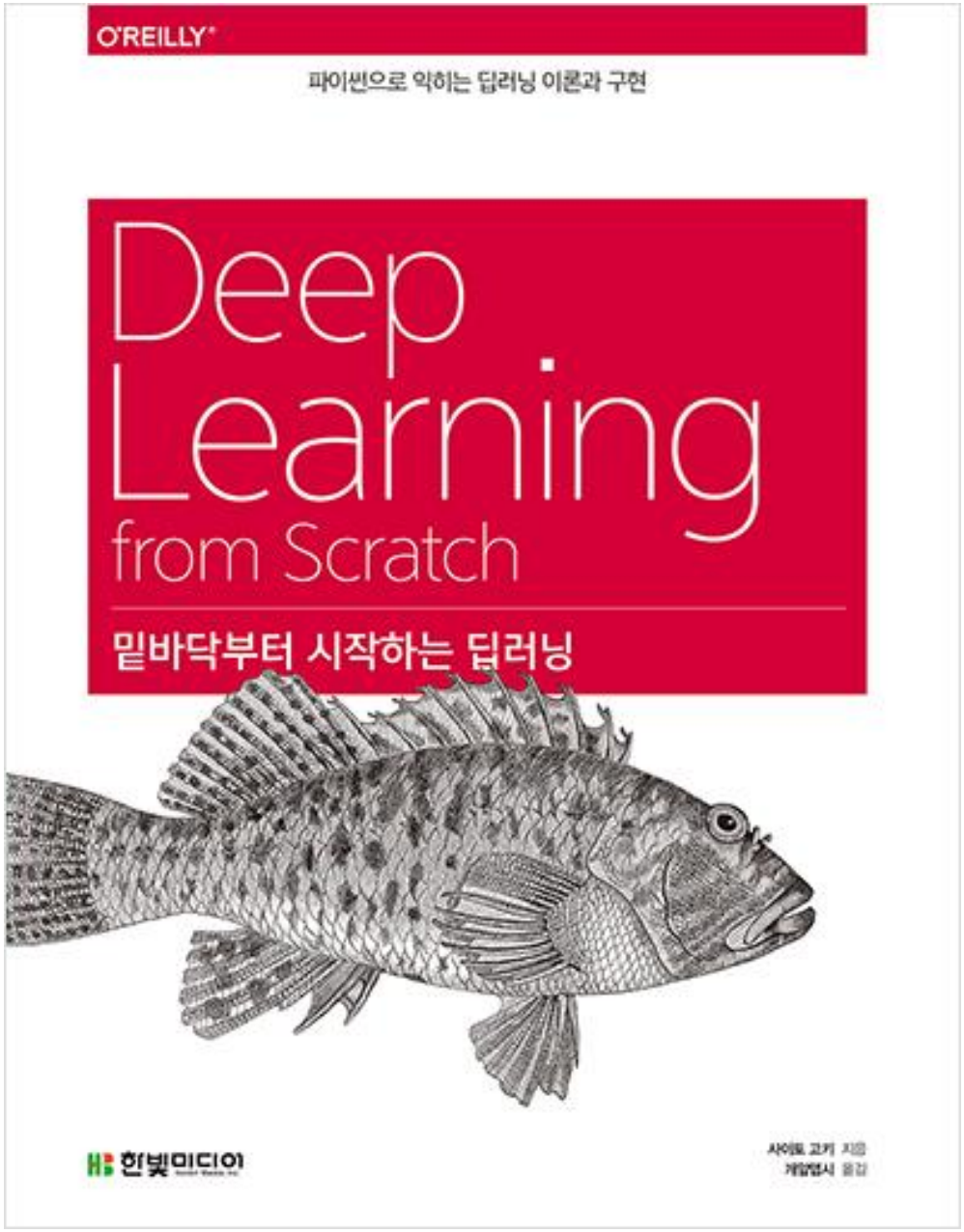



〈밑바닥부터 시작하는 딥러닝〉

# 전반부 정리

## Ch 02~05

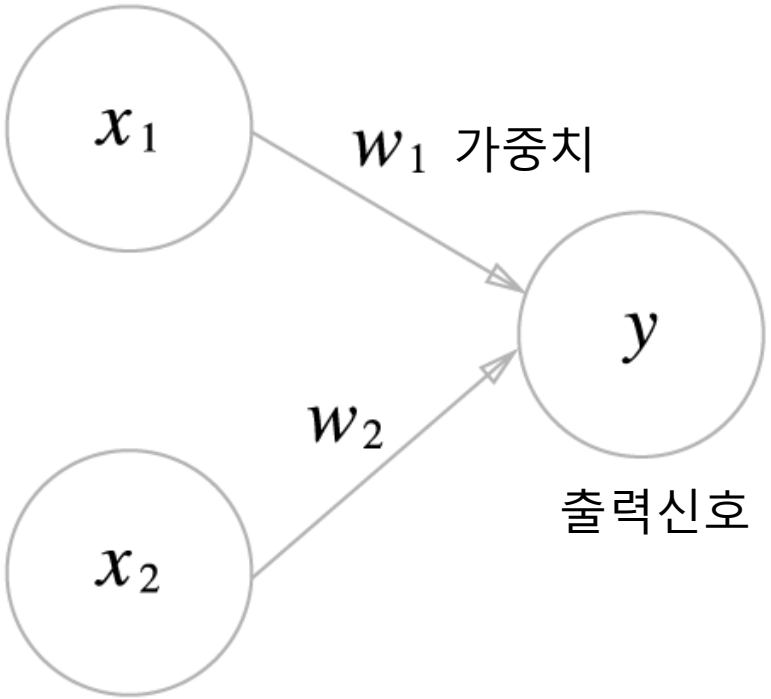
2021.02.20 (토) 5주차 스터디

발표자 : 김유진

# chapter2. 퍼셉트론

# 퍼셉트론이란?

- 신경망(딥러닝)의 기원이 되는 알고리즘
- 다수의 신호를 입력으로 받아 하나의 신호를 출력




임계값

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

신호의 총합이 임계값 이상일 때만 1 출력

# 단순 논리 회로

AND

| $x_1$ | $x_2$ | $y$ |
|---|---|---|
| 1 | 1 | 1 |

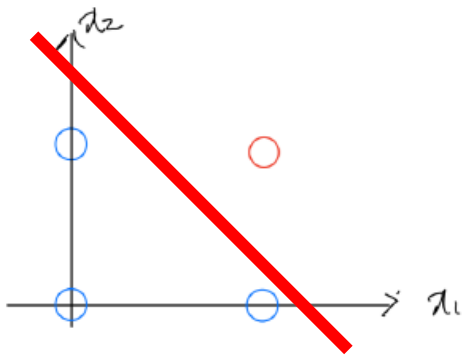
NAND

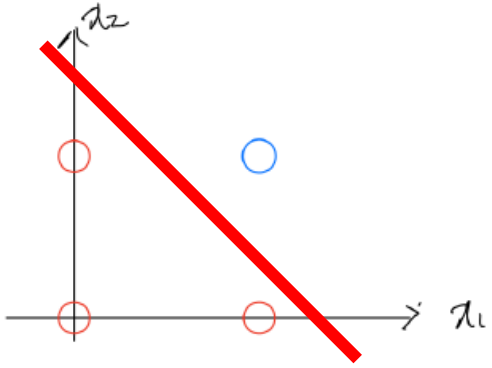
| $x_1$ | $x_2$ | $y$ |
|---|---|---|
| 1 | 1 | 0 |

