

# Deep Learning 기초

임세진

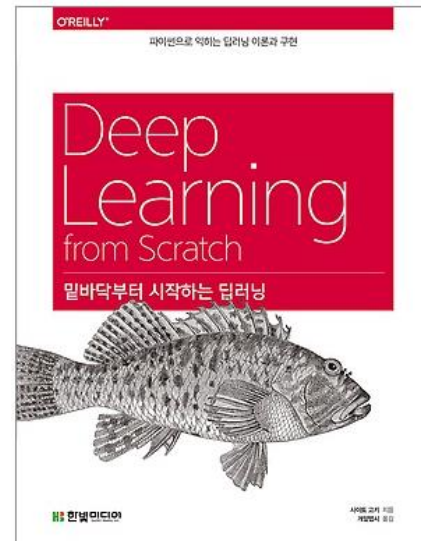
<https://youtu.be/Dnt3TAH9QHc>

# Contents

01. 퍼셉트론과 신경망

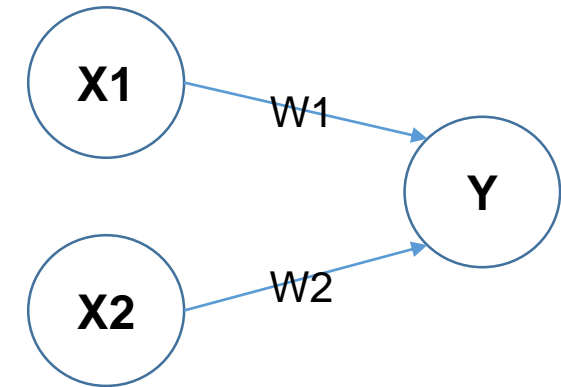
02. 활성화 함수

03. 신경망 학습



# 01. 퍼셉트론과 신경망

- 퍼셉트론 (perceptron)
  - 다수의 신호를 입력으로 받아 하나의 신호를 출력하는 알고리즘
  - 퍼셉트론 신호는 흐른다 or 안 흐른다 (1 or 0) 두가지 값만 가짐
  - 뉴런에서 보내온 신호의 총합 > 임계값 → 1을 출력



$$y = \begin{cases} 0 & (w_1 x_1 + w_2 x_2 \leq \theta) \\ 1 & (w_1 x_1 + w_2 x_2 > \theta) \end{cases} \quad y = \begin{cases} 0 & (b + w_1 x_1 + w_2 x_2 \leq 0) \\ 1 & (b + w_1 x_1 + w_2 x_2 > 0) \end{cases}$$

↘ 편향(bias)

- 퍼셉트론의 매개변수 값은 사람이 정해야함

(결과가 1로 출력)

편향(bias) : 뉴런이 얼마나 쉽게 **활성화** 하느냐를 조절하는 매개변수

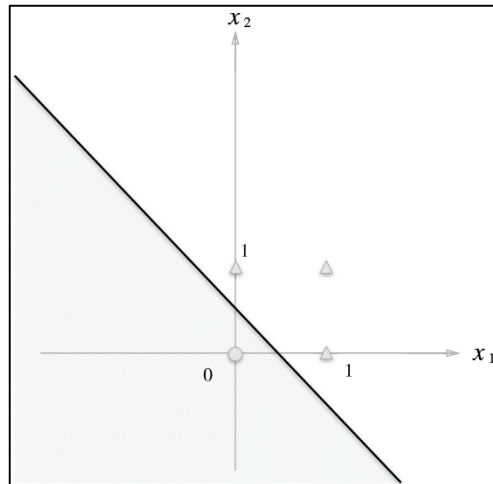
가중치(weight) : 입력신호가 결과에 주는 영향력을 조절하는 매개변수

