

# **신약 개발에 필요한 머신러닝 이해**

**강원대학교**

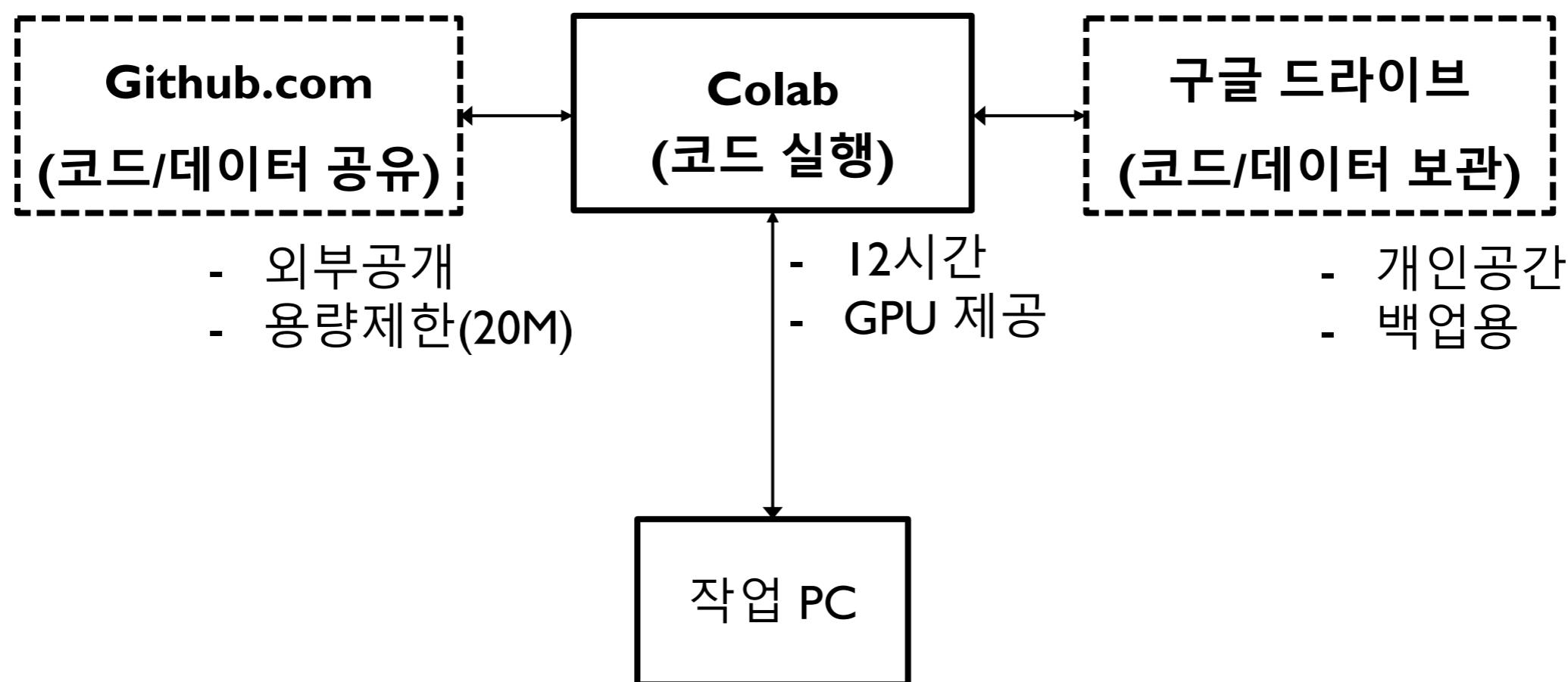
**김화종**

# Contents

---

- ▶ 신약개발과 AI
- ▶ Data Handling
- ▶ Molecule Representation
- ▶ Machine Learning
- ▶ Deep Learning
- ▶ Graph Neural Network
- ▶ Model Optimization
- ▶ Virtual Screening
- ▶ Generative Model

# Colab 실습 환경



# 선수 지식

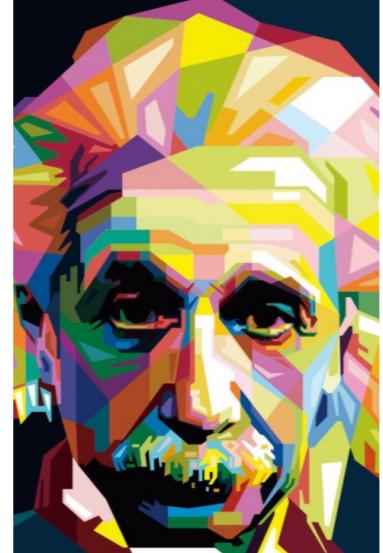
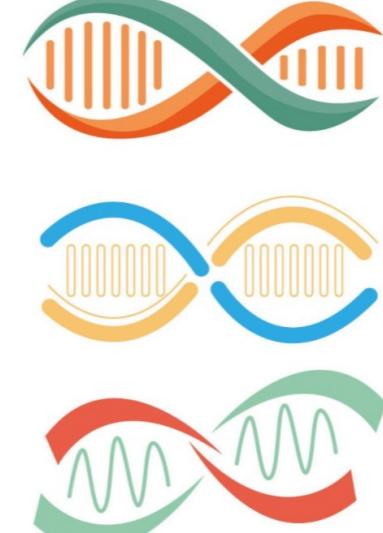
---

- ▶ 파이썬 기초
  - ▶ import
  - ▶ int, float, str
  - ▶ if, else, for, in, def, map, apply
- ▶ 데이터 다루기
  - ▶ list, tuple, dictionary, matplotlib, plot, range
- ▶ 데이터프레임
  - ▶ DataFrame, index, columns, drop, loc, iloc, concat
- ▶ ndarray
  - ▶ arange, reshape, concatenate

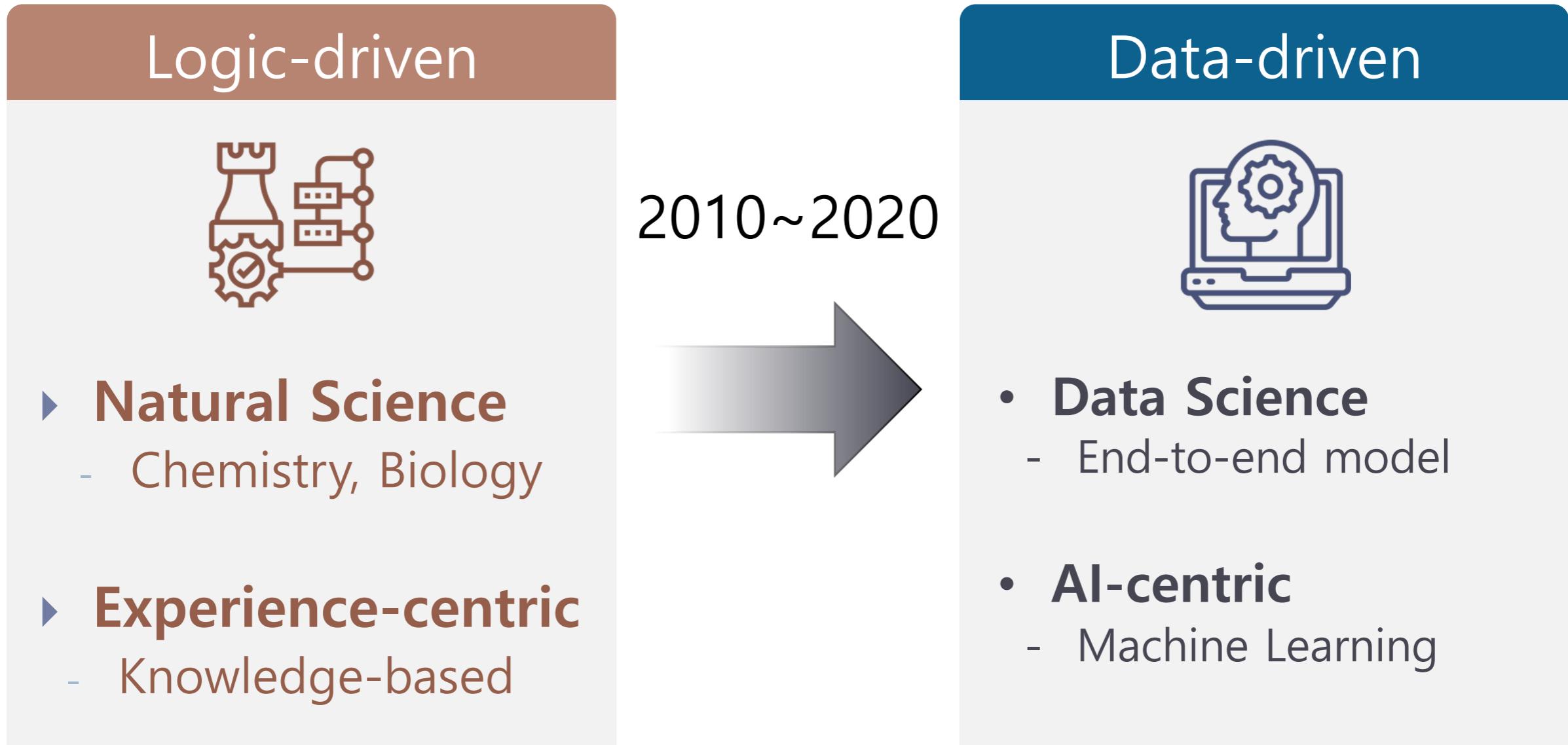
# 신약개발과 AI

# Epoch of Science

---

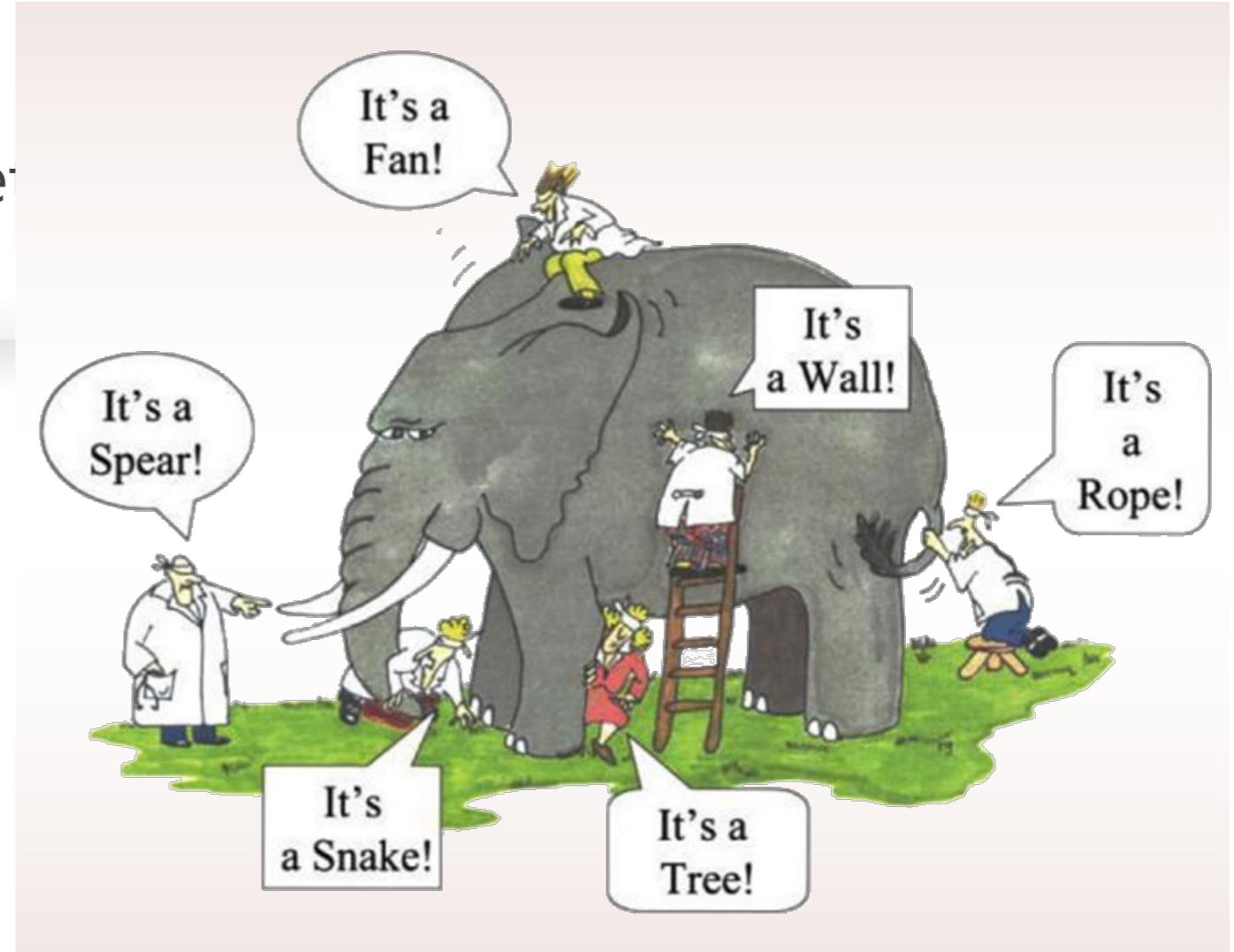
1870	1900	1950	1990	2020
Chemistry	Physics	Computing	Biology	<b>Digital Biology</b>
				

# Science Paradigm Shift



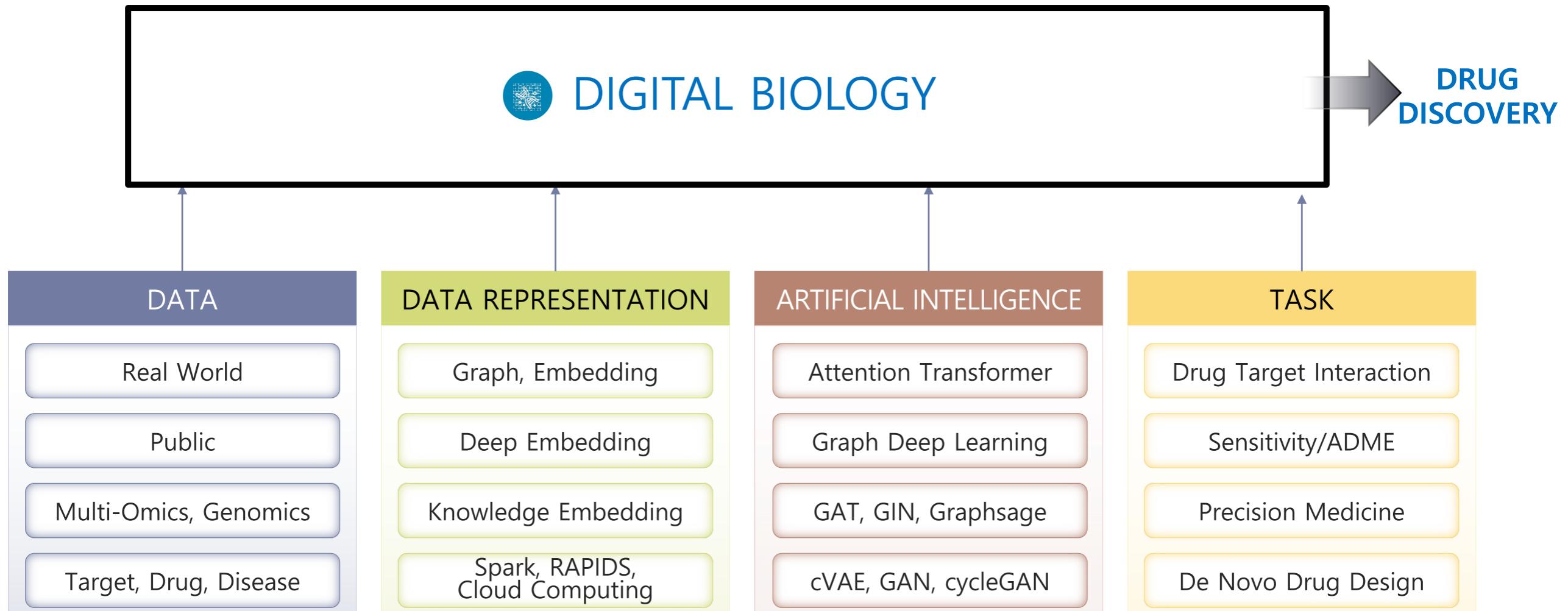
# Bio Data Issues

- **Phenotype data**
  - variant, noisy, hard to interpret
- **Hard to understand**
  - Needs end-to-end models
- **Real world data sharing**
  - Privacy
- **Multimodality**



(Image: Daily Fintech)

# Digital Biology



# AI for Drug Discovery

- Reduces risk, time, and money
  - Improves efficacy and safety
- Generates novel drug candidates
  - De novo drug design
  - New data types hugely accelerate progress
    - multi-omics, personal immune profile, lifestyle data, etc.