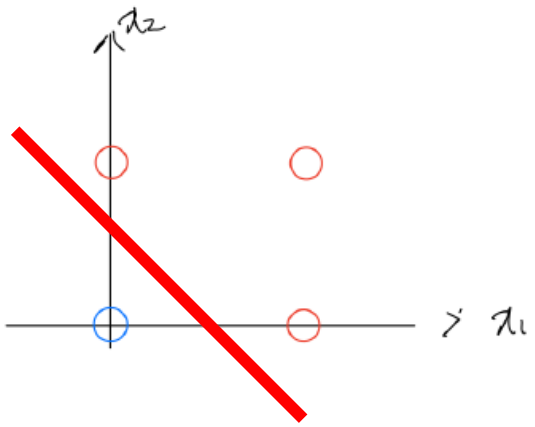
OR

| $x_1$ | $x_2$ | $y$ |
|---|---|---|
| 1 | 1 | 1 |

- 매개변수(가중치와 임계값)을 적절히 조정했을 뿐 모두 같은 퍼셉트론 구조
  - 학습이란 적절한 매개변수 값을 정하는 작업
- 기계학습 문제는 매개변수 값을 정하는 작업을 컴퓨터가 자동으로 하도록!

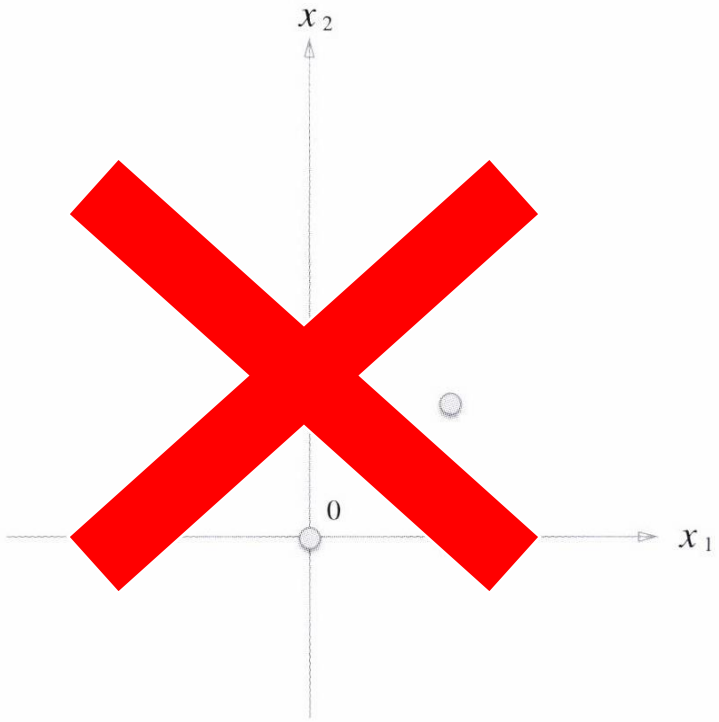


# 퍼셉트론의 한계 : XOR 게이트

- 직선 하나로 나눈 영역(=선형 영역)만 표현할 수 있다는 한계

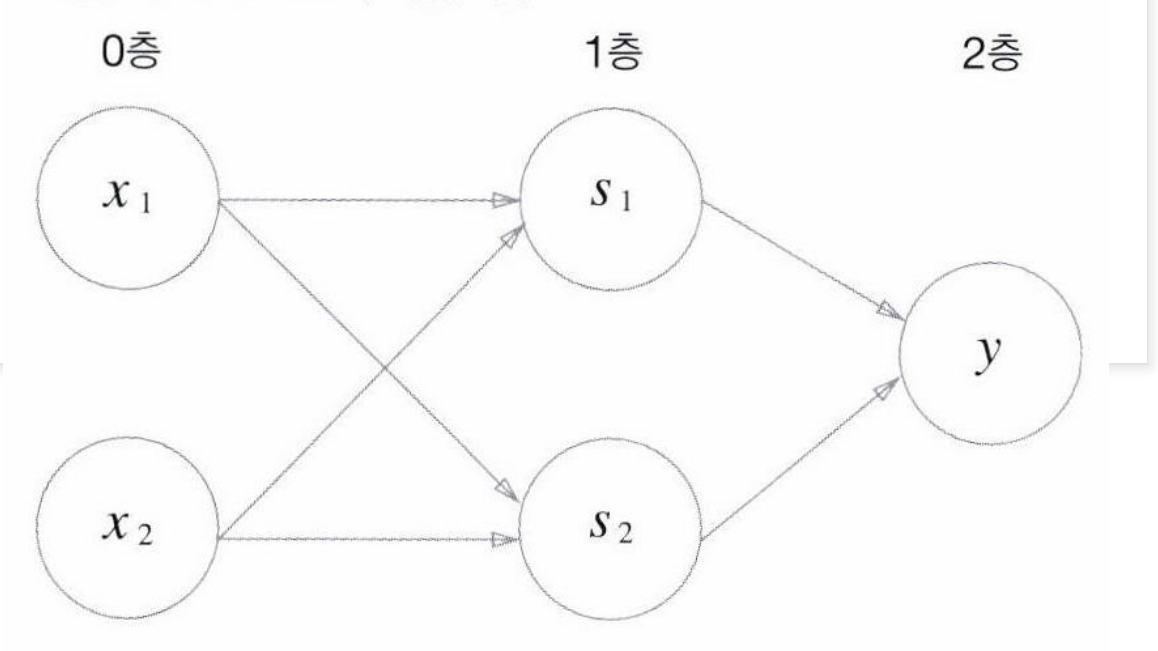


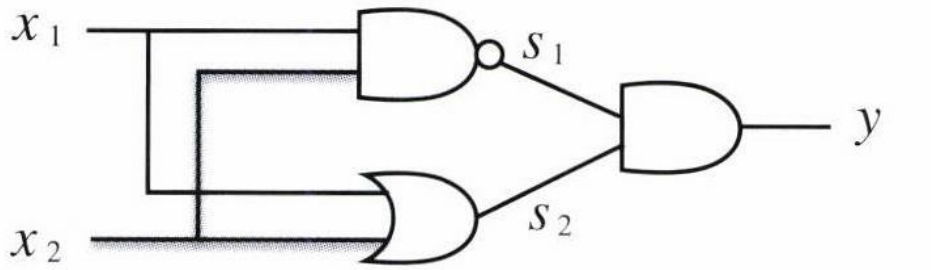

| $X_1$ | $X_2$ | $desire\_Y$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

