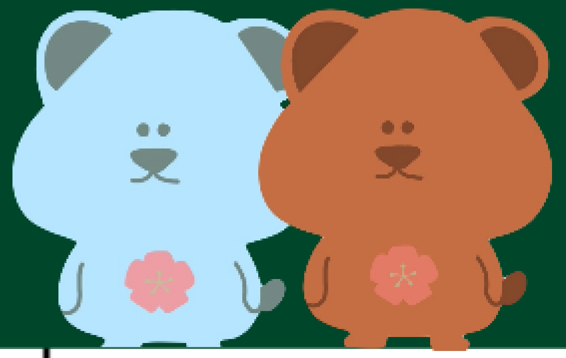


Week4 발표

조승연

목차

#01 활성화 함수

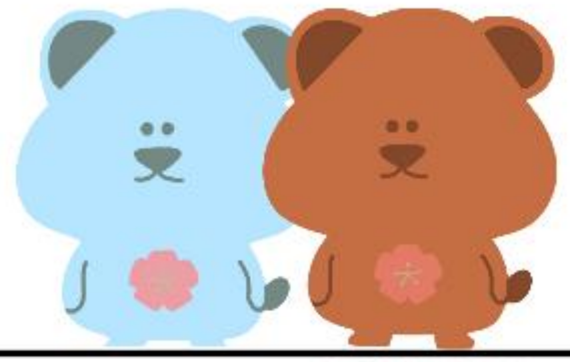
#02 왜 비선형 활성화 함수?

#03 활성화 함수의 미분

#04 신경망 네트워크와 경사 하강법

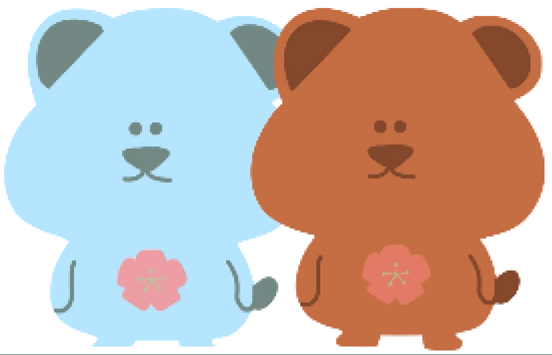
#05 역전파에 대한 이해

#06 랜덤 초기화



EWHA EURON

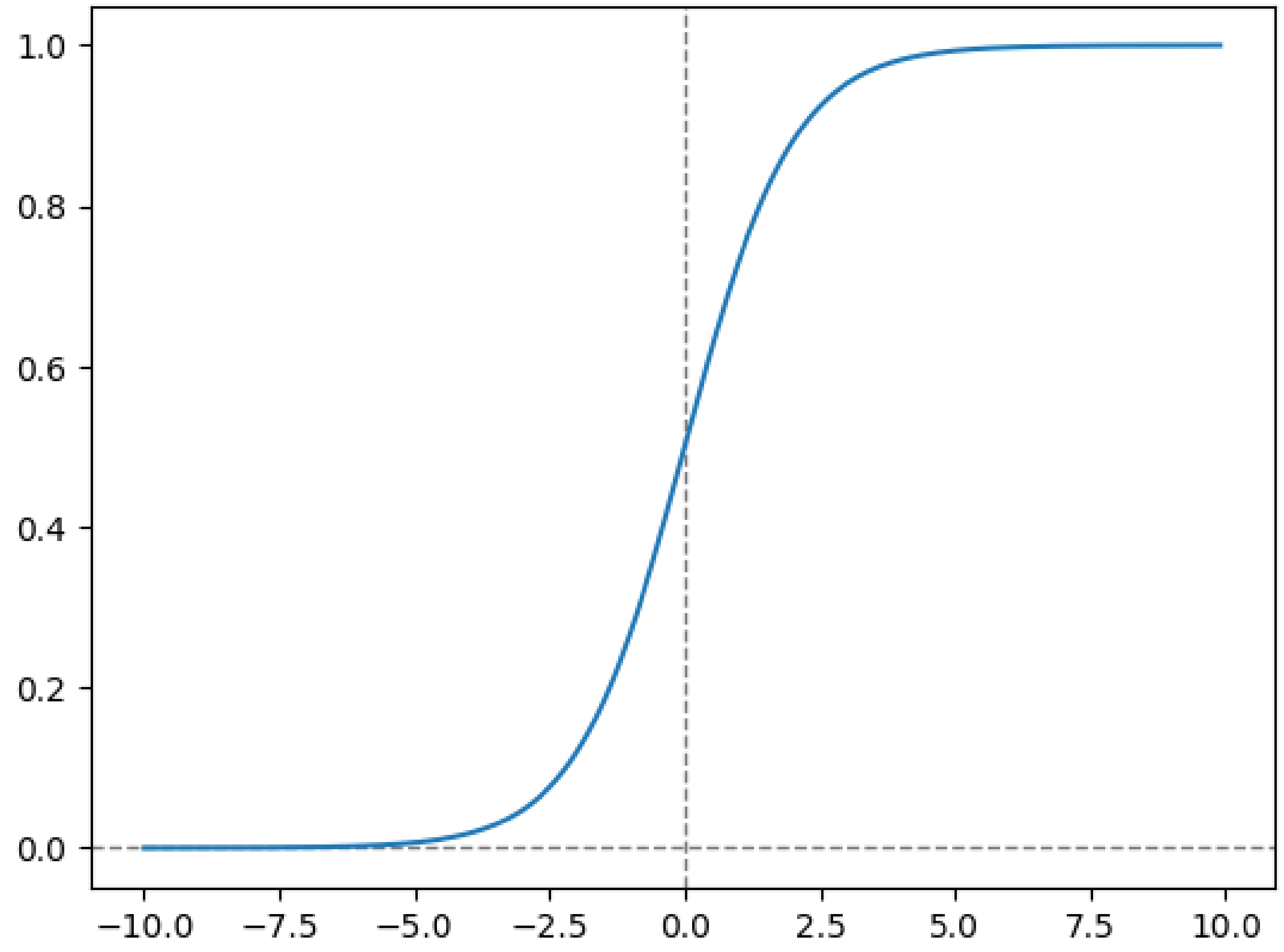
활성화 함수



#Sigmoid

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$\sigma'(z) = \frac{e^{-z}}{(1+e^{-z})^2}$$



항상 0과 1 사이의 값을 반환

입력값의 절대값이 커짐에 따라 기울기가 0에 급격히 가까워짐
→ 경사하강법이 느려지는 게 문제!!