(Source. Frost & Sullivan)

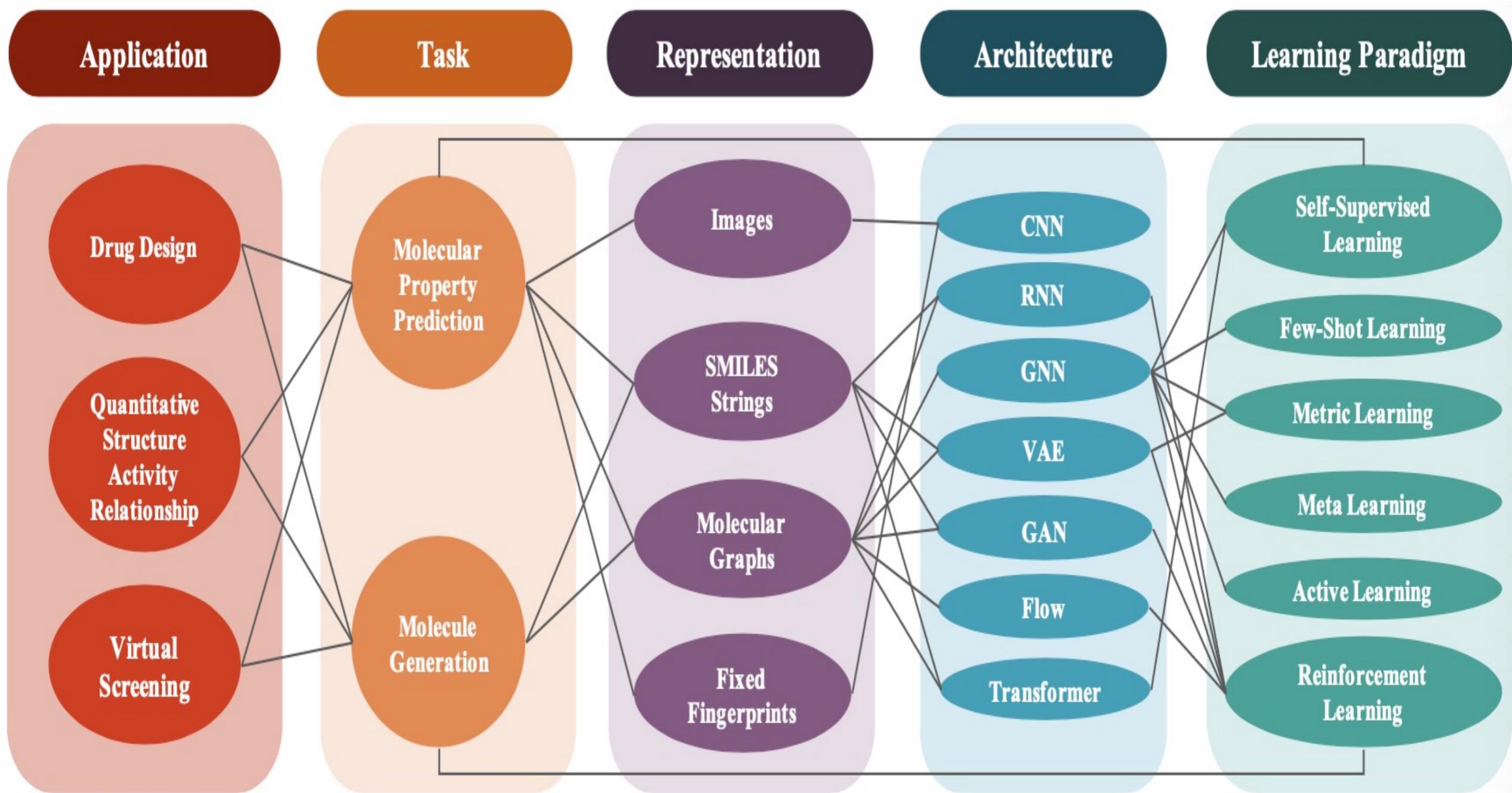
Just getting started!

# AI in Drug Discovery

---

- ▶ Drug Discovery Timeline을 줄일 수 있다
  - ▶ target identification
  - ▶ virtual screening
  - ▶ de novo drug design
  - ▶ drug repositioning
- ▶ Drug의 효능(efficacy) 및 안전성(safety) 예측 정확도 향상
  - ▶ ADME/T prediction
- ▶ Drug Discovery 파이프라인 다양화
  - ▶ Fast setup, fast failure
- ▶ Types:
  - **Predictive task**
  - **Generative task**

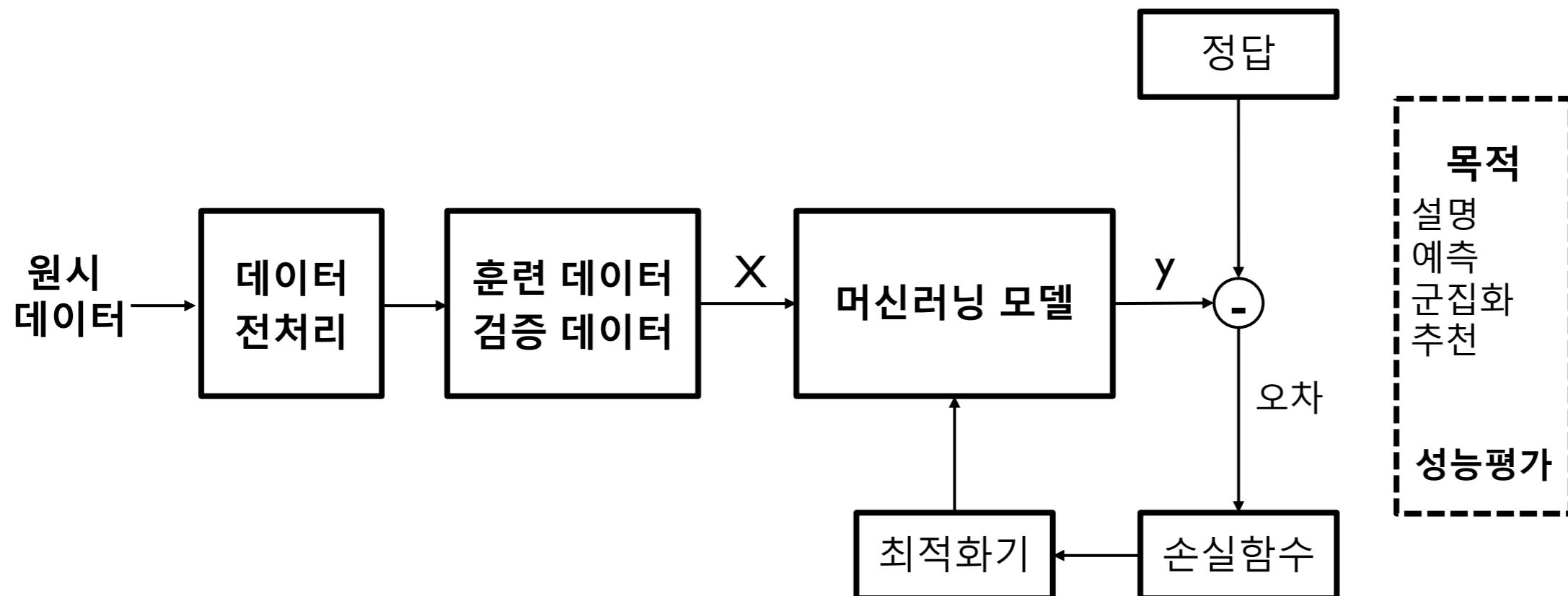
# AI Technologies



# 머신러닝 이해

# 머신러닝 모델

- ▶ 수식과 논리가 아니라 “데이터 기반”的 모델을 사용한다
- ▶ 훈련/검증 데이터
  - ▶ 모델을 만드는 데는 훈련 데이터를, 성능을 검증하는 데는 검증 데이터를 사용한다

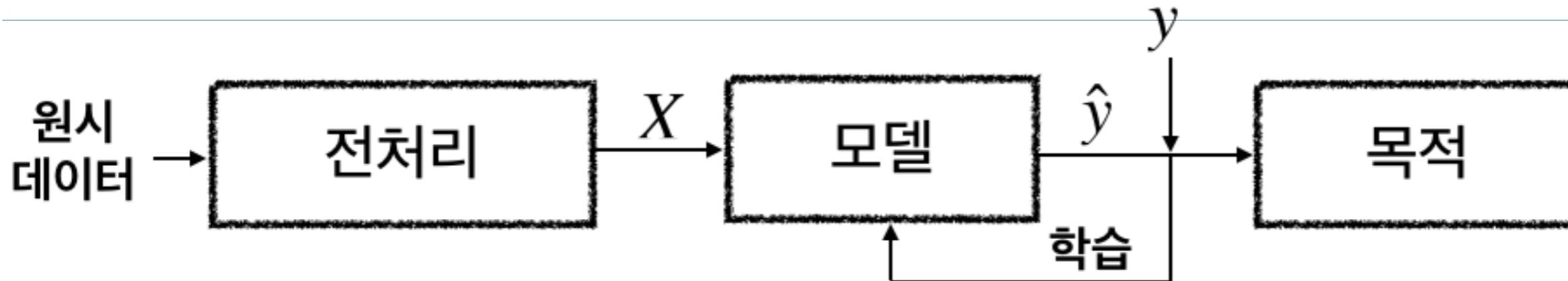


# 손실함수와 성능지표

---

- ▶ **손실함수**를 사용하는 목적은 모델이 얼마나 잘 훈련되는지를 측정하기 위해서이다.
  - ▶ 모델은 손실함수를 최소화 하는 방향으로 학습한다.
- ▶ **모델의 성능지표**는 모델이 궁극적으로 얼마나 잘 동작하는지를 평가하는 척도이다.
  - ▶ 성능이 높은 모델을 만드는 것이 목적이다

# 머신러닝 프로세스



결측치 처리  
오류값 처리  
스케일링  
데이터 변환  
· 카테고리 변환  
· 로그, 역수 변환  
특성공학  
· 차원축소  
· PCA

선형모델  
로지스틱회귀  
**SVM**  
결정트리  
랜덤포레스트  
그라디언트부스트  
**kNN, Bayes**  
**CNN**  
**RNN**

클러스터링  
설명적 분석  
. EDA, 시각화  
회귀 예측  
분류 예측  
추천

# 머신러닝 모델 특징

머신러닝 유형	알고리즘	특징	
지도학습 (예측모델)	선형 계열	선형 모델, SVM 로지스틱회귀	곱셈과 덧셈으로 점수를 구하고 이를 이용하여 회귀와 분류 예측
	신경망	MLP, CNN, RNN, Transformer	매트릭스 연산을 기반으로 점수를 계산하며 활성화 함수 도입
	트리 계열	결정 트리, 랜덤포레스트, 그라디언트부스팅	True/False 선택을 반복하여 회귀와 분류 예측 수행. 스케일링이 필요없다
	기타	kNN, 베이즈	특성 공간상의 거리를 기준, 또는 조건부 확률을 기준으로 예측
비지도학습 (데이터 처리)	클러스터링	k-means, DBSCAN	특성 공간상 거리와 유사도를 기준으로 샘플을 그루핑
	데이터 변환	스케일링, 로그변환, 카테고리 인코딩	효과적인 데이터 전처리
	차원 축소	PCA, t-SNE	계산량과 모델 성능 향상, 의미 있는 시각화

# 머신러닝 모델 유형

---

- ▶ 선형계열 모델
  - ▶ 입력 특성(features)들에 대해 가중합(weighted sum) 연산을 수행하여 점수를 구하고 이 점수를 사용하여 회귀 및 분류를 수행한다
- ▶ 트리계열 모델
  - ▶ 연산을 수행하지 않고, 특성별로 조건식을 적용하여 이진 분류를 순차적으로 수행하여 회귀 및 분류를 수행한다
- ▶ 신경망 모델
  - ▶ 선형계열 모델처럼 가중합 연산을 수행한 점수를 사용하되, 매우 크고 다양한 가중 매트릭스를 사용하여 성능을 개선한다

# 손실함수와 성능지표

## ▶ 대표적인 손실함수와 성능 평가 지표

	손실함수	성능평가지표
정의	손실함수를 줄이는 방향으로 모델이 학습을 함	성능을 높이는 것이 머신러닝을 사용하는 최종 목적임
회귀 모델의 대표적인 값	MSE (Mean Squared Error)	$R^2$
분류 모델의 대표적인 값	크로스 엔트로피	정확도, 정밀도, 재현률, F1점수, ROC-AUC

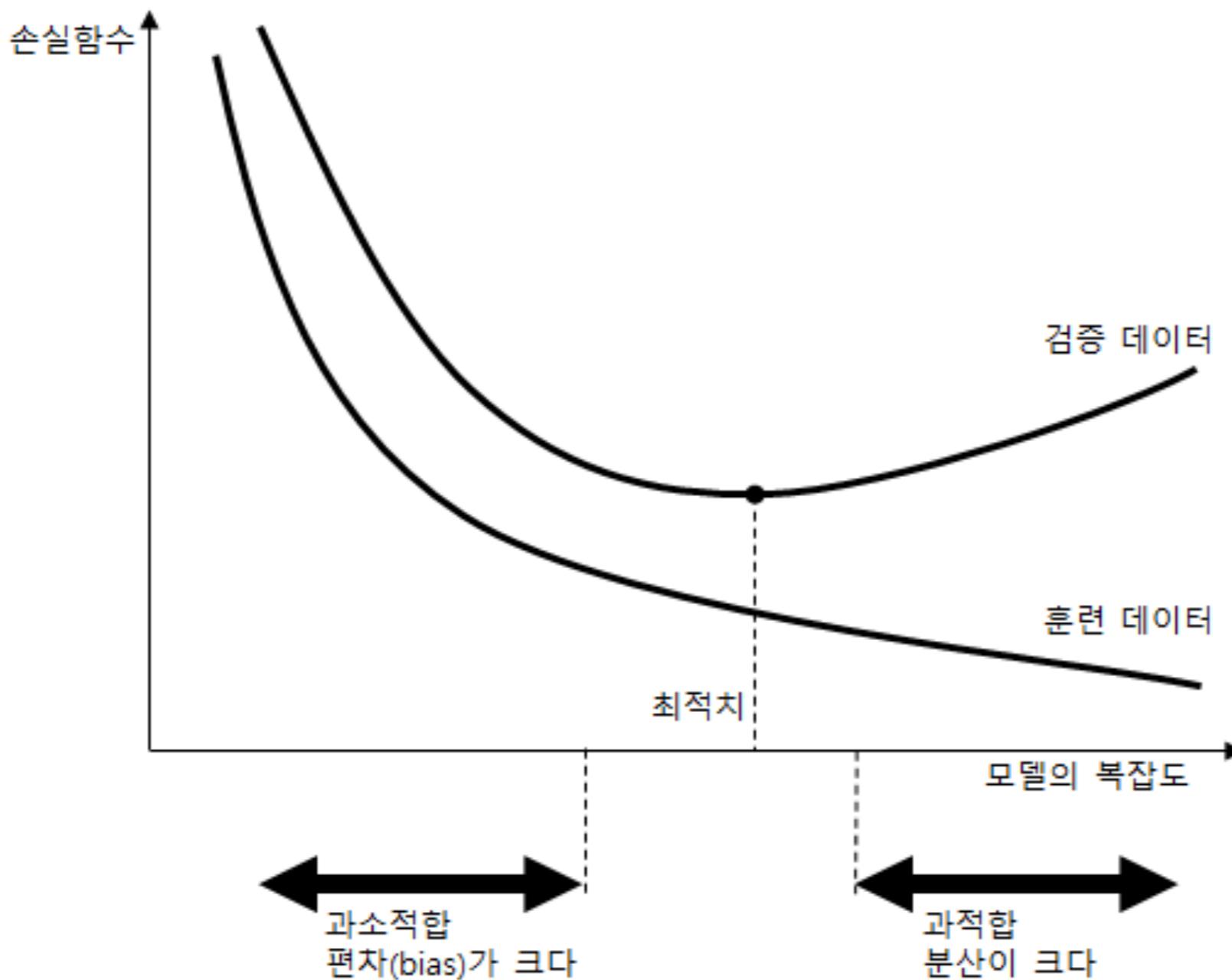
# 과대적합

---

- ▶ 모델이 훈련 데이터에 대해서는 잘 동작하지만 새로운 데이터에 대해서는 오히려 잘 동작하지 못하는 경우를 과대적합(*over fitting*)되었다고 한다.
- ▶ 과대적합은 주어진 훈련 데이터를 너무 세밀하게 학습에 반영하여 발생하는 현상이다.
- ▶ 과대적합을 줄이려면 더 많고 다양한 학습 데이터를 사용하거나 모델에 제한을 두어 일반화하는 것이 필요하다

