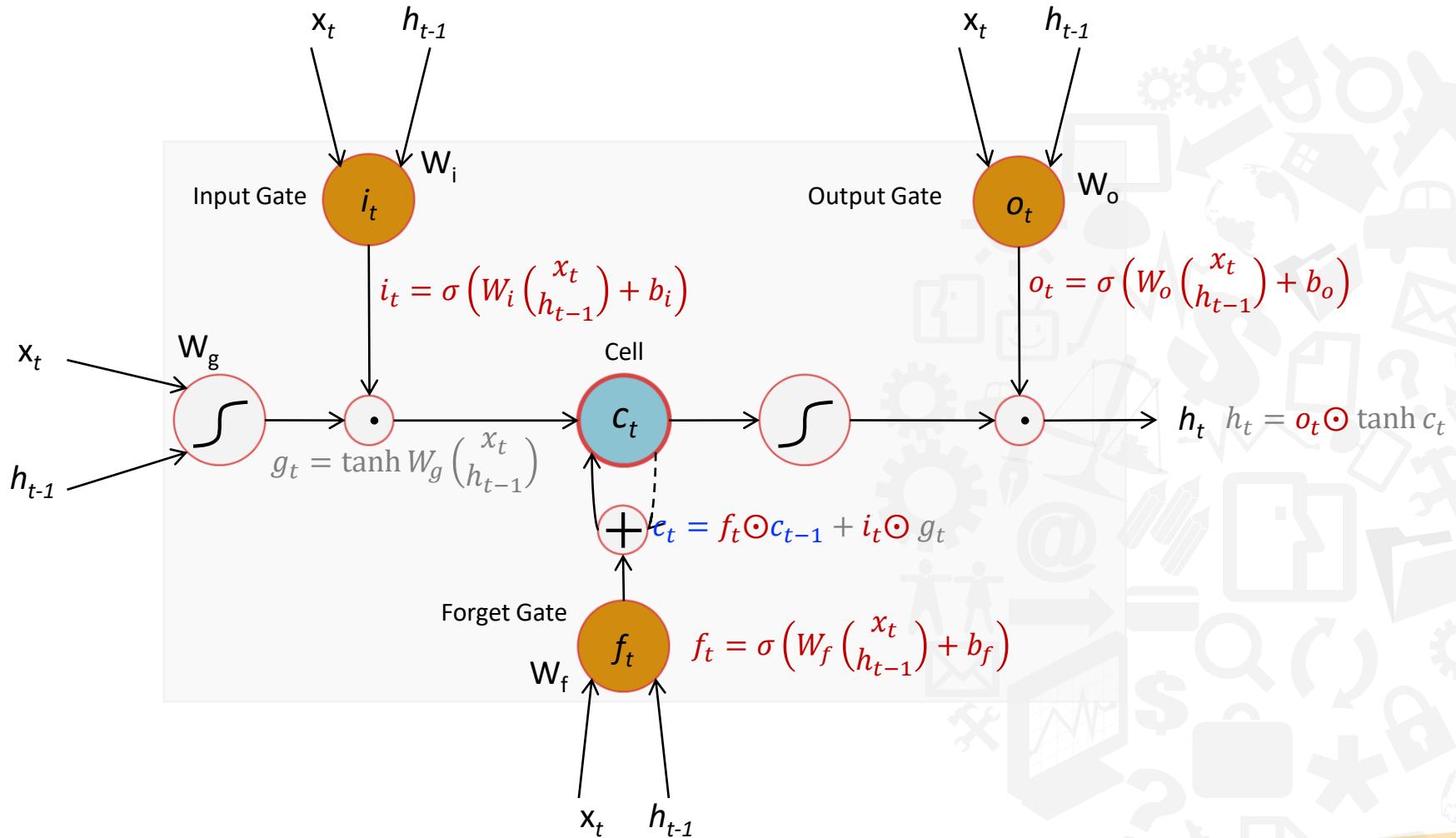


*A man in a baseball uniform throwing a ball*

야구 유니폼을 입은 사람이 공을 던지고 있다.

