

Machine Learning

Chapter 3

신경망

3. 신경망

3.1 퍼셉트론에서 신경망으로

3.2 활성화 함수

3.3 다차원 배열의 계산

3.4 3층 신경망 구현하기

3.5 출력층 설계하기

3.6 손글씨 숫자 인식

3.7 정리

3.1 퍼셉트론에서 신경망으로

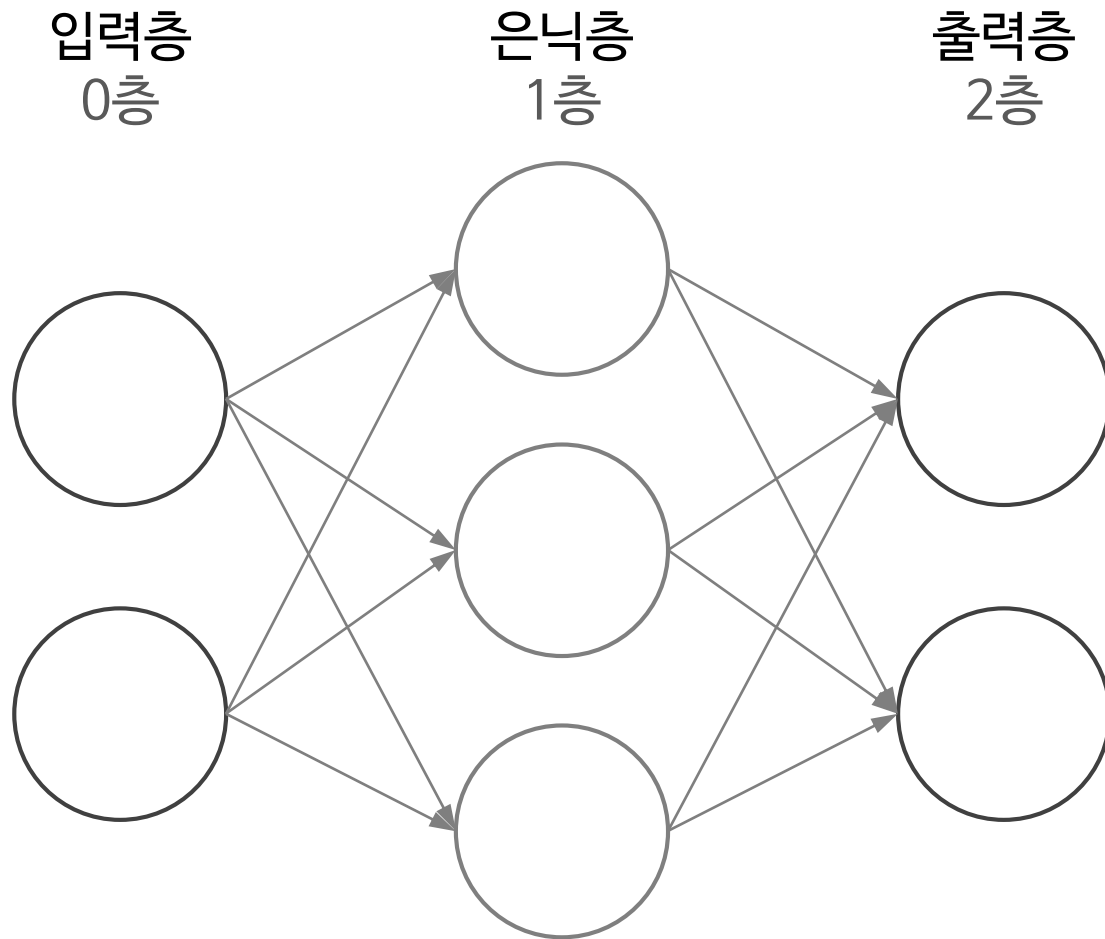
■ 퍼셉트론

- 복잡한 함수 표현 가능
- 가중치 설정은 사람이 수동으로 해야함

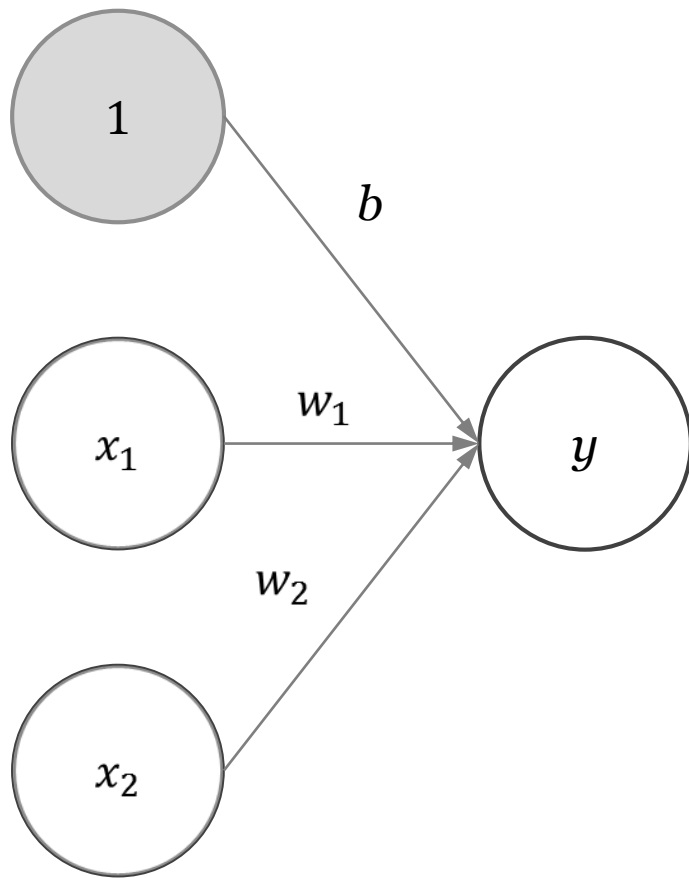
■ 신경망

- 가중치 매개변수의 적절한 값을 데이터로부터 자동 학습

3.1 퍼셉트론에서 신경망으로



3.1 퍼셉트론에서 신경망으로



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

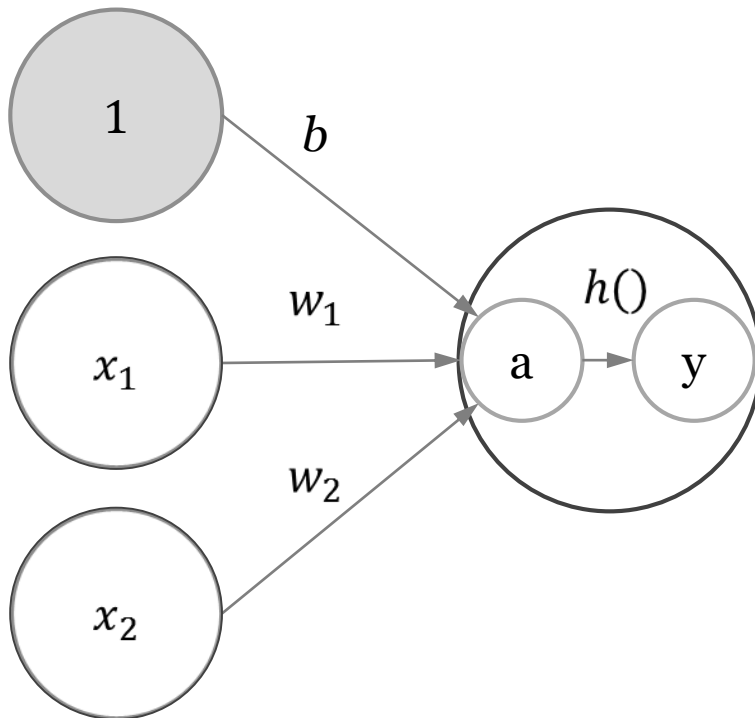