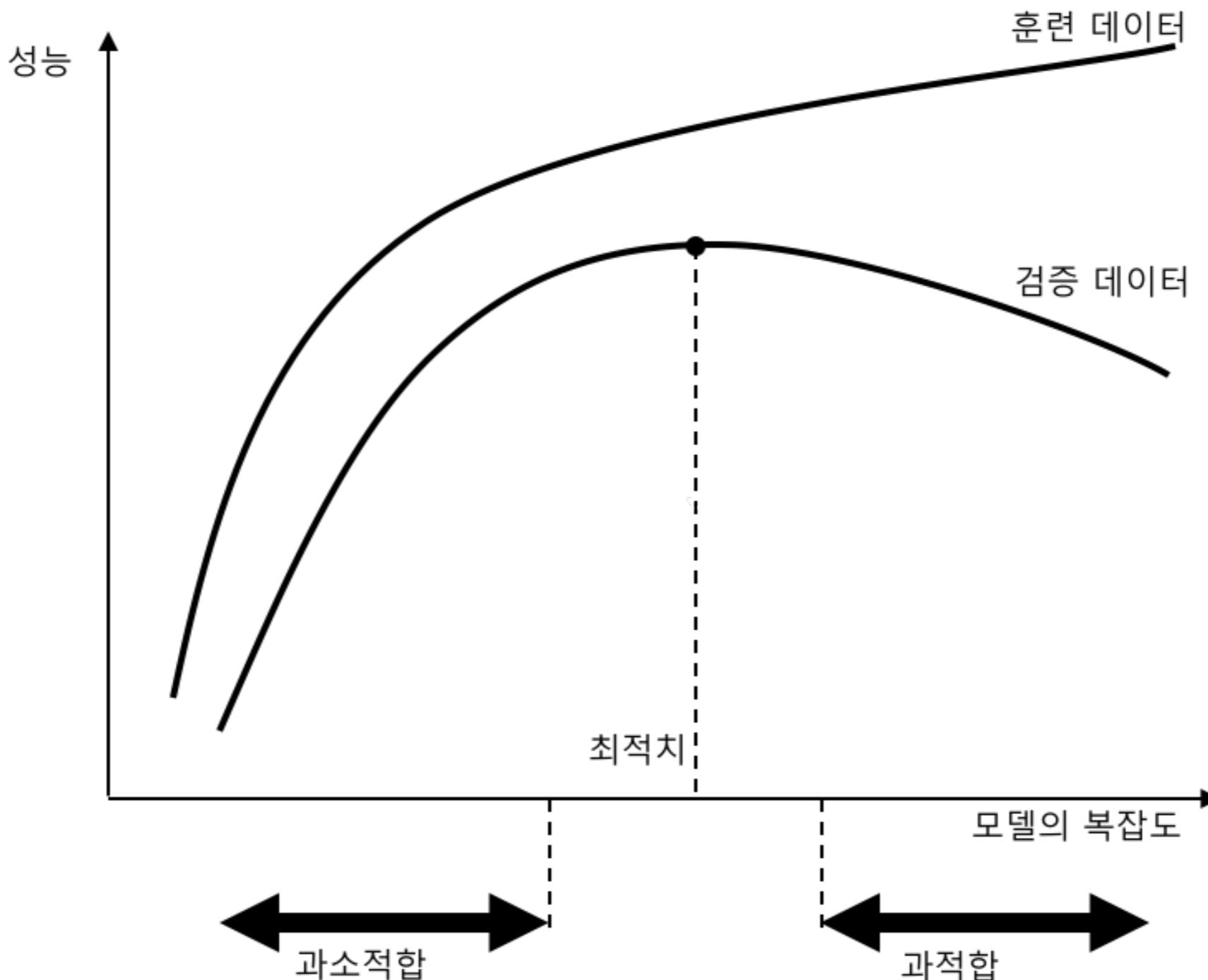
# 과대적합 검증

- ▶ 훈련 데이터와 검증 데이터에 대한 손실함수를 비교한다

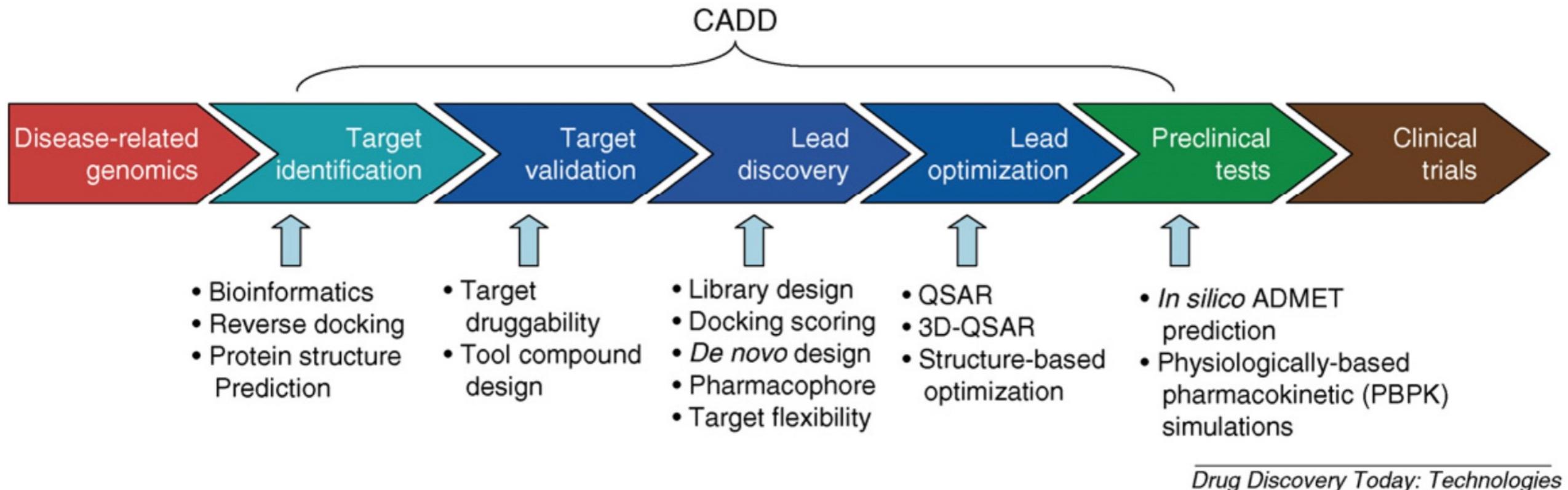


# 과대적합 검증

- ▶ 훈련 데이터와 검증 데이터에 대한 성능을 비교한다



# CADD with ML



*Drug Discovery Today: Technologies*

## ▶ Machine Learning Models

### ▶ Traditional models (1990~)

- ▶ Linear, logistic regression, support vector machines (SVM)
- ▶ Decision Tree, Random Forest, Boosting

### ▶ Deep Neural Networks (2012~)

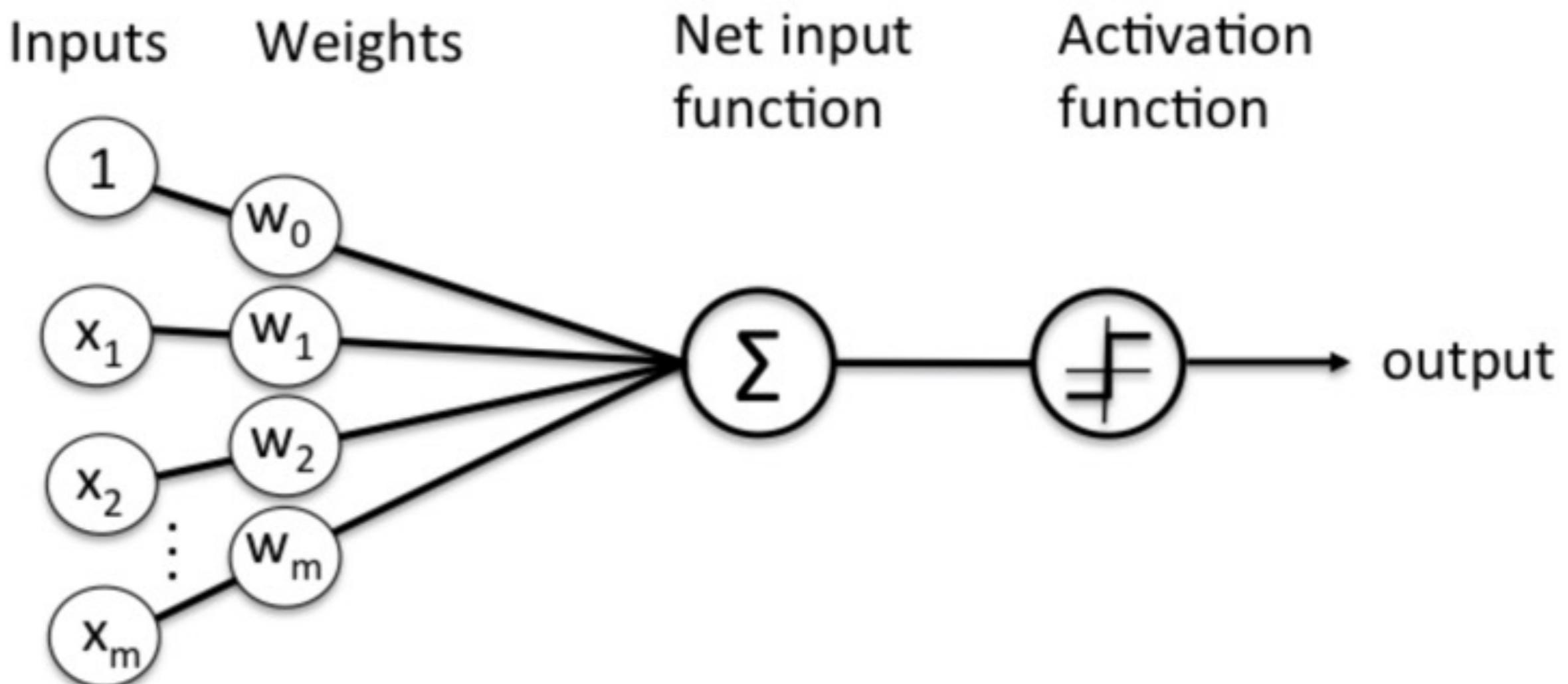
- ▶ MLP, CNN, RNN, Graph NN, Transformer

# 딥러닝 이해

# **MLP**

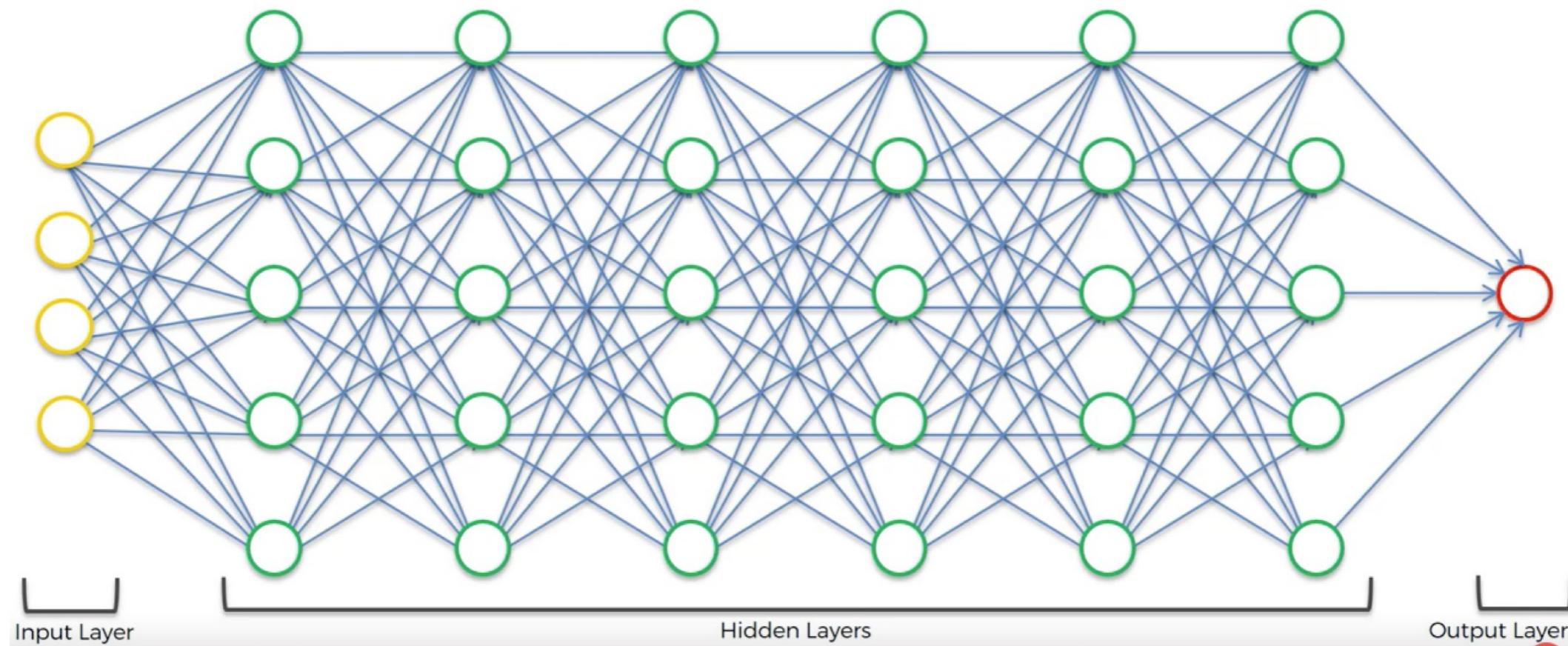
# 다층 퍼셉트론

- ▶ 신경망은 선형회귀 모델에서 발전하였고 비선형 기능이 추가되었다.
- ▶ 신경망의 최초 모델은 퍼셉트론(Perceptron)



# 다층 퍼셉트론 (MLP)

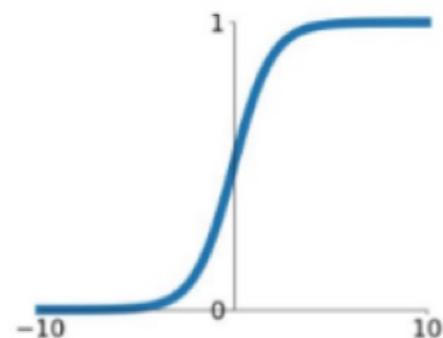
- ▶ 퍼셉트론을 여러 계층 쌓은 구조를 다층퍼셉트론(Multi layer Perceptron, MLP)이라고 한다.



