

# Deep learning

1871005 강예준

<https://youtu.be/5Yt6l6QwPOg>

# Contents

1. Deep learning

2. Data

3. Parameter

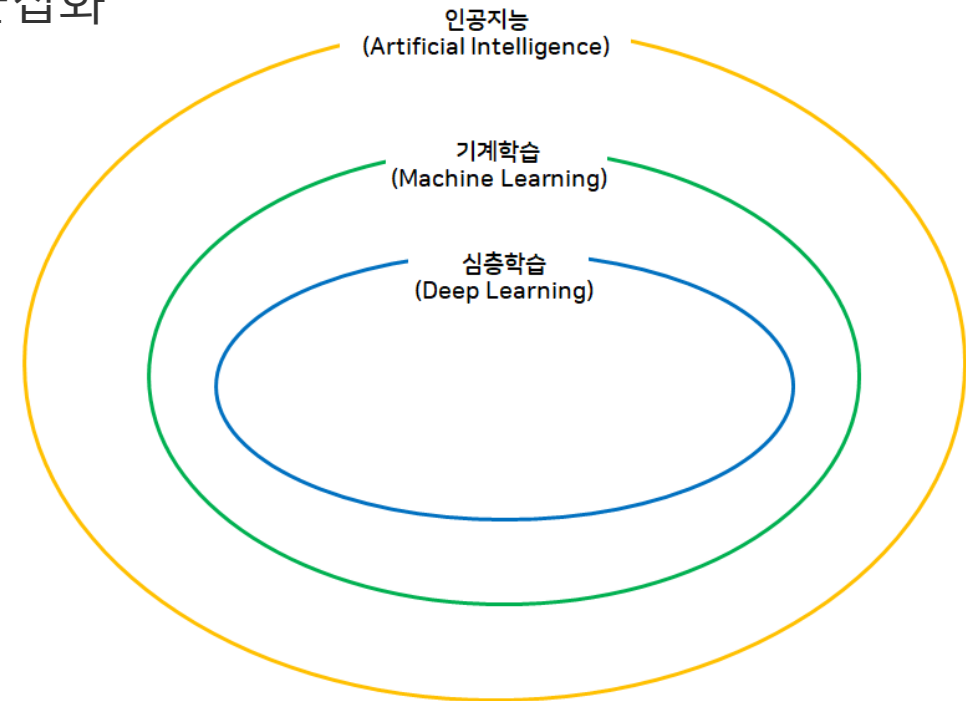
4. Activation function



# machine learning

## ◆ machine learning

- 지도학습 : 입력값이 주어지면 입력값에 대한 **Label**를 주어 학습
- 비지도학습 : 정답을 따로 알려주지 않고, 비슷한 데이터들을 군집화
- 강화학습 : 자신이 한 행동에 대해 보상(reward)를 받으며 학습
- 목표 : 최적의 parameter 설정을 통한 최적화 -> 높은 정확도  
=> **일반화**



# Deep learning

## ◆ Neural Network 종류

1. 합성곱신경망 : Convolutional Neural Network (CNN)
2. 순환신경망 : Recurrent Neural Network (RNN)
3. 생산적 적대 신경망 : Generative Adversarial Neural NetWork (GAN)

# Deep learning process

- Deep learning process
  1. Data generation
  2. Modeling
  3. Training
  4. Validation
  5. Test

# deep learning

- Data의 구성

1. Training set : 매개변수(가중치와 편향) 학습
2. Validation set : 모델의 성능을 평가하기 위한 데이터
3. Test set : 최종 선택된 모델을 통해 분류할 데이터

# deep learning

- 과적합 (Overfitting)

- > 한 데이터셋에만 지나치게 최적화된 상태
- > 학습 데이터에 대해 과하게 학습하여 실제 데이터에 대한 **오차가 증가**하는 현상
- > Training data에 대해 학습 결과 데이터들에 대해 높은 정확도를 갖지만,  
test data 사용 시 일반화 성능이 떨어지는 현상 (**일반화X**)