$$y = h(b + w_1x_1 + w_2x_2)$$

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

3.1 퍼셉트론에서 신경망으로

■ 활성화 함수의 등장

- 입력 신호의 총합을 출력 신호로 변환하는 함수

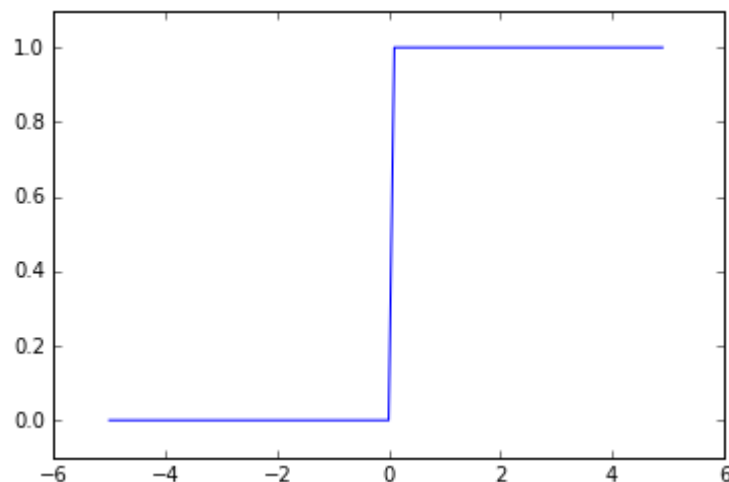


$$a = b + w_1x_1 + w_2x_2$$
$$y = h(a)$$

3.2 활성화 함수

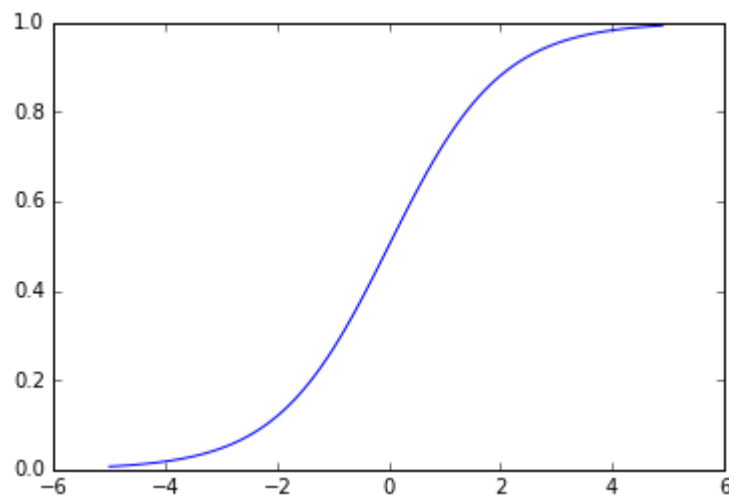
- 계단 함수

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



- 시그모이드 함수

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

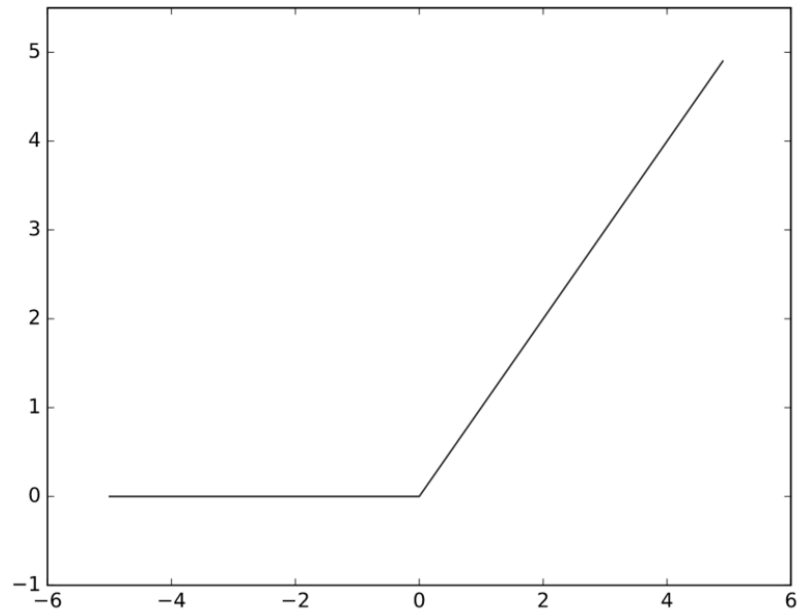