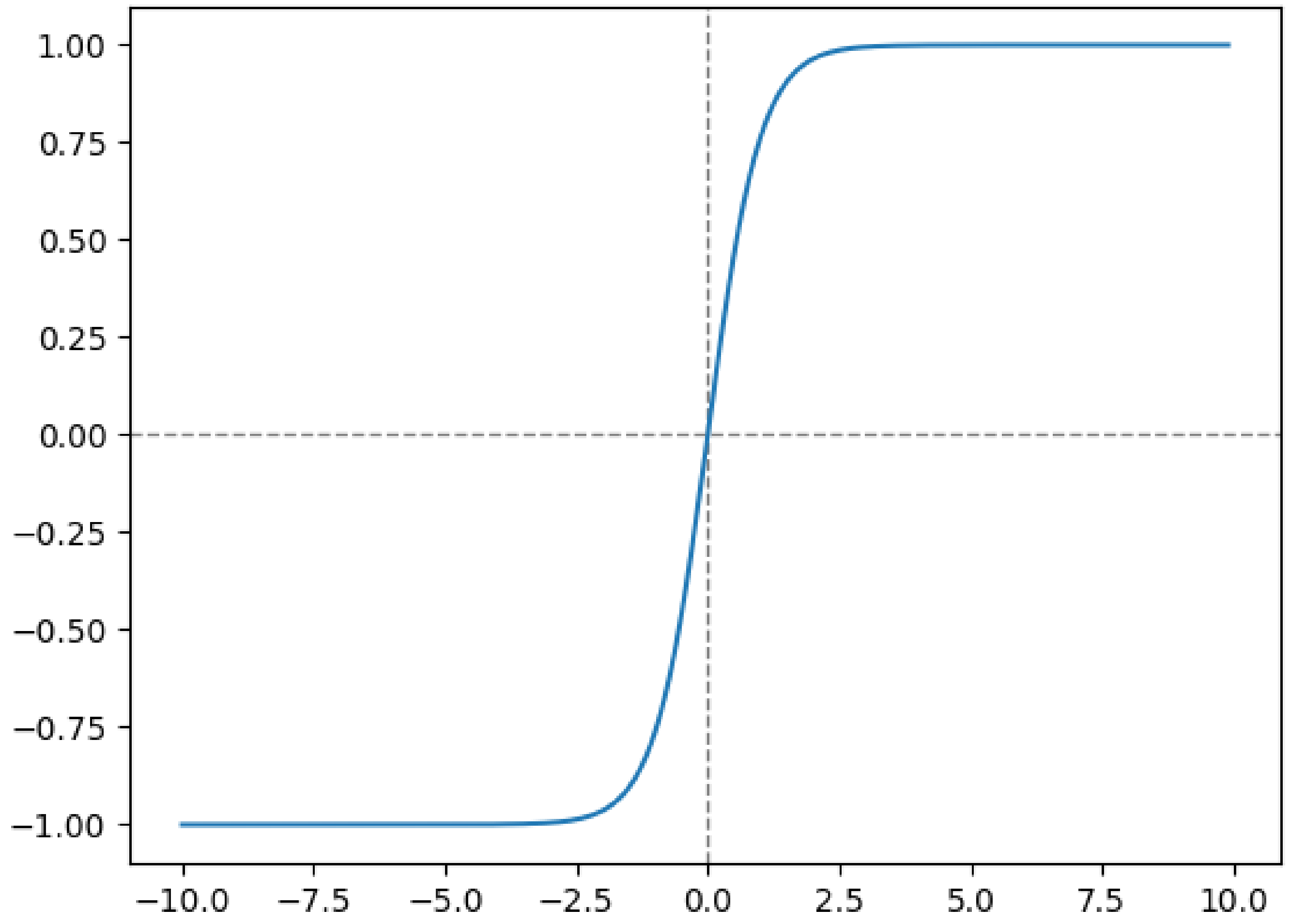
그래서 잘 안쓰이지만 이진 분류의 출력층에선 꽤 자주 쓰인다~!

EWHA EURON

#tanh

$$tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$tanh'(z) = 1 - \frac{(e^z - e^{-z})^2}{(e^z + e^{-z})^2}$$