# 01. 퍼셉트론과 신경망

- 선형과 비선형

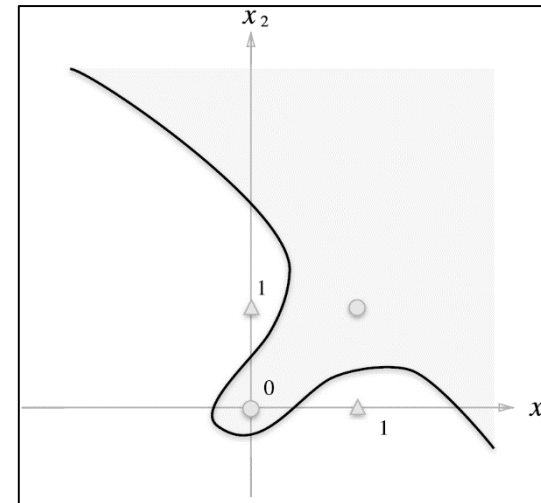
$x_1$	$x_2$	$y$
0	0	0
1	0	1
0	1	1
1	1	1

OR Gate



$x_1$	$x_2$	$y$
0	0	0
1	0	1
0	1	1
1	1	0

XOR Gate



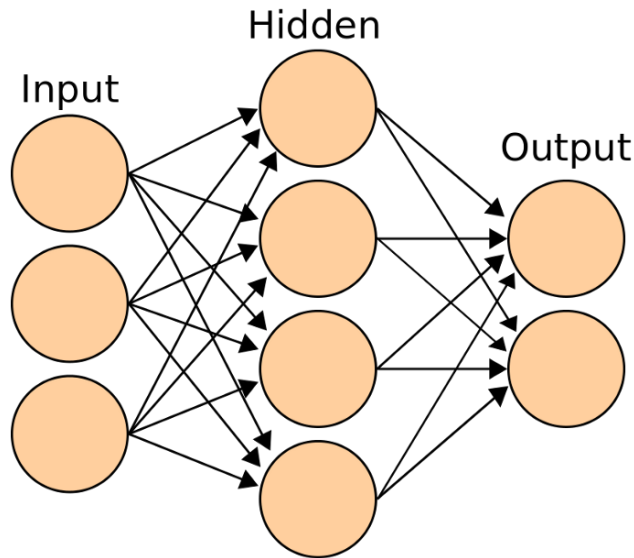
한계 : (단층) 퍼셉트론은 직선 하나로 나눈 영역만 표현할 수 있음 → 다층 퍼셉트론을 통해 비선형 영역도 표현 가능

# 01. 퍼셉트론과 신경망

- 신경망

- 퍼셉트론의 장점 : 층을 쌓음으로써 복잡한 함수도 표현이 가능
- 퍼셉트론의 단점 : 매개변수를 사람이 수동으로 정해야함

➔ 신경망은 매개변수의 적절한 값을 데이터로부터 자동 학습 (자동으로 정해줌)



$$y = \begin{cases} 0 & (b + w_1x_1 + w_2x_2 \leq 0) \\ 1 & (b + w_1x_1 + w_2x_2 > 0) \end{cases}$$



$$y = h(b + w_1x_1 + w_2x_2)$$

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

## 02. 활성화 함수

- 활성화 함수 (activation function)

- 입력 신호의 총합을 출력 신호로 변환하는 함수
- 입력 신호의 총합이 활성화를 일으키는지를 정하는 역할

$$y = h(b + w_1x_1 + w_2x_2)$$

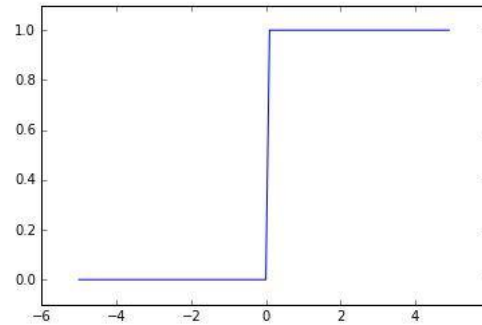
$$h(x) = \begin{cases} 0 & (x \leq 0) \\ 1 & (x > 0) \end{cases}$$

- 계단 함수(step function)

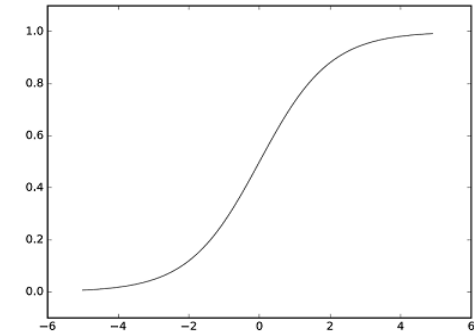
퍼셉트론에서 사용하는 활성화 함수

- 시그모이드 함수 (sigmoid function)