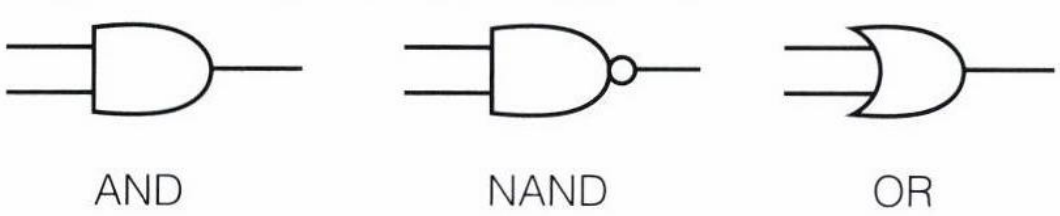
# 다층 퍼셉트론

- 다층 레이어를 통해 선형성 극복

| $x_1$ | $x_2$ | $s_1$ | $s_2$ | $y$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 |



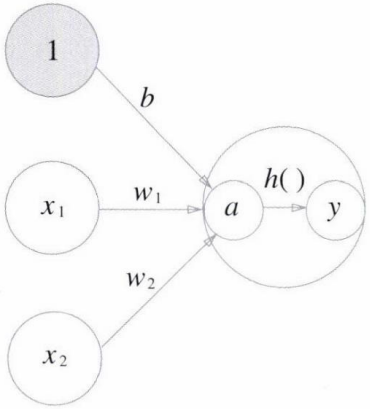



# 2장 퍼셉트론 정리

- 퍼셉트론은 입출력을 갖춘 알고리즘이다. 입력을 주면 정해진 규칙에 따른 값을 출력한다.
- 퍼셉트론에서는 '가중치'와 '편향'을 매개변수로 설정한다.
- 퍼셉트론으로 AND, OR 게이트 등의 논리 회로를 표현할 수 있다.
- XOR 게이트는 단층 퍼셉트론으로는 표현할 수 없다.
- 2층 퍼셉트론을 이용하면 XOR 게이트를 표현할 수 있다.
- 단층 퍼셉트론은 직선형 영역만 표현할 수 있고, 다층 퍼셉트론은 비선형 영역도 표현할 수 있다.
- 다층 퍼셉트론은 (이론상) 컴퓨터를 표현할 수 있다.

# chapter3. 신경망

# 활성화 함수

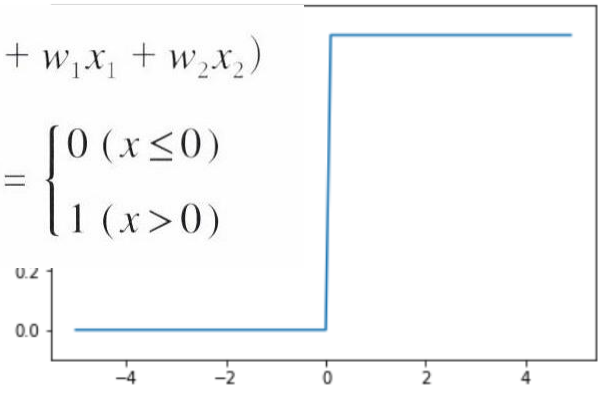



- 신호의 총합을 출력 신호로 변환하는 함수
- 신경망에서는 활성화 함수로 비선형 함수를 사용

1) 계단함수

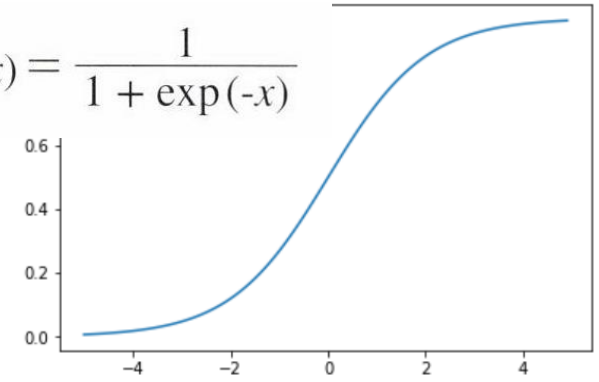
$$y = h(b + w_1x_1 + w_2x_2)$$

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



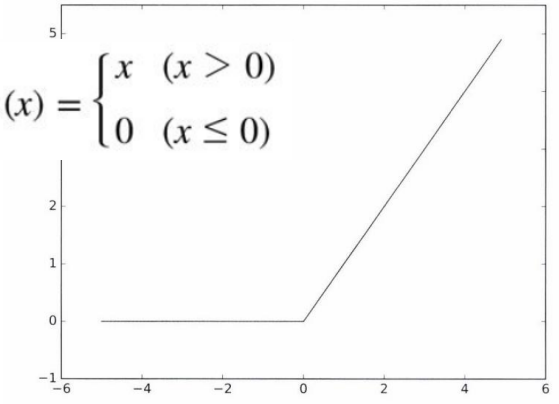

2) 시그모이드 함수

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




3) ReLU 함수

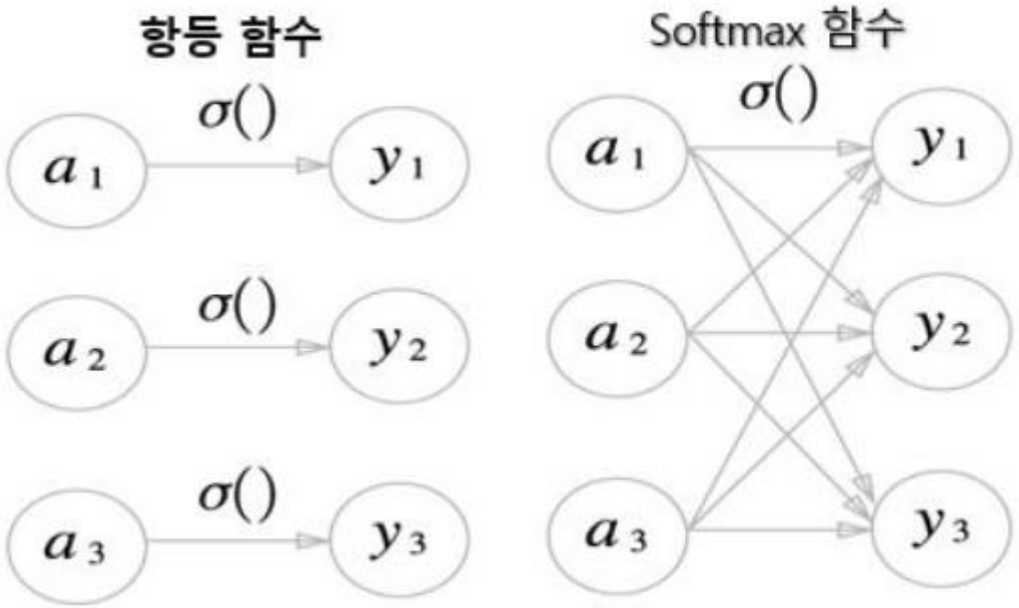
$$h(x) = \begin{cases} x & (x > 0) \\ 0 & (x \leq 0) \end{cases}$$