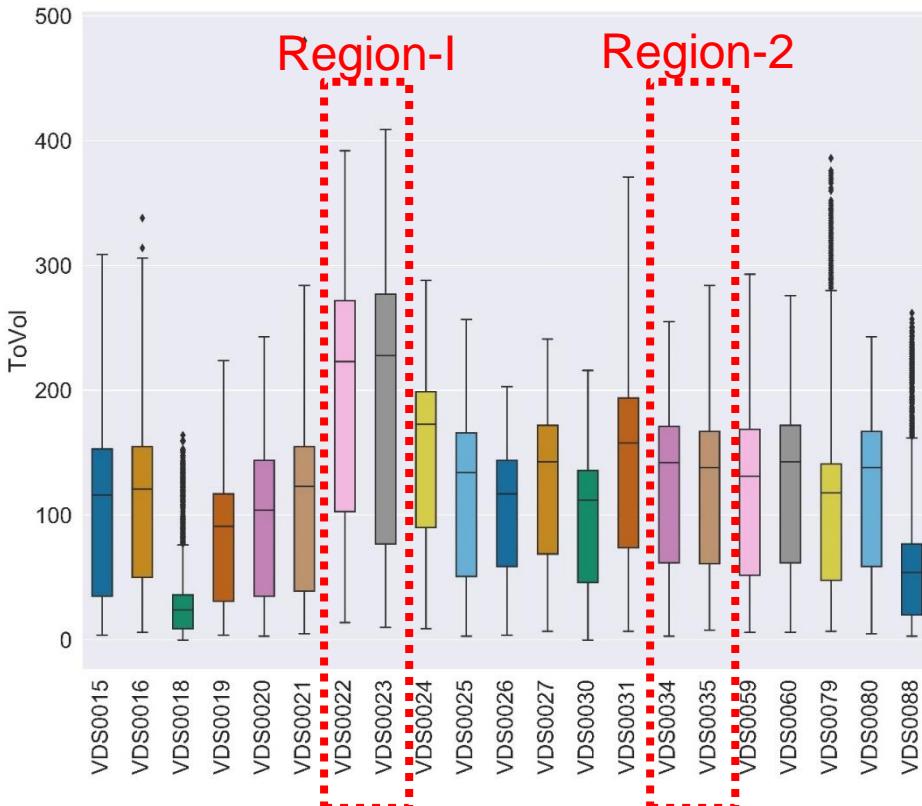


# 장단기 메모리(LSTM) 구조

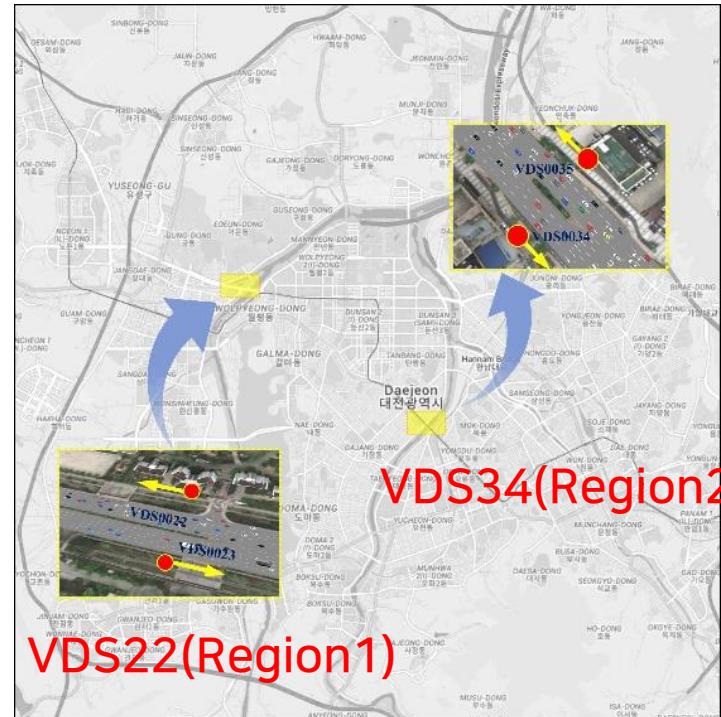




# 장단기 메모리(LSTM) 교통 적용(1)



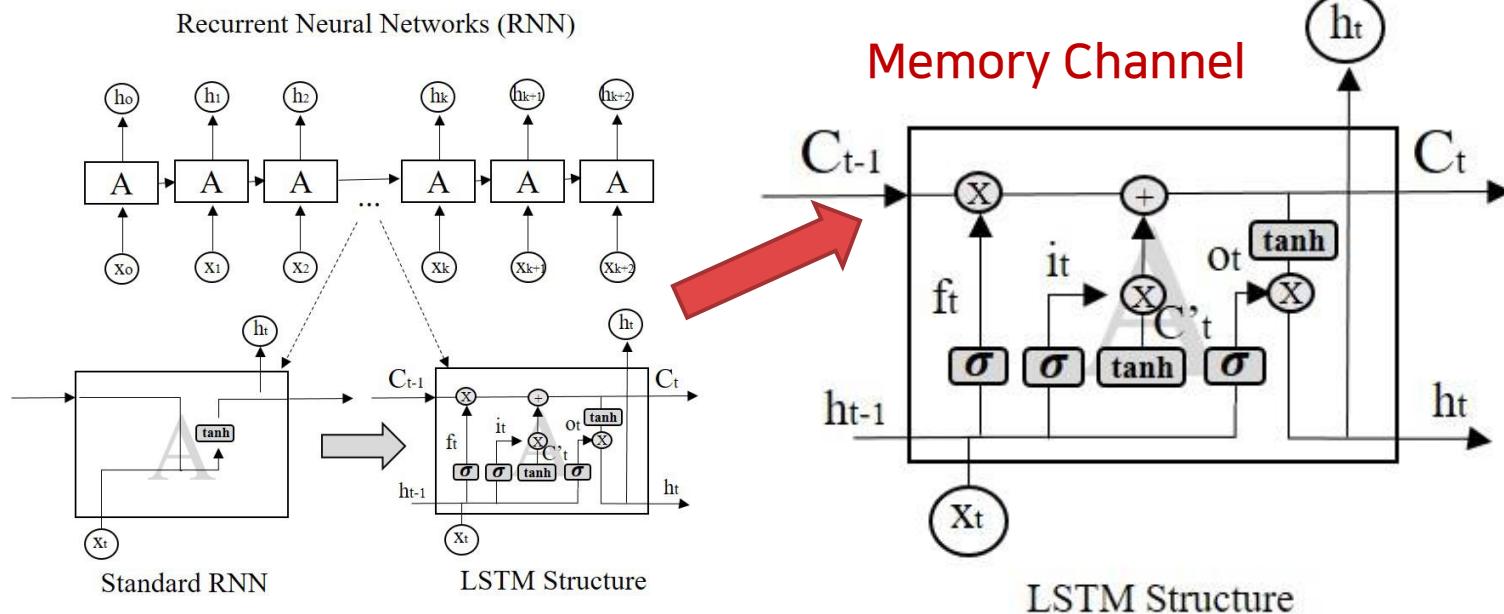
#18 and #22 are the highest and lowest traffic volume in the area, respectively. In this regard, we classify the area into three types of traffic flow such as high (#22), and low (# 18) densities for taking into consideration.