수학적으로 시그모이드를 약간 옮긴 형태

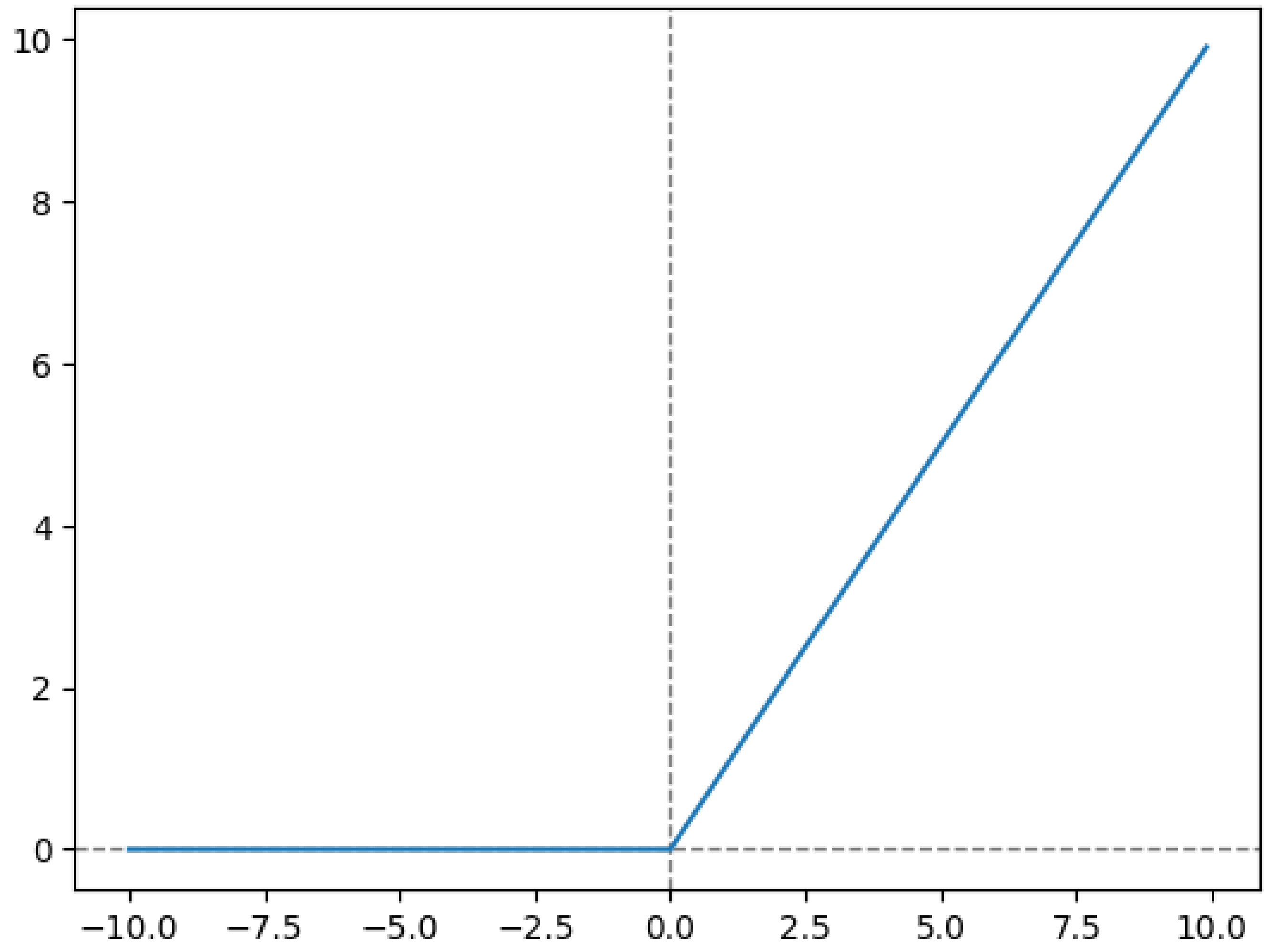
-1부터 1까지의 값을 반환

중심을 0.5가 아닌 0으로 맞춰줄 수 있어서 유리

#ReLu

$$ReLu(z) = max(0, z)$$

$$ReLu'(z) = \begin{Bmatrix} 0\,, & if \;\; z < 0 \\ 1\,, & if \;\; z > 0 \end{Bmatrix}$$



