ReLU의 문제점과 변형들

이도형

목차

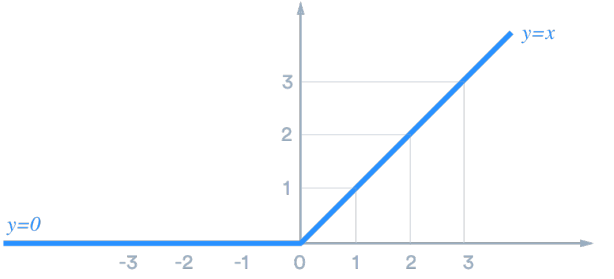
- ReLU란?
- ReLU의 장점과 단점
- ReLU의 변형들

ReLU란?

- 활성화 함수 중 하나
- Rectified Linear Unit의 약자
- Rectifier(정류자)

non-linear한 변형들로 LU의 이름 달고 있던데 왜그럴까?
ex) GELU, ELU,

- Input: x Output: max(0,x)



A Practical Guide to ReLU. Start using and understanding ReLU… | by Danqing Liu | Medium

ReLU의 장점

- Sparse activation

  어떤부분 없애는게 에러가 확실하다.
  대부분은 그렇지.

- Fewer vanishing gradient problem

  뒤에서

- Efficient computation

  비교, 크면 gradient 1
  안하면 0.

- Scale invariant

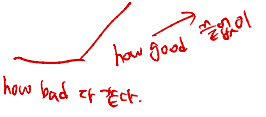
  $\max(0, ax) = a \max(0, x)$ for $a \geq 0$

ReLU의 잠재적 문제점

- Non-differential at 0

0 이나 1로 정하기 나름.

- Not 0 centered



- Unbounded

- Dying ReLU problem

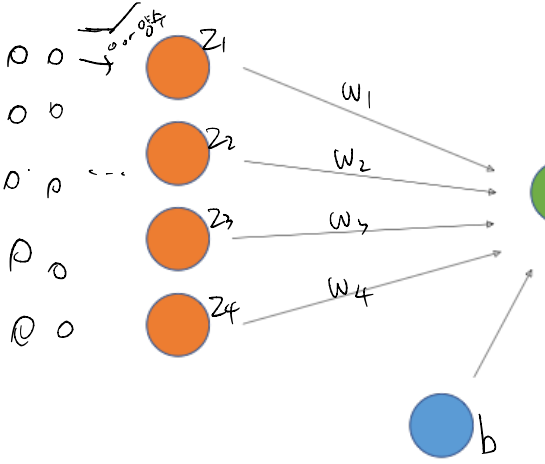
Dying ReLU problem

$\frac{e^x}{1+e^x}$ Sigmoid

$\sigma'(x) = \sigma(x)(1-\sigma(x))$

최대 $x=0$일때 $\frac{1}{4}$

학습할 신호를 0.



$\text{net}\,h = w_1 z_1 + \cdots + w_4 z_4 + b$

out h

$\text{out}\,h = \text{ReLU}(\text{net}\,h)$

back propagation weight update

$$\frac{\partial L}{\partial w_i} = \frac{\partial \text{net}\,h}{\partial w_i} \cdot \frac{\partial \text{out}\,h}{\partial \text{net}\,h} \cdot \frac{}{\partial \text{out}\,h} \cdots \frac{\partial \text{out}\,h'}{\partial \text{net}\,h'} \cdot \frac{\partial L}{\partial \text{out}\,h'}$$

$$w_{new} = w - \underbrace{\alpha}_{\text{learning rate}} \cdot \frac{\partial L}{\partial w_i}$$

lr 커서 w_i들 너무 작아져버렸다.

뒤 input 넣어도 out h: 0, update 안됨

→dead cell 한번이면 복원, 중독 될이경우!

변형들 - linear

Leaky ReLU

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0.01x & \text{otherwise} \end{cases}$$

살려는 드릴게!

조금이라 보수인 유저들로 주는 보수임게 반들어볼수 있지 않을까?

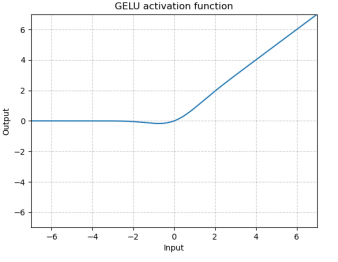
Parametric ReLU (PReLU)

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ ax & \text{otherwise.} \end{cases}$$

변형들 – non-linear

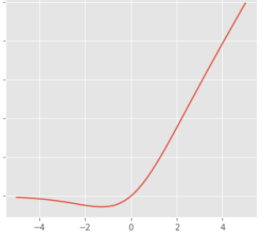
Gaussian Error Linear Unit (GELU)

$$f(x) = x \cdot \Phi(x),$$



Sigmoid Linear Unit (SiLU,Swish)

$$f(x) = x \cdot \mathrm{sigmoid}(x)$$



GELU — PyTorch 1.10.1 documentation

Introduction to Activation Function - Subinium의 코딩일지

변형들 – non-linear

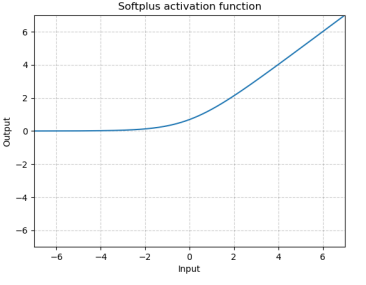
Softplus(SmoothReLU)

$$f(x) = \ln(1 + e^x),$$

$$f'(x) = \frac{e^x}{1+e^x} : \text{Sigmoid}$$

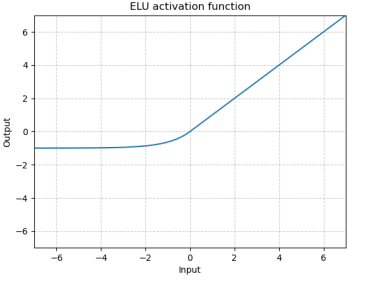
$x \to \infty : 1$

$x \to -\infty : 0$



ELU(Exponential Linear Unit)

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ a\left(e^x - 1\right) & \text{otherwise,} \end{cases}$$



Try to make the mean activations closer to 0.

ReLU보다 분류성능 향상.

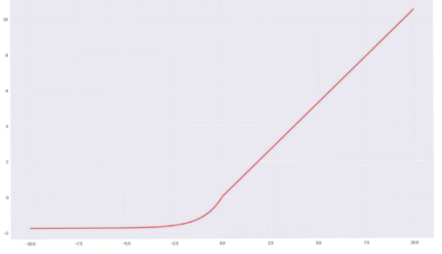
계산 오래걸린다.

Softplus — PyTorch 1.10.1 documentation

ELU — PyTorch 1.10.1 documentation

변형들 - nonlinear

SELU(Scaled Exponential Linear Unit)



$$\mathrm{selu}(x) = \lambda \begin{cases} x & \text{if } x > 0 \\ \alpha e^x - \alpha & \text{if } x \leqslant 0 \end{cases}$$

Self-normalizing : automatically converges to zero mean and unit variance

when
α=1.67…
λ=1.05…

[1706.02515] Self-Normalizing Neural Networks (arxiv.org)