- 시그모이드 함수는 입력에 따라 출력이 연속적으로 변화 (연속적인 실수 출력)
- 계단함수는 0을 경계로 출력이 갑자기 변화 (0 혹은 1 출력)

# 3층 신경망 구현하기

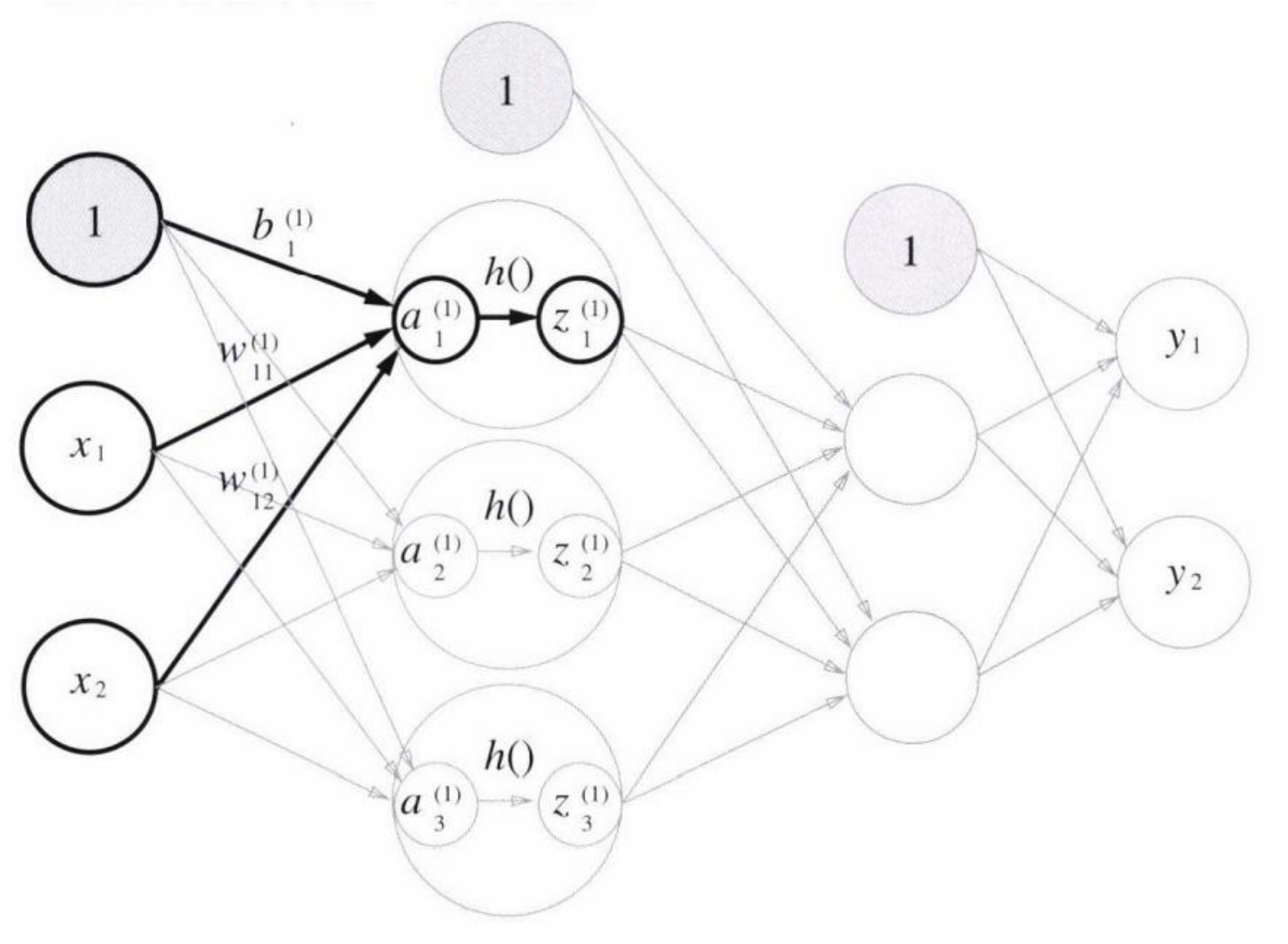
- 신경망에서 행렬 곱은 np.dot 함수를 사용해서 구현
- 출력층의 활성화 함수는 푸고자 하는 문제의 성질에 맞게 정의
  - 회귀에는 항등함수, 분류에는 소프트 맥스 함수를 사용하는 것이 일반적



