



Activation Function

미래연구소 14기 4주차

o. 배경 지식

① $dW^{[1]} = \frac{1}{m} (W^{[2]T} dZ^{[2]} * g^{[1]'}(Z^{[1]})) X^T$ \longrightarrow ② $\text{Vanishing gradient } (g' \downarrow \longrightarrow \frac{\partial J}{\partial W} \downarrow)$
(gradient descent 속도↓)

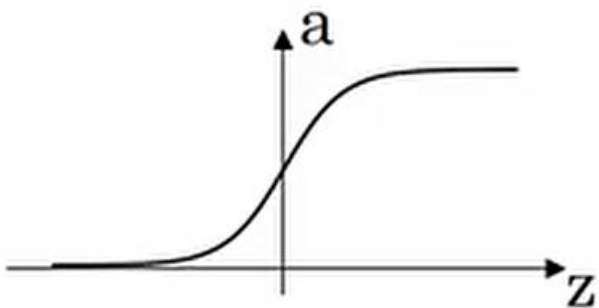
1) **activation**의 미분계수가 0에 가까우면 = **activation**의 접선 기울기가 0에 가까우면

-> dW 의 감소 -> **gradient descent**가 잘 일어나지 않음 (학습이 일어나지 않음)

(saturation: weight의 update가 잘 일어나지 않는 현상)

(vanishing gradient : gradient가 0에 가까워지는 (소실되는) 현상)

1. Sigmoid



$$1) 0 \leq g(z) \leq 1$$

장점 1: binary classification의
output layer라는 특수한 상황에 적합

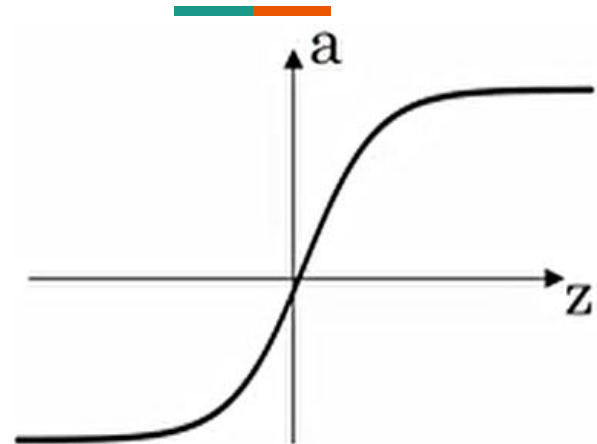
$$g(z) = \frac{1}{1 + e^{-z}}$$

$$g'(z) = a(1 - a)$$

2) $g'(z)$ 가 0에 가까운
구간이 많다.

단점 2: gradient descent 속도 저하 ↓

2. tanh (hyperbolic tangent)



$$1) -1 \leq g(z) \leq 1$$

2) $g'(z)$ 가 0에 가까운 구간이 많다.

$$(0 < g'(z) < 1)$$

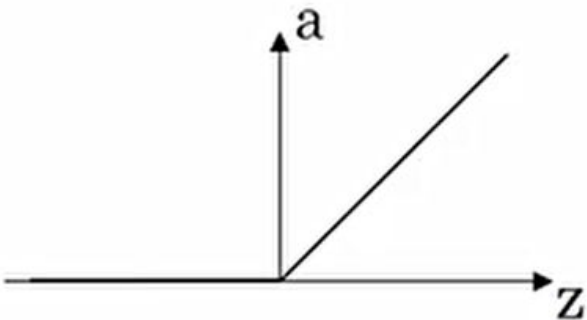
$$g(z) = -\frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$g'(z) = (1 - a^2)$$

장점 1: sigmoid보다는 vanishing gradient가 덜하다. ($g'(z)$ 의 최댓값이 1)

단점 1: gradient descent 속도 저하↓

3. ReLU (Rectified Linear Unit)



1) $g'(z) = 1$ 인 구간이
절반이다.

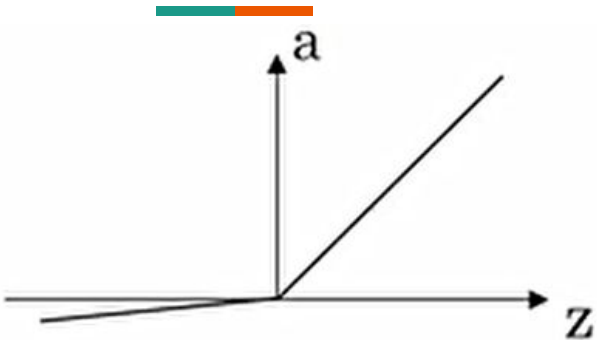
장점 1: sigmoid, tanh의 vanishing
gradient 문제 해결

단점 1: 그래도 절반이 gradient가 0
(dying ReLU 현상)

$$g(z) = \begin{cases} z & z > 0 \\ 0 & z < 0 \end{cases}$$

$$g'(z) = \begin{cases} 1 & z > 0 \\ 0 & z < 0 \end{cases}$$

4. Leaky ReLU



1) gradient가 0인 구간이 없다.

장점 1: dying ReLU 현상을 해결
(GAN 과 같은 train이 어려운 경우에 사용)

단점 1: ReLU 보다는 비선형성이 덜하다.

$$g(z) = \begin{cases} z & z > 0 \\ 0.01z & z < 0 \end{cases}$$

$$g'(z) = \begin{cases} 1 & z > 0 \\ 0.01 & z < 0 \end{cases}$$

5. output layer의 activation function

	Regression	Binary classification	multi-class classification
output layer의 activation function		sigmoid	softmax

softmax는 C3 W3에 있으나 6주차에 소개할 계획입니다.

6. activation이 non-linear해야 하는 이유

0. 딥러닝은 거대한 합성함수이다.

1. activation이 linear하다.

- 1) 선형변환이다. = 일차함수이다.
- 2) 선형변환을 연속하면(일차함수를 계속 합성하면) 일차함수 꼴이 나온다.
- 3) 딥러닝은 그러면 안 된다. ex> binary에게 가장 이상적인 activation = step function
- 4) data를 고차원 공간에 두고 이를 함수로 구분 짓는 일 -> 구분을 위해 왜곡된 함수가 필요 (하지만 일차함수는 왜곡이 불가능)

2. non-linear activation이 필요한 이유

- 1) 그림 자료 -> 다양한 모양의 함수를 만들 수 있다.
- 2) w , b 가 gradient descent에 따라 변하면서 최적의 딥러닝 함수를 찾는다.
- 3) 선형함수를 쓰면 이런 역할 불가능

3. ReLU / Leaky ReLU