

## 01 이론1 머신러닝과 딥러닝 개념

## 01 머신러닝과 딥러닝 개념

### ④ 오늘 학습을 통해 우리는

- 인공지능의 범주와 머신러닝, 딥러닝 간의 관계를 알아봅니다.
- 머신러닝의 기본적인 절차와, 이 과정 중 사용되는 개념에 대하여 복기합니다.
- 딥러닝의 학습 방법과, 이 때 일어날 수 있는 문제점을 알아봅니다.



# 목차

머신러닝과 딥러닝  
개념

01 인공지능 개요

02 머신러닝 학습

03 딥러닝 기초

01

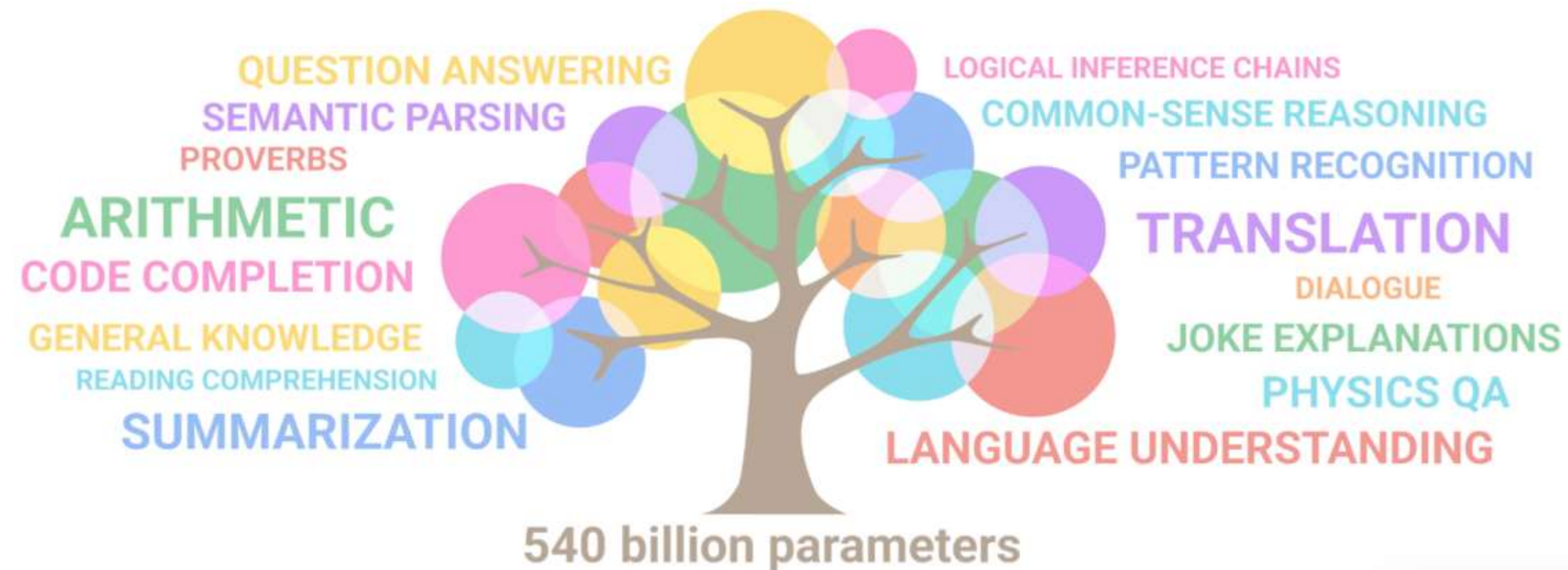
# 인공지능 개요

## 01 인공지능 개요

### 📌 2023년의 키워드: LLM(Large Language Model)

- Large

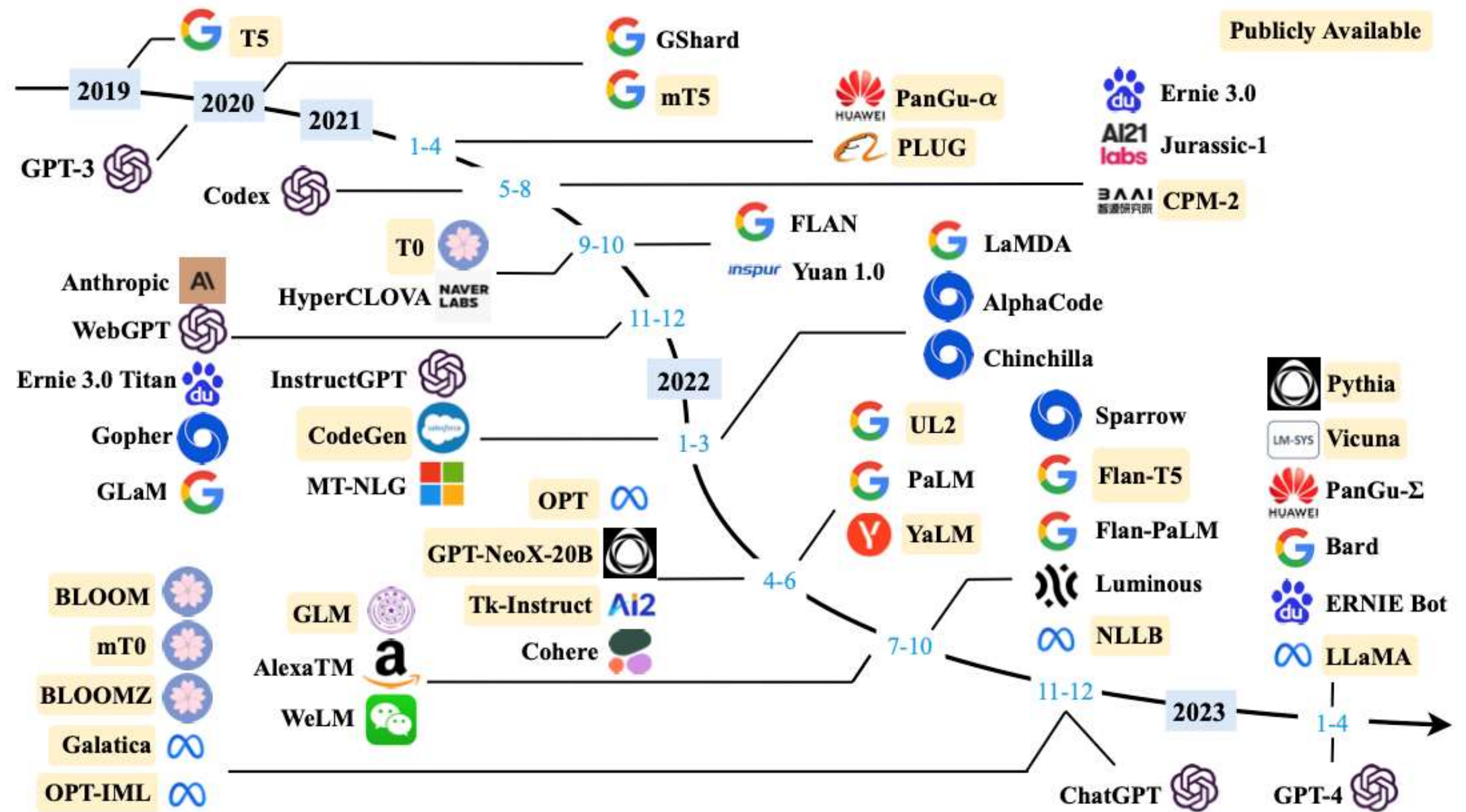
- 대용량 데이터셋
- 대형 구조(파라미터)
- 긴 학습시간



## 01 인공지능 개요

### 📌 2023년의 키워드: LLM(Large Language Model)

- 수많은 LLM의 등장
- 이를 활용한 다양한 서비스
- 다양한 도메인에서 활용





## 01 인공지능 개요

### ☑ 2023년의 키워드: LLM(Large Language Model)

- OpenAI: GPT-4(2023 3월)
- Google: PaLM2(2023년 5월)
- Meta: LLaMA2(2023년 7월)



## 01 인공지능 개요

### ④ LLM을 공부하기 위한 기초 지식

- Programming: Linux, Python, DL Frameworks(Pytorch, Hugging Face)
- Math: Linear algebra, Statistics, Calculus, Probabilistics
- Machine learning: Supervised / Unsupervised / Semi-supervised learning
- Deep learning: RNN, LSTM, Attention mechanism, Transformer
- NLP: Tokenization, Word embedding, Semantics



## 01 인공지능 개요

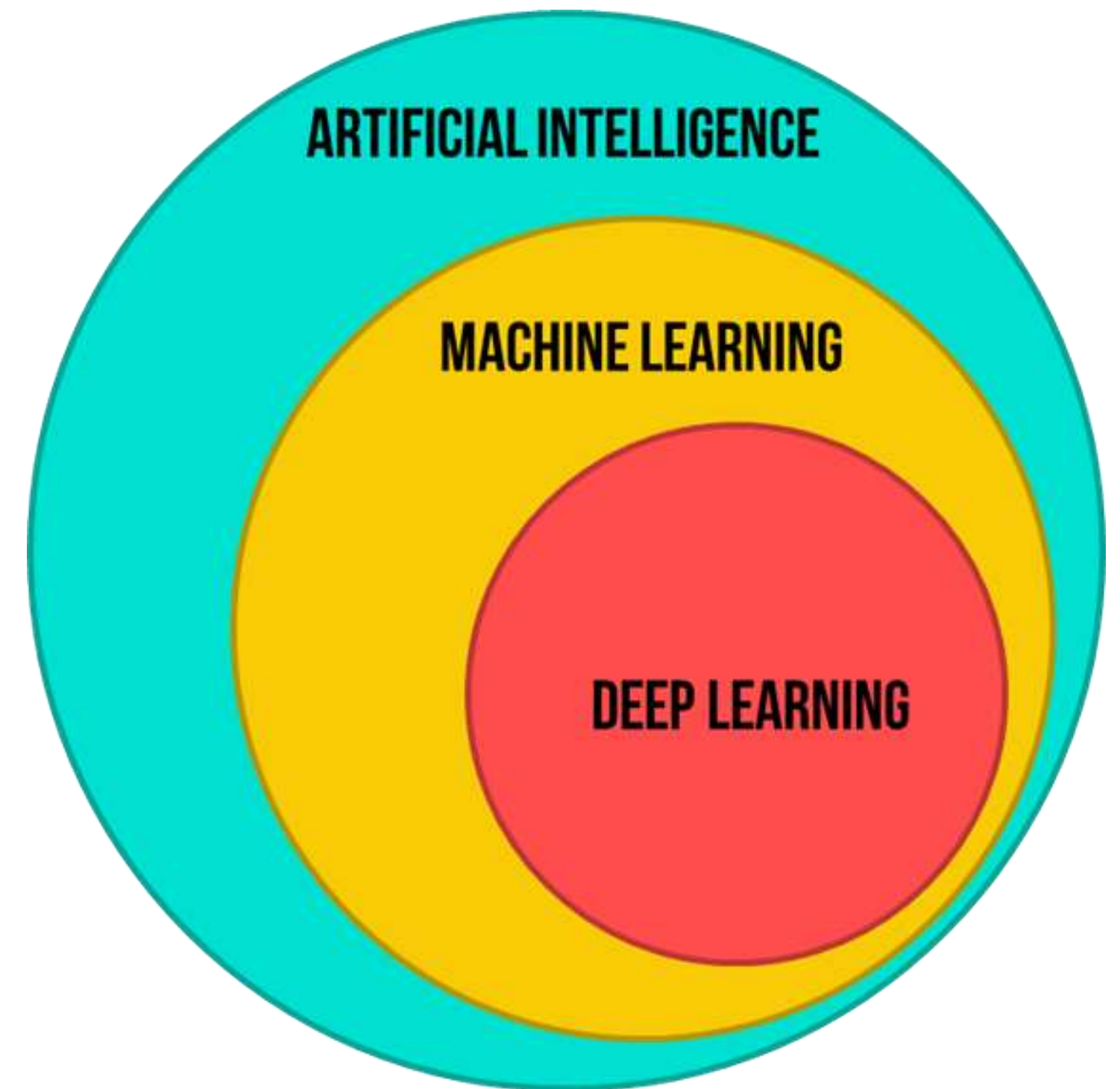
### ④ 인공지능이란?

- 지능: 학습, 추론, 지각
- 인공: 기계적인 구현
- 약(Weak)인공지능
- 강(Strong)인공지능 or 인공 일반 지능(Artificial General Intelligence, AGI)

## 01 인공지능 개요

### ☑ 인공지능, 머신러닝, 딥러닝

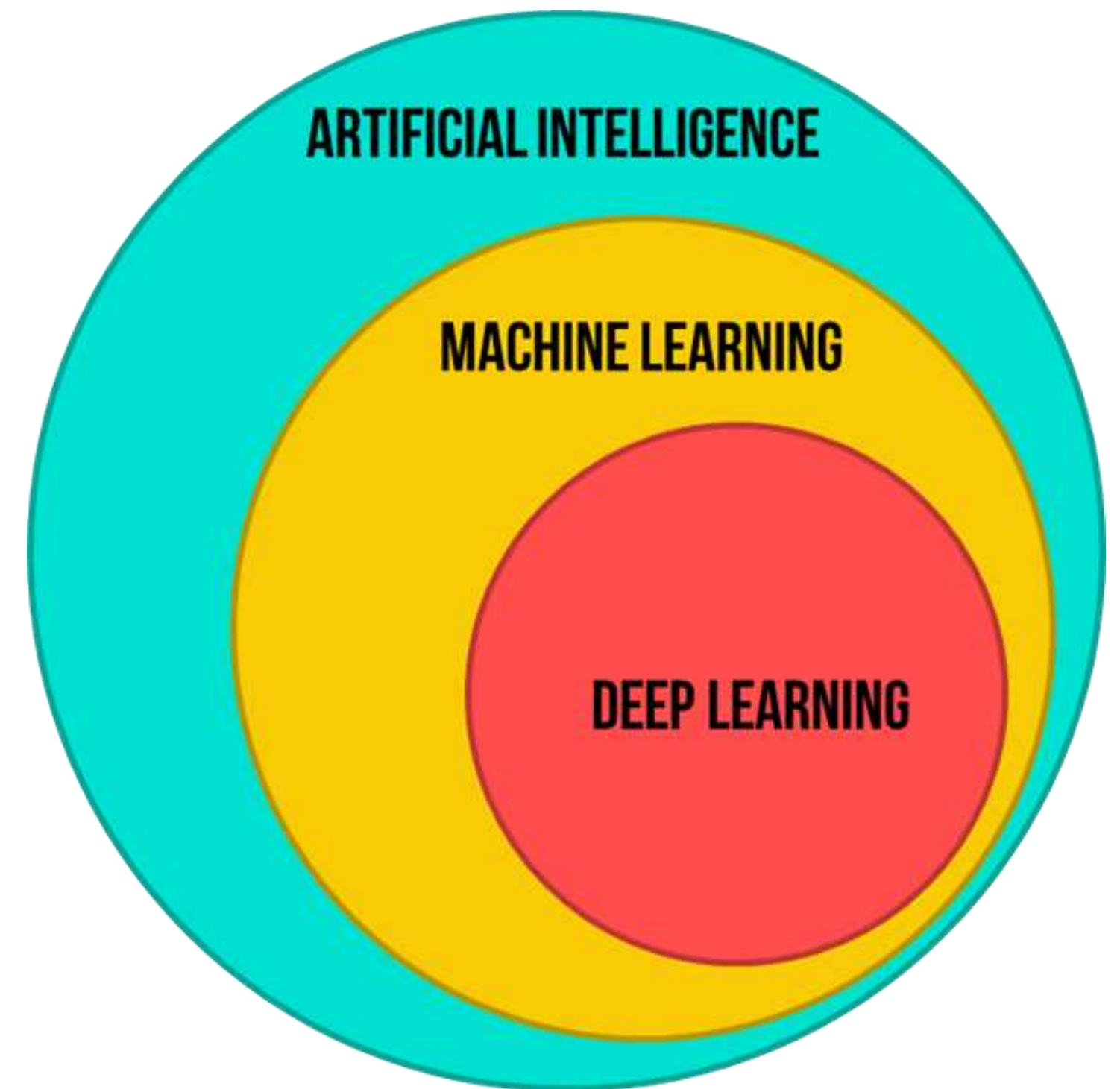
- 세 개념은 서로 포함관계
- 딥러닝은 머신러닝 알고리즘 중 하나
- 단, 일반적으로 좁은 의미의 머신러닝과 딥러닝은 구분됨



