

가천대 회화·조소과 AI 특강

2021-06-27

조형래

활성화 함수

(activation function)

활성화 함수(activation function)을 사용하는 이유

<XOR문제> - 선형분류기의 한계

XOR과 같은 non-linear한 문제는
선형분류기의 한계



AND

AND		
Input_A	Input_B	Output
0	0	0
0	1	0
1	0	0
1	1	1



OR

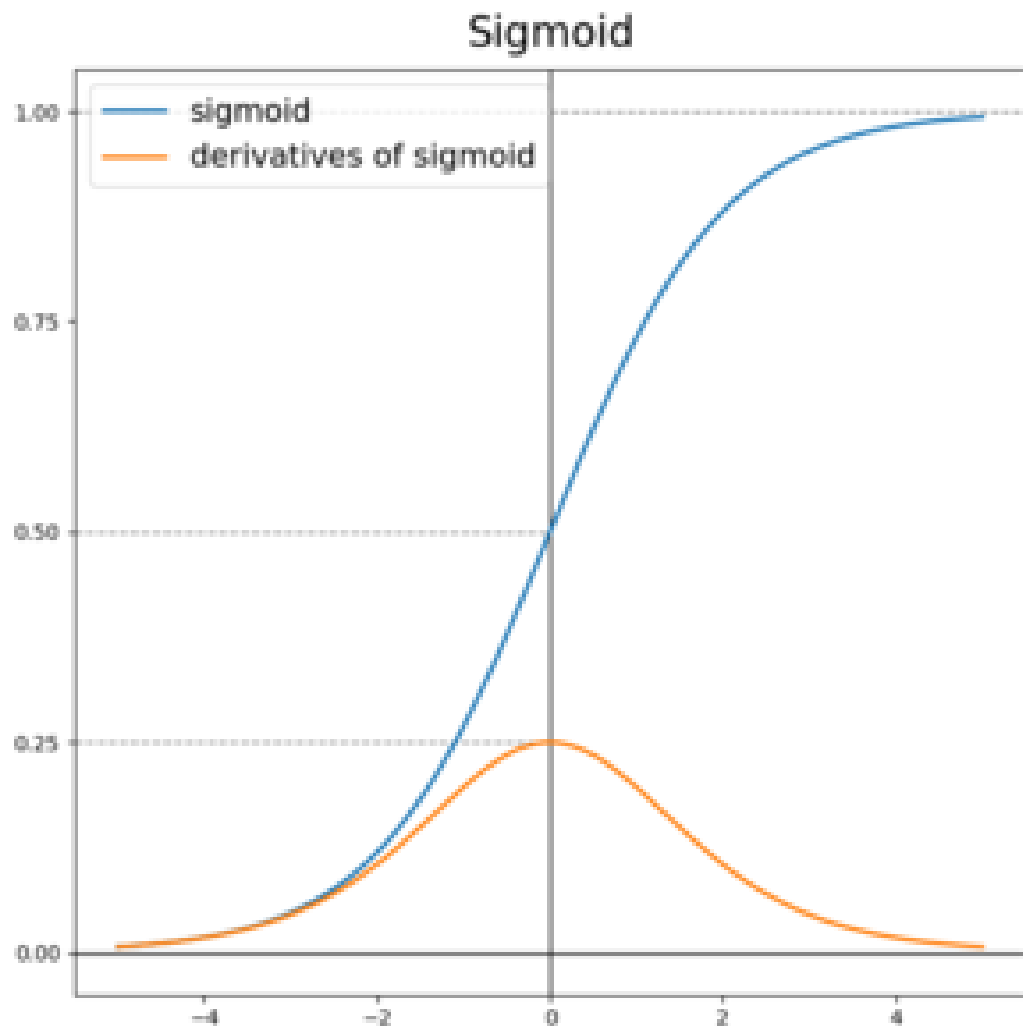
OR		
Input_A	Input_B	Output
0	0	0
0	1	1
1	0	1
1	1	1



XOR

XOR		
Input_A	Input_B	Output
0	0	0
0	1	1
1	0	1
1	1	0

시그모이드(Sigmoid)



$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad \sigma'(x) = \sigma(x)(1 - \sigma(x))$$

- output값을 0에서 1사이로 만들어준다. 데이터의 평균은 0.5를 갖게 된다.