- Overfitting 방지 기법

- 가중치 감소(weight decay) : 큰 가중치에 대해서는 그에 상응하는 큰 **패널티** 부과
- Drop out : 뉴런을 임의로 삭제하면서 학습하는 방법

# Parameter

- Parameter

- ❖ Parameter

- **가중치**와 **편향**
    - 각 layer를 거치면서 계산되는 값
    - 학습 과정에서 모델이 알아서 수정

- ❖ Hyper-parameter

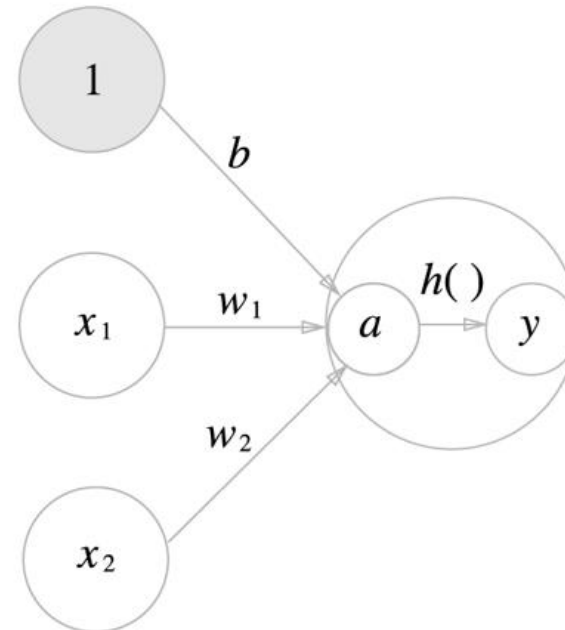
- Data와 Model에 맞게 **사람이 직접** 설정
    - 노드 개수, 활성화 함수, metric 등
    - Hyper-parameter 수정을 통해 **일반화 성능 개선**



# 활성화 함수

- Activation function

- 입력 신호의 총합을 출력 신호로 변환하는 함수
- 입력 신호의 총합이 활성화를 일으키는지 정하는 역할
- **비선형함수**를 사용해야 함
- Step, Sigmoid, ReLU, SoftMax 등



$$a = b + w_1x_1 + w_2x_2$$

$$y = h(a)$$

# 활성화 함수

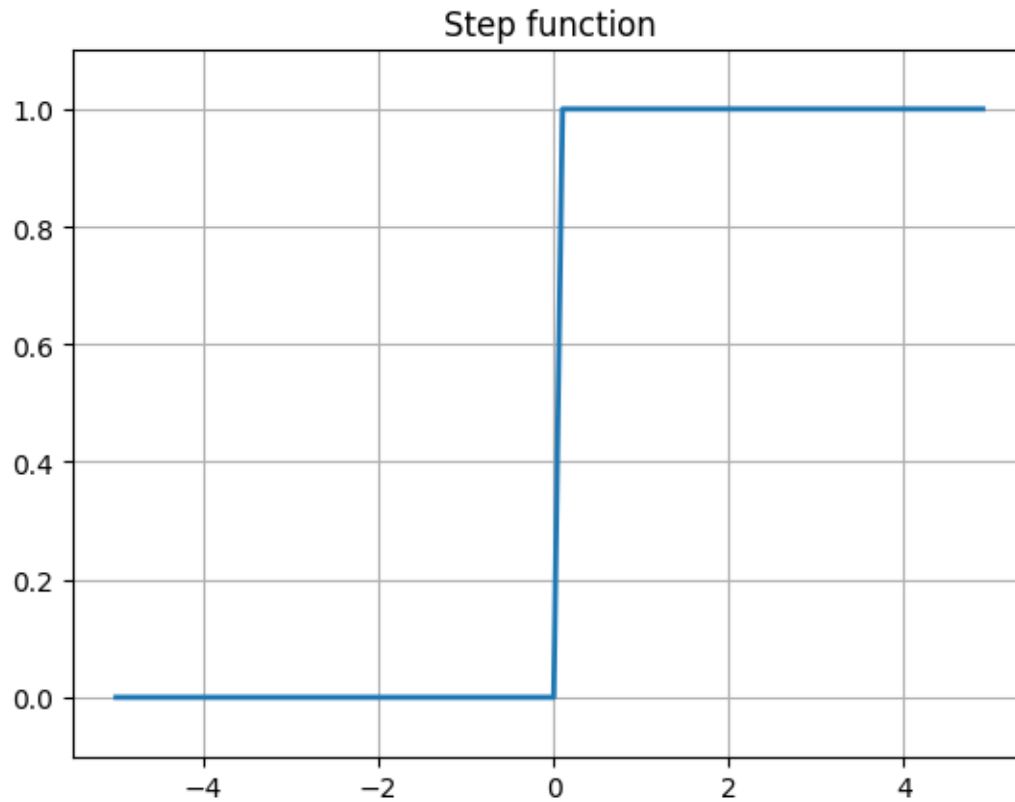
- 활성화 함수에 선형함수 사용시 은닉층의 존재 이유가 사라짐  
-> 은닉층이 없는 네트워크가 됨