대부분의 정의역에 대해 기울기가 0이 아닌 함수
-> 대부분의 충분한 은닉 유닛의 정의역은 0보다 크기 때문에 잘 작동하게 됨

0에서 미분 불가, 그렇지만 0을 정확히 값으로 갖는 경우가 많지 않음

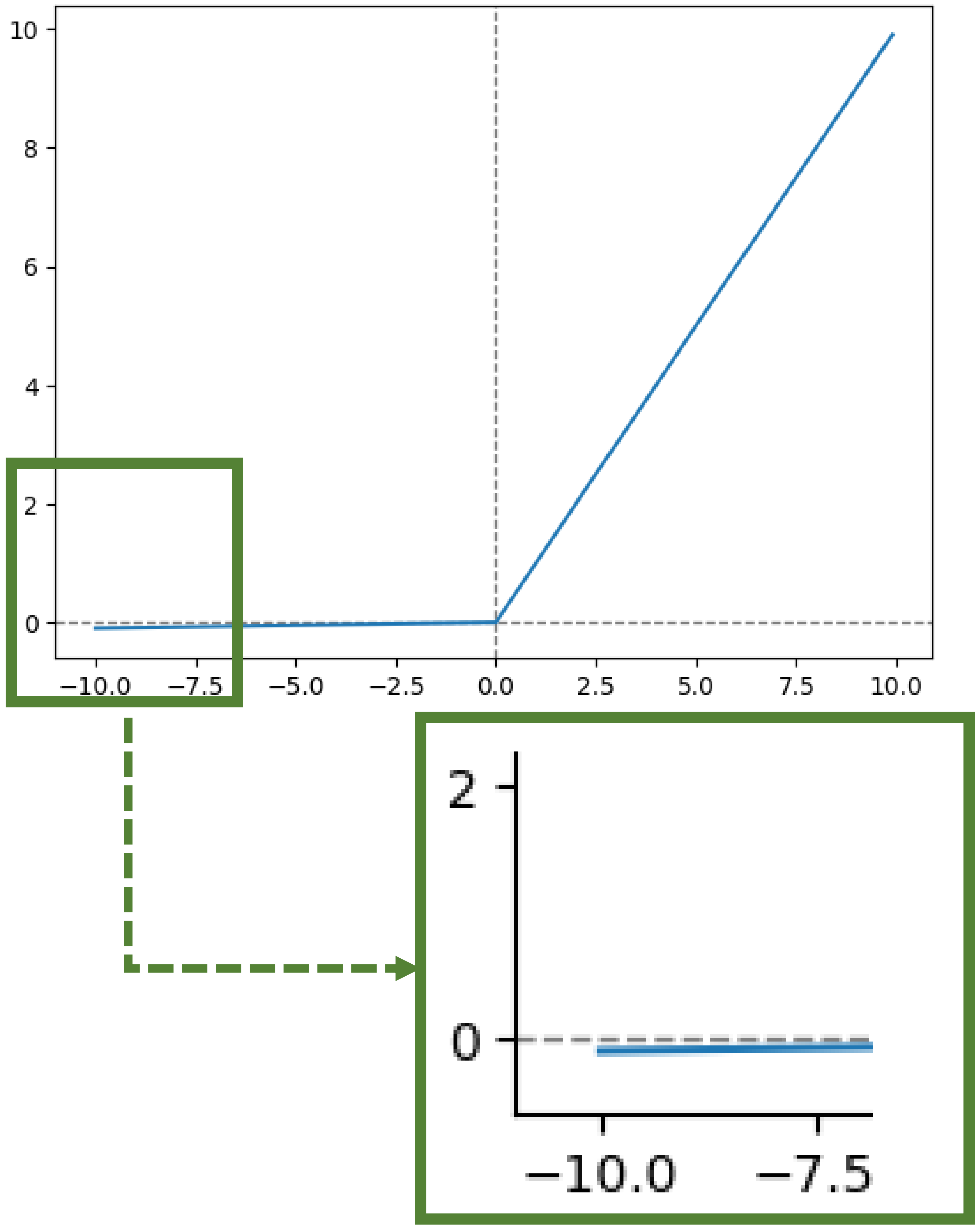
가장 많이 쓰이는, 뭘 쓸지 모르겠으면 냅다 사용가능한 그런 함수! 👍

EWHA EURON

#Leaky ReLu

$$Leaky(z) = max(0.01 * z, z)$$

$$L'(z) = \begin{cases} 0.01, & if \;\; z < 0 \\ 1, & if \;\; z > 0 \end{cases}$$

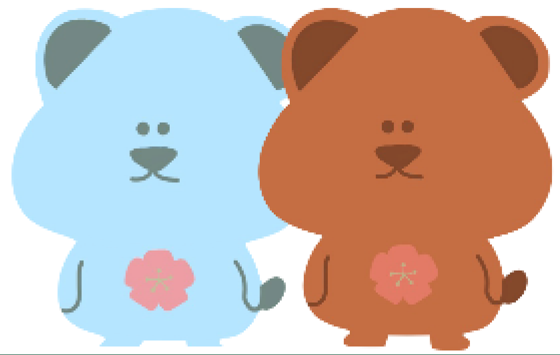


기존의 ReLu 함수의 음수부분 기울기를 살짝 만든 함수

입력값이 음수일 때의 기울기는 조절 가능

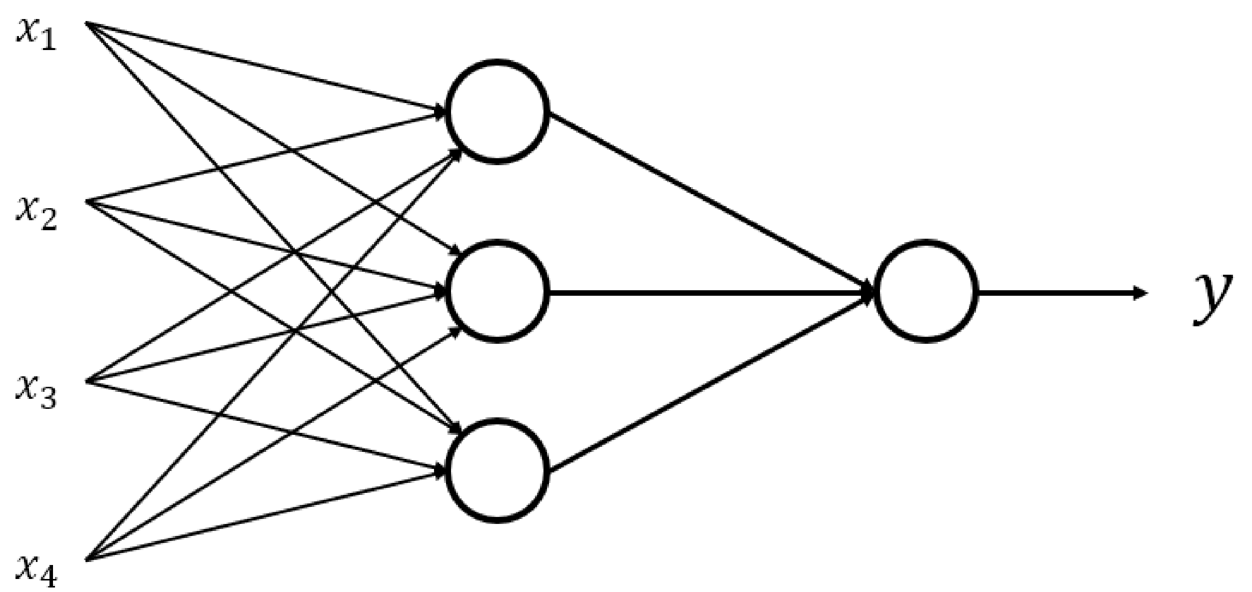
보통 ReLu보다 성능이 좋은 편이지만 자주 쓰이진 않음

왜 비선형 활성화 함수?



왜 비선형 활성화 함수를 써야하는가?

#if 활성화 함수: 선형함수 g(z) = z



$$z^{[1]} = W^{[1]} \cdot x + b^{[1]}$$
$$a^{[1]} = g^{[1]}(z^{[1]}) = z^{[1]}$$
$$z^{[2]} = W^{[2]} \cdot a^{[1]} + b^{[2]}$$
$$a^{[2]} = g^{[2]}(z^{[2]}) = z^{[2]} = W^{[2]} \cdot (W^{[1]} \cdot x + b^{[1]}) + b^{[2]}$$

신경망이 입력값의 선형식만을 출력
-> 결국 기본 로지스틱 회귀와 같은 형태의 신경망

활성화 함수의 미분

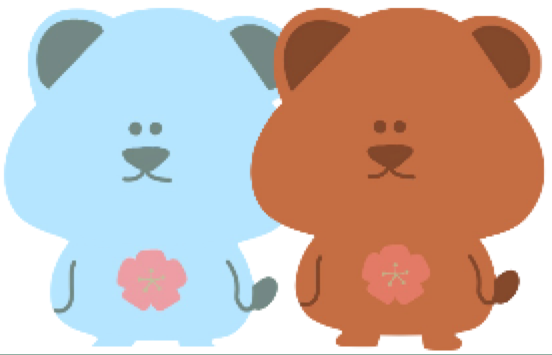
활성화 함수의 미분

$$\sigma'(z) = \frac{e^{-z}}{(1+e^{-z})^2} = \sigma(z)\cdot(1-\sigma(z)) = a(1-a)$$

$$tanh'(z) = 1 - \frac{(e^z - e^{-z})^2}{(e^z + e^{-z})^2} = 1 - (tanh(z))^2 = 1 - a^2$$