# 활성화 함수 종류

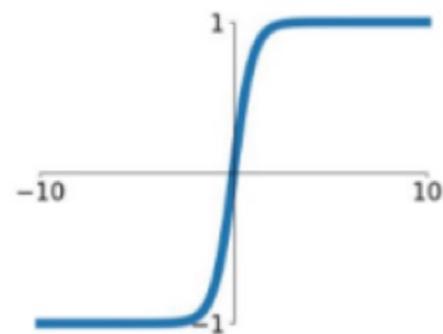
**Sigmoid**

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



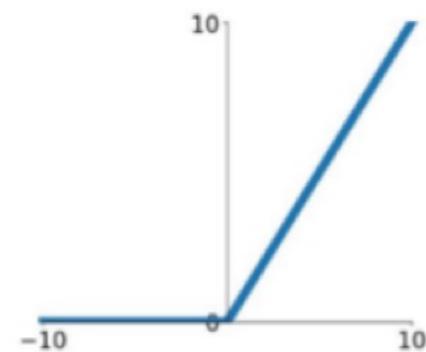
**tanh**

$$\tanh(x)$$



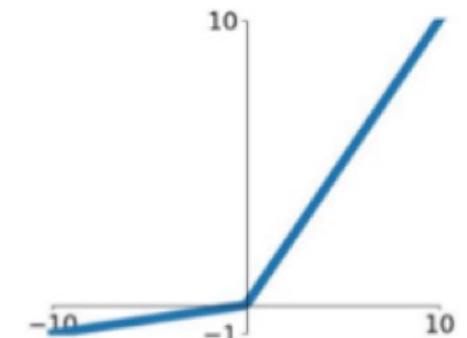
**ReLU**

$$\max(0, x)$$



**Leaky ReLU**

$$\max(0.1x, x)$$

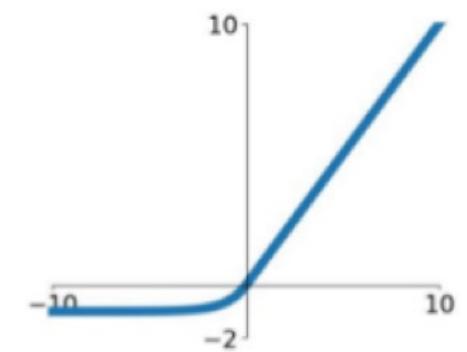


**Maxout**

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

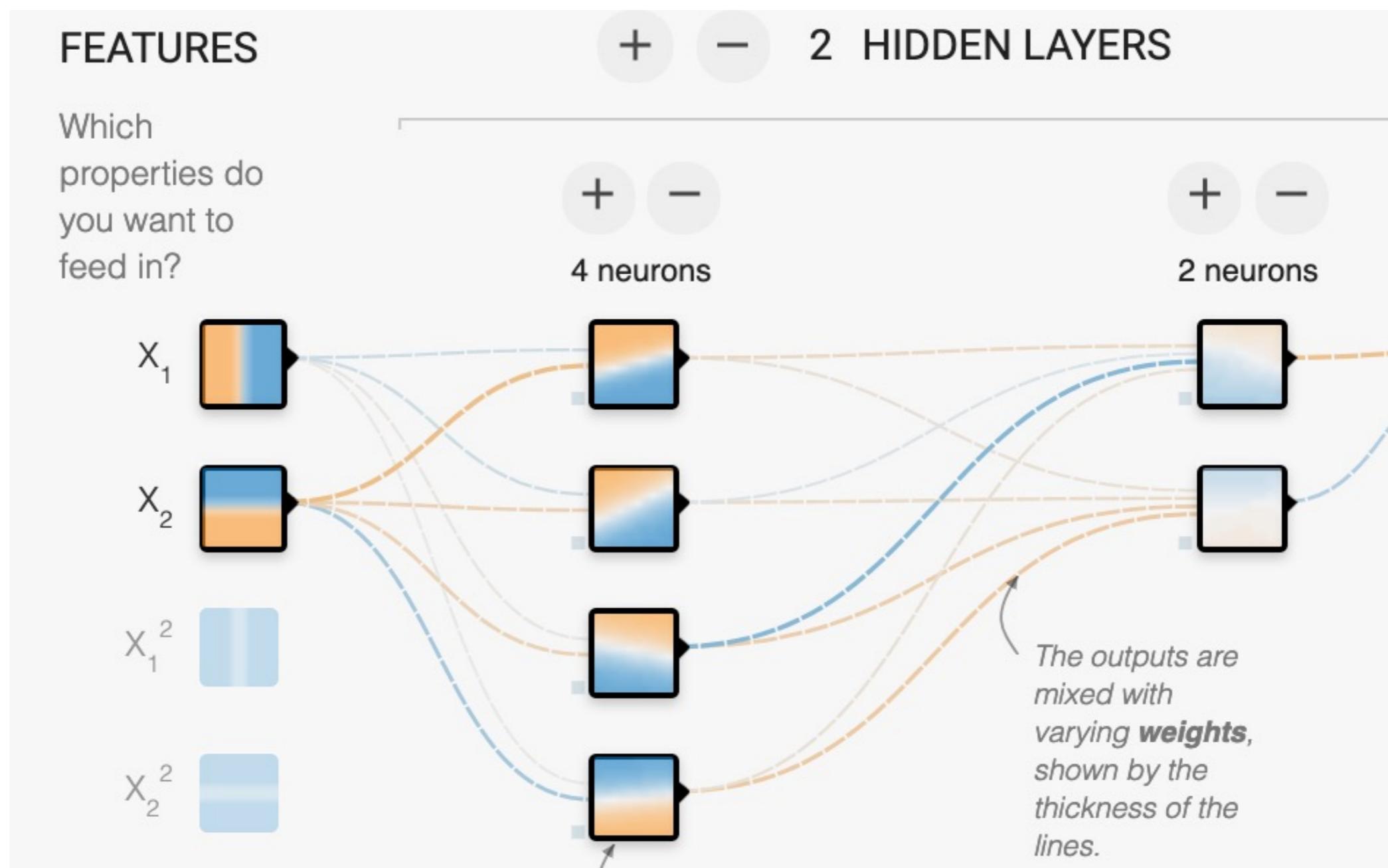
**ELU**

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



# 플레이그라운드

- ▶ MLP의 동작 시뮬레이션 도구
  - ▶ [playground.tensorflow.org](https://playground.tensorflow.org)



# CNN

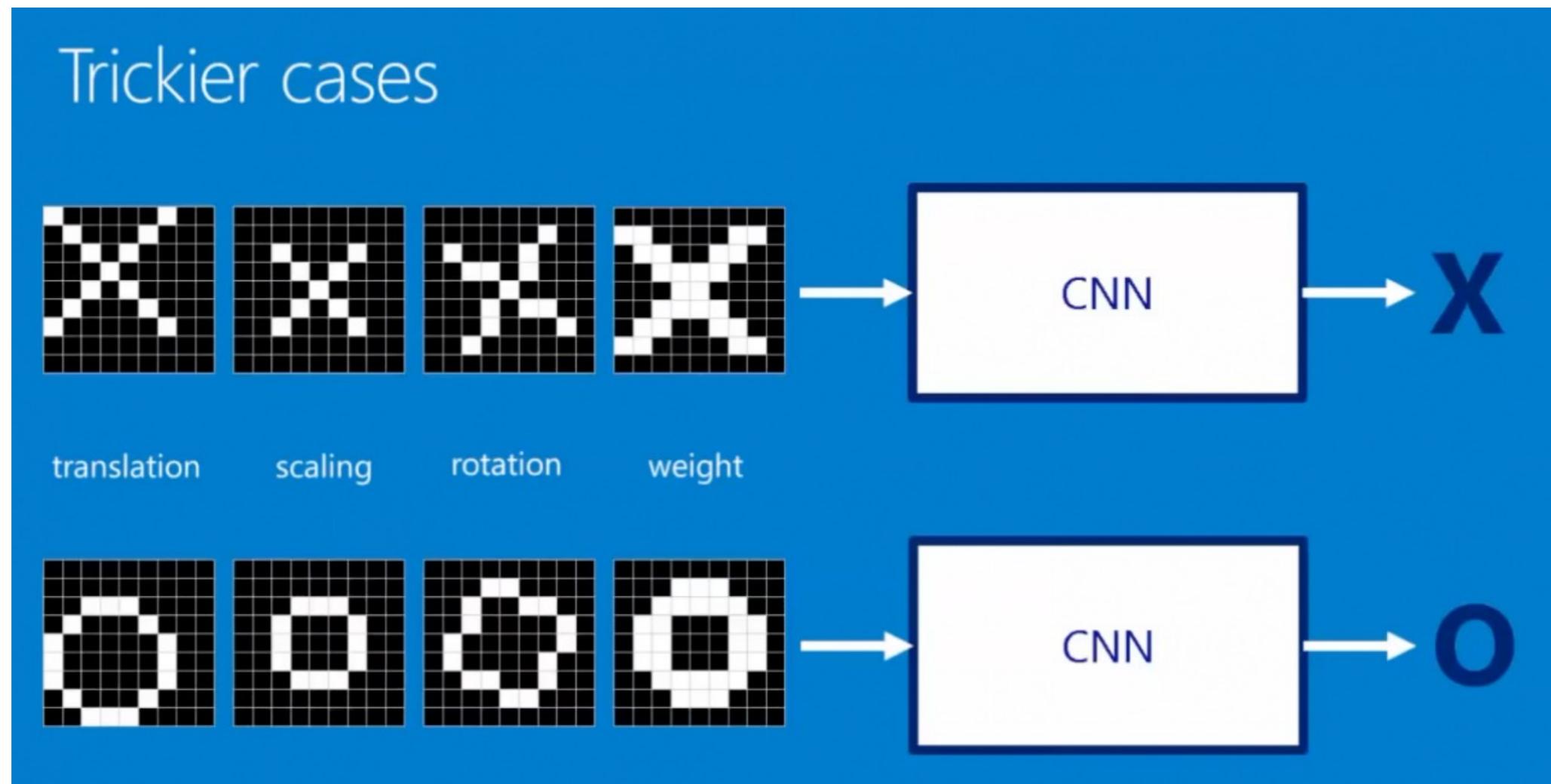
# 합성곱신경망 (CNN)

- ▶ 아래와 같이 임의의 크기와 모양을 가진 숫자 등 다양한 이미지 분석에서는 MLP가 잘 동작하지 않는다.
- ▶ 합성곱 신경망 (Convolution Neural Network, CNN)은 이를 개선하였다.

