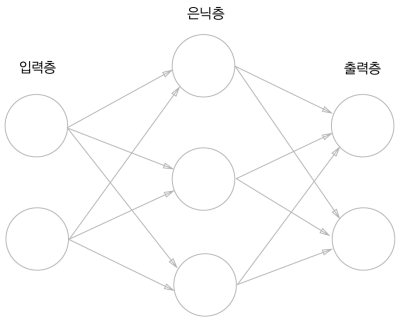
CHAPTER3 신경망

- 퍼셉트론에서 신경망으로
- 활성화 함수
- 신경망 구현하기
- 출력층 설계하기
- 정리

3.1 퍼셉트론에서 신경망으로

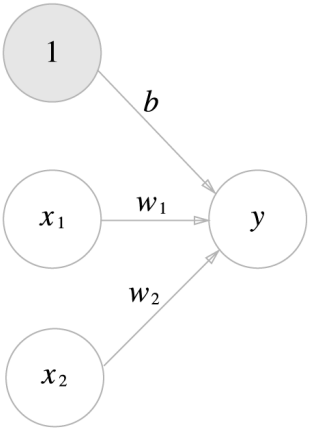
- 퍼셉트론은 가중치를 설정하는 작업을 사람이 수동해야 하는 한계점이 있습니다.
- 신경망은 이러한 퍼셉트론의 한계점을 해결해줍니다.

3.1.1 신경망의 예 ¶



- 신경망은 크게 입력층, 은닉층, 출력층으로 나뉘어 있습니다.
- 은닉층의 뉴런은 사람 눈에는 보이지 않습니다.

3.1.2 퍼셉트론 활용



- b(bias)를 활용한 수식

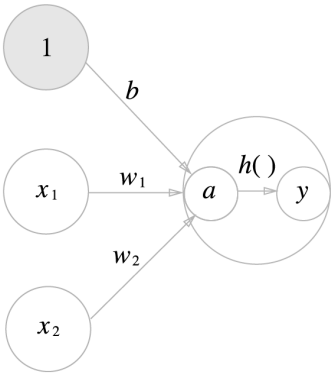
$$y = \begin{cases} 0 & (b + w_1x_1 + w_2x_2 \leq 0) \\ 1 & (b + w_1x_1 + w_2x_2 > 0) \end{cases}$$

- 퍼셉트론의 θ를 b로 변환해 하나의 뉴런으로 생성합니다.
- 생성된 뉴런을 식에 적용합니다.

3.2 활성화 함수

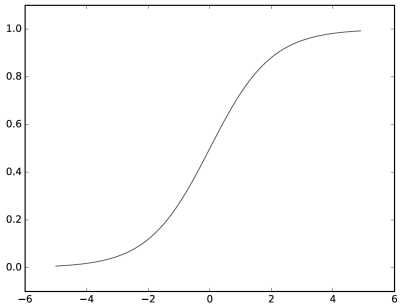
$$y = h(b + w_1x_1 + w_2x_2)$$

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



- $h(x)$와 같이 입력신호의 총합을 출력신호로 변환해주는 함수를 의미합니다.
- $h(x)$의 경우 입력이 0을 넘으면 1을 돌려주고 그렇지 않으면 0을 돌려줍니다.
- 활성화 함수는 비선형 함수를 사용해야 합니다.
- 비선형 함수란 말그대로 선형 함수가 아닌 함수를 말합니다. (Not 직선)
- 선형함수는 층을 깊게해도 선형에만 머물러있기 때문입니다. ($h(x) = cx$, $h(h(h(x))) = c^3x$)

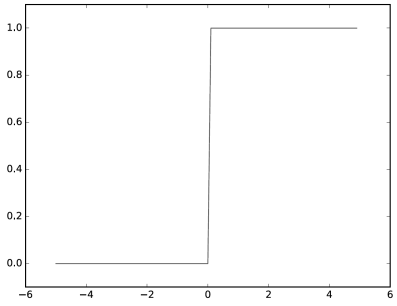
3.2.1 시그모이드 함수 (Sigmoid Function)



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

- exp는 자연상수로 2.7182..의 값을 갖는 실수 입니다.
- y값이 0~1사이의 값을 가지며 S자 모양을 하고 있어 이름이 붙여졌습니다.

3.2.2 계단 함수 (Step Function)



- 임계값을 경계로 출력이 바뀌는 함수를 계단함수라 합니다.
- 퍼셉트론에서는 활성화 함수로 계딴함수를 이용한다고 할 수 있습니다.

※ 시그모이드 함수 vs 계단 함수

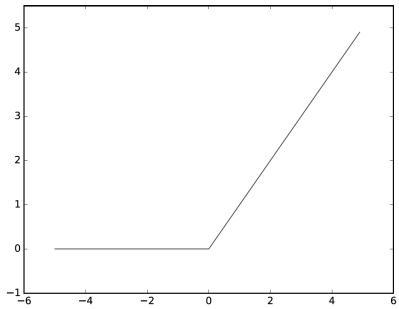
1) 공통점

- 입력이 작을 때 출력은 0에 가깝고 입력이 커지면 출력이 1에 가까워집니다.
- 입력값은 0에서 1사이입니다.
- 비선형 함수입니다.