#시그모이드와 쌍곡 탄젠트 함수에 대해 위와 같은 공식 성립 가능

신경망의 경사 하강법



#신경망의 경사 하강법

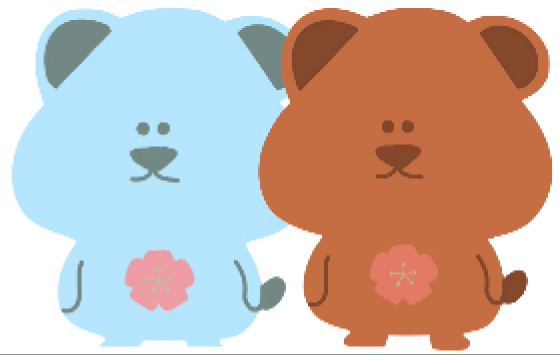
#가중치를 경사하강법을 통해 다음과 같이 업데이트
a : learning_rate, J : Cost function

$$W^{[1]} := W^{[1]} - \alpha \cdot \frac{dJ}{dW^{[1]}}$$
$$b^{[1]} := b^{[1]} - \alpha \cdot \frac{dJ}{db^{[1]}}$$

$$W^{[2]} := W^{[2]} - \alpha \cdot \frac{dJ}{dW^{[2]}}$$
$$b^{[2]} := b^{[2]} - \alpha \cdot \frac{dJ}{db^{[2]}}$$