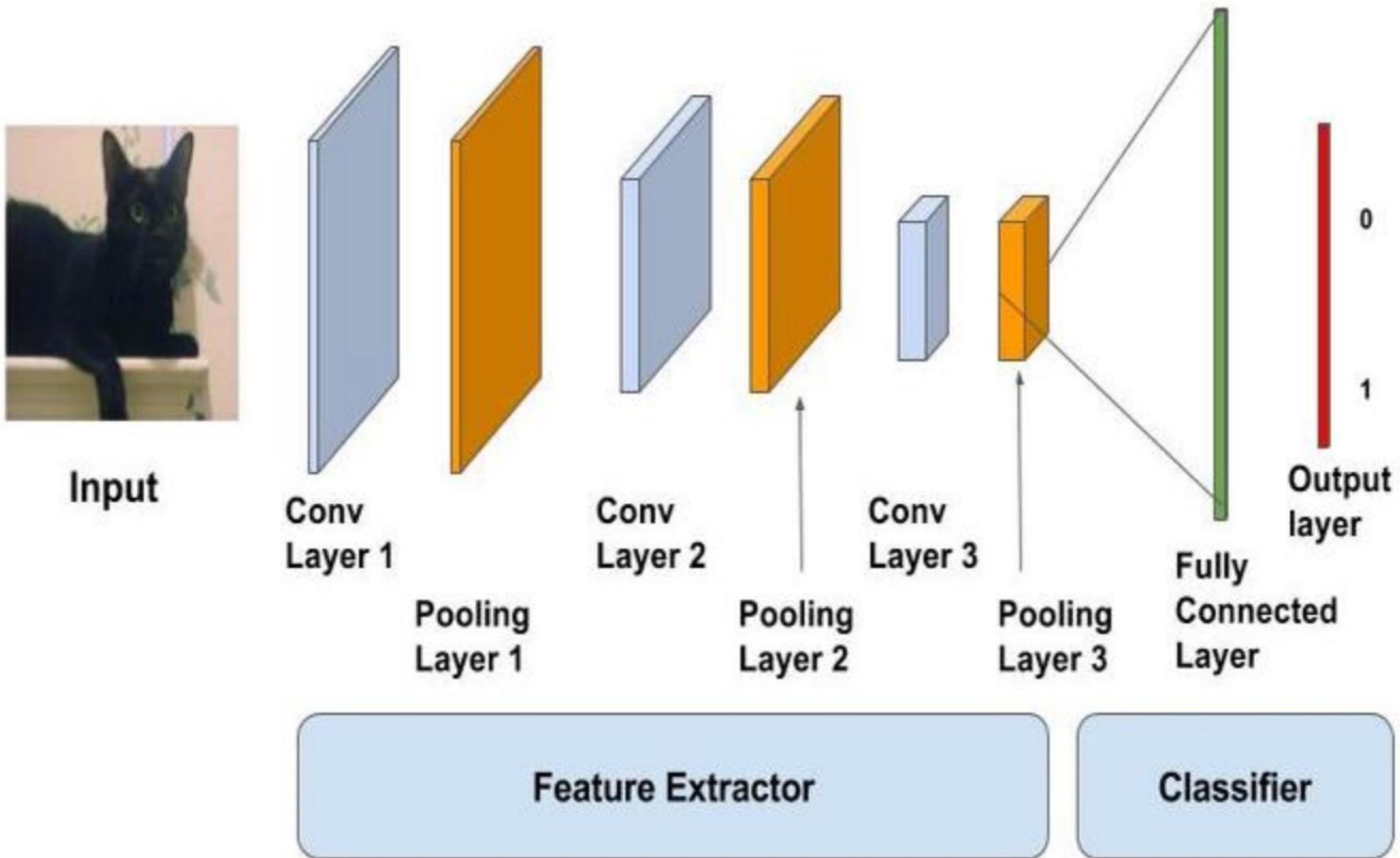


# CNN 개념

## ▶ Convolution Neural Network

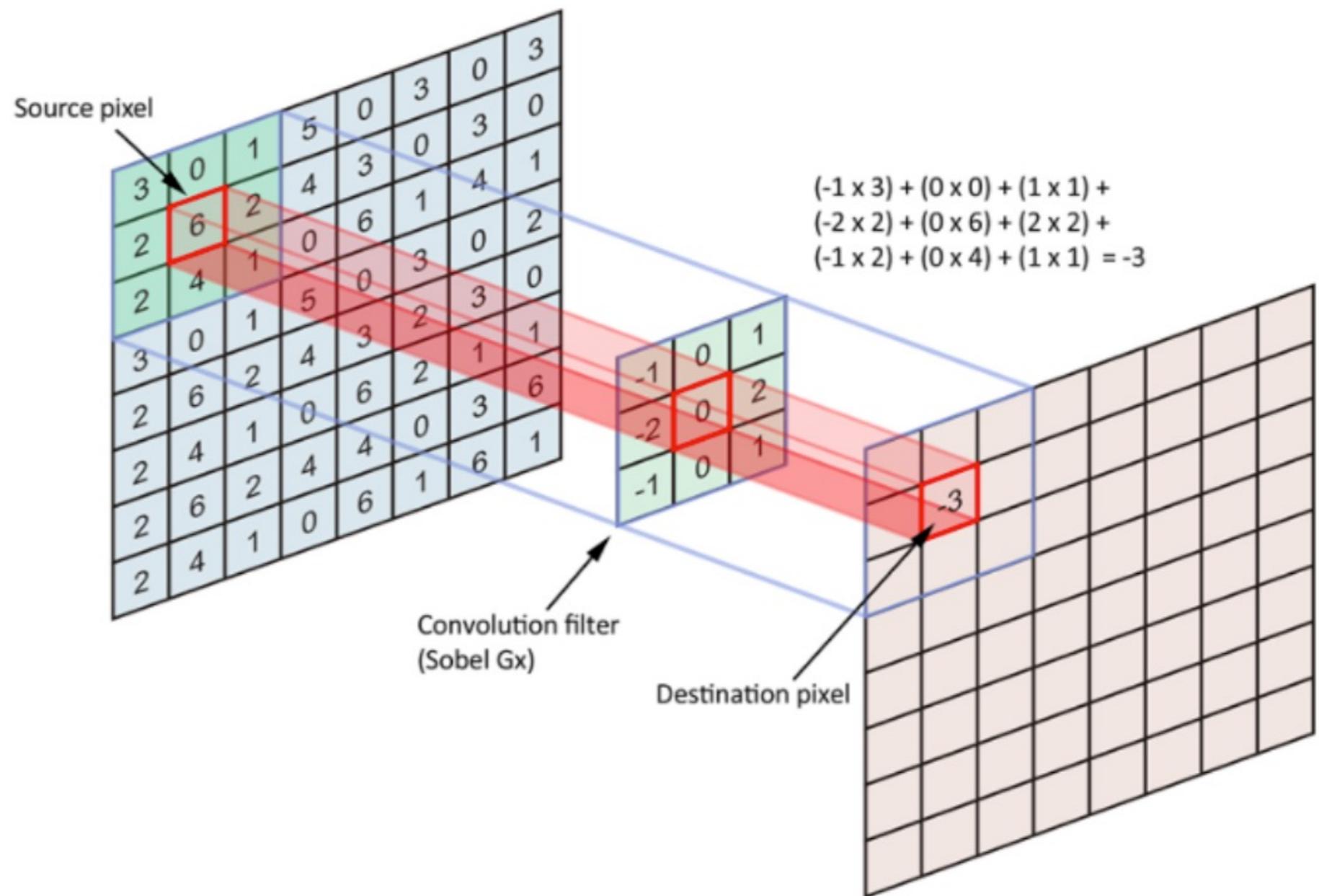


# CNN 개념



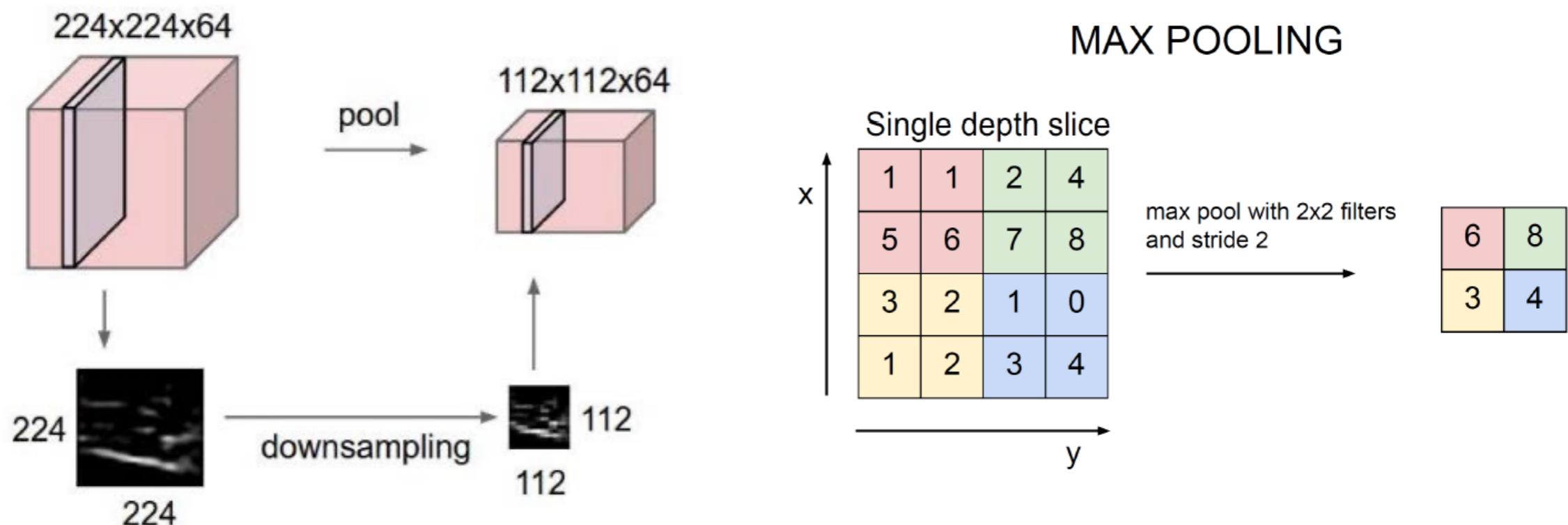
# Convolution 동작

- ▶ 합성곱동작
- ▶ 다층 필터링

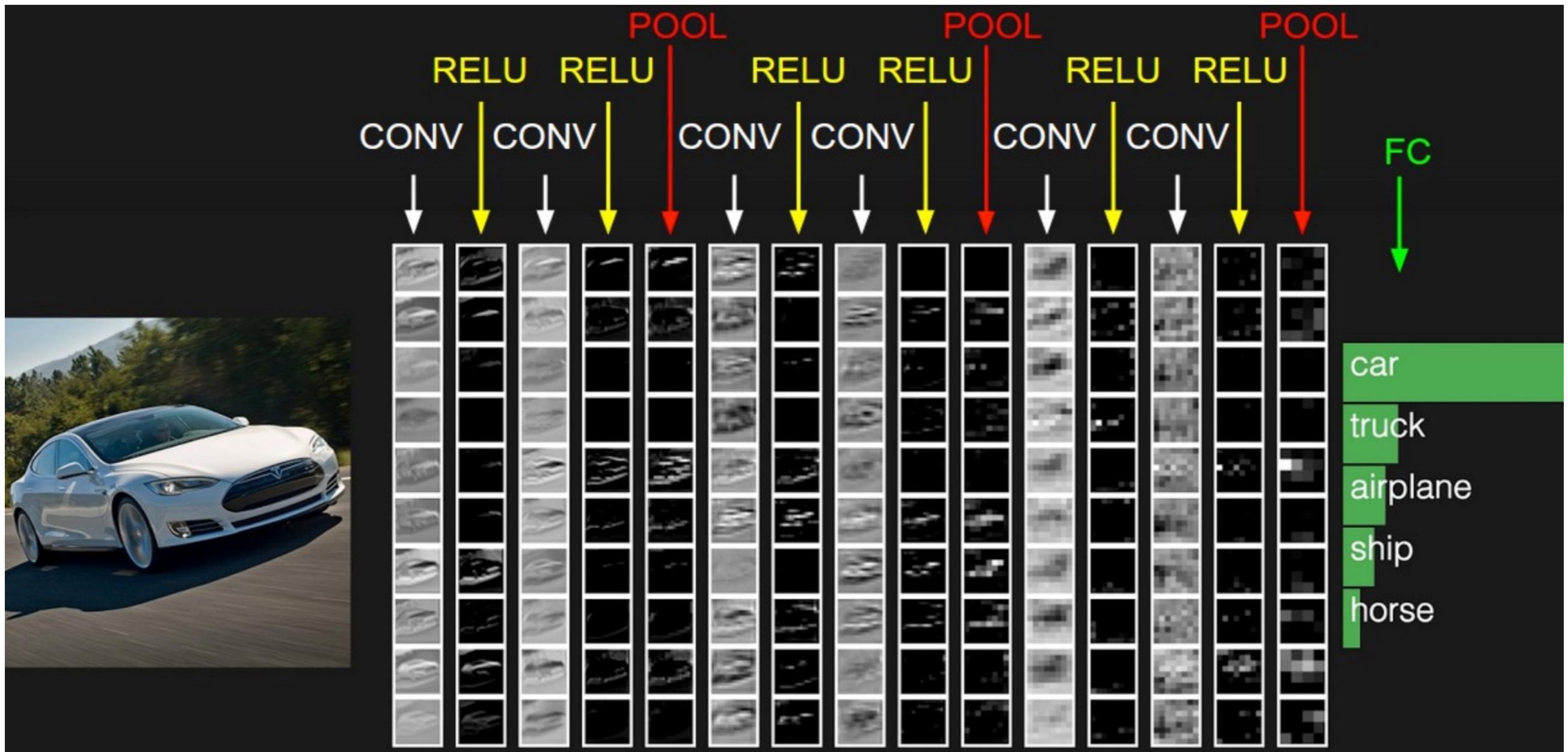


# Pooling

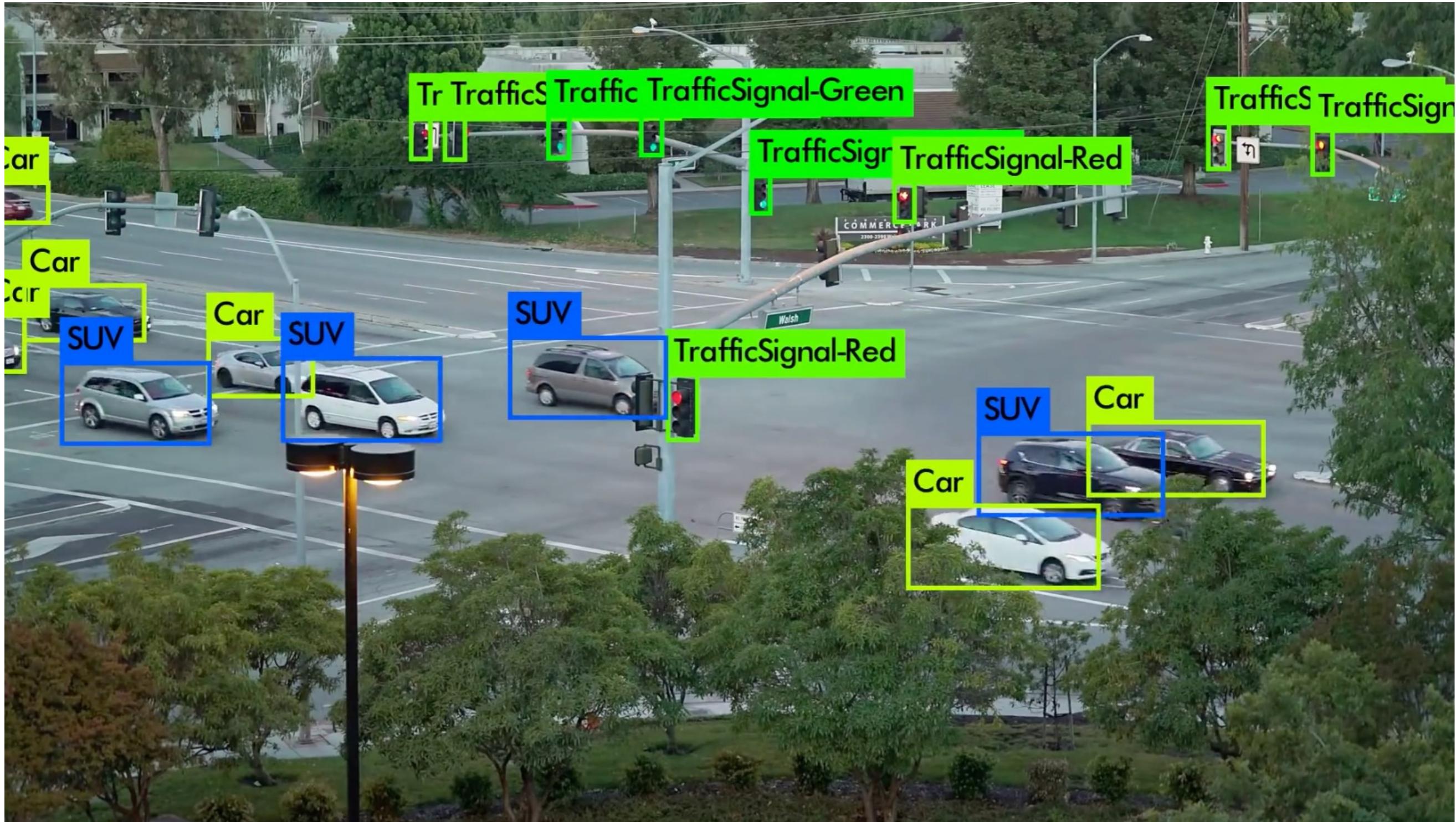
- ▶ 특정한 패턴이 공간상의 어느 위치에 있든 이 활성값이 다음 단계로 넘어가면서 좌우로 조금씩 움직일 수 있다
- ▶ 풀링은 과대적합을 해소하는 데에도 기여한다.