3.2 활성화 함수

■ ReLU 함수

- 시그모이드 함수는 신경망 분야에서 오래전부터 이용해왔으나, 최근에는 ReLU Rectified Linear Unit, 렐루 함수를 주로 이용

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



3.3 다차원 배열의 계산

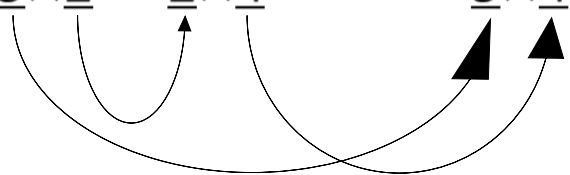
- 행렬의 내적 (행렬곱)

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}$$

A **B**

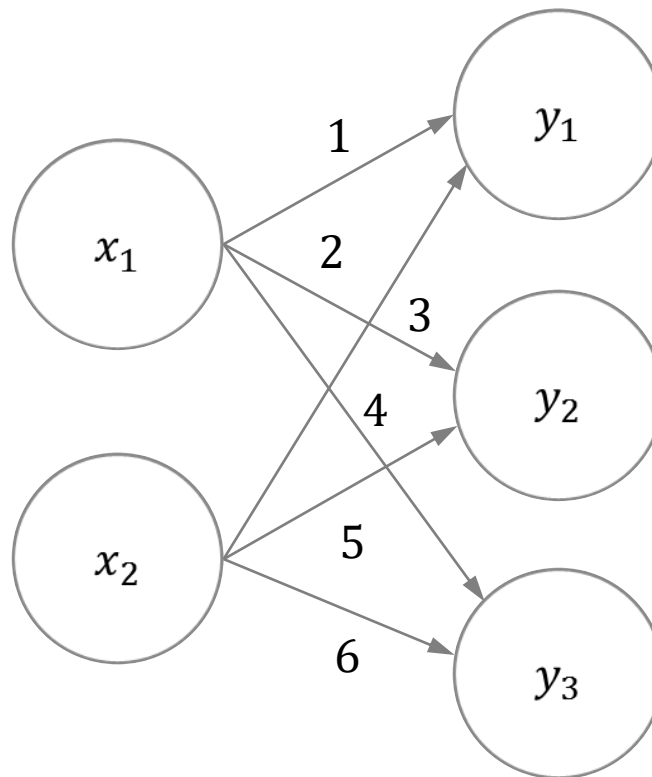
-행렬의 곱에서는 대응하는 차원의 원소 수를 일치시켜야 함.

EX) $\underline{3 \times 2} \quad \underline{2 \times 4} = \underline{3 \times 4}$