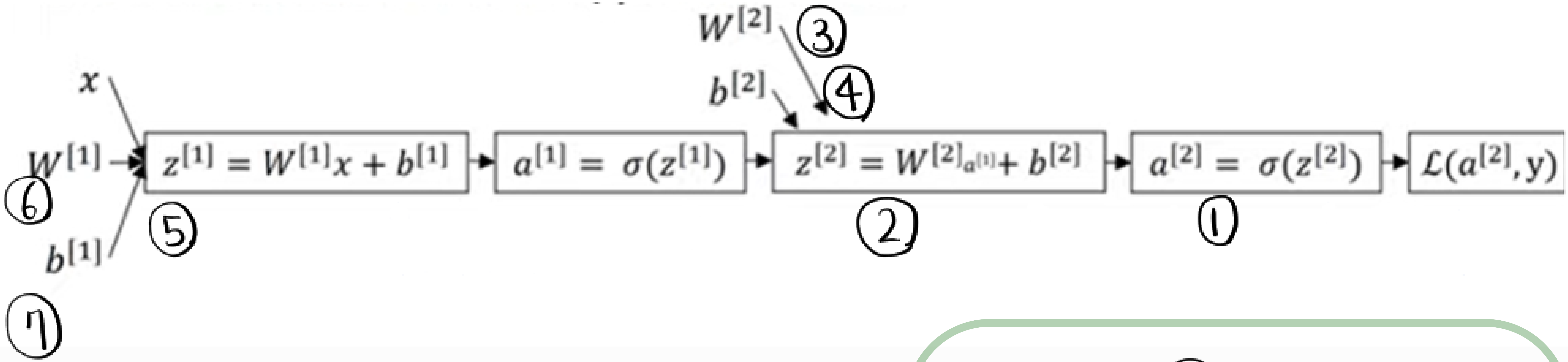
#그럼 각 도함수는 어떻게 구하는지? -> 역전파

역전파에 대한 이해

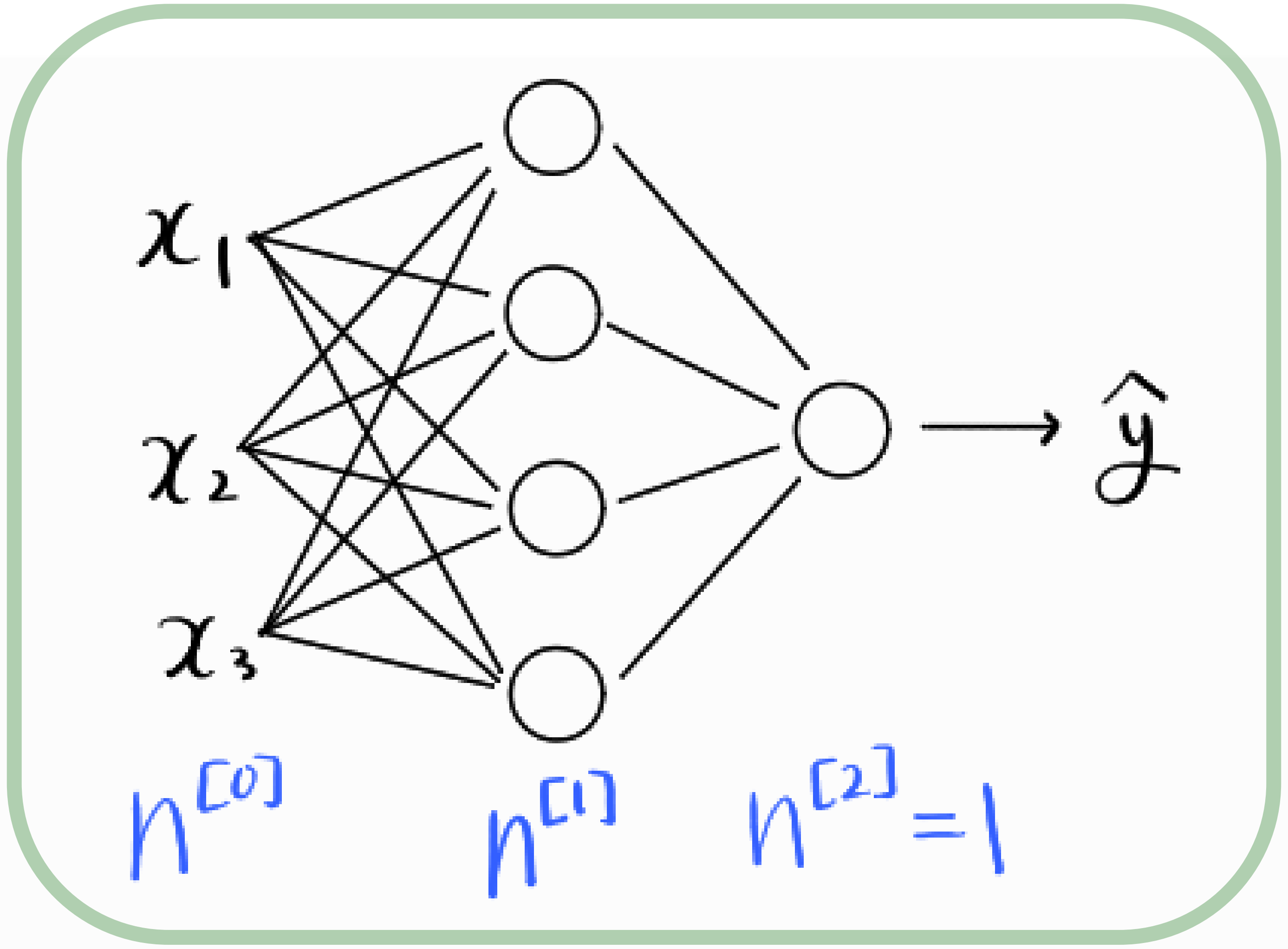


EWHA EURON

#역전파에 대한 이해



$$L = -(y\log a^{[2]} + (1-y)\log(1-a^{[2]}))$$



① $da^{[2]} = \dfrac{dL}{da^{[2]}} = -\dfrac{y}{a^{[2]}} + \dfrac{(1-y)}{1-a^{[2]}}$

② $dz^{[2]} = a^{[2]}(1-a^{[2]}) \cdot \dfrac{dL}{da^{[2]}} = a^{[2]} - y$

③ $dw^{[2]} = \dfrac{dL}{dz^{[2]}} \cdot \dfrac{dz^{[2]}}{dw^{[2]}} = dz^{[2]} \cdot a^{[1]T}$

④ $db^{[2]} = \dfrac{dL}{dz^{[2]}} \cdot \dfrac{dz^{[2]}}{db^{[2]}} = dz^{[2]}$

$W^{[2]} \rightarrow (n^{[2]}, n^{[1]})$

$z^{[2]}, dz^{[2]} \rightarrow (n^{[2]}, 1)$

$a^{[1]} \rightarrow (n^{[1]}, 1)$

⑤ $dz^{[1]} = \underbrace{W^{[2]T} dz^{[2]}}_{(n^{[1]},1)} * \overbrace{g^{[1]\prime}(z^{[1]})}^{(n^{[1]},1)}$

$(n^{[1]}, 1) \leftarrow$

$W^{[2]} \rightarrow (n^{[2]}, n^{[1]})$

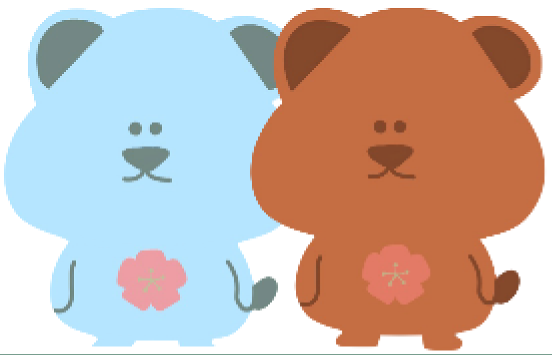
$z^{[2]}, dz^{[2]} \rightarrow (n^{[2]}, 1)$

⑥ $dw^{[1]} = dz^{[1]} \cdot x^{T}$

⑦ $db^{[1]} = dz^{[1]}$

EWHA EURON

랜덤 초기화



#랜덤 초기화

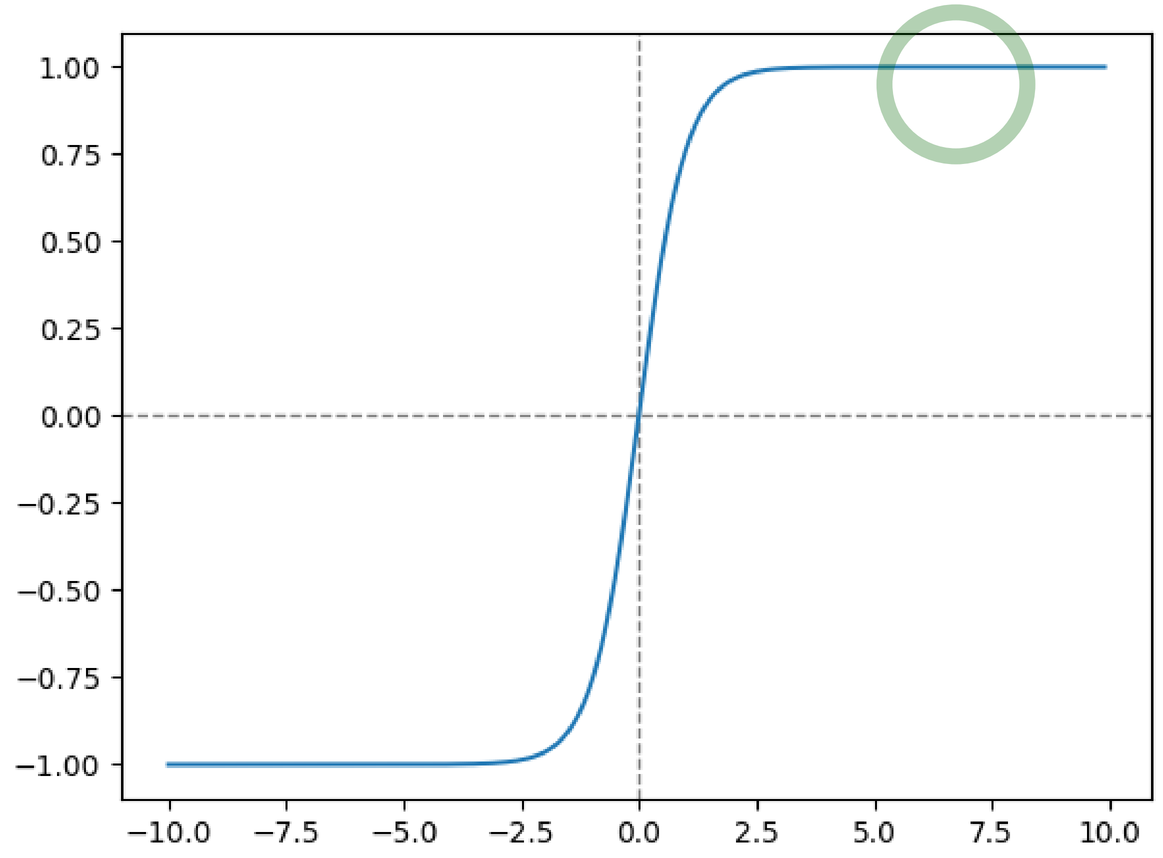
#가장 처음의 w 값을 무엇으로 둘 것인지?
→ w의 초기값을 0으로 두고 신경망을 학습
→ 모든 z의 값이 같음 / 결국 모든 층에 대해 dw 값이 같음