## 01 인공지능 개요

### ☑ 인공지능, 머신러닝, 딥러닝

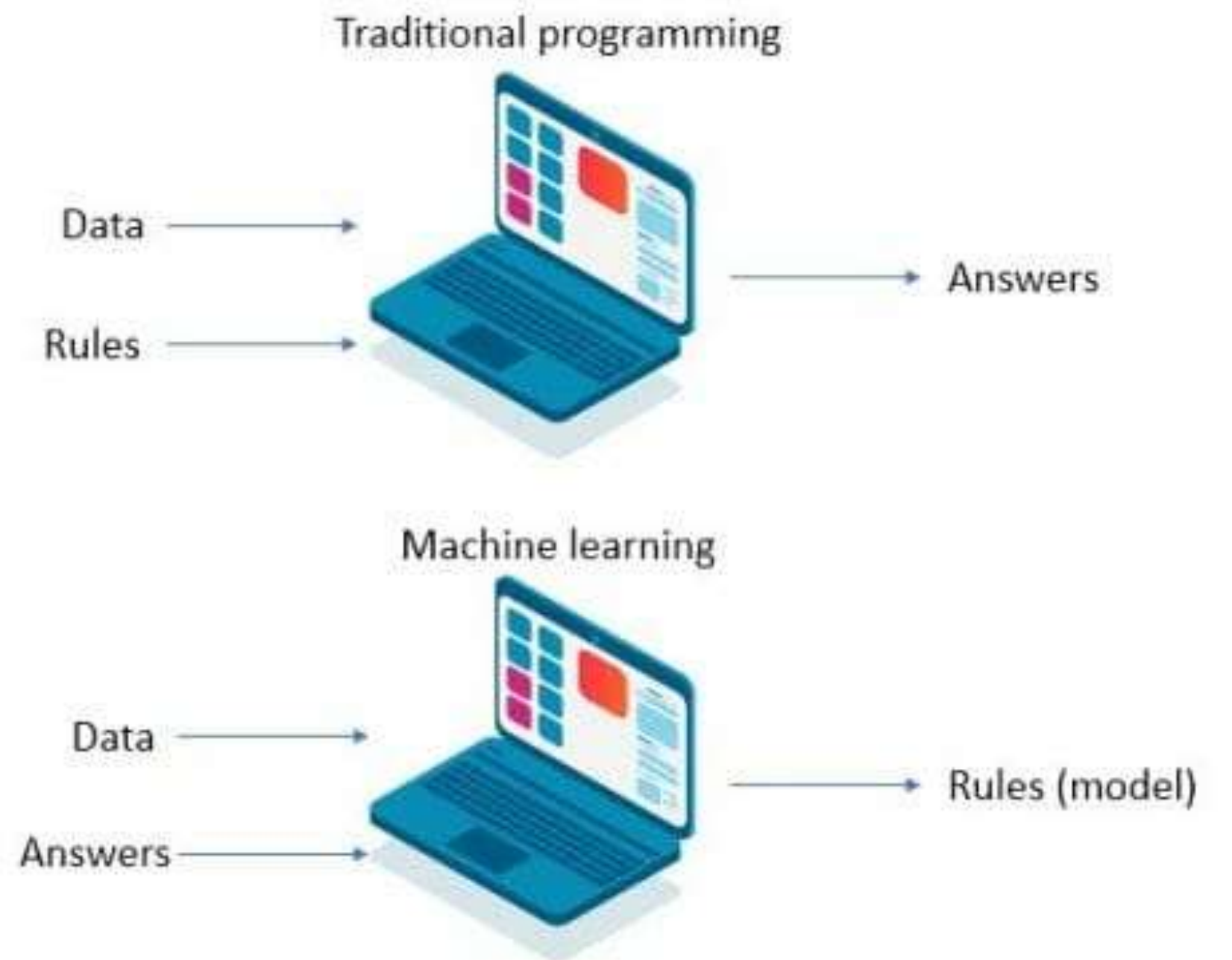
- AI: 알고리즘(인간/기계)을 기계가 스스로 처리
- ML: AI 중 데이터를 기반으로 학습한 알고리즘
- DL: ML 중 신경망(NN)을 사용한 알고리즘



## 01 인공지능 개요

### ④ 넓은 의미의 머신러닝

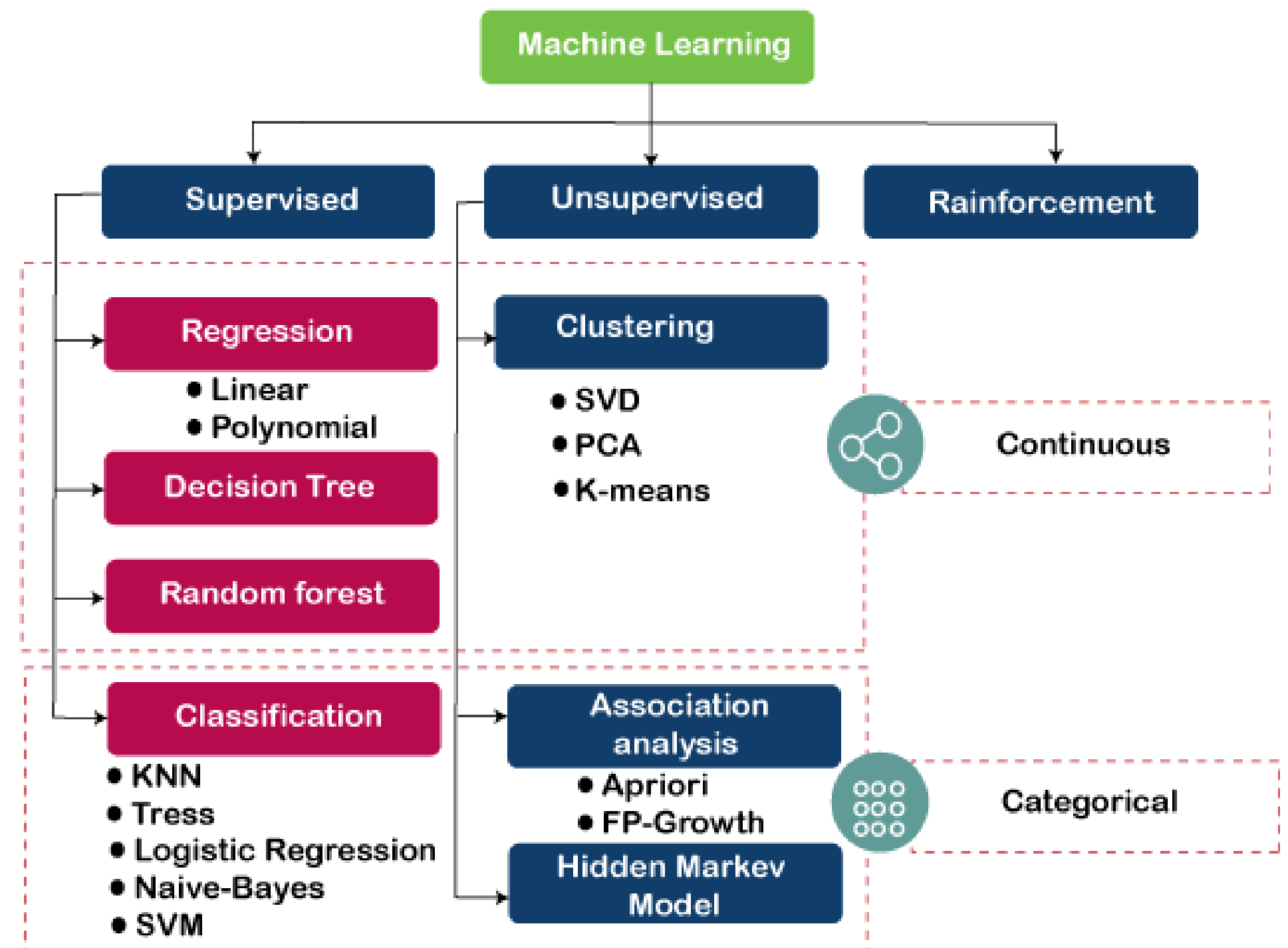
- 전통적 프로그래밍: 해답을 찾음
- 머신러닝: 데이터와 답을 바탕으로 풀이법을 찾음



## 01 인공지능 개요

### ☑ 좁은 의미의 머신러닝

- 전통적인 학습 알고리즘을 사용하는 AI
  - 고급 통계 기술을 사용
- 또는, 신경망을 사용하지 않는 알고리즘
- 분류: KNN, Tree, Logistic regression
- 회귀: Linear / polynomial regression



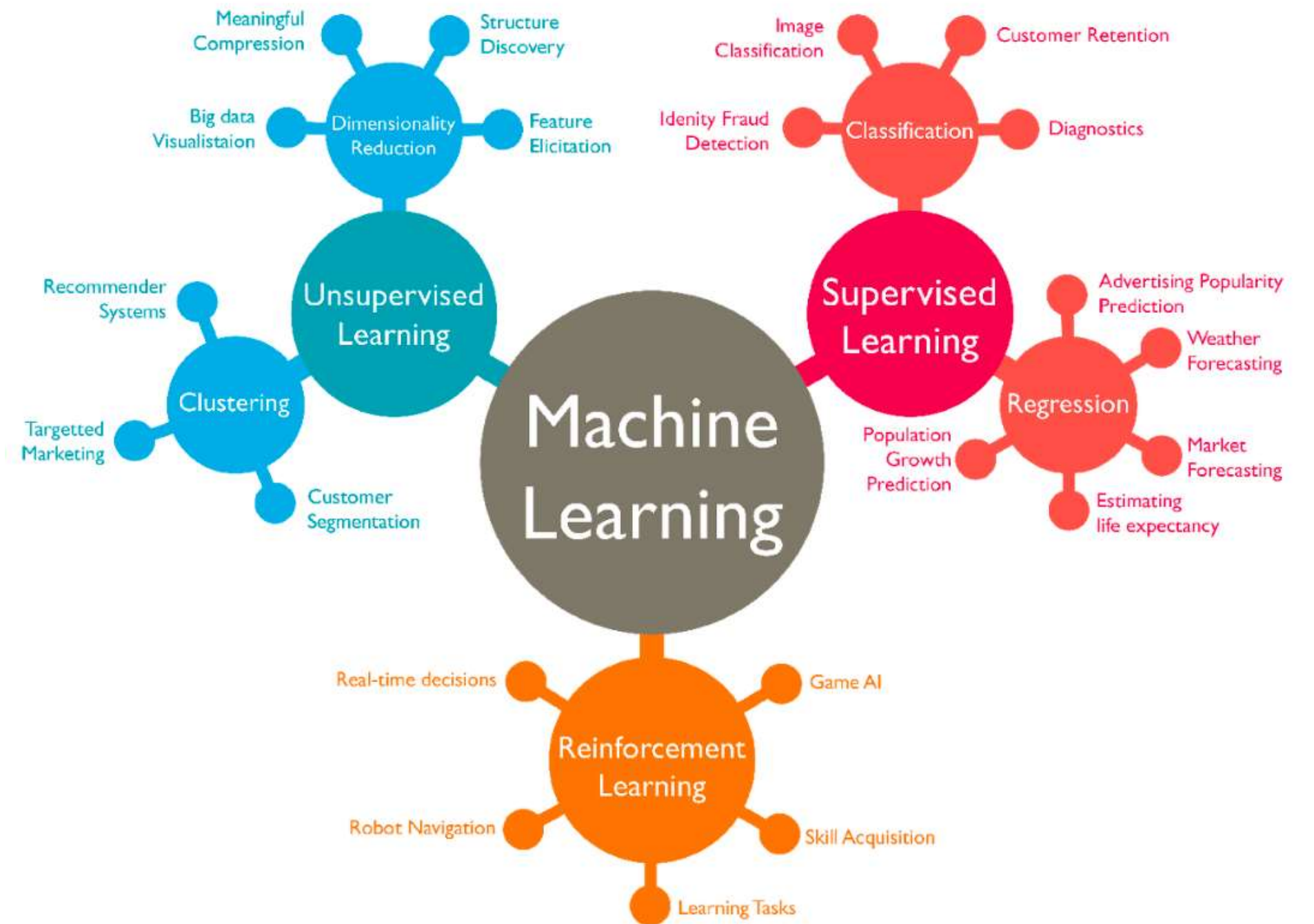


# 01 인공지능 개요

## ☑ 머신러닝의 학습 분류

### • 학습 방식에 따라

- 지도학습(Supervised learning)
  - 분류(Classification), 회귀(Regression)
- 비지도학습(Unsupervised learning)
  - 군집화(Clustering), 차원 축소(Dimensionality reduction)
- 강화학습(Reinforcement learning)





## 01 인공지능 개요

### ④ 지도 학습(Supervised Learning)

- 주어진 데이터를 바탕으로 학습하여 Label(정답)을 맞춤
- Train / test 데이터를 구분하여 사용함
- 장점
  - 정답이 있으므로, 모델 성능 평가 및 정량적 측정이 쉬움
  - 데이터가 충분할 경우 학습이 매우 정확하게 진행됨
- 단점
  - Annotation 비용이 비싸고 어려움

# 01 인공지능 개요

## ☑ 지도 학습(Supervised Learning)

### • Classification(분류)

- 범주를 맞춤(이산변수)
- 이진분류, 다중분류
- Ex) 품종 분류, 스팸메일 구분, 질병 진단

### • Regression(회귀)

- 값을 맞춤(연속변수)
- 집값 예측, 기온 예측 등



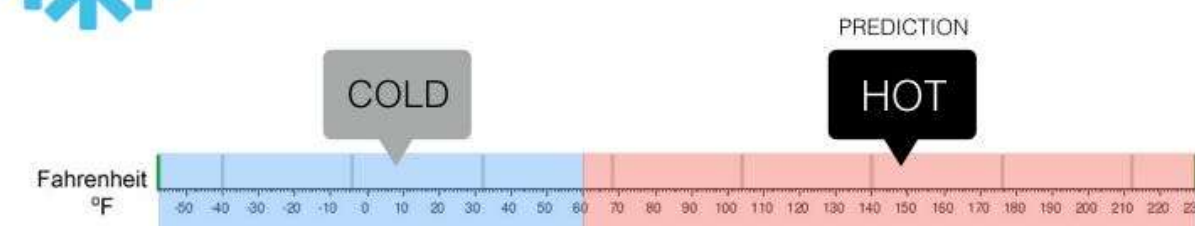
#### Regression

What is the temperature going to be tomorrow?



#### Classification

Will it be Cold or Hot tomorrow?



## 01 인공지능 개요

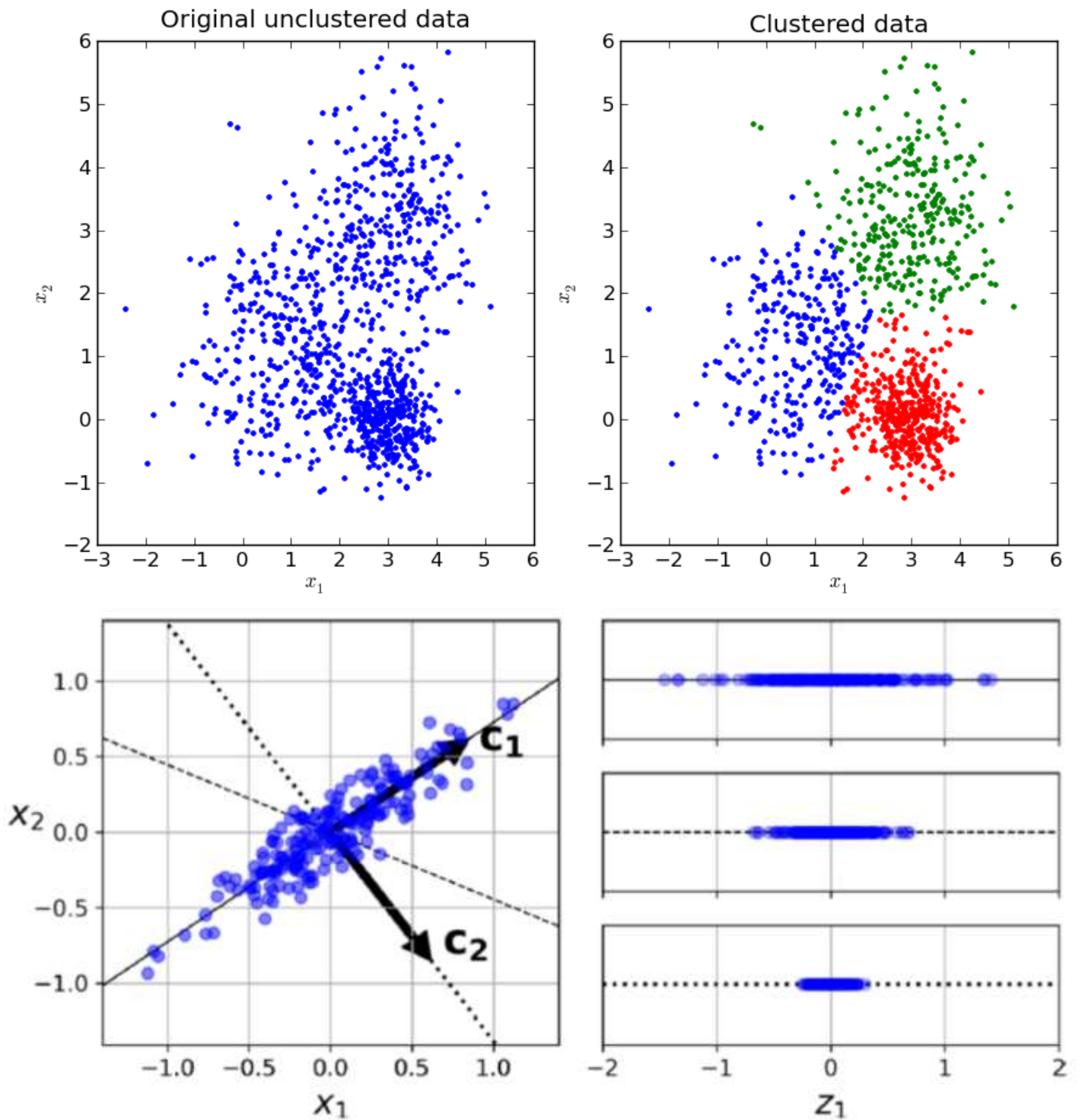
### ④ 비지도 학습(Unsupervised Learning)

- Label 없이 입력 데이터만을 사용하여 패턴을 찾아내는 알고리즘
- 모델이 데이터 내에서, 또는 데이터 간 상관관계, 특징 등을 파악하여 학습
- 장점
  - Annotation 비용이 없거나 저렴
  - 인간이 찾아내지 못한, 데이터에서 숨은 특징이나 패턴을 기계적 관점으로 파악
- 단점
  - 모델 학습 과정이나 성능의 정량적인 평가가 어려움

# 01 인공지능 개요

## ☑ 비지도 학습(Unsupervised Learning)

- Clustering(군집화)
  - 정답 없는 데이터를 기계의 재량으로 묶음
  - Label이 없는 분류
- Dimensionality reduction(차원 축소)
  - 고차원으로(복잡하게) 구성된 데이터들을 구분하기 위해 저차원으로 간소화
- 대다수의 거대 언어 모델이 비지도학습 활용



## 01 인공지능 개요

### ④ 강화 학습

- 에이전트(Agent)라고 불리는 학습 시스템이 환경과 상호작용하며, 시행착오를 통해 학습하고 보상을 최대화하기 위한 전략, 즉 "정책(policy)"을 배우는 것을 목표
- Agent: 학습 주체
- Environment: 학습 환경
- State: 상태
- Action: 행동
- Reward: 보상



## 01 인공지능 개요

### ☑ 강화학습

- 알파고가 대표적인 강화학습의 결과물
- 로봇 학습, 자율 주행 차량, 자원 관리, 통신 네트워크, 전력 시스템, 금융 거래 등 다양한 분야에서 응용