3.3 다차원 배열의 계산

■ 신경망의 내적

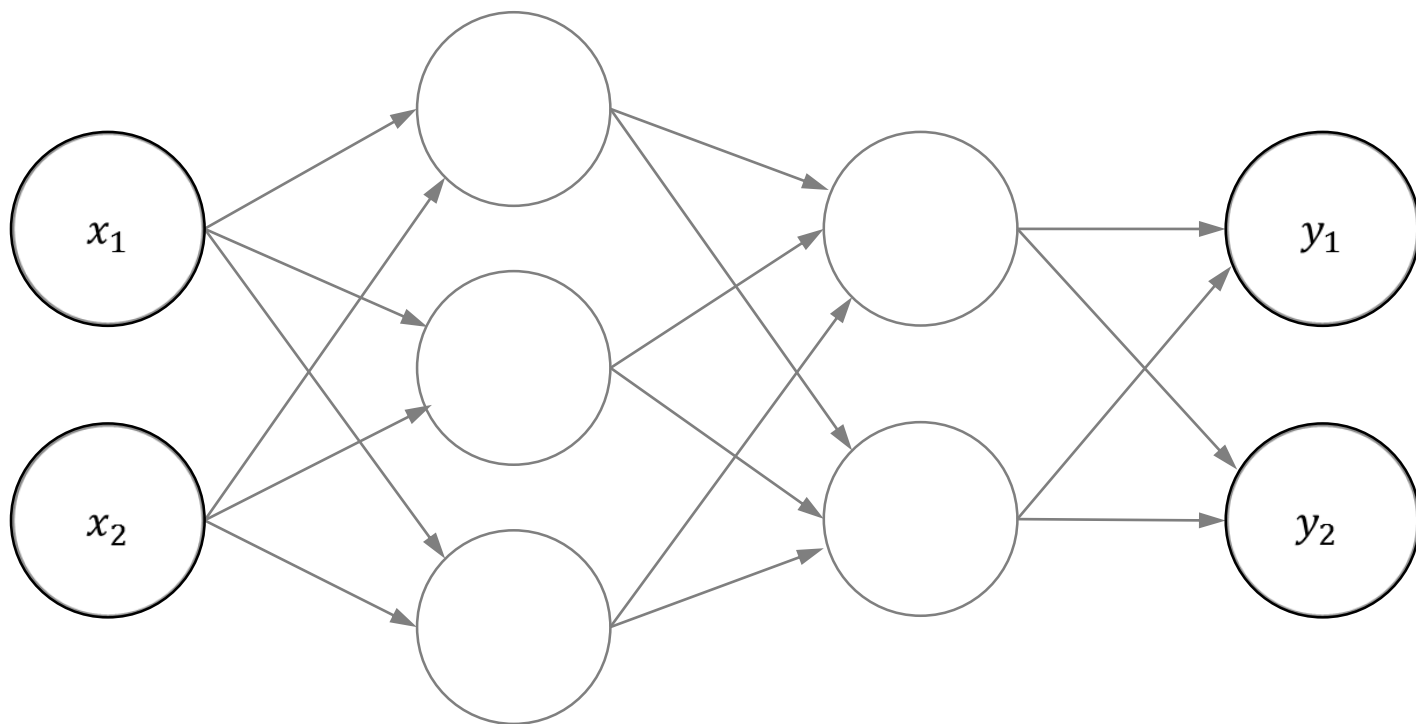


$$2 \quad 2 \times 3 = 3$$

3.4 3층 신경망 구현하기

■ 3층 신경망

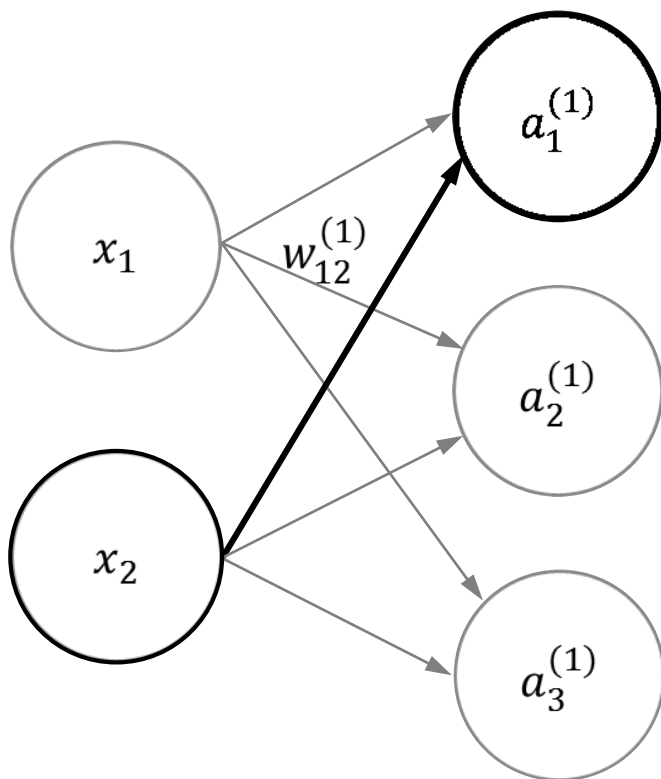
- 입력층 2개, 첫 번째 은닉층 3개, 두 번째 은닉층 2개, 출력층 2개의 뉴런으로 구성