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



계단 함수



시그모이드 함수

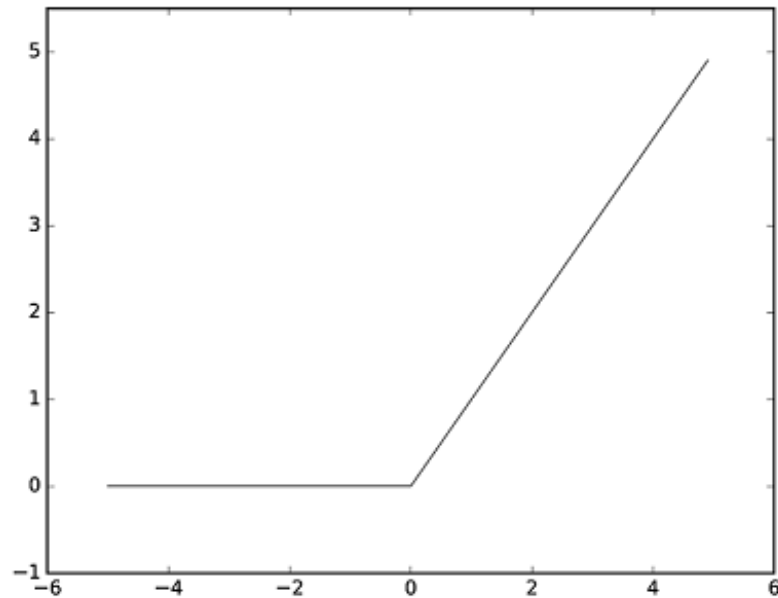
두 함수의 공통점 : 입력 값이 중요도와 출력값의 크기가 비례,  $[0 \leq \text{출력값} \leq 1]$ , 비선형 함수

두 함수의 차이점 : 연속적인 결과값 여부, 실수 값 반환 여부

## 02. 활성화 함수

- ReLU 함수 (Rectified Linear Unit)

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



## 02. 활성화 함수

- 출력층 설계 - 문제의 종류 파악

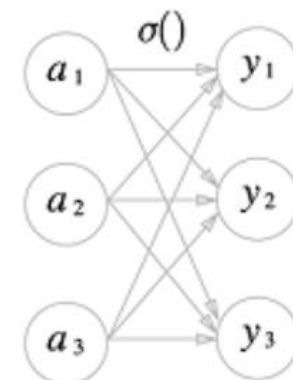
- 분류 (classification) : 데이터가 어느 클래스에 속하는지 분류하는 문제

- Ex) 강아지 사진 → 강아지인지 고양이인지 분류

- 소프트맥스 함수 (softmax function)

- 출력층의 각 뉴런이 모든 입력 신호에서 영향을 받음
- 0과 1사이의 실수값
- 원소의 대소 관계는 변하지 않음 → 보통 추론 시에는 소프트맥스 함수를 생략함
- 출력의 총합은 1 → 소프트맥스 함수의 출력을 '확률'로 해석 가능

$$y_k = \frac{\exp(a_k)}{\sum_{i=1}^n \exp(a_i)} \quad \begin{array}{l} n : \text{출력층의 뉴런 수} \\ k : k\text{번째 출력} \end{array}$$

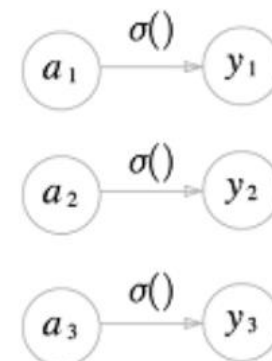


소프트맥스 함수

- 회귀 (regression) : 입력 데이터에서 (연속적인) 수치를 예측하는 문제

- Ex) 사람 사진 → 사진 속 인물의 몸무게 예측 (50.3kg ?)

- 항등 함수 (identity function) : 입력을 그대로 출력



항등 함수

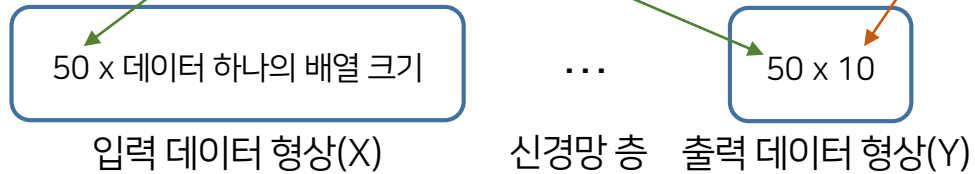


# 03. 신경망 학습

- 배치 처리

- Ex) 이미지 여러 장(50장)을 한꺼번에 입력하는 경우 + 10개의 class로 분류

→ 50장 분량의 입력 데이터의 결과 한 번에 출력



데이터 하나의 분류 결과 값

X \ Y	0	1	2	3	4	5	6	7	8	9
0	0.01	0.8	0.01	0.02	0.16	0	0	0	0	0
1										
2										
⋮										
49										

- 배치 (batch) : 하나로 묶은 입력 데이터

- 배치 처리의 이점 : 이미지 1장 당 처리 시간을 대폭 줄여줌 [효율+신속]