$$h(x) = cx \quad \rightarrow \quad y(x) = h(h(h(x))) \quad \rightarrow \quad y(x) = c * c * c * x \quad \rightarrow \quad y(x) = ax$$

$$a=c^3$$

# 활성화 함수

- 계단함수 (Step function)

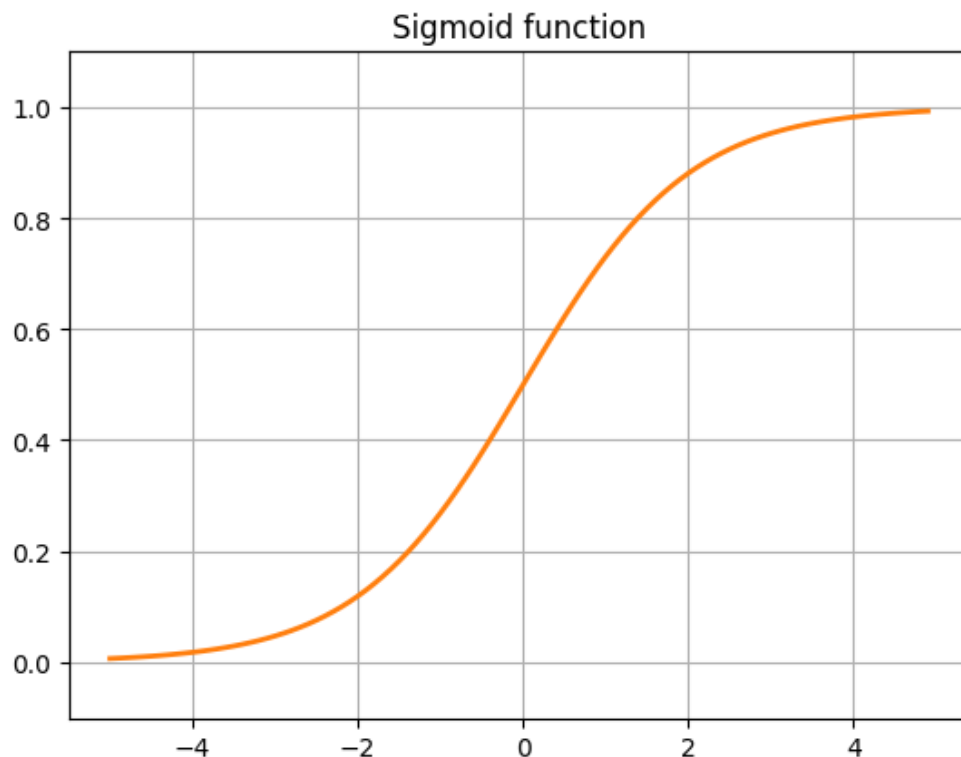


$$h(x) = \begin{cases} 0 & (x \leq 0) \\ 1 & (x > 0) \end{cases}$$

# 활성화 함수

- 시그모이드 함수 (Sigmoid function)