3.4 3층 신경망 구현하기

- 중요한 표기

중요한 표기



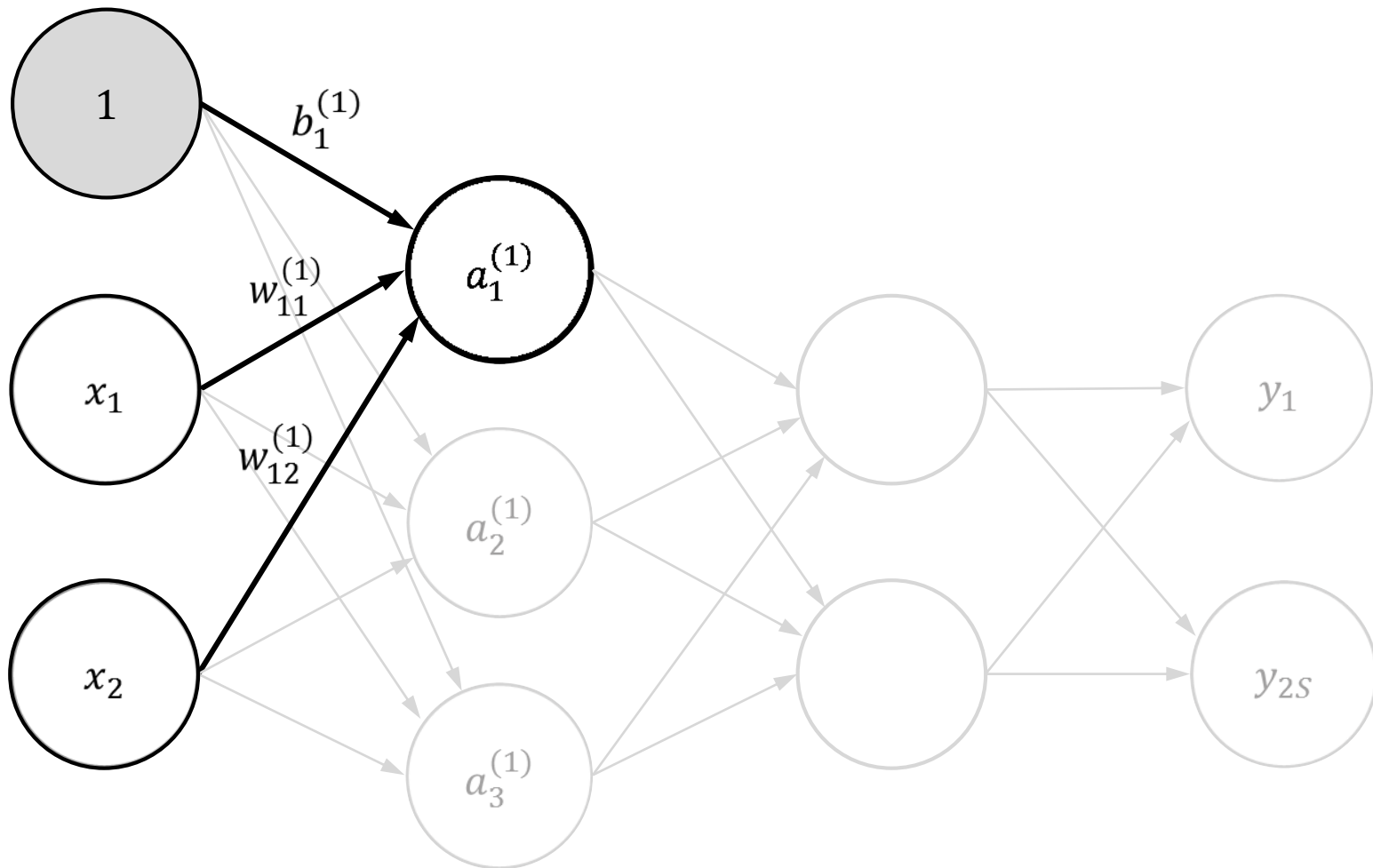
1층의 가중치

$w_{12}^{(1)}$

앞 층의 2번째 뉴런

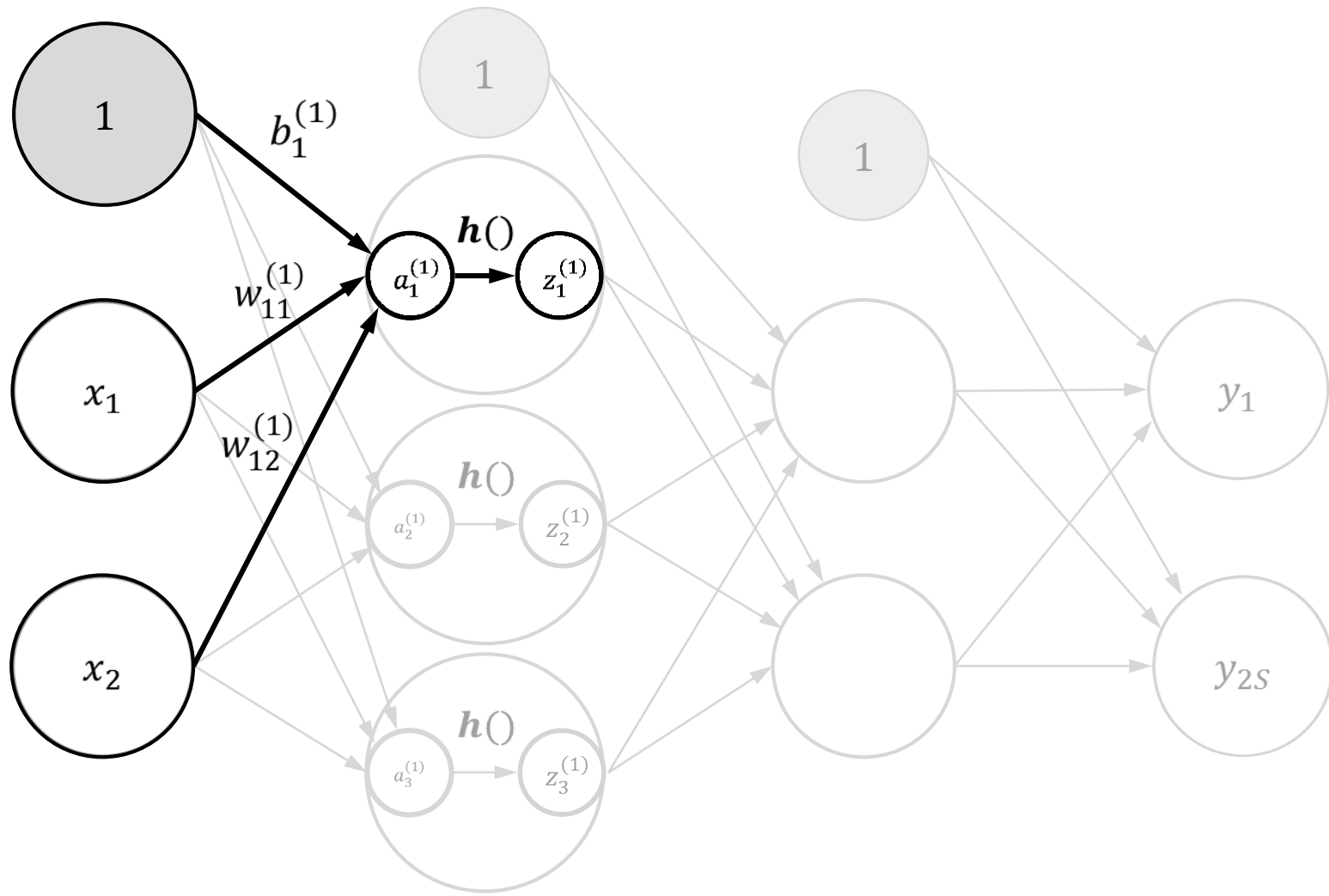
다음 층의 1번째 뉴런

3.4 3층 신경망 구현하기



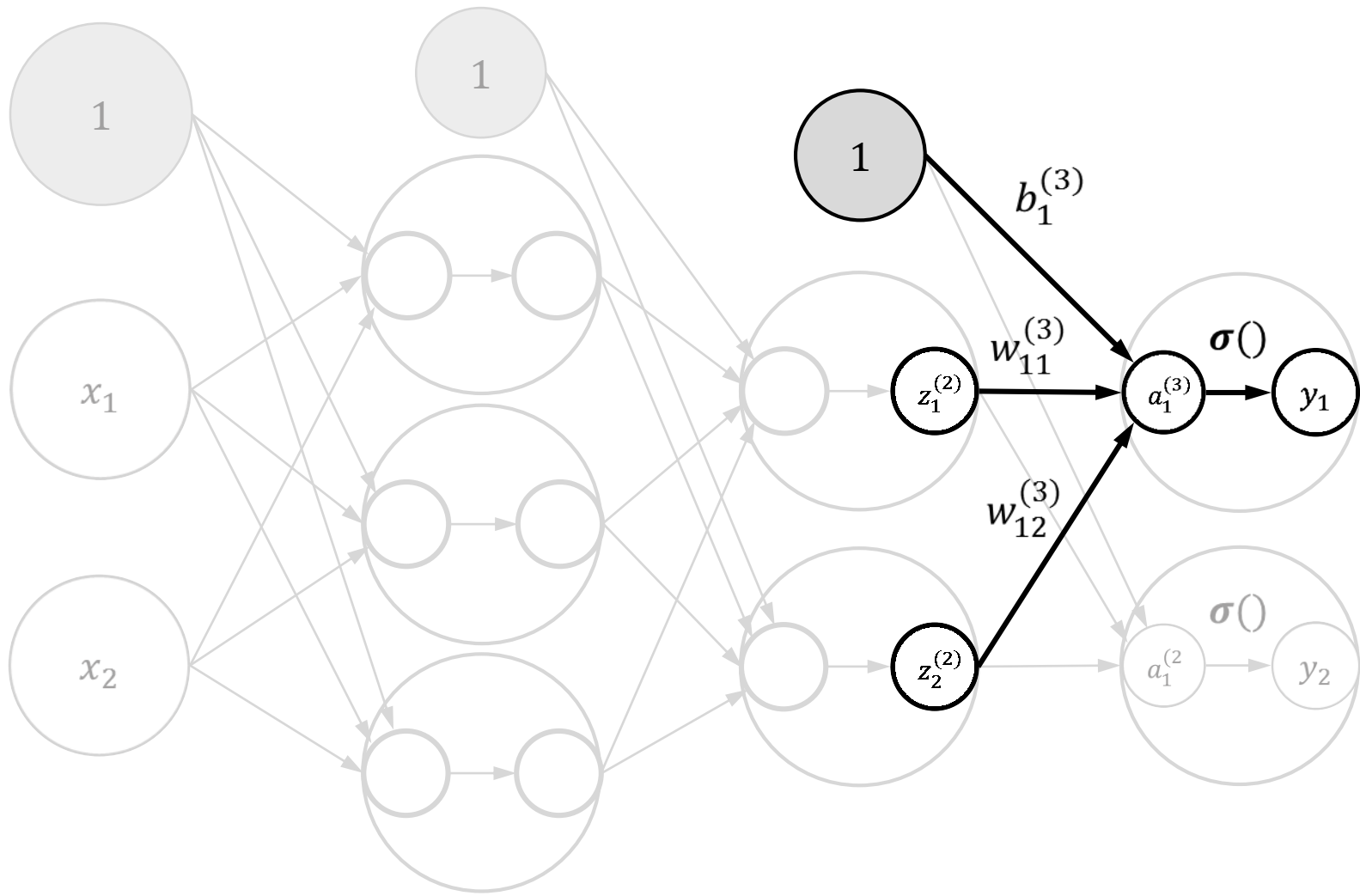
입력층에서 1층으로의 신호 전달

3.4 3층 신경망 구현하기



입력층에서 1층으로의 신호 전달

3.4 3층 신경망 구현하기



2층에서 출력층으로의 신호 전달

