세미나(4)

딥러닝 비전

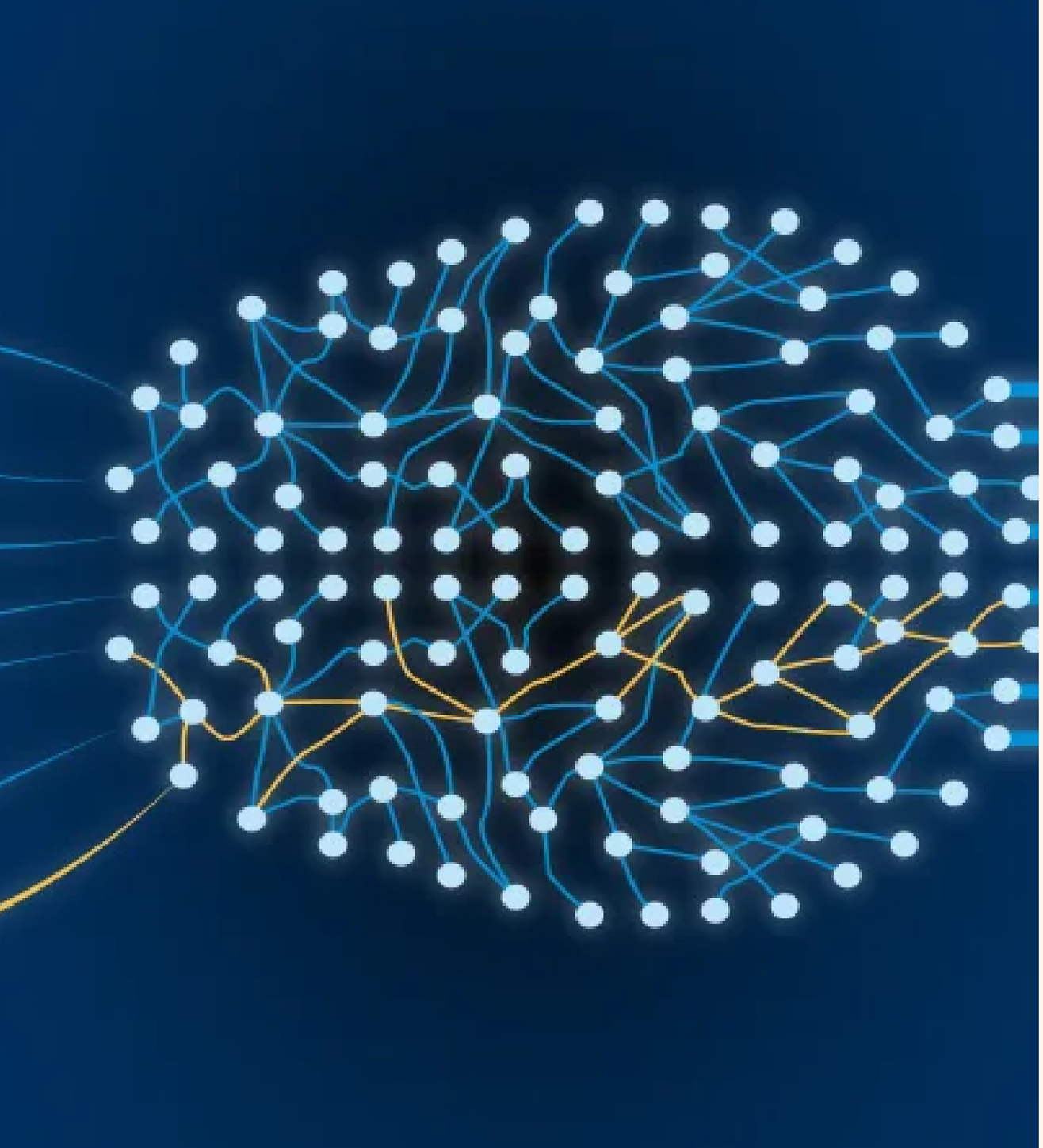
정보통계학과 이현섭

전달 내용

1 딥러닝과 기계학습

2 인공 신경망과 퍼셉트론

3 학습 알고리즘



딥러닝과 기계학습

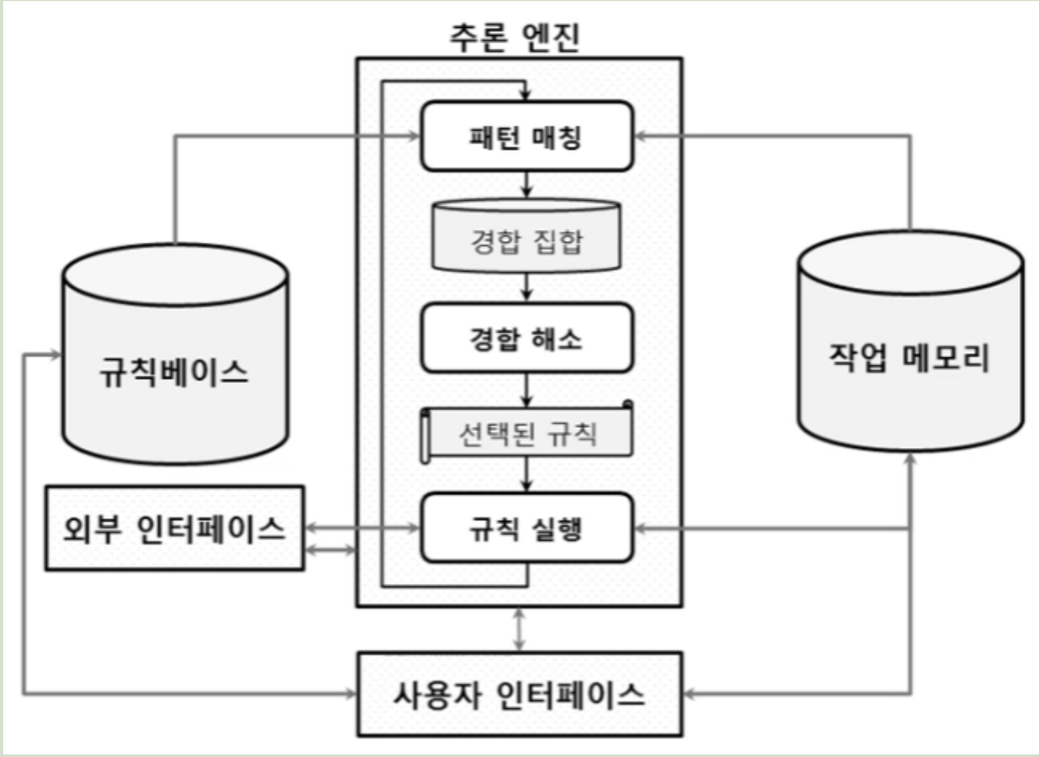
방법론의 대전환

- 지금까지 공부한 내용들의 대부분 : 고전적 컴퓨터 비전
(지역 검출, 영역 검출 등)
- 고전적 컴퓨터 비전에서 사용한 방법 : 규칙 기반(rule-based)