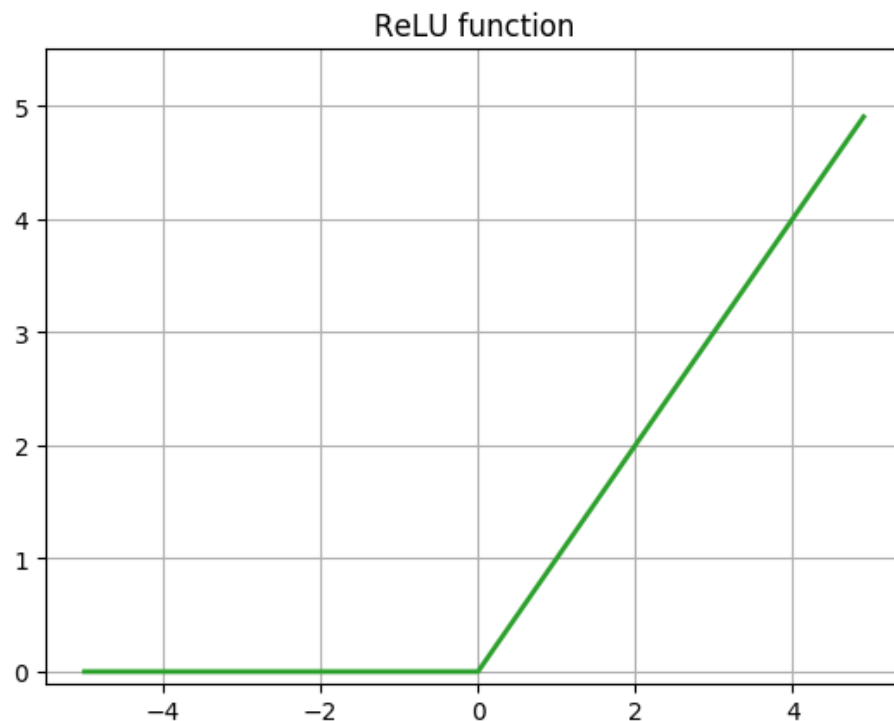
- **Binary** 분류에 주로 사용 ∵ 그래프 양극화
- **기울기 소실** (Vanishing gradient) : 출력값이 0과 1사이 -> 층이 많아질 수록 최종 기울기가 **0에 수렴**



$$h(x) = \frac{1}{1 + \exp(-x)}$$

# 활성화 함수

- ReLU 함수 (Rectified Linear Unit)
  - 은닉층에서 주로 사용
  - 학습속도 ↑, 연산비용 ↓, 간단한 구현
  - $x \leq 0$ 면 뉴런이 죽음
  - Leaky ReLU, PReLU, ELU



$$h(x) = \begin{cases} x & (x > 0) \\ 0 & (x \leq 0) \end{cases}$$

# 선형함수

- 함수가 선형성을 갖기 위한 조건

1. 중첩 (Superposition)

- $f(x_1+x_2) = f(x_1)+f(x_2)$
- $f(-1+1) = f(-1)+f(1) \rightarrow \text{ReLU에서는 틀린 수식}$
- $f(0) = 0 \quad f(-1) = 0 \quad f(1) = 1$

2. 동질성 (Homogeneity )