3.5 출력층 설계하기

■ 회귀 regression - 항등 함수 사용

- 입력 데이터에서 연속적인 수치를 예측
ex) 사진 속 인물의 몸무게를 예측하는 문제

■ 분류 classification - 소프트맥스 함수 사용

- 데이터가 어느 클래스에 속하는지
ex) 사진 속 인물의 성별을 분류하는 문제

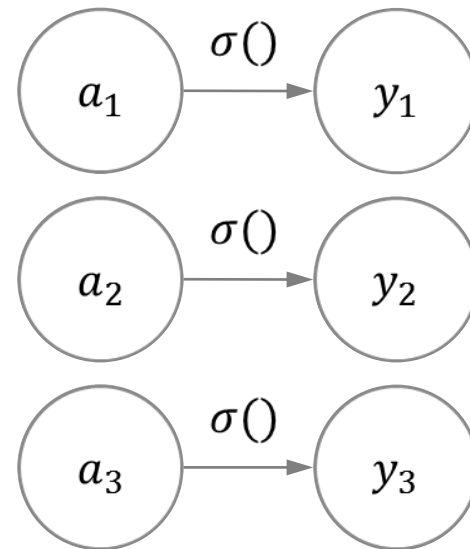
출력층의 활성화 함수는 풀고자 하는 문제의 성질에 맞게 정한다.

회귀에는 항등함수를, 2클래스 분류에는 시그모이드 함수를, 다중 클래스 분류에는 소프트맥스 함수를 사용하는 것이 일반적이다.