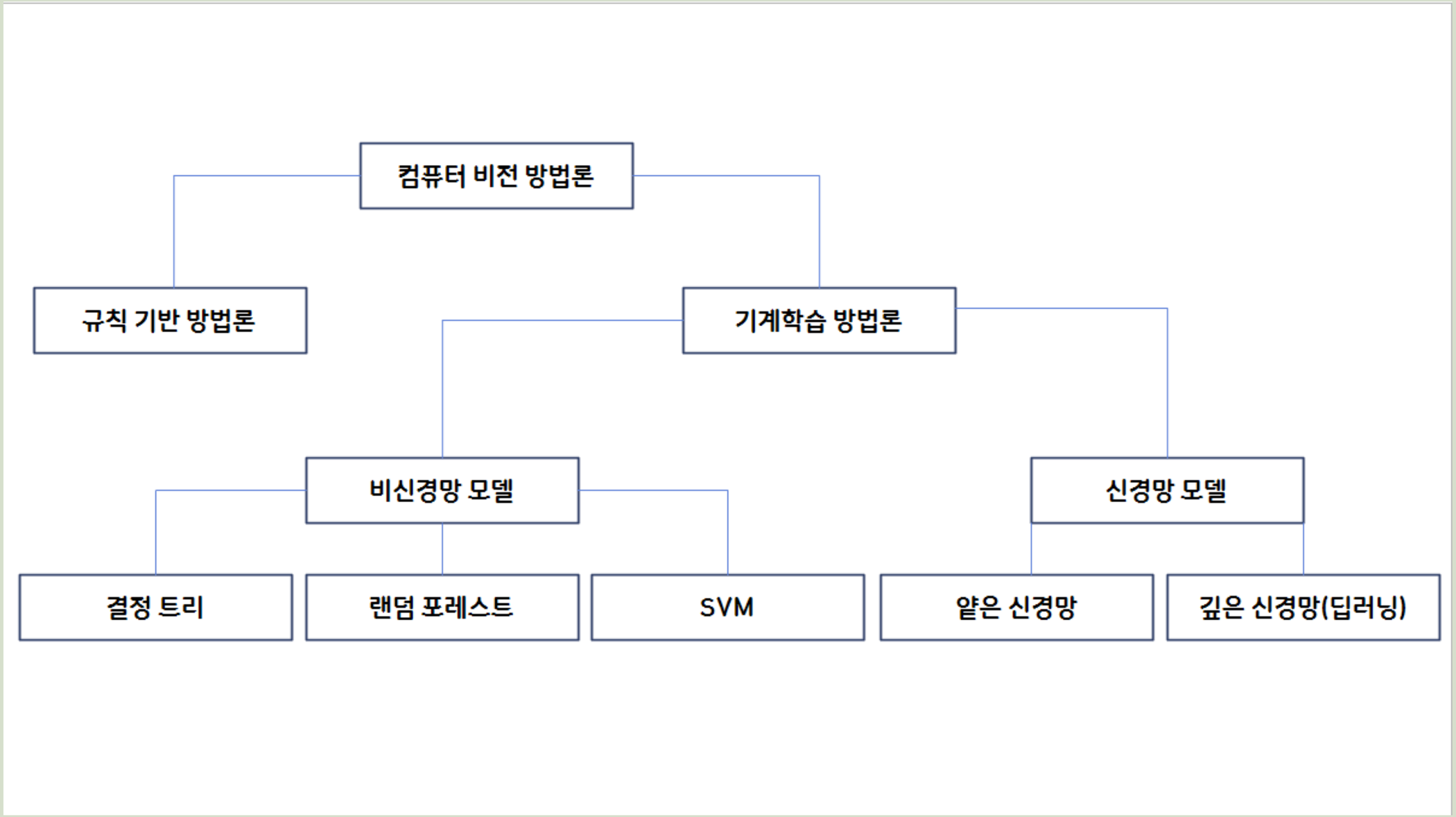
- 하지만 규칙 기반에는 한계가 존재!
-> 규칙 기반은 '수작업' 특징을 가지고 있음.
-> 이는 대부분 규칙으로 표현되는데, 데이터에 따라 최적의 필터를 설계해야 한다.

- 그러나 규칙 기반을 사용하는 고전적 컴퓨터 비전은 이런 질문에 명확한 답을 내놓지 못함!

- 따라서, 그 이상을 돌파하기 위해 새로운 방법론이 필요!



딥러닝의 대전환


컴퓨터 비전 방법론
규칙 기반 방법론
기계학습 방법론
비신경망 모델
신경망 모델
결정 트리
랜덤 포레스트
SVM
얕은 신경망
깊은 신경망(딥러닝)

딥러닝의 대전환

- 신경망은 얕은 모델로 출발하여 점점 깊어지며 발전

-> 퍼셉트론, 다층 퍼셉트론, 깊은 다층 퍼셉트론

- 딥러닝의 영향력

-> 자연어 처리(NLP), ex. 챗봇, Chat GPT

-> 지능 게임 분야 ex. 알파고

- 이러한 딥러닝 기술이 전 세계에 주목을 받고있으며, 큰 영향을 미치고 있다!

기계학습 기초

- 기계학습의 단순한 예

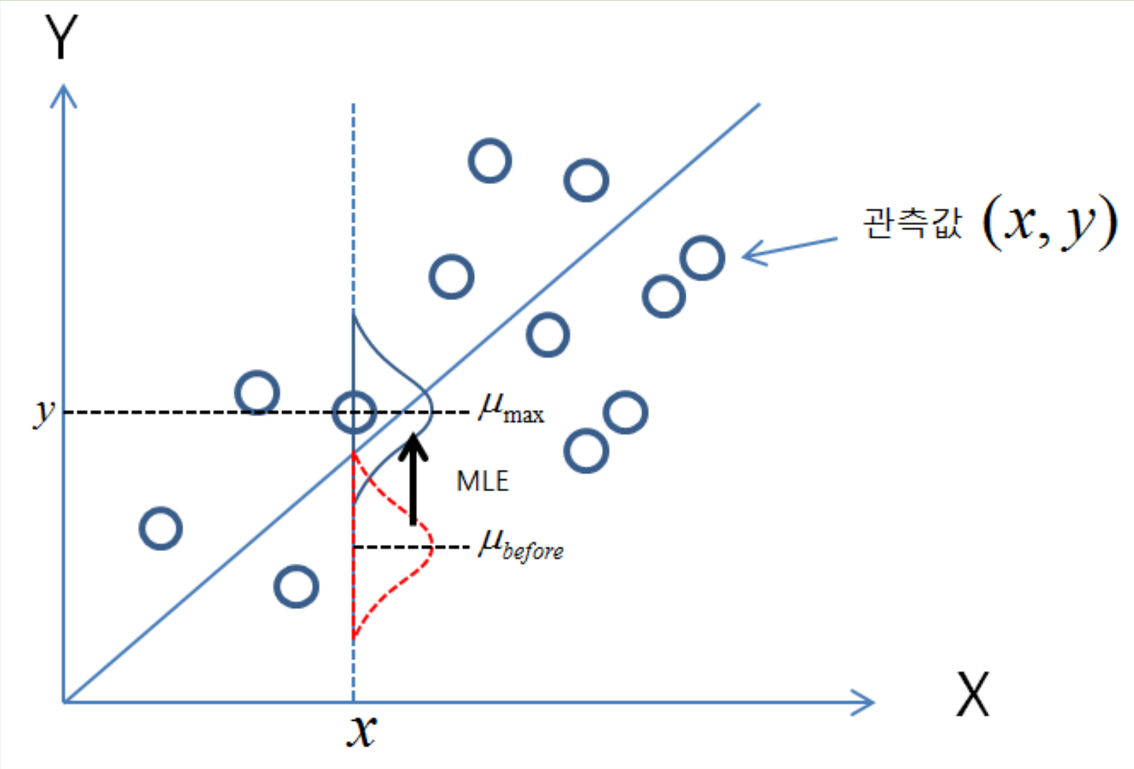
-Ex. 보일러로 설명

1. 기름을 더 분사하면 보일러 온도는 상승한다.
2. 보일러 공학자는 기름 분사량을 X, 온도를 Y로 두었다.
3. 이 두 변수 간의 관계를 표현하는 함수에 관심을 둠.

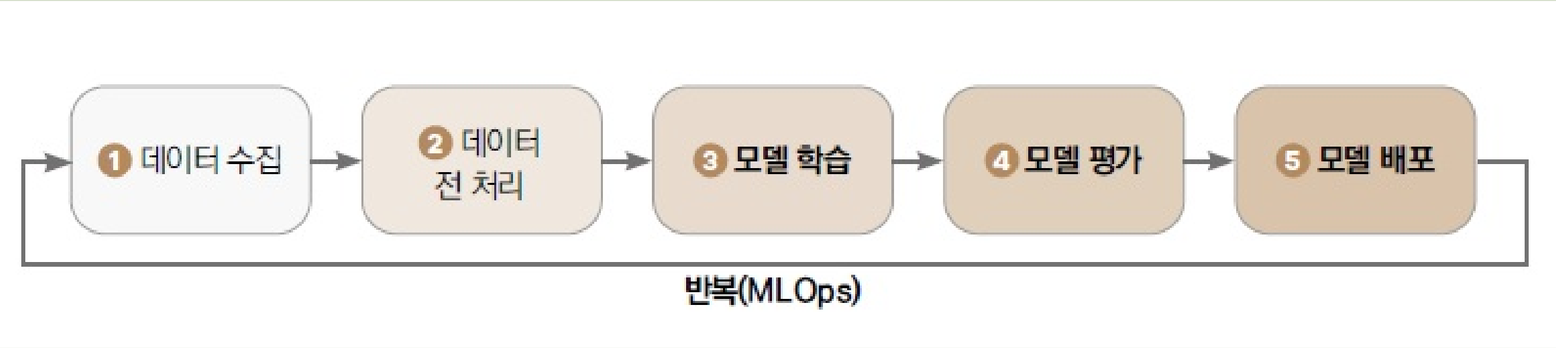
-> 여기서 기계학습은 함수를 모델이라고 부르고, 수집한 데이터로 방정식을 풀어 함수를 알아내는 일을 '학습'이라고 한다.