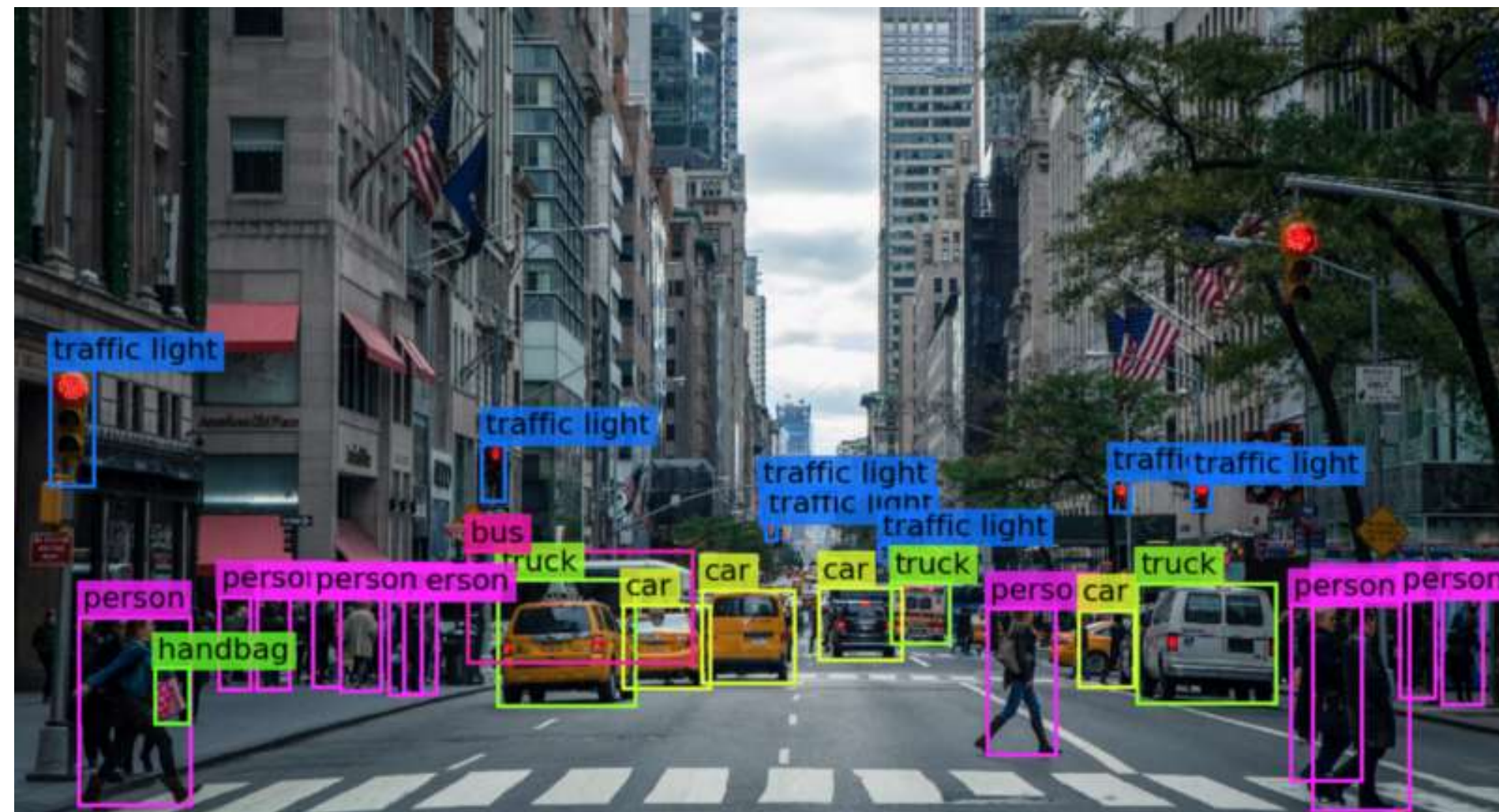




# 01 인공지능 개요

## ☑ 머신러닝 분야

- Computer Vision(CV)
  - 이미지, 비디오 등에서 유용한 정보를 추출
  - 인간의 시각 능력을 모방



출처 : <https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e>

# 01 인공지능 개요

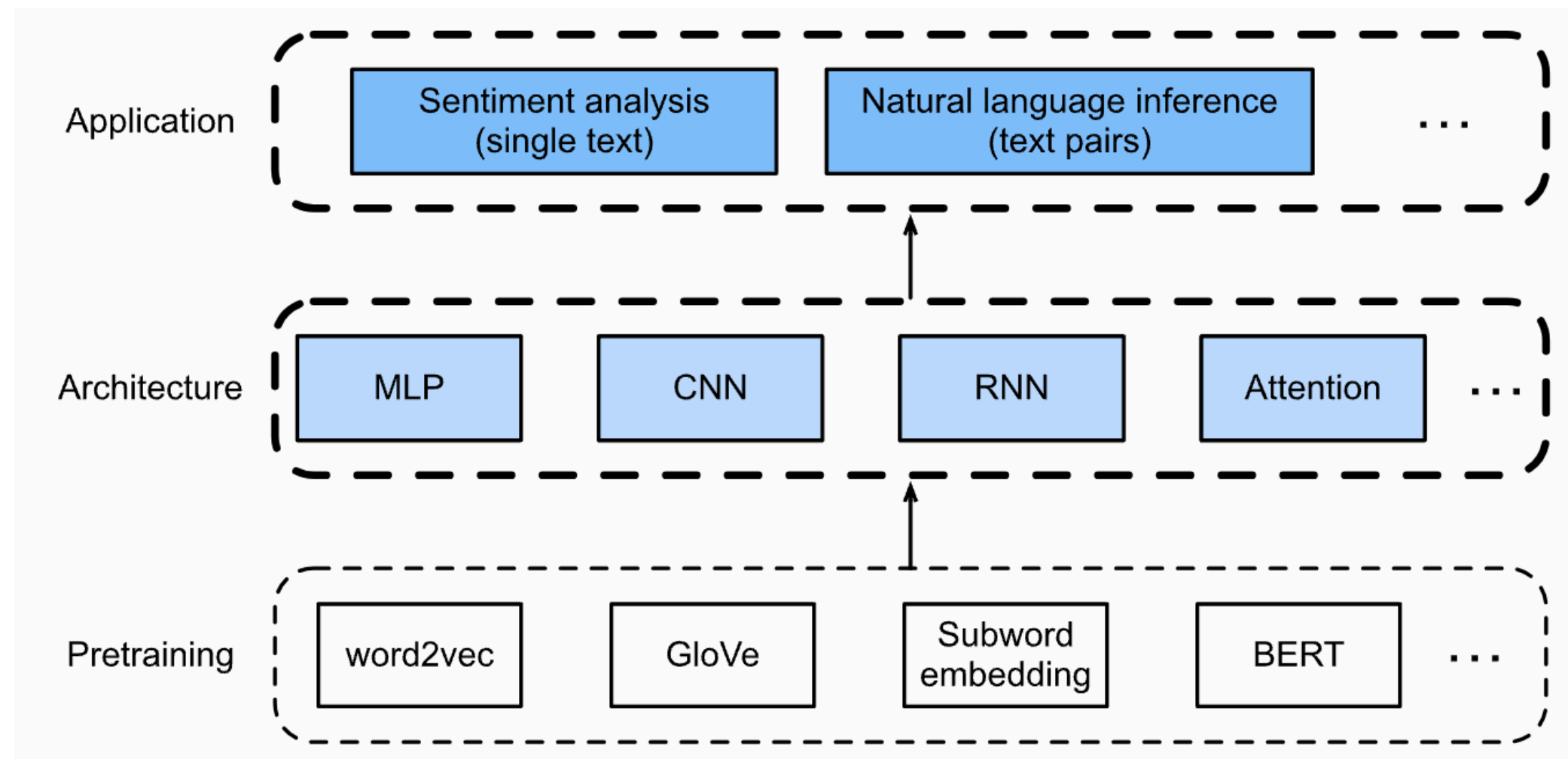
## ④ 머신러닝 분야

- 주요 Task
  - 이미지 분류 Image classification
  - 객체 인식 Object detection
  - 이미지 생성 Image generation
  - 광학 문자 인식 OCR
  - 이미지 분할 Image segmentation
  - 포즈 추출 Pose estimation

# 01 인공지능 개요

## ☑ 머신러닝 분야

- Natural Language Process(NLP)
  - 기계로 인간의 언어를 이해하고 처리
  - 텍스트 데이터를 처리하고 분석



# 01 인공지능 개요

## ④ 머신러닝 분야

- 주요 Task
  - 감성 분석(Sentiment analysis)
  - 문서 분류(Document classification)
  - 개체명 인식(NER, Named Entity Recognition)
  - 텍스트 요약(Text summarization)
  - 기계 번역(Machine translation)
  - 대화 시스템(Dialogue system)
  - 텍스트 랭킹(Text ranking)

## 01 인공지능 개요

### ④ 머신러닝 분야

- 음성 인식(Speech recognition)
- 추천 시스템(Recommendation system)
- 시계열 분석(Time series analysis)
- 이상 탐지(Anomaly detection)
- 생체 인식(Biometrics)

# 01 인공지능 개요

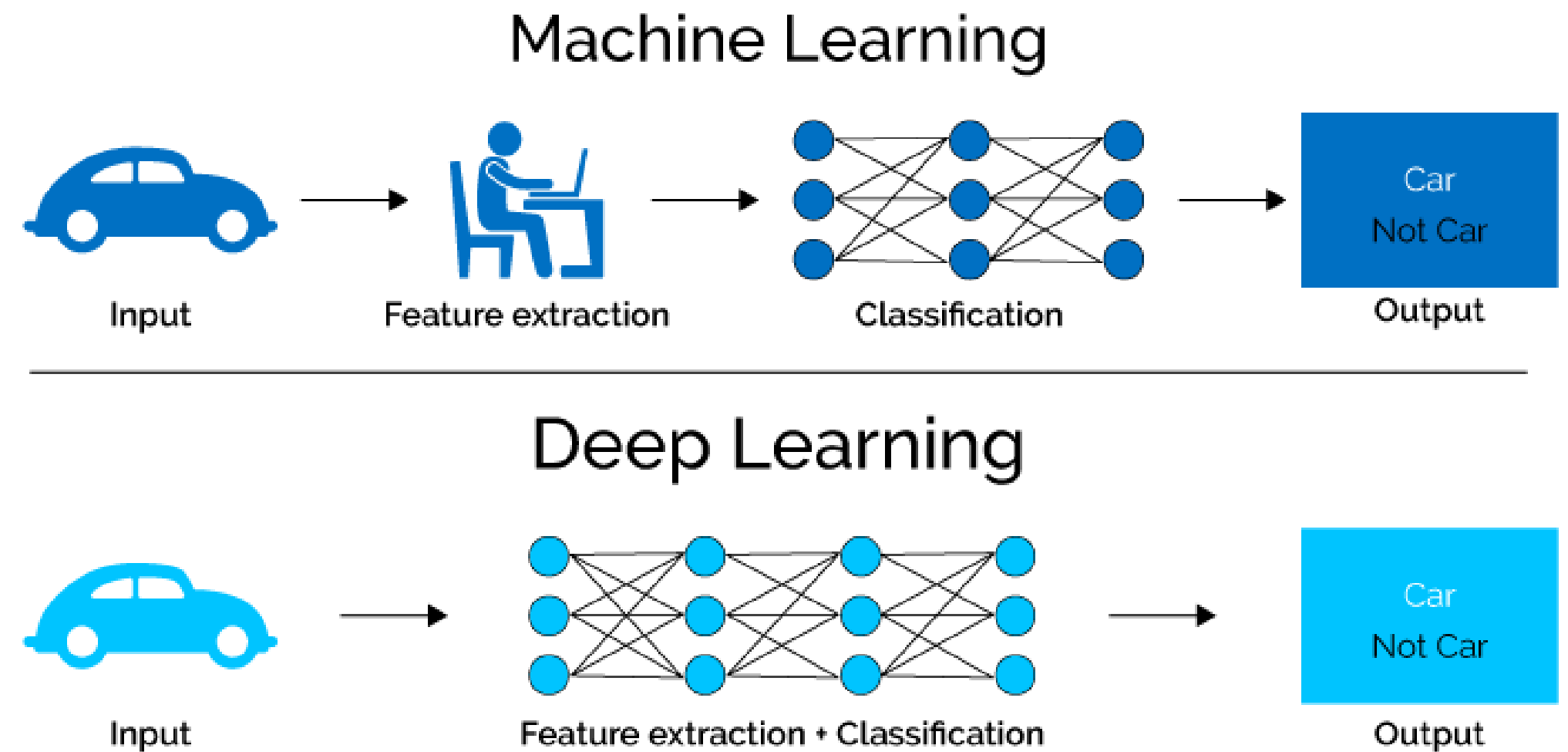
## ④ 좁은 의미의 ML과 DL

### • Traditional ML

- Feature engineering과 모델 선별 과정에 **전문가의 개입**이 필요
- 문제를 **분할**하여 해결 후 **결과를 결합**

### • DL

- **End-to-end** 학습
- **많은 파라미터를 사용**하여 고수준의 정보를 학습





# 01 인공지능 개요

## ④ 좁은 의미의 ML과 DL

- ML을 써야할 때
  - 데이터의 양이 적을 때
  - 해석 가능성이 중요할 때: Tree, Logistic regression 등 모델은 해석하기가 쉬움
  - 계산에 필요한 장비와 리소스가 제한적일 때

## 01 인공지능 개요

### ④ 좁은 의미의 ML과 DL

- DL을 써야할 때

- 데이터의 양이 많을 때: DL은 대량의 데이터에서 복잡한 패턴을 잘 학습함
- 데이터가 고차원, 비정형일 때: 이미지, 오디오, 텍스트
- 해결해야하는 문제가 복잡하고 복합적일 때

02

# 머신러닝 학습

## 02 머신러닝 학습

### ④ 머신러닝 학습 과정 개요

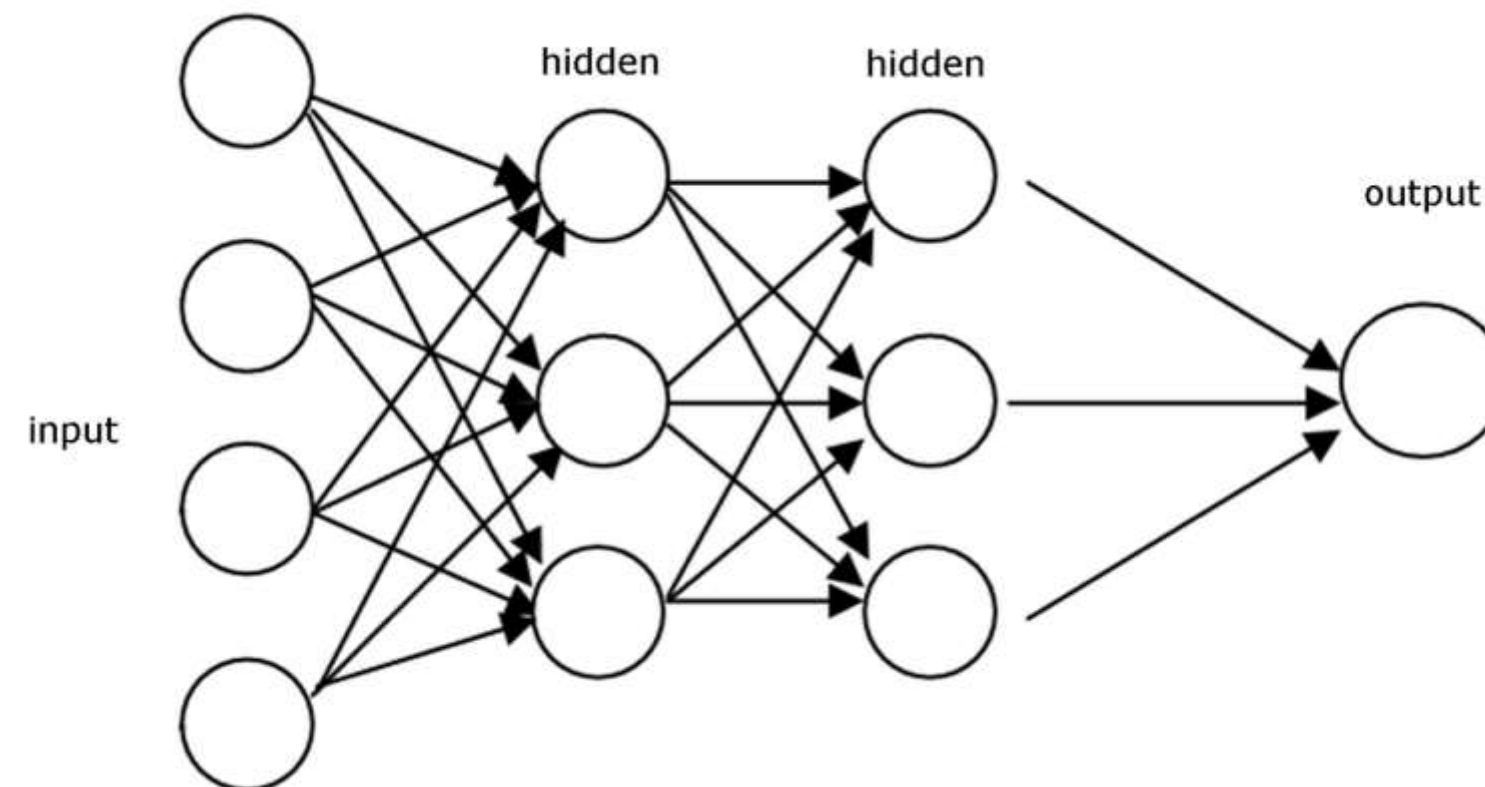
- 머신러닝의 학습 과정은 크게 세 가지 단계 구성
  - 순전파(Forward Propagation)
  - 손실계산(Loss Calculation)
  - 역전파(Backpropagation).
- 이 세 단계는 모델이 데이터로부터 학습하고, 예측 오류를 최소화하는 방향으로 파라미터를 업데이트하는 데 필수적