3.5 출력층 설계하기

■ 항등 함수 identity function

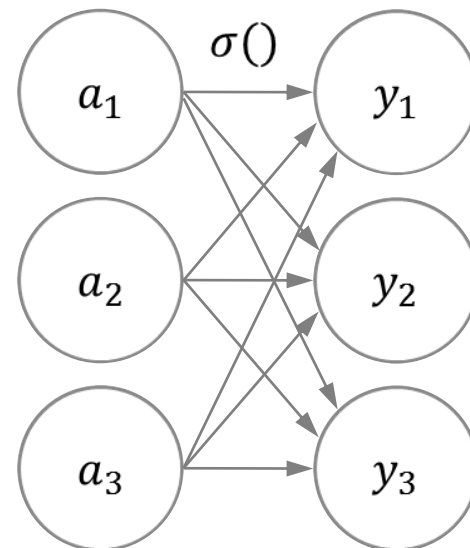
- 입력을 그대로 출력



■ 소프트맥스 함수 softmax function

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

- n = 출력층 뉴런 수
- y_k = k 번째 출력



3.5 출력층 설계하기

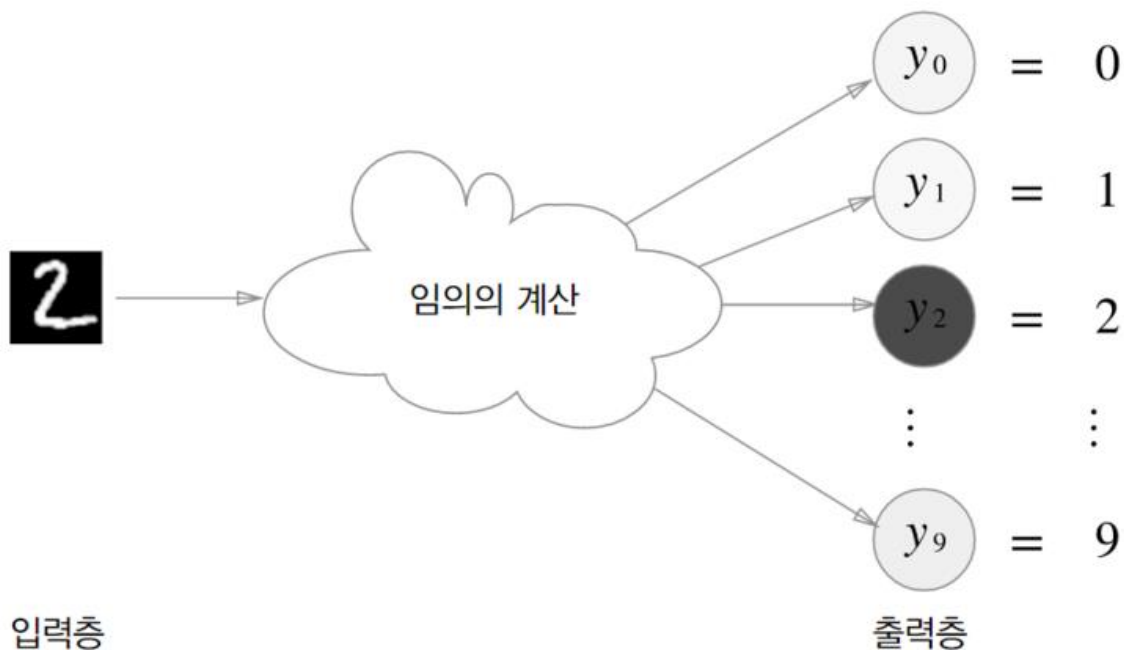
■ 소프트맥스 함수의 특징

- 출력은 0에서 1.0 사이의 실수, 출력의 총합은 1
 - 소프트맥스 함수의 출력을 '확률'로 해석할 수 있다.
- 함수를 적용해도 각 원소의 대소 관계는 변하지 않는다.
 - $y = \exp(x)$ 가 단조 증가 함수이기 때문 ($a \leq b$ 일 때 $f(a) \leq f(b)$)
- 신경망을 이용한 분류에서는 일반적으로 가장 큰 출력을 내는 뉴런에 해당하는 클래스로만 인식한다.
 - 추론 단계에서는 출력층의 소프트맥스 함수를 생략하는 것이 일반적 (지수 함수 계산에 드는 자원 낭비)
 - 학습 단계에서는 출력층에서 소프트맥스 함수를 사용

3.5 출력층 설계하기

■ 출력층의 뉴런 수 정하기

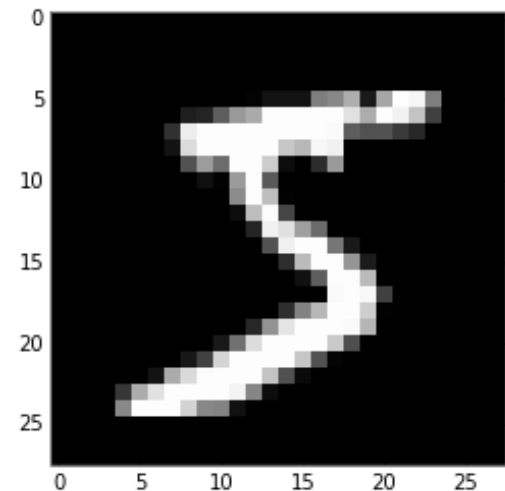
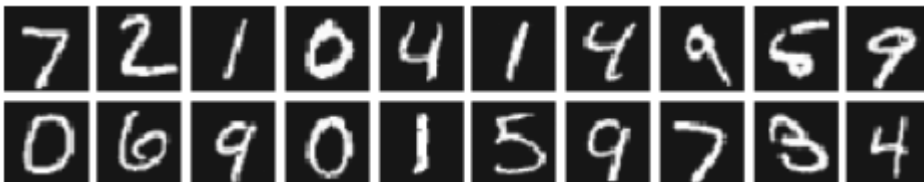
- 출력층의 뉴런 수는 풀려는 문제에 맞게 적절히 정한다.
- 분류에서는 분류하고 싶은 클래스 수로 설정하는 것이 일반적