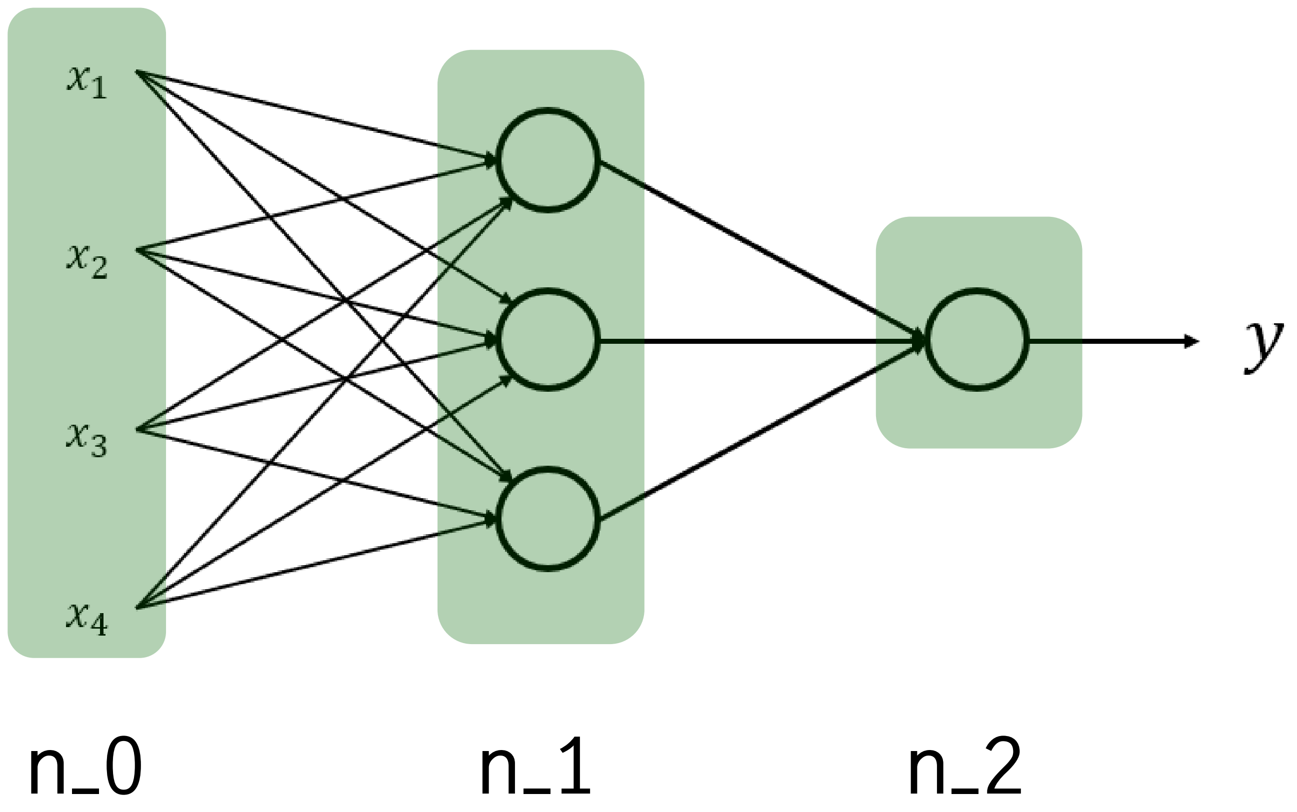
→ 랜덤한 수를 뽑아서 진행
→ 작은 수로 생성

```
W1 = np.random.randn(n_1, n_0)*0.01
b1 = np.zeros((n_1, 1))
W2 = np.random.randn(n_2, n_1)*0.01
b2 = np.zeros((n_2, 1))
```