기계학습 모델 : Y = f(X)

학습된 모델로 특정 분사량에 대해 온도를 예측하는 일 '예측'



기계학습의 4단계



1. 데이터 수집 : 회귀, 분류의 문제를 해결하기 위해서 데이터셋의 구성을 확인하는 것이 필요.
데이터 수집에는 많은 비용이 들며, 레이블링이 불가능한 경우도 있기에 적절한 데이터셋을 찾는 것도 중요.

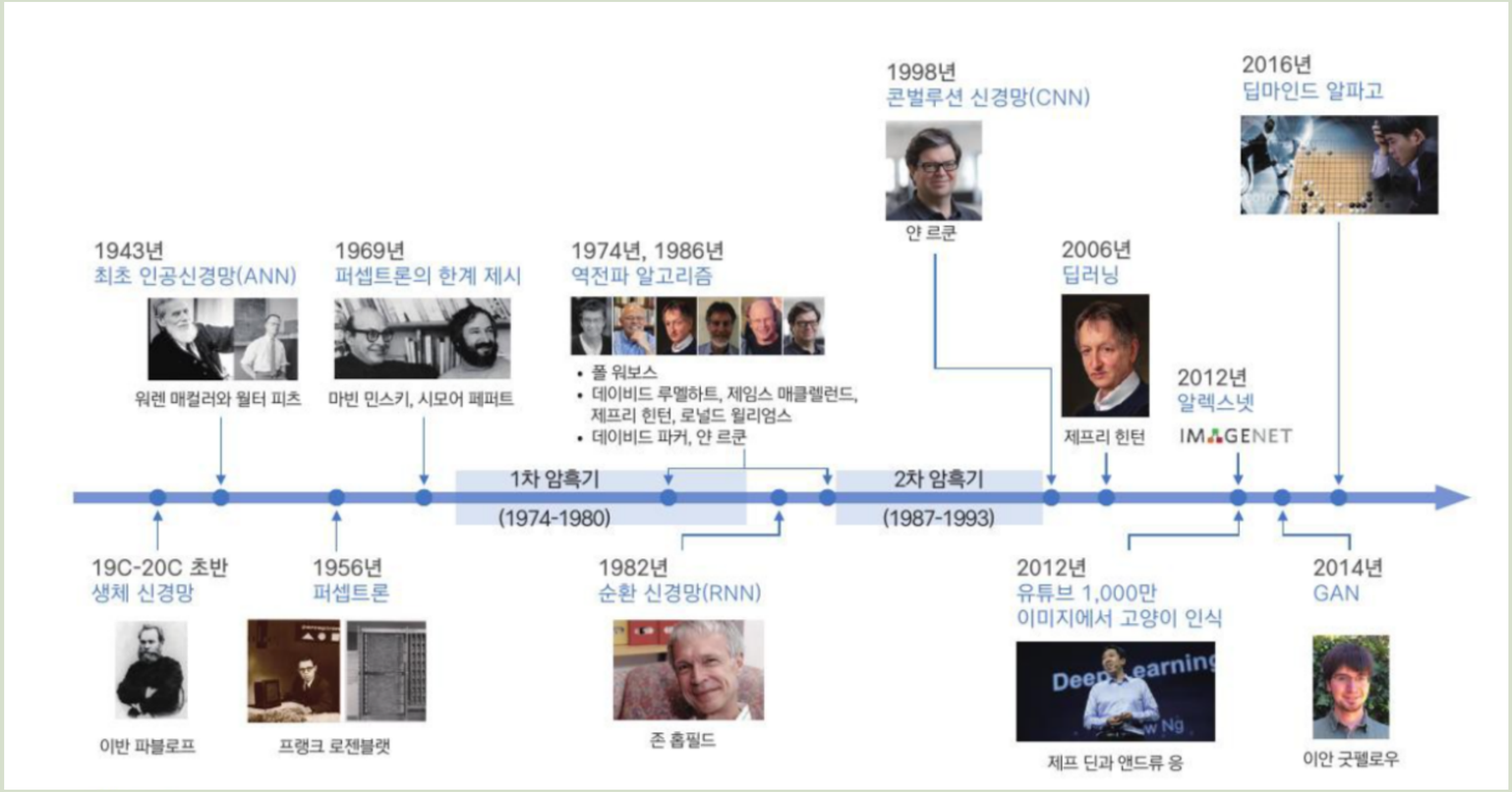
2. 모델 선택 : 직선형 모델, 비선형 모델 등 적절한 모델을 선택.
Ex. X와 Y의 관계가 더 복잡한 경우라면 비선형 모델을 사용. (가중치, 층 등)

3. 학습 : 훈련 집합에 있는 샘플을 최소 오류로 맞히는 최적의 가중치 값을 알아내는 작업.

4. 예측 : 학습을 마친 모델, 가중치의 최적값을 가진 모델에 학습에 사용하지 않던 새로운 특징 벡터를 입력하고 출력을 구하는 과정.
Ex. K-겹 교차 검증.