## 02 머신러닝 학습

### ④ 순전파(Forward Propagation)

- 순전파의 정의

- 순전파는 입력 데이터가 모델의 각 계층을 거쳐 예측값( $y_{pred}$ )을 생성하는 과정
- 여기서 계층은 신경망의 레이어를 의미함
- 이 과정에서 모델은 파라미터(가중치와 편향)를 사용하여 예측을 수행함



## 02 머신러닝 학습

### ④ 순전파(Forward Propagation)

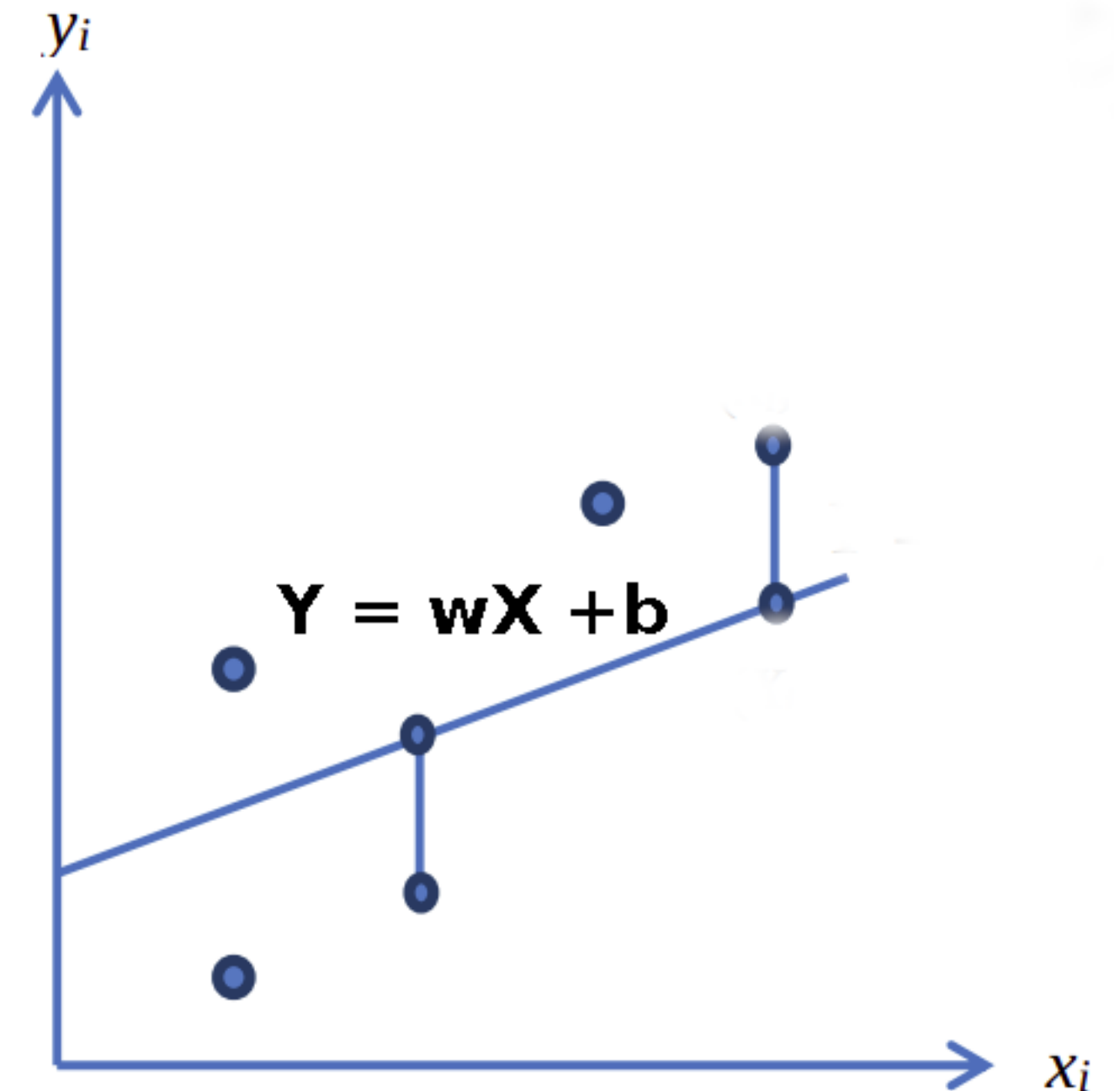
- 입력 데이터
  - 머신러닝 모델이 예측을 수행하기 위해 필요한 정보
  - 종종 여러 특징(Features)으로 구성된 벡터 형태로 제공됨.
- 가중치와 편향
  - 모델의 학습 가능한 파라미터로, 모델의 출력을 조정하는 데 사용됨
  - 초기에는 무작위 값으로 설정되지만, 학습 과정에서 점차 최적화됨.



## 02 머신러닝 학습

### ☑ 파라미터(Parameter)

- 모델의 학습 가능한 부분
  - $w$ : Weight
  - $b$ : Bias
- 이들의 값을 변경함으로써 모델은 학습 데이터로부터 패턴을 학습하고, 예측 성능을 개선함
- 파라미터 값은 학습 과정에서 업데이트되며, 이 업데이트는 역전파 과정에서 이루어짐

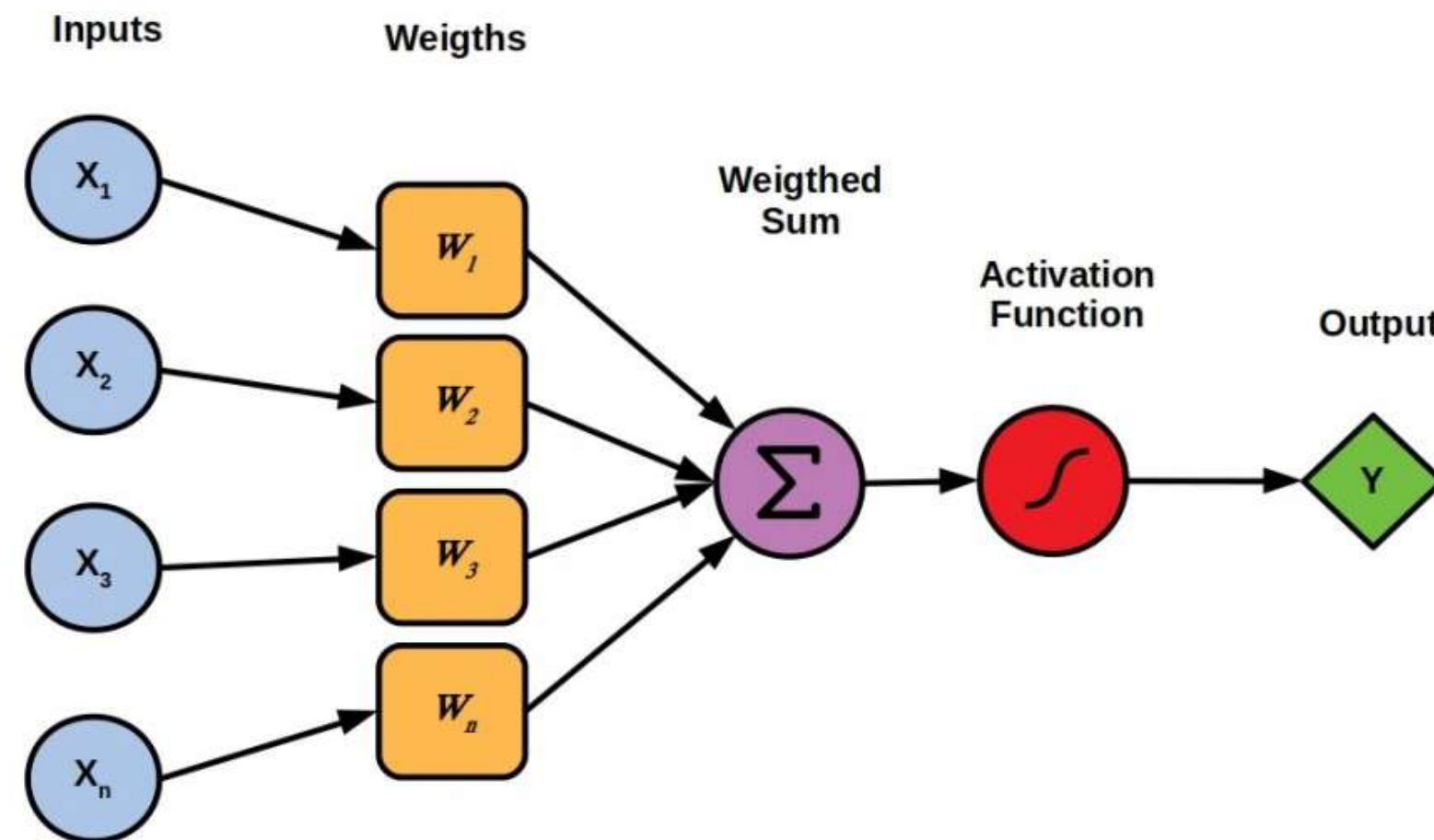


## 02 머신러닝 학습

### ④ 순전파의 과정

- 계층별 연산:

- 각 계층에서는 입력 데이터와 파라미터를 연산하여 결과를 생성함
- 대부분의 경우, 이 연산은 단순한 행렬 곱셈과 덧셈으로 이루어짐( $wx+b$ )



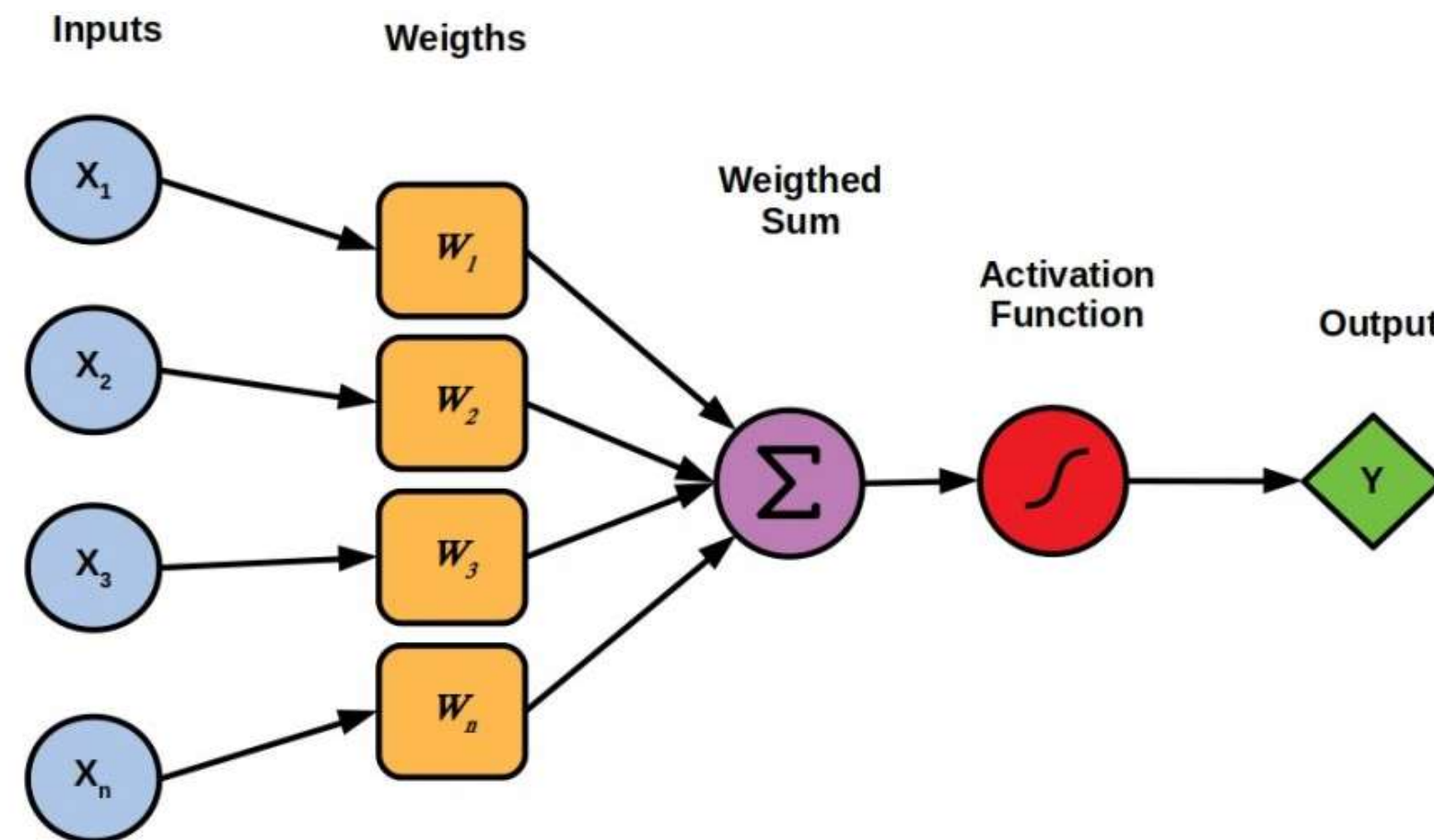
## 02 머신러닝 학습

### ④ 순전파의 과정

- 활성화 함수

- 각 계층의 출력은 활성화 함수를 통과

- 모델의 **중간(은닉층)**에 쓰이는 활성화 함수: **비선형성**을 추가하여 **복잡한 패턴**을 학습
    - 모델의 **출력층**에 쓰이는 활성화 함수: **출력 값**을 특정 범위 (예: 0과 1 사이)로 제한하는 기능