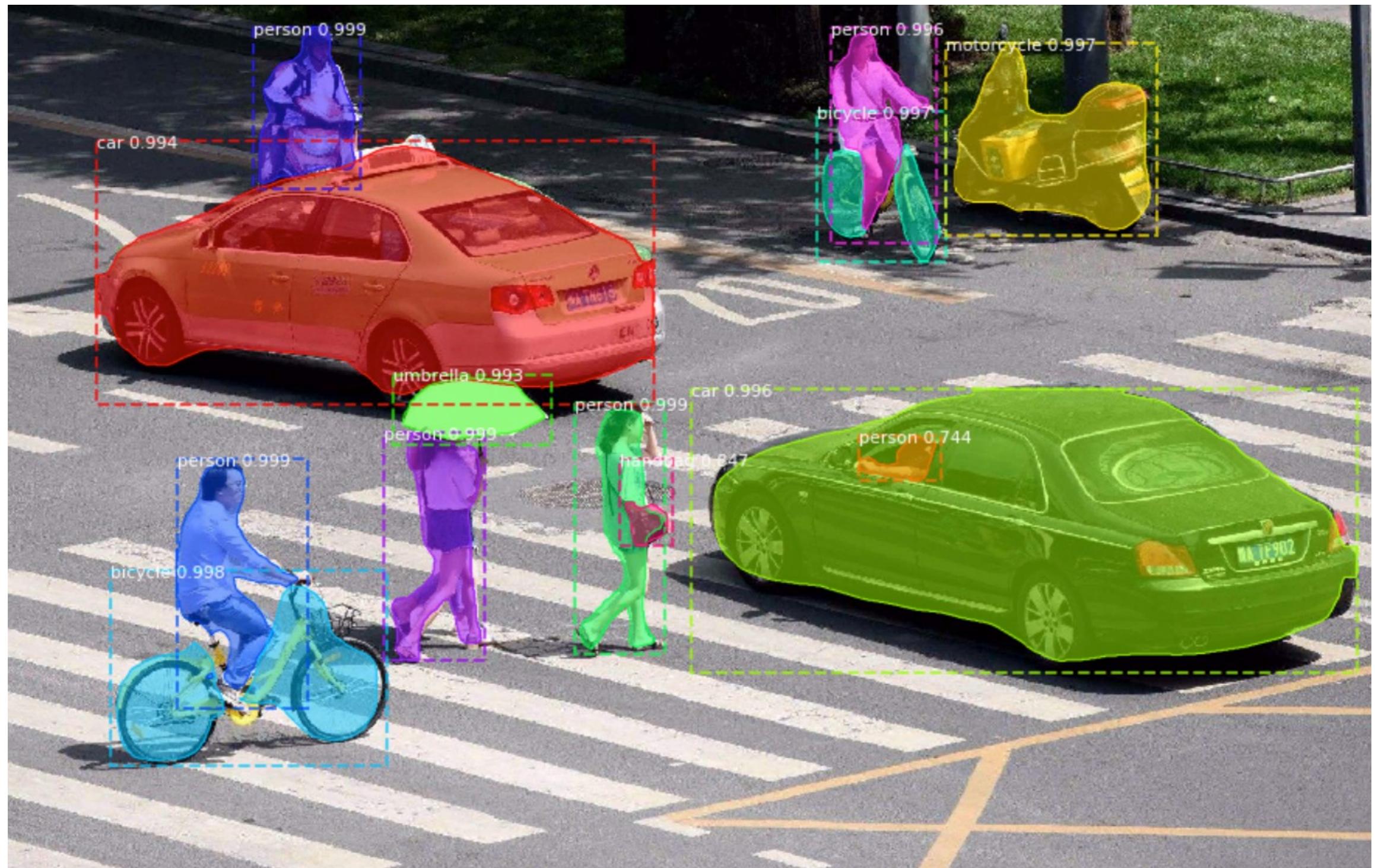
# 이미지 인식



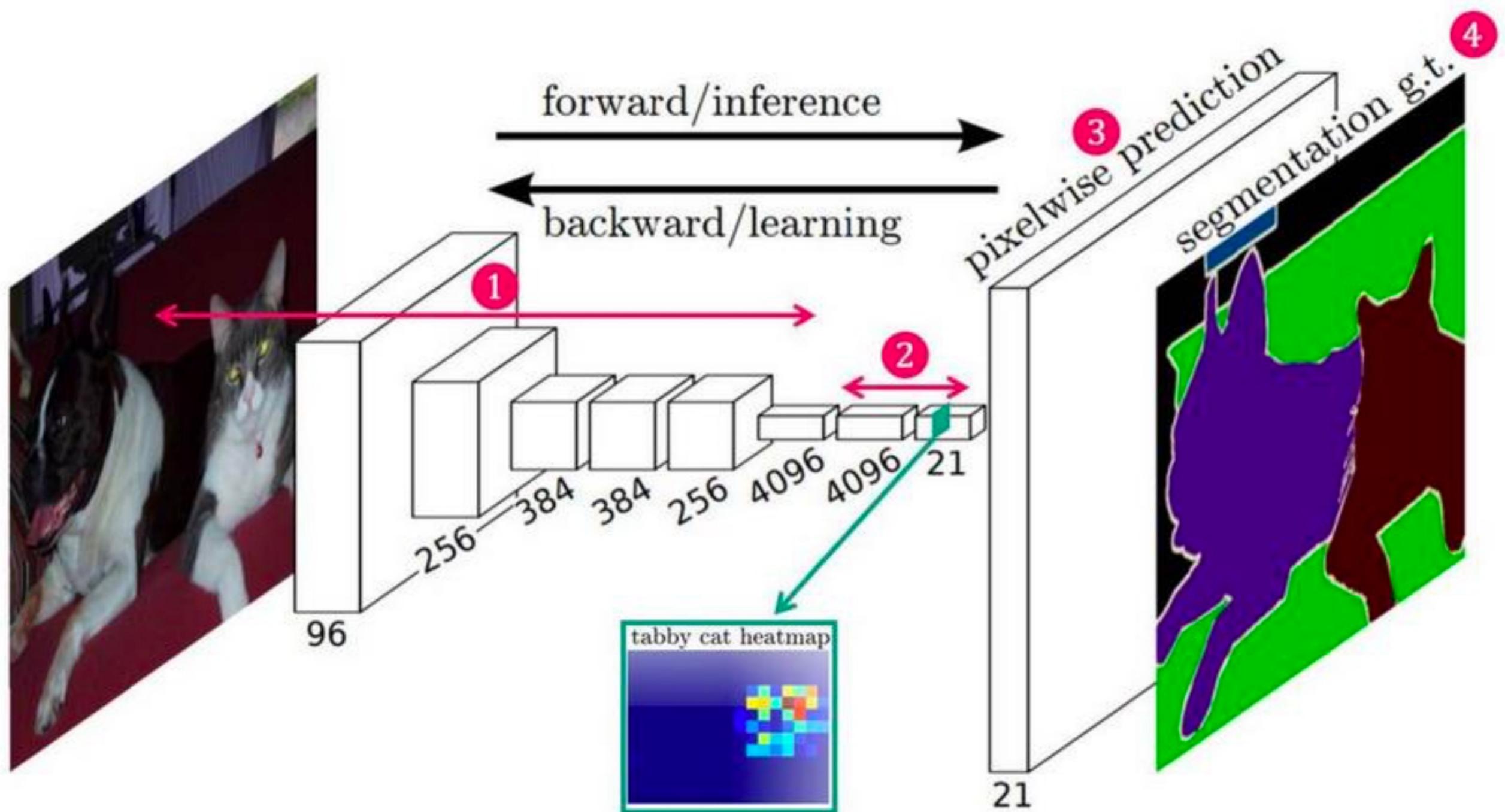
# Object Detection



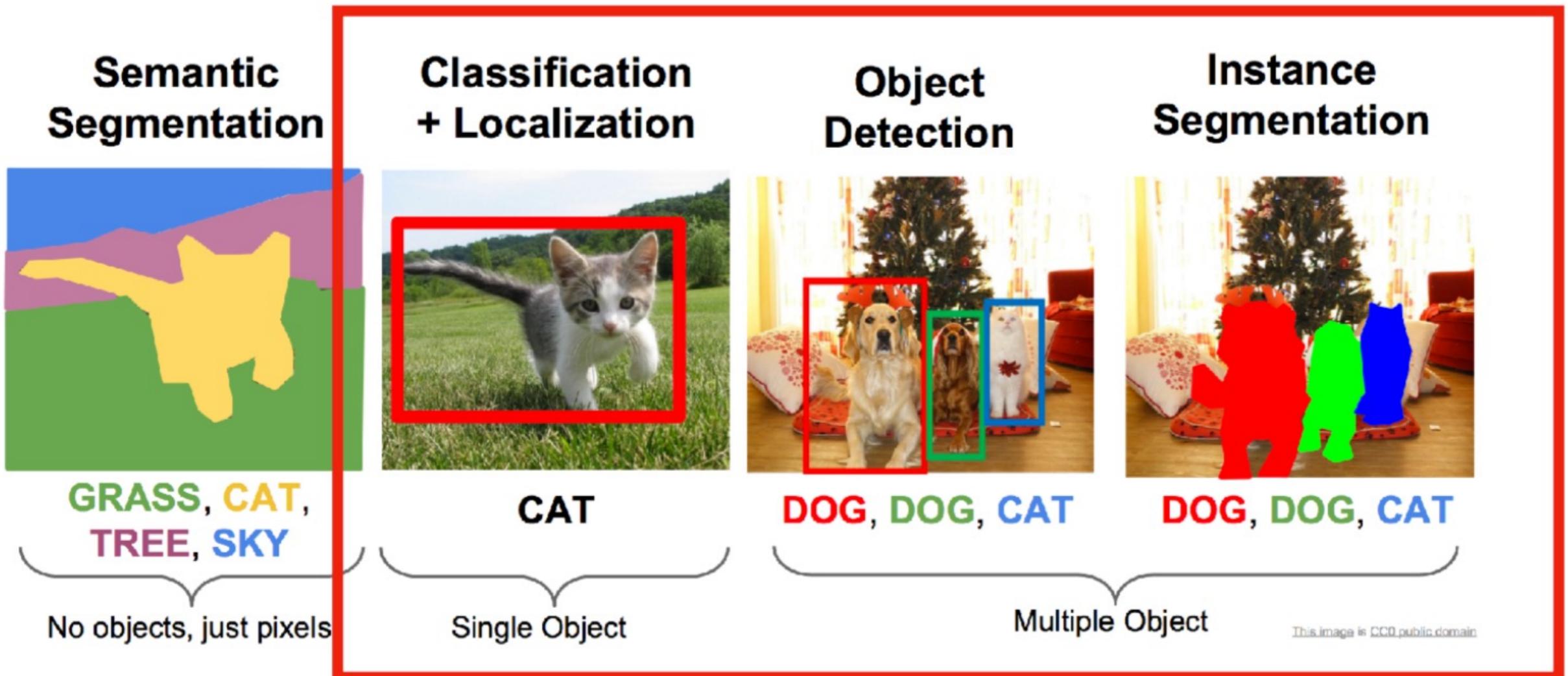
# Object Segmentation



# Object Segmentation



# Image Analysis



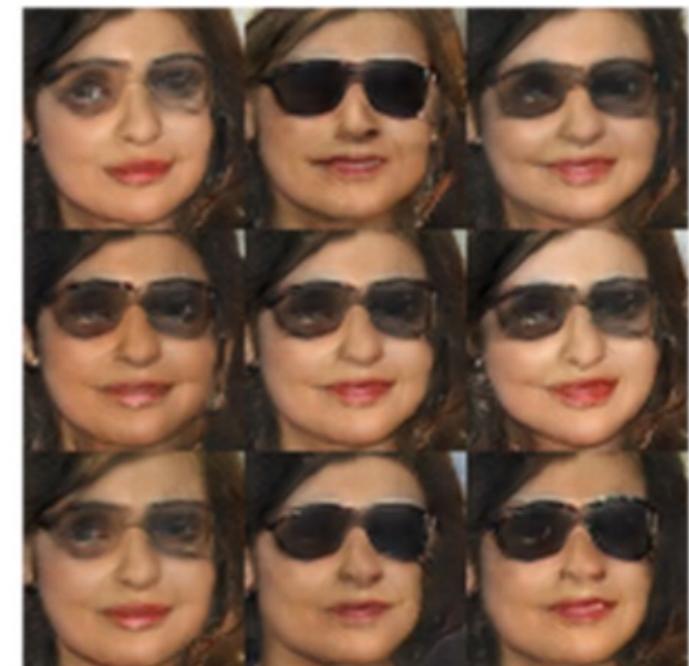
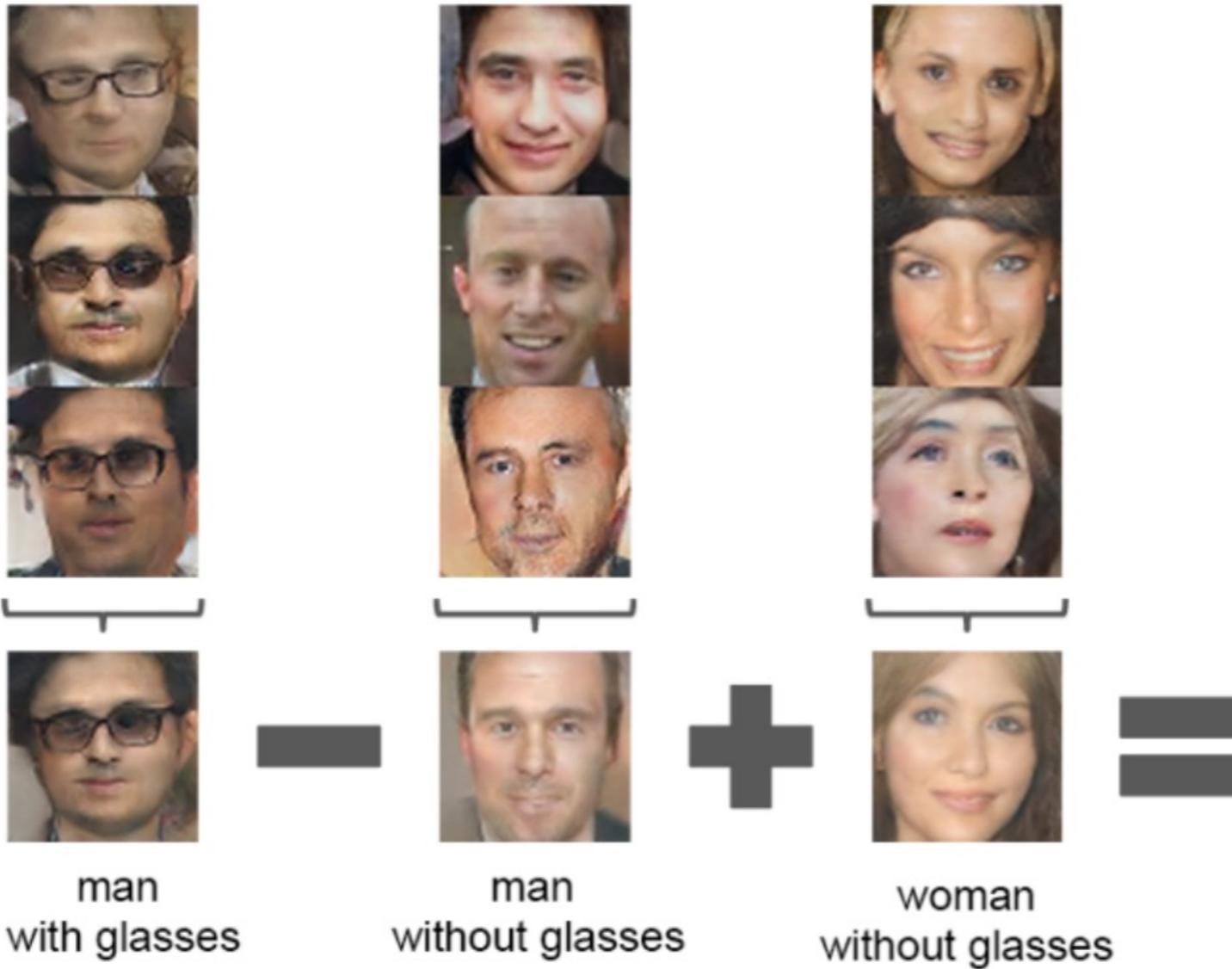
# Image Enhancement

---

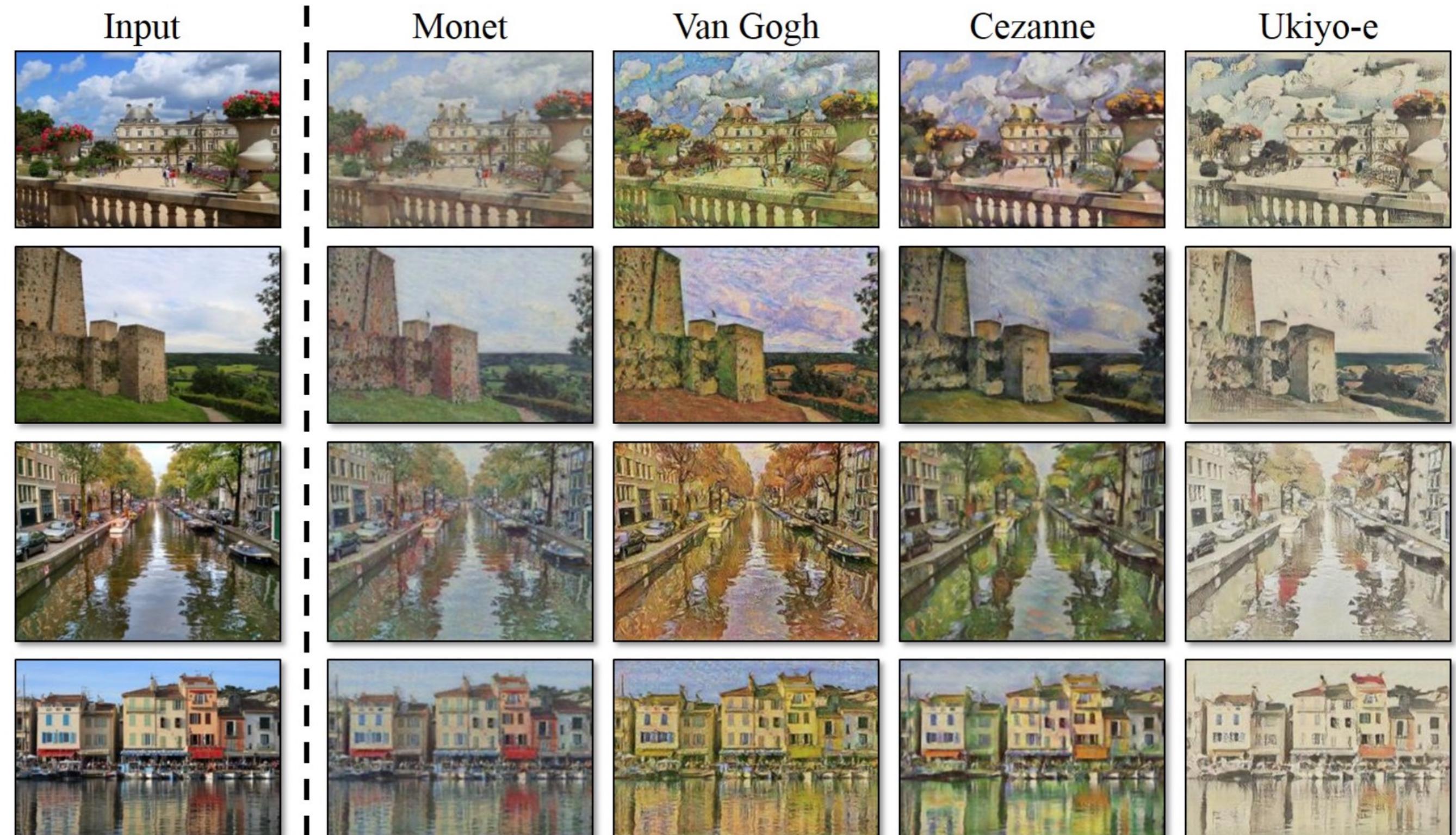


# 딥러닝 응용

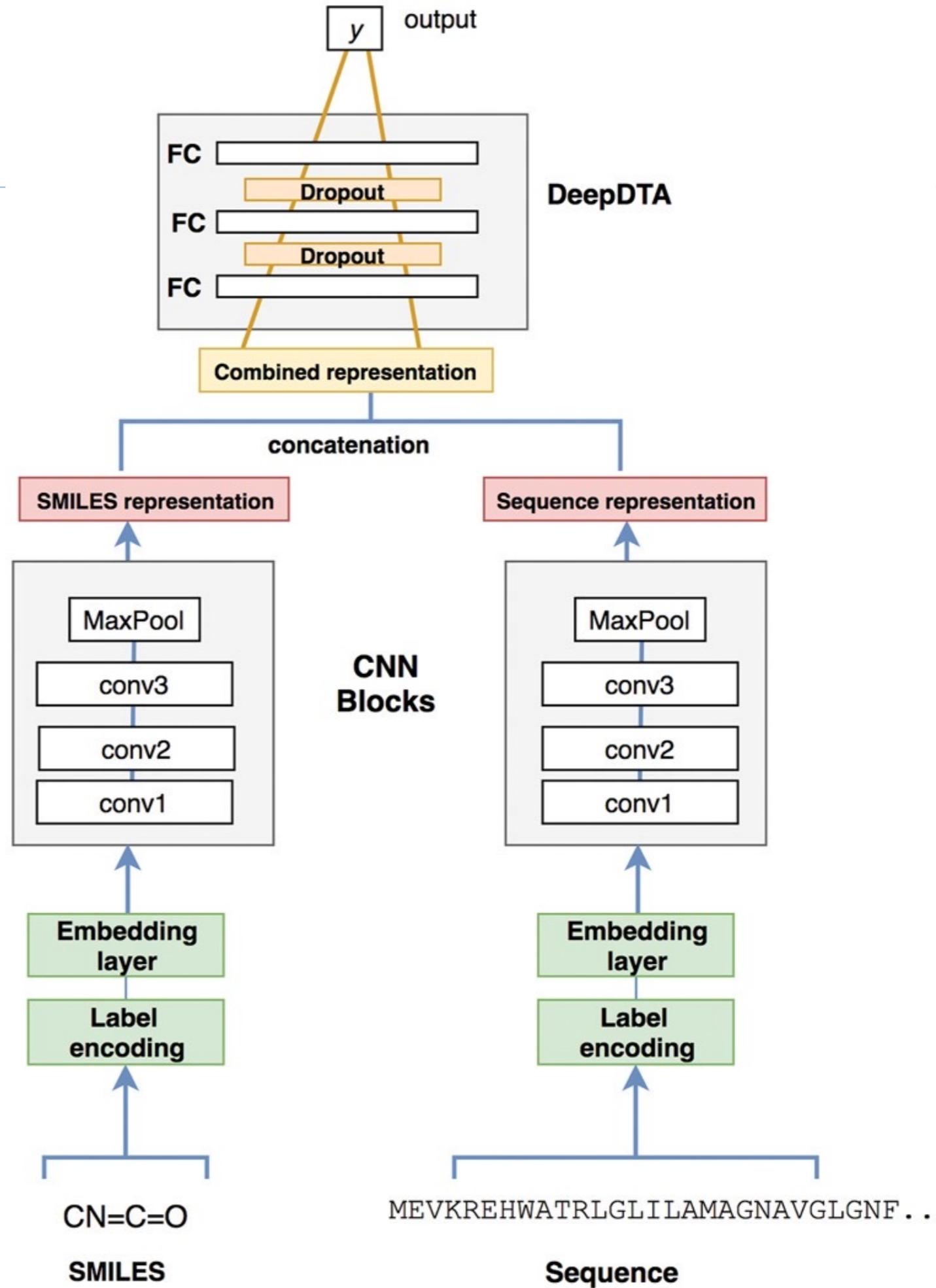
- ▶ GAN
  - ▶ [bit.ly/InterpCeleb](http://bit.ly/InterpCeleb)
- ▶ 잠재공간 (latent space) 연산