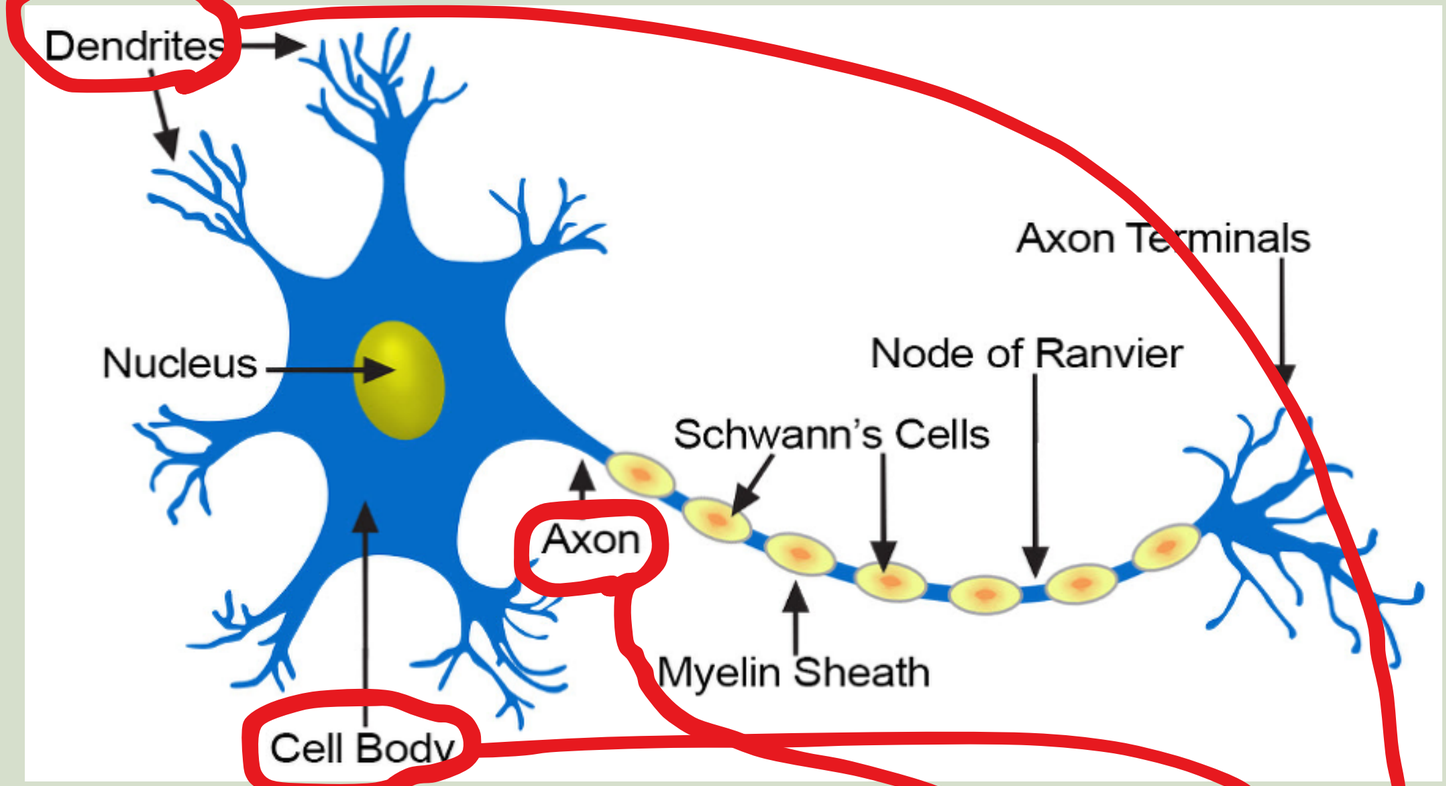
인공 신경망과 퍼셉트론

신경망 역사



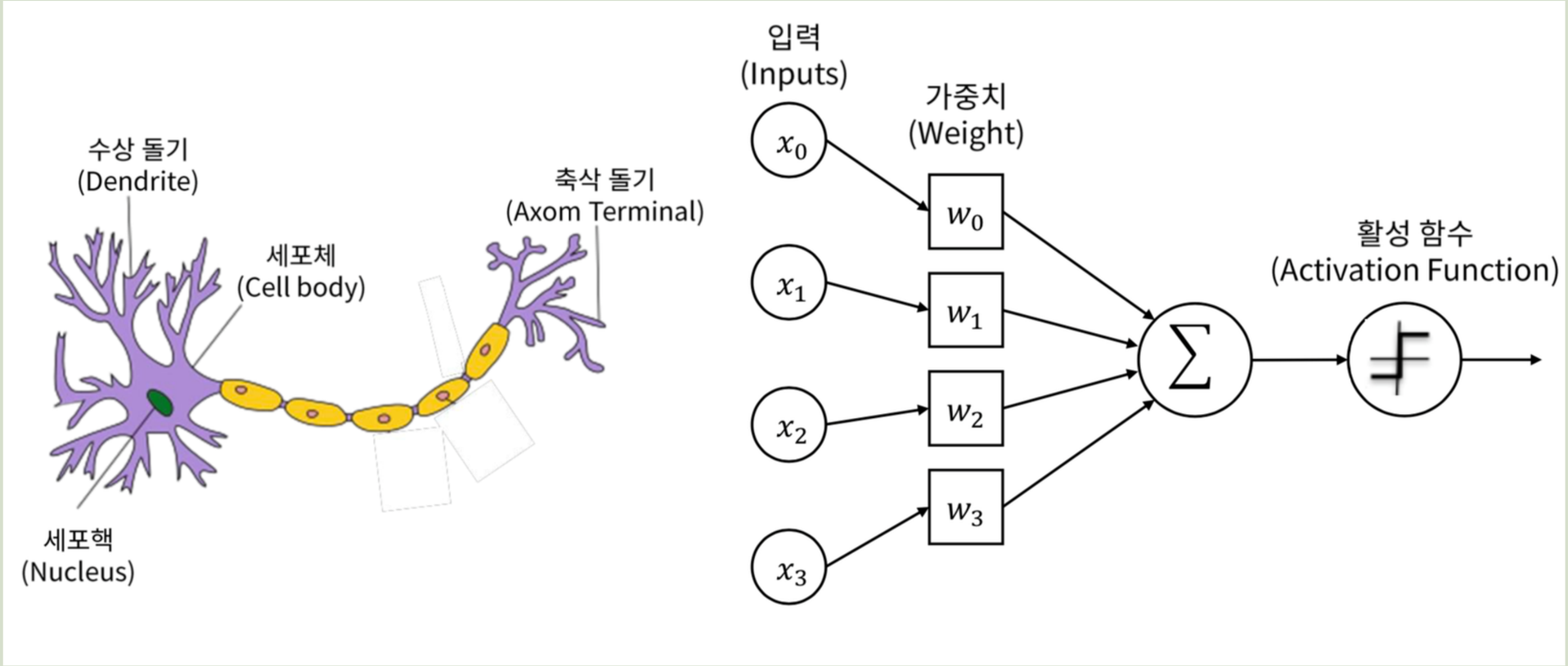
- 1900년대에 들어 인간 뇌에 대한 활발한 연구에 힘입어 뉴런의 정체가 밝혀짐.
- 1946년에는 세계 최초의 전자식 컴퓨터인 에니악이 탄생함.
- 이후 신경 과학자들은 컴퓨터로 인공 신경망을 구현하여 사람처럼 인식할 수 있는 기계를 만들려는 발상을 진행.

신경망



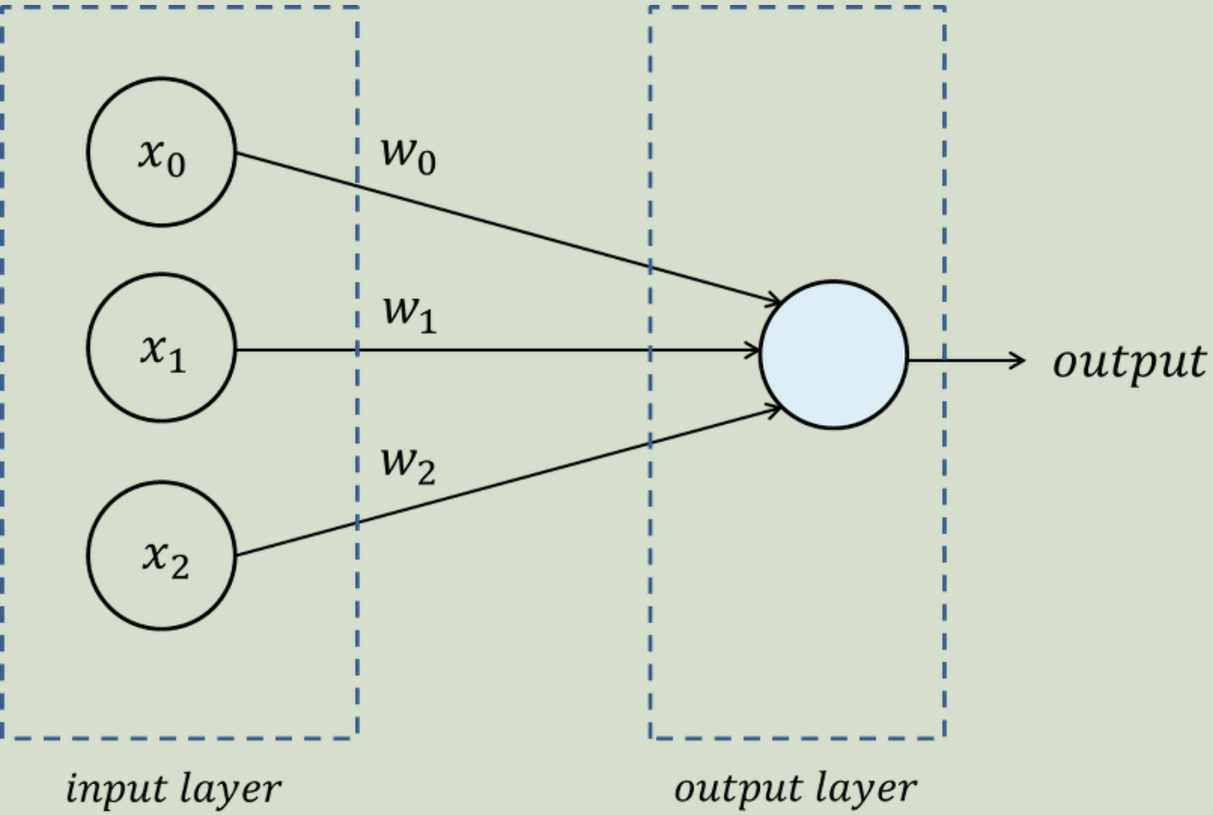
생물학적 신경망	인공 신경망
세포체	노드(node)
수상돌기(가지돌기)	입력(input)
축삭돌기(축색돌기)	출력(output)
시냅스	가중치(weight)

퍼셉트론



- 퍼셉트론은 로젠 블렛이 1957년 고안한 알고리즘, 입력 데이터를 2개의 부류 중 하나로 분류하는 분류기.
- 노드, 가중치, 층과 같은 개념이 도입되어 딥러닝을 포함하여 현대 신경망의 중요한 구성요소들을 이해하는데 의미가 있음.
- 2개의 부류 중 하나로 분류하는 분류기 -> 특정 공간을 2개의 부분 공간으로 나누는 함수.