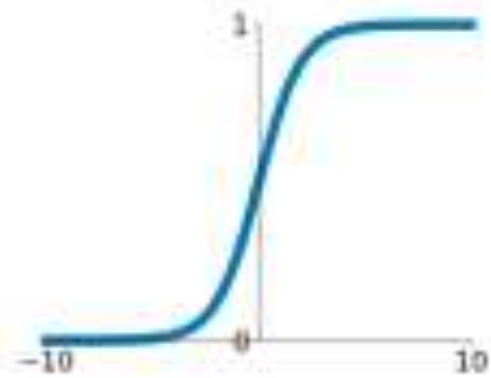


### ☑ 활성화 함수(Activation function)

# Activation Functions

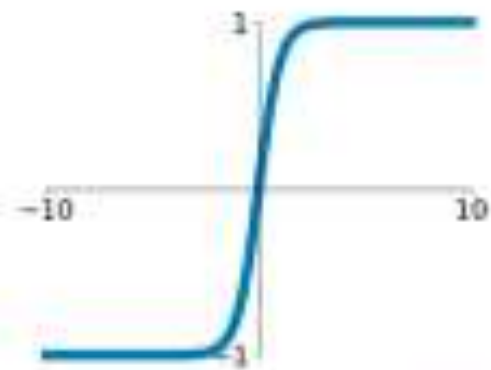
## Sigmoid

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



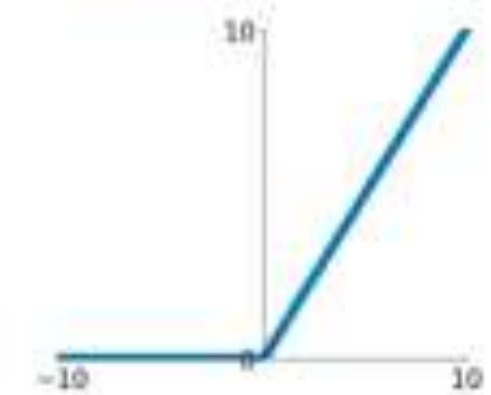
## tanh

$$\tanh(x)$$



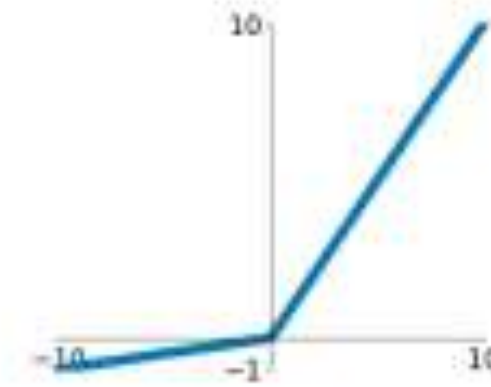
## ReLU

$$\max(0, x)$$



## Leaky ReLU

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

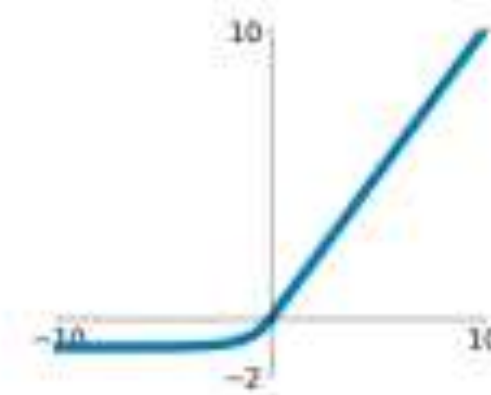


## Maxout

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

## ELU

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

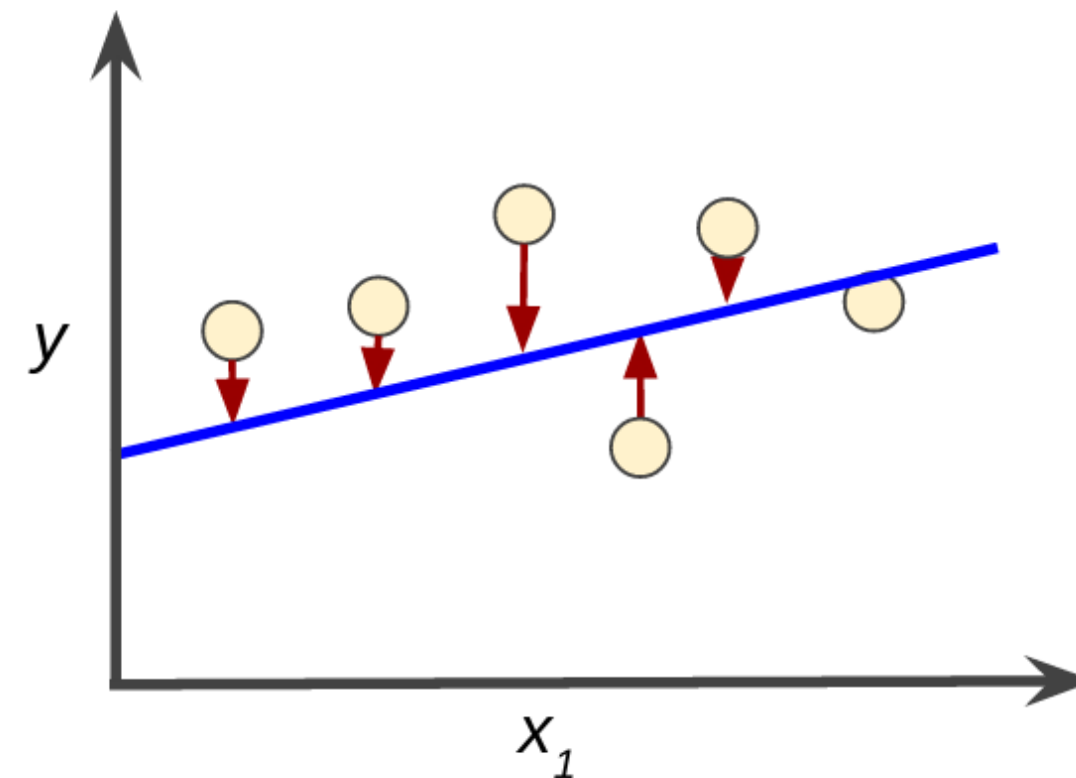
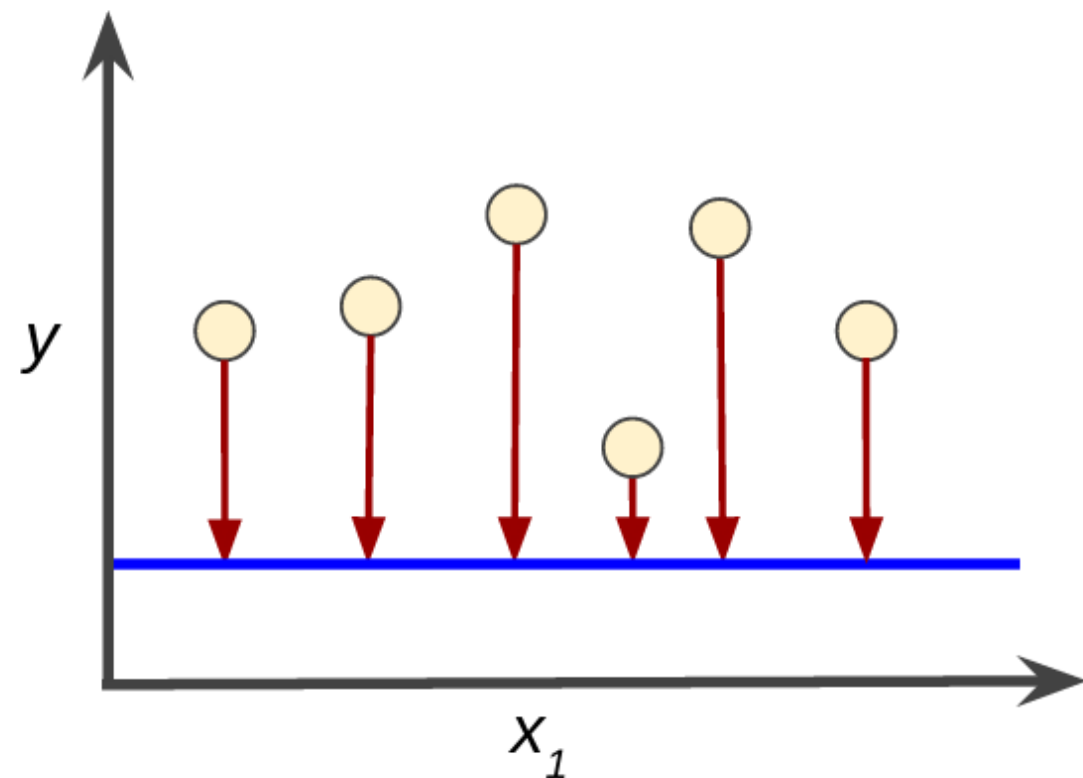


## 02 머신러닝 학습

### ④ 손실(Loss) 계산

- 손실계산의 정의

- 손실계산은 모델의 예측 결과( $y_{pred}$ )와 실제 값( $y_{true}$ )과의 차이, 즉 '오차'를 계산하는 과정.
- 이를 위해 손실 함수(Loss Function)를 사용
- 손실 값은 모델의 성능을 평가하는 데 사용되기도 함



## 02 머신러닝 학습

### ④ 손실 함수(Loss Function)의 역할

- =Objective function = Cost function
- 손실 함수는 예측 값과 실제 값의 차이를 수치화
- 모델은 손실함수의 결과(손실)값이 최소화하는 방향으로 학습되게끔 설계됨
- 손실 함수를 잘 설계했다는 가정 하에, 손실 값이 작을 수록 모델의 성능이 우수함



## 02 머신러닝 학습

### ④ 손실 값의 활용

- 손실 값은 **파라미터 업데이트**에 사용됨
  - 손실 값이 **최소화되는 방향**으로 파라미터가 개선됨
- 손실 값으로 **모델의 성능이 평가**됨
  - 손실 값이 줄어들면, 모델의 **예측 성능이 개선**되었다는 의미

## 02 머신러닝 학습

### ④ 다양한 손실 함수

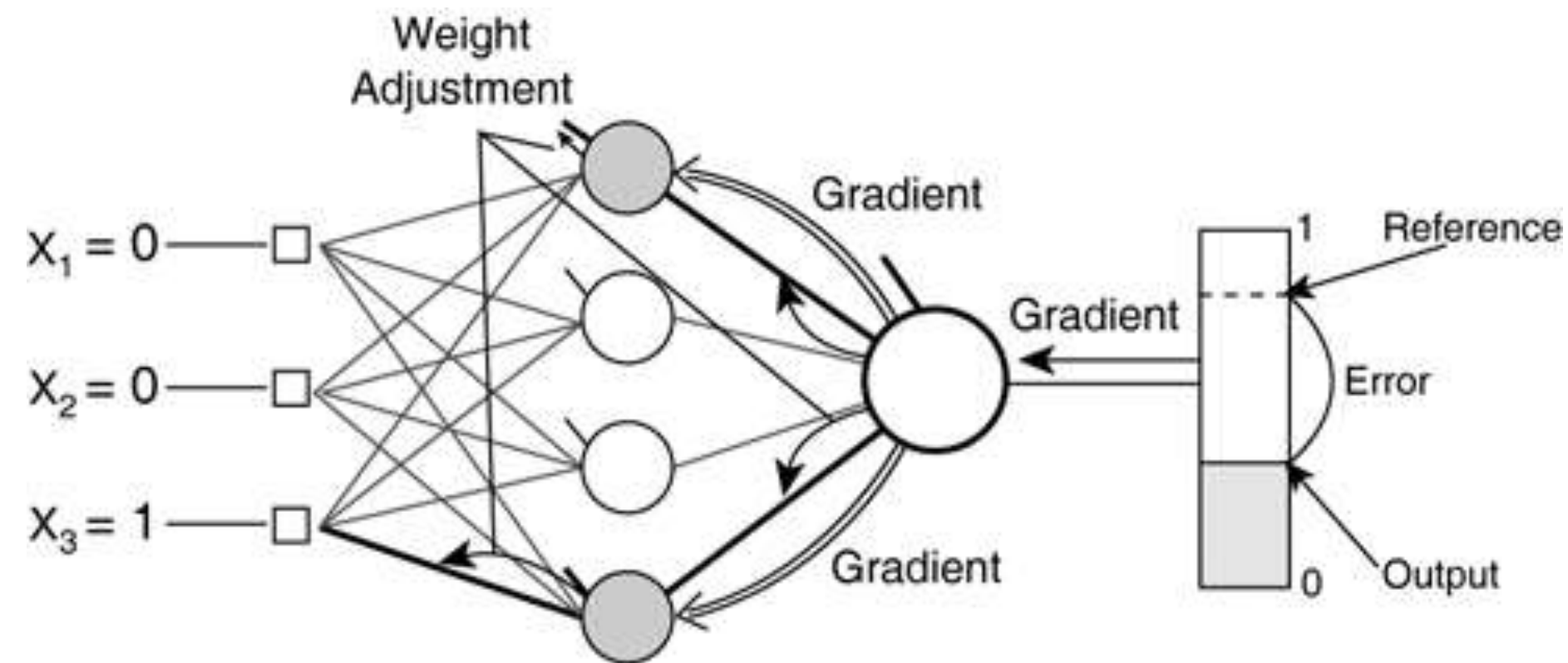
- 손실 함수는 모델의 학습 방향을 결정함
- 그러므로, 문제의 성격에 따라 사용되는 함수의 종류가 달라짐
  - 회귀(Regression): MSE(Mean Squared Error)
  - 분류(Classification): Cross-Entropy loss
  - 이미지 분할(Image segmentation): Jaccard loss, Dice loss, Focal loss
  - QA, 추천 시스템: Rank loss, Cosine similarity

## 02 머신러닝 학습

### ④ 역전파(Backpropagation)

#### • 역전파의 정의

- 역전파는 손실 값을 줄이기 위해 **각 파라미터에 대한 손실 함수의 기울기를 계산**하고, 이를 바탕으로 **파라미터를 업데이트**하는 과정.
- 모델이 학습하면서 더 나은 예측 값을 출력하기 위해 **자신의 오차를 개선**하는 방법



## 02 머신러닝 학습

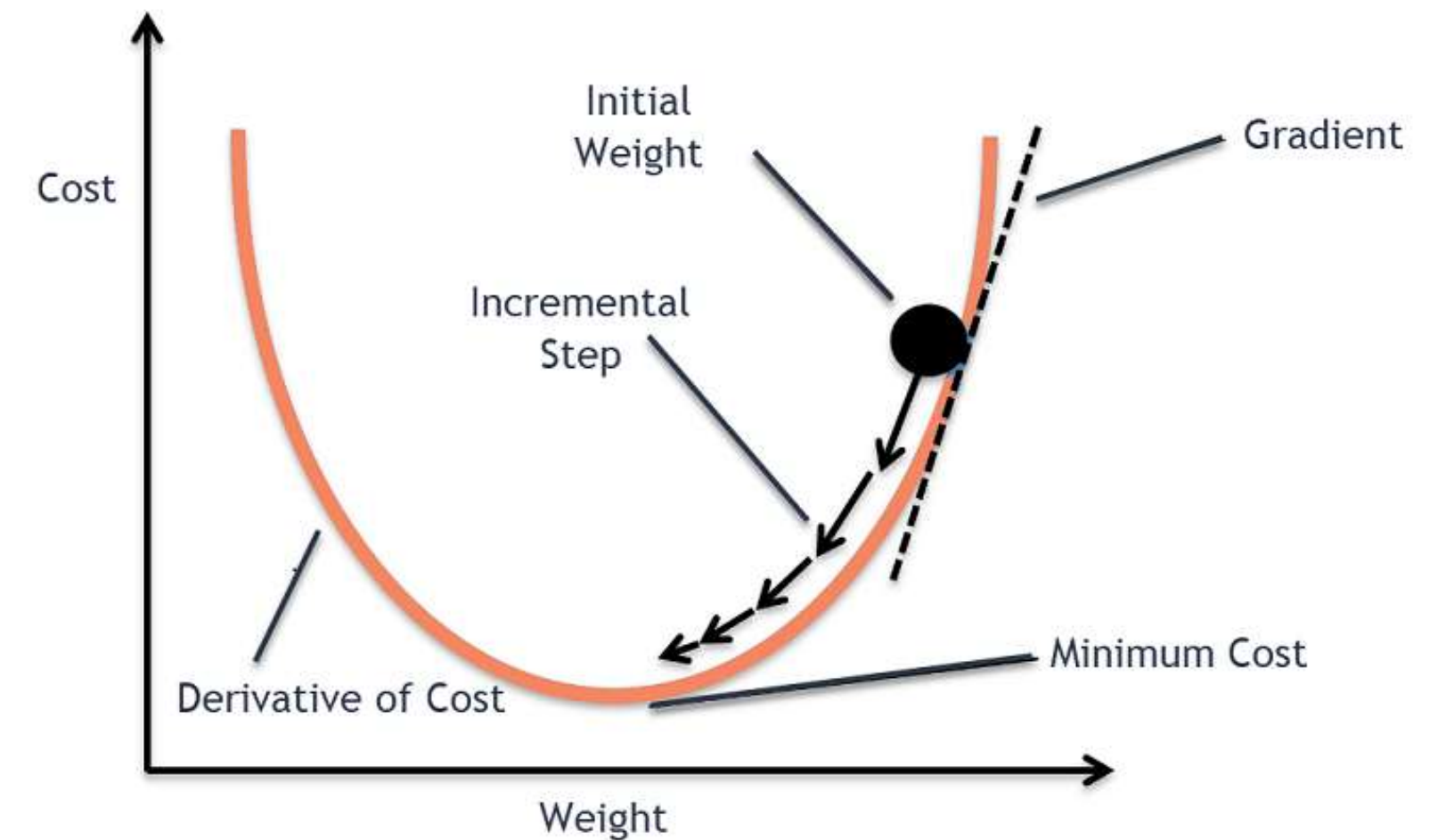
### ④ 역전파의 과정

- 손실 함수의 기울기 계산
  - 각 파라미터에 대한 손실 함수의 미분 값을 계산
  - 이 값은 해당 파라미터가 손실에 얼마나 큰 영향을 미치는지를 나타냄
- 파라미터 업데이트
  - 손실 함수의 기울기와 학습률(Learning Rate)을 곱한 값을 파라미터에서 빼는 방식으로 업데이트를 수행함
  - 이는 손실을 줄이는 방향으로 파라미터를 조정함
  - 손실이 최저가 되는 지점을 향해, 가장 기울기가 가파른 길을 선택

## 02 머신러닝 학습

### ④ 옵티마이저(Optimizer)

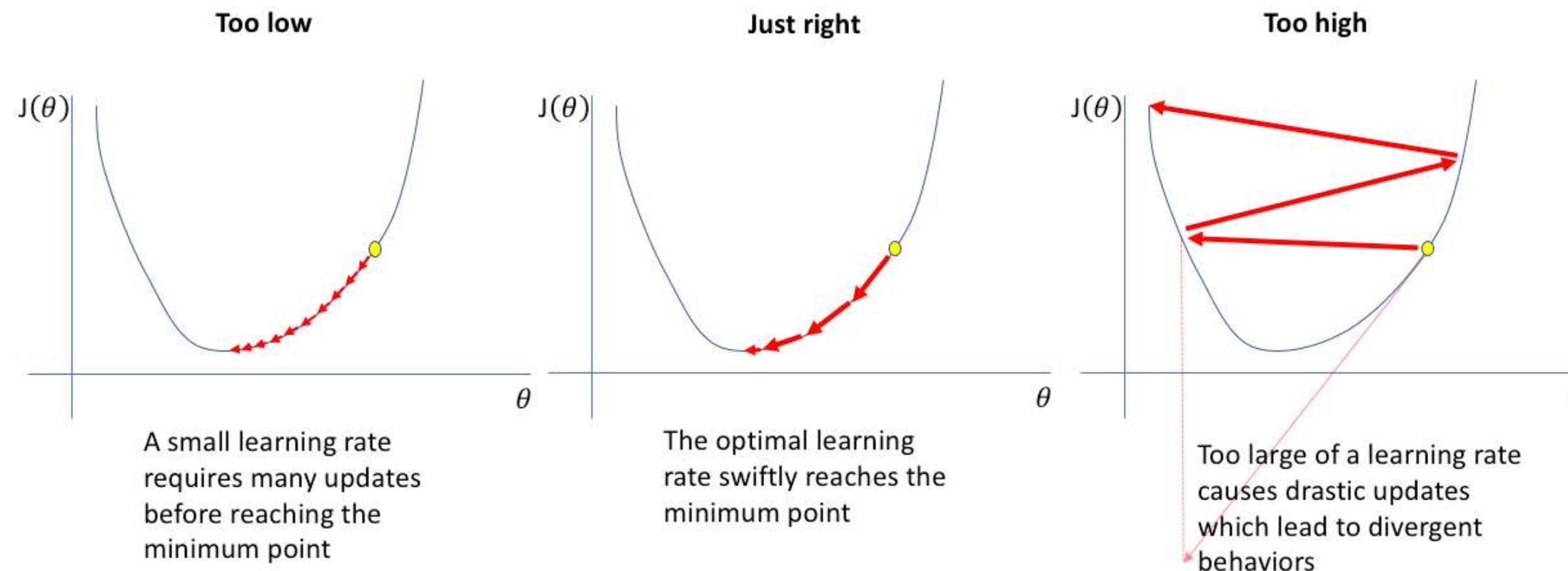
- 경사하강법(Gradient Descent): 손실이 최저가 되는 지점을 향해, 가장 기울기가 가파른 방향으로 파라미터를 업데이트하는 알고리즘
- 옵티마이저
  - 경사하강법 등 파라미터 업데이트 알고리즘
  - SGD, RMSProp, Momentum 등 다양한 방법이 존재
  - Adam이 가장 무난하게 좋은 성능을 보임



## 02 머신러닝 학습

### ④ 학습률(Learning Rate)

- 파라미터 업데이트의 '스텝 크기'를 결정
  - 값이 크면 학습은 빠르지만, 최적의 솔루션을 지나칠 위험이 있음
  - 값이 작으면 학습은 느리지만, 안정적으로 최적의 솔루션을 찾을 가능성이 높아짐
- 실험에서는 Learning rate scheduler를 사용하여, 최적의 학습률 값을 탐색