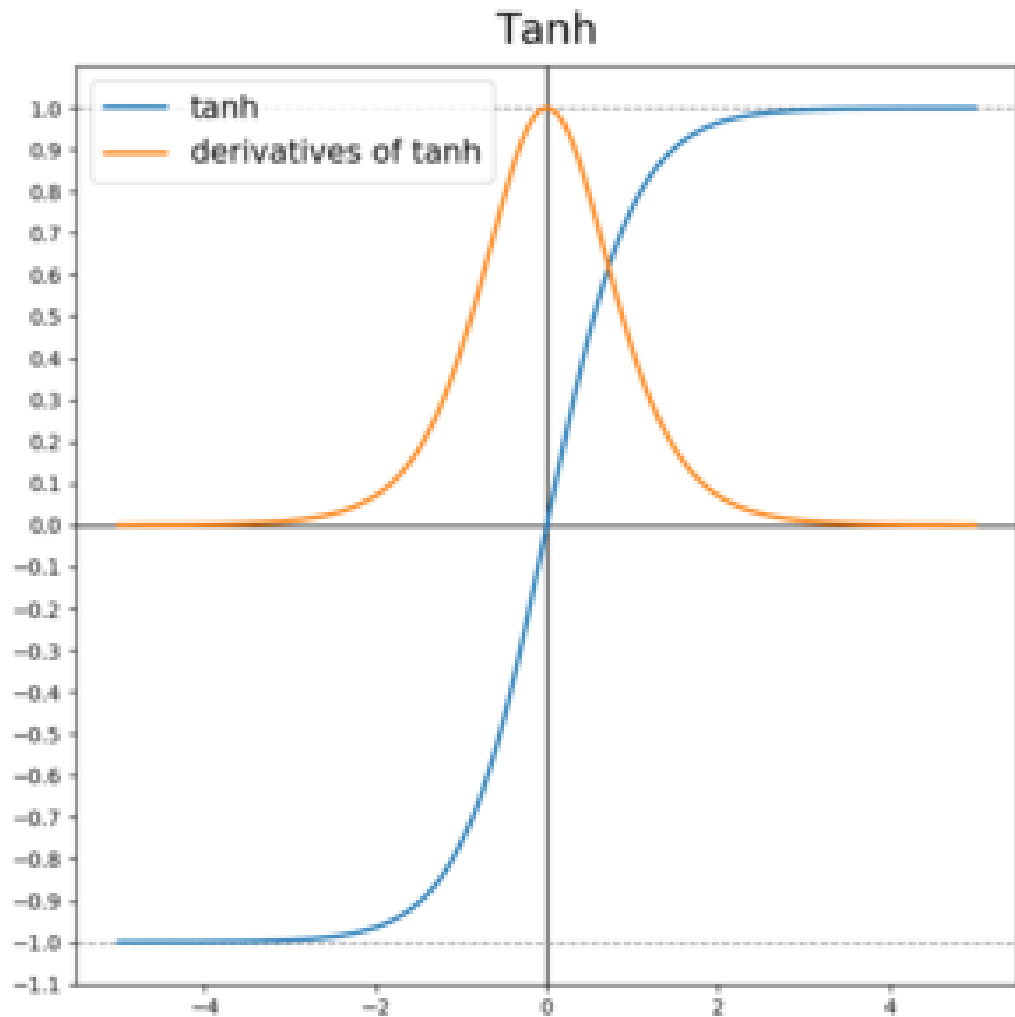
Vanishing gradient

Input 쪽으로 weight를 업데이트 해나갈 때, 활성화 함수의 미분값이 계속해서 곱해지게 되는데 시그모이드 함수의 미분값은 최대값이 0.25 그리고 x 값이 0에서 멀어질수록 0에 매우 근접한다. 따라서 이 값을 계속 곱해줄수록 Weight의 변화량은 매우 작아지는 현상이 발생하기 때문에, Input 으로 가까워질수록 Weight들이 잘 업데이트 되지 않는다. 이것을 ' Gradient Vanishing ' 이라고 한다.