# Style Transfer (cycle GAN)



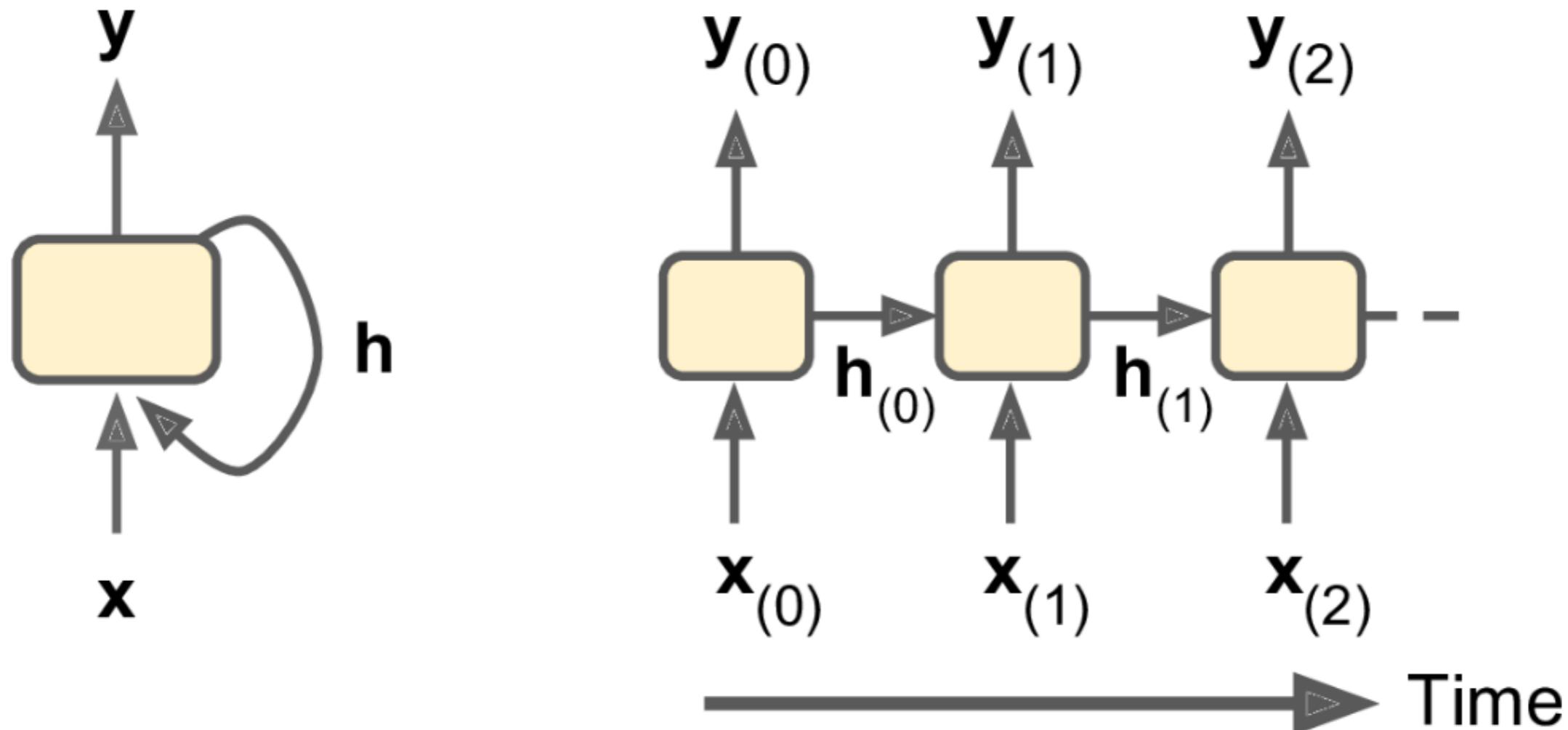
# DeepDTA



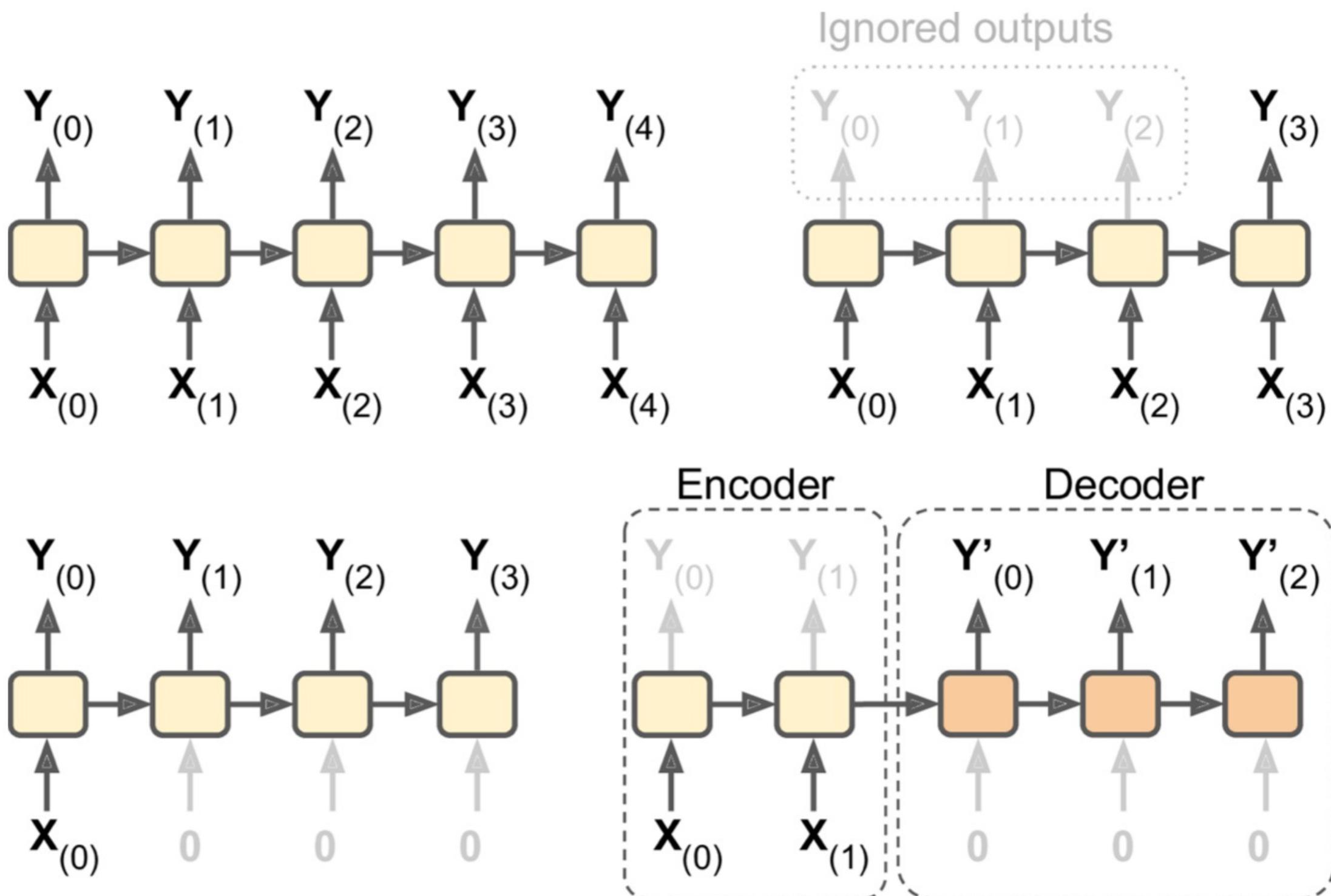
# RNN

# Recursive Neural Network

- ▶ 과거의 입력에 대한 상태 정보를 순환적으로 재사용



# RNN 이용 유형



# 이미지 캡션



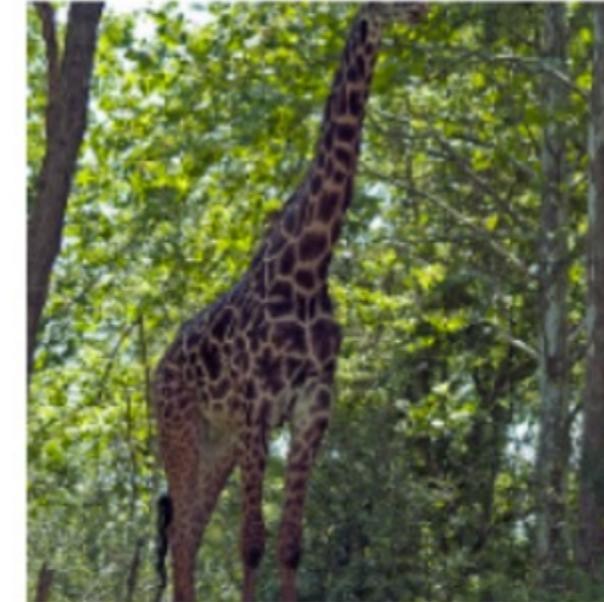
A dog is standing on a hardwood floor.



A stop sign is on a road with a mountain in the background.



A group of people sitting on a boat in the water.

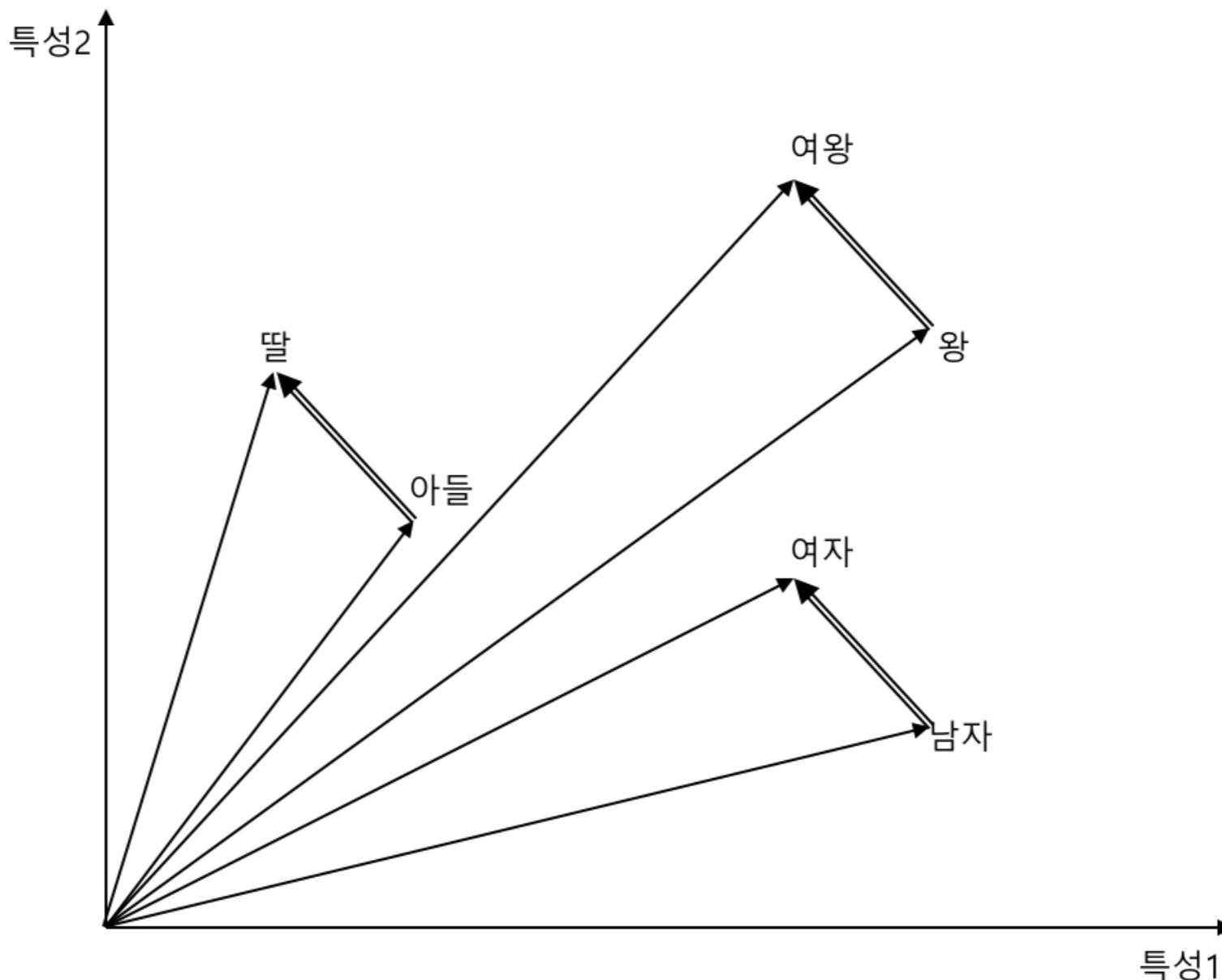


A giraffe standing in a forest with trees in the background.

# **NLP (자연어 처리)**

# 단어 임베딩 (word vector)

- ▶ 예를 들어 왕:여왕 = 아들:?에서 딸을 추정한다



# Transformer

---

- ▶ NLP에서 RNN 기반의 Seq2seq 모델을 개선
  - ▶ 문장의 길이가 길어지면 멀리 떨어진 단어에 대한 상호 정보가 줄어들어 제대로된 예측이 불가능해짐
  - ▶ 순차적으로 연산을 하면 연산의 병렬화가 불가능해 연산 속도가 저하됨
- ▶ 트랜스포머
  - ▶ 입력 토큰의 셀프 어텐션을 사용한 모델

# 셀프 어텐션

---

- ▶ 셀프 어텐션
  - ▶ 시퀀스 요소들 가운데 태스크 수행에 중요한 요소에 집중하고 그렇지 않은 요소는 무시해 태스크 수행 성능을 올리는 개념
  - ▶ 기계 번역에서 처음 도입
- ▶ 다른 딥러닝 모델과 비교
  - ▶ CNN은 합성곱 필터 크기를 넘어서는 문맥은 읽어내기 어렵다는 단점이 있다
  - ▶ RNN은 시퀀스 길이가 길어질수록 정보 압축에 문제가 발생한다

# 셀프 어텐션

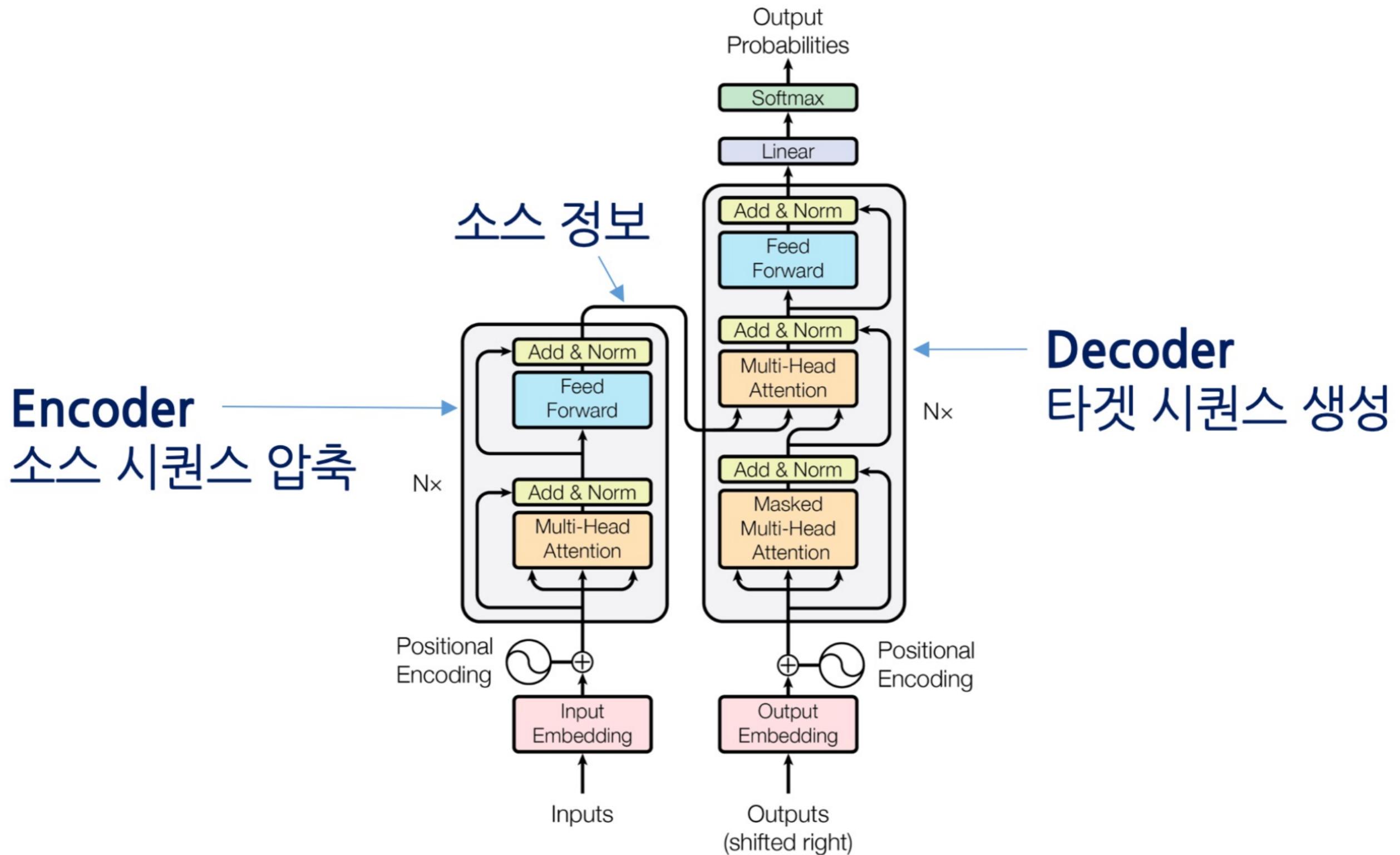
- ▶ (참고) 초기 어텐션: café를 디코딩할 때 주의해서 볼 내용은?
  - ▶ RNN 구조에서 동작 seq2seq