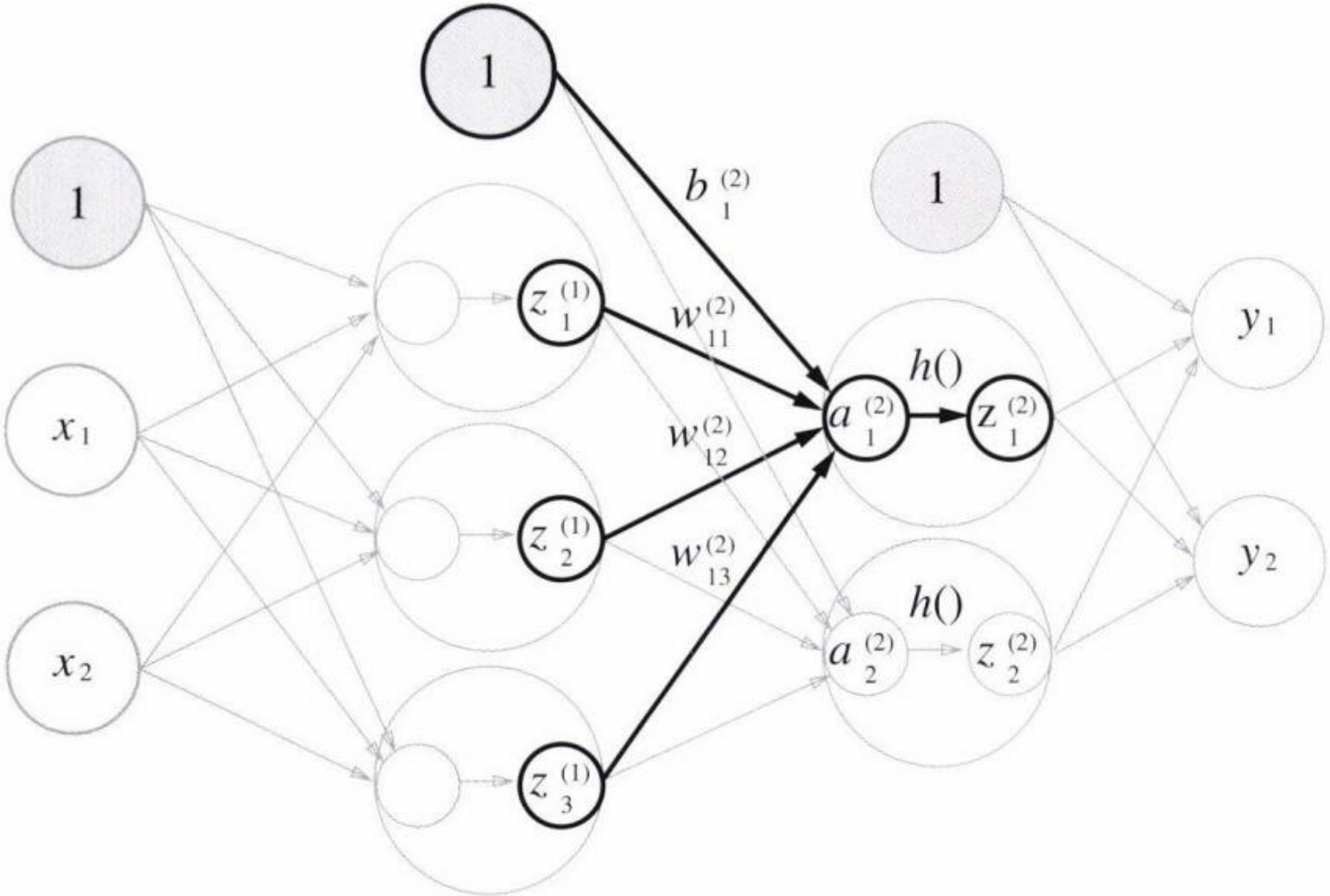

$$y_k = \frac{e^{a_k}}{\sum_{i=1}^{n} e^{a_i}} = \frac{Ce^{a_k}}{C\sum_{i=1}^{n} e^{a_i}} = \frac{e^{(a_k + \log C)}}{\sum_{i=1}^{n} e^{(a_i + \log C)}} = \frac{e^{(a_k + C')}}{\sum_{i=1}^{n} e^{(a_i + C')}}$$

- 오버플로우를 방지하기 위해 입력 신호 중 최댓값을 빼주는 것이 일반적
- 소프트맥스의 출력은 확률로 해석 가능

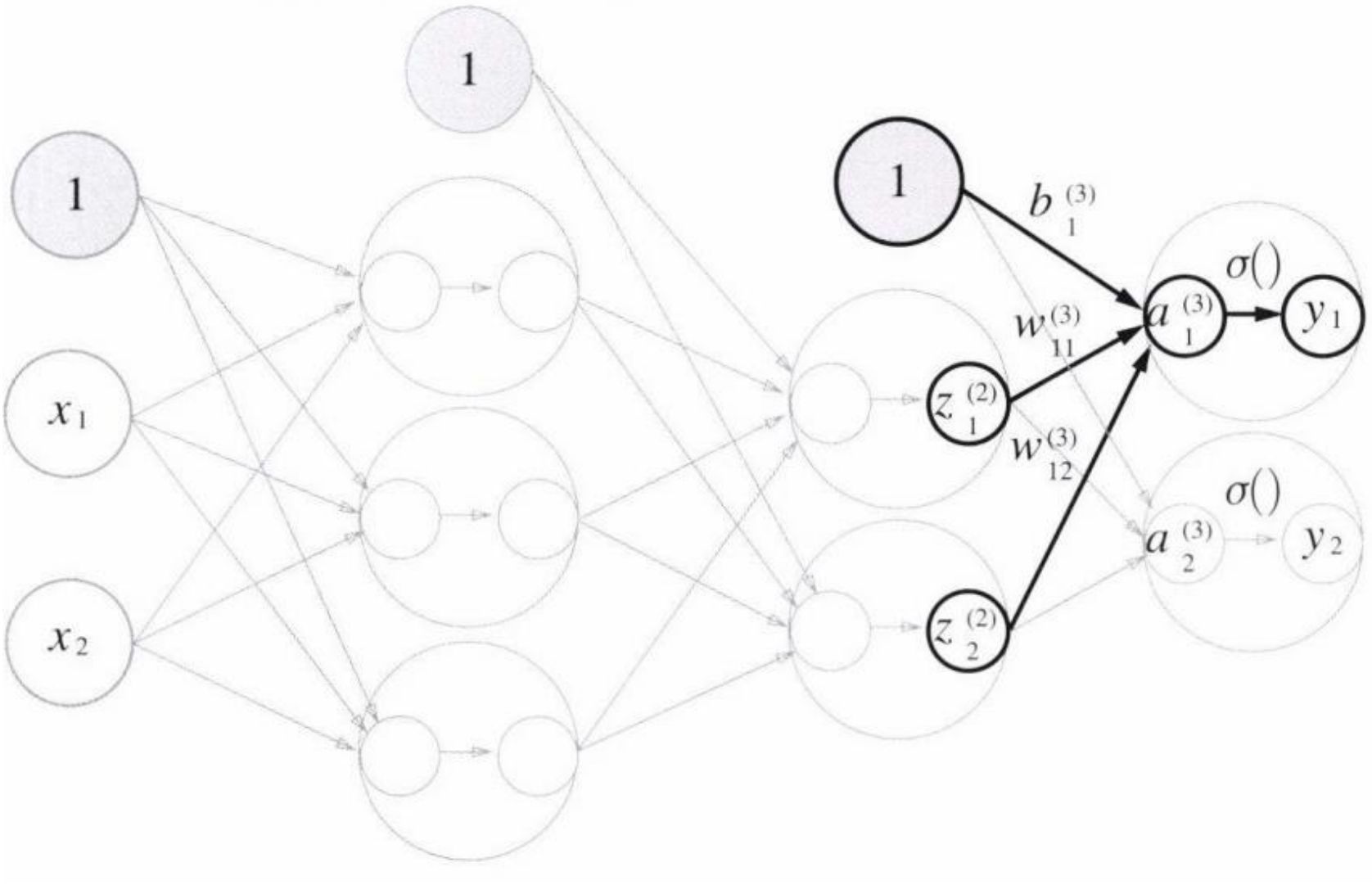
# 3층 신경망 구현하기 : 순전파(forward propagation)



# 3층 신경망 구현하기 : 순전파(forward propagation)



# 3층 신경망 구현하기 : 순전파(forward propagation)



# 3장 신경망 정리

- 신경망에서는 활성화 함수로 시그모이드 함수와 ReLU 함수 같은 매끄럽게 변화하는 함수를 이용한다.
- 넘파이의 다차원 배열을 잘 사용하면 신경망을 효율적으로 구현할 수 있다.
- 기계학습의 문제는 크게 회귀와 분류로 나눌 수 있다.
- 출력층의 활성화 함수로는 회귀에서는 주로 항등 함수를, 분류에서는 주로 소프트맥스 함수를 사용한다.
- 분류에서는 출력층의 뉴런 수를 분류하려는 클래스 수와 같게 설정한다.
- 입력 데이터를 묶은 것을 배치라 하며, 추론 처리를 이 배치 단위로 진행하면 결과를 훨씬 빠르게 얻을 수 있다.