시그모이드를 사용하는 경우

binary classification 경우 출력 층 노드가 1개이므로 이 노드에서 0~1사이의 값을 가져야 마지막에 cast를 통해(ex. 0.5이상이면 1, 미만이면 0) 1혹은 0값을 output으로 받을 수 있다. 따라서 이때 시그모이드를 사용한다.

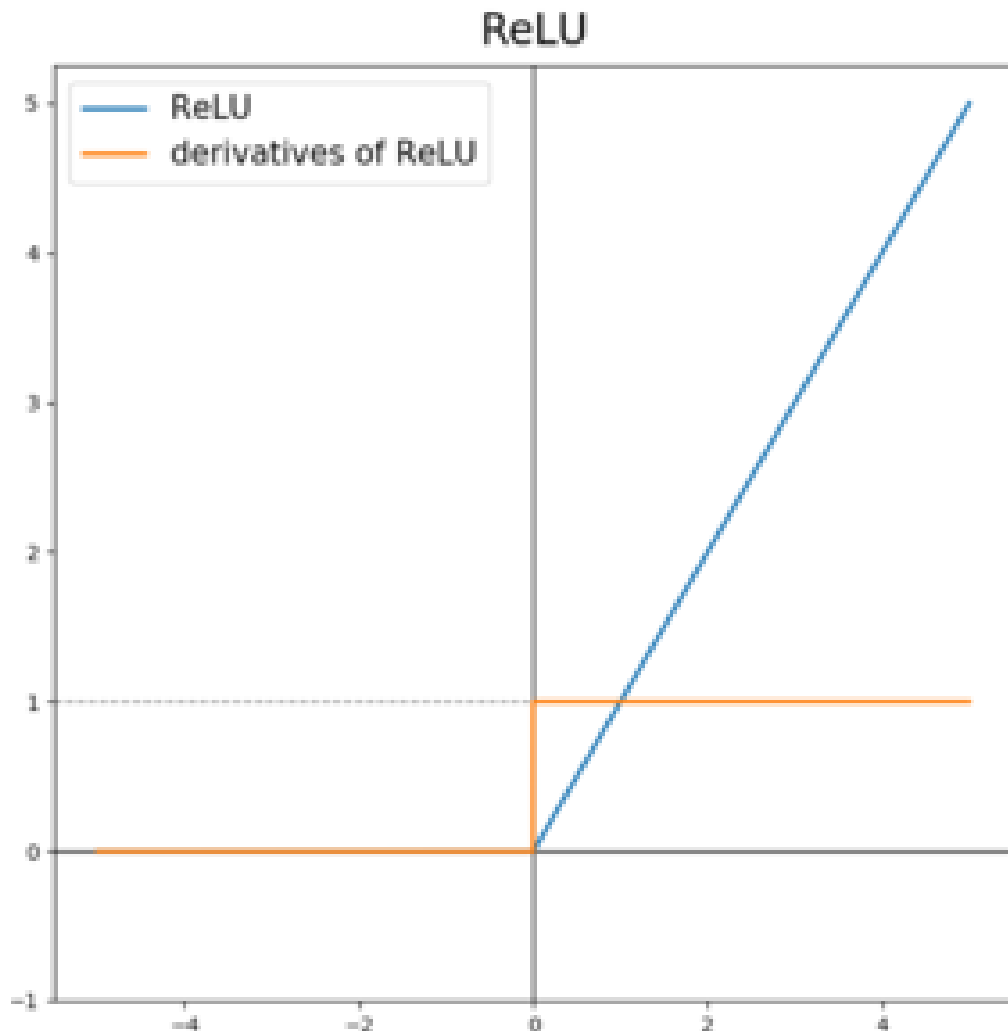
Tanh



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

- Tanh는 -1~1값을 가지고 데이터의 평균이 0
- 시그모이드와 마찬가지로 Vanishing gradient라는 단점이 있다.
- output데이터의 평균이 0으로써 시그모이드 보다 대부분의 경우에서 학습이 더 잘 된다.