# 장단기메모리(LSTM) 교통 적용(2)

➤ RNN (Recurrent Neural Network) vs LSTM (Long-Short Term Memory) Architecture

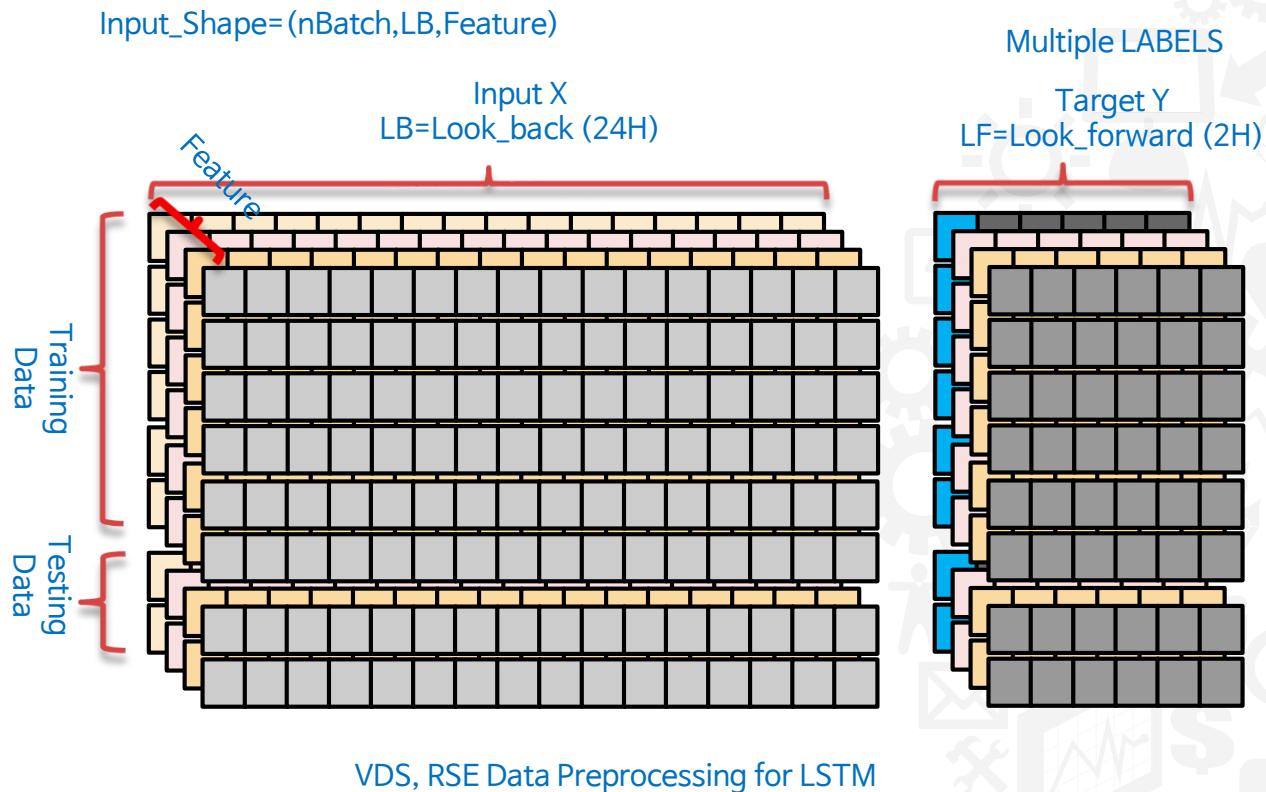


Cf. GRU (Gated Recurrent Unit)