- 대부분의 수치 계산 라이브러리가 큰 배열을 효율적으로 처리하도록 최적화됨
- 큰 규모의 신경망의 경우, 데이터 전송 부분에서 병목이 일어나는 경우가 종종 있음 → 배치 처리로 버스에 주는 부하 감소

즉, 컴퓨터가 큰 배열을 한꺼번에 계산하는 속도 > 분할된 작은 배열을 여러 번 계산하는 속도

## 03. 신경망 학습

- 학습
  - 훈련 데이터로부터 **가중치 매개변수의 최적값**을 **자동**으로 획득하는 것 [자동 결정]
  - 신경망의 학습 지표 : 손실 함수
  - 학습의 목표 : 손실 함수의 결과값을 **가장 작게 만드는** 가중치 매개변수 찾기

범용적으로 사용할 모델을 만들기 위해 데이터를 나눈 것임 (범용 능력을 평가하기 위해)

데이터

훈련 데이터 (training data) : 이 데이터만을 사용하여 학습 → 최적의 매개변수 찾기

시험 데이터 (test data) : 이 데이터를 사용하여 위에서 훈련한 모델의 실력 평가

## 03. 신경망 학습

- 손실 함수 (loss function)
  - 신경망 학습에서 최적의 매개변수 값을 탐색하기 위해 사용하는 지표
  - 오차제곱합 (sum of squares for error, SSE)

$$E = \frac{1}{2} \sum_k (y_k - t_k)^2$$

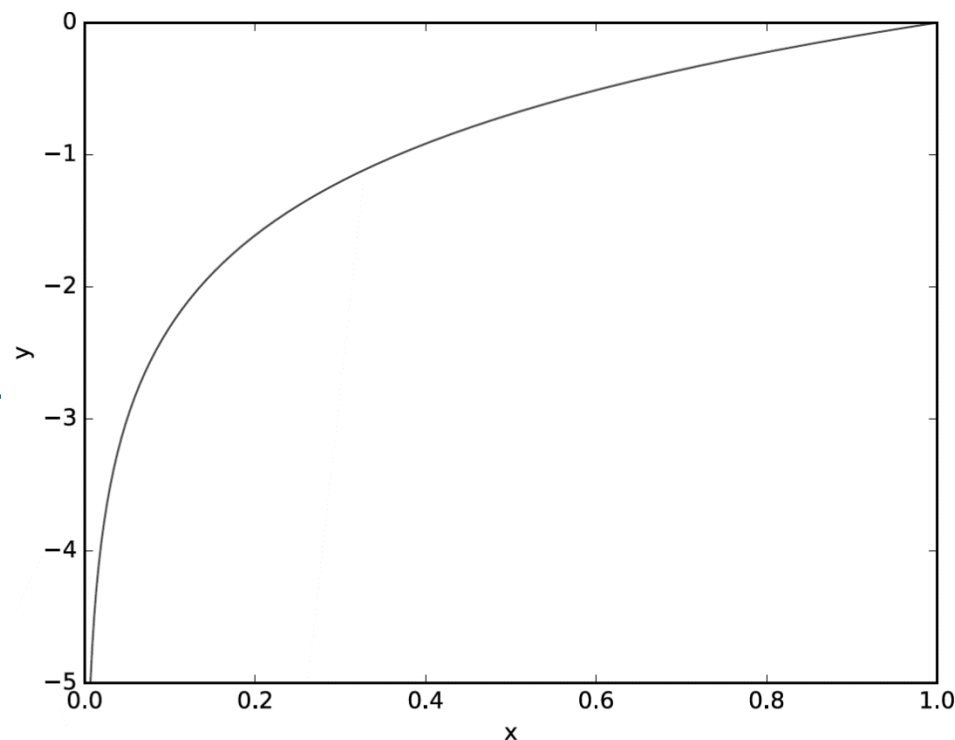
$y$  : 신경망의 출력 (신경망이 추정한 값)  
 $t$  : 정답 레이블  
 $k$  : 데이터의 차원 수

정답에 더 가까움

- 교차 엔트로피 오차 (cross entropy error, CEE)

$$E = - \sum_k t_k \log y_k$$

$t$ 는 정답이 아니면 0이므로 정답일 때만 계산하게 됨



감사합니다