ReLU

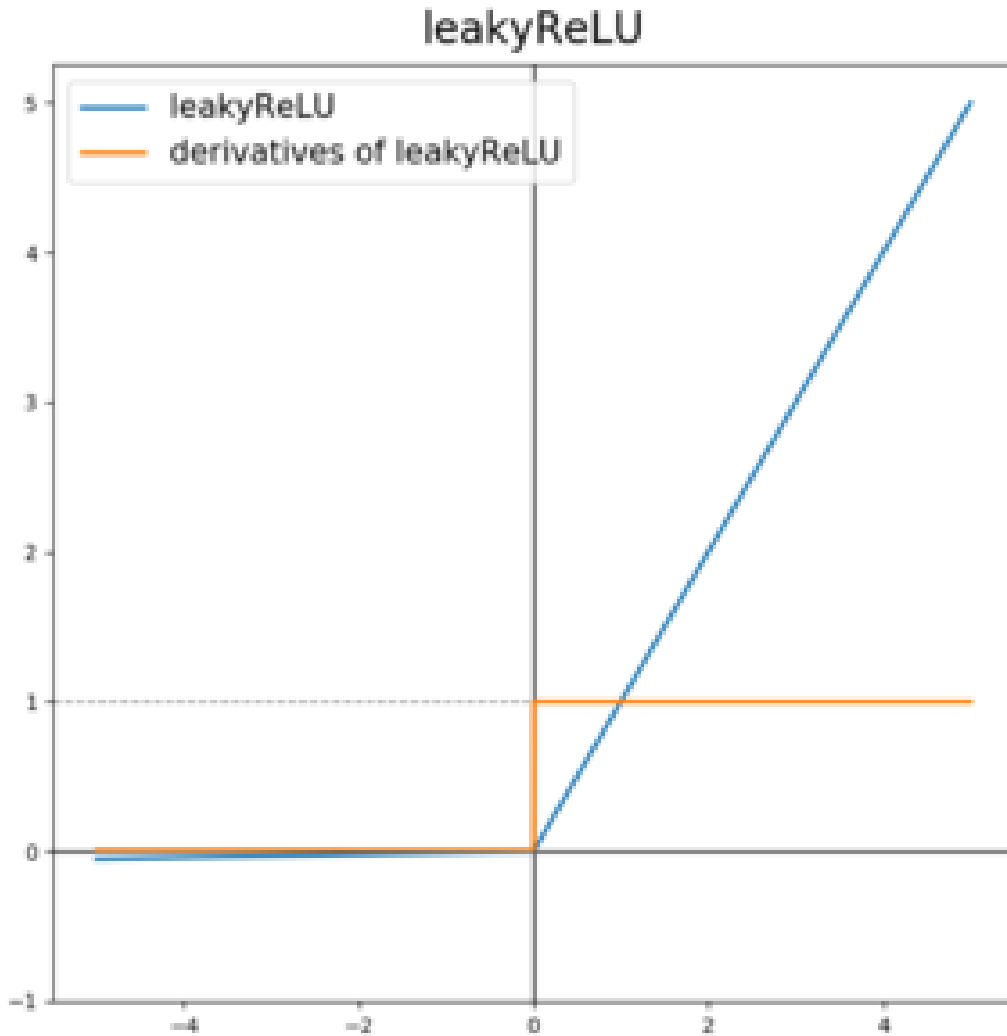


$$f(x) = \max(0, x)$$

- 활성화 함수 중 가장 많이 쓰는 ReLU
- 대부분의 input값에 대해 기울기가 0이 아니기 때문에 학습이 빨리 된다. 학습을 느리게 하는 원인이 gradient가 0이 되는 것인데 이를 대부분의 경우에서 막아주기 때문에 시그모이드, Tanh같은 함수보다 학습이 빠르다.
- x가 음의 값을 가지면 0을 출력하고, 양의 값을 가지면 x를 그대로 출력한다.
- 그림을 보면 input이 0보다 작을 경우 기울기가 0이나 실제로 hidden layer에서 대부분 노드의 z값은 0보다 크기 때문에 기울기가 0이 되는 경우가 많지 않다.
- 단점) 음의 값을 가지면 전부 그냥 0을 출력하기 때문에, 몇몇 Weight들이 업데이트 되지 않는 문제가 발생(Dying ReLU 현상)