# 장단기예모리(LSTM) 교통 적용(3)

- LSTM Architecture: Multi-feature, Stateful, Multi-dimensional, Stateful

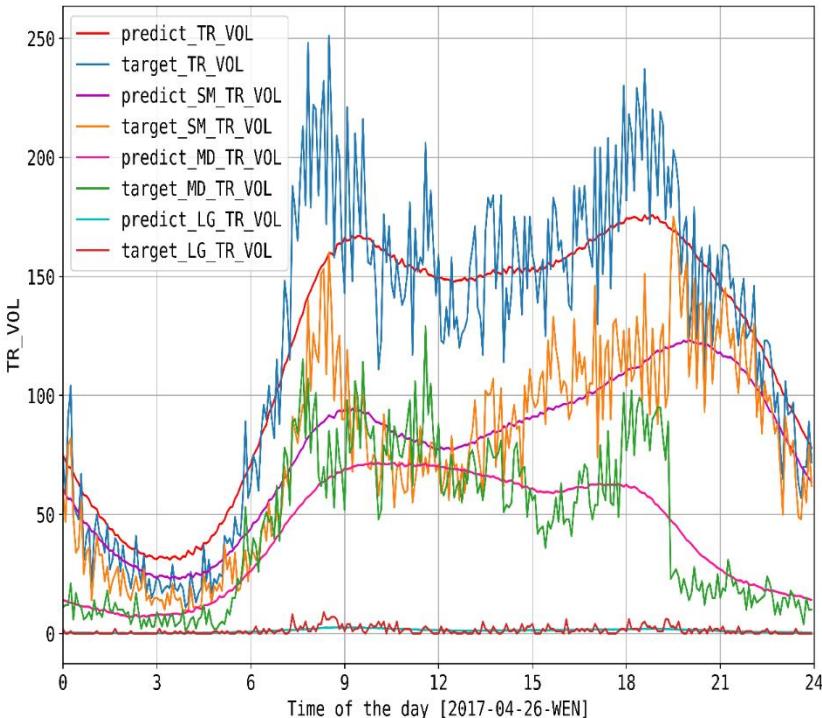




# 장단기예모리(LSTM) 교통 적용(4)

Traffic Volume Count is counting of number of vehicles passing through a road over a period of time.

VDS Data Analysis



VDS_ID	Date	Total Tra.V	SmVol	MeVol	LaVol	Speed	Occupy
VDS0015	201704010700	39	6	33	0	48.9	1.79
VDS0015	201704010705	39	11	27	1	48.3	2.18
VDS0015	201704010710	40	5	34	1	49.8	1.96
VDS0015	201704010715	58	12	46	0	59	2.25
VDS0015	201704010720	27	7	19	1	51.7	1.36
VDS0015	201704010725	46	10	36	0	55.6	1.99
VDS0015	201704010730	52	16	36	0	54	2.37
VDS0015	201704010735	64	22	40	2	47.7	3.03
VDS0015	201704010740	29	7	22	0	48.2	1.37

we classify vehicles into 3 types based on their shape such as Small, Middle, and Large sizes.

## 07. 강화학습의 이해

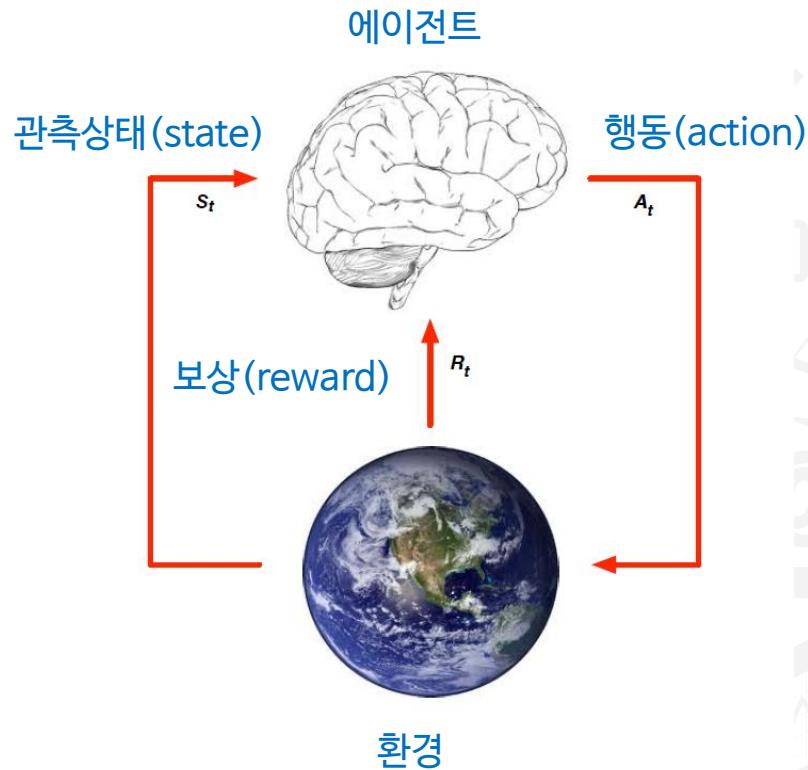
이홍석 ([hsyi@kisti.re.kr](mailto:hsyi@kisti.re.kr))