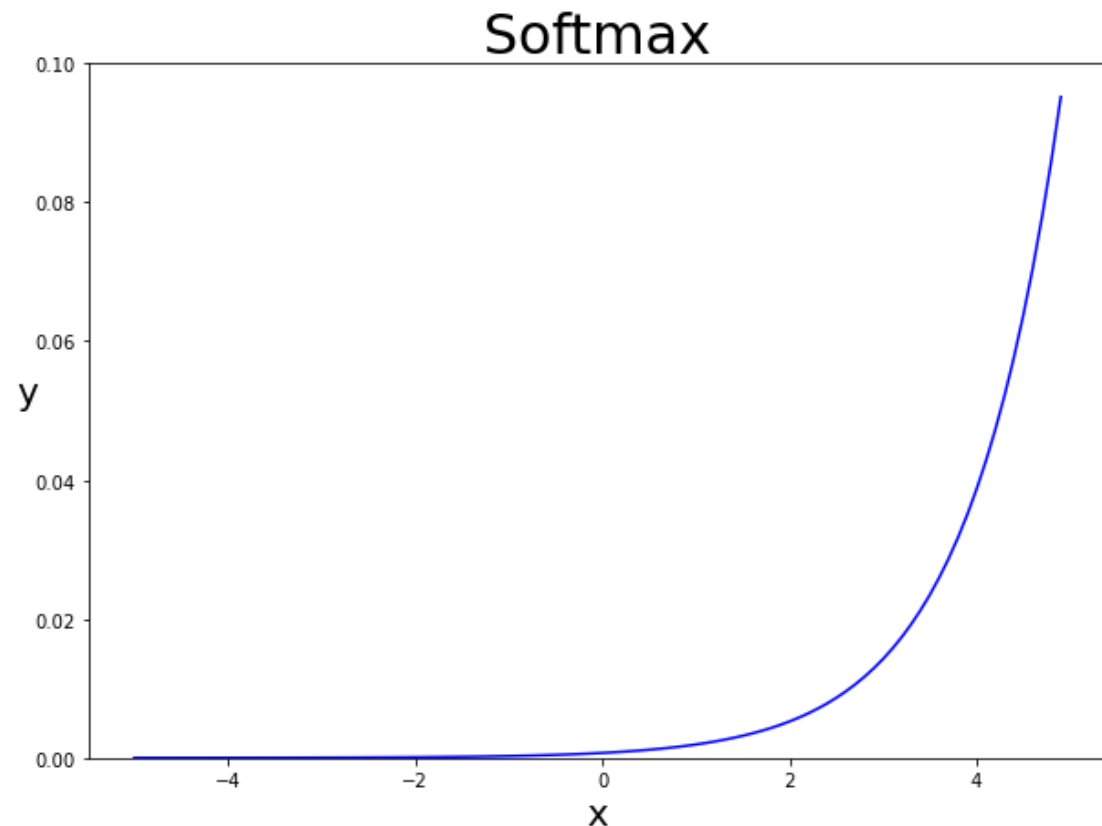
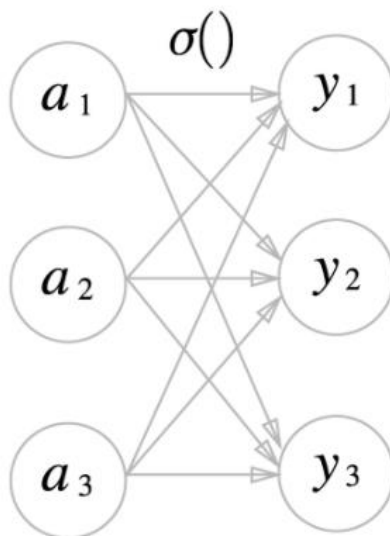
- $f(ax) = af(x)$

# 활성화 함수

- 소프트 맥스 함수

- 출력은 **모든 입력 신호로부터** 화살표를 받음
  - > 출력층의 각 뉴런이 모든 입력 신호로부터 영향을 받음
- **다중 클래스** 분류
- 출력값은 0에서 1.0 사이의 실수
- => 모든 출력값의 합은 1

$$y_k = \frac{\exp(a_k)}{\sum_{i=1}^n \exp(a_i)}$$



Q & A