leaky ReLU

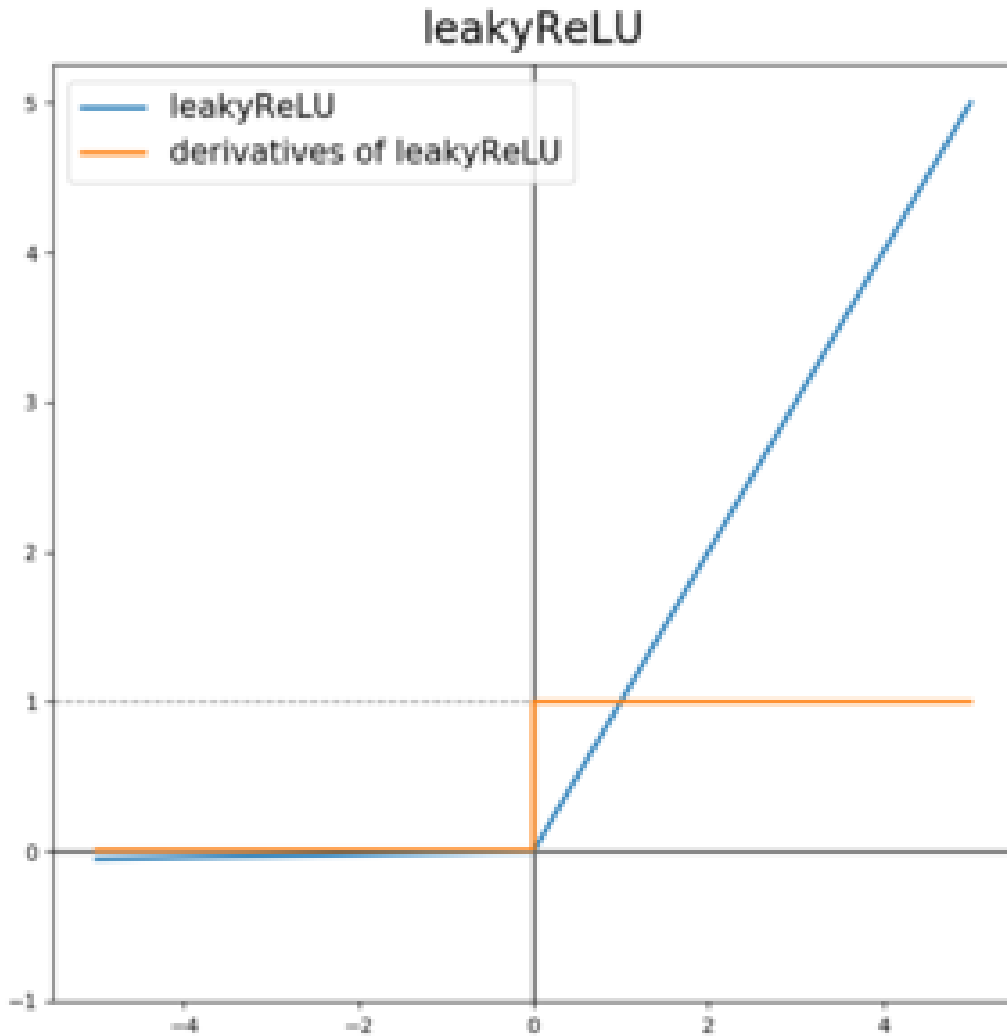
$$a = \max(0.01z, z)$$



- $\max(0.01z, z)$
즉, input값인 z 가 음수일 경우 기울기가 0이 아닌 0.01값을 갖게 된다.
- leaky ReLU를 일반적으로 많이 쓰진 않지만 ReLU보다 학습이 더 잘 되긴 한다. Dying ReLU 현상을 어느정도 막을 수 있습니다.