# 강화학습 개요 (1)

- 에이전트와 환경





# 마르코프 결정 과정 (MDP)

- **마르코프 결정 과정 (Markov Decision Process):** 가장 중요
  - 순간 순간 개념이 필요. 지금 순간과 다음 순간 밖에 없다.
  - 다음 순간( $t+1$ )의 상태는 현재 순간( $t$ )의 상태로 결정된다.

A Markov decision process (MDP) is an *environment* in which all states are **Markov**.

$$\mathbb{P}[S_{t+1} | S_t, A_t = a] = \mathbb{P}[S_{t+1} | S_1, \dots, S_t, A_t = a]$$

## MDP

A *Markov Decision Process* has the following  $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$  is a finite set of states
- $\mathcal{A}$  is a finite set of actions
- $\mathcal{P}$  is a state transition probability matrix,
- $\mathcal{P}_{ss'}^a = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$
- $\mathcal{R}$  is a reward function,  $\mathcal{R}_s^a = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$



# 강화학습이 핵심은 순간 순간

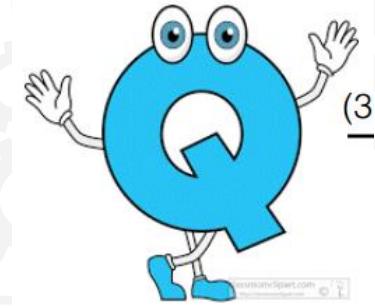
순간 순간 올바른 상황, 관계를 고려해서 행동

Moment to moment, Situation, Relation, Function

無 무

有 유

- ✓ 환경
- ✓ 에이전트
- ✓ 상태
- ✓ 행동
- ✓ 보상

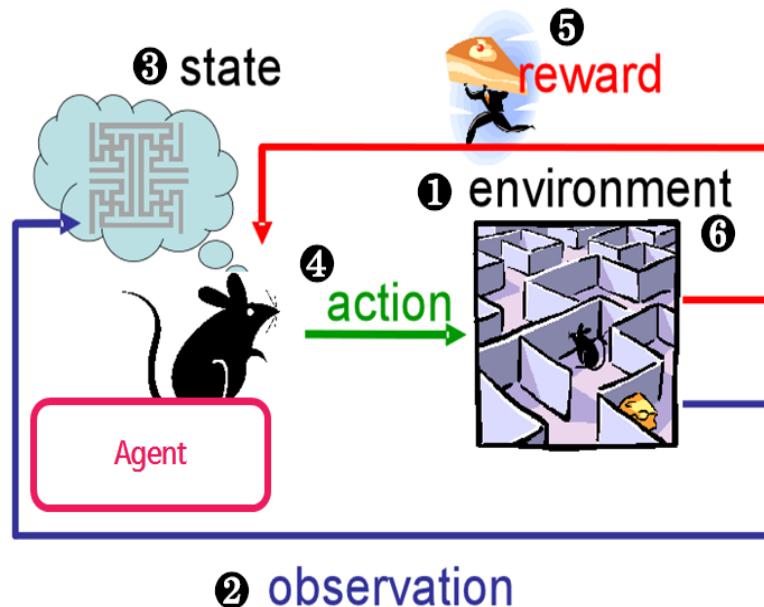
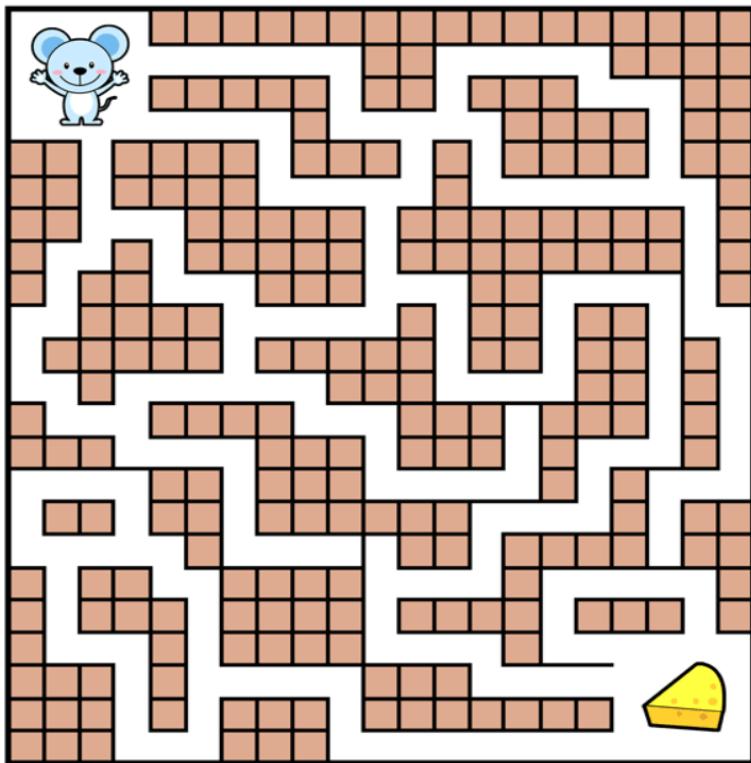


- ✓ 큐(Q) 님?
- ✓ 정책은?



# 에이전트 학습

여기서, 주는 뇌(기억)가 없어서 순간-순간 최선의 행동을 선택한다.





# 큐(Q)에게 상담을 해보자.

큐(Q)는 미로의 구조를 알고 있다고 가정!

바로 직전 행동(t)  
상태(위치, 좌표)