2) 차이점

- 시그모이드가 더 매끄럽습니다. (신경망에서 매우 중요한 역할)
- 시그모이드에선 연속적인 실수가 흐릅니다. (계단은 0, 1)

3.2.3 ReLU 함수 (Rectified Linear Unit Function)

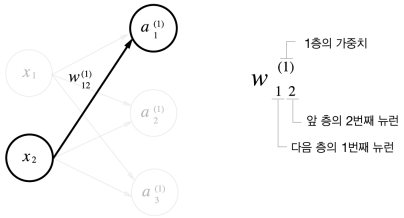


$$h(x) = \begin{cases} 0 & (x \leq 0) \\ x & (x > 0) \end{cases}$$

- 입력이 0을 넘으면 그 입력을 그대로 출력하고, 0이하면 0을 출력하는 함수입니다.
- 최근 신경망 분야에서 시그모이드 함수보다 ReLU 함수를 많이 사용합니다.

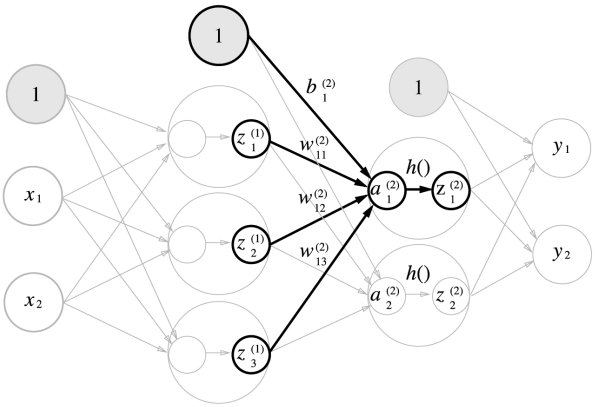
3.3 신경망 구현하기

3.3.1 표기법

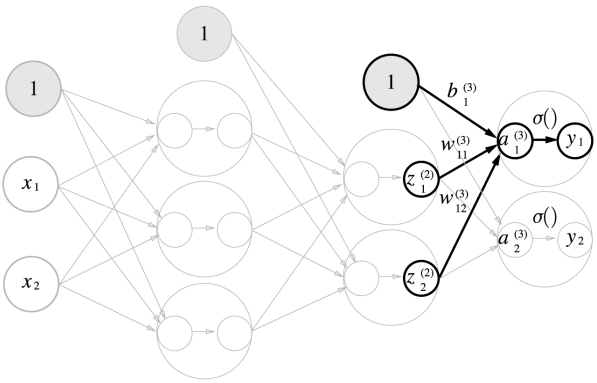


- $W_{12}^{(1)}$ 에서 앞에 1은 다음 층의 1번째 뉴런, 2는 앞층의 2번째 뉴런, (1)은 1층의 가중치를 의미합니다.

3.3.2 각 층의 신호 전달



$$a_1^{(1)} = w_{11}^{(1)}x_1 + w_{12}^{(1)}x_2 + b_1^{(1)}$$



3.4 출력층 설계

- 출력층의 활성화 함수는 풀고자 하는 문제의 성질에 맞게 정합니다.

문제	함수
회귀문제	항등 함수
2클래스	시그모이드 함수
다중클래스	소프트맥스 함수

3.4.1 소프트맥스 함수 (Softmax Function)

$$y_k = \frac{exp(a_k)}{\sum exp(a_i)}$$

- 분모는 모든 입력 신호 지수함수의 합을 의미합니다.
- 분자는 입력신호 a_k의 지수함수를 의미합니다.
- 소프트맥스 함수의 출력은 0 ~ 1 사이의 실수입니다.
- 소프트맥스 함수의 총합은 1입니다.

※ 주의점

- 소프트맥스 함수의 문제점은 오버플로 문제가 생길 수 있다는 점입니다.
- 예를 들어 $exp(100)$은 0이 40개가 넘는 큰 값이 되고 따라서 수식을 안정화시킬 필요성이 있습니다.

$$y_k = \frac{exp(a_k)}{\sum exp(a_i)} = \frac{Cexp(a_k)}{C\sum exp(a_i)} = \frac{exp(a_k + logC)}{\sum exp(a_i + logC)} = \frac{exp(a_k + C')}{\sum exp(a_i + C')}$$

- 똑같은 상수를 곱하고 exp에 넣어준다. $logC = C'$로 치환합니다.
- 일반적으로 C'는 입력신호 중 최대값에 (-)를 붙입니다.
- 이 값을 빼서 전체적으로 값을 낮추는 역할을 합니다.

3.4.2 출력층의 뉴런 수 정하기

- 출력층의 뉴런 수는 문제에 맞게 적절히 정합니다.
- 분류에선 분류하고 싶은 클래스 수로 설정하는 것이 일반적입니다.

3.5 정리

- 신경망에서는 활성화 함수로 시그모이드 함수와 ReLU 함수 같은 매끄럽게 변화하는 함수를 이용합니다.
- Numpy 다차원 배열을 잘 사용하면 신경망을 효율적으로 구현할 수 있습니다.
- Machine Learning은 크게 regression과 classification로 나눌 수 있습니다.
- 출력층의 활성화 함수로는 regression은 주로 항등 함수를, classification에서는 소프트맥스 함수를 이용합니다.
- 분류에서는 출력층의 뉴런 수를 분류하려는 클래스 수와 같게 설정합니다.
- 입력 데이터를 묶은 것을 배치라 하며, 추론 처리를 이 배치 단위로 진행하면 결과를 훨씬 빠르게 얻을 수 있습니다.