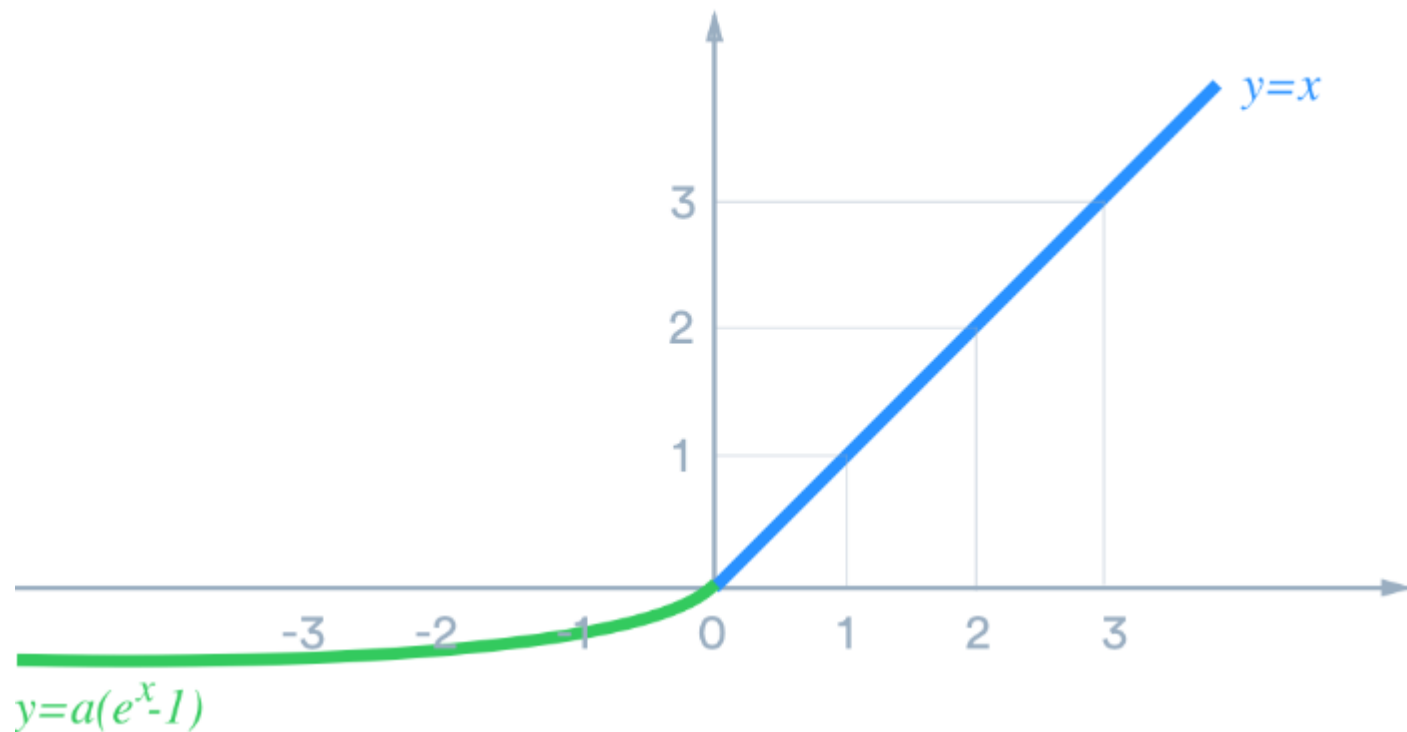
leaky ReLU

$$a = \max(0.01z, z)$$



- $\max(0.01z, z)$
즉, input값인 z 가 음수일 경우 기울기가 0이 아닌 0.01값을 갖게 된다.
- leaky ReLU를 일반적으로 많이 쓰진 않지만 ReLU보다 학습이 더 잘 되긴 한다. Dying ReLU 현상을 어느정도 막을 수 있습니다.

Exponential Linear Unit (ELU)



$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(\exp(x) - 1) & \text{if } x \leq 0 \end{cases}$$

ReLU의 음의 값에 대하여
exponential 값이 추가된 함수로
Dying ReLU 현상 해결한
방법입니다.

Maxout

$$h_i(x) = \max_{j \in [1, k]} z_{ij}$$

$$f(x) = \max(w_1^T x + b_1, w_2^T x + b_2)$$

$$z_{ij} = x^T W_{\dots ij} + b_{ij}$$

실제로 maxout 함수가 성능이 제일 좋다고 하나
계산량이 복잡한 단점이 있다.

- H ; hidden layer를 나타내는 함수
입력으로 들어오는 값 들에서 최대값을 취한다.
- W 와 b ; 학습을 통해서 결정이 되는 파라미터