## 02 머신러닝 학습

### ④ 머신러닝에서 평가지표의 역할

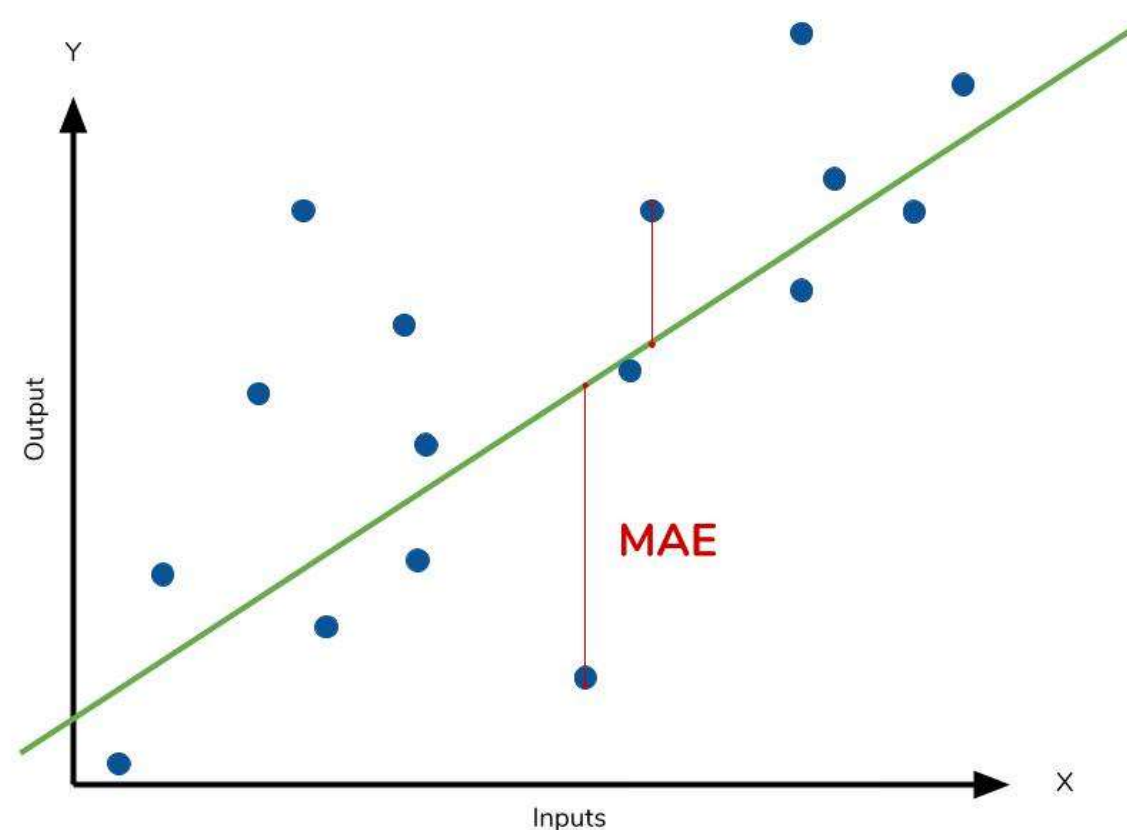
- 모델 간의 성능을 비교할 수 있는 객관적 수단
  - Task마다 주로 사용하는 평가지표가 상이함(Classification, regression 등)
- 학습 과정 모니터링
  - 모델이 학습하는 중 훈련 데이터셋과 검증 데이터에 대한 성능을 추적할 수 있음
  - 학습 중 발생하는 문제 또한 조기에 발견할 수 있음
- 모델 해석
  - 모델의 결과를 바탕으로 강점과 약점을 판단할 수 있음
  - 경우에 따라 모델의 신뢰도 또한 측정할 수 있음



## 02 머신러닝 학습

### ④ 평가지표의 다양성

- Regression 문제의 경우
  - 연속 변수 간의 차이를 계산
  - 예측값과 실제 값의 차이(Error)를 바탕으로 성능을 평가함
  - MAE(Mean Absolute Error 평균 절대 오차), MSE(Mean Squared Error 평균 제곱 오차)



## 02 머신러닝 학습

### ④ 평가지표의 다양성

- Classification 문제의 경우
  - 이산 변수 간의 차이를 계산
- Accuracy, Precision, Recall, F1-score 등이 사용됨
- 위의 평가지표들은 모두 Confusion Matrix를 바탕으로 계산할 수 있음

### ④ 평가지표의 다양성 – Confusion Matrix

- 분류 문제에 관한 정답과 모델의 예측을 바탕으로, 모델의 성능을 확인할 수 있는 표
  - 모델이 이진 분류 문제를 풀었다고 가정할 경우,
  - 데이터의 정답과 모델의 예측 간 관계를 4개의 구역으로 표현(TP, TN, FP, FN)

		Predicted Value		
		Yes	No	
Actual Value	Yes	16 True Positives	30 False Negatives	Recall
	No	10 False Positives	144 True Negatives	Specificity
		Prevalence	Precision	Negative Predictive Value
				Accuracy

## 02 머신러닝 학습

### ④ 평가지표의 다양성 – Confusion Matrix

- 이진 분류에서, 클래스는 Positive와 Negative로 나뉨
  - Positive: 이진 분류 중, 실험자가 관심을 갖는 클래스
  - Negative: 실험자가 관심을 갖지 않는 클래스
- 정답과 예측의 일치여부는 True와 False로 나뉨
  - True: 정답과 예측이 일치
  - False: 정답과 예측이 불일치

## 02 머신러닝 학습

### ④ 평가지표의 다양성 – Confusion Matrix

- 앞자리(일치 여부) + 뒷자리(예측 클래스)로 네 구역에 이름을 붙임
  - True Positive: Positive로 예측하여, 옳게 분류함
  - True Negative: Negative로 예측하여, 옳게 분류함
  - False Positive: Positive로 예측하여, 옳지 않게 분류함
  - False Negative: Negative로 예측하여, 옳지 않게 분류함

		Predicted Value		
		Yes	No	
Actual Value	Yes	16 True Positives	30 False Negatives	Recall
	No	10 False Positives	144 True Negatives	Specificity
		Prevalence	Precision	Negative Predictive Value
				Accuracy

## 02 머신러닝 학습

### ④ 평가지표의 다양성 – Accuracy

- 모든 예측 중, **정확한 예측**의 비율
- $TP + TN / TP + TN + FP + FN$
- 장점: **쉽고 직관적인** 지표
- 단점:
  - 클래스가 불균형할 경우, 모델의 실제 성능보다 훨씬 높거나 낮은 정확도를 보임
    - 클래스 비율이 9:1인 데이터를 하나의 클래스로 전부 예측하는 경우
  - FP, FN을 고려하지 못함

A confusion matrix diagram illustrating classification results. The vertical axis is labeled 'Actual Value' and the horizontal axis is labeled 'Predicted Value'. The matrix is divided into four quadrants: True Positives (16), False Negatives (30), False Positives (10), and True Negatives (144). The overall accuracy is noted as 160/200.

	Predicted Value	
	Yes	No
Yes	16 True Positives	30 False Negatives
No	10 False Positives	144 True Negatives

Accuracy: 160 / 200

## 02 머신러닝 학습

### ④ 평가지표의 다양성 – Precision(정밀도)

- 예측값이 양성인 것의 신뢰도
- $TP / TP + FP$
- Precision이 높을 수록, FP가 낮다는 뜻
- 단점
  - FN을 고려하지 못함
  - 정답이 양성이지만, 음성으로 예측한 경우를 감지하지 못함(질병 진단 등)

		Predicted Value	
		Yes	No
Actual Value	Yes	16 True Positives	30 False Negatives
	No	10 False Positives	144 True Negatives

Precision is calculated based on the True Positives and False Positives cells.



## 02 머신러닝 학습

### ④ 평가지표의 다양성 – Recall(Sensitivity, 민감도)

- 모델이 양성 값을 찾아낼 수 있는 능력
- $TP / TP + FN$
- 민감도가 높다는 것은 FN이 낮다는 뜻
- 단점
  - FP를 고려하지 못함
  - 정답이 음성이지만, 양성으로 예측한 경우를 감지하지 못함(스팸 메일 분류 등)

		Predicted Value		
		Yes	No	
Actual Value	Yes	16 True Positives	30 False Negatives	Recall
	No	10 False Positives	144 True Negatives	

## 02 머신러닝 학습

### ④ 평가지표의 다양성 – Precision/Recall의 Trade-off

- Precision과 Recall은 Trade-off 관계
  - $\text{Precision} = \text{TP} / \text{TP} + \text{FP}$
  - $\text{Recall} = \text{TP} / \text{TP} + \text{FN}$
- 두 지표는 Accuracy의 단점을 보완하기에 좋으므로, 두 지표를 모두 사용하는 방법이 고안됨
  - F1-Score
  - ROC Curve와 AUC
  - PR Curve와 AP

## 02 머신러닝 학습

### ④ 평가지표의 다양성 – F1-Score

- Precision과 Recall의 조화평균

- $$\text{F1 Score} = 2 \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- FP, FN에 모두 반응하는 평가지표

- 불균형한 데이터셋에도 효과가 좋음

- 그러나 두 지표를 동등하게 고려하므로, 어느 하나가 월등하게 좋을 경우, 나머지의 수치가 높지 않더라도 좋은 결과를 반환함

03

# 딥러닝 기초

## 03 딥러닝 기초

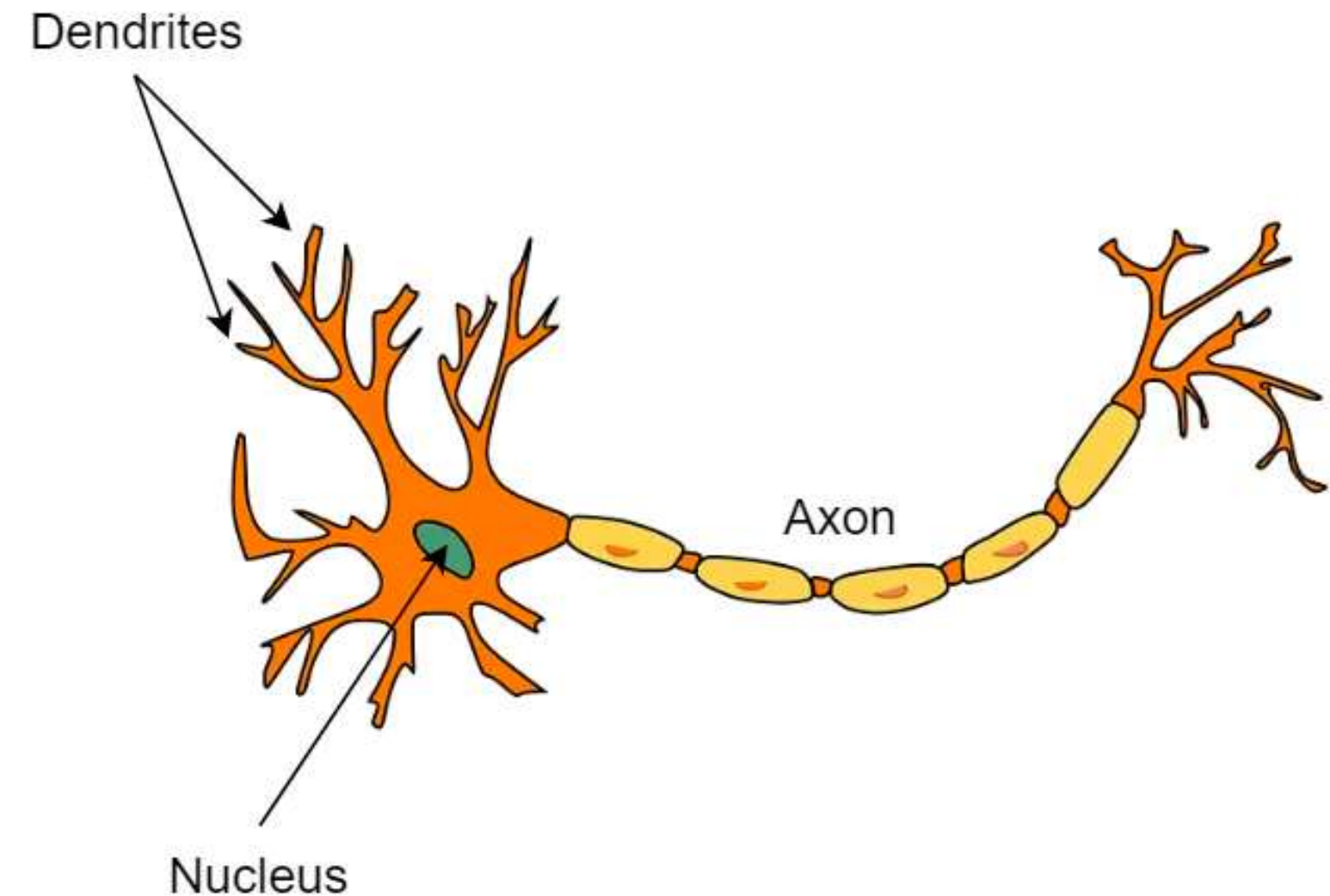
### ④ 인공신경망(Artificial Neural Network)

- 뇌의 생물학적 신경망에서 영감을 받아 만들어진 컴퓨터 시스템
- 인간다운 인공지능을 구현하기 위해 인간의 신경 시스템을 모방
  - 인간 신경계의 기본 단위는 신경세포(뉴런, Neuron)
  - 인공신경망의 기본 단위는 뉴런을 모방한 퍼셉트론(Perceptron)

## 03 딥러닝 기초

### ④ 신경 세포(Neuron)

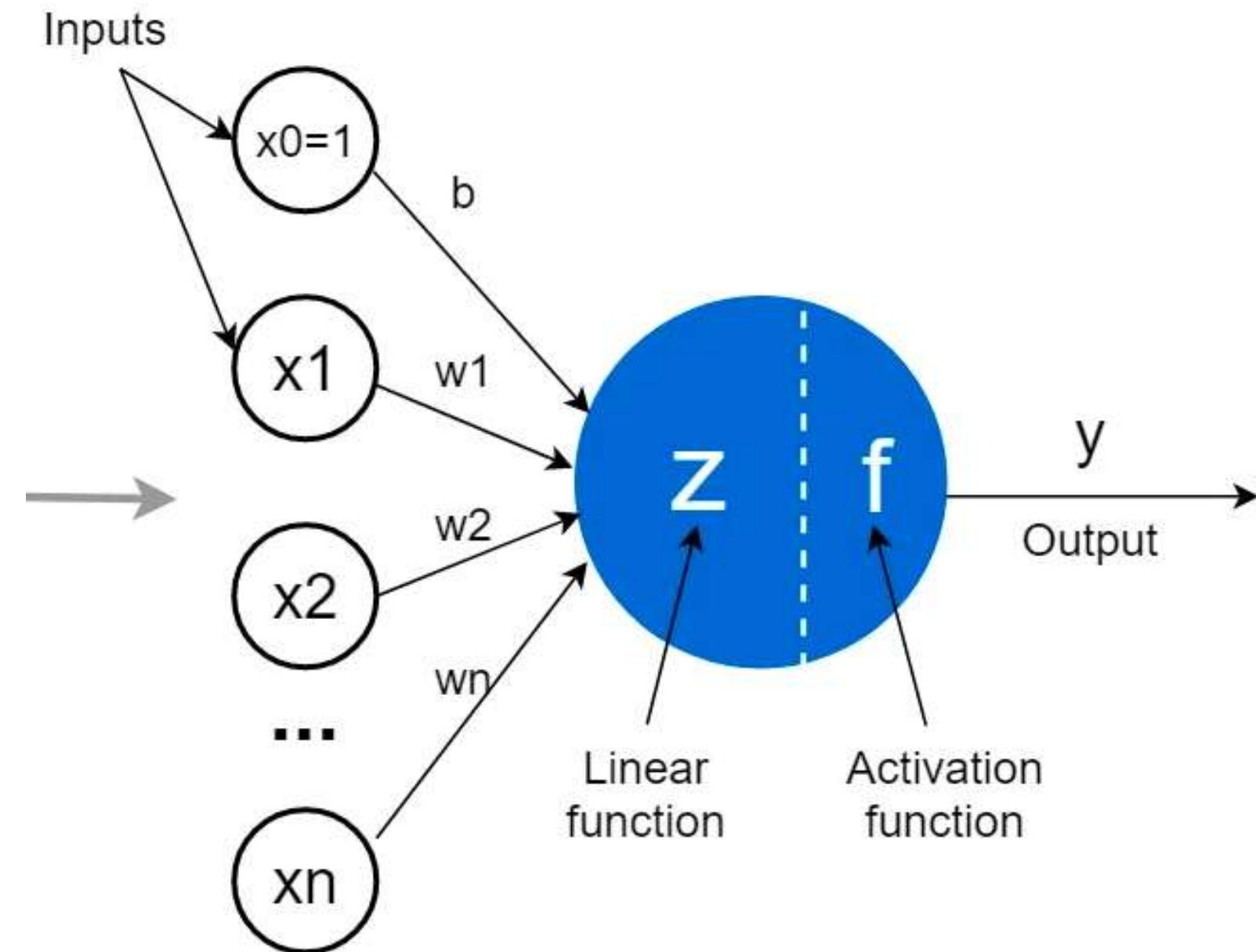
- 신경계 기본 구성 단위
- 감각기관에서 받은 정보를 전달
- 다른 여러 뉴런으로부터 자극을 받아
- 다른 뉴런으로 신호를 전달
- 경우에 따라 신호가 전달되지 않음