퍼셉트론의 구조



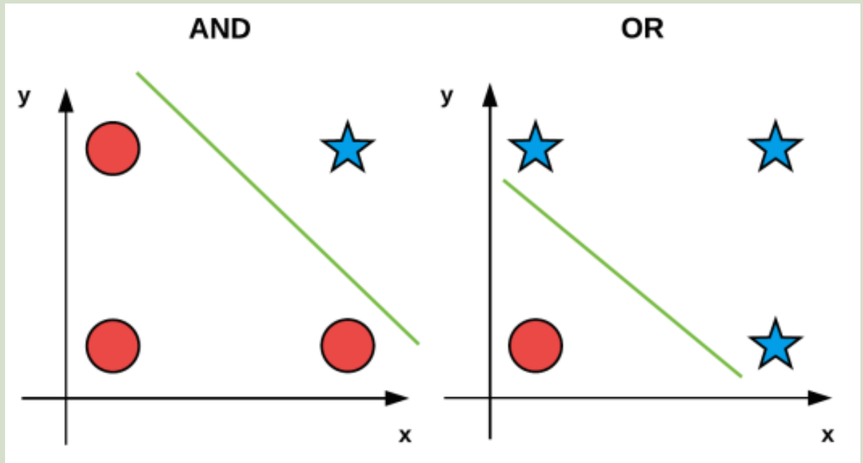
- 입력층에는 D차원 특징 벡터를 받기 위한 d개 노드와 1개의 바이어스 노드가 있다.
- 출력층에는 노드가 하나 있다.
- 입력 노드와 출력 노드를 연결하는 에지에 있는 W값은 가중치라고 한다.

- 출력 노드는 입력 노드의 값 x와 에지에 있는 가중치 w를 곱해서 얻은 d+1개의 곱셈 결과를 더해 s(Logit)를 구한 다음, s를 함수에 통과시켜 얻은 값 Output을 출력하는 구조.

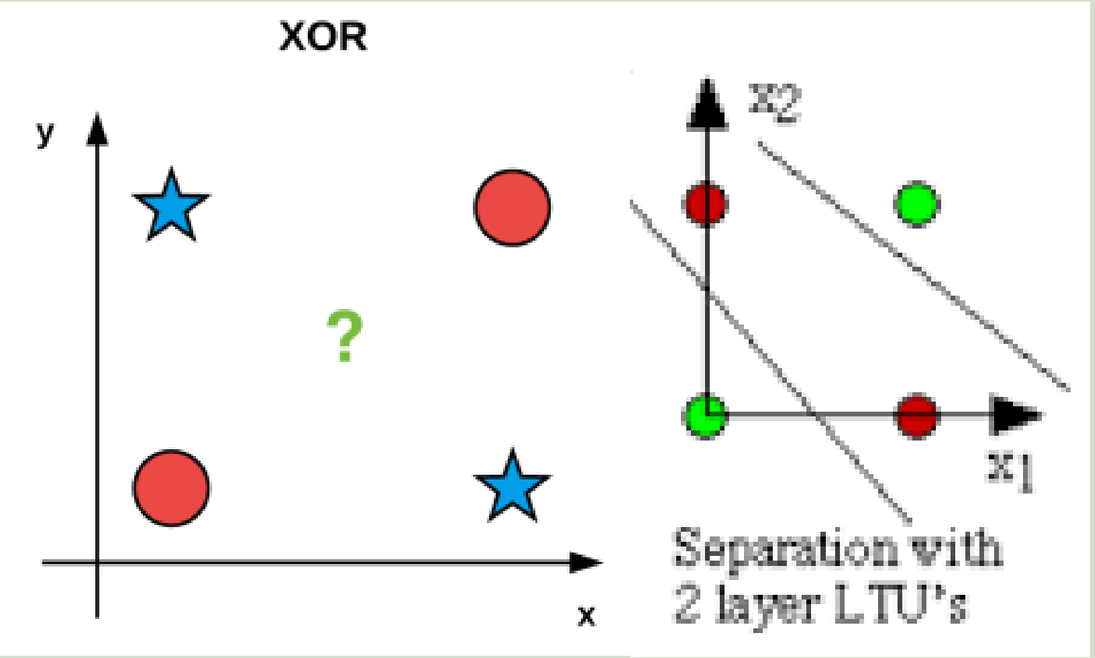
퍼셉트론의 연산

- 퍼셉트론은 학습에 의해 선형분리 가능 문제를 해결할 수 있음.

X1	X2	X1 AND X2	X1 OR X2
0 (F)	0 (F)	0 (F)	0 (F)
0 (F)	1 (T)	0 (F)	1 (T)
1 (T)	0 (F)	0 (F)	1 (T)
1 (T)	1 (T)	1 (T)	1 (T)