# chapter 4. 신경망 학습

# 손실 함수

- Q. 왜 '정확도' 대신 '손실 함수'를 신경망 학습 기준 지표로 설정하는가 ?
- A. 정확도는 불연속적인 수치 & 미소한 변화에는 반응 X
  → 정확도를 지표로 하면 매개변수의 미분이 대부분의 장소에서 0이 되기 때문!
- 이는 신경망에서 대부분의 장소에서 미분값이 0이 되는 계단 함수를 활성화 함수로 사용하지 않고, 매끄러운 함수를 사용하는 것과 같은 이유

1) 오차제곱합 (sum of squares for error, SSE)

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

2) 교차 엔트로피 오차 (cross entropy error, CEE)

$$E = -\sum_{k} t_k log y_k$$

- 데이터가 N개 일 때

$$E = -\frac{1}{N}\sum_{n}\sum_{k} t_{nk} log y_{nk}$$

# 기울기를 이용한 신경망 학습 (경사법)

- 신경망의 손실 함수를 작게 만드는 기법으로 함수의 기울기를 활용
- 수치 미분 : 아주 작은 차분으로 미분하는 것
  - 오차역전파법을 이용하면 빠른 계산이 가능

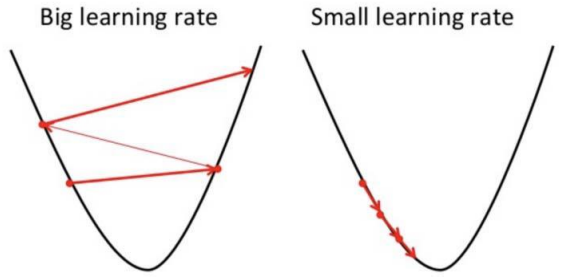
$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- 기울기가 가리키는 쪽은 각 장소에서 함수의 출력 값(= 손실 함수 값)을 가장 크게 줄이는 방향
- 신경망 학습에는 경사법을 사용

$$x_0 = x_0 - \eta \frac{\partial f}{\partial x_0}$$

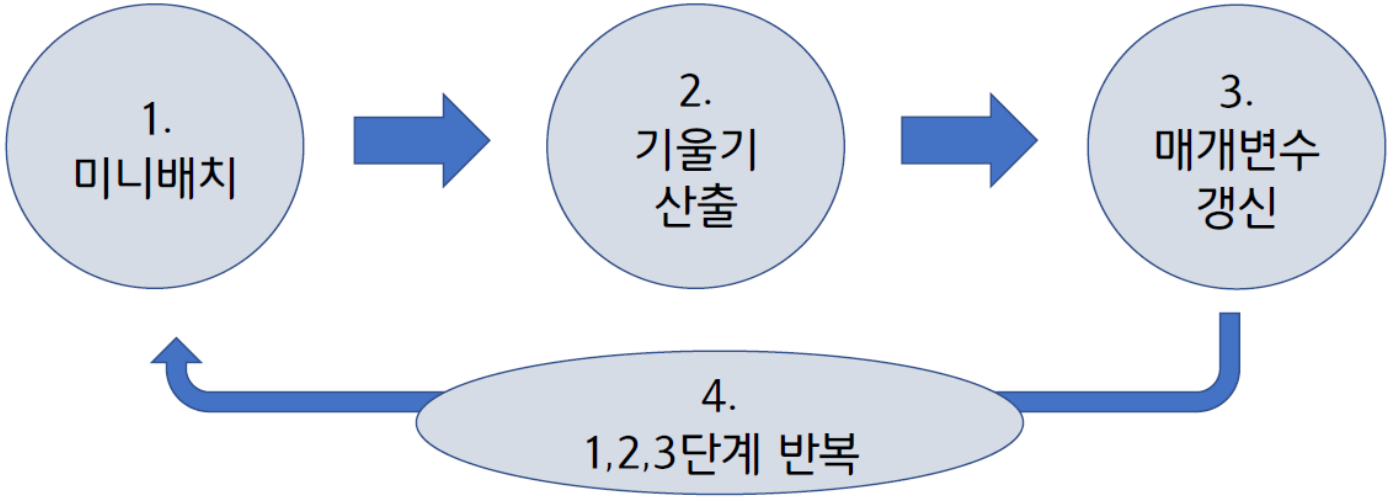
$$x_1 = x_1 - \eta \frac{\partial f}{\partial x_1}$$

learning rate



# 학습 알고리즘 구현하기

- 가중치와 편향을 훈련 데이터에 적응하도록 조정하는 과정 = 학습




- 데이터를 미니배치로 무작위 선정하기 때문에 확률적 경사하강법 (SGD) 이라고 부름.

# 4장 신경망 학습 정리

- 기계학습에서 사용하는 데이터셋은 훈련 데이터와 시험 데이터로 나눠 사용한다.
- 훈련 데이터로 학습한 모델의 범용 능력을 시험 데이터로 평가한다.
- 신경망 학습은 손실 함수 지표로, 손실 함수의 값이 작아지는 방향으로 가중치 매개변수를 갱신한다.
- 가중치 매개변수를 갱신할 때는 가중치 매개변수의 기울기를 이용하고, 기울어진 방향으로 가중치 값을 갱신하는 작업을 반복한다.
- 아주 작은 값을 주었을 때의 차분으로 미분하는 것을 수치 미분이라고 한다.
- 수치 미분을 이용해 가중치의 매개변수의 기울기를 구할 수 있다.
- 수치 미분을 이용한 계산에는 시간이 걸리지만 구현은 간단한다.

# chapter 5. 오차역전파법

수치 미분은 단순하고 구현하기 쉽지만 계산 시간이 오래 걸린다는 단점 有

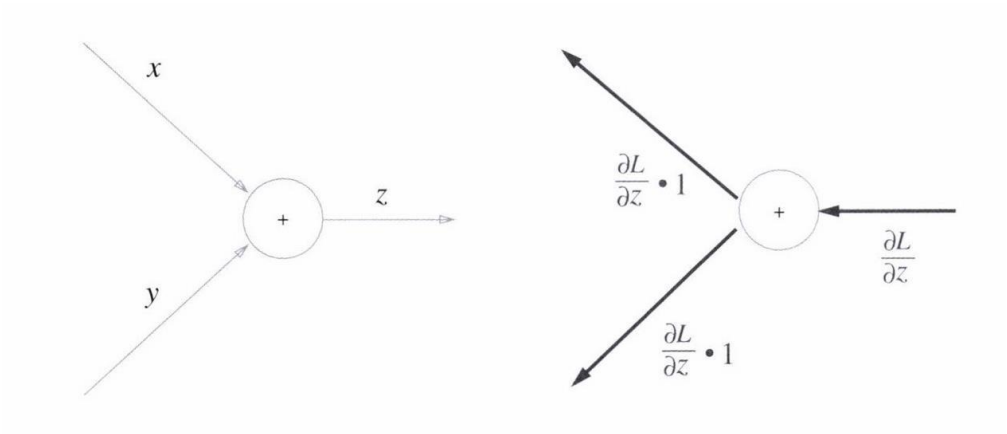
→가중치의 매개변수의 기울기를 효율적으로 계산하는 '오차역전파법' 을 배워봅시다!