3.6 손글씨 숫자 인식(MNIST)

■ MNIST 데이터셋

- mnist?
 - 손글씨 숫자 이미지 집합
 - 0 ~ 9까지의 숫자 이미지로 구성
 - 훈련 이미지 60,000장(학습), 시험 이미지 10,000장(분류)

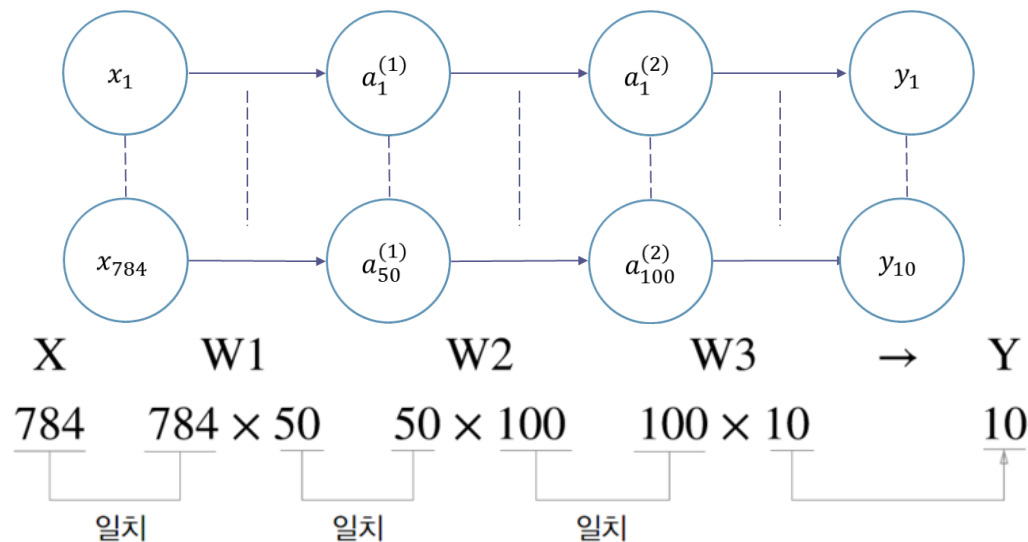


3.6 손글씨 숫자 인식(MNIST)

■ MNIST 데이터셋

- 신경망 구성

- 입력층 뉴런 784개(28×28), 출력층 뉴런 10개(0~9)
- 은닉층 2개(첫 번째 - 50개 뉴런, 두 번째 - 100개 뉴런)
- 은닉층 뉴런의 개수는 임의지정 값

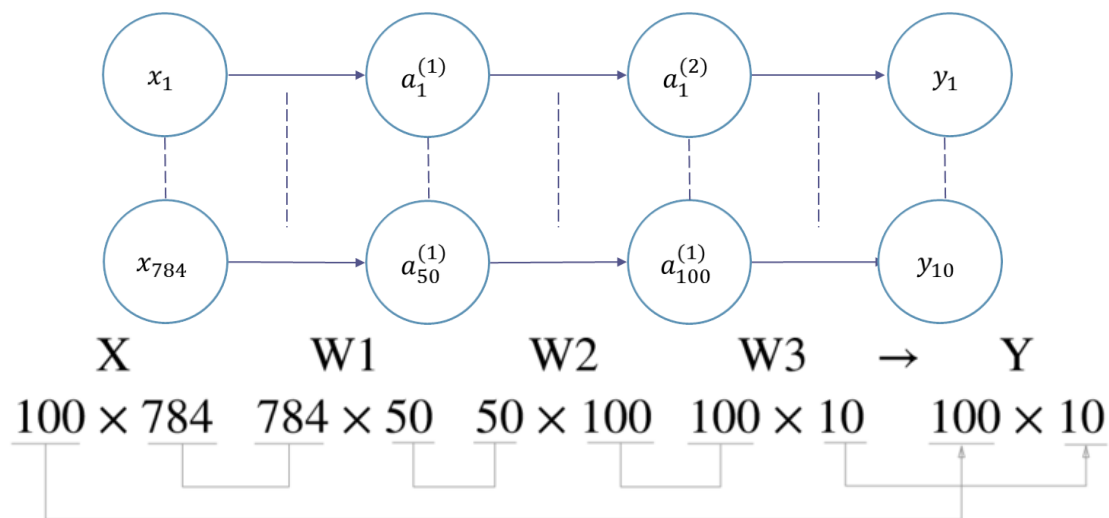


3.6 손글씨 숫자 인식(MNIST)

■ MNIST 데이터셋

- 신경망 구성

- 입력층 뉴런 784개(28×28), 출력층 뉴런 10개(0~9)
- 은닉층 2개(첫 번째 - 50개 뉴런, 두 번째 - 100개 뉴런)
- 은닉층 뉴런의 개수는 임의지정 값



3.7 정리

■ 신경망의 순전파와 활성화 함수

이번 장에서 배운 내용

- 신경망에서는 활성화 함수로 시그모이드 함수와 ReLU 함수 같은 매끄럽게 변화하는 함수를 이용한다.
- 넘파이의 다차원 배열을 잘 사용하면 신경망을 효율적으로 구현할 수 있다.
- 기계학습 문제는 크게 회귀와 분류로 나눌 수 있다.
- 출력층의 활성화 함수로는 회귀에서는 주로 항등 함수를, 분류에서는 주로 소프트맥스 함수를 이용한다
- 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.