- ▶ 셀프 어텐션: 입력 자신 전체에 대해 수행하는 어텐션
  - ▶ CNN과 RNN의 단점을 개선



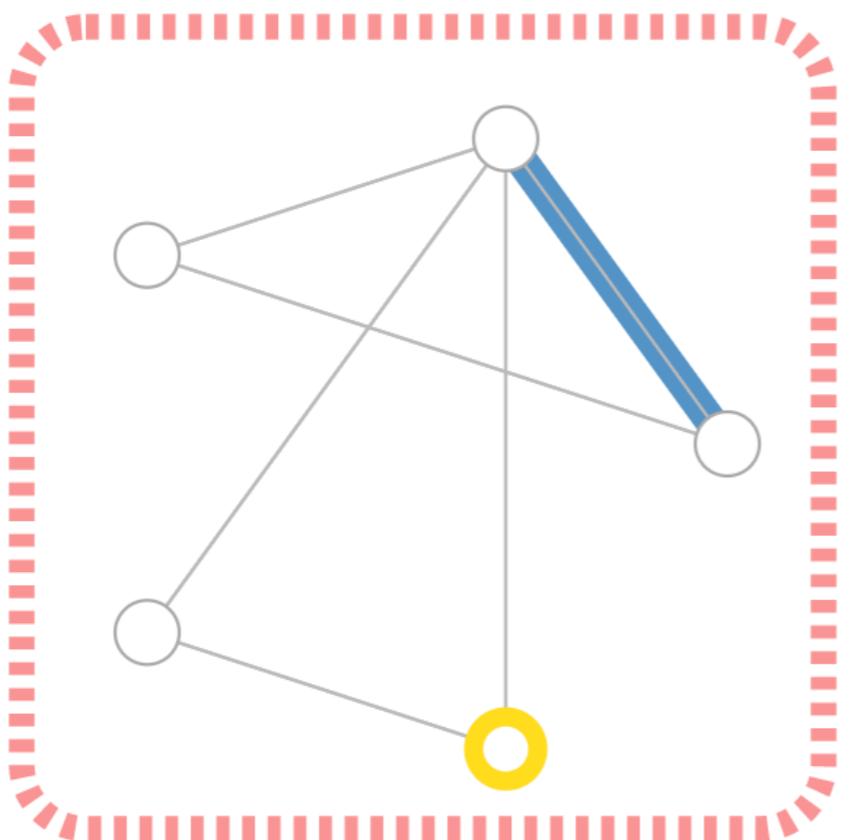
# 트랜스포머



# **Graph Neural Network**

# Graph Network

- ▶ Distill
- ▶ 그래프에 포함되는 정보



Vertex (or node) embedding



Edge (or link) attributes and embedding

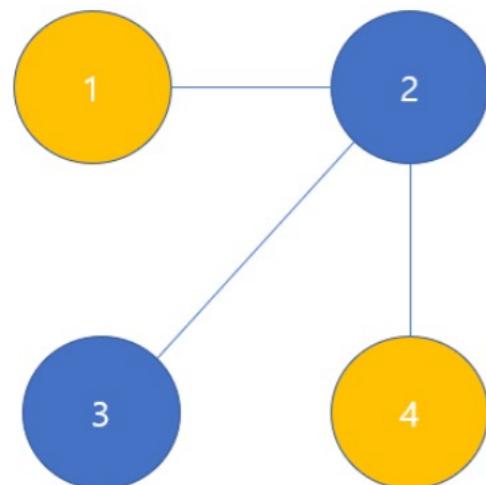


Global (or master node) embedding



# Adjacency/Feature Matrix

- ▶ **Adjacency matrix**
  - ▶ node 간의 관계(edge)를 표현하며 아래와 같이 방향성은 없는 경우 대칭적이다
- ▶ **Feature matrix**
  - ▶ node에 담긴 정보를 나타내며, 아래의 경우 특성이 3개라고 가정함 (색을 구분)



$$\text{Adjacency matrix } A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\text{Feature matrix } F = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

# 그래프 예측 태스크

---

- ▶ graph-level
  - ▶ 그래프 전체의 특성 예측 (예: 문자의 냄새, 링 포함, 수용체 결합도 예측 등)
  - ▶ 이미지 분류나 문서의 감성예측과 유사한 작업
- ▶ node-level
  - ▶ 노드의 identity나 역할을 예측 (예: 각 노드가 어느 특정 노드와 가까운지 분류 예측)
  - ▶ 이미지 세그멘테이션(각 픽셀의 역할 예측), 문장에서 POS 예측과 유사한 작업
- ▶ edge-level
  - ▶ 노드간의 관계를 예측 (예: 이미지 객체간의 관계 기술)

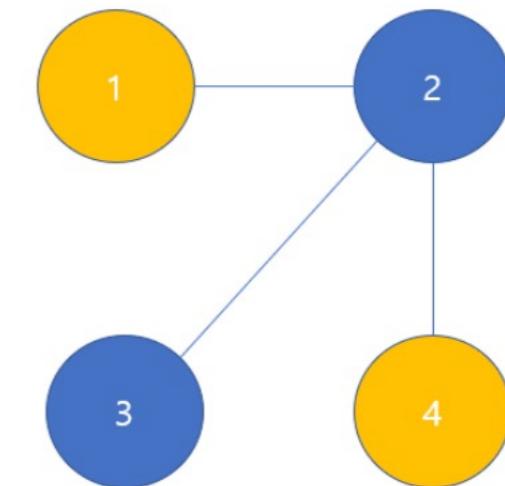
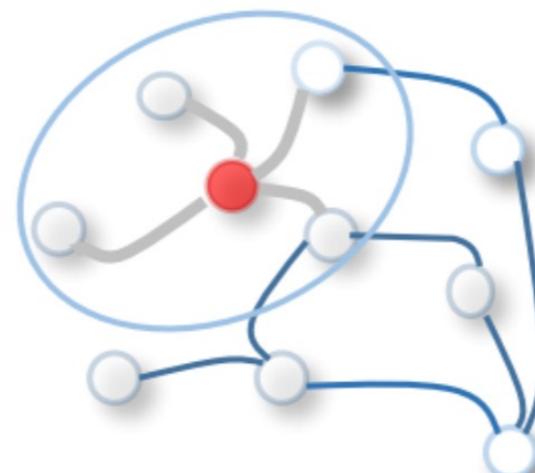
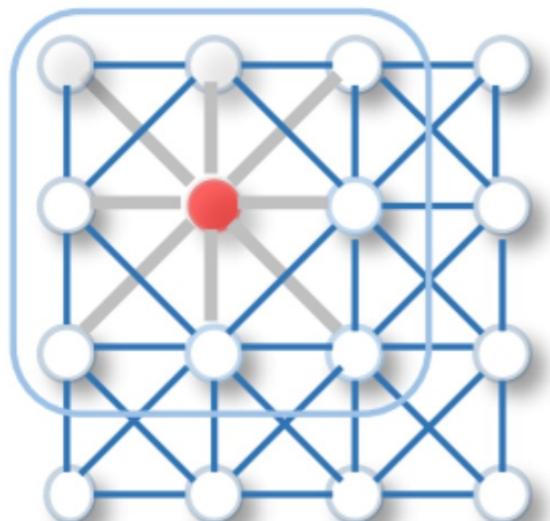
# 그래프 컨볼류션 모델(GCN)

---

- ▶ CNN에서 커널 계수를 학습으로 찾아내듯이 문자 구조를 기술하는 계수를 학습으로 찾는다
  - ▶ 문자 그래프의 노드와 엣지를 벡터로 변환한다
- ▶ 다양한 변형
  - ▶ 그래프 컨볼류션 (GraphConvModel),
  - ▶ 위브 모델 (Weave model)
  - ▶ 메시지 전달 신경망 (MPNNModel),
  - ▶ 딥 텐서 신경망 (DTNNModel)
- ▶ 단점
  - ▶ 문자 그래프만 사용하므로 문자 구조에 대한 정보가 사라진다
    - ▶ 거대 문자에는 잘 동작하지 않는다

# GCN

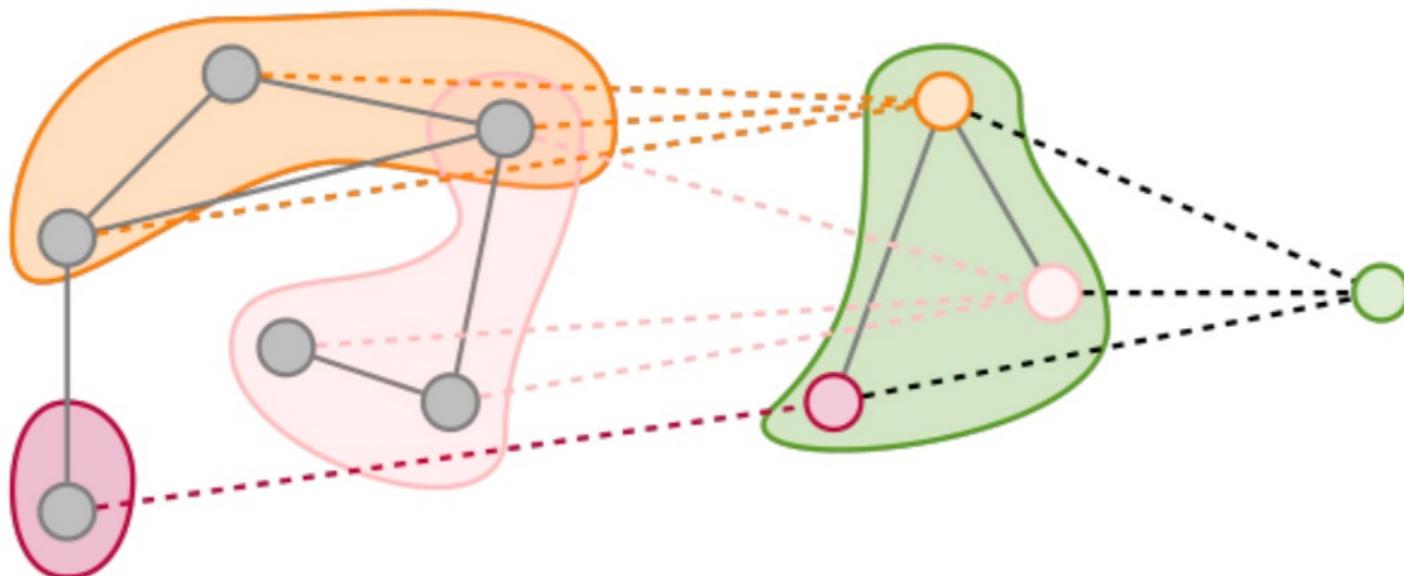
- ▶ edge로 연결된 node끼리의 정보 교환을 표현하는 방법이 필요
  - ▶ 컨볼류션을 사용하는 방법을 채택
  - ▶ 주변 node의 정보를 spatial하게 얻어 정보를 업데이트하는 방법
- ▶ 2D 콘볼류션과 그래프 콘볼류션의 차이



- ▶ (예) 2번 node를 업데이트할 때는 1,3,4 node를 사용하지만 다른 node는 2번 node만을 사용한다

# Graph pooling

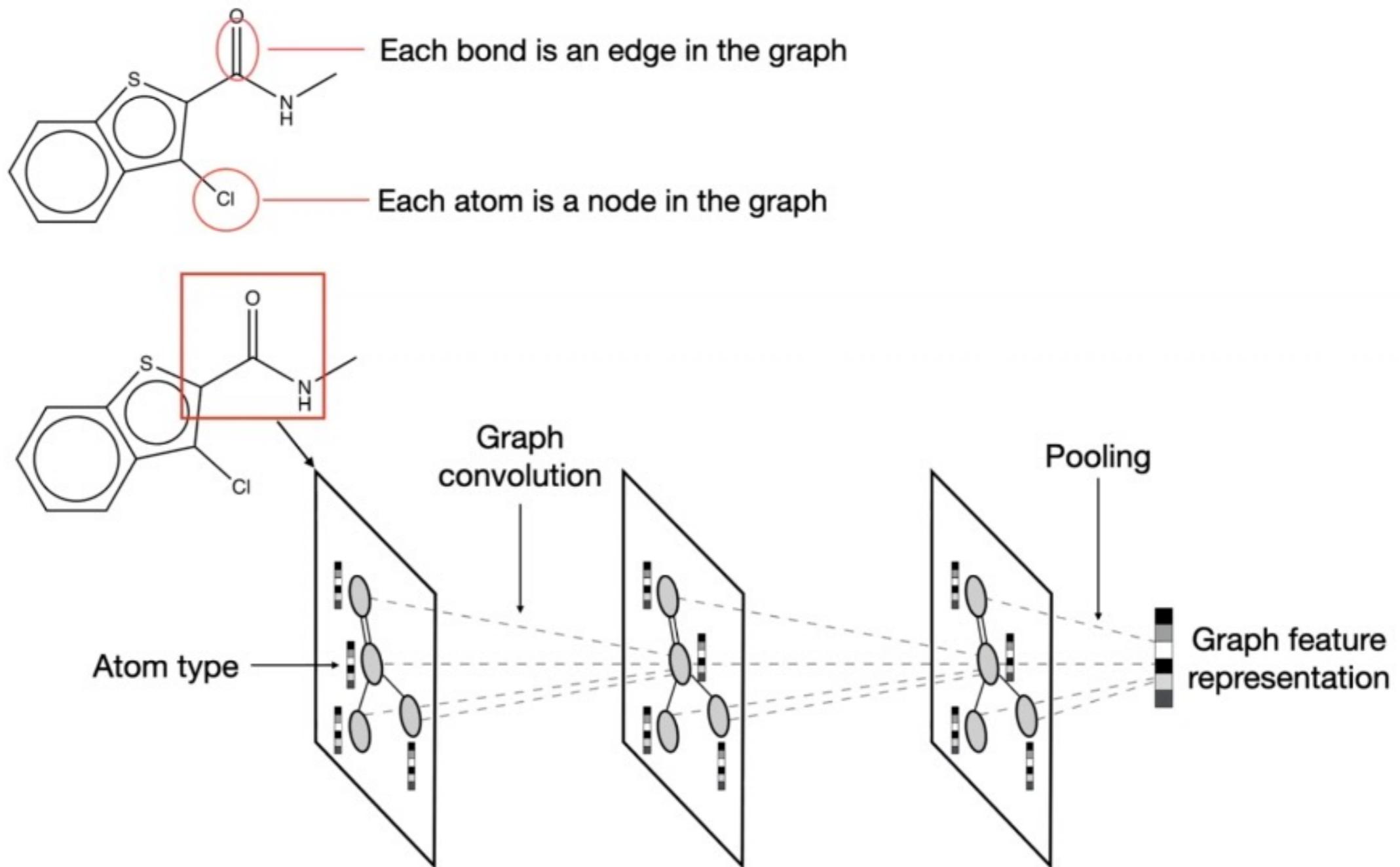
- ▶ 복잡한 구조의 graph structure를 단순하게 만드는데 사용한다
  - ▶ feature에서 max값만 뽑거나 평균을 내어서 low dimension으로 만드는 작업을 수행



- ▶ edge나 node의 정보를 서로 전달하고자 할 때도 사용할 수 있다
  - ▶ Edge별로 연결된 node의 정보(features)를 모은다.
  - ▶ 정보(features)를 합쳐서 edge로 보낸다. (pooling)
  - ▶ Parameter를 이용해서 classification을 수행한다. (prediction)

# GCN

- ▶ 3D 분자의 구조적 정보를 반영



# Message passing

- ▶ node 혹은 edge 간의 연관성을 고려하면서 feature를 업데이트하는 방법
- ▶ (예) node를 주변 node 정보를 이용해서 업데이트하고 싶을 때 message passing은 다음과 같이 이루어진다
  - ▶ Edge로 연결되어 있는 node의 정보(features, messages)를 모은다
  - ▶ 모든 정보를 aggregate function (sum, average 등)을 이용하여 합친다
  - ▶ Update function(parameter)을 이용해서 새로운 정보로 업데이트한다
- ▶ Message passing을 여러번 반복하여 receptive field를 넓힐 수 있고 더 좋은 representation을 얻는다