# 계산 그래프의 역전파

- 계산 그래프 = 계산 과정을 그래프(노드, 엣지)로 표현한 것

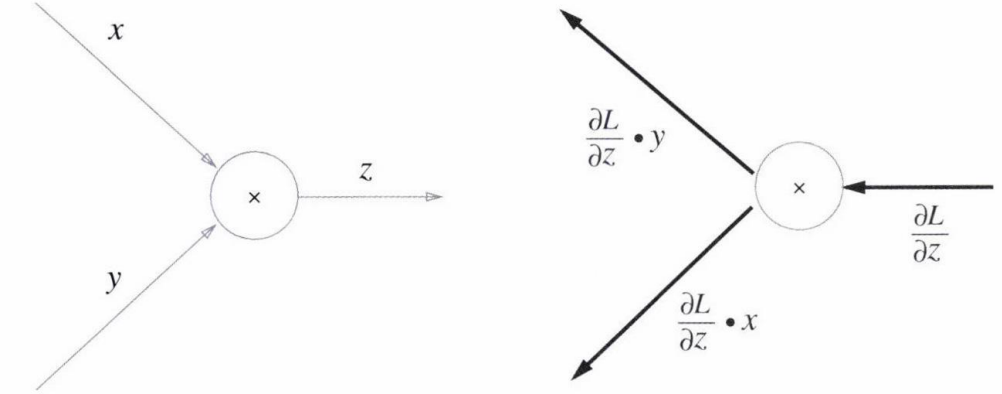
- 덧셈 노드의 역전파

:입력 값을 그대로 다음노드로 흘려 보냄




## 2) 곱셈 노드의 역전파

: 상류의 값에 순전파 때의 입력 신호들을 서로 바꾼 값을 곱해 하류로 보냄



# 활성화 함수 계층 구현하기

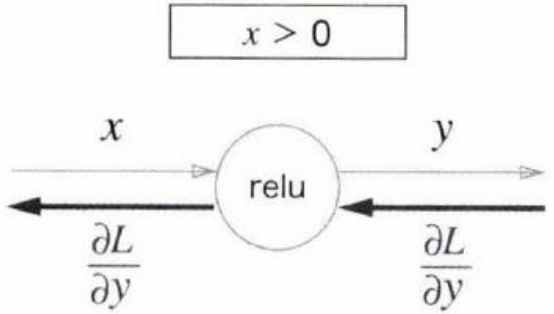
- ReLU 계층

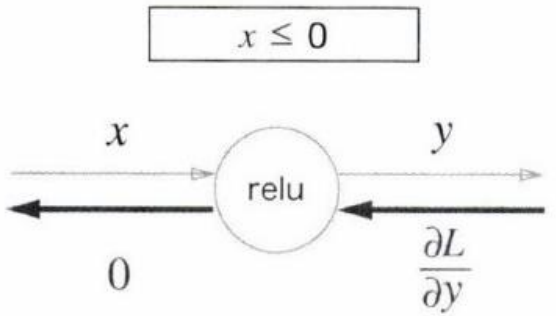
$$y = \begin{cases} x & (x > 0) \\ 0 & (x \le 0) \end{cases}$$

역전파

$$\frac{\partial y}{\partial x} = \begin{cases} 1 & (x > 0) \\ 0 & (x \le 0) \end{cases}$$

그림 5-18 ReLU 계층의 계산 그래프



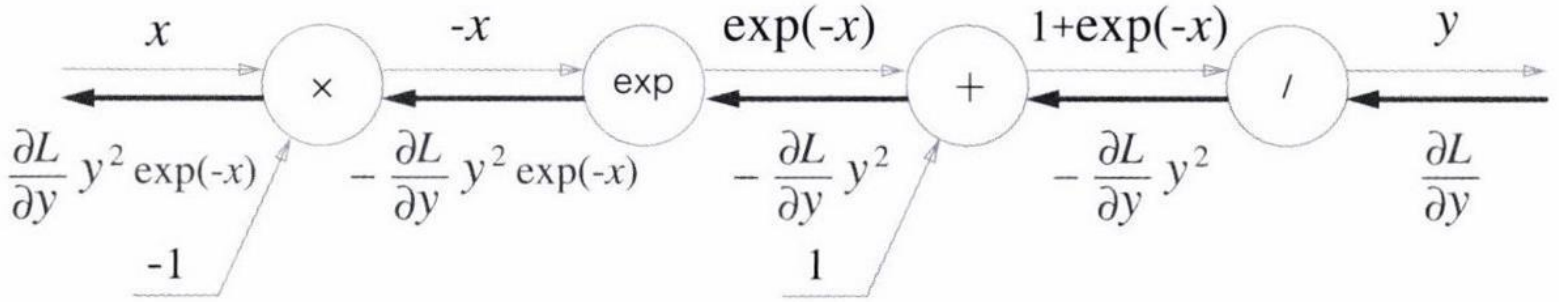



# 활성화 함수 계층 구현하기

- Sigmoid 계층

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

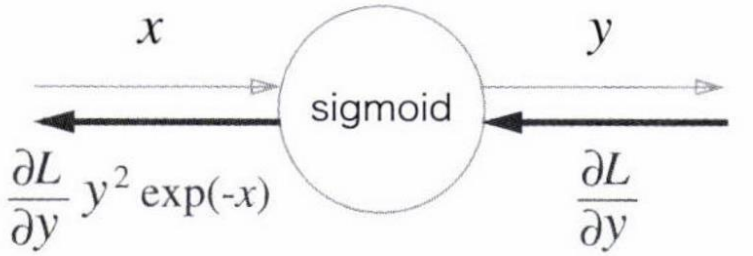
그림 5-20 Sigmoid 계층의 계산 그래프

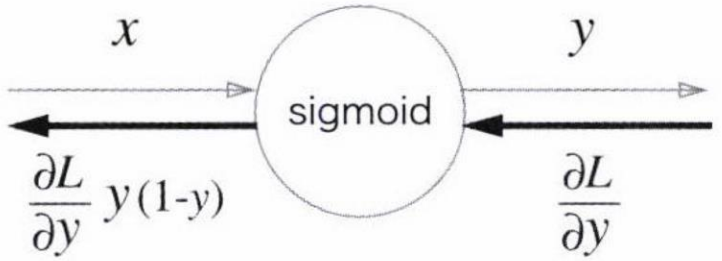



$$\frac{\partial L}{\partial y} y^2 \exp(-x) = \frac{\partial L}{\partial y} \frac{1}{(1 + \exp(-x))^2} \exp(-x)$$

$$= \frac{\partial L}{\partial y} \frac{1}{1 + \exp(-x)} \frac{\exp(-x)}{1 + \exp(-x)}$$