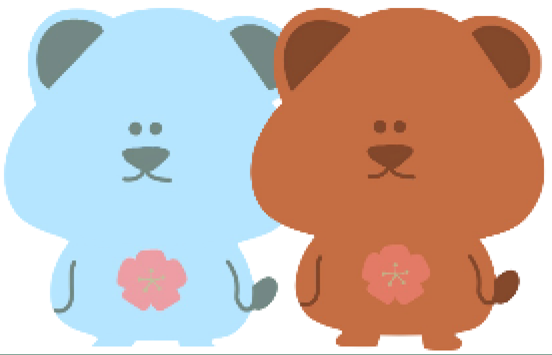
#-> 주어진 데이터의 크기에 맞춰서 랜덤 행렬 생성

np.random.randn : 정규분포 따르는 랜덤한 수 생성





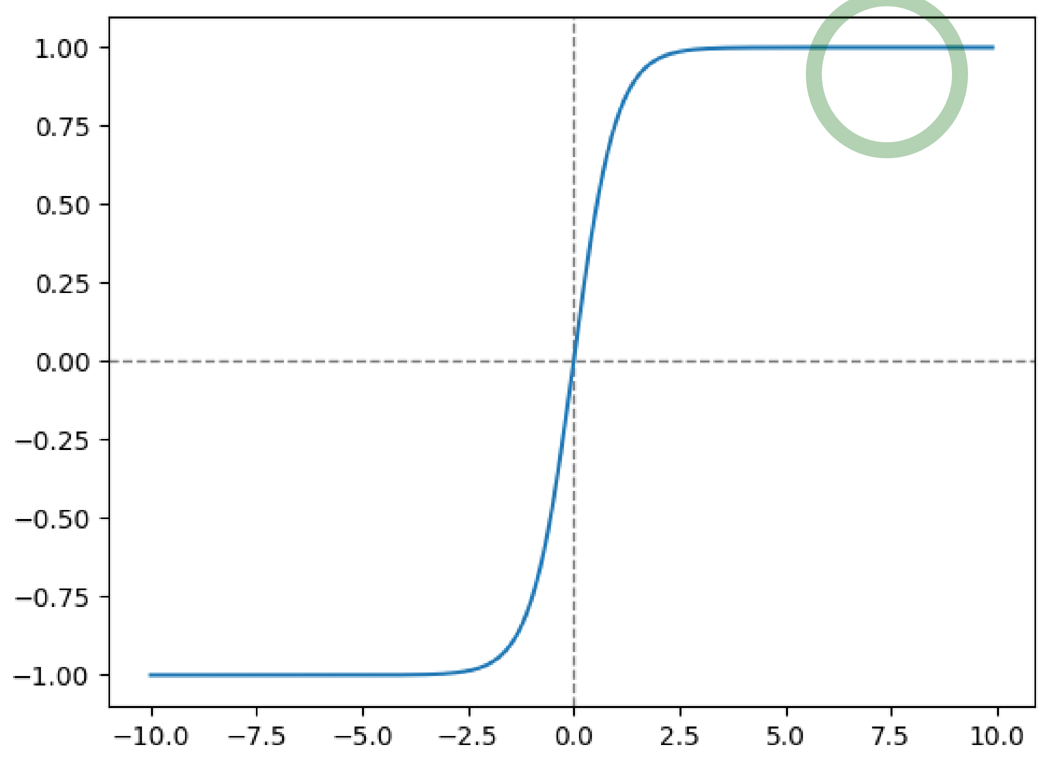
출석 퀴즈 리뷰



#3번

3. hidden layer에 tanh 활성화 함수를 이용한 네트워크를 만들었습니다. 가중치를 np.random.randn(..,..)*1000으로 상대적으로 크게 설정했다면 어떤 일이 일어날까요? * 1점

- 가중치를 랜덤으로 설정하는 한, 가중치가 크지 작은지 여부는 중요하지 않다. 따라서 아무 일도 일어나지 않는다.
- tanh의 입력이 매우 커지게 되어 기울기도 커진다. 따라서 발산을 방지하기 위해서는 α를 아주 작게 설정해야 하고, 이는 학습 속도를 느리게 할 것이다.
- tanh의 입력이 매우 커지게 되어 각 유닛들이 강하게 활성화된다. 따라서 가중치가 작은 값에서 시작하는 경우보다 학습 속도가 빨라진다.
- (선택됨) tanh의 입력이 매우 커지게 되어 기울기가 0에 가까워진다. 따라서 최적화 알고리즘의 속도가 느려진다.



#8번

8. 다음과 같이 하나의 은닉층을 가진 신경망이 있을 때, 다음 설명 중 옳은 것을 **모두** 골라주세요. * 1점

1. $W^{[2]}$의 shape는 (1,4)이다.
2. $b^{[1]}$의 shape는 (2,1)이다.
3. $b^{[2]}$의 shape는 (4,1)이다.
4. $W^{[1]}$의 shape는 (2,4)이다.
5. $b^{[2]}$의 shape는 (1,1)이다.
6. $b^{[1]}$의 shape는 (4,1)이다.
7. $W^{[2]}$의 shape는 (4,1)이다.

#8번

$$W^{[1]}x + b^{[1]} = \underbrace{\begin{bmatrix} \text{———} \\ \text{———} \\ W^{[1]} \\ \vdots \\ \text{———} \end{bmatrix}}_{} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \underbrace{\begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ \vdots \\ b_{n_1}^{[1]} \end{bmatrix}}_{n_1 \times 1} = \underbrace{\begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ \vdots \\ z_{n_1}^{[1]} \end{bmatrix}}_{n_1 \times 1} = z^{[1]}$$

$W^{[1]}: (n_1, n_0)$ $b^{[1]}, z^{[1]}, a^{[1]} : (n_1, 1)$

$W^{[2]}: (n_2, n_1)$ $b^{[2]}, z^{[2]}, a^{[2]} : (n_2, 1)$

$g(z^{[1]}) = a^{[1]} \rightarrow a^{[1]}$ Shape : $n_1 \times 1$

$$W^{[2]}x + b^{[2]} = \underbrace{\begin{bmatrix} \text{———} \\ \text{———} \\ W^{[2]} \\ \vdots \\ \text{———} \end{bmatrix}}_{n_2 \times n_1} \underbrace{\begin{bmatrix} a_1^{[1]} \\ \vdots \\ a_{n_1}^{[1]} \end{bmatrix}}_{n_1 \times 1} + \underbrace{\begin{bmatrix} b_1^{[2]} \\ b_2^{[2]} \\ \vdots \\ b_{n_1}^{[2]} \end{bmatrix}}_{n_2 \times 1} = \underbrace{\begin{bmatrix} z_1^{[2]} \\ z_2^{[2]} \\ \vdots \\ z_{n_1}^{[2]} \end{bmatrix}}_{n_2 \times 1} = z^{[2]}$$

$g(z^{[2]}) = a^{[2]} \rightarrow a^{[2]}$ Shape : $n_2 \times 1$

$$W^{[1]} : (n_1, n_0)$$

$$b^{[1]}, z^{[1]}, a^{[1]} : (n_1, 1)$$

$$W^{[2]} : (n_2, n_1)$$

$$b^{[2]}, z^{[2]}, a^{[2]} : (n_2, 1)$$

THANK YOU