## 03 딥러닝 기초

### ④ 퍼셉트론(Perceptron)

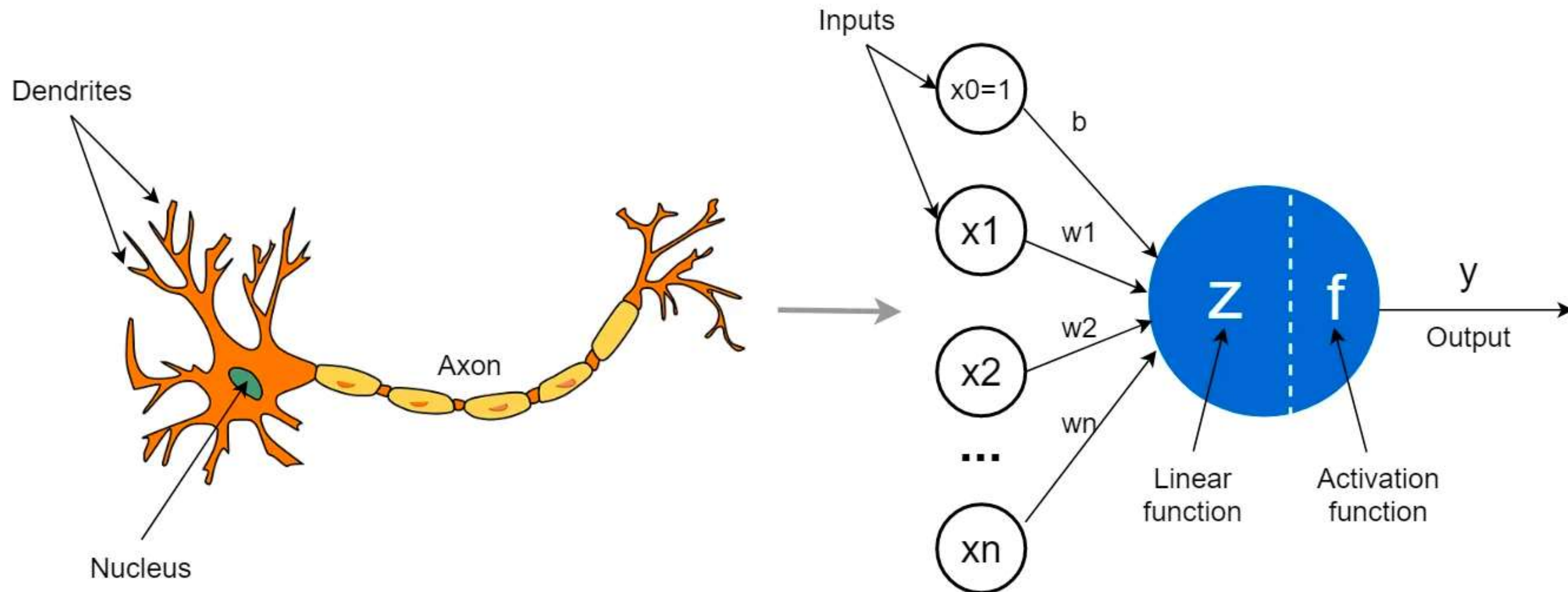
- 인공신경망 기본 구성 단위
- Input에서 받은 Feature을 전달
- 다른 퍼셉트론에서 신호를 받아
- 다음 퍼셉트론으로 신호를 전달
- 경우에 따라 신호의 양이 조절됨





### 03 딥러닝 기초

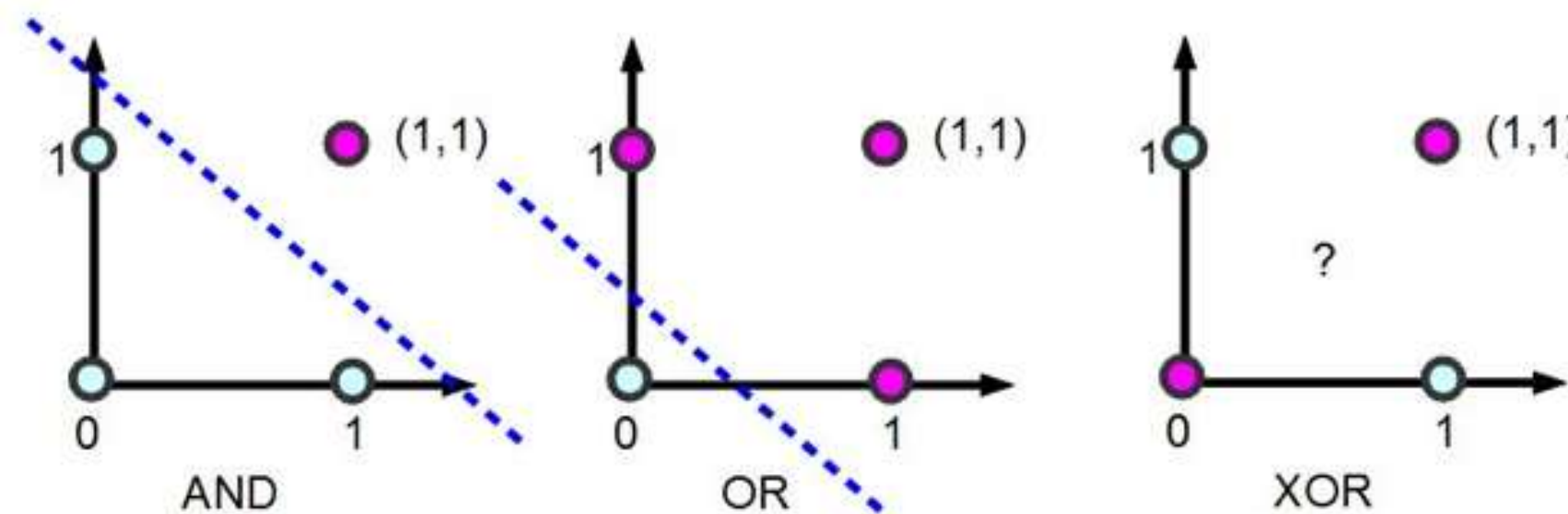
#### ④ 신경세포와 퍼셉트론



## 03 딥러닝 기초

### ④ 단일 퍼셉트론(SLP, Single Layer Perceptron)

- 선형분류 문제를 해결함
- 인간 신경계를 성공적으로 모방
- 하지만 선형분류가 불가능한 모든 문제 해결 불가
  - 직선이 아니거나
  - 선의 형태가 아닌 문제(e.g. XOR gate)



## 03 딥러닝 기초

### ④ 선형 분류가 불가능한 문제의 해결법

- 입력을 **비선형 변환**하여 비선형 분류 문제 해결
  - 활성화 함수
- 퍼셉트론을 **여러 층으로 쌓아** 비선형 분류 문제 해결
  - 복잡한 형태의 함수를 생성

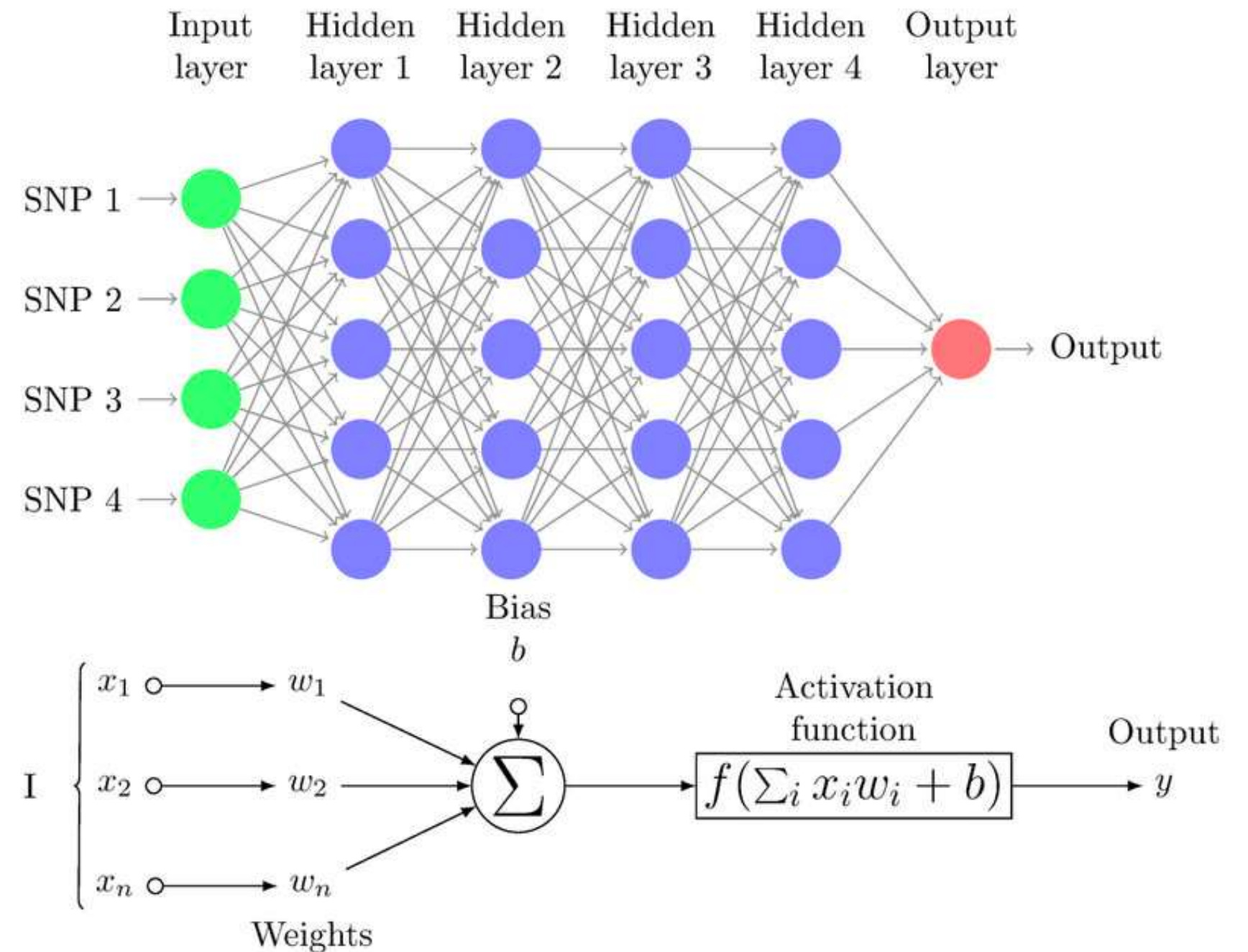
## 03 딥러닝 기초

### ④ MLP(Multi Layer Perceptron)

- 여러 층의 퍼셉트론으로 구성

- 입력층(Input layer)
- 은닉층(Hidden layer)
- 출력층(Output layer)

- 각 노드(퍼셉트론)에 활성화 함수 포함
  - 비선형성 부여

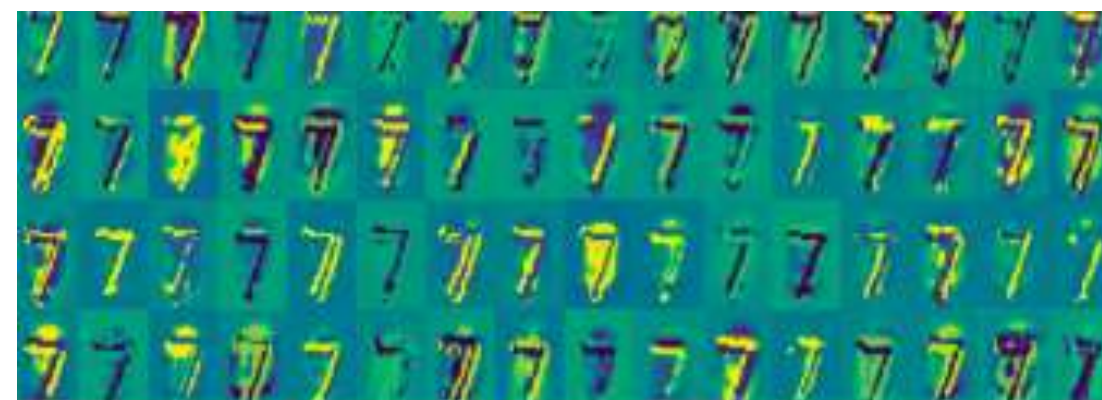




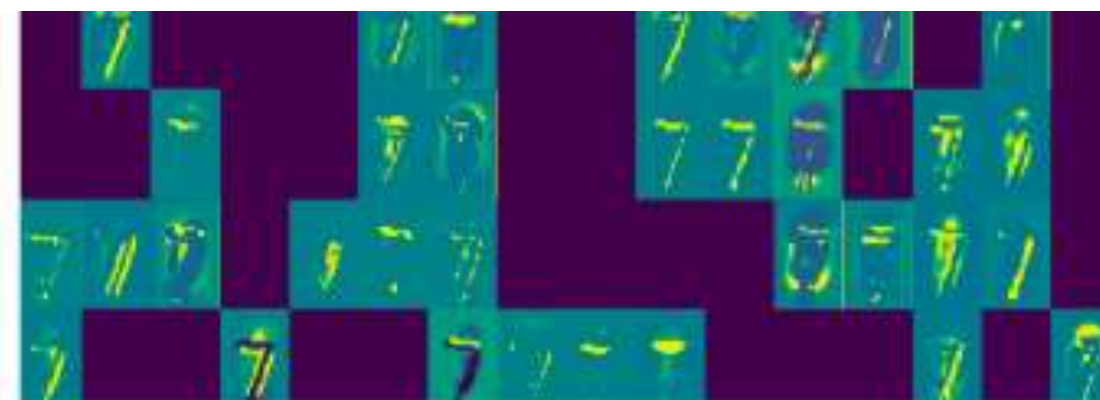
## 03 딥러닝 기초

### ④ 선형성과 활성화 함수

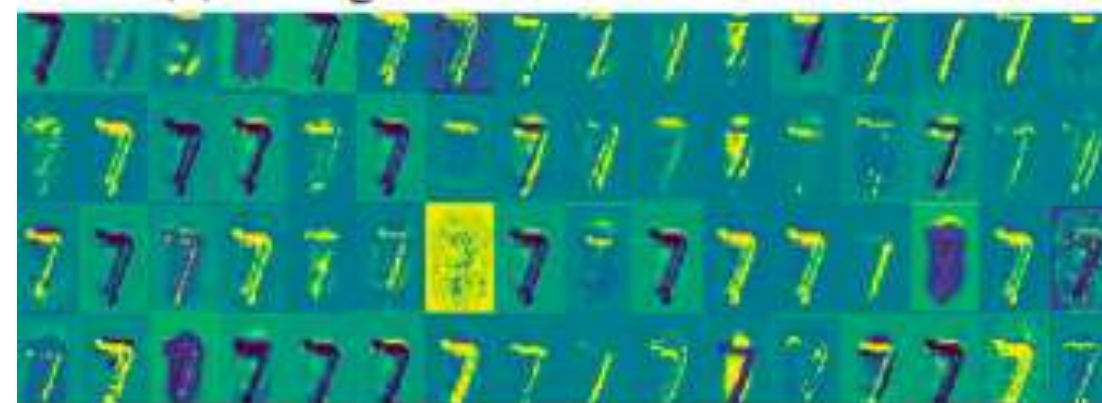
- 비선형성이 포함되어야 함수의 표현력이 강화
- 함수의 표현력이 강화되면, 데이터의 복잡한 특징도 파악하기 용이
- 무엇이 가장 좋은가? → 환경에 따라 다름



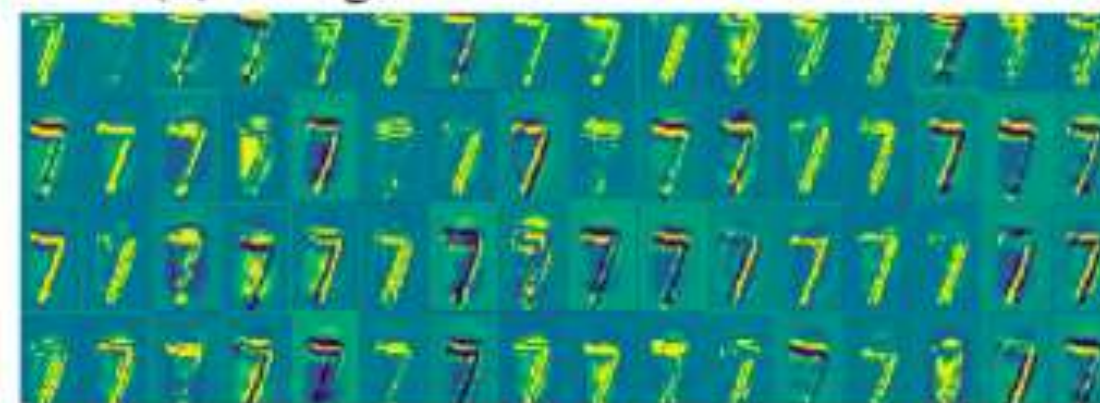
(a) Using *Tanh* activation function



(b) Using *ReLU* activation function



(c) Using *Swish* activation function



(d) Using *LiSHT* activation function

## 03 딥러닝 기초

### ④ 오버피팅(Overfitting)

- 모델이 학습 데이터에 과도하게 적합하여 학습 데이터에서는 높은 성능을 보이지만, 새로운 데이터나 테스트 데이터에서는 성능이 저하되는 현상
  - 일반화 능력이 낮아지는 현상을 의미
- 모델이 학습 데이터에서 필요한 특징 뿐만 아니라, 노이즈나 이상치까지 지엽적으로 과도하게 학습하는 경우를 의미
- 새로운 데이터를 바라볼 때, 학습 데이터에서 나타난 노이즈나 이상치를 기준으로 판단하게 됨