새로운 행동(t+1)을 알려줌

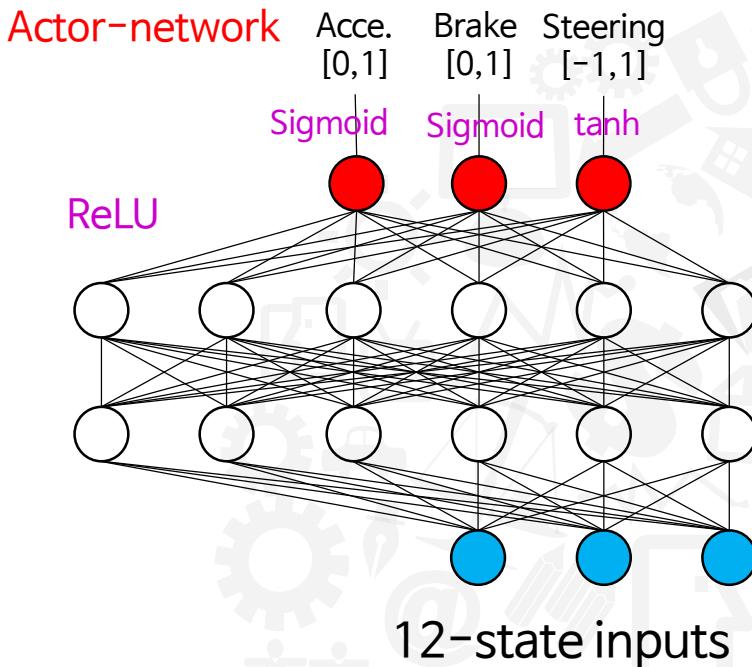
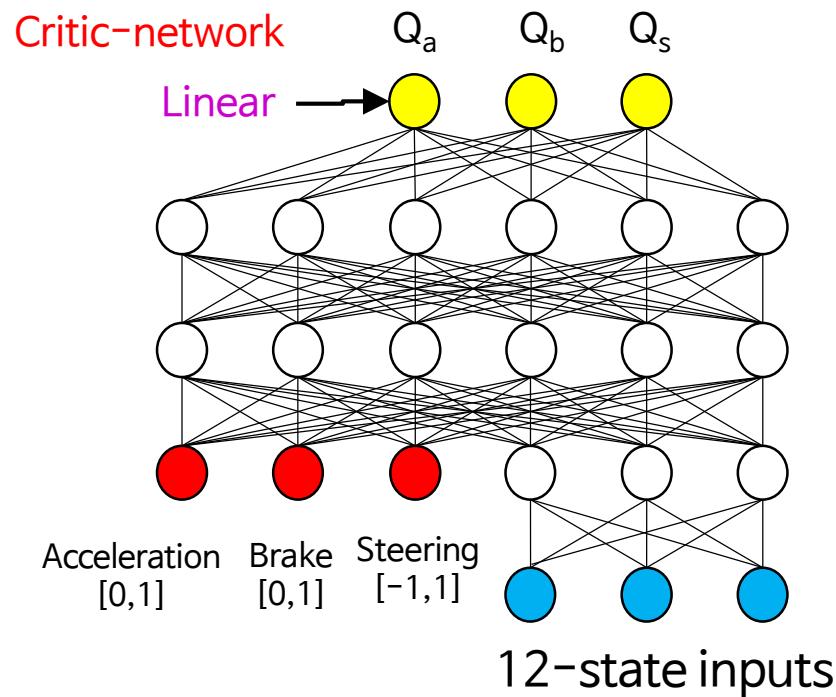


보상을 최대!

$$\hat{Q}(s, a) \leftarrow r + \max_{a'} \hat{Q}(s', a')$$



# Q 값은 신경망으로 구하자



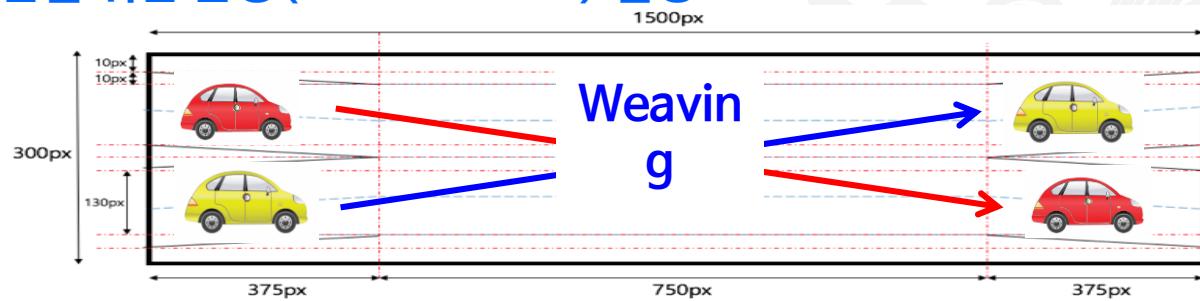


# 강화학습 적용 : 자율주행(1)

Weaving 구간 (카이스트교) : 차선 변경이 빈번히 발생하는 위험도로 구간



강화학습을 위한 환경(Environment) 설정



# Thank You!

[www.ust.ac.kr](http://www.ust.ac.kr)