$$= \frac{\partial L}{\partial y} y(1 - y)$$




- Sigmoid 계층의 역전파를 순전파의 출력 y만으로 계산할 수 있음

# Affine/Softmax 계층 구현하기

- 신경망의 순전파 때 수행하는 행렬의 곱 (Affine 변환)을 처리하는 계층

## 1) Affine 계층의 역전파

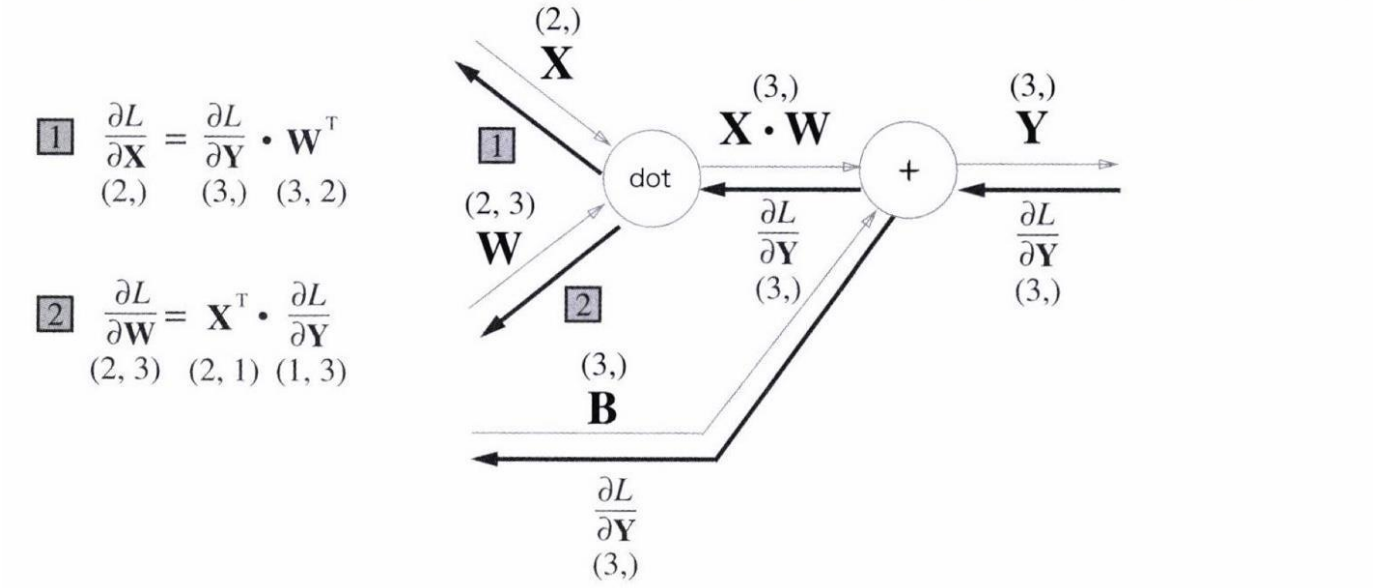
a, b: 상수 벡터

A: 상수 행렬

y, z: x와 함수관계를 갖는 벡터

| 식 | x 로 미분 결과 |
|---|---|
| $\mathbf{a}\cdot\mathbf{x}=\mathbf{a}^T\mathbf{x}=\mathbf{x}^T\mathbf{a}$ | $\mathbf{a}^T$ |
| $A\mathbf{x}$ | $A$ |
| $\mathbf{x}^TA$ | $A^T$ |

**그림 5-25** Affine 계층의 역전파 : 변수가 다차원 배열임에 주의. 역전파에서의 변수 형상은 해당 변수명 아래에 표기했다.



# Affine/Softmax 계층 구현하기

## 2) softmax-with-Loss 계층

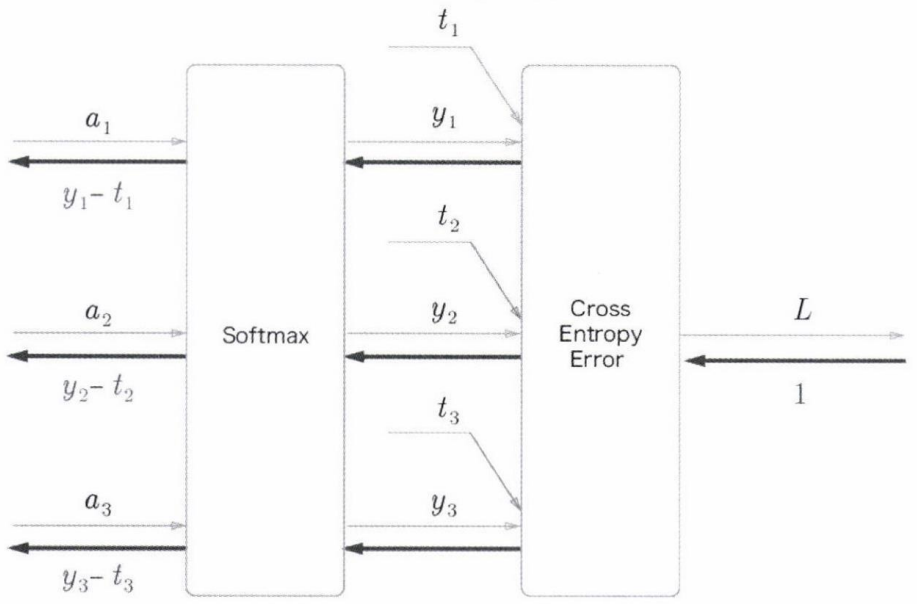
• 손실 함수 '교차 엔트로피 오차 '도 포함하여

'Softmax -with with-Loss' 계층으로 계층으로 구현

- 3클래스 분류를 가정
- 역전파의 결과 (y1-t1, y2 -t2, y3-t3)

= softmax 계층의 출력과 정답 레이블의 차분

- 신경망의 역전파에서는 오차가 앞 계층에 전해짐



# 오차역전파법 구현하기

- 앞에서 구현한 계층을 조합하여 신경망을 구축

- 기울기 확인 : 수치 미분으로 오차역전파법을 정확히 구현했는지 확인
  - 수치 미분과 오차역전파법으로 구한 결과 차이가 매우 작으면 잘 구현한 것.

# 5장 오차역전파법 정리

- 계산 그래프를 이용하면 계산 과정을 시각적으로 파악할 수 있다.
- 계산 그래프의 노드는 국소적 계산으로 구성된다. 국소적 계산을 조합해 전체 계산을 구성한다.
- 계산 그래프의 순전파는 통상의 계산을 수행한다.

  한편, 계산 그래프의 역전파로는 각 노드의 미분을 구할 수 있다.
- 신경망의 구성 요소를 계층으로 구현하여 기울기를 효율적으로 계산할 수 있다. (= 오차역전파법)
- 수치 미분과 오차역전파법의 결과를 비교하면 오차역전파법의 구현에 잘못이 없는지 확인할 수 있다. (= 기울기 확인)