## 03 딥러닝 기초

### ④ 오버피팅의 발생 원인

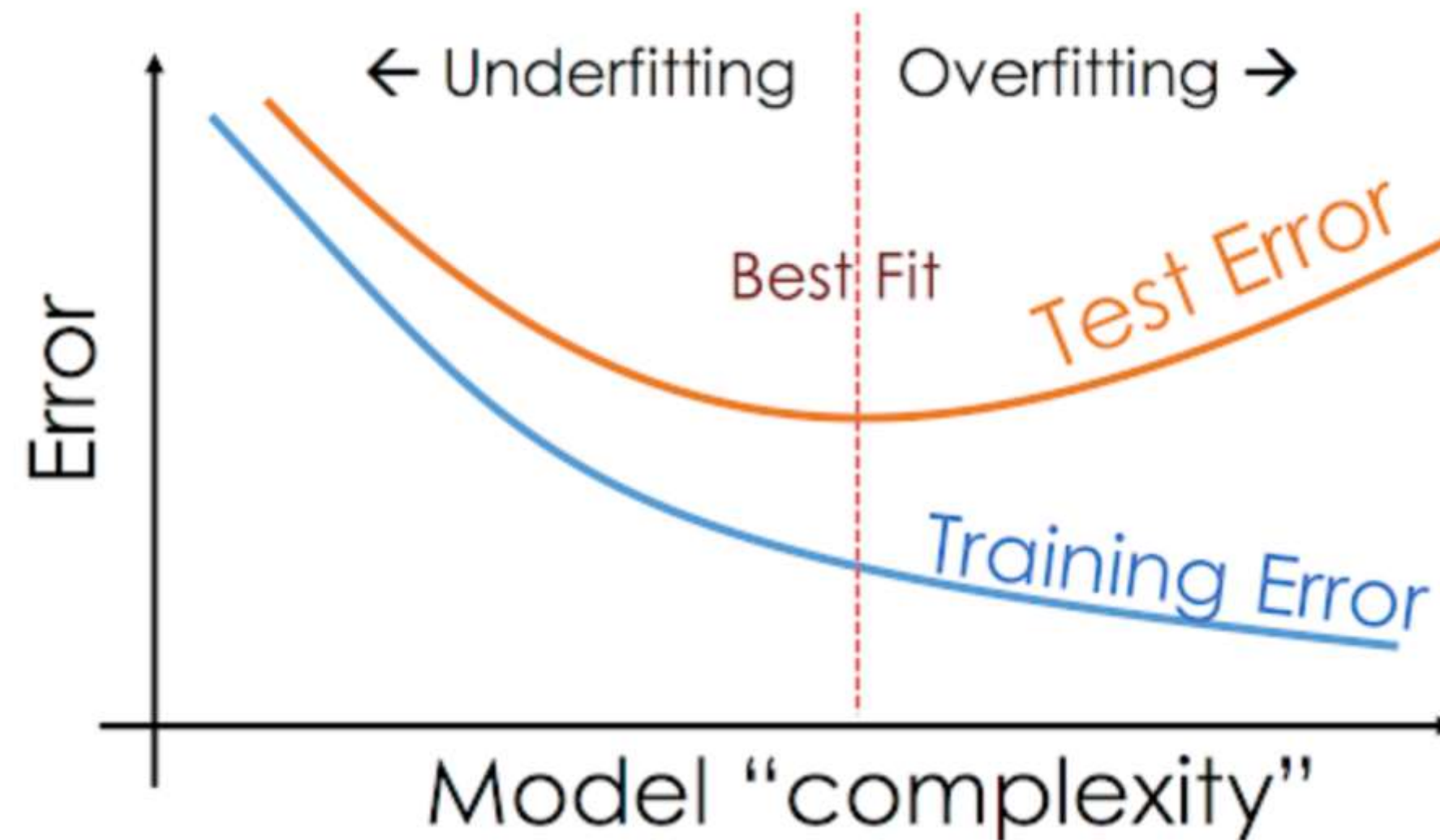
- 데이터의 질적·양적 규모에 비해 지나치게 큰 모델을 사용할 경우, 모델의 데이터의 작은 변동을 감지할 수 있음
- 반대로 데이터가 지나치게 부족한 경우, 모델이 세상에 존재하는 데이터의 다양한 측면을 학습하지 못하고, 국부적인 요소만을 기준으로 판단하게 됨
- 같은 데이터에 대해 지나치게 오래 학습을 시킬 경우, 모델이 학습 데이터에 지나치게 익숙해지게 됨



## 03 딥러닝 기초

### ④ 오버피팅 현상 해석

- 학습 데이터와 검증(Validation) 데이터를 바탕으로 모델을 학습시킬 때,
  - Train loss와 Validation loss가 멀어지는 순간, 혹은
  - Validation loss가 Train loss에 비해 지나치게 상승하는 순간 오버피팅 발생



## 03 딥러닝 기초

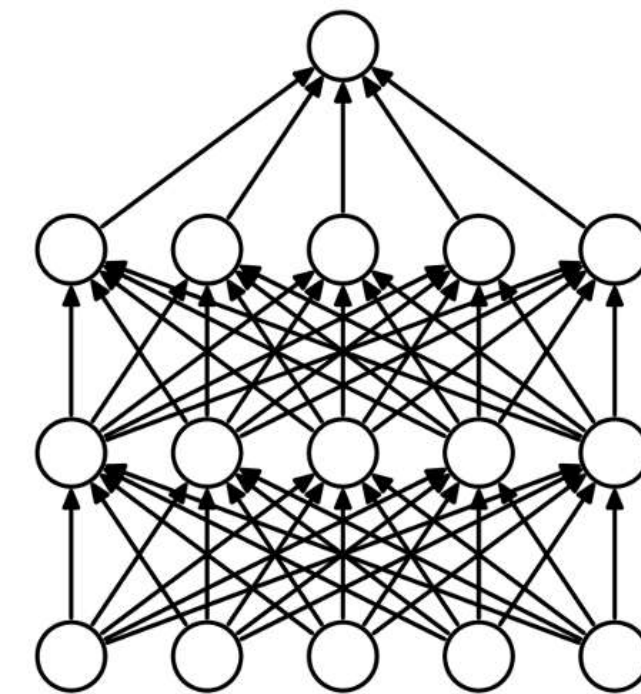
### ④ 규제(Regularization)

- 오버피팅을 막기 위한 수단을 일반적으로 규제(Regularization)라 함
- 머신러닝과 딥러닝에서 사용되는 규제의 종류는 다소 차이가 있음
- 딥러닝에서 자주 사용하는 규제는 다음과 같음
  - Dropout
  - Batch normalization
  - Data augmentation

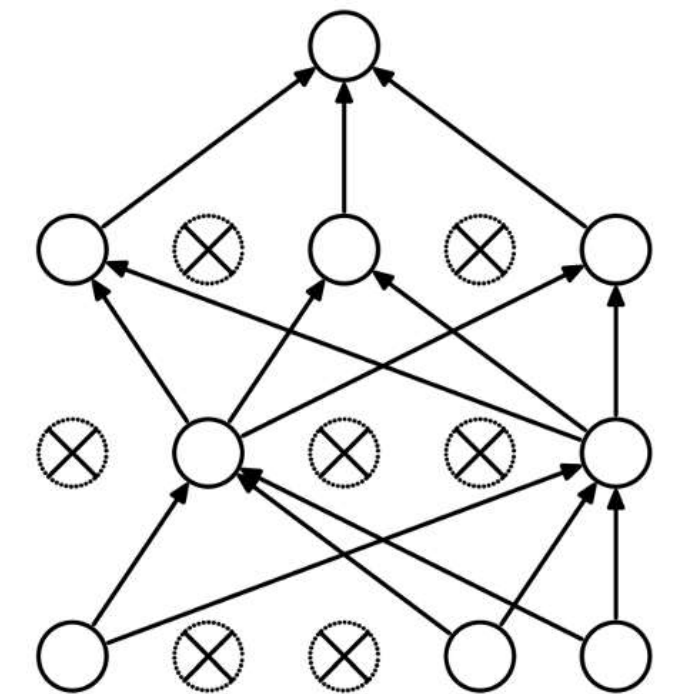
## 03 딥러닝 기초

### ④ Dropout

- 학습 과정 중, 뉴런의 연결을 비활성화시키는 방법
  - 모델 학습 중 특정 뉴런의 의존도를 낮추는 방법
  - 테스트 단계에서는 모든 뉴런을 사용하여 예측함
- 매 Step마다 사전에 정의된 비율( $p$ )만큼 임의의 뉴런을 비활성화 시킨 후 학습을 진행
- $p$ 값은 확률이므로 0~1 사이에서 표현되며, 보통 0.2~0.5 사이의 값을 사용함



(a) Standard Neural Net

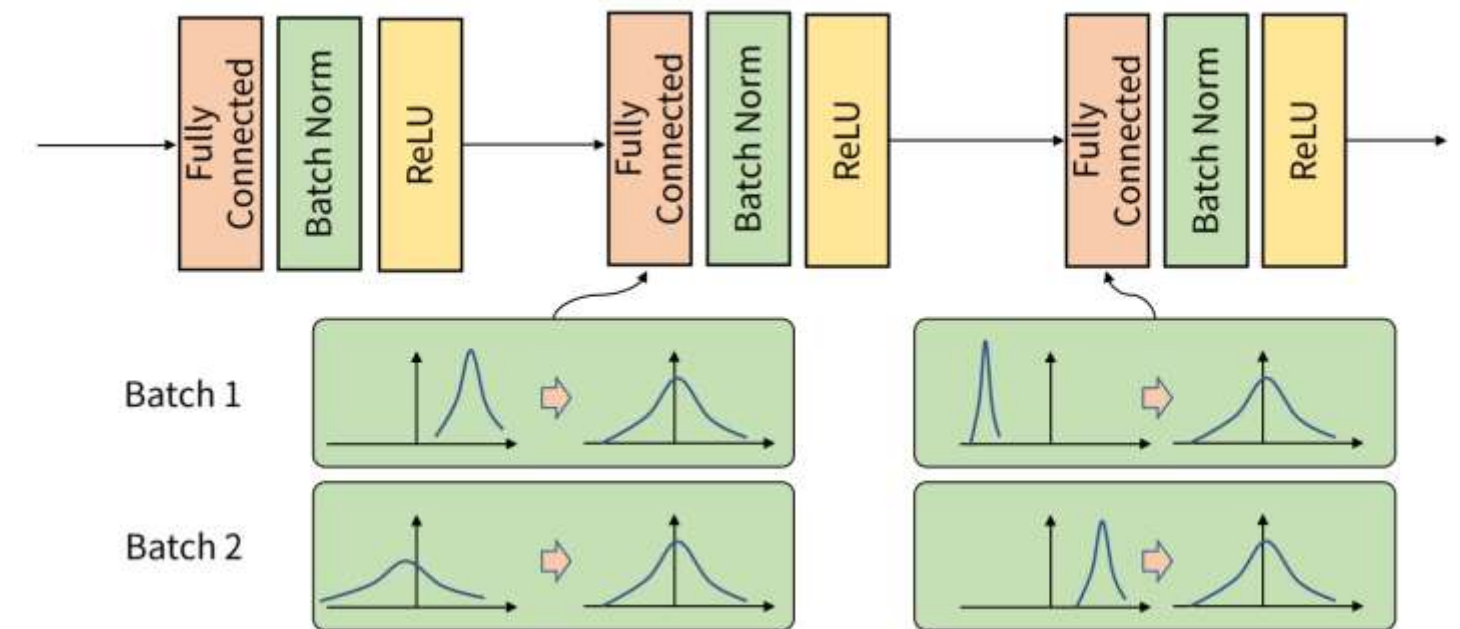


(b) After applying dropout.

## 03 딥러닝 기초

### ④ Batch Normalization

- 일반적으로 모델은 데이터를 Batch 단위로 입력받음
  - 그러나 모든 Batch에서 데이터가 균형있게 들어가지 못함
  - 이로 인해 특정 레이블이나 특징에 대해 불균형이 초래
- 이러한 Batch 단위 입력의 불균형이 모델의 학습 속도를 저하
  - 가중치 업데이트 시 안정적인 수렴이 불가능
- Batch 단위로 값을 정규화(Normalization)하면 입력마다 차이를 줄일 수 있음

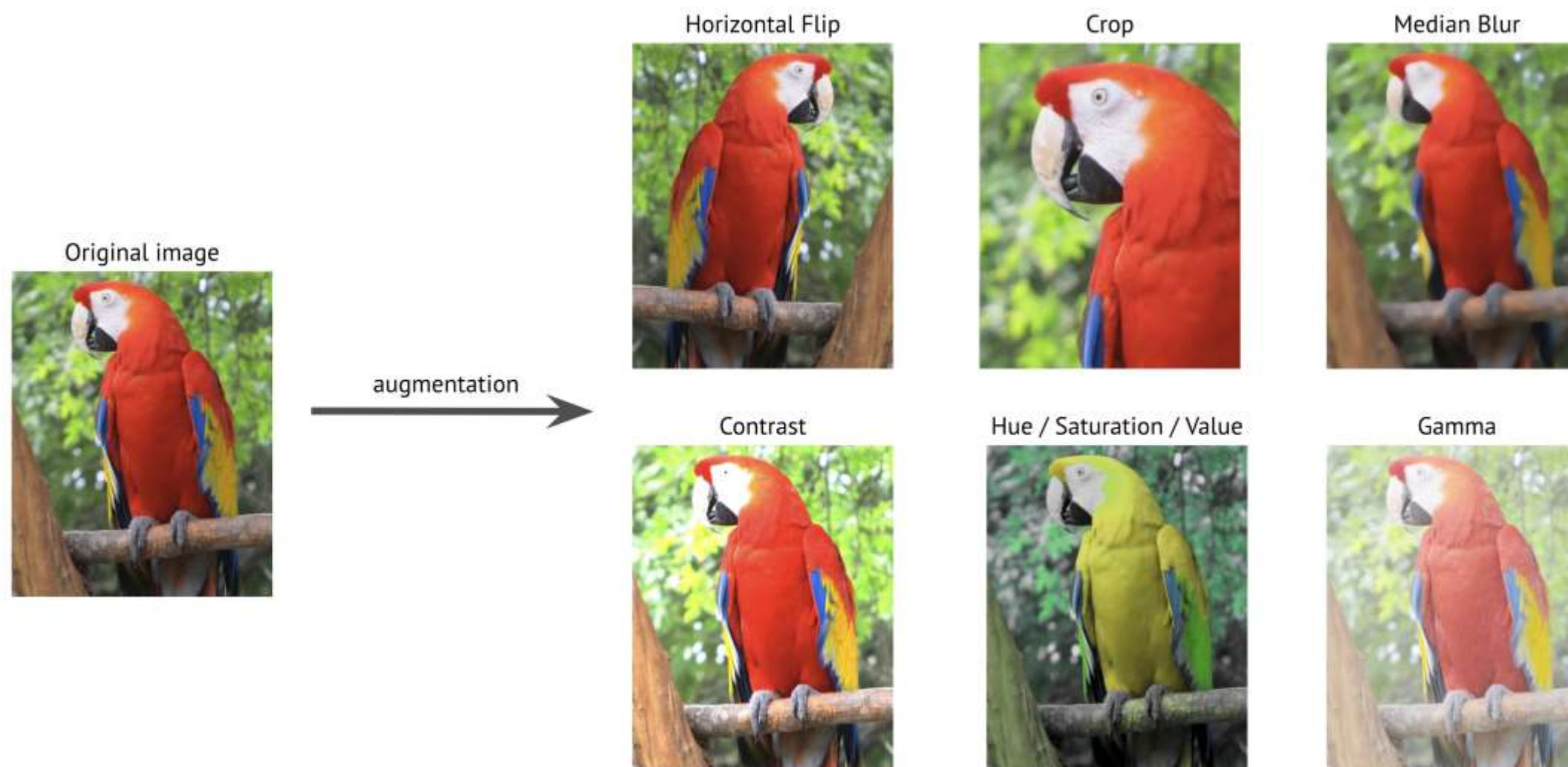




## 03 딥러닝 기초

### ☑ Data Augmentation

- 기존의 학습 데이터를 변형하여 새로운 학습 데이터를 생성하는 방법
  - 데이터의 양이 늘어났으므로 오버피팅의 가능성이 줄고, 일반화 능력이 상승함
  - 또한, 인위적인 노이즈를 적당히 줌으로써, 모델을 강건(Robust)하게 학습시킬 수 있음



## 03 딥러닝 기초

### ④ Data Augmentation

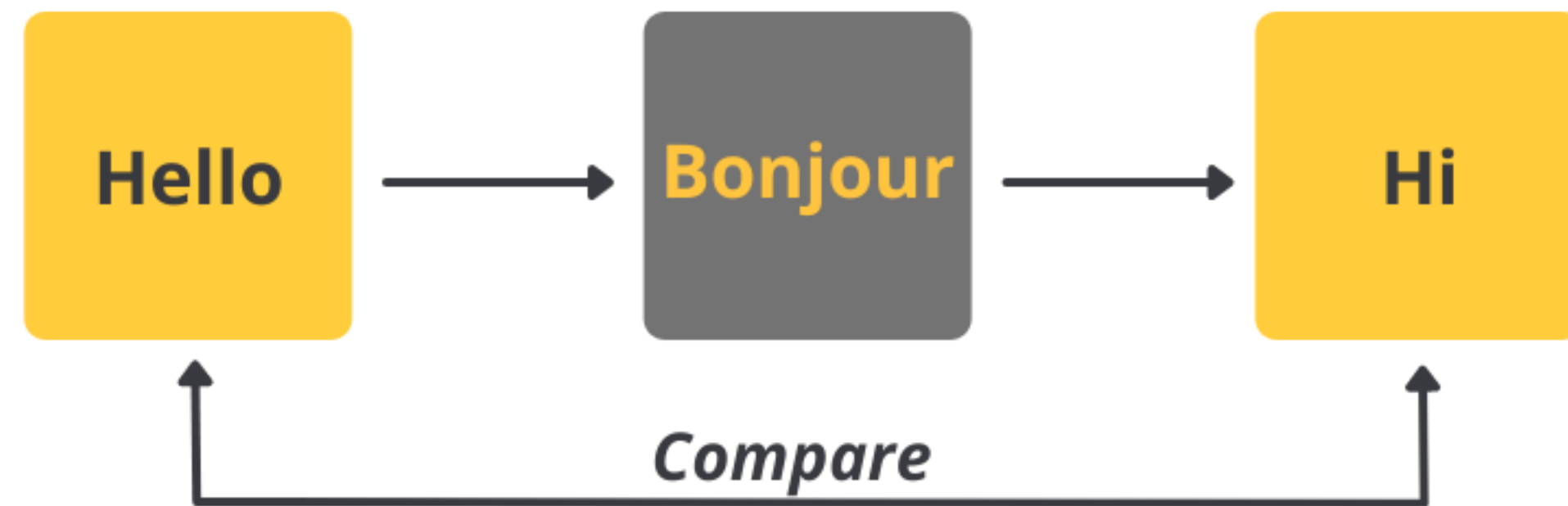
- NLP 분야에서 자주 사용하는 Augmentation의 종류
  - Lexical substitution(어휘 대체): 데이터 내 문장 속 단어를 유의어/반의어 등으로 대체



## 03 딥러닝 기초

### ☑ Data Augmentation

- NLP 분야에서 자주 사용하는 Augmentation의 종류
  - Back translation: 기계 번역 분야에서, 원문장을 다른 언어로 번역한 후, 번역된 문장을 원문장으로 다시 번역하는 기법
    - Ex) A 모델로 영어를 한글로 번역한 뒤, B모델로 번역된 한글 문장을 영어로 다시 번역
    - 이 경우 데이터셋이 두 배가 됨





## 03 딥러닝 기초

### ④ Data Augmentation

- NLP 분야에서 자주 사용하는 Augmentation의 종류
  - Random noise injection: 오타, 특정 단어 공백처리 등 노이즈를 추가하는 방법

This is very cool → This is **ber**very cool



This is very cool → This \_ very cool