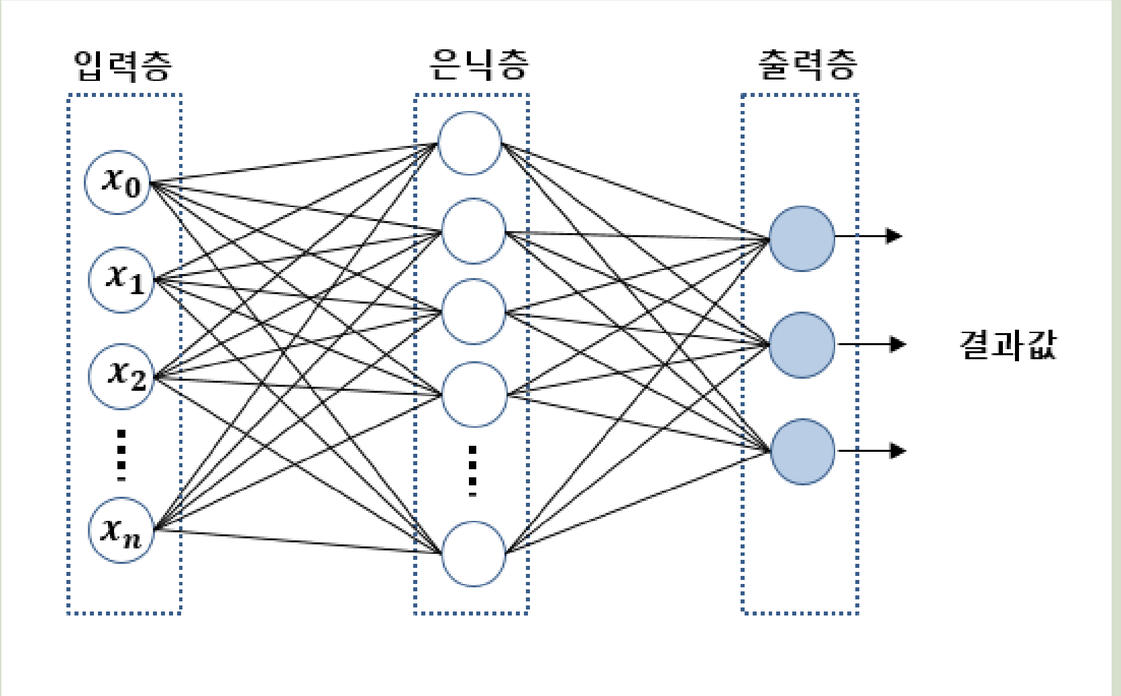
다층 퍼셉트론



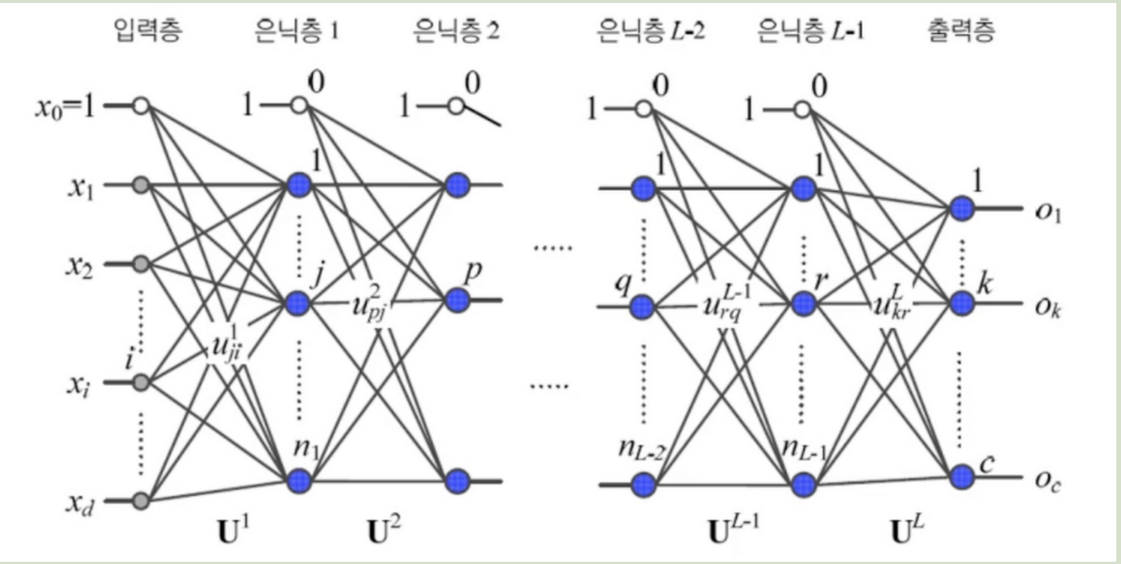
- XOR 논리연산을 단층 퍼셉트론이 풀 수 없음.

- 하지만, 다층 퍼셉트론을 사용하였을 때 이 문제를 해결할 수 있다.
- 입력층과 출력층 사이에 은닉층이 존재하는 퍼셉트론.

(이것이 단층 퍼셉트론과의 차이!)



깊은 다층 퍼셉트론



- L번째 층의 가중치 행렬
- 층 번호는 입력층을 0번째 층, 은닉층 l을 l번째 층, 출력층을 L번째 층으로 매긴다.

- L-1개의 은닉층
- 마지막 L층은 분류층

가중치 행렬: $\mathbf{U}^l = \begin{pmatrix} u_{10}^l & u_{11}^l & \cdots & u_{1n_{l-1}}^l \\ u_{20}^l & u_{21}^l & \cdots & u_{2n_{l-1}}^l \\ \vdots & \vdots & \ddots & \vdots \\ u_{n_l0}^l & u_{n_l1}^l & \cdots & u_{n_ln_{l-1}}^l \end{pmatrix}, l = 1, 2, \cdots, L$

활성 함수

- 입력 신호의 총합을 출력 신호로 변환하는 함수.
- 입력 신호의 총합이 활성화를 일으키는지를 정하는 역활을 한다.

- 활성화 함수를 사용하는 이유 : 입력값에 대한 출력값이 linear하게 나오지 않으므로, 선형분류기를 비선형 시스템으로 만들 수 있기 때문.

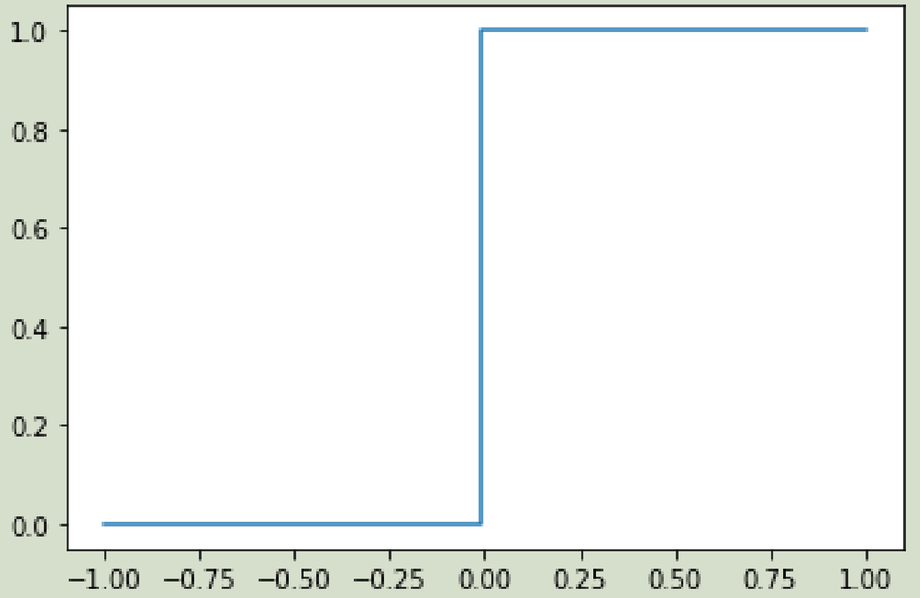
활성 함수 예시

계단 함수

- 입력이 0을 넘으면 1을 출력, 그 외에는 출력하는 함수.
- 이진 계단 함수는 0 or 1만을 출력하기 때문에 0인 부분은 미분이 불가능
- 1인 부분은 미분 시 모두 0으로 바뀐다.

-> 따라서 역전파 알고리즘을 통한 학습은 불가능함.

$$f(x)=\begin{cases}0 & \text{for } x<0\\ 1 & \text{for } x\geq 0\end{cases}$$

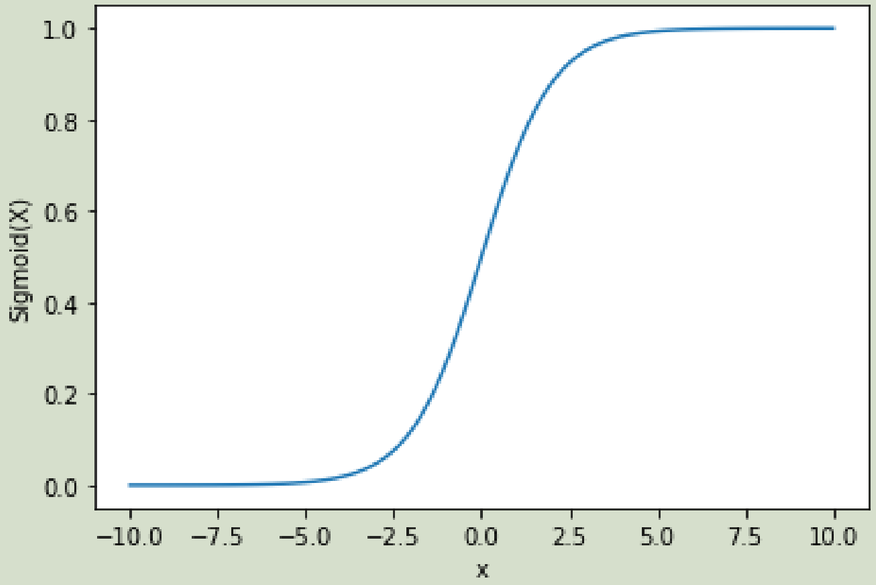


활성 함수

시그모이드 함수

- Logistic 함수라고 불리기도 하는 함수
- x의 값에 따라 0~1의 값을 출력하는 S자형 함수.
- 하지만 음수 값을 0에 가깝게 표현하기 때문에 기울기 소실 문제가 발생함.

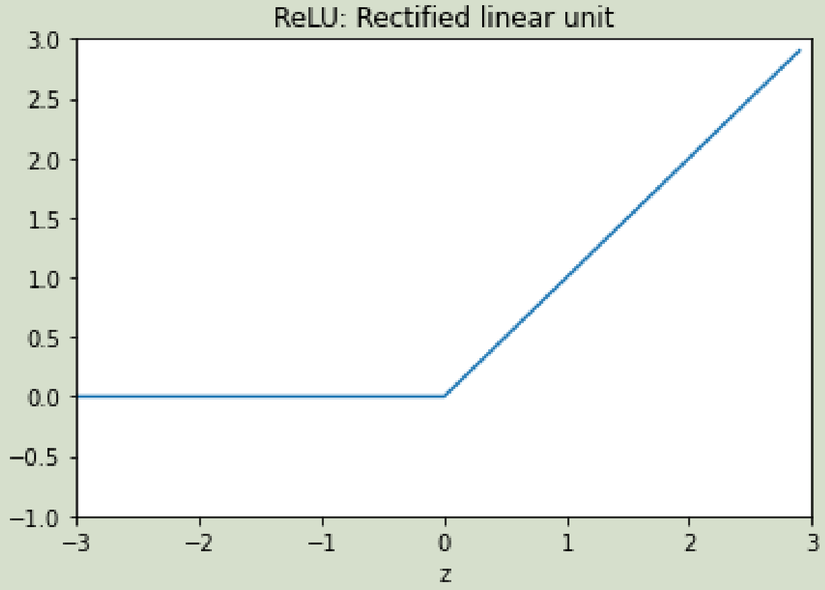
$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$



RELU 함수

- 현재 많이 사용하고 있는 활성화 함수.
- 입력이 0을 넘으면 그 입력을 그대로 출력, 0 이하이면 0을 출력하는 함수
- 이는 0보다 작은 값들에서 뉴런이 죽을 수 있는 단점을 야기
- 시그모이드 함수보다 학습이 빠르고, 연산 비용이 적으며, 구현이 간단하다는 장점이 있음.

$$f(x) = max(0, x)$$



활성 함수

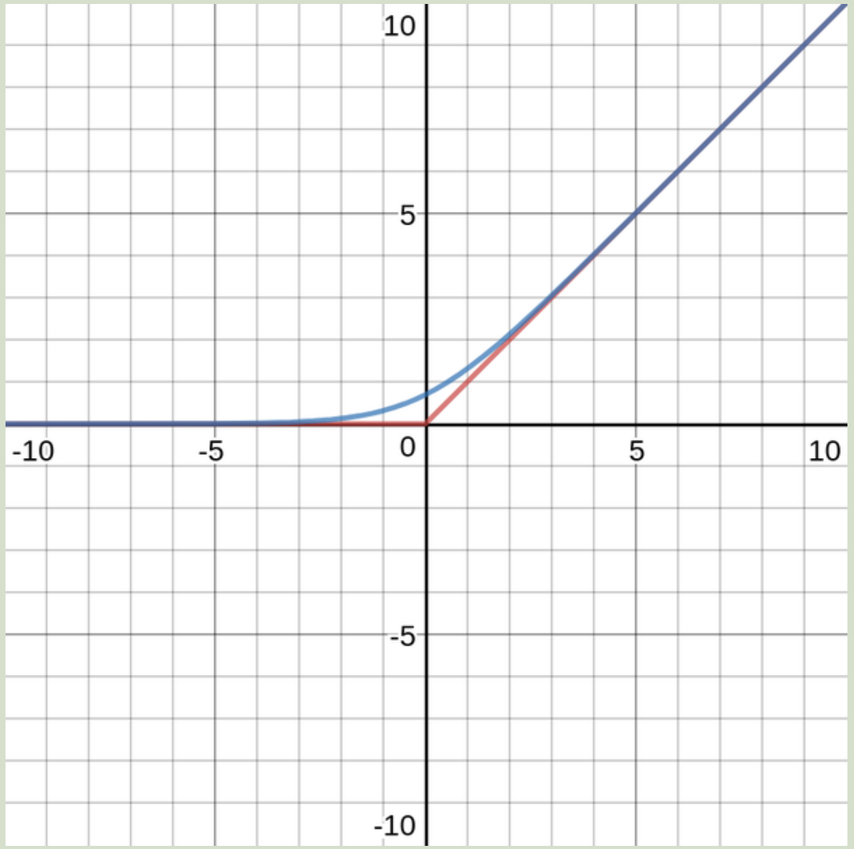
Softmax 함수

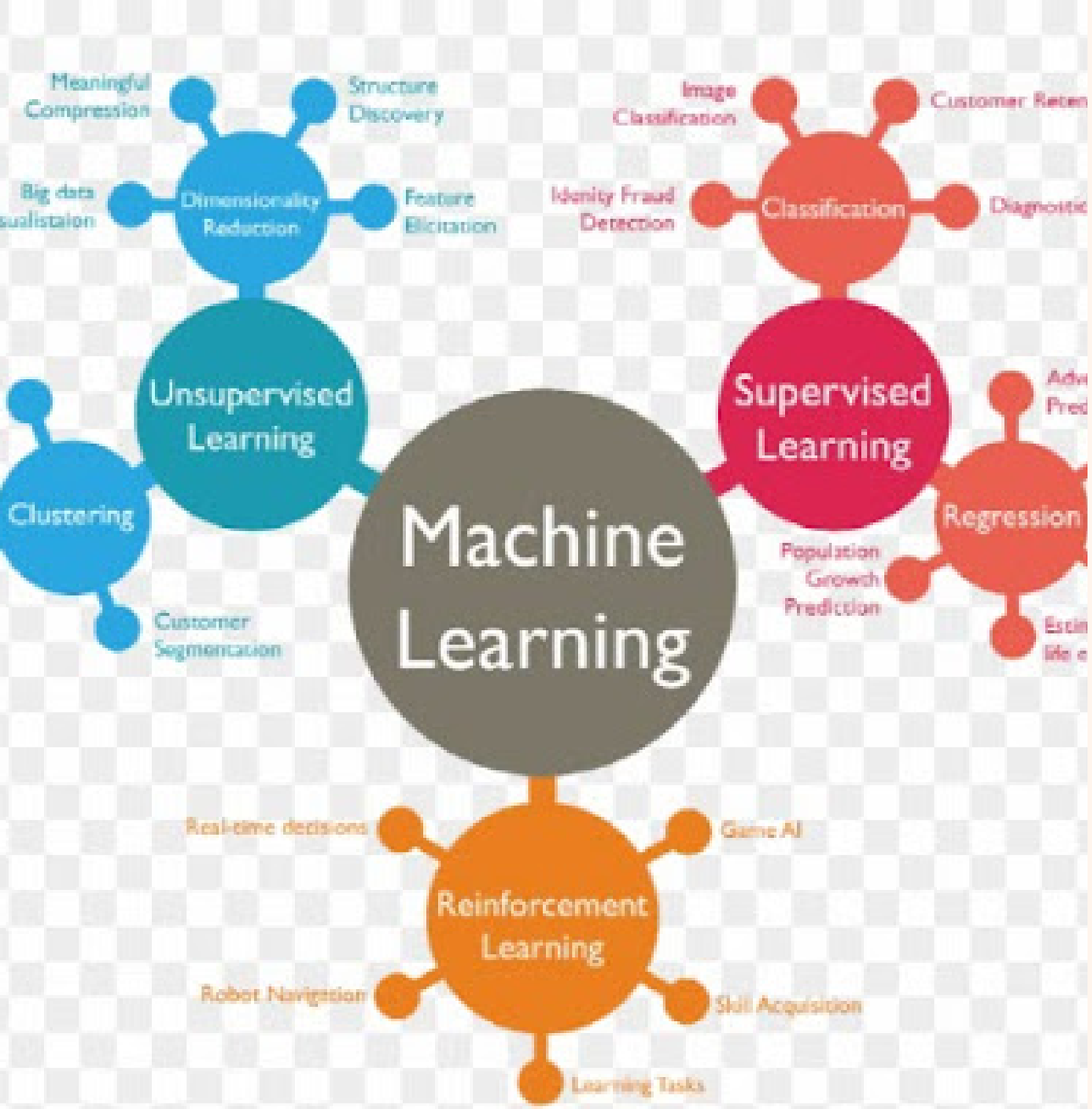
- 신경망의 출력층에서 주로 사용되는 함수.
- 모든 출력 노드를 같이 고려함.

- 확률을 계산할 때 사용하는 함수, 출력 벡터는 원소의 합이 1인 확률함수.
- 벡터의 각 원소를 정규화 하기위해 소프트멕스 함수는 입력값을 모두 지수 값으로 바꿔줌.

- 다중분류 로지스틱이라는 것을 나타낼 때 유용하게 사용.
- 확률로 간주할 수 있기에 신경망의 최종 출력으로 적합.

$$softmax(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

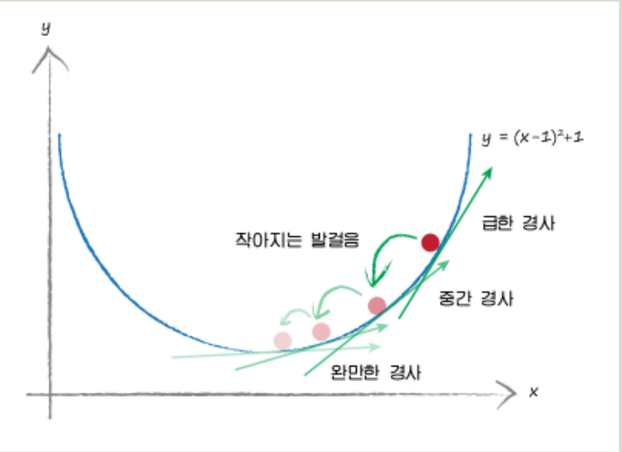



Machine Learning
Unsupervised Learning
Supervised Learning
Reinforcement Learning
Dimensionality Reduction
Clustering
Classification
Regression
Meaningful Compression
Structure Discovery
Feature Elicitation
Big data visualisation
Image Classification
Identify Fraud Detection
Customer Segmentation
Population Growth Prediction
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

학습 알고리즘

경사 하강법

- 미분값을 보고 함수가 낮아지는 방향을 찾아 이동하는 일을 반복하여 최저점에 도달하는 알고리즘.
- 고차원 방정식에 대한 문제를 해결해주면서, 비용 함수를 최소화하는 방법을 직관적으로 제공.



- 핵심은 오류가 작아지는 방향으로 가중치 값을 업데이트 하는 방법.

$$x_{i+1} = x_i - \alpha \frac{df}{dx}(x_i)$$

스토캐스틱 경사하강법으로 확장

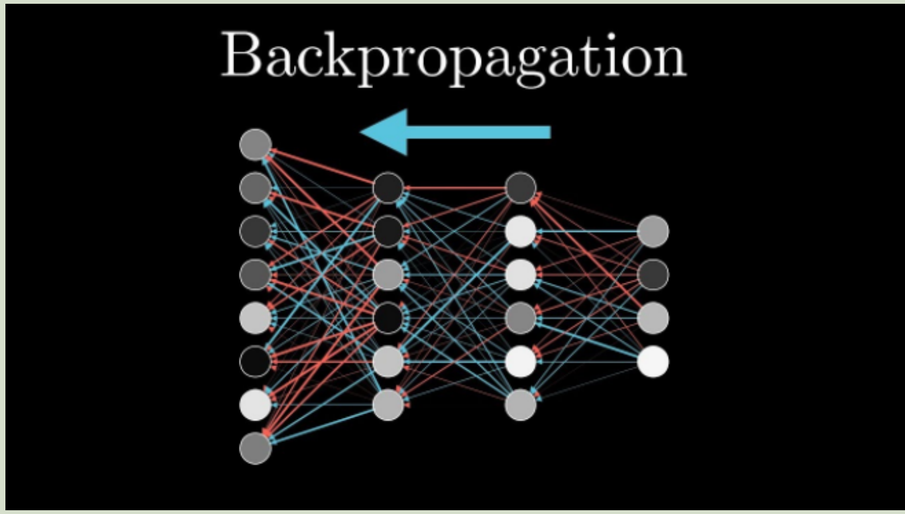
- 딥러닝에 적용하기 위해 알고리즘을 확장해야 하는 것에서 배경.

- 미니 배치를 사용하는 확장된 경사하강법, 보통 SGD라고 부른다.(배치 크기는 1)
- 데이터 세트에서 무작위로 균일하게 선택한 하나의 예를 의존하여 각 단계의 예측 경사를 계산

역전파 알고리즘

- 오차가 본래 진행방향과 반대방향으로 전파된다고 하여 붙여진 이름.
- 다층 퍼셉트론을 학습한다는 것은 최종 출력값과 목표값의 오차가 최소화되도록 가중치를 결정하는 것(순전파, 역전파)

- 과정 : 순전파를 진행하여, 순전파 과정에서 발생한 오차를 역방향으로 진행. (출력층 -> 은닉층 -> 입력층), 오차가 최소화되는 방향으로 가중치를 조정하는 과정이다. (오차가 0에 가까워질 때까지 반복 학습)



Chain Rule

- 합성함수의 미분에 관한 법칙, 합성함수의 미분은 합성 함수를 구성하는 각 함수의 미분의 곱으로 나타낼 수 있다.

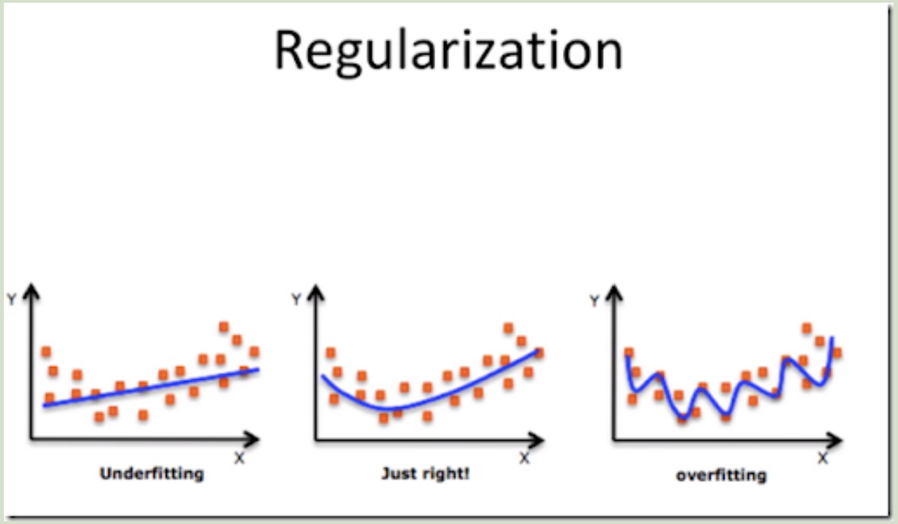
$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t} * \frac{\partial t}{\partial x}$$

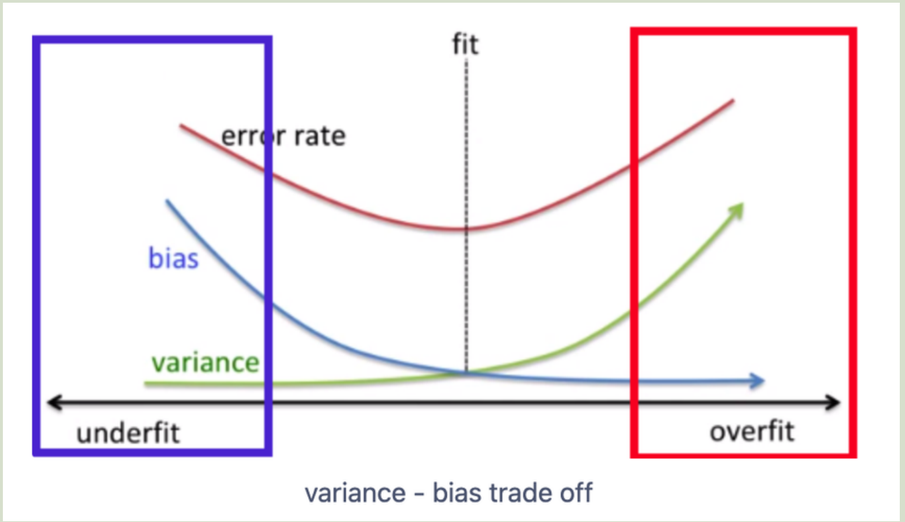
$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t} * \frac{\partial t}{\partial x}$$

Overfitting

- 기존의 다층 신경망에서 가장 큰 문제는 ‘학습 데이터‘를 너무 과하게 학습함으로써 발생하는 문제
- 이를 ‘오버피팅’ 문제라고 말함.

- 학습 데이터에 대해서는 학습이 너무 과하게 되어 높은 성능을 나타내지만, 실제 적용 시에는 모델이 일반화가 되지 않아 성능이 떨어지는 현상.





- 모델 복잡도 면에서 언더피팅(underfitting)과 오버피팅(overfitting)과의 트레이드 오프(trade-off) 관계를 보여주고 있다.
- 위의 그림을 통해서 대게 모델이 복잡할수록 예측 오차는 점점 작아지다가 다시 커지는 현상이 발생하는 것을 볼 수 있다.

- 따라서 최적의 모델을 찾는 것이 중요하다.(파라미터 조정, Dropout, 규제 등)